Imaging Young Stellar Objects with Optical Interferometry and Normalizing Flows

Prof. Fabien Baron, Physics & Astronomy, Georgia State University

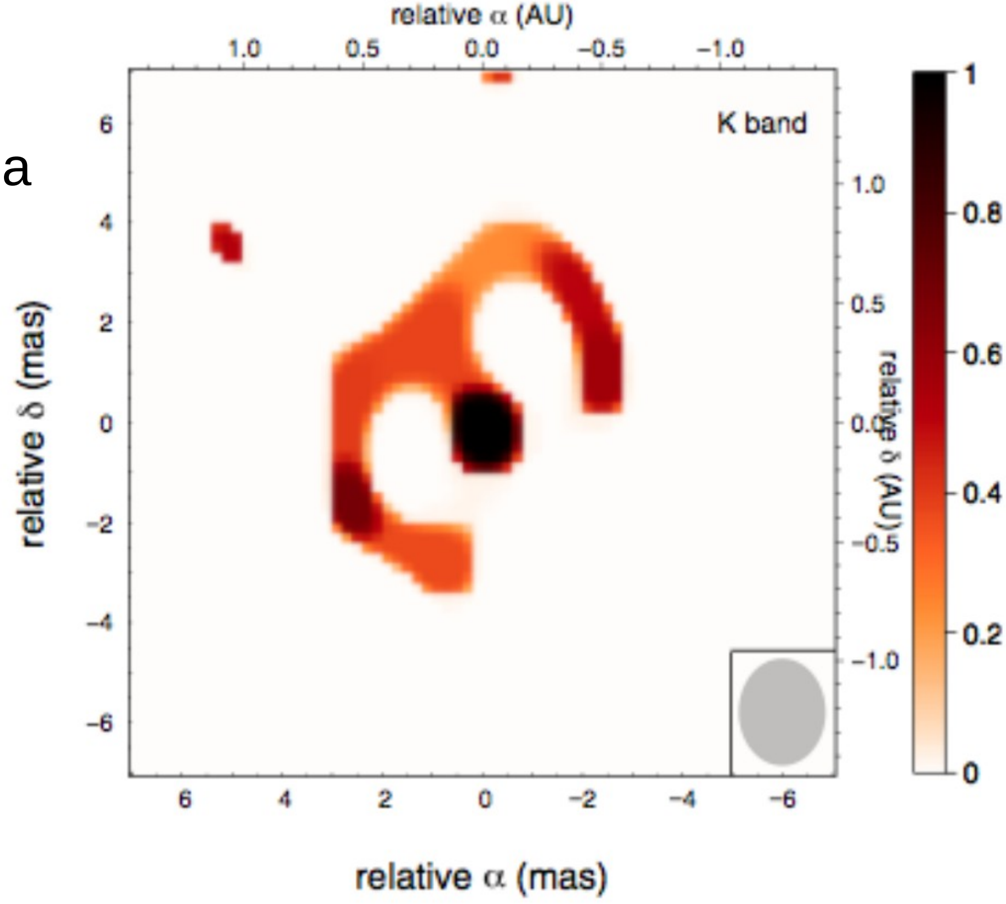
With thanks to:

GRA Rafael Orozco
Seismic Laboratory for Imaging and Modeling, Georgia Institute of Technology

Prof. Katherine L. Bouman & GRA Berthy Fang
Computing & Mathematical Sciences Department, CalTech

Imaging Disks around Young Stars

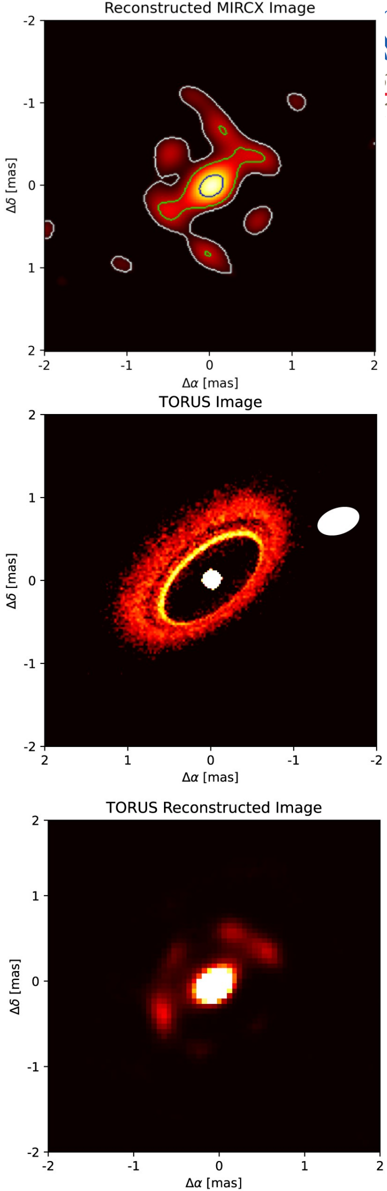
- Right figure = first interferometric image of a YSO (Benisty et al. 2011)
- Regularized via total variation
  - Over-regularized disc
  - Compact artifacts beyond the disk
  - Central star bleeding onto the disk



Benisty et al. 2011

The YSO imaging problem

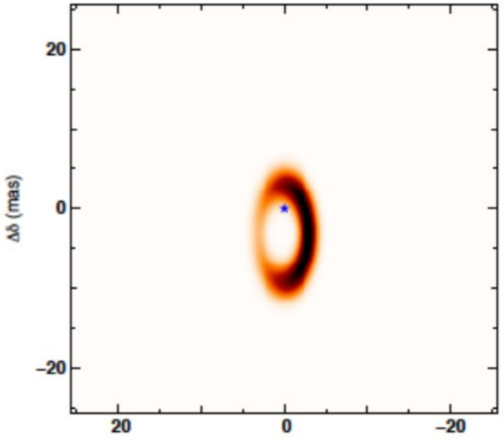
- Right: SU Aur from Labdon et al., 2023, *Imaging the warped dusty disk wind environment of SU Aurigae with MIRC-X.* Top: reconstructed image, middle TORUS model, bottom: best fit TORUS model.
- Dynamic environment on unknown timescales
- CHARA data repeatability issues?
- Polychromatic star and environment
- across visible, near IR, mid-IR + lines
- Theoretical models (e.g. TORUS) doesn't always mesh well with observations
- Issues linked to data calibration, systematic errors (→ implies automatically variational inference?)



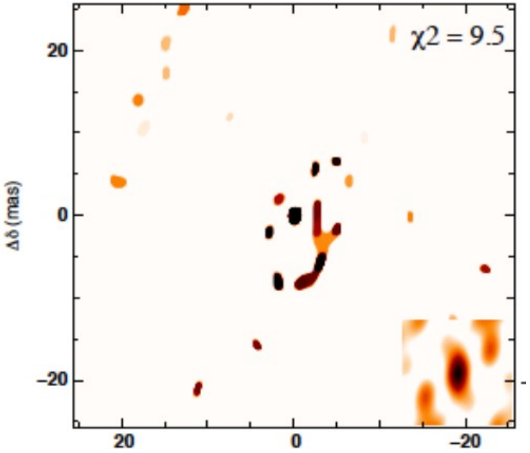
SPARCO

Kluska et al. 2014

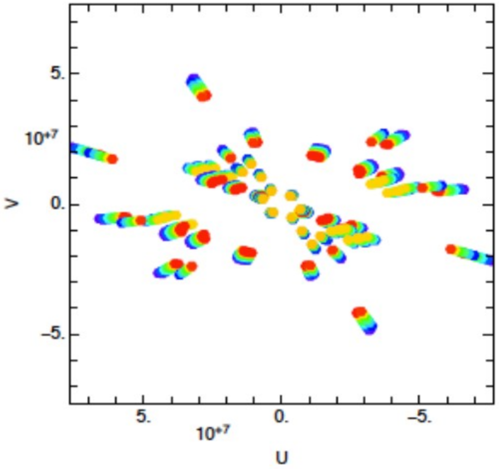
Model image



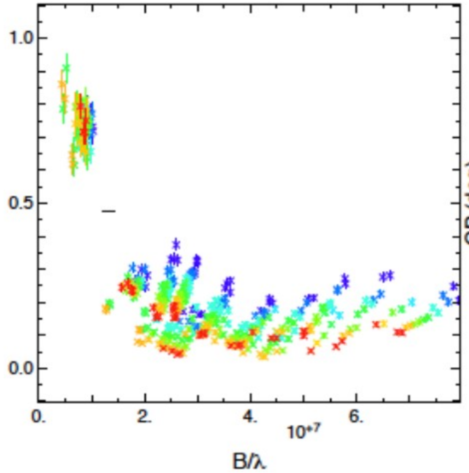
Monochromatic MiRA image reconstruction



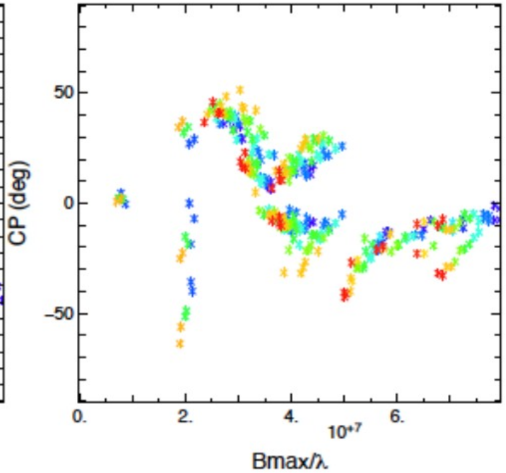
uv-plane



Simulated V2 (high S/N)

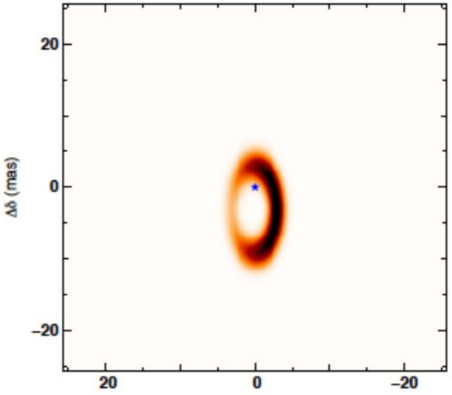


Simulated CP (high S/N)

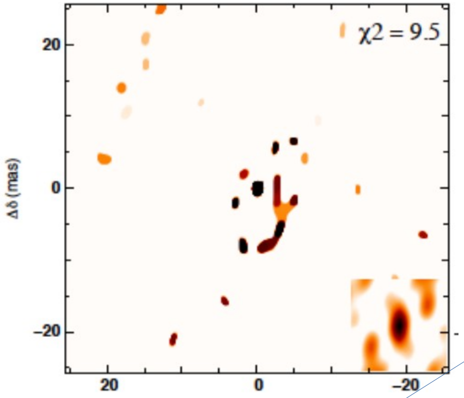


SPARCO

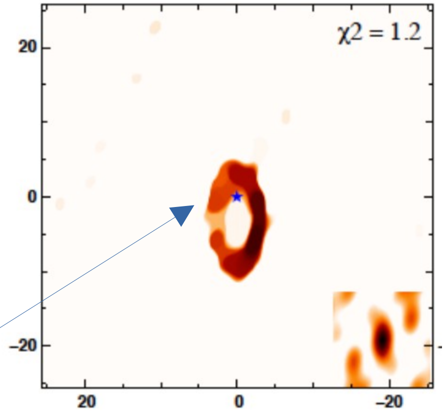
Model image



Monochromatic MiRA image reconstruction



MiRA/SPARCO image reconstruction



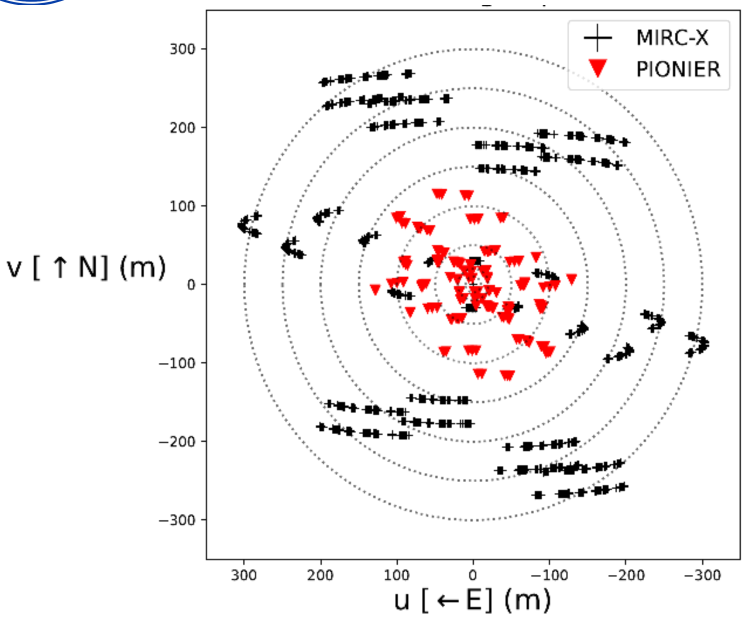
$$\widetilde{V}_{\rm tot}\left(\frac{\boldsymbol{b}}{\lambda},\lambda\right) = \frac{\boxed{f_*^0\left(\frac{\lambda}{\lambda_0}\right)^{-4} + \left(1-f_*^0\right)\left(\frac{\lambda}{\lambda_0}\right)^{d_{\rm env}}\boxed{\widetilde{V}_{\rm env}\left(\frac{\boldsymbol{b}}{\lambda}\right)}}}{\boxed{f_*^0\left(\frac{\lambda}{\lambda_0}\right)^{-4} + \left(1-f_*^0\right)\left(\frac{\lambda}{\lambda_0}\right)^{d_{\rm env}}}}.$$

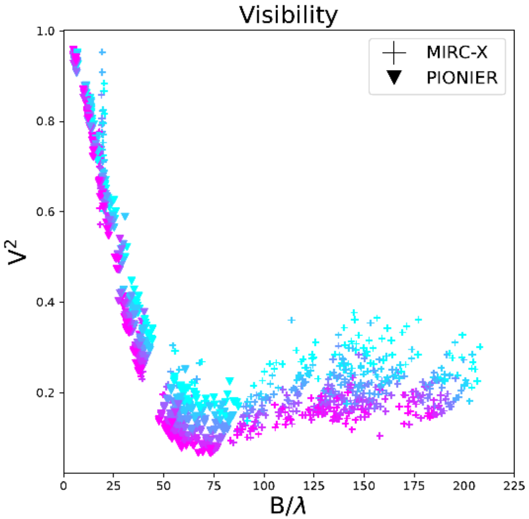
Unnormalized visibilities of point source and environment (= image). Both weighted by respective fluxes.

