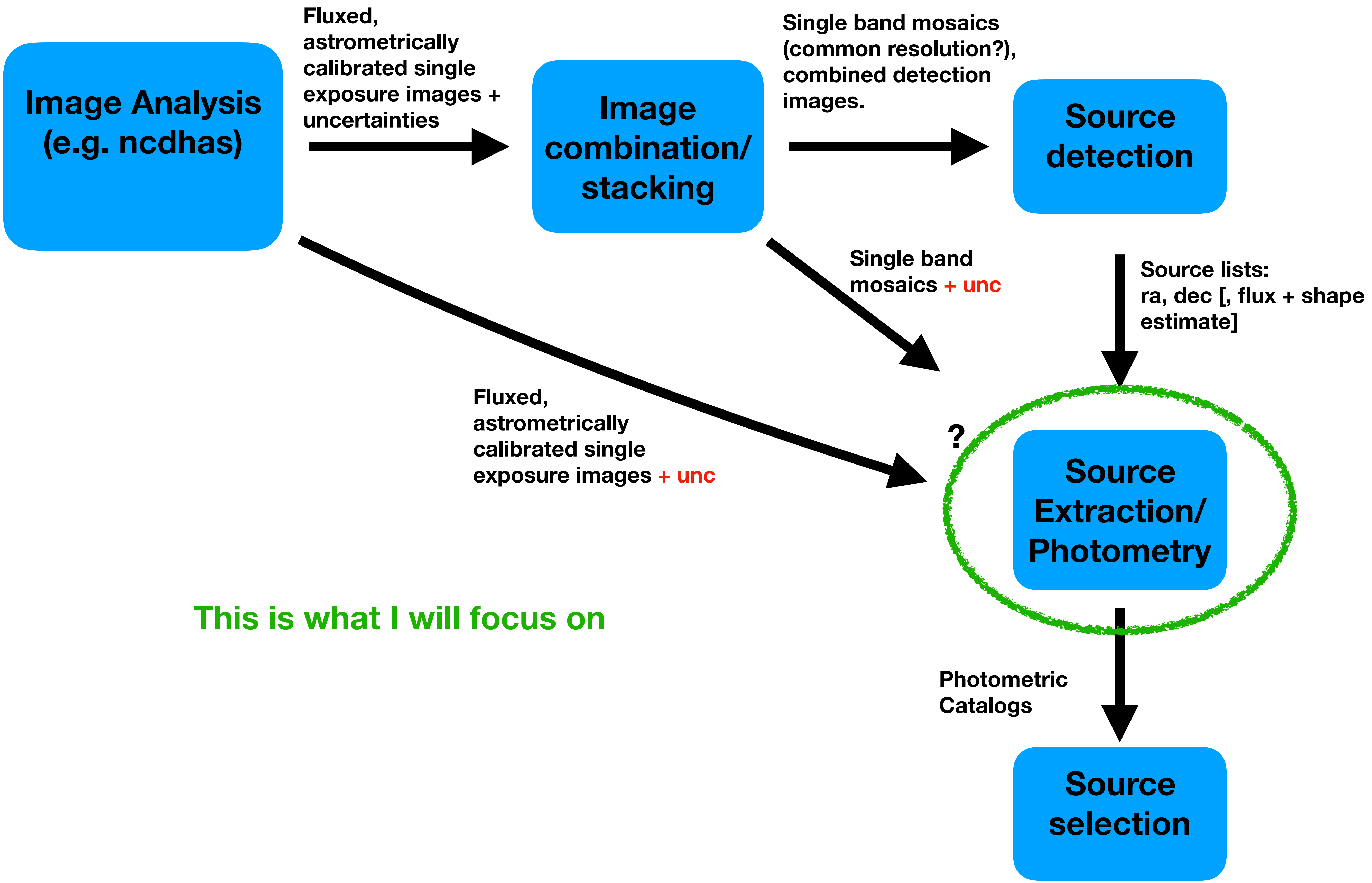
NIRCAM Multiband Galaxy Photometry

Ben Johnson
(Harvard)
w/ D. Eisenstein, S. Tachella, others

Tucson Jan 17th 2018

Galaxy Multiband Photometry



Special NIRCam considerations

- Undersampled PSF
- Different PSF sampling in different bands
- Complicated PSF
- Many pixels per object
- Overlapping sources
- no need for s/g classification
- Cosmic rays
- Distortions
- Want realistic uncertainties, covariances thereof
- (Want analytic gradients of the model)

