Generative models for two-ground-truth partitions in networks

Lena Mangold [*] and Camille Roth

Computational Social Science Team, Centre Marc Bloch, Friedrichstr. 191, 10117 Berlin, Germany;
Centre national de la recherche scientifique (CNRS), 3 rue Michel-Ange, 75 016 Paris, France;
and Centre d'Analyse et de Mathématique Sociales (CAMS), École des hautes études en sciences sociales (EHESS),
54 Bd Raspail, 75006 Paris, France

(Received 8 February 2023; revised 4 July 2023; accepted 9 October 2023; published 13 November 2023)

A myriad of approaches have been proposed to characterize the mesoscale structure of networks most often as a partition based on patterns variously called communities, blocks, or clusters. Clearly, distinct methods designed to detect different types of patterns may provide a variety of answers' to the networks mesoscale structure. Yet even multiple runs of a given method can sometimes yield diverse and conflicting results, producing entire landscapes of partitions which potentially include multiple (locally optimal) mesoscale explanations of the network. Such ambiguity motivates a closer look at the ability of these methods to find multiple qualitatively different "ground truth" partitions in a network. Here we propose the stochastic cross-block model (SCBM), a generative model which allows for two distinct partitions to be built into the mesoscale structure of a single benchmark network. We demonstrate a use case of the benchmark model by appraising the power of stochastic block models (SBMs) to detect implicitly planted coexisting bicommunity and core-periphery structures of different strengths. Given our model design and experimental setup, we find that the ability to detect the two partitions individually varies by SBM variant and that coexistence of both partitions is recovered only in a very limited number of cases. Our findings suggest that in most instances only one—in some way dominating—structure can be detected, even in the presence of other partitions. They underline the need for considering entire landscapes of partitions when different competing explanations exist and motivate future research to advance partition coexistence detection methods. Our model also contributes to the field of benchmark networks more generally by enabling further exploration of the ability of new and existing methods to detect ambiguity in the mesoscale structure of networks.

DOI: 10.1103/PhysRevE.108.054308

I. INTRODUCTION

Network structure is frequently characterized at the mesoscale level by the configuration of what is broadly denoted as "communities"—groupings of nodes that display some sort of similarity in terms of their connectivity in the network. Networks may exhibit a wide variety of mesoscale structures, such as densely connected or cohesive clusters, assortative or disassortative communities, core-periphery structures, equivalence classes, or combinations thereof [1,2]. In turn, there is often more than one scientifically plausible way to divide the nodes of a real-world network, as demonstrated, for instance, by the coexistence of both cohesive clusters and core-periphery structures in multiple cases [3,4].

Clearly, methods designed to identify distinct types of mesoscale structures yield different partitions for a given network. Perhaps more interestingly, results produced by different algorithms aimed at identifying one specific type of mesoscale structure may still vary considerably for the same network. A commonly studied empirical example is the Karate Club (KC) network, a friendship network of 34 members of a sports club which split into two new clubs after a fall-out between its members [5]. While the existing literature has repeatedly produced a partition of two cohesive groupings similar in terms of node membership to the division caused by the split of the club [6,7], variability in what is detected as the optimal partition of this network has been demonstrated in terms not only of community membership of nodes [6] but also of the total number of communities recovered [8–10]. Additionally, other types of mesoscale structures can be detected as plausible explanations for the KC network [11], including a core-periphery-type structure of leaders and followers.

Competing explanations of mesoscale structure in real networks, such as the KC example, motivate a further exploration of ambiguity on this scale; perhaps the reason for conflicting results is that multiple qualitatively different "ground truths" and partitions were responsible for the generative process of a network and its mesoscale configuration [12]. In fact, recent work on stochastic block models (SBMs), which have become increasingly popular for mesoscale network description, has emphasized the importance of exploring the variability of the entire partition landscapes that they return, instead of forcing a global consensus from a distribution of partitions (i.e., choosing one among many by maximizing some objective) [11].

[*] lena.mangold@cmb.hu-berlin.de, she/her/hers

A further important phenomenon in the context of mesoscale variability is that of detectability limits, where known structures are no longer detected due to lacking signal strength and which have been shown to exist due to the presence of phase transitions in networks generated by SBMs [13,14].

On the whole, mesoscale variability may thus stem both from the intrinsic ambiguity in the generative processes of the network and from the stochastic ambiguity of the generative blockmodel. In this paper we are generally interested in appraising the accuracy of mesoscale structure detection under ambiguity constraints. We choose (1) to start from the KC example as one of the simplest configurations possessing jointly a *core-periphery* and a *bicommunity* (two equally sized communities) structure, and (2) to rely on SBMs as a theoretical framework accommodating many types of mesoscale structures, beyond clusters and including the two above ones, and that may further be used not only for generating but also for detecting mesoscale structures. More specifically, we are motivated to explore the ability of SBMs to detect certain structures when we implicitly introduce some level of measurable ambiguity into the mesoscale. We propose a framework for a generative benchmark model, the stochastic cross-block model (SCBM), which can have such ambiguity built into its mesoscale structure by allowing for two qualitatively distinct partitions (i.e., two "ground truths") to be planted into the same network. Using this framework, the two partitions are defined respectively through block matrices that specify the connectivity within and between blocks which—similar to the planted partition model [15]—facilitates the analysis of SBMs (or any community detection algorithm) for varying strengths of the planted mesoscale structures. We use our framework to plant two qualitatively different partitions into a synthetic network and try to recover them using two SBM variants. In this way, we analyze the ability of SBMs to detect two competing structures present in a network, appraising both the extent to which each partition is recovered individually as well as the successful detection of the *coexistence* of both partitions: the appearance of two implicitly planted structures within the posterior distribution of inferred partitions of a given graph.

This paper is structured as follows. In Sec. II we provide some background on conflicting explanations of mesoscale structure in networks (Sec. II A), the interplay between ambiguity and detectability of block structures (Sec. II B), as well as a more general overview of the existing literature on SBMs (Sec. II C) and generative benchmark models of mesoscale structures (Sec. II D). We then introduce our model framework in Sec. III, covering the derivation of the model in the two-partition case and two variants of generative processes of edge placement in the network. In Sec. IV we illustrate a use case for our model in form of a set of simulations, the results of which we discuss in Sec. V. We summarize our main results in Sec. VI and briefly touch on possible future work that could extend our simulations and the model itself.

II. BACKGROUND

A. Community detection and partition landscapes

Community detection often adopts a clustering perspective and focuses on cohesive communities, which denote groups of nodes more densely connected to other nodes of the same group than to nodes in other groups [16], as opposed to other types of meso-level structures more generally [2]. Corresponding methods to identify community structure are designed to perform best with specific types of data and networks [12], and many authors have chosen distinct routes to optimize for the most plausible partition (see [17] for a review of different methods). Existing algorithms therefore have at their basis a multitude of measures, such as modularity [7], spectral properties [18], generative models [19], betweenness centrality [6], or information-theoretic methods [20], which is only one of the causes for the diversity in results from algorithms that use different approaches.

Similarly, the detection of core-periphery structure—the division of a network into a well-connected, cohesive core group and a sparsely connected peripheral group, first rigorously formulated by Borgatti and Everett [21]—has been approached in a number of different ways [22], including methods based on edge density [3,21,23], path length [23,24], and generative network models [4,25]. Some works also explore the coexistence of community and core-periphery structures in a nested way, in particular in the form of communities that exhibit core-periphery structures internally [3,26–28].

The diversity in methods for similar node aggregation tasks unsurprisingly results in a diversity in partitions returned by such methods. While most community detection algorithms of the 20th century (from cliques to k-cores through CONCOR [29–31]) as well as the Girvan-Newman algorithm [32] popular in the 2000s were deterministic, this challenge has been further exacerbated by the advent of approaches whose results are nondeterministic by nature, such as Louvain [9] and SBM [19,33,34]. In that case, even multiple results yielded by one given community detection algorithm may not clearly indicate a consensus partition, leading to a partition selection problem on top of the above-mentioned issue around model selection. In existing work, this has been addressed by finding some kind of consensus in a distribution of partitions to identify an optimal partition, for example, by averaging over results from multiple runs of the same algorithm [35,36].

Such consensus-seeking methods run into problems, however, when multiple partitions that are close to the optimum are qualitatively different from each other, revealing the need for considering multiple local consensus partitions that may provide different, similarly likely explanations to the network structure at hand [11]. The issue of multiple locally optimal partitions was also addressed by Peel *et al.* [12], who demonstrated that many real-world networks have multiple plausible (high-likelihood) partitions and that different sets of node metadata may correlate with different aspects of the structure of the network. These recent results suggest that by accommodating for a diversity of ground truths in the generative process, the stochastic nature of some community detection methods is, in fact, not only an issue to deal with but a feature of these methods. In this work, we propose a model that accepts any combination of two partitions, including structures that somewhat resemble nested core-periphery pairs (as in Sec. IV), yet without being limited to these specific structures. We also aim to extend the literature by proposing a model that ensures the consistency with two distinct planted

structures and thus allows for the exploration of networks in which connectivity patterns induced by multiple block structures are satisfied (i.e., the coexistence of multiple structures) and the extent to which both are detected and detectable.

B. Community detectability and ambiguity

The recent focus on the importance of analyzing the variability of partition distributions calls for an exploration of what we will call mesoscale ambiguity. In an effort to identify prototypical network partitions representative of various regions of an entire partition landscape Kirkley and Newman [37] aimed at introducing (and detecting) what they call "ambiguity" on the mesoscale of synthetic networks. By specifying such ambiguity through certain edge probabilities between blocks in an SBM (see Sec. II C for a review of SBMs) they demonstrated the ability of their model to detect a set of representative partitions which identify different aspects of the introduced ambiguity.

This specific example of ambiguity calls for an investigation of the distinction between truly ambiguous block structures on the one hand, and weak or noisy signals which prevent algorithms from correctly detecting mesoscale structures. The stochastically generated SBM ensemble may exhibit some variability in their block structure, but if only one "ground truth" partition is planted, are qualitatively different recovered partitions the result of real ambiguity or merely of detectability issues? And should the detection of partitions which could not have been generated from the planted model (i.e., that lie outside of the distribution of possible networks with the specified parameters) be viewed as a failure of the partitioning algorithm rather than successfully recovered ambiguity?

To distinguish between the correct recovery of some type of ambiguity on the mesoscale and the inability of a community detection algorithm to identify the true partition due to noise that is "blurring" the signal of the block structure, we need to provide some understanding of the detectability phase transitions that have been demonstrated to exist in community detection. Overall, it has been shown that the detectability of block structure in networks depends on the overall density of the network, the difference between the connectivity of the blocks, as well as the number of blocks (see [38,39] for extensive reviews). When the structural signal in a network exists but is too weak or too noisy, it becomes impossible for community detection algorithms to identify such structures. At a certain phase transition, algorithms will mistake a network for a random graph when the structural traces of underlying communities are not sufficiently tangible in the actual network. Prior work on the detectability of modules in network has shown, both analytically as well as heuristically, the existence [40] and positions of such phase transitions, notably for spectral community detection methods [14] and for methods using Bayesian maximum-likelihood [41]. Much of this early work on phase transitions focused on the symmetric case of the traditional (Poisson degree-distributed) SBM. Since then, others have worked on networks with heterogeneous node degree distributions and have argued for the existence of phase transitions in such cases [42] and demonstrated that heterogeneity in networks facilitates the detection of communities in the case of modularity maximization [43]. While many efforts have gone into the appraisal of phase transitions for community detection for decisive (albeit sometimes noisy) structures, and others have demonstrated that such detectability thresholds do not exist in core-periphery structures [4], little work has focused on the limits of SBMs cases where some level of ambiguity is introduced specifically. Exploring this further seems particularly important, since the application of community detection methods is primarily aimed at real-world networks, which arguably exhibit more "ambiguity" and for which the possible existence of multiple locally optimal solutions has been demonstrated repeatedly (see above). As mentioned previously, empirical networks have also been shown to have different types of mesoscale structures all at once, complicating the issue even further.

Overall, the question around a clear differentiation between the issues of detectability of certain block structures and "true" ambiguity in the sense of multiple different ground truths appears challenging and is—to the best of our knowledge—an open research question, that carries with it the question of how such ambiguity can be described. Owing to the lack of a clear definition, we from now on characterise *ambiguity* as the simultaneous existence of multiple planted partitions in one single network. We denote this simultaneous existence of implicit structures by *coexistence* of structures, where the network's connectivity aligns consistently and concurrently with the connectivity of the said structures.

C. Stochastic block models

In this work, we exploit the features of SBMs twofold. On the one hand, we use SBMs for *generating* synthetic networks with planted mesoscale structure. On the other hand, we explore the issue of ambiguity in mesoscale structure in networks by fitting SBMs to synthetic graphs and thus using SBMs as a way of *detecting* mesoscale structures. Bayesian inference methods, such as SBMs, are especially suited for the exploration of partition distributions due to their stochastic nature.

SBMs originate in mathematical sociology, where they built on the concept of node similarity expressed through equivalent connectivity patterns of *blocks* of nodes, coining the term *block modeling* for the grouping of such nodes [44–46]. Early strict notions of this type of equivalence were later relaxed in form of *stochastic equivalence*, which holds for nodes that connect to other node sets with the same probability [33]. The latter work also manifested the first appearance of *stochastic block models*, generative models that create networks (or entire distributions over networks) by first dividing nodes into blocks and then placing edges between node pairs with a probability depending solely on the block membership of each node.

The SBM is therefore an extension of the simple random graph model, where constant edge probabilities are specific to block pairs rather than being the same for the entire network. An SBM takes as parameters (a) a block membership vector, with entries indicating the block membership of each node, and (b) a square connectivity matrix of size equal to the number of blocks, whose elements indicate the probability of a connection between the respective blocks (or within a

block for the elements on the diagonal). Since their first appearance [33,44–46], SBMs have been repurposed repeatedly to function as a baseline model for addressing the community detection issue as an inference problem [19,34,47]. It is increasingly popular in theoretical and applied network science research, partly due to its flexibility grounded in a relatively general definition of what it means for nodes to be similar i.e., to belong to a block or a community. The idea is that one can "reverse" the generative process of an SBM for the purpose of block structure detection: statistical inference methods can be used to fit SBMs to network data, to recover the parameters of the model (essentially block memberships) that offer the most likely explanation of the generative processes of a network.

Famously, edge placement between two nodes in the "traditional" SBM only depends on the nodes' block assignment. It therefore does not resemble the structure of many real networks, which tend to have heterogeneous node degree distributions. One way to model degree heterogeneity is by adding "degree correction" into the SBM, through which edge placements also depend on the respective degree of each node. Using the degree-corrected version as the generative model assumed in the process of detecting block structure considerably improved the ability of SBMs to pick up community structures in real networks [19]. More SBM extensions have since been developed, including hierarchical [48], overlapping [49,50], and multilayer [51] variants, many of which were demonstrated to be an improvement in the goodness of fit for certain types of networks, compared even to the degree-corrected version. Others have exploited the flexibility of SBMs (in terms of the types of mesoscale structures that can be detected) to demonstrate the diversity of core-periphery structures in real networks [25]. In the existing SBM literature, most work focuses on the recovery of a single partition that is identified by optimizing some model selection criterion; however, some recent work has gone beyond the single-partition approach and has emphasized the need to consider the entire partition landscape returned by SBMs [11,12,37]. In this work, we intend to contribute to this particular subfield of the SBM literature, by drawing attention to the existence and detectability of more than one planted structure in a single network.

