Journal of Machine Learning Research 18 (2018) 1-48 Submitted 7/17; Revised 12/17; Published 4/18

Significance-based community detection in weighted networks

John Palowitch* PALOJJ@EMAIL.UNC.EDU
Shankar Bhamidi BHAMIDI@EMAIL.UNC.EDU
Andrew B. Nobel NOBEL@EMAIL.UNC.EDU
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599
**Now at Google*

Editor: Michael Mahoney

Abstract

Community detection is the process of grouping strongly connected nodes in a network. Many community detection methods for *un*-weighted networks have a theoretical basis in a null model. Communities discovered by these methods therefore have interpretations in terms of statistical significance. In this paper, we introduce a null for weighted networks called the continuous configuration model. First, we propose a community extraction algorithm for weighted networks which incorporates iterative hypothesis testing under the null. We prove a central limit theorem for edge-weight sums and asymptotic consistency of the algorithm under a weighted stochastic block model. We then incorporate the algorithm in a community detection method called CCME. To benchmark the method, we provide a simulation framework involving the null to plant "background" nodes in weighted networks with communities. We show that the empirical performance of CCME on these simulations is competitive with existing methods, particularly when overlapping communities and background nodes are present. To further validate the method, we present two real-world networks with potential background nodes and analyze them with CCME, yielding results that reveal macro-features of the corresponding systems.

Keywords: Community detection; Multiple testing; Network models; Weighted networks; Unsupervised learning

1. Introduction

For decades, the development of graph theory and network science has produced a wide array of quantitative tools for the study of complex systems. Network-based data analysis methods have driven advances in areas as diverse as social science, systems biology, life sciences, marketing, and computer science (cf. Palla et al., 2007; Barabasi and Oltvai, 2004; Lusseau and Newman, 2004; Guimera and Amaral, 2005; Reichardt and Bornholdt, 2007; Andersen et al., 2012). Thorough surveys of the network science and methodology literature have been provided by Newman (2003) and Jacobs and Clauset (2014), among others.

A common exploratory technique for networks called "community detection" involves the search for node subsets ("communities") that are somehow internally related. Usually, it is assumed that links or “edges” between nodes in the same community are more common than those between nodes in different communities. Thus, the specific goal of community detection is often to find groups of nodes that are strongly intraconnected and weakly interconnected (Newman, 2004b), which is known as “assortative” structure. Some approaches allow for “disassortative” structure, with *weak* intraconnection and *strong* interconnection (Karrer and Newman, 2011; Aicher et al., 2014). Regardless of one’s broad notion of community structure, there are hundreds of distinct criteria and methods for its discovery. Community detection most often proceeds in an *unsupervised* fashion, so that communities are found through the network edge structure alone.

Since communities in real-world networks often arise due to some explanatory systematic mechanism, community detection has been an important part of both data-driven decision-making and scientific inquiry (Danon et al., 2005). For instance, community detection has been used to facilitate recommender systems in online social networks (e.g. Sahebi and Cohen, 2011; Xin et al., 2014), and has been used to “hone in” on regions of genomes (human and otherwise) for a variety of downstream analyses (e.g. Cabreros et al., 2016; Platig et al., 2015; Fan et al., 2012). Myriad examples of community detection applications can be found in Porter et al. (2009) and Fortunato (2010), and the references therein.

Many community detection methods are based on a null model, which in this context means a random network model without explicit community structure. For un-weighted networks the most common null is the configuration model (Bollobás, 1980; Bender, 1974) or a related model like that of Chung and Lu (2002a,b). Historically, the most common approach involving a null model is the use of a node partition score that is large when nodes within the cells of the partition are highly interconnected, relative to what is expected under the null (Fortunato, 2010; Newman, 2006). Arguably the most famous example of such a criterion is modularity, introduced by Newman and Girvan (2004). Various algorithms have been created to search directly for partitions of a network with large modularity (see Clauset et al., 2004; Blondel et al., 2008), while other approaches use modularity as an auxiliary criterion (see Langone et al., 2011). More recent approaches incorporate community-specific criteria which are large when the community is preferentially intra-connected, allowing for community *extraction* algorithms (e.g. Zhao et al., 2011; Lancichinetti et al., 2011; Wilson et al., 2014).

Generally speaking, communities found by null-based community detection methods can be said to have exhibited behavior strongly departing from the null. The results of these methods therefore carry a statistical *testing* interpretation unavailable to most methods without this foundation, like spectral clustering (White and Smyth, 2005; Zhang et al., 2007) or those based on random walks (e.g. Pons and Latapy, 2006). In particular, recent methods put forth by Lancichinetti et al. (2011) and Wilson et al. (2014) for binary networks use the theoretical properties of the configuration model to detect “background” nodes that are not significantly connected to any community. These methods incorporate tail behavior of node-wise graph statistics under the configuration model in a way that others do not. Note that another class of methods capable of assessing significance are focused on specifying *alternative* (non-null) models for communities. Once fit, the models can be compared with the null-partition via likelihood ratios (e.g. Yan et al., 2014) or Bayes factors (e.g. Peixoto, 2015).

A significant drawback of null-based community detection methodology is that no explicit null model exists for edge-weighted networks. Edge weights are commonplace in network data, and can provide information that improves community detection power and specificity (Newman, 2004a). While many existing community detection methods have been established for weighted and un-weighted networks alike, due to the absence of an appropriate weighted-network null model, these methods do not provide rigorous significance assessments of weighted-network communities. For instance, the aforementioned method from Lancichinetti et al. (2011), called OSLOM, can incorporate edge weights, but uses an exponential function to calculate nominal tail probabilities for edge weight sums, a testing approach which is not based on an explicit null. As a consequence, communities in *weighted* networks identified by OSLOM may in some cases be spurious or unreliable, especially when no "true" communities exist.

The key methodological contributions in this article are as follows: (i) we provide an explicit null model for networks with weighted edges, (ii) we present a community extraction method based on hypothesis tests under the null, and (iii) we analyze the consistency properties of the method's core algorithm with respect to a weighted stochastic block model. These contributions provide the beginnings of a rigorous statistical framework with which to study communities in weighted networks. Through extensive simulations, we show that the accuracy of our proposed extraction method is highly competitive with other community detection approaches on weighted networks with both disjoint and overlapping communities, and on weighted networks with background nodes. Importantly, the weighted stochastic block model employed (in both the theoretical and empirical studies) allows for arbitrary expected degree and weighted-degree distributions, reflecting degree heterogeneity observed in real-world networks. To further validate the method, we apply it to two real data sets with (arguably) potential overlapping communities and background nodes. We show that the proposed method recovers sensible features of the real data, in contrast to other methods.

1.1 Paper organization

The rest of the paper is organized as follows. We start by introducing general notation in Section 1.2. In Section 2 we motivate and state the continuous configuration model. In Section 3, we introduce a core algorithm to search for communities using multiple hypothesis testing under the model. In Section 4, we prove both a central limit theorem and a consistency result for the primary test statistic in the core algorithm. We describe the implementation and application of the core algorithm in Section 5, and evaluate its empirical efficacy on simulations and real data in Section 6 and 7 (respectively). We close with a discussion in Section 8.

1.2 Notation and terminology

We denote an undirected weighted network on n nodes by a triple $\mathcal{G} := (N, A, W)$, where $N := \{1, \ldots, n\}$ is the node set with u, v as general elements, A is the adjacency matrix with $A_{uv} = 1$ if and only if there is an edge between u and v, and W is the weight matrix with non-negative entries W_{uv} containing edge weights between nodes u and v. Note that $A_{uv} = 0$ implies $W_{uv} = 0$, but W_{uv} may be zero even when $A_{uv} = 1$. This allows for networks with potentially zero edge weights; for instance, an online social network from which friendship links are edges and message counts are edge weights. The degree of a node u is defined by $d(u) := \sum_{v \in N} A_{uv}$, and we denote the vector of node degrees by $\mathbf{d} = (d_1, \ldots, d_n)$. In an analogous fashion, we define the *strength* of a node by $s(u) := \sum_{v \in N} W_{uv}$, and the strength vector of the network by $\mathbf{s} = (s(1), \ldots, s(n))$. The total degree and strength of $\mathcal{G}$ are given by $d_T := \sum_{v \in N} d(v)$ and $s_T := \sum_{v \in N} s(v)$, respectively.

2. The continuous configuration model

To motivate the null model, we first explain the intuition behind the binary configuration model for unweighted networks. The binary configuration model for an n-node network is based on a given degree vector $\mathbf{d}$ corresponding to the nodes. Studied originally in Bollobás (1980) and Bender (1974), the model is equivalent to a process in which each node u receives $d(u)$ half-edges, which are paired uniformly-at-random without replacement until no half-edges remain (Molloy and Reed, 1995). In other words, the model guarantees a graph with degrees $\mathbf{d}$ but otherwise uniformly distributed edges. Therefore, given an observed network with degrees $\mathbf{d}$, a typical draw from the configuration model under $\mathbf{d}$ represents that network without any community structure.

Due to these characteristics, many community detection methods proceed by identifying node sets having intra-connectivity significantly beyond what is expected under the model. For instance, the modularity measure, introduced by Newman and Girvan (2004), scores node partitions of binary networks according to the observed versus configuration model-expected edge densities of the communities. The methods OSLOM (Lancichinetti et al., 2011) and ESSC (Wilson et al., 2014) use the configuration model to assess the statistical significance of the deviations graph statistics from their configuration model-expected values. Methods based on the configuration model are able to properly distinguish true community structure from "false positive" community structure arising from inherently high connectivity. For instance, Karrer and Newman (2011) showed that on a political blogs network, modularity maximization found communities that agreed with political affiliation, whereas the same algorithm without node degree input found a naíve split of the network into more active and less active blogs.

The degrees $\mathbf{d}$ of the configuration model can be thought of as the nodes' relative propensities to form ties. Chung and Lu made this notion explicit by defining a Bernoulli-based model for a n-node unweighted network with a given expected degree sequence (Chung and Lu, 2002b). Under this model, the probability of nodes u and v sharing an edge is exactly $d(u)d(v)/d_T$. As null models for community detection, the Chung-Lu and configuration are often interchangeable (Durak et al., 2013). Indeed, for sparse graphs it can be shown that the probability of an edge between u and v under the configuration model is approximately the Chung-Lu probability. The *continuous* configuration model, introduced below, extends the spirit of the configuration and Chung-Lu models by taking both observed degrees $\mathbf{d}$ and strengths $\mathbf{s}$ as node propensities for (respectively) edge connection and edge weight.

In the next section, we use the following notation to concisely express the model. Given a vector $\mathbf{x}$ of dimension n, we define for any indices $u, v \in N$ the ratio

$$r_{uv}(\mathbf{x}) := \frac{x(u)x(v)}{\sum_{w \in N} x(w)} \tag{1}$$

Define $\tilde{r}_{uv}(\mathbf{x}) := \min\{1, r_{uv}(\mathbf{x})\}$. Note that when $\mathbf{x}$ is a degree sequence $\mathbf{d}$, $r_{uv}(\mathbf{d})$ is the Chung-Lu probability of an edge between nodes u and v. Finally, for a vector $\mathbf{y}$ of dimension n, define $f_{uv}(\mathbf{x}, \mathbf{y}) := r_{uv}(\mathbf{y})/\tilde{r}_{uv}(\mathbf{x})$.

2.1 Model statement

The continuous configuration model on n nodes has the parameter triple $\theta := (\mathbf{d}, \mathbf{s}, \kappa)$, where $\mathbf{d} \in \{1, 2, 3, \ldots\}^n$ is a degree vector, $\mathbf{s} \in [0, \infty)^n$ is a strength vector, and $\kappa > 0$ is a variance parameter. Let F be a distribution on the non-negative real line with mean one and variance κ. The model specifies a random weighted graph $\mathcal{G} := (N, A, W)$ on n nodes as follows:

1. $\mathbb{P}(A_{uv} = 1) = \tilde{r}_{uv}(\mathbf{d})$ independently for all node pairs $u, v \in N$
2. For each node pair u, v, generate an independent random variable ξ_{uv} according to F, and assign edge weights by: $W_{uv} = \begin{cases} f_{uv}(\mathbf{d}, \mathbf{s})\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 0 \end{cases}$

The edge generation defined by step 1 is equivalent to the Chung-Lu model: edge indicators are Bernoulli, with probabilities adjusted by the propensities $\mathbf{d}$. The weight generation in step 2 mirrors this process. Edge weights follow the distribution F, with means adjusted by the propensities $\mathbf{s}$, through $f(\mathbf{d}, \mathbf{s})$. Note that $\tilde{r}_{uv}(\mathbf{d})$ differs from $r_{uv}(\mathbf{d})$ only when the degree distribution is extremely right-skewed or highly dispersed, and then only when one of u or v has a large degree. Truncating edge probabilities is standard in approaches based on the Chung-Lu model (e.g. Zhao et al., 2012).

It is easily derived from the model that

$$P(A_{uv} = 1) = \frac{d(u)d(v)}{d_T} \quad \text{and} \quad \mathbb{E}\left(W_{uv}\right) = \frac{s(u)s(v)}{s_T}, \tag{2}$$

equations which extend the binary-network notion of null behavior to edge weights. (The first equation will be approximate for any u, v for which $r_{uv}(\mathbf{d}) > 1$.) The equations in (2) imply that

$$\mathbb{E}\left(D(u)\right) = d(u) \quad \text{and} \quad \mathbb{E}\left(S(u)\right) = s(u) \;\; \text{for all} \;\; u \in N. \tag{3}$$

where $D(u)$ and $S(u)$ are the (random) degree and strength of u under the model. Thus, the continuous configuration model can be thought of as null weighted network with given expected degrees and given expected strengths.

2.2 Null model parameter specification

When the *binary* configuration model is used for community detection, the degree parameter of the model is set to the observed degree distribution of the network. In a sense, this is an *estimation* of the nodes' connection propensities under the null. Similarly, to use the continuous configuration model in practice, we derive the parameter θ from the data at hand. Given an observed network $\mathcal{G}$, we straightforwardly use the observed degrees and strengths $\mathbf{d}$ and $\mathbf{s}$ as the first two parameters of the model. The third parameter of the continuous configuration model, κ, is also computed from the $\mathcal{G}$, and meant to capture its observed average edge-weight variance. We use the following method-of-moments estimator to specify κ:

$$\hat{\kappa}(\mathbf{d}, \mathbf{s}) := \sum_{u,v:A_{uv}=1} (W_{uv} - f_{uv}(\mathbf{d}, \mathbf{s}))^2 \,/ \sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{d}, \mathbf{s})^2 \tag{4}$$

This estimator is derived as follows. Under the continuous configuration model with $\mathbf{d}$ and $\mathbf{s}$,

$$\operatorname{Var}\left(W_{uv} \,\middle|\, A_{uv} = 1\right) \;=\; f_{uv}(\mathbf{d}, \mathbf{s})^2 \operatorname{Var}(\xi_{uv}) \;=\; f_{uv}(\mathbf{d}, \mathbf{s})^2 \,\kappa. \tag{5}$$

Therefore

$$\begin{aligned} \mathbb{E}\left\{ \sum_{u,v:A_{uv}=1} (W_{uv} - f_{uv}(\mathbf{d}, \mathbf{s}))^2 \,\middle|\, A \right\} &= \sum_{u,v:A_{uv}=1} \operatorname{Var}\left(W_{uv} \,\middle|\, A_{uv} = 1\right) \\ &= \kappa \sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{d}, \mathbf{s})^2, \end{aligned}$$

Dividing through by $\sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{d}, \mathbf{s})$ motivates equation 4.

Strictly speaking, the distribution F is also a parameter of the model. However, for testing purposes we do not require a null specification of F. As we discuss in the next section, p-values from the model will be based on a central limit theorem that requires only a third-moment assumption on F. While estimating F could improve the model's efficacy as a null, in general this would require potentially costly computational procedures, and additional theoretical assumptions that might be difficult to support or verify in practice. The specification of F will be most useful for applications of the model that involve simulations or likelihood-based analyses.

In subsequent sections, we describe a procedure for testing nodal connectivity against the continuous configuration model, with parameters derived from the real data (as above). These parameters pertain only to a *null* model, and not to any particular community-laden generative network model. Our procedure (to be introduced) is built upon hypothesis testing under this estimated null. Note that this is in contrast to a broad class of community detection methods which are all based on estimating a "Stochastic Block Model" (SBM), a generative network model with parameters controlling the presence and strength of communities (e.g. Nowicki and Snijders, 2001; Karrer and Newman, 2011; Aicher et al., 2014; Peixoto, 2017). In particular, the work of Peixoto (2017) introduces a weighted-edge SBM estimation procedure which automatically chooses the number of communities, and allows for a notion of community overlap. We feature this method in our simulation experiments, introduced in Section 6.

3. Test statistic and update algorithm

In this section we introduce a core testing-based community detection algorithm based on the continuous configuration model. The algorithm *updates* an input node set via node-to-set hypothesis testing. We describe the repeated application of the algorithm as part of an iterative set-update approach to community detection, following some recently-introduced methods (e.g. Lancichinetti et al., 2011; Wilson et al., 2014).

Formally, we define the core algorithm operation as a set-update map $U_\alpha(\,\cdot\,, \mathcal{G}) : \mathbb{2}^N \mapsto \mathbb{2}^N$, indexed by a parameter $\alpha \in (0,1)$. Given a weighted network $\mathcal{G}$ and candidate set $B \subseteq N$, the update $U_\alpha(B, \mathcal{G})$ outputs a new set B' formed by the nodes from N that have statistically significant association to B at level α, after a multiple-testing correction. The node-wise hypothesis tests are based on the summary statistic

$$S(u, B, \mathcal{G}) := \sum_{v \in B} W_{uv}, \tag{6}$$

which is the sum of all weights on edges incident with u and B. When the observed value of $S(u, B, \mathcal{G})$ is much larger than its expectation under the continuous configuration model, there is evidence to support an association between u and B resulting from some form of "ground-truth" community structure in the network. We assess the strength of evidence, that is, the significance of $S(u, B, \mathcal{G})$, with the p-value

$$p(u, B, \mathcal{G}) := \mathbb{P}\left(S(u, B, \widetilde{\mathcal{G}}) > S(u, B, \mathcal{G})\right), \tag{7}$$

where $\widetilde{\mathcal{G}}$ is random with respect to $\mathbb{P}$, the distribution of the continuous configuration model with parameters $\mathbf{d}, \mathbf{s}$, and $\hat{\kappa}(\mathbf{d}, \mathbf{s})$ (see Section 2.2). The update U_α is then:

Core update U_α

1. Given: graph $\mathcal{G}$ with nodes N and input set $B \subseteq N$
2. Calculate p-values $\mathbf{p} := \{p(u, B, \mathcal{G}) : u \in N\}$
3. Obtain threshold $\tau(\mathbf{p})$ from a multiple-testing procedure
4. Output set $B' = \{u : p(u, B, \mathcal{G}) \leqslant \tau(\mathbf{p})\}$

Many methods to compute a multiple-testing threshold $\tau(\mathbf{p})$ are available, the most stringent being the well-known Bonferroni correction. The correction we employ is the false discovery rate (FDR) control procedure of Benjamini and Hochberg (1995). Given a set of p-values $\mathbf{p} := \{p_u\}_{u \in N}$ corresponding to n hypothesis tests and a target FDR $\alpha \in (0,1)$, each p-value $p_u \in \mathbf{p}$ is associated with an adjusted p-value $p_u^* := n\, p_u \,/\, j(u)$ where $j(u)$ is the rank of p_u in $\mathbf{p}$, and $\tau(\mathbf{p}) := \max\{p_u : p_u^* \leqslant \alpha\}$. Benjamini and Hochberg show that, if the p-values corresponding to true null hypotheses are independent, the threshold $\tau(\mathbf{p})$ bounds the expected number of false discoveries at α.

