dLux

‘Derivatives Through Light’

A fully differentiable, open-source, optical modelling framework

Supervisors:

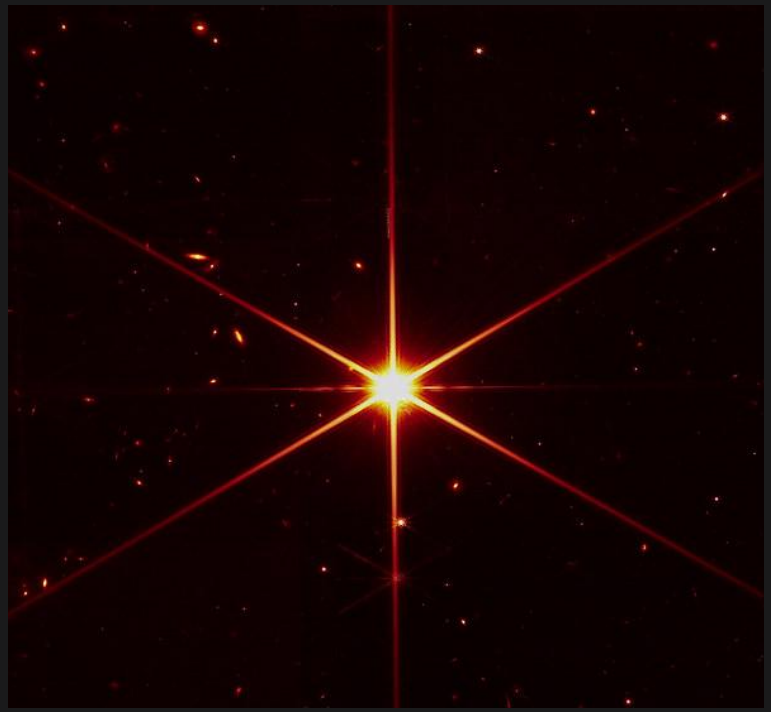
Peter Tuthill & Benjamin Pope

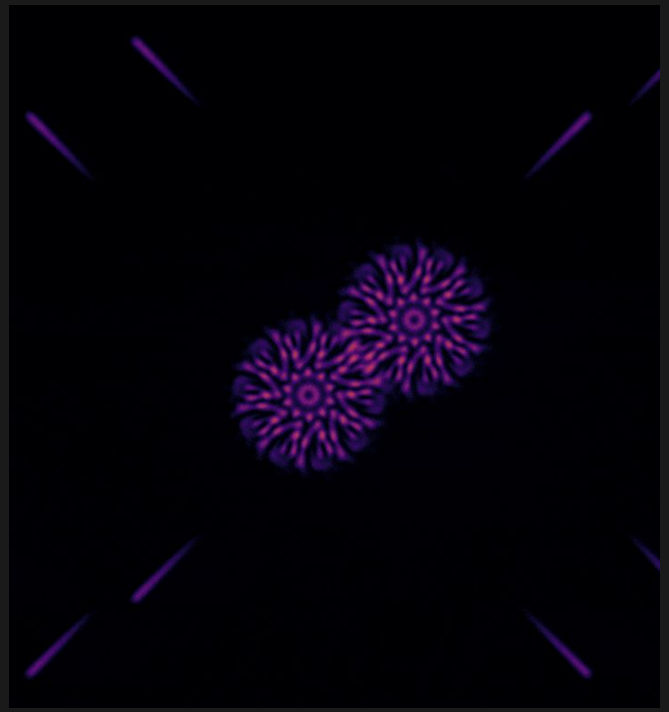
Where are we going?

- Part 1: Context & Theory
- Part 2: Instrumental Calibration
- Part 3: Optical Analysis & Design
- Part 4: Preliminary JWST Fitting

Contex & Theory

JWST & Toliman





Differentiable Optics

Optics and Neural Networks:

- Composed of the same base mathematical operations
- Defined by a series of operations on an input vector
- All operations fully differentiable

Benefits of Jax/Autodiff

- Highly efficient optimisation and inference in high dimensions
- XLA compiled at run time
- Natively deploys across GPUs
- Integrates with machine learning optimisation packages (ie Optax)

