Message-passing on hypergraphs: detectability, phase transitions and higher-order information

Author(s):
Ruggeri, Nicolò; Lonardi, Alessandro; De Bacco, Caterina

Publication date:
2024-04

Permanent link:
https://doi.org/10.3929/ethz-b-000670511

Originally published in:
Journal of Statistical Mechanics: Theory and Experiment 2024(4), https://doi.org/10.1088/1742-5468/ad343b

OPEN ACCESS

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

PAPER: Interdisciplinary statistical mechanics

Message-passing on hypergraphs: detectability, phase transitions and higher-order information

Nicolò Ruggeri[1,2,3], **Alessandro Lonardi**[1,3,*] **and Caterina De Bacco**[1]

[1] Max Planck Institute for Intelligent Systems, Cyber Valley, Tübingen 72076, Germany
[2] Department of Computer Science, ETH, Zürich 8004, Switzerland
E-mail: alessandro.lonardi@tuebingen.mpg.de, nicolo.ruggeri@tuebingen.mpg.de and caterina.debacco@tuebingen.mpg.de

Received 8 January 2024
Accepted for publication 6 March 2024
Published 23 April 2024

Online at stacks.iop.org/JSTAT/2024/043403
https://doi.org/10.1088/1742-5468/ad343b

Abstract. Hypergraphs are widely adopted tools to examine systems with higher-order interactions. Despite recent advancements in methods for community detection in these systems, we still lack a theoretical analysis of their detectability limits. Here, we derive closed-form bounds for community detection in hypergraphs. Using a message-passing formulation, we demonstrate that detectability depends on the hypergraphs' structural properties, such as the distribution of hyperedge sizes or their assortativity. Our formulation enables a characterization of the entropy of a hypergraph in relation to that of its clique expansion, showing that community detection is enhanced when hyperedges highly overlap on pairs of nodes. We develop an efficient message-passing algorithm to learn communities and model parameters on large systems. Additionally, we devise an exact sampling routine to generate synthetic data from our probabilistic model. Using these methods, we numerically investigate the boundaries of

[3] Equal contribution.
[*] Author to whom any correspondence should be addressed.

community detection in synthetic datasets, and extract communities from real systems. Our results extend our understanding of the limits of community detection in hypergraphs and introduce flexible mathematical tools to study systems with higher-order interactions.

Keywords: inference of graphical models, message-passing algorithms, statistical inference

Contents

J. Stat. Mech. (2024) 043403

1. Introduction

Modeling complex systems as graphs has broadened our understanding of the macroscopic features that emerge from the interaction of individual units. Among the various aspects of this problem, community detection stands out as a fundamental task, as it provides a coarse-grained description of a network's structural organization. Notably, community structure is observed across different systems, such as food webs [1], spatial migration and the gene flow of animal species [2], as well as in social networks [3], power grids [4] and other systems [5].

In the case of networks with only pairwise interactions, there are solid theoretical results on detectability limits, describing whether the task of community detection can or cannot succeed [6–11]. However, many complex systems with interactions that extend beyond pairs are better modeled by hypergraphs [12], which generalize the simpler case of dyadic graphs. Phenomena that have been investigated on graphs are now readily explored on hypergraphs, with examples including diffusion processes, synchronization, phase transitions [13] and, more recently, community structure [14–18].

Extending the rigorous results of detectability transitions for networks to higher-order interactions is a relevant open question.

One of the main obstacles in modeling hypergraphs is their intrinsic complexity, which poses both theoretical and computational challenges and restricts the range of results available in the literature. The difficulty of defining communities in hypergraphs and of deriving theoretical thresholds for their recovery has limited investigations in the study of d-uniform hypergraphs, i.e. hypergraphs that only contain interactions among exactly d nodes [19–27].

A related line of literature focuses on the detection of planted sub-hypergraphs [28, 29] and testing for the presence of community structure in hypergraphs [30, 31].

J. Stat. Mech. (2024) 043403

Generally, extracting recovery results on non-uniform hypergraphs has proved to be demanding, with scarce literature on the subject.

Recently, Chodrow *et al* [32] conjectured a recoverability threshold for their spectral clustering algorithm on non-uniform hypergraphs. Closer to the scope of our work, Dumitriu and Wang [18] provided a probabilistic model and bounds for the theoretical recovery of communities under the same model. However, such detectability bounds are based on algorithms that are not feasible in practice, and no empirical demonstration of the predicted recovery is provided. Furthermore, all these methods lack a variety of desirable probabilistic features, such as the estimation of marginal probabilities of a node to belong to a community, a principled procedure to sample synthetic hypergraphs with prescribed community structure, and the possibility to investigate the energy landscape of a problem via free energy estimations.

In this work, we address these issues by deriving a precise detectability threshold for hypergraphs that depends on the node degree distribution, the assortativity of the hyperedges, and crucially, on higher-order properties such as the distribution of hyperedge sizes. Additionally, we show how these properties can be formally described via notions of entropy and information, leading to a clear interpretation of the role of higher-order interaction in detectability.

Our approach is based on a probabilistic generative model and a related Bayesian inference procedure, which we utilize to study the limits of the community detection problem using a message-passing (MP) formulation [33–35], originating from the cavity method in statistical physics [36, 37]. We focus on an extension to hypergraphs of the stochastic block model (SBM) [38, 39], a generative model for networks with community structure. Several variants of the SBM [15], and of its mixed-membership version [16, 17], have been extended to hypergraphs. The model we utilize is an extension of the dyadic SBM to hypergraphs and allows generalizing the seminal detectability results of Decelle *et al* [6, 7] to higher-order interactions.

In addition to our theoretical contributions, we derive an algorithmic implementation for inferring both communities and parameters of the models from the data. Our implementation scales well to both large hypergraphs and large hyperedges, owing to a dynamic-program formulation.

Finally, we show how, with additional combinatorial arguments, one can efficiently sample hypergraphs with arbitrary communities from our probabilistic model. This problem, often studied in conjunction with inference, deserves its own attention when dealing with hypergraphs, as recently discussed in related work [40, 41].

Through numerical experiments, we confirm our theoretical calculations by showing that our algorithm accurately recovers the true community structure in synthetic hypergraphs all the way down to the predicted detectability threshold. We also illustrate that our approach gives insights into the community organization of real hypergraphs by analyzing a dataset of group interactions between students in a school. To facilitate reproducibility, we release the code that implements our inference and sampling procedures open source [42].

J. Stat. Mech. (2024) 043403

2. The hypergraph stochastic block model (HySBM)

Consider a hypergraph $H = (V, E)$, where $V = \{1, \ldots, N\}$ is the set of nodes and E the set of hyperedges. A hyperedge e is a set of two or more nodes. We define $\Omega = \{e : 2 \leqslant |e| \leqslant D\}$, the set of all possible hyperedges up to some maximum dimension $D \leqslant N$, with $|e|$ being the size of a hyperedge, i.e. the number of nodes it contains. Notice that $E \subseteq \Omega$. We denote with $A_e = 1$ all $e \in E$ and with $A_e = 0$ hyperedges $e \in \Omega \setminus E$.

Our HySBM is an extension of the classical SBM for graphs [38, 39]. It partitions nodes into K communities by assigning a hard membership $t_i \in [K] \equiv \{1, \ldots, K\}$ to each node $i \in V$, with $t = \{t_i\}_{i \in V}$ being the membership vector. It does so probabilistically, assuming that the likelihood of observing a hyperedge A_e is a Bernoulli distribution with a parameter that depends on the memberships $\{t_i\}_{i \in e}$ of its nodes. Formally, the probabilistic model is summarized as

$$t_i \sim \mathrm{Cat}\,(n) \quad \forall i \in V \tag{1}$$

$$A_e \,|\, t \sim \mathrm{Be}\left(\frac{\pi_e}{\kappa_{|e|}}\right) \quad \forall e \in \Omega\,, \tag{2}$$

where $n = (n_1, \ldots, n_K)$ is a vector of prior categorical probabilities for the hard assignments t_i. The Bernoulli probabilities are given by

$$\pi_e = \sum_{i<j \in e} p_{t_i t_j}\,, \tag{3}$$

with $0 \leqslant p_{ab} \leqslant 1$ being elements of a symmetric probability matrix (also referred to as an affinity matrix) and $\kappa_{|e|}$ a normalizing constant that only depends on the hyperedge size $|e|$. This can take on any value, provided that it yields sparse hypergraphs where $\pi_e/\kappa_{|e|} = O(1/N)$ and valid probabilities $\pi_e/\kappa_{|e|}$. We develop our theory for a general form of $\kappa_{|e|}$ and elaborate more on its choice in appendix A. In our experiments, we utilize the value $\kappa_d = \binom{N-2}{d-2}\frac{d(d-1)}{2}$ [17, 41].

Our specific formulation of the likelihood is only one among many alternatives to model communities in hypergraphs. The likelihood we propose has three main properties. First, the HySBM reduces to the standard SBM when only pairs are present (as $\kappa_2 = 1$). Since we aim to develop a model that generalizes the SBM to hypergraphs, this is an important condition to satisfy. Second, it enables us to develop the MP equations presented in the following section, which in turn leads to a theoretical characterization of the detectability limits and a computationally efficient algorithmic implementation. Third, the likelihoods based on expressions similar to equation (3) have been shown to accurately describe higher-order interactions that possibly contain nodes from different communities [41].

For convenience, we work with a rescaled affinity matrix $c = Np$, which is of order $c = O(1)$ in sparse hypergraphs. The log-likelihood $\mathcal{L} \equiv \mathcal{L}(A, t \,|\, p, n)$ evaluates to

J. Stat. Mech. (2024) 043403

$$\begin{aligned}\mathcal{L} &= \sum_{e\in\Omega}\left[A_e \log\left(\frac{\pi_e}{\kappa_e}\right) + (1-A_e)\log\left(1-\frac{\pi_e}{\kappa_e}\right)\right] + \sum_{i\in V}\log n_{t_i} \\ &= \sum_{e\in\Omega}\left[A_e \log\left(\sum_{i<j\in e} c_{t_i t_j}\right) + (1-A_e)\log\left(1-\frac{\sum_{i<j\in e} c_{t_i t_j}}{N\kappa_e}\right)\right] + \sum_{i\in V}\log n_{t_i} + \text{const.}\,,\end{aligned} \tag{4}$$

where const. denotes quantities that do not depend on the parameters of the model.

3. Inference and generative modeling

3.1. Induced factor graph representation

The probabilistic model in equations (1) and (2) has a negative log-likelihood that can be interpreted as the Hamiltonian of a Gibbs–Boltzmann distribution on the community assignments t:

$$p(t\,|\,A,p,n) = \frac{p(A,t\,|\,p,n)}{p(A\,|\,p,n)} = \frac{\exp\mathcal{L}(A,t\,|\,p,n)}{Z}\,, \tag{5}$$

where Z is the partition function of the system, which corresponds to the marginal likelihood of the data. The quantity $F = -\log Z$ is also called the free energy. The equivalence in equation (5) allows interpreting the probabilistic model in terms of factor graphs [34]. Here, the function nodes are hyperedges $f \in \Omega$, and variable nodes are elements of V. The interactions between function and variable nodes can be read directly from the log-likelihood in equation (4). In other words, the probabilistic model induces a factor graph $F = (\mathcal{V},\mathcal{F},\mathcal{E})$ with variable nodes $\mathcal{V} = V$, function nodes $\mathcal{F} = \Omega$ and edges $\mathcal{E} = \{(i,e) \in \mathcal{V}\times\mathcal{F} : i \in e\}$. In figure 1 we show a graphical representation of the equivalence between hypergraphs and factor graphs. For any variable node i and function node f of the factor graph we define the neighbors, or boundaries, as $\partial i = \{f \in \mathcal{F} : (i,e) \in \mathcal{E}\}$, being all function nodes adjacent to i, and $\partial f = \{i \in \mathcal{V} : (i,e) \in \mathcal{E}\}$ being all variable nodes adjacent to f.

3.2. Message-passing (MP)

Given the factor graph representation of HySBM, we can perform Bayesian inference of the community assignments via MP. Originally obtained from the cavity method on spin glasses [36, 37], MP allows estimating marginal distributions on the variable nodes of a graphical model by iteratively updating messages, auxiliary variables that operate on the edges of the factor graph. The efficiency of MP comes from the fact that the structure of the factor graph favors locally distributed updates. Although exact theoretical results are only proven on trees, MP has been shown to obtain a strong performance also on locally tree-like graphs [34], and it has been extended to dense graphs with short loops [43, 44].

Applying MP to our model, the inference procedure yields expressions for the marginal probabilities $q_i(a)$ of a node i to be assigned to any given community $a \in [K]$. Their

J. Stat. Mech. (2024) 043403

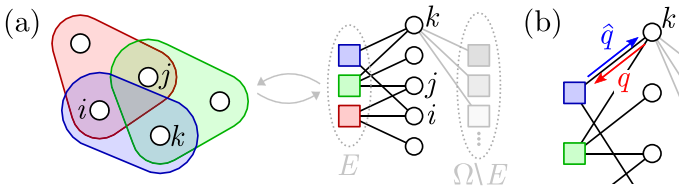

Figure 1. Representing hypergraphs as factor graphs. (a) We depict a hypergraph and its factor graph equivalent. In the factor graph $\mathcal{F}$, function nodes represent hyperedges. Notice that, while the node sets are the same in both representations, due to the presence of all possible hyperedges in the log-likelihood in equation (4), the factor graph not only contains the observed interactions E (black), but also the unobserved ones $\Omega \setminus E$ (gray). (b) In factor graphs, there are two types of messages: variable-to-function node q (red), and function-to-variable node $\hat{q}$ (blue).

values are obtained as solutions to closed-form fixed-point equations, which involve messages $q_{i\to e}(t_i)$ from variable to function nodes, and $\widehat{q}_{e\to i}(t_i)$ from function to variable nodes. The messages follow the sum-product updates

$$q_{i\to e}(t_i) \propto n_{t_i} \prod_{f\in\partial i\setminus e} \widehat{q}_{f\to i}(t_i) \tag{6}$$

$$\widehat{q}_{e\to i}(t_i) \propto \sum_{t_j : j\in\partial e\setminus i} \left(\frac{\pi_e}{\kappa_e}\right)^{A_e} \left(1-\frac{\pi_e}{\kappa_e}\right)^{1-A_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)\,, \tag{7}$$

and yield marginal distributions as

$$q_i(t_i) \propto n_{t_i} \prod_{e\in\partial i} \widehat{q}_{e\to i}(t_i)\,. \tag{8}$$

Notice that, compared to those for graphs, the MP equations for hypergraphs in equations (6)–(8) present additional challenges. First, in graphs, the updates can be simplified. One can in fact collapse the two types of messages (and equations) into a unique one, since paths (i,f,j) in the factor graph reduce to pairwise interactions (i,j) between nodes. This simplification is not possible in hypergraphs, as one function node may connect more than two variable nodes. Second, the dimensionality of the MP equations grows faster when accounting for higher-order interactions. Here, the number of function nodes is equal to $|\mathcal{F}| = |\Omega| = \sum_{d=2}^{D} \binom{N}{d}$, yielding $|\mathcal{F}| = O(2^N)$ at large $D = N$. In contrast, one gets $O(N^2)$ pairwise messages in the updates for graphs. To produce computationally feasible MP updates one can assume sparsity, as already done in the dyadic case. We outline such updates in the following theorem.

J. Stat. Mech. (2024) 043403

Theorem 1. *Assuming sparse hypergraphs where $c = O(1)$, the MP updates satisfy the following fixed-point equations to leading order in N. For all hyperedges $e \in E$ and nodes $i \in e$, the messages and marginals are given by*

$$q_{i\to e}(t_i) \propto n_{t_i} \left(\prod_{\substack{f\in E \\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i) \right) \exp\left(-h(t_i)\right) \tag{9}$$

$$\widehat{q}_{e\to i}(t_i) \propto \sum_{t_j : j\in\partial e\setminus i} \pi_e \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \tag{10}$$

$$q_i(t_i) \propto n_{t_i} \left(\prod_{\substack{f\in E \\ f\in\partial i}} \widehat{q}_{f\to i}(t_i) \right) \exp\left(-h(t_i)\right) \tag{11}$$

$$h(t_i) = \frac{C'}{N} \sum_{j\in V} \sum_{t_j} c_{t_i t_j} q_j(t_j)\,, \tag{12}$$

where $C' = \sum_{d=2}^{D} \binom{N-2}{d-2} \frac{1}{\kappa_d}$.

A proof of theorem 1 is provided in appendix B. The updates in equations (9)–(12) are in principle computationally feasible, as products of function nodes $f \in E$ have replaced products over the entire space $f \in \Omega$. In sparse graphs, which we observe in many real datasets, E is much smaller than the original Ω, thus significantly decreasing the computation cost. An intuitive justification of theorem 1, which we formalize in its proof, is that the observed interactions $f \in E$ hold most of the weight in the updates of their neighbors, while the unobserved ones $f \in \Omega \setminus E$ send approximately constant messages and thus can be absorbed in the external field h introduced in equation (12). This idea is inspired by the dyadic MP equations in Decelle *et al* [6]. However, in contrast to MP on graphs, a vanilla implementation of the updates is still not scalable in hypergraphs, as the computational cost of equation (10) is $O(K^{|e|-1})$. To tackle this issue, we develop a dynamic programming approach that reduces the complexity to $O(K^2|e|)$. Dynamic programming is exact, as it does not rely on further approximations of the MP updates, and its detailed derivations are provided in appendix D.1.

