

---

# Inferring community characteristics in labelled networks

---

**Anonymous Author(s)**

Affiliation  
Address  
email

## Abstract

1 Labelled networks form a very common and important class of data, naturally  
2 appearing in numerous applications in science and engineering. A typical inference  
3 goal is to determine how the vertex labels (or *features*) affect the network's  
4 graph structure. A standard approach has been to partition the network into blocks  
5 grouped by distinct values of the feature of interest. A block-based random graph  
6 model – typically a variant of the stochastic block model – is then used to test  
7 for evidence of asymmetric behaviour within these feature-based communities.  
8 Nevertheless, the resulting communities often do not produce a natural partition  
9 of the graph. In this work, we introduce a new generative model, the feature-first  
10 block model (FFBM), which is more effective at describing vertex-labelled undi-  
11 rected graphs and also facilitates the use of richer queries on labelled networks.  
12 We develop a Bayesian framework for inference with this model, and we present a  
13 method to efficiently sample from the posterior distribution of the FFBM parame-  
14 ters. The FFBM's structure is kept deliberately simple to retain easy interpretability  
15 of the parameter values. We apply the proposed methods to a variety of network  
16 data to extract the most important features along which the vertices are partitioned.  
17 The main advantages of the proposed approach are that the whole feature-space  
18 is used automatically, and features can be rank-ordered implicitly according to  
19 impact. Any features that do not significantly impact the high-level structure can  
20 be discarded to reduce the problem dimension. In cases where the vertex features  
21 available do not readily explain the community structure in the resulting network,  
22 the approach detects this and is protected against over-fitting. Results on several  
23 real-world datasets illustrate the performance of the proposed methods.

24 

## 1 Introduction

25 An important characteristic of many real-world networks is that they exhibit strong community  
26 structure, with most nodes often belonging to a densely connected cluster. Finding ways to recover  
27 the latent communities from the observed graph is an important task in many applications, including  
28 graph/network compression [1] and link prediction in incomplete networks [4].

29 In this work, we restrict our attention to vertex-labelled networks, and we refer to the vertex labels as  
30 *features*. A common goal is to determine whether a given feature impacts graphical structure. To  
31 answer this from a Bayesian perspective requires the use of a random graph model; the standard is  
32 the stochastic block model (SBM) [8]. This is a latent variable model where each vertex belongs to a  
33 single block and the probability two nodes are connected depends only on the block memberships  
34 of each. Numerous variants of this model have been considered – the most popular ones being the

35 mixed-membership stochastic block model (MMSBM) [2] and the overlapping stochastic block model  
 36 (OSBM) [17]. Effectively, these extend the model to allow each vertex to belong to multiple blocks  
 37 simultaneously. However, one drawback of these graphical models as applied to labelled networks  
 38 is that they do not automatically include vertex features in the random graph generation process.  
 39 Approaches based on graph neural networks [7] that utilise vertex features have been developed but  
 40 these lack the easy interpretability of the simpler models.

41 To analyse a labelled network using one of the simple SBM variants, a typical inference procedure  
 42 would be to first partition the graph into blocks grouped by distinct values of the feature of interest,  
 43 and then use the associated model to test for evidence of heterogeneous connectivity between the  
 44 feature-grouped blocks. Nevertheless, this approach is limited in that it can only consider one feature  
 45 at a time. This makes it difficult to rank-order the features by magnitude of impact. Lastly, the  
 46 feature-grouped blocks are often an unnatural partition of the graph, leading to a poor model fit. We  
 47 would instead prefer to partition the graph into its most natural blocks and then find which of the  
 48 available features – if any – best predict the resulting partition.

49 Thus motivated, we present a novel framework for modelling labelled networks, which we call the  
 50 feature-first block model (FFBM). This is an extension of the SBM to labelled networks. We go on to  
 51 present an efficient algorithm for sampling from the parameters of the feature-to-block generator, and  
 52 to describe how the sampled FFBM parameters can be interpreted to determine which features have  
 53 the largest impact on overall graphical structure.

## 54 2 Preliminaries

55 We first need a model for community-like structure in a network. For this we adopt the widely-used  
 56 stochastic block model (SBM). Each node in the graph belongs to a unique community called a  
 57 block. The probability that two nodes are connected depends only on the block memberships of  
 58 each. Specifically, we will use the microcanonical variant of the SBM, proposed by Peixoto [13].  
 59 To allow for degree-variability between members of the same block, we adopt the degree-corrected  
 60 formulation (DC-SBM), defined in (2.1).

61 For each integer  $K \geq 1$ , we use the notation  $[K] := \{1, 2, \dots, K\}$ .

62 **Definition 2.1 (Microcanonical DC-SBM)** *Let  $N \geq 1$  denote the number of vertices in an undirected graph. The block memberships are encoded by a vector  $b \in [B]^N$ , where  $B$  is the number of non-empty blocks. Let  $e = (e_{rs})$  be the  $N \times N$  symmetric matrix of edge counts between blocks, such that  $e_{rs}$  is the number of edges from block  $r$  to block  $s$  – or twice that number if  $r = s$ . Let  $k = (k_i)$  denote the vector of length  $N$  containing the degree sequence of the graph, with  $k_i$  being the degree of vertex  $i$ .*

68 The graph's adjacency matrix  $A \in \{0, 1\}^{N \times N}$  is generated by placing edges uniformly at random,  
 69 but respecting the constraints imposed by  $e$ ,  $b$  and  $k$  (hence the qualification 'microcanonical').  
 70 Specifically,  $A$  must satisfy the following, for all  $r, s \in [B]$  and all  $i \in [N]$ :

$$e_{rs} = \sum_{i,j \in [N]} A_{ij} \mathbb{1}\{b_i = r\} \mathbb{1}\{b_j = s\} \quad \text{and} \quad k_i = \sum_{j \in [N]} A_{ij}. \quad (1)$$

71 We use the following notation to indicate this distribution:

$$A \sim \text{DC-SBM}_{\text{MC}}(b, e, k). \quad (2)$$

72 **3 Feature-first block model**

73 In this section we propose a novel generative model for labelled networks. We call this the feature-first  
 74 block model (FFBM).

