

Modularity belief propagation on multilayer networks to detect significant community structure

William Weir, Benjamin Walker, Lenka Zdeborová, Peter J Mucha

▶ **To cite this version:**

William Weir, Benjamin Walker, Lenka Zdeborová, Peter J Mucha. Modularity belief propagation on multilayer networks to detect significant community structure. 2020. cea-02529171

HAL Id: cea-02529171
https://cea.hal.science/cea-02529171

Preprint submitted on 2 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

arXiv:1908.04653v1 [cs.SI] 13 Aug 2019

MODULARITY BELIEF PROPAGATION ON MULTILAYER NETWORKS TO DETECT SIGNIFICANT COMMUNITY STRUCTURE*

WILLIAM H. WEIR†, BENJAMIN WALKER‡, LENKA ZDEBOROVÁ§, AND PETER J. MUCHA†‡

Abstract. Modularity based community detection encompasses a number of widely used, efficient heuristics for identification of structure in single- and multilayer networks. Recently, a belief propagation approach to modularity optimization provided a useful guide for identifying non-trivial structure in a way that other optimization heuristics have not. In this paper, we extend modularity belief propagation to multilayer networks. As part of this development, we also directly incorporate a resolution parameter. We show that the resolution parameter affects the convergence properties of the algorithm and yields different community structures than the baseline. We demonstrate our extension on synthetic multilayer networks, showing how our tool achieves near optimal performance and prevents overfitting. We highlight these advantages in comparison to another widely used tool, GenLouvain for multilayer modularity. Finally, we apply multilayer modularity belief propagation to two real-world multilayer networks and discuss practical concerns in implementing our method, which we have released as a Python package for general use.

Key words. community detection, modularity, belief propagation, networks, multilayer networks, message passing, resolution parameter

AMS subject classifications. 68Q25, 68R10, 68U05

1. Introduction.

Networks provide useful models for understanding complex systems across a wide range of problems in different domains, including biology, engineering and the social sciences. Recent attention has focused on developing tools to understand the expanded class of *multilayer* networks. The multilayer framework is quite flexible, allowing for the representation of multiplex pairwise interactions, dynamic networks, different classes of nodes, and "networks of networks" [22]. One class of ongoing challenges in network science, in particular for multilayer networks, is the detection and representation of high-level structure and communities (for review, see e.g. [11, 13, 43, 46, 48])

There are many computational approaches to identifying structure within networks. One family of approaches attempts to fit a statistical model to the observed network and uses hypothesis testing to assess the significance of proposed community structure. Many of these models are derived from the stochastic block model (SBM) [18, 21, 42]. Another approach is to define an objective function that measures the quality of community structure in a particular sense, and then optimize that objec-

* Submitted to the editors DATE.

Funding: This work was supported by grant R01DK111930 from the National Institute of Diabetes and Digestive and Kidney Diseases and by the James S. McDonnell Foundation 21st Century Science Initiative - Complex Systems Scholar Award grant #220020315, as well as by the NIH BD2K grant #T32CA201159 and NIH BCB training grant #T32GM67553. Additional support was provided by the Army Research Office (MURI award W911NF-18-1-0244). We acknowledge the support and hospitality of the Duke University where this work was initiated. The content is solely the responsibility of the authors and does not necessarily represent the official views of any of the funding organizations.

†Bioinformatics and Computational Biology, University of North Carolina, Chapel Hill, NC (wweir@med.unc.edu).

‡Carolina Center for Interdisciplinary Applied Mathematics, Department of Mathematics, University of North Carolina, Chapel Hill, NC

§Institut de Physique Théorique, CEA, CNRS, Université Paris-Saclay, CNRS

tive. For example, the Infomap approach [45] detects communities as groups of nodes that minimize the description length to encode random walks on a network. Another popular quality function, the modularity score developed by Newman and Girvan [35], compares the observed edge weight within groups to that expected under a null model of the network. Modularity was extended to multilayer networks by incorporating interlayer edges as additional coupling between the layers [33]. Generalizing a re-derivation of modularity from a Laplacian dynamics perspective [25, 26], Mucha *et al.* [33] developed a formula for multilayer modularity that can be written generally in the supra-matrix form

$$Q(\gamma,\omega) = \sum_{i,j} \left(A_{ij} - \gamma P_{ij} + \omega C_{ij}\right) \delta(c_i, c_j) \tag{1.1}$$

where i and j each index distinct node-layer objects, possibly in different layers, A_{ij} is the supra-adjacency[1] encoding the intralayer edges, P_{ij} describes the expected number of intralayer edges based on the selected random model(s), and C_{ij} encodes the interlayer connections. The normalizing factor traditionally written in front of the summation above has been absorbed here into the constituent terms for notational convenience. We here assume for simplicity the Newman-Girvan model for undirected edges within each layer, writing the null model contribution (prior to absorbing the normalizing factor) from layer l as

$$P_{ij} = \begin{cases} \frac{d_i d_j}{2m_{l_i}} & l_i = l_j \\ 0 & l_i \neq l_j \end{cases} \tag{1.2}$$

where l_i is the layer containing node-layer i, (*e.g.* $i \in \mathcal{V}_{l_i}$), $d_i = \sum_j A_{ij}$, and $m_{l_i} = \sum_{i,j\in\mathcal{V}_{l_i}} A_{ij}$ is the total weight of edges in layer l_i. We have implicitly assumed that a given node-layer i only participates in intralayer edges within its own layer (by definition). In the case where edge weights are binary ($A_{ij} \in \{0,1\}$), d_i is the degree of node i. For weighted networks, A_{ij} is continuous and $d_i = \sum_j A_{ij}$ is called the 'strength' of node i. Similar null models are available for bipartite graphs, directed networks, and networks with signed edges (see, e.g., the supplement of [33] for references to appropriate forms for P_{ij} in different contexts).

In general, maximizing Eq 1.1 over the combinatorially large space of possible partitions is NP-hard [6]. Several fast and efficient algorithms exist for locally optimizing modularity, including Louvain [5] and the GenLouvain [20] extension for optimizing multilayer modularity. One of the main problems with optimizing modularity as a means of community detection is that partitions of high modularity often exist even in randomly generated networks without underlying structure (see for example [2, 7]).

Zhang and Moore [57] were able to surmount several of the issues with modularity-based methods by treating modularity optimization in terms of the statistical physics of the spin-glass system with Hamiltonian $\mathcal{H} = -mQ(\{t_i\})$, where $\{t_i\} = [t_1, \ldots, t_N]$ with $t_i \in \{1, ..., q\}$ indicating the assignment of node i (of N) to one of q communities. As such, the distribution of states of the system is given by the Boltzmann distribution

$$P(\{t_i\}) \propto e^{-\beta\mathcal{H}(\{t_i\})} \tag{1.3}$$

[1] In the supra-adjacency representation, a single block diagonal matrix is used to represent all intralayer connections, each block representing a single-layer, with no connections between the blocks. A different matrix, C_{ij} encodes the interlayer connections. Note that $\dim(A_{ij}) = \dim(C_{ij}) = \dim(P_{ij})$.

where β represents the nondimensional inverse temperature that sets the sharpness of the energy landscape. Maximizing the *joint* distribution $P(\{t_i\})$ is then equivalent to globally optimizing modularity and identifying the ground state of the system. Instead of searching for a global modularity maximum, Zhang and Moore attempt to solve for the marginals of each node, $P(t_i = q)$, in the finite temperature regime. By looking for a "consensus of good partitions" rather than seeking a single "best" partition, the algorithm converges only if there is broad underlying structure within the network. If there exist several local maxima that are widely separated (*i.e.* uncorrelated) in the space of partitions, the belief propagation will oscillate between these partitions and fail to converge. In particular, Zhang and Moore demonstrated that the algorithm's convergence properties distinguish between synthetic networks with or without known underlying structure, even when the nominal modularity values of the identified partitions are quite similar. Maximization of the marginals has the additional benefit of producing a soft partition wherein nodes are partly assigned to multiple communities, thereby revealing which node labels the algorithm is most uncertain about.

Several tools are available to compute the marginals of Eq 1.3, including Markov Chain Monte Carlo sampling, Gibb's sampling, and the class of algorithms known as belief propagation (alternately, the cavity method) [30]. Belief propagation is a general algorithm for calculating the marginals of a joint distribution by a series of iterative updates. Belief propagation was initially developed for trees [39] for which it is an exact algorithm, but has been shown to provide good approximations on graphs with loops (*i.e.* "loopy" belief propagation) [40, 30]. Zhang and Moore apply belief propagation to modularity maximization, deriving update conditions for the node beliefs in terms of the message $\psi_t^{i\to k}$ from node i to k concerning community t that helps determine what node k "believes" its own community to be:

$$\psi_t^{i\to k} \propto \exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^\beta-1)\right)\right] \tag{1.4}$$

where m is the total number of edges; $d_i = \sum_j A_{ij}$ is the degree of node i; $\partial_i \setminus k$ is the neighborhood of node i except node k; and $\theta_t = \sum_i d_i\psi_t^i$ serves as a field-like term, approximating the null model contribution in modularity in terms of each node's belief about its own community, ψ_t^i, given by its marginal

$$\psi_t^i = \frac{1}{Z_i}\exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial_i}\log\left(1+\psi_t^{j\to i}(e^\beta-1)\right)\right] \tag{1.5}$$

where the belief includes contributions from all neighbors of node i, and $Z_i = \sum_s \psi_s^i$ is a normalization constant. That is, ψ_t^i is the properly normalized version of $\psi_t^{i\to i}$ insofar as i is already not a member of its own neighborhood, ∂_i, so there is no excluded element in the sum over j.

Fixed points of the "loopy" belief propagation algorithm are minimizers of the Bethe free energy

$$f_{\text{Bethe}} = -\frac{1}{N\beta}\left(\sum_{i\in\mathcal{V}}\log Z_i - \sum_{(i,j)\in\mathcal{E}}\log Z_{ij} + \frac{\beta}{4m}\sum_t\theta_t^2\right), \tag{1.6}$$

where $\mathcal{V}$ is the set of N nodes, $\mathcal{E}$ is the set of edges, and $Z_{ij} = \sum_{st} e^{\beta\delta_{st}}\psi_s^i\psi_t^j$ is the normalization constant for the pairwise joint marginals. Computing marginals for

each node, Zhang and Moore defined a "retrieval partition" assigning the community for each node according to its greatest marginal $t_i = \arg\max_t \psi_t^i$, with randomly broken ties. Retrieval modularity can be computed from the retrieval partition using Eq 1.1. We note that while this approach uses the modularity score to establish the energy landscape over which optimization is performed, ultimately the belief propagation minimizes the free energy; while lower free energy often corresponds to higher modularity for the retrieved partition, this relationship is in no way required and indeed is sometimes violated.

Our work introduces a belief propagation approach for the more general multilayer modularity framework, suitable for a variety of multilayer topologies. Specifically, we extend Zhang and Moore's *modbp* method in three ways. We explicitly allow weighted edges, which can greatly influence the communities detected (see [36]). We incorporate a resolution parameter γ into the modularity quality function as done in [44] and show that this can create a wider retrieval phase and achieve better performance in the case where the number of communities is not known *a priori*. Finally, we extend *modbp* to the multilayer modularity framework developed by Mucha *et al.* [33] and demonstrate the use of this tool on both synthetic and real world data. Our resulting method, *multimodbp*, can be used on both multilayer networks and single-layer networks. We have developed a *multimodbp* python package implementing our method in a fast, efficient manner that interfaces with other standard networks tools. (To our knowledge, the only other multilayer community detection method that incorporates belief propagation instead uses a specific version of the stochastic block model to capture temporal dynamics [14].)

The rest of this paper is organized as follows. In Section 2, we introduce the changes we have made to Zhang and Moore's belief propagation including the incorporation of the resolution parameter and the extension to the multilayer framework. Next, in Section 3, we show how incorporation of the resolution parameter can improve performance in the context of synthetic data as well as a real world network in the single-layer case. To demonstrate the ability of our approach to detect communities within the multilayer framework, we showcase *multimodbp* on two types of synthetic multilayer models with differing interlayer topology, and compare the performance of our model with another popular multilayer modularity based approach, *GenLouvain* [20]. Finally, we demonstrate the utility of our model on two real world datasets, showing how a belief propagation approach reveals an additional layer of information about a network's structure above and beyond other methods. We conclude with a brief discussion and remark on other possible improvements.

