Journal of Complex Networks (2020) **00**, 1–26
doi: 10.1093/comnet/cnaa018

Configuration models of random hypergraphs

PHILIP S. CHODROW†
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
†Corresponding author. Email: pchodrow@gmail.com

Edited by: James Gleeson

[Received on 13 December 2019; editorial decision on 13 April; accepted on 29 June 2020]

Many empirical networks are intrinsically polyadic, with interactions occurring within groups of agents of arbitrary size. There are, however, few flexible null models that can support statistical inference in polyadic networks. We define a class of null random hypergraphs that hold constant both the node degree and edge dimension sequences, thereby generalizing the classical dyadic configuration model. We provide a Markov Chain Monte Carlo scheme for sampling from these models and discuss connections and distinctions between our proposed models and previous approaches. We then illustrate the application of these models through a triplet of data-analytic vignettes. We start with two classical topics in network science—triadic clustering and degree-assortativity. In each, we emphasize the importance of randomizing over hypergraph space rather than projected graph space, showing that this choice can dramatically alter both the quantitative and qualitative outcomes of statistical inference. We then define and study the edge intersection profile of a hypergraph as a measure of higher-order correlation between edges, and derive asymptotic approximations for this profile under the stub-labeled null. We close with suggestions for multiple avenues of future work. Taken as a whole, our experiments emphasize the ability of explicit, statistically grounded polyadic modelling to significantly enhance the toolbox of network data science.

1. Introduction

Graphs provide parsimonious mathematical descriptions of systems in which agents, represented by nodes, partake in dyadic relationships, represented by edges. When analysing a graph derived from empirical data, a common task is to compare an observable of interest to its distribution under a suitably specified null model. The outcomes of comparisons often depend strongly on the choice of, and careful consideration by the analyst is therefore required. A widely used 'default' is the class of *configuration models* [1–4]. Configuration models preserve the graph's degree sequence, which counts the number of edges incident to each node. These counts are natural first-order statistics of the graph, which are known to constrain many macroscopic graph properties [5]. Preserving these counts gives a natural null model constraint: properties observed in data that are not present in a configuration model require explanation in terms of higher-order graph structures.

In many systems of contemporary interest, groups of arbitrary size may interact simultaneously. Social contact networks may record gatherings of groups of arbitrary size [6, 7]. Collaboration networks log papers or projects in which many have participated [8–11]. Digital communications may be sent to many receivers simultaneously [12]. Arbitrary numbers of classifiers—ranging from patent categories [13] to online tags [14]—can be used to describe objects. In the past decades, the dominant approach to these systems has been to represent these networks dyadically, allowing the analyst to apply standard techniques of dyadic network science, including the configuration model. Recent work, however, has highlighted

limitations of the dyadic paradigm in modelling of polyadic systems, in theory [15] and in application domains including neuroscience [16], ecology [17], computational social science [18, 19] among others [14]. Scholarly collaboration networks illustrate one of these limitations. Suppose that A, B and C are collaborators in the same field. A three-author paper by A, B and C can be represented by a polyadic edge of the form $e = (A, B, C)$. The standard dyadic projection enumerates all pairwise relationships contained in this single interaction, generating the *three* edges $e_1 = (A, B)$, $e_2 = (B, C)$ and $e_3 = (A, C)$. This is exactly the same representation as would be obtained if each pair of authors wrote a two-author paper. The same dyadic representation thus fails to distinguish a world containing one three-author paper from a world containing three two-author papers. These worlds differ in the number of papers produced; the inferred productivity of the scholars; and likely in the content of the papers as well, since author lists are strong indicators of subject matter. In cases where these differences are relevant to our study questions, we can expect the dyadic projection to yield misleading results.

Because of these limitations, it is desirable to construct random models for polyadic data that inherit the useful properties of the dyadic configuration model. In this article, we construct two such models and demonstrate their utility for polyadic network data science. Along the way, we argue for two principle theses. First, the choice between dyadic and polyadic null models can determine the qualitative findings of standard network analyses. Second, the use of polyadic nulls allows the analyst to measure and test novel, rich measures of polyadic structure, thereby expanding the network-scientific toolbox.

Outline

We begin in Section 2 with a survey of the landscape of null models for relational data, including the dyadic configuration model, random hypergraphs, and random simplicial complexes. In Section 3, we define stub- and vertex-labeled configuration models of random hypergraphs. Practical application of these models requires a sampling scheme, which we provide in Section 4. We then turn to a triplet of illustrative applications in Section 5. We first consider triadic closure, showing that benchmarking against hypergraph nulls can distinguish between different generative mechanisms in polyadic social networks. We then turn to degree-assortativity, where hypergraph data representations allow us to define and test multiple assortativity coefficients, each of which captures a qualitatively different form of intra-edge degree correlation. Finally, we introduce a novel measure of correlation between polyadic edges, which can be tested against either the full configuration model or analytic approximations. We close in Section 6 with a summary of our findings and suggestions for future development.

2. Graphs, hypergraphs and simplicial complexes

Random graph null modelling has a rich history; see [3] for a review. In this section, we take a rapid tour through some of the most important results in configuration-type models for graphs and their generalizations. We begin with a brief review of the configuration model for dyadic graphs.

DEFINITION 1. (Graph). A *graph* $G = (V, E)$ consists of a set V of nodes or vertices and a multiset E. An element of E is an unordered pair $e = (u, v)$ of nodes, also called an *edge*. We assume that both sets V and E are endowed with an (arbitrary) order. An edge of the form (u, u) is called a *self-loop*. Two distinct edges e_1 and e_2 are *parallel* if they are equal as sets. We denote by $\mathcal{G}^{\circlearrowleft}$ the set of all graphs on n nodes, and by $\mathcal{G} \subset \mathcal{G}^{\circlearrowleft}$ the set of graphs on n nodes without self-loops.

While it is indeed possible to define configuration models on $\mathcal{G}^{\circlearrowleft}$ [3, 20], we do not do so here. We rule-out self-loops because (a) their presence considerably complicates sampling algorithms and (b) most polyadic data sets do not possess a meaningful notion of self-interaction. We will therefore present most of our results for elements of $\mathcal{G}$ and its generalization to hypergraphs, only discussing $\mathcal{G}^{\circlearrowleft}$ below in the context of certain technical issues.

Let $n = |V|$ and $m = |E|$ be fixed and finite. Let $\mathbb{I}$ be the indicator function of its argument.

DEFINITION 2. (Degree sequence). The *degree sequence* $\mathbf{d} = \deg(G)$ of a graph $G = (V, E)$ is the vector $\mathbf{d} \in \mathbb{Z}^n$ defined componentwise as

$$d_v = \sum_{e \in E} \mathbb{I}(v \in e). \tag{1}$$

A configuration model is a probability distribution on the set $\mathcal{G}_{\mathbf{d}} = \{G \in \mathcal{G} : \deg(G) = \mathbf{d}\}$ of graphs with degree sequence $\mathbf{d}$. There are two closely related model variants which should be distinguished [3]. On its first introduction [2], the configuration model was defined mechanistically through a 'stub-matching' algorithm. To perform stub-matching, we place d_v labeled half-edges (or 'stubs') into an urn for each node v. We draw half-edges two at a time, with each draw producing an edge. The result of this process is a set of pairs of half-edges: a stub-labeled graph.

DEFINITION 3. (Stub-labeled graphs). For a fixed node set V and degree sequence $\mathbf{d}$, define the multiset

$$\Sigma_{\mathbf{d}} = \biguplus_{v \in V} \{v_1, \ldots, v_{d_v}\},$$

where $\uplus$ denotes multiset union. The copies $v_1, \ldots, v_{d_v}$ are called *stubs* of node v. A *stub-labeled graph* $S = (V, E)$ consists of the node set V and an edge set E which partitions $\Sigma_{\mathbf{d}}$ into unordered pairs. Each element of E has the form (u_i, v_j) for some nodes u and v and stub indices i and j. An edge of the form (v_i, v_j) is called a *self-loop*. Let $\mathcal{S}^{\circlearrowleft}$ be the set of stub-labeled graphs, and $\mathcal{S} \subset \mathcal{S}^{\circlearrowleft}$ the set of stub-labeled graphs without self-loops.

Technically speaking, one should remember that the set $\mathcal{S}^{\circlearrowleft}$ of stub-labeled graphs is not a subset of the set $\mathcal{G}^{\circlearrowleft}$ of graphs, since the objects in the edge set are of different logical types. The same is true of the sets $\mathcal{S}$ and $\mathcal{G}$. These technical considerations will also apply when we generalize to hypergraphs below but will not present any major practical issues.

There is a natural surjection $g : \mathcal{S}^{\circlearrowleft} \to \mathcal{G}^{\circlearrowleft}$. If $S \in \mathcal{S}^{\circlearrowleft}$, $g(S) \in \mathcal{G}^{\circlearrowleft}$ is the graph obtained by erasing the stub-labels: each stub v_i in S is recorded as an unlabeled copy of v in $g(S)$. The stub-labeled graph S and graph $g(S)$ are topologically identical, differing only in the presence of stub-labels in S. We use the notation $A = g^{-1}(G)$ to refer to the preimage $A \subseteq \mathcal{S}$ of $G \subseteq \mathcal{G}$ by g. We emphasize that g is not a bijection, and the symbol g^{-1} should not be interpreted as an inverse of g. We define $\mathcal{S}^{\circlearrowleft}_{\mathbf{d}}$ to be $g^{-1}\left(\mathcal{G}^{\circlearrowleft}_{\mathbf{d}}\right)$. Note that an edge $\tilde{e} \in S$ is a self-loop if and only if its image $e \in g(S)$ is. It follows that $\mathcal{S} = g^{-1}(\mathcal{G})$. It is therefore natural to define $\mathcal{S}_{\mathbf{d}} = g^{-1}(\mathcal{G}_{\mathbf{d}})$.

DEFINITION 4. (Vertex-labeled dyadic configuration model [3]). The *vertex-labeled configuration model* with degree sequence $\mathbf{d}$ is the uniform distribution $\eta_{\mathbf{d}}$ on $\mathcal{G}_{\mathbf{d}}$.

DEFINITION 5. (Stub-labeled dyadic configuration model [3]). Let $\lambda_{\mathbf{d}}$ be the uniform distribution on $\mathcal{S}_{\mathbf{d}}$. The *stub-labeled configuration model* with degree sequence $\mathbf{d}$ is the distribution $\mu_{\mathbf{d}} = \lambda_{\mathbf{d}} \circ g^{-1}$.

Here and below, the binary operator $\circ$ denotes composition of functions: $(\lambda_{\mathbf{d}} \circ g^{-1})(G) = \lambda_{\mathbf{d}}(g^{-1}(G))$. As originally defined by the authors of [3], the stub-labeled configuration model is the uniform distribution $\lambda_{\mathbf{d}}$ over the space of stub-labeled graphs. We find it somewhat more natural to define the stub-labeled model as the pushforward $\mu_{\mathbf{d}}$ of that distribution to the space of (non-stub-labeled) graphs, which enables clearer reasoning about comparisons to the vertex-labeled model.

Intuitively, the vertex-labeled configuration model assigns the same probability to each graph with degree sequence $\mathbf{d}$, while the stub-labeled model weights these graphs according to their likelihood of being realized via stub-matching. One of the key insights of [2], since generalized by works such as [4, 21], is that these two models are related. Let $\mathcal{G}_{\text{simple}}$ be the set of *simple graphs*, which contain neither self-loops nor parallel edges. Then, a small amount of algebra is sufficient to show that $\mu_{\mathbf{d}}(G|G \in \mathcal{G}_{\text{simple}}) = \eta_{\mathbf{d}}(G|G \in \mathcal{G}_{\text{simple}})$. Furthermore, when the degree sequence is sampled from a fixed distribution with finite second moment, $\mu_{\mathbf{d}}(G \in \mathcal{G}_{\text{simple}})$ is bounded away from zero as n grows large (see e.g. [21]). This implies that repeated sampling from $\mu_{\mathbf{d}}$ will produce a simple graph in a number of repetitions that is asymptotically constant with respect to n. As a result, in the 'large, sparse regime,' it is possible to sample from the stub-labeled configuration model until a simple graph is obtained, which will then be distributed according to the vertex-labeled model. This relationship is extremely convenient, enabling asymptotic analytic expressions for many quantities of theoretical and practical interest [5].

