Network Tight Community Detection

Jiayi Deng [1] **Xiaodong Yang** [2] **Jun Yu** [3] **Jun S. Liu** [2] **Zhaiming Shen** [4] **Danyang Huang** [1] **Huimin Cheng** [5]

Abstract

Conventional community detection methods often categorize all nodes into clusters. However, the presumed community structure of interest may only be valid for a subset of nodes (named as "tight nodes"), while the rest of the network may consist of noninformative "scattered nodes". For example, a protein-protein network often contains proteins that do not belong to specific biological functional modules but are involved in more general processes, or act as bridges between different functional modules. Forcing each of these proteins into a single cluster introduces unwanted biases and obscures the underlying biological implication. To address this issue, we propose a tight community detection (TCD) method to identify tight communities excluding scattered nodes. The algorithm enjoys a strong theoretical guarantee of tight node identification accuracy and is scalable for large networks. The superiority of the proposed method is demonstrated by various synthetic and real experiments.

1. Introduction

Community detection, a task pervasive in numerous scientific realms, aims to extract coarse-grain community structures where nodes within each community are densely connected, and nodes between communities are sparsely connected. Community detection has been widely used in many applications, including identifying allegiances in social networks (Cheng et al., 2021), elucidating biological function in metabolic networks (Guimera & Nunes Amaral, 2005), and exploring homology in genetic similarity networks (Haggerty et al., 2014). Over the years, the field of community detection methods has witnessed a surge (You et al., 2016; Liu et al., 2019), starting with greedy algorithms (Clauset et al., 2004; Newman & Girvan, 2004), advancing to probabilistic model-based methods (Celisse et al., 2012; Bickel et al., 2013), and further to spectral clustering methods (Rohe et al., 2011; Jin, 2015; Deng et al., 2024).

One critical assumption in many community detection methods is that every node in the network can and should be allocated to a community. However, real-world networks often contain "scattered nodes" that do not fit into any specific community, thereby challenging this assumption. For example, a protein-protein network often contains proteins that do not belong to specific biological functional modules. Such proteins may be involved in more general processes such as system maintenance, or may act as bridges between different functional modules. Assigning these non-specific proteins to clusters can introduce biases and obscure underlying biological implications. Another example is an email network within a university, where each node represents an email address, and a connection between two nodes indicates communication. Spam accounts scattered unsolicited messages across groups, posing security threats (Shrivastava et al., 2008). Isolating these uninformative scattered nodes from tight communities mitigates privacy and financial risks.

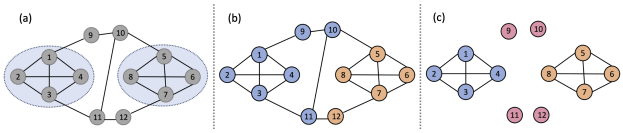

Figure 1. (a) A network with 12 nodes. (b) Spectral clustering results. (c) TCD results, where nodes 9-12 are flagged as scattered nodes, while two tight communities are extracted: nodes 1-4 and nodes 5-8.

Despite the urgent need, many conventional community detection methods overlook the presence of scattered nodes, yielding undesirable results. To illustrate, consider a toy example in Figure 1(a), where two clear communities are formed by nodes 1-4 and 5-8, while nodes 9-12 appear as scattered entities without apparent community structures. Without taking into account scattered nodes, conventional methods such as the spectral clustering method tend to inappropriately force these scattered nodes into existing clusters, as shown in Figure 1(b). This obscures the intrinsic com-

[1]Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China [2]Department of Statistics, Harvard University, Cambridge MA, USA [3]School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, China [4]Department of Mathematics, University of Georgia, Athens GA, USA [5]Department of Biostatistics, Boston University, Boston MA, USA. Correspondence to: Huimin Cheng <huimin23@bu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

munity structure of the network. To accurately model such networks, we assume a network has n tight nodes and m scattered nodes. Tight nodes preferentially connect within their own communities rather than between communities. Tight communities are community structures formed exclusively by these tight nodes. Scattered nodes do not show any community structure, connecting with other nodes in the graph in an arbitrary way.

To take into account scattered nodes, Cai & Li (2015) proposed a robust community detection method. However, this method focuses on minimizing the impact of scattered nodes on the clustering process, without assigning labels to individual nodes to indicate whether they are scattered or tight. A more recent work (Gaucher et al., 2021) identifies scattered nodes using optimization but relies on the assumption that these scattered nodes are hubs and their counts are significantly lower than that of tight nodes. Additionally, (Dey et al., 2023) proposed a node-based metric to identify scattered nodes. Another line of research related to our work is outlier detection with statistical (Yamanishi & Takeuchi, 2001; Rousseeuw & Hubert, 2011; 2018), proximity-based (Ramaswamy et al., 2000; Aggarwal & Aggarwal, 2017; Abuzaid, 2020), and neural network (Hawkins et al., 2002; Chen et al., 2017; Goodge et al., 2022) approaches. Outliers in the aforementioned literature are often seen as nodes belonging to multiple communities or having weak connections (Xu et al., 2007; Dey et al., 2023). Our approach, however, considers scattered nodes as those without any community affiliation. Other related topics include community extraction (Zhao et al., 2011; Gibbs et al., 2022) and local clustering (Andersen et al., 2006; Kloster & Gleich, 2014; Mahoney et al., 2012; Li et al., 2018; Lai & Mckenzie, 2020; Lai & Shen, 2023; Shen et al., 2023), however, they either suffer from a lack of theoretical guarantee or computational efficiency.

In this work, we propose a tight community detection (TCD) method, motivated by a tight clustering method for i.i.d data (Tseng & Wong, 2005) and the network sub-sampling based community detection method in (Mukherjee et al., 2021). TCD first employs a network sub-sampling approach to get multiple sub-networks, detects community structure for each sub-network, and constructs an average co-membership matrix with its (i, j)th entry representing the frequency of nodes i and j being clustered in the same community. Then, it uses a depth-first search (DFS) method to search for stable tight components in the co-membership matrix, which will be treated as tight nodes. It finally applies spectral clustering to the sub-network consisting of only tight nodes to identify the tight community structure. The computational cost of TCD is scalable at $O(N^2)$, where N is the number of nodes. We demonstrate the excellent empirical performance of TCD in comparison with existing methods by both extensive simulation studies and a real protein-protein network.

2. Model Setup

Let $G = (V, E)$ denote a random graph, where V represents a fixed set of nodes, and E, a random set of edges. Let $\mathbf{A} = (a_{ij})_{1\le i,j\le n}$ denote the adjacency matrix of this graph, where n is the number of nodes, and $a_{ij} = 1$ or 0 if node i and j are or are not connected by an edge. We only consider a network with no self-loops, so all diagonal entries of $\mathbf{A}$ are 0. In this paper, we present our idea based on undirected networks ($\mathbf{A}$ is symmetric), but it can be easily extended to directed networks ($\mathbf{A}$ is not symmetric).

Stochastic block model (SBM). The stochastic block model (SBM) is a popular probabilistic framework for modeling the connectivity in random networks (Holland et al., 1983; Karrer & Newman, 2011; Rohe et al., 2011; Lei & Rinaldo, 2015; Paul & Chen, 2020; Yang et al., 2024). In SBM, nodes are assigned to specific latent groups, known as communities, and the probability of edge formation between any two nodes is determined by their community membership. Let $z_i \in \{1, \cdots, K\}$ denote the community label of node i, where K is the number of communities. Let $\mathbf{\Theta} \in \{0, 1\}^{n\times K}$ denote the membership matrix, where $\mathbf{\Theta}_{iz_i} = 1$. For any node pair (i, j), $a_{ij} \sim \text{Bernoulli}(p_{ij})$, where $p_{ij} = \mathbf{B}_{z_i z_j}$, and $\mathbf{B}_{z_i z_j}$ is the connecting probability between any node in community z_i and any node in community z_j. SBM usually assumes that the connectivity probability within a community is larger than that between communities, In sum, SBM is parameterized by

$$\widetilde{\mathbf{P}} := \mathbf{\Theta}\mathbf{B}\mathbf{\Theta}^T \in (0, 1)^{n\times n}, \tag{1}$$

where $\mathbf{B} = (\mathbf{B}_{kl}) \in (0, 1)^{K\times K}$ is the community connectivity matrix. Extensions of SBM include degree-corrected SBM (Karrer & Newman, 2011), the overlapping SBM (Latouche et al., 2011), the mixed membership SBM (Airoldi et al., 2008), and binary tree SBM (Li et al., 2022).

General stochastic block model (GSBM). Despite their popularity, all aforementioned models do not consider the scenario where scattered nodes exist. Cai & Li (2015) proposes a general stochastic block model (GSBM) that takes into account the presence of scattered nodes. Following the notation of GSBM (Cai & Li, 2015), we assume that our network $G = (V, E)$ has $N := n + m$ nodes, among which there are n *tight nodes* having community structures and m *scattered nodes* having no community structure. Let $\mathcal{T}$ denote the set of the tight nodes, and $\mathcal{S}$ denote the set of the scattered nodes, such that $V = \mathcal{T} \cup \mathcal{S}$.

The ordered probability matrix under GSBM is

$$\mathbf{P} = \begin{pmatrix} \widetilde{\mathbf{P}} & \mathbf{D}_1 \\ \mathbf{D}_1^\top & \mathbf{D}_2 \end{pmatrix} \in (0, 1)^{N\times N}, \tag{2}$$

where $\widetilde{\mathbf{P}} \in (0,1)^{n\times n}$ is a block probability matrix in the usual SBM, which models the connection between tight nodes. $\mathbf{D}_1 = (\mathbf{D}_{1ij}) \in (0,1)^{n\times m}$ models the connectivity between tight nodes and scattered nodes. $\mathbf{D}_2 = (\mathbf{D}_{2ij}) \in (0,1)^{m\times m}$ models the connectivity between scattered nodes. Scattered nodes do not admit community structure and do not belong to any specific community. Therefore, the connection probability matrix $\mathbf{P}$ should not exhibit block structure except within the tight community subsection $\widetilde{\mathbf{P}}$. Here, we present three scenarios that adhere to this structure in the order of increasing generality.

- *Erdős–Rényi (ER)-type scattered nodes.* A scattered node connects to any other node with identical probability. That is, $(\mathbf{P})_{ij} = \zeta$ if any of i or j is in $\mathcal{S}$. The ER-type connection is arguably the most basic form of non-informative structure.
- *Inhomogeneous ER (IER)-type scattered nodes.* A scattered node connects with any other node with a probability drawn from a uniform distribution, i.e., $U(\zeta_{min}, \zeta_{max})$. This captures the essence of the non-block structure of scattered nodes while permitting variability in edge probabilities across node pairs. Here the expected degree of each scattered node is the same, $N(\zeta_{min} + \zeta_{max})/2$.
- *Heterogenous degree (HetD)-type scattered nodes.* A scattered node $i \in \mathcal{S}$ connects with any other node with a probability drawn from its own uniform distribution, i.e., $U(\zeta^i_{min}, \zeta^i_{max})$. This can adopt arbitrary degree distributions for the scattered nodes.

Examples of SBM and GSBM. Figure 2(a) shows an example of the ordered probability matrix under SBM. In this example, there are $n = 120$ nodes partitioned into $K = 3$ communities, i.e., community 1 contains nodes 1-40, community 2 contains nodes 41-80, and community 3 contains nodes 81-120. The within-group probability is set to 0.3, and the cross-group probability is 0.03. For the GSBM example, all 120 nodes are considered tight nodes, and the model includes 10 scattered nodes. These scattered nodes are of different types: ER-type, $\zeta = 0.1$; IER-type, $\zeta_{\min} = 0, \zeta_{\max} = 0.2$; HetD-type, $\zeta^i_{\min} = 0$, and $\zeta^i_{\max}$ is randomly generated from the interval $(0.1, 0.2)$, for all $i = 1, \cdots, m$. The corresponding ordered probability matrices are shown in Figure 2(b)-(d), respectively.

Given that scattered nodes lack community structure, applying spectral clustering directly to networks containing such nodes may yield contaminated clustering results. Specifically, spectral clustering tends to assign scattered nodes to communities arbitrarily rather than leaving them unassigned or clustering them together. This arbitrary assignment of scattered nodes makes it more difficult to distinguish them from tight nodes, as they are not consistently associated with any specific group. To demonstrate this issue, we employ an averaged co-membership matrix that quantifies the co-clustering frequency of node pairs. By simulating 100 networks from the same underlying model and applying spectral clustering to them, we obtain a set of co-membership matrices. Averaging these co-membership matrices, we obtain a matrix where (i, j)th entry represents the proportion of nodes i and j being grouped together.

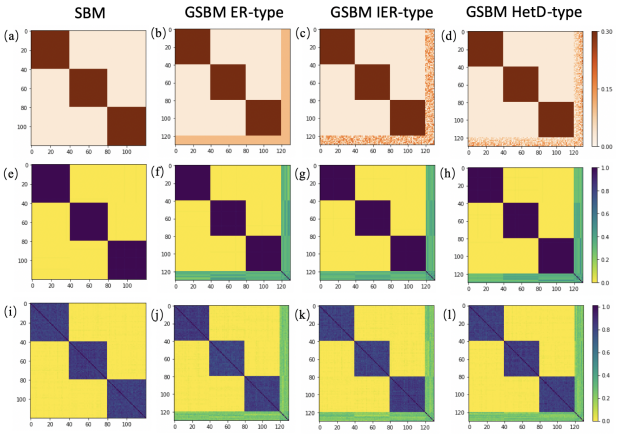

Figure 2. Column 1: SBM. Columns 2-4: GSBM with ER, IER, and HetD scattered nodes. (a)-(d) Ordered probability matrices. (e)-(h), (i)-(l) are averaged co-membership matrices for 3 and 4 communities, respectively.

