Appendices

A - Code for the double Sérsic light profile fitting analysis
B – Code for the single Sérsic light profile fitting analysis
C – Simulated galaxy, “bar_0”, subplot dataset
D – Simulated galaxy, “bar_1”, subplot dataset
E – Simulated galaxy, “no_bar”, subplot dataset
F – JWST data, “dataset_3”, subplot dataset
G – JWST data, “dataset_4”, subplot dataset
H – JWST data, “dataset_6”, subplot dataset
I – Double Sérsic fit to JWST “dataset_3” subplot
J – Single Sérsic fit to JWST “dataset_3” subplot
K – Double Sérsic fit to JWST “dataset_4” subplot
L – Single Sérsic fit to JWST “dataset_4” subplot
M – Double Sérsic fit to JWST “dataset_6” subplot
N – Single Sérsic fit to JWST “dataset_6” subplot

A - Code for the double Sérsic light profile fitting analysis

```
!pip install autoconf==2024.11.13.2 autofit==2024.11.13.2 autoarray==2024.11.13.2 a
!pip install numba
!git clone https://github.com/Jammy2211/BSc_Galaxies_Project
import os
from autoconf import conf
os.chdir("/content/BSc_Galaxies_Project")
conf.instance.push(
new_path="/content/BSc_Galaxies_Project/",
output_path="/content/BSc_Galaxies_Project/output"
,
)
import numpy as np
import autogalaxy as ag
import autogalaxy.plot as aplt
from os import path
import autofit as af
import autofit as af
dataset_name = "" #enter required dataset here
dataset_path = path.join("dataset", "task_6a_bars", dataset_name)
dataset = ag.Imaging.from_fits(
data_path=path.join(dataset_path, "data.fits"),
noise_map_path=path.join(dataset_path, "noise_map.fits"),
psf_path=path.join(dataset_path, "psf.fits"),
pixel_scales=0.06,
)
dataset_plotter = aplt.ImagingPlotter(dataset=dataset)
dataset_plotter.subplot_dataset()
# Masking
mask = ag.Mask2D.circular(
shape_native=dataset.shape_native, pixel_scales=dataset.pixel_scales, radius=6.
)
dataset = dataset.apply_mask(mask=mask)
dataset_plotter = aplt.ImagingPlotter(dataset=dataset)
dataset_plotter.subplot_dataset()
# Galaxy
# Disk component
disk = ag.lp.Sersic(
centre=(0.0, 0.0),
ell_comps=(0.0, 0.0),
intensity=0.01,
effective_radius=6.0,
```

```
sersic_index=3.0,
)
# Bulge component
bulge = ag.lp.Sersic(
centre=(0.0, 0.0),
ell_comps=(0.4, 0.01),
intensity=3.00,
effective_radius=2.0,
sersic_index=0.5,
upp
upp
)
# Galaxy model
galaxy_model = af.Model(ag.Galaxy, redshift=0.5, disk=ag.lp_linear.Sersic, bulge=ag
model = af.Collection(galaxies=af.Collection(galaxy=galaxy_model))
print(model.info)
# Priors
#Priors disk
model.galaxies.galaxy.disk.centre.centre_0 = af.UniformPrior(lower_limit=-0.01,
model.galaxies.galaxy.disk.centre.centre_1 = af.UniformPrior(lower_limit=-0.01,
model.galaxies.galaxy.disk.ell_comps.ell_comps_0 = af.UniformPrior(
lower_limit=-0.1, upper_limit=0.1
)
model.galaxies.galaxy.disk.ell_comps.ell_comps_1 = af.UniformPrior(
lower_limit=-0.1, upper_limit=0.1
)
model.galaxies.galaxy.disk.effective_radius = af.UniformPrior(
lower_limit=0.0, upper_limit=6.0
)
model.galaxies.galaxy.disk.sersic_index = af.UniformPrior(
lower_limit=0.0, upper_limit=2.0
)
# Priors bulge
model.galaxies.galaxy.bulge.centre.centre_0 = af.UniformPrior(lower_limit=-0.1,
model.galaxies.galaxy.bulge.centre.centre_1 = af.UniformPrior(lower_limit=-0.1,
model.galaxies.galaxy.bulge.ell_comps.ell_comps_0 = af.UniformPrior(
lower_limit=-1.0, upper_limit=1.0
)
model.galaxies.galaxy.bulge.ell_comps.ell_comps_1 = af.UniformPrior(
lower_limit=-1.0, upper_limit=1.0
)
model.galaxies.galaxy.bulge.effective_radius = af.UniformPrior(
lower_limit=0.0, upper_limit=6.0
)
model.galaxies.galaxy.bulge.sersic_index = af.UniformPrior(
lower_limit=0.50, upper_limit=0.51
)
upp
uppprint(model.info)
#Analysis
analysis = ag.AnalysisImaging(dataset=dataset)
# Nested sampling algorthm
search = af.Nautilus(
name="example_0",
n_live=1000,
iterations_per_update=3000,
force_x1_cpu=True # This ensures the Google Colab code runs correctly
)
print(
"""
The non-linear search has begun running.
This Jupyter notebook cell with progress once the search has completed
"""
)
result = search.fit(model=model, analysis=analysis)
print("The search has finished run - you may now continue the notebook.")
print(result.info)
fit_plotter = aplt.FitImagingPlotter(fit=result.max_log_likelihood_fit)
fit_plotter.subplot_fit()
```

