

Relating Modularity Maximization and Stochastic Block Models in Multilayer Networks*

A. Roxana Pamfil[†], Sam D. Howison[†], Renaud Lambiotte[†], and Mason A. Porter[‡]

Abstract. Characterizing large-scale organization in networks, including multilayer networks, is one of the most prominent topics in network science and is important for many applications. One type of mesoscale feature is community structure, in which sets of nodes are densely connected internally but sparsely connected to other dense sets of nodes. Two of the most popular approaches for community detection are to maximize an objective function called “modularity” and to perform statistical inference using stochastic block models. Inspired by Newman’s work on monolayer networks [*Phys. Rev. E*, 94 (2016), 052315], we show in multilayer networks that maximizing modularity is equivalent, under certain conditions, to maximizing the posterior probability of community assignments under suitably chosen stochastic block models. We derive versions of this equivalence for various types of multilayer structures, including temporal, multiplex, and multilevel networks. We consider cases in which the key parameters are constant, as well as ones in which they vary across layers; in the latter case, this yields a novel, layer-weighted version of the modularity function. Our results also help address a longstanding difficulty of multilayer modularity-maximization algorithms, which require the specification of two sets of tuning parameters that have been difficult to choose in practice. We show how to perform this parameter selection in a statistically grounded way, and we demonstrate the effectiveness of our approach on both synthetic and empirical networks.

Key words. community detection, modularity maximization, stochastic block models, multilayer networks

AMS subject classifications. 62H30, 91C20, 94C15, 90C35

DOI. 10.1137/18M1231304

1. Introduction. Networks are useful representations of systems of interacting entities, such as cities linked by railways, computers connected on the internet, or neurons interacting through synapses [50]. The simplest kind of network is a graph: a collection of nodes (i.e., vertices) that are linked to each other, in pairwise fashion, by edges. Variants include weighted networks, in which real numbers associated to edges indicate the strength of the associated connections, and directed networks, which one can use to encode asymmetric interactions between entities.

An enormous amount of research is dedicated to studying mesoscale structures in networks. Examples include community structure [23], core–periphery structure [14, 65], and role similarity [66]. In the present paper, we focus on community detection [1, 21, 23, 58, 62, 70].

*Received by the editors December 7, 2018; accepted for publication (in revised form) July 18, 2019; published electronically October 8, 2019.

<https://doi.org/10.1137/18M1231304>

Funding: The first author was funded by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in partnership with dunnhumby.

[†]Mathematical Institute, University of Oxford, Oxford OX2 6GG, United Kingdom (feier@maths.ox.ac.uk, howison@maths.ox.ac.uk, lambiotte@maths.ox.ac.uk).

[‡]Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095 (mason@math.ucla.edu).

There is no consensus definition of what constitutes a community, and the precise definition typically depends on the methodology that one uses to detect communities. In qualitative terms, communities are sets of nodes that are densely connected to each other and sparsely connected to other nodes in a network. Communities can reveal meaningful structure in networks, such as functional pathways in metabolic networks, related pages in the World Wide Web, groups of friends in social networks, and more [50, 62].

Many existing community-detection methods involve the optimization of some quality function, such as modularity [48, 52], Markov stability [20, 41, 42], the “map equation” in INFOMAP [17, 67], and likelihood functions or posterior probabilities from generative network models [37, 51, 56, 59]. Among these approaches, we distinguish two types (see also [25]). The first set of methods relies on the definition of a quality function based on heuristic, information-theoretic, or other arguments. For example, one may construe a partition of a network as “better” if its communities have a “surprising” number of connections compared with some null model (modularity), if they form bottlenecks to random walks (Markov stability), or if they give a good compression of a network from an information-theoretic perspective (the map equation). The second set of methods encompasses generative models of networks with mesoscale structures (in particular, community structure), which one then fits to data using some form of statistical inference. These models take many forms, and it is particularly popular to use various types of stochastic block models (SBMs). A recent paper by Newman [49] showed that, under certain conditions, maximizing modularity is equivalent to maximizing the likelihood of a simplified SBM known as the “planted-partition model” (PPM) [13]. This result plays an important role in our paper, so we discuss it in detail in section 2.

For many applications, it is important to move beyond ordinary graphs (i.e., “monolayer networks”) to consider more complicated network structures, such as by studying a collection of interrelated networks. Examples include temporal networks [6, 29, 60], in which edges and/or nodes change in time, and multiplex networks (a generalization of edge-colored multigraphs), in which edges correspond to different types of interactions [47, 77]. We consider these examples (and others) using the formalism of multilayer networks [9, 16, 38, 61].

Detecting communities in multilayer networks is an active area of research. There are now several methods, with a variety of approaches [15, 17, 35, 47, 57, 72], and they have been employed on numerous applications. For temporal networks, multilayer community detection has been applied to the study of brain networks [7], financial correlations [6], scientific citations [30], biological contagions [69], and more. For multiplex networks, examples of previous studies include investigations of genes with different types of interactions [12], social support networks in Indian villages [15], and microbial interactions at different sites in the human body [72].

Given the popularity of modularity maximization for detecting communities in monolayer networks, it is no surprise that researchers have extended it to multilayer settings. Mucha et al. [47] generalized previous studies that described communities as bottlenecks to certain random-walker dynamics [40, 41, 42] to derive a multilayer modularity function, which one then seeks to optimize. Multilayer modularity requires the specification of two sets of parameters, and selecting appropriate values is crucial for uncovering meaningful community structure. In the simplest case, there are exactly two parameters: the resolution γ , which influences the sizes of detected communities; and the interlayer coupling ω , which influences

how much detected communities change across layers.¹ Even in this simplified setting, determining appropriate values for γ and ω is an open problem, and interlayer-coupling parameters are especially poorly understood. In the absence of other guidance, researchers typically run modularity maximization for different values of γ and ω and then use heuristic criteria (e.g., looking for a range of values that yield qualitatively similar community structures) to select a subset of values for in-depth investigation. Some researchers have attempted more systematic approaches, and we briefly indicate two of them. Bassett et al. [4] chose values of γ and ω that maximize the difference between the modularity Q of a network and the modularity Q^{null} of an associated network without community structure. This procedure reveals “cohesive regions” in the (γ, ω) plane in which a given network is significantly more modular than a random network with similar characteristics. Weir et al. [78] proposed a different approach. Their method takes as input a set of partitions of a network into communities and then discards all “nonadmissible” partitions, which they construed as those that are dominated by others for any choice of γ and ω . Both of these methods require a large number of calculations to adequately sample the space of multilayer partitions (by performing modularity maximization for many combinations of γ and ω values).

SBMs and other generative models of networks with planted mesoscale structures have also been extended to multilayer settings. However, one shortcoming of existing SBMs is that they restrict how mesoscale structure can vary across layers, and many multilayer SBMs require that communities be identical across all layers [57, 60, 73, 81]. Several authors have relaxed this assumption in various ways. Ghasemian et al. [26] described a Markov model for temporal multilayer networks in which nodes move to new communities between successive layers with probabilities given by a transition matrix. Because their goal was to study detectability thresholds [46] in these networks, they made several simplifications that reduce the utility of their model for inferring mesoscale structures in many empirical networks. (For example, their model produces networks in which all nodes have the same expected degree.) The degree-corrected, mixed-membership generative model of De Bacco et al. [15] assumed that block structure is identical across layers, but these blocks can induce different types of structures in different layers (e.g., assortative, disassortative, core-periphery, and so on). Stanley et al. [72] proposed a model that allows different layers to have different communities. However, their algorithm involves a clustering step that assigns layers to “strata,” with each stratum described by a single SBM. Consequently, their model does not include an explicit mechanism for generating different communities in different layers. Vallès-Català et al. [76] proposed a model in which different layers can have different communities, but they assumed that one observes only an aggregate monolayer network. Consequently, one cannot fit this model to multilayer network data. Bazzi et al. [5] proposed a general probabilistic model of multilayer networks in which mesoscale structure can vary arbitrarily across layers. One of the potential uses of their model is in community detection, although an inference algorithm that works in this general setting has not yet been developed. Nevertheless, the ideas of Bazzi et al. play an important role in the derivation of our results in section 3 and in some of the numerical

¹Recently, Vaiana and Muldoon [74] examined the interplay between these two parameters. They demonstrated the existence of a multilayer “resolution limit,” in that modularity maximization is unable to detect community merges across layers for some values of γ and ω .

experiments in section SM1 of our supplementary materials.

The primary goal of our paper is to demonstrate that the problem of maximizing modularity in multilayer networks is related to the problem of fitting particular multilayer SBMs to network data. We achieve this by using the formulation of multilayer modularity from [47] as a starting point and seeking generative models that recover this objective function. Our work sheds light on some of the assumptions that are implicit in multilayer modularity maximization. It also provides a general method for devising new modularity functions, including a layer-weighted modularity that is appropriate for multilayer networks with statistical properties that may be heterogeneous across layers. A second contribution of our paper is an iterative algorithm for estimating resolution and interlayer-coupling parameters when performing multilayer modularity maximization. This algorithm enables one to detect communities that can vary in arbitrary ways across layers—something that existing SBM approaches are unable to do. Throughout the paper, we work with unweighted networks; we briefly discuss possible extensions of our results to weighted networks in section 6.

The rest of our paper is organized as follows. In section 2, we review Newman’s work in [49] that established a connection between modularity maximization and inference in a degree-corrected PPM for monolayer networks. In section 3, we show that similar results hold for certain types of multilayer networks (including temporal and multiplex networks). We then use these results to determine appropriate values for the resolution and coupling parameters in multilayer modularity. In section 4.1, we describe an iterative algorithm for estimating these parameters. In section 4.2, we demonstrate its performance on synthetic networks. We then apply our methods to a multiplex social network in section 5, and we conclude with some ideas for future work in section 6. In the supplementary materials, we include derivations of our equivalence result for directed and multilevel networks, additional details about our iterative parameter-estimation algorithm, and further numerical tests on synthetic networks.

2. Newman’s result for monolayer networks. Modularity maximization and inference in SBMs are two of the most widely used methods for identifying mesoscale structures in networks [23]. In [49], Newman uncovered a connection between these two methods. We briefly summarize his argument in this section before considering the more general multilayer setting from section 3 onwards.

Modularity is an objective function that measures the “quality” of a partition of a network into disjoint sets of nodes known as “communities.” Larger values of modularity correspond to partitions in which more edges fall within groups than expected by chance, as quantified by a null model. Let g_i denote the (unknown) community label of node i . One seeks to determine a vector \mathbf{g} of community assignments that maximizes modularity. In practice, because this is an NP-hard problem [10], one generically finds only local optima of this objective function.

For undirected monolayer networks, the version of modularity that is used most often employs the Newman–Girvan null model [52] (which is closely related to configuration models [24]). In this case, the modularity function $Q(\mathbf{g})$ compares the entries of a given network’s adjacency matrix \mathbf{A} with the expectation of the entries after uniformly randomly rewiring the

edges such that the expected degrees of the nodes are preserved:

$$(2.1) \quad Q(\mathbf{g}) = \frac{1}{2m} \sum_{i,j=1}^N \left(A_{ij} - \gamma \frac{d_i d_j}{2m} \right) \delta(g_i, g_j),$$

where N is the number of nodes in the network, d_i is the degree of node i (and d_j is the degree of node j), and m is the number of edges. The Kronecker delta $\delta(\cdot, \cdot)$ is equal to 1 whenever its arguments are the same, and it is equal to 0 when they are different. Therefore, the only terms that contribute to the sum in (2.1) are ones in which nodes i and j are in the same community. The resolution parameter γ , which is equal to 1 in the traditional formulation of modularity [48, 52], was introduced in [63] and mitigates (but does not remove) resolution issues, wherein communities below a certain size are undetectable using the original definition of modularity [22, 28]. By tuning this resolution parameter, one can detect coarser or finer community structure in a network.

In [49], Newman compared the modularity function (2.1) with likelihood functions that arise from statistical network models. Following his derivation, consider a degree-corrected SBM that is parametrized by the matrix $\Theta = (\theta_{rs})_{1 \leq r, s \leq K}$, where K is the number of communities. In SBMs, one typically needs to assume that K is given (or learned through some other means, such as Bayesian inference [64]), but knowing the correct value of K is not necessary to establish the results of this section. The parameters θ_{rs} indicate the propensities with which nodes in community r connect to nodes in community s . Suppose that the number of edges between two nodes, i and j , follows a Poisson distribution with mean $d_i d_j \theta_{g_i g_j} / (2m)$. The log-likelihood of the data (i.e., the adjacency matrix) \mathbf{A} , given parameters Θ and \mathbf{g} , is then

$$(2.2) \quad \log \mathbb{P}(\mathbf{A} | \Theta, \mathbf{g}) = \frac{1}{2} \sum_{i,j=1}^N \left[A_{ij} \log \frac{d_i d_j}{2m} + A_{ij} \log \theta_{g_i g_j} - \log(A_{ij}!) - \frac{d_i d_j}{2m} \theta_{g_i g_j} \right].$$

The factor of $1/2$ is necessary to avoid double-counting of edges for undirected networks. For directed networks, one can omit this factor from the log-likelihood, because the entries A_{ij} and A_{ji} are independent random variables. (For multilayer networks, this difference in scaling becomes significant for reasons that we discuss in section SM1 of our supplementary materials.)

If we know the parameters Θ , our goal is to maximize the log-likelihood (2.2) with respect to the partition vector \mathbf{g} . We can ignore any terms in the right-hand side that do not depend on \mathbf{g} and write

$$(2.3) \quad \log \mathbb{P}(\mathbf{A} | \Theta, \mathbf{g}) = \frac{1}{2} \sum_{i,j=1}^N \left(A_{ij} \log \theta_{g_i g_j} - \frac{d_i d_j}{2m} \theta_{g_i g_j} \right) + (\text{const.}).$$

Following [49], we consider a type of SBM known as a “planted-partition model” (PPM) [13]. In this model, the parameters θ_{rs} take one of two values: θ_{in} for $r = s$ and θ_{out} for $r \neq s$. When $\theta_{\text{in}} > \theta_{\text{out}}$, the PPM generates networks with assortative communities, for which the density of edges is higher within sets of nodes than it is between sets. We may now write

$$(2.4) \quad \begin{aligned} \theta_{g_i g_j} &= (\theta_{\text{in}} - \theta_{\text{out}}) \delta(g_i, g_j) + \theta_{\text{out}}, \\ \log \theta_{g_i g_j} &= (\log \theta_{\text{in}} - \log \theta_{\text{out}}) \delta(g_i, g_j) + \log \theta_{\text{out}}. \end{aligned}$$

Substituting (2.4) into (2.3) yields

$$\begin{aligned} \log \mathbb{P}(\mathbf{A}|\boldsymbol{\Theta}, \mathbf{g}) &= \frac{1}{2} (\log \theta_{\text{in}} - \log \theta_{\text{out}}) \sum_{i,j=1}^N \left(A_{ij} - \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\log \theta_{\text{in}} - \log \theta_{\text{out}}} \frac{d_i d_j}{2m} \right) \delta(g_i, g_j) \\ (2.5) \quad &+ (\text{const.}). \end{aligned}$$

The connection between the modularity in (2.1) and the log-likelihood in (2.5) now becomes clear. Both are functions of \mathbf{g} that take the same form (up to additive and multiplicative constants), provided we set

$$(2.6) \quad \gamma = \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\log \theta_{\text{in}} - \log \theta_{\text{out}}}.$$

In particular, the optimal community assignments \mathbf{g} are the same for both functions; maximizing one is the same as maximizing the other.

The above comparison between modularity maximization and maximum-likelihood estimation (MLE) for the PPM is useful for two reasons. First, it reveals some features of modularity that otherwise are not clear—including the fact that it detects communities that are “statistically equivalent” in the sense that they are characterized by the same parameters θ_{in} and θ_{out} . Second, this argument suggests that choosing the resolution γ as in (2.6) is a reasonable default when detecting communities using modularity maximization. (One can estimate the unknown values θ_{in} and θ_{out} that appear in (2.6) using iterative approaches [49].) This is especially relevant when one is interested in choosing a single resolution value for a network, rather than exploring multiscale community structure [71].

A similar equivalence holds for directed and bipartite networks, and the associated resolution value is the same as the one in (2.6). We expect that establishing an analogous result for weighted networks is considerably more difficult.

3. Multilayer networks. In this section, we derive a connection between multilayer modularity maximization and statistical inference in certain multilayer SBMs. We start with a general discussion of multilayer modularity in section 3.1 and multilayer SBMs in section 3.2. We then show how these formulations are related for temporal networks in section 3.3 and for multiplex networks in section 3.4. In section SM2 of our supplementary materials, we examine a scenario in which nodes are part of a hierarchical “multilevel” structure and thus are not restricted to representing identical entities across layers.

3.1. Multilayer modularity maximization. We now introduce some terminology and notation. We study “fully interconnected” multilayer networks [38], in which each node is present in all layers, although the edges can differ from layer to layer. (In situations where this is not the case, one can “pad” a network with additional nodes that are not adjacent to any other nodes in their layer.) Let T denote the number of layers, and let N denote the number of nodes in each layer. We write \mathbf{A}^t for the $N \times N$ adjacency matrix of the network in layer $t \in \{1, \dots, T\}$. We assume for simplicity that the intralayer networks are undirected, and we note that a similar derivation works for directed networks. (In fact, the example in section 5 involves directed networks.) In addition to these intralayer edges, for any two distinct layers

$s, t \in \{1, \dots, T\}$, the $N \times N$ matrix $\mathbf{\Omega}^{st} = (\omega_{ij}^{st})_{1 \leq i, j \leq N}$ specifies the weights of interlayer connections between nodes in those layers. It is convenient to assume for simplicity that these interlayer couplings are “diagonal” [38], so they only connect copies of the same node in different layers. In this case, the matrices $\mathbf{\Omega}^{st}$ are themselves diagonal, and we use $\omega_i^{st} \equiv \omega_{ii}^{st}$ to denote the i th diagonal entry.

Detecting communities in a multilayer network amounts to assigning each node-layer pair (i, t) to a community, which we label g_i^t . For simplicity, and unless we specify otherwise, we use the term “node” to refer to such a node-layer tuple. (These are called “state nodes” in some papers, such as [5, 68].) On occasion, it is useful to consider all node-layer pairs that correspond to the same entity. We use the term “physical node” to refer to this common entity, following terminology from [68] that was later adapted for multilayer networks [5, 17].

