Durham University
Institute for Computational Cosmology

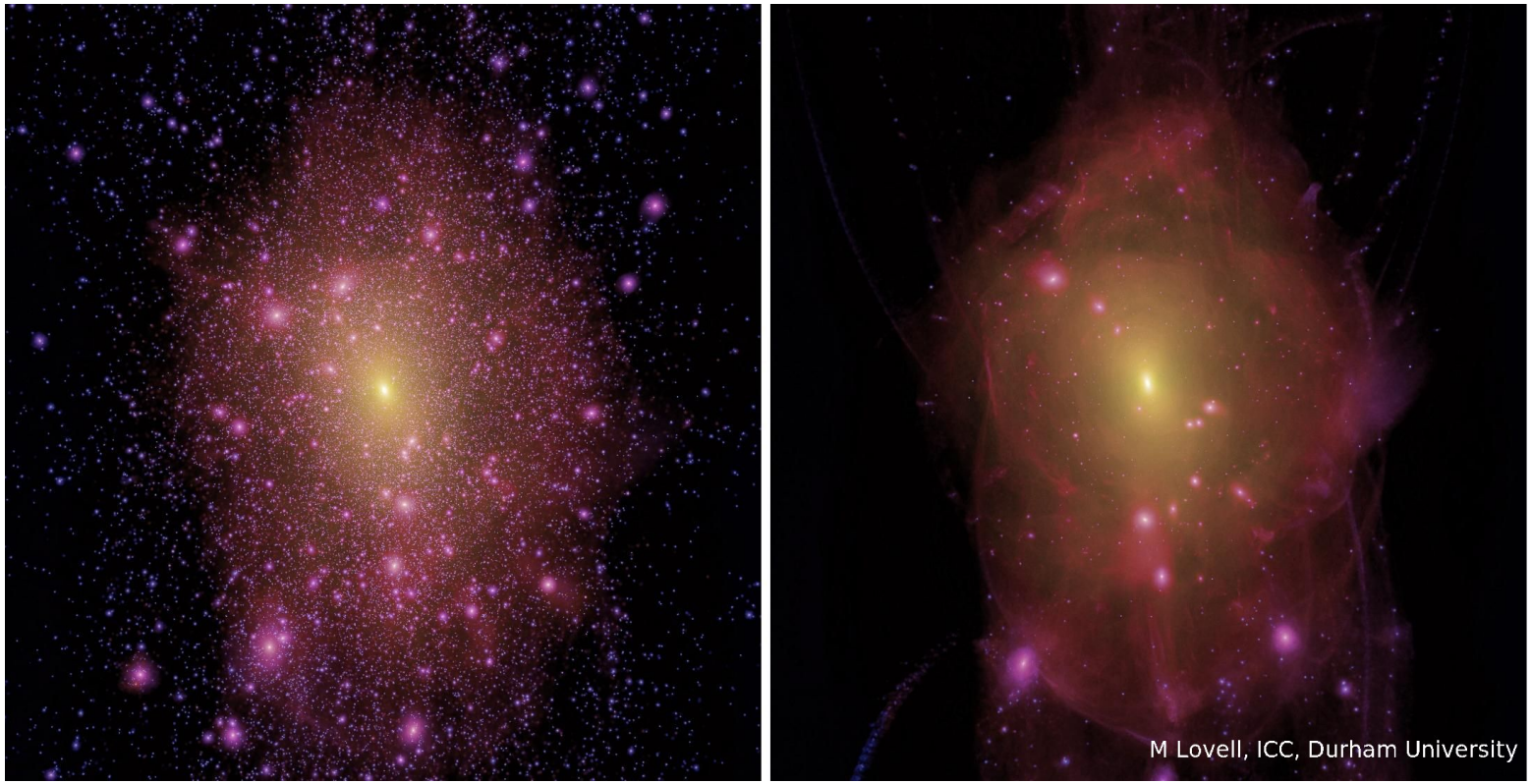
PyAutoLens: Current Nested Sampling methods for Gravitational Lens modelling

Sam Lange (Y2 PhD)
Durham University
Samuel.c.lange@durham.ac.uk

Durham Dark Matter/GGSL group:
C. Frenk, S.Cole, R.Massey, J.Nightingale, A.Amvrosiadis, Q.He, R.Li, X. Cao, S.Lange

Motivation: Dark Matter Searches

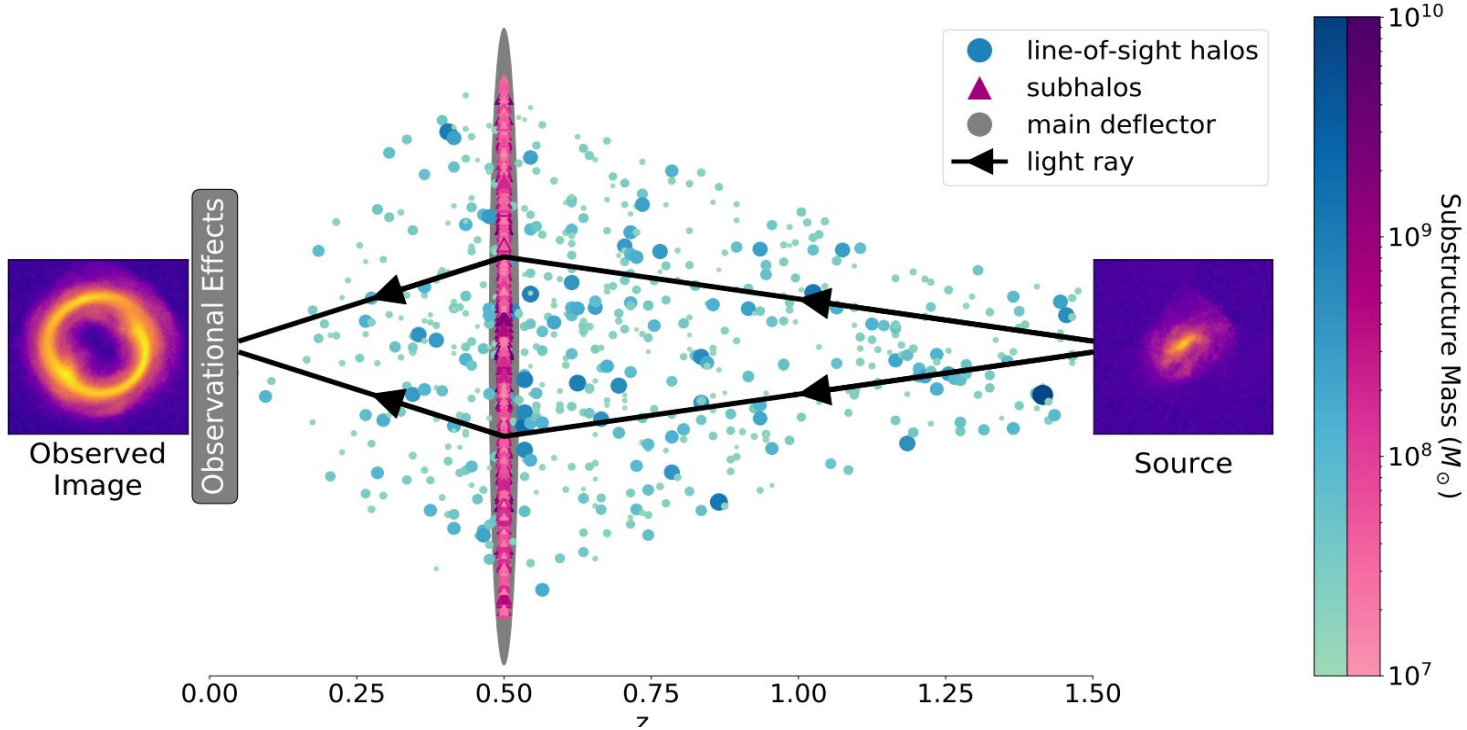
- Primarily CDM vs. WDM.
- CDM predicts many low-mass structures (subhalos).
- Almost all other models smooth subhalos out below a certain mass.