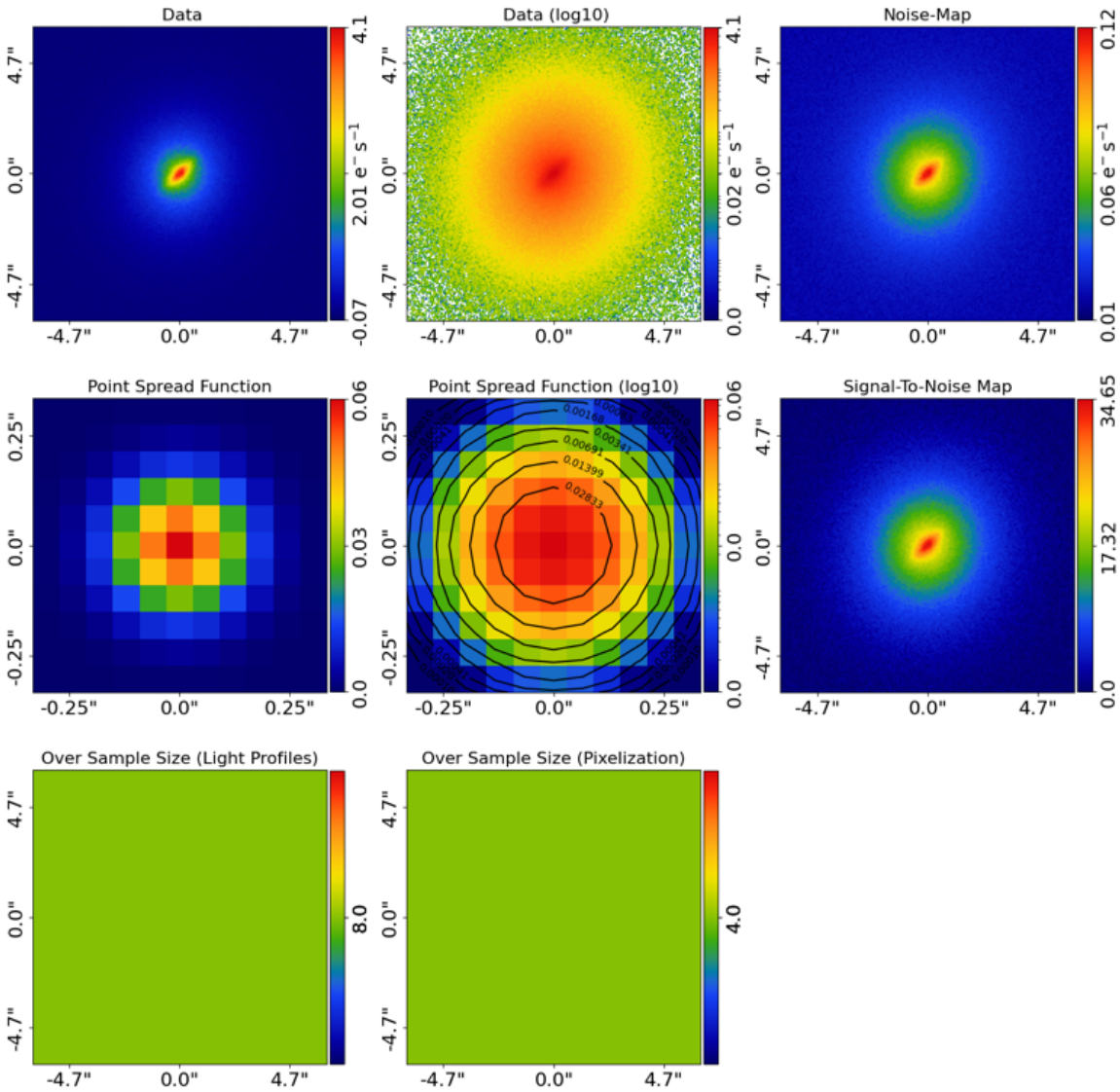
```
from google.colab import files
!zip -r output.zip /content/BSc_Galaxies_Project/output
files.download("output.zip")
files.download("output.zip")
from google.colab import files
!zip -r output.zip /content/BSc_Galaxies_Project/output
files.download("output.zip")
```