75 Let  $N$  denote the number of nodes,  $B$  the number of blocks in the graph, and  $\mathcal{X}$  the set of values  
 76 each feature can take. We define the vector  $x_i \in \mathcal{X}^D$  as the feature vector of vertex  $i$ , where  $D$  is the  
 77 number of features associated with each vertex. For the datasets we analyse, we deal with binary  
 78 feature flags so  $\mathcal{X} = \{0, 1\}$ . We write  $X$  for the  $N \times D$  *feature matrix* containing the feature vectors  
 79  $\{x_i\}_{i=1}^N$  as its rows.

80 For the FFBM, we start with the feature matrix  $X$  and generate a random vector of block memberships  
 81  $b \in [B]^N$ . For each vertex  $i$ , the block membership  $b_i \in [B]$  is generated based on the feature vector  
 82  $x_i$ , independently between vertices. The conditional distribution of  $b_i$  given  $x_i$  also depends on  
 83 a collection of weight vectors  $\theta = \{w_k\}_{k=1}^B$ , where each  $w_k$  has dimension  $D$ . Specifically, the  
 84 distribution of  $b$  given  $X$  and  $\theta$  is,

$$p(b|X, \theta) = \prod_{i=1}^N p(b_i|x_i, \theta) = \prod_{i=1}^N \phi_{b_i}(x_i; \theta) = \prod_{i=1}^N \frac{\exp(w_{b_i}^T x_i)}{\sum_{k=1}^B \exp(w_k^T x_i)}. \quad (3)$$

85  $\phi_{b_i}$  is the form of a softmax activation function; we deliberately exclude a bias term to ensure that we  
 86 only leverage feature information. We will later also find it convenient to write the parameters  $\theta$  as a  
 87  $B \times D$  matrix of weights  $W$ .

88 More complex models based on different choices for the distributions  $\phi_{b_i}$  above are also possible, but  
 89 then deriving meaning from the inferred parameter distributions is more difficult.

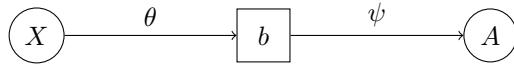


Figure 1: The feature-first block model (FFBM)

90 Once the block memberships  $b$  have been generated, we then draw the graph  $A$  from the microcanonical  
 91 DC-SBM with additional parameters  $\psi = \{\psi_e, \psi_k\}$ :

$$A \sim \text{DC-SBM}_{\text{MC}}(b, \psi_e, \psi_k). \quad (4)$$

92 **3.1 Prior selection**

93 To complete the description of our Bayesian framework, priors on  $\theta$  and  $\psi$  must also be specified.  
 94 We place a Gaussian prior on  $\theta$  with zero mean and covariance matrix  $\sigma_\theta^2 I$ , so that each element of  $\theta$   
 95 has an independent  $\mathcal{N}(0, \sigma_\theta^2)$  prior, with hyperparameter  $\sigma_\theta^2$ :

$$p(\theta) \sim \mathcal{N}(\theta; 0, \sigma_\theta^2 I). \quad (5)$$

96 Interestingly, this choice of prior leads to a particularly simple distribution on the block membership  
 97 vector  $b$  given  $X$ . We show in Appendix B.1 that, after integrating out  $\theta$ ,  $b$  is uniformly distributed  
 98 given  $X$ :

$$p(b|X) = \int p(b|X, \theta)p(\theta)d\theta = B^{-N}. \quad (6)$$

99 This is an important simplification as evaluating  $p(b|X)$  does not require an expensive Monte Carlo  
 100 integration over  $\theta$  nor does it require the exact value of  $X$ . Peixoto [13] proposes careful choices  
 101 for the priors on the additional microcanonical SBM parameters  $\psi$ , which we adopt. The idea is to  
 102 write the joint distribution on  $(b, e, k)$  as a product of conditionals,  $p(b, e, k) = p(b)p(e|b)p(k|e, b) =$   
 103  $p(b)p(\psi|b)$ . In our case, conditioning on  $X$  is also necessary, leading to,

$$p(b, \psi|X) = p(b|X)p(\psi|b, X) = p(b|X)p(\psi|b), \quad (7)$$

104 where we used the fact  $\psi$  and  $X$  are conditionally independent given  $b$ . The exact form of the prior  
 105  $p(\psi|b)$  is described in Appendix A.1. All that concerns the main argument is that it has an easily  
 106 computable form.

107 **4 Inference**

108 Having completed the definition of the FFBM, we wish to leverage it to perform inference. Given a  
 109 vertex-labelled graph  $(A, X)$ , the goal is to sample from the posterior distribution of  $\theta$ :

$$\theta \sim p(\theta|A, X). \quad (8)$$

110 We propose an iterative approach to obtain samples  $\theta^{(t)}$  from  $p(\theta|A, X)$ . We first draw samples  $b^{(t)}$   
 111 from the block membership posterior distribution, and then use each  $b^{(t)}$  to obtain a corresponding  
 112 sample  $\theta^{(t)}$ :

$$b^{(t)} \sim p(b|A, X), \quad (9)$$

$$\theta^{(t)} \sim p(\theta|X, b^{(t)}). \quad (10)$$

113 Both of these sampling steps can be implemented using Markov chain Monte Carlo via a Metropolis-  
 114 Hastings algorithm [5], as a two-level Markov chain.