This close relationship between models is likely the reason why the distinction between them has often been elided in applied network science. Recently, however, the authors of [3] pointed out that, in many data sets, these two models are not interchangeable. It is important to distinguish them when the data may possess multi-edges or self-loops and when the number of edges $m = |E|$ is high relative to the number of nodes n. The first condition is important because stub- and vertex-labeled models agree only on the subspace of simple graphs, not the full space of multigraphs. When we wish to study data represented by non-simple graphs, the convenient equivalence described above no longer applies. The second condition locates us away from the large, sparse regime and implies that parallel edges will occur under stub-matching with high probability. In this case, though equivalence between models is formally preserved, an impractical number of repetitions of stub-matching may be required to produce a simple graph. These considerations motivate the authors' development of dedicated Monte Carlo schemes for sampling from both vertex- and stub-labeled configuration models.

From a modelling perspective, the choice of vertex- or stub-labeling must depend on domain-specific reasoning about counterfactual comparisons. Roughly, stub-labeling should be used when, for a fixed graph $G \in \mathcal{G}$, the elements of the set $g^{-1}(G) \subset \mathcal{S}$ have distinct identities in the context of the application domain. This corresponds to asking whether permutations of stubs lead to meaningfully different counterfactual data sets. In contrast, when stub-permutations are either non-sensical or are considered to leave the observed data unchanged, vertex-labeling is to be preferred. For example, in [3], the authors argue that vertex-labeled nulls are most appropriate for studying a collaboration network of computational geometers. Their reason is that stubs in this case correspond to an author's participation in a paper. It is nonsensical to say that A's first collaboration with B is B's second collaboration with A, and therefore stub-labeling is inappropriate.

2.1 *Beyond graphs*

Configuration models and their variants have played a fundamental role in the development of modern network science. The seminal paper by Molloy and Reed [22] has, according to Google Scholar, been

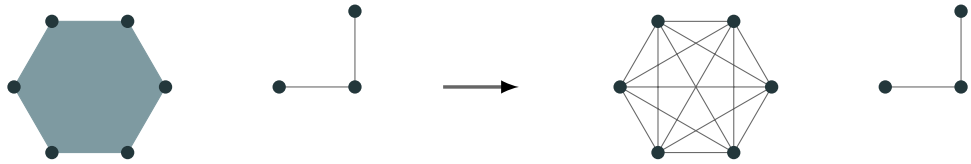

FIG. 1. A synthetic coauthorship network with $n = 9$ nodes. On the left, the network represented as a hypergraph with 3 hyperedges. On the right, the projected graph with 17 dyadic edges.

cited at least 2,000 times since its publication, and over 800 times since 2015. How can we extend these extremely useful models for application to polyadic data sets? A direct approach, taken in early studies such as [10], is to compute the *projected (dyadic) graph*. The projected graph represents each k-adic interaction as a k-clique, which contains an edge between each of the possible $\binom{k}{2} = \frac{n(n-1)}{2}$ pairs of nodes (Fig. 1). The resulting dyadic graph may then be randomized according to vertex- or stub-labeled dyadic configuration models. Projecting, however, can have unintended and occasionally counterintuitive consequences. First and most clearly, all properties which depend explicitly on the presence of higher-dimensional interactions are lost. Second, other observables such as node degrees and edge multiplicities may be transformed in undesirable ways; for example, a single interaction between six agents becomes 15 pairwise interactions after projection. As consequence, each of the six agents involved in a single 6-adic interaction are dyadically represented as nodes of degree 5. Third, and most subtly, projecting transforms the null space for downstream hypothesis testing in ways that may not be intended. For example, projecting the network in Fig. 1 prior to randomization implicitly chooses a null space of counterfactuals consisting of 17 dyadic interactions. This may be undesirable, especially when the null is viewed as a candidate data generating process. Given that the data possesses higher-order interactions, a null model that is by construction unable to produce such interactions may be implausible as a generator of relevant counterfactuals.

2.2 *Random hypergraphs*

Considerations such as these motivate the development of dedicated random null models for polyadic data. Such models enable the analyst to delay or omit dyadic projection when conducting null hypothesis testing. We now make a brief survey of efforts to define configuration-type models for polyadic data. Hypergraphs provide the most general data structure for such models.

DEFINITION 6. (Hypergraph). A *hypergraph* $H = (V, E)$ consists of a node set V and an edge set $E = \{\Delta_j\}_{j=1}^m$ which is a multiset of multisets of V. Each subset is called a *hyperedge*, *edge*, or, in some contexts, a *simplex*. Two hyperedges are *parallel* if they are equal as multisets. A hyperedge is *degenerate* if it contains two copies of the same node. Let $\mathcal{H}^{\circlearrowleft}$ be the set of all hypergraphs and $\mathcal{H}$ the set of all hypergraphs without degenerate edges.

Hypergraphs are straightforward generalizations of graphs in which each edge is permitted to have an arbitrary number of nodes. Degenerate hyperedges generalize the notion of self-loops in dyadic graphs. As before, parallel edges are permitted in $\mathcal{H}$. We continue to let $n = |V|$ and $m = |E|$. The definition of $\mathbf{d}$ in (1) remains unchanged.

Extant literature provides several approaches to defining null distributions on hypergraphs. One of the earliest approaches takes a somewhat indirect route through bipartite graphs [5]. A bipartite graph

contains nodes of two classes, with connections permitted only between nodes of differing classes. To construct a bipartite graph B from a hypergraph H, one constructs a class of nodes in B corresponding to the nodes V of H, and a second class of nodes in B corresponding to the edges E of H. A node v is linked to an edge-node e iff $v \in e$ in H. We can now apply dyadic configuration models to randomize B, before recovering a hypergraph by projecting B onto its node layer. This approach is natural and convenient, but as we will discuss in Section 4 can be limiting in terms of the types of hypergraph nulls from which we can conveniently sample.

A more direct approach is to define a null distribution directly over $\mathcal{H}$. In [23], the authors define an analog of the stub-labeled configuration model over the set of hypergraphs in which all edges have three nodes, in the service of studying a tripartite tagging network on an online platform. Somewhat more general models have been formulated for the purposes of community detection in hypergraphs via modularity maximization, which requires the specification of a suitable null. In [24], the authors develop a degree-preserving randomization via a 'corrected adjacency matrix', which may then be used for modularity maximization on the projected dyadic graph. In [25], the authors explicitly generalize the model of Chung and Lu [26], which preserves degrees in expectation, to non-uniform hypergraphs.

One subspecies of hypergraph has received additional attention. A *simplicial complex* is a hypergraph with additional structure imposed by a subset-inclusion relation: if edge $\Delta \in E$, then, for any subset $\Gamma \subseteq \Delta$, $\Gamma \in E$. Simplicial complexes are attractive tools when studying topological aspects of discrete data [27], since the inclusion condition enables often-dramatic data compression while preserving topological features of interest. Configuration models of simplicial complexes provide one route for conducting null hypothesis tests of such features. The model of [28] achieves analytic tractability by restricting to simplicial complexes with maximal hyperedges of uniform dimension. The authors [29] allow heterogeneous dimensions but sacrifice analytic tractability, instead applying Markov Chain Monte Carlo to sample from the space. In applying any model based on simplicial complexes, it is important to remember that subset inclusion is strong property suited only to certain data-scientific contexts. Particular problems arise when edges possess the semantics of interaction, as in collaboration networks. Suppose that authors A, B and C jointly coauthor a paper. Using hypergraphs, we would represent this collaboration via an edge (A, B, C). In the setting of simplicial complexes, on the other hand, subset inclusion would also require us to include the edges (A, B), (B, C), (A, C), (A), (B) and (C). This poses a similar issue to that of the dyadic projection described above: we are not guaranteed that B and C, say, wrote a two-author paper. While simplicial complex modelling may be useful in carefully selected application areas, in other cases we may require more flexible configuration models defined on more general hypergraphs.

3. Two hypergraph configuration models

We now construct two configuration models for general hypergraphs. Our models generalize the vertex- and stub-labeled dyadic configuration models of [3] reviewed in the previous section.

Throughout this section, H denotes a fixed hypergraph. We use Greek characters to denote edges of a hypergraph H and English characters to denote tuples of nodes. For example, the statement $\Delta = R$ describes the event that an edge Δ of H has fixed location $R = (u_1, u_2, u_3, \ldots)$. When R is a set, $\binom{R}{\ell}$ will denote the set of all subsets of R of size ℓ. When k is a scalar, $\binom{k}{\ell}$ will denote the standard binomial coefficient. The identity $\left|\binom{R}{\ell}\right| = \binom{|R|}{\ell}$ relates these two notational uses. Let $\mathbb{I}$ give the indicator function of its argument.

DEFINITION 7. (Degree and dimension sequences). The *degree sequence* $\mathbf{d} \in \mathbb{Z}_+^n$ and *dimension sequence* $\mathbf{k} \in \mathbb{Z}_+^m$ of a hypergraph H are defined componentwise as

$$d_v = \sum_{e \in E} \mathbb{I}(v \in e) \quad \text{and} \quad k_e = \sum_{v \in V} \mathbb{I}(v \in e) \ .$$

Let $\mathcal{H}^{\circlearrowleft}_{\mathbf{d},\mathbf{k}}$ and $\mathcal{H}_{\mathbf{d},\mathbf{k}}$ denote the sets of hypergraphs with the specified degree and edge dimension sequences with and without degenerate hyperedges, respectively. We say that the sequences $\mathbf{d}$ and $\mathbf{k}$ are *configurable* if $\mathcal{H}_{\mathbf{d},\mathbf{k}} \neq \emptyset$. A criterion for configurability in terms of the majorization of vectors appears to have been obtained independently in early work by Gale and Ryser [30, 31].

DEFINITION 8. (Vertex-labeled hypergraph configuration model). The *vertex-labeled configuration model* $\eta_{\mathbf{d},\mathbf{k}}$ is the uniform distribution on $\mathcal{H}_{\mathbf{d},\mathbf{k}}$.

The stub-labeled configuration model is defined similarly to the dyadic case.

DEFINITION 9. (Stub-labeled hypergraphs). Let

$$\Sigma = \biguplus_{v \in N} \{v_1, \ldots, v_{d_v}\}$$

be a multiset of *stubs*. For each v, d_v copies of v appear in Σ. A *stub-labeled hypergraph* S has as its edge set E a partition of Σ.

The map g extends naturally to the space of hypergraphs. Let $\mathcal{S}$ be the set of stub-labeled hypergraphs. Then, if $S \in \mathcal{S}$, $H = g(S) \in \mathcal{H}$ is the hypergraph obtained by erasing stub-labels: each stub v_i in S is recorded as an unlabeled copy of v in $g(S)$. Define $\mathcal{S}_{\mathbf{d},\mathbf{k}} = g^{-1}(\mathcal{H}_{\mathbf{d},\mathbf{k}})$

DEFINITION 10. (Stub-labeled hypergraph configuration model). Let $\lambda_{\mathbf{d},\mathbf{k}}$ be the uniform distribution on $\mathcal{S}_{\mathbf{d},\mathbf{k}}$. Then, the *stub-labeled configuration model* is the distribution $\mu_{\mathbf{d},\mathbf{k}} = \lambda_{\mathbf{d},\mathbf{k}} \circ g^{-1}$.

