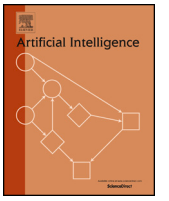
Artificial Intelligence 331 (2024) 104129

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Polarized message-passing in graph neural networks

Tiantian He [a,b], Yang Liu [c,*], Yew-Soon Ong [a,d], Xiaohu Wu [e], Xin Luo [f]

[a] Center for Frontier AI Research, Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore
[b] Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore
[c] Department of Computer Science, Hong Kong Baptist University, Hong Kong Special Administrative Region, PR China
[d] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[e] National Engineering Research Center of Mobile Network Technologies, Beijing University of Posts and Telecommunications, PR China
[f] College of Computer and Information Science, Southwest University, PR China

ARTICLE INFO

Keywords:
Graph neural networks
Message-passing graph neural networks
Representation learning
Graph analysis

ABSTRACT

In this paper, we present Polarized message-passing (PMP), a novel paradigm to revolutionize the design of message-passing graph neural networks (GNNs). In contrast to existing methods, PMP captures the power of node-node similarity and dissimilarity to acquire dual sources of messages from neighbors. The messages are then coalesced to enable GNNs to learn expressive representations from sparse but strongly correlated neighbors. Three novel GNNs based on the PMP paradigm, namely PMP graph convolutional network (PMP-GCN), PMP graph attention network (PMP-GAT), and PMP graph PageRank network (PMP-GPN) are proposed to perform various downstream tasks. Theoretical analysis is also conducted to verify the high expressiveness of the proposed PMP-based GNNs. In addition, an empirical study of five learning tasks based on 12 real-world datasets is conducted to validate the performances of PMP-GCN, PMP-GAT, and PMP-GPN. The proposed PMP-GCN, PMP-GAT, and PMP-GPN outperform numerous strong message-passing GNNs across all five learning tasks, demonstrating the effectiveness of the proposed PMP paradigm.

1. Introduction

Message-passing graph neural networks (MPGNNs) [1,2] are prominent tools for analyzing graph-structured data. MPGNNs heavily rely on a two-stage message-passing paradigm to learn representations for diverse downstream tasks. In the first stage, messages (i.e., node features) are conveyed to each central node from all of its neighbors. Simultaneously, the correlation (similarity) of each neighbor to the central node is computed, typically as weights, such as attention coefficients [3,4] or transition probabilities [5], and transmitted to the central node. In the second stage, the representation of each central node is generated through the weighted aggregation of received messages.

Diverse properties of real-world graphs, such as the friendship paradox [6] in social graphs and impact imbalance [7] in citation graphs, have demonstrated that many nodes are created differently. However, the message-passing paradigms followed by most MPGNNs focus on conveying similarity-based messages, that is, the representation is learned through the aggregation of the features of neighbors, in which the relevant neighbors are determined by some similarity measures. Properties regarding dissimilarity in the graphs are consequently under-exploited by existing MPGNNs.

* Corresponding author.
E-mail address: csygliu@comp.hkbu.edu.hk (Y. Liu).

https://doi.org/10.1016/j.artint.2024.104129
Received 22 June 2023; Received in revised form 9 January 2024; Accepted 24 March 2024

Available online 27 March 2024

In this paper, we hypothesize that fully exploiting the properties of dissimilarity present in graph data can significantly enhance the learning performance of MPGNNs. To this end, we propose Polarized message-passing (PMP) for graph neural networks (GNNs). Unlike the learning paradigm adopted by conventional MPGNNs, PMP allows for the concurrent propagation of similarity- and dissimilarity-based messages. By leveraging the composite effect of both similarity and dissimilarity learned from graph data, PMP is capable of preserving, reducing, or negating the messages conveyed from neighboring nodes. PMP-based GNNs can learn more expressive representations by largely concentrating on a significantly reduced number of strongly correlated neighbors.

To develop the PMP paradigm, we introduce the following two technical contributions. First, the weight matrix used by the layers in conventional MPGNNs to aggregate the features from neighboring nodes is decomposed into two separate learnable matrices. The first matrix can capture the correlations between each central node and its neighbors (e.g., node-node similarities regarding features). This matrix can reflect how the messages from neighboring nodes should be preserved. The second matrix can acquire the difference between nodes (i.e., the exponential of negative node-node distances), indicating whether the messages from neighbors should be reduced or negated.

Second, we construct a new sparse matrix that enables the composite propagation of similarity- and dissimilarity-based messages (i.e., polarized messages) in GNNs. The corresponding entries of this matrix are calculated as the normalized products of the matrices pertaining to similarity and dissimilarity. With the newly constructed sparse matrix, PMP can learn to preserve, reduce, or negate the messages. More expressive representations can be learned through the sparse aggregation of the messages from neighbors.

The main contributions of this paper can be summarized as follows.

- We present a novel learning paradigm, named PMP, for GNNs. In contrast to conventional message-passing paradigms, PMP allows the concurrent propagation of similarity- and dissimilarity-based messages, enabling GNNs to learn to preserve, reduce, or negate the messages conveyed by neighbors.
- We propose three novel PMP-based GNNs, namely PMP graph convolutional network (PMP-GCN), PMP graph attention network (PMP-GAT), and PMP graph PageRank network (PMP-GPN). The GNNs are designed for learning more expressive representations for various downstream tasks by largely concentrating on sparse but strongly correlated neighbors.
- We conduct a comprehensive theoretical analysis, showing that the proposed PMP-GCN, PMP-GAT, and PMP-GPN are more expressive than representative MPGNNs.
- The proposed PMP-based GNNs are tested on five learning tasks, namely scientific article classification, blog and content-based image clustering, coauthor analysis, and rich-text classification from 12 real-world datasets. The notable results demonstrate the effectiveness of the proposed PMP.

2. The polarized message-passing paradigm

In this section, the PMP paradigm is first elaborated. The theoretical analysis demonstrating that the proposed PMP paradigm is more expressive than those adopted by conventional MPGNNs is then presented.

2.1. Notations

In this paper, a graph containing N nodes and $|E|$ edges is denoted as $G = \{V, E, \mathbf{X}\}$, where V, E, and $\mathbf{X} \in \mathbb{R}^{N \times D}$ represent the node and edge sets and the input node feature matrix, respectively. Each node in G belongs to one of the C classes, and their one-hop neighbors form a set denoted as $\mathcal{N}_i$. $\mathbf{A} \in \{0,1\}^{N \times N}$ denotes the adjacency matrix of G. $\mathbf{W}^l$, $\mathbf{H}^l$, and $\mathbf{P}^l$ represent the matrices of learnable weights, node features, and latent positions at the lth layer. Thus, $\mathbf{H}^1 = \mathbf{X}$.

2.2. Formulating polarized message-passing at a single layer

At each layer of conventional MPGNNs, the output representations for each node are generated through the addition of the node feature and the messages received from its neighbors [8,9]. The message from each neighbor is typically formulated through the multiplication of the neighboring feature and the weight obtained via a similarity/correlation strategy. However, many phenomena frequently observed in real-world graphs, e.g., the friendship paradox, have demonstrated that many nodes in the graph are created differently. These differences in the characteristics of nodes in real-world graphs, which are contrary to the notion of similarity widely considered by existing GNNs, have not been well explored. Therefore, in this paper, we propose PMP, which can leverage the composite effect of both similarity and dissimilarity to learn to preserve, reduce, and negate the messages from neighbors. PMP-based GNNs can learn more expressive representations from a significantly reduced number of correlated nodes in the graph.

Different from existing message-passing strategies used by conventional GNNs, PMP constructs two learnable matrices, $\mathbf{C}$ and $\mathbf{S}$, to quantify the correlation and differentiation between pairs of nodes, respectively. A novel matrix $\mathbf{M}$ is then constructed through the combination of the joint effect of $\mathbf{C}$ and $\mathbf{S}$. Finally, polarized messages from neighbors are composed through the multiplication of $\mathbf{M}$-induced weights and the corresponding features of neighbors.

Computing $\mathbf{C}$ *and* $\mathbf{S}$ To quantify the node correlations ($\mathbf{C}$) at a layer, PMP computes the feature and structural similarities between pairwise nodes that are connected:

$$\begin{aligned} &\mathbf{H} = \mathbf{H}^l \mathbf{W}_h^l, \mathbf{P} = \mathbf{P}^l \mathbf{W}_p^l, \\ &\mathbf{C}_{ij} = \mathbf{A}_{ij} \cdot [\mathbf{a}(\mathbf{H}_{i,:} || \mathbf{H}_{j,:})^T + \mathbf{P}_{i,:} diag(\mathbf{b}) \mathbf{P}_{j,:}^T], \end{aligned} \tag{1}$$

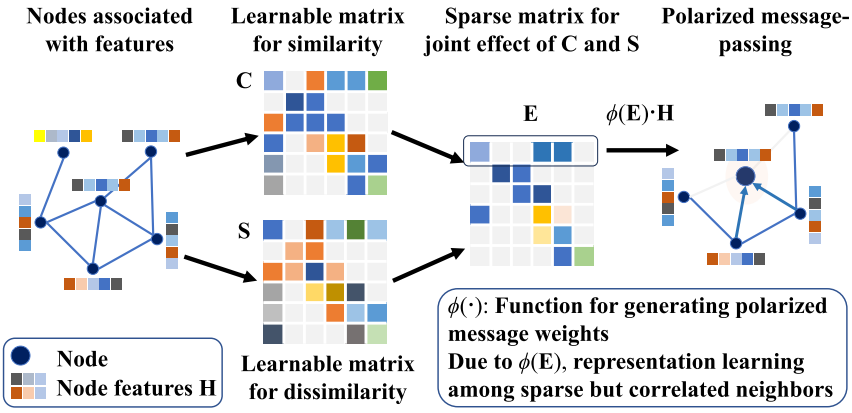


Fig. 1. Graphical view of the proposed polarized message-passing (PMP) paradigm. By appropriately merging neighboring similarity- and dissimilarity-based messages, PMP allows GNNs to learn more expressive representations with sparse but strongly correlated neighbors.

where $\mathbf{H}_{i,:}$ and $\mathbf{P}_{i,:}$ are row vectors representing the feature and latent positions of node i, $||$ is the operator for vector concatenation, $diag(\cdot)$ is the diagonal matrix, and $\mathbf{a}$ and $\mathbf{b}$ are vectors of learnable parameters. To quantify the node dissimilarities, PMP computes the weighted distances in terms of feature and structure:

$$\mathbf{S}_{ij} = \mathbf{A}_{ij} \cdot [\alpha|\mathbf{H}\mathbf{W}_{i,:} - (\mathbf{H}\mathbf{W})_{j,:}|^2 + (1-\alpha)|\mathbf{P}_{i,:} - \mathbf{P}_{j,:}|^2], \tag{2}$$

where $\alpha \in (0,1)$ is a learnable parameter balancing the relative significance of distances caused by the node feature and the graph structure.

Coalescing polarized messages for representation learning To learn the output representation at each layer, the polarized messages can be directly generated according to the weights induced by $\mathbf{C}$ and $\mathbf{S}$. However, this procedure is computationally demanding as it requires double computation, that is, the multiplication of the weights induced by $\mathbf{C}$ and $\mathbf{S}$ and the features of all neighbors. In this paper, we propose an alternative procedure to efficiently generate polarized messages. The joint effect of $\mathbf{C}$ and $\mathbf{S}$ is first computed. The output representations can then be learned using messages and weights derived from the joint effect. To achieve this, we introduce a new sparse matrix denoted as $\mathbf{E}$, which is derived from $\mathbf{C}$ and $\mathbf{S}$. The derived $\mathbf{E}$ is utilized to generate polarized messages, which play a crucial role in the learning of output representations. Specifically, $\mathbf{E}$ is the element-wise exponential of the difference between $\mathbf{C}$ and $\mathbf{S}$. This straightforward strategy for the derivation of $\mathbf{E}$ (i.e., taking the element-wise exponential of the difference between $\mathbf{C}$ and $\mathbf{S}$) has been found to perform quite well in various downstream tasks, although complicated alternative strategies [42] are available. With $\mathbf{E}$ at hand, PMP can be established for a graph neural layer (See Fig. 1 for a graphical view). The output representations for each node (say i) are acquired following the generic procedure presented below:

$$\begin{aligned}
&\text{FEATURE MAPPING: } \mathbf{H} = \mathbf{H}^l\mathbf{W}_h^l, \mathbf{P} = \mathbf{P}^l\mathbf{W}_p^l,\\
&\text{POLARIZED MESSAGE: Compute } \mathbf{C} \text{ and } \mathbf{S},\\
&\qquad \mathbf{E}_{ij} = \mathbf{A}_{ij} \cdot \exp(\mathbf{C}_{ij} - \beta\mathbf{S}_{ij}),\\
&\qquad \mathbf{M}_i = \sum_{j\in\mathcal{N}_i} \phi(\mathbf{E}_{ij}) \cdot \mathbf{H}_{j,:},\\
&\text{FEATURE UPDATE: } \mathbf{H}_{i,:}^{l+1} = \frac{\theta}{|\mathcal{N}_i|}\mathbf{H}_{i,:} + \mathbf{M}_i,
\end{aligned} \tag{3}$$

where $\phi(\cdot)$ is an appropriately function leveraging $\mathbf{E}$ to generate the weights for computing messages from neighbors, β is a learnable positive parameter, and θ is a learnable positive irrational that is independent of the learning of M_i. Given Eq. (3), the proposed PMP paradigm for GNNs substantially differs from those used in existing MPGNNs in the following three aspects. First, the messages generated by $\mathbf{E}$ are more comprehensive, as $\mathbf{E}$ considers the joint effect of similarities and dissimilarities between nodes in the graph. Second, sparse messages from neighbors can be learned with $\mathbf{E}$, as it considers the differences between correlations and dissimilarities. Lastly, the proposed PMP is highly flexible for constructing diverse graph neural layers, such as graph convolution layers and attention layers, by properly setting $\phi(\cdot)$. This enables PMP-based GNNs to effectively tackle a wide range of downstream tasks.

2.3. Expressiveness of polarized message-passing

As shown in Eq. (3), capturing the joint effect of similarity and dissimilarity between connected nodes for the composition of polarized messages is a core step of PMP. This step endows PMP-based GNNs with the capability of learning more expressive

representations, regardless of the selection of feature aggregation strategy. Here, a qualitative analysis is conducted to demonstrate such superior capability possessed by PMP-based GNNs, compared with existing MPGNNs.

