The Journal of Supercomputing (2024) 80:2196–2222
https://doi.org/10.1007/s11227-023-05435-5

Dual graph neural network for overlapping community detection

Xiaohong Li[1] · Qixuan Peng[1] · Ruihong Li[1] · Huifang Ma[1]

Accepted: 24 May 2023 / Published online: 28 July 2023

Abstract

Community detection has long been designed to find communities with different structures in various networks. It is now widely believed that these communities often overlap with each other. However, due to the complexity and diversity of the network, it is often difficult to accurately identify the overlapping community structure in many real networks. Considering the above problem, we introduce a dual graph neural network for overlapping community detection (DGOCD) under the framework of the extended Bernoulli–Poisson. First, we build two graphs to model information of different orders between nodes, respectively, and use a set of GCNs as a backbone to learn semantic representations of the above graphs in parallel. Then we introduce the concept of topological potential matrix to aggregate the embedding representations of the two channel graphs. Moreover, for learning the affiliations between nodes and communities, we carry out network reconstruction based on the former information. Finally, the reconstructed network is sent into the GCN to get the final community division. Experimental results on real network datasets demonstrate that the proposed DGOCD consistently outperforms existing methods.

Keywords Overlapping community detection · Extended Bernoulli–Poisson model · Dual graph neural network · Topological potential matrix

Qixuan Peng has contributed equally to this work.

✉ Xiaohong Li
xiaohongli@nwnu.edu.cn

Qixuan Peng
2021222181@nwnu.edu.cn

Ruihong Li
2021212111@nwnu.edu.cn

Huifang Ma
mahuifang@yeah.net

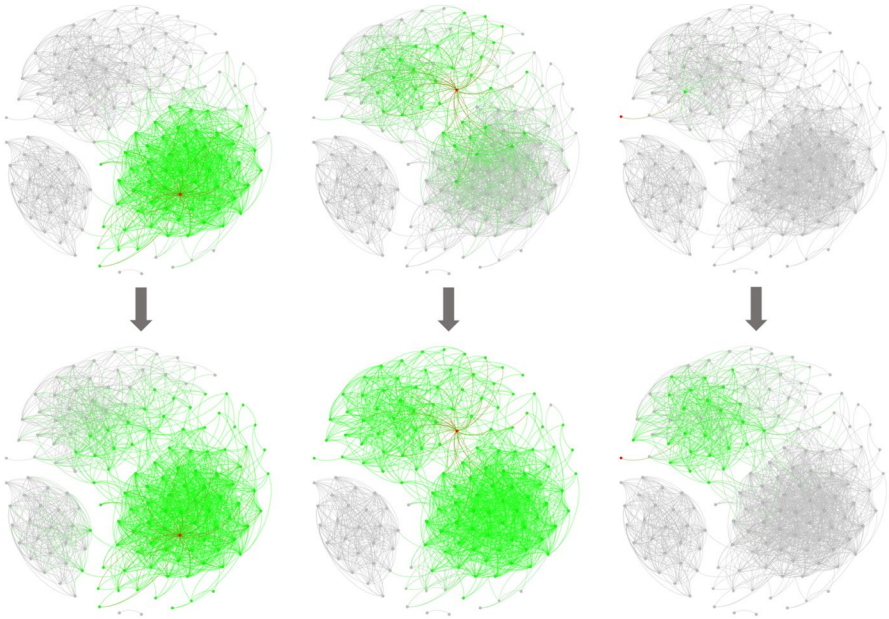
[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, People's Republic of China

1 Introduction

Community detection, as a classic graph partitioning problem, reveals the features and connections between different community members. A community is intuitively viewed as a group of nodes with more connections among its members than between its members and the remainder of the network [20, 29]. In networks with different functions such as social networks [3, 37] and biological networks [26], these groups of nodes (i.e., communities) are often interpreted as different meanings. Through community detection, we can analyze the importance, relevance and evolution of communities in detail and identify its research trends. It is now widely accepted that these communities often overlap with each other [41]. As we all know, graphs are powerful structures for modeling specific kinds of data because of their ability to combine object-level information with the underlying inter-object relationships [29]. With the remarkable development of graph deep learning techniques in recent years, graph neural network (GNNs) architectures have been widely used in community detection tasks by virtue of their advantages in processing high-dimensional network data [38].

In recent years, community detection work based on GNNs has been mainly carried out from two perspectives. One class of methods focuses on learning the vector representation of nodes in the graph and then uses traditional clustering algorithms, such as K-means, to achieve the final community detection. The downside of this class of methods is that they rely on a multi-step optimization procedure that does not allow the optimization of the objective via gradient descent end-to-end [23]. It cannot directly address node partitioning or the estimation of clusters within computational graphs; it is only suitable for detecting non-overlapping communities and has poor scalability. Another class of methods tries to build an end-to-end community detection model [19, 27, 30, 31], which requires only minimal hyperparameter tuning to achieve community segmentation with low complexity. Recently, some GNN methods that explicitly optimize community detection have attracted our interest. Among all the varieties of GNNs, graph convolutional neural networks (GCN) [36] are a simple but representative and salient one, which introduces the concept of convolution to GNNs that means to share weights for nodes within a layer. An easy and intuitive way to understand GCNs is to think of them as a message passing mechanism [10] where each node accepts information from its neighboring nodes to update its representation. This message-passing mechanism is highly effective in many scenarios. However, compared with traditional methods, such GCN-based community detection methods not only lead to an annoying over-smoothing problem, but also fail to clearly distinguish the differences between these observed edges and those not observed [6]. This further blocks us from mining fine-grained information in the graph.

Duan et al. [6] also describe the defects in the GCN-based approach, which argues that representations with the observed edges between nodes should be positively correlated. Yet, beyond these observed edges, there is a dark world that could provide diverse and useful information to the representation updates and help to overcome the over-smoothing problem at the same time. Many previous studies [19, 27, 31] have shown GCN methods that explicitly optimize community detection



(a) Central Node (b) Overlapping Node (c) Fringe Node

Fig. 1 Information transmission within two hops of different types of nodes

work best with two or three layer stacks. This is exactly in line with the three-degree influence principle proposed by Fowler et al. [7]. They believe that a node can not only influence its neighbors (one degree), but also that this influence can be passed on again. As long as they are within three degree of strong connections, there is the possibility of triggering behavior. If it exceeds more than three degree, the mutual influence between nodes will disappear. Of course, there is plenty of evidence of this. Topological potential [9] is a branch of traditional community detection methods, and the impact factor σ often controls the field range of nodes within two to three hops. As a classic model, topological potential can not only refine the attractiveness between each node pair, but also determine the community affiliations of nodes according to their positions in the inherent peak-valley structure of the topological potential field [43]. The concept obviously piqued our interest. Based on the above problems and observations, to dig out more useful information from the dark world, we try to integrate the topological potential with GCN to realize overlapping community detection. According to the experimental observation, whether it is a central node, an overlapping node or a fringe node, information can be transmitted in the relevant community under the principle of three-degree influence. Figure 1 shows the information transmission within two hops of different types of nodes.

Unlike many existing approaches, they update the representation of each node by treating adjacent nodes as positive samples and non-adjacent nodes as negative samples. In this paper, we propose DGOCD. By simulating the way of communication between entities in the real world (i.e., directly, without a third party), it uses a set of

GCNs to independently mine the interaction information between node and its first-order neighbors and second-order neighbors and dynamically adjusts the proportion of each part of information in the node embedding by topological potential matrix. The whole model is based on the extended Bernoulli–Poisson model (EBP), which implements the final overlapping community division through an end-to-end form. We show that DGOCD not only restores community affiliations more accurately, but also reconstructs the overlapping scale of the original communities as closely as possible. Specifically, we contribute:

- We introduce a new dual graph neural network model DGOCD, for overlapping community detection on graphs. The model inherits the core idea of the EBP model, breaks the limitation of the relationship between nodes and community members, and greatly improves the application range of this generative model.
- We separate the first-order information and second-order information in the graph and independently send them to a set of GCNs to learn the pairwise relationship between nodes and their first-order neighbors and second-order neighbors. Finally, the topological potential matrix is introduced to realize the fusion of multi-order information. This helps to promote more precise community segmentation.
- We evaluate our model on various real datasets and demonstrate that DGOCD outperforms state-of-the-art methods. Furthermore, the introduction of finer-grained information enables our model to find more accurate overlapping communities.

