Knowledge-Based Systems 268 (2023) 110377

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

DPGNN: Dual-perception graph neural network for representation learning

Li Zhou, Wenyu Chen *, Dingyi Zeng, Shaohuan Cheng, Wanlong Liu, Malu Zhang, Hong Qu *

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China

ARTICLE INFO

Article history:
Received 3 October 2022
Received in revised form 2 February 2023
Accepted 6 February 2023
Available online 26 February 2023

Keywords:
Graph neural networks
Graph representation learning
Semi-supervised learning
Message passing

ABSTRACT

Graph neural networks (GNNs) have drawn increasing attention in recent years and achieved remarkable performance in many graph-based tasks, especially in semi-supervised learning on graphs. However, most existing GNNs are based on the message-passing paradigm to iteratively aggregate neighborhood information in a single topology space. Despite their success, the expressive power of GNNs is limited by some drawbacks, such as inflexibility of message source expansion, negligence of node-level message output discrepancy, and restriction of single message space. To address these drawbacks, we present a novel message-passing paradigm, based on the properties of multi-step message source, node-specific message output, and multi-space message interaction. To verify its validity, we instantiate the new message-passing paradigm as a Dual-Perception Graph Neural Network (DPGNN), which applies a node-to-step attention mechanism to aggregate node-specific multi-step neighborhood information adaptively. Our proposed DPGNN can capture the structural neighborhood information and the feature-related information simultaneously for graph representation learning. Experimental results on six benchmark datasets with different topological structures demonstrate that our method outperforms the latest state-of-the-art models, which proves the superiority and versatility of our method. To our knowledge, we are the first to consider node-specific message passing in the GNNs.

1. Introduction

There exists many graph data in life, such as knowledge graph [1], citation networks [2,3] and traffic networks [4], which traditional deep neural networks (DNNs) are very limited to process. And in recent years, Graph Neural Networks (GNNs) [5] have been widely adopted in various graph-based tasks, such as node classification [6,7], graph classification [8,9], link prediction [10–12], clustering tasks [13,14], and knowledge tracing tasks [15,16].

GNNs form an effective framework for the representation learning of graphs [17,18]. And most existing GNNs [2,3,19] are mainly based on the message-passing paradigm, which iteratively aggregates neighborhood information to update a new representation of each node. However, one layer of GNNs only considers immediate neighbors and the performance degrades greatly when stacking multi-layer GNNs for a larger neighborhood receptive field. Recent studies have attributed this phenomenon to the over-smoothing problem [20,21]. To learn more effective node representations, various approaches dedicate to the breadth, depth, and strength of models. Some efforts focus on how to obtain multi-hop neighborhood information on a single-layer network [7,22,23]; some researches concentrate on designing deep GNN frameworks [24–26]; and some methods purpose to grow into a data augmenter [27–29].

However, despite GNNs revolutionizing graph representation learning, there are limitations to the expressive power of GNNs [30,31]. On the one hand, these models mostly follow the traditional message-passing paradigm in which the iterative operations are characteristic of the entire process. In this message-passing paradigm, the message source expansion is not flexible, and the node-level message output discrepancy for different neighborhood ranges is also not considered [32]. On the other hand, the existing message-passing paradigm is limited in a single topology space. Some experiments have verified that the original graph topology [21] is the fundamental reason for the over-smoothing problem, because the nodes may receive messages with low information-to-noise ratio.

* Corresponding authors.
E-mail addresses: li_zhou@std.uestc.edu.cn (L. Zhou), cwy@uestc.edu.cn (W. Chen), zengdingyi@std.uestc.edu.cn (D. Zeng), shaohuancheng@std.uestc.edu.cn (S. Cheng), liuwanlong@std.uestc.edu.cn (W. Liu), maluzhang@uestc.edu.cn (M. Zhang), hongqu@uestc.edu.cn (H. Qu).

https://doi.org/10.1016/j.knosys.2023.110377
0950-7051/

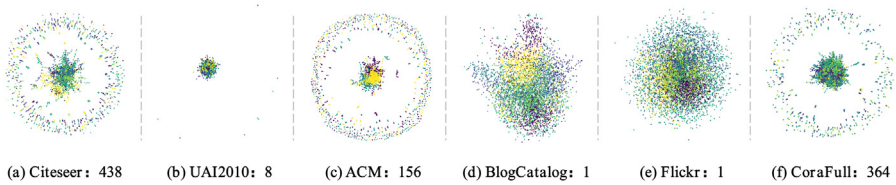


Fig. 1. The topological views of six graph datasets. For intuitive presentation, we adopt different colors to distinguish nodes of different classes and set edges colorless. The number after the graph dataset name indicates the number of connected subgraphs of the graph. For example, there are 438 connected subgraphs in Citeseer, while Flickr is a big connected graph. Graph datasets Citeseer, ACM and CoraFull have similar topological structures, which are mainly composed of a large connected subgraphs (the middle part) and some small connected sub-graphs or isolated nodes (the surrounding part), while the topological structures of graph datasets UAI2010, Flickr and BlogCatalog are denser.

Most exiting GNNs primarily utilize graph topological structure for information propagation and representation learning, in which unfavorable topological structure may lead nodes receive too much noise after multiple steps of information propagation. Fig. 1 shows topological views of six graph datasets drawn by *networkx* [33], in which nodes are positioned by Fruchterman–Reingold force-directed algorithm [34]. Ideally, nodes of the same class desire to be more connected by edges, while there are many inter-class edges in some graph topologies. And some nodes exist in small connected subgraphs, including a multi-hop information capturing limitation.

In fact, graphs with nodes connected with different classes are common in the real world, and it is also common that nodes are strongly correlated but in different connected subgraphs. For example, different amino acid types are more likely to connect in protein structures [35], fraudsters are more likely to connect to accomplices than to other fraudsters in online purchasing networks [36], and preferences for seasonal goods may be the same in different regions. To break the bottleneck caused by graph topological structures in GNNs, some works [37–39] began to focus on other potential message-passing spaces to enrich graph representation learning and some methods [13,40,41] propose graph structure learning techniques used for message passing. But most of these methods regard Graph Convolutional Networks (GCN) as the base encoder framework.

Motivated by observations like the above, in this paper, we summarize the drawbacks of the existing message-passing paradigm including inflexibility of message source expansion, negligence of node-level message output discrepancy, and restriction of single message space. Then we propose an improved message-passing paradigm that can support both node-specific multi-step message aggregation and multi-space interaction. Based on the new message-passing paradigm, we propose a novel Dual-Perception Graph Neural Network (DPGNN) for graph representation learning.

Our main contributions are summarized as follows:

- We formalize the existing message-passing paradigm, analyze its drawbacks, and present a novel improved message-passing paradigm based on the properties of multi-step message source, node-specific message output, and multi-space message interaction.
- We instantiate the new message-passing paradigm as a Dual-Perception Graph Neural Network (DPGNN), which applies a node-to-step attention mechanism to aggregate node-specific multi-step neighborhood information adaptively and captures the structural neighborhood information and the feature-related information simultaneously for graph representation learning.
- we apply DPGNN in the semi-supervised node classification task on six graph datasets with different topological structures. The experimental results demonstrate that our instantiated DPGNN outperforms related state-of-the-art GNNs, and we conduct analysis experiments to prove the superiority and versatility of our proposed message-passing paradigm.

The remainder of this article is organized as follows: the most related previous works are reviewed briefly in Section 2; the improved message-passing paradigm is defined in Section 3; the instantiated methods and experimental evaluations are presented in Section 4 and V respectively; and the conclusion is given in the last part.

2. Related works

In this section, we review relevant research in our field. We divide most current GNNs into three categories: (1) GNNs based on graph topology, (2) GNNs based on node features, and (3) GNNs based on graph structure learning.

2.1. GNNs based on graph topology

The original GNNs apply message passing mainly based on graph topology, which are improved on the breadth, depth, and strength of models for better performance. Inspired by the effectiveness of Convolutional Neural Networks (CNNs) on grid-like data such as images [42–44], the vanilla GCN is proposed to show an efficient variant of convolutional neural networks which can operate directly on graphs. The propagation rule of vanilla GCN [2] can be explained via the approximation of the spectral graph convolutions [45,46]. GAT [3] introduces an attention-based architecture to compute the hidden representations of each node in the graph. MixHop [22] can learn a general class of neighborhood mixing relationships by repeatedly mixing feature representations of neighbors at various distances. GraphMix [29] is a regularization method in which a fully-connected network is jointly trained with the graph neural network via parameter sharing and interpolation-based regularization.

However, these methods are too shallow to consider high-order nodes and the size of the utilized neighborhood is hard to extend. To address this, Klicpera et al. [24] present a personalized propagation of neural predictions (PPNP) and its fast approximation APPNP based on personalized PageRank for a large neighborhood receptive field. Based on this, Chen et al. [26] propose a deep GCN model with initial residual and identity mapping. At each layer, initial residual constructs a skip connection from the input layer, while identity mapping adds an identity matrix to the weight matrix. And DAGNN [25] can adaptively incorporate information from large receptive fields by decoupling representation transformation and propagation. Feng et al. [28] first design a random propagation strategy to perform graph data augmentation and leverage consistency regularization in GRAND model to mitigate the issues of over-smoothing and non-robustness.

2.2. GNNs based on node features

Real-world graphs are noisy, i.e. adjacent nodes may not be similar, and similar nodes are not necessarily adjacent in a topological structure. Therefore, there are limitations in information aggregation and representation learning only considering topology space. In light of this, some works begin to focus on multi-space representation fusion [39,47,48]. In this work, we concentrate on the perspective of node features. Gao et al. [49] propose a learnable graph convolutional layer (LGCL) to select a fixed number of neighboring nodes for each feature based on value ranking. In this way, the generic graph can be transformed into grid-like data that regular convolutional networks can operate on. So this approach considers feature importance to change the form of graph data, but the convolution aggregator in LGCL cannot be directly applied to graphs. LA-GCN [50] also concentrates on the importance of different features, introducing a learnable aggregator for GCN and proposing a new attention mechanism allowing both node-level and feature-level attention. Sambaran et al. [51] propose an unsupervised algorithm MIRand, which creates a multi-layer graph (including structure layer and content layer) and employs a random walk that exploits the informativeness of a node by unifying its structure and attributes. In MIRand, the constructed content-layer graph is always directed and weighted. RolEANE [17] propose a neighbor optimization strategy, which is used to efficiently and seamlessly integrate the network topological structure and attribute information to improve representation learning performance. Geom-GCN [37] maps node features to representation vectors, which can be considered as the position of each node in a latent continuous space. Then, based on the original graph and the latent space, Geom-GCN builds a structural neighborhood, and applies GCN to aggregate information of different neighborhoods. Both AMGCN [38] and SCRL [52] construct feature graphs by input features of nodes and then apply GCN encoders to extract effective information. The difference is that the former learns specific and common embeddings from both feature and topology graph, constraining their consistency and diversity during training. And the latter mainly designs a self-supervised loss to maximize the agreement of the embeddings of the same node in two view graphs.

