Data-Driven Strong Gravitational Lensing Analysis in the Era of Large Sky Surveys

Laurence Perreault-Levasseur

Ciela Institute

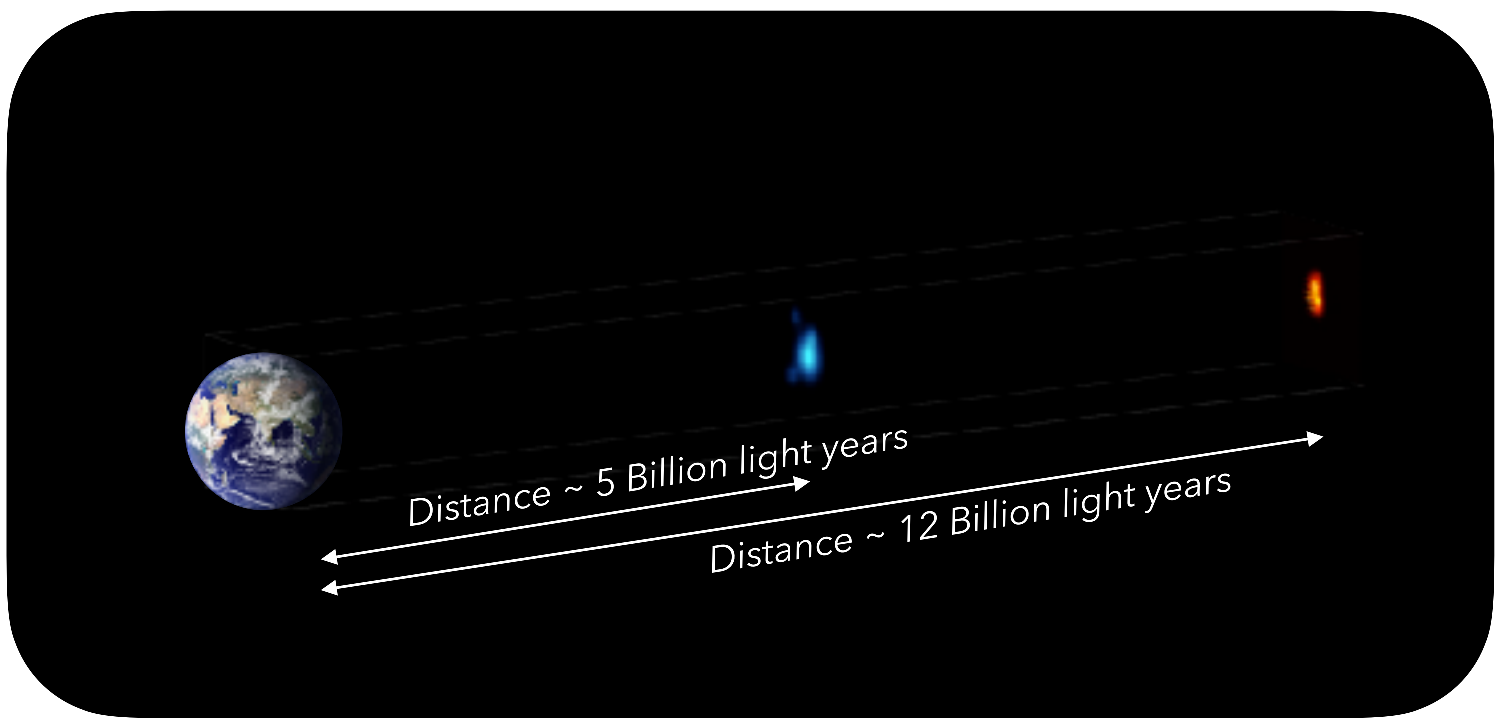
Université de Montréal

Mila

Flatiron Institute
Center for Computational Astrophysics

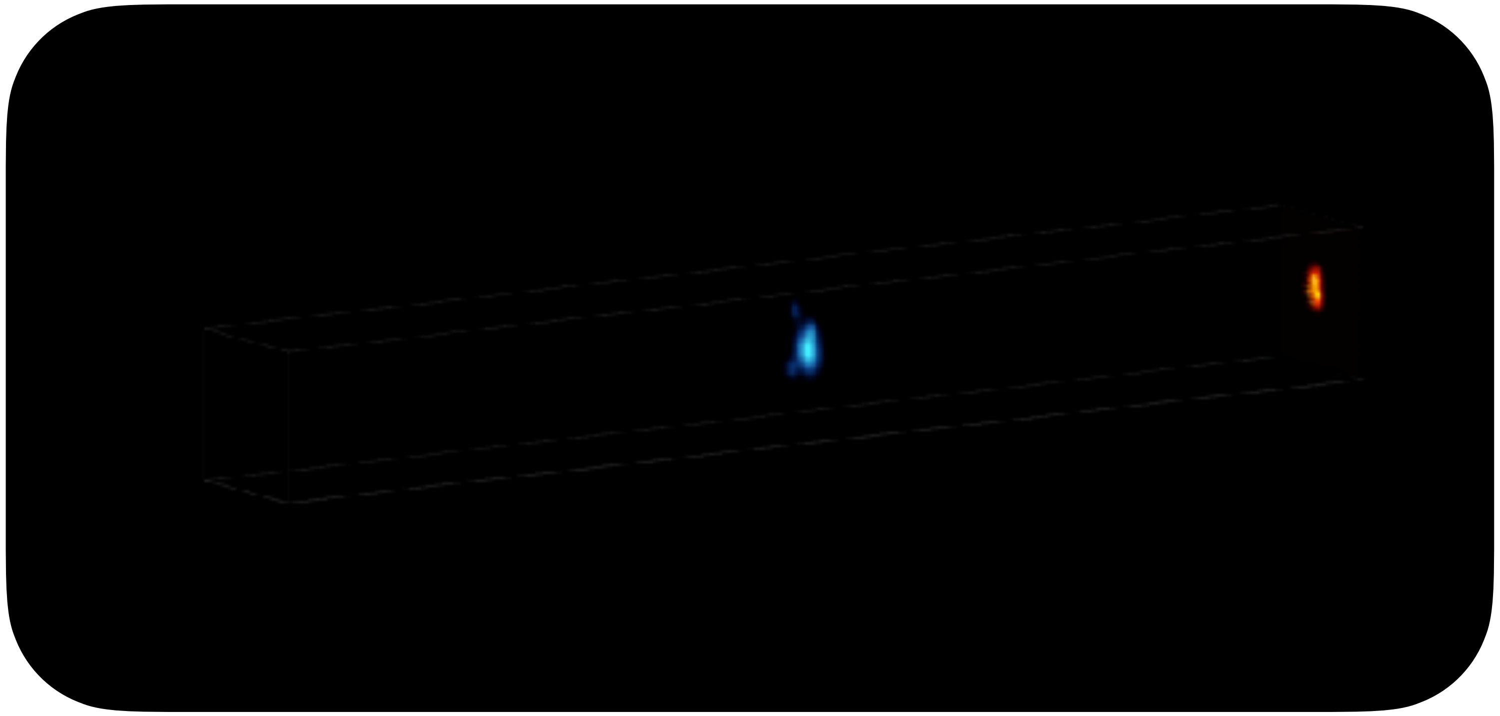
Strong Gravitational Lensing

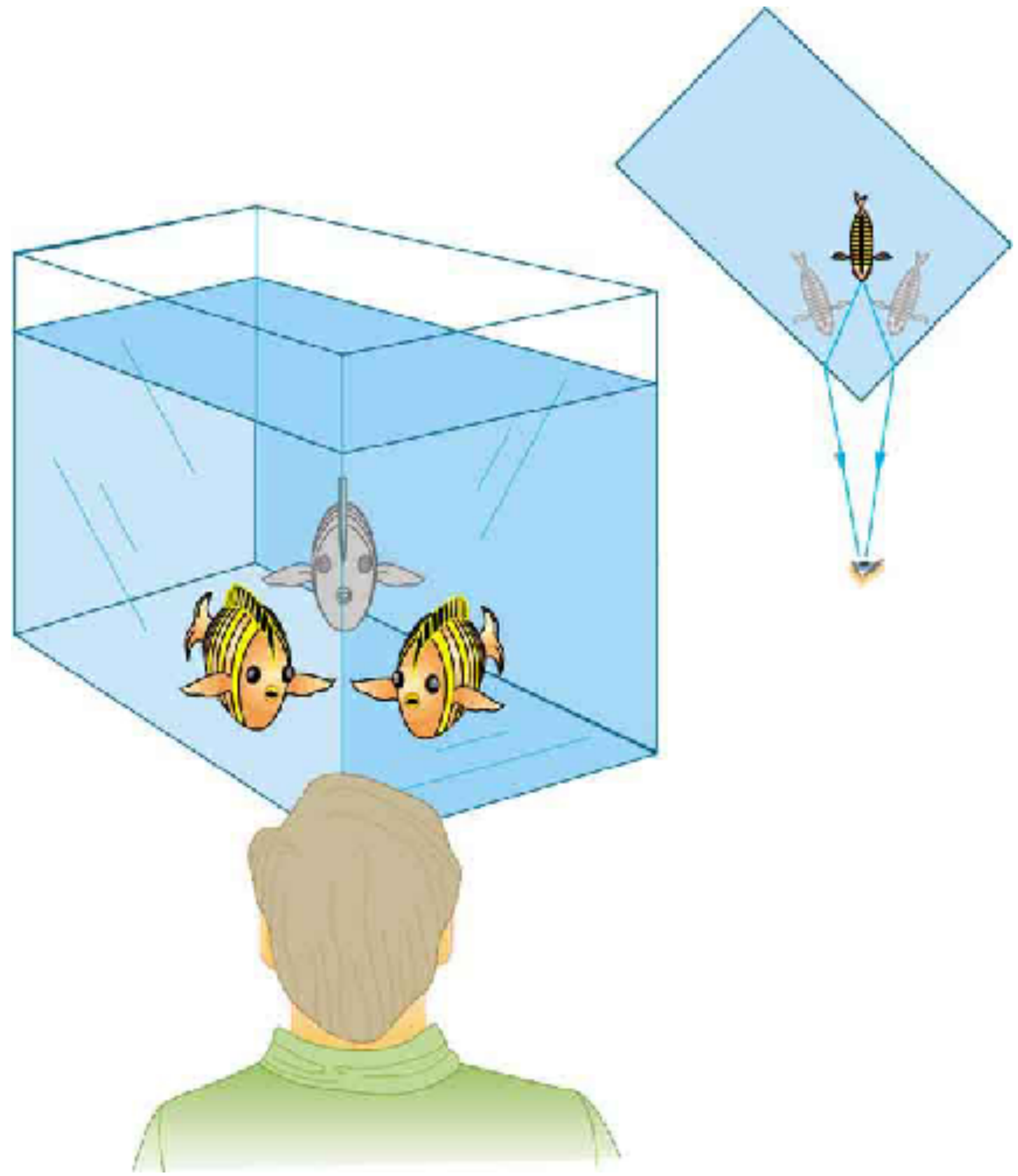
Formation of **multiple images** of a single distant object due to the **deflection of its light** by the **gravity** of intervening structures.

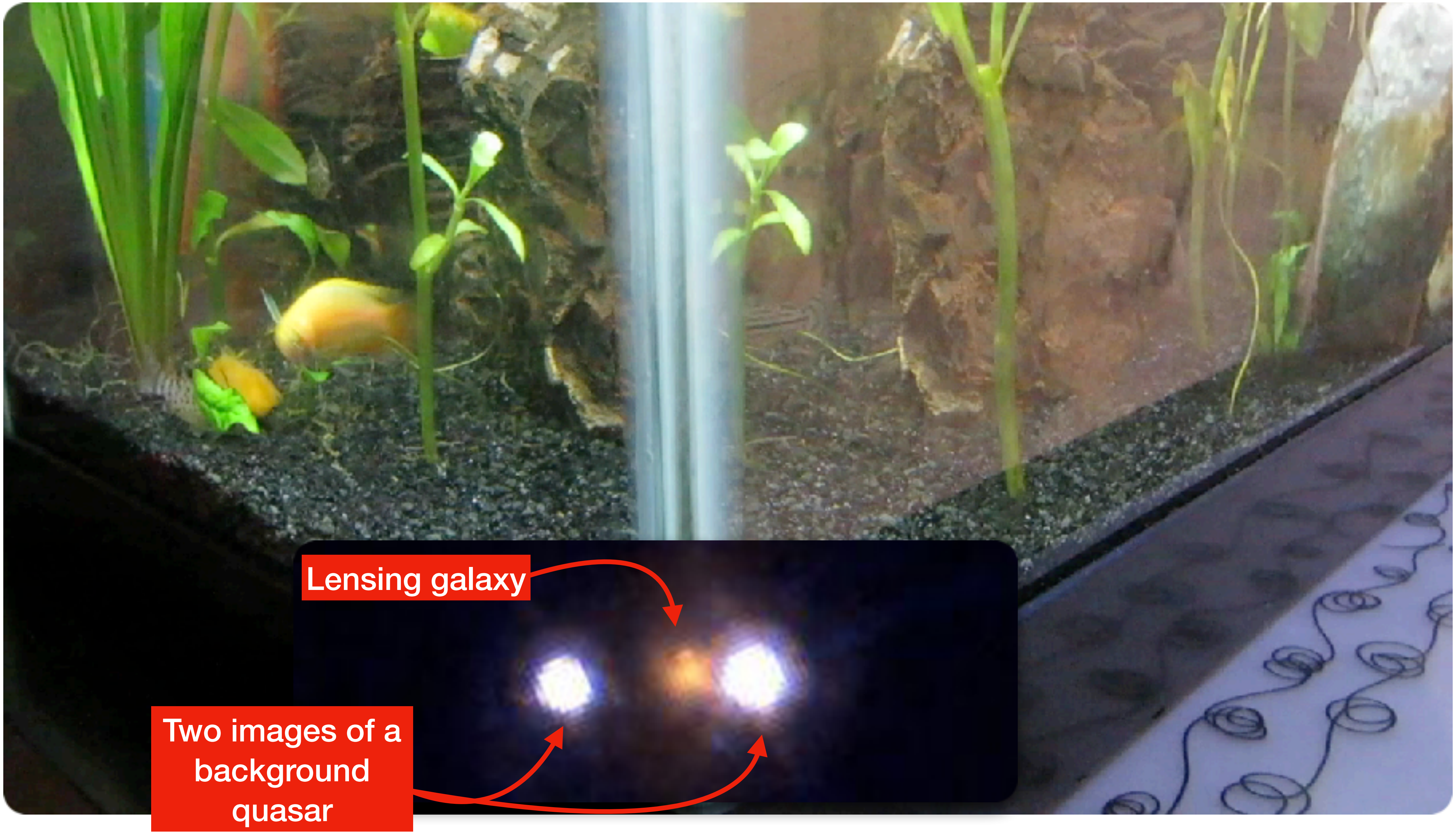


Strong Gravitational Lensing

Formation of **multiple images** of a single distant object due to the **deflection of its light** by the **gravity** of intervening structures.



