

A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning

Author

Jin, Di, Yu, Zhizhi, Jiao, Pengfei, Pan, Shirui, He, Dongxiao, Wu, Jia, Yu, Philip S, Zhang, Weixiong

Published

2023

Journal Title

IEEE Transactions on Knowledge and Data Engineering

Version

Accepted Manuscript (AM)

DOI

[10.1109/TKDE.2021.3104155](https://doi.org/10.1109/TKDE.2021.3104155)

Rights statement

This work is covered by copyright. You must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a specified licence, refer to the licence for details of permitted re-use. If you believe that this work infringes copyright please make a copyright takedown request using the form at <https://www.griffith.edu.au/copyright-matters>.

Downloaded from

<http://hdl.handle.net/10072/429385>

Griffith Research Online

<https://research-repository.griffith.edu.au>

A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning

Di Jin, Zhizhi Yu, Pengfei Jiao, Shirui Pan, Dongxiao He, Jia Wu,
Philip S. Yu and Weixiong Zhang

Abstract—Community detection, a fundamental task for network analysis, aims to partition a network into multiple sub-structures to help reveal their latent functions. Community detection has been extensively studied in and broadly applied to many real-world network problems. Classical approaches to community detection typically utilize probabilistic graphical models and adopt a variety of prior knowledge to infer community structures. As the problems that network methods try to solve and the network data to be analyzed become increasingly more sophisticated, new approaches have also been proposed and developed, particularly those that utilize deep learning and convert networked data into low dimensional representation. Despite all the recent advancement, there is still a lack of insightful understanding of the theoretical and methodological underpinning of community detection, which will be critically important for future development of the area of network analysis. In this paper, we develop and present a unified architecture of network community-finding methods to characterize the state-of-the-art of the field of community detection. Specifically, we provide a comprehensive review of the existing community detection methods and introduce a new taxonomy that divides the existing methods into two categories, namely probabilistic graphical model and deep learning. We then discuss in detail the main idea behind each method in the two categories. Furthermore, to promote future development of community detection, we release several benchmark datasets from several problem domains and highlight their applications to various network analysis tasks. We conclude with discussions of the challenges of the field and suggestions of possible directions for future research.

Index Terms—Complex Network, Community Detection, Graph Clustering, Statistical Modeling, Deep Learning.



1 INTRODUCTION

NETWORK science is the study of complex systems in the form of networks using theories and techniques of computer science, mathematics and physics. In particular, network structures [1] (see an example in Fig 1) have been studied extensively under the notions of subgraphs, network modules, and communities. Identification of network structures or community detection is to divide nodes in a network into groups where the nodes in a group are densely connected whereas nodes in different groups are sparsely linked. Mining network structures is also the key to revealing and comprehending organizational principles and operational functions of complex network systems. For example, community detection has been applied to recommendation [2], [3], anomaly detection [4], [5], and terrorist organization identification [6], just to name a few. Much effort has also been devoted to the analysis of network structural properties, e.g., the small world effect (i.e., the average distance between nodes is short [8]) and the scale free property (i.e., the distribution of node degrees follows a power law distribution [9]).

Many algorithms for community detection have been proposed, a majority of which employ exclusively the information of network topology. They include hierarchical

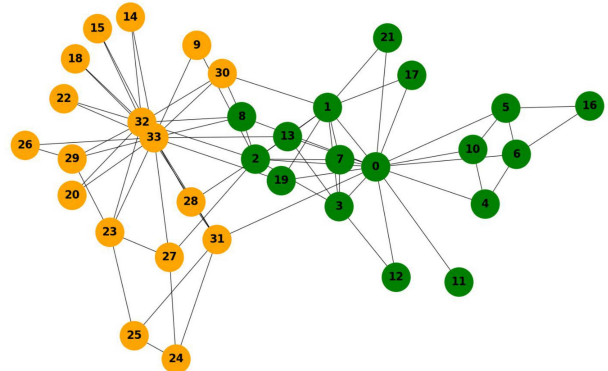


Fig. 1: An illustrative example (Zachary's karate club network [7]) showing community structure. The nodes of this network are divided into two groups, with most connections falling within groups and only a few between groups.

clustering [10] [11], modularity optimization [12] [13] [14], spectral clustering [15] [16] and statistical inference [17] [18]. New methods were developed to utilize node semantics or node attributes in addition to network topology to improve the quality of resulting communities and meanwhile provide explanation to the results. These include heuristic optimization (multi-objective) [19] [20], matrix factorization [21] [22] and Bayesian model [23]. As more complex network problems were tackled, complex network data from multiple sources, e.g., network topology and node semantics, must be effectively integrated. As a result, it became difficult for these traditional approaches to perform data fusion effectively on data of very high dimensions and diverse

- D. Jin, Z. Yu, P. Jiao and D. He are with Tianjin University, China.
- S. Pan is with Monash University, Australia.
- J. Wu is with Macquarie University, Australia.
- P. S. Yu is with University of Illinois at Chicago, USA.
- W. Zhang is with Washington University, USA.
- Corresponding author: Pengfei Jiao.

properties. The technique of deep learning was recently adopted to handle the high dimensional network data and learn low-dimensional representation of network structures. Examples include methods based on the auto-encoder [24] [25] and the generative adversarial approach [26] [27].

An important and effective idea for community detection is to learn an adequate representation of the network structure of a given network. We call such approaches *learning-based community detection*. Among these methods are the model-based generative models. The most popular and representative example is the stochastic block model (SBM) [28], which detects communities by formalizing a generative process of a network as a sequence of rigorous probability distributions. Several extensions and improvements have been introduced to boost the performance of SBM [29] [30]. Another model-based learning approach adopts Markov random field (MRF), an undirected graphical model, to take advantage of neighborhood structures in networks [31]. A primary recent development in learning-based methods exploits the low-dimensional representation capability of deep learning. For instance, convolutional neural network (CNN) [32] utilizes convolution and pooling operations to reduce the dimensions of network data, so as to effectively discover communities in networks. Graph convolutional network (GCN) [33], which inherits the advantages of CNN and directly operates on network structured data, has also been explored to derive community representation [34].

Despite the endured effort to develop effective methods for community detection [35], [36], there is still a lack of understanding of the theoretical and methodological underpinning of community detection, particularly that based on learning. To bridge this gap, in this paper, we will provide a synthesized survey of the existing representative methods. We focus particularly on two general lines of approaches, one based on probabilistic modeling and the other on deep learning. We start with a detailed description of each line of the work and provide a thorough review and comparison of the methods. We then consider several applications of community detection in diverse fields. We finish with the discussion of some critical challenges of the field of network analysis and directions for future rewarding research.

One of the major objectives of our survey is to provide a new perspective on the existing methods to help better understand the fundamental issues and enabling techniques for community detection. Our survey differs from the published ones in three aspects. First, we summarize the existing methods by focusing on learning, a central issue of community modeling and community detection, whereas the existing reviews [37] [38] generally discuss the chronological development of the existing methods. Second, we present a recent trend in the development of methods for community detection, i.e., from statistical modeling to deep learning, while the others focus mainly on individual techniques, e.g., evolutionary computation [39], statistical inference [40] or deep learning [41]. Third, we present a unified system architecture to characterize the existing methods, which provides a novel and synthesized perspective on methods based on statistical modeling and deep learning, which go significantly beyond some of the existing surveys [42] [43]. Last but not least, in the era of deep learning, as network data become increasingly more complex and various ideas

and techniques have been proposed, a survey is urgently needed to comprehensively unravel the inherent relationships among the existing methods for community detection.

Aiming at offering a general guidance to researchers and practitioners who are interested in network science and network data analysis, we make our unique contributions in this work as summarized below.

- We present the most comprehensive and extensive overview of learning-based community detection and divide them into two categories, probabilistic graphical model and deep learning. To the best of our knowledge, this is the first attempt devoted to community detection from the perspective of learning. It offers a solid foundation for understanding the intuition behind community detection, and can be used as a guideline for designing and utilizing different methods for community detection.
- We provide a thorough theoretical analysis of learning-based community detection methods, discuss their similarities and differences, identify critical challenges that remain poorly addressed and point out five directions for future development.
- We gather abundant resources on learning-based community detection, including state-of-the-arts benchmark datasets and applications.

The rest of this survey paper is organized as follows. Section 2 gives the preliminaries and categorization of existing community detection approaches. Section 3 presents a technical overview of research progress in statistical modeling for community detection. Section 4 overviews the research on deep learning-based community detection. Section 5 discusses applications of community detection. We suggest promising future research directions in Section 6 and conclude in Section 7.

2 PRELIMINARIES AND CATEGORIZATION

We first introduce the terms and notations, and then present a classification of the methods for community detection that we will discuss in this paper.

2.1 Definitions, Terms and Notations

Definition 1. Network. A network $G = (V, E, X)$ consists of n nodes $V = \{v_1, v_2, \dots, v_n\}$, m edges $E = \{e_{ij}\} \subseteq V \times V$, and a maximal number q of attributes x_i on a node v_i , where all x_i 's collectively give rise to an $n \times q$ attribute matrix $X = (x_i)_{n \times q}$. The topological structure of G can be defined by an $n \times n$ adjacency matrix $A = (a_{ij})_{n \times n}$, where $a_{ij} = 1$ if $e_{ij} \in E$, or 0, otherwise. G is undirected if $a_{ij} = a_{ji}$, or directed, otherwise [42].

Definition 2. Community. The network G contains k communities $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$, where \mathcal{C}_i is a subgraph of G and the nodes within \mathcal{C}_i are densely connected whereas the nodes across \mathcal{C}_i and \mathcal{C}_j are sparsely connected. The communities are non-overlapping when $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \forall i, j$.

Definition 3. Community Detection. Given a network G , community detection is to design a mapping \mathcal{F} to assign every node v_i of G into at least one of the k communities, i.e., to label v_i at least one community identity $c_i \in \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$. Equivalently, the problem is to derive a community assignment of nodes $C = (c_1, c_2, \dots, c_n)$.

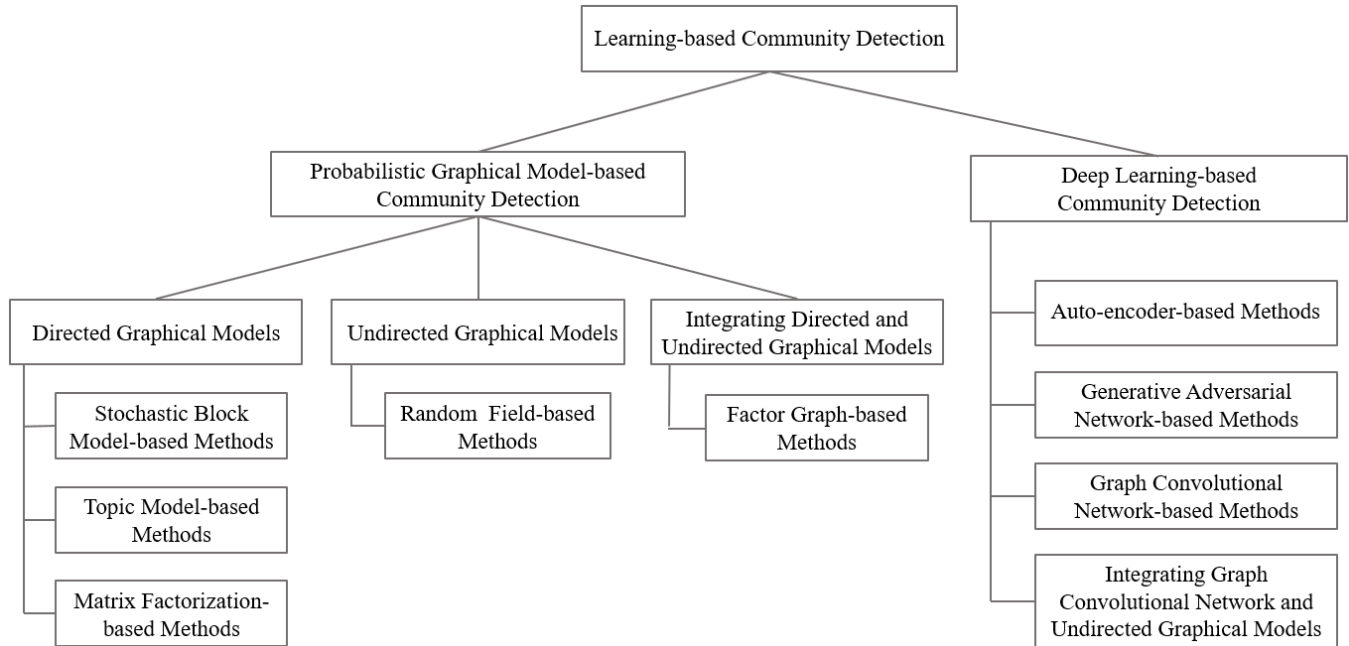


Fig. 2: Classification breakdown of methods for community detection.

2.2 A Taxonomy of Community Detection Approaches

To help better understand the existing learning-based methods and facilitate our discussion in the rest of the paper, we introduce a taxonomy of the methods for community detection (Fig. 2) where the methods are grouped into two categories, probabilistic graphical model and deep learning.

Probabilistic graphical model-based methods employ heuristics or meta-heuristics on community generation to discover network communities. These methods typically adopt some models of network structures to describe the dependencies among the entities (i.e., nodes) via the edges of the networks. Depending the type of probabilistic graphical models used, community detection can be further divided into three main categories: directed graphical models, undirected graphical models and hybrid graphical models. Specifically, directed graphical models are mainly based on hidden variables (i.e., variables not observed in the sample), leveraging similarity of nodes or block structures, to generate the observed edges in a network; undirected graphical models are usually based on field structures, using the constraints of unary and pairwise potentials (e.g., the community label agreement between nearby nodes) to discover community; hybrid graphical models normally transform these two types of models into a unified factor graph to take advantages of both models for community detection.

Deep learning-based methods aim to identify community structures utilizing a new type of community-oriented network representation. It derives the new network representation through some learning strategies that map network data from the original input space to a low-dimensional feature space, with the advantages of low computational complexity and high capability for parallelization. Depending on the learning strategies used, deep learning-based methods fall into four main categories: auto-encoder-based, generative adversarial network-based, graph convolutional network (GCN)-based, and integrat-

ing graph convolutional networks and undirected graphical models. Concretely, auto-encoder-based methods exploit unsupervised auto-encoder, which encodes a network into a low-dimensional representation in the latent space and reconstructs the network, along with its community structures, from the low-dimensional representation. Generative adversarial network-based methods adopt the idea of adversarial learning. They detect communities via an adversarial game between a generator and a discriminator. Graph convolutional network-based methods extract communities by the propagation and aggregation of features on network topology. Hybrid graphical model-based methods integrate graphical convolutional networks and undirected graphical models by, for example, converting a Markov random field (MRF) layer into GCN, to take advantages of both models.

3 COMMUNITY DETECTION WITH PROBABILISTIC GRAPHICAL MODEL

Probabilistic graphical models-based methods ordinarily detect communities through network modeling, i.e., employing graphical models to explain the generation process of networks. In this section, we will focus on three general methods: directed graphical models, undirected graphical models, and hybrid graphical models integrating directed and undirected graphs.

