Appendices

A Additional material

A.1 SBM prior choice explanation

Here we recall for reference the priors $p(\psi|b)$ from [13]:

$$p(\psi_e = e, \psi_k = k|b) = p(e|b)p(\psi_k|e,b) = \left[\left\{\!\!\left\{ {B \atop 2} \right\}\!\!\right\}^{-1} \cdot \left[\prod_r \frac{\prod_j \eta_j^r!}{n_r! q(e_r, n_r)}\right], \tag{27}$$

where $\left\{\!\!\left\{ {n \atop m} \right\}\!\!\right\}$ is shorthand for $\binom{n+m-1}{m} = \frac{(n+m-1)!}{(n-1)!(m)!}$, which can be thought of as the total number of distinct histograms produced by m samples in n bins. The value $E = \frac{1}{2}\sum_{r,s} e_{rs}$ is the total number of edges in the graph. Importantly, E is not allowed to vary and so $p(e|b)$ is uniform in e. The variable η_j^r denotes the number of vertices in block r that have degree j; formally, $\eta_j^r \coloneqq \sum_i \mathbb{1}\{b_i = r\}\,\mathbb{1}\{k_i = j\}$. The denominator $q(m,n)$ denotes the number of different histograms produced by m samples in at most n non-zero bins that sum to m. Finally, $e_r \coloneqq \sum_s e_{rs}$ is the total number of half edges in block r and $n_r \coloneqq \sum_i \mathbb{1}\{b_i = r\}$ is the number of vertices assigned to block r.

These were chosen carefully in [13] to more closely match the structure of empirical networks than simple uniform priors. We do not repeat these arguments here.

A.2 Choosing the MALA step-size

Recall that in Section 4.2 we used the Metropolis Adjusted Langevin Algorithm (MALA) to sample from the θ-chain of the block membership generator parameters. At iteration t, the proposed sample is generated by:

$$\theta' = \theta^{(t)} - h_t \nabla U(\theta^{(t)}) + \sqrt{2h_t} \cdot \xi. \tag{28}$$

There are two competing objectives when choosing the step-size h_t. On the one hand, h_t needs to be large so that the sampler arrives at a high density region quickly, while too large a step-size would lead to low acceptance rates and thus inefficient sampling. An effective strategy is to use *simulated annealing*: allow h_t to slowly decrease with t, as long as $h_t > 0$ for all t and also:

$$\sum_{t=1}^{\infty} h_t = \infty, \qquad \text{and} \qquad \sum_{t=1}^{\infty} h_t^2 < \infty. \tag{29}$$

Following Welling and Teh [16], we adopt the polynomially decaying step-sizes, $h_t = \alpha(\beta + t)^{-\gamma}$, where $\alpha > 0$, $\beta > 0$ and $\gamma \in (1/2, 1]$ are hyper-parameters. We make the specific choices,

$$\alpha = \frac{250 \cdot s}{N}, \qquad \beta = 1000, \qquad \gamma = 0.8, \tag{30}$$

where N is the number of data-points and s, the *step-size scaling*, is the only free parameter.

A.3 Burn-in and thinning

When sampling from the b and θ-chains described in Section 4, we respectively generate T_b and T_θ samples total. We discard an initial proportion $\kappa_\star \in (0,1)$ of the samples as corresponding to a "burn-in" period required for the distribution of the chain to reach a distribution close to our target, and we also "thin" the remaining samples to obtain a less-dependent version. For $\star \in \{b, \theta\}$, the remaining sample sets are denoted $\mathcal{T}_\star$ in the notation of Section 4.3,

$$\mathcal{T}_\star = \{T_\star \kappa_\star + i\lambda_\star : 0 \le i \le \lfloor T_\star(1-\kappa_\star)/\lambda_\star \rfloor\}, \tag{31}$$

where $\lambda_\star$ controls the thinning. The choice of $\kappa_\star$ can be determined by plotting the log-target (either $S(b^{(t)})$ or $U(\theta^{(t)})$ as a function of t, and choosing $\kappa_\star$ to encompass the region where the log-target has roughly reached equilibrium. As we do not leverage sample independence, $\lambda_\star$ can be chosen less rigorously; we often simply use $\lambda_b = 5$ and $\lambda_\theta = 10$.