2 Related work

As the main method to understand the relationship between structure and inherent high-order information in the graph, community detection is of great significance in graph analysis and data mining. Community structure is an important area of research. It has received quite a bit of attention from the scientific community. Khan and Niazi [12] reviewed the evolution and latest progress of community detection in the field of traditional methods. At present, community detection is mainly divided into two categories: non-overlapping community detection, also called partitioning, whose purpose is to seek to assign each node to exactly one community. Overlapping community detection seeks a soft assignment of nodes into potentially multiple communities. Among them, the Girvan-Newman [21, 22] algorithm is a classic split-based community detection technique that separates communities from each other by removing intercluster edges in a network based on low similarity. Louvain [2] acts as a heuristic greedy algorithm that iteratively merges nodes based on the gain of modularity until the modularity no longer increases. The label propagation algorithm (LPA) [25] is known for its ability to find the community structure in the network within the linear complexity, but running LPA multiple times is likely to obtain different community structures, and the stability is worrying. MWLP [15] uses representative motifs to characterize the high-order features of the network on the basis of LPA. By integrating high-order and low-order structure information and

designing a new weighted network, a new voting strategy NaS is proposed to reduce the randomness in the label propagation process to obtain a more stable community structure. This further demonstrates the importance of higher-order information [13, 14]. Different from the previous methods, the NLA algorithm attempts to change the granularity of graph information integrated into the model [8, 24, 43], which uses the PageRank algorithm to evaluate the node quality and determine the community belonging of the node based on its position in the inherent peak-valley structure of the topological potential field. This is an interesting method of overlapping community detection. Similarly, AGM [39] and BIGCLAM [40] describe community detection as a variant of non-negative matrix factorization (NMF) to maximize the likelihood of the model. Almost all overlapping community detection methods before BIGCLAM make a hidden assumption that the density of connections in the overlapping part of the community is lower than that in the non-overlapping part of the community. This unnatural modeling assumption runs counter to the real world. EPM [44] and SNetOC [30] are based on the Bernoulli–Poisson model, and SNMF [33] uses non-negative matrix factorization.

Compared with the classic traditional methods, in recent years, a large number of community work have made breakthroughs under the rapid development of graph deep learning technology. Su et al. [29], Wu et al. [38], Liu et al. [16] summarize the community detection findings under the latest deep learning models. For example, some approaches attempted to focus on graph representation. In order to generate smooth features, AGE [5] introduces a Laplacian smoothing filter, adaptively selects positive and negative samples according to the similarity of node pairs and uses the final node representation for simple community division. CommDGI [42] applies k-means to node clustering to achieve joint optimization of the linear combination of DGI objectives, mutual information and modularity. DNE-SBP [28] uses semi-supervised stacked autoencoder (SAE) to learn a low-dimensional nonlinear graph representation and obtains the network embedding by reconstructing the adjacency matrix. SDNE [32] preserves the global and local structure of the network by introducing the concepts of first-order proximity and second-order proximity. All of the above methods are applied to non-overlapping communities, which aim to assign each node to a community. Although graph deep learning has achieved great success in non-overlapping community detection, applying it directly to overlapping communities does not achieve great results. Compared to non-overlapping community detection, overlapping community detection seeks to soft-assign nodes to potentially multiple communities. NOCD [27] inherits the community concept of BIGCLAM and is a rare GNN that explicitly optimizes overlapping community detection. It achieves effective community division by maximizing the possibility of Bernoulli–Poisson model. While the communities discovered by NOCD are able to recover the graph structure to the greatest extent, this approach limits the overlapping of community structures. DMoN [30], as an advanced non-overlapping community detection model, has also tried to apply it to overlapping communities due to its nature of maximizing modularity. At the time of writing, UCoDe [31] proposes a state-of-the-art method that can both detect overlapping and non-overlapping communities and can compete with NOCD, especially in overlapping community detection.

Table 1 The key mathematical notations and descriptions

Symbol	Description
$G = (V, E)$	An undirected simple graph
G^{1st}	First-order graph
G^{2nd}	Second-order graph
$\boldsymbol{X}$	Attribute matrix
$\widetilde{\boldsymbol{E}} = \{\boldsymbol{E}^{1st}, \boldsymbol{E}^{2nd}\}$	Multi-order edge set
$\widetilde{\boldsymbol{A}} = \{\boldsymbol{A}^{1st}, \boldsymbol{A}^{2nd}\}$	Multi-order adjacency matrix set
$\widetilde{\boldsymbol{Z}} = \{\boldsymbol{Z}^{1st}, \boldsymbol{Z}^{2nd}\}$	Multi-order node relationship matrix set
$\boldsymbol{Z}^{(1)}$	Node embedding representation matrix under the first set of GCNs
$\boldsymbol{Z}^{(2)}$	Node embedding representation matrix under the second set of GCNs
$\boldsymbol{F}$	Community affiliation matrix
$\boldsymbol{T}$	Topological potential matrix
$\mathcal{C}$	Penalty term

3 Preliminary

In this section, we briefly introduce the basic concepts of topological potential and Bernoulli–Poisson model. Table 1 summarizes the key mathematical symbols and descriptions used in this paper.

3.1 Topological potential

Field theory, as a classical mathematical model for describing the non-contact interactions between objects, can be used to describe the interaction and the association among network nodes. Each node is regarded as a field source, and these nodes interact with each other, forming a field called a topological potential field [35, 43]. Topological potential was first introduced and applied in the field of community detection by Li Deyi et al. [9]. They believe that the interaction between nodes has local characteristics, and the influence of each node will decay rapidly with the increase of network distance.

Given a graph $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_N\}$ represent its node set and $E = \{(v_i, v_j), v_i, v_j \in V\}$ represent its edge set, the topological potential of node v_i can be defined as follows:

$$\varphi(v_i) = \sum_{l=1}^{s} \left[m(v_l) \times \exp\left(-\left(\frac{d_{il}}{\sigma} \right)^2 \right) \right] \tag{1}$$

where v_l indicates the node within the influence scope of node v_i, s represents the total number of nodes within the influence scope, $1 \le s \le n-1$, $1 \le l \le s$, $m(v_l)$

refers to the mass of node v_l, d_{il} denotes the hops between nodes v_i and v_l, and impact factor σ used to control the influence scope of nodes where the maximum scope is $[3\sigma/\sqrt{2}]$ hops.

3.2 Bernoulli–Poisson model

Bernoulli–Poisson (BP) model [27, 30, 40, 44] is a graph generative model that allows overlapping communities. It inherits the notion of overlapping communities from Yang and Leskovec [40]. The model shows that the reason for the emergence of the community is from the close affiliation between the nodes. The higher the weight of a node's affiliation to the community, the more likely the node is to connect with other members of the community. The flexibility of the affiliation network allows us to model a wide range of network community structures.

For a graph $G = (V, E)$, given a bipartite community affiliation $B(V, C, M)$, we need to specify the process that generates the edges E of G. In that, V, C represent the node set and the community set, respectively; $(i, c) \in M$ denotes that node i belongs to community c. In addition, the graph assigns a non-negative weight F_{ic} to each affiliated edge between node $i \in V$ and community $c \in C$. ($F_{ic} = 0$ means no affiliation). Given the affiliation matrix $\boldsymbol{F} \in \mathbb{R}_{\geq 0}^{N \times C}$, we assume that each community c connects its member nodes depending on the value of $\boldsymbol{F}$. In particular, each community c connects its member nodes i, j independently with probability $p_c(i, j)$

$$p_c(i,j) = 1 - \exp\left(-\sum_{c=1}^{|c|} F_{ic} \cdot F_{jc}\right) \tag{2}$$

where $|c|$ denotes the number of common communities of a pair of nodes. As you can see, the more communities a pair of nodes has in common, the higher the probability of an edge. Further, according to the BP model, the graph is generated as follows. Given affiliations $\boldsymbol{F}$, adjacency matrix entries A_{ij} are sampled i.i.d as

$$A_{ij} \sim Bernoulli\left(1 - \exp\left(-\left(\boldsymbol{F}_i \cdot \boldsymbol{F}_j^T\right)\right)\right) \tag{3}$$

where $\boldsymbol{F}_i$ is the i's row of the matrix $\boldsymbol{F}$ and represents a weight vector of node i. Now that, the negative log-likelihood of the BP model is

$$\begin{aligned} L(\boldsymbol{F}) &= -\log P(\boldsymbol{A}|\boldsymbol{F}) \\ &= -\sum\nolimits_{(i,j)\in E} \log\left(1 - \exp\left(-\boldsymbol{F}_i \boldsymbol{F}_j^T\right)\right) \\ &\quad + \sum\nolimits_{(i,j)\notin E} \boldsymbol{F}_i \boldsymbol{F}_j^T \end{aligned} \tag{4}$$

Obviously, in this way, we actually model the level of participation of each node in a particular community.

