Hierarchical Message-Passing Graph Neural Networks

Zhiqiang Zhong, Cheng-Te Li, and Jun Pang

Abstract—Graph Neural Networks (GNNs) have become a promising approach to machine learning with graphs. Since existing GNN models are based on flat message-passing mechanisms, two limitations need to be tackled. One is costly in encoding global information on the graph topology. The other is failing to model meso- and macro-level semantics hidden in the graph, such as the knowledge of institutes and research areas in an academic collaboration network. To deal with these two issues, we propose a novel *Hierarchical Message-Passing Graph Neural Networks* framework. The main idea is to generate a hierarchical structure that re-organises all nodes in a graph into multi-level clusters, along with intra- and inter-level edge connections. The derived hierarchy not only creates shortcuts connecting far-away nodes so that global information can be efficiently accessed via message passing, but also incorporates meso- and macro-level semantics into the learning of node embeddings. We present the first model to implement this framework, termed *Hierarchical Community-aware Graph Neural Network* (HC-GNN), based on hierarchical communities detected from the graph. Experiments conducted on eight datasets under transductive, inductive, and few-shot settings exhibit that HC-GNN can outperform state-of-the-art GNN models in network analysis tasks, including node classification, link prediction, and community detection.

Index Terms—Graph neural networks, hierarchical structure, representation learning, network communities

1 INTRODUCTION

GRAPHS are a ubiquitous data structure that models objects and their relationships, such as social networks, biological protein-protein networks, recommendation systems and etc. [14]. Learning node embeddings from a large graph has been proved as a useful approach for a wide variety of network analysis tasks, including link prediction [40], node and graph classification [38], [41], recommendation system [9] and community detection [7].

Graph Neural Network (GNN) is currently one of the most popular paradigms to learn and exploit node embeddings due to its effective capability of encoding both node features and graph topology in tasks of transductive, inductive, and few-shot settings. Existing GNN frameworks follow a similar methodology that a node embedding is obtained by a GNN layer, which aggregates the sampled neighbouring node's features, via non-linear transformation and aggregation functions. GNN layers are able to incorporate local information surrounded by each node. To make node embeddings encode features in the high-order neighbourhood, i.e., global information, two strategies are widely adopted. The first one is *increasing iterations* [35]: executing the GNN learning process up to sufficient times of iterations to make all nodes' information spread over the entire graph. Increasing learning iterations of GNNs can be considered as propagating node information towards the global in the form of a subtree-like cascade. The second is *stacking GNN layers*: stacking more GNN layers [36] so that features of nodes located in multiple hops away from a node can be captured. Nevertheless, existing GNN message-passing approaches to capture global information are inherently *flat*, indicating that information is propagated through *observed edges* in the graph. Flat message passing has two main limitations.

On one hand, both increasing iterations and stacking GNN layers cannot properly work in practical applications. *Increasing iterations of GNN learning is infeasible for semi-supervised and few-shot learning* [37]. The reason is that this strategy requires sufficient training samples so that messages on label information can spread globally in the entire graph. However, both semi-supervised and few-shot learning, which adopt a small set of training samples, do not meet such a requirement. As for the strategy of stacking multiple layers, it cannot effectively work because *GNN models are not robust to too many layers*. Recent studies have proved that as a special form of Laplacian smoothing, GNN models are not robust to multiple layers since they would become over-smoothing, i.e., node embeddings from different clusters become indistinguishable [5], [22]. Besides, too many GNN layers lead to vanishing gradient during training [21].

On the other hand, existing GNNs rely on only encoding features of topological neighbours. We argue that the hierarchical semantics behind the graph structure provides more useful information and should be incorporated into the learning of node embeddings. Taking the collaboration network in Figure 1(a) as an example, author nodes highlighted in light yellow come from the same institutes, and nodes filled with different colours indicate authors in various research areas. In order to generate the node

- *Zhiqiang Zhong is with the Faculty of Science, Technology and Medicine, University of Luxembourg, Esch-sur-Alzette, Luxembourg. E-mail: zhiqiang.zhong@uni.lu*
- *Cheng-Te Li is with the Institute of Data Science and the Department of Statistics, National Cheng Kung University, Tainan, Taiwan. E-mail: chengte@mail.ncku.edu.tw*
- *Jun Pang is with the Faculty of Science, Technology and Medicine, and the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette, Luxembourg. E-mail: jun.pang@uni.lu*

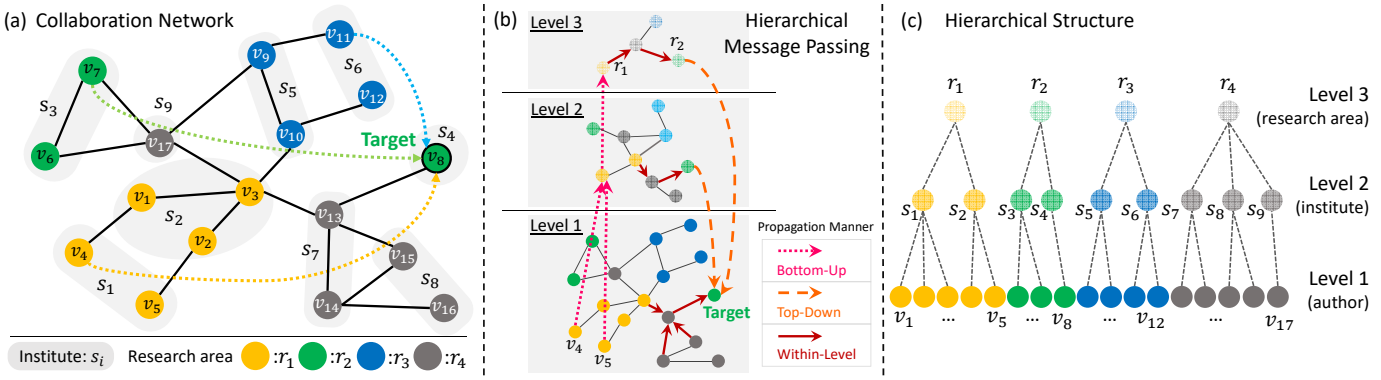


Fig. 1. Elaboration of the proposed hierarchical message passing: (a) a collaboration network, (b) an illustration of hierarchical message-passing mechanism based on (a) and (c), and (c) an example of the identified hierarchical structure.

embedding of a given author, existing GNNs mainly capture the co-author level information from neighbouring nodes. However, information hidden at *meso* and *macro* levels is neglected. In the example of Figure 1, meso-level information means authors belong to the same institutes and their connections to adjacent institutes. Macro-level information refers to authors belongs to the same research areas and their relationship with related research areas. Both meso-level and macro-level knowledge cannot be directly modelled through flat message passing via observed edges.

Existing studies deliver various attempts to model global information but still have some restrictions. Cluster-GCN [8] restricts the neighbourhood search to be within graph clusters so that the training efficiency of GCN can be improved. However, the model requires a sufficient number of training samples and neglects the potential interactions between clusters. Position-aware Graph Neural Networks (P-GNNs) [39] creates a different information aggregation mechanism that utilises sampled anchor nodes to impose topological position information into the learning of node embeddings. While global information can be captured by P-GNNs, the hierarchical semantics mentioned above is still overlooked. Besides, the anchor-set sampling process is time-costly for large graphs. P-GNNs also cannot work well under the inductive setting. Another popular approach to model global information is graph pooling, such as AttPool [17], and DiffPool [38]. However, what they target at is not node-level but graph-level representation learning for the task of graph classification. Graph pooling is also adopted for node-level graph neural networks, such as g-U-Nets [11], H-GCN [16]. Yet meso- and macro-level semantic information is still not considered. Although FDGNN [10] alternatively leverages a recurrent neural network to progressively propagate local information towards the global, it is still graph-level, and unclear whether it can be adapted to generate node-level embeddings.

