# Centrality-Aware Collaborative Network Embedding for Overlapping Community Detection

Xi Cheng, Wenjie Zhu, *Member, IEEE,* and Wei Qi Yan, *Senior Member, IEEE*

***Abstract*—Community detection is crucial for uncovering the intricate structures and dynamics within various networks, ranging from social interactions to biological systems. Despite significant advancements in community detection approaches, the task of learning network embeddings for complex networks characterized by overlapping communities presents considerable challenges. Specifically, nodes with higher centrality tend to contribute to increased overlaps across multiple communities, complicating the network embedding process. In this paper, we propose a Centrality-Aware Collaborative Learning (CACL) framework that leverages higher-order information through collaborative learning, incorporating a centrality penalty for overlapping community detection. The CACL framework integrates symmetric non-negative matrix factorization and kernel regression models, effectively addressing the limitations associated with traditional single-model techniques. Additionally, it emphasizes the contributions of nodes with higher centrality in the collaborative paradigm. By employing both first-order and second-order information, CACL preserves the intrinsic structure of the network, allowing for a more comprehensive representation of community relationships. To optimize the learning process of CACL, we implement a coordinate descent optimization scheme that tailors network embeddings specifically for overlapping community detection, thereby avoiding ad-hoc processing methods. Extensive experiments on ten LFR benchmark networks and six real-world networks demonstrate that CACL outperforms competing methods in accurately identifying overlapping communities, highlighting its efficacy in centrality-aware collaborative learning.**

***Index Terms*—Overlapping community detection, Centrality-aware collaborative learning, Symmetric non-negative matrix factorization, Kernel regression.**

## I. Introduction

COMPLEX network analysis has gained significant attention in the era of big data, with applications spanning social [1], biological [2], and information networks [3]. In network analysis, community detection is one of the most fundamental concepts that aim to identify cohesive subsets of nodes [4]. Communities or clusters typically consist of nodes that are more connected to each other than to external nodes, thereby revealing underlying patterns and functional modules within the network. In recent decades, various community detection methods [4]–[8] have been dedicated to uncovering the structure and dynamics of communities, providing valuable insights into the organization and functionality of complex networks.

Community detection approaches can be broadly categorized into two main types [9]: Disjoint Community Detection (DCD) and Overlapping Community Detection (OCD). DCD refers to the identification of communities in a network where each node belongs to exactly one community, meaning that there is no overlap between communities [10]. The existing DCD methods include those based on optimal modularity [6], matrix factorization [7], [11], multi-objective optimization [12], [13], and graph embedding [14], [15]. Given the explosive growth and diversification of information resources, multiple relationships often exist between nodes in the network [16]. For example, individuals in online social networks like Facebook or Twitter belong to multiple communities based on various factors such as interests, hobbies, and professional relationships [17]. Consequently, understanding the overlapping structure of networks has become increasingly essential. There is a growing body of research in OCD that focuses on extracting information from networks where nodes can belong to multiple communities [18], [19]. Various methodologies are being explored, including clique percolation [20], [21], label propagation [22], multi-objective optimization [23], [24], matrix factorization [25], and deep learning [26], [27], all aimed at enhancing the detection of these overlapping community structures.

To address the problem of OCD, a common approach involves setting a threshold on the fuzzy memberships of nodes based on existing detection results [6]. However, determining a universal threshold applicable to all nodes proves challenging [28]. Besides, previous OCD methods have raised several issues as follows:

- Highly centralized nodes often belong to multiple communities, resulting in dense memberships and overlapping communities. This complexity as well as the post-processing of threshold makes accurate learning of network embeddings difficult. How can we effectively capture nodes with multiple community memberships in a learnable manner?
- The existing OCD methods typically rely on a single model, which may result in sub-optimal network embeddings. Can we develop a collaborative learning framework that incorporates multiple models to achieve augmented network embeddings?

This work was supported in part by the Natural Science Foundation of Zhejiang Province of China under Grant LY24F030005, in part by National Key Research and Development Program of China under Grant 2021YFC3340402, and in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant 2024YW99. (Corresponding author: Wenjie Zhu.)

Xi Cheng and Wenjie Zhu are with Zhejiang-New Zealand Joint Vision-Based Intelligent Metrology Laboratory, College of Information Engineering, China Jiliang University, Hangzhou 310018, China (e-mail: zhwj@cjlu.edu.cn).

Wei Qi Yan is with Zhejiang-New Zealand Joint Vision-Based Intelligent Metrology Laboratory, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand (e-mail: weiqi.yan@aut.ac.nz).

Motivated by the contemplation of these issues, we propose a novel Centrality-Aware Collaborative Learning (CACL) framework for network embedding in the OCD task. CACL leverages higher-order information and provides a robust solution to the issues raised above learnable overlapping communities and single-model bias. Specifically, CACL employs two models, i.e., Symmetric Non-negative Matrix Factorization (SNMF) and Kernel Regression (KR), that work together to enhance network embeddings. Additionally, we introduce a centrality penalty mechanism to manage nodes that belong to multiple communities, and incorporate both first- and second-order information into the collaborative learning process to better preserve the network's intrinsic structure. The main contributions of our work are summarized as follows:

- We introduce the CACL framework, which integrates SNMF and KR models to collaboratively learn network embeddings. An auxiliary projection matrix is employed to maintain consistency in community memberships between the two models.
- We propose a centrality penalty mechanism to handle the challenge of nodes with multiple community memberships. The incorporation of both first- and second-order information further improves community detection accuracy. Furthermore, a coordinate descent method is proposed to solve the discrete optimization problem of CACL, thereby enhancing overall performance.
- Experimental results show that CACL outperforms the existing methods on ten LFR benchmark networks and six real-world networks, demonstrating its effectiveness and potential for advancing overlapping community detection.

The remainder of this paper is organized as follows: Section II provides a brief review of matrix factorization-based overlapping community detection. Section III details the proposed CACL framework, including the optimization algorithm and complexity analysis. Section IV presents the experimental results and their analysis. Finally, Section V concludes this work and provides directions for future work.

## II. Related Work

In this section, we provide a brief review of traditional overlapping community detection algorithms and NMF-based methods. We critically examine their strengths and limitations within the community detection framework and compare several prominent methods.

### A. Traditional Overlapping Community Detection Algorithms

Traditional overlapping community detection algorithms predominantly utilize graph structures and their inherent characteristics to assess both intra-community density and inter-community overlap. Representative approaches include the clique percolation method [29], link partitioning [30], and label propagation [31].

The Clique Percolation Method (CPM), introduced by Palla et al. [29], defines communities as unions of adjacent $k$-cliques, where a $k$-clique is a complete subgraph consisting of $k$ nodes. CPM identifies overlapping community structures by first locating all maximal $k$-cliques and then merging those that share at least $k-1$ nodes, thus naturally revealing overlaps. However, CPM suffers from excessive overlap among clusters, resulting in redundancy and inefficiency. To address these shortcomings, Ramesh et al. [20] proposed an MErged-maximal clique-based Multi-Objective Evolutionary Algorithm (MEMOEA), which reduces chromosome length by merging overlapping cliques. The MEMOEA method incorporates a re-labeling strategy to avoid redundant solutions and enhance population diversity, leading to improved community partitions in fewer generations compared to conventional approaches. Alternatively, to address CPM's reliance on densely connected subgraphs, Yang et al. [21] developed a Quadratic Optimization-based Clique Expansion (QOCE) approach. QOCE selects high-quality maximal cliques as seeds and employs a short random walk to sample a subgraph around each seed. The community extraction process is formulated as a quadratic optimization problem that minimizes the Cheeger cut, integrating an intra-clique connectivity measure within a quadratic programming model to guide expansion.

Link partitioning [30] partitions the edges of a network into overlapping communities. In link partitioning, communities are defined as groups of links; because edges can belong to multiple communities, nodes inherently participate in several communities. However, traditional link partitioning typically requires a predefined number of communities or relies on manual parameter tuning. Shi et al. [32] introduced a genetic algorithm-based link clustering method, GaoCD, to address challenges in automatically determining community numbers and capturing global structures. GaoCD encodes edges into chromosomes, applies genetic operators to generate candidate partitions, and employs a partition density objective to assess quality. To further improve performance, Ponomarenko et al. [33] proposed Link Partitioning Around Medoids (LPAM), which integrates link partitioning with medoid-based clustering. LPAM constructs a line graph of the original network, quantifies edge similarities using commute time metrics, and applies $k$-medoids clustering to segment edges into non-overlapping communities. Subsequently, node overlaps are inferred by applying a membership coefficient threshold, striking a balance between community separation and overlap.

The Label Propagation Algorithm (LPA) assigns affiliation coefficients to nodes by averaging those of their neighbors [31]. Initially, each node is assigned a unique label. Subsequently, labels are iteratively updated based on the most frequent (or weighted) labels of adjacent nodes until convergence or a preset iteration limit is reached. To enhance convergence speed and stability, Yan et al. [22] developed a Fast Label Propagation Algorithm (FLPA) that dynamically adjusts the propagation order by incorporating node importance (e.g., PageRank) and neighbor similarity. FLPA reduces network complexity through graph compression, computes node influence to modulate label weights, and establishes a fixed propagation sequence. Separately, Yu et al. [34] proposed a hybrid framework, termed Label Propagation Algorithm with Node Importance and Similarity (LPANIS), that integrates seed expansion with label propagation to concurrently detect overlapping and non-overlapping communities.

### B. Learning Model-Based Overlapping Community Detection

Learning model-based overlapping community detection aims to capture node features and network structural information to generate latent representations of nodes, thus uncovering community structures within networks. Among these models, NMF has been widely applied due to its strong interpretability and its capacity to extract fundamental information inherent in clustering paradigms [35], [36]. To facilitate presentation, we use uppercase bold letters to denote matrices and lowercase bold letters for vectors. The main notations used this paper are described in Table I. Given a non-negative matrix

TABLE I
NOTATIONS USED IN THIS PAPER

| Symbol | Definition and Description |
|---|---|
| $n$ | Number of nodes |
| $k$ | Number of communities |
| $\mathbf{A} \in \mathbb{R}^{n\times n}$ | Adjacency matrix |
| $\mathbf{S} \in \mathbb{R}^{n\times n}$ | Higher-order adjacency matrix |
| $\mathbf{D} \in \mathbb{R}^{n\times n}$ | Degree matrix |
| $\mathbf{1}_{(k\times n)} \in \mathbb{R}^{k\times n}$ | A $k \times n$ all-ones matrix |
| $\mathbf{0}_{(n\times n)} \in \mathbb{R}^{n\times n}$ | A $n \times n$ all-zero matrix |
| $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ | Identity matrix |
| $\mathbf{1}_n \in \mathbb{R}^n$ | An $n$-element all-ones vector |
| $\mathbf{X}_{i,:}$, $\mathbf{X}_{:,j}$ | The $i$-th row, $j$-th column of $\mathbf{X}$ |
| $\mathbf{X}_{ij}$ | The element in the $i$-th row and $j$-th column of $\mathbf{X}$ |
| $Tr(\mathbf{X})$ | Matrix trace of $\mathbf{X}$ |
| $\|\mathbf{X}\|_F$ | Frobenius norm of $\mathbf{X}$ |
| $\langle \mathbf{X}, \mathbf{Y} \rangle$ | Inner product of $\mathbf{X}$ and $\mathbf{Y}$ |
| $\mathbf{X} \odot \mathbf{Y}$ | Hadamard product of $\mathbf{X}$ and $\mathbf{Y}$ |

$\mathbf{X} = \{\mathbf{X}_{ij}\} \in \mathbb{R}^{m\times n}$. NMF aims to represent the data matrix $\mathbf{X}$ as the product of two non-negative low-rank matrices, $\mathbf{U}$ and $\mathbf{V}$, i.e., $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$. The optimization problem [37] with Euclidean distance for NMF is expressed as:

$$\min_{\mathbf{U}\geq 0, \mathbf{V}\geq 0} \left\| \mathbf{X} - \mathbf{U}\mathbf{V}^T \right\|_F^2, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$ are the basis and coefficient matrices, respectively, and $r$ is the rank of the factorization. Since the objective function in Eq. (1) is non-convex, it is solved by using the iterative multiplication update rules under the following relationship:

$$\begin{aligned} \mathbf{U} &\leftarrow \mathbf{U} \odot \left( \frac{\mathbf{X}\mathbf{V}}{\mathbf{U}\mathbf{V}^T\mathbf{V}} \right), \\ \mathbf{V} &\leftarrow \mathbf{V} \odot \left( \frac{\mathbf{X}^T\mathbf{U}}{\mathbf{V}\mathbf{U}^T\mathbf{U}} \right), \end{aligned} \tag{2}$$

where $\odot$ represents the Hadamard product, an element-wise multiplication operation between two matrices or vectors of the same dimensions.

