# Performance Analysis of Distributed Radio Interferometric Calibration

Sarod Yatawatta

ASTRON

The Netherlands Institute for Radio Astronomy,

Dwingeloo, The Netherlands

in collaboration with

Netherlands eScience Center,

Science Park, Amsterdam, The Netherlands

(project DIRAC, grant 27016G05).

# Introduction

□ Calibration of radio telescopes: essential for correcting systematic errors (beam,ionosphere), removal of strong contaminating signals (foregrounds): for high quality imaging.

$$g_f(\mathsf{J}_f) = \sum_{p,q} \|\mathsf{V}_{pqf} - \mathsf{A}_p\mathsf{J}_f\mathsf{C}_{pqf}(\mathsf{A}_q\mathsf{J}_f)^H\|^2$$

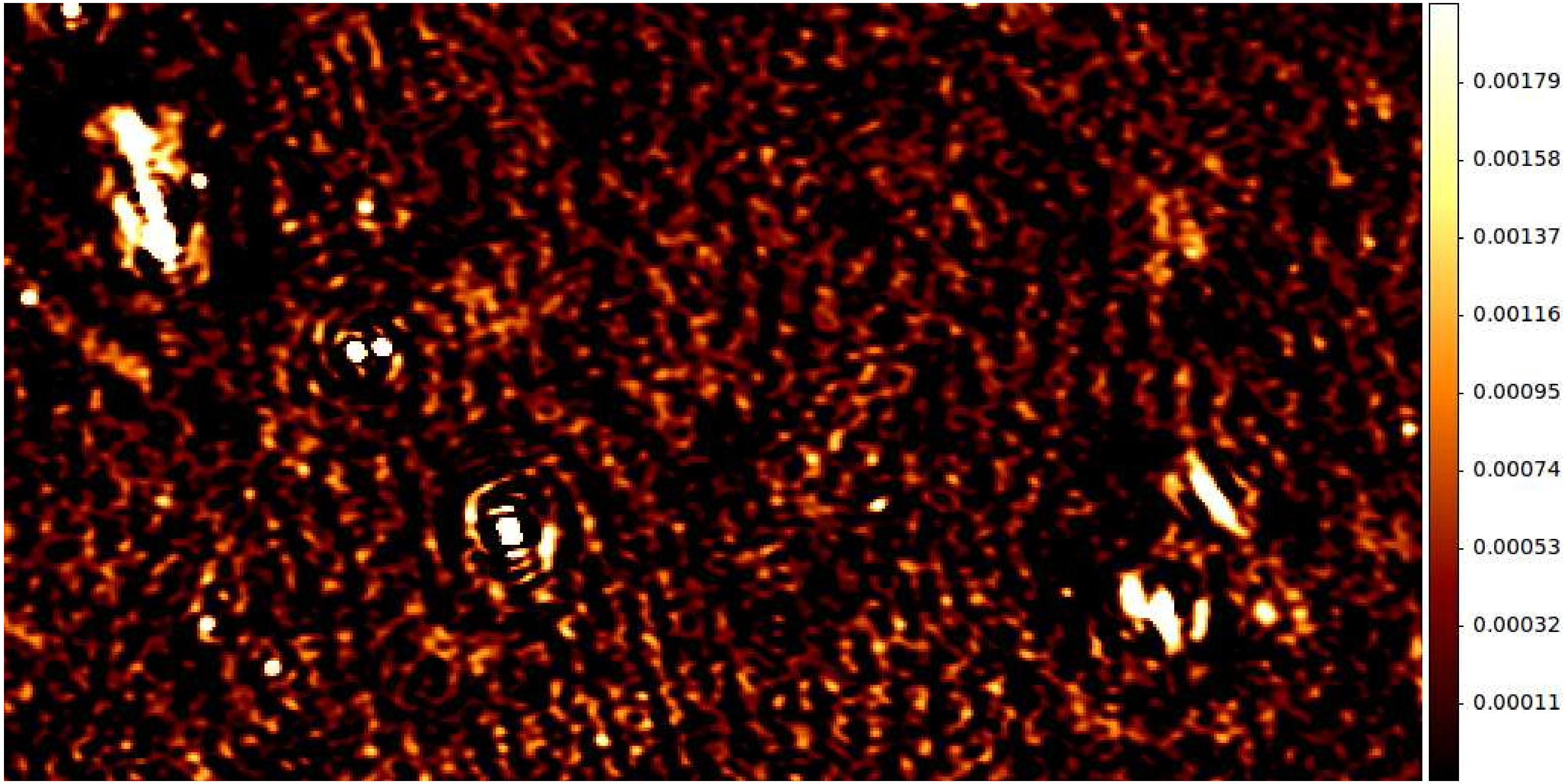
$\mathsf{V}_{pqf}$ : data, $\mathsf{C}_{pqf}$ : model, $\mathsf{J}_f$ : parameters, at frequency $f$.