Learning tasks performed by GNNs typically concern predicting the labels of test data samples, given the representations learned from the training set. Let the ground-truth label be Y. Given a learning task, a GNN performs it by conceptually maximizing the following conditional likelihood $p(Y|\mathbf{z})$, where $\mathbf{z}$ is the representation learned by a GNN. We know that conventional MPGNNs learn representations mainly with similarity/correlation-based messages. To learn the representation $\mathbf{z}$, a conventional MPGNN has to utilize the message set generated for each central node in the graph. Thus, for node i, its representation can be represented as $\mathbf{z}_i = f_\psi(C_i)$, where C_i denotes the set of similarity-based messages for node i and $f_\psi(\cdot)$ represents the function for representation learning. And the objective function for representation learning can be conceptually formulated as follows:

$$\min_{\psi} \mathbb{E}_{i\sim I_{train}, C_i\sim p(C)}[L(Y_i, f_\psi(C_i))], \tag{4}$$

where L represents a function measuring the divergence between Y_i and $f_\psi(\cdot)$.

Given Eq. (3), we know that PMP-based GNNs leverage polarized messages to learn representations for downstream tasks. For node i, the representation learned by a PMP-based GNN differs from that learned by a conventional MPGNN: $\mathbf{z}_i = f_\psi(C_i \oplus S_i)$, where S_i denotes the set of dissimilarity-based messages for node i, and $\oplus$ represents the strategy of merging polarized messages. Thus, the objective function for a PMP-based GNN can be formulated as follows:

$$\min_{\psi} \mathbb{E}_{i\sim I_{train}, C_i\sim p(C), S_i\sim p(S)}[L(Y_i, f_\psi(C_i \oplus S_i))]. \tag{5}$$

Optimizing Eqs. (4) and (5) produces node representations drawn from the distribution of C and the joint distribution of C and S, respectively. Thus, analyzing the mutual information between ground-truth labels and learned representations in Eqs. (4) and (5) can quantify the expressiveness of the proposed PMP and paradigms adopted in conventional MPGNNs. Let $f_\psi(\cdot)$ be a function for representation learning. The following theorem demonstrates that a GNN equipped with the proposed PMP in Eq. (3) is more expressive than conventional MPGNNs.

Theorem 1. *Suppose $f_\psi(\cdot)$ is a learnable function for representation learning adopted by PMP-based GNNs and conventional MPGNNs, $C = \{C_i|i = 1, ..., n\}$, and $S = \{S_i|i = 1, ..., n\}$ denote the sets of similarity and dissimilarity messages that are used to learn node representations, and $Y = \{Y_i|i = 1, ..., n\}$ denotes the set of node labels, where n denotes the total number of nodes for training. The representations learned by PMP-based GNNs and conventional MPGNNs are denoted as $\mathbf{z}_i^{PMP} = f_\psi(C_i \oplus S_i), i = 1, ..., n$, and $\mathbf{z}_i^{MP} = f_\psi(C_i), i = 1, ..., n$, respectively. For any $i \in \{1, ..., n\}$, the following inequality holds:*

$$I(Y_i; \mathbf{z}_i^{PMP}) \geq I(Y_i; \mathbf{z}_i^{MP}), \tag{6}$$

where $I(\cdot;\cdot)$ represents the function of mutual information.

We leave all proofs in Section 4 to keep the content uncluttered. Theorem 1 demonstrates that PMP-based GNNs are more expressive in learning representations than conventional MPGNNs, as the mutual information between representations learned by PMP-based GNNs and ground-truth labels is higher than that between representations learned by conventional MPGNNs and ground-truth labels. Theorem 1 provides an information-theoretic analysis of the expressiveness gap between the proposed PMP and the message-passing paradigms used by conventional MPGNNs. Leveraging the joint effect of similarity- and dissimilarity-based messages enables PMP-based GNNs to be more likely to correctly predict the data labels, which indicates a significant performance gain of PMP-based GNNs against conventional MPGNNs.

3. Polarized message-passing graph neural networks

Using the proposed PMP paradigm (Eq. (3)), we can build GNNs (i.e., PMP-GNNs) to effectively perform various downstream tasks. As previously mentioned, the proposed PMP paradigm offers high flexibility in constructing diverse GNNs by appropriately setting $\phi(\cdot)$. In this section, we present three novel GNNs, namely PMP-GCN, PMP-GAT, and PMP-GPN. These three GNNs are designed to evaluate the efficacy of the proposed PMP paradigm, as demonstrated in the experiments. Moreover, our theoretical analysis demonstrates that the proposed PMP-GNNs are more expressive than conventional MPGNNs.

3.1. Polarized message-passing graph convolutional network

Graph convolutional networks (GCNs) [1,10] have been recognized as prominent MPGNNs and are used to tackle a wide range of predictive tasks in graph-structured data. However, most existing GCNs do not consider leveraging polarized messages to learn representations. Here, we present PMP-GCN, whose layers adopt the proposed message-passing paradigm to learn more expressive representations. The output representation is learned through the following procedure for each layer in PMP-GCN:

$$\begin{aligned}
&\mathbf{H} = \mathbf{H}^l \mathbf{W}_h^l, \mathbf{P} = \mathbf{P}^l \mathbf{W}_p^l, \mathbf{E}_{ij} = \mathbf{A}_{ij} \cdot \exp\left(\mathbf{C}_{ij} - \beta \mathbf{S}_{ij}\right), \\
&\mathbf{M}_i = \frac{1}{\sqrt{\sum_{k \in \mathcal{N}_i} \mathbf{E}_{ik}}} \sum_{j \in \mathcal{N}_i} \frac{\mathbf{E}_{ij}}{\sqrt{\sum_{k \in \mathcal{N}_j} \mathbf{E}_{jk}}} \cdot \mathbf{H}_{j,:}, \\
&\mathbf{H}_{i,}^{l+1} = \frac{\theta}{|\mathcal{N}_i|} \mathbf{H}_{i,:} + \mathbf{M}_i.
\end{aligned} \tag{7}$$

According to Eq. (7), the following two aspects differentiate the proposed PMP-GCN from conventional GCNs. First, the weights utilized by PMP-GCN to compose messages for representation learning (i.e., the graph Laplacian) are learnable rather than fixed as in most conventional GCNs. This enables PMP-GCN to effectively capture the structural information at each layer, thereby enhancing its learning capability. Second, as $\mathbf{E}_{ij}$ is computed using $\mathbf{C}$ and $\mathbf{S}$, neighbors considerably different from the central node are assigned very low weights. The importance of subsequent messages can be reduced or even negated during the generation of output representations. Thus, the convolutional operator in PMP-GCN can learn representations from the preserved messages from strongly correlated neighbors.

3.2. Polarized message-passing graph attention network

Graph attention networks (GATs) [3,11] are also typical MPGNNs. At each layer of a conventional GAT, the weights used to generate messages are always computed solely based on node correlations. To endow attention-based GNNs with the capability of learning representations with polarized messages, we present PMP-GAT. For each layer in PMP-GAT, the output representation for each node is learned via the following procedure:

$$\begin{aligned}
&\mathbf{H} = \mathbf{H}^l \mathbf{W}_h^l, \mathbf{P} = \mathbf{P}^l \mathbf{W}_p^l, \mathbf{E}_{ij} = \mathbf{A}_{ij} \cdot \exp\left(\mathbf{C}_{ij} - \beta \mathbf{S}_{ij}\right), \\
&\mathbf{M}_i = \sum_{j \in \mathcal{N}_i} \frac{\mathbf{E}_{ij}}{\sum_{k \in \mathcal{N}_i} \mathbf{E}_{ik}} \cdot \mathbf{H}_{j,:}, \\
&\mathbf{H}_{i,}^{l+1} = \frac{\theta}{|\mathcal{N}_i|} \mathbf{H}_{i,:} + \mathbf{M}_i.
\end{aligned} \tag{8}$$

Compared with conventional GATs, PMP-GAT pays less attention to or even ignores dissimilar neighbors when aggregating features from neighbors, owing to the effect of $\mathbf{E}$. This is equivalent to reducing or negating the messages conveyed by neighbors during the generation of output representations. PMP-GAT, therefore, can learn representations by focusing on messages from truly correlated neighbors.

3.3. Polarized message-passing graph PageRank network

Graph PageRank networks (GPNs), such as APPNP [12] and GPR [13], are also powerful MPGNNs. At each layer, the representation is typically generated by summing over the initial features of each central node and features computed by conventional graph convolution. Most PageRank GNNs do not consider leveraging polarized messages to learn representations. Here, we propose PMP-GPN to endow Graph PageRank networks with the capability to learn representations with polarized messages. PMP-GPN is mainly built upon APPNP. For each layer of PMP-GPN, the output representation of each node is generated through the following procedure:

$$\begin{aligned}
&\mathbf{H}^{in} = MLP(\mathbf{X}), \mathbf{P}^{in} = MLP(\mathbf{P}^0), \mathbf{E}_{ij} = \mathbf{A}_{ij} \cdot \exp\left(\mathbf{C}_{ij} - \beta \mathbf{S}_{ij}\right), \\
&\mathbf{M}_i = \frac{1}{\sqrt{\sum_{k \in \mathcal{N}_i} \mathbf{E}_{ik}}} \sum_{j \in \mathcal{N}_i} \frac{\mathbf{E}_{ij}}{\sqrt{\sum_{k \in \mathcal{N}_j} \mathbf{E}_{jk}}} \cdot \mathbf{H}_{j,:}, \\
&\mathbf{H}_{i,}^{l+1} = \frac{\theta}{|\mathcal{N}_i|} \mathbf{H}_{i,:} + (1-\alpha)\mathbf{M}_i + \alpha \mathbf{H}_{i,:}^{in},
\end{aligned} \tag{9}$$

where $\mathbf{H}^{in}$ and $\mathbf{P}^{in}$ are the node features and latent positions learned by multi-layer perception, and α is a small positive number representing the teleport probability used to control the neighborhood influence. In PMP-GPN, all layers share the same $\mathbf{E}$, which is computed using $\mathbf{H}^{in}$ and $\mathbf{P}^{in}$, for efficient purposes. In contrast to existing PageRank GNNs, the weights for composing messages for representation learning are learnable rather than fixed. Similar to PMP-GCN, neighbors that significantly differ from the central node are assigned very low weights, leading to very limited or no influence on the subsequent generation of output representations. Thus, PMP-GPN can learn representations by concentrating on multi-hop message-passing among strongly correlated neighbors.

3.4. Loss function

After message passing over multiple layers, PMP-GNNs learn the output representations, which are finally fed into the training loss to optimize the network parameters. Two modules constitute the loss function of PMP-GNNs, including task-specific loss and latent-position loss. The loss function can be conceptually formulated as follows:

$$L = L_{task} + tr(\mathbf{P}^{outT}\mathbf{L}\mathbf{P}^{out}), \tag{10}$$

where $\mathbf{L}$ is the Laplacian matrix of $\mathbf{A}$, and $\mathbf{P}^{out}$ represents the output latent positions of all nodes in the graph.

3.5. Computational complexity of PMP-GNNs

The computational complexity of the proposed PMP-GNNs is similar to that of conventional MPGNNs. Assuming the input and output feature dimensions are d and d', for each layer in the PMP-GNN, the complexity of feature mapping is approximately $\mathcal{O}(2Ndd')$. The complexity of computing $\mathbf{C}$ and $\mathbf{S}$ is about $\mathcal{O}(4|E|d')$. The complexity of composing polarized messages and learning representations is approximately $\mathcal{O}(|E|(1+d'))$. Therefore, the overall complexity of each layer in PMP-GCN and PMP-GAT is about $\mathcal{O}(|E|(1+5d') + 2Ndd')$. The complexity of each layer in PMP-GPN is approximately $\mathcal{O}(|E|d)$ as PMP-GPN completes feature mapping and computing $\mathbf{C}$ and $\mathbf{S}$ before learning representations (i.e., $d = d'$). Considering the complexity of conventional GCN, GAT and APPNP are about $\mathcal{O}(2|E|d' + Ndd')$, $\mathcal{O}(4|E|d' + Ndd')$, and $\mathcal{O}(|E|d)$, the complexity of the proposed PMP-GNNs allows them to perform various learning tasks on massive datasets.

3.6. Expressiveness of PMP-GNNs

The expressiveness of GNNs [14–16] has drawn considerable attention recently. It concerns whether a GNN layer can acquire distinct representations for diverse (sub-) graph structures. Previous studies have demonstrated that GNNs with greater expressiveness can perform better in downstream learning tasks [17,18]. Accordingly, a comprehensive analysis is conducted here to demonstrate the greater expressiveness of the proposed PMP-GCN, PMP-GAT, and PMP-GPN compared to conventional MPGNNs.

For all MPGNNs, the upper limit of expressiveness is the one-dimensional Weisfeiler-Lehman test (1-WL test) [17,19]. Similar to what the 1-WL test performs in the Weisfeiler-Lehman algorithm, this test verifies whether an aggregation function in a GNN layer can successfully differentiate all distinct (sub-) graph structures with node features. Therefore, analyzing the expressiveness of PMP-GNNs lies in whether the process of feature aggregation for representation learning can discriminate all different graph structures. In this paper, we derive the following theorem to establish the expressiveness of the proposed PMP-GNNs when they learn representations with different graph structures.

Theorem 2. *Assume the node features of two graphs are denoted as two multi-sets $X_i = (M_i, \mu_i)$ and $X_j = (M_j, \mu_j)$, $X_i, X_j \in \mathcal{X}$, where $\mathcal{X}$ represent the countable feature space, M_i is the underlying set containing the distinct elements in X_i, and $\mu_i \in \mathbb{N}^*$ stands for the number of repeating times of each element in M_i, c_i and c_j represent the features of the central nodes in these two graphs. The process of feature aggregation for c_i and c_j at each layer of PMP-GCN, PMP-GAT, and PMP-GPN is denoted as $h(c_i, X_i)$ and $h(c_j, X_j)$. For any two graphs with distinct structures, the following inequality holds:*

$$h(c_i, X_i) \neq h(c_j, X_j). \tag{11}$$

Theorem 2 shows that the proposed PMP-GNNs can discriminate all diverse (sub-) graph structures characterized by distinct node features. This implies that PMP-GNNs are more expressive than most MPGNNs, whose aggregation functions do not meet the criteria of the 1-WL test. The strong expressiveness of the proposed PMP-GNNs suggests they can perform superiorly in diverse downstream tasks.

