Random graphs with node and block effects: models, goodness-of-fit tests, and applications to biological networks

Elizabeth Gross, Sonja Petrović, Despina Stasi

Abstract: Many popular models from the networks literature can be viewed through a common lens. We describe it here and call the class of models log-linear ERGMs. It includes degree-based models, stochastic blockmodels, and combinations of these. Given the interest in combined node and block effects in network formation mechanisms, we introduce a general directed relative of the degree-corrected stochastic blockmodel: an exponential family model we call $p_1$-SBM. It is a generalization of several well-known variants of the blockmodel.

We study the problem of testing model fit for the log-linear ERGM class. The model fitting approach we take, through the use of quick estimation algorithms borrowed from the contingency table literature and effective sampling methods rooted in graph theory and algebraic statistics, results in an exact test whose $p$-value can be approximated efficiently in networks of moderate sizes.

We showcase the performance of the method on two data sets from biology: the connectome of *C. elegans* and the interactome of *Arabidopsis thaliana*. These two networks, a neuronal network and a protein-protein interaction network, have been popular examples in the network science literature, but a model-based approach to studying them has been missing thus far.

1. Introduction

Data in the form of graphs are common in many biological contexts. In systems biology, graphs are used to record protein-protein interactions. For example, in [Con11], the authors construct a network from 5664 experimentally observed interactions between 2661 proteins from *Arabidopsis thaliana*, a small flowering plant used as a model organism. In neuroscience, graphs are used to record synaptic contacts between neurons, as in the gap junction and chemical synapse networks constructed for *C. elegans* in [WS98]; see also a recent study of these networks in [VCP+11]. Both protein-protein interaction networks and neuronal networks have been used as examples of scale-free networks ([JTA+00], [Tan05], [WF01], [Alb05], [CSN09], [BBB+16]), that is, networks whose degree distribution follows a power law, suggesting degree-based edge formation mechanisms in these data. Although descriptive statistics of these networks have been studied, somewhat surprisingly, they have yet to be rigorously analyzed using a model-fitting approach. Part of the difficulty in studying such networks within a model-based setting is that statistical theory regarding fitting random graph models is still in its infancy, since it poses several challenging combinatorial and algorithmic problems.

The present manuscript adds a chapter to the 30+-year-long story on random graph models and the difficult question of testing model fit. Even for exponential family random graph models (ERGMs), the most studied class of network models, quantitative methods for goodness-of-fit testing are still lacking; see [Kol17, §2.3.4]. However, a peek back at the literature over the last three decades

---

1Authors in alphabetical order, contribution is equal. The initial phase of this research was supported by the U.S. Air Force Office of Scientific Research Grant #FA9550-14-1-0141 to Illinois Institute of Technology. EG was supported by the National Science Foundation under grant DMS-1620109; SP and DS under grant NSF DMS-1522662. The second and third author are grateful to the Isaac Newton Institute for Mathematical Sciences at Cambridge for its hospitality during their visit to the 2016 Theoretical Foundations for Statistical Network Analysis Program supported by EPSRC grant #EP/K032208/1.
suggest that implementable goodness-of-fit tests are within reach if we restrict to ERGMs that can be interpreted as log-linear models on contingency tables.

The connection between random graph models and log-linear models was made concretely in [FW81b], a comment on [HL81] that introduced the $p_1$ model, a degree-based model for directed networks. This comment paper re-interpreted the model in log-linear form by organizing the network data into a four-dimensional table; this drew a natural connection to contingency tables. The authors of [FW81b] proved the equivalence of the network and table models, linking random graphs to the exciting developments in categorical data analysis that were taking place at that time and, consequently, to the fitting methods developed over the years and discussed below. The connections continued—during the same decade, the authors of [FMW85] introduced the first variants of the stochastic blockmodel [HLL83] for directed networks, also log-linear in form, while more recently, the paper [YJFL19] discusses the asymptotic theory of the directed network model with covariates, a model whose log-linear interpretation is given in [FR12].

In the decades that followed [FW81b], the authors of [WP96], [PW99], and [RPW99] introduced the more general class of ERGMs that included dyad dependent models, i.e. models without a natural log-linear interpretation; see also Markov graphs [FS86]. Since then, much of the network modeling literature has focused on implementing goodness-of-fit algorithms for specific models or families of models. For example, [Lei16] provides an asymptotic test for the basic variant stochastic blockmodel, along with a non-asymptotic version, while [BM] offers a hypothesis test for the stochastic blockmodel vs. the Erdős-Rényi model, with optimality guarantees under certain degree growth conditions. For general ERGMs, the question of testing model goodness of fit has been generally acknowledged as difficult. It took until 2008 for the first comprehensive, albeit heuristic, approach to appear in [HGH08] along with an R package for a handful of popular network models; see also [CKHG15]. The approach in [HGH08] is for general ERGMs, however, the method is graphical, relying on the user to visually assess plots of local summary statistics.

In the case where the ERGM belongs to the class of log-linear models, the goodness-of-fit question has been theoretically answered [DS98], even if methods for effective implementation have remained elusive. Specifically, for these ERGMs, there are two main approaches to goodness-of-fit testing based on running a finite-sample approximation of an exact conditional test. The first uses Markov bases and a Monte Carlo Markov chain algorithm [DS98], while the second relies on sequential importance sampling [ZC13], [BD10] to sample graphs with the prescribed network statistics, for example, the degree sequence. The former approach, modified for network applications in [OHT13] and [PRF10], is guaranteed to work in all log-linear settings, while the latter approach is limited to a small number of log-linear models, but is computationally more effective. In this manuscript, our main focus is on the effective use of the first method, the method of Markov bases, for log-linear ERGMs.

While the method of Markov bases is theoretically sound, it does exhibit difficulties in implementation and practice as pointed out by [ZC13], namely the computational complexity of obtaining a full Markov basis and the limited results on mixing times. However, since the publication of [ZC13], there has been work to address both of these issues. Indeed, dynamic approaches to generating Markov bases elements have recently been explored to remedy the computational strain of computing full bases [Dob12], [GPS16], [OHT13]. For example, the second work gives a combinatorial method for goodness-of-fit testing for the $p_1$ random directed graph model [HL81] with dyad-dependent reciprocation. The exact conditional test implemented therein relies on the Markov bases methodology, but avoids the usual computational bottleneck by constructing Markov moves based on the current state of the chain, rather than computing the full basis up front.

In this manuscript, we significantly extend the approach of [GPS16], describing and implementing goodness-of-fit algorithms for several models beyond the $p_1$-model with dyad-dependent (called edge-dependent therein) reciprocation effect. While the methodology applies to any log-linear ERGM, we
specifically focus on the $\beta$-model (see [CDS11]), the $p_1$-model with three different reciprocation effects [HL81, FW81b], block versions of these (see [Yan15] for the $\beta$-blockmodel and [FMW85] for the first $p_1$-blockmodel variants), as well as versions with structural zeros [BF07, Section 5.1]. In addition, we generalize the $p_1$-blockmodels from [FMW85] in this manuscript. These models can be viewed as the directed version of the degree-corrected stochastic block model with additional possible reciprocation effects.

In regards to the mixing time of the proposed algorithms in this manuscript, we note that [Dil16] recently showed that the dynamic Markov bases algorithm from [GPS16] is rapidly mixing on many fibers, since it contains the simple switch chain well-known in the graph theory literature [KTV99a]. Although herein we propose many more chains than [GPS16], most of these chains also contain the simple switch chain, which provides ample evidence that the algorithms will mix rapidly on most fibers, even if a mixing time analysis is outside the scope of this paper.

The remainder of the manuscript is organized as follows. In Section 2, we introduce the models we study in detail, along with a historical perspective on their connection to log-linear models. In particular, a new model family for directed networks that combines block and node effects, which we call $p_1$-SBM, is proposed in Section 2.2. Section 3 gives an overview of the exact test we develop and goes into further detail about test validity and some algorithmic considerations. Our method is illustrated on two experimental datasets, a neuronal network [VCP+11] and a protein-protein interaction network [Con11]; both networks have interested applied researchers for over a decade. The experimental results in Section 4 show that none of the variants of the $p_1$ fit the directed part of the neuronal data, meaning that the edge formation in these neuronal networks is not mainly driven by the attractiveness and expansiveness of nodes (neurons). But a run of edge-dependent $p_1$ for the union of directed and undirected neuronal network gives $p$-value $> 0.3$. Interestingly, computations also show that the $p_1$-model with structural zeros fits the protein-protein network pretty well; to determine whether this fit is an artifact of the data collection process remains to be addressed in further work. The concluding Section 5 discusses the context of our results and outlines some open problems, both of theoretical and algorithmic nature.

2. The models: log-linear ERGMs

Let $\mathcal{G}_n$ be a class of graphs on $n$ vertices and let $\phi : \mathcal{G}_n \to \mathbb{R}^\ell$ be an embedding such that for all $G = (V, E)$ in $\mathcal{G}_n$ we have $\phi(G) = \sum_{e \in E} \phi(e)$. We will implicitly use the embedding $\phi$ to represent $G$ as a vector. For example, for the class of simple directed graphs, one reasonable choice of $\phi$ would embed $\mathcal{G}_n$ into $\mathbb{R}^{n^2}$ by representing $G$ as its vectorized adjacency matrix, while, for the class of simple undirected graphs, for which the adjacency matrix is symmetric with all-zero diagonal entries, $\mathbb{R}^{\binom{n}{2}}$ would be sufficient. For directed graphs, another suitable embedding is discussed in Section 2.1 and maps $\mathcal{G}_n$ into $\mathbb{R}^{n \times n \times 2 \times 2}$ by representing graphs by their vectorized $n \times n \times 2 \times 2$ Fienberg-Wasserman table. For the rest of this section, we will treat graphs as vectors, without explicitly referring to the embedding $\phi$.

