Hierarchical community structure in networks

Michael T. Schaub,[1,*] Jiaze Li,[2] and Leto Peel[2,†]

[1]*Department of Computer Science, RWTH Aachen University, 52074 Aachen, Germany*
[2]*Department of Data Analytics and Digitalisation, School of Business and Economics, Maastricht University, 6211 LM Maastricht, The Netherlands*

(Received 24 October 2020; revised 25 August 2021; accepted 24 April 2023; published 23 May 2023)

Modular and hierarchical community structures are pervasive in real-world complex systems. A great deal of effort has gone into trying to detect and study these structures. Important theoretical advances in the detection of modular have included identifying fundamental limits of detectability by formally defining community structure using probabilistic generative models. Detecting hierarchical community structure introduces additional challenges alongside those inherited from community detection. Here we present a theoretical study on hierarchical community structure in networks, which has thus far not received the same rigorous attention. We address the following questions. (1) How should we define a hierarchy of communities? (2) How do we determine if there is sufficient evidence of a hierarchical structure in a network? (3) How can we detect hierarchical structure efficiently? We approach these questions by introducing a definition of hierarchy based on the concept of stochastic externally equitable partitions and their relation to probabilistic models, such as the popular stochastic block model. We enumerate the challenges involved in detecting hierarchies and, by studying the spectral properties of hierarchical structure, present an efficient and principled method for detecting them.

DOI: 10.1103/PhysRevE.107.054305

I. INTRODUCTION

Hierarchical organization has been a central theme in the study of complex systems, dating back to the seminal work of Herbert Simon [1], who observed that a large proportion of complex systems exhibit hierarchical structure. Decomposing a complex system into such a hierarchy provides an interpretable summary, or coarse-grained description of the system at multiple resolutions. As networks have become ubiquitous for modeling complex systems, these ideas have re-emerged as the identification of hierarchical groups, or *communities*, of nodes within a network [2–5]. Community detection in networks has received a lot of attention because it can reveal important insights about social [6–8] and biological [8–11] systems, among others. A hierarchical description of communities provides the additional utility that it enables a consistent multiscale description, linking the organizational structure of a system across multiple resolutions. Hierarchical communities thereby circumvent a prominent issue of community detection, namely, deciding an appropriate resolution [12–14] or number of communities to detect [15,16]. On the other hand, detecting hierarchical communities inherits, and even exacerbates, many of the theoretical and computational challenges of detecting network communities at a single scale. Specifically, major challenges for detecting hierarchical communities are (i) how should we define a hierarchy of communities, (ii) how should we determine if a hierarchical structure exists in a network, and (iii) how can we detect hierarchical structure efficiently? Recently, we have seen important developments in the theory of community detection and its limitations [17–21] (see also Refs. [22,23] for reviews). Here we lay the foundations for developing such theory for detecting hierarchical community structure in networks.

The notion of hierarchy in networks is widespread and has been discussed from a plethora of different perspectives [24]. For instance, if edges denote some type of flow (e.g., information, data, mass, nutrients, money) this may induce a hierarchy among the nodes [25,26] in which nodes higher up in the hierarchy have more links directed towards nodes at lower levels of the hierarchy (or vice versa, depending on the convention of the directionality). To be clear, these types of nodal rankings are not the hierarchies we are looking for. Rather, we are interested in the hierarchical organization of community structure, i.e., communities that are again composed of communities etc. Existing models and methods for detecting hierarchical structure are often constrained to find dense assortative community structures [27–29]. Here we consider general probabilistic descriptions of mesoscopic hierarchical group structures, which can be combinations of assortative and disassortive structure.

There are currently many methods available that perform “hierarchical” community detection. Some methods are algorithmically hierarchical [30–32] and produce a hierarchy as a by-product and without guarantees of hierarchical structure in the network. Yet another class of methods involve fitting a hierarchical model [2–5,33]. However, in some cases, the design of these models have been motivated by objectives other than detecting hierarchies such as to produce networks with certain statistical properties [33] or to identify communities beyond the resolution limit [4]. Consequently, whenever we use one of these approaches (either a hierarchical algorithm or hierarchical model), we run the risk of identifying

[*] michael.schaub@rwth-aachen.de
[†] l.peel@maastrichtuniversity.nl

©2023 American Physical Society

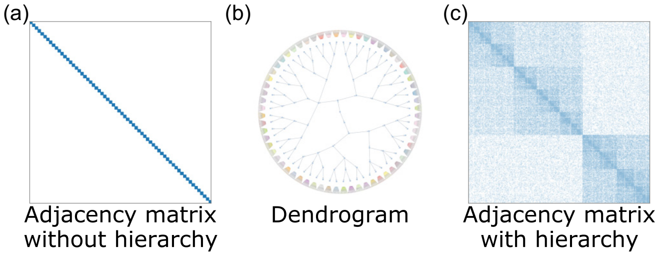

FIG. 1. A hierarchical model does not guarantee hierarchical community structure. (a) An adjacency matrix of a network with 64 groups of fully connected nodes (cliques), each containing ten nodes each. This network contains an unambiguously "flat" partition that contains no hierarchy. (b) The dendrogram representing the hierarchy found by detecting communities using a hierarchical model. (c) An adjacency matrix containing hierarchically structured block densities that is consistent with the dendrogram structure in (b).

a hierarchy that has greater complexity than the data can support.

To demonstrate this point, Fig. 1 illustrates a network containing 64 cliques that each contain ten nodes. It is relatively uncontroversial to suggest that the desired output of a community detection algorithm for this network would be to recover those sixty-four cliques as communities. Furthermore, because the cliques are structurally identical, any hierarchical grouping is compatible—any clique can be swapped with any other, all putative hierarchical configurations are effectively equivalent and there exists no preferred hierarchical grouping. Naïvely applying a hierarchical community detection method may produce a hierarchical clustering, as shown in Fig. 1(b). We can consider this detection of superfluous hierarchical levels as analogous to identifying spurious communities in an Erdős-Rényi random graph.

These issues typically arise when we simply optimize an objective function, e.g., maximizing modularity, likelihood or posterior probability. For instance, the maximum *a posteriori* solution may contain multiple hierarchical levels that provide a more compact description within the chosen model class, i.e., a "simpler" description of the data, and is therefore optimal with respect to chosen objective. However, this notion of simplicity conflicts with the intuition that there is no further structure beyond partitioning the network into sixty-four groups. Note that this is not to say that the maximum a-posteriori solution is bad, as it does present a plausible model that is compatible with the observed data, but rather that it presents an unintuitive interpretation of the hierarchical structure in the data. Some solutions to this problem exist in the realm of Bayesian inference, where we can take an average or form a consensus according to a distribution over solutions. Such solutions have been successfully demonstrated for both the regular [34,35] and hierarchical [2,36] variants of the community detection problem. However, these methods of statistical inference can be computationally demanding. Previous approaches either employ Markov chain Monte Carlo methods [2,4], for which convergence can be slow and difficult to diagnose, or rely on approximate heuristics that scale quadratically with the number of nodes in the network [3] and are thus limited to relatively small networks. Recently, however, fast spectral methods based on the nonbacktracking [37] and Bethe Hessian [38] operators have been developed that can efficiently detect communities right down to the theoretical limit of detectability [37].

Spectral algorithms have also been studied in the context of hierarchical communities. [5,39–41]. For instance, White and Smyth [31] and Newman [32] present spectral algorithms based on the modularity matrix that recursively bipartition a network. These algorithms output a hierarchy in the form of a binary dendrogram, but with the goal of simply recovering a single partition of the network. Lyzinski [5] analyze the performance of spectral algorithms under a hierarchical generative model based on a random dot product graph model. Local spectral algorithms have also been shown to provide good solutions when optimizing conductance based scores [42–44], which are of particular interest for very large graphs in case we do not need to partition the graph as a whole.

In this work, we propose a number of important theoretical advances for the detection of hierarchical communities. We first provide a definition of hierarchical communities by introducing the concept of stochastic externally equitable partitions and drawing a connection to the popular stochastic block model and various node equivalence classes (Sec. II). Second, we discuss specific challenges that pertain to the detection of hierarchical communities with a specific focus on identifiability issues, which demonstrate that even well-defined hierarchies do not have a unique representation (Sec. III). Then we turn our attention to the spectral properties of networks with planted hierarchical structures. Using these spectral properties, we develop an efficient method for detecting if a hierarchy of communities exists and identifying a hierarchy when it is present (Sec. IV). We conduct numerical experiments that demonstrate the efficacy of our approach on synthetic networks (Sec. V) and real-world networks (Sec. VI). Finally we conclude with a discussion of possible extensions of our work, including theoretical consideration and extensions to other type of network models.

II. HIERARCHICAL STRUCTURE IN NETWORKS

Before we can detect hierarchies, it is necessary to define precisely what we mean by a hierarchical structure. Any hierarchy can be represented as a rooted tree, sometimes referred to as a dendrogram. The root of this tree represents the group of all nodes in the network. Starting from the root, at each branch of the dendrogram each parent group is partitioned into child subgroups (see Fig. 2 for a schematic example).

In hierarchical community detection, as considered here, we aim to identify groups of similar nodes in a network, such that with each further subdivision of the nodes, the resulting groups should contain increasingly similar nodes. Each subgroup should therefore also have inherited certain similarities from its parent group. A relevant way to define similarity is in terms of *stochastically equivalent* nodes, i.e., groups of nodes r and s such that any node in group r has the same probability $\mathbf{\Omega}_{rs}$ of linking to any node in group s. In this setting one can represent the community structure of a network with n nodes using the stochastic block model (SBM) [45,46]. The SBM defines the probability of a link existing between two nodes depending on their community assignment. We represent this

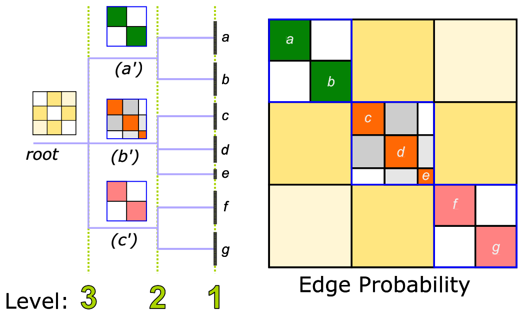

FIG. 2. Schematic representation of the link probabilities in a hierarchical graph model. The dendrogram on the left represents the hierarchical partition associated with the hierarchical organization of the edge probabilities on the right. At each branch of the dendrogram the link probabilities pattern within the diagonal blocks is refined, as indicated by the smaller block patterns on branch (a')–(c'). The resulting in an overall edge probability pattern is shown on the right.

group assignment as a group indicator matrix $\boldsymbol{H} \in \{0, 1\}^{n\times k}$, in which $\boldsymbol{H}_{ir} = 1$ if node i is assigned to group r and $\boldsymbol{H}_{ir} = 0$ otherwise. Then the probability of nodes i and j being linked is given by

$$P[\boldsymbol{A}_{ij} = 1] = \boldsymbol{H}_{i\cdot}\boldsymbol{\Omega}\boldsymbol{H}_{j\cdot}^{\top}, \tag{1}$$

where $\boldsymbol{H}_{i\cdot}$ is the ith row of $\boldsymbol{H}$ and $\boldsymbol{A} \in \{0, 1\}^{n\times n}$ is the adjacency matrix in which $\boldsymbol{A}_{ij} = 1$ if there is a link between i and j and $\boldsymbol{A}_{ij} = 0$ otherwise. Ordering the rows and columns of the adjacency matrix according to the group assignment of nodes allows us to represent $\boldsymbol{A}$ as a set of blocks with link densities given by the affinity matrix $\boldsymbol{\Omega}$. Note that in the above description we have allowed self-loops and we will only consider undirected graphs, for simplicity. Some comments on extensions to directed graphs are provided in the discussion section. Based on such an SBM, one way to generate a hierarchy of communities is by recursively subdividing a block into more blocks and describe it as a type of a hierarchical random graph (HRG) model [2] (or its generalized variant [47]). Figure 2 illustrates such a hierarchy of communities. Starting at the root of the dendrogram in the left of the figure, we generate the total expected number of edges in the network m, the number of groups $k^{(1)}$, and a $k^{(1)} \times k^{(1)}$ expected edge count matrix $\boldsymbol{m}_{rs}^{(1)}$ that describes how the m links are distributed between the groups, i.e., $\sum_{rs}\boldsymbol{m}_{rs}^{(1)} = m$. Note that by convention $\boldsymbol{m}_{rr}^{(1)}$ is equal to twice the number of (undirected) edges in group $r^{(1)}$. The process continues by subdividing each of the groups in the same manner. For instance, we can subdivide the $n_r^{(1)}$ nodes in group $r^{(1)}$ by defining an edge count matrix $\boldsymbol{m}_{rs}^{(2)}$ that describes how the $\boldsymbol{m}_{rr}^{(1)}$ edges are distributed among the subgroups, i.e., $\sum_{rs}\boldsymbol{m}_{rs}^{(2)} = \boldsymbol{m}_{rr}^{(1)}$.

Multiple branches may occur simultaneously at the same *level* of the hierarchy, e.g., branches (a'), (b'), and (c') occur at the same level in the example in Fig. 2. We can represent each level u by an assignment of nodes to groups and an affinity matrix,

$$\boldsymbol{\Omega}_{rs}^{(u)} = \frac{\boldsymbol{m}_{rs}^{(u)}}{n_r^{(u)}n_s^{(u)}},$$

of connection probabilities that includes all groups in the network at level u. In other words, each level may be seen as an SBM that captures a particular resolution of the system. Each of the subgroups shares the stochastic equivalence inherited from the parent group, such that all child subgroups of the same parent share the same set of external connection probabilities to other groups. Specifically, the probability of a link between two nodes will be governed by the nearest common ancestor in the dendrogram.

Describing hierarchical communities in this way suggests that we should observe a particular pattern of edge densities in the adjacency matrix when the rows and columns are ordered appropriately. We observe such an example in Fig. 2, in which there is a hierarchical refinement of the block structure in the block diagonal of the adjacency matrix and a homogeneous density of edges in the off-diagonal. This notion of hierarchical group structure is one of the most common conceptualizations of hierarchical structure encountered in the literature [2,3,5]. We refer to this type of hierarchy as an *assortative hierarchy*.

These assortative hierarchical communities, however, may be limited in their representation of network connection patterns. For instance, Fig. 3(a) illustrates an assortative hierarchy, which allows us to capture disassortative structures only to some extent, i.e., the off-diagonal blocks can have a higher density than the diagonal blocks. But the assortative hierarchy may fail to capture the community structure when the distinction between resolutions is contained in the off-diagonal, e.g., Fig. 3(b) in which the diagonal blocks are homogeneous. A common example of networks of this type are bipartite networks in which the diagonal blocks contain no edges. A more general hierarchical structure may be constructed, as depicted in Fig. 3(c), by combining both assortative and disassortive hierarchical features.

A. Stochastic externally equitable partitions

To capture these types of generalized hierarchies, we define hierarchical communities by introducing the concept of stochastic externally equitable partitions, and describe their relationship to the stochastic block model. Figure 4 provides an overview of relevant concepts and equivalence relations and how they relate to each other.

For a given set of parameters, the SBM provides a parametric probability distribution over adjacency matrices. The expected adjacency matrix of this distribution can be calculated from the affinity matrix $\boldsymbol{\Omega}$ and group indicator matrix $\boldsymbol{H}$,

$$\mathbb{E}[\boldsymbol{A}] = \boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{H}^{\top}. \tag{2}$$

The expected adjacency matrix $\mathbb{E}[\boldsymbol{A}]$ induces a connected weighted graph, in which all nodes in the same group are associated with exactly the same pattern of weighted edges. In the network described by $\mathbb{E}[\boldsymbol{A}]$, nodes in the same group are therefore *structurally equivalent* [48] as they have the exact same set of neighbors and the same set of edge weights. In a network, represented by an adjacency matrix $\boldsymbol{A}$ generated from the SBM, nodes in the same group are *stochastically equivalent* as they connect to the rest of the nodes in the network according to the same set of probabilities (which are

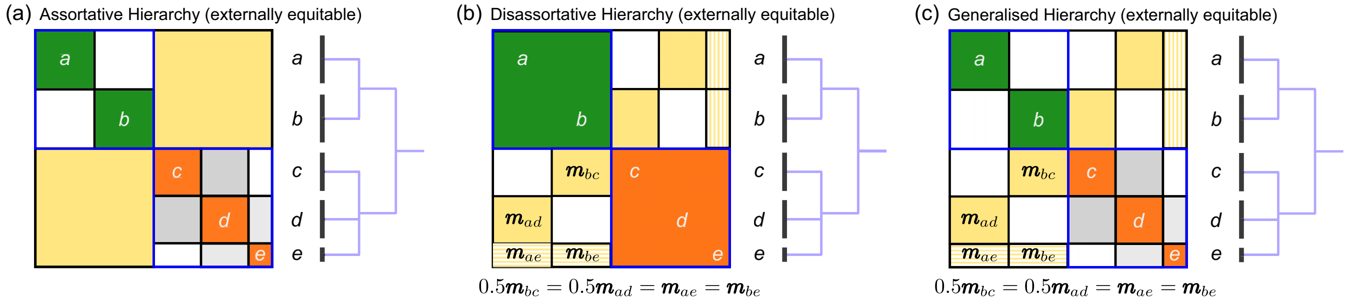
(a) Assortative Hierarchy (externally equitable)
(b) Disassortative Hierarchy (externally equitable)
(c) Generalised Hierarchy (externally equitable)
a
b
c
d
e
m_{bc}
m_{ad}
m_{ae}
m_{be}
$0.5m_{bc} = 0.5m_{ad} = m_{ae} = m_{be}$
$0.5m_{bc} = 0.5m_{ad} = m_{ae} = m_{be}$

FIG. 3. The block structure of configurations of hierarchical communities. (a) A simple assortative hierarchy of communities in which the refinement of the block structure between levels of the hierarchy occurs along the block diagonal and the off-diagonal blocks have homogeneous density. This type of hierarchy is the most frequently considered in the literature. Although we refer to this structure as an assortative hierarchy, the communities may be disassortative if the off-diagonal blocks are higher density than the diagonal blocks. (b) A disassortative hierarchy of communities in which the refinement of the block structure between levels of the hierarchy occurs in the off-diagonal blocks. Note that for the disassortative hierarchy to be an externally equitable partition, it must satisfy the stricter constraint that the sum of densities of the refined off-diagonal blocks should be equal along the rows and along the columns, i.e., $\boldsymbol{m}_{ad} + \boldsymbol{m}_{ae} = \boldsymbol{m}_{bc} + \boldsymbol{m}_{be}$ and $\boldsymbol{m}_{ad} = \boldsymbol{m}_{bc} = \boldsymbol{m}_{ae} + \boldsymbol{m}_{be}$. (c) A generalized hierarchy in which the refinement occurs in both the diagonal and off-diagonal blocks.

precisely $\mathbb{E}[\boldsymbol{A}]$). Put differently, groups of nodes in a network that share the exact same set of connections are structurally equivalent. When groups of nodes share the exact same set of connections *in expectation* then they are stochastically equivalent. In this way, we can consider stochastic equivalence as a probabilistic relaxation of structural equivalence [Fig. 4(a) top row].

