Community detection in bipartite networks with stochastic block models

Tzu-Chi Yen [1,*] and Daniel B. Larremore [1,2,†]

[1] *Department of Computer Science, University of Colorado, Boulder, Colorado 80309, USA*
[2] *BioFrontiers Institute, University of Colorado, Boulder, Colorado 80303, USA*

(Received 22 January 2020; accepted 23 July 2020; published 25 September 2020)

In bipartite networks, community structures are restricted to being disassortative, in that nodes of one type are grouped according to common patterns of connection with nodes of the other type. This makes the stochastic block model (SBM), a highly flexible generative model for networks with block structure, an intuitive choice for bipartite community detection. However, typical formulations of the SBM do not make use of the special structure of bipartite networks. Here we introduce a Bayesian nonparametric formulation of the SBM and a corresponding algorithm to efficiently find communities in bipartite networks which parsimoniously chooses the number of communities. The biSBM improves community detection results over general SBMs when data are noisy, improves the model resolution limit by a factor of $\sqrt{2}$, and expands our understanding of the complicated optimization landscape associated with community detection tasks. A direct comparison of certain terms of the prior distributions in the biSBM and a related high-resolution hierarchical SBM also reveals a counterintuitive regime of community detection problems, populated by smaller and sparser networks, where nonhierarchical models outperform their more flexible counterpart.

DOI: 10.1103/PhysRevE.102.032309

I. INTRODUCTION

A bipartite network is defined as having two types of nodes, with edges allowed only between nodes of different types. For instance, a network in which edges connect people with the foods they eat is bipartite, as are other networks of associations between two classes of objects. Recent applications of bipartite networks include studies of plants and the pollinators that visit them [1], stock portfolios and the assets they comprise [2], and even U.S. Supreme Court justices and the cases they vote on [3]. More abstractly, bipartite networks also provide an alternative representation for hypergraphs in which the two types of nodes represent the hypergraph's nodes and its hyperedges, respectively [4,5].

Many networks exhibit community structure, meaning that their nodes can be divided into groups such that the nodes within each group connect to other nodes in other groups in statistically similar ways. Bipartite networks are no exception, but they exhibit a particular form of community structure because type-I nodes are defined by how they connect to type-II nodes, and *vice versa*. For example, in the bipartite network of people and the foods they eat, vegetarians belong to a group of nodes which are defined by the fact that they never connect to nodes in the group of meat-containing foods; meat-containing foods are defined by the fact that they never connect to vegetarians. While the group structure in this example comes from existing node categories, one can also ask whether statistically meaningful groups could be derived solely from the patterns of the edges themselves. This problem, typically called *community detection*, is the unsupervised task of partitioning the nodes of a network into statistically meaningful groups. In this paper, we focus on the community detection problem in bipartite networks.

There are many ways to find community structure in bipartite networks, including both general methods—which can be applied to any network—and specialized methods derived specifically for bipartite networks. We focus on a family of models related to the stochastic block model (SBM), a generative model for community structure in networks [6]. Since one of the SBM's parameters is a division of the nodes into groups, community detection with the SBM simply requires a method to fit the model to network data. With inference methods becoming increasingly sophisticated [7], many variants of the SBM have been proposed, including those that accommodate overlapping communities [8,9], broad degree distributions [10], multilayer networks [11], hierarchical community structures [12], and networks with metadata [13–15]. SBMs have also been used to estimate network structure or related observational data even if the measurement process is incomplete and erroneous [1,16–18]. In fact, a broader class of so-called mesoscale structural inference problems, like core-periphery identification and imperfect graph coloring, can also be solved using formulations of the SBM, making it a universal representation for a broad class of problems [19,20].

At first glance, the existing SBM framework is readily applicable to bipartite networks. This is because, at a high level, the two types of nodes should correspond naturally to two blocks with zero edges within each block, implying that SBMs should detect the bipartite split without that split being explicitly provided. However, past work has shown that providing node type information *a priori* improves both the quality of partitions and the time it takes to find them

[*] tzuchi.yen@colorado.edu
[†] daniel.larremore@colorado.edu

[21]. Unfortunately those results, which relied on local search algorithms to maximize model likelihood [10,21], have been superseded by more recent results which show that fitting fully Bayesian SBMs using Markov chain Monte Carlo can find structures in a more efficient and nonparametric manner [7,22,23]. These methods maximize a posterior probability, producing similar results to traditional cross validation by link predictions in many (but not all) cases [24,25]. In this sense, they avoid overfitting the data, i.e., they avoid finding a large number of communities whose predictions fail to generalize. This raises the question of whether the more sophisticated Bayesian SBM methods gain anything from being customized for bipartite networks, like the previous generation of likelihood-based methods did [21].

In this paper, we begin by introducing a nonparametric Bayesian bipartite SBM (biSBM) and show that bipartite-specific adjustments to the prior distributions improve the resolution of community detection by a factor of $\sqrt{2}$, compared with the general SBM [26]. As with the general SBM, the biSBM automatically chooses the number of communities and controls model complexity by maximizing the posterior probability.

After introducing a bipartite model, we also introduce an algorithm, designed specifically for bipartite data, that efficiently fits the model to data. Importantly, this algorithm can be applied to both the biSBM and its general counterpart, allowing us to isolate both the effects of our bipartite prior distributions and the effects of the search algorithm itself. As in the maximum likelihood case [21], the ability to customize the search algorithm for bipartite data provides both improved community detection results, as well as a more sophisticated understanding of the solution landscape, but unlike that previous work, this algorithm does more than simply require that blocks consist of only one type of node. Instead, the algorithm explores a two-dimensional (2D) landscape of model complexity, parameterized by the number of type-I blocks and the number of type-II blocks. This contributes to the growing body of work that explores the solution space of community detection models, including methods to sample the entire posterior [23], count of the number of metastable states [27], and determine the number of solution samples required to describe the landscape adequately [28].

In the following sections, we introduce a degree-corrected version of the bipartite SBM [21], which combines and extends two recent advances. Specifically, we recast the bipartite SBM [21] in a *microcanonical* and Bayesian framework [22] by assuming that the number of edges between groups and degree sequence are fixed exactly, instead of only in expectation. We then derive its likelihood, introduce prior distributions that are bipartite-specific, and describe an algorithm to efficiently fit the combined nonparametric Bayesian model to data. We then demonstrate the impacts of both the bipartite priors and algorithm in synthetic and real-world examples, and explore their impact on the maximum number of communities that our method can find, i.e., its resolution limit, before discussing the broader implications of this work.

II. THE MICROCANONICAL BIPARTITE SBM

Consider a bipartite network with N_I nodes of type I and N_II nodes of type II. The type-I nodes are divided into B_I blocks and the type-II nodes are divided into B_II blocks. Let $N = N_\mathrm{I} + N_\mathrm{II}$ and $B = B_\mathrm{I} + B_\mathrm{II}$. Rather than indexing different types of nodes separately, we index the nodes by $i = 1, 2, \dots, N$ and annotate the block assignment of node i by $b_i = 1, 2, \dots, B$. A key feature of the biSBM is that each block consists of only one type of node.

Having divided nodes into blocks, we can now write down the propensities for nodes in each block to connect to nodes in the other blocks. Let e_{rs} be the total number of edges between blocks r and s. Then, let k_i be the degree of node i. Together, $\boldsymbol{e} = \{e_{rs}\}$ and $\boldsymbol{k} = \{k_i\}$ specify the degrees of each node and the patterns by which edges are placed between blocks. The number of edges attached to a group r must be equal to the sum of its degrees, such that $e_r = \sum_s e_{rs} = \sum_{b_i=r} k_i$ for any r. For bipartite networks, $e_{rr} = 0$ for all r. We use n_r to denote the number of nodes in block r.

Given the parameters above, one can generate a network by placing edges that satisfy the constraints imposed by $\boldsymbol{e}$ and $\boldsymbol{k}$. However, that network would be just one of an ensemble of potentially many networks, all of which satisfy the constraints, analogous to the configuration model [29,30]. Peixoto showed how to count the number of networks in this ensemble [31], so that for a uniform distribution over that ensemble, the likelihood of observing any particular network is simply the inverse of the ensemble size. This means that, given $\boldsymbol{e}, \boldsymbol{k}$, and the group assignments $\boldsymbol{b} = \{b_i\}$, computing the size of the ensemble $\|\Omega(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})\|$ is tantamount to computing the likelihood of drawing a network with adjacency matrix $\boldsymbol{A}$ from the model, $P(\boldsymbol{A} \mid \boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}) = \|\Omega(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})\|^{-1}$. Thus, treating networks as equiprobable microstates in a microcanonical ensemble leads to the microcanonical stochastic block model, whose bipartite version we now develop, specifically to find communities in real-world bipartite networks. This derivation follows directly from combining the bipartite formulation of the SBM [21] with the microstate counting developed in Ref. [31]. We introduce an algorithm to fit the model in Sec. IV.

III. NONPARAMETRIC BAYESIAN SBM FOR BIPARTITE NETWORKS

We first formulate the community detection problem as a *parametric* inference procedure. The biSBM is parameterized by a partition of nodes into blocks $\boldsymbol{b}$, the number of edges between blocks $\boldsymbol{e}$, and the number of edges for each node, $\boldsymbol{k}$. However, for empirical networks, we need only search the space of partitions $\boldsymbol{b}$. This is because the microcanonical model specifies the degree sequence $\boldsymbol{k}$ exactly, so the only way that an empirical network can be found in the microcanonical ensemble is if the parameter $\boldsymbol{k}$ is equal to the empirically observed degree sequence. Note that, when $\boldsymbol{k}$ and $\boldsymbol{b}$ are both specified, $\boldsymbol{e}$ is also exactly specified. As a consequence, community detection requires only a search over partitions of the nodes into blocks $\boldsymbol{b}$.

In the absence of constraints on $\boldsymbol{b}$, the maximum likelihood solution is simply for the model to memorize the data, placing each node into its own group and letting $\hat{\boldsymbol{e}} = \boldsymbol{A}$. To counteract this tendency to dramatically overfit, we adapt the Bayesian nonparametric framework of Ref. [22], where the number of groups and other model parameters are determined from the data, and customize this framework for the situation in which the data are bipartite. We start by factorizing the joint distribution for the data and the parameters in the form

$$P(\boldsymbol{A}, \boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}) = P(\boldsymbol{A} \mid \boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})P(\boldsymbol{k} \mid \boldsymbol{e}, \boldsymbol{b})P(\boldsymbol{e} \mid \boldsymbol{b})P(\boldsymbol{b}), \quad (1)$$

where $P(\boldsymbol{k}|\boldsymbol{e}, \boldsymbol{b})$, $P(\boldsymbol{e}|\boldsymbol{b})$, and $P(\boldsymbol{b})$ are prior probabilities that we will specify in later subsections. Thus, Eq. (1) defines a complete generative model for data and parameters.

The Bayesian formulation of the SBM is a powerful approach to community detection because it enables model comparison, meaning that we can use it to choose between different model classes (e.g., hierarchical versus flat) or to choose between parametrizations of the same model (e.g., to choose the number of communities). Two approaches to model comparison, producing equivalent formulations of the problem, are useful. The first formulation is that of simply maximizing Eq. (1), taking the view that the model which maximizes the joint probability of the model and data is, statistically most justified. The second formulation is that of minimizing the so-called *description length* [32], which has a variety of interpretations (for a review and an update, see Refs. [33,34]). Perhaps the most useful interpretation for our purposes is that of compression, which takes the view that the best model is one which allows us to most compress the data, while accounting for the cost to describe the model itself. In this phrasing, for a model class M, the description length $\Sigma_M(\boldsymbol{A}, \boldsymbol{b})$ is given by $\Sigma_M(\boldsymbol{A}, \boldsymbol{b}) = -\ln P(\boldsymbol{A}|\boldsymbol{b}, M) - \ln P(\boldsymbol{b}|M)$. These two terms can be interpreted as the description cost of compressing the data $\boldsymbol{A}$ using the model and the cost of expressing the model itself, respectively. Therefore, the minimum description length (MDL) approach can be interpreted as optimizing the tradeoff between better fitting but larger models. Asymptotically, MDL is equivalent to the Bayesian Information Criterion (BIC) [35] for stochastic block models under compatible prior assumptions [12,36].