Analogously to the monolayer case, multilayer modularity is a function that measures the quality of a multilayer partition \mathbf{g} of a set of nodes into communities, relative to some null model. Using the standard Newman–Girvan null model within each layer [48, 52], Mucha et al. [47] derived an expression for multilayer modularity that is equivalent to

$$\begin{aligned} Q(\mathbf{g}) &= \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t - \gamma_t \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + \sum_{t=1}^T \sum_{s \neq t} \sum_{i=1}^N \omega_i^{st} \delta(g_i^s, g_i^t) \\ (3.1) \quad &= Q_{\text{intra}}(\mathbf{g}) + Q_{\text{inter}}(\mathbf{g}), \end{aligned}$$

where d_i^t is the degree of node i in layer t , the quantity m_t is the number of edges in layer t , and g_i^t is the community label of node i in layer t (and g_j^t is defined analogously). The layer-specific resolution parameters γ_t control the importance of the null-model network relative to the observed network. The node-specific and layer-specific coupling parameters ω_i^{st} influence the extent to which one assigns instances of the same physical node in different layers to the same community. In practice, it is common to assume, as we do in sections 3.3 and 3.4, that some or all of these interlayer-coupling parameters are equal [38]. The multilayer modularity from (3.1) is the sum of intralayer terms $Q_{\text{intra}}(\mathbf{g})$ (which add the monolayer modularity values that are contributed by each layer) and interlayer terms $Q_{\text{inter}}(\mathbf{g})$, which depend on the multilayer partition \mathbf{g} and on the coupling between the layers. One can perform community detection by finding multilayer partitions \mathbf{g} that maximize (3.1).

It is helpful to introduce an alternative formulation of the modularity function in (3.1). A convenient representation of multilayer networks is to use a *supra-adjacency matrix* [27, 38], which has the following block structure:

$$(3.2) \quad \mathcal{A} = \begin{bmatrix} \mathbf{A}^1 & \mathbf{\Omega}^{12} & \dots & \mathbf{\Omega}^{1T} \\ \mathbf{\Omega}^{21} & \mathbf{A}^2 & \dots & \mathbf{\Omega}^{2T} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Omega}^{T1} & \mathbf{\Omega}^{T2} & \dots & \mathbf{A}^T \end{bmatrix}.$$

For each layer t , we also define a *modularity matrix* \mathbf{B}^t , with entries

$$(3.3) \quad B_{ij}^t = A_{ij}^t - \gamma_t \frac{d_i^t d_j^t}{2m_t}.$$

This results in a multilayer modularity matrix \mathbf{B} with diagonal blocks \mathbf{B}^t and off-diagonal blocks $\mathbf{\Omega}^{st}$, analogously to the supra-adjacency matrix \mathbf{A} . Introducing global indexes for the nodes, $i' = i + (t - 1)N$ (so that nodes in the first layer are labeled $1, \dots, N$, nodes in the second layer are labeled $N + 1, \dots, 2N$, and so on), the modularity function in (3.1) is

$$(3.4) \quad Q(\mathbf{g}) = \sum_{i', j'=1}^{NT} B_{i'j'} \delta(g_{i'}, g_{j'}).$$

This objective function is the one that is maximized in the software known as GENLOUVAIN [34], a multilayer generalization of the popular Louvain method for monolayer modularity maximization [8]. Having an algorithm that is able to perform community detection at this level of generality is important for some of the models that we consider in section 3.3 and in the supplementary materials. Such a general algorithm also makes it possible to optimize new versions of modularity that one can derive using the methods in the present paper; we provide some suggestions in this direction in section 6.

3.2. Multilayer SBMs. We now describe a generative model whose posterior probability recovers the modularity function in (3.1). Many different types of multilayer SBMs have been studied [5, 15, 26, 57, 60, 72, 73, 76, 81], and there are many more that one can propose, so we start by pointing out some requirements for such models to be suitable for our purpose.

The first key requirement is that the statistical model has a posterior probability over multilayer partitions \mathbf{g} that one can write in closed form; otherwise, we are unable to make a meaningful comparison with multilayer modularity. This rules out the general benchmark model in [5], which uses Monte Carlo methods to sample from the distribution of multilayer partitions. The second requirement is that the model should allow arbitrary variations in community structure across layers. This is important, because the modularity function in (3.1) also allows this type of generality. This requirement rules out many existing multilayer SBMs [57, 60, 73, 81]. Finally, we demand that any interdependencies between layers are induced solely through the multilayer partition \mathbf{g} and the coupling between nodes in different layers, rather than through intralayer edges. This rules out, for example, the model proposed in [3], as it includes an “edge persistence” parameter that favors intralayer edges remaining active over multiple temporal layers. Note that the modularity function in (3.1) also encapsulates all interlayer dependencies in a separate term $Q_{\text{inter}}(\mathbf{g})$, which does not involve any of the intralayer adjacency matrices \mathbf{A}^t .

With these requirements in mind, we are ready to describe the building blocks of a multilayer SBM that recovers the modularity function in (3.1). In the process, we have to place further restrictions on the generative model; this is unsurprising, because SBMs allow more general mesoscale structures than the type of communities that are detectable through modularity maximization. (In the monolayer setting from section 2, one such restriction is that the SBM only has two edge probabilities, θ_{in} and θ_{out} .)

We assume that edges are placed independently in each layer, conditioned on the multilayer partition \mathbf{g} . In particular, any coupling between nodes in different layers (through interlayer edges or otherwise) has no direct influence on the placement of edges within each layer. This is in line with many existing multilayer SBMs, including those in [5, 26], which are the existing

models that most closely resemble the ones that we propose in the present paper. Using a degree-corrected SBM with a set of parameters Θ^t for each layer, the conditional log-likelihood of observing a network specified by \mathcal{A} is

$$(3.5) \quad \log \mathbb{P}(\mathcal{A}|\mathbf{g}, \Theta) = \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t \log \theta_{g_i^t g_j^t}^t - \frac{d_i^t d_j^t}{2m_t} \theta_{g_i^t g_j^t}^t \right) + (\text{const.}).$$

(For reasons that we explain in section SM1 of our supplementary materials, we omit the factor of 1/2 in front of the double sum.) By taking the same approach as in section 2, we can relate each term in (3.5) to the corresponding intralayer modularity term from (3.1). Assuming that the SBM in each layer takes the form of a PPM with two parameters, θ_{in}^t and θ_{out}^t , the log-likelihood becomes

$$(3.6) \quad \log \mathbb{P}(\mathcal{A}|\mathbf{g}, \theta_{\text{in}}^t, \theta_{\text{out}}^t) = \sum_{t=1}^T (\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t) \sum_{i,j=1}^N \left(A_{ij}^t - \gamma_t \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + (\text{const.}),$$

where

$$(3.7) \quad \gamma_t = \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t}.$$

For now, we assume that the SBM parameters, θ_{in}^t and θ_{out}^t , are the same for all layers. (We discuss the more general setting, in which these parameters can differ across layers, in section 3.3.) In this case, the log-likelihood from (3.6) becomes

$$(3.8) \quad \log \mathbb{P}(\mathcal{A}|\mathbf{g}, \theta_{\text{in}}, \theta_{\text{out}}) = (\log \theta_{\text{in}} - \log \theta_{\text{out}}) \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t - \gamma \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + (\text{const.}),$$

where

$$(3.9) \quad \gamma = \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\log \theta_{\text{in}} - \log \theta_{\text{out}}}.$$

The expression in (3.8) now matches the term $Q_{\text{intra}}(\mathbf{g})$ from (3.1) (up to scaling by an expression that does not depend on \mathbf{g}). The “optimal” resolution value from (3.9) is, perhaps not surprisingly, the same as the one in (2.6) that was derived for the monolayer setting [49]. Note additionally that this derivation holds irrespective of the interdependency structure between the layers and whether a multilayer network is temporal, multiplex, or something else.

We have now accounted for the intralayer modularity terms in (3.1). These terms, as well as the log-likelihood in (3.8), lack any form of coupling between the community assignments \mathbf{g}^t for different layers. Without such coupling, maximizing either multilayer modularity or the SBM log-likelihood with respect to \mathbf{g} is no different from performing the optimization separately for each layer. We introduce layer interdependencies in our generative model through a prior distribution on \mathbf{g} . This is a prior distribution, because it encodes our belief of what community assignments are probable before observing the network data \mathcal{A} . As one might expect,

the form of this prior depends markedly on the specific coupling between layers. Therefore, in our subsequent discussions, we can no longer consider the most general representations of multilayer networks. We thus focus on specific types of multilayer structure; we examine temporal networks in section 3.3, multiplex networks in section 3.4, and multilevel networks in section SM2 of our supplementary materials.

3.3. Temporal networks. In a multilayer representation of a temporal network, each layer encodes interactions that occur in some period of time. Although there are exceptions (e.g., see [75]), it is common to assume that interlayer edges in these temporal networks are “diagonal” (they only connect copies of the same node in different layers), “layer-coupled” (their weight depends only on the layers, rather than on the nodes), and “ordinal” (they only connect successive layers) [38]. We also take these interlayer edges to be directed, indicating that information flows along the arrow of time. Let ω_t denote the weight of interlayer edges from layer $t - 1$ to layer t , where $t \in \{2, \dots, T\}$. The supra-adjacency matrix \mathcal{A} then has the following structure:

$$(3.10) \quad \mathcal{A} = \begin{bmatrix} \mathbf{A}^1 & \omega_2 \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^2 & \omega_3 \mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \omega_T \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}^T \end{bmatrix}.$$

It is common to assume for simplicity that the parameters $\omega_2, \dots, \omega_T$ are all equal to some value ω ; this is known as *uniform coupling* [6, 47]. In this case, the expression (3.1) for multilayer modularity becomes

$$(3.11) \quad Q(\mathbf{g}) = \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t - \gamma \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + \omega \sum_{t=2}^T \sum_{i=1}^N \delta(g_i^{t-1}, g_i^t).$$

We also assume for now that the resolution parameter γ is the same for all layers; we will relax this assumption later in this section. After specifying values for γ and ω , modularity is a function of the community labels \mathbf{g} , where g_i^t gives the community assignment of node i in layer t .

We relate the vectors \mathbf{g}^t of community assignments for different layers using a prior probability $\mathbb{P}(\mathbf{g})$. Following previous approaches for sampling multilayer partitions [5, 26], assume that the community membership of node i in layer t is copied from the previous layer with probability p and is sampled from some specified null distribution \mathbb{P}_0 with probability $1 - p$. Because of the temporal structure, we sample labels in the first layer from the null distribution, rather than copying them from some other layer. The probability of generating a particular multilayer partition \mathbf{g} is then

$$\mathbb{P}(\mathbf{g}) = \mathbb{P}(\mathbf{g}^1) \prod_{t=2}^T \mathbb{P}(\mathbf{g}^t | \mathbf{g}^{t-1}) = \prod_{i=1}^N \mathbb{P}_0(g_i^1) \prod_{t=2}^T \left\{ \prod_{i=1}^N [p \delta(g_i^{t-1}, g_i^t) + (1 - p) \mathbb{P}_0(g_i^t)] \right\}.$$

Taking logarithms and rearranging yields

$$\begin{aligned}
 \log \mathbb{P}(\mathbf{g}) &= \sum_{i=1}^N \log \mathbb{P}_0(g_i^1) + \sum_{t=2}^T \sum_{i=1}^N \log \left[1 + \frac{p \delta(g_i^{t-1}, g_i^t)}{(1-p) \mathbb{P}_0(g_i^t)} \right] + \sum_{t=2}^T \sum_{i=1}^N \log [(1-p) \mathbb{P}_0(g_i^t)] \\
 (3.12) \quad &= \sum_{i=1}^N \log \mathbb{P}_0(g_i^1) + \sum_{t=2}^T \sum_{i=1}^N \log \left[1 + \frac{p}{(1-p) \mathbb{P}_0(g_i^t)} \right] \delta(g_i^{t-1}, g_i^t) \\
 &\quad + \sum_{t=2}^T \sum_{i=1}^N \log [(1-p) \mathbb{P}_0(g_i^t)].
 \end{aligned}$$

The second equality follows from noting that any logarithmic term in the second sum is 0 whenever $\delta(g_i^{t-1}, g_i^t) = 0$. The ability to move the Kronecker delta outside the logarithmic expression is the key technical trick that allows us to relate this prior probability to the interlayer modularity terms from (3.1). This is also the crucial step that makes certain cases, such as multiplex networks (see section 3.4), more difficult to analyze.

Comparing the expression in (3.12) with the interlayer modularity terms in (3.11), we start to see a resemblance: both involve weighted sums over nodes i and layers t of Kronecker deltas of the form $\delta(g_i^{t-1}, g_i^t)$. To obtain an exact match, we have to make further assumptions. Suppose that the null distribution \mathbb{P}_0 is uniform over community assignments. Specifically, if there are K communities (possibly including empty ones), let $\mathbb{P}_0(g_i^t) = 1/K$ for any node i and layer t , such that each node is equally likely to be assigned to each of the communities. The first and third terms in (3.12) become constants, and the only term that depends on \mathbf{g} is the second one. From now on, we also write $\log \mathbb{P}(\mathbf{g}|p, K)$ for the log-prior to emphasize its dependence on the parameters p and K .

We now use Bayes' rule to combine the intralayer and interlayer terms that arise in our proposed multilayer SBM into a single posterior probability:

$$\begin{aligned}
 \log \mathbb{P}(\mathbf{g}|\mathcal{A}, \theta_{\text{in}}, \theta_{\text{out}}, p, K) &= \log \left(\frac{\mathbb{P}(\mathcal{A}, \mathbf{g}|\theta_{\text{in}}, \theta_{\text{out}}, p, K)}{\mathbb{P}(\mathcal{A}|\theta_{\text{in}}, \theta_{\text{out}}, p, K)} \right) \\
 &= \log \left(\frac{\mathbb{P}(\mathcal{A}|\mathbf{g}, \theta_{\text{in}}, \theta_{\text{out}}) \mathbb{P}(\mathbf{g}|p, K)}{\mathbb{P}(\mathcal{A}|\theta_{\text{in}}, \theta_{\text{out}}, p, K)} \right) \\
 (3.13) \quad &= \log \mathbb{P}(\mathcal{A}|\mathbf{g}, \theta_{\text{in}}, \theta_{\text{out}}) + \log \mathbb{P}(\mathbf{g}|p, K) - \log \mathbb{P}(\mathcal{A}|\theta_{\text{in}}, \theta_{\text{out}}, p, K).
 \end{aligned}$$

The first two terms of (3.13) correspond, respectively, to the log-likelihood from (3.8) and the log-prior from (3.12). The third term is difficult to compute in practice, but it does not depend on \mathbf{g} , so we can treat it as a constant for our purposes. We can also ignore multiplicative constants to obtain

$$\begin{aligned}
 \log \mathbb{P}(\mathbf{g}|\mathcal{A}, \theta_{\text{in}}, \theta_{\text{out}}, p, K) &\propto \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t - \gamma \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) \\
 (3.14) \quad &\quad + \sum_{t=2}^T \sum_{i=1}^N \frac{\log \left(1 + \frac{p}{1-p} K \right)}{\log \theta_{\text{in}} - \log \theta_{\text{out}}} \delta(g_i^{t-1}, g_i^t) + (\text{const.}).
 \end{aligned}$$

By maximizing the right-hand side of (3.14) with respect to \mathbf{g} , we find the mode of the posterior distribution over multilayer partitions, given the multilayer network \mathcal{A} and model parameter values θ_{in} , θ_{out} , p , and K . We now ensure that maximizing multilayer modularity (3.11) is equivalent to maximizing the posterior probability (3.14) by choosing

$$(3.15) \quad \gamma = \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\log \theta_{\text{in}} - \log \theta_{\text{out}}},$$

$$(3.16) \quad \omega = \frac{1}{\log \theta_{\text{in}} - \log \theta_{\text{out}}} \log \left(1 + \frac{p}{1-p} K \right).$$

It is useful to point out some limiting cases for the interlayer-coupling parameter ω . When $p = 0$, which entails that there is no copying of community assignments from one layer to the next, (3.16) gives $\omega = 0$. This implies that community detection proceeds independently for each layer. When $p \rightarrow 1$, we obtain $\omega \rightarrow \infty$, which ensures that maximizing modularity recovers the same communities in each layer, as it should. Perhaps less obvious is the dependence on θ_{in} and θ_{out} . Consider the ratio $\epsilon = \theta_{\text{out}}/\theta_{\text{in}}$, which determines the strength of the planted community structure. According to (3.16), a weaker community structure in a network (i.e., a larger value of ϵ) should entail a larger value of ω . One explanation is that by putting more emphasis on persistence of communities across layers, we take advantage of correlations between layers to increase the “signal” of underlying community structure, while simultaneously decreasing layer-specific “noise.” Finally, ω increases (weakly) with the number K of communities. For fixed θ_{in} and θ_{out} , it is typically more difficult to detect a larger number of communities than a smaller number [26]. Increasing the amount of coupling improves community-detection results by taking advantage of correlations between layers.

Thus far, we have assumed that all layers have the same resolution parameter γ and are connected by interlayer edges with the same weight ω . This amounts to assuming that the layers are “statistically similar,” in that they can be described by a common set of parameters θ_{in} , θ_{out} , p , and K . For many applications, this assumption is unrealistic, so it is important to investigate what happens when these parameters are layer-dependent.

Proceeding analogously to the layer-independent case, the log-likelihood of observing a network \mathcal{A} conditioned on a multilayer partition \mathbf{g} is

$$(3.17) \quad \begin{aligned} \log \mathbb{P}(\mathcal{A} | \mathbf{g}, \boldsymbol{\theta}_{\text{in}}, \boldsymbol{\theta}_{\text{out}}) &= \sum_{t=1}^T \left[(\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t) \sum_{i,j=1}^N \left(A_{ij}^t - \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t} \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) \right] \\ &\quad + (\text{const.}). \end{aligned}$$

To impose a prior on the multilayer partition \mathbf{g} , let p_t be the probability that a node in layer t copies its community label from layer $t-1$, where $t \in \{2, \dots, T\}$. Recall that \mathbb{P}_0 is the null distribution for community labels when nodes do not copy their community assignments from the previous layer. We now make this distribution layer-dependent, such that \mathbb{P}_0^t is the null distribution for layer t , to allow layers with nonuniform characteristics. As before, suppose that each null distribution \mathbb{P}_0^t is uniform, which entails that $\mathbb{P}_0^t(g_i^t) = 1/K_t$, where K_t is the number of communities (possibly including empty ones) in layer t . Ignoring any terms that

no longer depend on \mathbf{g} , the log-prior is then

$$\begin{aligned} \log \mathbb{P}(\mathbf{g}) &= \sum_{i=1}^N \log [\mathbb{P}_0^1(g_i^1)] + \sum_{t=2}^T \sum_{i=1}^N \log [p_t \delta(g_i^{t-1}, g_i^t) + (1 - p_t) \mathbb{P}_0^t(g_i^t)] \\ (3.18) \quad &= \sum_{t=2}^T \sum_{i=1}^N \log \left(1 + \frac{p_t}{1 - p_t} K_t \right) \delta(g_i^{t-1}, g_i^t) + (\text{const.}). \end{aligned}$$

Combining (3.17) and (3.18) and multiplying by a constant for notational convenience, we see that \mathbf{g} has the posterior probability distribution

$$\begin{aligned} (3.19) \quad \log \mathbb{P}(\mathbf{g} | \mathcal{A}, \boldsymbol{\theta}_{\text{in}}, \boldsymbol{\theta}_{\text{out}}, \mathbf{p}, \mathbf{K}) \\ \propto \sum_{t=1}^T \left[\frac{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t} \sum_{i,j=1}^N \left(A_{ij}^t - \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t} \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) \right] \\ + \sum_{t=2}^T \sum_{i=1}^N \frac{\log \left(1 + \frac{p_t}{1 - p_t} K_t \right)}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t} \delta(g_i^{t-1}, g_i^t) + (\text{const.}), \end{aligned}$$

where $\langle \cdot \rangle_t$ denotes a mean across all layers $t \in \{1, \dots, T\}$.