One existing strand of work within the SBM literature that is particularly relevant in the context of planting multiple ground truth structures is the mixed membership SBM (MMSBM) [49]. This model allows nodes to belong to multiple blocks to varying degrees, expressed by mixed membership vectors assigned to each node, the elements of which denote the probabilities of the node belonging to the different blocks. In an MMSBM, nodes may therefore embody connectivity patterns from more than one block at a time, making it relevant to our case of planting multiple partitions. There is a certain conceptual similarity between the MMSBM and our method, and the MMSBM can be shown to be equivalent to our method in some cases; however, we will see later that an MMSBM-based approach to the issue has significant limitations that can be overcome with our method.

D. Generative benchmark models

As opposed to real-world networks whose exact generative processes are not known, synthetic networks are a natural choice to plant a specific structure and serve as a benchmark, in particular for appraising the success of a certain method in recovering various mesoscale structures, including communities. Such benchmark frameworks allow for certain mesoscale structures to be "built into" a synthetic network, on which the performance of algorithms can be tested by measuring the extent to which the predefined structure is successfully recovered. In general, one or several parameters may be adjusted to explore the potential limits of an algorithm and to imitate the features of certain types of real networks. One of the earliest such models is the Girvan-Newman (GN) benchmark [32], a network of 128 nodes divided into four equally sized groups and relying on one parameter controlling intergroup connectivity strength through the external (out-community) degree of nodes.

To overcome some of its shortcomings, such as its general inflexibility, small size, and unrealistic features, Lancichinetti and Fortunato [52] proposed a benchmark accounting for heterogeneous degree and community size distributions. Other existing benchmarks allow for the specification of the within- and between-group connectivity through the use of SBMs, such as the planted partition model [15] which has been extended to other special cases including multilayer networks [53]. In general, the aim of generative benchmark models is to resemble features of empirical networks, and while many of the existing benchmarks account for one or several such features, the ambiguity in mesoscale structures has as yet been neglected. Our contribution is to focus on this particular aspect and to complement single ground truth benchmarks with a framework that generates networks with multiple built-in ground truths.

III. MODEL FRAMEWORK

We propose a generative network model whose parameters aim to simultaneously respect two partitions: edges are placed between node pairs in a way such that the resulting network exhibits a block structure that takes into account each of these two partitions at the same time. We discuss later how this framework can be extended to more complex structures with more than two partitions, but we focus on the simple two-partition case in the majority of this work.

The difficulty in generating a network that exhibits two coexisting structures primarily lies in connecting node pairs in a way that is consistent with the connectivity patterns of *both* planted structures. The probability of placing an edge between each node pair must depend on the block memberships of each node in each of the planted structures. We thus aim to design a generative process that specifies the appropriate probabilities for the desired connectivity patterns. An obvious choice would be a constrained MMSBM, in which nodes are members of two blocks with equal probability. It turns out that finding the appropriate normalization constants to make the additive edge probabilities of the MMSBM consistent with the two planted structures requires extra calculations that we do not need if we implement a simpler single-membership SBM approach that considers the overlaps of the blocks as simple blocks. In the following, we first lay out the outline of the method, explain the limitations of the MMSBM-based

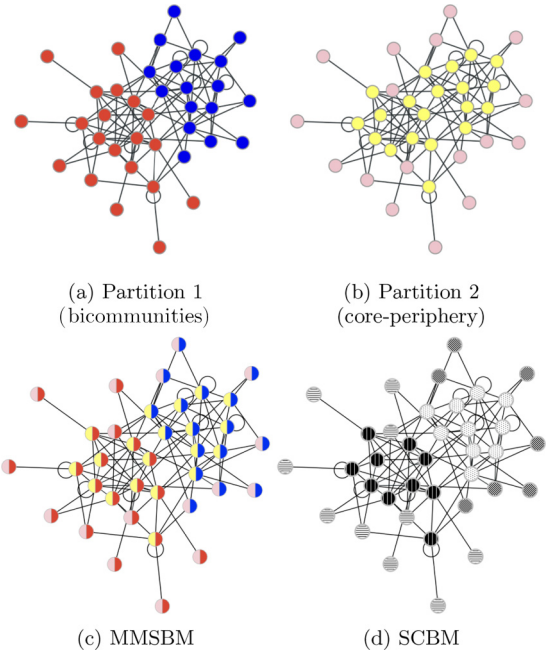


FIG. 1. Example graph with nodes colored according to their block memberships following different ways of formulating the partitions: the planted bicommunity partition (a), the planted core-periphery partition (b), a visualization of the MMSBM formulation of the two-partition-problem, where nodes belong to two blocks with equal probability (c), and the SCBM approach (d).

approach, and finally introduce the *stochastic cross-block model* (SCBM) as an alternative approach.

As stated above, we are generally guided by the KC example where both a bicommunity (a partition of the network into two assortative communities) and a core-periphery structure may be found and where it is likely that the two structures jointly explain the network formation. Figures 1(a) and 1(b) show the same example graph with nodes colored according to a bicommunity and core-periphery partition, respectively.

A. Formal outline

We generate a network with N nodes and E edges, into which we plant a set of P partitions with K_p blocks in partition p. Edges between nodes are described by the adjacency matrix $\mathbf{Y}$ of size $N \times N$. We focus on the undirected, unweighted case, whereby $Y_{ij} = Y_{ji} = 1$ if i and j are connected by an edge and 0 otherwise. The block structure in each partition p is defined as an SBM with two parameters: (1) the set of block membership vectors $\mathbf{b}_i^p$ of length K_p assigned to each node i and (2) the block matrix $\boldsymbol{\theta}_p$ of size $K_p \times K_p$, where diagonal (resp. off-diagonal) elements indicate the probability of an edge within (resp. between) blocks. Note that each $\mathbf{b}_i^p$ is a one-hot vector, which is a binary vector with only one element set to 1 (indicating the block membership of node i in partition p) and all others set to 0. We denote by $\mathbf{B}_p$ the $K_p \times K_p$ matrix of expected edge counts, which, for graphs with self-loops, has elements $\theta_{p_{rs}} n_{p_r} n_{p_s}$, where n_{p_r} is the number of nodes in block r in partition p. Note that block matrices are symmetric since the graphs we are generating are undirected and that for $r = s$ the elements of $\mathbf{B}_p$ denote twice the number of expected edge counts, for convenience of calculation and notation.

For illustrative purposes, we focus on the simple case of $P = 2$ partitions and $K_1 = K_2 = 2$ blocks in each partition, and we define the block matrices for the two implicitly planted partitions by $\boldsymbol{\theta}_1 = \{\theta_{1_{rs}}\}$ and $\boldsymbol{\theta}_2 = \{\theta_{2_{rs}}\}$. In Sec. III C we outline how our framework can be extended to more complex partition combinations.

B. MMSBM formulation

Given the parameters needed to plant two different partitions, we need to define the generative process which yields a network that satisfies the connectivity patterns of both. One possible approach is to formulate the two-partition scenario as a special case of an MMSBM with appropriate normalization. In the original MMSBM formulation [49], each node belongs to all latent groups with certain probability expressed through a mixed membership vector specific to each node. The existence of an edge between two nodes depends on the block memberships of the two nodes, which is repeatedly drawn from the mixed membership vector for each node pairing; nodes thus inherit connectivity patterns from multiple blocks, and the expected density at the "overlap" of multiple blocks is a weighted average of their individual densities [50]. In order to frame our two-ground-truth partition problem as an MMSBM, we consider the blocks in the two planted partitions as four latent blocks, but we constrain the generative process in a way that forces certain overlaps to be empty. In particular, we need to (a) constrain the mixed membership vectors so that the probability of nodes being members of certain combinations of blocks is zero, and (b) specify the within- and between-block edge probabilities in a way such that the connectivity of the generated network is consistent with that of the two planted partitions given by $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. The schematic in Fig. 1(c) visualizes these constraints by showing nodes colored according to their block memberships in two planted partitions.

It turns out that the required normalization of the additive block probabilities induced by the MMSBM is not straightforward. In particular, one needs to determine a normalization constant for each combination of block pairs, for which one needs to obtain the solution to an underdetermined system of equations. As we detail in Appendix A the minimum norm solution to this system (that can be obtained with a least squares solver) is negative for certain combinations of planted structures. This means different methods for approximate solutions are needed for different planted structures, which is likely to have unexpected side effects to an exploration of planted structures.

In order to be able to have enough flexibility for an exploration of a sufficiently large range of structure combinations, we thus propose an alternative to the MMSBM-based approach: it turns out that we can circumvent the extra step involved in finding the normalization constants by formulating the two-partition problem as a single membership SBM and by replacing the additive probabilities imposed by the MMSBM by multiplicative ones. Instead of generating the network through the mixed membership of nodes in two blocks, our proposed model is a simplification of the

problem which considers the overlaps between the blocks of the individual planted partitions as the blocks of an SBM.

C. Stochastic cross-block model

In the stochastic cross-block model (SCBM), a node is assigned a combination of the blocks it inhibits in multiple partitions, called a *cross-block*. We denote the set of cross-blocks by $\mathbf{\Gamma} = \{(r, r')\}$, where r and r' are the blocks in partitions 1 and 2, respectively. We can then rephrase our problem as follows: To plant two partitions in one single network, we generate a network in which we explicitly plant one single *cross-partition* with $K_{\mathrm{SCBM}} = K_1 K_2$ cross-blocks in a way that is consistent with the expected densities from the block matrices $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Figure 1(d) illustrates the four cross-blocks in the cross-partition resulting from the partitions visualized in Figs. 1(a) and 1(b). Note that the cross-partition—the explicit division of the network into K_{SCBM} cross-blocks—is different from the *partition coexistence*, which refers to the property of the network to be consistent with the connectivity of the two implicitly planted structures with K_1 and K_2 blocks, respectively.

Our generative network then simply becomes a standard SBM where the probability of an edge between two nodes i and j is determined entirely by the probability of a node between the cross-block u of i and the cross-block v of node j. To generate the final network, we straightforwardly create the one-hot cross-block membership vectors $\mathbf{b}_i^{\mathrm{SCBM}}$ for each node i from vectors $\mathbf{b}_i^p$. For the placement of edges, we need to determine the connectivity between and within the cross-blocks by defining $\boldsymbol{\theta}_{\mathrm{SCBM}}$ and $\mathbf{B}_{\mathrm{SCBM}}$. As above, element θ_{uv} denotes the probability of an edge between cross-blocks u and v and B_{uv} denotes the average expected number of such edges, where we have dropped the subscript. The block matrix is created in a way in which the edge probabilities are consistent with the elements of the block matrices $\boldsymbol{\theta}_p$ for each planted partition p. The deciding difference between this cross-partition approach and an MMSBM formulation of our problem is that we replace the additive probabilities that result from the MMSBM by normalized multiplicative probabilities. In other words, for each cross-block, the probability of an edge is calculated by multiplying the edge probabilities in the original blocks that cause the overlap and by normalizing appropriately.

In Fig. 2 we visualize two example block matrices $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and the resulting cross-block matrix $\boldsymbol{\theta}_{\mathrm{SCBM}}$; note that we include two differently ordered visualizations of the same cross-block matrix in Figs. 2(c) and 2(d) to emphasize that the cross-partition is consistent with both $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. The particular choice of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ visualized here is also responsible for the partitions visualized in Figs. 1(a) and 1(b), and the cross-block matrices thus correspond with the graph visualized in Fig. 1(d).

Equal block sizes. In the first instance, we focus on the case of equal block sizes, both in the planted partitions as well as in the cross-partition, so that $n_r = n$ for all blocks r and $2n = N$. We allow for self-loops, so we have $B_{p_{rs}} = n_r n_s \theta_{p_{rs}} = n^2 \theta_{p_{rs}}$ for $p \in \{1, 2\}$. We denote the vector of cross-block sizes by $\boldsymbol{\nu} = \{\nu_u\}$ and constrain cross-blocks to be equally sized: $\nu_u = \nu$ for $u \in \mathbf{\Gamma}$ and thus $2\nu = n$. We construct the edge

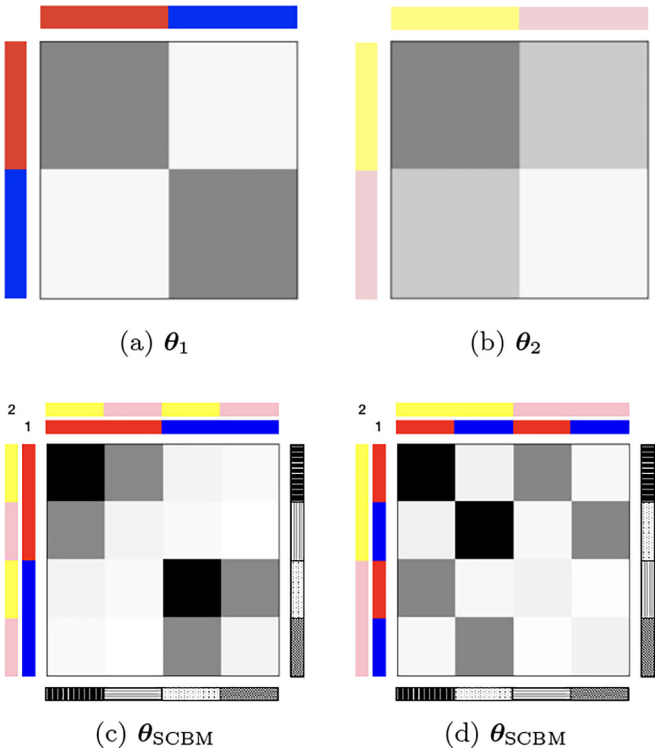


FIG. 2. Block matrix visualizations for two planted partitions (a, b) and resulting cross-partition (c, d). Note that (c) and (d) are equivalent but ordered differently, as visualized by the colored bars (blocks in partitions 1 and 2) and patterned bars (cross-blocks). Matrix elements are colored according to the expected edge densities in the block matrices, with black elements representing the maximum edge probability and white elements representing an edge probability of 0; note that the colors are consistent so that they are comparable across the three graphs.

probabilities within and between cross-blocks through the product of the probabilities of the blocks in the two partitions, with an appropriate normalization. In particular, we define the block matrix of the SCBM as $\boldsymbol{\theta}_{\mathrm{SCBM}} = \{\theta_{uv}\} = x\{\theta_{1_{rs}}\theta_{2_{r's'}}\}$ with cross-blocks $u = (r, r') \in \mathbf{\Gamma}$ and $v = (s, s') \in \mathbf{\Gamma}$, where the constant x needs to be chosen in a way such that the expected number of edges within and between certain cross-block pairs in $\mathbf{B}_{\mathrm{SCBM}}$ adds up to the respective elements of $\mathbf{B}_1$ and $\mathbf{B}_2$, so that the connectivity in the generated network is consistent with that of partitions 1 and 2. In the case of equal block sizes, this is satisfied for $x = 1/\rho$ where $\rho = 2E/N^2$ is the overall expected edge density of a network with self-loops (see Appendix B for the derivation). The elements of the expected edge count matrix are denoted by the matrix elements $B_{uv} = \nu^2 \theta_{uv}$, where we have dropped the SCBM subscripts in B and θ.