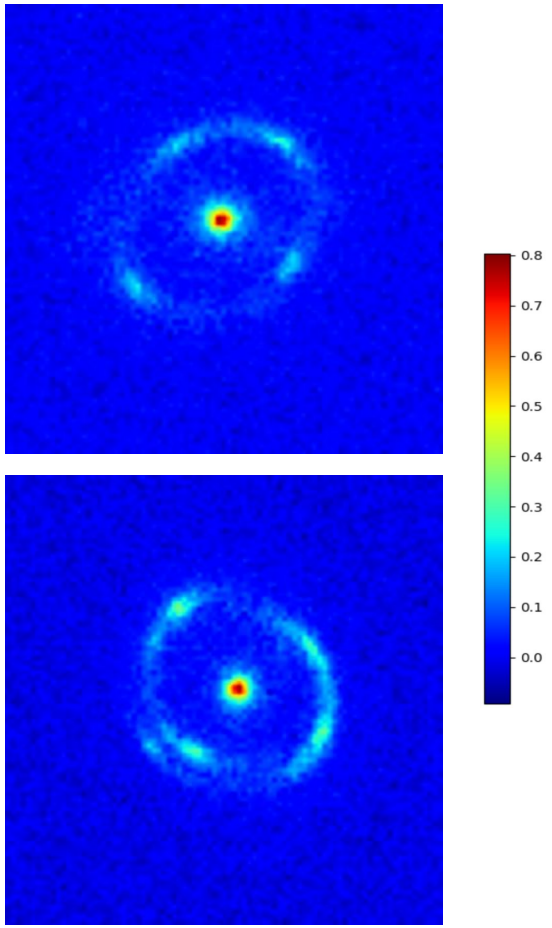


For Reference:
Galaxy Clusters: $10^{14-15} M_{\odot}$
Milky Way: $10^{12} M_{\odot}$
Local Dwarves: $10^{9} M_{\odot}$
Our interest starts around $10^{8.5} M_{\odot}$

Strong Lensing for Substructure

- Strong lensing produces multiple images and/or Einstein rings.
- Additional mass clumps produce perturbations in the image.



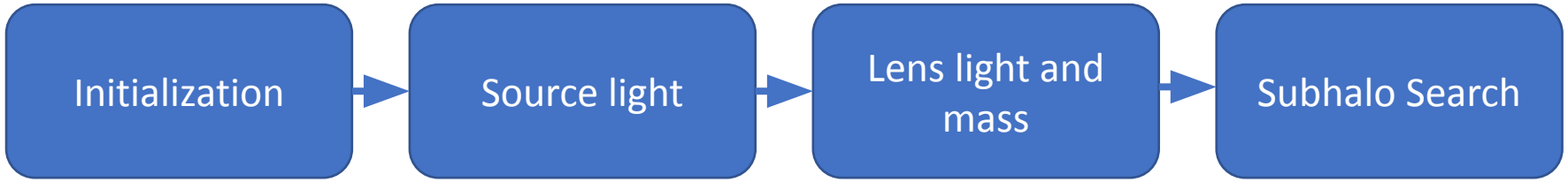


Wagner-Carena, *et al.*, 2022. doi.org/10.48550/arXiv.2203.00690

How do we model?

PyAutoLens

Strong Lensing software developed in Durham:
https://doi.org/10.21105/joss.02825
https://github.com/Jammy2211/PyAutoLens