B – Code for the single Sérsic light profile fitting analysis

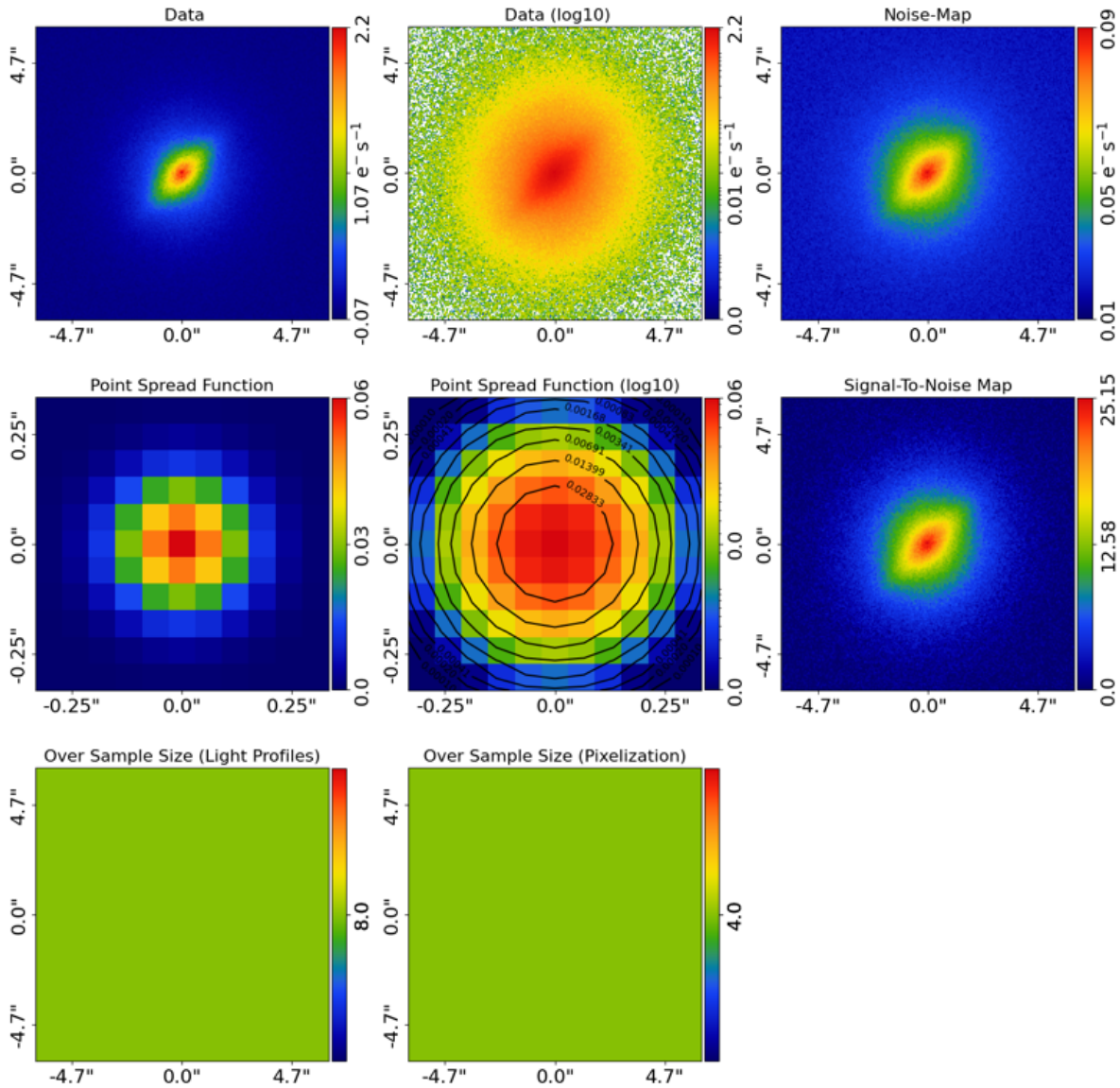
```
!pip install autoconf==2024.11.13.2 autofit==2024.11.13.2 autoarray==2024.11.13.2 a
!pip install numba
!git clone https://github.com/Jammy2211/BSc_Galaxies_Project
import os
from autoconf import conf
os.chdir("/content/BSc_Galaxies_Project")
conf.instance.push(
new_path="/content/BSc_Galaxies_Project/",
output_path="/content/BSc_Galaxies_Project/output"
,
)
import numpy as np
import autogalaxy as ag
import autogalaxy.plot as aplt
from os import path
import autofit as af
import autofit as af
dataset_name = "" #enter required dataset here
dataset_name = "" #enter required dataset here
dataset_path = path.join("dataset", "task_6a_bars", dataset_name)
dataset = ag.Imaging.from_fits(
data_path=path.join(dataset_path, "data.fits"),
noise_map_path=path.join(dataset_path, "noise_map.fits"),
psf_path=path.join(dataset_path, "psf.fits"),
pixel_scales=0.06,
)
dataset_plotter = aplt.ImagingPlotter(dataset=dataset)
dataset_plotter.subplot_dataset()
# Masking
mask = ag.Mask2D.circular(
shape_native=dataset.shape_native, pixel_scales=dataset.pixel_scales, radius=6.
)
dataset = dataset.apply_mask(mask=mask)
dataset_plotter = aplt.ImagingPlotter(dataset=dataset)
dataset_plotter.subplot_dataset()
# Galaxy
# Disk component
disk = ag.lp.Sersic(
centre=(0.0, 0.0),
ell_comps=(0.0, 0.0),
intensity=0.01,intensity=0.01,
effective_radius=6.0,
sersic_index=3.0,
)
# Galaxy model
galaxy_model = af.Model(ag.Galaxy, redshift=0.5, disk=ag.lp_linear.Sersic)
model = af.Collection(galaxies=af.Collection(galaxy=galaxy_model))
print(model.info)
# Priors
#Priors disk
model.galaxies.galaxy.disk.centre.centre_0 = af.UniformPrior(lower_limit=-0.01,
model.galaxies.galaxy.disk.centre.centre_1 = af.UniformPrior(lower_limit=-0.01,
model.galaxies.galaxy.disk.ell_comps.ell_comps_0 = af.UniformPrior(
lower_limit=-0.1, upper_limit=0.1
)
model.galaxies.galaxy.disk.ell_comps.ell_comps_1 = af.UniformPrior(
lower_limit=-0.1, upper_limit=0.1
)
```

```
model.galaxies.galaxy.disk.effective_radius = af.UniformPrior(
lower_limit=0.0, upper_limit=6.0
)
model.galaxies.galaxy.disk.sersic_index = af.UniformPrior(
lower_limit=0.0, upper_limit=2.0
)
print(model.info)
#Analysis
analysis = ag.AnalysisImaging(dataset=dataset)
# Nested sampling algorthm
search = af.Nautilus(
name="example_0",
n_live=300,
iterations_per_update=1000,
force_x1_cpu=True # This ensures the Google Colab code runs correctly
upp
upp
)
print(
"""
The non-linear search has begun running.
This Jupyter notebook cell with progress once the search has completed - this c
"""
)
result = search.fit(model=model, analysis=analysis)
print("The search has finished run - you may now continue the
notebook.")print(result.info)
fit_plotter = aplt.FitImagingPlotter(fit=result.max_log_likelihood_fit)
fit_plotter.subplot_fit()
files.download("output.zip")
from google.colab import files
!zip -r output.zip /content/BSc_Galaxies_Project/output
files.download("output.zip")
```

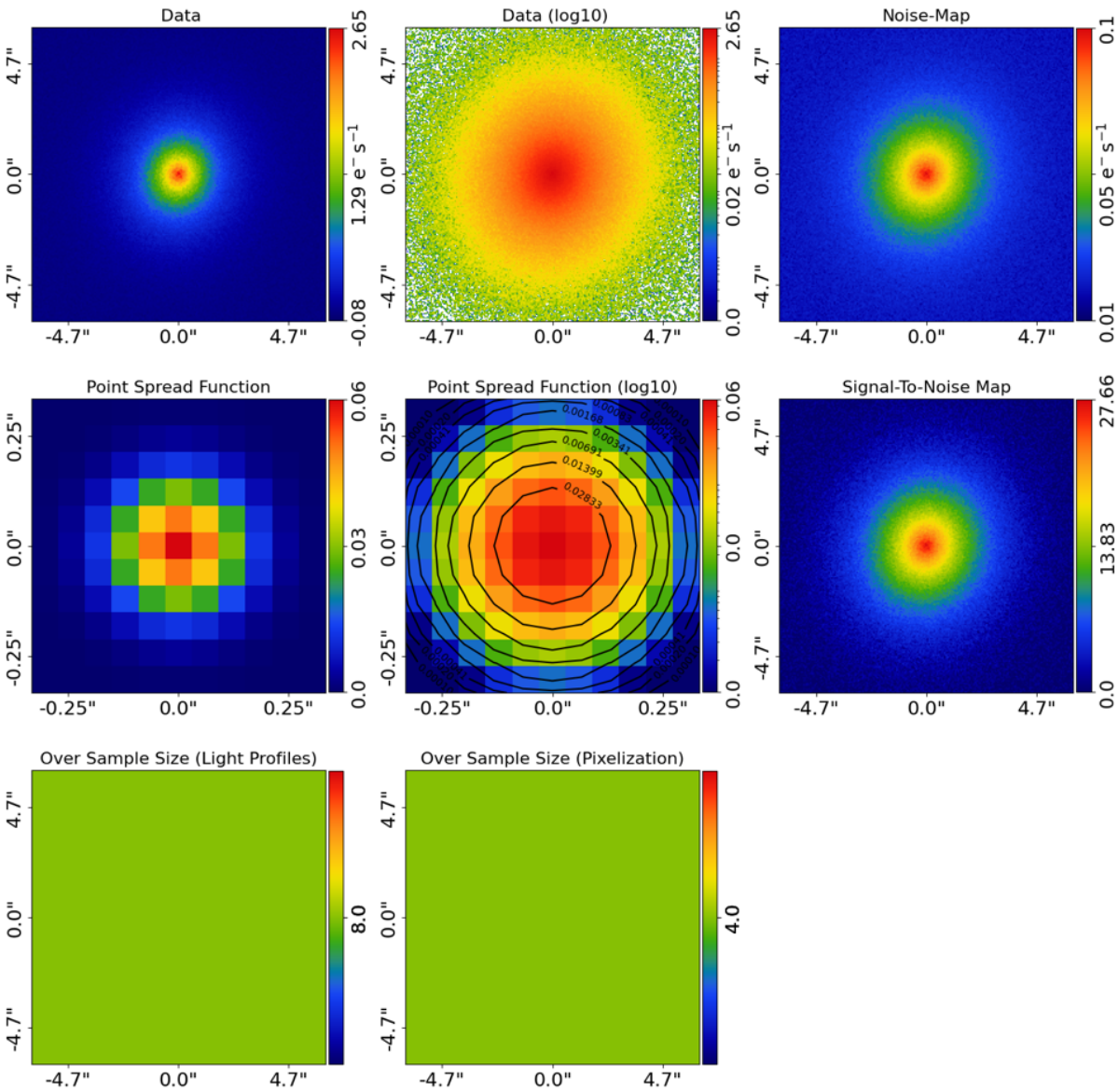
C – Simulated galaxy, “bar_0”, subplot dataset



