LAMARR
INSTITUTE FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

RADIONETS
Deep learning-based imaging in radio interferometry

Deep learning-based imaging of Multi-source observations

Kevin Schmitz

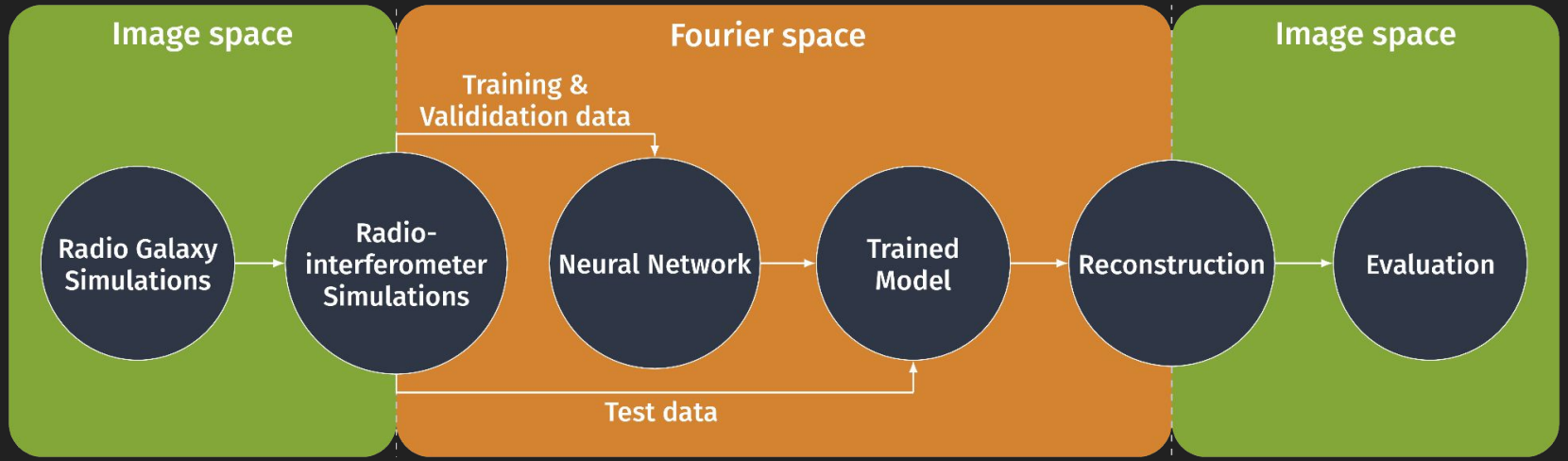
TU Dortmund University
LAMARR Institute for ML and AI

15.12.2025

astroparticle physics

PYVISGEN
Visibility Simulations in Python

radionets-project



Simulations

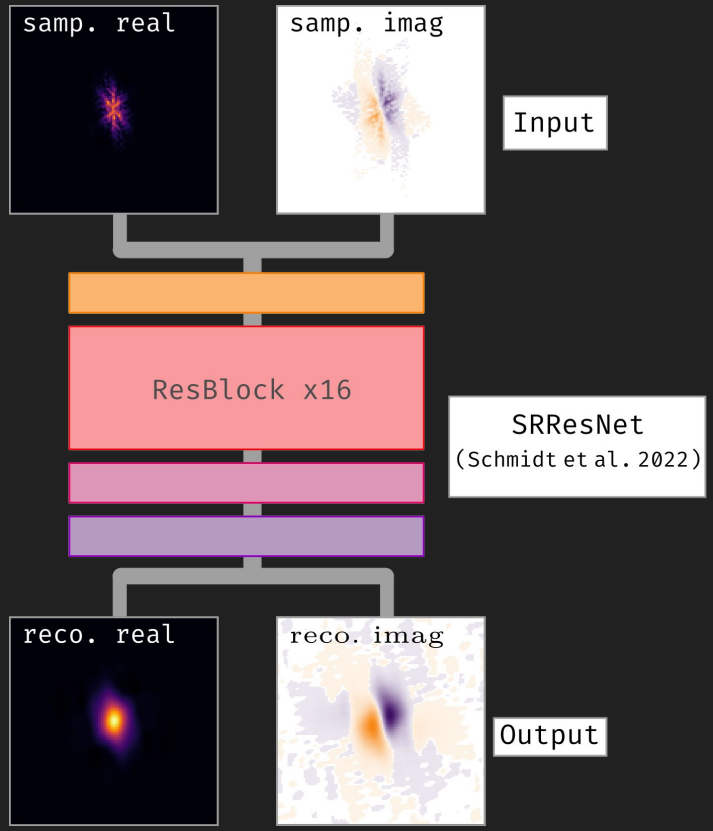
PYVISGEN
Visibility Simulations in Python