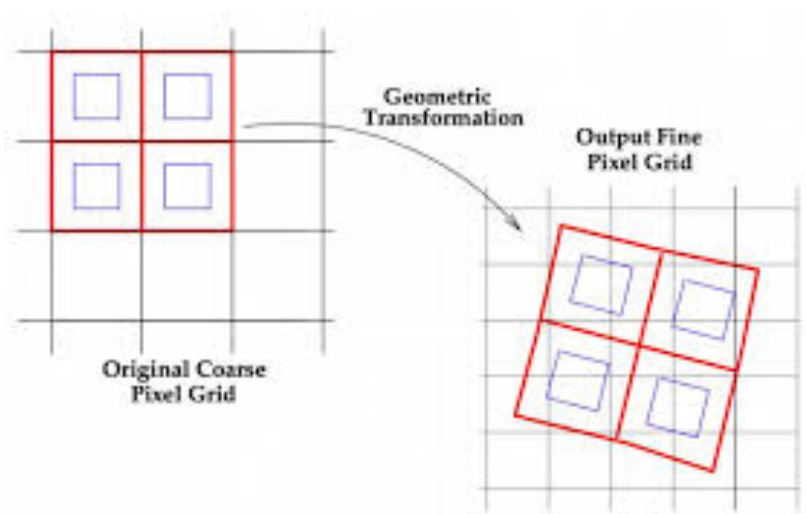
Current Practice

SExtractor: Detection + Photometry

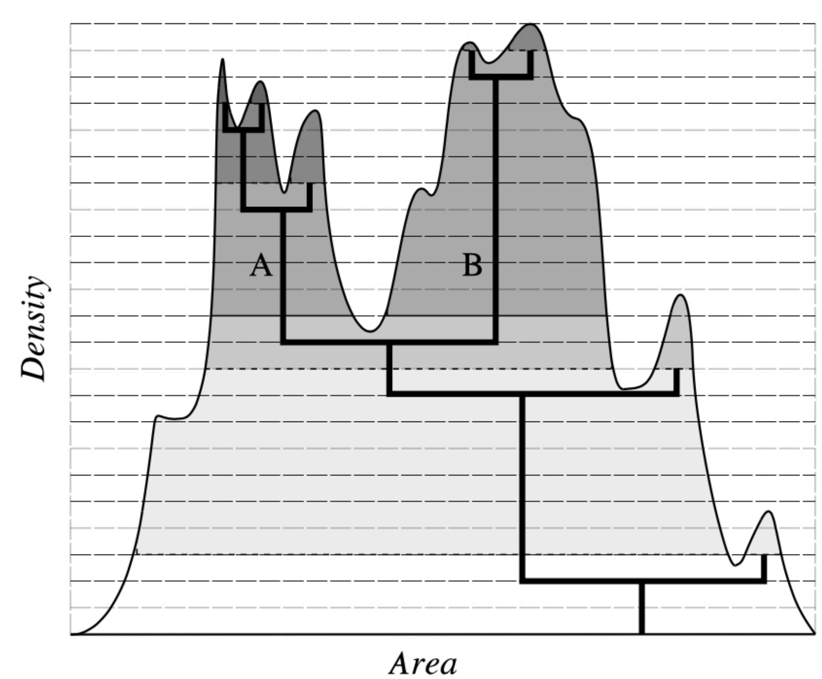
- ✦ drizzled images (induces covariant noise)
- ✦ convolved to common PSF (throws away information, induces covariant noise)
- ✦ dual image mode with “detection image” (must tune thresholds)
- ✦ deblending and segmentation (based on well motivated heuristics, must tune)
- ✦ aperture photometry + corrections (overlapping sources?, source profiles?)

It’s Fast!!

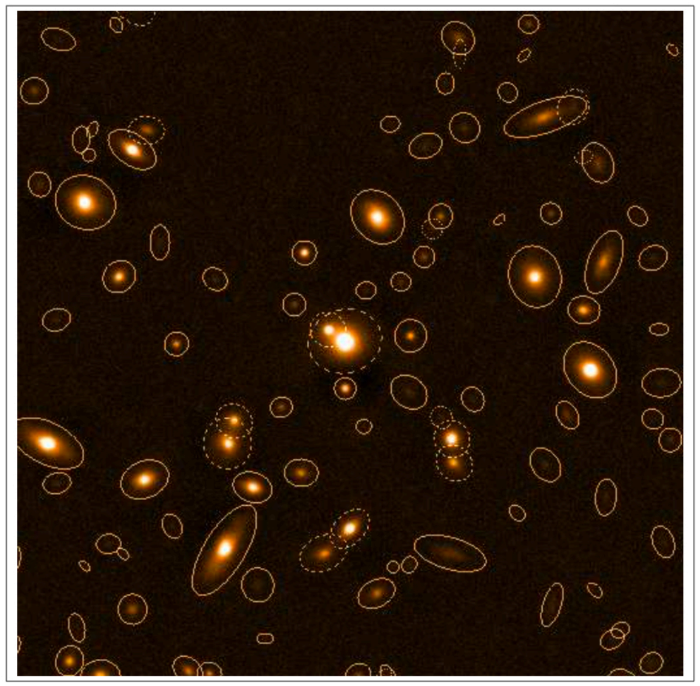
Drizzling



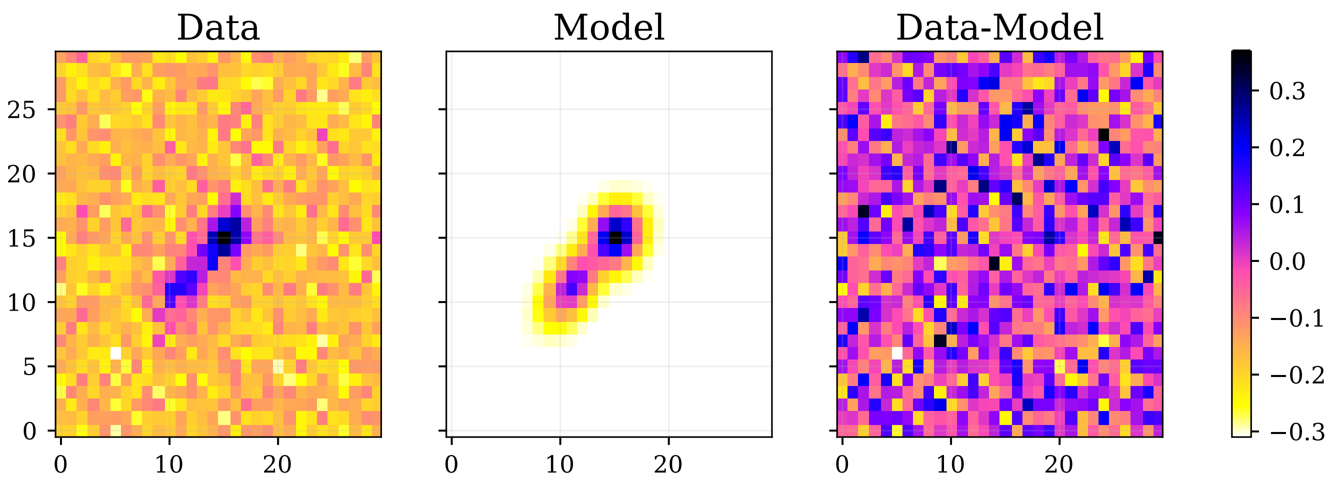
Deblending

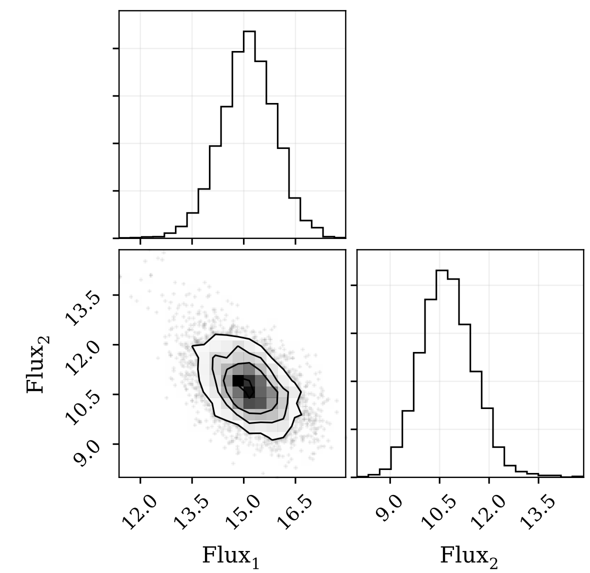


Segmentation



Generative modeling





The step beyond SExtractor is to introduce generative models of the data.
=> Produce a model of the data given parameters, and compute the likelihood of the observed data, given those parameters.

Combined with parameter priors and Bayes rule you can then infer the probability of the parameters given the data.

This process will allow:

1) **proper, probabilistic assignment of flux in overlapping sources**
2) **full propagation of errors and covariances (e.g. between shape, position and flux)**
3) **Incorporation of useful prior information**
4) **Use of *ALL* the information in the data from all the bands**
5) **(potentially, probabilistic detection)**

However, it can be time consuming to produce the many models of the data needed.