The update U_α is an exploratory tool for moving an input set B closer to a "target" community. Consider that, if the initial set B has a majority group of nodes from some strongly-connected community C, the statistic $S(u, B, \mathcal{G})$ will be large for $u \in C$, and small otherwise. In this case, U_α applied to B will often return many nodes in C, and few nodes in C^c. Note that U_α can *remove* nodes from B, if they are not significantly connected, as well as add them.

If C is a community with strong enough signal, we should expect $U_\alpha(C, \mathbf{D})$ to return C. This reasoning motivates an algorithm that searches for "stable communities" C satisfying $U_\alpha(C, \mathbf{D}) = C$. By definition, all interior nodes of a stable community C are significantly connected to C, and exterior nodes are not. We define a stable community search procedure, which iteratively applies U_α until convergence:

Stable community search (SCS) algorithm

```
1. Given weighted graph $\mathcal{G}$ with nodes $N$ and initial set $B_0 \subseteq N$; $t \leftarrow 0$
2. Set $B_{t+1} \leftarrow U_\alpha(B_t, \mathcal{G})$ and $t \leftarrow t + 1$.
3. If $B_t = B_{t'}$ for some $t' < t$, terminate. Else, return to step 2.
```

Since the number of possible node subsets B_t is finite, SCS is guaranteed to terminate. There are some technicalities regarding use of this algorithm, like how to obtain B_0, and when in rare cases $t' < t - 1$. We relegate resolution of these issues to Section 5. For now, the update U_α and SCS raise two theoretical questions:

1. Is the p-value $p(u, B, \mathcal{G})$ analytically tractable? If not, is there a useful distributional approximation based on the continuous configuration model?

2. Consistency: with what power can SCS detect ground-truth community structure?

These questions are the focus of the next section.

4. Theoretical Results

We now address the theoretical questions raised at the end of the previous section by analyzing the distribution of the test statistic $S(u, B, \mathcal{G})$ under the continuous configuration model (for question 1) and an appropriate alternative model with planted community structure (for question 2). Both analyses have an asymptotic setting consisting of a sequence of random weighted networks. Denote this sequence by $\{\mathcal{G}_n\}_{n>1}$. If $\mathcal{G}_n$ is a continuous configuration model with parameters $\theta := (\mathbf{s}, \mathbf{d}, \kappa)$, the following proposition gives general expressions for the mean and standard deviation of $S(u, B, \mathcal{G}_n)$:

Proposition 1 *Let $\mathcal{G} = (N, A, W)$ be a random network generated by the continuous configuration model with parameters $\theta = (\mathbf{s}, \mathbf{d}, \kappa)$. For any $(u, B) \in N \times 2^N$, let $\mu(u, B|\theta)$ and $\sigma(u, B|\theta)$ be, respectively, the mean and standard deviation of $S(u, B, \mathcal{G})$ under $\mathcal{G}$. Then*

$$\mu(u, B|\theta) \equiv \mu(u, B|\mathbf{s}) = \sum_{v \in B} r_{uv}(\mathbf{s}) \tag{8}$$

and

$$\sigma(u, B|\theta)^2 = \sum_{v \in B} r_{uv}(\mathbf{s}) f_{uv}(\mathbf{d}, \mathbf{s}) \left(1 - \tilde{r}_{uv}(\mathbf{d}) + \kappa\right) \tag{9}$$

The proof, given in Appendix A, follows from easy calculations with the model's generating procedure (see Section 2.1). All theoretical results will make use of the expressions defined in equations 8 and 9.

4.1 Asymptotic Normality of $S(u, B, \mathcal{G})$

A central limit theorem under the null model is now established for $S(u, B, \mathcal{G})$, yielding a closed-form approximation for the p-value in equation (7). This result is motivated by the fact that, under most non-trivial null parameter specifications, the distribution of $S(u, B, \mathcal{G})$ is not analytically tractable.

In the setting of the theorem, for any $n > 1$, a random network $\mathcal{G}_n$ is generated by a continuous configuration model with parameter $\theta_n := (\mathbf{d}_n, \mathbf{s}_n, \kappa_n)$ and common weight distribution F. The following regularity conditions are required on the sequence $\{\theta_n\}_{n>1}$. Let λ_n denote the average entry of $\mathbf{d}_n$, (which is the average expected degree of $\mathcal{G}_n$). For each $r \geqslant 0$ let $L_{n,r} := n^{-1} \sum_{u \in N} (d_n(u)/\lambda_n)^r$ be the normalized r^{th}-moment of $\mathbf{d}_n$. Note that $L_{n,1} = 1$. The regularity conditions are then as follows:

Assumption 1 *Define $e_n(u|\beta) := s_n(u)/d_n(u)^{1+\beta}$. There exists $\beta > 0$ such that*

$$0 < \liminf_{n\to\infty} \min_{u\in N} e_n(u|\beta) \quad \text{and} \quad \limsup_{n\to\infty} \max_{u\in N} e_n(u|\beta) < \infty.$$

Assumption 2 *Let β be as in Assumption 1. There exists $\varepsilon > 0$ such that, for both $r = 4\beta + 2$ and $r = 4\beta + 2 + \varepsilon$,*

$$0 < \liminf_{n\to\infty} L_{n,r} \quad \text{and} \quad \limsup_{n\to\infty} L_{n,r} < \infty$$

Assumption 3 $\limsup_{n\to\infty} \sup_{u,v\in N} r_{uv}(\mathbf{d}_n) < \infty.$

Assumption 4 *The sequence $\{\kappa_n\}_{n\geqslant 1}$ is bounded away from zero and infinity, and F has finite third moment.*

Assumption 1 reflects the common relationship between strengths and degrees in real-world weighted networks (Barrat et al., 2004; Clauset et al., 2009). Assumptions 2-3 are needed to control the extremal behavior of the degree distribution. They exclude, for instance, cases with a few nodes having $d_n(u) \asymp n$ and the remaining nodes having $d_n(u) = O(1)$. We note that the Assumption 2 becomes more stringent as β increases, since as β increases the strength-degree power law becomes more severe.

Theorem 2 *For each $n > 1$, let $\mathcal{G}_n$ be generated by the continuous configuration model with parameter θ_n and weight distribution F. Suppose $\{\theta_n\}_{n\geqslant 1}$ and F satisfy Assumptions 1-4. Fix a node sequence $\{u_n\}_{n\geqslant 1}$ with $u_n \in N$ and a positive integer sequence $\{b_n\}_{n\geqslant 1}$ with $b_n \leqslant n$. Suppose $d_n(u_n)b_n/n \to \infty$ as $n \to \infty$. Let $B_n \subseteq N$ be a node set chosen independently of $\mathcal{G}_n$ according to the uniform distribution on all sets of size b_n. Then*

$$\frac{S(u_n, B_n, \mathcal{G}_n) - \mu_n(u_n, B_n|\theta_n)}{\sigma_n(u_n, B_n|\theta_n)} \Rightarrow \mathcal{N}(0,1) \quad \text{as} \quad n \to \infty \tag{10}$$

The proof is given in Appendix B. Essentially, Theorem 2 says that $S(u, B, \mathcal{G})$ is asymptotically Normal provided that B is typical (i.e. a uniform draw) and that $d(u)$ and B are sufficiently large. The theorem justifies the following approximation of the p-value in (7):

$$p(u, B) \approx 1 - \Phi\left(\frac{S(u, B, \mathcal{G}) - \mu(u, B|\theta)}{\sigma(u, B|\theta)}\right) \tag{11}$$

Above, $\theta = (\mathbf{d}, \mathbf{s}, \hat{\kappa}(\mathbf{d}, \mathbf{s}))$ is computed from $\mathcal{G}$, as described in Section 2.2. Note that the numerator $S(u, B, \mathcal{G}) - \mu(u, B|\theta)$ is the contribution of B to the standard modularity score, as will be discussed in more detail in Sections 4.2.2 and 4.2.3. The variance $\sigma(u, B|\theta)$ thus standardizes the magnitude of B's modularity, providing an appropriate test statistic for community effect.

4.2 Consistency of SCS

In this section, we evaluate the ability of the SCS algorithm to identify true communities in a planted-community model. Explicitly, we consider a sequence of networks $\{\mathcal{G}_n\}_{n>1}$ where each network in the sequence is generated by a weighted stochastic block model (WSBM). The WSBM we employ is similar to that presented in Aicher et al. (2014), but is generalized to include node-specific weight parameters. In other words, it is "strength-corrected" as well as degree-corrected, in a manner analogous to the original degree-corrected SBM (Coja-Oghlan and Lanka, 2009). The proofs of Theorem 4 and Theorem 5 are given in Appendix C.

4.2.1 The weighted stochastic block model

For fixed $K > 1$, we define a K-block WSBM on $n > 1$ nodes as follows. Let $\mathbf{c}_n$ be a community partition vector with $c_n(u) \in \{1, \ldots, K\}$ giving the community index of u. Denote community i by $C_{i,n} := \{u : c_n(u) = i\}$. Define $\pi_{i,n} := n^{-1}|C_{i,n}|$ with $\boldsymbol{\pi}_n$ the associated vector. Let $\mathbf{P}$ and $\mathbf{M}$ be fixed $K \times K$ symmetric matrices with non-negative entries encoding intra- and inter-community baseline edge probabilities and edge weight expectations, respectively. Let ϕ_n and ψ_n be arbitrary n-vectors with positive entries, which are parameters giving nodes individual propensities to form edges and assign weight (separately from $\mathbf{P}$ and $\mathbf{M}$). To ensure proper edge probabilities, we assume that $\max(\phi_n)^2 \max(\mathbf{P}) \leqslant 1$. For identifiability, we assume the vectors ϕ_n and ψ_n sum to n. Finally, let F be a distribution on the non-negative real line with mean 1 and variance $\sigma^2 \geqslant 0$. The WSBM can then be specified as follows:

1. Place an edge between nodes u and v with probability $\mathbb{P}_n(A_{uv} = 1) = r_{uv}(\phi_n)\mathbf{P}_{c_n(u)c_n(v)}$, independently across node pairs.

2. For node pair u, v with $A_{uv} = 1$, generate an independent random variable ξ_{uv} according to F. Determine edge weight W_{uv} by:

$$W_{uv} = \begin{cases} f_{uv}(\psi_n, \phi_n)\mathbf{M}_{c_n(u)c_n(v)}\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 0 \end{cases}$$

The many parameters involved with this model allow for node heterogeneity and community structure. When $\mathbf{P}$ and $\mathbf{M}$ are proportional to a $K \times K$ matrix of ones, the WSBM reduces to the continuous configuration model with parameters $\mathbf{d} \propto \phi$, $\mathbf{s} \propto \psi$, and $\kappa = \sigma^2$. Community structure is introduced in the network by allowing the diagonal entries of $\mathbf{P}$ and $\mathbf{M}$ to be arbitrarily larger than the off-diagonals.

4.2.2 Consistency theorem

The consistency analysis of SCS involves a sequence of random networks $\{\mathcal{G}_n\}_{n>1}$, where $\mathcal{G}_n$ is generated by a K-community WSBM. In this setting, we incorporate an additional parameter ρ_n, and let $\mathbf{P}_n := \rho_n \mathbf{P}$ replace $\mathbf{P}$ for each $n > 1$. This lets us distinguish the role of the asymptotic order of the average expected degree, defined $\lambda_n := n\rho_n$, from the profile of edge densities within and between communities ($\mathbf{P}$). Importantly, our results require only that $\lambda_n / \log n \to \infty$, reflecting the sparsity of real-world networks. Throughout this section, we denote the vector of (random) strengths from $\mathcal{G}_n$ by $\mathbf{S}_n$.

We now define an explicit notion of consistency in terms of the SCS algorithm. Recall from Section 3 that for fixed FDR $\alpha \in (0, 1)$, a stable community in a network $\mathcal{G}_n$ is defined as a node set $C \subseteq N$ satisfying $U_\alpha(C, \mathcal{G}_n) = C$.

Definition 3 *We say that SCS is consistent for a sequence of WSBM random networks* $\{\mathcal{G}_n\}_{n>1}$ *if for any FDR level* $\alpha \in (0, 1)$*, the probability that the true communities* $C_{1,n}, \ldots, C_{K,n}$ *are stable approaches 1 as* $n \to \infty$*.*

To assess the conditions that allow a target set C to be a stable community, we seek more general conditions under which the update $U_\alpha(\,\cdot\,, \mathcal{G})$ outputs C given any initial set B. If $U_\alpha(B, \mathcal{G}_n) = C$, all nodes $u \in C$ must have significant connectivity to B, as judged by the p-value approximation defined in 11. It is clear from that p-value expression that, for the update to return C, the test statistic $S(u, B, \mathcal{G}_n)$ must be significantly larger than $\mu(u, B|\mathbf{S}_n)$, its expected value under the continuous configuration model. Therefore, our first result hinges on asymptotic analysis of that deviation, which we denote by

$$A(u, B, \mathcal{G}_n) := S(u, B, \mathcal{G}_n) - \mu_n(u, B|\mathbf{S}_n). \tag{12}$$

The asymptotics of $A(u, B, \mathcal{G}_n)$ depend on its *population* version, in which all random quantities are replaced with their expected values under the WSBM. Let $\mathbf{s}_n$ be the expected value of $\mathbf{S}_n$ under $\mathcal{G}_n$. We define the (normalized) population version of $A(u, B, \mathcal{G}_n)$ by

$$\tilde{a}_n(u, B) := \lambda_n^{-1} \left(\mathbb{E}S(u, B, \mathcal{G}_n) - \mu_n(u, B|\bar{\mathbf{s}}_n) \right), \tag{13}$$

where λ_n is the order of the average expected degree. The value $\tilde{a}_n(u, B)$ is crucial to the primary condition of Theorem 4. Given a sequence of initial sets $\{B_n\}_{n>1}$ and target sets $\{C_n\}_{n>1}$, Theorem 4 establishes that $U_\alpha(B_n, \mathcal{G}_n) = C_n$ with probability approaching 1 if $\tilde{a}_n(u, B)$ is bounded away from zero, and is positive if and only if $u \in C_n$. The theorem requires the following two assumptions:

Assumption 5 *There exist constants* $m_+ > m_- > 0$ *such that, for all* $n > 1$*, the entries of* ϕ_n*,* ψ_n*,* $\mathbf{P}$*,* $\mathbf{M}$*, and* $\boldsymbol{\pi}_n$ *are all bounded in the interval* $[m_-, m_+]$*.*

Assumption 6 F *is independent of* n *and has support* $(0, \eta)$ *with* $\eta < \infty$*.*

Assumption 5 is standard in consistency analyses involving block models (e.g. Zhao et al., 2012; Bickel and Chen, 2009). We note that this assumption requires the minimum community size to scale with n. Assumption 6 allows the use of Bernstein's inequality throughout the proof, but may be relaxed if there are constraints on the moments of F allowing the use of a similar inequality. We now state Theorem 4, the proof of which is given in Appendix C.

Theorem 4 *Fix $K > 1$. For each $n > 1$, let $\mathcal{G}_n$ be a n-node random network generated by a K-community WSBM with parameters satisfying Assumptions 5 - 6. Suppose $\lambda_n / \log n \to \infty$. Let $\{B_n\}_{n>1}$, $\{C_n\}_{n>1}$ be sequences of node sets satisfying the following: there exist constants $q \in (0, 1]$ and $\Delta > 0$ such that for all n sufficiently large, $|B_n|, |C_n| \geqslant qn$, and*

$$\tilde{a}_n(u, B_n) \geqslant \Delta, \;\; u \in C_n, \quad \textit{and} \quad \tilde{a}_n(u, B_n) \leqslant -\Delta, \;\; u \notin C_n. \tag{14}$$

Then if the update U_α uses the p-value approximation given in Equation (11),

$$\mathbb{P}_n\big(U_\alpha(B_n, \mathcal{G}_n) = C_n\big) \to 1 \;\; \textit{as } n \to \infty.$$

To prove the consistency of SCS, we show that condition 14, when it involves the community sequence, is guaranteed by a concise condition on the model parameters. Let $\tilde{\pi}_{i,n} := \sum_{v \in C_{i,n}} \psi_n(v)$, and let $\tilde{\boldsymbol{\pi}}_n$ be the vector of $\tilde{\pi}_{i,n}$'s. The consistency theorem requires the following additional assumption, an analog to which can be found in Zhao et al. (2012) for consistency of modularity under the degree-corrected SBM:

Assumption 7 $\tilde{\boldsymbol{\pi}}_n \equiv \tilde{\boldsymbol{\pi}}$ *does not depend on n.*

Assumption 7 is made mainly for clarity. Without it, the condition in (15) of Theorem 5 (below) must hold for sufficiently large n, something which is inconsequential to the proof. Define $\mathbf{H} := \mathbf{P} \cdot \mathbf{M}$, the entry-wise product, which is symmetric because $\mathbf{P}$ and $\mathbf{M}$ are each symmetric. Note that when ϕ and ψ are proportional to 1-vectors, $\mathbb{E}(W_{uv}) = \mathbf{H}_{c(u)c(v)}$ for all $u, v \in N$. Thus, the interpretation of $\mathbf{H}$ is as the baseline inter/intra-community weight expectations after integrating out edge presence. Defining $\tilde{\mathbf{\Pi}} := \tilde{\boldsymbol{\pi}}\tilde{\boldsymbol{\pi}}^t$, we state the consistency theorem:

Theorem 5 *Fix $K > 0$. Let $\{\mathcal{G}_n\}_{n>1}$ be a sequence of networks generated by a K-community WSBM satisfying Assumptions 5-7. Suppose that the matrix*

$$\mathcal{M} := \mathbf{H} - \frac{\mathbf{H}\tilde{\mathbf{\Pi}}\mathbf{H}}{\tilde{\boldsymbol{\pi}}^t\mathbf{H}\tilde{\boldsymbol{\pi}}} \tag{15}$$

has positive diagonal entries and negative off-diagonal entries. If $\lambda_n / \log n \to \infty$, SCS is consistent for $\{\mathcal{G}_n\}_{n>1}$.

The proof of Theorem 5 is given in Appendix C. Some understanding of condition 15 can be had by considering when $K = 2$, when it reduces to the requirement $\mathbf{H}_{11}\mathbf{H}_{22} > \mathbf{H}_{12}^2$. More generally, and broadly speaking, the matrix $\mathcal{M}$ reveals whether or not appropriate signal exists in the model, with respect to the continuous configuration null. Notice that this signal need not be present in both $\mathbf{P}$ and $\mathbf{M}$. For instance, the condition can be satisfied even if $\mathbf{H}$ is a scalar multiple of $\mathbf{M}$, that is, if $\mathbf{P}$ is proportional to the $\mathbf{1}$-matrix. This entails that SCS is consistent even when the edge structure of $\mathcal{G}_n$ is Erdős-Renyi, as long as the edge weight signal (encoded in $\mathbf{M}$) is properly assortative. Of course, the opposite also holds, namely that SCS is consistent even when assortative community signal is only present in $\mathbf{P}$. In fact, one matrix may even be *disassortative* (some or all off-diagonal entries larger than diagonals) as long as the other is sufficiently assortative.