When we partition an adjacency matrix $\boldsymbol{A}$ such that every node in a group r has simply the same number of links to nodes in group s, then we call such a partition of a graph an *equitable partition* [49]. Equitable partitions are a generalization of structural equivalence in which each node in the same group has the same sum of weights connecting it to every other group. (Note that here we will use the convention that the number of links, or *degree*, of a node refers to the sum of edge weights when the graph is weighted.) However, it is not necessary that nodes in the same group have exactly the same connections. Equitable partitions are closely related, but not identical to, graph automorphism groups [49,50], and regular equivalence [51,52]. Regular equivalence, for instance, does not require equivalent nodes to have the *same number of links* to equivalent nodes, whereas equitable partitions do have this requirement.

We can extend the concept of equitable partitions to random graph models by introducing a probabilistic relaxation, which we will call a *stochastic equitable partition* [Fig. 4(a) middle row]. Partitioning the expected adjacency matrix $\mathbb{E}[\boldsymbol{A}]$ according to $\boldsymbol{H}$ creates a stochastic equitable partition such that every node in group r has the same expected number of links to nodes in group s.

We can define equitable partitions algebraically using an aggregated graph with adjacency matrix $\boldsymbol{A}^g \in \mathbb{R}^{k\times k}$ in which each node represents a group and the weighted links indicate the sum of link weights between groups in a graph $\boldsymbol{A}$:

$$\boldsymbol{A}^g = \boldsymbol{H}^\top \boldsymbol{A}\boldsymbol{H}. \tag{3}$$

However, since groups may be of different sizes it is often more practical to use the quotient graph with weighted adjacency matrix $\boldsymbol{A}^\pi$ in which the aggregated graph $\boldsymbol{A}^g$ is normalized by the size of the groups:

$$\boldsymbol{A}^\pi = \boldsymbol{N}^{-1}\boldsymbol{H}^\top \boldsymbol{A}\boldsymbol{H} = \boldsymbol{H}^\dagger \boldsymbol{A}\boldsymbol{H}, \tag{4}$$

where $\boldsymbol{N} := \boldsymbol{H}^\top\boldsymbol{H}$ is a diagonal matrix in which N_{rr} is the number of nodes in group r and $\boldsymbol{H}^\dagger := \boldsymbol{N}^{-1}\boldsymbol{H}^\top$ is the Moore-Penrose pseudoinverse of $\boldsymbol{H}$. Then each element of the adjacency matrix of the quotient graph $\boldsymbol{A}^\pi_{rs}$ tells us the mean number of edges connecting a node in group r to nodes in group s. When $\boldsymbol{H}$ represents an equitable partition of $\boldsymbol{A}$ the value $\boldsymbol{A}^\pi_{rs}$ is the actual number of links that every node in group r has with nodes of group s, i.e., we have the following algebraic relation:

$$\boldsymbol{A}\boldsymbol{H} = \boldsymbol{H}\boldsymbol{A}^\pi \quad \text{for all} \quad \boldsymbol{H} \in \mathcal{H}^A_{\text{EP}}, \tag{5}$$

where $\mathcal{H}^A_{\text{EP}}$ is the set of equitable partitions of $\boldsymbol{A}$.

When we consider partitions that are equitable only between different groups $r \neq s$, then the partition is called an *externally equitable partition* (EEP). We can characterize EEPs algebraically by following Eqs. (3)–(5) and substituting the combinatorial graph Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ in place of the adjacency matrix [53], where $\boldsymbol{D} = \text{diag}(\boldsymbol{A}\mathbf{1})$ is a diagonal matrix of degrees. This substitution gives

$$\boldsymbol{L}\boldsymbol{H} = \boldsymbol{H}\boldsymbol{L}^\pi, \quad \boldsymbol{H} \in \mathcal{H}^A_{\text{EEP}}, \tag{6}$$

where $\mathcal{H}^A_{\text{EEP}}$ is the set of external equitable partitions of $\boldsymbol{A}$, $\boldsymbol{L}^\pi$ is the Laplacian of the quotient graph,

$$\begin{aligned} \boldsymbol{L}^\pi &= \boldsymbol{N}^{-1}\boldsymbol{H}^\top(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{H} && (7)\\ &= \boldsymbol{D}^\pi - \boldsymbol{A}^\pi, && (8) \end{aligned}$$

and $\boldsymbol{D}^\pi = \text{diag}(\boldsymbol{A}^\pi\mathbf{1})$ is the diagonal matrix of node degrees by group. Substituting the Laplacian for the adjacency matrix enables us to ignore the internal connectivity and only constrain the external connections to be equitable. The reason that the quotient Laplacian ignores the internal connectivity is its invariance under the addition of edges in the diagonal blocks of the adjacency matrix $\boldsymbol{A}$, as the following proposition illustrates.

Proposition 1. Let $\boldsymbol{H}$ be the indicator matrix of an EEP and $\boldsymbol{A}' = \boldsymbol{A} + \Delta\boldsymbol{A}$ be an adjacency matrix with additional

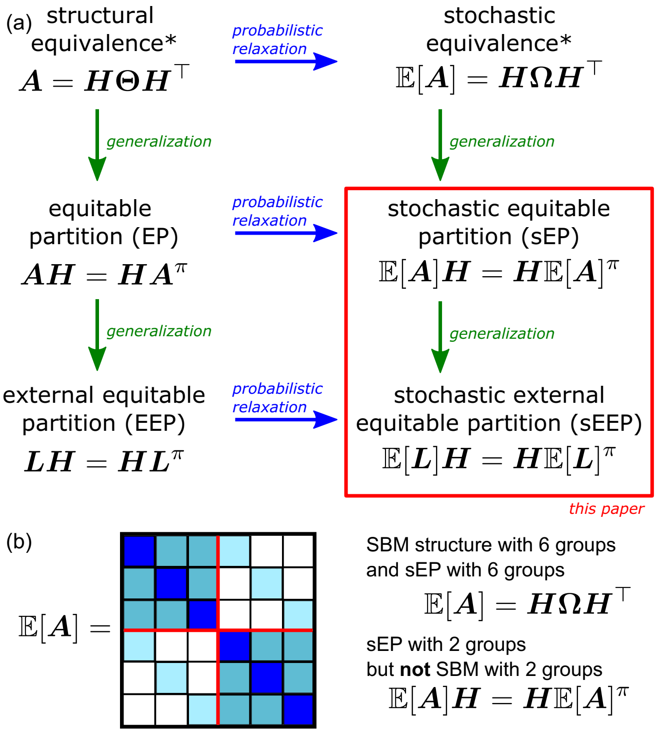
(a)
structural equivalence*
$A = H\Theta H^\top$
probabilistic relaxation
stochastic equivalence*
$\mathbb{E}[A] = H\Omega H^\top$
generalization
generalization
equitable partition (EP)
$AH = HA^\pi$
probabilistic relaxation
stochastic equitable partition (sEP)
$\mathbb{E}[A]H = H\mathbb{E}[A]^\pi$
generalization
generalization
external equitable partition (EEP)
$LH = HL^\pi$
probabilistic relaxation
stochastic external equitable partition (sEEP)
$\mathbb{E}[L]H = H\mathbb{E}[L]^\pi$
this paper
(b)
$\mathbb{E}[A] =$
SBM structure with 6 groups and sEP with 6 groups
$\mathbb{E}[A] = H\Omega H^\top$
sEP with 2 groups but **not** SBM with 2 groups
$\mathbb{E}[A]H = H\mathbb{E}[A]^\pi$

FIG. 4. Overview of network partition equivalence relationships. (a) The left column describes partitions (represented by a partition indicator matrix H) into groups of equivalent nodes in a given graph (represented by an adjacency matrix A). The right column presents the corresponding probabilistic relaxation in which the equivalence relation is considered in terms of the expected adjacency matrix $\mathbb{E}[A]$ over the ensemble of networks generated by a random graph model (* Note that for simplicity we allow for graphs with self-loops in the algebraic expressions of structural and stochastic equivalence). *Structural equivalence:* nodes are equivalent if they link to the same neighbors. Here Θ is a $\{0, 1\}$ matrix. *Stochastic equivalence:* nodes are structurally equivalent in expectation. *Equitable partition:* nodes are equivalent if they have the same number of links to equivalent nodes. *Stochastic equitable partition:* the partition is an EP *in expectation*. *Externally equitable partition:* nodes are equivalent if they have the same number of links to equivalent nodes, outside their own group. *Stochastic externally equitable partition:* the partition is an EEP *in expectation*. (b) Example of a network model with a partition into 6 and 2 groups. The partition into 6 groups is consistent with an SBM and an sEP. The partition into 2 groups is consistent with an sEP (and an sEEP) but not an SBM because the link probabilities are not uniform within the blocks.

within-group edges, i.e., edges that occur within the diagonal blocks. Then $L^\pi(A') = L^\pi(A)$.

Proof.

$$\begin{aligned} L^\pi(A') &= N^{-1}H^\top(D' - A')H \\ &= N^{-1}H^\top(D - A + \Delta D - \Delta A)H \\ &= D^\pi - A^\pi + \mathbf{0} = L^\pi(A), \end{aligned} \tag{9}$$

where ΔD is the diagonal matrix diag$(\Delta A\mathbf{1})$. The final equality in Eq. (9) is due to the fact that ΔA only contains edges in the diagonal blocks and so the diagonal matrix $H^\top \Delta A' H = H^\top \Delta D' H$ is equal to the group sum of degrees. ■

As for the EP, we propose a probabilistic relaxation for an EEP: a *stochastic externally equitable partition* (sEEP) is a partition that is externally equitable *in expectation* [Fig. 4(a) bottom row]. A stochastic EEP is precisely the type of relationship we find at each level of a simple assortative hierarchy. For instance, the internal structure within the block diagonal of an assortative hierarchy may be further refined, but the probability of connections within the off-diagonal blocks should be uniform. This construction can be precisely captured by an sEEP. As a concrete example, in Fig. 3(a), both the partition $\{a, b, c, d, e\}$ and the partition $\{\{a, b\}, \{c, d, e\}\}$ of the expected adjacency matrix are externally equitable. However, stochastic EEPs also enable us to describe the more general forms of hierarchical structure shown in Figs. 3(b) and 3(c). Specifically, in an sEEP the links between nodes inside a block do not need to be uniformly distributed but merely the expected degree with respect to every external block has to be the same. Together with the fact that the distribution of the parameters inside the diagonal blocks in an sEEP is flexible this constitutes the main difference from the SBM [see Fig. 4(b)]. Specifically, in the canonical SBM all elements within a block of $\mathbb{E}[A]$ have equal weight, in an sEEP all rows and all columns within a block of $\mathbb{E}[A]$ sum to the same value, whereas in the microcanonical SBM [54] it is the number of edges (or sum of weights) in a block of A that is fixed. This difference allows for a more flexible modeling of hierarchies than the canonical SBM, while maintaining a conceptually well defined setup.

We therefore use the concept of a *stochastic externally equitable partition* (sEEP) as the basic building block for hierarchical modular structure in networks. Specifically, we say the communities of a graph are hierarchically organized, if the graph's adjacency matrix can be partitioned into a sequence of nested stochastic externally equitable partitions. More precisely, there is sufficient evidence for a hierarchical partition if at each level of the putative hierarchy the partition is a stochastic externally equitable partition (an EEP in expectation).

If we want to recover such a hierarchy, our goal is therefore to obtain the partitions at each of the hierarchical levels, including the number of levels and number of groups at each level. However, before we discuss any specific method of inference, it is important to discuss some conceptual issues we face when inferring hierarchical structure from a network. In particular, we need not only determine when a hierarchy exists, but also how many levels are contained within the hierarchy and in which order those levels occur. As we will see in the next section, it is in general not possible to identify these aspects uniquely, even if we have access to the *expected* adjacency matrix.

III. IDENTIFIABILITY OF HIERARCHICAL CONFIGURATIONS

Our discussion above provides us with the necessary condition for defining a set of hierarchical partitions, i.e., that they form a nested sEEP structure. However, this condition alone is insufficient to fully define a set of hierarchical communities, as we still need to resolve issues of identifiability, which we will discuss here in this section.

Identifiability is a necessary condition to guarantee that we can recover the model parameters and the hierarchy given sufficient data. Models of community detection (and clustering, more generally) suffer from a certain degree of nonidentifiability because the community labels are permutation invariant. This means that there are $k!$ ways to label the same k groups. However, this nonidentifiability does not pose any problems in practice as our interpretation of each of these $k!$ solutions is identical. When we detect hierarchical communities, we face similar issues of identifiability. At any given level u of the hierarchy, the labels of the $k^{(u)}$ groups are permutation invariant and, as with community detection, all possible labellings of these groups represent an identical solution. On top of this, we can represent a hierarchy as multiple distinct dendrograms by changing how we assign branches to hierarchical levels, the order of agglomeration and/or the number of levels.

A. Assigning branches to levels

Let us assume that we already know the dendrogram structure, i.e., the rooted tree of splits of the nodes into groups, and the assignment of nodes to the groups at each branch. All that remains is to determine how to assign each of the branches to levels in the hierarchy. Figure 5 shows some examples of different ways to assign branches of a dendrogram to levels in a hierarchy. Figures 5(a)–5(c) shows three different assignments for the same dendrogram, one assignment into three levels and two assignments into four levels. In each case, the main left and right branches are independent of each other and do not provide information about how we should arrange their subbranches relative to each other. All three provide the same information about the hierarchical group assignment. When confronted with equivalent solutions, a natural strategy is to take an Occam's razor approach and choose the simplest or most compact solution. In this case, we might therefore decide that the configuration displayed in Fig. 5(a) is the best choice since it has only two levels.

In other situations the "simplest" assignment of branches to levels may be more ambiguous. Figures 5(d) and 5(e) show a dendrogram for which we can align the split in the left branch with either the second level (i) or the third level (ii) of the right branch. Both representations contain the same information about how the nodes are partitioned. However, the choice between (d) and (e) provides different aggregated affinity matrices $\mathbf{\Omega}^{(2)}$ [see Figs. 5(d) and 5(e)] that describe how the groups of the system interact.

It may be that in these situations a specific choice of model selection may prefer one configuration over another. However, as we recover the same set of groups the solutions are equivalent and therefore we should treat both solutions as the same, just as we treat partitions with different permutations of labels as being the same. Stated differently, the tree structure of the dendrogram remains the same, even though we interpret its branching points differently.

B. Order of agglomeration

Now consider the setting in which, instead of knowing the dendrogram, we know the desired number of layers ℓ in the hierarchy. We will also assume that we are given the partition at the finest resolution. All that remains is to decide which communities we should aggregate and in which order, i.e., we want to identify the dendrogram that describes the hierarchy of communities. Figure 6 displays two example configurations for which this question is a priori ambiguous.

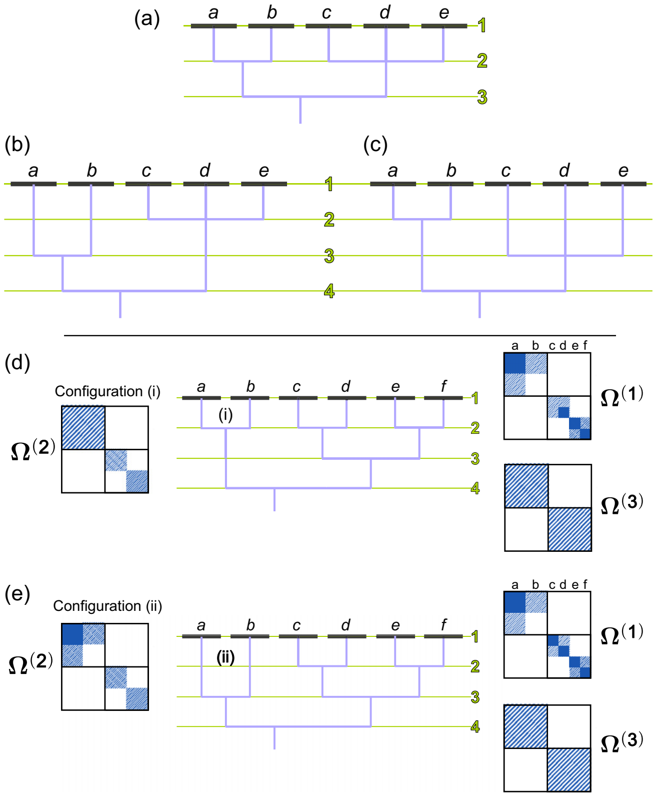

FIG. 5. Assigning dendrogram branches to levels. [(a)–(c)] Three possible assignments of hierarchical levels based on the same dendrogram. Note that (a) provides a simpler description (in terms of number of levels) and may thus be preferable. [(d) and (e)] Two possible hierarchical configuration for the same network. As the assignment of the dendrogram to levels has the same complexity, unless additional information is provided we cannot decide on a specific hierarchy based on the network alone. Note that the matrices $\mathbf{\Omega}^{(u)}$ correspond to level u of the hierarchy, with groups a to f as indicated for $\mathbf{\Omega}^{(1)}$.

Figure 6(a) shows an example in which the parameter matrix $\mathbf{\Omega}^{(1)}$ of the finest level is the same for both configurations (i) and (ii), where one is just a simple permutation of the other. However, the affinity matrices $\mathbf{\Omega}^{(2)}$ at the coarser level are different and so the decision of which configuration to pick depends on which version of $\mathbf{\Omega}^{(2)}$ we prefer. An appropriate form of model selection may prefer one configuration over another. For instance, configuration (i) has more zero blocks than configuration (ii) and so will have a higher likelihood if we assume the network was created from a nested SBM [4].