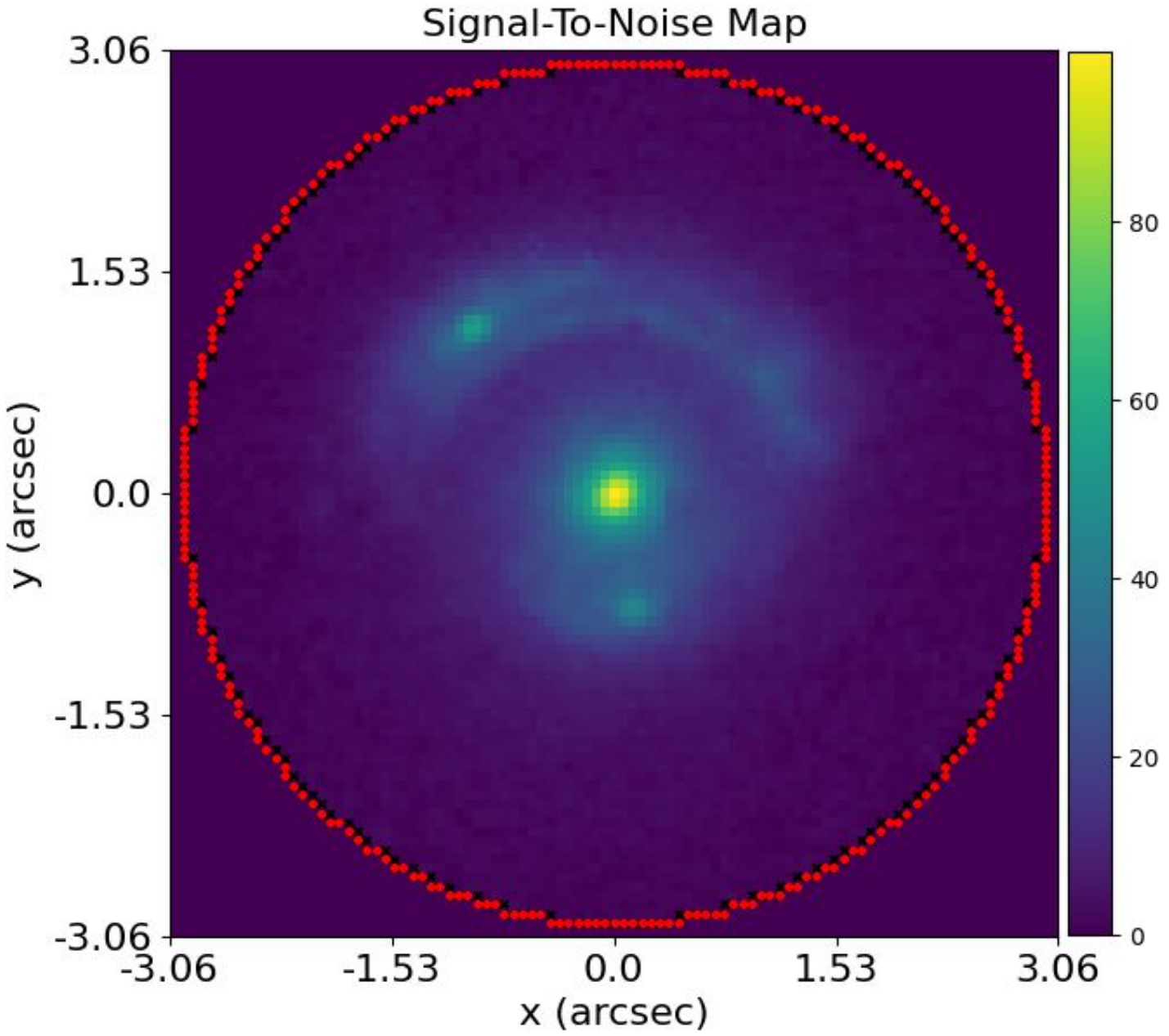
3 fit stages

5 fit stages

1 complex fit stage

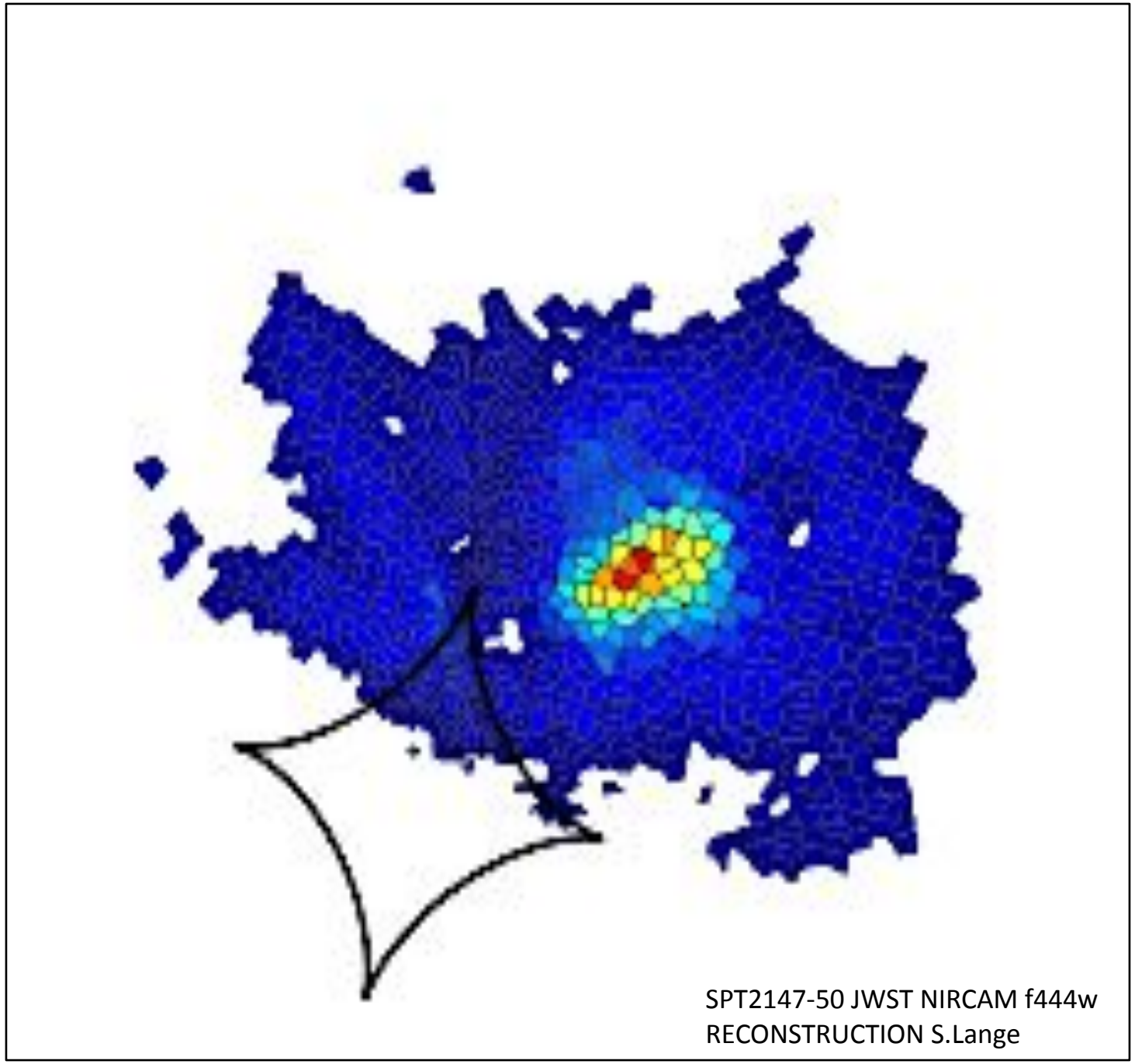
3 fit stages (27 total fits)

Turning this:



SPT2147-50 JWST NIRCAM f444w
S.Lange

…into this:

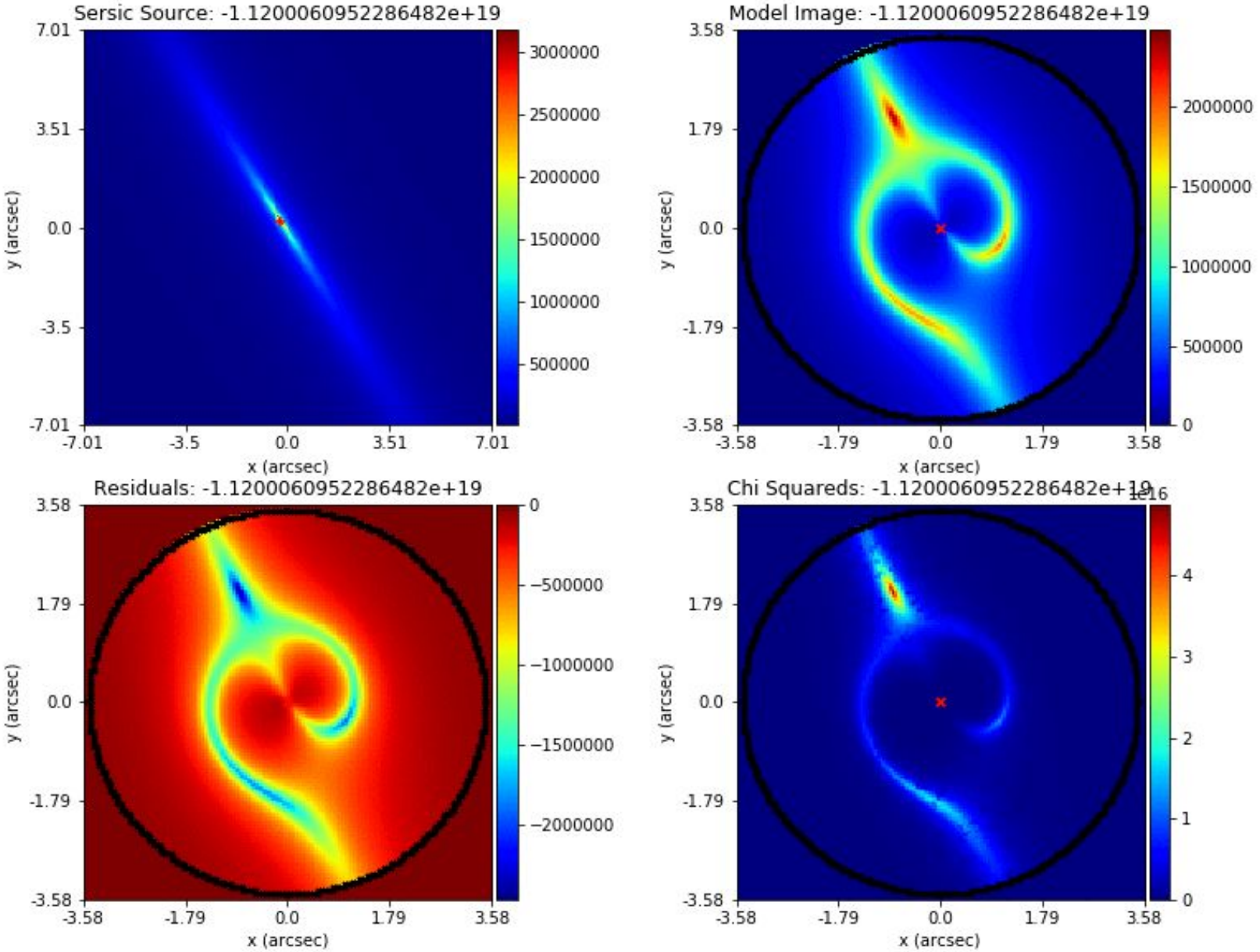


SPT2147-50 JWST NIRCAM f444w
RECONSTRUCTION S.Lange

Amvrosiadis+23 (in prep.)
Lange+23 (in prep.)

Initialisation: Sersic fits

1. Model lens light only
2. Model Lens mass and Source Light
3. Re-model Lens light



Credit: Amy Etherington (Durham)

Source Pixelization: Voronoi Natural Neighbours

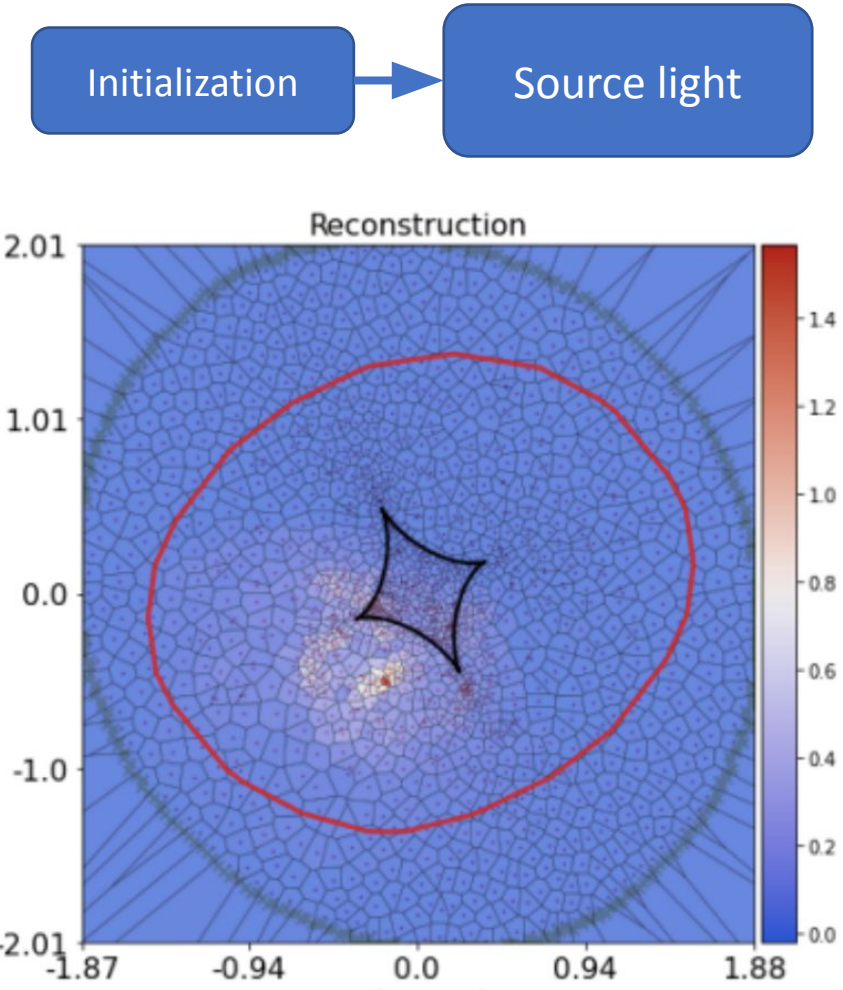
PyAutoLens: VoronoiNN

Adapted from C code by Pavel Sakov based on Fan+05

https://github.com/sakov/nn-c

https://github.com/Jammy2211/PyAutoArray/tree/master/autoarray/util/nn

https://www.researchgate.net/publication/220981984_Hardware-Assisted_Natural_Neighbor_Interpolation



- Voronoi-natural-neighbours mesh allows pixels to change shape to adapt to new points.
- Centres adjusted to give higher resolution to brighter regions.

