MANCHESTER 1824
The University of Manchester

Funded by the European Union

European Research Council
Established by the European Commission

Joint Bayesian calibration and map-making for intensity mapping experiments

Zheng Zhang

Jodrell Bank Centre for Astrophysics, Department of Physics
University of Manchester
Manchester, UK

JBCA Colloquium, Manchester, 28 May 2025

What this talk is not

Not an end-to-end Bayesian model - by design

Reasons

- Component separation is not necessary/essential for gain/noise calibration
- No reliable ground-truth sky model $\rightarrow$ model dependence becomes a liability

Not: "*We model this way to minimise degeneracy*"

Instead

- Model ***degeneracies*** when they exist
- Model ***ignorance*** with general, safe representations
- Stay true to what we exactly know

What this talk is about

A Bayesian conservative's framework for calibration and map-making

Timestream of Data

System Temperature

$$d(t_a) = \boxed{G(t_a)}\,\boxed{T_{\rm sys}(t_a)} \left(1 + \hat{w}\right),$$

Instrumental gain

White noise: $\hat{w} \sim \mathcal{N}(0, 1/(\mathcal{T}\Delta\nu))$

Key ingredients

- Instrumental gain modelling (smooth component & $1/f$ noise)
- Multiplicative noise treatment (no noise model conditionals)
- Joint inference of smooth gain, $1/f$ noise, and system temperature components
- Experimental setup for demonstration: MeerKAT-type observation in mind

Instrumental Gain Modelling

1/*f* noise
(stochastic fractional gain)

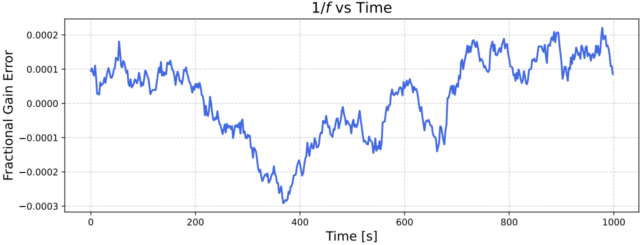
$$G(t_a) = \boxed{g(t_a)}\,(1 + \boxed{\hat{\epsilon}})$$

Smooth gain

$$g(t_a) = \sum_{n=0}^{4} a_n\, P_n(x), \qquad x = \frac{2(t_a - t_{\min})}{t_{\max} - t_{\min}} - 1.$$

Flicker Noise

$$P_{\hat{\epsilon}}(f) = \begin{cases} 0, & |f| < f_c, \\ \left(f_0/|f|\right)^{\alpha}, & |f| \geq f_c. \end{cases}$$



Stochastic Gain Model

1/*f* noise
(stochastic fractional gain)

$$G(t_a) = \boxed{g(t_a)}\,(1 + \boxed{\hat{\epsilon}})$$

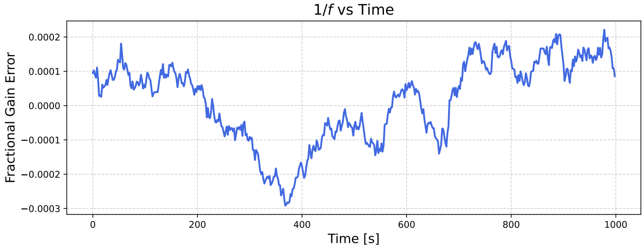
Smooth gain

$$g(t_a) = \sum_{n=0}^{4} a_n\, P_n(x), \qquad x = \frac{2(t_a - t_{\min})}{t_{\max} - t_{\min}} - 1.$$

Flicker Noise
(Complete Model)

Gaussianity and stationarity.