2.3. GNNs based on graph structure learning

In addition, several efforts have been made to alleviate the imperfection that GNNs rely on the good quality of raw graph topological structure. So some graph structure learning methods are proposed. For homogeneous graphs, Franceschi et al. propose a framework LDS [40] that can learn the graph structure and the parameters of a GNN simultaneously. GLCN [41] aims to learn an optimal graph structure by integrating both graph learning and graph convolution in a unified network architecture. And IDGL [53] is proposed to jointly learn the graph structure and graph embeddings by optimizing a joint loss combining both task prediction loss and graph regularization loss. For heterogeneous graphs, GTN [54] is capable of generating new graph structures, which involves identifying useful connections between unconnected nodes on the original graph. HGSL [55] is designed to learn an optimal heterogeneous graph structure.

These methods focus on the reconstruction and consideration of graph neural network frameworks for better graph representation learning. However, their message-passing paradigm remains an iterative design and ignores the node-level message output discrepancy. In light of this, we further formalize and analyze the existing message-passing paradigm, and redefine an improved paradigm.

3. An improved message-passing paradigm

A graph is formally defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_u$ represents the union of N_l labeled nodes ($\mathcal{V}_l$) and N_u unlabeled nodes ($\mathcal{V}_u$), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges between nodes, $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$ represent the number of nodes and edges respectively. The node features are denoted as $\mathbf{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N-1}\} \in \mathbb{R}^{N\times d}$, where d is the dimension of node features and each node feature $\boldsymbol{x}_i$ is a non-negative vector. $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix of $\mathcal{G}$, with each element $\mathbf{A}_{ij} = 1$ associating there exists an edge between node i and node j, otherwise $\mathbf{A}_{ij} = 0$. In an undirected graph, $\mathbf{A}_{ij} = \mathbf{A}_{ji}$. Noteworthy, $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{D}^{-1/2}$ is the symmetric normalization of the adjacency matrix, in which $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ indicates that all nodes in the graph have added self-loop edges, $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ is the diagonal degree matrix. The labels of N_l labeled nodes are denoted as $\mathbf{Y}_l = \{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_l-1}\} \in \mathbb{R}^{N_l\times C}$, where $\boldsymbol{y}_i$ is a one-hot vector and C is the number of classes.

For semi-supervised classification, only a few nodes observe their labels, while other nodes' labels are missing, i.e., $0 < N_l \ll N_u$. The message-passing space mostly relies on the given $\mathcal{G}$. So the task is to design a graph neural network $\mathbf{Z} = f(\mathbf{X}, \mathbf{A}; \Theta)$ to learn the node representations and predict the label of unlabeled nodes finally, where Θ represents the trainable parameters.

The core of GNNs is the message-passing paradigm, which defines the way of information interaction between nodes and affects the representation learning ability of GNNs. In this section, we first discuss the existing message-passing paradigm of GNNs, analyze its limitations, and present an improved message-passing paradigm. For the convenience of subsequent discussion, we define some notations.

Notation 1. *Let $d(i,j)$ be the shortest path length from node i to node j in the $\mathcal{G}$. Particularly, $d(i,i) = 0$, $d(i,j) = 1$ if $(i,j) \in \mathcal{E}$, and $d(i,j) = \infty$ if there is no path between node i and node j.*

Notation 2. *Let $c_\iota(i,j)$ count the number of paths with length ι from node i to node j. Particularly, for any ι, $c_\iota(i,j) = 0$ if $d(i,j) = \infty$.*

Notation 3. *Let $\mathcal{N}_\iota(i)$ be the set of nodes in which $\forall j \in \mathcal{N}_\iota(i)$ satisfies $d(i,j) = \iota$. Particularly, $\mathcal{N}_0(i) = \{i\}$, and $\mathcal{N}_1(i)$ denotes the immediate neighbors of node i.*

Notation 4. *Let $\mathcal{J}_\iota(i)$ be the set of nodes in which $\forall j \in \mathcal{J}_\iota(i)$ satisfies $c_\iota(i,j) > 0$. Particularly, $\mathcal{N}_\iota(i) \subseteq \mathcal{J}_\iota(i)$.*

3.1. The existing message-passing paradigm of GNNs

The goal of GNNs is to learn meaningful node representations, which can contain enough abundant but distinguishable information. And the message-passing paradigm of most existing GNNs is iterative passing and aggregation of local neighborhood messages based on topological structure. Let $\mathfrak{m}_i^{(t)}$ be the message of node i obtained in the iteration t, and its message aggregation can be expressed simply as the sum of the immediate neighborhood messages:

$$\mathfrak{m}_i^{(t)} = \alpha_i \mathfrak{m}_i^{(t-1)} + \sum_{j\in\mathcal{N}_1(i)} \beta_j \mathfrak{m}_j^{(t-1)}, \tag{1}$$

where α_i controls the amount of message retention at node i and β_j controls the amount of message input from node j.

As shown in Fig. 2(a), the existing message-passing paradigm in each iteration, akin to a star-shaped pattern, can define the breadth of each message-passing iteration. The iteration of the existing message-passing paradigm enables message interaction between nodes that have reachable paths but are not directly connected. We regard this phenomenon as message transitivity. And as shown in Fig. 2(b), message transitivity, akin to a chain-shaped pattern, can ensure the depth of message passing over multiple iterations.

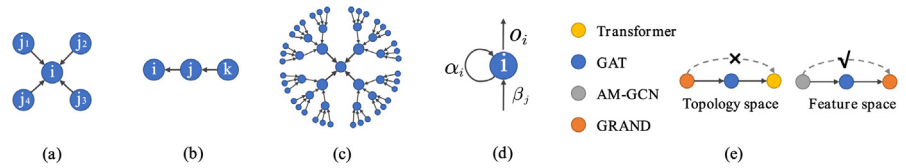


Fig. 2. (a) The existing message-passing paradigm, akin to a star-shaped pattern, defines the breadth of each message-passing iteration. (b) Message transitivity, akin to a chain-shaped pattern, ensures the depth of message passing over multiple iterations. (c) The boosting breadth of incoming messages requires increasing the depth of message passing. (d) The message-passing paradigm is analyzed from three aspects: message retention, message input, and message output. (e) An example illustrating the heterogeneity of different message-passing spaces.

3.2. Drawbacks of the existing message-passing paradigm

Although most GNNs follow the message-passing paradigm mentioned above, there still exists some drawbacks, which are summarized in three aspects.

(1) **Inflexibility of message source expansion:** In each iteration, each node i can only aggregate its immediate neighborhood information, and its aggregation paradigm is static, i.e. the amount of message retention α_i and the amount of message input β_j for node j are independent and unchanged. Therefore, on the one hand, boosting the breadth of incoming messages requires increasing the depth of message passing. As shown in Fig. 2(c), when breadth is realized by increasing depth, the incoming messages increase exponentially, in which each node tends to obtain the entire graph information, and becomes indistinguishable finally. On the other hand, although iteratively transitivity can introduce messages from far-hop nodes, it also weakens messages from near-hop nodes.

(2) **Negligence of node-level message output discrepancy:** From Eq. (1), we can observe that the message-passing paradigm only focuses on message retention and input in each iteration, ignoring the amount of message output for each node after message aggregation. As shown in Fig. 2(d),[1] in the existing message-passing paradigm, the output control coefficient o_i of the aggregated message sum in each iteration for each node i is always set to 1, while in fact, each node, may pay different attention to different neighborhood ranges. See Appendix for the further instance analysis.

(3) **Restriction of single message space.** In most GNNs, message passing is based on the original topological structure. However, the distance between nodes may be varied in different message spaces. For example, in the topology space, short-distance dependent nodes are not necessarily feature-correlated with each other, while long-distance dependent nodes may be very close in the feature space. More specifically, as shown in Fig. 2(e), in topology space, the paper GAT [3] cites Transformer [56] and GRAND [28] cites GAT, but the correlation between GRAND and Transformer is very weak, and the information interaction between them may introduce noise information to each other. However, in feature space, the content of AM-GCN [38] is similar to GAT, and GAT is similar to GRAND, so AM-GCN is likely to be related to GRAND. Therefore, with the heterogeneity of different spaces, messages passing in only one space will lose other latent information interaction opportunities.

3.3. The proposed message-passing paradigm

To resolve the aforementioned drawbacks, we extend the original message-passing paradigm from three aspects: multi-step message source, node-specific message output, and multi-space message interaction.

[1] All the message-passing paradigm can be generalized as $\mathbf{m}_i^{(t)} = o_i(\alpha_i \mathbf{m}_i^{(t-1)} + \sum_{j\in\mathcal{N}_1(i)} \beta_j \mathbf{m}_j^{(t-1)})$. When $o_i = 1$, the above equation is simplified to the existing message-passing paradigm (i.e., Eq. (1)).

Multi-step message source: we broaden the message sources setting without extra iteration, in which the incoming messages are not limited to the immediate neighborhood. We regard the nodes which can be reached at the given step length ι as the message sources. To enrich message interactions from nodes of different step lengths, we define ι as the range of 0 to L. When $\iota = 0$, the incoming messages come from the node itself. The aggregated message of node i is formally defined as follows:

$$\bar{\mathfrak{m}}_i = \sum_{\iota=0}^{L} \left(\sum_{j\in\mathcal{J}_\iota(i)} w_{\iota,ij} \mathfrak{m}_j \right), \tag{2}$$

where $0 < w_{\iota,ij} < 1$ is a transfer coefficient for evaluating the amount of message transfer from node j to node i under step length ι. We use '-' to distinguish between messages before aggregation and messages after aggregation.