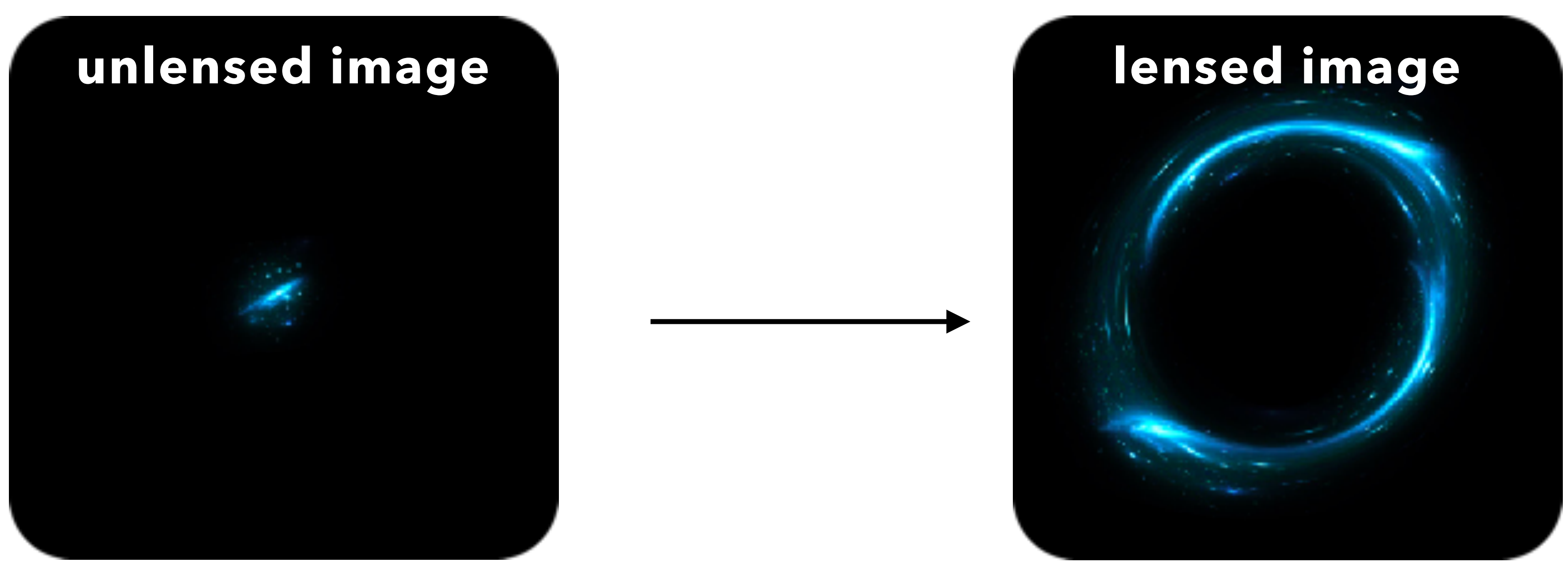



Lensing galaxy
Two images of a background quasar

SCIENCE MOTIVATIONS FOR STRONG LENSING

1 - Use strong lensing as a **cosmic telescope**.

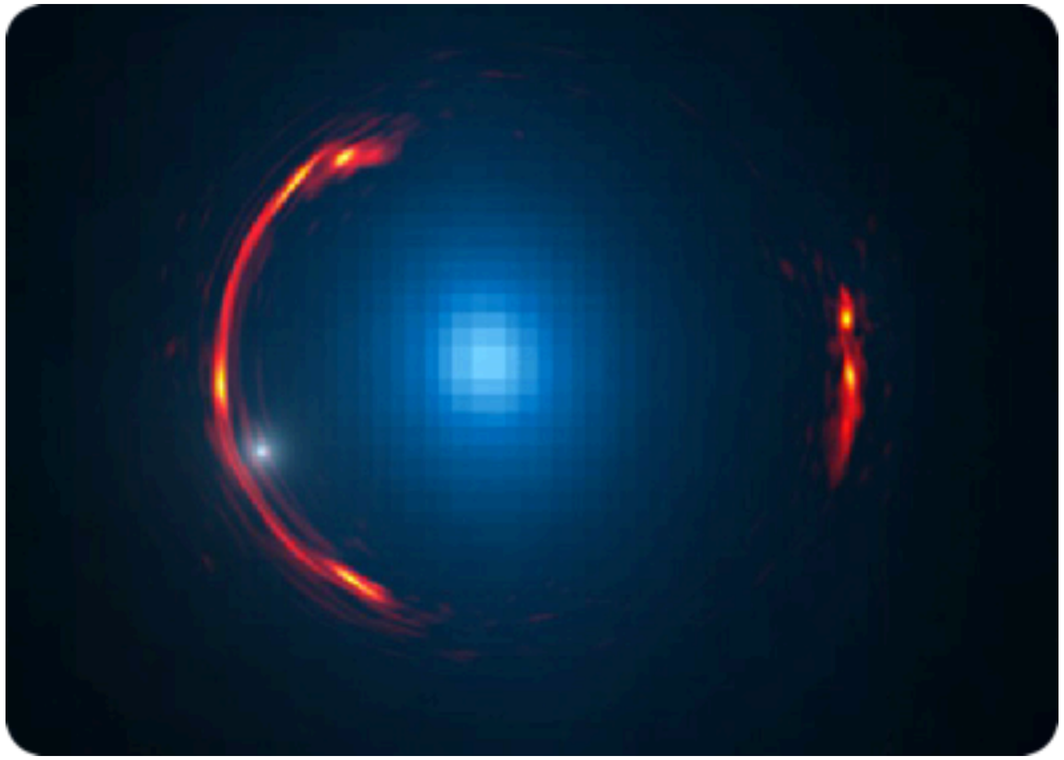
- Lensing **magnifies** the images of sources and makes them appear **brighter**.
- This allows us to study some of the most distant galaxies of the universe that would have been otherwise below our sensitivity or resolution limits.



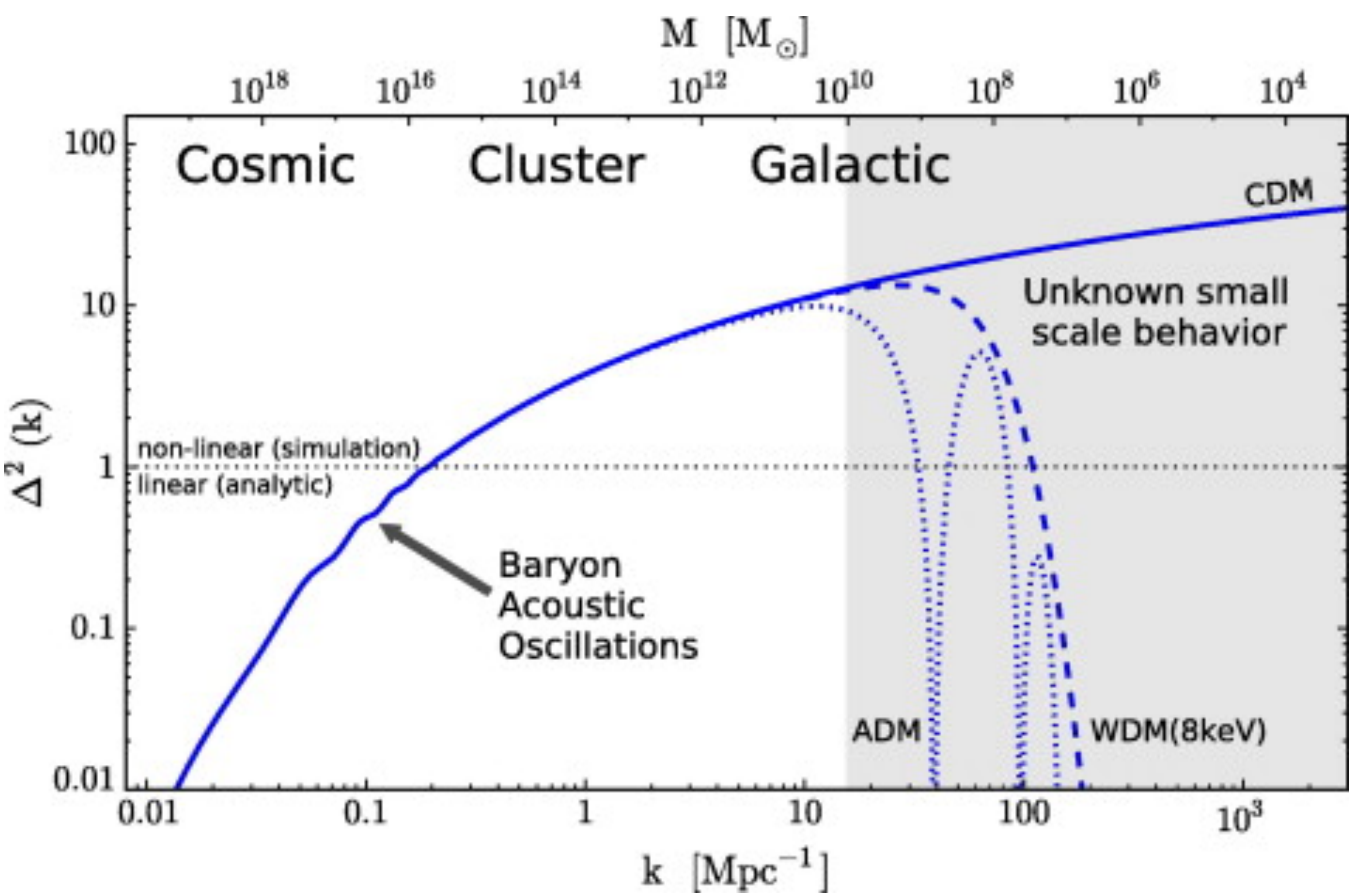
SCIENCE MOTIVATIONS FOR STRONG LENSING

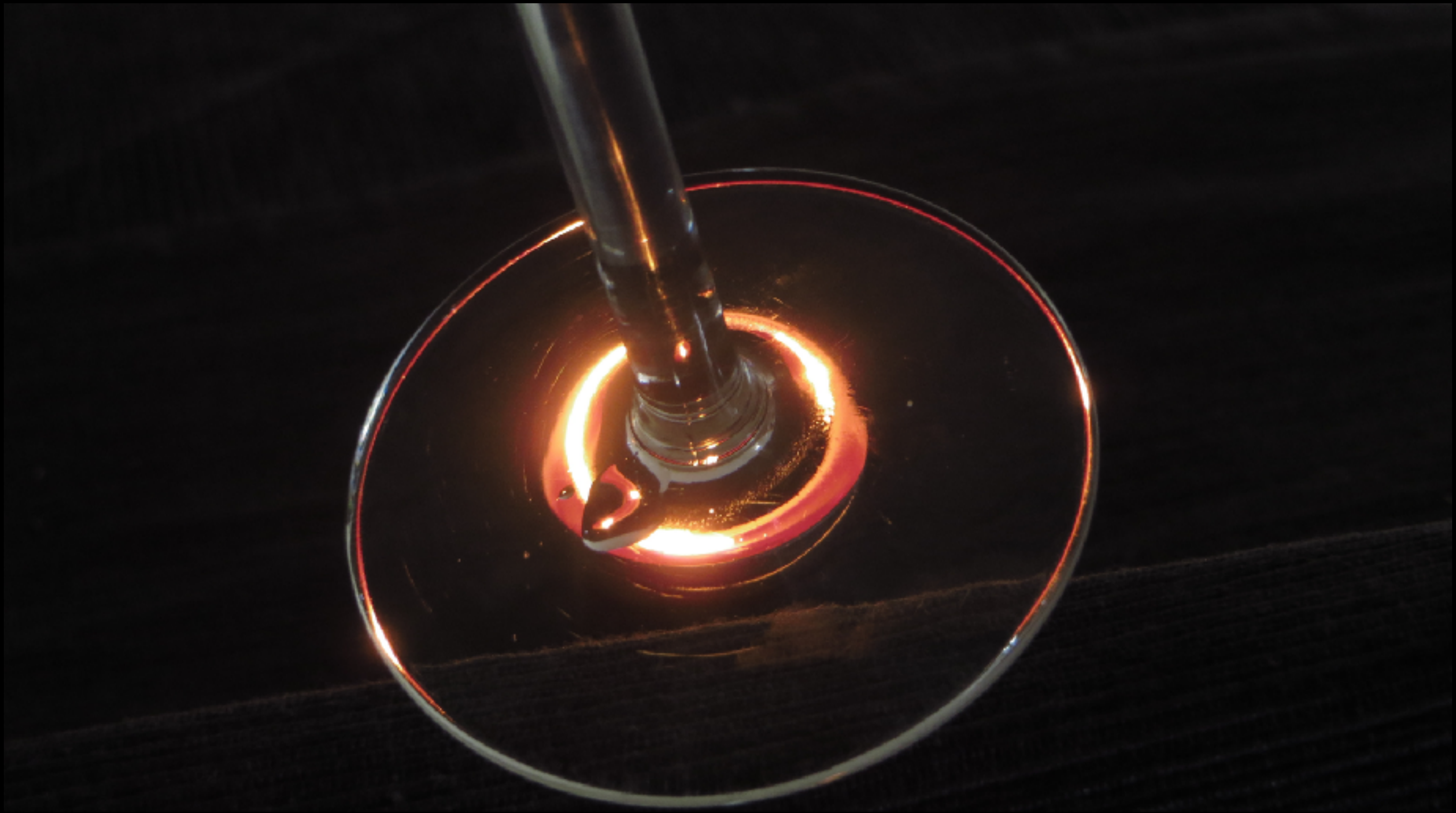
2 - Use lensing to probe the **distribution of matter** in the lensing structures.