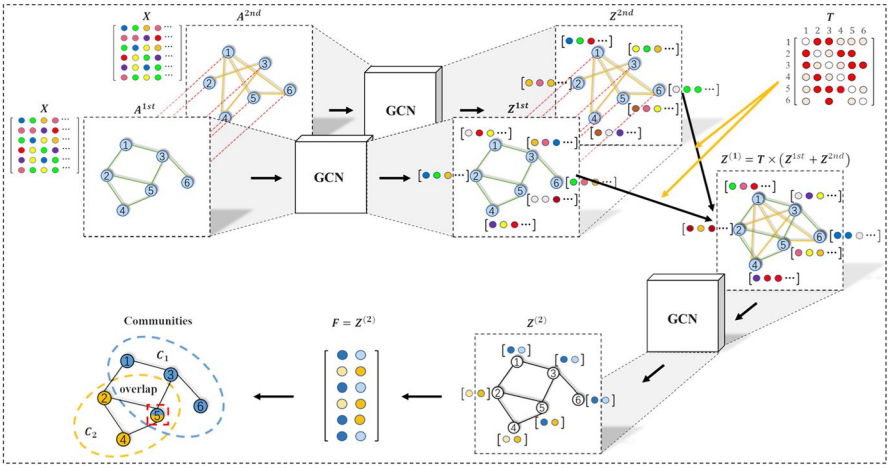


Fig. 2 Overlapping community detection model

4 Model description

In the following section, we present the proposed model for overlapping community detection in detail, starting with the graph neural network module. Then we describe the construction of dual graph neural network in detail. Finally, we introduce the generation of overlapping community affiliations. The entire overlapping community detection model is shown in Fig. 2.

4.1 Graph neural network module

The traditional BP model links the generation of edges with the affiliation matrix $\boldsymbol{F}$. The probability of node connection is proportional to the strength of shared members (the number of public communities of two nodes). In order to support a potentially infinite number of communities and more accurately mine the hidden connections in the original graph [44], we propose an extended Bernoulli–Poisson model (EBP). Based on the BP generative model, it further generalizes the probability density in the model, as shown in follows

$$A_{ij} \sim Bernoulli(1 - \exp(-(\boldsymbol{Z}_i \cdot \boldsymbol{Z}_j^T)) \tag{5}$$

In particular, $\boldsymbol{Z} \in \mathbb{R}^{N\times\bullet}$. Different from formula (3), we replace the community affiliation matrix $\boldsymbol{F}$ with the node relationship matrix $\boldsymbol{Z}$, which further generalizes the concept of affiliations and makes the generation of edges is no longer limited to the binary association between nodes and communities. In summary, the main idea of BP model is that if two nodes belong to more public communities (that is, the more similar the two nodes are), the greater the probability of generating edge between the two nodes. Similarly, the EBP model also inherits the idea, and its core goal is to seek more public dimensions instead of more public communities between node

pairs. The difference between $\boldsymbol{Z}$ and $\boldsymbol{F}$ can be briefly summarized as the change in the vertical dimension of the matrix. Frankly speaking, the core idea of BP model is to aggregate neighbor information to obtain the affiliations $\boldsymbol{F}$ between nodes and communities, where the vertical dimension of $\boldsymbol{F}$ is artificially set as the number of communities C. When C loses its realistic meaning, it only represents the dimension of a matrix. To this end, we can fully view the BP model as a way to obtain the node representation, which can be used to aggregate different types of neighbor information, respectively, in the subsequent work.

Existing works propose to perform inference in the BP model using maximum likelihood estimation with coordinate ascent [40] or Markov chain Monte Carlo [44]. [27] replaced the free variable model with a neural network model (GNN, MLP) for the first time and confirmed that the graph-based neural network architecture is indeed beneficial to the community detection task. With the advantage of graph convolutional neural network (GCN), we construct a graph neural network module. It can dynamically adjust the input of the whole module to obtain more accurate node relationship embeddings. The process of generating $\boldsymbol{Z}$ with GCN is as follows

$$\boldsymbol{Z} := GCN_{\theta}(\boldsymbol{A}, \boldsymbol{X}) \tag{6}$$

A ReLU nonlinearity is applied element-wise to the output layer to ensure non-negativity of $\boldsymbol{Z}$. As confirmed in [27], compared with the free variable model, the graph-based neural network structure can not only better retain the community structure, but also facilitate the generation of overlapping modules. This is extremely beneficial for overlapping community detection tasks. Unlike the traditional approaches of directly optimizing the node membership matrix $\boldsymbol{Z}$, we search for neural network parameters $\boldsymbol{\theta}$ that minimize the negative log-likelihood

$$\begin{aligned} L(\boldsymbol{Z}) &= -\log P(\boldsymbol{A}|\boldsymbol{Z}) \\ &= -\sum\nolimits_{(i,j)\in E} \log\left(1 - \exp\left(-\boldsymbol{Z}_i \boldsymbol{Z}_j^T\right)\right) \\ &\quad + \sum\nolimits_{(i,j)\notin E} \boldsymbol{Z}_i \boldsymbol{Z}_j^T \end{aligned} \tag{7}$$

$$\boldsymbol{\theta} = \arg\min_{\theta} L(GCN_{\theta}(\boldsymbol{A}, \boldsymbol{X}) \tag{8}$$

4.2 Overlapping community detection framework

To obtain a more accurate community division, we try to solve this problem from two different perspectives. On the one hand, as shown in the empirical observations, the proportion of first-order information in the graph is less. Therefore, we introduce a dual graph neural network, which enriches the original graph by introducing second-order information, and construct a topological potential matrix to realize the fusion and distinction of multi-type edge information. On the other hand, the extreme imbalance between edge information and non-edge information in the graph

will lead to large-scale bias in the contribution to the loss function. We solve this problem by introducing the standard technology in unbalanced classification.

4.2.1 Construction of dual graph neural network

Dual graph convolutional neural network (DGCN) was first proposed in (Zhuang and Ma [45]). Two convolutional neural networks are devised to embed the local-consistency-based and global-consistency-based knowledge, respectively. DGCN uses two simple feedforward networks in parallel, which differ only in the input graph structure information. Since then, a lot of work has gone into building different types of dual-graph frameworks to accommodate a wide variety of downstream tasks. For example, the framework proposed in (Alchihabi and Guo [1]) contains two GNN-based node prediction modules. The primary module uses the input graph structure to induce regular node embeddings and predictions with a regular GNN baseline, while the auxiliary module constructs a new graph structure through fine-grained spectral clusterings and learns new node embeddings and predictions. And our work also builds on this dual-graph framework.

As we have observed, in the topological potential, the range of influence controlled by σ is always kept at about two hops [9, 35]. Coincidentally, many GNN-based community detection methods [27, 31] also keep their optimal stacking layers around two layers. This further illustrates the importance of higher-order information, especially second-order information. Therefore, we construct a dual graph neural network module and design two convolutional neural networks to embed knowledge based on first-order neighbor information and second-order neighbor information, respectively. The introduction of topological potential matrix can better integrate and distinguish these two parts of knowledge.

Here are some definitions and an example. By extracting information from the original graph, we use first-order graph and second-order graph, respectively, to represent different types of relations between nodes. Usually, the first-order neighbor information of each node is stored in the adjacency matrix $\boldsymbol{A}(\boldsymbol{A} = \boldsymbol{A}^{1st})$. If there is a direct connection between nodes i and j, then $A_{ij}^{1st} = 1$; otherwise, $A_{ij}^{1st} = 0$ (the self-edge is not considered). Likewise, we introduce the concept of second-order adjacency matrix $\boldsymbol{A}^{2nd}$ to store the indirect connection between nodes; if there is a second-order shortest path between nodes i and j, then $A_{ij}^{2nd} = 1$; otherwise, $A_{ij}^{2nd} = 0$ (the round-trip path is not considered). By feeding the first-order graph G^{1st} and the second-order graph G^{2nd} into the dual graph neural network module, we could obtain the node relationship matrices $\boldsymbol{Z}^{1st}$ and $\boldsymbol{Z}^{2nd}$ corresponding to the first-order connected edges and the second-order connected edges. It contributes the model to learn information about different types of edges independently. According to the module, the two graphs are generated as follows in Fig. 3.