A complete and explicit formulation of model comparison will be provided in the context of our studies of empirical data in Sec. VII, using strict MDL approaches. For now, we proceed with calculating the likelihood and prior probabilities for the microcanonical biSBM and its parameters.

A. Likelihood for microcanonical bipartite SBM

The observed network $\boldsymbol{A}$ is just one of $\|\Omega(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})\|$ networks in the microcanonical ensemble which match $\{\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}\}$ exactly. Assuming that each configuration in the network ensemble is equiprobable, computing the likelihood is equivalent to taking the inverse of the size of the ensemble. We compute the size of the ensemble by counting the number of networks that match the desired block structure $\Omega(\boldsymbol{e})$ and dividing by the number of equivalent network configurations without block structure $\Xi(\boldsymbol{A})$, yielding

$$P_{\text{bi}}(\boldsymbol{A} \mid \boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b}) = \|\Omega(\boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})\|^{-1} \equiv \frac{\Xi(\boldsymbol{A})}{\Omega(\boldsymbol{e})}. \quad (2)$$

As detailed in Ref. [22], the number of networks that obey the desired block structure determined by $\boldsymbol{e}$ is given by

$$\Omega(\boldsymbol{e}) = \frac{\prod_r e_r!}{\prod_{r<s} e_{rs}!}. \quad (3)$$

This counting scheme assumes that half-edges are distinguishable. In other words, it differentiates between permutations of the neighbors of the same node, which are all equivalent (i.e., correspond to the same adjacency matrix). To discount equivalent permutations of neighbors, we count the number of half-edge pairings that correspond to the bipartite adjacency matrix $\boldsymbol{A}$,

$$\Xi(\boldsymbol{A}) = \frac{\prod_i k_i!}{\prod_{i<j} A_{ij}!}. \quad (4)$$

Note that while self-loops are forbidden, this formulation allows the possibility of multiedges.

B. Prior for the degrees

The prior for the degree sequence follows directly from Ref. [22] because $\boldsymbol{k}$ is conditioned on $\boldsymbol{e}$ and $\boldsymbol{b}$, which are bipartite. The intermediate degree distribution $\boldsymbol{\eta} = \{\eta_k^r\}$, with η_k^r being the number of nodes with degree k that belong to group r, further factorizes the conditional dependency. This allows us to write

$$P(\boldsymbol{k} \mid \boldsymbol{e}, \boldsymbol{b}) = P(\boldsymbol{k} \mid \boldsymbol{\eta})P(\boldsymbol{\eta} \mid \boldsymbol{e}, \boldsymbol{b}), \quad (5)$$

where

$$P(\boldsymbol{k} \mid \boldsymbol{\eta}) = \prod_r \frac{\prod_k \eta_k^r!}{n_r!} \quad (6)$$

is a uniform distribution of degree sequences constrained by the overall degree counts, and

$$P(\boldsymbol{\eta} \mid \boldsymbol{e}, \boldsymbol{b}) = \prod_r q(e_r, n_r)^{-1} \quad (7)$$

is the distribution of the overall degree counts. The quantity $q(m, n)$ is the number of restricted partitions of the integer m into at most n parts [37]. It can be computed via the following recurrence relation:

$$q(m, n) = q(m, n-1) + q(m-n, n), \quad (8)$$

with boundary conditions $q(m, 1) = 1$ for $m > 0$, and $q(m, n) = 0$ for $m \leqslant 0$ or $n \leqslant 0$. With this, computing $q(m, n)$ for $m \leqslant M$ and $n \leqslant m$ requires $O(M^2)$ additions of integers. In practice, we precompute $q(m, n)$ using the exact Eq. (8) for $m \leqslant 10^4$ (or $m \leqslant E$ when the network is smaller) and resort to approximations [22] only for larger arguments.

For sufficiently many nodes in each group, the hyperprior Eq. (7) will be overwhelmed by the likelihood, and the distribution of Eq. (5) will approach the actual degree sequence. In such cases, the prior and hyperprior naturally learn the true degree distribution, making them applicable to heterogeneous degrees present in real-world networks.

C. Prior for the node partition

The prior for the partitions $\boldsymbol{b}$ also follows Ref. [22] in its general outline, but the details require modification for bipartite networks. We write the prior for $\boldsymbol{b}$ as the following Bayesian hierarchy:

$$P_{\text{bi}}(\boldsymbol{b}) = P(\boldsymbol{b} \mid \boldsymbol{n})P(\boldsymbol{n} \mid B)P(B), \quad (9)$$

where $\boldsymbol{n} = \{n_r\}$, the number of nodes in each group. We then assume that this prior can be factorized into independent priors for the partitions of each type of node, i.e., $P_{\mathrm{bi}}(\boldsymbol{b}) = P(\boldsymbol{b}_{\mathrm{I}})P(\boldsymbol{b}_{\mathrm{II}})$. This allows us to treat the terms of Eq. (9) as

$$P(\boldsymbol{b} \mid \boldsymbol{n}) = \left(\frac{\prod_{\substack{\text{type-I}\\ \text{groups } r}} n_r!}{N_{\mathrm{I}}!}\right)\left(\frac{\prod_{\substack{\text{type-II}\\ \text{groups } s}} n_s!}{N_{\mathrm{II}}!}\right), \tag{10}$$

$$P(\boldsymbol{n} \mid B) = \binom{N_{\mathrm{I}} - 1}{B_{\mathrm{I}} - 1}^{-1}\binom{N_{\mathrm{II}} - 1}{B_{\mathrm{II}} - 1}^{-1}, \tag{11}$$

and

$$P(B) = N_{\mathrm{I}}^{-1} N_{\mathrm{II}}^{-1}. \tag{12}$$

Equation (11) is a uniform hyperprior over all such histograms on the node counts $\boldsymbol{n}$, while Eq. (12) is a prior for the number of nonempty groups itself. This Bayesian hierarchy over partitions accommodates heterogeneous group sizes, allowing it to model the group sizes possible in real-world networks.

D. Prior for the bipartite edge counts

We now introduce the prior for edge counts between groups, $\boldsymbol{e}$, which also requires modification for bipartite networks. While the edge count prior for general networks is parameterized by the number of groups B, the analogous prior for bipartite networks is parameterized by B_{I} and B_{II}. We therefore modify the counting scheme of Ref. [22], written for general networks, to avoid counting nonbipartite partitions that place edges between nodes of the same type. Our prior for edge counts between groups is therefore

$$P_{\mathrm{bi}}(\boldsymbol{e} \mid \boldsymbol{b}) = \left(\!\!\binom{B_{\mathrm{I}}B_{\mathrm{II}}}{E}\!\!\right)^{-1}, \tag{13}$$

where $B_{\mathrm{I}}B_{\mathrm{II}}$ counts the number of group-to-group combinations when edges are allowed only between type-I and type-II nodes. The notation $\left(\!\binom{B_{\mathrm{I}}B_{\mathrm{II}}}{E}\!\right) = \binom{B_{\mathrm{I}}B_{\mathrm{II}} + E - 1}{E}$ counts the number of histograms with $B_{\mathrm{I}}B_{\mathrm{II}}$ bins whose counts sum to E. Similar to the uniform prior for general networks [22], it is unbiased and maximally noninformative, but by neglecting mixed-type partitions, this prior results in a more parsimonious description. In later sections, we show that this modified formulation enables the detection of smaller blocks, improving the so-called resolution limit, by reducing model complexity for larger B_{I} and B_{II}.

E. Model summary

Having fully specified the priors in previous subsections, we now substitute our calculations into Eq. (1), the joint distribution for the biSBM, yielding

$$P_{\mathrm{bi}}(\boldsymbol{A},\boldsymbol{k},\boldsymbol{e},\boldsymbol{b}) = \frac{\prod_i k_i! \prod_{r<s} e_{rs}!}{\prod_r e_r! \prod_{i<j} A_{ij}!} \prod_r \frac{\prod_k \eta_k^r!}{n_r!} \frac{1}{q(e_r, n_r)} \left(\!\!\binom{B_{\mathrm{I}}B_{\mathrm{II}}}{E}\!\!\right)^{-1} \frac{\prod_r n_r!}{N_{\mathrm{I}}!N_{\mathrm{II}}!} \binom{N_{\mathrm{I}}-1}{B_{\mathrm{I}}-1}^{-1} \binom{N_{\mathrm{II}}-1}{B_{\mathrm{II}}-1}^{-1} \frac{1}{N_{\mathrm{I}}N_{\mathrm{II}}}. \tag{14}$$

Inference of the biSBM reduces to the task of sampling this distribution efficiently and correctly. Although Eq. (14) is somewhat daunting, note that $\boldsymbol{k}$ and $\boldsymbol{e}$ are implicit functions of the partition $\boldsymbol{b}$, meaning Eq. (14) depends only on the data and the partition $\boldsymbol{b}$. This opens the door to efficient sampling of the posterior distribution via Markov chain Monte Carlo which we discuss in Sec. IV.

F. Comparison with the hierarchical SBM

In deriving the biSBM, we replaced the SBM's uniform prior for edge counts with a bipartite formulation Eq. (13). However, one can instead replace it with a Bayesian hierarchy of models [Eq. (B2) [12]]. In this hierarchical SBM (hSBM), the matrix $\boldsymbol{e}$ is itself considered as an adjacency matrix of a multigraph with B nodes and E edges, allowing it to be modeled by a second SBM. Of course, the second SBM also has an edge count matrix with the same number of edges and fewer nodes, so the process of modeling each edge count matrix using another SBM can be done recursively until the model has only one block. In so doing, the hSBM typically achieves a higher posterior probability (which corresponds to higher compression, from a description length point of view) than nonhierarchical (or "flat") models, and can therefore identify finer-scale community structure.

The hSBM's edge count prior allows it to find finer-scale communities and more efficiently represent network data. However, as we will see, when the network is small and has no hierarchical structure, the hSBM can actually underfit the data, finding too few communities, due to the overhead of specifying a hierarchy even when none exists. The scenarios in which the flat bipartite prior has advantages over its hierarchical counterpart are explored in Sec. V.

IV. FITTING THE MODEL TO DATA

The mathematical formulation of the biSBM takes full advantage of a network's bipartite structure to arrive at a better model. Here we again make use of that bipartite structure to accelerate and improve our ability to fit the model, Eq. (14), to network data.

At a high level, our algorithm for model fitting consists of two key routines. The first routine is typical of SBM inference and uses Markov chain Monte Carlo importance sampling [38–40], followed by simulated annealing, to explore the space of partitions, conditioned on fixed community counts. In this routine, we accelerate mixing time by making use of the bipartite constraint, specifying a Markov chain only over states (partitions) with one type of node in each block. Importantly, this constraint has the added effect that we must fix both block counts, B_{I} and B_{II}, separately.

The second routine of our algorithm consists of an adaptive search over the 2D space of possible $(B_{\mathrm{I}}, B_{\mathrm{II}})$, using the ideas of dynamic programming [41,42]. It attempts to move quickly through those parts of the $(B_{\mathrm{I}}, B_{\mathrm{II}})$ plane that are low probability under Eq. (14) without calling the MCMC routine, and instead allocating computation time for the regions that better explain the data. The result is an effective algorithm, with two separable routines, which makes full use of the network's bipartite structure, allowing us to either maximize or sample from the posterior Eq. (14).

One advantage of having decoupled routines in this way is that the the partitioning engine is a modular component which can be swapped out for a more efficient alternative, should one be engineered or discovered. Reference implementations of two SBM partitioning algorithms, a Kernighan-Lin-inspired local search [10,21,43] and the MCMC algorithm, are freely available as part of the `bipartiteSBM` library [44].

Alternative methods for model fitting exist. For instance, it is possible to formulate a Markov chain over the entire space of partitions whose stationary distribution is the full posterior, without conditioning on the number of groups. In such a scheme, transitions in the Markov chain can create or destroy groups [23], and the Metropolis-Hastings principles guarantee that this chain will eventually mix. However, this approach turns out to be too slow to be practical because the chain gets trapped in metastable states, extending mixing times.

