VLBI Imaging Techniques

Andrew Chael
NHFP Fellow
Princeton University

March 18, 2020

Event Horizon Telescope

Please fill out evaluation survey at
http://bit.ly/BHPIRE-Imaging !

Outline

1. VLBI Review
2. VLBI Imaging Methods
   - CLEAN
   - RML
3. Validating an Image
4. Extensions

VLBI Review

Why do we need VLBI?

M87 is supermassive, so its shadow is big:

$$d_{\text{shadow}} \approx 650 \text{ AU}$$

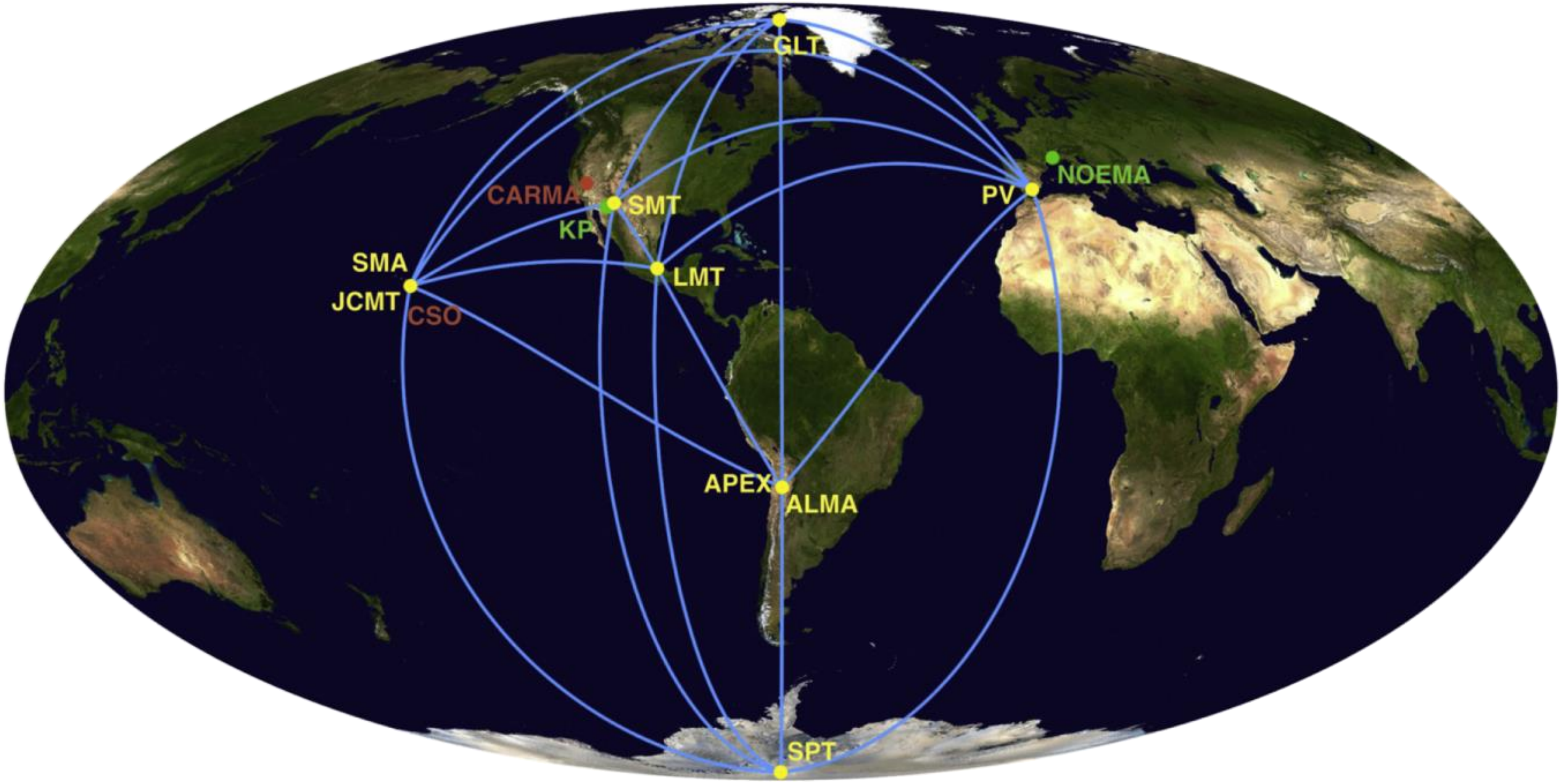
Unfortunately, M87 is really far away…..

$$D_{\text{M87}} \approx 50 \text{ million ly}$$

To us, M87's shadow is really, really, really small

$$\frac{d_{\text{shadow}}}{D_{\text{M87}}} \approx 40\mu\text{as} \approx 10^{-8}\text{deg}$$

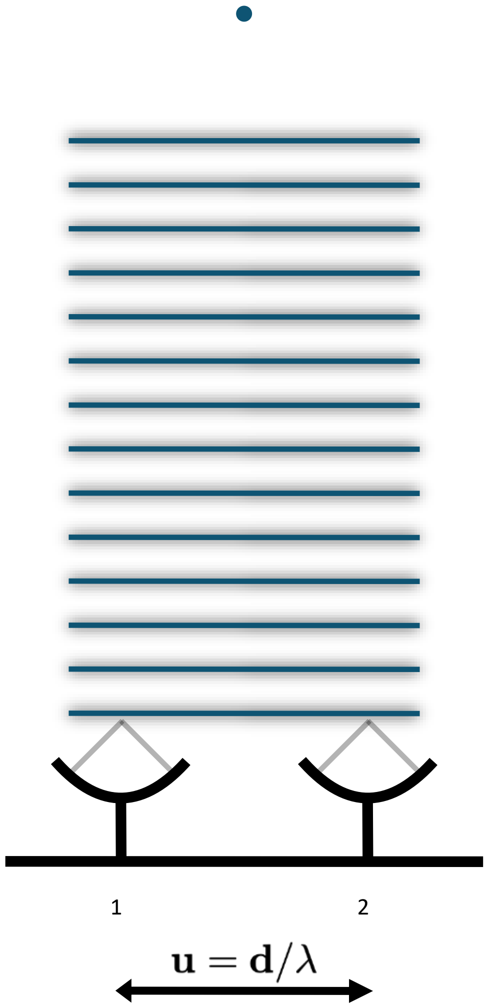
The Event Horizon Telescope



$$\text{Resolution} \approx \frac{\lambda}{d_{\text{Earth}}} \approx \frac{1.3\,\text{mm}}{1.3 \times 10^{10}\,\text{mm}} \approx 20\mu as$$

Image Credit:
EHT Collaboration 2019 (Paper II)

VLBI Measures "Visibilities", which correspond to **Spatial Coherence** of an EM Wavefront

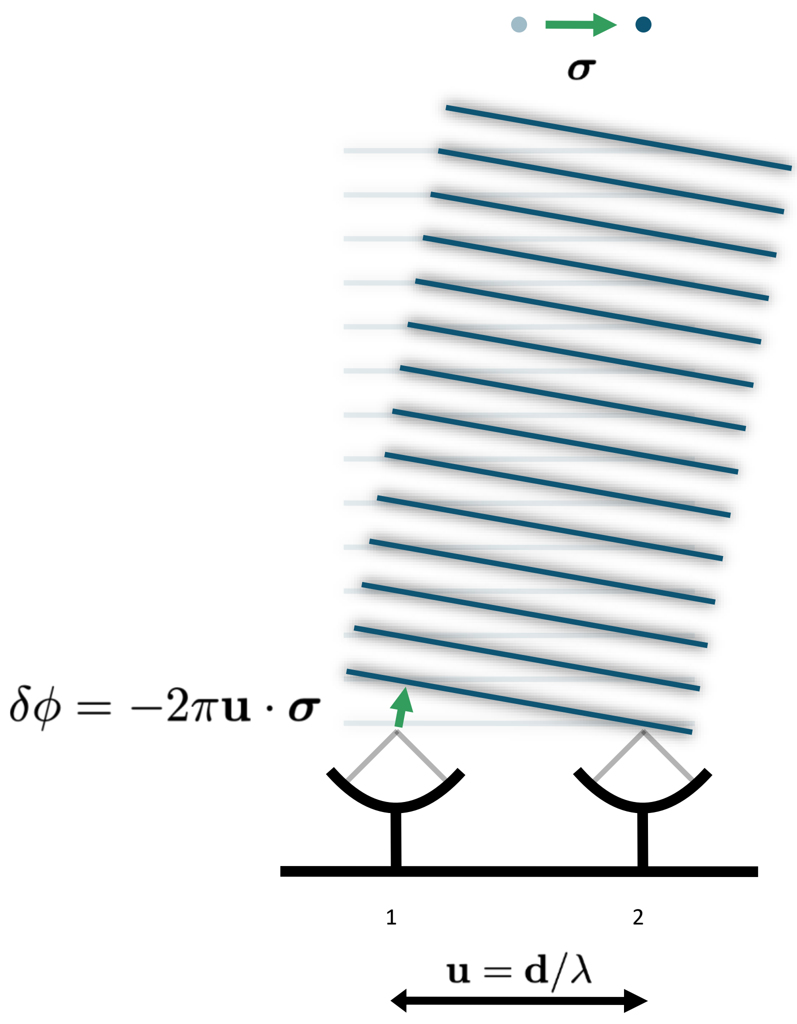


point source

$$\langle E_1 E_2^* \rangle = I_\nu$$

Sides from Lindy Blackburn

VLBI Measures "Visibilities", which correspond to **Spatial Coherence** of an EM Wavefront



point source

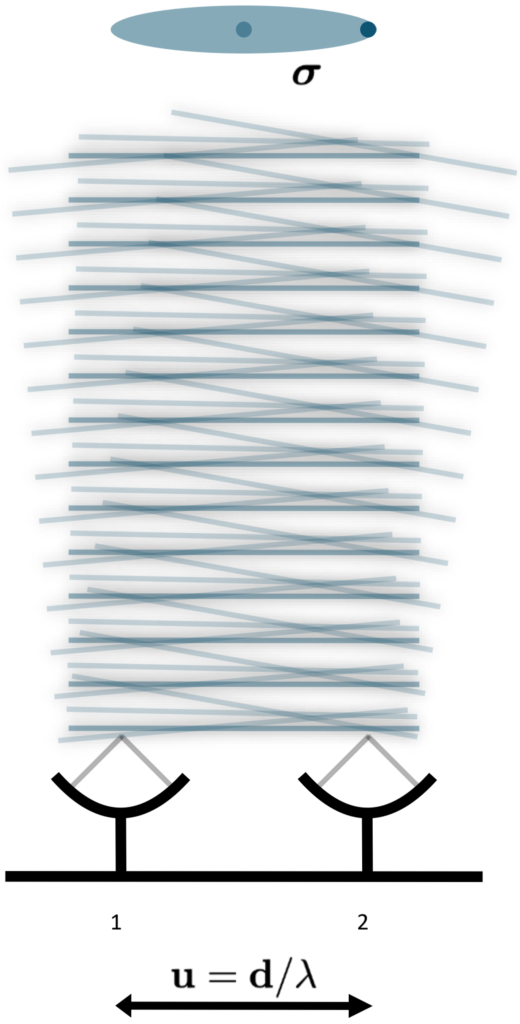
$$\langle E_1 E_2^* \rangle = I_\nu$$

shifted point source

$$\langle E_1 E_2^* \rangle = e^{-2\pi i \mathbf{u} \cdot \boldsymbol{\sigma}} I_\nu$$

Sides from Lindy Blackburn

VLBI Measures "Visibilities", which correspond to The **Fourier transform** of a sky image



point source

$$\langle E_1 E_2^* \rangle = I_\nu$$

shifted point source

$$\langle E_1 E_2^* \rangle = e^{-2\pi i \mathbf{u}\cdot\boldsymbol{\sigma}} I_\nu$$

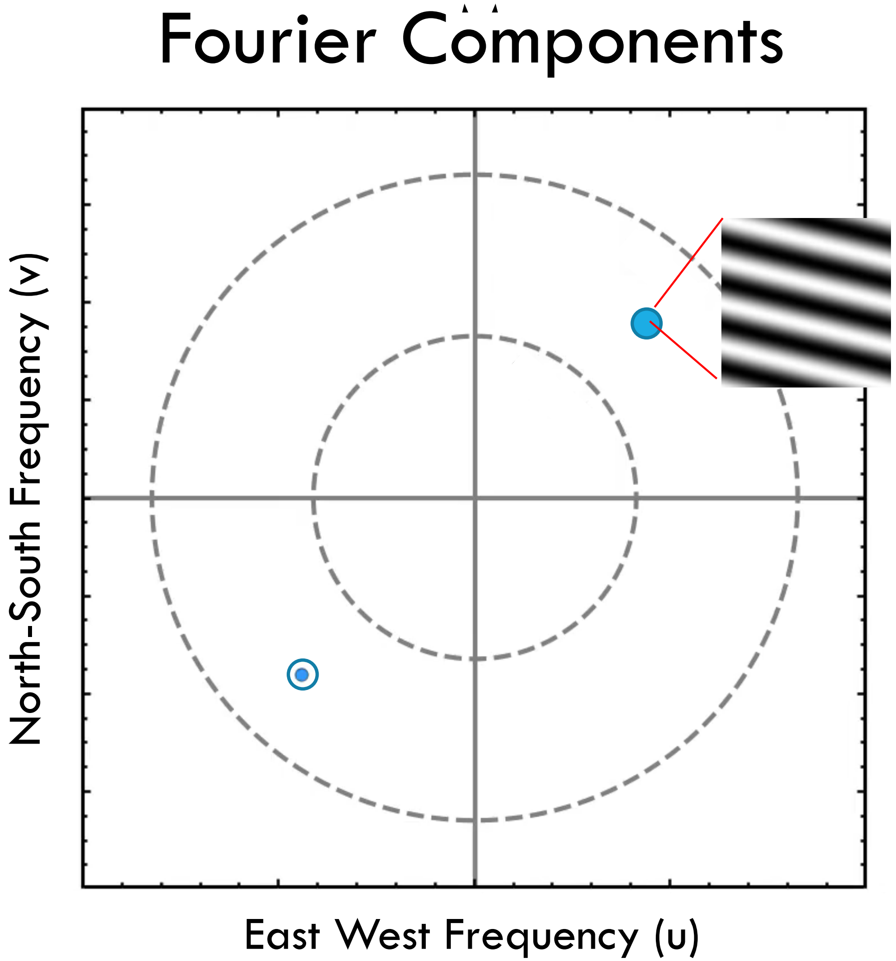
extended source (integration over many point sources)

$$\langle E_1 E_2^* \rangle = \int e^{-2\pi i \mathbf{u}\cdot\boldsymbol{\sigma}} I_\nu(\boldsymbol{\sigma}) d\Omega$$

$\boxed{= \mathcal{V}(\mathbf{u}) \text{“Visibility”}}$

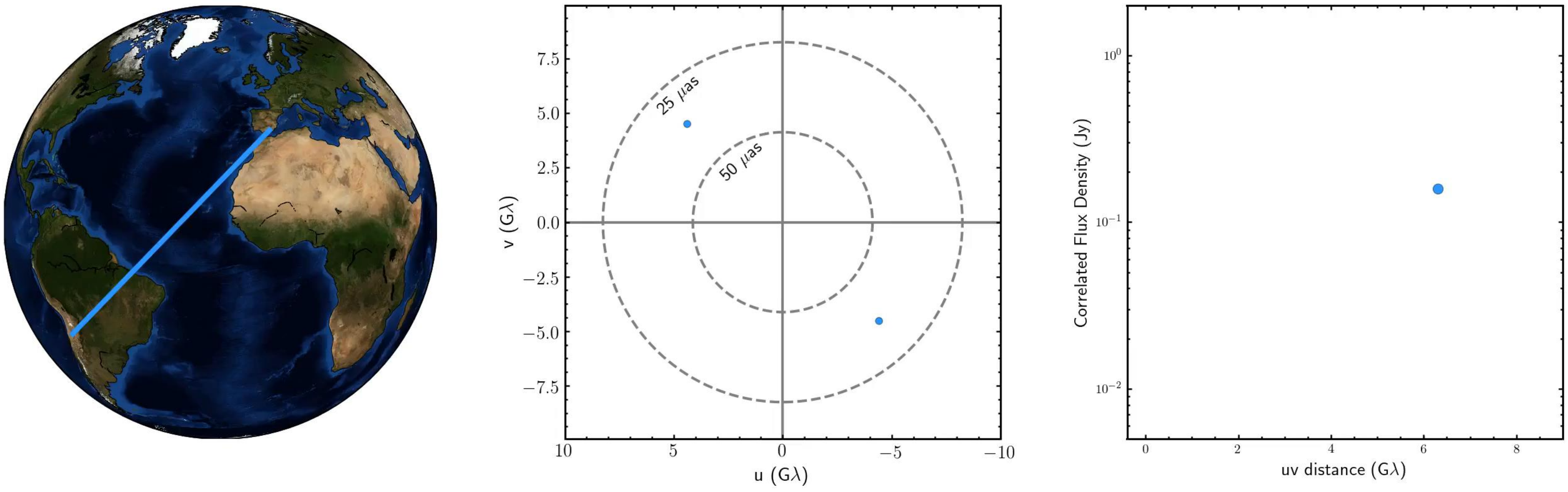
Sides from Lindy Blackburn

VLBI Measures **Fourier Components** of the sky image on **baselines** between telescopes