3.1 Directed Graphical Models

We will review the recent development of directed graphical models for community detection, including stochastic block model, topic model and matrix factorization. These methods have solid theoretical basis and reasonably good performance, and have been broadly applied.

3.1.1 Stochastic Block Model-based Methods

Stochastic block model (SBM), an effective generative model of network block structures, adopts statistical modeling for

community detection for the first time [28]. The method probabilistically assigns nodes in a network to different communities (block structures) using a node membership likelihood function, and then progressively infers the probabilities of node memberships by inferencing on the likelihood function to derive hidden communities in the network. Note that there are several SBM variants for community detection, but their core generation process is the same. The basic generation process can be divided into two steps: the first is to iteratively assign a community to each node in the network, and the second is to compute or update the probability of two nodes connected by an edge.

Taking a social network as an example, SBM can be used to capture a probabilistic generation process with the community distribution as a hidden variable. The communities can be reconstructed by maximizing a likelihood function of the node community membership. In this social network, the nodes are partitioned into k disjoint communities with probability $\omega = \{\omega_1, \dots, \omega_k\}$. Assuming there are two nodes v_i and v_j belonging to two communities \mathcal{C}_r and \mathcal{C}_s , represented by c_{ir} and c_{js} . The probability that nodes v_i and v_j connected by an edge, i.e., a_{ij} (0 or 1), obeys a Bernoulli distribution with parameter π_{rs} . The use of a link probability between nodes in two communities makes the model flexible with various types of network structures [44]. The network generation distribution can be defined as:

$$P(C|\omega) = \prod_{i=1}^n \text{Multinomial}(c_i; 1, \omega) = \prod_{i=1}^n \prod_{r=1}^k \omega_r^{c_{ir}}, \quad (1)$$

$$\begin{aligned} P(A|C, \pi) &= \prod_{i<j} P(a_{ij}|c_i, c_j) = \prod_{i<j} \prod_{r,s} \text{Bernoulli}(a_{ij}|\pi_{rs})^{c_{ir}c_{js}} \\ &= \prod_{i<j} \prod_{r,s} (\pi_{rs}^{a_{ij}} (1 - \pi_{rs})^{(1-a_{ij})})^{c_{ir}c_{js}}, \end{aligned} \quad (2)$$

where $C = (c_1, c_2, \dots, c_n)$ denotes the community assignment of nodes, and $\pi = (\pi_{rs})_{k \times k}$ represents the community connection probability matrix. Then, the likelihood function can be described as:

$$\begin{aligned} P(A, C|\omega, \pi) &= P(A|C, \pi)P(C|\omega) \\ &= \prod_{i=1}^n \prod_{r=1}^k \omega_r^{c_{ir}} \times \prod_{i<j} \prod_{r,s} (\pi_{rs}^{a_{ij}} (1 - \pi_{rs})^{(1-a_{ij})})^{c_{ir}c_{js}}. \end{aligned} \quad (3)$$

Based on the likelihood function, Gibbs sampling and the expectation-maximization (EM) algorithm can be used to obtain the model parameters, e.g., ω and π . Finally, we can derive the community partition with the interaction model between communities. The time complexity of the basic SBM is $O(n^2k^2)$, and the process of community detection based on SBM with a Bernoulli distribution is shown in Appendix B [28]. The generation process of the basic SBM may also employ a Poisson distribution [45].

Later, Zhang *et al.* [46] study the problem of hidden class inference in basic SBM. They present a comparative study between three inference approaches, i.e., heuristic spectral methods, mean field approximation and belief propagation, on synthetic networks.

Mixed membership SBMs. Since the basic SBM is only suitable under the assumption that a node belongs to only one community, Airoldi *et al.* [29] propose a mixed membership stochastic blocks model (MMSB) that introduces mixed membership to the stochastic model so that one node may belong to multiple communities. MMSB allows communities to overlap on a directed network, where $a_{i \rightarrow j}$ indicates whether there is a link (arrow) from nodes v_i to v_j . For each node v_i , c_i obeys a Multinomial distribution. If $v_i \in \mathcal{C}_r$ and $v_j \in \mathcal{C}_s$, the community connection probability π_{rs} follows a Beta distribution and $a_{i \rightarrow j} \sim \text{Multinomial}(\omega_i)$, $a_{i \leftarrow j} \sim \text{Multinomial}(\omega_j)$, where ω is the mixed membership parameter of nodes. The links between communities are represented by a Bernoulli distribution. The joint distribution of MMSB can be formulated as:

$$\begin{aligned} P(A, \pi, \omega, a_{i \rightarrow j}, a_{i \leftarrow j} | \alpha, \beta) &= \prod_i P(\omega_i | \alpha) \times \\ &P(\pi | \beta) \prod_{i,j} P(a_{i \rightarrow j} | \omega_i) P(a_{i \leftarrow j} | \omega_j) P(a_{ij} | a_{i \rightarrow j}, a_{i \leftarrow j}, \pi). \end{aligned} \quad (4)$$

The time complexity of MMSB is $O(n^2k)$, and the process of community detection based on MMSB is described in Appendix B [29], which assumes that the parameters are estimated by inference methods such as EM.

The original MMSB is not good at handling diverse types of the information of nodes in community, e.g., the nodes may represent people who are connected one another based on different social relationships. To address this problem, Fan *et al.* [47] propose a novel MMSB-based method, named Copula mixed membership stochastic block model (cMMSB) to introduce a Copula function into MMSB to model dependencies among nodes. Moreover, Miller *et al.* [48] improve the inference of MMSB by introducing a Dirichlet process mixture to a mixture of finite mixtures (MFMs). Pal *et al.* [49] propose a mixed membership degree-corrected SBM and develop an inference method of the posterior distribution with Markov chain Monte Carlo (MCMC), to boost the embedding performance of MMSB. The degree-corrected SBM is widely used, which we will discuss next.

Degree-corrected SBMs. Newman *et al.* [30] reason that the basic SBM divides nodes according to their degrees that are usually nonuniform distributed. To accommodate the possible broad degree distributions, they propose degree-corrected SBM (DCSBM), which introduces a degree parameter to every node to scale the edge probabilities and make expected degrees match observed degrees. The probability function of network G is defined as follows:

$$\begin{aligned} P(G|\pi, c) &= \prod_{i<j} \frac{(\pi_{c_i c_j})^{a_{ij}}}{a_{ij}!} \exp(-\pi_{c_i c_j}) \\ &\times \prod_i \frac{(\frac{1}{2}\pi_{c_i c_i})^{a_{ii}/2}}{(a_{ii}/2)!} \exp(-\frac{1}{2}\pi_{c_i c_i}), \end{aligned} \quad (5)$$

where $\pi_{c_i c_j}$ is the expected value of the adjacency matrix element a_{ij} .

There are also several variants of DCSBM model. Guilikers *et al.* [51] propose an improved DCSBM which is feasible for sparse networks. Some other distinct lines of effort are to improve and extend DCSBM by model in-

TABLE 1: Summary of SBM-based community detection, where "AD k " describes whether the approach can automatically determine the number of communities, i.e., Yes or No.

Categories	Approaches	Sketches	Overlapping	AD k	
Basic	SBM (1983) [28]	Propose a stochastic model for social network.	No	No	
	MMSB (2008) [29]	Extend blockmodels for relational data to ones that capture mixed membership latent relational structure.	Yes	No	
MMSB	cMMSB (2016) [47]	Combine an individual Copula function with MMSB with improved in capturing group interactions.	Yes	No	
	MMDCB (2019) [49]	Propose a mixed membership degree-correct SBM and develop an inference method of the posterior distribution with Markov chain Monte Carlo (MCMC).	Yes	No	
DCSBM	DCSBM (2012) [50]	Introduce expected values to basic SBM to adapt multi-edges and self-edges contained in social networks.	Yes	No	
	sparseDCSBM (2017) [51]	Propose a spectral clustering algorithm with normalized adjacency matrix based on DCSBM.	No	Yes	
	CMM (2018) [52]	Establish a convexified modularity maximization approach for estimating the hidden communities based on DCSBM.	No	No	
	NCV (2018) [53]	Provide a network cross-validation approach to determine the hidden communities based on DCSBM.	No	No	
DynSBM	dMMSB (2009) [54]	Propose a state space MMSB which can track across time dynamic evolution.	Yes	No	
	DynamicSBM (2010) [55]	Propose a novel Bayesian approach for network tomographic inference buliding on MMSB and apply in dynamic network.	No	No	
	DSBM (2011) [56]	Capture the evolution of community by modeling the transition of community memberships for individual nodes.	No	No	
	DBTDP (2014) [57]	Propose a dynamic stochastic block model with temporal Dirichlet process for hidden community.	Yes	No	
	SBTM (2015) [58]	Provide a local search algorithm for the inference procedure of time evolution.	No	No	
	dDCSBM (2016) [59]	Propose a dynamic DCSBM to model and monitor dynamic networks that undergo a significant structural change.	No	Yes	
	DPSBM (2019) [60]	Establish a fully Bayesian generation model with the heterogeneity of node degree.	No	Yes	
	SNR-DSBM/ER (2020) [61]	Focus on estimating the location of a single change point in a dynamic stochastic block model and take a least squares criterion function for evaluating each point in time.	No	Yes	
	OSBM	OSBM (2011) [62]	Provide a global and local variational technique for discovering community.	Yes	No
		K-LAFTER (2018) [63]	Present a small-variance asymptotics based SBM for overlapping community detection.	Yes	Yes
MNPAOCD (2020) [64]		Optimize the inference process and expect parameters in proceeding.	Yes	Yes	
LSBM	LMBP (2015) [18]	Combine heterogeneous distribution with SBM to link community detection.	Yes	Yes	
GNNsBM	DGLRFM (2019) [65]	Design a GNN-based overlapping SBM framework and can be adapted readily for other types of SBMs.	Yes	Yes	

ference. Chen *et al.* [52] present a convexified modularity maximization approach for estimating hidden communities under DCSBM. Chen *et al.* [53] provide a network cross-validation (NCV) approach based on a block-wise node-pair splitting to determine the hidden communities for DCSBM.

Dynamic SBMs. Different from the above methods, analyzing dynamic networks based on SBM is also a relatively active field. Yang *et al.* [56] suggest a dynamic stochastic block model named DSBM, which progressively updates the probabilistic model to find community in large dynamic sparse network. Specifically, DSBM uses the distribution of model parameter instead of the most likely values for the model parameters in prediction, and provides an offline inference and an online inference to estimate the parameters. DSBM assumes that nodes in a dynamic network remain unchanged. Letting $C_T = \{C^{(1)}, C^{(2)}, \dots, C^{(T)}\}$ be the collection of community assignments of all nodes over T discrete time steps, and the likelihood function of the model

is as follows:

$$P(W^{(T)}, C^{(T)} | \omega, \pi, A) = \prod_{t=1}^T P(W^{(t)} | C^{(t)}, \pi) \prod_{t=2}^T P(C^{(t)} | C^{(t-1)}, A) P(C^{(1)} | \omega), \quad (6)$$

where $W^{(t)}$ and $C^{(t)}$ denote the snapshot of a network and the community assignments of nodes at a given time step t . The time complexity of DSBM is $O(nT + mT + k^2T)$, and its generation process is illustrated in Appendix B [56],

Following DSBM, Tang *et al.* [57] introduce the Dirichlet process to SBM, which can find the optimal number of communities of evolution, and in turn, alleviate the problem of fixed community number of dynamic social networks. Xu *et al.* [66] propose a new approach named stochastic block transition model (SBTM) that includes two hidden Markov assumptions for dynamic network. Wu *et al.* [67] propose a full Bayesian generation model, which incorporates the

heterogeneity of the degrees of nodes to model dynamic complex networks. Bhattacharjee *et al.* [61] optimize SBM with change-point estimation in dynamic social networks. Inspired by the success of MMSB, some dynamic SBMs based on MMSB have been proposed. Xing *et al.* [55] propose a variant MMSB model for dynamic networks. Fu *et al.* [54] design a state space mixed membership stochastic block model with crossing time. Besides, there also exists DCSBM-based approaches. Wilson *et al.* [59] suggest a dynamic version of DCSBM to model and monitor dynamic networks that undergo a significant structural change.

Others. In addition to the above methods, there are several other extensions to the basic SBM where communities can overlap, as summarized in Table 2. For instance, OSBM represents the SBM that are designed to find overlapping communities and LSBM represents the SBM that are extended to find link community. Specifically, Latouche *et al.* introduce OSBM with global and local variational technology. Jin *et al.* [64] provide a stochastic model to accommodate the relative importance and the expected degree of every node in each community, and improve the inference technique that it uses.

Link communities are often more informative and intuitive than node communities, because links usually have unique identities, whereas nodes may have multiple roles. For instance, in a social network, most individuals belong to multiple communities such as families or friends, while the link between two individuals often exists for a dominant reason, which may represent family ties or friendship. Furthermore, multiple links connecting to a node may belong to distinct link communities, so that the node can be assigned to multiple communities of links. He *et al.* [18] combine heterogeneous distributions (e.g., power law distribution) of community sizes with SBM for link community detection. They suggest a stochastic model for link community and extend the model by introducing a scheme of interactive bipartition. Besides the above models, Mehta *et al.* [65] introduce graph neural network into SBM, which integrate deep learning and SBM for the first time.

3.1.2 Topic Model-based Methods

Topic model, such as Latent Dirichlet Allocation (LDA) [68], is a statistical model capable of modeling hidden topics behind texts in natural language processing. LDA models topics by employing latent variables, which have attracted significant interests and have been widely used in detecting communities. Topic models can be grouped into two categories: one models network structures as documents and the other models attributes of network, such as user interests, to detect communities.

Modeling network structures as documents. We take LDA as an example to describe the principle of the methods in the first category. To be specific, a method in this group first assumes that each node in a network may belong to multiple communities, and thus the communities are regarded as "topics" while the nodes are taken as "documents". It then selects several initial communities, and iteratively updates the communities according to the topology of the network to obtain resulting communities. Among the existing methods, a representative model is SSN-LDA [69], which is a LDA-based hierarchical Bayesian algorithm on

link networks where communities are modeled as latent variables. Nodes in such a social network are regarded as social actors and edges as social interaction. Social interaction profile (SIP) of each social actor, consisted of a set of neighbors and weights, is used to characterize the actor. Specifically, in SSN-LDA, a social network is viewed as a corpus, where social interaction profiles are regarded as documents and the occurrence of social interaction is deemed as words. The nodes are modeled as a corpus by SSN-LDA, which mines communities on transformed corpus, and this problem is equivalent to topic detection on corpus utilizing LDA. The generation process of SSN-LDA for one social interaction profile (SIP_i) is clarified in Appendix C [69], and the joint distribution is written as:

$$P(a_i, c_i, \vec{\theta}_i, \vec{\phi} | \vec{\alpha}, \vec{\beta}) = \prod_{j=1}^{N_i} P(a_{ij} | \vec{\phi}_{c_i}) P(c_i | \vec{\theta}_i) P(\vec{\theta}_i | \vec{\alpha}) P(\vec{\phi} | \vec{\beta}), \quad (7)$$

where $\vec{\phi}$ is the mixture component of community c_i , N_i is the number of social interactions in a social interaction profile (SIP_i), $\vec{\theta}_i$ is the community mixture proportion for SIP_i , and $\vec{\alpha}$ and $\vec{\beta}$ are the Dirichlet prior distribution hyperparameters that are known. The time complexity of SSN-LDA is $O(mk)$.

