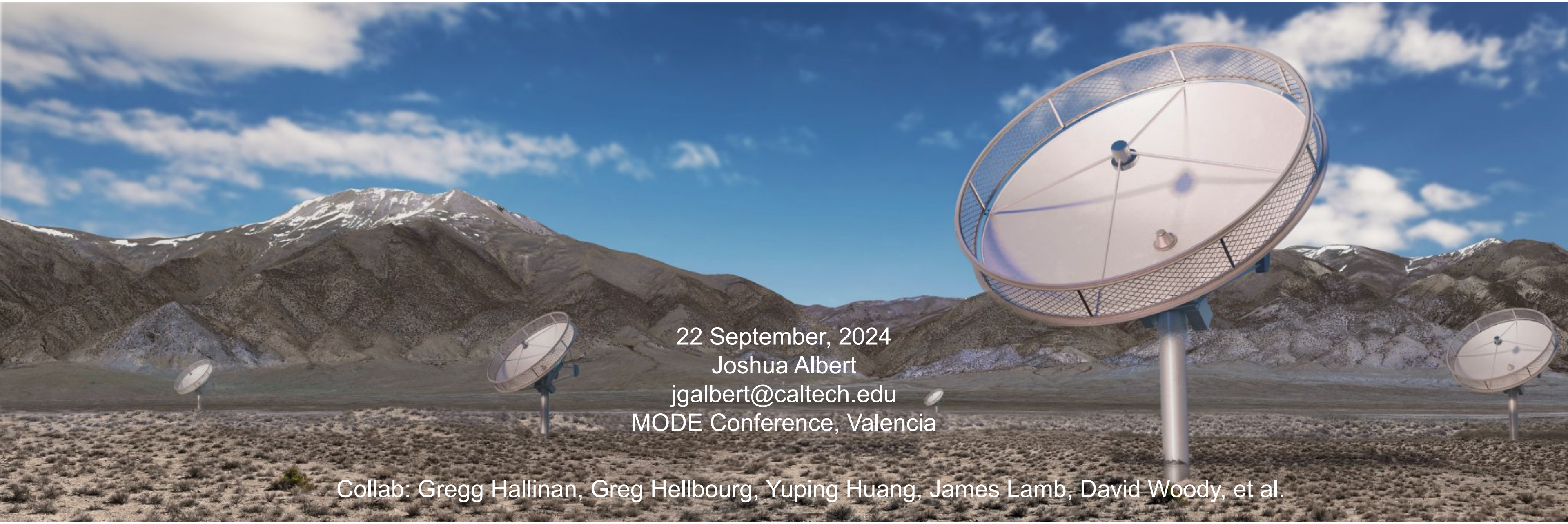
Optimising the DSA-2000



22 September, 2024

Joshua Albert

jgalbert@caltech.edu

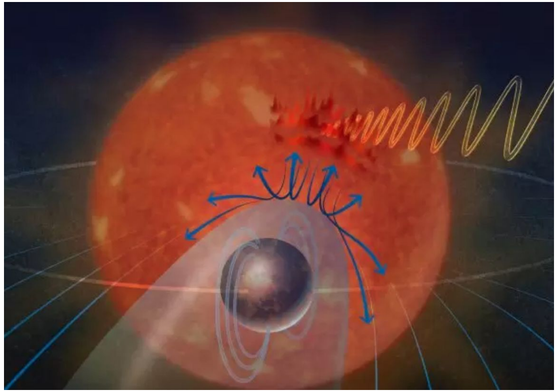
MODE Conference, Valencia

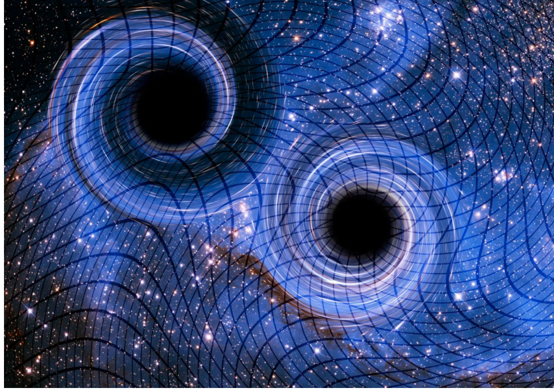
Collab: Gregg Hallinan, Greg Hellbourg, Yuping Huang, James Lamb, David Woody, et al.

Caltech

Schmidt Sciences

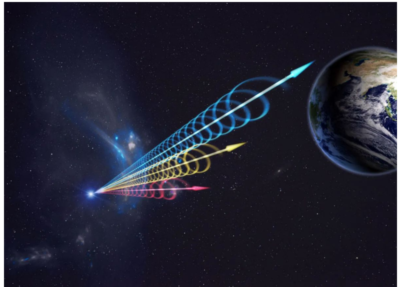
NANOGrav Physics Frontiers Center

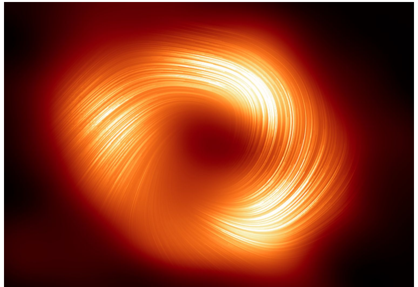




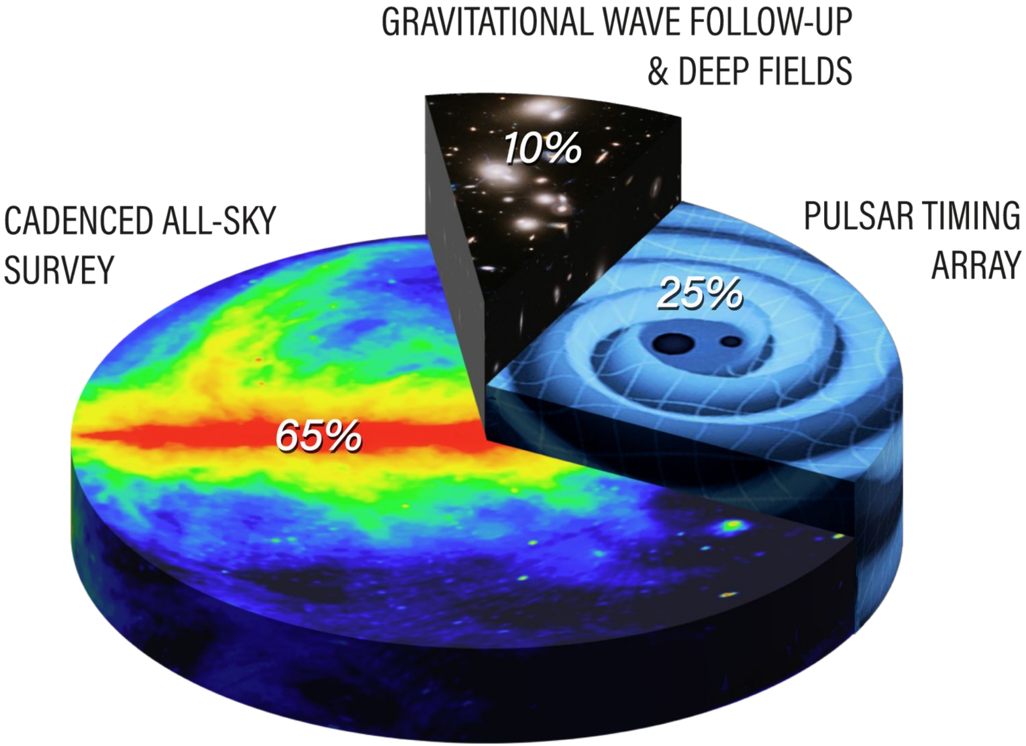
In this talk

1. Why forward modelling is key in achieving science outcomes for big experiments
2. Three examples of how it’s being used for DSA-2000