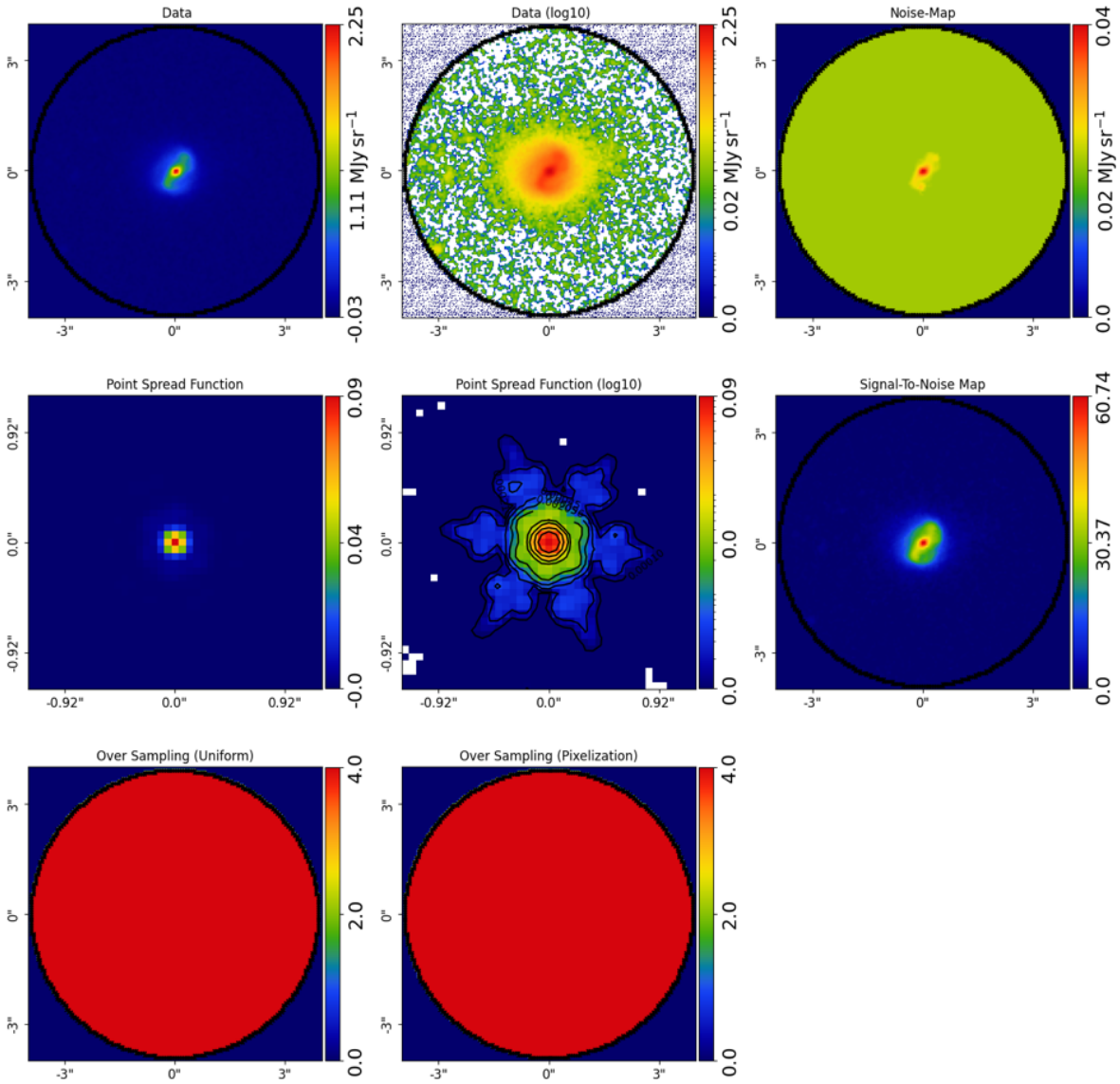
D – Simulated galaxy, “bar_1”, subplot dataset