Another alternative approach is to avoid our 2D search over B_{I} and B_{II}, and instead search over $B = B_{\mathrm{I}} + B_{\mathrm{II}}$. This is the approach of Ref. [26], where, after proving the existence of an optimal number of blocks B, a golden-ratio one-dimensional (1D) search is used to efficiently find it.

A. Inference routine

The task of the MCMC inference routine is to maximize Eq. (14), conditioned on fixed values of B_{I} and B_{II}. Starting from an initial partition $\boldsymbol{b}_{\mathrm{init}}$, the MCMC algorithm explores the space of partitions with fixed B_{I} and B_{II} by proposing changes to the block memberships $\boldsymbol{b}$, and then accepting or rejecting those moves with carefully specified probabilities. As is typical, those probabilities are chosen so that the probability that the algorithm is at any particular partition is equal to the posterior probability of that partition, given B_{I} and B_{II}, by enforcing the Metropolis-Hastings criterion.

Rather than initializing the MCMC procedure from a fully random initial partition, we instead use an agglomerative initialization [12] which reduces burn-in time and avoids getting trapped in metastable states that are common when group sizes are large. The agglomerative initialization amounts to putting each node in its own group and then greedily merging pairs of groups of matching types until the specified B_{I} and B_{II} remain.

After initialization, each step consists of proposing to move a node i from its current group r to a new group s. Following [22], proposal moves are generated efficiently in a two-step procedure. First, we sample a random neighbor j of node i and inspect its group membership b_j. Then, with probability $\epsilon B/(e_{b_j} + \epsilon B)$ we choose s uniformly at random from $\{1, 2, \dots, B\}$; otherwise, we choose s with probability proportional to the number of edges leading to that group from group b_j, i.e., proportional to $e_{b_j s}$.

A proposed move which would violate the bipartite structure by mixing node types, or which would leave group r empty, is rejected with probability one. A valid proposed move is accepted with probability

$$a = \min\left\{1, \frac{p(b_i = s \to r)}{p(b_i = r \to s)} \exp\left(-\beta \Delta S\right)\right\}, \tag{15}$$

where

$$p(b_i = r \to s) = \sum_t R_t^i \frac{e_{ts} + \epsilon}{e_t + \epsilon B}. \tag{16}$$

Here R_t^i is the fraction of neighbors of node i which belong to block t and and $\epsilon > 0$ is an arbitrary parameter that enforces ergodicity. The term β is an inverse-temperature parameter, and ΔS is the difference between the entropies of the biSBM's microcanonical ensemble in its current state and in its proposed new state. With this in mind,

$$\Delta S = S|_{b_i=s} - S|_{b_i=r} = \ln \frac{P(\boldsymbol{A}, \boldsymbol{k}, \boldsymbol{e}, \boldsymbol{b})}{P\big(\boldsymbol{A}', \boldsymbol{k}', \boldsymbol{e}', \boldsymbol{b}'\big)}, \tag{17}$$

where variables without primes represent the current state ($b_i = r$) and variables with primes correspond to the state being proposed ($b_i = s$).

The initialization, proposal, and evaluation steps of the algorithm above are fast. With continuous bookkeeping of the incident edges to each group, proposals can be made in time $O(k_i)$, and are engineered to substantially improve the mixing times since they remove an explicit dependency on the number of groups which would otherwise be present with the fully random moves [12]. Then, when evaluating Eq. (17), we need only a number of terms proportional to k_i. In combination, the cost of an entire "sweep," consisting of one proposed move for each node in the network, is $O(E)$. The overall number of steps necessary for MCMC inference is therefore $O(\tau E)$, where τ is the average mixing time of the Markov chain, independent of B.

Our `bipartiteSBM` implementation [44] has the following default settings, chosen to stochastically maximize Eq. (14) for fixed B_{I} and B_{II} via a simulated annealing process. We first let $\epsilon = 1$, and perform 10^3 sweeps at $\beta = 1$ to reach equilibrated partitions. Then we perform zero-temperature ($\beta \to \infty$) sweeps, in which only moves leading to a strictly lower entropy are allowed. We keep track of the system's entropy during this process and exit the MCMC routine when no record-breaking event is observed within a 2×10^3 sweeps window, or when the number of sweeps exceeds 10^4, whichever is earlier. The partition $\boldsymbol{b}$ at the end corresponds to the lowest entropy. Equivalently stated, this partition $\boldsymbol{b}$ corresponds to the minimum description length or highest posterior probability, for fixed B_{I} and B_{II}. The minimal entropy at each stage is bookmarked for future decision-making processes.

The bipartite MCMC formulation is more than just similar to its general counterpart. In fact, one can show that for fixed B_{I} and B_{II}, the Markov chain transition probabilities dictated by Eq. (17) are identical for the uniform bipartite edge count prior Eq. (13) and its general equivalent introduced in Ref. [22]. This means that the MCMC algorithm explores the same entropic landscape for both bipartite and general networks when B_{I} and B_{II} are fixed. As we will demonstrate in Sec. V, however, by combining the MCMC routine with both the search routine over the block counts and the more sensitive biSBM priors, we can better infer model parameters in bipartite networks.

B. Search routine

The task of the search routine is to maximize Eq. (14) over the $(B_\mathrm{I}, B_\mathrm{II})$ plane, i.e., to find the optimal number of groups. However, maximizing Eq. (14) for any fixed choice of $(B_\mathrm{I}, B_\mathrm{II})$ requires the MCMC inference introduced above, motivating the need for an efficient search. If we were to treat the network as unipartite, a 1D convex optimization on the total number of groups $B = B_\mathrm{I} + B_\mathrm{II}$ with a search cost of $O(\ln N)$ [26] could be used. On the other hand, exhaustively exploring the plane of possibilities would incur a search cost of $O(B_\mathrm{max}^2)$, where B_max is the maximum value of B which can be detected. In fact, our experiments indicate that neither the general unipartite approach nor the naive bipartite approach is optimal. The plane search is too slow, while the line search undersamples local maxima of the $(B_\mathrm{I}, B_\mathrm{II})$ landscape, which is typically multimodal. Instead, we present a recursive routine that runs much faster than exhaustive search, which parameterizes the tradeoff between search speed and search accuracy by rapidly finding the high-probability region of the $(B_\mathrm{I}, B_\mathrm{II})$ plane without too many calls to the more expensive MCMC routine.

We provide only a brief outline of the search algorithm here, supplying full details in Appendix A. The search is initialized with each node in its own block. Blocks are rapidly agglomerated until $\min(B_\mathrm{I}, B_\mathrm{II}) = \lfloor\sqrt{2E}/2\rfloor$. This is the so-called *resolution limit*, the maximum number of communities that our algorithm can reliably find, which we discuss in detail in Sec. VI. Equation (14) will never be maximized prior to reaching this frontier. During this initial phase, we also compute the posterior probability of the trivial bipartite partition with (1,1) blocks, as a reference for the next phase.

Next, we search the region of the $(B_\mathrm{I}, B_\mathrm{II})$ plane within the resolution frontier to find a local maximum of Eq. (14) by adaptively reducing the number of communities. In this context, a local maximum is defined as an MCMC-derived partition with exactly $(B_\mathrm{I}, B_\mathrm{II})$ blocks, whose posterior probability is larger than the posterior probabilities for MCMC-derived partitions at nearby values $(B_\mathrm{I} \pm h, B_\mathrm{II} \pm h)$, for a chosen neighborhood size h. From the initial partition at the resolution frontier, we merge blocks, selected greedily from a stochastically sampled set of proposed merges. Here, because the posterior probability is a tiny value, it is computationally more convenient to work with the model entropy S, which is related to the posterior probability by $S = -\ln P$. Proposed merges are evaluated by their entropy after merging, but without calling the MCMC routine to optimize the postmerge partition. Because MCMC finds better (or no worse) fits to the data, this means that these postmerge entropies are approximate upper bounds of the best-fit entropy, given the postmerge number of blocks. We therefore use this approximate upper bound to make the search adaptive: whenever a merge would produce an upper-bound approximation that is a factor $1 + \Delta_0$ higher than the current best S, a full MCMC search is initialized at the current grid point. Otherwise, merges proceed rapidly since the approximate entropy is extremely cheap to compute.

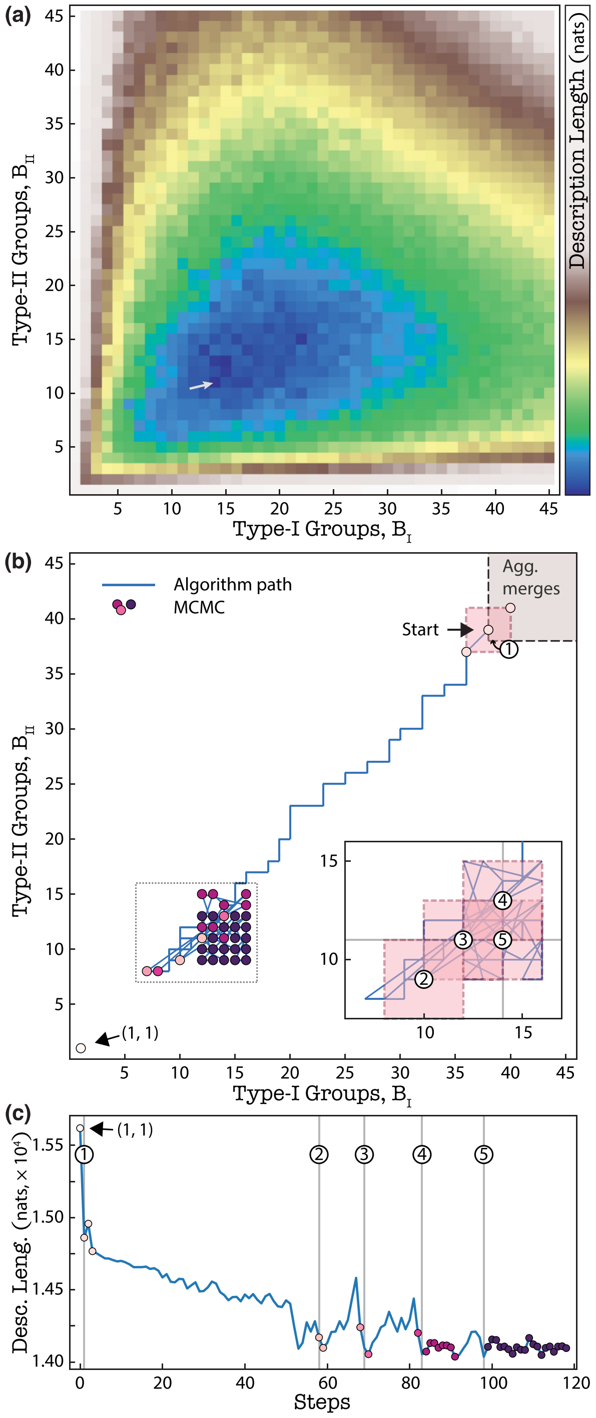

FIG. 1. Diagram showing the biSBM community detection algorithm on the description length landscape of the malaria gene-substring network [45]. (a) Each square in the heat map shows the result of fitting a model using MCMC at the specified $(B_\mathrm{I}, B_\mathrm{II})$. The color bar scales linearly. An arrow indicates the minimizing point. (b) Trajectory of the efficient search routine over the landscape shown in the top panel. Circles indicate where MCMC inference was required. Pink shaded regions show neighborhoods of exhaustive local search, with sequential order indicated by ① to ⑤. (c) Change of description length values as the algorithm progresses. Shaded circles show the steps at which the 36 MCMC calculations were performed. The minimizing point at (11,14) was found during local search ④ and confirmed during local search ⑤.