By applying the multiplicative update rules in Eq. (2), the matrices $\mathbf{U}$ and $\mathbf{V}$ are updated iteratively, effectively capturing intricate interactions within the network structure and detecting the communities. Various advanced techniques have been developed to detect the overlapping communities using $\mathbf{V}$. Psorakis et al. [38] applied NMF to overlapping community detection by exploiting a soft partitioning approach and introducing a probabilistic method based on Bayesian Non-negative Matrix Factorization (BNMF). While BNMF provides interpretable membership distributions, it neglects higher-order structural patterns and node centrality, leading to suboptimal performance on networks with sparse overlaps. To address scalability, Yang and Leskovec [25] developed the Cluster Affiliation Model (BigCLAM), which scales to large networks by estimating community membership strengths. However, BigCLAM assumes dense overlaps, limiting its applicability to networks with sparse connectivity. Sørensen et al. [39] introduced a Semi-Binary Matrix Factorization (SBMF) model that integrates K-means clustering with semi-non-negative matrix factorization. SBMF addresses overlapping communities in complex networks through a coupled matrix-tensor decomposition approach; however, its computational process is still complex. Despite integrating collaborative learning between matrices and tensors, SBMF fails to capture node centrality and hierarchical relationships.

Symmetric NMF, a specialized variant of NMF tailored for symmetric matrices, was introduced to address the limitations of traditional NMF in handling nonlinear and structured data [40], [41]. Given a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, the goal is to find a low-rank approximation of this matrix by using the latent factor matrix $\mathbf{U} \in \mathbb{R}^{n\times k}$, such that $\mathbf{A} \approx \mathbf{U}\mathbf{U}^T$, where $k$ denotes the number of communities and $k < n$. The optimization problem for SNMF [42] is formulated as follows:

$$\min_{\mathbf{U}\geq 0} \left\| \mathbf{A} - \mathbf{U}\mathbf{U}^T \right\|_F^2. \tag{3}$$

To solve this optimization problem, an iterative multiplicative update rule is implemented for the matrix $\mathbf{U}$ as follows:

$$\mathbf{U} \leftarrow \mathbf{U} \odot \left( \frac{\mathbf{U}\mathbf{U}^T\mathbf{U} + \mathbf{A}\mathbf{U}}{2\mathbf{U}\mathbf{U}^T\mathbf{U}} \right). \tag{4}$$

SNMF-based methods have been widely employed for overlapping community detection. Ye et al. [43] introduced Discrete Non-negative Matrix Factorization (DNMF), which efficiently generates binary community membership matrices using SNMF. Despite its computational efficiency, DNMF faces challenges with large-scale networks. Huang et al. [44] improved scalability with Ego-splitting SNMF (ESNMF), which partitions networks into ego-centric subnetworks. While ESNMF captures local neighborhoods (including second-order neighbors), it does not adaptively penalize central nodes, causing over-assignment in hubs. Recently, Zhuo et al. [45] designed ECOCD to iteratively expand/contract communities using permanence-based pruning. Although ECOCD incorporates higher-order affiliation metrics, it relies solely on SNMF without collaborative learning.

In the context of OCD, an important observation is that overlapping community nodes frequently occupy positions analogous to structural hole spanners in social network theory [46], [47]. Both types of nodes serve as critical bridges connecting otherwise weakly linked communities, facilitating information diffusion, resource exchange, and innovation spread [30]. Prior studies have proposed dedicated methods for identifying such bridging nodes, including approaches that jointly detect communities and structural hole spanners via harmonic modularity [48] and recent deep learning frameworks leveraging behavior embeddings to discover them in large-scale online social networks [49]. The bridging role of such nodes is essential for maintaining network connectivity

TABLE II
COMPARISON WITH RELATED WORK

| Method | Higher-order | Collaborative Learning | Node Centrality |
|---|---|---|---|
| MEMOEA [20] | ✓ | × | ✓ |
| QOCE [21] | × | × | ✓ |
| GaoCD [32] | × | × | × |
| LPAM [33] | ✓ | × | × |
| FLPA [22] | ✓ | × | ✓ |
| LPANIS [34] | ✓ | × | ✓ |
| BNMF [38] | × | × | × |
| BigCLAM [25] | ✓ | × | × |
| SBMF [39] | × | ✓ | × |
| DNMF [43] | × | ✓ | × |
| ESNMF [44] | ✓ | × | × |
| ECOCD [45] | ✓ | × | ✓ |
| CACL (Ours) | ✓ | ✓ | ✓ |

and promoting cross-community interactions. However, if the community assignments of such bridging nodes are overly ambiguous, the resulting community boundaries may become blurred, weakening interpretability and obscuring functional roles in the network. Recognizing and accurately characterizing these nodes is therefore vital for both precise community detection and understanding broader network dynamics. As shown in Table II, the existing overlapping community detection methods exhibit key limitations. The extension methods for traditional overlapping community detection, such as MEMOEA [20], LPAM [33], FLPA [22] and LPANIS [34], indirectly utilize neighbor information but are confined to local structures (e.g., direct neighbors or small-sized cliques). Most approaches employ single-model architectures without incorporating multi-model collaboration mechanisms. Additionally, many methods explicitly rely on node centrality measures (degree, PageRank, K-Shell, etc.) to filter seed nodes or partition communities. Furthermore, numerous NMF-based methods, such as BNMF [38] and SBMF [39], rely solely on adjacency matrices, lacking higher-order information. While methods like BigCLAM [25] and ESNMF [44] incorporate multi-hop relationships, they overlook collaborative learning, limiting adaptability. Moreover, most methods fail to consider node centrality, with ECOCD [45] being a heuristic exception. To address these gaps, we propose a novel collaborative learning framework that uniquely integrates higher-order information, collaborative learning, and node centrality, enabling more accurate overlapping community detection.

## III. PROPOSED METHOD

### A. Overview of CACL Framework

The proposed CACL framework leverages collaborative learning to address the sub-optimal network embeddings that arise from relying on a single model. In overlapping community detection, identifying overlapping communities requires considering multiple connections between nodes. While the existing methods based on NMF or SNMF demonstrate strengths in probabilistic modeling [38], [45], scalability [25], [44], and computational efficiency [43], they often neglect the sparse connectivity of central nodes and fail to integrate higher-order structural information. Addressing node centrality sparsity is essential to prevent the over-assignment of nodes to multiple communities, and incorporating higher-order structural information helps capture complex relationships between distant nodes. Our novel CACL framework, illustrated in Fig. 1, comprises three modules, namely higher-order adjacency construction, collaborative learning, and sparse centrality measure. The higher-order adjacency construction module explores a more comprehensive network structure; the collaborative learning module integrates the SNMF model for soft community memberships and the KR model for hard community memberships; and the sparse centrality measure module applies penalties based on node centrality to refine the collaborative learning process, thereby enhancing the accuracy of overlapping community detection. The objectives of these modules are further detailed in Section III-B.

### B. Objective of the Proposed CACL Framework

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices $\mathcal{V} = \{v_1, v_2, v_3, \ldots, v_n\}$ and $\mathcal{E}$ is the set of edges $\mathcal{E} = \{e_1, e_2, e_3, \ldots, e_m\}$, we focus on undirected graphs represented by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $\mathbf{A}_{ij} \in \{0, 1\}$ indicates whether an edge exists between vertices $v_i$ and $v_j$: if an edge exists, $\mathbf{A}_{ij} = 1$; otherwise, $\mathbf{A}_{ij} = 0$. The goal of community detection is to partition $\mathcal{G}$ into its constituent communities. Assume $\mathcal{G}$ is composed of $k$ communities. We denote $\mathcal{C}$ as the set of these communities, $\mathcal{C} = \{C_i | C_i \neq \emptyset, C_i \neq C_j, 1 \leq i, j \leq k\}$, where $C_i$ represents the $i$-th community. In overlapping community detection, it is acknowledged that a node may be assigned to both $C_i$ and $C_j$ for $i \neq j$, indicating the potential for nodes to belong to multiple communities.

*1) Higher-Order Adjacency Construction:* In community detection, accurately capturing the relationships between nodes within a community is essential [50]. Most methods rely on the first-order adjacency matrix $\mathbf{A}$, where $\mathbf{A}_{ij}$ is 1 if nodes $i$ and $j$ are directly connected, and 0 otherwise. While this approach is straightforward, it has notable limitations. Many community networks are sparse, meaning that numerous nodes within a community may lack direct connections with each other [51]. Consequently, the first-order adjacency matrix $\mathbf{A}$ might not fully represent the relationships among nodes. For example, two nodes could be closely related through indirect connections or shared neighbors but would appear unconnected in the first-order adjacency matrix.

Previous research work has made significant strides in overlapping community detection but has not fully leveraged higher-order structural information. In this paper, we propose using higher-order adjacency matrices, specifically the second-order adjacency matrix, to provide a more comprehensive depiction of node relationships. This matrix reflects connections through intermediate nodes and captures indirect links that the first-order adjacency matrix might miss. We suggest constructing a higher-order adjacency matrix $\mathbf{S}$ by combining the first-order and second-order matrices through weighted summation. Formally, the first-order adjacency matrix $\mathbf{A}^{(1)} \in \mathbb{R}^{n \times n}$ represents direct connections between nodes, i.e., $\mathbf{A}^{(1)} = \mathbf{A}$, where $n$ denotes the total number of nodes in the network. In contrast, the second-order adjacency matrix $\mathbf{A}^{(2)} \in \mathbb{R}^{n \times n}$

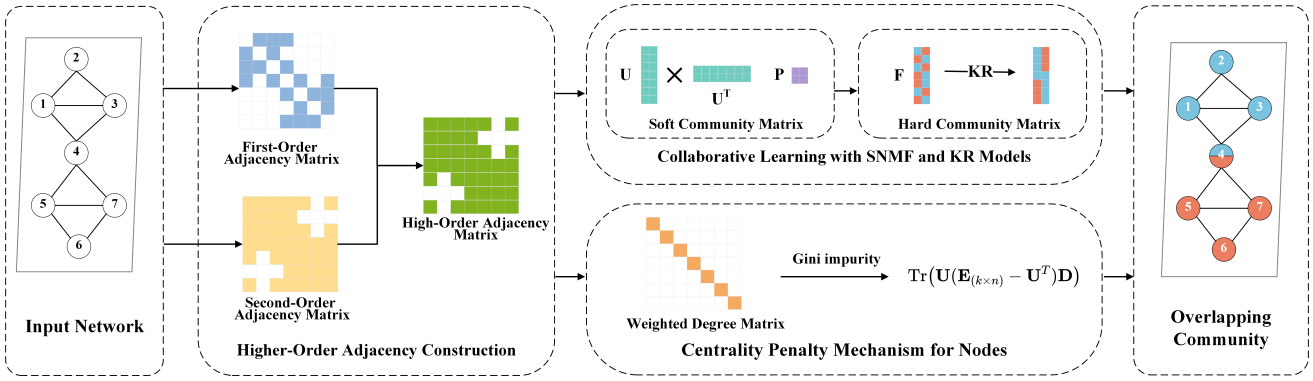



Fig. 1. The CACL framework for overlapping community detection consists of three modules, i.e., higher-order adjacency construction, collaborative learning, and centrality penalty mechanism for nodes. Given an input network, we compute higher-order adjacency matrices and then apply a collaborative learning model with a centrality penalty to obtain the final community partition.

captures indirect links between nodes using cosine similarity, defined as

$$\mathbf{A}_{ij}^{(2)} = \frac{\mathcal{N}_i \cdot \mathcal{N}_j}{\|\mathcal{N}_i\|\|\mathcal{N}_j\|}, \tag{5}$$

where $\mathcal{N}_i = \left(\mathbf{A}_{i1}^{(1)}, \mathbf{A}_{i2}^{(1)}, \cdots, \mathbf{A}_{in}^{(1)}\right)$ denotes the first-order neighbors of $v_i$ among the other elements. To achieve a comprehensive representation of node relationships, the higher-order adjacency matrix is defined as:

$$\mathbf{S} = \mathbf{A}^{(1)} + \eta \mathbf{A}^{(2)}, \tag{6}$$

where $\eta$ is a balancing parameter that adjusts the relative importance of direct $\mathbf{A}^{(1)}$ and indirect $\mathbf{A}^{(2)}$ connections. As suggested in [52], the parameter $\eta$ is set to 5 in this study. The higher-order matrix $\mathbf{S}$ integrates both direct and indirect connections, offering a more nuanced depiction of community structure.