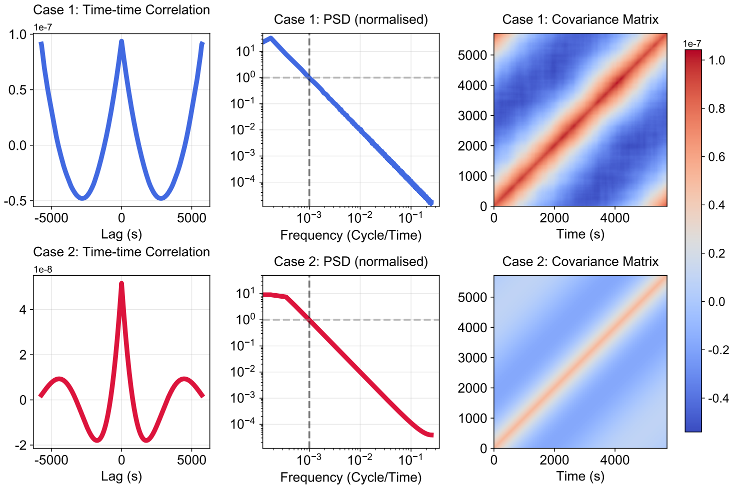
$$P_{\hat{\epsilon}}(f) = \begin{cases} 0, & |f| < f_c, \\ \left(f_0/|f|\right)^{\alpha}, & |f| \geq f_c. \end{cases}$$



Stochastic Gain Model

But how to implement the $1/f$ noise covariance?

- **Case 1:**
  Diagonal in DFT space
- **Case 2:**
  Diagonal in CFT space



Case 1: Diagonal in FFT space; Case 2: Diagonal in CFT space

Stochastic Gain Model

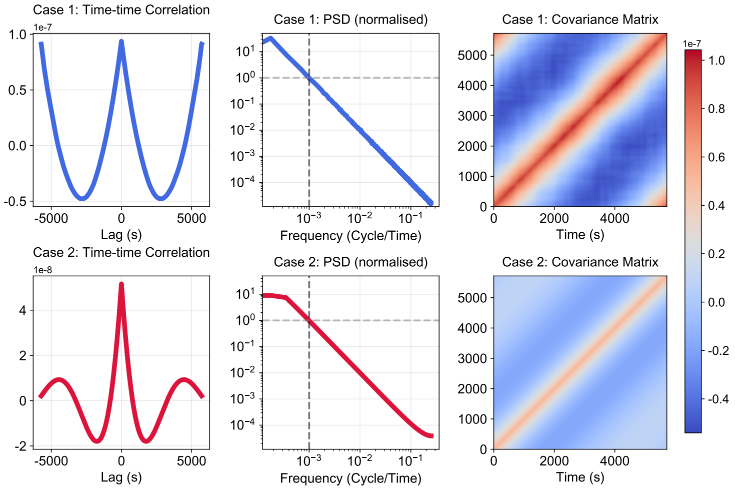
But how to implement the $1/f$ noise modelling?

- **Case 2:** Diagonal in CFT space

$$(\mathbf{N}_{\mathrm{corr}})_{aa'} = \xi(t_{a'} - t_a)$$

$$\xi(\tau) = \frac{\Theta_0^{\alpha}}{\pi\tau}\mathrm{Re}\left[\Gamma(\mu, i\Theta_c)\, e^{-i\frac{\pi}{2}\mu}\right]$$

$$\mu = 1 - \alpha, \Theta_0 = \tau f_0, \Theta_c = \tau f_c$$



Case 1: Diagonal in FFT space; Case 2: Diagonal in CFT space

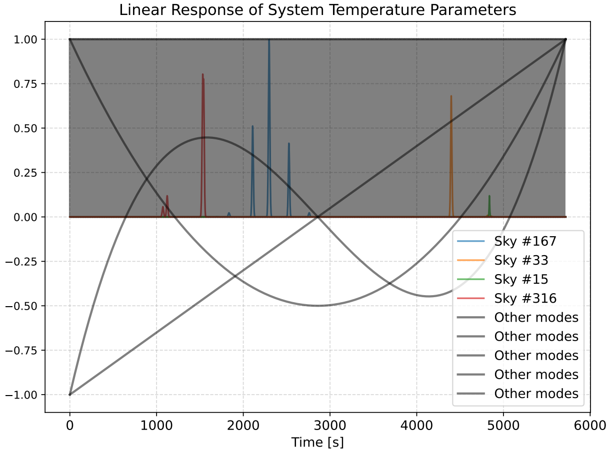
System Temperature Model

Sky + Receiver + Ground + Atmospheric + ...