This work presents a novel GNN framework, *Hierarchical Message-passing Graph Neural Networks*, to enhance the message-passing pipeline of GNNs. The ultimate goal is to make the GNN learning process aware of both global information and hierarchical semantics within the graph. We use Figure 1 to elaborate the proposed idea. We generate a hierarchical structure to endow a GNN layer to receive messages from all over the graph at different levels. It has two advantages over existing GNN models: (a) enabling a novel hierarchical message-passing mechanism that allows nodes to efficiently aggregate their features within the broader context of the graph structure; (b) generating node embeddings that encode not only micro-level information, i.e., graph topology and node features, but also meso- and macro-level information from hierarchical cluster structure, i.e., the relationships between nodes and clusters and the relationship between clusters. In detail, our framework can be organised into four phases:

- **Hierarchical structure generation**. To overcome long-distance obstacles in the process of GNN message-passing, we propose to use a hierarchical structure to reduce the size of graph G gradually, where nodes at each level k are integrated into different clusters $(s_1^{k+1}, s_2^{k+1}, \dots s_n^{k+1})$ at each level $k+1$.
- **k-level super graph construction**. In order to allow the message passing among generated same-level clusters, we construct a super graph G_k based on the connections between nodes at its lower level $k-1$.
- **Hierarchical message propagation**. With the generated hierarchical structure for a given graph, we develop a method that can propagate messages among nodes within the same level and across levels in both bottom-up and top-down manners.
- **Model learning**. Last, we leverage task-specific loss functions and a gradient descent procedure to train the model.

How to design a feasible hierarchical structure is crucial for *Hierarchical Message-passing Graph Neural Networks*, as it determines how messages can be passed through different levels and what kind of meso- and macro-level information to be encoded in node embeddings. In this paper, we consider (but not restricted to) network communities. Community, as a natural graph property, has been proved very useful for many graph mining tasks [32], [33]. Lots of community detection methods can generate hierarchical community structures. Here, we propose an implementation model for the proposed framework, *Hierarchical Community-aware Graph Neural Network* (HC-GNN). HC-GNN exploits a well-known hierarchical community detection method, i.e., the *Louvain* method [2] to build up the hierarchical structure, which is then used for the hierarchical message-passing

mechanism.

Extensive experiments are conducted on eight graph datasets to reveal the performance of HC-GNN on a variety of tasks, i.e., link prediction, node classification, and community detection, under both transductive and inductive settings. The results show that HC-GNN consistently outperforms the state-of-the-art approaches, with up to 11.4% AUC and 8.2% Micro-F1 improvements respectively for link prediction and node classification. In the few-shot learning setting, where only 5 samples of each label are used to train the model, HC-GNN achieves a significant performance improvement, up to 16.4%. We also deliver a few empirical insights: (a) the lowest level contributes most in node embeddings; (b) how to generate the hierarchical structure has a significant impact on the quality of node embeddings; (c) HC-GNN maintains an outstanding performance for graphs with different levels of sparsity.

Contributions. The contribution of this paper is four-fold:

1) We propose a novel *Hierarchical Message-passing Graph Neural Networks* framework, which allows nodes to conveniently capture global information and encode hierarchical semantics hidden behind the given graph.
2) We present the first implementation of our framework, namely HC-GNN[1], by detecting and utilising hierarchical community structures for message passing.
3) Experimental results show that HC-GNN significantly outperforms competing GNN methods on several prediction tasks under transductive, inductive, and few-shot settings.
4) Further empirical analysis is conducted to derive insights on the impact of the hierarchical structure and graph sparsity on HC-GNN.

The rest of this paper is organised as follows. We begin by briefly reviewing related work in Section 2. Then in Section 3, we introduce the preliminaries of this study. In Section 4, we introduce our proposed framework *Hierarchical Message-passing Graph Neural Networks* and its first implementation HC-GNN. Experimental results are shown in Section 5. Finally, we conclude the paper and discuss the future work in Section 6.

2 RELATED WORK

Flat message-passing GNNs. Most of the existing GNN models rely on the flat message-passing mechanism in the topology of the given graph. They perform graph convolution, directly aggregate node features from neighbours in the given graph, and stack multiple GNN layers to capture long-range node dependencies. For information aggregation, GCN adopts mean pooling [19] and GraphSAGE concatenates nodes' features with mean/max/LSTM pooled neighbouring information [15]. GAT aggregates neighbourhood information based on trainable attention weights [31]. GIN divides GNN layers into two parts, AGGREGATE and COMBINE, and performs optimisation according to the Weisfeiler-Lehman test to maximise the power of GNN under the neighbourhood aggregation framework [35].

1. Code and data are available at https://github.com/zhiqiangzhongddu/HC-GNN

TABLE 1
Model comparison in aspects of supervised training paradigm (**SUP**), node attributes (**NA**), community structure (**CS**), transductive inference (**TI**), inductive inference (**II**), global information (**GI**), and hierarchical semantics (**HS**).

	SUP	NA	CS	TI	II	GI	HS
GCN [19]	√	√		√	√		
GraphSAGE [15]	√	√		√	√		
GAT [31]	√	√		√	√		
GIN [35]	√	√		√	√		
P-GNNs [39]	√	√		√		√	
Cluster-GCN [8]	√	√		√	√		
H-GCN [16]	√	√		√	√		√
g-U-Nets [11]	√	√		√	√		√
HARP [6]		√		√			√
LouvainNE [1]		√	√	√			√
GraphRNA [18]	√	√		√			
HC-GNN	√	√	√	√	√	√	√

GraphRNA [18] presents graph recurrent networks to capture interactions between far-away nodes, but it cannot be applied to inductive learning settings.

Hierarchical representation GNNs. Inspired by convolutional neural networks (CNNs), in recent years some studies further generalise the pooling mechanism of CNNs to GCNs for hierarchical representation learning [11], [16], [17], [26], [38]. However, most of them, such as DIFFPOOL [38], SAGPOOL [17], ASAP and STRUCTPOOL [26], are designed for graph classification tasks, rather than learning node embeddings, hence they cannot be directly applied to node-level tasks, such as node classification and link prediction. H-GCN [16] introduces a graph coarsening layer to merge nodes with the same structural equivalence together as well as a refining layer to restore the original topological structure, in order to generate node embeddings. g-U-Nets [11] defines a gPool layer to adaptively form a smaller graph based on nodes' scalar projection values to incorporate global topological information. In spite of the success of H-GCN and g-U-Nets on producing graph-level embeddings, they cannot model the hierarchical semantics and their correlation behind the graph. HARP [6] and LouvainNE [1] are two unsupervised network embedding approaches that adopt a hierarchical structure, but they do not support the unsupervised training paradigm to optimise for specific tasks and they cannot be applied with inductive settings.

Global attentive GNNs. To generate node embeddings with global attention, two solutions were proposed recently. P-GNNs [39] incorporate a novel global information aggregation mechanism, which firstly samples the graph into several anchor-sets and learn a non-linear distance-weighted aggregation scheme over the anchor-sets based on the distance of a given target node to each anchor set. FDGNN [10] leverages a recurrent neural network to represent each input graph within a dynamic system to generate graph embedding. However, P-GNNs sacrifice the ability of existing GNNs on inductive node-wise tasks and the anchor-set sampling operation brings a high computational cost for large-size graphs. The versatility of FDGNN in node embeddings remains to be explored. Owing to limitations of existing GNNs, our proposed HC-GNN aims at efficiently encoding comprehensive global information

TABLE 2
Summary of main notations.

Notation	Description
G	an attributed graph
V, E	the set of nodes and edges on G, resp.
X	the matrix of node features
d	the pre-defined embedding dimension
$H \in \mathbb{R}^d$	the node embedding matrix
k	the number of hierarchy level
G_k	the super graph at level-k
s_n^k	the n-th cluster of G_k at level-k
$\mathcal{H}$	the set of constructed super graphs
$\mathcal{N}(v)$	the set of neighbour nodes of node v

and hierarchical semantics to generate node embeddings while maintaining all the benefits of flat message-passing GNNs.

Table 1 summarises the key advantages of the proposed HC-GNN, and compares it with a number of state-of-the-art methods published recently. We are the first to present the hierarchical message passing to model global information and hierarchical semantics. In addition, our HC-GNN can utilise the community structures, and be applied for both transductive and inductive inference.

3 PROBLEM STATEMENT

An attributed graph can be represented as $G = (V, E, X)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the node set, $E \subseteq V \times V$ denotes the set of edges, and $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{n \times \pi}$ is the feature matrix, in which each vector $x_i \in X$ is the feature vector associated with node v_i, and π is the dimension of input feature vector of each node.