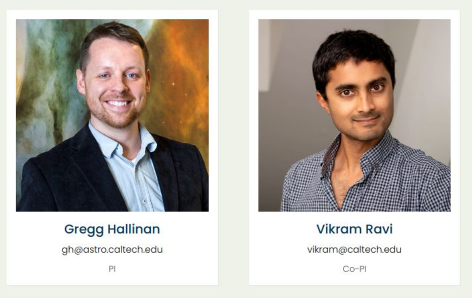


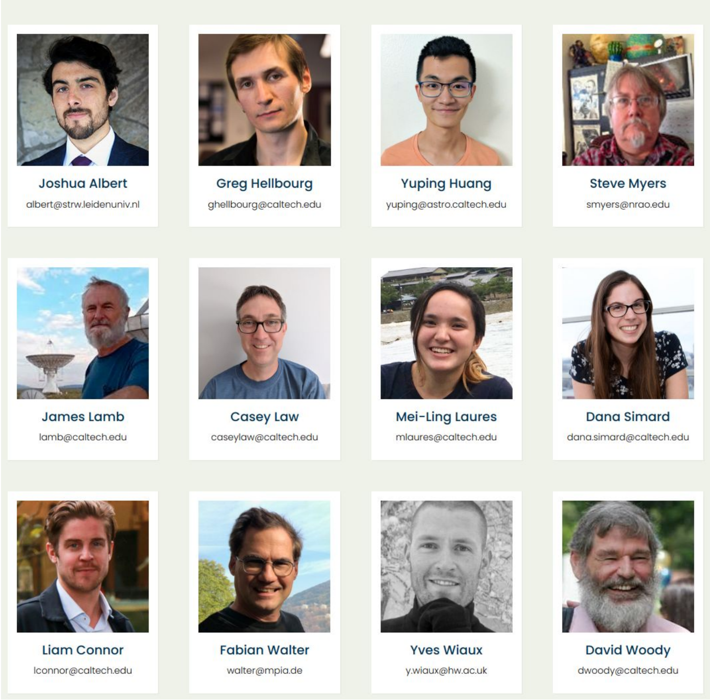


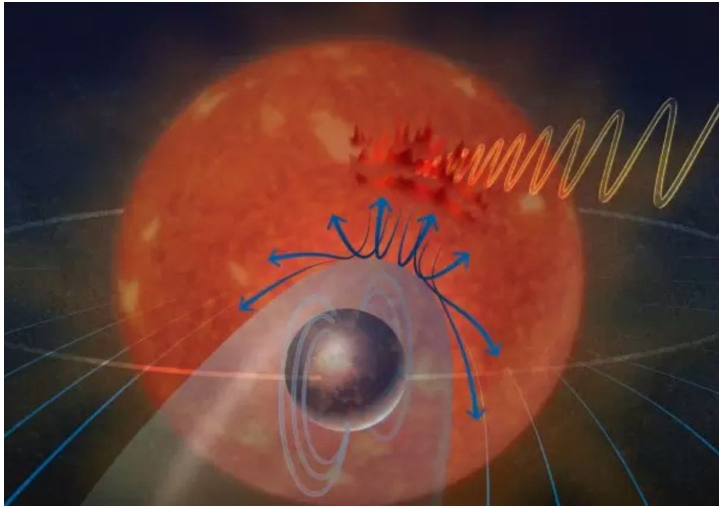
DSA-2000 Science and Team

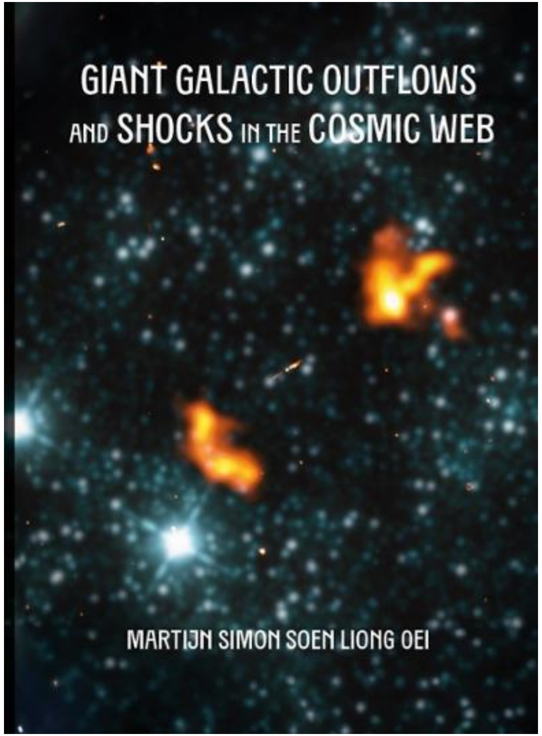


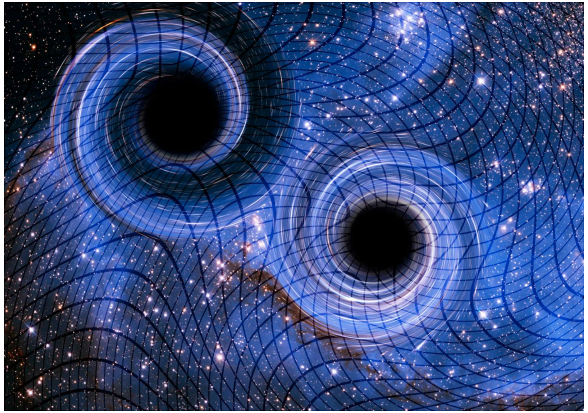
> 100x known sources, and a novel gravitational wave observatory!