2. Methods.

2.1. Belief propagation update equations. We describe here the modifications to the *modbp* update equations used in *multimodbp*. Formal justification for these modifications is described in Section S.1 of the supplement. We have made the updates as follows. First, by incorporating a resolution parameter [44], γ, we effectively treat the field term and the edge term in the update equations as though they are at different temperatures. Second, we explicitly include the corresponding edge weights in the sum over contributions from neighbors of node i. Finally, to appropriately handle the null model in the multilayer modularity framework (see Eq 1.2), we have adapted the field term, θ_t^l to be layer specific and to only contribute to the beliefs originating from nodes within a given layer, l. We also introduce an additional interlayer contribution, scaled by interlayer coupling parameter ω, to account for interlayer edges in a manner similar to the intralayer contributions to multilayer

modularity, leading to the new update equation:

$$\psi_t^{i\to k} \propto \exp\left[\gamma\frac{\beta d_i}{2m_{l_i}}\theta_t^{l_i} + \sum_{j\in\partial_i\setminus k}\log\left(1+\psi_t^{j\to i}(e^{\tilde{A}_{ij}\beta}-1)\right)\right] \tag{2.1}$$

where l_i is the layer containing node i, (*i.e.* $i \in \mathcal{V}_{l_i}$), the field term $\theta_t^{l_i} = \sum_{j\in\mathcal{V}_{l_i}} d_j\psi_t^j$ and node strength $d_i = \sum_{j\in\mathcal{V}_{l_i}} A_{ij}$ only include contributions from layer l_i; the field term $\theta_t^{l_i} = \sum_{j\in\mathcal{V}_{l_i}} d_j\psi_t^j$ and node strength $d_i = \sum_{j\in\mathcal{V}_{l_i}} A_{ij}$ only include contributions from layer l_i; $\tilde{A}_{ij} = A_{ij}\delta(l_i,l_j) + \omega C_{ij}(1-\delta(l_i,l_j))$, with l_j the layer containing j, combines the intralayer and interlayer edges according to whether i and j are present in the same layer; and we again define ψ_t^i to be the normalized ($Z_i = \sum_s \psi_s^i$) version of $\psi_t^{i\to i}$ with $\partial_i \setminus i = \partial_i$ for the sum in Eq. 2.1. We note that the block description of A_{ij} and C_{ij} considered here makes the $\delta(\cdot,\cdot)$ indicators in $\tilde{A}_{ij}$ unnecessary; but we include them to help clarify the notation in terms of the layers containing i and j.

The solution to the above iterative equations is the minimizer of the following Bethe free energy equation, as derived in Section S.2 of the supplement:

$$f_{\text{Bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}}\log Z_{ij} + \sum_l \frac{\beta}{4m_l}\sum_t(\theta_t^l)^2\right) \tag{2.2}$$

where $Z_{ij} = \sum_{st} e^{\beta\delta_{st}}\psi_s^i\psi_t^j$ is the normalization factor for the pairwise joint marginals.

While we demonstrate *multimodbp* below in the context of the specific multilayer topology corresponding to a multiplex network, our formulation is flexible enough to handle any type of multilayer network consisting of two classes of edges (i.e., intralayer and interlayer edges). In particular, we remark that, similar to the weights in A_{ij}, the contribution from C_{ij} is explicitly included here, allowing for different interlayer weights. In principle, the method could also be extended to networks with multiple types of edges, such as encountered in representing network data that is both longitudinal and multiplex, with each new edge type introducing its own coupling parameter, ω_i.

2.2. Choice of β. By analyzing the linearized stability of the fixed point to small, uncorrelated perturbations, Zhang and Moore provided a heuristic for selecting the appropriate value of $\beta = \beta^*$ at which point the trivial, factorized solution ($\psi_t^{j\to i} = 1/q$ for all beliefs) is no longer stable. If a retrieval phase exists (i.e. there is detectable community structure), running the algorithm with $\beta = \beta^*$ should converge if the network has a reasonable degree distribution. If significant structure is not present within the network, for values of $\beta > \beta^*$ the algorithm enters the 'spin-glass' phase in which convergence never occurs. Practically this can be used to eliminate or at least reduce one of the free parameters involved in running the algorithm. We provide a similar heuristic approach for selecting an appropriate value of β^* in the multilayer setting that accounts for edge weights.

The linear stability of the factorized solution is characterized by the derivatives of the messages with respect to each other at the fixed point ($1/q$). To identify β^*, the critical value for instability with respect to random, uncorrelated perturbations, we linearize the *multimodbp* update equations (Eq 2.1) and then analyze the stability of the equations under repeated iteration. We use the notation from Zhang and Moore. However, in our case, the linearized form depends on whether the incoming message,

$\psi_s^{j\to i}$ travels along an intralayer or interlayer edge:

$$\left.\frac{\partial \psi_t^{i\to k}}{\partial \psi_s^{j\to i}}\right|_{1/q} = \begin{cases} T_{st} & \text{if } (i,j) \in \mathcal{E}_{intra} \\ R_{st} & \text{if } (i,j) \in \mathcal{E}_{inter} \end{cases} \tag{2.3}$$

where $\mathcal{E}_{intra}$ and $\mathcal{E}_{inter}$ are the sets of intralayer ($l_i = l_j$) and interlayer ($l_i \neq l_j$) edges, respectively, and T_{st} and R_{st} are the intralayer and interlayer linearization matrices, respectively (see Supplement Equation S.22). Linearizing about the factorized fixed point $\psi_t^{i\to j} = \frac{1}{q}$ yields

$$\psi_t^{i\to k} \propto \sum_{s,j} \psi_s^{j\to i} \left[T_{st}\delta(l_i,l_j) + R_{st}(1-\delta(l_i,l_j))\right] . \tag{2.4}$$

In Supplementary Section S.4, we derive the form for the linearization matrices T_{st} and R_{st} and show that their largest eigenvalues are given by

$$\lambda_T = \frac{e^{\beta}-1}{q+e^{\beta}-1} \qquad \text{and} \qquad \lambda_R = \frac{e^{\omega\beta}-1}{q+e^{\omega\beta}-1} . \tag{2.5}$$

In the linearized approximation, interlayer edges have the same form as if they were weighted edges in a single-layer network (with weight ω). This observation led us to posit that we could establish a similar stability condition for the multilayer case by looking at the stability of the trivial solution on weighted single-layer networks. In the case where a fixed, weight w is applied to each edge uniformly, we have the following stability condition:

$$c\lambda_w^2 \leq 1 , \quad \text{where} \quad \lambda_w = \frac{e^{w\beta}-1}{q+e^{w\beta}-1} . \tag{2.6}$$

where c is the average excess degree of the network. Solving this equation at equality for $\beta^*(c,q,w)$ gives:

$$\beta^*(c,q,w) = \frac{1}{w}\log(\frac{q}{\sqrt{c}-1}+1) . \tag{2.7}$$

In practice we have found that for randomly distributed weights on the edges, one can modify Equation 2.7 to use the average weight on the network $\langle w\rangle$ and still robustly identify the retrieval phase (see Supplement S.5) :

$$\beta^*(c,q,w) = \frac{1}{\langle w\rangle}\log(\frac{q}{\sqrt{c}-1}+1) . \tag{2.8}$$

In the case of the multilayer networks, we calculate the average edge strength $\langle w\rangle$, treating the coupling strength ω as the weights for the interlayer edges. We have found that this heuristic works well in identifying values of β for which our method converges. We note that β^* represents the stability of the solution for uncorrelated perturbations in the beliefs. In the case when detectable community structure exists, the messages become correlated with each other and the transition from the trivial paramagnetic phase to the retrieval phase is generally lower than β^* [57]. Thus, choosing values of β near β^* works well in practice. Additionally, Schülke *el al.* showed that in many networks there can be multiple zones of the retrieval phase corresponding to

detecting communities at different scales [47]. Therefore, in our experiments, we run the algorithm for a range of $\{\beta_i^*\} = [\beta^*(c, q = 2), \ldots \beta^*(c, q = q_{\max})]$, where $q_{\max}$ is some reasonable upper limit for the number of communities in a particular network. We emphasize that like the original Zhang and Moore approach, our heuristic assumes randomly distributed edges and weights and provides no guarantees that β^* will be found within the retrieval phase. For certain networks, scanning a larger range of β will be necessary, though in practice we have found that the approach above is fairly robust.

2.3. Selection of number of communities, q. One critical issue with many community detection algorithms is in selecting the appropriate number of communities. In the context of modularity, adjusting the resolution parameter γ can reveal communities of different scale and size, overcoming the "resolution limit of detection" first raised by [12]. Since then there have been several approaches showing how the scale of the community structures identified varies with the resolution parameter (see, e.g., the discussion and references in [54])

Zhang and Moore do not include a resolution parameter in deriving their *modbp* algorithm (thereby implicitly setting $\gamma = 1$ in Eq 1.4), instead suggesting an alternative approach for selecting the appropriate number of communities. They show in several examples that the maximum modularity achieved in the retrieval phase of the algorithm peaks at certain numbers of communities. They suggest that this peak identifies the correct value for q, the number of communities, where there is no additional increase in the retrieval modularity $Q(\{t_i\})$. However, this approach requires running *modbp* for many possible values of q, and then choosing an arbitrary threshold when modularity is no longer sufficiently increasing to establish the correct value of q. In many cases, selecting an exact value of q is made difficult because of fluctuations in the retrieval modularity near the β^* value derived by Zhang and Moore. Figure S.13 in the Supplement illustrates how choosing q is challenging in practice by these considerations. Meanwhile, selecting the number of communities in this manner implicitly uses the value $\gamma = 1$, which has been shown to return non-ideal partitions in synthetic and real-world networks (see, e.g., [1, 12, 37, 51]). We show in Section 3 the positive impact of using different values for γ on several different networks.

There have been two other approaches to selecting the appropriate number of communities using *modbp* without having to run the algorithm at many values of q. Both approaches involve selecting a $q_{\max}$, the largest possible number of communities, and then using similarities in the marginal probabilities of assignments to evaluate the true number of communities. Lai *et al.* [24] noted that in the event that q is too large, many of the marginal community assignments will be highly correlated, and highly correlated states (community assignments) can be condensed into a single group. Similarly, Ref. [47] condenses the community assignments on the basis of the average distance between the marginals across all nodes in the network. In practice, we have found that for the default resolution ($\gamma = 1$), choosing the number of communities this way all but obliterates the retrieval phase if $q_{\max}$ is chosen to be too much larger than the actual number. We have implemented the method in Ref. [47], letting the number of communities float up to a pre-specified $q_{\max}$ (See Section 2.2), and condensing together communities that have closely aligned marginals. We show that incorporation of a resolution parameter γ restores the width of the retrieval phase and returns values closer to the correct number of communities. As previously mentioned, because we do not specify a single value of q; rather, we run the algorithm across a

range of $\beta = [\beta^*(c, q = 2), \ldots, \beta^*(c, q = q_{\max})]$ where the formula for $\beta^*(c, q)$ is given in Eq. 2.8. We have found that this provides a reasonable range of β values to search within and that performance of the algorithm does not depend on the precise value of β, as long as it is within the retrieval phase.

2.4. Assessing Partition Alignment with AMI. We use the information theoretic measure Adjusted Mutual Information (AMI) [52] throughout our analysis to assess the agreement between the predicted partition and either (1) the known underlying truth in the case of the generative models tested, or (2) relevant metadata for real world networks. Mutual information measures how much entropy or uncertainty is removed from one variable by observation of another. The adjusted mutual information specifically measures the overlap between two partitions, with a value of 1 representing perfect agreement and a value of 0 representing overlap no better than expected under random chance.

2.5. Cross-layer community alignment. In running *multimodbp* at low levels of interlayer coupling (ω) on multilayer networks with a temporal coupling topology (e.g the dynamic stochastic block model described in Section 3.2.1), we frequently observed that the intralayer marginals would rapidly converge to communities that remained misaligned between layers. Such misalignment would then typically lead to "fragmented" partitions as shown in Figure S.15. For these partitions, within any single-layer the AMI of the partition with the ground truth with that layer would be very high, but the total AMI over the entire multilayer data would become much lower. To correct for this issue, we implemented a greedy heuristic to explicitly permute the community assignments within certain layers in order to maximize local alignment between neighboring layers. Specifically, we identify the layer x that has the greatest number of nodes (of those present in both layers) that change community identity from the previous layer, y. We then find the matching of community labels in x that best matches those observed in y; that is, we minimize the total number of mismatches across layers x and y:

$$C(x, y) = \sum_{i \in x} \sum_{j \in y} [(c_i \neq c_j) \wedge \mathbb{I}((i, j) \in \mathcal{E}_{inter})] \tag{2.9}$$

Once the optimal bipartite matching has been identified [23], the community labels in layer x and every subsequent layer are rearranged according to that matching (with community labels in subsequent layers that are not present in either layer x or y remaining unchanged). We then repeat this procedure until no further labels are changed (*i.e.* the optimal matching is the identity at the layer where the greatest change occurs). This heuristic is similar to the *interlayer merging* developed by Bazzi *et al.* to overcome a similar problem encountered when optimizing multilayer modularity with the GenLouvain algorithm [3].

We note that this procedure does not alter the community structure identified within any particular layer, maintaining nodes that have been grouped together. Rather, this procedure aligns the community labels between layers in a way that always increases the retrieval modularity, thereby improving the computed results. This approach assumes a notion of persistent community across inherently *ordered* layers which is appropriate in the temporal multilayer setting. We do not require a strict one-to-one mapping of node identities between layers. However, this matching approach does require an inherent ordering of layers across the multilayers. If an unordered interlayer coupling is defined (e.g., categorically multiplex connections), this matching feature can be disabled.