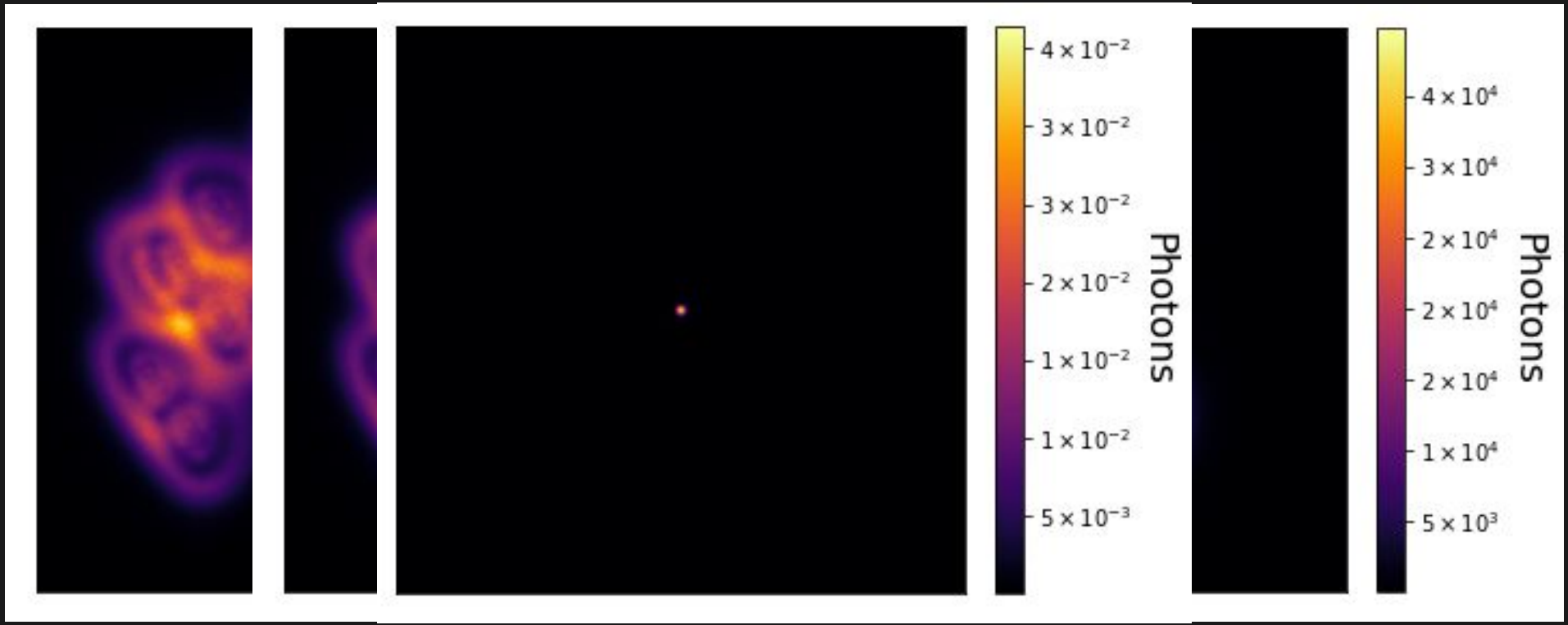
∂Lux!

Features

- Open source
- Built in Python with Jax (Numpy like API)
- Modular & object-oriented
- End to end differentiable
- Native multithreading across wavelengths/stars/images
- Integration with Neural Networks
- Fast, Flexible & Easy to use!
- Can faithfully reproduce poppy/webbpsf models

Instrumental Calibration

Modelling a Telescope



Model optimisation

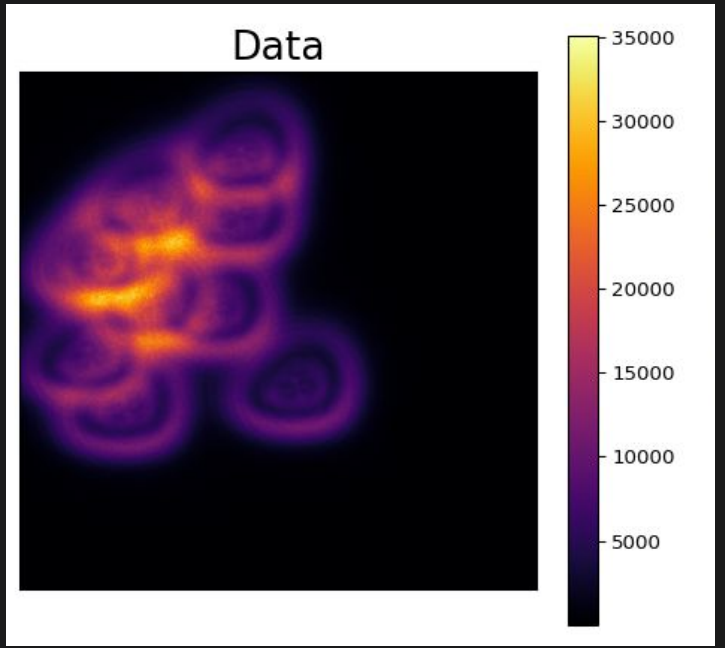
Noise Sources:

- Photon
- Detector

Optimised Parameters:

- Positions
- Fluxes
- Defocus
- Zernike Aberrations
- Pixel Response

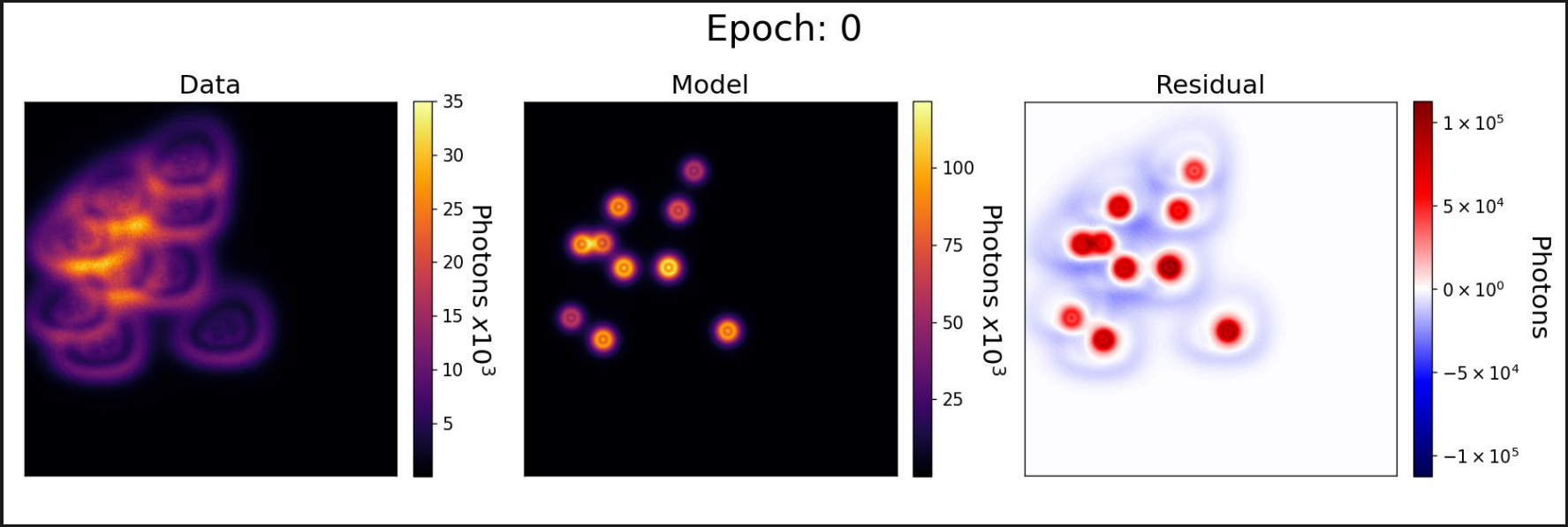
x4