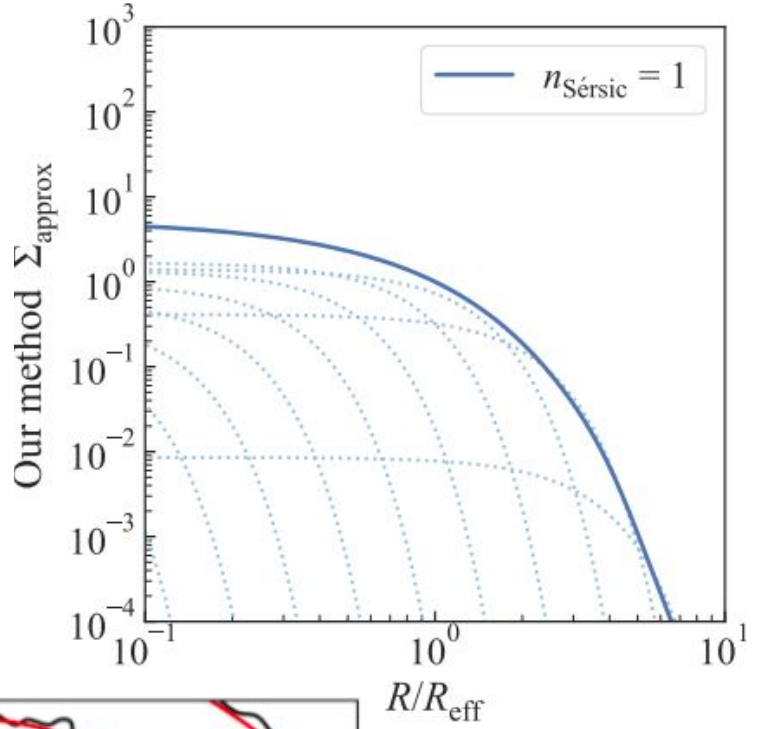
We run 5 fit stages:

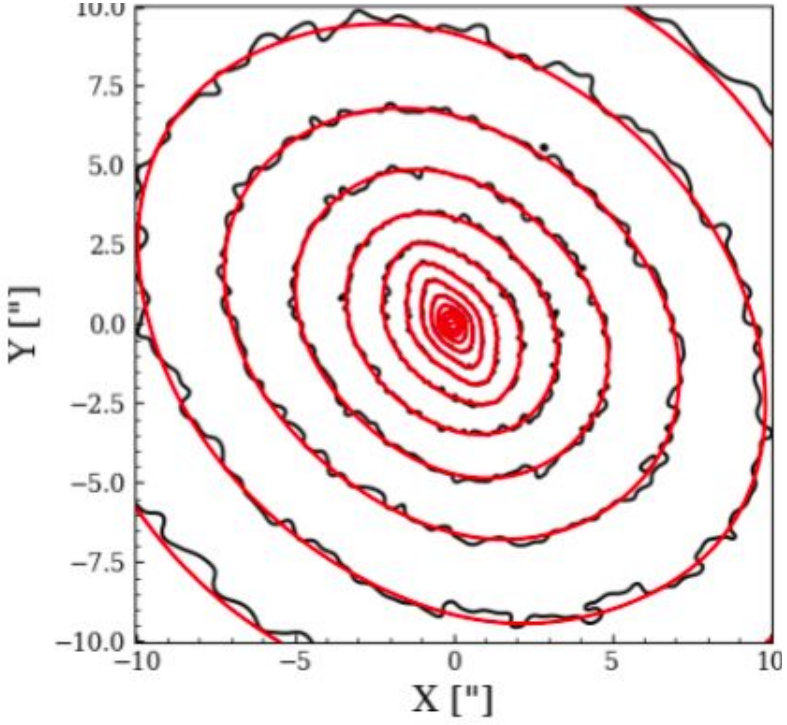
1. Fit a simple source pix/reg
2. Refit lens
3. Fit complex source pix/reg
4. Refit lens
5. Refit complex source adjusting for noise hyperparameters

Lens Modelling: Multiple-Gaussian-Expansio

- Effectively model lens light as the sum of multiple concentric gaussian light distributions.
- We fix centres to each other to reduce parameter space.
- We set up a number of basis (directions) with a fixed number of gaussians on each.
- We fit the ellipticity for each basis using a constant regularization.

Not "our" method – taken from Shajib 19





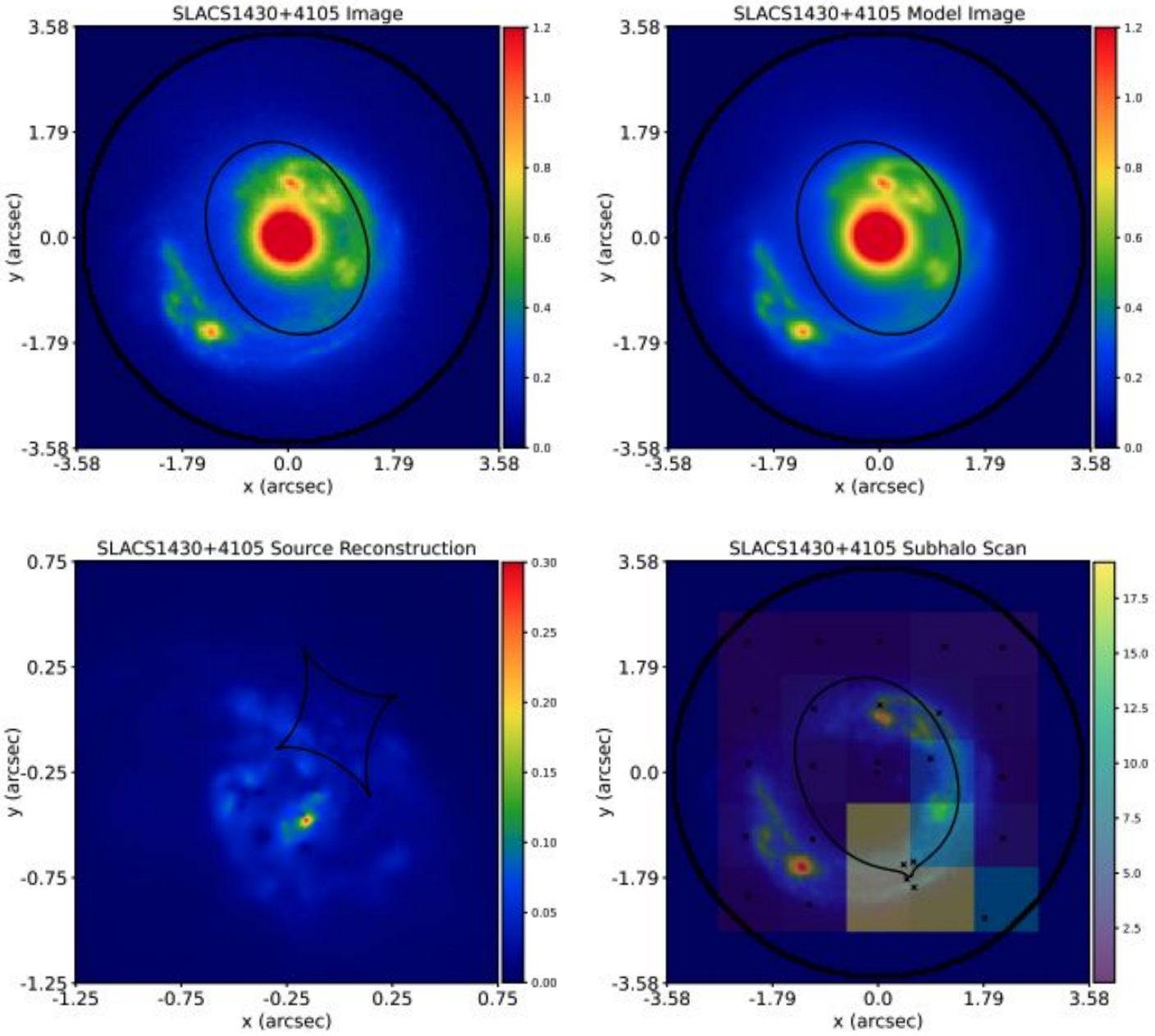
Credit: Q.He
Red: MGE fit
Black: Galaxy light contours from simulation

PyAutoLens: MGE

Custom implementation by Quihan He (He+23 in prep.) based on:
Cappelari 02 https://arxiv.org/abs/astro-ph/0201430
Shajib 19 https://arxiv.org/abs/1906.08263

How do we model: Subhalo

- Split image into grid squares.
- Place subhalo in that square and fit.
- Analyse change in likelihood.