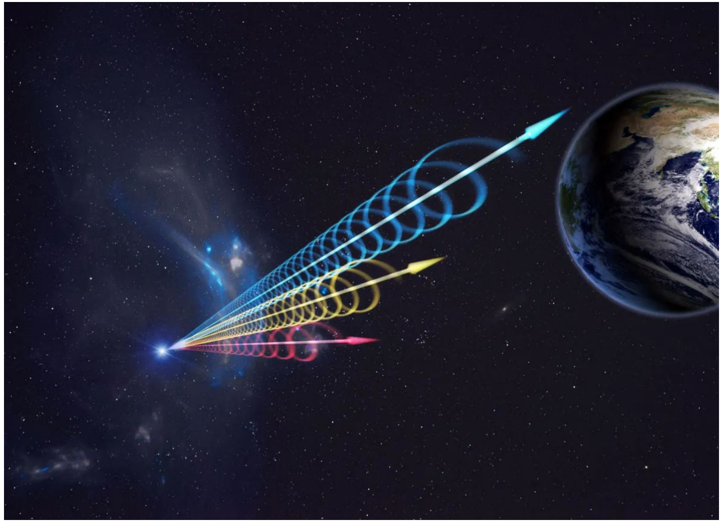


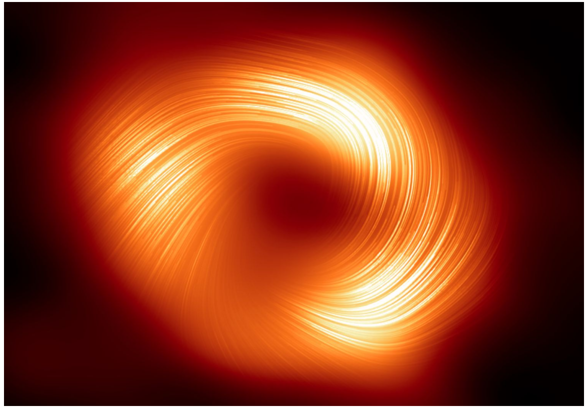




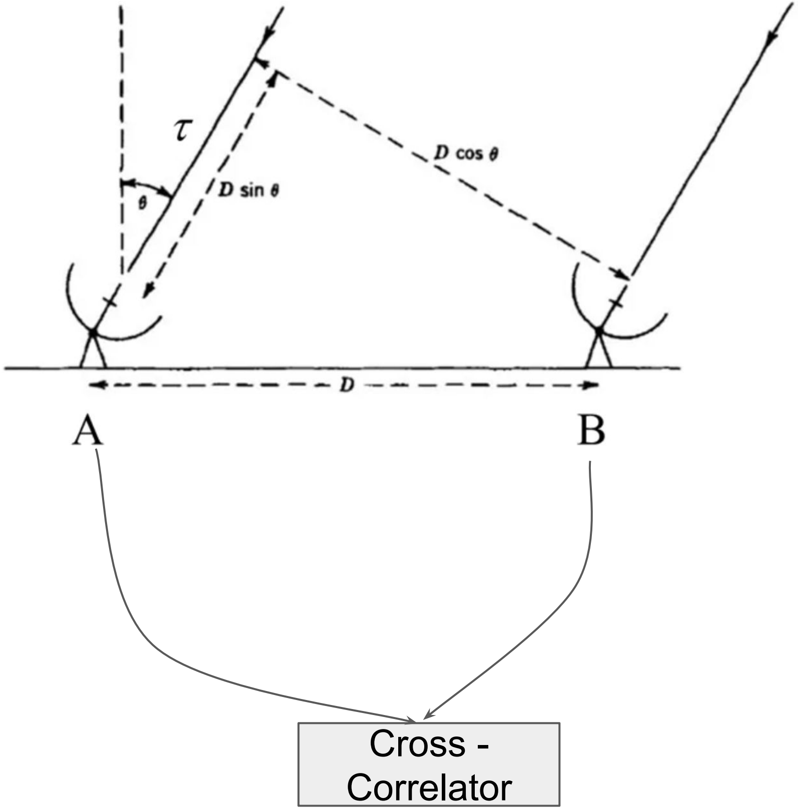

GIANT GALACTIC OUTFLOWS
AND SHOCKS IN THE COSMIC WEB
MARTIJN SIMON SOEN LIONG OEI

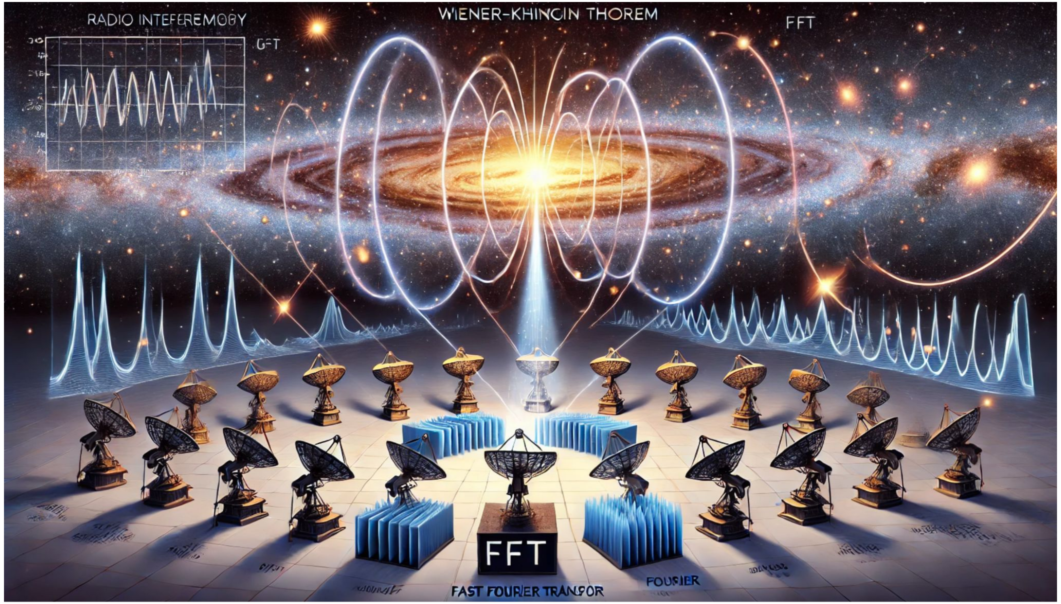






What is radio interferometry? Some basics…

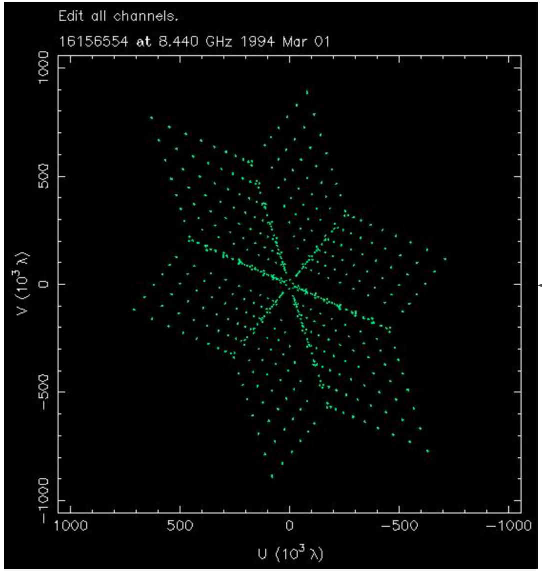




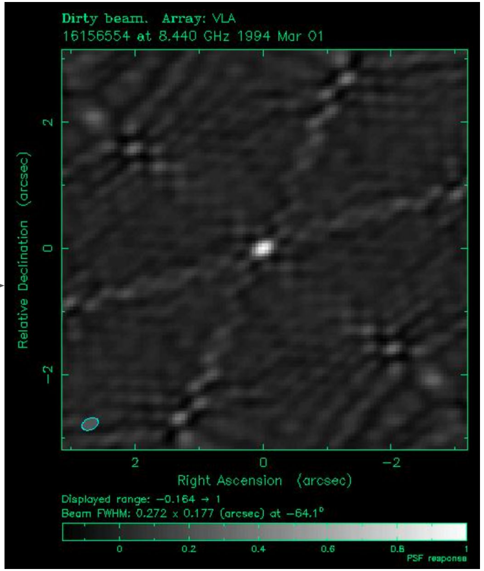
Wiener-Khinchin theorem states:
Sky Brightness = FFT(Electric field coherence)

Traditional radio imaging problem

Sampled aperture leads to complex **Point Spread Function (PSF)** that necessitates some form of inference