$$T^{\mathrm{sys}}_{j,p}(t_a) = \sum_X B^X_p(t_a) * T^{\mathrm{sky},X} + T^{\mathrm{el}}_p(t_a) + T^{\mathrm{nd}}_{j,p}(t_a) + T^{\mathrm{rec}}_{j,p}$$

Linear Model

$$\vec{T}_{\mathrm{sys}} = \mathbf{U}_{\mathrm{ce}}\, \vec{p}_{\mathrm{ce}} + \mathbf{U}_{\mathrm{res}}\, \vec{p}_{\mathrm{res}}$$



Joint Bayesian Analysis Framework

- The data model can be rewritten as

$$d(t_a) \simeq g(t_a)\,T_{\rm sys}(t_a)\,[1 + \hat{w} + \hat{\epsilon}]\,.$$

- the likelihood function of the data model is given by

$$L(\boldsymbol{n}|g, T_{\rm sys}, \mathbf{N}) = (2\pi)^{-\frac{N}{2}}|\mathbf{N}|^{-\frac{1}{2}}\exp\left\{\left[-\frac{1}{2}\boldsymbol{n}^T\mathbf{N}^{-1}\boldsymbol{n}\right]\right\},$$

  where $\mathbf{N} = \mathbf{N}_w + \mathbf{N}_{\rm corr}$.

- Gibbs sampling steps:

$$g^{(i+1)} \leftarrow \mathcal{P}_{\rm post}(g|\boldsymbol{d}, T_{\rm sys}^{(i)}, \mathbf{N}^{(i)})$$
$$\mathbf{N}^{(i+1)} \leftarrow \mathcal{P}_{\rm post}(\mathbf{N}|\boldsymbol{d}, T_{\rm sys}^{(i)}, g^{(i+1)})$$
$$T_{\rm sys}^{(i+1)} \leftarrow \mathcal{P}_{\rm post}(T_{\rm sys}|\boldsymbol{d}, g^{(i+1)}, \mathbf{N}^{(i+1)})$$

Tractable Implementation for Gain and Tsys samplers

Iterative GLS sampler

- Conditional on noise, the data model can be generally written as

$$\boldsymbol{d} = (\mathbf{U}\boldsymbol{p} + \mu) \circ (1 + \boldsymbol{n}).$$

- Rearrangement - effective additive form:

$$\boldsymbol{d}' = \mathbf{U}\boldsymbol{p} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \Sigma),$$

where $\boldsymbol{d}' = \boldsymbol{d} - \mu$ and the noise covariance structure is given by:

$$\Sigma = \mathrm{Diag}(\mathbf{U}\boldsymbol{p}+\mu)\,\mathbf{N}\,\mathrm{Diag}(\mathbf{U}\boldsymbol{p}+\mu).$$

Estimating Σ with iterative GLS:

1. Initialize $\boldsymbol{p}^{(0)}$ using OLS:

$$\boldsymbol{p}^{(0)} = (\mathbf{U}^\top \mathbf{U})^{-1}\mathbf{U}^\top \boldsymbol{d}'.$$

2. At iteration k, compute

$$\Sigma^{(k)} = \mathrm{diag}(\mathbf{U}\boldsymbol{p}^{(k)} + \mu)\,\mathbf{N}\,\mathrm{diag}(\mathbf{U}\boldsymbol{p}^{(k)} + \mu).$$

3. Update $\boldsymbol{p}$ by solving:

$$\left[\mathbf{U}^\top \left(\Sigma^{(k)}\right)^{-1} \mathbf{U}\right] \boldsymbol{p}^{(k+1)} = \mathbf{U}^\top \left(\Sigma^{(k)}\right)^{-1} \boldsymbol{d}'.$$

4. Repeat until convergence:

$$\|\boldsymbol{p}^{(k+1)} - \boldsymbol{p}^{(k)}\| < \mathrm{tol}$$

Tractable Implementation for Gain and Tsys samplers

Gaussian Constrained Realisation (GCR) equations

Per TOD set sampling:

$$\left(\mathbf{C}^{-1} + \mathbf{U}^{\top}\Sigma^{-1}\mathbf{U}\right)\boldsymbol{p}_{\mathrm{sample}} = \mathbf{U}^{\top}\left(\Sigma^{-1}\boldsymbol{d}' + \Sigma^{-\frac{1}{2}}\omega\right) + \mathbf{C}^{-1}\bar{\boldsymbol{p}} + \mathbf{C}^{-\frac{1}{2}}\eta,$$