The fixed-point equations of theorem 1 naturally suggest an algorithmic implementation of the MP inference procedure. We present a pseudocode for it in appendix D.2.

3.3. Expectation–Maximization to learn the model parameters

We have presented an MP routine for inferring the community assignments $\{t_i\}_{i\in V}$. Now, we derive closed-form updates for the model parameters c, n via an Expectation–Maximization (EM) routine [45]. Differentiating the log-likelihood in equation (4) with respect to n, and imposing the constraint $\sum_{a=1}^{K} n_a = 1$, yields the update

$$n_a = \frac{N_a}{N}\,. \tag{13}$$

Notice that this update depends on the MP results, as $N_a = |\{i \in V : \arg\max_b q_i(b) = a\}|$ is the count of nodes assigned to a community a according to the inferred marginals. To update the rescaled affinity c we adopt a variational approach, where we maximize a lower bound of the log-likelihood, or, equivalently, minimize the variational free energy. In appendix C, we show detailed derivations for the following fixed-point updates:

$$c_{ab}^{(t+1)} = c_{ab}^{(t)} \frac{2\sum_{e\in E} \#_{ab}^{e}/\pi_e}{NC'\left(Nn_a n_b - \delta_{ab} n_a\right)}, \tag{14}$$

where $\#_{ab}^{e} = \sum_{i<j\in e} \delta_{t_i a}\delta_{t_j b}$ is the count of dyadic interactions between two communities a, b within a hyperedge e. In practice, when inferring t, n, c, one proceeds by alternating the MP inference of t, as presented in section 3.2, with the updates of c and n in equations (13) and (14) until convergence. A pseudocode for the EM procedure is presented in appendix D.2.

3.4. Sampling from the generative model

One of the main advantages of using a probabilistic formulation is the ability to generate data with the desired community structure. Among other tasks, this can be used in particular to test detectability results like the ones we theoretically derive in the following section. However, in hypergraphs, writing a probabilistic model does not directly imply the ability to sample from it, as is typically the case for graphs [40, 41]. In fact, while the $O(N^2)$ configuration space of graphs allows performing sampling explicitly, in the case of hypergraphs the exploding configuration space Ω makes this task prohibitive, even for hypergraphs with a moderate number of nodes and hyperedge sizes.

We propose a sampling algorithm that can efficiently scale and produce hypergraphs of dimensions in the tens or hundreds of thousands of nodes. We exploit the hard-membership nature of the assignments to obtain exact sampling via combinatorial arguments, as opposed to the approximate sampling in recent work for mixed-membership models [41]. The key observation to obtain an efficient algorithm is that the hyperedge probabilities do not depend on the nodes they contain, but only on their community assignments, as implied by equation (3).

With this in mind, we define the auxiliary quantity

$$\#_a^e = \sum_{i\in e} \delta_{t_i a}, \tag{15}$$

for a hyperedge e and community $a \in [K]$, which is the count of nodes in e that belong to a community a. Crucially, the hyperedge probability depends only on these counts:

$$\pi_e = \sum_{a<b\in[K]} \#_a^e \#_b^e p_{ab} + \sum_{a\in[K]} \frac{\#_a^e(\#_a^e - 1)}{2} p_{aa}. \tag{16}$$

Therefore, all hyperedges with different nodes, but the same counts $\#_1^e, \ldots, \#_K^e$, have equal probability.

J. Stat. Mech. (2024) 043403

Using equation (16), we sample hypergraphs as in algorithm 1 with the following steps:

(i) Iterate over the combinations.
For hyperedges of size $d=2$, sample all the $N(N-1)/2$ edges directly. Otherwise, iterate steps (ii)–(iv) for the hyperedge sizes $d=3,\ldots,D$ and vectors $\#=(\#_1,\ldots,\#_K)$ of community counts (where we omitted the superscript e to highlight that the same counts yield an identical equation (16)), satisfying $\sum_{a=1}^{K}\#_a=d$.

(ii) Compute the probability.
For a given count vector $\#$, the hyperedge probability $\pi_{\#}$ is given in equation (16). Notice that there are $N_{\#}=\binom{N_1}{\#_1}\cdot\ldots\cdot\binom{N_K}{\#_K}$ hyperedges satisfying the count $\#$, since we can choose $\#_a$ nodes from the N_a nodes in each community a.

(iii) Sample the number of hyperedges.
Importantly, we do not sample the individual hyperedges, but the *number* of observed hyperedges. Since the individual hyperedges are independent Bernoulli variables with the same probability, their sum X follows a binomial distribution:

$$X\sim\mathrm{Binom}\left(N_{\#},\frac{\pi_{\#}}{\kappa_d}\right) \tag{17}$$

with probability $\pi_{\#}$ fixed, determined by $\#$, and number of realizations $N_{\#}$. Sampling directly from equation (17) is numerically challenging for large $N_{\#}$ and κ_d, hence we adopt a series of numerical approximations summarized in appendix E.1.

(iv) Sample the hyperedges.
Given the count X of hyperedges sampled from equation (17), we can sample the hyperedges. This operation is performed by independently sampling X times $\#_a$ nodes from each community a. Note that this procedure might yield repeated hyperedges, which are not allowed. In sparse regimes, this event has low probability [46]. As a sensible approximation, we delete repeated hyperedges.

Due to this sampling procedure, our results are not limited to theoretical derivations, but can be tested numerically on synthetic data, as we show in appendix E.2. In appendix E.1 we give a detailed analysis of the complexity, which is asymptotically upper bounded by $O(N\log N)$. A pseudocode for this procedure is shown in algorithm 1, and we provide an open source implementation of the sampling procedure [42].

J. Stat. Mech. (2024) 043403

Algorithm 1. Sampling hypergraphs.

Inputs: D, maximum size of hyperedges
N, number of nodes
K, number of communities
n, prior of the community memberships
p, affinity matrix

sample node memberships using equation (1)
for $d = 2, \ldots, D$ **do** ▷ (i)
  if $d = 2$ **then**
    sample $N(N-1)/2$ (hyper)edges from equation (2)
  else
    for each $\# = (\#_1, \ldots, \#_K)$ such that $\sum_{a=1}^{K} \#_a = d$ **do** ▷ (i)
      compute $\pi_{\#}$ with equation (16) ▷ (ii)
      sample X from equation (17) ▷ (iii)
      for $a = 1, \ldots, K$ **do**
        sample X times $\#_a$ nodes ▷ (iv)
      end for
    end for
    delete repeated hyperedges
  end if
end for

4. Phase transition

4.1. Detectability bounds

Besides providing a valid and efficient inference algorithm, one of the main advantages of MP is the possibility of deriving closed-form expressions for the detectability of planted communities. The transition from detectable to undetectable regimes was first shown to exist in MP-based inference models for graphs [6], and gave rise to an extensive body of literature on theoretical detectability limits and sharp phase transitions [8, 9]. Here, we extend these classical arguments to hypergraphs, and find relevant differences when higher-order interactions are considered.

In line with previous work, we restrict our study to the case where groups have constant expected degrees. In fact, in settings where such an assumption does not hold, it is possible to obtain good classification by simply clustering nodes based on their degrees [6]. Formally, we assume

$$\sum_{b=1}^{K} c_{ab} n_b = c\,, \tag{18}$$

for some fixed constant c. Notice that equation (18) does not immediately imply a constant degree for the groups, as in hypergraphs the expected degree is defined differently than the left-hand side of the equation above. Nevertheless, in appendix F.1 we prove

J. Stat. Mech. (2024) 043403

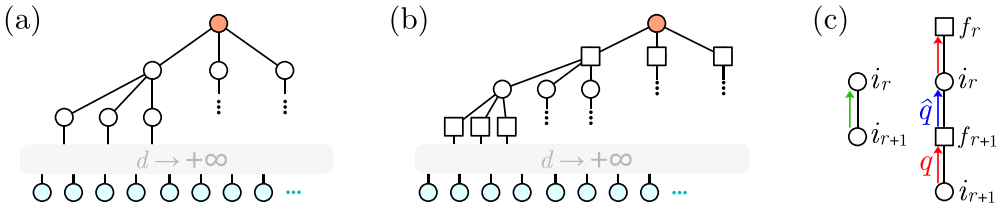

Figure 2. Local tree assumption. (a) The classical local tree assumption for graphs. Here, it is assumed that the neighborhoods of nodes are approximately trees. (b) The tree assumption for factor graphs. Here, a path from a leaf (light blue) to a root (orange) consists of steps alternating variable nodes and function nodes. These two representations coincide in the case of graphs. (c) The perturbations propagate up the tree via the messages. In graphs (a), they reach the root passing from nodes i_{r+1} to i_r (green). In hypergraph-induced factor graphs, perturbations spread from a node i_{r+1}, at depth $r+1$, to its neighboring function nodes f_{r+1} (red), and up to node i_r at depth r (blue) in an alternating fashion.

that imposing the condition in equation (18) does indeed imply a constant average degree. More precisely,

Proposition 1. *Assuming equation (18), the following holds:*

- *all the groups have the same expected degree;*
- *the fixed points for the messages read*

$$q_{i\to e}(t_i) = n_{t_i} \quad \forall e \in E, i \in e \tag{19}$$

$$\widehat{q}_{e\to i}(t_i) = \frac{1}{K} \quad \forall e \in E, i \in e\,. \tag{20}$$

We want to study the propagation of perturbations around the fixed points of equations (19) and (20). We assume that the factor graph is locally tree-like, i.e. neighborhoods of nodes are approximately trees. We provide a visualization of this in figure 2. Classically, it has been proven that for sparse graphs almost all nodes have local tree-like structures up to distances of order $O(\log N)$ [34]. We are not aware of similar statements for hypergraphs. While our empirical results prove that these assumptions are reasonable and approximately valid, we leave the formalization of such an argument for future work.

Referring to figure 2(b), one can see that between every leaf and the root, there is a single connecting path. Thus, perturbations on the leaves propagate through a tree to the root, and transmit via the following transition matrix:

$$\tilde{T}_r^{ab} = \frac{\partial q_{i_r\to f_r}(a)}{\partial q_{i_{r+1}\to f_{r+1}}(b)}\,, \tag{21}$$

where i_r, f_r are respectively the rth variable node and function node in the path. In other words, this is the dependency of a message on the message one level below in

the path. In appendix F.2 we show that, to leading terms in N, the transition matrix evaluates to

$$\tilde{T}_r^{ab} = \frac{2\,n_a}{|f_r|\,(|f_r|-1)}\left(\frac{c_{ab}}{c}-1\right). \tag{22}$$

A related expression previously obtained for the transition matrix on graphs is $T^{ab} = n_a\,(c_{ab}/c-1)$ [6]. Hence, we can compactly write $\tilde{T}_i^{ab} = [2/(|f_r|(|f_r|-1))]\,T^{ab}$. This connection highlights an important difference between the two cases: the hyperedges induce a higher-order prefactor with a 'dispersion' effect. The larger the hyperedge, the lower the magnitude of this transition. Instead, if the hyperedge is a pair, this prefactor reduces to one, and we recover the result on graphs. A perturbation $\epsilon_{t_d}^{k_d}$ of a leaf node k_d influences the perturbation $\epsilon_{t_0}^{k_0}$ on the root t_0 by

$$\epsilon_{t_0}^{k_0} = \sum_{\{t_r\}_{r=1,\dots,d}} \left(\prod_{r=0}^{d-1} \tilde{T}_i^{t_r t_{r+1}}\right) \epsilon_{t_d}^{k_d}. \tag{23}$$

We can also express this connection in matrix form as

$$\epsilon^{k_0} = \left(\prod_{r=0}^{d-1} \frac{2}{|f_r|\,(|f_r|-1)}\right) T^d \epsilon^{k_d}, \tag{24}$$

where T is the matrix with entries T^{ab} (in equation (24) raised to the power of d), and ϵ^{k_d} the array of $\epsilon_{t_d}^{k_d}$ values. Now, similarly to Decelle *et al* [6], we consider paths of length $d \to +\infty$. In this case, the r-dependent prefactor in equation (24) converges almost surely to

$$\mu = \exp\left(\mathbb{E}\left[d \log \frac{2}{|f|\,(|f|-1)}\right]\right), \tag{25}$$

where the expectation is taken with respect to randomly drawn hyperedges $f \in E$. If λ is the leading eigenvector of T, then

$$\epsilon^{k_0} \approx \mu\,\lambda^d \epsilon^{k_d}. \tag{26}$$

Aggregating over the leaves, and since the perturbations have an expected value of zero, we obtain the variance:

$$\langle (\epsilon_{t_0}^{k_0})^2 \rangle \approx \left\langle \left(\sum_{k=1}^{[d_0(F-1)]^d} \mu\,\lambda^d \epsilon_t^k \right)^2 \right\rangle \tag{27}$$

$$\overset{\text{i.i.d.}}{=} (d_0\,(F-1))^d\,\mu^2\,\lambda^{2d} \langle (\epsilon_t^k)^2 \rangle, \tag{28}$$

where d_0 is the average node degree and F the average hyperedge size. The expression in equation (28) yields the following stability criterion, the key result of our derivations:

$$d_0\,(F-1)\left(\exp\mathbb{E}\left[\log\frac{2}{|f|\,(|f|-1)}\right]\right)^2\lambda^2<1\,. \tag{29}$$

This generalizes the seminal result $c\lambda^2<1$ of Decelle *et al* [6] to hypergraphs. When equation (29) holds, the influence of the leaves on the root decays when propagating up the tree in figure 2(b). Conversely, if equation (29) is not satisfied, it grows exponentially.

To obtain more interpretable bounds, we focus on a benchmark scenario where the affinity matrix contains all equal on- and off-diagonal elements, i.e. $c_{aa}=c_{\rm in}$ for all $a\in[K]$ and $c_{ab}=c_{\rm out}$ for all $a\neq b$. In this case, condition equation (18) becomes $c_{\rm in}+(K-1)c_{\rm out}=Kc$, the leading eigenvalue of T is $\lambda=(c_{\rm in}-c_{\rm out})/Kc$, and the stability condition in equation (29) reads

$$|c_{\rm in}-c_{\rm out}|>\frac{Kc}{\sqrt{d_0\,(F-1)}}\exp\left(-\mathbb{E}\left[\log\frac{2}{|f|\,(|f|-1)}\right]\right). \tag{30}$$

When hypergraphs only contain dyadic interactions, equation (30) reduces to the bound $|c_{\rm in}-c_{\rm out}|>K\sqrt{c}$ previously derived for graphs [6], also known as the Kesten–Stigum bound [47, 48].

4.2. Phase transition in hypergraphs

We test the bound obtained in equation (30) by running MP on synthetic hypergraphs generated via the sampling algorithm of section 3.4. In our experiments, we fix $K=4$ and sample hypergraphs with $N=10^4$ nodes. We also fix $c=10$ and change the ratio $c_{\rm out}/c_{\rm in}$. In this setup, for graphs, one expects a continuous phase transition between two regimes where the system is undetectable and detectable [6]. In the former, where the inequality yielded by the Kesten–Stigum bound does not hold, and the graph does not carry sufficient information about the community assignments, community detection is impossible. In the latter, communities can be efficiently recovered by MP. In figure 3 we plot the overlap $=(\sum_i q_i^\star/N-\max_a n_a)/(1-\max_a n_a)$ with $q_i^\star\equiv q_i(a_i^\star)$ and $a_i^\star=\arg\max_b q_i(b)$, against $c_{\rm out}/c_{\rm in}$. Our results are in agreement with the theoretical predictions: the overlap is low in the undetectable region, high in the detectable region, and we observe a continuous phase transition at the Kesten–Stigum bound for graphs, i.e. when $D=2$.

We expect the presence of higher-order interactions to improve detectability, as this yields greater overlap for any $c_{\rm out}/c_{\rm in}$ and shifts the theoretical transition to larger values. We empirically validate this prediction by evaluating equation (30) for hyperedges up to size $D=50$ and performing MP inference in figure 3. Diverging convergence times for larger $c_{\rm out}/c_{\rm in}$, i.e. when the free energy landscape gets progressively rugged, further demonstrate this behavior, as shown in appendix F.3.

J. Stat. Mech. (2024) 043403

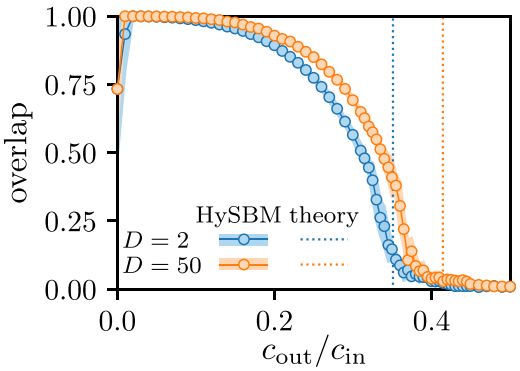

Figure 3. Phase transition. The overlap between ground truth and inferred communities varies for different $c_{\text{out}}/c_{\text{in}}$ ratios. The values attained are positive in the detectable region (left of the dotted theoretical bounds) and continuously drop to zero as the phase transition boundary approaches. Values for hyperedges up to size $D = 50$ (orange) always yield higher overlap compared to $D = 2$ (light blue). Shaded areas are standard deviations over five random initializations of MP.