4. Theoretical analysis

In this section, all theorems revealing the superior expressiveness of the proposed PMP and PMP-GNNs are proved.

4.1. Proof of Theorem 1

Proof. To prove Theorem 1, the following properties of mutual information and entropy will be used:

$$\begin{aligned} &I(x;y) = H(y) - H(y|x), \\ &I(x;z|y) = H(x|y) - H(x|y,z), \\ &H(x,y) = H(x|y) + H(y), \end{aligned} \tag{12}$$

where $I(\cdot;\cdot)$ represents the mutual information between random variables such as x, y, and z, $H(\cdot)$ represents the marginal entropy, and $H(\cdot|\cdot)$ represents the conditional entropy. Let $I(Y_i; \mathbf{z}_i^{PMP})$ be the mutual information between the ground-truth label Y_i and the representation $\mathbf{z}_i^{PMP} = f_\psi(C_i \oplus S_i)$ that a PMP-GNN learns from the polarized message sets of C_i and S_i. Given Eq. (3), we know that the proposed PMP first captures the joint effect of C_i and S_i, and then leverages the shared function $\phi(\cdot)$ to compose messages for representation learning. The above procedure is equivalent to directly leveraging C_i and S_i to formulate dual messages for the subsequent learning of representations. Thus, $I(Y_i; f_\psi(C_i \oplus S_i))$ can be rewritten as $I(Y_i; f_\psi(C_i), f_\psi(S_i))$, where $(f_\psi(C_i)$ and $f_\psi(S_i))$ are drawn from a joint distribution. Given $I(Y_i; f_\psi(C_i), f_\psi(S_i))$, we have:

$$
\begin{aligned}
&I(Y_i; f_\psi(C_i), f_\psi(S_i)) = H(f_\psi(C_i), f_\psi(S_i)) - H(f_\psi(C_i), f_\psi(S_i)|Y_i),\\
&= H(f_\psi(C_i), f_\psi(S_i)) + H(Y_i) - H(f_\psi(C_i), f_\psi(S_i), Y_i)\\
&= H(Y_i) - H(Y_i|f_\psi(C_i), f_\psi(S_i))\\
&= H(Y_i) - H(Y_i|f_\psi(C_i)) + H(Y_i|f_\psi(C_i)) - H(Y_i|f_\psi(C_i), f(S_i))\\
&= I(Y_i; f_\psi(C_i)) + I(Y_i; f_\psi(S_i)|f_\psi(C_i)) \geq I(Y_i; f_\psi(C_i)),
\end{aligned}
\tag{13}
$$

where $I(Y_i; f_\psi(C_i))$ is the mutual information between Y_i and the representation learned by conventional MPGNNs ($\mathbf{z}_i^{MP}$). Mutual information is non-negative, so the above inequality always holds for any $f_\psi(\cdot)$. Since dissimilarities widely exist in graph data and both C_i and S_i are determined by the same set of node features, $I(Y_i; f_\psi(S_i)|f_\psi(C_i))$ is probably larger than zero, leading to a greater gap between $I(Y_i; f_\psi(C_i), f_\psi(S_i))$ and $I(Y_i; f_\psi(C_i))$ for any node in the graph. □

The proof of Theorem 1 demonstrates that PMP-GNNs are constantly more likely to correctly predict node labels in the graph than conventional MPGNNs as long as $f_\psi(\cdot)$ adopted by PMP-GNN and MPGNNs is well defined.

4.2. Proof of Theorem 2

Before proving Theorem 2, we first introduce necessary notations and preliminaries. For each central node and its neighbors, their features form a multi-set $X = (M, \mu)$, where $X \in \mathcal{X}$, $\mathcal{X}$ is a countable feature space, M is the underlying set containing the distinct elements in X, and $\mu \in \mathbb{N}^*$ stands the number of repeating times of each element in M. To learn representations in each layer, a PMP-GNN (i.e., PMP-GCN, PMP-GAT, and PMP-GPN) aggregates the polarized messages from neighbors according to the message weights. For each central node in the graph, this learning procedure is conceptually represented as follows:

$$
\begin{aligned}
&\mathbf{E}_{cx} = \exp\{a_{cx} + b_{cx} - \beta(k_{cx} + l_{cx})\},\\
&w_{cx} = \phi(\mathbf{E}_{cx}),\\
&h(c, X) = \frac{\theta}{|X|} g(c) + \sum_{x \in M} \sum_{\substack{y=x,\\ y \in X}} w_{cy} g(y),
\end{aligned}
\tag{14}
$$

where c and x represent the features of the central node and neighbor, a_{cx} and b_{cx} represent the feature and structural similarities between the central node and a neighbor, k_{cx} and l_{cx} represent the feature and structural dissimilarities between the central node and a neighbor, $\phi(\cdot)$ is the function for computing the message weight, θ is a positive irrational independent of the message weights, $|X|$ is the size of the multi-set, and $g(\cdot)$ represents a valid mapping function in the countable feature space, respectively.

4.2.1. Schematic methodology to complete the proof of Theorem 2

To prove Theorem 2, that is, the learning procedure in Eq. (14) can discriminate all distinct graph structures, we have to validate that the learned representations for each pair of different graphs are distinct:

$$
h(c_i, X_i) \neq h(c_j, X_j), \text{ if } |X_i| \neq |X_j|, \forall i \neq j. \tag{15}
$$

In general, Theorem 2 can be verified through *proof by contradiction*, which is a formal framework widely used to verify whether a GNN layer can pass the 1-WL test [15,17]. Accordingly, the generic procedure for completing the proof can be described as follows. First, we can assume the statement shown in Eq. (15) is FALSE. In other words, a layer in each PMP-GNN learns an identical representation for two central nodes with different structures. Second, we identify all possible conditions that determine the aggregation function ($h(c, X)$) of the PMP-GNN to generate the representation and assume the statement shown in Eq. (15) is FALSE when these conditions are met. At last, we can complete the proof by demonstrating that when all the possible conditions are met, the assumption that Eq. (15) is FALSE can result in some easy-to-verify contradictions.

Given Eqs. (7)-(9), and (14), we can easily identify that the following two factors determine how $h(c, X)$ generates representations at each PMP-GNN layer, including the underlying set M, and the feature of the central node c. Thus, the representations of a pair of central nodes generated by each PMP layer can be analyzed in the following four cases, including $M_1 \neq M_2$ and $c_1 \neq c_2$, $M_1 \neq M_2$ but $c_1 = c_2$, $M_1 = M_2$ but $c_1 \neq c_2$, and $M_1 = M_2$ and $c_1 = c_2$. In what follows, we will prove that all the proposed PMP-GNNs can pass the 1-WL test by showing that the assumption that Eq. (15) is FALSE can result in contradictions in all the previously enumerated cases.

4.2.2. The equivalence between the PMP-GCN layer and the 1-WL test

Proof. We can complete the proof by following the procedure introduced in Section 4.2.1. Let us first assume the statement regarding PMP-GCN in Theorem 2 is FALSE, that is, there exists a pair of graph structures X_1 and X_2 with central node features c_1 and c_2, that the PMP-GCN layer can NOT discriminate. Based on Eq. (14), this assumption can be written as follows:

$$\begin{aligned}
&\mathbf{E}_{c_i x} = \exp\{a_{c_i x} + b_{c_i x} - \beta(k_{c_i x} + l_{c_i x})\},\\
&w_{c_i x} = \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x\in X_i}\mathbf{E}_{c_i x}\cdot\sum_{z\in X_y}\mathbf{E}_{yz}}},\\
&h(c_1, X_1) = h(c_2, X_2), |X_1| \neq |X_2|,
\end{aligned} \tag{16}$$

where $h(,)$ is defined as Eq. (7) shows. As mentioned previously, we know that the representations for c_1 and c_2 generated by a PMP-GCN layer can be analyzed in four different cases, including $M_1 \neq M_2$ and $c_1 \neq c_2$, $M_1 \neq M_2$ but $c_1 = c_2$, $M_1 = M_2$ but $c_1 \neq c_2$, and $M_1 = M_2$ and $c_1 = c_2$. We will then demonstrate that $h(c_1, X_1) = h(c_2, X_2)$ can not hold as possible contradictions are identified under all these four cases.

$M_1 \neq M_2$ and $c_1 \neq c_2$, or $M_1 \neq M_2$ but $c_1 = c_2$ Here, we analyze the representations generated by a PMP-GCN layer when $M_1 \neq M_2$. If $h(c_1, X_1) = h(c_2, X_2)$, we have:

$$\begin{aligned}
h(c_1, X_1) &= \frac{\theta}{|X_1|}g(c_1) + \sum_{x\in M_1}\sum_{\substack{y=x,\\ y\in X_1}} w_{c_1 y}g(y)\\
&= \frac{\theta}{|X_2|}g(c_2) + \sum_{x\in M_2}\sum_{\substack{y=x,\\ y\in X_2}} w_{c_2 y}g(y) = h(c_2, X_2).
\end{aligned} \tag{17}$$

As each $w_{c_i x}$ is positive, each distinct element in either M_1 or M_2 will contribute to the representations of c_1 and c_2. We deduce $M_1 = M_2$ if $h(c_1, X_1) = h(c_2, X_2)$ and reach a contradiction. Thus, $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$ does not hold when $M_1 \neq M_2$.

$M_1 = M_2$ but $c_1 \neq c_2$ Next, we analyze the representations learned by the PMP-GCN layer when $M_1 = M_2$ but $c_1 \neq c_2$. Accordingly, we assume $M_1 = M_2 = M$, $c_1 \neq c_2$, and $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$. Thus, we have the following:

$$\begin{aligned}
&w_{c_i x} = \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x\in X_i}\mathbf{E}_{c_i x}\cdot\sum_{z\in X_y}\mathbf{E}_{yz}}},\\
&h(c_1, X_1) = \frac{\theta}{|X_1|}g(c_1) + \sum_{y\in X_1} w_{c_1 y}g(y)\\
&= \frac{\theta}{|X_2|}g(c_2) + \sum_{y\in X_2} w_{c_2 y}g(y) = h(c_2, X_2).
\end{aligned} \tag{18}$$

Given $M_1 = M_2 = M$, the message weights of each $x \in M$ composed of the representations of c_1 and c_2 should be identical if $h(c_1, X_1) = h(c_2, X_2)$. Denote the underlying sets for X_1 and X_2 as $M = \{c_1, c_2, ...\}$, and central node features as c_1 and c_2. For $x = c_1$, we have the following:

$$\begin{aligned}
&w_{c_i x} = \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x\in X_i}\mathbf{E}_{c_i x}\cdot\sum_{z\in X_y}\mathbf{E}_{yz}}},\\
&\frac{\theta}{|X_1|}g(c_1) + \sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y}g(y) = \sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y}g(y).
\end{aligned} \tag{19}$$

If $\sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y} = \sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y}$, we have:

$$\frac{\theta}{|X_1|} = 0, \tag{20}$$

which contradicts $\frac{\theta}{|X_1|} > 0$. If $\sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y} \neq \sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y}$, we have:

$$|X_1| = \frac{\theta}{\sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y} - \sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y}}. \tag{21}$$

As θ is independent of message weights, the RHS of the equation above can be an irrational number, which contradicts the LHS as a positive integer. Thus, $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$ does not hold when $M_1 = M_2$ but $c_1 \neq c_2$.

$M_1 = M_2$ and $c_1 = c_2$ At last, we analyze the representations learned by the PMP-GCN layer when $M_1 = M_2$ and $c_1 = c_2$. Similarly, we can assume that $h(c_1, X_1) = h(c_2, X_2)$, which is then shown to be FALSE due to contradictions. If $h(c_1, X_1) = h(c_2, X_2)$, the message weights of each $x \in M$ that are used to generate the representations of c_1 and c_2 should be the same. For an x in M, say $x = c$, we have the following:

$$
\begin{aligned}
&w_{cx} = \phi(\mathbf{E}_{cx}) = \frac{\mathbf{E}_{cx}}{\sqrt{\sum_{x\in X_i}\mathbf{E}_{cx}\cdot\sum_{z\in X_y}\mathbf{E}_{yz}}},\\
&\frac{\theta}{|X_1|}g(c) + \sum_{\substack{y=c,\\ y\in X_1}} w_{cy}g(y) = \frac{\theta}{|X_2|}g(c) + \sum_{\substack{y=c,\\ y\in X_2}} w_{cy}g(y).
\end{aligned}
\tag{22}
$$

Accordingly, Eq. (22) can be rewritten as follows:

$$
\frac{|X_1|}{|X_2|} = 1 + \frac{|X_1|}{\theta}[\sum_{\substack{y=c,\\ y\in X_1}} w_{cy} - \sum_{\substack{y=c,\\ y\in X_2}} w_{cy}].
\tag{23}
$$

The LHS of Eq. (23) is a positive rational number that is not equal to 1, which contradicts that the RHS is either 1 or an irrational number. Under all possible conditions, Eq. (23) does not hold. Combining the conducted analysis of all four cases, the assumption shown in Eq. (16) is FALSE. In other words, there does NOT exist a pair of graph structures that the PMP-GCN layer cannot distinguish. Therefore, any layer in a PMP-GCN can pass the 1-WL test. □

4.2.3. The equivalence between the PMP-GAT layer and the 1-WL test

Proof. Similarly, we can validate that any layer in a PMP-GAT can pass the 1-WL test via *proof by contradiction*. First, assume that Eq. (14) for PMP-GAT is FALSE, i.e., there exists a pair of graphs that the PMP-GAT layer cannot distinguish. This assumption can be written as follows:

$$
\begin{aligned}
&\mathbf{E}_{c_ix} = \exp\{a_{c_ix} + b_{c_ix} - \beta(k_{c_ix} + l_{c_ix})\},\\
&w_{c_ix} = \phi(\mathbf{E}_{c_ix}) = \frac{\mathbf{E}_{c_ix}}{\sum_{x\in X_i}\mathbf{E}_{c_ix}},\\
&h(c_1, X_1) = h(c_2, X_2), |X_1| \neq |X_2|,
\end{aligned}
\tag{24}
$$

where $h(,)$ is defined as Eq. (8) shows. We know that the representations for c_1 and c_2 generated by a PMP-GAT layer can be analyzed in four different cases, including $M_1 \neq M_2$ and $c_1 \neq c_2$, $M_1 \neq M_2$ but $c_1 = c_2$, $M_1 = M_2$ but $c_1 \neq c_2$, and $M_1 = M_2$ and $c_1 = c_2$. We will then demonstrate that $h(c_1, X_1) = h(c_2, X_2)$ can not hold as possible contradictions are identified under all these four cases.