We have now defined two hypergraph configuration models, generalizing the vertex- and stub-labeled models of [3]. The vertex-labeled configuration model is the entropy-maximizing distribution on $\mathcal{H}_{\mathbf{d},\mathbf{k}}$ in the case that the identities of stubs are not meaningful, while the stub-labeled configuration model is the entropy-maximizing distribution when these identities are meaningful. The same considerations discussed in [3] (and briefly in the previous section) apply to the question of when to use which null model.

Figure 2 illustrates some of the considerations in play. At left, we show two stub-labeled hypergraphs S_1 and S_2, depicted as bipartite graphs. Circles represent nodes, while squares represent edges. The associated hypergraph is $H = g(S_1) = g(S_2) = \{(u, v, w), (u, v, w)\}$, which contains two parallel edges of dimension 3. The stub-labelling is reflected as labels on the bipartite edges. The stub-labeled configuration model treats each of S_1 and S_2 as distinct objects in sample space, while the vertex-labeled model treats them as alternative representations of the same object. Because of this, stub-labelling should generally only be chosen when the distinct arrangements of stubs can be given valid interpretations in domain context. As the authors of [3] note, such cases are rare, and vertex-labelling is usually to be preferred. Figure 2 also provides a flow chart that illustrates the kind of reasoning that can be used when selecting between these models, using the stub-labeled hypergraphs S_1 and S_2 as guides.

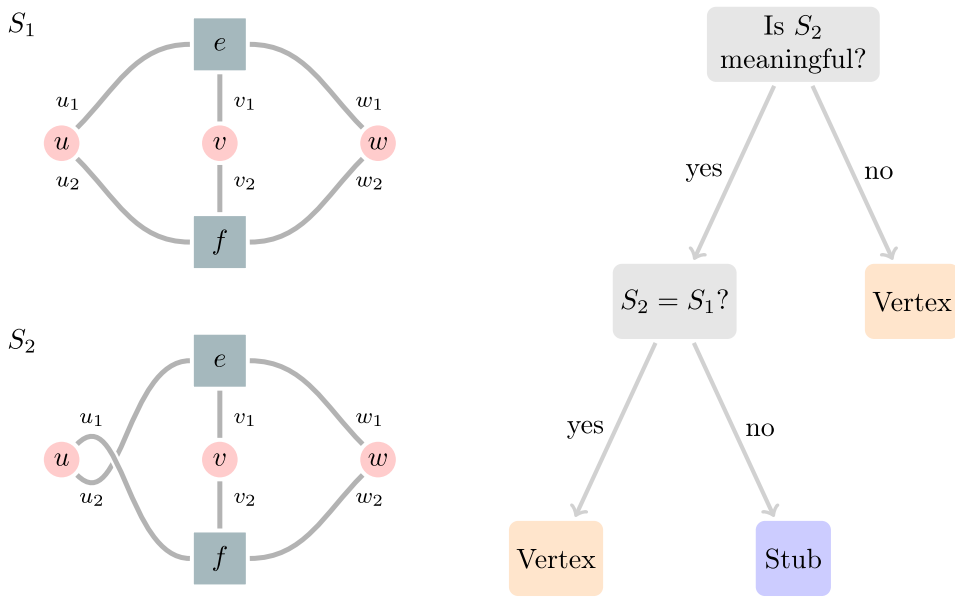

FIG. 2. Choosing between stub- and vertex-labeled models. (Left): Bipartite representations of two stub-labeled hypergraphs S_1 and S_2 which correspond to the same hypergraph: $g(S_1) = g(S_2)$. The squares e and f denote edges; circles u, v and w denote nodes, and an edge (u, e) denotes that $u \in e$. (Right): A schematic flow chart for choosing stub- or vertex-labeled randomization depending on the interpretation of S_1 and S_2 in domain context.

4. Sampling

Stub-matching is a classical method for sampling from the stub-labeled dyadic configuration model [2], and extends naturally to the case of random hypergraphs.

Algorithm 1: Hypergraph stub-matching

Input: Configurable $\mathbf{d} \in \mathbb{Z}_+^n$ and $\mathbf{k} \in \mathbb{Z}_+^m$.
Initialization: $j \leftarrow 1$, $S \leftarrow \emptyset$, $\Sigma \leftarrow \biguplus_{v \in V} \{v_1, \ldots, v_{d_v}\}$
for $j = 1, \ldots, m$ **do**
  $R \leftarrow \text{Uniform}\binom{\Sigma}{k_j}$
  $\Sigma \leftarrow \Sigma \setminus R$
  $S \leftarrow S \cup \{R\}$
end
Output: S

Since any stub-labeled graph S is as likely as any other under Algorithm 1, the output, conditioned on non-degeneracy, is distributed according to $\mu_{\mathbf{d},\mathbf{k}}$. As in the dyadic setting, there is non-zero probability for the output of stub-matching to produce a degenerate hypergraph. This probability will generally be large in the presence of highly heterogeneous node degrees—a common phenomenon in empirical data. Many iterations of Algorithm 1 may therefore be necessary in order to generate a single valid sample.

Because of this, pure stub-matching is often not a practical method for generating random hypergraphs. Stub-matching nevertheless will be helpful in our development, as we will appeal to it for several proofs involving $\mu_{\mathbf{d},\mathbf{k}}$.

We consider a Markov Chain Monte Carlo (MCMC) approach to sampling, in which we use successive, small alterations to the edge set E in order to systematically explore the space $\mathcal{H}_{\mathbf{d},\mathbf{k}}$. Our scheme extends to hypergraphs the broad class of edge-swap MCMC samplers, which have also been formulated for marginal-constrained binary matrices [32, 33]; degree-regular [34–36] and degree-heterogeneous [37–40] simple graphs; degree-constrained graphs [3]; bipartite graphs with degree constraints [41] and graphs with prescribed degree correlations [42].

It is most convenient to work in the space $S_{\mathbf{d},\mathbf{k}}$ of stub-labeled hypergraphs. Recall that $\lambda_{\mathbf{d},\mathbf{k}}$ is the uniform distribution on this space.

DEFINITION 11. (Pairwise reshuffle). Let $S \in \mathcal{S}_{\mathbf{d},\mathbf{k}}$, and $\Delta, \Gamma \in S$. A *pairwise reshuffle* $b(\Delta, \Gamma|S)$ of Δ and Γ is a sample from the conditional distribution $\mu(\cdot|E \setminus \{\Delta, \Gamma\})$.

Depending on context, we will regard a pairwise reshuffle as either a random map on stub-labeled hypergraphs or on pairs of hyperedges. To illustrate, there are $\binom{5}{3} = 10$ pairwise reshuffles of edges $e_1 = (u, v)$ and $e_2 = (x, y, z)$, of which two are $\{(u, x, z), (v, w)\}$ and $\{(x, y), (u, v, z)\}$.

LEMMA 1. Let $S \in \mathcal{S}$. Let $b(\Delta, \Gamma|H) = (\Delta', \Gamma')$ be a pairwise reshuffle which results in $S' \in \mathcal{S}$. Then,

(1) The degree and dimension sequences are preserved: $\deg(S) = \deg(S')$ and $\dim(S) = \dim(S')$.
(2) The intersection of Δ and Γ is preserved: $\Delta' \cap \Gamma' = \Delta \cap \Gamma$.
(3) Any given realization of b occurs with probability

$$q_\mu(\Delta, \Gamma) = 2^{-|\Delta \cap \Gamma|} \binom{|\Delta| + |\Gamma| - 2|\Delta \cap \Gamma|}{|\Delta| - |\Delta \cap \Gamma|}^{-1}. \tag{2}$$

Proof. A pairwise reshuffle may be performed via the following sequence, which is an alternative description of the final two iterations, conditioned on non-degeneracy, of stub-matching (Algorithm 1).

(1) Delete Δ and Γ from E.
(2) Construct Δ' and Γ' as (initially empty) node sets.
(3) For each node $v \in \Delta \cap \Gamma$, add a v-stub to both Δ' and Γ'.
(4) From the remaining stubs, select $|\Delta \setminus \Gamma|$ stubs uniformly at random and add them to Δ'. Add the remainder to Γ'.
(5) Add Δ' and Γ' to E.

Each node begins with the same number of edges as it started, so degrees are preserved. Next, by construction, $|\Delta'| = |\Delta \cap \Gamma| + |\Delta \setminus \Gamma| = |\Delta|$, and similarly $|\Gamma'| = |\Gamma|$. The edge dimension sequence is thus also preserved.

Finally, by construction, step 2 above preserves the intersection $\Delta \cap \Gamma$. There are $2^{|\Delta \cap \Gamma|}$ ways to assign stubs to this intersection. There are a total of $|\Delta| + |\Gamma| - 2|\Delta \cap \Gamma|$ remaining stubs, and of these one

must choose $|\Delta| - |\Delta \cap \Gamma|$ to be placed in Δ. We infer that any given pairwise reshuffle is realized with probability given by (2), as was to be shown. □

We now define the transition kernel of a first-order Markov chain on the space $\mathcal{S}_{\mathbf{d},\mathbf{k}}$. Write $S \sim_{\Delta,\Gamma} S'$ if there exists a pairwise shuffle b such that $b(\Delta, \Gamma|S) = S'$. Note that, since each element of each edge has a distinct label in $\mathcal{S}_{\mathbf{d},\mathbf{k}}$, for any S and S' there is at most one pair (Δ, Γ) such that $S \sim_{\Delta,\Gamma} S'$. If no such pair exists, we write $S \not\sim S'$. Then, let

$$\tilde{p}_\mu(S'|S) = \begin{cases} \binom{m}{2}^{-1} q_\mu(\Delta, \Gamma) & S \sim_{\Delta,\Gamma} S' \\ 0 & S \not\sim S' \,, \end{cases} \tag{3}$$

with $q_\mu(\Delta, \Gamma)$ as defined in (2). To sample from $\tilde{p}_\mu(\cdot|S)$, it suffices to sample two uniformly random edges from E and perform a reshuffle. The prefactor $\binom{m}{2}^{-1}$ gives the probability that any two given edges are chosen.

The sequence $\{S_t\}$ is Markovian by construction. The following lemma and its corollary ensure that the sequence $\{H_t\} = \{g(S_t)\}$ is also a Markov chain.

LEMMA 2. Let $H, H' \in \mathcal{H}$. Suppose that $S_1, S_2 \in g^{-1}(H)$ and $S_1', S_2' \in g^{-1}(H')$. Then, $\tilde{p}_\mu(S_1'|S_1) = \tilde{p}_\mu(S_2'|S_2)$.

Proof. The objects S_1 and S_2 may each be considered arbitrary stub-labellings of part-edges in H. Similarly, S_1' and S_2' are each arbitrary labellings of part-edges in H'. However, by (3), $\tilde{p}_\mu(\cdot|S)$ depends only on the sizes of edges and their intersections in H, not their labels. □

COROLLARY 1. The process $\{H_t\} = \{g(S_t)\}$ on $\mathcal{H}_{\mathbf{d},\mathbf{k}}$ is a Markov chain.

Proof. Indeed, we can construct H_t mechanistically from H_{t-1} by choosing $S_{t-1} \in g^{-1}(H_{t-1})$, setting $S_t \sim \tilde{p}_\mu(\cdot|S_{t-1})$, and then letting $H_t = g(S_{t-1})$. Lemma 2 ensures that the distribution of H_t depends only on H_{t-1}, and not on the choices of S_{t-1} and S_t. □

THEOREM 1. The Markov chain $\{S_t\}$ on $\mathcal{S}_{\mathbf{d},\mathbf{k}}$ defined by the kernel $\tilde{p}_\mu$ is irreducible and reversible with respect to $\lambda_{\mathbf{d},\mathbf{k}}$, the uniform distribution on $\mathcal{S}_{\mathbf{d},\mathbf{k}}$. If in addition at least two entries of $\mathbf{k}$ are two or larger, $\{S_t\}$ is also aperiodic. In this case, $\lambda_{\mathbf{d},\mathbf{k}}$ is the equilibrium distribution of $\{S_t\}$. Furthermore, $\mu_{\mathbf{d},\mathbf{k}}$ is the equilibrium distribution of the process $\{H_t\} = \{g(S_t)\}$.