115 As we show in the following section, the resulting samples for  $\theta^{(t)}$  are unbiased in the sense that the  
 116 expectation of their distribution is the target posterior distribution:

$$\mathbb{E}_{b^{(t)}} \left[ p(\theta|X, b^{(t)}) \right] = \sum_{b \in [B]^N} p(\theta|X, b) p(b|A, X) = \sum_{b \in [B]^N} p(\theta, b|A, X) = p(\theta|A, X). \quad (11)$$

117 This is an example of a pseudo-marginal approach. Indeed, Andrieu and Roberts [3] show that (11) is  
 118 sufficient to prove that, for large enough  $t$ ,  $\theta^{(t)} \sim \mathbb{E}_{b^{(t)}} [p(\theta|X, b^{(t)})] = p(\theta|A, X)$ , as required.

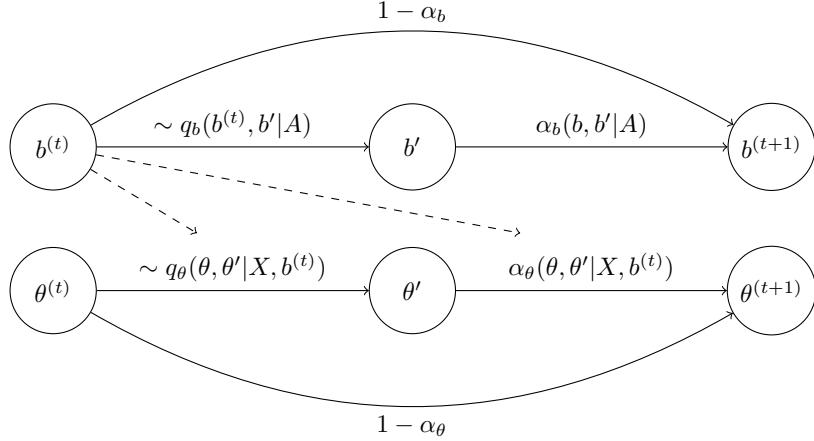


Figure 2: Sampling sequence.

119 The reason for splitting the Markov chain into two levels is that the summation over all latent states  
 120  $b \in [B]^N$  required to directly compute the likelihood,  $p(A|X, \theta) = \sum_{b \in [B]^N} p(A|b)P(b|X, \theta)$ , is  
 121 intractable, as the sum involves  $B^N$  terms. Figure 2 shows an overview of the proposed method. The  
 122 Metropolis-Hastings accept/reject probabilities for the proposed  $b$  and  $\theta$  samples are denoted  $\alpha_b$  and  
 123  $\alpha_\theta$ , respectively.

124 Note the importance of the simplification in (6). As  $p(b|X)$  in fact does not depend on  $X$ , knowing  
 125  $X$  is not necessary in order to obtain samples of the block membership vector  $b$ . And on the other  
 126 level, in order to obtain samples of the parameter vector  $\theta$  we use only  $b$  but not  $A$ , as  $\theta$  and  $A$  are  
 127 conditionally independent given  $b$ .

128 **4.1 Sampling block memberships**

129 We adopt the Markov chain Monte Carlo procedure of [11], which relies on writing the posterior in  
130 the following form:

$$p(b|A, X) \propto p(A|b, X) \cdot p(b|X) = \pi_b(b). \quad (12)$$

131 Now  $\pi_b(\cdot)$  is the un-normalised target density, which can be expressed as:

$$\pi_b(b) = p(b|X) \sum_{\psi} p(A, \psi|b, X) = p(b|X)p(A, \psi^*|b, X) = p(A|b, \psi^*) \cdot p(\psi^*|b) \cdot p(b|X). \quad (13)$$

132 Since we are using the microcanonical SBM formulation, there is only one value of  $\psi$  that is  
133 compatible with the given  $(A, b)$  pair; recall the constraints in (1). We denote this value  $\psi^* =$   
134  $\{\psi_k^*, \psi_e^*\}$ . Therefore, the summation over all  $\psi$  reduces to just the single  $\psi^*$  term. We also define the  
135 microcanonical entropy of the configuration as,

$$S(b) = -\log \pi_b(b) = -\left( \log p(A|b, \psi^*) + \log p(\psi^*, b|X) \right). \quad (14)$$

136 This entropy can be thought of as the optimal ‘‘description length’’ of the graph. The exact form  
137 of the proposal  $q_b$  is explored thoroughly in [11] and not repeated here. There is a widely used  
138 library for Python made available under LGPL called `graph-tool` [12], which implements this  
139 algorithm. The only modification we make is in the block membership prior  $p(b)$  that we replace  
140 with  $p(b|X) = B^{-N}$ , which cancels out in the MH accept-reject step as it is independent of  $b$ .