Nightingale, *et al.*, 2022. https://doi.org/10.48550/arXiv.2209.10566

The Data

Nested Sampling with Dynesty

Dynesty

"Dynamic nested sampling for estimating Bayesian posteriors and evidences."

https://dynesty.readthedocs.io/en/latest/index.html

https://academic.oup.com/mnras/article/493/3/3132/5721521?login=true

- Nested Sampling attempts to estimate the Evidence rather than the Posterior and breaks this down into slices.
- Dynamic Iteration:
  - Sample distribution
  - Calculate 'importance function' (where posterior mass is higher)
  - Allocate new live points to important regions
  - Re-sample, etc.

$$\mathcal{Z} \equiv \int_{\Omega_\Theta} \mathcal{L}(\Theta)\pi(\Theta)\mathrm{d}\Theta.$$

Evidence — Likelihood — Prior

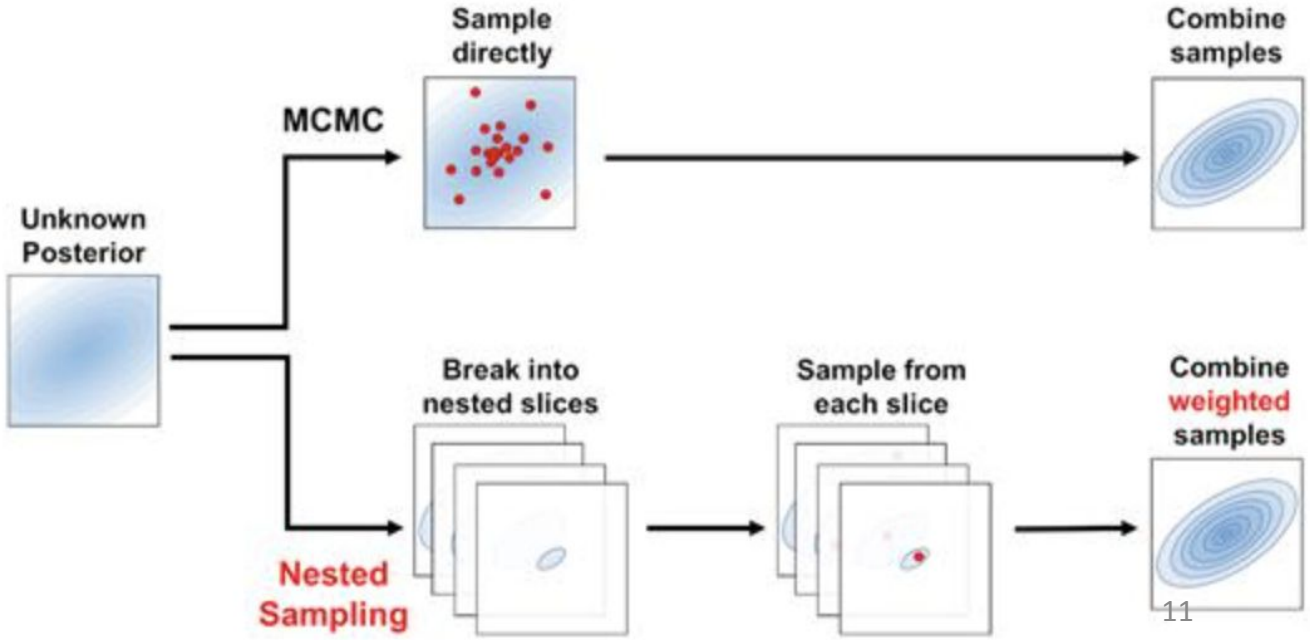
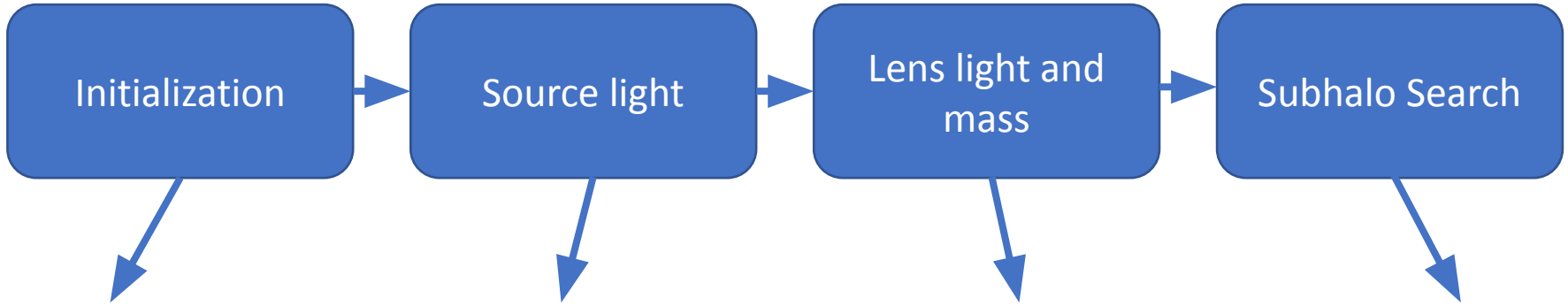


Image: Speagle20: doi:10.1093/mnras/staa278

Number of fit parameters

Example dataset size:

- JWST NIRCAM image 0.063”/pixel with 3” circular mask = 7129 pixels
- Maps to e.g. 1614 Voronoi source pixels



1. 11 parameters
2. 14 parameters
3. 25 parameters

1. 3 parameters
2. 7 parameters
3. 6 Parameters
4. 7 parameters
5. 6 parameters

1. 18/20 parameters

1. 18/20 parameters
2. 25 x 21/23 Parameters
3. 21/23 Parameters

We carry out a number of non-linear searches in chain where the previous posteriors affect the new priors. At each sample, we carry out a linear inversion…

PyAutoFit

Probabalistic programming in Python.

https://pyautofit.readthedocs.io/en/latest/index.html

Inversion and χ^2

Nightengale+18
https://doi.org/10.1093/mnras/sty1264

- Inversion process:
  - Produce model lens light, blur with psf, subtract from image.
  - Produce model lens mass.
  - Trace image pixel light backwards using the lens mass potential to source plane.
  - Trace those source pixels back to image plane to get model image (and residuals).
  - Compute the best superposition of psf-smeared source pixel images to best reproduce the observed image (minimise χ^2).

$$\chi^2 = \sum_{j=1}^{J} \left[\frac{\left(\sum_{i=1}^{I} s_i f_{ij} \right) + b_j - d_j}{\sigma_j} \right]$$

- s_i: source pixel i
- f_{ij}: mapping from image pixel j to source pixel i
- b_j: model lens light value
- d_j, σ_j: observed image flux in pixel j with statistical uncertainty σ.

https://github.com/Jammy2211/autolens_workspace/blob/release/notebooks/imaging/log_likelihood_function/pixelization.ipynb