The situation is different in Fig. 6(b). Even though we have different affinity matrices $\mathbf{\Omega}^{(2)}$, the difference simply amounts to a different permutation of the same values and so the hierarchical configuration is nonidentifiable, unless we once again include additional criteria, e.g., instead of maximising zero blocks, we might include a preference for assortative communities [55,56]. Stated differently, in both cases, had we

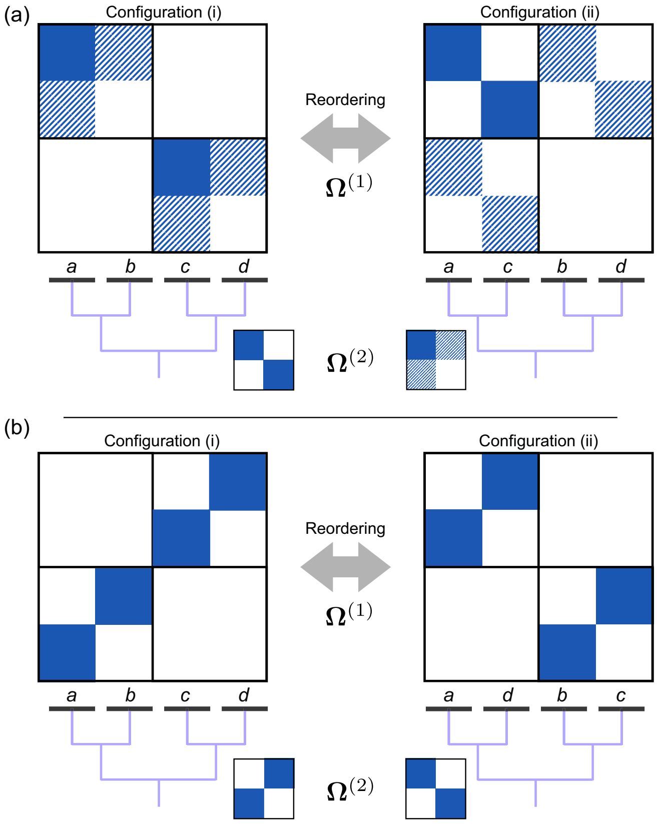

FIG. 6. Deciding the order of hierarchical agglomeration. We display the pattern of two possible orderings of an affinity matrix $\mathbf{\Omega}^{(1)}$ (and its possible aggregated version $\mathbf{\Omega}^{(2)}$), indicating two possible ways of hierarchical aggregation. Blue represents an arbitrary link probability, white color represents zero link probability. (a) A network with two possible perfect hierarchical configurations: the ordering displayed as configuration (i) may be described as two communities with inherent core-periphery structure, the ordering displayed as configuration (ii), might be thought of as a core-periphery organization with an inherent assortative modular structure. (b) A network with two possible hierarchical configurations, interpretable as a bipartite structure of bipartite structures; or an assortative partition of bipartite structures.

planted one or the other hierarchy in a synthetic network, determining which hierarchy was planted would be impossible to infer from the observed data.

C. Number of levels

Finally, let us consider the setting in which all we know is the finest partition of the network and we need to decide how to aggregate groups and how many levels there should be in the hierarchy. Similar to the task of assigning branches to levels, it may be desirable to identify the simplest hierarchy. However, in this case, we do not know the branches and must decide if adding levels to the hierarchy will be meaningful. As previously demonstrated in Fig. 1, aggregating groups into any hierarchy with a particular number of levels $\ell > 1$ does not imply evidence of a unique hierarchical arrangement of communities (as defined previously) in the network. At the very least we would like to avoid including vacuous levels in the hierarchy, as in the case of Fig. 1. A clear signal of a vacuous level is a degeneracy with respect to which groups we choose to agglomerate.

As a concrete example, consider a flat partition generated from a planted partition model with k groups. In a planted partition model the affinity matrix $\mathbf{\Omega} = (a - b)\boldsymbol{I} + b\mathbf{1}\mathbf{1}^\top$ can be described with only two parameters: a, the probability that a pair of nodes in the same group will connect, and b, the probability that a pair of nodes in different groups will connect. Note that the example given in Fig. 1 is a special case of the planted partition model with $k = 64$, $a = 1$, and $b = 0$. The partition of $\mathbb{E}[\boldsymbol{A}]$ into k groups will be an EEP. If we form a new partition into κ groups (where $\kappa < k$) by simply merging some of the k groups, then the new partition will also be an EEP. In fact *any* partition formed by merging these groups will create an EEP and so every partition into κ groups will be equivalent to each other. This degeneracy of partitions therefore indicates the absence of a meaningful level in the hierarchy.

D. Dealing with nonidentifiability and degenerate hierarchies

As our above discussion shows, even if we had perfect knowledge about the expected adjacency matrix $\mathbb{E}[\boldsymbol{A}]$, uniquely identifying an underlying hierarchy is in general impossible without imposing further assumptions. In other words, we need to impose some rules on how to break the nonidentifiability issues encountered above. In the following, we will develop a set of tools based on spectral properties associated to sEEPs, both in terms of eigenvectors as well as eigenvalues, which we will employ to detect hierarchical block structures in networks. We emphasize that the discussion above applies generally and is not tied to any of these developments. Specifically, our spectral approach is not the only way to resolve these issues of nonidentifiability and other methods using different assumptions are conceivable as well, e.g., the already mentioned nested blockmodel by Peixoto [4].

IV. DETECTING HIERARCHIES VIA SPECTRAL METHODS

Thus far we have conceptualized hierarchical modular structure in terms of sequences of sEEPs based on the expected adjacency $\mathbb{E}[\boldsymbol{A}]$ that relates to the affinity matrix of the finest partition $\mathbf{\Omega}^{(1)}$. When we want to perform community detection in practice, however, we typically only have access to an observed sample adjacency matrix $\boldsymbol{A}$. Therefore we have to either infer the precise affinity matrix $\mathbf{\Omega}^{(1)}$, which is only possible in the thermodynamic limit under certain conditions [20], or we have to define conditions for concluding that an sEEP is present based on the observed adjacency matrix $\boldsymbol{A}$. In combination with the identifiability issues described in the previous section the problem of detecting hierarchical communities is thus, in general, an ill-posed problem.

In the following we will employ spectral methods to infer hierarchical community structure in a network, which correspond to a particular way of resolving the above nonidentifiability issues. Before we address these issues directly, we first outline our overall strategy to detect hierarchical community structure.

A. *Identify the initial finest-grained network partition.* We first identify the finest level of the hierarchy (i.e., the level furthest from the root of the dendrogram) such that all nodes within a group are stochastically equivalent and ignoring the trivial partition into n groups that each contain a single node. Using this initial partition we can estimate the affinity matrix of the finest partition:

$$\widehat{\boldsymbol{\Omega}}^{(u)} = \boldsymbol{H}^{\dagger(u)}\boldsymbol{A}(\boldsymbol{H}^{\dagger(u)})^{\top}, \tag{10}$$

where $u = 1$ (Sec. IV A).

B. *Identify possible agglomerations and hierarchical levels.* Treating the estimated affinity matrix $\widehat{\boldsymbol{\Omega}}^{(u)}$ as a weighted adjacency matrix, we then identify candidate partitions to form the next level in the hierarchy $\boldsymbol{H}^{(u+1)}$ by merging groups in the current partition such that they form an approximate sEEP (Sec. IV B).

C. *Agglomerate and repeat.* Based on the identified partitions we select the most suitable agglomeration and estimate an affinity matrix at the next level:

$$\widehat{\boldsymbol{\Omega}}^{(u+1)} = \boldsymbol{H}^{\dagger(u+1)}\boldsymbol{N}^{(u)}\widehat{\boldsymbol{\Omega}}^{(u)}\boldsymbol{N}^{(u)}(\boldsymbol{H}^{\dagger(u+1)})^{\top}.$$

Note that $\boldsymbol{H}^{(u+1)}$ maps the nodes in the aggregated graph at level u in the hierarchy to the nodes in level $(u+1)$ and so the dimensions of $\boldsymbol{H}^{(u+1)}$ will be $n^{(u)} \times n^{(u+1)}$, where the number of nodes at a given level are $n^{(u+1)} = k^{(u)}$ the number of groups at the previous level. We then return to the previous step and repeat until no further agglomerations are found (Sec. IV C).

The key elements for addressing the nonidentifiability issues are contained in steps B and C. First, we consider an order of agglomeration (cf. Sec. III B) that is induced by a singular value (or *spectral*) decomposition associated with the estimated affinity matrices. This step may be interpreted as trying to find an agglomeration into κ groups (where $\kappa < k^{(u)}$) that are compatible with the best rank-κ approximation of the affinity matrix. Second, we assess the significance of any putative agglomeration via spectral criteria to avoid inserting "vacuous" levels into the hierarchy (cf. Sec. III C). This step makes use of certain degeneracies that may exist in the spectrum, which we will discuss. In the next sections we explain each of the above outlined steps in detail.

A. Establishing an initial partition

At this stage, one may wonder why the identification of the initial partition is different from identifying partitions at subsequent levels in the hierarchy. Typically the networks we observe are sparse, meaning that the number of edges tends to scale linearly with the number of nodes $O(n)$, rather than scale according to the number of possible edges $O(n^2)$. In contrast, when detecting subsequent partitions we will use a (weighted) denser aggregated graph, in which nodes represent groups in the partition of the previous level. Different methods are better suited to sparse or dense graphs. In particular, sparsity is known to cause issues for detecting communities, particularly when employing spectral methods [37,57].

For detecting the initial partition we will perform spectral clustering using the Bethe Hessian, which can detect communities in sparse networks right down to the theoretical limit of detectability [38]. Furthermore, the Bethe Hessian comes equipped with a simple spectral model selection criterion that enables us to infer the number of groups [38,58]. Our experimental results confirm these theoretical studies and empirically we find that spectral clustering with the Bethe Hessian reliably identifies the finest detectable partition.

Given the adjacency matrix $\boldsymbol{A}$ of a graph and the associated degree matrix $\boldsymbol{D} = \text{diag}(\boldsymbol{A}\boldsymbol{1})$, the Bethe Hessian [38] is defined as follows:

$$\boldsymbol{B}_{\eta} = (\eta^2 - 1)\boldsymbol{I} + \boldsymbol{D} - \eta\boldsymbol{A}, \tag{11}$$

where η is a regularization parameter, which allow us to modify the spectral properties of $\boldsymbol{B}_{\eta}$ so that we can use it to detect community structure even for sparse graphs and graphs with heterogeneous degree distributions [59]. Notice that when $\eta = 1$, we recover the combinatorial graph Laplacian $\boldsymbol{B}_1 = \boldsymbol{L}$.

Setting the regularization parameter η to a positive value favors the discovery of assortative communities, whereas a negative value favors disassortative communities. As we are interested in both forms of community structure, we set the regularization parameter to the positive and negative square root of the average degree $\eta = \pm\sqrt{\boldsymbol{1}^{\top}\boldsymbol{A}\boldsymbol{1}/n}$ [38,58]. For these settings, the number of negative eigenvalues provide a consistent estimate of the number of groups according to the SBM (see Theorem 4.3. in Ref. [58]). Therefore we can use the spectral clustering with the Bethe Hessian to infer both the number of groups and the node assignments to groups at the finest hierarchical level.

We describe the exact algorithm to establish an initial partition using the Bethe Hessian in Algorithm 1 in Appendix E.

B. Identifying candidate levels in the hierarchy

Having found an initial partition $\boldsymbol{H}^{(1)}$, we can estimate the affinity matrix $\widehat{\boldsymbol{\Omega}}^{(1)}$ at the finest level of the hierarchy. Treating $\widehat{\boldsymbol{\Omega}}^{(1)}$ as a weighted adjacency matrix $\boldsymbol{A}^{(2)}$ for the second level (i.e., $\boldsymbol{A}^{(2)} = \widehat{\boldsymbol{\Omega}}^{(1)}$), our task is now to evaluate whether or not there is sufficient evidence for a hierarchy of communities in the network.

Like other graph partitioning problems, finding all EEPs within a graph can be a computationally demanding task due to its combinatorial nature. If we had access to the exact affinity matrix, we could adopt tools from computational group theory, which have recently shown great promise in the related problem of identifying orbit partitions within graphs [60–62]. However, these tools are not suitable for our task as they are only able to identify exact EEPs of the adjacency matrix, whereas we need to identify *stochastic* EEPs, which are exact EEPs but only of the unobserved expected adjacency matrix $\mathbb{E}[\boldsymbol{A}^{(2)}] = \boldsymbol{\Omega}^{(1)}$. In the best case, when a network is generated from a hierarchical model using an affinity matrix $\boldsymbol{\Omega}^{(1)}$ that contains a nested set of exact EEPs, our estimate $\widehat{\boldsymbol{\Omega}}^{(1)} \to \boldsymbol{\Omega}^{(1)}$ only converges asymptotically. Even if we knew the true finest partition $\boldsymbol{H}^{(1)}$ of the generating model, statistical variation will result in minor perturbations in the estimated affinity matrix $\widehat{\boldsymbol{\Omega}}^{(1)}$ relative to the true $\boldsymbol{\Omega}^{(1)}$. We therefore require a new approach that enables us to define and identify stochastic EEPs within $\widehat{\boldsymbol{\Omega}}^{(1)}$. To do so, we introduce the notion of an *approximate* EEP. Noting that $\widehat{\mathbf{\Omega}} \approx \mathbf{\Omega}$, a partition that is an exact EEP of $\mathbf{\Omega}$ will be approximately an EEP of $\widehat{\mathbf{\Omega}}$. We now turn our attention to detecting approximate EEPs as a proxy for sEEPs.

1. Finding approximate EEPs

Central to our pursuit of identifying approximate EEPs is the fact that the presence of an (exact) EEP induces a particular structure on the eigenspaces of the Laplacian [53,63].

Proposition 2. Let $\boldsymbol{L}$ be the graph Laplacian of a weighted, undirected graph with an EEP consisting of k groups, described by the indicator matrix $\boldsymbol{H}$. Then, there exist k eigenvectors $\boldsymbol{V}_k = [\boldsymbol{v}_{\cdot 1}, \dots, \boldsymbol{v}_{\cdot k}]$ and corresponding eigenvalues $[\lambda_1, \dots, \lambda_k]$, where $\boldsymbol{L}\boldsymbol{v}_{\cdot i} = \lambda_i \boldsymbol{v}_{\cdot i}$, such that the values of $\boldsymbol{v}_{\cdot i}$ are piecewise constant for nodes within each group.

Proof. If $\boldsymbol{H}$ represents an EEP and the corresponding quotient Laplacian $\boldsymbol{L}^\pi$ has a matrix of eigenvectors $\boldsymbol{V}_k^\pi = [\boldsymbol{v}_{\cdot 1}^\pi, \dots, \boldsymbol{v}_{\cdot k}^\pi]$, then

$$\boldsymbol{L}\boldsymbol{V}_k = \boldsymbol{L}\boldsymbol{H}\boldsymbol{V}_k^\pi = \boldsymbol{H}\boldsymbol{L}^\pi\boldsymbol{V}_k^\pi = \boldsymbol{H}\boldsymbol{V}_k^\pi\boldsymbol{\Lambda}^\pi = \boldsymbol{V}_k\boldsymbol{\Lambda}^\pi, \tag{12}$$

where $\boldsymbol{\Lambda}^\pi$ is the diagonal matrix of eigenvalues of $\boldsymbol{L}^\pi$. ■

The above proposition tells us that when a network contains an EEP, then there exists a set of eigenvectors $\boldsymbol{V}_k$ that can be written as a linear combination of the group indicator matrix $\boldsymbol{H}$, i.e., there exists a matrix $\boldsymbol{Q} \in \mathbb{R}^{k\times k}$ such that $\boldsymbol{V}_k = \boldsymbol{H}\boldsymbol{Q}$. Thus $\boldsymbol{V}_k = \boldsymbol{H}\boldsymbol{V}_k^\pi$ is a valid set of eigenvectors of the Laplacian $\boldsymbol{L}$ that are constant for nodes within the same group. We will refer to these eigenvectors that contain this special piecewise structure as *structural eigenvectors*.

For an exact EEP the variation of any structural eigenvector $\boldsymbol{v}$ within each group is zero. It follows then that we can characterize an approximate EEP according to the error of approximating the eigenvectors as piecewise constant. To calculate this error, we use the matrix $\boldsymbol{H}\boldsymbol{H}^\dagger$, in which $[\boldsymbol{H}\boldsymbol{H}^\dagger]_{ij} = 1/n_r$ if nodes i and j belong to the same group r and 0 otherwise, to define a projection orthogonal to the partition $\boldsymbol{H}$

$$\boldsymbol{P}_{\boldsymbol{H}} := [\boldsymbol{I} - \boldsymbol{H}\boldsymbol{H}^\dagger], \tag{13}$$

in which $\boldsymbol{H}\boldsymbol{H}^\dagger$ is used to calculate a groupwise mean such that the operator $\boldsymbol{P}_{\boldsymbol{H}}$ computes the matrix of residuals. Then we can calculate the *squared projection error* using the Frobenius norm $||\cdot||_{\mathrm{F}}$:

$$\varepsilon(\boldsymbol{H}, \boldsymbol{V}_k) := \|\boldsymbol{P}_{\boldsymbol{H}}\boldsymbol{V}_k\|_{\mathrm{F}}^2. \tag{14}$$

In Sec. A, we provide evidence that minimizing this projection error is consistent with finding approximate EEPs. Consequently we can search for an approximate EEP $\widehat{\boldsymbol{H}}$ by minimizing the projection error:

$$\widehat{\boldsymbol{H}} = \arg\min_{\boldsymbol{H}\in\mathcal{H}_k} \|\boldsymbol{P}_{\boldsymbol{H}}\boldsymbol{V}_k\|_{\mathrm{F}}^2, \tag{15}$$

where $\mathcal{H}_k$ is the set of all partition indicators matrices with k nonempty groups.

Geometrically, the above optimization problem amounts to finding k group-indicator vectors in an n-dimensional space, such that the k vectors $\boldsymbol{V}_k$ will have the smallest possible variation within each group (i.e., they will be approximately constant in each group). Interestingly, rather than having to devise a new optimization algorithm for the above problem, we can solve the above problem using k-means to cluster the rows of the matrix $\boldsymbol{V}_k$. We provide this proof in Sec. B.

Since there exist well developed algorithms to solve the k-means problem this duality enables us to efficiently search for a candidate EEP when given a set of putative structural eigenvectors. In particular, there exist algorithms that can provide us with a provable $(1+\delta)$ approximation of the true solution of the k-means problem [64].

Connections between spectral clustering of graphs and k-means have previously been reported in the literature (see, e.g., Ref. [65]), but only in relation to simple *assortative* clusters. The duality we present here shows that the k-means procedure, when applied to the relevant eigenvectors of the Laplacian, is also related to the identification of more general EEP structures, both assortative and disassortative.

2. Selecting relevant eigenvectors

We have established that if a network contains an approximate EEP then we can use k-means with a relevant set of k eigenvectors $\boldsymbol{V}_k$ to identify the partition. In principle, we could search all possible combinations of k eigenvectors to determine the relevant set, but this approach becomes increasingly inefficient as the network size increases.