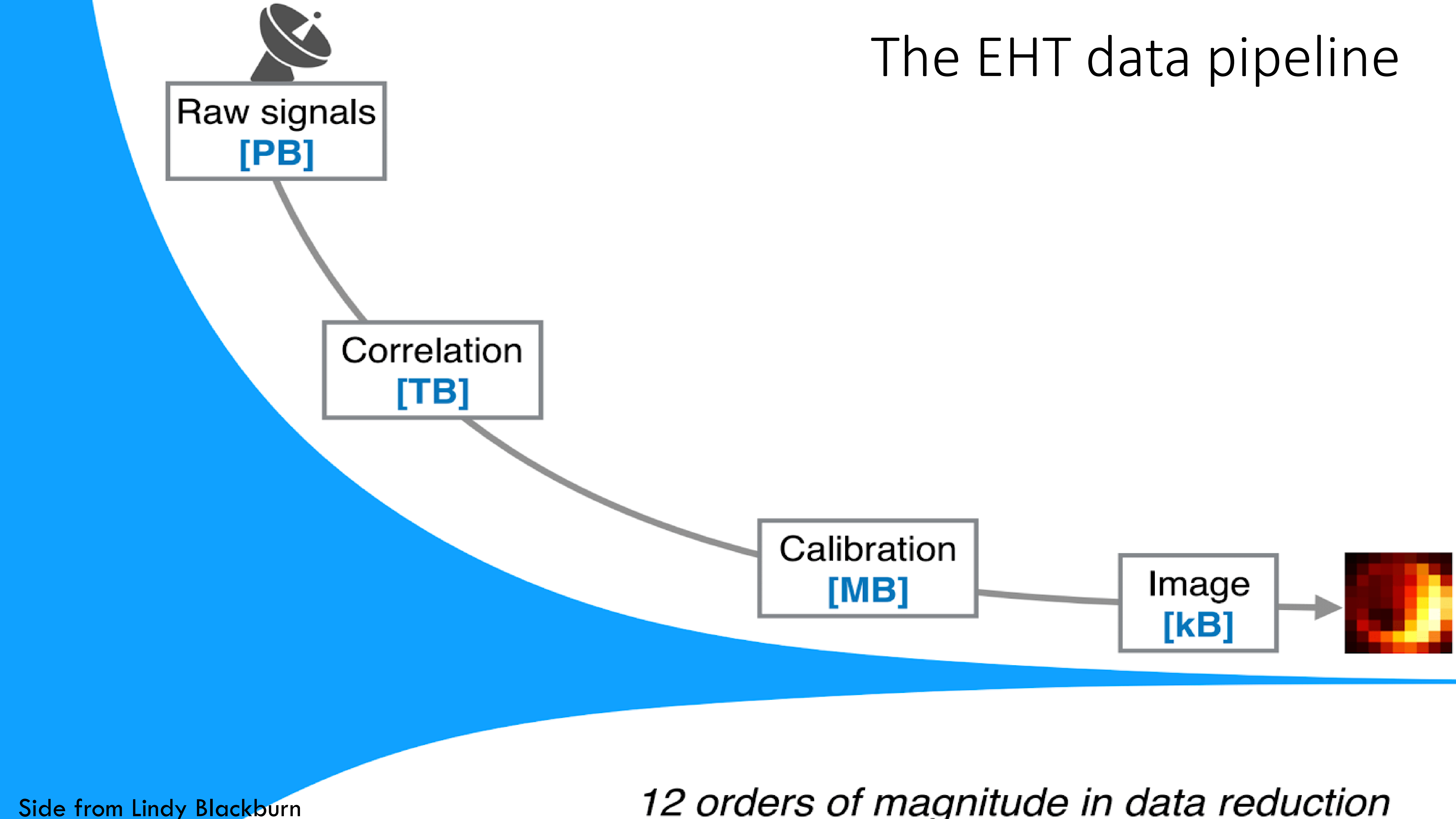
Slide credit: Katie Bouman

Earth’s Rotation provides more measurements



Animation credit: Daniel Palumbo

The EHT data pipeline



Side from Lindy Blackburn

12 orders of magnitude in data reduction

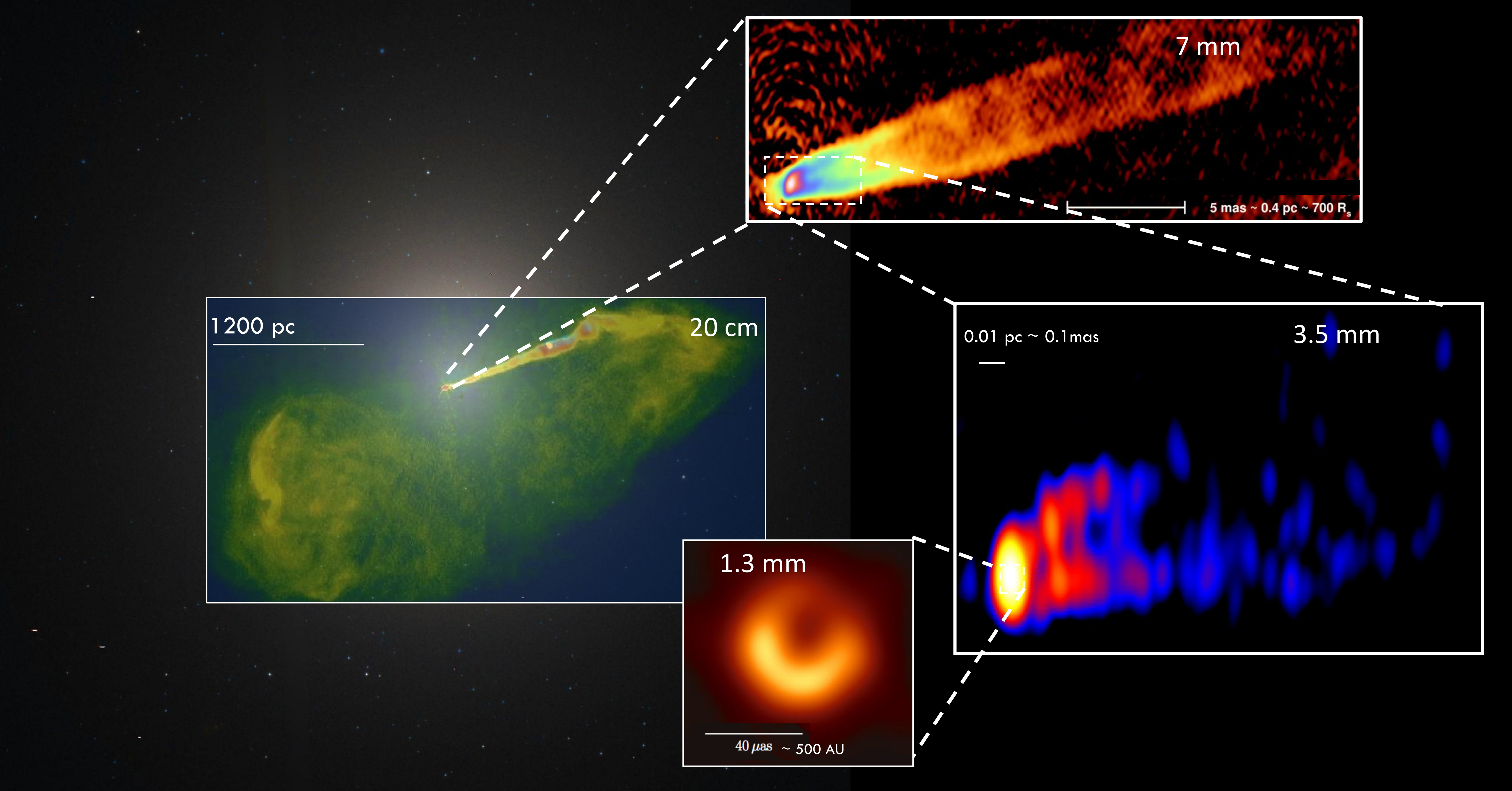
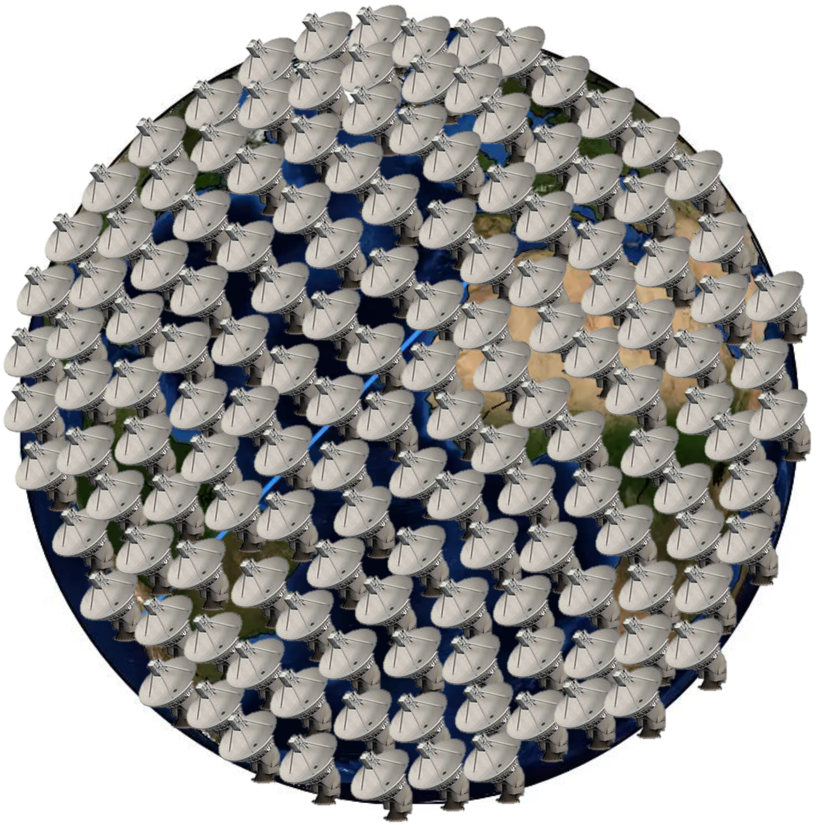


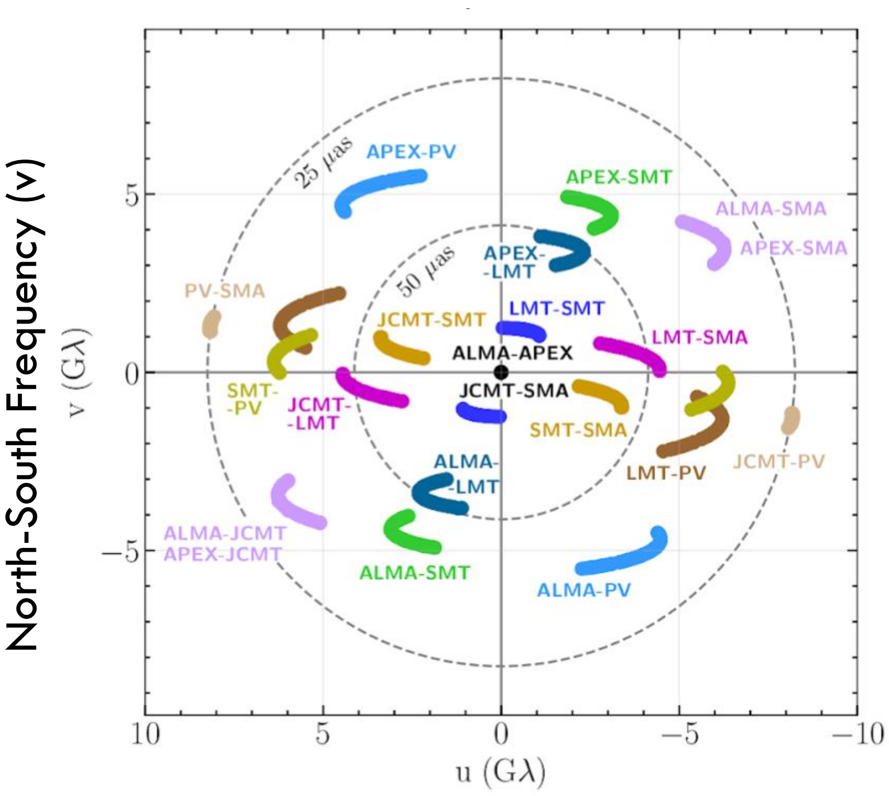
Image Credits: NRAO (VLA),
Craig Walker (7mm VLBA), Kazuhiro Hada (VLBA+GBT 3mm),
EHT (1.3 mm)

VLBI Imaging Methods

We don’t have enough measurements to directly image



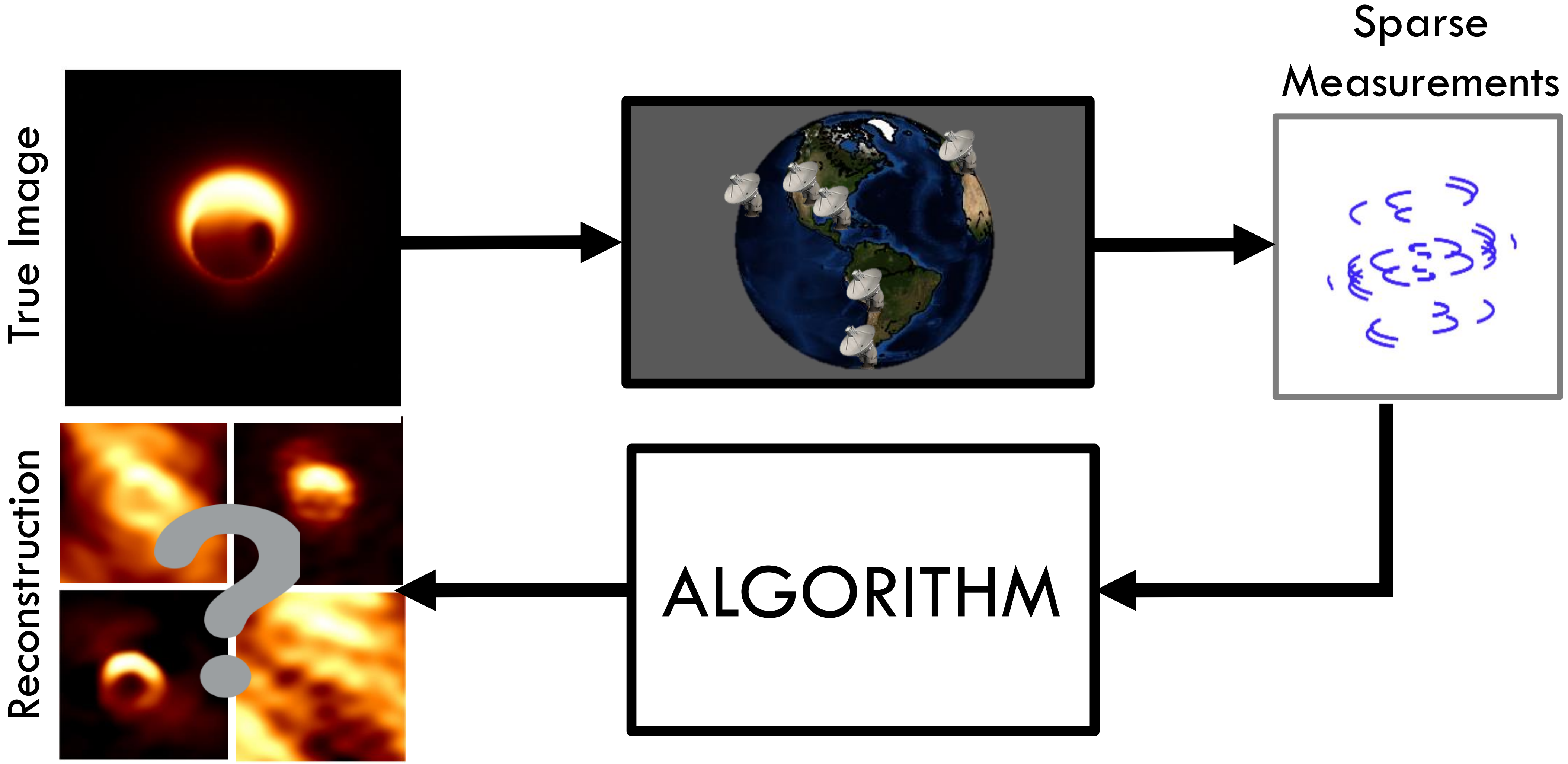
Frequency Measurements



East West Frequency (u)

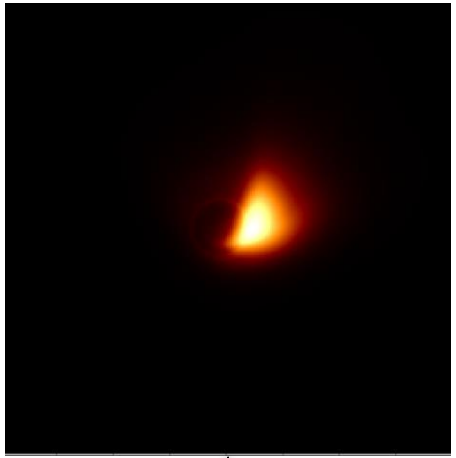
Slide credit: Katie Bouman

The Imaging Problem



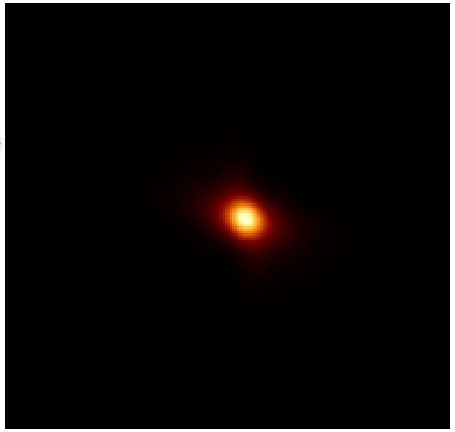
The Imaging Problem

Source Image



$\updownarrow \mathcal{F}$

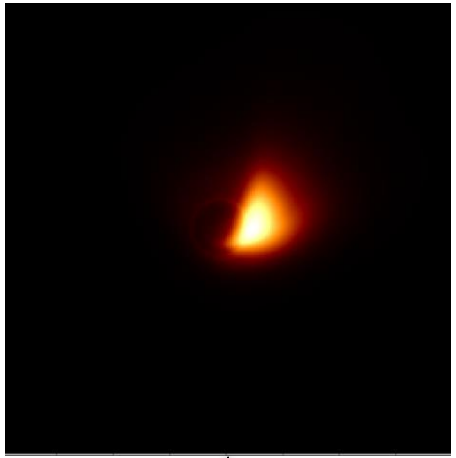
Fourier Transform



Simulation Credit: Avery Broderick

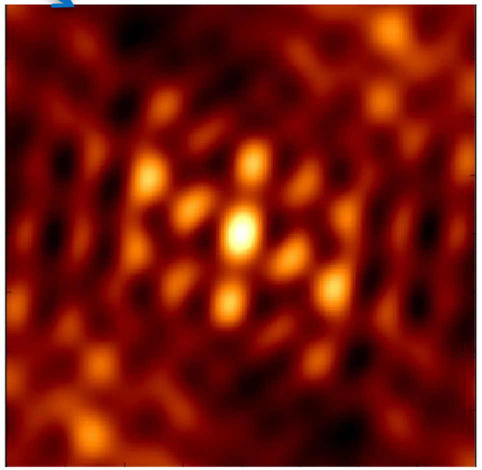
The Imaging Problem

Source Image



$*$

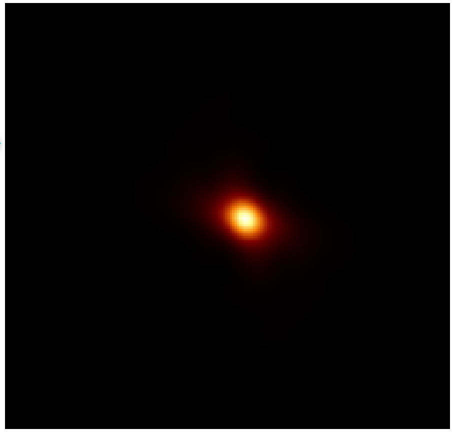
“Dirty Beam”