Problem definition. Given an attributed graph $G = (V, E, X)$ and a pre-defined embedding dimension d, the goal is to learn a mapping function $f : G \rightarrow H$, where $H \in \mathbb{R}^d$ and each row $h_i \in H$ corresponds to the node v_i's embedding so that both graph topology G and node attributes X in the structural neighbourhood of v_i can be preserved as much as possible. The effectiveness of this mapping f is evaluated by applying H to different tasks, including node classification, link prediction, and community detection. Table 2 lists the mathematical notation used in the paper.

4 PROPOSED APPROACH

We propose a novel GNN framework, *Hierarchical Message-passing Graph Neural Networks*. The core idea is to use a hierarchical structure to enable a GNN layer to receiving both long-range messages and meso- and-macro-level semantics from different levels of the hierarchy. Figure 2 provides an overview of the proposed framework, which consists of four components. First, we create a hierarchical structure to coarsen the input graph G gradually. Nodes at each level k of the hierarchy are grouped into different clusters $(s_1^k, s_2^k, \ldots s_n^k)$. Second, we further organise level-k generated clusters into a super graph G_{k+1} at level $k+1$ based on the connections between nodes at level k, in order to enable message passing that encodes the interactions between generated clusters. Third, we develop three different propagation schemes to allow messages to be propagated among nodes within the same level and across different levels. In addition to the within-level propagation (Figure 1(b), i.e., the same as existing GNN neighbour aggregation, we propose two new message-passing pipelines, i.e., bottom-up propagation and top-down propagation shown in Figure 1(b). Such three pipelines form a closed loop to allow nodes to efficiently capture global information from all nodes in a graph. Different levels of the hierarchy can also provide meso- and macro-level semantics for node embeddings to encode. Last, after obtaining node embeddings, we use the task-specific loss function and a gradient descent procedure to train the model.

4.1 Hierarchical Message-passing GNNs

Hierarchical Message-passing Graph Neural Networks shown in Figure 2 consists of four phases: (1) hierarchical structure generation, (2) k-level super graph construction, (3) hierarchical message propagation, and (4) model learning.

I. Hierarchical structure generation. An attributed graph G can be naturally organised by cluster structures, in which densely inter-connected nodes are grouped. For example in Figure 1(a), authors $\{v_1, v_2, \ldots, v_{17}\}$ can be grouped into different clusters $\{s_1, s_2, \ldots, s_9\}$ based on their institutes. Institutes also can be grouped into higher-level clusters $\{r_1, \ldots, r_4\}$ according to research areas. Meanwhile, there is a relationship between nodes at different levels, as indicated by dashed lines in Figure 1(c). Hence, we can generate a hierarchical structure to depict both of the inter- and intra- relationships among authors, institute, and research areas. We will discuss how to implement the generation of hierarchical structure in Section 4.2.

II. k-Level super graph construction. The level-k super graph G_k is constructed based on level-$(k-1)$ super graph, where $k \geq 2$ and G_1 represents the original graph G. Given all nodes at level k–1, i.e., $s_1^{k-1}, s_2^{k-1}, \ldots, s_m^{k-1}$, we consider every node s_i^{k-1} belonging to the same cluster as a cluster node in the super graph G_k, and create an edge between cluster nodes s_i^{k-1} and s_j^{k-1} if there exist more than λ edges in G_{k-1} connecting elements in s_i^{k-1} and elements in s_j^{k-1}, where λ is a hyper-parameter and $\lambda = 1$ by default. In this way, we represent the hierarchical structure $\mathcal{H}$ as a list of graphs $\mathcal{H} = \{G_1, G_2, \ldots, G_K\}$. In which inter-level edges are created to depict the relationships between cluster nodes at levels k and k–1 if a level-(k–1) node has a corresponding cluster node at level k, as shown in Figure 1(c). We initialise the feature vectors of generated cluster nodes to be zero vectors with the same length as the original node feature vector x_i. Taking the collaboration network in Figure1 as an example. At the micro-level (level 1), we have authors and their co-authorship relations. At the meso-level (level 2), we organise authors according to their affiliations and also establish the relations between institutes. At the macro-level (level 3), institutes are further grouped together according to their research areas, and we have the relations among the research areas. In addition, inter-level links are also created to depict the relationships between authors and institutes, and between institutes and research areas.

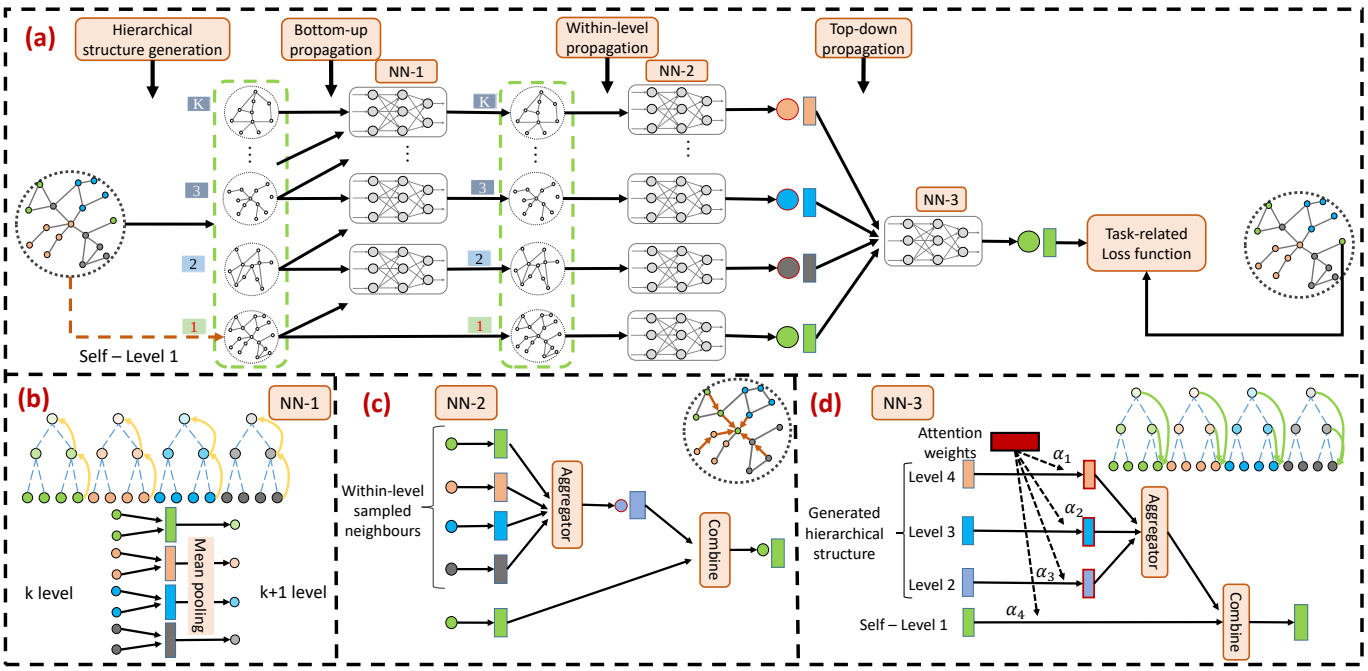


Fig. 2. (a) The architecture of *Hierarchical Message-passing Graph Neural Networks*: we first generate a hierarchical structure, in which each level is formed as a super graph, use the level-k graph to update nodes of level-$(k+1)$ graph (bottom-up propagation), apply the typical neighbour aggregation on each level's graph (within-level propagation), use the generated node embeddings from level $2 \leq k \leq K$ to update node embeddings at the level 1 (top-down propagation), and optimises the model via a task-specific loss. (b) NN-1: bottom-up propagation. (c) NN-2: within-level propagation. (d) NN-3: top-down propagation.

III. Hierarchical message propagation. The hierarchical message passing mechanism of a neural network layer ℓ consists of three steps.

1) *Bottom-up Propagation.* We perform bottom-up propagation, i.e., NN-1 in Figure 2(b), using node embeddings in G_{k-1} to update node embeddings in G_k ($k \geq 2$) in the hierarchy $\mathcal{H}$, as follows:

$$a_{s^k}^{\ell} = \frac{1}{\delta_{s^k}+1}\left(\sum_{s^{k-1}\in s^k} h_{s^{k-1}}^{\ell-1} + h_{s^k}^{\ell-1}\right), \quad (1)$$

where s^k is a cluster node in the graph G_k, and s^{k-1} is a node in G_{k-1} that belongs to cluster s^k in G_k. $h_{s^k}^{\ell-1}$ is the node embedding of s^k that generated by NN layer $\ell-1$ in graph G_k, δ_{s^k} is the number of nodes belonging to cluster s^k, and $a_{s^k}^{\ell}$ is the update embedding of s^k, which combines information from nodes at levels $k-1$ and k.