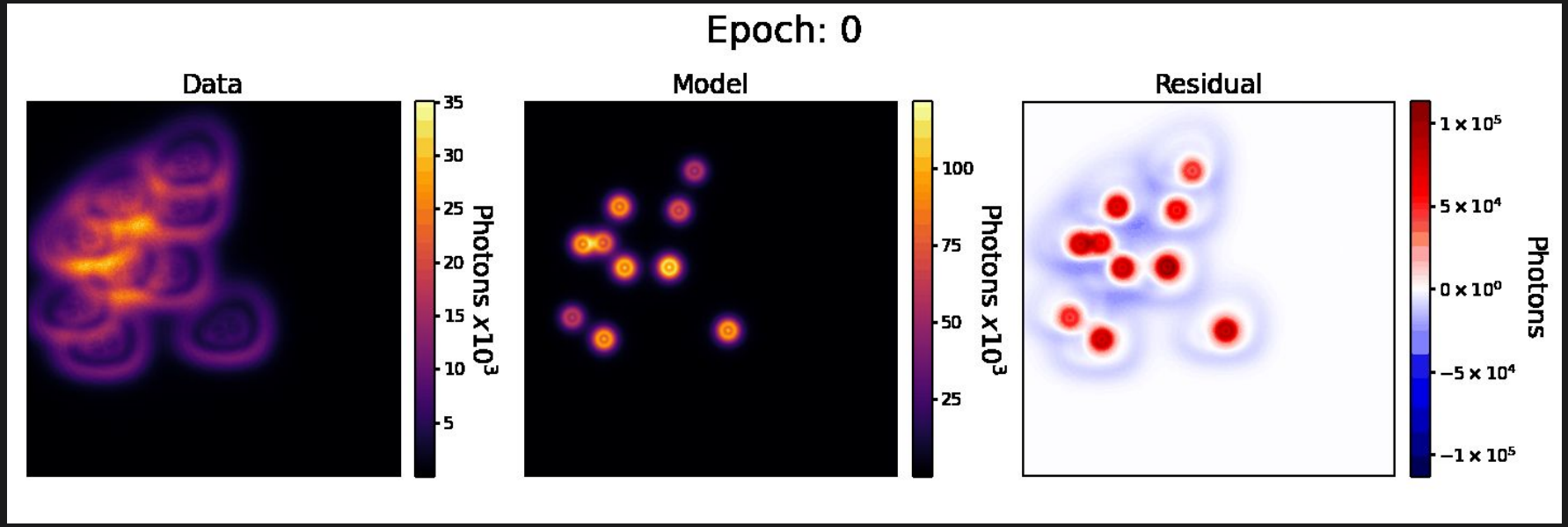
PSFs per model: 200

Total parameters: 262’181

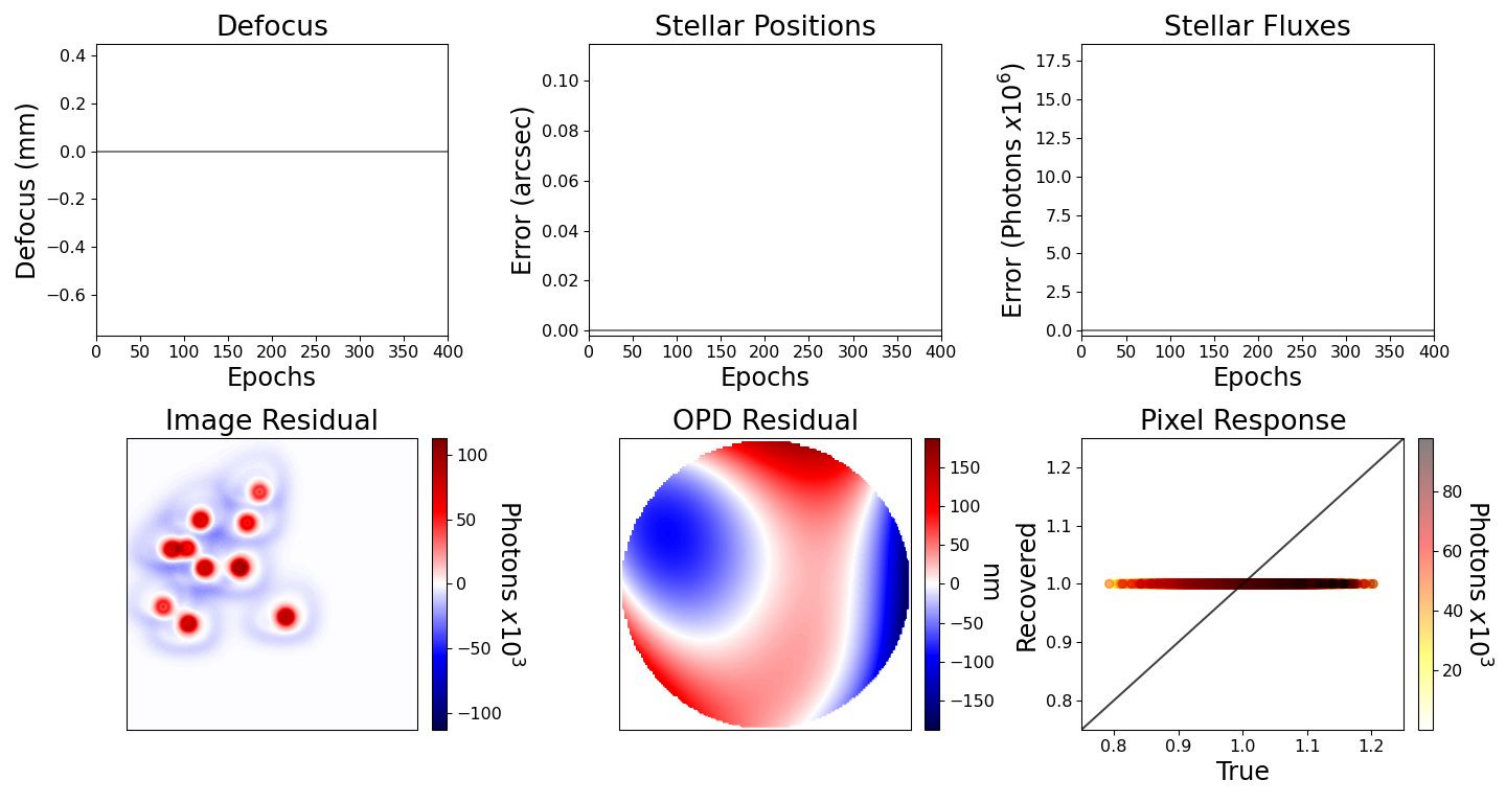
Residuals (single image)