Proof. We will first show reversibility with respect to $\lambda_{\mathbf{d},\mathbf{k}}$. Fix S. Let $S \sim_{\Delta,\Gamma} S'$ and $S' \sim_{\Delta',\Gamma'} S$. In this case, we have $\Delta', \Gamma' = b(\Delta, \Gamma|S)$. Then,

$$\tilde{p}_\mu(S'|S) = \binom{m}{2}^{-1} q_\mu(\Delta, \Gamma) = \binom{m}{2}^{-1} q_\mu(\Delta', \Gamma') = \tilde{p}_\mu(S|S') \,,$$

as required. The second equality follows from Lemma 1, since $q(\Delta, \Gamma)$ depends only on $|\Delta|$, $|\Gamma|$, and $|\Delta \cap \Gamma|$, all of which are preserved by reshuffles.

Our proof approach for irreducibility generalizes that of [3]. We need to construct a path of non-zero probability between two arbitrary elements S_1 and S_2 of $\mathcal{S}_{\mathbf{d},\mathbf{k}}$. Let E_1 and E_2 be the edge-sets of S_1 and

S_2, respectively. We first describe a procedure for generating a new stub-labeled hypergraph S_3 such that $|E_2 \setminus E_3| < |E_2 \setminus E_1|$. Since $E_1 \neq E_2$ and $|E_1| = |E_2|$, we may pick $\Delta = \{\delta_1, \ldots, \delta_\ell\} \in E_2 \setminus E_1$. Since $\Delta \notin E_1$ and the edge dimension sequences must agree, there exists an edge $\Psi \in E_1 \setminus E_2$ such that $|\Psi| = |\Delta| = \ell$. Now, for each i, since $\Delta \notin E_1$, δ_i belongs to a different edge (call it Γ_i) in E_1. Note that we may have $\Gamma_i = \Gamma_{i'}$ for some $i' \neq i$ in case δ_i and $\delta_{i'}$ belong to the same hyperedge in E_1. Suppose we have $j \leq \ell$ such edges. Since δ_i is a stub, δ_i can belong to only one edge in each hypergraph, and therefore $\Gamma_k \notin E_2$ for each $k = 1, \ldots, j$. For each $k = 1, \ldots, j$, let $\left(\Psi_k, \Gamma_k'\right) = b_k(\Psi_{k-1}, \Gamma_k)$, where b_k assigns all elements of the set $\Delta \cap (\Psi_{k-1} \cup \Gamma_{k-1})$ to Ψ_k and uniformly distributes the remainder. Since $\Delta \subseteq \left(\bigcup_{k=1}^{j} \Gamma_k\right)$ by construction, by the end of this procedure we have $\Psi_j = \Delta$. Call the resulting stub-labeled hypergraph S_3 with edge set E_3. Since we have only modified the edges $\{\Gamma_k\}$ and Ψ, which are elements of $E_1 \setminus E_2$, we have not added any edges to the set $E_1 \setminus E_2$, but we have removed one, namely Ψ. We therefore have $|E_2 \setminus E_3| < |E_2 \setminus E_1|$, as desired. Applying this procedure inductively, we obtain a path of non-zero probability between S_1 and S_2, proving irreducibility.

To prove aperiodicity, we will construct supported cycles of length 2 and 3 in $\mathcal{S}$. Since the lengths of these cycles are relatively prime, aperiodicity will follow. To construct a cycle of length 2, pick two edges Δ and Γ and any valid reshuffle $b : (\Delta, \Gamma) \mapsto (\Delta', \Gamma')$. Then, $b^{-1} : (\Delta', \Gamma') \mapsto (\Delta, \Gamma)$ is also a valid reshuffle, and the sequence (b, b^{-1}) of transitions constitutes a supported cycle through $\mathcal{S}$ of length 2. To construct a cycle of length 3, choose two edges Δ and Γ which each contain two or more nodes, writing $\Delta = \{\delta_1, \delta_2, \ldots\}$ and $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$. This is always possible by hypothesis. Then, the following sequence of pairwise reshuffles constitutes a cycle of length 3:

$$\begin{aligned} \{\delta_1, \delta_2, \ldots\}, \{\gamma_1, \gamma_2, \ldots\} &\mapsto \{\gamma_1, \delta_2, \ldots\}, \{\delta_1, \gamma_2, \ldots\} \\ &\mapsto \{\gamma_2, \delta_2, \ldots\}, \{\delta_1, \gamma_1, \ldots\} \\ &\mapsto \{\delta_1, \delta_2, \ldots\}, \{\gamma_1, \gamma_2, \ldots\} . \end{aligned}$$

We have shown reversibility, irreducibility and aperiodicity, completing the proof. □

A small modification enables sampling from the vertex-labeled model $\eta_{\mathbf{d},\mathbf{k}}$. Let m_Δ give the number of edges parallel to edge Δ in hypergraph H, including Δ itself. Define an *acceptance rate*

$$a_\eta(S'|S) = \begin{cases} \frac{2^{|\Delta \cap \Gamma|}}{m_\Delta m_\Gamma} & S \sim_{\Delta,\Gamma} S' \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

THEOREM 2. Let $\tilde{p}_\eta(S'|S) = a(S'|S)\tilde{p}_\mu(S'|S)$. Let $\{S_t\}$ be the Markov chain generated by $\tilde{p}_\eta$. Then, the process $\{H_t\} = \{g(S_t)\}$ is a Markov chain. Furthermore, $\{H_t\}$ is irreducible and reversible with respect to $\eta_{\mathbf{d},\mathbf{k}}$. If in addition $\mathbf{k}$ has at least two entries larger than 2, $\{H_t\}$ is aperiodic. In this case, $\eta_{\mathbf{d},\mathbf{k}}$ is the equilibrium distribution of $\{H_t\}$.

Proof. Markovianity of $\{H_t\}$ follows from the same argument as Corollary 1. Irreducibility and aperiodicity follow from Theorem 1, since the state space $\mathcal{H}$ is a partition of $\mathcal{S}$ into equivalence classes induced by g. It remains to demonstrate reversibility with respect to $\eta_{\mathbf{d},\mathbf{k}}$. Let p_η be the transition kernel of H_t. Fix H and H'. Fix $S \in g^{-1}(H)$ and $S^* \in g^{-1}(H')$. Then, we can write

$$p_\eta(H'|H) = \sum_{S' \in g^{-1}(H')} \tilde{p}_\eta(S'|S) = \sum_{S' \in g^{-1}(H')} a(S'|S)\tilde{p}_\mu(S'|S) = a(S^*|S) \sum_{S' \in g^{-1}(H')} \tilde{p}_\mu(S'|S).$$

The expressions appearing in this calculation are independent of the specific choices of S or S^* following the same argument as in the proof of Lemma 2. We now evaluate the sum in the third line. The summand is nonzero if and only if $S \sim_{\Delta,\Gamma} S'$, in which case its value depends only on $|\Delta|$, $|\Gamma|$ and $|\Delta \cap \Gamma|$. We therefore count terms. There are $2^{|\Delta \cap \Gamma|}$ ways to arrange the intersection of Δ and Γ in $\mathcal{S}$, and $m_\Delta m_\Gamma$ ways to choose two edges parallel to Δ and Γ to reshuffle, all of which generate a distinct element of $g^{-1}(H')$. The sum therefore possesses precisely $a(S^*|S)^{-1}$ terms. We find that $p_\eta(H'|H) = \tilde{p}_\mu(S'|S)$ for any $S \in g^{-1}(H)$ and $S' \in g^{-1}(H')$. Reversibility of p_η thus follows from reversibility of $\tilde{p}_\mu$. □

Since the value of $a_\eta(S'|S)$ does not depend on the specific choices of S' and S, only H' and H, we can equivalently write $a_\eta(H'|H)$. For notational convenience, define an acceptance rate for the stub-labeled model, $a_\mu(H'|H) = 1$ for all S and S'. Algorithm 2 supplies pseudocode for sampling from the stub- and vertex-labeled hypergraph configuration models.

Algorithm 2: Markov Chain Monte Carlo for hypergraph configuration models

Input: degree sequence $\mathbf{d}$, edge dimension sequence $\mathbf{k}$, target distribution $\nu \in \{\mu_{\mathbf{d},\mathbf{k}}, \eta_{\mathbf{d},\mathbf{k}}\}$, initial hypergraph $H_0 \in \mathcal{H}_{\mathbf{d},\mathbf{k}}$, sample interval $h \in \mathbb{Z}_+$, desired sample size $s \in \mathbb{Z}_+$.
Initialization: $t \leftarrow 0$, $H \leftarrow H_0$
for $t = 1, 2, \ldots, sh$ **do**
  sample (Δ, Γ) uniformly at random from $\binom{E_t}{2}$
  $H' = b(\Delta, \Gamma|H_t)$
  if Uniform$([0,1]) \leq a_\nu(H'|H)$ **then**
    $H_{t+1} \leftarrow H'$ `// accept transition`
  else
    $H_{t+1} \leftarrow H_t$ `// reject transition`
  end
end
Output: $\{H_t$ such that $t|h\}$

Theorem 1 and 2 constitute a guarantee that, for sufficiently large sample intervals h, the hypergraphs sampled from Algorithm 2 will be asymptotically independent and asymptotically distributed according to the desired distribution. Unfortunately, we are unaware of any mixing-time bounds for this class of Markov chain. It is therefore possible in principle that the scaling in the mixing time as a function of system size is extremely poor, a result suggested by work on related classes of edge-swap Markov chains [43–45]. Our experience indicates, however, that sampling is possible for configuration models with many thousands of edges in practical time. For example, the `email-enron` data used here contains 10,887 edges. In the implemented code, it is possible to take 10^5 steps of stub-labeled MCMC in under 4 seconds on a single core of the author's personal laptop, and of vertex-labeled MCMC in under 30. The significantly larger timing for vertex-labeled MCMC is due (a) the computational cost of computing the acceptance probability and (b) the corresponding reduction in the number of successful transitions per loop of Algorithm 2. In addition to potential software optimizations, sampling tasks can usually also be parallelized, leading to shorter compute times when required.

4.1 *Connections to random bipartite graphs*

As briefly mentioned in Section 2, a hypergraph $H = (V, E)$ corresponds in a natural way to a bipartite dyadic graph B. The graph B consists of a node set $V \cup E$. An edge (u, e) exists between $u \in V$ and $e \in E$ iff $u \in e$ (in H). In this setting, the degree of u (in H) is equal to its degree in B, and the dimension of e (in H) is similarly equal to its degree in B. Let h be the function that assigns to each hypergraph its associated bipartite graph. When both nodes and edges are uniquely labeled, h is a bijection. It follows that a probability measure ν on the space $\mathcal{B}_{\mathbf{d},\mathbf{k}}$ of bipartite graphs with node degrees $\mathbf{d}$ and $\mathbf{k}$ induces a probability measure $\nu \circ h^{-1}$ on $\mathcal{H}_{\mathbf{d},\mathbf{k}}$. Several extant papers (e.g. [5, 46]) use this equivalence to construct random models of polyadic data. While it is sometimes thought that bipartite randomization supplies a complete solution to null hypergraph sampling, we show in this section that the natural scope of the bipartite method is limited to stub-labeled models.