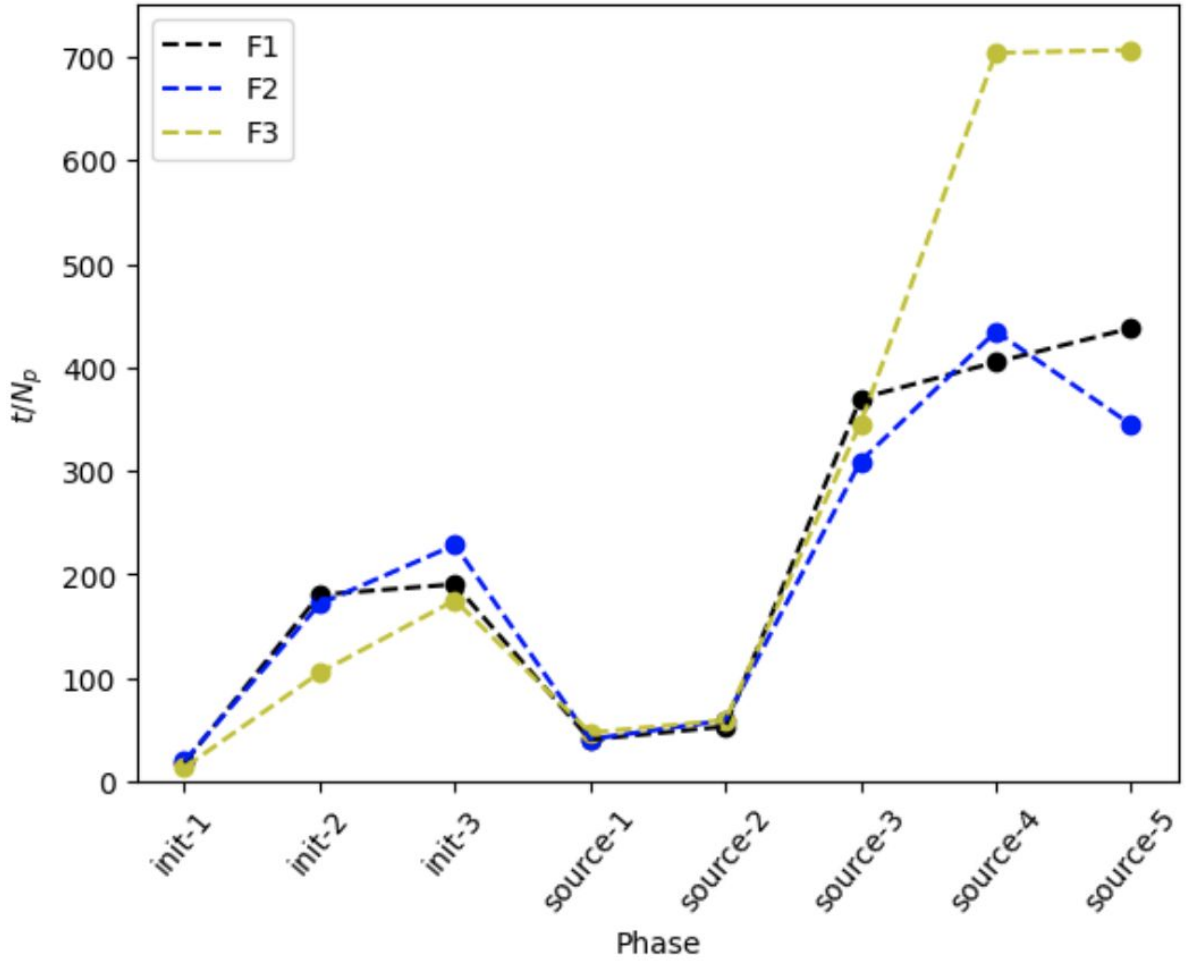
Computing costs

Runtimes

All taken from real data runs on the same system in 3 different filters

Parameter-normalised runtimes per phase

	init-1	init-2	init-3	source-1	source-2	source-3	source-4	source-5	MGE
F1	17.6	180.3	190.4	40.3	52.9	370.0	405.0	437.8	25178.6
F2	19.4	171.4	228.4	40.7	59.0	309.2	434.6	344.7	25531.2
F3	13.3	104.9	174.9	47.0	59.1	345.8	703.9	706.7	7492.2



Runtimes

Number of samples

	init-1	init-2	init-3	source-1	source-2	source-3	source-4	source-5	MGE
F1	44520	198550	329860	4160	14350	7720	12510	10065	320636
F2	49010	187430	385320	3890	15735	5690	11500	7555	332550
F3	45109	139790	367060	3690	11968	3551	10758	9441	317810

All taken from real data runs on the same system in 3 different filters

Total runtime (just to prepare for what we want!)
3-7 days on average...

Current advances in the works

- Changing some numpy processing to custom-built processing.
- Exploration of GPUs.
- Re-working to reduce the number of fit stages.
- Smoothing MGE processing.

Could SWYFT help?

Thank You!

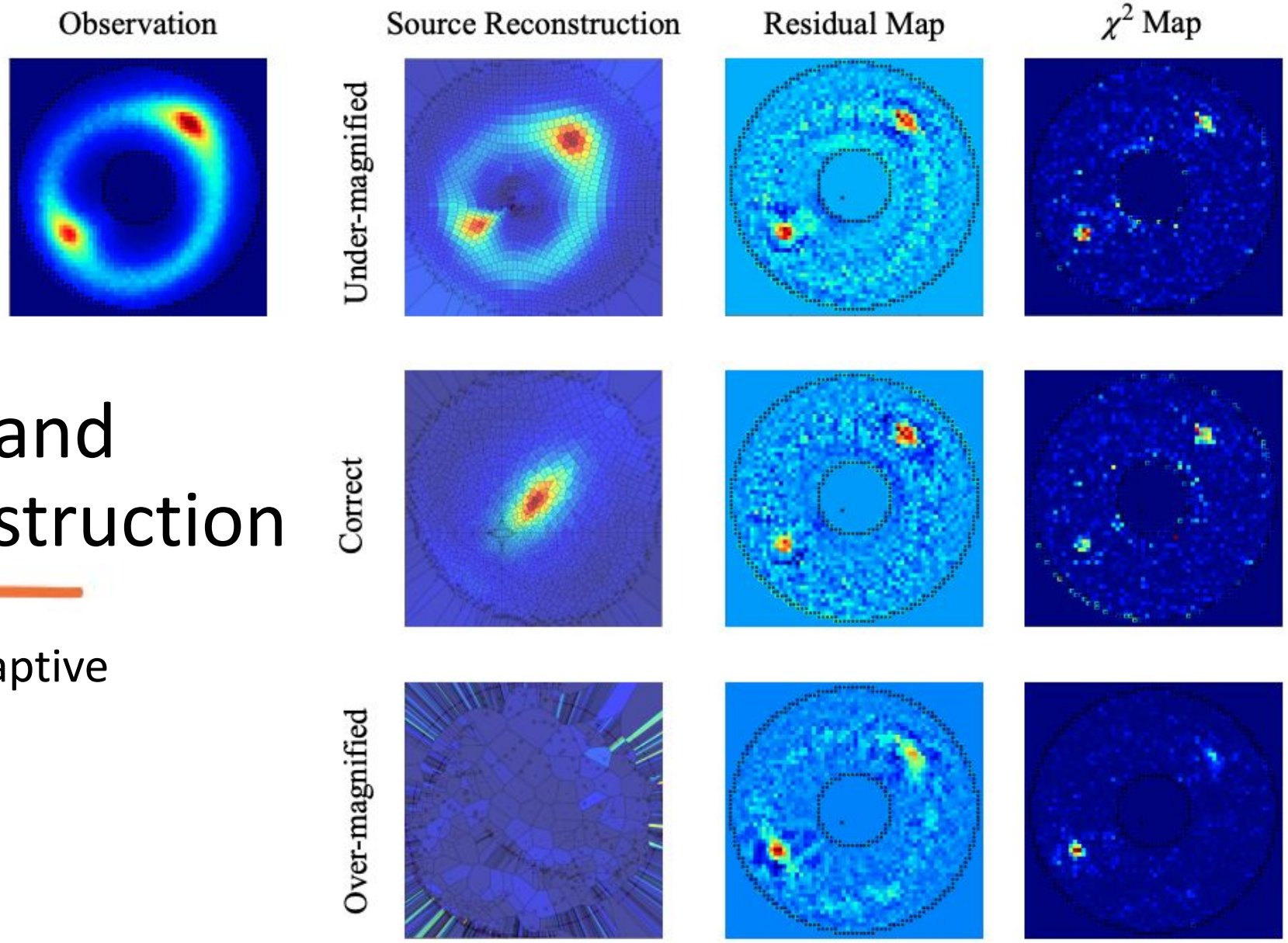
samuel.c.lange@durham.ac.uk

- Pros:
  - The reductions in sampling of $\mathcal{O}(3)$ could certainly speed the process.
  - Ability to truncate to only the interesting posteriors would potentially reduce many fit stages and focus the research.
  - We would have the ability to easily detect multiple subhalos per dataset and line-of-sight subhalos.

- Cons (?):
  - Any NN has an element of 'black box' to it – the current framework, whilst complex, can be followed intuitively.
  - Skipping over parameters means we could lose some interesting science – e.g. our group currently is doing interesting source science as an aside.