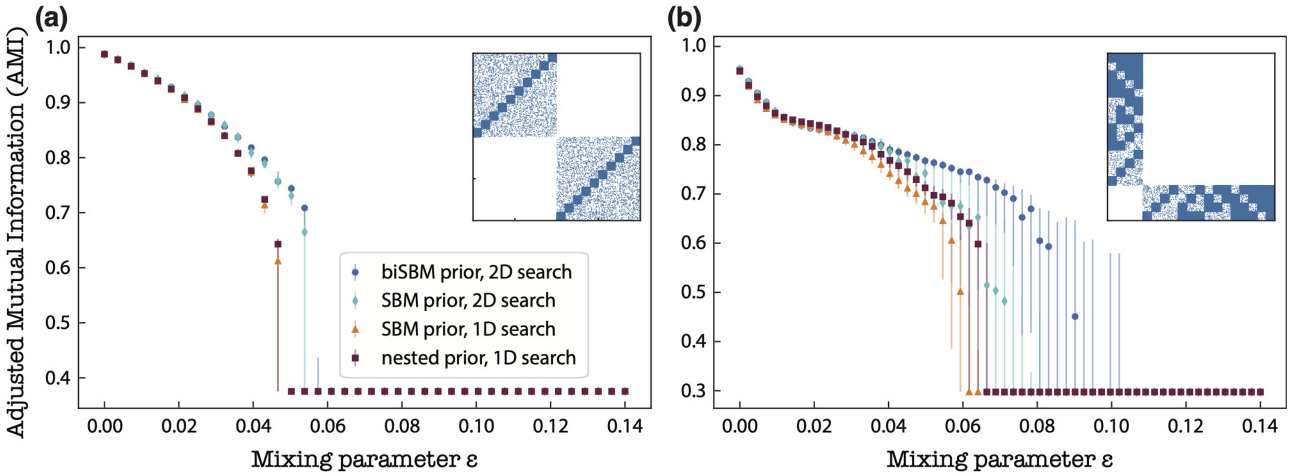

FIG. 2. Numerical tests of the recovery of planted structure in synthetic networks with $N = 10^4$ nodes. Each point shows the median of 10^2 replicates of the indicated model and algorithm (see legend) and error bars show 25%–75% quantiles. Insets show the structure of the problems at moderate ϵ. (a) A test meant to be easy: mean degree 5, equally sized groups, and $B_\text{I} = B_\text{II} = 10$. (b) A test meant to be challenging: mean degree 15, equally sized groups, and $B_\text{I} = 4$ and $B_\text{II} = 15$.

Throughout this process, the value of Δ_0 is estimated from the data to balance accuracy and efficiency, and it adaptively decreases as the search progresses (Appendix A). The algorithm exits when it finds a local minimum on the entropic landscape, returning the best overall partition explored during the search.

In practice, a typical call to the algorithm takes the form of (1) a rapid agglomerative merging phase from $(N_\text{I}, N_\text{II})$ blocks to the resolution limit frontier, (2) many agglomerative merges to move along candidate local minima that rely on approximated entropy, and (3) more deliberate and MCMC-reliant neighborhood searches to examine candidate local minima. These phases are shown in Fig. 1. The algorithm has total complexity $O(mh^2)$, where m is the number of times that an exhaustive neighborhood search is performed. When $h = 2$, we find $m < 3$ for most empirical networks examined. This algorithm is not guaranteed to find the global optimum, but due to the typical structure of the $(B_\text{I}, B_\text{II})$ optimization landscape for bipartite networks, we have found it to perform well for many synthetic and empirical networks, and it tends consistently estimate the number of groups (see Sec. VI). An implementation is available in the `bipartiteSBM` library [44].

V. RECONSTRUCTION PERFORMANCE

In this section, we examine our method's ability to correctly recover the block structure in synthetic bipartite networks where known structure has been intentionally hidden. In each test, we begin by creating a bipartite network with unambiguous block structure and then gradually mix that structure with noise until the planted blocks disappears entirely, creating a sequence of community detection problems that are increasingly challenging [46]. The performance of a community detection method can then be measured by how well it recovers the known partition over this sequence of challenges.

The typical synthetic test for unipartite networks is the *planted partition model* [47] in which groups have $\omega_{rr} = \omega_\text{in}$ assortative edges, and $\omega_{rs} = \omega_\text{out}$ disassortative edges for $r \neq s$. When the total expected degree for each group is fixed, the parameter $\epsilon = \omega_\text{out}/\omega_\text{in}$ controls the ambiguity of the planted blocks. Unambiguous assortative structure corresponds to $\epsilon = 0$ while $\epsilon = 1$ corresponds to a fully random graph. Here we consider a straightforward translation of this model to bipartite networks in which the nodes are again divided into blocks according to a planted partition. As in the unipartite planted partition model, nonzero entries of the block affinity matrix take on one of two values but due to the fact that all edges are disassortative, we replace ω_in and ω_out with ω_+ or ω_- to avoid confusion (see insets of Fig. 2). By analogy, we let $\epsilon = \omega_-/\omega_+$ while fixing the total expected degree for each group, so that $\epsilon = 0$ corresponds to highly resolved communities which blend into noise as ϵ grows.

We present two synthetic tests using this bipartite planted partition model, designed to be easy and difficult, respectively. In the easy test, the unambiguous structure consists of $N_\text{I} = N_\text{II} = \frac{1}{2}10^4$ nodes, divided evenly into $B_\text{I} = B_\text{II} = 10$ blocks of 500 nodes each, with a mean degree $\langle k \rangle = 5$. Each type-I block is matched with a type-II block so that the noise-free network consists of exactly 10 bipartite components, with zero edges placed between nodes in different components by definition. In the hard test, the unambiguous structure consists of $N = 10^4$ nodes divided evenly into $B_\text{I} = 4$ and $B_\text{II} = 15$ blocks of approximately equal size, with mean degree $\langle k \rangle = 15$. The relationships between the groups in the hard test are more complex, so the insets of Fig. 2 provide schematics of the adjacency matrices of both tests under a moderate amount of noise. In both cases, node degrees were drawn from a power-law distribution with exponent $\alpha = 2$, and for a fixed ϵ, networks were drawn from the canonical degree-corrected stochastic block model [10,21].

We test four methods' abilities to recover the bipartite planted partitions, in combinations that allow us to separate the effects of using our bipartite *model* (Sec. III) and our bipartite *search algorithm* (Sec. IV), in comparison to existing methods. The first method maximizes the biSBM posterior using our 2D search algorithm. The second method keeps the 2D search algorithm, but examines the effects of the bipartite-specific edge count prior by replacing it with the general SBM's edge count prior [i.e., replacing Eq. (13) with Eq. (B1)]. The third method uses the same general SBM edge count prior as the second, but uses a 1D bisection search [26] to examine the effects of the 2D search. The fourth method maximizes the hierarchical SBM posterior using a 1D bisection search. For the first two cases, we use our `bipartiteSBM` library [44], while for the latter two, we use the `graph-tool` library [48]. In all cases, we enforce type-specific MCMC move proposals to avoid mixed-type groups.

In the easy test, we find that the bipartite search algorithm introduced in Sec. IV performs better than the 1D searches [Fig. 2(a)]. Because the 1D search algorithm assumes that the optimization landscape is unimodal, we reasoned that other modes may emerge as ϵ increases. To test this, we generated networks within the transition region ($\epsilon \approx 0.054$) and then conducted an exhaustive survey of plausible $(B_\mathrm{I}, B_\mathrm{II})$ values using MCMC with the general SBM. This revealed two basins of attraction, located at (8, 8) and (1,1), explaining the SBM's performance. This bimodal landscape can therefore hinder search in one dimension by too quickly attracting the algorithm to the trivial bipartite partition. Perhaps surprisingly, then, a similar exhaustive survey of the $(B_\mathrm{I}, B_\mathrm{II})$ plane using the bipartite model revealed that near the transition ϵ, the biSBM has a local optimum with *more than* the planted (10,10) blocks.

In the hard case, we find that it is not the bipartite search that enables the biSBM to outperform the other methods, but rather the bipartite posterior [Fig. 2(b)]. An exploration of the outputs of the general searches shows that when they fail, they tend to find an incorrect number of blocks, which should total 19 [corresponding to the planted (4,15) blocks]. To understand this failure mode in more detail, we fixed $B = 19$ and used MCMC to fit the general SBM [48]. This led to solutions in which $B_\mathrm{I} \approx B_\mathrm{II}$, revealing that the performance degradation, relative to the biSBM, was due to a tendency for that particular algorithmic implementation of the SBM to find more balanced numbers of groups. Interestingly, near their respective transitions values of ϵ, both the SBM and biSBM tend to find more groups than were planted in the hard test, thus overfitting the data. To explore this further, we again conducted exhaustive surveys of the $(B_\mathrm{I}, B_\mathrm{II})$ plane using MCMC and found that under both models, the posterior surfaces are consistently multimodal, with attractive peaks corresponding to more communities than the planted (4,15). However, only the bipartite search algorithm introduced in Sec. IV finds overfitted partitions with too many groups; the unipartite search algorithms instead return underfitted models with too few groups, balanced between the node types.

In sum, our synthetic network tests reveal two phenomena. First, the biSBM with bipartite search is able to extract structure from higher levels of noise than the alternatives, making it an attractive option for bipartite community detection with real data. However, our tests also reveal that the posterior surfaces of both the SBM and biSBM degenerate in unexpected ways near the detectability transition [49–52].

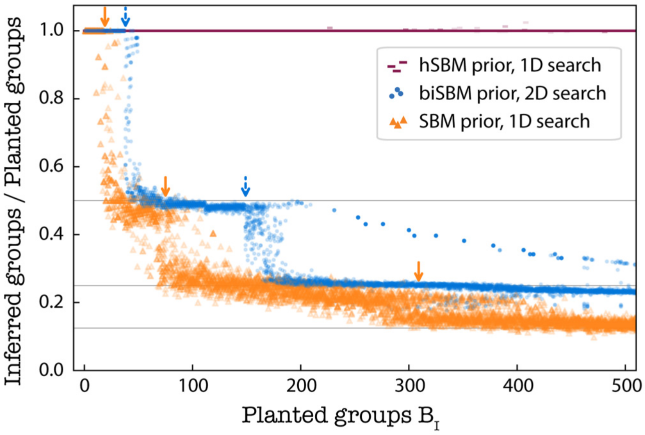

FIG. 3. A numerical experiment on bipartite cliques to demonstrate the resolution limit. As an increasing number of bipartite cliques with 10 nodes of each type are presented to the SBM, biSBM, and hSBM (see legend), the hSBM continues to find all cliques while the SBM and biSBM begin to merge pairs, quartets, and eventually octets of cliques. Arrows indicate analytical predictions of merge transitions from posterior odds ratios, with colors matching the legend. Note that biSBM transitions occur at twice the value of B as SBM transitions, showing the biSBM's expanded resolution limit.

VI. RESOLUTION LIMIT

Community detection algorithms exhibit a *resolution limit*, an upper bound on the number of blocks that can be resolved in data, even when those blocks are seemingly unambiguous. For instance, using the general SBM, only $B_\mathrm{max} = O(N^{1/2})$ groups can be detected [22], while the higher resolution of the hierarchical SBM improves this scaling to $B_\mathrm{max} = O(N/\ln N)$ [12]. In this section we investigate the resolution limit of the biSBM numerically and analytically.

Our numerical experiment considers a network of $B_\mathrm{I} = B_\mathrm{II} = \tilde{B}$ bipartite cliques of equal size, with 10 nodes of each type per biclique and therefore 100 edges per biclique. To this network, we repeatedly apply the SBM, the hSBM, and biSBM, and record the number of blocks found each time, varying $\tilde{B}$ between 1 and 510. For small values of $\tilde{B}$, all three algorithms infer $\tilde{B}$ blocks, but as the number of blocks increases, solutions which merge pairs, then quartets, and then octets become favored (Fig. 3). The hSBM continues to find $\tilde{B}$ blocks, as expected.

The exact value of $\tilde{B}$ at which merging blocks into pairs becomes more attractive can be derived by asking when the corresponding posterior odds ratio, comparing a model with $\tilde{B}$ bicliques to a model with $\tilde{B}/2$ biclique pairs, exceeds one:

$$\Lambda(\tilde{B}) = \frac{P(\boldsymbol{A}, \boldsymbol{k}, \boldsymbol{e}_\mathrm{cliquepairs}, \boldsymbol{b}_\mathrm{cliquepairs})}{P(\boldsymbol{A}, \boldsymbol{k}, \boldsymbol{e}_\mathrm{cliques}, \boldsymbol{b}_\mathrm{cliques})}. \tag{18}$$

When there are 10 nodes of each type per biclique and 100 edges, $\Lambda(\tilde{B})$ exceeds 1 when $\tilde{B} = 19$ for the SBM and $\tilde{B} = 38$ for the biSBM (Fig. 3; arrows). A similar calculation predicts the transition from biclique pairs to biclique quartets at $\tilde{B} = 75$ for the SBM and $\tilde{B} = 149$ for the biSBM (Fig. 3; arrows).