A.4 Initializing the b-chain

For the purposes of the FFBM model, the number of blocks B is a constant which must be specified. If the choice of B is influenced by the observed data, then the analysis is no longer "fully Bayesian" and belongs to the class of methods referred to as "empirical Bayes." However, as the number of blocks only specifies the coarseness of the analysis, it is reasonable to allow it to vary. Indeed, Peixoto [10] shows that for a fixed average degree the maximum number of detectable blocks scales as $O(\sqrt{N})$ where N is the number of vertices.

If B is allowed to vary in the b-chain (i.e., when new blocks can be created and empty blocks are allowed), then the chain can be run until a minimum description length (MDL) solution is reached. We take the number of non-empty blocks in the MDL solution to be our fixed block number B for subsequent analysis. Indeed, it is prudent to start the b-chain at this MDL solution as then the necessary burn-in time can be greatly reduced.

B Derivations

B.1 Derivation of conditional block distribution given feature matrix

We determine the form of $p(b|X)$ by integrating out the parameters θ. From the definitions we have:

$$\begin{aligned} p(b|X) &= \int p(b,\theta|X,\theta)d\theta = \int p(b|X,\theta)p(\theta|X)d\theta \\ &= \int p(b|X,\theta)p(\theta)d\theta = \int \prod_{i\in[N]} \phi_{b_i}(x_i;\theta)p(\theta)d\theta \\ &= \prod_{i\in[N]} \int \frac{\exp(w_{b_i}^T \tilde{x}_i)\prod_{j\in[B]}\mathcal{N}(w_j;0,\sigma_\theta^2 I)}{\sum_{k\in[B]}\exp(w_k^T\tilde{x}_i)} dw_{1:B}. \end{aligned}$$

The key observation here is that the value of the integral is independent of the value of $b_i \in [B]$ as the integrand has the same form regardless of b_i. This is because the prior is the same for each w_j. Therefore, the integral is only a function of $\tilde{x}_i$ and σ_θ^2, which means that, as a function of b, $p(b|X) \propto 1$. As b takes values in $[B]^N$, we necessarily have:

$$p(b|X) = \frac{1}{\left|[B]^N\right|} = B^{-N}. \tag{32}$$

B.2 Derivation of $U(\theta)$

Recall from (15) in Section 4.2 that,

$$\pi_\theta(\theta) \propto p(\theta|X,b) \propto p(b|X,\theta)p(\theta) \propto \exp\left(-U(\theta)\right),$$

so that U can be expressed as,

$$U(\theta) = -\left(\log p(b|X,\theta) + \log p(\theta)\right) + \text{const.}$$

Writing, $y_{ij} \coloneqq \mathbb{1}\{b_i = j\}$ and $a_{ij} \coloneqq \phi_j(x_i;\theta)$, we have that,

$$\log p(b|X,\theta) = \sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\log a_{ij} \quad \text{and} \quad \log p(\theta) = -\frac{(D+1)(B)}{2}\log 2\pi - \frac{1}{2\sigma_\theta^2}\|\theta\|^2, \tag{33}$$

where $\|\theta\|^2 = \sum_i \theta_i^2 = \sum_{j\in[B]} \|w_j\|^2$ is the Euclidean norm of the vector of parameters θ. Therefore, discarding constant terms, we obtain exactly the representation (16), as claimed.