Guided by (3.19), we define a modularity function for temporal networks with nonuniform parameters using the following formula:

$$(3.20) \quad Q(\mathbf{g}) = \sum_{t=1}^T \beta_t \sum_{i,j=1}^N \left(A_{ij}^t - \gamma_t \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + \sum_{t=2}^T \omega_t \sum_{i=1}^N \delta(g_i^{t-1}, g_i^t).$$

In contrast to previous definitions of multilayer modularity [47], our definition includes a new set of layer weights β_t in addition to the resolution parameters γ_t and interlayer-coupling parameters ω_t . Comparing (3.20) with the log-probability in (3.19), we see that these parameters should take the following values:

$$(3.21) \quad \gamma_t = \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t},$$

$$(3.22) \quad \omega_t = \frac{1}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t} \log \left(1 + \frac{p_t}{1 - p_t} K_t \right),$$

$$(3.23) \quad \beta_t = \frac{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t}.$$

The expressions in (3.21)–(3.23) are consistent with the ones that we obtained for the case of uniform coupling. If the parameters θ_{in} , θ_{out} , p_t , and K_t are layer-independent, expressions (3.21) and (3.22) reduce to (3.15) and (3.16), respectively. Additionally, (3.23) gives $\beta_t = 1$, which is why these weights do not appear explicitly when one considers uniform coupling.

The role of the parameters β_t is to give different weightings to intralayer modularity terms that correspond to different layers. To the best of our knowledge, this is the first

definition of multilayer modularity that explicitly incorporates such layer weightings.² Their values depend on the strength of the community structure in their respective layers. Letting $\epsilon_t = \theta_{\text{out}}^t / \theta_{\text{in}}^t$, we see that $\beta_t \propto \log(1/\epsilon_t)$. This implies that layers with strong community structure (i.e., ϵ_t close to 0) get a larger weight when maximizing multilayer modularity than layers in which the community structure is weak (i.e., ϵ_t close to 1). It is also worth noting what happens when some layers have disassortative structure (i.e., when $\epsilon_t > 1$). The formula (3.23) suggests taking $\beta_t < 0$, pushing the method towards finding multilayer partitions that *minimize* monolayer modularity for these layers. Indeed, it is known that minimizing modularity for monolayer networks identifies partitions with many edges between groups and few edges within groups [48, 49]. The multilayer setting is especially interesting, because it allows (at least in principle) a combination of assortative and disassortative layers [15].

Although the proposed multilayer modularity from (3.22) takes a nonstandard form because of the weights β_t , algorithms such as GENLOUVAIN [34], which are designed for any maximization problem of the form in (3.4), will continue to work. We also note that one can explicitly consider situations in which the interlayer-coupling parameters are uniform and the resolution parameters are layer-dependent, and vice versa.

A key facet of the results of this section is that they give a principled method for choosing heterogeneous interlayer-coupling weights ω_t . Although the original derivation of multilayer modularity in [47] is general enough to include scenarios with nonuniform parameters, most computations that researchers have done in practice have used uniform coupling and resolution values [4, 47, 78]. For most applications, this is a simplifying assumption, rather than a hypothesis that these parameters are in fact the same for all layers. Our work suggests ways to remove this assumption and perform computations when parameters are layer-dependent, which we expect to be relevant for almost all applications.

The derivation from this section holds whenever the graph of dependencies between layers is a tree. In that case, community labels propagate from the root of this tree to the leaves, yielding a prior distribution $\mathbb{P}(\mathbf{g})$ that is similar to the one from (3.12). One example of potential interest for applications is the star graph, in which one central layer influences community structure in all remaining layers.

3.4. Multiplex networks. In a multiplex network, nodes are adjacent to each other via multiple types of edges; each edge type corresponds to a layer in a multilayer network [38, 77]. In this section, we establish a relationship between modularity maximization and certain multilayer SBMs for multiplex networks.

As in section 3.3, we continue to assume that our multilayer networks are fully interconnected, diagonal, and layer-coupled. To construct a multiplex SBM, we follow [5] and let \mathbf{P} denote the matrix of interlayer dependencies, where p_{st} (with $s \neq t$) is the probability that a node in layer t copies its community label from layer s . (We assume that $p_{ss} = 0$.) The matrix \mathbf{P} may or may not be symmetric. We say that the case in which all probabilities p_{st} are equal to the same value p has “uniform coupling.”

²One can write the most general version of multilayer modularity in [47] in the form of (3.20) by rescaling the adjacency matrices \mathbf{A}^t to incorporate the weights β_t . However, this scenario is not one that the authors of [47] (or other researchers) have considered explicitly.

Recall from section 3.2 that the intralayer modularity $Q_{\text{intra}}(\mathbf{g})$ takes the same functional form as the conditional log-likelihood $\mathbb{P}(\mathcal{A}|\mathbf{g}, \boldsymbol{\theta}_{\text{in}}, \boldsymbol{\theta}_{\text{out}})$, regardless of the dependency structure between layers. We therefore focus our attention on devising a prior $\mathbb{P}(\mathbf{g})$ that recovers the interlayer modularity $Q_{\text{inter}}(\mathbf{g})$ for multiplex networks.

In a multiplex network, there is no natural ordering of the layers (in contrast to temporal and multilevel networks). Consider the vector $\mathbf{g}_i = (g_i^1, \dots, g_i^T)$ of community labels for some node i . From the definition of conditional probability, it follows that

$$(3.24) \quad \mathbb{P}(\mathbf{g}_i) = \mathbb{P}(g_i^1) \mathbb{P}(g_i^2 | g_i^1) \times \dots \times \mathbb{P}(g_i^T | g_i^1, \dots, g_i^{T-1}).$$

However, we can also write

$$(3.25) \quad \mathbb{P}(\mathbf{g}_i) = \mathbb{P}(g_i^T) \mathbb{P}(g_i^{T-1} | g_i^T) \times \dots \times \mathbb{P}(g_i^1 | g_i^T, \dots, g_i^2),$$

and we can write a similar expression for any other permutation of the T layers. Of course, all of these products have to evaluate to the same quantity (namely, $\mathbb{P}(\mathbf{g}_i)$). It is difficult to define a probabilistic model for \mathbf{g} that is internally consistent in this way. In particular, a closed-form model (i.e., one in which we can write $\mathbb{P}(\mathbf{g})$ explicitly) that is otherwise similar to the one in [6] does not have these properties.

We take a different approach. Considering each node i independently, we first sample a permutation σ of the layers from the symmetric group S_T with probability $\mathbb{P}(\sigma)$, and we then update community labels in the order indicated by this permutation: first, we sample a community label for i in what is now layer 1; we then sample a label in layer 2, conditioned on the label from layer 1; and so on.³ The permutation σ that indicates the update order can be different for different nodes. The resulting prior is

$$(3.26) \quad \mathbb{P}(\mathbf{g}) = \prod_{i=1}^N \sum_{\sigma \in S_T} \mathbb{P}(\sigma) \mathbb{P}(g_i^{\sigma^{-1}(1)}) \prod_{t=2}^T \mathbb{P}(g_i^{\sigma^{-1}(t)} | g_i^{\sigma^{-1}(t-1)}).$$

Note that we assume that the update process is memoryless, so the community label in layer t (after applying the permutation σ) depends only on the community label in layer $t-1$. This is the same assumption that we (and others [5, 26]) have made for temporal networks.

For simplicity, we write ς_t for $\sigma^{-1}(t)$. As in section 3.3, we use a copying and resampling process to generate community labels. The conditional probabilities in (3.26) are then

$$(3.27) \quad \mathbb{P}(g_i^{\varsigma_t} | g_i^{\varsigma_{t-1}}) = p_{\varsigma_{t-1}\varsigma_t} \delta(g_i^{\varsigma_{t-1}}, g_i^{\varsigma_t}) + (1 - p_{\varsigma_{t-1}\varsigma_t}) \mathbb{P}_0^{\varsigma_t}(g_i^{\varsigma_t}).$$

For the first layer,

$$(3.28) \quad \mathbb{P}(g_i^{\varsigma_1}) = \mathbb{P}_0^{\varsigma_1}(g_i^{\varsigma_1}).$$

With a uniform prior $\mathbb{P}_0^{\varsigma_t}(g_i^{\varsigma_t}) = 1/K_{\varsigma_t}$, we obtain

$$(3.29) \quad \mathbb{P}(\mathbf{g}) = \prod_{i=1}^N \sum_{\sigma \in S_T} \left\{ \mathbb{P}(\sigma) \frac{\prod_{t=2}^T (1 - p_{\varsigma_{t-1}\varsigma_t})}{\prod_{t=1}^T K_{\varsigma_t}} \prod_{t=2}^T \left[1 + \frac{p_{\varsigma_{t-1}\varsigma_t}}{1 - p_{\varsigma_{t-1}\varsigma_t}} K_{\varsigma_t} \delta(g_i^{\varsigma_{t-1}}, g_i^{\varsigma_t}) \right] \right\}.$$

³Alternatively, one can consider sampling over all possible spanning trees with T nodes and assume that the update propagates from the root of each tree to the leaves.

Taking logarithms and treating any terms that do not depend on \mathbf{g} as constant yields

$$\begin{aligned} \log \mathbb{P}(\mathbf{g}) = & \sum_{i=1}^N \log \left\{ \sum_{\sigma \in S_T} \mathbb{P}(\sigma) \prod_{t=2}^T (1 - p_{\varsigma_{t-1}\varsigma_t}) \left[1 + \frac{p_{\varsigma_{t-1}\varsigma_t}}{1 - p_{\varsigma_{t-1}\varsigma_t}} K_{\varsigma_t} \delta(g_i^{\varsigma_{t-1}}, g_i^{\varsigma_t}) \right] \right\} \\ (3.30) \quad & + (\text{const.}). \end{aligned}$$

As we mentioned in section 3.3, being able to take the Kronecker delta outside the logarithmic expression is the key step that allows us to relate this prior probability to the corresponding interlayer modularity terms. However, we cannot do this easily in the present case because of the sum over permutations $\sigma \in S_T$. Instead, we bound the expression from (3.30) using Jensen's inequality. Because the logarithm is a concave function,

$$(3.31) \quad \log \left(\sum_i x_i \right) \geq \sum_i q_i \log \left(\frac{x_i}{q_i} \right)$$

for any $q_i \geq 0$ with $\sum_i q_i = 1$; equality holds for $q_i = x_i / \sum_i x_i$. Applying the inequality (3.31) to (3.30) and ignoring additive terms that do not depend on \mathbf{g} yields

$$(3.32) \quad \log \mathbb{P}(\mathbf{g}) \geq \sum_{i=1}^N \sum_{\sigma \in S_T} q_i^\sigma \sum_{t=2}^T \log \left(1 + \frac{p_{\varsigma_{t-1}\varsigma_t}}{1 - p_{\varsigma_{t-1}\varsigma_t}} K_{\varsigma_t} \right) \delta(g_i^{\varsigma_{t-1}}, g_i^{\varsigma_t}) + (\text{const.}).$$

We want to rearrange the right-hand side of (3.32) into an expression that resembles the interlayer modularity terms in (3.1). To do this, we fix s and t and consider all of the terms in (3.32) that involve $\delta(g_i^s, g_i^t)$. There are $(T-1)!$ permutations σ with $\varsigma_{t'-1} = s$ and $\varsigma_{t'} = t$ for some $t' \in \{2, \dots, T\}$. (There are $T-1$ ways to choose t' , which fixes the values of $\sigma(s)$ and $\sigma(t)$; and there are $(T-2)!$ ways of permuting the remaining layers.) For convenience, let $S_T^{(s,t)}$ denote the subset of S_T that consists of permutations that map layers s and t to two successive layers. The lower bound of the log-prior then becomes

$$(3.33) \quad \log \mathbb{P}(\mathbf{g}) \geq \sum_{i=1}^N \sum_{s \neq t} \sum_{\sigma \in S_T^{(s,t)}} q_i^\sigma \log \left(1 + \frac{p_{st}}{1 - p_{st}} K_t \right) \delta(g_i^s, g_i^t) + (\text{const.}).$$

The fact that we have a lower bound for this log-prior, rather than an equality as in section 3.3, may seem problematic. However, recall that we obtain equality if the quantities q_i^σ take the specific values

$$(3.34) \quad q_i^\sigma = \frac{\mathbb{P}(\sigma) \prod_{t=2}^T (1 - p_{\varsigma_{t-1}\varsigma_t}) \left[1 + \frac{p_{\varsigma_{t-1}\varsigma_t}}{1 - p_{\varsigma_{t-1}\varsigma_t}} K_{\varsigma_t} \delta(g_i^{\varsigma_{t-1}}, g_i^{\varsigma_t}) \right]}{Z_i},$$

where, for each node $i \in \{1, \dots, N\}$, the quantity Z_i is a proportionality constant that ensures that $\sum_{\sigma \in S_T} q_i^\sigma = 1$. One can interpret each q_i^σ as the posterior probability $\mathbb{P}(\sigma | \mathbf{g}_i)$ of a layer permutation σ , given the observed community labels for node i . This posterior probability is proportional to the product of the prior probability $\mathbb{P}(\sigma)$ and the likelihood $\mathbb{P}(\mathbf{g}_i | \sigma)$.

By comparing the right-hand side of (3.33) with the intralayer modularity terms from (3.1), we arrive at the following optimal values for the interlayer-coupling parameters:

$$(3.35) \quad \omega_{st} = \frac{1}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t} \log \left(1 + \frac{p_{st}}{1 - p_{st}} K_t \right) \sum_{\sigma \in S_T^{(s,t)}} q_i^\sigma.$$

Given a multilayer partition \mathbf{g} and values for the other parameters of our generative model, one can (in principle) calculate values for the weights q_i^σ using (3.34). Substituting these values into (3.35) then gives estimates for the coupling weights ω_{st} . In practice, the sum over permutations $\sigma \in S_T^{(s,t)}$ from (3.35) has $(T-1)!$ terms. Although there are many empirical data sets with a sufficiently small number of layers (say, $T \lesssim 6$) for which this calculation is feasible, this brute-force approach will not work in general.

We can make some simplifications in the case of uniform coupling, for which $p_{st} = p$ for $s, t \in \{1, \dots, T\}$. Furthermore, we assume that the number of communities is the same for each layer ($K_t = K$) and that we sample each permutation σ with equal probability (i.e., $\mathbb{P}(\sigma) = 1/T!$). In this case, the layers of our multiplex network are statistically equivalent, so on average any permutation of the layers should produce the same results. This implies that the quantities $q_i^\sigma = \mathbb{P}(\sigma | \mathbf{g}_i)$ (with $\sigma \in S_T$) are equal in expectation, which yields

$$(3.36) \quad \sum_{\sigma \in S_T^{(s,t)}} q_i^\sigma \approx \frac{1}{T} \sum_{\sigma \in S_T} q_i^\sigma = \frac{1}{T}.$$

It is possible to make this argument mathematically rigorous in the limit $T \rightarrow \infty$ by approximating the quantities q_i^σ (with $\sigma \in S_T^{(s,t)}$) with quantities $q_i^{\sigma'}$ (with $\sigma' \in S_T$) of similar magnitude. In Table 1, we provide some numerical evidence that our approximation is reasonable even for small T . Indeed, we observe that the sample mean of the expression $(\sum_{\sigma \in S_T^{(s,t)}} q_i^\sigma) - 1/T$ is close to 0 for different values of the copying probability p and the number T of layers. (For these examples, we use $K = 5$.) From a practical standpoint, we expect that any errors in the approximation (3.36) will have a small impact on the results of modularity maximization compared to other sources of error (e.g., the algorithm identifying different local optima of the modularity function across different runs).

Table 1

Sample mean and standard deviation (in parentheses) of $\sum_{\sigma \in S_T^{(s,t)}} q_i^\sigma - 1/T$ across 50 trials.

	$T = 3$	$T = 4$	$T = 5$	$T = 6$	$T = 7$	$T = 8$	$T = 9$	$T = 10$
$p = 0.5$	-0.013 (0.052)	0.027 (0.098)	0.009 (0.090)	0.008 (0.090)	-0.013 (0.076)	0.005 (0.078)	-0.004 (0.069)	-0.020 (0.047)
$p = 0.7$	0.000 (0.068)	-0.001 (0.073)	0.016 (0.082)	0.003 (0.096)	-0.008 (0.071)	-0.003 (0.086)	0.025 (0.100)	0.010 (0.082)
$p = 0.9$	-0.003 (0.016)	0.006 (0.048)	-0.004 (0.064)	0.007 (0.075)	0.004 (0.051)	0.010 (0.080)	0.008 (0.059)	-0.007 (0.052)

The approximation (3.36) gives

$$(3.37) \quad \omega = \frac{1}{T \langle \log \theta_{\text{in}} - \log \theta_{\text{out}} \rangle_t} \log \left(1 + \frac{p}{1 - p} K \right)$$

for multiplex networks with uniform coupling, so we see that $\omega_{\text{multiplex}} = \omega_{\text{temporal}}/T$. This scaling makes intuitive sense, as the number of directed interlayer edges per physical node is equal to $T - 1$ for temporal networks and is equal to $T(T - 1)$ for multiplex networks.

4. Parameter estimation. In section 3, we showed an equivalence between modularity maximization and statistical inference in SBMs, given certain assumptions, for different types of multilayer networks. In this section, we use these results to determine appropriate values for the resolution and interlayer-coupling parameters that arise in the formulation of multilayer modularity. For concreteness, we focus on the uniform setting, in which there are only two parameters (γ and ω). However, one can straightforwardly extend all of our results in this section to the layer-dependent case.