$\updownarrow \mathcal{F}$

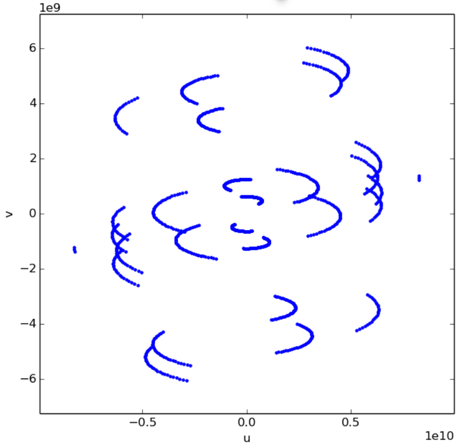
$\updownarrow \mathcal{F}$

Fourier Transform



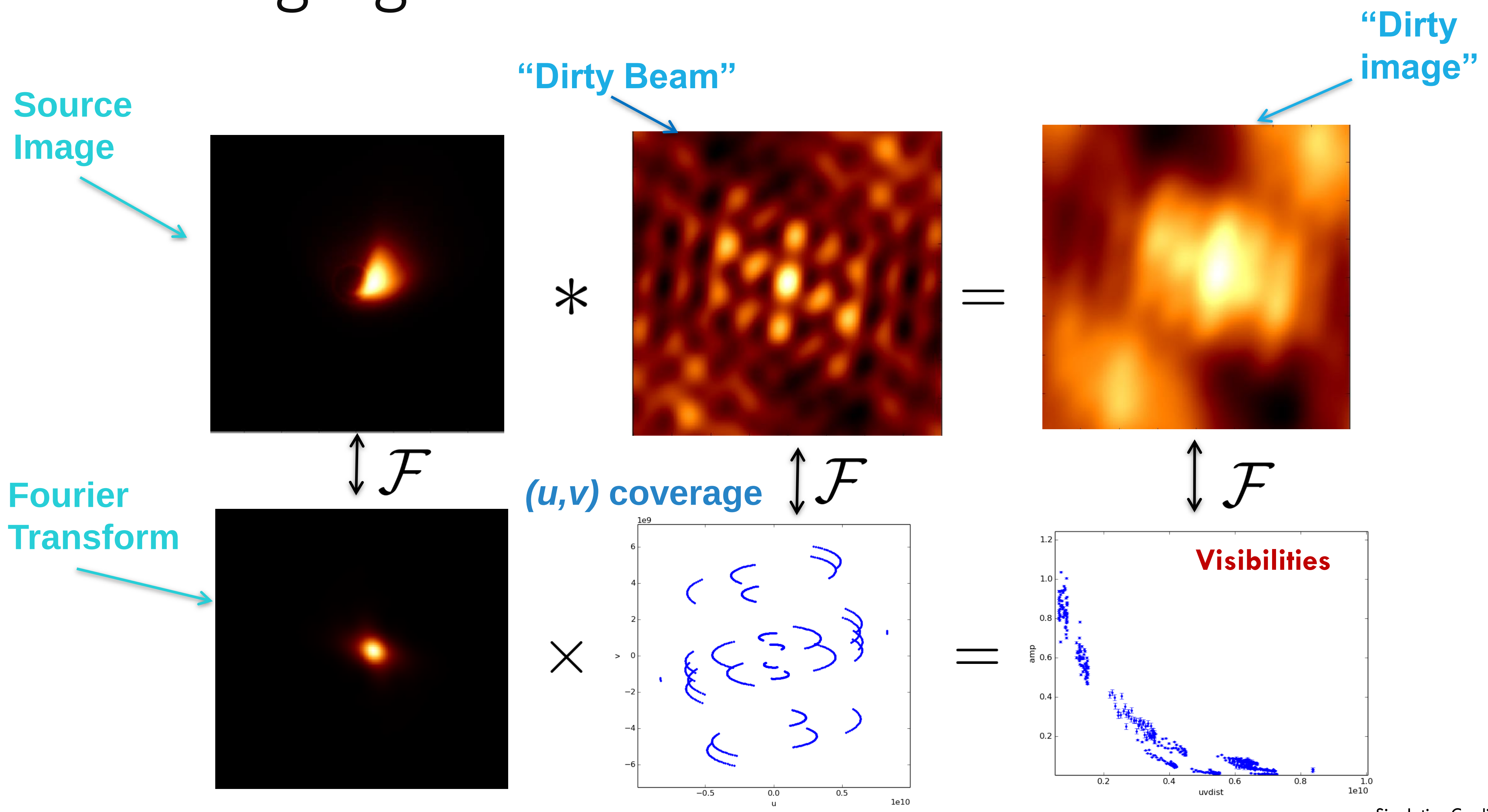
$\times$

***(u,v)* coverage**



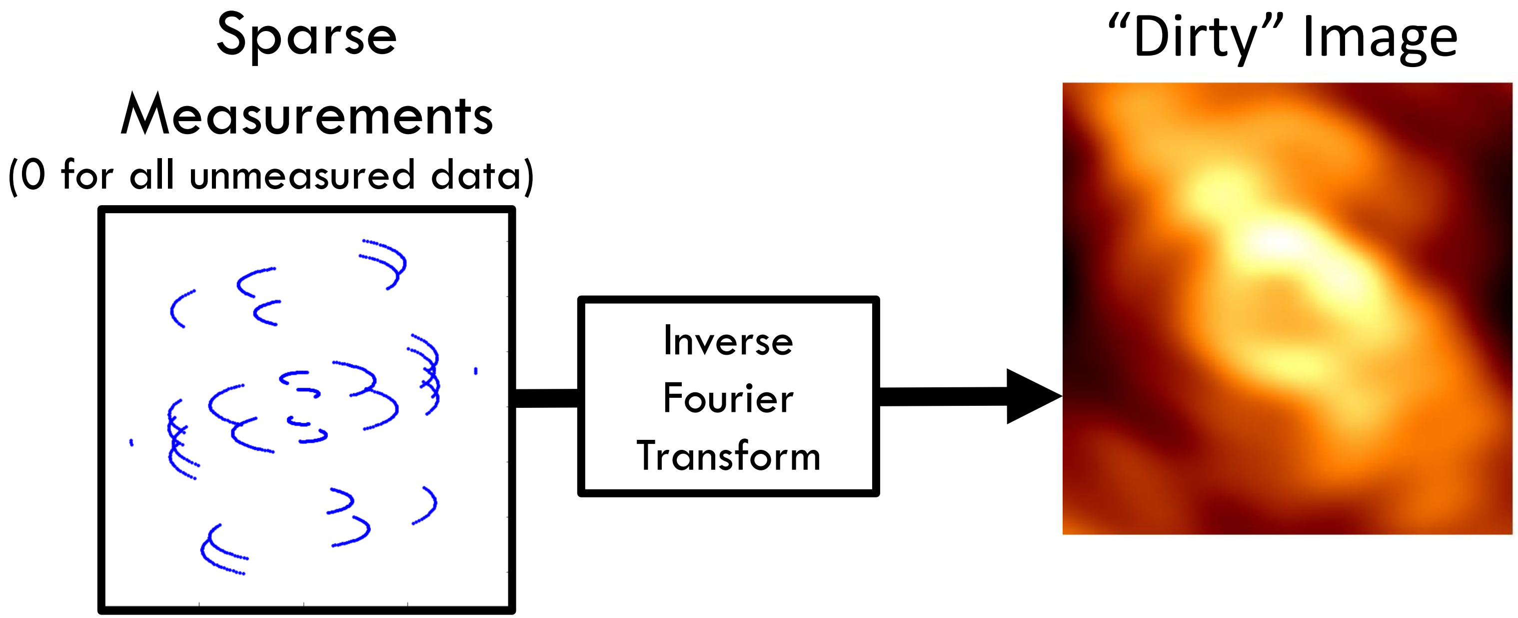
Simulation Credit: Avery Broderick

The Imaging Problem



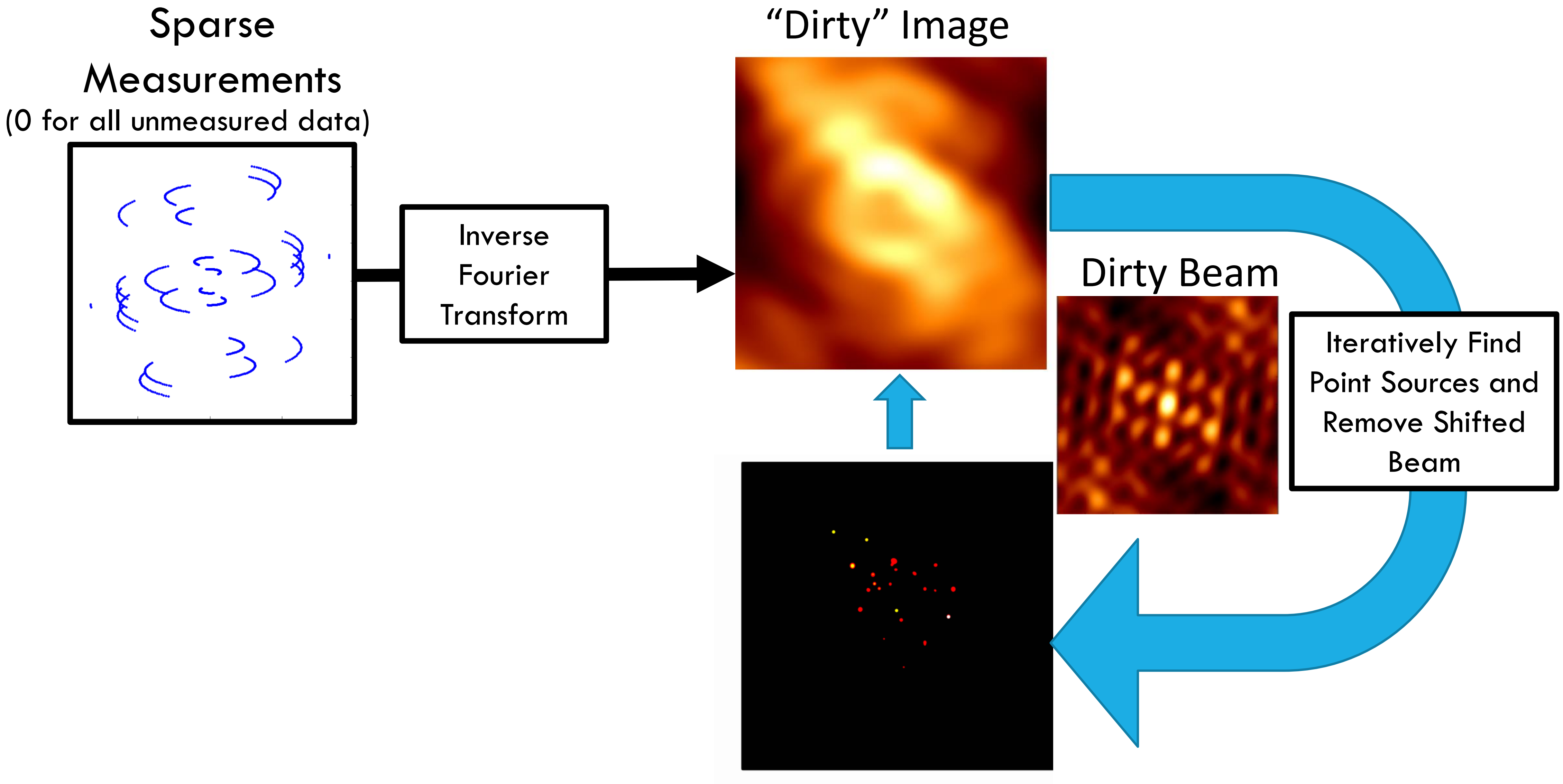
Simulation Credit: Avery Broderick

CLEAN Algorithm



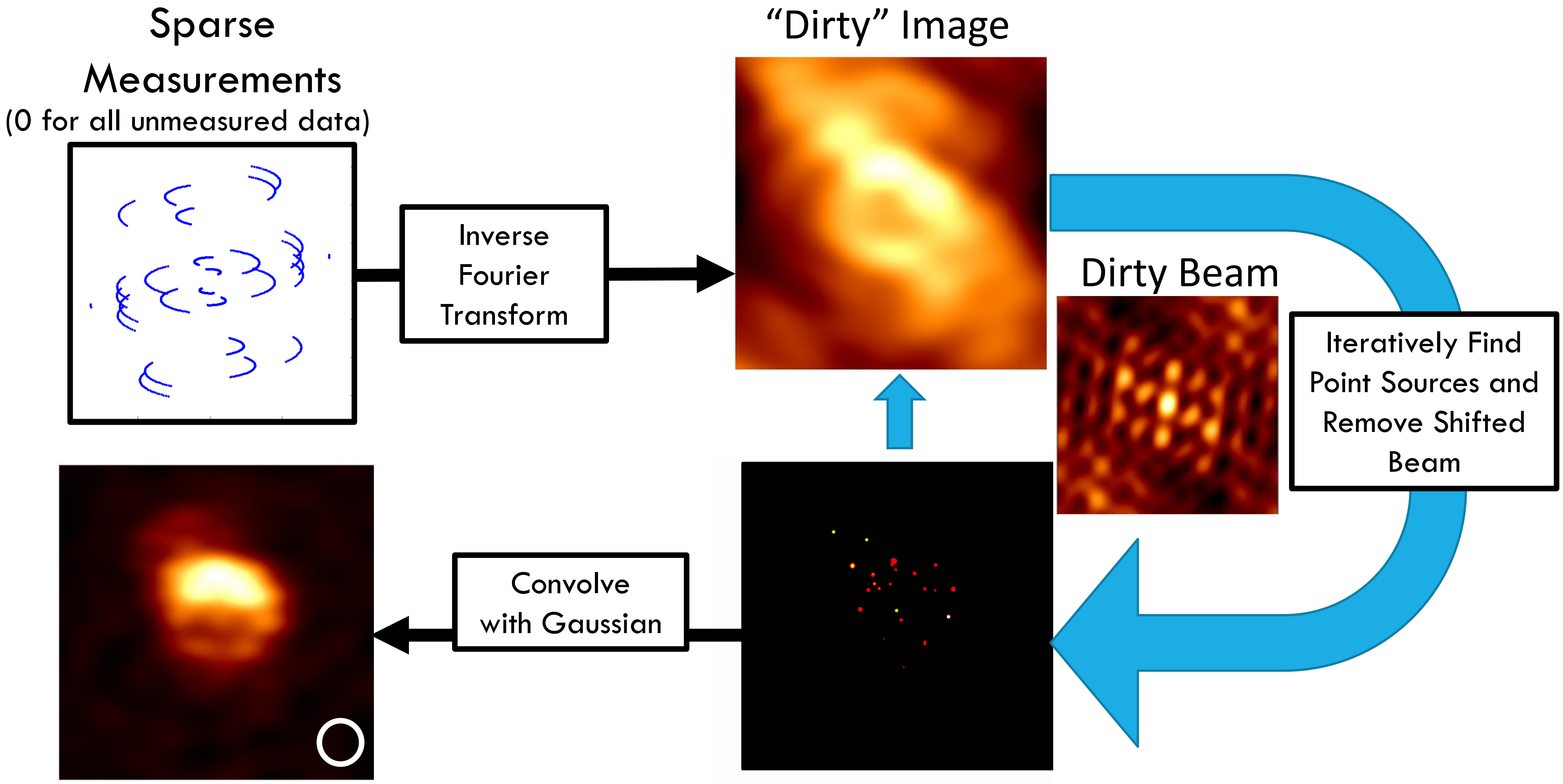
Simulation Credit: Avery Broderick

CLEAN Algorithm



Simulation Credit: Avery Broderick

CLEAN Algorithm



Simulation Credit: Avery Broderick

Pros of CLEAN:

1. In cases of good uv coverage, CLEAN produces images consistent with the data almost down to the noise level.

2. Each run of CLEAN takes a very short time

3. CLEAN is a standard, time-tested method and runs on a variety of platforms (Difmap, CASA, AIPS).

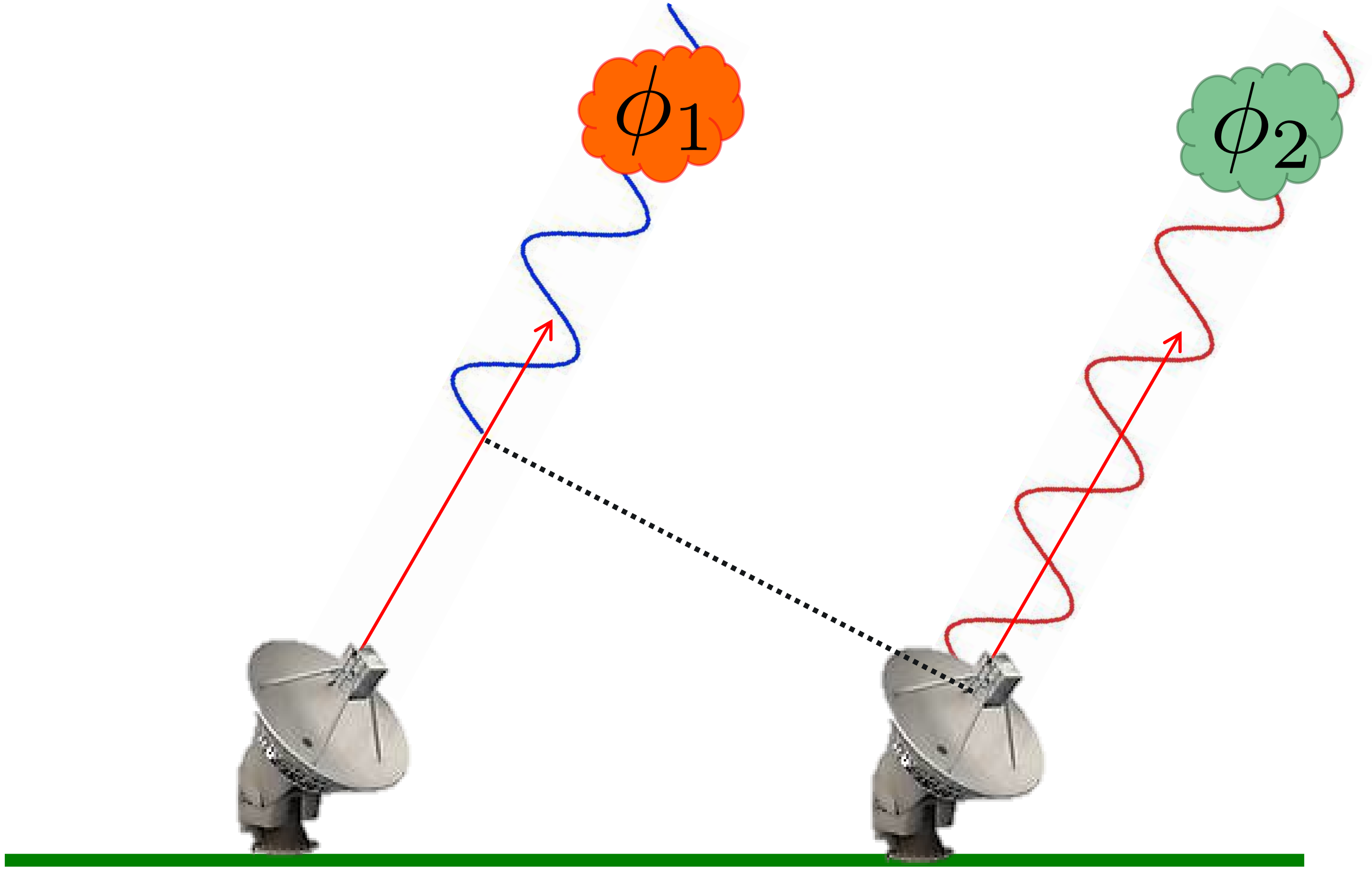
Cons of CLEAN:

1. CLEAN tends to break up extended features into multiple smaller features.

2. The final, “restored” image will not fit the data

3. CLEAN requires phase-calibrated data
    - EHT and other high frequency VLBI data requires a “self calibration” process

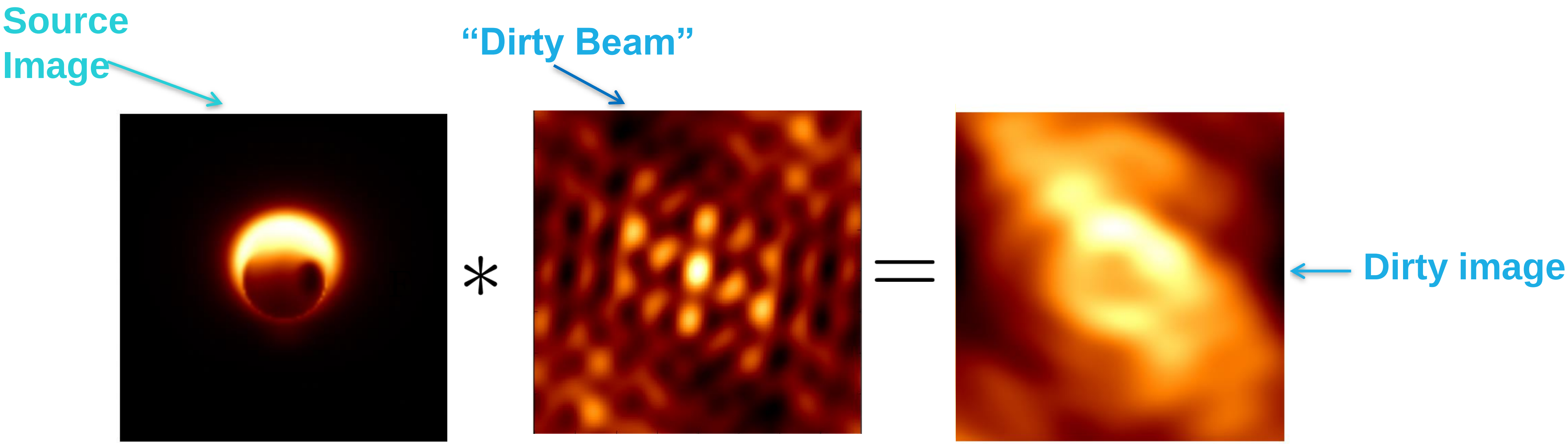
Phase Error from the atmosphere



$$V_{\text{measured}} = e^{i(\phi_1 - \phi_2)} V_{\text{true}}$$

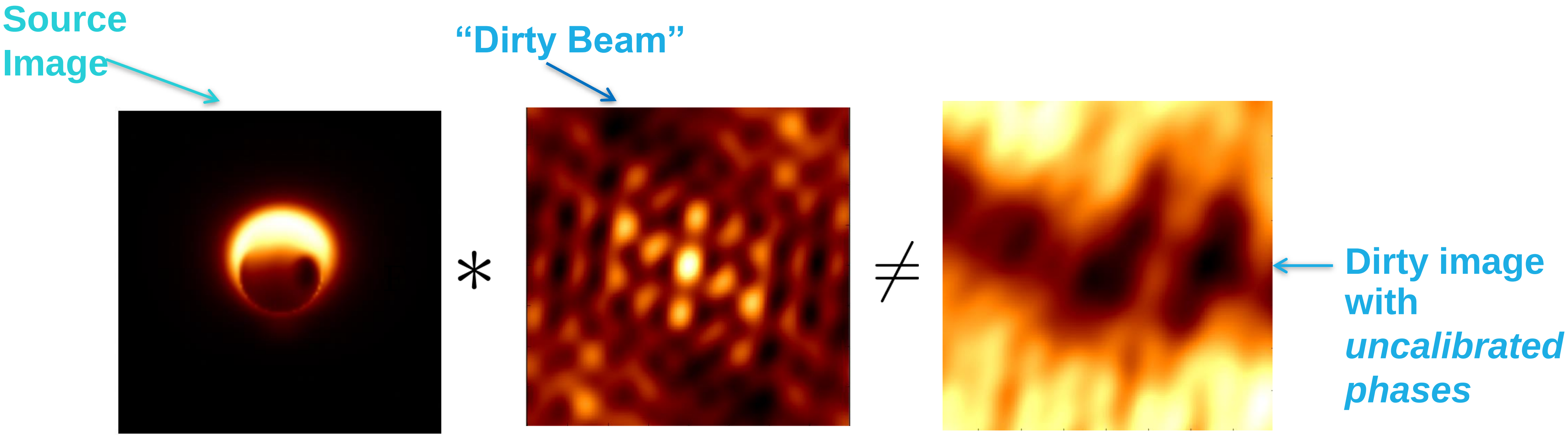
Figure credit: Katie Bouman

The importance of phase



Simulation Credit: Avery Broderick

The importance of phase



Simulation Credit: Avery Broderick

Closure Phase is a robust observable

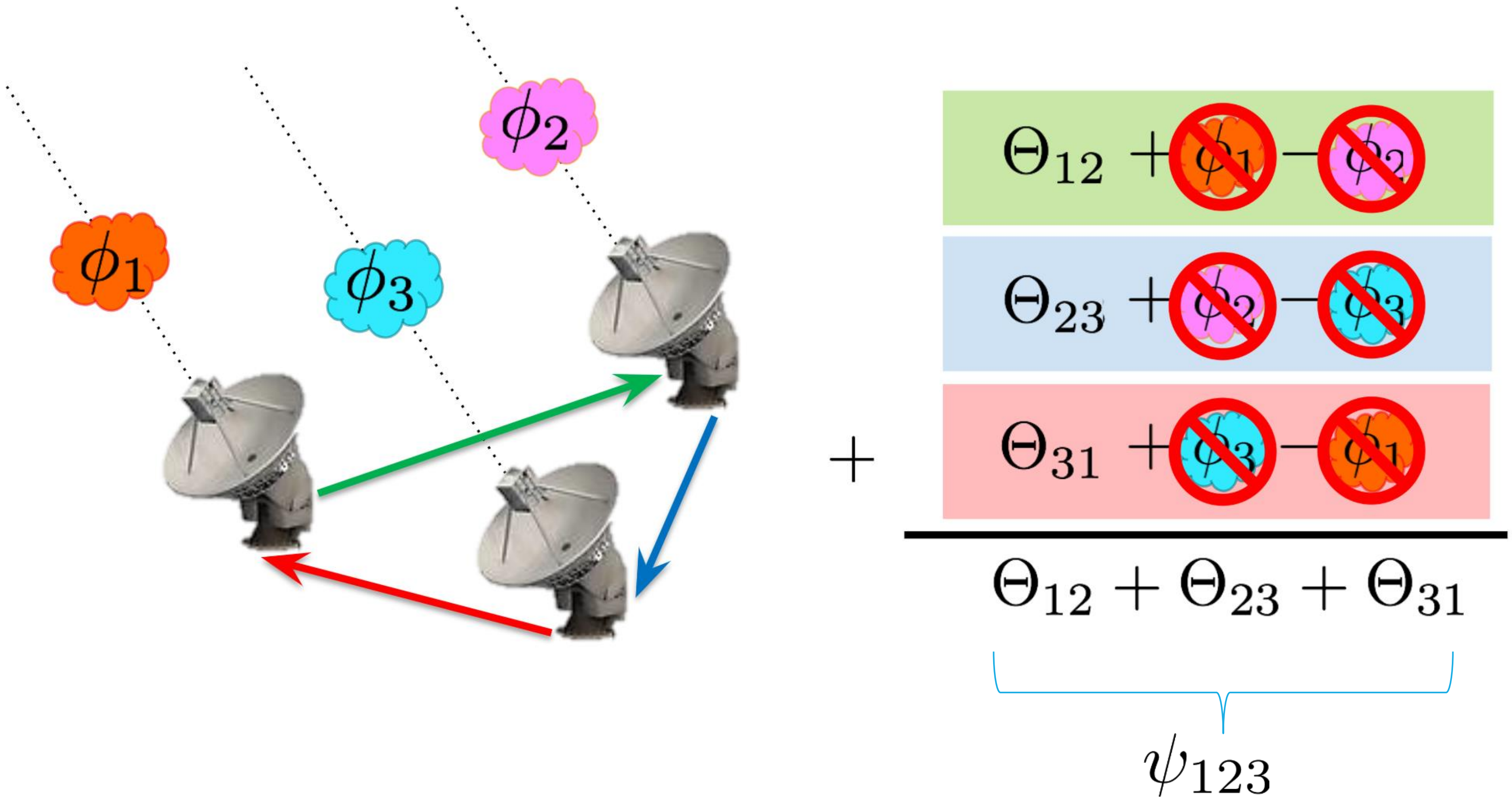


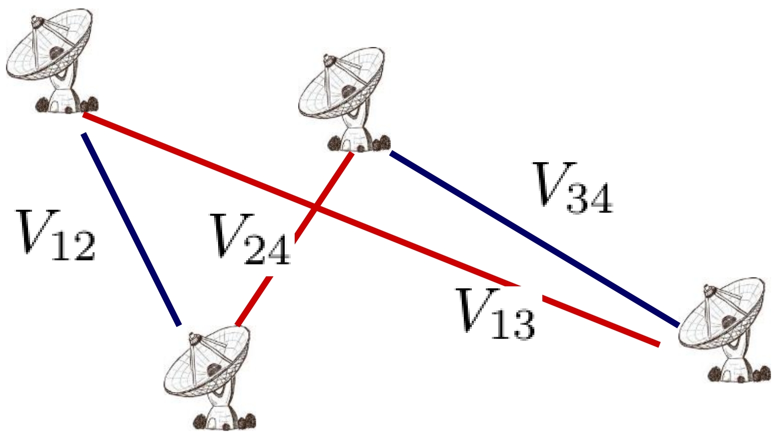
Figure credit: Katie Bouman

Amplitude gain errors and **Closure Amplitudes**

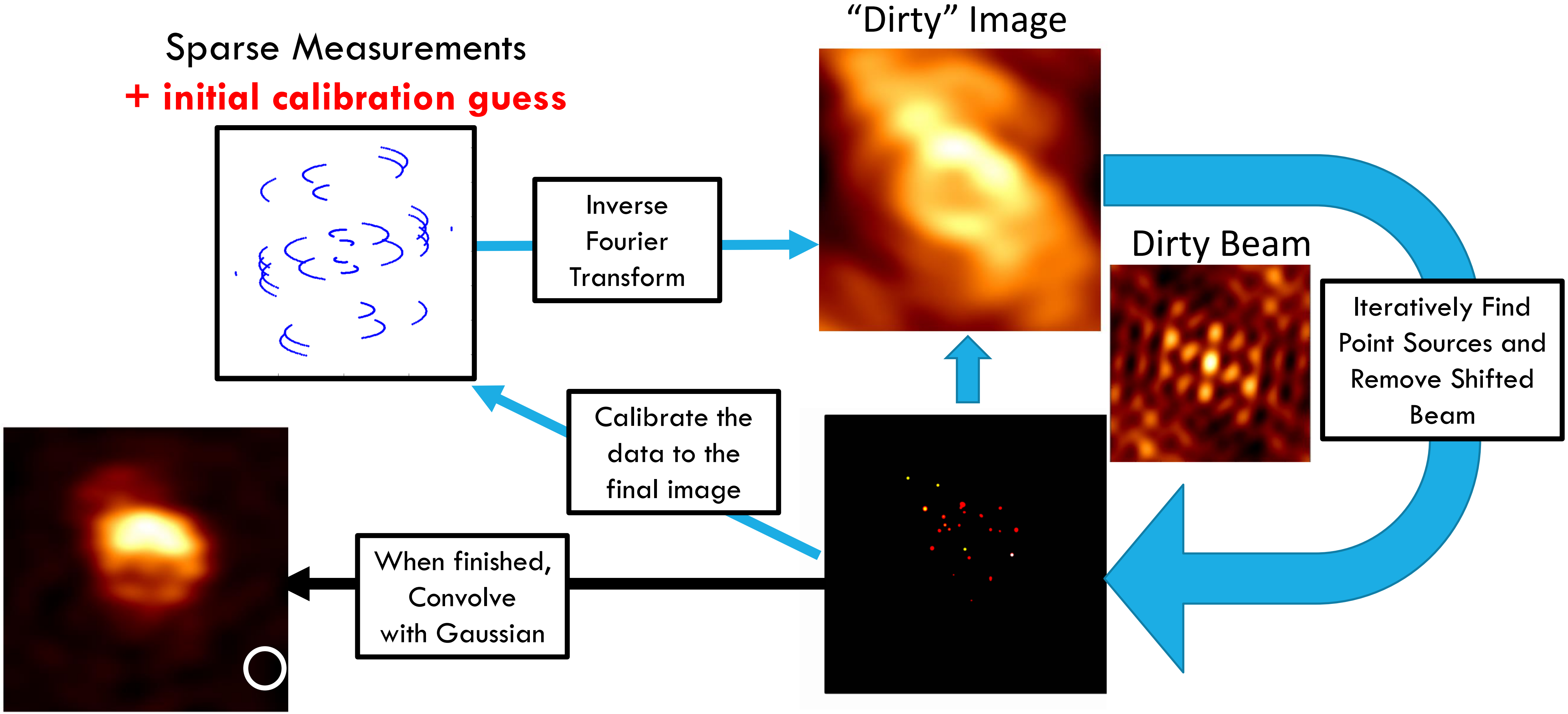
- In addition to the loss of phase from the atmosphere, individual telescopes can also have imperfect amplitude calibration