Model Building

Evaluation

RADIONETS
Deep learning based imaging in radio interferometry

Deep Learning-based Reconstruction



MeerKat data set

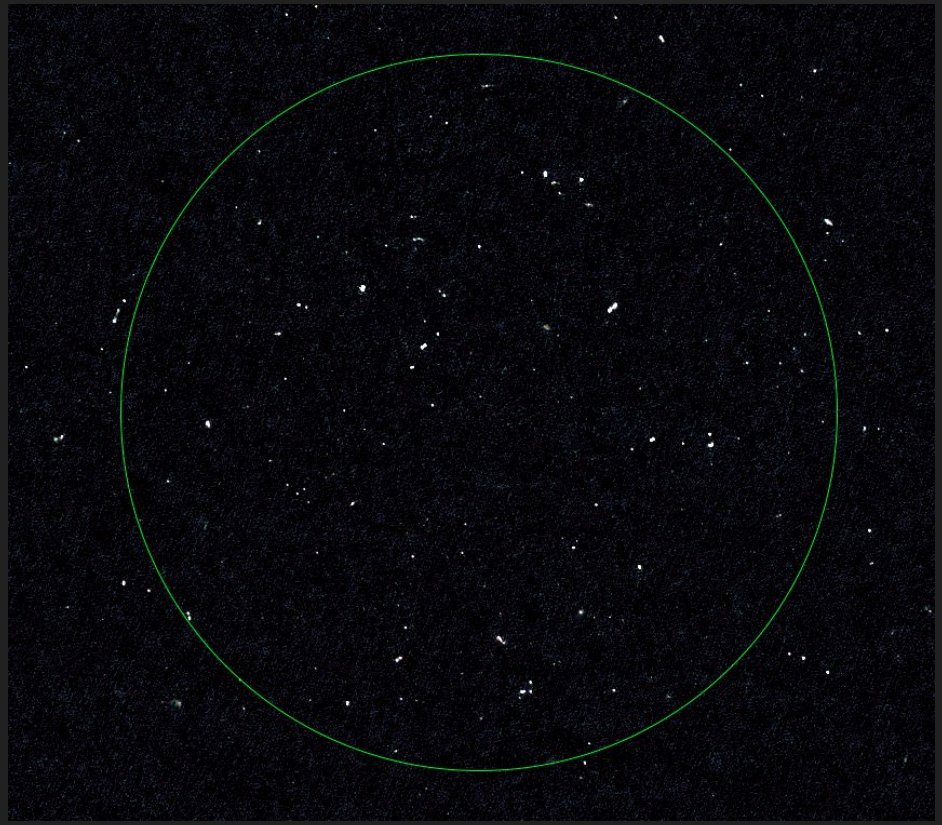
L- Band observations of Virgo cluster

pointing at the outer part

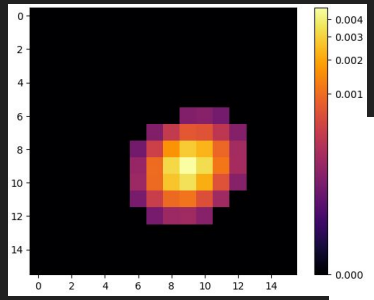
2 arcsec resolution

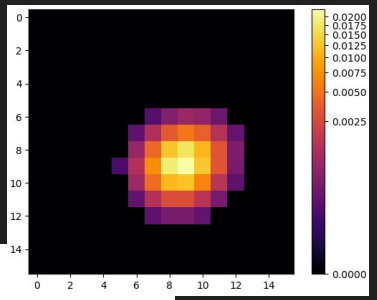
350.000 visibilities

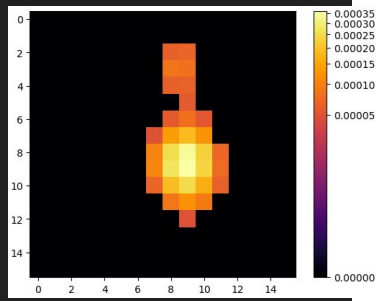
1.3 GHz – 16 channels

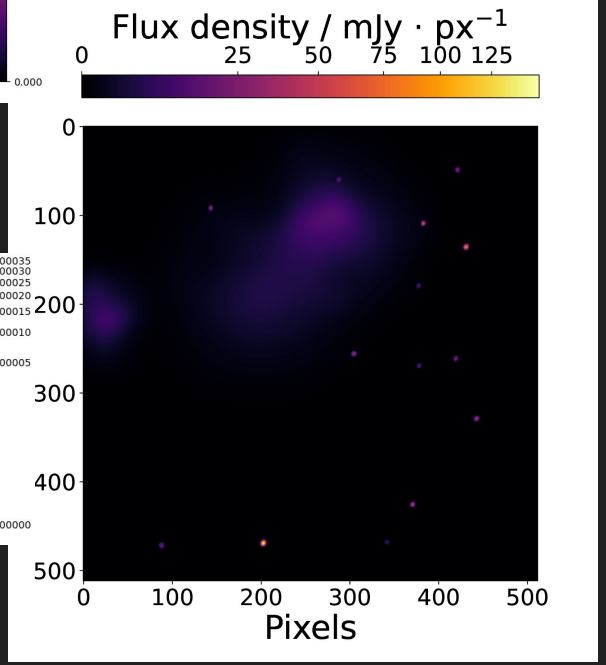


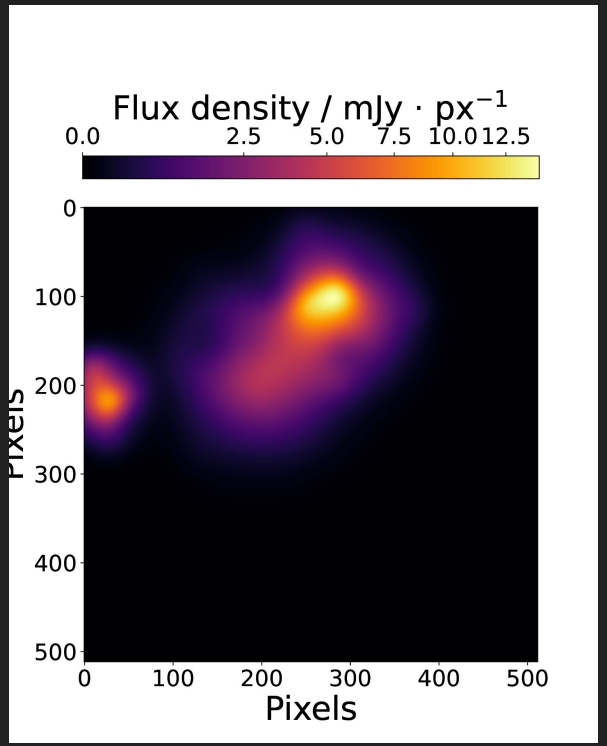
Synthetic radio skies









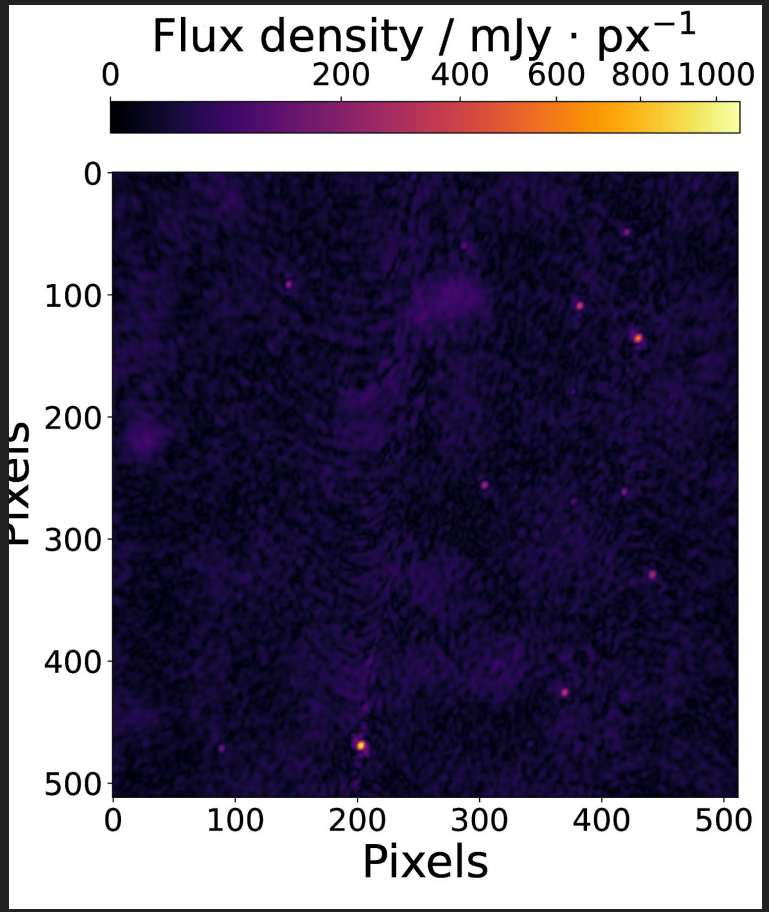


Visibility simulations

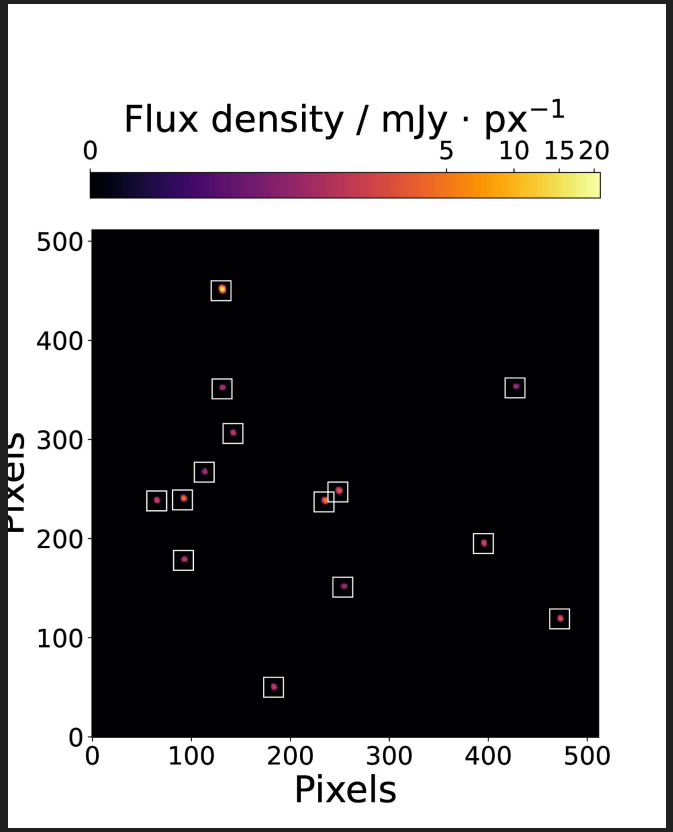
Radio interferometer measurement equation

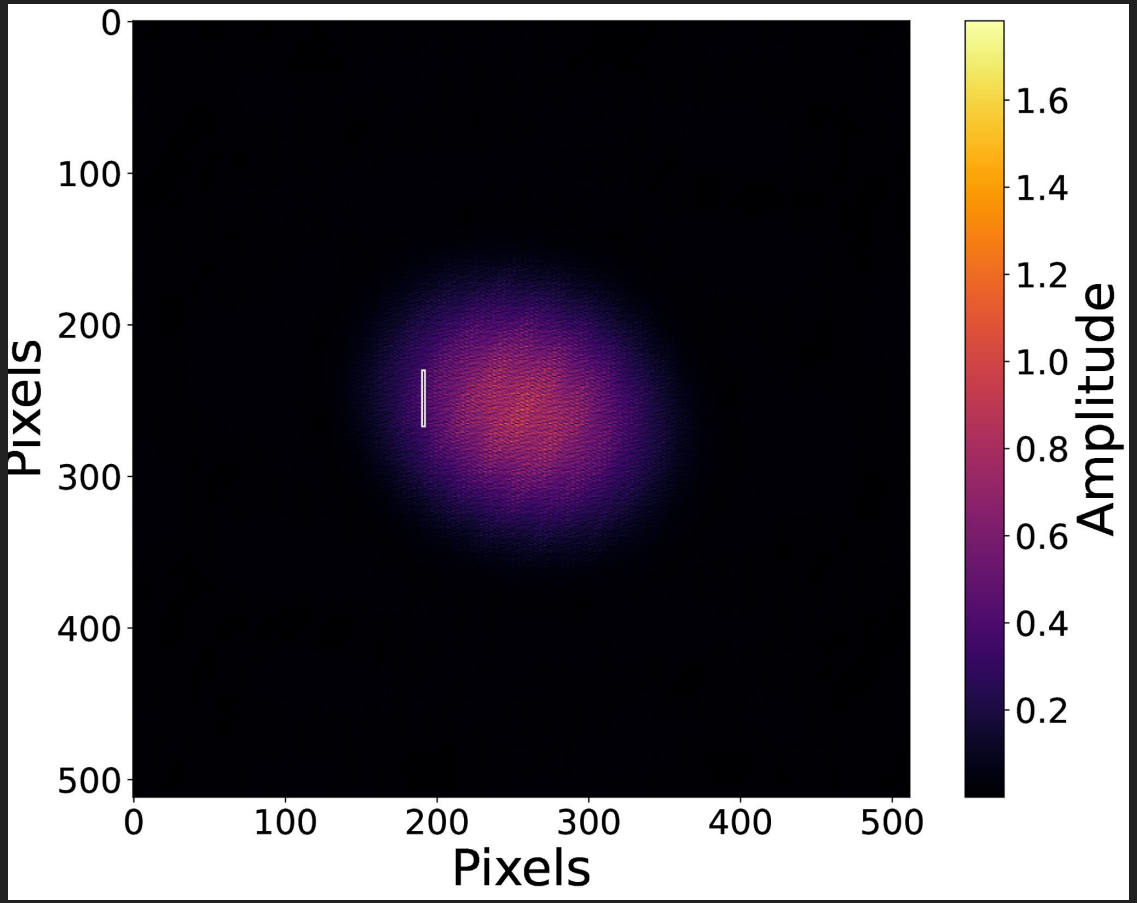
Considering (u, v, w) and (l, m, n)
-> non-coplanar effects