141 **4.2 Sampling feature-to-block generator parameters**

142 The invariant distribution we wish to target for the  $\theta$  samples is the posterior of  $\theta$  given the values of  
143 the pair  $(X, b)$ . We write this as,

$$\pi_\theta(\theta) \propto p(\theta|X, b) \propto p(b|X, \theta)p(\theta) \propto \exp(-U(\theta)), \quad (15)$$

144 where we write  $U(\theta)$  for the negative log-posterior. Let  $y_{ij} := \mathbb{1}\{b_i = j\}$  and  $a_{ij} := \phi_j(x_i; \theta)$ .  
145 Discarding constant terms, we can write  $U(\theta)$  as,

$$U(\theta) = \left( \sum_{i \in [N]} \sum_{j \in [B]} y_{ij} \log \frac{1}{a_{ij}} \right) + \frac{1}{2\sigma_\theta^2} \|\theta\|^2 = N \cdot \mathcal{L}(\theta) + \frac{1}{2\sigma_\theta^2} \|\theta\|^2; \quad (16)$$

146 See Appendix B.2 for the derivation. The function  $U(\theta)$  is a typical objective function for neural  
147 network training. The first term is introduced by the likelihood. We collect it into  $N \cdot \mathcal{L}(\theta)$ , which is  
148 the cross-entropy between the graph-predicted and feature-predicted block memberships summed  
149 over all vertices. The second, introduced by the prior, brings a form of regularisation, guarding against  
150 over-fitting. Unlike in many applications where the goal is to find the minimiser of  $U(\theta)$ , our goal is  
151 to draw samples from the posterior  $\pi_\theta(\cdot) \propto \exp(-U(\cdot))$ . We can use  $\nabla U$  as a useful heuristic to bias  
152 our proposal towards regions of higher target density. We therefore adopt the Metropolis-adjusted  
153 Langevin algorithm (MALA) [14]. Given the current sample  $\theta$ , we generate a new sample  $\theta'$  as,

$$\theta' = \theta - h \nabla U(\theta) + \sqrt{2h} \cdot \xi,$$

154 where  $\xi \sim \mathcal{N}(0, I)$  and  $h$  is a step-size parameter which may vary with the sample index (see  
155 Appendix A.2). This leads to the proposal distribution,

$$q_\theta(\theta, \theta') \sim \mathcal{N}(\theta'; \theta - h \nabla U(\theta), 2hI).$$

156 Without the injected noise term, MALA is equivalent to gradient descent. We require the noise term  
157  $\xi$  to fully explore the parameter space. The term  $\nabla U$  has an easy to compute analytic form (derived  
158 in Appendix B.3). By noting that  $\theta = \{w_k\}_{k=1}^B$ , we write the derivative with respect to each  $w_k$  as,

$$\frac{\partial U}{\partial w_k} = - \left( \sum_{i=1}^N \left\{ \hat{x}_i(y_{ik} - a_{ik}) \right\} - \frac{w_k}{\sigma_\theta^2} \right). \quad (17)$$

159 After a proposed move is generated from  $q_\theta$ , it is either accepted or rejected in the standard Metropolis-  
160 Hastings accept/reject step.

161 **4.3 Sampling sequence**

162 Up to this point, each  $\theta^{(t)}$  update uses its corresponding  $b^{(t)}$  sample. This means that the evaluation  
 163 of  $U(\theta)$  and  $\nabla U(\theta)$  has high variance. This may lead to longer burn-in for the resulting Markov  
 164 chain. The only link between  $b^{(t)}$  and  $\theta^{(t)}$  is in the evaluation of  $U(\theta)$  and  $\nabla U(\theta)$  which depends  
 165 only on the matrix  $y^{(t)}$  with entries  $y_{ij}^{(t)} := \mathbb{1}\{b_i^{(t)} = j\}$ . We would rather deal with the expectation  
 166 of each  $y_{ij}^{(t)}$ :

$$\mathbb{E} [y_{ij}^{(t)}] = \mathbb{E}_{b^{(t)}} [\mathbb{1}(b_i^{(t)} = j)] = p(b_i = j | A, X). \quad (18)$$

167 We can obtain an unbiased estimate for this quantity using the thinned  $b$ -samples after burn-in; the  
 168 length of the burn-in and the details of the thinning are described in Appendix A.3. Let  $\mathcal{T}_b$  denote the  
 169 retained set of indices for the  $b$ -samples and  $\mathcal{T}_\theta$  similarly for the  $\theta$ -chain. The unbiased estimate for  
 170  $y_{ij}^{(t)}$  using the restricted sample set  $\mathcal{T}_b$  is denoted  $\hat{y}_{ij}$ :

$$\hat{y}_{ij} := \frac{1}{|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} y_{ij}^{(t)} = \frac{1}{|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} \mathbb{1}\{b_i^{(t)} = j\}. \quad (19)$$

171 The same matrix  $\hat{y}$  is used in the update step for each  $\theta^{(t)}$ . This way, it is not necessary to run the  $b$   
 172 and  $\theta$  Markov chains concurrently. Instead, we run the  $b$ -chain to completion and use it to generate  $\hat{y}$ .  
 173 This affords us the flexibility to vary the lengths of the  $b$  and  $\theta$ -chains. Furthermore, the changeover  
 174 from  $y^{(t)}$  to  $\hat{y}$  reduces the burn-in time for the  $\theta$ -chain by reducing the variance in the evaluation of  
 175  $U$  and  $\nabla U$ . A detailed description of the overall algorithms is given in Appendix C.1.