We first define a bipartite, dyadic, configuration model. Let $\nu_{\mathbf{d},\mathbf{k}}$ be the measure on $\mathcal{B}_{\mathbf{d},\mathbf{k}}$ obtained by performing stub-matching with the node-set $V \cup E$, conditioned on the events that (a) all edges have the form (u, e) for $u \in V$ and $e \in E$, and (b) the bipartite graph is simple, without multi-edges or self-loops. Note that conditioning on the event that B is simple implies that the resulting stub-labeled and vertex-labeled models on bipartite models are identical. The work of Kannan *et al.* [41] considers the problem of sampling from $\nu_{\mathbf{d},\mathbf{k}}$ via bipartite edge-swaps. Such a swap maps $(u, e), (v, f) \mapsto (u, f), (v, e)$. By construction, this swap preserves $\mathbf{d}$ and $\mathbf{k}$. The authors show that a Markov chain which performs successive, random bipartite edge-swaps (while avoiding ones that would lead to a non-simple bipartite graph) is ergodic and therefore sufficient to sample from $\nu_{\mathbf{d},\mathbf{k}}$. Such a swap when viewed in the space $\mathcal{H}_{\mathbf{d},\mathbf{k}}$ amounts to swapping the edge memberships of nodes u and v. Importantly, a sequence of such swaps is special case of the pairwise reshuffle Markov chain on $\mathcal{S}_{\mathbf{d},\mathbf{k}}$. This implies following relationship:

PROPOSITION 1. The configuration model on simple bipartite graphs is equivalent to the stub-labeled hypergraph configuration model, in the sense that $\mu_{\mathbf{d},\mathbf{k}} = \nu_{\mathbf{d},\mathbf{k}} \circ h^{-1}$.

Proposition 1 makes precise the primary sense in which bipartite randomization provides an approach to random hypergraph modelling. This is a convenient result, since a dyadic edge-swap Markov chain on B can be used to produce samples from $\mu_{\mathbf{d},\mathbf{k}}$. This equivalence may also be used to give alternative proofs of Theorem 1. However, as discussed in [3], many data sets in which we aim to apply null modelling are better compared to vertex-labeled null distributions. Importantly, there is no obvious route for vertex-labeled sampling through bipartite random graphs. In particular, there is no analogue of Proposition 1 for this case. Thus, even though the work of [41] treats vertex-labeled sampling from $\mathcal{B}_{\mathbf{d},\mathbf{k}}$, this does not directly suffice for vertex-labeled sampling from $\mathcal{H}_{\mathbf{d},\mathbf{k}}$. The reason is that sampling from the vertex-labeled measure $\eta_{\mathbf{d},\mathbf{k}}$ requires adjusting for permutations of parallel hyperedges. When H contains multiple hyperedges of dimension three or greater, it is necessary to track multiple node-edge incidence relations in order to check when hyperedges are parallel.

It is possible to write down a version of Algorithm 2 for vertex-labeled sampling in which the fundamental data structure is a bipartite graph rather than a hypergraph. However, the result would not, to the author's knowledge, correspond to any standard random bipartite graph model. Expressing both models directly on the space $\mathcal{H}_{\mathbf{d},\mathbf{k}}$ of hypergraphs supports both conceptual clarity and a convenient formulation of MCMC for both stub- and vertex-labeled models. Incidentally, we note that this discussion highlights another, separate setting in which rigid adherence to dyadic methods can limit our data-analytical

horizons. An exclusive focus on nulls realizable through bipartite methods obscures the possibility of vertex-labeled polyadic models.

5. Network analysis with random hypergraphs

We now illustrate the application of hypergraph configuration models through three simple data-analytic vignettes. We first study triadic closure in polyadic networks, finding that the use of polyadic nulls can generate opposite findings when compared to dyadic nulls, and in doing so highlight distinctions between data sets that dyadic nulls miss. We then turn to degree-assortativity, defining and testing three distinct measures of association via polyadic data representations and randomizations. Finally, we study the tendency of edges to intersect on multiple vertices in the `email-Enron` data set, finding through simulation and analytical methods that large intersections occur at much higher rates than would be expected by random chance. Collectively, these cases illustrate the use of polyadic methods to define and analyse richer measures of network structure, and the use of polyadic nulls in contextualizing the results.

The data sets for case study were gathered, cleaned and generously made public by the authors of [14]. In certain experiments, data were temporally filtered in order to reduce their size; these cases have been explicitly noted in figure captions. The filtering procedure IS described in Appendix A. Importantly, in no case was the filtering operation motivated by the expense of Monte Carlo sampling from hypergraph configuration models; rather, the bottlenecks were (a) Monte Carlo sampling from projected dyadic graphs and (b) expensive graph computations such as triangle-counting.

5.1 *Triadic closure*

Triadic closure refers to the phenomenon that, in many networks, if two nodes u and v interact with a third node w, then it is statistically likely that u and v also interact with each other. Studies such as [5, 10, 47] observed triadic closure in many empirical networks and highlighted the fact that dyadic configuration models tend to be unable to reproduce this behaviour. Traditionally, triadic closure is measured by a ratio of the number of triangles (closed cycles on three nodes) that are present in the graph, compared to the number of 'wedges' (subgraphs on three nodes in which two edges are present).[1] Local and global variants of this ratio have been proposed. We follow the choice of [47] and work with the *average local clustering coefficient*. Let T_v denote the number of triangles incident on v, and W_v the number of wedges. Note that $W_v = \binom{d_v}{2}$. The average local clustering coefficient is

$$\bar{C} = \frac{1}{|N|} \sum_{v \in N} \frac{T_v}{W_v} \,. \tag{5}$$

It is direct to show that, in dyadic configuration models and under mild sparsity assumptions, $\bar{C}$ decays to zero as n grows large [47, 49].

The average local clustering coefficient $\bar{C}$ is a natively dyadic metric, in the sense that 'wedges' and 'triangles' are defined explicitly in terms of 2-edges. To compute $\bar{C}$ in polyadic data, it is therefore necessary to project a hypergraph down to a dyadic graph. In the context of hypothesis testing, there is some subtlety involved in the choice of when to do this. One method is to project first and then randomize via a dyadic null model. This is the most common historical approach, used for example in

[1] Recent measures have been developed for higher-order notions of clustering on larger subgraphs; see [48].

TABLE 1 *Average local clustering coefficients for selected data sets, compared to their expectations computed under vertex- and stub-labelling of hypergraph and projected graph models. Parentheses show standard deviations in the least-significant figure under the equilibrium distribution of each null model. Starred* data sets have been temporally filtered as described in Appendix A.*

		Hypergraph		Projected	
	$\bar{C}$	Vertex	Stub	Vertex	Stub
`congress-bills*`	0.608	0.601(1)	0.622(2)	0.451(2)	0.611(1)
`coauth-MAG-Geology*`	0.8200	0.8196(7)	0.8186(7)	0.00035(3)	0.00035(3)
`email-Enron`	0.658	0.825(3)	0.808(4)	0.638(5)	0.797(3)
`email-Eu*`	0.540	0.569(4)	0.601(4)	0.398(4)	0.598(4)
`tags-ask-ubuntu*`	0.571	0.609(4)	0.631(5)	0.183(4)	0.499(6)
`threads-math-sx*`	0.293	0.435(3)	0.426(3)	0.041(1)	0.093(2)

[5]. Alternatively, one may randomize via polyadic nulls prior to projection. This approach has the effect of preserving clustering induced by polyadic edges, since an edge of dimension k contains $3\binom{k}{3}$ wedges and $3\binom{k}{3}$ ordered triangles.

Table 1 summarizes a sequence of experiments performed on two collaboration networks (top) and four communication networks (bottom). For each network, we computed the observed local clustering coefficient $\bar{C}$ on the unweighted projected graph. We chose the unweighted projected graph, rather than the weighted projected graph, in order to more closely match previous analyses (such as those of [5]), as well as to avoid ambiguities that arise in the definition of $\bar{C}$ in the presence of multi-edges. We then compared the observed value to its null distribution under four randomizations. We first randomized using the vertex- and stub-labeled hypergraph configuration models, *prior* to projecting and measuring $\bar{C}$. These results are shown in the second and third columns. We then reversed the order, first computing the projected graph and randomizing via dyadic configuration models. The results are shown in the fourth and fifth columns.

Benchmarking against dyadic configuration models yields mixed results. Vertex-labeled configuration models conclude in all cases that the observed degree of clustering is significantly higher than would be expected by random chance. Stub-labeled benchmarking concludes that `congress-bills` and the two email data sets have significantly less clustering than expected, while the remainder have significantly more. The stub-labeled results should be approached with caution—for reasons discussed in detail in [3], the stub-labeled configuration model is a less-relevant comparison for these data sets than the vertex-labeled model.

Hypergraph randomization leads to different conclusions. First, the expected values of $\bar{C}$ under both hypergraph vertex- and stub-labeled nulls are much closer than under dyadic nulls, indicating that the polyadic statistical test is much less sensitive than the dyadic test to the choice of vertex- and stub-labelling. Second, the vertex-labeled null separates the two collaboration networks from the four communication networks. These two data sets are only slightly more clustered than expectation under the vertex-labeled model. Chebyshev's inequality implies that, at 95% confidence, `congress-bills` would be considered 'significantly more clustered' than its expectation under the vertex-labeled model, while the opposite conclusion would be reached under the stub-labeled model. On the other hand, `coauth-MAG-Geology` falls within two standard deviations of its expectation under each model; it would be necessary to inspect

the full sampling distribution to conduct a significance test at 95% confidence in this case. In contrast, Chebyshev's inequality implies that the four communication networks are all significantly *less* clustered than either vertex- or stub-labeled nulls would expect. Not only is there no clustering beyond that implied by the edge dimensions; triadic closure even appears to be inhibited in these data sets.

From a purely statistical perspective, these examples highlight the importance of careful null model selection in hypothesis testing for triadic closure, as differing statistical conclusions would be reached under different choices. More physically, the hypergraph nulls—and not the dyadic nulls—distinguish data sets by their generative mechanisms. Under the hypergraph nulls, the communication networks are all less clustered than expected, while the collaboration networks are approximately as clustered as expected. This result is to some extent intuitive. Collaborations between many agents often have nontrivial coordination costs that scale with the number of agents involved. It may be easier to assemble and coordinate a set of overlapping groups than a single large collective. In such cases, one may expect to observe clustering near or above that expected at random, since overlaps between related groups would generate triangles. In contrast, in digital communications it is essentially effort-free to construct interactions between larger groups of agents. Examples include adding an email address to the 'cc' field or introducing participants to thread on a forum. In such cases, triangles composed of distinct edges are energetically unnecessary, and may reflect redundant information flow. We therefore hypothesize that these systems have a tendency to absorb potential triangles into higher-dimensional interactions, resulting in lower levels of clustering than would be expected under polyadic nulls. A more detailed study of this hypothesis would require accounting for additional structure in these phenomena not captured by the hypergraph representation, including distinctions between bill authors and cosponsors; email senders and recipients; and similar *node roles* in polyadic interactions. Null models for hypergraph with distinguished roles, such as those provided by [50] and [51], may be of interest in this context.

5.2 *Degree-assortativity*

A network is degree-assortative when nodes of similar degree preferentially interact with each other. Early studies found that different categories of social, biological and technological networks display different patterns of assortative mixing by degree [52–54]. Social networks, for example, are frequently measured to be degree-assortative. In this context, degree-assortativity is often taken to indicate a tendency for popular or productive agents to interact with each other.