2) *Within-level Propagation.* We explore the typical AGGREGATE and COMBINE functions [19], [35] to propagate information in each level's graph $\{G_1, G_2, ..., G_K\}$, i.e., NN-2 in Figure 2(c). The aim is to aggregate neighbours' information and update within-level node embeddings. The AGGREGATE and COMBINE steps at level k are depicted as follows:

$$b_v^{\ell} = W \cdot MEAN\{a_u^{\ell}\}, \forall u \in \mathcal{N}(v) \cup \{v\}, \quad (2)$$

where W is a learnable matrix, $MEAN$ is an element-wise mean pooling, a_u^{ℓ} is the node embedding of u after bottom-up propagation at the ℓ-th NN layer, $\mathcal{N}(v)$ is a set of nodes adjacent to v at level k, and b_v^{ℓ} is the aggregated node embedding of v based on local neighbourhood information.

3) *Top-down Propagation.* The top-down propagation is illustrated by NN-3 in Figure 2(d). We use node embeddings in $G_2, \ldots, G_K$ to update the embeddings of original nodes in G_1. The contribution values of messages at different levels are different for different tasks. Hence, we adopt the graph attention mechanism [31] to adaptively learn the contribution weights of different levels during information integration, given by:

$$h_v^{\ell} = ReLU(W \cdot MEAN\{\alpha_{uv} b_u^{\ell}\}), \forall u \in \mathcal{C}(v) \cup \{v\}, \quad (3)$$

where α_{uv} is a trainable normalised attention coefficient between node v to cluster u or itself, $\mathcal{C}(v)$ denotes the set of different-level cluster nodes from level $2, \ldots K$ that node v belongs to ($|\mathcal{C}(v)| = K-1$), and ReLU is the activation function. We generate the output node embeddings of the last layer via:

$$z_v = \sigma(W \cdot MEAN\{\alpha_{uv} b_u^{\ell}\}), \forall u \in \mathcal{C}(v) \cup \{v\}, \quad (4)$$

where σ is the Euclidean normalisation function to reshape values into $[0, 1]$.

IV. Model learning. The proposed *Hierarchical Message-passing GNNs* could be trained in unsupervised, semi-supervised, or supervised settings. Here we only discuss the supervised setting used for node classification in our experiments. We define the loss function based on cross entropy, as follows:

$$\mathcal{L} = -\sum_{v \in V} y_v^{\top} \log\left(softmax(z_v)\right), \quad (5)$$

Algorithm 1: *Hierarchical Message-passing GNNs*

Input: graph $G = (V, E, X)$.
Output: node embeddings $\mathbf{z}_v$, cluster node embeddings $\mathbf{z}_c$.

1 $h_v^0 \leftarrow x_v$;
2 Generate hierarchical structure: $\{\mathrm{G}_i | i = 1, 2, \ldots, K\}, G_1 = G$;
3 $h_c^0 \leftarrow \{0\}^\pi$;
4 **for** $\ell \leftarrow \{1, 2, \ldots, L\}$ **do**
5 **for** $k \leftarrow 2$ **to** K **do**
6 **for** $s^k \in G_k$ **do**
7 $a_{s^k}^\ell = \frac{1}{\delta_{s^k}+1}(\sum_{s^{k-1} \in s^k} h_{s^{k-1}}^{\ell-1} + h_{s^k}^{\ell-1})$;
8 **end**
9 **end**
10 **for** $k \leftarrow 1$ **to** K **do**
11 **for** $v \in G_k$ **do**
12 $b_v^\ell = W \cdot MEAN\{a_u^\ell\}, \forall u \in \mathcal{N}(v) \cup v$;
13 **end**
14 **end**
15 **for** $v \in G_1$ **do**
16 $h_v^\ell = ReLU(W \cdot MEAN\{\alpha_{uv} b_u^\ell\}), \forall u \in \mathcal{C}(v) \cup v$;
17 $z_v = \sigma(W \cdot MEAN\{\alpha_{uv} b_u^\ell\}), \forall u \in \mathcal{C}(v) \cup v$;
18 **end**
19 **end**
20 $\mathbf{z}_v \in \mathbb{R}^d, \forall v \in V$;
21 $\mathbf{z}_c \in \mathbb{R}^d, \forall c \in \{V_2 \cup V_3 \cup \cdots \cup V_K\}$;

where y_v is a one-hot vector denoting the label of node v. We allow $\mathcal{L}$ to be customised for other task-specific objective function, e.g., the negative log likelihood loss [31].

We summarise the process of *Hierarchical Message-passing GNNs* in Algorithm 1. Given a graph G, we first generate the hierarchical structure and combine it with the original graph G, to obtain $\{G_i | i = 1, 2, \ldots, K\}$, where $G_1 = G$ (line 2). We initialise the newly generated cluster node embeddings as zero vectors with the same length π as the input node feature vector (line 3). For each node, including original and generated cluster nodes, in each NN layer, we perform three primary operations in order: (1) bottom-up propagation (line 5-9), (2) within-level propagation (line 10-14), and (3) top-down propagation (line 15-18). After getting the embedding vector of each node and cluster at all levels, we use node v's embedding and the loss function $\mathcal{L}$ in Eq. 5 to train the model.

4.2 Hierarchical Community-aware GNN

How to identify hierarchical clusters for the proposed *Hierarchical Message-passing GNNs* is the most crucial step as it determines how the information will be propagated within and between levels. We consider *hierarchical network communities* to construct the hierarchy. Network community has been proved useful for assisting typical network analysis tasks, including node classification [32], [33] and link prediction [27], [29]. Taking the algorithm efficiency into account, we adopt the well-known *Louvain* algorithm [2] to build the first implementation of *Hierarchical Message-passing GNNs*, termed as *Hierarchical Community-aware Graph Neural Network* (HC-GNN). The *Louvain* algorithm returns to us a hierarchical structure as described in Section 4.1, based on which we perform message propagation and generate node embeddings.

4.3 Complexity Analysis and Model Comparison

In this section, we analyse the model complexity and compare it with related models in the literature.

- For GNN models with a *flat* message passing mechanism, such as GCN and GAT, their computational complexity of one GCN layer is $\mathcal{O}(n^3)$ [16]. Assuming GCN model contains ℓ layers, the computational complexity of this model is $\mathcal{O}(\ell n^3)$. For GAT, except for the same convolutional operation as GCN, the additional masked attention over all nodes requires $\mathcal{O}(\ell n^2)$ computational complexity. Therefore, overall it takes $\mathcal{O}(\ell(n^3 + n^2))$ complexity.
- For the hierarchical representation model, g-U-Nets, its computational complexity is $\mathcal{O}(2\ell n^3)$, mainly because its unpooling operation introduces another $\mathcal{O}(\ell n^3)$ complexity, in addition to the convolutional operations as GCN.
- For HC-GNN, the hierarchical structure construction with *Louvain* algorithm has optimal $O(n \log c)$ computational complexity [30], where c is the average degree. The top-down propagation allows each node of G to receive k different messages from k levels with different weights, this introduces $\mathcal{O}(kn)$ computational complexity, where k is the number of levels, and we assume $k \ll n$. Altogether, the complexity of HC-GNN is $\mathcal{O}(\ell n^3 + n \log c + kn)$, which is more efficient than GAT and g-U-Nets.

5 EXPERIMENTS

We conduct extensive experiments to answer 5 research questions (RQ).

- **RQ1:** How does HC-GNN performs *vs.* state-of-the-art methods for node classification (**RQ1-1**), community detection **(RQ1-2)**, and link prediction (**RQ1-3**)?
- **RQ2:** Can HC-GNN leads to satisfying performance under settings of transductive, inductive, and few-shot learning?
- **RQ3:** How do different levels in the hierarchical structure contribute to the effectiveness of node embeddings?
- **RQ4:** How do various hierarchical structure generation methods affect the performance of HC-GNN?
- **RQ5:** Does HC-GNN survive from low sparsity of graphs?