In the case of an undirected graph with six nodes, the first-order neighbors of node 2 include nodes 1,4 and 5, and the second-order neighbors of node 2 include only node 3. We connect node 2 with its first-order and second-order neighbors with green and orange edges, respectively. Traversing all nodes, first-order graph and second-order graph are generated. Going back to our model, given the multi-order

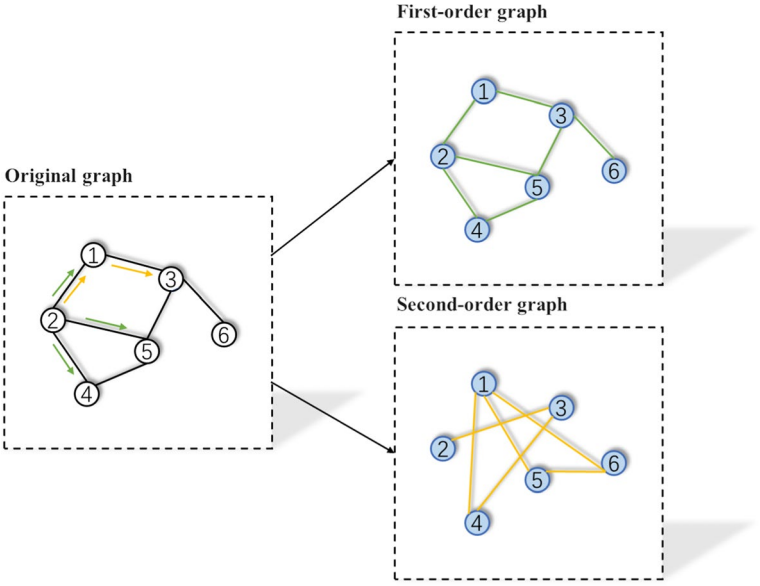


Fig. 3 The generation of first-order graph and second-order graph

node relationship matrix set $\widetilde{Z} = \{\boldsymbol{Z}^{1st}, \boldsymbol{Z}^{2nd}\}$, each matrix entity in the multi-order adjacency matrix set $\widetilde{A} = \{\boldsymbol{A}^{1st}, \boldsymbol{A}^{2nd}\}$ is sampled i.i.d as

$$A_{ij}^{1st} \sim Bernoulli\left(1 - \exp\left(-\left(\boldsymbol{Z}_i^{1st} \cdot (\boldsymbol{Z}_j^{1st})^T\right)\right)\right) \tag{9}$$

$$A_{ij}^{2nd} \sim Bernoulli\left(1 - \exp\left(-\left(\boldsymbol{Z}_i^{2nd} \cdot (\boldsymbol{Z}_j^{2nd})^T\right)\right)\right) \tag{10}$$

where A_{ij}^{1st} represents an element in the first-order adjacency matrix $\boldsymbol{A}^{1st}$; $\boldsymbol{Z}_i^{1st}$ is the row vector of relationship of node i (the i' row of the first-order node relationship matrix $\boldsymbol{Z}_i^{1st}$).

Intuitively, the more similar the nodes i and j are in the same dimension (i.e., the higher the dot product $(\boldsymbol{Z}_i^{1st} \cdot (\boldsymbol{Z}_j^{1st})^T)$ is), the more likely they are to be connected by an first-order edge. The second-order case is the same as above. Therefore, the node relationship matrices $\boldsymbol{Z}^{1st}$ and $\boldsymbol{Z}^{2nd}$ can be generated with the graph neural network module in Sect. 4.1, which are expressed as

$$\widetilde{Z} := GCN_{\widetilde{\theta}}(\widetilde{A}, \boldsymbol{X}) \tag{11}$$

where $\widetilde{\theta} = \{\theta^1, \theta^2\}$. For better integrate $\boldsymbol{Z}^{1st}$and $\boldsymbol{Z}^{2nd}$, we first need to distinguish the proportion of first-order information and second-order information in the whole network. Topological potential seems to help us. In the concept of topological potential, since $\varphi(v_i)$ denotes the sum of the potential of nodes within the influence scope of node v_i, we try to take it apart to get

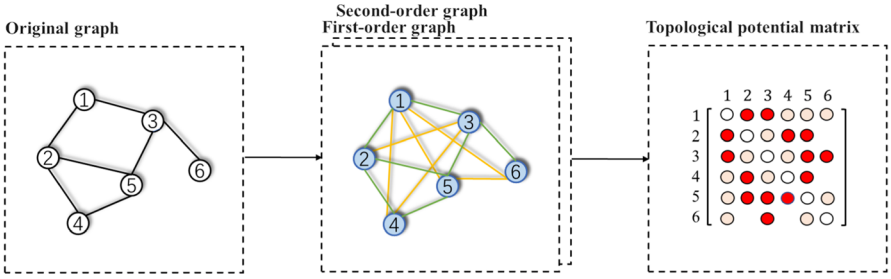


Fig. 4 The generation process of the $\boldsymbol{T}$ without the node mass

$$\varphi(\vec{e}_{ij}) = \left[m(v_j) \times \exp\left(-\left(\frac{d_{ij}}{\sigma} \right)^2 \right) \right] \tag{12}$$

$$\varphi(\vec{e}_{ji}) = \left[m(v_i) \times \exp\left(-(\frac{d_{ji}}{\sigma})^2 \right) \right] \tag{13}$$

In formula (12), if node $v_j \in v_l$, it means nodes v_j within the influence scope of node v_i, where $\varphi(\vec{e}_{ij})$ denotes the topological potential component of node v_i acting on node v_j. At this point, v_i represents the seed node. Otherwise, it means nodes v_j without the influence scope of node v_i, where $\varphi(\vec{e}_{ij}) = 0$. When the seed node is converted to v_j, $\varphi(\vec{e}_{ji})$ denotes the topological potential component of node v_j acting on node v_i in formula (13). Thus, we propose a new concept of the topological potential matrix $\boldsymbol{T} \in \mathbb{R}^{N \times N}$, which consists of topological potential components. Obviously, when we ignore the node mass, $\boldsymbol{T}$ is a symmetric matrix. Otherwise, $\boldsymbol{T}$ is an asymmetric matrix $\varphi(\vec{e}_{ij}) \neq \varphi(\vec{e}_{ji})$. Obviously, the topological potential value of a node is calculated without considering its mass. When the mass difference between two nodes is too large, the topological potential component centered on each node cannot express the relationship well. To better express this relationship between node v_i and node v_j, we average the two symmetric elements to form a new symmetric topological potential matrix, as shown in formula (14)

$$\varphi(e_{ij}) = \left[\frac{m(v_i) + m(v_j)}{2} \times \exp\left(-\left(\frac{d_{ij}}{\sigma} \right)^2 \right) \right] \tag{14}$$

where $T_{ij} = \varphi(e_{ij})$. In Fig. 4, we describe the generation process of the topological potential matrix without the node mass. The red dots describe the importance of first-order relationships (green edges), and the pink dots describe the importance of second-order relationships (orange edges). Finer grained information is fused on the basis of this when node mass is considered.

Then, $\boldsymbol{Z}^{1st}$ and $\boldsymbol{Z}^{2nd}$ are connected by the topological potential matrix $\boldsymbol{T}$

$$\boldsymbol{Z}^{(1)} = \boldsymbol{T} \times (\boldsymbol{Z}^{1st} + \boldsymbol{Z}^{2nd}) \tag{15}$$

Among them, $\boldsymbol{Z}^{(1)}$ represents the node embedding representation matrix incorporating multi-order edge information, which can more comprehensively and truly reflect the features in the original graph. Next, we feed the obtained node embedding representation matrix $\boldsymbol{Z}^{(1)}$ into a novel graph neural network module to get the final community affiliation $\boldsymbol{F}$

$$\boldsymbol{F} = \boldsymbol{Z}^{(2)} := GCN_{\theta}(\widetilde{\boldsymbol{A}}, \boldsymbol{Z}^{(1)}) \tag{16}$$

4.2.2 Generation of overlapping community affiliations

After fully integrating multi-order information, we realize the final overlapping community division under the framework of the dual graph neural network. The final negative log-likelihood function is as follows

$$\begin{aligned} L(\boldsymbol{Z}^{(d)}) &= -\log P\left(\boldsymbol{A}|\boldsymbol{Z}^{(d)}\right) \\ &= -\sum\nolimits_{(i,j)\in\widetilde{E}} \log\left(1-\exp\left(-\boldsymbol{Z}_i^{(d)}(\boldsymbol{Z}_j^{(d)})^T\right)\right) \\ &\quad + \sum\nolimits_{(i,j)\notin\widetilde{E}} \boldsymbol{Z}_i^{(d)}(\boldsymbol{Z}_j^{(d)})^T \end{aligned} \tag{17}$$

where $\widetilde{E} = \{E^{1st}, E^{2nd}\}$ represents a multi-order edge set corresponding to the first-order graph G^{1st} and the second-order graph G^{2nd}, respectively. Real-world network is usually sparse, which means non-edge parts contribute more to the loss of the whole model. As shown in Shchur and Günnemann [27], we try to counteract this by using standard techniques in imbalanced classification [11].

$$\begin{aligned} L(\boldsymbol{Z}^{(d)}) &= -E_{(i,j)\sim P_{\widetilde{E}}}\left[\log\left(\left(1-\exp\left(-\boldsymbol{Z}_i^{(d)}(\boldsymbol{Z}_j^{(d)})^T\right)\right)\right)\right] \\ &\quad + E_{(i,j)\sim P_N}\left[\boldsymbol{Z}_i^{(d)}\left(\boldsymbol{Z}_j^{(d)}\right)^T\right] \end{aligned} \tag{18}$$