Node-specific message output: we focus on the differentiated control of node-specific message aggregation under various step lengths. So we further extend the message-passing paradigm (Eq. (2)) as follows:

$$\bar{\mathfrak{m}}_i = \sum_{\iota=0}^{L} o_{i\iota} \left(\sum_{j\in\mathcal{J}_\iota(i)} w_{\iota,ij} \mathfrak{m}_j \right), \tag{3}$$

where $\sum_{\iota=0}^{L} o_{i\iota} = 1$ and $o_{i\iota} > 0$, indicating that node i pays attention to the message sources in step ι.

Multi-space message interaction: we expand the original message-passing paradigm from a single space to multiple spaces, which can improve the correlation degree of node messages. The formalization is defined as:

$$\mathfrak{m}_i = Agg\left(\left\{ \bar{\mathfrak{m}}_i^{\varsigma} | \varsigma \in \Omega \right\} \right), \tag{4}$$

where $Agg(\cdot)$ is a message aggregator function, Ω is the message-passing space set, $\bar{\mathfrak{m}}_i^{\varsigma}$ denotes the aggregated message of node i in corresponding space ς. In this paper, we realize $\Omega = \{t, f\}$, related to the topology space and feature space. So $\bar{\mathfrak{m}}_i^{t}$ is derived from message aggregation in the original topology graph $\mathcal{G}^t$, while $\bar{\mathfrak{m}}_i^{f}$ comes from message aggregation in a new feature graph $\mathcal{G}^f$. The new feature graph is constructed by the similarity of node features with KNN algorithm [38,52].

Overall, we summarize and propose a new generalized message-passing paradigm, defined as follows:

$$\begin{aligned} \mathfrak{m}_i &= Agg\left(\left\{ \bar{\mathfrak{m}}_i^{\varsigma} | \varsigma \in \Omega \right\} \right), \\ \bar{\mathfrak{m}}_i^{\varsigma} &= \sum_{\iota=0}^{L^{\varsigma}} o_{i\iota}^{\varsigma} \left(\sum_{j=0}^{N-1} w_{\iota,ij}^{\varsigma} \mathfrak{m}_j^{\varsigma} \right). \end{aligned} \tag{5}$$

Particularly, a carefully designed $w_{\iota,ij}^{\varsigma}$ should exhibit the following properties:

(1) **Message closeness:** $w_{\iota,ij_1}^{\varsigma} \le w_{\iota,ij_2}^{\varsigma}$ if $d(i,j_1) \ge d(i,j_2)$ in the message space ς, which implies that the closer the distance between nodes, the more their information interaction.

(2) **Message denseness:** $w_{\iota,ij_1}^{\varsigma} \le w_{\iota,ij_2}^{\varsigma}$ if $c_\iota(i,j_1) \le c_\iota(i,j_2)$ in the message space ς, which implies that the more accessible paths between nodes, the greater their potential association.

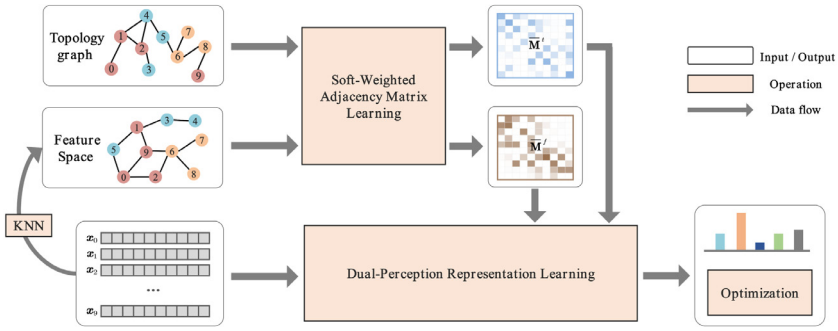


Fig. 3. The framework of DPGNN model. (1) *KNN Module*: which constructs a feature graph $\mathcal{G}^f$ by the similarity of node features [38,52]. (2) *Soft-Weighted Adjacency Matrix Learning Module*: which learns two soft-weighted adjacency matrices $\overline{\mathbf{M}}^t$ and $\overline{\mathbf{M}}^f$ based on the two message-passing space. (3) *Dual-Perception Representation Learning Module*: which captures the structural neighborhood information and the feature-related information simultaneously for each node. (4) *Optimization Module*: which defines the loss function for DPGNN.

(3) **Message irrelevance:** $w^{\varsigma}_{\iota,ij} = 0$ if $c_\iota(i,j) = 0$, which implies that if there is no reachable path of length ι between node i and node j, there will be no message passing.

In the new message-passing paradigm, each node i can acquire abundant and unique information. Specifically, on the one hand, the new message-passing paradigm directly increases the message receptive field without iteration operations and provides a diversity of message sources. On the other hand, each node i can capture rich information from different message-passing space.

4. Dual-perception graph neural network

Generally, there are many different ways of implementing our improved message-passing paradigm, leading to GNNs with different expressive powers. In this section, we propose a novel Dual-Perception Graph Neural Network (DPGNN) that is an instantiation of our improved message-passing paradigm.

To facilitate implementation, we reconstruct Eq. (5) as follows:

$$\bar{\mathbf{m}}^{\varsigma}_i = \sum_{\iota=0}^{L^{\varsigma}} \sum_{j=0}^{N-1} o^{\varsigma}_{i\iota} w^{\varsigma}_{\iota,ij} \mathbf{m}^{\varsigma}_j = \sum_{j=0}^{N-1} \mathbf{M}^{\varsigma}_{ij} \mathbf{m}^{\varsigma}_j, \tag{6}$$

$$\mathbf{M}^{\varsigma}_{ij} = \begin{cases} \sum_{\iota=0}^{L^{\varsigma}} o^{\varsigma}_{i\iota} w^{\varsigma}_{\iota,ij}, & if\ d(i,j) \neq \infty, \\ 0, & otherwise, \end{cases} \tag{7}$$

where $\mathbf{M}^{\varsigma}_{ij}$ quantifies the message interaction between node i and j in the message-passing space ς, and $\mathbf{M}^{\varsigma} = \left(\mathbf{M}^{\varsigma}_{ij}\right)_{i,j\in\mathcal{V}}$ is a soft-weighted adjacency matrix.

Therefore, the key of DPGNN is: (1) how to obtain $\mathbf{M}^{\varsigma}$ in each message-passing space ς, and (2) how to learn the final representation of nodes. This corresponds exactly to its two modules: *Soft-Weighted Adjacency Matrix Learning* & *Dual-Perception Representation Learning*. The framework of DPGNN is shown in Fig. 3.

4.1. Soft-weighted adjacency matrix learning

In this part, we will introduce the soft-weighted adjacency matrix learning method $\varphi : \mathcal{G}(\mathcal{V},\mathcal{E}) \to \mathbf{M}$ for a general message-passing space $\mathcal{G}$. We firstly provide a definition of $w_{\iota,vu}$ that satisfies the properties of message closeness, message denseness, and message irrelevance. We find that $w_{\iota,ij} \propto \mathbf{A}^{\iota}_{ij}$, where $\mathbf{A}^{\iota}$ is a matrix product of ι copies of $\mathbf{A}$, and $\mathbf{A}^{\iota}_{ij} = c_\iota(i,j)$. To normalize message aggregation, we apply $\hat{\mathbf{A}}$ as the transfer coefficient base, so we define $w_{\iota,ij}$ as follows:

$$w_{\iota,ij} = \hat{\mathbf{A}}^{\iota}_{ij}. \tag{8}$$

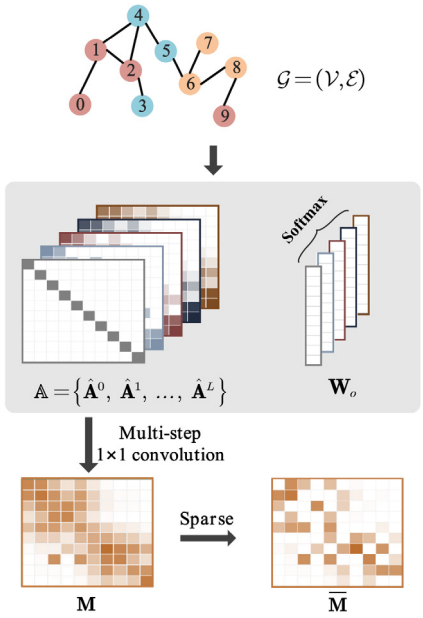


Fig. 4. Soft-Weighted Adjacency Matrix Learning Module.

Directly, as shown in Fig. 4, we obtain the matrix set $\mathbb{A} = \left\{\hat{\mathbf{A}}^0, \hat{\mathbf{A}}^1, \ldots, \hat{\mathbf{A}}^L\right\}$, which involves a series of neighborhood information of different step lengths. Then we compute the convex combination of each node by 1×1 convolution with non-negative weights by using node-to-step attention mechanism:

$$\mathbf{M}_{ij} = \sum_{\iota=0}^{L} o_{i\iota} \hat{\mathbf{A}}^{\iota}_{ij}, \tag{9}$$

where $o_{i\iota}$ is the element of the learnable $\mathbf{W}_o$ after softmax operation, $\mathbf{W}_o \in \mathbb{R}^{N\times L}$, and $o_{i\iota}$ can represents the attention weight of node i to the information in step length ι.

Intuitively, we can regard $\mathbf{M}$ as the weighted adjacency matrix of a generated graph structure, which expands the original graph structure by adding edges to nodes that are not directly connected but have an accessible path, acting like a message-passing extender and controller. However, the added edges cause the graph to become denser, which behaves like a double-edged sword. To avoid redundancy of information aggregation, we introduce a random graph sparse strategy. Formally, we first randomly sample a binary mask $\epsilon_{ij} \sim Bernoulli(1-p)$ for each node pair.