□ Terabytes of data observed, data split into thousands of frequency channels, also stored at different locations in a network.

□ Distributed calibration (and imaging): enable the use of many compute agents to calibrate data faster and better.

□ How can we measure the performance of distributed calibration?

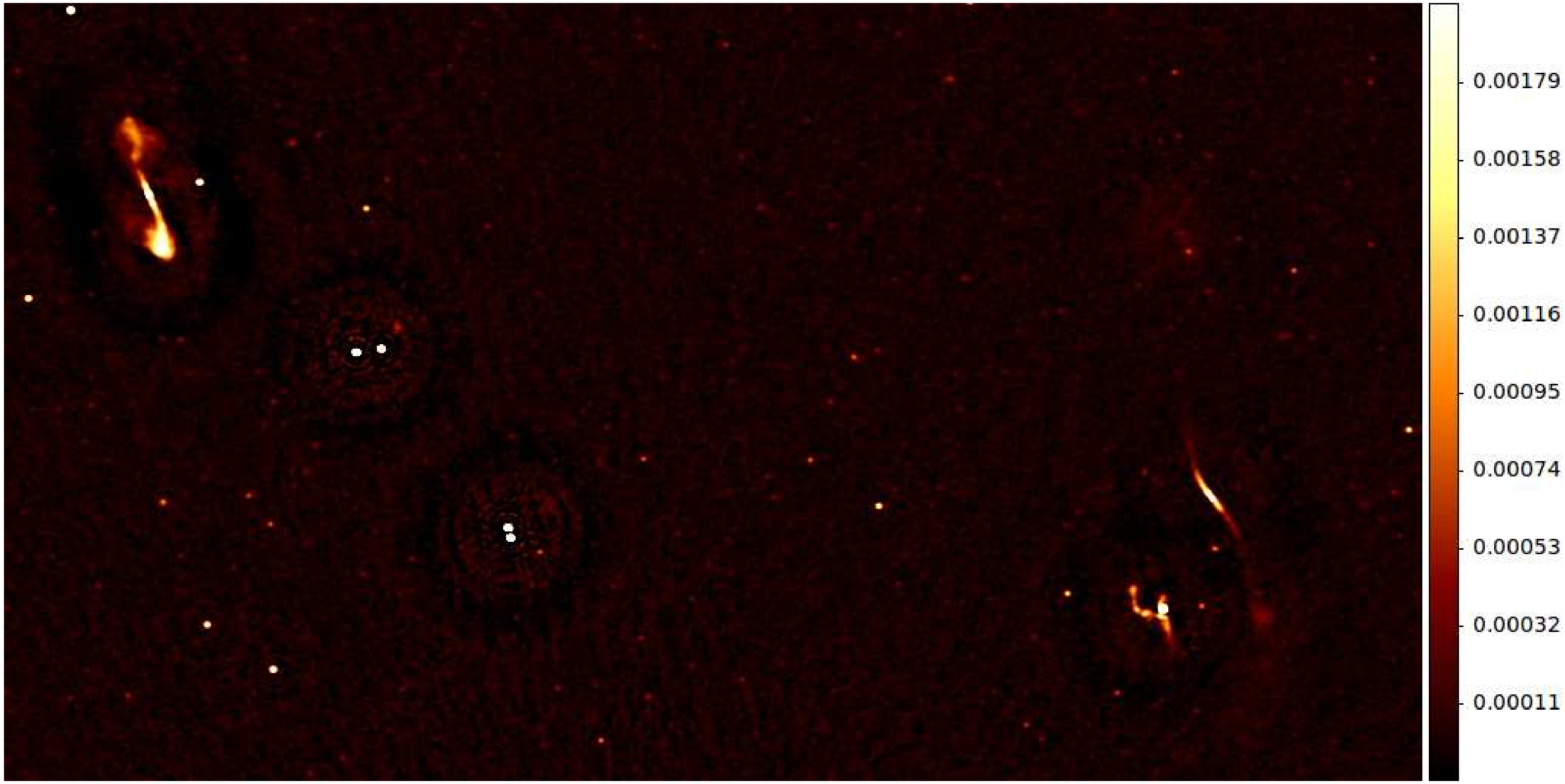
# Uncalibrated Image



about $1 \times 1.5$ degrees in the sky, no calibration

Data corrupted by systematic errors: ionosphere, beam, receiver.