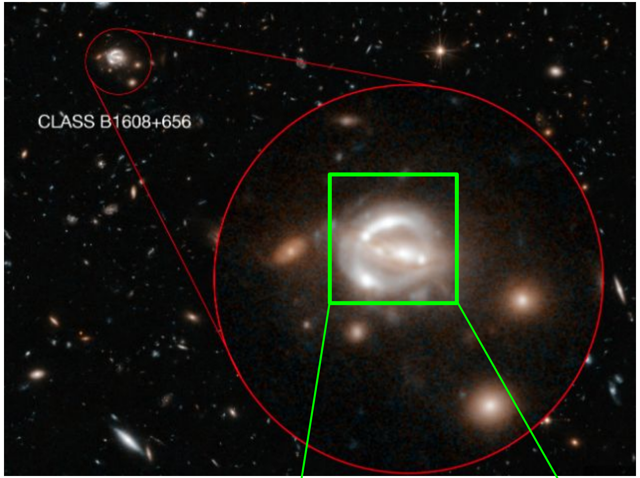


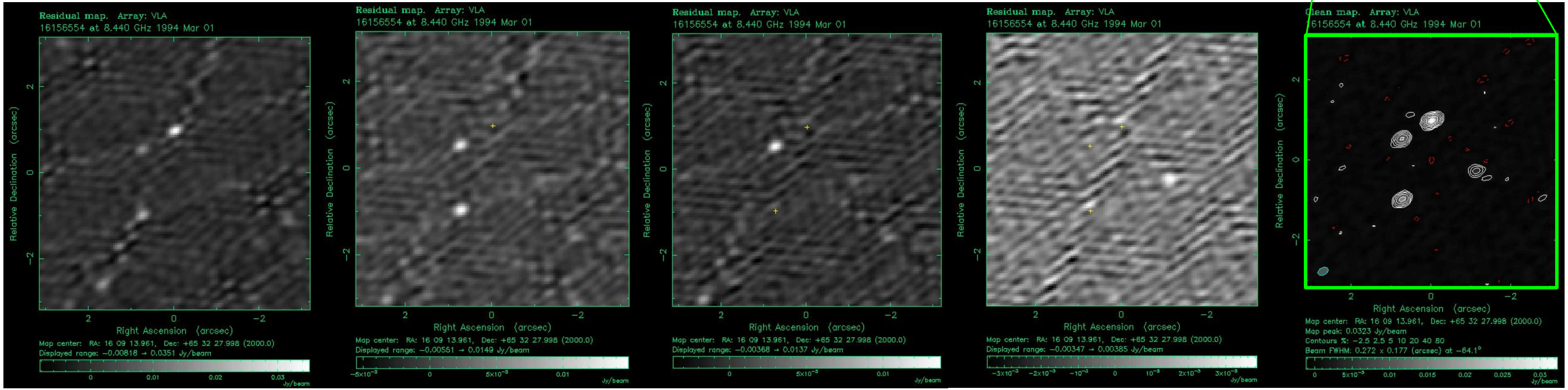
FFT



Traditional radio imaging problem

Classic iterative solution can takes days/weeks to analyse a single modern large dataset!





Traditional radio imaging problem

As the number of antennas (N) grows, we get better images but longer processing times and more data volume.

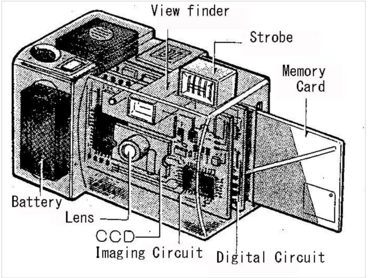
Traditionally (small N)	**Radio Camera (large N)**
Data volume $\propto N^2$	$\Rightarrow$ **Data volume $\propto N^2$**
Thermal Noise $\propto N^{-1}$	$\Rightarrow$ **Thermal Noise $\propto N^{-1}$**
Survey Speed $\propto N^2$	$\Rightarrow$ **Survey Speed $\propto N^2$**
Dynamic Range $\propto N$	$\Rightarrow$ **Dynamic Range $\propto N$**
Processing $\propto N^2\log(N)$	$\Rightarrow$ **Processing $\propto N^2\log(N)$**

The Challenge

To go where no one has gone before.

Schmidt Sciences

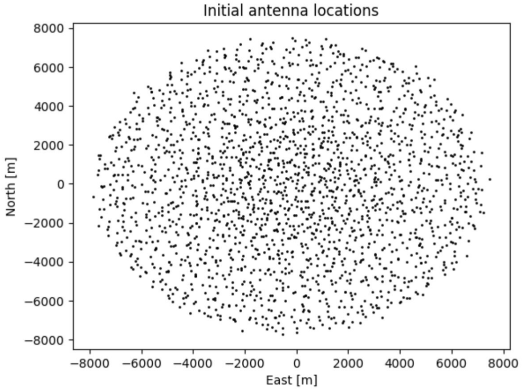
DSA-2000 the first point-and-shoot radio camera: revolutionising radio astronomy



Caltech OVRO NANOGrav

What makes a radio camera?

1. **Large N** fills in the aperture, “approaching a CCD”
2. **Real-time** capability, “don’t need to wait for imaging”



Radio Camera (large N)

⇒ **Data volume $\propto N^2$**

⇒ **Thermal Noise $\propto N^{-1}$**

⇒ **Survey Speed $\propto N^2$**

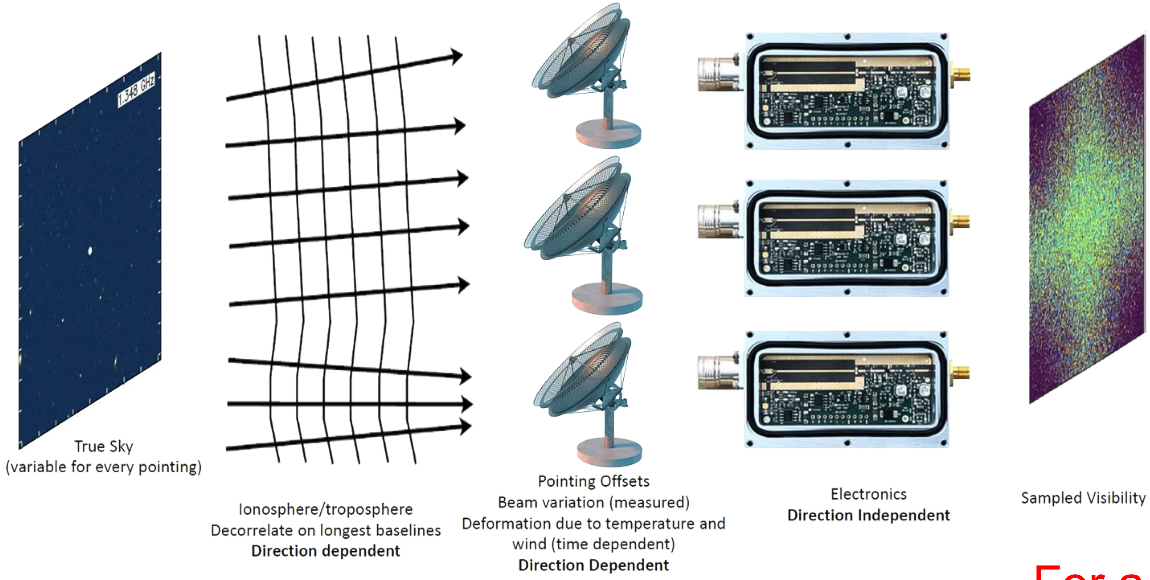
⇒ **Dynamic Range $\propto N$**

⇒ **Processing $\propto N^2 \log(N)$**

The Challenge

To make a streaming radio interferometer with thousands of antennas.

What is forward modelling?



Enables

Validating requirements

Design optimisation

Generating infinite datasets

For a 10.3min observation

Signal Simulation → Instrument Simulation → Data Simulation → Calibration → Imaging