3. Results.

3.1. Single-layer networks. We begin by examining how our modifications affect the ability of *modbp* to detect communities within synthetically generated data in the single-layer case. For single-layer networks, our method is equivalent to Zhang and Moore's except two main differences (see also **2. Methods**): First, we have included a resolution parameter γ that adjusts the relative balance of the terms in the update equation. Like other implementations of modularity, this effectively controls the size of the identified partitions. Second, we have set an upper limit $q_{\max}$ on the number of communities and incorporated the approach from [47] to select an effective number of communities based on the overlap of the marginals (see Section 2.3).

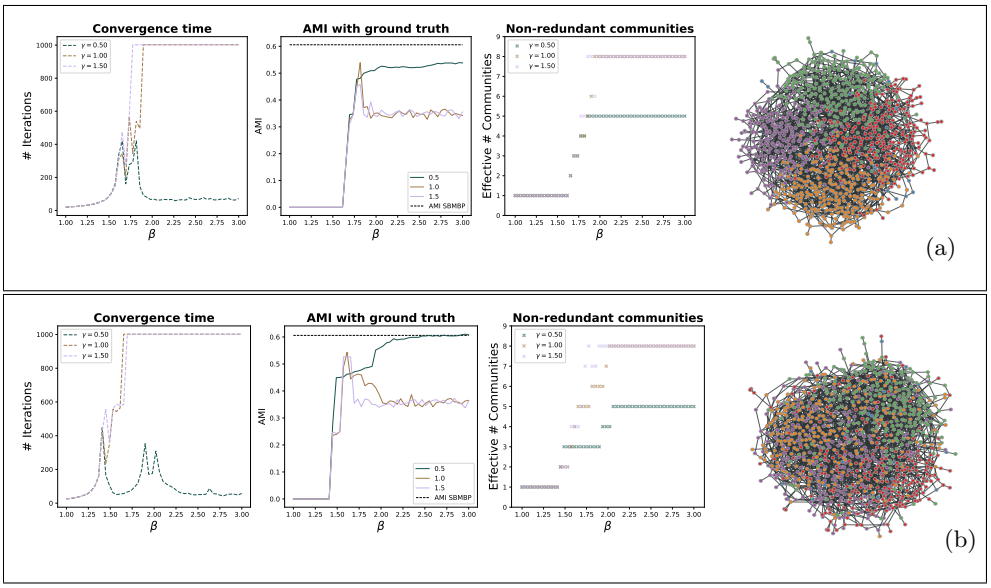


Fig. 1: Demonstration of *multimodbp* on two realizations of the SBM model. From left to right, the plots show the retrieval modularity, number of iterations to convergence, and the AMI of the retrieval partition with known community assignments and the effective number of communities. **(a)** 4 community SBM with $n = 1000$, $\epsilon = \frac{p_{\text{out}}}{p_{\text{in}}} = .1$, $c_{\text{avg}} = 4$, and even community sizes and **(b)** and 4 community SBM with $n = 1000$, $\epsilon = .1$, $c_{avg} = 4$, with uneven community sizes ($\nu = [300, 200, 300, 200]$). For each network we also show the performance of the *sbmbp* with parameters for the SBM supplied (middle plot, dotted black line)

3.1.1. Single-layer stochastic block model. We examine the behavior of *multimodbp* on instances of a four-community stochastic block model for different values of the resolution parameter γ. First, we show that in the setting with several smaller communities, a higher value of γ produces a much wider retrieval phase and thus makes detection of communities more robust to selection of β. To investigate this robustness, we generated a single realization of an SBM and scanned a range of β values to characterize the behavior of the algorithm seen in **Figure 1**. For an SBM network with four even-sized communities, **Figure 1a** shows that the retrieval phase for both $\gamma = 1.0$ and $\gamma = 1.5$ are very narrow (leftmost panel) with a small corresponding peak in the AMI of detected communities (middle panel). In contrast, for $\gamma = 0.5$ the retrieval phase widens out with a broader and higher set of AMI values

for the detected communities. Furthermore, the number of communities identified for $\gamma = 0.5$ plateaus at 5, which is more closely aligned with the underlying model (actually there are only 4 main communities detected as shown in far right panel of **Figure 1a**)

We also tested the performance of the algorithm in the case where the sizes of the planted communities were uneven, shown in **Figure 1b**. The relative performance for varying γ is even more disparate in this case. There is a small retrieval phase for $\gamma = 1$, but it is much smaller than that of $\gamma = 0.5$ and the AMI is again consistently lower. For $\gamma = 0.5$ we actually detect two retrieval phases. In the first retrieval phase ($\beta \in [1.4, 2.0]$), only nodes within the two larger communities are labeled correctly. Then, as β increases ($\beta \in [2.0, 2.75]$), the smaller two communities also become identifiable. This is consistent with the multiphase behavior observed in [47], though we note that in their example, the phase transition is observed for the default value of $\gamma = 1$. In both of these examples the AMI of the identified partition by *multimodbp* is close to the result achieved by a belief propagation implementation of the SBM model, which has been shown to achieve the optimal bounds for this model [8, 9].

3.1.2. Comparison of *multimodbp* with SBMBP on LFR benchmark networks. We compare the performance of our algorithm *multimodbp*, with a belief propagation approach to fit the Stochastic Block Model (SBM) developed and implemented in Ref. [8], which we refer to as *sbmbp*. This Expectation-Maximization (EM) implementation of *sbmbp* alternates between iteratively updating the marginals using belief propagation with fixed SBM parameters, and updating the SBM parameters using likelihood maximization for the fixed marginals. Their implementation requires setting a fixed q however, so for testing we ran *sbmbp* across a range of q values ($q \in 2, 3, ..., 8$) and selected the partition with the lowest free energy density.

Our test dataset is the Lancichinetti-Fortunato-Radicchi (LFR) benchmark generator [27], an algorithm developed to generate networks with more diverse community structures. We tested our *multimodbp* with several values of the resolution parameter γ against *sbmbp* across a range of parameters of the LFR model. We vary the LFR mixing parameter μ, which sets the detectability of the underlying communities. The LFR algorithm also has a parameter $\hat{\gamma}$ to set the exponent of the power law for the degree distribution and a parameter $\hat{\beta}$ to set the exponent of the community size distribution. We tested both algorithms for two sets of $(\hat{\gamma}, \hat{\beta})$ in **Figure 2**.

Figure 2 shows that the modularity based approach outperforms the stochastic block model across a range of μ, the mixing parameter, all the way down to the detectability limit. The flexibility of the modularity approach allows for better identification of communities with for real world degree distribution (since the classic SBM assume homogenous degree distribution within a community). The comparison was done using *sbmbp*'s EM approach which is not well suited to determine the number of communities. In contrast, using our approach as described in Section 2.3, the *multimodbp* algorithm was able to identify the correct number of communities and get more accurate community assignment using a resolution parameter value of $\gamma = 0.5$ (though other values of γ also performed well).

3.1.3. NCAA Division I-A College Football network. We now demonstrate that inclusion of the resolution parameter γ in the modularity objective function can significantly improve performance on real-world networks. As an example of a real-world network with stable community structure we selected the 2000-2001 NCAA Division I-A College football network, which has 115 nodes representing teams (schools) and 613 unweighted edges connecting teams that played at least one

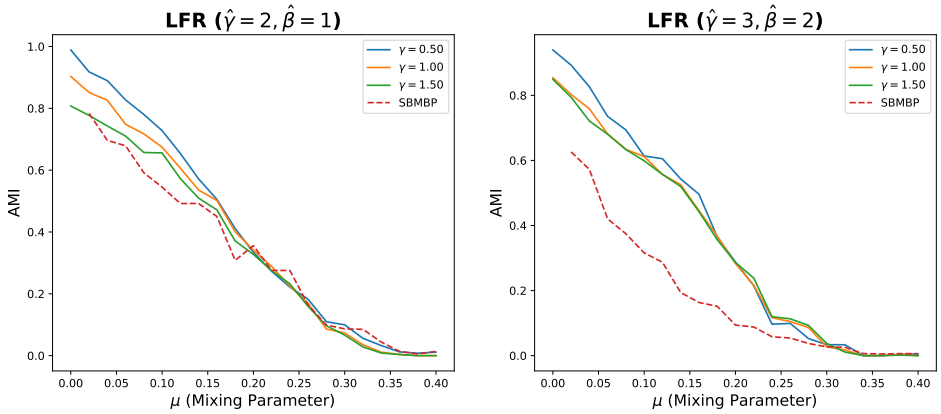


Fig. 2: Performance of *multimodbp* and *sbmbp* over many LFR benchmark realizations with a range of values for the mixing parameter μ. Each point represents an average over 100 realizations of LFR with 1000 nodes, an average degree of 3 (with a max of 10), and other parameters set to default values.

game [10, 15]. Our previous work suggests that modularity optimization produces the best community partition in a range $\gamma \in [1.4, 4]$ [54, 55].

To investigate how the value of γ affects the retrieval phase, we ran *multimodbp* for a range of values of the parameter γ and examined the minimal number of iterations for which non-trivial structure was identified, shown in **Figure 3**. For each value of γ, *multimodbp* was run over 30 evenly-spaced values over $\beta \in [0.5, 4.5]$. For each value of γ we show the minimum number of iterations over all values of β for which non-trivial structure was identified and the AMI of the partition of the corresponding partition (the partition identified with the minimum number of iterations). Runs that did not converge after 500 iterations suggest that for that value of γ the retrieval phase was either very small or nonexistent. It is possible that a retrieval phase exists outside the chosen range for β, though we verified for a few arbitrary values of γ that the algorithm did not converge across a much wider range. Furthermore, **Figure 3** demonstrates that the AMI of the retrieval partition increases as a function of γ from $\gamma = 1$ up until it plateaus from $\gamma = [1.7, 3.4]$ at a stable 11 community partition (shown in the far right panel). In **Figure 4**, we show the algorithm convergence properties as well as performance for a few values of γ on this network. We also compare the performance of the *multimodbp* algorithm with the *sbmbp* approach, showing that even when the SBM approach identifies the correct number of communities (middle panel dashed line), *multimodbp* still achieves more accurate identification of the underlying community structure (right panel).

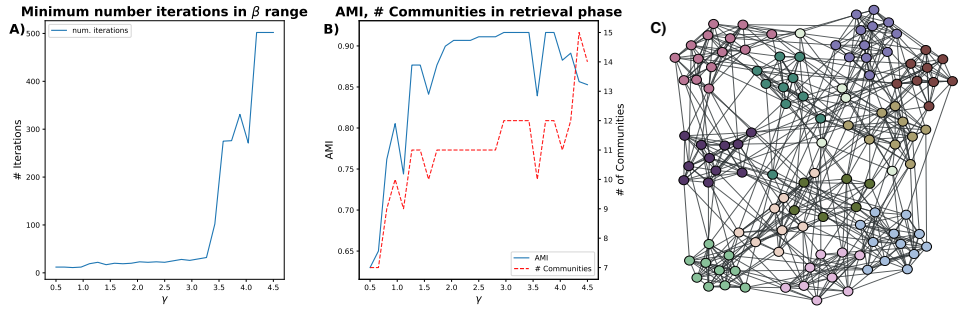


Fig. 3: Testing *multimodbp* on the 2000-2001 Division I-A College Football network [10, 15]. **A)** The average number of iterations until convergence in the retrieval phase across a range of γ values. **B)** The average number of communities detected in the retrieval phase as γ increases and the corresponding adjusted mutual information (AMI) of those partitions. **C)** ForceAtlas2 [19] layout of the football network with each node colored according to a partition identified using $\gamma = 3.0$, demonstrating excellent alignment to the conference structure.

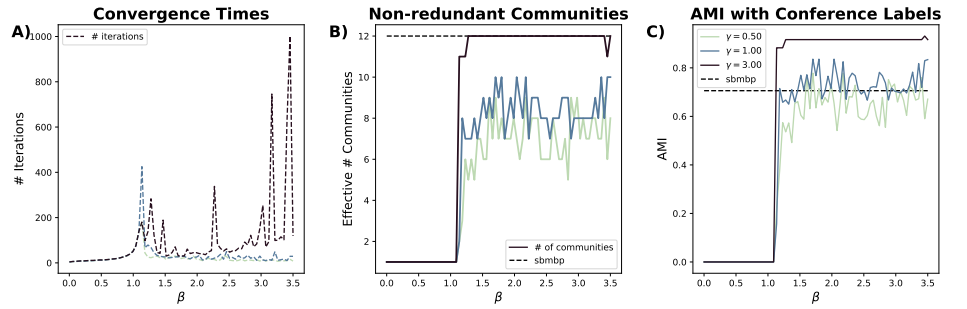


Fig. 4: We show the performance characteristics of the algorithm for 3 different values of γ on the 2000-2001 NCAA Division I-A College football network **A)** Although all three values of γ produce a wide retrieval phase, the communities identified within each retrieval phase are different. **B)** The number of non-redundant communities is higher as γ increases with $\gamma = 3$ producing the number of communities that lines up well with the ground truth (the conferences) in this example, with **C)** showing corresponding higher values of AMI for $\gamma = 3$. Horizontal black dashed line shows that *sbmbp* identifies correct number of communities in **B)** but has less agreement with the known conference structure of the network.