An exponential family random graph model, or an ERGM for short, is a collection of probability distributions on $\mathcal{G}_n$ that places the following probability on each graph $G \in \mathcal{G}_n$:

$$P_\theta(G) = Z(\theta)e^{\theta \cdot t(G)},$$

where $G$ is uniquely represented as a vector in $\mathbb{R}^\ell$, $\theta$ is a row vector of parameters of length $q$, the map $t: \mathbb{R}^\ell \to \mathbb{R}^q$ computes the sufficient statistics, and $Z(\theta)$ is a normalizing constant. The image of the sufficient statistic map $t$ is a vector where each entry is a network statistic used to specify the
model, such as edge count, degree of a given vertex, number of edges in a given block of vertices, etc. For more on exponential random graph models see [AWC99], [New10, Section 15.2], [RPKL07].

**Definition 2.1.** We call such a model a **log-linear ERGM** if the sufficient statistic map \( t \) in the ERGM specification (1) is a linear map \( t : \mathbb{R}^\ell \to \mathbb{R}^q \) from the space of graphs to the space of the minimal sufficient statistics of the model.

The term *log-linear* in Definition 2.1 is not chosen arbitrarily. Indeed, it is chosen to reflect the connection to categorical data analysis. An ERGM is log-linear exactly when there exists a contingency table representation of \( G \) such that the sufficient statistic \( t \) corresponds to a set of table marginals. The requirement that \( G \) be a simple graph translates to restricting the sample space to 0/1 tables. The connection to contingency tables is crucial as it allows for the easy transfer of fitting and testing methods for contingency tables to graphs. Several random graph models have been studied from this point of view, including the \( \beta \)-model [RPF13] and some basic variants of the stochastic blockmodel from [FMW85].

Under the assumption of linearity of \( t \) and for a vector representation of \( G \) that satisfies \( G = \sum_{e \in G} e \), one obtains:

\[
es_\theta t(G) = e^{\theta t(\sum_{e \in G} e)} = e^{\theta \sum_{e \in G} t(e)}.
\]

In other words, dyadic independence is implied by the linearity of the sufficient statistics map \( t \).

We use the exponential family form to define the models we study in this paper, by defining the relevant linear function \( t \) for a family of distributions over \( G_n \). The reader should note, however, that it is more common to define these models through their dyadic probabilities. Since \( t \) is linear for these models, and by the preceding discussion, the two definitions lead to equivalent models.

While dyadic independence may, at first, seem restrictive, the class includes degree-based models, e.g. the \( \beta \) and \( p_1 \) models, and edge-count-based models, e.g. the standard blockmodels. It also includes degree-based extensions of block-models, as well as extensions that include covariates [YJFL19]. Furthermore, by focusing on models with dyadic independence, we avoid issues that have been raised in recent years for ERGMs with dyadic dependence. For example, ERGMs that include \( k \)-stars or triangle counts in the sufficient statistics have been shown to exhibit model degeneracy ([CD13], [HRS+03], [YJFL19, §1.1]), and dyadic dependence results in nonprojectability, a troublesome property for maximum likelihood estimation, as detailed in [SR13].

In this section, we review the log-linear ERGMs that we will focus on in detail in this manuscript. First, in Section 2.1, we will describe the degree-based models, and then, in Section 2.2, we will discuss blockmodels and variants.

### 2.1. Degree-based models

Let us briefly recall two familiar models for undirected and directed graphs whose sufficient statistics capture node degrees. Note that while degree-based models have limitations in modeling global or triadic effects in general, the class of degree-based models is already quite diverse and challenging to study. The generally difficult problem of sampling graphs with a fixed statistic has a better solution in cases when the statistic is degree-based, due to its the history of sampling graphs with fixed node degrees in graph theory literature.

**The \( \beta \)-model.** The sufficient statistic for the \( \beta \)-model for undirected graphs is the degree sequence

\[
t(G) = (d_1, \ldots, d_n),
\]

where \( d_i \) counts the number of edges incident to vertex \( i \). Associated to each vertex \( i \) there is a parameter \( \beta_i \) that controls for the propensity of vertex \( i \) to form edges. In exponential family form, the model is
\[ P_{\beta_1, \beta_2, \ldots, \beta_n}(G) = Z(\beta_1, \beta_2, \ldots, \beta_n)e^{d_1 \beta_1 + \cdots + d_n \beta_n}, \quad \beta_i \in \mathbb{R}. \]

Dyadic independence leads to an alternative specification of the model, using the probabilities \( p_{i,j} \) of each dyad \( i,j \) being connected in \( g \) or, equivalently, their log-odds:

\[
p_{i,j} = \frac{e^{\beta_i + \beta_j}}{1 + e^{\beta_i + \beta_j}} \quad \text{equiv.,} \quad \log \left( \frac{p_{i,j}}{1 - p_{i,j}} \right) = \beta_i + \beta_j, \quad i, j \in \{1, 2, \ldots, n\}.
\]

**The \( p_1 \)-model.** The \( p_1 \)-model for directed graphs, introduced by Holland and Leinhardt [HL81] and extended by Fienberg and Wasserman [FW81b], has three variants that capture different reciprocation effects: zero reciprocation, constant reciprocation, and dyad-specific reciprocation (to which they referred as differential reciprocity). To define them, let us describe reciprocation effects: zero reciprocation, constant reciprocation, and dyad-specific reciprocation (to which they referred as differential reciprocity). To define them, let us describe reciprocation effects: zero reciprocation, constant reciprocation, and dyad-specific reciprocation (to which they referred as differential reciprocity).

Here the parameters \( \alpha \) and \( \beta \) record the rates at which the node \( i \) sends and receives links, respectively; for a more thorough discussion see [HL81]. The sufficient statistic for the \( p_1 \)-model with constant reciprocation consists of the bidegree sequence and the reciprocated edge count \( m \):

\[
t(G) = (d_1^{\text{in}}, d_2^{\text{in}}, d_1^{\text{out}}, \ldots, d_n^{\text{out}}, m) = (d_1^{\text{in}}, d_2^{\text{out}}, m)
\]

and the model parameter vector is specified as \( \theta = (\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n) \). The additional parameter \( \rho \) is the reciprocation parameter identifying the overall propensity of nodes to reciprocate links, which is assumed to be constant across the entire network. The sufficient statistic for the \( p_1 \)-model with dyad-specific reciprocation consists of the reciprocated degree sequence and the number of reciprocated edges \( m_i \) incident to each vertex:

\[
t(G) = (d_1^{\text{in}}, d_2^{\text{out}}, m) = (d_1^{\text{in}}, d_2^{\text{out}}, d_3^{\text{out}}, m_1, \ldots, m_n)
\]

with model parameter vector \( \theta = (\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \rho_1, \ldots, \rho_n) \). Here, each parameter \( \rho_i \) encodes the rate at which node \( i \) is likely to reciprocate links, thus allowing one to record different reciprocation effects for each dyad.

Since each dyad can be in 1 of 4 configurations: no link, directed from \( i \) to \( j \), directed from \( j \) to \( i \), or reciprocated, we say that a dyad \((i,j)\) can be in one of the following states: \((0,0)\), \((1,0)\), \((0,1)\), or \((1,1)\), respectively. Following [HL81, FW81b], we denote \( p_{ijkl} \) the probability of the dyad \((i,j)\) to be in state \((k,l)\). The dyadic probabilities in the \( p_1 \)-model are then specified as follows:

\[
\begin{align*}
\log p_{ij00} &= \lambda_{ij}, \\
\log p_{ij10} &= \lambda_{ij} + \alpha_i + \beta_j + \delta, \\
\log p_{ij01} &= \lambda_{ij} + \alpha_j + \beta_i + \delta, \\
\log p_{ij11} &= \lambda_{ij} + \alpha_i + \alpha_j + \beta_j + \beta_j + 2\delta + \rho_{ij}.
\end{align*}
\]

Note that the model specification includes additional parameters: Namely, there is \( \delta \), the density parameter (as [HL81] put it, “similar to the notion of gross expansiveness in the affective sociometric context”; also called the “choice parameter” in [FW81b]). There are also \((\frac{n}{2})\) dyadic effects, \( \lambda_{ij} \), which are normalizing constants that ensure the four dyad probabilities add to 1 for each dyad. These parameters will, of course, add corresponding sufficient statistics to the vector \( \theta \); however, they are not minimal: for example, the sufficient statistics corresponding to \( \delta \) is the total number of edges in the graph, which can be computed from node degrees.
As Fienberg and Wasserman showed, the model is equivalent to an log-linear model for an \( n \times n \times 2 \times 2 \) contingency table \( u \). The three variants of \( p_1 \) — zero (\( \rho_{ij} = 0 \)), constant (\( \rho_{ij} = \rho \)), and dyad-specific (\( \rho_{ij} = \rho_i + \rho_j \)) reciprocation — correspond to log-linear models specified by following marginals: \([12][13][14][23][24], [12][13][14][23][24][34], \) and \([12][134][234], [12][13][14][23][24][34]\), respectively. Here, \([ij]\) stands for the marginal of the table \( u \) computed with respect to the \( i \)-th and the \( j \)-th variables: \( \sum_{k,l} u_{ijkl} \).