176 **4.4 Dimensionality reduction**

177 Once the samples  $\{\theta^{(t)}\} \sim p(\theta | A, X)$  have been obtained, we can compute the empirical mean and  
 178 standard deviation of each component of  $\theta$ . Switching back to matrix notation we define  $\theta = W$ ,  
 179 such that  $W_{ij}$  is the weight component for block  $i$  and feature  $j$ , we can define:

$$\hat{\mu}_{ij} := \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} W_{ij}^{(t)} \quad \text{and} \quad \hat{\sigma}_{ij}^2 := \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} (W_{ij}^{(t)} - \hat{\mu}_{ij})^2 \quad (20)$$

180 A simple heuristic to discard the least important features requires specifying a cutoff  $c > 0$  and a  
 181 multiplier  $k > 0$ . We define the function  $\mathcal{F}_i(j)$  as in (21) and only keep features with indices  $d \in \mathcal{D}'$ ,  
 182 where  $\mathcal{D}'$  is given in (22).

$$\mathcal{F}_i(j) := (\hat{\mu}_{ij} - k\hat{\sigma}_{ij}, \hat{\mu}_{ij} + k\hat{\sigma}_{ij}) \cap (-c, +c), \quad (21)$$

$$\mathcal{D}' := \{j \in [D] : \exists i \in [B] \text{ s.t. } \mathcal{F}_i(j) = \emptyset\} \quad (22)$$

183 Intuitively, this means discarding any feature  $j$  for which  $\hat{\mu}_{ij} \pm k\hat{\sigma}_{ij}$  lies outside the region  $(-c, c)$   
 184 for all block indices  $i$ . If we were to use the Laplace approximation for the posterior  $p(W_{ij} | A, X) \approx$   
 185  $\mathcal{N}(W_{ij}; \hat{\mu}_{ij}, \hat{\sigma}_{ij}^2)$ , then this is analogous to a hypothesis test on the value of  $W_{ij}$  as in (23); then  $\mathcal{D}'$   
 186 comprises all features  $i$  for which  $H_1$  is not rejected at least once for some  $j \in [B]$

$$H_0 : |W_{ij}| \leq c \quad H_1 : |W_{ij}| > c \quad (23)$$

187 The multiplier  $k$  in equation 21 determines the degree of significance of the result. However, as the  
 188 Laplace approximation is not exact we will only treat this dimensionality reduction method as a  
 189 useful heuristic and not an exact method. Conversely, we could fix  $k = k_0$  and the dimension of our  
 190 reduced feature set  $|\mathcal{D}'| = D'$ . We would then like to find the largest value of  $c$  such that  $|\mathcal{D}'| = D'$   
 191 given  $k = k_0$ . This is summarised in equation 24.

$$c^* = \arg \max \{c > 0 : |\mathcal{D}'| = D', k = k_0\}. \quad (24)$$

192 For an algorithmic implementation of this method refer to appendix C.1. This approach has the  
 193 advantage that it fixes the number of reduced dimensions.

194 **5 Experimental results**

195 We apply the developed methods to a variety of datasets. These are chosen to span a range of node  
 196 counts  $N$ , edge counts  $E$  and feature space dimension  $D$ . We consider the following:

- 197 • **Political books** [9] ( $N = 105, E = 441, D = 3$ ) – network of Amazon book sales about U.S.  
 198 politics, published close to the presidential election in 2004. Two books are connected if they were  
 199 frequently co-purchased by customers. Vertex features encode the political affiliation of the author  
 200 (liberal, conservative, or neutral).
- 201 • **Primary school dynamic contacts** [15] ( $N = 238, E = 5539, D = 13$ ) – network of face-to-face  
 202 contacts amongst students and teachers at a primary school in Lyon, France. Two nodes are  
 203 connected if the two parties shared a face-to-face interaction over the school-day. Vertex features  
 204 include class membership (one of 10 values: 1A-5B), gender (male, female) and teacher status  
 205 encoded as an 11th school-class. No further identifiable information is retained. We choose to  
 206 analyse just the second day of results.
- 207 • **Facebook egonet** [6] ( $N = 747, E = 30025, D = 480$ ) – an assortment of Facebook users'  
 208 friends lists. Vertex features are extracted from each user's profile and are fully anonymised. They  
 209 include information about education history, languages spoken, gender, home-town, birthday etc.  
 210 We focus on the egonet with id 1912.

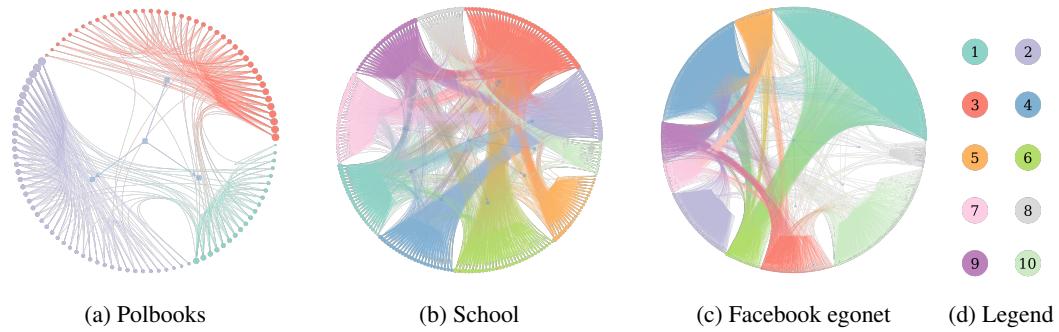


Figure 3: Networks laid out and coloured according by inferred block memberships  $\hat{y}$  for a given experiment iteration. Visualisation performed using *graph-tool* [12].