3.2. Multilayer modularity belief propagation.

3.2.1. Dynamic stochastic block model. We test the multilayer functionality of *multimodbp* by application to a multilayer SBM called the dynamic stochastic block model (DSBM) as described in [14]. In the DSBM, each layer is drawn from a regular stochastic block model with q communities and edge probabilities described by probabilities p_{in} within communities and p_{out} between communities. Each node's community assignment has a fixed probability η of remaining the same between subsequent layers (and $1 - \eta$ probability of choosing a new community). Conditioned on the node community assignments, each layer's edges are independent of all other layers. For a fixed average degree c, the strength of community structure within each

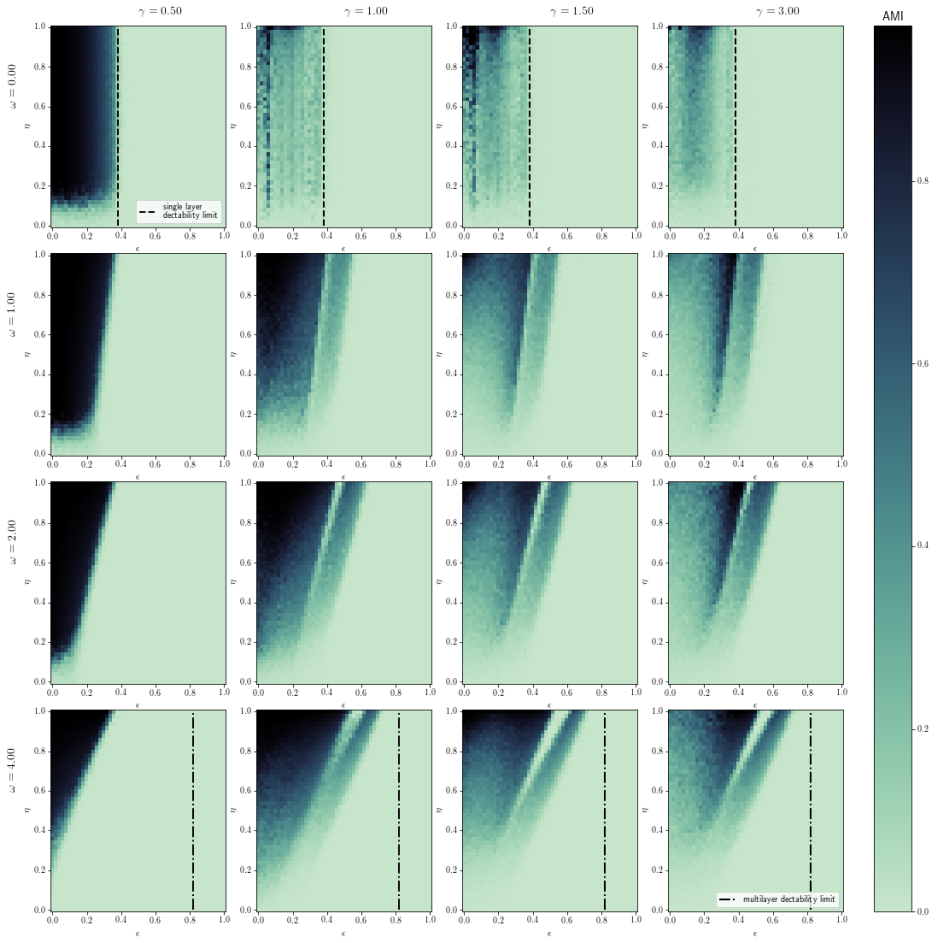


Fig. 5: Accuracy of the multilayer modbp algorithm on a multilayer SBM for different values of model parameters ϵ, and η (x and y axes respectively) and for *multimodbp* parameters γ and ω (moving horizontally and vertically vertically across panels). For these generated networks, $N = 250$, $n_{layers} = 20$, $c = 10$, and $q_{true} = 2$.

layer is given by the parameter $\epsilon = p_{\text{out}}/p_{\text{in}}$. The DSBM represents a temporal multilayer network where each entity in the network is represented by a single node in each layer. The correspondence between identified nodes is represented by a single interlayer edge between adjacent layers.

In Figure 5 we show the average adjusted mutual information (AMI) score of the *multimodbp* algorithm on the dynamic stochastic block model for a range of parameters. We consider DSBM networks created using values of ϵ and η ranging from 0 to 1. For each choice of ϵ and η we created 50 networks and computed the AMI between partitions identified using *multimodbp* and the ground truth. Because the value of q is usually not known beforehand, for each (γ, ω) point we scan a range of possible values of β^* corresponding respectively to possible values of q as given by Eq 2.8 with $q_{\max} = 4$ set to twice the true number of communities (2). For each trial, we select the partition with the highest retrieval modularity among all that converged.

We apply the *multimodbp* algorithm in this analysis with several choices of the

resolution parameter, γ (columns in Figure 5) and coupling parameter, ω (rows of Figure 5). Figure 5 shows that incorporation of a resolution parameter makes a large difference for detectability of community structure based on the DSBM parameters used to generated the network. For lower values of ϵ (i.e., increased intralayer community signal), $\gamma = 0.5$ clearly outperforms the higher values of γ, especially when frequent community switching is occurring (lower η values). Additionally, within the first column we see that incorporation of information across layers increases detectability of communities except for the lowest values of η.

Within the middle and right columns ($\gamma = 1.0, 1.5$), we observe that an additional regime of detectability is created at higher values of ϵ, seen in the band at around $\epsilon = .4$ for the top middle panel ($\gamma = 1.00, \omega = 0.00$), as well as the top right panel. This furthest band is consistent with the limit of detectability in the single-layer network given by the condition $N(p_{\text{in}} - p_{\text{out}}) > q\sqrt{c}$ in [9] and [34] (depicted by the vertical dashed line in upper row of Figure 5).

By increasing ω, the AMI along this band is increased, and the range for detection is increased to a maximum of $\epsilon = .75$ for $\eta = 1.0$ (no community switching). This behavior is consistent with the limits of detectability that are achieved through aggregation of layers as discussed in [50]. They derive a modified limit of detectability in the case where each layer is drawn from the same SBM with the community labels fixed throughout the layers, unlike our model where each nodes' community assignment is allowed to vary. They compute a detectability threshold of $NL(p_{\text{in}} - p_{\text{out}}) = \sqrt{4NL\rho(1-\rho)}$ where $\rho = \frac{1}{2}(p_{\text{in}} + p_{\text{out}})$. For parameters used in this experiment the theoretical detectability limit is $\epsilon \approx .82$ (shown by the dashed lines in Figure 5). Thus the additional flexibility in our approach allows for achieving near optimal performance depending on the parameters of the underlying model.

3.2.2. Comparison with GenLouvain on Multiplex LFR Benchmark Networks. We next illustrate the effectiveness of our method by showing that it outperforms the commonly used GenLouvain [20] modularity optimization algorithm on a multilayer version of the Lancichinetti-Fortunato-Radicchi (LFR) benchmark networks [27]. In this generative model of multilayer networks, each node is connected to its corresponding node in adjacent layers through an interlayer edge. Community assignments persist through a given number of layers, and then are reassigned at random, using a power-law distribution of community sizes (see Figure 6.B for schematic representation of multilayer structure). Given the community assignments, each layer represents an independent realization of the original LFR benchmark network [27].

Figure 6.B shows a comparison of *multimodbp* and GenLouvain (both using parameters $\gamma = 1.0$ and $\omega = 1.0$) while varying the mixing parameter, μ of the LFR Multiplex model. (See Figure S.14 for performance over a range of γ and ω values.) Each point represents the average over 100 trials on different realizations of the model. For each trial, GenLouvain was run iteratively to match the number of β values run, with the previous community identified used as the starting partition for the next run, and using the "moverandw" setting. The partition that maximized modularity was used to calculate the final values for each trial. Figure 6.A demonstrates several benefits of using the belief propagation approach over the direct modularity maximization scheme in GenLouvain. The *multimodbp* algorithm consistently achieves a higher agreement with the ground truth all the way up to the value of μ where neither algorithm is aligned with the known structure. Furthermore, by looking at the convergence of the algorithm (indicated by blue x's), *multimodbp* gives an indication of whether any significant community structure exists in the network at all. When

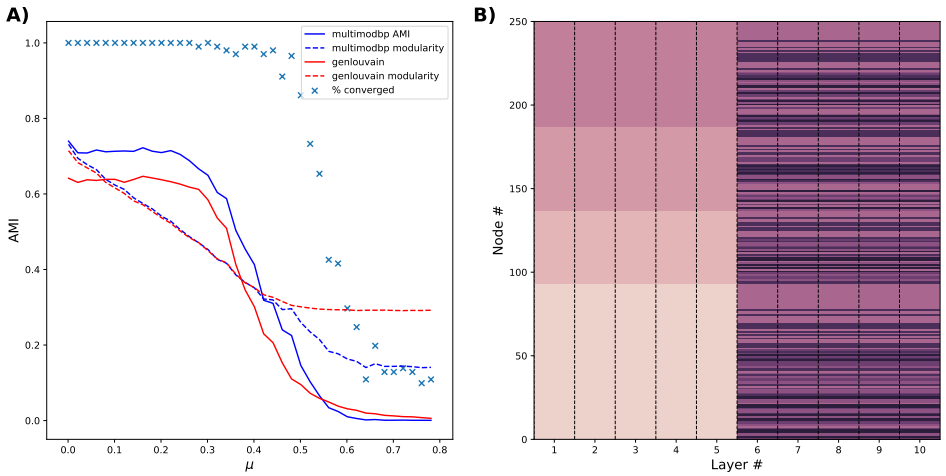


Fig. 6: **A)** Performance of *multimodbp* and GenLouvain [20] on multiplex LFR benchmark realizations with a range of values for the mixing parameter μ. Both algorithms were run at $(\gamma = 1, \omega = 1.0)$. Each point represents an average over 100 realizations of a multilayer LFR with 250 nodes in each layer, 10 different layers, and 2 different community assignments, each persisting 5 layers. The average degree was set to 10 (with a max of 20), and the maximum and minimum possible community sizes were 100 and 30 nodes respectively. Other LFR parameters were set to default values $(\hat{\gamma} = 2.0, \hat{\beta} = 1.0)$. Solid lines represent the AMI of detected communities with ground truth, while dashed lines represent the modularity of the identified communities. Blue lines are the results for *multimodbp* and red lines are from GenLouvain. **B)** Representation of the community structure of an example realization of the LFR multiplex benchmark. Original communities are drawn and persist for 5 layers, and then new community assignments are made for last five layers, resulting in a discrete break in the structure.

the *multimodbp* partition stops aligning with the ground truth partition ($\mu \geq 0.4$), *multimodbp* stops converging, showing that it is not identifying a meaningful partition. In contrast, the value of modularity for GenLouvain (dashed red line) only declines slightly far past the point where the performance is no better than random. We see that *multimodbp* is able to avoid overfitting on random structure because the node marginals only converge when there is enough correlation between possible high modularity partitions. GenLouvain on the other hand is usually able to find at least one high modularity partition even in the absence of real community structure.

3.2.3. Real world multilayer networks. We conclude our results by demonstrating the inferences that can be made on real-world networks using *multimodbp*.

We begin with the US Senate voting similarity network as introduced by [53] and analyzed in [32]. This dataset represents the voting similarity patterns of 1,884 U.S. Senators over 110 Congresses starting in 1789. Each 2-year Congress beginning in the January following an election is represented as a layer within this network. A node within a layer represents a Senator serving in that Congress with Senators serving in consecutive Congresses linked through interlayer edges. In the analysis performed here, the network was modified to sparsify the intralayer connections by taking the K-nearest neighbors (KNN) of each Senator based on voting correlations (using K=10)

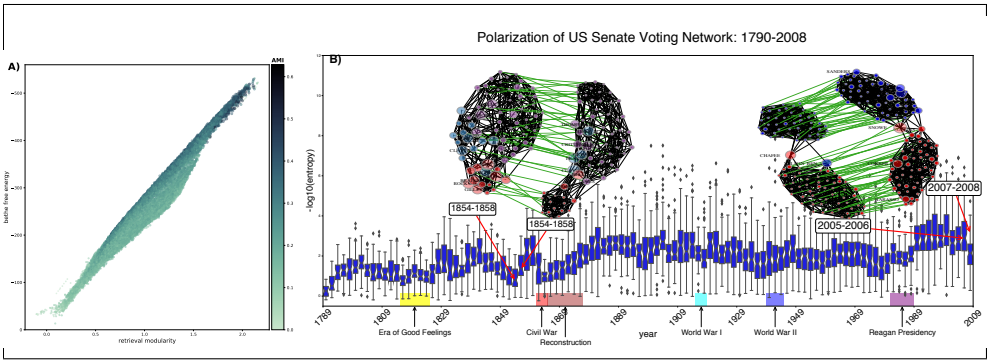


Fig. 7: We ran *multimodbp* on the US Senate voting similarity network comprised of 1884 Senators across the first 110 Congresses [32, 53]. **A)** The relationship between the retrieval modularity (x-axis) and the Bethe free energy is given by equation Eq 2.2. The Bethe free energy correlates strongly with modularity of a partition, and the partitions with the lowest free energy tend to correspond best with the underlying party structure. **B)** We examined the distribution of the average Senator entropy for each Congress (layer) in the network. Inset graphs depict how changes in average entropy correspond with network structure and the overall level of polarization within the network. Node size depicts the average entropy level of Senators with "high entropy" Senators labeled.

while keeping the edges with the original weights based on voting correlations.