$$V_{\text{measured}} = G_1 G_2 e^{i(\phi_1 - \phi_2)} \, V_{\text{true}}$$

- **Closure amplitudes** are invariant to these gain errors



Dealing with Amplitude and phase calibration: CLEAN + **Self Calibration loops**



Simulation Credit: Avery Broderick

Another Imaging Approach: Bayesian Model Inversion

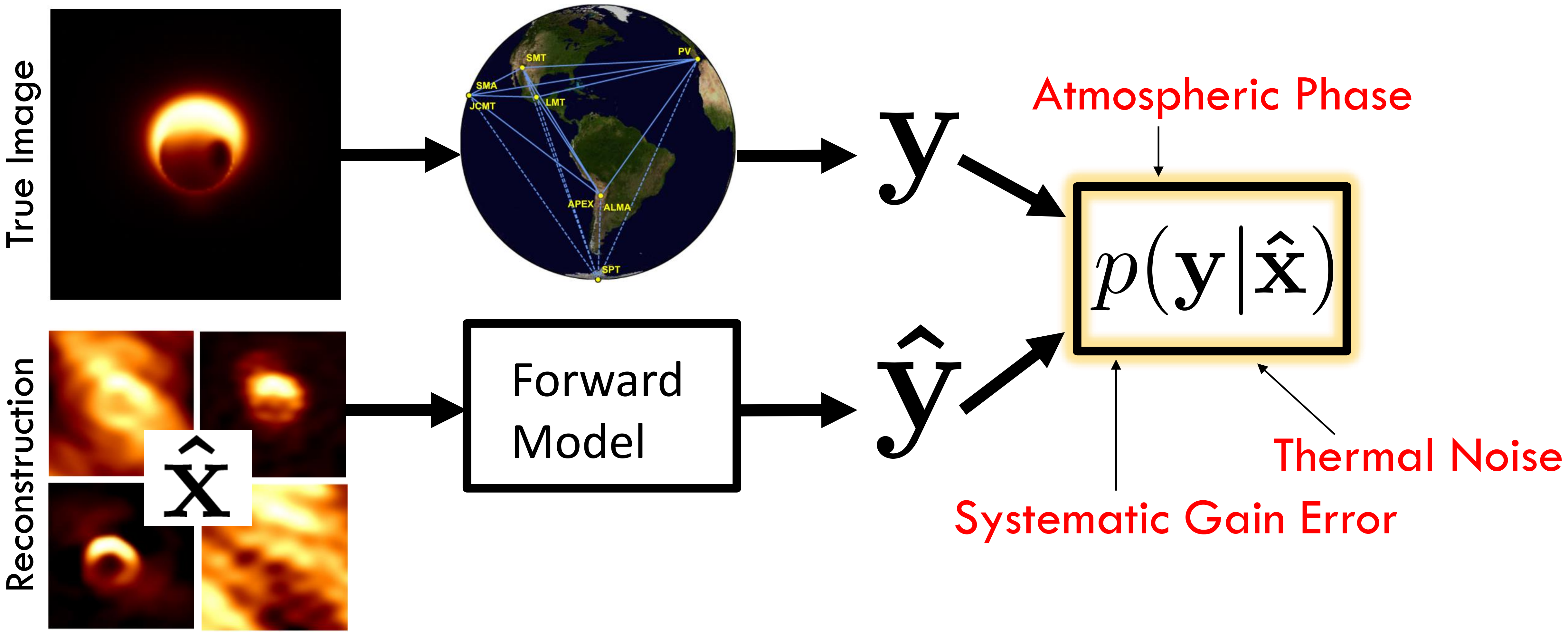


Image Credit: Katie Bouman

Simulation Credit: Avery Broderick

Bayesian Model Inversion

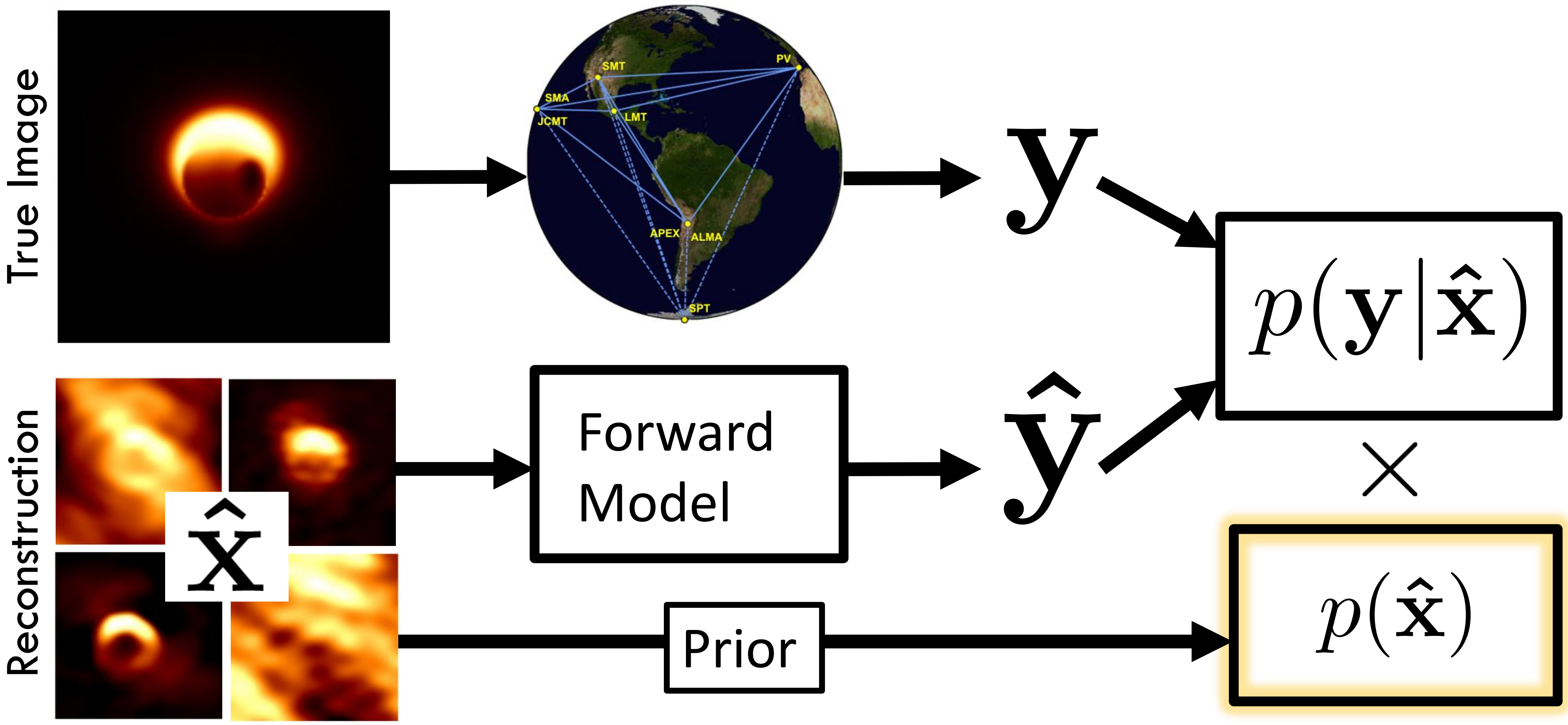


Image Credit: Katie Bouman

Simulation Credit: Avery Broderick

Regularized Maximum Likelihood

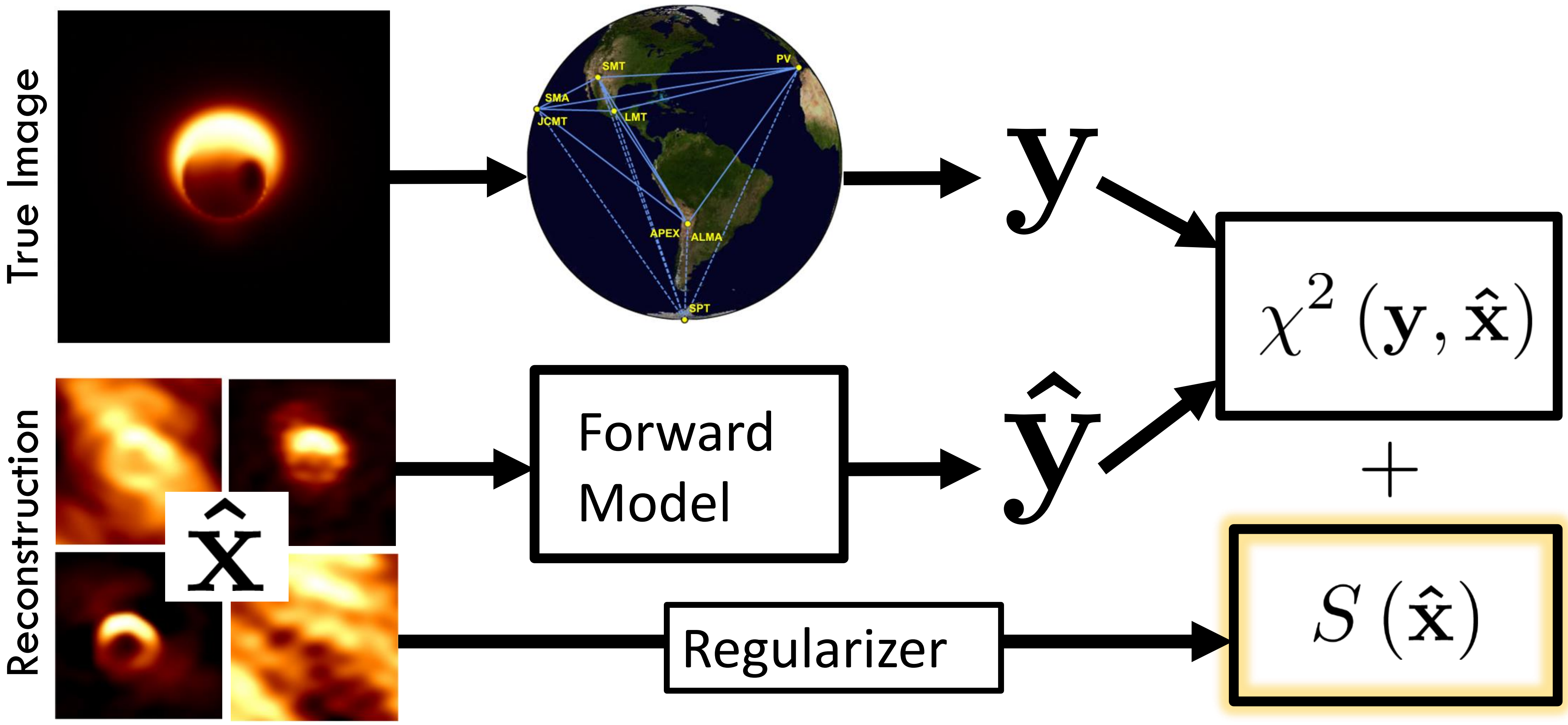


Image Credit: Katie Bouman
Simulation Credit: Avery Broderick

Imaging with Regularized Maximum Likelihood

“hyperparameters”

Minimize: $$J(\mathbf{I}) = \sum_{\text{data terms}} \alpha_D \chi^2_D\left(\mathbf{I}, \mathbf{d}\right) - \sum_{\text{regularizers}} \beta_R S_R\left(\mathbf{I}\right).$$

Any **data product**
(with approx. Gaussian errors)

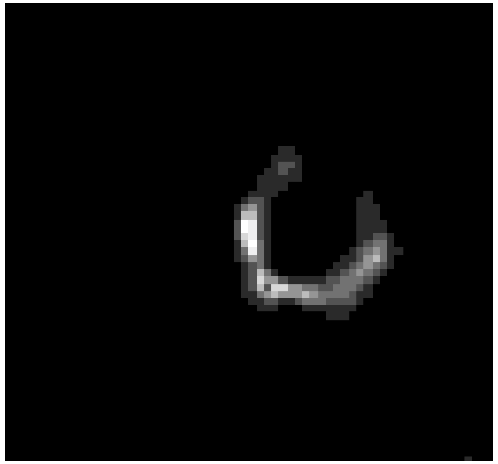
Regularizers

- Flexible framework enables development of new data and regularizer terms
- **Hyperparameters** weight relative importance of the different terms.

Example Regularizer terms:

L1 norm:
Minimizes total number of bright pixels

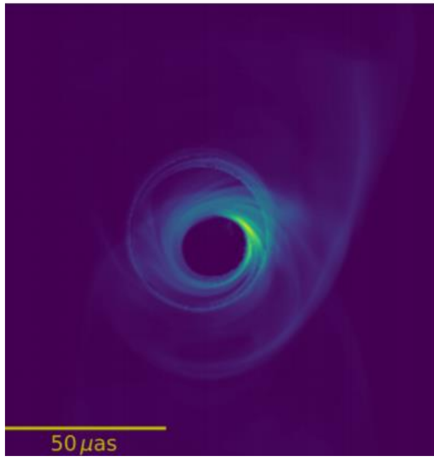
$$S_{\ell 1} = -\frac{1}{\zeta} \sum_i |I_i|$$



TV: prefers piecewise flat patches and sparse image gradients

$$S_{TV} = -\frac{1}{\zeta} \sum_l \sum_m \left[(I_{l+1,m} - I_{l,m})^2 + (I_{l,m+1} - I_{l,m})^2 \right]^{1/2}$$

RML imaging:
we can use robust closure data directly



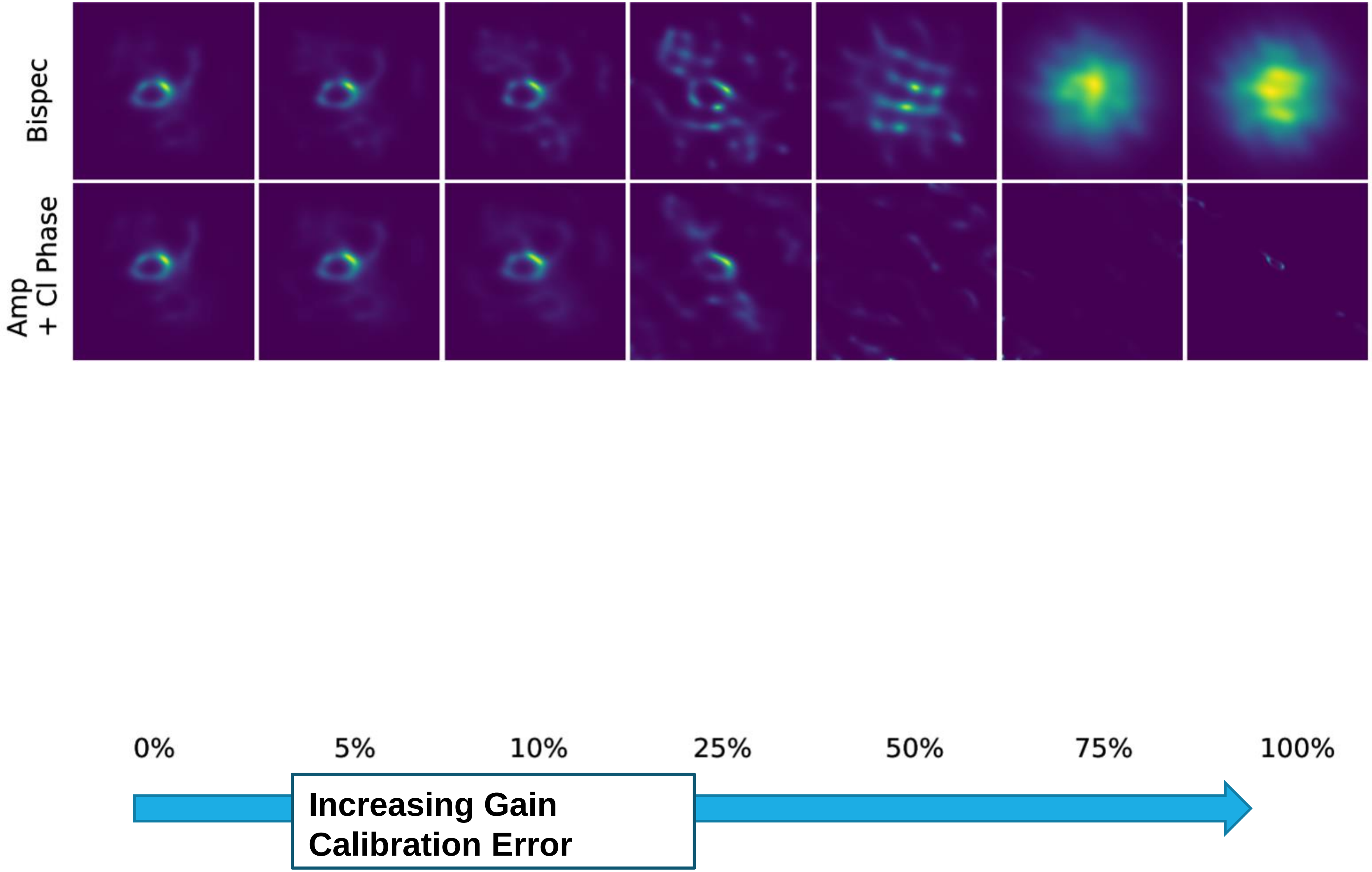
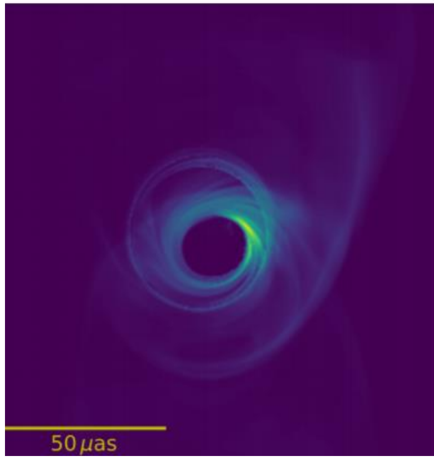


Image Credit: Chael+ 2018a
Simulation Credit: Roman Gold

RML imaging:
we can use robust closure data directly



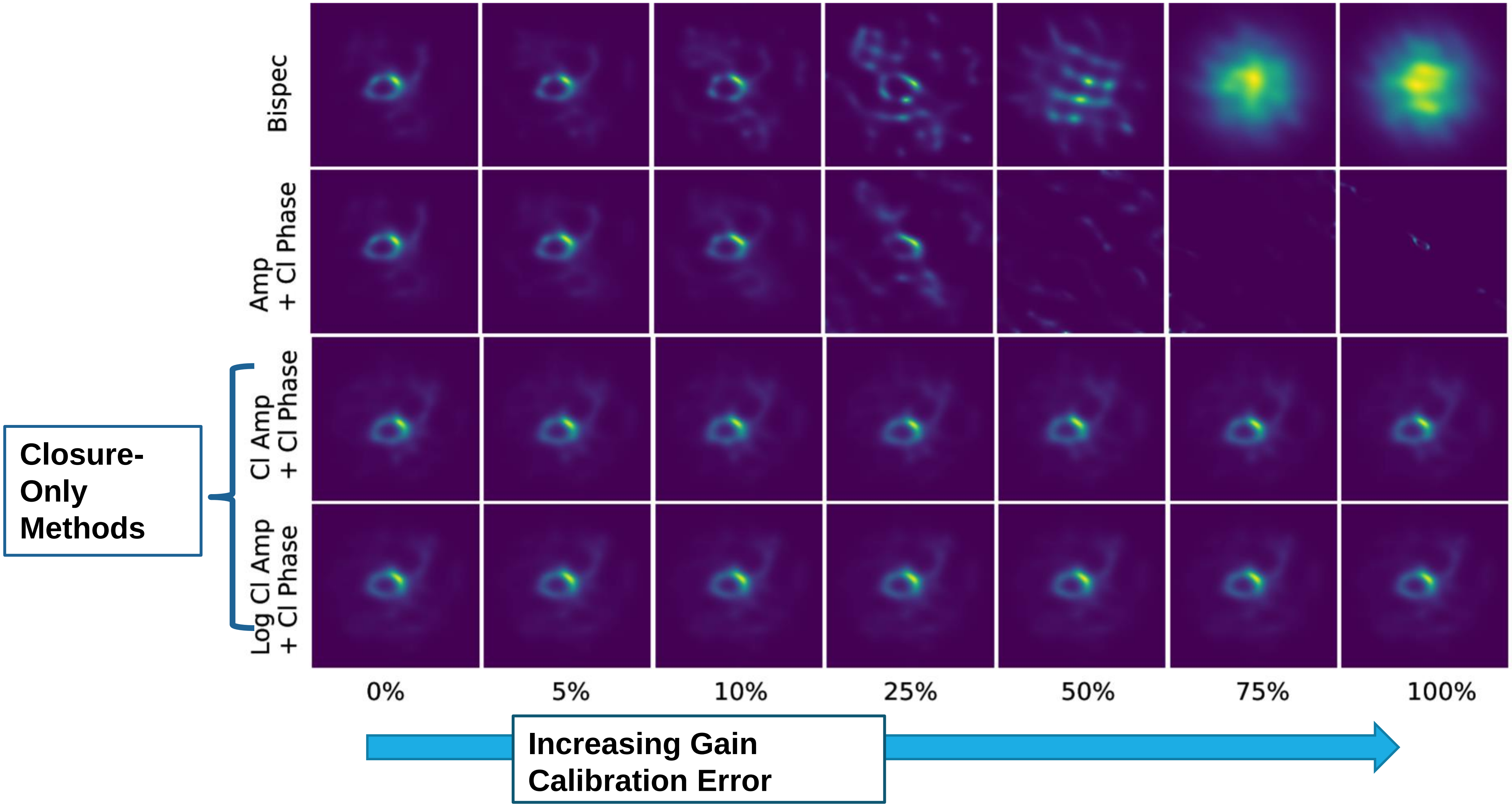
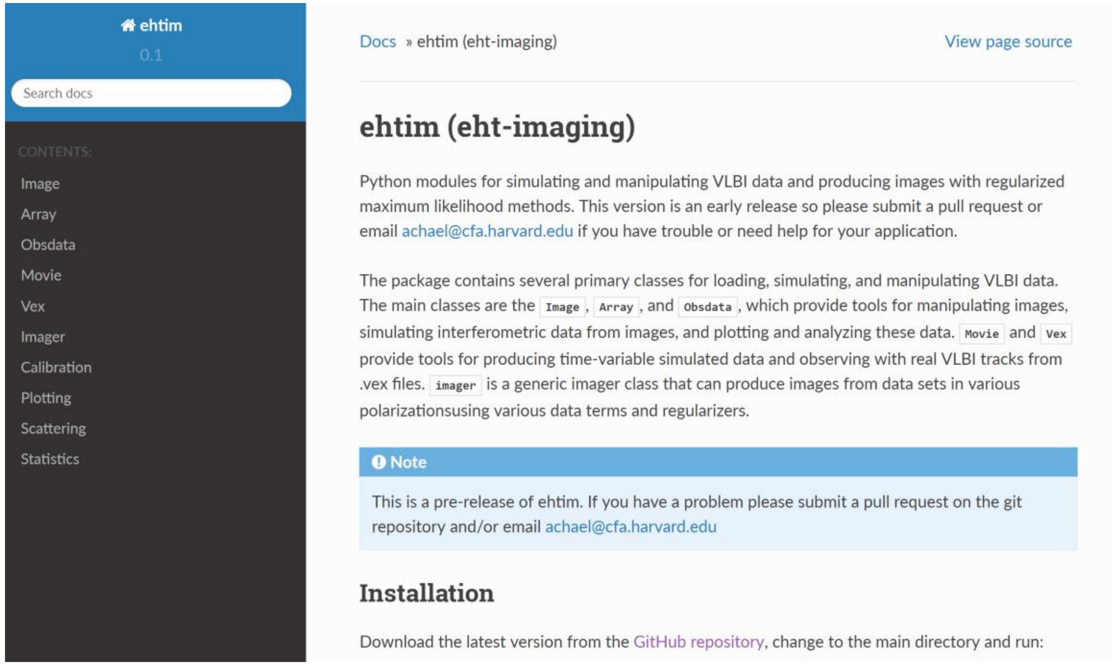


Image Credit: Chael+ 2018a
Simulation Credit: Roman Gold

RML Imaging software developed for the EHT

eht-imaging: Chael +



ehtim (eht-imaging)

Python modules for simulating and manipulating VLBI data and producing images with regularized maximum likelihood methods. This version is an early release so please submit a pull request or email achael@cfa.harvard.edu if you have trouble or need help for your application.