As demonstrated in the model, $P_{\widetilde{E}}$ and P_N denote uniform distributions over edges and non-edges, respectively. This overlapping community detection framework consists of two sets of GCNs. In the first set, we independently sample edges in first-order and second-order graphs in an attempt to learn unique features for different types of edges. Then, we use the topological potential matrix to fuse multi-order node relationships and obtain the final community affiliation matrix $\boldsymbol{F}$ through the second set of GCNs. In order to alleviate the shock caused by high-order information, we add a penalty term $\mathcal{C}(\mathcal{C} = \exp(-T_{ij}))$ to the original loss function, to enable more accurate community detection. We search for neural network parameters $\boldsymbol{\theta}$ by minimize the negative log-likelihood function.

$$\begin{aligned} L(\boldsymbol{Z}^{(d)}) &= -E_{(i,j)\sim P_{\widetilde{E}}}\left[\log\left(\mathcal{C}\left(1-\exp\left(-\boldsymbol{Z}_i^{(d)}(\boldsymbol{Z}_j^{(d)})^T\right)\right)\right)\right] \\ &\quad + E_{(i,j)\sim P_N}\left[\boldsymbol{Z}_i^{(d)}(\boldsymbol{Z}_j^{(d)})^T\right] \end{aligned} \tag{19}$$

$$\theta = \arg\min_{\theta} L(GCN_{\theta}(\widetilde{\boldsymbol{A}}, \boldsymbol{X}) \tag{20}$$

5 Experiments

In this section, we detail the experimental setup and corresponding results. To illustrate the effectiveness of our proposed model, we compare it with some strong baselines for community detection tasks. We start with four research questions (RQ) to lead the experiments and the following discussions.

- (*RQ1*) Compared with the baseline models, does DGOCD achieve state-of-the-art performance on overlapping community detection tasks?
- (*RQ2*) Especially for the task of community detection, what are the advantages of dual graph neural network in our work compared to traditional graph convolution?
- (*RQ3*) Can the introduction of topological potential matrix well distinguish the importance of first-order information and second-order information? How do we balance the model complexity and efficiency?
- (*RQ4*) As an overlapping community detection method, can DGOCD accurately recover the overlapping scale and structure of the community?

5.1 Experimental setup

5.1.1 Dataset description

For a more comprehensive comparison, we select two widely used datasets. The detailed statistics of all datasets are summarized in Table 2.

- (*Facebook*) [17] is a collection of small (50–800 nodes) self-networks in the Facebook graph, and it is also a large graph dataset with reliable ground truth overlapping community information and node attributes (10K+ nodes). We selected six networks commonly used in Facebook to evaluate the community detection performance of the proposed model.
- (*Chemistry*, *Computer Science*, *Medicine*, *Engineering*) [34] are co-authorship networks, constructed from the Microsoft Academic Graph. Communities correspond to research areas in respective fields, and node attributes are based on keywords of the papers by each author.

5.1.2 Compared methods

We use six baseline methods, including BP-based and NMF-based models such as BigCLAM, SNetOC and SNMF, as well as GNN-based end-to-end models such as

Table 2 Dataset statistics. K stands for 1000. N represents the number of nodes, E denotes the number of edges, D indicates the dimension of the attribute matrix, and C expresses the number of communities

Dataset	Network type	N	E	D	C
Facebook-348	Social	224	3.2K	21	14
Facebook-414	Social	150	1.7K	16	7
Facebook-686	Social	168	1.6K	9	14
Facebook-698	Social	61	210	6	13
Facebook-1684	Social	786	14K	15	17
Facebook-1912	Social	747	30K	29	46
Chemistry	Co-authorship	35.4K	157.4K	4.9K	14
Computer science	Co-authorship	22.0K	96.8K	7.8K	18
Medicine	Co-authorship	63.3K	810.3K	5.5K	17
Engineering	Co-authorship	14.9K	49.3K	4.8K	16

NOCD, DMoN and UCoDe. It is worth noting that NOCD, DMoN and UCoDe are the state-of-the-art models proposed recently.

- (*BigCLAM*) [40] as a classic model is based on the BP model and uses coordinate ascent to learn $\boldsymbol{F}$.
- (*SNetOC*) [30] is also an overlapping community detection method based on the BP model. It uses Markov chains (MCMC) for inference to learn $\boldsymbol{F}$.
- (*SNMF*) [33] is an efficient method for finding overlapping communities based on non-negative matrix factorization (NMF).
- (*NOCD*) [27] combines a BP probabilistic model and a two-layer GCN to obtain the final community structure by minimizing the negative log-likelihood of BP and then setting a threshold to continuously identify and remove weak affiliations.
- (*DMoN*) [19] is originally proposed for non-overlapping communities, but it can be used for overlapping community detection, because its essence is to maximize modularity through GCN, and it is not associated with disjoint clusters, which can generate soft cluster assignments.
- (*UCoDe*) [31] is a GNN for community detection with a single contrastive loss by maximize modularity while explicitly allowing overlaps among communities. The model specifically designed for overlapping and non-overlapping community detection with an end-to-end model that requires minimal hyperparameter tuning.

5.1.3 Evaluation protocol

As mentioned in Shchur and Günnemann [27], commonly used metrics to quantify the agreement between true and detected communities, such as Jaccard and F1 scores, can give arbitrarily high scores for community assignments that are completely uninformative. As an alternative, we report overlapping normalized information (NMI). On the basis of retaining NMI, in order to be able to better verify the recovery of most models in the overlapping part of the community, we introduce the community overlap rate (COR) to evaluate the ability of the model to identify the correct cluster members from multiple perspectives.

- (*Overlapping NMI*) as defined in (McDaid et al. [18]), can correctly handle the degenerate case, as shown in formula (21)

$$NMI = \frac{I(X : Y)}{\max(H(X), H(Y))} \tag{21}$$

where X and Y refer to the class labels and predicted cluster indexes, respectively. Since the approximation is biased, we choose the average mutual information $I(X : Y)$. $H(\cdot)$ indicates the information entropy corresponding to the class labels. Eventually we will use the corresponding normalized mutual information to measure the accuracy of the experiment.

- (*COR*) mainly proposes that a large number of evaluation indicators for measuring overlapping communities focus on the accuracy of detection prediction values, but neglect to consider the recovery of "overlapping regions." It inherits ideas from (Chen et al. [4]), as shown in formula (22)

$$COR = \frac{\sum_{k=1}^{C} |C_k^p|}{\sum_{k=1}^{C} |C_k^{gt}|} \tag{22}$$

where $|C_k^p|$ and $|C_k^{gt}|$ represent the number of nodes in the predicted community and the ground-truth community in the graph, respectively. Subscript k represents the number of the community.

5.1.4 Implementation

The proposed model contains two sets of GCNs. The first set of GCNs is responsible for extracting and fusing multi-order information, and only one layer of graph convolution structure is set. We fix the size of the first layer to be 128 dimensions. The second set of GCNs is responsible for detecting community structure, and we keep the size of its output layer as the number of communities k. We independently sample and process the first-order and second-order graphs in the first set of GCNs. Instead of using all entries of $\widetilde{A}$ when computing the loss (Formula 18), we refer to the experiment in (Shchur and Günnemann [27]) and sample a mini-batch of edges and non-edges at each training epoch. The learning rate is set to 10^{-3}, and the threshold is kept at 0.5.

In the experiments in this paper, considering the high complexity of topological potential calculation, we temporarily choose to ignore the node mass difference in topological potential in large datasets and only discuss the influence of first-order and second-order information. One advantage of the BP model is that it allows to efficiently evaluate the loss and its gradients. By using a caching trick [40], the computational complexity of these operations can be reduced from $O(N^2)$ to $O(N + M)$. Shchur and Günnemann [27] did not use all entries of $\boldsymbol{A}$ in calculating losses, but sample a mini-batch of S edges and non-edges at each training epoch. Thus, approximately computing ∇L in $O(S)$. On this basis, we introduce the topological potential matrix without the node mass, and its calculation is independent of the number of nodes N. Therefore, the time complexity of the entire model is approximately $O(S)$. In conclusion, under the same time complexity, our model has better performance than NOCD. Clearly our model can achieve decent success in this case. We repeat each experiment 50 times, train the model for 500 epochs and report the average results for each run. For other methods, we experiment with the best settings in the original paper.