Figure 2(e)-(h) shows the co-membership matrix results when setting the number of communities as three when applying the spectral clustering method. In the case of an SBM without scattered nodes, as depicted in Figure 2(e), the averaged co-membership matrix demonstrates perfect community recovery: the nodes within the predefined clusters (1-40, 41-80, and 81-120) are consistently grouped together with a co-membership frequency of one. Figure 2(f) shows the result under a GSBM with additional ER-type scattered nodes, while the original communities of nodes 1-40, 41-80, and 81-120 remain intact, the scattered nodes do not display such consistent clustering behavior. The co-membership frequency between any scattered node and the rest of the network is similar, reflecting their random assignment rather than a systematically clustered status. We can reach similar conclusions for GSBM with IER-type and HetD-type scattered nodes, as illustrated in Figure 2(g)-(h). We further demonstrate that simply treating all scattered nodes as an additional single community fails to mitigate this issue. As shown in Figure 2(i)-(l), setting the number of communities to 4 in spectral clustering, tight nodes still show a tendency to co-cluster, albeit at a reduced frequency, while scattered nodes remain randomly distributed.

These results reveal a phenomenon that is seemingly trivial, yet extremely important: tight nodes within communities exhibit a strong tendency to co-cluster, while scattered nodes exhibit no such tendency. This suggests that properly estimated co-membership frequencies could enable distinguish-

ing tight nodes from scattered nodes. However, in practice, we observe only a single instance of a network, yielding only one co-membership matrix from clustering rather than an averaged co-membership matrix which could capture the frequency. To surmount this challenge, we show in the following section how to employ a network sub-sampling approach to generate multiple sub-samples of the original network, based on which we propose a tight community detection (TCD) method.

3. Tight Community Detection

The main idea of TCD is to use a network sampling procedure to get multiple sub-networks and detect community structure for each sub-network. If a pair of nodes are stably clustered together, they are more likely to be tight nodes from the same community. Figure 3 shows the workflow of TCD, which works in the following steps.

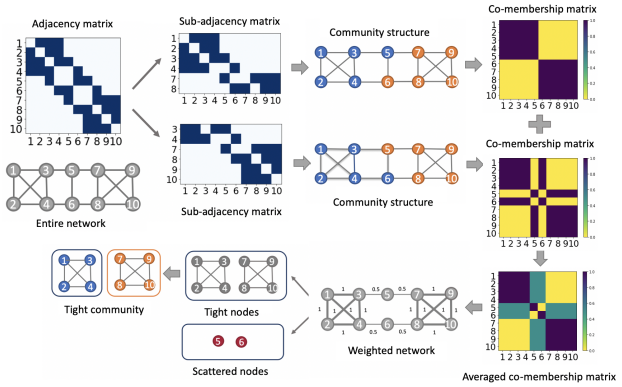

Figure 3. Workflow of TCD demonstrated using a toy network with ten nodes. TCD randomly selects $L = 2$ sets of nodes, $\{1, 2, 3, 4, 7, 8\}$ and $\{3, 4, 7, 8, 9, 10\}$, which give rise to two rectangular sub-adjacency matrices. We apply SVD to each rectangular matrix to obtain its $\tilde{K} = 2$ leading right singular vectors, and then conduct k-means clustering on the rows of these singular vectors to group all nodes into $\tilde{K} = 2$ clusters. We then construct an average co-membership matrix. DFS is utilized to differentiate tight from scattered nodes, with spectral clustering subsequently applied to tight nodes to recover community structures.

Network sub-sampling. To use the network sub-sampling method in Chen & Lei (2018), we randomly select a set of nodes $\tilde{V}$ and then construct a rectangular sub-adjacency matrix $\tilde{\mathbf{A}} \in \{0, 1\}^{\tilde{N} \times N}$ by selecting rows corresponding to $\tilde{V}$ from the original adjacency matrix $\mathbf{A}$, where $\tilde{N} = |\tilde{V}|$ is the number of selected nodes. We repeat this network sub-sampling procedure L times to obtain L rectangular matrices $\tilde{\mathbf{A}}^{(1)}, \tilde{\mathbf{A}}^{(2)}, \ldots, \tilde{\mathbf{A}}^{(L)}$. For instance, in Figure 3, we obtain two 6×10 sub-adjacency matrices by randomly selecting $L = 2$ subsets of nodes from a 10-node network. We then use each rectangular matrix to get an estimate of community labels, as we will show below.

Intermediate community detection. For each rectangular matrix $\tilde{\mathbf{A}}^{(l)}, l = 1, \ldots, L$, we apply singular value decomposition (SVD) to obtain its leading $\tilde{K}$ right singular vectors, where $\tilde{K}$ is pre-specified number of communities. We then conduct k-means clustering on the rows of these singular vectors to get $\tilde{K}$ clusters, thus obtaining estimated community labels for all nodes. For example, as shown in column 3 of Figure 3, the clustering result based on each of the sub-adjacency matrices with $\tilde{K} = 2$ is indicated by blue (cluster 1) and orange (cluster 2) labels on the nodes. Based on the clustering results using $\tilde{\mathbf{A}}^{(l)}$, we construct a co-membership matrix $\mathbf{C}^{(l)} = (\mathbf{C}^{(l)}_{ij}) \in \{0, 1\}^{N \times N}$, where $\mathbf{C}^{(l)}_{ij} = 1$ if nodes i and j are grouped into the same cluster, and $\mathbf{C}^{(l)}_{ij} = 0$ otherwise.

Average co-membership matrix construction. To aggregate the results across the L replications, we average the obtained co-membership matrices to obtain $\bar{\mathbf{C}} = \sum_{l=1}^{L} \mathbf{C}^{(l)}/L$. Critically, an (i, j)th entry $\bar{\mathbf{C}}_{ij}$ in $\bar{\mathbf{C}}$ represents the frequency that nodes i and j are clustered into the identical community over L replications. For example, in Figure 3, nodes 5 and 7 are not clustered into the same community based on the first sub-adjacency matrix, so their corresponding co-membership label is 0. However, they do cluster together in the second iteration, giving a co-membership value of 1. Averaging these, the final co-membership proportion is 0.5. Intuitively, two tight nodes within the same latent community have a higher chance to be consistently co-clustered together across L resamples. This motivates us to use $\bar{\mathbf{C}}$ to search for tight nodes.

Distinguishing tight nodes from scattered nodes. In this step, we first construct a weighted graph representation of the average co-membership matrix $\bar{\mathbf{C}}$, where the edge weight between nodes i and j denotes the resampling frequency $\bar{\mathbf{C}}_{ij}$ that the two nodes i and j are assigned to the same community. We then employ the depth-first search (DFS) (Tarjan, 1972) to extract potential tight community candidates from this weighted graph. Specifically, the DFS starts at a random node i, initializing the visited node set $\mathcal{V} = \{i\}$. It then moves to an unvisited neighboring node j, if its connection strength with all nodes in $\mathcal{V}$ exceeds $1 - \alpha$, where $\alpha \in (0, 1)$ is a hyperparameter close to zero. Expand $\mathcal{V}$ to include j. The search continues until no neighboring unvisited nodes remain, marking the end of a path and identifying all nodes in $\mathcal{V}$ as a tight component, which are then excluded from further search. The process restarts from another unvisited node if any remain, iterating until every node has been visited. If our method identifies a total of $K_{\mathcal{V}}$ such tight components, we can represent them as $\mathcal{V}_1, \cdots, \mathcal{V}_{K_{\mathcal{V}}}$. The union of nodes encompassed within these tight components is then labeled as the set of estimated tight nodes, denoted by $\hat{\mathcal{T}} = \cup_{k=1}^{K_{\mathcal{V}}} \mathcal{V}_k$. The remaining nodes are identified as scattered nodes, denoted by $\hat{\mathcal{S}} = V \setminus \hat{\mathcal{T}}$. For example, in Figure 3, we identify $K_{\mathcal{V}} = 2$ tight components, $\{1, 2, 3, 4\}$ and $\{7, 8, 9, 10\}$. So the output tight nodes in

this example are nodes $1-4$ and $7-10$, and output scattered nodes are $\{5, 6\}$.

Tight community structure recovery. The final step is to extract the submatrix between estimated tight nodes $\hat{\mathcal{T}}$. A subsequent spectral clustering onto this submatrix gives the community labels for tight nodes.

Computational complexity. The pseudo-code of TCD is summarized in Algorithm 1. Under the assumption that the number of replications L is a constant, the computational complexity of TCD is $O(N^2)$, where N is the number of nodes. The specific details for analyzing the computational complexity can be found in Appendix D. We also note that the L resampling replications are trivially parallelizable. Details on the implementation of the parallel computing variant are provided in Appendix D. Thus, our algorithm is scalable and efficient for large-scale network analysis.

Algorithm 1 Tight Community Detection Algorithm (TCD)

Input: Adjacency matrix $\mathbf{A}$, number of communities $\tilde{K}$, sub-sampling repetitions L, sub-sampling size $\tilde{N}$, and tightness hyperparameter α.
Output: Scattered nodes $\hat{\mathcal{S}}$, tight nodes $\hat{\mathcal{T}}$, and the estimated community label vector $\hat{\mathbf{z}}$.
Step 1. For $l = 1, \cdots, L$:

- **Step 1.1. Network sub-sampling.** Randomly select $\tilde{N}$ rows from the entire adjacency matrix, and obtain a rectangular sub-adjacency matrix $\tilde{\mathbf{A}}^{(l)} = (a_{ij}^{(l)})$.
- **Step 1.2. Intermediate community detection.** Perform SVD on $\tilde{\mathbf{A}}^{(l)}$, obtaining its leading $\tilde{K}$ right singular vectors. Apply k-means clustering on the rows of the vectors to estimate $\tilde{z}^{(l)}$.
- **Step 1.3. Co-membership matrix construction.** Calculate the co-membership matrix $\mathbf{C}^{(l)}$ based on $\tilde{z}^{(l)}$ by $\mathbf{C}_{ij}^{(l)} = \mathbb{1}\{\tilde{z}_i^{(l)} = \tilde{z}_j^{(l)}\}$.

Step 2. Get averaged co-membership matrix across L replications, i.e., $\bar{\mathbf{C}} = \sum_{l=1}^{L} \mathbf{C}^{(l)}/L$.
Step 3. Perform the DFS to obtain tight component $\mathcal{V}_k$ such that for any $i, j \in \mathcal{V}_k$, $\bar{\mathbf{C}}_{ij} > 1 - \alpha$. Estimated tight nodes are $\hat{\mathcal{T}} = \cup_k \mathcal{V}_k$, and scattered nodes are $\hat{\mathcal{S}} = V - \hat{\mathcal{T}}$.
Step 4. Perform spectral clustering to the square sub-adjacency matrix $(a_{ij})_{i,j\in\hat{\mathcal{T}}}$ to obtain the tight community label $\hat{\mathbf{z}} = (\hat{z}_i)_{i\in\hat{\mathcal{T}}}$.

4. Theoretical Properties

To establish the theoretical properties of TCD, we impose the following assumptions.

Assumption 4.1 (Edge probability between tight nodes)**.** The connectivity matrix between communities takes the form of $\mathbf{B} = \rho_n \mathbf{B}_0$ where $\rho_n = \Omega(\log n/n)$ and $\mathbf{B}_0$ is a fixed matrix with distinct rows and K non-degenerate eigenvalues.

Assumption 4.1 allows the edge probability between tight nodes to decrease at a rate ρ_n as n increases, requiring ρ_n to be no greater than $\log n/n$. It ensures that the network becomes sparser with increasing n, at a controlled rate that retains sufficient connectivity to obtain accurate community structure recovery. Similar conditions have been used by other community detection literature (Lei & Rinaldo, 2015; Paul & Chen, 2020; Deng et al., 2024).

Assumption 4.2 (Edge probability involving scattered nodes)**.** The connectivity of scattered nodes is uniformly upper bounded, satisfying

$$\max\{\mathbb{P}(a_{ij} = 1) : i \in \mathcal{S} \text{ or } j \in \mathcal{S}\} = O(\sqrt{\rho_n/m}).$$

Assumption 4.2 postulates that the edge probability involving scattered nodes is not arbitrary but is constrained by an upper bound. For example, when $m = n/\log n$ and $\rho_n = \log n/n$, the connectivity upper bound of scattered nodes is $\log n/n$. In this case, tight and scattered nodes have indistinguishable connectivity strength, indicating that brute-force identification of tight nodes solely based on their degree centrality measure will be ineffective.

Assumption 4.3 (Number of scatter nodes)**.** The number of scattered nodes is no greater than the minimum community size.

Assumption 4.3 requires m to be no greater than the minimum community size. In particular, for a network with the same number of nodes in each community, we require $m \leq n/K$, a less restrictive condition compared with existing literature (Cai & Li, 2015; Gaucher et al., 2021).

Assumption 4.4 (Community Structure)**.** The connection between tight nodes follows the the Stochastic Block Model (SBM).

Assumption 4.4 clearly states that the considered graph follows the SBM, which guarantees the existence of community structure among tight nodes. In this scenario, theoretical properties of eigenvalues and eigenvectors of tight and scattered nodes are derived in the following lemma.