Overview of Generative modeling codes

Most of these assume sources already detected/identified

Code	Application	Notes
GalFit	General	Incredibly versatile. Maybe not so speedy? C binary, unmodifieable
GIM2D	HST	single source. rather old
Photo	SDSS/HSC/LSST	cModel quantities
The Tractor	WISE/Legacy Surveys	Has gradients! In python+numpy
Celeste	SDSS	“Tractor on steroids”. In julia, with deep parallelization support
TPHOT	CANDELS, DEEP	Used for data with very different resolutions
MOFONGO	UDF, 3DHST	“”
XID+	Herschel	“”
im3shape	Lensing/DES	designed with single band images in mind
ngmix	Lensing	mostly undocumented
PCAT	SDSS/Fermi	Sophisticated probabilistic detection
GalSim	Lensing	Used to build images for lensing challenges. No gradients
UFIG	Lensing	
PhoSim	LSST	Incredibly detailed optical path simulation (including atmosphere)
Guitarra	NIRCAM	I’ve heard of that!
DAOPHOT	General	The standard. Has peak-finding algorithms for detection
Dolphot	HST	
MOPEX	2MASS/Spitzer/General	
crowdsource	Legacy Surveys	

Overview of Generative modeling codes

Common(ish) Strategies in Generative Modeling and Fitting

Convolution:

1) FFTs (more accurate)
2) Gaussian mixtures (GM; faster)
3) Photon Shooting (PS; model contains poisson noise)
4) Phonions

Undersampling:

1) use “drizzled” images (correlated noise)
2) generate intermediate critically sampled images and then downsample them (memory and time intensive)

Galaxy model parameterization:

1) Sersic Profiles
2) higher resolution images
3) shapelets, etc..

Fitting:

1) Optimization (least-squares or with gradients)
2) Sampling
3) Variational Inference

Overview of Generative modeling codes

Code	PSF Convolution	Undersampling support	Galaxy Parameterization	Fitting?
Photo	?	No	DeV + Exp	?
Tractor	FFT, GM	No	Sersic GM, high-res image	Optimization. sampling possible
Celeste	GM	No	Sersic GM	Variational Inference
TPHOT, XID+, MOFONGO	FFT	No	high-res image	Least-squares optimization
GalFit	FFT	?	Many!!!	Least-squares optimization
im3shape	FFT	Yes (via intermediate oversampled images)	Sersic	Yes
UFIG	Photon Shooting	Yes	Sersic	None
GalSim	PS or FFT	Yes (via intermediate oversampled images)	Sersic	None

None of these packages are quite suitable out of the box

Instead of modifying them, we are rolling our own, tailored for NIRCAM: forcepho (for now)

This gives us more flexibility in optimizing the code, designing it to our purposes, and testing different approaches

Forcepho

Common(ish) Strategies in Generative Modeling and Fitting

Convolution:

1) FFTs (more accurate)
2) Gaussian mixtures (GM; much faster)
3) Photon Shooting (PS; model contains poisson noise)
4) Phonions

Undersampling:

1) use “drizzled” images (correlated noise)
2) generate intermediate critically sampled images and then downsample them (memory and time intensive)
3) second order approximation to pixel integral

Galaxy model parameterization:

1) Sersic Profiles
2) higher resolution images
3) shapelets, etc..

Fitting:

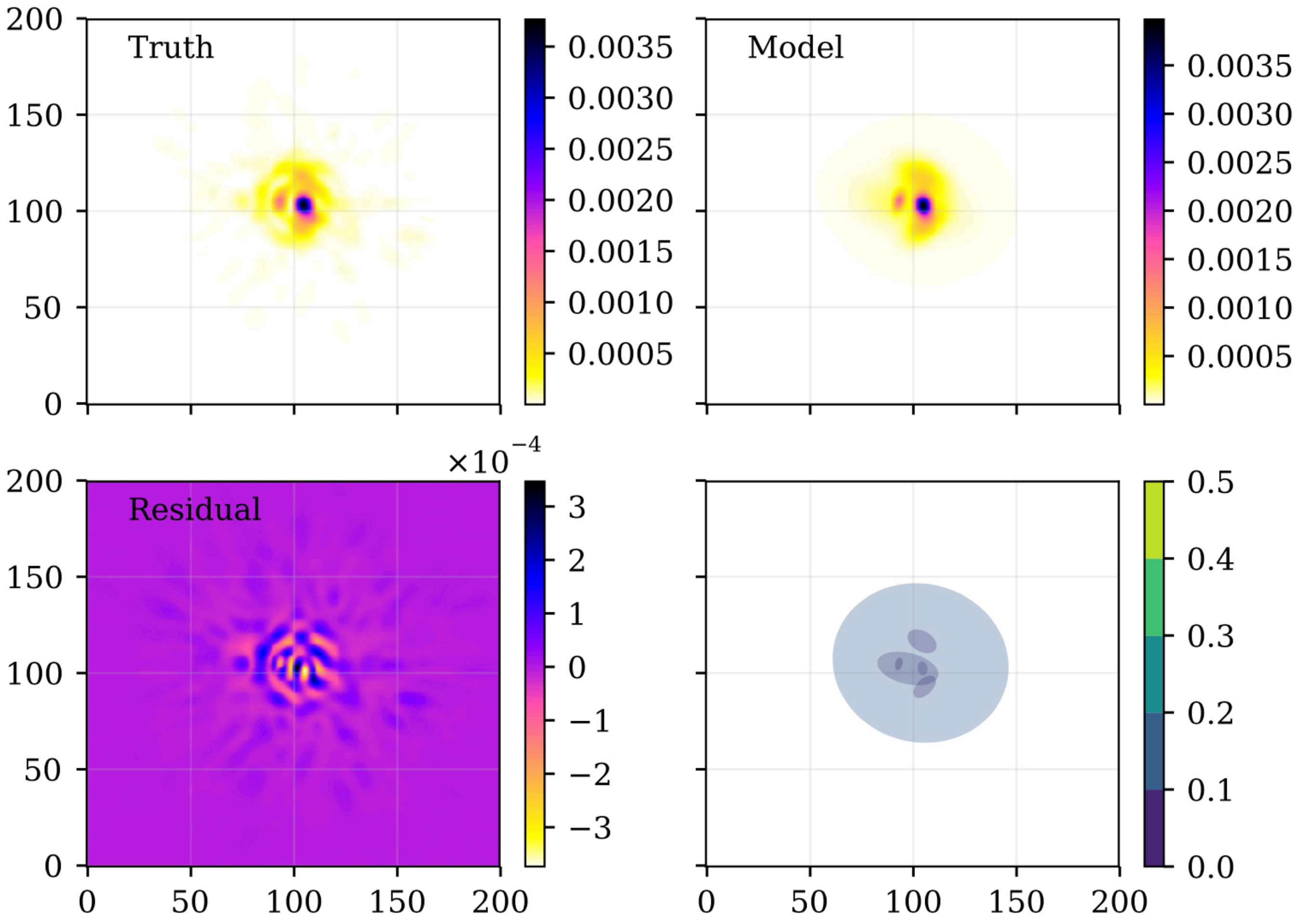
1) Optimization (least-squares or with gradients)
2) Sampling
3) Variational Inference