- Distortions in images are caused by **gravity**.
- They can be used to map the **distribution of matter** in the lens.
- Particularly useful for studying **dark matter.**



Matter power spectrum



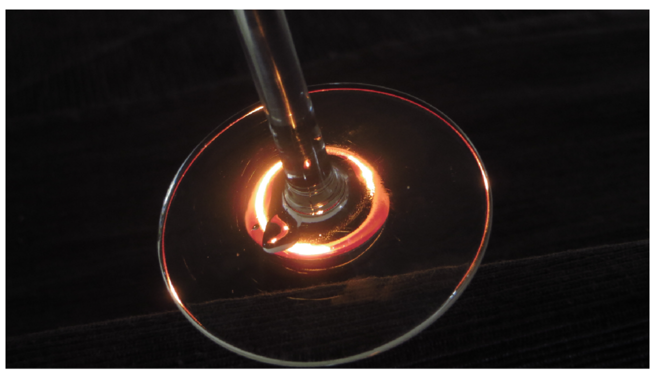


SCIENCE MOTIVATIONS FOR STRONG LENSING

3 - Measure **comological parameters** (H_0).

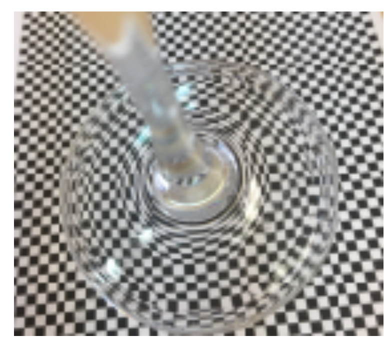
- Different images are produced because light follows **different paths**.
- These paths are of **different lengths**.
- If the source has time variability, this will cause **time delays** between different images.

Lensing analysis



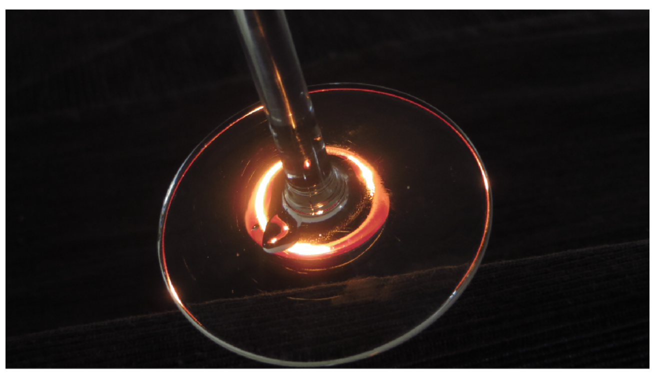
Data

1: Morphology of the background source (the true, undistorted image of the candle)



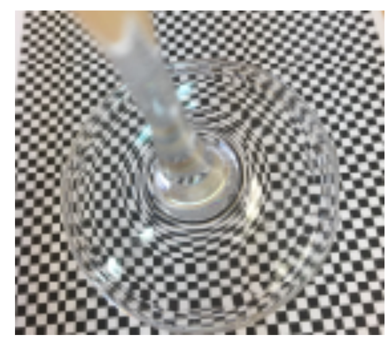
2: Matter distribution in the lens (the shape of the wineglass)

Lensing analysis



Data

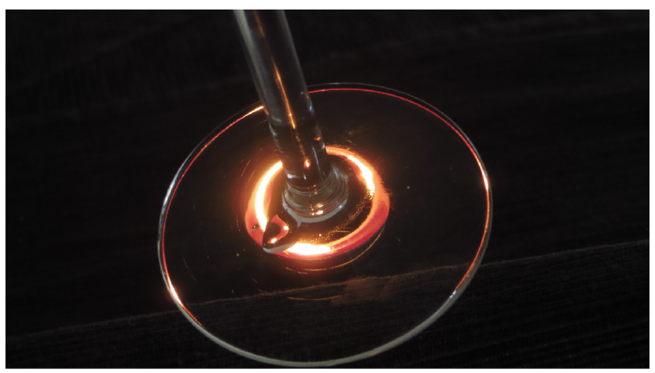
1: Morphology of the background source (the true, undistorted image of the candle)



2: Matter distribution in the lens (the shape of the wineglass)

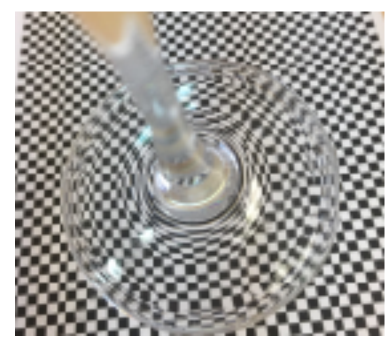
$$y = L(p)x + n$$

Lensing analysis



Data

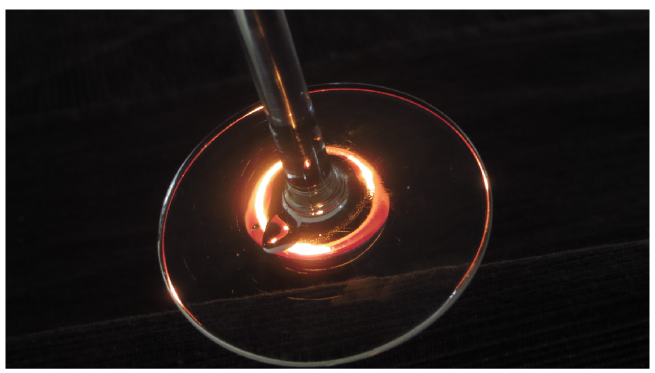
1: Morphology of the background source (the true, undistorted image of the candle)



2: Matter distribution in the lens (the shape of the wineglass)

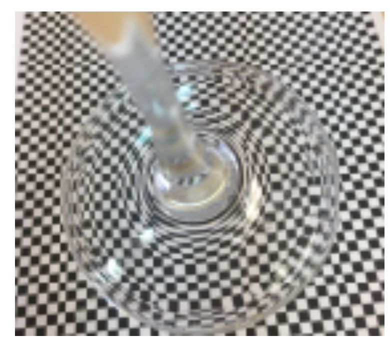
Data → $$y = L(p)x + n$$

Lensing analysis



Data

1: Morphology of the background source (the true, undistorted image of the candle)

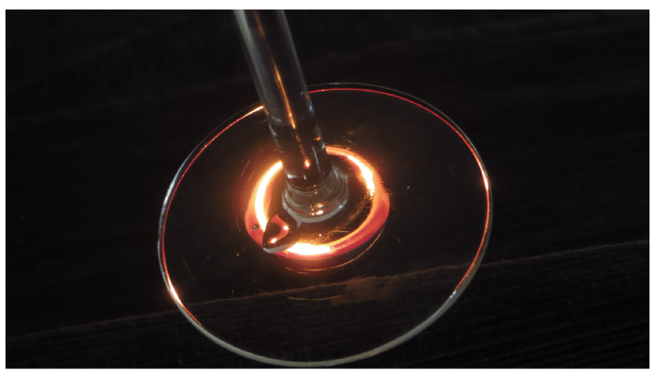


2: Matter distribution in the lens (the shape of the wineglass)

Source Parameters (linear)

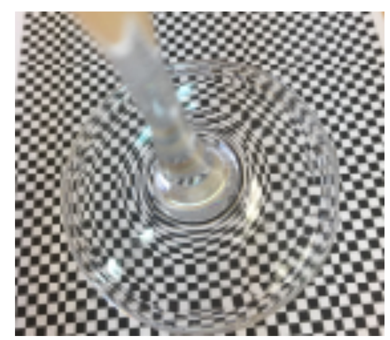
Data → $$y = L(p)x + n$$

Lensing analysis



Data

1: Morphology of the background source (the true, undistorted image of the candle)



2: Matter distribution in the lens (the shape of the wineglass)

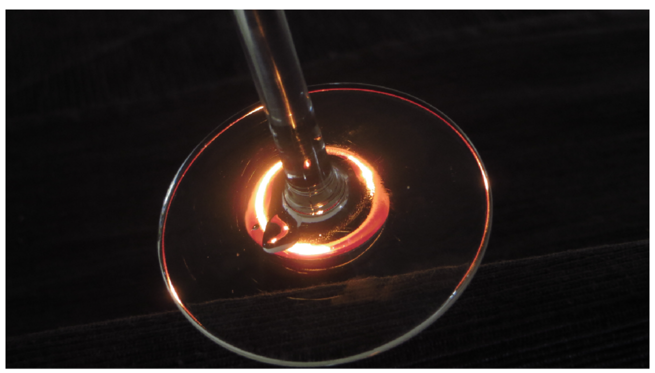
Source Parameters (linear) → x

Data → y

$$y = L(p)x + n$$

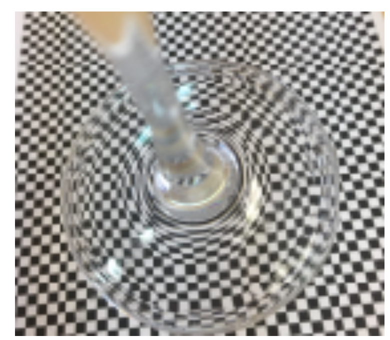
Lens Parameters (non-linear) → p

LENSING ANALYSIS



Data

1: Morphology of the background source (the true, undistorted image of the candle)



2: Matter distribution in the lens (the shape of the wineglass)

Source Parameters (linear) ↓

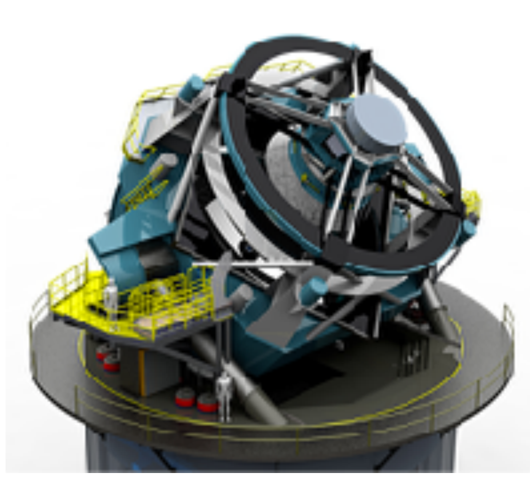
Data → $y = L(p)x + n$

↑ Lens Parameters (non-linear)

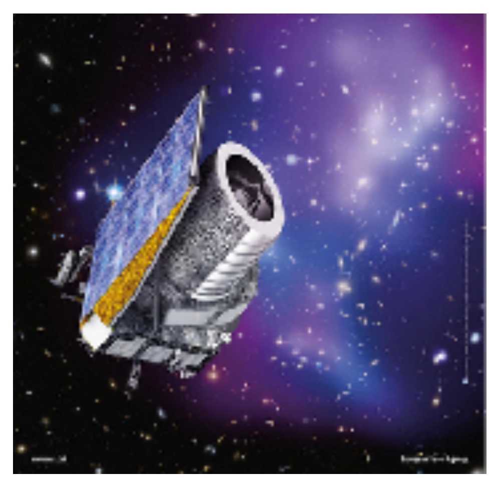
↑ Noise

LOOKING INTO THE FUTURE

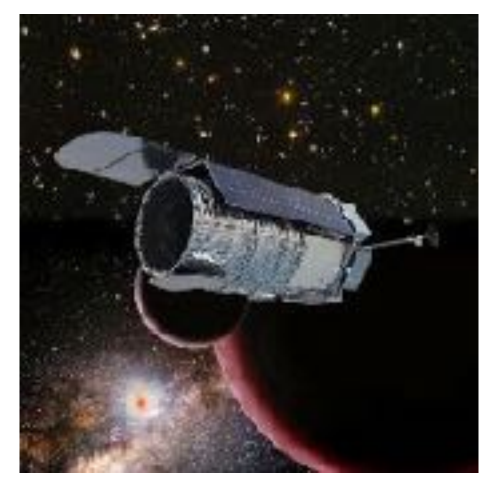
In the next few years, we're expecting to discover more than 170,000 new lenses.



LSST



Euclid consortium



NANCY GRACE ROMAN SPACE TELESCOPE

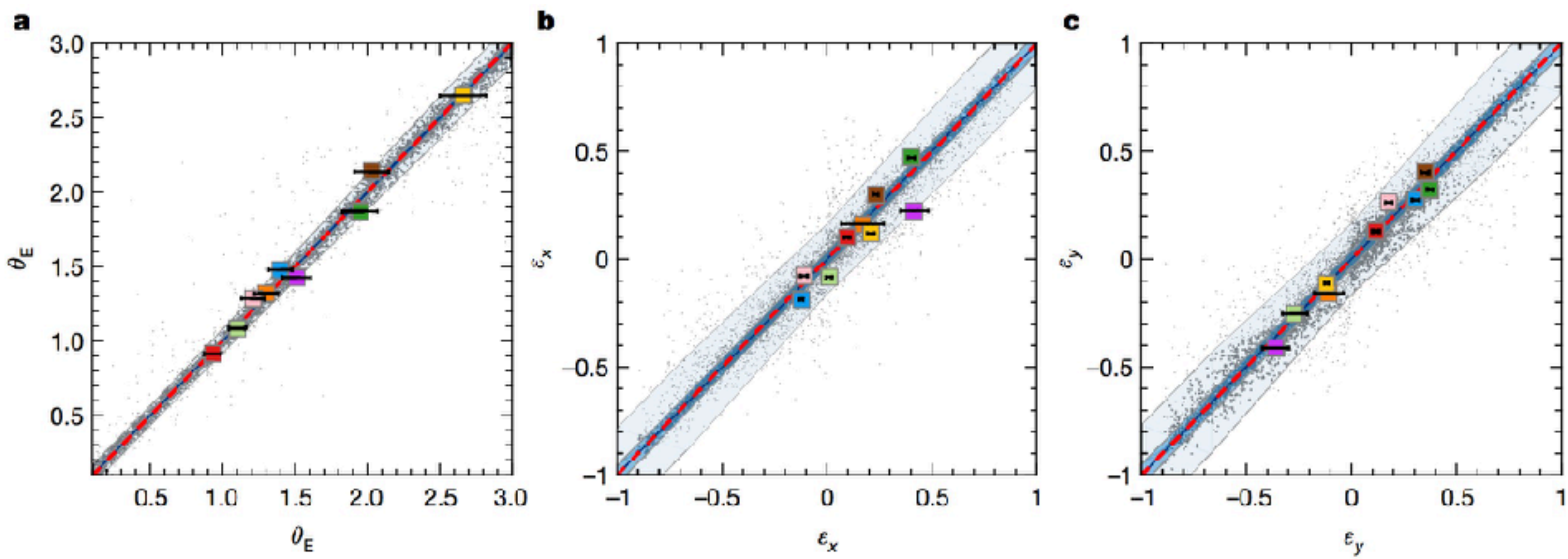
Methods for the future:

How are we going to analyze 170,000 lenses?