$M_1 \neq M_2$ and $c_1 \neq c_2$ or $M_1 \neq M_2$ but $c_1 = c_2$ Let us first analyze the representations learned by a PMP-GAT layer when $M_1 \neq M_2$. If $h(c_1, X_1) = h(c_2, X_2)$, we have:

$$
\begin{aligned}
h(c_1, X_1) &= \frac{\theta}{|X_1|}g(c_1) + \sum_{x\in M_1}\sum_{\substack{y=x,\\ y\in X_1}} w_{c_1y}g(y)\\
&= \frac{\theta}{|X_2|}g(c_2) + \sum_{x\in M_2}\sum_{\substack{y=x,\\ y\in X_2}} w_{c_2y}g(y) = h(c_2, X_2).
\end{aligned}
\tag{25}
$$

As each w_{c_ix} is positive, each distinct element in either M_1 or M_2 will contribute to the representations of c_1 and c_2. We deduce $M_1 = M_2$ if $h(c_1, X_1) = h(c_2, X_2)$, which contradicts $M_1 \neq M_2$. Thus, $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$ does not hold when $M_1 \neq M_2$.

$M_1 = M_2$ but $c_1 \neq c_2$ Next, we analyze the representations learned by a PMP-GAT layer when $M_1 = M_2$ but $c_1 \neq c_2$. Accordingly, we assume $M_1 = M_2 = M$, $c_1 \neq c_2$, and $h(c_1, X_1) = h(c_2, X_2)$. According to Eq. (24), we have the following:

$$
\begin{aligned}
&w_{c_ix} = \phi(\mathbf{E}_{c_ix}) = \frac{\mathbf{E}_{c_ix}}{\sum_{x\in X_i}\mathbf{E}_{c_ix}},\\
&h(c_1, X_1) = \frac{\theta}{|X_1|}g(c_1) + \sum_{x\in M}\sum_{\substack{y=x,\\ y\in X_1}} w_{c_1y}g(y)\\
&= \frac{\theta}{|X_2|}g(c_2) + \sum_{x\in M}\sum_{\substack{y=x,\\ y\in X_2}} w_{c_2y}g(y) = h(c_2, X_2).
\end{aligned}
\tag{26}
$$

Given $M_1 = M_2 = M$, the message weights of each $x \in M$ composed of the representations of c_1 and c_2 should be identical if $h(c_1, X_1) = h(c_2, X_2)$. Denote the underlying sets for X_1 and X_2 as $M = \{c_1, c_2, ...\}$, and central node features as c_1 and c_2. For $x = c_1$, we have the following:

$$
\begin{aligned}
&w_{c_i x} = \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sum_{x \in X_i} \mathbf{E}_{c_i x}}, \\
&\frac{\theta}{|X_1|} g(c_1) + \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 x} g(y) = \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 x} g(y).
\end{aligned} \tag{27}
$$

If $\sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} = \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y}$, we have:

$$
\frac{\theta}{|X_1|} = 0, \tag{28}
$$

which contradicts $\frac{\theta}{|X_1|} > 0$. If $\sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} \neq \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y}$, we have:

$$
|X_1| = \frac{\theta}{\sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y} - \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y}}. \tag{29}
$$

The RHS of the equation above can be an irrational number, which contradicts the LHS as a positive integer. Thus, $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$ does not hold when $M_1 = M_2$ but $c_1 \neq c_2$.

$M_1 = M_2$ and $c_1 = c_2$ At last, we analyze the representations learned by the PMP-GAT layer when $M_1 = M_2$ and $c_1 = c_2$. If $h(c_1, X_1) = h(c_2, X_2)$, the message weights of each $x \in M$ that are used to generate the representations of c_1 and c_2 should be the same. Thus, for $x = c$ and $x \neq c$, we have the follows:

$$
\begin{aligned}
&\frac{\theta}{|X_1|} g(c) + \sum_{\substack{y=c, \\ y \in X_1}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_1} \mathbf{E}_{cx}} g(y) = \frac{\theta}{|X_2|} g(c) + \sum_{\substack{y=c, \\ y \in X_2}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_2} \mathbf{E}_{cx}} g(y), \\
&\sum_{\substack{x \neq c, \\ x \in M}} \sum_{\substack{y=x, \\ y \in X_1}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_1} \mathbf{E}_{cx}} g(y) = \sum_{\substack{x \neq c, \\ x \in M}} \sum_{\substack{y=x, \\ y \in X_2}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_2} \mathbf{E}_{cx}} g(y).
\end{aligned} \tag{30}
$$

Since the sum of attention coefficients equals 1, we have the following:

$$
\begin{aligned}
&\sum_{\substack{y=c, \\ y \in X_1}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_1} \mathbf{E}_{cx}} = 1 - \sum_{\substack{x \neq c, \\ x \in M}} \sum_{\substack{y=x, \\ y \in X_1}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_1} \mathbf{E}_{cx}} \\
&= 1 - \sum_{\substack{x \neq c, \\ x \in M}} \sum_{\substack{y=x, \\ y \in X_2}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_2} \mathbf{E}_{cx}} = \sum_{\substack{y=c, \\ y \in X_2}} \frac{\mathbf{E}_{cy}}{\sum_{x \in X_2} \mathbf{E}_{cx}}.
\end{aligned} \tag{31}
$$

Accordingly, for $x = c$, Eq. (30) can be rewritten as follows:

$$
\frac{\theta}{|X_1|} = \frac{\theta}{|X_2|}. \tag{32}
$$

This equation does not hold as it contradicts $|X_1| \neq |X_2|$. Combining the conducted analysis of all four cases, the assumption shown in Eq. (24) is FALSE. In other words, there does NOT exist a pair of graph structures that the PMP-GAT layer cannot distinguish. Therefore, any layer in a PMP-GAT can pass the 1-WL test. □

4.2.4. The equivalence between the PMP-GPN layer and the 1-WL test

Proof. Like the proof for PMP-GCN and PMP-GAT, we can prove that any layer in a PMP-GAT can pass the 1-WL test via *proof by contradiction*. First, assume that Eq. (14) for PMP-GPN is FALSE, i.e., there exists a pair of graphs that the PMP-GPN layer cannot distinguish. This assumption can be written as follows:

$$
\begin{aligned}
&\mathbf{E}_{c_i x} = \exp\{a_{c_i x} + b_{c_i x} - \beta(k_{c_i x} + l_{c_i x})\}, \\
&w_{c_i x} = \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x \in X_i} \mathbf{E}_{c_i x} \cdot \sum_{z \in X_y} \mathbf{E}_{yz}}}, \\
&h(c_1, X_1) = h(c_2, X_2), |X_1| \neq |X_2|,
\end{aligned} \tag{33}
$$

where $h(,)$ is defined as Eq. (9) shows. We know that the representations for c_1 and c_2 generated by a PMP-GPN layer can be analyzed in four different cases, including $M_1 \neq M_2$ and $c_1 \neq c_2$, $M_1 \neq M_2$ but $c_1 = c_2$, $M_1 = M_2$ but $c_1 \neq c_2$, and $M_1 = M_2$ and $c_1 = c_2$. We will then demonstrate that $h(c_1, X_1) = h(c_2, X_2)$ can not hold as possible contradictions are identified under all these four cases.

$M_1 \neq M_2$ and $c_1 \neq c_2$ or $M_1 \neq M_2$ but $c_1 = c_2$ Let us first analyze the representations learned by a PMP-GPN layer when $M_1 \neq M_2$. If $h(c_1, X_1) = h(c_2, X_2)$, we have:

$$\begin{aligned} h(c_1, X_1) &= \frac{\theta}{|X_1|} c_1 + (1-\alpha) \sum_{x \in M_1} \sum_{\substack{y=x, \\ y \in X_1}} w_{c_1 y} y + \alpha c_1^{in} \\ &= \frac{\theta}{|X_2|} c_2 + (1-\alpha) \sum_{x \in M_2} \sum_{\substack{y=x, \\ y \in X_2}} w_{c_2 y} y + \alpha c_2^{in} = h(c_2, X_2), \end{aligned} \tag{34}$$

where c_1^{in} and c_2^{in} represent the features of the two central nodes before the message-passing of the first PMP-GPN layer. As each $w_{c_i x}$ is positive, each distinct element in either M_1 or M_2 will contribute to the representations of c_1 and c_2. We deduce $M_1 = M_2$ if $h(c_1, X_1) = h(c_2, X_2)$, which contradicts $M_1 \neq M_2$. Thus, $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$ does not hold when $M_1 \neq M_2$.

$M_1 = M_2$ but $c_1 \neq c_2$ Next, we analyze the representations learned by a PMP-GPN layer when $M_1 = M_2$ but $c_1 \neq c_2$. Accordingly, we assume $M_1 = M_2 = M$, $c_1 \neq c_2$, and $h(c_1, X_1) = h(c_2, X_2)$. According to Eq. (33), we have the following:

$$\begin{aligned} w_{c_i x} &= \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x \in X_i} \mathbf{E}_{c_i x} \cdot \sum_{z \in X_y} \mathbf{E}_{yz}}}, \\ h(c_1, X_1) &= \frac{\theta}{|X_1|} c_1 + (1-\alpha) \sum_{x \in M} \sum_{\substack{y=x, \\ y \in X_1}} w_{c_1 y} y + \alpha c_1^{in}, \\ h(c_2, X_2) &= \frac{\theta}{|X_2|} c_2 + (1-\alpha) \sum_{x \in M} \sum_{\substack{y=x, \\ y \in X_2}} w_{c_2 y} y + \alpha c_2^{in}, \end{aligned} \tag{35}$$

where c_1^{in} and c_2^{in} represent the features of the two central nodes before the message-passing of the first PMP-GPN layer. Given $M_1 = M_2 = M$, the message weights of each $x \in M$ composed of the representations of c_1 and c_2 should be identical if $h(c_1, X_1) = h(c_2, X_2)$. Denote the underlying sets for X_1 and X_2 as $M = \{c_1, c_2, \ldots\}$, and central node features as c_1 and c_2. For $x = c_1$ at the first PMP-GPN layer, we have the following:

$$\begin{aligned} w_{c_i x} &= \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x \in X_i} \mathbf{E}_{c_i x} \cdot \sum_{z \in X_y} \mathbf{E}_{yz}}}, \\ \frac{\theta}{|X_1|} c_1 &+ (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} y + \alpha c_1 = (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y} y. \end{aligned} \tag{36}$$

If $(1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} + \alpha = (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y}$, we have:

$$\frac{\theta}{|X_1|} = 0, \tag{37}$$

which contradicts $\frac{\theta}{|X_1|} > 0$. If $(1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} + \alpha \neq (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y}$, we have:

$$|X_1| = \frac{\theta}{(1-\alpha)[\sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y} - \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y}] - \alpha}. \tag{38}$$

The RHS of the equation above can be an irrational number, which contradicts the LHS as a positive integer. For $x = c_1$ at a higher PMP-GPN layer, we have the following:

$$\begin{aligned} w_{c_i x} &= \phi(\mathbf{E}_{c_i x}) = \frac{\mathbf{E}_{c_i x}}{\sqrt{\sum_{x \in X_i} \mathbf{E}_{c_i x} \cdot \sum_{z \in X_y} \mathbf{E}_{yz}}}, \\ \frac{\theta}{|X_1|} c_1 &+ (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} y + \alpha c_1^{in} = (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y} y. \end{aligned} \tag{39}$$

The equation above can be further rewritten as:

$$\frac{\theta}{|X_1|} + (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_1}} w_{c_1 y} y + \alpha \frac{c_1^{in}}{c_1} = (1-\alpha) \sum_{\substack{y=c_1, \\ y \in X_2}} w_{c_2 y} y. \tag{40}$$

$\frac{c_1^{in}}{c_1}$ is a non-zero number as c_1 and c_1^{in} are countable features. If $(1-\alpha)\sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y} + \alpha\frac{c_1^{in}}{c_1} = (1-\alpha)\sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y}$, we have $\frac{\theta}{|X_1|} = 0$, which contradicts $\frac{\theta}{|X_1|} > 0$. If $(1-\alpha)\sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y} + \alpha\frac{c_1^{in}}{c_1} \neq (1-\alpha)\sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y}$, we have:

$$|X_1| = \frac{\theta}{(1-\alpha)[\sum_{\substack{y=c_1,\\ y\in X_2}} w_{c_2 y} - \sum_{\substack{y=c_1,\\ y\in X_1}} w_{c_1 y}] - \alpha\frac{c_1^{in}}{c_1}}. \tag{41}$$

The RHS of the equation above can be an irrational number, which contradicts the LHS as a positive integer. Thus, $h(c_1, X_1) = h(c_2, X_2)$, $|X_1| \neq |X_2|$ does not hold when $M_1 = M_2$ but $c_1 \neq c_2$.

$M_1 = M_2$ and $c_1 = c_2$ At last, we analyze the representations learned by the PMP-GPN layer when $M_1 = M_2$ and $c_1 = c_2$. If $h(c_1, X_1) = h(c_2, X_2)$, the message weights of each $x \in M$ that are used to generate the representations of c_1 and c_2 should be the same. For an x in M, say $x = c$, we have the following:

$$\begin{aligned} w_{cx} &= \phi(\mathbf{E}_{cx}) = \frac{\mathbf{E}_{cx}}{\sqrt{\sum_{x\in X_i}\mathbf{E}_{cx}\cdot\sum_{z\in X_y}\mathbf{E}_{yz}}}, \\ \frac{\theta}{|X_1|}c + (1-\alpha)\sum_{\substack{y=c,\\ y\in X_1}} w_{cy}y + \alpha c^{in} &= \frac{\theta}{|X_2|}c + (1-\alpha)\sum_{\substack{y=c,\\ y\in X_2}} w_{cy}y + \alpha c^{in}. \end{aligned} \tag{42}$$

Accordingly, Eq. (42) can be rewritten as follows:

$$\frac{1}{|X_2|} - \frac{1}{|X_1|} = \frac{(1-\alpha)}{\theta}[\sum_{\substack{y=c,\\ y\in X_1}} w_{cy} - \sum_{\substack{y=c,\\ y\in X_2}} w_{cy}]. \tag{43}$$

The LHS of Eq. (43) is a non-zero rational number, which contradicts that the RHS of Eq. (43) can be zero, or an irrational number. Combining the conducted analysis of all four cases, the assumption shown in Eq. (33) is FALSE. In other words, there does NOT exist a pair of graph structures that the PMP-GPN layer cannot distinguish. Therefore, any layer in a PMP-GPN can pass the 1-WL test. □

Combining the theoretical analysis conducted in Sections 4.2.2-4.2.4, Theorem 2 is proved. Previous studies have demonstrated that most conventional MPGNNs cannot pass the 1-WL test [16,17,20]. Therefore, the theoretical results shown in Theorems 1 and 2 establish the constantly higher expressiveness of the proposed PMP-GNNs compared with conventional MPGNNs, additionally providing a theoretical guarantee of the robust performance of PMP-GNNs in diverse learning tasks.