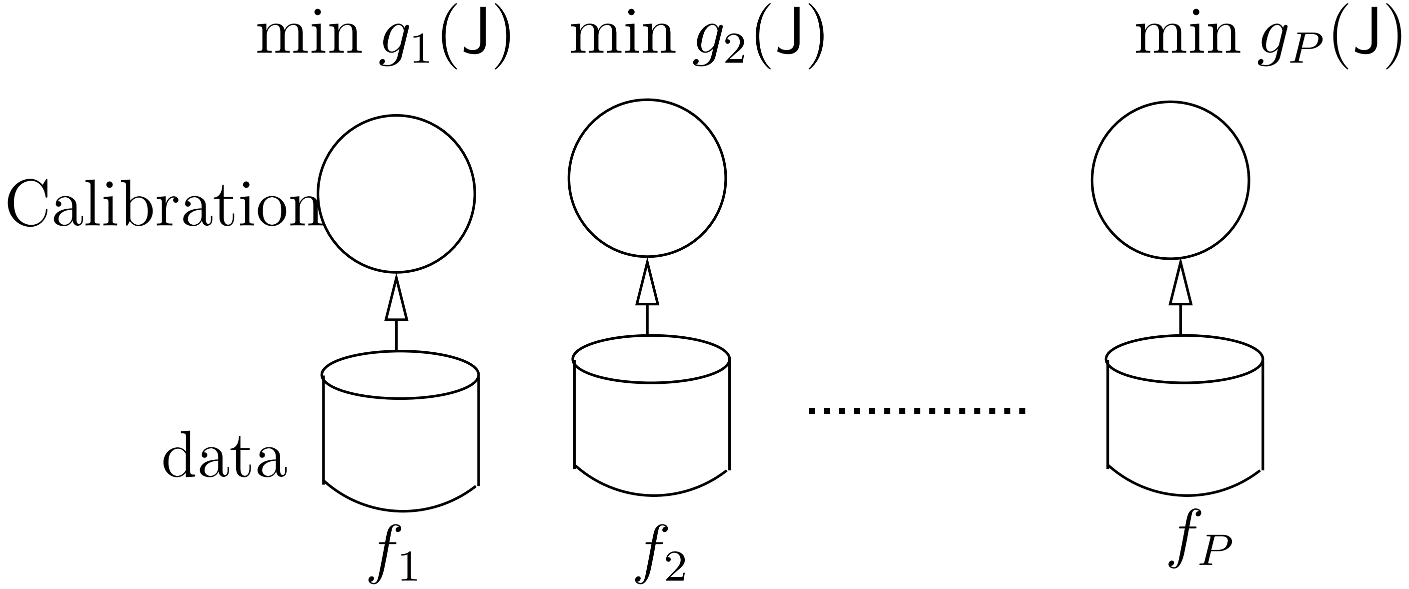
# Calibrated Image



about $1 \times 1.5$ degrees in the sky, after calibration

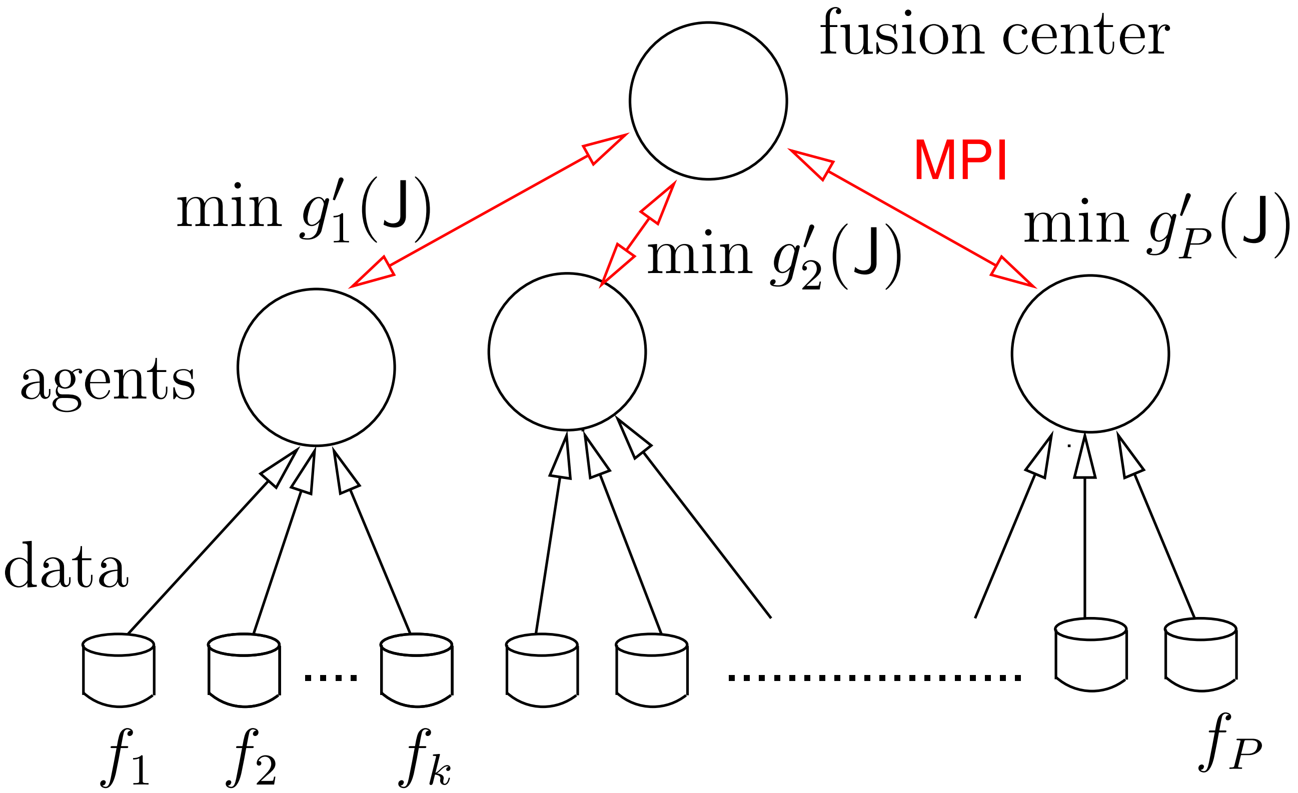
Is this good enough? depends on the science: most challenging is going deep. Some science is based on things visible in the image, some science is based on things invisible (unknown unknowns). *Ground truth in radio astronomy: there is no ground truth!*

# Normal Calibration



Data stored over a network of computers, divided into different subbands (frequencies).
Each calibration operates independently on data at different frequencies $f_i$.

# Distributed Calibration



Each agent works on subsets of data. Information is passed via the fusion center to reach consensus. A polynomial basis $\mathsf{B}_{f_i}$ in frequency exploits the natural behavior of systematic errors to improve calibration.