4.2.3 Connection to weighted modularity and related work

The conditions of Theorem 4 and Theorem 5 have a deep relationship to the modularity measure, discussed in Section 2. Explicitly, let the *weighted* modularity (WM) be the modularity metric with degrees replaced by strengths, as introduced in (Newman, 2004a). For fixed $n > 1$, let $\mathbf{c}$ be any partition of N. Define $K := \max\{\mathbf{c}\}$ and $C_u := \{v : c(v) = c(u)\}$. Then the (random) WM of $\mathbf{c}$ on $\mathcal{G}_n$ can be written

$$
\begin{aligned}
Q^w(\mathbf{c}, \mathcal{G}_n) &:= \frac{1}{S_{n,T}} \sum_{uv \in N} \{W_{uv} - r_{uv}(\mathbf{S}_n)\} \, \mathbb{1}\{c(u) = c(v)\} \\
&= \frac{1}{S_{n,T}} \sum_{i=1}^{K} \sum_{c(u)=c(v)} W_{uv} - r_{uv}(\mathbf{S}_n) = \frac{1}{S_{n,T}} \sum_{u \in N} \sum_{v \in C_u} W_{uv} - r_{uv}(\mathbf{S}_n) \\
&= \frac{1}{S_{n,T}} \sum_{u \in N} S(u, C_u, \mathcal{G}_n) - \mu_n(u, C_u | \mathbf{S}_n) = \frac{1}{S_{n,T}} \sum_{u \in N} A(u, C_u, \mathcal{G}_n)
\end{aligned}
$$

Thus, the contribution of u to WM with its assignment C_u is precisely the random association from u to C_u. Writing the population WM as $\bar{q}_n^w(\mathcal{C}) := n^{-1} \sum_u \tilde{a}_n(u, C_u)$, it is easily shown that condition (15) implies q_n^w is maximized by $\mathcal{C}_n$, the true community partition.

The consistency analysis of the (binary) modularity metric under the degree-corrected SBM, provided by Zhao et al. (2012), similarly hinges on maximization of population modularity. It is unsurprising, then, that the parameter condition for their result can be (analogously) expressed as a fixed $K \times K$ matrix having positive diagonals and negative off-diagonals. In fact, if the WSBM parameter $\mathbf{M}$ is proportional to a matrix of 1s, and the parameter ψ is a scalar multiple of ϕ, condition 15 in Theorem 5 is equivalent to the parameter assumptions on modularity consistency in Zhao et al. (2012). Furthermore, their analysis also requires that $\lambda_n / \log n \to \infty$. However, both the definition of consistency and proof approach for the theorems in this section are entirely novel.

5. The Continuous Configuration Model Extraction method

In the previous section, we established an asymptotic result showing that ground-truth communities are, with high probability, fixed points of the SCS algorithm. This result demonstrates the in-principle sensibility of the algorithm. In practice, we must rely on local, heuristic algorithms for initialization and termination, as with other exploratory methods. For instance, k-means is often used to initialize the EM algorithm, and modularity can be locally maximized through agglomerative pairing (Clauset et al., 2004). We incorporate SCS in a general community detection method for weighted networks entitled Continuous Configuration Model Extraction (CCME), written in loose detail as follows:

The CCME Community Detection Method for Weighted Networks

1. Given an observed weighted network $\mathcal{G}$, obtain initial node sets $\mathcal{B}_1 \subseteq 2^N$.
2. Apply SCS to each node set in $\mathcal{B}_0$, resulting in fixed points $\mathcal{C}$.
3. Remove sets from $\mathcal{C}$ that are empty or redundant.

These steps are described in more detail below. Importantly, the method has no connection to any graph-partition criteria. It proceeds solely by the SCS algorithm, which assesses communities independently. This allows CCME to adaptively return communities that share nodes ("overlap"), and, through the multiple testing procedure, ignore nodes not significantly connected to any stable communities ("background").

5.1 Step 1: Initialization

Just as mixture-models can be initialized with heuristic methods like k-means, it is possible to initialize CCME with partition-based community detection method. However, we have observed this approach to perform somewhat poorly in practice. Instead, we initialize with a novel search procedure based on the continuous configuration model. For fixed nodes $u, v \in N$, we define

$$z_u(v) := \max\left\{\frac{W_{uv} - f_{uv}(\mathbf{s}, \mathbf{d})}{\sqrt{\theta} f_{uv}(\mathbf{s}, \mathbf{d})},\ 0\right\}$$

The measure $z_u(v)$ acts like a truncated z-statistic, quantifying the extremity of the weight W_{uv}. The initial node set corresponding to u is formed by sampling $d(u)$ nodes with replacement from N with probability proportional to $z_u(v)$. The intuition behind this procedure is that if u is part of a highly-connected node set C, then $z_u(v)$ for nodes $v \in C$ will be larger (on average) than for other nodes.

5.2 Step 2: Application of SCS

Recall that, given an initial set B_1, SCS proceeds (via the update U_α) along a sequence of sets $B_2, B_3, \ldots, B_t, \ldots$ until $B_t = B_{t'}$ for some $t' < t$. Since the number of possible node subsets is finite, SCS is guaranteed to terminate in one of two states:

1. A stable community C, satisfying $U_\alpha(C, \mathcal{G}) = C$.
2. A stable sequence of communities $C_1, \ldots, C_J$ satisfying

$$U_\alpha(C_1, \mathcal{G}) = U_\alpha(C_2, \mathcal{G}) = \ldots = U_\alpha(C_J, \mathcal{G}) = U_\alpha(C_1, \mathcal{G}).$$

In practice, on empirical and simulated data, case 1 is the majority. Furthermore, empirically speaking, stable sequences (case 2) usually consist only of two communities that overlap. Though case 2 does not result in a clear-cut community, a stable sequence may still be of practical interest if the constituent sets have high overlap. In Appendix D.2, we give a routine to either re-initialize or terminate SCS when it encounters a stable sequence.

5.3 Step 3: Filtering of $\mathcal{C}$

The CCME community detection method returns a final collection of communities $\mathcal{C}$, containing the results of the SCS algorithm for each initial set in $\mathcal{B}_0$. By default, we remove any empty or duplicate sets from $\mathcal{C}$. In some applications, pairs of sets in $\mathcal{C}$ will have high Jaccard similarity. In Appendix D.3, we detail a method of pruning these near-duplicates from $\mathcal{C}$. Additionally, in Appendix D.3, we describe routines to suppress the application of SCS to initial sets that are "weakly" intra-connected, or with high overlap to already-extracted communities. These routines greatly reduce the runtime of CCME, and, on some simulated networks, improve accuracy.

5.4 Remarks

1. The parameter α, used in the set update operation U_α, must be specified by the user of CCME. Having a natural interpretation as the false-discovery rate for each update, α was set to 0.05 for all simulations and real data analyses introduced in this paper. We found that $\alpha = 0.05$ was a universally effective default setting, and that CCME's results change negligibly for other values of α within a reasonable window.

2. Multiple testing corrections computed within CCME are valid only for the input network. If other edge-weighted networks with the same node set are available, say, from an alternate data source or different set of node covariates, then it is necessary to apply a multiple-testing correction across communities discovered by the separate runs of CCME. The community-wise z-statistic introduced in Appendix D.3 will be useful to this end.

3. CCME can be extended to directed networks, as described in Appendix E.

6. Simulations

This section contains a performance analysis of CCME and existing methods on a benchmarking simulation framework. Simulated networks were generated from the Weighted Stochastic Block Model (see Section 4.2.1), with slight modifications to include overlapping communities and background nodes, when necessary. The performance measures, competing methods, simulation settings, and results are described below.

6.1 Performance measures and competing methods

We used three community detection method performance metrics:

1. **Overlapping Normalized Mutual Information (oNMI):** Introduced by Lancichinetti et al. (2009), oNMI is an information-based measure between 0 and 1 that approaches 1 as two covers of the same node set become similar and equals 1 when they are the same. From a method's results, we calculate oNMI with respect to the true communities *only* for the nodes the method placed into communities.

2. **Community nodes in background (%C.I.B.):** The percentage of true community nodes incorrectly assigned to background.

3. **Background nodes in communities (%B.I.C.):** The percentage of true background nodes (if present) incorrectly placed into communities.

In addition to CCME, three other weighted-network methods capable of identifying overlapping nodes were applied to our simulation ensemble. One of these was OSLOM (Lancichinetti et al., 2011), described in Section 1. Another was SLPAw, a weighted-network version of an overlapping label propagation algorithm (Xie et al., 2011). Finally, we applied the weighted Stochastic Block Model fitting procedure introduced by Peixoto (2017), which we will call "WeightedGT" after it's implementation's package ("graph-tool"). Also included were four commonly used score-based methods implemented in the `R` package `igraph` (Csárdi and Nepusz, 2006): Fast-Greedy, which performs approximate modularity optimization via a hierarchical agglomeration (Clauset et al., 2004); Louvain, an approximate modularity optimizer that proceeds through node membership swaps (Blondel et al., 2008); Walktrap, an agglomerative algorithm that locally maximizes a score based on random walk theory (Pons and Latapy, 2006); Infomap, an information-flow mapping algorithm that uses random walk transition probabilities (Rosvall and Bergstrom, 2008).

Remark. Being extraction methods, only CCME and OSLOM naturally specify background nodes, via testing. As such, we will often make direct comparative comments between OSLOM and CCME with respect to background node handling. For other methods, we take as background any nodes in singleton communities. However, these methods almost never returned singleton communities, even when the simulation had weak or non-existent signal.

6.2 Simulation settings and results

We now give an overview of the simulation procedure for the benchmarking framework. A complete account is given in Appendix F. We first describe "default" parameter settings of the WSBM; in the simulation settings below, individual parameters are toggled around their default values, to reveal the dependence of the methods to those parameters. At each unique parameter setting, 20 random networks were simulated. The points in each plot from Figure 1 show the average performance measure of the methods over the 20 repetitions.

The default WSBM setting has the number of nodes at $n = 5,000$. The community memberships were set by obtaining community sizes from a power law, then assigning nodes uniformly at random. This process produced approximately 3 to 7 communities per network. Many of these simulation features (as well as those described in what follows) mirror those used in existing frameworks (e.g. Lancichinetti et al., 2011); full details are provided in Appendix F. Recall the parameters $\mathbf{P}$ and $\mathbf{M}$, which induce baseline intra- and inter-community edge and weight signal. In the default setting, these matrices have off-diagonals equal to 1 and diagonals equal to constants $s_e = 3$ and $s_w = 3$ (respectively). In some simulation settings, overlapping and background nodes are added (as described later in this section), but the default setting includes neither overlap nor background.

Common parameter settings. For all simulated networks (regardless of the setting), the node-wise edge parameters ϕ were drawn from a power law to induce degree heterogeneity. The parameter ϕ is scaled so that the expected average degree of each network was equal to $\sqrt{n}$, which induces sparsity in the network. The parameter ψ is set by the formula $\psi = \phi^{1.5}$ to ensure a non-trivial relationship between expected degrees and expected strengths (see Appendix F).

6.2.1 Networks with varying signal levels

The first simulation setting tested the methods' dependence on s_e and s_w. These values were moved along an even grid on the range $[1, 3]$. Plots A-1 and B-1 in Figure 1 show the performance measure results when s_w is fixed at 3, plots A-2 and B-2 show results when $s_e = 3$, and plots A-3 and B-3 show results when s_e and s_w are moved along $[1, 3]$ together. Many methods had large oNMI scores in this simulation setting. We transformed the oNMI scores using the function

$$\text{t-oNMI}_a(x) := \left(\tfrac{1}{1-x+a} - \tfrac{1}{1+a}\right)/\left(\tfrac{1}{a} - \tfrac{1}{1+a}\right)$$

with $a = 0.05$. This is a monotonic, one-to-one transformation from $[0, 1]$ to itself, which stretches the region close to 1, allowing a clearer comparison between the methods' performances. CCME consistently out-performed all competing methods, especially when either the edge or weight signal was completely absent. (The t-oNMI transformation masks the trends in lower-performing methods, so we show these plots with un-transformed oNMI in Appendix F.3.)

The plots in row B show that when either s_e or s_w were near 1, OSLOM and CCME assigned many background nodes. This is consistent with these methods' unique abilities to leave nodes unassigned when they are not significantly connected to communities. That said, %C.I.B. can be seen as a measure of sensitivity, since ideally no nodes would be assigned to background when any signal is present. In this regard, CCME outperformed OSLOM across the range of model parameters.

6.2.2 Networks with overlapping communities

The second setting involved networks with overlapping nodes. To add overlapping nodes to the default network, two parameters were introduced: o_n, the number of overlapping nodes, and o_m, the number of memberships for each overlapping node. The particular overlapping nodes and community memberships were chosen uniformly-at-random. Plots C-1 and C-2 show performance results from the setting with o_n moving away from 0 and $o_m = 2$. Plot C-3 shows results from the setting with $o_n = 500$ and $o_m \in \{1, \ldots, 4\}$. We find that CCME consistently outperforms all methods in terms of accuracy (oNMI), and outperforms OSLOM in terms of sensitivity (%C.I.B.).

6.2.3 Networks with overlapping communities and background nodes

The final simulation setting involved networks with both overlap and background nodes. The number of background nodes was fixed at 1,000, and number of community nodes varied from $n = 500$ to $n = 5,000$. For each network, $o_n = n/4$ nodes were randomly chosen to overlap $o_m = 2$ communities (also chosen at random). Background nodes were created by first simulating the n-node community sub-network, and then generating the 1,000-node background sub-network according to the continuous configuration model, using empirical degrees and strengths from the community sub-network. The complete details of this procedure are given in Appendix F.

The results of this simulation setting are shown in row D from Figure 1. From plot D-1, we see that OSLOM and CCME had the highest oNMI scores, favoring OSLOM when the number of community nodes decreased. Because this simulation setting involved background nodes, the %B.I.C. metric is relevant, and can be taken as a measure of specificity: ideally, nodes from the background sub-network should be excluded from communities. From plot D-2, we see that methods incapable of assigning background had %B.I.C. equal to 1. We found that CCME correctly ignored background nodes as the network size increased, whereas OSLOM became increasingly *anti*-conservative for larger networks. Furthermore, CCME again had lower %C.I.B. than OSLOM.

7. Applications

In this section, we discuss applications of CCME, OSLOM, and SLPAw (the methods capable of returning overlapping communities) to two real data sets.

7.1 U.S. airport network data

The first application involves commercial airline flight data, obtained from the Bureau of Transportation Statistics (www.transtats.bts.gov). For each month from January to July of 2015, we created a weighted network with U.S. airports as nodes, edges connecting airports that exchanged flights, and edges weighted by aggregate passenger count. We also constructed a year-aggregated network, formed simply by taking the union of the month-wise edge sets, and adding the month-wise weights. In Figure 2, we display the methods' results when applied to the June and year-aggregated data sets from 2015. Each discovered community (within-method) has a unique color and shape. Each overlapping node is plotted multiple times, one for each community in which it was placed. To allow for a clearer visualization of communities, background nodes are not shown.

Overall, the CCME results, in contrast to results from OSLOM and SLPAw, suggest that many airports in the U.S. airport system may not participate in meaningful community behavior. The fact that CCME performs multiple testing against an explicit null model gives this result some validity. Furthermore, airports in significant communities tend to be located near large hubs or in geographically isolated areas. We also see that, with the monthly data, OSLOM and CCME tended to find communities consistent with geography, whereas SLPAw placed most of the network into one community. With the year-aggregated data, OSLOM also agglomerated most airports, whereas CCME continued to respect the geography. Since the aggregated data is much more edge-dense, this suggests the performance of OSLOM and SLPA may suffer on weighted graphs with high or homogeneous edge-density, whereas CCME is able to detect proper community structure from the weights alone. This aligns with the simulation results described in Section 6.2.1.

7.2 ENRON email network

An email corpus from the company ENRON was made available in 2009. The unweighted network formed by linking communicating email addresses is well-studied; see www.cs.cmu.edu/~./enron for references and Leskovec et al. (2010) for the data. For the purposes of this paper, we derived a *weighted* network from the original corpus, using