By considering only reciprocated edges, a simple parameter substitution recovers the \( \beta \)-model as a submodel of \( p_1 \) with a similar contingency table representation: \( n \times n \times 2 \) table with the following marginals as sufficient statistics: \([13], [23], [12]\).

### 2.2. Blockmodels

Blockmodels are natural models for network data where nodes belong to groups, i.e. blocks, according to nodal attributes. The general form of a stochastic blockmodel was introduced in \([FW81a]\) and discussed, along with its latent version, in the foundational papers \([HLL83]\) and \([FMW85]\). In \([HLL83]\) and \([FMW85]\), the authors also showed that the non-latent versions of these models are log-linear in form. Models for relational data where node attributes are not observed address a certain set of applications different from the applications we address here, see \([NS01]\) for an overview.

As in the \( p_1 \)-model above, we use \( p_{ijkl} \) to denote the probability of the dyad \((i, j)\) to be in state \((k, l)\) where \((k, l) \in \{0, 1\}^2\). Note that if the network is undirected, the model simply collapses to having only two dyadic states: \((0,0)\) and \((1,1)\). The general class of models with nodal attributes that partition nodes into blocks was defined in \([FMW85]\) as follows:

\[
\begin{align*}
\log p_{ij00} &= \lambda_{ij} \\
\log p_{ij10} &= \lambda_{ij} + \delta^{b(i)b(j)} \\
\log p_{ij01} &= \lambda_{ij} + \delta^{b(j)b(i)} \\
\log p_{ij11} &= \lambda_{ij} + \delta^{b(i)b(j)} + \delta^{b(j)b(i)} + \rho^{b(i)b(j)},
\end{align*}
\]

where each node in the graph belongs to one of \( K \) blocks, \( B_1, \ldots, B_K \), and \( b(i) \) denotes the (known) block assignment of vertex \( i \). Fienberg and Wasserman refer to parameters \( \delta^{rs}, 1 \leq r, s, \leq K, \) as choice effects, and the \( \rho^{rs} \) as reciprocity effects. However, unlike in the standard \( p_1 \)-model, the reciprocity effects are on the level of blocks rather than nodes.

There are now various special cases of these: one can assume that the reciprocity parameters are all equal for any pair of blocks \( B_r \) and \( B_s \), resulting in only one parameter \( \rho \) constant across the network; or assume an additive structure akin to what Holland and Leinhardt did for \( p_1 \); or something entirely different. For example, choosing \( \delta^{rs} = \delta + \alpha^r + \beta^s \) and \( \rho^{rs} = \rho \), as in \([FMW85, Equation (2.10)]\), provides what looks like a block-version of Holland and Leinhardt’s \( p_1 \) — in fact, it’s precisely the \( p_1 \) on the blocks rather than on the nodes; this model was also suggested in \([Bre81]\). The sufficient statistics are the block in-degrees and the block out-degrees, rather than node in- and out-degrees, and the total number \( m \) of reciprocated edges in the network:

\[
t(G) = (d_{B}^{in}, d_{B}^{out}, m) := (d_{B_1}^{in}, \ldots, d_{B_k}^{in}, d_{B_1}^{out}, \ldots, d_{B_k}^{out}, m),
\]

where the in-degree of block \( B_j \) is \( d_j^{in} = \sum_{i \in B_j} d_i^{in} \), the sum of degrees of all nodes in the block, and the out-degree is defined similarly.

While the block models above do not use any node-specific parameters, the authors note that “there are more elaborate subgroup models [...] that combine individual actor with subgroup parameters”, although they do not discuss them explicitly. We discuss them next in some detail.
The \(\beta\)-stochastic blockmodel.\footnote{The \(\beta\)-SBM combines the basic undirected blockmodel above with the \(\beta\)-model, thus incorporating the node effects into the blockmodeling framework. Its exponential family form was formally introduced in [Yan15]. We note that this model is the exponential family variant of the famous \textit{degree-corrected stochastic blockmodel} ([KN11], see also [CWA11]) that has seen numerous applications. Instead of inferring block structure from the relational data akin to some popular community detection algorithms that relate to the degree-corrected SBM, we use background information on the nodes – node attributes – to place them into natural blocks instead (cf. [FW81a]). From the point of view of neuronal data, for example (see Section 4), this consideration is quite natural, as node attributes for groupings are already given by the data.}\footnote{The \(\beta\)-SBM postulates that each node in the graph belongs to one of \(K\) blocks, \(B_1,\ldots,B_K\). Denoting by \(b(i)\) the (known) block assignment of vertex \(i\), the \(\beta\)-SBM gives the following log-odds for the probability \(p_{ij}\) of each dyad \(ij\) being connected:} The \(\beta\)-SBM gives the following log-odds for the probability \(p_{ij}\) of each dyad \(ij\) being connected:

\[
\log \left( \frac{p_{ij}}{1 - p_{ij}} \right) = \beta_i + \beta_j + \alpha_{b(i), b(j)}. \tag{4}
\]

The node-specific parameters \(\beta_i\) play the same role as in the \(\beta\)-model, while the block-specific parameters \(\alpha_{b(i), b(j)}\) encode the propensities of blocks \(i\) and \(j\) being connected and represent the undirected version of \(\delta^{(i)b(j)}\) from Equation (3). The \(\binom{2}{2}\) equations (4) define an exponential family model in which the minimal sufficient statistics are the degree sequence of the nodes and the number of edges between and within blocks, i.e., \(|E_{k,l}|\), where \(1 \leq k \leq l \leq K\):

\[
t(G) = (d_1, \ldots, d_n, |E_{1,1}|, |E_{1,2}|, \ldots, |E_{K,K}|).
\]

The reader should note that restrictions should be placed on the parameter space to ensure identifiability of natural parameters in all of these models; for example, the \(p_1\)-model assumes \(\sum \alpha_i = \sum \beta_i = 0\). For our purposes of testing model fit, we work with the mean value parametrization, so we do not concern ourselves with the natural parameter restrictions.

Using the redundant table representation as before, we arrive at the log-linear model on contingency tables equivalent to the \(\beta\)-blockmodel: the table dimensions are \(n \times n \times 2 \times (K + \binom{K}{2})\) and the marginals are \([13], [23], [34], [124]\).

The \(\beta\)-SBM is an extension of both the usual \(\beta\)-model and the simple blockmodel, as it combines both individual parameters (i.e., the \(\beta\)'s, \(1 \leq i \leq n\)) and the group interaction parameters (i.e., the \(\alpha_{r,s}\)'s, \(1 \leq r, s \leq K\)). There is a natural extension of this model that allows for directed links, so that the symmetry \(\logit(p_{ij}) = \logit(p_{ji})\) no longer holds. Using this as motivation, there are several ways to similarly extend the \(p_1\)-model by incorporating additional group-interaction parameters. To that end, we propose the following general class of degree-based blockmodels for directed networks.

The \(p_1\)-blockmodel. Consider the following general combination of node-specific and block-specific parameters for a directed network: \(\alpha_i, \beta_i, \rho_{ij}, 1 \leq i, j \leq n\), as in the \(p_1\)-model and \(\delta^{rs}\) and \(\delta^{wr}\), \(1 \leq r, s \leq K\), in place of the \(\alpha_{rs}\) from the \(\beta\)-SBM.

**Definition 2.2.** With the parameter setup above, the resulting general \(p_1\)-SBM model family follows the following dyad state probabilities:

\[
\begin{align*}
\log p_{ij00} &= \lambda_{ij} \\
\log p_{ij10} &= \lambda_{ij} + \alpha_i + \beta_j + \delta^{b(i)b(j)} \\
\log p_{ij01} &= \lambda_{ij} + \alpha_j + \beta_i + \delta^{b(j)b(i)} \\
\log p_{ij11} &= \lambda_{ij} + \alpha_i + \alpha_j + \beta_i + \beta_j + \delta^{b(i)b(j)} + \delta^{b(j)b(i)} + \rho_{ij}.
\end{align*}
\]
Because this most general version of the $p_1$-SBM, one may further specify, for example, the reciprocation parameter $\rho_{ij}$ to be constant across the network, i.e., $\rho_{ij} = \rho$, following Holland-Leinhardt. Or, more interestingly, one may specify block-specific reciprocation parameters, $\rho_{ij} = \rho^{b(i)b(j)}$, following Fienberg-Meyer-Wasserman. In this case, a sufficient statistics vector is

$$t(G) = (d^{in}, d^{out}, e_{1,1}, e_{1,2}, \ldots, e_{K,K}),$$

where $e_{k,l}$, $1 \leq k \leq l \leq K$, is the number of edges (edge density), by type, between and within blocks.

Other variants of the model can be specified to allow the parameter interpretation to suit the application at hand. For example, the block parameters can be set to ignore the direction by $\delta^{b(i)b(j)} = \delta^{b(i)b(i)}$; the corresponding statistic counts the total number of edges between a pair of blocks, disregarding type. Or the reciprocation parameter can be defined to be block-specific, measuring the tendency of each block of nodes to reciprocate edges, by setting $\rho^{b(i)b(j)} = \rho^{b(i)} + \rho^{b(j)}$. In this case, the sufficient statistic vector would include the number of reciprocated edges incident to each block. Note that setting the node-specific parameters $\alpha_i$ and $\beta_j$ to zero recovers the general blockmodel [FMW85], while setting the block-specific parameters $\delta^{rs}$ to zero recovers the $p_1$-model [FW81b]; therefore this is a generalization of both.