# Consensus Optimization

Distributed calibration minimizes augmented Lagrangian

$$L(\mathsf{J}_{f_1},\ldots,\mathsf{Z},\mathsf{Y}_{f_1},\ldots) = \sum_i g_{f_i}(\mathsf{J}_{f_i}) + \|\mathsf{Y}_{f_i}^H(\mathsf{J}_{f_i} - \mathsf{B}_{f_i}\mathsf{Z})\| + \frac{\rho}{2}\|\mathsf{J}_{f_i} - \mathsf{B}_{f_i}\mathsf{Z}\|^2.$$

Iterative optimization with $n = 1, 2, \ldots$

□ Locally optimize to find

$$(\mathsf{J}_{f_i})^{n+1} = \arg\min_{\mathsf{J}} L_i\left(\mathsf{J}, (\mathsf{Z})^n, (\mathsf{Y}_{f_i})^n\right)$$

□ Globally find average (closed form solution)

$$(\mathsf{Z})^{n+1} = \arg\min_{\mathsf{Z}} \sum_i L_i\left((\mathsf{J}_{f_i})^{n+1}, \mathsf{Z}, (\mathsf{Y}_{f_i})^n\right)$$

□ Locally update Lagrange multiplier

$$(\mathsf{Y}_{f_i})^{n+1} = (\mathsf{Y}_{f_i})^n + \rho((\mathsf{J}_{f_i})^{n+1} - \mathsf{B}_{f_i}(\mathsf{Z})^{n+1})$$

Topic of this talk: Performance analysis of distributed calibration.