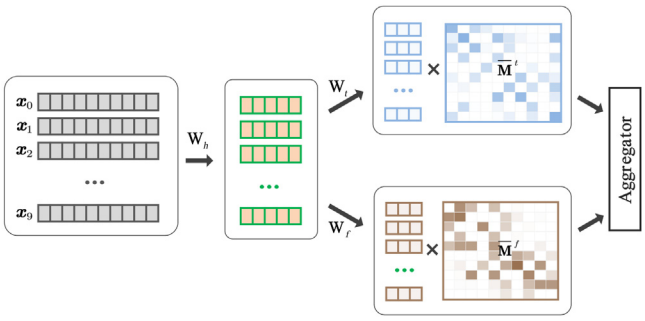


Fig. 5. Dual-Perception Representation Learning Module.

Second, we obtain the sparse weighted adjacency matrix $\overline{\mathbf{M}}$ by multiplying each edge weight with its corresponding mask:

$$\overline{\mathbf{M}}_{ij} = \epsilon_{ij} \cdot \mathbf{M}_{ij}. \tag{10}$$

Finally, we scale $\overline{\mathbf{M}}$ with the factor of $\frac{1}{1-p}$ to guarantee the weighted adjacency matrix is in expectation equal to $\mathbf{M}$.

By this strategy, we increase the randomness and the diversity of message passing, avoid the strong dependence of node representation learning on the newly generated graph structure, and reduce the computational complexity of message aggregation. Note that the random graph sparse strategy is only performed during training. During inference, we directly set $\overline{\mathbf{M}}$ as the original $\mathbf{M}$. In this way, based on topology graph and feature graph, we construct two soft-weighted adjacency matrix $\overline{\mathbf{M}}^t$ and $\overline{\mathbf{M}}^f$.

4.2. Dual-perception representation learning

In this part, we propose a Dual-Perception representation learning method based on topology space and feature space simultaneously, which adapt to the new message-passing paradigm. As shown in Fig. 5, we firstly apply a weight-shared layer to transform each node feature into a low-dimensional space:

$$\boldsymbol{h}_i = \mathrm{ReLU}\left(\boldsymbol{x}_i \mathbf{W}_h + \boldsymbol{b}_h\right), \tag{11}$$

where $\mathbf{W}_h \in \mathbb{R}^{d \times d_h}$, $\boldsymbol{b}_h \in \mathbb{R}^{d_h}$ are learnable parameters, d is the dimension of node features, and d_h is the hidden dimension. Secondly, two weight-exclusive layers are adopted to learn the node representations based on topology space and feature space in parallel:

$$\boldsymbol{m}_i^t = \boldsymbol{h}_i \mathbf{W}_t + \boldsymbol{b}_t, \tag{12}$$

$$\boldsymbol{m}_i^f = \boldsymbol{h}_i \mathbf{W}_f + \boldsymbol{b}_f, \tag{13}$$

where $\mathbf{W}_t, \mathbf{W}_f \in \mathbb{R}^{d_h \times C}$, $\boldsymbol{b}_t, \boldsymbol{b}_f \in \mathbb{R}^C$, C represents the number of classes. Now $\boldsymbol{m}_i^t$ and $\boldsymbol{m}_i^f$ represent the message of node i in corresponding space. Then we apply information aggregation for each message-passing space:

$$\bar{\boldsymbol{m}}_i^t = \sum_{j=0}^{N-1} \overline{\mathbf{M}}_{ij}^t \boldsymbol{m}_j^t, \tag{14}$$

$$\bar{\boldsymbol{m}}_i^f = \sum_{j=0}^{N-1} \overline{\mathbf{M}}_{ij}^f \boldsymbol{m}_j^f. \tag{15}$$

Finally, we choose a message aggregator to obtain the final representation $\boldsymbol{z}_i$ of node i:

$$\boldsymbol{z}_i = Agg(\bar{\boldsymbol{m}}_i^t, \bar{\boldsymbol{m}}_i^f) \in \mathbb{R}^C. \tag{16}$$

The aggregator function mainly integrates topology-based node representations and feature-based node representations. In practice, we primarily examined three aggregator functions: Attention aggregator, Mean-pooling aggregator and Max-pooling aggregator. Based on DPGNN, node representations learning can capture a large range of neighborhood information in two different spaces adaptively.

4.3. Optimization

To verify the ability of representation learning, we apply DPGNN in the semi-supervised node classification task. Inspired by recent advances in semi-supervised learning [28,57], we define the loss function as two parts: Cross-Entropy loss and Low-Entropy loss, which helps to train a low-entropy and strong-robustness model.

Cross-Entropy Loss: This part follows the loss function of most semi-supervised learning node classification tasks, only focusing on n_l labeled nodes used for training. Firstly, the predicted probability $\boldsymbol{p}_i$ of each node i is obtained from the final node representation:

$$\boldsymbol{p}_i = softmax\left(\boldsymbol{z}_i\right) = \exp\left(\boldsymbol{z}_i^c\right) / \sum_{c=0}^{C-1} \exp\left(\boldsymbol{z}_i^c\right), \tag{17}$$

where $\boldsymbol{z}_i^c$ is cth element of $\boldsymbol{z}_i$. Then the Cross-Entropy loss is defined as follows:

$$CE\left(\mathbf{P}_l, \mathbf{Y}_l\right) = -\sum_{i=0}^{N_l-1} \boldsymbol{y}_i^{\mathrm{T}} \log\left(\boldsymbol{p}_i\right), \tag{18}$$

where $\mathbf{P}_l = \{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{N_l-1}\} \in \mathbb{R}^{N_l \times C}$ and $\mathbf{Y}_l = \{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_l-1}\} \in \mathbb{R}^{N_l \times C}$ are the predicted probability distribution and true labels of N_l labeled nodes respectively.

Low-Entropy Loss: As a model for classification, we expect to realize an ideal model that is very certain about the predicted results. This means that the predicted probability distribution tends to have a low entropy. So we apply a sharpening function $sharpen\left(\cdot\right)$ to reduce the entropy of the label distribution [57] to obtain an anchor label $\tilde{\boldsymbol{p}}_i$, the element of which is defined as follow:

$$\tilde{\boldsymbol{p}}_i = sharpen\left(\boldsymbol{p}_i, \tau\right) = \boldsymbol{p}_i^{c\,\frac{1}{\tau}} / \sum_{c=0}^{C-1} \boldsymbol{p}_i^{c\,\frac{1}{\tau}}, \tag{19}$$

where $\boldsymbol{p}_i^c$ represents the probability that node i belongs to class c, τ is a hyper-parameter named temperature. The low temperature stimulates the model to produce lower-entropy predictions. When $\tau \to 0$, $\tilde{\boldsymbol{p}}_i$ will approach a one-hot distribution. Then we evaluate the gap between the predicted probability distribution of all nodes and their corresponding anchors, and hope to reduce this gap during training, which is defined as a Low-Entropy loss for all nodes:

$$LE\left(\tilde{\mathbf{P}}, \mathbf{P}\right) = \sum_{i=0}^{N-1} \left\| \tilde{\boldsymbol{p}}_i - \boldsymbol{p}_i \right\|^2, \tag{20}$$

where $\tilde{\mathbf{P}} = \{\tilde{\boldsymbol{p}}_0, \tilde{\boldsymbol{p}}_1, \ldots, \tilde{\boldsymbol{p}}_{N-1}\} \in \mathbb{R}^{N \times C}$ and $\mathbf{P} = \{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{N-1}\} \in \mathbb{R}^{N \times C}$ are the anchor label and predicted probability distribution of all nodes respectively.

Combination Loss: In each epoch, we employ both the Cross-Entropy loss in Eq. (18) and the Low-Entropy loss in Eq. (20) as the final combination loss. Considering the randomness of the sparse strategy and the additivity of entropy, we can perform multiple random sparse on the generated soft-weighted adjacency matrix

in each epoch, which increases the diversity to message passing. The loss function is expanded as follows:

$$\mathcal{L} = \frac{1}{S}\sum_{s=1}^{S}\left(CE\left(\mathbf{P}_l^{(s)}, \mathbf{Y}_l\right) + \lambda LE\left(\tilde{\mathbf{P}}, \mathbf{P}^{(s)}\right)\right), \tag{21}$$

where S is the number of random graph sparse, λ is a hyper-parameter controlling the balance of Cross-Entropy loss and Low-Entropy loss, and the anchor label $\tilde{\mathbf{P}}$ is updated as *sharpen* $\left(\frac{1}{S}\sum_{s=1}^{S}\mathbf{P}^{(s)}, \tau\right)$.

5. Experiments

In this section, we conduct experiments with the aim of answering the following research questions.

- *RQ1:* Does our proposed DPGNN outperform the state-of-the-art GNNs?
- *RQ2:* Is the design of each part indispensable for DPGNN?
- *RQ3:* Does our improved message-passing paradigm show superiority over the traditional message-passing paradigm?
- *RQ4:* Does differential control of node-specific message aggregation under different step lengths make sense?
- *RQ5:* Do the settings of hyper-parameters impact the performance of DPGNN?
- *RQ6:* Does our model expend a lot of training time?

In what follows, we foremost present the details of the experiment, including datasets, baselines, and implementation details. Then, we exhibit experiment results to reply the above six research questions.

5.1. Experiments setup

5.1.1. Evaluation datasets

We evaluate our proposed model on six real-world datasets (Citeseer [2], UAI2010 [58], ACM [59], BlogCatalog [60], Flickr [60], and CoraFull [61]) with different topological structures.

- Citeseer is a public research citation network, with nodes representing papers and edges representing citation links. Attributes of each node are bag-of-words representations of the relevant paper.
- UAI2010 contains 3067 nodes in 19 classes and it has been tested in GCN for community detection.
- ACM is extracted from the original ACM database, where the nodes represent papers and the edges indicate that there are co-authors between two papers. Node features are bag-of-words of paper keywords.
- BlogCatalog is a blogger's social network, which is collected from the BlogCatalog website, and the node features are constructed by the keywords of user profiles.
- Flickr is an interest social network, where nodes represent users and edges represent relationships between users.
- CoraFull is the larger version of well-known Cora dataset, in which nodes are papers and edges represent the citation between node pairs.

