resolve

Mathematical Concepts

Jakob Roth, MPA Garching

\+ Andreas Popp, ESO Garching

30.01.2026

Bayesian Imaging

Bayes' theorem

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

- Data: d (e.g. visibilities)
- Signal: s (e.g. sky brightness)

Bayesian Imaging

Bayes' theorem

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

- Data: d (e.g. visibilities)
- Signal: s (e.g. sky brightness)

- How to choose the prior $\mathcal{P}(s)$?

Bayesian Imaging

Bayes' theorem

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

- Data: d (e.g. visibilities)
- Signal: s (e.g. sky brightness)

- How to choose the prior $\mathcal{P}(s)$?
- How to construct the likelihood $\mathcal{P}(d|s)$?

Bayesian Imaging

Bayes' theorem

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

- Data: d (e.g. visibilities)
- Signal: s (e.g. sky brightness)

- How to choose the prior $\mathcal{P}(s)$?
- How to construct the likelihood $\mathcal{P}(d|s)$?
- How to compute the posterior $\mathcal{P}(s|d)$?

Bayesian Imaging – Software

Bayesian Imaging – Software

NIFTy

- `https://gitlab.mpcdf.mpg.de/ift/nifty`
- Prior Models
- Inference Algorithms

Bayesian Imaging – Software

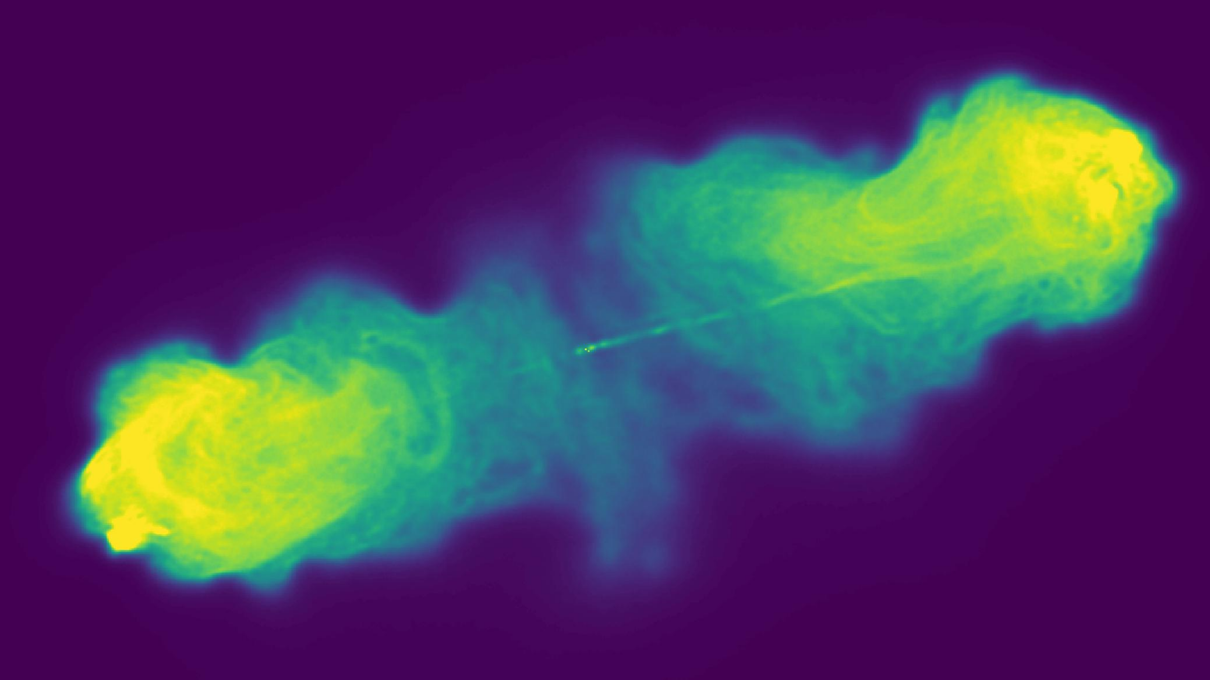
NIFTy

- `https://gitlab.mpcdf.mpg.de/ift/nifty`
- Prior Models
- Inference Algorithms

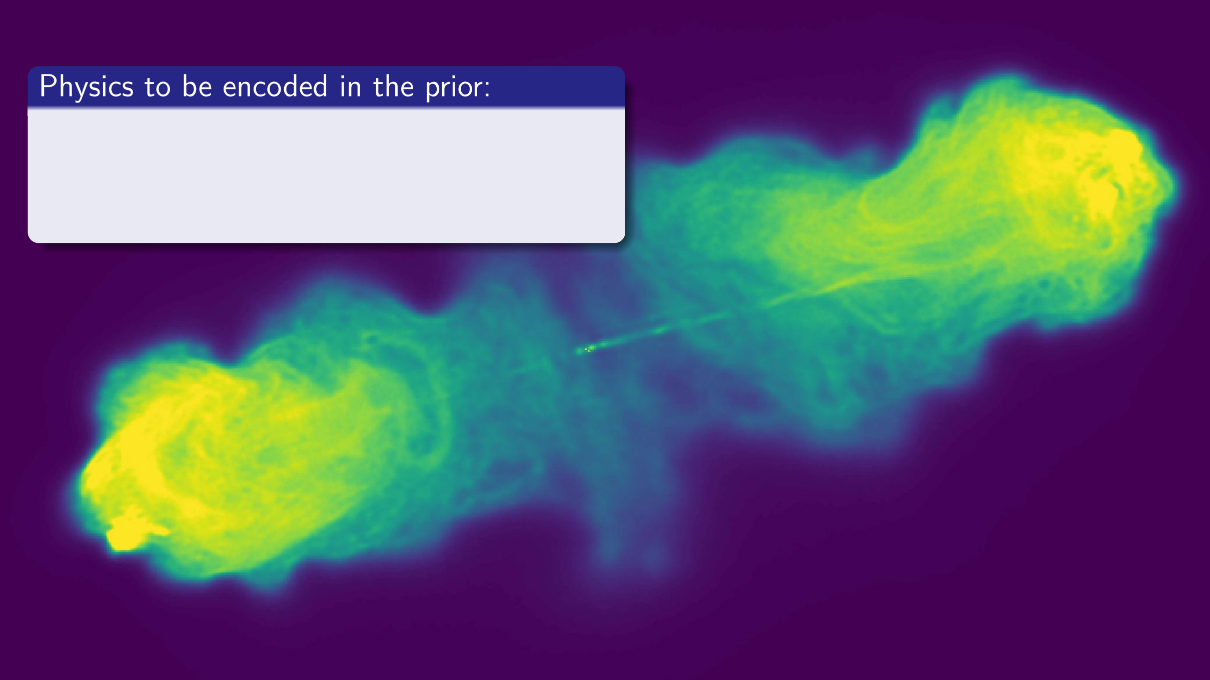
`resolve`

- `https://gitlab.mpcdf.mpg.de/ift/resolve`
- Handling radio interferometric data
- Measurement equation

How to choose the prior $\mathcal{P}(s)$?

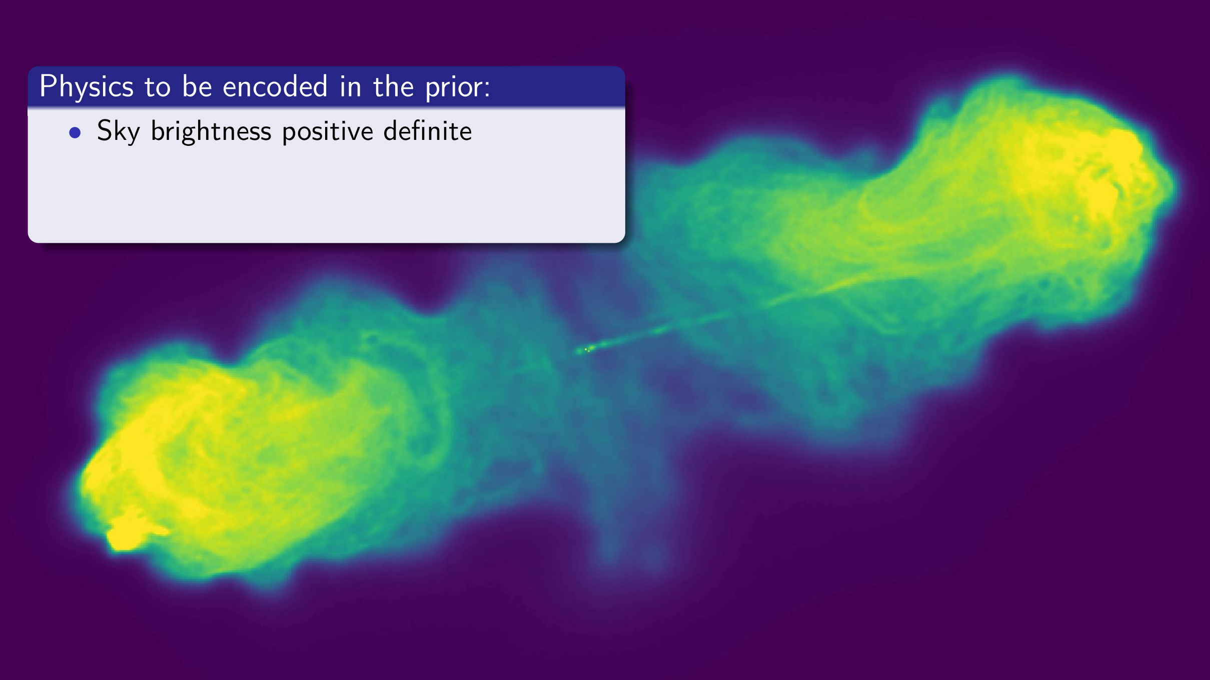


Physics to be encoded in the prior:



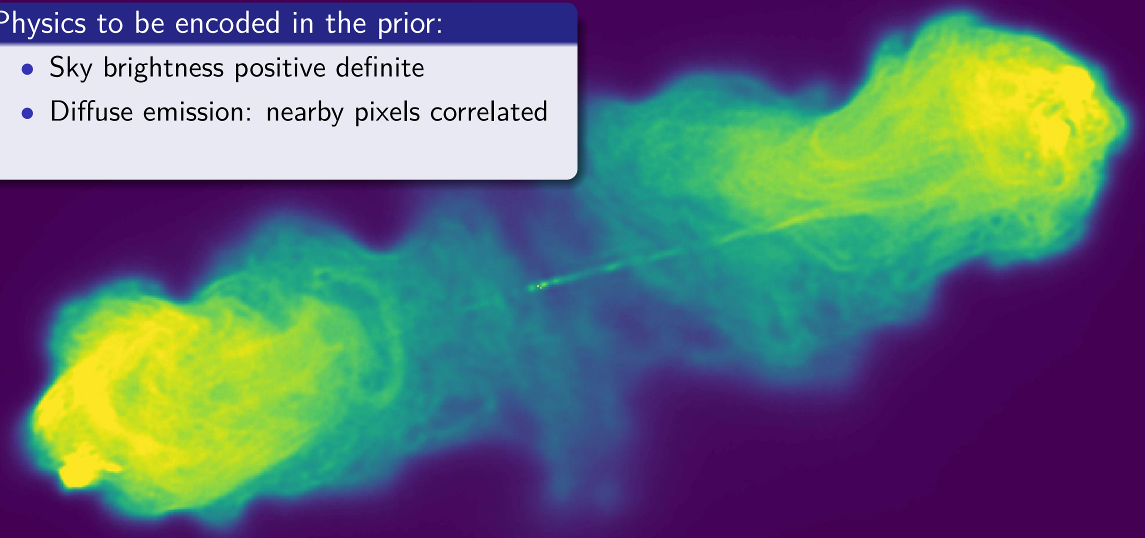
Physics to be encoded in the prior:

- Sky brightness positive definite



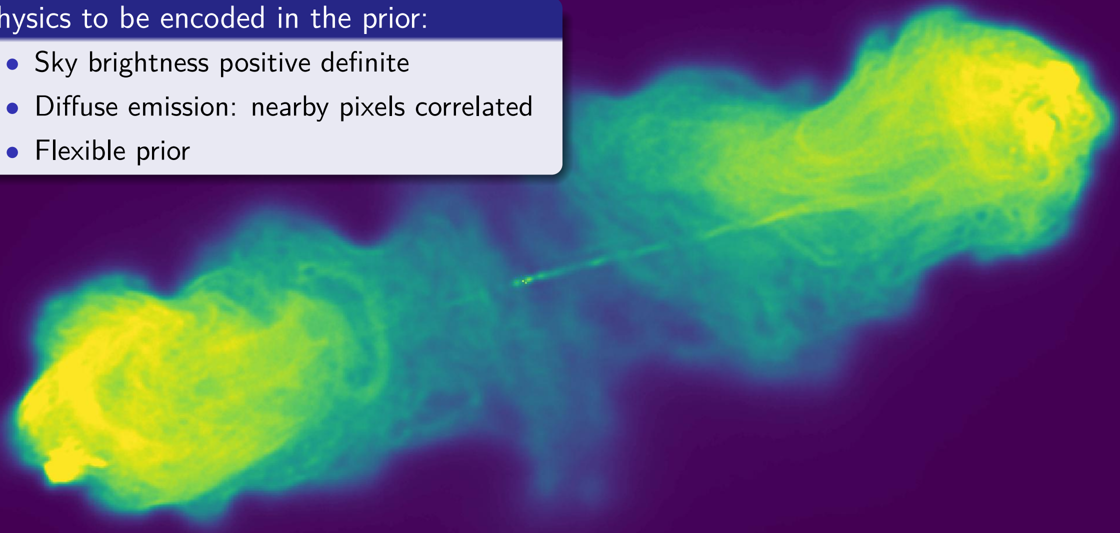
Physics to be encoded in the prior:

- Sky brightness positive definite
- Diffuse emission: nearby pixels correlated



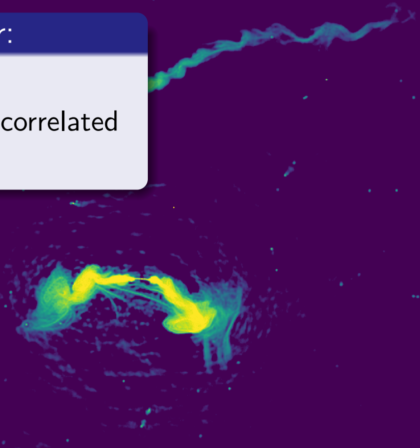
Physics to be encoded in the prior:

- Sky brightness positive definite
- Diffuse emission: nearby pixels correlated
- Flexible prior



Physics to be encoded in the prior:

- Sky brightness positive definite
- Diffuse emission: nearby pixels correlated
- Flexible prior



Physics to be encoded in the prior:

- Sky brightness positive definite
- Diffuse emission: nearby pixels correlated
- Flexible prior

Physics to be encoded in the prior:

- Sky brightness positive definite
- Diffuse emission: nearby pixels correlated
- Flexible prior

Complex prior distribution

$\rightarrow$ Prior as generative model

Prior – Generative Models

Standardized Generative Model

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0, 1)$

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0, 1)$
- $\xi \mapsto s$

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0, 1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\mathrm{CDF}^{-1}_{P(s)} \circ \mathrm{CDF}_{\mathcal{G}}\right)(\xi)$

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0,1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\mathrm{CDF}^{-1}_{P(s)} \circ \mathrm{CDF}_{\mathcal{G}}\right)(\xi)$

Example: Log Normal

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0, 1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\text{CDF}^{-1}_{P(s)} \circ \text{CDF}_{\mathcal{G}}\right)(\xi)$

Example: Log Normal

- $\xi \sim \mathcal{G}(0, 1)$

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0, 1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\mathrm{CDF}^{-1}_{P(s)} \circ \mathrm{CDF}_{\mathcal{G}}\right)(\xi)$

Example: Log Normal

- $\xi \sim \mathcal{G}(0, 1)$
- $s(\xi) = e^{\xi}$

Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0, 1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\mathrm{CDF}^{-1}_{P(s)} \circ \mathrm{CDF}_{\mathcal{G}}\right)(\xi)$

Example: Log Normal

- $\xi \sim \mathcal{G}(0, 1)$
- $s(\xi) = e^{\xi}$
- $s \sim \mathrm{LogNorm}(0, 1)$

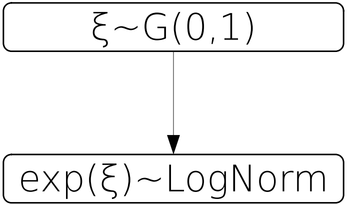
Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0,1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\mathrm{CDF}^{-1}_{P(s)} \circ \mathrm{CDF}_{\mathcal{G}}\right)(\xi)$

Example: Log Normal

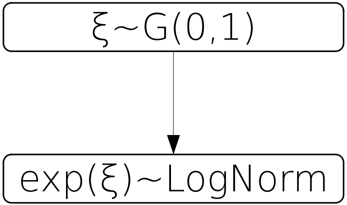
- $\xi \sim \mathcal{G}(0,1)$
- $s(\xi) = e^{\xi}$
- $s \sim \mathrm{LogNorm}(0,1)$



Prior – Generative Models

Standardized Generative Model

- Latent parameters ξ
- $\mathcal{P}(\xi) = \mathcal{G}(0,1)$
- $\xi \mapsto s$
- In general: $s(\xi) = \left(\text{CDF}^{-1}_{P(s)} \circ \text{CDF}_{\mathcal{G}}\right)(\xi)$



Example: Log Normal

- $\xi \sim \mathcal{G}(0,1)$
- $s(\xi) = e^{\xi}$
- $s \sim \text{LogNorm}(0,1)$

$\surd$ Positivity

Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

- $\boldsymbol{\xi} \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$

Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

- $\boldsymbol{\xi} \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = A\boldsymbol{\xi}$ with $A = \sqrt{C}$

Prior – Generative Models

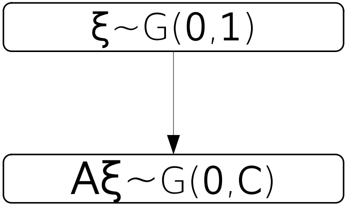
Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

- $\boldsymbol{\xi} \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = A\boldsymbol{\xi}$ with $A = \sqrt{C}$
- $s \sim \mathcal{G}(\mathbf{0}, C)$

Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

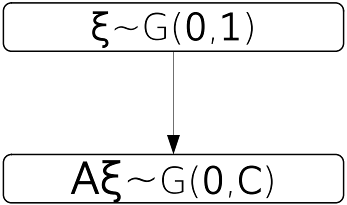
- $\boldsymbol{\xi} \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = A\boldsymbol{\xi}$ with $A = \sqrt{C}$
- $s \sim \mathcal{G}(\mathbf{0}, C)$

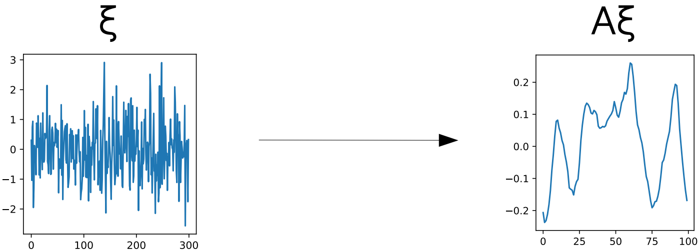


Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

- $\xi \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = A\xi$ with $A = \sqrt{C}$
- $s \sim \mathcal{G}(\mathbf{0}, C)$

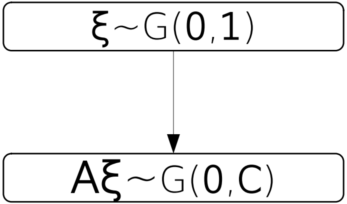


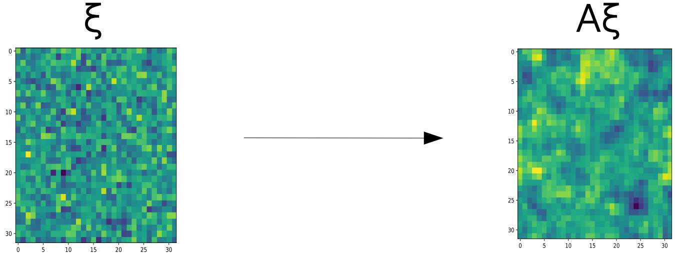


Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

- $\xi \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = A\xi$ with $A = \sqrt{C}$
- $s \sim \mathcal{G}(\mathbf{0}, C)$

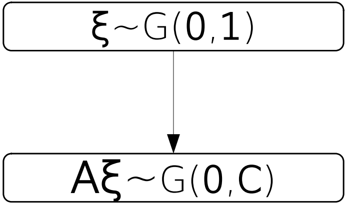


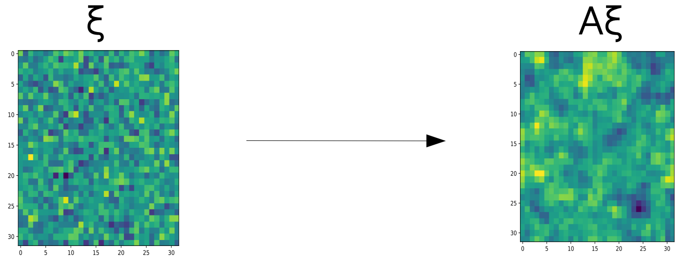


Prior – Generative Models

Example: Correlated Gaussian $\mathcal{G}(\mathbf{0}, C)$

- $\xi \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = A\xi$ with $A = \sqrt{C}$
- $s \sim \mathcal{G}(\mathbf{0}, C)$



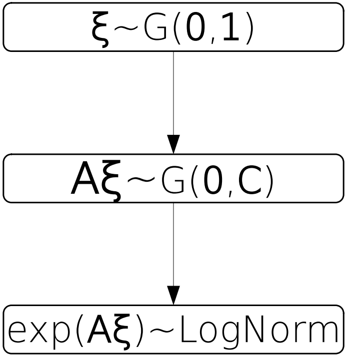


√ Correlation

Prior – Generative Models

Example: Correlated Log Normal

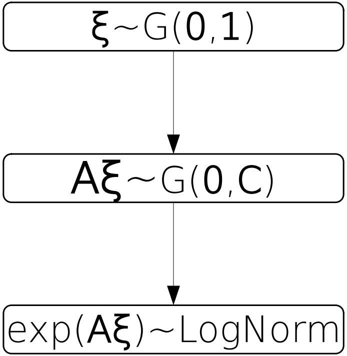
- $\boldsymbol{\xi} \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = \exp(A\boldsymbol{\xi})$ with $A = \sqrt{C}$
- $s \sim \mathrm{LogNorm}(\mathbf{0}, C)$

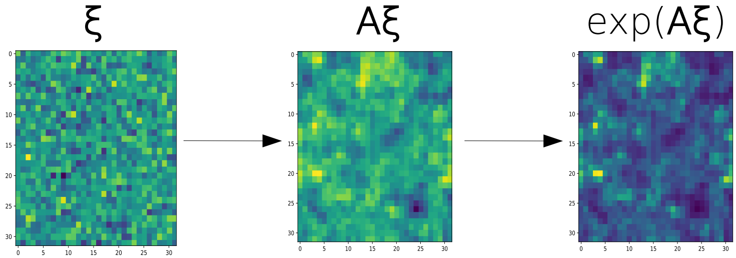


Prior – Generative Models

Example: Correlated Log Normal

- $\xi \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = \exp(A\xi)$ with $A = \sqrt{C}$
- $s \sim \mathrm{LogNorm}(\mathbf{0}, C)$