- Lens modeling is **very slow**.
- Simple lens model takes ~3 days

=> **1,400 years !**

ESTIMATING THE MATTER DISTRIBUTION PARAMETERS WITH CNNS

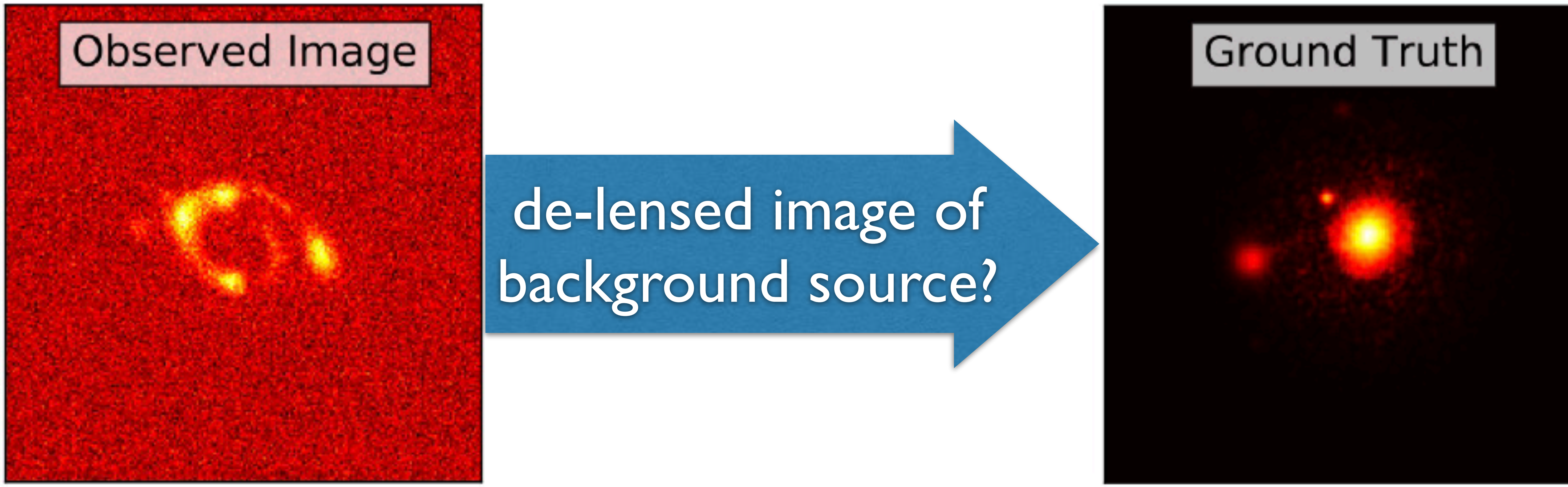


10 million times faster than traditional lens modeling.

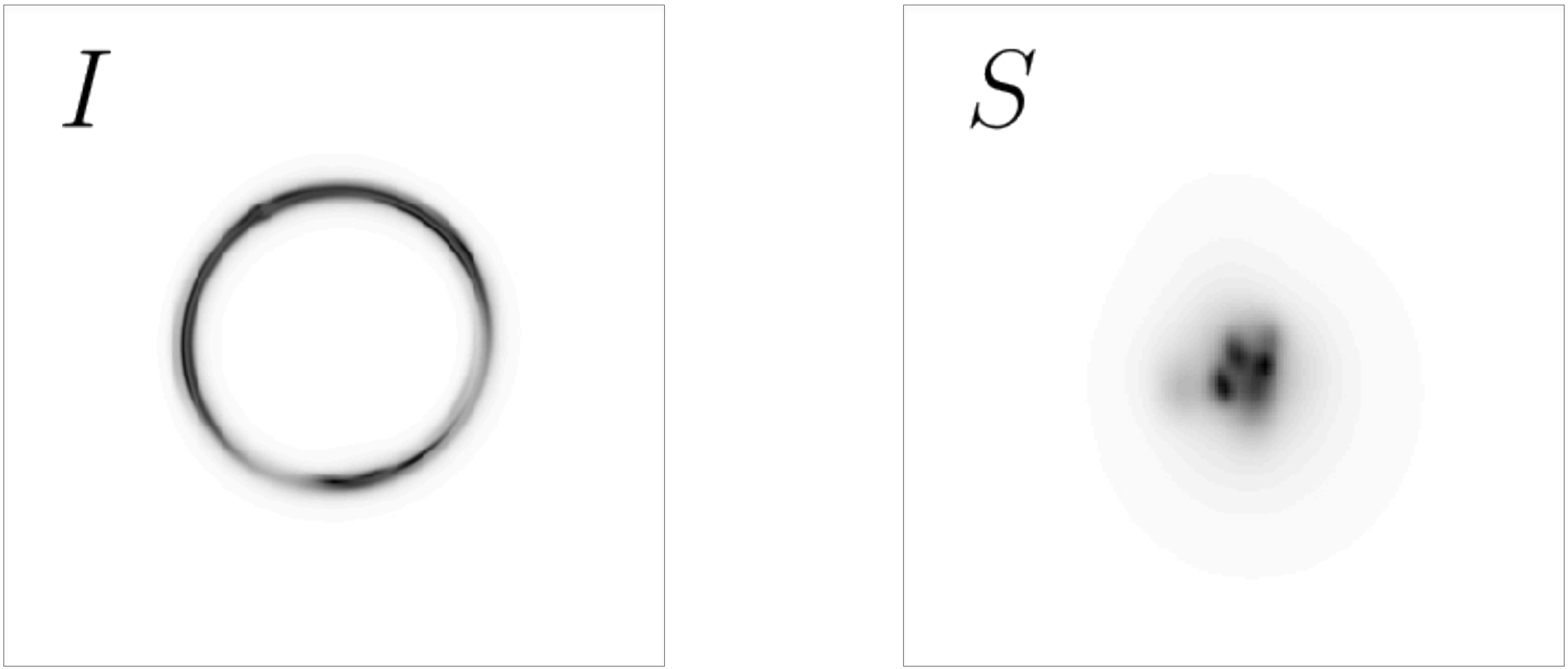
0.01 seconds on a **single GPU**

Hezaveh, Perreault Levasseur, Marshall, Nature, 2017

UNDISTORTED IMAGE OF THE BACKGROUND SOURCE

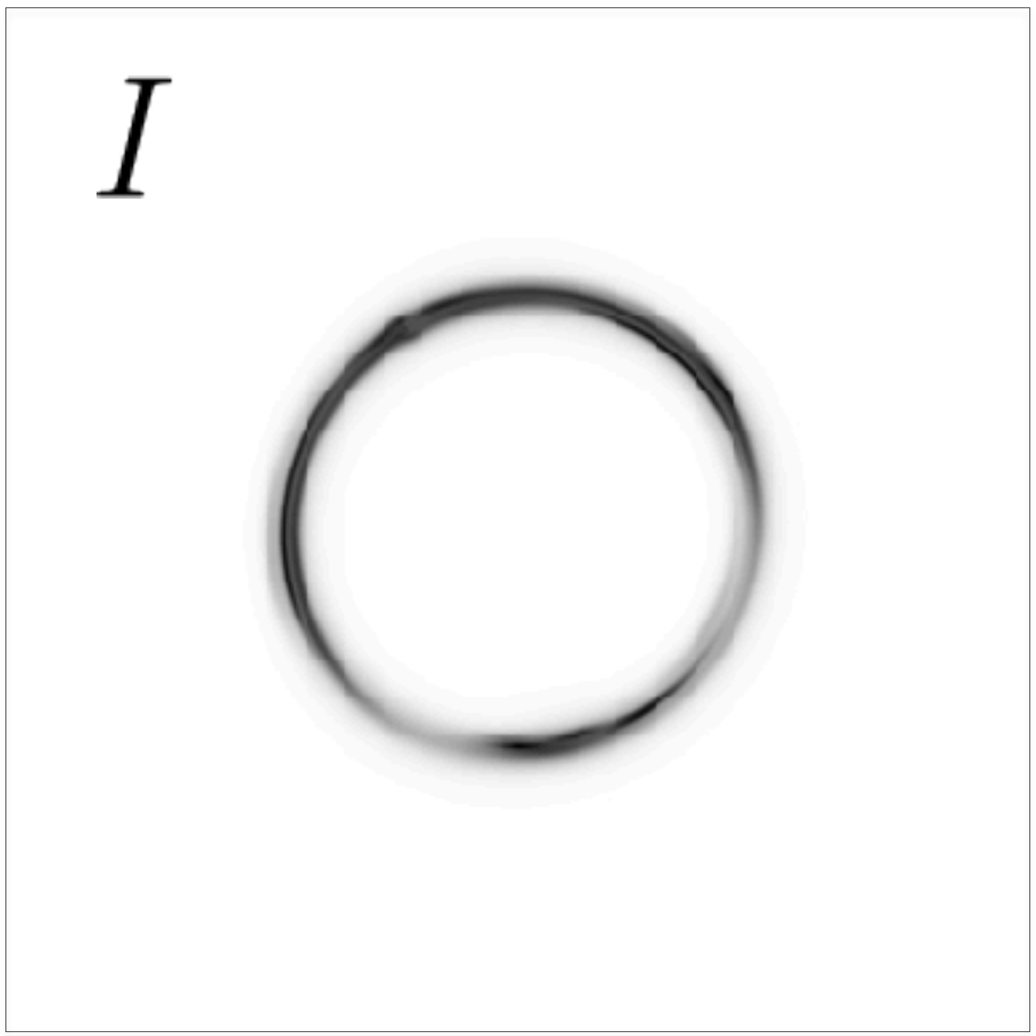


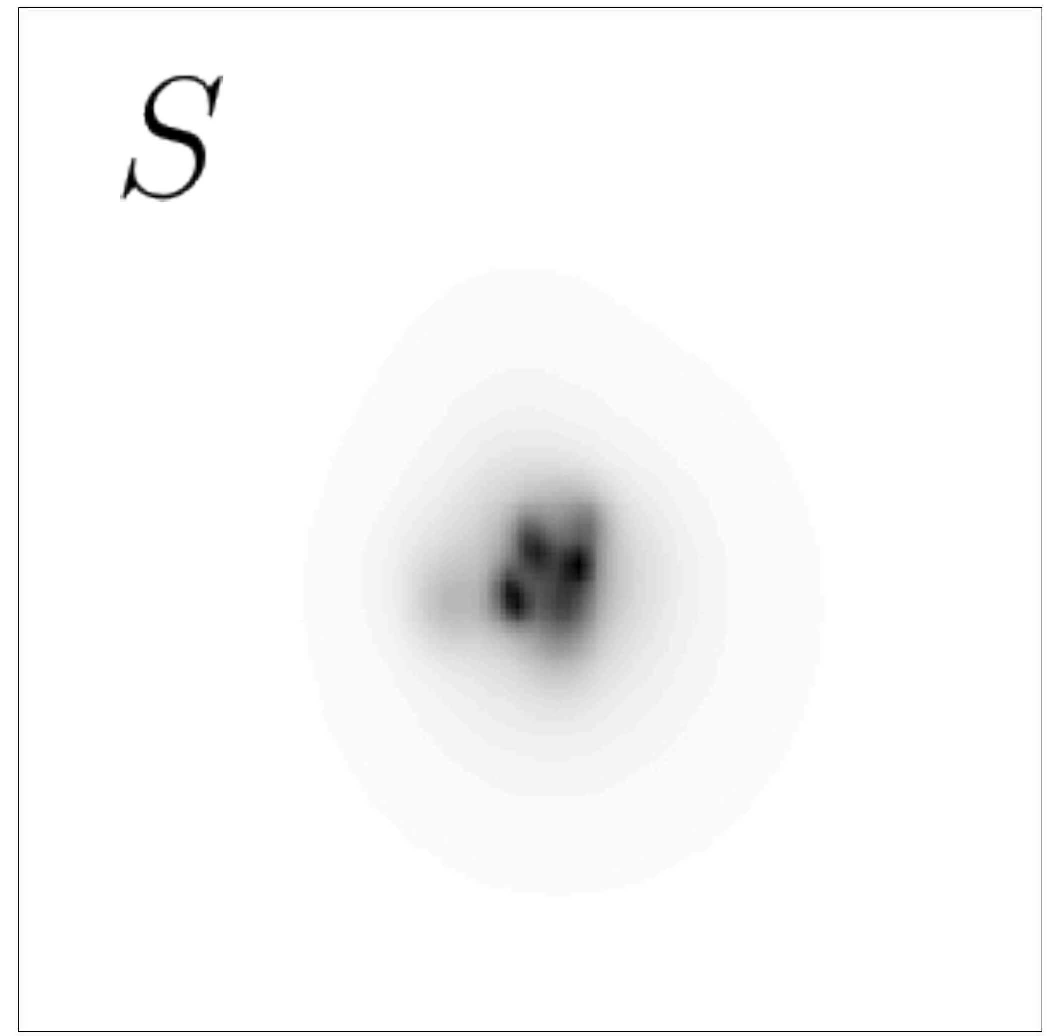
Pixel values of the background source are linear parameters