Forcepho: Convolution

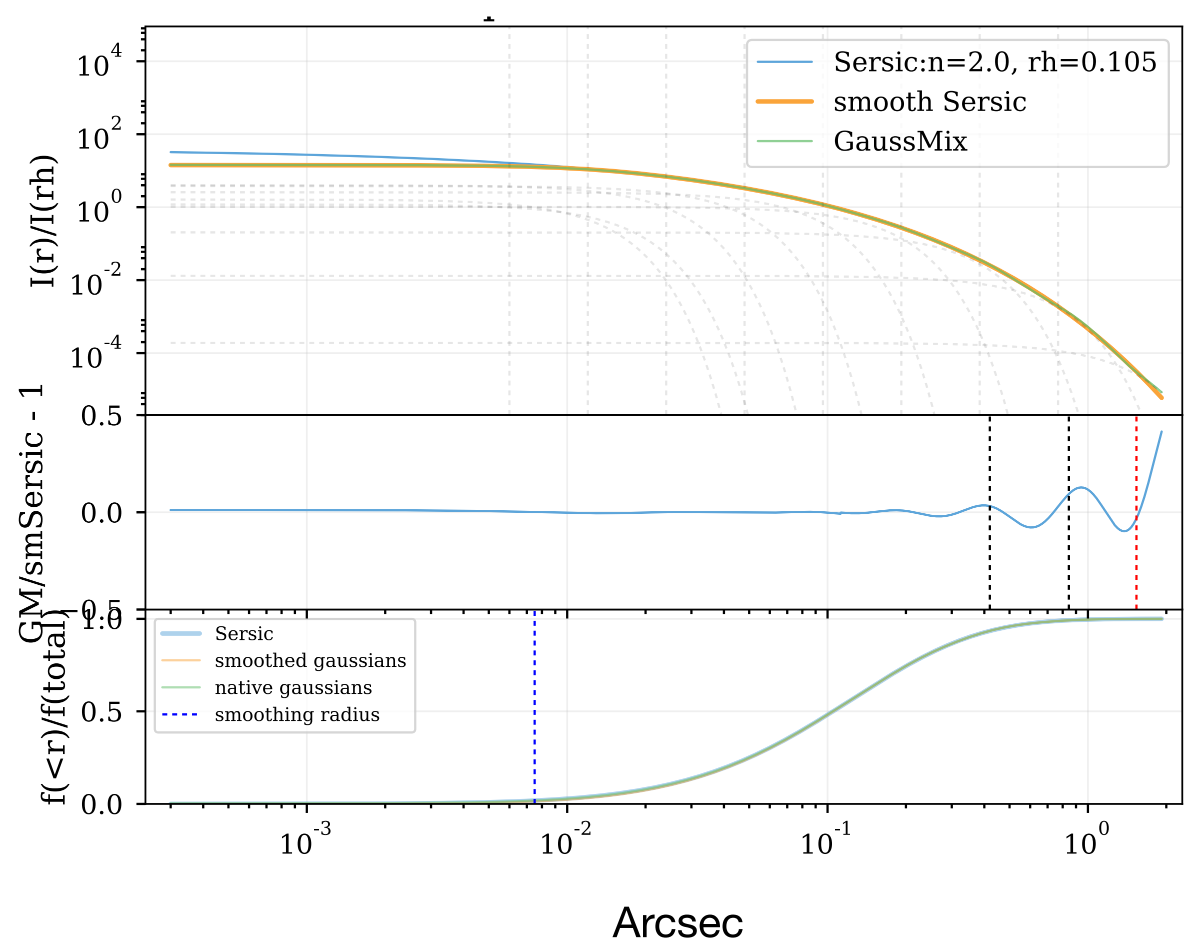
Avoid FFTs by using Gaussian Mixtures for PSF and sources (following Tractor, Lang et al.)

If necessary/desired, can replace with FFTs (gradient images also have to be FFT'd, intermediate oversampled images are necessary)

F090W PSF as a GM
(using expectation maximization)



Sersic Profiles as GM
fixed widths, amplitudes f^{n} of n_{sersic}, r_h



Forcepho: Undersampling

Common practice is to use drizzled images that restore Nyquist sampling. However, this comes at the cost of introducing covariant noise and messing with the PSF

We wanted to at least have the ability to fit to undrizzled, undersampled, individual exposures.

This can be done by creating intermediate images with smaller pixels and rebinning them, but this is slow.

We are instead using a 2nd order approximation to the integral of a gaussian over a square pixel at a particular location.

Forcepho: Galaxy Parameterization

Sersic with 6 parameters: α, δ, b/a, PA, n_{sersic}, $r_{half\text{-}light}$

Approximated by a mixture of gaussians, fit to analytic Sersic profiles (after some small smoothing well below the PSF scale)

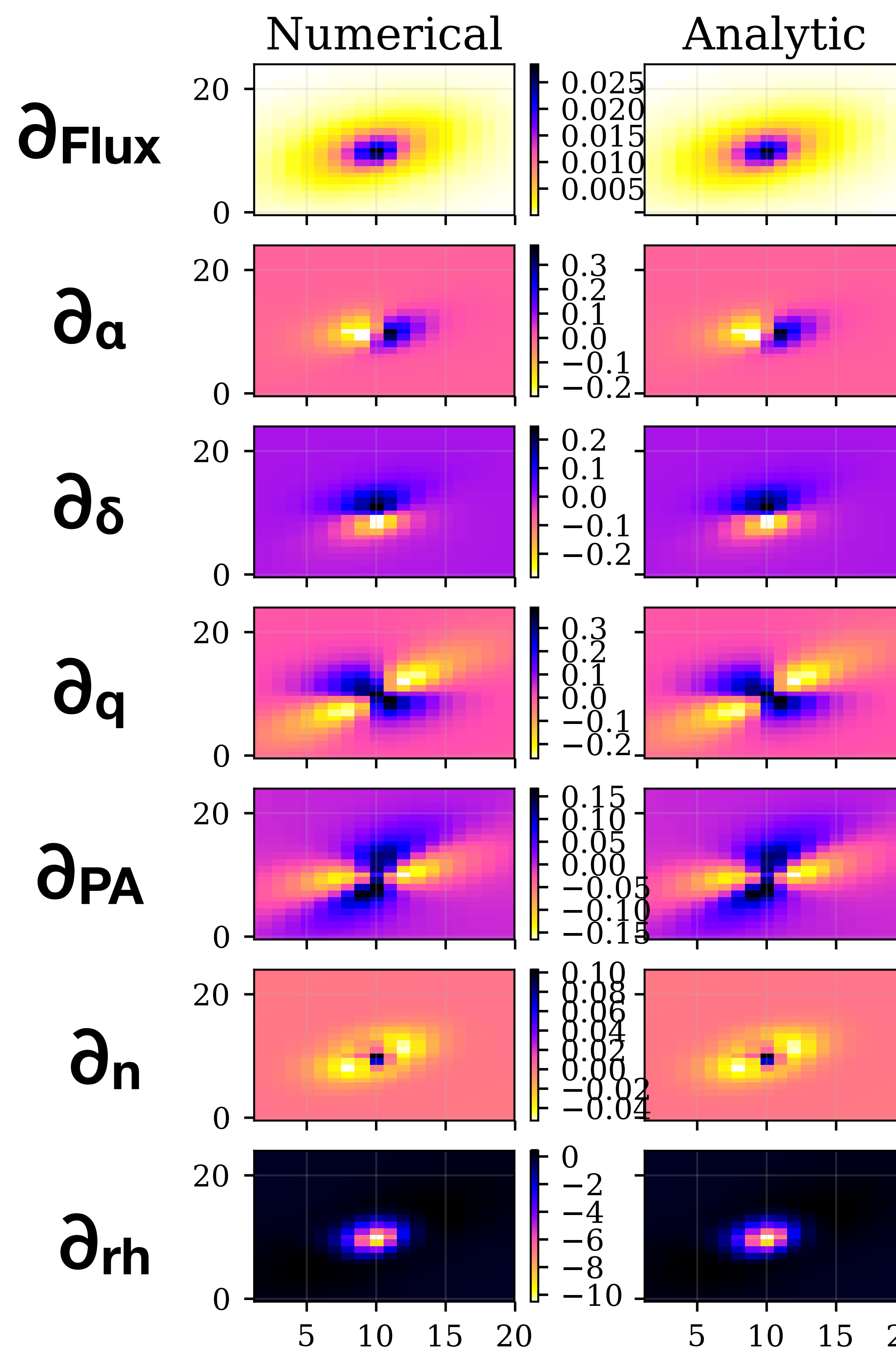
Amplitudes of gaussians are spline functions of n_{sersic}, $r_{half\text{-}light}$