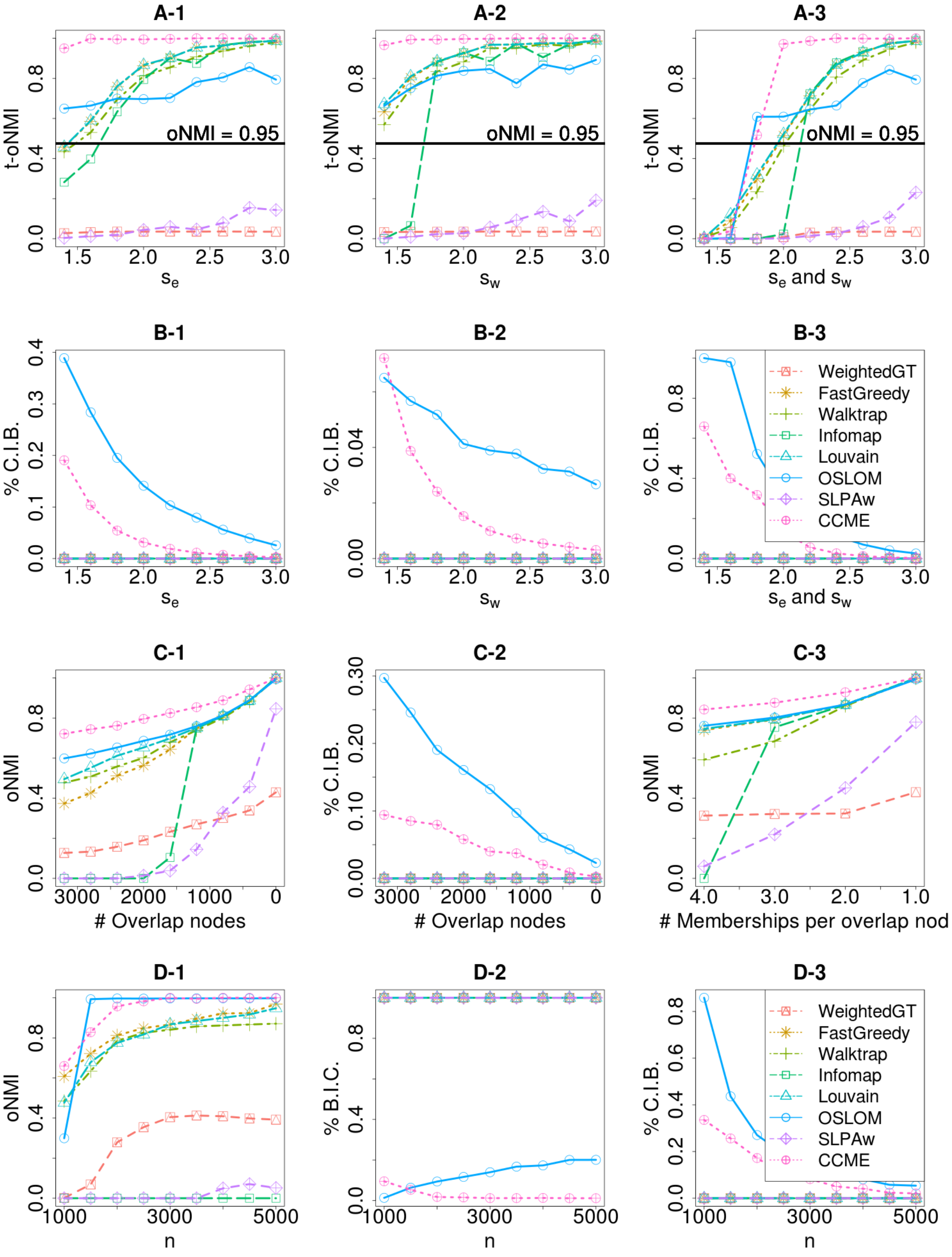
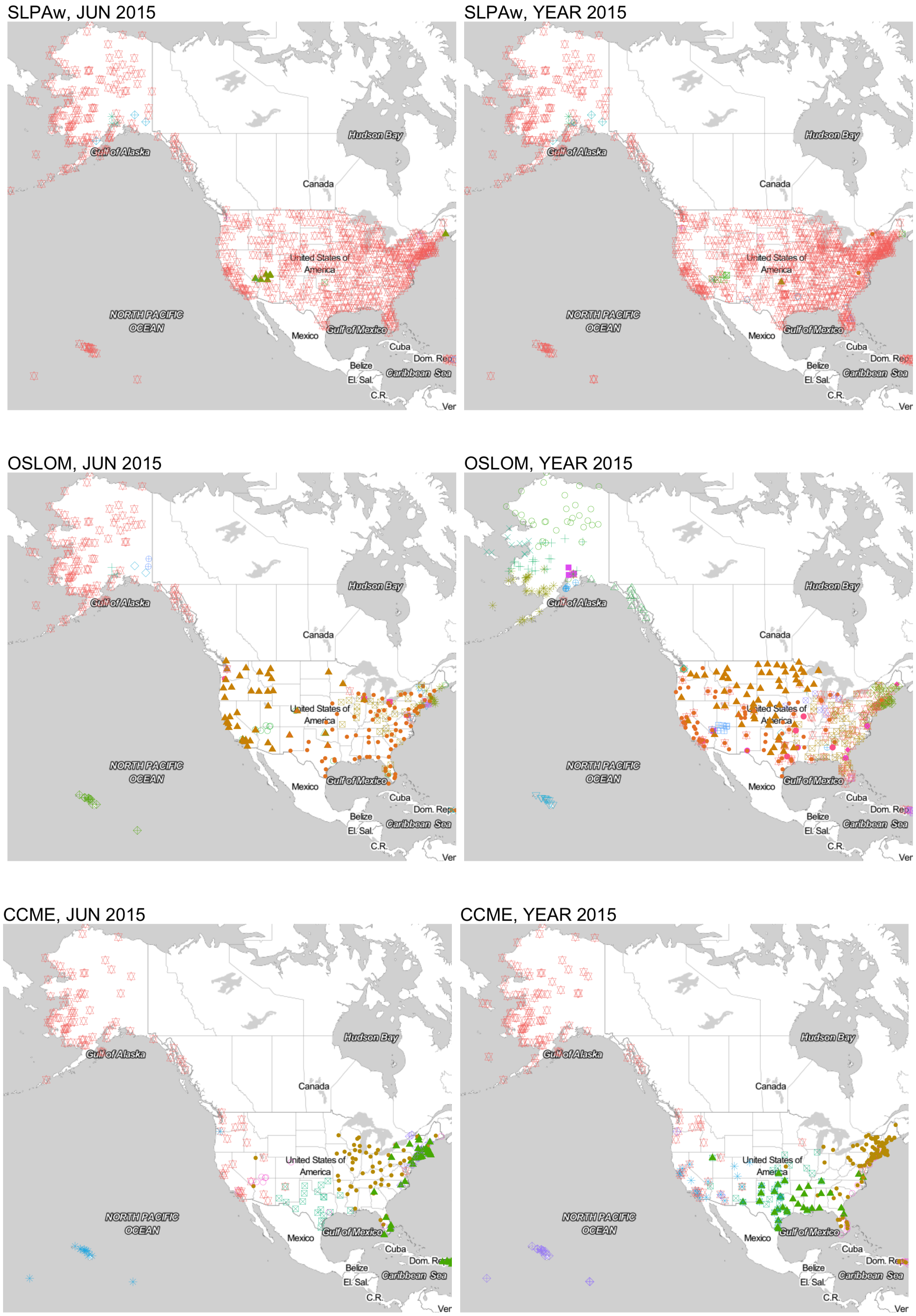


Figure 1: Simulation results described in Sections 6.2.1-6.2.3. Legends refer to all plots.



SLPAw, JUN 2015

SLPAw, YEAR 2015

OSLOM, JUN 2015

OSLOM, YEAR 2015

CCME, JUN 2015

CCME, YEAR 2015

Figure 2: SLPAw, OSLOM, and CCME results from June 2015 and 2015-year-aggregated U.S. airport networks. Maps created with `ggmap` (Kahle and Wickham, 2013).

message count between addresses as edge weights. Though the corpus was formed from email folders of 150 ENRON executives, we made the network from addresses found in *any* message. This full network has 80,702 unique addresses, 47,481 of which were external to the company. Thus, the network likely has many spam or irrelevant senders, i.e. "true" background nodes. We applied CCME, OSLOM, and SLPAw to the network to see which methods best focused on company-specific areas of the data.

Tables 1 and 2 give basic summaries of the results, which show noticeable differences between the outputs of the methods. CCME placed far fewer into nodes into communities, but detected larger communities with more overlapping nodes. Notably, CCME had the highest percentage of ENRON addresses among nodes it placed into communities (see Table 3). These results suggest that CCME was more sensitive to critical relationships in the network.

Table 1: Metrics from methods' results on ENRON network: number of communities, min. comm. size, median comm. size, max. comm. size, # nodes in communities.

	Num.Comms	Min.size	Med.size	Max.size	Num.Nodes
CCME	185	2	687	5416	14552
OSLOM	405	2	19	770	17635
SLPAw	2138	2	4	4793	79316

Table 2: Metrics from methods' results on ENRON network: number of overlapping nodes, minimum # of memberships, median # of mem'ships, max. # of mem'ships.

	Num.OL.Nodes	Min.mships	Med.mships	Max.mships
CCME	8104	2	9	78
OSLOM	462	2	2	8
SLPAw	3860	2	2	4

Table 3: Top domains associated with community nodes from each method, by proportion.

CCME.Domains	Prop.
enron.com	0.784
aol.com	0.008
cpuc.ca.gov	0.006
pge.com	0.004
socalgas.com	0.003
dynegy.com	0.003

OSLOM.Domains	Prop.
enron.com	0.529
aol.com	0.029
haas.berkeley.edu	0.016
hotmail.com	0.015
yahoo.com	0.009
jmbm.com	0.005

SLPAw.Domains	Prop.
enron.com	0.423
aol.com	0.039
hotmail.com	0.023
yahoo.com	0.016
haas.berkeley.edu	0.007
msn.com	0.006

8. Discussion

In this paper, we introduced the continuous configuration model, which is, to the best of our knowledge, the first null model for community detection on weighted networks. The explicit generative form of the null model allowed the specification of CCME, a community extraction method based on sequential significance testing. We showed that a standardized statistic for the tests is asymptotically normal, a result which enables an analytic approximation to p-values used in the method. We also proved asymptotic consistency under a weighted stochastic block model for the core algorithm of the method.

On simulated networks the proposed method CCME is competitive with commonly-used community detection methods. CCME was the dominant method for simulated networks with large numbers of overlapping nodes. Furthermore, on networks with background nodes, CCME was the only method to correctly label true background nodes while maintaining high detection power and accuracy for nodes belonging to communities. On real data, CCME gave results that were both interpretable and revelatory with respect to the natural system under study.

We expect that the continuous configuration model will have applications outside the setting of this paper, just as the binary configuration model has been studied in diverse contexts. One may investigate the distributional properties of many different graph-based statistics under the model, as a means of assessing statistical significance in practice. For instance, an appropriate theoretical analysis could yield an approach to the assessment of statistical significance of weighted modularity. The proof approach used for Theorem 2 may serve as a basis for this endeavor. Another benefit of an explicit null for weighted networks is the potential for simulation. Using the continuous configuration model, and parts of the framework presented in this paper, one can generate weighted networks having true background nodes with arbitrary expected degree and strength distributions.

8.1 Acknowledgements and Remarks

The authors wish to thank three anonymous referees for their thoughtful suggestions, and Dr. Peter J. Mucha for helpful discussions. The R code for the CCME method is available in the github repository 'jpalowitch/CCME'. The code for reproducing the analyses in Sections 6 and 7 is available at the github repository 'jpalowitch/CCME_analyses'. The work of JP and ABN was supported in part by NIH grant MH101819-01. The work of SB and ABN was supported in part by NSF grant DMS-1310002. SB was also supported in part by NSF grants DMS-1105581, DMS-161307, and DMS-160683, SES grant 1357622, and ARO grant W911NF-17-1-0010. ABN was also supported in part by NSF DMS-1613072 and NIH HG009125-01.

Appendix A. Proof of Proposition 1

Equation 8 follows immediately from the observation in equation 3 and the definition of $r_{uv}(\mathbf{s})$. Next, note that

$$\mathbb{E}\left(W_{uv}|A_{uv}\right) = f_{uv}(\mathbf{d},\mathbf{s})A_{uv}, \quad \text{and} \quad \mathrm{Var}(W_{uv}|A_{uv}) = \kappa f_{uv}(\mathbf{d},\mathbf{s})^2 A_{uv}.$$

Thus, using the law of total variance,

$$\begin{aligned}\mathrm{Var}(W_{uv}) &= f_{uv}(\mathbf{d},\mathbf{s})^2\mathrm{Var}(A_{uv}) + \kappa f_{uv}(\mathbf{d},\mathbf{s})^2\mathbb{E}(A_{uv})\\ &= f_{uv}(\mathbf{d},\mathbf{s})^2\tilde{r}_{uv}(\mathbf{d})(1-\tilde{r}_{uv}(\mathbf{d})) + \kappa f_{uv}(\mathbf{d},\mathbf{s})^2\tilde{r}_{uv}(\mathbf{d})\\ &= r_{uv}(\mathbf{s})f_{uv}(\mathbf{d},\mathbf{s})\left(1-\tilde{r}_{uv}(\mathbf{d})+\kappa\right)\end{aligned}$$

Summing over $v \in B$ gives equation 9. ■

Appendix B. Proof of Theorem 2 and supporting lemmas.

Here we give the proof of Theorem 2 in Section 4.1. We start with supporting lemmas. Recall the definition of the average degree parameter λ_n, the normalized r^{th}-moment $L_{n,r}$, and other associated definitions from Section 4.1. For the purposes of the results below, we define the following generalization of $L_{n,r}$, given a node set $B_n \subseteq N$ with $b_n := |B_n|$:

$$L_{n,r}(B_n) := b_n^{-1} \sum_{u \in B_n} \{d_n(u)/\lambda_n\}^r$$

Note that $L_{n,r}(N) = L_{n,r}$. Recall that in the setting of Theorem 2, the node set B_n is chosen uniformly from the node set N. The first result involves a *deterministic* sequence $\{B_n\}_{n \geqslant 1}$:

Lemma 6 *For each $n > 1$, let $\mathcal{G}_n$ be generated by the continuous configuration model with parameters $\theta_n = (\mathbf{d}_n, \mathbf{s}_n, \kappa_n)$ and common weight distribution F. Fix a node sequence $\{u_n\}_{n>1}$ with $u_n \in N$ and a positive integer sequence $\{b_n\}_{n>1}$ with $b_n \leqslant n$. Suppose the parameter sequence $\{d_n(u_n)\}_{n \geqslant 1}$ satisfies*

$$\frac{d_n(u_n) b_n}{n} \to \infty \text{ as } n \to \infty$$

Fix $\varepsilon > 0$ as in Assumption 2, and choose $\delta \in (0, 1)$ such that $2\beta\delta < \varepsilon$. Fix a sequence of sets $\{B_n\}_{n>1}$ with $|B_n| = b_n$ for all n, and suppose that for $r = 2\beta + 1$ and $r = \beta(2+\delta)+1$, the sequence $\{L_{n,r}(B_n)\}_{n>1}$ is bounded away from zero and infinity. Then

$$\frac{S(u_n, B_n, \mathcal{G}_n) - \mu_n(u_n, B_n|\theta_n)}{\sigma_n(u_n, B_n|\theta_n)} \Rightarrow \mathcal{N}(0, 1) \quad \text{as} \quad n \to \infty$$

Proof In what follows, the functions r_{uv} and $\tilde{r}_{uv}$ from Section 1.2 will be used extensively. Note that for any nodes u, v, $\mathbb{E}W_{uv} = r_{uv}(\mathbf{s})$. Thus by the classical Lyapunov central limit theorem it suffices to show that

$$\frac{\sum\limits_{v \in B_n} \mathbb{E}|W_{u_n,v} - r_{u_n v}(\mathbf{s}_n)|^{2+\delta}}{\left(\sqrt{\sum\limits_{v \in B_n} \mathbb{E}\left\{(W_{u_n,v} - r_{u_n v}(\mathbf{s}_n))^2\right\}}\right)^{2+\delta}} \to 0 \tag{16}$$

as n tends to infinity. The following derivations hold for any fixed $n > 1$, so we suppress dependence on n from u_n, and B_n, and similar expressions. For the numerator of (16), we have

$$\begin{aligned} \mathbb{E}|W_{u,v} - r_{uv}(\mathbf{s})|^{2+\delta} &= \left(\frac{r_{uv}(\mathbf{s})}{\tilde{r}_{uv}(\mathbf{d})}\right)^{2+\delta} \mathbb{E}\left(|\xi_{uv} A_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right) \\ &= f_{uv}(\mathbf{d}, \mathbf{s})^{2+\delta} \cdot \mathbb{E}\left(|\xi_{uv} A_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right), \end{aligned} \tag{17}$$

by definition of the model in Section 2.1. Moreover, by the law of total variance,

$$\begin{aligned}\mathbb{E}(|\xi_{uv}A_{uv}-\tilde{r}_{uv}(\mathbf{d})|^{2+\delta}) &= (1-\tilde{r}_{uv}(\mathbf{d}))\tilde{r}_{uv}(\mathbf{d})^{2+\delta}+\tilde{r}_{uv}(\mathbf{d})\mathbb{E}|\xi_{uv}-\tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\\ &= \left\{(1-\tilde{r}_{uv}(\mathbf{d}))\tilde{r}_{uv}(\mathbf{d})^{1+\delta}+\mathbb{E}|\xi_{uv}-\tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right\}\cdot\tilde{r}_{uv}(\mathbf{d})\\ &\leqslant C\cdot\tilde{r}_{uv}(\mathbf{d}) \qquad (18)\end{aligned}$$

for some positive constant C, by Assumption 4. Next, we note that by Assumption 1, there exist positive constants $a<c$ such that for all $v\in N$,

$$a\cdot d_n(v)^\beta\leqslant\frac{s_n(v)}{d_n(v)}\leqslant c\cdot d_n(v)^\beta,$$

for n sufficiently large. Thus, if $r_{uv}(\mathbf{d})\leqslant 1$, $\tilde{r}_{uv}(\mathbf{d})=r_{uv}(\mathbf{d})$, and

$$f_{uv}(\mathbf{d},\mathbf{s})=\frac{r_{uv}(\mathbf{s})}{\tilde{r}_{uv}(\mathbf{d})}=\left(\frac{d_T}{s_T}\right)\frac{s(u)s(v)}{d(u)d(v)}\leqslant c\cdot\left(\frac{d_T}{s_T}\right)\{d(u)d(v)\}^\beta. \qquad (19)$$

If $r_{uv}(\mathbf{d})>1$, $\tilde{r}_{uv}(\mathbf{d})=1$, and by Assumption 3 there exists c' such that

$$\begin{aligned}f_{uv}(\mathbf{d},\mathbf{s})=\frac{s(u)s(v)}{s_T}&\leqslant c\cdot\left(\frac{d(u)d(v)}{s_T}\right)\{d(u)d(v)\}^\beta\\ &= c\cdot\left(\frac{d_T}{s_T}\right)r_{uv}(\mathbf{d})\{d(u)d(v)\}^\beta\leqslant c'\cdot\left(\frac{d_T}{s_T}\right)\{d(u)d(v)\}^\beta. \qquad (20)\end{aligned}$$

Therefore, combining (18)-(20) with (17), there exists $C>0$ such that

$$\begin{aligned}\mathbb{E}|W_{u,v}-r_{uv}(\mathbf{s})|^{2+\delta} &\leqslant C\left(\frac{d_T}{s_T}\right)^{2+\delta}\cdot\{d(u)d(v)\}^{\beta(2+\delta)}\tilde{r}_{uv}(\mathbf{d})\\ &= C\left(\frac{d_T}{s_T}\right)^{2+\delta}\cdot\{d(u)d(v)\}^{\beta(2+\delta)}\frac{d(u)d(v)}{d_T}\\ &\leqslant C\cdot d_T^{1+\delta}s_T^{-(2+\delta)}\cdot\{d(u)d(v)\}^{\beta(2+\delta)+1} \qquad (21)\end{aligned}$$

A similar analysis of the summands in the denominator of (16) gives

$$\mathbb{E}\left\{(W_{u,v}-r_{uv}(\mathbf{s}))^2\right\}\geqslant C'\cdot d_Ts_T^{-2}\cdot\{d(u)d(v)\}^{2\beta+1} \qquad (22)$$

for appropriately chosen C'. Let $b=|B|$. Combining (21) and (22), with some algebra, we find that the left side of (16) is (up to a constant) less than

$$\left(\frac{d(u)}{d_T}\right)^{-\delta/2}\cdot\frac{\sum_{v\in B}d(v)^{\beta(2+\delta)+1}}{\left(\sum_{v\in B}d(v)^{2\beta+1}\right)^{1+\delta/2}}$$

$$=\left(\frac{d(u)}{d_T}b\lambda\right)^{-\delta/2}\cdot\frac{b^{-1}\sum_{v\in B}(d(u)/\lambda)^{\beta(2+\delta)+1}}{\left\{b^{-1}\sum_{v\in B}(d(u)/\lambda)^{2\beta+1}\right\}^{1+\delta/2}}$$

$$=\left(\frac{d(u)}{d_T}b\lambda\right)^{-\delta/2}\cdot\frac{L_{n,\beta(2+\delta)+1}(B)}{(L_{n,2\beta+1}(B))^{1+\delta/2}}\;=\;O\left\{\left(\frac{d(u)}{d_T}b\lambda\right)^{-\delta/2}\right\} \qquad (23)$$

where the final term follows from our assumptions on $L_{n,\beta(2+\delta)+1}(B_n)$ and $L_{n,2\beta+1}(B_n)$. By definition, $d_{n,T} = n\lambda_n$, so the final expression above is $O\left\{(d_n(u_n)b_n/n)^{-\delta/2}\right\} = o(1)$ by assumption. Thus (16) holds and the result follows. ■

We now proceed with the proof of Theorem 2. Proposition 6 yields the CLT for $S(u_n, B_n, \mathcal{G}_n)$ for a deterministic sequence of vertex sets $\{B_n\}_{n\geqslant 1}$ satisfying regularity properties. The remainder of the argument shows that if B_n is selected uniformly at random then, under the assumptions of Theorem 2, these regularity properties are satisfied with high probability. We begin with a few preliminary definitions and results.