5.2 Result analysis

5.2.1 Experimental results and analysis

RQ1 The comparison results of our proposed model and baselines on two types of datasets are reported in Tables 3 and 4. Among them we provide results in two forms for each algorithm in NOCD, DMoN, UCoDe and DGOCD. The input to GCN in Model-X is node attributes; the input to Model-G is the concatenation of adjacency attributes and node attributes. The major findings from experimental results are summarized as follows:

- In the social network, we select six small and medium datasets to verify the effect of the model. As shown in Table 3, our model DGOCD achieves excellent performance in five out of six datasets, and the remaining one also performs well. We find that DGOCD-G performs a little better in most datasets, followed by NOCD-G. Overall, our method generally outperforms the second-best method by one to three percentage points in overlapping NMI values, and the performance on the Fb-1684 dataset is even more ferocious. But there are exceptions, in the Fb-348 dataset, NOCD-G performs the best, followed by our method. In the Fb-686 dataset, the performance of UCoDe-X can be described as a sudden rise, and the performance of Model-X is generally better than that of Model-G (which is contrary to the performance on most datasets).
- In the co-authorship network, we choose four large datasets to verify the effect of the model. As shown in Table 4, our model DGOCD is consistently better than all baselines. We find that DGOCD performs better and more stable on large datasets than the small one. This further demonstrates the effectiveness of our method for overlapping community discovery tasks under large sparse networks.
- Among these two types of baseline tasks, some traditional methods (BigCLAM, SNetOC, SNMF) are limited by the scale of the datasets. With the sharp increase in the number of nodes and edges in the network, the amount of computation and data has become a problem that we cannot ignore. For contrast, some GNN-based methods (NOCD, DMoN, UCoDe, DGOCD) are more suitable for large-scale sparse networks and are more adaptable to real-world situations. So far, our approach seems to have found a balance between time complexity and large-scale data and has achieved good results.

5.2.2 Advantages of dual graph neural network

RQ2 Traditional graph convolutional network (GCN) acts as the first-order local approximation of spectral graph convolution. Each convolutional layer only processes its first-order neighborhood and then stacks several convolutional layers to pass multi-order information. Based on the sparsity of the real-world network,

Table 3 NMI (in %) for overlapping community detection results on social networks. Results for DGOCD are averaged over 50 initializations. Best performer in bold; second best underlined

	Fb-348	Fb-414	Fb-686	Fb-698	Fb-1684	Fb-1912
BigCLAM	21.5	41.6	13.2	38.3	30.1	24.3
SNetOC	20.4	43.8	10.3	37.6	25.0	22.6
SNMF	12.3	26.7	11.6	23.5	11.9	25.6
NOCD-X	31.6	50.3	<u>19.5</u>	33.7	29.0	37.6
NOCD-G	**33.8**	<u>54.5</u>	16.1	<u>43.4</u>	34.2	<u>39.2</u>
DMoN-X	18.2	38.5	10.5	18.3	5.4	23.0
DMoN-G	24.6	41.3	11.4	38.1	20.7	25.6
UCoDe-X	27.1	47.2	**21.9**	23.6	19.2	27.8
UCoDe-G	22.7	42.6	12.9	40.2	29.5	25.3
DGOCD-X_{best}	30.2	51.9	19.3	42.8	<u>34.7</u>	37.9
DGOCD-G_{best}	<u>32.9</u>	**56.0**	18.6	**45.5**	**42.3**	**41.0**

Table 4 NMI (in %) for overlapping community detection results on co-authorship networks. Results for DGOCD are averaged over 50 initializations. Best performer in bold; second best underlined. DNF did not finish in 12 h or ran out of memory

	Chemistry	Computer science	Medicine	Engineering
BigCLAM	0	0	0	6.8
SNetOC	DNF	DNF	DNF	DNF
SNMF	1.8	8.6	4.1	9.6
NOCD-X	20.6	39.2	24.0	16.5
NOCD-G	<u>44.3</u>	<u>48.7</u>	<u>36.4</u>	<u>37.2</u>
DMoN-X	8.6	21.5	12.6	8.1
DMoN-G	29.3	30.4	19.2	21.6
UCoDe-X	13.2	23.7	16.4	10.5
UCoDe-G	29.8	32.9	20.6	21.8
DGOCD-X_{best}	41.1	42.7	34.0	33.8
DGOCD-G_{best}	**48.5**	**52.9**	**42.3**	**43.2**

the advantage of GCN is that it can naturally integrate the attribute information in the graph for learning, but multiple stacking will make the result of GCN over-smooth and unable to distinguish different types of edges. In view of the observation that GCN models usually perform best when stacking two to three layers in community detection tasks, we no longer try to deepen the model, but independently learn representations of information of different orders through a shallow graph neural network module. Corresponding to our empirical observation, we feed the first-order graph and the second-order graph to dual graph neural network, respectively, and then use the topological potential matrix to realize the fusion of multi-order information. In contrast, this dual graph neural network can quickly and efficiently obtain a rough representation of each node, which is fused by a topological potential matrix. In particular, the interaction between a node and its second-order neighbors is direct without third-party intervention. This is more in line with the laws of information exchange and transmission in the real world. Furthermore, the introduction of topological potential matrix allows us to manually adjust the fusion granularity of graph information according to the actual situation, which is what traditional graph convolution cannot bring us. As shown in Table 5 and Fig. 5, we introduce NOCD to achieve stronger contrast, which can be viewed as a GNN-based model. Obviously, the model based on dual graph neural network can achieve better results than the GNN-based model.

5.2.3 The balance of complexity and efficiency

RQ3 As shown in Tables 3 and 4, the performance capability of our method confirms that the introduction of the topological potential matrix is a correct choice. However, this is inseparable from the balance between penalty terms and information of different granularities. The introduction of fine-grained information can indeed define the tightness between node pairs more accurately, but it will also increase

Table 5 Ablation study of penalty term and different granularity information on social networks, measured by NMI (in %). Best performer in bold; second best underlined. We introduce NOCD to achieve stronger contrast. Subscript p means with a penalty term. Subscript mp indicates that the node mass is considered in the penalty term

	Fb-348	Fb-414	Fb-686	Fb-698	Fb-1684	Fb-1912
NOCD-X	31.6	50.3	**19.5**	33.7	29.0	37.6
NOCD-G	**33.8**	<u>54.5</u>	16.1	43.4	34.2	39.2
DGOCD-X	23.0	35.4	12.6	29.5	26.0	27.9
DGOCD-G	23.5	46.2	10.4	38.7	34.0	34.6
DGOCD-X_p	29.3	47.2	18.1	40.1	33.2	36.5
DGOCD-G_p	31.2	54.3	17.2	<u>43.6</u>	<u>41.0</u>	<u>39.8</u>
DGOCD-X_{mp}	30.2	51.9	<u>19.3</u>	42.8	34.7	37.9
DGOCD-G_{mp}	<u>32.9</u>	**56.0**	18.6	**45.5**	**42.3**	**41.0**
DGOCD-X_{best}	30.2	51.9	<u>19.3</u>	42.8	34.7	37.9
DGOCD-G_{best}	<u>32.9</u>	**56.0**	18.6	**45.5**	**42.3**	**41.0**

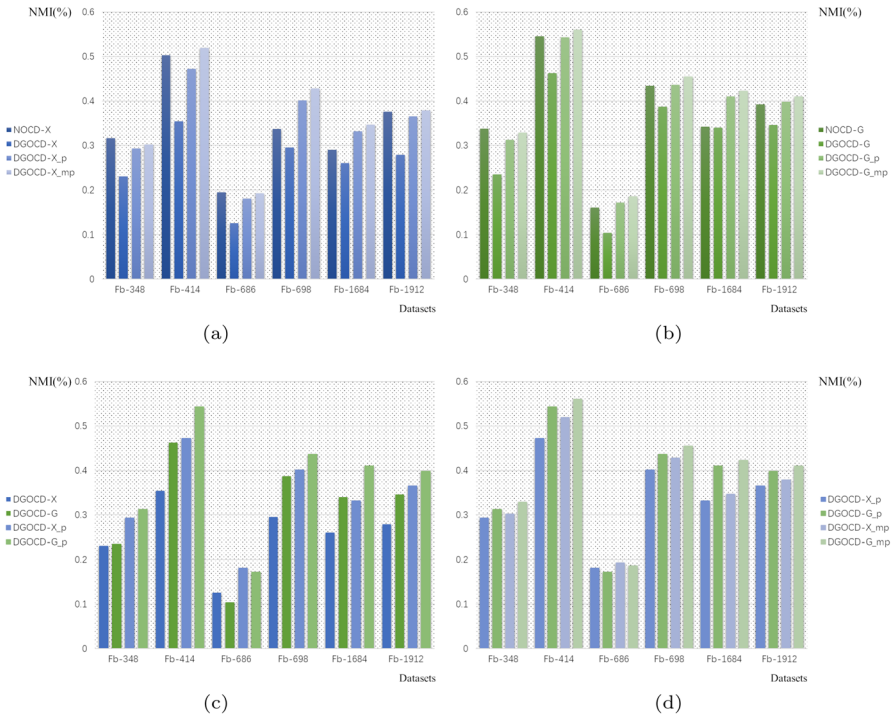


Fig. 5 Visualization of ablation study from different perspectives

the time complexity of the entire model. According to experimental observations, the refined information is not suitable for all datasets and sometimes brings negative effects to the community detection task. To strike a balance between complexity

and efficiency, we take different measures for different scale datasets. Our model can achieve refinement of graph information granularity in two places, both by adding node mass. The first is in the topological potential matrix, and the second is in the penalty term. In the co-authorship network datasets, limited by its scale, we do not consider the addition of node mass in the above two places to pursue efficiency. However, in the social network datasets, we choose to include node mass in the penalty term to achieve better results. As we expected, as shown in Table 5, the addition of node mass in the penalty term takes the model to the next level.