5.1 Evaluation Setup

Datasets. We perform experiments on both synthetic and real-world datasets. We choose one pairwise task, i.e., link prediction, and two node-wise tasks, i.e., node prediction and community detection. For link prediction, we adopt three datasets:

- Grid [39]. 2D grid graph representing a 20×20 grid with $|V| = 400$ and no node features.

TABLE 3
Summary of dataset statistics. LP: Link Prediction, NC: Node Classification, CD: Community Detection, N.A. means a dataset does not contain node features.

Dataset	Task	#Nodes	#Edges	#Features	#Classes
Grid	LP	400	760	N.A.	N.A.
Cora	LP&NC	2,708	5,278	1,433	7
Power	LP	4,941	6,594	N.A.	N.A.
Citeseer	NC	3,312	4,660	3,703	6
Pubmed	NC	19,717	44,327	500	3
Emails	CD	799	10,182	N.A.	18
PPI	NC	56,658	818,435	50	121
Protein	NC	42,576	79,482	29	3

- Cora [28]. A citation network consists of 2,708 scientific publications and 5,429 links. Each publication is described by a 1,433 dimensional word vector as a node feature.
- Power [34]. An electrical grid of western US with 4,941 nodes and 6,594 edges, and no node features.

There is only Cora dataset has labelled nodes, therefore, we use the following datasets for node class prediction:

- Cora. The same above-mentioned Cora dataset which contains 7 classes of nodes. Each node is labelled with the class it belongs to.
- Citeseer [28]. A citation network consists of 3,312 scientific publications classified into one of 6 classes, and the dataset contains 4,660 edges. Each node has 3,703-dimensional node features.
- Pubmed [25]. A dataset consists of 19,717 scientific publications from PubMed database pertaining to diabetes classified into one of 3 classes. Each node is described by a TF/IDF weighted word vector from a dictionary which consists of 500 unique words.
- PPI [42]. 24 protein-protein interaction networks and nodes of each graph have 50 dimensional feature vector.
- Protein [3]. 1113 protein graphs and nodes of each graph have 29 dimensional feature vector. Each node is labelled with a functional role of the protein.

For node community detection, we use an email communication graph:

- Emails [20]. 7 real-world email communication graphs from SNAP with no node features. Each graph has 6 communities, and each node is labelled with the community it belongs to.

The data statistics is summarised in Table 3.

Experimental settings. We evaluate HC-GNN under the settings of transductive and inductive learning. For node classification, we additionally conduct experiments with the few-shot setting.

- *Transductive Learning*. The model is trained and tested on a given graph with a fixed node ordering and has to be re-trained whenever the node order is changed or some new nodes added to the graph. For link prediction, we follow the experimental settings of P-GNNs [39] to use 10% existing links and an equal number of non-existent links as validation and test sets. The remaining 80% existing links and a dual number of non-existent links are used as the training set. For node classification, we follow the semi-supervised experimental settings of GCN [19]: if there are enough nodes, for each class, we randomly sample 20 nodes for training, 500 nodes for validation, and 1000 nodes for testing. For the Emails dataset, we follow the supervised learning settings of GraphRNA [18] to randomly select 80% nodes as the training set, and use the two halves of remaining as the validation and test set, respectively. We report the test performance when the best validation performance is achieved. The experiments are repeated 10 times and average results are reported. Note that we use only node features with unique one-hot identifiers to differentiate different nodes if there are no given node features from the datasets, and use the original node features if they are available.
- *Inductive Learning*. This aims at examining a model's ability to transferring the learned knowledge from existing nodes to future ones that are newly connected to existing nodes in a graph. Hence, we hide the validation and testing graphs during training. We conduct the experiments for inductive learning using PPI and Protein datasets. We train models on 80% protein-protein interaction graphs to learn an embedding function f and apply it on the remaining 20% graphs to generate embedding of new-coming nodes. We further predict the function of these new-coming nodes as validation and test sets, respectively. Other settings follow the above transductive node classification.
- *Few-shot Learning*. As a recently proposed idea, few-shot learning [12] studies model performance with only a tiny number of training samples. Since the cost of collecting massive labelled datasets is high, having a few-shot learning model would be quite useful for practical applications. Few-shot learning can be also considered as an indicator to evaluate the robustness of a deep learning model. We perform few-shot node classification, in which only 5 samples of each class are used for training. The sampling strategies for testing and validation sets follow those in transductive learning.

Evaluation metrics. We adopt the area under the receive operating characteristic (AUC) to measure the performance of link prediction. For node classification, we use micro-average and macro-average F1 scores. The normalised mutual information (NMI) score is utilised for community detection.

Competing methods. To validate the effectiveness of the proposed HC-GNN, we compare it with 8 competing methods which include four GNN models with the flat message-passing mechanism, two hierarchical GNN models, one global attention GNN models and another state-of-the-art model.

- GCN [2] [19] is the first deep learning model which generalises the convolutional operation on graph data and introduces the semi-supervised train paradigm.
- GraphSAGE [3] [15] extends the convolutional operation of GCN to mean/ max/ LSTM convolutions and introduces a sampling strategy before employing convolutional operations on neighbour nodes.
- GAT [4] [31] employs trainable attention weight during message aggregation from neighbours, which makes the information received by each node different and provide interpretable results.
- GIN [5] [35] summarises previous existing GNN layers as two components, AGGREGATE and COMBINE, and models injective multiset functions for the neighbour aggregation.
- HARP [6] [6] is a hierarchical structure by various collapsing methods for unsupervised node representation learning.
- P-GNNs [7] [39] introduces anchor-set sampling to generate node embedding with global position-aware.
- g-U-Nets [8] [11] generalises the U-nets architecture of CNNs for graph data to get better node representation. It constructs a hierarchical structure with the help of pooling and unpooling operators.
- GraphRNA [9] [18] proposes using recurrent neural networks to capture the long-range node dependencies to assist GNN to obtain better node embedding.

Reproducibility. For the proposed HC-GNN, we exploit GCN for neighbour feature aggregation, and the number of HC-GNN layers is varied and denoted as 1L, 2L or 3L. In Section 5.3, HC-GNN adopts the number of layers leading to the best performance for model analysis i.e., 2L for the Cora dataset, 1L for the Citeseer and Pubmed datasets. For *Louvain* community detection, we use the implementation of a given package,[10] which does not require any hyper-parameters. Note that for the strong competitor, P-GNNs, since its embedding dimension is related to the number of nodes in a graph, we add a linear regression layer at the end of P-GNNs for node classification tasks to ensure its end-to-end structure as same as other models [18]. For fair comparison, all methods adopt the same embedding dimension (32), learning rate ($1e-3$) and the number of iterations (200). We use PyTorch Geometric[11] to implement all models that mentioned in this paper. More details are referred to the README file in our code file. [12]

2. https://github.com/tkipf/pygcn
3. https://github.com/williamleif/GraphSAGE
4. https://github.com/PetarV-/GAT
5. https://github.com/weihua916/powerful-gnns
6. https://github.com/GTmac/HARP
7. https://github.com/JiaxuanYou/P-GNN
8. https://github.com/HongyangGao/Graph-U-Nets
9. https://github.com/xhuang31/GraphRNA_KDD19
10. https://python-louvain.readthedocs.io/en/latest/api.html
11. https://pytorch-geometric.readthedocs.io/en/latest/

5.2 Experimental Results

Transductive node classification (RQ1-1&RQ2). We present the results of transductive node classification in Table 4. We can see that HC-GNN consistently outperforms all of the competing methods in the five datasets, and even the shallow HC-GNN model with only 1 layer leads to better results. The improvement is up to 8.2% in terms of Micro-F1. We think the outstanding performance of HC-GNN results from two aspects: (a) the hierarchical structure allows the model to capture long-range global information of graph topology, i.e., propagating messages from and to far-away nodes in the graph; and (b) the meso- and macro-level semantics reflected by the hierarchy is encoded through bottom-up, within-level, and top-down propagation. On the other hand, P-GNNs, HARP, and GraphRNA perform worse in semi-supervised node classification. The possible reason is they need more training samples, such as using 80% of existing nodes as training set, as described in their papers [18], [39], but we have only 20 nodes for training in the semi-supervised setting.