4.1. Iterative algorithm. In section 3, we derived expressions for γ and ω that depend on the unknown quantities θ_{in} , θ_{out} , p , and K . We therefore have to estimate the values of all of these parameters. To do this, we use an iterative procedure similar to the one that was proposed in [49] for monolayer networks. Given some initial guesses $\gamma^{(0)}$ and $\omega^{(0)}$, the iterative process alternates between running modularity maximization with the current estimates of these parameters and using the resulting community structure to estimate θ_{in} , θ_{out} , p , and K , thereby obtaining new values for γ and ω . The algorithm stops once γ and ω converge, up to some prespecified tolerance. We give the pseudocode for this iterative procedure in Algorithm 4.1. The iteration proceeds similarly when the parameters are layer-dependent. It is also possible to use the framework of Algorithm 4.1 to update only one of the two parameters, γ and ω , while keeping the other one fixed. A MATLAB implementation of Algorithm 4.1 (which we call ITERMODMAX) is available at <https://github.com/roxpamfil/IterModMax>.

Algorithm 4.1. Iterative algorithm for performing modularity maximization and estimating resolution and interlayer-coupling parameters in a multilayer network.

```

function ITERATIVEMODULARITYMAXIMIZATION( $\mathcal{A}$ )
  initialize  $\gamma = \gamma^{(0)}$  and  $\omega = \omega^{(0)}$ 
  while not converged do
     $\mathbf{g} \leftarrow \text{MAXIMIZE\_MODULARITY}(\mathcal{A}, \gamma, \omega)$             $\triangleright$  Run multilayer modularity maximization (e.g.,
                                                         using GENLOUVAIN [34])
     $\theta_{\text{in}}, \theta_{\text{out}}, p, K \leftarrow \text{ESTIMATE\_SBMPARAMETERS}(\mathcal{A}, \mathbf{g})$ 
                                                          $\triangleright$  Use detected communities to estimate SBM
                                                         parameters
     $\gamma \leftarrow \text{UPDATE\_GAMMA}(\theta_{\text{in}}, \theta_{\text{out}})$           $\triangleright$  Update  $\gamma$  using (3.9)
     $\omega \leftarrow \text{UPDATE\_OMEGA}(\theta_{\text{in}}, \theta_{\text{out}}, p, K)$     $\triangleright$  Update  $\omega$  using (3.16) or (3.37)
  end while
  return  $\gamma, \omega$                                             $\triangleright$  Optimal modularity parameters
  return  $\mathbf{g}$                                                   $\triangleright$  Communities detected using optimal
                                                         parameters
end function

```

A key step of Algorithm 4.1 is estimating θ_{in} , θ_{out} , p , and K , given a multilayer network's supra-adjacency matrix \mathcal{A} and the current multilayer partition \mathbf{g} . This task is relatively straightforward for temporal and multilevel networks, although it is more involved for multiplex networks. We explain how to perform this estimation in section SM3 of our supplementary

materials.

The most time-consuming part of Algorithm 4.1 is the modularity-maximization step of each iteration. Additionally, we observe empirically that the number of iterations that we need for convergence does not increase with the size of a network. (Instead, it depends on the strength of a network's community structure.) Therefore, the overall complexity of Algorithm 4.1 is the same as that of the modularity-maximization algorithm that one uses for community detection. We use GENLOUVAIN, whose computational complexity scales roughly as $\mathcal{O}(N^2T)$ for temporal networks and roughly as $\mathcal{O}(NT(N+T))$ for multiplex networks [34].⁴

Although Algorithm 4.1 is serial, it is possible to perform some related tasks in parallel. For instance, it can be useful to consider different choices of initial γ and ω (see section 5) or to run consensus clustering [43] at each iteration to obtain higher-quality partitions. (See our discussion in section 6.) One can easily parallelize either of these tasks to reduce the overall running time. Additionally, it appears within reach to implement a parallel version of the subroutine that maximizes multilayer modularity [11]; this would make it possible to apply Algorithm 4.1 to larger networks.

4.2. Numerical examples. We first illustrate the performance of Algorithm 4.1 on a simple network. Consider a network with $N = 1000$ nodes, $T = 2$ layers, and $K_1 = 20$ communities in the first layer that merge pairwise into $K_2 = 10$ communities in the second layer. We place edges independently in the two layers with probabilities $p_{\text{in}} = 0.32$ (for nodes in the same community) and $p_{\text{out}} = 0.1$ (for nodes in different communities). We choose these values near the detectability threshold for a monolayer network to make the community-detection task sufficiently difficult. Note that the parameters p_{in} and p_{out} are not the same as the parameters θ_{in} and θ_{out} from our generative model, as the latter are also multiplied by node degrees to generate edge probabilities. In Figure 1, we show the detected communities for two parameter choices: (1) the parameter values $\gamma = 1$ and $\omega = 1$ and (2) the values $\gamma \approx 1.60$ and $\omega \approx 1.30$ to which the iterative process from Algorithm 4.1 converges. The naive approach recovers the correct structure in the second layer, but not in the first; in particular, it identifies 10, rather than 20, communities in the first layer. By contrast, our iterative modularity-maximization algorithm identifies a multilayer partition that is close to the planted one.

We also test our parameter-estimation approach on existing multilayer network benchmarks, in which one can easily tune the strength of the planted community structure by changing one parameter. This flexibility allows us to study the community-detection problem at various levels of difficulty in a systematic way. In this section, we provide results for the generative temporal network model that was proposed by Ghasemian et al. [26]. Apart from the lack of degree correction, their model is the same as the one that we used in section 3.3 to study multilayer representations of temporal networks. This is a limitation, as real networks are likely to have different characteristics from those that are assumed by this (or some other) model. Therefore, in section SM4 of our supplementary materials, we use the models of Bazzi et al. [5] to test our approach in more general settings.

⁴The quadratic scaling is worse than that of the original Louvain algorithm [8] and is due to the fact that GENLOUVAIN is designed to work with any modularity matrix \mathbf{B} . Using an approach similar to that in [8], it is possible to implement a specialized version of GENLOUVAIN that maximizes multilayer modularity and that runs in approximately $\mathcal{O}(NT \log(N))$ time for sparse networks.

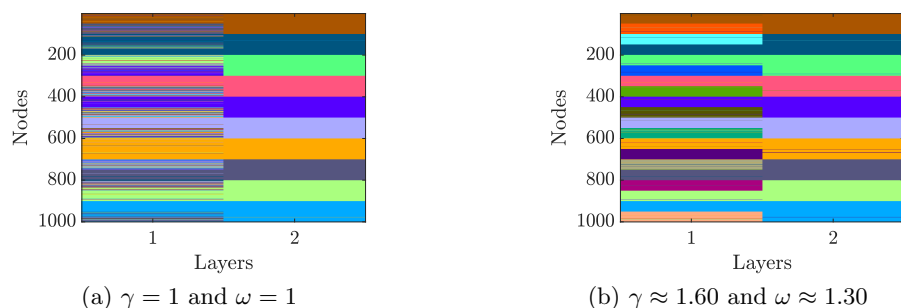


Figure 1. Detected multilayer partitions for a toy network with 20 communities in the first layer that merge pairwise into 10 communities in the second layer. We set the values of γ and ω to (a) $\gamma = 1$ (the “default” value) and $\omega = 1$ and (b) the optimal values $\gamma \approx 1.60$ and $\omega \approx 1.30$ that we infer using Algorithm 4.1. In both cases, the normalized mutual information (NMI) between the detected partitions and the planted one is close to 1 for the second layer. For the first layer, NMI is approximately 0.5 using the naive approach, and it is about 0.95 for the optimized approach.

The model of Ghasemian et al. uses the copying and resampling process that we described in section 3.3 to propagate community labels across temporal layers. In any layer other than the first, a node keeps its community label from the previous layer with probability η , and it samples a label uniformly at random (from K available labels) with probability $1 - \eta$. The parameter η is the same as our copying probability p , but we use different notation to emphasize that one is a tuning parameter for generating a network and the other is a parameter that we infer while detecting communities in that network. The expected community sizes are equal to N/K for all communities and all layers.

After assigning nodes to communities in this way, we place edges independently in each layer with probabilities p_{in} (if the nodes are in the same community) and p_{out} (if the nodes are in different communities). Because this generative model has no degree correction, it produces networks in which each node has the same expected degree in every layer. The ratio $\epsilon = p_{\text{out}}/p_{\text{in}}$ controls the strength of the planted community structure.

For their numerical experiments in [26], Ghasemian et al. limited themselves to situations with $K = 2$ communities, and we do the same in the present paper. We construct networks with $T = 40$ layers and $N = 512$ nodes in each layer. Each node has a mean degree of $c = 32$, and we vary η and ϵ in our calculations. In qualitative terms, the ability of a community-detection algorithm to recover a planted partition should decrease as ϵ increases (because community structure is becoming weaker), and it should increase as η increases (because the multilayer partition is changing less between successive layers). We show the results of these tests in Figure 2. (See Figure SM2 in our supplementary materials for the analogous results when using the model from Bazzi et al. [5] to generate temporal multilayer networks.) Note that our plots use the theoretically optimal values of γ and ω from (3.15) and (3.16), rather than iterating from some initial values. We choose to proceed in this manner because the authors of [26] also assumed that one knows all parameters aside from the community assignments \mathbf{g} . Towards the bottom of each plot, we also show the detectability thresholds that were derived in [26]; beyond these values, no algorithm can recover the planted communities. One derives these thresholds, which depend on η , in the limit in which the number T of layers

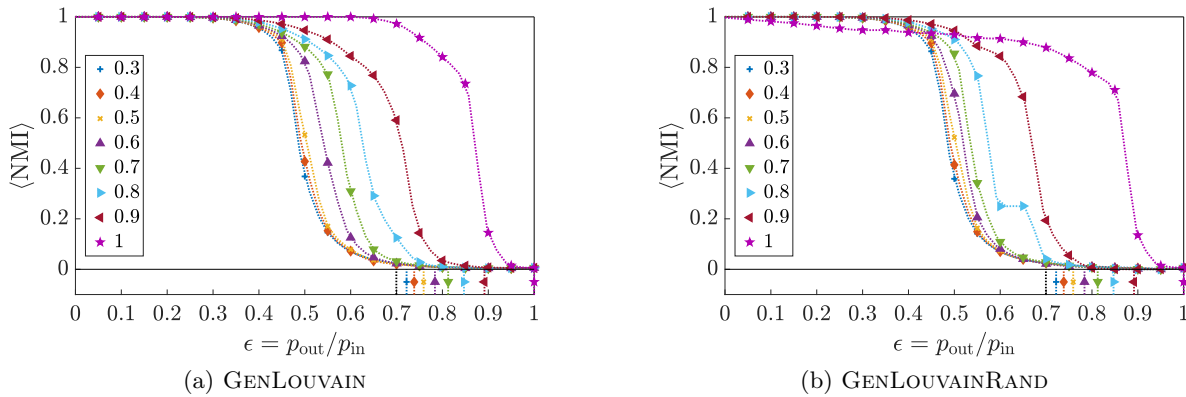


Figure 2. Results for the temporal multilayer benchmark network from [26]. The plots show layer-averaged NMI scores between the planted partition and the one detected by modularity maximization as a function of community strength ϵ for two different versions of the GENLOUVAIN algorithm [34]. Each line and set of markers corresponds to a different value of the copying probability η , and each data point is a mean over 100 trials. We use the theoretically optimal γ and ω for each choice of parameters. The bottom parts of the plots (vertical lines) indicate the detectability thresholds from [26], beyond which no algorithm is able to detect the planted structure. The plots differ in how the different versions of the GENLOUVAIN algorithm explore the space of multilayer partitions to increase modularity: (a) choose the move that yields the largest increase in modularity; (b) choose a move with a probability proportional to the resulting increase in modularity. The algorithm in panel (a) seems to perform better, especially for large η , although we note that this version of the algorithm can lead to poor behavior in certain cases for sufficiently large ω [6].

tends to infinity. We measure the level of success of each run by taking a mean over all layers of the normalized mutual information⁵ (NMI) between the planted partition g_{true} and the algorithmically detected one:

$$(4.1) \quad \langle \text{NMI}(\mathbf{g}, \mathbf{g}_{\text{true}}) \rangle = \frac{1}{T} \sum_{t=1}^T \text{NMI}(\mathbf{g}^t, \mathbf{g}_{\text{true}}^t).$$

The two different plots in Figure 2 correspond to two different versions of the GENLOUVAIN algorithm for optimizing multilayer modularity [34]. When the GENLOUVAIN algorithms seek to increase the value of modularity by reassigning nodes to communities, one can either (1) select the move that leads to the largest increase in modularity (GENLOUVAIN) or (2) use a “weighted” random move, which occurs with a probability that is proportional to the resulting increase in modularity (GENLOUVAINRAND). Previous work [6] found that the original version of the GENLOUVAIN algorithm can sometimes lead to undesirable behavior in which interlayer modularity terms are overemphasized relative to intralayer terms for sufficiently large ω . The GENLOUVAINRAND version of the algorithm was designed to avoid this problem. For our numerical experiments in Figure 2, we find that GENLOUVAIN tends to perform better. This is due to two factors. First, for small values of η , the theoretically optimal values of ω do not fall in the problematic regime that was described in [6]. Second, for large values of η , the fact that GENLOUVAIN puts so much emphasis on persistence of communities across layers tends

⁵We use the version of NMI in which one normalizes by the mean entropy of the two input partitions [45].

to help, rather than hinder, the algorithm's performance. Note that the results in Figure 2 do not contradict the earlier findings from [6].

Our results in Figure 2 exhibit the same qualitative features as similar plots in the paper that introduced these benchmarks [26]. Specifically, for each value of η , the success rate of the algorithms (as measured by NMI) is close to 1 for small values of ϵ , and it then drops rapidly to 0 near some critical value of ϵ that depends on η . This degradation in performance occurs before the theoretically predicted detectability thresholds that were derived in [26]. By contrast, the belief-propagation algorithm proposed in [26] performs well closer to these thresholds. There are several reasons for this observation. The first is that belief-propagation methods are known to give asymptotically optimal accuracy, in that they are able to recover planted communities all the way down to the threshold, as long as there is sufficient data (in terms of network size and number of layers) and as long as one initializes the relevant parameters sufficiently close to their optimal values [18, 19, 46]. By contrast, there are no such theoretical guarantees for modularity maximization. Additionally, the belief-propagation algorithm in [26] looks only for multilayer partitions with two communities, whereas our modularity-maximization algorithm also considers other partitions. Another possible reason for the suboptimal results in Figure 2 may be that our values of γ and ω , which we calculated from (3.15) and (3.16), are not the best possible values to use in practice. We investigate this next.

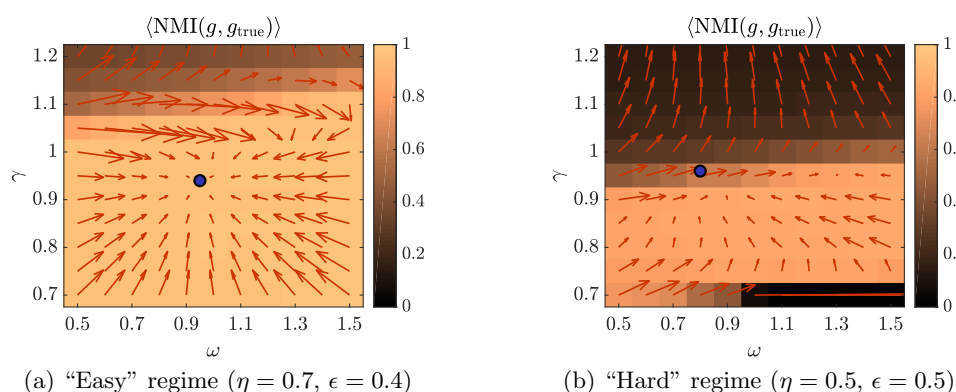


Figure 3. Illustration of the dynamics of the iterative process in Algorithm 4.1. In each plot, the heat map shows NMI scores between the planted partition \mathbf{g}_{true} and the algorithmically detected partition \mathbf{g} , averaged over 10 runs for each combination of γ and ω values. (We also tested parameter values outside the ranges in these plots.) The arrow centered at each (γ, ω) grid cell indicates the direction that yields the new estimates of γ and ω (averaged over the 10 trials) that we obtain from (3.15)–(3.16) after performing modularity maximization. For clarity, we have scaled down the arrow sizes. The blue disk in each plot indicates the location of the correct parameter values, which one can calculate from the planted partition. (a) An example with $\eta = 0.7$ and $\epsilon = 0.4$ (an “easy” case). As the arrows indicate, the iterative process converges to the true parameter values for many initial values $\gamma^{(0)}$ and $\omega^{(0)}$. (b) An example with $\eta = 0.5$ and $\epsilon = 0.5$ (a “hard” case). The region in the plane that results in the largest NMI values is roughly $0.8 \lesssim \gamma \lesssim 0.9$ and $0.7 \lesssim \omega \lesssim 1.5$. The optimal parameter values, $\gamma_{\text{opt}} \approx 0.96$ and $\omega_{\text{opt}} \approx 0.80$, lie slightly outside this regime. Although many of the arrows point in the direction of the blue disk, once the iterative process continues from there, it results in increasingly large resolution values and thus never converges. However, by returning the largest-modularity partition that it encounters during the iterative process, the algorithm identifies a solution that is close to the planted structure. (The NMI is about 0.86.)

In Figure 3, we show heat maps of mean NMI values for two different choices of η and ϵ . Specifically, after generating a multilayer network for each of the two settings, we sample values of γ and ω from a two-dimensional grid and run GENLOUVAINRAND 10 times for each pair of values. The mean NMI values in each grid cell indicate the level of agreement between the true partition and the one that we detect algorithmically. Figure 3(a) corresponds to an “easy” case, in which modularity maximization correctly recovers the planted partition at the optimal parameter values; and Figure 3(b) corresponds to a “hard” case, in which we are only partially able to recover the planted structure. (Situations in which there is no overlap between the detected partition and the planted one are not particularly interesting.) In each of these plots, we mark the optimal parameter values, γ_{opt} and ω_{opt} , with a blue disk. The arrows centered at each (γ, ω) grid point indicate the direction of the new γ and ω estimates (averaged over the 10 trials) that we obtain from (3.15)–(3.16) after performing modularity maximization. (For clarity, we have scaled down the arrow sizes.) This visualization indicates how our iterative approach explores (on average) the two-dimensional parameter space from different starting positions. In Figure 3(a), most of the arrows point to the location of the optimal point $(\gamma_{\text{opt}}, \omega_{\text{opt}})$. We are thus in the “easy” regime, in which the iterative process in Algorithm 4.1 converges to the correct parameter values and recovers the correct partition. The situation is more complicated in Figure 3(b). We observe that the optimal parameter values lie slightly outside the region in the (γ, ω) plane with the largest mean NMI values. Although many of the arrows point in the direction of the optimal point $(\gamma_{\text{opt}}, \omega_{\text{opt}})$, GENLOUVAINRAND detects suboptimal partitions into communities at these values. (We confirmed that the detected partitions are indeed suboptimal, as they give a smaller modularity value than the planted partition.) The arrows also illustrate why the iterative approach fails to converge in this case. From the optimal point, the resolution value becomes increasingly large, with no hope of returning to the large-NMI regime that is roughly in the region $0.7 \lesssim \gamma \lesssim 0.9$. This observation suggests that there is no stable fixed point of the iterative process in Figure 3, although there may be an unstable fixed point.