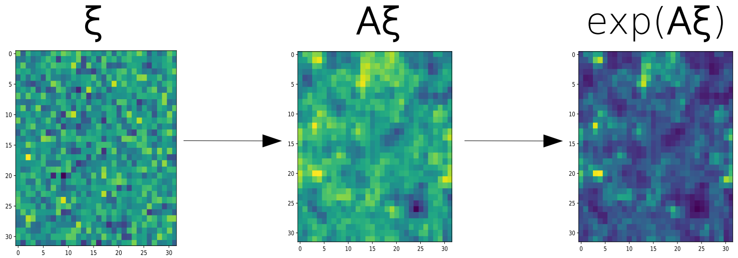


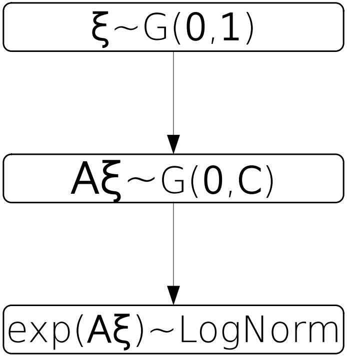


Prior – Generative Models

Example: Correlated Log Normal

- $\xi \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = \exp(A\xi)$ with $A = \sqrt{C}$
- $s \sim \mathrm{LogNorm}(\mathbf{0}, C)$



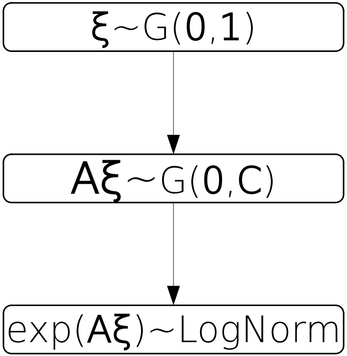


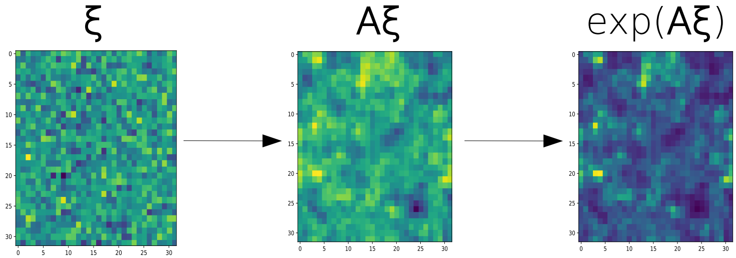
√ Positivity

Prior – Generative Models

Example: Correlated Log Normal

- $\xi \sim \mathcal{G}(\mathbf{0}, \mathbb{I})$
- $s(\xi) = \exp(A\xi)$ with $A = \sqrt{C}$
- $s \sim \mathrm{LogNorm}(\mathbf{0}, C)$

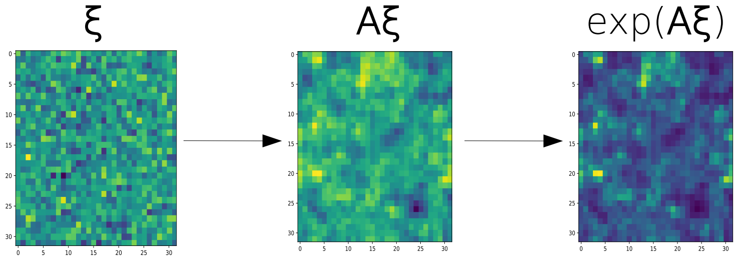


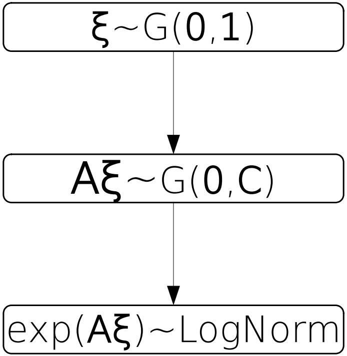


√ Positivity
√ Correlation

Prior – Generative Models

statistical homogeneity and isotropy

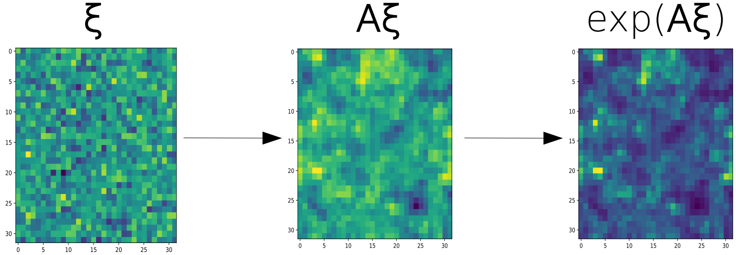


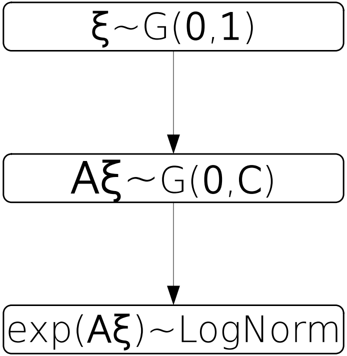


Prior – Generative Models

statistical homogeneity and isotropy

- For statistical homogeneity and isotropy: A diagonal in Fourier space.

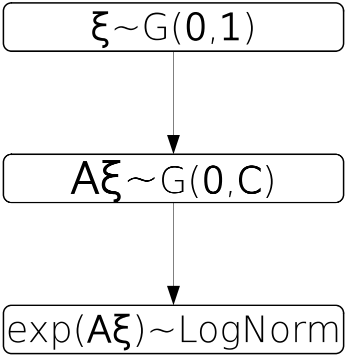


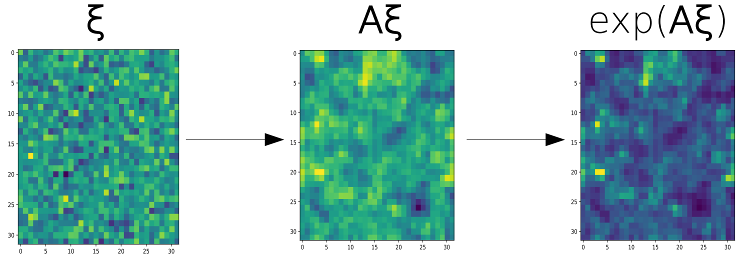


Prior – Generative Models

statistical homogeneity and isotropy

- For statistical homogeneity and isotropy: A diagonal in Fourier space.
- A fully described by power spectrum

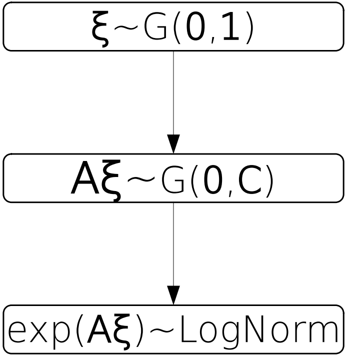


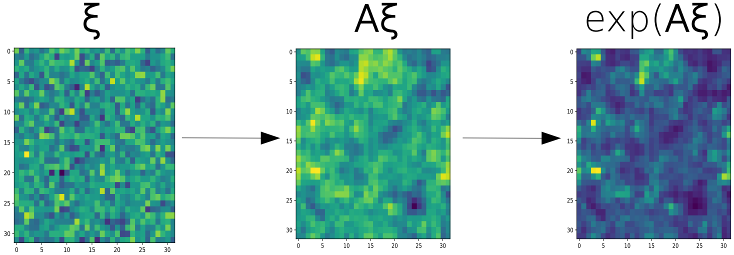


Prior – Generative Models

statistical homogeneity and isotropy

- For statistical homogeneity and isotropy: A diagonal in Fourier space.
- A fully described by power spectrum
- Generative model in in Fourier space: $s(\xi) = \mathcal{F}\tilde{A}\xi$ with $\tilde{A}$ diagonal

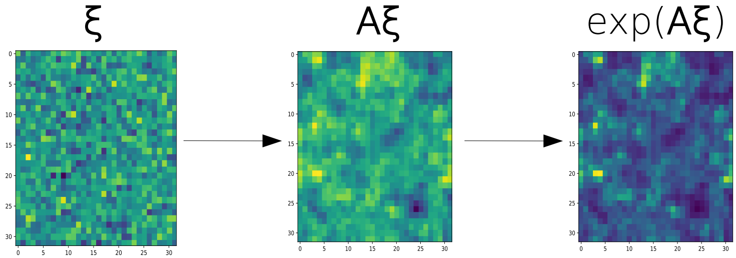


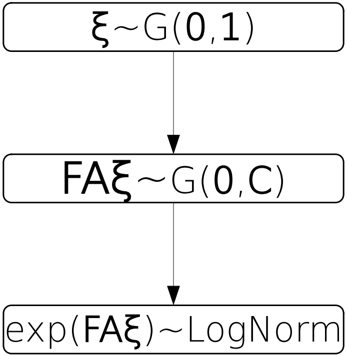


Prior – Generative Models

statistical homogeneity and isotropy

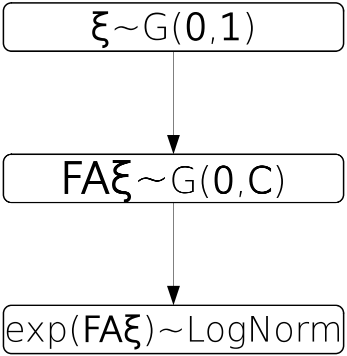
- For statistical homogeneity and isotropy: A diagonal in Fourier space.
- A fully described by power spectrum
- Generative model in in Fourier space: $s(\xi) = \mathcal{F}\tilde{A}\xi$ with $\tilde{A}$ diagonal



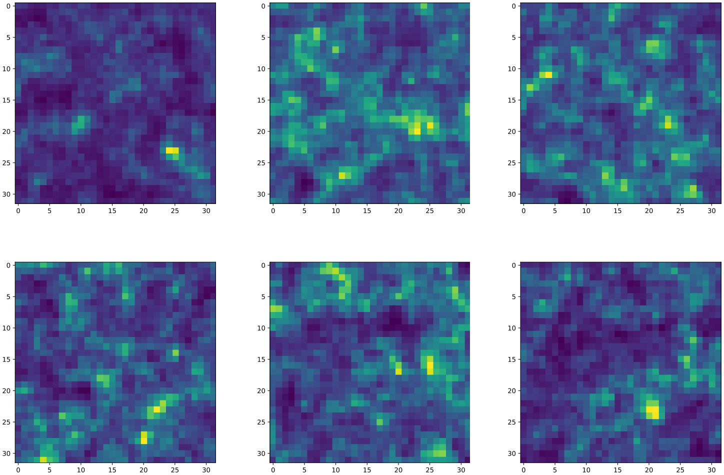


Prior – Generative Models

```python
import numpy as np
import nifty8 as ift

def A_spec(k):
  return 1./(20. + k**2)

...

HT = ift.HarmonicTransformOperator(...)
sky = ift.exp(HT(ift.DiagonalOperator(A)))

xi = ift.from_random(sky.domain)
sample = sky(xi)
```