In **Figure 7**.A we show the correspondence between the retrieval modularity, the Bethe free energy (Eq 2.2), and the AMI with party labels of partitions identified across a range of the (γ, ω) parameter space. Each point represents a partition identified using *multimodbp*. The belief propagation algorithm fixed points are actually minimizers of the Bethe free energy (rather than optimizers of the retrieval modularity). We see in general that partitions that minimize the Bethe free energy produce high retrieval modularities. Optimizing the Bethe free energy also produces partitions that accurately reflect the underlying known structure in the data set (*i.e.*, the political party affiliations of the Senators), shown by the color of the scatter points in Figure 7. We show a comparison of these partitions with the real party layouts in Figure S.17. It appears that the most appropriate choice (in this sense) of the *multimodbp* parameters are around $(\omega = 6, \gamma = 0.5)$.

One of the main benefits of using the belief propagation approach for community detection is that we can obtain a measure of how confident we are in the predicted community for each node. In Figure 7.B, we show the distribution of Senator entropies for each Congress, averaged over the top 200 partitions identified (by AMI with parties). On the y-axis we plot the distribution of $-\log_{10}(\text{entropy})$ across all Senators as a measure of how strongly identified the communities are and thus how polarized Congress is along party lines. We have highlighted several periods of American history such as the *Era of Good Feelings* with corresponding low polarization/high entropy, or the high level of polarization immediately preceding the Civil War. The inset shows how the corresponding changes in entropy from Congress to Congress are reflected in the community structure of the graph. This is consistent with the increasing level of polarization identified by Moody *et al.* in their study of this data set[31]. Our method gives the further benefit of providing a node level metric to identify how strongly a node is connected with its community. In Figure 7.B we have

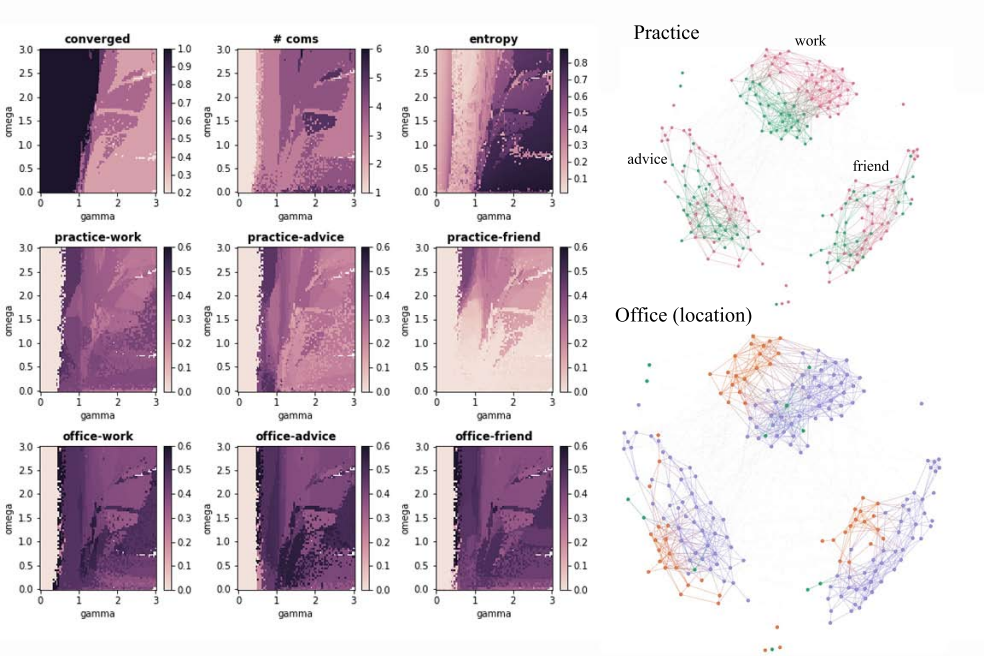


Fig. 8: Several visualizations of the Lazega Lawyer network [28]. On the right we show several characteristics of partitions identified with *multimodbp* at various values of γ (x-axes) and ω (y-axes). In the top row, from left to right, we show how many times the algorithm converged over 10 runs at different β values, the number of communities identified by the best run for each set of parameters (based on lowest Bethe free energy), and the average entropy of the marginals across all of the nodes for each of these partitions. In the next two rows we show the AMI of the identified partition within a single-layer and a specified metadata attribute. For example in the left most panel of the second row, we show how the "practice" (which type of law practiced by each node) attribute lines up with the partitioning of work layer. To the right in **B)** we show the three layers of the network (advice, work, friends) colored by two of the metadata attributes, practice (which specialty of law each person is involved in) and office (which is the location the person works in). Showing the partitions in this manner demonstrates how different metadata attributes affect the community structure in the different layers and how this is best captured by *multimodbp* for different values of γ and ω.

labeled the "high entropy" Senators, those whose voting patterns indicate a measure of bipartisanship (or independence from the party as in the case of Bernie Sanders in the 2007–08 Congress).

The second real-world network that we have analyzed is the Lazega Lawyer network introduced by [28]. We scan the (γ, ω) parameter range $[0, 3] \times [0, 3]$ and select the partition with the greatest retrieval modularity ($Q(\{\hat{t}\})$) at each point. In Figure 8, we show the number of iterations taken by the converged partitions for different parameter choices of (γ, ω). Within the lower right quadrant (high γ, low ω) the algorithm only converged for a small range of β values. In the top row, middle panel, we see that for this network three communities were chosen for a large portion of the parameter space, although the structure of the identified partitions varied widely. In the top right panel of Figure 8.A, we look at the average entropy per node across the parameter space to identify regions where node ambiguity is minimized. These suggest

another way to identify regions of the (ω, γ) with corresponding strong community structure. We see that there are a few partitions with quite low entropy for $\gamma < 1$, and that average entropy tends to increase past this threshold. The region where the algorithm converged for very few values of β (lower right corner) also tends to have the highest entropy. In the bottom two rows we have explored how each partition overlays with a particular metadata attribute within a given layer. For instance the panel titled "office-friends" shows the AMI of all partitions with the office attribute only within the friends layer. We see that within different parts of the parameter space, different features of the metadata align more closely with the partitions identified. For instance there is a narrow band from $\gamma \in (0.5, 0.9)$ for which the practice attribute strongly aligns with the community structure of the advice network. For higher values of γ the advice layer switches to being more aligned with the office metadata. Similarly, within the "work" layer, we see that the practice attribute contributes most significantly to the structure identified at a similar $\gamma \in (0.5, 0.9)$ regime, however the office attribute actually contributes more at higher γ and lower ω. Our results for this network complement those derived in [41] suggesting that no single metadata attribute explains the structure of this network. These results highlight the need to explore and summarize partitions across different parameter ranges.

4. Discussion. We have presented *multimodbp*, an extension of the modularity-based belief propagation framework to multilayer modularity. Like the original belief propagation framework for modularity [57], there are a number of features of *multimodbp* that make it a useful tool for identifying community structure in real-world multilayer networks. At its core, modularity and its multilayer extension are objective functions for assessing community structure and do not allow for true statistical inference (cf. generative approaches like the stochastic block model, e.g., [16, 42, 49, 50] for example). However, by formulating multilayer modularity optimization from the perspective of a Boltzmann ensemble, we can obtain an estimate of the uncertainty of assignment at each node from its marginal. The marginals reflect how much shifting a node from one community to another changes the modularity and thus is a measure of how strongly a node prefers a certain community. In this sense we can find a "soft" partitioning of the nodes, in which one node may belong to multiple communities, along with confidence levels corresponding to each community. Most modularity-based algorithms do not allow for overlapping communities with a few notable exceptions including OverMod [4] and the fuzzy c-means [58], both of which require an initial disjoint partitioning of the network in order to identify overlaps. Other versions of overlapping modularity-like approaches include [17, 29]. Our approach is useful in that it can be used for either a hard or soft partitioning of the network depending on the desired context.

Meanwhile, although the method of Zhang and Moore allows for the selection of the number of communities by identifying the value of q for which the retrieval modularity plateaus [57], we have shown that this approach fails to perform optimally in a number of cases. This underscores the need for greater flexibility as provided by incorporation of the resolution parameter γ. Rather than searching along the domain of q, we allow q to float (up to a certain point $q_{\max}$) and search along the γ domain to characterize network structure. The flexibility added by the resolution parameter becomes even more important in the multilayer context. We have shown that performance of *multimodbp* is optimized by different combinations of (γ, ω) in different parameter regimes of the dynamic stochastic block model. This is consistent with the work of Newman who demonstrated a link between the resolution parameter γ of

modularity and the p_{in} and p_{out} parameters of the degree-corrected stochastic block model [37]. Recently, Pamfil *et al.* extended this approach to multilayer modularity, deriving a similar mapping between the coupling parameter, ω, and the parameters of a model very similar to the DSBM studied here [38].

One of the greatest benefits of the *multimodbp* approach is that the convergence of the algorithm to non-trivial solutions reveals the existence of significant community structure above what would be expected at random. Several prior works have shown that even in randomly-generated networks without underlying structure, modularity optimization heuristics are capable of finding high-modularity partitions. For this reason alone we believe an extension of modularity belief propagation for multilayer networks provides a valuable new tool for network analysis.

There remain a number of technical challenges for implementing *multimodbp* at scale. The runtime of the algorithm depends greatly on the number of iterations of belief propagation that are required to run. As described in Zhang and Moore, this tends to spike as you approach the retrieval phase, and their formula for β^* which we have modified tends to yield values slightly above where this spike occurs. Ideally, one could have an adaptive solution, identifying a value of β for which the algorithm appears to be converging quickly early on and adjusting β once the algorithm is closer to converging. Eventually, we would like to devise an automatic method of selecting an appropriate value for β based on a preliminary scan of convergence rates across the β domain, similarly to how we iteratively select the appropriate number of communities as the algorithm runs. Another issue is the dependency of the runtime and memory of the algorithm on the number of marginals being optimized. We try to reduce the dimension of the marginals after the algorithm has run, by attempting to combine redundant dimensions (those that are highly correlated). One could imagine attempting such a reduction earlier on after a few course-grained runs of the algorithm to produce additional performance gains.

To facilitate use of (and possible improvements on) our method, we have written and distributed a Python package available on PyPI [56].

REFERENCES

[1] A. Arenas, A. Fernández, and S. Gómez, *Analysis of the structure of complex networks at different resolution levels*, New Journal of Physics, 10 (2008), p. 053039.

[2] J. P. Bagrow, *Communities and bottlenecks: trees and treelike networks have high modularity*, Physical Review E, 85 (2012), p. 066118.

[3] M. Bazzi, M. Porter, S. Williams, M. McDonald, D. Fenn, and S. Howison, *Community Detection in Temporal Multilayer Networks, with an Application to Correlation Networks*, Multiscale Modeling & Simulation, 14 (2016), pp. 1–41.

[4] L. Bennett, A. Kittas, S. Liu, L. G. Papageorgiou, and S. Tsoka, *Community Structure Detection for Overlapping Modules through Mathematical Programming in Protein Interaction Networks*, PLOS ONE, 9 (2014), p. e112821.

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast unfolding of communities in large networks*, Journal of Statistical Mechanics: Theory and Experiment, 2008 (2008), p. P10008.

[6] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, *On modularity clustering*, IEEE Transactions on Knowledge and Data Engineering, 20 (2007), pp. 172–188.

[7] F. de Montgolfier, M. Soto, and L. Viennot, *Asymptotic modularity of some graph classes*, International Symposium on Algorithms and Computation, (2011), pp. 435–444.

[8] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, *Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications*, Physical Review E, 84 (2011), p. 290.