Inductive node classification (RQ1-1&RQ2). The results are reported in Table 5.[13] We can find that the proposed HC-GNN is still able to show some performance improvement over existing GNN models. But the improvement gain is not so significant and inconsistent in different layers of HC-GNN, compared to the results in transductive learning. The possible reason is that different graphs may have different hierarchical community structures. Nevertheless, the results lead to one observation: the effect of transferring hierarchical semantics between graphs for inductive node classification is rather limited.

Few-shot node classification (RQ1-1&RQ2). We exhibit the results in Table 6. The proposed HC-GNN demonstrates better performance in few-shot learning than all competing methods across three datasets. Such results indicate that the hierarchical message passing is able to transfer supervised information through inter- and intra-level propagations. In addition, hierarchical clusters further enlarge the influence range of supervised information from a small number of training samples. With effective and efficient pathways to broadcast information, HC-GNN is proven to be quite promising in few-shot learning.

Community detection (RQ1-2). The results of community detection conducted on the Emails dataset are also shown in Table 4. It can be seen that HC-GNN again outperforms all competing methods. We believe this is because the communities identified by Louvain are further exploited by learning their hierarchical interactions in HC-GNN. In other words, HC-GNN is able to reinforce the intra- and inter-community effect and encode it into node embeddings.

Link prediction (RQ1-3). Here, we motivate our idea by considering pairwise relation prediction between nodes. Suppose a pair of nodes u, v are labelled with label y, and

12. Code and data are available at https://github.com/zhiqiangzhongddu/HC-GNN

13. Since HARP, P-GNNs and GraphRNA cannot be applied in the inductive setting, we do not present their results in Table 5.

TABLE 4
Results in Micro-F1 and Macro-F1 for transductive semi-supervised node classification, and results in NMI for community detection (i.e., on the Emails data in the last column). Standard deviation errors are given.

	Cora		Citeseer		Pubmed		Emails
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	NMI
GCN	0.802 ± 0.019	0.786 ± 0.020	0.648 ± 0.019	0.612 ± 0.012	0.779 ± 0.027	0.777 ± 0.026	0.944 ± 0.010
GraphSAGE	0.805 ± 0.013	0.792 ± 0.009	0.650 ± 0.027	0.611 ± 0.020	0.768 ± 0.031	0.763 ± 0.030	0.925 ± 0.014
GAT	0.772 ± 0.019	0.761 ± 0.023	0.620 ± 0.024	0.594 ± 0.015	0.775 ± 0.036	0.770 ± 0.022	$\underline{0.947} \pm 0.009$
GIN	0.762 ± 0.020	0.759 ± 0.018	0.615 ± 0.023	0.591 ± 0.020	0.744 ± 0.036	0.733 ± 0.041	0.640 ± 0.047
P-GNNs	0.438 ± 0.044	0.431 ± 0.040	0.331 ± 0.019	0.314 ± 0.018	0.558 ± 0.033	0.551 ± 0.036	0.598 ± 0.020
HARP	0.363 ± 0.020	0.350 ± 0.021	0.343 ± 0.023	0.317 ± 0.017	0.441 ± 0.024	0.329 ± 0.019	0.371 ± 0.014
GraphRNA	0.354 ± 0.070	0.244 ± 0.040	0.352 ± 0.050	0.259 ± 0.047	0.476 ± 0.054	0.355 ± 0.089	0.434 ± 0.047
g-U-Nets	$\underline{0.805} \pm 0.017$	$\underline{0.796} \pm 0.018$	$\underline{0.673} \pm 0.015$	$\underline{0.628} \pm 0.012$	$\underline{0.782} \pm 0.018$	$\underline{0.781} \pm 0.019$	0.939 ± 0.015
HC-GNN-1L	$\mathbf{0.819} \pm 0.002$	$\mathbf{0.816} \pm 0.005$	$\mathbf{0.728} \pm 0.005$	$\mathbf{0.686} \pm 0.003$	$\mathbf{0.812} \pm 0.009$	$\mathbf{0.806} \pm 0.009$	$\mathbf{0.961} \pm 0.005$
HC-GNN-2L	$\mathbf{0.834} \pm 0.007$	$\mathbf{0.816} \pm 0.006$	$\mathbf{0.696} \pm 0.002$	$\mathbf{0.652} \pm 0.006$	$\mathbf{0.809} \pm 0.004$	$\mathbf{0.804} \pm 0.005$	$\mathbf{0.962} \pm 0.005$
HC-GNN-3L	$\mathbf{0.813} \pm 0.008$	$\mathbf{0.806} \pm 0.006$	$\mathbf{0.686} \pm 0.006$	$\mathbf{0.633} \pm 0.008$	$\mathbf{0.804} \pm 0.004$	0.780 ± 0.020	0.935 ± 0.014

TABLE 5
Micro-F1 results for inductive node classification. Standard deviation errors are given.

	PPI	Protein
GCN	0.444 ± 0.004	0.542 ± 0.018
GraphSAGE	0.409 ± 0.014	$\underline{0.637} \pm 0.018$
GAT	0.469 ± 0.062	0.608 ± 0.077
GIN	$\underline{0.571} \pm 0.008$	0.631 ± 0.016
g-U-Nets	0.433 ± 0.012	0.547 ± 0.011
HC-GNN-1L	0.48 ± 0.091	$\mathbf{0.638} \pm 0.027$
HC-GNN-2L	$\mathbf{0.584} \pm 0.087$	0.622 ± 0.031
HC-GNN-3L	$\mathbf{0.584} \pm 0.002$	0.582 ± 0.025

our goal is to predict y for unseen pairs. From the perspective of representation learning, we can solve the problem via learning an embedding function f that computes the node embedding z_v, where the objective is to maximise the likelihood of distribution $p(y|z_u, z_v)$. The results are summarised in Table 7, from which we draw three insights.

- First, the proposed HC-GNN leads to better performance, except for Cora-Feat, comparing to all competing methods. Since P-GNNs also model global position of nodes in the graph, its performance is better than the other methods on two datasets without node features (i.e., Grid and Power). Such results exhibit the importance of model global information in link prediction. Besides, while our HC-GNN achieves the best performance, with up to 11.4% AUC improvement, the results further validate the usefulness of hierarchical semantics.
- Second, on the Cora dataset with node features (Cora-Feat), HC-GNN is slightly worse than g-U-Nets, and the improvement gain of HC-GNN compared with other flat message-passing GNN models is relatively limited. We think the adaptive pooling in g-U-Nets better depicts pairwise features so that links can be better predicted. Nevertheless, our HC-GNN provides flexibility to incorporate adaptive pooling so that the most significant cluster-informed features can be extracted. We leave this point as a future extension.
- To examine whether HC-GNN cannot better work on link prediction without node features, we conduct the same experiment on Cora without using node features (i.e., Cora-NoAtt). HC-GNN leads to the best results. Such results indicate that when predicting links without node features, HC-GNN can better model graph topology and hierarchical semantics to capture the underlying relation between nodes.

5.3 Model Analysis

Contribution of different levels (RQ3). Since HC-GNN highly relies on the generated hierarchical structure, we aim to examine how different levels in the hierarchy contribute to the prediction. We report the performance of transductive semi-supervised node classification by varying the number of levels (from 1 to 4) adopted to construct the hierarchy. GCN is also selected for comparison because it considers no hierarchy, i.e., only within-level propagation in the original graph. The results are shown in Figure 3(a), in which 1H and 2H indicate only the first level and the first two levels are adopted, respectively. We can find that HC-GNN using more levels for hierarchy construction lead to better results. The flat message passing of GCN cannot work well. Such results provide strong evidence that GNNs can significantly benefit from the hierarchical message-passing mechanism. In addition, more hierarchical semantics can be encoded if more levels are adopted.

Influence of hierarchy generation approaches (RQ4). The proposed *Hierarchical Message-passing Graph Neural Networks* is implemented by HC-GNN based on the *Louvain* community detection algorithm. It is termed HC-GNN-*Louvain* in this paragraph. We aim to validate (A) whether the community information truly benefits the classification tasks, and (B) how different approaches to generate the hierarchical structure affect the performance.