# Performance Analysis of Calibration

Various ways exist:

□ Cramer Rao lower bound : gives variance of solutions.

□ Jacobian leverage : gives variance of residuals.

□ Various hand-crafted simulations.

We follow a different approach:

□ We need to study the residual (not the solutions), and need a simple way.

□ Signals hidden in the residual are weak, we are after their statistics.

□ We study the probability density functions (PDF) of input (data) and output (residual).

□ Power spectra $\sim$ autocorrelation $\sim$ PDF.

□ Inspired by optimal mass transport problems (Monge-Kantorovich).

# Basic Method

Let $\boldsymbol{x}$ be input (data) and $\boldsymbol{y}$ be the residual. Assumed model for input $\boldsymbol{x}$

$$\boldsymbol{x} = \boldsymbol{s}(\boldsymbol{\theta}) + \boldsymbol{n}$$

$\boldsymbol{n}$ model error, unmodelled signal, noise, ..
In calibration, parameters $\boldsymbol{\theta}$ are estimated as

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\arg\min} f(\boldsymbol{x}, \boldsymbol{\theta})$$

Cost function based on noise model $f(\cdot,\cdot)$ e.g.,

$$f(\boldsymbol{x}, \boldsymbol{\theta}) \overset{\triangle}{=} \|\boldsymbol{x} - \boldsymbol{s}(\boldsymbol{\theta})\|^2$$

and other cost functions can be used here (for imaging, foreground removal etc.).
Residual (where hidden signals remain)

$$\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{s}(\widehat{\boldsymbol{\theta}})$$

$\boldsymbol{x} \sim p_X(\boldsymbol{x})$ and $\boldsymbol{y} \sim p_Y(\boldsymbol{y})$ PDF of data and residual.
How are they related?

# PDF

Assume a bijective mapping $\boldsymbol{T}(\cdot)$ that transforms data into the residual. All operations on data (calibration, deconvolution, foreground removal) are inside $\boldsymbol{T}(\cdot)$.

$$\boldsymbol{y} = \boldsymbol{T}(\boldsymbol{x}), \ \ p_X(\boldsymbol{x}) = |\mathcal{J}| \ \ p_Y(\boldsymbol{T}(\boldsymbol{x}))$$

where

$$\mathcal{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \ldots & \frac{\partial y_1}{\partial x_D} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \ldots & \frac{\partial y_2}{\partial x_D} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial y_D}{\partial x_1} & \frac{\partial y_D}{\partial x_2} & \ldots & \frac{\partial y_D}{\partial x_D} \end{bmatrix}$$

and for the residual, this can be simplified as

$$\mathcal{J} = \boldsymbol{I}_D + \mathcal{A}, \ \ |\mathcal{J}| = \exp\left(\sum_{j=1}^{D} \log\left(1 + \lambda_j(\mathcal{A})\right)\right).$$

Ideally, $\boldsymbol{T}(\cdot)$ is the identity map ($|\mathcal{J}| = 1$), but in practice, because of ambiguities of calibration, a few eigenvalues of $\mathcal{A}$ are always non zero.

# PDF

Taking derivative of $\arg\min(\cdot)$ functions,

$$\mathcal{A} \triangleq \frac{\partial \boldsymbol{s}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \left(f_{\theta\theta}(\boldsymbol{x},\boldsymbol{\theta})\right)^{-1} \left[f_{X_1\theta}(\boldsymbol{x},\boldsymbol{\theta}) \ldots f_{X_D\theta}(\boldsymbol{x},\boldsymbol{\theta})\right]\Big|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}}$$

Jacobian of the model is $\frac{\partial \boldsymbol{s}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}$.
Hessian of the cost function is

$$f_{\theta\theta}(\boldsymbol{x},\boldsymbol{\theta}) \triangleq \frac{\partial^2 f(\boldsymbol{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}.$$

Derivative of the gradient of the cost function is

$$f_{X_m\theta}(\boldsymbol{x},\boldsymbol{\theta}) \triangleq \frac{\partial^2 f(\boldsymbol{x},\boldsymbol{\theta})}{\partial x_m \partial \boldsymbol{\theta}}.$$

All are evaluated at the solution $\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}$.

Extending this analysis to calibration along multiple directions is straightforward.