4.3. The impact of higher-order interactions on detectability

As mentioned above, the transition matrix in equation (22) reduces to the classic T^{ab} [6] when only dyadic interactions are present. In fact, the additional prefactor $2/(|f_r|(|f_r|-1))$ is equal to one for two-dimensional hyperedges. However, when hyperedges of higher sizes are present, this prefactor is strictly smaller than one. This dampens the perturbations ϵ^{k_0} when they propagate up the tree in figure 2(b). It is unclear whether this higher-order effect aids or hinders detectability, as it could prevent signals from being propagated, but also noise from accumulating at the root.

With this in mind, we investigate the impact of higher-order interactions on detectability by disentangling the effect that K, c and, most importantly, D have on the detectability bound set by equation (29). To this end, we rewrite equation (30) as

$$\left|\rho_{\text{in}} - \frac{1}{Kc}\right| > \Phi\left(K, c, D\right). \tag{31}$$

Here, we utilized $c_{\text{in}}/Kc = \rho_{\text{in}} \in [0,1]$, a degree-independent rescaling of c_{in}, where we normalize by its maximum possible value Kc, as per equation (18). The term $\Phi(K,c,D)$ is the value of the theoretical bound on the rhs of equation (30), normalized by Kc as well. In this way, we obtain the decomposition $\Phi(K,c,D) = \alpha(K)\beta(c)\gamma(D)$ as a product of three independent terms:

$$\alpha\left(K\right) = \frac{K-1}{K} \tag{32}$$

$$\beta\left(c\right) = \frac{1}{\sqrt{c}} \tag{33}$$

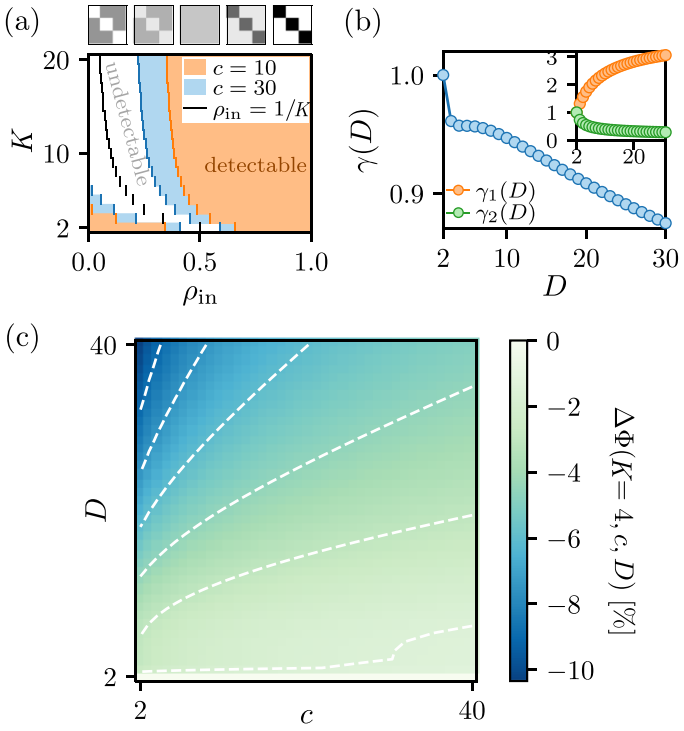

Figure 4. Theoretical phase transition. Due to the decomposition of our bound in equations (32)–(34) it is possible to separately describe the effects of K, c and D on the predicted phase transition. (a) Detectability bounds for networks ($D = 2$). Increasing c yields a broader range of detectable configurations (colored areas) for ρ_{in}. The number of communities skews detectability: while for $K = 2$ communities can be detected in extremely disassortative regimes (ρ_{in} close to zero), when more communities are present, only assortative networks are detectable. (b) Effect of the maximum hyperedge size D. The term $\gamma(D)$ in equation (34) can be split into the product $\gamma_1(D)\gamma_2(D)$, as defined in equations (35) and (36). The non-trivial decrease of $\gamma(D)$ results from the interplay of $\gamma_1(D)$ and $\gamma_2(D)$, having opposite monotonicity. (c) The percentage decrease $\Delta\Phi(K,c,D) = (\Phi(K,c,D) - \Phi(K,c,2))/\Phi(K,c,2)$ in detectability for different c, D values shows that higher-order interactions steadily improve detection, especially in sparse regimes.

$$\gamma(D) = \frac{\exp\left(-\mathbb{E}\left[\log \frac{2}{|f|(|f|-1)}\right]\right)}{\sqrt{C(F-1)/2}}, \tag{34}$$

where $C = \sum_{d=2}^{D} \binom{N-2}{d-2} \frac{d}{\kappa_d}$

In our experiments we choose $\kappa_d = \binom{N-2}{d-2} \frac{d(d-1)}{2}$, which conveniently returns $C = 2H_{D-1}$ (see appendix A), with H_{D-1} being the $(D-1)$th harmonic number. However, our theory holds true for any κ_d yielding sparse hypergraphs.

The classic effect of $\alpha(K)$ and $\beta(c)$ is summarized in figure 4(a), where the maximum hyperedge size is fixed to $D = 2$, hence $\gamma(D) = 1$. Here, we observe that the undetectability gap reduces when increasing c. Graphs with higher average degrees are more

J. Stat. Mech. (2024) 043403

detectable even when there is a larger inter-community mixing. The effect of larger K is that of skewing the detectability phase transition. This is because the edges contributing to c_{out} are spread over $K-1$ communities, while those accounting for c_{in} are concentrated in a single one. Intuitively, increasing K allows us to have more in–out edges, and detectability is still possible because of the dominating c_{in} term. The limit value $\rho_{\text{in}} = 1/K$ constitutes the perfect mixing case $c_{\text{in}} = c_{\text{out}} = c$, where detectability is unfeasible for any K and finite degree c. One should notice that, while the bounds drawn in figure 4 hold theoretically, for large K it may be exponentially hard to retrieve communities even in the detectable region [6, 49].

The higher-order effects on detectability are shown in figures 4(b) and (c). The presence of hyperedges with $D > 2$ enters in equation (34) as the product of two separate contributions, $\gamma(D) = \gamma_1(D)\gamma_2(D)$, where

$$\gamma_1(D) = \exp\left(-\mathbb{E}\left[\log \frac{2}{(|f|(|f|-1))}\right]\right) \tag{35}$$

$$\gamma_2(D) = \frac{1}{\sqrt{C(F-1)/2}}. \tag{36}$$

These two terms have contrasting effects that multiply to obtain the overall trend of $\gamma(D)$: $\gamma_1(D)$ is monotonically increasing while $\gamma_2(D)$ is monotonically decreasing. If we were to consider only the 'dispersion' contribution γ_1, we would enlarge the detectability gap by increasing Φ. However, the γ_2 term factors in the increasing number of interactions observed with larger hyperedges. The result is an overall higher-order contribution to detectability $\gamma(D) = \gamma_1(D)\gamma_2(D)$, where the value of γ_2 dominates over γ_1, giving rise to the non-trivial, monotonically decreasing, profile of figure 4(b).

The overall effect of higher-order terms is illustrated by plotting the relative difference $\Delta\Phi(K,c,D) = (\Phi(K,c,D) - \Phi(K,c,2))/\Phi(K,c,2)$ for a range of c and D values, with $K = 4$, as shown in figure 4(c). We observe how higher-order interactions lead to better detectability for all c, especially in sparse regimes, where c is small and pairwise information is not sufficient for the recovery of the communities.

4.4. Entropy and higher-order information

Hypergraphs are often compared against their clique decomposition, i.e. the graph obtained by projecting all hyperedges onto their pairwise connections, as a baseline network structure [50–52].

The clique decomposition yields highly dense graphs. For this reason, most theoretical results on sparse graphs are not directly applicable, algorithmic implementations become heavier—many times unfeasible—and storage in memory is suboptimal. Previous work also showed that algorithms developed for hypergraphs tend to work better in many practical scenarios [16]. Intuitively, hypergraphs 'are more informative' than graphs [53], as there exists only one clique decomposition induced by a given hypergraph, but possibly more hypergraphs corresponding to a given clique decomposition.

J. Stat. Mech. (2024) 043403

Here, we provide a theoretical basis to this common intuition and find that, within our framework, we can quantify the extra information carried by higher-order interactions.

For a given hypergraph $H = (V, E)$, edge $(i, j) \in V^2$ and hyperedge $e \in E$, we define the probability distribution

$$p_H(\{i,j\}, e) = \begin{cases} \dfrac{1}{E}\dfrac{2}{|e|(|e|-1)} & \text{if } i,j \in e \\ 0 & \text{otherwise}. \end{cases} \tag{37}$$

This distribution represents the joint probability of drawing a hyperedge uniformly at random among the possible E in the hypergraph and a dyadic interaction $\{i,j\}$ out of the possible $\binom{|e|}{2}$ within the hyperedge e. From equation (37) we can derive the following marginal distributions:

$$p_E(e) = \frac{1}{E} \tag{38}$$

$$p_{\mathcal{C}}(\{i,j\}) = \frac{1}{E} \sum_{e \in E : i,j \in e} \frac{2}{|e|(|e|-1)}, \tag{39}$$

for all $e \in E$ and pairs of nodes $i \neq j$. The distribution p_E is a uniform random draw of hyperedges. The distribution $p_{\mathcal{C}}$ represents the probability of drawing a weighted interaction $\{i,j\}$ in the clique decomposition of H.

With equations (37)–(39) at hand, it is possible to rewrite $\gamma_1(D)$ in equation (35) as

$$\log \gamma_1(D) = \mathcal{H}(\{i,j\} \mid f), \tag{40}$$

where $\mathcal{H}(\cdot|\cdot)$ is the conditional entropy. This entropy is minimized when $p_{\mathcal{C}}(\{i,j\})$ is very different than $p_H(\{i,j\}|f)$, i.e. when conditioning a pair $\{i,j\}$ to be in f brings additional information with respect to the interaction $\{i,j\}$ alone. This happens when $\{i,j\}$ appears in several hyperedges and it is difficult to reconstruct the hypergraph from its clique decomposition. As lower values of γ_1 imply easier recovery, equation (40) suggests that recovery is favored in hypergraphs where hyperedges overlap substantially and that cannot be easily distinguished from their clique decomposition.

We obtain a similar result by rewriting equation (40) as

$$\gamma_1(D) = \frac{\exp \mathcal{H}(p_H)}{\exp \mathcal{H}(p_E)} = \frac{\mathrm{PP}(p_H)}{\mathrm{PP}(p_E)}, \tag{41}$$

which is the ratio of two exponentiated entropies. In information theory, PP is referred to as perplexity [54], and it is an effective measure of the number of possible outcomes in a probability distribution [55]. Once we fix the number of hyperedges E (and therefore $\mathrm{PP}(p_E)$), the number of effective outcomes is given by the number of likely drawn $\{i,j\}$ pairs. This number is minimized when there is high overlap between hyperedges, thus confirming the interpretation of equation (40).

Finally, we set a different focus by rewriting γ_1 as

$$\log \gamma_1(D) = \mathcal{H}(p_{\mathcal{C}}) - \mathrm{KL}(p_H \,||\, p_{\mathcal{C}} \otimes p_E), \tag{42}$$

J. Stat. Mech. (2024) 043403

where KL is the Kullback–Leibler divergence and $\otimes$ the product probability distribution. Here, we pose the question: given a fixed clique decomposition and number of hyperedges, what is the hypergraph attaining the highest detectability? From the equation, such a hypergraph is that with the highest $\mathrm{KL}\left(p_H \,||\, p_{\mathcal{C}} \otimes p_E\right) = I(\{i,j\}, f)$. In this case, the KL divergence between a joint distribution and its marginals, also called the mutual information I [56] of the two random variables, describes the information shared between pairwise interactions and single hyperedges. Hypergraphs with high KL divergence, i.e. high information about a given $\{i,j\}$ in a single hyperedge f, will yield better detectability. In other words, it is preferable to choose hypergraphs that, while still producing the observed clique decomposition (thus achieving low entropy $\mathcal{H}(p_H)$), have largely overlapping hyperedges. The results discussed in this section provide theoretical guidance for the construction of hypergraphs that explain an observed graph made of only pairwise interactions [57], a problem relevant in datasets where higher-order interactions are not explicitly tracked.

5. Experiments on real data

Our model leads to a natural algorithmic implementation to learn communities in hypergraphs. In fact, alternating MP and EM rounds, our algorithm outputs marginal probabilities $q_i(t_i)$ for a node i to belong to a community t_i, as well as the community ratios n and the affinity matrix p. We illustrate an application of this procedure on a dataset of interactions between high school students (High School) [58]. Here, nodes are students, and hyperedges represent whether a group of students was observed in close proximity, as recorded by wearable devices. The hypergraph contains $N = 327$ nodes and $E = 7818$ hyperedges. In figure 5(a) we show the communities inferred on the dataset where only hyperedges up to size $D = 2,3,4$ are kept. We observe a clear progression in how the nodes are gradually allocated into different groups when higher-order interactions are progressively taken into account. This suggests that interactions beyond pairs carry information that would get lost if only edges were to be observed.

To get a qualitative interpretation, we compare the communities inferred with the nine classes attended by the students, an attribute available with the dataset. We illustrate the hypergraph of student interactions, coloring each node according to its class, in figure 5(b). Previous studies have shown that in this dataset a number of interactions occur with stronger prevalence within students of the same class [58]. In figure 5(c), we compare the communities inferred with different maximum hyperedge size D with the classes, and observe that there is a stronger alignment between them when larger hyperedges are utilized for inference. In figure 5(d) we show, at $D = 2,3,4$, the Normalized Mutual Information between inferred communities and class attributes, the area under the receiver operating curve (AUC) with respect to the full dataset, and the fraction ρ_D of hyperedges with size equal to D. In addition, our algorithm detects connection patterns that were previously observed between the different student classes as captured by the affinity matrix p; see appendix G.2 for details.

A feature that sets MP apart from other inference methods is the possibility to approximately compute the evidence $Z = p(A \,|\, p, n)$ of the whole dataset, or, equivalently, the free energy $F = -\log Z$. In appendix G we discuss how to make the free energy

J. Stat. Mech. (2024) 043403

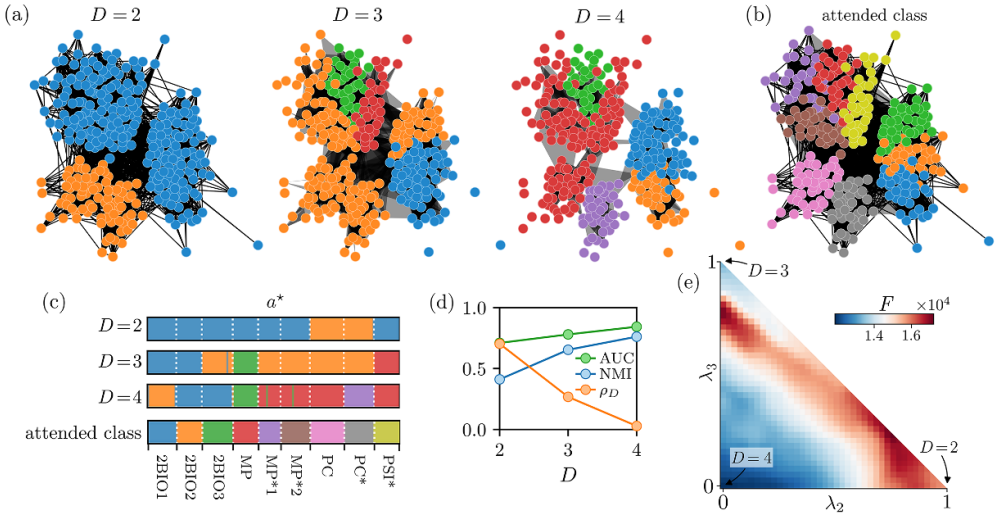

Figure 5. Experiments on the High School dataset. We infer the communities via MP and EM on the High School dataset. In all cases, we run inference with $K=10$ communities. (a) Inferred communities on the High School dataset, only utilizing hyperedges up to a maximum size D. Taking into account higher-order information, up to $D=4$, results in more granular partitions. (b) Graphical representation of the students' partition into classes. We draw only hyperedges of size D. (c) We compare the inferred partitions with the 'attended class' covariate of the nodes, i.e. the classes students participate in. We comment further on this comparison in appendix G.2. (d) A quantitative measurement complementing that of panel (b): the Normalized Mutual Information (NMI) between inferred communities and attended classes, the AUC on the full dataset, as well as the ratio ρ_D of hyperedges of size equal to D. (e) Free energy landscape. We consider the parameters (p_2, n_2), (p_3, n_3) and (p_4, n_4) inferred from the dataset with, respectively, $D=2,3,4$. With these, we build the simplex of convex combinations $p=\sum_{i\in\{2,3,4\}}\lambda_i p_i$, where $\sum_{i\in\{2,3,4\}}\lambda_i=1$ and $0\leqslant\lambda_i\leqslant 1$ (similarly for n). For every point in the simplex, we compute the free energy on the full dataset, i.e. with $D=5$. More details on these computations are provided in appendix G.1.

computations feasible by exploiting classical cavity arguments, as well as a dynamic program similar to that employed for MP. We present the results of these estimates on the High School dataset in figure 5(e). Here, we take the values of n and p inferred by cutting the dataset at maximum hyperedge sizes $D=2,3,4$. Then, we compute the free energy on the full dataset ($D=5$) in the simplex of n,p parameters outlined by the three vertices. We notice that interactions of size $D=5$ seem to be less informative and lead to suboptimal inference; see appendix G.3. Similarly to what was observed on graphs [6], the energy landscape appears rugged and complex. EM converges to solutions that are local attraction points, i.e. valleys of low-energy configurations. Moreover, the free energy of the p,n parameters inferred with only pairwise interactions (i.e. $D=2$, lower-right) is higher than that inferred for $D=3$ (upper-left), which is in turn higher than the one of $D=4$ (bottom-left).

