A Supplementary Material

A.1 Automatic Relevance Determination and Kernel

In this work we use the exponentiated quadratic (also known as the “squared exponential”) ARD kernel:

\[ K(x, x') = \prod_{r=1}^{R} \exp \left( -\frac{(x_r - x'_r)^2}{2\alpha_r} \right). \]  

(18)

A.2 Derivation of lower bound

The lower bound Equation 6 is derived as follows:

\[
\log p(\mathcal{D}_{1:S}, A_{1:S} | \Theta) = \log \left[ \int \prod_{t} \int dp(f_t | u_t) q(u_t) \, du_t \, \log [p(\mathcal{D}_{1:S}, A_{1:S} | f_{1:T})] \right] \\
= \int \prod_{t} \int dp(f_t | u_t) q(u_t) \, du_t \, \log \left[ \frac{p(u_{1:T})}{q(u_{1:T})} \right] \\
= E_{q(f_{1:T})} [\log p(\mathcal{D}_{1:S}, A_{1:S} | f_{1:T})] - KL(q(u_{1:T}) \parallel p(u_{1:T})) \\
\triangleq \mathcal{L}(\mathcal{D}_{1:S}, A_{1:S}; \Theta).
\]

(19)

A.3 Definition of KL

The KL term in Equation 6 is the Kullback–Leibler divergence between \( T \) pairs of independent Gaussians distribution and is defined by:

\[
KL(q(u_{1:T}) \parallel p(u_{1:T})) = \frac{1}{2} \sum_{t} \left[ \text{tr} \left( K_{zz}^{-1} S_t \right) + (\bar{u}_t - \mu)^T K_{zz}^{-1} (\bar{u}_t - \mu) - M + \log \left| K_{zz}^{-1} \right| \right].
\]

(20)

A.4 Definition of \( \tilde{G} \)

The function \( \tilde{G} \) that appears in the expectation \( E_{q(f_{1:T})} [\log f_{s,t,n}^2] = \int_{-\infty}^{\infty} \log(f_{s,t,n}^2) \mathcal{N}(f_{s,t,n}; \bar{f}_{s,t,n}, \sigma_{s,t,n}^2) \, df_{s,t,n} \), Equations 9, is a specialised version of the partial derivative of the confluent hyper-geometric function,

\[
i F_1(a, b, z) = \sum_{k=0}^{\infty} \frac{a_k z^k}{b_k k!},
\]

(21)

with respect to its first argument and is defined by:

\[
\tilde{G}(z) = i F_1^{(1,0,0)} \left( 0, \frac{1}{2}, z \right) = 2z \sum_{j=0}^{\infty} \frac{j! z^j}{(2j)(1/2)^{2j}},
\]

(22)

where \((\cdot)_j\) denotes the rising Pochhammer series \((a)_0 = 1, (a)_j = a(a+1)(a+2) \ldots (a+j-1)\).

A.5 Definition of \( \Psi_{zz} \)

For the ARD Kernel the function \( \Psi(z, z') = \int_{x'} K(z, x)K(x, z') \, dx \) can be computed in closed form:

\[
\Psi(z, z') = \prod_{r=1}^{R} \sqrt{\frac{\pi \alpha_r}{2}} \exp \left( -\frac{(z_r - z'_r)^2}{4\alpha_r} \right) \left[ \text{erf} \left( \frac{z_r - \tilde{z}_r}{\sqrt{\alpha_r}} \right) - \text{erf} \left( \frac{z_r - \tilde{z}_r}{\sqrt{\alpha_r}} \right) \right],
\]

(23)

where \( \tilde{z}_r = \frac{1}{2}(z_r + z'_r) \).
A.6 Detailed Derivation of the Collapsed Bound

The set of all possible assignments is:
\[ \{ A_S^{(1)} = 1, \ldots, A_S^{(N_s)} = 1 \}, \ldots, \{ A_S^{(1)} = T, \ldots, A_S^{(N_s)} = T \} \],