Numerical experiments confirm these analytical predictions, but noisily, due to the stochastic search algorithms involved, and the fact the optimization landscapes are truly multimodal, particularly near points of transition.

The posterior odds ratio calculations above can be generalized, and show that the biSBM extends the resolution transitions twice as far as the SBM for the transitions from $B \to \frac{1}{2}B \to \frac{1}{4}B \to \cdots$, and so on but still undergoes the same transitions eventually. Thus, both models exhibit the same resolution limit scaling $B_{\max} = O(N^{1/2})$, but with resolution degradations that occur at N for the SBM occurring at $2N$ for the biSBM. Therefore, the resolution limit of the biSBM is $\sqrt{2}$ larger than the SBM for the same number of nodes. One can alternatively retrace the analysis of Ref. [22] but for the biSBM applied to bicliques to derive the same $\sqrt{2}$ resolution improvement.

This constant-factor improvement in resolution limit may seem irrelevant, given that the major contribution of the hierarchical SBM was to change the order of the limit to $B_{\max} = O(N/\ln N)$ [12]. However, we find that, on the contrary, the $\sqrt{2}$ factor improvement for the biSBM expands a previously uninvestigated regime in which flat models outperform their hierarchical cousin. When given the biclique data, the hSBM finds a hierarchical division where at each level l, the number of groups decreases by a factor σ_l, except at the highest level where it finds a bipartite division. Assuming that $\sigma_l = \sigma$, we have $B_l = 2\tilde{B}_l$, where $B_l = \tilde{B}/\sigma^{l-1}$. The hSBM's prior for edge counts [Eq. (B2)] can be factored into uniform distributions over multigraphs at lower levels and over an SBM at the topmost level, leading to

$$P_{\text{lower}}(\boldsymbol{e}) = \prod_{l=1}^{\log_\sigma \tilde{B}} \left(\!\!\binom{\sigma^2}{E\sigma^l/\tilde{B}}\!\!\right)^{-\tilde{B}/\sigma^l} \times \frac{\sigma!^{2\tilde{B}/\sigma^l}}{(\tilde{B}/\sigma^{l-1})!^2} \binom{\tilde{B}/\sigma^{l-1} - 1}{\tilde{B}/\sigma^l - 1}^{-2} \qquad (19)$$

and

$$P_{\text{topmost}}(\boldsymbol{e}) = \left(\!\!\binom{\left(\!\binom{2}{2}\!\right)}{E}\!\!\right)^{-1}. \qquad (20)$$

By comparing $P_{\text{hier}} = P_{\text{lower}} P_{\text{topmost}}$ with the corresponding terms from the biSBM [Eq. (13)] or the corresponding equation for the SBM [Eq. (B1)], we can identify regimes in which a flat model better describes network data than the nested model.

Figure 4 shows regimes in which the flat model is preferred for both the SBM and biSBM. These regimes are larger for the biSBM than the SBM, as expected, and are larger when the hierarchical branching factor σ decreases—indeed, if the data are less hierarchical, the hierarchical model is expected to have less of an advantage. The flat-model description is also favored when there are fewer edges and more groups, suggesting that in order for the nested model to be useful, it requires sufficient data to support its more costly nested architecture. A number of real-world networks that fall into this flat-model regime are described in the following section. We note that our definition of this regime relies on assumptions of perfect inference and a fixed branching factor at each level of the hSBM's hierarchy. These assumptions may not always hold.

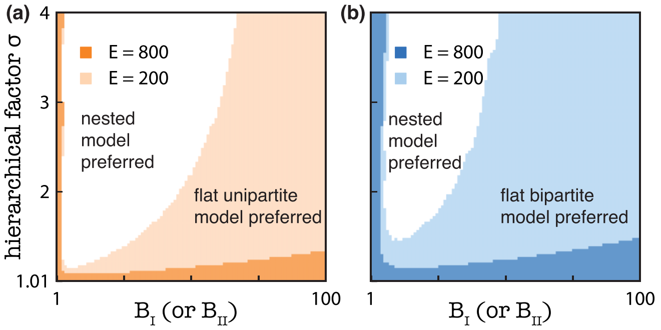

FIG. 4. Comparison of the description lengths resulting from prior distribution over edge counts using the biSBM, SBM, and hSBM priors. Regions where a flat prior has a lower description length than the hierarchical prior are shaded for (a) the SBM and (b) the biSBM. Flat priors are favored when there are fewer edges, more groups, and a smaller hierarchical branching factor σ (defined in Sec. VI). The flat-model regime is larger for the biSBM than the SBM, as described in Sec. VI.

VII. EMPIRICAL NETWORKS

We now examine the application of the biSBM to a corpus of real-world networks ranging in size from $N = 32$ to $N = 135\,568$ nodes, across social, biological, linguistic, and technological domains. While it was typical of past studies to measure a community detection method by its ability to recapitulate known metadata labels, we acknowledge that this approach is inadvisable for a number of theoretical and practical reasons [15] and instead compare the biSBM to the SBM and hSBM using Bayesian model selection.

In general, to compare one partition-model pair $(\boldsymbol{b}_0, M_0)$ and an alternative pair $(\boldsymbol{b}_1, M_1)$, we can compute the posterior odds ratio:

$$\Lambda = \frac{P(\boldsymbol{b}_0, M_0|\boldsymbol{A})}{P(\boldsymbol{b}_1, M_1|\boldsymbol{A})} = \frac{P(\boldsymbol{A}, \boldsymbol{b}_0|M_0)}{P(\boldsymbol{A}, \boldsymbol{b}_1|M_1)} \times \frac{P(M_0)}{P(M_1)}. \qquad (21)$$

Model $(\boldsymbol{b}_0, M_0)$ is favored when $\Lambda > 1$ and model $(\boldsymbol{b}_1, M_1)$ is favored when $\Lambda < 1$, with the magnitude of difference from $\Lambda = 1$ indicating the degree of confidence in model selection [76]. In the absence of any *a priori* preference for either model, $P(M_0) = P(M_1)$, meaning that the ratio of probabilities Λ can be alternatively expressed via the difference in description lengths, $\Lambda \equiv \exp(\Sigma_1 - \Sigma_0)$. [Recall that the description length Σ_ℓ for the combined model $(\boldsymbol{b}_\ell, M_\ell)$ and data $\boldsymbol{A}$ can be written as the negative log of the posterior probability, as introduced in Sec. III.] In what follows, we compare the hSBM to the biSBM and without loss of generality choose M_1 to be whichever model is favored so that Λ simply expresses the magnitude of the odds ratio. Note that by construction, the biSBM always outperforms the flat SBM.

As predicted in the previous section, the biSBM's flat prior is better when networks are smaller and sparser, while for larger networks the hSBM generally performs better by building a hierarchy that results in a more parsimonious model (Table I). Indeed, the majority of larger networks are better

TABLE I. Results for 24 empirical networks. Number of nodes n_I, n_II, mean degree $\langle k \rangle$, number of type-I groups B_I, number of type-II groups B_II, and description length per edge Σ/E. Superscripts: b-biSBM, g-SBM, h-hSBM. L indicates the number of levels found by the hSBM. Reported values indicate best of 100 independent runs. Unless otherwise noted, data are accessible from the Colorado Index of Complex Networks (ICON) [53]. The confidence level is marked with asterisks.[a]

Data set	N_I	N_II	$\langle k \rangle$	$(B_\mathrm{I}^\mathrm{b}, B_\mathrm{II}^\mathrm{b})$	$(B_\mathrm{I}^\mathrm{g}, B_\mathrm{II}^\mathrm{g})$	$(B_\mathrm{I}^\mathrm{h}, B_\mathrm{II}^\mathrm{h})$	$\langle L+1 \rangle$	Σ^b/E	Σ^h/E
Southern women interactions [54]	18	14	5.56	(1, 1)	(1, 1)	(1, 1)	2.0	**2.15***	2.26
Joern plant-herbivore web [55]	22	52	4.97	(2, 2)	(1, 1)	(1, 1)	2.0	**2.64***	2.74
Swingers and parties [56]	57	39	4.83	(1, 1)	(1, 1)	(1, 1)	2.0	**2.92***	2.97
McMullen pollination web [57]	54	105	2.57	(2, 2)	(2, 2)	(1, 1)	2.0	**2.87***	3.02
Ndrangheta criminals [58]	156	47	4.48	(3, 4)	(3, 3)	(3, 4)	2.87	**3.44***	3.49
Abu Sayyaf kidnappings[b] [59]	246	105	2.28	(2, 2)	(1, 1)	(1, 1)	2.0	**4.50***	4.54
Virus-host interactome [60]	53	307	2.52	(2, 2)	(1, 1)	(1, 1)	2.0	**3.78***	3.81
Clements-Long plant pollinator [61]	275	96	4.98	(1, 1)	(1, 1)	(1, 1)	2.0	**3.45***	3.47
Human musculoskeletal system [62]	173	270	4.30	(7, 8)	(5, 5)	(8, 8)	4.01	**3.94**	3.94
Mexican drug trafficking[b] [63]	765	10	16.1	(12, 8)	(8, 7)	(10, 6)	3.11	**1.26***	1.29
Country-language network [64]	254	614	2.89	(4, 5)	(2, 2)	(4, 3)	2.11	**4.53***	4.56
Malaria gene similarity [45]	297	806	5.38	(15, 16)	(6, 6)	(25, 20)	4.95	4.73	**4.67***
Protein complex drug [65]	739	680	5.20	(20, 22)	(14, 14)	(35, 39)	5.06	3.65	**3.50****
Robertson plant pollinator [66]	456	1428	16.2	(20, 18)	(11, 11)	(20, 19)	4.0	**3.10***	3.10
Human gene disease network [67]	1419	516	4.06	(13, 14)	(9, 9)	(35, 36)	5.04	5.02	**4.80****
Food ingredients flavors web [68]	1525	1107	27.9	(27, 69)	(20, 29)	(42, 130)	4.91	2.55	**2.51****
Wikipedia doc word network [69]	63	3140	24.8	(22, 206)	(18, 23)	(29, 71)	4.17	1.58	**1.51****
Foursquare check-ins [70]	2060	2876	11.0	(65, 66)	(40, 40)	(244, 248)	5.2	5.92	**5.09****
Ancient metabolic network [71]	5651	5252	4.22	(18, 22)	(5, 5)	(17, 21)	4.26	**5.68****	5.82
Marvel Universe characters [72]	6486	12 942	9.95	(68, 72)	(67, 62)	(365, 314)	6.24	4.70	**4.42*****
Reuters news stories [73]	19 757	38 677	33.5	(396, 440)	(87, 108)	(294, 463)	6.25	4.22	**4.16*****
IMDb movie-actor data set[c]	53 158	39 768	6.49	(91, 92)	(69, 68)	(264, 265)	6.22	7.40	**7.30*****
YouTube group memberships [74]	94 238	30 087	4.72	(62, 66)	(37, 38)	(221, 238)	5.9	**7.07****	7.13
DBpedia writer network [75]	89 355	46 213	2.13	(22, 26)	(2, 3)	(2, 3)	2.16	**10.32****	10.41

[a] Via the posterior odds ratio: $^{*}: \Lambda < 10^{-2}$; $^{**}: \Lambda < 10^{-100}$; $^{***}: \Lambda < 10^{-10\,000}$.

[b] Temporal data with time stamps are aggregated, making a multigraph.

[c] Data available at [77]. IMDb copyright permits redistribution of data only in unaltered form.

described using the hSBM (Table I; rightmost columns), but exceptions do exist, including the ancient metabolic network [71], YouTube memberships [74], and DBpedia writer network [75], which share the common feature of low density. The Robertson plant-pollinator network [66], on the other hand, is neither small nor particularly sparse, and yet the biSBM is still weakly preferred over the hSBM.