We measure degree-assortativity in hypergraphs via a generalization of the standard Spearman rank assortativity coefficient to hypergraphs. Importantly, there are multiple possible generalizations, each of which measures distinct structural information about intra-edge degree correlations. Let $E_{\geq 2} = \{\Delta \in E : |\Delta| \geq 2\}$. Let $h : E_{\geq 2} \to N^2$ be a (possibly random) *choice function* that assigns to each edge Δ two distinct nodes $u, v \in \Delta$. Three possibilities of interest are:

$$h(\Delta) = (u, v) \sim \text{Uniform}\binom{\Delta}{2} \tag{Uniform}$$

$$h(\Delta) = (u, v) = \underset{w,w' \in \binom{\Delta}{2}}{\text{argmax}}\, d_w d_{w'} \tag{Top-2}$$

$$h(\Delta) = (u, v) = \left(\underset{w\in\Delta}{\text{argmax}}\, d_w,\ \underset{w\in\Delta}{\text{argmin}}\, d_w\right) \tag{Top-Bottom}$$

he Uniform choice function selects two distinct nodes at random. The Top-2 choice function selects the two distinct nodes in the edge with largest degree. The Top-Bottom choice function selects the nodes with largest and smallest degree.

Let $r : N \to \mathbb{R}$ be a ranking function on the node set; we will always take $r(u)$ to be the rank of node u by degree in the hypergraph. For fixed h, let $f : E \to \mathbb{R}^2$ be defined componentwise by $f_j(\Delta) = (r \circ h_j)(\Delta)$. Then, the *generalized Spearman assortativity coefficient* is the empirical correlation coefficient between $f_1(\Delta)$ and $f_2(\Delta)$:

$$\rho_h = \frac{\sigma^2\left(f_1(\Delta), f_2(\Delta)\right)}{\sqrt{\sigma^2\left(f_1(\Delta), f_1(\Delta)\right)\sigma^2\left(f_2(\Delta), f_2(\Delta)\right)}}, \tag{6}$$

where $\sigma^2(X, Y) = \langle XY\rangle - \langle X\rangle\langle Y\rangle$ and brackets express averages over pairs of edges in E.

In the case of dyadic graphs, the three choice functions above are trivially identical, since there is only one way to pick two nodes from an edge of size two. On polyadic data, however, the resulting Spearman coefficients capture usefully different classes of information. For example, in studying coauthorship networks, they may be used to test hypotheses such as the following:

(1) **Generic assortativity**: On a given paper, most coauthors will simultaneously be more or less prolific than average.

(2) **Senior–senior assortativity**: The two most prolific authors on a paper will tend to be simultaneously more or less prolific than average.

(3) **Junior–senior assortativity**: The least prolific author on a paper will tend to be relatively more prolific if the most prolific author is relatively more prolific.

While the corresponding Spearman coefficients may in general be correlated, substantial variation manifests across study data sets. Figure 3 shows measurements and significance tests for one synthetic data set and the six empirical data sets studied in the previous section. The synthetic data consists of five copies of the hypergraph shown in Fig. 1. For each data set, we compute the dyadic assortativity coefficient on the projected graph (first column), as well as each of the three polyadic assortativity coefficients defined above.

The synthetic data (first row) illustrates a stark case in which dyadic hypothesis testing leads to a finding of statistically significant assortativity, while all three polyadic hypothesis tests find statistically significant *dis*assortativity. The choice between dyadic and polyadic nulls in this case would determine the sign of the effect size in an experiment that produced these data. In each of the empirical data sets, the dyadic and polyadic tests show qualitative agreement. However, the polyadic tests highlight several features of the data missed by the dyadic tests. In the two email data sets, all four coefficients are positive and to the right of the null distributions, though projecting (first column) increases the significance of the coefficients relative to hypergraph randomization (second column). The two forum data sets (`threads-math-sx` and `tags-ask-ubuntu`) are disassortative when compared to vertex-labeled nulls. The fact that `tags-ask-ubuntu` is statistically disassortative despite a positive uniform hypergraph Spearman coefficient speaks to the importance of carefully specified null hypothesis testing. Interestingly, the misspecified stub-labeled randomization would lead to the opposite finding. The coauthorship network `coauth-MAG-Geology` is highly assortative in all metrics—including the top-bottom measure, which is negative. The `congress-bills` data set is also assortative in all measures. Unlike the other data sets, the uniform hypergraph coefficient lies farther from the bulk of its null

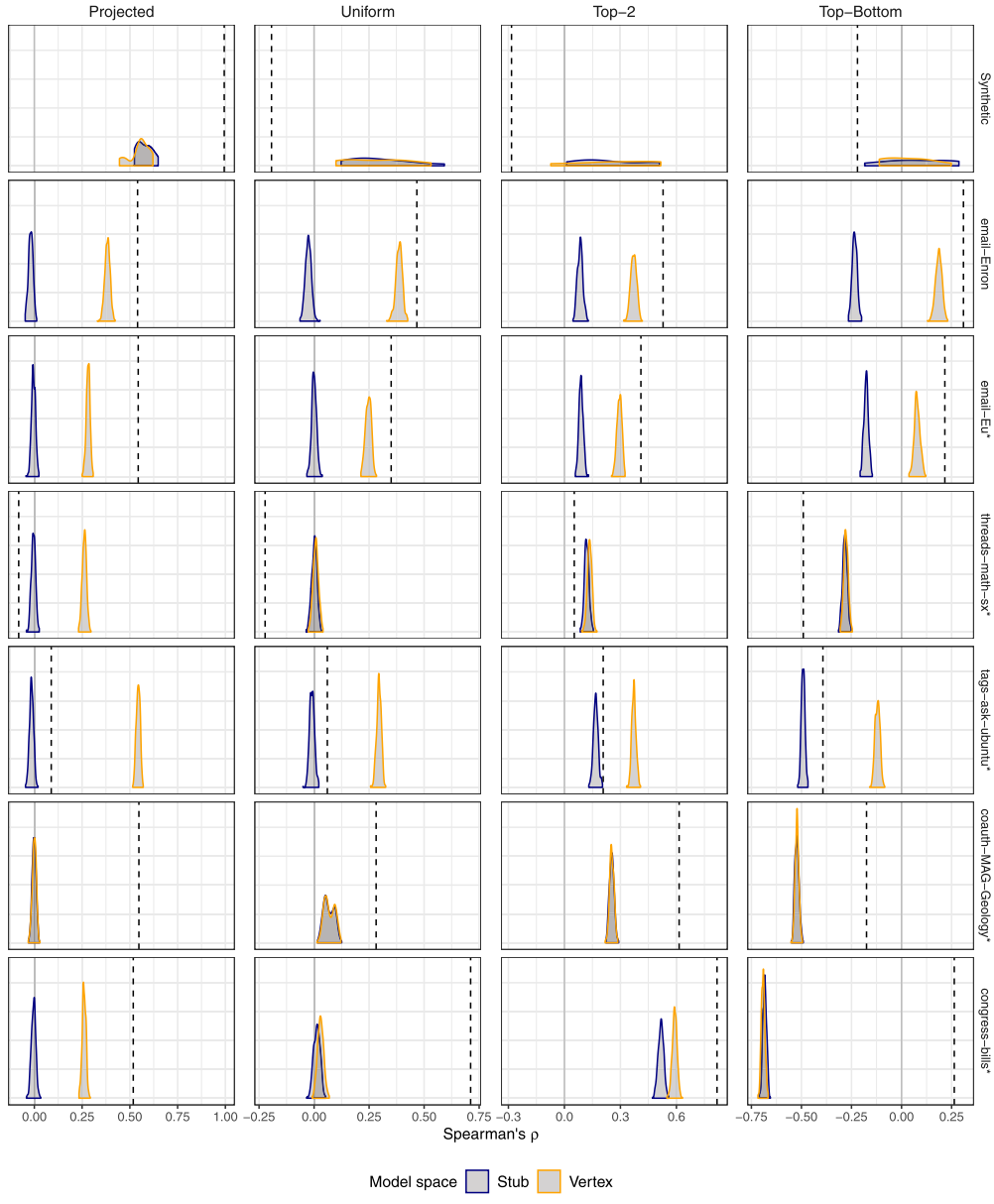

FIG. 3. Significance tests of degree-assortativity in synthetic and empirical networks. The synthetic data consists of five copies of the illustrative network shown in Fig. 1. In each figure, the dashed line gives the observed Spearman correlation, and densities give the null distributions under vertex- and stub-labeled configuration models. In the first column only, the hypergraph was projected down to an unweighted dyadic graph prior to randomization. Starred* data sets have been temporally filtered as described in the SI.

distribution than does the projected coefficient. We note in passing that, whereas the stub-labeled and vertex-labeled hypergraph distributions had similar expected rates of triadic closure (Table 1), their

distributions of degree-assortativity coefficients vary substantially, and would in some cases lead to qualitatively different study conclusions.

When studying triadic closure, we saw how hypergraph null models could lead us to contextualize standard graph metrics differently by delaying dyadic projection until after randomization. When studying assortativity, use of hypergraph nulls allows us to forgo the dyadic projection entirely, and thereby define a family of polyadic assortativity measures. Hypergraph null models thus enable us to measure and interpret novel structural patterns in polyadic data. We expand on this theme further in the next vignette.

5.3 *Hyperedge intersection profiles*

Let $\Delta, \Gamma \in H$. What is the size of their intersection? In the case of dyadic graphs, the intersection can have size at most two (when Δ is parallel to Γ). In hypergraphs, intersections of arbitrary sizes may occur. The existence of large intersections in a data set may indicate the emergence of polyadic social ties between groups of agents, or interpretable event sequences such as email threads or series of related scholarly papers. Several recent papers [14, 55] have studied similar questions by considering the rate at which 'holes' in the hypergraph tend to be 'filled in' by higher-order interactions. We take a simpler approach, defining a measure which is both easily computed and amenable to analytical approximation.

DEFINITION 12. (Intersection profile). For fixed $k, \ell \in \mathbb{Z}_+$, the *conditional intersection profile* of a hypergraph $H \in \mathcal{H}$ is the distribution

$$r_{k\ell}(j|H) = \langle \mathbb{I}(|\Delta \cap \Gamma| = j) \rangle_{k\ell} \,,$$

where $\langle \cdot \rangle_{k\ell}$ denotes the empirical average over all hyperedges Δ of size k and Γ of size ℓ. The *marginal intersection profile* is

$$r(j|H) = \langle \mathbb{I}(|\Delta \cap \Gamma| = j) \rangle \,,$$

with the average taken over all pairs of distinct edges in E.

Large values of $r_{k\ell}(j|H)$ indicate that edges of size k and ℓ frequently have intersections of size j in H. Empirical data sets may possess complex patterns of correlation between edges of various sizes. Evaluating whether an observed conditional or marginal intersection profile is noteworthy requires comparison to appropriately chosen null models.

Figure 4 demonstrates the use of hypergraph configuration models to study the intersection profile of the `email-Enron` data set. In Fig. 4(a), we compare the empirical average intersection size $\langle J \rangle_{k\ell} = \sum_{j=0}^{\infty} j r_{k\ell}(j|H)$ to its average $\langle \hat{J} \rangle_{k\ell}$ under the vertex-labeled configuration model. Higher values of the ratio $\frac{\langle J \rangle_{k\ell}}{\langle \hat{J} \rangle_{k\ell}}$ indicate the presence of denser intersections between edges of sizes k and ℓ. Notably, the empirical averages are not uniformly higher than the null model averages, even on the diagonal. There is apparent block structure, indicating that edges of certain sizes tend to correlate most strongly with certain other sizes. Edges of dimension 3 through 6 tend to interact strongly with each other, as do edges of dimension 7 and 8. However, edges in the smaller group interact more weakly with edges in the larger group than would be expected by chance. Further, more detailed study may be able to shed light on the groups of agents involved in these overlapping communications.