Forcepho: Optimization and Sampling

We will use a χ^2 ln-likelihood

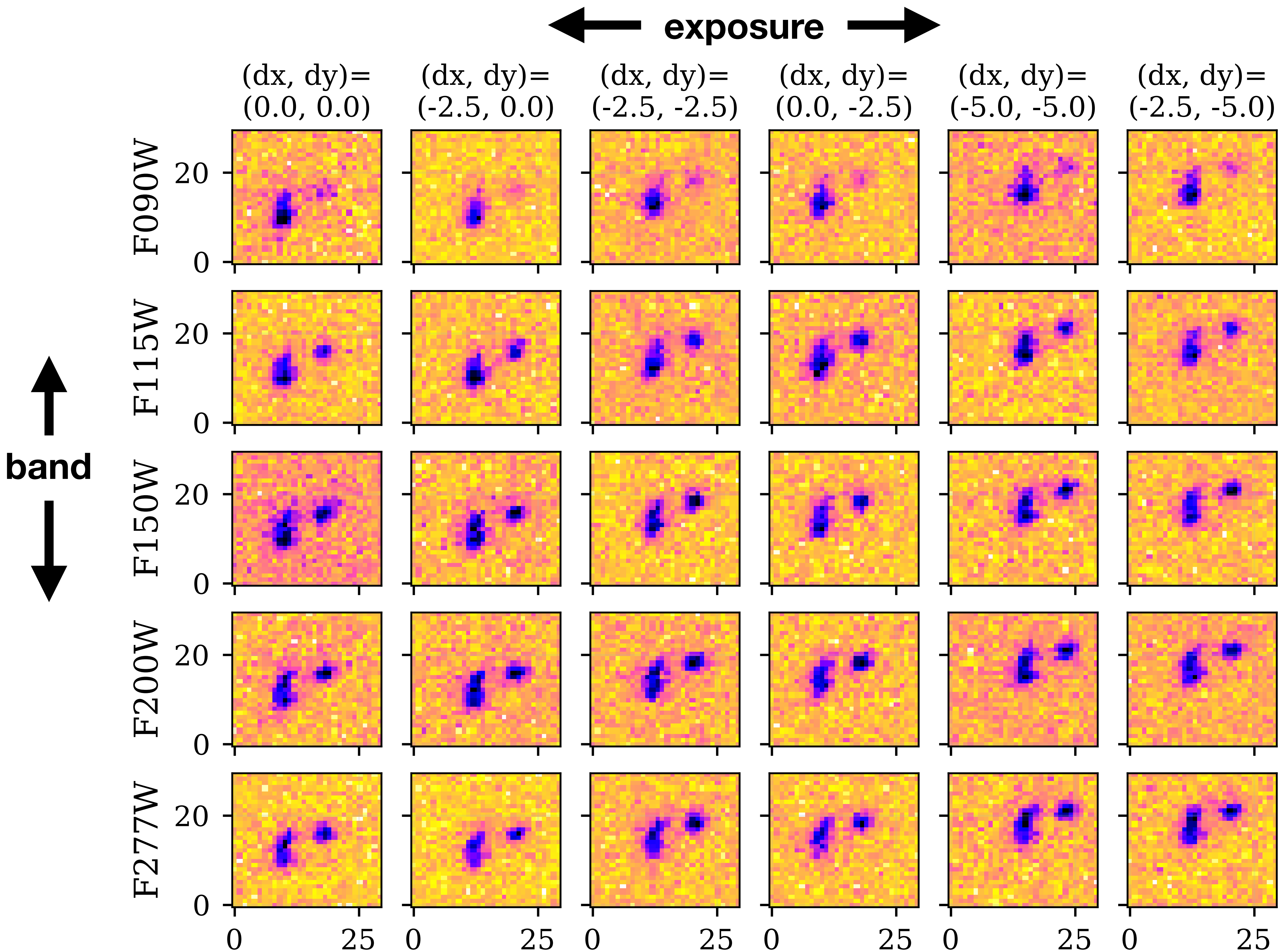
We will have many parameters:
N_{source} x (N_{band} + 6)

- **For so many parameters, typical sampling algorithms can fail**
- **We are planning to use Hamiltonian Monte Carlo, which is efficient but requires analytic ln-likelihood gradients**
- **Analytic gradients also help with optimization**
- **Accurate noise estimates are critical!!!**