Differences between models, based only on their *maximum a posteriori* (i.e., minimum description length) estimates, may overlook additional complexity in the models' full posterior distributions. We repeatedly sample from the posterior distributions of the SBM, biSBM, and hSBM for eight networks from Table I, showing both posterior description length distributions and inferred block count distributions (Fig. 5). Generally, all three models exhibit similar description-length variation, but due to the 2D search introduced in Sec. IV, the biSBM returns partitions with wider variation in B_I and B_II. For instance, the drug trafficking network [63], a multigraph with $N_\mathrm{I} \gg N_\mathrm{II}$, has a bimodal distribution of description lengths under the hSBM, while the biSBM finds plausible partitions for a wide variety of B_I values [Fig. 5(b)]. On the other hand, posterior distributions for the country-language network [64] are all unimodal, but the biSBM finds probable states with wide variation in description length and block counts, while the hSBM samples from a small region [Fig. 5(c)]. This can happen when the network is small, since the hSBM requires sufficiently complicated data to justify a hierarchy, while the biSBM finds a variety of lower description length partitions. In fact, viewing the same data sets through the lenses of these different models' priors can quite clearly shift the location of posterior peaks. This is most clearly visible in the Reuters network [73], for which the models have unambiguous and nonoverlapping preferred states [Fig. 5(f)].

Briefly, we note that model comparison is possible here due to the fact that all of the models we considered are SBMs with clearly specified posterior distributions. Broader comparisons between community detection models of entirely different classes are also possible, for which we suggest Ref. [78].

VIII. DISCUSSION

This paper presented a bipartite microcanonical stochastic block model (biSBM) and an algorithm to fit the model to network data. Our work is built on two foundations, developing a bipartite SBM [21] with a more sophisticated microstate counting approach [31]. The model itself follows in the footsteps of Bayesian SBMs [12,22] but with key modifications to the prior distribution and the search algorithm that more correctly account for the fact that some partitions are strictly prohibited when a network is bipartite. As a result, the biSBM is able to resolve community structure in bipartite

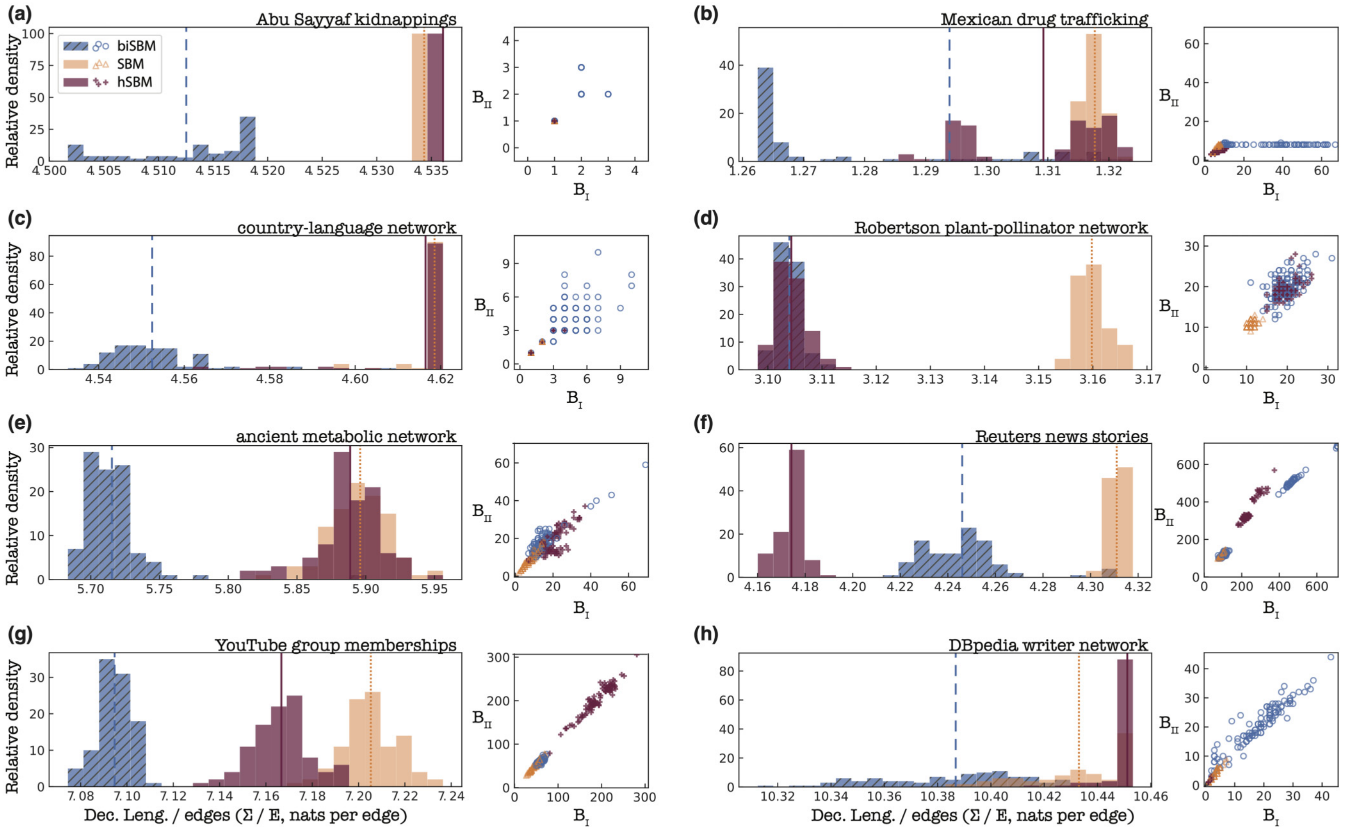

FIG. 5. Repeated application of models (see legend in panel a) with default algorithms produces distributions of the description length and the number of groups, for eight of the empirical networks listed in Table I. Vertical lines mark the value of the mean description length.

networks better than the general SBM, demonstrated in tests with synthetic networks (Fig. 2).

The resolution limit of the biSBM is greater than the general SBM by a factor of $\sqrt{2}$. We demonstrated this mathematically and in a simple biclique-finding test (Fig. 3). This analysis led us to directly compare the priors for the biSBM and the hierarchical SBM, which hinted at an unexpected regime in which the biSBM provides a better model than the hSBM. This regime, populated by smaller, sparser, and less hierarchical networks, was found in real data where model selection favored the biSBM (Table I).

How should we understand these networks that are better described by our flat model than a hierarchical one? One possibility is that these networks are simply "flat" and so any hierarchical description simply wastes description-length bits on a model which is too complex. Another possibility is that this result can be explained not by the mathematics of the models but by the algorithms used to fit the models. In fact, our tests with synthetic networks show clear differences between models *and* algorithms, with the 2D search algorithm introduced here providing better fits to data than a 1D search (Fig. 2). However, this finding alone does not actually differentiate between the two possible explanations, and so we constructed the following simple test.

To probe the differences between the biSBM and hSBM as *models* versus differences in their model-fitting *algorithms*, we combined both approaches in a two-step protocol: Fit the biSBM to network data and then build an optimal hierarchical model upon that fixed biSBM base. Unless the data are completely flat, this hierarchy-building process will further reduce the description length, providing a more parsimonious model. If the hybrid h-biSBM provides a superior description length to the hSBM, our observations can be attributed to differences in model-fitting algorithms. In fact, this is precisely what we find.

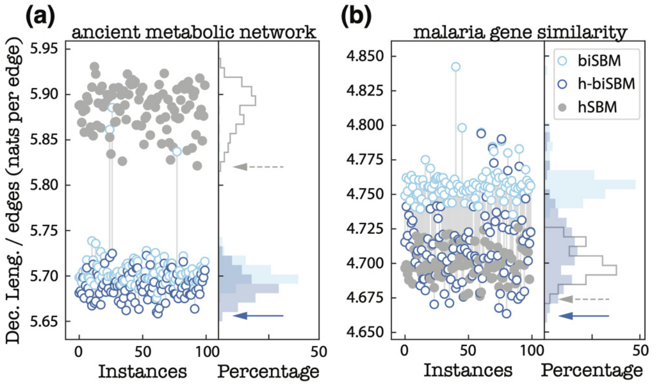

FIG. 6. Scatter plots and histograms of description length for the ancient metabolic (a) [71] and malaria gene similarity (b) [45] networks from 100 independent experiments. Gray vertical lines connect biSBM results with their matching h-biSBM hierarchical results. Arrows in histograms mark the MDL points from the hSBM (gray) and by the h-biSBM (blue).

Figure 6 shows repeated application of the biSBM, hSBM, and hybrid h-biSBM to the ancient metabolic network [71] and the malaria genes network [45]. In the ancient metabolic network, the biSBM already outperformed the hSBM, so the hybrid model results in only marginal improvements in description length. However, doing so also creates hierarchies with an average depth of $\langle L \rangle = 3.85$ layers, compared with the $\langle L \rangle = 3.27$ layers found by hSBM natively. In other words, we can achieve a deeper hierarchy in addition to a more parsimonious model when using the flat biSBM partition at the lowest level. This suggests that, in fact, not all of the hSBM's underperformance can be attributed to the ancient metabolic network's being "flat," since a hierarchy can be constructed upon the biSBM's inferred structure. In the malaria genes network, although the hSBM outperformed the biSBM, the hybrid model was superior to both. Since the hybrid partitions are, in principle, available to the hSBM, our conclusion is that the 2D search algorithm we presented is actually finding better partitions. Put another way, there are further opportunities to improve the depth and speed of algorithms to fit stochastic block models to real-world data, particularly when bipartite or other structure in the data can be exploited.

Finally, this work shows how both models and algorithms can reflect the structural constraints of real-world network data, and how doing so improves model quality. While our work addresses only community detection for bipartite networks, generalizations of both the mathematics and search algorithms could in principle be derived for multipartite networks in which more complicated rules exist for how node types are allowed to connect.

ACKNOWLEDGMENTS

The authors thank T. Peixoto, T. Kawamoto, P. Zhang, J. Grochow, and J.-G. Young for stimulating discussions. D.B.L. was supported in part by the Santa Fe Institute Omidyar Fellowship. The authors thank the BioFrontiers Institute at the University of Colorado Boulder and the Santa Fe Institute for the use of their computational facilities.

APPENDIX A: RECURSIVE 2D SEARCH ALGORITHM

In this Appendix, we elaborate on the recursive search algorithm sketched in Sec. IV B. Our overarching problem is to find the $(B_{\mathrm{I}}, B_{\mathrm{II}})$ pair that minimizes the description length. We use dynamic programming to solve this problem efficiently, observing that it has the following two properties:

(1) *Optimal substructure.* If we collectively inspect the solutions that lead to local minima, then the best of those determines the global minimum.

(2) *Overlapping subproblems.* To verify the existence of a local minimum, we have to compute the description length for its neighborhood points. There are many subproblems which are solved again and again, and their solutions can be stored in a table so that these need not to be recomputed.

Our recursive algorithm is summarized in Algorithms 1 and 2. Due to its recursive construction, a base case (or smallest subproblem) represents a leaf node in the recursive search tree, at which we either terminate the algorithm or traceback and try a different node. The target point of the base case is $\mathbf{p}^* = (B_{\mathrm{I}}^*, B_{\mathrm{II}}^*)$ only if its entropy S^* is (1) minimal over all algorithmic history and (2) locally minimal compared with points $\mathbf{p}$ within the neighborhood

$$\mathcal{B}_h(\mathbf{p}) = \{\mathbf{p}^* \in \mathbb{Z}_+^2 \mid \|\mathbf{p} - \mathbf{p}^*\|_\infty \leqslant h\}, \tag{A1}$$

where h is a user-defined parameter that controls the size of the subproblem.