# Performance Metrics

Derivative of solutions w.r.t. data

$$
\begin{aligned}
&\mathrm{vec}\left(\frac{\partial \mathbf{J}_f}{\partial x_{p'q'r}}\right)\\
&= \left(\mathcal{D}_{\mathbf{J}}\mathrm{grad}(g_f(\mathbf{J}_f))\right.\\
&\left.+\frac{\rho}{2}\mathbf{I}_2 \otimes \left(\mathbf{F}^H\mathbf{F}\left(\mathbf{I}_{2N} + \left(\mathbf{I}_{2N} - \mathbf{F}^H\mathbf{F}\right)^{-1}\mathbf{F}^H\mathbf{F}\right)\right)\right)^{-1}\\
&\times \left(\mathbf{A}_{q'}\mathbf{J}_f\mathbf{C}^H_{p'q'f}\right)^T \otimes \mathbf{A}^T_{p'}\mathrm{vec}\left(\frac{\partial \mathbf{V}_{p'q'}}{\partial x_{p'q'r}}\right).
\end{aligned}
$$

Derivative of residual w.r.t. data

$$
\begin{aligned}
\mathrm{vec}\left(\frac{\partial \mathbf{R}_{pqf}}{\partial x_{p'q'r}}\right) = &\quad \mathrm{vec}\left(\frac{\partial \mathbf{V}_{pqf}}{\partial x_{p'q'r}}\right)\\
&- \left(\mathbf{C}_{pqf}\mathbf{J}^H_f\mathbf{A}^T_q\right)^T \otimes \mathbf{A}_p\mathrm{vec}\left(\frac{\partial \mathbf{J}_f}{\partial x_{p'q'r}}\right).
\end{aligned}
$$

Making $\rho = 0$ gives performance of normal calibration. Consensus polynomials construct $\mathbf{F}$.

# Simulations

□ LOFAR array with $N = 47$ stations, $B = 1081$ baselines,

□ Calibration done for $10$ time samples, $120$ frequencies in $[115, 185]$ MHz, so $D = 120 \times 10 \times 1081$ data points.

□ Normal calibration: each frequency is calibrated individually.

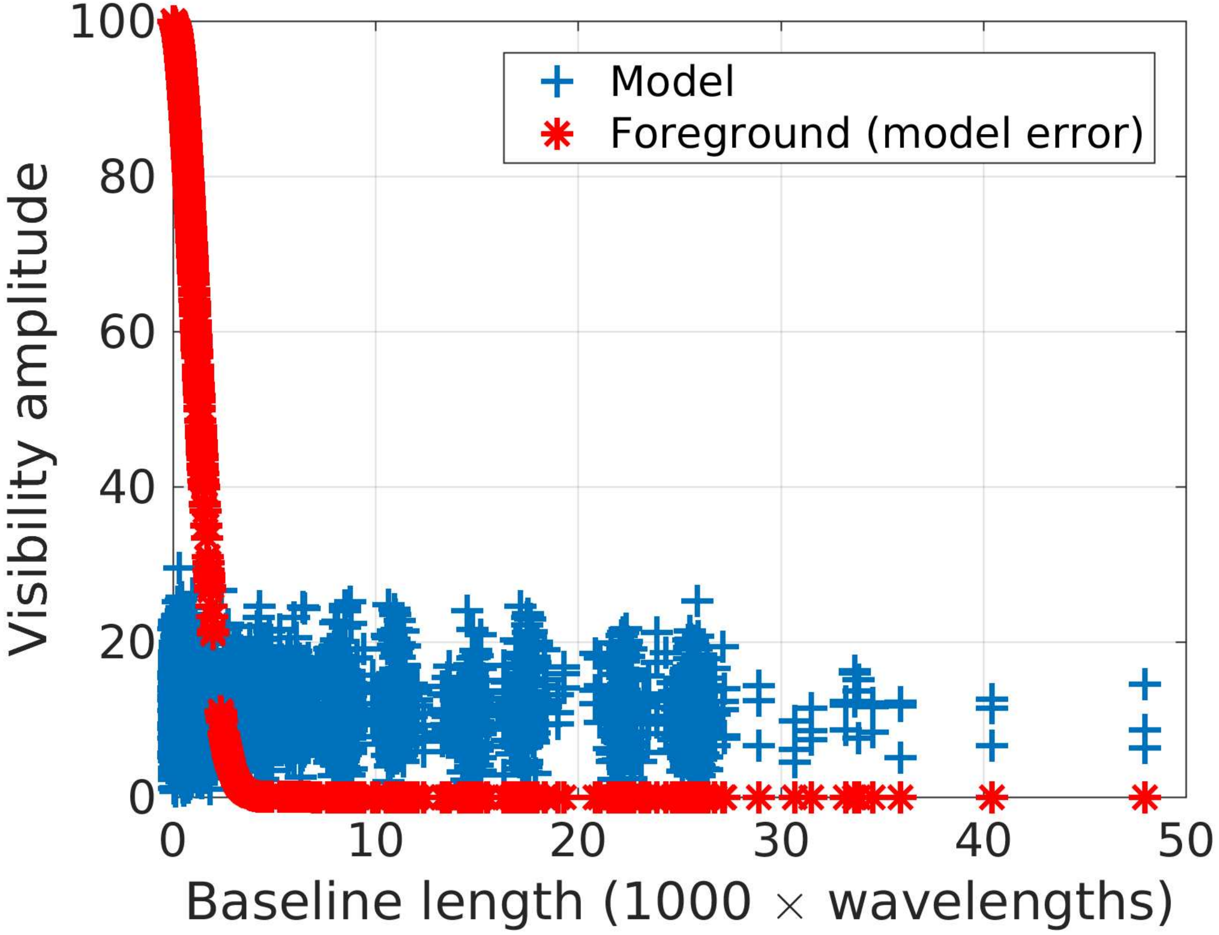
□ Distributed calibration: consensus optimization with a $3$-rd order polynomial in frequency, $\rho = 500$.