Bonus slides

Maresca, Dye & Li, 2021.
https://doi.org/10.1093/mnras/stab387



Source Inversion and Unphysical reconstruction

- Reconstruct onto adaptive pixel-grid.
- Likelihood quirks can sometimes give *local* minima.

Source Pixelization: Voronoi Natural Neighbours

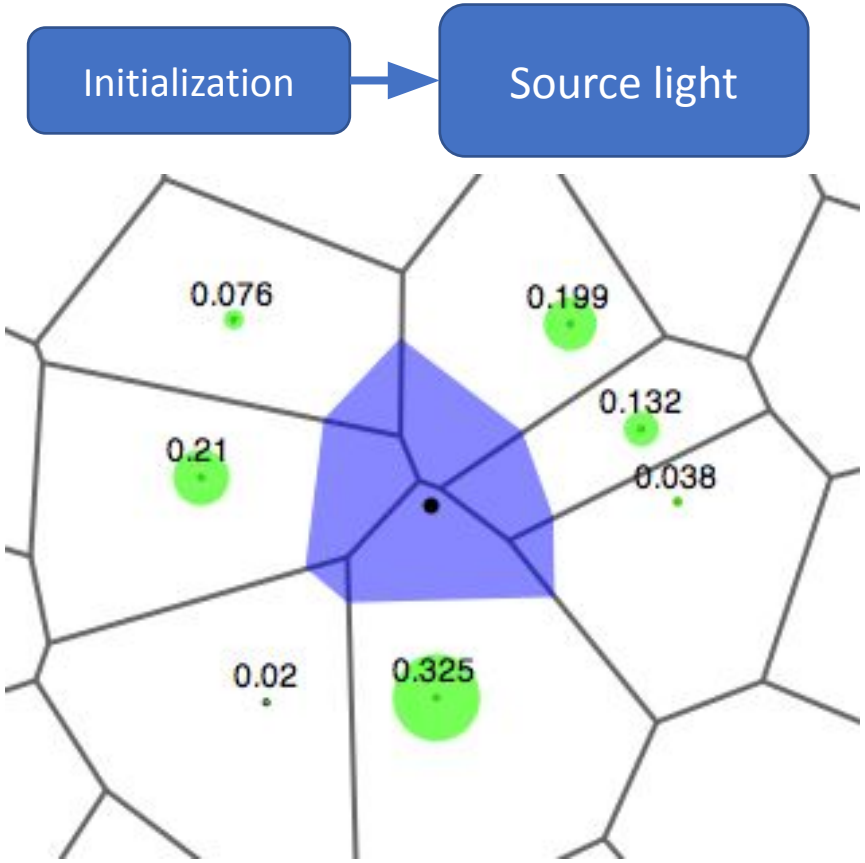
PyAutoLens: VoronoiNN

Adapted from C code by Pavel Sakov based on Fan+05

https://github.com/sakov/nn-c

https://github.com/Jammy2211/PyAutoArray/tree/master/autoarray/util/nn

https://www.researchgate.net/publication/220981984_Hardware-Assisted_Natural_Neighbor_Interpolation



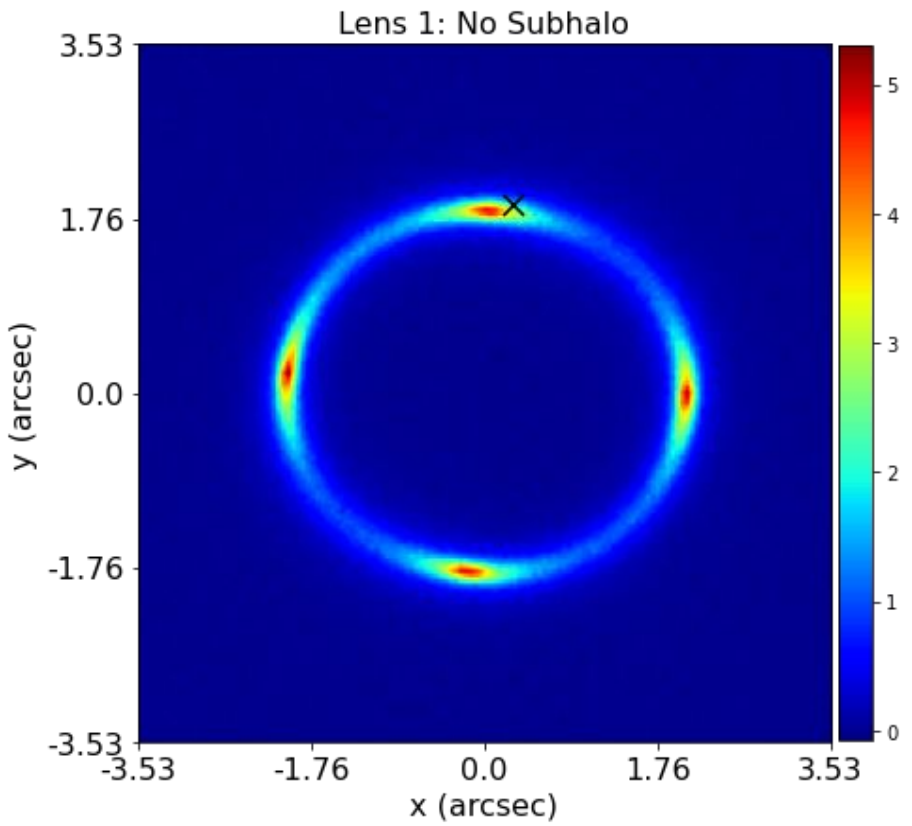
Inserting x, G is the new estimate with w_i the weight for $f(x_i)$ – the known data at x_i.