Forcepho: Demo

Mock **data: 3 sources, 5 bands, 6 (offset) exposures per band. ~30k pixels**

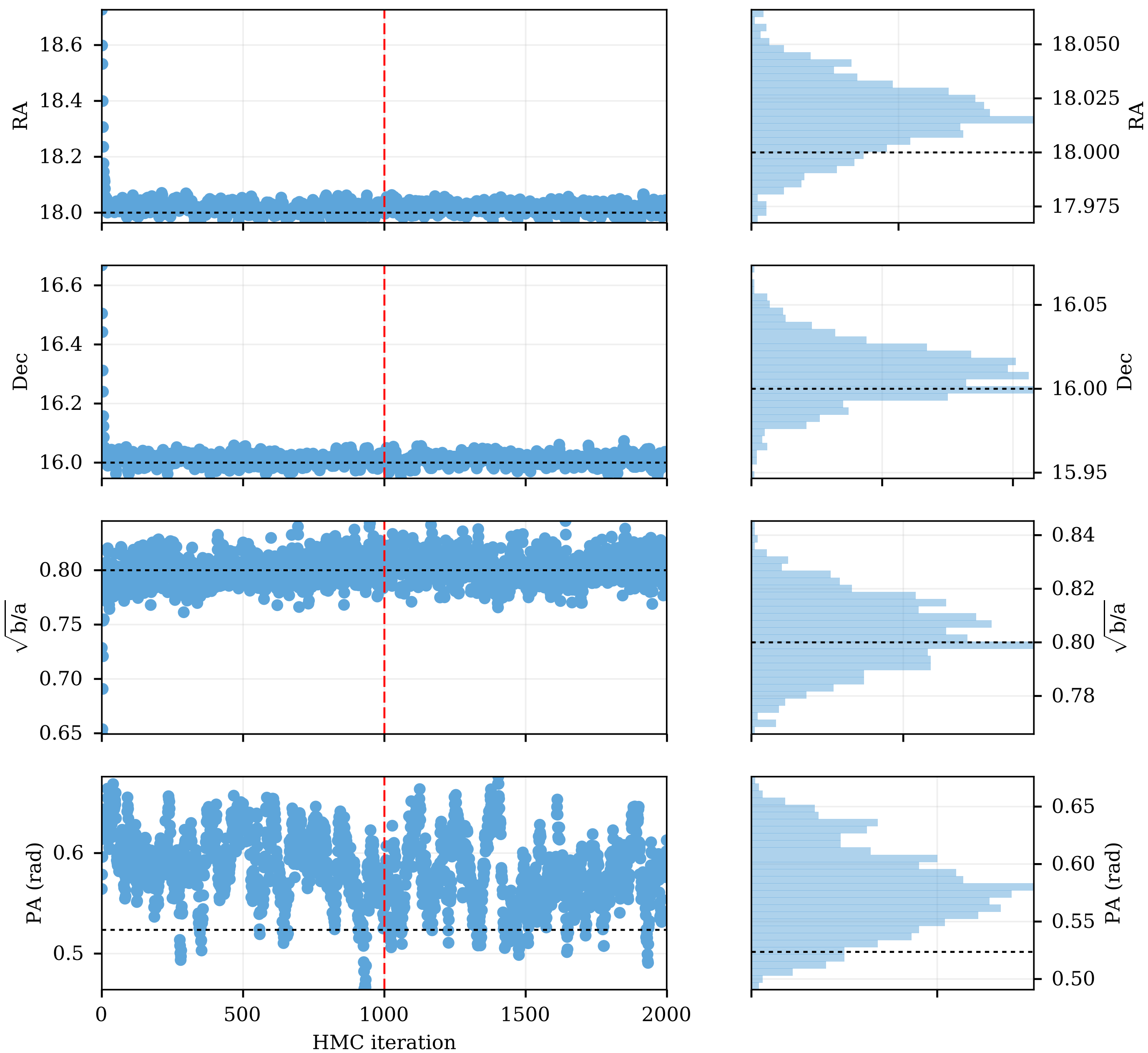


Forcepho: Demo

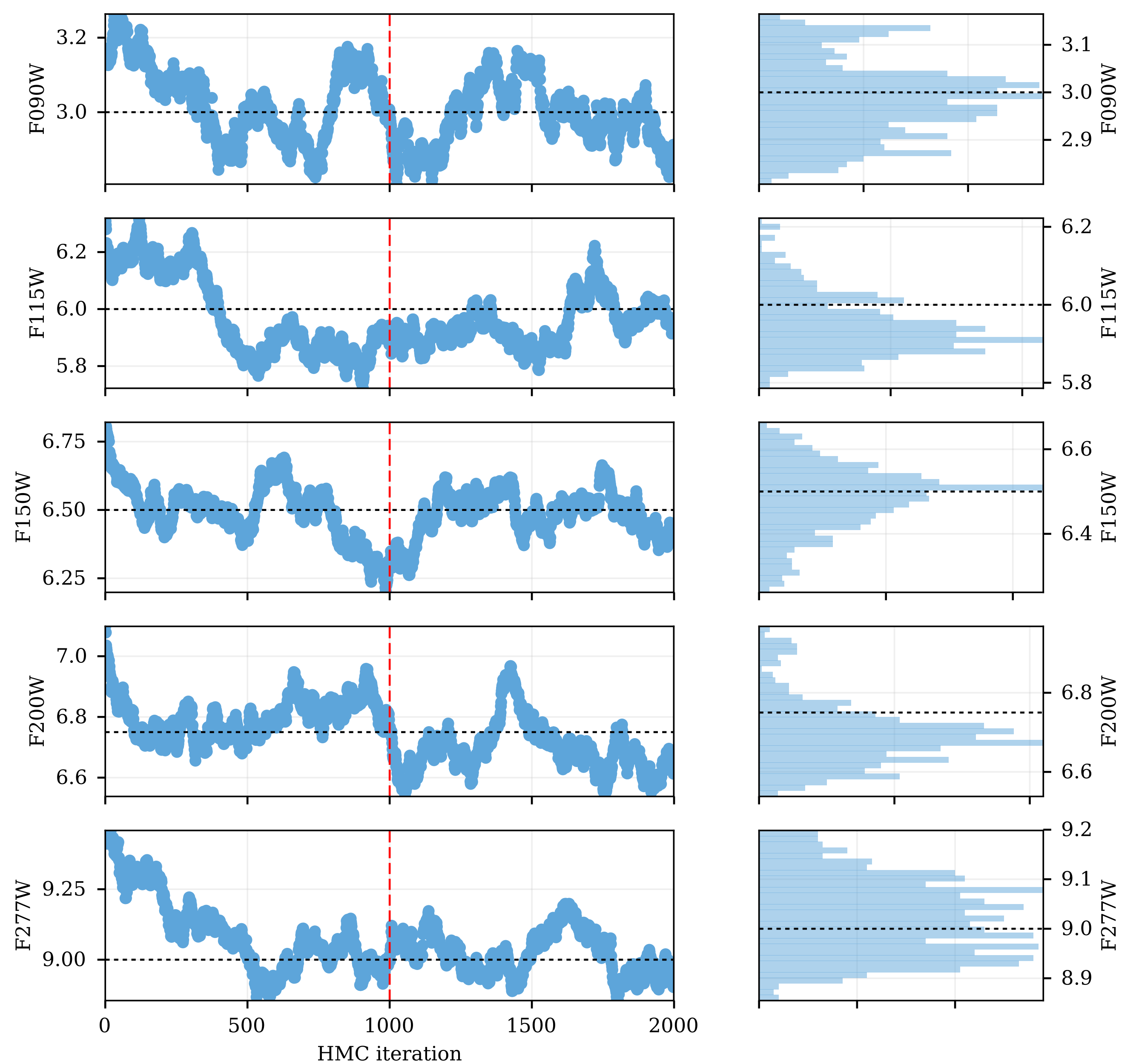
Sampling: 27 parameters total, fit simultaneously using HMC.