Multiple TOD sets joint fitting:

$$\left(\mathbf{C}^{-1} + \sum_{j}\mathbf{U}_j{}^{\top}\Sigma_j^{-1}\mathbf{U}_j\right)\boldsymbol{p}_{\mathrm{sample}} = \sum_{j}\mathbf{U}_j{}^{\top}\left[\Sigma_j^{-1}\boldsymbol{d}_j' + \Sigma_j^{-\frac{1}{2}}\omega_j\right] + \mathbf{C}^{-1}\bar{\boldsymbol{p}} + \mathbf{C}^{-\frac{1}{2}}\eta,$$

Tractable Implementation for Sampling $1/f$ Parameters

CoMat: An $O(n^2)$ implementation

$$\mathbf{N}_{\text{corr}} = \begin{pmatrix} \xi_0 & \xi_1 & \xi_2 & \cdots & \xi_{n-1} \\ \xi_1 & \xi_0 & \xi_1 & \cdots & \xi_{n-2} \\ \xi_2 & \xi_1 & \xi_0 & \cdots & \xi_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \xi_{n-1} & \xi_{n-2} & \xi_{n-3} & \cdots & \xi_0 \end{pmatrix}$$

- Recursive Determinant

$$\det \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(A)\det(D-CA^{-1}B).$$

- Levinson-Durbin Recursion

$$\mathbf{T}^n \begin{bmatrix} 0 \\ \vec{b}^{n-1} \end{bmatrix} = \begin{bmatrix} t_0 & \dots & t_{-n+2} & t_{-n+1} \\ \vdots & & & \\ t_{n-2} & & \mathbf{T}^{n-1} & \\ t_{n-1} & & & \end{bmatrix} \begin{bmatrix} 0 \\ \\ \vec{b}^{n-1} \end{bmatrix} = \begin{bmatrix} \epsilon_b^n \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

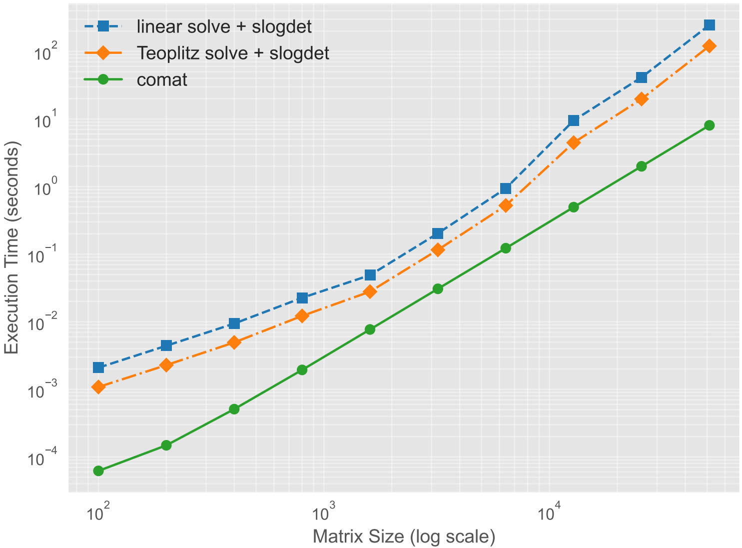
- Perturbative Scenario

$$(\Lambda + \mathbf{P})^{-1} \simeq \Lambda^{-1} - \sum_{k=0}^{m-1} \Lambda^{-1}\mathbf{P}\left[-\mathbf{P}\Lambda^{-1}\right]^k \Lambda^{-1}.$$