The usual approach to selecting relevant eigenvectors for spectral clustering is to choose the eigenvectors associated with the first k eigenvalues [66], where "first" refers to either the smallest or largest values (either in terms of the real or absolute value) depending on the specific operator used. If we take this approach using the combinatorial Laplacian $\boldsymbol{L}$ then we would be constrained to identify either only assortative groups (if we use the lowest) or only disassortative groups (if we use the highest). In order to detect both assortative and disassortative groups at the same time, we propose to use the eigenvectors associated to eigenvalues with the k largest absolute values of the *uniform random walk transition matrix* $\boldsymbol{W}$:

$$\boldsymbol{W} = \boldsymbol{I} - \frac{1}{d_{\max}}\boldsymbol{L}, \tag{16}$$

where $d_{\max} = \max_i(D_{ii})$ is the maximal weighted degree of any node in the graph. Notice that $\boldsymbol{W}$ is simply a shifted and scaled version of the Laplacian, which has previously been considered in the analysis of consensus dynamics and distributed averaging [67].

The matrix $\boldsymbol{W}$ is a doubly stochastic matrix that describes a diffusion process on the network. Specifically, a diffusion process that has a uniform stationary distribution such that all nodes are visited with equal probability. Importantly, $\boldsymbol{W}$ has the same eigenvectors as $\boldsymbol{L}$ and so the aforementioned desirable spectral properties of $\boldsymbol{L}$ also apply to $\boldsymbol{W}$. The difference is that the set of eigenvalues $\{\lambda_i\}$ of $\boldsymbol{W}$ are normalized such that $\lambda_i \in [-1, 1]$. Eigenvectors associated with positive eigenvalues correspond to assortative partitions. The eigenvector associated with the largest possible positive eigenvalue $\lambda_i = 1$ is the vector of ones $\mathbf{1}$ and groups all nodes into a single group (assuming the network comprises a single connected component). Eigenvectors associated with negative eigenvalues correspond to disassortative partitions, where an eigenvector associated with eigenvalue $\lambda_i = -1$ will describe a bipartite split in a network with a uniform degree distribution, if such a partition is possible. Choosing the eigenvectors of $\boldsymbol{W}$ associated with the k eigenvalues with the largest magnitude therefore allows us to detect both assortative and disassortative groups.

Note that the choosing the top k eigenvectors according to absolute value may equivalently be interpreted in terms of choosing the top k singular values and associated singular vectors of the matrix $\boldsymbol{W}$, i.e., performing the best possible rank-k of $\boldsymbol{W}$. Rewriting Eq. (16) in terms of the affinity matrix $\boldsymbol{\Omega}$ and its degree matrix $\boldsymbol{D}_{\boldsymbol{\Omega}} = \mathrm{diag}(\boldsymbol{\Omega}\mathbf{1})$:

$$\boldsymbol{W}(\boldsymbol{\Omega}) = \boldsymbol{I} - \frac{1}{d_{\max}}\boldsymbol{D}_{\boldsymbol{\Omega}} + \frac{1}{d_{\max}}\boldsymbol{\Omega}, \tag{17}$$

we see that our choice of eigenvectors corresponds essentially to performing a rank-k approximation of the affinity matrix, i.e., we try to find partitions into k groups that best approximate the (rescaled and shifted) affinity matrix.

C. Assembling the hierarchy

We have described an approach to detect approximate EEPs with a prescribed number of groups within an estimated affinity matrix $\widehat{\boldsymbol{\Omega}}^{(i)}$. We now describe how we can use this approach to detect and construct a hierarchy of communities from a network. Specifically, in the following we discuss how to determine if a partition into k groups is significant enough to be included in the hierarchy and how we can identify degeneracies to avoid constructing misleading hierarchies.

1. Assessing the significance of approximate EEPs

Using the duality between k-means clustering and minimizing the projection error we can efficiently search for the partition closest to an EEP given a set of k eigenvectors $\boldsymbol{V}_k$. Optimizing Eq. (15) via k-means will however always provide a result, even if the inferred partition $\widehat{\boldsymbol{H}}$ is far from being an EEP. Therefore it is necessary to check if the resulting partition $\widehat{\boldsymbol{H}}$ is significantly close to being an EEP.

We test for significance by comparing the projection error $\varepsilon(\widehat{\boldsymbol{H}}, \boldsymbol{V}_k)$ against the expected projection error $\mathbb{E}[\varepsilon(\widehat{\boldsymbol{H}}, \boldsymbol{U})]$ under the null hypothesis that the set of eigenvectors is sampled uniformly at random from the set of all orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{n\times k}$, i.e., those matrices $\boldsymbol{U}$ for which $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$.

To see how we can calculate this expectation, let us start by examining a random matrix $\boldsymbol{U} \in \mathbb{R}^{n\times k}$ of k orthonormal vectors of dimension n. The squared Frobenius norm $\|\boldsymbol{U}\|_{\mathrm{F}}^2 = \mathrm{trace}(\boldsymbol{U}^\top\boldsymbol{U})$ of such a matrix will be equal to k. We can compute the expectation of the square of each individual entry of $\boldsymbol{U}$ as

$$\begin{aligned} k = \mathbb{E}\big[\|\boldsymbol{U}\|_{\mathrm{F}}^2\big] &= \mathbb{E}\left[\sum_{j=1}^{k}\sum_{i=1}^{n} U_{ij}^2\right] = \sum_{j=1}^{k}\sum_{i=1}^{n}\mathbb{E}\big[U_{ij}^2\big] \\ &= kn\cdot\mathbb{E}\big[U_{ij}^2\big], \end{aligned} \tag{18}$$

where in the last step we used the fact all of the entries U_{ij} are statistically equivalent. We can conclude by symmetry that

$$\mathbb{E}\big[U_{ij}^2\big] = 1/n, \tag{19}$$

for all indices $i = 1, \dots, n$ and $j = 1, \dots, k$.

Now, let us consider the spectral decomposition of the projection matrix $\boldsymbol{P}_{\boldsymbol{H}}$ associated with a partition into k groups:

$$\boldsymbol{P}_{\boldsymbol{H}} = \boldsymbol{I} - \boldsymbol{H}\boldsymbol{H}^\dagger = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top, \tag{20}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a diagonal matrix with $\Lambda_{ii} = 1$ for $i = 1, \dots, n-k$ and $\Lambda_{ii} = 0$ otherwise. We can then write the expected projection error in terms of the spectral decomposition:

$$\varepsilon_0(k) = \mathbb{E}\big[\|\boldsymbol{P}_{\boldsymbol{H}}\boldsymbol{U}\|_{\mathrm{F}}^2\big] = \mathbb{E}\big[\big\|\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top\boldsymbol{U}\big\|_{\mathrm{F}}^2\big]. \tag{21}$$

We can remove the left $\boldsymbol{Q}$ from this equation because it is an orthogonal matrix and so does not change the norm. Furthermore, as $\boldsymbol{Q}^\top\boldsymbol{U}$ is simply an orthogonal transformation of unit vectors, it will have the same distribution as $\boldsymbol{U}$. We can therefore simplify the above as

$$\begin{aligned} \varepsilon_0(k) = \mathbb{E}\big[\|\boldsymbol{\Lambda}\boldsymbol{U}\|_{\mathrm{F}}^2\big] &= \mathbb{E}\left[\sum_{i=1}^{n-k}\sum_{j=1}^{k} U_{ij}^2\right] \\ &= (n-k)k\cdot\mathbb{E}\big[U_{ij}^2\big] = \frac{(n-k)k}{n}, \end{aligned} \tag{22}$$

where we have made use of the fact that $\boldsymbol{\Lambda}$ simply picks out the first $n-k$ rows from $\boldsymbol{U}$ and then used our previously established result on $\mathbb{E}[U_{ij}^2]$.

The above derivation assumes that $\boldsymbol{U}$ and $\boldsymbol{H}$ are statistically independent of each other. However, in our actual calculations the eigenvectors will correspond to dominant eigenvectors of the uniform random walk matrix. Hence, we know that $\mathbf{1}$ is always included in $\boldsymbol{V}_k$ and moreover, since $\boldsymbol{H}\mathbf{1} = \mathbf{1}$ for any partition indicator matrix $\boldsymbol{H}$, we know that there is always a one-dimensional subspace shared between the subspace spanned by $\boldsymbol{P}_H$ and $\boldsymbol{V}_k$. As we show in Appendix C, we thus have to adjust the expected error to

$$\mathbb{E}[\varepsilon(\widehat{\boldsymbol{H}}, \boldsymbol{U})] = \mathbb{E}\big[\|\boldsymbol{P}_{\widehat{\boldsymbol{H}}}\boldsymbol{U}\|_{\mathrm{F}}^2\big] = \frac{(n-k)(k-1)}{n-1}.$$

The intuition here is that we have to exclude the subspace spanned by $\mathbf{1}$ and are now looking for the projection of a $(k-1)$-dimensional (rather than k-dimensional) subspace in an $(n-1)-(k-1) = (n-k)$-dimensional space. Moreover, both of these subspaces are restricted to be orthogonal to $\mathbf{1}$ and thus the degree of freedom for choosing such subspaces is reduced, resulting in the change of the denominator from n to $n-1$. In other words, our calculations have to account for the fact that we know that there is a one-dimensional EEP present in any connected graph. Since the expected error only depends on the number of groups k and not the specific partition $\widehat{\boldsymbol{H}}$, we will refer to the above expected error simply as $\varepsilon_0(k)$,

$$\varepsilon_0(k) := \frac{(n-k)(k-1)}{n-1}. \tag{23}$$

The error $\varepsilon_0(k)$ above is a good null model to test the hypothesis that a single approximate EEP exists in the network because it is the expected error when there are no approximate EEPs in the network. However we are ultimately interested in detecting hierarchies, i.e., nested sequences of approximate EEPs, so we need to create an alternative hypothesis that

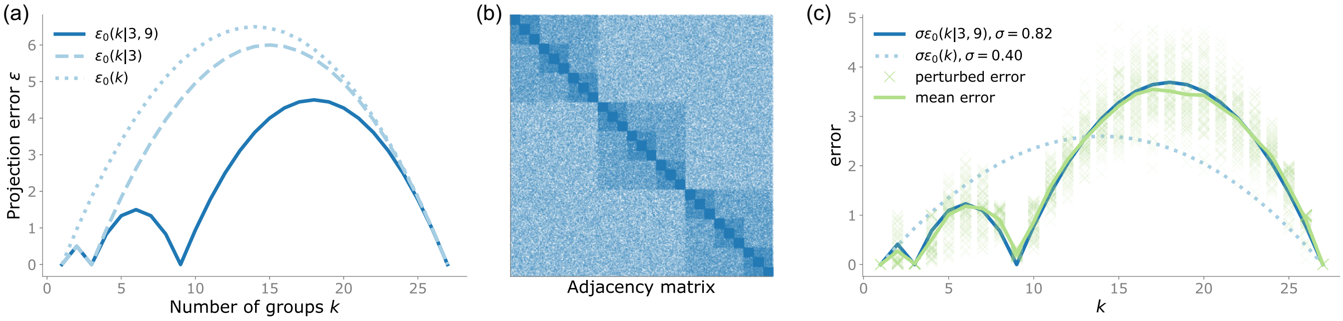

FIG. 7. Identifying levels in the hierarchy. (a) The expected projection error $\varepsilon_0(k)$ [Eq. (23)] assuming there are no hierarchical levels and the conditional expected errors $\varepsilon_0(k|\kappa)$ for $\kappa = \{3\}, \{3, 9\}$ [Eq. (24)]. (b) A spy plot of the adjacency matrix for a network with hierarchical partitions into 3, 9 and 27 groups. (c) Comparison of mean perturbed error against the expected error of the null hypothesis (no EEPs) $\sigma\varepsilon_0(k)$ and an alternative hypothesis $\sigma\varepsilon_0(k|3, 9)$ (EEPs into 3 and 9 groups). We set σ in each case to minimize the mean squared logistic error [Eq. (29)]. We clearly see the correspondence between mean error and $\varepsilon_0(k|3, 9)$. Crosses indicate the distribution of projection errors for 10^2 random perturbations [Eq. (27)].

accounts for the presence of other potential EEPs in the network. Specifically, if we would like to test the hypothesis that the eigenvectors $\boldsymbol{V}_k$ include a subset of κ eigenvectors of a coarser-grained approximate EEP into κ groups (i.e., $\kappa < k$). Then we calculate the expected error $\varepsilon_0(k|\kappa)$ conditioned on an existing EEP into κ groups as (see Appendix C for details)

$$\varepsilon_0(k|\kappa) = \begin{cases} \frac{(n-k)(k-\kappa)}{n-\kappa} & \text{if } \kappa \leqslant k \leqslant n \\ \frac{(\kappa-k)(k-1)}{\kappa-1} & \text{if } 1 \leqslant k \leqslant \kappa \end{cases}. \qquad (24)$$

Figure 7(a) illustrates these expected error functions. The expected error $\varepsilon_0(k)$ for when there are no further approximate EEPs is shown as the dotted parabola. However, if there is clear evidence for other levels in the hierarchy, then we need to adjust our expected error to account for these. For example, the network represented by the spy plot in Fig. 7(b) has hierarchical partitions into 3, 9, and 27 groups. To account for these possible levels we can calculate the conditional expected errors $\varepsilon_0(k|\kappa_1 = 3)$ and $\varepsilon_0(k|\kappa_1 = 3, \kappa_2 = 9)$ shown in Fig. 7(a), according to the general formula:

$$\varepsilon_0(k|\kappa_1, \dots, \kappa_c) = \begin{cases} \frac{(n-k)(k-\kappa_c)}{n-\kappa_c} & \text{if } \kappa_c \leqslant k \leqslant n, \\ \vdots & \\ \frac{(\kappa_2-k)(k-\kappa_1)}{\kappa_2-\kappa_1} & \text{if } \kappa_1 \leqslant k \leqslant \kappa_2, \\ \frac{(\kappa_1-k)(k-1)}{\kappa_1-1} & \text{if } 1 \leqslant k \leqslant \kappa_1, \end{cases} \qquad (25)$$

which can be derived analogously to Eq. (24) for the general case.

We can decide if our candidate EEP $\widehat{\boldsymbol{H}}$ is significant by comparing the expected error without EEPs $\varepsilon_0(k)$ and with EEPs $\varepsilon_0(k|\kappa)$ with the observed error. However, before we do so, we must take precautions to prevent detecting degenerate hierarchies.

2. *Spectral signatures of degenerate EEPs and hierarchies*

By comparing the observed projection error for each putative partition using the above derived formulas, we can assess whether or not a partition is significantly close to being an sEEP. However, as stated previously, we want to avoid constructing degenerate hierarchies, and thus we do not want to accept all possible sEEP as new hierarchical levels.

To see how this can be done, let us return to the example of a flat, nonhierarchical partition generated from a planted partition model. After we found the split into k groups, we treat the affinity matrix $\boldsymbol{\Omega} = (a-b)\boldsymbol{I} + b\boldsymbol{1}\boldsymbol{1}^\top$, as a $k \times k$ weighted adjacency matrix. The corresponding Laplacian is $\boldsymbol{L}(\boldsymbol{\Omega}) = (k-1)b\boldsymbol{I} - b\boldsymbol{1}\boldsymbol{1}^\top$ and is easily identifiable as a flat partition from its spectrum: the Laplacian $\boldsymbol{L}(\boldsymbol{\Omega})$ has an eigenvalue $\lambda_1 = 0$, associated with the constant eigenvector $\boldsymbol{1}$, and $(k-1)$ repeated eigenvalues $\lambda_r = kb$, for $2 \leqslant r \leqslant k$, associated with an invariant subspace of dimension $k-1$. These repeated eigenvalues of $\boldsymbol{L}(\boldsymbol{\Omega})$ clearly identify that there is no further structure in $\boldsymbol{\Omega}$ and there is an inherent symmetry associated to the groups. The implication for our flat partition is that there exists a set of orthogonal matrices $\mathcal{V}$,

$$\mathcal{V} = \{\boldsymbol{V} \in \mathbb{R}^{k\times(k-1)} | \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I} \text{ and } \boldsymbol{V}^\top\boldsymbol{1} = \boldsymbol{0}\}, \qquad (26)$$

where the columns of *every* matrix in $\mathcal{V}$ form a valid set of linearly independent eigenvectors for $\boldsymbol{L}$.

Consider now assessing the projection error of an EEP with indicator matrix $\boldsymbol{H}_\kappa$ that forms a partition on $\boldsymbol{\Omega}$ into κ groups, where $1 < \kappa < k$. We know that there exists a matrix $\boldsymbol{V}_\kappa$ containing κ dominant eigenvectors for which the projection error $\boldsymbol{P}_{\boldsymbol{H}_\kappa}\boldsymbol{V}_\kappa$ is exactly zero. Based on the above observation it is easy to see that these κ eigenvectors correspond to a particular choice of the first κ dominant eigenvectors that are associated with one possible way to partition the network into κ groups. Given that we have a flat partition, we know that *any* partition into κ groups will form an EEP and that for each partition there exists a corresponding set of κ eigenvectors for which the projection error is zero. However, any given eigenvector matrix $\boldsymbol{V}_\kappa$ can only be piecewise constant on one of the $S(k, \kappa)$ possible EEPs into κ groups, where the Sterling partition number $S(n, k)$ is the number of ways to partition a set of n objects into k nonempty subsets. So although we can only obtain κ independent dominant eigenvectors, there are far more possible EEPs with κ groups, which indicates that the eigenspace is degenerate.

The above argument can be applied analogously to situations where there are more than one level in the hierarchy and a nonidentifiable set of compatible EEPs. To capture such situations, we say that an EEP into κ groups with indicator matrix $\boldsymbol{H}$ is *degenerate*, if some of the structural eigenvectors associated to $\boldsymbol{H}$ are contained within a degenerate eigenspace. Notably, the situation here is analogous to the situation we already considered before: we are effectively picking an arbitrary subspace (corresponding to degenerate structural eigenvectors of an EEP) out of a larger degenerate eigenspace.

3. Avoiding degenerate hierarchies

We can use the degeneracy of EEPs to our advantage to avoid finding "spurious" hierarchical levels within our framework as follows. Recall that to find an EEP into κ groups based on $\boldsymbol{\Omega}$, we consider the first dominant eigenvectors of $\boldsymbol{W}(\boldsymbol{\Omega})$. Now assume that the obtained EEP into κ groups is indeed degenerate. When we numerically compute the first dominant κ eigenvectors, we are presented with one specific (but arbitrary) choice of eigenvectors, which depends on the specific details of the algorithm implemented. However, applying a small random perturbation to the affinity matrix will, with high probability, result in a different set of eigenvectors that relate to a different EEP. This idea also readily applies to the practical case in which we only have an estimate of the affinity matrix, $\widehat{\boldsymbol{\Omega}}$. The corresponding eigenspaces are only approximately degenerate since the eigenvalues will, in general, be only approximately equal.