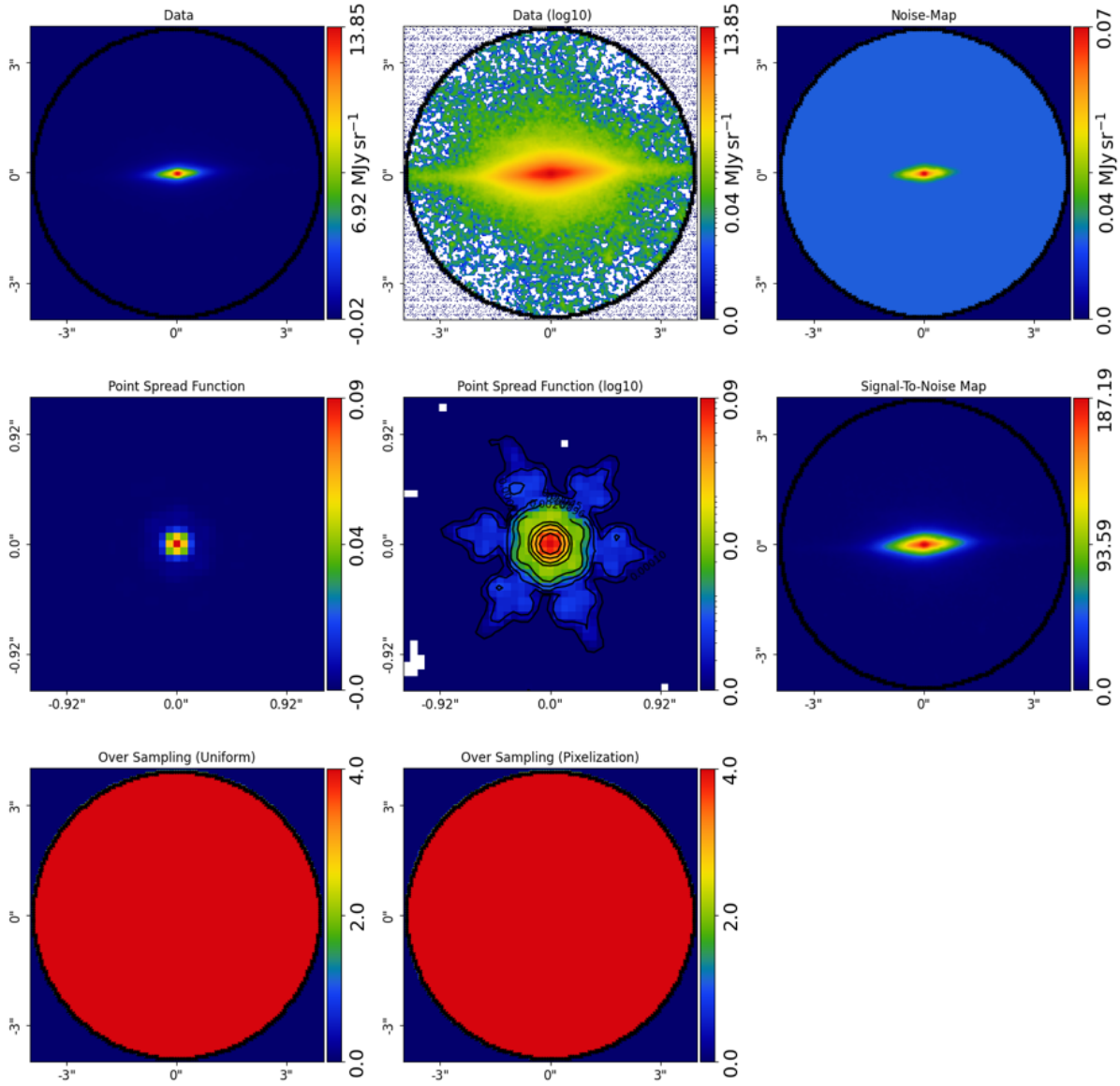
E – Simulated galaxy, “no_bar”, subplot dataset



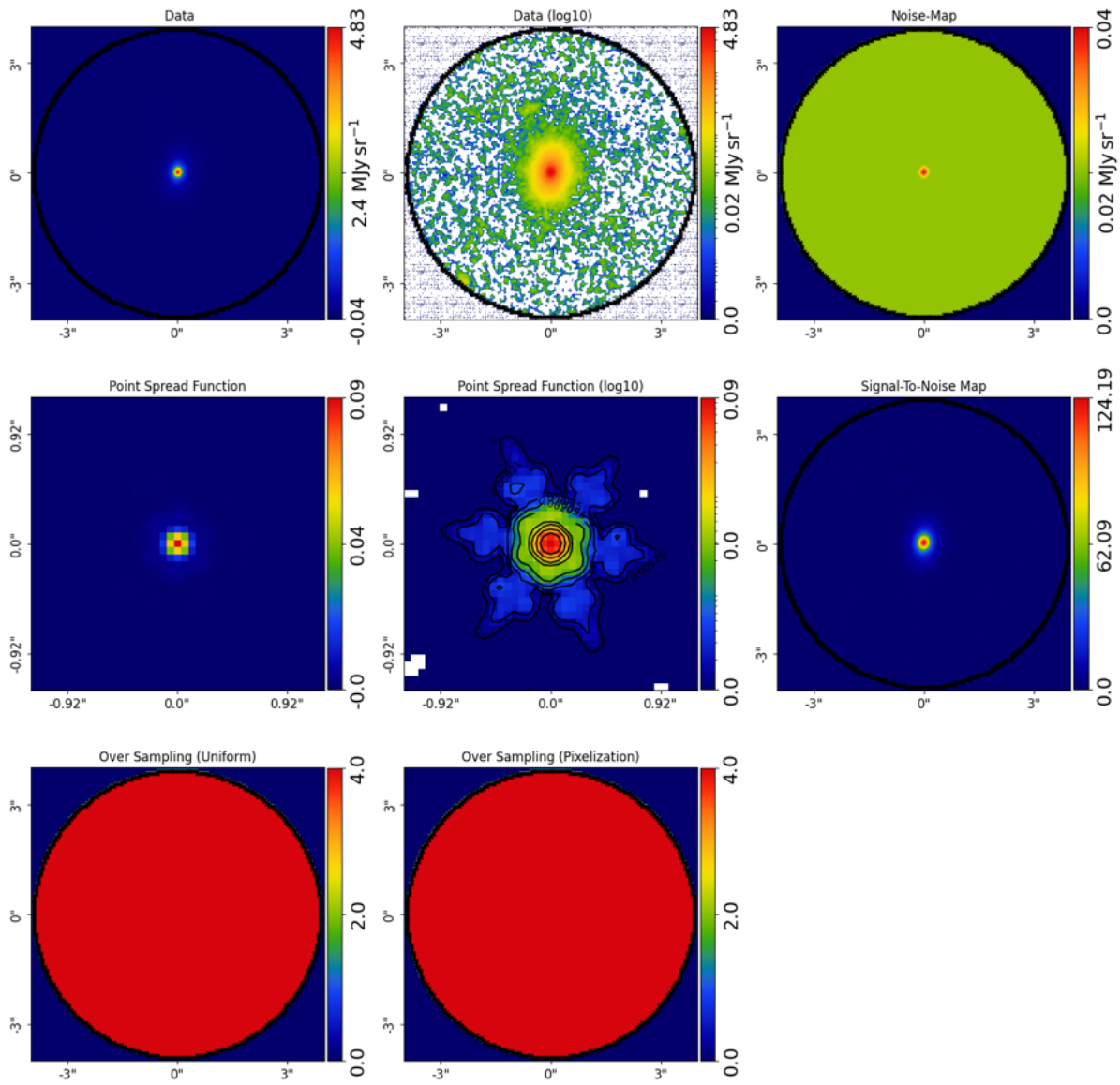
F – JWST data, “dataset_3”, subplot dataset