Varying block sizes. In any other more general case in terms of varying block sizes, a constant x satisfying both planted partitions exists only if we introduce other constraints, such as equal edge probabilities across all blocks of the two planted partitions, i.e., planting a random graph (see Appendix C). To determine $\boldsymbol{\theta}_{\mathrm{SCBM}}$ for nontrivial partitions with varying block sizes, we need to find a set of normalization constants $\{x_{uv}\}$, one for each cross-block pairs, so that

$\boldsymbol{\theta}_{\mathrm{SCBM}} = \{\theta_{uv}\} = \{x_{uv}\theta_{1_{rs}}\theta_{2_{r's'}}\}$. To find $\{x_{uv}\}$ in a way in which $\boldsymbol{\theta}_{\mathrm{SCBM}}$ generates a network consistent with $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ we must, again, solve an underdetermined system of equations for which we can compute a minimum norm solution, which is unique and always exists if any solution to the system exists (see Appendix D). In this case, the minimum norm solution returns a set of entirely positive $\{x_{uv}\}$ across our entire parameter space in Sec. IV; while we explore the equal-block case for our simulations, we have shown here that a more general setup is also possible using the SCBM approach.

Multiple blocks and partitions. In our simulations in Sec. IV, we focus on the simple two-partition case described above to demonstrate detectability issues of two "ground truth" partitions. However, it may in some cases be interesting to explore more complex structures with more coexisting partitions and/or more blocks. While a thorough exploration of such cases is beyond the scope of this work, we briefly outline an extension of the described simple case. In fact, keeping the number of partitions at $P = 2$ but planting more than two blocks in one or both partitions can be done straightforwardly with the same normalization constant in the case of equal block sizes and with a larger system of equations having to be solved in the case of varying block sizes. Since planting more than two partitions can also be reframed as recursively planting sets of two partitions until a final cross-partition is reached, this is also possible. In the case of equal block sizes, the SCBM block matrix can then be generalized as $\boldsymbol{\theta}_{\mathrm{SCBM}} = 1/\rho^{p-1}\{\prod_p \theta_{p_{rs}}\}$. In the case of varying block sizes, one needs to consider the possible limitation of scalability that arises for large numbers of partitions and/or blocks, since the number of cross-blocks is $K = \prod_p K_p$ and we need to find $K(K+1)/2$ normalization constants. The least square method for finding the minimum norm solution involves computing the pseudo-inverse of the coefficient matrix, which can be computationally expensive for large $n \times m$ matrices [with a computational complexity of approximately $O(n^2 m)$]. Limitations of computational time and memory thus need to be considered for very complex combinations of planted partitions.

D. Generative SBM

We generate the final network from the connectivity matrices according to the "traditional" SBM [33], which uses matrix $\boldsymbol{\theta}_{\mathrm{SCBM}}$ alongside cross-block membership vectors $\mathbf{b}$ to determine whether or not an edge exists between two nodes. More specifically, we will place an edge between each pair of nodes (i, j) independently at random, with probability θ_{uv}, where $u, v \in \mathbf{\Gamma}$ are the cross-blocks of i and j, respectively. We thus sample the value of the interaction between i and j with $Y_{ij} \sim \mathrm{Bernoulli}(\mathbf{b_i}^T \boldsymbol{\theta}_{\mathrm{SCBM}} \mathbf{b_j})$. In this version of the SBM, the expected edge counts in $\mathbf{B}_{\mathrm{SCBM}}$ are satisfied on average.

We also consider the microcanonical SBM [34], in which the (rounded) elements of the given matrix $\mathbf{B}_{\mathrm{SCBM}}$ are satisfied exactly (rather than on average) and which is based on the configuration model [54]. Specifically, we consider the degree-corrected extension of this microcanonical SBM, in which the probability of an edge being placed between two nodes does not depend solely on the elements of a connectivity matrix but also on a given degree sequence or distribution. This SBM variant has been demonstrated to have characteristics that more closely resemble empirical networks, by producing synthetic networks with the type of within-block degree variability that is more likely to occur in real networks [19]. Given a degree sequence $\{k_i\}$ in which k_i denotes the degree of node i, this works by assigning k_i half-edges to node i and then choosing two half-edges in the network at random (allowing for self-edges) and connecting them under the condition that the expected given within- and between-block edge counts are satisfied. However, the elements of $\mathbf{B}_{\mathrm{SCBM}}$ can be real numbers and must therefore be rounded in the network generation process. This introduces small differences between the (implicitly planted) expected edge count matrices $\mathbf{B}_p$ and the generated networks in terms of the total number of edges as well as the within- and between- cross-block edge counts. It also means that a given degree sequence can be satisfied exactly only if $\mathbf{B}_{\mathrm{SCBM}} \in \mathbb{Z}^{N\times N}$; in our simulations we sample node degrees from a power-law distribution rather than satisfying the exact degree sequence. Note that in the graphs we generate in this way in our simulations below, any pair of nodes is connected by a maximum of one unweighted edge; removing this constraint and producing multigraphs instead is straightforward.

The version of our model which generates networks according to the traditional SBM will from now on be called the *canonical model* to distinguish it from the latter version, which we will call the *microcanonical model*. This is to avoid confusion in the notation between the SBM variants we use to *generate* our networks from those we use to infer partitions.

IV. SIMULATIONS

We now explore the extent to which two built-in ground truths are recovered by SBMs, by generating a set of networks in which we implicitly plant two partitions. Clearly, there are many interesting two-partition structures one may explore; as indicated before, we are interested in the type of structure present in our motivating example, the KC network. For this network, samples of the posterior distribution of inferred partitions yield a number of plausible explanations of the mesoscale structure [11]; one local consensus partition is the famous two-faction division of the network into two assortative communities, another is a leader-follower partition that resembles a core-periphery structure. We are interested in the recovery limits of these two types of structures in networks and we therefore plant similar structures into an ensemble of synthetic networks according to our generative framework. In our simulations, we build both a bicommunity as well as core-periphery structure into a set of graphs, and we fit two different SBM variants to our networks to infer the posterior distribution of partitions for each of them. We finally calculate the similarity between the recovered partitions and the planted partitions and present the results for the partitions planted by each model variant and recovered by each SBM variant.

A. Parameters

We focus on the case of equal block sizes here, for which the multiplicative probabilities in $\boldsymbol{\theta}_{\mathbf{SCBM}}$ can be normalized simply by the constant $x = 1/\rho$ and we do not have to rely

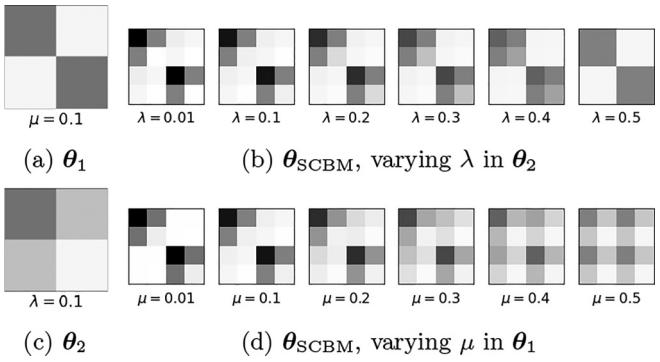


FIG. 3. Schematic block matrix $\boldsymbol{\theta}_{\text{SCBM}}$ and, $\boldsymbol{\theta}_1$ when fixing parameter μ and increasing λ in (a) and (b), respectively, or $\boldsymbol{\theta}_2$ when, similarly, fixing λ and varying μ in (c) and (d).

on the minimum norm solution (see Sec. III C). We plant networks with $N = 400$, and we run three sets of simulations with varying average degree $c = 5, 10, 20$. The total expected edge density is therefore different between the three sets of simulations but held constant within each set. This is to ensure that any differences we are seeing in the detectability of the two planted partitions are due to a different distribution of the edges within the network (which we will induce by varying the block connectivities), rather than differences in total edge density. Nodes are chosen uniformly at random and assigned to each block to create block membership vectors $\mathbf{b}_1$ and $\mathbf{b}_2$. To plant the two partitions, we define the symmetric block matrices $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ in Eq. (1a) and Eq. (1b) where $\beta = 2E/n^2$ with E being the total number of edges in the network. Partition 1 ($\boldsymbol{\theta}_1$) configures the bicommunity structure whereby a parameter μ controls the expected intra- vs interblock connectivity strength for two planted equal-sized communities. Partition 2 ($\boldsymbol{\theta}_2$) configures the core-periphery (CP) partition, in which the expected edge density within the core and the expected edge density among peripheral nodes is controlled by parameter λ. Note that the edge probability between blocks in partition 2 is fixed in this way so that for $\lambda \in [0, 0.5)$, we always have $\theta_{2_{11}} > \theta_{2_{12}} > \theta_{2_{22}}$, in line with common definitions of core-periphery structure, and to have equal probabilities within and between blocks for both $\mu = 0.5$ and $\lambda = 0.5$:

$$\theta_1 = \beta \begin{pmatrix} 1-\mu & \mu \\ \mu & 1-\mu \end{pmatrix}, \tag{1a}$$

$$\theta_2 = \beta \begin{pmatrix} 1-\lambda & \frac{1}{2} \\ \frac{1}{2} & \lambda \end{pmatrix}. \tag{1b}$$

We use our model to generate multiple sets of networks, sweeping parameters μ and λ from 0.01 to 0.5 at increments of 0.01 in each case. Note that we exclude $\mu = 0$ as it would yield a disconnected graph of two components and exclude $\lambda = 0$ for symmetry in the two dimensions.

Low values of μ generate assortative community structure (in the sense that most edges are placed within blocks and few between blocks), while μ close to 0.5 produce a network close to a random graph. Values of λ close to zero generate "clear" core-periphery structure, with most edges being placed within the core and few among peripheral nodes, while $\lambda = 0.5$ produces a random graph. In Fig. 3 we show the behavior of $\boldsymbol{\theta}_{\text{SCBM}}$, for fixed μ and varying λ and vice versa. The block matrices in Figs. 3(b) and 3(d) illustrate that the cross-partitions for low values of both parameters resemble a nested structure of two communities with internal core-periphery structures, similar to existing work on core-periphery pairs in networks [3,26–28]. For increasing λ while fixing μ, the block densities of the bicommunity partition remain constant and the core-periphery structure becomes weaker until, at $\lambda = 0.5$, we are left with a bicommunity partition as $\boldsymbol{\theta}_2$ now defines a random graph. Similarly, for fixing λ and increasing μ to $\mu = 0.5$, we finally reach a core-periphery structure [by reordering the rows and columns of $\boldsymbol{\theta}_{\text{SCBM}}$ in Fig. 3(d)].

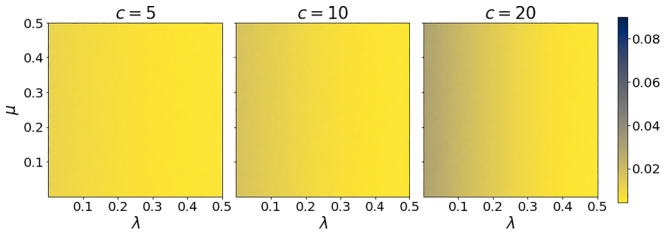
Note that what we are explicitly generating with the SCBM is a simple SBM with the cross-partition induced by the cross-block matrix $\boldsymbol{\theta}_{\text{SCBM}}$—the connectivity patterns of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are explicitly satisfied. When we infer the most likely partitions from the generated matrices, we expect that both the explicitly planted cross-partition as well as the two implicitly planted partitions 1 and 2 are recovered to some extent, potentially varying across the (λ, μ) space.

B. Generated graphs

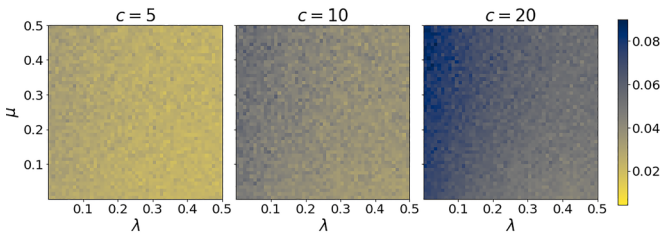
To appraise differences in degree distributions, we use the canonical model in one set of simulations and the microcanonical model in another. We thus generate two sets of networks for each expected degree c. In the canonical model, node degrees follow a Poisson degree distribution and edge counts within and between blocks are satisfied on average. In the microcanonical case, as described in Sec. III D, edge counts do not fluctuate across different runs of the model, and we impose further constraints on the node degrees, which we sample from a power-law distribution with exponent $\gamma = 3$. We use a soft constraint, in the sense that the final network does not have to match the given degree sequence exactly, but only on average. See Appendix F for a summary of the small deviations of the edge counts in the generated graphs from the planted edge count matrices due to rounding errors. In both cases, we generate eight networks for each (λ, μ)-pair, to account for possible fluctuations in the generative process.

Before attempting to recover planted partitions in the two sets of graphs, we explore structural characteristics introduced into the networks for different (λ, μ)-pairs and through the two different generative processes of the canonical and microcanonical model. Figure 4 demonstrates that, unsurprisingly, degree variance is highest across the entire (λ, μ) space for graphs generated by the microcanonical model, since we sample a heterogeneous power-law degree distribution. It is considerably lower for graphs generated by the canonical version. Notwithstanding, using the canonical model, higher degree heterogeneity is introduced into networks for lower values of λ, for which we are imposing a strong core-periphery structure (see Fig. 11 in Appendix F for a rescaled version of the top row of Fig. 4).

While graphs produced by the microcanonical model exhibit the highest degree heterogeneity overall, Fig. 5 illustrates that their core and periphery blocks are more similar in terms of node degree distributions than in the canonical case, measured by the Jensen-Shannon distance [55] between the degree distributions of the core nodes and those of the peripheral



(a) Canonical model

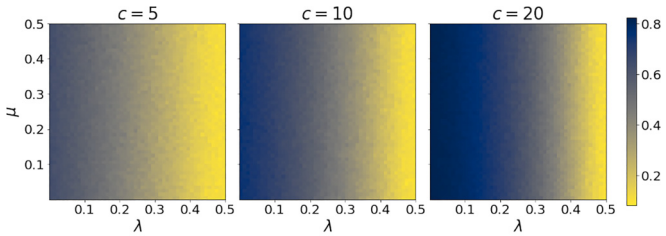


(b) Microcanonical model

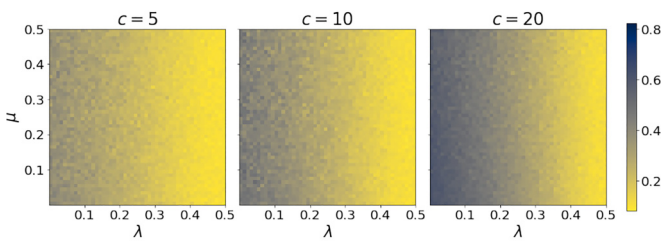
FIG. 4. Mean normalized degree variance for graphs with varying expected degrees c.

nodes. In the canonical case, node degrees are Poisson distributed for the network as a whole, as well as for nodes within each block. However, to accommodate for the core-periphery structure, the mean of the distribution is lower in the periphery than in the core, which in the Poisson case leads to a relatively small overlap between the two distributions [4]. In the microcanonical case, we introduce degree heterogeneity through the degree distribution so that planting CP structures does not produce the same differences in the block degree distributions.

This means that were we to simply assign nodes with above average degree to the core and those with below average degree to the periphery in graphs generated by the canonical model we would retrieve more correctly assigned nodes than



(a) Canonical model



(b) Microcanonical model

FIG. 5. Mean Jensen-Shannon distance between core and periphery node degree distributions for graphs with varying expected degrees c.

in networks produced by the microcanonical model [4], where by "correctly" we refer to the CP block planted by $\mathbf{B}_2$ as defined in (1b). The heterogeneity in degree distributions in the microcanonical case may generate other types of core-periphery patterns than the planted ones. (See Appendix F for a comparison of the number of correctly classified nodes in this way for the two models and of the degree distributions of the core and the periphery nodes for two example graphs.)