Consider the uniform random walk matrix $\boldsymbol{W}$ of an estimated affinity matrix $\widehat{\boldsymbol{\Omega}}^{(u)}$ and a perturbed version $\boldsymbol{W}_{\mathrm{p}} := \boldsymbol{W}(\widehat{\boldsymbol{\Omega}}_{\mathrm{p}}^{(u)})$ corresponding to an affinity matrix with a slight perturbation. We can estimate a partition $\widehat{\boldsymbol{H}}$ using spectral clustering on $\boldsymbol{W}(\widehat{\boldsymbol{\Omega}}^{(u)})$. Based on the Davis-Kahan theorem (and following an argument analogous to that in Appendix A), we see that the difference between the eigenvectors of $\boldsymbol{W}$ and $\boldsymbol{W}_{\mathrm{p}}$ will depend on how close the eigenvalues of $\boldsymbol{W}$ are to being degenerate. Specifically, if the obtained eigenvectors of $\boldsymbol{W}_{\mathrm{p}}$ and $\boldsymbol{W}$ are very similar, and the partition $\widehat{\boldsymbol{H}}$ is indeed an approximate EEP of $\boldsymbol{W}$, then both the projection error $\varepsilon(\widehat{\boldsymbol{H}}, \boldsymbol{V}(\boldsymbol{W}))$ and the projection error $\varepsilon(\widehat{\boldsymbol{H}}, \boldsymbol{V}(\boldsymbol{W}_{\mathrm{p}}))$ will be small and significant (in the manner described in Section IV C 1). The robustness to small perturbations indicates that the found EEP is nondegenerate. However, if a small perturbation creates a $\boldsymbol{W}_{\mathrm{p}}$ whose eigenvectors have large projection error with respect to the partition $\widehat{\boldsymbol{H}}$ estimated from $\boldsymbol{W}$, then we know that the EEP corresponds to a degenerate configuration.

In practice, once we have inferred the finest level partition into $k^{(u)}$ groups and estimated $\widehat{\boldsymbol{\Omega}}^{(u)}$, for each $k_i \in \{2, \dots, k^{(u)} - 1\}$, we estimate a partition $\widehat{\boldsymbol{H}}_{k_i}$ using the k_i dominant eigenvectors of $\boldsymbol{W}$. We then add a perturbation to the estimated affinity matrix such that

$$\widehat{\boldsymbol{\Omega}}_{\mathrm{p}}^{(u)} = \widehat{\boldsymbol{\Omega}}^{(u)} + \gamma \boldsymbol{\Gamma}, \tag{27}$$

$$\gamma = \gamma' \frac{\|\widehat{\boldsymbol{\Omega}}^{(u)}\|_2}{\|\boldsymbol{\Gamma}\|_2}, \tag{28}$$

where $\boldsymbol{\Gamma}$ is a symmetric matrix of i.i.d. random pertubations, $\|\cdot\|$ stands for the induced 2-norm (operator norm), and the prefactor γ scales the perturbation of the affinity matrix to have a constant relative strength of γ' (measured in terms of the ℓ_2 norm).

Taking the average over perturbations gives us a mean error $\epsilon(\widehat{\boldsymbol{H}}_{k_i}, \boldsymbol{V}_{k_i})$ that we can compare against the expected errors described in the previous section. We perform this comparison using the mean squared logistic error (MSLE):

$$\mathrm{MSLE}(k) = \frac{1}{k} \sum_{k_i=1}^{k} \left(\ln\left[\epsilon\left(\widehat{\boldsymbol{H}}_{\boldsymbol{k}_i}, \boldsymbol{V}_{k_i}\right) + 1\right] - \ln[\sigma \varepsilon_0(k_i) + 1] \right)^2, \tag{29}$$

where σ is a scale parameter that we set by minimizing the MSLE. The MSLE is a regularized relative error that has the property of incurring a greater penalty when the expected error is small. This is desirable because we are more concerned with identifying the troughs, to locate approximate EEPs, than we are with matching the curvature of the peaks. Figure 7(c) illustrates this comparison between the mean perturbed error and the expected error without EEPs, $\sigma \varepsilon_0(k)$ ($\sigma = 0.40$), and with EEPs, $\sigma \varepsilon_0(k|3, 9)$ ($\sigma = 0.82$). The mean error clearly is a better match with $\sigma \varepsilon_0(k|3, 9)$ indicating that there are significant EEPs into 3 and 9 groups.

4. Building a dendrogram

Putting all of the above together, we can detect hierarchies by first identifying the finest partition and using this to estimate the affinity matrix $\widehat{\boldsymbol{\Omega}}^{(1)}$, which we treat as a weighted network. Next we use this weighted network to identify possible partitions into $k_i \in \{2, \dots, k^{(1)} - 1\}$ groups and compute the corresponding projection errors (averaged over 20 perturbations) as a function of k_i. We then build up a set of candidate partitions $\{\widehat{\boldsymbol{H}}_{\kappa_i}\}$ using a greedy heuristic. First we find the κ_1 that minimizes the MSLE between the mean perturbed projection error $\epsilon(\widehat{\boldsymbol{H}}_{\boldsymbol{k}}, \boldsymbol{V}_k)$ and the expected error $\sigma \varepsilon_0(k|\kappa_1)$, i.e.,

$$\arg\min_{\kappa_1} = \mathrm{MSLE}(k|\kappa_1), \quad \kappa_1 \in \{2, \dots, k-1\}. \tag{30}$$

If $\mathrm{MSLE}(k|\kappa_1) < \mathrm{MSLE}(k)$ then we add $\widehat{\boldsymbol{H}}_{\kappa_1}$ to the set of candidate partitions. We repeat this process to add significant partitions (into κ_2, κ_3 etc.) until there is no further reduction in the MSLE. Note that there is no restriction on the ordering of these candidate partitions, so $\kappa_i > \kappa_{i+1}$ or $\kappa_i < \kappa_{i+1}$. This results in a set of candidate agglomerations into $\{\kappa\}$ groups. We pick the maximal $\max_i\{\kappa_i\}$, i.e., the finest approximate EEP to form the next level in the hierarchy and form the new affinity matrix $\widehat{\boldsymbol{\Omega}}^{(2)}$. We repeat this whole process until we no longer identify significant partitions.

Full details of the precise algorithm are given in Appendix E. A reference PYTHON implementation of the here presented algorithms will be made available in Ref. [68].

V. NUMERICAL EXPERIMENTS ON SYNTHETIC DATA

We validate the spectral algorithm introduced above for hierarchical community detection on a number of classes of synthetic networks with planted hierarchies: assortative, disassortative, symmetric, and asymmetric hierarchies.

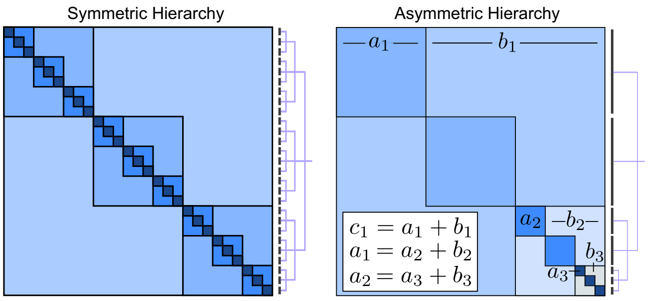

FIG. 8. Schematic: expected adjacency matrices of synthetic hierarchical test networks. We consider a symmetric and an asymmetric hierarchical network construction. Both start from a planted partition model with a specified signal to noise ratio. We then iteratively refine the hierarchy by treating the network induced by each subnetwork as another instance of a planted partition model with the same signal to noise ratio. Here we impose the additional contraint that the expected degree (the average connection probability) of the nodes in this subnetwork is such that it matches with the specification of the layer above (see text for details). In the symmetric variant of our model, each group is recursively subdivided such that we obtain a hierarchy of 3 × 3 × 3 = 27 groups. In the asymmetric varianot of the model, only one of the groups is subdivided further, leading highly skewed group sizes at the lowest level of the hierarchy (see the indicated dendrogram).

A. Experimental setup

The synthetic network models are based on iteratively applying a planted partition model structure as follows. We start with a planted partition model for a graph with n nodes and $k = 3$ groups. We denote the probability of a link between a pair of nodes in the same group by α/n, and denote the probability of a link between a pair of nodes in different groups by β/n. We set the parameters α, β by fixing an expected degree c_1 for each node and signal-to-noise ratio SNR, defined as

$$\mathrm{SNR}(\alpha, \beta) = \frac{(\alpha - \beta)^2}{k\alpha + k(k-1)\beta}. \tag{31}$$

SNR = 1 corresponds precisely to the detectability limit of the SBM [18,22]. For each node, the expected number of connections to nodes in the same group is $a_1 = \alpha/k$, and the expected number of connections to nodes in different groups is $b_1 = \beta(k-1)/k$, such that the total expected degree for each node is $c_1 = a_1 + b_1$.

Next we recursively plant finer partitions, while maintaining the average node degree. We divide each of the k groups again into k subgroups, such that the expected degree of the nodes in this subnetwork is $c_2 = a_1 = a_2 + b_2$, consistent with the coarser, initial planted partition. Figure 8 illustrates a schematic of these parameters for the symmetric and asymmetric hierarchies. The parameters a_2 and b_2 (respectively their connection probabilities) within each subnetwork are chosen such that the specified SNR is maintained.

We validate our method for each class of network models for varying levels of SNR. To evaluate the similarity of two partitions we use the adjusted mutual information score [69] defined as

$$\mathrm{AMI}(\boldsymbol{H}_1,\boldsymbol{H}_2) = \frac{I(\boldsymbol{H}_1, \boldsymbol{H}_2) - \mathbb{E}[I(\boldsymbol{H}_1, \boldsymbol{H}_2)]}{(\mathrm{Ent}(\boldsymbol{H}_1) + \mathrm{Ent}(\boldsymbol{H}_2))/2 - \mathbb{E}[I(\boldsymbol{H}_1, \boldsymbol{H}_2)]},$$

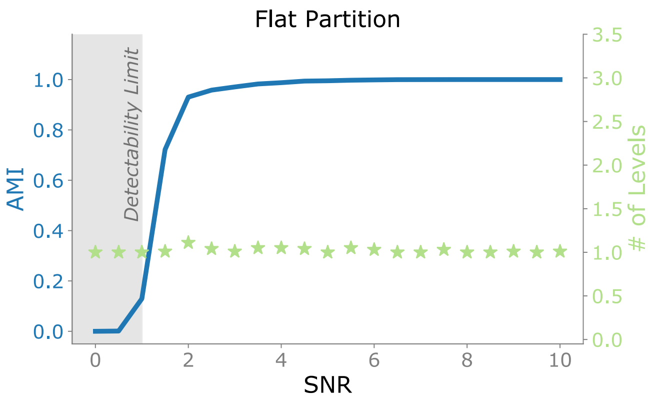

FIG. 9. Detecting the absence of hierarchy (flat partitions). We plant a flat partition into 64 groups, similar to the example in Fig. 1. Overall our method is consistent in identifying a flat partition across the full range of SNR values.

where $I(\boldsymbol{H}_1, \boldsymbol{H}_2)$ and $\mathbb{E}[I(\boldsymbol{H}_1, \boldsymbol{H}_2)]$ are the mutual information and its expected value respectively, and $\mathrm{Ent}(\cdot)$ is the Shannon entropy of the partition assignment. Here the expectation is taken over the so-called permutation null model [69], in which partitions are generated uniformly at random subject to the constraint that the number of clusters and points in each clusters are commensurate with the inputs [69,70]. Note that the AMI score typically lies in the range [0,1][1] with 0 denoting a result as expected by chance and 1 perfect recovery.

We denote the ℓ planted partitions within our model networks as $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_\ell$ and denote $\hat{\ell}$ hierarchical partitions detected by our algorithm as $\widehat{\boldsymbol{H}}_1, \ldots, \widehat{\boldsymbol{H}}_{\hat{\ell}}$. Using the AMI score, we define the score matrix Ξ with entries

$$\Xi_{i,j} = \mathrm{AMI}(\boldsymbol{H}_i, \widehat{\boldsymbol{H}}_j) \quad \text{for } i = 1, \ldots, \ell,\ j = 1, \ldots, \hat{\ell}, \tag{32}$$

that measures the pairwise matching between any of the planted and recovered partitions. We summarize the detection performance in the score matrix using precision and recall, defined as

$$\mathrm{Precision} = \frac{1}{\hat{\ell}} \sum_j \max_i \Xi_{i,j}, \tag{33}$$

$$\mathrm{Recall} = \frac{1}{\ell} \sum_i \max_j \Xi_{i,j}. \tag{34}$$

The precision is large if, for every estimated partition, there is a planted partition that provides a good match. The recall is large if for every planted partition, there is an estimated partition that matches closely.

B. Results

In our first experiment we confirm that our approach does not identify degenerate hierarchies. We plant a flat partition into 64 groups using a planted partition model, akin to the example in Fig. 1, and vary the SNR. Figure 9 shows that our

[1] It is possible to have slightly negative AMI values due to the adjustment for chance.

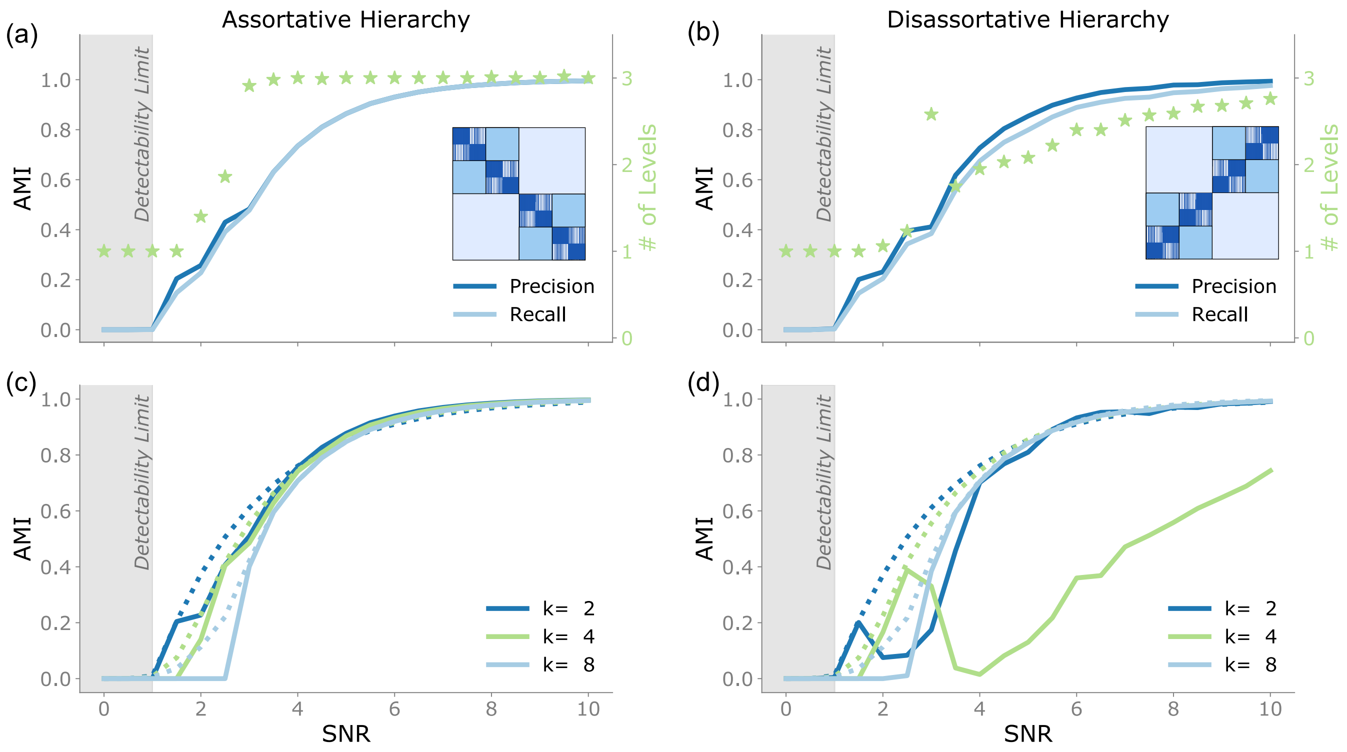

FIG. 10. Detecting assortative and disassortative hierarchical communities planted in synthetic test networks. Synthetic networks are drawn from the assortative and disassortative hierarchical random graph model with $n = 2^{14} \approx 16,400$ nodes with average degree 50. Levels in the hierarchy are partitioned into 2, 4, and 8 groups. (a) Results for the assortative hierarchical network model (see inset for schematic). We show the precision and recall statistic of the overall hierarchy, as defined in the text, as a function of the signal-to-noise ratio (SNR). Stars denote the number of hierarchical levels identified by our algorithm (right y axis). (b) The corresponding results for the disassortative hierarchical network model (see inset for schematic). (c) The mean adjusted mutual information (AMI) of the best matching inferred partition with each level in the planted assortative hierarchy. The dotted lines indicate the performance of spectral clustering with the Bethe Hessian with known number of groups. (d) The mean AMI of the best matching inferred partition with each level in the planted disassortative hierarchy. We observe poorer performance in recovering the disassortative hierarchy, in particular the level into $k = 4$ groups because of the degeneracy of disassortative groups (see text for details).

approach is broadly consistent at identifying a single partition in the absence of a hierarchy.

Next we consider assortative and disassortative hierarchies. In both cases we generate symmetric hierarchical partitions into 2, 4, and 8 groups. We generate the disassortative hierarchies in the same way as the assortative hierarchies, except that we reverse the columns of the affinity matrix $\mathbf{\Omega}^{(1)}$ before generating the network [see insets Figs. 10(a) and 10(b)].

Figure 10 shows the performance in recovering the assortative [(a) and (c)] and disassortative [(b) and (d)] hierarchies. In the case of the assortative hierarchy we see that the performance increases monotonically with the SNR, both overall [Fig. 10(a)] and at each level [Fig. 10(c)]. We observe poorer overall performance in recovering the disassortative hierarchies and require a much higher SNR to consistently identify three levels in the hierarchy [Fig. 10(b)]. Closer inspection of the performance at individual levels [Fig. 10(d)] shows that we can recover the finest partition into 8 groups using the Bethe Hessian with comparable performance as the assortative case. We can also detect the coarsest partition into 2 groups relatively well, particularly at SNR $>$ 4. However the middle level is harder to detect. The reason for the poorer performance is due to a degeneracy that occurs for disassortative partitions meaning that we have multiple distinct ways to form an EEP into 4 groups [71]. This degeneracy creates an identifiability issue, similar to the one described in Fig. 6 (see Appendix D for a visual description), and means that our algorithm often fails to detect a level in the hierarchy that partitions the network into 4 groups. Identifiability issues notwithstanding, these results indicate that our approach is still effective at recovering disassortative hierarchies.