Definition 7 *A sequence of random variables* $\{X_n\}_{n\geqslant 1}$ *is said to be* asymptotically uniformly integrable *if*

$$\lim_{M\to\infty} \limsup_{n\to\infty} \mathbb{E}\left\{|X_n|\mathbb{1}(|X_n| > M)\right\} = 0$$

Theorem 8 *Let* $f : \mathbb{R}^k \mapsto \mathbb{R}^k$ *be measurable and continuous at every point in a set* C*. Suppose* $X_n \xrightarrow{w} X$ *where* X *takes its values in an interval* C*. Then* $\mathbb{E}f(X_n) \to \mathbb{E}f(X)$ *if and only if the sequence of random variables* $f(X_n)$ *is asymptotically uniformly integrable.*

Proof See Asymptotic Statistics (Van der Vaart 2000), page 17. ■

We now give a technical lemma (needed for a subsequent result) which uses Theorem 8:

Lemma 9 *Let* $X_1, X_2, \ldots$ *be non-negative random variables and let* $s, \varepsilon > 0$*. If the sequences* $\{\mathbb{E}X_n^s\}_{n\geqslant 1}$ *and* $\{\mathbb{E}X_n^{s+\varepsilon}\}_{n\geqslant 1}$ *are bounded away from zero and infinity, then* $\{\mathbb{E}X_n^r\}_{n\geqslant 1}$ *is bounded away from zero and infinity for every* $r \in (0, s+\varepsilon)$*.*

Proof Suppose by way of contradiction that there exists $t \in (0, s+\varepsilon)$ such that $\liminf_n \mathbb{E}X_n^t = 0$. Then $\lim_k \mathbb{E}X_{n_k}^t = 0$ along a subsequence $\{n_k\}$. As the random variables $X_{n_k}^t$ are non-negative, $X_{n_k}^t \xrightarrow{d} 0$, and it follows from the continuous mapping theorem that $X_{n_k} \xrightarrow{w} 0$. As $M^{\varepsilon/s}\, X_n^s\, \mathbb{1}(X_n^s > M) \leqslant X_n^{s+\varepsilon}$, we find that

$$\lim_{M\to\infty} \limsup_{k\to\infty} \mathbb{E}\{X_{n_k}^s \mathbb{1}(X_{n_k}^s > M)\} \;\leqslant\; \lim_{M\to\infty} M^{-\varepsilon/s} \limsup_{k\to\infty} \mathbb{E}(X_{n_k}^{s+\varepsilon}) = 0$$

as $\mathbb{E}(X_n^{s+\varepsilon})$ is bounded by assumption. It then follows from Theorem 8 and the fact that $X_{n_k}^s \xrightarrow{w} 0$ that $\mathbb{E}X_{n_k}^s \to 0$ as $k \to \infty$, violating our assumption that $\mathbb{E}X_n^s$ is bounded away from zero. We conclude that $\mathbb{E}X_n^r$ is bounded away from zero for $r \in (0, s+\varepsilon)$. On the other hand, if $r \in (0, s+\varepsilon)$ then for each $n \geqslant 1$

$$\mathbb{E}\{X_n^r \mathbb{1}(X_n > 1)\} \;\leqslant\; \mathbb{E}\{X_n^{s+\varepsilon}\mathbb{1}(X_n > 1)\} \;\leqslant\; \sup_n \mathbb{E}\{X_n^{s+\varepsilon}\}$$

As the last term is finite by assumption and $\mathbb{E}\{X_n^r \mathbb{1}(X_n \leqslant 1)\}$ is at most one, it follows that $\mathbb{E}(X_n^r)$ is bounded. ■

Lemma 10 *Suppose a degree parameter sequence $\{\mathbf{d}_n\}_{n \geqslant 1}$ satisfies Assumption 2 from Section 4.1. For each n, let B_n be a randomly chosen subset of N of size b_n, where $b_n \to \infty$. Fix $\varepsilon > 0$ as in Assumption 2, and choose δ so that $2\beta\delta < \varepsilon$. Then for every $r \in (0,\ \beta(2+\delta)+1]$, there exists an interval $I_r = (a_r, b_r)$ with $0 < a_r < b_r < \infty$ such that $\mathbb{P}\{L_{n,r}(B_n) \in I_r\} \to 1$ as $n \to \infty$.*

Remark: Note that the function $L_{n,r}(\cdot)$ is non-random. The probability appearing in the conclusion of Lemma 10 depends only on the random choice of the vertex set B_n.

Proof Let D_n and D'_n be drawn uniformly-at-random from $\mathbf{d}_n$ without replacement, and fix $r \in (0,\ \beta(2+\delta)]$. A routine calculation gives

$$\operatorname{Var}\{L_{n,r}(B_n)\} = b_n^{-1}\lambda_n^{-2r}\left[\operatorname{Var}\{D_n^r\} + \{b_n - 1\}\operatorname{Cov}\{D_n^r, (D'_n)^r\}\right].$$

Note that $\mathbb{E}(D_n^r) = \lambda_n^r L_{n,r}$ and $\mathbb{E}(D_n^{2r}) = \lambda_n^{2r} L_{n,2r}$, so $\operatorname{Var}(D_n^r) = \lambda_n^{2r}(L_{n,2r} - L_{n,r})$. Furthermore, a simple calculation shows that $\operatorname{Cov}\{D_n^r, (D'_n)^r\}$ is negative for every r, and therefore $\operatorname{Var}\{L_{n,r}(B_n)\} \leqslant b_n^{-1}(L_{n,2r} - L_{n,r})$. Our choice of δ ensures that $2r < 4\beta + 2 + \varepsilon$, and it then follows from Lemma 9 and Assumption 2 that $L_{n,2r}$ and $L_{n,r}$ are bounded. Thus $\operatorname{Var}\{L_{n,r}(B_n)\} = O(b_n^{-1})$. Define $\Delta := \liminf_n\ L_{n,r}/2$, which is positive by Assumption 2, and let

$$I_r := \left(\liminf_{n\to\infty} L_{n,r} - \Delta,\ \limsup_{n\to\infty} L_{n,r} + \Delta\right) \tag{24}$$

As $\mathbb{E}\{L_{n,r}(B_n)\} = L_{n,r}$, an application of Chebyshev's inequality yields the bound

$$\begin{aligned}\mathbb{P}\{L_{n,r}(B_n) \notin I_r\} &\leqslant \mathbb{P}\{|L_{n,r}(B_n) - \mathbb{E}[L_{n,r}(B_n)]| > \Delta/2\} \\ &\leqslant \frac{4\operatorname{Var}\{L_{n,r}(B_n)\}}{\Delta^2} = O(b_n^{-1}).\end{aligned}$$

As b_n tends to infinity with n, the result follows. ■

B.1 Completing the proof of Theorem 2.

Let ε and δ be as in Proposition 6 and Lemma 10. Note that since $d_n(u_n) \leqslant n$ for all n, our assumption that $b_n d_n(u_n)/n \to \infty$ implies $|B_n| = b_n \to \infty$. Hence by lemma 10, we have that for both $r = \beta(2+\delta)+1$ and $r = 2\beta + 1$, there exists a positive, finite interval I_r such that $\mathbb{P}\{L_{n,r}(B_n) \in I_r\} \to 1$ as $n \to \infty$. Thus given any subsequence $\{n_k\}_{k\geqslant 1}$ we can find a further subsequence $\{n'_k\}_{k\geqslant 1}$ such that $L_{n'_k,r}(B_{n'_k}) \in I_r$ almost surely as $k \to \infty$, which means this sequence is bounded away from zero and infinity in k. Now using Proposition 6, for almost every ω we have

$$\frac{S_{n'_k}(u_{n'_k}, B_{n'_k}, \mathcal{G}_{n'_k}) - \mu_{n'_k}(u_{n'_k}, B_{n'_k}|\theta_{n'_k})}{\sigma_{n'_k}(u_{n'_k}, B_{n'_k}|\theta_{n'_k})} \Rightarrow \mathcal{N}(0,1) \text{ as } k \to \infty$$

Applying the subsequence principle completes the proof. ■

Appendix C. Proof of Theorems 4-5 and supporting lemmas.

Throughout this section, notation and conventions from Section 4.2.1 will be used, though we suppress dependence on n for convenience. Further recall functions r and f from Section 1.2. The following additional notation will be used throughout this section:

- Define $\phi_T := \sum_{v\in N}\phi(v)$ and $\psi_T := \sum_{v\in N}\psi(v)$. For each $K \geqslant j \geqslant 1$, define $\tilde{\pi}_j^0 := \sum_{v\in\mathcal{C}_j}\phi(v)/\phi_T$ and $\tilde{\pi}_j := \sum_{v\in\mathcal{C}_j}\psi(v)/\psi_T$. Let $\tilde{\boldsymbol{\pi}}^0$ and $\tilde{\boldsymbol{\pi}}$ be the associated vectors.
- Let $\langle\cdot,\cdot\rangle$ denote the vector dot-product. For a general symmetric matrix $\mathbf{A}$, let $\mathbf{A}_{ij}$ be the i,j-th entry, and $\mathbf{A}_i$ the i-th column. Define $\mathbf{H} := \mathbf{P}\cdot\mathbf{M}$, the entry-wise product.
- Let $D(u), S(u)$ be the random degree, strength of node $u \in N$, let $\tilde{d}(u)$, $\tilde{s}(u)$ be the corresponding expectations, and let $\mathbf{D}, \mathbf{S}, \bar{\mathbf{d}}, \bar{\mathbf{s}}$ be the associated n-vectors. Define $\bar{s}_T := \sum_{v\in N}\bar{s}(v)$ and $\bar{d}_T := \sum_{v\in N}\bar{d}(v)$.

We now define a *empirical* population version of the variance estimate:

Definition 11 *Fix $n > 1$ and let A and W be the edge and weight matrices from $\mathcal{G}_n$, the n-th random weighted network from the sequence in the setting of Theorem 4. Let $\mathbf{x}, \mathbf{y}$ be arbitrary n-vectors with positive entries. For nodes $u, v \in N$, define*

$$V_{uv}(\mathbf{x},\mathbf{y}) := \left(W_{uv} - f_{uv}(\mathbf{x},\mathbf{y})\right)^2, \qquad v_{uv}(\mathbf{x},\mathbf{y}) := \mathbb{E}\left\{V_{uv}(\mathbf{x},\mathbf{y})\middle|A_{uv} = 1\right\}.$$

Define the empirical *population variance estimator as follows:*

$$\kappa_*(\mathbf{x},\mathbf{y}) := \frac{\sum_{u,v:A_{uv}=1} v_{uv}(\mathbf{x},\mathbf{y})}{\sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{x},\mathbf{y})^2}$$

The estimator $\kappa_*(\mathbf{x},\mathbf{y})$ is called "empirical" because it depends on the random edge set E. Despite this, it has a deterministic bound, a fact which is part of Lemma 12. Throughout the remaining results, denote $\Theta := (\mathbf{D}, \mathbf{S}, \hat{\kappa}(\mathbf{D},\mathbf{S}))$ and $\theta_* := (\bar{\mathbf{d}}, \bar{\mathbf{s}}, \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}}))$, where the estimator $\hat{\kappa}$ is the estimator from Section 2.2.

Recall the definition of the asymptotic order of the average degree $\lambda_n := n\rho_n$, from Section 4.2.2 in the main text. With this and the conventions above, Lemma 12 establishes basic facts about the WSBM:

Lemma 12 *Fix $n > 1$, and let $\mathcal{G}_n$ be a random network generated by a WSBM. For all nodes $u, v \in N$, under Assumptions 5 and 6,*

(1) $\bar{d}(u) = \lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}}^0, \mathbf{P}[c(u)]\rangle$ *and* $\bar{s}(u) = \lambda_n\psi(u)\langle\tilde{\boldsymbol{\pi}}, \mathbf{H}[c(u)]\rangle$

(2) $m_-^2 \leqslant \bar{d}(u)/\lambda_n \leqslant m_+^2$ *and* $m_-^3 \leqslant \bar{s}(u)/\lambda_n \leqslant m_+^3$

(3) $m_- \leqslant \bar{d}_T/n\lambda_n \leqslant m_+$ *and* $m_-^2 \leqslant \bar{s}_T/n\lambda_n \leqslant m_+^2$

(4) $m_-^4/m_+^1 \leqslant r_{uv}(\bar{\mathbf{d}})/\rho_n \leqslant m_+^4/m_-^1$ *and* $m_-^6/m_+^2 \leqslant r_{uv}(\bar{\mathbf{s}})/\rho_n \leqslant m_+^6/m_-^2$

(5) $m_-^2/m_+^2 \leqslant f_{uv}(\phi,\psi) \leqslant m_+^2/m_-^2$ *and* $m_-^{10}/m_+^3 \leqslant f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant m_+^{10}/m_-^3$

(6) $0 \leqslant V_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2$

(7) $0 \leqslant \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant g(\eta, m_-, m_+)$ *where* g *is a deterministic function.*

(8) There exist global constants $0 < m_1 < m_2 < \infty$ *independent of* n *such that for any node set* $B \subseteq N$*,*

$$m_1|B|\rho_n \;\leqslant\; \mu(u, B|\bar{\mathbf{s}}),\; \sigma(u, B|\theta_*)^2 \;\leqslant\; m_2|B|\rho_n$$

Proof For (1), we have

$$\begin{aligned}\bar{s}(u) := \mathbb{E}S(u) &= \sum_{j=1}^{K}\sum_{v\in\mathcal{C}_j}\mathbb{E}W_{uv} = \sum_{j=1}^{K}\sum_{v\in\mathcal{C}_j}\rho_n r_{uv}(\phi)\mathbf{H}_{c(u)j} \\ &= \rho_n\sum_{j=1}^{K}\phi(u)n\tilde{\pi}_j\mathbf{H}_{c(u)j} = \lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}}, \mathbf{H}_{c(u)}\rangle\end{aligned}$$

An identical calculation yields the expression for $\bar{d}(u)$. The inequalities in (2) then follow from Assumption 5. For (3), we again apply Assumption 5 to the equation

$$\bar{s}_T = \sum_{i=1}^{K}\sum_{v\in\mathcal{C}_i}\bar{s}(u) = \sum_{i=1}^{K}n\lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}}, \mathbf{H}_i\rangle = n\lambda_n\tilde{\boldsymbol{\pi}}^T\mathbf{H}\tilde{\boldsymbol{\pi}}$$

An identical equation yields the inequality for $\bar{d}_T$. (2) and (3) directly yield the inequalities in (4). Note that Assumption 5 implies $m_-^2 \leqslant nr_{uv}(\phi), nr_{uv}(\psi) \leqslant m_+^2$, which yields the first inequality of (5). The second inequality of (5) follows from (4). For part (6), note that by Assumption 6 and the first inequality in (5), we have

$$W_{uv} := f_{uv}(\phi, \psi)\xi_{uv} \leqslant (m_+^2/m_-^2)\eta \tag{25}$$

The second inequality in (5) then yields (6). For part (7), recalling the definition of $\kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})$ from Definition 11, note first that, by (6), $0 \leqslant v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2$. Thus, by the second inequality (5),

$$0 \leqslant \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) := \frac{\sum_{u,v:A_{uv}=1} v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_{u,v:A_{uv}=1} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})} \leqslant \frac{(\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2}{m_-^{10}/m_+^3}$$

For part (8), recall that

$$\mu(u, B|\bar{\mathbf{s}}) := \sum_{v\in B} r_{uv}(\bar{\mathbf{s}})$$

The first inequality in (8) follows from applying the second inequality in (4). Similarly,

$$\sigma(u, B|\theta_*)^2 := \sum_{v\in B} r_{uv}(\bar{\mathbf{s}})f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})(1 - \tilde{r}_{uv}(\bar{\mathbf{d}}) + \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}))$$

The second inequality in part (8) follows from parts (4), (5), and (7). ■

The next lemma shows that, if the degrees and strengths of $\mathcal{G}_n$ are bounded around their expected values, the empirical estimate of variance is bounded around the conditional population estimate, and the coefficient of variation of $S_n(u, B)$ is bounded around its population value. Define $D_T := \sum_{u\in N} D(u)$ as the (random) total degree. Recall that λ_n is the asymptotic order the average of the *expected* degrees $\bar{d}_T$.