It is thus worth elaborating on the potential consequences of specifying the degree distribution as an extra parameter in the generative process in particular, additionally to the planted connectivity matrix and block assignments of nodes. While we expect the graphs produced by the canonical model to exhibit structure closely related to what we explicitly plant (i.e., the block connectivity matrices), we may implicitly be introducing additional structure through the constraints imposed on node degrees in the microcanonical case. Heterogeneity in degree distributions, for example, may lead to groups of nodes that display similarities in their connectivity with the rest of the network in terms of their number of connections; it is possible that the dividing lines between these groups do not correspond with those imposed by our planted block structure, which may lead to structures other than those explicitly planted being picked up. These differences in degree heterogeneity introduced through the generative process are thus likely to have an impact on the extent to which SBM variants recover the (coexistence of the) planted partitions in different regions of the (λ, μ) plane, as is confirmed in Sec. V.

C. Similarity measure

To quantify the similarity between planted and recovered partitions, we calculate the maximum partition overlap $\omega(p, q)$, namely, the proportion of nodes in one partition p assigned to the same, assumed correct, block of the other partition q [11]. This is calculated by finding the bijection $p' = \zeta(q)$ of the group labels of q, so that the number of nodes that have the same block label in p' and p is maximized, so that we have

$$\omega(p, q) = \frac{1}{N} \max_{\zeta} \sum_{i} \delta_{p_i, \zeta(q_i)}. \tag{2}$$

Specifically, this is done by solving the maximum weighted bipartite matching problem for two partitions using a function from the graph-tool python library [56], which is based on the Kuhn-Munkres [57,58] algorithm. Note that here we use the normalized partition overlap, which is between 0 and 1. Therefore, $\omega = 1$ when all nodes of two partitions coincide. Note that if both of the compared partitions have two blocks, $\omega = 0.5$ is the lower bound and implies that half of the nodes are classified correctly and therefore the two partitions are not correlated. When one partition has more than two blocks, we can have $\omega < 0.5$.

The partition overlap measure is a suitable choice of similarity measure since it is easier to interpret than information theoretic measures, such as those based on mutual information, and since it does not depend on the number and size of blocks in the two partitions being compared, which is an issue for some pair-counting methods such as the rand index [59]. We demonstrate the robustness of our results by calcu-

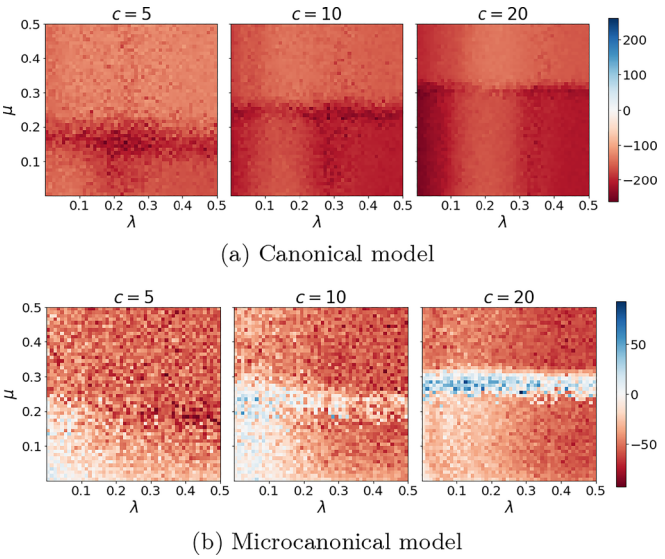
lating the partition similarity for one set of simulations using reduced mutual information [60] and variation of information, which has also been shown to behave well for unbalanced partitions [61], and we show the results in Appendix E. Both similarity measures yield comparable results to those calculated using the partition overlap measure. In particular, detectability thresholds appear to be located identically (or at least extremely similarly) in the (λ, μ) space.

V. RESULTS

To infer partitions of the generated graphs, we fit two SBM variants (traditional and degree-corrected) using the graph-tool Python library [56]. We retrieve a distribution of 50 partitions for each of the eight graph and therefore a total of 400 partitions for each combination of λ and μ. To ensure that the chosen number of samples is sufficient to explore the posterior we ran the same simulations[1] with 4000 partitions for each (λ, μ)-pair (sampling 1000 partitions for each of four graphs) and found no qualitative difference. In its function as an *inference* method, we from now refer to the traditional SBM as NDC (non-degree-corrected SBM) and to the degree-corrected variant as DC, to avoid confusion with the models (canonical and microcanonical) used to *generate* our networks. We finally calculate the partition overlap ω between inferred partitions and planted partitions for the two planted structures as well as between the inferred partitions and the planted cross-partition.

A. Model fit

We start by evaluating which of the two SBM variants used for the detection of mesoscale structures provides a better fit to our generated networks. We calculate the (log) model evidence, summed over all partitions for each run, calculated by subtracting the entropy of the posterior distribution from the negative average description length (over all partitions) [34]. Figure 6(a) demonstrates that for the canonical version NDC is the preferred model across the entire (λ, μ) space; this is unsurprising as edge placement in the generative process is independent of node degree. When we use the microcanonical version (in which we *do* take into account the node degrees in the generative process), we may have expected DC to be a better description of the generated networks across the entire (λ, μ) plane. However, Fig. 6(b) demonstrates that this is not the case: we observe a small region of λ and μ values for which DC has the larger model evidence; for increasing expected degree, this region becomes more pronounced and exists across the entire λ range, while restricted to more and more narrow values of μ. Everywhere else, NDC still provides a better model fit. This suggests that the higher complexity of DC is justified only for networks with a high level of heterogeneity in the degree distribution *and* bicommunity structure of a certain strength which depends on c. It seems that—in terms of the number of model parameters—DC provides



(a) Canonical model

(b) Microcanonical model

FIG. 6. Difference between the log evidence of the DC and NDC model class for graphs with varying expected degrees c. Negative values (red) indicate a better fit of the NDC model; positive values (blue) indicate a better fit of the DC model.

an overly complex description everywhere else in the (λ, μ) plane, although degree heterogeneity is still high.[2]

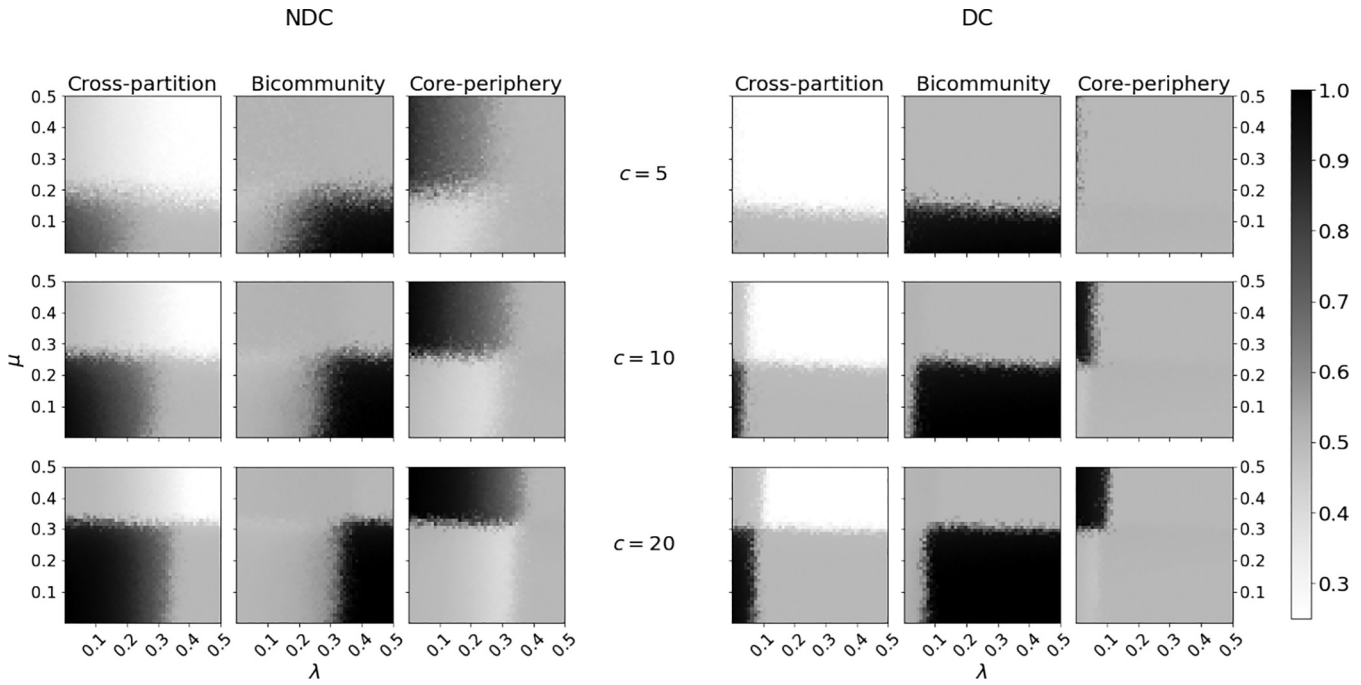
B. Recovery of planted structures

In Fig. 7 we show the mean partition overlap $\langle\omega\rangle$ for all (explicitly and implicitly) planted partitions in the posterior distribution of inferred partitions for each (λ, μ) pair, for networks generated by two models. The leftmost columns in all four quadrants show $\langle\omega\rangle$ between the inferred partitions and the explicitly planted cross-partition ($\boldsymbol{\theta}_{\text{SCBM}}$); the middle and right columns show the same for the implicitly planted bicommunity ($\boldsymbol{\theta}_1$) and core-periphery ($\boldsymbol{\theta}_2$) partitions, respectively. Note that here we focus on the detection of the individual partitions, and we refer to Sec. V D for an outline of the detection of partition coexistence: the extent to which *both* implicitly planted structures appear in the posterior distribution of inferred partitions of a given generated graphs.

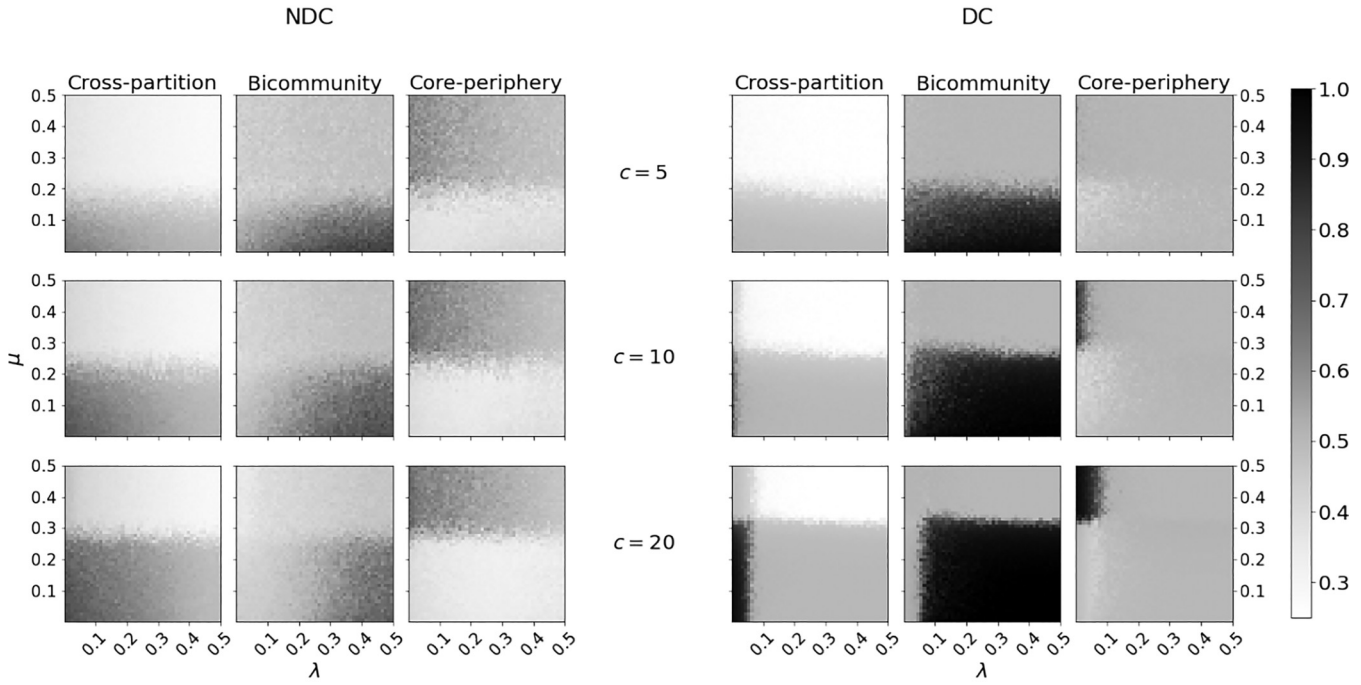
As expected, there appear to be some clear thresholds separating areas in the (λ, μ)-space in which the cross-partition is recovered from areas in which either of the two implicitly planted structures are detected. The locations of these thresholds vary by expected degree, by generating model and by SBM variant used for the partition inference. The question around the detectability of the two implicitly planted structures thus appears to be related to the detectability of the

[1]For graphs with $N = 400$, $c = 10$ generated by the canonical model and partitions inferred using the non-degree-corrected SBM.

[2]If we allow for multiple edges between node pairs (using the graph-tool Python library [56]), we find an additional area in the bottom left of the (λ, μ) plane, roughly for values $\lambda < 0.2$ and $\mu < 0.33$,in which DC is preferred. A possible explanation could be the larger degree variance introduced in this case, for which the higher complexity of DC is justified. As the partition recovery patterns for these multigraphs are equivalent to the ones we discuss in Sec. V, we restrict our analysis to simple graphs.



(a) Canonical model.



(b) Microcanonical model.

FIG. 7. Mean partition overlap $\langle\omega\rangle$ between the planted partitions and the partitions in the posterior distributions for graphs with varying expected degrees c.

cross-partition: as lower values of both parameters mean a stronger signal for the cross-partition, it is detected up until some threshold. Only once the signal for the cross-partition becomes weak enough are the implicitly planted partitions favored.

We first focus on the partitions inferred from the graphs generated by the canonical model, where node degrees follow Poisson distributions. In Fig. 7(a) we show $\langle\omega\rangle$ between planted partitions and those recovered by NDC (on the left-hand side) and DC (on the right-hand side). Both variants detect the bicommunity structure frequently up to a certain threshold value of μ, which increases for higher c. It turns out that the locations of the thresholds are roughly in line with what is described in existing literature on detectability

thresholds for the planted partition model [13,14]. According to this work, community structure planted by edge count matrix $\mathbf{B}_1$ as defined in (1a) is detectable for $c > \frac{1}{(1-2\mu)^2}$; in our case this should place our threshold μ_T for detectability at $\mu_T \approx 0.276$ ($c = 5$), $\mu_T \approx 0.342$ ($c = 10$), and $\mu_T \approx 0.388$ ($c = 20$). In our simulations, both variants detect bicommunities up to a similar value of μ, which is slightly below μ_T. We observe a second type of threshold for bicommunity detection, this time along the λ dimension. For low values of λ, both variants fail to detect the bicommunity partition even though $\mu < \mu_T$, and they appear to uncover the cross-partition instead. While the threshold along the μ dimension is similar for NDC and DC, we find that the λ threshold is much higher for NDC than for DC. This means that NDC recovers the cross-partition up to higher values of λ than DC, after which bicommunities are detected. The λ threshold also increases with growing c.