5. Related work

In this section, previous works that are closely related to the proposed PMP-GNNs are briefly reviewed.

5.1. Graph neural networks

GNNs [8,17,21] are recognized as powerful tools for analyzing graph-structured data. They have been successfully adapted for diverse real-world applications, including scientific document classification [22], coauthor analysis [5], conversation generation [23], social community detection [20], and text classification [24]. Typically, the low-dimensional representations tailored for downstream tasks are learned using a GNN through multi-layer node feature aggregation and projection. Consequently, the aggregation of neighbor features for each central node is crucial for GNNs to learn meaningful representations.

5.2. Message-passing graph neural networks

Most present GNNs can be conceptualized as MPGNNs [25,26]. In each layer of an MPGNN, the output representation is learned through a two-step approach. Messages from the neighbors of the central node are first composed through the multiplication of neighbor features with weights that indicate their importance. Then the messages are conveyed to each central node. The output representations are generated by updating the features of the central node and the received messages, through aggregation or max or min operations.

The strategy of composing messages, that is, the calculation of weights indicating the importance of neighbor's features, plays a central role in MPGNNs. Various message-passing paradigms for GNNs emerge from different selections of weight calculation strategies. Several approaches for computing message weights have been proposed. For instance, some GNNs leverage an attention mechanism [3,20,27] that quantifies the feature correlation or similarity between pairwise nodes to formulate messages. Correlations induced by the graph structure, such as the graph Laplacian [1,16,28] and graph diffusion [12,29], can also be considered measures of importance, which are subsequently used to convey messages to central nodes in the graph. To improve the efficiency of MPGNNs, sampling strategies [8,21,30] can be implemented to select a subset of similar neighbors, enabling the transmission of messages from the neighbors to the central node for representation learning.

Existing MPGNNs concentrate on learning representations from messages derived from neighbor similarity or correlation. Messages from the opposite side, that is, the side indicating the dissimilarity between pairwise nodes in the graph, are unexplored. Dissimilarity among data has been investigated in other domains, such as multi-view and manifold learning [31,32]. However, these approaches cannot be readily adapted to existing MPGNNs. Therefore, we propose PMP to enable GNNs to simultaneously leverage messages related to both similarity and dissimilarity to learn more expressive representations suitable for various downstream tasks.

6. Experimental setup

In this section, we introduce the setup of our experiments to validate the proposed PMP-GCN, PMP-GAT, and PMP-GPN to reveal the effectiveness of the PMP paradigm.

6.1. Comparison baselines

To validate the effectiveness of the proposed PMP-GCN, PMP-GAT, and PMP-GPN, we selected twelve state-of-the-art GNNs as comparison baselines: GCN [1], MoNet [33], GraphSAGE [8], ARMA [28], GAT [3], GATv2 [34], HardGAT [30], MAGNA [4], JKNet [29], APPNP [5], SGC [10], and GPR [13]. Specifically, GCN, MoNet, GraphSAGE, and ARMA are four strong GNNs that are built upon various graph convolutional layers. GAT, GATv2, HardGAT, and MAGNA are four representative attention-based GNNs. JKNet, APPNP, SGC, and GPR are four strong GNNs whose layers are constructed according to PageRank (graph diffusion). Comparing PMP-GCN, PMP-GAT, and PMP-GPN with baselines whose message for representation learning is solely based on neighbor correlation or similarity can demonstrate the effectiveness of both PMP-GNNs.

6.2. Datasets and real-world analytical tasks

Twelve real-world datasets are selected to experimentally test the effectiveness of all considered approaches. The datasets are Cora, Cite, PubMed [35], Blog, Flickr [36], CoauthorCS, CoauthorPH [37], Sport, BBC, IrishTimes, Guardian, and Wikipedia [38]. The details of the test datasets are presented below.

Cora, cite, PubMed Cora, Cite, and PubMed are three classical graph datasets representing scientific article citations. The nodes, edges, and node features represent the scientific articles, article-article citations, and article keywords, respectively. Cora consists of 2,708 nodes, 5,429 edges, and 1,433 features. The Cite dataset consists of 3,327 nodes, 4,732 edges, and 3,703 features. PubMed consists of 19,717 nodes, 44,338 edges, and 500 features. All articles (nodes) in these three datasets are assigned class labels representing the research domains to which they belong. For Cora, Cite, and PubMed, there are seven, six, and three ground-truth classes, respectively.

Blog Blog is a social graph dataset extracted from blogcatalog.com. It consists of 5,196 nodes, 171,743 edges, and 8,189 features, representing blog sites, blog-blog hyperlinks, and contextual descriptions, respectively. Each blog site in this dataset is assigned one of the six class labels representing blog categories.

Flickr Flickr is a graph dataset collected from flickr.com. It consists of 7,575 nodes, 239,738 edges, and 12,047 features. The nodes, edges, and features in this dataset represent images, image-image links, and the content descriptions of images. Each of the 7,575 images is assigned to one of nine ground-truth classes.

CoauthorCS and CoauthorPH CoauthorCS and CoauthorPH are collaboration graphs extracted from the computer science and physics societies of arxiv.com. In both datasets, each node represents a researcher, an edge between nodes indicates that the researchers have coauthored a paper published on Arxiv, and features represent the research topics on which the researchers worked. CoauthorCS consists of 18,333 nodes divided into 15 research domains of computer science, 327,576 edges, and 6,805 features. CoauthorPH consists of 34,493 nodes assigned to five research domains of physics, 991,848 edges, and 8,415 features.

Sport, BBC, IrishTimes, Guardian, and Wikipedia Unlike the abovementioned graph datasets, the Sport, BBC, IrishTimes, Guardian, and Wikipedia datasets are collected from rich-text media, namely BBC Sports news, BBC news, news from The Times (Ireland), news from The Guardian, and Wikipedia content. In these five datasets, nodes, features, and edges represent pieces of news or wiki pages, textual contents, and news-news or page-page similarities, respectively. The Sport dataset consists of 737 nodes belonging to five news classes, 17,242 edges, and 966 features. The BBC dataset consists of 2,225 nodes belonging to five disjoint news classes, 9,127 edges, and 3,121 features. IrishTimes consists of 3,246 nodes belonging to seven news classes, 490,325 edges, and 4,823 features. Guardian consists of 6,521 nodes belonging to six news classes, 96,279 edges, and 10,801 features. The Wikipedia dataset consists of 5,739 nodes belonging to six classes, 524,692 edges, and 17,311 features.

As these test datasets are collected from five distinct real-world scenarios, five diverse analytical tasks, namely scientific article classification (Cora, Cite, and PubMed), blog clustering (Blog), text-based image clustering (Flickr), coauthor analysis (CoauthorCS and CoauthorPH), and rich-text classification (Sport, BBC, IrishTimes, Guardian, and Wikipedia), are considered in the experiment to test the effectiveness of all approaches. Through these diverse downstream tasks, the effectiveness of different GNNs can be comprehensively validated. The characteristics of the 12 abovementioned datasets are summarized in Table 1. Besides, the accumulative

Table 1
Statistics of the testing datasets.

	Cora	Cite	PubMed	Blog	Flickr	CoauthorCS	CoauthorPH	Sport	BBC	IrishTimes	Guardian	Wikipedia
N	2708	3327	19717	5196	7575	18333	34493	737	2225	3246	6521	5739
$\|E\|$	5429	4732	44338	171743	239738	327576	991848	17242	9127	490325	96279	524692
D	1433	3703	500	8189	12047	6805	8415	966	3121	4823	10801	17311
C	7	6	3	6	9	15	5	5	5	7	6	6
Type	Scientific article			Blog	Text-based image	Collaboration		Rich text				

distribution of normalized feature and structure distances among neighbors on all test datasets is plotted in Fig. A.3. It is observed that most neighbors on all test datasets are very distant, which is consistent with previous studies showing real-world graphs are created differently [6,7]. Conducting representation learning tasks in such real-world graphs allows the proposed PMP-GNNs to perform robustly in diverse downstream tasks.

6.3. Experimental settings

Configurations of comparison baselines To conduct fair comparisons, all GNNs generally employ a two-layer network structure to perform all downstream tasks mentioned in Section 6.2. In other words, each GNN in the experiment contains one hidden layer, followed by the output layer. The official source codes of all baseline GNNs are used in the experiment and configured using the recommended settings. Specifically, the hidden layer dimension and learning rate of GCN, MoNet, GraphSAGE, and ARMA are set to 32 and 0.01, respectively. The dropout rate of GCN, MoNet, and GraphSAGE is set to 0.5, while that of ARMA is 0.75. For attention-based GNN baselines, including GAT, GATv2, and MAGNA, the learning rate is set to 0.005 in all analytical tasks, while that of HardGAT is set to 0.01. The dropout rates of GAT, GATv2, and HardGAT are set to 0.6, while that of MAGNA is set to 0.5 in all analytical tasks. Eight attention heads are adopted by the four attention-based GNN baselines to construct the hidden layer, the dimension of each head adopted by GAT, GATv2, and HardGAT is set to 8 in the scientific article classification task, and 32 in the remaining tasks. The hidden dimension of each head adopted by MAGNA is set to 32 and 512 in the scientific article classification task and the remaining tasks. One attention head is adopted by the four attention-based GNNs to build the output layer. For JKNet, APPNP, and GPR, the dropout rate is set to 0.5, while that of SGC is set to 0 as recommended. The hidden layer dimension is set to 32 for JKNet, 64 for APPNP and GPR, and the class number of each dataset for SGC. The learning rate of JKNet and SGC is set to 0.005, and that of APPNP and GPR is set to 0.01.

Configurations of PMP-GCN, PMP-GAT and PMP-GPN The settings of PMP-GCN, PMP-GAT, and PMP-GPN are similar to those of GNNs based on convolutional, attention, and PageRank layers. Specifically, the hidden layer dimension and dropout rate are 32 and 0.75 for PMP-GCN and 0.6 and 64 for PMP-GAT, respectively. The hidden layer dimension and dropout rate are 64 and 0.5 for PMP-GPN, respectively. The learning rates are set to 0.01 for PMP-GCN and PMP-GPN and 0.005 for PMP-GAT in all downstream tasks. For PMP-GAT, we only use one attention head to construct the hidden layer. All GNNs are executed on a workstation installed with an NVIDIA RTX3090 graphics card, Python 3.8.5, and CUDA 11.1. All GNNs are trained using the Adam optimizer [39] and run 10 times in each learning task to obtain the average performance with the standard deviation. To evaluate the performance of all GNNs, we use $Accuracy$ (Acc), a widely accepted metric in various real-world learning tasks.

Data split in all learning tasks The data split for training, validation, and testing significantly differs among the five considered learning tasks. Specifically, for scientific article classification, established data splits [1,3,40] are adopted. For each test dataset, 20 nodes with labels from each ground-truth class are used for model training. Thus, Cora, Cite, and PubMed consist of 140, 120, and 60 nodes for model training. For these three datasets, 500 and 1,000 nodes are used for validation and testing, respectively. For the blog clustering, text-based image clustering, and coauthor analysis tasks, five sets of training and validation splits are randomly generated, as there are no established data splits. For each dataset, the size of the training, validation, and test splits is equal to the number of ground-truth classes multiplied by 20, 500, and all nodes, respectively. Consequently, 120, 180, 300, and 100 nodes are sampled for training on Blog, Flickr, CoauthorCS, and CoauthorPH, respectively. For the rich-text classification task on each test dataset, 60%, 20%, and 20% of nodes are sampled as training, validation, and testing splits, respectively, and five sets of splits are generated to test the performance of all GNNs.

7. Results and analysis

In this section, the predictive performances of the proposed PMP-GCN, PMP-GAT, and PMP-GPN are compared with those of the well-established GNN baselines mentioned in Section 6 on five learning tasks based on 12 real-world datasets. In addition, the significant difference between the proposed PMP-GNNs and other MPGNNs is discussed according to the corresponding results.

Table 2
Comparison of average performance in scientific article classification between convolutional GNNs and PMP-GCN. The best performance on each dataset is highlighted in bold.

	GCN	MoNet	GraphSAGE	ARMA	PMP-GCN
Cora	0.814 ± 0.019	0.820 ± 0.005	0.819 ± 0.001	0.791 ± 0.009	**0.841 ± 0.002**
Cite	0.716 ± 0.007	0.642 ± 0.002	0.704 ± 0.012	0.700 ± 0.016	**0.727 ± 0.003**
PubMed	0.797 ± 0.004	0.798 ± 0.003	0.788 ± 0.009	0.801 ± 0.008	**0.823 ± 0.002**

Table 3
Comparison of average performance in scientific article classification between attention-based GNNs and PMP-GAT. The best performance on each dataset is highlighted in bold.

	GAT	GATv2	MAGNA	HardGAT	PMP-GAT
Cora	0.835 ± 0.003	0.841 ± 0.008	0.829 ± 0.002	0.794 ± 0.005	**0.850 ± 0.003**
Cite	0.708 ± 0.003	0.714 ± 0.005	0.672 ± 0.008	0.719 ± 0.005	**0.721 ± 0.004**
PubMed	0.815 ± 0.005	0.819 ± 0.003	0.813 ± 0.007	0.810 ± 0.003	**0.833 ± 0.004**

Table 4
Comparison of average performance in scientific article classification between PageRank GNNs and PMP-GPN. The best performance on each dataset is highlighted in bold.

	SGC	GPR	JKNet	APPNP	PMP-GPN
Cora	0.814 ± 0.001	0.818 ± 0.005	0.794 ± 0.009	0.825 ± 0.008	**0.846 ± 0.003**
Cite	0.713 ± 0.000	0.720 ± 0.006	0.651 ± 0.008	0.720 ± 0.005	**0.723 ± 0.002**
PubMed	0.773 ± 0.000	0.827 ± 0.004	0.807 ± 0.006	0.821 ± 0.005	**0.836 ± 0.002**

Table 5
Comparison of average performance in blog and text-based image clustering, and coauthor analysis between convolutional GNNs and PMP-GCN. The best performance in each dataset is highlighted in bold.