To circumvent the issue of the resolution γ increasing indefinitely for some networks (specifically, ones that tend to have relatively weak community structure), we find it useful to decrease γ in the iterations of Algorithm 4.1 whenever the number of communities exceeds some user-specified threshold.⁶ Specifically, we decrease γ by 20% (leaving ω unchanged) whenever K exceeds some given K_{max} . Although this approach introduces additional parameters to tweak, detected communities are less sensitive to the value of K_{max} than they are to the values of γ and ω . Decreasing γ when the number of communities is too large helps our algorithm explore the space of resolution and interlayer-coupling parameters in a useful way, even when it fails to converge to a fixed point. The algorithm can then return the largest-modularity partition that it encounters during this exploration. For the multilayer network in Figure 3(b), this approach finds a solution with an NMI score of 0.86. This example illustrates an important point: even when our proposed iterative algorithm does not converge to a fixed point, it can still identify meaningful community structure in a multilayer network. One way to select K_{max} is by applying an existing method for estimating the number of communities [64] in one

⁶When lack of convergence is not an issue, one should set this threshold to be large enough (e.g., larger than the number of nodes) so that it does not affect the iterative process.

layer of a network. Given that researchers often analyze monolayer network representations before constructing multilayer networks from the same data, this approach seems reasonable. Nonetheless, it is ad hoc, and we expect that better heuristics are possible.

5. Application: Lazega Law Firm. We illustrate our approach on the Lazega Law Firm network [44].⁷ This multiplex network encompasses interactions between $N = 71$ partners and associates who work at the same law firm. The network has $T = 3$ layers, each with directed edges, which encode *co-work*, *friendship*, and *advice* relationships. The data set also includes seven pieces of metadata, which we can use to analyze detected communities. These are (1) status (partner or associate); (2) gender; (3) office (Boston, Hartford, or Providence); (4) seniority (specifically, years with the firm); (5) age; (6) practice (litigation or corporate); and (7) law school (Harvard, Yale, University of Connecticut, or other). We group both the seniority and age metadata into five-year bins.

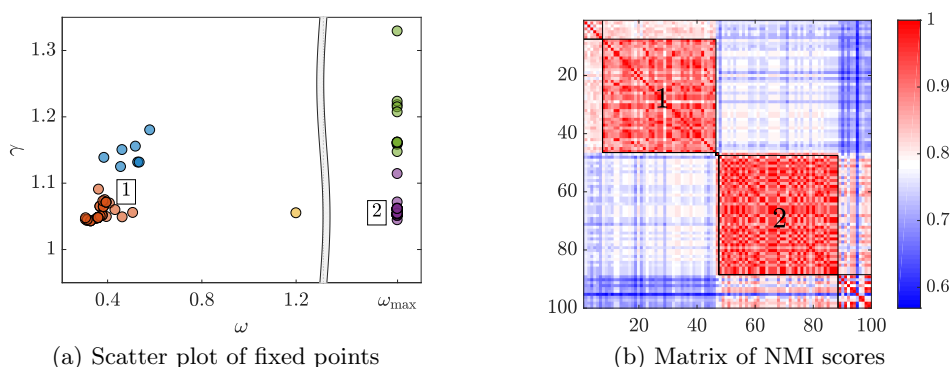


Figure 4. (a) Scatter plot of $(\gamma_{\text{opt}}, \omega_{\text{opt}})$ fixed points that our iterated modularity-maximization algorithm identifies when we apply it 100 times to the Lazega Law Firm network. The colors indicate membership to one of five groups that we identify using the K-means clustering algorithm [32]. About 80 runs converge to one of the groups that we label as “1” and “2”. (b) The matrix of layer-averaged NMI values between all pairs of partitions that we identify across 100 trials of our iterated modularity-maximization algorithm. The five diagonal blocks correspond to the five groups from panel (a).

We apply a version of Algorithm 4.1 for directed networks (see section SM1 of our supplementary materials for a derivation of this case) to the Lazega Law Firm data set, and we use GENLOUVAINRAND for community detection. (Using GENLOUVAIN gives very similar results.) Specifically, we run 100 trials of our iterative algorithm using initial values of γ and ω that we sample uniformly at random from the intervals $[0, 5]$ and $[0, 1]$, respectively. (For other applications, one can use a larger or smaller number of initializations, depending on network size and one’s computational budget.) Each trial converges within 30 iterations (which is our specified maximum), and typically our procedure reaches a fixed point in a

⁷For applications of our method to larger networks, see the recent doctoral dissertations [39, 53]. In [39] (see Appendix B), Klimm investigated changes in community structure in a temporal protein-interaction network of human white-blood cells under an inflammatory trigger. In [53] (see Chapter 5), Pamfil analyzed community structure in multilevel and temporal networks of customers who are linked to grocery products that they purchase.

much smaller number of iterations. In Figure 4(a), we show a scatter plot of the resulting fixed points $(\gamma_{\text{opt}}, \omega_{\text{opt}})$. This plot has two distinct regions of interest, depending on whether the copying probability p that our algorithm learns is strictly smaller than 1 or is equal to 1. In the former case, there is some variation in community structure across layers. In the latter case, we obtain identical communities across all three layers, and the theoretically optimal coupling value is $\omega_{\text{opt}} = \infty$. In the algorithm, we set ω equal to $\omega_{\text{max}} = 1000$ to introduce a very large penalty to nodes that switch communities across layers. We apply the K -means clustering algorithm to the (suitably normalized) points in Figure 4(a) to assign them to five groups,⁸ of which three correspond to the runs with $p < 1$ and the other two correspond to the runs with $p = 1$. Of the 100 runs, about 80 converge to one of the groups that we label as “1” and “2” in Figure 4(a). To compare the outputs of different trials, we calculate the layer-averaged NMI between all pairs of partitions that we obtain across our 100 trials, and we show the resulting similarity matrix in Figure 4(b). The borders around the diagonal blocks of this similarity matrix correspond to the five groups from Figure 4(a). We observe large similarity scores among trials in the same group, especially for the two dominant groups (the ones that we label as “1” and “2”). Because one of the groups is associated with $p < 1$ and the other is associated with $p = 1$, we analyze them separately.

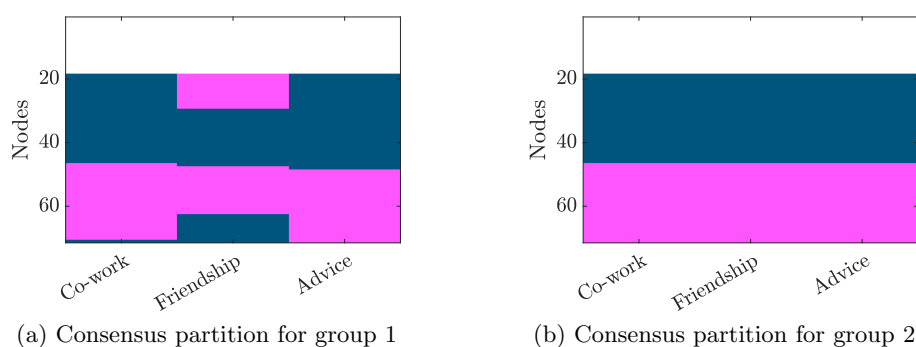


Figure 5. Visualizations of the consensus partitions that correspond to the two largest groups from Figure 4(a). We use the same ordering of the rows in both panels. The partition in (a) reveals a set of nodes that are in a different community in the friendship layer than they are in the other two layers. The partition in (b) has identical communities in the three layers.

We use consensus clustering [43] with the null model from [33] to extract a representative partition for the trials in each of the two labeled groups, and we visualize these consensus partitions in Figure 5. As expected from Figure 4(a), the consensus partition for group 2 has identical communities across the three layers. The structure of the consensus partition for group 1 is more interesting; it reveals a set of lawyers who are in a different community in the friendship layer than in the co-work and advice layers. In fact, the friendship layer is what distinguishes the two partitions from Figures 5(a) and 5(b). In Figure 6, we show sparsity plots of the adjacency matrix for the friendship layer; we order the rows and columns of this matrix according to the two consensus partitions from Figure 5. The three diagonal blocks in

⁸We base the number K of K -means clusters (which is not related to the number of communities) on visual inspection of Figure 4(a); other choices give similar results.

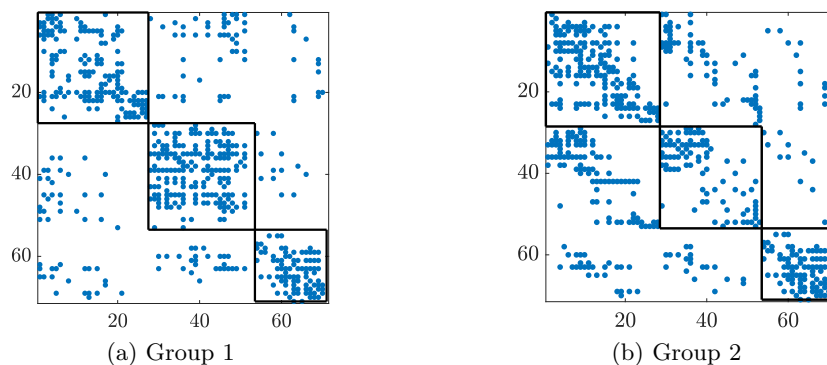


Figure 6. Sparsity plots of the adjacency matrix for the friendship layer, with rows and columns ordered according to their community assignments in the consensus partitions for (a) group 1 and (b) group 2.

each panel correspond to the three communities. There is a larger number of intracommunity entries in Figure 6(a) than in Figure 6(b). This suggests that the consensus partition for group 1 identifies stronger assortative structure in the friendship layer than the consensus partition for group 2. This simple example illustrates a key trade-off in multilayer community detection between detecting “optimal” communities in each layer (as the partition for group 1 tends to do) and detecting communities that are stable across layers (as is the case for the consensus partition for group 2).

Table 2

NMI scores between metadata and consensus partitions for the Lazega Law Firm data set. We highlight the largest value in each column.

	Office	Practice	Age	Seniority	Status	Gender	Law school
Partition 1	0.587	0.334	0.146	0.147	0.150	0.035	0.024
Partition 2	0.610	0.469	0.098	0.052	0.040	0.026	0.007
All 100 runs	0.577	0.406	0.125	0.106	0.093	0.037	0.022

We can gain further insight into the two consensus partitions by computing NMI scores with each type of metadata. We report our results in Table 2. For comparison, the bottom row of this table gives the mean NMI score across all 100 runs of our iterative algorithm. Our results are in line with a previous study of this data set [55]; it found that the office and practice metadata are related more strongly to network community structure than the law-school or gender metadata. We also observe that the two consensus partitions tend to correlate with different pieces of metadata. From Figure 5, we know that these differences arise from variations in the friendship layer. This suggests that age, seniority, and status are more important determinants of friendship than of co-work and advice relationships. By contrast, office location and type of practice (litigation or corporate) have a larger impact on co-work and advice relationships than on friendships.

As this example shows, there is no guarantee that our ITERMODMAX algorithm identifies a single set of optimal parameter values and thus a single multilayer assignment of a network’s nodes to communities. However, we feel that this is a strength, rather than a shortcoming,

of our method. Many empirical networks may have multiple meaningful partitions into communities; for example, these may align best with different pieces of metadata and appear as distinct peaks in a likelihood surface [55]. Even without the guarantee of a unique output, our method gives a systematic and principled way to reduce the number of cases to investigate using a much smaller number of computations than other similar methods [78].

6. Conclusions. The ability to coarse-grain a network by identifying its constituent communities is important for many applications. Many recent developments in community detection have focused on multilayer networks, which one can use to encode time-dependent interactions, multiple types of interactions, and other complications that can arise in complex systems. There exist a variety of methods for community detection in multilayer networks. In this paper, we focused on two approaches: modularity maximization and statistical inference using stochastic block models. Our main results connect these two types of methods. By considering various types of multilayer structure, we showed that the multilayer modularity objective function is, under specific circumstances, the same as the posterior log-probability that corresponds to certain multilayer SBMs. Therefore, maximizing one expression with respect to community assignments is equivalent to maximizing the other. This link between the two approaches highlights some implicit assumptions of classical multilayer modularity that previously were not clear. These include the assumption that layers are “statistically equivalent” (which entails that they are described by the same sets of parameters, θ_{in} and θ_{out}) and that all communities in a layer have the same expected sizes (because the null distributions \mathbb{P}_0^t are uniform and the copying probabilities p_t are the same for all nodes in a layer).

For temporal networks, we examined both the situation in which the resolution and interlayer-coupling parameters are uniform across all layers and the one in which they are layer-dependent. In the latter case, we proposed a novel layer-weighted version of modularity that is appropriate for situations in which communities in different layers have different statistical properties. We also briefly explored the use of layer-dependent models to detect structural change points [54], which correspond to layers in which a network’s mesoscale structure undergoes a major reorganization. (See section SM4 of our supplementary materials.) For multiplex networks, we proposed a new way of generating multilayer partitions by permuting the layers and then updating community labels in the order that is indicated by the permutation. In contrast to other generative models of multiplex networks [5], our approach makes it possible to write down the probability of generating a particular multilayer partition in closed form, and we expect that this will be useful for future efforts (e.g., for performing inference in associated SBMs).

Our work also suggests a principled method for determining appropriate values for the resolution and interlayer-coupling parameters that arise in the formulation of multilayer modularity. Our iterative algorithm provides a way to explore parameter space in a more efficient way than an exhaustive grid search, and we can thereby use fewer computations than previous approaches [4, 78]. We tested our algorithm on two families of multilayer benchmark networks. Our results showed the expected qualitative features, with better performance in networks with stronger community structure and more persistent communities across layers. We also used our algorithm on the Lazega Law Firm data set, which is a multiplex network with three layers. Our analysis revealed a set of lawyers who belong to a community in the

friendship layer that is different from their community in the co-work and advice layers.

There are many possible extensions of our work. One can derive similar results for multilayer networks with different forms of interlayer coupling from those that we discussed in this paper. Examples include temporal networks with memory, in which a layer depends not only on its predecessor, but rather on all other layers that came before it; networks with community-dependent coupling parameters; interconnected networks, in which interlayer connections are no longer diagonal; and multilayer networks with more than one aspect (e.g., networks that are both temporal and multiplex [38]). More broadly, our work (as well as Newman's previous work [49]) provides a recipe for deriving new versions of modularity for both monolayer and multilayer networks by starting from an SBM and making appropriate simplifications. Previous authors have used such a procedure to propose a version of modularity that incorporates information from metadata [79] by simplifying the SBM from [51]. We expect that one can similarly derive modularity functions from other SBMs that incorporate node annotations [31, 82] in both monolayer and multilayer settings. Another idea is to incorporate parameters that control expected community sizes in the definition of modularity, effectively relaxing the assumption that the null distributions \mathbb{P}_0^t are uniform [80].

We also anticipate that it is possible to improve on Algorithm 4.1 in several ways. For example, rather than using a single run of modularity maximization to estimate new resolution and interlayer-coupling parameter values, one can combine information from multiple runs to choose a direction in which to move and a step size to use in a move in that direction. Another practical issue arises when iterating over both γ and ω does not yield a fixed point. In these cases, we proposed a simple heuristic of decreasing the resolution when the number of communities becomes too large. Based on our observations, failure to converge is almost always due to the resolution increasing indefinitely, so an alternative solution is to fix γ throughout the iterations and optimize only ω [53]. One can then combine this version of our iterative algorithm with the multiresolution consensus-clustering approach from [36] to obtain a hierarchy of consensus partitions that span multiple scales. Another challenging task is to extend the results of the present paper to weighted networks. A key observation is that such weights only impact intralayer terms. Therefore, to study extensions of our analysis to weighted networks, it is sufficient to relate weighted modularity to a suitable monolayer SBM for weighted networks [2, 59].

In summary, we showed that two common and independently developed methods for community detection in multilayer networks—modularity maximization and SBM-based statistical inference—are closely related. This connection goes far beyond being only an interesting theoretical finding, as we believe that the techniques that we developed in the present paper can provide practical insights into devising more robust community-detection algorithms that do not require ad hoc experimentation with parameter values.

Acknowledgments. We thank Marya Bazzi, Lucas Jeub, and Florian Klimm for useful discussions.

REFERENCES

- [1] E. ABBE, *Community detection and stochastic block models: Recent developments*, J. Mach. Learn. Res., 18 (2018), pp. 1–86.
- [2] C. AICHER, A. Z. JACOBS, AND A. CLAUSET, *Learning latent block structure in weighted networks*, J. Complex Networks, 3 (2014), pp. 221–248.