Algorithm 1 Pseudocode for the search of the minimal entropy (or description length) point on the 2D landscape. The function ADAPTIVE_SEARCH and its dependency LOCAL-MINIMUM_CHECK are described in Algorithm 2.

```
Input: Network $G = (V, E)$ with adjacency matrix $\boldsymbol{A}$ and the partition $\boldsymbol{b}^0$ which expresses each node belongs to which type.
Parameters (we used the default values throughout this paper unless otherwise noted):
•$n_m = 10$ number of merges attempted for each block
•$\sigma = 1.01$ greediness of agglomerative merges
•$\alpha = 0.9$ adaptive parameter in case of overshooting
•$c = 3$ trade-off parameter to determine $\Delta_0$
•$h = 2$ neighborhood size
Output: Memoization table $\Xi$, which includes the MDL point.
Phase I: Initialization
$\Delta_0 \leftarrow 1$
Compute entropy $S^0$ for the trivial partition $\boldsymbol{b}^0$
Initiate memorization table $\Xi[1, 1] \leftarrow (\boldsymbol{b}^0, S^0)$
$(B_{\mathrm{I}}^{\mathrm{init}}, B_{\mathrm{II}}^{\mathrm{init}}) \leftarrow$ AGGLOMERATIVE-MERGE$(\boldsymbol{A}, n_m)$
Phase II: Dynamic Programming
Run ADAPTIVE_SEARCH$(B_{\mathrm{I}}^{\mathrm{init}}, B_{\mathrm{II}}^{\mathrm{init}})$
```

In Phase I, we perform MCMC at the trivial bipartite partition to create a reference description length. Then, starting from an initial state in which each node belongs to its own group, we apply an `Agglomerative-Merge` algorithm (also summarized in the main text) to reach a partition at $(B_{\mathrm{I}}^{\mathrm{init}}, B_{\mathrm{II}}^{\mathrm{init}})$, where $\min(B_{\mathrm{I}}^{\mathrm{init}}, B_{\mathrm{II}}^{\mathrm{init}}) = \lfloor\sqrt{2E}/2\rfloor$. The algorithm works as follows. In each sweep, we attempt n_m block changes according to Eq. (16) for each block. These proposal moves are not uniformly random, but are instead based on the current block structure, treating the edge count matrix $\boldsymbol{e}$ as the adjacency matrix of a multigraph so that blocks can be thought of as nodes in this higher-level representation. Potential merges of blocks are then ranked according to increasing ΔS, and exactly $\lceil B(1 - \sigma^{-1})\rceil$ block merges are performed in that order, in each sweep. To minimize the impact of bad merges done in the earlier steps, at the end of each sweep we apply MCMC algorithm described in the main text at zero temperature, allowing block changes that strictly decrease the entropy. This algorithm has an overall complexity of $O(E \ln^2 N)$ [40], which is dwarfed when compared with the MCMC calculation. Note that in Sec. IV A, we perform the same agglomerative merge algorithm right before the MCMC inference but merge blocks to a specific number of groups $(B_{\mathrm{I}}, B_{\mathrm{II}})$, rather than to a threshold given by $\min(B_{\mathrm{I}}^{\mathrm{init}}, B_{\mathrm{II}}^{\mathrm{init}}) = \lfloor\sqrt{2E}/2\rfloor$.

Phase II is the core recursive algorithm. Starting from the $(B_{\mathrm{I}}^{\mathrm{init}}, B_{\mathrm{II}}^{\mathrm{init}})$ partition, we check whether it is a local minimum in the description length landscape, where the radius of the

Algorithm 2 Pseudocode for the subroutines used in Algm. 1. Here MDL is equivalent to the minimal MCMC-calculated entropy.

```
function Adaptive_Search($B_{\rm I}$, $B_{\rm II}$)
    if Local-Minimum_Check($B_{\rm I}$, $B_{\rm II}$) then
        return $\Xi$
    else
        $\Delta S \leftarrow 0$
        Update $S_{\rm ref}$ to current MDL
        while $\Delta S < \Delta_0 \times S_{\rm ref}$ do
            if $B_{\rm I} \times B_{\rm II} = 1$ then
                break
            else
                $r, s, \min(\{\delta_{r\sim s}\}) \leftarrow$ Rand-Merge($\boldsymbol{e}$, $n_m$)
                $\Delta S \leftarrow \min(\{\delta_{r\sim s}\}) \times S_{\rm ref}$
                if $\Delta_0 = 1$ then
                    Update $\Delta_0$ if Eq. (A2) is True
                    break
                end if
                Update $B_{\rm I}$, $B_{\rm II}$ from merged block pair $r, s$
                Update $\boldsymbol{e}$ accordingly
            end if
        end while
        return Adaptive_Search($B_{\rm I}$, $B_{\rm II}$)
    end if
end function

function Local-Minimum_Check($B_{\rm I}$, $B_{\rm II}$)
    Compute entropy $S$ at ($B_{\rm I}$, $B_{\rm II}$) via Eq. (14)
    $\Xi[B_{\rm I}, B_{\rm II}] \leftarrow (\boldsymbol{b}, S)$, where $\boldsymbol{b}$ is the optimal partition
    if $S > S^0$ then
        return False
    end if
    if $S >$ current MDL then
        $\Delta_0 \leftarrow \alpha \Delta_0$
        Update $B_{\rm I}$, $B_{\rm II}$ to which that give current MDL
    end if
    for point $\mathbf{p} \in \mathcal{B}_h(B_{\rm I}, B_{\rm II})$ do
        Compute entropy $S^{\mathbf{p}}$ via Eq. (14)
        $\Xi[\mathbf{p}_1, \mathbf{p}_2] \leftarrow (\boldsymbol{b}^{\mathbf{p}}, S^{\mathbf{p}})$
    end for
    if $S >$ currentMDL then
        Update $B_{\rm I}$, $B_{\rm II}$ to which that give current MDL
        return False
    else
        $\Xi[B_{\rm I}, B_{\rm II}] \leftarrow (\boldsymbol{b}, S)$
        return True
    end if
end function
```

local neighborhood is h, as defined in Eq. (A1). If the current point is indeed a local minimum, the algorithm terminates. If it is not, the algorithm finds another candidate point in the $(B_{\rm I}, B_{\rm II})$ grid by calling the Rand-Merge routine, which works by proposing many ways in which pairs of blocks could be merged, and then choosing the best merge. In particular, Rand-Merge proposes, for each block r, n_m other blocks s to which r could be merged, selected uniformly at random. From among those candidate merges, we choose the pair of blocks r, s with the smallest relative entropy deviation $\delta_{r\sim s} = (S_{r\sim s} - S_{\rm ref})/S_{\rm ref}$. Here $S_{\rm ref}$ is the minimum of all MCMC-calculated entropies explored globally and $S_{r\sim s}$ is the entropy that would result from a hypothetical merging of blocks r and s.

At this point, the algorithm gains its efficiency from avoiding calling the costly MCMC routine while still moving toward a local minimum in the $(B_{\rm I}, B_{\rm II})$ plane. To do so requires that we accept the merge and entropy change $\delta^* = \min(\{\delta_{r\sim s}\})$ from Rand-Merge without pausing to re-fit the model using MCMC at the new $(B_{\rm I}, B_{\rm II})$. If this process is repeated, the entropy after accumulating merges will deviate more and more from the optimal entropy, were we to refit the model using MCMC at the current $(B_{\rm I}, B_{\rm II})$. We therefore balance speed and efficiency by introducing a parameter that forces a full MCMC fit only when the accumulated entropy from repeated merges becomes intolerable. Let $0 < \Delta_0 < 1$ such that when the a block merge does not deviate from the entropy too much ($\delta^* < \Delta_0$), we accept the merge and attempt the next successive merges. Otherwise, we seek to terminate the algorithm by calling Local-Minimum_Check again at the current $(B_{\rm I}, B_{\rm II})$.

The key to efficiency is that computing the approximated partition by block merges from an optimized partition is faster than finding it from scratch. Note that when the state of the algorithm is far from a local minimum, $\delta^* = \min(\{\delta_{r\sim s}\})$ is typically small and negative, meaning that a large number of merges can often be performed before a full MCMC is required. Thus, choosing Δ_0 is important. If we choose a large Δ_0, the algorithm can overshoot the local minimum, requiring it to only gradually rediscover that minimum by inspecting many neighboring points. On the other hand, if we choose a small Δ_0, there will be a larger number of MCMC calculations, which we also want to avoid. To this end, we determine Δ_0 from the data on the fly during the Adaptive_Search step; namely, Δ_0 is the first outlier δ^* based on the Interquartile Rule,

$$\delta^* > c\,\mathrm{IQR}(\{\delta^*\}) + Q_3(\{\delta^*\}), \tag{A2}$$

where $\{\cdot\}$ collects the δ^* at earlier sweeps and the IQR is the interquartile range, being equal to the difference between 75th and 25th percentiles. However, with this choice, we may still overshoot. In such cases, we reduce Δ_0 by a factor α and relocate our attention to the $(B_{\rm I}^*, B_{\rm II}^*)$ whose entropy S^* is minimal so far, and then call Local-Minimum_Check. During the neighborhood check, if we find an even better point nearby, we will relocate the tip to that point, and continue with the Adaptive_Search step. The algorithm ends if a local minimum is found.

Because of our dynamic programming approach, the time complexity of the total algorithm cannot be computed directly from the recursion, nor do we know the exact number of subproblems (local searches using MCMC) that the algorithm will need to call. Indeed, as we found for synthetic networks near the detectability limit, and for networks near transitions in the resolution limit, the $(B_{\rm I}, B_{\rm II})$-optimization landscape becomes degenerate and multimodal, making a general algorithm complexity result hopeless.

Nevertheless, the time complexity of the search algorithm scales with the number of MCMC calculations. Heuristic arguments suggest the number of MCMC calculations should be on the order of $O(mh^2)$, where m is the number of times that the most expensive for-loop (line 11 of `Local-Minimum_Check`) is called. However, even this is an approximation, due to the fact that, at times, a local-minimum check reveals a point within the h-neighborhood that is better than the point currently being checked. In this way, subproblems may overlap, making the total cost somewhat cheaper. Empirically, for most networks in Table I, $m < 3$.

TABLE II. List of model likelihood and prior functions, which contribute to the overall posterior probability function of different model variants used in this paper.

Model variant	SBM	biSBM	hSBM		
Hierarchy level	–		0	$1,\dots,L-1$	L
$P(\boldsymbol{e}\mid\boldsymbol{b})$	Eq. (B1)	Eq. (13)	–	–	Eq. (B1)
$P(\boldsymbol{A}\mid\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$	Eq. (2)		Eq. (2)	Eq. (B3)	
$P(\boldsymbol{b})$	Eq. (9)		Eq. (9)	Eq. (B4)	
$P(\boldsymbol{k}\mid\boldsymbol{e},\boldsymbol{b})$	Eq. (5)		Eq. (5)	–	–
Dense ensemble?	No		No	Yes	

APPENDIX B: PRIOR FOR EDGE COUNTS IN THE HIERARCHICAL BISBM

In this Appendix, we provide the prior for edge counts in the hierarchical bipartite SBM, corresponding to Eq. (19) and Eq. (20). We begin with the flat SBM, whose prior for edge counts is

$$P(\boldsymbol{e}|\boldsymbol{b}) = \left(\!\!\binom{\binom{B}{2}}{E}\!\!\right)^{-1}. \tag{B1}$$

In the hSBM, it might seem as if $P(\boldsymbol{e}\mid\boldsymbol{b})$ should be written as a product of SBM likelihoods by repeatedly reusing Eq. (2) at each additional level. However, at higher levels, the networks are multigraphs, and the SBM likelihood does not generate multigraphs uniformly because it is based on a uniform generation of configurations [i.e., $\Omega(\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$] [22]. Therefore, the correct way to build up the product is to directly count the number of multigraphs at each higher level using the dense ensemble [31], with each network instance occurring with the same probability.

Assuming that we have built L higher-level models, the prior for edge counts between groups can be rewritten as

$$P_{\text{hier}}(\boldsymbol{e}\mid\boldsymbol{b}) = \prod_{l=1}^{L} P(\boldsymbol{e}_l\mid\boldsymbol{e}_{l+1},\boldsymbol{b}_l)P(\boldsymbol{b}_l), \tag{B2}$$

where

$$P(\boldsymbol{e}_l\mid\boldsymbol{e}_{l+1},\boldsymbol{b}_l) = \prod_{r<s}\left(\!\!\binom{n_r^l n_s^l}{e_{rs}^{l+1}}\!\!\right)^{-1} \tag{B3}$$

and

$$P(\boldsymbol{b}_l) = \frac{\prod_r n_r^{l+1}!}{B_l!}\binom{B_l-1}{B_{l+1}-1}^{-1}\frac{1}{B_l}. \tag{B4}$$

At the highest level ($l = L$), $\boldsymbol{e}_L$ denotes a single-node multigraph with E self-loops. Because there is no further block structure, we enforce $P(\boldsymbol{e}_L\mid\boldsymbol{e}_{L+1},\boldsymbol{b}_l) = P(\boldsymbol{e}_L\mid\boldsymbol{b}_l)$ and assume that the block is generated by a uniform prior, and reuse Eq. (B1).