Import when creating dirty images

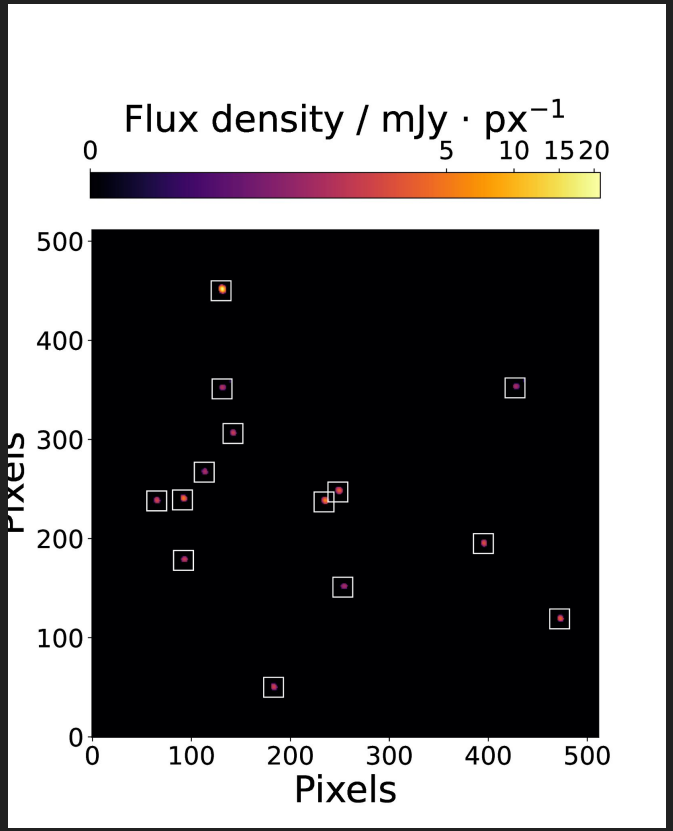


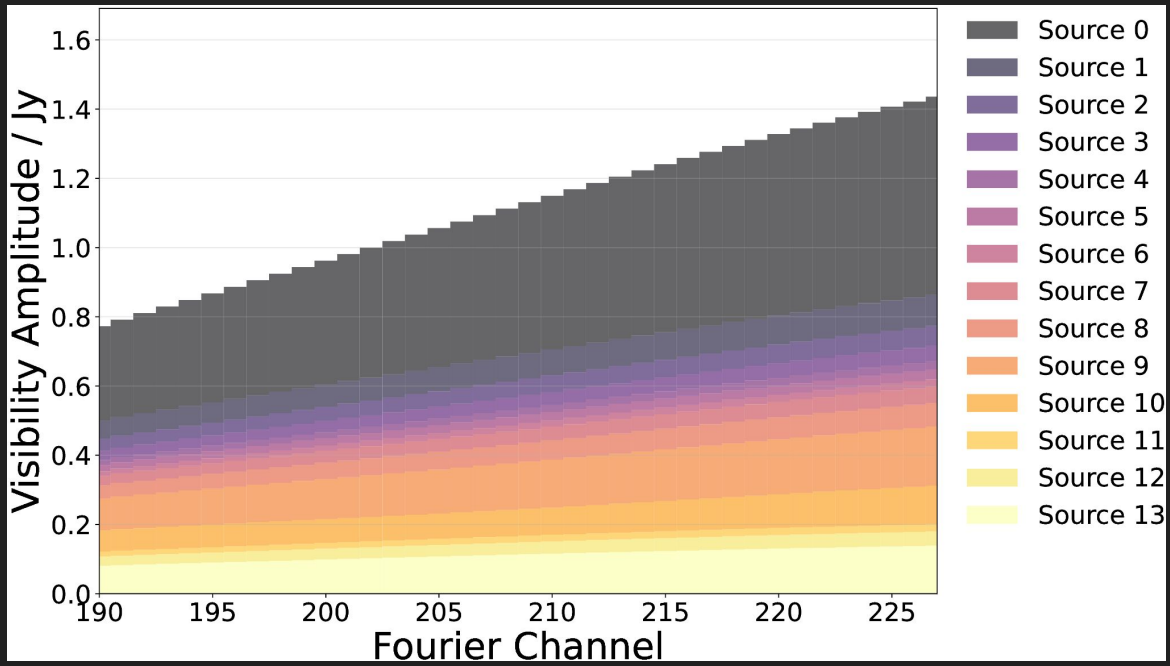
Visibility data



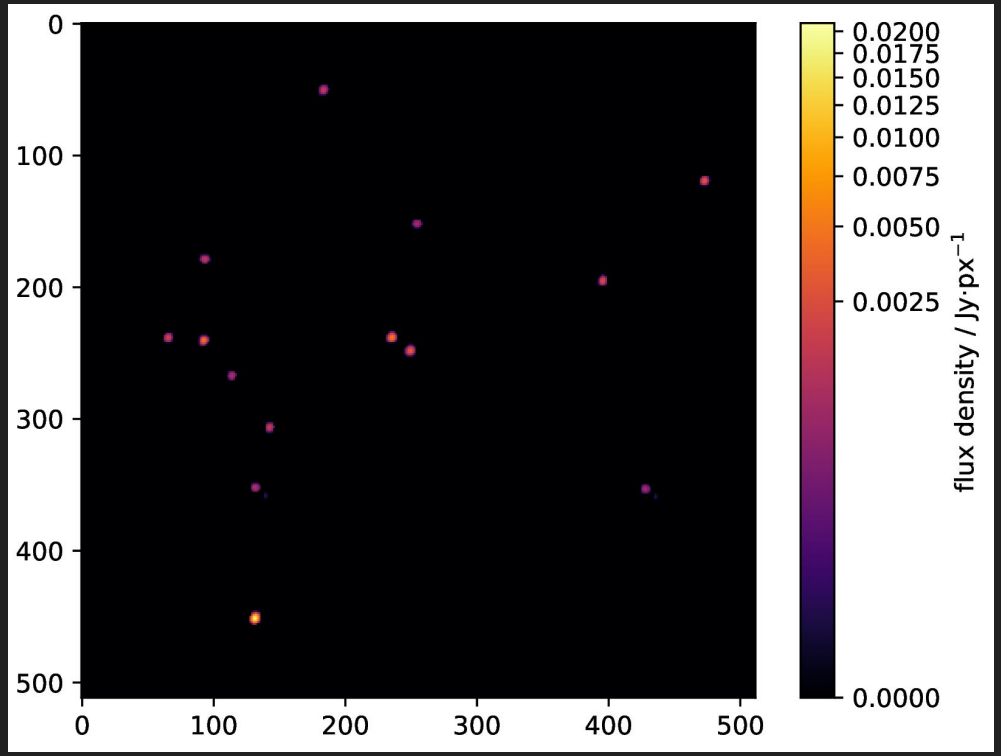


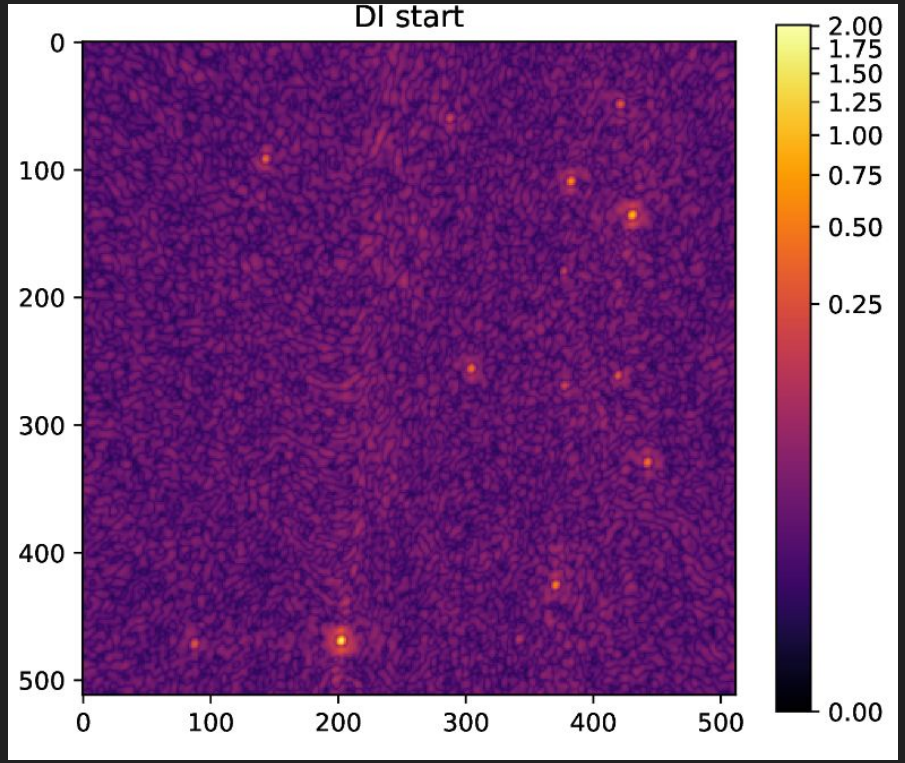
Visibility data



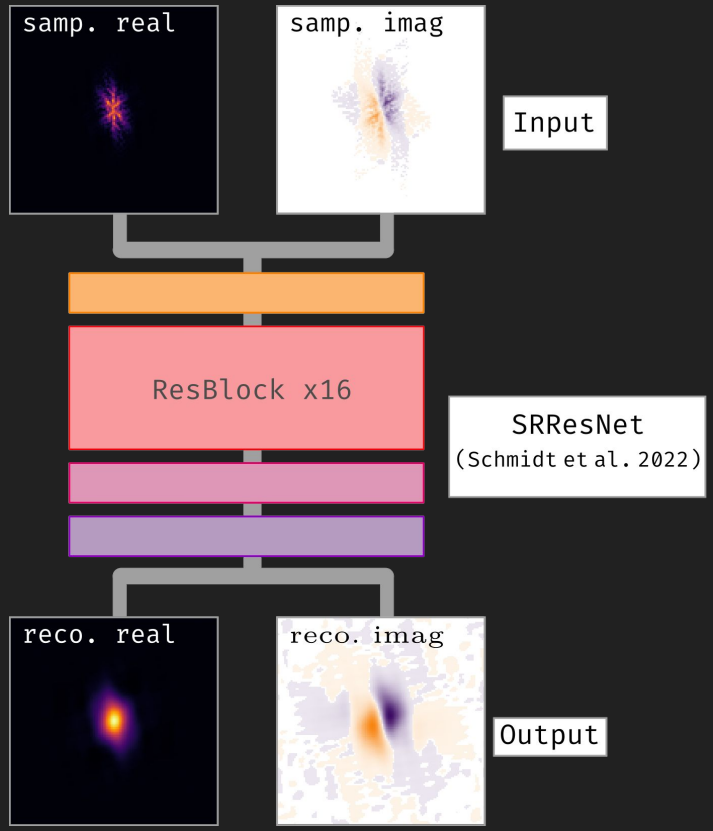


Visibility data





Deep Learning-based Reconstruction

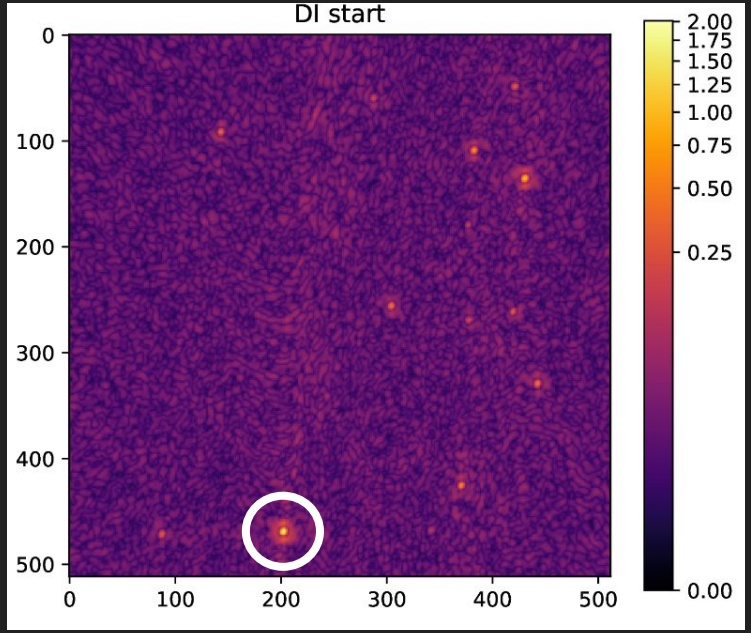


Process sky chunks

Create sky chunk with brightest source

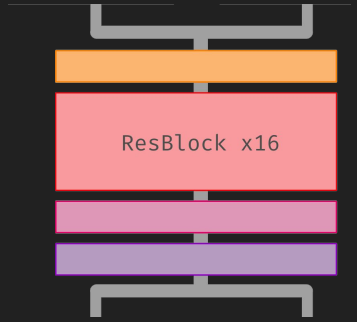
Reconstruction in visibility space with neural network