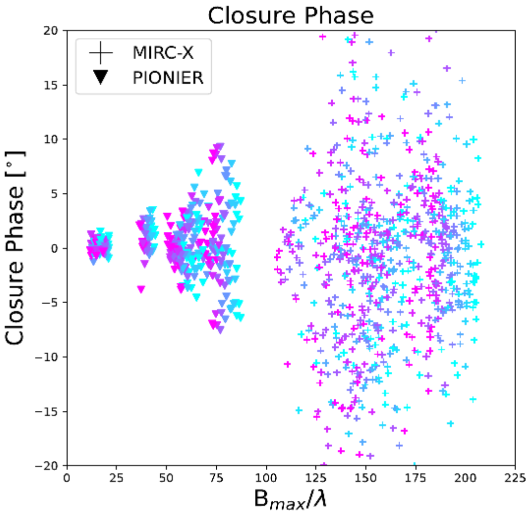
Flux normalization

Analytic model of a polychromatic point source (power law -4) + single image with spectral power law d_{env}

Ibrahim et al. 2023: data







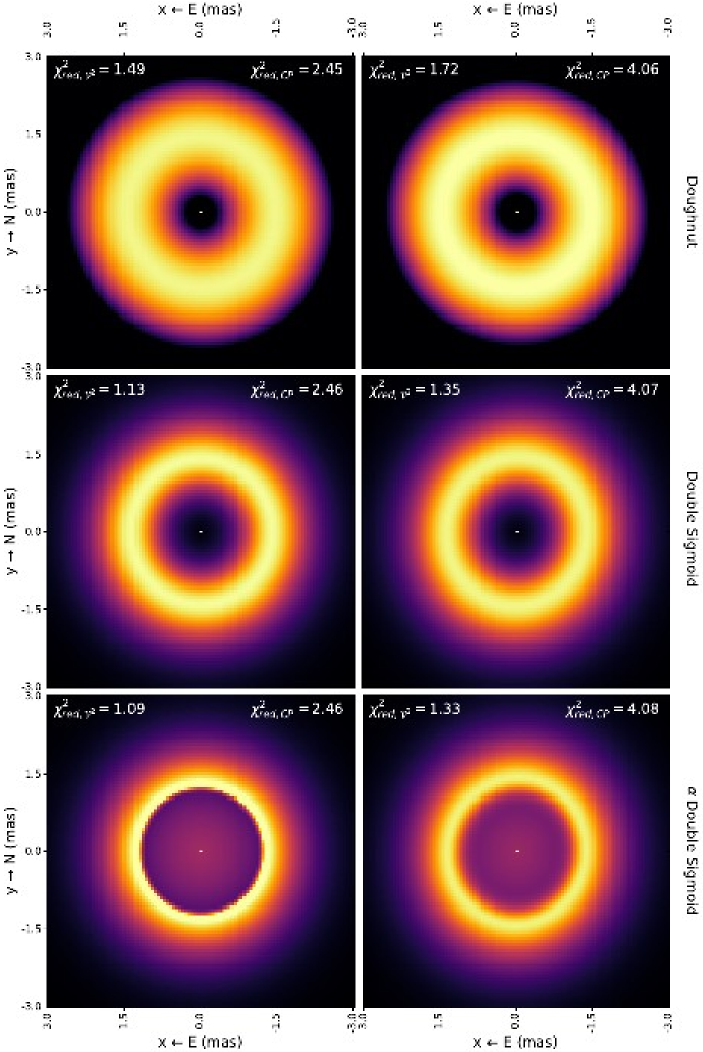
Epoch A data

Property	v1295 Aql
α (J2000)	$20^h03^m02^s.51$
δ (J2000)	+5°44'16".66
Spectral Type	B9
T_{eff}	9750 ±125K
R_*	9.68 ±0.44$R_\odot$
Distance	847.9±22 pc
$\text{Log}(L_*)$ $L_\odot$	2.88 ± 0.03
Mass	6.0 ±0.2 $M_\odot$
Vmag	7.79±0.06
Hmag	6.61±0.07

Table 1. V1295 Aql Stellar properties from Guzmán-Díaz et al. (2021)

Imaging the Inner Astronomical Unit of the Herbig Be Star HD 190073, Ibrahim et al. 2023

Ibrahim et al. 2023: modeling



$$f(r) = 1 - \left(\frac{r - \bar{r}}{2(r_{max} - r_{min})}\right)^2$$

$$f(r) = \frac{1}{1 + e^{\frac{-(r - R_{in})}{\sigma_{in}}}} * \frac{1}{1 + e^{\frac{(r - R_{out})}{\sigma_{out}}}}$$

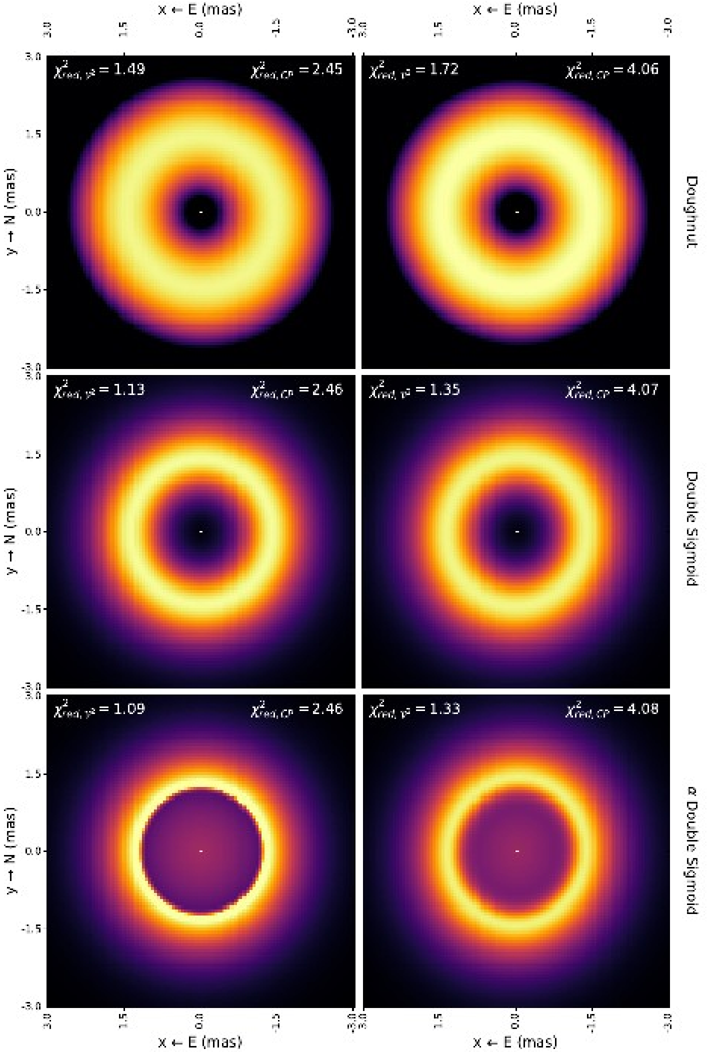
$$f(r) = \left(\alpha + \frac{1 - \alpha}{1 + e^{\frac{-(r - R_{in})}{\sigma_{in}}}}\right) * \frac{1}{1 + e^{\frac{(r - R_{out})}{\sigma_{out}}}}$$

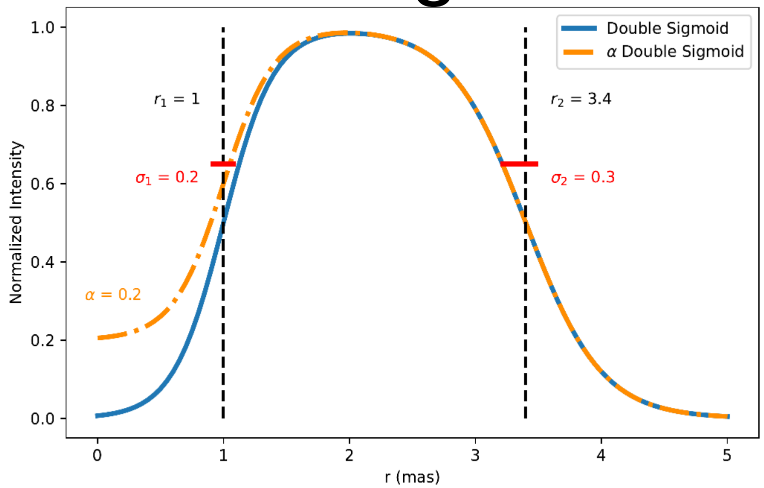
Disc almost face on

$$i \lesssim 20^\circ$$

Imaging the Inner Astronomical Unit of the Herbig Be Star HD 190073, Ibrahim et al. 2023