Lemma 13 *Fix $n > 1$. Suppose Assumption 5 holds. Define*

$$M(\mathbf{D}, \mathbf{S}) := \max_{u \in N} \left\{ |S(u) - \bar{s}(u)|, |D(u) - \bar{d}(u)| \right\}. \tag{26}$$

Then the following statements hold:

(1) There exists small enough $t > 0$ such that if $M(\mathbf{D}, \mathbf{S}) \leqslant \lambda_n t$,

$$\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| = \left| \frac{\sum_{u,v:A_{uv}=1} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_{u,v:A_{uv}=1} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + D_T \rho_n O(t)} \right| + \rho_n O(t)$$

(2) Fix a constant $\varepsilon > 0$ independent of n. Assume $|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})| \leqslant \varepsilon$. Then then there exists small enough $t > 0$ (not depending on ε) such that if $M(\mathbf{D}, \mathbf{S}) \leqslant t$, for all $B \subseteq N$, we have

$$\left| \frac{\mu(u, B|\Theta)}{\sigma(u, B|\Theta)} - \frac{\mu(u, B|\theta_*)}{\sigma(u, B|\theta_*)} \right| = \sqrt{|B|\rho_n} O(t)$$

Proof $M(\mathbf{D}, \mathbf{S}) \leqslant \lambda_n t$ implies there exists a n-vector $\mathbf{a}_t$ with components in the interval $[-1, 1]$ such that $S(u) = \bar{s}(u) + \lambda_n t a_t(u)$. Therefore, defining $\bar{a}_t := n^{-1} \sum_v a_t(v)$,

$$\begin{aligned}
r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}}) &= \frac{\{\bar{s}(u) + \lambda_n a_t(u)t\}\{\bar{s}(v) + \lambda_n a_t(v)t\}}{\bar{s}_T + n\lambda_n \bar{a}_t t} - \frac{\bar{s}(u)\bar{s}(v)}{\bar{s}_T} \\
&= \frac{\bar{s}_T\{\bar{s}(u)a_t(v) + \bar{s}(v)a_t(u) + \lambda_n a_t(u)a_t(v)t\}\lambda_n t - \bar{s}(u)\bar{s}(v)n\lambda_n \bar{a}_t t}{\bar{s}_T\{\bar{s}_T + n\lambda_n \bar{a}_t t\}} \\
&= \left\{ \frac{\bar{s}(u)a_t(v) + \bar{s}(v)a_t(u) + \lambda_n a_t(u)a_t(v)t - r_{uv}(\bar{s})n\bar{a}_t}{\bar{s}_T + n\lambda_n \bar{a}_t t} \right\} \lambda_n t
\end{aligned}$$

Using parts (2)-(4) of Lemma 12, for sufficiently small t we have

$$\left|r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}})\right| \leqslant \frac{2\lambda_n m_+^3 + \lambda_n t + \rho_n (m_+^6/m_-^2)n}{n\lambda_n m_-^2 - n\lambda_n t} \lambda_n t = \frac{2m_+^3 + t + (m_+^6/m_-^2)}{m_-^2 - t} \rho_n t$$

Therefore,

$$|r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}})| = \rho_n O(t) \tag{27}$$

as $t \to 0$. By a similar argument, $|r_{uv}(\mathbf{D}) - r_{uv}(\bar{\mathbf{d}})| = \rho_n O(t)$. It follows that

$$|f_{uv}(\mathbf{D}, \mathbf{S}) - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})| = \rho_n O(t). \tag{28}$$

Therefore, using Equations 27-28 and part (7) of Lemma 12,

$$\begin{aligned}
V_{uv}(\mathbf{D}, \mathbf{S}) &:= (W_{uv} - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&= (W_{uv} - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}))^2 + 2(W_{uv} - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}))(f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S})) \\
&\quad + (f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&= V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + 2V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})(f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S})) + (f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&\leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + \rho_n O(t) + \rho_n^2 O(t^2) = V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + \rho_n O(t)
\end{aligned}$$

Define the following:

$$V_T := \sum_{u,v:A_{uv}=1} V_{uv}(\mathbf{D},\mathbf{S}), \qquad \bar{V}_T := \sum_{u,v:A_{uv}=1} V_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}).$$

Since $D_T := \sum_{u\in N} D(u) = \sum_{u,v:A_{uv}=1} 1$, the above inequality implies that $V_T = \bar{V}_T + D_T\rho_n O(t)$. Define similarly:

$$g_T := \sum_{u,v:A_{uv}=1} f_{uv}(\mathbf{D},\mathbf{S})^2, \qquad \bar{g}_T := \sum_{u,v:A_{uv}=1} f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})^2.$$

Similar logic gives $g_T = \bar{g}_T + D_T\rho_n O(t)$. Finally, define $\bar{v}_T := \sum_{u,v:A_{uv}=1} v_{uv}(\bar{\mathbf{s}},\bar{\mathbf{d}})$. Then

$$\begin{aligned} \left|\hat{\kappa}(\mathbf{D},\mathbf{S}) - \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right| &= \left|\frac{V_T}{g_T} - \frac{\bar{v}_T}{\bar{g}_T}\right| = \left|\frac{\bar{V}_T + D_T\rho_n O(t)}{\bar{g}_T + D_T\rho_n O(t)} - \frac{\bar{v}_T}{\bar{g}_T}\right| \\ &= \left|\frac{\bar{V}_T + D_T\rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T}\left\{\bar{g}_T + D_T\rho_n O(t)\right\}}{\bar{g}_T + D_T\rho_n O(t)}\right| \\ &\leqslant \left|\frac{\bar{V}_T - \bar{v}_T}{\bar{g}_T + D_T\rho_n O(t)}\right| + \left|\frac{D_T\rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T} D_T\rho_n O(t)}{\bar{g}_T + D_T\rho_n O(t)}\right| \end{aligned}$$

Note that $\bar{v}_T/D_T$ and $\bar{g}_T/D_T$ are, each, by parts (5) and (6) of Lemma 12, bounded above and below by constants independent of A, t, and n. Therefore, dividing through by D_T,

$$\left|\frac{D_T\rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T} D_T\rho_n O(t)}{\bar{g}_T + D_T\rho_n O(t)}\right| \leqslant \frac{\rho_n O(t)}{\bar{g}_T/D_T + \rho_n O(t)} = \rho_n O(t)$$

This proves part 1. For part 2, first recall that $\mu(u,B|\Theta) \equiv \mu(u,B|\mathbf{S}) := \sum_{v\in B} r_{uv}(\mathbf{S})$. Therefore by Equation 27, we have

$$|\mu(u,B|\Theta) - \mu(u,B|\theta_*)| = \big|\sum_{v\in B} r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}})\big| = |B|\rho_n O(t) \tag{29}$$

Recall further that

$$\sigma(u,B|\Theta)^2 := \sum_{v\in B} r_{uv}(\mathbf{S}) f_{uv}(\mathbf{D},\mathbf{S})\left(1 - r_{uv}(\mathbf{D}) + \hat{\kappa}(\mathbf{D},\mathbf{S})\right)$$

Using some straightforward algebra and applying Equations 27-28, we have

$$\begin{aligned} \left|\sigma(u,B|\Theta)^2 - \sigma(u,B|\theta_*)^2\right| &= |B|\left(1 + \left|\hat{\kappa}(\mathbf{D},\mathbf{S}) - \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right|\right)\rho_n O(t) \\ &= |B|\rho_n O(t) \end{aligned} \tag{30}$$

where the second line follows from the assumption that $|\hat{\kappa}(\mathbf{D},\mathbf{S}) - \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})| \leqslant \varepsilon$. We will now bound $\sigma(u,B|\Theta)$ close to $\sigma(u,B|\theta_*)$ using Equation 30 and a Taylor expansion. Define the function $h(x,\sigma) := \sqrt{\sigma^2 + x}$. For fixed σ, a Taylor expansion around $x = 0$ gives

$h(x,\sigma) = \sigma + \sum_{k=1}^{\infty}(-1)^k \frac{x^k}{k!\sigma^{2k-1}}$. Setting $x = \sigma(u,B|\Theta)^2 - \sigma(u,B|\theta_*)^2$ and $\sigma = \sigma(u,B|\theta_*)$ and applying Equation 30, we obtain

$$\begin{aligned} \sigma(u,B|\Theta) &= h(x, \sigma(u,B|\theta_*)) \\ &= \sigma(u,B|\theta_*) + \sum_{k=1}^{\infty}(-1)^k \frac{|B|^k \rho_n^k O(t^k)}{k!\sigma(u,B|\theta_*)^{2k-1}} \end{aligned} \tag{31}$$

Part (8) of Lemma 12 implies that $\sigma(u,B|\theta_*) \asymp \sqrt{|B|\rho_n}$. Equation 31 therefore gives

$$\sigma(u,B|\Theta) = \sigma(u,B|\theta_*) + \sqrt{|B|\rho_n}O(t) \tag{32}$$

using Equations 29 and 32, we write

$$\left|\frac{\mu(u,B|\Theta)}{\sigma(u,B|\Theta)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| = \left|\frac{\mu(u,B|\theta_*) + |B|\rho_n O(t)}{\sigma(u,B|\theta_*) + \sqrt{|B|\rho_n}O(t)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| \tag{33}$$

As shorthands, define $\bar{\mu}_n := \mu(u,B|\theta_*)/|B|\rho_n$ and $\bar{\sigma}_n := \sigma(u,B|\theta_*)/\sqrt{|B|\rho_n}$. Part (8) of Lemma 12 implies that $\bar{\mu}_n, \bar{\sigma}_n \asymp 1$. Thus, using Equation 33 and dividing through by the appropriate factors,

$$\begin{aligned} \left|\frac{\mu(u,B|\Theta)}{\sigma(u,B|\Theta)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| &= \sqrt{|B|\rho_n}\left|\frac{\bar{\mu}_n + O(t)}{\bar{\sigma}_n + O(t)} - \frac{\bar{\mu}_n}{\bar{\sigma}_n}\right| \\ &= \sqrt{|B|\rho_n}O(t) \end{aligned}$$

This completes part 2. ■

The proof of Lemma 4 from the main text (below) makes use of Lemma 13 by showing that its assumption holds with high probability, for appropriate t.

C.1 Proof of Theorem 4

Throughout, we will sometimes suppress dependence on n for notational convenience. Recall that $A(u,B,\mathcal{G}) := S(u,B,\mathcal{G}) - \mu(u,B|\mathbf{S})$, the deviation of the CCME test statistic from its expected value under the continuous configuration model. Recalling that $\Theta := (\mathbf{D}, \mathbf{S}, \hat{\kappa}(\mathbf{D}, \mathbf{S}))$, define also the random Z-statistic

$$Z(u,B,\mathcal{G}|\Theta) := \frac{A(u,B,\mathcal{G})}{\sigma(u,B|\Theta)}. \tag{34}$$

Define the random p-value

$$P(u,B,\mathcal{G}|\Theta) := 1 - \Phi(Z(u,B,\mathcal{G}|\Theta)). \tag{35}$$

The random variable $P(u,B,\mathcal{G}|\Theta)$ is the random version of the p-value $p(u,B_n|\theta)$ obtained from the approximation in Equation (11). As a consequence of the Benjamini-Hochberg procedure, the event $\{U_\alpha(B_n,\mathcal{G}) = C_n\}$ will occur if

$$\begin{aligned} &P(u,B_n,\mathcal{G}_n|\Theta) \leqslant q\alpha, \quad \text{for all } u \in C_n, \quad \text{and} \\ &P(u,B_n,\mathcal{G}_n|\Theta) > q\alpha, \quad \text{for all } u \notin C_n, \end{aligned} \tag{36}$$

since by assumption $|C_n| > qn$. Let h be the density function of a standard-Normal. By a well-known inequality for the CDF of a standard-Normal, if $Z(u, B_n, \mathcal{G}_n|\Theta) > 0$,

$$P(u, B_n, \mathcal{G}_n|\Theta) \leqslant \frac{1}{Z(u, B_n, \mathcal{G}_n|\Theta)} h(Z(u, B_n, \mathcal{G}_n|\Theta)). \tag{37}$$

By symmetry, if $Z(u, B_n, \mathcal{G}_n|\Theta) < 0$, then

$$P(u, B_n, \mathcal{G}_n|\Theta) \geqslant 1 + \frac{1}{Z(u, B_n, \mathcal{G}_n|\Theta)} h(Z(u, B_n, \mathcal{G}_n|\Theta)). \tag{38}$$

We therefore analyze the concentration properties of $Z(u, B_n, \mathcal{G}_n|\Theta)$ and apply Inequalities 37 and 38 to show that for sufficiently large n, the event in Equation 36 occurs with high probability. We will focus on the first line of 36 first; the second is shown similarly. Recall that θ_* is the empirical population null parameters of $\mathcal{G}_n$, defined after Definition 11. For the derivation below we use the following shorthands: $Y \equiv S(u, B_n, \mathcal{G}_n)$, $\mu \equiv \mu(u, B_n|\mathbf{S}_n)$, $\sigma := \sigma(u, B_n|\Theta)$, $\bar{y} \equiv \mathbb{E}Y$, $\bar{\mu} \equiv \mu(u, B_n|\theta_*)$, and $\bar{\sigma} := \sigma(u, B_n|\theta_*)$. Note

$$\begin{aligned} Z(u, B_n, \mathcal{G}_n|\Theta) := \frac{Y-\mu}{\sigma} = \frac{Y-\bar{\mu}}{\bar{\sigma}} - \left(\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right) &= \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} + \frac{Y-\bar{y}}{\bar{\sigma}} - \left(\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right) \\ &\geqslant \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} - \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \end{aligned} \tag{39}$$

Define

$$\bar{z}(u, B_n|\theta_*) := \frac{\bar{y}-\bar{\mu}}{\bar{\sigma}} = \lambda_n \frac{\tilde{a}(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)}$$

where $\tilde{a}(u, B_n|\bar{\mathbf{s}})$ is the normalized population version of $A(u, B_n|\mathbf{S})$, as defined in Equation 13 from the main text. The definition above works with Equation 39 to produce the illustrative inequality

$$Z(u, B_n, \mathcal{G}_n|\Theta) \geqslant \bar{z}(u, B_n|\theta_*) - \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|. \tag{40}$$

Inequality 40 exemplifies that, if the right-hand terms vanish, $Z(u, B_n, \mathcal{G}_n|\Theta)$ can be approximated by a population version. Our analysis therefore reduces to bounding the right-hand order terms in probability.

Explicitly, consider that by part (8) of Lemma 12, there exists $m_2 > 0$ such that $\sigma(u, B_n|\theta_*)^2 \leqslant m_2 n\rho_n = m_2\lambda_n$. Combining this with the crucial assumption on $\tilde{a}(u, B_n)$ from line 14 from the main text, we have that for all $u \in C_n$,

$$\bar{z}(u, B_n|\theta_*) = \lambda_n \frac{\tilde{a}(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)} \geqslant \sqrt{\lambda_n}\frac{\Delta}{\sqrt{m_2}} \tag{41}$$

Therefore, the rest of the proof is mainly dedicated to showing that the final two terms in line (40) are $o_P(\sqrt{\lambda_n})$. This will imply that $Z(u, B_n, \mathcal{G}_n|\Theta) = \Omega_P(\sqrt{\lambda_n})$ and, using Inequality 37, that $\{P(u, B_n, \mathcal{G}_n|\Theta) \leqslant q\alpha,\ \forall\, u \in C_n\}$ has probability approaching 1.

Step 1: $|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}| = O_P(\sqrt{\log n})$

For $t > 0$, define the event

$$\mathcal{E}_1(t) := \left\{ \max_{u \in N} |S(u) - \bar{s}(u)|, \max_{u \in N} \left|D(u) - \bar{d}(u)\right| \leqslant \lambda_n t \right\} \tag{42}$$

Fix arbitrary $b > 0$ independent of all other quantities and define $t_n(b) := \sqrt{\frac{b \log n}{\lambda_n}}$. Note that $t_n(b) \to 0$ for any b, by the assumptions of the Theorem. Recall that $D_T := \sum_{u \in N} D(u)$, the (random) total degree. For notational convenience, let $E := \{\text{pairs } u, v : A_{uv} = 1\}$. By part 1 of Lemma 13, the event $\mathcal{E}_1(t_n(b))$ implies

$$\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| = \left| \frac{\sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + D_T \rho_n O(t_n(b))} \right| + \rho_n O(t_n(b)) \tag{43}$$

By Lemma 12 part (5),

$$0 \leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant (\eta m_+^2 / m_-^2 + m_+^{10} / m_-^3)^2.$$

Recall that $v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) := \mathbb{E} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})$, and that the edge weights that comprise the (upper-triangle of the) weight matrix W are independent. For a fixed adjacency matrix A, Bernstein's Inequality therefore gives

$$\mathbb{P}\left(\left| \sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \right| > \sqrt{b \log n} \,\middle|\, A \right) \leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_1 + \frac{2}{3} a_2 \sqrt{b \log n}} \right\} \tag{44}$$

Now by Lemma 12 part (6), $\sum_E f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 \geqslant D_T \frac{m_-^{10}}{m_+^3}$. Thus

$$\sum_E f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + D_T \rho_n O(t_n(b)) \geqslant D_T \frac{m_-^{10}}{m_+^3} / 2$$

for large enough n, since $\rho_n t_n(b) \to 0$. Therefore there exist constants $a_1, a_2 > 0$ depending only on m_+, m_-, and η such that

$$\mathbb{P}\left(\left| \frac{\sum_E V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{s}}, \bar{\mathbf{d}})^2 + D_T \rho_n O(t_n(b))} \right| > \sqrt{\frac{b \log n}{D_T}} \,\middle|\, A \right) \leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_1 + \frac{2}{3} a_2 \sqrt{\frac{b \log n}{D_T}}} \right\} \tag{45}$$

The above expression is conditional on a fixed adjacency matrix A. We now bound in probability the functionals of A on which the expression depends. It is easily derivable from the statement of the WSBM and Assumption 5 that there exist constants a_3, a_4 depending on m_+ and m_- such that $\mathbb{E}(D_T) = a_3 n \lambda_n$ and $\mathrm{Var}(D_T) = a_4 n \lambda_n$. Therefore, by another application of Bernstein's Inequality,

$$\mathbb{P}\left(|D_T - a_3 n \lambda_n| > \sqrt{n \lambda_n b \log n} \right) \leqslant 2 \exp\left\{ \frac{-2b \log n}{2a_4 + \frac{2}{3} \sqrt{\frac{b \log n}{n \lambda_n}}} \right\} \tag{46}$$

Applying this to inequality (45), the law of total probability gives

$$\mathbb{P}\left(\left|\frac{\sum_E V_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})-v_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{s}},\bar{\mathbf{d}})^2+D_T\rho_nO(t_n(b))}\right|>\sqrt{\frac{b\log n}{a_3n\lambda_n-\sqrt{n\lambda_nb\log n}}}\right)$$
$$\leqslant 2\exp\left\{\frac{-2b\log n}{2a_1+\frac{2}{3}a_2\sqrt{\frac{b\log n}{a_3n\lambda_n-\sqrt{n\lambda_nb\log n}}}}\right\}+2\exp\left\{\frac{-2b\log n}{2a_4+\frac{2}{3}\sqrt{\frac{b\log n}{n\lambda_n}}}\right\}=O(n^{-b}) \qquad (47)$$

for sufficiently large n. Along with Equation (43), this implies there exists a constant A_0 depending on parameter constraints such that

$$\mathbb{P}\left\{\left|\hat{\kappa}(\mathbf{D},\mathbf{S})-\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right|\leqslant A_0\left(\sqrt{\frac{b\log n}{n\lambda_n}}+\rho_nt_n(b)\right)\right\}\geqslant\mathbb{P}(\mathcal{E}_1(t_n(b)))-O(n^{-b}) \qquad (48)$$

for sufficiently large n. We now assess $\mathbb{P}(\mathcal{E}_1(t_n(b)))$. Note that for all $u\in N$, $\mathrm{Var}(S(u))=O(\lambda_n)$. Furthermore, recall from Inequality 25 (in the proof of Lemma 12) that $W_{uv}\leqslant m_+^2\eta/m_-^2$ for all $u,v\in N$. For fixed $b>0$, Bernstein's Inequality therefore gives, for any $u\in N$,

$$\mathbb{P}\left(|S(u)-\bar{s}(u)|>\sqrt{b\log n\lambda_n}\right)\leqslant 2\exp\left\{\frac{-2a_0b\log n}{2+\frac{2}{3}\sqrt{\frac{b\log n}{\lambda_n}}}\right\}, \qquad (49)$$

where a_0 is a constant independent of n. The constant a_0 may be chosen so that, similarly,

$$\mathbb{P}\left(\left|D(u)-\bar{d}(u)\right|>\sqrt{b\log n\lambda_n}\right)\leqslant 2\exp\left\{\frac{-2a_0b\log n}{2+\frac{2}{3}\sqrt{\frac{b\log n}{\lambda_n}}}\right\} \qquad (50)$$

Applying a union bound, equations (49) and (50) give

$$\mathbb{P}(\mathcal{E}_1(t_n(b)))\geqslant 1-2n\exp\left\{\frac{-2a_0b\log n}{2+\frac{2}{3}\sqrt{\frac{b\log n}{\lambda_n}}}\right\}-2n\exp\left\{\frac{-2a_0b\log n}{2+\frac{2}{3}\sqrt{\frac{b\log n}{\lambda_n}}}\right\}$$
$$=1-O(n^{-b+1}) \qquad (51)$$

for sufficiently large n. Returning to the inequality in (48), we therefore have

$$\mathbb{P}\left\{\left|\hat{\kappa}(\mathbf{D},\mathbf{S})-\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right|\leqslant A_0\left(\sqrt{\frac{b\log n}{n\lambda_n}}+\rho_nt_n(b)\right)\right\}\geqslant\mathbb{P}(\mathcal{E}_1(t_n(b)))-O(n^{-b})$$
$$\geqslant 1-O(n^{-b+1}) \qquad (52)$$

for sufficiently large n. Recall that by assumption, $\lambda_n/\log n\to\infty$. Thus $t_n(b)\to 0$, and

$$\sqrt{\frac{b\log n}{n\lambda_n}}+\rho_nt_n(b)=t_n(b)/\sqrt{n}+\rho_nt_n(b)\leqslant 1/\sqrt{n}=o(1).$$

Thus, Inequality 52 implies that

$$\mathbb{P}\left(\left|\hat{\kappa}(\mathbf{D},\mathbf{S})-\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right|\leqslant\varepsilon\right)\geqslant 1-O(n^{-b+1}), \qquad (53)$$

for sufficiently large n. For $\varepsilon > 0$, define the event $\mathcal{E}_2(\varepsilon) := \left\{\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| \leqslant \varepsilon\right\}$. By part 2 of Lemma 3, the event $\mathcal{E}_1(t_n(b)) \cap \mathcal{E}_2(\varepsilon)$ implies

$$\begin{aligned}\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| := \left|\frac{\mu(u, B_n|\mathbf{S})}{\sigma(u, B_n|\Theta)} - \frac{\mu(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)}\right| &= \sqrt{|B_n|\rho_n}O(t_n(b)) \\ &\leqslant \sqrt{\lambda_n}O(t_n(b)). \\ &= O(\sqrt{b\log n}) \qquad (54)\end{aligned}$$

Therefore, there exists a constant $A_2 > 0$ such that, by Inequalities 51 and 53,

$$\mathbb{P}\left(\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \leqslant A_2\sqrt{b\log n}\right) = 1 - O(n^{-b+1}) \qquad (55)$$

for sufficiently large n. This completes Step 1.