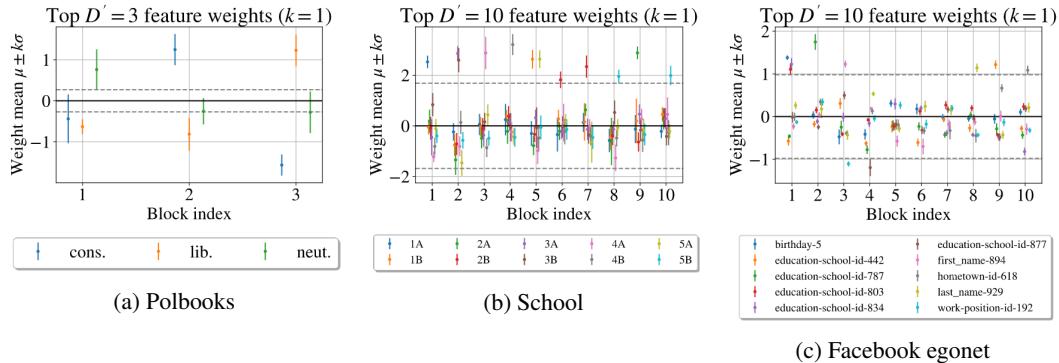


Figure 4: Reduced dimension feature-to-block generator weight samples. Dotted line is  $\pm \sigma^*$

211 We require metrics to assess performance. This can be split into two separate components: the  
 212 microcanonical SBM fit (concerned with the  $b$ -samples) and the fit of the feature-to-block generator  
 213 (concerned with the  $\theta$ -samples).

214 Starting with the SBM, recall that the quantity  $S(b)$  in (14) can be interpreted as an ideal “description  
 215 length” of the partition imposed by  $b$ . We define a simple metric  $\bar{S}_e$  to gauge the fit of the SBM,

Table 1: Experimental results averaged over  $n = 10$  iterations (mean  $\pm$  standard deviation).

Dataset	$B$	$D$	$D'$	$S_e$	$\bar{\mathcal{L}}_0$	$\bar{\mathcal{L}}_1$	$c^*$	$\bar{\mathcal{L}}'_0$	$\bar{\mathcal{L}}'_1$
Polbooks	3	3	–	$2.250 \pm 0.000$	$0.563 \pm 0.042$	$0.595 \pm 0.089$	–	–	–
School	10	13	10	$1.894 \pm 0.004$	$0.787 \pm 0.127$	$0.885 \pm 0.129$	$1.198 \pm 0.249$	$0.793 \pm 0.132$	$0.853 \pm 0.132$
FB egonet	10	480	10	$1.626 \pm 0.003$	$1.326 \pm 0.043$	$1.538 \pm 0.069$	$0.94 \pm 0.019$	$1.580 \pm 0.150$	$1.605 \pm 0.106$

as the description length per entity (i.e., divided by the total number  $N + E$  of nodes plus edges), averaged over the  $b$ -samples:

$$\bar{S}_e := \frac{1}{(N+E)|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} S(b^{(t)}). \quad (25)$$

Next, to assess the performance of the feature-to-block predictor, we partition the vertex set  $[N]$  into training and test sets. We choose to randomly partition the vertices on each experiment run, so that a constant fraction  $f$  of the available vertices go to form the training set  $\mathcal{G}_0$  and the remainder are held out to form the test set  $\mathcal{G}_1$ . The  $b$ -chain is run using the whole network but we only use vertices  $v \in \mathcal{G}_0$  to train the  $\theta$ -chain. Because  $|\mathcal{G}_0| \neq |\mathcal{G}_1|$  in general, we cannot use the un-normalised log-target  $U$  from (16) for comparison, as the total cross-entropy loss scales with the size of each data set but the prior term stays constant. We therefore use the average cross-entropy loss over each set,

$$\bar{\mathcal{L}}_\star := \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} \mathcal{L}_\star(\theta^{(t)}), \quad \text{where } \mathcal{L}_\star(\theta^{(t)}) := \frac{1}{|\mathcal{G}_\star|} \sum_{i \in \mathcal{G}_\star} \sum_{j \in [B]} \hat{y}_{ij} \log \frac{1}{\phi_j(x_i; \theta^{(t)})}, \quad (26)$$

where  $\star \in \{0, 1\}$  indicates whether the training or test set is being considered.

Table 1 summarises the results for each experiment; a complete list of all the hyperparameters used is given in Appendix C.2. We also apply the dimensionality reduction method of Section 4.4 to the two higher dimensional datasets (the school and FB egonet). For this we use equation (24) with  $k = 1$ , in order to reduce the dimension from  $D$  to a desired  $D'$ . We then retrain the feature-block predictor using only the retained feature set  $\mathcal{D}'$ , and report the log-loss over the training and test sets for the reduced classifier – denoted  $\bar{\mathcal{L}}'_0$  and  $\bar{\mathcal{L}}'_1$  respectively. These values are also shown in Table 1.

Based on these results, some remarks are in order. Firstly, the variance of the test loss  $\bar{\mathcal{L}}_1$  tends to be higher than the training loss  $\bar{\mathcal{L}}_0$ . This is expected, as the test set is smaller than the training set and hence more susceptible to variability in its construction. Indeed, most of the variance in the evaluation of  $\bar{\mathcal{L}}_0$  and  $\bar{\mathcal{L}}_1$  comes from the random partitioning of the graph into training and test sets. Secondly, it can be seen that the dimensionality reduction procedure brings the training and test losses closer together. This implies that the features we keep are indeed correlated with the underlying graphical partition and that the approach generalises correctly.