The statistics of the six datasets are shown in Table 1, from which we can see that Flickr and BlogDatalog are denser than the other datasets in topological structure. The same conclusion can be directly observed from Fig. 1, from which we can observe the six graph datasets have different topological structures.

Based on the previous analysis, the unfavorable message-passing structure is a disaster event for information aggregation. The high proportion of inter-class edges implies an imperfect message-passing structure. Given a graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of an arbitrary message space, the inter-class rate of edges is defined as follow:

$$\mathrm{ICR}_{\mathcal{G}} = \frac{\left|\left\{(i,j) \mid (i,j) \in \mathcal{E} \ \wedge \boldsymbol{y}_i \neq \boldsymbol{y}_j\right\}\right|}{|\mathcal{E}|}, \tag{22}$$

where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are the labels of node i and j. To further understand the original topology graph $\mathcal{G}^t$ and the constructed feature graph $\mathcal{G}^f$ of each dataset, we count their $\mathrm{ICR}_{\mathcal{G}^t}$ and $\mathrm{ICR}_{\mathcal{G}^f}$. The statistical results are shown in Fig. 6. All datasets have different inter-class rate in both spaces, which implies the diversity of experimental datasets. For example, the inter-class rate of Citeseer in topology space is lower than that in feature space. ACM has low inter-class rates in topology and feature space. Flickr has a extremely high inter-class rate in topology graph, while it has a quiet low inter-class rate in feature graph.

Table 1
The statistics of six datasets. The average degree is computed by $\frac{2M}{N}$.

Datasets	Nodes	Edges	Classes	Avg degree	Features
Citeseer	3327	4732	6	2.84	3703
UAI2010	3067	28311	19	18.46	4973
ACM	3025	13128	3	8.68	1870
BlogCatalog	5196	171743	6	66.11	8189
Flickr	7575	239738	9	63.30	12047
CoraFull	19793	65311	70	6.60	8710

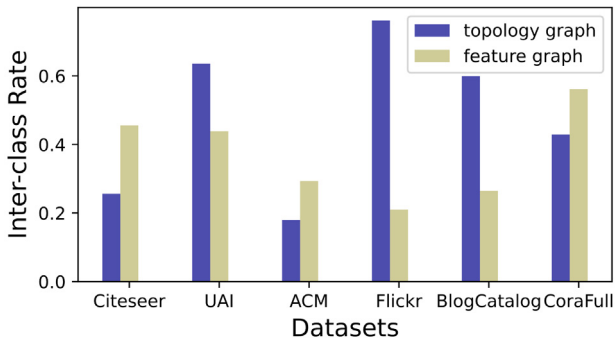


Fig. 6. Inter-class rate comparison in topology graph and feature graph of six graph datasets.

5.1.2. Baseline

We compare DPGNN with eight methods, including a classic neural network: i.e., Multilayer Perceptron(MLP), two topology-based shallow GNNs, i.e. GCN [2], SGC [19], four topology-based deep GNNs, i.e. APPNP [24], DAGNN [25], GCNII [26], GRAND [28], and two feature-based GNNs, i.e. AMGCN [38], SCRL [52].

5.1.3. Implementation details

Like most semi-supervised graph node classification tasks, we select 20 labeled nodes per class for training, 500 nodes for validation and 1000 nodes for testing. Specifically, in order for the impartial experimental comparison, we split the datasets are split in the same standard way as most topology-based GNNs, and make the data splits of UAI2010, ACM, BlogCatalog, Flickr equal to the feature-based model AMGCN. We conducted 100 runs with different random weight initialization for our proposed DPGNN, the results of which are averaged with 90% confidence level. We perform a grid search [62] to tune our associated hyper-parameters for each dataset and apply the same standard for reproducing unreported datasets on other methods. We run all experiments on the Pytorch platform with Intel(R) Xeon(R) Gold 6140 CPU and GeForce RTX 2080 Ti GPU. We use Adam optimizer [63] with learning rate 0.01 and weight decay 5e-4, and

Table 2
Results on six datasets in terms of node classification accuracy (in percent) and F1 score. (Bold: best; Underline: runner-up)

Dataset	Citeseer		UAI		ACM	
Metrics	ACC	F1	ACC	F1	ACC	F1
MLP	60.6 ±0.4	58.3 ±0.4	65.9 ±1.0	50.2 ±1.5	76.3 ±0.9	76.3 ±0.8
GCN	71.6 ±0.3	68.2 ±0.3	63.1 ±0.7	50.0 ±1.3	85.4 ±1.3	85.6 ±1.2
SGC	71.9 ±0.1	67.8 ±0.3	63.9 ±0.1	54.8 ±0.1	83.2 ±0.4	83.4 ±0.4
APPNP	72.4 ±0.5	68.6 ±0.6	68.6 ±0.9	53.5 ±1.6	88.4 ±0.7	88.4 ±0.7
DAGNN	73.3 ±0.6	68.6 ±0.5	64.9 ±1.0	49.0 ±1.3	88.2 ±0.7	88.2 ±0.7
GCNII	73.4 ±0.6	69.3 ±0.6	64.2 ±1.9	49.3 ±2.9	89.1 ±0.5	89.1 ±0.5
GRAND	75.4 ±0.4	70.2 ±0.3	67.9 ±0.5	54.7 ±0.7	88.0 ±0.6	88.1 ±0.5
AMGCN	73.1	68.4	70.1	55.6	90.4	90.4
SCRL	73.6	69.8	**72.9**	57.8	91.8	91.8
DPGNN (Ours)	**76.2 ±0.2**	**70.4 ±0.3**	71.1 ±0.8	**58.4 ±1.0**	**92.5 ±0.2**	**92.5 ±0.2**
Dataset	BlogCatalog		Flickr		CoraFull	
Metrics	ACC	F1	ACC	F1	ACC	F1
MLP	74.0 ±0.9	72.8 ±0.8	54.3 ±0.5	55.0 ±0.5	46.9 ±0.5	42.4 ±0.5
GCN	75.7 ±0.4	74.5 ±0.5	50.7 ±0.5	50.1 ±0.6	60.8 ±0.4	55.9 ±0.5
SGC	71.3 ±0.1	70.1 ±0.1	43.0 ±0.1	41.4 ±0.1	57.4 ±0.1	51.8 ±0.1
APPNP	82.0 ±1.0	80.8 ±1.1	58.3 ±0.8	57.5 ±0.9	57.2 ±0.5	52.5 ±0.5
DAGNN	82.3 ±3.4	81.3 ±3.7	62.8 ±1.4	62.4 ±1.6	60.5 ±0.6	55.4 ±0.8
GCNII	69.0 ±4.1	69.1 ±4.0	52.9 ±1.5	55.3 ±1.4	58.1 ±0.5	53.1 ±0.5
GRAND	88.8 ±0.9	87.9 ±1.0	68.3 ±0.5	67.5 ±0.5	60.0 ±0.3	53.0 ±0.4
AMGCN	82.0	81.4	75.3	74.6	58.9	54.7
SCRL	90.2	89.9	79.5	78.9	–	–
DPGNN (Ours)	**90.9 ±0.3**	**90.4 ±0.3**	**81.8 ±0.4**	**81.9 ±0.4**	**62.7 ±0.3**	**57.7 ±0.6**

apply early stopping strategy. Specifically, we range step length L from 2 to 8 for soft-weighted adjacency matrix learning in both topology space and feature space, and set $S \in \{1, 2, 3, 4, 5, 6\}$ for the random sparse operations.

5.2. Performance comparison (RQ1)

The results on six datasets are summarized in Table 2, where our DPGNN generally achieves the best performance on all datasets compared with all the latest state-of-the-art models. Notably, all performance results have a small standard deviation, demonstrating the weak dependence on initialization parameters and the superior stability of our model. For all datasets, whether the topological structure is favorable or not, our model can achieve a competitive performance. Because our model can capture structural neighborhood information and feature-related information simultaneously, which proves the versatility of our proposed model. Particularly, for the Flickr dataset with poor topological structure, our model improves accuracy by at least 2.9% and F1 score by at least 3.8% compared with other methods. And we discover that most topology-based GNNs perform poorly on Flickr. The main reason is that Flickr's topological structure will produce a lot of noise in information propagation. Furthermore, most latest state-of-the-art models except GCNII perform well on Blogcatalog whose topological structure is slightly better than Flickr. We suspect that is because APPNP, DAGNN and GRAND decouple the two operations of representation transformation and information propagation, which alleviates the poor performance of topology structure to a certain extent. However, SGC, which also decouples these two operations, performs poorly on Flickr, even worse than MLP. That is because there is only a linear classifier for representation learning in SGC, and Flickr has a particularly high initial feature dimension.

5.3. Ablation study for DPGNN (RQ2)

To prove the contribution of each component of DPGNN, we conducted an ablation study on the public dataset Citeseer. We used the average accuracy as the standard. The result is shown in Table 3. We find that: (1) when we adopt the average attention instead of using the learning parameter $\mathbf{W}_o$ without considering the node-specific message output discrepancy, the accuracy drops 1.4%; (2) when we do not use the soft-weighted adjacency matrix $\overline{\mathbf{M}}^f$ based on feature space or $\overline{\mathbf{M}}^t$ based on topology space for message passing and representation learning, the accuracy drops 1.2% and 6.3% respectively, (3) when we only perform the random graph sparse once (i.e. $S = 1$), the accuracy drops 1.2%; (4) when we do not apply random sparse strategy in the condition of $S = 1$ (i.e. directly adopt $\mathbf{M}^t$ and $\mathbf{M}^f$), the accuracy drops 3.7%; (5) when we do not introduce Low-Entropy loss in the condition of $S = 1$ (i.e. directly set $\lambda = 0$), the accuracy drops 3.9%. Above the aforementioned observation, each component of DPGNN contributes to the model, including classification accuracy and model stability. Particularly, case (1) illustrates the importance of focusing on the node-specific message output discrepancy. Case (2) shows that multi-space message interactions is complementary to graph representation learning, and our proposed message-passing paradigm can also achieve an ideal performance on Citeseer without introducing the original topology space. Case (3) proves that the diversity of message passing contributes to the improvement of graph representation learning ability during training. Based on the comparison between case (3) and case (4), we find that the random graph sparse strategy plays a significant role in model stability. And case (5) demonstrates the significance of introducing Low-Entropy loss.