Ibrahim et al. 2023: modeling

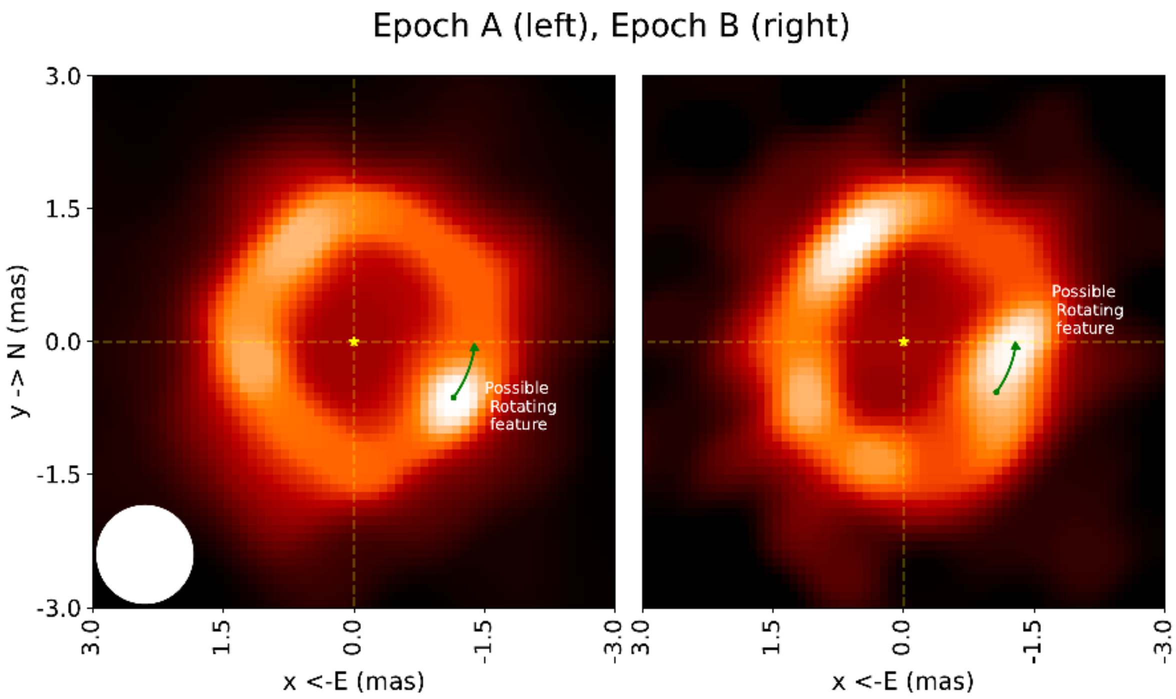




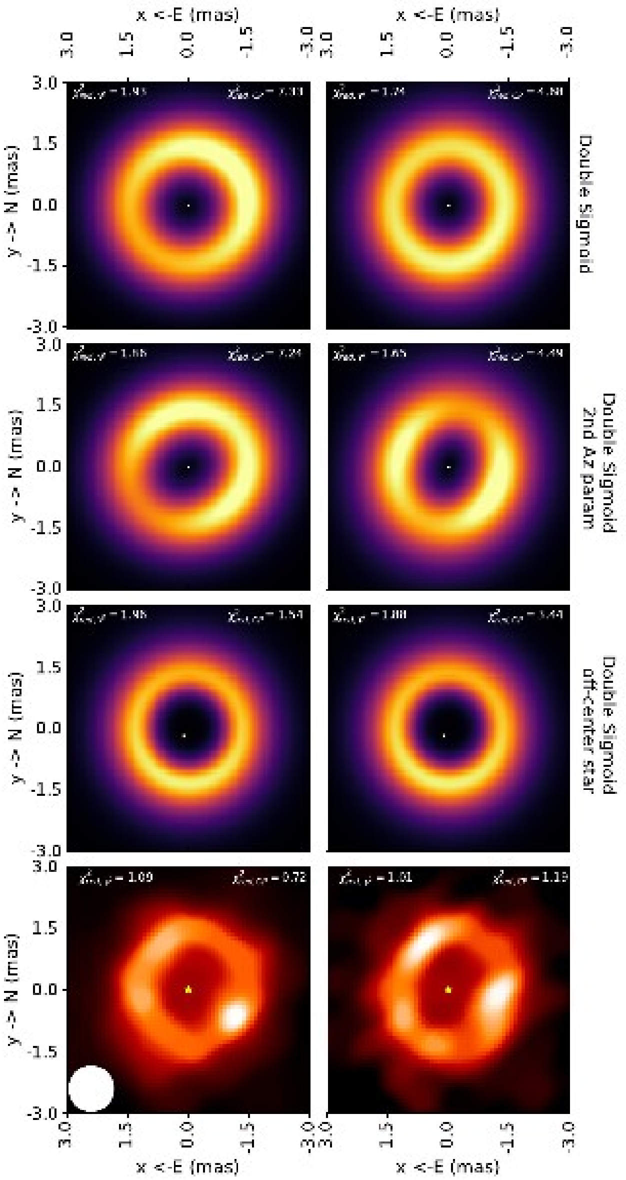
Model	epoch	incl. (deg)	PA (deg)	R_{in} (mas)	R_{out} (mas)	σ_{in} (mas)	σ_{out} (mas)	%f star	%f disk	%f halo	disk slope	halo slope	χ^2_{red}	α
Doughnut	A	10.94 ±2.58	42.30 ±14.1	0.29 ±0.01	2.60 ±0.02	-	-	43	51	6	-0.10 ±0.14	2.08 ±1.16	1.49	-
	B	10.05 ±2.99	55.4 ±17.9	0.35 ±0.01	2.65 ±0.02	-	-	42	49	9	-0.32 ±0.17	2.10 ±1.01	1.72	-
Double Sigmoid	A	10.14 ±2.95	3.4 ±14.8	1.259 ±0.040	1.252 ±0.047	0.262 ±0.020	0.426 ±0.016	42	54	4	-0.07 ±0.12	3.65 ±2.24	1.13	-
	B	18.12 ±1.75	9.58 ±5.71	1.276 ±0.038	1.268 ±0.046	0.270 ±0.019	0.452 ±0.018	42	54	4	-0.25 ±0.15	3.87 ±2.21	1.35	-
α Double Sigmoid	A	9.91 ±2.97	0.2 ±16.0	1.20 ±0.150	1.20 ±0.15	0.006 ±0.067	0.479 ±0.021	41	56	3	0.09 ±0.12	4.56 ±2.80	1.09	0.193 ±0.014
	B	19.07 ±1.61	11.35 ±5.12	1.251 ±0.018	1.243 ±0.039	0.103 ±0.042	0.497 ±0.017	41	55	4	-0.13 ±0.15	4.62 ±2.65	1.33	0.167 ±0.035

Imaging the Inner Astronomical Unit of the Herbig Be Star HD 190073, Ibrahim et al. 2023

Ibrahim et al. 2023: imaging



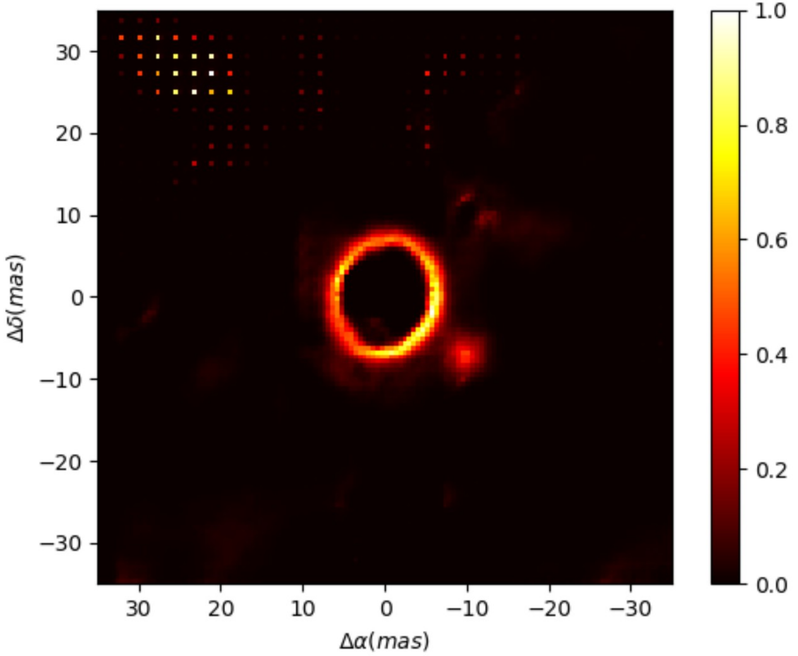
Separation ~32 days



- Reconstruction used SPARCO + radial regularization with auto-determined angles (based around spatial gradient sparsity ideas)

Normalizing flow networks: overview

- There is such a thing as "the probability of x being a YSO"
- To learn it we need some sort of general probability density approximator
- Possibilities include Generative Adversarial Networks, Variational Autoencoders, Normalizing flows, Diffusion models, etc.
- E.g. Claes et al. 2020 used GAN
- Problems with GAN:
  - mode collapse/training issues
  - no exact probability calculation



Claes et al. 2020

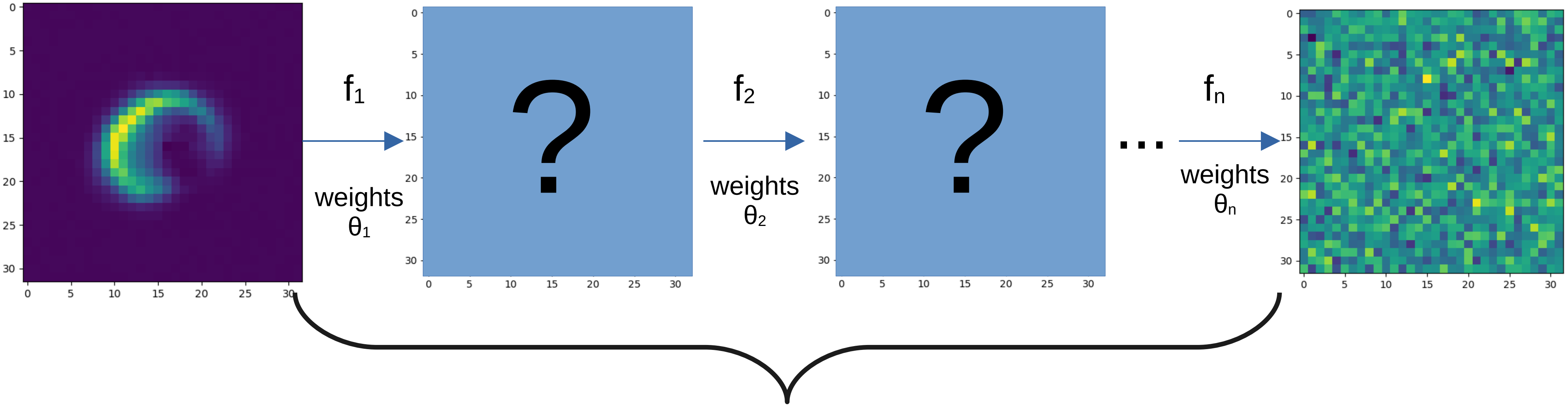
Normalizing flow networks: overview

- Normalizing flows are based on invertible transforms (“invertible networks”) with multiple advantages over GANs
  - efficient and exact probability calculation
  - fast sampling from the distribution as a generator
  - no mode collapse during training
  - speed and memory requirement low enough to work from laptop

Normalizing flows = invertible networks

Image variables x

“Latent” variables z
normally distributed



$$z = f_1 \circ f_2 \circ \cdots \circ f_n(x) = G_\theta(x)$$

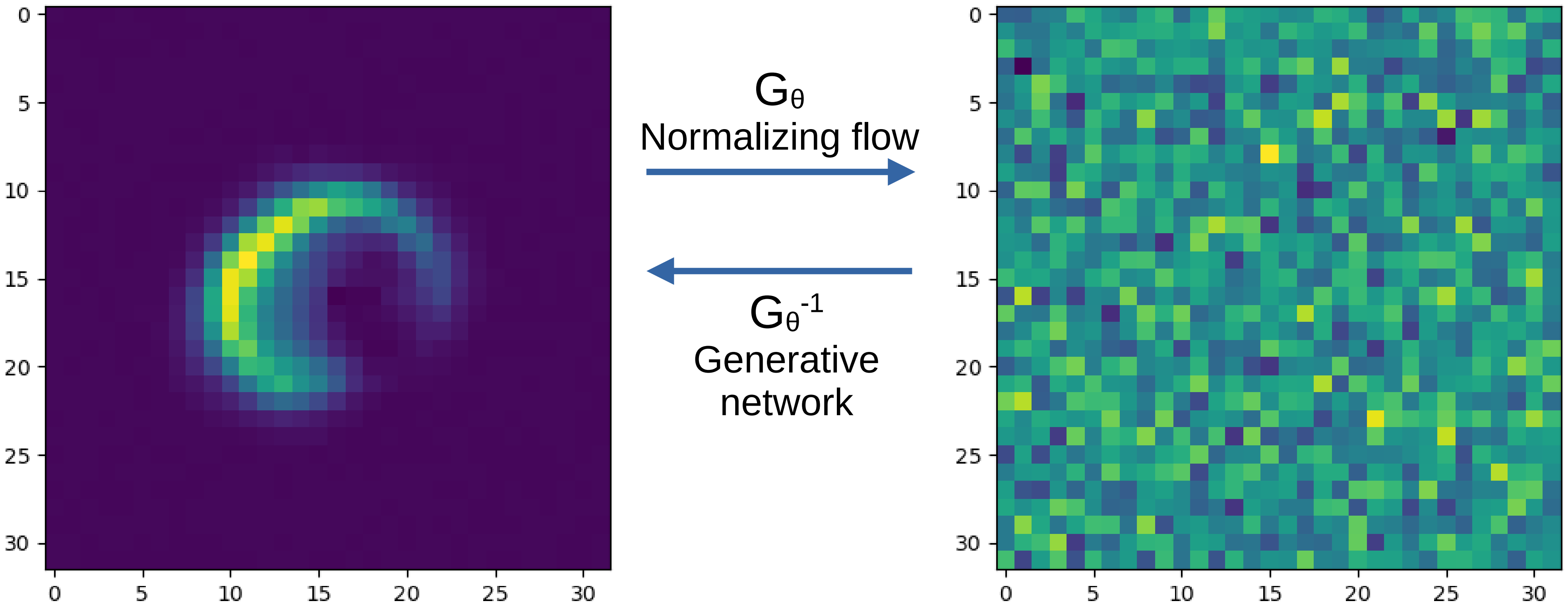
$f_1 \cdots f_n$ are bijective functions

Normalizing flows = invertible networks

$$x = f_n^{-1} \circ \cdots \circ f_2^{-1} \circ f_1^{-1}(z) = G_\theta^{-1}(z), z \sim \mathcal{N}(0,1)$$

Image variables x

Latent variables z



The probability of being a YSO is...

$$\log p_{\mathrm{YSO}}(x) = \log p(z) - \log\det\left|\frac{dG_\theta(z)}{dz}\right|$$