B.3 Derivation of $\nabla U(\theta)$

Here we show how the gradient $\nabla U(\theta)$ can be computed explicitly. Recall the expression for $U(\theta)$ in (16). Writing θ as $\theta = \left[w_1^T, w_2^T \ldots w_B^T\right]^T$, in order to compute the gradient $\nabla U(\theta)$ we need to

compute each of its components, $\partial U/\partial w_k$, $1 \leq k \leq B$. To that end, we first compute,

$$\begin{aligned}\frac{\partial a_{ij}}{\partial w_k} &= \frac{\tilde{x}_i \exp(w_j^T \tilde{x}_i)\delta_{jk} \cdot \sum_{r\in[B]} \exp(w_r^T \tilde{x}_i) - \exp(w_j^T \tilde{x}_i) \cdot \tilde{x}_i \exp(w_k^T \tilde{x}_i)}{\left(\sum_{r\in[B]} \exp(w_r^T \tilde{x}_i)\right)^2} \\ &= \tilde{x}_i \left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right), \end{aligned} \tag{34}$$

where $\delta_{jk} \coloneqq \mathbb{1}\{j = k\}$, and we also easily find,

$$\frac{\partial}{\partial w_k}\|\theta\|^2 = \frac{\partial}{\partial w_k}\left(\sum_{r\in[B]} \|w_r\|^2\right) = 2w_k. \tag{35}$$

Using (34) and (35), we obtain,

$$\begin{aligned}\frac{\partial U}{\partial w_k} &= \sum_{i\in[N]}\sum_{j\in[B]} y_{ij}\left(-\frac{\tilde{x}_i}{a_{ij}}\left(a_{ij}\delta_{jk} - a_{ij}a_{ik}\right)\right) + \frac{w_k}{\sigma_\theta^2} \\ &= -\left(\sum_{i\in[N]} \tilde{x}_i \left(y_{ik} - a_{ik}\sum_{j\in[B]} y_{ij}\right) - \frac{w_k}{\sigma_\theta^2}\right) \\ &= -\left(\sum_{i\in[N]} \left\{\tilde{x}_i(y_{ik} - a_{ik})\right\} - \frac{w_k}{\sigma_\theta^2}\right). \end{aligned} \tag{36}$$

This can be computed efficiently through matrix operations. The only property of y_{ij} we have used in the derivation is the constraint $\sum_{j\in[B]} y_{ij} = 1$, for all i.

C Computational details

C.1 Algorithms

Algorithm 1 Block membership sample generation

procedure SAMPLEBLOCKMEMBERSHIPS(A, T_b)
  $b^{(0)} \leftarrow \arg\min_b S(b|A)$ ▷ Implemented as greedy heuristic in *graph-tool* library
  for $t \in \{0, 1 \dots T_b - 1\}$ **do**
    $b' \leftarrow\sim q_b(b^{(t)}, b'|A)$
    $\log \alpha_b \leftarrow \log \alpha_b(b^{(t)}, b'|A)$
    $\eta \leftarrow\sim \text{Unif}(0, 1)$
    if $\log \eta < \log \alpha_b$ **then**
      $b^{(t+1)} \leftarrow b'$
    else
      $b^{(t+1)} \leftarrow b^{(t)}$
    end if
  end for
  return $\{b^{(t)}\}_{t=1}^{T_b}$
end procedure

Algorithm 2 FFBM parameter pseudo-marginal inference

procedure SAMPLEFEATUREWEIGHTS($X, \{b^{(t)}\}, \mathcal{T}_b, \sigma_\theta, s$)
  $\hat{Y}_{ij} \leftarrow \frac{1}{|\mathcal{T}_b|} \sum_{t \in \mathcal{T}_b} \mathbb{1}\{b_i^{(t)} = j\} \quad \forall i, j$
  $\theta^{(0)} \leftarrow\sim \mathcal{N}(0, \sigma_\theta I)$

  for $t \in \{0, 1 \dots T_\theta - 1\}$ **do**
    $\xi \leftarrow\sim \mathcal{N}(0, I)$
    $h_t \leftarrow \frac{s}{N} \cdot 250(1000 + t)^{-0.8}$
    $g_t \leftarrow \nabla U(\theta^{(t)}|X, \hat{Y})$