Figure 4(b) gives a global view using the marginal intersection profile. The observed profile (points in Fig. 4(b)) appears nearly linear on semilog axes through $j = 6$, suggesting that the decay in the intersection

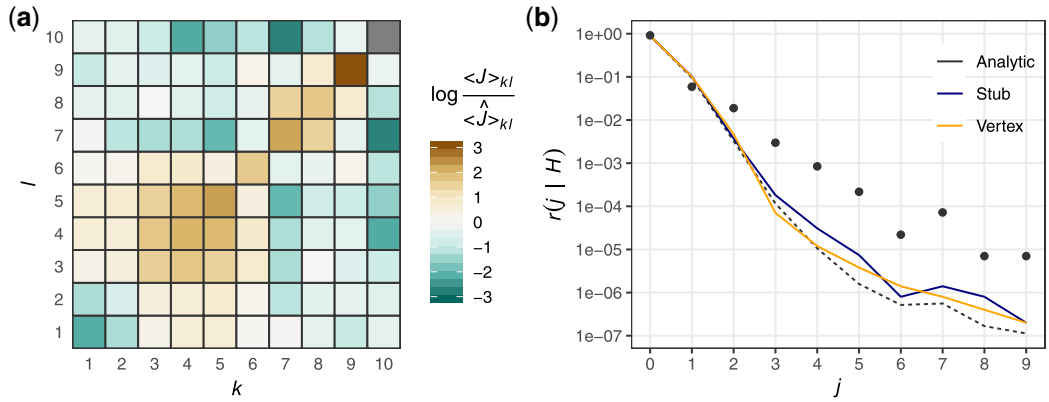

FIG. 4. Analysis of intersection profiles in the `email-Enron` data set. (a) The average of the intersection size normalized by the expectation $\langle \hat{J} \rangle_{k\ell}$ under the vertex-labeled configuration model. Positive values indicate that the data display larger intersections than expected under the configuration model for the corresponding values of k and ℓ. Colours are shown on a log scale. The missing value at $(k, \ell) = (10, 10)$ indicates that no non-empty intersections were observed between edges of these sizes in the Monte Carlo sampling runs. (b) Marginal intersection profile (points) of the empirical data, compared to null distributions under the stub- and vertex-labeled configuration models. The dashed line gives the analytic approximation of (7). Note the logarithmic vertical axis.

size is roughly exponential. In order to evaluate whether this behaviour indicates non-random clustering between edges, we compute the expectation $\hat{r}(j) = \mathbb{E}_\nu[r(j|H)]$ of the marginal intersection profile under a configuration model $\nu \in \{\mu_{\mathbf{d},\mathbf{k}}, \eta_{\mathbf{d},\mathbf{k}}\}$. This expectation quantifies the typical behaviour of a random comparable hypergraph. The solid lines in Fig. 4(b) give these expected profiles under both stub- and vertex-labeled models, which qualitatively agree. The observed data show fewer intersections on single vertices than would be expected by chance. On the other hand, for $j \geq 3$, $r(j|H)$ exceeds $\hat{r}(j)$ by an order of magnitude or more, suggesting substantial higher-order correlation in the data. These results likely reflect the passing of multiple messages between the same sets of users.

Some data sets may be too large to practically estimate $\hat{r}(j)$ by Monte Carlo methods. In such cases, it is possible to approximate $\hat{r}(j)$ under the stub-labeled configuration model analytically, using the following asymptotic result.

THEOREM 3. Fix ℓ, k and j. Let $\mathbf{D} \in \mathbb{Z}_+^n$ be a vector of i.i.d. copies of positive, discrete random variable $D \in \mathbb{Z}_+$ such that $D \leq d_{\max}$ with probability one for some $d_{\max}$. Let $\mathbf{K} \in \mathbb{Z}_+^m$ be any vector of edge dimensions configurable with $\mathbf{D}$. Let $H \sim \mu_{\mathbf{D},\mathbf{K}}$, and let Δ and Γ be uniformly random edges of H. Then, with high probability (w.h.p.) as n grows large,

$$\hat{r}_{k\ell}(j) = (1 + O(n^{-1}))j! \binom{k}{j} \binom{\ell}{j} \left(\frac{1}{n} \frac{\mathbb{E}[D^2] - \mathbb{E}[D]}{\mathbb{E}[D]^2} \right)^j. \tag{7}$$

Proof. Let $\langle d \rangle = \frac{1}{n} \sum_{u \in N} d_u$ denote the empirical mean degree of a given degree sequence $\mathbf{d}$. Assume without loss of generality that $\Delta = \{\delta_1, \ldots, \delta_k\}$ and $\Gamma = \{\gamma_1, \ldots, \gamma_\ell\}$ are the first two hyperedges formed by Algorithm 1, conditioned on non-degeneracy. There are $\binom{k}{j}$ ways to choose the j elements of Δ contained in $\Delta \cap \Gamma$, and similarly $\binom{\ell}{j}$ ways to choose the elements of Γ. There are then $j!$ ways to place these two sets in bijective correspondence. Define the event $A = \{\delta_h = \gamma_h,\ h = 1, \ldots, j\}$. Then,

$\hat{r}_{k\ell}(j) = j!\binom{k}{j}\binom{k}{\ell}\mu_{\mathbf{D},\mathbf{K}}(A)$. To compute $\mu_{\mathbf{D},\mathbf{K}}(A)$, we may explicitly enumerate

$$\mu_{\mathbf{D},\mathbf{K}}(A) = \sum_{u\in N} \frac{d_u}{n\langle d\rangle}\frac{d_u-1}{n\langle d\rangle - 1}\left[\sum_{v\in N\setminus\{u\}} \frac{d_v}{n\langle d\rangle - d_u}\frac{d_v-1}{n\langle d\rangle - d_u - 1}\left[\sum_{w\in N\setminus\{u,v\}} \cdots\right]\right],$$

with a total of j sums appearing. In each summation, the first factor gives the probability that $\delta_1 = u$ and the second that $\gamma_1 = u$. Consider the innermost summation, which may be written

$$S_R = \sum_{z\in N\setminus R} \frac{d_z}{n\langle d\rangle - \sum_{y\in R} d_y}\frac{d_z - 1}{n\langle d\rangle - \sum_{y\in R} d_y - 1} \tag{8}$$

for a set R of size $j-1$. Since $D \leq d_{\max}$ with probability one by hypothesis, we may employ Chebyshev's inequality to find that $(n\langle d\rangle)^{-1}\sum_{y\in R} d_y = O(n^{-1})$ w.h.p. We may therefore w.h.p. expand both factors within (8), obtaining the expression

$$\sum_{z\in N\setminus R} \frac{d_z(d_z-1)}{n^2\langle d\rangle^2}\left(1 + O(n^{-1})\right) .$$

Using Chebyshev's inequality again, we also obtain asymptotic behaviour on the other expressions appearing above. $\langle d\rangle = \left(1 + O\left(n^{-1}\right)\right)\mathbb{E}[D]$ and $\sum_{z\in N\setminus R}\frac{d_z(d_z-1)}{n} = \left(1 + O\left(n^{-1}\right)\right)(\mathbb{E}[D^2] - \mathbb{E}[D])$, both w.h.p. We have therefore shown that

$$S_R = \left(1 + O\left(n^{-1}\right)\right)\left(\frac{1}{n}\frac{\mathbb{E}[D^2] - \mathbb{E}[D]}{\mathbb{E}[D]^2}\right) \tag{9}$$

w.h.p. This argument may be repeated inductively for each of the remaining $j-1$ sums, each of which contributes the same factor appearing in (9), proving the theorem. □

Figure 4 shows the resulting approximation for $\hat{r}_{k\ell}$ as a dashed line, finding excellent qualitative agreement. In Figure 5, we demonstrate the use of Theorem 3 to estimate a null intersection profile in data sets of arbitrary size. The top set of panels shows four data sets in which the approximate null intersection profile consistently underestimates the rates of large intersections by several orders of magnitude, clearly indicating the presence of correlation structure over and above what would be expected under hypergraph randomization. The lower panels show four additional data sets in which the approximate null profile more closely approximates the observed data.

6. Discussion

Configuration models of random hypergraphs preserve the first moments of the data—the degree and edge-dimension sequences—while remaining maximally ignorant about additional data structure. These models extend the widely used configuration models for dyadic graphs, and serve as natural null models for polyadic network data analysis. We have demonstrated how to define, sample from, analyse and apply these models. We have seen that the choice between nulls can greatly impact the qualitative findings of studies of empirical polyadic data. The analyst faced with such a choice must therefore carefully consider whether dyadic simplification will lead to data representations and null spaces that are relevant for their

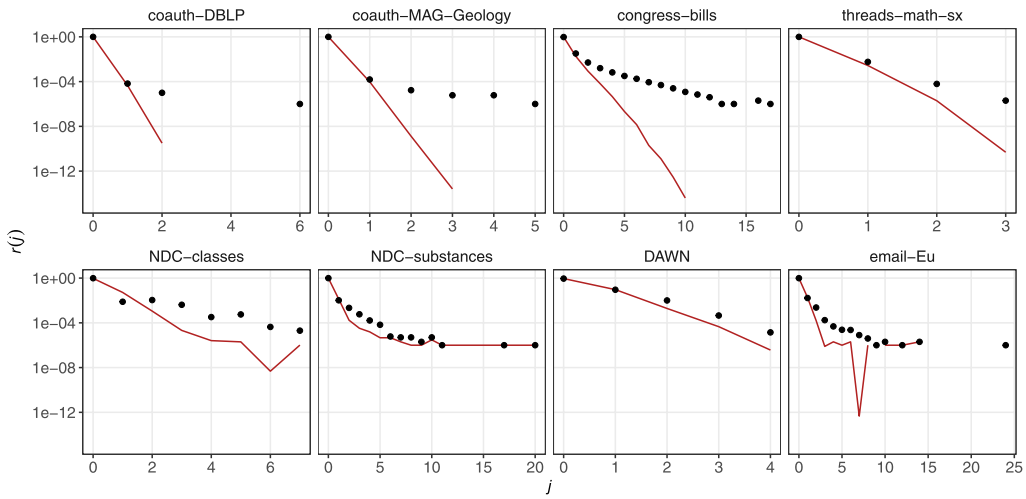

FIG. 5. Points give the observed intersection profile for six large polyadic data sets. The solid line gives the null intersection profile of Theorem 3. In this visualization, full data sets were used—the temporal filtering described in Appendix A was not performed.

application area. If the answer is negative, polyadic models such as those developed here should be employed. Second, employing polyadic nulls often allows the analyst to define novel measures that can illuminate higher-order structure in data. We have illustrated this with extended assortativity measures and intersection profiles, but many more extensions are possible. We hope that the introduction of statistically grounded hypergraph nulls will encourage analysts to design, measure and carefully interpret many novel techniques for networks with polyadic structure.

There are several directions of future work on configuration models of random hypergraphs. Beginning with theory, many classical asymptotic results on dyadic configuration models invite generalization. These include probabilistic characterization of component sizes; cycles and parallel edges; and the diameter of the connected component in various regimes. We also highlight two applications of potential interest. The first is motif analysis. A network motif is a subgraph that appears with higher-than-expected frequency in a given network [56], relative to a given null model. Considering the explicit dependence of this definition on the null, we conjecture that motif-discovery algorithms based on polyadic nulls may highlight importantly distinct structure when compared to dyadic nulls. A second promising application is in hypergraph clustering and community detection. A recent paper [25] offers a definition of modularity—a common quality function for network partitioning—based on a polyadic generalization of the Chung-Lu model [26]. In this case, the modularity of a given partition may be computed analytically. The same calculations used to prove Theorem 3 can also be used to show that the stub-labeled configuration model will give an asymptotically equivalent expression. However, for the large class of data sets more appropriately modelled by vertex-labeled nulls, other methods may be necessary. We anticipate that pursuing these tasks will pose interesting theoretical and computational challenges.

Funding

PSC acknowledges support from the National Science Foundation's Graduate Research Fellowship Program under award number 1122374.