In the collapsed bound we sum over all the possible assignments to each of the allocation variables:
\[
\log p(\mathcal{D}_{1:S} | \Theta) = \log \sum_{A_{1:S}} p(\mathcal{D}_{1:S}, A_{1:S} | \Theta) \\
\geq \log \sum_{A_{1:S}} \exp (\mathcal{L}(\mathcal{D}_{1:S}, A_{1:S} | \Theta)) \\
= \log \sum_{A} \exp \left( \mathfrak{B} + \sum_{a} \sum_{n} \sum_{t} \mathbb{1}\{A_{a}^{(n)} = t\} A_{s,t,n} \right) \\
= \log \left[ \exp (\mathfrak{B}) \times \prod_{a} A_{a}^{(n)} \right] \\
= \log \left[ \exp (\mathfrak{B}) \times \prod_{a} A_{a}^{(n)} \right] \\
= \mathfrak{B} + \sum_{a} \sum_{n} \log \sum_{t} \exp A_{s,t,n} \\
\triangleq \mathcal{L}(\mathcal{D}_{1:S} | \Theta)
\]  

A.7 Benchmark

The benchmark kernel smoother optimises the leave-one-out training objective:
\[ \Sigma_* = \arg\max \Sigma \sum_{i=1}^{N_s} \log \sum_{j \neq i}^{N_s} \mathcal{N}_\Sigma(x^{(s,i)}; x^{(s,j)}, \Sigma). \]

We can construct the test log-likelihood for the held-out datasets as:
\[
\log p(\mathcal{D}_{1:S} | \mathcal{D}_{1:S}, \Sigma_* | \Theta) = \sum_{s=1}^{S} \sum_{n=1}^{N_s} \log \sum_{t=1}^{T} \sum_{a} a_{s,t} \ln b_{t,m(h,n)} - |\Delta x| \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{b=1}^{B} a_{s,t} \ln b_{t,b}
\]

where \( m(h,n) \) is a function that maps a test data point \( \tilde{x}^{(h,n)} \) into the \( d^{th} \) grid-cell. For the ct case the weight matrix \( A \) is optimised for the test data.

A.8 Mixed Continuous Discrete Co-ordinate Spaces

This \( \Psi \)-function in the mixed co-ordinate space case is \( \Psi(z_r, z'_r) = \sum_{r} K(z_r, x_r)K(x_r, z_r) \). When using Kronecker structure \( \Psi_{z_2 z_2} \) is simply \( K_{z_2 z_2} K_{z_2 z_2} \) if \( Z \) contains all feeding station locations and the discrete dimension is \( r = 2 \).

A.9 Adapting LPPA to Model Dynamic Interaction Networks

LPPA can be used to model dynamic pairwise interactions between \( V \) nodes, where is each sender \( i \) and receiver \( j \) is associated with a set of observations \( \{ D_{ij} \}_{i,j=1} \) and a rate functions \( \lambda_{i,j} \). A straight forward approach is a triple factorisation typical of network models (Schmidt and Morup, 2013). Each rate function is constructed as \( \lambda_{i,j} = \sum_{w=1}^{C} \sum_{c=1}^{C} \Omega_{i,v} f_{c,w} \Omega_{j,w} \), where \( C \) is the number of “communities”.

To modify LPPA we simply need to map \( D_{ij} \) and \( \lambda_{i,j} \) to \( D_s \) and \( \lambda_s \), to map \( f_{c,w} \) to \( f_{c} \) and compute \( \gamma_{s,t} \) from \( \Omega_{i,v} \) and \( \Omega_{j,w} \). These mappings will be different depending on whether we wish to model a symmetric network with \( D_{ij} = D_{ji} \), and/or a network in which reflexive interaction is by definition empty, i.e. \( D_{ii} = 0 \), thus making no contribution to the likelihood.

Since the cost of this algorithm increases quadratically as \( P = C^2 \), we might also consider a simpler model in which only intra-community interaction is allowed. In this case we model the rate function as \( \lambda_{i,j} = \sum_{t} \Omega_{i,t} f_{t} \Omega_{j,t} \) for symmetric networks, or \( \lambda_{i,j} = \sum_{t} \Omega_{i,t} f_{t} Y_{j,t} \) where asymmetry is introduced via a third factor \( Y \).