Only showing source #3

Shape+position parameters

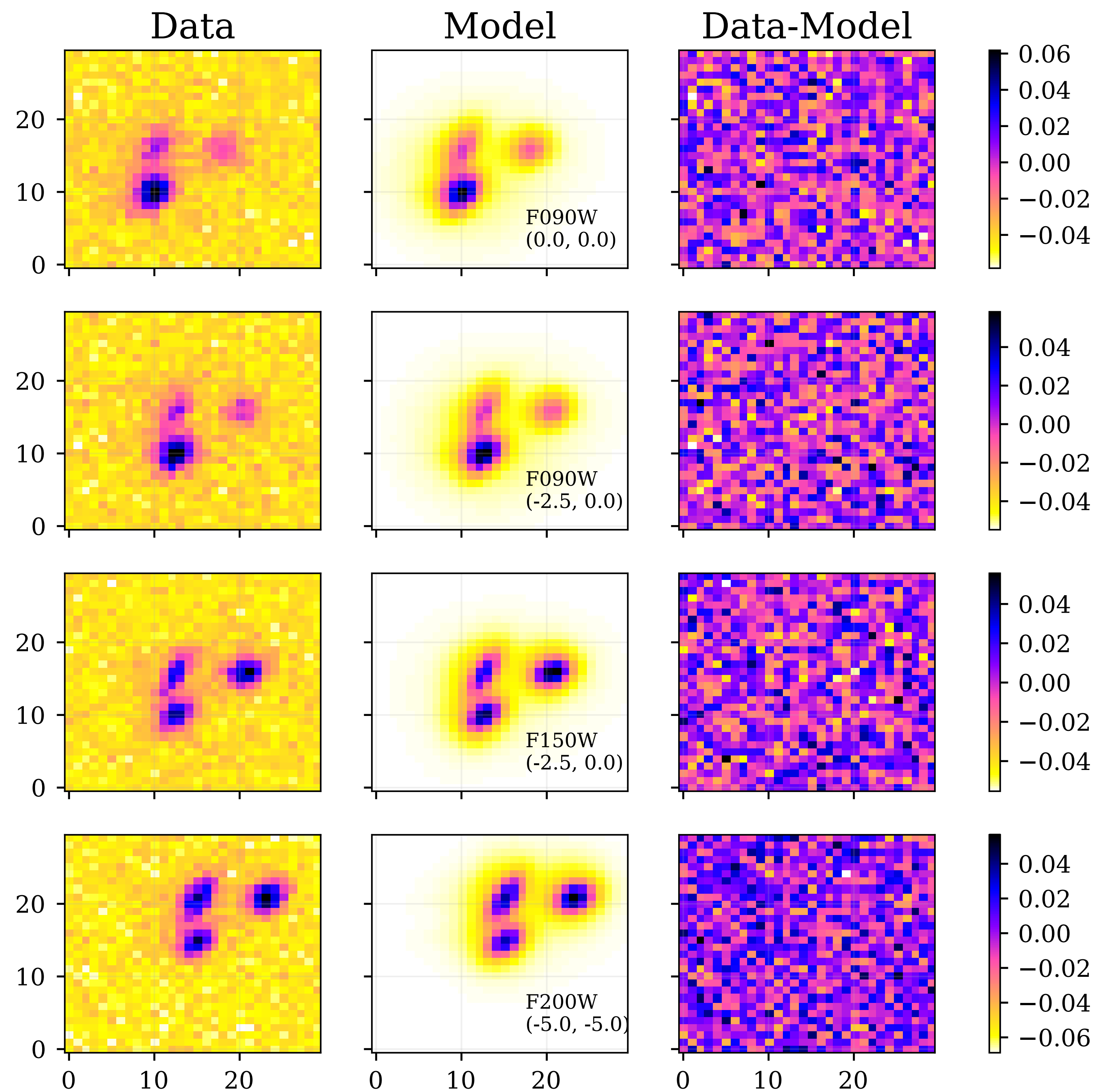


Flux parameters



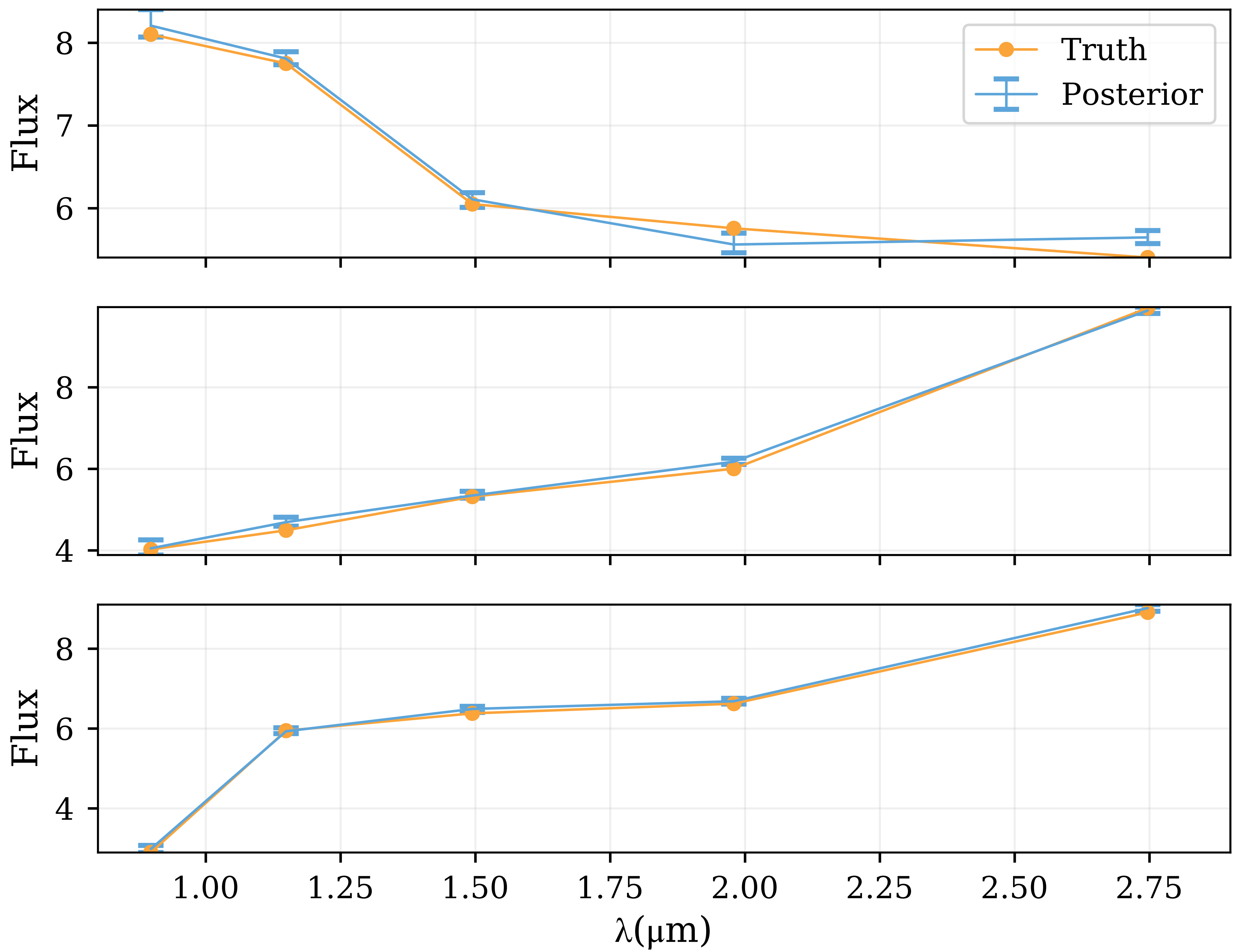
Forcepho: Demo

(Some of) mock data, model, and residuals for the last HMC iteration



Forcepho: Demo

Input and posterior SEDs for the 3 sources



Tricky points and proposed solutions

- **Speed:** Base level can be compiled/parallelized/GPU. Also be clever in choosing only relevant gaussians to compute
- **Initialization/detection:** aggressive source finding algorithm on combined stack. Spurious sources will have posteriors consistent with zero.
- **Identification (source swapping):** use positional priors to keep sources from moving too much and swapping places.
- **Source shredding/blending:** careful *a posteriori* recombination (we are punting)
- **Model mismatch: is a [sum of] Sersic's enough?** ?
- **Cosmic rays, bad pixels, junk:** ?
- **Efficient sampling:** various HMC tuning algorithms
- **Individual exposures or stacks?** hopefully individual exposures, if fast enough; obviously a challenge for DEEP