Prior – Generative Models



$$\xi \sim \mathcal{G}(0,1)$$

↓

$$FA\xi \sim \mathcal{G}(0,C)$$

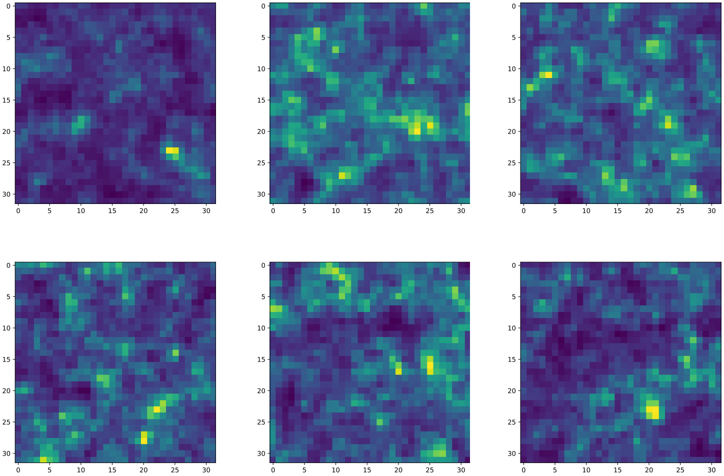
↓

$$\exp(FA\xi) \sim \text{LogNorm}$$

√ Positivity

√ Correlation

Prior – Generative Models



$\xi \sim \mathcal{G}(0,1)$

↓

$FA\xi \sim \mathcal{G}(0,C)$

↓

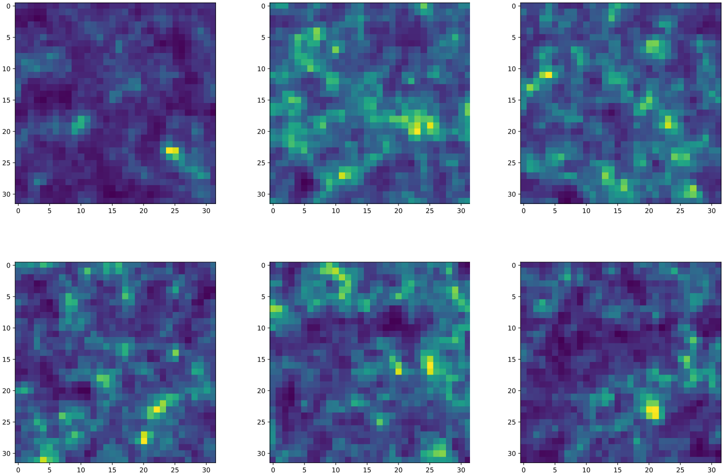
$\exp(FA\xi) \sim \text{LogNorm}$

√ Positivity

√ Correlation

× Flexible

Prior – Generative Models



$\xi \sim G(0,1)$

↓

$FA\xi \sim G(0,C)$

↓

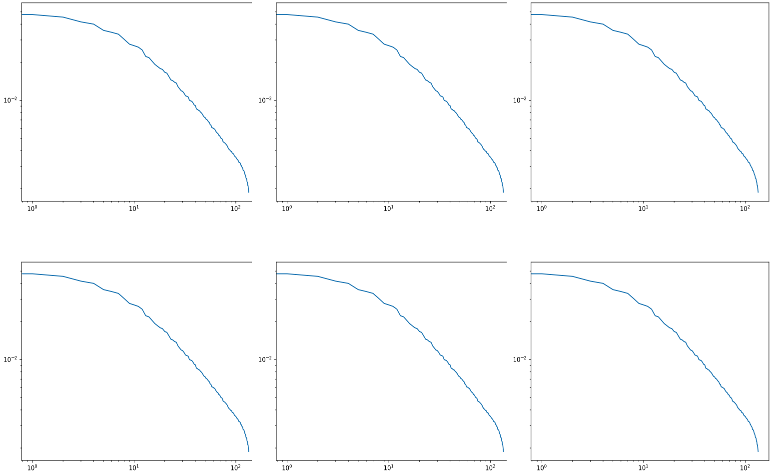
$\exp(FA\xi) \sim \text{LogNorm}$

√ Positivity

√ Correlation

× Flexible

Prior – Generative Models



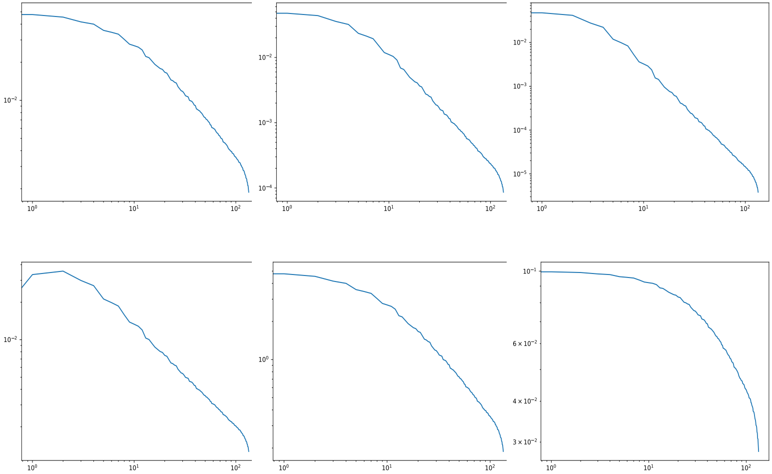
$\xi \sim G(0,1)$

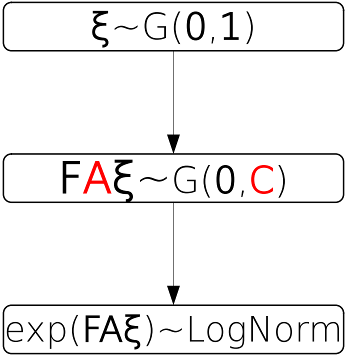
$FA\xi \sim G(0,C)$

$\exp(FA\xi) \sim \mathrm{LogNorm}$

- √ Positivity
- √ Correlation
- × Flexible

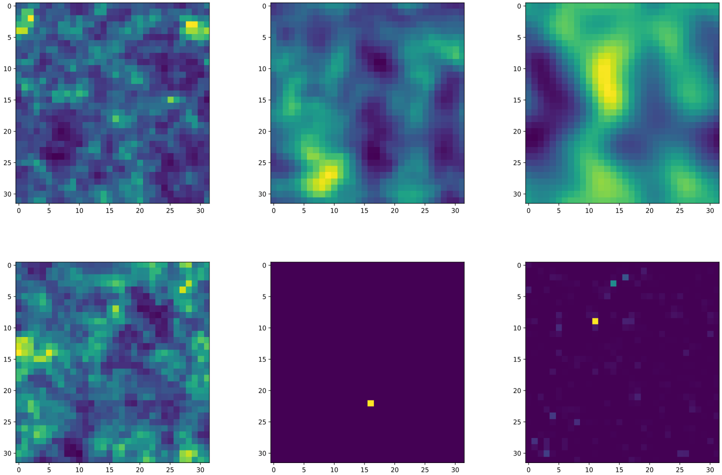
Prior – Generative Models

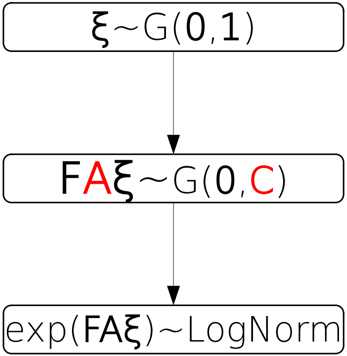




√ Positivity
√ Correlation
× Flexible

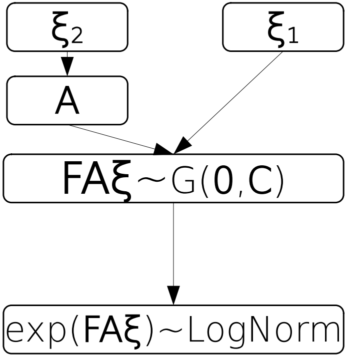
Prior – Generative Models



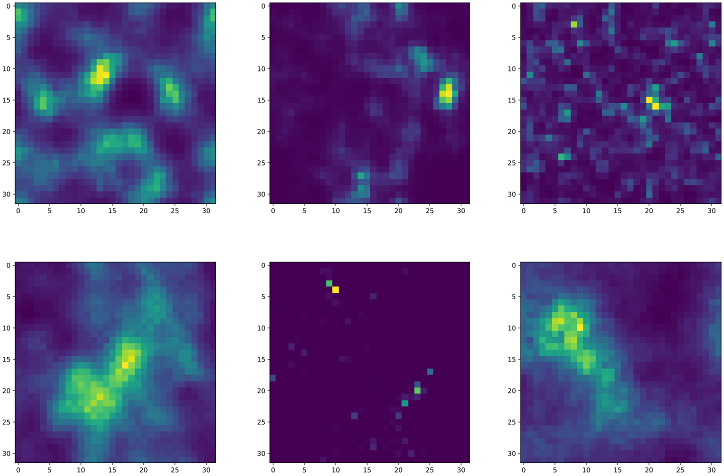


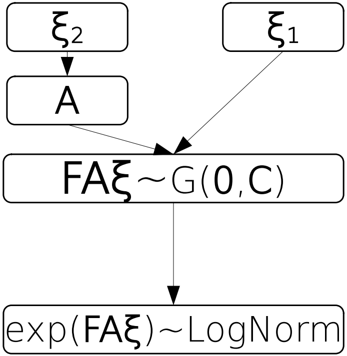
- √ Positivity
- √ Correlation
- × Flexible

Prior – Generative Models



Prior – Generative Models

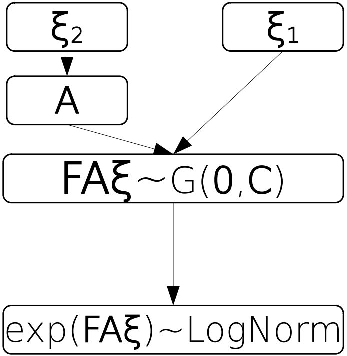




- √ Positivity
- √ Correlation
- √ Flexible

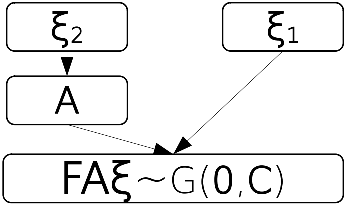
Prior – Generative Models

```
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])
cfm = ift.SimpleCorrelatedField(x_space, ...)

sky = ift.exp(cfm)

xi = ift.from_random(sky.domain)
sample = sky(xi)
```



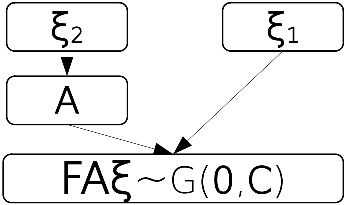
Prior – Generative Models

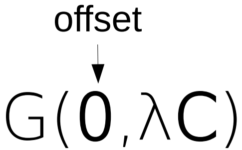
```python
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])

args = {
'offset_mean': 0.,
'offset_std': (3., 1.),

'fluctuations': (1.5, 0.5),

'loglogavgslope': (-3., .5),
'flexibility': None,
'asperity': None
}

cfm = ift.SimpleCorrelatedField(x_space, **args)
```



Prior – Generative Models

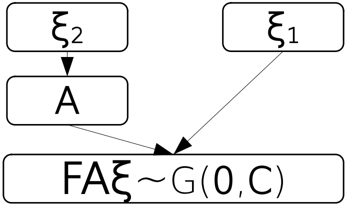
```python
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])

args = {
'offset_mean': 0.,
'offset_std': (3., 1.),

'fluctuations': (1.5, 0.5),

'loglogavgslope': (-3., .5),
'flexibility': None,
'asperity': None
}

cfm = ift.SimpleCorrelatedField(x_space, **
    args)
```