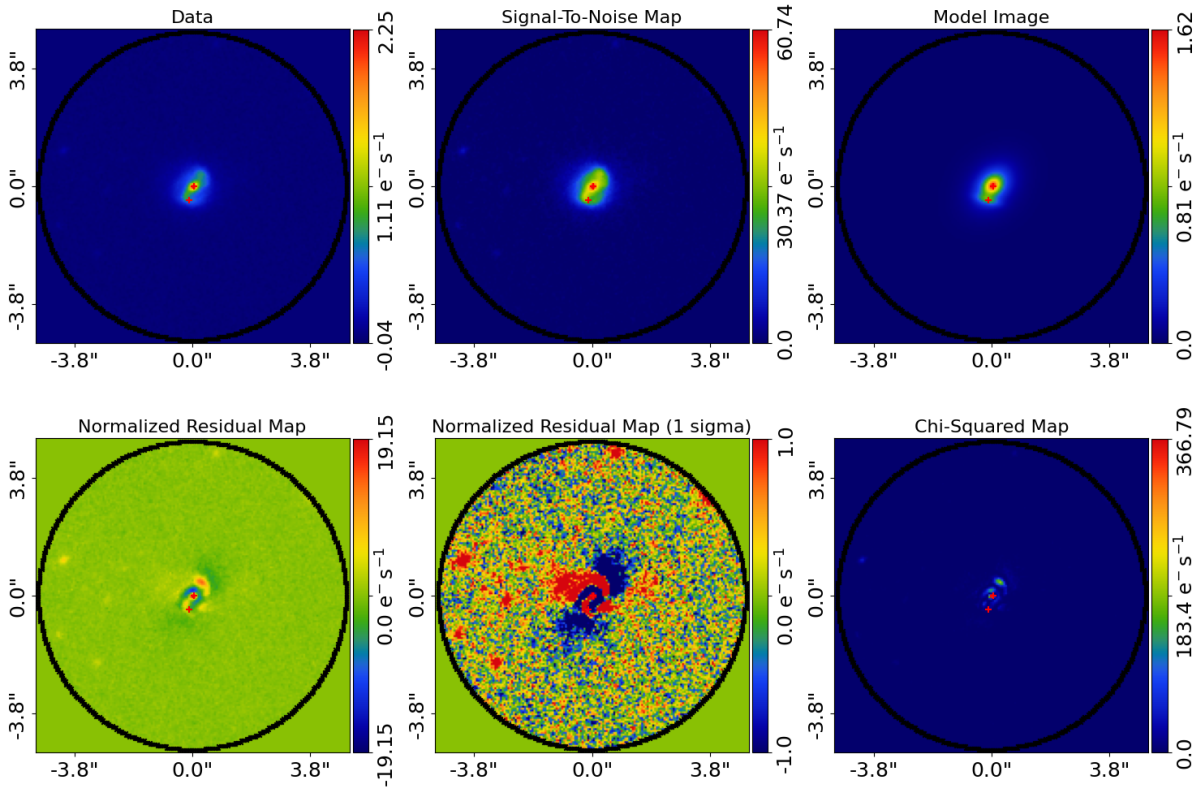
G – JWST data, “dataset_4”, subplot dataset



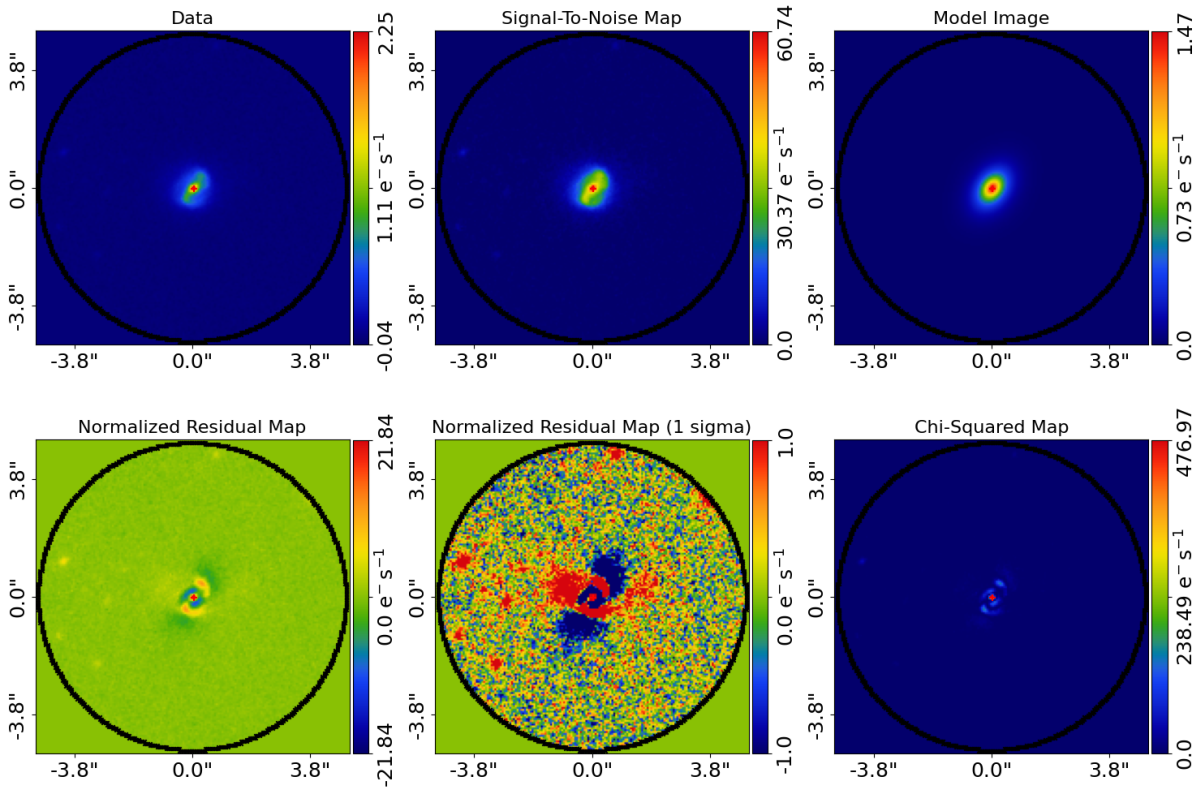
H – JWST data, “dataset_6”, subplot dataset