	GCN	MoNet	GraphSAGE	ARMA	PMP-GCN
Blog	0.654 ± 0.005	0.713 ± 0.012	0.572 ± 0.007	0.625 ± 0.007	**0.792 ± 0.007**
Flickr	0.471 ± 0.009	0.524 ± 0.007	0.370 ± 0.005	0.408 ± 0.003	**0.546 ± 0.009**
CoauthorCS	0.887 ± 0.007	0.875 ± 0.009	0.899 ± 0.009	0.824 ± 0.003	**0.905 ± 0.005**
CoauthorPH	0.923 ± 0.003	0.918 ± 0.003	**0.936 ± 0.002**	0.917 ± 0.007	0.927 ± 0.007

7.1. Classification performance on scientific articles

Scientific article classification is a widely accepted testbed for evaluating GNN performance. Three classic datasets, Cora, Cite, and PubMed, are used in our experiment to reveal the effectiveness of different GNNs. The comparisons regarding classification performances are summarized in Tables 2, 3, and 4. The proposed PMP-GCN outperforms all GNN baselines built with various convolutional layers (Table 2). Specifically, PMP-GCN achieves a performance improvement of 2.56% on Cora, 1.53% on Cite, and 2.75% on PubMed. The proposed PMP-GAT also outperforms all baselines of attention-based GNNs. PMP-GAT achieves a performance gain of 1.07%, 0.28%, and 1.71% on Cora, Cite, and PubMed, respectively (Table 3). Compared with PageRank GNNs, the GNN based on the proposed PMP paradigm still performs robustly. The proposed PMP-GPN achieves a performance gain of 2.55%, 0.42%, and 1.09% on Cora, Cite, and PubMed, respectively (Table 4). The experimental results presented in Tables 2, 3, and 4 demonstrate that the proposed PMP paradigm is very effective for scientific article classification. GNNs based on the proposed message-passing paradigm can learn more expressive representations from strongly correlated articles.

7.2. Clustering performance on blogs and text-based images

Blog clustering and text-based image clustering are two challenging tasks, as all nodes in the datasets are classified with very limited supervision. In our experiment, Blog and Flickr are used to test the effectiveness of all GNNs in these two challenging tasks. The corresponding results are summarized in Tables 5, 6 and 7. The proposed PMP-based GNNs still perform robustly compared with the various types of GNN baselines. PMP-GCN outperforms MoNet in clustering Acc on Blog and Flickr datasets by 11.08% and 4.20%, respectively. PMP-GAT outperforms attention-based GNNs by 32.84% on the Blog dataset and 11.32% on the Flickr dataset. PMP-GPN outperforms PageRank GNNs by 13.73% and 10.99% on Blog and Flickr datasets, respectively. Compared with the datasets of scientific articles, Blog and Flickr are denser, potentially resulting in a larger number of highly dissimilar nodes that are connected. The proposed PMP paradigm can significantly improve the learning performance of GNNs on these dense datasets by considerably reducing the importance of messages from dissimilar neighbors during the generation of output representations for each node. This explains the significant performance gains of PMP-GCN, PMP-GAT, and PMP-GPN compared with other MPGNNs.

Table 6
Comparison of average performance in blog and text-based image clustering, and coauthor analysis between attention-based GNNs and PMP-GAT. The best performance in each dataset is highlighted in bold.

	GAT	GATv2	MAGNA	HardGAT	PMP-GAT
Blog	0.489 ± 0.004	0.606 ± 0.007	0.570 ± 0.006	0.447 ± 0.006	**0.805 ± 0.007**
Flickr	0.468 ± 0.005	0.357 ± 0.004	0.365 ± 0.007	0.420 ± 0.003	**0.521 ± 0.005**
CoauthorCS	0.880 ± 0.004	0.884 ± 0.004	0.857 ± 0.006	0.890 ± 0.005	**0.905 ± 0.002**
CoauthorPH	0.922 ± 0.003	0.921 ± 0.004	0.908 ± 0.004	0.927 ± 0.002	**0.939 ± 0.002**

Table 7
Comparison of average performance in blog and text-based image clustering, and coauthor analysis between PageRank GNNs and PMP-GPN. The best performance in each dataset is highlighted in bold.

	SGC	GPR	JKNet	APPNP	PMP-GPN
Blog	0.501 ± 0.007	0.675 ± 0.004	0.714 ± 0.007	0.686 ± 0.009	**0.812 ± 0.006**
Flickr	0.352 ± 0.001	0.441 ± 0.006	0.430 ± 0.003	0.473 ± 0.008	**0.525 ± 0.007**
CoauthorCS	0.885 ± 0.003	0.854 ± 0.001	0.879 ± 0.005	0.887 ± 0.010	**0.903 ± 0.002**
CoauthorPH	0.919 ± 0.001	0.927 ± 0.001	0.932 ± 0.002	0.922 ± 0.006	**0.934 ± 0.001**

Table 8
Comparison of average performance in rich-text classification between convolutional GNNs and PMP-GCN. The best performance in each dataset is highlighted in bold.

	GCN	MoNet	GraphSAGE	ARMA	PMP-GCN
Sport	0.966 ± 0.006	0.961 ± 0.006	0.955 ± 0.005	0.972 ± 0.007	**0.991 ± 0.003**
BBC	0.956 ± 0.004	0.964 ± 0.003	0.973 ± 0.003	0.938 ± 0.010	**0.981 ± 0.002**
IrishTimes	0.898 ± 0.004	0.684 ± 0.005	0.927 ± 0.004	0.868 ± 0.005	**0.937 ± 0.002**
Guardian	0.951 ± 0.002	0.795 ± 0.005	0.936 ± 0.004	0.934 ± 0.004	**0.959 ± 0.001**
Wikipedia	0.883 ± 0.003	0.659 ± 0.008	0.880 ± 0.008	0.882 ± 0.002	**0.904 ± 0.005**

7.3. Analytical performance in coauthor graphs

Coauthor analysis is one of the major tasks performed by modern GNNs. It usually involves leveraging limited supervision to correctly categorize authors who frequently collaborate or work within correlated research domains. In our experiment, we use two large collaborative graphs, CoauthorCS and CoauthorPH, to test the performance of different GNNs. The comparative results are summarized in Tables 5, 6, and 7. The proposed PMP-GCN, PMP-GAT, and PMP-GPN demonstrate a robust analytical performance compared with the other state-of-the-art GNNs. PMP-GCN outperforms the best convolutional GNN baseline (GraphSAGE) by 0.67% on the CoauthorCS dataset and performs second best on the CoauthorPH dataset (Table 5). After investigating the data characteristics and comparing the message-passing paradigms of PMP-GCN and GraphSAGE, we may deduce the reasons leading PMP-GCN not to perform the best on the CoauthorPH dataset. It is observed from Fig. A.3 that the portion of very large distances (i.e., very dissimilar neighbors pertaining to structure or feature) on the CoauthorPH dataset is relatively lower compared with other test datasets. Therefore, a GNN is expected to perform better if it leverages more information from neighbors to learn representations. As the proposed PMP-GCN pursues learning representations from sparse and strongly correlated neighbors, under the experimental settings for all test datasets, PMP-GCN still leans toward learning zero or very-close-to-zero message weights for a large number of neighbors on the CoauthorPH dataset (See Figs. A.5). As a result, PMP-GCN might not involve a small portion of neighbors that might be meaningful to generate representations. In contrast, the message weights adopted by GraphSAGE are defined as $\frac{1}{d_i}$, where d_i is the degree of central node i. GraphSAGE performs slightly better than the proposed PMP-GCN, possibly because GraphSAGE leverages more neighbors to learn representations on the CoauthorPH dataset.

In contrast to PMP-GCN, the other two PMP-GNNs (PMP-GAT and PMP-GPN) exhibit the best performance on both datasets. As shown in Table 6, PMP-GAT outperforms HardGAT by 1.69% and 1.29% on CoauthorCS and CoauthorPH. PMP-GPN outperforms APPNP by 1.80% on CoauthorCS, and is better than JKNet by 0.21% on CoauthorPH (Table 7). The results demonstrate that leveraging polarized messages to learn representations for coauthor analysis tasks allows GNNs to reduce the influence of dissimilar coauthors. The subsequent representations are learned by concentrating on the strongly correlated coauthors.

7.4. Performance in rich-text classification

Rich-text classification is a fundamental task in natural language processing. It involves the accurate categorization of rich-content documents, such as news and wiki pages. In our experiment, we select five representative text datasets, namely Sport, BBC, IrishTimes, Guardian, and Wikipedia, to verify the effectiveness of different GNNs. The corresponding results are summarized in Tables 8, 9, and 10. The proposed PMP-GCN, PMP-GAT, and PMP-GPN outperform various types of MPGNNs in the task of rich-text classification. PMP-GCN outperforms convolution-based GNNs on all datasets. For example, it outperforms ARMA by 1.95% on the Sport dataset. PMP-GCN outperforms GraphSAGE by 0.82% on BBC, and 1.08% on IrishTimes. PMP-GCN outperforms GCN on Guardian and Wikipedia datasets by 0.84% and 2.38%, respectively. The proposed PMP-GAT exhibits a robust performance

Table 9
Comparison of average performance in rich-text classification between attention-based GNNs and PMP-GAT. The best performance in each dataset is highlighted in bold.

	GAT	GATv2	MAGNA	HardGAT	PMP-GAT
Sport	0.965 ± 0.008	0.964 ± 0.007	0.954 ± 0.006	0.923 ± 0.001	**0.984 ± 0.003**
BBC	0.963 ± 0.003	0.962 ± 0.003	0.959 ± 0.006	0.965 ± 0.008	**0.977 ± 0.001**
IrishTimes	0.880 ± 0.004	0.898 ± 0.004	0.885 ± 0.005	0.780 ± 0.003	**0.925 ± 0.004**
Guardian	0.940 ± 0.003	0.929 ± 0.003	0.925 ± 0.002	0.923 ± 0.007	**0.949 ± 0.002**
Wikipedia	0.876 ± 0.002	0.874 ± 0.004	0.869 ± 0.008	0.844 ± 0.002	**0.889 ± 0.002**

Table 10
Comparison of average performance in rich-text classification between PageRank GNNs and PMP-GPN. The best performance in each dataset is highlighted in bold.

	SGC	GPR	JKNet	APPNP	PMP-GPN
Sport	0.973 ± 0.000	0.936 ± 0.003	0.955 ± 0.009	0.969 ± 0.009	**0.993 ± 0.001**
BBC	0.962 ± 0.000	0.967 ± 0.003	0.944 ± 0.008	0.963 ± 0.004	**0.974 ± 0.002**
IrishTimes	0.748 ± 0.000	0.729 ± 0.002	0.825 ± 0.007	0.872 ± 0.006	**0.932 ± 0.005**
Guardian	0.920 ± 0.000	0.926 ± 0.006	0.881 ± 0.003	0.930 ± 0.009	**0.948 ± 0.001**
Wikipedia	0.814 ± 0.000	0.869 ± 0.005	0.729 ± 0.006	0.858 ± 0.006	**0.905 ± 0.004**

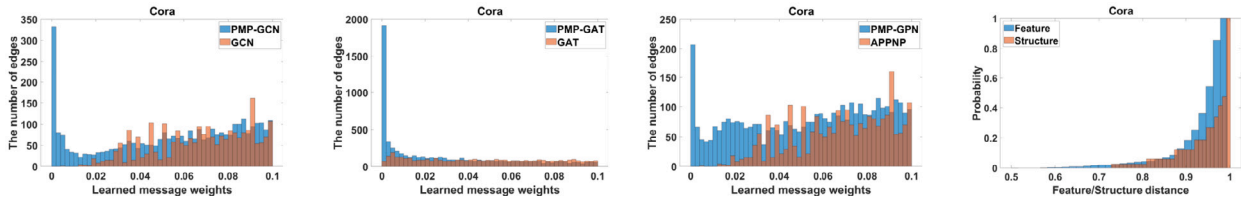
Table 11
Performance comparisons between PMP-based GNNs and their variants.

Graph Convolution												
	Cora	Cite	PubMed	Blog	Flickr	CoauthorCS	CoauthorPH	Sport	BBC	IrishTimes	Guardian	Wikipedia
GCN	0.814	0.716	0.797	0.654	0.471	0.887	0.923	0.966	0.956	0.898	0.951	0.883
PMP-GCN/d	0.838	0.718	0.809	0.767	0.530	0.899	0.922	0.963	0.966	0.902	0.952	0.903
PMP-GCN	0.841	0.727	0.823	0.792	0.546	0.905	0.927	0.991	0.981	0.937	0.959	0.904
Graph Attention												
	Cora	Cite	PubMed	Blog	Flickr	CoauthorCS	CoauthorPH	Sport	BBC	IrishTimes	Guardian	Wikipedia
GAT	0.835	0.708	0.815	0.489	0.468	0.880	0.922	0.965	0.963	0.880	0.940	0.876
PMP-GAT/d	0.836	0.709	0.818	0.802	0.497	0.893	0.915	0.979	0.962	0.921	0.942	0.885
PMP-GAT	0.850	0.721	0.833	0.805	0.521	0.905	0.939	0.984	0.977	0.925	0.949	0.889
Graph PageRank												
	Cora	Cite	PubMed	Blog	Flickr	CoauthorCS	CoauthorPH	Sport	BBC	IrishTimes	Guardian	Wikipedia
APPNP	0.825	0.720	0.821	0.686	0.473	0.887	0.922	0.969	0.963	0.872	0.930	0.858
PMP-GPN/d	0.835	0.715	0.825	0.674	0.474	0.903	0.925	0.987	0.964	0.922	0.931	0.899
PMP-GPN	0.846	0.723	0.836	0.812	0.525	0.908	0.934	0.993	0.974	0.932	0.948	0.905

compared with attention-based GNNs. PMP-GAT outperforms GAT by 1.97%, 0.96%, and 1.48% on Sport, Guardian, and Wikipedia, respectively. PMP-GAT outperforms HardGAT by 1.24% on BBC, and GATv2 by 3.01% on IrishTimes. PMP-GPN also performs the best when compared with PageRank GNNs. PMP-GPN outperforms SGC by 2.06% on Sport. PMP-GPN outperforms APPNP by 6.88% and 1.94% on IrishTimes and Guardian, respectively. PMP-GPN outperforms GPR by 0.72% and 4.14% on BBC and Wikipedia, respectively. In contrast to the existing GNN learning representations that solely rely on messages related to correlation, PMP-GCN, PMP-GAT, and PMP-GPN can identify dissimilar documents. PMP-GNNs can thus utilize messages from truly correlated neighbors to learn more expressive representations.