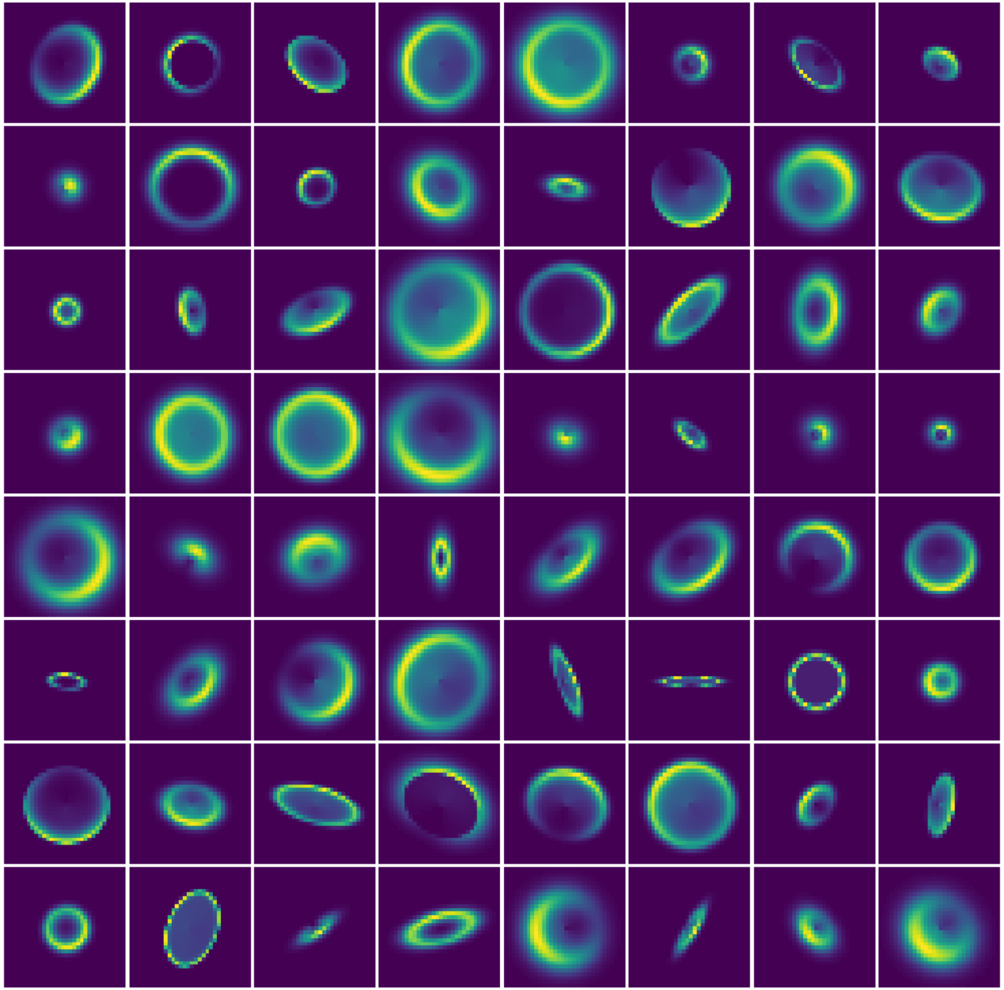
- The probability that x is a YSO is equal to that of z being Normal, minus the determinant of the network Jacobian
- This determinant can be calculated during the forward transform (x → z) or the inverse transform (z → x)
- Fast normalizing network iff efficient computation of the determinant (e.g. Glow, RealNVP networks)
- All you need now is to train the network to Normalize YSO images

Training set 1

Analytic “double sigmoid” models with a range of inclinations and sizes

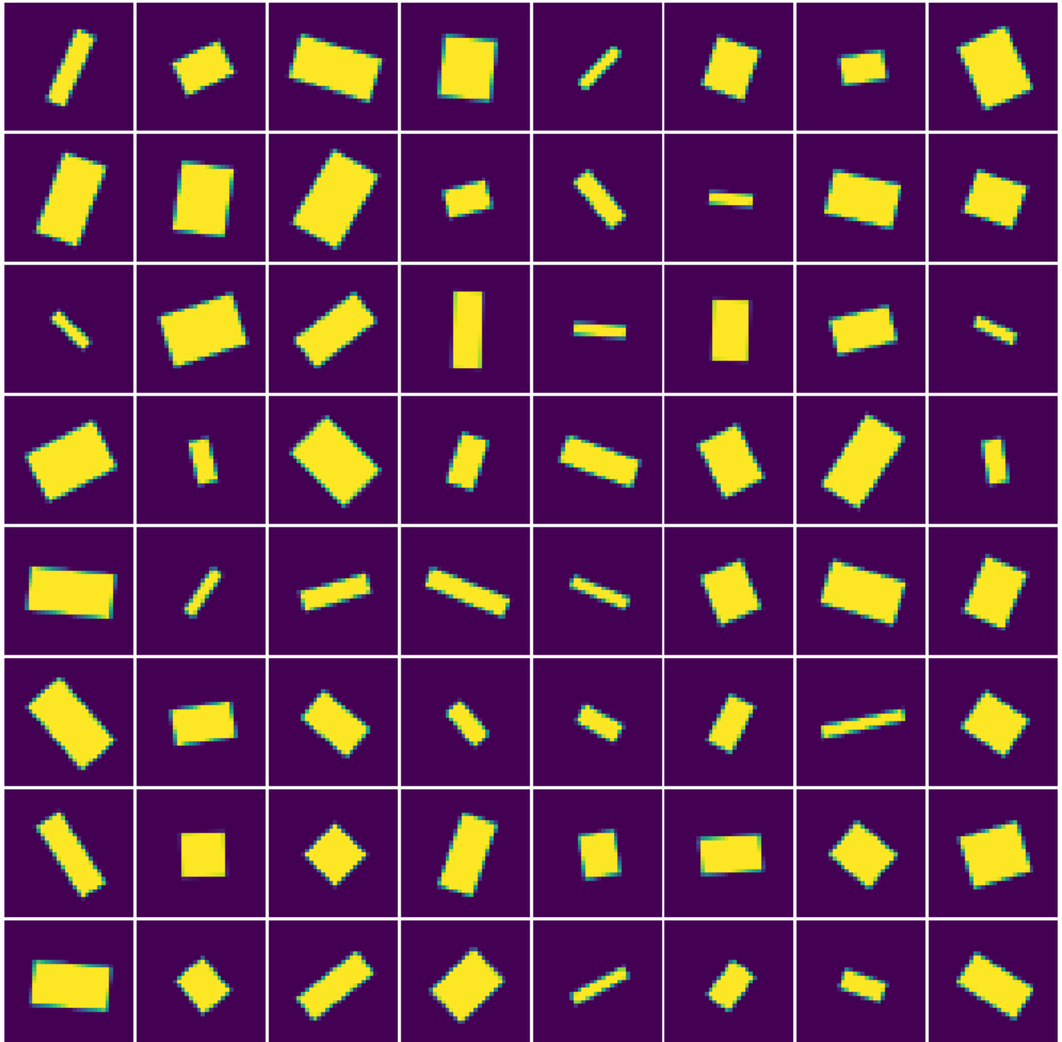
Filtered out images with low number of pixels

Here shown for the 32x32 pixel models

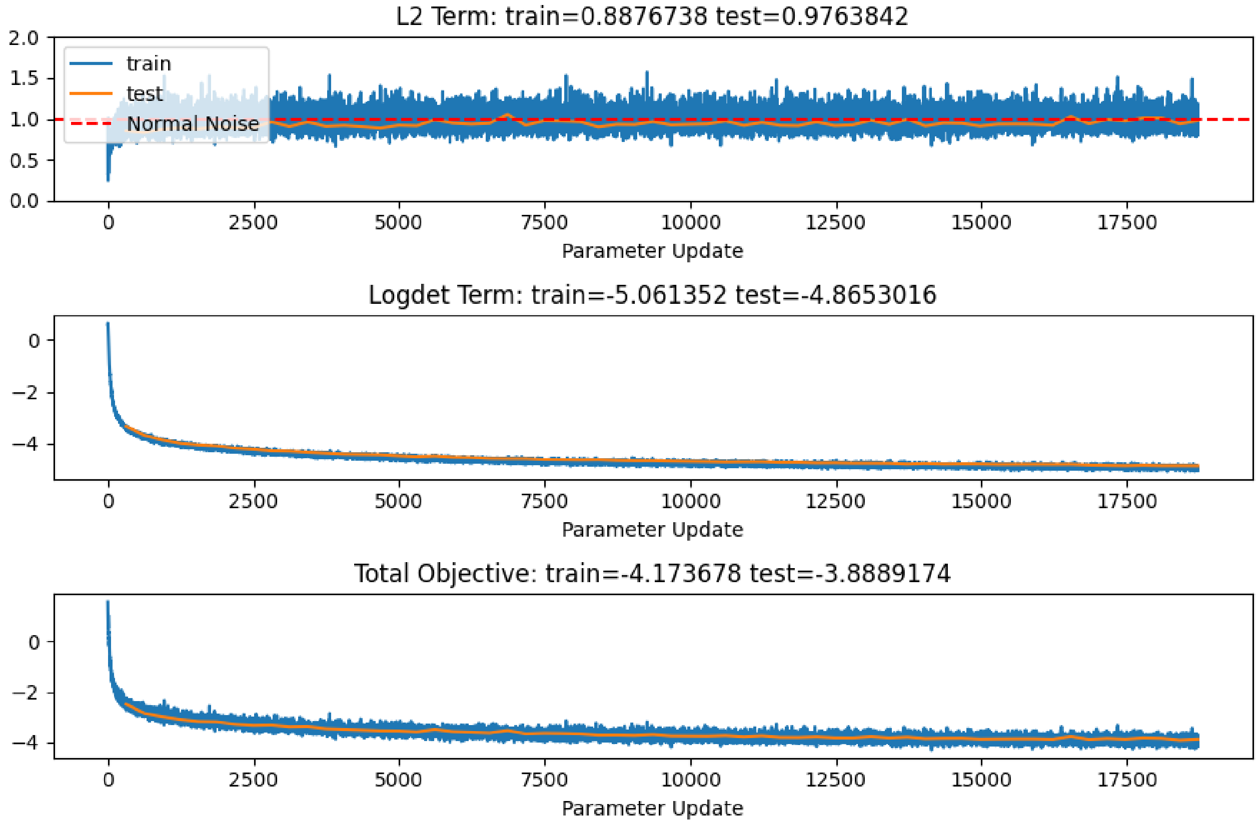


Training set 2

A super realistic model (this is what Fabien thinks YSO should look like?)



Training on YSO models



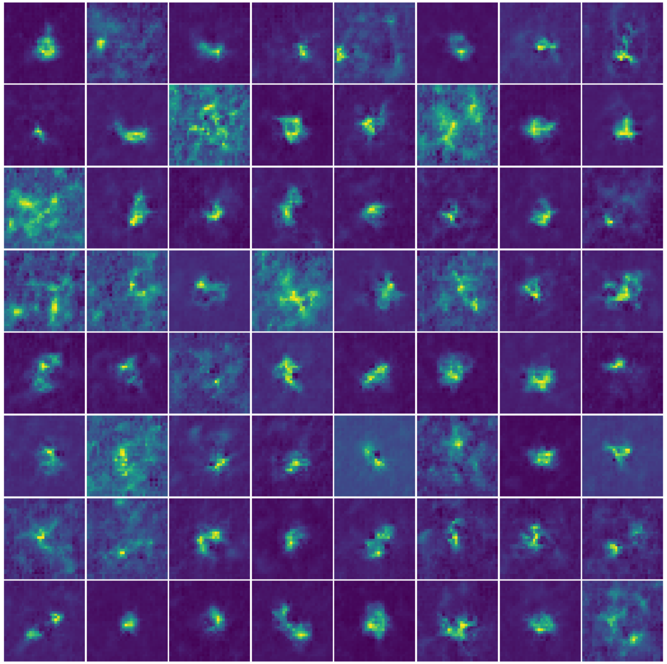
CPU training takes under 40 minutes with 32x32 pixel images, 50,000 training images