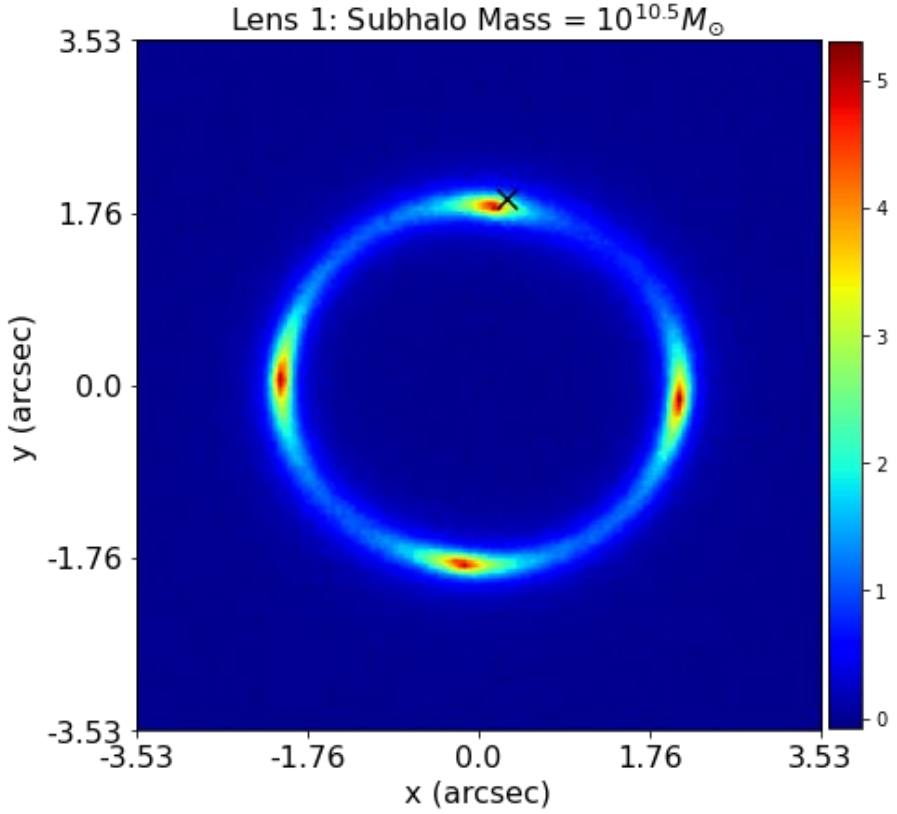
$$G(x) = \sum_{i=1}^{n} w_i(x) f(x_i)$$

Image: Wikipedia https://en.wikipedia.org/wiki/Natural_neighbor_interpolation

How do we model: Subhalo

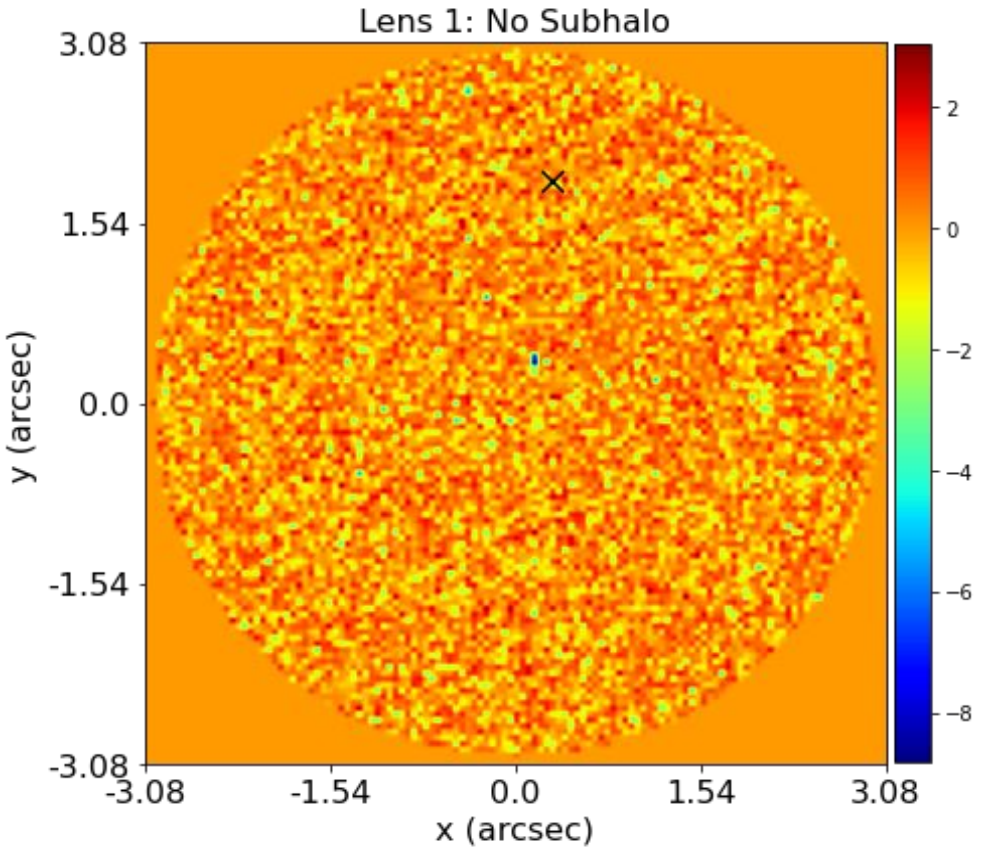
- Placed a (rather large) subhalo on the ring.
- We may not see any difference by eye here…

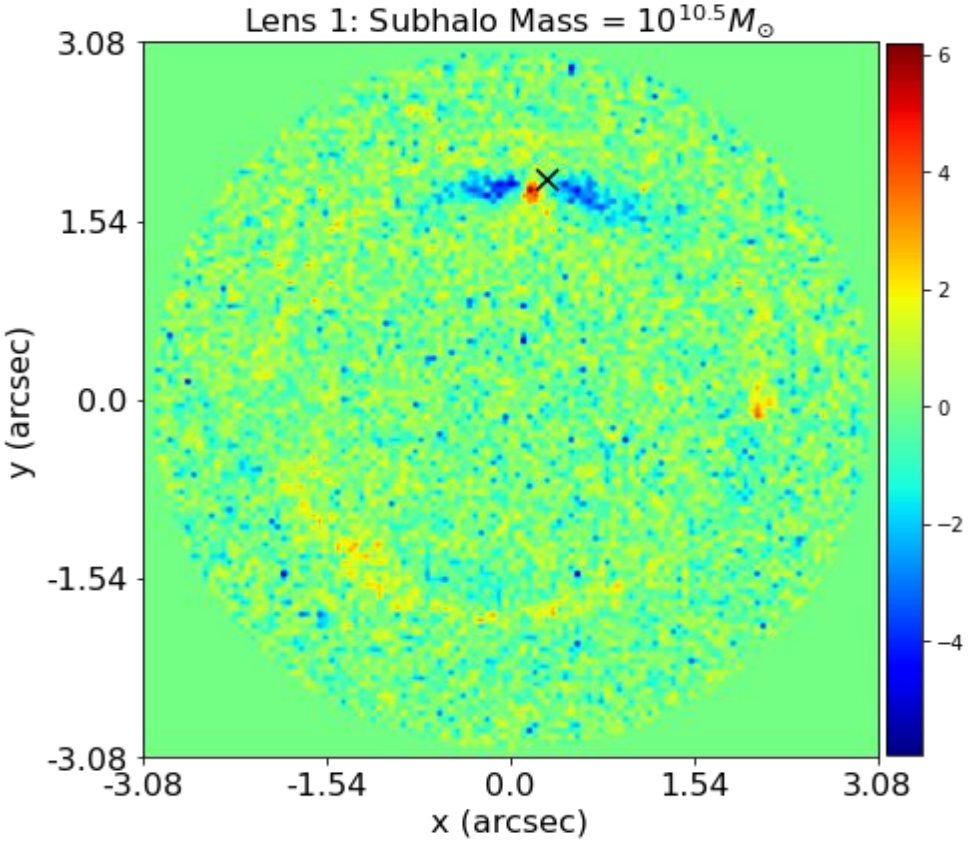




How do we model: Subhalo

- …but we certainly see the difference here!
- Applying pipeline up to now leaves these clear residuals around where the subhalo is.





Dynasty Algorithm

Algorithm 1: Static Nested Sampling

```
// Initialize live points.
```
Draw K "live" points $\{\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K\}$ from the prior $\pi(\boldsymbol{\Theta})$.
```
// Main sampling loop.
```
while stopping criterion not met **do**
- Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.
- Add the kth live point $\boldsymbol{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of "dead" points.
- Sample a new point $\boldsymbol{\Theta}'$ from the prior subject to the constraint$\mathcal{L}(\boldsymbol{\Theta}') \geq \mathcal{L}^{\min}$.
- Replace $\boldsymbol{\Theta}_k$ with $\boldsymbol{\Theta}'$.
- `// Check whether to stop.`
- Evaluate stopping criterion.

end
```
// Add final live points.
```
while $K > 0$ **do**
- Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.
- Add the kth live point $\boldsymbol{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of "dead" points.
- Remove $\boldsymbol{\Theta}_k$ from the set of live points.
- Set $K = K - 1$.

end

Algorithm 3: Iterative Dynamic Nested Sampling

```
// Baseline Nested Sampling run.
```
Run Static Nested Sampling (Algorithm 1) with:
(a) K live points
(b) sampled uniformly from the prior $\pi(\boldsymbol{\Theta})$
(c) until the default Static Nested Sampling stopping criterion is met.
```
// Main sampling loop.
```
while stopping criterion not met **do**
- `// Find region where new samples should be allocated.`
- Compute relative importance $\{\hat{\mathcal{I}}(\hat{X}_i)\}$ over all dead points $\{\boldsymbol{\Theta}_i\}$.
- Use $\{\hat{\mathcal{I}}_i\}$ to assignlower $\mathcal{L}^{\text{low}} = \mathcal{L}(\hat{X}^{\text{high}})$ and upper $\mathcal{L}^{\text{high}} = \mathcal{L}(\hat{X}^{\text{low}})$ likelihood bounds.
- `// Batch Nested Sampling run.`
- Run Static Nested Sampling (Algorithm 1) with:
- (a) K' live points
- (b) sampled uniformly from the constrained prior $\pi_\lambda(\boldsymbol{\Theta})$based on the lower likelihood bound $\lambda = \mathcal{L}^{\text{low}}$
- (c) until the likelihood $\mathcal{L}(\boldsymbol{\Theta})$ of the last dead pointexceeds the upper likelihood bound $\mathcal{L}^{\text{high}}$.
- `// Merge samples from batch.`
- Merge new batch of dead points $\{\boldsymbol{\Theta}'_i\}$into the previous set of dead points $\{\boldsymbol{\Theta}_i\}$.
- `// Check whether to stop.`
- Evaluate stopping criterion.

end

Speagle20: doi:10.1093/mnras/staa278

Likelihood function (Bayesian Evidence)

$$-2\ln\epsilon = \chi^2 + s^T H s + \ln\,[\det(F+H)] - \ln\,[\det(H)] + \sum_{j=1}^{J} \ln\left[2\pi(\sigma_j)^2\right].$$

$\boldsymbol{H}$: *Regularization matrix.*

$\boldsymbol{F}$: *Curvature matrix* $F_{ik} = \sum_{j=1}^{J} \frac{f_{ij} f_{kj}}{\sigma_j^2}$

The final term means lower S/N images have lower evidences.

MGE search cornerplot (messy!)