7.5. Comparisons of similarity-, dissimilarity-, and PMP-based GNNs

As the proposed PMP-GNNs leverage similarity and dissimilarity between neighbors to learn representations, comparing them with GNN variants considering either similarity or dissimilarity can further demonstrate the effectiveness of the proposed methods. Specifically, we let the learnable matrix $\mathbf{C} = \mathbf{0}$, leading the proposed PMP-based GNNs to learn representations only considering neighbor dissimilarity (PMP-GCN/d, PMP-GAT/d, and PMP-GPN/d). Taking conventional GCN, GAT, and APPNP as GNN variants only considering neighbor similarity for representation learning, PMP-GCN/d, PMP-GAT/d, and PMP-GPN/d as GNN variants only concerning neighbor dissimilarity, we compare their predictive performances with the proposed PMP-based GNNs on all learning tasks. The corresponding results have been summarized in Table 11. As the table shows, PMP-GCN/d, PMP-GAT/d, and PMP-GPN/d still can obtain competitive performance on most testing datasets. Considering that a very large portion of neighbors are shown to be very distant in Fig. A.3, we can conclude that solely leveraging dissimilarity widely existing in real-world graph data to learn message weights is an auspicious way to guide message-passing GNNs to learn expressive representations. Nevertheless, the best performances obtained by PMP-based GNNs demonstrate that considering both similarity and dissimilarity can lead message-passing GNNs to learn more expressive representations compared with those considering similarity or dissimilarity only.



(a) Message weights (b) Distances

Fig. 2. Message weights learned by PMP-GNNs and conventional message-passing GNNs, and feature/structure distances on Cora dataset. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

7.6. Comparisons of learned message weights

As previously mentioned, the proposed PMP paradigm enables GNNs to learn to reduce and negate messages transmitted by dissimilar neighbors so as to preserve messages from strongly correlated neighbors. To demonstrate this advantageous capability, we conduct a comparative analysis of the small message weights learned by PMP-GCN, PMP-GAT, and PMP-GPN, as well as the conventional GCN, GAT, and APPNP. The corresponding results on the Cora dataset are exemplified in Fig. 2(a), and the results on other datasets are depicted in Figs. A.4-A.6 in Appendix. Besides, the accumulative distribution of normalized neighbor distances regarding node features and graph structure is depicted in Fig. 2(b) to show the strong correlations between message weights learned by PMP-GNNs and the practical data context.

The notable contrast regarding the message weights of PMP-GNNs and conventional message-passing GNNs is shown in Fig. 2 and Figs. A.4-A.6 in Appendix. Taking the result on the Cora dataset as an example, PMP-GCN and PMP-GPN acquire ∼1,000 and ∼700 messages whose weights are extremely low (≤ 0.02), whereas conventional GCN and APPNP identify ∼50 low-weight messages. These comparative results strongly indicate that PMP-GCN and PMP-GPN are much better than conventional GCN and APPNP in learning to reduce or negate messages conveyed by dissimilar neighbors while preserving messages from a significantly reduced number of correlated neighbors. Similarly, PMP-GAT can learn over 3,000 messages with very low weights (≤ 0.02), whereas GAT identifies only ∼1,000 low-weight messages. As message weights (attention scores) for each central node are summed to one, PMP-GAT can concentrate more on truly correlated neighbors in the subsequent feature aggregation stage.

Considering a large portion of neighbors are very distant regarding node features or structure (Fig. 2(b) and Fig. A.3), the results depicted in Fig. 2 and Figs. A.4-A.6 demonstrate that the proposed PMP paradigm enables GNNs to better capture the sparseness of real-world graph data. Therefore, the proposed PMP paradigm empowers GNNs to reduce or negate more messages from dissimilar neighbors, enabling them to learn more expressive representations by appropriately preserving messages conveyed by much fewer but highly similar neighbors.

8. Conclusion

In this paper, we have proposed Polarized message-passing (PMP), a novel paradigm that can lead to novel designs of GNNs. Different from existing strategies, PMP leverages similarity and dissimilarity to acquire neighboring messages, which enables GNNs to learn expressive representations with a significantly reduced number of highly correlated neighbors. Three novel GNNs based on the proposed PMP, namely PMP-GCN, PMP-GAT, and PMP-GPN, are constructed. Theoretical analysis is further conducted to verify the strong expressiveness of PMP-GCN, PMP-GAT, and PMP-GPN. An empirical study involving five diverse downstream tasks from 12 datasets is also conducted to reveal the superior performances of PMP-GCN, PMP-GAT, and PMP-GPN. The results demonstrate that PMP-GCN, PMP-GAT, and PMP-GPN outperform established MPGNNs in all downstream tasks, indicating the effectiveness of the proposed PMP paradigm. In the future, we will further improve the proposed PMP by designing and incorporating novel measures for generating message weights that can reveal the intrinsic properties of graphs. Moreover, we will devote efforts toward adapting PMP to solve more challenging tasks, such as multi-view graph learning.

CRediT authorship contribution statement

Tiantian He: Conceptualization, Formal analysis, Methodology, Validation, Writing – original draft, Writing – review & editing. **Yang Liu:** Conceptualization, Formal analysis, Methodology, Validation, Writing – original draft, Writing – review & editing, Project administration. **Yew-Soon Ong:** Conceptualization, Supervision, Writing – original draft, Writing – review & editing. **Xiaohu Wu:** Conceptualization, Validation, Writing – original draft, Writing – review & editing. **Xin Luo:** Conceptualization, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work is supported in part by the Ministry of Science and Technology of China under Grants 2021ZD0112501 and 2021ZD0112502, theGeneral Research Fund of Hong Kong SAR under Grants RGC/HKBU 12202220 and 12203122, the A*STAR I&E GAP funding (NO. I23D1AG080), the Center for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), and Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE), Nanyang Technological University.

Appendix A. Complete results of data characteristics and learned message weights

A.1. Feature and structure distances of all test datasets

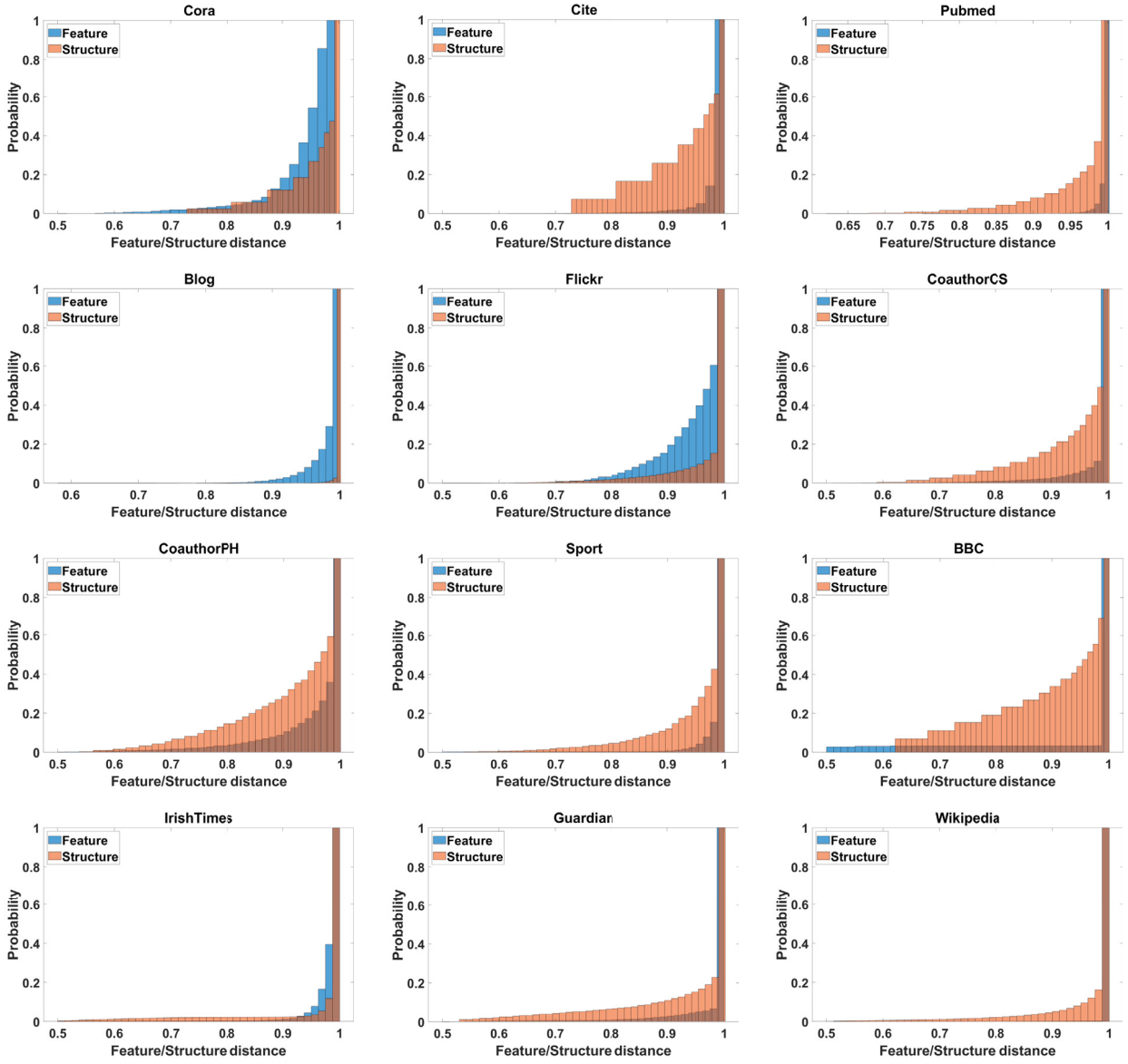


Fig. A.3. Feature and structure distances of all test datasets.

A.2. Message weights learned by PMP-based GNNs and conventional message-passing GNNs

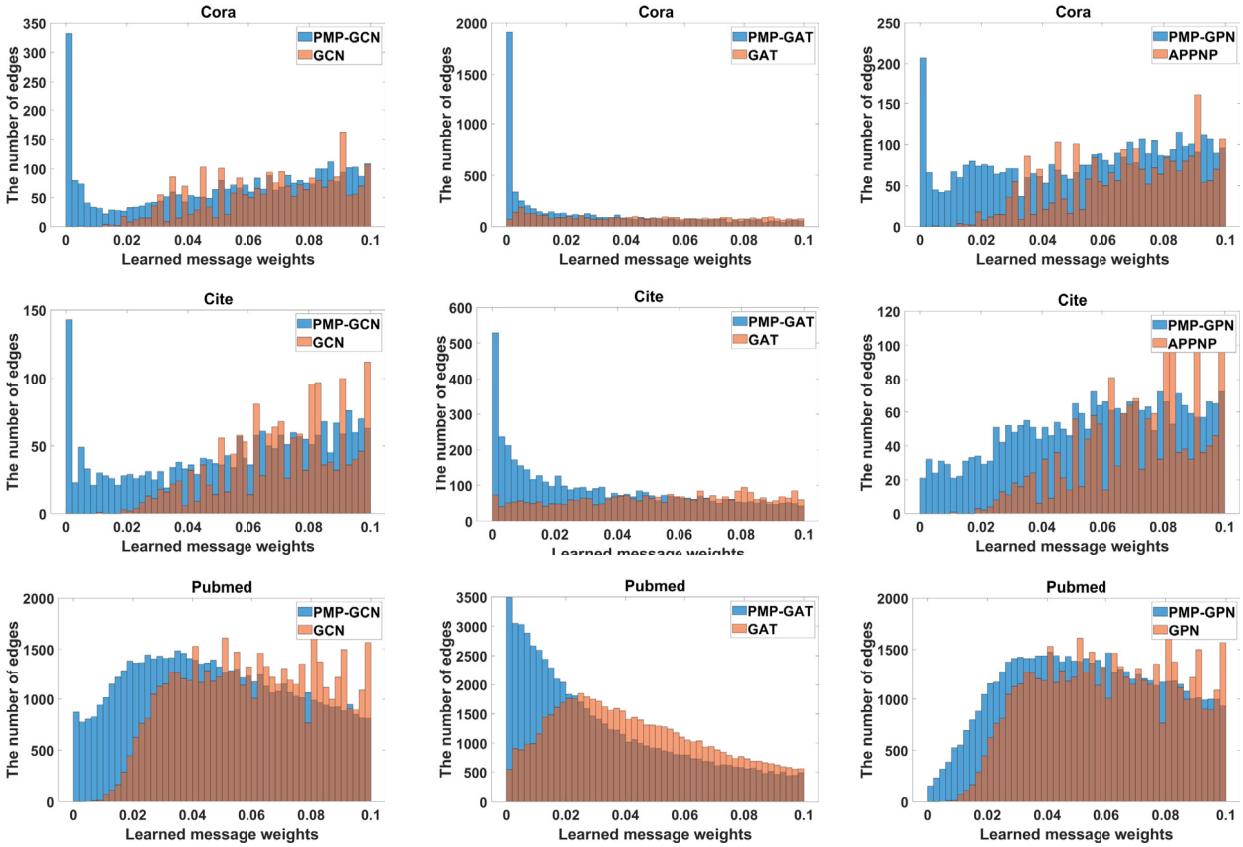


Fig. A.4. Small message weights learned by PMP-based GNNs and conventional message-passing GNNs on article citation graphs.