*2) Collaborative Learning:* To effectively address the limitations associated with traditional single-model community detection, The proposed CACL framework adopts collaborative learning strategy and incorporates a centrality penalty mechanism for nodes to improve the network embeddings.

**(1) SNMF-based Soft Community Matrix Generation**

Symmetric NMF is one of the models used in the collaborative learning module to derive a soft community membership matrix $\mathbf{U} \in \mathbb{R}^{n\times k}$, where $k$ denotes the number of potential communities [53]. Each entry $\mathbf{U}_{ij}$ represents the probability that node $i$ belongs to the community $C_j$. This approach allows nodes to belong to multiple communities with varying degrees of membership, reflecting the complexity of real-world networks. Our objective is to minimize the difference between the actual adjacency matrix $\mathbf{S}$ and its low-rank approximation $\mathbf{U}\mathbf{U}^T$, i.e., $\|\mathbf{S}-\mathbf{U}\mathbf{U}^T\|_F^2$. Solving this problem yields the soft community membership matrix $\mathbf{U}$, offering a flexible view of node-community relationships. To derive precise community memberships, these continuous values need to be discretized. We propose an auxiliary transformation to convert $\mathbf{U}$ into a hard membership matrix $\mathbf{F}$. This involves using a projection matrix $\mathbf{P} \in \mathbb{R}^{k\times k}$ and ensuring orthogonality, so $\mathbf{U} \approx \mathbf{F}\mathbf{P}^T$ and $\mathbf{F}\mathbf{F}^T = \mathbf{F}\mathbf{P}^T\mathbf{P}\mathbf{F}^T = (\mathbf{F}\mathbf{P}^T)(\mathbf{F}\mathbf{P}^T)^T \approx \mathbf{U}\mathbf{U}^T$. Consequently, the optimization problem of SNMF-based soft community matrix generation is:

$$\min_{\mathbf{U},\mathbf{F},\mathbf{P}} \mathcal{J}_{\text{SNMF}} = \left\|\mathbf{S} - \mathbf{U}\mathbf{U}^T\right\|_F^2 + \alpha \left\|\mathbf{U} - \mathbf{F}\mathbf{P}^T\right\|_F^2, \tag{7}$$

where $\alpha$ is a weighting parameter, and $\mathbf{F} \in \{\mathbf{F} \mid \mathbf{F} \in \mathbb{R}^{n\times k} \in \{0,1\}$ and $\sum_{j=1}^{k} \mathbf{F}_{ij} \geq 1$ for $i = 1, \cdots, n\}$ ensures sparsity and accurate community membership. By minimizing $\mathcal{J}_{\text{SNMF}}$, we effectively convert from soft to hard memberships while retaining higher-order adjacency information. This approach also allows us to adaptively adjust thresholds based on community overlap, leading to more accurate community assignments for data points. It is important to note that we do not directly use $\mathbf{F}$ to determine hard community memberships due to the complexity of solving the bi-quadratic discrete optimization problem. However, the introduction of $\mathbf{P}$ makes the learning process for the objective model more manageable.

**(2) KR-based Hard Community Matrix Generation**

Given the higher-order adjacency matrix $\mathbf{S} = \{\mathbf{S}_{:,1}, \mathbf{S}_{:,2}, \ldots, \mathbf{S}_{:,n}\} \in \mathbb{R}^{n\times n}$ and the hard community membership matrix $\mathbf{F} = \{\mathbf{F}_{:,1}, \mathbf{F}_{:,2}, \ldots, \mathbf{F}_{:,k}\} \in \mathbb{R}^{n\times k}$. We can learn a nonlinear regression model in KR whose parameters are determined by minimizing the following regularized sum of squares cost function:

$$\mathcal{J}(\mathbf{W}, \mathbf{c}) = \left\|\mathbf{F} - \phi^T(\mathbf{S})\,\mathbf{W} - \mathbf{1}_n\mathbf{c}^T\right\|_F^2 + \gamma \mathrm{Tr}(\mathbf{W}^T\mathbf{W}), \tag{8}$$

where $\phi : \mathbb{R}^n \to \mathcal{H}$ defines a nonlinear kernel space mapping function, $\mathbf{c} \in \mathbb{R}^k$ denotes the bias vector. Let $\mathbf{W} \in \mathbb{R}^{n\times k}$, where $\mathbf{W} = \phi(\mathbf{S})\mathbf{Z}$ and $\mathbf{Z} \in \mathbb{R}^{n\times k}$ denotes the kernel coefficients. Additionally, $\mathbf{K} = \phi^T(\mathbf{S})\phi(\mathbf{S})$, we rewrite the objective in Eq. (8) as

$$\begin{aligned}\mathcal{J}(\mathbf{Z}, \mathbf{c}) &= \left\|\mathbf{F} - \phi^T(\mathbf{S})\,\phi(\mathbf{S})\,\mathbf{Z} - \mathbf{1}_n\mathbf{c}^T\right\|_F^2 \\ &\quad + \gamma \mathrm{Tr}\left(\mathbf{Z}^T\phi^T(\mathbf{S})\phi(\mathbf{S})\mathbf{Z}\right) \\ &= \left\|\mathbf{F} - \mathbf{K}\mathbf{Z} - \mathbf{1}_n\mathbf{c}^T\right\|_F^2 + \gamma \mathrm{Tr}\left(\mathbf{Z}^T\mathbf{K}\mathbf{Z}\right).\end{aligned} \tag{9}$$

Let $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ is a centering matrix that enforces the bias term to be zero, thereby facilitating subsequent calculations. The mapped kernel data can be centered as

$\tilde{\phi}(\mathbf{S}) = \phi(\mathbf{S})\mathbf{H}$ by using a Gaussian kernel. Accordingly, the centered kernel matrix is computed as

$$\tilde{\mathbf{K}} = \tilde{\phi}^T(\mathbf{S})\tilde{\phi}(\mathbf{S}) = \mathbf{H}^T\phi(\mathbf{S})^T\phi(\mathbf{S})\mathbf{H} = \mathbf{H}^T\mathbf{K}\mathbf{H}. \quad (10)$$

Replacing the kernel matrix $\mathbf{K}$ with the center kernel matrix $\tilde{\mathbf{K}}$ in Eq. (9), the objective function becomes:

$$\mathcal{J}(\mathbf{Z}, \mathbf{c}) = \left\|\mathbf{F} - \tilde{\mathbf{K}}\mathbf{Z} - \mathbf{1}_n\mathbf{c}^T\right\|_F^2 + \gamma \mathrm{Tr}\left(\mathbf{Z}^T\tilde{\mathbf{K}}\mathbf{Z}\right). \quad (11)$$

Since the data is centered, we have $\tilde{\mathbf{K}}\mathbf{1}_n = 0$. By setting the derivatives with respect to $\mathbf{Z}$ and $\mathbf{c}$, the following is obtained:

$$\begin{cases} \mathbf{c} = \frac{1}{n}\mathbf{F}^T\mathbf{1}_n, \\ \mathbf{Z} = (\tilde{\mathbf{K}} + \gamma\mathbf{I}_n)^{-1}\mathbf{F}. \end{cases} \quad (12)$$

Substituting Eq. (12) into Eq. (11), the optimization problem is formulated as following:

$$\min_{\mathbf{F}} \mathcal{J}_{\mathrm{KR}} = \mathrm{Tr}\left(\mathbf{F}^T\mathbf{Q}\mathbf{F}\right), \quad (13)$$

where $\mathbf{Q} = \mathbf{H} - (\tilde{\mathbf{K}} + \lambda\mathbf{I}_n)^{-1}\tilde{\mathbf{K}}$. Transforming the soft community membership matrix into a hard community membership matrix using KR enhances the accuracy of community structure identification. Essentially, KR captures the inherent non-linear relationships within network data, providing a more accurate representation of the complex and overlapping nature of real-world networks.

*3) Centrality Penalty Mechanism for Nodes:* In network analysis, core nodes are typically prominent due to elevated node degree and centrality, which measure the importance of these nodes within the network [54]. Node centrality sparsity involves making the community membership of central nodes, those with a higher number of connections, more distinct and focused during community allocation.

Node centrality is often measured by a node's degree. However, in this paper, we use a weighted degree matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$, where $\mathbf{D}_{ii} = \sum_{j=1}^{n}\mathbf{S}_{:,j}$, and $\mathbf{S}$ is the higher-order adjacency matrix. Gini impurity is employed to assess the clarity of a node's community membership and is defined as follows:

$$\mathrm{Gini}(v_i) = \sum_{l=1}^{k}\mathbf{U}_{il}\left(\mathbf{1}_{(n\times n)} - \mathbf{U}_{il}\right), \quad (14)$$

where $\mathbf{U}_{il}$ is the probability of node $v_i$ belonging to community $\mathcal{C}_l$, and $k$ is the number of communities. To promote node centrality sparsity, our goal is to reduce the Gini impurity of higher-centrality nodes during the optimization process. This objective can be accomplished through the following sparsity regularization:

$$\sum_{i=1}^{n}\mathbf{D}_{ii}\mathrm{Gini}(v_i). \quad (15)$$

In order to integrate the Gini [55] impurity into the objective of the proposed CACL framework, we express it by using its matrix form:

$$\mathrm{Gini}(v_i) = \sum_{j=1}^{k}\mathbf{U}_{ij}\left(1 - \mathbf{U}_{ij}\right) = \left(\mathbf{U}\left(\mathbf{1}_{(k\times n)} - \mathbf{U}^T\right)\right)_{ii}, \quad (16)$$

where $\mathbf{1}_{(k\times n)} \in \mathbb{R}^{k\times n}$ is a $k \times n$ all-ones matrix. Combining the weighted degree matrix $\mathbf{D}$ and the matrix representation of Gini impurity, the centrality penalty term is written as

$$\mathcal{J}_{\mathrm{CP}} = \mathrm{Tr}\left(\mathbf{U}\left(\mathbf{1}_{(k\times n)} - \mathbf{U}^T\right)\mathbf{D}\right). \quad (17)$$

In the context of community detection, penalizing nodes with high centrality by minimizing $\mathcal{J}_{\mathrm{CP}}$ in Eq. (17), particularly those with diffuse community memberships, aligns with concepts from community-aware centrality [56], [57]. Centrality, often based on degree or other metrics like betweenness or closeness, identifies nodes that are pivotal within a network. However, in the setting of community detection, nodes with high centrality but weak affiliations to specific communities can act as "bridges" or "global influencers". These nodes are often critical for maintaining global connectivity but can obscure clear community structures. Penalizing these nodes helps to enhance the detection of more cohesive, well-defined communities by discouraging the prominence of these "bridges". Thus, the penalty applied to high-centrality nodes with diffuse community memberships encourages the discovery of communities that are both internally connected and externally less influenced by "global influencers", ultimately improving the quality of community detection algorithms. For now, we integrate equations Eq. (7), Eq. (13) and Eq. (17) to derive the overall optimization problem as follows:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{F},\mathbf{P}} \mathcal{J}_{\mathrm{SNMF}} + \beta\mathcal{J}_{\mathrm{KR}} + \lambda\mathcal{J}_{\mathrm{CP}} \\ &\quad = \underbrace{\|\mathbf{S} - \mathbf{U}\mathbf{U}^{\mathbf{T}}\|_F^2 + \alpha\|\mathbf{U} - \mathbf{F}\mathbf{P}^{\mathbf{T}}\|_F^2 + \beta\,\mathrm{Tr}(\mathbf{F}^T\mathbf{Q}\mathbf{F})}_{\text{collaborative learning}} \\ &\quad + \underbrace{\lambda\,\mathrm{Tr}\left(\mathbf{U}\left(\mathbf{1}_{(k\times n)} - \mathbf{U}^T\right)\mathbf{D}\right)}_{\text{centrality penalty}}, \\ &\text{s.t. } \mathbf{U} \geq \mathbf{0}, \mathbf{P}^T\mathbf{P} = \mathbf{I}_k, \mathbf{F} \in \{\mathbf{F} \mid \mathbf{F} \in \mathbb{R}^{n\times k} \in \{0,1\} \text{ and} \\ &\quad \sum_{j=1}^{k}\mathbf{F}_{ij} \geq 1 \text{ for } i = 1, \cdots, n\}, \end{aligned} \quad (18)$$

where $\beta$ and $\lambda$ are the weight parameters. $\mathcal{J}_{\mathrm{SNMF}}$ and $\mathcal{J}_{\mathrm{KR}}$ are part of the collaborative learning module, while $\mathcal{J}_{\mathrm{CP}}$ belongs to the centrality penalty module. The objective function can be addressed using the coordinate descent scheme. Through iterative updates of the three variables, $\mathbf{U}$, $\mathbf{F}$, and $\mathbf{P}$, the objective function will converge to a local minimum, effectively identifying and delineating the overlapping community structures within the network.