Lemma 4.5. *Let* $\mathbf{D} = \begin{pmatrix} \mathbf{0} & \mathbf{D}_1 \\ \mathbf{D}_1^\top & \mathbf{D}_2 \end{pmatrix}$, *where* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *are the ordered probability matrix in Eq. (2). Under Assumptions 4.1–4.4, one has*

$$\frac{\lambda_K(\boldsymbol{P})}{\lambda_{\max}(\mathbf{D})} = \Omega\left(\sqrt{n\rho_n}\right).$$

Due to this eigengap, the top-K eigenvectors of $\mathbf{P}$ *defined in Eq. (2) is captured by its upper left block* $\tilde{\mathbf{P}}$*, and thus can be approximated by* $\mathbf{U} = \begin{pmatrix} \boldsymbol{\Theta}\mathbf{R} \\ \mathbf{0} \end{pmatrix}$. *Here* $\boldsymbol{\Theta} \in \{0, 1\}^{n\times K}$ *is the membership matrix of* n *tight nodes, where* (i, k)*th entry is one if* i *belongs to the community* k *and zero otherwise.* $\sqrt{K/n}\mathbf{R} \in \mathbb{R}^{K\times K}$ *is a rotation matrix, and* $\mathbf{0} \in 0^{m\times K}$.

The proof of Lemma 4.5 is given in Appendix B.1. Lemma 4.5 states that the ratio of the K-th largest eigenvalue of the probability matrix $\mathbf{P}$ and the largest eigenvalue of a matrix $\mathbf{D}$ is at least on the order of $\sqrt{n\rho_n}$. This eigenvalue gap indicates that the subspace spanned by the top-K eigenvectors can be primarily characterized by the upper right block $\tilde{P}$. Thus, these eigenvectors can be approximated by $\mathbf{U}$. The projection of the tight nodes onto the top-K eigenvector space is $\mathbf{\Theta R}$, while the projection of the scattered nodes is represented by the bottom m rows of zeros. Thus, the distance from a scattered node to any community centroid is equivalent, i.e., $\sqrt{2K/n}$. The equidistance implies that when the k-means algorithm is applied to $\mathbf{U}$, a scattered node will be randomly assigned a community label. This lemma provides a theoretical explanation for the empirical finding, which observes that scattered nodes are assigned random community labels in each resampling replication. The following theorem derives some statistical properties of $\bar{\mathbf{C}}_{ij}$ for two tight nodes.

Theorem 4.6. *Under Assumptions 4.1–4.4, there exists a constant C, with high probability over $1 - 1/2N$, we have*

$$\frac{2}{n(n-1)} \sum_{1 \le i \ne j \le n} \left| \bar{\mathbf{C}}_{ij} - \left(\mathbf{\Theta}\mathbf{\Theta}^\top\right)_{ij} \right| \le \frac{C}{n\rho_n},$$

where i, j are two distinct tight nodes. Moreover, there exists an event $\mathcal{E}$, which is only related to the generating process of $\mathbf{A}$ and happens with probability $1 - 1/2N$, such that repeated sub-sampling greatly reduces conditional estimation variance,

$$\frac{2}{n(n-1)} \sum_{1 \le i \ne j \le n} \mathrm{Var}(\bar{\mathbf{C}}_{ij} \mid \mathcal{E}) \le \frac{C}{Ln\rho_n}.$$

A proof of the theorem is given in Appendix B.2. Theorem 4.6 implies that differences between the average co-membership entries $\bar{\mathbf{C}}_{ij}$ for tight nodes, obtained using TCD, and the true co-membership $\left(\mathbf{\Theta}\mathbf{\Theta}^\top\right)_{ij}$ is bounded by a term that decays to 0 as the number of tight nodes $n \to 0$. In addition, performing sub-sampling multiple times reduces the overall variance of the co-membership estimates, making the estimation more reliable.

We then analyze the behavior of $\bar{\mathbf{C}}_{ij}$ for scattered nodes. From Lemma 4.5, we know that in the top-K eigenvector space of $\mathbf{P}$, each scattered node has the same distance to any community centroid, thus will be assigned to each community with probability $1/K$. This theoretical underpinning is substantiated by empirical evidence, where we observe that, in practice, a scattered node is randomly assigned to a community in each resampling replication, as shown in Figure 8. This implies that $\bar{\mathbf{C}}_{ij}$ is bounded away from 0 and 1, where i or j is a scattered node. From Theorem 4.6, we know that $\bar{\mathbf{C}}_{ij}$ for two tight nodes will converge to either one or zero when the number of replications L converges to infinity. Thus, there is a distinction in the co-membership patterns when comparing tight and scattered nodes. Leveraging this distinction, we can employ the co-membership patterns to differentiate between tight and scattered nodes. This explains why our algorithm works.

5. Simulation Studies

In this section, we present the simulation setup and empirical results for synthetic datasets generated using the GSBM model with IER-type scattered nodes. In the appendix, we also show some results for the GSBM model with ER-type and HetD-type scattered nodes. For evaluation purposes, we consider the following two metrics. (1) To quantify the accuracy in identifying scattered nodes, we use F-score:$= 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$ where precision is the ratio of the number of identified true scattered nodes over the total number of identified scattered nodes, and recall is the ratio of the number of identified true scattered nodes over the total number of true scattered nodes. (2) To measure the tight community detection accuracy, we adopt the misclustering rate for all tight nodes, which is widely used in the community detection literature (Lei & Rinaldo, 2015). For example, suppose we have 100 true tight nodes: nodes 1-50 are community 1, and nodes 51-100 are community 2. We identify nodes 1-90 as tight nodes: nodes 1-50 as community 1, and nodes 51-90 as community 2. Missed tight nodes 91-100 are labeled as community 3. We then compare $(\mathbf{I}_{50}, 2\mathbf{I}_{40}, 3\mathbf{I}_{10})$ with $(\mathbf{I}_{50}, 2\mathbf{I}_{50})$ to calculate misclustering rate, where $\mathbf{I}_n$ is an n-length vector of ones. A higher F-score and a lower misclustering rate indicate better results.

We compare our method with Community Boundary Nodes (CBNs) (Dey et al., 2023) and also benchmark against three recent graph outlier detection methods. These methods, RADAR (Li et al., 2017), ANOMALOUS (Peng et al., 2018), and ONE (Bandyopadhyay et al., 2019), have outlier definitions different from the scattered nodes definition in this paper. Following reviewers' suggestions, we incorporate six additional methods for comparison: Two local clustering approaches, PageRank-Nibble (PRN) (Andersen et al., 2006) and a heat kernel-based method (HK; Kloster & Gleich 2014); two community extraction techniques, Extraction (Zhao et al., 2011) and ECoHeN (Gibbs et al., 2022); and two community detection methods, hierarchical community detection (HCD; Li et al. 2022), and robust community detection (RCD; Cai & Li 2015). Note that HCD and RCD cannot detect scattered nodes, leading to non-applicable (NA) F-scores. All numerical experiments were implemented in Python 3.10 on a Linux server consisting of a 2.2 GHz 24-core Intel Xeon E5-2650 v4 CPU and 64GB of RAM memory capacity.

Hyperparameters. The TCD algorithm has four hyperpa-

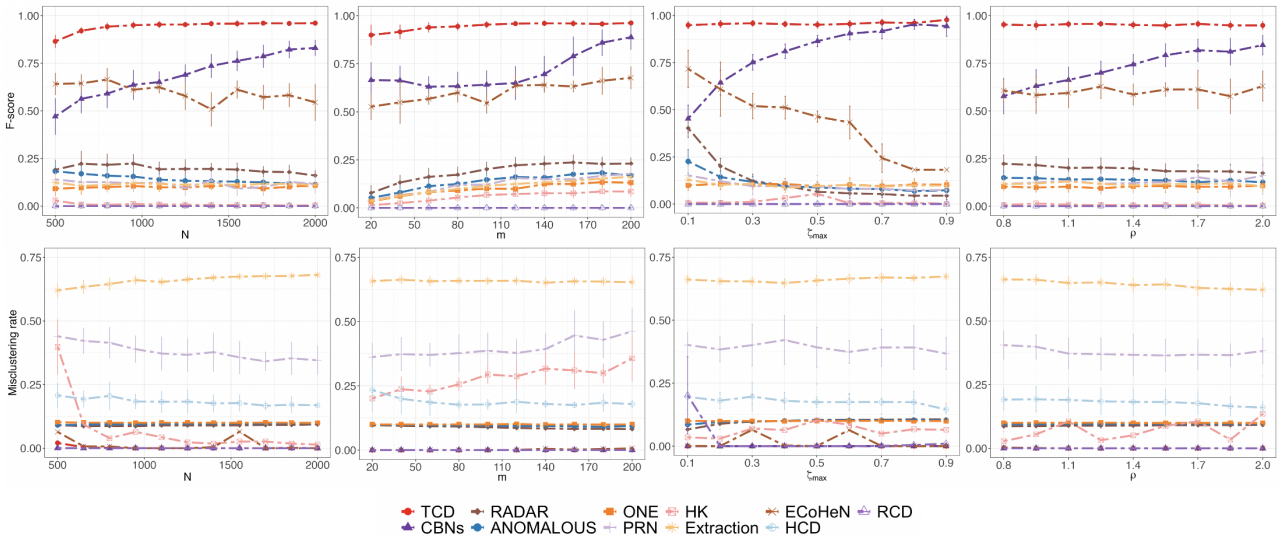

Figure 4. The top panel illustrates the F-score for identifying scattered nodes. Meanwhile, the bottom panel shows the misclustering rate of recovering community labels for tight nodes. The error bars in both panels represent the standard deviation calculated from 100 repetitions.

rameters. (1) The network sub-sampling procedure involves the number of subsampled nodes, $\tilde{N}$. The sub-sampling proportion is $\tilde{N}/N$. Empirical results, as depicted in Figure 5, demonstrate that TCD maintains stable accuracy for sub-sampling rates between 45% and 75%. (2) The count of sub-sampling replications L is also a hyperparameter. Our numerical results in Figure 5 suggest that the accuracy is stable for $L \in (10, 250)$. (3) The tight component search step involves a hyperparameter α, which controls the strictness in identifying tight components. While the numerical results in Figure 5 advocate for an α value of 0.1 for optimal performance, the algorithm exhibits a commendable tolerance to variations around this value, indicating its robustness in delineating tightly-knit communities. (4) The last hyperparameter is the number of communities, $\tilde{K}$. This hyperparameter is determined through a cross-validation technique, as proposed in the studies by Chen & Lei (2018) and Li et al. (2020). In all of our examples, we set $\tilde{N}/N = 0.7$, $\alpha = 0.1$, $L = 50$.

Simulation setup. We generated synthetic datasets according to GSBM with IER-type scattered nodes. In Appendix C.1, we also show simulation results for ER-type and HetD-type scattered nodes. Specifically, we generated an N-node network with m scattered nodes and $N - m$ tight nodes. We employed a three-community SBM to generate communities among the tight nodes. Let p denote the intra-community connection probability and q the inter-community connection probability. A scattered node is connected randomly to all the other nodes in the network with a probability drawn from $U(0, \zeta_{\max})$. We considered the following simulation scenarios. Scenario 1: We vary $N \in \{500, 700, \ldots, 2000\}$, while fixing $p = 0.3, q = 0.12, \zeta_{\max} = 0.2, m = 0.1N$, to examine asymptotic performances. Scenario 2: We vary m from 20 to 200 while fixing $N = 1000, p = 0.3, q = 0.12, \zeta_{\max} = 0.2$, to investigate the impact of the number of scattered nodes. Scenario 3: We vary $\zeta_{\max}$ from 0.1 to 0.9, while fixing $N = 1000, m = 100, p = 0.3, q = 0.12$, to investigate the impact of the scattered node connectivity strength. Scenario 4: We introduce a new notation ρ to control the overall network density. We vary ρ from 0.8 to 2, while fixing $N = 1000, m = 100, p = 0.3\rho, q = 0.12\rho, \zeta_{\max} = 0.2\rho$, to investigate the impact of overall network density.

Simulation results. Figure 4 shows the simulation results, demonstrating that TCD outperformed other methods in both F-score and misclustering rate across all scenarios. The top panel shows the F-score results and the bottom panel shows the misclustering rate results over 100 replications. The four columns represent Scenarios 1-4, respectively. In Scenario 1, as the network size N increases, TCD's F-scores increase to one and its misclustering rates decay to zero, very fast with low variance. CBNs performed the second best, but with a high variance. In Scenario 2, as the number of scattered nodes m increases from 20 to 80, the F-score of TCD increases from 0.8 to nearly 1 with decreasing variances. This performance improvement is attributed to the stronger signal provided by the greater number of scattered nodes, enabling more accurate detection. TCD demonstrate robust performance for both Scenario 3 and Scenario 4, with an F-score remaining close to one and a misclustering rate close to zero. In contrast, CBNs exhibits inferior performance at lower ζ_{max} and lower ρ, due to its theoretical reliance on the presence of hub-like scattered nodes to enable effective detection.

Table 1. Results (mean $\pm$ *std*) on benchmark datasets with scattered nodes scaled to one-fourth of original network size. NA indicates non-applicable results.