    $\theta' \leftarrow \theta^{(t)} - h_t \cdot g_t + \sqrt{2h_t} \cdot \xi$
    $\log \alpha_\theta \leftarrow \log \alpha_\theta(\theta^{(t)}, \theta'|A, \hat{Y})$
    $\eta \leftarrow\sim \text{Unif}(0, 1)$
    if $\log \eta < \log \alpha_\theta$ **then**
      $\theta^{(t+1)} \leftarrow \theta'$
    else
      $\theta^{(t+1)} \leftarrow \theta^{(t)}$
    end if
  end for
  return $\{\theta^{(t)}\}_{t=1}^{T_\theta}$
end procedure

Algorithm 3 Dimensionality reduction

procedure REDUCEDIMENSION($\{W^{(t)}\}, \mathcal{T}_\theta, k, D'$)
  $(B, D) \leftarrow W^{(0)}$.shape
  $\hat{\mu}_{ij} \leftarrow \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} W_{ij}^{(t)} \quad \forall i \in [B], j \in [D]$
  $\hat{\sigma}_{ij} \leftarrow \frac{1}{|\mathcal{T}_\theta|} \sum_{t \in \mathcal{T}_\theta} \left(W_{ij}^{(t)} - \hat{\mu}_{ij} \right)^2 \quad \forall i \in [B], j \in [D]$

  for $d \in [D]$ **do**
    for $i \in [B]$ **do**
      $l_i \leftarrow \hat{\mu}_{id} - k \cdot \hat{\sigma}_{id}$
      $u_i \leftarrow \hat{\mu}_{id} + k \cdot \hat{\sigma}_{id}$
      if $l_i \leq 0$ and $u_i \geq 0$ **then**
        $l_i, u_i \leftarrow 0$
      end if
    end for
    $c_d \leftarrow \max_i \min(|l_i|, |u_i|)$
  end for

  indexArray $\leftarrow$ indexSort(c, descending=True)$[0 : D']$
  $d^* \leftarrow$ indexArray[-1]
  $\mathcal{D}' \leftarrow$ Set(indexArray)
  $c^* \leftarrow c_{d^*}$
  return $\mathcal{D}', c^*$
end procedure

C.2 Hyperparameter values

Table 2: Hyper-parameter values for each experiment.

Dataset	B	f	σ_θ	T_b	κ_b	λ_b	T_θ	κ_θ	λ_θ	s	k	D'	T'_θ	κ'_θ	λ'_θ	s'
Polbooks	3	0.7	1	1,000	0.2	5	10,000	0.4	10	0.05	–	–	–	–	–	–
School	10	0.7	1	1,000	0.2	5	10,000	0.4	10	0.2	1	10	10,000	0.4	10	0.2
FB Egonet	10	0.7	1	1,000	0.2	5	10,000	0.4	10	0.017	1	10	10,000	0.4	10	0.5

C.3 Implementation details

All data analysis and visualisation was implemented in Python. Full source code is available in the supplementary material. The scripts were run using a standard PC using the Windows Subsystem for Linux (WSL) environment. Specs are:

- **CPU**: Intel(R) Core(TM) i7-1065G7
- **RAM**: 8GB
- **GPU**: Intel(R) Iris(R) Plus Graphics

On this hardware each experiment iteration took the following amount of time to execute:

Table 3: Compute-time for each experiment

Dataset	b-chain	θ-chain	Reduced θ-chain	Overall compute time
Polbooks	$\sim$1s	$\sim$10s	–	$\sim$11s
School	$\sim$1s	$\sim$7s	$\sim$7s	$\sim$15s
FB Egonet	$\sim$2s	$\sim$50s	$\sim$8s	$\sim$60s