### C. Optimization

In this section, we divide the problem in Eq. (18) into three subproblems and propose gradient descent-based optimization algorithms to solve each one. Firstly, we rewrite the problem

in trace form as follows:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{F},\mathbf{P}} & \operatorname{Tr}(\mathbf{S}^T\mathbf{S}) - 2\operatorname{Tr}\left(\mathbf{S}^T\mathbf{U}\mathbf{U}^\mathbf{T}\right) + \operatorname{Tr}\left(\mathbf{U}\mathbf{U}^\mathbf{T}\mathbf{U}\mathbf{U}^\mathbf{T}\right) \\ & + \alpha\operatorname{Tr}(\mathbf{U}^T\mathbf{U}) - 2\alpha\operatorname{Tr}(\mathbf{U}^T\mathbf{F}\mathbf{P}^\mathbf{T}) + \alpha\operatorname{Tr}\left(\mathbf{F}^T\mathbf{F}\right) \\ & + \beta\operatorname{Tr}\left(\mathbf{F}^T\mathbf{Q}\mathbf{F}\right) + \lambda\operatorname{Tr}\left(\mathbf{U}\left(\mathbf{1}_{(k\times n)} - \mathbf{U}^T\right)\mathbf{D}\right), \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{P}^T\mathbf{P} = \mathbf{I}_k, \mathbf{F} \in \{\mathbf{F} \mid \mathbf{F} \in \mathbb{R}^{n\times k} \in \{0,1\} \\ & \text{and} \sum_{j=1}^{k} \mathbf{F}_{ij} \geq 1 \text{ for } i = 1, \cdots, n\}.\end{aligned} \tag{19}$$

The optimization problem is decomposed into three subproblems, and each subproblem is solved iteratively as follows:

**(1) U-subproblem**

With $\mathbf{F}$ and $\mathbf{P}$ fixed, we solve the following $\mathbf{U}$ subproblem:

$$\begin{aligned}\min_{\mathbf{U}} & \operatorname{Tr}(\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{U}^T) - \operatorname{Tr}(\mathbf{S}\mathbf{U}\mathbf{U}^\mathbf{T}) + \alpha\operatorname{Tr}(\mathbf{U}^T\mathbf{U}) \\ & - 2\alpha\operatorname{Tr}(\mathbf{F}\mathbf{P}^+\mathbf{U}^T) + 2\alpha\operatorname{Tr}(\mathbf{F}\mathbf{P}^-\mathbf{U}^T) \\ & + \lambda\operatorname{Tr}(\mathbf{U}\mathbf{1}_{(k\times n)}\mathbf{D}) - \lambda\operatorname{Tr}(\mathbf{U}\mathbf{U}^T\mathbf{D}), \\ \text{s.t.} \quad & \mathbf{U} \geq 0.\end{aligned} \tag{20}$$

After omitting terms unrelated to $\mathbf{U}$, the update rule for $\mathbf{U}$ according to the Karush-Kuhn-Tucker condition [58] is as follows:

$$\mathbf{U} \leftarrow \mathbf{U} \odot \left(\frac{2\mathbf{S}\mathbf{U} + \alpha\mathbf{F}\mathbf{P}^+ + 2\lambda\mathbf{U}\mathbf{D}}{2\mathbf{U}\mathbf{U}^T\mathbf{U} + \alpha\mathbf{U} + \alpha\mathbf{F}\mathbf{P}^- + \lambda\mathbf{1}_{(k\times n)}\mathbf{D}}\right)^{\frac{1}{4}}, \tag{21}$$

where $\mathbf{P}^+$ and $\mathbf{P}^-$ are defined as the positive and negative parts of $\mathbf{P}$, respectively, such that:

$$\mathbf{P}^+ = \frac{|\mathbf{P}| + \mathbf{P}}{2}, \quad \mathbf{P}^- = \frac{|\mathbf{P}| - \mathbf{P}}{2},$$

where $|\mathbf{P}|$ denotes the absolute value of each element in $\mathbf{P}$.

**(2) F-subproblem**

With $\mathbf{U}$ and $\mathbf{P}$ fixed, the $\mathbf{F}$-subproblem can be expanded into the following form:

$$\begin{aligned}\min_{\mathbf{F}} & \operatorname{Tr}\left(\mathbf{F}^T\left(\beta\mathbf{Q} + \alpha\mathbf{I}_n\right)\mathbf{F}\right) - 2\alpha\operatorname{Tr}\left(\mathbf{U}\mathbf{P}^T\mathbf{F}\right), \\ \text{s.t.} \quad & \mathbf{F} \in \{\mathbf{F} \mid \mathbf{F} \in \mathbb{R}^{n\times k} \in \{0,1\} \\ & \text{and} \sum_{j=1}^{k} \mathbf{F}_{ij} \geq 1 \text{ for } i = 1, \cdots, n\},\end{aligned} \tag{22}$$

where $\mathbf{Q}' = \beta\mathbf{Q} + \alpha\mathbf{I}_n$. We employ the discrete coordinate descent method to update $\mathbf{F}$, expressed as follows:

$$\begin{aligned}\min_{\mathbf{F}_{i,:}} & \mathbf{Q}'_{ii}\mathbf{F}_{i,:}\mathbf{F}_{i,:}^T + 2(\mathbf{Q}'_{i,:}\mathbf{F}' - \alpha\mathbf{P}_{i,:})\mathbf{F}_{i,:}^T, \\ \text{s.t.} \quad & \mathbf{F}_{i,:} \in \mathbb{R}^k \in \{0,1\} \\ & \text{and} \sum_{j=1}^{k} \mathbf{F}_{ij} \geq 1 \text{ for } i = 1, \cdots, n.\end{aligned} \tag{23}$$

To simplify the calculations, we define $\mathbf{h} = \mathbf{Q}'_{ii}\mathbf{1}_k^T + 2(\mathbf{Q}'_{i,:}\mathbf{F}' - \alpha\mathbf{P}_{i,:})$. The assignment rule for the optimal solution $\mathbf{F}$ in relation to the minimum value of $\mathbf{h}$ given by the following equation:

$$\mathbf{F}_{ij} = \begin{cases} 1, & \text{if } \mathbf{h}_j = \min(\mathbf{h}) \quad \text{or} \quad \mathbf{h}_j < 0, \\ 0, & \text{if } \mathbf{h}_j \neq \min(\mathbf{h}) \quad \text{and} \quad \mathbf{h}_j \geq 0. \end{cases} \tag{24}$$

**(3) P-subproblem**

With $\mathbf{U}$ and $\mathbf{F}$ fixed, the solution to $\mathbf{P}$ subproblem is obtained by the following maximization problem.

$$\begin{aligned}\max_{\mathbf{P}} & \operatorname{Tr}\left(\mathbf{U}^T\mathbf{F}\mathbf{P}\right), \\ \text{s.t.} \quad & \mathbf{P}^T\mathbf{P} = \mathbf{I}_k.\end{aligned} \tag{25}$$

Let $\mathbf{U}^T\mathbf{F} = \mathbf{\Omega}_1\mathbf{\Sigma}\mathbf{\Omega}_2^T$, the solution of the problem in Eq. (25) can be solved using singular value decomposition as follows:

$$\mathbf{P} = \mathbf{\Omega}_2\mathbf{\Omega}_1^T. \tag{26}$$

The proposed CACL framework for overlapping community detection is outlined in Algorithm 1. By alternately optimizing the three subproblems, the algorithm steadily reduces the objective function's value during each iteration until it converges. Once convergence is achieved, the community membership for each node is determined by matrix $\mathbf{F}$ (see **Step 10** $\sim$ **14** in Algorithm 1). The algorithm then returns the final overlapping communities, denoted as $\mathcal{C}$.

**Algorithm 1** CACL for Overlapping Community Detection

**Input:** The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and the weight parameters $k, \alpha, \beta, \gamma, \lambda$.
**Output:** The final overlapping communities $\mathcal{C}$.

```
Compute the adjacency matrix S via Eq. (6);
Initialize U ≥ 0, F ≥ 0, P ≥ 0;
while not converged do
    Fixing F and P, update matrix U via Eq. (21);
    Fixing U and P, update matrix F via Eq. (24);
    Fixing U and F, update matrix P via Eq. (26);
end while
Initialize the communities C = {C_1, C_2, ..., C_k}
for each i from 1 to n do
    for each j such that F_ij = 1 do
        C_j = C_j ∪ {v_i};
    end for
end for
return C
```

### D. Complexity Analysis

In this section, we analyze the time complexity of CACL. Let $n$ denote the number of nodes, $k$ represent the number of communities to be detected, and $t$ indicate the total number of iterations in Algorithm 1. The computational complexity of Algorithm 1 consists of three primary components: updating $\mathbf{U}$, $\mathbf{F}$, and $\mathbf{P}$. Updating $\mathbf{U}$ involves multiple matrix multiplications and element-wise divisions, leading to a per-iteration complexity of $O(n^2k + nk^2)$. The update of $\mathbf{F}$ includes an inner loop iterating over individual elements $f_i$, resulting in a per-iteration complexity of $O(nk)$. Given that this process requires multiple iterations for convergence, the total complexity for updating $\mathbf{F}$ is $O(n^2k)$. The update of $\mathbf{P}$ requires $O(nk^2 + k^3)$ operations. Since $k \ll n$, we approximate the complexity as $O(nk^2)$, assuming that the $k^3$ term remains asymptotically insignificant compared to $nk^2$. By aggregating the complexity of all components, the overall time complexity of the algorithm is $O(t(n^2k + nk^2))$.

In terms of space complexity, the CACL method constructs both the adjacency matrix and the higher-order adjacency matrix as dense structures, incurring a space complexity of $\mathcal{O}(n^2)$. Additionally, storing the feature matrix $\mathbf{U} \in \mathbb{R}^{n\times k}$ requires an extra $\mathcal{O}(nk)$ space. Since $k \ll n$, the overall space complexity is therefore dominated by the quadratic term, i.e., $\mathcal{O}(n^2)$. To overcome this limitation, we leverage the EasyGraph toolkit [59] for memory optimization, which employs row-based sparse storage formats to reduce both memory usage and matrix-copy overhead. As a result, the storage requirement for the adjacency and similarity matrices is effectively reduced to $\mathcal{O}(m)$, where $m$ denotes the number of edges. Although the feature matrix remains dense, its significantly smaller size means it only contributes $\mathcal{O}(nk)$ to the total space cost. Consequently, the overall space complexity is substantially reduced to $\mathcal{O}(m + nk)$.

## IV. Experiments

### A. Implementation and Evaluation Metrics

In this section, we conduct a group of experiments on ten LFR benchmark networks and six real-world network datasets to verify the effectiveness of the proposed CACL method. All experiments were conducted on a 64-bit architecture computer with 32 GB RAM and an Intel(R) Core(TM) i7-13700F 2.10 GHz processor. To evaluate the overlapping community detection performance, we employ two commonly used metrics in the experiments, i.e., Modular Expansion (EQ) and Overlapping Normalized Mutual Information (ONMI). These metrics provide critical insights into the structure and effectiveness of community detection in complex networks.