Dataset	Metric (%)	TCD	CBNs	RADAR	Anomalous	ONE	PRN
Football	F-score ↑	**83.0** $\pm$ 2.8	12.2 $\pm$ 4.2	36.8 $\pm$ 3.7	16.1 $\pm$ 2.3	14.3 $\pm$ 2.6	24.5 $\pm$ 4.5
	misclustering rate ↓	**11.4** $\pm$ 2.6	19.5 $\pm$ 2.3	16.0 $\pm$ 3.1	23.1 $\pm$ 2.8	21.0 $\pm$ 3.3	40.1 $\pm$ 7.6
Polbooks	F-score ↑	**68.7** $\pm$ 3.2	18.7 $\pm$ 5.2	20.6 $\pm$ 3.2	15.4 $\pm$ 2.9	19.0 $\pm$ 3.7	30.2 $\pm$ 8.0
	misclustering rate ↓	31.0 $\pm$ 3.2	34.0 $\pm$ 4.2	27.7 $\pm$ 3.9	28.8 $\pm$ 3.2	31.0 $\pm$ 4.0	50.9 $\pm$ 9.5
Polblogs	F-score ↑	**54.7** $\pm$ 2.3	10.4 $\pm$ 3.0	12.2 $\pm$ 3.2	14.4 $\pm$ 2.1	19.6 $\pm$ 1.6	10.0 $\pm$ 3.2
	misclustering rate ↓	**13.3** $\pm$ 2.6	37.1 $\pm$ 4.2	32.9 $\pm$ 2.9	35.5 $\pm$ 2.2	33.9 $\pm$ 3.7	36.8 $\pm$ 5.0
BlogCata	F-score ↑	**43.6** $\pm$ 3.2	16.6 $\pm$ 4.2	11.2 $\pm$ 2.8	13.4 $\pm$ 3.9	12.6 $\pm$ 3.3	1.89 $\pm$ 0.2
	misclustering rate ↓	**35.3** $\pm$ 3.6	65.1 $\pm$ 5.1	43.9 $\pm$ 4.1	42.1 $\pm$ 3.8	43.9 $\pm$ 4.2	68.9 $\pm$ 8.7
Dataset	Metric (%)	HK	Extraction	ECoHeN	HCD	RCD	
Football	F-score ↑	26.4 $\pm$ 6.5	39.7 $\pm$ 5.3	7.7 $\pm$ 4.2	NA	NA	
	misclustering rate ↓	48.1 $\pm$ 7.3	46.2 $\pm$ 4.8	45.0 $\pm$ 13.0	56.2 $\pm$ 3.8	65.1 $\pm$ 5.2	
Polbooks	F-score ↑	31.1 $\pm$ 8.4	54.1 $\pm$ 8.7	4.4 $\pm$ 7.0	NA	NA	
	misclustering rate ↓	51.3 $\pm$ 16.6	56.7 $\pm$ 5.3	53.4 $\pm$ 1.1	56.0 $\pm$ 3.7	**21.1** $\pm$ 3.3	
Polblogs	F-score ↑	28.9 $\pm$ 13.8	22.6 $\pm$ 1.2	1.1 $\pm$ 0.6	NA	NA	
	misclustering rate ↓	43.9 $\pm$ 12.3	56.9 $\pm$ 1.8	49.2 $\pm$ 1.6	34.3 $\pm$ 4.7	26.5 $\pm$ 1.2	
BlogCata	F-score ↑	10.4 $\pm$ 2.3	16.7 $\pm$ 4.3	0.0 $\pm$ 0.0	NA	NA	
	misclustering rate ↓	64.1 $\pm$ 39.4	70.3 $\pm$ 13.2	82.3 $\pm$ 0.0	56.9 $\pm$ 6.2	64.1 $\pm$ 6.2	

6. Real Application

In this section, we evaluate the performance of our algorithm on some constructed semi-real data sets and a real data set.

6.1. Semi-Real Data

Since it is usually unknown to us which nodes are the ground truth scattered nodes in a real network data, we manually generate some scattered nodes and add them to some real networks. In this way, we create some semi-real data and we are able to evaluate the performance of TCD in detecting tight and scattered nodes in such data. Specifically, we consider four networks with ground truth community labels: (1) football (Girvan & Newman, 2002) with 115 nodes and 12 communities, (2) polbooks (Krebs, 2005) with 105 nodes and 3 communities, (3) polblogs (Adamic & Glance, 2005) with 1222 nodes and 2 communities, (4) BlogCatalog (Zafarani & Liu, 2009) with 5196 nodes and 6 communities. These datasets, with sizes ranging from 100 to 5000 and 2 to 12 communities, enable a comprehensive analysis across diverse network scale and community granularity.

When simulating scattered nodes, we considered the IER-type, i.e., each scattered node connects to any other nodes with a probability drawn from a uniform distribution $U(0, \rho_n)$, where ρ_n is the edge density of the original network. To thoroughly assess our method, we added varying proportions of scattered nodes to each network, i.e., one-sixth, one-fourth, and one-half of the original node size. Table 1 shows the results of adding scattered nodes with the size one-fourth of the original network. As we can see, TCD outperforms the other methods in most cases. Appendix C.2 shows some results of adding scattered nodes with the size one-sixth and one-half of the original network. By comparing Tables 1, 4, 5, we also observe that as the proportion of the scattered nodes increases, the F-scores generally increase in most cases, while the misclustering rate does not exhibit a clear trending pattern.

6.2. COVID-19 Protein-Protein Interaction Network

We constructed a real-world COVID-19 protein-protein interaction network, where each node represents a protein related to SARS-CoV-2 infection, and an edge is formed between two proteins according to the String database (Varusai et al., 2020). This network has 291 nodes and 4,380 edges.

We initially estimate the number of communities using a cross-validation technique proposed by (Chen & Lei, 2018), resulting in $\hat{K} = 2$. Subsequently, we applied TCD to this network to identify scattered nodes and to partition tight nodes into communities. We identified 37 scattered nodes, and 254 tight nodes clustered into two communities. Further details are shown in Table 6 in the Appendix, which lists protein names of tight nodes and scattered nodes. Figure 9 in the Appendix C.2 visualizes the protein network, where the detected scattered nodes are colored in red.The orange and blue dots represent the two communities, consisting of 206 and 48 tight nodes, respectively.

To elucidate the biological relevance of the identified node clusters and scattered nodes, we further conducted a KEGG pathway enrichment analysis. Given a set of proteins, we employed the hypergeometric test (Rivals et al., 2007) to determine if the representation of proteins within specific KEGG pathways is significantly greater than what random chance would predict. The enrichment analysis unveils that the tight nodes within the orange community are significantly associated with pathways including "Coronavirus disease - COVID-19", "Toll-like receptor signaling pathway", and "Hepatitis B". This suggests that proteins in the orange community play an important role in directing viral pathogenesis, innate immune response, and potentially shared response mechanisms across different viral infections. The tight nodes within the blue community showed significant enrichment in "Nucleocytoplasmic transport", "Amyotrophic lateral sclerosis", and "Spliceosome", indicating that these proteins play crucial roles in maintaining cellular function, mediating stress or damage responses due to infection, and influencing gene expression during the viral life cycle. Interestingly, the scattered nodes did not exhibit significant enrichment in any of the aforementioned pathways.

Furthermore, to enhance our evaluation, we introduced two quantitative metrics. (1) Normalized Cut (Ncut): A widely adopted metric that evaluates the strength of the identified community structure (Von Luxburg, 2007). A smaller Ncut value indicates the clusters are well separated from each other. (2) Overlapping enriched pathway ratio (OEPR), which is a novel metric proposed by us specifically for the biological context of PPI networks. In particular, for PPI network, we aim to find clutsers of proteins which share the same biological function, and find scattered proteins which do not share biological functions with tight nodes. OEPR measures the overlap between enriched KEGG pathways of scattered and tight proteins: OEPR $= \frac{\sum_{k=1}^{K} EP_S \cap EP_{T_k}}{\sum_{k=1}^{K} EP_{T_k}}$, where EP_{T_k} denotes the set of enriched KEGG pathways for proteins in the identified tight community k, $k = 1, \ldots, K$, EP_S denotes the enriched pathways for scattered proteins. Lower OEPR indicates less functional overlap between scattered and tight proteins, suggesting successful identification.

Table 2 shows, TCD achieves the best (lowest) Ncut value of 0.087, outperforming all other methods. This indicates that TCD is able to identify the tight and well-separated community structure in the PPI network. In addition, TCD has the best (lowest) OEPR value of zero, validating its effectiveness in discerning functionally distinct proteins. In comparison, other methods either have high Ncut value or high OEPR values. While the PRN method also reports an OEPR of zero, this is due to an artifact of the small number of scattered proteins that are not enriched in any KEGG pathway, rather than an ability to discern functionally distinct scattered proteins.

Table 2. PPI network results. Lower Ncut and OEPR values indicate better performance. For HCD and RCD, the OEPR is NA (non-applicable), as they cannot identify scattered nodes.

	TCD	CBNs	RADAR	Anomalous
Ncut	**0.087**	0.915	0.104	0.114
OEPR	**0.000**	0.286	0.667	0.167
	ONE	PRN	HK	Extraction
Ncut	0.109	0.110	0.915	0.251
OEPR	0.833	**0.000**	0.667	0.667
	ECoHeN	HCD	RCD	
Ncut	5.537	0.958	0.366	
OEPR	0.028	NA	NA	

7. Discussion

Despite the outstanding empirical performances of our method, it has several limitations. First, similar to many other spectral-based techniques, our approach may experience challenges when applied to extremely sparse graphs where the connectivity scales as $1/n$. In such extremely sparse settings, the spectral properties of the graph may not be as informative, and our co-membership-based approach could fail to accurately capture the underlying community structure. Second, although our empirical investigation showcases TCD's superior performances in both SBM and DCSBM settings, our theoretical framework is currently restricted to the SBM. Rigorous theoretical results under DCSBM setting are still under investigation. Third, similar to many other spectral clustering methods, our method requires a predefined number of communities.

There are several possible extensions to pursue following the proposed framework. One interesting direction is investigate how to generalize the current framework for the setting in which additional node attributes are available. In this case, the notion of a tight community could be extended such that nodes in the same community not only exhibit dense internal connectivity but also share similar attributes. Another interesting direction is to employ SBM variations to model the tight community structure. In Appendix C.1, we empirically found our current framework still achieves superior performance under degree-corrected SBM (DCSBM). It would be interesting to investigate theoretical properties under DCSBM or other SBM variants. Finally, the idea of distinguishing core and scattered structures could be generalized beyond community structures. For instance, exploring analogous structure decomposition in nonparametric models such as graphon poses an open question.

Impact Statement

This paper explores a new approach to community detection by strategically removing scattered nodes in observed networks. This innovative methodology significantly enhances the accuracy and interpretability of community structures in complex networks, providing valuable insights for researchers and practitioners in machine learning. Our work has broad practical applications, from improving social network analysis to optimizing resource allocation in various areas. By unveiling cohesive structures and eliminating noise introduced by scattered nodes, our research contributes to more efficient interventions and targeted decision-making, ultimately advancing the field of network science and machine learning with tangible, real-world impact.

Acknowledgements

This work was supported in part by the National Institute of Health of the USA (grant R01 GM152814-01), the National Natural Science Foundation of China (grant numbers 12071477, 71873137, and 72271232), the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001), and the Beijing Municipal Natural Science Foundation (grant number 1232019). The authors gratefully acknowledge the support of Public Computing Cloud at Renmin University of China and the Beijing Institute of Technology Research Fund Program for Young Scholars.

References

Abuzaid, A. H. Identifying density-based local outliers in medical multivariate circular data. *Statistics in Medicine*, 39(21):2793–2798, 2020.

Adamic, L. A. and Glance, N. The political blogosphere and the 2004 u.s. election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery (LinkKDD)*, pp. 36–43, 2005.

Aggarwal, C. C. and Aggarwal, C. C. Proximity-based outlier detection. *Outlier Analysis*, pp. 111–147, 2017.

Airoldi, E. M., Blei, D., Fienberg, S., and Xing, E. Mixed membership stochastic blockmodels. *Advances in Neural Information Processing Systems*, 21, 2008.

Andersen, R., Chung, F., and Lang, K. Local graph partitioning using pagerank vectors. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 475–486. IEEE, 2006.

Bandyopadhyay, S., Lokesh, N., and Murty, M. N. Outlier aware network embedding for attributed networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 12–19, 2019.

Bickel, P., Choi, D., Chang, X., and Zhang, H. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922 – 1943, 2013. doi: 10.1214/13-AOS1124.

Cai, T. T. and Li, X. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *The Annals of Statistics*, 43(3):1027–1059, 2015.

Celisse, A., Daudin, J.-J., and Pierre, L. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899, 2012.

Chen, J., Sathe, S., Aggarwal, C., and Turaga, D. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 90–98. SIAM, 2017.

Chen, K. and Lei, J. Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251, 2018.

Cheng, H., Wang, Y., Ma, P., and Murdie, A. Communities and brokers: How the transnational advocacy network simultaneously provides social power and exacerbates global inequalities. *International Studies Quarterly*, 65 (3):724–738, 2021.

Clauset, A., Newman, M. E., and Moore, C. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

Deng, J., Huang, D., Ding, Y., Zhu, Y., Jing, B., and Zhang, B. Subsampling spectral clustering for stochastic block models in large-scale networks. *Computational Statistics & Data Analysis*, 189:107835, 2024. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2023.107835.

Dey, A., Kumar, B. R., Das, B., and Ghoshal, A. K. Outlier detection in social networks leveraging community structure. *Information Sciences*, 634:578–586, 2023.

Feng, X., Yu, W., and Li, Y. Faster matrix completion using randomized svd. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 608–615. IEEE, 2018.

Gaucher, S., Klopp, O., and Robin, G. Outlier detection in networks with missing links. *Computational Statistics & Data Analysis*, 164:107308, 2021.

Gibbs, C. P., Fosdick, B. K., and Wilson, J. D. Ecohen: A hypothesis testing framework for extracting communities from heterogeneous networks. *arXiv preprint arXiv:2212.10513*, 2022.

Girvan, M. and Newman, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

Goodge, A., Hooi, B., Ng, S.-K., and Ng, W. S. Lunar: Unifying local outlier detection methods via graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6737–6745, 2022.