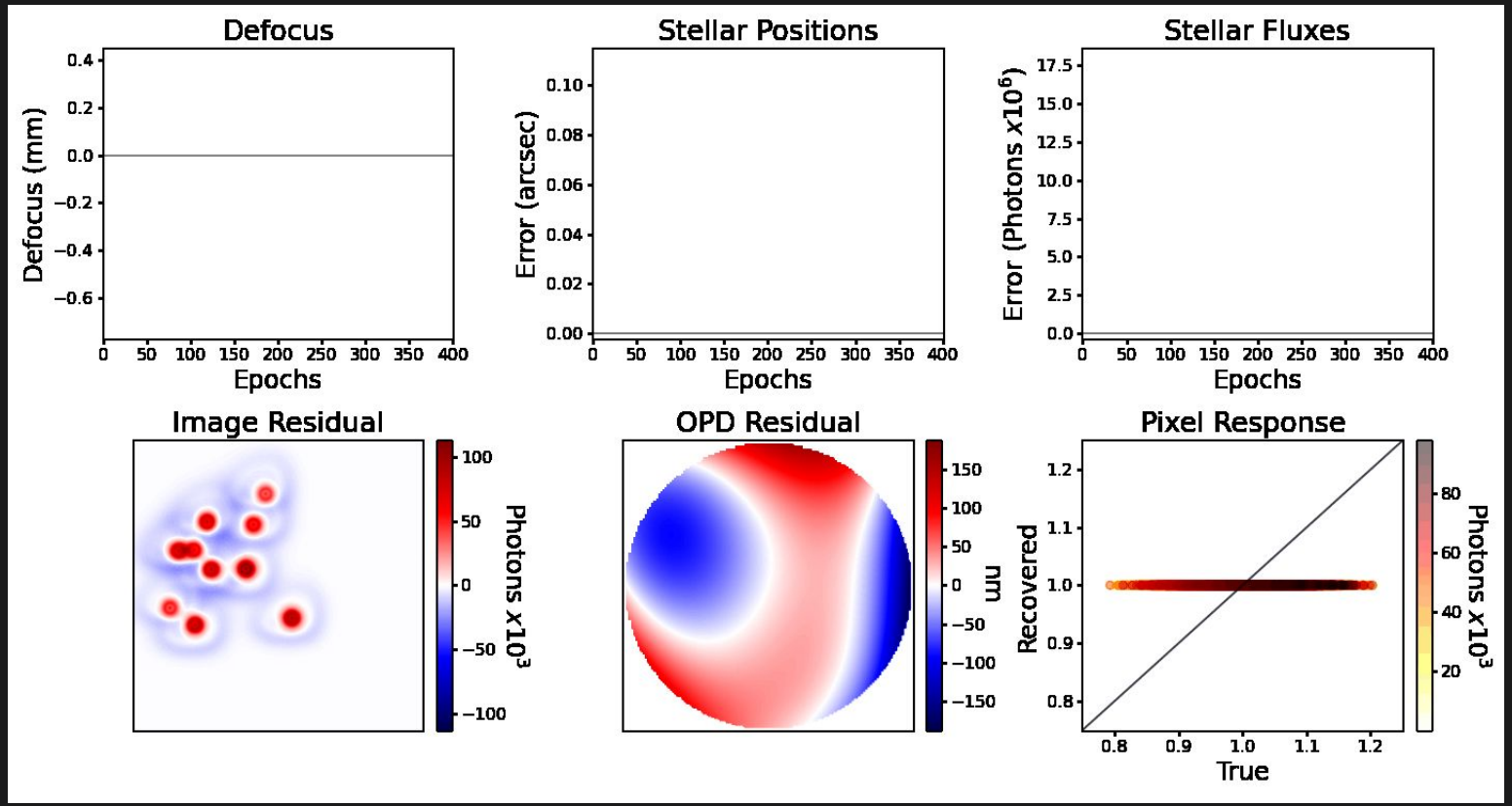
Residuals (single image)