In order to more intuitively understand the role of penalty terms and different granularity information in the model, we discuss the importance of each item from different perspectives, as shown in Fig. 5. Figure 5 is a visualization of the data in Table 5. Figure 5a shows the NMI of the four methods under Model-X. Figure 5b shows the NMI of the four methods under Model-G. These two sub-graphs fully reflect the impact of the penalty term and node mass on the model. The addition of the penalty term has greatly improved the accuracy of the entire model, and the integration of node mass has improved the model effect by another level. Obviously, our method is always better than NOCD in most cases. Figure 5c and d shows the necessity of penalty term and node mass under Model-X and Model-G, respectively. These two sub-graphs once again confirm the efficiency of Model-G to a certain extent. Of course, this is highly consistent with our previous experimental conclusions.

5.2.4 Recovery of overlapping community scale

RQ4 As we mentioned earlier, a good overlapping community detection algorithm must be able to restore the original graph structure to the greatest extent possible, whether it is the affiliation of the community or the scale of the overlap. In the experimental test, how to measure a good community affiliation often corresponds to a higher overlapping NMI value, and a good overlap scale needs to be able to truly show the gap between the predicted and the ground truth scale. This is why we introduce COR as an evaluation indicator.

As shown in Fig. 6, we plot the COR surface plot of the best models (DGOCD and NOCD) based on the conclusions in Tables 3 and 4. We find that sparse information leads to gaps in recovery performance for overlapping regions when real-world networks are small (especially for the two datasets Fb-348 and Fb-686). When the network scale gradually expands, various models based on graph neural network can achieve better results. However, regardless of the datasets size, our method is closer to the ground truth, which further confirms the effectiveness of our model. To get a clearer picture of why our model has poor COR on these specific datasets, we visualize the adjacency matrix sorted by the predicted communities, as shown in Fig. 7. As a small and medium datasets, the information that these six Facebook networks can provide is very limited. In particular, the community affiliation matrix in the ground truth itself has not many nodes divided into each community, and some communities only have a few nodes. This leads to the fact that in the final prediction process of the model, there will inevitably be an empty community (that is, there is a community but no node). For the premise of a fixed number of communities, this

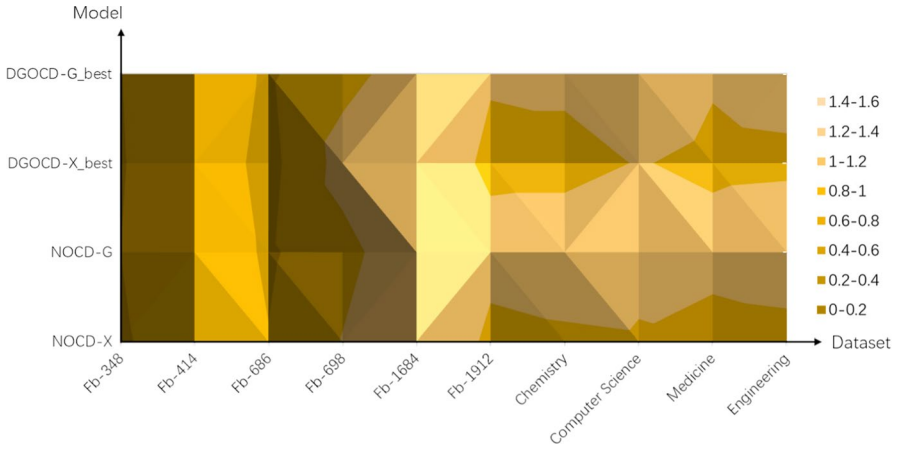


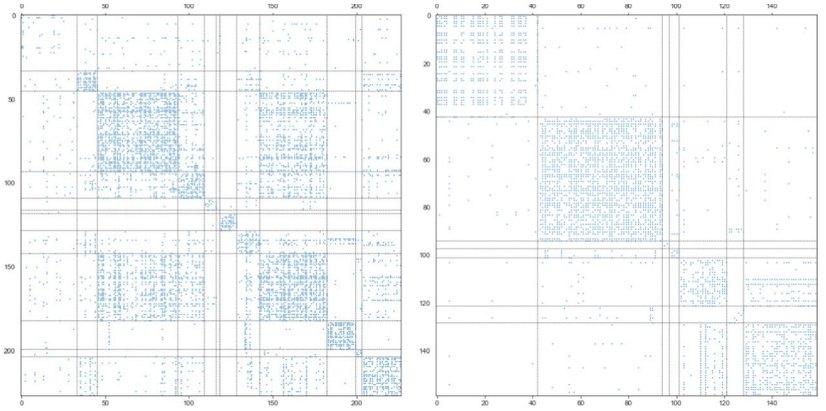
Fig. 6 COR index of NOCD and DGOCD on datasets

obviously does not make sense. Of course, this is also a problem we want to focus on and solve in the future.

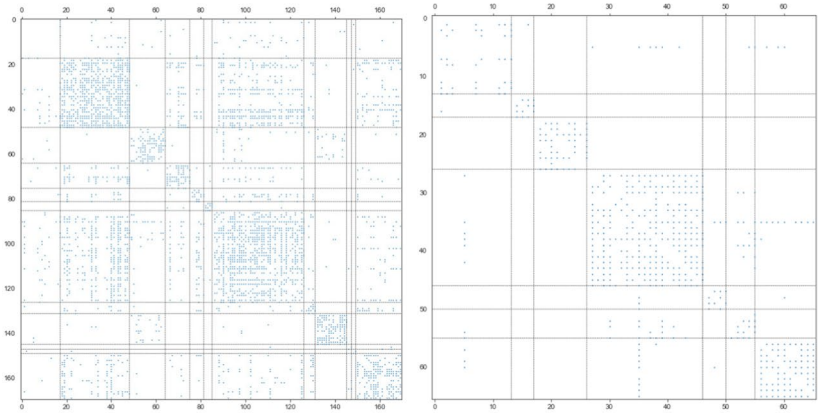
Figure 8 shows the community detection results of our model on the Fb-414 network. As a comparison, the left side of the figure shows the result of the community detection in the ground truth, and the right side shows the result of the community detection predicted by the DGOCD model. In Fig. 8a, the number of community nodes represented by green and purple both accounts for a quarter of the total. Correspondingly, in Fig. 8b, the number of nodes in the community represented by blue accounts for 31% of the total, which is the highest among all communities. The number of community nodes represented by orange accounts for 22% of the total. Of course, there are many structural features such as non-overlapping, nesting and overlapping. From the overall visual point of view, it is not difficult to find that the final community detection of the DGOCD model works well.

6 Conclusion

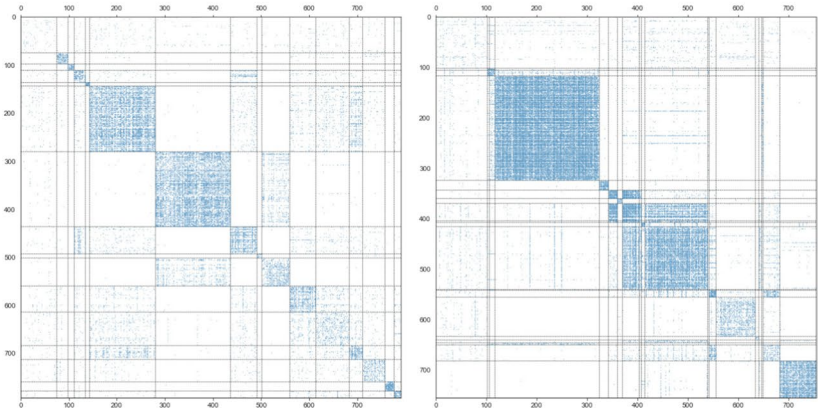
In this paper, we propose DGOCD, a dual graph neural network for overlapping community detection. The model learns different content through two sets of shallow graph neural network modules. We always believe that a good overlapping community detection method can not only accurately recover the real-world affiliation, but also reconstruct the overlapping scale of the original community as much as possible. Experimental results show that our method not only achieves better community detection, but also dynamically adjusts the information granularity on graphs of different datasets to achieve a balance between complexity and efficiency. In the future, we hope to further improve the whole model, remove the limitation on the number of communities and adapt to more real-world networks.