The detection of the planted core-periphery partition also depends on the expected degree of the networks. In fact, the thresholds of CP detection correspond with those described above for bicommunity detection: along the μ dimension, CP structure is detected once cross-partitions and bicommunities are no longer recovered; along the λ axis, CP structure is detected until its structure is too weak, at which point bicommunities are detected. This is somewhat contradictory to the work by Zhang *et al.* [4], who find no evidence for a detectability threshold in the case of CP structures. A likely explanation for the narrow recovery range of the CP structure by DC compared to NDC is that degree correction aims to account for degree heterogeneity in a network in favor of detecting community structure, while NDC has a higher tendency to split networks into blocks of lower and blocks of higher degree [19], which here corresponds to the implicitly planted CP structure.

One of the main differences between NDC and DC in the canonical case is therefore the thresholds at which structures do and do not get detected along the λ axis. For all values of c and for both SBM variants, the cross-partition is recovered when the bicommunity and CP structures are strong. Along each direction, both variants then start picking up the respective two-block structure once the signal becomes weaker. Since both the bicommunity and CP structures are recovered only when the signal for the respective other structure is weak, the coexistence of both structures in the inferred partition distribution is rare; we revisit this in Sec. V D. As both thresholds (along λ and μ) are higher for larger c, the cross-partition detection region increases for denser graphs; this phenomenon is more pronounced for NDC, which provides a better model fit than DC across the entire (λ, μ) plane.

C. Influence of degree distribution

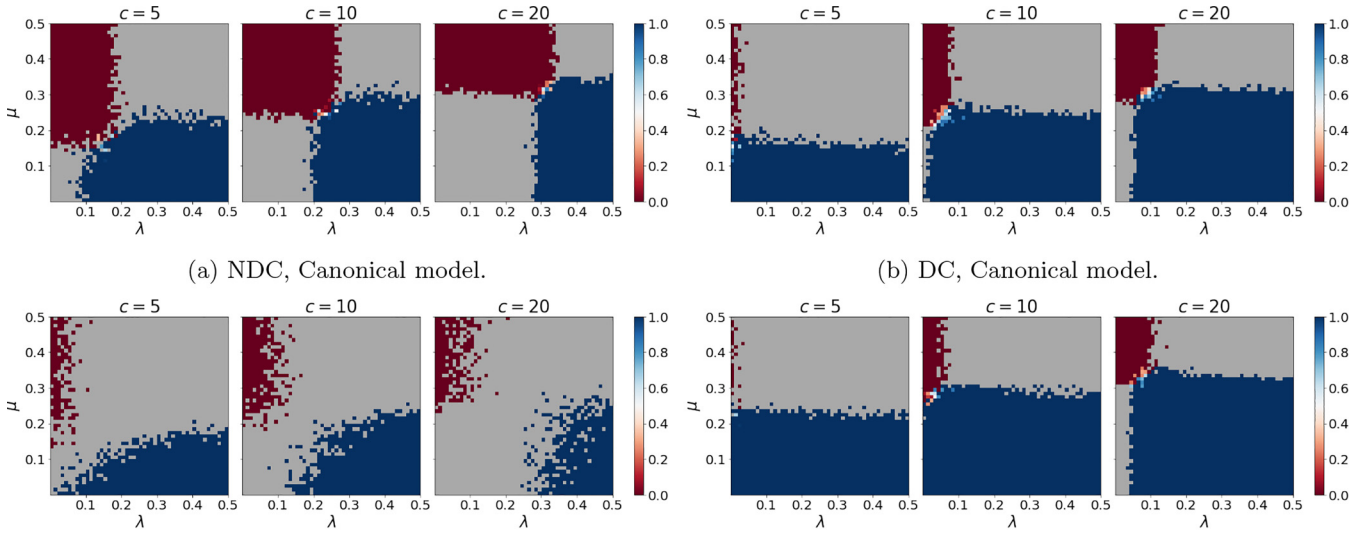
To explore the influence of a heterogeneous degree distribution on partition recovery, we fit the two SBM variants to a set of networks generated by the microcanonical model. Figure 7(b) illustrates the mean recovery of partitions in this case by NDC (left) and DC (right). The overall detection patterns resemble those discussed for the canonical case. For DC, all partitions are recovered in similar regions with similar thresholds, albeit slightly more "fuzzy" boundaries on said thresholds. However, we observe a substantial difference in the performance of the NDC variant, for which $\langle\omega\rangle$ is considerably lower for all planted partitions across the entire (λ, μ) plane. In the first instance, this is somewhat surprising, since we have seen in Sec. V A that even for graphs generated by the microcanonical model NDC provides a better description. A plausible explanation of this phenomenon is the additional structural features that we introduce through the extra constraint on node degrees in our microcanonical model and the thereby imposed degree heterogeneity (see Sec. IV B). It turns out that for relatively strong bicommunity structures, the NDC variant recovers layered CP structures nested within each of the two community blocks, both for very strong planted CP structures but also when no explicit CP structure is planted at all (see example graphs in Appendix H). When CP structures are planted explicitly with strong signal, the layered CP partition recovered by NDC bears some resemblance to the cross-partition; when no explicit CP structure is planted, the layered CP structures within two assortative blocks resemble more the bicommunity partition (according to ω and upon visual inspection of the example networks). As can be seen in Fig. 7(b), the DC variant detects partitions much closer to the planted cross-partition (for lower λ) and the bicommunity (for higher λ) value than NDC. However, the NDC variant has the better model fit; this implies that by forcing heterogeneous degrees, we may to some extent be "overfeeding" more structure into the network than solely that defined through the block connectivity matrices.

D. Structure coexistence

Other than the individual recovery of the two planted partitions, we are naturally also interested in the appearance of both structures in different regions of the partition landscape detected in a given network, that we denote as coexistence. Specifically, we want to know whether the posterior distribution of partitions inferred by SBM features both planted partitions (rather than only one or the other) for any particular set of (λ, μ) pairs. We measure coexistence recovery by setting the threshold for the partition overlap to $\omega_T = 0.75$, for which we consider an inferred partition to be close enough to the planted partition to be considered a "successful recovery."

To illustrate the coexistence detection, we plot the fraction $\alpha = \frac{q_1}{q_1+q_2}$, where q_1 denotes the proportion of partitions in the posterior distribution that resemble the bicommunity partition, given ω_T, and q_2 denotes the equivalent for CP partitions. The results for each model or variant combination are shown in Fig. 8. For $\alpha = 1$ (dark blue) we detect only the bicommunity structure, for $\alpha = 0$ (dark red) only the CP structure is present in the posterior distribution, and values close to $\alpha = 0.5$ (white) indicate a more balanced posterior distribution, which features both partitions to some extent; such values are found where the detection areas for the two planted partitions appear to be touching or even overlapping. The gray region represents (λ, μ) pairs for which α is undefined since neither of the two structures is recovered successfully.

We observe in Fig. 8(c) that, as expected from the results in Sec. V C, fitting NDC to graphs generated by the microcanonical model does not yield partition distributions anywhere in our (λ, μ)-space that feature both bicommunity



(a) NDC, Canonical model.

(b) DC, Canonical model.

(c) NDC, Microcanonical model.

(d) DC, Microcanonical model.

FIG. 8. Fraction α of recovered bicommunity partitions out of all successfully recovered partitions for varying expected degrees c for $\omega_T = 0.75$; at $\alpha = 1$ (dark blue) only the bicommunity structure is detected; at $\alpha = 0$ (dark red), only the CP structure is.

and CP partitions. In all other cases, there are small regions for which coexistence is detected, for this relatively generous threshold value of $\omega_T = 0.75$. Note that for stricter ω_T thresholds, we observe reduced recovery regions for each partition in all cases and therefore an even smaller or completely absent region of overlap in which both partitions feature in a given posterior distribution (see Appendix G). Clearly, choosing this value of ω_T to mean "successful recovery" is somewhat arbitrary, and one might argue that a maximum of 75% of nodes being assigned to the (implicitly planted) "correct" blocks does not indicate strong partition similarity. In fact, we choose to display the results with this low thresholds here, since it emphasizes the finding that even for such a generous threshold coexistence recovery is very small.

E. Discussion

Overall, we found detectability thresholds for each individual planted structure, and we discovered that coexistence of the two structures is detected only in a very small number of cases. We also observed considerable differences in successful partition recovery between the NDC and DC variants, which are more pronounced when they are fitted to graphs generated by the microcanonical model than when fitted to those produced by the canonical version.

We briefly discuss the effect of degree heterogeneity introduced into networks through the generative process of the microcanonical model. We found that, by constraining the degree distributions in this way, we are inadvertently introducing CP divisions *beyond* the explicitly planted CP structure. The additional CP structures are picked up by NDC since it does not correct for node degrees, and it thus comparatively "underperforms" at detecting individual planted partitions and coexistence of multiple partitions. These findings are consistent with existing SBM literature, including the original work in which the degree-corrected variant was introduced [19]. They should serve as a reminder that a network may exhibit multiple conflated structural properties, which could in turn complicate the detection of certain types of partitions or even lead to the detection of spurious mesoscale structures. In general, and specifically if previous knowledge exists about structural properties that are likely to be present in a network (e.g., high degree variance) or about certain types of structures that are of interest, one should consider carefully the SBM variant that is appropriate. Methods which specifically aim to disentangle conflated structural properties [19,62] or to recover certain types of structures, such as assortative communities [63], could be considered.

Second, we focus on the canonical model and the extent to which the two SBM variants recover the bicommunity and CP partitions relative to each other, jointly (coexisting in a given posterior distribution) and relative to the cross-partition. Overall, we have found that thresholds for the detection of the individual planted partitions and for the detection of structure coexistence depend on the expected degree of a network as well as the SBM variant used to detect the structure. The NDC variant, which has higher model evidence, does better at picking up the cross-partition at the expense of recovery of the CP structure. It recovers coexistence of bicommunity and CP partitions to a slightly lesser extent than DC. Since we are *explicitly* planting the cross-partition, and only *implicitly* planting the bicommunity and CP structures by making sure the edge probabilities within and between the respective cross-blocks satisfy those within and between the blocks of the two two-block partitions, it is perhaps not surprising that the variant with the better model fit is the one that favors the cross-partition at the cost of coexistence detection.

Irrespective of whether we use NDC or DC, we find that even in this relatively simple case of planting only two qualitatively different ground-truth partitions, the region in our structural strength landscape for which the coexistence of the bicommunity and CP structures is detected is limited to an extremely small area. This is concerning since mesoscale structures in real networks are unlikely to be so simple, and

a larger number of coexisting network partitions and a multitude of different structures may be present. Clearly, more research is necessary to better understand whether the lack of coexistence recovery is due to a detectability limit after which it is impossible for any algorithm to detect coexistence (similar to the known community detection detectability threshold [13,14]), or whether some other SBM variant would be able to do a better job at detecting coexistence of multiple planted structures. More work should then also focus on expanding the relatively recent literature on partition diversity [11,37] by advancing existing methods or developing new tools. The aim should be to enable researchers to reliably explore multiple coexisting ground truth partitions that may have been responsible for the generation of a given network, which we suspect to be the case in real graphs [12]. In this sense, our framework and findings should be seen as a motivation to test new SBM variants developed for this purpose. The regions of coexistence discovery shown in Fig. 8 may be used as an orientation for the possible locations of detectability thresholds in the case of multiple ground-truth partitions.

More generally, the fact that even for existing methods coexistence *is* detected in certain regions of our structural strength landscape emphasizes again the importance of acknowledging the diversity and possible dissensus in partition distributions, and for more researchers in the field of applied network science to consider multiple plausible explanations of the mesoscale of network.

VI. CONCLUSION

We have proposed the stochastic cross-block model (SCBM), a framework for generative network models that exhibit predefined ambiguity in their mesoscale structure. This framework complements existing generative networks as a *two-ground-truth benchmark* that can be used to measure the extent to which mesoscale structure detection algorithms recover the ambiguity introduced by two simultaneously planted partitions. Our work also generally emphasizes the need to explore the question around ambiguity in network structure, in the sense that the cross-partition that we plant explicitly is what we plant unambiguously, whereas the ambiguity stems from the two implicitly planted partitions. While we focus on the two-partition case in our simulations, we also outline how our approach can be extended to the multipartition case, and we encourage future work on more complicated cases of multiple ground truth structures, which are arguably closer to what may occur in real networks.

We detail a possible way to frame the multiple ground truth problem as a special case of the MMSBM and explain why our method simplifies the MMSBM approach. We found that the coexistence of two qualitatively different partitions (bicommunity and core-periphery structure) is detected only in a very small region in our "structural strength" space, which varies in size and shape for different versions of our model and for different SBM variants used for mesoscale structure detection. Only when both structures are sufficiently strong and neither dominates the other can we recover the existence of both. In the majority of cases, each of the two planted partitions is recovered when the strength of the other structure is weak. We have thus uncovered a type of detectability threshold in the case where multiple types of mesoscale structures influence the network construction. Since the coexistence of more than one plausible explanation for mesoscale structures appears to be a common phenomenon in real networks [12], we believe that exposing the presence of such an, as of yet understudied, detectability threshold is an important contribution to the SBM literature, especially as most community detection approaches still aim at uncovering a single partition and at validating it against a single ground truth. More work is required to explore the nature of these detectability thresholds analytically, and to appraise detectability limits exhibited by other types of coexisting structures, for example, including more than one community partition or bipartite structures; the combination of certain types of structures may be more or less prone to detectability issues than others, especially given the ability (or lack) of certain SBM variants to detect certain types of structures. Other future work may include fitting structure-specific SBM variants to graphs generated by the SCBM benchmark. For example, SBMs designed to detect assortative structure [63] or core-periphery structure [25], may be used in conjunction with the minimum description length principle [64] to investigate if certain models provide a better fit for certain parameter choices and to explore the performance of such models in terms of detecting specific implicitly planted structures. Another SBM variant that may be of interest for future research including the SCBM is the hierarchical SBM, e.g., [62], which was demonstrated to prevent underfitting of SBMs, to explore whether it would perceive the coexisting partitions as structures nested within each other or in the form of partitions appearing in the same posterior distribution of flat partitions. Finally, we conclude that future work around the theory of methods for mesoscale structure detection in networks should focus on improving existing methods to be able to identify coexisting structures. More broadly, and in line with recent work [11,37], we believe that researchers applying existing methods on real networks should focus on the possibility of discovering multiple dimensions of segmenting the network, rather than accepting unidimensional solutions, that may be even be averaged over "multimodal" partition distributions. In particular, possible contexts in which considering multiple plausible network divisions seem particularly important include the field of computational social science, which deals with the analysis of interaction dynamics in online public spaces. Appraising the coexistence of multiple types of structures in social media interaction networks, such as community/CP structures or even qualitatively different community partitions (maybe generated through nonaligned political dimensions, as has been recently shown on Twitter affiliations [65]), could have considerable benefits for researching online conversation dynamics. In this context, researchers should consider the coexistence of qualitatively different structures and be mindful of the detectability issues addressed here. In our simulations, we focus on the special case of equally sized blocks in both the planted partitions as well as the cross-partition and on the community-CP case. However, the benchmark model is flexible to a diverse range of structures of varying block sizes, degree distributions, and planted mesoscale structures. Future work may use this benchmark to analyze the recovery of ambiguity in networks of different sizes and expected degrees or with other

combinations of mesoscale structures. This work may also be extended by testing other types of detection algorithms (beyond SBM) on this benchmark model. Further extensions of the benchmark model itself could allow for more than two blocks in each planted partition, more than two planted partitions or a directed version of the model.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772743).