Subtract solution from complete visibility set

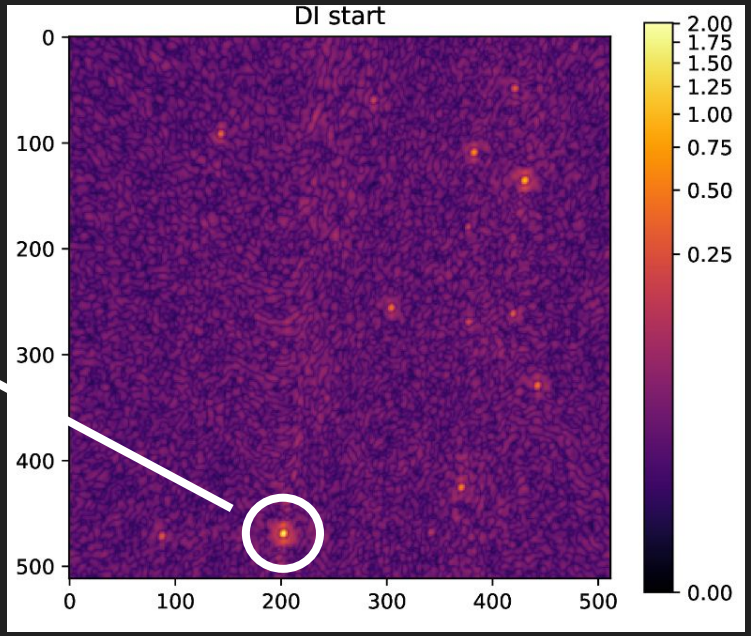


Process sky chunks

Amplitude & Phase

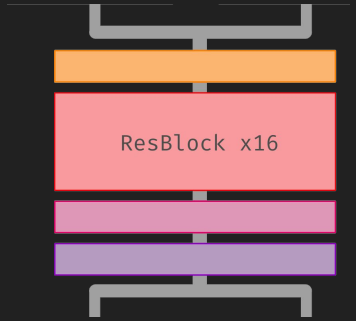


Model Visibilities
(Point sources)



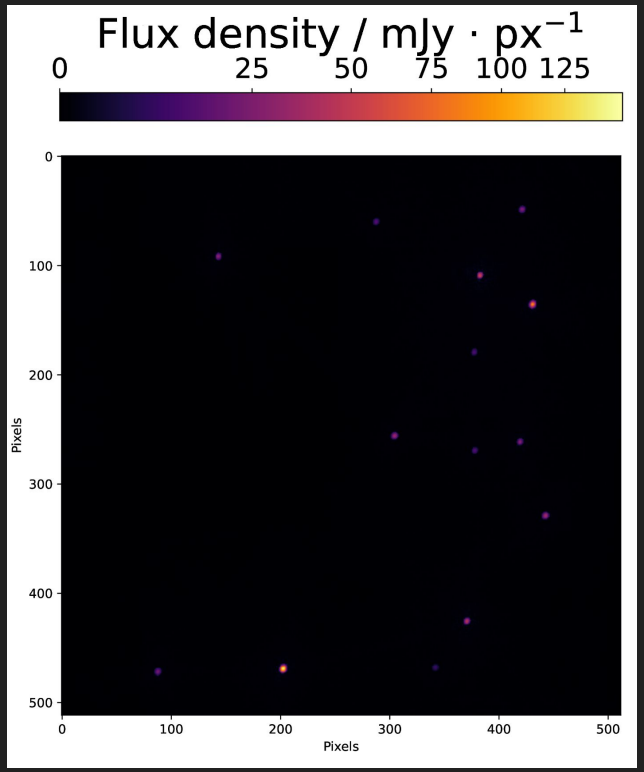
Process sky chunks

Amplitude & Phase



Model Visibilities
(Point sources)

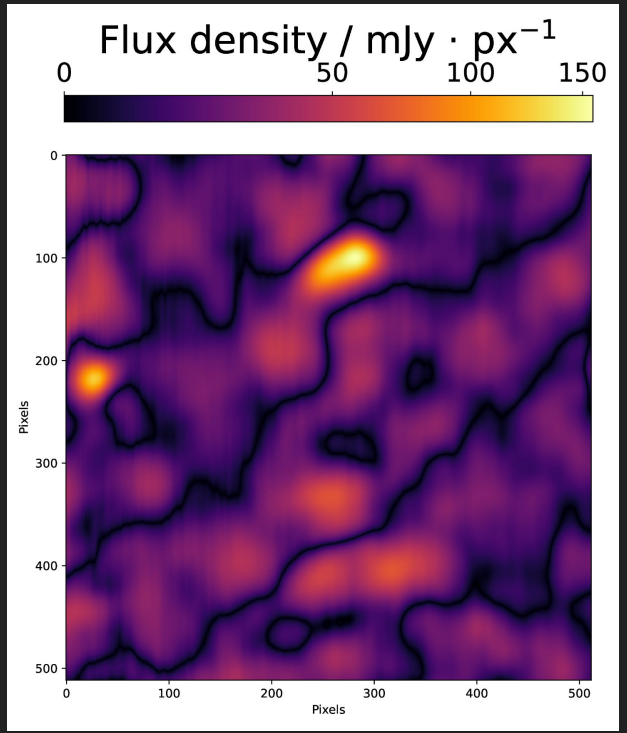
Point source sky model



Process sky chunks

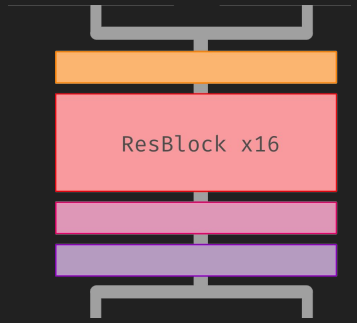
Visibilities − Model Visibilities (Point sources) →