Using social network attributes. Numerous topic models utilize attributes of social network, e.g., user interests, to discover community. Yin *et al.* [70] propose to integrate community detection and topic model, which gives rise to latent community topic analysis (LCTA). Their method divides the sampling process into user node and link samplings. The process is to sample all network connections after sampling a user node, and exploit the sampling results of these two stages as the sampling result of user node. LCTA assigns community membership attributes to each user node and link. After the sampling process, user nodes can be assigned to communities based on community membership. The advantage is that the two-stage sampling process forms a sampling area with user nodes as the core, which can simulate the semantic influence of user nodes on surrounding links. The disadvantage is that LCTA does not consider the link relationship of social network when assuming the degree of community membership, which may disconnect individual communities. Further, Cha *et al.* [71] design a tree relationship model according to the topic information of followers in social network, use hierarchical LDA to model the text information in tree relationship model, and propose HLDA for semantic social network analysis.

A method of **combining topic model with Bayesian model** is proposed recently by Xu *et al.* [72]. They define a joint probability distribution for all possible attributed networks. For a given attributed social network to be clustered, the model assigns a probability to each possible clustering of nodes. Therefore, the clustering problem can be transferred to the problem of finding the clusters that have the highest probability. The algorithm for clustering attribute communities is shown in Appendix C [72]. The Bayesian probabilistic

model for clustering attributed networks is as follows:

$$\begin{aligned} & P(\alpha, \theta, \phi, A, X, C | \varepsilon, \lambda, \mu, \nu) \\ & = P(\alpha | \varepsilon) P(\theta | \lambda) P(\phi | \mu, \nu) P(C | \alpha) P(A | C, \phi) P(X | C, \theta), \end{aligned} \quad (8)$$

where α denotes the probability of the nodes belonging to different communities, θ represents the attribute probability distribution of nodes, ϕ denotes the edge occurrence probability between communities, ε and λ are the Dirichlet prior distribution hyper-parameters, and μ and ν are the Beta prior distribution hyper-parameters.

Later, He *et al.* [73] introduce a generative model for simultaneously identifying communities and deriving their semantic description. They combine a nested EM algorithm with belief propagation, and explore the hidden correlation between the two parts to improve resulting communities and description. Jin *et al.* [74] observe that the attributes usually embody a hierarchical semantic structure. To handle this, they propose a novel Bayesian model named BTLSC, which distinguishes words from background and general from specialized topics.

Unlike traditional topic models that assume that the topics of social network are independent, **topic embedding methods** focus on describing correlations between topics by embedding words and topics into topic models. He *et al.* [75] present a topic embedding model that combines distributed representation learning with topic correlation modeling. Jin *et al.* [76] develop a novel topic embedding model named community-enhanced topic embedding (CeTe), which combines topic documents and network structures to detect communities. CeTe consists of three components: a document component for describing topics, a topological component for representing network communities, and a probabilistic transition mechanism connecting the first two parts. Specifically, CeTe uses a DCSBM to describe the sub-component of network community, where communities obey a Dirichlet distribution, and topics obey a Uniform distribution. For each document, the community assignment is drawn from a Multinomial distribution, whereas the link between two documents obeys a Bernoulli distribution. For each word, CeTe draws topic distribution following a Multinomial distribution.

3.1.3 Matrix Factorization-based Methods

Non-negative matrix factorization (NMF) [77] is another directed graphical model for community detection. Specifically, the NMF-based methods assume there are k communities in a network, and deem the adjacency matrix $A = (a_{ij})_{n \times n} \in \mathbb{R}_+$ as a non-negative matrix to be decomposed, where a_{ij} denotes the likelihood if there is a connection between nodes v_i and v_j . We define $\mathcal{W} = (w_{ir})_{n \times k} \in \mathbb{R}_+$ and $\mathcal{H} = (h_{jr})_{n \times k} \in \mathbb{R}_+$, whose elements w_{ir} and h_{jr} represent the likelihoods that v_i generates an out-edge, i.e., an edge starting from v_i , and v_j generates an in-edge, i.e., an edge ending at v_j , belonging to the r -th community. Then, the likelihood that nodes v_i and v_j are connected can be described as:

$$\tilde{a}_{ij} = \sum_{r=1}^k w_{ir} h_{jr}^T. \quad (9)$$

As a result, the community detection problem can be represented as $\tilde{A} = \mathcal{W}\mathcal{H}^T$. In general, there are two classic loss functions to evaluate the performance of NMF. One is the square of the Frobenius norm of the difference between A and \tilde{A} [78], and the other is the KL-divergence that measures their difference.

Furthermore, for a undirected network with A being symmetric, the non-negative factorization matrices \mathcal{W} and \mathcal{H} should be equal. In this paper, we use B to represent these matrices, and the loss function can be written as:

$$J = \min_{B \geq 0} \|A - BB^T\|_F^2. \quad (10)$$

NMF is initially used to identify non-overlapped community. Since it is easily extendable, NMF has been adopted to solve other types of community detection problems, such as overlapping, attributed, dynamic and semi-supervised, as summarized in Table 3.

Basic NMF. Kuang *et al.* [79] propose a general approach, which inherits the advantages of NMF by enforcing non-negativity on the clustering assignment matrix, for graph clustering. Shi *et al.* [80] present a novel pairwise constrained non-negative symmetric matrix factorization (PCSNMF) method, which imposes pairwise constraints generated from ground-truth community information, to improve the performance of community detection. Sun *et al.* [81] design a non-negative symmetric encoder-decoder approach to derive a better latent representation to improve community detection. Unlike other NMF-based methods that merely pay attention to the loss of the decoder, they combine the loss of the decoder and encoder to construct a unified loss function, so that the community membership of each node obtained is clearer and more explanatory.

Overlapping NMF. Overlapping community detection is another active research topic due to the overlapping and nesting properties of real-world networks. Wang *et al.* [21] develop a NMF framework to identify non-overlapping and overlapping community structure, and give a symmetric NMF formula for undirected networks. Moreover, they clarify the methods on asymmetric NMF and joint NMF, where the former is capable of identifying community structures in directed networks, while the latter is more suitable for compound networks (e.g., an automatic movie recommendation system which contains three networks: user network, movie network and user-movie network). Yang *et al.* [82] present a cluster affiliation model BIGCLAM to detect densely overlapping, hierarchically nested, and non-overlapping communities in a massive network. Specifically, BIGCLAM first builds on communities based on community affiliation of nodes, i.e., each node has an affiliation strength to each community via assigning node-community pair a non-negative latent factor, and then combines the NMF with block stochastic gradient descent, so as to estimate the non-negative latent factors to detect communities in large networks. The loss function of the model is defined as:

$$J = \sum_{e_{ij} \in E} \log(1 - \exp(-b_i b_j^T)) - \sum_{e_{ij} \notin E} b_i b_j^T. \quad (11)$$

The time complexity of BIGCLAM is $O(n)$.

Attribute NMF. Recently, it has attracted a substantial amount of interest to the semantic information of com-

TABLE 2: Summary of NMF-based community detection, where "AD k " describes whether the approach can automatically determine the number of communities, i.e., Yes or No.

Categories	Approaches	Sketches	Objective Functions	Overlapping	AD k
Basic	SymNMF (2012) [79]	Develop a symmetric NMF framework based on Newton-like for graph clustering.	$\ A - BB^T\ _F^2$	No	No
	PCSNMF (2015) [80]	Present a symmetric NMF method with pairwise constraints generated from the ground-truth community information.	$\ A - BB^T\ _F^2 + \alpha[\text{Tr}(B^T \mathcal{M} B \mathcal{Q}) + \text{Tr}(B^T \mathcal{P} B)]$	No	No
	NSED (2017) [81]	Pose a non-negative symmetric encoder and decoder approach to obtain a better network representation.	$\ A - \mathcal{W} \mathcal{H}\ _F^2 + \ \mathcal{H} - \mathcal{W}^T A\ _F^2$	No	No
Overlapping	SNMF, ANMF, JNMF (2011) [21]	Apply NMF to community detection firstly.	$\ A - BB^T\ _F^2$	Yes	No
	BIGCLAM (2013) [82]	Present a cluster affiliation model for overlapping, hierarchically nested community detection in large scale networks.	$\sum_{e_{ij} \in E} \log(1 - \exp(-b_i b_j^T)) - \sum_{e_{ij} \notin E} b_i b_j^T$	Yes	Yes
Attribute	NMTF (2015) [83]	Develop a NMF clustering framework combining nodes' relations and users' contents to detect community structure.	$\ M_{u-u} - \mathcal{U} H_1 \mathcal{U}^T\ _F^2 + \ M_{t-f} - \mathcal{V} H_2 \mathcal{N}^T\ _F^2 + \ M_{u-f} - \mathcal{U} H_3 \mathcal{N}^T\ _F^2$	No	No
	SCI (2016) [84]	Propose a semantic community identification method, which can annotate semantic as well as detect community.	$\ B - XS\ _F^2 + \alpha \sum_{r=1}^k \ S(:, r)\ _1^2 + \beta \ A - BB^T\ _F^2$	No	No
Dynamic	DBNMF (2016) [85]	Present a Bayesian probabilistic model based on NMF to identify overlapping communities in temporal networks.	$-\log P(V_t B_t) - \log P(B_t B_{t-1}, \alpha) - \log P(B_t \beta_t) - \log P(\beta_t)$	Yes	Yes
	sE-NMF (2017) [86]	Develop a semi-supervised evolutionary NMF framework for dynamic community detection via prior information.	$\ A_t - \tilde{B}_t \tilde{B}_t'\ _F^2$	No	No
Semi-supervised	USSF (2015) [87]	Present a unified semi-supervised community detection algorithm based on combination of prior and topology information aimed NMF.	$\mathcal{L}_\alpha(A, B) + \lambda \mathcal{R}_\beta(O, B)$	No	No

munity structure utilizing NMF, i.e., delineating the corresponding community semantic information while identifying community structure [83], [84], [88]. In particular, Pei *et al.* [83] combine social relations and content of users to detect communities via a non-negative matrix trifactORIZATION (NMTF)-based clustering with three types of graph regularization. However, the above method merely exploits network topology and content information to discover communities, without considering how to utilize the mined contents, i.e., semantic information, to explain the meaning of communities. To address this issue, wang *et al.* [84] propose a semantic community identification called SCI, which integrates the community membership matrix denoting network topology and community attribute matrix representing semantic information. The loss function of the model is defined as:

$$\min_{B \geq 0, S \geq 0} J = \|B - XS\|_F^2 + \alpha \sum_{r=1}^k \|S(:, r)\|_1^2 + \beta \|A - BB^T\|_F^2, \quad (12)$$

where S represents attribute community matrix, α is a trade-off hyper-parameter between the first error and the second sparsity term, and β is a positive parameter for setting the

proportion of the contribution of network topology. The time complexity of SCI is $O((nqk + n^2k))$, where q is the dimension of node attribute.

Dynamic and semi-supervised NMF. It deserves further attention that, during recent years, several investigators have extended NMF in the field of dynamic and semi-supervised community detection, and achieved encouraging results. For **dynamic** community detection, Wang *et al.* [85] utilize a Bayesian model based on NMF to identify overlapping communities on temporal networks, and automatically derive the number of communities in each snapshot network based on automatic relevance determination. The loss function is as follows:

$$J_t = -\log P(V_t|B_t) - \log P(B_t|B_{t-1}'', \alpha) - \log P(B_t|\beta_t) - \log P(\beta_t), \quad (13)$$

where V_t is a snapshot of a temporal network, B_t is the non-negative matrix obtained via V_t , and B_{t-1}'' is the new B_{t-1} which has been adjusted according to the node distribution of B_t . $\beta_{(\cdot)}$ is a parameter from a half normal distribution, and α is a parameter to balance the clustering results of the current and previous snapshot networks. Later, Ma *et al.* [86] show that NMF can be applied to dynamic community detection via clarifying the equivalence relationship among

evolutionary spectral clustering, evolutionary NMF and optimization of evolutionary modularity density. Therefore, they employ the above equivalence relationship to develop a semi-supervised evolutionary NMF method, named sE-NMF, to integrate the prior information to detect communities in dynamic temporal networks.

For **semi-supervised** community detection, Yang *et al.* [87] put forward a unified semi-supervised algorithm by combining prior information and topology information aimed at two non-negative matrices generated by NMF. Moreover, with the must-link prior information (i.e., the prior information that a node pair composed of two nodes must belong to the same community [94]), they add a graph regularization item as a penalty function to the loss function to minimize the difference between nodes in the same community, thereby improving the performance of community detection. The loss function is defined as:

$$J(B|A, O) = \mathcal{L}_\alpha(A, B) + \lambda \mathcal{R}_\beta(O, B), \quad (14)$$

where O denotes the matrix of prior information. $\mathcal{L}_\alpha(A, B)$ is the loss function of NMF, where $\alpha \in \{LSE, KL, SYM, MOD, ADJ, LAP, NLAP\}$ is the parameter to measure similarity. $\lambda \mathcal{R}_\beta(O, B)$ is a graph regulation term, where λ is the tradeoff parameter between the loss function and graph regulation term, and $\beta \in \{LSE, KL\}$ is the specific graph regulation term.

3.1.4 Summary for Direct Graphical Models

The direct graphical models typically transform the community detection problem into a Bayesian inference problem based on the observed network data and then optimize model parameters utilizing a maximum likelihood function or a maximum of a posteriori to obtain hidden variables of the model to discover network community structures. However, these methods often ignore the diversity of community patterns (e.g., community structure with homophily or heterophily) in the real world, and the network topology used is often noisy and sparse, limiting the performance of community detection.

3.2 Undirected Graphical Models

To the best of our knowledge, the existing studies of undirected graphical models for community detection mainly exploit Markov random field (MRF) [95]. MRF, a kind of random field, has enjoyed much success covering a variety of applications, such as computer vision and image processing. Particularly, we are interested in its applications to community detection. The MRF-based methods can be grouped into three categories (as summarized in Table 4): the first is the modeling based on MRF that detects community relation based on network topology, the second exploits the information of semantic attributes, and the third combines MRF with graph neural network (GNN).