Acknowledgements

I am grateful to Patrick Jaillet for several helpful discussions from which this work beneftited substantially. I also thank Mason Porter for helpfully emphasizing the possible role of node roles in polyadic interactions referenced at the end of the discussion of triadic closure.

Software

A `hypergraph` class, written in Python 3.5, is available at https://github.com/PhilChodrow/hypergraph.

This class includes implementations of Monte Carlo sampling for both stub- and vertex-labeled configuration models, as well as a simple tutorial illustrating the use of the software.

Additionally, a repository illustrating the analysis pipeline—including data acquisition, batch computations, and visualization script—is available at https://github.com/PhilChodrow/hypergraph_analysis.

This repository has considerably more moving parts, and is recommended only to those aiming to reproduce the results of this paper. Those who wish only to conduct novel analyses with hypergraph configuration models should refer to the first repository.

References

1. Bender, E. A. & Canfield, E. R. (1978) The asymptotic number of labeled graphs with given degree sequences. *J. Combin. Theory A*, **24**, 296–307.
2. Bollobás, B. (1980) A probabilistic proof of an asymptotic formula for the number of labeled regular graphs. *Eur. J. Combin.*, **1**, 311–316.
3. Fosdick, B. K., Larremore, D. B., Nishimura, J. & Ugander, J. (2018) Configuring random graph models with fixed degree sequences. *SIAM Rev.*, **60**, 315–355.
4. Molloy, M. & Reed, B. (1998) The size of the giant component of a random graph with a given degree sequence. *Combin., Prob. Comput.*, **7**, 295–305.
5. Newman, M. E. J., Strogatz, S. H. & Watts, D. J. (2001) Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, **64**, 17.
6. Mastrandrea, R., Fournet, J. & Barrat, A. (2015) Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLoS One*, **10**, 1–26.
7. Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J. F., Quaggiotto, M., van den Broeck, W., Régis, C., Lina, B. & Vanhems, P. (2011) High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS One*, **6**, 1–13.
8. Barabasi, A.-L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A. & Vicsek, T. (2002) Evolution of the social network of scientific collaborations. *Physica A*, **311**, 590–614.
9. Fowler, J. H. (2006) Legislative cosponsorship networks in the U.S. House and Senate. *Soc. Netw.*, **28**, 454–465.
10. Newman, M. E. J. (2001) Scientific collaboration networks. I. Network construction and fundamental results. *Phys. Rev. E*, **64**, 1–8.
11. Porter, M. A., Mucha, P. J., Newman, M. E. J. & Warmbrand, C. M. (2005) A network analysis of committees in the U. S. House of Representatives. *Proc. Natl. Acad. Sci. USA*, **102**, 7057–7062.
12. Klimt, B. & Yang, Y. (2004). The Enron Corpus: A new dataset for email classification research. Machine Learning: ECML 2004, (Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D. eds) Springer, Berlin, pp 217–226.
13. Youn, H., Strumsky, D., Bettencourt, L. M. A. & Lobo, J. (2015) Invention as a combinatorial process: evidence from US patents. *J. R. Soc. Interface*, **12**, 1–8.
14. Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A. & Kleinberg, J. (2018) Simplicial closure and higher-order link prediction. *Proc. Natl. Acad. Sci. USA*, **115**, 11221–11230.
15. Schaub, M. T., Benson, A. R., Horn, P., Lippner, G. & Jadbabaie, A. (2020) Random walks on simplicial complexes and the normalized Hodge Laplacian. SIAM Review, **62**, 353–391. pp. 1–36.

16. Giusti, C., Ghrist, R. & Bassett, D. S. (2016) Two's company, three (or more) is a simplex: algebraic-topological tools for understanding higher-order structure in neural data. *J. Comput. Neurosci.*, **41**, 1–14.
17. Grilli, J., Barabás, G., Michalska-Smith, M. J. & Allesina, S. (2017) Higher-order interactions stabilize dynamics in competitive network models. *Nature*, 210–213.
18. Benson, A. R., Gleich, D. F. & Leskovec, J. (2016) Higher-order organization of complex networks. *Science*, **353**, 163–166.
19. Ugander, J., Backstrom, L., Marlow, C. & Kleinberg, J. (2012) Structural diversity in social contagion. *Proc. Natl. Acad. Sci. USA*, **109**, 5962–5966.
20. Nishimura, J. (2018) The connectivity of graphs of graphs with self-loops and a given degree sequence. *J. Compl. Netw.*, **6**, 927–947.
21. Angel, O., van der Hofstad, R. & Holmgren, C. (2019) Limit laws for self-loops and multiple edges in the configuration model. In Annales de l'Institut Henri Poincaré, Probabilités et Statistiques, pp. 1509–1530.
22. Molloy, M. & Reed, B. (1995) A critical point for random graphs with a given degree sequence. *Random Struct. & Algorithms*, **6**, 161–180.
23. Ghoshal, G., Zlatić, V., Caldarelli, G. & Newman, M. E. J. (2009) Random hypergraphs and their applications. *Phys. Rev. E.*, **79**, 066118.
24. Kumar, T., Vaidyanathan, S., Ananthapadmanabhan, H., Parthasarathy, S. & Ravindran, B. (2019) Hypergraph clustering: a modularity maximization approach. *PloS One*, **14**.
25. Kaminski, B., Poulin, V., Pralat, P., Szufel, P. & Theberge, F. (2019) Clustering via hypergraph modularity. *PloS ONE*, **14**, 1–17.
26. Chung, F. & Lu, L. (2002) The average distances in random graphs with given expected degrees. *Proc. Natl. Acad. Sci. USA*, **99**, 15879–15882.
27. Carlsson, G. (2009) Topology and data. *Bull. Am. Math. Soc.*, **46**, 255–308.
28. Courtney, O. T. & Bianconi, G. (2016) Generalized network structures: the configuration model and the canonical ensemble of simplicial complexes. *Phys. Rev. E*, **93**, 1–26.
29. Young, J. G., Petri, G., Vaccarino, F. & Patania, A. (2017) Construction of and efficient sampling from the simplicial configuration model. *Phys. Rev. E*, **96**, 1–6.
30. Gale, D. (1957) A theorem on flows in networks. *Pac. J. Math.*, **7**, 1073–1082.
31. Ryser, H. J. (1960) Matrices of zeros and ones. *Bull. Am. Math. Soc.*, **66**, 442–464.
32. Artzy-Randrup, Y. & Stone, L. (2005) Generating uniformly distributed random networks. *Phys. Rev. E*, **72**, 056708.
33. Verhelst, N. D. (2008) An efficient MCMC algorithm to sample binary matrices with fixed marginals. *Psychometrika*, **73**, 705.
34. Jerrum, M. & Sinclair, A. (1990) Fast uniform generation of regular graphs. *Theor. Comput. Sci.*, **73**, 91–100.
35. McKay, B. D. & Wormald, N. C. (1990) Uniform generation of random regular graphs of moderate degree. *J. Algorithms*, **11**, 52–67.
36. Viger, F. & Latapy, M. (2005) Efficient and simple generation of random simple connected graphs with prescribed degree sequence. *International Computing and Combinatorics Conference*. Springer, New York, New York, pp. 440–449.
37. Blitzstein, J. & Diaconis, P. (2011) A sequential importance sampling algorithm for generating random graphs with prescribed degrees. *Internet Math.*, **6**, 489–522.
38. Carstens, C. J. (2015) Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast curveball algorithm. *Phys. Rev. E*, **91**, 042812.
39. Del Genio, C. I., Kim, H., Toroczkai, Z. & Bassler, K. E. (2010) Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *PLoS One*, **5**, e10012.
40. Strona, G., Nappo, D., Boccacci, F., Fattorini, S. & San-Miguel-Ayanz, J. (2014) A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nat. Commun.*, **5**, 4114.
41. Kannan, R., Tetali, P. & Vempala, S. (1999) Simple Markov-chain algorithms for generating bipartite graphs and tournaments. *Random Struct. Algorithms*, **14**, 293–308.

42. Amanatidis, G., Green, B. & Mihail, M. (2015) Graphic realizations of joint-degree matrices. *arXiv:1509.07076*, pp. 1–18.
43. Erdős, P. L., Greenhill, C., Mezei, T. R., Miklós, I., Soltész, D. & Soukup, L. (2019) The mixing time of the swap (switch) Markov chains: a unified approach. *arXiv:1903.06600*.
44. Greenhill, C. (2011) A polynomial bound on the mixing time of a Markov chain for sampling regular directed graphs. *Electron. J. Combin.*, **18**, 234.
45. Greenhill, C. (2014) The switch Markov chain for sampling irregular graphs. *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, pp. 1564–1572.
46. Saracco, F., Di Clemente, R., Gabrielli, A. & Squartini, T. (2015) Randomizing bipartite networks: the case of the World Trade Web. *Sci. Rep.*, **5**, 1–18.
47. Strogatz, S. H. & Watts, D. J. (1998) Collective dynamics of "small-world" networks. *Nature*, **393**, 440–442.
48. Yin, H., Benson, A. R. & Leskovec, J. (2018) Higher-order clustering in networks. *Phys. Rev. E*, **97**, 052306.
49. Newman, M. E. J. (2010) *Networks: An Introduction*. Oxford, UK: Oxford University Press.
50. Allard, A., Hébert-Dufresne, L., Young, J.-G. & Dubé, L. J. (2015) General and exact approach to percolation on random graphs. *Phys. Rev. E*, **92**, 062807.
51. Chodrow, P. & Mellor, A. (2020) Annotated hypergraphs: models and applications. *Appl. Netw. Sci.*, **5**, 9.
52. Colizza, V., Flammini, A., Serrano, M. A. & Vespignani, A. (2006) Detecting rich-club ordering in complex networks. *Nat. Phys.*, **2**, 110–115.
53. Newman, M. E. J. (2002) Assortative mixing in networks. *Phys. Rev. Lett.*, **89**, 1–5.
54. Newman, M. E. J. (2003) Mixing patterns in networks. *Phys. Rev. E*, **67**, 13.
55. Patania, A., Petri, G. & Vaccarino, F. (2017) The shape of collaborations. *EPJ Data Sci.*, **6**, 1–16.
56. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. & Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Science*, **298**, 824–827.
57. Fowler, J. H. (2006) Connecting the congress: a study of cosponsorship networks. *Polit. Anal.*, **14**, 456–487.
58. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-J. P. & Wang, K. (2015) An overview of Microsoft Academic Service (MAS) and applications. *Proceedings of the 24th International Conference on World Wide Web*. ACM Press, New York, New York.

A. Data

The data sets used in this article were prepared by the authors of [14] and accessed from https://www.cs.cornell.edu/~arb/data/. Some data sets have been filtered to exclude edges prior to a temporal threshold τ in order to promote practical compute times on triangle counting and mixing of vertex-labeled models in projected dyadic graph spaces. Notably, in no cases was sampling from hypergraph configuration models the computational bottleneck. Thresholds were chosen to construct data with edge sets of approximate size $m \approx 10^4$ but are otherwise arbitrary. Temporal data subsets were used in the generation of Table 1 and Fig. 3. Table A.1 gives the node and edge counts of both the original data and the data after temporal subsetting when applicable.

TABLE A.1 *Summary of data preparation. When τ is given, the filtered data set consists in all edges that occurred after time τ.*

	Original			Filtered	
	n	m	τ	n	m
`email-Enron`	143	10,886	–	–	–
`email-Eu`	1,006	235,264	1.105×10^9	817	32,117
`congress-bills` [9, 57]	1,719	260,852	7.315×10^5	537	6,661
`coauth-MAG-Geology` [58]	1,261,130	1,591,167	2017	73,436	23,434
`threads-math-sx`	201,864	719,793	2.19×10^{12}	11,880	22,786
`tags-ask-ubuntu`	200,975	192,948	2.6×10^{12}	2,120	19,338