EQ [60] quantifies the cohesion of community structures and is defined as

$$\mathrm{EQ} = \frac{1}{2m}\sum_{i,j}\left(\mathbf{A}_{ij} - \frac{d_i d_j}{2m}\right)\frac{1}{\mathcal{C}_i\mathcal{C}_j}, \tag{27}$$

where $m$ denotes the total number of edges in the network, $\mathbf{A}_{ij}$ represents an element of the adjacency matrix, $d_i$ and $d_j$ denote the degrees of nodes $i$ and $j$, respectively, and $\mathcal{C}_i$ is the membership number of the node $i$. A higher EQ value indicates a more cohesive community structure, reflecting better performance in community detection.

ONMI [61] is another essential metric for evaluating the similarity between detected and true community structures, especially in networks where nodes can belong to multiple communities. ONMI is defined as follows:

$$\mathrm{ONMI}(X,Y) = 1 - \frac{1}{2}\left(\sum_{i=1}^{k}\frac{H(X_i|Y)}{H(X_i)} + \sum_{j=1}^{k}\frac{H(Y_j|X)}{H(Y_j)}\right), \tag{28}$$

where $X$ indicating detected community assignments, and $Y$ representing true community assignments. Let $H(X_i)$ and $H(Y_j)$ denote the entropies of the $i$-th community in $X$ and the $j$-th community in $Y$, respectively. Similarly, $H(X_i|Y)$ and $H(Y_j|X)$ represent the conditional entropies of $X_i$ given $Y$ and $Y_j$ given $X$, respectively. A higher ONMI value indicates more accurate detection of overlapping communities, thus reflecting better performance in community detection.

Given that the LFR benchmark networks provide ground truth data, we employ ONMI for evaluation. In alignment with the existing body of work [43], [45] on OCD, EQ is used as the performance assessment metric for real-world networks that lack overlapping ground-truth communities. These metrics effectively quantify intra-community cohesion, inter-community separation, and the precision of detected community structures in the analysis of complex networks with overlapping communities.

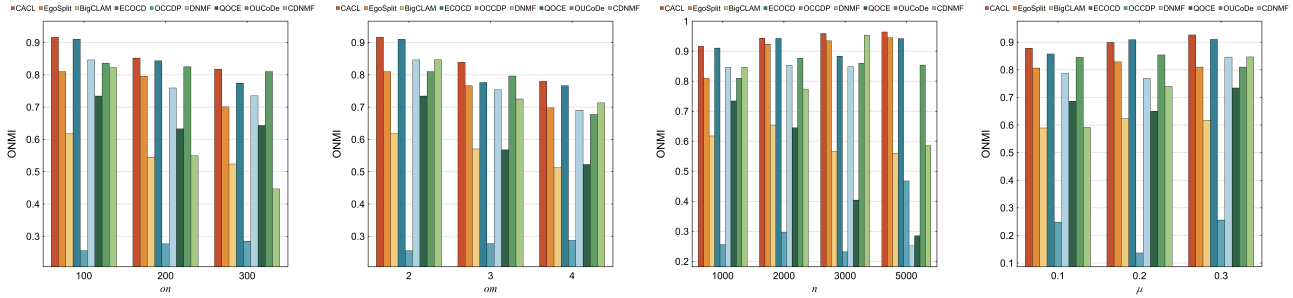
### B. Compared Methods

To evaluate the overlapping community detection performance of our CACL approach, we select eight representative OCD methods, i.e., BigCLAM [25], EgoSplit [62], OCDDP [63], DNMF [43], QOCE [21], ECOCD [45], OUCoDe [64] and CDNMF [65], as the comparison methods in the experiment. During the experiments, we fine-tune the parameters of all baseline methods according to the authors' recommendations. Specifically, for the CACL method, we fix $\eta = 5$ and vary $\alpha$ over $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$, $\beta$ and $\gamma$ over $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ and $\lambda$ over $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. For BigCLAM, DNMF, and CACL methods, we set the number of communities $k$ to match the ground truth, allowing us to evaluate their ability to identify real communities. For the QOCE algorithm, we employ the Bron-Kerbosch algorithm [66] to enumerate all maximal cliques in graph $\mathcal{G}$, and then, according to the method described in QOCE [21], select high-quality maximal cliques to form the initial seed. For deep learning-based comparison methods that rely on attributed networks, considering that the current study focuses on unattributed networks, we set the attribute matrices in OUCoDe [64] and CDNMF [65] to all-one matrices to ensure a fair comparison.

### C. Experiments on Synthetic Networks

The LFR benchmark networks [61] are synthetic datasets that can be applied to generate community structures with different sizes and degrees of overlap and are widely used to evaluate and compare the performance of overlapping community detection algorithms. The parameters for the LFR benchmarks are described in Table III. We constructed ten LFR benchmark networks by varying the total number of nodes $n$, the number of overlapping nodes $on$, the number of memberships per overlapping node $om$, and the mixing parameter $\mu$ (indicating inter-community connectivity), while keeping all other parameters constant, as detailed in Table IV.

To evaluate the performance of the CACL algorithm on LFR benchmark networks, we report the community detection results in Table V and Fig. 2. The community detection results, as measured by EQ and ONMI, clearly show that the proposed CACL achieves superior performance compared to the baseline methods. This is primarily due to the integration of centrality-aware penalties and a collaborative learning framework, enabling the model to effectively manage overlapping communities and diverse network topologies. In contrast, while the EgoSplit and ECOCD algorithms deliver




(a) $on$ (b) $om$ (c) $n$ (d) $\mu$

Fig. 2. Performance comparison in terms of ONMI on ten LFR benchmark networks.

TABLE III
PARAMETER DESCRIPTION OF THE LFR BASELINE NETWORK

| Parameters | Descriptions | Value |
|---|---|---|
| $n$ | Number of nodes | Variable |
| $on$ | Number of overlapping nodes | Variable |
| $om$ | Number of overlapping memberships | Variable |
| $\mu$ | Mixing Parameter for Topology | Variable |
| $davg$ | Average node degree | 20 (Fixed) |
| $dmax$ | Maximum node degree | 50 (Fixed) |
| $t1$ | Node degree exponent | 2 (Fixed) |
| $t2$ | Community size exponent | 1 (Fixed) |
| $Cmin$ | Minimum community size | 50% of $n$ (Fixed) |
| $Cmax$ | Maximum community size | 10% of $n$ (Fixed) |

TABLE IV
PARAMETERS FOR DIFFERENT LFR BENCHMARK NETWORK DATA WHERE $k$ IS THE NUMBER OF COMMUNITIES

| Networks | $n$ | $k$ | $on$ | $om$ | $\mu$ | $davg$ | $dmax$ | $t1$ | $t2$ | $Cmin$ | $Cmax$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LFR1 | 1000 | 27 | 100 | 2 | 0.3 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR2 | 1000 | 20 | 200 | 2 | 0.3 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR3 | 1000 | 23 | 300 | 2 | 0.3 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR4 | 1000 | 23 | 100 | 3 | 0.3 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR5 | 1000 | 24 | 100 | 4 | 0.3 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR6 | 2000 | 24 | 100 | 2 | 0.3 | 20 | 50 | 2 | 1 | 40 | 200 |
| LFR7 | 3000 | 19 | 100 | 2 | 0.3 | 20 | 50 | 2 | 1 | 60 | 300 |
| LFR8 | 1000 | 23 | 100 | 2 | 0.2 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR9 | 1000 | 23 | 100 | 2 | 0.3 | 20 | 50 | 2 | 1 | 20 | 100 |
| LFR10 | 5000 | 19 | 100 | 2 | 0.3 | 20 | 50 | 2 | 1 | 100 | 500 |

strong results, they fall short of CACL's performance due to their heavy reliance on local ego-network partitioning. This approach is highly sensitive to initial community assignments and struggles to adapt to complex or large-scale networks. The DNMF and BigCLAM algorithms perform less effectively, primarily because they fail to capture higher-order network information, which limits their scalability in complex community structures. Furthermore, the QOCE and OCDDP algorithms consistently produce suboptimal results across all configurations. The poor performance of QOCE is due to its seed-set expansion strategy, which has difficulty handling sparse or highly overlapping communities, while OCDDP's density peak-based core selection mechanism proves inadequate for managing networks with varying community densities and overlap. Meanwhile, OUCoDe and CDNMF are originally designed for attributed networks, where node features play a key role in enhancing community detection performance. However, the LFR benchmark networks used in this study contain only topological information and lack node attributes. As a result, the contrastive learning mechanisms in CDNMF and the unified co-detection strategies in OUCoDe cannot fully exploit their advantages, leading to suboptimal performance under these conditions.

In further analysis, we explore how varying $on$ values influence the performance of the algorithms using the LFR1 to LFR3 datasets, as shown in Fig. 2(a). The findings reveal that CACL outperforms all baseline methods in terms of ONMI, regardless of the $on$ value. As seen in Fig. 2(a), most methods exhibit a decline in performance as $on$ increases from 100 to 300, which corresponds to the increasing complexity of the network as more nodes participate in multiple communities. On the other hand, the performance of OCDDP improves with rising $on$, although it remains relatively low, indicating that OCDDP faces challenges in capturing community structures in more intricate networks.

To further investigate the effect of varying $om$ values on the algorithms' performance, we conduct experiments on the LFR1, LFR4, and LFR5 datasets, as shown in Fig. 2(b). Both Fig. 2(a) and Fig. 2(b) demonstrate a consistent pattern: as either $on$ or $om$ increases, most methods show a performance decline. A higher $on$ value means more nodes belong to multiple communities, while a larger $om$ value indicates that nodes are part of more communities, making the task of community detection increasingly difficult. As a result, most methods see a reduction in performance, with ONMI values decreasing as the community structure becomes more complex. However, CACL maintains its superior performance, consistently outperforming all baseline methods in terms of ONMI.

We evaluated the performance of CACL across various network sizes, using the LFR1, LFR6, LFR7, and LFR10 datasets. These network sizes were selected to reflect the scale of representative real-world systems: LFR1 (1000 nodes) approximates scientific co-authorship networks such as Netscience [67]; LFR6 (2000 nodes) and LFR7 (3000 nodes) correspond to citation networks like Cora [68] or biological networks like CE-HT [69]; and LFR10 (5000 nodes) mirrors infrastructure networks such as the U.S. Power Grid [70]. The results presented in Fig. 2(c) show that while ECOCD, EgoSplit, and DNMF achieve strong performance, CACL consistently outperforms all competitors. Meanwhile, BigCLAM and OCDDP exhibit the lowest performance. As network size