~1GB | 12.6 PB | 200 TB | 800 TB | 16.7 GB

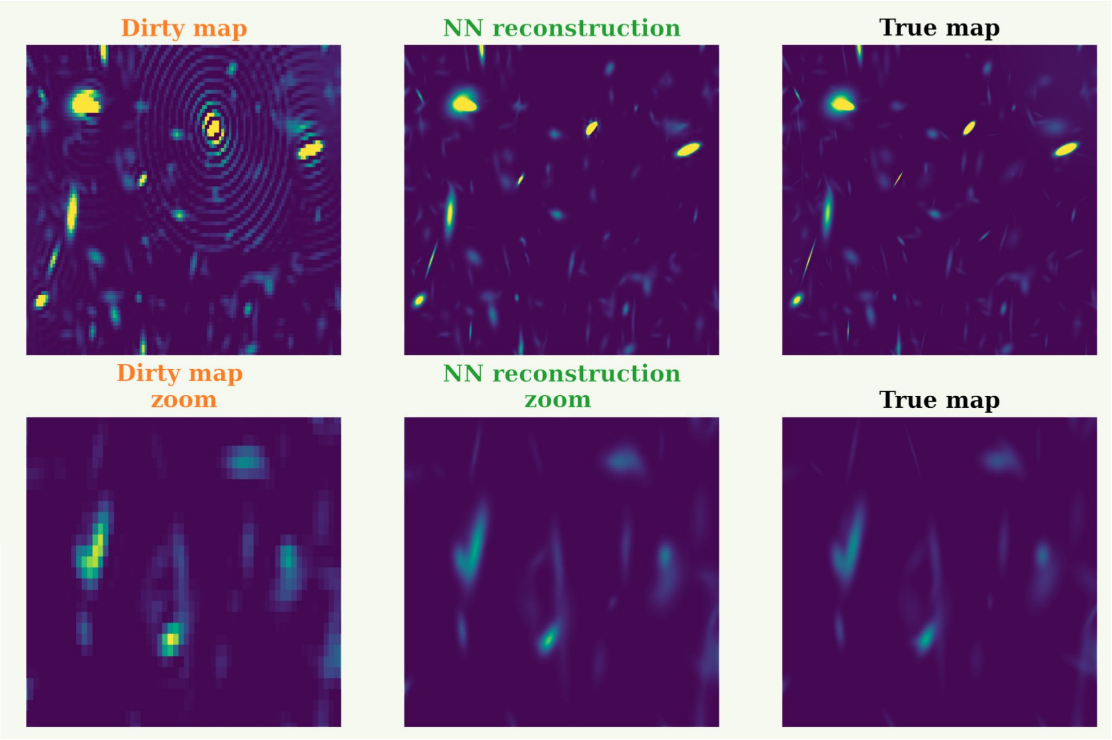
Generative data useful for surrogate ML components

With infinite datasets and realistic systematics you can explore ML surrogates.

E.g. deconvolving the PSF with CNNs.

(POLISH; Connor et al. 2021)

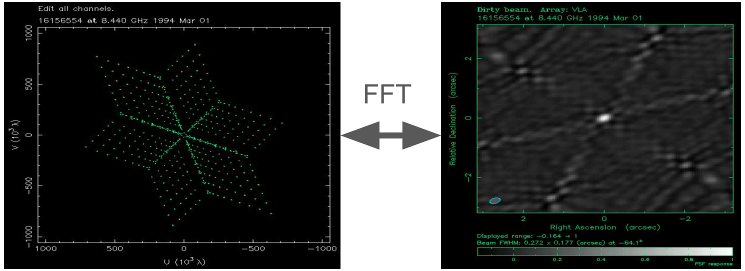

Liam Connor

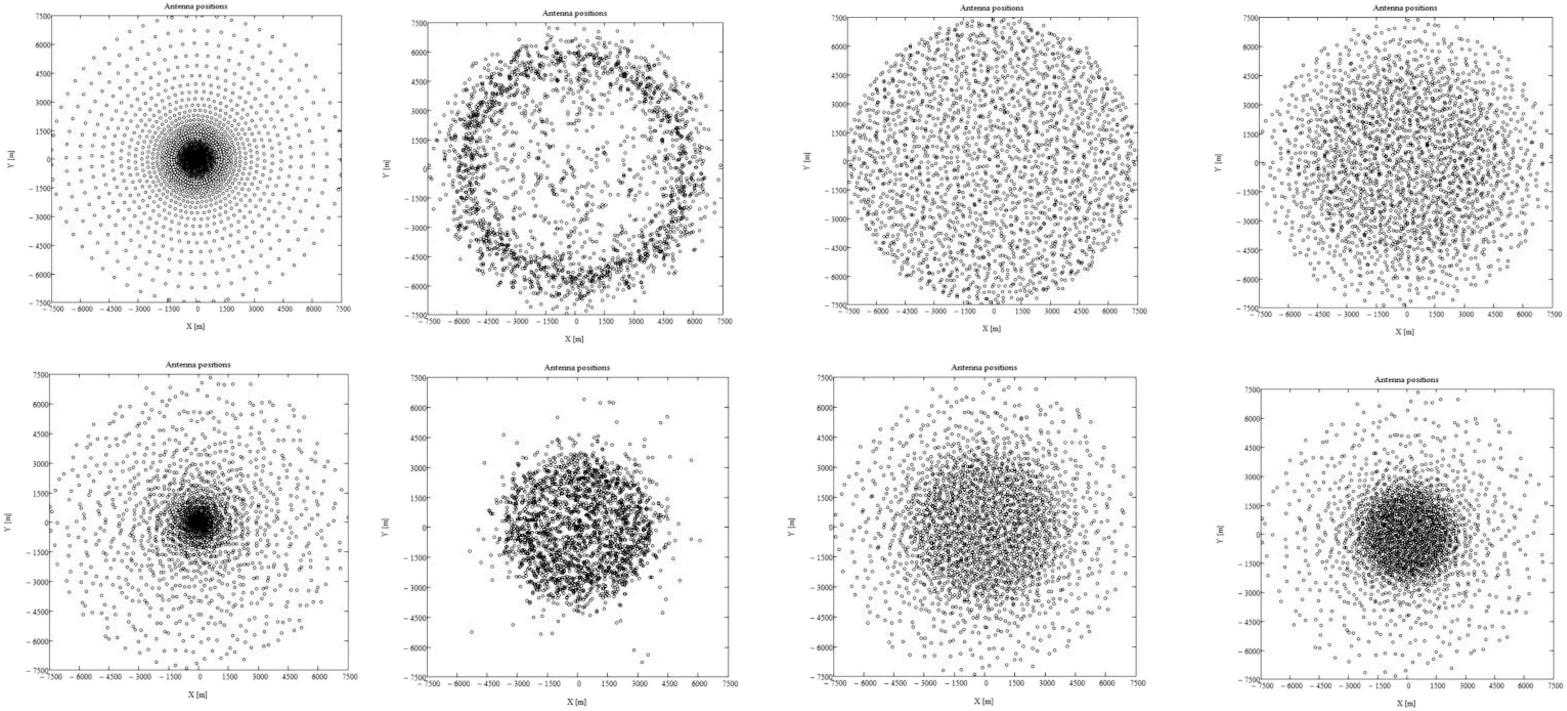


Three examples of optimising DSA-2000

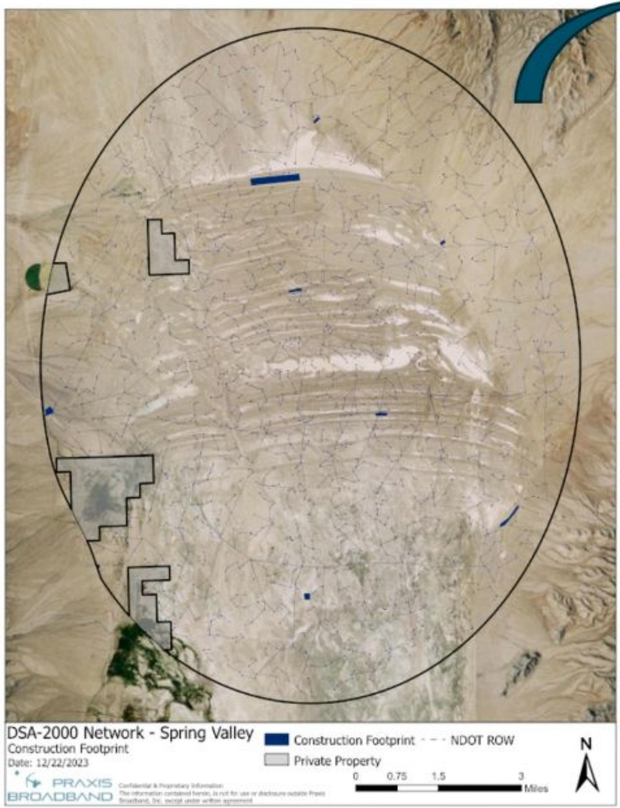
1. Array layout, subject to land allotment constraints.
2. Dish design, subject to manufacturing constraints.
3. Calibration hyper-parameters, subject to real-time.