APPENDIX A: DETAILED MMSBM FORMULATION

In the original MMSBM formulation [49], each node belongs to all latent groups with certain probability, denoted—rather than by a one-hot binary block membership vector—by a *mixed membership* vector $\boldsymbol{\pi}_i$ for node i, in which element π_i^k denotes the probability of the node i belonging to block k and so $\sum_k \pi_i^k = 1$. Mixed membership vectors are drawn from a Dirichlet distribution for each node i, with some fixed parameter that is equal across all nodes. For each pair of nodes i and j, the block memberships r and s are drawn from a multinomial distribution parameterized by the nodes' mixed membership vectors; the existence of an edge between i and j is sampled from a Bernoulli distribution based on the predefined edge probability between blocks r and s. The generative process of the network is thus similar to a Bernoulli SBM, except that the block membership of a given node i is drawn repeatedly for each node j it is paired with. This means that nodes may belong to different blocks depending on the pairing that is considered and that they inherit connectivity patterns from multiple blocks. This process leads to the particular feature that the density at the "overlap" of multiple blocks is a weighted average of their individual densities [50].

For ease of explanation, we denote the two blocks in partition 1 by labels $\{a, b\}$ and the blocks in partition 2 by labels $\{c, d\}$. We define the block matrices denoting the expected edge densities within and between blocks by $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ in Eq. (A1a) and Eq. (A1b). Note that block matrices are symmetric since the graphs we are generating are undirected (for example, $\theta_{ab} = \theta_{ba}$) and—according to convention in undirected networks and to simplify calculations—elements on diagonals denote twice the within-block edge densities:

$$\boldsymbol{\theta}_1 = \begin{pmatrix} \theta_{aa} & \theta_{ab} \\ \theta_{ab} & \theta_{bb} \end{pmatrix}, \tag{A1a}$$

$$\boldsymbol{\theta}_2 = \begin{pmatrix} \theta_{cc} & \theta_{cd} \\ \theta_{cd} & \theta_{dd} \end{pmatrix}. \tag{A1b}$$

To fit the MMSBM formulation, we consider the blocks in the two planted partitions as four latent blocks $\{a, b, c, d\}$, but we constrain the generative process in a way that forces certain overlaps to be empty. In particular, the overlaps of block a or b in partition 1 and of block c or d in partition 2 are empty. This constraint is illustrated in Fig. 1(c), where the nodes in the overlaps are colored according to their block membership colors in the two planted partitions. Our mixed membership vectors thus take on the form $\boldsymbol{\pi}_i = \frac{1}{2}(\mathbf{b}_i^1, \mathbf{b}_i^2)$. For example, if $\boldsymbol{\pi}_i = \frac{1}{2}(1, 0, 1, 0)$, then node i belongs to blocks a and c with equal probability $\pi_i^a = \pi_i^c = \frac{1}{2}$. The constraint compared to general mixed membership vectors is therefore that nodes belong to exactly two blocks with equal probability and that $\boldsymbol{\pi}_i = \frac{1}{2}(1, 1, 0, 0)$ and $\pi_i = \frac{1}{2}(0, 0, 1, 1)$ are forbidden, since a node can never belong to block a and b (or to c and d). After drawing a mixed membership vector for each node i, we draw the block memberships r and s for each node pair i and j independently from their mixed membership vectors. We then place an edge between i and j according to some probability that depends on r and s.

1. Normalization

A first intuition might be to connect two nodes with probability θ_{rs} if $r, s \in \{a, b\}$ (or $r, s \in \{c, d\}$) and with zero probability for any other combination of r and s. However, since each node receives edges based on the connectivity of two blocks independently, this process will not be consistent with the connectivity of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. If we generate a network according to these probabilities, the expected edge density $\hat{\theta}_{rs}$ between two nodes i and j that belong to blocks r and s, respectively, is

$$\hat{\theta}_{rs} = \frac{1}{4}\sum_{r's'} \left(\pi_i^r \pi_j^s \theta_{rs} + \pi_i^u \pi_j^v \theta_{r's'}\right) = \frac{1}{4}\theta_{rs} + \frac{1}{16}\sum_{r's'} \theta_{r's'} \tag{A2}$$

for $r, s \in \{a, b\}$ and $r', s' \in \{c, d\}$ (or $r, s \in \{c, d\}$ and $r', s' \in \{a, b\}$). In order to accommodate for the connectivity of both planted partitions, we thus need to normalize the edge probabilities appropriately, so that $\hat{\theta}_{rs} = \theta_{rs}$. The additive edge probabilities prevent us from finding a single normalization constant x. Instead we may consider finding x_{rs} for each block pair, so that nodes are connected with probability $x_{rs}\theta_{rs}$. To guarantee $\hat{\theta}_{rs} = \theta_{rs}$ we thus need

$$\theta_{rs} = \frac{1}{4}x_{rs}\theta_{rs} + \frac{1}{16}\sum_{r's'} x_{r's'}\theta_{r's'}. \tag{A3}$$

Finding suitable x_{rs} requires us to solve an underdetermined system of six equations that is consistent as long as the sum of the probabilities in $\boldsymbol{\theta}_1$ equals the sum of probabilities in $\boldsymbol{\theta}_2$, and thus has infinitely many solutions. However, it turns out that this system of equations does not have any non-negative solutions for certain combinations of connectivity patterns planted in $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, in particular when the differences between block densities in both partitions are large (see Appendix A 2 for a proof).

An alternative way to normalize the additive edge probabilities is to determine a normalization constant for each combination of block pairs that two nodes can occupy in the two partitions. In other words, nodes i and j for which block memberships r and s have been drawn are connected with probability $x_{rsr's'}\theta_{rs}$, where r' and s' are the other two blocks that i and j are also members of. To determine the set of $\{x_{rsr's'}\}$ we rewrite the expected density as

$$\hat{\theta}_{rs} = \frac{1}{4}\sum_{r's'} x_{rsr's'}(\theta_{rs} + \theta_{r's'}) \tag{A4}$$

and set up a system of six equations so that $\hat{\theta}_{rs} = \theta_{rs}$ is satisfied—one for each of the upper triangular entries in $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$—and solve it for $\mathbf{x} = \{x_{rsr's'}\}$. This is, again, an underdetermined system of equations with infinitely many solutions, and here we can find non-negative solutions for all combinations of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ that we use in our simulations in Sec. IV. We can use a least squares solver to compute a minimum norm solution to the system [66]. However, for certain combinations of planted structures, the minimum norm solution returns negative values for some of the $\{x_{rsr's'}\}$. This means that for certain combinations of edge probabilities in the two planted structures, we must either use a non-negative least squares solver—-which may return solutions that include zero values—or a linear programming solver [67,68] to find strictly positive solutions.

2. Existence of non-negative solutions

For the simplified case where $\theta_{aa} = \theta_{bb}$ and $\theta_{cc} = \theta_{dd}$, we can use Farkas' lemma [69] to show that the system does not have any non-negative solutions for a certain combination of densities planted in the two structures—namely, if and only if $|\theta_{aa} - \theta_{ab}| + |\theta_{cc} - \theta_{cd}| > \frac{E}{n^2} = 2\rho$ where ρ is the overall density of the network.

To find the set of normalization constants, we need to solve equations $\theta_{rs} = \frac{1}{4}x_{rs}\theta_{rs} + \frac{1}{16}\sum_{uv} x_{uv}\theta_{uv}$ for $\mathbf{x} = \{x_{rs}\}$. Without loss of generality, we assume $\theta_{aa} = \theta_{bb}$ and $\theta_{cc} = \theta_{dd}$. We also start by assuming that $\theta_{aa} > \theta_{ab}$ and $\theta_{cc} > \theta_{cd}$. The system becomes $\mathbf{Ax} = \mathbf{y}$ with

$$\mathbf{A} = \begin{pmatrix} \frac{1}{4}\theta_{aa} & 0 & \frac{1}{8}\theta_{cc} & \frac{1}{8}\theta_{cd} \\ 0 & \frac{1}{4}\theta_{ab} & \frac{1}{8}\theta_{cc} & \frac{1}{8}\theta_{cd} \\ \frac{1}{8}\theta_{aa} & \frac{1}{8}\theta_{ab} & \frac{1}{4}\theta_{cc} & 0 \\ \frac{1}{8}\theta_{aa} & \frac{1}{8}\theta_{ab} & 0 & \frac{1}{4}\theta_{cd} \end{pmatrix} \tag{A5}$$

and $\mathbf{y} = (\theta_{aa}, \theta_{ab}, \theta_{cc}, \theta_{cd})$.

Written as a *theorem of alternatives*, Farkas' lemma states that exactly one of the following two statements are true for $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{x} \in \mathbb{R}^n$:

(1) $\exists \mathbf{x} \in \mathbb{R}^m : \mathbf{Ax} = \mathbf{y}$ and $\mathbf{x} \geqslant 0$

(2) $\exists \mathbf{v} \in \mathbb{R}^n : \mathbf{v}^T\mathbf{A} \geqslant 0$ and $\mathbf{v}^T\mathbf{y} \leqslant 0$.

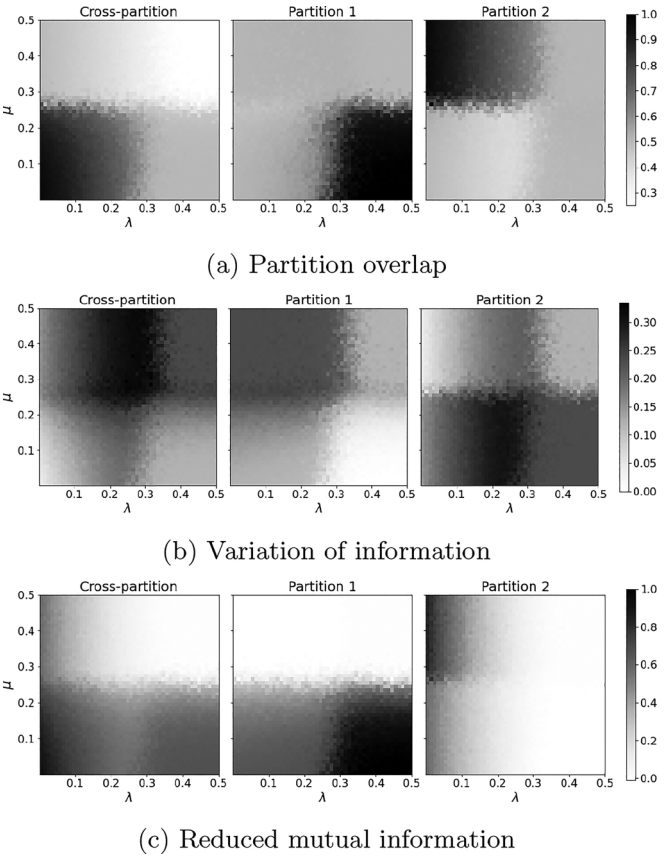
This means that if we can find a vector $\mathbf{v}$ for which the second alternative is always true, this implies that a non-negative solution $\mathbf{x}$ to the system cannot be found. If we choose $\mathbf{v} = (-1, 1, 0, 2)$, we have $\mathbf{v}^T\mathbf{A} = (0, \frac{1}{2}\theta_{ab}, 0, \frac{1}{2}\theta_{cd})$ and $\mathbf{v}^T\mathbf{y} = -\theta_{aa} + \theta_{ab} + 2\theta_{cd}$. Therefore, we have $\mathbf{v}^T\mathbf{y} < 0$ if and only if $\theta_{aa} - \theta_{ab} > 2\theta_{cd}$. Since $\frac{2E}{n^2} = 2\theta_{cc} + 2\theta_{cd}$, we can write

$$\theta_{aa} - \theta_{ab} > 2\theta_{cd}, \tag{A6a}$$

$$\theta_{aa} - \theta_{ab} > \tfrac{E}{n^2} - \theta_{cc} + \theta_{cd}, \tag{A6b}$$

$$\theta_{aa} - \theta_{ab} + \theta_{cc} - \theta_{cd} > \tfrac{E}{n^2} = \tfrac{4E}{N^2} = 2\rho, \tag{A6c}$$

where ρ is the overall density of the network. Since we have assumed that $\theta_{aa} > \theta_{ab}$ and $\theta_{cc} > \theta_{cd}$, Eq. (A6c) states that we do not have any non-negative solutions if the within-block densities are sufficiently larger than the between-block densities in both planted partitions. It is straightforward to find a vector $\mathbf{v}$ for all cases $\theta_{aa} > \theta_{ab}$ and $\theta_{cc} < \theta_{cd}$, $\theta_{aa} < \theta_{ab}$ and $\theta_{cc} < \theta_{cd}$, and $\theta_{aa} < \theta_{ab}$ and $\theta_{cc} > \theta_{cd}$, so that finally we can say that the system does not have any non-negative solutions if and only if $|\theta_{aa} - \theta_{ab}| + |\theta_{cc} - \theta_{cd}| > \frac{E}{n^2} = 2\rho$.



(a) Partition overlap

(b) Variation of information

(c) Reduced mutual information

FIG. 9. Mean partition overlap, variation of information and mutual information between the planted partitions and the partitions in the posterior distributions for graphs with $c = 10$.

In more qualitative terms, this means that when we induce large differences between block densities in both partitions, we cannot find appropriate normalization constants to solve the system of equations that would enable us to create a network in which the connectivities are consistent with both planted partitions. This limitation would severely restrict our ability to test the detectability of certain combinations of partitions. In fact, it would enable us to explore only half of the parameter space we explore in our simulations in Sec. IV, where we plant a bicommunity partition and a core-periphery partition by sweeping two parameters that determine the strength of each structure.