TABLE V
PERFORMANCE COMPARISON OF MODULAR EQ IN SYNTHETIC NETWORKS. THE BEST VALUE IS INDICATED IN BOLD WHILE THE SECOND-BEST VALUE IS UNDERLINED

| Network | BigCLAM | EgoSplit | OCDDP | DNMF | QOCE | ECOCD | OUCoDe | CDNMF | CACL |
|---|---|---|---|---|---|---|---|---|---|
| LFR1 | 0.4710±0.0178 | 0.5768±0.0002 | 0.0161±0.0041 | 0.5654±0.0054 | 0.1450±0.0151 | 0.5752±0.0257 | 0.4802±0.0462 | 0.5173±0.1729 | **0.5795±0.0069** |
| LFR2 | 0.3928±0.0117 | 0.5166±0.0011 | 0.0234±0.0008 | 0.4222±0.0111 | 0.0490±0.0099 | 0.5154±0.0224 | 0.3921±0.0213 | 0.2128±0.0001 | **0.5194±0.0080** |
| LFR3 | 0.3707±0.0095 | 0.4721±0.0004 | 0.0299±0.0077 | 0.4509±0.0034 | 0.0500±0.0120 | 0.4612±0.0132 | 0.3765±0.0248 | 0.1801±0.0219 | **0.4769±0.0031** |
| LFR4 | 0.4365±0.0094 | 0.5599±0.0001 | 0.0228±0.0007 | 0.5543±0.0057 | 0.0700±0.0132 | 0.5410±0.0165 | 0.3906±0.0246 | 0.4362±0.1610 | **0.5621±0.0039** |
| LFR5 | 0.4413±0.0100 | 0.5497±0.0008 | 0.0277±0.0010 | 0.5366±0.0050 | 0.0930±0.0162 | 0.5438±0.0189 | 0.3990±0.0179 | 0.4251±0.1906 | **0.5519±0.0024** |
| LFR6 | 0.4364±0.0077 | 0.6191±0.0001 | 0.0402±0.0031 | 0.6053±0.0100 | 0.0990±0.0095 | 0.6192±0.0134 | 0.4094±0.0237 | 0.5049±0.2296 | **0.6203±0.0087** |
| LFR7 | 0.3345±0.0078 | 0.6185±0.0001 | 0.0321±0.0029 | 0.6184±0.0120 | 0.0830±0.0079 | 0.5961±0.0194 | 0.3504±0.0111 | 0.6185±0.0000 | **0.6188±0.0116** |
| LFR8 | 0.5701±0.0215 | 0.6657±0.0001 | 0.0342±0.0073 | 0.6536±0.0085 | 0.0920±0.0078 | 0.6664±0.0156 | 0.5245±0.0343 | 0.3620±0.2574 | **0.6689±0.0072** |
| LFR9 | 0.4500±0.0123 | 0.5760±0.0002 | 0.0372±0.0073 | 0.5763±0.0069 | 0.1020±0.0152 | 0.5767±0.0141 | 0.4328±0.0334 | 0.4050±0.2405 | **0.5786±0.0058** |
| LFR10 | 0.2756±0.0083 | **0.6251±0.0013** | 0.0436±0.0078 | 0.6026±0.0140 | 0.0201±0.0004 | 0.6165±0.0137 | 0.3321±0.0041 | 0.4631±0.0030 | **0.6251±0.0011** |
| Average | 0.4179±0.0116 | 0.5780±0.0004 | 0.0307±0.0043 | 0.5586±0.0082 | 0.0803±0.0107 | 0.5712±0.0173 | 0.4088±0.0241 | 0.4125±0.1475 | **0.5802±0.0059** |

increases, ONMI values for CACL and EgoSplit improve, indicating the robustness and stability of CACL in large-scale networks, while other algorithms tend to see a decrease in performance with growing network size. We also observe that OUCoDe maintains relatively stable performance across different network sizes, although its effectiveness does not reach that of CACL. In contrast, CDNMF exhibits a significant decline in ONMI values on the LFR10 dataset, suggesting that its performance is less stable on networks lacking attribute information. Overall, the results confirm that CACL maintains the highest ONMI values across all network sizes.

Finally, we assess the performance of the proposed CACL with varying $\mu$ values on the LFR1, LFR8, and LFR9 datasets, as shown in Fig. 2(d). The results indicate that CACL's ONMI values consistently improve as $\mu$ increases, further validating its adaptability and excellent performance across different network configurations. This reinforces that CACL remains highly effective in handling networks with varying degrees of complexity and overlapping structures, solidifying its superior performance.

### D. Experiments on Real-World Networks

*1) Datasets:* To evaluate the performance of the proposed CACL method, we utilize six real-world, overlapping, undirected, and unweighted social networks[1][2] from different domains. Table VI provides a summary of these networks, where $n$ is the total number of nodes, $m$ denotes the total number of edges, and $k$ represents the number of pre-set communities. We determine the number of communities $k$ detected by BigCLAM, DNMF, and CACL using the SNMF-method [42], exploring values of $k$ from 5 to 50 with a step size of 5. The optimal $k$ for each network, based on the highest EQ, is presented in Table VI.

*2) Experimental Results:* The experimental results are presented in Table VII, which details the EQ values for each real network. These results demonstrate that the CACL algorithm consistently achieves the highest EQ metrics across multiple networks, including Karate, Dolphins, Jazz, Metabolic, and Netscience. While ECOCD performs exceptionally well in the Football network, its EQ scores across other datasets exhibit larger variability, highlighting its inconsistency. In

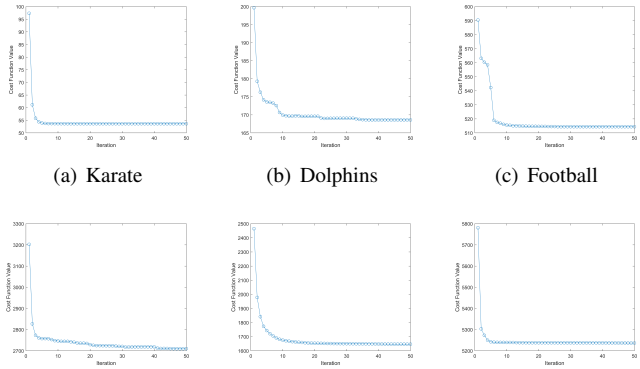
TABLE VI
SUMMARY OF THE REAL-WORLD NETWORKS

| Network | $n$ | $m$ | $k$ |
|---|---|---|---|
| Karate | 34 | 78 | 5 |
| Dolphins | 62 | 159 | 5 |
| Football | 115 | 613 | 10 |
| Jazz | 198 | 2742 | 5 |
| Metaboic | 453 | 4596 | 20 |
| Netscience | 1589 | 2742 | 35 |

terms of average EQ scores, CACL solidifies its superiority with a significant lead over other methods, followed by DNMF and ECOCD in second and third place, respectively. Interestingly, the rankings of algorithms in real-world networks differ from those in LFR synthetic datasets due to fundamental structural differences. LFR networks are generated with controlled parameters, such as the mixing ratio $\mu$, overlapping node ratio $on$, and membership per node $om$, resulting in regular and predictable structures. These controlled settings tend to benefit algorithms like ECOCD and EgoSplit, which rely on assumptions about community density and boundary sharpness. In contrast, real-world networks are characterized by irregularities, such as heterogeneous degree distributions, sparse connectivity, and ambiguous community boundaries. These complexities challenge methods such as EgoSplit and QOCE, which are heavily dependent on localized partitioning or seed expansion strategies. In contrast, they underscore the effectiveness of CACL in addressing such irregularities. This superior performance of CACL is largely attributed to its ability to address two key challenges in overlapping community detection. The first challenge arises from highly centralized nodes with multiple community memberships, which can lead to dense overlaps and hinder accurate network embeddings. CACL overcomes this issue by employing a centrality-aware penalty mechanism that mitigates the influence of high-centrality nodes while preserving essential network structures. This approach enables precise and learnable representations of overlapping memberships. The second challenge stems from the limitations of single-model approaches, which often fail to balance local and global network features, leading to suboptimal embeddings. CACL addresses this by incorporating a collaborative learning framework that combines symmetric non-negative matrix factorization and kernel regression. By leveraging the strengths of multiple models, CACL effectively

[1] http://www-personal.umich.edu/~mejn/netdata/
[2] https://linqs.soe.ucsc.edu/

captures both first-order and higher-order network information, enabling it to adapt to diverse and complex network configurations. These innovations explain why CACL consistently outperforms methods like DNMF and BigCLAM, which are constrained by their reliance on single-model designs.



(a) Karate (b) Dolphins (c) Football

(d) Jazz (e) Metabolic (f) Netscience

Fig. 3. Convergence analysis of different real-world networks.

*3) Convergence:* In this part, we analyze the convergence behavior of various real-world network datasets using the CACL algorithm. Figure 3 illustrates the algorithm's stability across diverse network structures, emphasizing its efficiency in attaining optimal solutions. This is particularly notable in networks such as Karate, Football, Metabolic, and Netscience, where the objective function values stabilize swiftly after the initial iterations. In the Dolphins network, a sharp decline in the objective function value is observed during the initial stages, followed by gradual stabilization, becoming apparent after approximately 30 iterations. A similar convergence pattern is observed in the Jazz network, where the objective function stabilizes within 40 iterations. Across all networks, the CACL algorithm consistently converges within 50 iterations. This rapid convergence highlights the algorithm's robustness in adapting to diverse network structures, facilitating the swift identification of optimal or near-optimal community divisions.

## E. Parameter Investigation

In this section, we investigate how different parameters of the proposed CACL framework affect its performance in overlapping community detection. Specifically, we analyze the parameters $\alpha$, $k$, $\beta$, $\gamma$, and $\lambda$ in Algorithm 1.

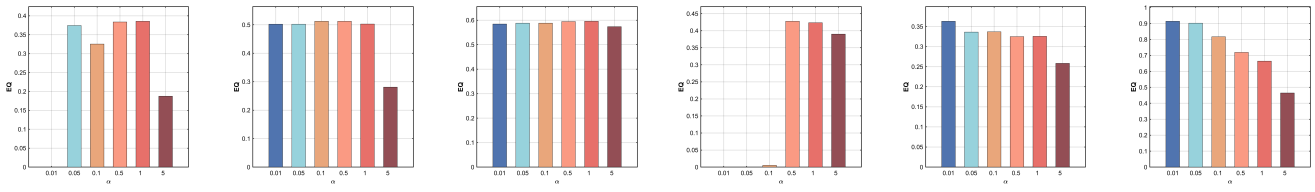
*1) Impact of $\alpha$:* The parameter $\alpha$ controls the balance between the generation of discrete community membership based on SNMF and other terms in Eq. (18). The compared results of EQ across six real-world networks with varying values of $\alpha$ are presented in Fig. 4. As $\alpha$ increases, the objective function gradually concentrates on the soft community matrix $\mathbf{U}$ as it approaches the hard community matrix $\mathbf{F}$. This shift causes the community division to gravitate towards hard partitioning, thereby reducing overlap and ambiguity. In contrast, when $\alpha$ is lower, the weight of the soft community matrix increases, facilitating the capture of overlapping community structures within complex networks. In the Karate, Dolphins, Football, and Jazz datasets, optimal EQ values are achieved when $\alpha$ falls within the range of 0.05 to 1. These datasets are characterized by high interconnectivity within social networks and small-scale community structures, in which moderate values of $\alpha$ (typically between 0.1 and 0.5) are more effective in accurately representing the community structure. In contrast, the Metabolic and Network Science datasets exhibit different behavior; in these cases, lower $\alpha$ values (e.g., below 0.01) yield superior community detection performance.

*2) Impact of $k$:* We show the effects of the number of communities on the performance of CACL in Fig. 5 with varying values of $k$ ranging from 5 to 50 in increments of 5, where the star symbol highlights the optimal outcome. Figure 5 validates the suitability of the predetermined values $k$ presented in Table VI, confirming the effectiveness of the CACL algorithm in identifying the optimal number of communities for each dataset. This validation highlights the algorithm's flexibility and accuracy in detecting suitable community structures.

*3) Impacts of $\beta$, $\gamma$, and $\lambda$:* In addition, we conduct experiments utilizing the proposed CACL across six real-world networks with varying values of $\beta$ and $\gamma$. The comparative EQ results vs. different cobinations of $\beta$ and $\gamma$ are shown in Fig. 6. Mathematically, the parameters $\beta$ and $\gamma$ influence the weights of the KR model within the overall objective function. The EQ results depicted in Fig. 6 indicate that the CACL algorithm consistently outperforms others, provided that the values of $\beta$ and $\gamma$ are not set too high simultaneously, as this would disrupt the balance of the model's contributions and negatively affect EQ. To assess the influence of $\lambda$, we conducted experiments on six real-world networks and present the EQ results across different $\lambda$ values in Fig. 7. The results reveal that lower values of $\lambda$ are associated with higher EQ scores, indicating that reducing $\lambda$ enhances the algorithm's ability to detect more accurate and meaningful community structures within the networks.