Guimera, R. and Nunes Amaral, L. A. Functional cartography of complex metabolic networks. *Nature*, 433(7028): 895–900, 2005.

Haggerty, L. S., Jachiet, P.-A., Hanage, W. P., Fitzpatrick, D. A., Lopez, P., O'Connell, M. J., Pisani, D., Wilkinson, M., Bapteste, E., and McInerney, J. O. A pluralistic account of homology: adapting the models to the data. *Molecular Biology and Evolution*, 31(3):501–516, 2014.

Hawkins, S., He, H., Williams, G., and Baxter, R. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pp. 170–180. Springer, 2002.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

Jin, J. Fast community detection by score. *The Annals of Statistics*, 43(1):57–89, 2015.

Karrer, B. and Newman, M. E. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83 (1):016107, 2011.

Kloster, K. and Gleich, D. F. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1386–1395, 2014.

Krebs, V. The political books network. 2005.

Kumar, A., Sabharwal, Y., and Sen, S. A simple linear time (1+/spl epsiv/)-approximation algorithm for k-means clustering in any dimensions. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 454–462. IEEE, 2004.

Lai, M.-J. and Mckenzie, D. Compressive sensing for cut improvement and local clustering. *SIAM Journal on Mathematics of Data Science*, 2(2):368–395, 2020.

Lai, M.-J. and Shen, Z. A compressed sensing based least squares approach to semi-supervised local cluster extraction. *Journal of Scientific Computing*, 94(3):63, 2023.

Latouche, P., Birmelé, E., and Ambroise, C. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Statistics*, 5(1): 309–336, 2011.

Lei, J. and Rinaldo, A. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, pp. 215–237, 2015.

Li, J., Dani, H., Hu, X., and Liu, H. Radar: Residual analysis for anomaly detection in attributed networks. In *IJCAI*, volume 17, pp. 2152–2158, 2017.

Li, T., Levina, E., and Zhu, J. Network cross-validation by edge sampling. *Biometrika*, 107(2):257–276, 2020.

Li, T., Lei, L., Bhattacharyya, S., Van den Berge, K., Sarkar, P., Bickel, P. J., and Levina, E. Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, 117(538):951–968, 2022.

Li, Y., He, K., Kloster, K., Bindel, D., and Hopcroft, J. Local spectral clustering for overlapping community detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):1–27, 2018.

Liu, X., Cheng, H.-M., and Zhang, Z.-Y. Evaluation of community detection methods. *IEEE Transactions on Knowledge and Data Engineering*, 32(9):1736–1746, 2019.

Mahoney, M. W., Orecchia, L., and Vishnoi, N. K. A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *The Journal of Machine Learning Research*, 13(1):2339–2365, 2012.

Martin, L., Loukas, A., and Vandergheynst, P. Fast approximate spectral clustering for dynamic networks. In *International Conference on Machine Learning*, pp. 3423–3432. PMLR, 2018.

Mukherjee, S. S., Sarkar, P., and Bickel, P. J. Two provably consistent divide-and-conquer clustering algorithms for large networks. *Proceedings of the National Academy of Sciences*, 118(44):e2100482118, 2021.

Newman, M. E. and Girvan, M. Finding and evaluating community structure in networks. *Physical Review E*, 69 (2):026113, 2004.

Paul, S. and Chen, Y. Spectral and matrix factorization methods for consistent community detection in multi-layer networks. *The Annals of Statistics*, 48(1):230–250, 2020.

Peng, Z., Luo, M., Li, J., Liu, H., Zheng, Q., et al. Anomalous: A joint modeling approach for anomaly detection on attributed networks. In *IJCAI*, pp. 3513–3519, 2018.

Ramaswamy, S., Rastogi, R., and Shim, K. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.

Rivals, I., Personnaz, L., Taing, L., and Potier, M.-C. Enrichment or depletion of a go category within a class of genes: which test? *Bioinformatics*, 23(4):401–407, 2007.

Rohe, K., Chatterjee, S., and Yu, B. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

Rousseeuw, P. J. and Hubert, M. Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79, 2011.

Rousseeuw, P. J. and Hubert, M. Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2):e1236, 2018.

Shen, Z., Lai, M.-J., and Li, S. Graph-based semi-supervised local clustering with few labeled nodes. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4190–4198, 2023.

Shrivastava, N., Majumder, A., and Rastogi, R. Mining (social) network graphs to detect random link attacks. In *2008 IEEE 24th International Conference on Data Engineering*, pp. 486–495. IEEE, 2008.

Tarjan, R. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

Tseng, G. C. and Wong, W. H. Tight clustering: A resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, 61(1):10–16, 2005.

Varusai, T., Haw, R., D'Eustachio, P., Jassal, B., Senff-Ribeiro, A., Orlic-Milacic, M., Stephan, R., Rothfels, K., and Gillespie, M. E. Sars-cov-2 infection, Sep 2020. URL https://doi.org/10.3180/R-HSA-9694516.1. Provided by Reactome. Citation Accessed on Wed Jan 24 2024.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

Xu, X., Yuruk, N., Feng, Z., and Schweiger, T. A. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 824–833, 2007.

Yamanishi, K. and Takeuchi, J.-i. Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 389–394, 2001.

Yang, X., Lin, B., and Sen, S. Fundamental limits of community detection from multi-view data: multi-layer, dynamic and partially labeled block models. *arXiv preprint arXiv:2401.08167*, 2024.

You, T., Cheng, H.-M., Ning, Y.-Z., Shia, B.-C., and Zhang, Z.-Y. Community detection in complex networks using density-based clustering algorithm and manifold learning. *Physica A: Statistical Mechanics and its Applications*, 464:221–230, 2016.

Zafarani, R. and Liu, H. Social computing data repository at ASU, 2009. URL http://socialcomputing.asu.edu.

Zhao, Y., Levina, E., and Zhu, J. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Zhao, Y., Levina, E., and Zhu, J. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

A. Preliminary Notations and Lemmas

In this section, we first provide a notation table to briefly introduce important notations in this manuscript. Then, we introduce four useful lemmas to be used in the proofs in Appendices B.

A.1. Details of Notation

We present the detailed expressions of notations widely used in the proposed model and algorithm in Table 3.

Table 3. Notations

Notations	Description
N	entire network size
n	number of tight nodes
m	number of scattered nodes
V	entire node set
E	entire edge set
$\mathcal{T}$	set of tight nodes
$\mathcal{S}$	set of scattered nodes
$\mathbf{A} \in \{0,1\}^{N\times N}$	adjacency matrix of entire network
K	underlying number of communities
$\tilde{K}$	predefined number of communities
$\mathbf{z} \in [K]^n$	ground truth community label
$\mathbf{B} \in (0,1)^{K\times K}$	community connectivity matrix
ζ	connectivity probability of scattered nodes
$\zeta_{\max}$	the upper bound of connectivity probability associated with scattered nodes
L	subsampling times
$\tilde{N}$	subsampling size
$\tilde{\mathbf{A}} \in \{0,1\}^{\tilde{N}\times N}$	adjacency matrix of subnetwork
$\hat{\mathcal{T}}$	set of estimated tight nodes
$\hat{\mathcal{S}}$	set of estimated scattered nodes
$\lvert\hat{\mathcal{T}}\rvert$	number of estimated tight nodes
$\hat{\mathbf{z}} \in [\tilde{K}]^{\lvert\hat{\mathcal{T}}\rvert}$	estimated community label
$\mathbf{C} \in \{0,1\}^{N\times N}$	co-membership matrix
$\bar{\mathbf{C}} \in (0,1)^{N\times N}$	the averaged co-membership matrix
α	tightness hyperparameter

A.2. Useful Lemmas

The following Lemma A.1 identifies an approximate low-rank structure for the sub-matrix $\mathbf{P}$, which supports the use of spectral clustering in our problem. We provide the detailed proofs in Appendix B.1.

Lemma A.1 (Structure of probability matrix)**.** *There exists* $\mathbf{R} \in \mathbb{R}^{K\times K}$ *with* $\|\mathbf{R}_{k\cdot} - \mathbf{R}_{l\cdot}\| = \sqrt{n_k^{-1} + n_l^{-1}}$ *for all* $1 \le k < l \le K$*, such that the eigen-decomposition of* $\widetilde{\mathbf{P}}$ *can be represented by* $(\mathbf{\Theta R})\mathbf{W}(\mathbf{\Theta R})^\top$*. Furthermore, we can decompose* $\mathbf{P}$ *by*

$$\mathbf{P} = \begin{pmatrix} \mathbf{\Theta R} \\ \mathbf{0}_{m\times K} \end{pmatrix} \mathbf{W} \begin{pmatrix} \mathbf{R}^\top \mathbf{\Theta}^\top & \mathbf{0}_{K\times m} \end{pmatrix} + \mathbf{D}, \tag{3}$$

with $\|\mathbf{D}\| \le C\xi\sqrt{m(m+n)}$*, where* C *is a constant.*

Below are three more useful lemmas in asserting the consistency of the spectral clustering algorithm. Lemma A.2 is a Wedin sinΘ theorem, addressing the error in right singular subspace when the matrix is perturbed. Readers can refer to Lemma 7 in (Chen & Lei, 2018) for a formal proof.

Lemma A.2 (Wedin sinΘ theorem). *Let $\mathbf{M}, \hat{\mathbf{M}}$ be two matrices of the same dimension, and $\mathbf{U}, \hat{\mathbf{U}}$ be two $n \times K$ orthonormal matrices corresponding to the top K right singular vectors respectively. Then there exists a $K \times K$ orthonormal matrix $\mathbf{Q}$ such that*

$$\|\hat{\mathbf{U}} - \mathbf{U}\mathbf{Q}\|_F \leq \frac{2\sqrt{2K}\|\hat{\mathbf{M}} - \mathbf{M}\|}{\sigma_K(\mathbf{M})}.$$

Lemma A.3 is a graph concentration inequality, asserting the error of $\|\mathbf{A} - \mathbf{P}\|$. This result firstly appeared as Theorem 5.2 in (Lei & Rinaldo, 2015) in the literature of network analysis. Here we are adopting a direct corollary where matrices are rectangular.

Lemma A.3 (Graph concentration inequality). *Let $\mathbf{A}$ be the adjacency matrix generated from a (corrupted) block model with $\mathbb{E}[\mathbf{A}] = \mathbf{P}$ and $\rho_n := \max_{ij} \mathbf{P}_{ij} \geq c \log n/n$ for a positive constant c. Let $\mathbf{A}^{(l)}$ be an arbitrary subset of rows of A and $\mathbf{P}^{(l)}$ be the corresponding submatrix of $\mathbf{P}$. We have, for some constant C,*

$$\mathbb{P}\left(\|\mathbf{A} - \mathbf{P}\| \leq C\sqrt{N\rho_N}\right) \geq 1 - \frac{1}{2N}.$$

Lemma A.4 asserts that an approximate k-means solution yields an approximately reliable clustering result. It is the same as Lemma 5.3 (Lei & Rinaldo, 2015) and Lemma 8 (Chen & Lei, 2018).

Lemma A.4 (Approximate k-means error bound). *Let $\hat{\mathbf{U}}$ and $\mathbf{U}$ be two $n \times K$ matrices. Let δ be the minimum distance between two distinct rows of $\mathbf{U}$, and $\mathbf{\Theta}$ be the membership vector given by clustering the rows of $\mathbf{U}$. Let $\hat{\mathbf{\Theta}}$ be the output of a k-means clustering algorithm on $\hat{\mathbf{U}}$, with objective value no larger than a constant factor of the global optimum. Assume that $\|\hat{\mathbf{U}} - \mathbf{U}\|_F \leq Cn\delta$ for some small enough constant C. Then $\hat{\mathbf{\Theta}}$ agrees with $\mathbf{\Theta}$ on all but $C^{-1}\|\hat{\mathbf{U}} - \mathbf{U}\|_F\delta^{-1}$ nodes after an appropriate label permutation.*

B. Proof of Theoretical Guarantees

In Appendix B, we first present the proof of Lemma 4.5 in Appendix B.1. Then, we provide the proof of Theorem 4.6 in Appendices B.2, respectively.

B.1. Proof of Lemma 4.5

Proof. To prove the claims in Lemma 4.5, we first provide detailed proofs for Lemma A.1. By the definition of a K-community stochastic block model, each entry of $\widetilde{\mathbf{P}}$ satisfies $\widetilde{\mathbf{P}}_{ij} = \mathbf{B}_{z_i z_j}$. Therefore, we find $\widetilde{\mathbf{P}} = \mathbf{\Theta}\mathbf{B}\mathbf{\Theta}^\top$ is of rank K. Let $\Delta = \operatorname{diag}\left(\sqrt{n_1}, \ldots, \sqrt{n_K}\right)$, then

$$\widetilde{\mathbf{P}} = \mathbf{\Theta}\mathbf{B}\mathbf{\Theta}^\top = \mathbf{\Theta}\Delta^{-1}(\Delta\mathbf{B}\Delta)\Delta^{-1}\mathbf{\Theta}^\top,$$

where $\mathbf{\Theta}\Delta^{-1}$ already has orthonormal column vectors. Let $\widetilde{\mathbf{U}}\mathbf{W}\widetilde{\mathbf{U}}^\top$ be the eigen-decomposition of $\Delta\mathbf{B}\Delta$. As a result, orthonormal $n \times K$ matrix $\mathbf{R} := \Delta^{-1}\widetilde{\mathbf{U}}$ serves as the eigenvector matrix of $\widetilde{\mathbf{P}}$. Taking

$$\mathbf{D} = \begin{pmatrix} \mathbf{0}_{n\times n} & \mathbf{D}_1 \\ \mathbf{D}_1^\top & \mathbf{D}_2 \end{pmatrix},$$

we can arrive at the probability matrix decomposition (3). Moreover, $\|\mathbf{D}\| \leq C\xi\sqrt{m(m+n)}$ directly follows from that $\xi = \max\{\|\mathbf{D}_1\|_\infty, \|\mathbf{D}_2\|_\infty\}$.