100 epochs, 128 mini-batches

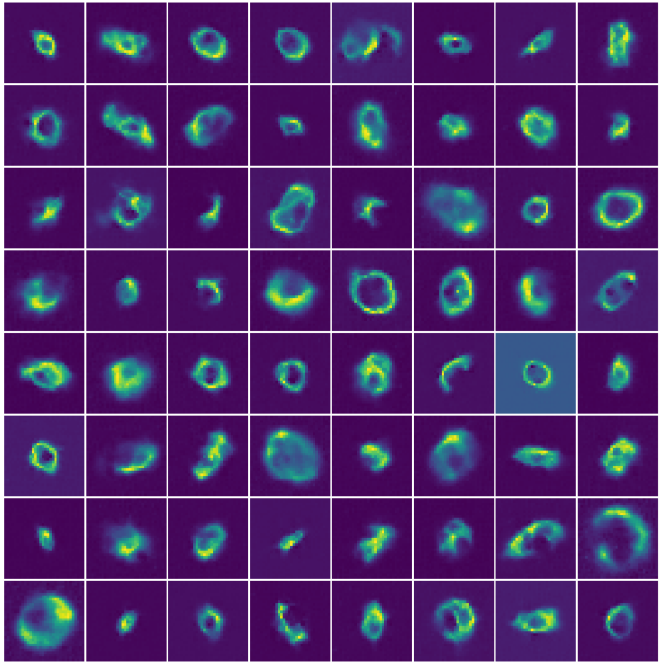
Glow model, 32 layers, trained with ADABelief (learning rate not optimized yet)