[9] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, *Inference and Phase Transitions*

in the Detection of Modules in Sparse Networks, Physical Review Letters, 107 (2011), p. 065701.
[10] T. S. Evans, *Clique graphs and overlapping communities*, Journal of Statistical Mechanics: Theory and Experiment, 2010 (2010), p. P12037.
[11] S. Fortunato, *Community detection in graphs*, Physics Reports, 486 (2010), pp. 75–174.
[12] S. Fortunato and M. Barthelemy, *Resolution limit in community detection*, Proceedings of the National Academy of Sciences, 104 (2007), pp. 36–41.
[13] S. Fortunato and D. Hric, *Community detection in networks: A user guide*, Physics Reports, 659 (2016), pp. 1–44.
[14] 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), p. 031005.
[15] M. Girvan and M. E. J. Newman, *Community structure in social and biological networks.*, Proceedings of the National Academy of Sciences, 99 (2002), pp. 7821–7826.
[16] Q. Han, K. S. X. 0001, and E. M. Airoldi, *Consistent estimation of dynamic and multi-layer block models.*, ICML, (2015).
[17] T. C. Havens, J. C. Bezdek, C. Leckie, K. Ramamohanarao, and M. Palaniswami, *A Soft Modularity Function For Detecting Fuzzy Communities in Social Networks*, IEEE Transactions on Fuzzy Systems, 21 (2013), pp. 1170–1175.
[18] P. W. Holland, K. B. Laskey, and S. Leinhardt, *Stochastic blockmodels: First steps*, Social Networks, 5 (1983), pp. 109–137.
[19] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, *ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software*, PLOS ONE, 9 (2014), pp. 1–12.
[20] L. G. Jeub, M. Bazzi, J. S. Inderjit, and P. J. Mucha, *A generalized Louvain method for community detection implemented in MATLAB*, URL http://github.com/GenLouvain/GenLouvain, (2011–19).
[21] B. Karrer and M. E. Newman, *Stochastic blockmodels and community structure in networks*, Physical Review E, 83 (2011), p. 016107.
[22] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, *Multilayer networks*, Journal of Complex Networks, 2 (2014), pp. 203–271.
[23] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistics Quarterly, 2 (1955), pp. 83–97.
[24] D. Lai, X. Shu, and C. Nardini, *Correlation enhanced modularity-based belief propagation method for community detection in networks*, Journal of Statistical Mechanics: Theory and Experiment, 05 (2016), pp. 053301–.
[25] R. Lambiotte, J.-C. Delvenne, and M. Barahona, *Laplacian dynamics and Multiscale Modular structure in Networks*, arxiv.org, 1 (2009), pp. 76–90.
[26] R. Lambiotte, J.-C. Delvenne, and M. Barahona, *Random Walks, Markov Processes and the Multiscale Modular Organization of Complex Networks*, IEEE transactions on network science and engineering, 1 (2014), pp. 76–90.
[27] A. Lancichinetti, S. Fortunato, and F. Radicchi, *Benchmark graphs for testing community detection algorithms*, Physical Review E, 78 (2008), p. 046110.
[28] E. Lazega and others, *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*, Oxford University Press on Demand, 2001.
[29] J. Liu, *Fuzzy modularity and fuzzy community structure in networks*, The European Physical Journal B, 77 (2010), pp. 547–557.
[30] M. Mézard and A. Montanari, *Information, physics, and computation*, Oxford University Press, 2009.
[31] J. Moody and P. J. Mucha, *Portrait of political party polarization*, Network Science, 1 (2013), pp. 119–121.
[32] P. J. Mucha and M. A. Porter, *Communities in multislice voting networks*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 20 (2010), p. 041108.
[33] 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.
[34] R. R. Nadakuditi and M. E. J. Newman, *Graph Spectra and the Detectability of Community Structure in Networks*, Physical Review Letters, 108 (2012), p. 188701.
[35] M. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Physical Review E, 69 (2004), p. 268.
[36] M. E. J. Newman, *Analysis of weighted networks*, Physical Review E, 70 (2004), p. 056131.
[37] M. E. J. Newman, *Equivalence between modularity optimization and maximum likelihood meth-*

ods for community detection, Physical Review E, 94 (2016), p. 052315.
[38] A. R. Pamfil, S. D. Howison, R. Lambiotte, and M. A. Porter, *Relating modularity maximization and stochastic block models in multilayer networks*, arXiv, abs/1804.01964 (2018).
[39] J. Pearl, *Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach*, in Proceedings of the Second AAAI Conference on Artificial Intelligence, Pittsburgh, Pennsylvania, 1982, pp. 133–136.
[40] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, (1988).
[41] L. Peel, D. B. Larremore, and A. Clauset, *The ground truth about metadata and community detection in networks*, Science Advances, 3 (2017), p. e1602548.
[42] T. P. Peixoto, *Nonparametric Bayesian inference of the microcanonical stochastic block model*, Physical Review E, 95 (2017), p. 012317.
[43] M. A. Porter, J.-P. Onnela, and P. J. Mucha, *Communities in networks*, Notices of the American Mathematical Society, 56 (2009), pp. 1082–1097.
[44] J. Reichardt and S. Bornholdt, *Statistical mechanics of community detection*, Physical Review E, (2006).
[45] M. Rosvall and C. T. Bergstrom, *Maps of random walks on complex networks reveal community structure*, Proceedings of the National Academy of Sciences, 105 (2008), pp. 1118–1123.
[46] M. T. Schaub, J.-C. Delvenne, M. Rosvall, and R. Lambiotte, *The many facets of community detection in complex networks*, Applied Network Science, 2 (2017), p. 4.
[47] C. Schülke and F. Ricci-Tersenghi, *Multiple phases in modularity-based community detection*, Physical Review E, 92 (2015), p. 042804.
[48] S. Shai, N. Stanley, C. Granell, D. Taylor, and P. J. Mucha, *Case studies in network community detection*, arXiv, (2017).
[49] N. Stanley, S. Shai, D. Taylor, and P. J. Mucha, *Clustering network layers with the strata multilayer stochastic block model.*, IEEE transactions on network science and engineering, 3 (2016), pp. 95–105.
[50] D. Taylor, S. Shai, N. Stanley, and P. J. Mucha, *Enhanced Detectability of Community Structure in Multilayer Networks through Layer Aggregation*, 116 (2016), p. 228301.
[51] V. A. Traag, G. Krings, and P. Van Dooren, *Significant scales in community structure*, Scientific reports, 3 (2013), p. 75.
[52] N. X. Vinh, J. Epps, and J. Bailey, *Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance*, Journal of Machine Learning Research, 11 (2010), pp. 2837–2854.
[53] A. S. Waugh, L. Pei, J. H. Fowler, P. J. Mucha, and M. A. Porter, *Party Polarization in Congress: A Network Science Approach*, arXiv.org, (2009), https://arxiv.org/abs/0907.3509v3.
[54] W. H. Weir, S. Emmons, R. Gibson, D. Taylor, and P. Mucha, *Post-Processing Partitions to Identify Domains of Modularity Optimization*, Algorithms, 10 (2017), p. 93.
[55] W. H. Weir, R. Gibson, and P. J. Mucha, *CHAMP package: Convex Hull of Admissible Modularity Partitions in Python and MATLAB*, 2017, https://github.com/wweir827/CHAMP.
[56] W. H. Weir and B. Walker, *multimodbp*, 2018, https://github.com/bwalker1/ModularityBP_Cpp.
[57] P. Zhang and C. Moore, *Scalable detection of statistically significant communities and hierarchies, using message passing for modularity*, Proceedings of the National Academy of Sciences, 111 (2014), pp. 18144–18149.
[58] S. Zhang, R.-S. Wang, and X.-S. Zhang, *Identification of overlapping community structure in complex networks using fuzzy c-means clustering*, Physica A: Statistical Mechanics and its Applications, 374 (2007), pp. 483–490.

S. Supplement.

S.1. Derivation of Multilayer Belief Propagation Update Equations. To derive the update equations for the multilayer belief propagation, we have relied heavily on the approach employed by Zhang and Moore [57]. In the belief propagation algorithm, also known as the sum-product algorithm or cavity method, each node sends a "message" ($\psi_t^{i\to k}$) to its neighboring nodes encoding the marginal probability of it occupying a given state, t (or, in our case, belonging to a given community). These updates are iterated over all nodes until the messages converge to a fixed point. The form of the message for a particular node in the single layer case from Zhang and Moore is given by:

$$\psi_t^{i\to k} = \frac{1}{Z_{i\to k}} \prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{\beta\delta_{st}} \psi_s^{j\to i} \prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{-\beta(d_i d_j/2m)\delta_{st}} \psi_s^{j\to i}\,. \tag{S.1}$$

We have made several changes to this equation to extend the belief updates to modularity on multilayer networks. We incorporate the weights of the edges between nodes i and j by changing the first sum-product term:

$$\prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{\beta\delta_{st}} \psi_s^{j\to i} \Rightarrow \prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{\beta\tilde{A}_{ij}\delta_{st}} \psi_s^{j\to i} \tag{S.2}$$

Where l_i indicates the layer associated with node-layer i and $\tilde{A}_{ij} = A_{ij}\delta(l_i,l_j) + \omega C_{ij}(1-\delta(l_i,l_j))$ applies the correct weight based on whether (i,j) is an intralayer or interlayer edge. Likewise, we incorporate the resolution parameter γ by modifying the part of the update equations that derives from the null model term in the modularity equation:

$$\prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{-\beta(d_i d_j/2m)\delta_{st}} \psi_s^{j\to i} \Rightarrow \prod_{j\in\partial i\setminus k} \sum_{s=1}^{q} e^{-\beta\gamma(d_i d_j/2m)\delta_{st}} \psi_s^{j\to i} \tag{S.3}$$

Zhang and Moore derive the final form of the messages used in the algorithm by approximating the dense interactions between every pair of nodes given by the null model with a field approximation that is holds when the graph is sparse. Derivation of our intralayer term in 2.1 follows the exact same steps. In the multilayer case however, the contribution to the field term of each outgoing belief comes only from node-layers that are within the same layer as the source node-layer for the belief. We therefore introduce a layer specific field term $\exp\{\frac{\beta d_i}{2m_l}\theta_t^{l_i}\}$ Where $\theta_t^{l_i} = \sum_{j\in\mathcal{V}_l} d_j\psi_t^j$ only depends on contributions from nodes in layer l_i, node strength $d_i = \sum_{j\in\mathcal{V}_{l_i}} A_{ij}$ only includes contributions from layer l_i, and $m_l = \sum_{j\in\mathcal{V}_{l_i}} d_j$ is the total edge strength in layer l_i.

These modification give us 2.1. We note that the message passed from node-layer i to node-layer k, $\psi_t^{i\to k}$ does *not* depend on the type of edge (i,k). Node i integrates information from its neighboring nodes (except node k), handling both edge weights and types appropriately, and passes this information to node k. The edge type (and weight) between node i and node k only comes into play when node k integrates all the information coming in from its neighboring nodes.

S.2. Derivation of Bethe Free Energy. We derive here the formula for the free energy of the single layer model given in Zhang and Moore [57]. In the next

section we will show how this naturally extends to the multilayer case with interlayer edges. For any model which has only pairwise interactions, the formula for the Bethe free energy approximation is given by

$$f_{\text{bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}} \log Z_{ij}\right) . \tag{S.4}$$

In the modularity model, there are really two types of edge interactions: those that are given explicitly by the underlying graph (i.e. the $A_{i,j}\delta c_i, c_j$ term), and the pairwise interaction term that comes from the null model (i.e. $P_{i,j} = \frac{k_i,k_j}{2m}\delta_{c_i,c_j}$). We can split these two apart:

$$f_{\text{bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}} \log Z_{ij} - \sum_{i\neq j} \log \hat{Z}_{ij}\right) \tag{S.5}$$

where we refer to the edges in the underlying graph as $\mathcal{E}$ and split out the non-edge interactions into another term with normalization $\hat{Z}^{ij}$. We write out the joint distribution for the "non-edges":

$$\psi^{ij}_{st} = \frac{1}{\hat{Z}_{ij}} e^{-\beta(d_i d_j/2m)\delta_{st}} \psi^i_s \psi^j_t \tag{S.6}$$

We use this to compute $\hat{Z}_{ij}$:

$$\hat{Z}_{ij} = \sum_t \sum_s e^{-\beta(d_i d_j/2m)\delta_{st}} \psi^i_s \psi^j_t \, , \tag{S.7}$$

$$\begin{aligned}
\sum_{i<j} \log \hat{Z}_{ij} &= \sum_{i<j} \log \sum_t \sum_s e^{-\beta(d_i d_j/2m)\delta_{st}} \psi^i_s \psi^j_t \\
&\approx \sum_{i<j} \log \left(\sum_t \sum_s 1 - \beta(d_i d_j/2m)\delta_{st}\psi^i_s\psi^j_t\right) \\
&\approx \sum_{i<j} \left(\sum_t \sum_s -\beta(d_i d_j/2m)\delta_{st}\psi^i_s\psi^j_t\right) \\
&= \sum_t \sum_{i<j} -\beta(d_i d_j/2m)\psi^i_t\psi^j_t \\
&= -\frac{\beta}{4m}\sum_t \sum_{i\neq j} d_i d_j \psi^i_t \psi^j_t \\
&= -\frac{\beta}{4m}\sum_t \theta_t^2 \, .
\end{aligned} \tag{S.8}$$

This gives us the expected full formula,

$$f_{\text{bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}} \log Z_{ij} + \frac{\beta}{4m}\sum_t \theta_t^2\right) \tag{S.9}$$

S.3. Multilayer Bethe Free Energy. We now extend the Bethe Free Energy equation to multilayer networks. The formula for multilayer modularity for undirected networks is given by Equation 1.1 in main text:

$$Q(\gamma,\omega) = \sum_{i,j} \left(A_{ij} - \gamma P_{ij} + \omega C_{ij}\right) \delta(c_i, c_j) \tag{S.10}$$