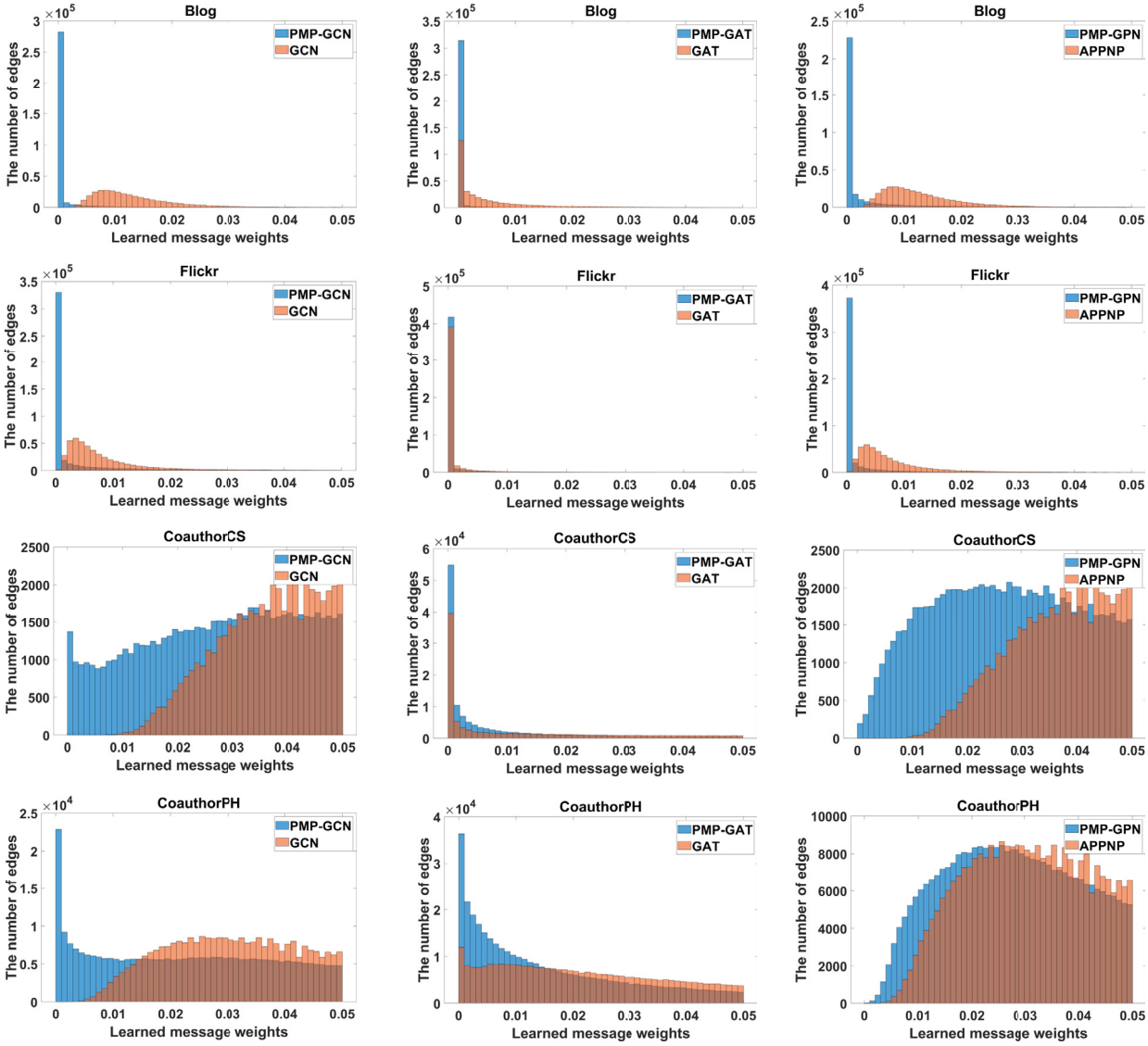


Fig. A.5. Small message weights learned by PMP-based GNNs and conventional message-passing GNNs on social and collaboration graphs.

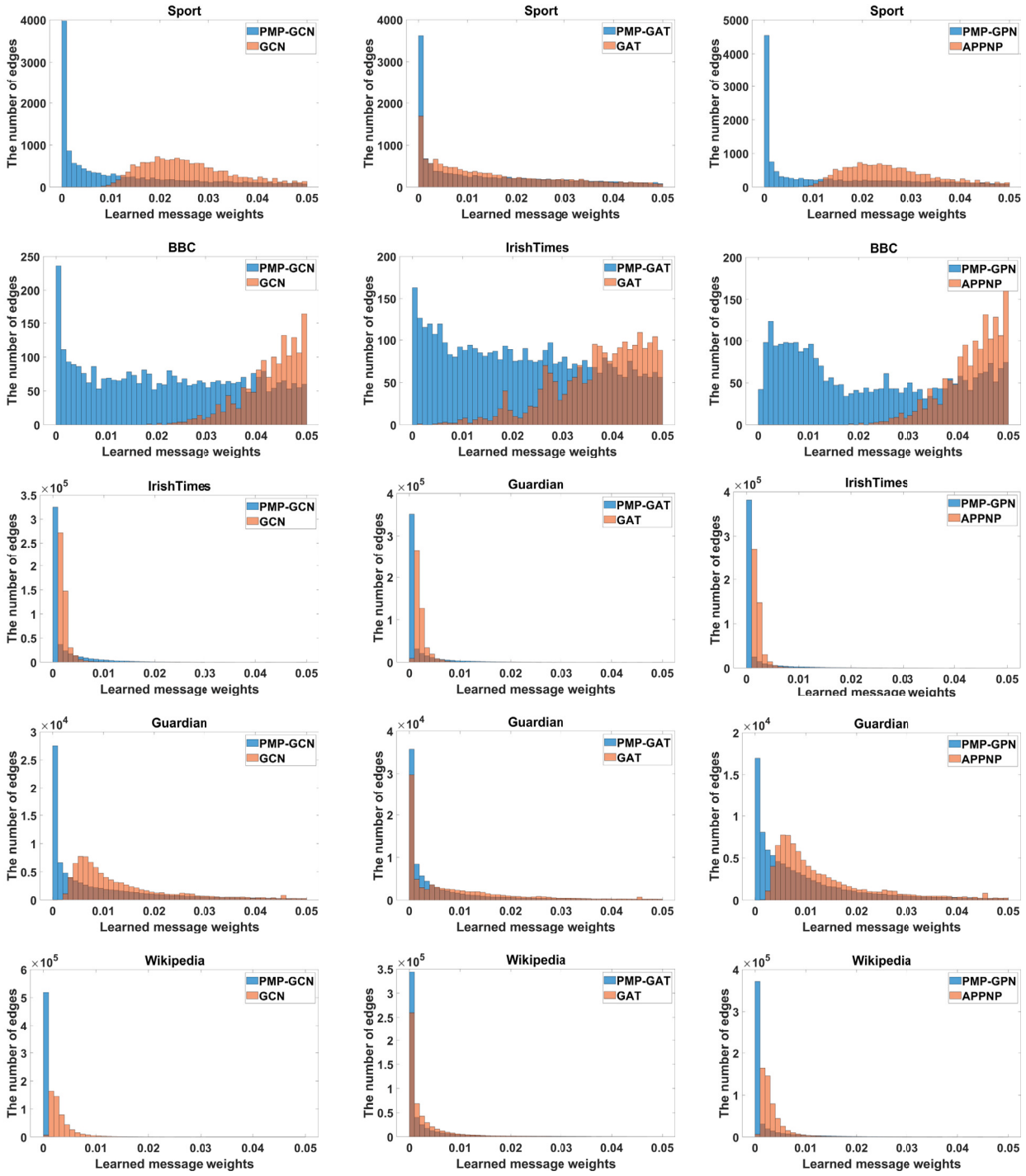


Fig. A.6. Small message weights learned by PMP-based GNNs and conventional message-passing GNNs on rich-text graphs.

Appendix B. Extension of Theorem 1

When multiple data sources or functions are used to compute similarity and dissimilarity between pairwise nodes, the proposed PMP is still more expressive than conventional message-passing GNNs. Several well-defined measures, such as normalized mutual information, distance correlation, and maximal information coefficient [41], can be adapted to reveal the previously mentioned property of the proposed PMP. In this paper, we achieve this by further extending the proof of Theorem 1. Assume that C and S

represent the polarized message sets that PMP has already used to learn representations. And let C' and S' represent new polarized message sets computed according to new data sources or functions. When $f_\psi(\cdot)$ stays unchanged, for a node label Y, we can verify the following:

$$\begin{aligned}
&I(Y;[f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')])\\
&=H([f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')])\\
&-H([f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')]|Y),\\
&=H([f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')])+H(Y)\\
&-H([f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')],Y)\\
&=H(Y)-H(Y|[f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')])\\
&=H(Y)-H(Y|[f_\psi(C),f_\psi(S)])+H(Y|[f_\psi(C),f_\psi(S)])\\
&-H(Y|[f_\psi(C),f_\psi(S)],[f_\psi(C'),f_\psi(S')])\\
&=I(Y;[f_\psi(C),f_\psi(S)])+I(Y;[f_\psi(C'),f_\psi(S')]|[f_\psi(C),f_\psi(S)])\\
&\geq I(Y;[f_\psi(C),f_\psi(S)]).
\end{aligned} \tag{B.1}$$

The analysis above shows that the learning capability of the proposed PMP can be further improved as long as the new data sources or functions are appropriately used to compute polarized messages for the subsequent learning of representations (i.e., $I(Y;[f_\psi(C'),f_\psi(S')]|[f_\psi(C),f_\psi(S)])>0$).

References

[1] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.
[2] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Message passing neural networks, in: Machine Learning Meets Quantum Physics, 2020, pp. 199–214.
[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.
[4] G. Wang, R. Ying, J. Huang, J. Leskovec, Multi-Hop Attention Graph Neural Networks, 2021, pp. 3089–3096.
[5] J. Gasteiger, A. Bojchevski, S. Günnemann, Predict then propagate: graph neural networks meet personalized pagerank, arXiv preprint, arXiv:1810.05997, 2018.
[6] S.L. Feld, Why your friends have more friends than you do, Am. J. Sociol. 96 (1991) 1464–1477.
[7] N. Alipourfard, B. Nettasinghe, A. Abeliuk, V. Krishnamurthy, K. Lerman, Friendship paradox biases perceptions in directed networks, Nat. Commun. 11 (2020) 707.
[8] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[9] J. You, R. Ying, J. Leskovec, Position-aware graph neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 7134–7143.
[10] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6861–6871.
[11] G. Wang, R. Ying, J. Huang, J. Leskovec, Improving graph attention networks with large margin-based constraints, arXiv preprint, arXiv:1910.11945, 2019.
[12] J. Klicpera, S. Weißenberger, S. Günnemann, Diffusion improves graph learning, in: Advances in Neural Information Processing Systems, 2019, pp. 13354–13366.
[13] E. Chien, J. Peng, P. Li, O. Milenkovic, Adaptive universal generalized pagerank graph neural network, in: International Conference on Learning Representations, 2021.
[14] G. Corso, L. Cavalleri, D. Beaini, P. Liò, P. Veličković, Principal neighbourhood aggregation for graph nets, arXiv preprint, arXiv:2004.05718, 2020.
[15] S. Zhang, L. Xie, Improving attention mechanism in graph neural networks via cardinality preservation, in: IJCAI: Proceedings of the Conference, vol. 2020, NIH Public Access, 2020, p. 1395.
[16] A. Wijesinghe, Q. Wang, A new perspective on "how graph neural networks go beyond Weisfeiler-Lehman?", in: International Conference on Learning Representations, 2022.
[17] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, arXiv preprint, arXiv:1810.00826, 2018.
[18] C. Morris, F. Geerts, J. Tönshoff, M. Grohe, Wl meet vc, arXiv preprint, arXiv:2301.11039, 2023.
[19] B. Weisfeiler, A. Leman, The reduction of a graph to canonical form and the algebra which appears therein, NTI Ser. 2 (1968).
[20] T. He, Y.S. Ong, L. Bai, Learning conjoint attentions for graph neural nets, Adv. Neural Inf. Process. Syst. 34 (2021) 2641–2653.
[21] A. Hasanzadeh, E. Hajiramezanali, S. Boluki, M. Zhou, N. Duffield, K. Narayanan, X. Qian, Bayesian graph neural networks with adaptive connection sampling, in: International Conference on Machine Learning, 2020.
[22] K. Yao, J. Liang, J. Liang, M. Li, F. Cao, Multi-view graph convolutional networks with attention mechanism, Artif. Intell. 307 (2022) 103708.
[23] Y. Liang, F. Meng, Y. Zhang, Y. Chen, J. Xu, J. Zhou, Emotional conversation generation with heterogeneous graph neural network, Artif. Intell. 308 (2022) 103714.
[24] L. Huang, D. Ma, S. Li, X. Zhang, H. Wang, Text level graph neural network for text classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3444–3450.
[25] J. You, J.M. Gomes-Selman, R. Ying, J. Leskovec, Identity-aware graph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 10737–10745.
[26] V. Garg, S. Jegelka, T. Jaakkola, Generalization and representational limits of graph neural networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 3419–3430.
[27] D. Chen, L. O'Bray, K. Borgwardt, Structure-aware transformer for graph representation learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 3469–3489.
[28] F.M. Bianchi, D. Grattarola, L. Livi, C. Alippi, Graph neural networks with convolutional arma filters, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2022) 3496–3507.
[29] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, S. Jegelka, Representation learning on graphs with jumping knowledge networks, in: International Conference on Machine Learning, PMLR, 2018, pp. 5453–5462.
[30] H. Gao, S. Ji, Graph representation learning via hard and channel-wise attention networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 741–749.

[31] X. He, L. Li, D. Roqueiro, K. Borgwardt, Multi-view spectral clustering on conflicting views, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10, Springer, 2017, pp. 826–842.
[32] Y. Liu, Y. Liu, K. Chan, Ordinal regression via manifold learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 25, 2011, pp. 398–403.
[33] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, M.M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model cnns, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5115–5124.
[34] S. Brody, U. Alon, E. Yahav, How attentive are graph attention networks?, in: International Conference on Learning Representations, 2022.
[35] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (2008) 93.
[36] X. Huang, J. Li, X. Hu, Label informed attributed network embedding, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, 2017, pp. 731–739.
[37] O. Shchur, M. Mumme, A. Bojchevski, S. Günnemann, Pitfalls of graph neural network evaluation, arXiv preprint, arXiv:1811.05868, 2018.
[38] D. Greene, D. O'Callaghan, P. Cunningham, How many topics? Stability analysis for topic models, in: Proc. European Conference on Machine Learning (ECML'14), 2014.
[39] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
[40] Z. Yang, W. Cohen, R. Salakhudinov, Revisiting semi-supervised learning with graph embeddings, in: International Conference on Machine Learning, PMLR, 2016, pp. 40–48.
[41] X. Liang, Y. Qian, Q. Guo, H. Cheng, J. Liang, AF: An association-based fusion method for multi-modal classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (2021) 9236–9254.
[42] F. Li, Y. Qian, J. Wang, C. Dang, L. Jing, Clustering ensemble based on sample's stability, Artif. Intell. 273 (2019) 37–55.