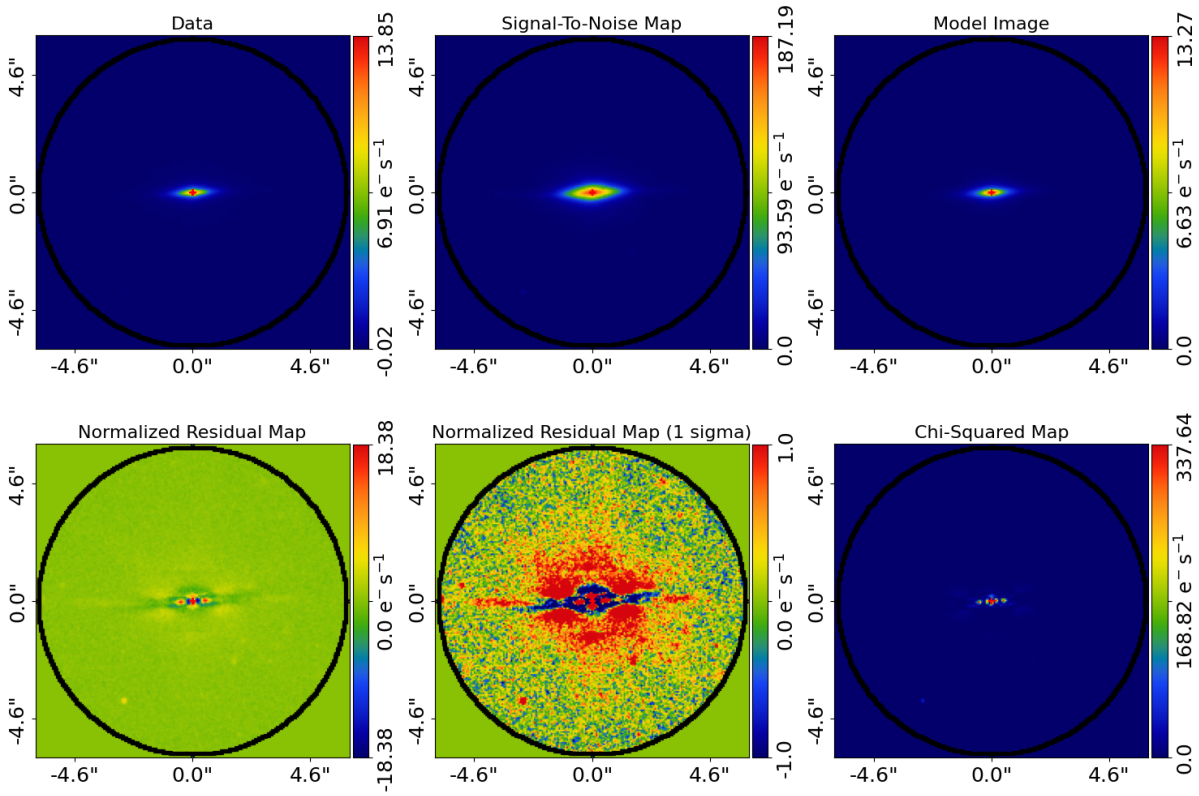
I – Double Sérsic fit to JWST “dataset_3” subplot



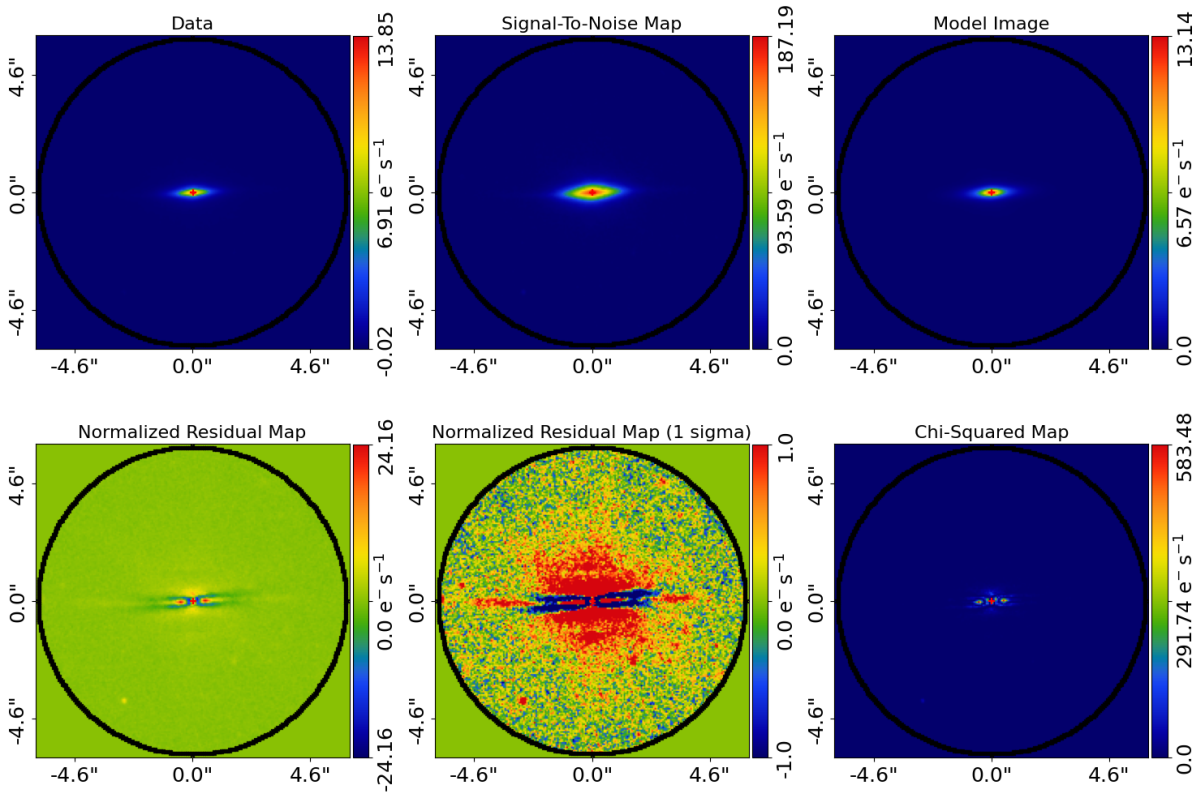
J – Single Sérsic fit to JWST “dataset_3” subplot