The package contains several primary classes for loading, simulating, and manipulating VLBI data. The main classes are the `Image`, `Array`, and `Obsdata`, which provide tools for manipulating images, simulating interferometric data from images, and plotting and analyzing these data. `Movie` and `Vex` provide tools for producing time-variable simulated data and observing with real VLBI tracks from .vex files. `imager` is a generic imager class that can produce images from data sets in various polarizationsusing various data terms and regularizers.

Note

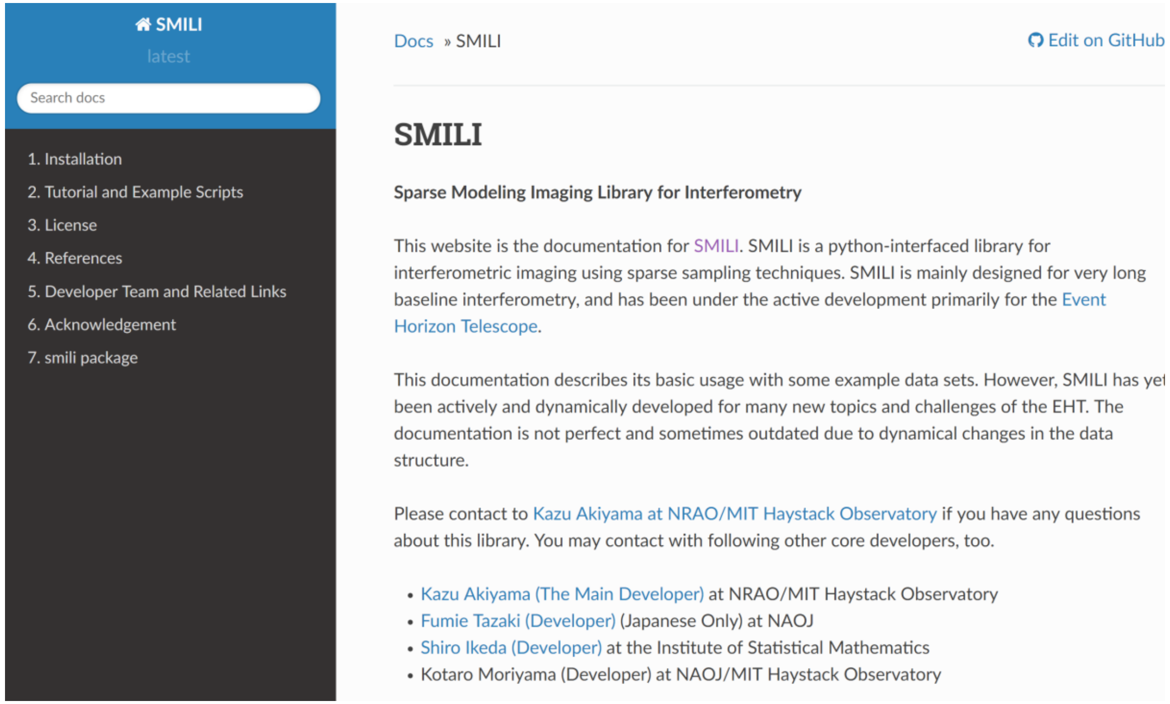
This is a pre-release of ehtim. If you have a problem please submit a pull request on the git repository and/or email achael@cfa.harvard.edu

Installation

Download the latest version from the GitHub repository, change to the main directory and run:

https://github.com/achael/eht-imaging

SMILI: Akiyama+



SMILI

Sparse Modeling Imaging Library for Interferometry

This website is the documentation for SMILI. SMILI is a python-interfaced library for interferometric imaging using sparse sampling techniques. SMILI is mainly designed for very long baseline interferometry, and has been under the active development primarily for the Event Horizon Telescope.

This documentation describes its basic usage with some example data sets. However, SMILI has yet been actively and dynamically developed for many new topics and challenges of the EHT. The documentation is not perfect and sometimes outdated due to dynamical changes in the data structure.

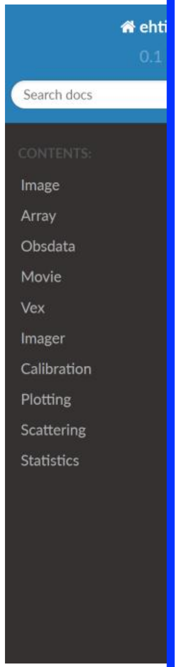
Please contact to Kazu Akiyama at NRAO/MIT Haystack Observatory if you have any questions about this library. You may contact with following other core developers, too.

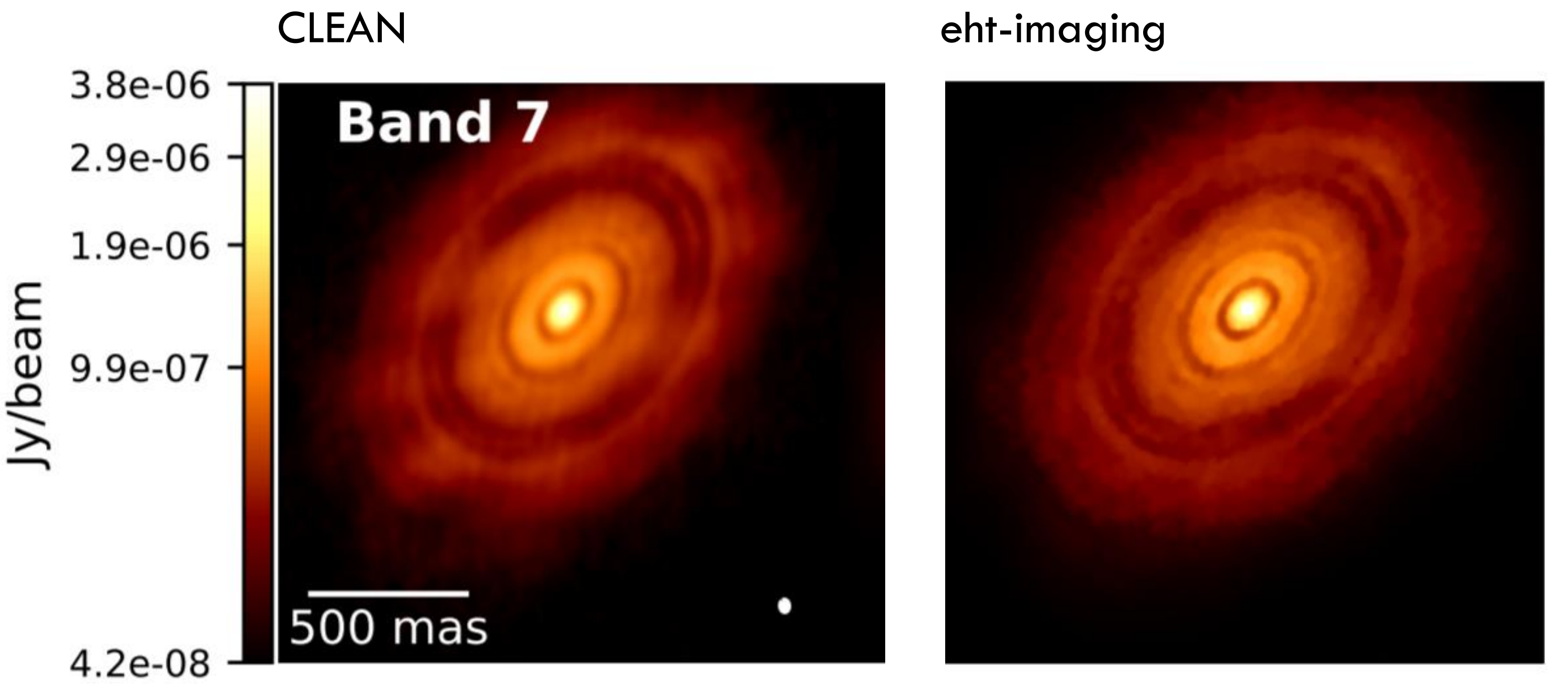
- Kazu Akiyama (The Main Developer) at NRAO/MIT Haystack Observatory
- Fumie Tazaki (Developer) (Japanese Only) at NAOJ
- Shiro Ikeda (Developer) at the Institute of Statistical Mathematics
- Kotaro Moriyama (Developer) at NAOJ/MIT Haystack Observatory

https://github.com/astrosmili/smili

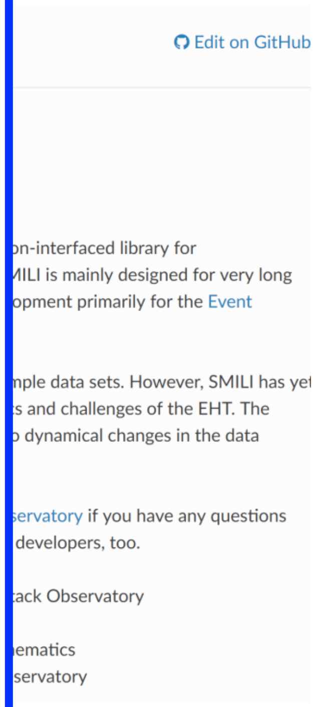
RML Imaging software developed for the EHT -- but with wide applicability

eht-im





ALMA Partnership+ 2015, Chael+ 2018



mili

An example eht-imaging script

1.) First we **define our objective function** using an observation, data term, and regularizer weights.

(We also choose the initial image, prior image (if used), maximum number of iterations, systematic noise to add to the error budget ….)

2.) Imaging is usually done in rounds followed by blurring the result and restarting from the blurred image. **Blurring and restarting** helps us escape local minima.

(Sometimes thresholding is also helpful to remove noisy off-source flux)

3.) We often **self-calibrate** to a result obtained from closure quantities and then continue imaging incorporating complex visibilities into the fit.

```
# Define the imager with an observation, initial gaussian, and data & regularizer  weights
imgr = eh.imager.Imager(observation, initial_gaussian, prior_im=initial_gaussian,
                        data_term={'amp':10, 'cphase':100, 'logcamp':100},
                        reg_term={'flux':10,'cm':10,'l1':100,'tv2':10},
                        maxit=500, systematic_noise=.1)

# Imaging, blurring, and re-imaging function for convergence
def converge(imgr):
    for repeat in range(maj_cycles):
        imgr.init_next = imgr.out_last().blur_circ(res)
        imgr.make_image_I(show_updates=False)

        for repeat2 in range(min_cycles):
            imgr.init_next = imgr.out_last()
            imgr.make_image_I(show_updates=False)

    return imgr

imgr = converge(imgr)
result = imgr.out_last()

# Self calibrate to the previous model (phase-only)
observation_selfcal = eh.self_cal.self_cal(observation, result, method='phase')

# Make an image -- now with complex visibilities
imgr.obs_next = observation_selfcal
imgr.dat_term_next = {'vis':10, 'cphase':100, 'logcamp':100},
imgr = converge(imgr)
result = imgr.out_last()

# Self calibrate to the previous model (amplitude and phase)
observation_selfcal = eh.self_cal.self_cal(observation_selfcal, result, method='both')

# Make an image -- now primarily with complex visibilities
imgr.obs_next = observation_selfcal
imgr.dat_term_next = {'vis':100, 'cphase':10, 'logcamp':10},
imgr = converge(imgr)
final_result = imgr.out_last()

# Final image
im_out = imgr.out_last()
```

Code: https://github.com/achael/eht-imaging -- see examples folder!

Documentation: https://achael.github.io/eht-imaging/

Pros of Regularized Maximum Likelihood:

1. Forward modelling allows for flexibility in data terms and regularizers used. The framework allows for easy experimentation with new methods.

2. The fundamental image representation is continuous: resolution of structure at ½ to ¼ the beam size is possible

3. Easily scriptable: possible to run jobs exploring a huge range of image parameter space

Cons of Regularized Maximum Likelihood:

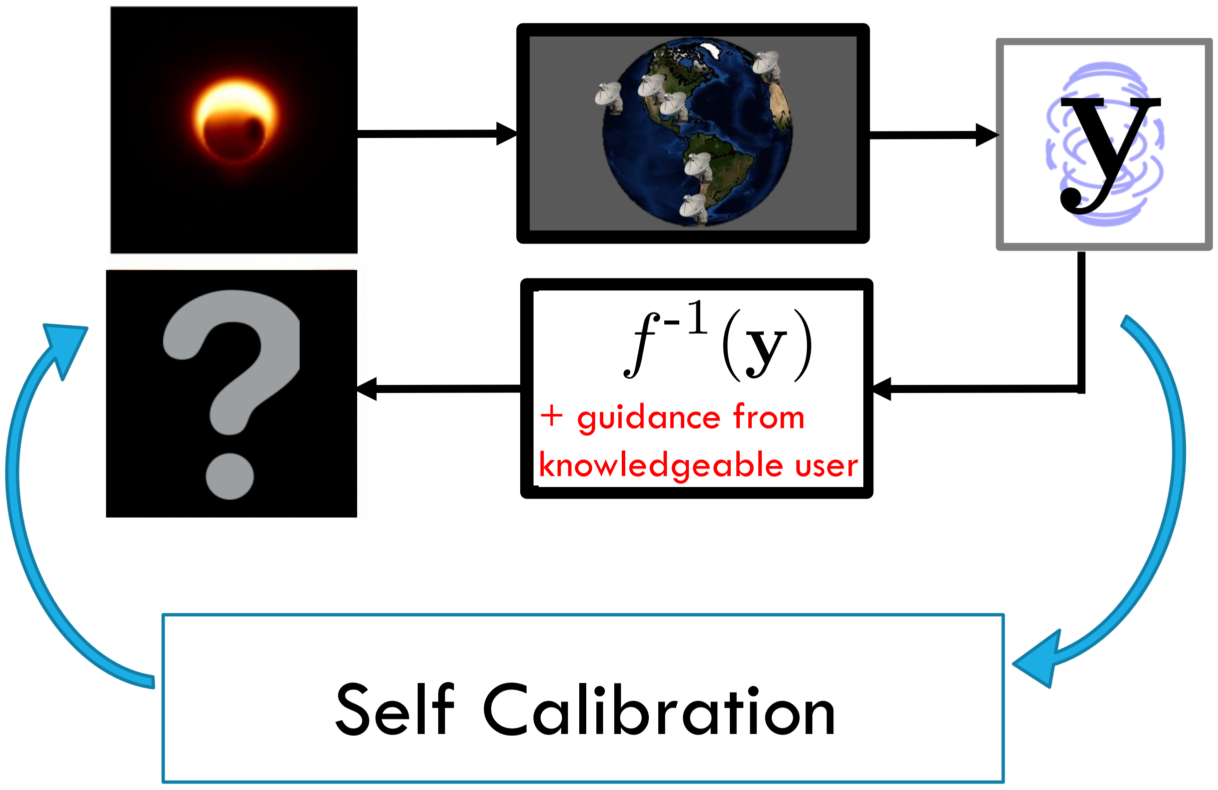
1. Convergence depends on having initial conditions well adapted to the source
   -- Easy for inexperienced imagers to fall into local minima with ghost images.

2. Slower: does not scale trivially to large datasets or images, especially when using closure quantities.

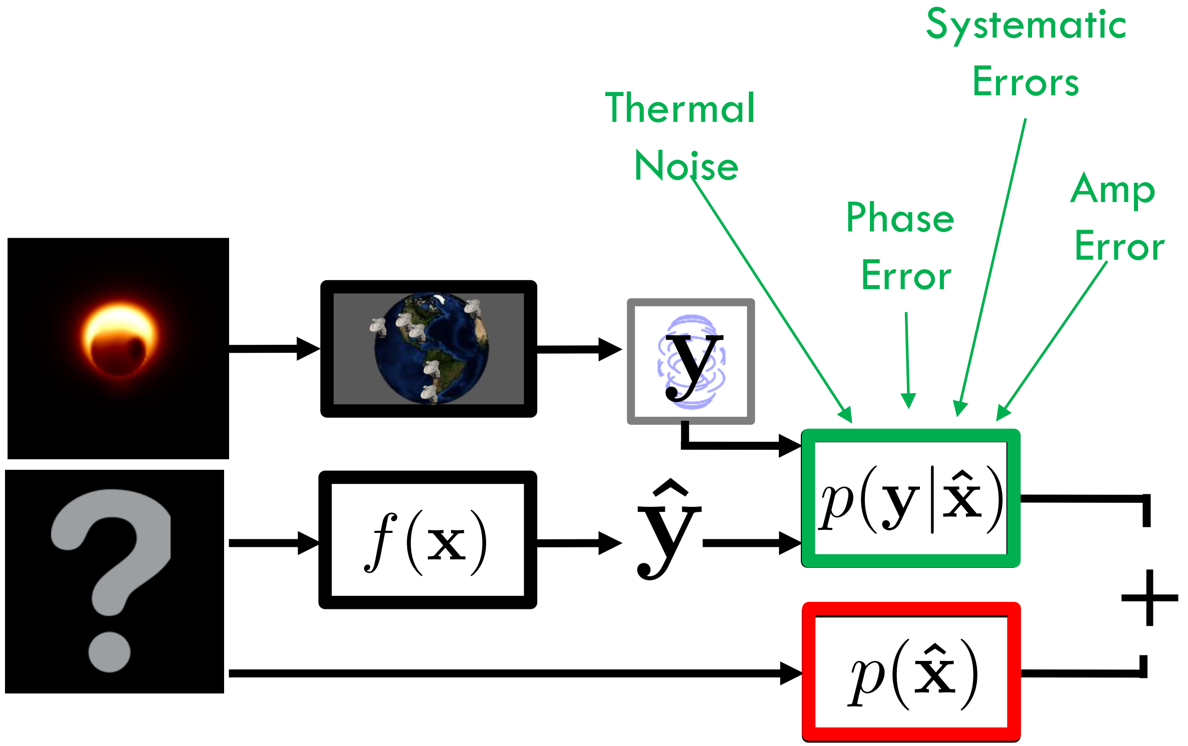
3. Non-Gaussian statistics and covariance among measurements are not yet implemented in our log – likelihoods (though they are coming!)