Topology MRF. He *et al.* [31] first apply MRF to network analysis where data are organized on networks with irregular structures, and propose a network-specific MRF approach, namely NetMRF, for community detection. This method effectively encodes the structural properties of an irregular network in an energy function so that the minimization of the energy function gives rise to the best community

structures. The energy function can be represented as the sum of pairwise potential functions, written as follows:

$$\begin{aligned} E(C; A) &= \sum_{v_i \neq v_j} \Theta_{ij}(c_i, c_j; a_{ij}) \\ &= \sum_{v_i \neq v_j} [-(-1)^{\delta(c_i, c_j)} \left(\frac{d_i d_j}{2m} - a_{ij} \right)], \end{aligned} \quad (15)$$

where δ is the probability of nodes v_i and v_j falling into the same community partition, d_i is the degree of node v_i , and m is the numbers of edges. According to [95], the smaller the function, the better the community partition. Further, Jin *et al.* [90] formalize the modularity function as a statistical model and propose a novel MRF method for community detection. This method redefines the energy function via the approach of modularity representation, and leverages the max-sum belief propagation (BP) to infer model parameters to improve the performance. The energy function is represented as follows:

$$E(C; A) = \sum_{v_i, v_j \in V} -\left(a_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j). \quad (16)$$

The time complexity is $O(m)$.

Moreover, to overcome the issue of losing vital structural information between nodes after network embedding, Jin *et al.* [89] propose a general MRF method to incorporate coupling relationship between pairs of nodes in network embedding to better detect community. In this method, the energy function is composed of two components: a set of unary potentials that make the network embedding to play a dominant role and a set of pairwise potentials that utilize constraints on node pairs to fine-tune unary potentials. Formally, the complete energy function can be defined as:

$$E(C; A) = \sum_{v_i} \Theta_i(c_i) + \sum_{c_{ij} \in E} \Theta_{ij}(c_i, c_j), \quad (17)$$

where Θ_i and Θ_{ij} are the unary potential function and pairwise potential function respectively.

Topology & attribute MRF. The combination of MRF and node semantic models (e.g., a topic model) have been a recent research focus. However, methods that directly integrate MRF with node semantic models cannot in general achieve satisfactory results. It is mainly because the parameters of the two models cannot be adjusted to support each other, making it difficult to combine the advantages of the two methods. He *et al.* [91] propose a new model, named attrMRF, to integrate LDA [68] and MRF to form an end-to-end learning system to train the parameters jointly. Concretely, attrMRF first transforms LDA and MRF into a unified factor graph, realizing the effective integration of directed graphic model (i.e., LDA) and undirected graphic model (i.e., MRF). Then it adopts a backpropagation (BP) algorithm to train the parameters simultaneously, resulting in an end-to-end learning of the two models. The global energy function of this model is represented as:

$$\begin{aligned} E(Z, C; A, X, \alpha, \beta) &= \sum_{v_i \neq v_j} \Theta_{ij}(c_i, c_j; a_{ij}) \\ &\quad - \sum_{r=1}^n \frac{1}{\beta_1} \ln f_{\theta_r} - \sum_{p=1}^q \frac{1}{\beta_1} \ln f_{\phi_p}, \end{aligned} \quad (18)$$

TABLE 3: Summary of MRF-based community detection.

Categories	Approaches	Sketches	Object Functions
Topology	NetMRF (2018) [31]	Apply MRF to community detection firstly.	$\sum_{v_i \neq v_j} [-(-1)^{\delta(c_i, c_j)} (\frac{d_i d_j}{2m} - a_{ij})]$
	GMRF (2019) [89]	Optimize network embedding and develop a general MRF framework by incorporating network embedding into MRF to better detect community structure.	$\sum_{v_i} \Theta_i(c_i) + \sum_{e_{ij} \in E} \Theta_{ij}(c_i, c_j)$
	ModMRF (2020) [90]	Propose a MRF method formalizing modularity as the energy function for community detection.	$\sum_{v_i, v_j \in V} -(a_{ij} - \frac{d_i d_j}{2m}) \delta(c_i, c_j)$
Topology & attribute	attrMRF (2019) [91]	Present a model integrating LDA into MRF to form an end-to-end learning system for community detection.	$\sum_{v_i \neq v_j} \Theta_{ij}(c_i, c_j; a_{ij})$ $-\sum_{r=1}^n \frac{1}{\beta_1} \ln f_{\theta_r} - \sum_{p=1}^q \frac{1}{\beta_1} \ln f_{\phi_p}$
Combining GNN	MRFaGCN (2019) [92]	Design a new approach based on the combination of GCN and MRF for semi-supervised community detection.	$-\sum_{i=1}^{n'} \sum_{j=1}^k Y_{ij} \ln Z_{ij}$
	GMNN (2019) [93]	Propose a new approach combining the advantages of both statistical relational learning and graph neural network for semi-supervised node classification.	$E_{q_\theta}(y_U x_V) [\log p_\phi(y_L, y_U x_V) - \log q_\theta(y_U x_V)]$

where $\sum_{v_i \neq v_j} \Theta_{ij}(c_i, c_j; a_{ij})$ denotes the global energy potential of MRF as defined in (15), β_1 is a temperature coefficient, f_{θ_r} and f_{ϕ_p} are the intermediate results generated by LDA joint probability distribution. Besides attrMRF, there are also several approaches incorporating probabilistic graphical models into deep learning, such as MRFaGCN [92] and GMNN [93], which will be covered in detail later.

The undirected graphical models mainly use the characteristics of MRF, i.e., unary and pairwise potentials for irregular network structure and node attribute, to identify community structures. Nevertheless, different energy functions need to be fine-tuned.

3.3 Integrating Directed and Undirected Models

Directed and undirected graphical models have also been integrated to detect communities in complex networks. This type of integration is typically implemented via a factor graph model. A factor graph [96] is a tuple $(V, \mathcal{F}, \varepsilon)$ consisting a set V of variable nodes, a set \mathcal{F} of factor nodes, and a set $\varepsilon \subseteq V \times \mathcal{F}$ of edges each of which connects a variable node and a factor node. Taking MRF as an example, the joint probability distribution of a factor graph is described as:

$$p(y) = \frac{1}{\mathbb{Z}} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)}), \quad (19)$$

where $\mathbb{Z} = \sum_{y \in Y} \prod_{F \in \mathcal{F}} \psi_F(y_{N(F)})$ denotes a normalization factor, and $N(F) = \{v_i \in V : (v_i, F) \in \varepsilon\}$ is the set of variable nodes adjacent to factor node F .

Yang *et al.* [97] first propose an instantiation model based on factor graph, which incorporates three layers, bottom layer (observed nodes), middle layer (hidden vector) and top layer (latent variables for communities). It utilizes node-feature and edge-feature functions to mine dependencies between bottom and top layer nodes to represent corresponding communities, so as to better detect communities. Further, Jia *et al.* [98] apply factor graph model to ego-center network (a kind of representation of human social networks, which is used to represent the network between an individual and others that the ego has a social relationship with [99]), and propose an ego-centered method to

analyze social academic influence on co-author networks. This method model the ego-centered community detection in a unified factor graph, employing a parameter learning algorithm to estimate the topic-level social influence, the social relationship strength between these nodes and community structures to detect ego-community structures.

These methods merely identify the structure of communities and ignore the semantic information of community, which is much critical for understanding the meaning of community structure. He *et al.* [91] employ a factor model to overcome the deficiency that directed graphical model (i.e., LDA) and undirected graphical model (i.e., MRF) are insufficient to integrate due to parameter sharing and joint training and to make the discovered community structure semantically interpretable. The joint probability distribution of MRF and LDA formulated in factor graph is written as:

$$P(Z, C; A, X, \alpha, \beta) = \frac{1}{\mathbb{Z}} \prod_{r=1}^n f_{\theta_r} \prod_{p=1}^q f_{\phi_p} \prod_{v_i \neq v_j} f_{\gamma_{ij}}, \quad (20)$$

where \mathbb{Z} denotes normalization term, f_{θ_r} and f_{ϕ_p} are defined in (18), and $f_{\gamma_{ij}}$ is the pairwise potential of nodes v_i and v_j . Their major contributions lie in that they adopt the fusion technique of MRF and LDA to deal with community detection, which can well overcome the difficulties that two model's parameters are hard to share and train together via factor graphs and belief propagation.

The emergence of factor graph models that integrate directed and undirected graphical models has greatly improved the performance of community detection. However, these probabilistic graph models generally adopt variational inference or Markov chain Monte Carlo (MCMC) sampling for model optimization, which inevitably leads to high computational complexity. Deep learning, with the ability to effectively optimize on high-dimensional network data, has a potential in handling community detection.

4 COMMUNITY DETECTION WITH DEEP LEARNING

In recent years, deep learning has drawn a great deal of attention and has been demonstrated to have great power on

TABLE 4: Summary of auto-encoder-based community detection, where "A" and "X" denote whether the approach utilize network topology and node attributes respectively, and "-" represents no constraint.

Categories	Approaches	A	X	Encoder	Decoder	Focus	Constraints
Stacked	Semi-DNR (2016) [100]	Yes	No	MLP	MLP	Network embedding	Pairwise constraint
	DIR (2017) [101]	Yes	Yes	MLP	MLP	Network embedding	-
	INSNCCD (2018) [102]	Yes	Yes	MLP	MLP	Network embedding	Modularity maximization
	AAGR (2018) [103]	Yes	Yes	MLP	MLP	Network embedding	Adaptive parameter
	CDDTA (2019) [104]	Yes	No	MLP	MLP	Network embedding	Regularization term
	DeCom (2019) [105]	Yes	No	MLP	MLP	Clustering result	Modularity maximization
	NEC (2020) [106]	Yes	Yes	GCN	Inner product	Embedding and clustering	Modularity maximization
Sparse	GraphEncoder (2014) [107]	Yes	No	MLP	MLP	Network embedding	Sparsity constraint
	DFuzzy (2018) [108]	Yes	No	MLP	MLP	Clustering result	Sparsity constraint and modularity maximization
	CDMEC (2020) [109]	Yes	No	MLP	MLP	Clustering result	Sparsity constraint
Denoising	MGAE (2017) [24]	Yes	Yes	GCN	GCN	Network embedding	Interplay exploitation
	GRACE (2017) [110]	Yes	Yes	MLP	MLP	Embedding and clustering	Propagation constraint
Variational	ARVGA (2018) [111]	Yes	Yes	GCN	Inner product	Network embedding	Prior constraint
	VGAECD (2018) [112]	Yes	Yes	GCN	Inner product	Embedding and clustering	-
	DAEGC (2019) [113]	Yes	Yes	GAT	Inner product	Embedding and Clustering	KL divergence constraint
	New VGAECD (2019) [114]	Yes	Yes	GCN	Inner product	Embedding and clustering	-
	Ladder VAE (2019) [115]	Yes	No	GCN	MLP	Embedding and clustering	-
	NetVAE (2019) [116]	Yes	Yes	MLP	MLP	Network embedding	Prior constraint

a wide variety of problems, including community detection. Classic deep learning explores and exploits convolutional neural networks (CNNs) and probability modeling for community detection. For example, Sperli *et al.* [32] design a novel approach, based on CNNs and the topological characteristics of adjacency matrices, for automatic community detection. Sun *et al.* [117] propose a probabilistic generative model, i.e., vGraph, to jointly detect overlapping (and non-overlapping) communities and learn node (and community) representation. vGraph represents each node by a mixture of communities and defines a community as a Multinomial distribution over nodes. Furthermore, Cavallari *et al.* [118] find that there is a closed-loop relationship among community detection, community embedding, and node embedding. Guided by this insight, they present a novel community embedding method, called ComE, to jointly solve the three tasks altogether. He *et al.* [119] design a novel self-translation network embedding (STNE) approach, which maps the content sequence to node identity sequence to improve community detection.

Although these methods have had reasonable performance on discovering communities, they are straightforward applications of deep learning to community detection [120], without considering the characteristics of networks, e.g., irregularity of network topology and complex network structures. In this section, we will discuss the following four types of methods that are designed for complex networks, i.e., auto-encoder-based, generative adversarial network-based, graph convolutional network-based, and methods integrating graph convolutional network and undirected graphical models.

4.1 Auto-encoder-based Methods

Auto-encoders [121] are simple but important neural models that convert high-dimensional (network) data into low-dimensional representations. Concretely, auto-encoders

learn a new representation of data in an unsupervised manner using the encoder and decoder components. They always have multi-hidden layers and a symmetrical architecture, and the output of one layer is the input to its successive layer. The objective of auto-encoders is to minimize the error between original input and reconstructed data to learn an optimal hidden representation, which can be denoted as:

$$Loss(\theta_1, \theta_2) = \sum_{i=1}^n l(x_i, g(f(x_i; \theta_1); \theta_2)), \quad (21)$$

where $f(\cdot; \theta_1)$ and $g(\cdot; \theta_2)$ are the encoder and decoder with parameters θ_1 and θ_2 , and $l(\cdot)$ is the loss function.

Herein, we choose several representative auto-encoder-based models for network community detection, and summarize their main characteristics in Table 4. Since most auto-encoder-based methods derive network embeddings as their outputs (e.g., [100], [103]), clustering, such as K-means and spectral clustering, is subsequently applied to extract communities. An alternative is to integrate clustering into the model (e.g., [105], [108]), to directly discover communities. Depending on the type of auto-encoder used, we divide the models into four types, namely stacked, sparse, denoising and variational auto-encoders. Stacked auto-encoder, a basic type that consists of a series of auto-encoders, is often used as a block for other types of auto-encoders. Particularly, when a stacked model has other targets, such as sparsity and denoise, we classify them as sparse or denoising auto-encoders.

Stacked auto-encoders. The semi-DNR [100] stacks a sequence of auto-encoders to form a deep nonlinear reconstruction of the input networks (DNR), and requires that each layer of the encoder contains fewer neurons than the previous layer to reduce the data dimension and extract the most salient features in the input data. Semi-DNR makes full use of the prior knowledge if v_i and v_j belong to the same community to incorporate pairwise constraints between the

two nodes in the network. Specifically, it defines a prior information matrix $O = (o_{ij})_{n \times n}$, where $o_{ij} = 1$ if v_i and v_j are known to be in the same community, or 0, otherwise. The loss function for semi-DNR is represented as:

$$Loss = l(M, Z) + \lambda \text{Tr}(H^T L H), \quad (22)$$

where $L = D - O$, $\text{Tr}(\cdot)$ is the trace of a matrix, M the modularity matrix, Z the reconstruction data, H the representation matrix and λ a parameter for making a trade-off between the reconstruction error and consistency of the new representation given the prior information. Further, a layer-wise stacked auto-encoder in DeCom [105] is adopted to find seed nodes and add nodes to communities according to the structure of the network. It is remarkable that DeCom is suitable for handling large networks and there is no need to pre-define the number of communities due to the adaptive learning process. Besides, CDDTA [104] effectively combines transfer learning and auto-encoder. AAGR [103] and DIR [101] utilize stacked auto-encoders to incorporate the information of topology and attributes adaptively, thus well realizing the balance between network topology and node attributes. NEC [106] employs graph convolutional networks to encode and decode network data, which takes topology and attribute information as input, but only selects to reconstruct the adjacency matrix to ensure that the model can still work without node attributes.