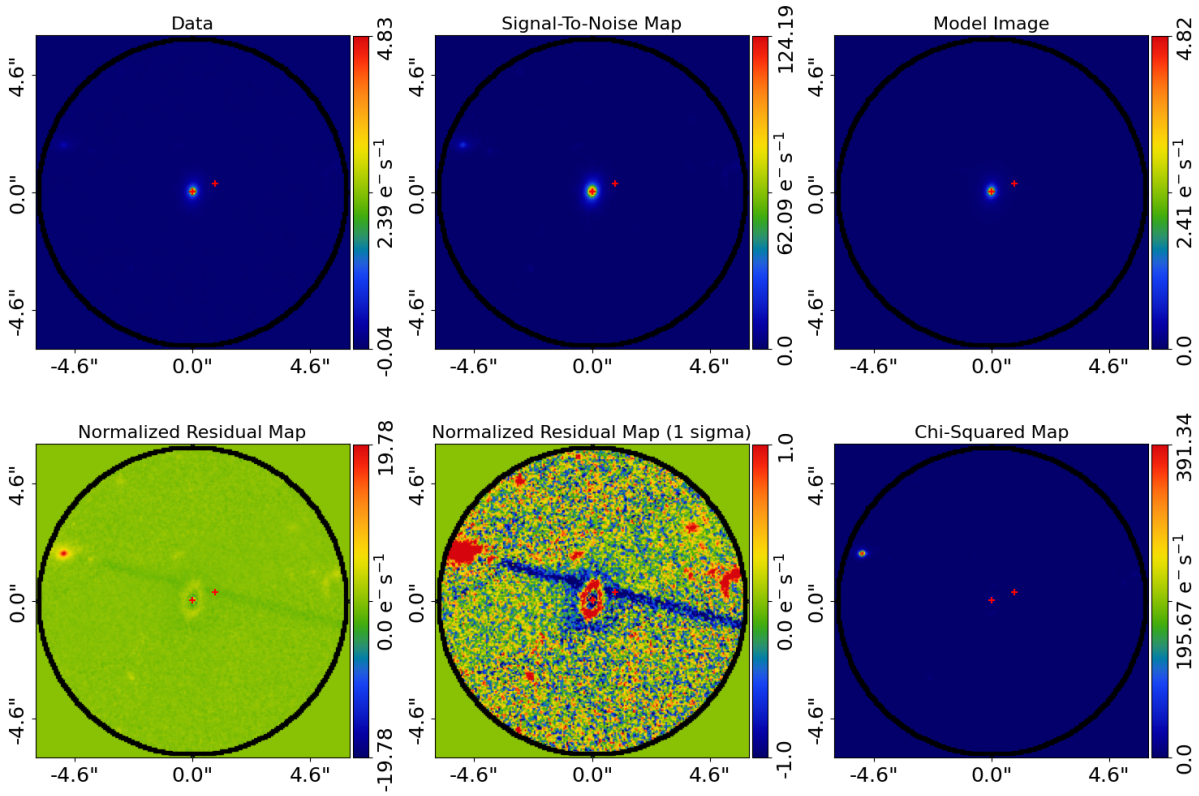
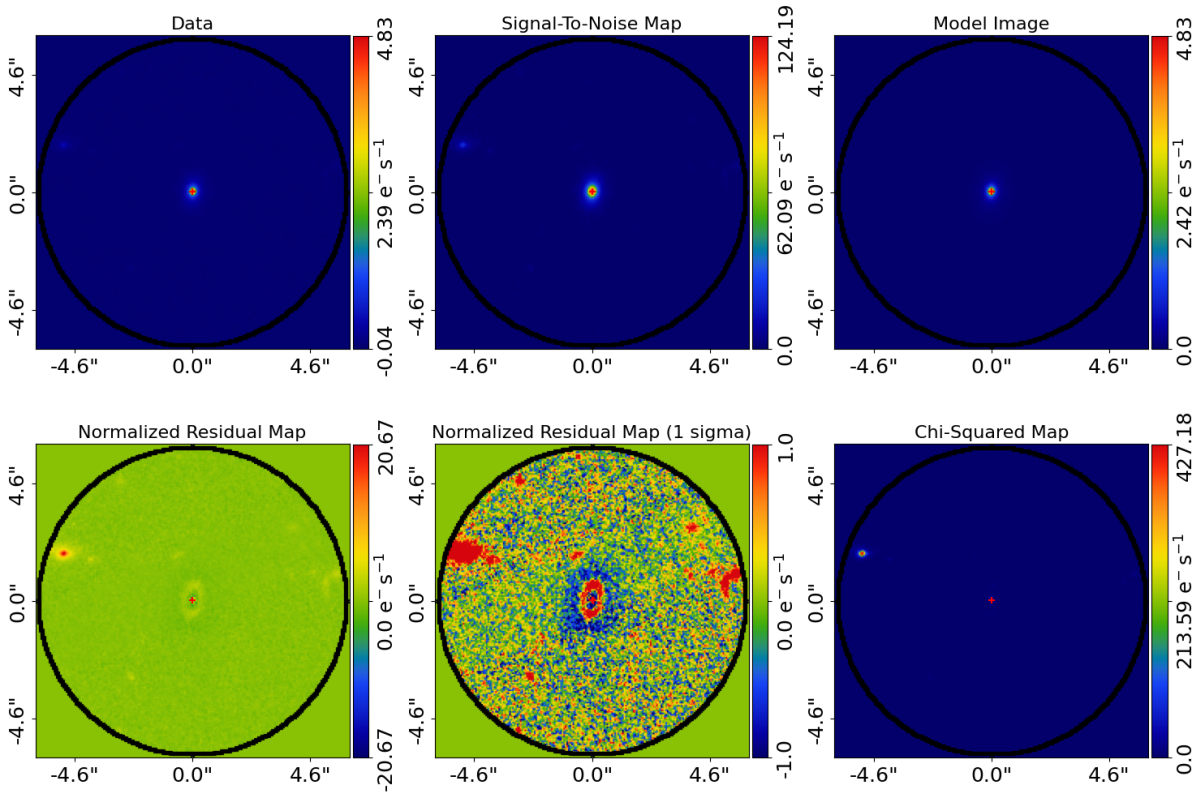
K – Double Sérsic fit to JWST “dataset_4” subplot



L – Single Sérsic fit to JWST “dataset_4” subplot



M – Double Sérsic fit to JWST “dataset_6” subplot



N – Single Sérsic fit to JWST “dataset_6” subplot