Using the redundant table representation as before, we arrive at the log-linear models on contingency tables equivalent to the various variants of the $p_1$-blockmodel. For example, when $\rho_{ij} = \rho^{b(i)b(j)}$ is block-constant, the table is of format $n \times n \times 2 \times 2 \times (K + \binom{K}{2})$ with the following margins as sufficient statistics: [125], [345], [134] and [234]. The first fixes the number of edges observed per dyad (set to 1 for simple graphs), the second records the edge count between each pair of blocks, and the last two represent the in-degrees and out-degrees of the nodes in the graph.

3. Goodness-of-fit testing

Given an observed network $G_0$ and a log-linear ERGM with sufficient statistic $t$, our goal is to provide a way to approximate an exact conditional $p$-value by sampling graphs from the conditional distribution given $t(G_0)$. When the sample space $G_n$ consists of simple graphs, $G_0$ can be regarded as a 0/1 contingency table and this conditional distribution is uniform. One can estimate the $p$-value for Fisher’s conditional exact test [Fis25] by sampling from this conditional distribution over the following set:

$$\mathcal{F}_t(G_0) = \{G \in G_n : t(G) = t(G_0)\}. \quad (6)$$

The set $\mathcal{F}_t(G_0)$ in Equation (6) is combinatorial in nature, as it is the set of all graphs with the same network statistics (e.g., edge count between/within blocks, degree sequence, etc.) as the observed graph $G_0$. We refer to it as the reference set or fiber of $G_0$ with respect to $t$.

The sample from the fiber is used to estimate the sampling distribution of a goodness-of-fit statistic of choice, the observed value of which is then compared to the reference distribution of sampled values. For the network models in this paper, we choose to use the chi-squared statistic, and thus by sampling, we are able to estimate the conditional $p$-value

$$Prob\{\chi^2(G) \geq \chi^2(G_0) \mid t(G) = t(G_0)\}.$$

We choose to use the chi-squared statistic as it provably provides a valid test—large values indicate departure from the null hypothesis that the proposed ERGM fits $G_0$ well (for a detailed discussion see [PRF10, Section 2.1] and references given therein).

Approximating the exact conditional test by sampling model fibers has been popular in contingency table analysis and particularly in the algebraic statistics literature, as briefly outlined in the
Introduction. As in contingency table analysis, the size of the fiber $F_t(G_0)$ explodes combinatorially and thus sampling using a Monte Carlo Markov chain becomes necessary. To address the concern of [GZFA09] cited in [ZC13] that “it is unclear whether proposals in this [algebraic statistics] literature are in fact reaching all possible tables associated with the distribution,” we note that the algebraic statistics literature provides proof that the chains are, in fact, reaching all possible tables. Specifically, the connection to algebra established in [DS98] implies explicitly that chains built on Markov bases (collections of moves computed using toric algebra) are irreducible for any log-linear model, and the authors of [HT10, Proposition 2.1] prove that in case of simple graphs, such as in the setting of this paper, chains built on Graver bases (larger collections of moves than Markov bases, but still computed using toric algebra) are irreducible. To sample from the conditional distribution on the fiber $F_t(G_0)$, we use the MCMC algorithm developed in [GPS16] and construct a move at each step using a random combinatorial process. Each element of the Graver basis has a non-zero probability of being constructed, and thus the chain we use is indeed irreducible. For more on Markov bases and Graver bases see [AHT12].

The MCMC algorithm in [GPS16] is a combinatorial modification of the MCMC algorithm in [DS98]. While the algorithm in [GPS16] overcomes some of the computational challenges associated with [DS98], it requires finding balanced edge-sets in the parameter hypergraph of the model, which is a non-trivial problem. Indeed, deciding whether a sub-hypergraph of the parameter hypergraph is a balanced edge-set is known to be NP-hard in general [PS14]. The authors of [GPS16] only explicitly describe how to the find balanced edge-sets for the $p_1$ model with dyad-specific reciprocation. Thus, we move the algorithm from [GPS16] from theory to general practice by explicitly stating solutions to the balanced edge-set problem for the additional log-linear ERGMs described in the previous section and implementing the resulting goodness-of-fit tests in R.

The method in [GPS16] uses the fact that the model is log-linear, which means there exists a linear map that computes $t(G)$ from $G$. This linear map is represented by a matrix, and in particular, for the models in Section 2, a 0/1 matrix. The authors of [GPS16] consider this matrix as the vertex-edge incidence matrix of a hypergraph, which is referred to as the parameter hypergraph of the model. In this setting, a graph $G \in F_t(G_0)$ corresponds to a multiset of edges of the parameter hypergraph, while the property that $t(G) = t(G_0)$ translates to the two multisets of hypergraph edges, one multiset corresponding to $G$ and the other corresponding to $G_0$, having the same degree sequence. Thus, the application of the algorithm in [GPS16] requires the ability to find multisets of edges of the parameter hypergraph that preserve the degree sequence of a given multiset of edges, with each new multiset of hyperedges moving the walk to a new graph.

As mentioned above, the problem of finding a multiset of edges of a hypergraph with a prescribed degree sequence is a difficult sampling problem. For example, efficient uniform sampling of subgraphs with a prescribed degree sequence remains an active area of research (see, e.g., [BD10], [EKMS15], [KDGBT12]). While we have the more general problem of working with hypergraphs, for the specific models in Section 2 the structure of the parameter hypergraph allows us to break the sampling algorithm into pieces that run on graphs. This insight is precisely what allows us to use the approach initiated in [GPS16] to these models.

The general idea of how we implement the walks for each model is as follows. For a complete description of the parameter hypergraphs for each model see Appendix A. The parameter hypergraph of the $\beta$ model is the simplest, since it is isomorphic to the complete graph on $n$ vertices, where $n$ is the number of vertices in the observed network, thus we use a variation of the classic edge swap algorithm ([KTV99b], [RJB96], [Rys87], [TRC11], [Tay81]). For the $\beta$ model, we sample a set of edges $E_1 = \{\{v_1, u_1\}, \ldots, \{v_k, u_k\}\}$ from the current network state and then use $E_1$ to form a closed even walk by adding the edges $E_2 = \{\{u_1, v_2\}, \{u_1, v_2\}, \ldots, \{u_k, v_1\}\}$. We then replace the edges in
E_1 by the edges in E_2, and if the resulting network is simple, we move from the current network state to a new network state, else the move is rejected. This chain is symmetric and ergodic.

The parameter hypergraph \( H \) of the dyad-specific \( p_1 \)-model has both the complete graph on \( n \) vertices \( K_n \) and the complete bipartite graph \( K_{n,n} \) with the \((i,i)\)th edge removed as induced subgraphs; these induced subgraphs correspond to the directed and reciprocated edges respectively. To preserve both the degree of and the total number of reciprocated edges incident to each vertex, the network is first split into directed and reciprocated edges, and then we use the variation of the classic edge swap algorithm described above to move among subgraphs with a fixed degree sequence of \( K_n \) and \( K_{n,n} \setminus \{(i,i) : 1 \leq n\} \). Combining two moves, one for each induced subgraph, gives us a move on the space of all multiset of edges of \( H \) with the prescribed fixed degree sequence. This is the symmetric and ergodic chain described in [GPS16].

For the zero-reciprocation variant of the \( p_1 \)-model, we see that \( K_{n,n} \setminus \{(i,i) : 1 \leq n\} \) is again an induced subgraph of the parameter hypergraph \( H \). Each reciprocated edge \( i \leftrightarrow j \) corresponds to a hyperedge of size four in \( H \), thus we first divide each of these hyperedges into two edges of size two, one which corresponds to the configuration \( i \leftarrow j \) and the other to \( i \rightarrow j \); this is akin to uncoupling the reciprocated edge \( i \leftrightarrow j \) into two directed edges, \( i \leftarrow j \) and \( i \rightarrow j \). After this uncoupling, we then generate a degree-sequence preserving move on \( K_{n,n} \setminus \{(i,i) : 1 \leq n\} \). The constant-reciprocation variant is implemented similarly to the zero reciprocation one with an additional step to ensure that the number of reciprocated edges in the final move is preserved. This step makes sure the vertex corresponding to \( \rho \) (see Appendix A) is covered the same number of times by both multisets of hyperedges. Each of these samplers proposes a combination of directed and/or reciprocated moves, as required, and then ensures that the proposed move is applicable by checking for edge conflicts with the current network.

The samplers for \( \beta \)-SBM and \( p_1 \)-SBM models also utilize the extension of the classic edge-swap algorithm primitive described above with the additional constraint of preserving the sufficient statistics dictated by the blocks. Finally, structural zeros for any of the models are handled by checking whether the proposed new network attempts to place an edge over a structural zero; in that case, a new proposed move is constructed.