J. Stat. Mech. (2024) 043403

6. Conclusion

We developed a probabilistic generative model and an MP-based inference procedure that led to several results advancing community detection on hypergraphs. In particular, we obtained closed-form bounds for the detectability of community configurations, extending the seminal results of Decelle *et al* [6] to higher-order interactions. Experimental validation of such bounds shows the emergence of a detectability phase transition when spanning from disassortative to assortative community structures. With these theoretical bounds at hand, we investigate the relationship between hypergraphs and graphs from an information-theoretical perspective. Characterizing the entropy and perplexity of pairs of nodes in hyperedges, we find that hypergraphs with many overlapping hyperedges are easier to detect. Besides these theoretical advancements, we develop two relevant algorithmic ones. First, we derive an efficient and scalable MP algorithm to learn communities and model parameters. Second, we propose an exact and efficient sampling routine that generates synthetic data with the desired community structure according to our probabilistic model in the order of seconds. Both of these implementations have been released open source [42].

The mathematical tools we propose here to obtain our results are valid for standard hypergraphs. We foresee that they could be generalized to dynamic hypergraphs where interactions change in time, using intuitions derived for dynamic graphs [10]. Similarly, it would be interesting to see how detectability bounds change when accounting for node attributes, as results in networks have shown that adding extra information can boost community detection [59–61]. Finally, from an empirical perspective, it would be interesting to see how our theoretical insights in terms of entropy of hypergraphs and clique expansion match measures that relate hypergraphs to simplicial complexes [62].

Acknowledgments

N R and A L contributed equally to this work. N R acknowledges support from the Max Planck ETH Center for Learning Systems. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting A L.

Appendix A. Expected degree and choice of κ_d

As we commented in section 2, the choice of the normalizing constant κ_d, for $d = 2,\dots,D$, controls the Bernoulli probabilities for all hyperedges $e \in \Omega$ via

$$\mathbb{P}(e\,|\,p,t) = \frac{\pi_e}{\kappa_{|e|}} = \frac{\sum_{i<j\in e} p_{t_i t_j}}{\kappa_{|e|}}.$$

Our theoretical analysis and results hold for general choices of κ_d, as long as these respect the following conditions. First, for any choice of a symmetric $0 \leqslant p_{ab} \leqslant 1$, we need valid probabilities $0 \leqslant \pi_e/\kappa_{|e|} \leqslant 1$. This implies that, necessarily,

$$\kappa_d \geqslant \frac{d(d-1)}{2} \quad \forall d = 2, \ldots, D\,. \tag{A.1}$$

Second, we want the ensemble to consist of sparse hypergraphs, in expectation. A good proxy for such a requirement is the average degree, which we can compute explicitly as

$$\begin{aligned} d_0 &= \frac{1}{N}\sum_{i\in V}\sum_{e\in\Omega: i\in e}\mathbb{P}(e\,|\,p,t) \\ &= \frac{1}{N}\sum_{e\in\Omega}\sum_{i\in e}\mathbb{P}(e\,|\,p,t) \\ &= \frac{1}{N}\sum_{e\in\Omega}|e|\mathbb{P}(e\,|\,p,t) \\ &= \frac{1}{N}\sum_{e\in\Omega}\frac{|e|}{\kappa_{|e|}}\sum_{i<j\in e}p_{t_i t_j} \\ &= \frac{C}{N}\sum_{i<j\in V}p_{t_i t_j} \\ &\approx \frac{C}{N}\sum_{a\leqslant b\in[K]}\frac{p_{ab}\,(Nn_a)\,(Nn_b)}{1+\delta_{ab}} \\ &= \frac{C}{2}\sum_{a,b\in[K]}c_{ab}n_a n_b\,, \end{aligned} \tag{A.2}$$

where

$$C = \sum_{d=2}^{D}\binom{N-2}{d-2}\frac{d}{\kappa_d}\,.$$

We assume $c_{ab} = O(1)$, i.e. it is in a sparse regime. Thus, the expected degree's scale is governed by C and, in turn, by the choice of κ_d, as

$$d_0 = O\,(C)\,.$$

Additionally, but not necessarily, we wish our model to extend the classical SBM, which imposes the additional condition $\kappa_2 = 1$. There exist many choices of κ_d obeying the constraints just discussed. A natural one is the minimum value satisfying equation (A.1), i.e. $\kappa_d = d(d-1)/2$. This gives

$$C = \frac{2}{N-1}\sum_{d=1}^{D-1}\binom{N-1}{d}$$

which, for $D = N$, returns $d_0 = O\left(2^N/(N-1)\right)$, which is too high to yield sparse hypergraphs. Note that, in practice, we rarely use $D = N$. However, such considerations are useful to evaluate how different κ_d values reflect on the properties of the hypergraph ensembles of the model.

A more interesting choice is given by

$$\kappa_d = \frac{d(d-1)}{2}\binom{N-2}{d-2}.$$

This corresponds to taking the average among the $d(d-1)/2$ interactions that yield π_e, and $\binom{N-2}{d-2}$ is a normalization: once an interaction is observed between two nodes i, j, the remaining $d-2$ are chosen at random. This gives

$$\begin{aligned} C &= 2\sum_{d=1}^{D-1}\frac{1}{d} \\ &= 2H_{D-1}\,, \end{aligned} \tag{A.3}$$

which is proportional to the $(D-1)$th harmonic number, hence growing more mildly at leading order as $C = O(\log D)$. Aside from having an interpretation in terms of null modeling, the value in equation (A.3), which we utilize experimentally, was shown to be a sensible choice in many real-life scenarios [17, 41].

Appendix B. MP derivations

MP equations have been developed in the case of general factor graphs, see for example Murphy *et al* [35], section 22.2.3.2. We consider approximate messages from hyperedges e to nodes i being $\widehat{q}_{e\to i}(t_i)$, and vice versa, $q_{i\to e}(t_i)$. The messages, for any $e \in \mathcal{F}, i \in \partial e$, satisfy the general updates

$$\begin{aligned} q_{i\to e}(t_i) &\propto n_{t_i}\prod_{f\in\partial i\setminus e}\widehat{q}_{f\to i}(t_i) \\ \widehat{q}_{e\to i}(t_i) &\propto \sum_{t_j : j\in\partial e\setminus i}\left(\frac{\pi_e}{\kappa_e}\right)^{A_e}\left(1-\frac{\pi_e}{\kappa_e}\right)^{1-A_e}\prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)\,. \end{aligned} \tag{B.1}$$

The marginal beliefs are given by

$$q_i(t_i) \propto n_{t_i}\prod_{e\in\partial i}\widehat{q}_{e\to i}(t_i)\,. \tag{B.2}$$

B.1. Message updates

First, we can distinguish the values of messages for function nodes e such that $A_e = 0$ or $A_e = 1$, i.e. if the hyperedge e is observed or not in the data.

J. Stat. Mech. (2024) 043403

If $A_e = 1$, i.e. $e \in E$, then

$$\begin{aligned}\widehat{q}_{e\to i}(t_i) &\propto \sum_{t_j : j\in\partial e\setminus i} \frac{\pi_e}{\kappa_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \\ &\propto \sum_{t_j : j\in\partial e\setminus i} \pi_e \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)\,. \end{aligned} \tag{B.3}$$

If $A_e = 0$, then $e \in \Omega \setminus E$. We start by computing

$$\begin{aligned}\widehat{q}_{e\to i}(t_i) &\propto \sum_{t_j : j\in\partial e\setminus i} \left(1 - \frac{\pi_e}{\kappa_e}\right) \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \\ &= \sum_{t_j : j\in\partial e\setminus i} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) - \sum_{t_j : j\in\partial e\setminus i} \frac{\pi_e}{\kappa_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \\ &= 1 - \sum_{t_j : j\in\partial e\setminus i} \frac{\pi_e}{\kappa_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \\ &= 1 - \frac{1}{N} \sum_{t_j : j\in\partial e\setminus i} \frac{\sum_{k<m\in e} c_{t_k t_m}}{\kappa_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)\,. \end{aligned} \tag{B.4}$$

We indicate with $\hat{Z}_{e\to i}(t_i)$ the convenient non-normalized rewriting of $\widehat{q}_{e\to i}(t_i)$ in equation (B.4). Therefore, we find

$$\begin{aligned} q_{i\to e}(t_i) &\propto n_{t_i} \prod_{f\in\partial i\setminus e} \widehat{q}_{f\to i}(t_i) \\ &= \frac{n_{t_i}}{\widehat{q}_{e\to i}(t_i)} \prod_{f\in\partial i} \widehat{q}_{f\to i}(t_i) && \text{(B.5)} \\ &\propto \frac{n_{t_i}}{\hat{Z}_{e\to i}(t_i)} \prod_{f\in\partial i} \widehat{q}_{f\to i}(t_i) && \text{(B.6)} \\ &= \frac{q_i(t_i)}{\hat{Z}_{e\to i}(t_i)}\,, && \text{(B.7)} \end{aligned}$$

where from equations (B.5) to (B.6) we used $\hat{Z}_{e\to i}(t_i)$ introduced in equation (B.4). We evaluate the expression in equation (B.7) for the limit $N \to +\infty$, which gives the node-to-hyperedge messages for $e \in \Omega \setminus E$ as

$$\begin{aligned} q_{i\to e}(t_i) &= q_i(t_i) + O\left(\frac{1}{N}\right) \\ &\approx q_i(t_i)\,, \end{aligned} \tag{B.8}$$

i.e. the nodes approximately (to leading order in $O(1/N)$) share their marginal belief to hyperedges that are not observed in the data. Using equation (B.8), we can also approximate equation (B.4) as

J. Stat. Mech. (2024) 043403

$$\widehat{q}_{e\to i}(t_i) \propto 1 - \frac{1}{N}\sum_{t_j : j\in\partial e\setminus i} \frac{\sum_{k<m\in e} c_{t_k t_m}}{\kappa_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)$$
$$\approx 1 - \frac{1}{N}\sum_{t_j : j\in\partial e\setminus i} \frac{\sum_{k<m\in e} c_{t_k t_m}}{\kappa_e} \prod_{j\in\partial e\setminus i} q_j(t_j)\,. \tag{B.9}$$

In the assumed sparsity regime, the term of order $O(1/N)$ in equation (B.9) is close to zero. Since for $x \approx 0$ the approximation $1 - x \approx e^{-x}$ is sufficiently accurate, we write

$$\widehat{q}_{e\to i}(t_i) \approx \exp\left(-\frac{1}{N}\sum_{t_j : j\in\partial e\setminus i} \frac{\sum_{k<m\in e} c_{t_k t_m}}{\kappa_e} \prod_{j\in\partial e\setminus i} q_j(t_j)\right). \tag{B.10}$$

We can put the hyperedge-to-node updates together using the two results in equation (B.3) and in equation (B.10). Specifically, we derive the following expression for the message $q_{i\to e}(t_i)$, where $e \in E$:

$$q_{i\to e}(t_i) \propto n_{t_i} \prod_{\substack{f\in\Omega:\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i)$$
$$= n_{t_i} \prod_{\substack{f\in E:\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i) \prod_{\substack{f\in\Omega\setminus E\\ f\in\partial i}} \widehat{q}_{f\to i}(t_i)$$
$$\approx n_{t_i} \left(\prod_{\substack{f\in E:\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i)\right) \left[\prod_{\substack{f\in\Omega\setminus E:\\ f\in\partial i}} \exp\left(-\frac{1}{N}\sum_{t_j : j\in\partial f\setminus i} \frac{\sum_{k<m\in f} c_{t_k t_m}}{\kappa_f} \prod_{j\in\partial f\setminus i} q_j(t_j)\right)\right] \tag{B.11}$$
$$= n_{t_i} \left(\prod_{\substack{f\in E:\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i)\right) \exp\left(-\frac{1}{N}\sum_{\substack{f\in\Omega\setminus E:\\ f\in\partial i}} \sum_{t_j : j\in\partial f\setminus i} \frac{\sum_{k<m\in f} c_{t_k t_m}}{\kappa_f} \prod_{j\in\partial f\setminus i} q_j(t_j)\right)$$
$$\approx n_{t_i} \left(\prod_{\substack{f\in E:\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i)\right) \exp\left(-\frac{1}{N}\sum_{\substack{f\in\Omega:\\ f\in\partial i}} \sum_{t_j : j\in\partial f\setminus i} \frac{\sum_{k<m\in f} c_{t_k t_m}}{\kappa_f} \prod_{j\in\partial f\setminus i} q_j(t_j)\right) \tag{B.12}$$
$$= n_{t_i} \left(\prod_{\substack{f\in E:\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i)\right) \exp\left(-h_i(t_i)\right). \tag{B.13}$$

In equation (B.11), we used the approximation introduced in equation (B.10). In equation (B.12) we passed from summing over $\Omega \setminus E$ to Ω. This approximation is sensible as long as the expected degree of the nodes grows at most as N, which is satisfied

in the assumed sparse regime, as discussed in appendix A. Finally, in equation (B.13) we introduced a node-dependent external field $h_i(t_i)$, whose definition naturally follows from the argument of the exponential in equation (B.12).

B.2. External field updates

We simplify the external field to remove the node dependency of $h_i(a)$. The node-dependent external field reads

$$\begin{aligned} h_i(t_i) &= \frac{1}{N}\sum_{f\in\partial i}\frac{1}{\kappa_f}\left(\sum_{t_j:j\in\partial f\setminus i}\sum_{k<m\in f} c_{t_k t_m}\prod_{r\in f\setminus i} q_r(t_r)\right) \\ &= \frac{1}{N}\sum_{f\in\partial i}\frac{1}{\kappa_f}\left(\sum_{t_j:j\in f\setminus i}\sum_{m\in f\setminus i} c_{t_i t_m}\prod_{r\in f\setminus i} q_r(t_r)\right) + \text{const.} \end{aligned} \tag{B.14}$$

The sum in parentheses in equation (B.14) can be simplified as

$$\begin{aligned} \sum_{t_j:j\in f\setminus i}\left[\left(\sum_{m\in f\setminus i} c_{t_i t_m}\right)\prod_{r\in f\setminus i} q_r(t_r)\right] &= \sum_{t_j:j\in f\setminus i}\sum_{m\in f\setminus i}\left(c_{t_i t_m}\prod_{r\in f\setminus i} q_r(t_r)\right) \\ &= \sum_{m\in f\setminus i}\sum_{t_j:j\in f\setminus i}\left(c_{t_i t_m}\prod_{r\in f\setminus i} q_r(t_r)\right) \\ &= \sum_{m\in f\setminus i}\sum_{t_m} c_{t_i t_m} q_m(t_m)\,. \end{aligned} \tag{B.15}$$

Plugging equation (B.15) into equation (B.14) we get, ignoring constants,

$$\begin{aligned} h_i(t_i) &= \frac{1}{N}\sum_{f\in\partial i}\frac{1}{\kappa_f}\sum_{m\in f\setminus i}\sum_{t_m} c_{t_i t_m} q_m(t_m) \\ &= \frac{C'}{N}\sum_{j\in V\setminus i}\sum_{t_j} c_{t_i t_j} q_j(t_j) \\ &\approx \frac{C'}{N}\sum_{j\in V}\sum_{t_j} c_{t_i t_j} q_j(t_j)\,, \end{aligned} \tag{B.16}$$

with $C' = \sum_{d=2}^{D}\binom{N-2}{d-2}\frac{1}{\kappa_d}$, and where in equation (B.16) we included i in the node summation. Since equation (B.16) does not depend on i, we define the node-independent external field

$$h(a) = \frac{C'}{N}\sum_{j\in V}\sum_{t_j} c_{at_j} q_j(t_j) \quad \forall a\in[K]\,. \tag{B.17}$$

J. Stat. Mech. (2024) 043403

B.3. Marginal belief updates

Notice, that, in passing from equations (B.11) to (B.13) and then in equation (B.17), we have shown that

$$\prod_{\substack{f\in\Omega\setminus E\\ f\in\partial i}} \widehat{q}_{f\to i}\left(t_i\right) \approx \exp\left(-h_i\left(t_i\right)\right) \approx \exp\left(-h\left(t_i\right)\right). \tag{B.18}$$

We use the same argument to treat the general expression of the marginal beliefs in equation (B.2), yielding

$$\begin{aligned} q_i\left(t_i\right) &\propto n_{t_i} \prod_{e\in\partial i} \widehat{q}_{e\to i}\left(t_i\right) \\ &= n_{t_i} \prod_{\substack{e\in E:\\ e\in\partial i}} \widehat{q}_{e\to i}\left(t_i\right) \prod_{\substack{e\in\Omega\setminus E:\\ e\in\partial i}} \widehat{q}_{e\to i}\left(t_i\right) \\ &\approx n_{t_i} \prod_{\substack{e\in E:\\ e\in\partial i}} \widehat{q}_{e\to i}\left(t_i\right) \exp\left(-h\left(t_i\right)\right). \end{aligned}$$

B.4. Summary: approximate MP updates

Putting all derivations together, the final MP equations read

$$\text{Node-to-observed hyperedge:} \quad q_{i\to e}\left(t_i\right) \propto n_{t_i} \left(\prod_{\substack{f\in E\\ f\in\partial i\setminus e}} \widehat{q}_{f\to i}(t_i) \right) \exp\left(-h\left(t_i\right)\right) \qquad \forall e\in E, i\in e$$

$$\text{Observed hyperedge-to-node:} \quad \widehat{q}_{e\to i}\left(t_i\right) \propto \sum_{t_j : j\in\partial e\setminus i} \pi_e \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \qquad \forall e\in E, i\in e \tag{B.19}$$

$$\text{External field:} \quad h\left(t_i\right) = \frac{C'}{N} \sum_{j\in V} \sum_{t_j} c_{t_i t_j} q_j\left(t_j\right) \tag{B.20}$$

$$\text{Marginals:} \quad q_i\left(t_i\right) \propto n_{t_i} \left(\prod_{\substack{f\in E\\ f\in\partial i}} \widehat{q}_{f\to i}(t_i) \right) \exp\left(-h\left(t_i\right)\right).$$

Notice that the MP updates cannot be naively implemented as presented. In fact, the update in equation (B.19) for $\widehat{q}_{e\to i}(t_i)$ has cost $O(K^{|e|-1})$, which does not scale with the hyperedge size. In appendix D we present a dynamic programming approach to perform this computation exactly with cost $O(K^2|e|)$, and comment on further algorithmic details to implement the MP updates in practice.