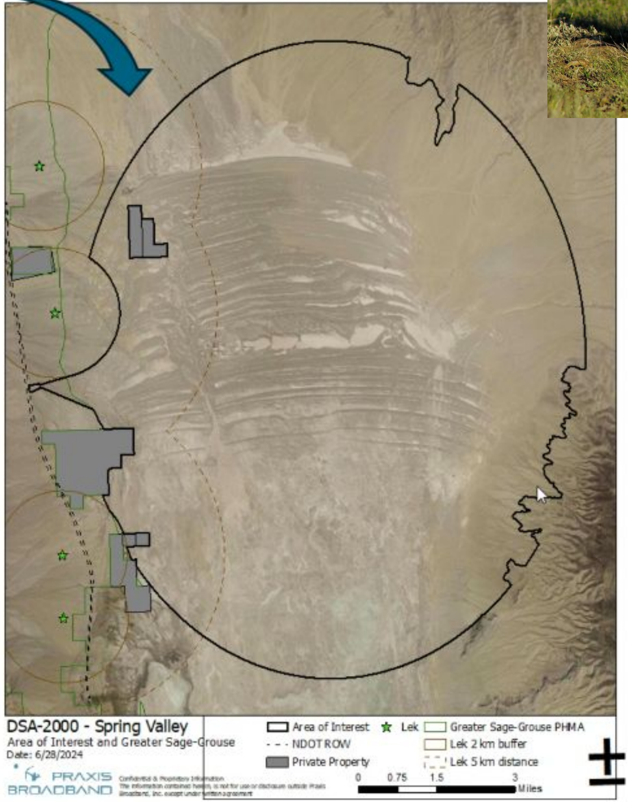
Optimising array layout





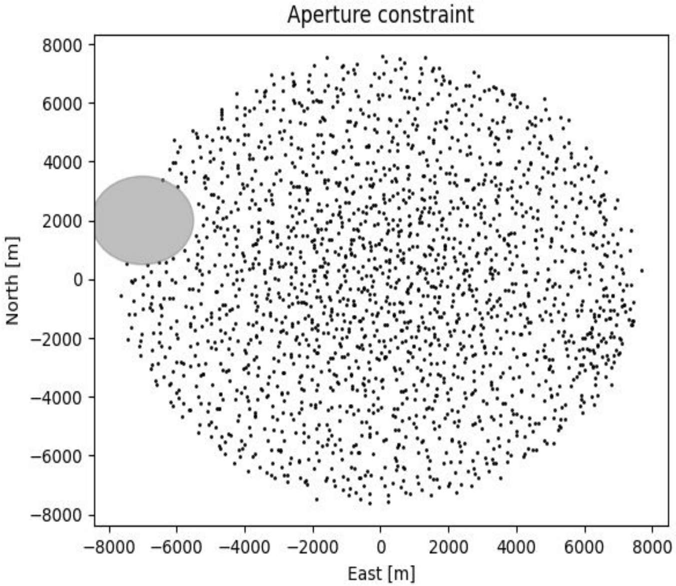
Choosing a site can be a challenge…

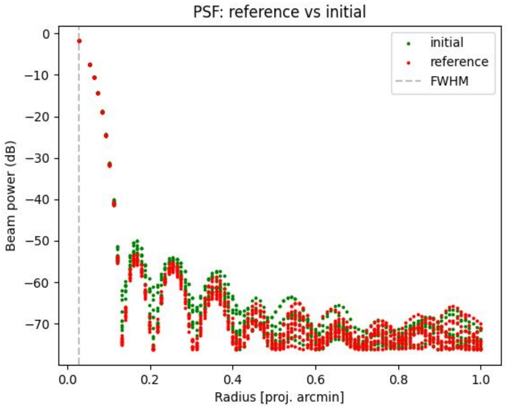


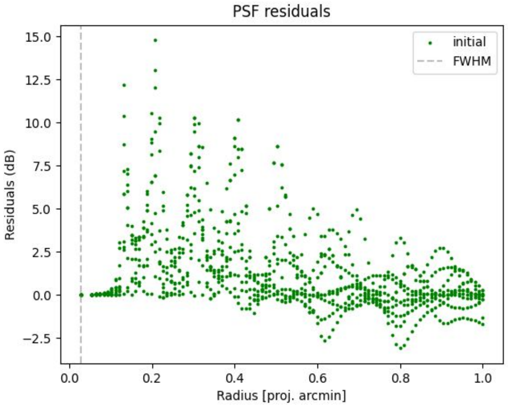


Optimisation objective is reference PSF

Science requirements have already be verified with reference PSF





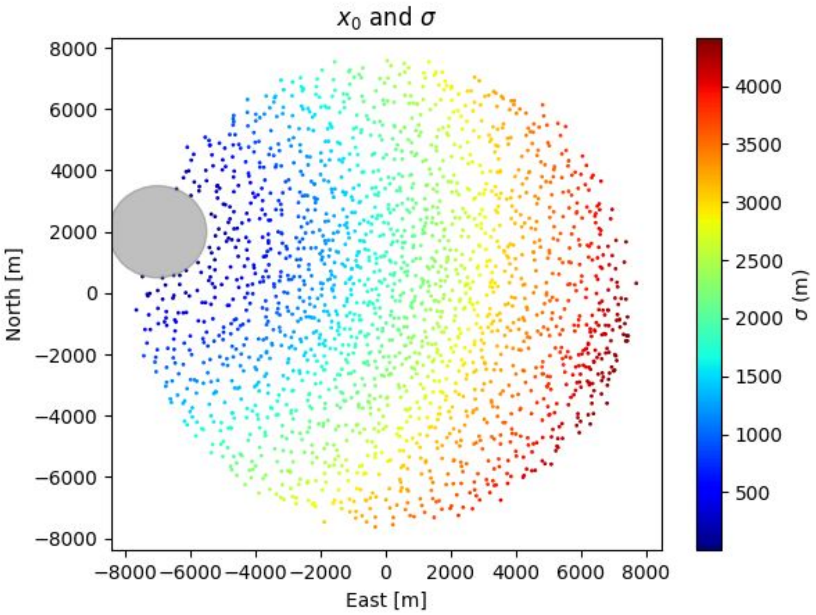


Defining problem with probabilistic programming

Formulate as a maximum likelihood problem.

Constraints defined via setting per-antenna $\mathbf{x}_0$ and σ

$$\begin{aligned}
\mathbf{x} &\sim \mathcal{N}[\mathbf{x}_0, \sigma^2 \mathbf{I}] \\
\mathbf{y} &= f_{\mathrm{PSF}}(\mathbf{x}) \\
p(\mathbf{y} \mid \mathbf{x}) &= \mathcal{N}[f_{\mathrm{PFS}}(\mathbf{x}_{\mathrm{ref}}), \epsilon^2 \mathbf{I}]
\end{aligned}$$



Implement probabilistic program with JAXNS

Formulate as a maximum a-posteriori problem.