Sparse auto-encoders. Large-scale networks are in general difficult to store and process, so it is necessary to have a sparse representation. A new line of research is to adaptively find the optimal representation by adding a sparse constraint to auto-encoder for this purpose. GraphEncoder [107] introduces an explicit regularization term for the hidden layer to restrict the size of hidden representation. If z_i is the i -th vector of reconstructed data, the reconstruction error with sparsity constraints are as follows:

$$Loss = \sum_{i=1}^n \|z_i - x_i\|_2 + \beta KL(\rho|\hat{\rho}), \quad (23)$$

where β controls the sparsity penalty, ρ and $\hat{\rho}$ are the sparsity parameters, where the former denotes the average activation of a neuron across a collection of training samples, and the latter denotes the average activation across all training samples. The time complexity of GraphEncoder is $O(nbd)$, where b is the maximum number of hidden layer nodes, and d is the average degree of the graph. DFuzzy [108] is a parallel and scalable fuzzy clustering model with sparse auto-encoders as building blocks. It trains an auto-encoder using personalized PageRank, which is effective for capturing relationships among network nodes. Besides, CD-MEC [109] combines transfer learning with auto-encoder, where input matrix A is used to build four similarity matrices of complex networks. CDMEC takes one matrix as the source domain, and the other three matrices as the target domain to obtain multiple distinct low-dimensional feature representations. All representations are then put into a clustering algorithm, and the clustering results are integrated into a new, consensus matrix Q . The consensus matrix Q is introduced to measure the co-occurrence of samples in the clustering result, where Q_{ij} represents the

average times that v_i and v_j are grouped into the same class.

Denosing auto-encoders. Denoising auto-encoders can be applied to noisy inputs to get node representation that is robust to noise. MAGE [24] first employs a convolutional network to integrate content and structure information, and then iteratively adds random noises to content information in the auto-encoder process. In this way, the structure information and content information are integrated into a unified framework, and the interplay between the two can be analyzed. Further, Yang *et al.* [110] propose GRACE to deal with dynamic networks. They model clusters under the consideration of network dynamics, and believe that the formation of clusters requires dynamic embedding to reach a stable state.

Variational auto-encoders. There are also approaches based on variational auto-encoder [122], which views the hidden representation as a latent variable with its own prior distribution. In variational inference, it exploits an approximation $q(H|X)$ of the true posterior $p(H|X)$ of the latent variable and tries to approximate the variational posterior $q(H|X)$ to the true prior $p(H)$ using the KL-divergence as a measure. For instance, the theme of ARVGA [111] is not only to minimize the reconstruction errors of network structure, but also to enforce the latent codes to match a prior distribution:

$$Loss = \mathbb{E}_{q(H|(X,A))} [\log p(\hat{A}|H)] - KL[q(H|X, A)||p(H)]. \quad (24)$$

During the training of VGAECD [112], the reconstruction loss deviates from its primary objective of clustering. The new VGAECD [114] rectifies this issue by introducing a dual variational objective. Further, Sarkar *et al.* [115] propose a gamma ladder VAE based deep generative model that infers multilayered embeddings for the nodes via multiple layers of stochastic latent variables to improve the performance of community detection. Jin *et al.* [116] propose the NetVAE that uses one encoder and one dual decoder with two different generative mechanisms to reconstruct network topology and node attributes separately.

4.2 Generative Adversarial Network-based Methods

Generative adversarial networks (GANs) [123], which are inspired by the minimax two-player game, have achieved unprecedented success in various fields. GANs typically consist of two modules, a generator \mathcal{G} and a discriminator \mathcal{D} . The generator is to capture the data distribution, i.e., to generate samples that are as similar to the real data as possible; while the discriminator is to estimate the probability that a sample is a piece of real data rather than synthetic data generated by the generator. Formally, the training process of GANs can be defined as:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{V}(\mathcal{G}, \mathcal{D}) = \min_{\mathcal{G}} \max_{\mathcal{D}} (\mathbb{E}_{x \sim p_{data}(x)} [\log \mathcal{D}(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - \mathcal{D}(\mathcal{G}(z)))]), \quad (25)$$

where the first expectation is the loss of discriminator for real data and the second is the loss of discriminator for synthetic data generated by the generator.

The inspiration of applying GANs to community detection came from the fact that GANs are usually unsupervised, and (in theory) the new data generated have the

same distribution as real data, which provides a powerful network data analysis capability. Jia *et al.* [26] propose a novel method called CommunityGAN to adopt the idea of affiliation graph model (AGM) to boost the performance by introducing the minimax competition between the network motif-level generator and the discriminator. It first composes some representation vectors of nodes by assigning each node-community pair a nonnegative factor that represents the degree of membership of the node to community, and then optimizes such representation through a specifically designed GAN to detect communities. The joint value function is formulated as:

$$\min_{\theta_G} \max_{\theta_D} \mathcal{V}(\mathcal{G}, \mathcal{D}) = \sum_{i=1}^n (\mathbb{E}_{m \sim p_{true}(\cdot|v_i)} [\log \mathcal{D}(m; \theta_D)] + \mathbb{E}_{s \sim \mathcal{G}(s|v_i; \theta_G)} [\log(1 - \mathcal{D}(\mathcal{G}(s; \theta_D)))]), \quad (26)$$

where θ_D (and θ_G) is the union of representation vectors of all nodes in the discriminator \mathcal{D} (and generator \mathcal{G}), m the motifs of networks and s the subset of nodes. By employing GANs, CommunityGAN can find overlapping communities and learn a graph representation altogether. Further, Zhang *et al.* [27] present a novel approach of seed expansion with generative adversarial learning (SEAL). It employs a discriminator to predict whether a community is real or not and a generator to construct communities to trick the discriminator by implicitly fitting features of real ones for learning heuristics for community detection. Tao *et al.* [124] propose an adversarial graph auto-encoder (AGAE) method, which incorporates ensemble clustering into a deep graph embedding process to guide the network training utilizing an adversarial regularizer.

There are also methods based on GANs to derive node representation that can be applied to community detection, e.g., employing clustering algorithms such as K-means for deriving embeddings to acquire resulting communities [125], [126], [127]. He *et al.* [128] further argue that the existing GANs-based methods do not make full use of the essential advantages of GANs, which are to learn the underlying representation mechanism rather than the representation itself. To this end, they propose to utilize adversarial idea on the representation mechanism to acquire node representation for downstream tasks. Specifically, the training loss is defined as follows:

$$\min_{\mathcal{E}, \mathcal{G}} \max_{\mathcal{D}} \mathcal{V}(\mathcal{G}, \mathcal{D}, \mathcal{E}) = \mathbb{E}_{x \sim p_{data}(x)} [\log \mathcal{D}(MI(x, \mathcal{E}(x)))] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - \mathcal{D}(MI(\mathcal{G}(z), z)))]), \quad (27)$$

where \mathcal{E} represents the encoder that derives node representation, $MI(x, \mathcal{E}(x))$ is the mutual information between the node attributes and node representation, \mathcal{D} is the discriminator that identifies the mutual information from either positive or negative samples, and \mathcal{G} is the generator that generates negative samples by calculating the mutual information between fake node attributes based on Gaussian noise. Yang *et al.* [129] argue that most GANs compare the results of embeddings with samples obtained from Gaussian distribution without rectification from real data, making them not truly beneficial for adversarial learning. Therefore, they design a joint adversarial network embedding (JANE) model, which jointly distinguishes the real and fake combi-

nations of embeddings, topology information and node attributes, to improve node embeddings and the performance of network analysis.

4.3 Graph Convolutional Network-based Methods

Graph convolutional networks (GCNs) [33], the most representative branch of *graph neural network* methods [130] for learning representation from graph data, have attracted a great deal of attention thanks to its success on supervised and semi-supervised classification of nodes in a network. Several novel GCNs-based algorithms have also been developed lately to exploit the power of GCNs for effectively modeling and inferring high-dimensional complex network data for community detection.

Jin *et al.* [131] raise the concern that embeddings derived from GCNs are not community-oriented and community detection is inherently unsupervised. To address this problem, they introduce an unsupervised model, named JGE-CD, for community detection through joint GCN embedding. It consists of three modules, a dual encoder that derives two embeddings using the original attribute network and its variant; a community detection module that stacks on top of dual encoder to detect community; and a topology reconstruction module that is employed to reconstruct network topology. Formally, the probability that the i -th node belongs to the r -th community is defined as:

$$u_{ir} = \frac{\exp(\theta_r^T h_i)}{\sum_{r=1}^k \exp(\theta_r^T h_i)}, \quad (28)$$

where h_i represents the embedding of node v_i obtained from GCN and θ the model parameters. Furthermore, He *et al.* [132] extend JGE-CD by designing a new GCN approach that casts MRFasGCN (to be discussed in Section 4.4 shortly) as an encoder, and exploits a community-centric dual encoder to reconstruct network topology and node attributes separately, so as to perform unsupervised community detection. In particular, the decoder for reconstructing network topology is denoted as:

$$\hat{A} = \text{sigmoid} \left(DUWU^T D^T \right), \quad (29)$$

where U is the probability distribution matrix of nodes belonging to different communities derived from the encoder, D the node degree matrix and W the weight matrix of neural networks. The decoder for reconstructing attributes is inspired by topic modeling, i.e., nodes in the same community are more likely to have similar distributions of attribute words. The attribute matrix can be generated by:

$$\hat{X} = U \cdot R, \quad (30)$$

where the definition of U is the same as that in (29) and R is the probability matrix of communities selecting attribute words from the entire word set.

More recently, some researches for community detection make use of GCNs on heterogeneous networks that contain a diversity of types of nodes and relationships. Zheng *et al.* [133] design a heterogeneous-temporal GCN, namely HTGCN, to detect community from heterogeneous and temporal networks. Concretely, it first obtains feature representation of each heterogeneous network at each time

step by adopting a heterogeneous GCN, and then utilizes a residual compressed aggregation mechanism to express both the static and dynamic characteristics of community. Beyond that, there are also certain approaches incorporating graph convolutional network with undirected graphical models, e.g., MRFAsgCN [92] and GMNN [93], which will be discussed next.

4.4 Integrating Graph Convolutional Network and Undirected Graphical Models

In the last few years, a number of studies have begun to integrate graph convolutional network (GCN) and undirected graphical models (e.g., MRF or CRF) for community detection. The main idea of this line of research is that GCN essentially constructs node embeddings through local feature smoothing, which does not consider community properties and makes the node embeddings not community-oriented. While undirected graphical models generally offer a good global objective to describe community, it does not consider information on nodes and requires a substantial amount of computation for learning the model. Therefore, GCN and undirected graphical models are complementary and can be combined to take advantage of their strengths.

A major work in this line is MRFAsgCN [92], which integrates GCN with MRF to solve the problem of semi-supervised community detection in attributed networks. The method first extends NetMRF (as discussed in Section 3.2) to extended MRF (eMRF) by adding both unary potentials and attribute information, and then reparameterizes the MRF model to make it fit to the GCN architecture. The energy function of eMRF is defined as:

$$E(C; A, X) = \sum_{v_i} -p(h_i^{c_i}) + \alpha \sum_{v_i \neq v_j} \mu(c_i, c_j) \eta(v_i, v_j), \quad (31)$$

where $-p(h_i^{c_i})$, whose value comes from the result of GCN, denotes the unary potential representing the probability that node v_i belongs to community c_i , $\mu(c_i, c_j) \eta(v_i, v_j)$ is the pairwise potential where $\mu(c_i, c_j)$ represents the similarity relationship between communities of nodes v_i and v_j , and $\eta(v_i, v_j)$ is the similarity of attributes of nodes v_i and v_j , and α is the parameter for making a tradeoff between the unary and pairwise potentials. The time complexity of MRFAsgCN is $O(mqlk^2)$, where q is the dimension of node attribute and l is the number of hidden units of the first layer.

After MRFAsgCN, several other lines of work incorporate MRF or CRF into GCN to learn node embedding for community detection. Qu *et al.* [93] propose a new approach, called graph Markov neural network (GMNN), combines the advantages of both statistical relational learning and graph neural networks. A GMNN is able to learn an effective node representation and model label dependency between different nodes, thereby completing the task of semi-supervised node classification. The model parameters can be learned by employing pseudolikelihood variational expectation-maximization [134] to optimize the evidence lower bound (ELBO) of log-likelihood function, which is formulated as:

$$\log p_\phi(y_L|x_V) \geq E_{q_\theta(y_U|x_V)}[\log p_\phi(y_L, y_U|x_V) - \log q_\theta(y_U|x_V)], \quad (32)$$

where $\log p_\phi(y_L|x_V)$ is the log-likelihood function of observed node labels, and $q_\theta(y_U|x_V)$ is any distribution over y_U . Noting that the equality holds when $q_\theta(y_U|x_V) = p_\phi(y_U|y_L, x_V)$. Gao *et al.* [135] find that the existing GCNs fail to preserve the similarity relationship between different nodes hidden in network data. To handle this issue, they add a CRF layer to GCNs to force similar nodes to have similar hidden features. This will enhance the quality of node embeddings and, in turn, improve the performance of network analysis.

4.5 Summary for deep learning

Deep learning-based methods usually realize community detection by designing a community-oriented or universal network representation, followed by some clustering algorithms. Different types of deep-learning methods have been adopted to derive network representations, e.g., GCN or GAN. Learning such representations is to map network data from a high-dimensional input space to a low-dimensional feature space, with the superiority of low computational complexity and effective data fusion. However, similar to the most probabilistic graphical models, the existing deep learning-based methods are more suitable for community structures with homophily (where the nodes in a community are densely connected whereas nodes in different communities are sparsely linked), which may limit the robustness of these models.

5 APPLICATIONS OF COMMUNITY DETECTION

We start our discussion with a summary of the benchmark datasets that have been used in the area of community detection. We then describe real applications of community detection in many application fields.

5.1 Open Datasets

We have put the detailed information of the datasets used for community detection in a publicly accessible web¹ to facilitate open research on this rapidly-developing topic. These datasets can be separated into two groups, synthetic networks and real-world networks.

5.1.1 Synthetic networks

There are two classes of randomly generated synthetic networks with known community structures, i.e., the Girvan-Newman (GN) [1] and LFR networks [136]. The GN network consists of four non-overlapping communities with the same size. Each community has 32 nodes, each of which connects with 16 other nodes on average. Among these 16 edges, Z_{in} edges connect to nodes of the same community, Z_{out} edges to nodes of different communities, and $Z_{in} + Z_{out} = 16$. The LFR network, another widely adopted benchmark for testing the performance of algorithms for community detection, has distributions of node degree and community size which follow power laws with tunable exponents. The LFR network captures several important features of real-world systems, e.g., the scale free property.

1. <http://bdilab.tju.edu.cn/>

5.1.2 Real-world Networks

The real-world networks that we examine include four types, i.e., social networks, citation networks, collaboration networks, and others, listed in Table 5. To be specific, social networks are formed by individuals and their interactions, including eight representative datasets such as Football and DBLP (Table 5). Citation networks consist of papers (or patents) and their relationships (e.g., citation or inclusion), including eight classic datasets such as Cora and arXiv. Collaboration networks are comprised of scientists and their collaborations (i.e., co-authoring papers), including four typical datasets such as Computer Science and Medicine. The nodes in these networks range from tens to millions, and the maximum number of edges reaches hundreds to hundreds of millions.

TABLE 5: The statistics of real-world networks.