$$I = L(p)S$$

Pixel values of the background source are linear parameters

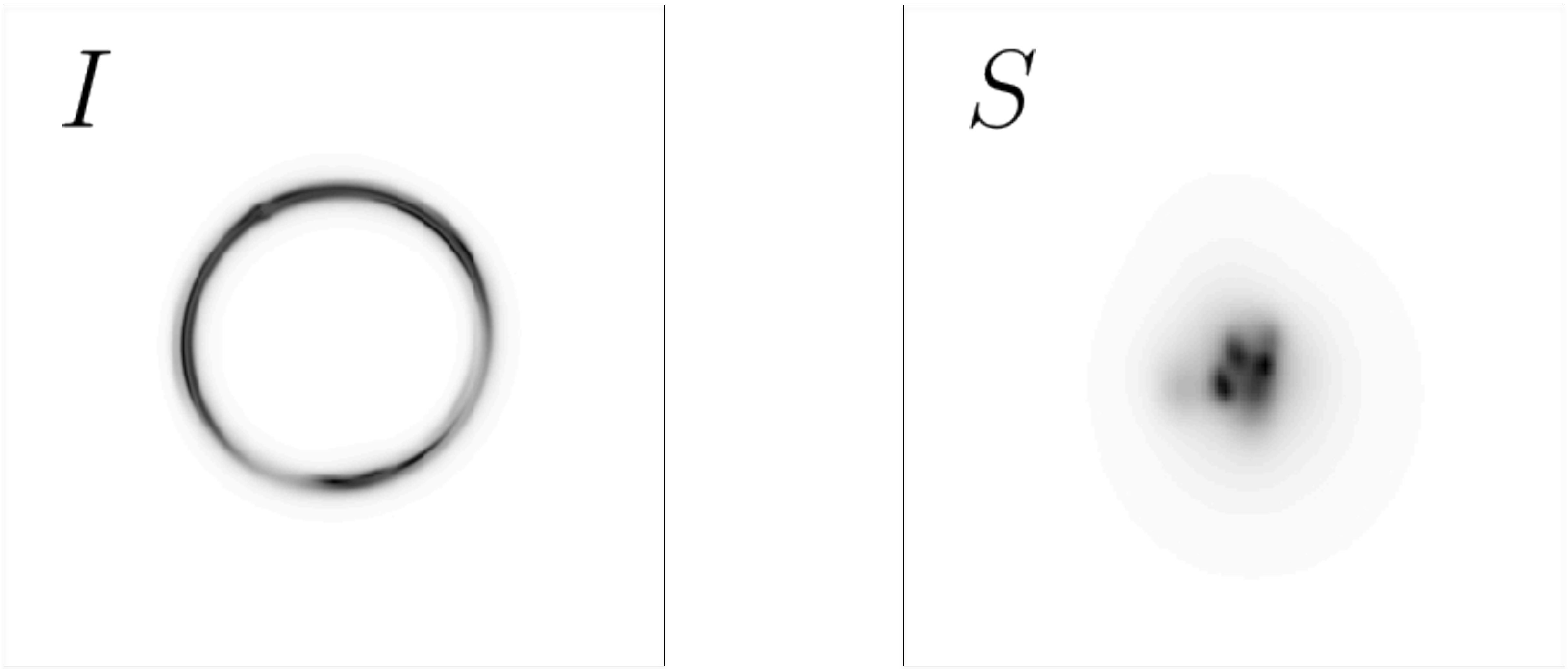




$$I = L(p)S$$

$$S = (L^T C_N^{-1} L)^{-1} L^T C_N^{-1} D$$

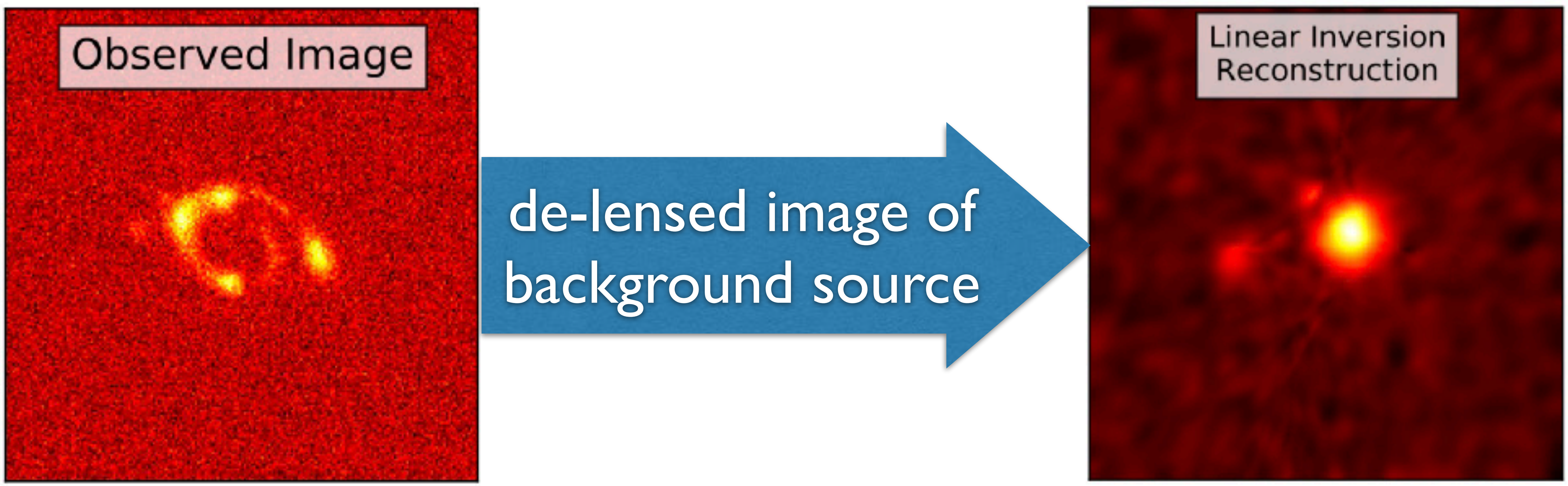
Pixel values of the background source are linear parameters



$$I = L(p)S$$

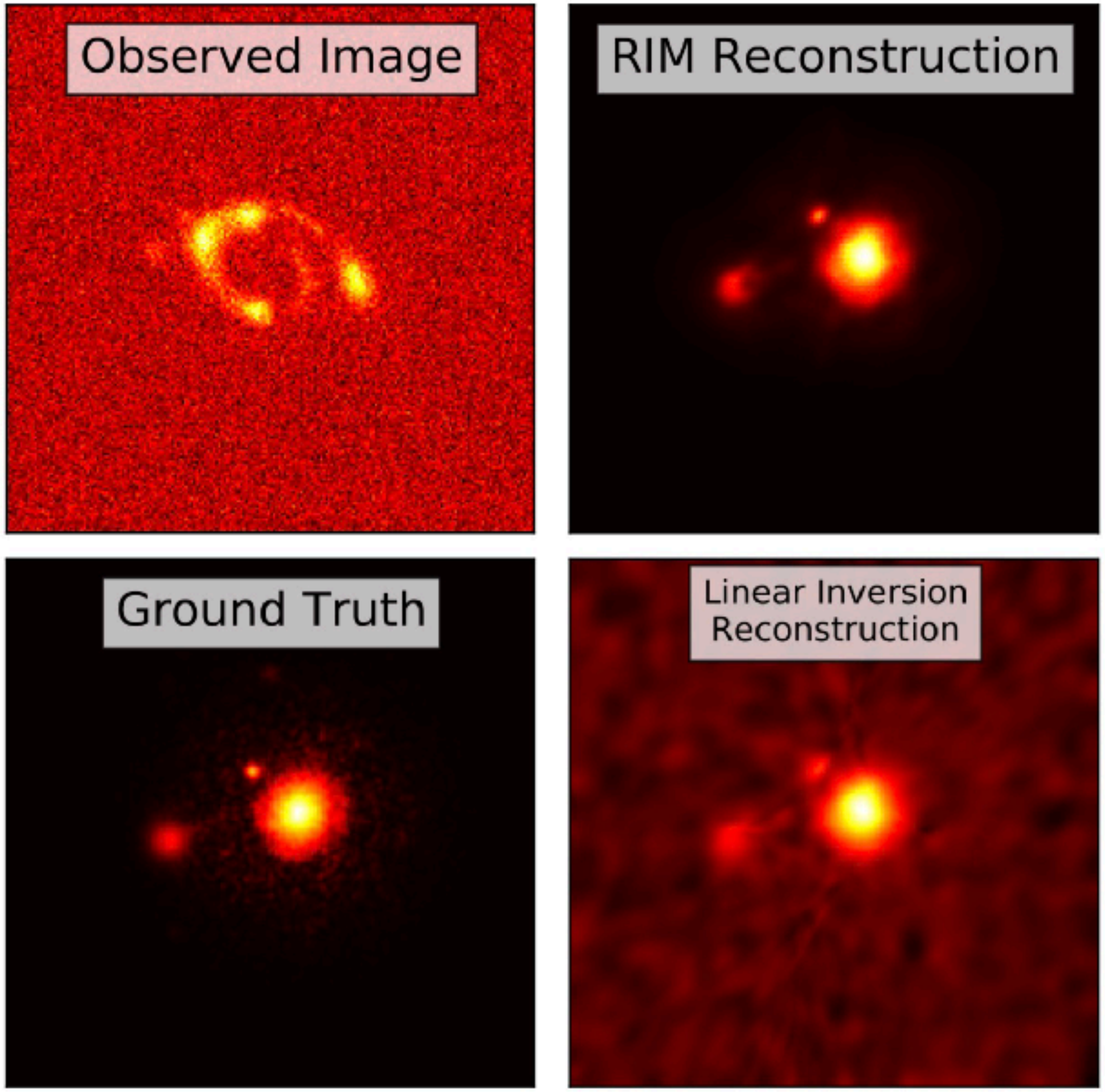
$$S = (L^T C_N^{-1} L + C_p^{-1})^{-1} L^T C_N^{-1} D$$

UNDISTORTED IMAGE OF THE BACKGROUND SOURCE



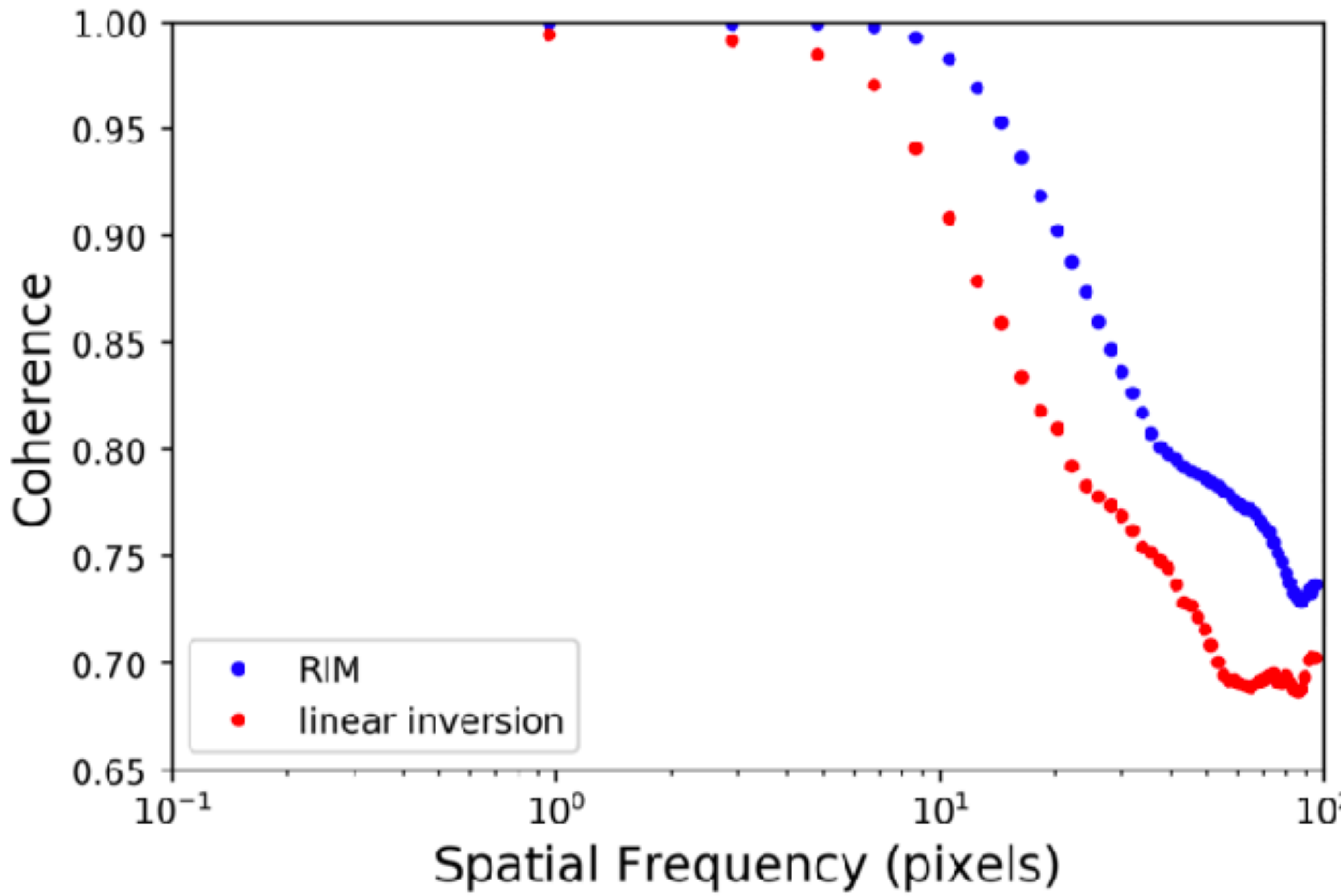
$$S = (L^T C_N^{-1} L + C_p^{-1})^{-1} L^T C_N^{-1} D$$

UNDISTORTED IMAGE OF THE BACKGROUND SOURCE WITH THE RECURRENT INFERENCE MACHINE (RIM)