To increase the efficiency of the sampling algorithm, we also experimented with optional inputs that modify the chains above, but do not change the limiting distribution. For example, when the graph is dense, we found that we able to explore the space faster by adding a tuning parameter to the algorithm that controls the frequency of using ‘small’ moves, set to be a value between 0 and 1 that controls for the percentage of moves of size four or less. The chain is guaranteed to remain connected as long as this parameter < 1, however, one needs to be careful, as the closer the value is set to one, the longer the chain may take to mix. In a similar vein, for the dyad-specific \( p_1 \) model, there is a tuning parameter vector of length 3 that controls the probabilities of selecting a move only on the directed component, selecting a move only on the reciprocated component, and selecting a combined move. This parameter can be helpful when one component is dense and the other is sparse, as moves are easier to construct on sparser graphs. For graphs with non-empty directed and reciprocated components, the chain remains connected as long as each entry of the vector is greater than zero. Furthermore, there is a tuning parameter that can be used to increase efficiency when testing fit for the \( p_1 \) model with constant reciprocation by allowing the user to use the algorithm for \( p_1 \) with dyad-specific reciprocation, which is more efficient, for a percentage of the moves. Finally, there is a tuning parameter for the \( \beta \)-SBM model that can favor small degree-preserving moves of particular form; namely, moves that allow degrees of individual vertices to change within a block and between two blocks, and do not change the number of edges within each block or between the two blocks.

Finally, it should be noted since we are using the chi-squared statistic, we compute the MLE at
(a) Two graphs simulated from the $\beta$ model on $n = 20$ nodes. The node degrees are heterogeneous and there is no apparent block structure in the networks.

(b) Two graphs simulated from the ER-SBM model on $n = 20$ nodes, $k = 3$ and $k = 4$, respectively. Node degrees appear quite homogeneous.

(c) Two graphs simulated from the $\beta$-SBM model, $n = 20$, $k = 3$ and $k = 4$, resp. Within each block, the subgraph looks similar to that of the $\beta$ model; otherwise node degrees appear less uniform.

Fig 1: Illustration of graphs simulated from the various undirected models considered in this paper.

the start and then at each step of the walk compare the current graph against the MLE. Utilizing the connection to log-linear models, we compute the MLE by standard IPS algorithms for contingency tables, for example, using \texttt{loglin} in R, and thus avoid the use of the ill-behaved MCMC-MLE algorithms for general ERGMs (cf. [HGH08, Section 5.1]).

4. Performance results

We summarize the goodness-of-fit test’s performance on simulated data in Section 4.1 and use it to study two real-world networks from biological applications: a neuronal network in Section 4.2 and protein-protein interaction network in Section 4.3.

4.1. Simulated networks

To test the power of the method, we tested the fit of the $\beta$-SBM for 100 graphs on $n = 100$ nodes simulated from a more complex model. We chose the $\beta$-SBM since tests of goodness of fit for the SBM [KPP+16] and some variants of the $p_1$ model [GPS16] appear in the literature. The simulated graphs were generated using the R package \texttt{ergm} [HHB+08]. We simulated from the ERGM specified
by the model terms degree, nodemix, and triangle; in terms sufficient statistics, these model terms correspond to the sufficient statistic of the $\beta$-SBM with the number of triangles appended, and thus the $\beta$-SBM is a proper submodel of the chosen ERGM. The parameters, chosen uniformly at random for the simulation study, are defined as follows: $\beta_1, \ldots, \beta_n$ are the node parameters controlling node degrees, $\alpha_{ij}$ for $1 \leq i, j \leq K$ are the block connection parameters, and $\theta$ is the triangle parameter. Of the 100 graphs, the $\beta$-SBM model was rejected, with $p$-values smaller than the nominal 0.05, for 90 graphs. For each run, the Markov chain was relatively short with 10,000 steps.

| number of nodes | node parameters | block parameters | triangle parameter |
|-----------------|-----------------|------------------|-------------------|
| 100             | $\beta_i \sim \text{Unif}(-1,1)$ | $\alpha_{ij} \sim \text{Unif}(-1,1)$ | $\theta \sim \text{Unif}(-1,1)$ |

Of the remaining 10 graphs where the model was not rejected, we ran longer chains to address non-mixing issues. Increasing the length of the walk to 30,000 steps only changed the outcome for 1 of the 10 graphs. We noted that the triangle signals in these 10 graphs were not very strong when compared to the distribution of triangles in the full sample. The one graph that did attain a smaller $p$-value, after a longer 30,000-step chain also showed improvement of mixing. The $p$ value dropped to 0.14 from the being previously in the range 0.3 – 0.8.

4.2. Neuronal network data

The neuronal network data set is from [VCP+11] and available online from the WormAtlas http://www.wormatlas.org/. The full data set is a reconstruction of the connectome of the hermaphrodite C. elegans worm. It contains information on 279 of the known 302 C. elegans neurons. In the data set, edges represent chemical or electrical connections between the neurons. The chemical connections are synaptic and directionality can be detected, so this subnetwork, which we will refer to as the chemical subnetwork, is represented as a directed graph. The electrical connections are recorded without direction, thus, this subnetwork, which we will refer to as the electrical subnetwork, is represented as an undirected graph. As the authors in [VCP+11], we will analyze the complete network (the union of the chemical and electrical subnetwork), as well as each subnetwork individually.

The data set also contains vertex attributes that we will use to test for block effects. The attributes that we will focus on are functional classification, which sorts the neurons into three types of neurons, sensory neurons, interneurons, and motor neurons; regional, which classifies neurons according to whether they are found in the head, mid-body, or tail of the worm; and ganglion group (AY NeuronType), a specification that partitions the neurons into 10 groups as described by [AY92].

Simulation results. For the chemical subnetwork of the neuronal network, we tested model fit for the three variants of the $p_1$ model: zero, non-zero constant, and dyad-specific reciprocation. All three variants were rejected for simulations with as many as 100,000 steps in the Markov chain. For illustration purposes, Figure 2a shows a typical output of a shorter run for the dyad-specific reciprocation variant; note that the result is obtained in < 20,000 steps. For the electrical subnetwork of the neuronal network, Figure 2b shows the results of testing the $\beta$ model, which is rejected at significance level > 0.04 (with $p$-values between 0.019 and 0.04 over five iterations of the fit test).

The fact that all four models are rejected with $p$-values often significantly smaller than 0.05 gives evidence against edge formation based on the attractiveness and expansiveness of the nodes alone, even after taking into account possible reciprocation effects. This supports the observation in [VCP+11] against a scale-free model of generation and is contrary to scale invariance hypothesis suggested in [TVA+13].
(a) Test of model fit for the dyad-specific $p_1$ model on the neuronal chemical (directed) network. Left: $p$-value quartiles for 3 iterations of the Markov chain. Right: typical $p$-value estimate from one of the iterations and the sampling distribution of the goodness-of-fit statistics (chi-square) along with its observed value.

(b) Test of model fit for the dyad-specific $p_1$ model on the neuronal electrical (undirected) network. Left: $p$-value quartiles for 5 iterations of the Markov chain. Right: typical $p$-value estimate from one of the iterations and the sampling distribution of the goodness-of-fit statistics (chi-square) along with its observed value.

Fig 2: $p_1$ model fit with dyad-specific reciprocation on two neuronal networks.
Next we consider the mixed neuronal network, consisting of the union of the directed graph—the chemical part—and the undirected graph—the gap junction part.

Since there are three natural neuron groupings—function, region, and ganglion group—we add a parameter for possible homophily to the $\beta$-model and again test model fit. For the neuronal mixed network we tested the $\beta$-SBM model with block assignments given by each of the three groupings; since the results were the same for all three, we show it for the region group block assignment. Test of model fit of the $\beta$-SBM to the mixed network where the network block assignment is given by $\text{blocks} = \text{regionBlocks}$: is illustrated in Figure 3a. Model rejected at any reasonable significance level, with $p$-value very close to 0 ($10^{-4}$).

Since the $p_1$ model naturally interprets undirected edges as reciprocated, testing dyad-specific reciprocation will help delineate the two types of edges. Figure 3b shows that this model actually fits the data well; whereas the variant with zero reciprocation does not (the simulation results omitted for length considerations, but the results look similar to the rejections above).

As we see in simulation results above, we reject the model for the $p_1$-model with zero reciprocation, which again gives evidence against degree-based edge formation. However, for the mixed network under the $p_1$-model with dyad specific reciprocation, the model is not rejected. This could be picking up the signal that some neurons are more likely to be a part of electrical connections as opposed to chemical connections, since the electrical connections are represented as reciprocated edges in the mixed network.

### 4.3. Protein-protein interaction network data

The protein-protein interaction network data set is from [Con11] and available from the Plant Interactome Database (http://interactome.dfci.harvard.edu/A_thaliana/). The data set is the union of two protein-protein interaction graphs which share a subset of proteins as vertices. The first protein-protein interaction graph is a literature-curated network consisting of 3,998 directed interactions on 2,160 proteins. The second protein-protein interaction graph is a partial map of the Arabidopsis thaliana interactome experimentally constructed in [Con11]. This second network consists of 2,661 vertices and 5,529 directed edges. The two networks, $G_1$ and $G_2$ respectively, have 477 proteins in common. We will consider the union of these two directed graphs, which results in a network with 4,344 vertices and 9,449 edges. For testing, we will treat edges with one vertex in $V(G_1) \setminus V(G_2)$ and the other in $V(G_2) \setminus V(G_1)$ (regardless of direction) as structural zeros.