- To answer (A), we construct a random hierarchical structure to generate randomised HC-GNN, termed HC-GNN-Random, in which hierarchical communities are detected by *Louvain* and nodes are randomly swapped among the same-level communities. In other words, the structure of hierarchy is maintained, but community memberships are perturbed. The results on semi-supervised node classification are exhibited in Figure 3(b). We can see that HC-GNN-Random works worse than GCN in Cora and Pudmed, and much worse than HC-GNN-*Louvain*. It implies that hierarchical communities generated from the graph topology truly lead to positive effect on information propagation.

TABLE 6
Micro-F1 results for few-shot node classification. Standard deviation errors are given.

	Cora	Citeseer	Pubmed
GCN	0.695 ± 0.049	0.561 ± 0.054	0.699 ± 0.059
GraphSAGE	$\underline{0.719} \pm 0.024$	0.559 ± 0.049	0.707 ± 0.051
GAT	0.630 ± 0.030	0.520 ± 0.054	0.664 ± 0.046
GIN	0.691 ± 0.038	0.509 ± 0.060	$\underline{0.714} \pm 0.036$
P-GNNs	0.316 ± 0.040	0.332 ± 0.011	0.547 ± 0.037
HARP	0.224 ± 0.033	0.260 ± 0.035	0.415 ± 0.039
GraphRNA	0.274 ± 0.063	0.206 ± 0.019	0.429 ± 0.042
g-U-Nets	0.706 ± 0.054	$\underline{0.567} \pm 0.044$	0.693 ± 0.036
HC-GNN-1L	0.681 ± 0.023	$\mathbf{0.639} \pm 0.019$	0.704 ± 0.043
HC-GNN-2L	$\mathbf{0.759} \pm 0.015$	$\mathbf{0.660} \pm 0.024$	$\mathbf{0.724} \pm 0.052$
HC-GNN-3L	$\mathbf{0.752} \pm 0.017$	$\mathbf{0.642} \pm 0.016$	$\mathbf{0.742} \pm 0.045$

TABLE 7
Results in AUC for link prediction. Cora-Feat means node features are used in the Cora dataset, and conversely, Cora-NoFeat means node features are not used. Standard deviation errors are given.

	Grid	Cora-Feat	Cora-NoFeat	Power
GCN	0.763 ± 0.036	0.869 ± 0.006	0.785 ± 0.007	0.624 ± 0.013
GraphSAGE	0.775 ± 0.018	0.870 ± 0.006	0.741 ± 0.017	0.569 ± 0.012
GAT	0.782 ± 0.028	0.874 ± 0.010	0.789 ± 0.012	0.621 ± 0.013
GIN	0.756 ± 0.025	0.862 ± 0.009	0.782 ± 0.010	0.620 ± 0.011
P-GNNs	$\underline{0.867} \pm 0.034$	0.818 ± 0.013	$\underline{0.792} \pm 0.012$	$\underline{0.704} \pm 0.006$
HARP	0.687 ± 0.021	0.837 ± 0.033	0.721 ± 0.017	0.529 ± 0.004
g-U-Nets	0.701 ± 0.032	$\underline{0.909} \pm 0.006$	0.772 ± 0.007	0.628 ± 0.024
HC-GNN-1L	0.823 ± 0.035	0.884 ± 0.006	$\mathbf{0.795} \pm 0.012$	0.682 ± 0.016
HC-GNN-2L	$\mathbf{0.913} \pm 0.011$	0.895 ± 0.007	$\mathbf{0.837} \pm 0.006$	$\mathbf{0.767} \pm 0.020$
HC-GNN-3L	$\mathbf{0.914} \pm 0.011$	0.891 ± 0.007	$\mathbf{0.839} \pm 0.004$	$\mathbf{0.784} \pm 0.017$

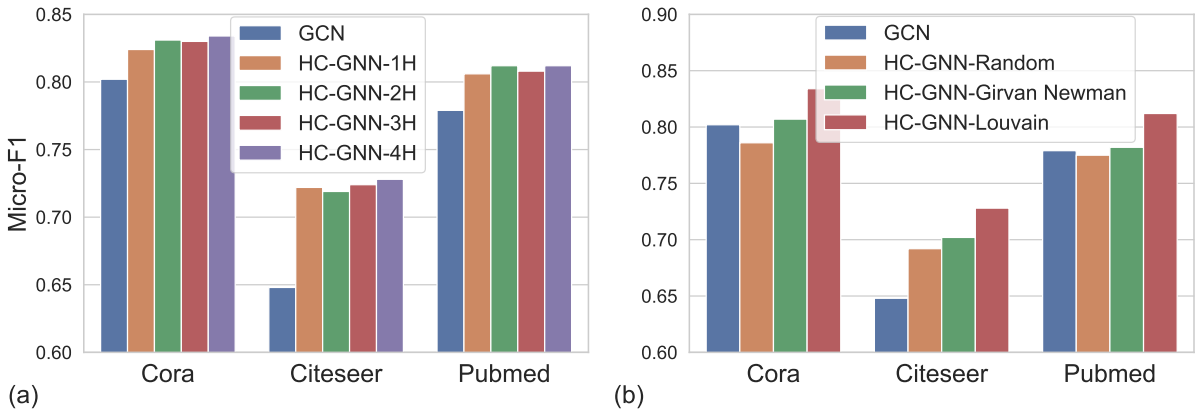


Fig. 3. Results in Micro-F1 for semi-supervised node classification using HC-GNN by varying: (a) the number of hierarchy levels adopted for message passing, and (b) the approaches to generate the hierarchical structure.

- To answer (B), we utilise *Girvan Newman* [13] to produce the hierarchical structure by following the same way as described in Section 4.1, and have a model named HC-GNN-*Girvan Newman*. The results shown in Figure 3(b). Although HC-GNN-*Girvan Newman* are not as effective as HC-GNN-*Louvain*, they still outperform GCN. Such a result indicates that the approaches to generate the hierarchical structure will influence the capability of HC-GNN. While HC-GNN-*Louvain* lead to promising performance, one can search for a proper hierarchical community detection method to obtain a more satisfying performance on different tasks.

Influence of graph sparsity (RQ5). Since community detection algorithms are sensitive to the sparsity of the graph [24], we aim at studying how HC-GNN perform under graphs with low sparsity values in the task of semi-supervised node classification. We consider two kinds of sparsity. One is graph sparsity by randomly removing a percentage of edges from all edges in the graph, i.e., $10\% - 50\%$. The other is node sparsity by randomly removing a percentage of edges incident to every node in the graph. The random removal of edges can be considered as that users hide partial connections due to privacy concern. The results for Cora and Citeseer are presented in Figure 4. HC-GNN significantly

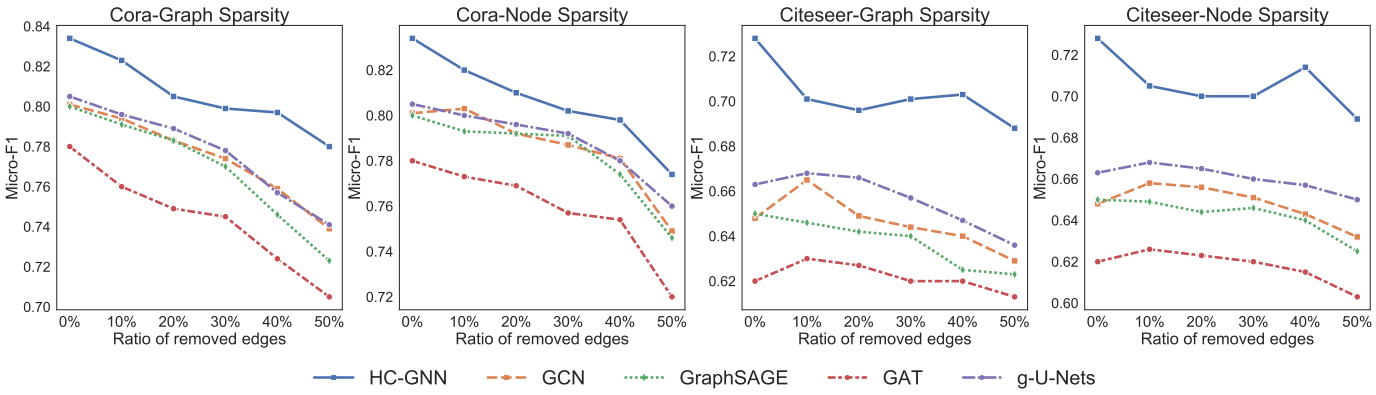


Fig. 4. Results on semi-supervised node classification in graphs by varying the percentage of removed edges.

outperforms the competing methods on both graph sparsity and node sparsity under different edge-removal percentages. Such results prove that even though communities are subject to sparse graphs, our HC-GNN are more robust than other GNN models.