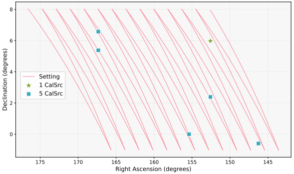
$$\ln[\det(\Lambda + \mathbf{P})] \approx \sum_{k=0}^{m-1} \frac{(-1)^k}{k+1} \operatorname{Tr}\left[\left(\mathbf{P}\Lambda^{-1}\right)^{k+1}\right] + \ln[\det(\Lambda)]$$

Tractable Implementation for Sampling $1/f$ Parameters

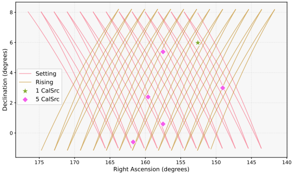
CoMat



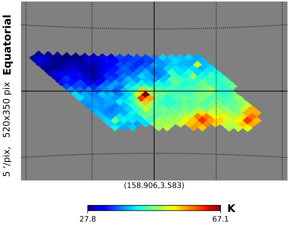
Illustrating Examples
Experimental Setup



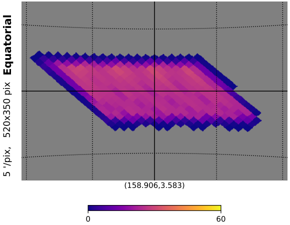
(a) $1\times$**TOD**: Scan



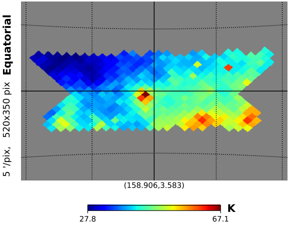
(b) $2\times$**TOD**: Scan



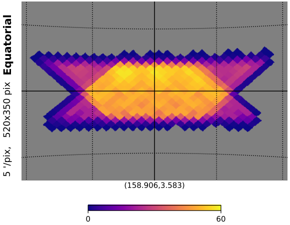
(a) $1\times$**TOD**: Sky



(a) $1\times$**TOD**: Integrated beam.



(b) $2\times$**TOD**: Sky



(b) $2\times$**TOD**: Integrated beam.

Illustrating Examples
Prior Setup

Smooth Gain Prior

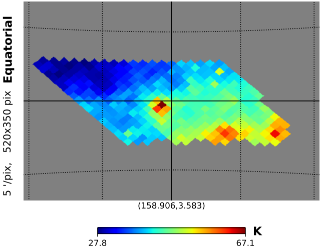
- **DC mode:**
  Prior STD $\sim$ 20%
- **Other modes:**
  Flat priors

$1/f$ Prior

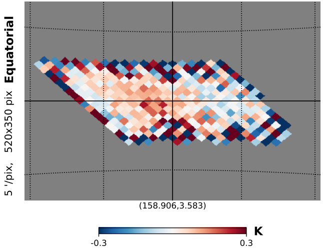
- **Power law index:**
  Flat prior
- **Reference (Knee) frequency:**
  Flat prior

System Temperature Prior (Rough Sky Knowledge + Cal. Src.)

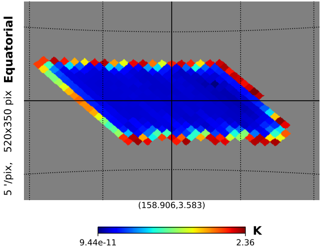
- Prior STD per pixel: 10 K
- Flux scale calibration:
  - Sharp prior on one pixel (“1 CalSrc”)
- Other components (receiver temperature, ground pickup, etc)
  - **No prior!** *If you choose to model ignorance, that’s because ignorance is your prior!*



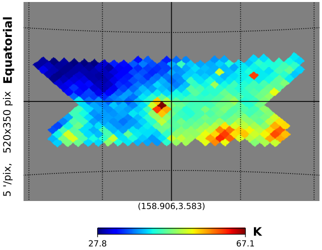
(a) Estimated (1 × TOD; 1 CalSrc)



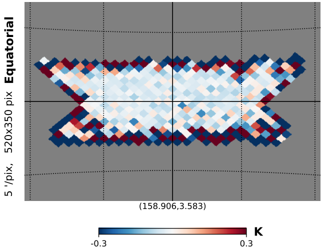
(a) Residual (1 × TOD; 1 CalSrc)