Appendix C. EM inference

Updates of the community priors n. We take the derivative of the log-likelihood in equation (4). By imposing the constraint $\sum_{a=1}^{K} n_a = 1$, we obtain the update in equation (13).

Updates of the affinity matrix p. We show here the updates in terms of c. These easily translate to those in terms of the affinity matrix p as the expression we derive below in equation (C.6) is invariant with respect to the substitution $c = Np$. Let $x_e = \sum_{i<j\in e} c_{t_i t_j}/N\kappa_e$. Then, ignoring additive constants, the log-likelihood reads

$$\begin{aligned}\mathcal{L} &= \sum_{e\in E}\log\left(\sum_{i<j\in e} c_{t_i t_j}\right) + \sum_{e\in\Omega\setminus E}\log(1-x_e)\\ &\approx \sum_{e\in E}\log\left(\sum_{i<j\in e} c_{t_i t_j}\right) - \sum_{e\in\Omega\setminus E} x_e\\ &= \sum_{e\in E}\log\left(\sum_{i<j\in e} c_{t_i t_j}\right) - \sum_{e\in\Omega\setminus E}\frac{\sum_{i<j\in e} c_{t_i t_j}}{N\kappa_e} \end{aligned} \tag{C.1}$$

where equation (C.1) is the linearization of $\log(1-x)\approx x$ around $x=0$, which is valid at leading order $O(1/N)$. We now take a variational approach to find a lower bound $\tilde{\mathcal{L}}$ of the log-likelihood:

$$\begin{aligned}\mathcal{L} &\approx \sum_{e\in E}\log\left(\sum_{i<j\in e} c_{t_i t_j}\right) - \sum_{e\in\Omega\setminus E}\frac{\sum_{i<j\in e} c_{t_i t_j}}{N\kappa_e}\\ &\geqslant \sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\log\left(\frac{c_{t_i t_j}}{\rho_{ij}^{e}}\right) - \sum_{e\in\Omega\setminus E}\frac{\sum_{i<j\in e} c_{t_i t_j}}{N\kappa_e}\\ &= \sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\log c_{t_i t_j} - \sum_{e\in\Omega\setminus E}\frac{\sum_{i<j\in e} c_{t_i t_j}}{N\kappa_e} + \text{const.}\\ &= \tilde{\mathcal{L}}(c) + \text{const.}, \end{aligned} \tag{C.2}$$

which is valid for any distribution ρ_{ij}^{e} such that $\sum_{i<j\in e}\rho_{ij}^{e} = 1$. In equation (C.2), we utilized Jensen's inequality. The lower bound is exact when

$$\rho_{ij}^{e} = \frac{c_{t_i t_j}}{\sum_{i<j\in e} c_{t_i t_j}} = \frac{c_{t_i t_j}}{N\pi_e} \quad . \tag{C.3}$$

J. Stat. Mech. (2024) 043403

We compute the derivative of the variational lower bound and approximate to leading terms in N:

$$\begin{aligned}\frac{\partial\tilde{\mathcal{L}}}{\partial c_{ab}} &= \frac{1}{c_{ab}}\sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\delta_{t_i a}\delta_{t_j b} - \frac{1}{N}\sum_{e\in\Omega\setminus E}\frac{1}{\kappa_e}\sum_{i<j\in e}\delta_{t_i a}\delta_{t_j b}\\ &\approx \frac{1}{c_{ab}}\sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\delta_{t_i a}\delta_{t_j b} - \frac{1}{N}\sum_{e\in\Omega}\frac{1}{\kappa_e}\sum_{i<j\in e}\delta_{t_i a}\delta_{t_j b} \qquad \text{(C.4)}\\ &= \frac{1}{c_{ab}}\sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\delta_{t_i a}\delta_{t_j b} - \frac{C'}{N}\sum_{i<j\in V}\delta_{t_i a}\delta_{t_j b}\\ &= \frac{1}{c_{ab}}\sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\delta_{t_i a}\delta_{t_j b} - \frac{C'}{2N}\left(N_a N_b - \delta_{ab}N_a\right)\\ &= \frac{1}{c_{ab}}\sum_{e\in E}\sum_{i<j\in e}\rho_{ij}^{e}\delta_{t_i a}\delta_{t_j b} - \frac{C'}{2}\left(N n_a n_b - \delta_{ab}n_a\right). \qquad \text{(C.5)}\end{aligned}$$

where $C' = \sum_{d=2}^{D}\binom{N-2}{d-2}\frac{1}{\kappa_d}$. Notice that the approximations in equations (C.4) and (C.5) hold valid only when considering c_{ab} in the expressions, as by assumption $c = O(1)$. Now, by setting equation (C.5) equal to zero, and substituting ρ_{ij}^{e} from equation (C.3), we obtain the update

$$c_{ab}^{(t+1)} = c_{ab}^{(t)}\frac{2\sum_{e\in E}\#_{ab}^{e}/\pi_e}{N C'\left(N n_a n_b - \delta_{ab}n_a\right)}, \qquad \text{(C.6)}$$

where $\#_{ab}^{e} = \sum_{i<j\in e}\delta_{t_i a}\delta_{t_j b}$.

Appendix D. Algorithmic and computational details

D.1. Dynamic programming for MP

In this section, we explain how the MP updates for the $\widehat{q}_{e\to i}(t_i)$ messages can be performed efficiently. In log-space, the messages can be compactly written as

$$\begin{aligned}\log\widehat{q}_{e\to i}(t_i) &= \log\sum_{t_j:j\in\partial e\setminus i}\pi_e\prod_{j\in e\setminus i}q_{j\to e}(t_j) + \text{const.}\\ &= \psi(e,i,t_i) + \text{const.} \quad . \qquad \text{(D.1)}\end{aligned}$$

Below, we focus on finding efficient updates for ψ as defined in equation (D.1), which should be exponentiated and properly normalized to find the original messages $\widehat{q}_{e\to i}(t_i)$. For this, we introduce an auxiliary quantity. For any subset $g\subseteq f$ of nodes in f, where $i\in g$, we define

$$\eta(g,i,t_i) = \log\left[\sum_{t_j:j\in g\setminus i}\left(\sum_{l<m\in g}p_{t_l t_m}\right)\prod_{j\in g\setminus i}q_{j\to f}(t_j)\right].$$

J. Stat. Mech. (2024) 043403

Hence, $\eta(f,i,t_i) = \psi(f,i,t_i) + \text{const.}$. This quantity is useful in that it allows us to obtain an efficient recursion formula for ψ by computing the η values starting from subsets g containing two nodes.

Without loss of generality, consider $f = \{1,\dots,m-1\}$ and $i = 1$. Consider $g = \{1,\dots,n-1\}$ for some $n \leqslant m$. We want to compute η for the set $\{1,\dots n\}$. Its exponential is given by

$$
\begin{aligned}
&\exp(\eta(\{1,\dots,n\},1,t_1)) \\
&\quad= \sum_{t_n}\sum_{t_2}\sum_{t_3}\dots\sum_{t_{n-1}}(p_{t_1t_2}+\dots+p_{t_{n-2}t_{n-1}}+p_{t_1t_n}+p_{t_2t_n}+\dots+p_{t_{n-1}t_n}) \\
&\qquad\times(q_{2\to f}(t_2)\dots q_{n-1\to f}(t_{n-1})q_{n\to f}(t_n)) \\
&\quad= \sum_{t_n} q_{n\to f}(t_n)\Bigg(\sum_{t_2}\sum_{t_3}\dots\sum_{t_{n-1}}(p_{t_1t_2}+\dots+p_{t_{n-2}t_{n-1}}) \\
&\qquad\times(q_{2\to f}(t_2)\dots q_{n-1\to f}(t_{n-1})) \\
&\qquad+\sum_{t_2}\sum_{t_3}\dots\sum_{t_{n-1}}p_{t_1t_n}(q_{2\to f}(t_2)\dots q_{n-1\to f}(t_{n-1})) \\
&\qquad+\sum_{t_2}\sum_{t_3}\dots\sum_{t_{n-1}}(p_{t_2t_n}+\dots+p_{t_{n-1}t_n})(q_{2\to f}(t_2)\dots q_{n-1\to f}(t_{n-1}))\Bigg) \\
&\quad= \sum_{t_n} q_{n\to f}(t_n)\Bigg(\exp(\eta(\{1,\dots,n-1\},1,t_1))+p_{t_1t_n} \\
&\qquad+\sum_{t_2}p_{t_2t_n}q_{2\to f}(t_2)+\dots+\sum_{t_{n-1}}p_{t_{n-1}t_n}q_{n-1\to f}(t_{n-1})\Bigg) \\
&\quad= \exp(\eta(\{1,\dots,n-1\},1,t_1)) \\
&\qquad+\sum_{t_n} q_{n\to f}(t_n)\Bigg(p_{t_1t_n}+\sum_{t_2}p_{t_2t_n}q_{2\to f}(t_2)+\dots+\sum_{t_{n-1}}p_{t_{n-1}t_n}q_{n-1\to f}(t_{n-1})\Bigg).
\end{aligned}
\tag{D.2}
$$

The recursion in equation (D.2) allows us to compute the value of $\eta(\{1,\dots,n\},1,t_1)$ from $\eta(\{1,\dots,n-1\},1,t_1)$ in time $O((n-2)K^2)$. However, we can further reduce the cost. For any $a \in [K]$, define

$$
s_n(a) = \sum_{t_2} p_{t_2a}q_{2\to f}(t_2)+\dots+\sum_{t_{n-1}}p_{t_{n-1}a}q_{n-1\to f}(t_{n-1}).
$$

Substituting the definition of $s_n(a)$ in equation (D.2), we obtain the final two-step dynamic update:

$$s_n(a) = s_{n-1}(a) + \sum_{t_{n-1}} p_{t_{n-1}a} q_{n-1\to f}(t_{n-1}) \tag{D.3}$$

$$\begin{aligned}\exp(\eta(\{1,\dots,n\},1,t_1)) = \exp(\eta(\{1,\dots,n-1\},1,t_1)) \\ + \sum_{t_n} q_{n\to f}(t_n)\left(p_{t_1 t_n} + s_n(t_n)\right).\end{aligned} \tag{D.4}$$

This yields a cost of $O(K)$ per recursion, and a total cost of $O(K|f|)$ to compute the final $\psi(f,1,t_1)$. In practice, for any e,i pair, we compute $\psi(e,i,t_i)$ for all values $t_i \in [K]$, which yields a total cost of $O(K^2|f|)$.

D.2. Implementation details

In our implementation of the MP and EM routines, we take some additional steps to ensure convergence to non-trivial local optima of the free energy landscape.

The initialization of the messages is performed taking into account the circular relationships in equations (9)–(12). We perform them as follows: (i) randomly initialize the messages $q_{i\to e}(t_i)$. For every i,e pair, the messages are drawn from a K-dimensional Dirichlet distribution. (ii) Similarly, randomly initialize the marginal beliefs $q_i(t_i)$. (iii) We infer all the other quantities from the initialized $q_{i\to e}(t_i)$ and $q_i(t_i)$; in fact, up to constants

$$\widehat{q}_{e\to i}(t_i) = \frac{q_i(t_i)}{q_{i\to e}(t_i)}.$$

All values are then normalized to have unitary sum. (iv) Finally, the external field is entirely determined by the marginals as per equation (12).

We check for convergence of the MP and EM inference routines by evaluating the absolute difference between parameters in consecutive steps. We present complete pseudocodes of the two routines in algorithms 2 and 3.

J. Stat. Mech. (2024) 043403

Algorithm 2. Inferring communities (MP).

Inputs: convergence threshold ϵ_{mp}
maximum iterations iter_{mp}
prior n, rescaled affinity matrix c

randomly initialize all $q_{i\to e}(t_i), \widehat{q}_{e\to i}(t_i), q_i(t_i), h(t_i)$

for step $= 1, \ldots, \text{iter}_{\text{mp}}$ **do**
  // Perform updates
  for all $e \in E, i \in e, t_i \in [K]$ **do**
    update messages $q_{i\to e}(t_i)$ ▷ equation (9)
  end for
  for all $e \in E, i \in e, t_i \in [K]$ **do**
    update messages $\widehat{q}_{e\to i}(t_i)$ ▷ equation (10)
  end for
  for all $e \in E, i \in e, t_i \in [K]$ **do**
    $q_i^{old}(t_i) \leftarrow q_i(t_i)$
    update marginals $q_i(t_i)$ ▷ equation (11)
  end for
  for $t_i \in [K]$ **do**
    update external field $h(t_i)$ ▷ equation (12)
  end for

  // Check for convergence
  $\Delta = \sum_{i=1}^{N} \sum_{t_i=1}^{K} |q_i^{\text{old}}(t_i) - q_i(t_i)|$
  if $\Delta < \epsilon_{\text{mp}}$ **then**
    break
  end if
end for

While algorithm 2 is presented as a completely parallel implementation of the MP equations (9)–(12), in practice we proceed in batches. In fact, we find that applying completely parallel updates, i.e. applying equation (9) for all i, e pairs, successively equation (10) for all i, e pairs, and then equation (11) for all nodes $i \in V$, results in fast convergence to degenerate fixed points where all nodes are assigned to the same community. For this reason, we apply dropout. Given a fraction $\alpha \in (0, 1]$, we select a random fraction α of all possible i, e pairs, and apply the update in equation (9) only for the selected pairs. We perform a new random draw, and update according to equation (10), and similarly for equation (11). Finally, we update the external field in equation (12). Empirically, we find that a value of $\alpha = 0.25$ works for synthetic data, where inference is simpler. Values below work as well. For real data we find that substantially lowering α yields more stable inference. On real data, where we alternate MP and EM, and learning is less stable, we utilize $\alpha = 0.01$. In practice, we also set a patience parameter, and only stop MP once a given number of iterations in a row falls

J. Stat. Mech. (2024) 043403

Algorithm 3. Inferring model parameters (EM).

Inputs: convergence threshold ϵ_{em}
maximum iterations iter_{em}

randomly initialize c, n

for step $= 1, \ldots, \text{iter}_{\text{em}}$ **do**
  // Perform updates
  perform Message-passing inference ▷ algorithm 2
  $n^{\text{old}} \leftarrow n$
  update n ▷ equation (13)
  $c^{\text{old}} \leftarrow c$
  update c ▷ equation (14)

  // Check for convergence
  $\Delta = \sum_{a=1}^{K} |n_a - n_a^{\text{old}}| + \sum_{a,b=1}^{K} |c_{ab} - c_{ab}^{\text{old}}|$
  if $\Delta < \epsilon_{\text{em}}$ **then**
    break
  end if
end for

below the threshold ϵ_{mp} in algorithm 2. For real datasets, we set the patience to 50 consecutive steps, and the maximum number of iterations $\text{iter}_{\text{mp}} = 2000$.