GPU implementation underway

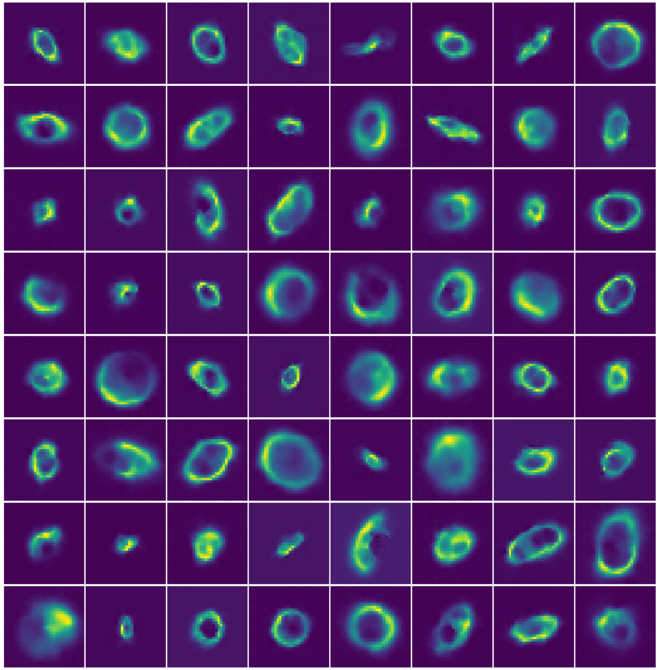
YSO training



Epoch 10

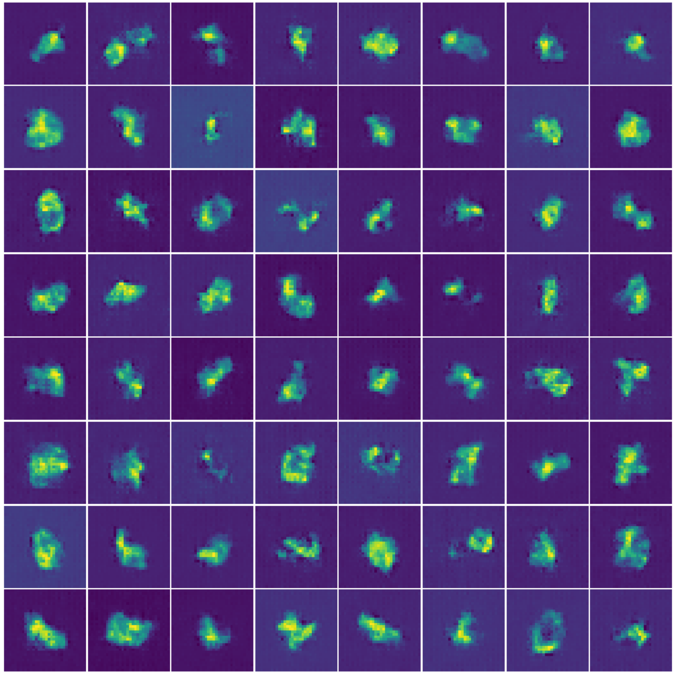


Epoch 20

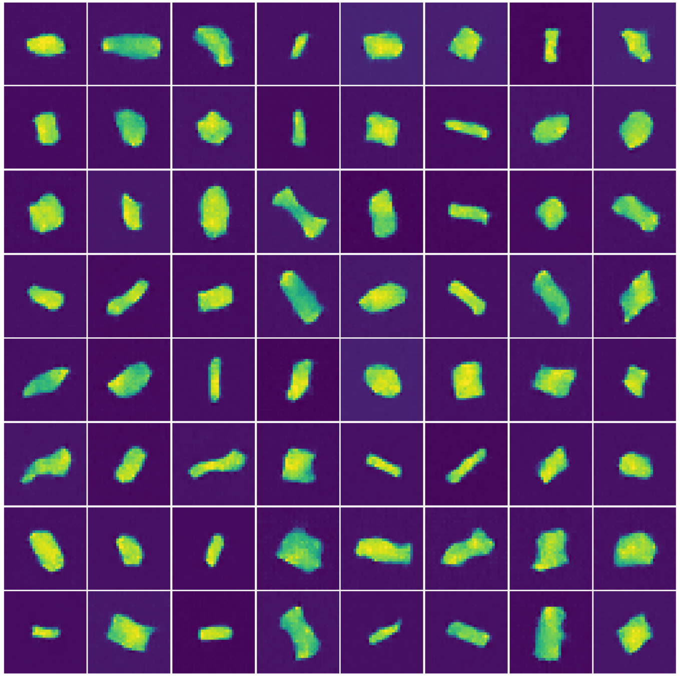


Epoch 100

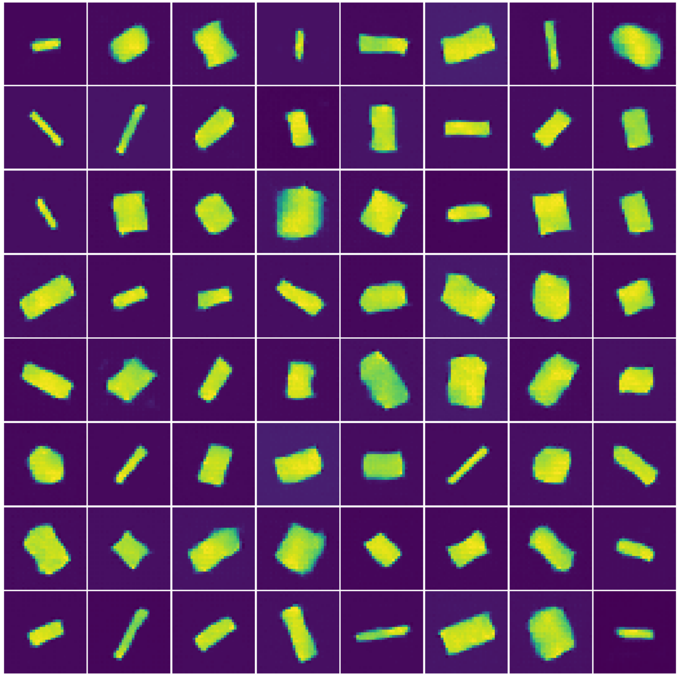
Rectangle training



Epoch 10

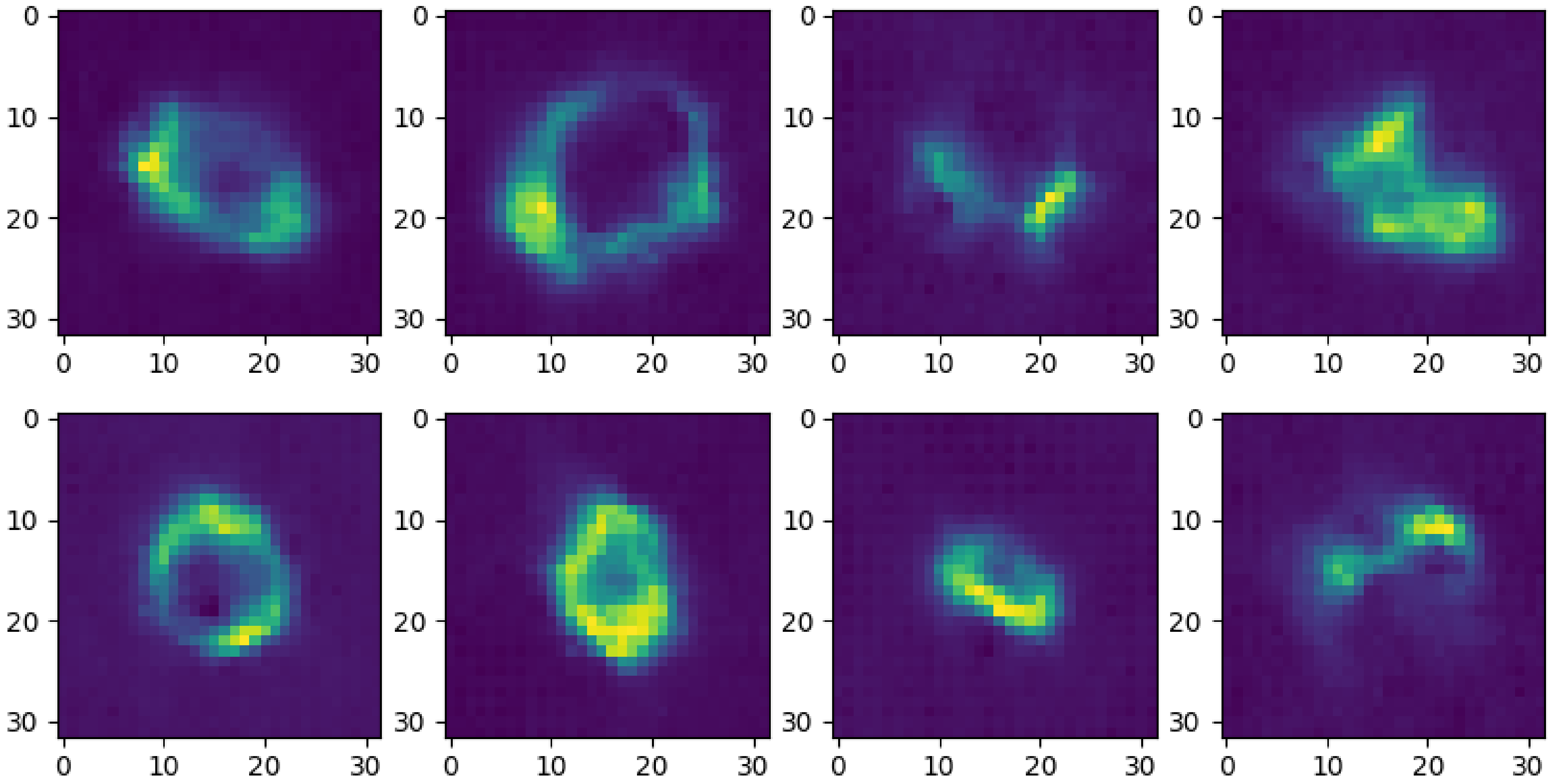


Epoch 20



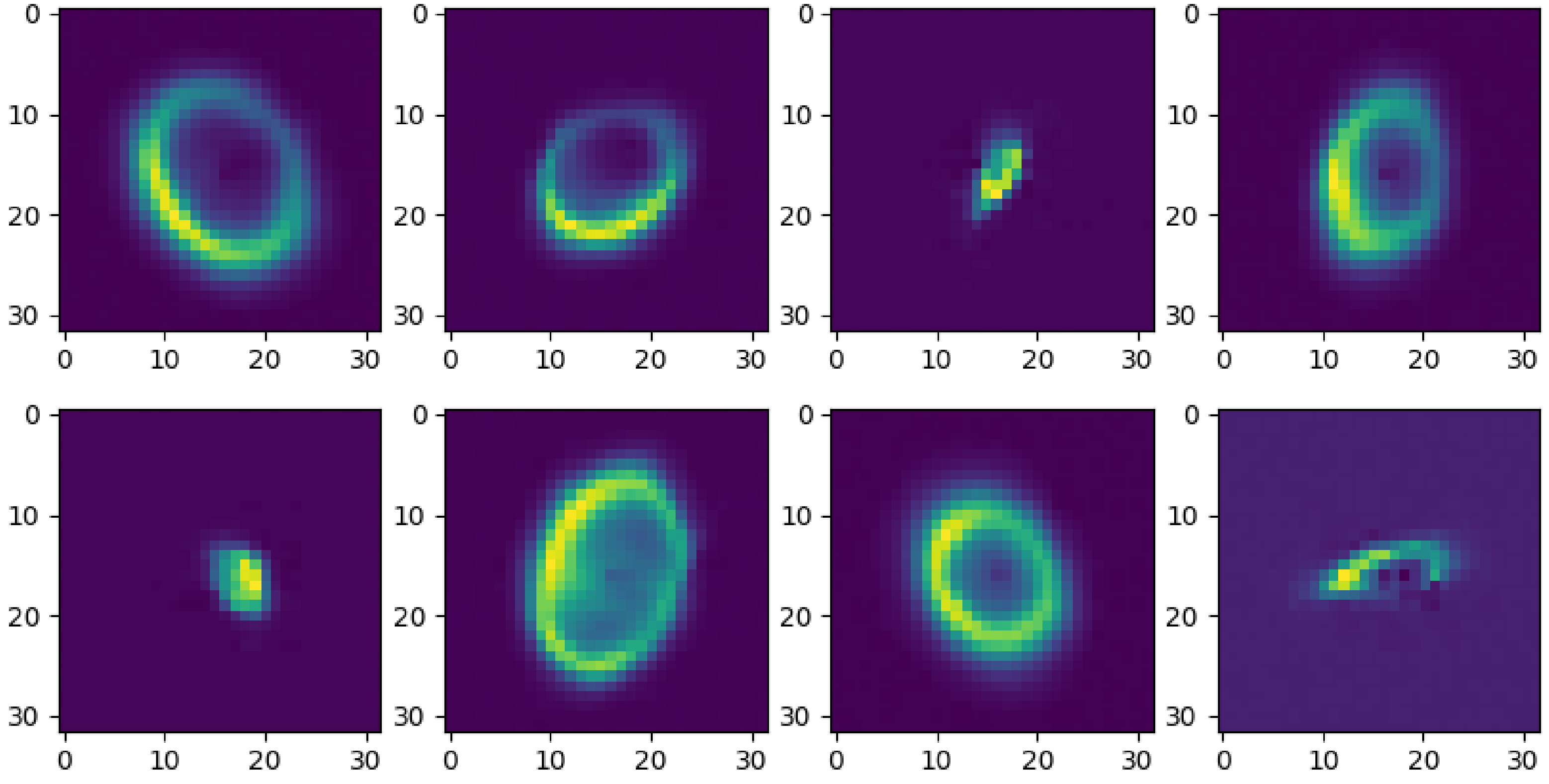
Epoch 100

Trained network used as generator



Epoch 10

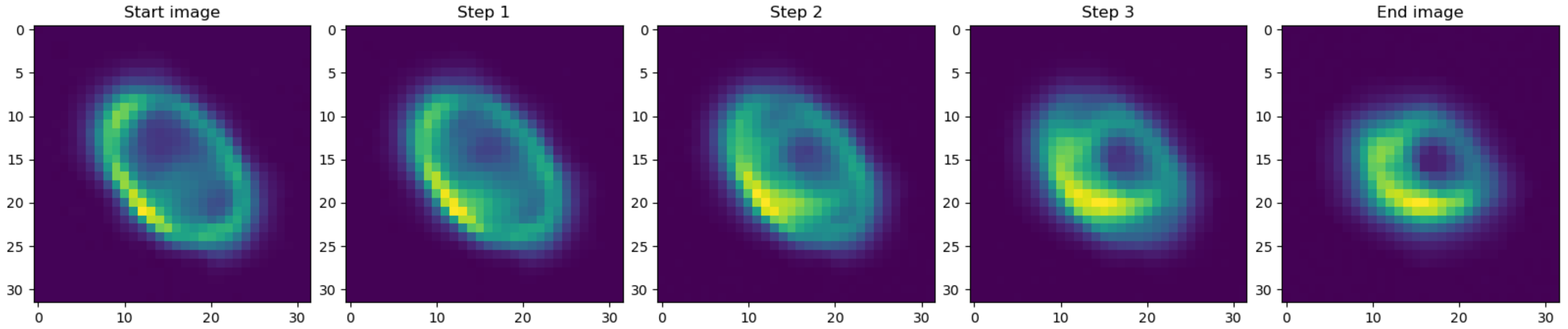
Trained network used as generator



Epoch 100

A fun example: YSO interpolation

Interpolation in image space



Interpolation in latent space

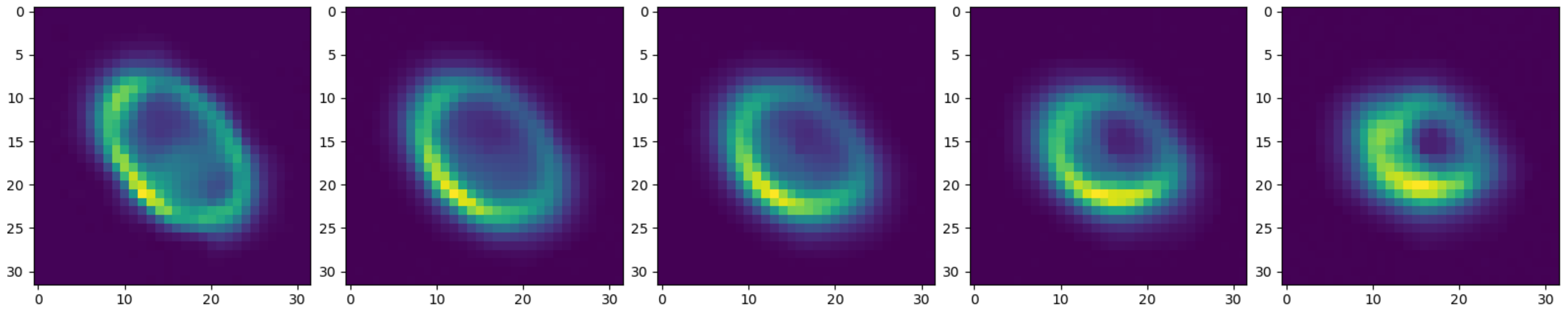


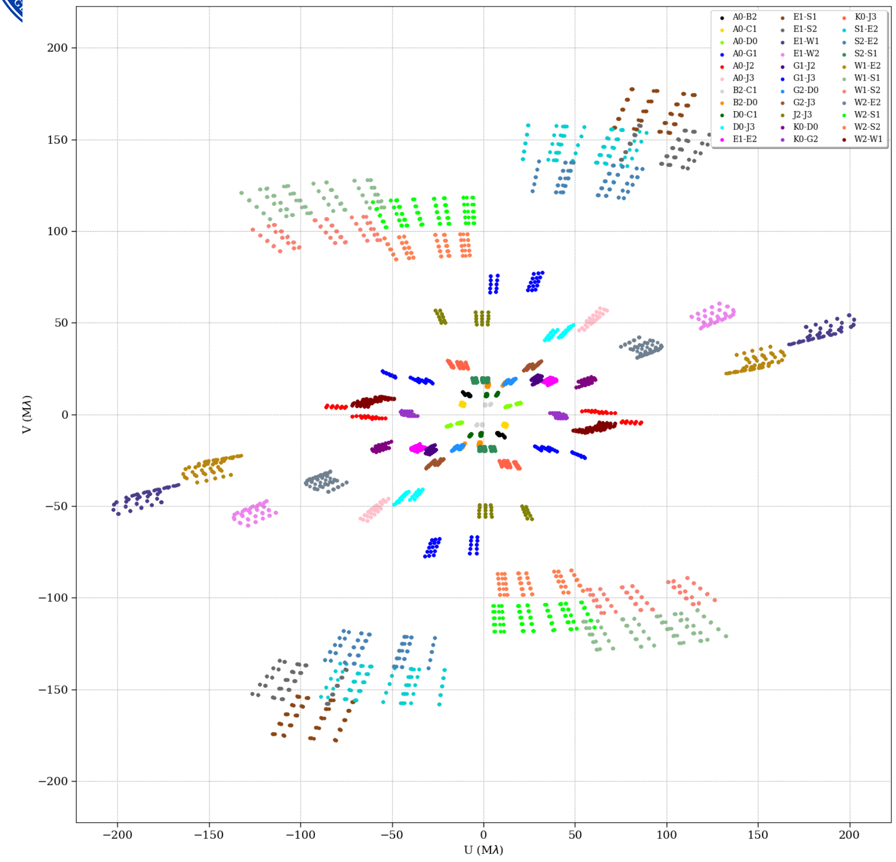
Image reconstruction with invertible networks

- Optimization in latent space

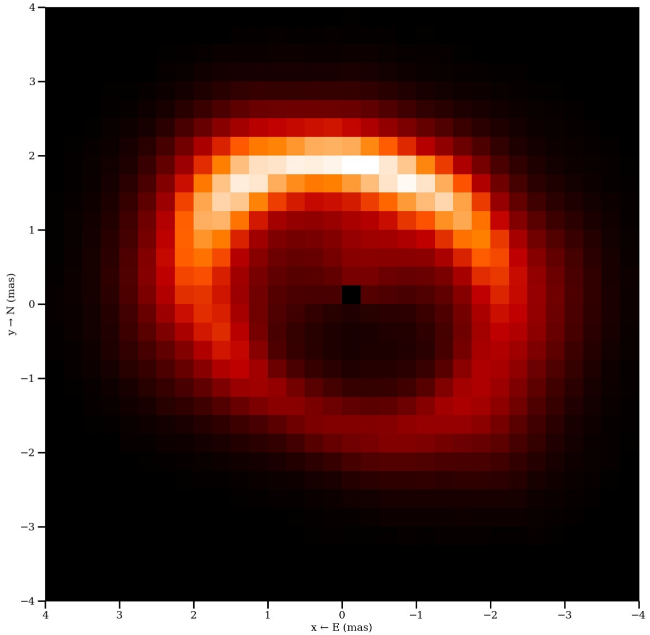
$$z_* = \operatorname{argmin} \mathcal{L}(G_\theta^{-1}(z)) + \frac{1}{2}||z||^2$$

- Gradient of inverse network can be computed by propagating $\nabla_x \mathcal{L}$ backward
- Using gradient descent
- Positivity imposed by `softplus()` activation of the last layer

Tests on simulations

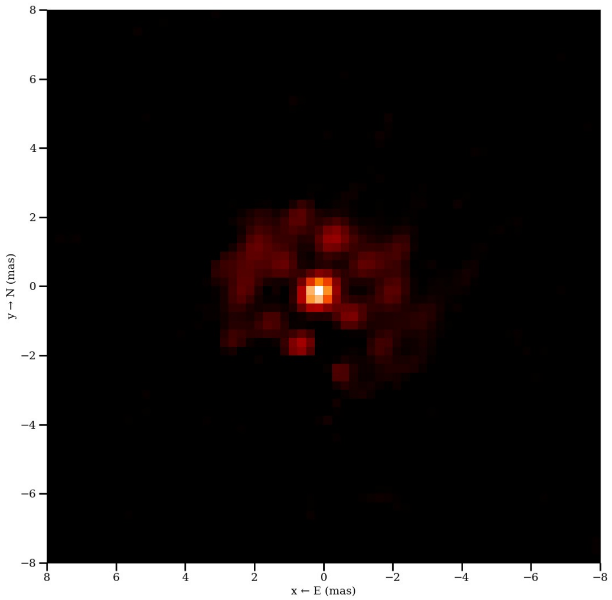


uv coverage and noise errors copied from real v1295 Aql data

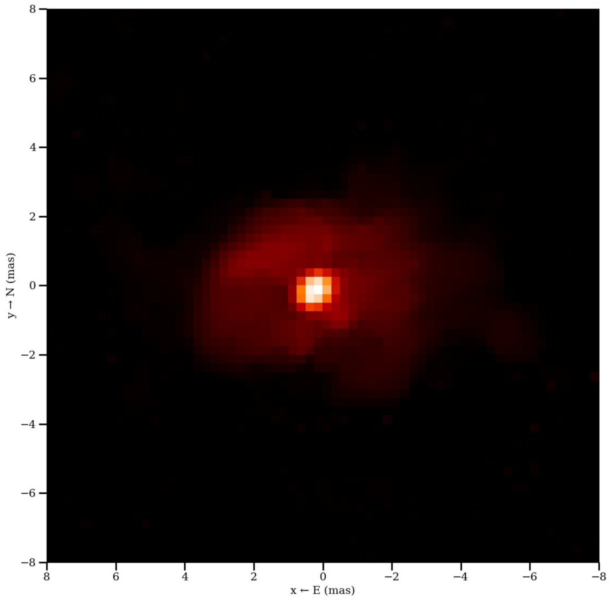


Ground truth object

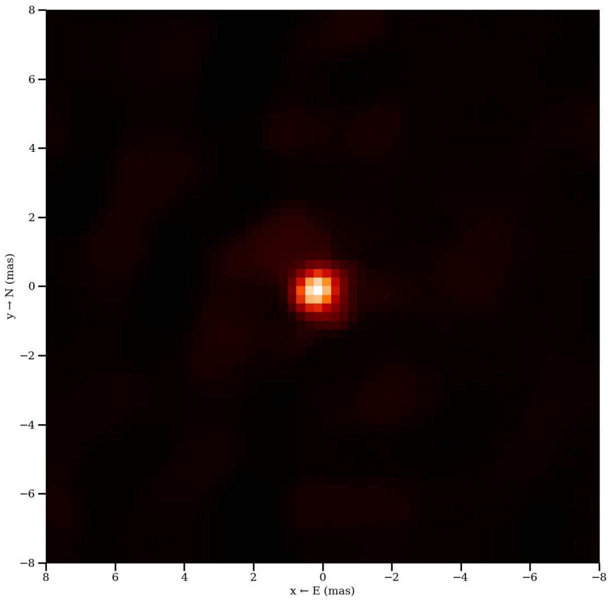
OITOOLS, no SPARCO, Total variation, fov 64x64