Diffuse residual



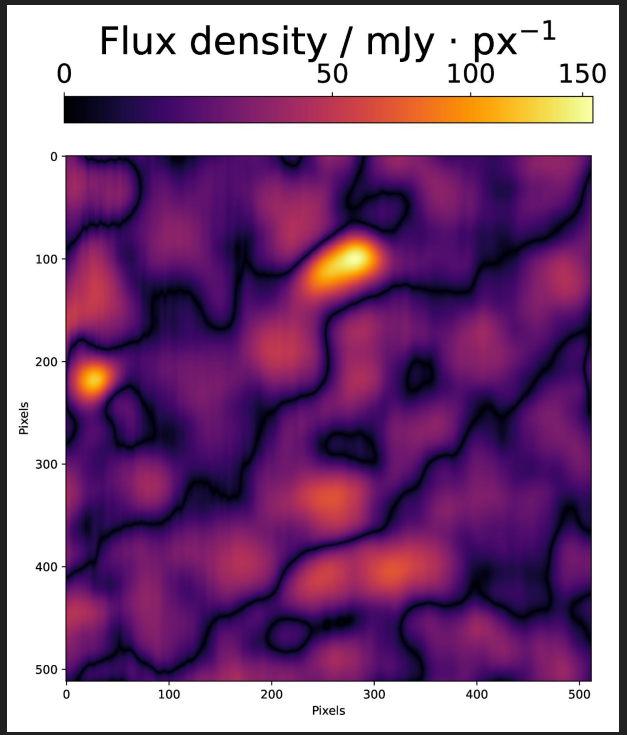
Process sky chunks

Amplitude & Phase



Model Visibilities
(Diffuse)

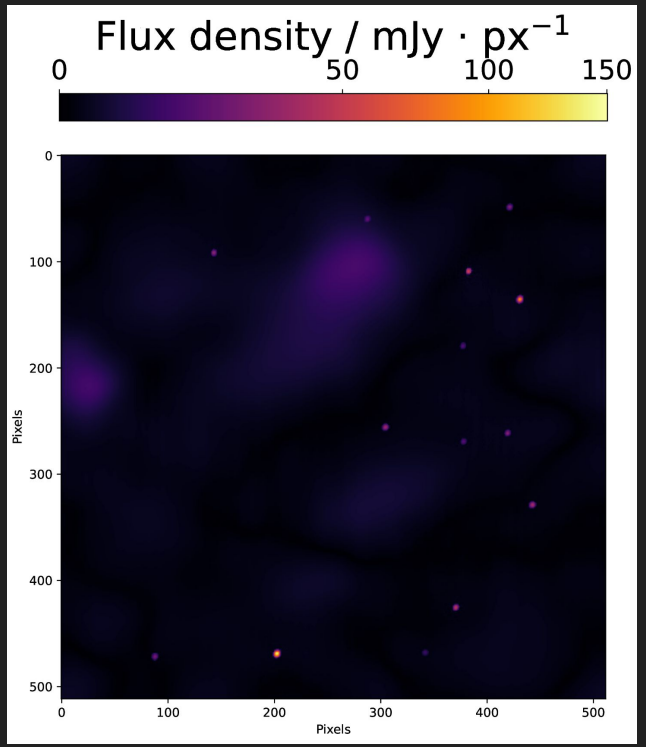
Diffuse residual



Process sky chunks

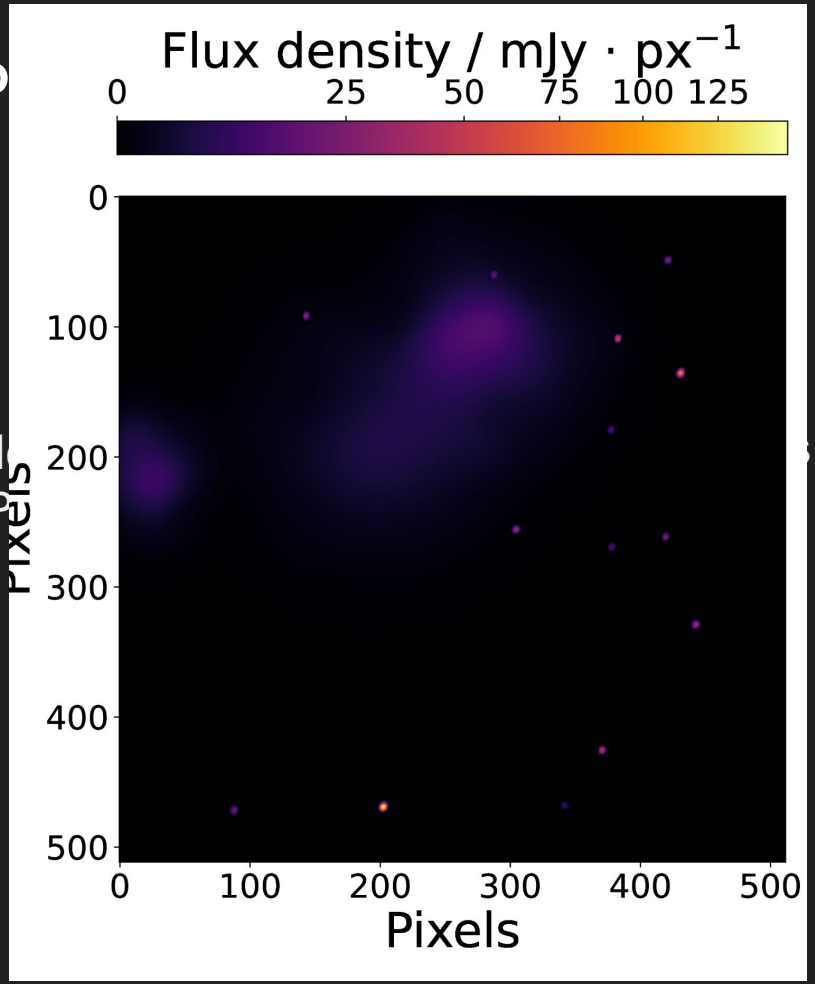
Model Visibilities (Point sources) + Model Visibilities (Diffuse) →