Finally, we examine the performance of recovering symmetric versus asymmetric hierarchies. Figure 11 displays the results for a symmetric hierarchy with three partitions into 3, 9, and 27 groups [Figs. 11(a) and 11(c)] alongside results for an asymmetric hierarchy partitioned into 3, 5, and 7 groups. Our algorithm shows overall good performance: not only do we recover the correct partition at the finest level, we can also detect right until the detectability limit. The fact that the precision and recall measures are well aligned indicates that our algorithm successfully rejects spurious hierarchical levels, as can also be seen from the number of hierarchical levels found (indicated by orange asterisks in Fig. 11). We detect additional levels only in a limited number of cases where the SNR increases sufficiently such that the intermediate levels become well defined.

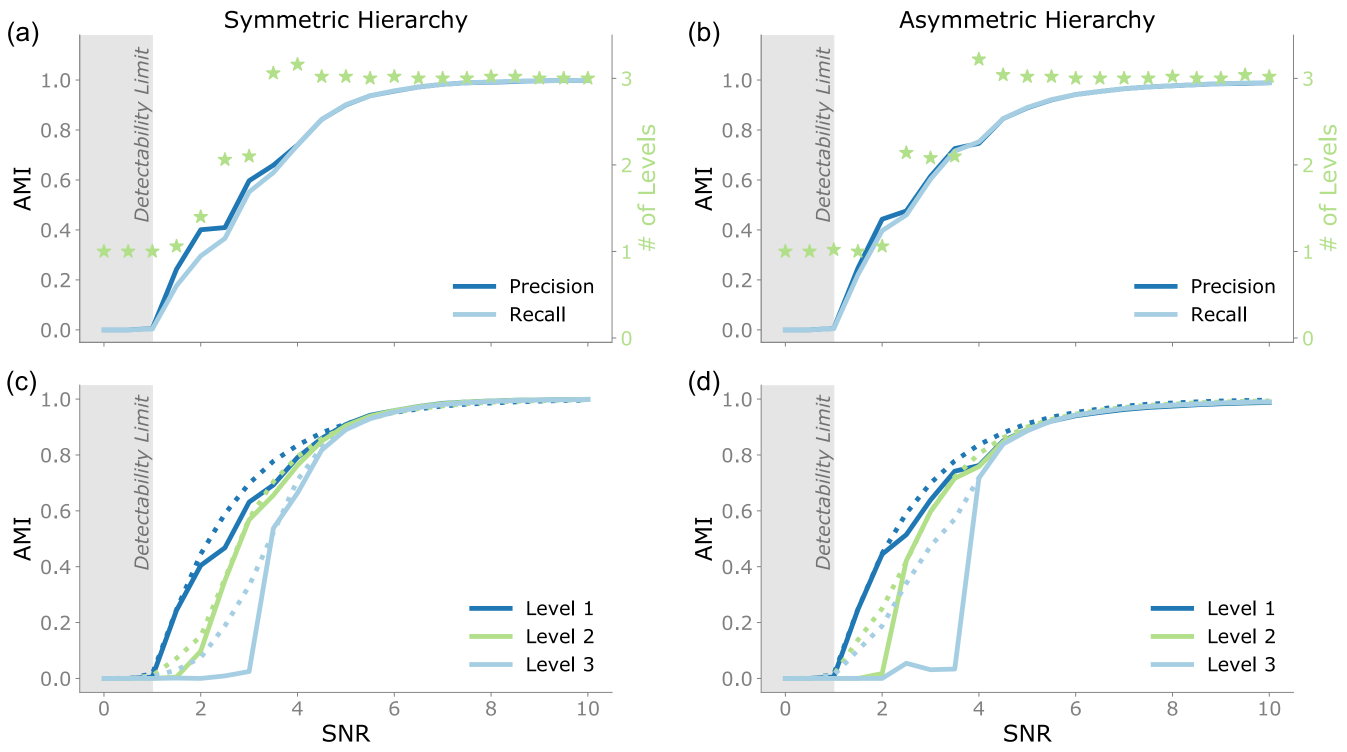

FIG. 11. Detecting symmetric and asymmetric hierarchical communities planted in synthetic test networks. Synthetic networks drawn are from the symmetric and asymmetric hierarchical random graph model with $n = 3^9 \approx 19,700$ nodes with average degree 50. Levels in the symmetric hierarchy are partitioned into 3, 9, and 27 groups, whereas the levels in the asymmetric hierarchy are partitioned into 3, 5, and 7 groups. (a) Precision and recall for the symmetric hierarchical networks as a function of signal-to-noise ratio (SNR). Stars denote the number of hierarchical levels detected by our algorithm (right y axis). (b) Precision and recall for the asymmetric hierarchical networks as a function of SNR. (c) The mean adjusted mutual information (AMI) of the best matching inferred partition with each level in the planted symmetric hierarchy. The dotted lines indicate the performance of spectral clustering with the Bethe Hessian with known number of groups. (d) The mean AMI of the best matching inferred partition with each level in the planted asymmetric hierarchy.

VI. DETECTING HIERARCHICAL STRUCTURES IN REAL-WORLD DATA

To validate our method on real-world networks, we consider a face-to-face contact network and a word-association network, described in the subsequent sections. The standard SBM has a well-known weakness for modeling real-world networks because, for network generated by the SBM, the degrees of nodes within a group are Poisson distributed [15]. Real-world networks tend to have a more heterogeneous degree distribution, which has motivated various forms of degree correction [15,72]. However, the Bethe Hessian is more robust to degree heterogeneity, but we further improve this by adjusting the regularization parameter according to Ref. [59] (see Algorithm 5 in Appendix E for details). Because our approach is agglomerative, where subsequent steps of the algorithm simply merge groups from the previous level, it is only necessary to account for degree heterogeneity in the initial detection of communities.

A. High-school network

We first consider a social contact network within a high school [73] to identify the presence of possible hierarchical structure. The network consists of $n = 327$ nodes and $m = 5818$ edges, denoting face-to-face contacts between students wearing RFID tags. The students are divided into nine classes according to their subject specialization: math and physics (MP, 3 classes), biology (BIO, 3 classes), physics and chemistry (PC, 2 classes), and engineering (PSI, 1 class).

Figure 12 shows the hierarchy that we identify using our spectral algorithm. We see that the hierarchical organization in the social contact structure of the network matches the class structure of the school. Specifically, individual classes are identified as individual communities at the finest level, which in turn merge with classes with the same specialization. Finally, the coarsest partition splits the students into two groups: those that specialize in biology and those whose specializations involve physics.

B. Word associations network

We constructed a network of English word associations using data from The Small World of Words project [74], a scientific project to map word meaning in various languages. The dataset was created based on a word association task, in which participants are asked to give three associated responses to a given cue word. The dataset includes over 3 million responses obtained from over 90 000 participants, for more than 12 000 cues. We created a network of stemmed

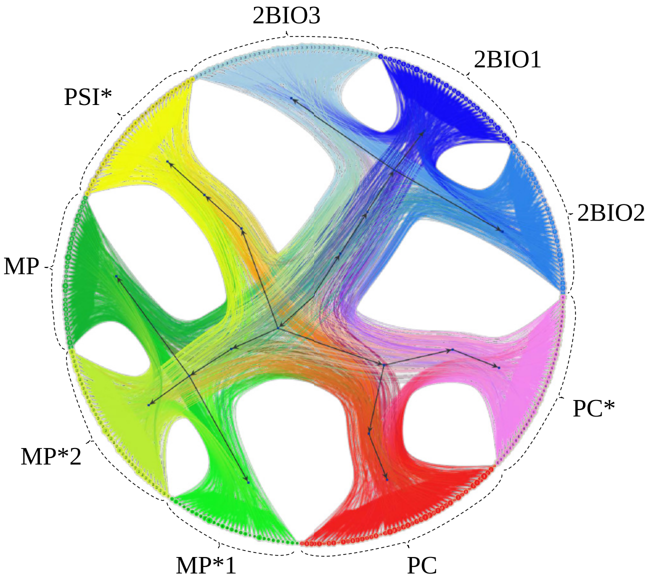

FIG. 12. Hierarchical community structure in a high-school social contact network. Using spectral clustering, we find 9 communities corresponding to the classes in the high-school network (as indicated by colors above). There are 3 classes specialized on math and physics (MP), 3 biology focused classes (BIO), 2 physics and chemistry focused classes (PC), and 1 class specialized for engineering (PSI). As depicted above we find a hierarchical structure commensurate with these specializations, using our spectral method.

words as nodes and cue–response pairs as edges, and applied our algorithm to identify the hierarchical structure of communities.

Figure 13 shows the dendrogram representing the detected hierarchical structure. Here we see at the coarsest level a partition into three groups that forms a core-periphery type of structure. The nodes in the dense core have a higher proportion of in-group links and are more likely to represent a cue word. The finer partitions of the core represent groups of words that are clearly associated, whereas the periphery contains groups of words that are less clearly associated due to the disassortative nature of the communities (i.e., lower proportion of in-group links).

VII. CONCLUSION

We have presented a thorough investigation on hierarchical community structure in networks. By introducing the concept of a stochastic externally equitable partition, we have provided a formal definition of hierarchical community structure that consists of a series of nested, nondegenerate stochastic externally equitable partitions. Stochastic externally equitable partitions provide a natural generalization of several concepts of node equivalence. In particular, it has a close relationship to the stochastic equivalence relation that underlies the stochastic block model. In light of our new definition of hierarchical community structure, we have discussed several identifiability issues that apply in general to the detection of hierarchical

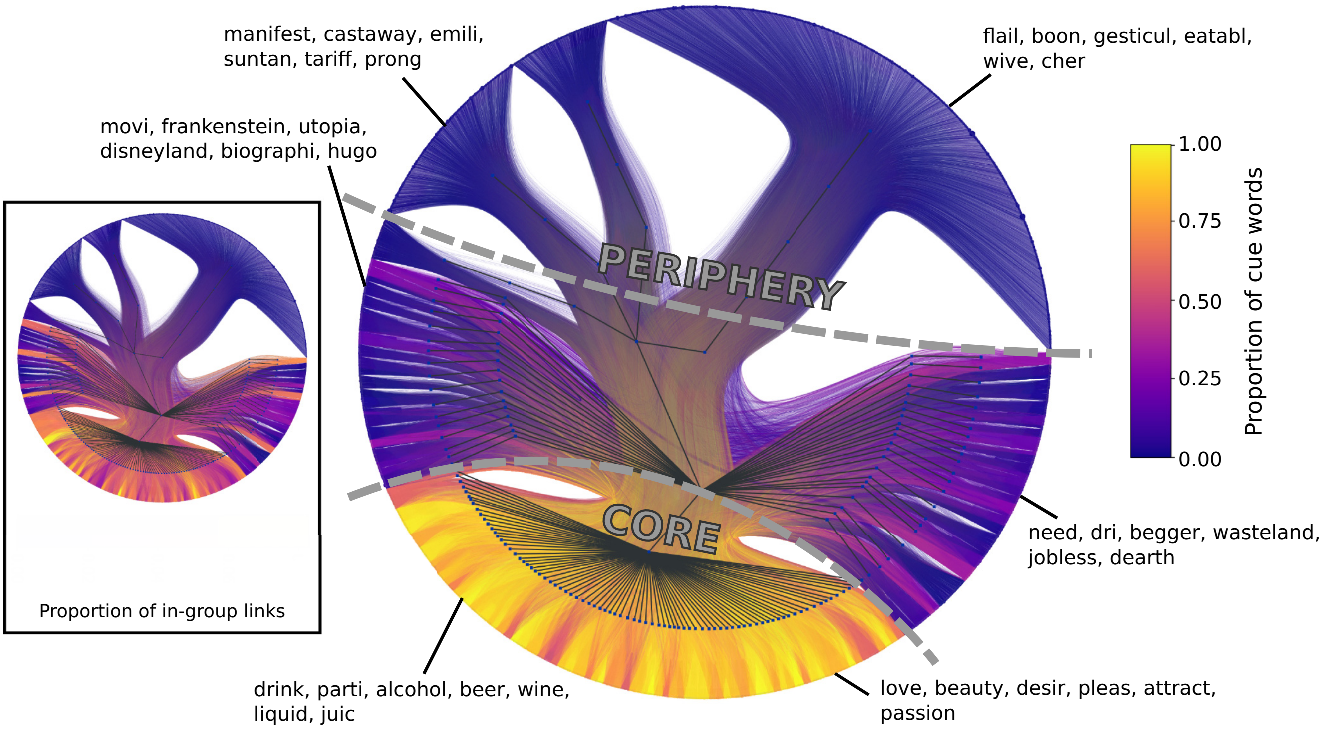
manifest, castaway, emili, suntan, tariff, prong
flail, boon, gesticul, eatabl, wive, cher
movi, frankenstein, utopia, disneyland, biographi, hugo
PERIPHERY
CORE
1.00
0.75
0.50
0.25
0.00
Proportion of cue words
need, dri, begger, wasteland, jobless, dearth
Proportion of in-group links
drink, parti, alcohol, beer, wine, liquid, juic
love, beauty, desir, pleas, attract, passion

FIG. 13. Hierarchical structure in a word association network. Using our spectral method developed here, we find a hierarchical core-periphery structure in this network, as depicted above. Each node in the network corresponds to a word (either a cue or a response), and two words are connected if they form a cue-response pair. Color depicts the amount of cue-words within a group. The core is formed of densely knit set of words corresponding mostly to cue words, whereas the more peripheral communities are formed mostly from response words (noncue words).

community structure. Specifically, we have identified a number of scenarios for which multiple good solutions exist. In these cases, the choice of which hierarchy is detected will be based on the specific bias of the detection method employed. We have also discussed how naïve use of hierarchical models, such as Ref. [4], may identify spurious hierarchies, in much the same way that community detection algorithms might identify spurious communities in an Erdős-Rényi network. In addition, we have identified characteristic spectral properties of hierarchical stochastic EEPs and developed a simple, efficient algorithm for hierarchical community detection that exploits these properties.

Our work opens a number of avenues for future research. On a theoretical level, our work lays the foundations for more detailed analysis of the asymptotic limits of detectability of community structure, particularly for networks that contain communities at multiple resolutions, as is the case for hierarchical communities [71]. Our experimental results further emphasize the issues of identifiability, in particular for disassortative hierarchies. We see that disassortative hierarchies are more likely to have degenerate solutions that make it harder to detect levels in the hierarchy and/or identify the specific planted partition over an equivalently good alternative solution. These observations warrant further investigation into the degeneracy of disassortative partitions, something that has been largely overlooked so far, possibly due to the bias in the literature towards assortative community structure. One potential solution to deal with the identifiability issues might be to incorporate a notion of equivalent hierarchies into the scoring functions we use to evaluate performance. We already employ a similar approach in community detection to deal with the fact that communities are invariant to their specific label assignment. However, this is not a consideration we have encountered so far in the body of work concerned with evaluating (hierarchical) community detection performance [75–77]. From an algorithmic perspective, we have focused on an agglomerative procedure that relies on accurately detecting the finest level in the hierarchy. Any errors in recovering the finest partition will be propagated to subsequent levels. However, it may be that a divisive algorithm could perform better in some settings, particularly if the coarser partitions contain a stronger community structure that is easier to detect. Investigating the relative benefits and weaknesses of agglomerative versus divisive algorithms may thus be a fruitful avenue for future research.

ACKNOWLEDGMENTS

The authors would like to thank Jean-Charles Delvenne, Karel Devriendt, Mauro Faccin, Renaud Lambiotte, Tiago Peixoto, Karl Rohe, and Michael Scholkemper for helpful conversations. MTS was partially supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 702410, and by the Ministry of Culture and Science (MKW) of the German State of North Rhine-Westphalia ("NRW Rückkehrprogramm").

APPENDIX A: CONSISTENCY OF THE PROJECTION ERROR CRITERION

Central to our approach is the identification of stochastic EEPs, which satisfy

$$\boldsymbol{L}(\boldsymbol{\Omega})\boldsymbol{H} = \boldsymbol{H}\boldsymbol{L}^{\pi} \quad \boldsymbol{H} \in \mathcal{H}_{\mathrm{EEP}}^{\Omega}, \tag{A1}$$

where $\mathcal{H}_{\mathrm{EEP}}^{\Omega}$ is the set of external equitable partitions of $\boldsymbol{\Omega}$, $\boldsymbol{L}^{\pi}$ is the Laplacian of the quotient graph,

$$\boldsymbol{L}^{\pi} = \boldsymbol{N}^{-1}\boldsymbol{H}^{\top}(\boldsymbol{D} - \boldsymbol{\Omega})\boldsymbol{H}. \tag{A2}$$

Since we do not observe the affinity matrix $\boldsymbol{\Omega}$, we estimate the corresponding affinity matrix $\widehat{\boldsymbol{\Omega}}$ as

$$\widehat{\boldsymbol{\Omega}} = \boldsymbol{N}^{-1}\boldsymbol{H}^{\top}\boldsymbol{A}\boldsymbol{H}\boldsymbol{N}^{-1} \in [0, 1]^{k\times k}, \tag{A3}$$

and assume that if $\boldsymbol{H}$ is an EEP of $\boldsymbol{\Omega}$ then $\boldsymbol{H}$ will be *approximately* an EEP of $\widehat{\boldsymbol{\Omega}}$, i.e.,

$$\boldsymbol{L}(\widehat{\boldsymbol{\Omega}})\boldsymbol{H} \approx \boldsymbol{H}\boldsymbol{L}^{\pi} \quad \boldsymbol{H} \in \mathcal{H}_{\mathrm{EEP}}^{\Omega}. \tag{A4}$$

Our question now is whether or not this is a reasonable assumption and if we can use the projection error as a measure of how well a partition of $\widehat{\boldsymbol{\Omega}}$ approximates an EEP of $\boldsymbol{\Omega}$. We make this argument below by demonstrating that (as $n \to \infty$) stochastic fluctuations in the realization of the adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{n\times n}$ cannot lead to a quotient graph whose Laplacian eigenvectors have a large projection error.