Table 3
Abalation study on dataset citeseer.

Model	ACC
DPGNN	**76.2 ±0.2**
-w/o (1) parameter $\mathbf{W}_o$	75.1 ±0.3 (↓ 1.4%)
-w/o (2) feature space	75.3 ±0.5 (↓ 1.2%)
-w/o (2) topology space	71.4 ±0.4 (↓ 6.3%)
-w/o (3) multiple random sparse	75.4 ±0.3 (↓ 1.0%)
-w/o (4) random sparse strategy & $S = 1$	73.4 ±4.2 (↓ 3.7%)
-w/o (5) Low-Entropy loss & $S = 1$	73.2 ±0.4 (↓ 3.9%)

5.4. Superiority of the improved message-passing paradigm (RQ3)

In this part, we compare DPGNN with four specially designed models on all datasets to analyze the effects of the new message-passing paradigm in DPGNN.

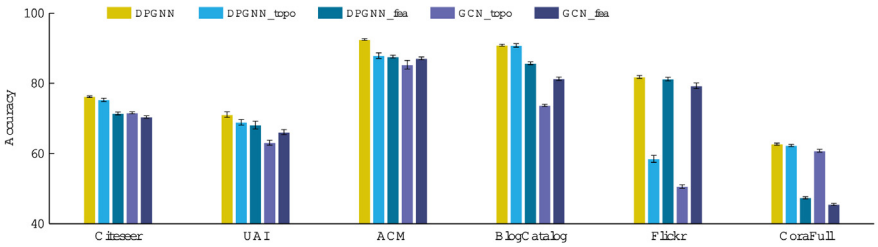


Fig. 7. The accuracy results of DPGNN compared with other specially designed models on six datasets.

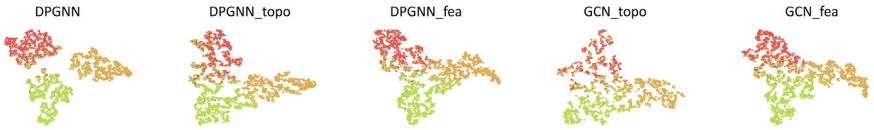


Fig. 8. The t-SNE visualization of node representations derived by DPGNN and four specially designed models on ACM dataset.

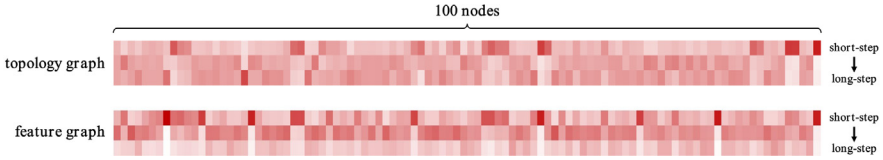


Fig. 9. The visualization of the output control coefficient (o_{vl} in Eq. (9)) learned in topology space and feature space for Citeseer dataset.

- **DPGNN_topo & DPGNN_fea**: DPGNN only based on topology space and DPGNN only based on feature space.
- **GCN_topo & GCN_fea**: GCN based on topology space and GCN based on feature space.

From the results in Fig. 7, we can draw the following conclusions: (1) The results of DPGNN are generally better than other four models with single perception, indicating the validity of multi-space message interaction in the new message-passing paradigm. This part corresponds to the dual-perception representation learning design of DPGNN. (2) The performance of DPGNN based single-perception network is superior to that of GCN based single-perception network (i.e. DPGNN_topo *vs* GCN_topo, DPGNN_fea *vs* GCN_fea), verifying the availability of multi-step message source in the new message-passing paradigm. This part is related to the soft-weighted adjacency matrix learning design of DPGNN. (3) Citeseer, UAI2010 and ACM can benefit from both message-passing spaces simultaneously, BlogCatalog and CoraFull benefit mainly from message passing in topology space, and Flickr benefits mainly from message passing in topology space. This shows the diversity of the evaluation baseline datasets, and our proposed DPGNN can achieve competitive performance in both favorable and unfavorable graph topology datasets, demonstrating the universality of our proposed model. To prove the effectiveness of our proposed model more intuitively, we draw the final node representations on ACM dataset in Eq. (16) by using t-SNE [64]. Fig. 8 shows the visualization of DPGNN performs best, in which the clearest distinct boundaries structures among different classes are exhibited compared to other methods.

5.5. Effects of node-specific message aggregation (RQ4)

To confirm the significance of node-specific message output design, we visualize the output control coefficient (o_{vl} in Eq. (9)) learned in topology space and feature space for Citeseer dataset. The step-length L^t in topology space and the step-length L^f in feature space for Citeseer are both set to 2. We selected 100 nodes to visualize. From Fig. 9, we can find that each node pays different attention to different-step message passing in topology space and feature space. Besides, the overall trend indicates that the attention value of the short-step messages is higher than that of the long-step messages. The aforementioned analysis implies that the differential control of node-specific message aggregation under different step lengths is meaningful.

5.6. Hyper-parameter sensitivity (RQ5)

In order to meticulously grasp the impact of hyper-parameters, we conduct experiments to investigate their influence. We firstly explore the influence of the count S of random graph sparse during training on model performance. Then because the step length of paths affects the breadth of message source, we conduct experiments to explore the comprehensive impact of step length L in both topology space and feature space. Lastly, we evaluate three aggregators for aggregating node representation generated in two message-passing spaces. Moreover, Table 4 reports the best hyperparameters of DPGNN we used for the results reported in Table 2.

5.6.1. Impact of S

To further verify the influence of hyper-parameter S for the random graph sparse operation, we evaluate our model by setting different S. The corresponding experimental results are shown in Fig. 10. As the S value increases, the message passing becomes more diverse. We can observe that on all datasets, the results when $S > 1$ are generally better than those when $S = 1$, which demonstrates the effectiveness of multiple random graph sparse operations. With the introduction of multiple random graph sparse operation, the density of the generated soft-weighted adjacency matrix is reduced and the diversity of message passing is increased, which enhances the message-passing paradigm robustness. Besides, we discover that for all data sets, a small S value enables the model to reach an ideal accuracy.

Table 4
Hyperparameters of DPGNN for six datasets.

Hyperparameter	Citeseer	UAI2010	ACM	BlogCatalog	Flickr	CoraFull
random graph-sparse times S	5	2	5	6	4	2
Step length L^t in topology space	2	3	2	3	2	3
Step length L^f in feature space	3	6	4	2	2	3
Aggregator	attention	attention	attention	mean-pooling	attention	attention

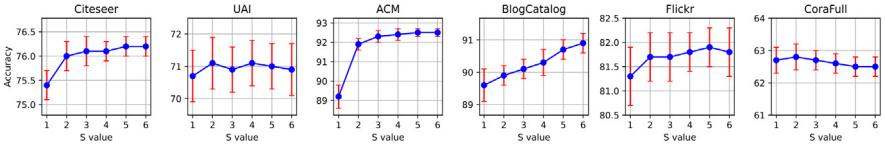


Fig. 10. Analysis of the influence of the count S of random graph sparse during training on model performance.

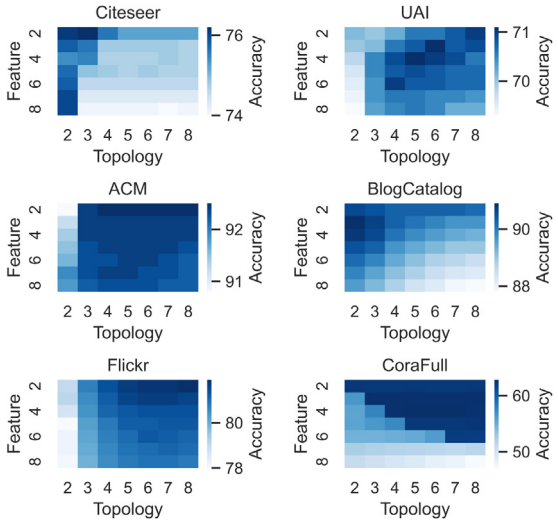


Fig. 11. Accuracy evaluation of the combinations of step length L^t in topology space and L^f in feature space. Both L^t and L^f are varied from 2 to 8.

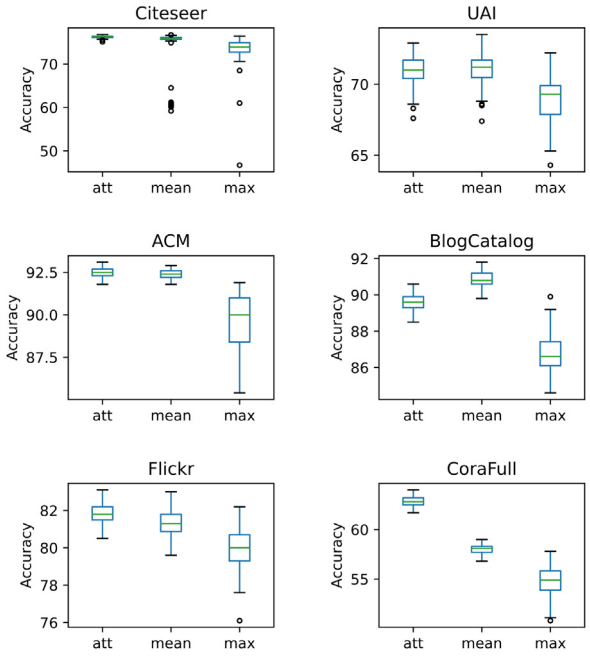


Fig. 12. Accuracy evaluation of three aggregators.

5.6.2. Impact of L

Step length settings in topology space and feature space determines the scope of message passing in the respective space, which can affect the performance of the model. Fig. 11 shows the performance under different combinations of L^t and L^f on six datasets. We find that different datasets have different sensitivities to them. For example, Citeseer prefers L^t set to 2, CoraFull prefers L^f set to 2, and ACM tends to L^t greater than 2.

5.6.3. Impact of aggregators

We evaluate three representation aggregators: Attention aggregator, Mean-pooling aggregator, and Max-pooling aggregator. As shown in Fig. 12, the Attention aggregator generally outperforms the other two aggregators except for BlogCatalog. The inferred reason is that Attention aggregator can aggregate node representations adaptively with its ability to distinguish the importance of node representation based on topology space and feature space. This also explains why our model can be adapted to datasets with different topological structures. Besides, Attention aggregator can also achieve a low small standard deviation, which means that Attention aggregator contributes to the stability of the model.