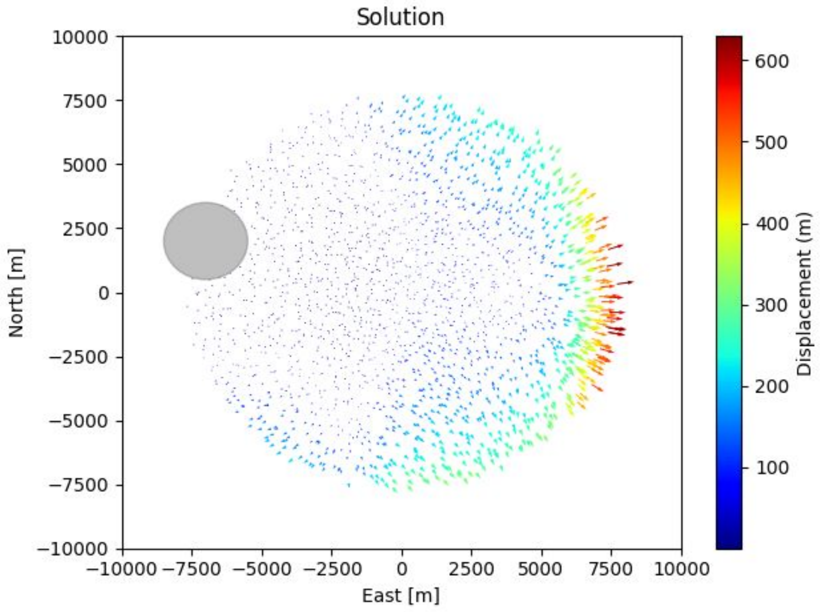
Constraints defined via setting per-antenna $\mathbf{x}_0$ and σ

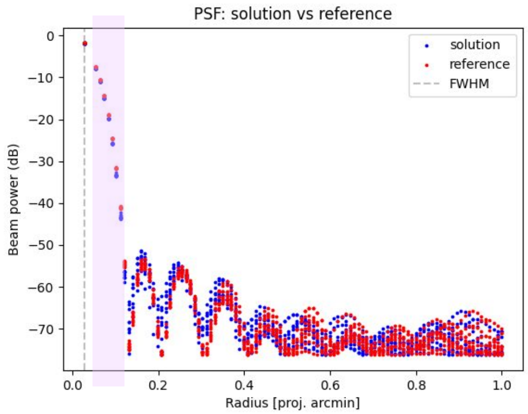
$$
\begin{aligned}
\mathbf{x} &\sim \mathcal{N}[\mathbf{x}_0, \sigma^2 \mathbf{I}] \\
\mathbf{y} &= f_{\mathrm{PSF}}(\mathbf{x}) \\
p(\mathbf{y} \mid \mathbf{x}) &= \mathcal{N}[f_{\mathrm{PFS}}(\mathbf{x}_{\mathrm{ref}}), \epsilon^2 \mathbf{I}]
\end{aligned}
$$

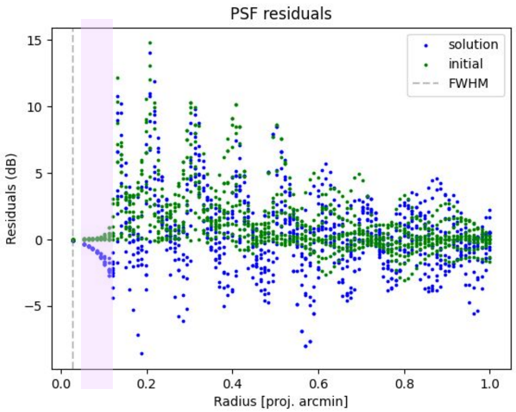
```python
from jaxns import Model, Prior
from jax.scipy import optimize
import tensorflow_probability.substrates.jax  as tfp

tfpd = tfp.distributions
x0 = ... # [n, 2]
sigma = ... # [n, 2]
ref_psf = ... # [m]

def prior_model():
    x = yield Prior(tfpd.Normal(x0, sigma),
                    'x').parametrised()
    y = compute_psf(x)
    return y

def log_likelihood(y):
    return tfpd.Normal(ref_psf, epsilon).log_prob(y)

model = Model(prior_model, log_likelihood)
result = optimize.minimize(
    lambda params: -model(params).log_prob_joint(),
    x0=model.params,
    method='BFGS'
)
print(result.x)
```

Optimised PSF vs Reference PSF

We are able to maintain the FWHM, while also decreasing sidelobes.





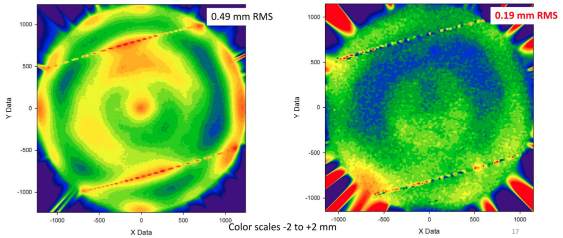


A similar application: Optimising dish manufacturing

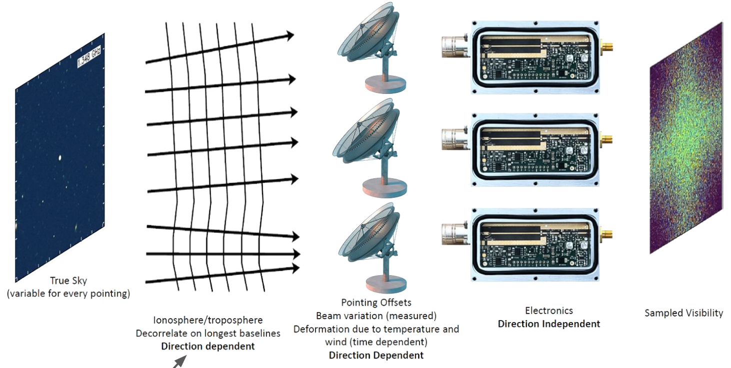
Systematics:

- Pointing errors
- Feed offsets
- Elevation-dependent gravitational deformations
- Surface RMS

Each has an impact on the signal path.



Optimising Calibration



Calibration involves fixed point solvers

$$\mathbf{g}^{n+1} = F(\mathbf{g}^{n}; \theta)$$

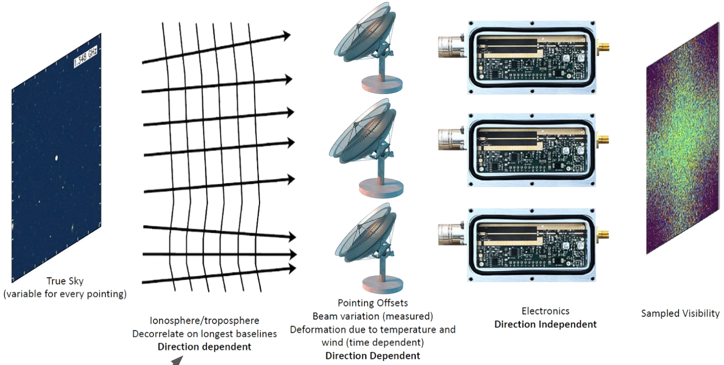
However, there are many parameters θ that affect performance.

1. Number of *approximate* vs *exact* steps
2. Damping parameters
3. Improvement threshold parameters
4. Metric p-norm

Adaptive Multi-step Levenberg-Marquardt (Fan et al. 2019) has **11 total parameters** that define the algorithm.

Question: Can we choose θ *to solve in real-time?*

Optimising Calibration



Question: Can we choose θ to solve in real-time?

$$\mathbf{g}^{n+1} = F(\mathbf{g}^n; \theta)$$

Options:

1. Implicit differentiation
2. Gradient-free optimisation

$$G(\mathbf{g};\theta) = F(\mathbf{g};\theta) - \mathbf{g} \quad (G(\mathbf{g}_*) = \mathbf{0})$$

$$\mathrm{JVP}_G(\mathbf{v};\mathbf{g}_*) = \nabla_\theta G(\mathbf{g}_*)\cdot\mathbf{v} \quad (\text{Implicit diff})$$

$$\delta > \mathbb{E}\left[||F(\mathbf{g}^n;\theta) - \mathbf{g}_*||\right] \quad \text{for small } n$$

Optimising Calibration: gradient-free global optimisation (JAXNS)