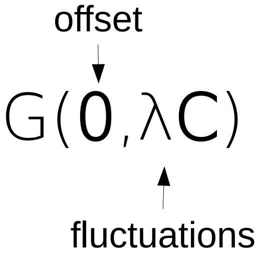




Prior – Generative Models

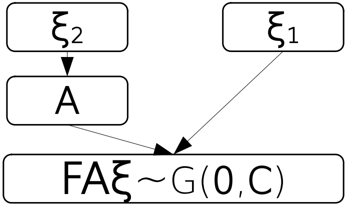
```python
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])

args = {
'offset_mean': 0.,
'offset_std': (3., 1.),

'fluctuations': (1.5, 0.5),

'loglogavgslope': (-3., .5),
'flexibility': None,
'asperity': None
}

cfm = ift.SimpleCorrelatedField(x_space, **
    args)
```

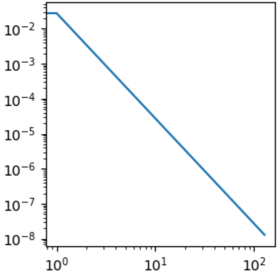




Prior – Generative Models

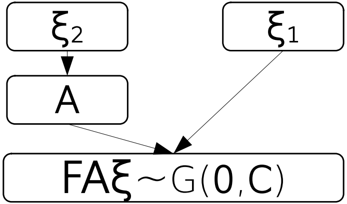
```
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])

args = {
'offset_mean': 0.,
'offset_std': (3., 1.),

'fluctuations': (1.5, 0.5),

'loglogavgslope': (-3., .5),
'flexibility': None,
'asperity': None
}

cfm = ift.SimpleCorrelatedField(x_space, **
    args)
```

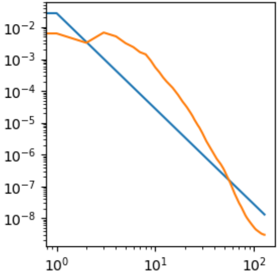




Prior – Generative Models

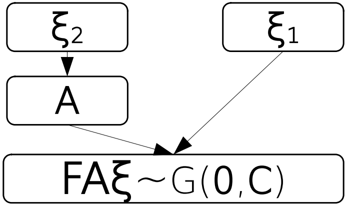
```python
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])

args = {
'offset_mean': 0.,
'offset_std': (3., 1.),

'fluctuations': (1.5, 0.5),

'loglogavgslope': (-3., .5),
'flexibility': (1., 0.5),
'asperity': None
}

cfm = ift.SimpleCorrelatedField(x_space, **
    args)
```

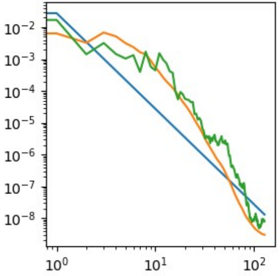




Prior – Generative Models

```
import numpy as np
import nifty8 as ift

x_space = ift.RGSpace([100, 100])

args = {
'offset_mean': 0.,
'offset_std': (3., 1.),

'fluctuations': (1.5, 0.5),

'loglogavgslope': (-3., .5),
'flexibility': (1., 0.5),
'asperity': (1., 0.5)
}

cfm = ift.SimpleCorrelatedField(x_space, **
    args)
```





$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

$$\Downarrow$$

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

$$\Downarrow$$

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}$$

$$\Downarrow$$

$$\mathcal{P}(\xi|d) = \frac{\color{red}{\mathcal{P}(d|s(\xi))}\color{teal}{\mathcal{P}(\xi)}}{\mathcal{P}(d)}$$

How to construct the likelihood $\mathcal{P}(d|s)$?

How to construct the likelihood $\mathcal{P}(d|s)$?

Response

$$\widetilde{V} = \int \frac{dldm}{n(l,m)} G_p(l,m,t) I(l,m) G_q^*(l,m,t) e^{-ik\cdot x}$$

- Sky brightness: I
- Antenna gain: G
- Model visibilities: $\widetilde{V}$

How to construct the likelihood $\mathcal{P}(d|s)$?

Response

$$\widetilde{V} = \int \frac{dldm}{n(l,m)} G_p(l,m,t) I(l,m) G_q^*(l,m,t) e^{-ik\cdot x}$$

- Sky brightness: I
- Antenna gain: G
- Model visibilities: $\widetilde{V}$

$$\widetilde{V} = RI$$

How to construct the likelihood $\mathcal{P}(d|s)$?

Measurement equation:

$$V = RI + n$$

- Visibilities V
- Response R
- Sky brightness I
- Noise n

How to construct the likelihood $\mathcal{P}(d|s)$?

Measurement equation:

$$V = RI + n$$

- Visibilities V
- Response R
- Sky brightness I
- Noise n
- Gaussian distributed noise $n \sim \mathcal{G}(0, N)$

How to construct the likelihood $\mathcal{P}(d|s)$?

Measurement equation:

$$V = RI + n$$

- Visibilities V
- Response R
- Sky brightness I
- Noise n
- Gaussian distributed noise $n \sim \mathcal{G}(0, N)$
- Likelihood:

$$\mathcal{P}(V|I) = \mathcal{G}(V - RI, N)$$

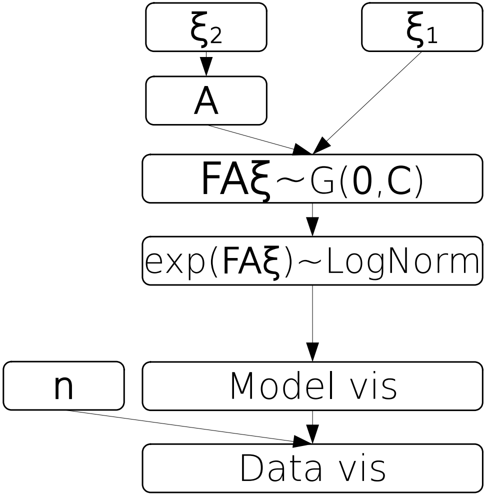
How to construct the likelihood $\mathcal{P}(d|s)$?

Measurement equation:

$$V = RI + n$$

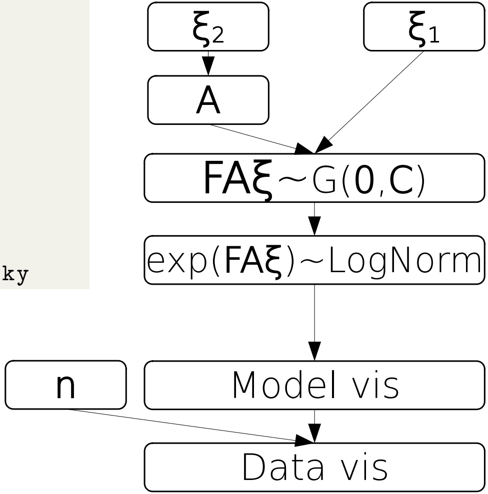
- Visibilities V
- Response R
- Sky brightness I
- Noise n
- Gaussian distributed noise $n \sim \mathcal{G}(0, N)$
- Likelihood:

$$\mathcal{P}(V|I) = \mathcal{G}(V - RI, N)$$



How to construct the likelihood $\mathcal{P}(d|s)$?

```python
import numpy as np
import nifty8 as ift
import resolve as rve

cfm = ift.SimpleCorrelatedField(...)
sky = ift.exp(cfm)

R = rve.InterferometryResponse(...)

log_lh = ift.GaussianEnergy(...) @ R @ sky
```



Inference

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

Inference

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

Inference

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

Inference

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

Variational Inference for posterior approximation:

Inference

$$\mathcal{P}(\xi|d) = \frac{\mathcal{P}(d|s(\xi))\mathcal{P}(\xi)}{\mathcal{P}(d)}$$

Variational Inference for posterior approximation:

```
import numpy as np
import nifty8 as ift
import resolve as rve

cfm = ift.SimpleCorrelatedField(...)
sky = ift.exp(cfm)
R = rve.InterferometryResponse(...)
log_lh = ift.GaussianEnergy(...) @ R @ sky

samples = ift.optimize_kl(log_lh, ...)
```

MGVI – Metric Gaussian Variational Inference[1]

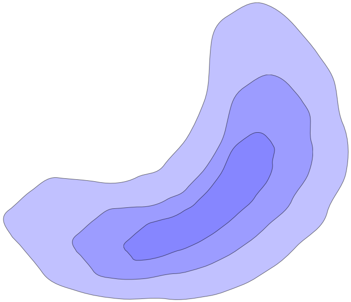
[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

MGVI – Metric Gaussian Variational Inference[1]

- Gaussian Prior: $P(\xi) = \mathcal{G}(0, 1)$

[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

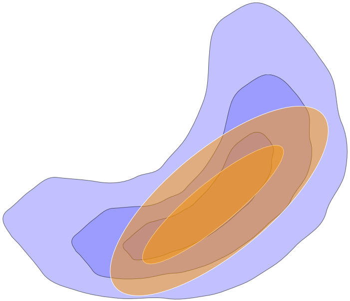
MGVI – Metric Gaussian Variational Inference[1]



- Gaussian Prior: $P(\xi) = \mathcal{G}(0, 1)$
- Non Gaussian Posterior: $P(\xi|d)$

[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

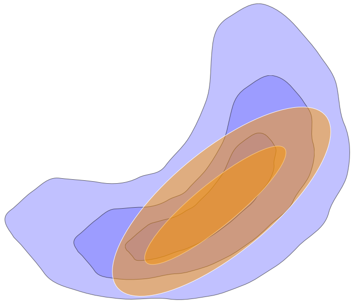
MGVI – Metric Gaussian Variational Inference[1]



- Gaussian Prior: $P(\xi) = \mathcal{G}(0, 1)$
- Non Gaussian Posterior: $P(\xi|d)$
- Gaussian approximation:
  Approximate $P(\xi|d)$ with $Q(\xi) = \mathcal{G}(\bar{\xi}, \Xi)$

[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

MGVI – Metric Gaussian Variational Inference[1]

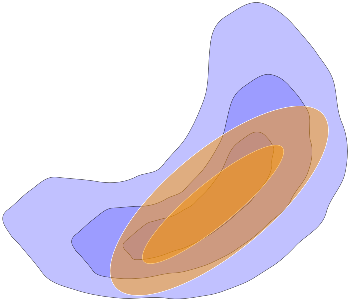


- Gaussian Prior: $P(\xi) = \mathcal{G}(0, 1)$
- Non Gaussian Posterior: $P(\xi|d)$
- Gaussian approximation:
  Approximate $P(\xi|d)$ with $Q(\xi) = \mathcal{G}(\bar{\xi}, \Xi)$

How to determine $\bar{\xi}$ and Ξ:

[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

MGVI – Metric Gaussian Variational Inference[1]



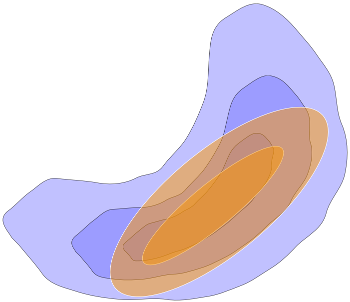
- Gaussian Prior: $P(\xi) = \mathcal{G}(0, 1)$
- Non Gaussian Posterior: $P(\xi|d)$
- Gaussian approximation:
  Approximate $P(\xi|d)$ with $Q(\xi) = \mathcal{G}(\bar{\xi}, \Xi)$

How to determine $\bar{\xi}$ and Ξ:

- $\bar{\xi} = \mathrm{argmin}_{\bar{\xi}}(\mathcal{D}_{KL}(Q||P))$

[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

MGVI – Metric Gaussian Variational Inference[1]



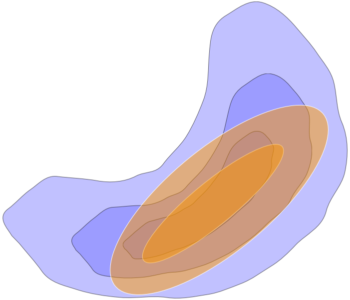
- Gaussian Prior: $P(\xi) = \mathcal{G}(0, 1)$
- Non Gaussian Posterior: $P(\xi|d)$
- Gaussian approximation:
  Approximate $P(\xi|d)$ with $Q(\xi) = \mathcal{G}(\bar{\xi}, \Xi)$

How to determine $\bar{\xi}$ and Ξ:

- $\bar{\xi} = \mathrm{argmin}_{\bar{\xi}}(\mathcal{D}_{KL}(Q||P))$
- $\Xi^{-1}(\bar{\xi}) = \left\langle \frac{\partial^2 \mathcal{H}(d,\bar{\xi})}{\partial\bar{\xi}\partial\bar{\xi}^\dagger} \right\rangle_{P(d|\bar{\xi})}$
  with $\mathcal{H} = -\ln(P)$
- Linear scaling with dimensions of ξ

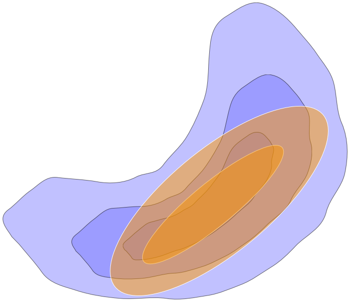
[1] Jakob Knollmüller and Torsten A. Enßlin. "Metric Gaussian Variational Inference". In: (Jan. 30, 2019). arXiv: 1901.11033v3 [stat.ML].