**Simulation results.** As with neuronal network, we also test the A. thaliana protein-protein interaction network for evidence of degree-based edge formation. However, if we test the full network with 4,344 vertices and 9,449 edges, there is a large number of edges that we need to treat as structural zeros since these edges were not tested in the experiment. For a data set of this size, that number of structural zeros slows down the algorithm.

Thus, we ran the goodness of fit test after some optimizing, namely, a parallel implementation, small moves tuning parameter that allows to tune the size of move and make the chain ‘move’ faster, etc., and tested the $p_1$ model with dyad-specific reciprocation effect, taking into account the structural zeros. Several short simulations completed gave conflicting results, with $p$-values ranging from 0.02 to 0.2. Due to the prohibitive computation time, we only ran chains of effective size 20,000 (steps counted disregarding rejected move proposals), arguably insufficient, and did notice that the chains had not mixed yet. A summary of one such run can be seen in Figure 4a. Figure 4b illustrates potential red flags in terms of $p$-value converging: three iterations of the model fit test were run in parallel, and produced different results. A closer look at the simulation reveals that not all the chains have mixed, but the move rejection rate was quite low.
(a) Test of model fit of the $\beta$-SBM model to the neuronal mixed network. Block assignment function is given by RegionBlocks. Left: $p$-value quartiles for 3 iterations on 200,000 steps each. Right: typical $p$-value estimate from one of the iterations.

(b) Test of model fit for the $p_1$ model with dyad-specific reciprocation effect to the neuronal mixed network consisting of the chemical (undirected) and gap junction (directed) graphs. Left: $p$-value quartiles for 5 iterations. Right: a typical $p$-value estimate from one of the iterations and the sampling distribution of the goodness-of-fit (chi-square) statistic along with its observed value.

Fig 3: Two models fits on the neuronal mixed network: the $\beta$-stochastic blockmodel and dyad-specific reciprocation $p_1$. 

(a) A 20,000-step Markov chain (effective walk size $\sim 11,000$) for the dyad-specific $p_1$ model test on the protein-protein interaction network with structural zeros. Three iterations were run, and the chain has not mixed; see also Figure 4b.

(b) $p$-value convergence for the dyad-specific $p_1$ model test on the protein-protein interaction network with structural zeros.

Fig 4: Protein-protein interaction network with a large number of edges that are structural zeros.
We also tested the same model without structural zeros on one of the component graphs and rejected the model after about 15,000 steps with p-values ranging from 0.0005 to 0.05.

The reader should note that the prohibitive computation time is not imposed by how our current implementation handles structural zeros or generation of new graphs. Rather, it is a result of other computational issues, such as computing the values of the chi-square statistic at each discovered network. For each new graph generated in the fiber, the chi-square computation requires on the order of a few billion small computations, even though we are using the most optimized version of it. We do not yet have a principled approach to address the challenge of this particular part of the computation to make it more scalable. The reader is referred to the Discussion section for more details.

### 4.4. Comparison with existing methods

Finally, in this section we compare our results with the goodness-of-fit test implemented in the `ergms.R` package [HHB+08]. For comparison, we will restrict our attention to the \( \beta \)-SBM model. It should be noted that not all models discussed in this paper are implemented in the `ergms` package, as an example, `ergms` does not have built in reciprocation terms for the \( p_1 \) model with edge-dependent reciprocation. However, \( \beta \)-SBM model can be fitted with `ergms`, and so it is a natural model to use as comparison.

The goodness-of-fit method `gof` implemented in `ergms.R` is based on [HGH08]. The method `gof` simulates networks from the fitted model and then computes network statistics of the observed and simulated networks. The method is a graphical test, so to evaluate the goodness-of-fit, the user visually inspects how the distribution of each statistic for the observed network compares with the distributions of the statistic for the simulated networks. Here, using the `ergm` package, we fit the \( \beta \)-SBM model to the mixed, undirected neuronal network with blocks assigned according to “region,” and then run `gof` using the three suggested statistics from [HGH08]: degree, edge-wise shared partners, and minimum geodesic distance. The three plots of these statistics are displayed in Figure 5; the black line represents the statistics of the neuronal network, while the grey lines represent the range for 95 percent of the simulated statistics. While the degree statistics of the neuronal network are within the range of the degree statistics of the simulated networks, the edge-wise shared partners and minimum geodesic distances of the neuronal network differ greatly from the simulated networks. Thus using these plots, we conclude that the \( \beta \)-SBM model is a poor fit for the data, a conclusion supported by Section 4.2.

We also further experiment with other MCMC algorithms to speed up computations, namely Besag and Clifford’s “parallel method” from [BC89]. In this method, one runs the chain backwards from the observed network \( G_0 \) for a prescribed number of steps to obtain a network \( G_1 \). From \( G_1 \), one then runs \( n - 1 \) independent chains forward to get \( n \) simulated random networks that form an approximately independent sample drawn from the uniform distribution on the fiber. While this method worked well for smaller networks, in unoptimized form, it did not scale well for the examples in this paper.

### 5. Conclusion

The guiding question of this work is how to perform statistically satisfying goodness-of-fit tests for a family of network models with good statistical properties and enough flexibility for broad use in applications. While non-asymptotic tests for general ERGMs remain a generally hard problem, we derive finite-sample tests for model fit for a meaningful subclass of ERGMs called log-linear
Fig 5: Comparison of degree, edge-wise shared partners, and minimum geodesic distance statistics between simulated networks and the mixed, undirected neuronal network.

ERGMs. This is done by importing tools from contingency table analysis and adapting them to the combinatorial setting of random graphs, namely, scalable estimation IPS algorithms and dynamic Markov bases for sampling from conditional distributions. The log-linear setting is a familiar setting for networks and, in particular, this paper is rooted in the connection established in the 1980s by [FW81b] and [FW81a].

As an application, we test two popular types of biological datasets in network science, a neuronal network and a protein-protein interaction network. In particular, we test whether the datasets fit several degree-based models, including models with homophily and reciprocation effects. These datasets were chosen since these networks routinely appear as examples of scale-free networks, suggesting degree-based edge formation mechanisms, however, while some authors have suggested exploring biological networks with ERGMs [SF07], [SHL11], up to this point, goodness of fit testing has only been ad hoc. Thus, this work adds not only to the conversation on goodness-of-fit testing for ERGMs, but also to the conversation in network science focused on understanding the general structure of protein-protein interaction networks and neuronal networks. In summary, we rejected most of the models that we fitted to these datasets, with the exception of the $p_1$-model with dyad specific reciprocation for the mixed neuronal network containing both the electrical and chemical subnetworks.

The largest network we tested was a protein-protein interaction network with 4,344 nodes and 9,449 edges (see Section 4.3). Testing how well this network fit different variants of the $p_1$-model with a large number of structural zeros required a prohibitive, or at least impractical, computation time, which illustrates the limits of our algorithms. One bottleneck in the computation is not the graph sampling, rather, it is that the $\chi^2$ statistic is computed at each step in the walk in order to obtain an estimate for the conditional $p$-value. For example, this computation alone requires finding the entry-wise difference between two matrices of size $4344 \times 4344 = 18,870,336$. As fields are moving toward larger and larger datasets, clever computing strategies will be needed to handle networks with a million or more nodes. However, we expect there is room for improvement by exploiting current research in combinatorial algorithms and the computer science side of data science.

While more work needs to be done to build an efficient algorithm for testing large networks, the
Gross, Petrović, Stasi / Exact tests for log-linear ERGMs

results of this work illustrate the feasibility of algorithms that utilize Markov bases for medium-sized datasets. Scalability of algorithms that utilize Markov bases have been an open question in the last decade, since such algorithms traditionally rely on pre-computing the entire Markov basis. Such computations are not only time and resource intensive but also produce mostly inapplicable moves since, by definition, these bases are independent of the observed data [DFR+08, Problem 5.5]; cf. [GZFA09], and are not specialized to any particular fiber of the model. With this issue in mind, various proposals have been made to make exact conditional tests for tables scalable to larger data or more complex models. For example, [Dob12] proposes a general dynamic approach to constructing applicable local moves for marginal table models and proves that they can be utilized to cover the entire fiber. Meanwhile, [HAT12] use a Poisson-size combination of the smallest possible set of moves to explore fibers of discrete logistic regression models, and [MdCCU17] use another small subset of a Markov basis for testing the Ising model on a large biological dataset. Each of these methods extends the use of Markov bases to larger and larger datasets: the approach in [Dob12] was demonstrated on tables up to 256 cells, [HAT12] show their method approaches its limits on $10 \times 10 \times 10$ tables, and [MdCCU17] are able to use their method on a biological dataset of size $800 \times 800$. In this paper, we are able to obtain good results for tables up of size $2661 \times 2661 \times 2 \times 2$ (networks with 2661 vertices) and show that our method slows down due to a goodness-of-fit statistic computation on tables of size $4334 \times 4334 \times 2 \times 2$ (networks with 4334 vertices).

One immediate benefit of using a contingency table representation and Markov bases is the ease of generalization of the goodness of fit test to models for multi-graphs, that is, graphs with positive integer weights on the edges. This can be achieved by simply removing the 0/1 table entry sampling restriction; the resulting estimation algorithms are not affected, and the sampling algorithms based on Markov bases, in fact, become simpler because they do not require the additional step of checking for the graph being simple. Due to this simplification, we expect the sampling algorithm’s mixing time and scalability to improve in this setting.