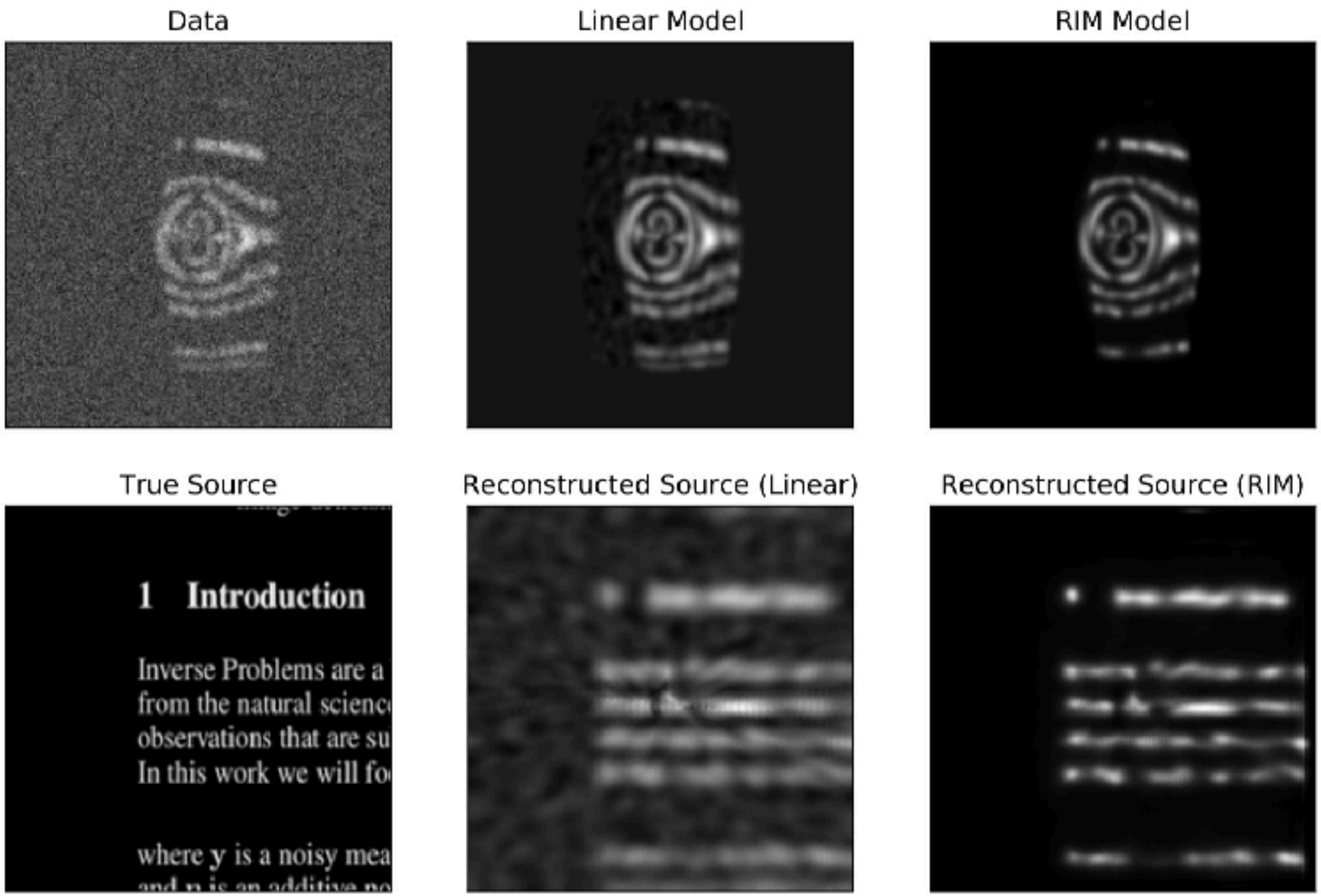
Morningstar, Perreault Levasseur et al., 2019

Background source reconstruction: Comparison to maximum likelihood methods

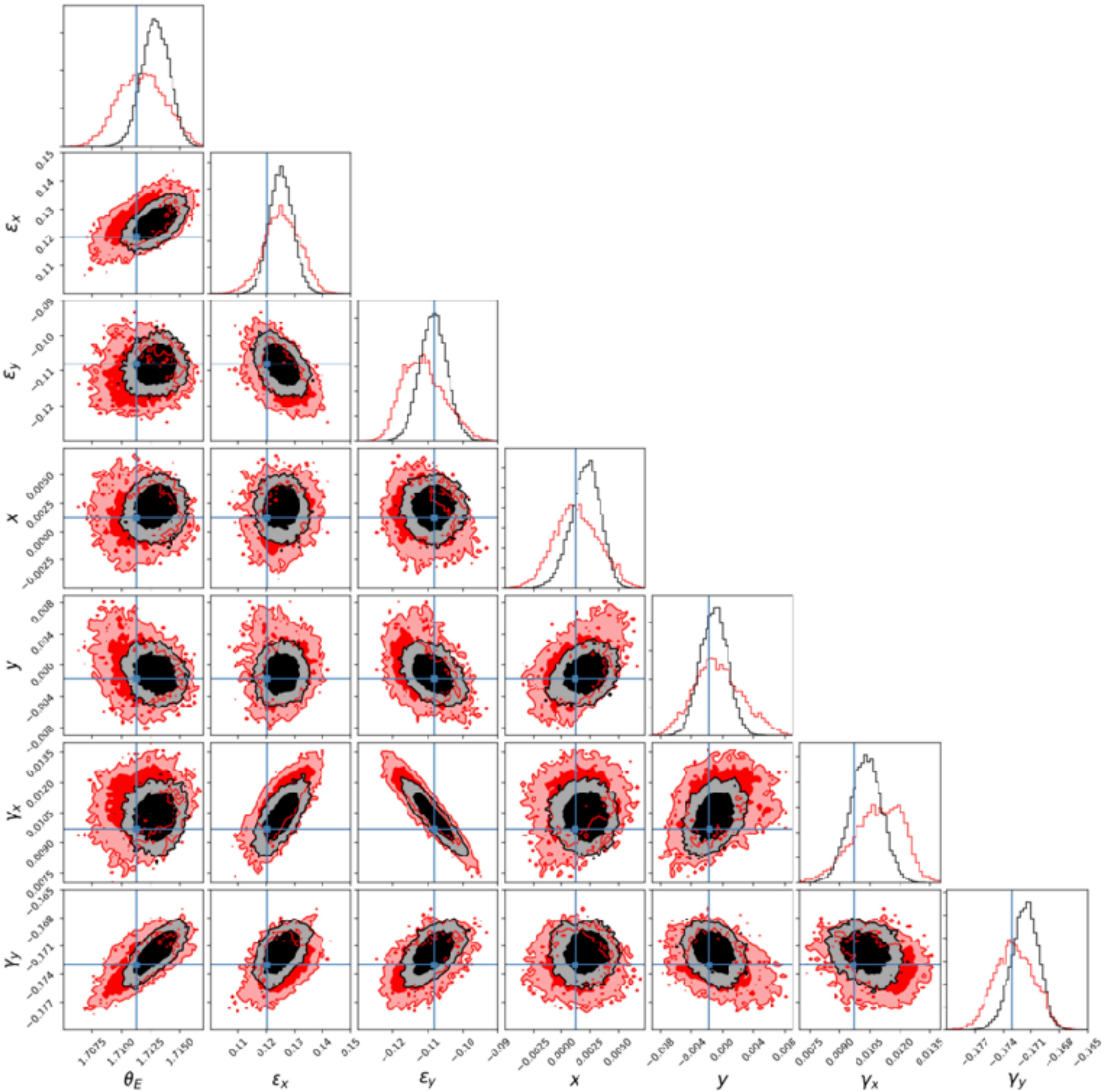


Morningstar, Perreault Levasseur et al., 2019

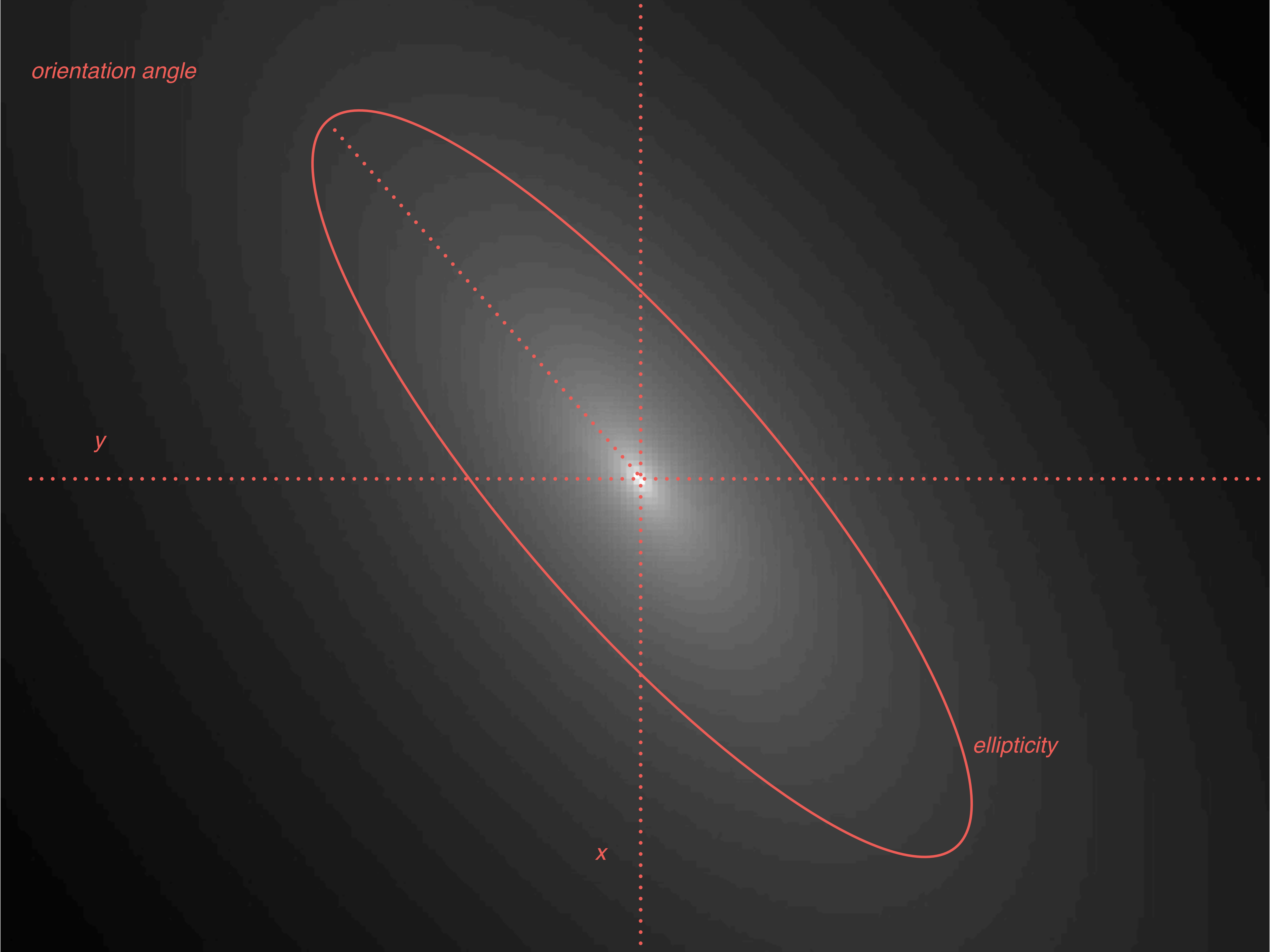
EXAMPLES OUTSIDE THE TRAINING DATA



Morningstar, Perreault Levasseur et al., 2019

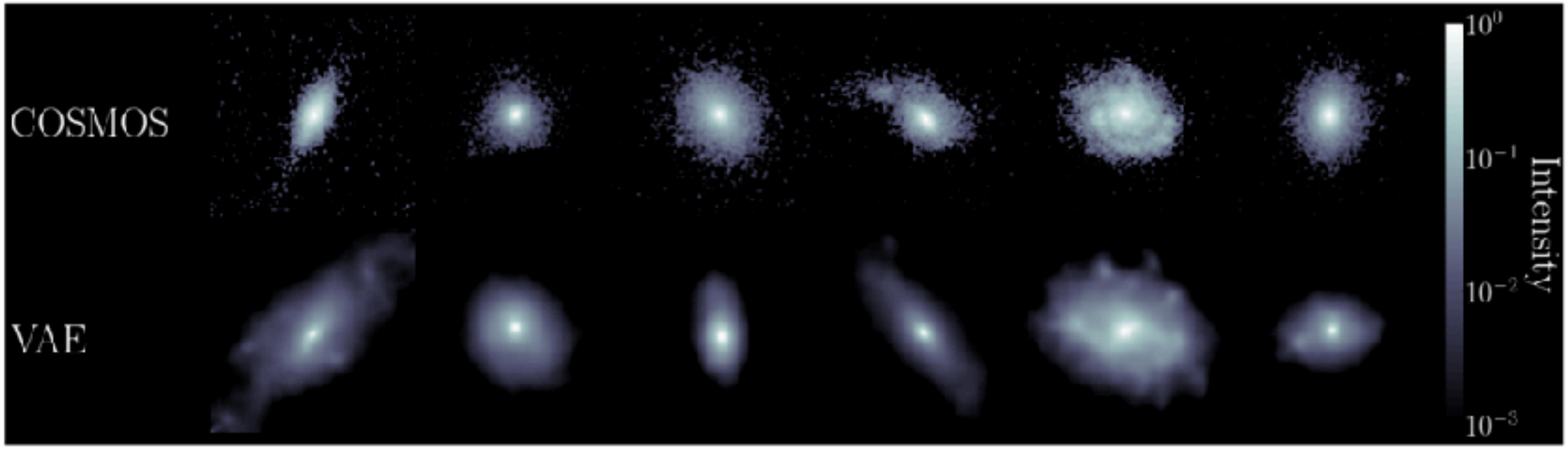


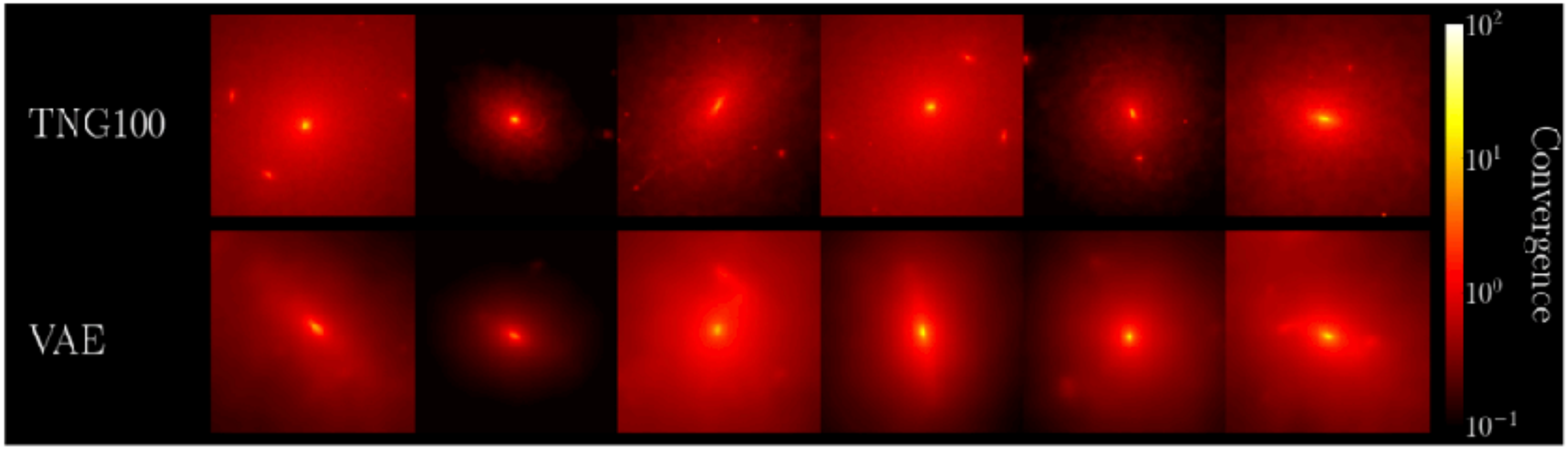
ε_x
ε_y
x
y
γ_x
γ_y
θ_E
ε_x
ε_y
x
y
γ_x
γ_y