To this end, we can use the conclusions in Lemma A.1 to demonstrate Lemma 4.5. Note that Lemma 4.5 is a direct corollary of Lemma A.1 as long as firstly using $|\lambda_K(\boldsymbol{P}) - \lambda_K(\boldsymbol{P} - \boldsymbol{D})| \leq \|\mathbf{D}\|$ and then there exists constant C_1 and C_2 such that

$$\frac{\lambda_K(\boldsymbol{P})}{\lambda_{\max}(\mathbf{D})} \geq \frac{\lambda_{\min}(\mathbf{W}) - \lambda_{\max}(\mathbf{D})}{C_1\xi\sqrt{m(m+n)}} \geq C_2\frac{n\rho_n}{\xi\sqrt{m(m+n)}} - 1 = \Omega\left(\sqrt{n\rho_n}\right).$$

□

B.2. Proof of Theorem 4.6

Proof. Throughout this proof, the probability, the expectation and the variance are all conditioned on the event $\mathcal{E} = \left\{\|\mathbf{A}-\mathbf{P}\| \le C\sqrt{N\rho_N}\right\}$ which happens with probability at least $1-1/(2N)$ by Lemma A.3.

During each repetition of our sub-sampling procedure, a spectral clustering algorithm is applied to a sub-rectangle matrix $\mathbf{A}^{(l)} \in \{0,1\}^{\tilde{N}\times N}$ with only $\tilde{N}$ randomly selected rows. The first step is to extract the top-K right singular vectors $\hat{\mathbf{U}}^{(l)} \in \mathbb{R}^{N\times K}$ of $\mathbf{A}^{(l)}$. According to Lemma A.1, $\mathbf{P} = \mathbf{P}_1 + \mathbf{D}$ is decomposed into two components where the top-K right singular vectors

$$\mathbf{U} = \begin{pmatrix} \mathbf{\Theta R} \\ \mathbf{0}_{m\times K} \end{pmatrix}$$

of $\mathbf{P}_1$ encodes all membership information of tight nodes. We are able to effectively control the discrepancy between $\mathbf{U}$ and $\hat{\mathbf{U}}^{(l)}$ as shown below. Lemma A.2 yields,

$$\begin{aligned}\min_{\mathbf{Q}\in\mathbb{O}_K} \|\hat{\mathbf{U}}^{(l)} - \mathbf{UQ}\|_F &\le \frac{2\sqrt{2K}}{\sigma_K(\mathbf{P}_1^{(l)})}\left\|\mathbf{A}^{(l)} - \mathbf{P}_1^{(l)}\right\| \\ &\le \frac{2C\sqrt{2K}}{n\rho_n}\left(\left\|\mathbf{A}^{(l)} - \mathbf{P}^{(l)}\right\| + \left\|\mathbf{P}^{(l)} - \mathbf{P}_1^{(l)}\right\|\right), \end{aligned} \tag{4}$$

where we used $\sigma_K(\mathbf{P}_1^{(l)}) \ge Cn\rho_n$ for some constant $C > 0$. Subsequently, Lemma A.1 helps to bound

$$\left\|\mathbf{P}^{(l)} - \mathbf{P}_1^{(l)}\right\| \le \|\mathbf{D}\| \le C\xi\sqrt{m(m+n)}.$$

Conditioned on event $\mathcal{E}$,

$$\|\mathbf{A}^{(l)} - \mathbf{P}^{(l)}\| \le \|\mathbf{A}-\mathbf{P}\| \le C\sqrt{(m+n)(\rho_n+\xi)}.$$

Equipped with these two upper bounds, continue from equation 4 to find a constant C_1 such that

$$\min_{\mathbf{Q}\in\mathbb{O}_K} \|\hat{\mathbf{U}}^{(l)} - \mathbf{UQ}\|_F \le \frac{C_1\sqrt{K}}{\sqrt{n\rho_n}},$$

where we already adopt $\xi = O(\sqrt{\rho_n/m})$ and $m \le n$. To this end, a clustering problem is posted on the row vectors of $\hat{\mathbf{U}}^{(l)}$,

$$\min \left\|\mathbf{H}^{(l)}\mathbf{R}^{(l)} - \hat{\mathbf{U}}^{(l)}\right\|_F^2, \quad \text{s.t.}\mathbf{H}^{(l)} \in \mathbb{M}_{N,K}, \mathbf{R}^{(l)} \in \mathbb{R}^{K\times K},$$

where $\mathbb{M}_{N,K}$ denote a set of membership matrix. Although solving an exact solution is NP-hard in nature, it suffices to obtain an $(1+\epsilon)$-approximate solution $(\hat{\mathbf{H}}^{(l)}, \hat{\mathbf{X}}^{(l)})$,

$$\left\|\hat{\mathbf{H}}^{(l)}\hat{\mathbf{X}}^{(l)} - \hat{\mathbf{U}}^{(l)}\right\|_F^2 \le (1+\epsilon)\min_{\mathbf{H},\mathbf{R}}\left\|\mathbf{H}^{(l)}\mathbf{R}^{(l)} - \hat{\mathbf{U}}^{(l)}\right\|_F^2,$$

which is guaranteed to be feasible by (Kumar et al., 2004). Based on $\hat{\mathbf{H}}^{(l)}$, define the estimated membership matrix for tight nodes as $\hat{\mathbf{\Theta}}^{(l)} = (\hat{\mathbf{H}}^{(l)}_{ij})_{1\le i,j\le n}$. Lastly, applying Lemma A.4, we know that the K-means clustering algorithm misclusters no more than C/ρ_n nodes, i.e.

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}(\mathbf{\Theta}_i \ne \hat{\mathbf{\Theta}}_i^{(l)}) \le \frac{C}{n\rho_n}.$$

These estimated labels give reliable co-membership matrix estimates $\hat{\mathbf{C}}_{ij}^{(l)} = \mathbb{1}(\hat{\mathbf{\Theta}}_i^{(l)} = \hat{\mathbf{\Theta}}_j^{(l)})$ whose error is bounded by

$$\frac{2}{n(n-1)}\sum_{1\le i<j\le n}\left|\mathbf{C}_{ij} - \hat{\mathbf{C}}_{ij}^{(l)}\right| \le 1 - \left(1-\frac{C}{n\rho_n}\right)^2 = O\left(\frac{1}{n\rho_n}\right).$$

This bound holds for any sub-sampled adjacency matrix $\mathbf{A}^{(l)}$. Therefore, while averaging over all repetitions

$$\bar{\mathbf{C}}_{ij} = \frac{1}{L}\sum_{l=1}^{L}\hat{\mathbf{C}}_{ij}^{(l)},$$

the error of the averaged co-membership matrix estimator is given by

$$\begin{aligned}&\frac{2}{n(n-1)}\sum_{1\le i<j\le n}\left|\mathbf{C}_{ij}-\bar{\mathbf{C}}_{ij}\right|\\&\le\frac{1}{L}\sum_{l=1}^{L}\frac{2}{n(n-1)}\sum_{1\le i<j\le n}\left|\mathbf{C}_{ij}-\hat{\mathbf{C}}_{ij}^{(l)}\right|=O\left(\frac{1}{n\rho_n}\right),\end{aligned}$$

therefore yielding the first assertion of Theorem 4.6. For the second assertion on conditional variance, we need to condition on the event $\mathcal{E}$,

$$\begin{aligned}&\frac{2}{n(n-1)}\sum_{1\le i<j\le n}\mathrm{Var}(\bar{\mathbf{C}}_{ij}\mid\mathcal{E})\\&=\frac{2}{Ln(n-1)}\sum_{1\le i<j\le n}\mathrm{Var}(\hat{\mathbf{C}}_{ij}^{(1)}\mid\mathcal{E})\\&=\frac{2}{Ln(n-1)}\sum_{1\le i<j\le n}\mathbb{E}\left\{\left[\hat{\mathbf{C}}_{ij}^{(1)}-\mathbb{E}(\hat{\mathbf{C}}_{ij}^{(1)}\mid\mathcal{E})\right]^2|\mathcal{E}\right\}\\&\le\frac{4}{Ln(n-1)}\sum_{1\le i<j\le n}\mathbb{E}\left\{\left|\hat{\mathbf{C}}_{ij}^{(1)}-\mathbb{E}(\hat{\mathbf{C}}_{ij}^{(1)}\mid\mathcal{E})\right||\mathcal{E}\right\}\\&\le\frac{8}{Ln(n-1)}\sum_{1\le i<j\le n}\mathbb{E}\left[\left|\hat{\mathbf{C}}_{ij}^{(1)}-\mathbf{C}_{ij}\right|\mid\mathcal{E}\right]=O\left(\frac{1}{Ln\rho_n}\right).\end{aligned}$$

□

C. Additional Experimental Results

In this section, we first conduct extensive synthetic studies on hyperparameter selection (see in Appendix C.1). Subsequently, we evaluate the performance of the TCD algorithm for networks generated from GSBM with ER-type and HetD-type scattered nodes. Additionally, we investigate the DCSBM in Appendix C.1. Finally, further results on real data analysis are presented in Appendix C.2.

C.1. Experiments on Synthetic Datasets

We perform a sensitivity analysis on hyperparameters, including subsample size $\tilde{N}$, sub-sampling times L, and the tightness parameter α. The experimental settings align with Scenario 1, keeping the entire network size fixed at $N = 1,000$ while varying the corresponding hyperparameter. The results are shown in Figure 5. Moreover, we also examine the estimation accuracy of the cross-validation method when applied to determine the number of clusters K. The identification results are displayed in Figure 6.

Subsequently, we evaluate the performance of the proposed TCD algorithm concerning ER-type and HetD-type scattered nodes. In the case of ER-type scattered nodes, we set $P(a_{ij} = 1) = \zeta = 0.1$ for $i \in \mathcal{S}$. For HetD-type scattered nodes, we define $P(a_{ij} = 1)$ to follow a uniform distribution with $U(0, \zeta^i_{\max})$, where $i \in \mathcal{S}$. Additionally, we specify $(\zeta^i_{\max})$ as $\zeta^i_{\max} = 0.1 + 0.8(i-1)/(m-1)$. The other parameter settings are kept the same as Scenario 1. Then, we increase the number of scattered nodes m from 20 to 200, while keeping the total number of network nodes fixed at $N = 1,000$. The simulation results are shown in Figure 7. The results demonstrate that the proposed TCD algorithm can effectively identify both ER-type and HetD-type scattered nodes, showcasing superior performance when compared with other approaches.

Additionally, to visually demonstrate the phenomenon of scattered nodes being randomly distributed across any community, we present the averaged co-membership matrix obtained by generating 100 sub-adjacency matrices and applying spectral clustering. The parameter settings align with those in the manuscript's example section. The experimental results are illustrated in Figure 8.

Based on existing literature (Lei & Rinaldo, 2015; Jin, 2015), the TCD algorithm can be easily extended to the degree-corrected stochastic block model (DCSBM) by modifying Step 1.2 as follows:

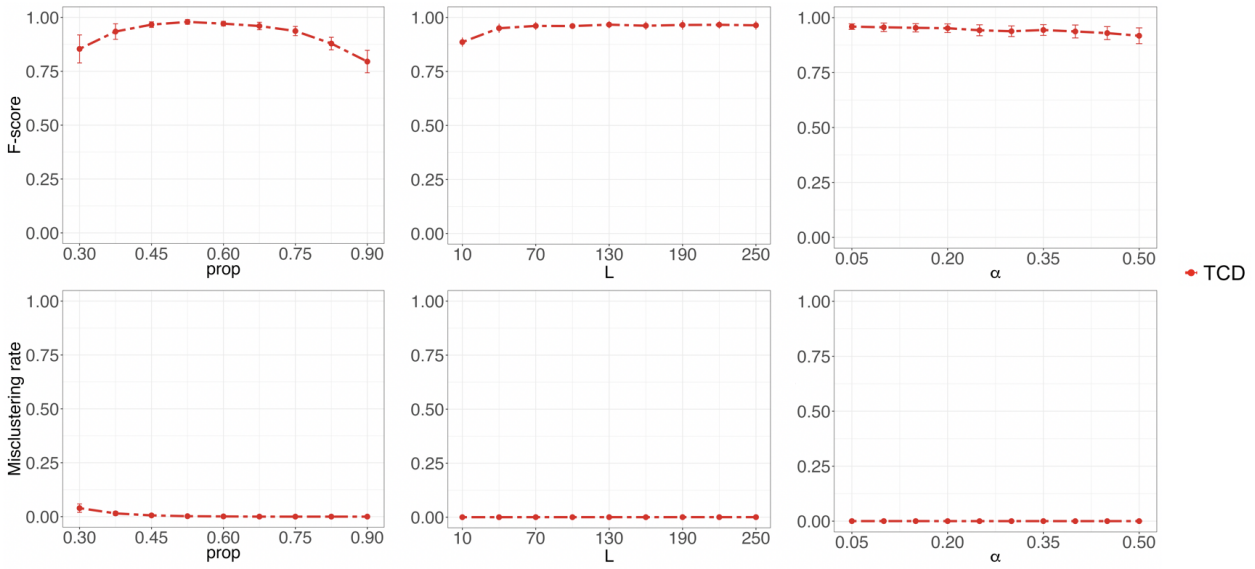

Figure 5. The performance of TCD under varying hyperparameters is presented. In the first column, results are displayed for varying the proportion of sub-sampling size, i.e., $\tilde{N} = N\text{prop}$, while keeping N fixed at $1,000$. The second column illustrates the impact of sub-sampling times L, which ranges from 10 to 250. The third column shows the variation in the tightness hyperparameter in TCD, ranging from 0.05 to 0.5.