Reconstructed sky

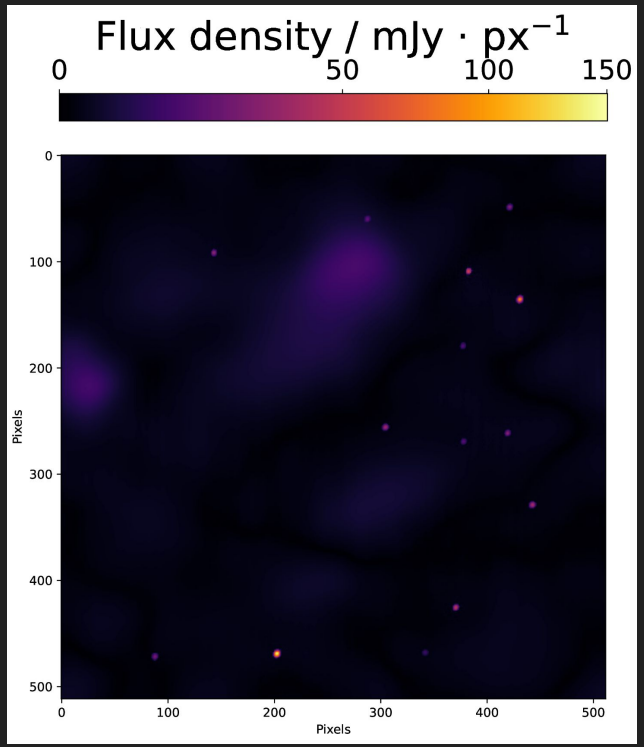


P

M
(P

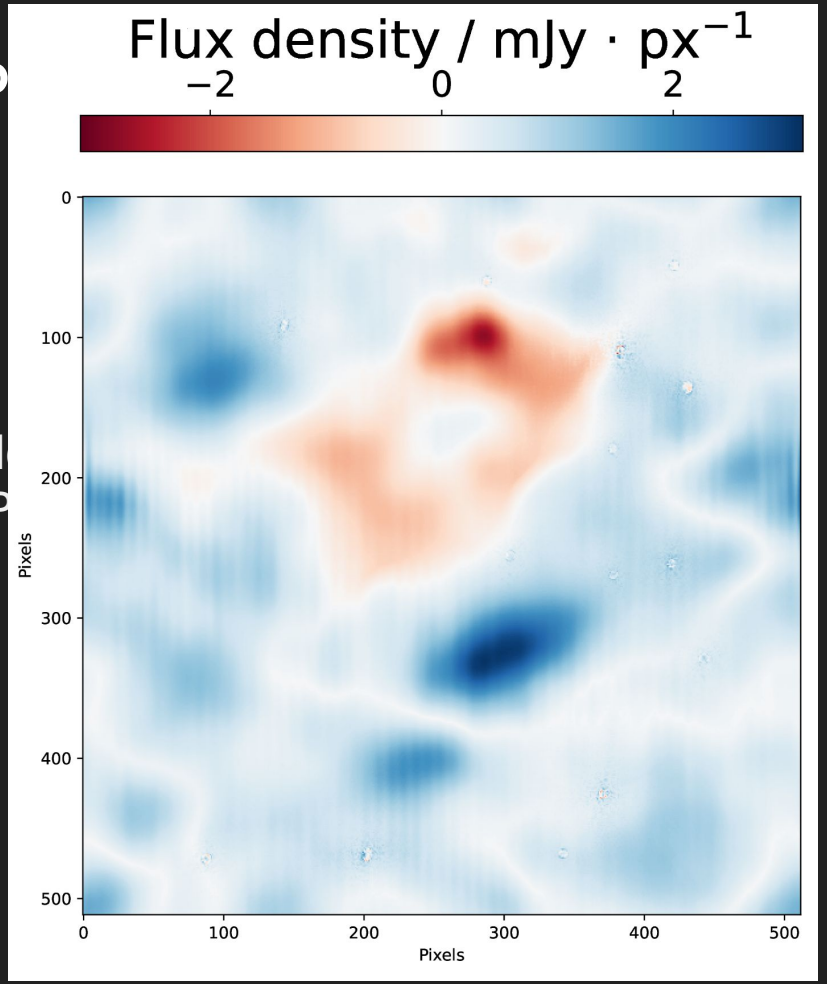


Reconstructed sky

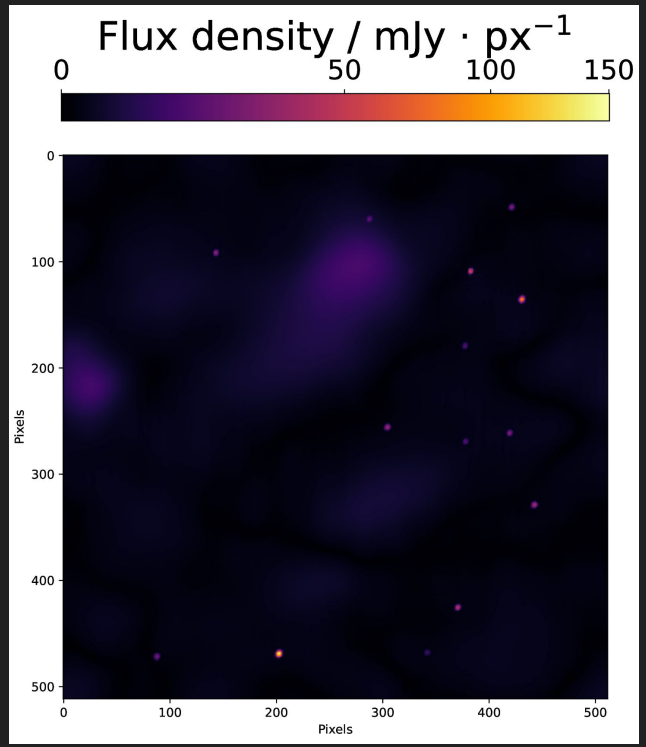


P

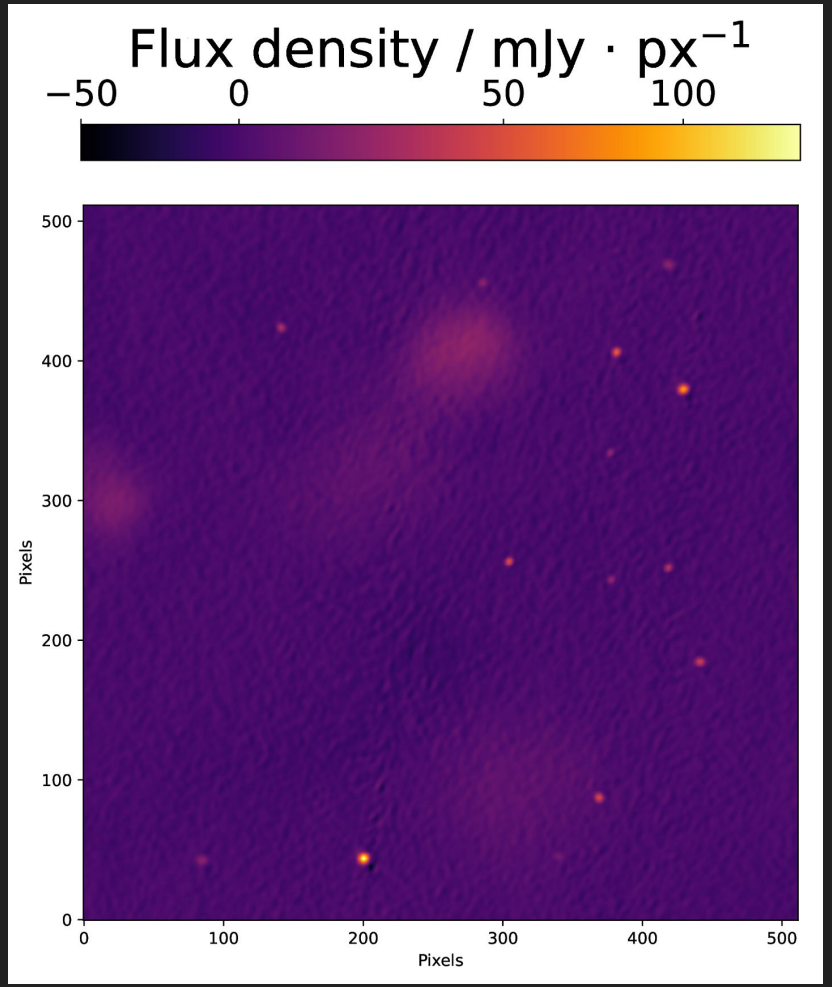
M
(P



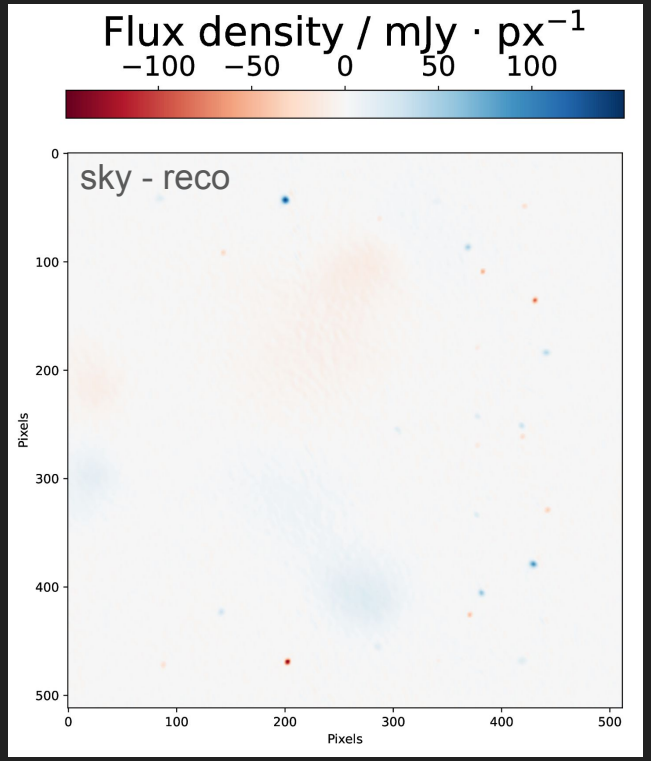
Reconstructed sky

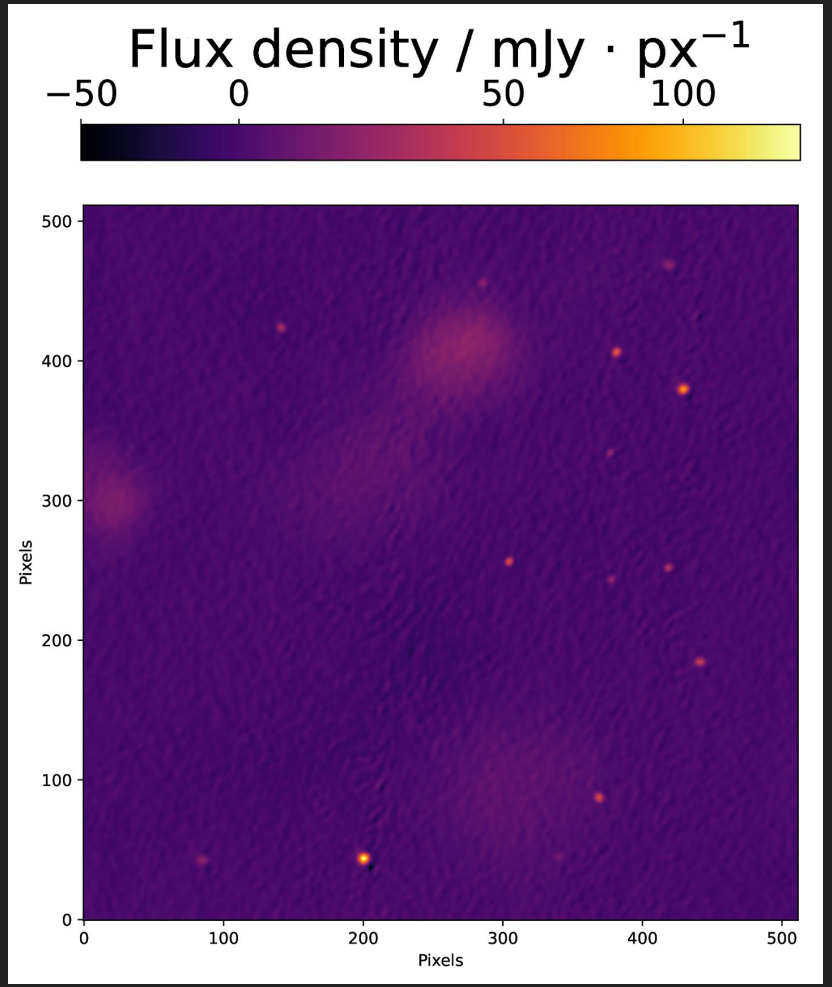


WSCLEAN

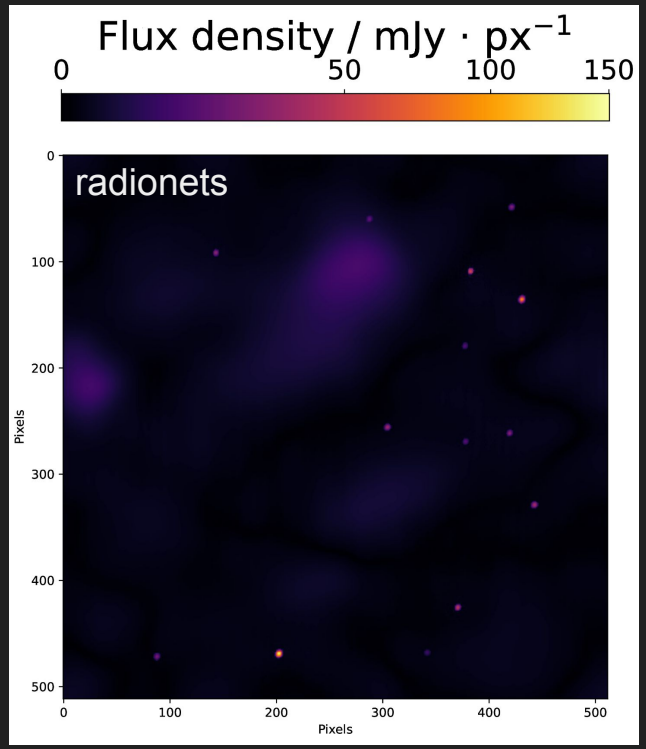

Flux density / mJy · px^{-1}
−50
0
50
100
500
400
300
200
100
0
Pixels
0
100
200
300
400
500
Pixels

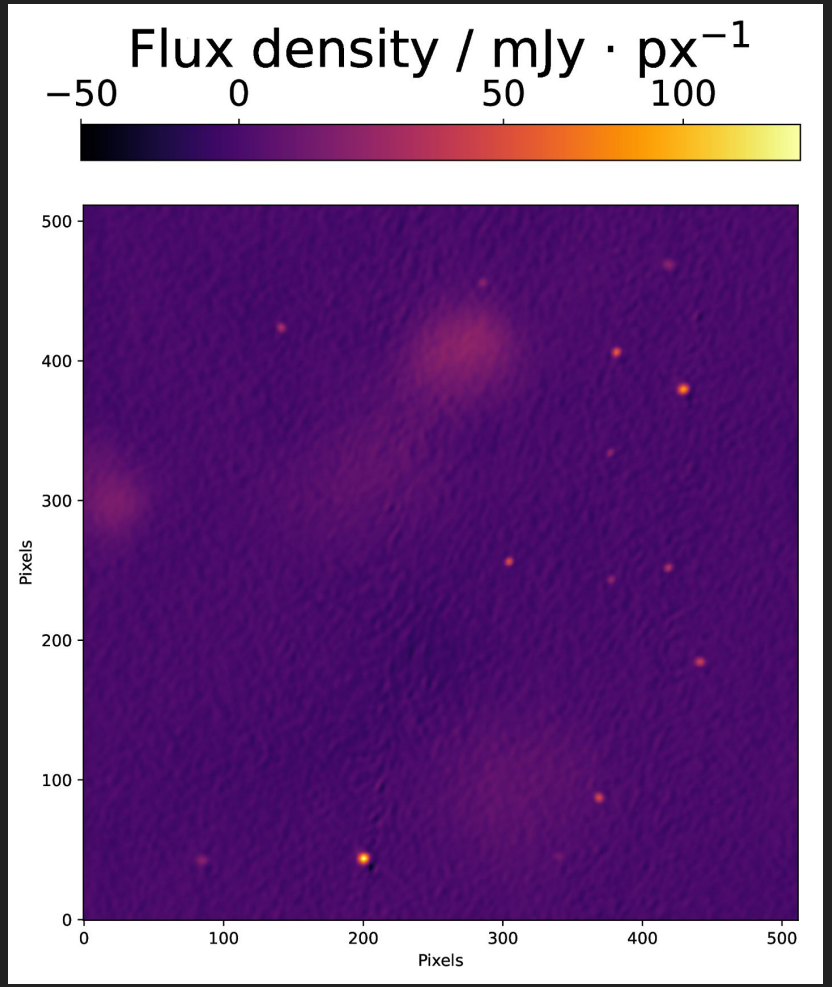
WSCLEAN


Flux density / mJy · px^{-1}
−100 −50 0 50 100
sky - reco
Pixels
Pixels
0
100
200
300
400
500


Flux density / mJy · px^{-1}
−50 0 50 100
Pixels
Pixels
0
100
200
300
400
500

WSCLEAN

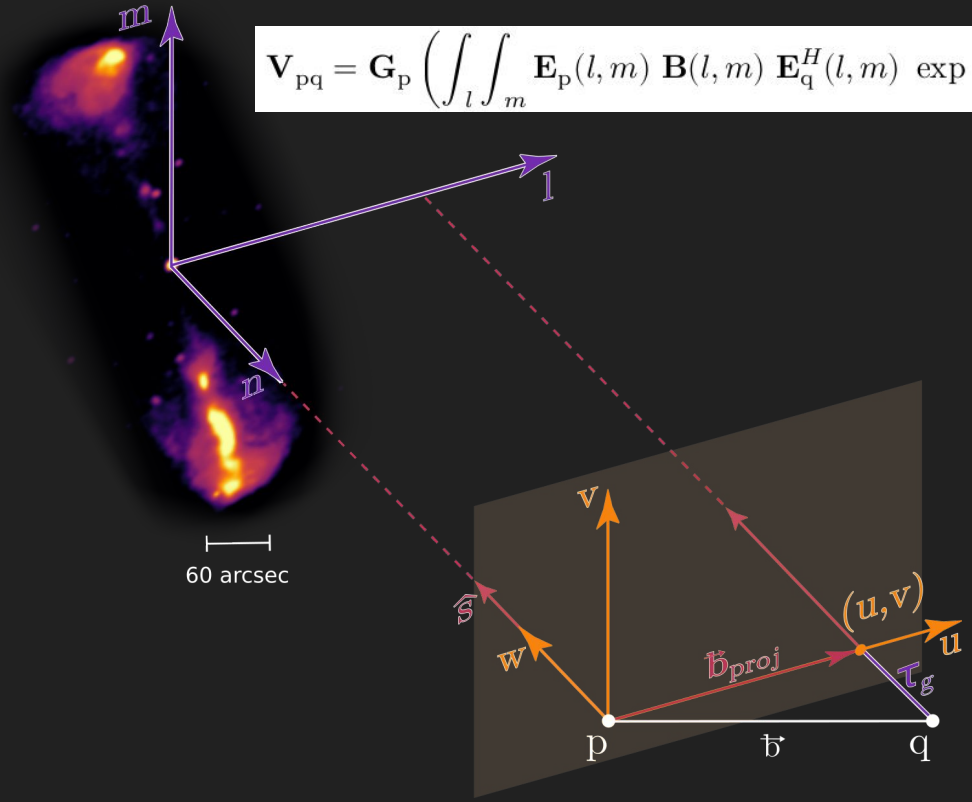