orientation angle
y
x
ellipticity

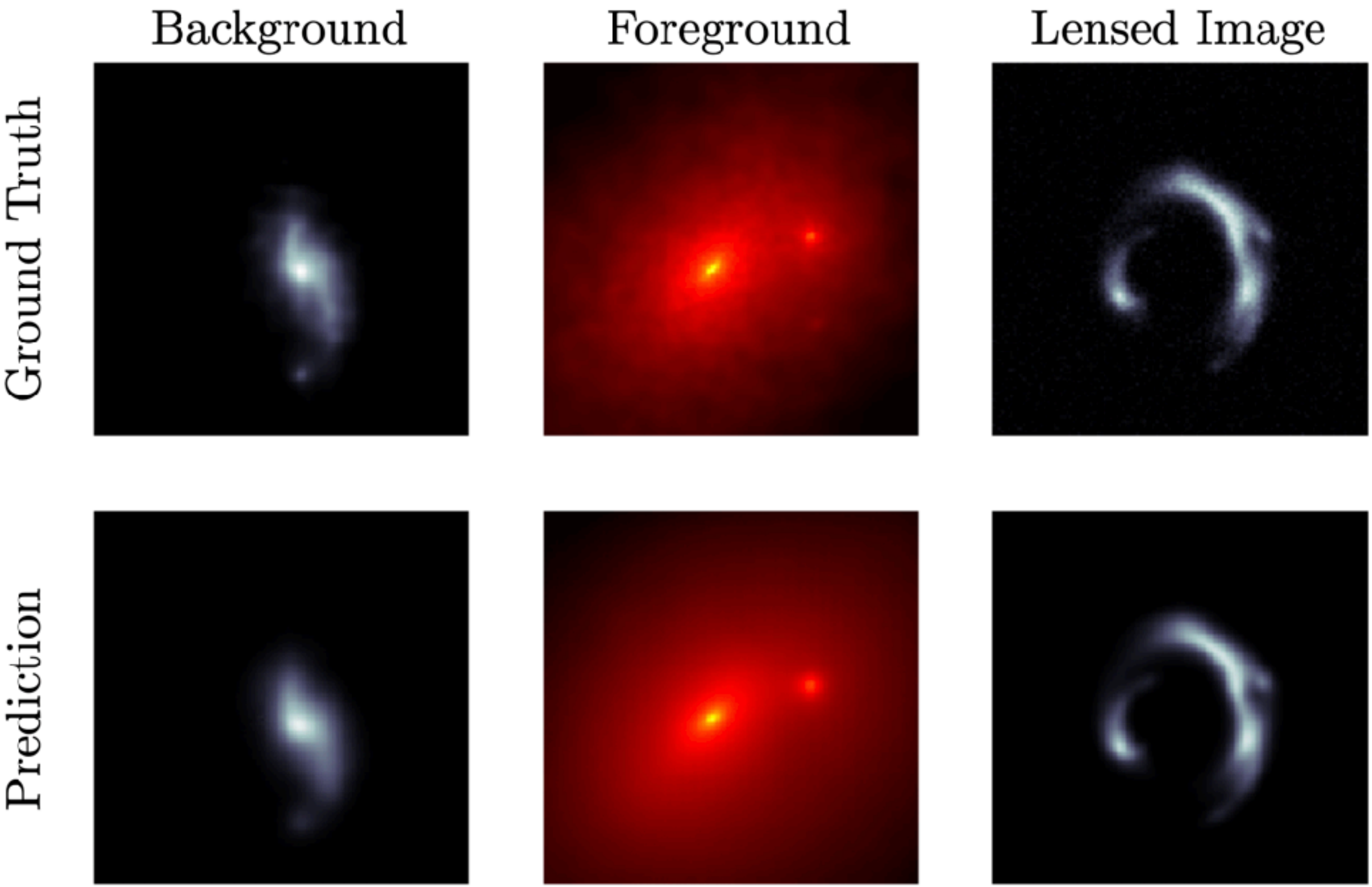
Simulated galaxies generated with a Variational Autoencoder