□ Sky model for calibration: point source with intensity $I = 10$ Jy.

□ Unknown sky: Gaussian with peak flux $5 \times I$, but only affecting short baselines.

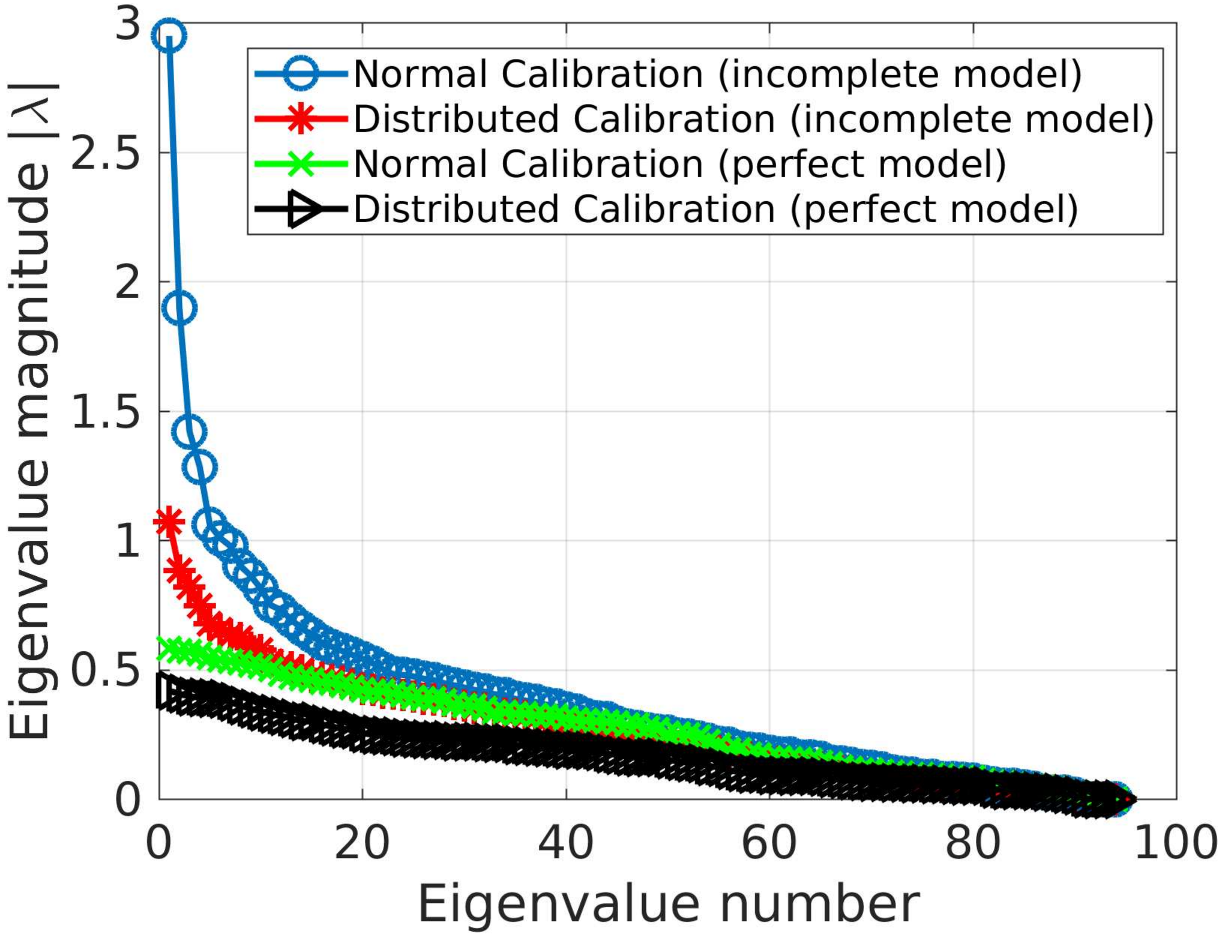
□ Random errors for $\mathbf{J}_f$, Gaussian noise with SNR $= 40$.