1e3 TV

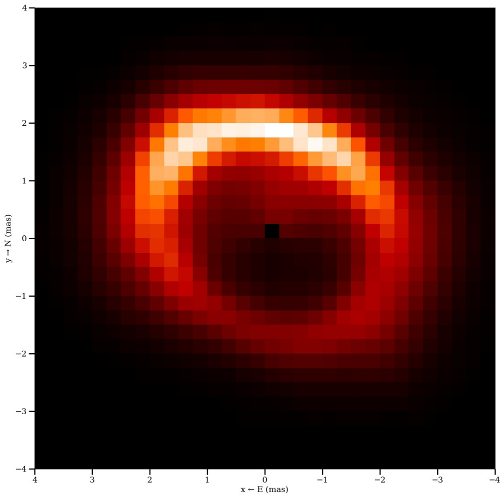


1e4 TV

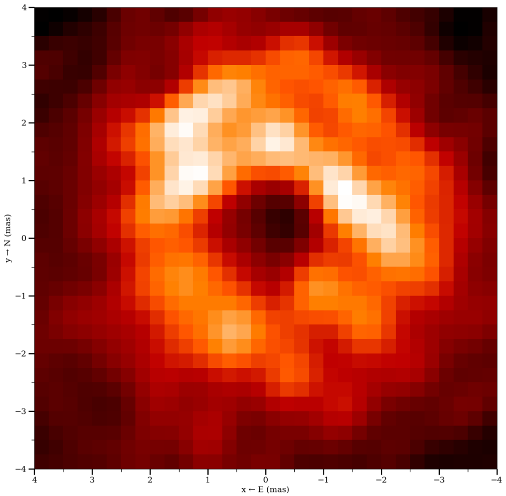


1e5 TV

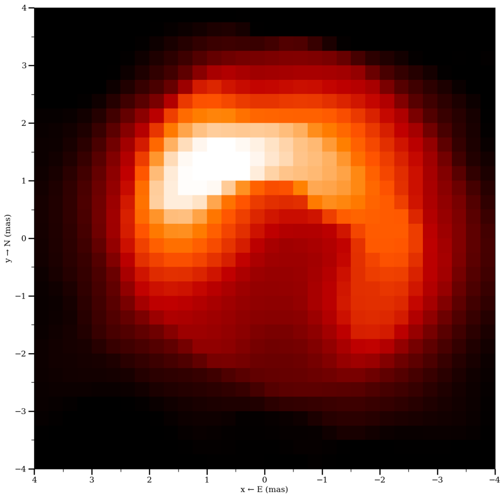
OITOOLS, SPARCO, Total variation, fov 32x32



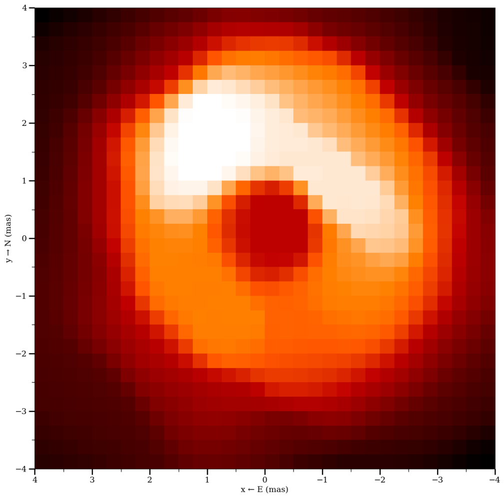
Ground truth



1e3 TV

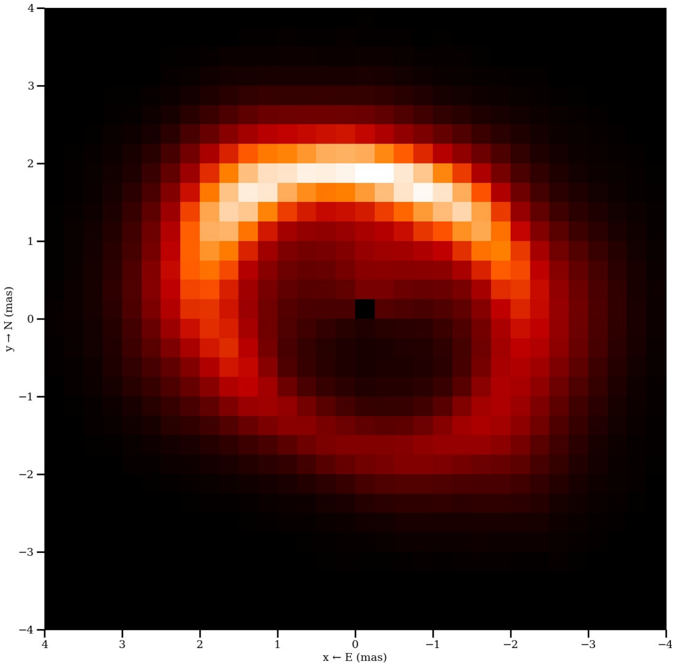


1e4 TV

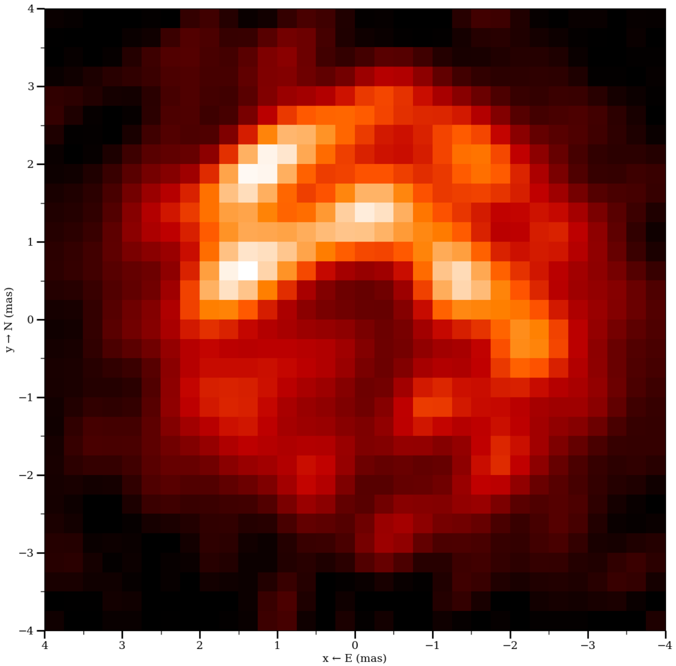


1e5 TV

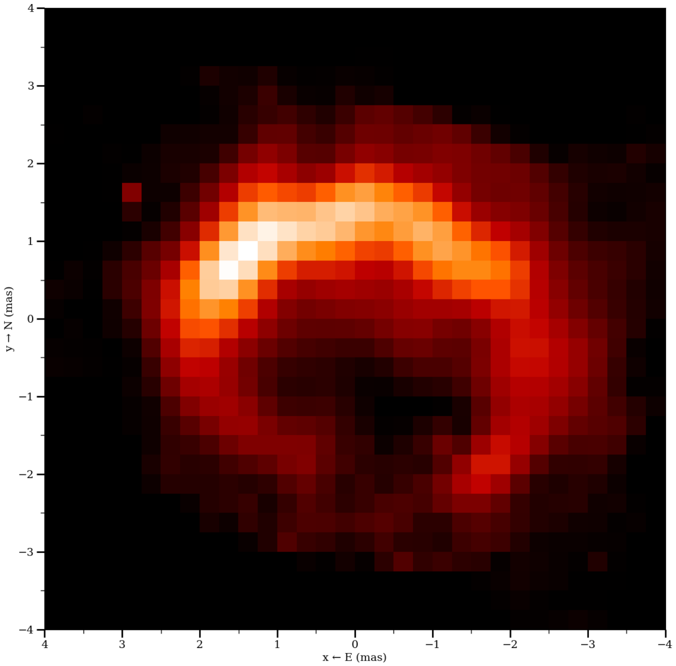
OITOOLS, with SPARCO, ℓ_1-ℓ_2, fov 32x32



Ground truth



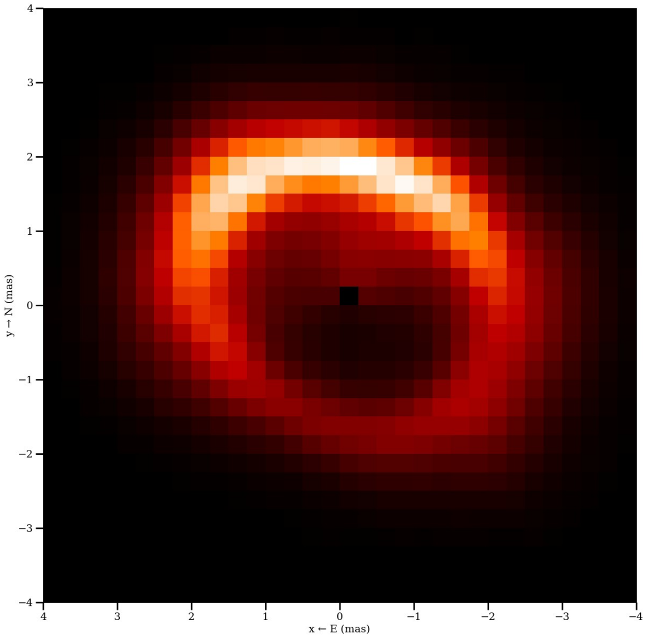
1e4 ℓ_1-ℓ_2



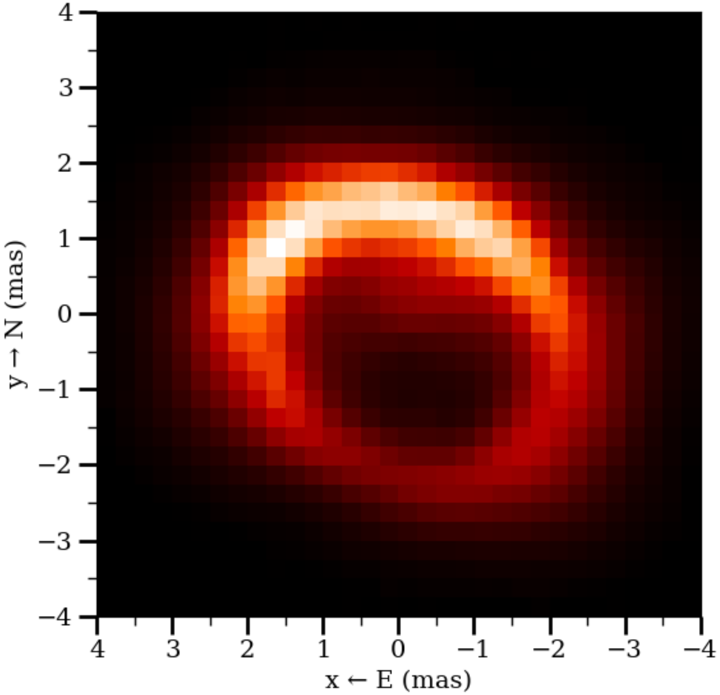
1e5 ℓ_1-ℓ_2

ℓ_1-ℓ_2 regularization is an “edge-preserving” regularization with smooth transition from ℓ_1 to ℓ_2 penalty on spatial gradient.