Parameter Recovery



Parameter Recovery



Optical Design & Analysis

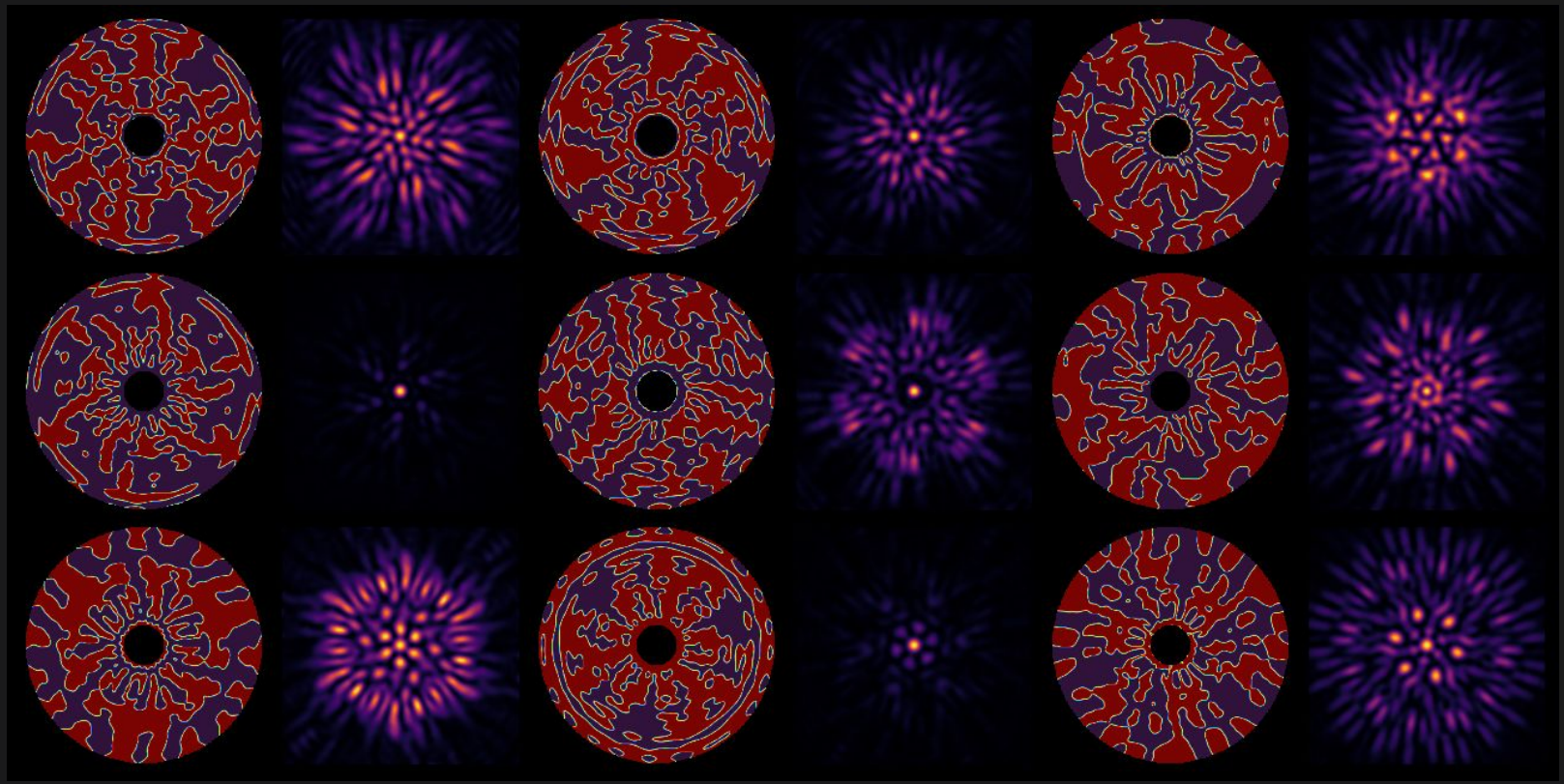
Diffractive Pupil Design

Toliman Design Goals:

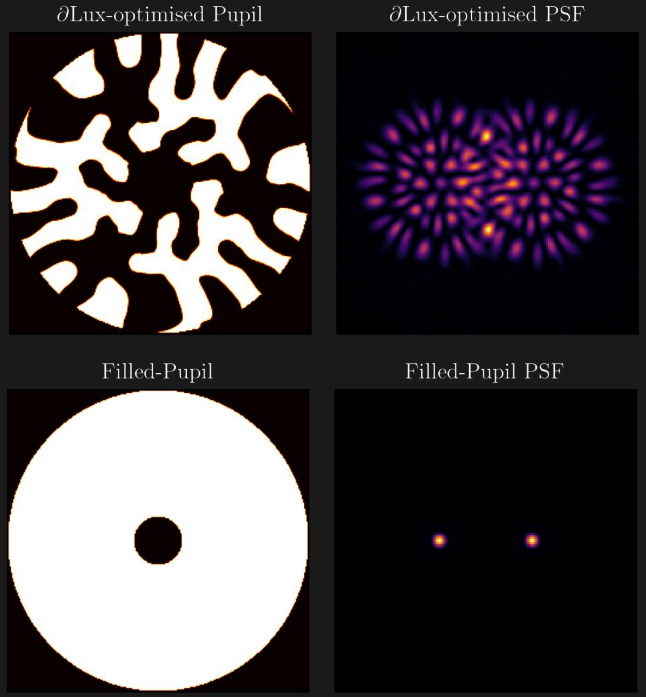
- Spread PSF into dense peaks of similar amplitude
- Overlap PSFs of at ~10 arcseconds
- Maximise positional registration of a single PSF
- Maximally constrain state of telescope