On a final note, if it is known that the data does not support dyadic independence, analysis using more complicated ERGMs outside of the log-linear ERGM class would be desirable. This is a looming question in the field of ERGMs. In such cases, the sufficient statistic is no longer linear in terms of the edges, and the geometry and combinatorics of the fibers of the resulting model (that is, graphs with the fixed value of the sufficient statistic) is poorly understood. While there exist models that do not assume dyadic independence, ranging from the very simple edge-triangle model [Str86, PN04, PN05] to those using global summary statistics [KPP+17], it is not clear how to use any available tools for testing the dyadic independence assumption on a given network from a statistical point of view. One may turn to estimating graphical models on dyads such as [FS86] or more recent general models from [SR20], however for these models, testing goodness of fit is an open problem as well.

References

[AHT12] Satoshi Aoki, Hisayuki Hara, and Akimichi Takemura, Markov bases in algebraic statistics, vol. 199, Springer Science & Business Media, 2012.

[Alb05] Reka Albert, Scale-free networks in cell biology, Journal of cell science 118 (2005), no. 21, 4947–4957.

[AWC99] Carolyn J Anderson, Stanley Wasserman, and Bradley Crouch, A p* primer: Logit models for social networks, Social networks 21 (1999), no. 1, 37–66.

[AY92] TB Achacoso and WS Yamamoto, Ay’s neuroanatomy of C. elegans for computation, CRC Press, 1992.
Barry Bentley, Robyn Branicky, Christopher L. Barnes, Yee Lian Chew, Eviatar Yemini, Edward T. Bullmore, Petra E. Vertes, and William R. Schafer, The multilayer connectome of caenorhabditis elegans, PLOS Computational Biology 12 (2016), no. 12, 1–31.

Julian Besag and Peter Clifford, Generalized Monte Carlo significance tests, Biometrika 76 (1989), no. 4, 633–642.

Joseph Blitzstein and Persi Diaconis, A sequential importance sampling algorithm for generating random graphs with prescribed degrees, Internet Math 6 (2010), 489–522.

Yvonne M Bishop and Stephen E Fienberg, Discrete multivariate analysis theory and practice.

D. Banerjee and Z. Ma, Optimal hypothesis testing for stochastic block models with growing degrees, Preprint, arXiv:1705.05305.

Ronald L. Breiger, An exponential family of probability distributions for directed graphs: Comment, Journal of the American Statistical Association 76 (1981), no. 373, 51–53.

Sourav Chatterjee and Persi Diaconis, Estimating and understanding exponential random graph models, The Annals of Statistics 41 (2013), no. 5, 2428–2461.

Sourav Chatterjee, Persi Diaconis, and Allan Sly, Random graphs with a given degree sequence, Annals of Applied Probability 21 (2011), no. 4, 1400–1435.

Nicole B. Carnegie, Pavel N. Krivitsky, David R. Hunter, and Steven M. Goodreau, An approximation method for improving dynamic network model fitting, Journal of Computational and Graphical Statistics 24 (2015), no. 2, 502–519.

Arabidopsis Interactome Mapping Consortium, Evidence for network evolution in an arabidopsis interactome map, Science 333 (2011), no. 6042, 601–607.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman, Power-law distributions in empirical data, SIAM review 51 (2009), no. 4, 661–703.

David S. Choi, Patrick Wolfe, and Edo M. Airoldi, Confidence sets for network structure, Advances in Nueral Information Processing Systems, vol. 24, NIPS 2011, 2011.

Adrian Dobra, Stephen E. Fienberg, Alessandro Rinaldo, Aleksandra Slavković, and Yi Zhou, Algebraic statistics and contingency table problems: Log-linear models, likelihood estimation and disclosure limitation, IMA Volumes in Mathematics and its Applications: Emerging Applications of Algebraic Geometry, Springer Science+Business Media, Inc, 2008, pp. 63–88.

Martin Dillon, Runtime for performing exact tests on the p1 statistical model for random graphs, MS thesis, 2016.

Adrian Dobra, Dynamic Markov bases, Journal of Computational and Graphical Statistics (2012), 496–517.

Persi Diaconis and Bernd Sturmfels, Algebraic algorithms for sampling from conditional distribution, Annals of Statistics 26 (1998), no. 1, 363–397.

Péter L. Erdős, Sándor Z Kiss, István Miklós, and Lajos Soukup, Approximate counting of graphical realizations, PloS one 10 (2015), no. 7, e0131300.

Ronald Aylmer Fisher, Statistical methods for research workers, Genesis Publishing Pvt Ltd, 1925.

Stephen E. Fienberg, Michael M. Meyer, and Stanley S. Wasserman, Statistical analysis of multiple sociometric relations, Journal of the American Statistical Association 80 (1985), no. 389, 51–67.
linear models., Annals of Statistics 40 (2012), 996–1023.

[FS86] Ove Frank and David Strauss, *Markov graphs*, Journal of the American Statistical Association 81 (1986), no. 395, 832–842.

[FW81a] Stephen E. Fienberg and Stanley S. Wasserman, *Categorical data analysis of single sociometric relations*, Sociological methodology 12 (1981), 156–192.

[FW81b] ———, *Discussion of Holland, P. W. and Leinhardt, S. “An exponential family of probability distributions for directed graphs”*, Journal of the American Statistical Association 76 (1981), 54–57.

[GPS16] Elizabeth Gross, Sonja Petrović, and Despina Stasi, *Goodness of fit for log-linear network models: Dynamic Markov bases using hypergraphs*, Annals of the Institute of Statistical Mathematics (2016), DOI: 10.1007/s10463-016-0560-2.

[GZFA09] Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi, *A survey of statistical network models*, Foundations and Trends in Machine Learning 2 (2009), no. 2, 129–233.

[HAT12] Hisayuki Hara, Satoshi Aoki, and Akimichi Takemura, *Running Markov chain without Markov basis*, Harmony of Gröbner Bases and the Modern Industrial Society (Takayuki Hibi, ed.), World Scientific, 2012.

[HGH08] David R. Hunter, Steven M. Goodreau, and Mark S. Handcock, *Goodness of fit of social network models*, Journal of the American Statistical Association 103 (2008), no. 481, 248–258.

[HBB+08] David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris, *ergm: A package to fit, simulate and diagnose exponential-family models for networks*, Journal of Statistical Software 24 (2008), no. 3, 1–29.

[HL81] Paul W. Holland and Samuel Leinhardt, *An exponential family of probability distributions for directed graphs (with discussion)*, Journal of the American Statistical Association 76 (1981), no. 373, 33–65.

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt, *Stochastic blockmodels: First steps*, Social networks 5 (1983), no. 2, 109–137.

[HRS+03] Mark S. Handcock, Garry Robins, Tom Snijders, Jim Moody, and Julian Besag, *Assessing degeneracy in statistical models of social networks*, Journal of the American Statistical Association 76 (2003), 33–50.

[HT10] Hisayuki Hara and Akimichi Takemura, *Connecting tables with zero-one entries by a subset of a markov basis*, Algebraic Methods in Statistics and Probability II (Marlos Viana and Henry Wynn, eds.), Contemporary Mathematics, vol. 516, American Mathematical Society, 2010.

[JTA+00] Hawoong Jeong, Béla Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási, *The large-scale organization of metabolic networks*, Nature 407 (2000), no. 6804, 651–654.

[KDGBT12] Hyunjoo Kim, Charo I Del Genio, Kevin E Bassler, and Zoltán Toroczkai, *Constructing and sampling directed graphs with given degree sequences*, New Journal of Physics 14 (2012), no. 2, 023012.

[KN11] Brian Karrer and Mark EJ Newman, *Stochastic blockmodels and community structure in networks*, Physical Review E 83 (2011), no. 1, 016107.

[Kol17] Eric D. Kolaczyk, *Topics at the frontier of statistics and network analysis: (re)visiting the foundations*, Cambridge University Press, 2017.

[KPP+16] Vishesh Karwa, Debdeep Pati, Sonja Petrović, Liam Solus, Nikita Alexeev, Mateja Rač, Dane Wilburne, Robert Williams, and Bowei Yan, *Exact tests for stochastic block
models, arXiv preprint arXiv:1612.06040, 2016.

[KPP+17] Vishesh Karwa, Michael Pelsmajer, Sonja Petrović, Despina Stasi, and Dane Wilburne, Statistical models for cores decomposition of an undirected random graph, Electronic Journal of Statistics 11 (2017), no. 1, 1949–1982.

[KTV99a] Ravi Kannan, Prasad Tetali, and Santosh Vempala, Simple Markov-chain algorithms for generating bipartite graphs and tournaments, Random Structures and Algorithms 14 (1999), no. 1, 293–308.

[KTV99b] , Simple markov-chain algorithms for generating bipartite graphs and tournaments, Random Structures & Algorithms 14 (1999), no. 4, 293–308.

[Lei16] Jing Lei, A goodness-of-fit test for stochastic block models, Annals of Statistics (2016).

[MdCCU17] Abraham Martin del Campo, Sarah Cepeda, and Caroline Uhler, Exact goodness-of-fit testing for the ising model, Scandinavian Journal of Statistics 44 (2017), no. 1, 1949–1982.

[New10] Mark Newman, Networks: an introduction, Oxford university press, 2010.

[NS01] Krzysztof Nowicki and Tom A. B. Snijders, Estimation and prediction for stochastic blockstructures, Journal of the American Statistical Association 96 (2001), no. 455, 1077–1087.