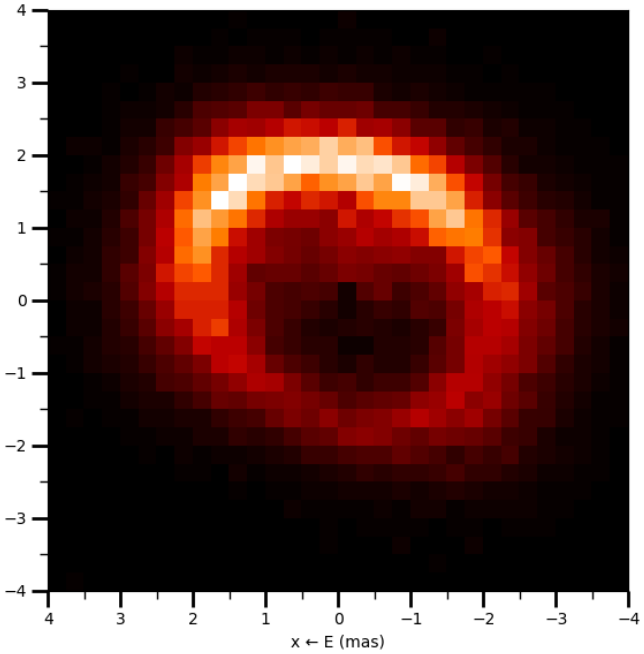
OITOOLS MAP with Normalizing Flows



Ground truth

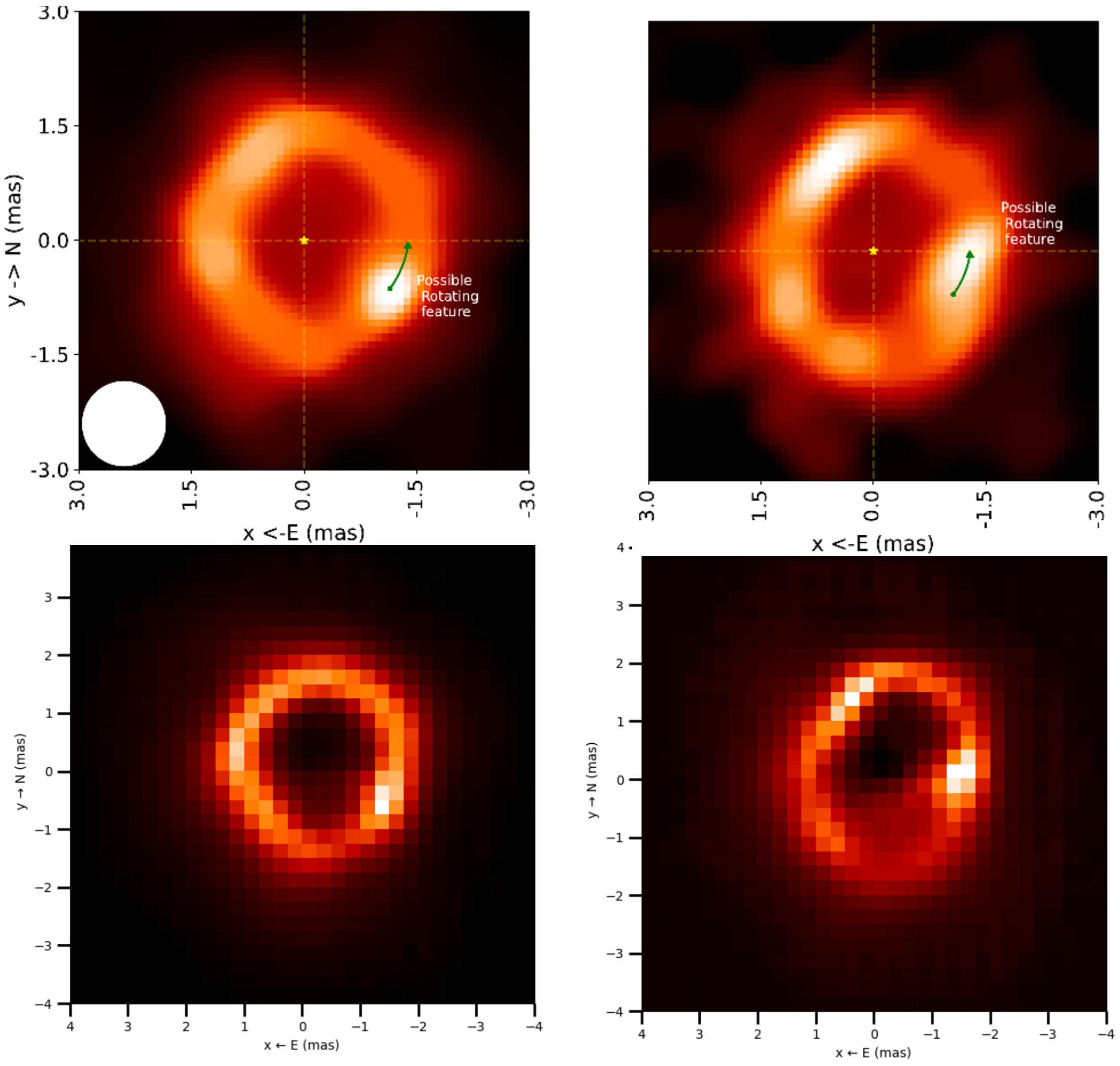


OITOOLS NF

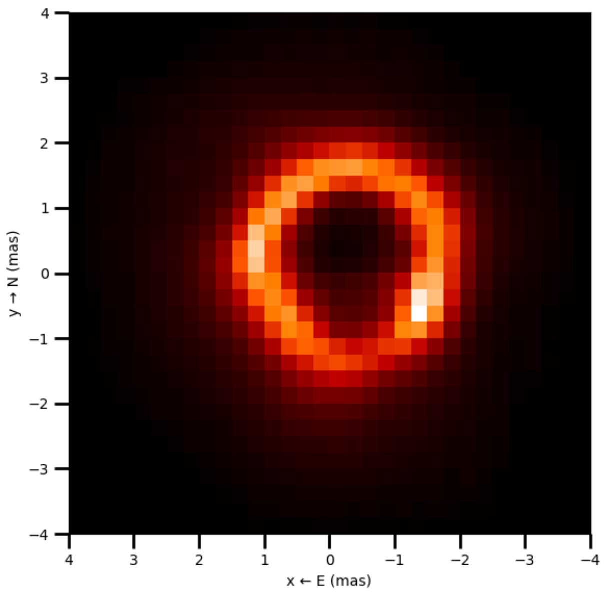


Squeeze Mean
Init = truth

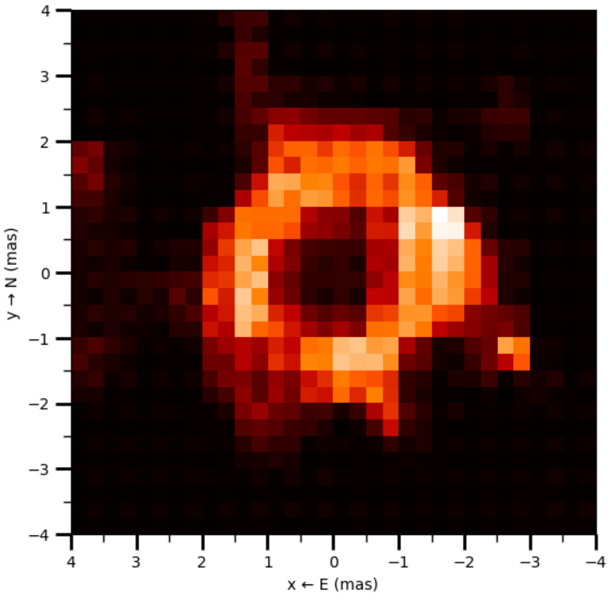
Tests on real data (v1295 Aql)



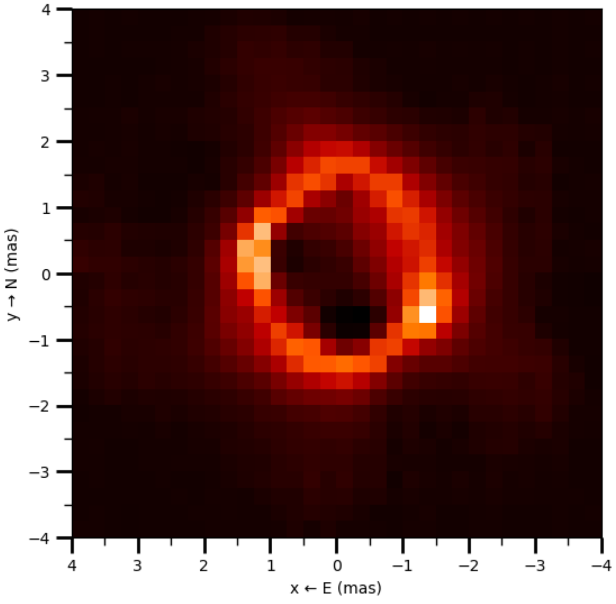
Tests on real data (v1295 Aql)
Influence of YSO training set



Double Sigmoid



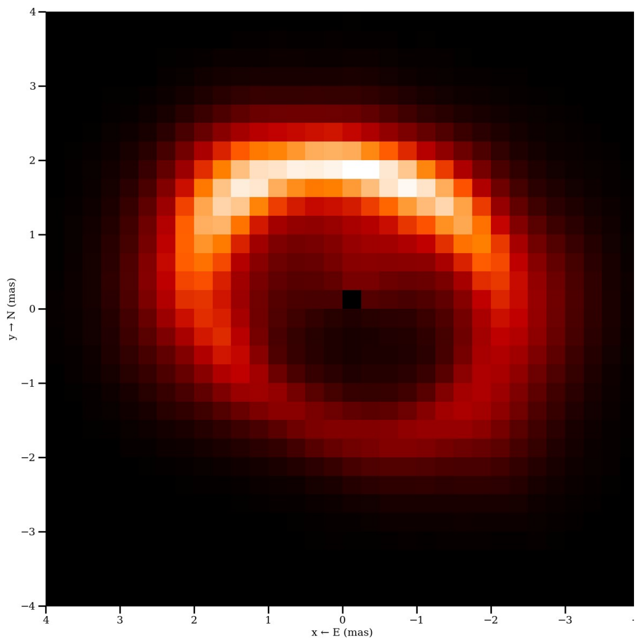
Rectangles



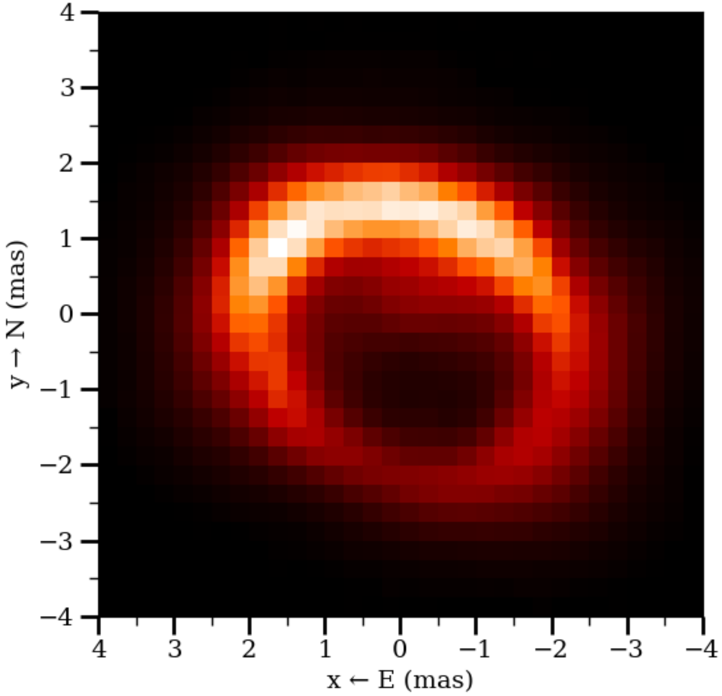
Mixed

Variational inference with Normalizing Flows

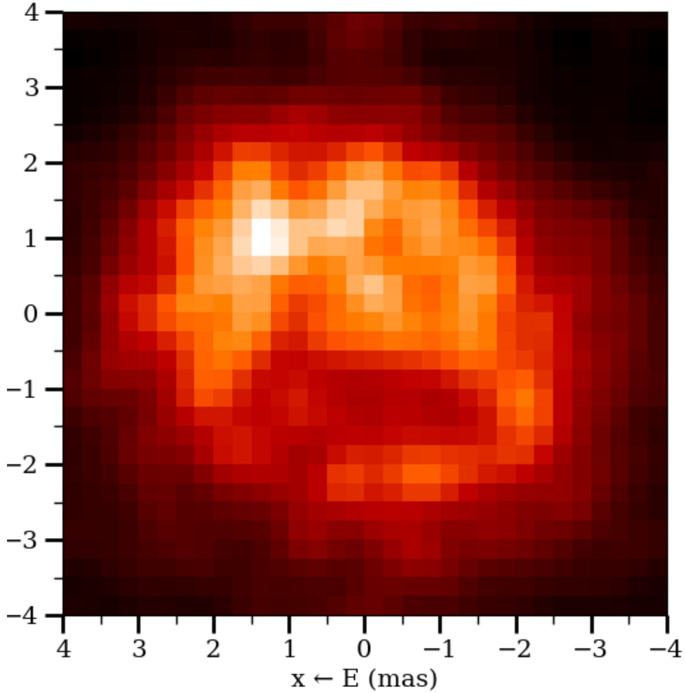
- A normalizing flow network can be used to approximate the posterior probability distribution itself (chi2+regularization)



Ground truth



OITOOLS NF VI Mean



OITOOLS NF VI STD

Work in progress / Conclusion

- Advantages:
  - Higher fidelity in reconstructions
  - Fast (<1 minute reconstruction)
  - No fiddling with regularization weights
- Issues:
  - Model dependent
    - Maybe expand the training set?
  - MAP can still stay stuck into local minima
    - but obvious/high chi2
    - Easy to get unstuck: add a small shift to the latent z
  - VI is slow
    - GPU work in progress