Since each entry of $\widehat{\boldsymbol{\Omega}}$ correspond to a sum over independent Bernoulli random variables, by the central limit theorem each entry $\widehat{\Omega}_{ij}$ will, for large n, be well approximated by a Gaussian random variable $\mathcal{N}(\mu_{ij}, \sigma_{ij})$, if the number of groups stays bounded, i.e., each group becomes sufficiently large. The empirical mean is an unbiased estimator, so the mean of each of these Gaussians will be given by the corresponding entry of true affinity matrix $\mu_{ij} = \Omega_{ij}$. Similarly, the variance of each entry will be $\sigma_{ij}^2 = \Omega_{ij}(1 - \Omega_{ij})/n_i n_j$, where n_i, n_j are the number of nodes in group i and j, respectively (where $\sum_j n_j = n$). It follows that the spectral properties of $\widehat{\boldsymbol{\Omega}}$ will closely approximate the true affinity matrix $\boldsymbol{\Omega}$. More precisely, it can be shown that the spectral norm $\|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}\|_2$ will be small with high probability (see, e.g., Ref. [78]).

Now, let $\boldsymbol{V}_{\kappa}$ be a matrix containing κ structural eigenvectors (out of k total eigenvectors) of $\boldsymbol{L}(\boldsymbol{\Omega})$, where $\boldsymbol{V}_{\kappa}$ is associated with κ consecutive eigenvalues $\lambda_i, \ldots, \lambda_j$. Here, we have ordered the eigenvalues in ascending order such that $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_k$. The Davis-Kahan theorem [79] implies that the corresponding eigenvectors $\widehat{\boldsymbol{V}}_{\kappa}$ of $\boldsymbol{L}(\widehat{\boldsymbol{\Omega}})$ are indeed close to eigenvectors of the true quotient Laplacian $\boldsymbol{L}(\boldsymbol{\Omega})$ (cf. Eq. 3 of Ref. [80]):

$$\|\widehat{\boldsymbol{V}}_{\kappa}\boldsymbol{O} - \boldsymbol{V}_{\kappa}\|_{\mathrm{F}} \leqslant \frac{\sqrt{8k}\|\boldsymbol{L}(\widehat{\boldsymbol{\Omega}}) - \boldsymbol{L}(\boldsymbol{\Omega})\|_2}{\Delta\lambda}, \tag{A5}$$

where $\boldsymbol{O}$ is a unitary matrix, $\|\cdot\|_2$ is the operator norm and

$$\Delta\lambda = \min(\lambda_i - \lambda_{i-1}, \lambda_{j+1} - \lambda_j) \tag{A6}$$

is the eigenvalue gap associated with the true quotient Laplacian. To make the above formula valid for any set of consecutive eigenvalues we define $\lambda_0 = -\infty$ and $\lambda_{k+1} = +\infty$.

As our estimated partition was such that $\boldsymbol{L}(\widehat{\boldsymbol{\Omega}}) \approx \boldsymbol{L}(\boldsymbol{\Omega})$, it follows that if the eigenvalue decomposition is unique and the eigenvalue gap is thus nonzero, the projection error [Eq. (14)] associated with the estimated structural eigenvectors $\widehat{\boldsymbol{V}}_\kappa$ will be small

$$\begin{aligned}\|\boldsymbol{P_H}\widehat{\boldsymbol{V}}_\kappa\|_{\mathrm{F}} &= \|\boldsymbol{P_H}\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O}\|_{\mathrm{F}}\\ &= \|\boldsymbol{P_H}\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O} - \boldsymbol{P_H}\boldsymbol{V}_\kappa\|_{\mathrm{F}}\\ &= \|\boldsymbol{P_H}(\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O} - \boldsymbol{V}_\kappa)\|_{\mathrm{F}}\\ &= \|\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O} - \boldsymbol{V}_\kappa - \boldsymbol{H}\boldsymbol{H}^\dagger(\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O} - \boldsymbol{V}_\kappa)\|_{\mathrm{F}}\\ &\leqslant \|\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O} - \boldsymbol{V}_\kappa\|_{\mathrm{F}} + \|\boldsymbol{H}\boldsymbol{H}^\dagger\|_{\mathrm{F}}\|\widehat{\boldsymbol{V}}_\kappa\boldsymbol{O} - \boldsymbol{V}_\kappa\|_{\mathrm{F}}\\ &\leqslant (1+\sqrt{k})\frac{\sqrt{8k}\|\boldsymbol{L}(\widehat{\boldsymbol{\Omega}}) - \boldsymbol{L}(\boldsymbol{\Omega})\|_2}{\Delta\lambda},\end{aligned}$$

where the first equality uses the fact that multiplication with a unitary matrix does not change the norm; the second equality comes from the fact that $\boldsymbol{V}_\kappa$ are structural eigenvectors, which means that $\boldsymbol{P_H}\boldsymbol{V}_\kappa = \boldsymbol{0}$ because the projection error is zero; the third equality is a simple rearrangement; the fourth equality follows from using the definition of the projection operator $\boldsymbol{P_H} := \boldsymbol{I} - \boldsymbol{H}\boldsymbol{H}^\dagger$; the first inequality uses the subadditive property of the norm; and the final inequality follows from the Davis-Kahan theorem and the fact that $\|\boldsymbol{H}\boldsymbol{H}^\dagger\|_{\mathrm{F}} = \sqrt{k}$. We can see from the above that $\|\boldsymbol{P_H}\widehat{\boldsymbol{V}}\|_{\mathrm{F}}$ will be small for large enough graphs with large enough groups as $\|\boldsymbol{L}(\widehat{\boldsymbol{\Omega}}) - \boldsymbol{L}(\boldsymbol{\Omega})\|_2 \to 0$, which follows from the fact that the estimated entries $\widehat{\Omega}_{ij} \to \Omega_{ij}$ for large enough group sizes.

APPENDIX B: *k*-MEANS IS THE DUAL OF MINIMIZING PROJECTION ERROR

Let us write out the objective function of k-means in which we take the rows $\boldsymbol{v}_{1\cdot}, \boldsymbol{v}_{2\cdot}, \dots, \boldsymbol{v}_{n\cdot}$ of $\boldsymbol{V}$ as $k \times 1$ vectors representing the elements to be clustered ($\boldsymbol{V}^\top = [\boldsymbol{v}_{1\cdot}, \boldsymbol{v}_{2\cdot}, \dots, \boldsymbol{v}_{n\cdot}]$):

$$\min_{\boldsymbol{H}} \sum_{j=1}^{k}\sum_{i=1}^{n} H_{ij}\|\boldsymbol{v}_{i\cdot} - \boldsymbol{\mu}_j\|^2, \quad \text{with } \boldsymbol{\mu}_j = \frac{1}{n_j}\sum_{i=1}^{n}\boldsymbol{v}_{i\cdot}H_{ij}, \tag{B1}$$

where n_j is the number of points in cluster j. Now observe that we can write $\boldsymbol{\mu}_j$ as:

$$\boldsymbol{\mu}_j = [\boldsymbol{V}^\top\boldsymbol{H}\boldsymbol{N}^{-1}]_{\cdot j}. \tag{B2}$$

Accordingly, we can see that $[\boldsymbol{V}^\top\boldsymbol{H}\boldsymbol{N}^{-1}\boldsymbol{H}^\top] = \boldsymbol{V}^\top\boldsymbol{H}\boldsymbol{H}^\dagger \in \mathbb{R}^{k\times n}$ corresponds to the matrix whose ith column represents the mean of the cluster that node i is assigned to. We can thus rewrite the k-means objective above as

$$\min_{\boldsymbol{H}} \sum_{i=1}^{n} \left\|\boldsymbol{V}_{\cdot i}^\top - [\boldsymbol{V}^\top\boldsymbol{H}\boldsymbol{H}^\dagger]_{\cdot i}\right\|^2. \tag{B3}$$

Using the Frobenius norm we can more compactly write this as

$$\min_{\boldsymbol{H}} \|\boldsymbol{V}^\top - \boldsymbol{V}^\top\boldsymbol{H}\boldsymbol{H}^\dagger\|_{\mathrm{F}}^2 = \min_{\boldsymbol{H}} \|\boldsymbol{V}^\top[\boldsymbol{I} - \boldsymbol{H}\boldsymbol{H}^\dagger]\|_{\mathrm{F}}^2. \tag{B4}$$

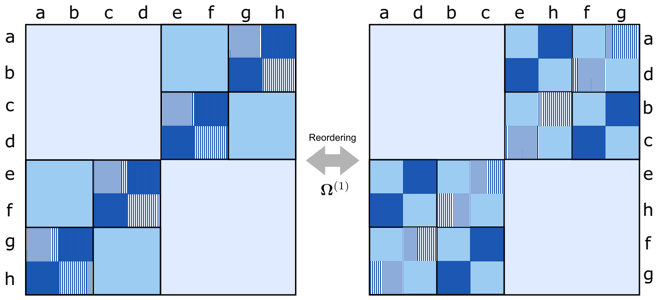

FIG. 14. Nonidentifiability of hierarchical configurations for a disassortative hierarhical network model. We show two (nonidentifiable) possible partitions of the affinity matrix $\boldsymbol{\Omega}^{(1)}$. The left corresponds to the "planted" hierarchical partition, the right to an equivalent partition, that is commensurate with a different hierarchy that preserves the coarsest and finest partition.

Finally, by noting that the Frobenius norm is unchanged by taking the transpose of its arguments we can establish the desired equality to our projection error criterion given in Eq. (15).

The above result shows that there is an striking duality between the problem of finding an EEP with minimal projection error and the k-means problem on the rows of the Laplacian eigenvectors: instead of searching for k partition indicator vectors in an n-dimensional space that minimize the projection error, we can consider a dual problem of finding the k centroids of n points in a k-dimensional space (where the centroids minimize the quantization error defined via the squared 2-norm).

APPENDIX C: EXPECTED PROJECTION ERROR

In this section, we derive the expressions for the expected projection errors given in Eqs. (23) and (24).

1. Accounting for shared subspaces

The derivation of ε_0 in Eq. (22) in the main text assumes that $\boldsymbol{U}$ and $\boldsymbol{H}$ are statistically independent of each other. However, this will not generally be true in the context of a partition indicator matrix $\boldsymbol{H}$ and a set of Laplacian eigenvectors $\boldsymbol{V}_k$. Of particular concern is that $\boldsymbol{1}$ is *always* an eigenvector of a Laplacian and $\boldsymbol{1} \in \operatorname{span}(\boldsymbol{H})$ (because $\boldsymbol{H}\boldsymbol{1}_k = \boldsymbol{1}$). We therefore need to adjust the above argument slightly to incorporate

Algorithm 1. `ClusterWithBetheHessian`.

```
Data: Adjacency matrix $\boldsymbol{A}$
Result: Partition indicator matrix $\boldsymbol{H}$
Compute $\eta = \sqrt{\boldsymbol{1}^\top\boldsymbol{A}\boldsymbol{1}/n}$;
Compute $\boldsymbol{B}_{\pm\eta}$ according to (11);
Compute spectral decompositions $\boldsymbol{B}_\eta = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$,
  $\boldsymbol{B}_{-\eta} = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^\top$;
Compute $k_+ \leftarrow |\{\lambda_i : \lambda_i \le 0\}|$ and $k_- \leftarrow |\{\theta_i : \theta_i \le 0\}|$;
Estimate number of groups $\hat{k} = k_+ + k_-$
Form $\boldsymbol{Q} = [\boldsymbol{V}_{k_+}, \boldsymbol{U}_{k_-}]$, containing the ($k_+$ and $k_-$)
  eigenvectors of $\boldsymbol{B}_{\pm\eta}$ with non-positive eigenvalues;
Run $k$-means clustering on the rows of $\boldsymbol{Q}$:
  $\boldsymbol{H} \leftarrow k\text{-means}(\boldsymbol{Q}^\top, \hat{k})$;
```

Algorithm 2. InferHierarchy.

```
Data: Adjacency matrix $\boldsymbol{A}$,
      Initial Partition $\boldsymbol{H} \in \{0,1\}^{n\times k}$
Result: Sequence of hierarchical partitions
hier_part_list ← $(\boldsymbol{H},)$;
more_levels ← true;
$u \leftarrow 1$;
while more_levels do
    #Compute affinity matrix of current partition $\boldsymbol{H}$;
    $\boldsymbol{\Omega} \leftarrow \boldsymbol{H}^{\dagger(u)} \boldsymbol{A} \left(\boldsymbol{H}^{\dagger(u)}\right)^\top$;          [Eq. (10)]
    # Find (sub)partitions of $\boldsymbol{\Omega}$ and associated proj. errors;
    $(\boldsymbol{H}_1, \dots, \boldsymbol{H}_k), \boldsymbol{\epsilon}$ ← IndentifyPartitionsAndErrors($\boldsymbol{\Omega}$);

    # determine candidates for hier. agglomeration;
    cand_list ← FindRelevantMinima($\boldsymbol{\epsilon}$);
    if cand_list $= \emptyset$ then
        # If no agglomeration candidates exists stop
        more_levels ← false;
    else
        # else keep finest agg. candidate and repeat;
        $\boldsymbol{H}$ ← cand_list.last;
        hier_part_list.append($\boldsymbol{H}$);
    end
end
return hier_part_list
```

Algorithm 3. IdentifyPartitionsAndErrors.

```
Data: Affinity matrix $\boldsymbol{\Omega} \in \mathbb{R}^{k\times k}$,
      number of perturbed samples $z$
Result: Sequence of partitions $(\boldsymbol{H}_1, \dots, \boldsymbol{H}_k)$,
      mean projection errors $\boldsymbol{\epsilon}$
# Compute Laplacian
$\boldsymbol{L} \leftarrow \operatorname{diag}(\boldsymbol{\Omega}\mathbf{1}) - \boldsymbol{\Omega}$;
# Create uniform random walk matrix $\boldsymbol{W}$
$\boldsymbol{W} \leftarrow \boldsymbol{I} - \frac{1}{d_{\max}} \boldsymbol{L}$          [Eq. (16)]
Compute spectral decomposition of $\boldsymbol{W}$:
  $\boldsymbol{W} \leftarrow \boldsymbol{V\Lambda V}^\top$ with $|\lambda_1| = 1 > \dots > |\lambda_k|$;
for $r = 2 : k-1$ do
    # Assemble matrix of first $r$ eigenvectors
    $\boldsymbol{V}_r \leftarrow \boldsymbol{V}_{\cdot,[1:r]}$ ;
    # Find partition into $r$ groups using $k$-means
    $\boldsymbol{H}_r \leftarrow$ k-means($\tilde{\boldsymbol{V}}_r^\top$);
end
# for $r = 1$ and $r = k$ there is only one possibe partition
$\boldsymbol{H}_1 \leftarrow \mathbf{1}$, $\boldsymbol{H}_k \leftarrow \boldsymbol{I}_k$;
for $\zeta = 1 : z$ do
    $\tilde{\boldsymbol{\Omega}} \leftarrow$ perturbation($\boldsymbol{\Omega}$);
    # Compute Laplacian;
    $\tilde{\boldsymbol{L}} \leftarrow \operatorname{diag}(\tilde{\boldsymbol{\Omega}}\mathbf{1}) - \tilde{\boldsymbol{\Omega}}$;
    # Create uniform random walk matrix $\widetilde{\boldsymbol{W}}$;
    $\widetilde{\boldsymbol{W}} \leftarrow \boldsymbol{I} - \frac{1}{d_{\max}} \tilde{\boldsymbol{L}}$          [Eq. (16)]
    # Compute spectral decomposition of $\widetilde{\boldsymbol{W}}$:
    $\widetilde{\boldsymbol{W}} \leftarrow \boldsymbol{U\Theta U}^\top$ with $|\theta_1| = 1 > \dots > |\theta_k|$;
    for $r = 1 : k$ do
        # Assemble matrix of first $r$ eigenvectors:
        $\boldsymbol{U}_r \leftarrow \boldsymbol{U}_{\cdot,[1:r]}$;
        # Compute projection error
        $\boldsymbol{P}_{\boldsymbol{H}_r} \leftarrow [\boldsymbol{I} - \boldsymbol{H}_r \boldsymbol{H}_r^\dagger]$;          [Eq. (13)]
        $\varepsilon^{(\zeta)}(r) \leftarrow \|\boldsymbol{P}_{\boldsymbol{H}_r} \boldsymbol{U}_r\|_{\mathrm{F}}^2$;          [Eq. (14)]
    end
end
# Compute mean projection error vector (over $z$ samples)
$\epsilon_r \leftarrow \frac{1}{z} \sum_\zeta \varepsilon^{(\zeta)}(r)$;
return $(\boldsymbol{H}_1, \dots, \boldsymbol{H}_k), \boldsymbol{\epsilon}$
```

the fact that our eigenvectors will include the eigenvector **1**. Consequently, we are actually looking for the projection of a $(k-1)$-dimensional (rather than k-dimensional) subspace in an $(n-1)-(k-1) = (n-k)$-dimensional space. In other words, we have to account for the fact that we know that there is a one-dimensional EEP present in any connected graph.

In general, if we know there is a κ-dimensional EEP present in the network and we are looking for the projection error associated with a set of $k > \kappa$ eigenvectors we obtain the expected error:

$$\varepsilon_0(k|\kappa) = \frac{(n-k)(k-\kappa)}{n-\kappa} \quad \text{for } \kappa \leqslant k \leqslant n, \tag{C1}$$

which can be derived in a similar way as before by replacing n with $(n-\kappa)$ and k with $(k-\kappa)$ in the above derivation. Note that the changed denominator corresponds to the fact that the effective dimension of the space in which we calculate the projection error shrinks, since we have to exclude the shared subspace from the calculation.

To see this, let us revisit our earlier calculation and consider a random matrix of $\boldsymbol{U} \in \mathbb{R}^{n\times k}$ of k orthonormal vectors of dimension n, i.e., $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$. This time, however, we will assume that we know a κ dimensional subspace $\mathcal{U}_\kappa \subset \mathcal{U} = \operatorname{im}(\boldsymbol{U})$ of the space spanned by the vectors in the matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_k]$. Without loss of generality, we may assume that we know the first κ of these vectors and that these are simply given by standard unit vectors (otherwise we can find an orthogonal transformation $\boldsymbol{Q}_1$ such that $\tilde{\boldsymbol{U}} = \boldsymbol{Q}_1 \boldsymbol{U}$ is of the desired form). Thus we can consider a matrix $\boldsymbol{U}$ of the form:

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{I}_\kappa & \mathbf{0}_{\kappa\times(n-\kappa)} \\ \mathbf{0}_{(n-\kappa)\times\kappa} & \boldsymbol{U}_{\text{sub}} \end{bmatrix}, \tag{C2}$$

where $\boldsymbol{U}_{\text{sub}}$ is an orthogonal matrix of size $(n-\kappa)\times(n-\kappa)$. Following a similar calculation as above, we obtain

$$\begin{aligned} k = \mathbb{E}\big[\|\boldsymbol{U}\|_{\mathrm{F}}^2\big] &= \kappa + \mathbb{E}\left[\sum_{j=1}^{k-\kappa} \sum_{i=\kappa+1}^{n} [U_{\text{sub}}]_{ij}^2\right] \\ &= \kappa + (n-\kappa)(k-\kappa)\mathbb{E}\big[U_{ij}^2\big]. \end{aligned} \tag{C3}$$

Hence, the expected value $\mathbb{E}[[\boldsymbol{U}_{\text{sub}}]_{ij}^2]$ that determines the denominator in the expected error calculation is now $1/(n-\kappa)$ instead of $1/n$.