APPENDIX B: CONSTANT MULTIPLICATIVE FACTOR

In the two-partition SCBM with blocks labeled $\{a, b\}$ in partition 1 and $\{c, d\}$ in partition 2, each node is assigned to one of the cross-blocks $\boldsymbol{\Gamma} = \{(a, c), (a, d), (b, c), (b, d)\}$. For illustrative purposes, we start by creating the symmetric 4×4 matrix $\boldsymbol{\theta}'$, whose elements are the products of the elements of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ corresponding to the respective blocks in 1 and 2 that make up the cross-blocks in $\boldsymbol{\Gamma}$. Note that we drop the SCBM subscript on our 4×4 cross-partition matrices for ease of readability. Matrix $\boldsymbol{\theta}'$ is thus the Kronecker product of $\boldsymbol{\theta}_1$

and $\boldsymbol{\theta}_2$, where the order depends on the index set J,

$$\boldsymbol{\theta}' = \boldsymbol{\theta}_1 \otimes \boldsymbol{\theta}_2 = \begin{array}{c} \\ (a,c) \\ (a,d) \\ (b,c) \\ (b,d) \end{array} \begin{array}{c} \begin{array}{cccc} (a,c) & (a,d) & (b,c) & (b,d) \end{array} \\ \begin{pmatrix} \theta_{aa}\theta_{cc} & \theta_{aa}\theta_{cd} & \theta_{ab}\theta_{cc} & \theta_{ab}\theta_{cd} \\ \theta_{aa}\theta_{cd} & \theta_{aa}\theta_{dd} & \theta_{ab}\theta_{cd} & \theta_{ab}\theta_{dd} \\ \theta_{ab}\theta_{cc} & \theta_{ab}\theta_{cd} & \theta_{bb}\theta_{cc} & \theta_{bb}\theta_{cd} \\ \theta_{ab}\theta_{cd} & \theta_{ab}\theta_{dd} & \theta_{bb}\theta_{cd} & \theta_{bb}\theta_{dd} \end{pmatrix} \end{array} \tag{B1}$$

The most general form of defining the multiplicative factor is finding x_{uv} for each pair of cross-blocks $u = (r, r')$ and $v = (s, s')$, such that $B_{uv} = x_{uv}\nu_u\nu_v\theta'_{uv}$. If we fix $\nu_u = \nu = \frac{n}{2}$ for all cross-blocks u (and allow for self-loops) we have that the maximum possible number of edges between and within each group is $\nu^2 = \frac{n^2}{4}$, so we have

$$B_{uv} = x_{uv}\nu^2\theta'_{uv}. \tag{B2}$$

We want to find the normalization constants $\{x_{uv}\}$, such that the expected edge counts within and between blocks of both implicitly planted partitions 1 and 2 are equal to the sums of the expected edge counts of the overlaps that make up each of the original blocks. Therefore, we require

$$B_{1_{rs}} = n^2\theta_{1_{rs}} = \sum_{r's'} x_{uv}\nu^2\theta_{1_{rs}}\theta_{2_{r's'}} \tag{B3}$$

for blocks r and s in partition 1 and the same for blocks in partition 2. We can rewrite this as

$$n^2 = \frac{1}{4}\sum_{r's'} x_{uv}n^2\theta_{2_{r's'}} \tag{B4}$$

and therefore also (for the elements of $\mathbf{B}_2$)

$$n^2 = \frac{1}{4}\sum_{rs} x_{uv}n^2\theta_{1_{rs}}, \tag{B5}$$

which is satisfied for a constant $x = x_{uv} = 2n^2/E = 1/\rho$, since

$$\sum_{rs} n^2\theta_{1_{rs}} = \sum_{r's'} n^2\theta_{2_{r's'}} = 2E. \tag{B6}$$

APPENDIX C: VARYING BLOCK SIZES

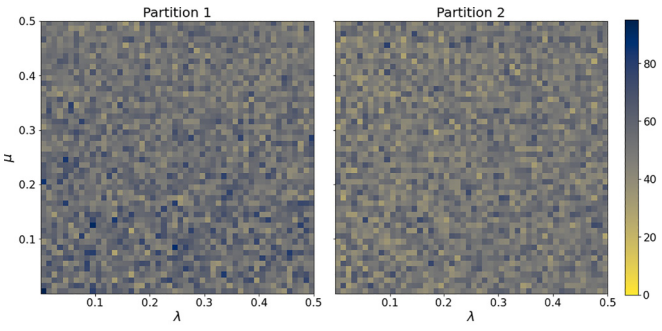
In the case of equal block sizes in the implicitly planted partitions 1 and 2, setting $2n = N$, but allowing for varying cross-block sizes, we can write $\nu_1 = \nu_{ac} = \nu_{bd}$ and $\nu_2 = \nu_{ad} = \nu_{bc}$, where ν_{ac} is the number of nodes in block (a, c). In this case, we have $\nu_1 + \nu_2 = n$. We no longer have the same maximum possible number of edges within and between each of the block pairs. Therefore, instead of (B2), we have

$$B_{uv} = x_{uv}\nu_u\nu_v\theta'_{uv}, \tag{C1}$$

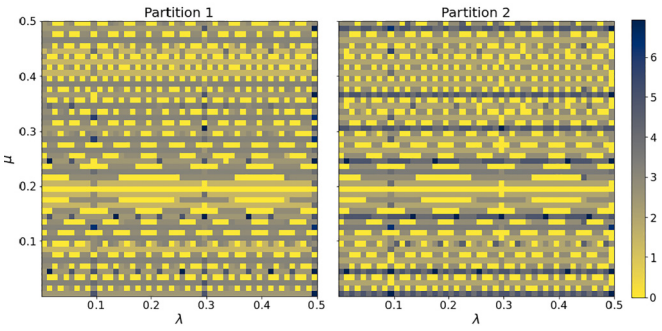
where $\nu_u\nu_v$ is one of $\{\nu_1^2, \nu_1\nu_2, \nu_2^2\}$.

Let us assume that a constant $x = x_{uv}$ does also exist for $n = \frac{N}{2}$, $\nu_1 \neq \nu_2$ and that we are planting nontrivial partitions where the probability of edge placement between blocks is not uniform across block pairs. To satisfy the connectivity in partitions 1 and 2, we now have

$$B_{1_{rs}} = n^2 = x\sum_{r's'} \nu_u\nu_v\theta_{2_{r's'}}. \tag{C2}$$



(a) Canonical model



(b) Microcanonical model

FIG. 10. Mean difference (Frobenius norm) between planted and generated edge count matrices in networks, for $c = 5$.

Specifically, for the within- and between-block densities of partition 1 to be satisfied, we need

$$\frac{n^2}{x} = \theta_{cc}\nu_1^2 + \theta_{dd}\nu_2^2 + 2\theta_{cd}\nu_1\nu_2, \tag{C3a}$$

$$\frac{n^2}{x} = \theta_{cd}\nu_1^2 + \theta_{cd}\nu_2^2 + \theta_{cc}\nu_1\nu_2 + \theta_{dd}\nu_1\nu_2, \tag{C3b}$$

$$\frac{n^2}{x} = \theta_{dd}\nu_1^2 + \theta_{cc}\nu_2^2 + 2\theta_{cd}\nu_1\nu_2, \tag{C3c}$$

where we have dropped the subscript for partition number 2 on θ for easier readability. Setting equal the first and last of these equations, we get

$$(\theta_{cc} - \theta_{dd})\left(\nu_1^2 - \nu_2^2\right) = 0. \tag{C4a}$$

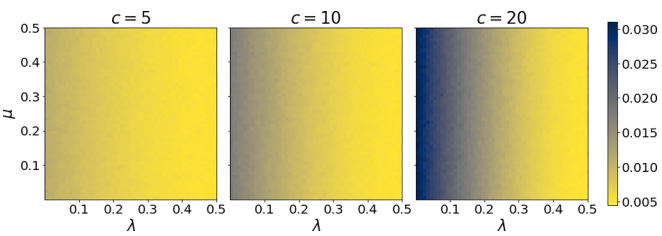


FIG. 11. Mean normalized degree variance for graphs with varying expected degrees c, only showing the graphs generated by the canonical model to illustrate the degree variance introduced for low values of λ.

Since $\nu_1 \neq \nu_2$, we have $\theta_{cc} = \theta_{dd}$ and therefore [from Eqs. (C3a) and (C3b)], we then have

$$\theta_{cc}\nu_1^2 + \theta_{cc}\nu_2^2 + 2\theta_{cd}\nu_1\nu_2 = \theta_{cd}\nu_1^2 + \theta_{cd}\nu_2^2 + 2\theta_{cc}\nu_1\nu_2, \tag{C5a}$$

$$(\theta_{cc} - \theta_{cd})(\nu_1 - \nu_2)^2 = 0, \tag{C5b}$$

and hence $\theta_{cc} = \theta_{cd} = \theta_{dd}$. Clearly, taking the same steps for $\mathbf{B}_2$ gives $\theta_{aa} = \theta_{ab} = \theta_{bb}$. This is a contradiction to our set of assumptions and therefore such a constant x does not exist (apart from the trivial case where we plant a random graph).

APPENDIX D: SOLVING UNDERDETERMINED SYSTEM OF EQUATIONS

For general cases in which each block in partitions 1 and 2 can take on any size, we need to determine a set of normalization constants $\{x_{uv}\}$ to create the final block matrix $\boldsymbol{\theta}_{\text{SCBM}} = \{\theta_{uv}\} = \{x_{uv}\theta_{1_{rs}}\theta_{2_{r's'}}\}$. In this case, the maximum possible number of edges between and within each block—in both the two planted partitions as well as the cross-partition—may differ for each block pair. The elements of the expected edge count matrices are therefore $B_{p_{rs}} = n_r r_s \theta_{p_{rs}}$ for $p \in \{1, 2\}$, and $B_{uv} = \nu_u\nu_v\theta_{uv} = \nu_u\nu_v x_{uv}\theta'$. The set of constants $\{x_{uv}\}$ need to be chosen in such a way that the elements of $\mathbf{B}_1$ and $\mathbf{B}_2$ are satisfied (on the diagonal and upper triangular), as is presented in Eqs. (D1a)–(D1f). We thus have a total of $2 \times \frac{K_s(K_s+1)}{2} = 6$ equations and $\frac{K(K+1)}{2} = 10$ unknowns. On the left-hand side, we have $b_{rs} = n_r n_s \theta_{rs} = B_{p_{rs}}$ and we have dropped the subscripts for partitions 1 and 2 for legibility. On the right-hand side, we have $\mathbf{B}'_{uv}$, where $B'_{uv} = \{\nu_u\nu_v\theta_{1_{rs}}\theta_{s_{r's'}}\}$ and subscripts are written as the indices of the cross-blocks; element $x_{11} = x_{(a,c)(a,c)}$, for example, is the normalization constant for within-block edges in block (a, c) (overlap of block a in s_1 and block c in s_2):

$$b_{aa} = x_{11}\hat{B}_{11} + 2x_{12}\hat{B}_{12} + x_{22}\hat{B}_{22}, \tag{D1a}$$

$$b_{ab} = x_{13}\hat{B}_{13} + x_{14}\hat{B}_{14} + x_{23}\hat{B}_{23} + x_{24}\hat{B}_{24}, \tag{D1b}$$

$$b_{bb} = x_{33}\hat{B}_{33} + 2x_{34}\hat{B}_{34} + x_{44}\hat{B}_{44}, \tag{D1c}$$

$$b_{cc} = x_{11}\hat{B}_{11} + 2x_{13}\hat{B}_{13} + x_{33}\hat{B}_{33}, \tag{D1d}$$

$$b_{cd} = x_{12}\hat{B}_{12} + x_{14}\hat{B}_{14} + x_{32}\hat{B}_{32} + x_{34}\hat{B}_{34}, \tag{D1e}$$

$$b_{dd} = x_{22}\hat{B}_{22} + 2x_{24}\hat{B}_{24} + x_{44}\hat{B}_{44}. \tag{D1f}$$

Equations (D1a)–(D1f) are an underdetermined system of six linear equations with ten unknowns. We can write it in matrix form as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{D2}$$

with

$$\mathbf{x} = (x_{11}, x_{12}, \ldots, x_{32}, x_{33}, x_{44}), \tag{D3}$$

$$\mathbf{b} = (b_{aa}, b_{ab}, b_{bb}, b_{cc}, b_{cd}, b_{dd}), \tag{D4}$$

so that $\mathbf{x} \in \mathbb{R}^{10}$, $\mathbf{b} \in \mathbb{R}^6$ and where $\mathbf{A} = \{a_{ij}\} \in \mathbb{R}^{6\times10}$ is the coefficient matrix with elements of $\hat{\mathbf{B}}$ respecting Eqs. (D1a)–(D1f). According to the Rouché-Capelli theorem, we know that such an underdetermined system has an infinite number of solutions if and only if the rank of its coefficient matrix is equal to the rank of its augmented matrix $\mathbf{W} = [\mathbf{A}|\mathbf{b}] \in \mathbb{R}^{6\times11}$. This is always true for two well-defined connectivity matrices $\mathbf{B}_1$ and $\mathbf{B}_2$; clearly, the total number of edges must be the same in the two planted partitions, and we can show that $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{W})$ if and only if $b_{aa} + b_{ab} + b_{bb} = b_{cc} + b_{cd} + b_{dd}$.

In general, an underdetermined linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}^{m\times n}$ where $m < n$, does not have a unique solution $\mathbf{x}$. Since the system is underconstrained, it has an infinitude of solutions, if it has any solutions at all. A popular method for solving under- (or over-) constrained systems of linear systems of equations is called *least squares* method. The idea behind the least squares method is to find a solution $\mathbf{x}$ which minimizes the squared Euclidean norm of the residual $r(\mathbf{x}) = \mathbf{b} - \mathbf{A}\mathbf{x}$. In other words, we want to find $\mathbf{x}$ that minimizes $\phi(\mathbf{x}) = ||r(\mathbf{x})||_2^2 = ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2$, which can be done by obtaining $\mathbf{x}$ such that $\nabla\phi(\mathbf{x}) = 0$. From this, we obtain the so-called *normal equations* $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ which can be solved analytically if $\mathbf{A}^T\mathbf{A}$ is invertible. In our case, $\mathbf{A}^T\mathbf{A} \in \mathbb{R}^{n\times n}$ has rank at most m, where $m < n$, and is therefore singular. This means that, in this underdetermined case, the normal equations cannot be solved analytically. Instead, we can find the particular least squares solution that minimizes the Euclidean norm $||\mathbf{x}||_2$ (or its square) with the constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$. When there are no other constraints, this *minimum norm solution* $\hat{\mathbf{x}}$ can be found by computing the singular value decomposition (SVD) in order to compute the Moore-Penrose pseudoinverse $\mathbf{A}^+$ of matrix $\mathbf{A}$. The minimum norm solution can then be calculated as $\bar{\mathbf{x}} = \mathbf{A}^+\mathbf{b}$, always exists and is unique [70].

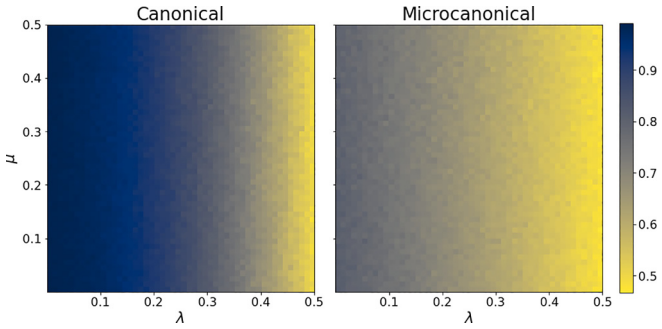


FIG. 12. Proportion of nodes correctly classified into core and periphery blocks by assigning nodes with degree higher than the expected degree $c = 20$ to the core and all others to the periphery.

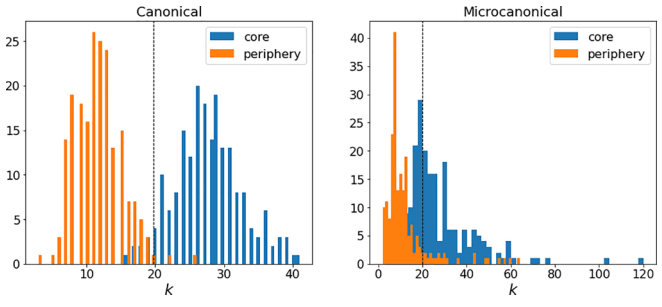
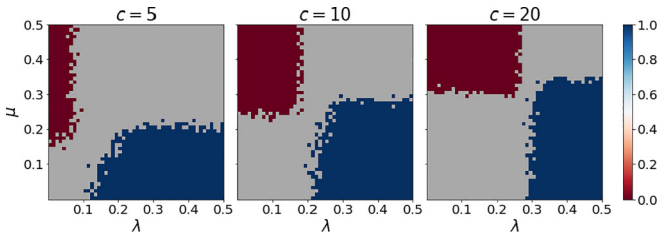
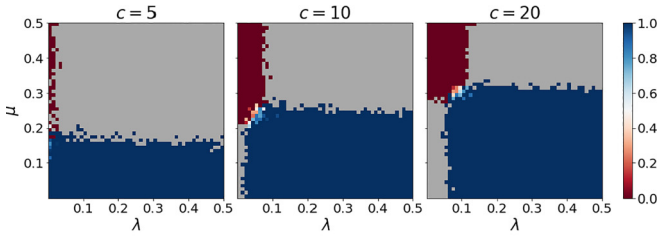


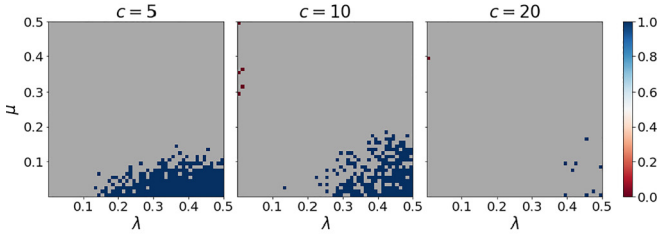
FIG. 13. Degree distributions of core nodes vs periphery nodes in the canonical and microcanonical model with overall expected degree (dashed line), for $\mu = 0.1$, $\lambda = 0.1$, and $c = 20$.