# Simulated data



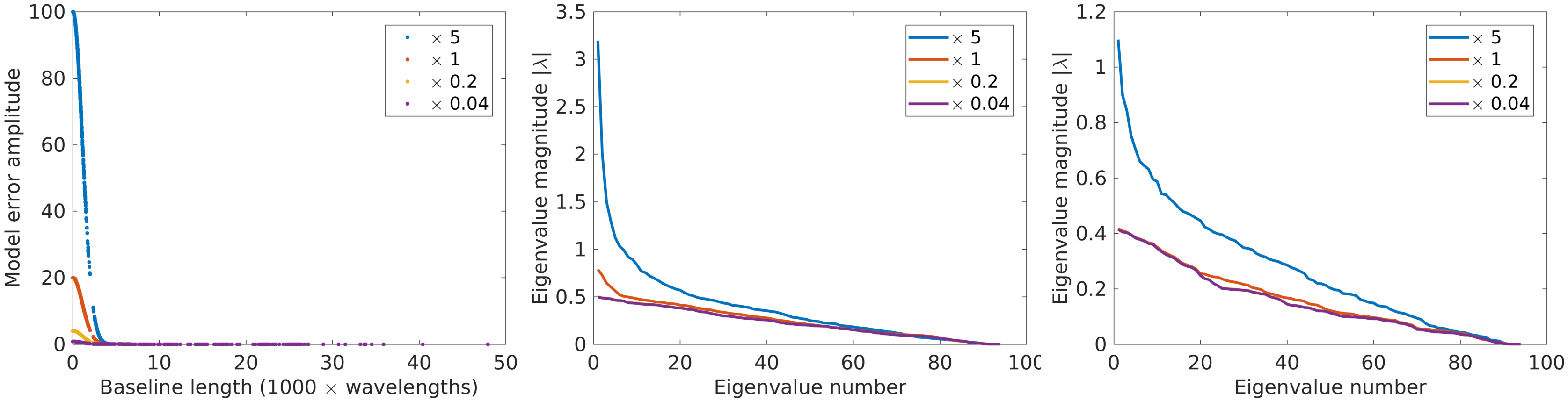
Model: point source, Model error: Gaussian with large spatial scale

# Eigenvalues of $\mathcal{A}$



$\mathcal{A}$ has only $\approx 2N$ non-zero eigenvalues (for real $XX$), rest is all $0$.

# How much model error can we tolerate?



(left) Model error (middle) normal calibration (right) distributed calibration

□ For this example, even a model error of $\times 1$ is tolerable with distributed calibration.

□ This result can be easily expanded to direction dependent calibration.

# Conclusions

- □ We have derived analytical relationships to measure the performance of distributed calibration.
- □ We can use these to study the effects of the sky model, the regularization parameter $\rho$, and the consensus polynomials on calibration.
- □ Even more challenging, but doable: infer the PDF of input data (where the actual science remains) from the PDF of the output residual.
- □ The same technique can be used to study the performance of other operations on data (other calibration methods, imaging, foreground subtraction etc.). Possible applications in other fields such as machine learning.
- □ All software (will be) available at https://github.com/nlesc-dirac.