6 CONCLUSION AND FUTURE WORK

This paper has presented a novel *Hierarchical Message-passing Graph Neural Networks* framework. It deals with two critical limitations of the flat message passing in existing GNN models, i.e., they are costly in capturing global information and infeasible in encoding meso- and macro-level graph semantics. We presented the first implementation of the proposed framework, HC-GNN, based on hierarchical communities detected from the graph. Extensive experiments conducted on eight graph datasets show that HC-GNN can consistently outperform state-of-the-art GNN models in three tasks, including node classification, link prediction, and community detection. We further showed that HC-GNN can lead to satisfying performance under settings of transductive, inductive, and few-shot learning. In fact, the proposed hierarchical message-passing GNN provides model flexibility. For instance, it allows different choices and customised designs of the hierarchical structure. In addition, recent advances in GNN techniques, such as hyperbolic space [4] and disentanglement [23], can be incorporated into our framework.

The proposed hierarchical message-passing GNNs provide a good starting point for exploiting graph hierarchy in GNN models. In the near future, we aim to incorporate the learning of the hierarchical structure into the model optimisation of GNNs such that a better hierarchy can be searched on-the-fly. It is also interesting to extend our framework for heterogeneous networks.

ACKNOWLEDGMENTS

This work is supported by the Luxembourg National Research Fund through grant PRIDE15/10621687/SPsquared. This work is also supported by Ministry of Science and Technology (MOST) of Taiwan under grants 109-2636-E-006-017 (MOST Young Scholar Fellowship) and 108-2218-E-006-036, and also by Academia Sinica under grant AS-TP-107-M05.

REFERENCES

[1] A. K. Bhowmick, K. Meneni, M. Danisch, J. Guillaume, and B. Mitra, "LouvainNE: Hierarchical Louvain method for high quality and scalable network embedding," in *Proceedings of the 2020 ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 2020, pp. 43–51.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[3] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, no. suppl_1, pp. i47–i56, 2005.

[4] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proceedings of the 2019 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2019, pp. 4869–4880.

[5] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2020.

[6] H. Chen, B. Perozzi, Y. Hu, and S. Skiena, "HARP: Hierarchical representation learning for networks," in *Proceedings of the 2018 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2018.

[7] Z. Chen, L. Li, and J. Bruna, "Supervised community detection with line graph neural networks," in *Proceedings of the 2019 International Conference on Learning Representations (ICLR)*, 2019.

[8] W. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 2019 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2019, pp. 257–266.

[9] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *Proceedings of the 2019 International Conference on World Wide Web (WWW)*. ACM, 2019, pp. 417–426.

[10] C. Gallicchio and A. Micheli, "Fast and deep graph neural networks," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2020.

[11] H. Gao and S. Ji, "Graph U-Nets," in *Proceedings of the 2019 International Conference on Machine Learning (ICML)*. JMLR, 2019.

[12] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *Proceedings of the 2018 International Conference on Learning Representations (ICLR)*, 2018.

[13] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, pp. 7821–7826, 2002.

[14] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, vol. 40, pp. 52–74, 2017.

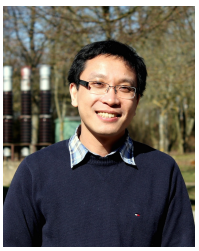
[15] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 2017 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2017, pp. 1025–1035.

[16] F. Hu, Y. Zhu, S. Wu, L. Wang, and T. Tan, "Hierarchical graph convolutional networks for semi-supervised node classification,"

in *Proceedings of the 2019 International Joint Conferences on Artificial Intelligence (IJCAI)*. IJCAI, 2019, pp. 4532–4539.
[17] J. Huang, Z. Li, N. Li, S. Liu, and G. Li, "Attpool: Towards hierarchical feature representation in graph convolutional networks via attention mechanism," in *Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 6480–6489.
[18] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proceedings of the 2019 ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2019, pp. 732–740.
[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 2017 International Conference on Learning Representations (ICLR)*, 2017.
[20] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.
[21] G. Li, M. Müller, G. Qian, I. C. Delgadillo, A. Abualshour, A. K. Thabet, and B. Ghanem, "Deepgcns: Can GCNs go as deep as CNNs?" in *Proceedings of the 2019 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 9267–9276.
[22] Q. Li, Z. Han, and X. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the 2018 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2018, pp. 3538–3545.
[23] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *Proceedings of the 2019 International Conference on Machine Learning (ICML)*. JMLR, 2019, pp. 4212–4221.
[24] R. R. Nadakuditi and M. E. J. Newman, "Graph spectra and the detectability of community structure in networks," *Physical Review Letters*, vol. 108, no. 18, p. 188701, 2012.
[25] G. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification," in *Proceedings of the 2012 Workshop on Mining and Learning with Graphs*, 2012, p. 8.
[26] E. Ranjan, S. Sanyal, and P. P. Talukdar, "ASAP: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2020.
[27] G. Rossetti, R. Guidotti, D. Pennacchioli, D. Pedreschi, and F. Giannotti, "Interaction prediction in dynamic networks exploiting community discovery," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. ACM, 2015, pp. 553–558.
[28] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–93, 2008.
[29] S. Soundarajan and J. E. Hopcroft, "Using community information to improve the precision of link prediction methods," in *Proceedings of the 2012 International Conference on World Wide Web (WWW)*. ACM, 2012, pp. 607–608.
[30] V. Traag, "Faster unfolding of communities: Speeding up the Louvain algorithm," *Physical Review E*, vol. 92, no. 3, p. 032801, 2015.
[31] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proceedings of the 2018 International Conference on Learning Representations (ICLR)*, 2018.
[32] J. Wang, J. Peng, and O. Liu, "An approach for hesitant node classification in overlapping community detection," in *Processings of the 2014 Pacific Asia Conference on Information Systems (PACIS)*, 2014, p. 47.
[33] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proceedings of the 2017 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2017.
[34] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, p. 440, 1998.
[35] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proceedings of the 2019 International Conference on Machine Learning (ICML)*. JMLR, 2019.
[36] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proceedings of the 2018 International Conference on Machine Learning (ICML)*. JMLR, 2018.
[37] H. Yao, C. Zhang, Y. Wei, M. Jiang, S. Wang, J. Huang, N. V. Chawla, and Z. Li, "Graph few-shot learning via knowledge transfer," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2020.
[38] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proceedings of the 2018 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2018, pp. 4805–4815.
[39] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," in *Proceedings of the 2019 International Conference on Machine Learning (ICML)*. JMLR, 2019.
[40] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Proceedings of the 2018 Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2018, pp. 5171–5181.
[41] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.
[42] M. Zitnik and J. Leskovec, "Predicting multicellular function through multi-layer tissue networks," *Bioinformatics*, vol. 33, pp. i190–i198, 2017.

Zhiqiang Zhong is currently pursuing his Ph.D. degree at Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg. His principal research interest is in graph neural networks and its applications in urban areas.

Cheng-Te Li is an Associate Professor at Institute of Data Science and Department of Statistics, National Cheng Kung University (NCKU), Tainan, Taiwan. He received my Ph.D. degree (2013) from Graduate Institute of Networking and Multimedia, National Taiwan University. Before joining NCKU, He was an Assistant Research Fellow (2014-2016) at CITI, Academia Sinica. Dr. Li's research targets at Machine Learning, Deep Learning, Data Mining, Social Networks and Social Media Analysis, Recommender Systems, and Natural Language Processing. He has a number of papers published at top conferences, including KDD, WWW, ICDM, CIKM, SIGIR, IJCAI, ACL, ICDM, and ACM-MM. He leads Networked Artificial Intelligence Laboratory (NetAI Lab) at NCKU.



Jun Pang is currently a Senior Researcher at Faculty of Science, Technology and Medicine & Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg. He received his Ph.D. degree from Vrije Universiteit Amsterdam in 2004. His principal research interest is in formal methods, social media mining, security and privacy.