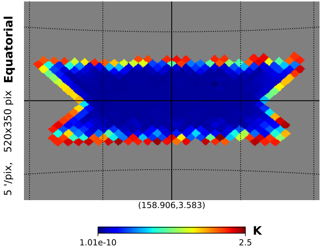
(a) Uncertainty (1 × TOD; 1 CalSrc)



(b) Estimated (2 × TOD; 1 CalSrc)



(b) Residual (2 × TOD; 1 CalSrc)



(b) Uncertainty (2 × TOD; 1 CalSrc)

An illustrating Example

Histogram of the residuals across pixels

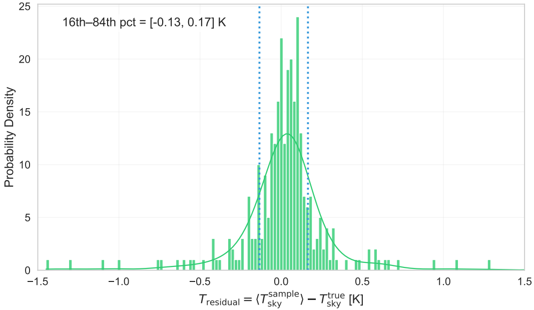


Figure: 1×TOD; 1 CalSrc

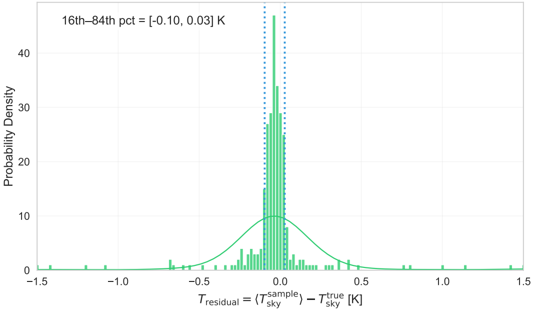


Figure: 2×TOD; 1 CalSrc

CONCLUSION & DISCUSSION

- CONSIDER MODEL IGNORANCE:
  Beyond end-to-end modelling with educated guesses, it's also valuable to explicitly model our ignorance—identifying and parameterising what we don't know with safety redundancy.
- LEVERAGE INTERNAL CONSTRAINTS:
  Strategically design experiments to leverage internal constraints in data models to achieve distinguishability between different components.
- THINK LIKE THE TELESCOPE:
  MeerKAT-like observations can distinguish system temperature components by their differing temporal behaviors—use this perspective to guide modelling.
- BAYESIAN MODELLING WITH IGNORANCE SHOULD EVOLVE:
  Building a pipeline is only the beginning—learning and adapting both the prior and the model itself is essential as understanding deepens.

Echoes from Earlier Sessions I

Ad: SpyDust

- an improved and extended Python implementation for modelling spinning dust emission

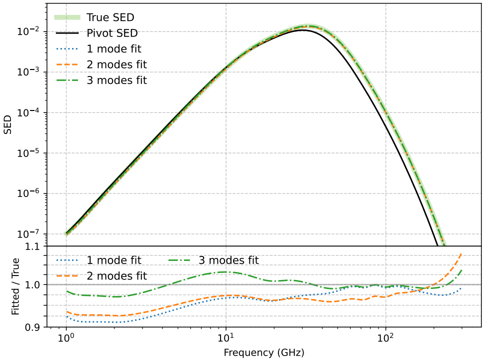
  Zhang & Chluba JCAP03(2025)038

- A full Stokes implementation with reduced dimensionality (via moment expansion)

  Zhang & Chluba in prep.

- Improved Fokker-Planck treatment - better rotational statistics

  Zhang & Chluba in prep.



Echoes from Earlier Sessions II

“redshift-space distortions” (RSD)

This is CORRECT!

$$P_s(k,\mu) = (1+\beta^2\mu(k)^2)^2 P_r(k)$$

- Large scale, but smaller than the curvature scale: Cartesian plane wave synthesis
- $\mu(k)$: defined with the normal mode peculiar velocity
- Scalar perturbations - $\vec{u}(\vec{k}) \parallel \vec{k}$
- $\mu(k)$: NOT directly given by LOS projection of real space peculiar velocity