Appendix E. Sampling from the generative model

E.1. Computational complexity

For a fixed hyperedge size d, there are two parts to the computational cost: iterating through the counts #, and sampling the hyperedges. The number of counts is fixed and given by $K^d/d!$, i.e. the number of possible ways to assign d nodes to K groups, without order. This cost corresponds to performing steps (ii) and (iii) of the sampling algorithm in section 3.4, where one needs to enumerate all possible counts #, which are $K^d/d!$ for every dimension d, and sample from a binomial distribution for each count. The cost of sampling the hyperedges in step (iv) in section 3.4 can also be precisely quantified. Every d-dimensional hyperedge is sampled with a computational cost of d since it is exactly the extraction of d nodes from V, and there are ω_d of such hyperedges. Calling Ω^d the space of all d-dimensional hyperedges, we find

J. Stat. Mech. (2024) 043403

$$\begin{aligned}
\mathbb{E}[\omega_d] &= \sum_{e\in\Omega^d} \mathbb{P}(e\,|\,p,t) \\
&= \sum_{e\in\Omega^d} \sum_{i<j\in e} \frac{p_{t_i t_j}}{\kappa_d} \\
&= \frac{1}{\kappa_d} \sum_{e\in\Omega^d} \sum_{i<j\in e} p_{t_i t_j} \\
&= \frac{\binom{N-2}{d-2}}{\kappa_d} \sum_{i<j\in V} p_{t_i t_j} \\
&\approx \frac{\binom{N-2}{d-2} N^2}{\kappa_d} \sum_{a\leqslant b=1}^{K} p_{ab} n_a n_b \\
&= \frac{\binom{N-2}{d-2} N}{\kappa_d} \sum_{a\leqslant b=1}^{K} c_{ab} n_a n_b \,.
\end{aligned}$$

Hence, the average computational cost is given by

$$\sum_{d=2}^{D} \left(\frac{K^d}{d!} + d\,\mathbb{E}[\omega_d] \right). \tag{E.1}$$

Given the large size of Ω^d, the cost in equation (E.1) tightly concentrates around the expected value. In sparse regimes, the term $K^d/d!$ dominates as the number of hyperedges ω_d is low, while the two terms both contribute to the cost when $\mathbb{E}[\omega_d]$ grows.

Precisely, we quantify the cost in equation (E.1) in terms of asymptotic complexity. The first summand $\sum_{d=2}^{D} \frac{K^d}{d!}$ absolutely converges to a constant for diverging D, and contributes to the complexity only as a constant relevant in sparse regimes. Defining $a_d = K^d/d!$, we can use the ratio test to assess convergence:

$$\lim_{d\to+\infty} \left| \frac{a_{d+1}}{a_d} \right| = \lim_{d\to+\infty} \frac{K^{d+1} d!}{K^d (d+1)\, d!} = \lim_{d\to+\infty} \frac{K}{d+1} = 0\,. \tag{E.2}$$

Substituting the value of $\kappa_d = \frac{d(d-1)}{2}\binom{N-2}{d-2}$ that we utilize in our experiments, it is also possible to quantify the second addend:

$$\begin{aligned}
\sum_{d=2}^{D} d\,\mathbb{E}[\omega_d] &\approx \sum_{d=2}^{D} d \left(\frac{\binom{N-2}{d-2} N}{\kappa_d} \sum_{a\leqslant b=1}^{K} c_{ab} n_a n_b \right) \\
&= \left(\sum_{a\leqslant b=1}^{K} c_{ab} n_a n_b \right) \left(2N \sum_{d=2}^{D} \frac{1}{d-1} \right).
\end{aligned}$$

J. Stat. Mech. (2024) 043403

Similar to the reasoning presented in equation (A.3), choosing the maximum possible cost, given by $D = N$ (which is higher than most practical use cases), the sum $\sum_{d=2}^{D} \frac{1}{d-1}$ grows like $O(\log N)$, therefore $\sum_{d=2}^{D} d\,\mathbb{E}[\omega_d] = O(N \log N)$, which yields an asymptotic bound of the total sampling complexity.

Finally, we remark that since sampling from equation (17) is computationally costly, we approximate the binomial with a Gaussian distribution[4], or with a Poisson if $N_\#$ is large and $\pi_\#/\kappa_d$ is small[5]. We use a Ramanujan approximation for large log-factorials appearing in the calculations[6].

E.2. Experiments

We employ the sampling algorithm to generate the hypergraphs used to study the phase transition of section 4.2. Here, we set the affinity matrix to have all equal in-degree $c_{aa} = c_{\text{in}}$ and out-degree $c_{ab} = c_{\text{out}}$, so that equation (18) becomes $c_{\text{in}} + (K-1)c_{\text{out}} = Kc$ for some K and c. In our experiments, we sample hypergraphs with $N = 10^4$ nodes by fixing $c = 10$ and $K = 4$, we span across 65 values of c_{out} in $[0, 500]$, and compute the corresponding $c_{\text{in}} = c_{\text{in}}(c_{\text{out}}; K, c)$. For each experimental configuration $c_{\text{in}}, c_{\text{out}}$, we draw five hypergraphs from different random seeds. This gives a total of 325 hypergraphs.

We use the expected number of d-dimensional hyperedges $\mathbb{E}[\omega_d]$ in equation (E.1) and the average degree d_0 in equation (A.2) to perform a sanity check between our sampling algorithm and theoretical derivations. For constant in- and out-degree, these two metrics evaluate to

$$\mathbb{E}[\omega_d] \approx \frac{Nc}{d(d-1)}\,,$$
$$d_0 \approx \frac{Cc}{2}\,.$$

The results in figure E1 show excellent agreement between theory and experiments. We also highlight that the sampling method is extremely fast and has an average sampling time of $t = 32.7 \pm 2.7$(s) in the experimental setup considered here.

Appendix F. Phase transition: complementary derivations and additional results

F.1. Proof of proposition 1

First, we want to prove that all communities have the same expected degree. In order to do that, we start by computing the expected degree d_{0i} of a given node $i \in V$. Following similar derivations to those for d_0 in appendix A, we find

[4] To deal with large $N_\#$ and κ_d that cannot be stored in memory, we approximate the binomial in equation (17) with a Gaussian $\mathcal{N}\left(\frac{\pi_\#}{\kappa_d} N_\#, \frac{\pi_\#}{\kappa_d} N_\# \left(1 - \frac{\pi_\#}{\kappa_d}\right)\right)$ Crucially, the Gaussian's mean and variance only involve the ratio $N_\#/\kappa_d$, which is numerically stable. We adopt this approximation when the Gaussian's variance exceeds 10.

[5] A Poisson approximation of the binomial $\text{Pois}\left(N_\# \frac{\pi_\#}{\kappa_d}\right)$ is used if $N_\# > 20$ and $N_\# \pi_\#/\kappa_d < 0.1$, or if $N_\# > 100$ and $N_\# \pi_\#/\kappa_d < 10$.

[6] For $n > 5$, we adopt the Ramanujan approximation [63] $\log n! \approx n \log n - n + \frac{\log\left(\frac{1}{30} + n(1 + 4n(1+2n))\right)}{6} + \frac{\log \pi}{2}$,giving error of order $O(1/n^3)$.

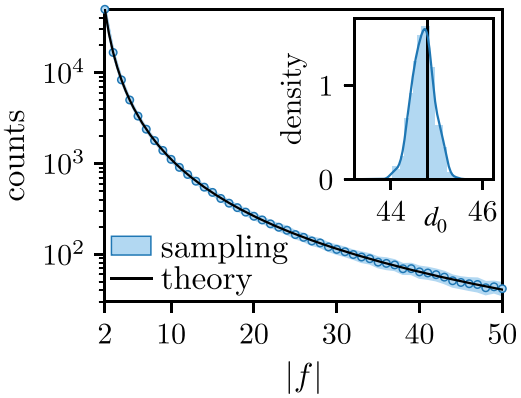

Figure E1. Sampling experiments. The expected number of $|f|$-dimensional hyperedges returned by our experiments (blue) is in great accordance with the theoretical prediction $\mathbb{E}[\omega_{|f|}]$ (black). Similarly, the experimental expected degrees are distributed around the analytical d_0. Shaded areas are standard deviations over five random hypergraph extractions at each $|f|$.

$$
\begin{aligned}
d_{0i} &= \sum_{e\in E: i\in e} \mathbb{P}(e\,|\,p,t) \\
&= \sum_{e\in E: i\in e} \frac{\pi_e}{\kappa_e} \\
&= C'\sum_{a=1}^{K} c_{t_i a} n_a + \frac{NC''}{2}\left(\sum_{b,d=1}^{K} c_{bd} n_b n_d + \sum_{b=1}^{K} c_{bb} n_b^2\right),
\end{aligned}
$$

where $C' = \sum_{d=2}^{D} \binom{N-2}{d-2}/\kappa_d$, as previously defined, and $C'' = \sum_{d=3}^{D} \binom{N-3}{d-3}/\kappa_d$. Therefore, the average degree $\langle b\rangle$ of a community $b \in [K]$ evaluates to

$$
\begin{aligned}
\langle b\rangle &= \frac{1}{N_b}\sum_{i\in V: t_i = b} d_{0i} \\
&= \frac{1}{N_b}\sum_{i\in V: t_i = b}\left[C'\sum_{a=1}^{K} c_{t_i a} n_a + \frac{NC''}{2}\left(\sum_{d,m=1}^{K} c_{dm} n_d n_m + \sum_{d=1}^{K} c_{dd} n_d^2\right)\right] \\
&= \frac{1}{N_b}\sum_{i\in V: t_i = b}\left[C'\sum_{a=1}^{K} c_{ba} n_a + \frac{NC''}{2}\left(\sum_{d,m=1}^{K} c_{dm} n_d n_m + \sum_{d=1}^{K} c_{dd} n_d^2\right)\right] \\
&= C'\sum_{a=1}^{K} c_{ba} n_a + \frac{NC''}{2}\left(\sum_{d,m=1}^{K} c_{dm} n_d n_m + \sum_{d=1}^{K} c_{dd} n_d^2\right) \\
&= C'c + \frac{NC''}{2}\left(\sum_{d,m=1}^{K} c_{dm} n_d n_m + \sum_{d=1}^{K} c_{dd} n_d^2\right),
\end{aligned}
$$

J. Stat. Mech. (2024) 043403

which is independent of the specific choice of group b, from which we conclude that all the groups yield equal expected degrees.

Second, we wish to demonstrate that MP's fixed points are as in equations (19) and (20). Notice that in the derivations here below, when convenient, we interchange equivalent summations over the function nodes' neighbors ∂e and hyperedge e. By treating all quantities that are independent of t_i in $q_{i\to e}(t_i), \widehat{q}_{e\to i}(t_i)$ as a constant, we evaluate equation (7) as

$$\begin{aligned}
\widehat{q}_{e\to i}(t_i) &\propto \sum_{t_j : j\in e\setminus i} \frac{\pi_e}{\kappa_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \\
&\propto \sum_{t_j : j\in e\setminus i} \sum_{r<s\in e} p_{t_r t_s} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \\
&= \sum_{t_j : j\in e\setminus i} \left(\sum_{r\in e\setminus i} p_{t_r t_i} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) + \sum_{r<s\in e\setminus i} p_{t_r t_s} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j) \right) \\
&= \sum_{r\in e\setminus i} \sum_{t_r} p_{t_r t_i} q_{r\to e}(t_r) + \sum_{r<s\in e\setminus i} \sum_{t_r,t_s} p_{t_r t_s} q_{r\to e}(t_r)\, q_{s\to e}(t_s) \\
&= \sum_{r\in e\setminus i} \sum_{t_r} p_{t_r t_i} n_{t_r} + \sum_{r<s\in e\setminus i} \sum_{t_r,t_s} p_{t_r t_s} n_{t_r} n_{t_s} \\
&= \frac{1}{N} \left(\sum_{r\in e\setminus i} c + \sum_{r<s\in e\setminus i} c \right) \\
&= \frac{c}{N} \left((|e|-1) + c\frac{|e|\,(|e|-1)}{2} \right).
\end{aligned} \tag{F.1}$$

Since messages $\widehat{q}_{e\to i}(t_i)$ are normalized to have unitary sum, equation (F.1) implies that $\widehat{q}_{e\to i}(t_i) = 1/K$. Substituting this result into equation (8), one also finds that $q_i(t_i) = n_{t_i}$. The variable-to-function node messages are updated with equation (9), which includes equation (12) for the external field $h(t_i)$. The external field evaluated at fixed points is also constant; in fact,

$$\begin{aligned}
h(t_i) &= \frac{C'}{N} \sum_{j\in V} \sum_{t_j} c_{t_i t_j} q_j(t_j) \\
&= \frac{C'}{N} \sum_{j\in V} \sum_{t_j} c_{t_i t_j} n_{t_j} \\
&= \frac{C'}{N} \sum_{j\in V} c \\
&= C'c.
\end{aligned} \tag{F.2}$$

The result of equation (F.2) implies that the messages in equation (9) read

$$
\begin{aligned}
q_{i\to e}(t_i) &\propto n_{t_i}\left(\prod_{\substack{f\in E\\ f\in\partial i\setminus e}}\sum_{t_j:j\in\partial f\setminus i}\pi_f\prod_{j\in\partial f\setminus i}q_{j\to f}(t_j)\right)\\
&= n_{t_i}\left(\prod_{\substack{f\in E\\ f\in\partial i\setminus e}}\sum_{t_j:j\in\partial f\setminus i}\pi_f\prod_{j\in\partial f\setminus i}n_{t_j}\right)\\
&\propto n_{t_i}\left(\prod_{\substack{f\in E\\ f\in\partial i\setminus e}}\sum_{r<s\in f}\sum_{t_j:j\in\partial f\setminus i}p_{t_rt_s}\prod_{j\in\partial f\setminus i}n_{t_j}\right)\\
&\propto n_{t_i}\left[\prod_{\substack{f\in E\\ f\in\partial i\setminus e}}\left(\sum_{r<s\in f}\sum_{t_rt_s}p_{t_rt_s}n_{t_r}n_{t_s}+\sum_{r\in f\setminus i}\sum_{t_r}p_{t_rt_i}n_{t_r}\right)\right]\\
&\propto n_{t_i}\left[\prod_{\substack{f\in E\\ f\in\partial i\setminus e}}\left(\frac{|f|(|f|-1)}{2}c+(|f|-1)c\right)\right]\\
&\propto n_{t_i},
\end{aligned}
$$

which is exactly equation (20).

F.2. Transition matrix formula

In this section, we derive the expression for the transition matrix $\tilde{T}_r^{ab}$ in equation (22). To simplify the notation, we indicate the (variable node, function node) pairs at level r as $(i_r, f_r) = (i, e)$, and similarly, at level $r+1$ we use $(i_{r+1}, f_{r+1}) = (j, f)$. Hence, the transition matrix becomes

$$
\tilde{T}_r^{ab} = \frac{\partial q_{i\to e}(a)}{\partial q_{j\to f}(b)}.
$$

In order to find a closed-form expression of $\tilde{T}_r^{ab}$, we claim that the two following lemmas hold.

Lemma 1. *Under the constant group degree assumption in equation (18):*

(i) for any hyperedge e and nodes $i \in e$:

$$
\sum_{t_j:j\in e\setminus i}\pi_e\prod_{k\in e\setminus i}q_{k\to e}(t_k) = \frac{c|e|(|e|-1)}{2N}; \tag{F.3}
$$

(ii) for any hyperedge e and nodes $i,j \in e$:

$$\sum_{t_k : k\in e\setminus i,j} \pi_e \prod_{m\in e\setminus i,f} q_{m\to e}(t_m) = \frac{1}{N}\left[c_{t_i t_j} + c\left(|e|-2\right)\left(2+\frac{|e|-3}{2}\right)\right]. \quad \text{(F.4)}$$

Lemma 2 (employing lemma 1). *Under the constant group degree assumption in equation (18):*

(i) the derivative $\partial \exp(-h(a))/\partial q_{i\to e}(b)$ is negligible to leading order in N;
(ii) the external field is constant $h(t_i) = \text{const.}$;
(iii) call $Z^{i\to e}$ the normalizing constant of $q_{i\to e}$, then

$$Z^{i\to e} = \prod_{g\in\partial i\setminus e} c\frac{|g|\left(|g|-1\right)}{2N} \quad \text{(F.5)}$$

$$\frac{\partial Z^{i\to e}}{\partial q_{j\to f}(b)} = \frac{c}{N}\left(\prod_{g\in\partial i\setminus e,f} c\frac{|g|\left(|g|-1\right)}{2N}\right)\left[1+\left(|f|-2\right)\left(2+\frac{|f|-3}{2}\right)\right]. \quad \text{(F.6)}$$

The claims allow us to derive the transition matrix. Particularly, we make explicit all derivatives and variable-to-function node messages as in equation (9). By also ignoring all terms relative to $h(t_i)$ thanks to lemma 2, we get

$$\begin{aligned}\frac{\partial q_{i\to e}(a)}{\partial q_{j\to f}(b)} \approx &-\frac{1}{\left(Z^{i\to e}\right)^2}\frac{\partial Z^{i\to e}}{\partial q_{j\to f}(b)} n_{t_i}\left(\prod_{g\in\partial i\setminus e}\sum_{t_m : m\in\partial g\setminus i}\pi_g\prod_{m\in\partial g\setminus i} q_{m\to g}(t_m)\right)\\ &+\frac{1}{Z^{i\to e}} n_{t_i}\left(\prod_{g\in\partial i\setminus e,f}\sum_{t_m : m\in\partial g\setminus i}\pi_g\prod_{m\in\partial g\setminus i} q_{m\to g}(t_m)\right)\\ &\times\left(\sum_{t_m : m\in\partial f\setminus i,j}\pi_f\prod_{m\in\partial f\setminus i,j} q_{m\to f}(t_m)\right).\end{aligned}$$

The terms involving $Z^{i\to e}$ are in lemma 2 (equations (F.5) and (F.6)), while the expressions in parentheses are in lemma 1 (equations (F.3) and (F.4)). By performing all the substitutions we get

J. Stat. Mech. (2024) 043403

$$
\begin{aligned}
\frac{\partial q_{i\to e}(a)}{\partial q_{j\to f}(b)} &= -n_{t_i}\left(\prod_{g\in\partial i\setminus e} c\frac{|g|(|g|-1)}{2N}\right)^{-2}\left(\prod_{g\in\partial i\setminus e,f} c\frac{|g|(|g|-1)}{2N}\right)\\
&\quad\times\left(\prod_{g\in\partial i\setminus e} c\frac{|g|(|g|-1)}{2N}\right)\frac{c}{N}\left[1+(|f|-2)\left(2+\frac{|f|-3}{2}\right)\right]\\
&\quad+n_{t_i}\left(\prod_{g\in\partial i\setminus e} c\frac{|g|(|g|-1)}{2N}\right)^{-1}\left(\prod_{g\in\partial i\setminus e,f} c\frac{|g|(|g|-1)}{2N}\right)\\
&\quad\times\frac{1}{N}\left[c_{t_i t_j}+c(|f|-2)\left(2+\frac{|f|-3}{2}\right)\right]\\
&= n_{t_i}\left(c\frac{|f|(|f|-1)}{2N}\right)^{-1}\left\{-\frac{c}{N}\left[1+(|f|-2)\left(2+\frac{|f|-3}{2}\right)\right]\right.\\
&\quad\left.+\frac{1}{N}\left[c_{t_i t_j}+c(|f|-2)\left(2+\frac{|f|-3}{2}\right)\right]\right\}\\
&= \frac{2}{|f|(|f|-1)}n_{t_i}\left(\frac{c_{t_i t_j}}{c}-1\right)\\
&= \frac{2}{|f|(|f|-1)}n_a\left(\frac{c_{ab}}{c}-1\right)
\end{aligned}
\tag{F.7}
$$

which is exactly the expression in equation (22).