As before we only have pairwise interactions within the model. However, note that in the multilayer formulation there are now both intra- and interlayer edges. We can split the edge term $\mathcal{E}$ of f_{bethe} into the contributions from interlayer and the intralayer edges:

$$\sum_{i,j\in\mathcal{E}} \log Z_{ij} = \sum_{i,j\in\mathcal{E}_{inter}} \log Z_{ij}^{inter} + \sum_{i,j\in\mathcal{E}_{inter}} \log Z_{ij}^{intra}\,. \tag{S.11}$$

Where $\mathcal{E}_{intra}$ and $\mathcal{E}_{inter}$ are given by the non-zero elements of A_{ij} and C_{ij} respectively. For the non-edge term in the multilayer case, we note that the non-edge interaction terms are all restricted to within a given layer. This means that nodes within different layers of the model only interact through the interlayer edge term and not through the null model term:

$$\frac{1}{N\beta}\sum_{i\neq j} \log \hat{Z}_{ij} = \frac{1}{N\beta}\sum_{l}\sum_{i\neq j, i,j\in l} \log \hat{Z}_{ij}^{l} \tag{S.12}$$

We can therefore split this term into a sum over the contributions from each of the layers with a similar form as from before:

$$\sum_{l}\sum_{i\neq j, i,j\in l} \log \hat{Z}_{ij}^{l} = -\sum_{l} \frac{\beta}{4m_l} \sum_{t} (\theta_t^l)^2 \tag{S.13}$$

and we can write the full Bethe free energy as

$$f_{\text{bethe}} = -\frac{1}{N\beta}\left(\sum_i \log Z_i - \sum_{i,j\in\mathcal{E}_{inter}} \log Z_{ij}^{inter} - \sum_{i,j\in\mathcal{E}_{inter}} \log Z_{ij}^{intra} + \sum_l \frac{\beta}{4m_l}\sum_t (\theta_t^l)^2\right) \tag{S.14}$$

where the Z_{ij}^{inter} can be computed from the pairwise marginals of the interlayer interactions:

$$\psi_{s,t}^{i,j} = \frac{1}{Z_{ij}^{inter}} e^{\beta\omega\delta_{s,t}} \psi_s^{i\to j} \psi_t^{j\to i}\,. \tag{S.15}$$

S.4. Formula for selection of β^*. Consider the update equation:

$$\psi_t^{i\to k} = \frac{1}{Z}\exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k} \log\left(1 + \psi_t^{j\to i}\left(e^{\beta} - 1\right)\right)\right] \tag{S.16}$$

We compute the derivative $\frac{\partial \psi_t^{i\to k}}{\partial \psi_s^{j\to i}}$ assuming both (i,j) and (i,k) are edges:

$$\begin{aligned}\frac{\partial \psi_t^{i\to k}}{\partial \psi_s^{j\to i}} &= \frac{\partial}{\partial \psi_s^{j\to i}}\left[\frac{1}{Z}\exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right]\right] \\ &= \frac{1}{Z}\frac{\partial}{\partial \psi_s^{j\to i}}\exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right] \\ &\quad + \exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right]\frac{\partial}{\partial \psi_s^{j\to i}}\left[\frac{1}{Z}\right]. \end{aligned} \tag{S.17}$$

We will consider each of these two derivatives separately. First,

$$\begin{aligned}&\frac{\partial}{\partial \psi_s^{j\to i}}\exp\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right] \\ &= \exp\left[...\right]\frac{\partial}{\partial \psi_s^{j\to i}}\left[\frac{\beta d_i}{2m}\theta_t + \sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right]. \end{aligned} \tag{S.18}$$

The derivative of the first term here,

$$\frac{\partial}{\partial \psi_s^{j\to i}}\frac{\beta d_i}{2m}\theta_t$$

is $O\left(\frac{d_i d_j}{2m}\right)$, which we can ignore given our assumption that the network is sparse ($d_i \ll \sqrt{m}$ for all i).

We are then left with

$$\frac{\partial}{\partial \psi_s^{j\to i}}\left[\sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right].$$

The only term in this sum that will lead to a non-zero derivative is if $s=t$, leading to

$$\frac{\partial}{\partial \psi_s^{j\to i}}\log\left(1+\psi_s^{j\to i}\left(e^\beta-1\right)\delta_{st}\right) = \delta_{st}\frac{e^\beta-1}{1+\psi_s^{j\to i}\left(e^\beta-1\right)}. \tag{S.19}$$

Evaluating at the fixed point, and combining with the previous $\frac{1}{Z}\exp(...) = \frac{1}{q}$ this term becomes

$$\delta_{st}\frac{e^\beta-1}{q+e^\beta-1} \tag{S.20}$$

Next we move on to the second term from the previous product rule expansion (Eq S.17):

$$\frac{\partial}{\partial \psi_s^{j\to i}}\frac{1}{Z} = -\frac{1}{Z^2}\frac{\partial Z}{\partial \psi_s^{j\to i}} = -\frac{1}{Z^2}\frac{\partial}{\partial \psi_s^{j\to i}}\left[\sum_t \exp\left(\sum_{j\in\partial i\setminus k}\log\left(1+\psi_t^{j\to i}\left(e^\beta-1\right)\right)\right)\right]$$

where we have ignored the θ term as before. This leads to

$$= -\frac{\exp(...)}{Z^2} \frac{e^\beta - 1}{1 + \psi_s^{j\to i}(e^\beta - 1)} \,. \tag{S.21}$$

If we bring the extra exp(...) from before back in, and evaluate at the fixed point, this leads to

$$= -\frac{1}{q} \frac{e^\beta - 1}{q + e^\beta - 1} \,. \tag{S.22}$$

So in total, we find

$$T_{st} = \frac{e^\beta - 1}{q + e^\beta - 1} \left(\delta_{st} - \frac{1}{q} \right) \tag{S.23}$$

exactly as expressed in Zhang and Moore.

We see that the form of the linearization of the interlayer messages, R_{st}, at the factorized fixed point proceeds along the exact same lines above, with the only difference being that β has been replaced with $\beta\omega$, and gives us

$$R_{st} = \frac{e^{\omega\beta} - 1}{q + e^{\omega\beta} - 1} \left(\delta_{st} - \frac{1}{q} \right) \,.$$

S.5. Selection of β^* on weighted networks. In the case where there are weights on the edges, the linearized approximation of the messages around the factorized solution depends on the value of each edge weight, w_{ij}:

$$\frac{\partial \psi_s^{i\to j}}{\partial \psi_t^{k\to i}} = T_{st}^{ij} = \frac{e^{w_{ij}\beta} - 1}{q + e^{w_{ij}\beta} - 1} \left(\delta_{st} - \frac{1}{q} \right)$$

with corresponding eigenvalue:

$$\lambda_{w_{ij}} = \frac{e^{w_{ij}\beta} - 1}{q + e^{w_{ij}\beta} - 1} \,.$$

We do not attempt to derive a rigorous stability condition based on the distribution of weights across the networks. However, we have found that in practice using the average weight on the edges, $\langle w \rangle$ with the originally derived stability criteria yields reasonable values of β for running the algorithm:

$$c\lambda_{\langle w \rangle}^2 > 1$$

gives us the following stability condition:

$$\beta^*(c, q, w) = \frac{1}{\langle w \rangle} \log\left(\frac{q}{\sqrt{c} - 1} + 1\right) . \tag{S.24}$$

As part of testing the formula for $\beta^*(c, q, w)$, we look at the effect of adding normally distributed edges weights on an Erdös-Rényi graph shown in Figure S.9. For the Erdös-Rényi graph with normally distributed weights, Equation 2.8 gives a very good estimate of where the divergence occurs, while the unmodified equation becomes less and less accurate for higher weights.

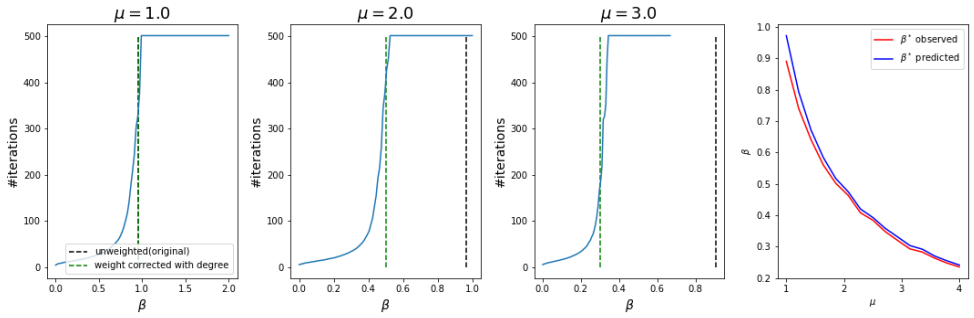


Fig. S.9: Stability boundary for Erdös-Rényi graph with weights assigned randomly from a $\mathcal{N}(\mu, \sigma = .5)$ normal distribution. Left three plots depict convergence curves of the algorithm for three different means of the normally distributed edge weights (μ =1,2, and 3 respectively). Each curve represents the average over 10 realizations of the ER random graph. The unweighted prediction for β^* is given by the black dashed line, while the weight adjusted prediction is given by the dashed green line. On far right plot β^* was empirically determined for several different mean weights (red line) and compared with the predicted values (blue line) showing good agreement.

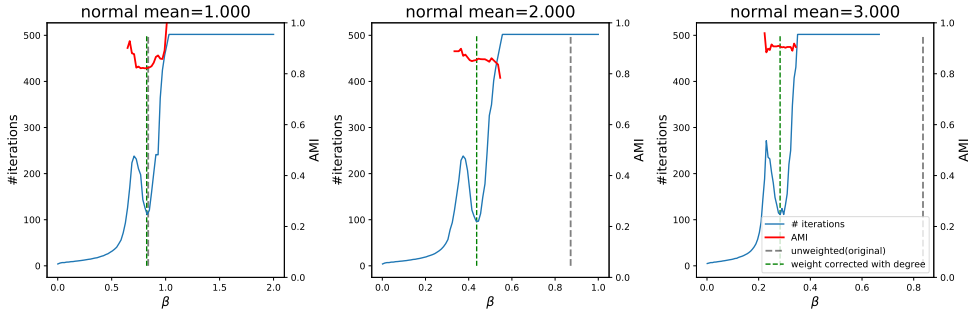


Fig. S.10: Stability boundary for 2 community stochastic block model graph with weights assigned randomly from a $\mathcal{N}(\mu, \sigma = .5)$ normal distribution. SBM's had $n = 200$ nodes with mean degree, $c = 6$, and $\epsilon = \frac{p_{out}}{p_{in}} = .1$. Each convergence curve was averaged over 10 realizations of the SBM model with different means of the normally distributed edge weights (μ =1,2, and 3 respectively). The unweighted prediction for β^* is given by the black dashed line, while the weight adjusted prediction is given by the dashed green line. Red curve shows the adjusted mutual information with the underlying ground truth. On far right plot β^* was empirically determined for several different mean weights (red line) and compared with the predicted values (blue line) showing good agreement.

Below in Figure S.10, we also demonstrate that for a 2 community SBM the modified formula for β^* occurs within the retrieval phase.

We have used Equation 2.8 to identify the value of β to run the algorithm at in all of the experiments within this manuscript. Since *a priori* the number of communities, q, isn't known in advance, we run the algorithm at several values

$\beta = [\beta^*(q = 2, c, \langle w \rangle), ...\beta^*(q = q_{max}, c, \langle w \rangle)]$ for a range of expected numbers of communities, $[2, q_{max}]$. For each run of the algorithm for a given value of β we do not reset the marginals, thus reducing the time until convergence, once we have found the retrieval phase. We reiterate that the heuristic derived works well in most cases, but makes no guarantees that β^* will be inside the retrieval phase for all degree distributions and distribution of edge weights. For some networks scanning a range of β values might be required.

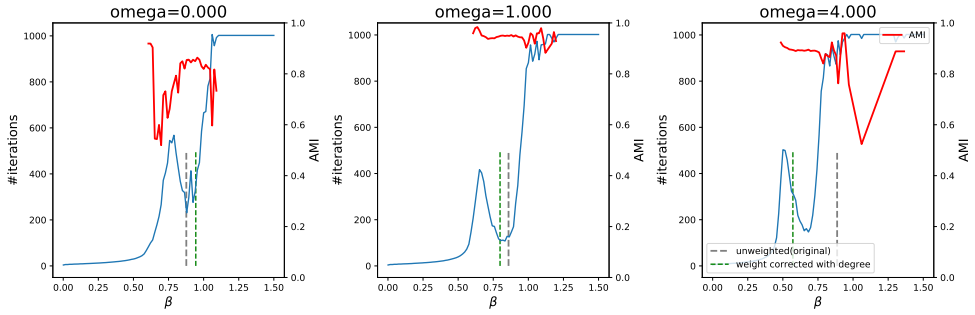


Fig. S.11: Stability boundary for 2 community unweighted multilayer dynamic stochastic block model graph. Network had $n = 100$ node within each layer with mean degree $c = 6$ and $\epsilon = \frac{p_{out}}{p_{in}} = .1$. Each convergence curve was averaged over 10 realizations of the SBM model with the algorithm run with different interlayer edge couplings (ω =0, 1, and 2 respectively). The unweighted prediction for β^* is given by the black dashed line, while the weight adjusted prediction is given by the dashed green line. Red curve shows the adjusted mutual information with the underlying ground truth. In the far right plot β^* was empirically determined for several different mean weights (red line) and compared with the predicted values (blue line) showing good agreement.