Flux density / mJy · px^{-1}
0
50
100
150
radionets
Pixels
Pixels
0
100
200
300
400
500

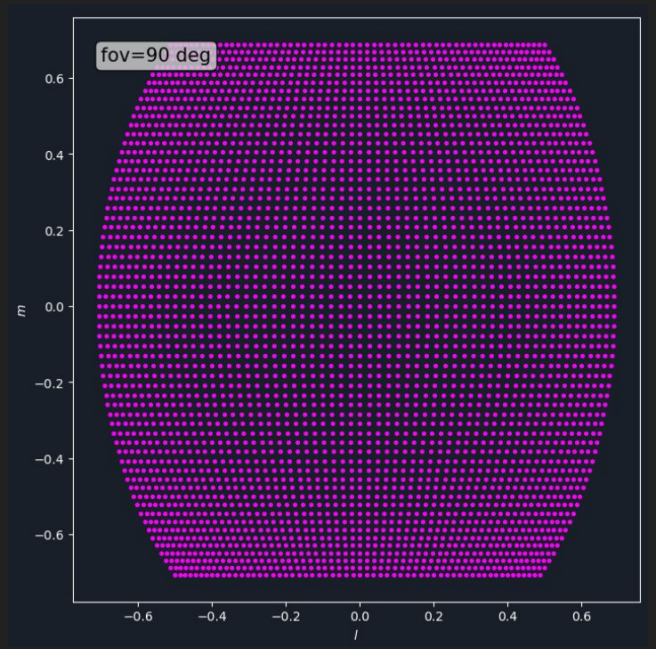

Flux density / mJy · px^{-1}
−50
0
50
100
Pixels
Pixels
0
100
200
300
400
500

BACKUP

Visibility simulations

$$\mathbf{V}_{\mathrm{pq}} = \mathbf{G}_{\mathrm{p}} \left(\int_l \int_m \mathbf{E}_{\mathrm{p}}(l,m)\ \mathbf{B}(l,m)\ \mathbf{E}_{\mathrm{q}}^{H}(l,m)\ \exp\left(-2\pi i(u_{\mathrm{pq}}l + v_{\mathrm{pq}}m)\right) \mathrm{d}l\mathrm{d}m \right) \mathbf{G}_{\mathrm{q}}^{H}$$





Custom Fourier kernel

Full control

Non-uniform coordinates included

Slow -> Speed-ups

Now:
≈ 1 M Vis. in 5 minutes

$$\mathbf{V}_{pq}(l, m) = \sum_{l,m} \mathbf{E}_p(l,m)\,\mathbf{K}_p(l,m)\,\mathbf{B}(l,m)\,\mathbf{K}_q^H(l,m)\,\mathbf{E}_q^H(l,m)$$