- [3] P. BARUCCA, F. LILLO, P. MAZZARISI, AND D. TANTARI, *Disentangling group and link persistence in dynamic stochastic block models*, J. Stat. Mech. Theory Exp., 2018 (2018), 123407.
- [4] D. S. BASSETT, M. A. PORTER, N. F. WYMBES, S. T. GRAFTON, J. M. CARLSON, AND P. J. MUCHA, *Robust detection of dynamic community structure in networks*, Chaos, 23 (2013), 013142.
- [5] M. BAZZI, L. G. S. JEUB, A. ARENAS, S. D. HOWISON, AND M. A. PORTER, *A Framework for the Construction of Generative Models for Mesoscale Structure in Multilayer Networks*, preprint, <https://arxiv.org/abs/1608.06196>, 2016.
- [6] M. BAZZI, M. A. PORTER, S. WILLIAMS, M. McDONALD, D. J. FENN, AND S. D. HOWISON, *Community detection in temporal multilayer networks, with an application to correlation networks*, Multiscale Model. Simul., 14 (2016), pp. 1–41, <https://doi.org/10.1137/15M1009615>.
- [7] R. F. BETZEL AND D. S. BASSETT, *Multi-scale brain networks*, NeuroImage, 160 (2017), pp. 73–83.
- [8] V. D. BLONDEL, J.-L. GUILLAUME, R. LAMBIOTTE, AND E. LEFEBVRE, *Fast unfolding of communities in large networks*, J. Stat. Mech. Theory Exp., 2008 (2008), P10008.
- [9] S. BOCCALETTI, G. BIANCONI, R. CRIADO, C. D. GENIO, J. GÓMEZ-GARDENES, M. ROMANCE, I. SENDINA-NADAL, Z. WANG, AND M. ZANIN, *The structure and dynamics of multilayer networks*, Phys. Rep., 544 (2014), pp. 1–122.
- [10] U. BRANDES, D. DELLING, M. GAERTLER, R. GÖRKE, M. HOEFER, Z. NIKOLOSKI, AND D. WAGNER, *On modularity clustering*, IEEE Trans. Knowledge Data Engrg., 20 (2008), pp. 172–188.
- [11] A. BROWET, P.-A. ABSIL, AND P. VAN DOOREN, *Fast Community Detection Using Local Neighbourhood Search*, preprint, <https://arxiv.org/abs/1308.6276>, 2013.
- [12] L. CANTINI, E. MEDICO, S. FORTUNATO, AND M. CASELLE, *Detection of gene communities in multi-networks reveals cancer drivers*, Sci. Rep., 5 (2015), 17386.
- [13] A. CONDON AND R. M. KARP, *Algorithms for graph partitioning on the planted partition model*, Random Structures Algorithms, 18 (2001), pp. 116–140.
- [14] P. CSERMELY, A. LONDON, L.-Y. WU, AND B. UZZI, *Structure and dynamics of core/periphery networks*, J. Complex Networks, 1 (2013), pp. 93–123.
- [15] C. DE BACCO, E. A. POWER, D. B. LARREMORE, AND C. MOORE, *Community detection, link prediction, and layer interdependence in multilayer networks*, Phys. Rev. E, 95 (2017), 042317.
- [16] M. DE DOMENICO, C. GRANELL, M. A. PORTER, AND A. ARENAS, *The physics of spreading processes in multilayer networks*, Nature Phys., 12 (2016), pp. 901–906.
- [17] M. DE DOMENICO, A. LANCICHINETTI, A. ARENAS, AND M. ROSVALL, *Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems*, Phys. Rev. X, 5 (2015), 011027.
- [18] 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 (2011), 066106.
- [19] A. DECELLE, F. KRZAKALA, C. MOORE, AND L. ZDEBOROVÁ, *Inference and phase transitions in the detection of modules in sparse networks*, Phys. Rev. Lett., 107 (2011), 065701.
- [20] J.-C. DELVENNE, S. N. YALIRAKI, AND M. BARAHONA, *Stability of graph communities across time scales*, Proc. Natl. Acad. Sci. USA, 107 (2010), pp. 12755–12760.
- [21] S. FORTUNATO, *Community detection in graphs*, Phys. Rep., 486 (2010), pp. 75–174.
- [22] S. FORTUNATO AND M. BARTHELEMY, *Resolution limit in community detection*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 36–41.
- [23] S. FORTUNATO AND D. HRIC, *Community detection in networks: A user guide*, Phys. Rep., 659 (2016), pp. 1–44.
- [24] B. K. FOSDICK, D. B. LARREMORE, J. NISHIMURA, AND J. UGANDER, *Configuring random graph models with fixed degree sequences*, SIAM Rev., 60 (2018), pp. 315–355, <https://doi.org/10.1137/16M1087175>.
- [25] A. GHASEMIAN, H. HOSSEINMARDI, AND A. CLAUSET, *Evaluating overfit and underfit in models of network community structure*, IEEE Trans. Knowledge Data Engrg. (2019), <https://doi.org/10.1109/TKDE.2019.2911585>.
- [26] A. GHASEMIAN, P. ZHANG, A. CLAUSET, C. MOORE, AND L. PEEL, *Detectability thresholds and optimal algorithms for community structure in dynamic networks*, Phys. Rev. X, 6 (2016), 031005.
- [27] S. GOMEZ, A. DIAZ-GUILERA, J. GOMEZ-GARDENES, C. J. PEREZ-VICENTE, Y. MORENO, AND A. ARENAS, *Diffusion dynamics on multiplex networks*, Phys. Rev. Lett., 110 (2013), 028701.

- [28] B. H. GOOD, Y.-A. DE MONTJOYE, AND A. CLAUSET, *Performance of modularity maximization in practical contexts*, Phys. Rev. E, 81 (2010), 046106.
- [29] P. HOLME AND J. SARAMÄKI, *Temporal networks*, Phys. Rep., 519 (2012), pp. 97–125.
- [30] D. HRIC, K. KASKI, AND M. KIVELÄ, *Stochastic block model reveals maps of citation patterns and their evolution in time*, J. Informetrics, 12 (2018), pp. 757–783.
- [31] D. HRIC, T. P. PEIXOTO, AND S. FORTUNATO, *Network structure, metadata, and the prediction of missing nodes and annotations*, Phys. Rev. X, 6 (2016), 031038.
- [32] A. K. JAIN, M. N. MURTY, AND P. J. FLYNN, *Data clustering: A review*, ACM Comput. Surveys (CSUR), 31 (1999), pp. 264–323.
- [33] L. G. S. JEUB, *Networks, Communities, and Consumer Behaviour*, Ph.D. thesis, University of Oxford, Oxford, UK, 2015.
- [34] L. G. S. JEUB, M. BAZZI, I. S. JUTLA, AND P. J. MUCHA, *A Generalized Louvain Method for Community Detection Implemented in MATLAB*, <https://github.com/GenLouvain/GenLouvain>, 2011–2018, version 2.1.
- [35] L. G. S. JEUB, M. W. MAHONEY, P. J. MUCHA, AND M. A. PORTER, *A local perspective on community structure in multilayer networks*, Network Sci., 5 (2017), pp. 144–163.
- [36] L. G. S. JEUB, O. SPORNS, AND S. FORTUNATO, *Multiresolution consensus clustering in networks*, Sci. Rep., 8 (2018), 3259.
- [37] B. KARRER AND M. E. J. NEWMAN, *Stochastic blockmodels and community structure in networks*, Phys. Rev. E, 83 (2011), 016107.
- [38] M. KIVELÄ, A. ARENAS, M. BARTHELEMY, J. P. GLEESON, Y. MORENO, AND M. A. PORTER, *Multilayer networks*, J. Complex Networks, 2 (2014), pp. 203–271.
- [39] F. KLIMM, *Generalised Networks for Protein Interaction Analysis*, Ph.D. thesis, University of Oxford, Oxford, UK, 2018.
- [40] R. LAMBIOTTE, *Multi-scale modularity in complex networks*, in Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, IEEE, Washington, DC, 2010, pp. 546–553.
- [41] R. LAMBIOTTE, J.-C. DELVENNE, AND M. BARAHONA, *Laplacian Dynamics and Multiscale Modular Structure in Networks*, preprint, <https://arxiv.org/abs/0812.1770>, 2008.
- [42] R. LAMBIOTTE, J.-C. DELVENNE, AND M. BARAHONA, *Random walks, Markov processes and the multi-scale modular organization of complex networks*, IEEE Trans. Network Sci. Engrg., 1 (2014), pp. 76–90.
- [43] A. LANCICHINETTI AND S. FORTUNATO, *Consensus clustering in complex networks*, Sci. Rep., 2 (2012), 336.
- [44] E. LAZEGA, *The Collegial Phenomenon: The Social Mechanisms of Cooperation among Peers in a Corporate Law Partnership*, Oxford University Press, Oxford, UK, 2001.
- [45] M. LI, J. H. BADGER, X. CHEN, S. KWONG, P. KEARNEY, AND H. ZHANG, *An information-based sequence distance and its application to whole mitochondrial genome phylogeny*, Bioinformatics, 17 (2001), pp. 149–154.
- [46] C. MOORE, *The computer science and physics of community detection: Landscapes, phase transitions, and hardness*, Bull. Eur. Assoc. Theoret. Comput. Sci., 121 (2017), pp. 25–61.
- [47] P. J. MUCHA, T. RICHARDSON, K. MACON, M. A. PORTER, AND J.-P. ONNELA, *Community structure in time-dependent, multiscale, and multiplex networks*, Science, 328 (2010), pp. 876–878.
- [48] M. E. J. NEWMAN, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), 036104.
- [49] M. E. J. NEWMAN, *Equivalence between modularity optimization and maximum likelihood methods for community detection*, Phys. Rev. E, 94 (2016), 052315.
- [50] M. E. J. NEWMAN, *Networks*, 2nd ed., Oxford University Press, Oxford, UK, 2018.
- [51] M. E. J. NEWMAN AND A. CLAUSET, *Structure and inference in annotated networks*, Nature Commun., 7 (2016), 11863.
- [52] M. E. J. NEWMAN AND M. GIRVAN, *Finding and evaluating community structure in networks*, Phys. Rev. E, 69 (2004), 026113.
- [53] A. R. PAMFIL, *Communities in Annotated, Multilayer, and Correlated Networks*, Ph.D. thesis, University of Oxford, Oxford, UK, 2018.

- [54] L. PEEL AND A. CLAUSET, *Detecting change points in the large-scale structure of evolving networks*, in Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI, Palo Alto, CA, 2015, pp. 2914–2920.
- [55] L. PEEL, D. B. LARREMORE, AND A. CLAUSET, *The ground truth about metadata and community detection in networks*, Sci. Adv., 3 (2017), e1602548.
- [56] T. P. PEIXOTO, *Parsimonious module inference in large networks*, Phys. Rev. Lett., 110 (2013), 148701.
- [57] T. P. PEIXOTO, *Inferring the mesoscale structure of layered, edge-valued, and time-varying networks*, Phys. Rev. E, 92 (2015), 042807.
- [58] T. P. PEIXOTO, *Bayesian Stochastic Blockmodeling*, preprint, <https://arxiv.org/abs/1705.10225>, 2017; in Advances in Network Clustering and Blockmodeling, P. Doreian, V. Batagelj, and A. Ferligoj, eds., John Wiley & Sons, New York, to appear.
- [59] T. P. PEIXOTO, *Nonparametric weighted stochastic block models*, Phys. Rev. E, 97 (2018), 012306.
- [60] T. P. PEIXOTO AND M. ROSVALL, *Modelling sequences and temporal networks with dynamic community structures*, Nature Commun., 8 (2017), 582.
- [61] M. A. PORTER, *WHAT IS... a multilayer network*, Notices Amer. Math. Soc., 65 (2018), pp. 1419–1423.
- [62] M. A. PORTER, J.-P. ONNELA, AND P. J. MUCHA, *Communities in networks*, Notices Amer. Math. Soc., 56 (2009), pp. 1082–1097, 1164–1166.
- [63] J. REICHARDT AND S. BORNHOLDT, *Statistical mechanics of community detection*, Phys. Rev. E, 74 (2006), 016110.
- [64] M. A. RIOLO, G. T. CANTWELL, G. REINERT, AND M. E. J. NEWMAN, *Efficient method for estimating the number of communities in a network*, Phys. Rev. E, 96 (2017), 032310.
- [65] P. ROMBACH, M. A. PORTER, J. H. FOWLER, AND P. J. MUCHA, *Core-periphery structure in networks (revisited)*, SIAM Rev., 59 (2017), pp. 619–646, <https://doi.org/10.1137/17M1130046>.
- [66] R. A. ROSSI AND N. K. AHMED, *Role discovery in networks*, IEEE Trans. Knowledge Data Engrg., 27 (2015), pp. 1112–1131.
- [67] M. ROSVALL AND C. T. BERGSTROM, *Maps of random walks on complex networks reveal community structure*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 1118–1123.
- [68] M. ROSVALL, A. V. ESQUIVEL, A. LANCICHINETTI, J. D. WEST, AND R. LAMBIOTTE, *Memory in network flows and its effects on spreading dynamics and community detection*, Nature Commun., 5 (2014), 4630.
- [69] M. SARZYNSKA, E. A. LEICHT, G. CHOWELL, AND M. A. PORTER, *Null models for community detection in spatially-embedded, temporal networks*, J. Complex Networks, 4 (2016), pp. 363–406.
- [70] M. T. SCHAUB, J.-C. DELVENNE, M. ROSVALL, AND R. LAMBIOTTE, *The many facets of community detection in complex networks*, Appl. Network Sci., 2 (2017), 4.
- [71] M. T. SCHAUB, J.-C. DELVENNE, S. N. YALIRAKI, AND M. BARAHONA, *Markov dynamics as a zooming lens for multiscale community detection: Non clique-like communities and the field-of-view limit*, PLoS ONE, 7 (2012), e32210.
- [72] N. STANLEY, S. SHAI, D. TAYLOR, AND P. J. MUCHA, *Clustering network layers with the strata multilayer stochastic block model*, IEEE Trans. Network Sci. Engrg., 3 (2016), pp. 95–105.
- [73] D. TAYLOR, S. SHAI, N. STANLEY, AND P. J. MUCHA, *Enhanced detectability of community structure in multilayer networks through layer aggregation*, Phys. Rev. Lett., 116 (2016), 228301.
- [74] M. VAIANA AND S. MULDOON, *Resolution Limits for Detecting Community Changes in Multilayer Networks*, preprint, <https://arxiv.org/abs/1803.03597>, 2018.
- [75] E. VALDANO, L. FERRERI, C. POLETTI, AND V. COLIZZA, *Analytical computation of the epidemic threshold on temporal networks*, Phys. Rev. X, 5 (2015), 021005.
- [76] T. VALLÈS-CATALÀ, F. A. MASSUCCI, R. GUIMERA, AND M. SALES-PARDO, *Multilayer stochastic block models reveal the multilayer structure of complex networks*, Phys. Rev. X, 6 (2016), 011036.
- [77] S. WASSERMAN AND K. FAUST, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1994.
- [78] W. H. WEIR, S. EMMONS, R. GIBSON, D. TAYLOR, AND P. J. MUCHA, *Post-processing partitions to identify domains of modularity optimization*, Algorithms, 10 (2017), 93.
- [79] J.-R. XIE AND B.-H. WANG, *Modularity-like objective function in annotated networks*, Frontiers Phys., 12 (2017), 128903.

- [80] P. ZHANG, C. MOORE, AND M. E. J. NEWMAN, *Community detection in networks with unequal groups*, Phys. Rev. E, 93 (2016), 012303.
- [81] X. ZHANG, C. MOORE, AND M. E. J. NEWMAN, *Random graph models for dynamic networks*, Eur. Phys. J. B, 90 (2017), 200.
- [82] Y. ZHANG, E. LEVINA, AND J. ZHU, *Community detection in networks with node features*, Electron. J. Stat., 10 (2016), pp. 3153–3178.

SUPPLEMENTARY MATERIALS: Relating Modularity Maximization and Stochastic Block Models in Multilayer Networks*

A. Roxana Pamfil[†], Sam D. Howison[†], Renaud Lambiotte[†], and Mason A. Porter[‡]

SM1. The Relative Weight of Intralayer and Interlayer Contributions for Undirected and Directed Networks. In this section, we compare the optimal modularity functions that one obtains for undirected and directed networks using our approach from section 3 of the main manuscript. These two objective functions should be identical when the intralayer adjacency matrices \mathbf{A}^t are symmetric, as treating each undirected edge as two directed edges should not change the objective function that one optimizes for community detection. To simplify our discussion, we focus on the uniform setting (in which all layers are described by the same parameters).

Recall from section 3.2 that the log-likelihood for undirected multilayer networks under the PPM is

$$\log \mathbb{P}(\mathbf{A}|\mathbf{g}, \theta_{\text{in}}, \theta_{\text{out}}) = (\log \theta_{\text{in}} - \log \theta_{\text{out}}) \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t - \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\log \theta_{\text{in}} - \log \theta_{\text{out}}} \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + (\text{const.}). \quad (\text{SM1.1})$$

As in (3.8), the fact that this expression involves a sum over all i and j , and not just over $i \leq j$, entails that the entries A_{ij}^t and A_{ji}^t of an intralayer adjacency matrix both contribute to the log-likelihood. A similar derivation for directed networks gives

$$\log \mathbb{P}(\mathbf{A}|\mathbf{g}, \theta_{\text{in}}, \theta_{\text{out}}) = (\log \theta_{\text{in}} - \log \theta_{\text{out}}) \sum_{t=1}^T \sum_{i,j=1}^N \left(A_{ij}^t - \frac{\theta_{\text{in}} - \theta_{\text{out}}}{\log \theta_{\text{in}} - \log \theta_{\text{out}}} \frac{d_{i,\text{out}}^t d_{j,\text{in}}^t}{m_t'} \right) \delta(g_i^t, g_j^t) + (\text{const.}), \quad (\text{SM1.2})$$

where $d_{i,\text{out}}^t$ is the out-degree of node i in layer t , the quantity $d_{j,\text{in}}^t$ is the in-degree of node j in layer t , and $m_t' = \sum_{i,j=1}^N A_{ij}^t$ is the number of directed edges in layer t . When using a PPM to generate directed networks, the number of directed edges from node i to j in layer t follows a Poisson distribution whose mean is either $d_{i,\text{out}}^t d_{j,\text{in}}^t \theta_{\text{in}} / m_t'$ or $d_{i,\text{out}}^t d_{j,\text{in}}^t \theta_{\text{out}} / m_t'$, depending on whether or not i and j are in the same community in layer t . The expression (SM1.2) corresponds to a sum of monolayer modularities with the directed null model from [SM8].

If the adjacency matrices \mathbf{A}^t are all symmetric, $d_{i,\text{out}}^t = d_{i,\text{in}}^t = d_i^t$, $d_{j,\text{out}}^t = d_{j,\text{in}}^t = d_j^t$, and $m_t' = \sum_{i,j=1}^N A_{ij}^t = 2m_t$. (The number of directed edges is twice the number of undirected edges.) The expressions (SM1.1) and (SM1.2) become identical. When adding the

*Supplementary materials for SIMODS MS#M123130.

[†]Mathematical Institute, University of Oxford, Oxford OX2 6GG, United Kingdom (feier@maths.ox.ac.uk, howison@maths.ox.ac.uk, lambiotte@maths.ox.ac.uk).

[‡]Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095 (mason@math.ucla.edu).

log-prior term $\log \mathbb{P}(\mathbf{g}|p, K)$ and comparing the resulting expression with multilayer modularity, one obtains the same optimal value of ω in both the undirected and directed settings. The assumption that A_{ij}^t and A_{ji}^t both contribute to the log-likelihood (SM1.1) for undirected networks is a crucial one: if one were to introduce a factor of $1/2$ to discount this double-counting of terms, the resulting optimal value of ω would be twice as large when assuming that the matrices \mathbf{A}^t encode undirected networks instead of directed ones.