Validating an Image

Two Classes of Imaging Algorithms



Standard Inverse Modeling
(CLEAN + Self-Calibration)



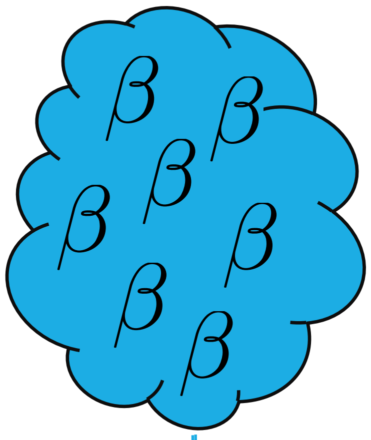
$$\hat{\mathbf{x}}_{\text{MAP}} = \operatorname{argmax}_{\mathbf{x}} \left[\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x})\right]$$

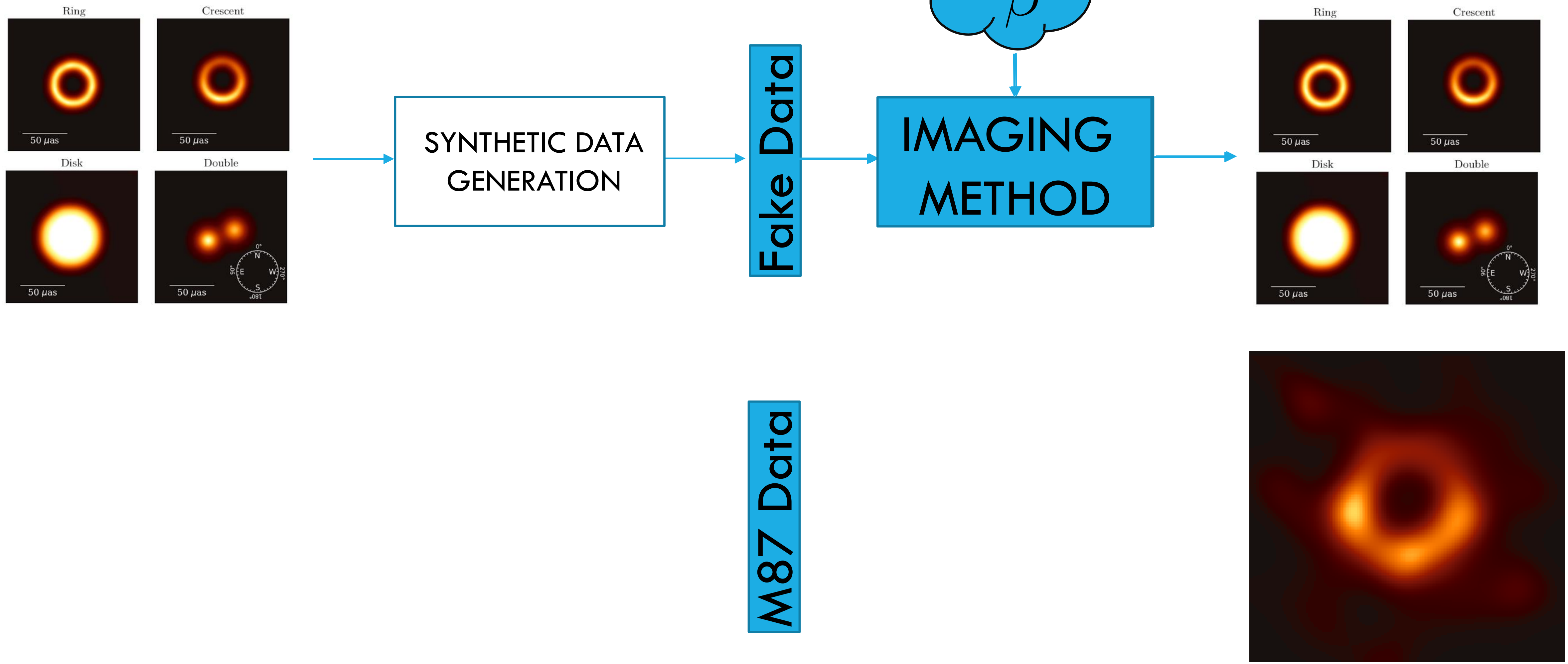
Forward Modeling
(Regularized Maximum Likelihood)

Imaging Parameter Surveys

DIFMAP (CLEAN + Self Calibration)	eht-imaging (Regularized Max Likelihood)	SMILI (Regularized Max Likelihood)
Compact Flux Stop Condition Weighting on ALMA Mask Size Data Weights	Compact Flux Initial Gaussian Size Systematic Error Regularizes MEM TV TSV L1	Compact Flux L1 Soft Mask Size Systematic Error Regularizes TV TSV L1

Testing thousands of parameter sets per method





Look for consistent features from different methods

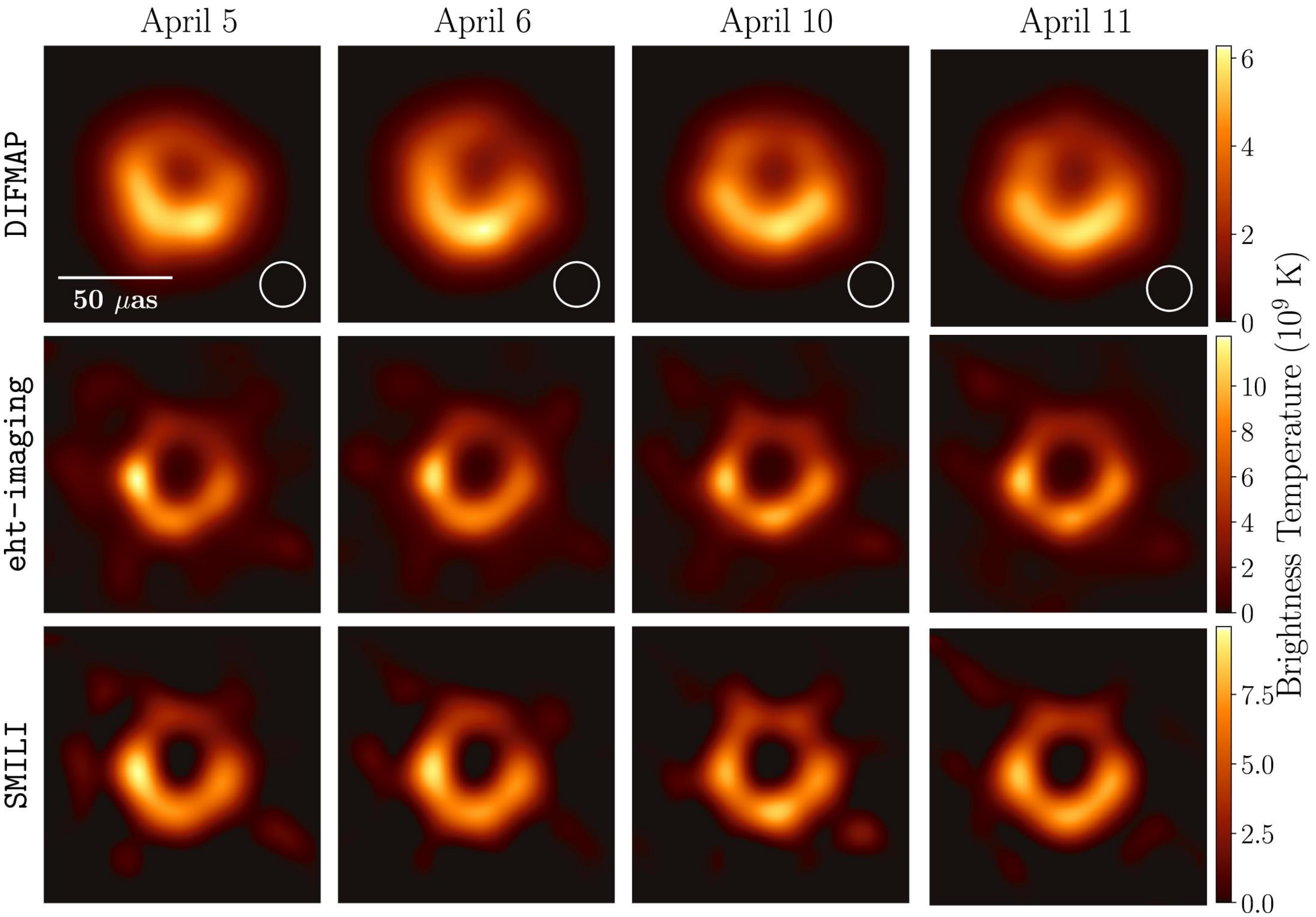
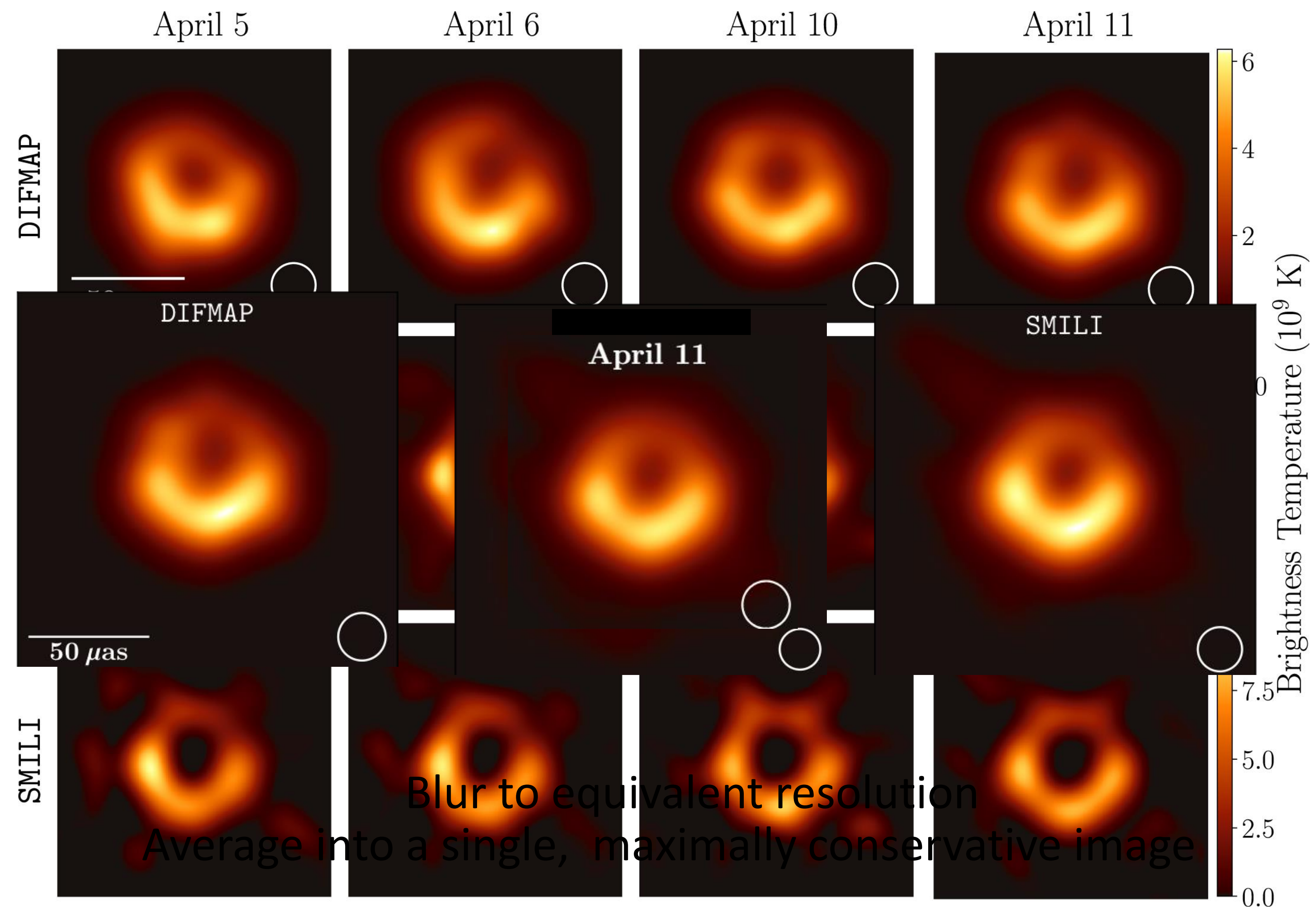


Image Credit: EHT Collaboration 2019 (Paper IV)

Look for consistent features from different methods

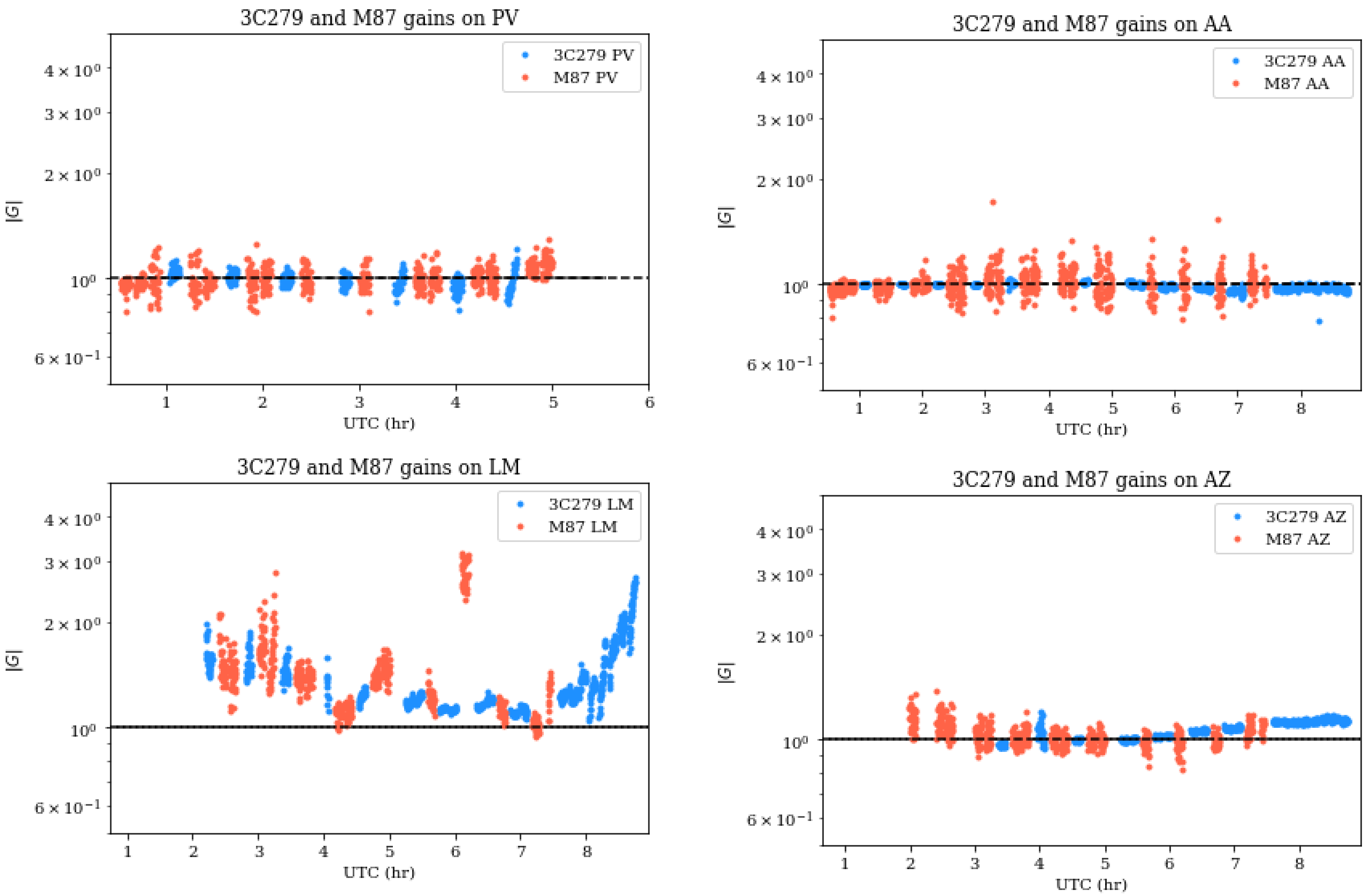


Blur to equivalent resolution

Average into a single, maximally conservative image

Image Credit: EHT Collaboration 2019 (Paper IV)