(a) Fb-348 (b) Fb-414

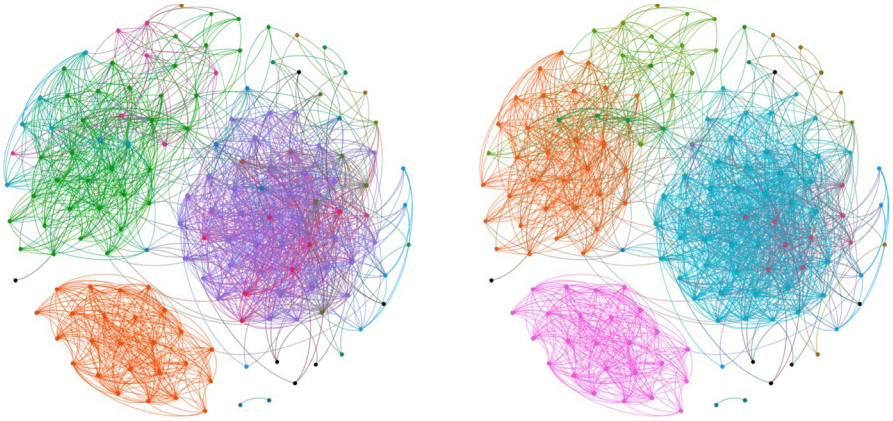


(c) Fb-686 (d) Fb-698



(e) Fb-1684 (f) Fb-1912

Fig. 7 Visualize the adjacency matrix sorted by the predicted communities



(a) The ground truth (b) Predicted community affiliations

Fig. 8 Visualization on the Fb-414 dataset

Acknowledgements This work was supported in part by National Natural Science Foundation of China (No. 61862058), Natural Science Foundation of Gansu Province (No. 20JR5RA518, 21JR7RA114), Industrial Support Project of Gansu Colleges (No. 2022CYZC11).

Author Contributions X.L and Q.P helped in conceptualization, methodology, formal analysis, software, investigation, validation, resources, writing—original draft, review and editing, visualization. R.L and H.M contributed to resources, writing—review and editing, supervision. XL and QP wrote the main manuscript text. RL conducted data processing, and Huifang Ma prepared figures. All authors reviewed the manuscript.

Data Availability The datasets analyzed during the current study were all derived from the following public domain resources. [Facebook: http://snap.stanford.edu/data/; Chemistry, Computer Science, Medicine and Engineering: https://kddcup2016.azurewebsites.net/].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent to participate The authors declare that they agree to participate.

Consent for publication The authors declare that they agree to publish.

References

1. Alchihabi A, Guo Y (2021) Dual GNNS: Graph neural network learning with limited supervision. arXiv preprint arXiv:2106.15755
2. Blondel VD, Guillaume JL, Lambiotte R et al (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 10:10008

3. Bowe M, Wakefield JR, Kellezi B et al (2022) The mental health benefits of community helping during crisis: coordinated helping, community identification and sense of unity during the covid-19 pandemic. J Comm Appl Soc Psychol 32(3):521–535
4. Chen Z, Li L, Bruna J (2020) Supervised community detection with line graph neural networks. In: International Conference on Learning Representations
5. Cui G, Zhou J, Yang C, et al (2020) Adaptive graph encoder for attributed graph embedding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 976–985
6. Duan W, Xuan J, Qiao M, et al (2022) Learning from the dark: boosting graph convolutional neural networks with diverse negative samples. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 6550–6558
7. Fowler JH, Christakis NA (2008) Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham heart study. Bmj 5:337
8. Fu S, Wang G, Xia S et al (2020) Deep multi-granularity graph embedding for user identity linkage across social networks. Knowl-Based Syst 193(105):301
9. Gan WY, He N, Li DY et al (2009) Community discovery method in networks based on topological potential. J Softw 20(8):2241–2254
10. Geerts F, Mazowiecki F, Perez G (2021) Let's agree to degree: comparing graph convolutional networks in the message-passing framework. In: International Conference on Machine Learning, PMLR, pp 3640–3649
11. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
12. Khan BS, Niazi MA (2017) Network community detection: a review and visual survey. arXiv preprint arXiv:1708.00977
13. Li P, Xie J, Wang Q, et al (2017) Is second-order information helpful for large-scale visual recognition? In: Proceedings of the IEEE International Conference on Computer Vision, pp 2070–2078
14. Li P, Xie J, Wang Q, et al (2018) Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 947–955
15. Li PZ, Huang L, Wang CD et al (2020) Community detection by motif-aware label propagation. ACM Trans Knowl Discov from Data (TKDD) 14(2):1–19
16. Liu F, Xue S, Wu J, et al (2021) Deep learning for community detection: progress, challenges and opportunities. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp 4981–4987
17. Mcauley J, Leskovec J (2014) Discovering social circles in ego networks. ACM Trans Knowl Discov from Data (TKDD) 8(1):1–28
18. McDaid AF, Greene D, Hurley N (2011) Normalized mutual information to evaluate overlapping community finding algorithms. arXiv preprint arXiv:1110.2515
19. Moradan A, Draganov A, Mottin D, et al (2021) Ucode: Unified community detection with graph convolutional networks. arXiv preprint arXiv:2112.14822
20. Naderipour M, Fazel Zarandi MH, Bastani S (2022) Fuzzy community detection on the basis of similarities in structural/attribute in large-scale social networks. Artif Intell Rev 55(2):1373–1407
21. Newman ME (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582
22. Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026,113
23. Perozzi B, Akoglu L, Iglesias Sánchez P, et al (2014) Focused clustering and outlier detection in large attributed graphs. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1346–1355
24. Petrenko M, Rajlich V (2009) Variable granularity for improving precision of impact analysis. In: 2009 IEEE 17th International Conference on Program Comprehension, IEEE, pp 10–19
25. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
26. Rahiminejad S, Maurya MR, Subramaniam S (2019) Topological and functional comparison of community detection algorithms in biological networks. BMC Bioinformatics 20(1):1–25
27. Shchur O, Günnemann S (2019) Overlapping community detection with graph neural networks. Comput Sci 50(2.0):492

28. Shen X, Chung FL (2018) Deep network embedding for graph representation learning in signed networks. IEEE Trans Cybern 50(4):1556–1568
29. Su X, Xue S, Liu F et al (2022) A comprehensive survey on community detection with deep learning. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2021.3137396
30. Todeschini A, Miscouridou X, Caron F (2020) Exchangeable random measures for sparse and modular graphs with overlapping communities. J R Stat Soc Series B (Sta Methodol) 82(2):487–520
31. Tsitsulin A, Palowitch J, Perozzi B, et al (2020) Graph clustering with graph neural networks. arXiv preprint arXiv:2006.16904
32. Wang D, Cui P, Zhu W (2016) Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1225–1234
33. Wang F, Li T, Wang X et al (2011) Community discovery using nonnegative matrix factorization. Data Min Knowl Disc 22(3):493–521
34. Wang K, Shen Z, Huang C et al (2020) Microsoft academic graph: when experts are not enough. Quant Sci Stud 1(1):396–413
35. Wang Z, Chen Z, Zhao Y et al (2014) A community detection algorithm based on topology potential and spectral clustering. Sci World J 2014:329325
36. Welling M, Kipf TN (2016) Semi-supervised classification with graph convolutional networks. In: J. International Conference on Learning Representations (ICLR 2017)
37. Wu L, Zhang Q, Chen CH et al (2020) Deep learning techniques for community detection in social networks. IEEE Access 8:96,016-96,026
38. Wu Z, Pan S, Chen F et al (2020) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32(1):4–24
39. Yang J, Leskovec J (2012) Community-affiliation graph model for overlapping network community detection. In: 2012 IEEE 12th International Conference on Data Mining, IEEE, pp 1170–1175
40. Yang J, Leskovec J (2013) Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp 587–596
41. Yang J, Leskovec J (2014) Structure and overlaps of ground-truth communities in networks. ACM Trans Intell Syst Technol (TIST) 5(2):1–35
42. Zhang T, Xiong Y, Zhang J, et al (2020) Commdgi: community detection oriented deep graph infomax. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp 1843–1852
43. Zhi-Xiao W, Ze-chao L, Xiao-fang D et al (2016) Overlapping community detection based on node location analysis. Knowl-Based Syst 105:225–235
44. Zhou M (2015) Infinite edge partition models for overlapping community detection and link prediction. In: Artificial intelligence and statistics, PMLR, pp 1135–1143
45. Zhuang C, Ma Q (2018) Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the 2018 World Wide Web Conference, pp 499–508