Forcepho: Status & Next Steps

Status

Working well on multi-band, multi-exposure, multi-source mock data (data generated using the model)

Some tests done with Guitarra simulated bright pointsource images:

1) **These stress test the GaussMix PSF model.**
2) **Some possible issues with simulated noise estimates, or PSF not good enough at high S/N — need to check at lower S/N**

Forcepho: Status & Next Steps

Todo

- Work out some issues with the galaxy gaussian mixtures (for n and rh)
- Improve PSF model
- Try to cut down on the number of gaussians
- Add some efficiency enhancements to the HMC sampler
- Test on simulated galaxy images
- Test on HST/UDF
- Write C++/GPU enabled base calculation code.
- Identify bottlenecks and optimize code

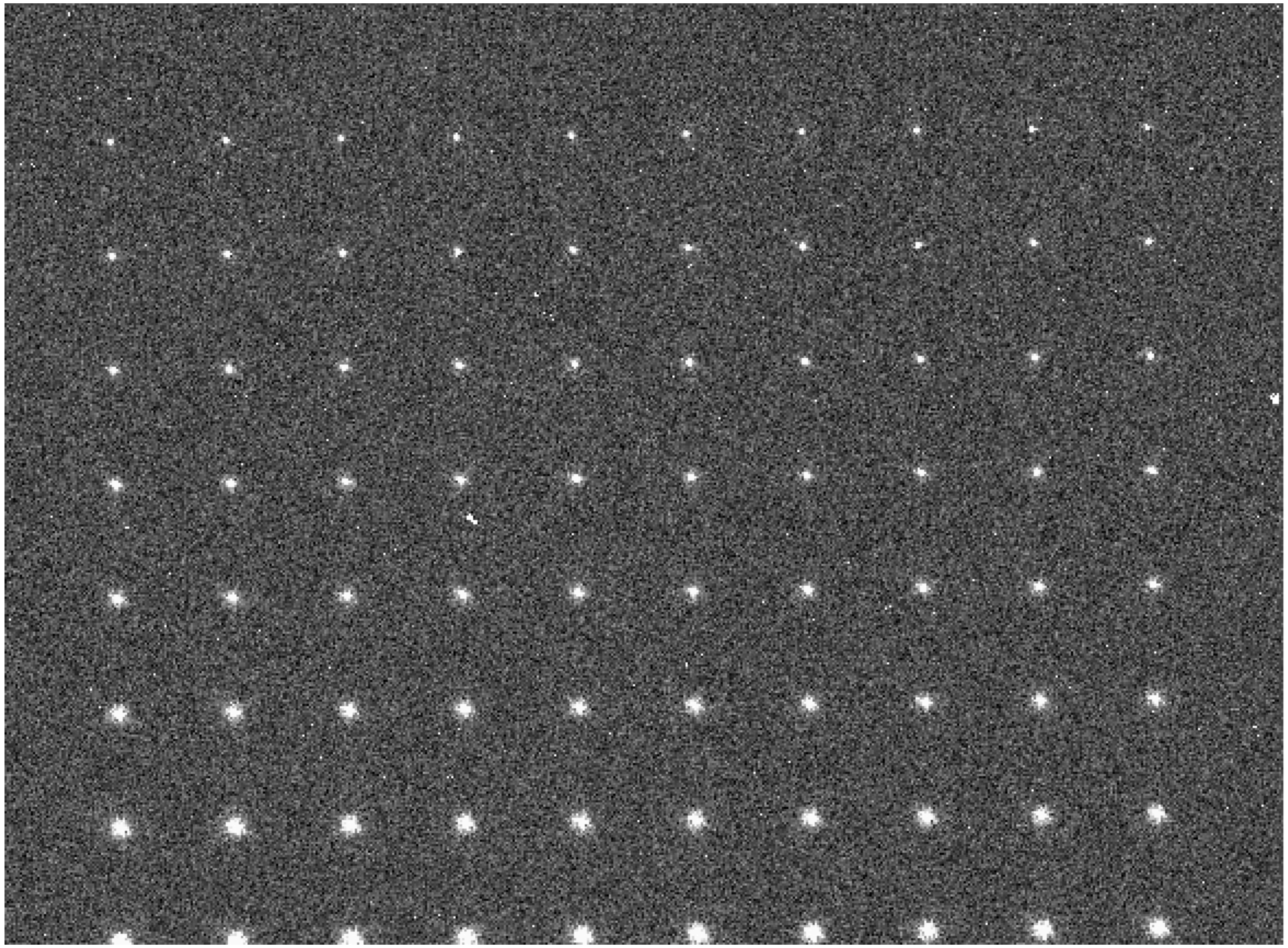
It will be important to

1) have at least one backup algorithm
2) have something like current practice

Need to liase with

1) Image simulation
2) Image analysis/stacking
3) selection

Demo on simulated point source image



Demo on simulated point source image