It is important to note that one of our paper’s key results, which relates the intralayer component of modularity to a sum of log-likelihoods and the interlayer component to a log-prior on community assignments, holds irrespective of the relative scaling between these two types of contributions. It is only when one uses this relation for parameter estimation (see section 4) that the formula for ω depends on this scaling. In our numerical experiments in section 4.2 and section SM4, we find that performance deteriorates if one rescales the log-likelihood (SM1.1) by a factor of $1/2$ to avoid double-counting edges. It is possible that one would see improved performance in our parameter-estimation algorithm by incorporating an additional tuning constant α and maximizing the modularity function that is equivalent (for suitable γ and ω) to the expression

$$(SM1.3) \quad \log \mathbb{P}(\mathcal{A}|\mathbf{g}, \theta_{\text{in}}, \theta_{\text{out}}) + \alpha \log \mathbb{P}(\mathbf{g}|p, K).$$

Such a constant is analogous to ones that have been introduced for community detection in the presence of metadata [SM4, SM10]. It is also akin to regularization constants that appear in other types of optimization problems, as the term $\log \mathbb{P}(\mathbf{g}|p, K)$ helps prevent overfitting to the community structure of individual layers. Investigating the effect of such a trade-off parameter is beyond the scope of the present paper, but it is an interesting topic for future work.

SM2. Derivation of Theoretically Optimal Parameter Values for Multilevel Networks.

In a multilevel network [SM9], the layers are a hierarchy of monolayer networks, and inter-layer edges between nodes indicate inclusion relationships. Like the temporal networks from section 3.3 of the main manuscript, the layers in multilevel networks have a natural ordering, which makes it straightforward to modify the approach from section 3.3 and obtain a similar result. This section complements our results for temporal and multiplex networks from section 3.3 and section 3.4, respectively.

Multilevel networks have been studied before, most notably by Snijders and collaborators, who used multilevel modeling for social network analysis [SM13, SM14, SM15]. The use of multilevel networks for community-detection applications is considerably rarer. We note a recent study by Barbillon et al. [SM1], who investigated a data set of collaborations and exchanges of resources between cancer researchers in France and their laboratories. Although Barbillon et al. modeled these interactions as a multiplex network of researchers, one can alternatively use a multilevel representation with researchers in one layer and laboratories in the other layer. A related example is a multilayer collaboration network, in which coauthored papers induce connections between individuals, research groups, departments, and universities (or a subset of these). A third potential application is international trade networks, in which nodes at higher levels represent larger geographical areas.

A key difference between multilevel networks and the other examples that we consider

in this paper (including temporal and multiplex networks) is that we need to consider non-diagonal interlayer edges. Additionally, nodes in different layers no longer necessarily correspond to the same set of entities. We therefore need to introduce some new notation. Let \mathcal{N}^t be the set of nodes in layer t , which has $|\mathcal{N}^t| = N^t$ nodes. Assume that \mathcal{N}^1 corresponds to the top level of the hierarchy. For $t \in \{2, \dots, T\}$, the functions $\pi^t : \mathcal{N}^t \rightarrow \mathcal{N}^{t-1}$ send nodes $i \in \mathcal{N}^t$ to their parents immediately higher up in the hierarchy. (For example, such a function can map researchers to their departments.) For convenience, we use the notation π_i^t to denote $\pi^t(i)$; note the implicit assumption that node i is from layer t .

This multilevel network's log-likelihood,

(SM2.1)

$$\log \mathbb{P}(\mathcal{A} | \mathbf{g}, \boldsymbol{\theta}_{\text{in}}, \boldsymbol{\theta}_{\text{out}}) = \sum_{t=1}^T \left[(\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t) \sum_{i,j \in \mathcal{N}^t} \left(A_{ij}^t - \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t} \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) \right] + (\text{const.}),$$

is similar to the one in (3.17) for nonuniform temporal networks. The prior on \mathbf{g} is also similar to that for temporal networks, although information flows from the top level of the hierarchy to the bottom level, rather than in the direction of time. First, we generate the vector \mathbf{g}^1 of community assignments in the top level by sampling from the null distribution \mathbb{P}_0^1 . To then obtain community assignments \mathbf{g}^t in subsequent layers, we assume that each node copies its community label from its parent with probability p_t and otherwise randomly samples a label from a null distribution \mathbb{P}_0^t . This process yields the following log-prior:

$$\begin{aligned} \log \mathbb{P}(\mathbf{g}) &= \sum_{i \in \mathcal{N}^1} \log \mathbb{P}_0^1(g_i^1) + \sum_{t=2}^T \sum_{i \in \mathcal{N}^t} \log [(1 - p_t) \mathbb{P}_0^t(g_i^t)] \\ &+ \sum_{t=2}^T \sum_{i \in \mathcal{N}^t} \log \left[1 + \frac{p_t}{(1 - p_t) \mathbb{P}_0^t(g_i^t)} \right] \delta(g_{\pi_i^t}^{t-1}, g_i^t). \end{aligned} \quad (\text{SM2.2})$$

As in section 3.3, we assume that the null distributions \mathbb{P}_0^t are uniform, so the first two terms in the right-hand side of (SM2.2) are independent of \mathbf{g} . In particular, we let $\mathbb{P}_0^t(g_i^t) = 1/K_t$. Ignoring constants, we then obtain the following expression for the log-prior:

$$\log \mathbb{P}(\mathbf{g}) = \sum_{t=2}^T \sum_{i \in \mathcal{N}^t} \log \left(1 + \frac{p_t}{1 - p_t} K_t \right) \delta(g_{\pi_i^t}^{t-1}, g_i^t) + (\text{const.}). \quad (\text{SM2.3})$$

From (SM2.1) and (SM2.3), it follows that the posterior distribution for \mathbf{g} is

$$\begin{aligned} \log \mathbb{P}(\mathbf{g} | \mathcal{A}, \boldsymbol{\theta}_{\text{in}}, \boldsymbol{\theta}_{\text{out}}, \mathbf{p}, \mathbf{K}) &= \sum_{t=1}^T \left[(\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t) \sum_{i,j \in \mathcal{N}^t} \left(A_{ij}^t - \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t} \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) \right] \\ &+ \sum_{t=2}^T \sum_{i \in \mathcal{N}^t} \log \left(1 + \frac{p_t}{1 - p_t} K_t \right) \delta(g_{\pi_i^t}^{t-1}, g_i^t) + (\text{const.}). \end{aligned} \quad (\text{SM2.4})$$

The corresponding layer-weighted modularity for multilevel networks is

$$(SM2.5) \quad Q(\mathbf{g}) = \sum_{t=1}^T \beta_t \sum_{i,j \in \mathcal{N}^t} \left(A_{ij}^t - \gamma_t \frac{d_i^t d_j^t}{2m_t} \right) \delta(g_i^t, g_j^t) + \sum_{t=2}^T \sum_{i \in \mathcal{N}^t} \omega_t \delta(g_{\pi_i^t}^{t-1}, g_i^t).$$

Comparing the expressions in (SM2.4) and (SM2.5), we arrive at the following values for the resolution parameters γ_t , coupling parameters ω_t , and layer weightings β_t :

$$(SM2.6) \quad \gamma_t = \frac{\theta_{\text{in}}^t - \theta_{\text{out}}^t}{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t},$$

$$(SM2.7) \quad \omega_t = \frac{1}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t} \log \left(1 + \frac{p_t}{1 - p_t} K_t \right),$$

$$(SM2.8) \quad \beta_t = \frac{\log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t}{\langle \log \theta_{\text{in}}^t - \log \theta_{\text{out}}^t \rangle_t}.$$

These are the same expressions as the ones for temporal networks in (3.21)–(3.23). Additionally, just as in the temporal case, one can also analyze the case of layer-independent parameters γ and ω . (In that case, the layer weights β_t are equal to 1.) Although this derivation is very similar to the one for temporal networks from section 3.3, the application to multilevel networks is markedly different, due to the interpretation of interlayer edges as inclusion relationships. Community detection in multilevel networks has been studied sparingly thus far, and we hope to see more such applications in the future.

SM3. Estimating SBM Parameters. In this section, we explain how to estimate, given a multilayer partition \mathbf{g} , the parameters of our various SBMs from section 3. This completes the description of our iterative modularity-maximization algorithm from section 4.1.

We approximate θ_{in} and θ_{out} in a manner similar to that presented in [SM11] for monolayer networks. Let M_{in}^t denote the random variable for the number of intracommunity edges in layer t under the degree-corrected SBM from section 3.2. The expected number of such edges across all layers is

$$(SM3.1) \quad \mathbb{E} \left[\sum_{t=1}^T M_{\text{in}}^t \right] = \frac{1}{2} \sum_{t=1}^T \sum_{i,j=1}^N \theta_{\text{in}} \frac{d_i^t d_j^t}{2m_t} \delta(g_i^t, g_j^t) = \frac{1}{2} \theta_{\text{in}} \sum_{t=1}^T \frac{\sum_r (\kappa_r^t)^2}{2m_t},$$

where κ_r^t denotes the sum of the degrees of nodes in community r in layer t . Replacing the expectation on the left-hand side of (SM3.1) with the observed number of intracommunity edges ($\sum_{t=1}^T m_{\text{in}}^t$) yields the estimate

$$(SM3.2) \quad \theta_{\text{in}} \approx \frac{\sum_{t=1}^T 2m_{\text{in}}^t}{\sum_{t=1}^T \frac{1}{2m_t} \sum_r (\kappa_r^t)^2}.$$

Using a similar approach, we estimate that

$$(SM3.3) \quad \theta_{\text{out}} \approx \frac{\sum_{t=1}^T 2m_{\text{out}}^t}{\sum_{t=1}^T \left[2m_t - \frac{1}{2m_t} \sum_r (\kappa_r^t)^2 \right]}.$$

In the layer-dependent case, one can similarly determine each set of parameters, θ_{in}^t and θ_{out}^t , by considering intracommunity and intercommunity edges separately for each layer.

We now consider the other two sets of parameters. We set the number K of communities equal to the observed number of distinct community labels in the most recent multilayer partition \mathbf{g} . That is,

$$(SM3.4) \quad K = |\{g_i^t : i \in \{1, \dots, N\}, t \in \{1, \dots, T\}\}|.$$

In the layer-dependent case, we set

$$(SM3.5) \quad K_t = |\{g_i^t : i \in \{1, \dots, N\}\}|, \quad t \in \{1, \dots, T\}.$$

Estimating the copying probabilities p (or p_t , if they are layer-dependent) is more difficult. The details of the calculation also depend on the type of multilayer network that one considers. We discuss temporal and multilevel networks in section SM3.1 and multiplex networks in section SM3.2.

SM3.1. Temporal and multilevel networks. To estimate the copying probability p , note that the probability that a node i is in the same community in two consecutive layers in our model is

$$(SM3.6) \quad \mathbb{P}(g_i^{t-1} = g_i^t) = p + \frac{1-p}{K}.$$

The first term corresponds to the probability that node i copies its community assignment from the previous layer, and the second term is the probability that node i samples that community uniformly at random from the K available choices. Estimating the same probability empirically from network data amounts to counting the number of times that a node stays in the same community (i.e., “persists”) across two consecutive layers:

$$(SM3.7) \quad \mathbb{P}(g_i^{t-1} = g_i^t) \approx \frac{1}{N(T-1)} \sum_{t'=2}^T \sum_{i'=1}^N \delta(g_{i'}^{t'-1}, g_{i'}^{t'}) = \frac{\text{Pers}(\mathbf{g})}{N(T-1)},$$

where we follow the notation in [SM3] and let $\text{Pers}(\mathbf{g})$ denote the number of instances in which a node belongs to the same community in consecutive layers. Equating the empirical probability (SM3.7) to the theoretical one in (SM3.6) yields

$$(SM3.8) \quad p + \frac{1-p}{K} \approx \frac{\text{Pers}(\mathbf{g})}{N(T-1)} \Rightarrow p \approx \left[\frac{\text{Pers}(\mathbf{g})}{N(T-1)} - \frac{1}{K} \right] \bigg/ \left(1 - \frac{1}{K} \right).$$

In the layer-dependent case, a similar derivation yields

$$(SM3.9) \quad p_t \approx \left[\frac{1}{N} \sum_{i=1}^N \delta(g_i^{t-1}, g_i^t) - \frac{1}{K} \right] \bigg/ \left(1 - \frac{1}{K} \right), \quad t \in \{2, \dots, T\}.$$

We follow an almost identical approach for multilevel networks. In the uniform case, our estimate for p is the same as in (SM3.8), although the calculation of persistence $\text{Pers}(\mathbf{g})$ is

different. We obtain

$$\begin{aligned}
 \text{(SM3.10)} \quad \text{Pers}(\mathbf{g}) &= \sum_{t=2}^T \sum_{i \in \mathcal{N}^t} \delta(g_{\pi_i^t}^{t-1}, g_i^t) \\
 \Rightarrow p &\approx \left[\frac{1}{T-1} \sum_{t=2}^T \frac{1}{N^t} \sum_{i \in \mathcal{N}^t} \delta(g_{\pi_i^t}^{t-1}, g_i^t) - \frac{1}{K} \right] \bigg/ \left(1 - \frac{1}{K} \right).
 \end{aligned}$$

Similarly, for the layer-dependent case, we obtain

$$\text{(SM3.11)} \quad p_t \approx \left[\frac{1}{N^t} \sum_{i \in \mathcal{N}^t} \delta(g_{\pi_i^t}^{t-1}, g_i^t) - \frac{1}{K} \right] \bigg/ \left(1 - \frac{1}{K} \right), \quad t \in \{2, \dots, T\}.$$

SM3.2. Multiplex networks. Estimating the probability p for the multiplex version of our SBM—or the probabilities p_{st} , if one is considering layer-dependent parameters—is significantly more difficult than for temporal or multilevel networks. We start by considering the uniform case.

Consider the probability $\mathbb{P}(g_i^s = g_i^t)$ for a randomly chosen node i and two arbitrary layers s and t (with $s \neq t$). For a given multilayer partition \mathbf{g} , we estimate $\mathbb{P}(g_i^s = g_i^t)$ by counting the number of instances in which two community labels are the same and dividing by the total number of instances:

$$\text{(SM3.12)} \quad \mathbb{P}(g_i^s = g_i^t) \approx \frac{1}{NT(T-1)} \sum_{i'=1}^N \sum_{t'=1}^T \sum_{s' \neq t'} \delta(g_{i'}^{s'}, g_{i'}^{t'}).$$

We now calculate $\mathbb{P}(g_i^s = g_i^t)$ for the generative model that we described in section 3.4. To do this, we need to account for all of the possible ways in which a node i can have the same community label in layers s and t . Suppose that we update labels in the order that is indicated by a permutation σ and that, in this permutation, there are $n-1$ intermediate layers between s and $t \equiv t_n$. We denote these layers by t_1, \dots, t_{n-1} . By symmetry, we may assume that layer s comes before layer t in the permutation, and we introduce a factor of 2 when necessary to account for the reverse case. We break down the probability $\mathbb{P}(g_i^s = g_i^t)$ into two components: either (1) node i in layer t copies the label from layer t_{n-1} and that label is identical to the one in layer s ; or (2) node i in layer t randomly samples the same label as the one in layer s . Therefore, we obtain the recursive relationship

$$\text{(SM3.13)} \quad \mathbb{P}(g_i^s = g_i^{t_n}) = p \mathbb{P}(g_i^s = g_i^{t_{n-1}}) + \frac{1-p}{K},$$

which holds for any n . We illustrate this process in Figure SM1. The base case is

$$\text{(SM3.14)} \quad \mathbb{P}(g_i^s = g_i^{t_1}) = p + \frac{1-p}{K}.$$

Let $f(x) = px + (1-p)/K$. Therefore, $f(1)$ is the probability that a node in one layer either copies the community label from the previous layer (after applying the permutation)

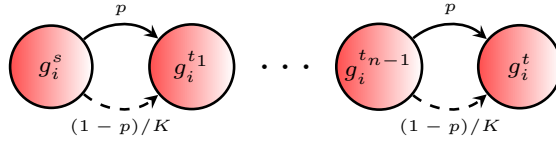


Figure SM1. Illustration of the copying and sampling process that propagates community labels across layers. Assume that after applying a permutation, the order of the layers is $s, t_1, \dots, t_{n-1}, t$. We are interested in the probability that a node i has the same community labels in layers s and t (i.e., the probability that $g_i^s = g_i^t$). For each layer from t_1, \dots, t_{n-1}, t , node i copies its community label from the previous layer with probability p (solid arrows), or it samples its label uniformly at random with probability $(1-p)/K$ (dashed arrows).

or that it randomly samples the same label. We summarize the recursion from (SM3.13) and (SM3.14) with the expression

$$(SM3.15) \quad \mathbb{P}(g_i^s = g_i^{t_n}) = f^{(n)}(1),$$

where $f^{(n)}$ denotes f composed with itself n times.

The expression (SM3.15) corresponds to the probability that nodes s and t share the same label when they are n layers apart after applying a permutation σ . To get the overall probability, we first have to count, for each n , the number of permutations with this property. Specifically, we need to consider all $\sigma \in S_T$ with $\sigma(u) = s, \sigma(u+1) = t_1, \dots, \sigma(u+n) = t_n \equiv t$ for some u . Using straightforward combinatorics, one can show that there are $(T-2)!(T-n)$ such permutations. This yields

$$(SM3.16) \quad \mathbb{P}(g_i^s = g_i^t) = \sum_{n=1}^{T-1} \frac{2(T-2)!(T-n)}{T!} f^{(n)}(1) = \sum_{n=1}^{T-1} \frac{2(T-n)}{T(T-1)} f^{(n)}(1).$$

For $p = 1$, note that $f(1) = 1$, so (SM3.16) gives $\mathbb{P}(g_i^s = g_i^t) = 1$, as expected. For $p < 1$, the function f has a unique fixed point at $1/K$. We can write

$$(SM3.17) \quad f^{(n)}(1) - \frac{1}{K} = p \left(f^{(n-1)}(1) - \frac{1}{K} \right) = \dots = p^n \left(1 - \frac{1}{K} \right),$$

which implies that

$$(SM3.18) \quad f^{(n)}(1) = p^n \frac{K-1}{K} + \frac{1}{K}.$$

This yields

$$(SM3.19) \quad \mathbb{P}(g_i^s = g_i^t) = \frac{2(1 - \frac{1}{K})}{T(T-1)} \sum_{n=1}^{T-1} p^n (T-n) + \frac{1}{K}.$$

We can therefore estimate p numerically by setting the right-hand side of (SM3.12) to be equal to that of (SM3.19). This leads to a polynomial root-finding problem that is easy to

solve using readily available algorithms. It is also possible to rewrite (SM3.19) using formulas for the geometric series and its derivatives, but that complicates the numerics without giving additional insights into the estimated value of p .