GeoVI – Geometric Variational Inference[2]



[2] Philipp Frank, Reimar Leike, and Torsten A. Enßlin. "Geometric Variational Inference". In: *Entropy* 23.7 (2021). ISSN: 1099-4300. DOI: 10.3390/e23070853. URL: https://www.mdpi.com/1099-4300/23/7/853.

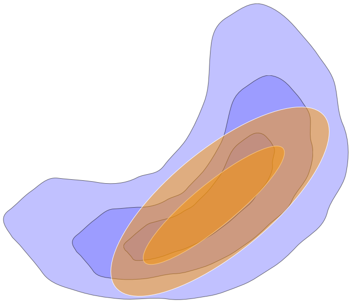
GeoVI – Geometric Variational Inference[2]



- Coordinate transformation in latent space

[2] Philipp Frank, Reimar Leike, and Torsten A. Enßlin. "Geometric Variational Inference". In: *Entropy* 23.7 (2021). ISSN: 1099-4300. DOI: 10.3390/e23070853. URL: https://www.mdpi.com/1099-4300/23/7/853.

GeoVI – Geometric Variational Inference[2]

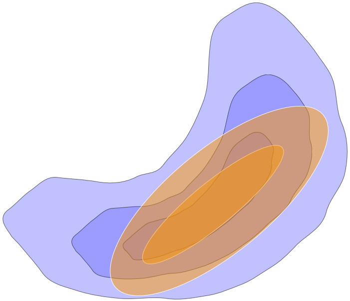


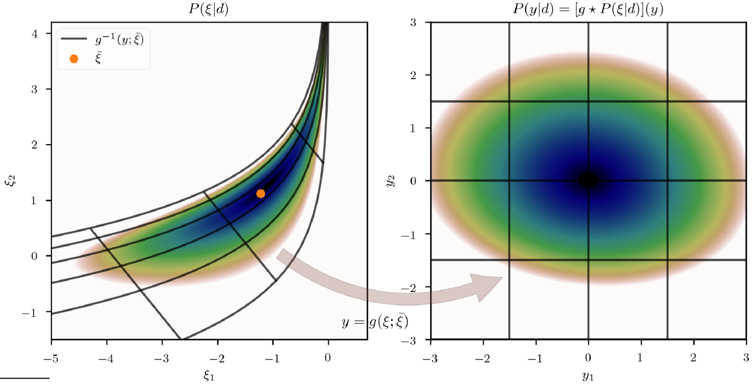
- Coordinate transformation in latent space
- Approximately transform the Posterior into a Gaussian

[2] Philipp Frank, Reimar Leike, and Torsten A. Enßlin. "Geometric Variational Inference". In: *Entropy* 23.7 (2021). ISSN: 1099-4300. DOI: 10.3390/e23070853. URL: https://www.mdpi.com/1099-4300/23/7/853.

GeoVI – Geometric Variational Inference[2]

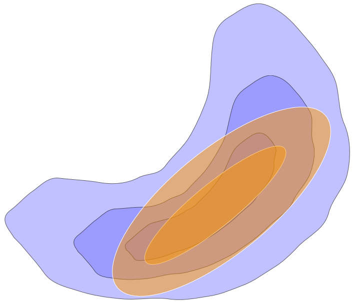
- Coordinate transformation in latent space
- Approximately transform the Posterior into a Gaussian



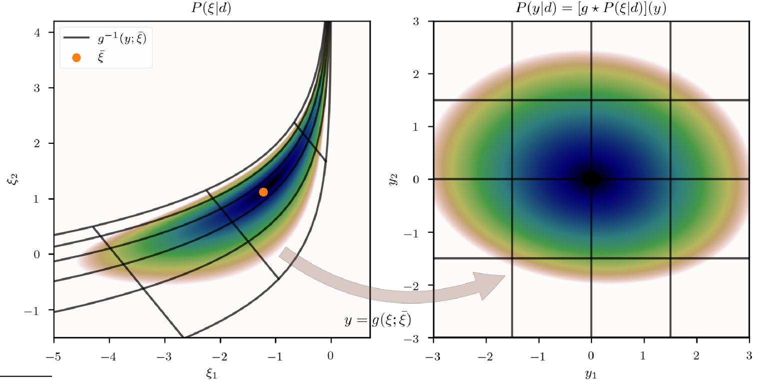


[2] Philipp Frank, Reimar Leike, and Torsten A. Enßlin. "Geometric Variational Inference". In: *Entropy* 23.7 (2021). ISSN: 1099-4300. DOI: 10.3390/e23070853. URL: https://www.mdpi.com/1099-4300/23/7/853.

GeoVI – Geometric Variational Inference[2]



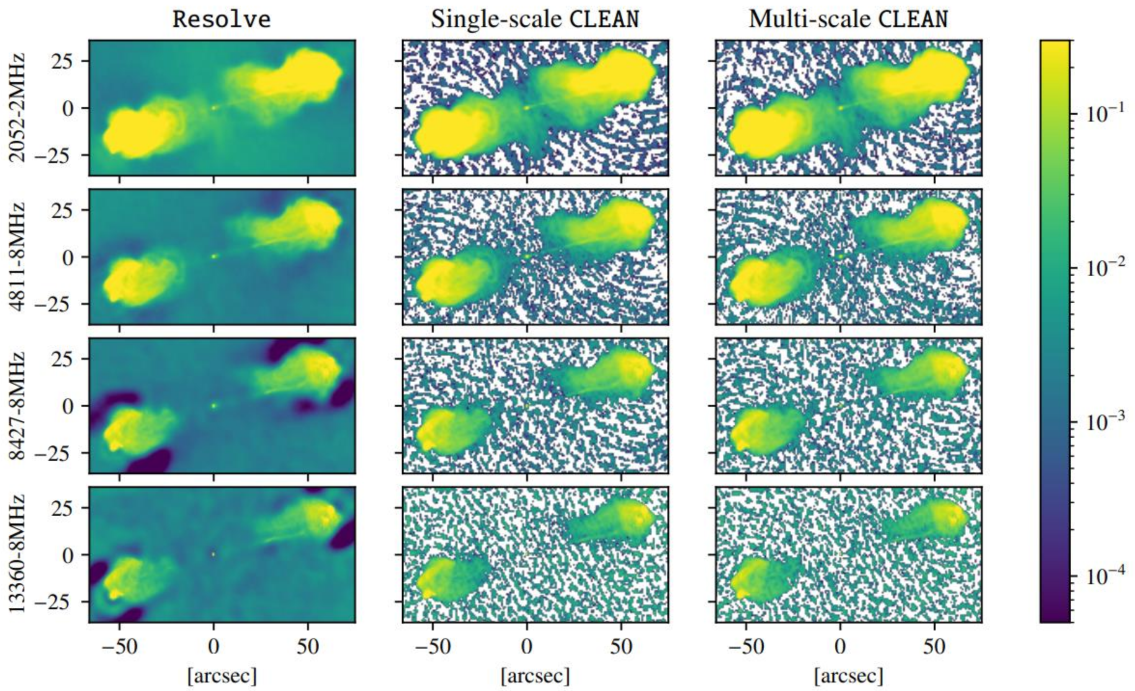
- Coordinate transformation in latent space
- Approximately transform the Posterior into a Gaussian
- Also linear scaling with number of dimensions



[2] Philipp Frank, Reimar Leike, and Torsten A. Enßlin. "Geometric Variational Inference". In: *Entropy* 23.7 (2021). ISSN: 1099-4300. DOI: 10.3390/e23070853. URL: https://www.mdpi.com/1099-4300/23/7/853.

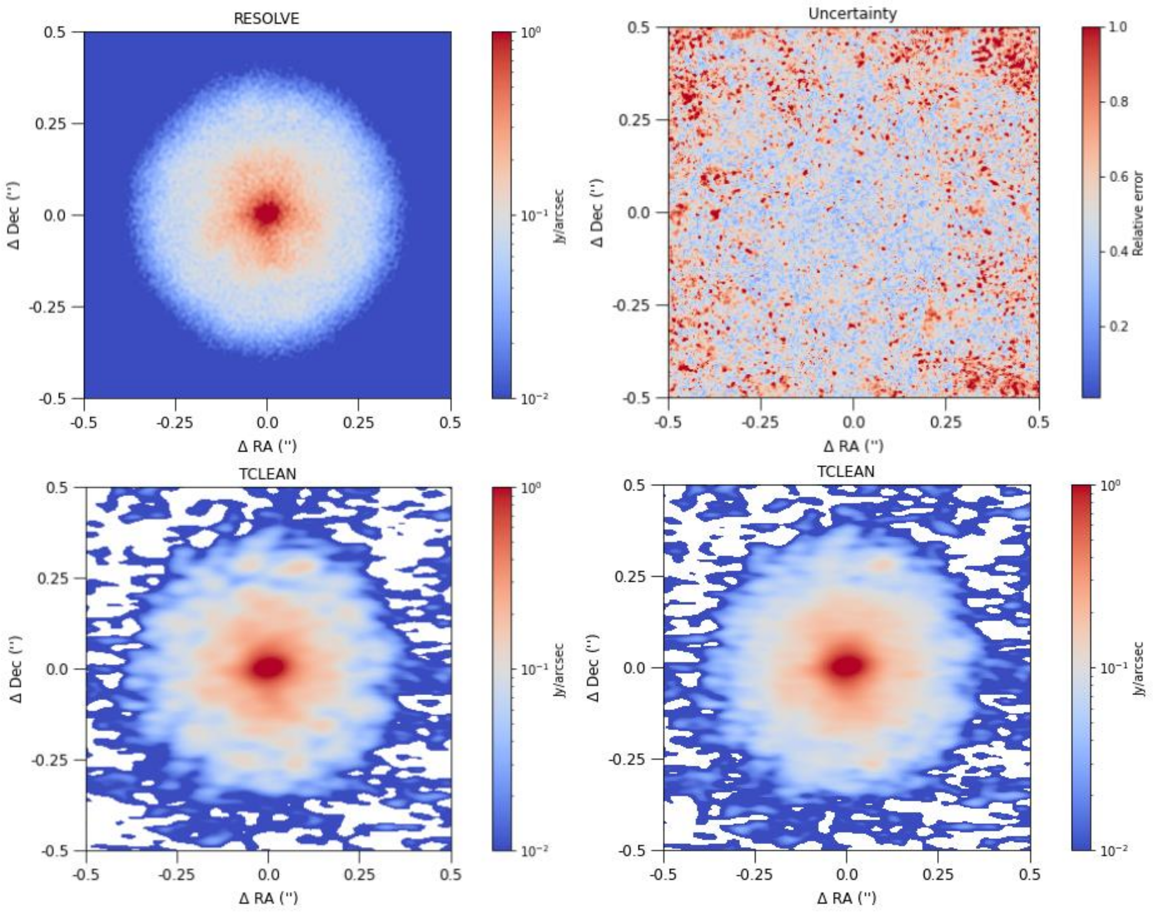
Imaging (VLA)