5.7. Complexity analysis (RQ6)

In our proposed DPGNN, the time complexity during training is mainly reflected in soft-weighted adjacency matrix learning and dual-perception representation learning. In the former part, the transfer coefficients can be precomputed, and because the 1×1 convolution operation can be parallelized across all nodes, the learning part can be computed with the time complexity $O(LN)$, where L is the step length and N is the number of nodes. In the latter dual-perception representation learning part, the time complexity of the weight-share layer and two weight-exclusive layers is $O(Nd_h(d+2C))$, and the time complexity of the message passing in topology space and feature space is $O(2pLMC)$, where d denotes the dimension of node feature, d_h is the hidden size, C is the number of classes, M is the number of edges, and $p \in (0, 1)$ is the probability in random graph sparse strategy. From the aforementioned analyses, the time complexity of our DPGNN is linear with the number of nodes and edges. To realize the time complexity of our model more intuitively, we compare the real running time of several models on Citeseer dataset. We calculate the average running time of each training epoch for all models, which is shown in Fig. 13. The size of each point corresponds to the running time of each model. We observe that DPGNN can achieve competitive performance in an acceptable running time.

Table A.5
Visualize the updated features of v_0 and u_0 under different message-passing paradigms.

Line	*Message-Passing Paradigm*	*The updated feature of v_0 in $\mathcal{G}_1$*	*The updated feature of u_0 in $\mathcal{G}_2$*
(1)	$\mathbf{m}_i^{(1)} = \alpha_i \mathbf{m}_i^{(0)} + \sum_{j \in \mathcal{N}_1(i)} \beta_j \mathbf{m}_j^{(0)}$	0.8 0.2 0	0.4 0.2 0.4
(2)	$\mathbf{m}_i^{(2)} = \alpha_i \mathbf{m}_i^{(1)} + \sum_{j \in \mathcal{N}_1(i)} \beta_j \mathbf{m}_j^{(1)}$	0.57 0.46 0.25	0.29 0.71 0.29
(3)	$\mathbf{m}_i = \sum_{\iota=0}^{2} \left(\sum_{j \in \mathcal{J}_\iota(i)} w_{\iota,ij} \mathbf{m}_j \right)$	0.68 0.33 0.13	0.34 0.45 0.34
(4)	$\mathbf{m}_i = \sum_{\iota=0}^{2} o_{i\iota} \left(\sum_{j \in \mathcal{J}_\iota(i)} w_{\iota,ij} \mathbf{m}_j \right)$	0.75 0.25 0.051	0.31 0.6 0.31

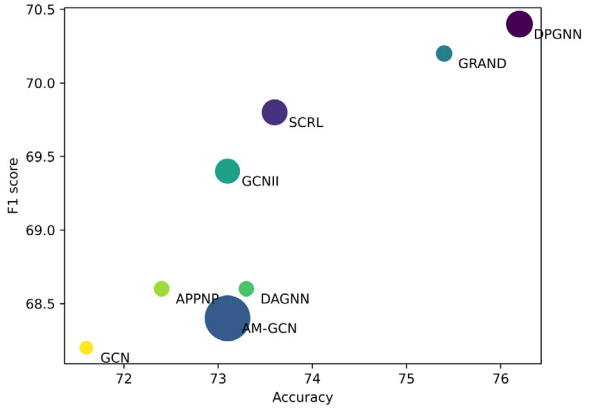


Fig. 13. A real running time comparison among several models.

6. Conclusion

In this paper, we formalize the existing message-passing paradigm and analyze its drawbacks including inflexibility of message source expansion, negligence of node-level message output discrepancy, and restriction of single message space. To address these issues, we present a novel message-passing paradigm and verify it by instantiating a Dual-Perception Graph Neural Network (DPGNN). The main ingredients contributing to the success of DPGNN are: (1) broadening the multi-step message sources without extra iteration; (2) considering the node-specific message output discrepancy; (3) adopting the multi-space message interaction. We quantify the differences in topology and feature space of six graph datasets and conduct extensive experiments on them. The experimental results demonstrate that our instantiated DPGNN outperforms related state-of-the-art GNNs, and we conduct analysis experiments to prove the superiority and versatility of our proposed message-passing paradigm. To our knowledge, we are the first to consider node-specific message passing in the GNNs. In the future, we will explore more efficient methods for the node-specific message passing, and extend DPGNN to more challenging tasks.

CRediT authorship contribution statement

Li Zhou: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Wenyu Chen:** Supervision, Investigation, Writing – review & editing. **Dingyi Zeng:** Writing – review & editing, Formal analysis. **Shaohuan Cheng:** Writing – review & editing. **Wanlong Liu:** Visualization. **Malu Zhang:** Writing – review & editing. **Hong Qu:** Supervision, Writing – review & editing, Funding acquisition.

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.

Acknowledgments

This work was supported by the National Science Foundation of China under Grant 61976043.

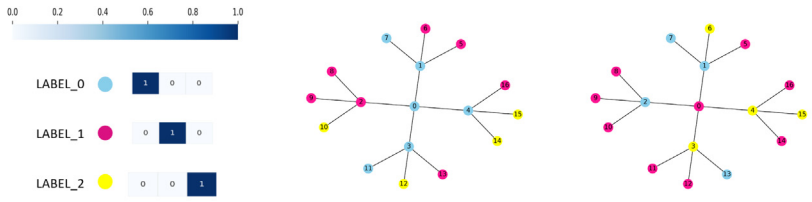
Appendix. Instance analysis

To clearly and intuitively verify that the existing message-passing paradigm ignores the amount of message output, we present two graphs with the same topology, denoted as $\mathcal{G}_1(\mathcal{V}, \mathcal{E})$ and $\mathcal{G}_2(\mathcal{U}, \mathcal{E})$ respectively, as shown in Fig. A.14. $\mathcal{G}_1$ and $\mathcal{G}_2$ are mainly composed of 3 types of nodes. The nodes of the same label are drawn by the same color. We assume that the initial features of nodes with the same type are identical, and the initial features are defined into a 3-dimensional one-hot vector. By comparing Figs. A.14(b) and A.14(c), we find that the central nodes v_0 in $\mathcal{G}_1$ and u_0 in $\mathcal{G}_2$ have the same neighborhood topology structure, but different neighborhood information distribution spaces. For v_0, its 1-hop neighborhood information is more important, while u_0 should concentrate more on its 2-hop neighborhood information.

In this experiment, we mainly focus on the central nodes v_0 and u_0 of $\mathcal{G}_1$ and $\mathcal{G}_2$, and detect their updated feature under different message-passing paradigms.[2] We use the message-passing method $\mathbf{m}_i^t = \sum_{j \in \mathcal{N}(i) \cup i} \frac{1}{\sqrt{deg(i)}\sqrt{deg(j)}} \mathbf{m}_j^{(t-1)}$ defined in GCN to materialize the amount of message retention α_i and the amount of message input β_j in the existing message-passing paradigm (Eq. (1)).

As shown in Table A.5, we visualize the updated features of v_0 and u_0 under different message-passing paradigms. In *Line* (1), we follow the existing message-passing paradigm, and get the updated feature of v_0 and u_0 after one iteration, which shows u_0 obtain an undesirable feature because its 1-hop neighbors brings noise. We further show the results after two iterations in *Line* (2). u_0 regains a desirable feature, while v_0 gains a non-ideal one. Therefore, the different neighborhood information space of each node is diverse, and the existing message-passing paradigm ignores node-level message output discrepancy, which is easier to bring noise information. To more intuitively demonstrate the advantages of our improved message-passing paradigm, we also visualize the related updated features. In *Line* (3), the message-passing paradigm focuses on multi-step message source, and in *Line* (4), our message-passing paradigm further considers the node-level message output discrepancy.

[2] Different from GNNs, we do not consider feature learning and only focus on message passing.



(a) The definition of initial node feature (b) $\mathcal{G}_1(\mathcal{V},\mathcal{E})$ (c) $\mathcal{G}_2(\mathcal{U},\mathcal{E})$

Fig. A.14. Two 4-regular graphs and their node initial feature definition. The initial features of the central node v_0 in $\mathcal{G}_1$ and u_0 in $\mathcal{G}_2$ are represented as $\mathbf{m}_{v_0}^{(0)} = [1, 0, 0]$ and $\mathbf{m}_{u_0}^{(0)} = [0, 1, 0]$ respectively.

References

[1] L. Li, X. Zhang, Y. Ma, C. Gao, J. Wang, Y. Yu, Z. Yuan, Q. Ma, A knowledge graph completion model based on contrastive learning and relation enhancement method, Knowl.-Based Syst. 256 (2022) 109889, http://dx.doi.org/10.1016/j.knosys.2022.109889.

[2] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.

[3] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.

[4] X. Ta, Z. Liu, X. Hu, L. Yu, L. Sun, B. Du, Adaptive spatio-temporal graph neural network for traffic forecasting, Knowl.-Based Syst. 242 (2022) 108199, http://dx.doi.org/10.1016/j.knosys.2022.108199.

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2021) 4–24, http://dx.doi.org/10.1109/TNNLS.2020.2978386.

[6] H. Chen, Z. Huang, Y. Xu, Z. Deng, F. Huang, P. He, Z. Li, Neighbor enhanced graph convolutional networks for node classification and recommendation, Knowl.-Based Syst. 246 (2022) 108594, http://dx.doi.org/10.1016/j.knosys.2022.108594.

[7] F. Yang, H. Zhang, S. Tao, Semi-supervised classification via full-graph attention neural networks, Neurocomputing 476 (2022) 63–74, http://dx.doi.org/10.1016/j.neucom.2021.12.077.

[8] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: Proceedings of the 34th International Conference on Machine Learning, Vol. 70, PMLR, 2017, pp. 1263–1272, URL https://proceedings.mlr.press/v70/gilmer17a.html.

[9] Y. Xie, C. Yao, M. Gong, C. Chen, A. Qin, Graph convolutional networks with multi-level coarsening for graph classification, Knowl.-Based Syst. 194 (2020) 105578, http://dx.doi.org/10.1016/j.knosys.2020.105578.