Step 2: $|\frac{Y-\bar{y}}{\bar{\sigma}}| = O_P(\sqrt{\log n})$.

Note that, as for Inequality 49, Bernstein's Inequality gives

$$\mathbb{P}\left(|S(u, B_n, \mathcal{G}_n) - \mathbb{E}S(u, B_n, \mathcal{G}_n)| > \sqrt{b\log n\lambda_n}\right) \leqslant 2\exp\left\{\frac{-2a_0 b\log n}{2 + \frac{2}{3}\sqrt{\frac{b\log n}{\lambda_n}}}\right\} \qquad (56)$$

By Lemma 12 part (8), there exists $m_2 > 0$ such that $\sigma(u, B_n|\theta_*)^2 \leqslant m_2\lambda_n$. Thus,

$$\left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| := \left|\frac{S(u, B_n, \mathcal{G}_n) - \mathbb{E}S(u, B_n, \mathcal{G}_n)}{\sigma(u, B_n|\theta_*)}\right| \geqslant \left|\frac{S(u, B_n, \mathcal{G}_n) - \mathbb{E}S(u, B_n, \mathcal{G}_n)}{m_2\sqrt{\lambda_n}}\right|,$$

so by Inequality 56, we have for sufficiently large n that

$$\mathbb{P}\left(\left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| \leqslant \sqrt{\frac{b\log n}{m_2}}\right) \geqslant 1 - O(n^{-b}). \qquad (57)$$

This completes Step 2.

We now recall inequality 40:

$$Z(u, B_n, \mathcal{G}_n|\Theta) \geqslant \bar{z}(u, B_n|\theta_*) - \left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|.$$

In step 1, we showed that there exists a constant A_2 depending only on the fixed WSBM model parameters such that for any fixed $b > 1$, for large enough n, $\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \leqslant A_2\sqrt{b\log n}$ with probability $1 - O(n^{-b+1})$. In step 2, we showed that there exists a constant m_2 depending only on the fixed WSBM model parameters such that for any fixed $b > 1$, for large enough n, $|\frac{Y-\bar{y}}{\bar{\sigma}}| \leqslant \sqrt{b\log n/m_2}$ with probability $1 - O(n^{-b})$. Recall furthermore from inequality 41 that $\bar{z}(u, B_n|\theta_*) \geqslant \Delta\sqrt{\lambda_n/m_2}$, where Δ is from condition 14 in the statement of the Theorem. We can therefore write that for any fixed $b > 1$, for large enough n,

$$Z(u, B_n, \mathcal{G}_n|\Theta) \geqslant \Delta\sqrt{\lambda_n/m_2} - \sqrt{b\log n/m_2} - A_2\sqrt{b\log n} = A_3\sqrt{\lambda_n} - A_4\sqrt{b\log n}$$

with probability at least $1 - O(n^{-b+1})$. Now, by assumption, $|C_n| \geqslant qn$. Therefore, using Inequality 37 and a union bound, we can write that for any fixed $b > 1$, for large enough n,

$$\max_{u \in C_n} P(u, B_n, \mathcal{G}_n|\Theta) \leqslant \exp\{-(A_3\sqrt{\lambda_n} - A_4\sqrt{b \log n})^2\} \tag{58}$$

with probability at least $1 - O(n^{-b+2})$. Note that for any fixed b, the right-hand-side of inequality 58 vanishes, due to the assumption that $\lambda_n/\log n \to \infty$. Thus, for $b > 2$, inequality 58 implies that for large enough n (now depending on choice of b), the event $\{P(u, B_n, \mathcal{G}_n|\Theta) \leqslant q\alpha,\ \forall\ u \in C_n\}$ has probability $1 - O(n^{-b+2}) \to 1$.

It can be similarly shown that the second half of the event in (36) has probability approaching 1. Instead of Inequality 40 we (similarly) derive

$$Z(u, B_n, \mathcal{G}_n|\Theta) \leqslant \bar{z}(u, B_n|\theta_*) + \left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| + \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \tag{59}$$

This is useful because if $u \notin C_n$, assumption (14) ensures that $\tilde{a}_n(u, B_n)\bar{\mathbf{s}} < -\Delta$, and hence

$$\bar{z}(u, B_n|\theta_*) := \frac{\bar{y} - \bar{\mu}}{\bar{\sigma}} = \lambda_n \frac{\tilde{a}(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)} \leqslant \lambda_n \frac{-\Delta}{\sigma(u, B_n|\theta_*)} \leqslant \sqrt{\lambda_n}\frac{-\Delta}{\sqrt{m}_1}$$

where the last inequality follows from part (8) of Lemma 12. Steps 1 and 2 therefore work to show that for any fixed $b > 1$, for large enough n,

$$Z(u, B_n, \mathcal{G}_n|\Theta) \leqslant -\Delta\sqrt{\lambda_n/m_2} + \sqrt{b \log n/m_2} + A_2\sqrt{b \log n} = A_3\sqrt{\lambda_n} - A_4\sqrt{b \log n}$$

With probability $1 - O(n^{-b+1})$. Inequality 38 then implies that

$$\mathbb{P}\left(\max_{u \notin C_n} P(u, B_n, \mathcal{G}_n|\Theta) \geqslant 1 - \exp\{-(A_3\sqrt{\lambda_n} - A_4\sqrt{b \log n})^2\}\right) \geqslant 1 - O(n^{-b+2}) \tag{60}$$

With reasoning identical to the result for $u \in C_n$, this implies that for any $b > 2$, for large enough $n(b)$, the event $\{P(u, B_n, \mathcal{G}_n|\Theta) > q\alpha,\ \forall\ u \notin C_n\}$ has probability at least $1 - O(n^{-b+2}) \to 1$. Applying a union bound to the event in (36) completes the proof. ■

C.2 Proof of Theorem 5

We will show that if the condition in (15) holds, then the condition in (14) from Theorem 4 holds when $B_n = C_n = C_{j,n}$ simultaneously across all $j \in \{1, 2, \ldots, K\}$. This involves representing (14) in terms of the model parameters when $B_n = C_n = C_{j,n}$. Specifically, we derive the normalized population deviation $\tilde{a}(u, C_{j,n}|\bar{\mathbf{s}}) := (\mathbb{E}S(u, C_{j,n}, \mathcal{G}_n) - \mu(u, C_{j,n}|\bar{\mathbf{s}}))/\lambda_n$. First, note that for any fixed $j \leqslant K$, part (1) of Lemma 12 gives

$$\sum_{v \in C_{j,n}} \bar{s}(v) \ = \ \lambda_n \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \cdot \sum_{v \in C_{j,n}} \psi(u) \ = \ n\lambda_n \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \tilde{\pi}_j$$

and thus

$$\bar{s}_T := \sum_{v \in N} \bar{s}(v) = \sum_{j=1}^{K} \sum_{v \in C_{j,n}} \bar{s}(v) = n\lambda_n \sum_{j=1}^{K} \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \tilde{\pi}_j = n\lambda_n \tilde{\boldsymbol{\pi}}^t \mathbf{H} \boldsymbol{\pi}.$$

Therefore, again applying part (1) of Lemma 12,

$$\begin{aligned}\mu(u, C_{j,n}|\bar{\mathbf{s}}) := \sum_{v \in C_{j,n}} r_{uv}(\bar{\mathbf{s}}) = \bar{s}(u) \sum_{v \in C_{j,n}} \frac{\bar{s}(v)}{\bar{s}_T} &= \bar{s}(u) \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \tilde{\pi}_j}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}} \\ &= \lambda_n \psi(u) \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_{c(u)} \rangle \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle \tilde{\pi}_j}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}}.\end{aligned}$$

Secondly,

$$\mathbb{E}S(u, C_{j,n}, \mathcal{G}_n) = \sum_{v \in C_{j,n}} \mathbb{E}W_{uv} = \sum_{v \in C_{j,n}} \rho_n r_{uv}(\psi) \mathbf{H}_{c(u)j} = \lambda_n \psi(u) \mathbf{H}_{c(u)j} \tilde{\pi}_j.$$

Thus,

$$\begin{aligned}\tilde{a}(u, C_{j,n}|\bar{\mathbf{s}}) &:= \frac{\mathbb{E}S(u, C_{j,n}, \mathcal{G}_n) - \mu(u, C_{j,n}|\bar{\mathbf{s}})}{\lambda_n} \\ &= \psi(u) \tilde{\pi}_j \left(\mathbf{H}_{c(u)j} - \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_{c(u)} \rangle \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}_j \rangle}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}} \right). \qquad (61)\end{aligned}$$

If $u \in C_{i,n}$, the expression in the parentheses from the right-hand-side of (61) is the i, j-th element of the matrix $\mathbf{H} - \mathbf{H} \tilde{\mathbf{\Pi}} \mathbf{H} / \tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}$, with $\tilde{\mathbf{\Pi}} := \tilde{\boldsymbol{\pi}} \tilde{\boldsymbol{\pi}}^t$. By Assumption 5, $\psi(u) \geqslant m_-$ for all $u \in N$ and $i \leqslant K$, and $\tilde{\pi}_j$ is fixed. Thus, (15) ensures that (14) holds when $C_n = C_{j,n}$, simultaneously for $j \leqslant K$. Assumption 5 also ensures that there exists $q > 0$ such that for all $j \leqslant K$ and $n > 1$, $|C_{j,n}| > qn$. This allows us to apply Theorem 4 to the sequences $B_n = C_n = C_{j,n}$, for each $j \leqslant K$. A union bound proves the result. ■

Appendix D. Further discussion of the CCME methodology

D.1 Variance in results due to initialization

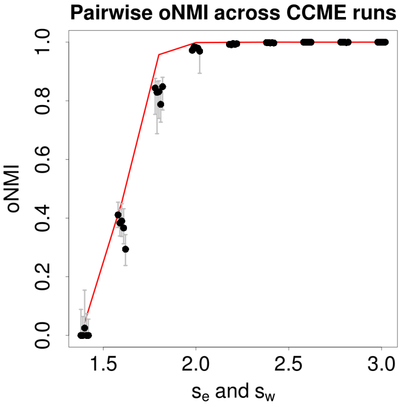


Figure 3: Stability tests for CCME.

Since CCME's initialization step is stochastic, it is natural to investigate the variance of results across many applications of CCME to the same network. To do so, we (randomly) selected five replications from each fixed parameter setting of the third experiment described in 6.2.1, in which both edge and edge weight signal levels are increased from low values. At each replication, we ran CCME thirty times, and recorded the pairwise Overlapping Normalized Mutual Information (oNMI, see Section 6) scores between the thirty resulting community sets. In Figure 3, we plot the median and 5%-95% quantile bars for the oNMI scores for each replication, and jitter the lines around the corresponding parameter value. It is clear from the plot that CCME results become more stable, and eventually non-varying, as community signal becomes stronger. This follows the straightforward intuition that when in-community edge/weight signal is weak, CCME finds spurious or imperfect communities, and will find *different* such communities when it is initialized with different random starting sets.

D.2 Cycles in Fixed Point Search

As remarked in Section 5.2, it is possible for the SCS algorithm to reach a stable sequence $C_1, \ldots, C_J$ that is traversed by the update $U_\alpha(\cdot, \mathcal{G})$. If this happens, we apply the following routine to re-start the algorithm, or return the union of the sequence:

1. If $C_i \cap C_{i+1} = \phi$ for any $i \leqslant J$, or if $C_J \cap C_1 = \phi$, terminate the iterations and do not extract a community.

2. Otherwise, define $C^* = \cup_{i=1}^J C_i$, and:

   (a) If C^* has been visited previously by SCS, extract C^* into $\mathcal{C}$.

   (b) Otherwise, re-initialize with C^*.

D.3 Filtering of $\mathcal{B}_0$ and $\mathcal{C}$

To filter through $\mathcal{B}_0$ and $\mathcal{C}$, we use an inference procedure based on a set-wise z-statistic, analogous to the node-set z-statistic presented in Section 4. Define $S(B) := \sum_{v \in B} S(v, B)$. Note that $S(B)$ has an easily derivable expectation and standard deviation under the continuous configuration model, which we denote (respectively) by $\mu(B|\theta)$ and $\sigma(B|\theta)$. We define the corresponding z-statistic and an approximate p-value by

$$z(B|\theta) := \frac{S(B) - \mu(B|\theta)}{\sigma(B|\theta)}, \qquad p(B|\theta) := 1 - \Phi(z(B|\theta))$$

Before initializing the SCS algorithm on sets in $\mathcal{B}_0$, we compute the p-value above for each member set, and remove any that are not significant at FDR level $\alpha = 0.05$. This greatly reduces the number of extractions CCME must perform, and reduces the probability of convergence on small, spurious communities.

We also use $z(B|\theta)$ to filter near-matches in $\mathcal{C}$, once all SCS extractions have terminated and empty sets have been removed. To do so, we require an overlap "tolerance" parameter $\tau \in [0, 1]$. First, we create a (non-symmetric) $|\mathcal{C}| \times |\mathcal{C}|$ matrix O with general element $O_{ij} := |C_i \cap C_j|/|C_i|$, which measures the proportional overlap of C_i into C_j. After setting the diagonal of O to zero, the filtering proceeds as follows:

1. Find indices $i \neq j$ corresponding to the maximum entry of O.

2. If $O_{ij} < \tau$, terminate filtering.

3. Remove either C_i or C_j from $\mathcal{C}$, whichever has the smaller $z(B|\theta)$.

4. Re-compute O, set its diagonal to zero, and return to step 1.

For all simulations and real-data analyses in this paper, we employed this algorithm with $\tau = 0.9$. To further decrease the computation time of CCME, as we proceed through $\mathcal{B}_0$, we skip sets that were formed from nodes that have already been extracted into $\mathcal{C}$. We find that, in practice, none of these adjustments harm CCME's ability to find statistically significant overlapping communities. Indeed, the simulation results mentioned in Section 6.2.2 show that CCME outperforms competing methods with overlap capabilities.

Appendix E. Extension to directed networks

Directed networks are networks for which edges and weights carry directionality, in the sense that nodes give and receive different quantities links and edge weight. Preserving directionality can in some applications be important for appropriate analysis of a network data set. The extension of the CCME method to directed networks is straightforward. First, both "in" and "out" observed degrees and strengths, denoted $\mathbf{d}^{\rightarrow} := \{\mathbf{d}_{in}, \mathbf{d}_{out}\}$ and $\mathbf{s}^{\rightarrow} := \{\mathbf{s}_{in}, \mathbf{s}_{out}\}$, are used in the null model. Let $A_{uv}^{\rightarrow}$ be the *ordered* entry of the adjacency matrix, corresponding to the edge from u *into* v, and similarly for weights $W_{uv}^{\rightarrow}$. Let $\mathbf{d}_{T,in}$ and $\mathbf{s}_{T,in}$ be the total in-degree and in-strength. Define

$$r_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow}) := \frac{d_{out}(u)d_{in}(v)}{d_{T,in}} \tag{62}$$

Define $\tilde{r}_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow}) := \min(1, r_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow}))$ and $f_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow}) := r_{uv}^{\rightarrow}(\mathbf{s}^{\rightarrow})/\tilde{r}_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow})$. Then the directed continuous configuration model is as follows:

1. $\mathbb{P}(A_{uv}^{\rightarrow} = 1) = \tilde{r}_{uv}^{\rightarrow}$ independently for all ordered node pairs $(u, v) \in N \times N$

2. For each ordered node pair (u, v), generate an independent random variable ξ_{uv} according to F, and assign edge weights by:

$$W_{uv}^{\rightarrow} = \begin{cases} f_{uv}^{\rightarrow}(\mathbf{s}^{\rightarrow}, \mathbf{d}^{\rightarrow})\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 0 \end{cases}$$

For directed networks, the formula for the variance parameter is

$$\hat{\kappa}^{\rightarrow}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow}) := \sum_{(u,v):A_{uv}=1} (W_{uv}^{\rightarrow} - f_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow}))^2 \,/ \sum_{(u,v):A_{uv}=1} f_{uv}^{\rightarrow}(\mathbf{d}^{\rightarrow}, \mathbf{s}^{\rightarrow})^2, \tag{63}$$

an equation that follows from the same method-of-moments logic presented in Section 2.2. Initialization, set updates, and overlap filtering proceed through use and testing of *out*-weights and *out*-weight sums. The proof of the corresponding Central Limit Theorem proceeds similarly to the proof of Theorem 2 with trivial modifications.

Appendix F. Simulation framework

Here we describe the benchmarking simulation framework used in Section 6. In Table 4, we list and name parameters controlling the network model:

Table 4: Simulation model parameters

n: Number of nodes in communities	n_b: Number of nodes in background
$m_{\max}$: Max community size	$m_{\min}$: Min community size
τ_1: Power-law for degree parameters	τ_2: Power-law for community sizes
k: Mean of degree parameter power-law	$k_{\max}$: Maximum degree parameter
s_e: Within-community edge signal	s_w: Within-community weight signal
o_n: Number of nodes in multiple communities	o_m: Number of memberships for overlap nodes
F: Distributions of edge weights	σ^2: Variance parameter for F
β: Power-law for strength parameters	

F.1 Simulation of community nodes

The framework is capable of simulating networks with or without background nodes. We first describe the simulation procedure without background nodes, i.e. with $n_b = 0$. Later, we describe how to simulate a network with background nodes, which involves a slight modification to the procedure in this subsection. Regardless of the presence of background nodes, the first step is to determine community sizes and node memberships.