The average description length per entity,  $\bar{S}_e$ , of the graph, has very small variance, suggesting that the detected communities can be found reliably (to within an arbitrary relabelling of blocks). For reference, we plot an inferred partition for each of the graphs in Figure 3. The polbooks graph yields the cleanest separation between blocks but nonetheless the inferred partitions for the other datasets do succeed at dividing the graph into densely connected clusters.

## 5.1 Political books

The goal here is to determine whether the authors' political affiliations are a good predictor of the overall network structure. We choose to partition the network into  $B = 3$  communities as we only have this many distinct values for political affiliation (conservative, liberal or neutral). From Figure 4a we see that all 3 blocks have a distinct political affiliation as their largest positive component. This strongly indicates that political affiliation is indeed the axis which best predicts the 3-way natural partition of the graph into blocks. Furthermore, in Table 1 we see that the training and test losses are very similar and both are low in magnitude. This provides further evidence to the claim that political affiliation is a very appropriate explanatory variable for the overall network structure.

253 **5.2 Primary school dynamic contacts**

254 We choose the number of communities  $B = 10$ , in line with the total number of school classes.  
255 As before, we sample the block-generator parameters  $\theta$  and employ the dimensionality reduction  
256 technique of Section 4.4, with standard deviation multiplier  $k = 1$  to pick out the top  $D' = 10$   
257 features. We then plot the weights for the resulting features  $d \in \mathcal{D}'$  in Figure 4b. It is rather  
258 obvious that only the pupils’ class memberships (1A-5B) have remained significant; gender and  
259 teacher/student status have been discarded, meaning that these are not good predictors of overall  
260 macro-structure.

261 The vast majority of blocks are composed of a single class. However, some blocks have two  
262 comparably good classes as their predictor. For example, block 2 contains classes 3A and 3B as its 2  
263 best predictors. This suggests that the social divide between classes is less pronounced for pupils in  
264 year 3. Conversely, some classes are found to extend over two detected blocks (class 2B spans blocks  
265 6 and 7) but we nonetheless do not have a feature which explains the division. The most surprising  
266 block is number 5, which has comparable weightings for classes 5A and 1B. Perhaps there was a  
267 joint event between those two classes on the day the data were collected.

268 **5.3 Facebook egonet**

269 We choose  $B = 10$  and  $D' = 10$  for this experiment. The selected features (Figure 4c) are those  
270 that best explain the high-level community structure. The majority of them are education related.  
271 Nevertheless, for  $D' = 10$  we only have good explanations for the makeup of some of the detected  
272 blocks; several blocks in Figure 4c do not have high-magnitude components for  $D' = 10$ .

273 When the feature dimension is very large, it becomes increasingly likely that a particular feature  
274 may uniquely identify a small set of nodes. If these nodes are all part of the same community, then  
275 the classifier will overfit for that particular parameter. The regularisation term imposed by the prior  
276 goes some way towards alleviating this problem. Nevertheless, we see in Figure 4c that the feature  
277 `birthday-5` has a very high weight as it relates to block 1 – but it is highly unlikely that birthdays  
278 determine graphical structure.

279 **6 Conclusion**

280 The Feature-First Block Model (FFBM) introduced in this paper is a new generative model for  
281 labelled networks, developed in order to address difficulties of other graphical models when testing  
282 how vertex features affect community structure. The idea is to divide the graph into its most natural  
283 partition and test whether the vertex features can accurately explain this partition. It is relatively easy  
284 to find vertex features that are in some way correlated with the graphical structure. Nonetheless, only  
285 when we find the feature that best describes the most pronounced partition do we have a stronger  
286 case for causation.

287 Using this new model, we go on to describe an efficient inference algorithm to sample the parameters  
288  $\theta$  of the feature-to-block generator. This takes the form of a two-level Markov chain, used to sample  
289 the block memberships  $b$  and block generator parameters  $\theta$ . These chains can either be executed in  
290 parallel or the empirical mean of the  $b$ -samples can be used as the input to the  $\theta$ -chain. The latter  
291 option reduces the variance in our evaluation of the target distribution and thus shortens burn-in.

292 The overall method is shown to be effective at extracting and describing the most natural communities  
293 in a labelled network. Nevertheless, the approach can only currently explain the structure at the  
294 macro-scale. We cannot explain structure within each block. Future work will benefit from extending  
295 the FFBM to be hierarchical in nature. That way, the structure of the network can be explained at all  
296 length-scales of interest. So long as data collection techniques remain ethical and care is taken to  
297 respect personal privacy, such empowered decision-making can only help humankind.