Source Distribution:

$$\mathbf{B}(l, m)$$

Phase Delay:

$$\mathbf{K}(l,m) = \exp(-2\pi \cdot i \cdot (ul + vm))$$

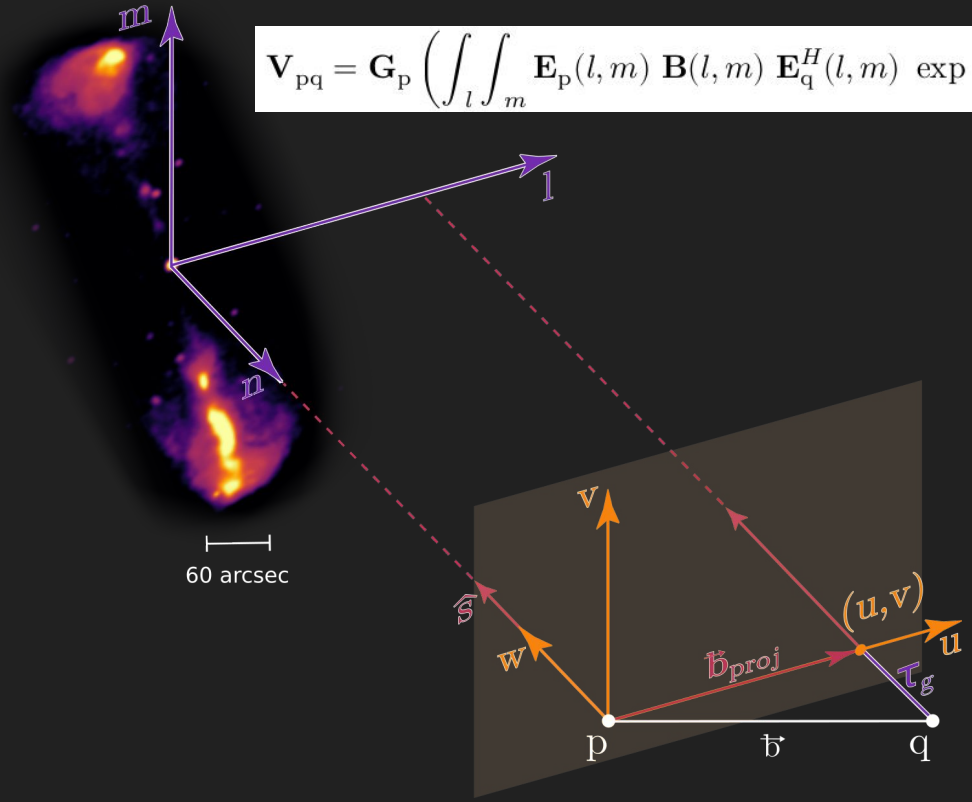
Telescope Beam:

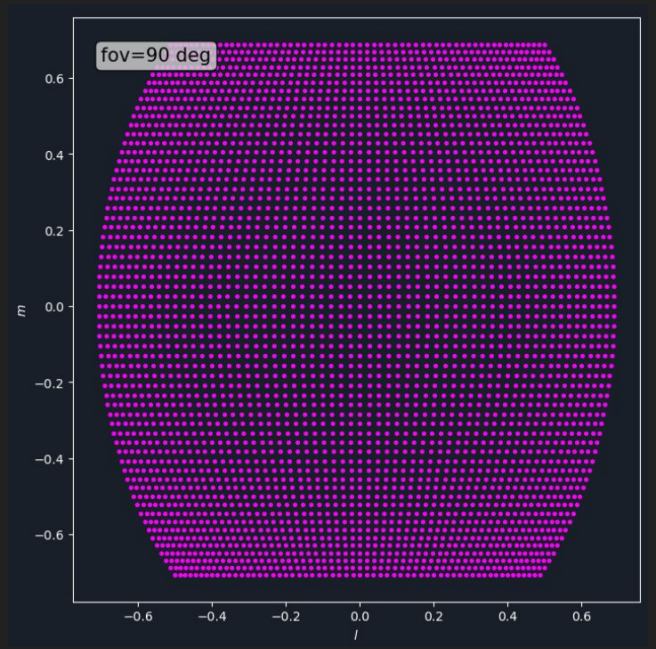
$$\mathbf{E}(l,m) = \mathrm{jinc}\left(\frac{2\pi}{\lambda_{obs}} d \cdot \theta_{lm}\right)$$

$$\mathrm{jinc}(x) = \frac{J_1(x)}{x}$$

Visibility simulations

$$\mathbf{V}_{\mathrm{pq}} = \mathbf{G}_{\mathrm{p}} \left(\int_l \int_m \mathbf{E}_{\mathrm{p}}(l,m)\ \mathbf{B}(l,m)\ \mathbf{E}_{\mathrm{q}}^{H}(l,m)\ \exp\left(-2\pi i(u_{\mathrm{pq}}l + v_{\mathrm{pq}}m)\right) \mathrm{d}l\mathrm{d}m \right) \mathbf{G}_{\mathrm{q}}^{H}$$

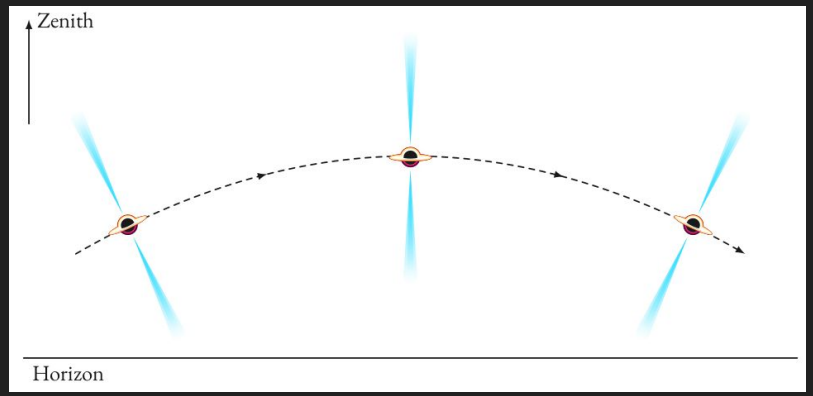




pyvisgen

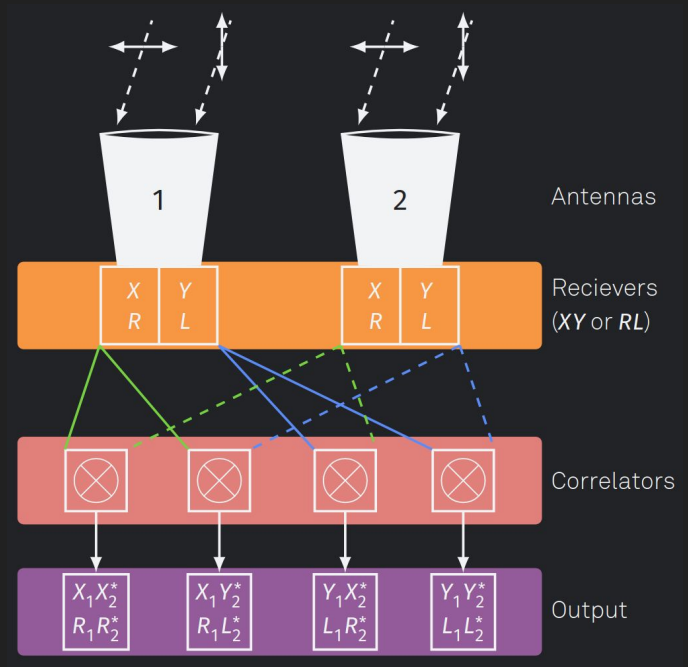
Polarization:
- linear feed
- circular feed

Parallactic angle rotation

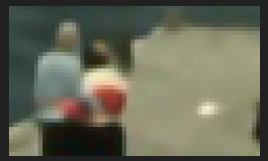


Simulations

PYVISGEN
Visibility Simulations in Python



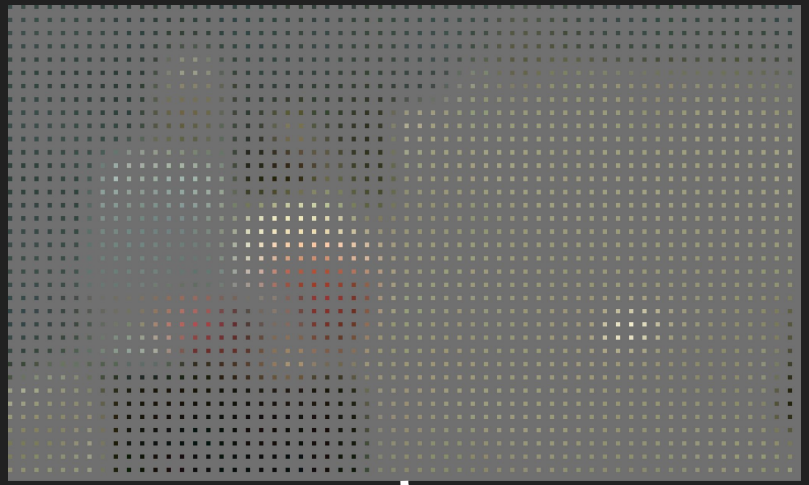
Deep learning model: image inpainting



Zoom

(He et al. 2015; Gross & Wilber 2016)

Deep learning model: SRResNet



SRResNet (He et al. 2015; Gross & Wilber 2016)

Deep learning model: Convolutional inpainting