What is left to complete all derivations is to prove lemmas 1 and 2, which is done next.

F.2.1. Proof of lemma 1

1. Derivation of equation (F.3):

$$
\begin{aligned}
\sum_{t_j:j\in e\setminus i}\pi_e\prod_{k\in e\setminus i}q_{k\to e}(t_k) &= \sum_{t_j:j\in e\setminus i}\sum_{r<s\in e}p_{t_r t_s}\prod_{k\in e\setminus i}q_{k\to e}(t_k)\\
&= \sum_{r<s\in e\setminus i}\sum_{t_r t_s}p_{t_r t_s}q_{r\to e}(t_r)\,q_{s\to e}(t_s)+\sum_{r\in e\setminus i}\sum_{t_r}p_{t_r t_i}q_{r\to e}(t_r)\\
&= \frac{1}{N}\sum_{r<s\in e\setminus i}\sum_{t_r,t_s}c_{t_r t_s}n_{t_r}n_{t_s}+\frac{1}{N}\sum_{r\in e\setminus i}\sum_{t_r}c_{t_r t_i}n_{t_r}\\
&= \frac{c}{N}\left[\frac{(|e|-1)(|e|-2)}{2}+(|e|-1)\right]\\
&= \frac{c(|e|-1)|e|}{2N}.
\end{aligned}
$$

2. Derivation of equation (F.4):

$$\begin{aligned}\sum_{t_k:k\in e\setminus i,j}\pi_e\prod_{m\in e\setminus i,j}q_{m\to e}(t_m) &= \sum_{t_k:k\in e\setminus i,j}\sum_{r<s\in e}p_{t_rt_s}\prod_{m\in e\setminus i,j}q_{m\to e}(t_m)\\ &= p_{t_it_j}+\sum_{r\in e\setminus i,j}\sum_{t_r}p_{t_rt_i}q_{r\to e}(t_r)+\sum_{r\in e\setminus i,j}\sum_{t_r}p_{t_rt_j}q_{r\to e}(t_r)\\ &\quad+\sum_{r<s\in e\setminus i,j}\sum_{t_r,t_s}p_{t_rt_s}q_{r\to e}(t_r)\,q_{s\to e}(t_s)\\ &= \frac{1}{N}\left(c_{t_it_j}+\sum_{r\in e\setminus i,j}\sum_{t_r}c_{t_rt_i}n_{t_r}+\sum_{r\in e\setminus i,j}\sum_{t_r}c_{t_rt_j}n_{t_r}\right.\\ &\quad\left.+\sum_{r<s\in e\setminus i,j}\sum_{t_r,t_s}c_{t_rt_s}n_{t_r}n_{t_s}\right)\\ &= \frac{1}{N}\left(c_{t_it_j}+\sum_{r\in e\setminus i,j}c+\sum_{r\in e\setminus i,j}c+\sum_{r<s\in e\setminus i,j}c\right)\\ &= \frac{1}{N}\left[c_{t_it_j}+c(|e|-2)\left(2+\frac{|e|-3}{2}\right)\right].\end{aligned}$$

F.2.2. Proof of lemma 2

1. Using equation (12), we write

$$\frac{\partial\exp(-h(a))}{\partial q_{i\to e}(b)}=\exp\left(-\frac{C'}{N}\sum_{v\in V}\sum_{t_k}c_{at_k}q_k(t_k)\right)\left(-\frac{C'}{N}\sum_{k\in V}\sum_{t_v}c_{at_k}\frac{\partial q_k(t_k)}{\partial q_{i\to e}(b)}\right). \tag{F.8}$$

   Only a few of the derivatives $\partial q_k(t_k)/\partial q_{i\to e}(b)$ entering equation (F.8) are non-zero. Hence, the full derivative has negligible order $O(1/N)$.
2. The fact that the external field is constant was already shown in equation (F.2) during the proof of proposition 1.
3. As just proved, we can ignore the external field in the expression of $Z^{i\to e}$, and find

$$\begin{aligned}Z^{i\to e}&\approx\sum_{t_i}q_{i\to e}(t_i)\\ &=\sum_{t_i}n_{t_i}\left(\prod_{g\in\partial i\setminus e}\sum_{t_j:j\in\partial g\setminus i}\pi_g\prod_{j\in\partial g\setminus i}q_{j\to g}(t_j)\right).\end{aligned} \tag{F.9}$$

J. Stat. Mech. (2024) 043403

Utilizing result equation (F.3) in lemma 1, equation (F.9) simplifies to

$$Z^{i\to e} = \sum_{t_i} n_{t_i} \left(\prod_{g\in\partial i\setminus e} c\frac{|g|\,(|g|-1)}{2N} \right) = \left(\prod_{g\in\partial i\setminus e} c\frac{|g|\,(|g|-1)}{2N} \right).$$

which results in equation (F.5), as desired. Similarly, to compute the derivative $\partial Z^{i\to e}/\partial q_{j\to f}(b)$ we can ignore all appearing $\partial\exp(-h(a))/\partial q_{j\to f}(b)$ and $h(t_i)$ thanks to the lemma's first two points (just proved). Hence,

$$\frac{\partial Z^{i\to e}}{\partial q_{j\to f}(b)} = \frac{\partial}{\partial q_{j\to f}(b)} \left[\sum_{t_i} n_{t_i} \left(\prod_{g\in\partial i\setminus e} \sum_{t_j : j\in\partial g\setminus i} \pi_g \prod_{j\in\partial g\setminus i} q_{j\to g}(t_j) \right) \right]$$

$$= \sum_{t_i} n_{t_i} \left(\prod_{g\in\partial i\setminus e,f} \sum_{t_m : m\in\partial g\setminus i} \pi_g \prod_{m\in\partial g\setminus i} q_{m\to g}(t_m) \right) \times \left(\sum_{t_m : m\in\partial f\setminus i,j} \pi_f \prod_{m\in\partial f\setminus i,j} q_{m\to f}(t_m) \right),$$

and using equations (F.3) and (F.4) from lemma 1, we conclude with

$$= \frac{1}{N} \left(\prod_{g\in\partial i\setminus e,f} c\frac{|g|\,(|g|-1)}{2N} \right) \left[\sum_{t_i} n_{t_i} c_{t_i t_j} + c\,(|f|-2) \left(2 + \frac{|f|-3}{2} \right) \right]$$

$$= \frac{c}{N} \left(\prod_{g\in\partial i\setminus e,f} c\frac{|g|\,(|g|-1)}{2N} \right) \left[1 + (|f|-2) \left(2 + \frac{|f|-3}{2} \right) \right].$$

F.3. Elapsed time of MP

In figure F1, we plot the running time of MP when performing the synthetic experiments of section 4.2. Elapsed times become prohibitively large when $c_{\text{out}}/c_{\text{in}}$ increases. For this reason, we threshold the maximum number of MP iterations and obtain the plateaus of figure F1.

J. Stat. Mech. (2024) 043403

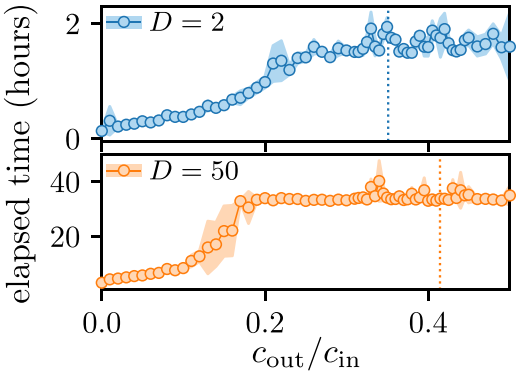

Figure F1. Elapsed time for MP. For both $D = 2$ and $D = 50$, the elapsed times plateau due to the threshold imposed on MP's maximum number of iterations. Shaded areas are standard deviations over five random initializations of MP. Vertical dotted lines are theoretical detectability bounds derived from equation (29).

Appendix G. Calculations of the free energy

After MP, it is possible to approximate the log-evidence of the data, i.e. the log-normalizing constant $\log Z$, as per equation (5). The equivalent quantity $F = -\log Z$, called the free energy of the system, can be obtained via the following cavity-based general formula:

$$F \approx -\sum_{i \in V} f_i + \sum_{e \in \Omega} (|e| - 1)\, f_e\,, \tag{G.1}$$

where

$$f_i = \log\left(\sum_{t_i} n_{t_i} \prod_{e \in \partial i} \sum_{t_j : j \in \partial e \setminus i} \left(\frac{\pi_e}{\kappa_e}\right)^{A_e} \left(1 - \frac{\pi_e}{\kappa_e}\right)^{1 - A_e} \prod_{j \in \partial e \setminus i} q_{j \to e}(t_j)\right)$$

$$f_e = \log\left(\sum_{t_j : j \in \partial e} \left(\frac{\pi_e}{\kappa_e}\right)^{A_e} \left(1 - \frac{\pi_e}{\kappa_e}\right)^{1 - A_e} \prod_{j \in \partial e} q_{j \to e}(t_j)\right).$$

Assuming that MP has converged, all messages $q_{j \to e}(t_j)$ are available. Notice, however, that naive computations of the f_i and f_e addends are unfeasible, due to the exploding sums over $t_j : j \in \partial e$. In the following, we show how such computations can be performed efficiently.

J. Stat. Mech. (2024) 043403

(i) Calculations of f_i.
As one can observe from equations (B.1) and (B.2), the f_i terms are the log-normalizing constants of q_i, therefore they can be computed similarly. In particular, ignoring constants, by equation (B.13), the following simplification holds:

$$f_i = \log\left(\sum_{t_i} n_{t_i} \prod_{\substack{e\in E\\ e\in\partial i}} \sum_{t_j : j\in\partial e\setminus i} \pi_e \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)\exp\left(-h(t_i)\right)\right).$$

The single terms indexed by $e \in E$, i.e. the values $\sum_{t_j : j\in\partial e\setminus i} \pi_e \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)$, are equivalent to the unnormalized messages $\widehat{q}_{e\to j}(t_j)$. For this reason, they can be computed with the same dynamic program presented in appendix D.1.

(ii) Calculations of f_e.
While the f_i terms in equation (G.1) are computed singularly, we take a different approach and calculate the whole sum $\sum_{e\in\Omega}(|e|-1)f_e$ without computing the single f_e, as this would be impossible due to their exploding number. First, we separate the terms over Ω in equation (G.1) as follows.

$$\begin{aligned}
\sum_{e\in\Omega}(|e|-1)\,f_e &= \sum_{e\in\Omega}(|e|-1)\log\left(\sum_{t_j : j\in\partial e}\left(\frac{\pi_e}{\kappa_e}\right)^{A_e}\left(1-\frac{\pi_e}{\kappa_e}\right)^{1-A_e}\prod_{j\in\partial e} q_{j\to e}(t_j)\right)\\
&= \log\left(\prod_{e\in\Omega}\left[\sum_{t_j : j\in\partial e}\left(\frac{\pi_e}{\kappa_e}\right)^{A_e}\left(1-\frac{\pi_e}{\kappa_e}\right)^{1-A_e}\prod_{j\in\partial e} q_{j\to e}(t_j)\right]^{|e|-1}\right)\\
&= \log\left(\prod_{e\in E}\left[\sum_{t_j : j\in\partial e}\pi_e\prod_{j\in\partial e} q_{j\to e}(t_j)\right]^{|e|-1}\right)\\
&\quad + \log\left(\prod_{e\in\Omega\setminus E}\left[\sum_{t_j : j\in\partial e}\left(1-\frac{\pi_e}{\kappa_e}\right)\prod_{j\in\partial e} q_{j\to e}(t_j)\right]^{|e|-1}\right) + \text{const.}.
\end{aligned}$$

This allows us to compute the last two addends separately.

Focusing on the second addend, and proceeding similarly as for the external field calculations that brought us to equation (B.20), we get

$$\begin{aligned}
&\log\prod_{e\in\Omega\setminus E}\left[\sum_{t_j : j\in\partial e}\left(1-\frac{\pi_e}{\kappa_e}\right)\prod_{j\in\partial e} q_{j\to e}(t_j)\right]^{|e|-1}\\
&\qquad\approx \log\prod_{e\in\Omega\setminus E}\left[\exp\left(-\frac{1}{N}\sum_{t_j : j\in\partial e}\left(\frac{\sum_{k<m\in e} c_{t_k t_m}}{\kappa_e}\right)\prod_{j\in\partial e} q_{j\to e}(t_j)\right)\right]^{|e|-1}
\end{aligned}$$

J. Stat. Mech. (2024) 043403

$$
\begin{aligned}
&\approx \log \prod_{e\in\Omega\setminus E}\left[\exp\left(-\frac{1}{N}\sum_{t_j:j\in\partial e}\left(\frac{\sum_{k<m\in e}c_{t_kt_m}}{\kappa_e}\right)\prod_{j\in\partial e}q_j\left(t_j\right)\right)\right]^{|e|-1}\\
&=\log\prod_{e\in\Omega\setminus E}\exp\left(\frac{1-|e|}{N}\sum_{t_j:j\in\partial e}\left(\frac{\sum_{k<m\in e}c_{t_kt_m}}{\kappa_e}\right)\prod_{j\in\partial e}q_j\left(t_j\right)\right)\\
&\approx\log\prod_{e\in\Omega}\exp\left(\frac{1-|e|}{N}\sum_{t_j:j\in\partial e}\left(\frac{\sum_{k<m\in e}c_{t_kt_m}}{\kappa_e}\right)\prod_{j\in\partial e}q_j\left(t_j\right)\right)\\
&=\sum_{e\in\Omega}\frac{1-|e|}{N}\sum_{t_j:j\in\partial e}\left(\frac{\sum_{k<m\in e}c_{t_kt_m}}{\kappa_e}\right)\prod_{j\in\partial e}q_j\left(t_j\right)\\
&=\frac{1}{N}\sum_{e\in\Omega}\frac{1-|e|}{\kappa_e}\sum_{k<m\in e}\sum_{t_j:j\in\partial e}c_{t_kt_m}\prod_{j\in\partial e}q_j\left(t_j\right)\\
&=\frac{1}{N}\sum_{e\in\Omega}\frac{1-|e|}{\kappa_e}\sum_{k<m\in e}\sum_{t_kt_m}c_{t_kt_m}q_k\left(t_k\right)q_j\left(t_j\right)\\
&=\frac{C'''}{N}\sum_{k<m\in V}\sum_{t_kt_m}c_{t_kt_m}q_k\left(t_k\right)q_j\left(t_j\right)
\end{aligned}
$$

where $C''' := \sum_{d=2}^{D}\frac{1-d}{\kappa_d}\binom{N-2}{d-2}$. We also define $q_V(a)=\sum_{k\in V}q_k(a)$. Then,

$$
\begin{aligned}
&\log\prod_{e\in\Omega\setminus E}\left[\sum_{t_j:j\in\partial e}\left(1-\frac{\pi_e}{\kappa_e}\right)\prod_{j\in\partial e}q_{j\to e}\left(t_j\right)\right]^{|e|-1}\\
&\quad\approx\frac{C'''}{N}\sum_{k<m\in V}\sum_{t_kt_m}c_{t_kt_m}q_k\left(t_k\right)q_j\left(t_j\right)\\
&\quad=\frac{C'''}{N}\sum_{ab}c_{ab}\sum_{k<m\in V}q_k\left(a\right)q_j\left(b\right)\\
&\quad=\frac{C'''}{2N}\sum_{ab}c_{ab}\left[\sum_{k,m\in V}q_k\left(a\right)q_j\left(b\right)-\sum_{k\in V}q_k\left(a\right)q_k\left(b\right)\right]\\
&\quad=\frac{C'''}{2N}\sum_{ab}c_{ab}\left[q_V\left(a\right)q_V\left(b\right)-\sum_{k\in V}q_k\left(a\right)q_k\left(b\right)\right]
\end{aligned}
$$

which can be computed in linear time $O(|V|K^2)$.