In Figure S.11, we also show that the retrieval phase of multilayer networks also varies with the strength of the coupling parameter, ω. The β^* predicted by Equation 2.8 consistently lies within the retrieval phase even as ω increases (in contrast to the value of β given from the unmodified equation).

S.6. Supplement Figures.

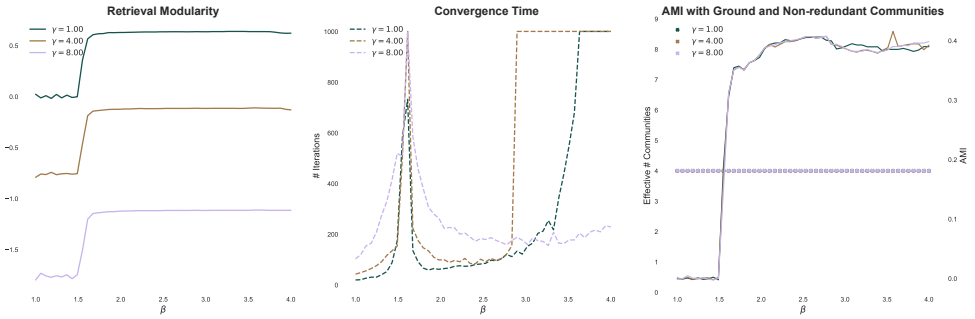


Fig. S.12: We compare the performance of the algorithm for a wide range of γ values in the event that the number of communities is fixed at the correct number ($q = 4$). Here we do not allow q to float as described in Section 2.3

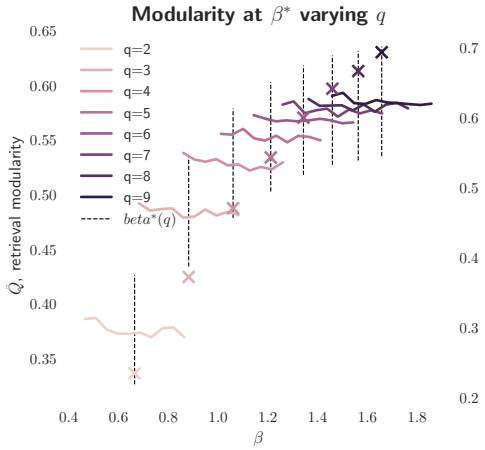


Fig. S.13: Using the method recommended by Zhang and Moore to select the appropriate value of q for the American NCAA Div-IA College Football Network [15, 10]. Each colored line corresponds to running *modbp* for a given value of q across a window of β around $\beta^*(q)$ (shown by black dashed line). Using this method would suggest an appropriate $q \in [6-8]$ depending on the threshold selected. We note that here, we do not collapse community labels as described in Section 2.3; for each run a single fixed value of q is used as well as the default resolution ($\gamma = 1$).

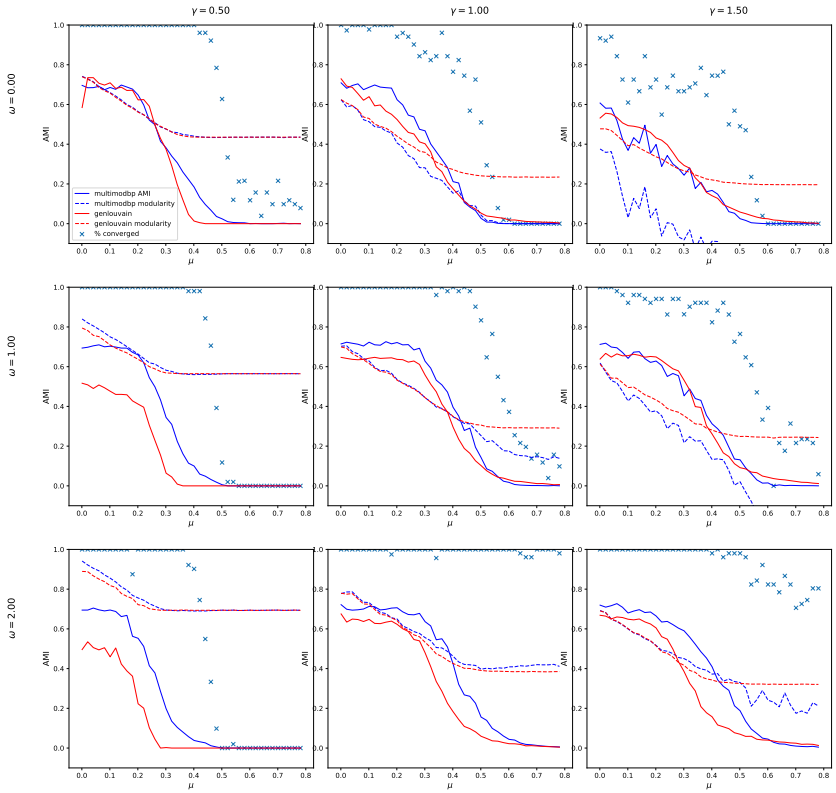


Fig. S.14: We compare *multimodbp* with GenLouvain using the multiplex LFR benchmark network (see Section 3.2.2 in main manuscript) across a range of γ and ω parameters.

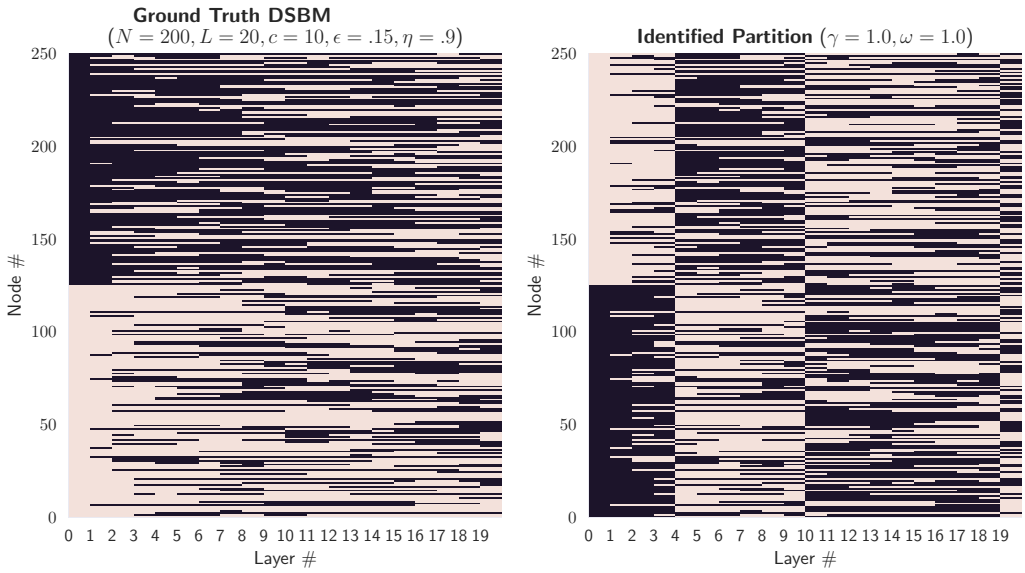


Fig. S.15: Demonstration of layer "splitting" on the multilayer dynamic stochastic block model (DSBM). Left shows the ground truth planted community assignments while the right shows the communities identified by *multimodbp* without the cross layer assignment procedure. We reiterate that this cross layer label permuting preserves all identified structure within a layer and always results in higher modularity.

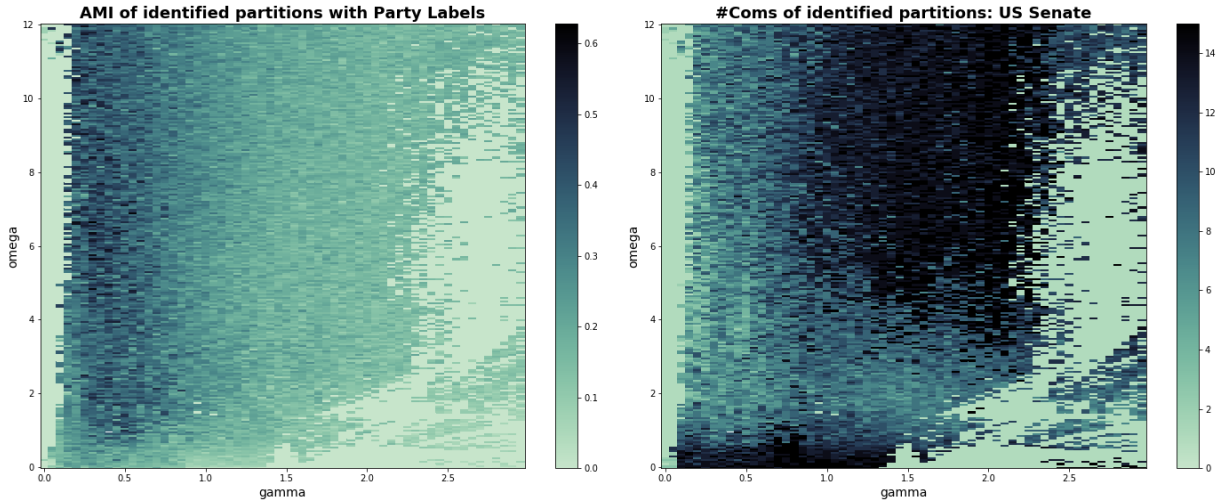


Fig. S.16: *multimodbp* applied to the US Senate Voting similarity network [53]. Left: AMI of identified partitions with the politcal party labels using *multimodbp* across a range of γ (x-axis) and ω values. Right: the number of communities identified by algorithm as a function of the parameters (γ, ω).

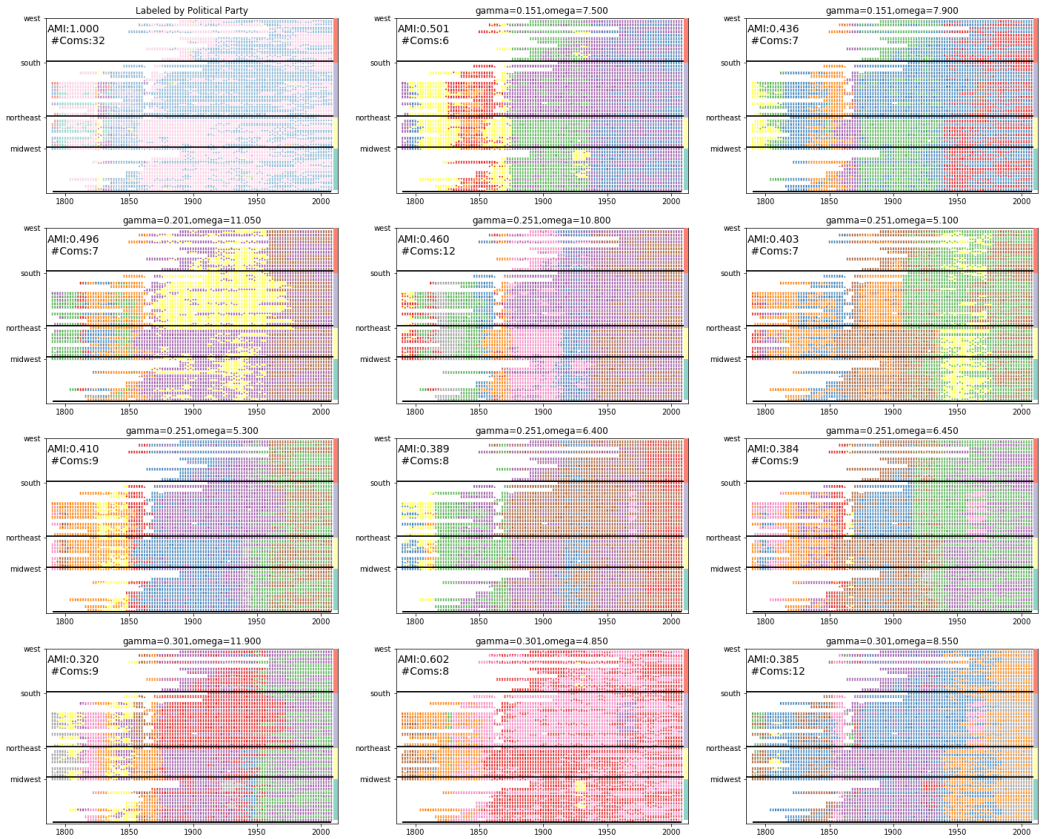


Fig. S.17: Top identified partitions based on minimization of the Bethe free energy on the US Senate voting similarity network. In each, each row represents the Senator for a particular State, organized by region, while the x-axis denotes the year of each Congress. Nodes are colored according to their identified partition, while the top left figure is colored by the political party affiliation of each senator.