298 **References**

- 299 [1] Emmanuel Abbe. Graph compression: The effect of clusters. In *2016 54th Annual Allerton  
300 Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2016. doi:  
301 10.1109/ALLERTON.2016.7852203.
- 302 [2] Edo M Airoldi, David Blei, Stephen Fienberg, and Eric Xing. Mixed membership  
303 stochastic blockmodels. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou,  
304 editors, *Advances in Neural Information Processing Systems*, volume 21. Curran As-  
305 sociates, Inc., 2009. URL <https://proceedings.neurips.cc/paper/2008/file/8613985ec49eb8f757ae6439e879bb2a-Paper.pdf>.
- 307 [3] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte  
308 Carlo computations. *The Annals of Statistics*, 37(2):697 – 725, 2009. doi: 10.1214/07-AOS574.  
309 URL <https://doi.org/10.1214/07-AOS574>.
- 310 [4] Solenne Gaucher, Olga Klopp, and Geneviève Robin. Outliers detection in networks with  
311 missing links, 2020.
- 312 [5] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications.  
313 *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. URL <http://www.jstor.org/stable/2334940>.
- 315 [6] Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego net-  
316 works. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, ed-  
317 itors, *Advances in Neural Information Processing Systems*, volume 25. Curran As-  
318 sociates, Inc., 2012. URL <https://proceedings.neurips.cc/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf>.
- 320 [7] Nikhil Mehta, Lawrence Carin Duke, and Piyush Rai. Stochastic blockmodels meet graph  
321 neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the  
322 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine  
323 Learning Research*, pages 4466–4474. PMLR, 09–15 Jun 2019. URL <http://proceedings.mlr.press/v97/mehta19a.html>.
- 325 [8] Krzysztof Nowicki and Tom A. B Snijders. Estimation and prediction for stochastic block-  
326 structures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001. doi:  
327 10.1198/016214501753208735. URL <https://doi.org/10.1198/016214501753208735>.
- 328 [9] Boris Pasternak and Ivor Ivask. Four unpublished letters. *Books Abroad*, 44(2):196–200, 1970.  
329 ISSN 00067431. URL <http://www.jstor.org/stable/40124305>.
- 330 [10] Tiago P. Peixoto. Parsimonious module inference in large networks. *Physical Review Letters*,  
331 110(14), Apr 2013. ISSN 1079-7114. doi: 10.1103/physrevlett.110.148701. URL <http://dx.doi.org/10.1103/PhysRevLett.110.148701>.
- 333 [11] Tiago P. Peixoto. Efficient monte carlo and greedy heuristic for the inference of stochastic  
334 block models. *Physical Review E*, 89(1), Jan 2014. ISSN 1550-2376. doi: 10.1103/physreve.89.  
335 012804. URL <http://dx.doi.org/10.1103/PhysRevE.89.012804>.
- 336 [12] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. doi: 10.6084/m9.figshare.  
337 1164194. URL [http://figshare.com/articles/graph\\_tool/1164194](http://figshare.com/articles/graph_tool/1164194).
- 338 [13] Tiago P. Peixoto. Nonparametric bayesian inference of the microcanonical stochastic block  
339 model. *Physical Review E*, 95(1), Jan 2017. ISSN 2470-0053. doi: 10.1103/physreve.95.012317.  
340 URL <http://dx.doi.org/10.1103/PhysRevE.95.012317>.
- 341 [14] Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions  
342 and their discrete approximations. *Bernoulli*, 2(4):341 – 363, 1996. doi: [bj/1178291835](http://dx.doi.org/10.1178291835). URL  
343 <https://doi.org/>.

- 344 [15] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton,  
345 Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems.  
346 High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*,  
347 6(8):1–13, 08 2011. doi: 10.1371/journal.pone.0023176. URL <https://doi.org/10.1371/journal.pone.0023176>.
- 349 [16] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynam-  
350 ics. In *Proceedings of the 28th International Conference on International Conference on*  
351 *Machine Learning*, ICML’11, page 681–688, Madison, WI, USA, 2011. Omnipress. ISBN  
352 9781450306195.
- 353 [17] Jun Zhu, Jiaming Song, and Bei Chen. Max-margin nonparametric latent feature models for  
354 link prediction, 2016.

355    **Checklist**

- 356    1. For all authors...
  - 357        (a) Do the main claims made in the abstract and introduction accurately reflect the paper's  
358              contributions and scope? [Yes] See experiments section 5
  - 359        (b) Did you describe the limitations of your work? [Yes] See conclusion 6 and experiments  
360              5 sections
  - 361        (c) Did you discuss any potential negative societal impacts of your work? [Yes] See  
362              conclusion 6
  - 363        (d) Have you read the ethics review guidelines and ensured that your paper conforms to  
364              them? [Yes]
- 365    2. If you are including theoretical results...
  - 366        (a) Did you state the full set of assumptions of all theoretical results? [Yes] See inference  
367              section 4
  - 368        (b) Did you include complete proofs of all theoretical results? [Yes] See inference section  
369              4
- 370    3. If you ran experiments...
  - 371        (a) Did you include the code, data, and instructions needed to reproduce the main experi-  
372              mental results (either in the supplemental material or as a URL)? [Yes] See supplemen-  
373              tary code
  - 374        (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they  
375              were chosen)? [Yes] See appendix C.2
  - 376        (c) Did you report error bars (e.g., with respect to the random seed after running experi-  
377              ments multiple times)? [Yes] See table 1
  - 378        (d) Did you include the total amount of compute and the type of resources used (e.g., type  
379              of GPUs, internal cluster, or cloud provider)? [Yes] See appendix C.3
- 380    4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
  - 381        (a) If your work uses existing assets, did you cite the creators? [Yes] See section experi-  
382              ments 5
  - 383        (b) Did you mention the license of the assets? [Yes] See section inference 4
  - 384        (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]  
385              Supplementary material
  - 386        (d) Did you discuss whether and how consent was obtained from people whose data you're  
387              using/curating? [Yes] Referred to original papers
  - 388        (e) Did you discuss whether the data you are using/curating contains personally identifiable  
389              information or offensive content? [Yes] Referred to original papers
- 390    5. If you used crowdsourcing or conducted research with human subjects...
  - 391        (a) Did you include the full text of instructions given to participants and screenshots, if  
392              applicable? [N/A]
  - 393        (b) Did you describe any potential participant risks, with links to Institutional Review  
394              Board (IRB) approvals, if applicable? [N/A]
  - 395        (c) Did you include the estimated hourly wage paid to participants and the total amount  
396              spent on participant compensation? [N/A]