[10] W. Liu, Y. Zhang, J. Wang, Y. He, J. Caverlee, P.P.K. Chan, D.S. Yeung, P.-A. Heng, Item relationship graph neural networks for E-commerce, IEEE Trans. Neural Netw. Learn. Syst. 33 (9) (2022) 4785–4799, http://dx.doi.org/10.1109/TNNLS.2021.3060872.

[11] W. Li, L. Ni, J. Wang, C. Wang, Collaborative representation learning for nodes and relations via heterogeneous graph neural network, Knowl.-Based Syst. 255 (2022) 109673, http://dx.doi.org/10.1016/j.knosys.2022.109673.

[12] Z. Chen, X. Wang, C. Wang, J. Li, Explainable link prediction in knowledge hypergraphs, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2022, pp. 262–271, http://dx.doi.org/10.1145/3511808.3557316.

[13] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, L. Tian, Structured graph learning for clustering and semi-supervised classification, Pattern Recognit. 110 (2021) 107627, http://dx.doi.org/10.1016/j.patcog.2020.107627.

[14] H. Liao, J. Hu, T. Li, S. Du, B. Peng, Deep linear graph attention model for attributed graph clustering, Knowl.-Based Syst. 246 (2022) 108665, http://dx.doi.org/10.1016/j.knosys.2022.108665.

[15] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, Z. Guan, JKT: A joint graph convolutional network based deep knowledge tracing, Inform. Sci. 580 (2021) 510–523, http://dx.doi.org/10.1016/j.ins.2021.08.100.

[16] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, A. Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274, http://dx.doi.org/10.1016/j.knosys.2022.108274.

[17] Z. Li, X. Wang, J. Li, Q. Zhang, Deep attributed network representation learning of complex coupling and interaction, Knowl.-Based Syst. 212 (2021) 106618.

[18] Z. Li, X. Liu, X. Wang, P. Liu, Y. Shen, Transo: a knowledge-driven representation learning method with ontology information constraints, World Wide Web (2022) 1–23.

[19] 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.

[20] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 3538–3545.

[21] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, X. Sun, Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, (04) 2020, pp. 3438–3445, http://dx.doi.org/10.1609/aaai.v34i04.5747.

[22] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G.V. Steeg, A. Galstyan, MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, PMLR, 2019, pp. 21–29, URL https://proceedings.mlr.press/v97/abu-el-haija19a.html.

[23] L. Zhou, T. Wang, H. Qu, L. Huang, Y. Liu, A weighted GCN with logical adjacency matrix for relation extraction, in: ECAI 2020, IOS Press, 2020, pp. 2314–2321, http://dx.doi.org/10.3233/FAIA200360.

[24] J. Klicpera, A. Bojchevski, S. Günnemann, Predict then propagate: Graph neural networks meet personalized pagerank, in: International Conference on Learning Representations, 2019.

[25] M. Liu, H. Gao, S. Ji, Towards deeper graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 338–348, http://dx.doi.org/10.1145/3394486.3403076.

[26] M. Chen, Z. Wei, Z. Huang, B. Ding, Y. Li, Simple and deep graph convolutional networks, in: Proceedings of the 37th International Conference on Machine Learning, Vol. 119, PMLR, 2020, pp. 1725–1735, URL https://proceedings.mlr.press/v119/chen20v.html.

[27] Y. Rong, W. Huang, T. Xu, J. Huang, Dropedge: Towards deep graph convolutional networks on node classification, in: International Conference on Learning Representations, 2020.

[28] W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, Q. Yang, E. Kharlamov, J. Tang, Graph random neural networks for semi-supervised learning on graphs, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 22092–22103, URL https://proceedings.neurips.cc/paper/2020/file/fb4c835feb0a65cc39739320d7a51c02-Paper.pdf.

[29] V. Verma, M. Qu, K. Kawaguchi, A. Lamb, Y. Bengio, J. Kannala, J. Tang, GraphMix: Improved training of GNNs for semi-supervised learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, http://dx.doi.org/10.1609/aaai.v35i11.17203.

[30] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks? in: International Conference on Learning Representations, 2019.

[31] K. Oono, T. Suzuki, Graph neural networks exponentially lose expressive power for node classification, in: International Conference on Learning Representations, 2020.

[32] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, S. Jegelka, Representation learning on graphs with jumping knowledge networks, in: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 5453–5462, URL https://proceedings.mlr.press/v80/xu18c.html.

[33] A. Hagberg, P. Swart, D.S. Chult, Exploring network structure, dynamics, and function using networkx, Technical Report, 2008, URL https://www.osti.gov/biblio/960616.

[34] T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Softw. - Pract. Exp. 21 (11) (1991) 1129–1164, http://dx.doi.org/10.1002/spe.4380211102.

[35] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, D. Koutra, Beyond homophily in graph neural networks: Current limitations and effective designs, in: Advances in Neural Information Processing Systems, Vol. 33 (2020) 7793–7804, URL https://proceedings.neurips.cc/paper/2020/file/58ae23d878a47004366189884c2f8440-Paper.pdf.

[36] S. Pandit, D.H. Chau, S. Wang, C. Faloutsos, Netprobe: a fast and scalable system for fraud detection in online auction networks, in: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 201–210, http://dx.doi.org/10.1145/1242572.1242600.
[37] H. Pei, B. Wei, K.C.-C. Chang, Y. Lei, B. Yang, Geom-GCN: Geometric graph convolutional networks, in: International Conference on Learning Representations, 2019.
[38] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, J. Pei, Am-gcn: Adaptive multi-channel graph convolutional networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1243–1253, http://dx.doi.org/10.1145/3394486.3403177.
[39] L. Wu, D. Wang, K. Song, S. Feng, Y. Zhang, G. Yu, Dual-view hypergraph neural networks for attributed graph learning, Knowl.-Based Syst. 227 (2021) 107185, http://dx.doi.org/10.1016/j.knosys.2021.107185.
[40] L. Franceschi, M. Niepert, M. Pontil, X. He, Learning discrete structures for graph neural networks, in: Proceedings of the 36th International Conference on Machine Learning, Vol. 97, PMLR, 2019, pp. 1972–1982, URL https://proceedings.mlr.press/v97/franceschi19a.html.
[41] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, Semi-supervised learning with graph learning-convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 11313–11320, http://dx.doi.org/10.1109/CVPR.2019.01157.
[42] M. Matsugu, K. Mori, Y. Mitari, Y. Kaneda, Subject independent facial expression recognition with robust face detection using a convolutional neural network, Neural Netw. 16 (5) (2003) 555–559, http://dx.doi.org/10.1016/S0893-6080(03)00115-1.
[43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[44] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[45] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, in: International Conference on Learning Representations, 2014.
[46] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, Vol. 29, Curran Associates, Inc., 2016, URL https://proceedings.neurips.cc/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.pdf.
[47] C. Xu, Z. Guan, W. Zhao, H. Wu, Y. Niu, B. Ling, Adversarial incomplete multi-view clustering., in: International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 3933–3939, http://dx.doi.org/10.24963/ijcai.2019/546.
[48] S. Chang, J. Hu, T. Li, H. Wang, B. Peng, Multi-view clustering via deep concept factorization, Knowl.-Based Syst. 217 (2021) 106807, http://dx.doi.org/10.1016/j.knosys.2021.106807.
[49] H. Gao, Z. Wang, S. Ji, Large-scale learnable graph convolutional networks, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1416–1424, http://dx.doi.org/10.1145/3219819.3219947.
[50] L. Zhang, H. Lu, A feature-importance-aware and robust aggregator for GCN, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1813–1822, http://dx.doi.org/10.1145/3340531.3411983.
[51] S. Bandyopadhyay, A. Biswas, H. Kara, M. Murty, A multilayered informative random walk for attributed social network embedding, in: ECAI 2020, IOS Press, 2020, pp. 1738–1745, http://dx.doi.org/10.3233/FAIA200287.
[52] C. Liu, L. Wen, Z. Kang, G. Luo, L. Tian, Self-supervised consensus representation learning for attributed graph, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 2654–2662, http://dx.doi.org/10.1145/3474085.3475416.
[53] Y. Chen, L. Wu, M. Zaki, Iterative deep graph learning for graph neural networks: Better and robust node embeddings, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 19314–19326, URL https://proceedings.neurips.cc/paper/2020/file/e05c7ba4e087beea9410929698dc41a6-Paper.pdf.
[54] J. Zhao, X. Wang, C. Shi, B. Hu, G. Song, Y. Ye, Heterogeneous graph structure learning for graph neural networks, in: 35th AAAI Conference on Artificial Intelligence, 2021, http://dx.doi.org/10.1609/aaai.v35i5.16600.
[55] S. Yun, M. Jeong, R. Kim, J. Kang, H.J. Kim, Graph transformer networks, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, pp. 11983–11993, URL https://proceedings.neurips.cc/paper/2019/file/9d63484abb477c97640154d40595a3bb-Paper.pdf.
[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, 2017, URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
[57] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, in: Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, URL https://proceedings.neurips.cc/paper/2019/file/1cd138d0499a68f4bb72bee04bbec2d7-Paper.pdf.
[58] W. Wang, X. Liu, P. Jiao, X. Chen, D. Jin, A unified weakly supervised framework for community detection and semantic matching, in: Advances in Knowledge Discovery and Data Mining, Springer International Publishing, 2018, pp. 218–230.
[59] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: The World Wide Web Conference, 2019, pp. 2022–2032, http://dx.doi.org/10.1145/3308558.3313562.
[60] Z. Meng, S. Liang, H. Bao, X. Zhang, Co-embedding attributed networks, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 393–401, http://dx.doi.org/10.1145/3289600.3291015.
[61] A. Bojchevski, S. Günnemann, Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking, in: International Conference on Learning Representations, 2018, URL https://openreview.net/forum?id=r1ZdKJ-0W.
[62] Q. Tang, UltraOpt : Distributed asynchronous hyperparameter optimization better than HyperOpt, 2021, http://dx.doi.org/10.5281/zenodo.4430148.
[63] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, 2015.
[64] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).