Arras et al., 2021



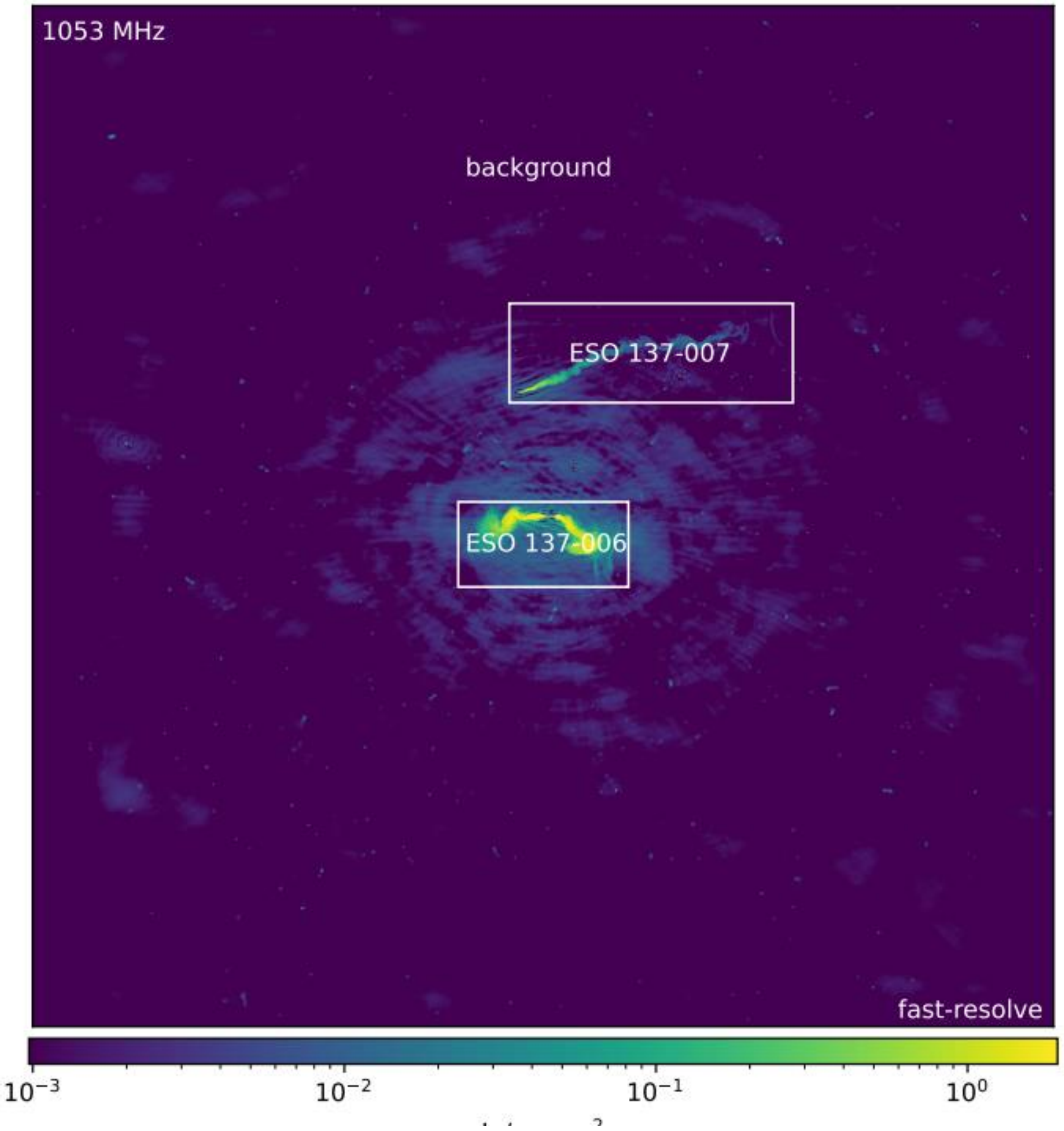
Imaging (ALMA)

Tychoniec et al., 2022



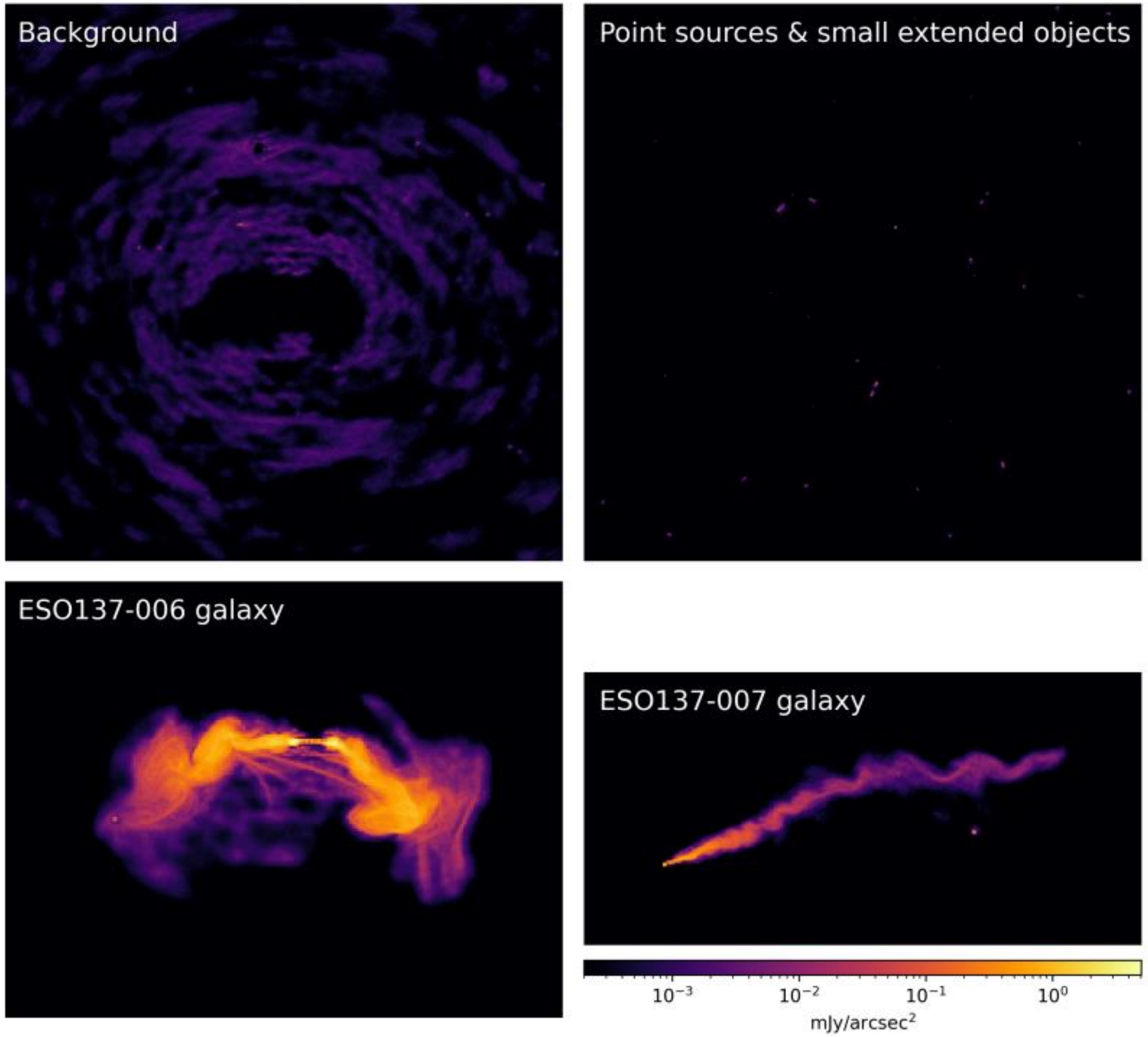
Imaging (MeerKAT)