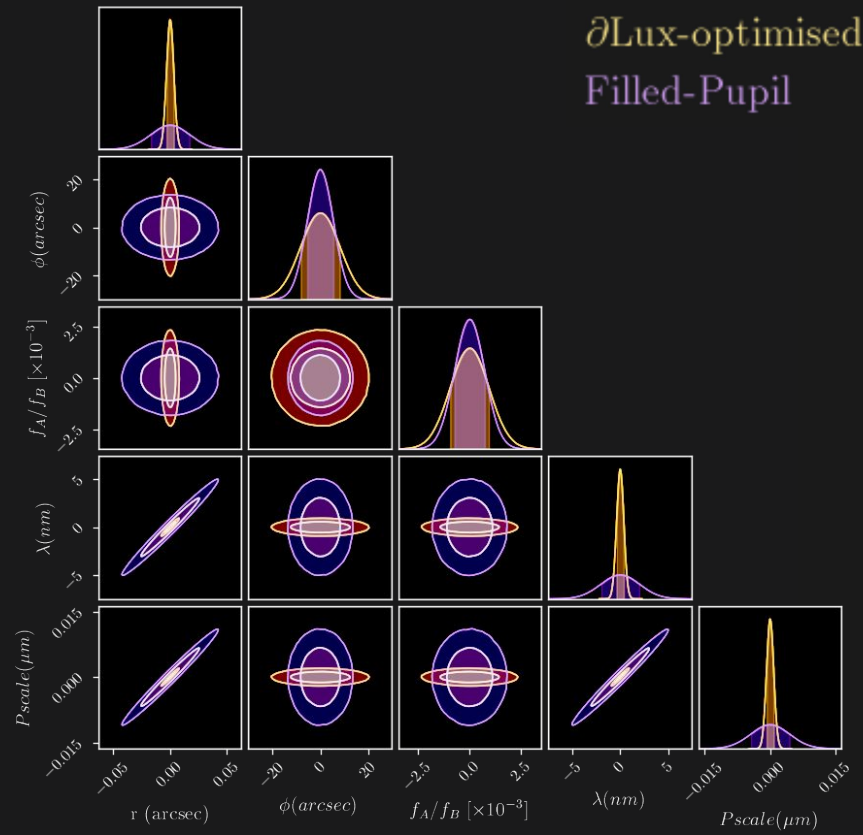
-> Maximise the PSF gradient energy!