Categories	Datasets	#Nodes	#Edges
Social Networks	Friendship7 [37]	68	220
	Football [1]	115	613
	Facebook [137]	1,045	26,749
	LiveJournal [138]	44,093	871,409
	Twitter [27]	87,760	1,293,985
	Orkut [138]	297,691	7,747,026
	DBLP [139]	317,080	1,049,866
	Youtube [139]	1,134,890	2,987,624
Citation Networks	Small-hep ²	397	812
	Polblogs [140]	1490	16718
	Cora [141]	2708	5429
	Citeseer [141]	3312	4732
	Large-hep ²	11,752	134,956
	Pubmed [142]	19,729	44,338
	arXiv [143]	576,000	6,640,000
	US patents [144]	3,700,000	16,500,000
Collaboration Networks	Computer Science [145]	22,000	96,800
	Chemistry [145]	35,400	157,400
	Medicine [145]	63,300	810,300
	Engineering [145]	14,900	49,300
Others	Amazon [139]	334,863	925,872
	Google [146]	875,000	4,320,000

5.2 Practical Applications

We first discuss the applications of community detection on different domains, and then extend to other network analysis tasks. We finish by discussing the potential of community detection to network science.

5.2.1 Applications in Different Areas

Community detection has diverse applications across different domains [147], such as online social networks, neuroscience and image understanding. Online social networks, including Facebook, Twitter and Wechat, comprise the interactions among people through the web. Discovering community in such networks is an effective way to infer the relationships of individuals, which has been adopted for tasks such as spammer detection and crisis response. Yin *et al.* [148] develop a unified probabilistic generative model, namely user-community-geo-topic (UCGT), based on a Bayesian network that infers users' social communities by incorporating spatio-temporal data and semantic information to improve the accuracy and interpretability of social community detection. Wu *et al.* [149] design a novel MRFwithGCN model, which introduce a MRF layer

that captures user following information to refine prediction made by GCN for social spammer detection.

Neuroscience is a discipline studying the nervous systems and brain. With the recent development of brain mapping and neuroimaging techniques, the brain has begun to be modeled as networks. A large amount of effort has been put forward to exploit such networks to help extract the functional subdivisions of the brain. Liu *et al.* [150] propose a framework of siamese community preserving graph convolutional network (SCP-GCN). The method first retains the community structure considering the intra-community and inter-community properties in learning process, and then uses siamese architecture that models the pair-wise similarity to guide this learning process. Jin *et al.* [151] argue that the existing studies typically construct community structures of brain networks employing resting-state functional magnetic resonance imaging (fMRI) data. They introduce the dynamic time warping (DTW) algorithm that analyzes the synchronization and asynchronism of fMRI time series to extract the correlation between brain regions.

Image understanding is to generate semantic descriptions of images. Recently, works on image understanding utilizing community have emerged. Li *et al.* [152] present a novel deep collaborative embedding (DCE) model, which learns knowledge from weakly supervised community-contributed resources, for multiple image understanding tasks simultaneously. Li *et al.* [153] propose a novel semi-supervised RSNMF model by explicitly exploring the block-diagonal structure of data for image representation.

5.2.2 Promotion of Network Analysis

With the great success of community detection, numerous application problems, e.g., recommendation and link prediction, have been formulated as finding community structures in network systems. We now discuss how the existing community detection methods are utilized to solve some of these problems.

Recommendation is a common task that addresses the issue of information overload for users by establishing a profile of user interests based on items in their purchasing or browsing history and later recommending similar items to users. The existing methods for recommendation including collaborative filtering [154] and neural networks [155]. In particular, the concept of community has been employed to improve the quality of recommendation. Eissa *et al.* [156] make recommendation based on interest-based communities that generated from topic based attributed social networks. Satuluri *et al.* [2] present a general-purpose representation layer, i.e., similarity-based clusters (SimClusters), which settles a multitude of recommendation tasks at Twitter via detecting bipartite communities from the user-user network and leveraging them as a representation space.

Link prediction is another important task in network mining. It deals with missing connections and predicts possible connections in the future through the analysis of observed network structure and external information. A large number of approaches have been proposed to facilitate link prediction via considering community. Xu *et al.* [157] indicate that the existing metric learning for link prediction ignores community that contain abundant structural information. Therefore, they design community specific similar-

2. <https://www.cs.cornell.edu/projects/kddcup/datasets.html>

ity metrics by means of joint community detection to deal with cold-start link prediction where edges between nodes are unavailable. De *et al.* [158] propose a stacked two-level learning framework, which first learns a local similarity model exploiting locality structures and node attributes, and then combines the model with community-level features that derive utilizing co-clustering for link prediction.

5.2.3 Promotion of Network Science

Network science is an interdisciplinary research area, which can be used not only in computer science, but also in other fields such as sociology and biology. Community detection, one of the most important problem in network analysis, can tremendously promote the development of network science. For example, in citation network, nodes represent papers and edges represent the citations among papers. By grouping the papers (i.e., discovering communities where papers have similar attributes such as belonging to the same author or topic), we can analyze the influence of authors and accurately grasp the latest research trends or technologies, which is of guiding significance for comprehending network and further analyzing network pattern [159]. Similarly, in sociology and biology, community detection also provides a deeper understanding of network structure and promotes its development both in academia and industry.

6 FUTURE DIRECTIONS

While learning-based community detection, including probabilistic graphical model and deep learning, has demonstrated superior performance across a variety of problems and domains, there are challenges that need to be addressed. In this section, we briefly discuss these challenges and future research directions potentially worth pursuing.

6.1 Large Networks

With the rapidly increasing scale of network data, more large networks have become the standard across many different scientific domains. These networks typically have tens of thousand or billions of nodes and edges as well as complex structural patterns. Most existing community detection methods face excessive constraints on such large networks due to the potentially prohibitive demand on memory and computation. They may require a large number of training instances [160] or model parameters [161] to make the existing methods effective. Moreover, the existing approaches typically handle these problems by network reduction or approximation, which may lose some important network information and affect the modeling accuracy. This raises the question of how to devise a framework that far exceeds the current benchmarks in accuracy and efficiency.

6.2 Community Interpretability

Although community detection has been studied for more than a decade, the interpretability of community remains an important and critical issue to be adequately addressed. Most current community detection methods utilize top ranked words or short phrases in the results to summarize communities, even though the attribute information of

nodes is typically complete sentences that have more information than individual words [73], [162]. However, these methods may not be intuitive enough for understanding the semantics of communities due to the small number of words and unclear relationship between words. How to make the best use of network information to provide a better semantic interpretation for community is one of future research directions.

6.3 Adaptive Community Model Selection

Adaptive model selection for community detection aims to choose the most appropriate algorithm for discovering community, according to the characteristics of different networks (e.g., heterogeneous or dynamic) or specific requirements of different tasks (e.g., the highest accuracy or the lowest time complexity). Although the existing methods can be extended from one network or task to another to some extent (which inevitably affects the accuracy and stability of the resulting model) [163], [164], few of them consider how to perform model adaptation. Thus, focus has shifted to designing a unified architecture that can automatically adapt to specific tasks or networks while maintaining model accuracy and stability instead of proposing diverse frameworks for different networks or tasks. This is an emerging research area that would be challenging but rewarding.

6.4 Networks with Complex Structures

Many real-world networks are heterogeneous, dynamic, hierarchical, or incomplete. Heterogeneous networks [165] are those that contain different types of nodes and edges, or different types of descriptions on nodes and edges, such as text and images. Dynamic networks [166] are networks whose topology and/or attributes change over time. Dynamic networks appear when nodes and edges are added or deleted, thus altering the properties of nodes or edges. Hierarchical networks [167] are composed of several layers, each of which has specific semantics and functions. Incomplete networks [168] are the ones with missing information of their topology, nodes, or edges. While these networks can be partly explored by the learning-based community detection, there still exist several serious issues. First, most existing methods assume homogeneous networks, which may in fact difficult to handle. Second, due to the variability of dynamic networks, most existing methods, especially the ones based on deep learning, need to be re-trained over a series of steps when the networks evolve, which is very time consuming and may not meet the real time processing demand. Third, hierarchical networks typically have different types of relationships across the network hierarchies, which are important while often not well handled by the existing methods. Moreover, almost all the existing methods regard the networks to be analyzed to be complete and accurately documented without noise. Unfortunately, this is rarely the case in practice as it is challenging to obtain complete information of the networks. Therefore, new methods should be developed to handle these issues to better improve the performance of community detection on these types of complex networks.

6.5 Integrating Statistical Modeling and Deep Learning

Although several methods have been proposed to combine statistical modeling with deep learning, such as MRFas-GCN, it is still a virgin but promising research area. For instance, the existing methods typically utilize the prior knowledge (e.g., communities) that statistical model offers to refine the embeddings of GCN to improve resulting communities. However, these methods may not fully consider the time complexity or interpretability of the models, raising enormous challenges to community detection in practice. In addition, the community patterns in the real world networks are usually diverse, e.g., community structure with heterophily (where the nodes in different communities are closely connected while the nodes in the same community are sparsely connected) or randomness (where the edges between nodes are more likely to be randomly generated), it is essential to design new robust methods by integrating statistical modeling and deep learning [92], thereby detecting the community structure in the network more accurately. Furthermore, it remains an open problem to integrate statistical modeling in deep learning methods. For example, it is difficult to apply a strategy for recommendation or medical diagnosis to make deep learning become better representation learning, which in turn facilitates more accurate recommendation or diagnose. New innovative algorithms are highly desirable to integrate statistical inference and deep learning to help deep learning more produce interpretable network representation models that are suitable for various network problems in broad application fields.

7 CONCLUSION

In this paper, we provide a comprehensive and up-to-date literature review on the community detection approaches. One of our main objectives is to organize and present most of the work conducted so far in a unified perspective. In a first step, we discuss in details the problem of community detection, and provide a new taxonomy to group most existing methods into two categories from the perspective of learning: probabilistic graphical model and deep learning. We then thoroughly review, compare, and summarize the existing methods in these two categories, and discuss how some of these methods can be interested. Moreover, since the problem is highly application oriented, we introduce a wide range of applications of community detection in various fields. We also highlight that more effort is needed to address several challenging open problems for the research of community detection. Last but not least, while we cannot list all the literature on community detection, we expect that our view that attempts to synthesize the state-of-the-art of community detection will contribute to a better understanding of this highly active and increasingly important area of study in network science, serve as a source of information for new researchers entering this field and the researchers working in this area, and promote future developments of next-generation community detection approaches.

REFERENCES

- [1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [2] V. Satuluri, Y. Wu, X. Zheng, Y. Qian, B. Wichers, Q. Dai, G. M. Tang, J. Jiang, and J. Lin, "Simclusters: Community-based representations for heterogeneous recommendations at twitter," in *Proceedings of SIGKDD*, pp. 3183–3193, 2020.
- [3] S. Mukherjee, H. Lamba, and G. Weikum, "Experience-aware item recommendation in evolving review communities," in *Proceedings of ICDM*, pp. 925–930, 2015.
- [4] M. R. Keyvanpour, M. B. Shirzad, and M. Ghaderi, "AD-C: a new node anomaly detection based on community detection in social networks," *Int. J. Electron. Bus.*, vol. 15, no. 3, pp. 199–222, 2020.
- [5] J. Wang and I. C. Paschalidis, "Botnet detection based on anomaly and community detection," *IEEE Trans. Control. Netw. Syst.*, vol. 4, no. 2, pp. 392–404, 2017.
- [6] F. Saidi, Z. Trabelsi, and H. B. Ghezala, "A novel approach for terrorist sub-communities detection based on constrained evidential clustering," in *Proceedings of RCIS*, pp. 1–8, 2018.
- [7] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, 1977.
- [8] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [9] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [10] C. C. Aggarwal and H. Wang, "Managing and mining graph data," *Advances in Database Systems*, vol. 40, pp. 275–301, 2010.
- [11] S. Jia, L. Gao, Y. Gao, J. Nastos, Y. Wang, X. Zhang, and H. Wang, "Defining and identifying cograph communities in complex networks," *New J. Phys.*, vol. 17, no. 1, p. 013044, 2015.
- [12] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," in *Proceedings of IJCAI*, pp. 2252–2258, 2016.
- [13] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 26113, 2004.
- [14] P. Zhang and C. Moore, "Scalable detection of statistically significant communities and hierarchies, using message passing for modularity," *Proc. Natl. Acad. Sci.*, vol. 111, no. 51, pp. 18144–18149, 2014.
- [15] M. Fanuel, C. M. Alaíz, and J. A. K. Suykens, "Magnetic eigenmaps for community detection in directed networks," *Phys. Rev. E*, vol. 95, no. 2, p. 022302, 2017.
- [16] Y. Li, K. He, D. Bindel, and J. E. Hopcroft, "Uncovering the small community structure in large networks: A local spectral approach," in *Proceedings of WWW*, pp. 658–668, 2015.
- [17] A. Anandkumar, R. Ge, D. J. Hsu, and S. M. Kakade, "A tensor approach to learning mixed membership community models," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2239–2312, 2014.
- [18] D. He, D. Liu, D. Jin, and W. Zhang, "A stochastic model for detecting heterogeneous link communities in complex networks," in *Proceedings of AAAI*, pp. 130–136, 2015.
- [19] C. Pizzuti and A. Socievole, "Multiobjective optimization and local merge for clustering attributed graphs," *IEEE Trans. Cybern.*, pp. 1–13, 2019.
- [20] Z. Li, J. Liu, and K. Wu, "A multiobjective evolutionary algorithm based on structural and attribute similarities for community detection in attributed networks," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 1963–1976, 2018.
- [21] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Min. Knowl. Discov.*, vol. 22, no. 3, pp. 493–521, 2011.
- [22] Y. Zhang and D. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," in *Proceedings of SIGKDD*, pp. 606–614, 2012.
- [23] B. Yang, X. Zhao, and X. Liu, "Bayesian approach to modeling and detecting communities in signed network," in *Proceedings of AAAI*, pp. 1952–1958, 2015.
- [24] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: marginalized graph autoencoder for graph clustering," in *Proceedings of CIKM*, pp. 889–898, 2017.
- [25] B. Sun, H. Shen, J. Gao, W. Ouyang, and X. Cheng, "A non-negative symmetric encoder-decoder approach for community detection," in *Proceedings of CIKM*, pp. 597–606, 2017.