The first addend requires different considerations. Since naive calculations of every sum on $t_j : j \in \partial e$ cost $O(K^{|e|})$, and thus are unfeasible, we design a dynamic

J. Stat. Mech. (2024) 043403

program similar to that of D.1. For simplicity, consider a hyperedge $e = 1, \ldots, m$. Proceeding as in appendix D.1, we define the quantities:

$$\tilde{\eta}(e,n) = \sum_{t_j : j=1,\ldots,n} \pi_e \prod_{j=1,\ldots,n} q_{j\to e}(t_j) \tag{G.2}$$

$$\tilde{s}_n(a) = \sum_{t_1} p_{at_1} q_{1\to e}(t_1) + \ldots + \sum_{t_{n-1}} p_{at_{n-1}} q_{n-1\to e}(t_{n-1}) \,. \tag{G.3}$$

Notice that $\tilde{\eta}(e,m) = \sum_{t_j : j\in\partial e} \pi_e \prod_{j:j\in\partial e} q_{j\to e}(t_j)$ is the quantity we need to compute. For equations (G.2) and (G.3), the following recursions hold:

$$\tilde{\eta}(e,n) = \tilde{\eta}(e,n-1) + \sum_{t_n} q_{n\to e}(t_n)\, \tilde{s}_n(t_n)$$

$$\tilde{s}_n(a) = \tilde{s}_{n-1}(a) + \sum_{t_{n-1}} p_{t_{n-1}a} q_{n-1\to e}(t_{n-1}) \,.$$

Here, computing the final $\tilde{\eta}(e,m)$ costs $O(K|e|)$, and computing it for all the observed hyperedges costs $\sum_{e\in E} O(K|e|)$. Note that utilizing the dynamic program in appendix D.1 would cost $\sum_{e\in E} O(K^2|e|^2)$, plus the processing needed to obtain the $\tilde{\eta}(e,m)$ value. Hence, the new recursions result in good computing savings with minimal changes to the numerical implementation.

G.1. Computation of the free energy landscape on High School data

We explain further how to obtain the free energy landscape of the High School dataset in figure 5. The three vertices are inferred using the dataset's hyperedges whose size is lower than or equal to D, with $D = 2,3,4$. After having performed inference on every vertex, we obtain the parameters $(p_2,n_2),(p_3,n_3),(p_4,n_4)$—each pair is associated with a value of D—for the affinity matrix and the community prior.

Every point in the simplex is generated with a convex combination of the three vertices. Particularly, we define the parameters

$$p_{\text{simplex}} = \lambda_2 p_2 + \lambda_3 p_3 + \lambda_4 p_4$$
$$n_{\text{simplex}} = \lambda_2 n_2 + \lambda_3 n_3 + \lambda_4 n_4 \,,$$

where $0 \leqslant \lambda_i \leqslant 1$ and $\sum_{i=2,3,4} \lambda_i = 1$. For any value of $p_{\text{simplex}}, n_{\text{simplex}}$, we compute the free energy on the whole High School dataset, i.e. taking all hyperedges. The free energy approximations following equation (G.1) require the messages, marginals and external field, which can be inferred via MP and in turn depend on $p_{\text{simplex}}, n_{\text{simplex}}$. For every point in the simplex, we fix $p_{\text{simplex}}, n_{\text{simplex}}$ and infer all the remaining quantities via MP, to then compute the free energy displayed in figure 5.

G.2. Inference of class affinity on High School data

We expand on the community patterns detected in the High School data for $D = 4$, which are represented in figure 5. The nine classes observed in the data are named after

J. Stat. Mech. (2024) 043403

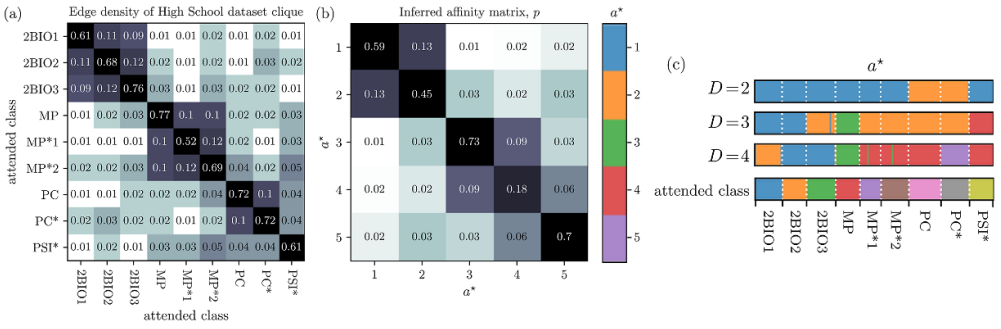

Figure G1. Affinity patterns on the High School dataset. Colors of the matrices' entries correspond to their log values, properly normalized to ease the figure's readability. (a) Edge density on the clique decomposition of the High School dataset. As in Mastrandrea *et al* [58], the edge density between two classes X and Y corresponds to the number of observed edges between nodes of the classes, normalized with respect to the total number of possible edges between X and Y. (b) Affinity matrix p inferred by the EM-MP scheme with $D=4$. The method detects five classes, whose affinity values are as in the matrix's entries. Colors of classes follow the color coding of figure 5(c). (c) Inferred communities of nodes and the partition in the classes of students. The panel is identical to figure 5(c).

their subjects of focus, and are: MP, MP*1, MP*2 (mathematics and physics), PC, PC* (physics and chemistry), PSI* (engineering), 2BIO1, 2BIO2, 2BIO3 (biology) [58].

We compare the the edge density patterns computed on the data in Mastrandrea *et al* [58], and shown in figure G1(a), with the affinity matrix p inferred on the High School dataset fixing $D=4$, shown in figure G1(b). Additionally, in figure G1(c), we plot the partition of the nodes into communities with their labeling in classes.

We observe that classes that are inferred in the same community appear to also belong to classes that have a larger number of external interactions with other classes in the same inferred community. For instance, the BIO classes belong to two communities that are disjoint from all others; see figure G1(c). Within the BIO classes, 2BIO2 and 2BIO3 are grouped in the same community as they have a slightly higher edge density of 0.12, compared to the 0.11 and 0.09 observed for 2BIO1.

The affinity matrix shown in figure G1(b) aligns well with the inter- and intra-community interactions. For instance, communities 1 and 2 (that contain the BIO classes) have an upper diagonal block that isolates them from all others. Communities 3 and 5, which largely match students from classes MP and PC, are disassortative with the remaining classes, grouped in community 4.

G.3. Further comments on higher-order interactions on High School data

The High School hypergraph contains interactions of orders ranging from 2 to 5. In our experiments, we observe that optimal inference is reached at a maximum hyperedge size of $D=4$, while utilizing interactions of order 5 slightly degrades the performance. We confirm this in various ways. The communities inferred (now shown) are less granular than those presented for $D=4$. A similar trend is observed in the free energy

Table G1. AUC scores from the High School dataset. We perform MP and EM inference on the High School dataset utilizing hyperedges up to size D. Then, we compute the AUC on the full dataset, i.e. on the hypergraph with all hyperedges up to $D=5$. The goodness of link prediction, represented by the AUC score, shows that interactions up to size 4 improve the quality of inference, while utilizing interaction of size 5 yields a slight drop in performance.

D	AUC
2	0.710 ± 0.002
3	0.780 ± 0.003
4	0.843 ± 0.004
5	0.813 ± 0.003

(not shown), which slightly increases when performing inference on the whole dataset. Finally, we measure the link prediction performances utilizing parameters inferred with $D=2,3,4,5$, and compute the AUC with respect to the full dataset, which we include in table G1. Here again we observe a slight drop in the AUC when utilizing parameters inferred at $D=5$, despite the AUC being computed with respect to all hyperedges, including those not observed when training on lower values of D.

There could be various reasons for this result. A possible explanation is that the interactions at $D=5$ are noisier and/or less aligned with the data-generating process assumed by our generative model. We recall that the data are collected via proximity sensors, and that social interactions in larger groups are harder to detect, and may arise from different types of link formation mechanisms.

References

[1] Girvan M and Newman M E J 2002 *Proc. Natl Acad. Sci.* **99** 7821–6
[2] Fletcher R J, Revell A, Reichert B E, Kitchens W M, Dixon J D and Austin J D 2013 *Nat. Commun.* **4** 2572
[3] Newman M E J 2001 *Proc. Natl Acad. Sci.* **98** 404–9
[4] Shekhtman L M, Shai S and Havlin S 2015 *New J. Phys.* **17** 123007
[5] Fortunato S 2010 *Phys. Rep.* **486** 75–174
[6] Decelle A, Krzakala F, Moore C and Zdeborová L 2011 *Phys. Rev.* E **84** 066106
[7] Decelle A, Krzakala F, Moore C and Zdeborová L 2011 *Phys. Rev. Lett.* **107** 065701
[8] Moore C 2017 The computer science and physics of community detection: landscapes, phase transitions and hardness (arXiv:1702.00467)
[9] Abbe E 2018 *J. Mach. Learn. Res.* **18** 1–86 (available at: https://jmlr.org/papers/v18/16-480.html)
[10] Ghasemian A, Zhang P, Clauset A, Moore C and Peel L 2016 *Phys. Rev.* X **6** 031005
[11] Taylor D, Shai S, Stanley N and Mucha P J 2016 *Phys. Rev. Lett.* **116** 228301
[12] Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J G and Petri G 2020 *Phys. Rep.* **874** 1–92
[13] Battiston F *et al* 2021 *Nat. Phys.* **17** 1093–8
[14] Zhou D, Huang J and Schölkopf B 2006 Learning with hypergraphs: clustering, classification and embedding *Advances in Neural Information Processing Systems* vol 19, ed B Schölkopf, J Platt and T Hoffman (MIT Press)
[15] Chodrow P S, Veldt N and Benson A R 2021 *Sci. Adv.* **7** eabh1303
[16] Contisciani M, Battiston F and De Bacco C 2022 *Nat. Commun.* **13** 7229
[17] Ruggeri N, Contisciani M, Battiston F and Bacco C D 2023 *Sci. Adv.* **9** eadg9159

J. Stat. Mech. (2024) 043403

[18] Dumitriu I and Wang H 2023 Exact recovery for the non-uniform Hypergraph Stochastic Block Model (arXiv:2304.13139)
[19] Angelini M C, Caltagirone F, Krzakala F and Zdeborová L 2015 Spectral detection on sparse hypergraphs *2015 53rd Annual Allerton Conf. on Communication, Control and Computing* (Allerton) pp 66–73
[20] Chien I, Lin C Y and Wang I H 2018 Community detection in hypergraphs: optimal statistical limit and efficient algorithms *Proc. 21st Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research* vol 84) ed A Storkey and F Perez-Cruz (PMLR) pp 871–9
[21] Liang J, Ke C and Honorio J 2021 Information theoretic limits of exact recovery in sub-hypergraph models for community detection *2021 IEEE Int. Symp. on Information Theory (ISIT)* pp 2578–83
[22] Pal S and Zhu Y 2021 *Random Struct. Algorithms* **59** 407–63
[23] Zhang Q and Tan V Y F 2023 *IEEE Trans. Inf. Theory* **69** 453–71
[24] Gu Y and Polyanskiy Y 2023 Weak recovery threshold for the hypergraph stochastic block model *PMLR* **195** 885–920
[25] Cole S and Zhu Y 2020 *Linear Algebr. Appl.* **593** 45–73
[26] Lin C Y, Chien I E and Wang I H 2017 On the fundamental statistical limit of community detection in random hypergraphs *2017 IEEE Int. Symp. on Information Theory (ISIT)* pp 2178–82
[27] Ghoshdastidar D and Dukkipati A 2014 Consistency of spectral partitioning of uniform hypergraphs under planted partition model *Advances in Neural Information Processing Systems* vol 27, ed Z Ghahramani, M Welling, C Cortes, N Lawrence and K Weinberger (Curran Associates, Inc.)
[28] Yuan M and Shang Z 2021 *Stat* **10** e407
[29] Corinzia L, Penna P, Szpankowski W and Buhmann J 2022 Statistical and computational thresholds for the planted k-densest sub-hypergraph problem *Proc. 25th Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research* vol 151) ed G Camps-Valls, F J R Ruiz and I Valera (PMLR) pp 11615–40
[30] Jin J, Ke Z T and Liang J 2021 Sharp impossibility results for hyper-graph testing *Advances in Neural Information Processing Systems* vol 34, ed M Ranzato, A Beygelzimer, Y Dauphin, P Liang and J W Vaughan (Curran Associates, Inc.) pp 7220–31
[31] Yuan M, Liu R, Feng Y and Shang Z 2022 *Ann. Stat.* **50** 147–69
[32] Chodrow P, Eikmeier N and Haddock J 2023 *SIAM J. Math. Data Sci.* **5** 251–79
[33] Pearl J 1982 Reverend bayes on inference engines: a distributed hierarchical approach *Proc. 2nd AAAI Conf. on Artificial Intelligence (AAAI'82)* (AAAI Press) pp 133–6
[34] Mézard M and Montanari A 2009 *Information, Physics and Computation* (Oxford University Press)
[35] Murphy K P 2012 *Machine Learning: A Probabilistic Perspective* (MIT Press)
[36] Mézard M, Parisi G and Virasoro M 1986 *Spin Glass Theory and Beyond* (World Scientific)
[37] Mézard M and Parisi G 2001 *Eur. Phys. J.* B **20** 217–33
[38] Holland P W, Laskey K B and Leinhardt S 1983 *Soc. Netw.* **5** 109–37
[39] Wasserman S and Faust K 1994 *Social Network Analysis: Methods and Applications Structural Analysis in the Social Sciences* (Cambridge University Press)
[40] Kamiński B, Prałat P and Théberge F 2023 *J. Complex Netw.* **11** cnad028
[41] Ruggeri N, Battiston F and Bacco C D 2023 A framework to generate hypergraphs with community structure *Phys. Rev.* E **109** 034309
[42] Ruggeri N, Lonardi A and De Bacco C 2023 Message-passing on hypergraphs: detectability, phase transitions and higher-order information *GitHub Repository* (available at: https://github.com/nickruggeri/hypergraph-message-passing)
[43] Cantwell G T and Newman M E J 2019 *Proc. Natl Acad. Sci.* **116** 23398–403
[44] Kirkley A, Cantwell G T and Newman M E J 2021 *Sci. Adv.* **7** eabf1211
[45] Dempster A P, Laird N M and Rubin D B 1977 *J. R. Stat. Soc.* B **39** 1–38
[46] Chodrow P S 2020 *J. Complex Netw.* **8** cnaa018
[47] Kesten H and Stigum B 1967 *J. Math. Anal. Appl.* **17** 309–38
[48] Kesten H and Stigum B P 1966 *Ann. Math. Stat.* **37** 1463–81
[49] Mézard M and Montanari A 2006 *J. Stat. Phys.* **124** 1317–50
[50] Schneidman E, Berry M J, Segev R and Bialek W 2006 *Nature* **440** 1007–12
[51] Giusti C, Pastalkova E, Curto C and Itskov V 2015 *Proc. Natl Acad. Sci.* **112** 13455–60
[52] Merchan L and Nemenman I 2016 *J. Stat. Phys.* **162** 1294–308
[53] Schneidman E, Still S, Berry M J and Bialek W 2003 *Phys. Rev. Lett.* **91** 238701
[54] Blei D M, Ng A Y and Jordan M I 2003 *J. Mach. Learn. Res.* **3** 993–1022 (available at: https://jmlr.csail.mit.edu/papers/v3/blei03a.html)

J. Stat. Mech. (2024) 043403

[55] Campbell L L 1966 *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* vol 5 (Wiley) pp 217–25
[56] Cover T and Thomas J 2006 *Elements of Information Theory* (Wiley)
[57] Young J G, Petri G and Peixoto T P 2021 *Commun. Phys.* **4** 135
[58] Mastrandrea R, Fournet J and Barrat A 2015 *PLoS One* **10** 1–26
[59] Newman M E J and Clauset A 2016 *Nat. Commun.* **7** 11863
[60] Contisciani M, Power E A and De Bacco C 2020 *Sci. Rep.* **10** 15736
[61] Badalyan A, Ruggeri N and De Bacco C 2023 Hypergraphs with node attributes: structure and inference (arXiv:2311.03857)
[62] Landry N W, Young J G and Eikmeier N 2023 The simpliciality of higher-order networks *EPJ Data Sc.* **13** 17
[63] Ramanujan S 1987 *The Lost Notebook and Other Unpublished Papers* (Narosa)

J. Stat. Mech. (2024) 043403