Diffractive Pupil Design



Posterior Analysis

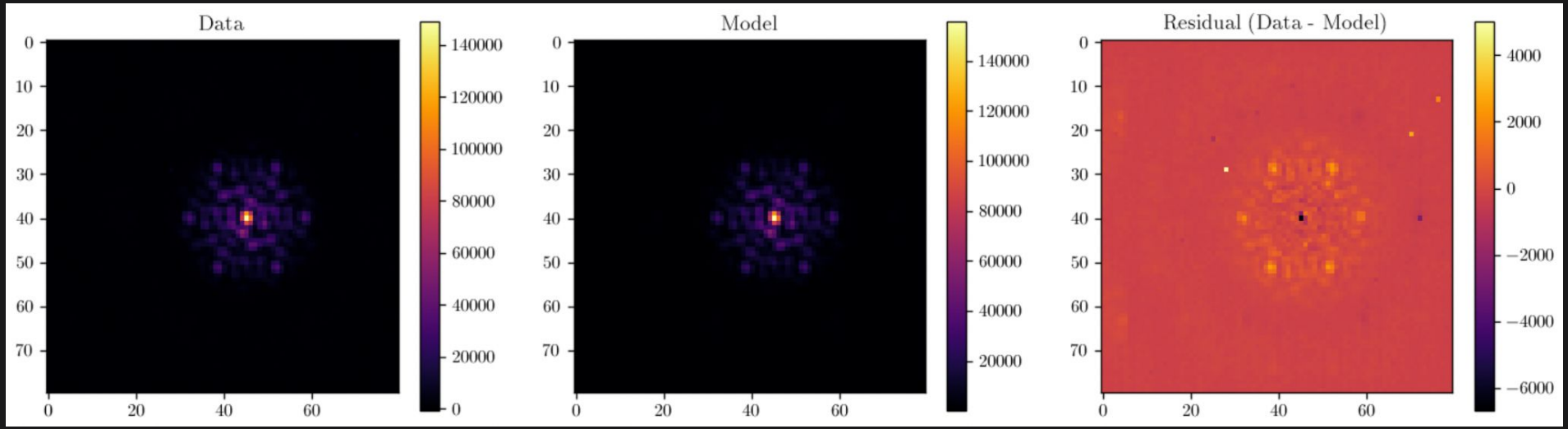




Preliminary JWST Fitting

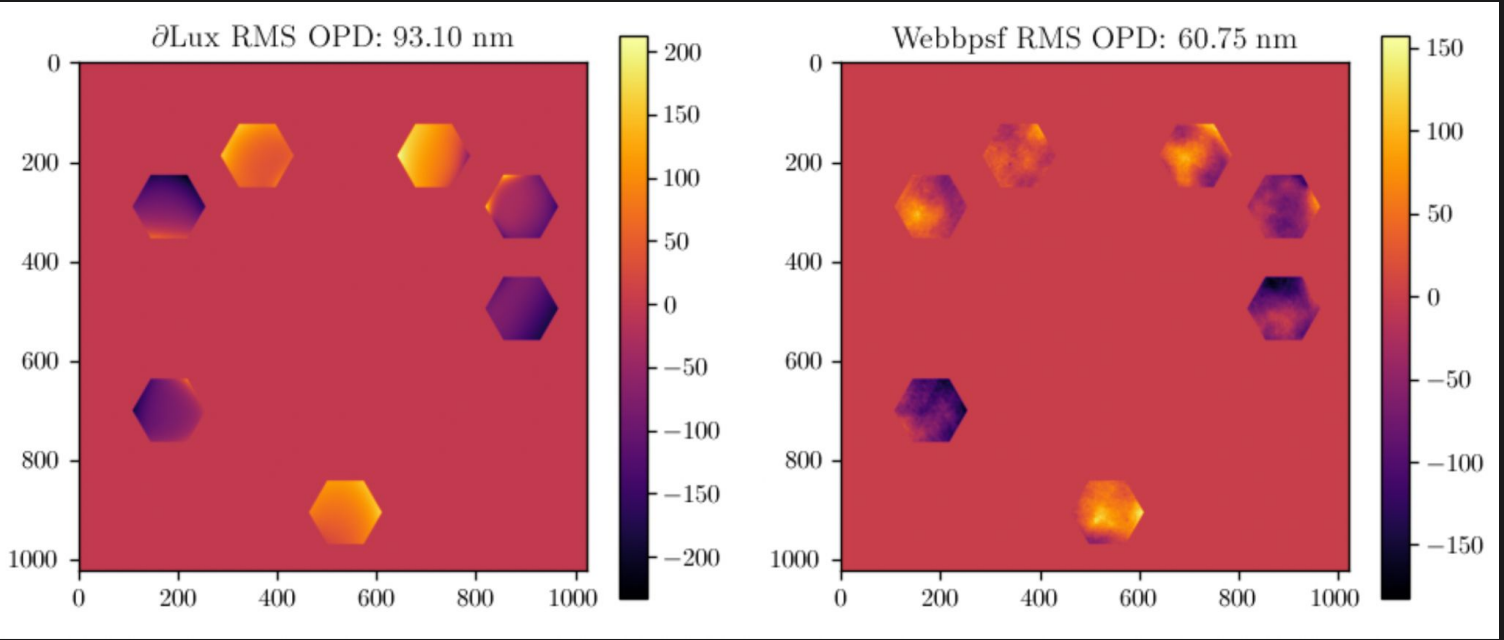
Preliminary JWST Fitting

PSF Residuals



Preliminary JWST Fitting

Phase retrieval



Questions

Contributors:

- Louis Desdoigts
- Jordan Dennis
- Benjamin Pope
- Peter Tuthill

Documentation and git:

- **‘dLux’**
- Documentation: https://louisdesdoigts.github.io/dLux/
- Github: https://github.com/LouisDesdoigts/dLux
- Contact: Louis.Desdoigts@sydney.edu.au

As open-source software, we invite all to both use and collaborate on this project!

∂Lux - Optimising a Model

1. Define Loss Function:

```
@eqx.filter_jit
@eqx.filter_value_and_grad
def loss_func(model, data):
    psfs = model[0]()
    return np.mean((psfs - data)**2)
```

2. Set Learning Rates:

```
optim = optax.multi_transform(
    {"null":        optax.adam(0.0),
     "positions":   optax.adam(2.5e-8),
     "fluxes":      optax.adam(2.5e6),
     "prop_dist":   optax.adam(5e-5),
     "coeffs":      optax.adam(5e-9),
     "FF":          optax.adam(FF_sched)},
    param_spec)
```

3. Optimise!

```
for i in tqdm(range(250)):
    loss, grads = loss_func(model_out, data)
    updates, opt_state = optim.update(grads, opt_state)
    model_out = eqx.apply_updates(model_out, updates)
```