[OHT13] Mitsunori Ogawa, Hisayuki Hara, and Akimichi Takemura, Graver basis for an undirected graph and its application to testing the beta model of random graphs, Annals of Institute of Statistical Mathematics 65 (2013), no. 1, 191–212.

[PN04] J. Park and Mark E. J. Newman, Solution of the two-star model of a network., Physical Review E 70 (2004), no. 3, 066146.

[PN05] , Solution for the properties of a clustered network, Physical Review E 72 (2005), no. 3, 026136.

[PRF10] Sonja Petrović, Alessandro Rinaldo, and Stephen E. Fienberg, Algebraic statistics for a directed random graph model with reciprocation, Algebraic Methods in Statistics and Probability II (M. Viana and H. Wynn, eds.), Contemporary Mathematics, vol. 516, American Mathematical Society, Providence RI, 2010, pp. 261–283.

[PS14] Sonja Petrović and Despina Stasi, Toric algebra of hypergraphs, Journal of Algebraic Combinatorics 39 (2014), no. 1, 187–208.

[PW99] Philippa Pattison and Stanley Wasserman, Logit models and logistic regressions for social networks: Ii. multivariate relations, British Journal of Mathematical and Statistical Psychology 52 (1999), no. 2, 169–193.

[RJB96] A Ramachandra Rao, Rabindranath Jana, and Suraj Bandypadhyay, A Markov chain Monte Carlo method for generating random (0, 1)-matrices with given marginals, Sankhya: The Indian Journal of Statistics, Series A (1996), 225–242.

[RPF13] Alessandro Rinaldo, Sonja Petrović, and Stephen E. Fienberg, Maximum likelihood estimation in the Beta model, Annals of Statistics 41 (2013), no. 3, 1085–1110.

[RPKL07] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher, An introduction to exponential random graph (p*) models for social networks, Social networks 29 (2007), no. 2, 173–191.

[RPW99] Garry Robins, Philippa Pattison, and Stanley Wasserman, Logit models and logistic regressions for social networks: Iii. valued relations, Psychometrika 64 (1999), no. 3, 371–394.

[Rys87] HJ Ryser, Combinatorial properties of matrices of zeros and ones, Classic Papers in Combinatorics (1987), 269–275.

[SF07] Zachary M Saul and Vladimir Filkov, Exploring biological network structure using ex-
ponential random graph models, Biinformatics 23 (2007), no. 19, 2604–2611.

[SHL11] Sean L Simpson, Satoru Hayasaka, and Paul J Laurienti, Exponential random graph modeling for complex brain networks, PloS one 6 (2011), no. 5, e20039.

[SR13] Cosma Rohilla Shalizi and Alessandro Rinaldo, Consistency under sampling of exponential random graph models, Annals of Statistics 41 (2013), no. 2, 508.

[SR20] Kayvan Sadeghi and Alessandro Rinaldo, Hierarchical models for independence structures of networks, Statistica Neerlandica, Netherlands Society for Statistics and Operations Research 74 (2020), no. 3, 439–457, Preprint. arXiv:1605.04565.

[Str86] D. Strauss, On a general class of models for interaction, SIAM review 28 (1986), 513–527.

[Tan05] Reiko Tanaka, Scale-rich metabolic networks, Physical review letters 94 (2005), no. 16, 168101.

[Tay81] Richard Taylor, Constrained switchings in graphs, Combinatorial Mathematics VIII, Springer, 1981, pp. 314–336.

[TRC11] Lionel Tabourier, Camille Roth, and Jean-Philippe Cointet, Generating constrained random graphs using multiple edge switches, Journal of Experimental Algorithmics (JEA) 16 (2011), 1–7.

[TVA+13] Emma K Towlson, Petra E Vértes, Sebastian E Ahnert, William R Schafer, and Edward T Bullmore, The rich club of the c. elegans neuronal connectome, Journal of Neuroscience 33 (2013), no. 15, 6380–6387.

[VCP+11] Lav R. Varshney, Beth. L. Chen, Eric Paniagua, David H. Hall, and Dmitri B. Chklovskii, Structural properties of the caenorhabditis elegans neuronal network, PLoS Computational Biology 7 (2011), no. 2.

[WF01] Andreas Wagner and David A Fell, The small world inside large metabolic networks, Proceedings of the Royal Society of London B: Biological Sciences 268 (2001), no. 1478, 1803–1810.

[WP96] Stanley Wasserman and Philippa Pattison, Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp, Psychometrika 61 (1996), no. 3, 401–425.

[WS98] Duncan J. Watts and Steven H. Strogatz, Collective dynamics of ‘small-world’ networks, Nature 393 (1998), 440–442.

[Yan15] Xiaolin Yang, Social network modeling and the evaluation of structural similarity for community detection, Ph.D. thesis, Carnegie Mellon University, 2015.

[YJFL19] Ting Yan, Binyan Jiang, Stephen E. Fienberg, and Chenlei Leng, Statistical inference in a directed network model with covariates, Journal of the American Statistical Association 114 (2019), no. 526, 857–868.

[ZC13] Jingfei Zhang and Yugu Chen, Sampling for conditional inference on network data, Journal of the American Statistical Association 108 (2013), no. 403.

Appendix A: Parameter hypergraphs

In this appendix, we define and give several examples of a combinatorial structure that encodes any log-linear model: the parameter hypergraph. One should think of this discrete structure object as a schematic representation of the model parametrization, as it encodes the way in which parameters of the model interact. For example in the $\beta$-model, log-odds of each edge probability $p_{ij}, \beta_i + \beta_j$, is encoded by grouping the two parameters $\{\beta_i, \beta_j\}$ together. Each such set represents one edge in the parameter hypergraph. In mathematically precise terms, a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is an ordered pair
where $\mathcal{V}$ is a set of elements called *vertices* and $\mathcal{E}$ is a set of non-empty subsets of $\mathcal{V}$ called *hyperedges*, or simply *edges*. Note that a graph is an instance of a hypergraph where all edges are subsets of $\mathcal{V}$ of size 2.

In our setup, the vertices are the model parameters, and the edges are sets of parameters that appear together in any one of edge probability (or, equivalently, their log-odds). The edge sets for the degree-based models below should be compared to the equations in Section 2.1.

**A.1. Combinatorics of $\beta$-model**

The parameter hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ of the $\beta$-model on $G_n$ has vertex set
\[ \mathcal{V} = \{\beta_i : 1 \leq i \leq n\} \]
and edge set
\[ \mathcal{E} = \{\{\beta_i, \beta_j\} : 1 \leq i < j \leq n\}. \]

Notice that for the $\beta$-model, the parameter hypergraph is the complete graph on $n$ vertices.

**A.2. Combinatorics of the $p_1$ model with zero reciprocation**

The parameter hypergraph of the $p_1$-model with zero reciprocation on $G_n$ is $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where
\[ \mathcal{V} = \{\alpha_i, \beta_j : 1 \leq i \leq n, 1 \leq j \leq n\} \] and \[ \mathcal{E} = \{\{\alpha_i, \beta_j\} : 1 \leq i, j \leq n, i \neq j\}. \]

Notice that by restricting to the edges of size two in $\mathcal{E}$, we obtain the complete bipartite graph $K_{n,n}$ with the edges $\{\alpha_i, \beta_i\}$ removed.

A careful reader may notice the absence of the parameters $\lambda_{ij}$. These are normalizers added to the model parametrization in order to ensure that each dyad is observed in exactly one of the four states. From a combinatorial point of view, they add complexity to the hypergraph, but do not change the underlying structure from which Markov moves for sampling are derived. So it is simpler to not include them in the structural consideration.

**A.3. Combinatorics of the $p_1$ model with constant reciprocation**

The parameter hypergraph of the $p_1$-model with constant reciprocation on $G_n$ is $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where
\[ \mathcal{V} = \{\alpha_i, \beta_j, \rho : 1 \leq i \leq n, 1 \leq j \leq n\} \] and \[ \mathcal{E} = \{\{\alpha_i, \beta_j\}, \{\alpha_i, \alpha_j, \beta_i, \beta_j, \rho\} : 1 \leq i, j \leq n, i \neq j\}. \]

Notice that, as in the zero reciprocation case, by restricting to the edges of size two in $\mathcal{E}$, we obtain the complete bipartite graph $K_{n,n}$ with the edges $\{\alpha_i, \beta_i\}$ removed.

**A.4. Combinatorics of the $p_1$ model with dyad-specific reciprocation**

The parameter hypergraph of the $p_1$-model with constant reciprocation on $G_n$ is $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where
\[ \mathcal{V} = \{\alpha_i, \beta_j, \rho : 1 \leq i \leq n, 1 \leq j \leq n\} \] and \[ \mathcal{E} = \{\{\alpha_i, \beta_j\}, \{\alpha_i, \alpha_j, \beta_i, \beta_j, \rho_i, \rho_j\} : 1 \leq i, j \leq n, i \neq j\}. \]

As in the previous two $p_1$ models, restricting to the edges of size two in $\mathcal{E}$, we obtain the complete bipartite graph $K_{n,n}$ with the edges $\{\alpha_i, \beta_i\}$ removed. Additionally, notice the induced hypergraph on $\mathcal{V}_\rho = \{\rho_i : 1 \leq i \leq n\}$ gives us the complete graph on $n$ vertices.