Reformulate as MAP problem

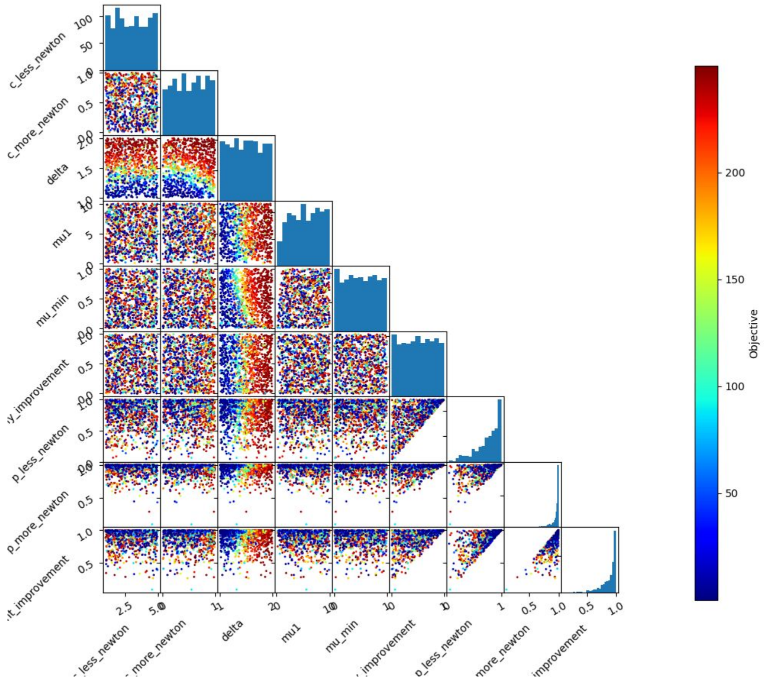
$$\mathbf{g}_* = \arg\max_{\mathbf{g}} p(\mathbf{y} \mid \mathbf{g}) p(\mathbf{g})$$

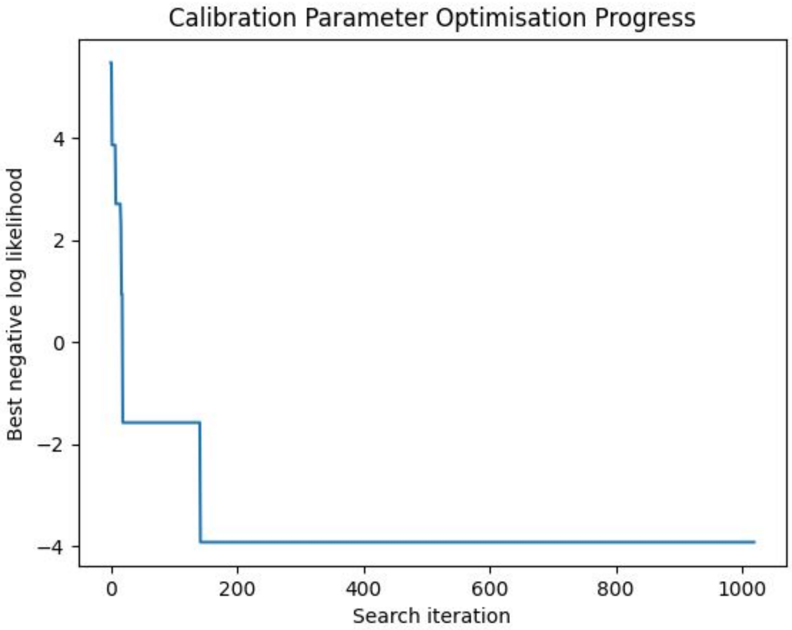
Use generative data and JAXNS for global optimisation,

$$\delta > \mathbb{E}\left[|| F(\mathbf{g}^{n}; \theta) - \mathbf{g}_* || \right] \quad \text{for small } n$$

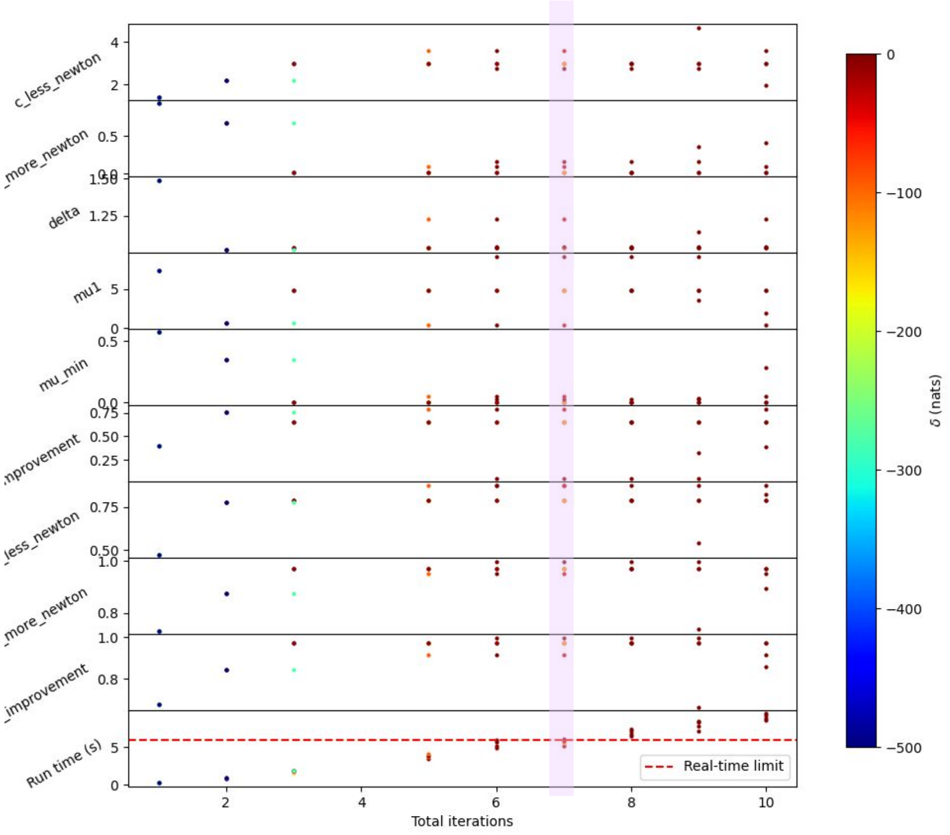
```python
from jaxns.experimental import (
    GlobalOptimisation,
    TerminationCondition
)

def prior_model() -> CalibrationParams:
    ...

def log_likelihood(params: CalibrationParams):
    ...

model = Model(
    prior_model=prior_model,
    log_likelihood=log_likelihood
)

go = GlobalOptimisation(model)
term_cond = TerminationCondition(
    max_likelihood_evaluations=1024
)
results = go.run(jax.random.PRNGKey(42), term_cond)
```

Optimising Calibration: global optimisation using JAXNS





Optimising Calibration: selecting optimal real-time parameters



Optimal **Adaptive Multi-step Levenberg-Marquardt** (Fan et al. 2019) parameters *for this particular dataset*, such that we are real-time.

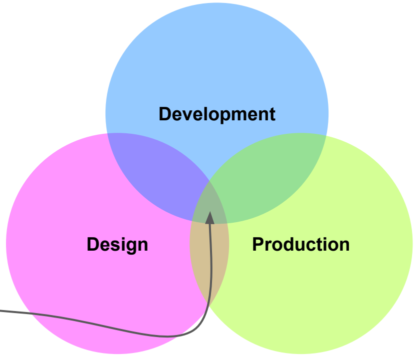
$$
\begin{aligned}
c_{\text{lessNewton}} &= 2.78 \\
c_{\text{moreNewton}} &= 0.16 \\
\delta &= 1.04 \\
\mu_1 &= 9.28 \\
\mu_{\min} &= 0.02 \\
p_0 &= 0.06 \\
p_{\text{lessNewton}} &= 0.88 \\
p_{\text{moreNewton}} = p_{\text{sufficient}} &= 1
\end{aligned}
$$

Summarising the importance of forward modelling

In today's science de-risking science outcomes is crucial to success,

- Science outcomes are de-risked via detailed forward modelling, and design optimisation (enabled by auto-diff etc.).
- The feasibility of analysis can be assessed via realistic generative models.
- High-level frameworks, e.g. JAX, enable a self-consistent platform to merge design, development, *and production.*

That's where you want your team focused.



Thank you