Validating with Calibrator Gains



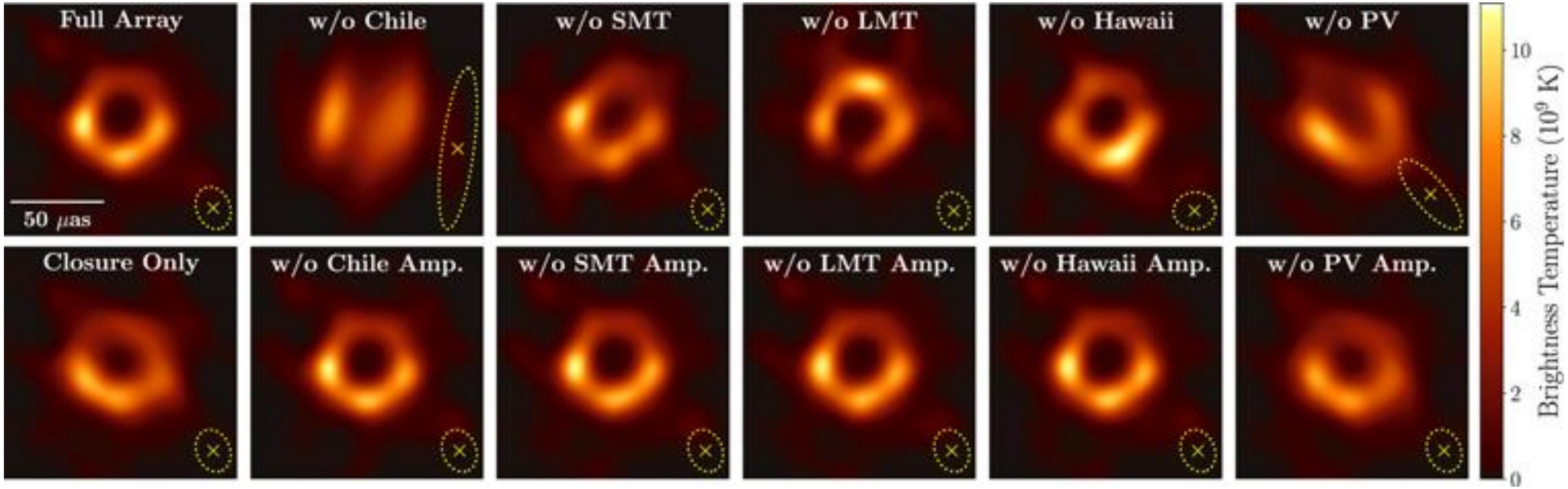
Gains from different sources at similar times should be consistent

(Inverse) Gains should not usually be < 1

- telescopes usually are less sensitive than estimates, not more

$$V_{\text{true}} = G_1 G_2 V_{\text{measured}}$$

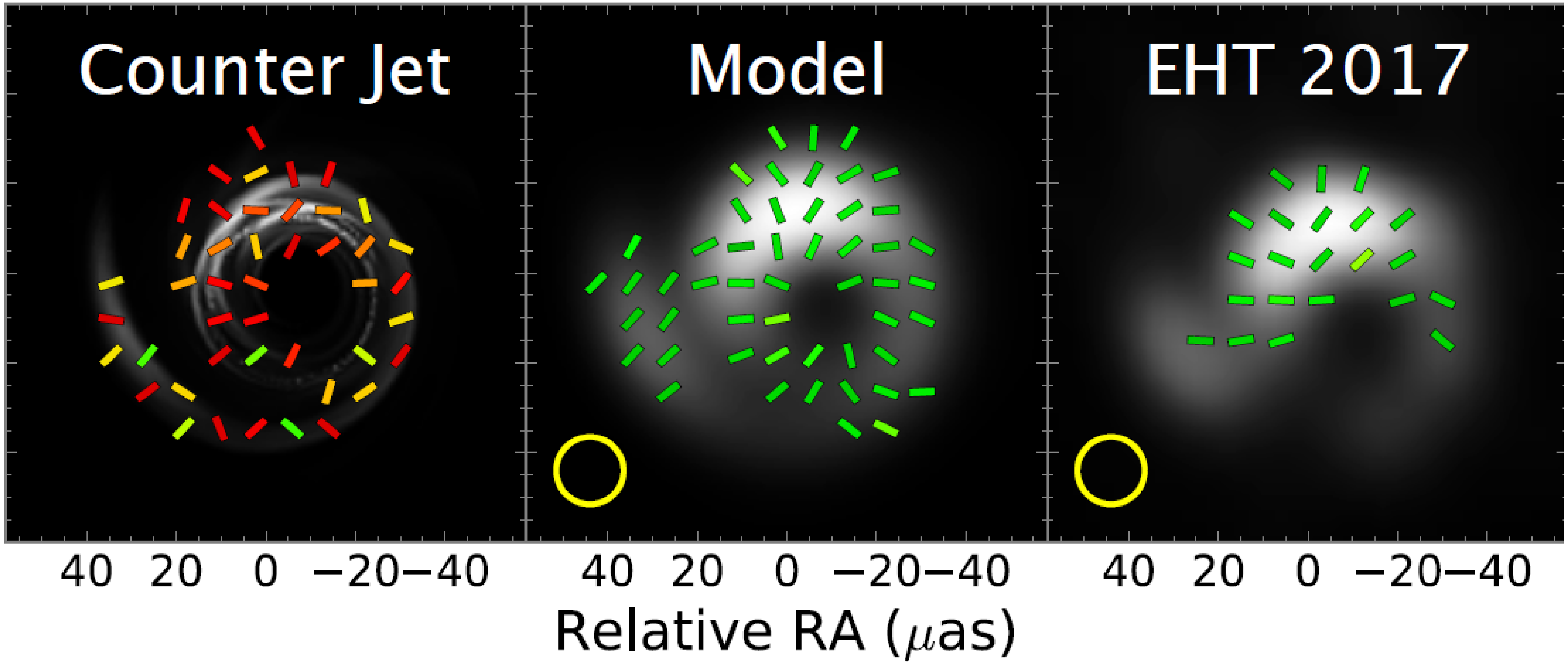
Validating by Omitting stations



Our images should not be too sensitive to the loss or miscalibration of any one telescope

Imaging Extensions

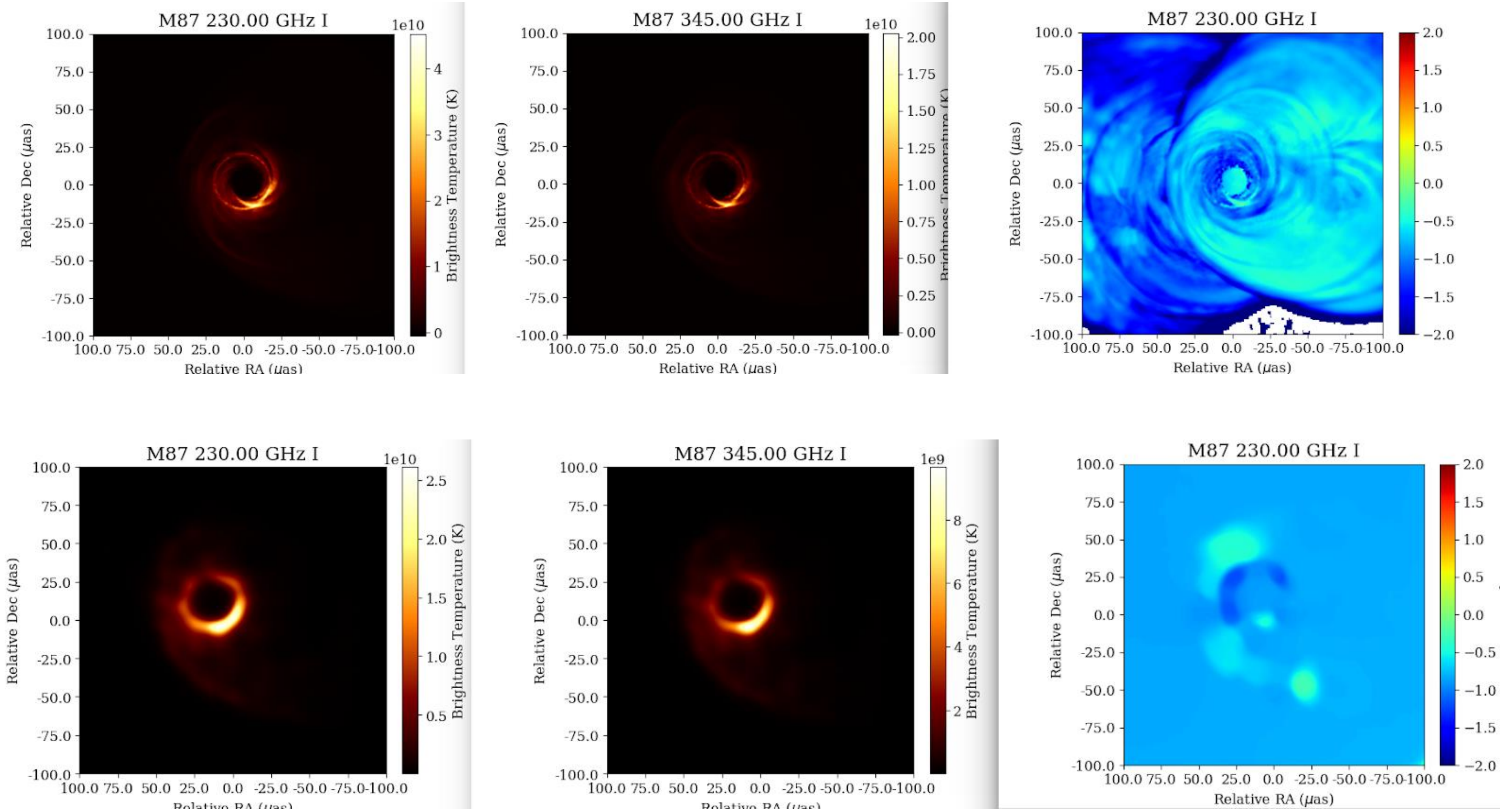
Extension 1: Polarization



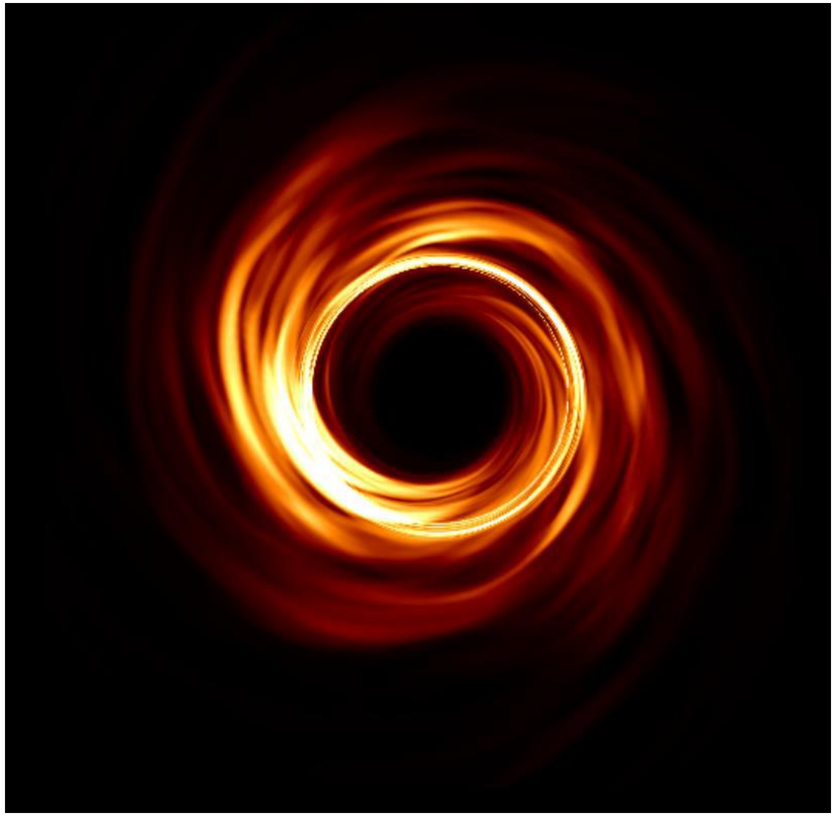
Simulation Credit: Jason Dexter & Monica Moscibrodzka (Bottom) Image Credit: Kazu Akiyama

Extension 2: Multi-frequency

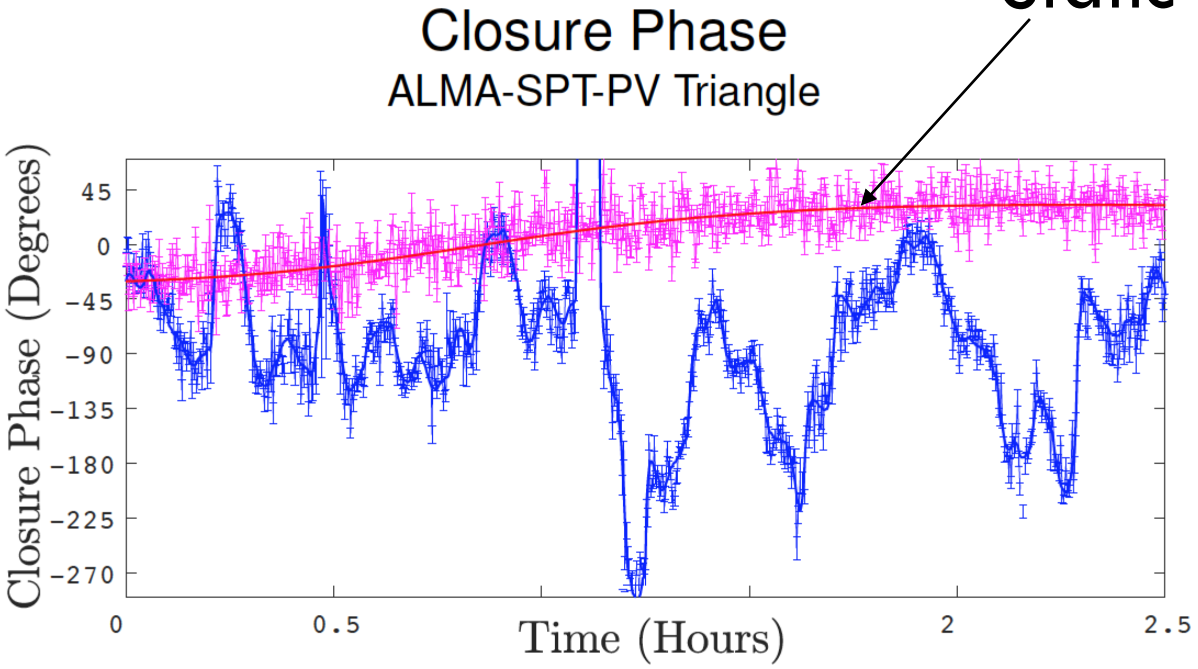
$$I_\nu(x, y) = I_0(x, y)\left(\frac{\nu}{\nu_0}\right)^{-\alpha(x,y)}$$



Extension 3: Dynamics



Static



Dynamic

Summary

- VLBI data is incompletely sampled – imaging algorithms are required to infer a best-guess image from the observed data

- Two important classes of imaging algorithms are:
  - CLEAN – fast, iterative, models image as point source
  - RML – works on closure quantities, flexible

- Imaging is path-dependent and requires careful validation

- Many open areas to explore in designing imaging techniques for EHT and other VLBI arrays!

Next Steps

Fill out the webinar survey at
http://bit.ly/BHPIRE-Imaging

Get started with eht-imaging at
https://github.com/achael/eht-imaging

Play with real M87 data and EHTC imaging scripts at
https://github.com/eventhorizontelescope/2019-D01-02

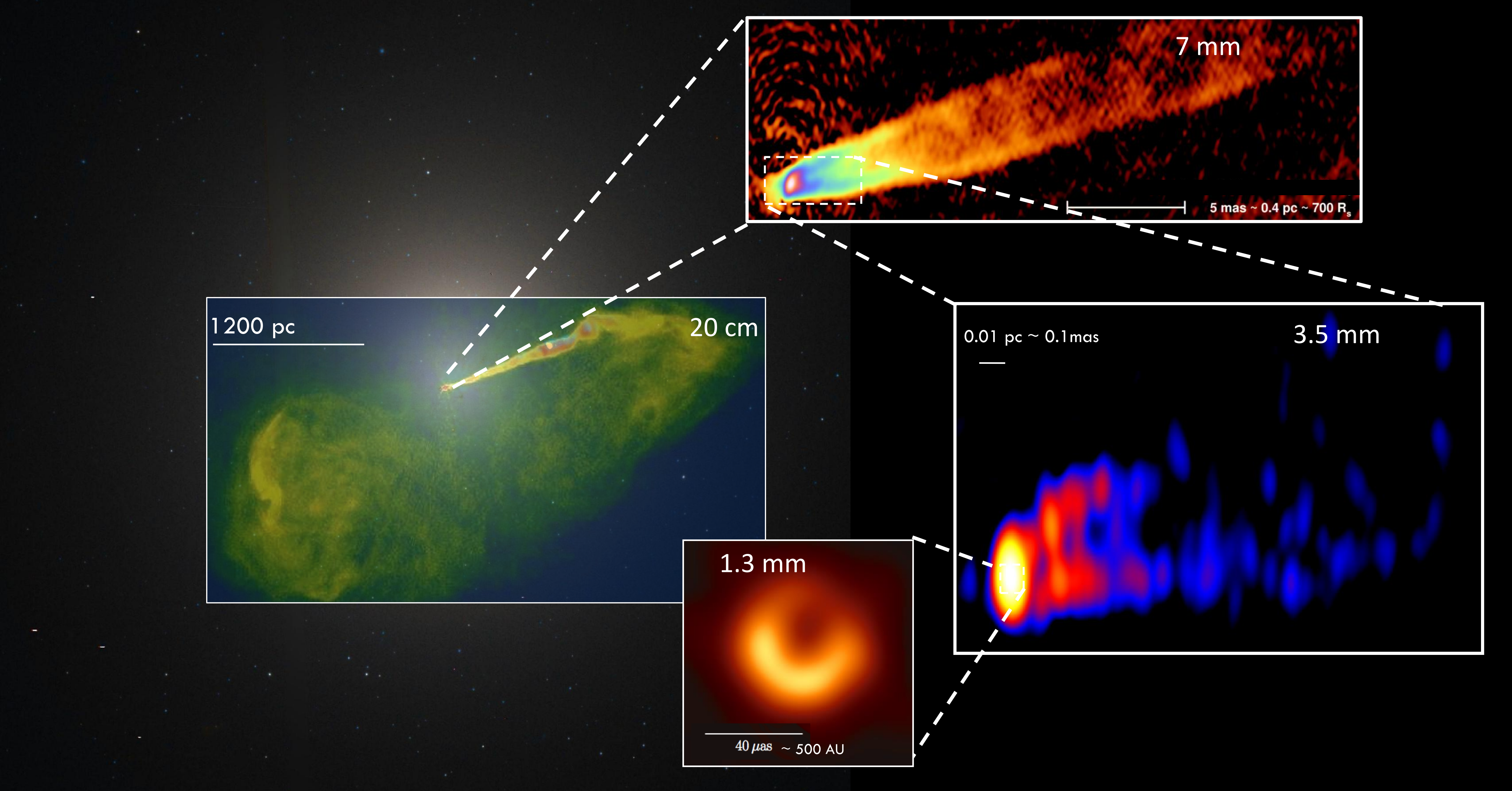


Image Credits: NRAO (VLA),
Craig Walker (7mm VLBA), Kazuhiro Hada (VLBA+GBT 3mm),
EHT (1.3 mm)