Roth et al., 2024



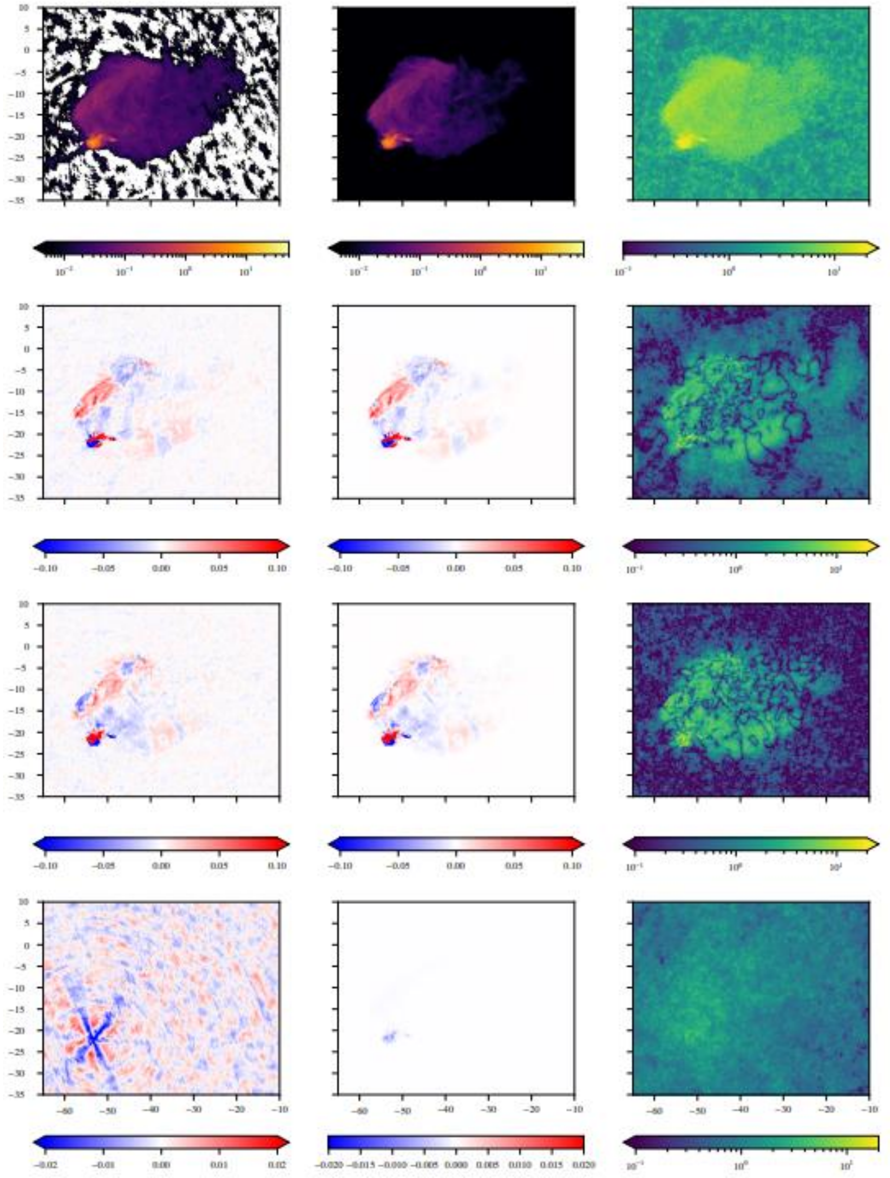
Component detection & separation

Fuchs et al., 2025



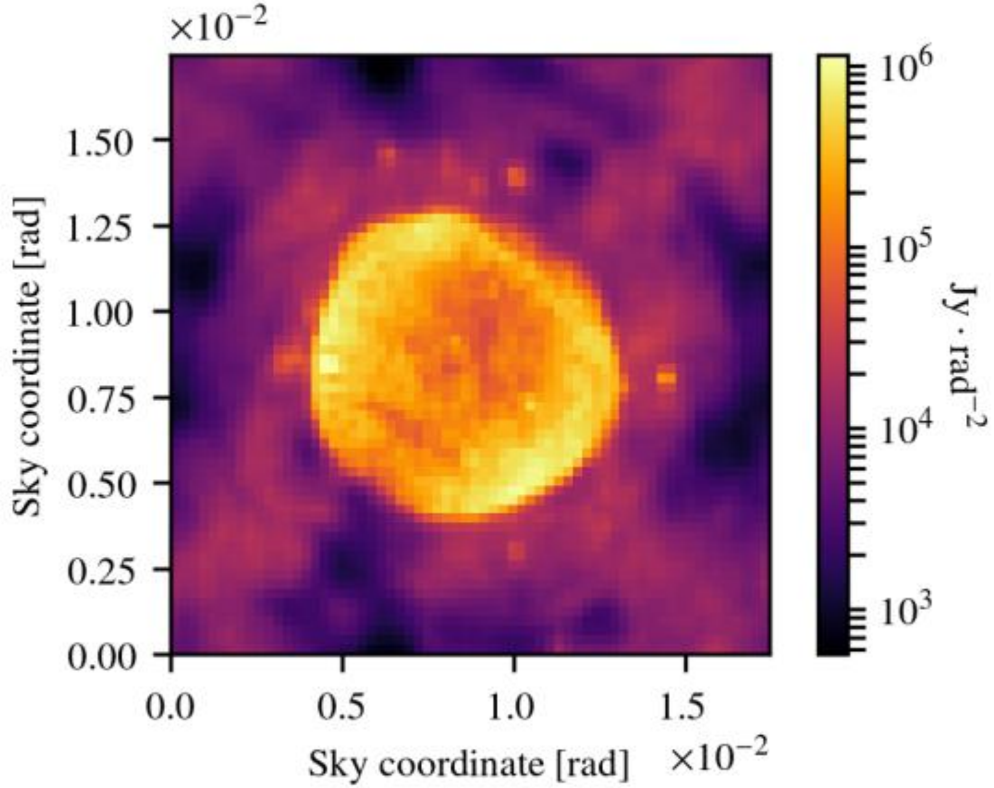
Polarization Imaging

Arras et al., 2025



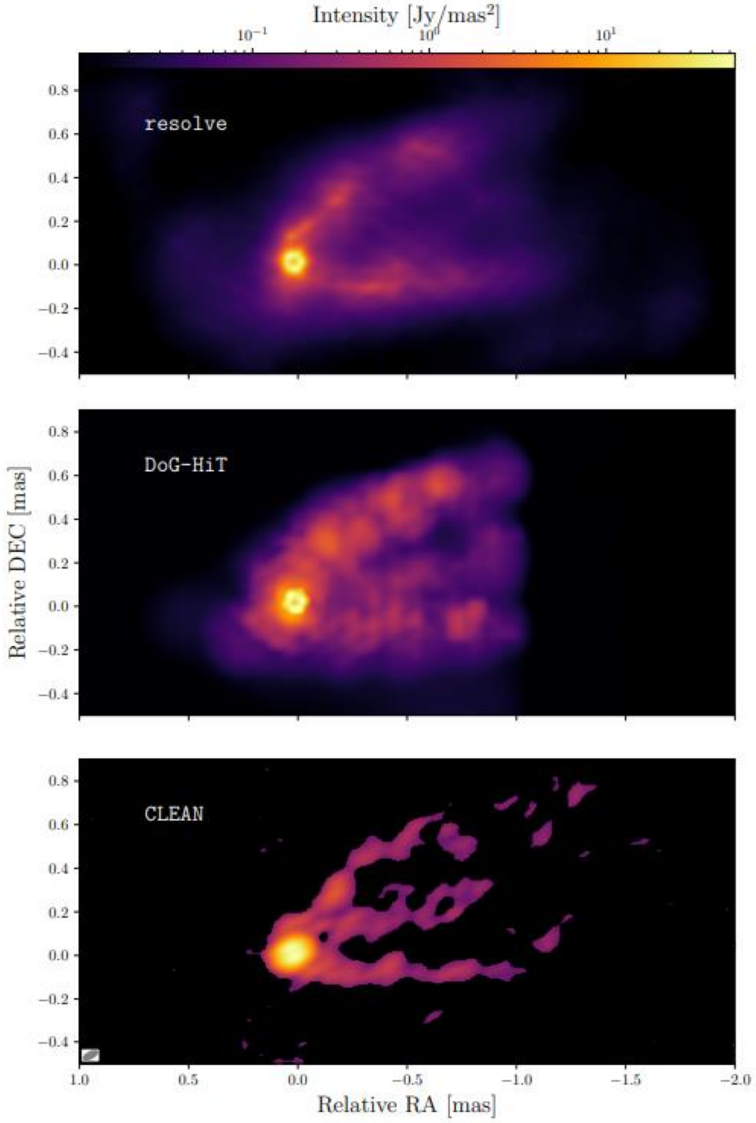
Imaging & Direction Independent Calibration

Arras et al., 2019



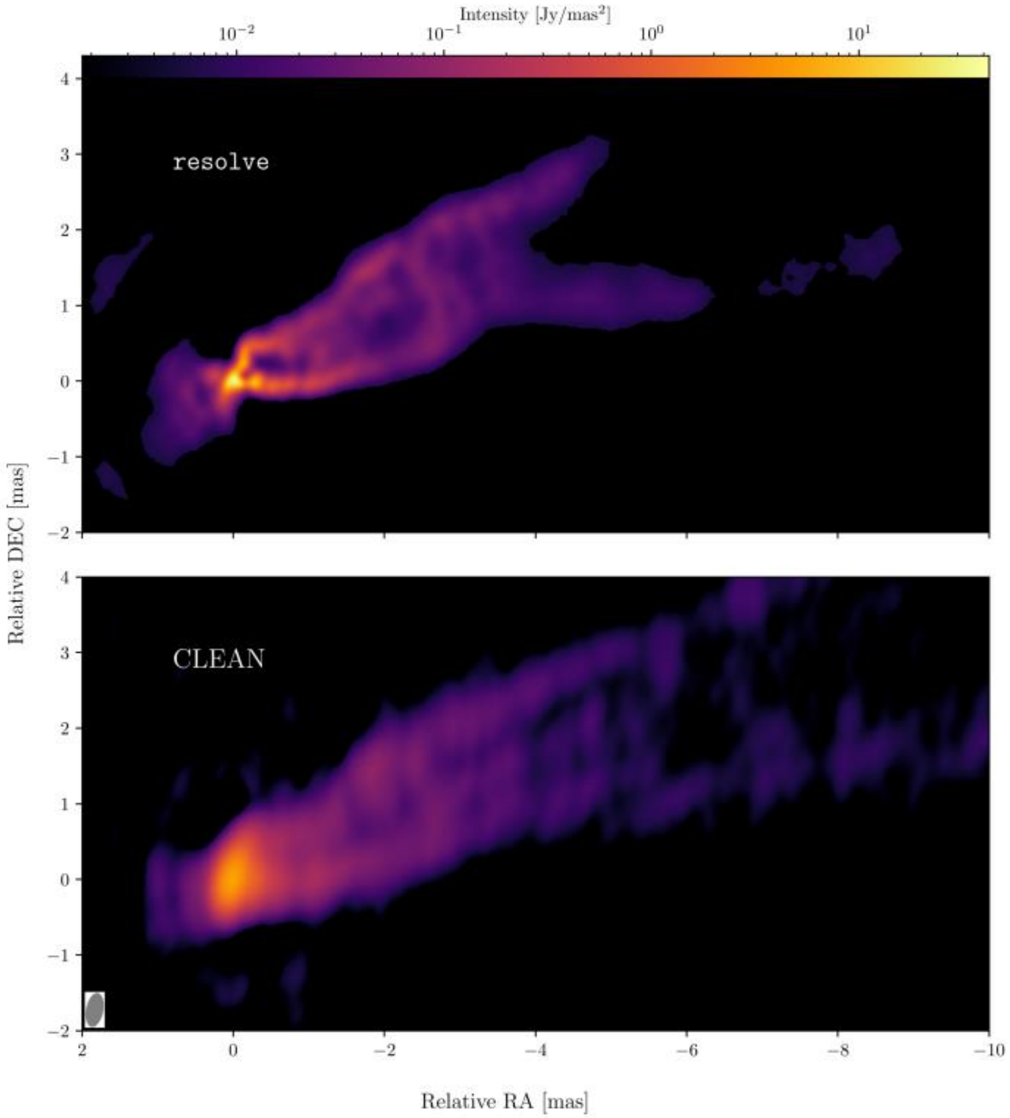
Imaging with VLBI

Kim et al., 2025



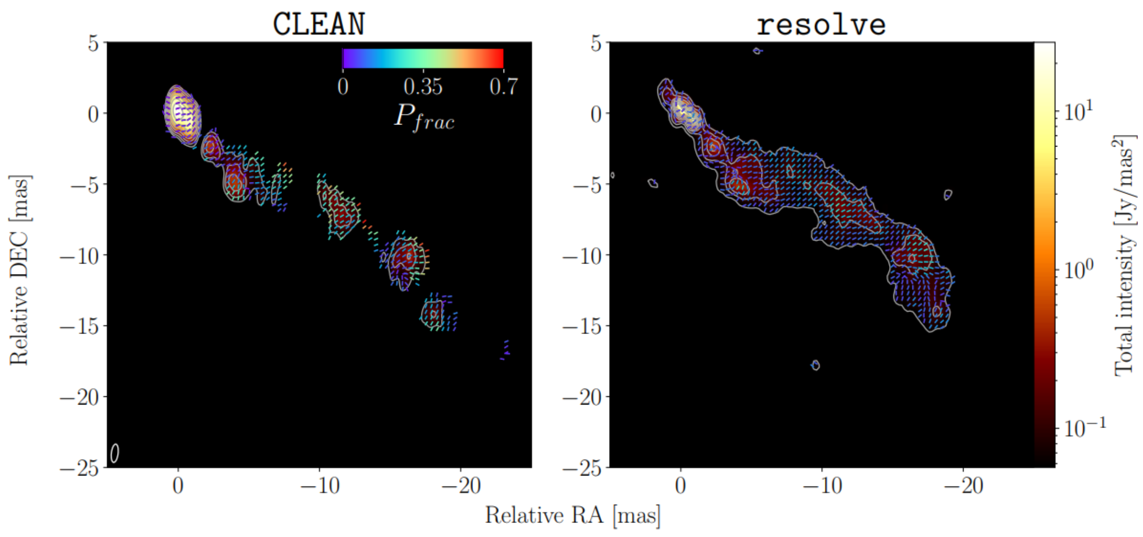
Imaging & Calibration with VLBI

Kim et al., 2024



Polarization Imaging & Calibration with VLBI

Kim et al., 2025



Where to find RESOLVE in the near future?

https://github.com/NIFTy-PPL/J-UBIK