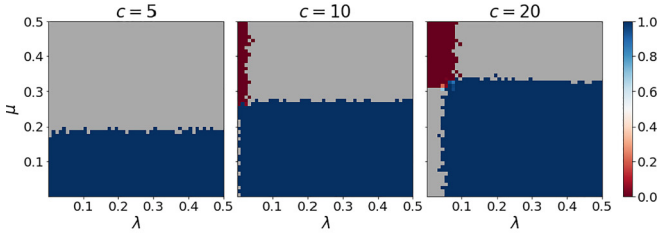
(a) NDC, Canonical model.



(b) DC, Canonical model.

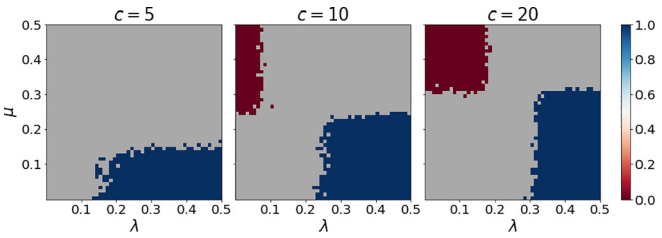


(c) NDC, Microcanonical model.

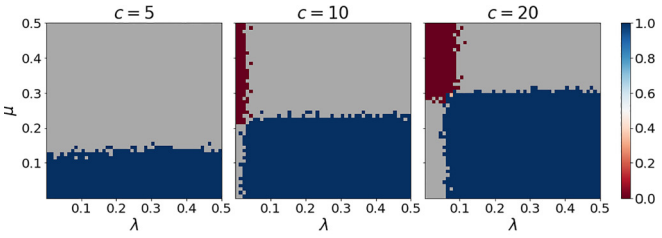


(d) DC, Microcanonical model.

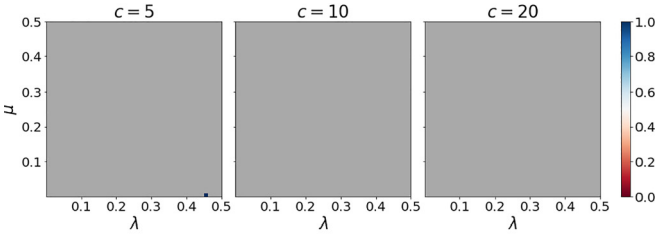
FIG. 14. Fraction α of recovered bicommunity partitions out of all successfully recovered partitions for varying c for $\omega = 0.85$; at $\alpha = 1$ (resp. $\alpha = 0$) only the bicommunity (resp. CP) structure is detected.



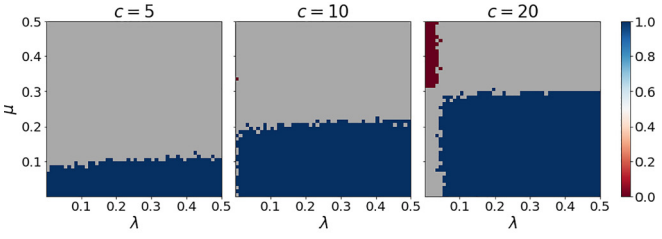
(a) NDC, Canonical model.



(b) DC, Canonical model.



(c) NDC, Microcanonical model.



(d) DC, Microcanonical model.

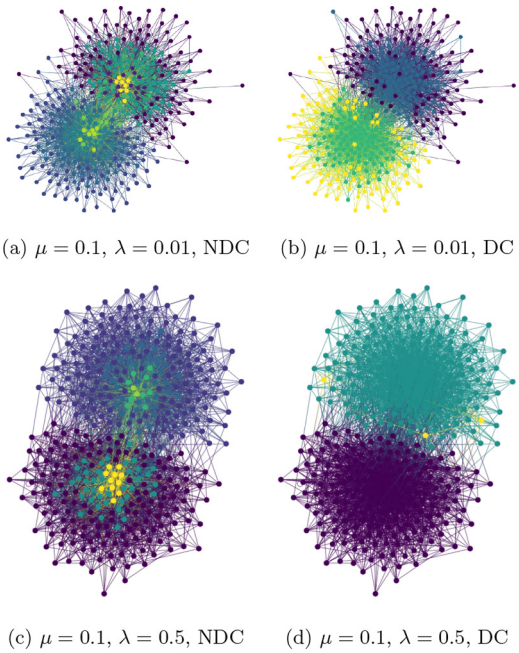
FIG. 15. Fraction α of recovered bicommunity partitions out of all successfully recovered partitions for varying c for $\omega = 0.95$; at $\alpha = 1$ (resp. $\alpha = 0$) only the bicommunity (resp. CP) structure is detected.

APPENDIX E: ALTERNATIVE SIMILARITY MEASURES

To demonstrate the robustness of the partition overlap as our similarity measure, we compare it to two alternative measures, variation of information [61] and reduced mutual information [60]. We use these two measures to calculate the partition similarity (or distance, in the case of variation of information, which is largest for partitions with the largest difference) for one particular set of simulations where $N = 400$, $c = 10$ and where graphs were generated by the canonical model and partitions inferred using the non-degree-corrected SBM. We show the results in Fig. 9. While there are some subtle differences in the mean similarity values, the regions in which the detectability of the different partitions appears to change are located in the same areas of the (λ, μ) space.

APPENDIX F: CHARACTERISTICS OF GENERATED GRAPHS

To ensure that the variability in the recovered partitions is not in fact due to the graphs we generate, we compare the planted edge count matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ with the actual edge counts in the generated graphs, $\mathbf{M}_1$ and $\mathbf{M}_2$ by calculating $||\mathbf{B}_1 - \mathbf{M}_1||_F$ and $||\mathbf{B}_2 - \mathbf{M}_2||_F$ for all values of μ and λ. Figure 10(a) shows the mean Frobenius norm for $\mu \in [0.01, 0.5]$ and $\lambda \in [0.01, 0.5]$ for graphs generated by the canonical model. Figure 10(b) shows the same plot for the microcanonical SBM. These figures show the distances for graphs with expected degree $c = 5$; we note that the patterns for the higher values of c are similar. We observe that the distances between the planted and generated edge count matrices in the microcanonical case are, by definition, much



(a) $\mu = 0.1$, $\lambda = 0.01$, NDC (b) $\mu = 0.1$, $\lambda = 0.01$, DC

(c) $\mu = 0.1$, $\lambda = 0.5$, NDC (d) $\mu = 0.1$, $\lambda = 0.5$, DC

FIG. 16. Example networks for a fixed value of $\mu = 0.1$, with $\lambda = 0.01$ (top row) and $\lambda = 0.5$ (bottom row).

lower than in the traditional case. The nonrandom patterns we observe in the microcanonical case are due to rounding that is necessary to create the edge count matrix $\mathbf{B}$ which is used to generate networks in this case (while in the canonical case the edge probabilities, rather than counts, are used).

Figure 11 shows the rescaled version of Fig. 4. Here we plot the mean normalized degree variance for graphs generated by the canonical model, to illustrate the higher variance introduced for lower values of λ, especially for higher values of c.

In Fig. 12 we plot the proportion of nodes correctly classified (according to the planted core and periphery blocks) by assigning nodes with above average degree to the core and those with below average degree to the periphery, for graphs with expected degree $c = 20$. In Fig. 13 we show two example degree distributions for $\mu = 0.1$, $\lambda = 0.1$, and $c = 20$.

APPENDIX G: STRICTER PARTITION-OVERLAP THRESHOLDS

Figures 14 and 15 illustrate the detection of the bicommunity and CP structures, as well as their coexistence, for the two SBM variants and for partition overlap $\omega = 0.85$ and $\omega = 0.95$ respectively.

APPENDIX H: EXAMPLE NETWORKS WITH HIGH DEGREE HETEROGENEITY

In Fig. 16 we plot two example networks generated by the microcanonical model. The network visualizations were generated by the graph-tool Python library [56]. We fix $\mu = 0.1$ for both networks, and we create one graph with a strong planted CP structure ($\lambda = 0.01$) and one for which no CP structure at all is planted explicitly through the edge count matrices ($\lambda = 0.5$). Figures 16(a) and 16(c) show an example of the type of partition frequently recovered by NDC for $\lambda = 0.01$ and $\lambda = 0.5$, respectively. Figures 16(b) and 16(d) show the same but for the DC variant. We observe that DC recovers the cross-partition for $\lambda = 0.01$ and the bicommunity partition for $\lambda = 0.5$, in line with the equivalent results for graphs generated by the microcanonical model and with what we explicitly planted. NDC, however, (which has higher model evidence) detects a similar structure for $\lambda = 0.01$ and $\lambda = 0.5$: a two-block partition, where each block contains a core and multiple layered peripheries. The difference between the two detected partitions is the number of layers in the core-periphery structures within each block and the size of the outer periphery. Due to these differences, the partition detected for $\lambda = 0.01$ is more similar to the cross-partition, while that detected for $\lambda = 0.5$ is more similar to the bicommunity partition.

[1] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, Internet Math. **6**, 29 (2009).
[2] M. Rosvall, J.-C. Delvenne, M. T. Schaub, and R. Lambiotte, in *Advances in Network Clustering and Blockmodeling* (John Wiley & Sons, Hoboken, 2019), pp. 105–119.
[3] P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, SIAM Rev. **59**, 619 (2017).
[4] X. Zhang, T. Martin, and M. E. J. Newman, Phys. Rev. E **91**, 032803 (2015).
[5] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).
[6] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[7] M. E. J. Newman, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).
[8] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).
[9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech.: Theory Exp. (2008) P10008.
[10] T. S. Evans, J. Stat. Mech.: Theory Exp. (2010) P12037.
[11] T. P. Peixoto, Phys. Rev. X **11**, 021003 (2021).
[12] L. Peel, D. B. Larremore, and A. Clauset, Sci. Adv. **3**, e1602548 (2017).
[13] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. E **84**, 066106 (2011).
[14] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. Lett. **107**, 065701 (2011).
[15] A. Condon and R. M. Karp, Random Struct. Algor. **18**, 116 (2001).
[16] R. D. Alba, J. Math. Sociol. **3**, 113 (1973).
[17] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[18] M. E. J. Newman, Phys. Rev. E **88**, 042822 (2013).
[19] B. Karrer and M. E. J. Newman, Phys. Rev. E **83**, 016107 (2011).
[20] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).
[21] S. P. Borgatti and M. G. Everett, Social Netw. **21**, 375 (2000).
[22] P. Holme, Phys. Rev. E **72**, 046111 (2005).

[23] S. H. Lee, M. Cucuringu, and M. A. Porter, Phys. Rev. E **89**, 032810 (2014).
[24] M. Cucuringu, P. Rombach, S. H. Lee, and M. A. Porter, Eur. J. Appl. Math. **27**, 846 (2016).
[25] R. J. Gallagher, J.-G. Young, and B. F. Welles, Sci. Adv. **7**, eabc9800 (2021).
[26] B. Yan and J. Luo, Netw. Sci. **7**, 70 (2019).
[27] B. Tunç and R. Verma, PLoS ONE **10**, e0143133 (2015).
[28] S. Kojaku and N. Masuda, Phys. Rev. E **96**, 052313 (2017).
[29] R. D. Luce and A. D. Perry, Psychometrika **14**, 95 (1949).
[30] S. B. Seidman, Social Netw. **5**, 269 (1983).
[31] R. L. Breiger, S. A. Boorman, and P. Arabie, J. Math. Psychol. **12**, 328 (1975).
[32] M. Girvan and M. E. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).
[33] P. W. Holland, K. B. Laskey, and S. Leinhardt, Social Netw. **5**, 109 (1983).
[34] T. P. Peixoto, Phys. Rev. E **95**, 012317 (2017).
[35] A. Lancichinetti and S. Fortunato, Sci. Rep. **2**, 336 (2012).
[36] A. Tandon, A. Albeshri, V. Thayananthan, W. Alhalabi, and S. Fortunato, Phys. Rev. E **99**, 042301 (2019).
[37] A. Kirkley and M. E. J. Newman, Commun. Phys. **5**, 1 (2022).
[38] C. Moore, arXiv:1702.00467 (2017).
[39] E. Abbe, J. Machine Learn. Res. **18**, 6446 (2017).
[40] J. Reichardt and M. Leone, Phys. Rev. Lett. **101**, 078701 (2008).
[41] R. R. Nadakuditi and M. E. J. Newman, Phys. Rev. Lett. **108**, 188701 (2012).
[42] X. Zhang, R. R. Nadakuditi, and M. E. J. Newman, Phys. Rev. E **89**, 042816 (2014).
[43] F. Radicchi, Phys. Rev. E **88**, 010801(R) (2013).
[44] F. Lorrain and H. C. White, J. Math. Sociol. **1**, 49 (1971).
[45] H. C. White, S. A. Boorman, and R. L. Breiger, Am. J. Sociol. **81**, 730 (1976).
[46] M. G. Everett and S. P. Borgatti, J. Math. Sociol. **19**, 29 (1994).
[47] M. B. Hastings, Phys. Rev. E **74**, 035102(R) (2006).
[48] T. P. Peixoto, Phys. Rev. X **4**, 011047 (2014).
[49] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, Adv. Neural Info. Proc. Syst. **21** (2008).
[50] T. P. Peixoto, Phys. Rev. X **5**, 011033 (2015).
[51] T. P. Peixoto, Phys. Rev. E **92**, 042807 (2015).
[52] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 056117 (2009).
[53] M. Bazzi, L. G. S. Jeub, A. Arenas, S. D. Howison, and M. A. Porter, Phys. Rev. Res. **2**, 023100 (2020).
[54] B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander, SIAM Rev. **60**, 315 (2018).
[55] J. Lin, IEEE Trans. Inf. Theory **37**, 145 (1991).
[56] T. P. Peixoto, https://doi.org/10.6084/m9.figshare.1164194.
[57] H. W. Kuhn, Naval Res. Logistics **2**, 83 (1955).
[58] J. Munkres, J. Soc. Ind. Appl. Math. **5**, 32 (1957).
[59] S. Wagner and D. Wagner, https://publikationen.bibliothek.kit.edu/1000011477.
[60] M. E. J. Newman, G. T. Cantwell, and J.-G. Young, Phys. Rev. E **101**, 042304 (2020).
[61] M. Meilă, J. Multivariate Anal. **98**, 873 (2007).
[62] T. P. Peixoto, Phys. Rev. X **12**, 011004 (2022).
[63] L. Zhang and T. P. Peixoto, Phys. Rev. Res. **2**, 043271 (2020).
[64] T. P. Peixoto, Phys. Rev. Lett. **110**, 148701 (2013).
[65] P. Ramaciotti Morales, in *Complex Networks and Their Applications XI* (Springer International Publishing, Cham, 2023), pp. 176–189.
[66] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems* (SIAM, Philadelphia, 1995).
[67] G. Dantzig, *Linear Programming and Extensions* (Princeton University Press, Princeton, 1963).
[68] Q. Huangfu and J. J. Hall, Math. Program. Comput. **10**, 119 (2018).
[69] J. Farkas, J. Reine Angew. Math. (Crelles J.) **1902**, 1 (1902).
[70] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).