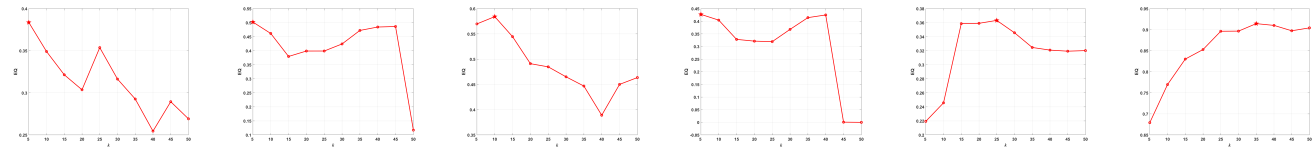
TABLE VII
PERFORMANCE COMPARISON OF MODULAR EQ IN REAL-WORLD NETWORKS. THE BEST VALUE IS INDICATED IN BOLD WHILE THE SECOND-BEST VALUE IS UNDERLINED. VALUES ARE PRESENTED AS MEAN ± STANDARD DEVIATION.

| Network | BigCLAM | EgoSplit | OCDDP | DNMF | QOCE | ECOCD | OUCoDe | CDNMF | CACL |
|---|---|---|---|---|---|---|---|---|---|
| Karate | 0.2928±0.1385 | 0.1757±0.0090 | 0.3702±0.0174 | 0.3499±0.0326 | 0.3190±0.0096 | 0.3710±0.0358 | 0.2638±0.1391 | 0.3280±0.0200 | **0.3749±0.0296** |
| Dolphins | 0.3423±0.1663 | 0.1940±0.0043 | 0.4680±0.0177 | 0.5053±0.0093 | 0.3710±0.0424 | 0.3850±0.0395 | 0.2156±0.1011 | 0.2121±0.0092 | **0.5056±0.0195** |
| Football | 0.1320±0.0251 | 0.1280±0.0000 | 0.5810±0.0515 | 0.5904±0.0096 | 0.2600±0.0039 | **0.5954±0.0044** | 0.4026±0.1491 | 0.5796±0.0031 | 0.5904±0.0051 |
| Jazz | 0.1775±0.0837 | 0.3830±0.0025 | 0.2576±0.0186 | 0.4016±0.0031 | 0.2150±0.0119 | 0.4170±0.0144 | 0.1232±0.0786 | 0.1868±0.0133 | **0.4203±0.0142** |
| Metabolic | 0.0882±0.0254 | 0.3360±0.0043 | 0.1230±0.0157 | 0.3509±0.0135 | 0.0524±0.0268 | 0.1068±0.0188 | 0.2272±0.0455 | 0.1373±0.0744 | **0.3529±0.0136** |
| Netscience | 0.2642±0.0259 | 0.7690±0.0018 | 0.8041±0.0274 | 0.8695±0.0098 | 0.2780±0.0051 | 0.7640±0.0190 | 0.7990±0.0171 | 0.8487±0.0726 | **0.8924±0.0123** |
| Average | 0.2162±0.0775 | 0.3310±0.0037 | 0.4340±0.0247 | 0.5113±0.0130 | 0.2492±0.0166 | 0.4399±0.0220 | 0.3386±0.0884 | 0.3821±0.0321 | **0.5228±0.0157** |



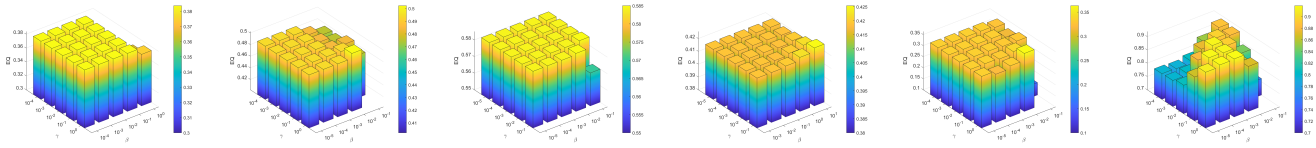
(a) Karate (b) Dolphins (c) Football (d) Jazz (e) Metabolic (f) Netscience

Fig. 4. The compared results of EQ across six real-world networks with varying values of $\alpha$.



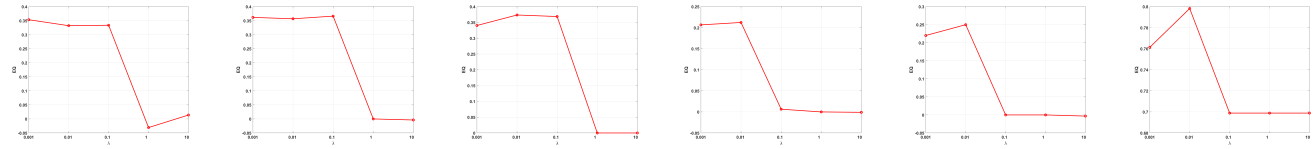
(a) Karate (b) Dolphins (c) Football (d) Jazz (e) Metabolic (f) Netscience

Fig. 5. The comparison of EQ using the proposed CACL across six real-world networks with varying values of $k$, where the star mark indicates the best result.



(a) Karate (b) Dolphins (c) Football (d) Jazz (e) Metabolic (f) Netscience

Fig. 6. The comparison of EQ using the proposed CACL across six real-world networks with varying values of $\beta$ and $\gamma$.



(a) Karate (b) Dolphins (c) Football (d) Jazz (e) Metabolic (f) Netscience

Fig. 7. The comparison of EQ using the proposed CACL across six real-world networks with varying values of $\lambda$.

### *F. Ablation Study*

In this section, we investigate the effectiveness of the following key components: (1) the contribution of collaborative learning by setting $\alpha = 0$ and $\beta = 0$ and preserving only the SNMF model in the overall objective, named as CACL_1; (2) the role of Higher-Order Adjacency Construction (dubbed as HAC) by setting $\lambda = 0$ in the overall objective, named as CACL_2; (3) the impact of the Centrality Penalty mechanism for nodes (dubbed as CP) by setting $\eta = 0$ in the overall objective, named as CACL_3. Table VIII defines three ablated variants of the CACL model.

TABLE VIII
ABLATION STUDY DESIGN OF CACL VARIANTS

| Methods | SNMF | KR | CP | HAC |
|---|---|---|---|---|
| CACL_1 | ✔ | ✘ | ✔ | ✔ |
| CACL_2 | ✔ | ✔ | ✘ | ✔ |
| CACL_3 | ✔ | ✔ | ✔ | ✘ |

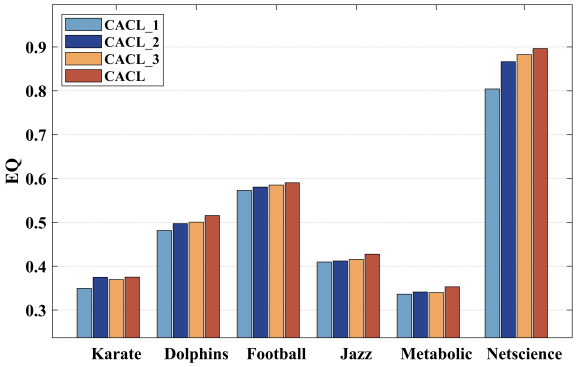



Fig. 8. EQ values on six real-world networks by CACL and its three variants.

We conduct comparative experiments using the proposed CACL model and its three ablated variants. Figure 8 reports

the average ONMI scores over 10 independent runs on six real-world networks, aiming to assess the individual contributions of each core module. From the results in Fig. 8, we can conclude the following observations:

- CACL_1 *vs.* CACL: The comparison between CACL_1 and CACL reveals that the full CACL model, with both KR and SNMF modules, consistently outperforms CACL_1 across the six real-world networks. The presence of the KR module significantly boosts the performance, highlighting the importance of collaborative learning in improving overlapping community detection accuracy. Therefore, the SNMF model alone (CACL_1) does not fully capture the complexities of the datasets, demonstrating the value of integrating additional KR module in the overall framework.
- CACL_2 *vs.* CACL: When comparing CACL_2 to the full CACL model, the results show that removing the CP module (setting $\lambda = 0$) slightly reduces the performance across most datasets, although the performance difference is minimal on the Karate and Dolphins datasets, where CACL_2 actually performs slightly better than CACL_3. This suggests that while the CP module contributes to the model's robustness, its role does slightly impact performance in some datasets.
- CACL_3 *vs.* CACL: The comparison between CACL_3 and CACL indicates that omitting the HAC module (by setting $\eta = 0$) results in a decrease in performance across all datasets. The HAC module plays a crucial role in improving classification performance, emphasizing the importance of centrality-based penalties in better capturing the underlying network structure. This conclusion underscores the need for integrating higher-order adjacency construction to enhance the model's ability to handle the overlapping community detection tasks.
- Summary of Results: It is evident that both CP and HAC contribute to performance improvements across different datasets by comparing the various ablated models with the CACL full model. However, when only the single SNMF model is used, the community detection accuracy experiences the most significant decline. This suggests that collaborative learning plays a crucial role in the overall framework and makes a significant contribution. Therefore, this demonstrates the effectiveness of incorporating collaborative learning within the proposed CACL framework and highlights the necessity of integrating CP and HAC.

### G. Time Performance Analysis

Figure 9 presents the average runtime comparison between the proposed CACL model and baseline methods. We evaluate the computational efficiency of CACL by measuring the average running time over ten trials on six real-world networks and comparing it against eight representative algorithms. The CACL model leverages a row-based sparse storage format, which significantly reduces computational overhead. As shown in the Fig. 9, CACL demonstrates faster execution than all baseline methods except for EgoSplit, which is based on ego-network decomposition. It is also worth noting that CDNMF, which incorporates a contrastive learning mechanism and deep matrix factorization, incurs high initialization and training costs. As a result, its runtime is relatively high on smaller datasets, but it becomes more efficient on larger networks due to its enhanced scalability. Despite this, the detection performance of CDNMF still falls short compared to CACL. Overall, CACL strikes an effective balance between computational efficiency and detection performance, making it a competitive choice for overlapping community detection tasks.

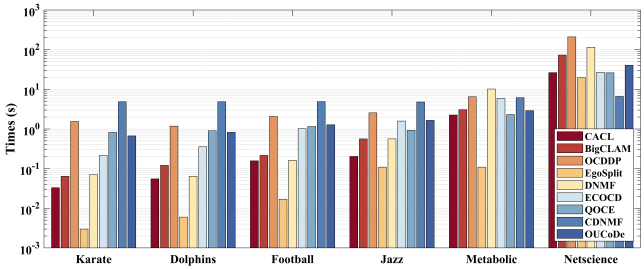
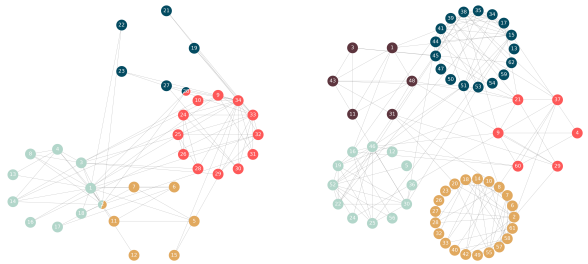



Fig. 9. Average running time comparison of the baselines and CACL on six real-world networks.



(a) Karate (b) Dolphins

Fig. 10. Community detection results on Karate and Dolphins networks using the CACL model. Nodes with the same color belong to the same community. Multicolored nodes represent overlaps.

### H. Visualization

To intuitively evaluate the quality of community detection, we visualize the community structures identified by the proposed CACL model on two standard real-world networks: Karate and Dolphins. As illustrated in Fig. 10, the community divisions produced by CACL exhibit clear structural separation. In these visualizations, nodes with multiple colors represent overlapping nodes that belong to more than one community. For example, in the Karate network, node 2 and node 20 are detected as overlapping nodes, each affiliated with two communities. In contrast, the Dolphins network contains no overlapping nodes. Importantly, these detected overlaps are not arbitrary but represent nodes in critical bridging positions, analogous to "structural hole spanners" from social network theory [46]. These nodes are functionally vital for network cohesion and inter-community communication. By accurately identifying these key bridges, CACL demonstrates its capacity to reveal not only clear structural separations but also the essential functional connectivity of the network.

## V. Conclusion

In this paper, we propose a novel CACL approach for overlapping community detection that collaboratively integrates SNMF and KR models to enhance network embedding. In addition to incorporating both first-order and second-order adjacency information, the CACL framework features a specially designed centrality penalty mechanism to optimize the distribution of community memberships in complex scenarios, where core nodes may simultaneously belong to multiple communities. Extensive experiments on ten synthetic and six real-world networks demonstrate the superiority of the proposed CACL approach over competitive overlapping community detection algorithms, highlighting its significant effectiveness in improving detection accuracy through collaborative learning for network embedding. As part of our future work, we aim to explore collaborative learning for multi-layer overlapping community detection to further enhance the model's performance and applicability.