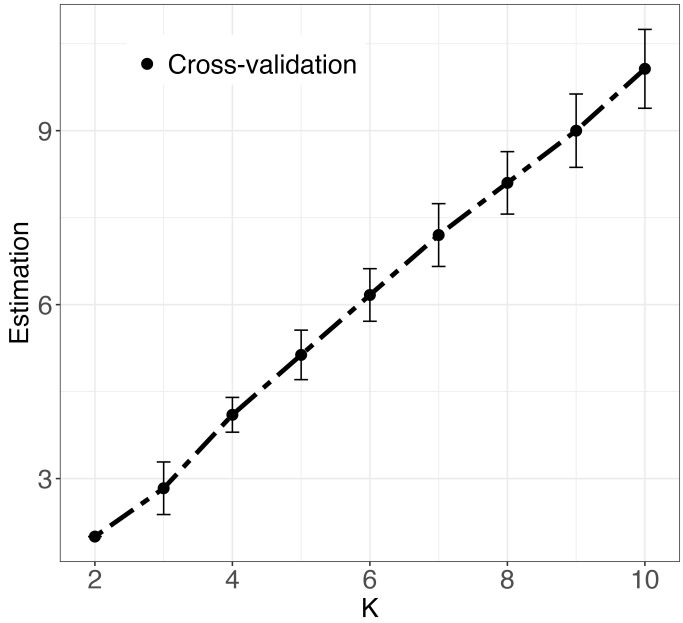

Figure 6. Estimating the number of clusters using a cross-validation approach (Chen & Lei, 2018). The x-axis represents the true underlying number of clusters, while the y-axis illustrates the results obtained through the cross-validation method. Error bars, derived from 100 repetitions, are also incorporated to provide a measure of variability in the estimation process.

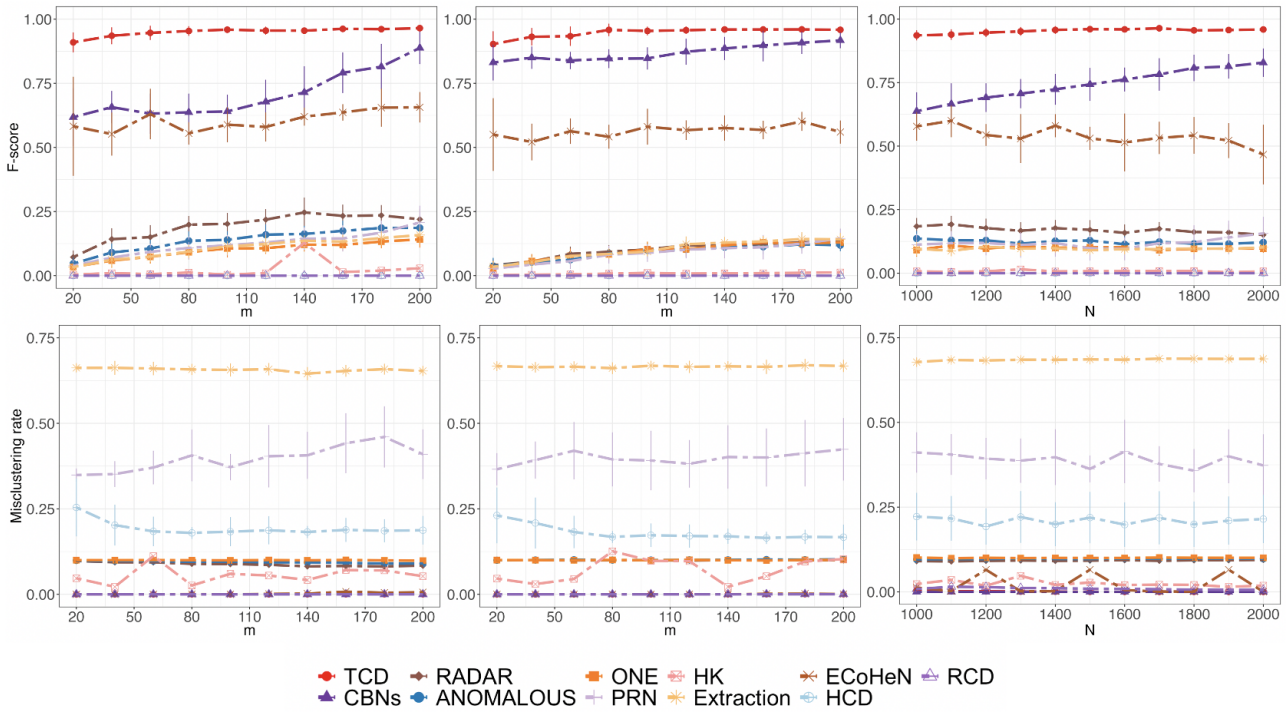

Figure 7. Simulation results for networks generated from GSBM with ER-type scattered nodes (first column) and GSBM with HetD-type scattered nodes (second column) are presented. The performance of each algorithm is further compared under DCSBM (third column).

Step 1.2. Intermediate community detection: Perform SVD on $\tilde{\mathbf{A}}^{(l)}$ to acquire its leading $\tilde{K}$ right singular vectors. Normalize each row of these singular vectors to unit length. Subsequently, employ k-means clustering on the rows of the normalized singular vectors to estimate $\tilde{z}^{(l)}$.

Furthermore, the performance under DCSBM is further investigated. Consider a DCSBM for tight nodes, where each tight node allows for a specific connectivity strength. Let ψ_i denote the degree parameter. For each $i, j \in \mathcal{T}$, $P(a_{ij} = 1) = \psi_i \boldsymbol{B}_{z_i z_j} \psi_j$. We then investigate the performance of TCD algorithm in a DCSBM. Specifically, under the settings in Scenario 1, we fix $N = 1,000$, $m = 0.1N$, $p = 0.3$, $q = 0.12$, and $\zeta_{\max} = 0.2$. According to (Zhao et al., 2012), we define $P(\psi_i = \delta x) = P(\psi_i = x) = 1/2$, with $x = 2/(\delta + 1)$. We set $\delta = 2$ in this experiment. The other parameter settings remain the same as in Scenario 1, while the network size varies from 1,000 to 2,000. The simulation results are shown in the third columns of Figure 7.

C.2. Experiments on Real Data Analysis

Here, we present more detailed results from the real data analysis, as shown in Tables 4-6 and Figure 9. Specifically, Tables 4 and 5 show the performance of each compared algorithm on simi-real data with scattered nodes, with sizes one-sixth and one-half of the original node size, respectively. Additionally, Figure 9 shows the identification results by TCD in the Covid-19 protein-protein interaction network, and Table 6 presents the nodes in each community, along with details of the scattered nodes.

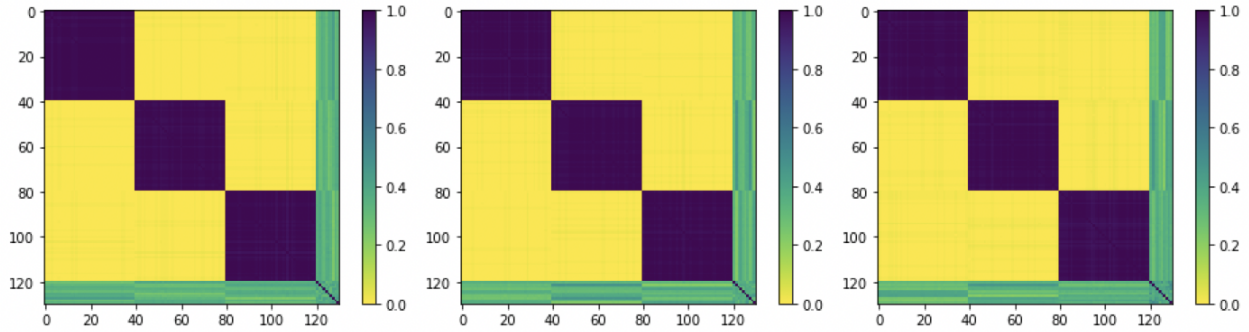

Figure 8. Averaged co-membership matrix for GSBM with ER (first column), IER (second column), and HetD (third column) scattered nodes, with 100 repetition results.

Table 4. Results (mean $\pm$ *std*) on benchmark datasets with scattered nodes scaled to one-sixth of original network size. NA indicates non-applicable results.

Dataset	Metric (%)	TCD	CBNs	RADAR	Anomalous	ONE	PRN
Football	F-score ↑	**77.3** $\pm$ 2.6	21.9 $\pm$ 4.1	27.6 $\pm$ 3.9	12.4 $\pm$ 2.9	14.1 $\pm$ 3.2	16.9 $\pm$ 5.3
	misclustering rate ↓	**11.6** $\pm$ 2.3	15.2 $\pm$ 2.7	15.4 $\pm$ 3.0	17.7 $\pm$ 2.9	17.1 $\pm$ 3.0	38.3 $\pm$ 7.9
Polbooks	F-score ↑	**60.6** $\pm$ 4.1	20.2 $\pm$ 4.9	34.5 $\pm$ 4.2	13.9 $\pm$ 3.9	13.3 $\pm$ 3.3	23.7 $\pm$ 6.9
	misclustering rate ↓	28.0 $\pm$ 2.9	31.3 $\pm$ 4.2	25.0 $\pm$ 3.6	26.2 $\pm$ 4.1	25.2 $\pm$ 3.9	45.9 $\pm$ 9.1
Polblogs	F-score ↑	**48.7** $\pm$ 3.2	12.4 $\pm$ 3.0	15.4 $\pm$ 3.5	11.1 $\pm$ 2.9	15.7 $\pm$ 3.6	6.41 $\pm$ 2.1
	misclustering rate ↓	**19.9** $\pm$ 3.2	39.1 $\pm$ 3.5	31.6 $\pm$ 4.2	34.9 $\pm$ 3.9	32.3 $\pm$ 4.5	36.2 $\pm$ 6.2
BlogCata	F-score ↑	**33.6** $\pm$ 3.1	18.2 $\pm$ 3.9	13.1 $\pm$ 2.9	15.2 $\pm$ 3.1	13.2 $\pm$ 3.2	1.70 $\pm$ 0.15
	misclustering rate ↓	**34.2** $\pm$ 3.7	68.2 $\pm$ 4.2	46.9 $\pm$ 4.2	40.1 $\pm$ 5.2	45.9 $\pm$ 3.9	66.2 $\pm$ 7.9
Dataset	Metric (%)	HK	Extraction	ECoHeN	HCD	RCD	
Football	F-score ↑	19.5 $\pm$ 5.8	35.6 $\pm$ 4.1	14.6 $\pm$ 20.7	NA	NA	
	misclustering rate ↓	38.3 $\pm$ 7.5	43.2 $\pm$ 3.7	37.0 $\pm$ 14.1	52.4 $\pm$ 3.1	64.4 $\pm$ 3.7	
Polbooks	F-score ↑	17.4 $\pm$ 6.9	32.6 $\pm$ 8.9	1.8 $\pm$ 3.0	NA	NA	
	misclustering rate ↓	53.0 $\pm$ 23.1	53.4 $\pm$ 6.1	45.5 $\pm$ 10.3	52.0 $\pm$ 5.2	**23.8** $\pm$ 2.7	
Polblogs	F-score ↑	16.1 $\pm$ 6.9	17.3 $\pm$ 1.4	3.6 $\pm$ 1.9	NA	NA	
	misclustering rate ↓	39.2 $\pm$ 17.4	56.1 $\pm$ 1.3	50.3 $\pm$ 1.5	21.8 $\pm$ 0.5	30.3 $\pm$ 4.2	
BlogCata	F-score ↑	12.4 $\pm$ 3.6	11.3 $\pm$ 2.4	0.1 $\pm$ 0.2	NA	NA	
	misclustering rate ↓	67.4 $\pm$ 28.6	69.2 $\pm$ 9.8	82.3 $\pm$ 0.0	52.1 $\pm$ 7.2	61.3 $\pm$ 4.2	

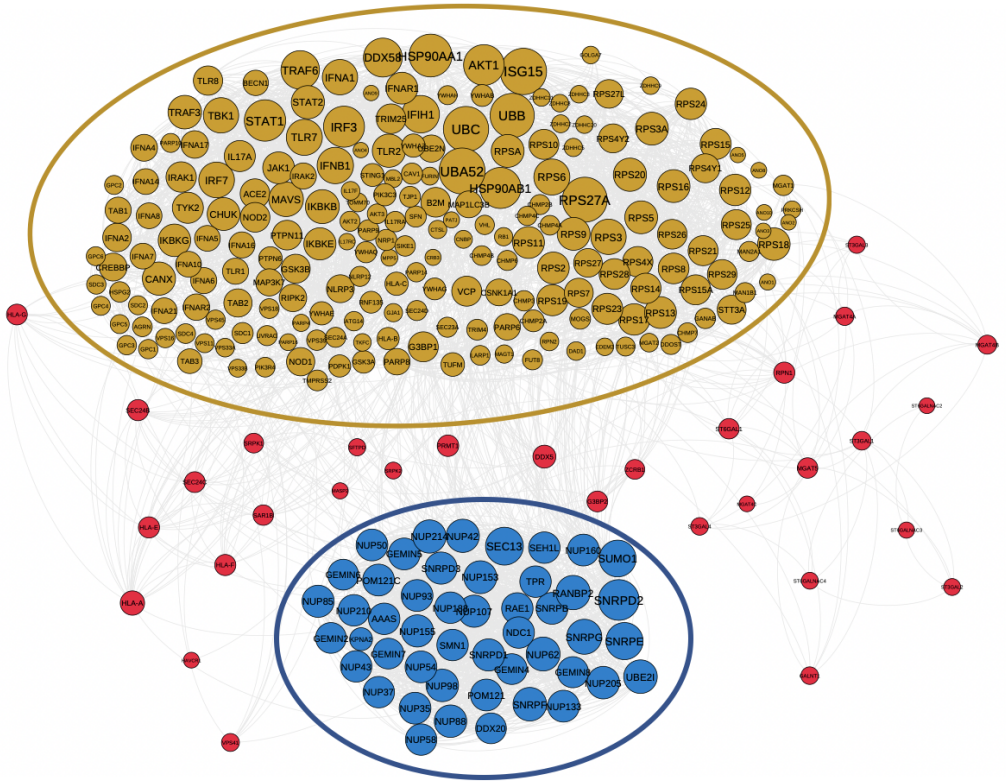

Figure 9. Clustering results for the COVID-19 protein-protein interaction network. The scattered nodes are labeled in red, while the tight nodes are partitioned into two clusters indicated by yellow and blue, respectively.