- [26] Y. Jia, Q. Zhang, W. Zhang, and X. Wang, "Communitygan: Community detection with generative adversarial nets," in *Proceedings of WWW*, pp. 784–794, 2019.
- [27] Y. Zhang, Y. Xiong, Y. Ye, T. Liu, W. Wang, Y. Zhu, and P. S. Yu, "SEAL: learning heuristics for community detection with generative adversarial networks," in *Proceedings of SIGKDD*, pp. 1103–1113, 2020.
- [28] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Soc. Networks*, vol. 5, no. 2, pp. 109–137, 1983.
- [29] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, 2008.
- [30] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E*, vol. 83, no. 1, 2011.
- [31] D. He, X. You, Z. Feng, D. Jin, X. Yang, and W. Zhang, "A network-specific Markov random field approach to community detection," in *Proceedings of AAAI*, pp. 306–313, 2018.
- [32] G. Sperl, "A deep learning based community detection approach," in *Proceedings of SAC*, pp. 1107–1110, 2019.
- [33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of ICLR*, 2017.
- [34] D. Jin, C. Huo, C. Liang, and L. Yang, "Heterogeneous graph neural network via attribute completion," in *Proceedings of WWW*, pp. 391–400, 2021.
- [35] Y. Yang, L. Chu, Y. Zhang, Z. Wang, J. Pei, and E. Chen, "Mining density contrast subgraphs," in *Proceedings of ICDE*, pp. 221–232, 2018.
- [36] D. Jin, X. Wang, D. He, J. Dang, and W. Zhang, "Robust detection of link communities with summary description in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2737–2749, 2021.
- [37] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 1–35, 2013.
- [38] C. C. Aggarwal and K. Subbian, "Evolutionary network analysis: A survey," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–36, 2014.
- [39] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 464–483, 2018.
- [40] E. Abbe, "Community detection and stochastic block models: Recent developments," *J. Mach. Learn. Res.*, vol. 18, pp. 177:1–177:86, 2017.
- [41] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, and P. S. Yu, "Deep learning for community detection: Progress, challenges and opportunities," in *Proceedings of IJCAI*, pp. 4981–4987, 2020.
- [42] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Phys. Rep.-Rev. Sec. Phys. Lett.*, vol. 533, no. 4, pp. 95–142, 2013.
- [43] T. Hartmann, A. Kappes, and D. Wagner, "Clustering evolving networks," vol. 9220, pp. 280–329, 2016.
- [44] C. Lee and D. J. Wilkinson, "A review of stochastic block models and extensions for graph clustering," *Appl. Netw. Sci.*, vol. 4, no. 1, p. 122, 2019.
- [45] T. A. Snijders and K. Nowicki, "Estimation and prediction for stochastic blockmodels for graphs with latent block structure," *J. Classif.*, vol. 14, no. 1, pp. 75–100, 1997.
- [46] P. Zhang, F. Krzakala, J. Reichardt, and L. Zdeborová, "Comparative study for inference of hidden classes in stochastic block models," *J. Stat. Mech.-Theory Exp.*, vol. 2012, no. 12, p. 12021, 2012.
- [47] X. Fan, R. Y. D. Xu, and L. Cao, "Copula mixed-membership stochastic block model," in *Proceedings of IJCAI*, 2016.
- [48] J. W. Miller and M. T. Harrison, "Mixture models with a prior on the number of components," *J. Am. Stat. Assoc.*, vol. 113, no. 521, pp. 340–356, 2018.
- [49] S. Pal and M. Coates, "Scalable MCMC in degree corrected stochastic block mode," in *Proceedings of ICASSP*, pp. 5461–5465, 2019.
- [50] Y. Zhao, E. Levina, and J. Zhu, "Consistency of community detection in networks under degree-corrected stochastic block models," *Ann. Stat.*, vol. 40, no. 4, pp. 2266–2292, 2012.
- [51] L. Gulikers, M. Lelarge, and L. Massoulié, "A spectral method for community detection in moderately sparse degree-corrected stochastic block models," *Adv. Appl. Probab.*, vol. 49, no. 3, pp. 686–721, 2017.
- [52] J. X. Yudong Chen, Xiaodong Li, "Convexified modularity maximization for degree-corrected stochastic block models," *Ann. Stat.*, vol. 46, no. 4, pp. 1573–1602, 2018.
- [53] K. Chen and J. Lei, "Network cross-validation for determining the number of communities in network data," *J. Am. Stat. Assoc.*, vol. 178, no. 113, p. 521, 2018.
- [54] W. Fu, L. Song, and E. P. Xing, "Dynamic mixed membership blockmodel for evolving networks," in *Proceedings of ICML*, pp. 329–336, 2009.
- [55] E. P. Xing, W. Fu, and L. Song, "A state-space mixed membership blockmodel for dynamic network tomography," *Ann. Appl. Stat.*, vol. 4, no. 2, pp. 535–566, 2010.
- [56] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, "Detecting communities and their evolutions in dynamic social networks - a bayesian approach," *Mach. Learn.*, vol. 82, no. 2, pp. 157–189, 2011.
- [57] X. Tang and C. C. Yang, "Detecting social media hidden communities using dynamic stochastic blockmodel with temporal dirichlet process," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 1–21, 2014.
- [58] K. S. Xu, "Stochastic block transition models for dynamic networks," in *Proceedings of AISTATS*, vol. 38, pp. 1079–1087, 2015.
- [59] J. D. Wilson, N. T. Stevens, and W. H. Woodall, "Modeling and detecting change in temporal networks via a dynamic degree corrected stochastic block model," *Qual. Reliab. Eng. Int.*, vol. 35, no. 5, pp. 1363–1378, 2016.
- [60] X. Wu, P. Jiao, Y. Wang, T. Li, W. Wang, and B. Wang, "Dynamic stochastic block model with scale-free characteristic for temporal complex networks," in *Proceedings of DASFAA*, pp. 502–518, 2019.
- [61] M. G. Bhattacharjee M, Banerjee M, "Change point estimation in a dynamic stochastic block model," *J. Mach. Learn. Res.*, vol. 21, no. 107, pp. 1–59, 2020.
- [62] P. Latouche, E. Birmelé, C. Ambroise, et al., "Overlapping stochastic block models with application to the french political blogosphere," *Ann. Appl. Stat.*, vol. 5, no. 1, pp. 309–336, 2011.
- [63] G. Arora, A. Porwal, K. Agarwal, A. Samdariya, and P. Rai, "Small-variance asymptotics for nonparametric bayesian overlapping stochastic blockmodels," in *Proceedings of IJCAI*, pp. 2000–2006, 2018.
- [64] D. Jin, B. Li, P. Jiao, D. He, H. Shan, and W. Zhang, "Modeling with node popularities for autonomous overlapping community detection," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–23, 2020.
- [65] N. Mehta, L. Carin, and P. Rai, "Stochastic blockmodels meet graph neural networks," in *Proceedings of ICML*, vol. 97, pp. 4466–4474, 2019.
- [66] K. S. Xu and A. O. Hero, "Dynamic stochastic blockmodels for time-evolving social networks," *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 4, pp. 552–562, 2014.
- [67] X. Wu, P. Jiao, Y. Wang, T. Li, W. Wang, and B. Wang, "Dynamic stochastic block model with scale-free characteristic for temporal complex networks," in *Proceedings of DASFAA*, vol. 11447, pp. 502–518, 2019.
- [68] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [69] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen, "An lda-based community structure discovery approach for large-scale social networks," in *Proceedings of ISI*, pp. 200–207, 2007.
- [70] Z. Yin, L. Cao, Q. Gu, and J. Han, "Latent community topic analysis: Integration of community discovery with topic modeling," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, pp. 63:1–63:21, 2012.
- [71] Y. Cha and J. Cho, "Social-network analysis using topic models," in *Proceedings of SIGIR*, pp. 565–574, 2012.
- [72] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, "A model-based approach to attributed graph clustering," in *Proceedings of COMAD*, pp. 505–516, 2012.
- [73] D. He, Z. Feng, D. Jin, X. Wang, and W. Zhang, "Joint identification of network communities and semantics via integrative modeling of network topologies and node contents," in *Proceedings of AAAI*, pp. 116–124, 2017.
- [74] D. Jin, K. Wang, G. Zhang, P. Jiao, D. He, F. Fogelman-Soulie, and X. Huang, "Detecting communities with multiplex semantics by distinguishing background, general and specialized topics," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2144–2158, 2020.
- [75] J. He, Z. Hu, T. Berg-Kirkpatrick, Y. Huang, and E. P. Xing, "Efficient correlated topic modeling with topic embedding," in *Proceedings of SIGKDD*, pp. 225–233, 2017.

- [76] D. Jin, J. Huang, P. Jiao, L. Yang, D. He, F. Fogelman-Soulie, and Y. Huang, "A novel generative topic embedding model by introducing network communities," in *Proceedings of WWW*, pp. 2886–2892, 2019.
- [77] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of NeurIPS*, pp. 556–562, 2000.
- [78] R.-S. Wang, S. Zhang, Y. Wang, X.-S. Zhang, and L. Chen, "Clustering complex networks and biological networks by non-negative matrix factorization with various similarity measures," *Neurocomputing*, vol. 72, no. 1-3, pp. 134–141, 2008.
- [79] D. Kuang, H. Park, and C. H. Q. Ding, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of SIAM*, pp. 106–117, 2012.
- [80] X. Shi, H. Lu, Y. He, and S. He, "Community detection in social network with pairwise constrained symmetric non-negative matrix factorization," in *Proceedings of ASONAM*, pp. 541–546, 2015.
- [81] B. Sun, H. Shen, J. Gao, W. Ouyang, and X. Cheng, "A non-negative symmetric encoder-decoder approach for community detection," in *Proceedings of CIKM*, pp. 597–606, 2017.
- [82] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proceedings of WSDM*, pp. 587–596, 2013.
- [83] Y. Pei, N. Chakraborty, and K. P. Sycara, "Nonnegative matrix tri-factorization with graph regularization for community detection in social networks," in *Proceedings of IJCAI*, pp. 2083–2089, 2015.
- [84] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic community identification in large attribute networks," in *Proceedings of AAAI*, pp. 265–271, 2016.
- [85] W. Wang, P. Jiao, D. He, D. Jin, L. Pan, and B. Gabrys, "Autonomous overlapping community detection in temporal networks: A dynamic bayesian nonnegative matrix factorization approach," *Knowl. Based Syst.*, vol. 110, pp. 121–134, 2016.
- [86] X. Ma and D. Dong, "Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1045–1058, 2017.
- [87] L. Yang, X. Cao, D. Jin, X. Wang, and D. Meng, "A unified semi-supervised community detection framework using latent space graph regularization," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2585–2598, 2015.
- [88] T. Guo, S. Pan, X. Zhu, and C. Zhang, "CFOND: consensus factorization for co-clustering networked data," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 706–719, 2018.
- [89] D. Jin, X. You, W. Li, D. He, P. Cui, F. Fogelman-Soulie, and T. Chakraborty, "Incorporating network embedding into Markov random field for better community detection," in *Proceedings of AAAI*, pp. 160–167, 2019.
- [90] D. Jin, B. Zhang, Y. Song, D. He, Z. Feng, S. Chen, W. Li, and K. Musial, "Modmrf: A modularity-based Markov random field method for community detection," *Neurocomputing*, vol. 405, pp. 218–228, 2020.
- [91] D. He, W. Song, D. Jin, Z. Feng, and Y. Huang, "An end-to-end community detection model: Integrating LDA into Markov random field via factor graph," in *Proceedings of IJCAI*, pp. 5730–5736, 2019.
- [92] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet Markov random fields: Semi-supervised community detection in attribute networks," in *Proceedings of AAAI*, pp. 152–159, 2019.
- [93] M. Qu, Y. Bengio, and J. Tang, "GMNN: graph Markov neural networks," in *Proceedings of ICML*, vol. 97, pp. 5241–5250, 2019.
- [94] X. Ma, L. Gao, X. Yong, and L. Fu, "Semi-supervised clustering algorithm for community structure detection in complex networks," *Physical A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 187–197, 2010.
- [95] S. Nowozin and C. H. Lampert, "Structured learning and prediction in computer vision," *Found. Trends Comput. Graph. Vis.*, vol. 6, no. 3-4, pp. 185–365, 2011.
- [96] J. Zeng, W. K. Cheung, and J. Liu, "Learning topic models by belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1121–1134, 2013.
- [97] Z. Yang, J. Tang, J. Li, and W. Yang, "Social Community Analysis via a Factor Graph Model," *IEEE Intell. Syst.*, vol. 26, no. 3, pp. 58–65, 2011.
- [98] Y. Jia, Y. Gao, W. Yang, J. Huo, and Y. Shi, "A novel ego-centered academic community detection approach via factor graph model," in *Proceedings of IDEAL*, vol. 8669, pp. 223–230, 2014.
- [99] A. Passarella, R. I. M. Dunbar, M. Conti, and F. Pezzoni, "Ego network models for future internet social networking environments," *Comput. Commun.*, vol. 35, no. 18, pp. 2201–2217, 2012.
- [100] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," in *Proceedings of IJCAI*, pp. 2252–2258, 2016.
- [101] J. Di, G. Meng, L. Zhixuan, L. Wenhuan, H. Dongxiao, and F. Fogelman-Soulie, "Using deep learning for community discovery in social networks," in *Proceedings of ICTAI*, pp. 160–167, 2017.
- [102] J. Cao, D. Jin, L. Yang, and J. Dang, "Incorporating network structure with node contents for community detection on large networks using deep learning," *Neurocomputing*, vol. 297, pp. 71–81, 2018.
- [103] J. Cao, D. Jin, and J. Dang, "Autoencoder based community detection with adaptive integration of network topology and node contents," in *Proceedings of KSEM*, vol. 11062, pp. 184–196, 2018.
- [104] Y. Xie, X. Wang, D. Jiang, and R. Xu, "High-performance community detection in social networks using a deep transitive autoencoder," *Inf. Sci.*, vol. 493, pp. 75–90, 2019.
- [105] V. Bhatia and R. Rani, "A distributed overlapping community detection model for large graphs using autoencoder," *Future Gener. Comput. Syst.*, vol. 94, pp. 16–26, 2019.
- [106] H. Sun, F. He, J. Huang, Y. Sun, Y. Li, C. Wang, L. He, Z. Sun, and X. Jia, "Network embedding for community detection in attributed networks," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 3, pp. 1–25, 2020.
- [107] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Liu, "Learning deep representations for graph clustering," in *Proceedings of AAAI*, pp. 1293–1299, 2014.
- [108] V. Bhatia and R. Rani, "Dfuzzy: a deep learning-based fuzzy clustering model for large graphs," *Knowl. Inf. Syst.*, vol. 57, no. 1, pp. 159–181, 2018.
- [109] R. Xu, Y. Che, X. Wang, J. Hu, and Y. Xie, "Stacked autoencoder-based community detection method via an ensemble clustering framework," *Inf. Sci.*, vol. 526, pp. 151–165, 2020.
- [110] C. Yang, M. Liu, Z. Wang, L. Liu, and J. Han, "Graph clustering with dynamic embedding," *arXiv*, 2017.
- [111] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proceedings of IJCAI*, pp. 2609–2615, 2018.
- [112] J. J. Choong, X. Liu, and T. Murata, "Learning community structure with variational autoencoder," in *Proceedings of ICDM*, pp. 69–78, 2018.
- [113] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: a deep attentional embedding approach," in *Proceedings of IJCAI*, pp. 3670–3676, 2019.
- [114] J. J. Choong, X. Liu, and T. Murata, "Optimizing variational graph autoencoder for community detection," in *Proceedings of BigData*, pp. 5353–5358, 2019.
- [115] A. Sarkar, N. Mehta, and P. Rai, "Graph representation learning via ladder gamma variational autoencoders," in *Proceedings of AAAI*, pp. 5604–5611, 2020.
- [116] D. Jin, B. Li, P. Jiao, D. He, and W. Zhang, "Network-specific variational auto-encoder for embedding in attribute networks," in *Proceedings of IJCAI*, pp. 2663–2669, 2019.
- [117] F. Sun, M. Qu, J. Hoffmann, C. Huang, and J. Tang, "vgraph: A generative model for joint community detection and node representation learning," in *Proceedings of NeurIPS*, pp. 512–522, 2019.
- [118] S. Cavallari, V. W. Zheng, H. Cai, K. C. Chang, and E. Cambria, "Learning community embedding with community detection and node embedding on graphs," in *Proceedings of CIKM*, pp. 377–386, 2017.
- [119] Z. He, J. Liu, Y. Zeng, L. Wei, and Y. Huang, "Content to node: Self-translation network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 431–443, 2021.
- [120] M. K. Rahman and A. Azad, "Evaluating the community structures from network images using neural networks," in *Proceedings of Complex Networks and Their Applications*, vol. 881, pp. 866–878, 2019.
- [121] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