## References


[1] L. Brahim, M. Mouad, C. Chihab-Eddine, and I. Ali, "A survey on community detection: Applications, algorithms, and challenges," *J. Theor. Appl. Inf. Techn.*, vol. 102, no. 12, pp. 4923–4945, 2024.

[2] L. M. Sanchez-Rodriguez, Y. Iturria-Medina, P. Mouches, and R. C. Sotero, "Detecting brain network communities: Considering the role of information flow and its different temporal scales," *NeuroImage*, vol. 225, p. 117431, 2021.

[3] A. K. Sangaiah, S. Rezaei, A. Javadpour, and W. Zhang, "Explainable AI in big data intelligence of community detection for digitalization e-healthcare services," *Appl. Soft Comput.*, vol. 136, p. 110119, 2023.

[4] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, 2016.

[5] L. Lv, D. Bardou, Y. Liu, and P. Hu, "Deep autoencoder-like non-negative matrix factorization with graph regularized for link prediction in dynamic networks," *Appl. Soft Comput.*, vol. 148, p. 110832, 2023.

[6] J. Xiao, Y.-F. Guo, Y.-Q. He, and X.-K. Xu, "Constrained fuzzy community detection by a new modularity optimization framework," *IEEE Trans. Network Sci. Eng.*, vol. 11, no. 5, pp. 4456–4469, 2024.

[7] Z. Zheng, X. Chen, and X. Lin, "Kernel based dual-channel attributed graph community detection," *IEEE Trans. Network Sci. Eng.*, vol. 11, no. 1, pp. 592–603, 2024.

[8] Q. Zhou, W. Zhu, H. Chen, and B. Peng, "Community detection in multiplex networks by deep structure-preserving non-negative matrix factorization," *Appl. Intell.*, vol. 55, no. 1, p. 26, 2025.

[9] S. S. Singh, S. Muhuri, S. Mishra, D. Srivastava, H. K. Shakya, and N. Kumar, "Social network analysis: A survey on process, tools, and application," *ACM Comput. Surv.*, vol. 56, no. 8, pp. 1–39, 2024.

[10] G. Pirrò, "Community deception from a node-centric perspective," *IEEE Trans. Network Sci. Eng.*, vol. 11, no. 1, pp. 969–981, 2023.

[11] C. Chen, W. Zhu, and B. Peng, "Differentiated graph regularized non-negative matrix factorization for semi-supervised community detection," *Physica A*, vol. 604, p. 127692, 2022.

[12] X. Shen, X. Yao, H. Tu, and D. Gong, "Parallel multi-objective evolutionary optimization based dynamic community detection in software ecosystem," *Knowledge-Based Syst.*, vol. 252, p. 109404, 2022.

[13] Y. Zhang, C. Wu, Y. Tian, and X. Zhang, "A co-evolutionary algorithm based on sparsity clustering for sparse large-scale multi-objective optimization," *Eng. Appl. Artif. Intel.*, vol. 133, p. 108194, 2024.

[14] J. Zhu, C. Wang, C. Gao, F. Zhang, Z. Wang, and X. Li, "Community detection in graph: An embedding method," *IEEE Trans. Network Sci. Eng.*, vol. 9, no. 2, pp. 689–702, 2022.

[15] H. Cheng, C. He, H. Liu, X. Liu, P. Yu, and Q. Chen, "Community detection based on directed weighted signed graph convolutional networks," *IEEE Trans. Network Sci. Eng.*, vol. 11, no. 2, pp. 1642–1654, 2024.

[16] W. Zheng, J. Sun, Q. Zhang, and Z. Xu, "Continuous encoding for overlapping community detection in attributed network," *IEEE Trans. Cybern.*, vol. 53, no. 9, pp. 5469–5482, 2023.

[17] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. Knowl. Discov. Data*, vol. 8, no. 1, pp. 1–28, 2014.

[18] A. Bouyer, H. Ahmadi Beni, B. Arasteh, Z. Aghaee, and R. Ghanbarzadeh, "FIP: A fast overlapping community-based influence maximization algorithm using probability coefficient of global diffusion in social networks," *Expert Syst. Appl.*, vol. 213, p. 118869, 2023.

[19] H. Qing and J. Wang, "Bipartite mixed membership distribution-free model. A novel model for community detection in overlapping bipartite weighted networks," *Expert Syst. Appl.*, vol. 235, p. 121088, 2024.

[20] A. Ramesh and G. Srivatsun, "Evolutionary algorithm for overlapping community detection using a merged maximal cliques representation scheme," *Appl. Soft Comput.*, vol. 112, p. 107746, 2021.

[21] Y. Yang, P. Shi, Y. Wang, and K. He, "Quadratic optimization based clique expansion for overlapping community detection," *Knowledge-Based Syst.*, vol. 247, p. 108760, 2022.

[22] R. Yan, W. Yuan, X. Su, and Z. Zhang, "FLPA: A fast label propagation algorithm for detecting overlapping community structure," *Expert Syst. Appl.*, vol. 234, p. 120971, 2023.

[23] L. Zhang, B. Li, H. Yang, F. Cheng, C. Zhang, and R. Cao, "Multi-objective optimization of local overlapping community detection: A formal model and novel evolutionary algorithm," *IEEE Trans. Network Sci. Eng.*, vol. 10, no. 4, pp. 2124–2140, 2023.

[24] R. Shang, S. Wang, W. Zhang, J. Feng, L. Jiao, and R. Stolkin, "Evolutionary multi-objective overlapping community detection based on fusion of internal and external connectivity and correction of node intimacy," *Appl. Soft Comput.*, vol. 154, p. 111414, 2024.

[25] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proc. ACM Int. Conf. Web Search Data Min.*, 2013, pp. 587–596.

[26] C. He, Y. Zheng, J. Cheng, Y. Tang, G. Chen, and H. Liu, "Semi-supervised overlapping community detection in attributed graph with graph convolutional autoencoder," *Inf. Sci.*, vol. 608, pp. 1464–1479, 2022.

[27] Y. Rashid and J. I. Bhat, "OlapGN: A multi-layered graph convolution network-based model for locating influential nodes in graph networks," *Knowledge-Based Syst.*, vol. 283, p. 111163, 2024.

[28] P. G. Sun, X. Wu, Y. Quan, and Q. Miao, "Rearranging 'indivisible' blocks for community detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 6252–6263, 2022.

[29] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[30] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[31] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, p. 103018, 2010.

[32] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, "A link clustering based overlapping community detection algorithm," *Data Knowl. Eng.*, vol. 87, pp. 394–404, 2013.

[33] A. Ponomarenko, L. Pitsoulis, and M. Shamshetdinov, "Overlapping community detection in networks based on link partitioning and partitioning around medoids," *PLoS One*, vol. 16, no. 8, p. e0255717, 2021.

[34] J. Yu, Y. Liu, W. Liang, X. Han, and N. N. Xiong, "A framework for overlapping and non-overlapping communities detection based on seed extension and label propagation," *Physica A*, vol. 660, p. 130362, 2025.

[35] W. Zhu and Y. Peng, "Elastic net regularized kernel non-negative matrix factorization algorithm for clustering guided image representation," *Appl. Soft Comput.*, vol. 97, p. 106774, 2020.

[36] M. Chen, M. Gong, and X. Li, "Feature weighted non-negative matrix factorization," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 1093–1105, 2023.

[37] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[38] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using bayesian non-negative matrix factorization," *Phys. Rev. E*, vol. 83, no. 6, p. 066114, 2011.

[39] M. Sørensen, N. D. Sidiropoulos, and A. Swami, "Overlapping community detection via semi-binary matrix factorization: Identifiability and algorithms," *IEEE Trans. Signal Process.*, vol. 70, pp. 4321–4336, 2022.

[40] Y. Jia, H. Liu, J. Hou, and S. Kwong, "Semisupervised adaptive symmetric non-negative matrix factorization," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2550–2562, 2020.

[41] Z. Liu, X. Luo, and M. Zhou, "Symmetry and graph bi-regularized non-negative matrix factorization for precise community detection," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1406–1420, 2024.

[42] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Min. Knowl. Disc.*, vol. 22, pp. 493–521, 2011.

[43] F. Ye, C. Chen, Z. Zheng, R.-H. Li, and J. X. Yu, "Discrete overlapping community detection with pseudo supervision," in *Proc. IEEE Int. Conf. Data Min.*, 2019, pp. 708–717.

[44] M. Huang, Q. Jiang, Q. Qu, and A. Rasool, "An overlapping community detection approach in ego-splitting networks using symmetric nonnegative matrix factorization," *Symmetry*, vol. 13, no. 5, p. 869, 2021.

[45] Z. Zhuo, B. Chen, S. Yu, and L. Cao, "Overlapping community detection using expansion with contraction," *Neurocomputing*, vol. 565, p. 126989, 2024.

[46] R. S. BURT, *Structural Holes: The Social Structure of Competition*. Harvard University Press, 1992.

[47] T. Lou and J. Tang, "Mining structural hole spanners through information diffusion in social networks," in *Proc. Int. Conf. World Wide Web*, 2013, pp. 825–836.

[48] L. He, C.-T. Lu, J. Ma, J. Cao, L. Shen, and P. S. Yu, "Joint community and structural hole spanner detection via harmonic modularity," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 875–884.

[49] Q. Gong, S. Gu, X. Wang, J. Yao, P. Hui, J.-D. Luo, X. Fu, and Y. Chen, "DeepHole: Identifying structural hole spanners in online social networks using behavior embedding," *IEEE Trans. Comput. Social Syst.*, 2025.

[50] W. Zhu, C. Chen, and B. Peng, "Unified robust network embedding framework for community detection via extreme adversarial attacks," *Inf. Sci.*, vol. 643, p. 119200, 2023.

[51] L. Huang, C.-D. Wang, and P. S. Yu, "Higher order connection enhanced community detection in adversarial multiview networks," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3060–3074, 2023.

[52] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 203–209.

[53] V. Jannesari, M. Keshvari, and K. Berahmand, "A novel nonnegative matrix factorization-based model for attributed graph clustering by incorporating complementary information," *Expert Syst. Appl.*, vol. 242, p. 122799, 2024.

[54] M. R. F. Mendona, A. M. S. Barreto, and A. Ziviani, "Approximating network centrality measures using node embedding and machine learning," *IEEE Trans. Network Sci. Eng.*, vol. 8, no. 1, pp. 220–230, 2021.

[55] E. Alpaydin, *Introduction to Machine Learning*. MIT press, 2020.

[56] Z. Ghalmane, M. El Hassouni, C. Cherifi, and H. Cherifi, "Centrality in modular networks," *EPJ Data Sci.*, vol. 8, no. 1, p. 15, 2019.

[57] S. Rajeh, M. Savonnet, E. Leclercq, and H. Cherifi, "Comparative evaluation of community-aware centrality measures," *Qual. Quant.*, vol. 57, no. 2, pp. 1273–1302, 2023.

[58] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 902–913, 2010.

[59] M. Gao, Z. Li, R. Li, C. Cui, X. Chen, B. Ye, Y. Li, W. Gu, Q. Gong, X. Wang, and Y. Chen, "EasyGraph: A multifunctional, cross-platform, and effective library for interdisciplinary network analysis," *Patterns*, vol. 4, no. 10, p. 100839, 2023.

[60] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A*, vol. 388, no. 8, pp. 1706–1712, 2009.

[61] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, p. 033015, 2009.

[62] A. Epasto, S. Lattanzi, and R. Paes Leme, "Ego-splitting framework: From non-overlapping to overlapping clusters," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 145–154.

[63] X. Bai, P. Yang, and X. Shi, "An overlapping community detection algorithm based on density peaks," *Neurocomputing*, vol. 226, pp. 7–15, 2017.

[64] A. Moradan, A. Draganov, D. Mottin, and I. Assent, "UCoDe: Unified community detection with graph convolutional networks," *Mach. Learn.*, vol. 112, no. 12, pp. 5057–5080, 2023.

[65] Y. Li, J. Chen, C. Chen, L. Yang, and Z. Zheng, "Contrastive deep nonnegative matrix factorization for community detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 6725–6729.

[66] C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," *Commun. ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[67] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, p. 036104, 2006.

[68] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: A discriminative approach," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2009, pp. 927–936.

[69] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015.

[70] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.