APPENDIX D: NONIDENTIFIABILITY FOR DISASSORTATIVE HIERARCHIES

As discussed in Sec. V B, when trying to detect a disassortative hierarchical partition planted in a network, such the networks generated for Figs. 10(b) and 10(d), we are confronted with certain nonidentifiability issues. In such cases an algorithm can pick any of the alternative hierarchies that provide an equivalent hierarchical description, instead of the "planted", dissassortative hierarchy. Figure 14 depicts the planted hierarchical affinity matrix (*left*) used in our experiments and one specific reordering of the affinity matrix (*right*)

Algorithm 4. `FindRelevantMinima`.

```
Data: Mean projection error $\boldsymbol{\epsilon}$
Result: Indices of hierarchical partition candidates
# get size of projection error vector
# equal to maximal number of groups
$k = \mathrm{length}(\boldsymbol{\epsilon})$;
cand_list ← $\emptyset$;
# Compute expected error $\boldsymbol{\varepsilon}_0$
$\varepsilon_0(r) \leftarrow \frac{(n-r)(r-1)}{n-1}$;                                   [Eq. (23)]
# Calculate Mean squared logistic error (MSLE)
$\mathrm{MSLE}_0 \leftarrow \min_\sigma \frac{1}{k}\sum_{r=1}^{k}\left(\log\frac{[\epsilon_r+1]}{[\sigma\varepsilon_0(r)+1]}\right)^2$          [Eq. (29)]
for $\kappa = 2 : k-1$ do
    # Get candidate levels and compute expected error
    $\kappa \leftarrow$ cand_list;
    Calculate $\varepsilon_0(r|\kappa)$ according to                         [Eq. (25)]
    # Calculate Mean squared logistic error (MSLE)
    $\mathrm{MSLE}_\kappa \leftarrow \min_\sigma \frac{1}{k}\sum_{r=1}^{k}\left(\log\frac{[\epsilon_r+1]}{[\sigma\varepsilon_0(r|\kappa)+1]}\right)^2$   [Eq. (29)]
    if $MSLE_\kappa < MSLE_0$ then
        cand_list.append($\kappa$);
        $\mathrm{MSLE}_0 \leftarrow \mathrm{MSLE}_\kappa$;
    end
end
return cand_list
```

that meets the nested sEEP requirement. Note how both the finest partition into 8 groups as well as the coarsest partition into 2 groups are preserved by this reordering. However, the two partitions into 4 groups are inconsistent with one another. This degeneracy means that, under small perturbations of the affinity matrix, we recover a mixture of these partitions, which effectively cancels each other out and means that we often do not detect the middle level of the hierarchy.

APPENDIX E: IMPLEMENTATION DETAILS

To detect the initial (finest) partition, we use the Bethe Hessian as described in Algorithm 1.

Algorithm 5. `ClusterWithDegreeCorrectedBetheHessian`.

```
Data: Adjacency Matrix $\boldsymbol{A}$
Result: Partition indicator matrix $\boldsymbol{H}$
Compute $k_+, k_-$ according to Alg. 1;
Estimate number of groups $\hat{k} = k_+ + k_-$;
Estimate the spectral radius $\rho = \frac{\mathbf{1}^\top \boldsymbol{A}\boldsymbol{A}\mathbf{1}/n}{\mathbf{1}^\top \boldsymbol{A}\mathbf{1}/n}$;
for $i = 2 : k_+$ do
    # Find $\eta \in (1, \sqrt{\rho})$ such that $\lambda_i(B_{+\eta}) = 0$
    $\zeta_i \leftarrow \eta$;
    # Compute spectral decomposition $\boldsymbol{B}_{\zeta_i} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$;
    $\boldsymbol{V}_i^+ \leftarrow \boldsymbol{v}_i(B_{+\zeta_i})$;
end
for $i = 1 : k_-$ do
    # Find $\eta \in (1, \sqrt{\rho})$ such that $\lambda_i(B_{-\eta}) = 0$
    $\zeta_i \leftarrow \eta$;
    # Compute spectral decomposition $\boldsymbol{B}_{-\zeta_i} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$;
    $\boldsymbol{V}_i^- \leftarrow \boldsymbol{v}_i(B_{-\zeta_i})$;
end
Form $\boldsymbol{V}^+ = [\boldsymbol{V}_2^+, \boldsymbol{V}_3^+, ..., \boldsymbol{V}_{k_+}^+]$, $\boldsymbol{V}^- = [\boldsymbol{V}_1^-, \boldsymbol{V}_2^-, ..., \boldsymbol{V}_{k_-}^-]$;
Form $\boldsymbol{Q} = [\boldsymbol{V}^+, \boldsymbol{V}^-]$;
# Run k-means clustering on rows of $\boldsymbol{Q}$:
$\mathbf{H} \leftarrow$ k-means($\boldsymbol{Q}^T$, $\hat{k} - 1$);
```

We take the partition thus found as our finest partition and build a hierarchy by agglomeration as described in Algorithm 2.

Here Algorithm 2 makes use of two subroutines. The first one (Algorithm 3) creates the (best) possible subpartitions of the affinity matrix of the currently considered hierarchical level, and computes the associated projection errors. Based on the computed projection errors we then decide whether there is evidence that there is a hierarchical refinement (Algorithm 4) and keep the finest such partition. We then build the affinity matrix of the next hierarchical level and repeat the procedure until no more additional hierarchical levels are found.

[1] H. A. Simon, Proc. Am. Philos. Soc. **106**, 467 (1962).

[2] A. Clauset, C. Moore, and M. E. J. Newman, Hierarchical structure and the prediction of missing links in networks, Nature **453**, 98 (2008).

[3] C. Blundell and Y. W. Teh, Bayesian hierarchical community discovery, in *Advances in Neural Information Processing Systems* (2013), pp. 1601–1609.

[4] T. P. Peixoto, Hierarchical Block Structures and High-Resolution Model Selection in Large Networks, Phys. Rev. X **4**, 011047 (2014).

[5] V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe, Community detection and classification in hierarchical stochastic blockmodels, IEEE Trans. Network Sci. Eng. **4**, 13 (2017).

[6] L. A. Adamic and N. Glance, The political blogosphere and the 2004 US election: divided they blog, in *Proc. of the 3rd Int. Workshop on Link Discovery* (ACM, 2005), pp. 36–43.

[7] C. Cortes, D. Pregibon, and C. Volinsky, Communities of interest, in *Advances in Intelligent Data Analysis*, Lecture Notes in Computer Science Vol. 2189, edited by F. Hoffmann, D. Hand, N. Adams, D. Fisher, and G. Guimaraes (Springer Berlin/Heidelberg, 2001), pp. 105–114.

[8] S. Shai, N. Stanley, C. Granell, D. Taylor, and P. J. Mucha, Case studies in network community detection, in *The Oxford Handbook of Social Networks* (Oxford University Press, Oxford, UK, 2017).

[9] L. S. Haggerty, P.-A. Jachiet, W. P. Hanage, D. A. Fitzpatrick, P. Lopez, M. J. O'Connell, D. Pisani, M. Wilkinson, E. Bapteste, and J. O. McInerney, A pluralistic account of homology: adapting the models to the data, Mol. Biol. Evol. **31**, 501 (2014).

[10] P. Holme, M. Huss, and H. Jeong, Subnetwork hierarchies of biochemical pathways, Bioinformatics **19**, 532 (2003).

[11] R. Guimera and L. A. N. Amaral, Functional cartography of complex metabolic networks, Nature **433**, 895 (2005).

[12] J. Reichardt and S. Bornholdt, Detecting Fuzzy Community Structures in Complex Networks with a Potts Model, Phys. Rev. Lett. **93**, 218701 (2004).

[13] J. Reichardt and S. Bornholdt, Statistical mechanics of community detection, Phys. Rev. E **74**, 016110 (2006).

[14] V. A. Traag, P. Van Dooren, and Y. Nesterov, Narrow scope for resolution-limit-free community detection, Phys. Rev. E **84**, 016114 (2011).

[15] B. Karrer and M. E. J. Newman, Stochastic blockmodels and community structure in networks, Phys. Rev. E **83**, 016107 (2011).

[16] M. E. J. Newman, Equivalence between modularity optimization and maximum likelihood methods for community detection, Phys. Rev. E **94**, 052315 (2016).

[17] E. Mossel, J. Neeman, and A. Sly, Belief propagation, robust reconstruction and optimal recovery of block models, Ann. Appl. Probab. **26**, 2211 (2016).

[18] E. Mossel, J. Neeman, and A. Sly, A proof of the block model threshold conjecture, Combinatorica **38**, 665 (2018).

[19] E. Abbe, A. S. Bandeira, and G. Hall, Exact recovery in the stochastic block model, IEEE Trans. Inf. Theory **62**, 471 (2016).

[20] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications, Phys. Rev. E **84**, 066106 (2011).

[21] L. Peel, D. B. Larremore, and A. Clauset, The ground truth about metadata and community detection in networks, Sci. Adv. **3**, e1602548 (2017).

[22] E. Abbe, Community detection and stochastic block models: Recent developments, J. Mach. Learn. Res. **18**, 1 (2018).

[23] C. Moore, The computer science and physics of community detection: Landscapes, phase transitions, and hardness, arXiv:1702.00467.

[24] B. Corominas-Murtra, J. Goñi, R. V. Solé, and C. Rodríguez-Caso, On the origins of hierarchy in complex networks, Proc. Natl. Acad. Sci. USA **110**, 13316 (2013).

[25] A. Clauset, S. Arbesman, and D. B. Larremore, Systematic inequality and hierarchy in faculty hiring networks, Sci. Adv. **1**, e1400005 (2015).

[26] M. Shrestha, S. V. Scarpino, E. M. Edwards, L. T. Greenberg, and J. D. Horbar, The interhospital transfer network for very low birth weight infants in the united states, EPJ Data Sci. **7**, 27 (2018).

[27] E. Ravasz and A.-L. Barabási, Hierarchical organization in complex networks, Phys. Rev. E **67**, 026112 (2003).

[28] M. Rosvall and C. T. Bergstrom, Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems, PLoS ONE **6**, e18209 (2011).

[29] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, Finding statistically significant communities in networks, PLoS ONE **6**, e18961 (2011).

[30] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. (2008) P10008.

[31] S. White and P. Smyth, A spectral clustering approach to finding communities in graphs, in *Proceedings of the 2005 SIAM International Conference on Data Mining* (SIAM, 2005), pp. 274–285.

[32] M. E. J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA **103**, 8577 (2006).

[33] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, Kronecker graphs: an approach to modeling networks, J. Mach. Learn. Res. **11**, 985 (2010).

[34] P. Zhang and C. Moore, Scalable detection of statistically significant communities and hierarchies, using message passing for modularity, Proc. Natl. Acad. Sci. USA **111**, 18144 (2014).

[35] M. A. Riolo and M. E. J. Newman, Consistency of community structure in complex networks, Phys. Rev. E **101**, 052306 (2020).

[36] T. P. Peixoto, Revealing Consensus and Dissensus between Network Partitions, Phys. Rev. X **11**, 021003 (2021).

[37] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborova, and P. Zhang, Spectral redemption in clustering sparse networks, Proc. Natl. Acad. Sci. USA **110**, 20935 (2013).

[38] A. Saade, F. Krzakala, and L. Zdeborová, Spectral clustering of graphs with the bethe hessian, in *Advances in Neural Information Processing Systems 27* (2014), pp. 406–414.

[39] T. Li, L. Lei, S. Bhattacharyya, K. Van den Berge, P. Sarkar, P. J. Bickel, and E. Levina, Hierarchical community detection by recursive partitioning, J. Am. Stat. Assoc., 1–18 (2020).

[40] L. Lei, X. Li, and X. Lou, Consistency of spectral clustering on hierarchical stochastic block models, arXiv:2004.14531.

[41] S. Balakrishnan, M. Xu, A. Krishnamurthy, and A. Singh, Noise thresholds for spectral clustering, Adv. Neural Inf. Process. Syst. **24**, 954 (2011).

[42] M. W. Mahoney, L. Orecchia, and N. K. Vishnoi, A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally, J. Mach. Learn. Res. **13**, 2339 (2012).

[43] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney, Think locally, act locally: Detection of small, medium-sized, and large communities in large networks, Phys. Rev. E **91**, 012821 (2015).

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

[45] P. W. Holland, K. B. Laskey, and S. Leinhardt, Stochastic blockmodels: First steps, Soc. Networks **5**, 109 (1983).

[46] K. Nowicki and T. A. B. Snijders, Estimation and prediction for stochastic blockstructures, J. Am. Stat. Assoc. **96**, 1077 (2001).

[47] L. Peel and A. Clauset, Detecting change points in the large-scale structure of evolving networks, in *Proceedings of the Twenty-Ninth International Conference on Artificial Intelligence (AAAI)* (2015), pp. 2914–2920.

[48] F. Lorrain and H. C. White, Structural equivalence of individuals in social networks, J. Math. Soc. **1**, 49 (1971).

[49] C. Godsil and G. F. Royle, *Algebraic Graph Theory* (Springer Science & Business Media, 2013), Vol. 207.

[50] SATORU Kudose, Equitable partitions and orbit partitions, Acta Mathematica Sinica, 1–9 (2009).

[51] D. R. White and K. P. Reitz, Graph and semigroup homomorphisms on networks of relations, Social Networks **5**, 193 (1983).

[52] U. Brandes and J. Lerner, Structural similarity in graphs, in *International Symposium on Algorithms and Computation* (Springer, 2004), pp. 184–195.

[53] M. T. Schaub, N. O'Clery, Y. N. Billeh, J.-C. Delvenne, R. Lambiotte, and M. Barahona, Graph partitions and cluster synchronization in networks of oscillators, Chaos: An Interdisciplinary Journal of Nonlinear Science **26**, 094821 (2016).

[54] T. P. Peixoto, Entropy of stochastic blockmodel ensembles, Phys. Rev. E **85**, 056122 (2012).

[55] B. Ball, B. Karrer, and M. E. J. Newman, Efficient and principled method for detecting communities in networks, Phys. Rev. E **84**, 036103 (2011).

[56] P. K. Gopalan and D. M. Blei, Efficient discovery of overlapping communities in massive networks, Proc. Natl. Acad. Sci. USA **110**, 14534 (2013).

[57] P. Zhang, F. Krzakala, J. Reichardt, and L. Zdeborová, Comparative study for inference of hidden classes in stochastic block models, J. Stat. Mech. (2012) P12021.

[58] C. M. Le and E. Levina, Estimating the number of communities in networks by spectral methods, Electron. J. Stat. **16**, 3315 (2022).

[59] L. Dall'Amico, R. Couillet, and N. Tremblay, Revisiting the bethe-hessian: improved community detection in sparse heterogeneous graphs, Adv. Neural Inf. Process. Syst. **32**, 4037 (2019).

[60] L. M. Pecora, F. Sorrentino, A. M. Hagerstrom, T. E. Murphy, and R. Roy, Cluster synchronization and isolated desynchronization in complex networks with symmetries. Nat. Commun. **5**, 4079 (2014).

[61] F. Sorrentino, L. M. Pecora, A. M. Hagerstrom, T. E. Murphy, and R. Roy, Complete characterization of the stability of cluster synchronization in complex dynamical networks, Sci. Adv. **2**, e1501737 (2016).

[62] R. J. Sánchez-García, Exploiting symmetry in network analysis, Commun. Phys. **3**, 87 (2020).

[63] N. O'Clery, Y. Yuan, G.-B. Stan, and M. Barahona, Observability and coarse graining of consensus dynamics through the external equitable partition, Phys. Rev. E **88**, 042805 (2013).

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

[65] I. S. Dhillon, Y. Guan, and B. Kulis, Kernel k-means: spectral clustering and normalized cuts, in *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004), pp. 551–556.

[66] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. **17**, 395 (2007).

[67] R. Olfati-Saber, J. A. Fax, and R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE **95**, 215 (2007).

[68] https://github.com/michaelschaub/HierarchicalCommunityDetection.

[69] N. X. Vinh, J. Epps, and J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, J. Mach. Learn. Res. **11**, 2837 (2010).

[70] A. J. Gates and Y.-Y. Ahn, The impact of random models on clustering similarity, J. Mach. Learn. Res. **18**, 1 (2017).

[71] L. Peel and M. T. Schaub, Detectability of hierarchical communities in networks, arXiv:2009.07525.

[72] A. Dasgupta, J. E. Hopcroft, and F. McSherry, Spectral analysis of random graphs with skewed degree distributions, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2004), pp. 602–610.

[73] R. Mastrandrea, J. Fournet, and A. Barrat, Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys, PLoS ONE **10**, e0136497 (2015).

[74] S. De Deyne, D. J. Navarro, A. Perfors, M. Brysbaert, and G. Storms, The "small world of words" english word association norms for over 12,000 cue words, Behavior Research Methods **51**, 987 (2019).

[75] A. J. Gates, I. B. Wood, W. P. Hetrick, and Y.-Y. Ahn, Element-centric clustering comparison unifies overlaps and hierarchy, Sci. Rep. **9**, 8574 (2019).

[76] A. Lancichinetti, S. Fortunato, and J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. **11**, 033015 (2009).

[77] J. I. Perotti, C. J. Tessone, and G. Caldarelli, Hierarchical mutual information for the comparison of hierarchical community structures in complex networks, Phys. Rev. E **92**, 062825 (2015).

[78] A. S. Bandeira, R. Van Handel *et al.*, Sharp nonasymptotic bounds on the norm of random matrices with independent entries, Ann. Probab. **44**, 2479 (2016).

[79] C. Davis and W. M. Kahan, The rotation of eigenvectors by a perturbation. iii, SIAM J. Numer. Anal. **7**, 1 (1970).

[80] Y. Yu, T. Wang, and R. J. Samworth, A useful variant of the davis–kahan theorem for statisticians, Biometrika **102**, 315 (2015).