The above calculation does not work in the layer-dependent case. Previously, all permutations that put $n - 1$ layers between s and t gave the same value for the probability $\mathbb{P}(g_i^s = g_i^t)$, but this is no longer the case when there are different copying probabilities p_{st} for different layers. Instead, one needs to consider every permutation individually to obtain an exact expression for $\mathbb{P}(g_i^s = g_i^t)$. Such a brute-force approach is feasible for a small number of layers (say, $T \lesssim 6$), but it will not work in general. One avenue for future research is exploring suitable approximations for estimating the parameters p_{st} in this more complicated setting.

SM4. Additional Numerical Examples. Our examples in section 4.2 in the main manuscript consist of temporal networks in which each node has the same expected degree and each community has the same expected size. To test our methodology on networks with heterogeneous degree distributions and heterogeneous community sizes, both of which are common features of empirical networks, we turn to the general multilayer model in [SM2]. We use the code provided in [SM6] to generate networks using this generative model.

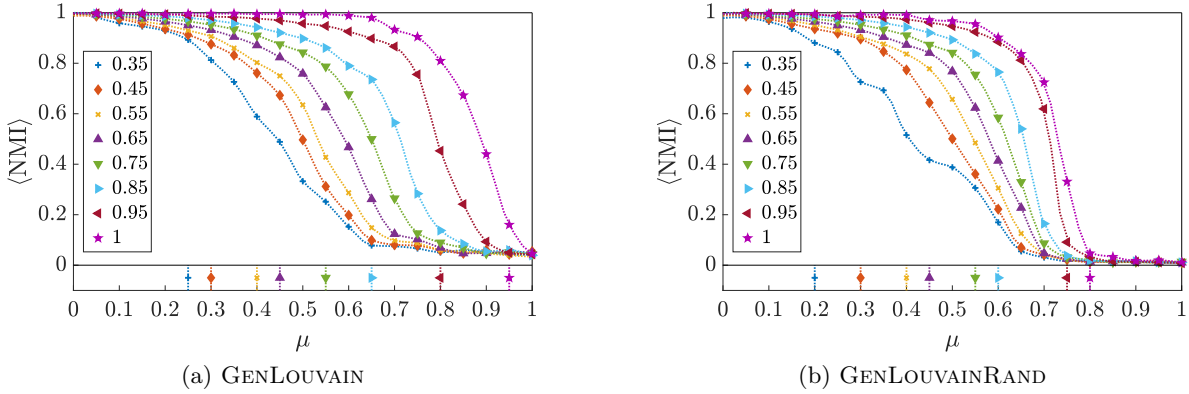


Figure SM2. Results for the temporal multilayer benchmark network from Bazzi et al. [SM2] using GENLOUVAIN and GENLOUVAINRAND to perform modularity maximization. We use the same parameter values that the authors used in their paper. The resulting networks have $T = 100$ layers and $N = 150$ nodes in each layer. There are $K = 5$ communities, whose expected sizes follow a symmetric Dirichlet distribution with a concentration parameter with value 1; the node degrees in each layer follow a truncated power-law distribution. (See [SM2] for details.) Our plots show layer-averaged NMI scores between the planted partition and the one that we detect by modularity maximization as a function of the mixing parameter μ . Each line and set of markers corresponds to a different value of the copying probability p_b , and each data point is a mean over 100 trials. We use Algorithm 4.1 to update the values of γ and ω from the initial values $\gamma^{(0)} = 1$ and $\omega^{(0)} = 1$. The lines at the bottom of each plot indicate the values of μ up to which at least 10% of the 100 runs converge to a fixed point.

Similar to the model of Ghasemian et al. [SM5], the simplest models from [SM2] have two parameters. A mixing parameter μ controls the strength of the planted community structure. When $\mu = 0$, all edges lie within communities; when $\mu = 1$, edges are equally likely to lie within communities or between communities. The second parameter of the benchmark from Bazzi et al. is the probability with which a node in one layer copies its community label from

a different layer; we denote this probability by p_b to distinguish it from our own parameter p . The parameters p and p_b are the same for temporal networks, but they are different for multiplex networks. The discrepancy in the latter case arises because our generative model from section 3.4 is different from the one from [SM2]. For multiplex networks, Bazzi et al. also introduced the quantity $\hat{p}_b = (T - 1)p_b$ for the probability that a node copies its community label from one of the other $T - 1$ layers. See [SM2] for additional details about these benchmarks.

In Figure SM2, we show the performance of our iterative modularity-maximization algorithm on the uniform temporal benchmark from [SM2]. We use the same parameter values to generate networks that Bazzi et al. [SM2] used for their calculations. The iterative process starts from $\gamma^{(0)} = 1$ and $\omega^{(0)} = 1$. The two plots use GENLOUVAIN and GENLOUVAINRAND to maximize modularity. GENLOUVAIN tends to perform better, especially for larger values of the copying probability p_b . The plots from Figure SM2 have the same qualitative features as those from Figure 2, with NMI scores close to 1 for μ close to 0 and rapid degradation in performance for critical values of μ that depend on the value of the copying probability p_b . We also observe a clear separation of the curves that correspond to different values of p_b , with better performance for progressively larger p_b . When $p_b = 1$, we can successfully recover at least some of the planted community structure up to $\mu = 1$. Towards the bottom of both plots, we show for each value of p_b the value of μ up to which at least 10% of the 100 runs converge to a fixed point in a maximum of 30 iterations. Recall that when our algorithm does not converge, it returns the largest-modularity partition that it encounters during the iterative process. For smaller values of p_b , the mean NMI between the algorithmically detected and planted communities remains large beyond these critical values of μ , suggesting the utility of our approach even when the iterative process fails to converge. For $p_b = 1$, the critical value of μ appears to coincide with the value of μ beyond which the calculated NMI is close to 0.

In [SM2], Bazzi et al. fixed $\gamma = 1$ and computed NMI between algorithmically detected and planted communities for different values of ω . They did not attempt to select an “optimal” set of parameter values, as this was not a goal of their paper. Using Algorithm 4.1 to select values of γ and ω in a principled way, we obtain NMI values that are comparable with or larger than those that were observed for the analogous calculations in [SM2] using the best possible value of ω (i.e., the one that yields the largest NMI between detected and planted partitions).¹

Bazzi et al. also examined temporal networks with nonuniform interlayer dependencies, with a focus on networks with change points [SM2]. We now apply our iterative algorithm with layer-dependent values of γ and ω to one such network with $T = 100$ layers using a mixing parameter $\mu = 0.4$. We set the copying probability to be $p_t = 0$ for $t \in \{25, 50, 75\}$ (the change-point layers), and we set $p_t = 0.9$ for all other layers. This induces a reorganization of mesoscale structure in layers 25, 50, and 75 of a sampled network. In Figure SM3, we show

¹Bazzi et al. used a different version of NMI (one that normalizes mutual information by the joint entropy of the input partitions). Therefore, it is not possible to systematically compare our results from Figure SM2 with theirs. However, we have recalculated some NMI values for specific choices of p_b and μ to enable a direct comparison to [SM2]. To give an example, for $p_b = 0.85$ and $\mu = 0.6$, we obtain $\langle \text{NMI} \rangle \approx 0.70$ using GENLOUVAIN and $\langle \text{NMI} \rangle \approx 0.65$ using GENLOUVAINRAND. Bazzi et al. obtained NMI values of up to 0.55 and 0.65, respectively, for the two versions of the algorithm.

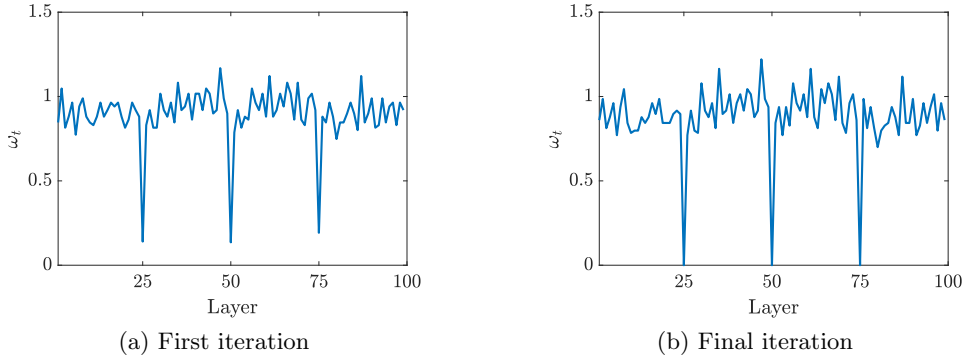


Figure SM3. Inferred values of the coupling parameters ω_t after (a) one iteration of our ITERMODMAX algorithm and (b) the final iteration (i.e., upon convergence) for a temporal network with change points. We use $p_t = 0$ for $t \in \{25, 50, 75\}$; this induces an abrupt change in community structure at these layers. For the remaining layers, we set $p_t = 0.9$. We let $\mu = 0.4$ for the mixing parameter; other choices give similar results. For the remaining benchmark parameters, we use the same values as in [SM2]. Specifically, we generate a network with $T = 100$ layers and $N = 150$ nodes in each layer. There are $K = 5$ communities, whose expected sizes follow a symmetric Dirichlet distribution with a concentration parameter with value 1; the node degrees follow a truncated power-law distribution. (See [SM2] for details.)

the inferred values of the coupling parameters ω_t using our layer-dependent ITERMODMAX algorithm both (a) after one iteration of the algorithm and (b) upon convergence (i.e., in the final iteration). Our algorithm correctly infers $\omega_t = 0$ at the three change points. The layer-averaged NMI between the output partition and the planted partition is approximately 0.96.

We make a few comments about Figure SM3. First, these results use a version of the ITERMODMAX algorithm that fixes $\gamma = 1$ throughout the iterative process. (Due to the large number of parameters in the layer-dependent case, Algorithm 4.1 does not converge when we vary both γ and ω .) We initialize all coupling parameters ω_t to 1. Second, we use a version of the GENLOUVAIN algorithm without the postprocessing step that increases multilayer modularity by relabeling communities to increase persistence. (See [SM3] and the implementation of GENLOUVAIN in [SM7] for details.) With postprocessing, our algorithm is unable to determine that layers 24 and 25 have unrelated community structures (and likewise for the other two change points). Third, although it can take our ITERMODMAX algorithm up to 20–30 iterations to converge to a solution like the one from Figure SM3(b), it only takes one iteration to identify the change-point layers [see Figure SM3(a)]. This suggests that waiting for the iterative algorithm to converge may not be necessary if one is interested only in the locations of change points (and not in the extent to which communities reorganize at change points).

In Figure SM4, we show the performance of our method on the uniform multiplex benchmark network from [SM2]. As with the temporal benchmark, we use the same parameter values to generate networks that Bazzi et al. used in their numerical experiments. We initialize our iterative algorithm with $\gamma^{(0)} = 1$ and $\omega^{(0)} = 1/T$, where T is the number of layers. (We justify our choice of $\omega^{(0)}$ using the scaling argument from section 3.4.) For both GEN-

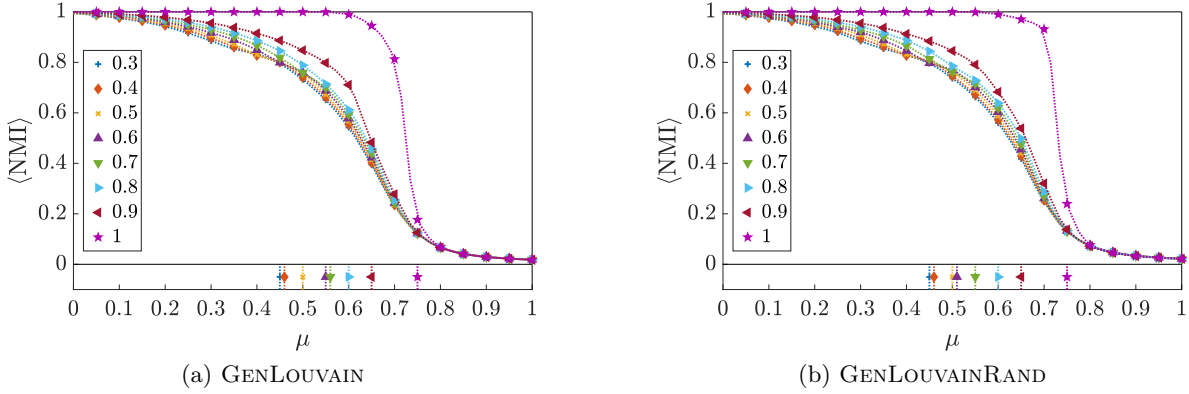


Figure SM4. Results for the multiplex benchmark from Bazzi et al. [SM2] using GENLOUVAIN and GENLOUVAINRAND to perform modularity maximization. We use the same parameter values that the authors used in their paper. The resulting networks have $T = 15$ layers and $N = 1000$ nodes in each layer. There are $K = 10$ communities, whose expected sizes follow a symmetric Dirichlet distribution with a concentration parameter with value 1; the node degrees in each layer follow a truncated power-law distribution. (See [SM2] for details.) Our plots show layer-averaged NMI scores between the planted partition and the one that we detect by modularity maximization as a function of the mixing parameter μ . Each line and set of markers corresponds to a different value of the copying probability $\hat{p}_b = (T - 1)p_b$, and each data point is a mean over 100 trials. We use Algorithm 4.1 to update the values of γ and ω from their initial values, which are $\gamma^{(0)} = 1$ and $\omega^{(0)} = 1/T$. The lines at the bottom of each plot indicate the values of μ up to which at least 10% of the 100 runs converge to a fixed point.

LOUVAIN and GENLOUVAINRAND, and with the exception of the case with $\hat{p}_b = 1$, we observe only marginal improvement in performance for larger values of the copying probability \hat{p}_b . Bazzi et al. made a similar observation in [SM2]. For $\hat{p}_b = 1$, the abrupt drop in performance coincides with the point at which fewer than 10% of runs converge to a fixed point. (Our choice of threshold is arbitrary; the results are similar for values other than 10%.) Our method of estimating the parameters γ and ω gives comparable NMI values to the best values that Bazzi et al. obtained in [SM2] for their analogous calculations.²

Our numerical experiments in this section demonstrate the promise of our approach for estimating resolution and interlayer-coupling parameters. We obtain NMI scores that often exceed the best NMI values that have been reported for multilayer modularity maximization without the parameter-estimation step [SM2]. It is inevitable that, for some networks, modularity maximization will underperform other community-detection methods [SM12] (e.g., the belief-propagation algorithm from [SM5] or inference algorithms that are based on general SBMs). However, when using modularity maximization for performing community detection, our way of estimating resolution and coupling parameters is a valuable tool.

²To be consistent with [SM2], we calculate a second set of NMI values; these normalize mutual information by the joint entropy of the input partitions, rather than by their mean entropy.

REFERENCES

- [1] P. BARBILLON, S. DONNET, E. LAZEGA, AND A. BAR-HEN, *Stochastic block models for multiplex networks: An application to a multilevel network of researchers*, Journal of the Royal Statistical Society: Series A (Statistics in Society), 180 (2017), pp. 295–314.
- [2] M. BAZZI, L. G. JEUB, A. ARENAS, S. D. HOWISON, AND M. A. PORTER, *A Framework for the Construction of Generative Models for Mesoscale Structure in Multilayer Network*, arXiv preprint arXiv:1608.06196 (2016).
- [3] M. BAZZI, M. A. PORTER, S. WILLIAMS, M. McDONALD, D. J. FENN, AND S. D. HOWISON, *Community detection in temporal multilayer networks, with an application to correlation networks*, Multiscale Modeling & Simulation: A SIAM Interdisciplinary Journal, 14 (2016), pp. 1–41.
- [4] S. EMMONS AND P. J. MUCHA, *Map equation with metadata: Varying the role of attributes in community detection*, Physical Review E, 100 (2019), 022301.
- [5] A. GHASEMIAN, P. ZHANG, A. CLAUSET, C. MOORE, AND L. PEEL, *Detectability thresholds and optimal algorithms for community structure in dynamic networks*, Physical Review X, 6 (2016), 031005.
- [6] L. G. S. JEUB AND M. BAZZI, *A generative model for mesoscale structure in multilayer networks implemented in MATLAB*. <https://github.com/MultilayerGM/MultilayerGM-MATLAB> (2016), version 1.0.
- [7] L. G. S. JEUB, M. BAZZI, I. S. JUTLA, AND P. J. MUCHA, *A generalized Louvain method for community detection implemented in MATLAB*. <https://github.com/GenLouvain/GenLouvain> (2011–2018), version 2.1.
- [8] E. A. LEICHT AND M. E. J. NEWMAN, *Community structure in directed networks*, Physical Review Letters, 100 (2008), p. 118703.
- [9] A. LOMI, G. ROBINS, AND M. TRANMER, *Editorial: Introduction to multilevel social networks*, Social Networks, 44 (2016), pp. 266–268.
- [10] A. C. MURPHY, S. GU, A. N. KHAMBHATI, N. F. WYMBS, S. T. GRAFTON, T. D. SATTERTHWAITE, AND D. S. BASSETT, *Explicitly Linking Regional Activation and Function Connectivity: Community Structure of Weighted Networks with Continuous Annotation*, arXiv preprint arXiv:1611.07962 (2016).
- [11] M. E. J. NEWMAN, *Equivalence between modularity optimization and maximum likelihood methods for community detection*, Physical Review E, 94 (2016), 052315.
- [12] L. PEEL, D. B. LARREMORE, AND A. CLAUSET, *The ground truth about metadata and community detection in networks*, Science Advances, 3 (2017), e1602548.
- [13] T. A. B. SNIJDERS AND C. BAERVELDT, *A multilevel network study of the effects of delinquent behavior on friendship evolution*, Journal of Mathematical Sociology, 27 (2003), pp. 123–151.
- [14] T. A. B. SNIJDERS, M. SPREEN, AND R. ZWAAGSTRA, *The use of multilevel modeling for analysing personal networks: Networks of cocaine users in an urban area*, Journal of Quantitative Anthropology, 5 (1995), pp. 85–105.
- [15] M. A. J. VAN DUIJN, J. T. VAN BUSSCHBACH, AND T. A. B. SNIJDERS, *Multilevel analysis of personal networks as dependent variables*, Social Networks, 21 (1999), pp. 187–210.