- [122] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of ICLR*, 2014.
- [123] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of NeurIPS*, pp. 2672–2680, 2014.
- [124] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, "Adversarial graph embedding for ensemble clustering," in *Proceedings of IJCAI*, pp. 3562–3568, 2019.
- [125] Y. Sun, S. Wang, T. Hsieh, X. Tang, and V. G. Honavar, "MEGAN: A generative adversarial network for multi-view network embedding," in *Proceedings of IJCAI*, pp. 3527–3533, 2019.
- [126] H. Gao, J. Pei, and H. Huang, "Progan: Network embedding via proximity generative adversarial network," in *Proceedings of SIGKDD*, pp. 1308–1316, 2019.
- [127] H. Hong, X. Li, and M. Wang, "GANe: A generative adversarial network embedding," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2325–2335, 2020.
- [128] D. He, L. Zhai, Z. Li, D. Jin, L. Yang, Y. Huang, and P. S. Yu, "Adversarial mutual information learning for network embedding," in *Proceedings of IJCAI*, pp. 3321–3327, 2020.
- [129] L. Yang, Y. Wang, J. Gu, C. Wang, X. Cao, and Y. Guo, "JANE: jointly adversarial network embedding," in *Proceedings of IJCAI*, pp. 1381–1387, 2020.
- [130] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, 2020.
- [131] D. Jin, B. Li, P. Jiao, D. He, and H. Shan, "Community detection via joint graph convolutional network embedding in attribute network," in *Proceedings of ICANN*, vol. 11731, pp. 594–606, 2019.
- [132] D. He, Y. Song, D. Jin, Z. Feng, B. Zhang, Z. Yu, and W. Zhang, "Community-centric graph convolutional network for unsupervised community detection," in *Proceedings of IJCAI*, pp. 3515–3521, 2020.
- [133] Y. Zheng, S. Chen, X. Zhang, and D. Wang, "Heterogeneous graph convolutional networks for temporal community detection," *arXiv*, 2019.
- [134] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, vol. 89, pp. 355–368, 1998.
- [135] H. Gao, J. Pei, and H. Huang, "Conditional random field enhanced graph convolutional neural networks," in *Proceedings of SIGKDD*, pp. 276–284, 2019.
- [136] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E*, vol. 80, no. 1, p. 016118, 2009.
- [137] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proceedings of NeurIPS*, pp. 548–556, 2012.
- [138] S. Harenberg, G. Bello, L. Gjeltama, S. Ranshous, and N. Samatova, "Community detection in large-scale networks: A survey and empirical evaluation," *WIREs Computational Statistics*, vol. 6, no. 6, pp. 426–439, 2014.
- [139] H. V. Lierde, T. W. S. Chow, and G. Chen, "Scalable spectral clustering for overlapping community detection in large-scale networks," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 754–767, 2020.
- [140] L. A. Adamic and N. S. Glance, "The political blogosphere and the 2004 U.S. election: divided they blog," in *Proceedings of SIGKDD*, pp. 36–43, 2005.
- [141] W. Ren, G. Yan, X. Liao, and L. Xiao, "Simple probabilistic algorithm for detecting community structure," *Phys. Rev. E*, vol. 79, no. 2, p. 036111, 2009.
- [142] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.
- [143] Ginsparg and Paul, "Arxiv at 20," *Nature*, vol. 476, no. 7359, pp. 145–7, 2011.
- [144] J. Leskovec, J. M. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of SIGKDD*, pp. 177–187, 2005.
- [145] O. Shchur and S. Günnemann, "Overlapping community detection with graph neural networks," *arXiv*, 2019.
- [146] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, 2009.
- [147] L. Chu, Z. Wang, J. Pei, Y. Zhang, Y. Yang, and E. Chen, "Finding theme communities from database networks," *Proc. VLDB Endow.*, vol. 12, no. 10, pp. 1071–1084, 2019.
- [148] H. Yin, Z. Hu, X. Zhou, H. Wang, K. Zheng, N. Q. V. Hung, and S. W. Sadiq, "Discovering interpretable geo-social communities for user behavior prediction," in *Proceedings of ICDE*, pp. 942–953, 2016.
- [149] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, "Graph convolutional networks with Markov random field reasoning for social spammer detection," in *Proceedings of AAAI*, pp. 1054–1061, 2020.
- [150] J. Liu, G. Ma, F. Jiang, C. Lu, P. S. Yu, and A. B. Ragin, "Community-preserving graph convolutions for structural and functional joint embedding of brain networks," in *Proceedings of BigData*, pp. 1163–1168, 2019.
- [151] D. Jin, R. Li, and J. Xu, "Multiscale community detection in functional brain networks constructed using dynamic time warping," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 52–61, 2020.
- [152] Z. Li, J. Tang, and T. Mei, "Deep collaborative embedding for social image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2070–2083, 2019.
- [153] Z. Li, J. Tang, and X. He, "Robust structured nonnegative matrix factorization for image representation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 5, pp. 1947–1960, 2018.
- [154] S. Zhang, L. Yao, L. V. Tran, A. Zhang, and Y. Tay, "Quaternion collaborative filtering for recommendation," in *Proceedings of IJCAI*, pp. 4313–4319, 2019.
- [155] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Light-gcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of SIGIR*, pp. 639–648, 2020.
- [156] A. H. B. Eissa, M. E. El-Sharkawi, and H. M. O. Mokhtar, "Towards recommendation using interest-based communities in attributed social networks," in *Proceedings of WWW*, pp. 1235–1242, 2018.
- [157] L. Xu, X. Wei, J. Cao, and P. S. Yu, "On learning mixed community-specific similarity metrics for cold-start link prediction," in *Proceedings of WWW*, pp. 861–862, 2017.
- [158] A. De, S. Bhattacharya, S. Sarkar, N. Ganguly, and S. Chakrabarti, "Discriminative link prediction using local, community, and global signals," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2057–2070, 2016.
- [159] H. Wan, Y. Zhang, J. Zhang, and J. Tang, "Aminer: Search and mining of academic social networks," *Data Intell.*, vol. 1, no. 1, pp. 58–76, 2019.
- [160] D. Jin, H. Wang, J. Dang, D. He, and W. Zhang, "Detect overlapping communities via ranking node popularities," in *Proceedings of AAAI*, pp. 172–178, 2016.
- [161] Y. Wang, D. Jin, K. Musial, and J. Dang, "Community detection in social networks considering topic correlations," in *proceedings of AAAI*, pp. 321–328, 2019.
- [162] H. Cai, V. W. Zheng, F. Zhu, K. C. Chang, and Z. Huang, "From community detection to community profiling," *Proc. VLDB Endow.*, vol. 10, no. 7, pp. 817–828, 2017.
- [163] J. Shao, Z. Zhang, Z. Yu, J. Wang, Y. Zhao, and Q. Yang, "Community detection and link prediction via cluster-driven low-rank matrix completion," in *Proceedings of IJCAI*, pp. 3382–3388, 2019.
- [164] Y. Li, C. Sha, X. Huang, and Y. Zhang, "Community detection in attributed graphs: An embedding approach," in *Proceedings of AAAI*, pp. 338–345, 2018.
- [165] X. Li, Y. Wu, M. Ester, B. Kao, X. Wang, and Y. Zheng, "Semi-supervised clustering in attributed heterogeneous information networks," in *Proceedings of WWW*, pp. 1621–1629, 2017.
- [166] D. J. DiTursi, G. Ghosh, and P. Bogdanov, "Local community detection in dynamic networks," in *Proceedings of ICDM*, pp. 847–852, 2017.
- [167] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He, "FASCINATE: fast cross-layer dependency inference on multi-layered networks," in *Proceedings of SIGKDD*, pp. 765–774, 2016.
- [168] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li, "Community detection in incomplete information networks," in *Proceedings of WWW*, pp. 341–350, 2012.

APPENDIX A

Here we list the key terms and notations in the main text in Table 1.

TABLE 6: Summary of notations.

Notations	Descriptions
G	A network.
V, E	The sets of nodes and edges of a network.
A, X	The adjacency matrix and node attribute matrix.
D	The node degree matrix.
n, m	The numbers of nodes and edges.
e_{ij}	The edge between nodes v_i and v_j .
a_{ij}	The connection between nodes v_i and v_j .
x_i	The attribute vector and degree of node v_i .
q	The maximal number of node attributes.
C	The set of communities.
C	The community assignments of nodes.
k	The numbers of communities.
c_i	The community which node v_i belongs to.
ω_r	The probability of nodes assigned to community C_r .
π_{rs}	The probability of link generation within two communities C_r and C_s .
$\delta(c_i, c_j)$	The probability of nodes v_i and v_j falling into the same community partition.
$E(C; A)$	The energy function in MRF.
Θ_i	The unary potential function in $E(C; A)$.
Θ_{ij}	The pairwise potential function in $E(C; A)$.
$KL(\cdot \cdot)$	KL-divergence.
\tilde{A}	The estimation matrix of A in NMF.
B, S	Community membership matrix and attribute community matrix in NMF.
H, W	Node representation matrix and weight matrix of neural networks.
M, L	Modularity matrix and Laplacian matrix.
\mathcal{F}	The set of factor nodes.
\hat{A}, \hat{X}	Reconstructed adjacency matrix and node attribute matrix.
\mathcal{G}, \mathcal{D}	The generator and discriminator of GAN.
\mathcal{E}	The encoder that derives node representation.

APPENDIX B

In Section 3.1.1, we have introduced several SBM variants for community detection. Here, we give the overall process of community detection based on the basic SBM with a Bernoulli distribution [28], MMSB [29] and DSBM [56] in algorithm 1, algorithm 2 and algorithm 3 respectively.

APPENDIX C

In Section 3.1.2, we have presented several topic models for community detection. Here, we give the generation process of SSN-LDA [69] for one social interaction profile in algorithm 4, and provide the process for clustering attribute communities [72] in algorithm 5.

Algorithm 1: The basic SBM-based method [28]

Input: n, k .

Output: the community assignments of nodes C .

- 1 Assume that nodes are independently divided into k communities;
 - 2 Inference the parameters ω, π of likelihood function by using EM algorithm;
 - 3 **for each node** v_i **do**
 - 4 **for each node** v_j **do**
 - 5 $c_i \stackrel{i.i.d}{\sim} \text{Multinomial}(1; \omega)$;
 - 6 $a_{ij} | c_{ir}, c_{js} \stackrel{i.i.d}{\sim} \text{Bernoulli}(\pi_{rs}) | 0 < r, s \leq k$;
 - 7 **return** C ;
-

Algorithm 2: The MMSB-based method [29]

Input: n, k, α, β .

Output: the community assignments of nodes C .

- 1 **for each node** v_i **do**
 - 2 $\omega_i \sim \text{Dirichlet}(\alpha)$;
 - 3 Assigns the community assignments c_i with ω_i ;
 - 4 **for each node** v_j **do**
 - 5 $\pi_{rs} | c_{ir}, c_{js} \sim \text{Beta}(\beta) | 0 < r, s \leq k$;
 - 6 $a_{i \rightarrow j} \sim \text{Multinomial}(\omega_i)$;
 - 7 $a_{i \leftarrow j} \sim \text{Multinomial}(\omega_j)$;
 - 8 $a_{ij} \sim \text{Bernoulli}(a_{i \rightarrow j}^T \pi_{rs} a_{i \leftarrow j})$;
 - 9 **return** C ;
-

Algorithm 3: The DSBM-based method [56]

Input: n, π, A, T .

Output: the community assignments of nodes $C^{(T)}$.

- 1 **if time** $t == 1$ **then**
 - 2 generate the social network followed by SBM;
 - 3 **for each time** $t > 1$ **do**
 - 4 generate $c_i^{(t)} \sim \pi(c_i^{(t)} | c_i^{(t-1)}, A) | 0 < i \leq n$;
 - 5 **for each pair of nodes** (v_i, v_j) **at time** t **do**
 - 6 generate
 - 7 $w_{ij}^{(t)} \sim \text{Bernoulli}(\cdot | \pi_{c_i^{(t)}, c_j^{(t-1)}}) | 0 < i, j \leq n$;
 - 7 **return** $C^{(T)}$;
-

Algorithm 4: The SSN-LDA method [69]

Input: $k, \vec{\alpha}, \vec{\beta}, \varepsilon$.

Output: the community assignment of one node c_i .

- 1 sample mixture components $\vec{\phi} \sim \text{Dirichlet}(\vec{\beta})$;
 - 2 choose $\vec{\theta}_i \sim \text{Dirichlet}(\vec{\alpha})$;
 - 3 choose $N_i \sim \text{Poisson}(\varepsilon)$;
 - 4 **for each neighbor** v_j **of** v_i **do**
 - 5 choose a community $c_i \sim \text{Multinomial}(\vec{\theta})$;
 - 6 choose a social interaction
 - 7 $a_{ij} \sim \text{Multinomial}(\vec{\phi}_{c_i})$;
 - 7 **return** c_i ;
-

Algorithm 5: The generation of Bayesian attributed graph clustering (BAGC) [72]

Input: n, k, T, \mathcal{C} , a time attributes set
 $\Lambda = \{\lambda^{(1)}, \dots, \lambda^{(T)}\}$, parameters ε, μ, ν .

Output: the community assignments of nodes C ,
attribute matrix X .

```

1 choose  $\alpha \sim \text{Dirichlet}(\varepsilon)$ ;
2 for  $\mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$  do
3   for each attribute  $\lambda^{(t)}$  do
4     choose  $\theta_i^{(t)} \sim \text{Dirichlet}(\lambda^{(t)})$ ;
5   for each community  $\mathcal{C}_j \in \{\mathcal{C}_i, \mathcal{C}_{i+1}, \dots, \mathcal{C}_k\}$  do
6     choose  $\phi_{ij} \sim \text{Beta}(\mu, \nu)$ ;
7 for each node  $v_i$  do
8   choose  $c_i \sim \text{Multinomial}(\alpha)$ ;
9   for each attribute  $\lambda^{(t)}$  do
10    choose  $x_i^{(t)} \sim \text{Multinomial}(\theta_{c_i}^{(t)})$ ;
11   for each node  $v_j$  with  $i > j$  do
12    choose  $a_{ij} \sim \text{Bernoulli}(\phi_{c_i c_j}) | 0 < i, j \leq n$ ;
13 return  $C, X$ ;
```