Table 5. Results (mean $\pm$ *std*) on benchmark datasets with scattered nodes scaled to one-half of original network size. NA indicates non-applicable results.

Dataset	Metric (%)	TCD	CBNs	RADAR	Anomalous	ONE	PRN
Football	F-score ↑	**82.9** $\pm$ 2.3	14.8 $\pm$ 3.1	28.7 $\pm$ 3.3	16.7 $\pm$ 3.2	17.9 $\pm$ 3.2	24.2 $\pm$ 3.8
	misclustering rate ↓	**10.4** $\pm$ 2.2	18.8 $\pm$ 2.9	13.5 $\pm$ 3.5	21.0 $\pm$ 3.7	18.4 $\pm$ 2.9	39.7 $\pm$ 6.1
Polbooks	F-score ↑	**75.0** $\pm$ 3.3	9.06 $\pm$ 2.9	20.8 $\pm$ 3.3	13.5 $\pm$ 3.2	17.4 $\pm$ 3.2	35.1 $\pm$ 8.3
	misclustering rate ↓	34.3 $\pm$ 3.7	36.6 $\pm$ 4.7	25.6 $\pm$ 3.8	26.4 $\pm$ 3.5	28.2 $\pm$ 4.4	47.3 $\pm$ 6.5
Polblogs	F-score ↑	**55.2** $\pm$ 3.1	10.7 $\pm$ 2.6	18.8 $\pm$ 3.1	14.9 $\pm$ 2.9	19.0 $\pm$ 3.2	16.7 $\pm$ 4.4
	misclustering rate ↓	**13.3** $\pm$ 2.6	34.9 $\pm$ 4.3	37.7 $\pm$ 3.9	38.9 $\pm$ 4.2	37.0 $\pm$ 4.1	39.2 $\pm$ 4.5
BlogCata	F-score ↑	**50.4** $\pm$ 3.5	20.6 $\pm$ 4.2	16.2 $\pm$ 4.8	16.4 $\pm$ 4.6	17.6 $\pm$ 4.1	3.69 $\pm$ 0.3
	misclustering rate ↓	**37.3** $\pm$ 3.9	62.2 $\pm$ 5.2	45.9 $\pm$ 4.6	46.1 $\pm$ 4.9	42.2 $\pm$ 5.2	57.3 $\pm$ 6.4
Dataset	Metric (%)	HK	Extraction	ECoHeN	HCD	RCD	
Football	F-score ↑	30.6 $\pm$ 4.6	41.2 $\pm$ 5.1	3.0 $\pm$ 3.2	NA	NA	
	misclustering rate ↓	43.3 $\pm$ 6.3	45.8 $\pm$ 4.9	53.7 $\pm$ 14.3	60.5 $\pm$ 2.1	64.2 $\pm$ 4.3	
Polbooks	F-score ↑	37.3 $\pm$ 8.4	56.2 $\pm$ 8.9	5.2 $\pm$ 10.3	NA	NA	
	misclustering rate ↓	54.6 $\pm$ 11.8	53.3 $\pm$ 7.9	48.7 $\pm$ 8.3	60.9 $\pm$ 5.1	**24.4** $\pm$ 9.8	
Polblogs	F-score ↑	34.5 $\pm$ 20.1	25.7 $\pm$ 0.8	1.4 $\pm$ 1.0	NA	NA	
	misclustering rate ↓	42.9 $\pm$ 8.0	57.0 $\pm$ 1.5	50.7 $\pm$ 1.8	35.1 $\pm$ 8.8	26.9 $\pm$ 6.8	
BlogCata	F-score ↑	8.38 $\pm$ 5.9	19.8 $\pm$ 6.2	0.5 $\pm$ 0.4	NA	NA	
	misclustering rate ↓	70.6 $\pm$ 29.4	73.2 $\pm$ 16.3	82.3 $\pm$ 0.0	59.2 $\pm$ 9.1	66.5 $\pm$ 8.5	

Table 6. Experiment results of the SARS-CoV-2 human protein-protein interaction network.

community 1	AAAS	SMN1	GEMIN6	DDX20	GEMIN2	GEMIN4	GEMIN7	SNRPB
	NUP50	RAE1	TPR	NUP153	NUP58	NUP42	NUP62	NUP160
	NUP35	NUP155	NUP214	NUP88	NDC1	SEH1L	RANBP2	KPNA2
	GEMIN8	GEMIN5	SUMO1	SNRPD3	SNRPD1	SNRPF	SNRPG	SNRPE
	SEC13	NUP37	NUP133	NUP98	NUP54	NUP107	NUP210	NUP43
	SNRPD2	POM121C	POM121	UBE2I	NUP188	NUP93	NUP85	NUP205
community 2	ACE2	IFNA7	TLR2	MPP5	NRP1	IFNAR1	IFNA1	TOMM70
	IFNA10	IFNA14	IFNA21	IFNB1	TLR7	IRF7	TMPRSS2	TYK2
	CANX	GPC1	GPC2	SDC2	SDC3	GPC4	SDC4	HSPG2
	YWHAE	RB1	RPS27A	GJA1	NOD2	YWHAG	STAT2	GSK3B
	CHUK	YWHAB	RPS6	YWHAQ	AKT2	TRAF3	YWHAZ	RPSA
	ANO10	ANO2	ANO3	ANO4	ANO9	ANO5	VPS18	TAB3
	TUFM	RPN2	DAD1	TUSC3	MGAT1	UBB	MAN2A1	MAGT1
	UBA52	VPS11	VPS45	CHMP4A	RIPK2	NOD1	IRAK2	TRIM25
	RPS5	RPS4X	RPS16	RPS8	RPS7	RPS27	RPS19	RPS24
	LARP1	RPS2	RPS10	GOLGA7	ZDHHC2	ZDHHC11	ZDHHC5	ZDHHC3
	RPS4Y2	PARP16	PARP4	PARP6	VPS41	VPS33B	RPS21	RPS29
	TLR8	NLRP3	CTSL	TJP1	STAT1	HSP90AB1	MBL2	DDX58
	ISG15	IFNA4	MAVS	JAK1	IFNA5	AKT1	IFNA17	IFIH1
	GPC6	GPC5	GPC3	SDC1	TAB1	GSK3A	ATG14	YWHAH
	TBK1	HSP90AA1	CAV1	SFN	PDPK1	UVRAG	PIK3R4	VCP
	PTPN6	IKBKB	TRAF6	UBC	B2M	MAP1LC3B	PTPN11	AKT3
	CHMP3	VPS39	VPS33A	VPS16	SEC24D	SEC23A	TAB2	HLA-C
	STT3A	MAN1B1	DDOST	PRKCSH	MOGS	GANAB	CHMP2A	CHMP4B
	RNF135	TRIM4	IKBKE	RPS3	RPS3A	RPS13	RPS12	RPS17
	RPS28	RPS15A	RPS14	RPS20	G3BP1	IFNAR2	SIKE1	TKFC
	ZDHHC9	ZDHHC20	ZDHHC8	PARP8	PARP10	RPS26	IL17F	IL17RA
	IFNA8	IFNA2	IFNA6	IFNA16	RPS9	SEC24A	RPS18	UBE2N
	IRF3	IL17A	FURIN	AGRN	CHMP2B	CHMP4C	CHMP6	CHMP7
	VHL	CSNK1A1	PIK3C3	CREBBP	RPS11	RPS15	RPS23	RPS25
	TLR1	BECN1	MAP3K7	IRAK1	NLRP12	PARP14	STING1	ST6GAL1
	IKBKG	PARP9	ANO1	ANO8	RPS4Y1	RPS27L		
Scattered Nodes Set	SFTPD	HLA-A	MASP2	DDX5	SRPK1	SRPK2	PRMT1	HLA-F
	ST3GAL1	FUT8	MGAT4A	GALNT1	MGAT5	MGAT2	MGAT4B	ST3GAL2
	SEC24C	HLA-E	HLA-G	SAR1B	HLA-B	SEC24B	HAVCR1	RPN1
	ST3GAL3	ST3GAL4	G3BP2	IL17RC	ZCRB1	MGAT4C	ST6GALNAC3	ST6GALNAC4
	EDEM2	CNBP	CRB3	PATJ	ST6GALNAC2			

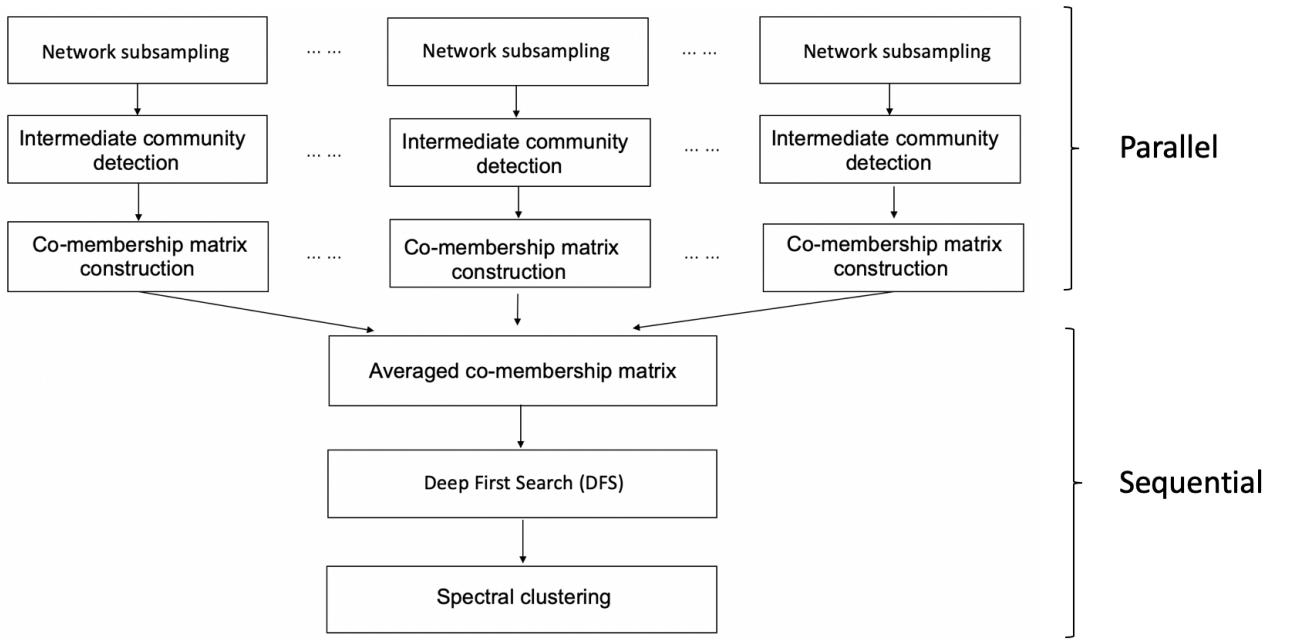

Figure 10. Framework of parallel tight community detection (PTCD) procedure.

D. Scalability of the Tight Community Detection Algorithm

In this section, we first discuss the computational complexity of TCD algorithm. Specifically, Step 1.1 involves a network sub-sampling procedure, which requires $O(N)$ computation. In Step 1.2 and Step 4, we employed a fast version of spectral clustering (Feng et al., 2018; Martin et al., 2018), which only requires a computational effort of $O(N^2)$. Step 3 involves a depth-first search algorithm, which requires $O(N + |E|)$ computation, with $|E|$ representing the number of edges, and typically, $|E| \ll N^2$. Consequently, the overall computational time for our algorithm is $O(N^2)$.

Notably, the TCD algorithm is ready for parallel computing, making computational tasks more feasible for large-scale data analysis. We introduce the parallel framework for conducting TCD in parallel, as illustrated in Figure 10.

Moreover, for large-scale networks, to tackle memory scalability issues, we have developed a strategy that calculates the elements of $\bar{\mathbf{C}}$ sequentially as required. Specifically, $\bar{\mathbf{C}}$ is only used in Step 3 of *our* algorithm, where we use depth-first search (DFS) to identify connected components in $\bar{\mathbf{C}}$. DFS works in a manner similar to a random walk, requiring only the selected node's connectivity information in each step of the walk. Thus, we only need to store the connectivity information of the currently explored node in each step of the walk, effectively reducing the space complexity from $O(N^2)$ to $O(N)$.