One peculiar consequence of forcing the hSBM, implemented in `graph-tool`, to consider only type-specific blocks is that even when a network has no statistically justifiable structure, the hSBM finds the trivial bipartite partition *and then* builds a final hierarchical level on that trivial bipartite partition. In other words, it cannot help but find a single group at the topmost level. This explains the otherwise perplexing distribution of model description lengths shown in Fig. 5(a): both the SBM and hSBM find the trivial partition, but this partition is more costly to express via the hSBM due to our having forced it to respect the network's bipartite structure. Table II summarizes all model likelihood and prior functions pertinent to this paper, for reference.

[1] J.-G. Young, F. S. Valdovinos, and M. E. J. Newman, Reconstruction of plant–pollinator networks from observational data bioRxiv 754077 (2019).

[2] T. Squartini, A. Almog, G. Caldarelli, I. van Lelyveld, D. Garlaschelli, and G. Cimini, Enhanced capital-asset pricing model for the reconstruction of bipartite financial networks, Phys. Rev. E **96**, 032315 (2017).

[3] R. Guimerà and M. Sales-Pardo, Justice blocks and predictability of U.S. Supreme Court votes, PLoS ONE **6**, e27188 (2011).

[4] G. Ghoshal, V. Zlatić, G. Caldarelli, and M. E. J. Newman, Random hypergraphs and their applications, Phys. Rev. E **79**, 066118 (2009).

[5] P. S. Chodrow, Configuration models of random hypergraphs, J. Compl. Netw. **8**, cnaa018 (2020).

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

[7] T. P. Peixoto, Bayesian stochastic blockmodeling, in *Advances in Network Clustering and Blockmodeling*, edited by P. Doreian, V. Batagelj, and A. Ferligoj (John Wiley & Sons, Hoboken, New Jersey, 2019).

[8] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, Mixed membership stochastic blockmodels, J. Mach. Learn. Res. **9**, 1981 (2008).

[9] A. Godoy-Lorite, R. Guimerà, C. Moore, and M. Sales-Pardo, Accurate and scalable social recommendation using mixed-membership stochastic block models, Proc. Natl. Acad. Sci. USA **113**, 14207 (2016).

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

[11] M. Tarrés-Deulofeu, A. Godoy-Lorite, R. Guimerà, and M. Sales-Pardo, Tensorial and bipartite block models for link prediction in layered networks and temporal networks, Phys. Rev. E **99**, 032307 (2019).

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

[13] D. Hric, T. P. Peixoto, and S. Fortunato, Network Structure, Metadata, and the Rrediction of Missing Nodes and Annotations, Phys. Rev. X **6**, 031038 (2016).

[14] M. E. J. Newman and A. Clauset, Structure and inference in annotated networks, Nat. Commun. **7**, 11863 (2016).

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

[16] M. E. J. Newman, Estimating network structure from unreliable measurements, Phys. Rev. E **98**, 062321 (2018).

[17] M. E. J. Newman, Network structure from rich but noisy data, Nat. Phys. **14**, 542 (2018).

[18] T. P. Peixoto, Reconstructing Networks with Unknown and Heterogeneous Errors, Phys. Rev. X **8**, 041011 (2018).

[19] J.-G. Young, G. St-Onge, P. Desrosiers, and L. J. Dubé, Universality of the stochastic block model, Phys. Rev. E **98**, 032309 (2018).

[20] S. C. Olhede and P. J. Wolfe, Network histograms and universality of blockmodel approximation, Proc. Natl. Acad. Sci. USA **111**, 14722 (2014).

[21] D. B. Larremore, A. Clauset, and A. Z. Jacobs, Efficiently inferring community structure in bipartite networks, Phys. Rev. E **90**, 012805 (2014).

[22] T. P. Peixoto, Nonparametric Bayesian inference of the microcanonical stochastic block model, Phys. Rev. E **95**, 012317 (2017).

[23] M. A. Riolo, G. T. Cantwell, G. Reinert, and M. E. J. Newman, Efficient method for estimating the number of communities in a network, Phys. Rev. E **96**, 032310 (2017).

[24] T. Kawamoto and Y. Kabashima, Cross-validation estimate of the number of clusters in a network, Sci. Rep. **7**, 1 (2017).

[25] T. Vallès-Català, T. P. Peixoto, M. Sales-Pardo, and R. Guimerà, Consistencies and inconsistencies between model selection and link prediction in networks, Phys. Rev. E **97**, 062316 (2018).

[26] T. P. Peixoto, Parsimonious Module Inference in Large Networks, Phys. Rev. Lett. **110**, 148701 (2013).

[27] T. Kawamoto and Y. Kabashima, Counting the number of metastable states in the modularity landscape: Algorithmic detectability limit of greedy algorithms in community detection, Phys. Rev. E **99**, 010301(R) (2019).

[28] J. Calatayud, R. Bernardo-Madrid, M. Neuman, A. Rojas, and M. Rosvall, Exploring the solution landscape enables more reliable network community detection, Phys. Rev. E **100**, 052308 (2019).

[29] B. Bollobás, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, Eur. J. Combin. **1**, 311 (1980).

[30] B. K. Fosdick, D. B. Larremore, J. Nishimura, and J. Ugander, Configuring random graph models with fixed degree sequences, SIAM Rev. **60**, 315 (2018).

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

[32] J. Rissanen, *Information and Complexity in Statistical Modeling* (Springer, New York, 2007).

[33] P. D. Grünwald, *The Minimum Description Length Principle* (MIT Press, Cambridge, MA, 2007).

[34] P. D. Grünwald and T. Roos, Minimum description length revisited, Intl. J. Math. Industry **11**, 1930001 (2019).

[35] G. Schwarz, Estimating the dimension of a model, Ann. Stat. **6**, 461 (1978).

[36] X. Yan, C. Shalizi, J. E. Jensen, F. Krzakala, C. Moore, L. Zdeborová, P. Zhang, and Y. Zhu, Model selection for degree-corrected block models, J. Stat. Mech. (2014) P05007.

[37] G. E. Andrews, *The Theory of Partitions* (Cambridge University Press, Cambridge, 1998).

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).

[39] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika **57**, 97 (1970).

[40] T. P. Peixoto, Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models, Phys. Rev. E **89**, 012804 (2014).

[41] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. (MIT Press, Cambridge, MA, 2009).

[42] J. Erickson, *Algorithms* (independently published, S.L., 2019).

[43] B. W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. **49**, 291 (1970).

[44] T.-C. Yen, The bipartiteSBM Python library, https://github.com/junipertcy/bipartiteSBM (2020).

[45] D. B. Larremore, A. Clauset, and C. O. Buckee, A network approach to analyzing highly recombinant malaria parasite genes, PLoS Comput. Biol. **9**, e1003268 (2013).

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

[47] A. Condon and R. M. Karp, Algorithms for graph partitioning on the planted partition model, Random Struct. Algor. **18**, 116 (2001).

[48] T. P. Peixoto, The graph-tool Python library, 10.6084/m9.figshare **1164194** (2014), https://graph-tool.skewed.de.

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

[50] E. Mossel, J. Neeman, and A. Sly, Reconstruction and estimation in the planted partition model, Probab. Theory Relat. Fields **162**, 431 (2015).

[51] T. Kawamoto and Y. Kabashima, Detectability thresholds of general modular graphs, Phys. Rev. E **95**, 012304 (2017).

[52] F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, Typology of phase transitions in Bayesian inference problems, Phys. Rev. E **99**, 042109 (2019).

[53] A. Clauset, E. Tucker, and M. Sainz, The Colorado Index of Complex Networks, https://icon.colorado.edu/ (2016).

[54] L. W. Jones, A. Davis, B. B. Gardner, and M. R. Gardner, Deep south: A social anthropological study of caste and class, South. Econ. J. **9**, 159 (1942).

[55] A. Joern, Feeding patterns in grasshoppers (Orthoptera: Acrididae): Factors influencing diet specialization, Oecologia **38**, 325 (1979).

[56] A.-M. Niekamp, L. A. G. Mercken, C. J. P. A. Hoebe, and N. H. T. M. Dukers-Muijrers, A sexual affiliation network of swingers, heterosexuals practicing risk behaviours that potentiate the spread of sexually transmitted infections: A two-mode approach, Social Netw. **35**, 223 (2013).

[57] C. K. McMullen, Flower-visiting insects of the Galápagos Islands, Pan-Pacific Entomologist **69**, 95 (1993).

[58] T. di Milano, Ordinanza di applicazione di misura coercitiva con mandato di cattura-art. (Operazione Infinito), Ufficio del giudice per le indagini preliminari (2011).

[59] L. M. Gerdes, K. Ringler, and B. Autin, Assessing the Abu Sayyaf group's strategic and learning capacities, Stud. Conflict Terrorism **37**, 267 (2014).

[60] O. Rozenblatt-Rosen, R. C. Deo, M. Padi, G. Adelmant, M. A. Calderwood, T. Rolland, M. Grace, A. Dricot, M. Askenazi, M. Tavares *et al.*, Interpreting cancer genomes using systematic host network perturbations by tumour virus proteins, Nature (London) **487**, 491 (2012).

[61] F. E. Clements and F. L. Long, *Experimental Pollination: An Outline of the Ecology of Flowers and Insects*, 336 (Carnegie Institution of Washington, Washington, DC, 1923).

[62] A. C. Murphy, S. F. Muldoon, D. Baker, A. Lastowka, B. Bennett, M. Yang, and D. S. Bassett, Structure, function, and control of the human musculoskeletal network, PLoS Biol. **16**, e2002811 (2018).

[63] M. Coscia and V. Rios, Knowing where and how criminal organizations operate using web content, in Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12 (ACM Press, Maui, Hawaii, 2012), pp. 1412–1421.

[64] J. Kunegis, KONECT: The Koblenz network collection, in *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13 Companion* (ACM Press, Rio de Janeiro, 2013), pp. 1343–1350.

[65] J. C. Nacher and J.-M. Schwartz, Modularity in protein complex and drug interactions reveals new polypharmacological properties, PLoS ONE **7**, e30028 (2012).

[66] C. Robertson, *Flowers and Insects; Lists of Visitors of Four Hundred and Fifty-three Flowers* (Science Press, Lancaster, PA, USA, 1929).

[67] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási, The human disease network, Proc. Natl. Acad. Sci. USA **104**, 8685 (2007).

[68] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, Flavor network and the principles of food pairing, Sci. Rep. **1**, 196 (2011).

[69] M. Gerlach, T. P. Peixoto, and E. G. Altmann, A network approach to topic models, Sci. Adv. **4**, eaaq1360 (2018).

[70] D. Yang, D. Zhang, Z. Yu, and Z. Yu, Fine-grained preference-aware location search leveraging crowdsourced digital footprints from LBSNs, in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13 (ACM Press, Zurich, 2013), pp. 479–488.

[71] J. E. Goldford, H. Hartman, T. F. Smith, and D. Segrè, Remnants of an ancient metabolism without phosphate, Cell **168**, 1126 (2017).

[72] R. Alberich, J. Miro-Julia, and F. Rossello, Marvel Universe looks almost like a real social network, arXiv:cond-mat/0202174.

[73] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, RCV1: A new benchmark collection for text categorization research, J. Mach. Learn. Res. **5**, 361 (2004).

[74] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, Measurement and analysis of online social networks, in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC '07* (ACM Press, San Diego, 2007), pp. 29–42.

[75] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, DBpedia: A nucleus for a web of open data, in *The Semantic Web*, edited by K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Lecture Notes in Computer Science (Springer, Berlin, 2007), Vol. 4825, pp. 722–735.

[76] S. H. Jeffreys, *The Theory of Probability* (Oxford University Press, New York, 1998).

[77] See, https://www.imdb.com/interfaces.

[78] A. Ghasemian, H. Hosseinmardi, and A. Clauset, Evaluating overfit and underfit in models of network community structure, IEEE Trans. Knowl. Data Eng. **32**, 1722 (2019).