F.1.1 Community structure and node degree/strength parameters

Here we describe how to obtain a cover $\mathcal{C} := \{C_1, \ldots C_K\}$ of n nodes. The following steps to obtain $\mathcal{C}$ are almost exactly as those from the LFR benchmark in Lancichinetti and Fortunato (2009), used extensively in Lancichinetti et al. (2011) and Xie et al. (2013):

1. Each of the o_n overlapping nodes will have o_m memberships. Let $n_m := n + o_n(o_m - 1)$ be the number of node *memberships* present in the network.

2. Draw community sizes from a power law with maximum value $m_{\max}$, minimum value $m_{\min}$, and exponent $-\tau_2$, until the sum of community sizes is greater than or equal to n_m. If the sum is greater than n_m, we reduce the sizes of the communities proportionally until the sum is equal to n_m.

3. Form a bipartite graph of community markers on one side and node markers on the other. Each community marker has number of empty node slots given by step (b), and each node has a number of memberships given by step (a). Sequentially pair node memberships and community node slots uniformly at random, without replacement, until every node membership is paired with a community.

With the community assignments in hand, simulation of the network proceeds according to the Weighted Stochastic Block Model as outlined in Section 6. We describe choices for particular components of this model in the following subsection.

F.1.2 Simulation of edges and weights

As described in Section 6, we set the $\mathbf{P}$ and $\mathbf{M}$ matrices to have diagonals equal to s_e and s_w (respectively, see Table 4), and off-diagonals equal to 1. We note that this homogeneity facilitates creating networks with overlapping communities. With variance in the diagonal of $\mathbf{P}$, for example, it would not be obvious with what probability to connect overlapping nodes that overlap to two of the same communities, simultaneously. It remains to obtain the strength and degree propensity parameters ψ and ϕ; we do so analogously to the simulation framework in Lancichinetti et al. (2011). We first draw ϕ from a power law with exponent τ_1, mean k, and maximum $k_{\max}$ (see Table 4). Next we set ψ by the formula $\psi(u) = \phi(u)^{\beta+1}$.

It is worth noting here that, under the model given below, the expected degree of node u is *approximately* $\phi(u)$ and the expected strength *approximately* $\psi(u)$. Therefore, heterogeneity/skewness in ϕ and ψ induce heterogeneity/skewness in the degrees and strengths of the simulated networks. However, by scaling ϕ and ψ, we can force the total expected degree and total expected strength of the simulated networks to exactly match ϕ_T and ψ_T, respectively. The scaling constants depend on $\mathbf{P}$ and $\mathbf{M}$ and are easily derivable from the model's generative algorithm (described in Section 4.2.1).

F.1.3 Parameter settings

Here we list the "default" settings of the simulation model, mentioned in Section 6. The following choices for parameters were made regardless of the simulation setting: $\tau_2 = -2$, $k = \sqrt{n}$, $k_{\max} = 3k$ (three settings which make the degree/strength distributions skewed and the network sparse), $\beta = 0.5$ (to induce a non-trivial power law between strengths and degrees), $\tau_1 = -1$, $m_{min} = n/5$, $m_{max} = 3m_{max}/2$ (settings which produce between about 3 and 7 communities per network with skewed size distribution), and $\sigma^2 = 1/2$. Other parameter choices are specific to the simulation settings described in Section 6.

F.2 Background node simulation

If $n_b > 0$, we generate a network with n community nodes, and then add n_b background nodes, generating all remaining edges and weights according to the continuous configuration null model introduced in the main text. First, we obtain node-wise parameters for all $n+n_b$ nodes, yielding vectors ϕ and ψ as in subsection F.1. In a simulated network without background, $\phi(u)$ and $\psi(u)$ are approximately $\mathbb{E}[d(u)]$ and $\mathbb{E}[s(u)]$, respectively. To ensure that this remains the case in a network for which background nodes are added after the simulation of community nodes, we must split up each $\phi(u)$ and $\psi(u)$ into community and background portions. A few other adjustments must also be made after the simulation of community nodes. To this end, define

- $N_C := \{1, \ldots, n\}$; $N_B := \{n+1, \ldots, n+n_b\}$ (community and background node sets)
- $\phi_{C,T} := \sum_{N_C} \phi(u)$; $\phi_{B,T} := \sum_{N_B} \phi(u)$ (target total degrees of community and background nodes)
- $\phi_C(u) := \frac{\phi_{C,T}}{\phi_T}\phi(u)$; $\phi_B(u) := \frac{\phi_{B,T}}{\phi_T}\phi(u)$ (target edge-counts between u and the community and background nodes)
- $\phi_{1,T} := \sum_{N_C} \phi_C(u)$; $\phi_{2,T} := \sum_{N_B} \phi_B(u)$ (target total degrees of community and background *subnetworks*)
- $d_C^o(u) := \sum_{v \in N_C} A_{uv}$; $d_B^o(u) := \sum_{v \in N_B} A_{uv}$ (observed edge-counts between u and the community and background nodes)

The above definitions exist analogously for the strength parameters ψ (replacing "d" with "s" where appropriate). The word "target" above indicates that we will set up the background simulation model so that these values are the approximate expected values of the graph statistics they represent.

F.2.1 Adjusted community-node simulation model

The only adjustment to be made to the simulation of community nodes, described in subsection F.1.2, is that the degree and strength parameters are set to a certain *fraction* of their original values. This accounts for the eventual addition of background nodes, where the remaining (random) part of each nodes degree and strength is to be simulated. So, the community-node simulation (if background nodes are to be added later) follows the process described in subsection F.1 with degree parameters $\{\phi_C(1), \ldots, \phi_C(n)\}$ and strength parameters $\{\psi_C(1) \ldots \psi_C(n)\}$.

F.2.2 Edges and weights for background

For the simulation of the background nodes (following the community nodes) our goal is to specify adjusted degree/strength parameters ϕ' and ψ' given the observed edge-sums $\{d^o_C(1), \ldots, d^o_C(n)\}$ and weight-sums $\{s^o_C(1), \ldots, s^o_C(n)\}$ from the community nodes. In what follows we describe this specification for ϕ' only; the specification for ψ' is exactly analogous. We first represent ϕ'_T, which we have yet to determine, into community and background totals:

$$\phi'_T = \phi'_{C,T} + \phi'_{B,T}$$

Since the background subnetwork has not yet been generated, we make the specification $\phi'(u) := \phi(u)$ for all $u \in N_B$, and hence $\phi'_{B,T} = \phi_{B,T}$ is known. To address $\phi'_{C,T}$, note that for each community node $u \in N_C$, $\phi'(u)$ may be represented similarly:

$$\phi'(u) = \phi'_C(u) + \phi'_B(u)$$

This reduces the problem of specifying $\phi'(u)$ to specifying $\phi'_C(u)$ and $\phi'_B(u)$. Since the community node subnetwork has already been generated, we set $\phi'_C(u) \leftarrow d^o_C(u)$. Next, recalling that $\phi_B(u) := \frac{\phi_{B,T}}{\phi_T}\phi(u)$, we make the specification $\phi'_B(u) := \frac{\phi_{B,T}}{\phi'_T}\phi(u)$ (which must be solved for via ϕ'_T, in the following). So, in total, we have

$$\phi'(u) = \begin{cases} d^o_C(u) + \frac{\phi_{B,T}}{\phi'_T}\phi(u), & u \in N_C \\ \phi(u), & u \in N_B \end{cases}$$

Therefore we can solve for ϕ'_T with the equation

$$\begin{aligned} \phi'_T &:= \sum_{u \in N_C \cup N_B} \phi'(u) \\ &= \sum_{u \in N_C} \left[d^o_C(u) + \frac{\phi_{B,T}}{\phi'_T}\phi(u) \right] + \sum_{u \in N_B} \phi(u) \\ &= d^o_{C,T} + \frac{\phi_{B,T}}{\phi'_T}\phi_{C,T} + \phi_{B,T} \end{aligned}$$

Where $d^o_{C,T} := \sum_{u \in N_C} d^o_C(u)$. The solution for ϕ'_T from this quadratic is

$$\phi'_T = \frac{\phi_{B,T} + d^o_{C,T}}{2} + \sqrt{\frac{(\phi_{B,T} + d^o_{C,T})^2}{4} + \phi_{C,T}\phi_{B,T}} \tag{64}$$

which then immediately gives the full vector ϕ'. We can now simulate the remaining edges in the network. Specifically, for each $u \in N_B$ and each $v \in N_C \cup N_B$, we simulate an edge according to

$$\mathbb{P}\left(A_{uv} = 1\right) = \frac{\phi'(u)\phi'(v)}{\phi'_T} \text{ independent across node pairs} \tag{65}$$

We solve for ψ' analogously. Then for each $u \in N_B$ and each $v \in N$, we simulate an edge weight according to

$$W_{uv} = \begin{cases} f_{uv}(\phi', \psi')\xi_{uv}, & A_{uv} = 1 \\ 0, & A_{uv} = 1 \end{cases}$$

where $\xi{\sim}F$, is as it was for the generation of the community node subnetwork.

The above simulation steps correspond precisely to the continuous configuration model with parameters $(\phi', \psi', F, \sigma)$. Some basic computational trials have shown that, for large networks, the solution for ϕ'_T is quite close to ϕ_T. Therefore, for each $u \in N_B$, $\mathbb{E}(d(u))$ is almost exactly $\phi(u)$, i.e. what it would be under the model in F.1.2, without background nodes. The same holds for the strengths and expected strengths. Together with equation 65, this implies the background nodes are behaving according to the continuous configuration model, even as they are a sub-network within a larger network with communities.

To illustrate these points, we simulated a sample network from the default framework with parameters $n = 5,000$, $n_b = 1,000$, $s_e = s_w = 3$, disjoint communities, and other parameters specified by F.1.3. These settings are akin to what was used in subsection 6 of the main text. First we plotted ϕ' and ψ' against the empirical strengths and degrees with lowess curves to check the match. Figure 4 shows the fit is essentially linear. Second, for

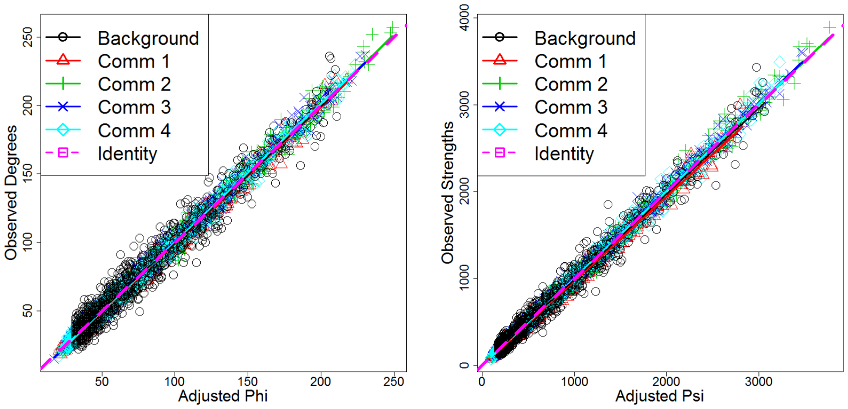


Figure 4: Empirical degrees/strengths vs. adjusted parameters for the example network

each node $u \in N$ and for each node block B (either a true community or the background node set) we may calculate an empirical z-score for $S(u, B, \mathcal{G})$, as described in subsection 4.1 of the main text. The z-score for $S(u, B, \mathcal{G})$ is a measure of connection significance, with respect to the continuous configuration model (and also modularity, see Section 4.2.3) between u and B. Let K be the number of true communities in the network. For each $i, j = 1, \ldots, K+1$, where $K+1$ is the index of the background node block, we computed the empirical average of z-statistics between nodes u from node block i the node block B corresponding to index j. Theses empirical averages can be arranged in a $(K+1) \times (K+1)$ matrix showing the average inter-block connectivities of the network. In Figure 5 we display a visualization of this matrix, which shows preferential connection within communities, and roughly null connection between the background nodes and all blocks.

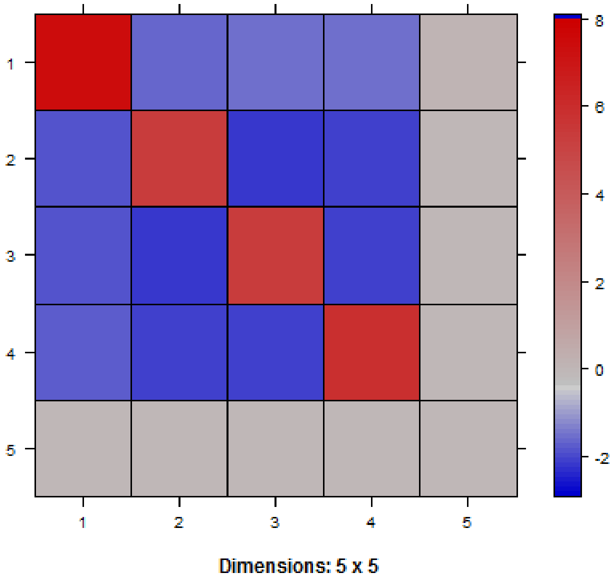


Figure 5: Average empirical z-statistics between nodes and node blocks

F.3 Additional simulation result displays

Here we display the results from our first simulation setting (described in Section 6.2.1) in plots with un-transformed y-axes. Figure 6 reveals that SLPAw seems to respond to increasing community signal, whereas WeightedGT seems to plateau, at least on this range of our parameters.

References

Christopher Aicher, Abigail Z Jacobs, and Aaron Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, page cnu026, 2014.

Reid Andersen, David F Gleich, and Vahab Mirrokni. Overlapping clusters for distributed computation. In *Proceedings of the Fifth ACM International Conference on Web Search*

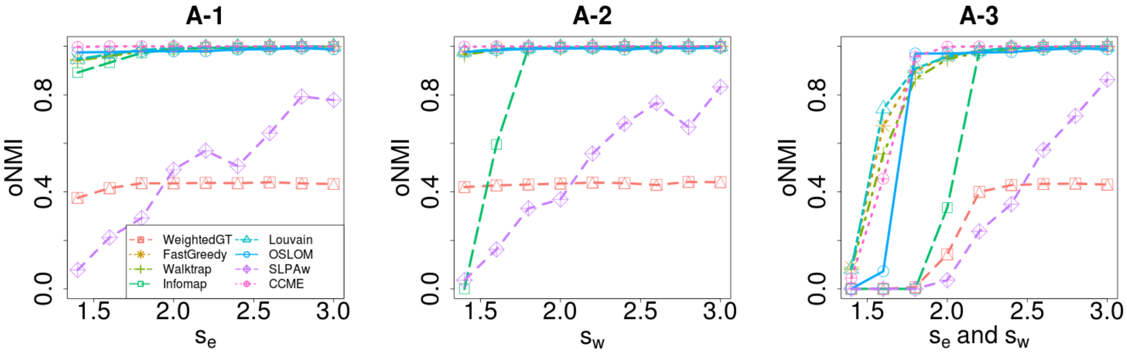


Figure 6: Simulation results described in Section 6.2.1, with un-transformed oNMI scores. Legend refers to all plots.

and Data Mining, pages 273–282. ACM, 2012.

Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.

Edward A Bender. The asymptotic number of non-negative integer matrices with given row and column sums. *Discrete Mathematics*, 10(2):217–223, 1974.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological)*, pages 289–300, 1995.

Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50): 21068–21073, 2009.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

Irineo Cabreros, Emmanuel Abbe, and Aristotelis Tsirigos. Detecting community structures in hi-c genomic data. In *Information Science and Systems (CISS), 2016 Annual Conference on*, pages 584–589. IEEE, 2016.

Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002a.

Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002b.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

Amin Coja-Oghlan and André Lanka. Finding planted partitions in random graphs with general degree distributions. *SIAM Journal on Discrete Mathematics*, 23(4):1682–1714, 2009.

Gabor Csárdi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005 (09):P09008, 2005.

Nurcan Durak, Tamara G Kolda, Ali Pinar, and C Seshadhri. A scalable null model for directed graphs matching all degree distributions: In, out, and reciprocal. In *Network Science Workshop (NSW), 2013 IEEE 2nd*, pages 23–30. IEEE, 2013.

Ming Fan, Ka-Chun Wong, Taewoo Ryu, Timothy Ravasi, and Xin Gao. Secom: A novel hash seed and community detection based-approach for genome-scale protein domain identification. *PLoS ONE*, 7:e39475, 06 2012.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

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

Abigail Z Jacobs and Aaron Clauset. A unified view of generative models for networks: models, methods, opportunities, and challenges. *arXiv:1411.4070*, 2014.

David Kahle and Hadley Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013. URL `http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf`.

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

Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3): 033015, 2009.

Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, Santo Fortunato, et al. Finding statistically significant communities in networks. *PloS One*, 6(4):e18961, 2011.

Rocco Langone, Carlos Alzate, and Johan AK Suykens. Modularity-based model selection for kernel spectral clustering. In *The 2011 International Joint Conference on Neural Networks*, pages 1849–1856. IEEE, 2011.

Jure Leskovec et al. Stanford network analysis project, 2010. URL `http://snap.stanford.edu`.

David Lusseau and Mark EJ Newman. Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London B: Biological Sciences*, 271(Suppl 6):S477–S481, 2004.

Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2): 167–256, 2003.

Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004a.

Mark EJ Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004b.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

Tiago P Peixoto. Model selection and hypothesis testing for large-scale network models with overlapping groups. *Physical Review X*, 5(1):011033, 2015.

Tiago P Peixoto. Nonparametric weighted stochastic block models. *arXiv:1708.01432*, 2017.

John Platig, Peter Castaldi, Dawn DeMeo, and John Quackenbush. Bipartite community structure of eqtls. *arXiv:1509.02816*, 2015.

Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms Applications*, 10(2):191–218, 2006.

Mason A Porter, Jukka-Pekka Onnela, and Peter J Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

Jörg Reichardt and Stefan Bornholdt. Clustering of sparse data via network communitiesa prototype study of a large online market. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06016, 2007.

Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

Shaghayegh Sahebi and William W Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web, RSWEB*, 2011.

Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, volume 5, pages 76–84. SIAM, 2005.

James D Wilson, Simi Wang, Peter J Mucha, Shankar Bhamidi, Andrew B Nobel, et al. A testing based extraction algorithm for identifying significant communities in networks. *The Annals of Applied Statistics*, 8(3):1853–1891, 2014.

Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *IEEE 11th International Conference on Data Mining*, pages 344–349. IEEE, 2011.

Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45 (4):43, 2013.

Liu Xin, E Haihong, Junde Song, Meina Song, and Junjie Tong. Book recommendation based on community detection. In *Pervasive Computing and the Networked World*, pages 364–373. Springer, 2014.

Xiaoran Yan, Cosma Shalizi, Jacob E Jensen, Florent Krzakala, Cristopher Moore, Lenka Zdeborová, Pan Zhang, and Yaojia Zhu. Model selection for degree-corrected block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(5):P05007, 2014.

Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007.

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

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, pages 2266–2292, 2012.