Alexandre Adam

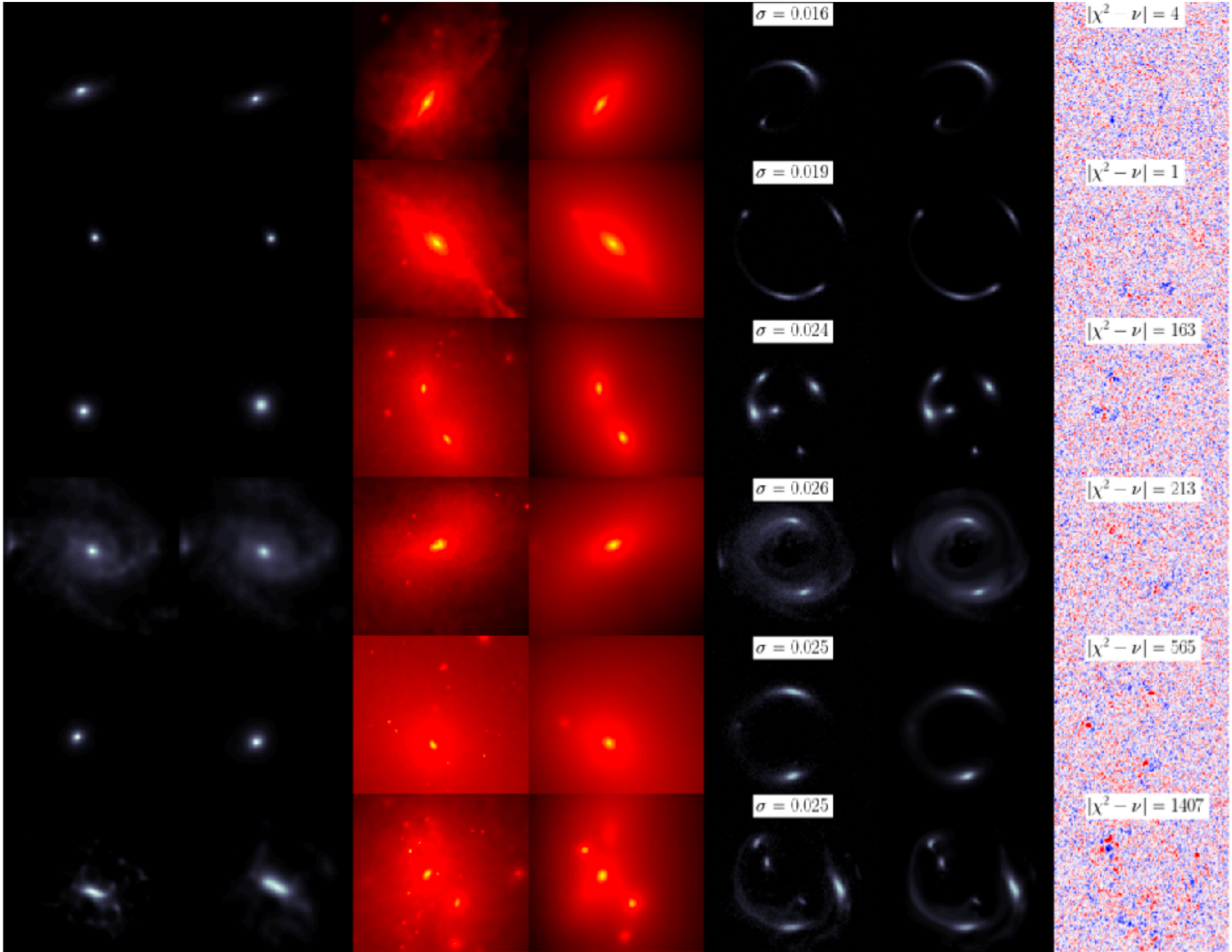




Training on hydrodynamical simulations

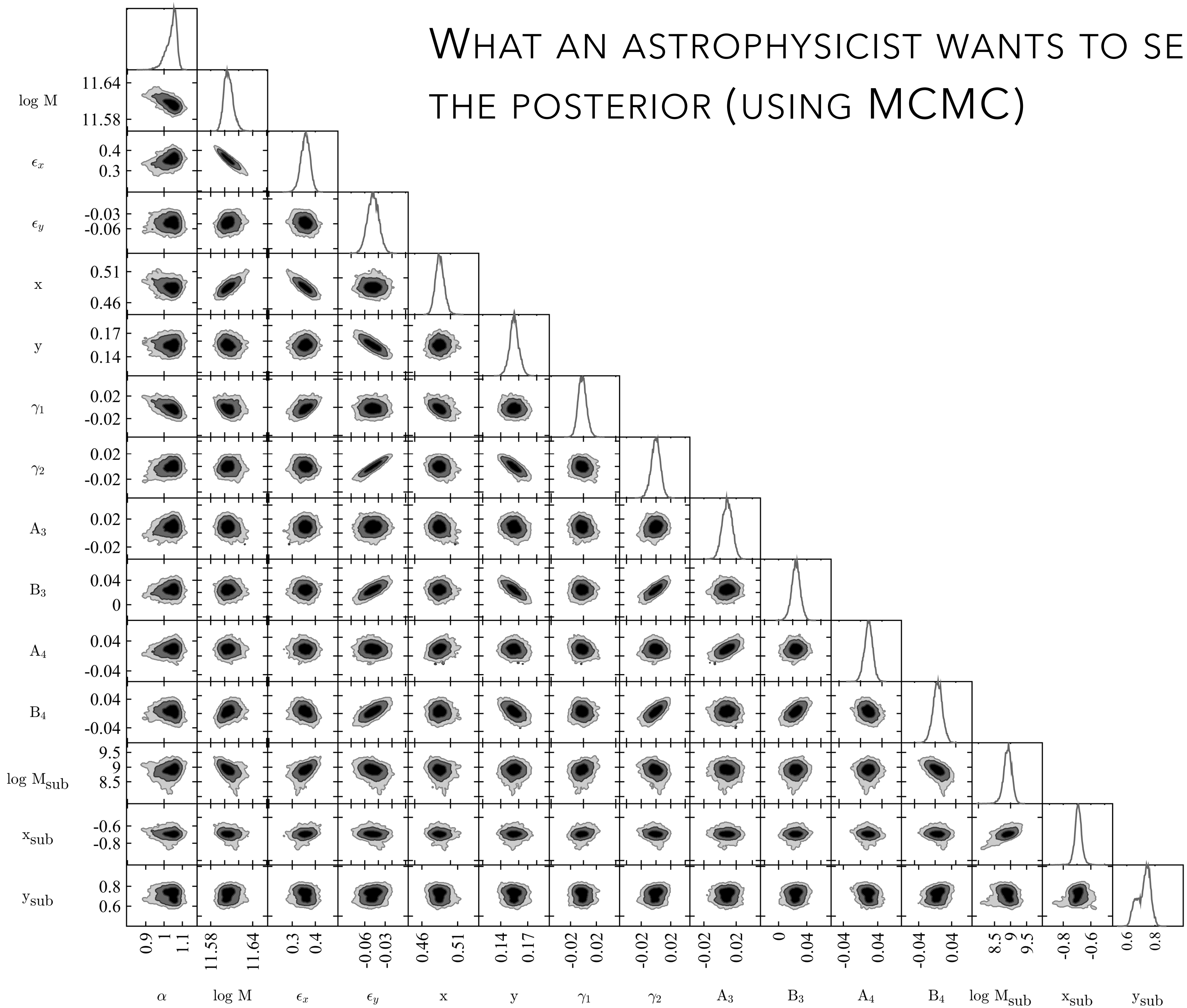


Adam, Perreault-Levasseur, Hezaveh, Welling, ApJ, 2023, arXiv:2301.04168



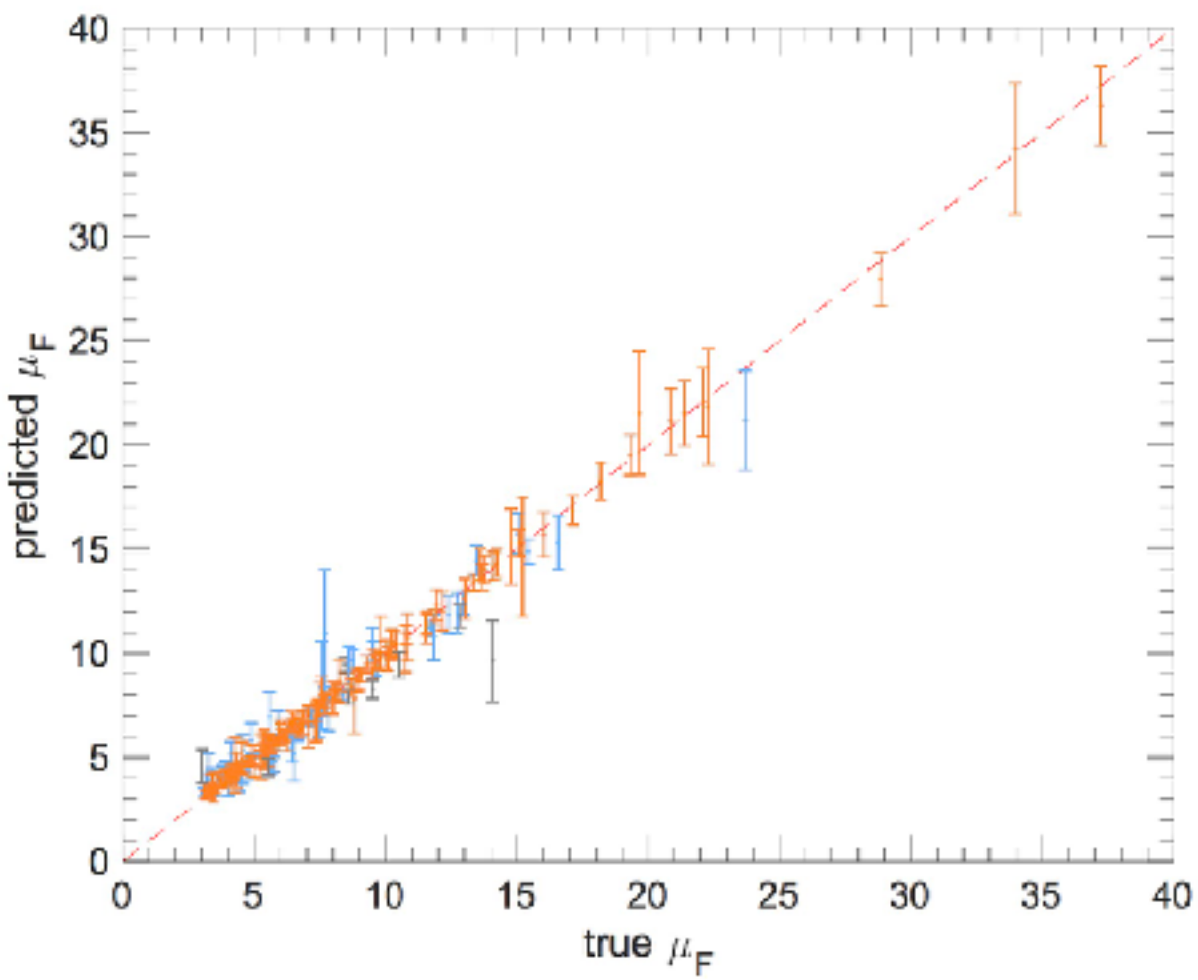
Adam, Perreault-Levasseur, Hezaveh, Welling, ApJ, 2023, arXiv:2301.04168

WHAT AN ASTROPHYSICIST WANTS TO SEE: THE POSTERIOR (USING MCMC)



Hezaveh, …,LPL, et al. ApJ 2016

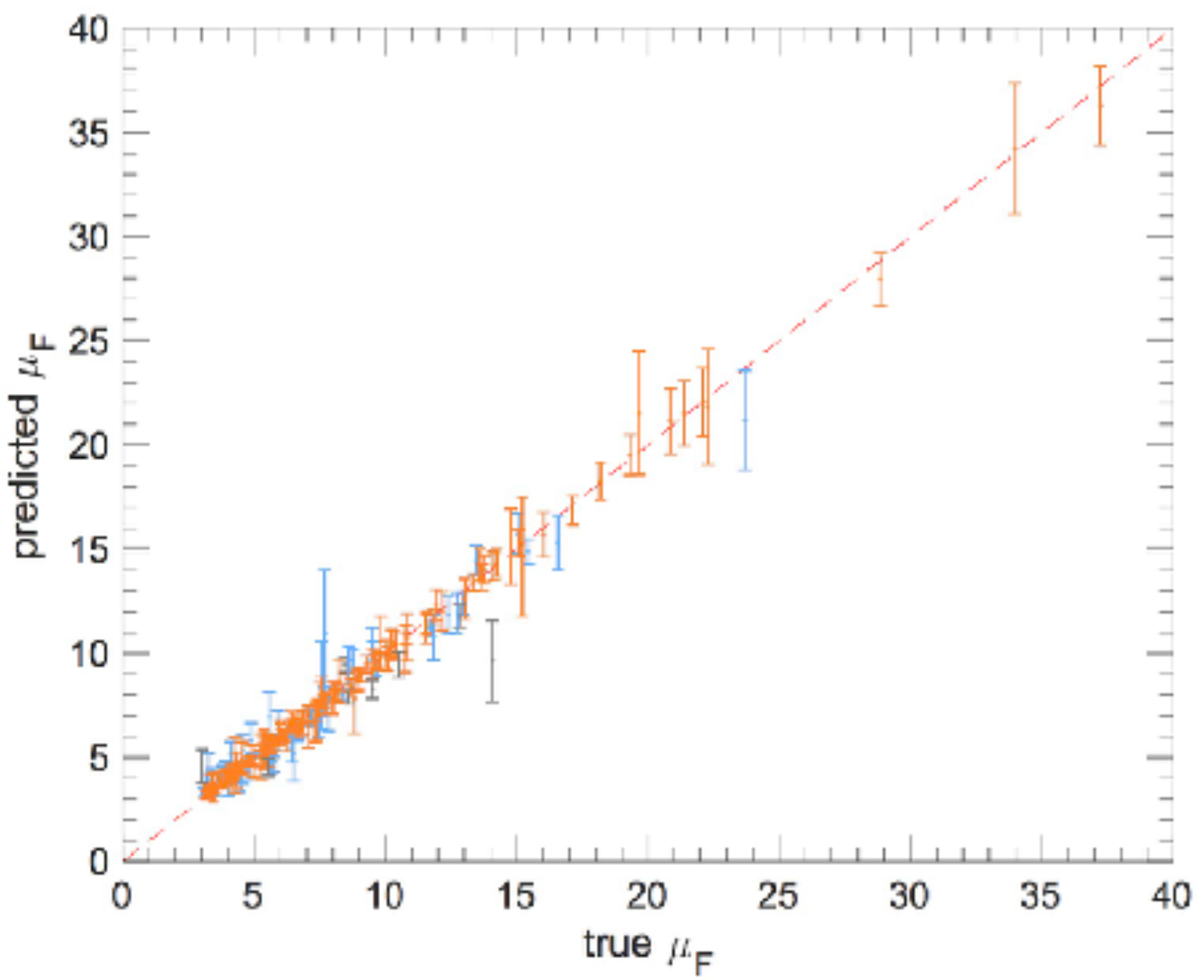
UNCERTAINTY ESTIMATION WITH APPROXIMATE BAYESIAN NEURAL NETWORKS

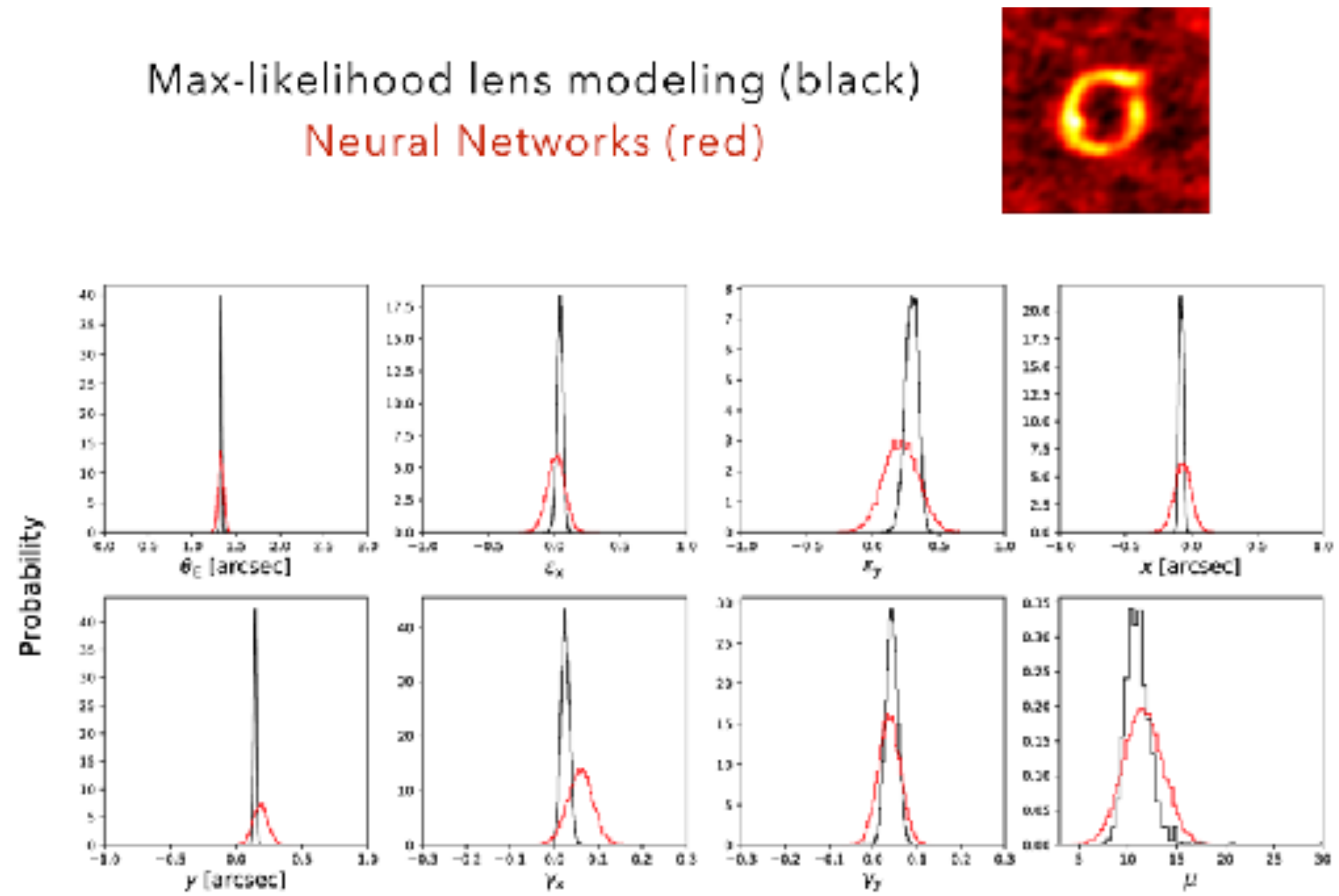


Perreault-Levasseur et al., 2017

Morningstar et al., 2018

UNCERTAINTY ESTIMATION WITH APPROXIMATE BAYESIAN NEURAL NETWORKS



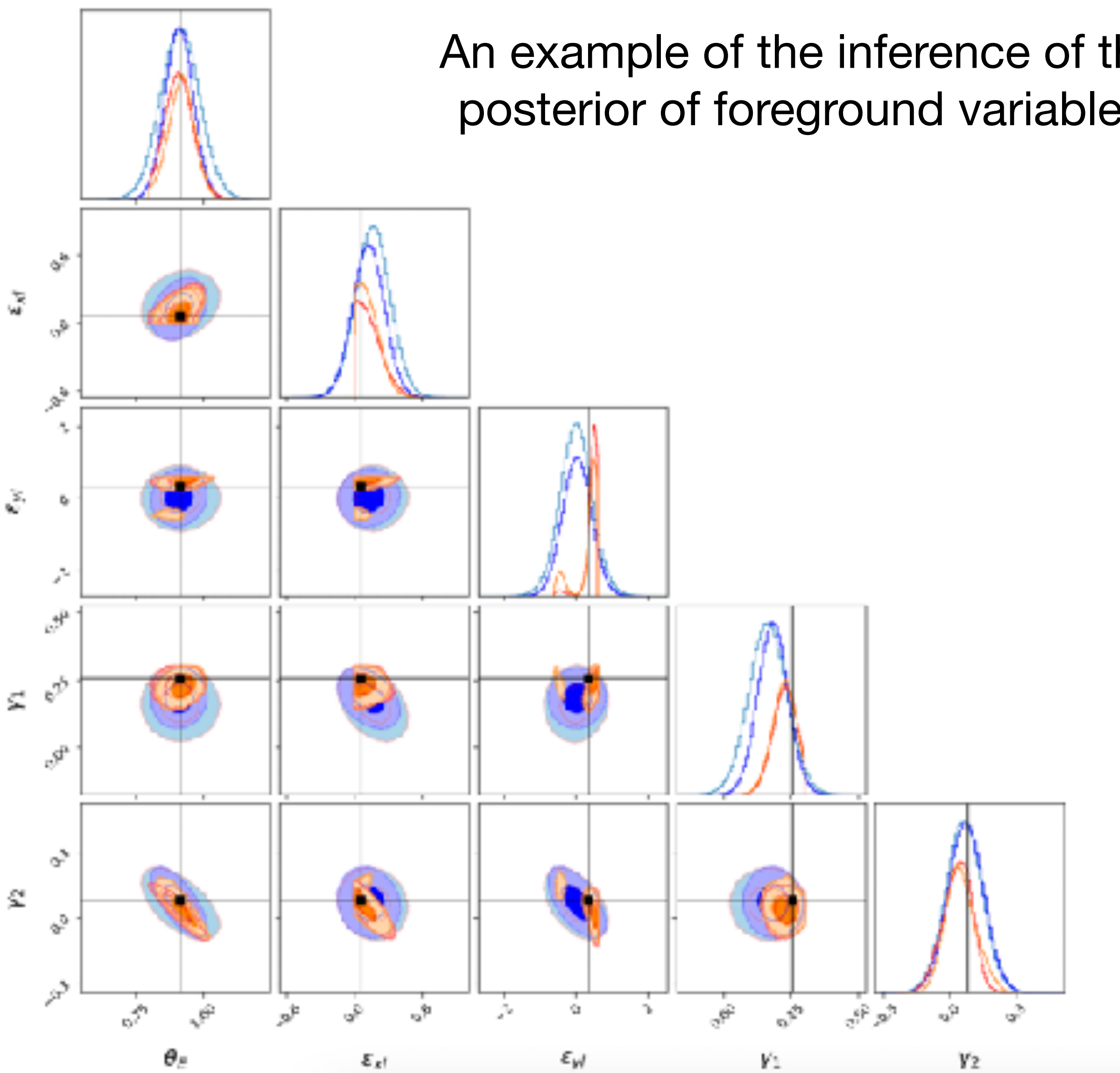


Perreault-Levasseur et al., 2017

Morningstar et al., 2018

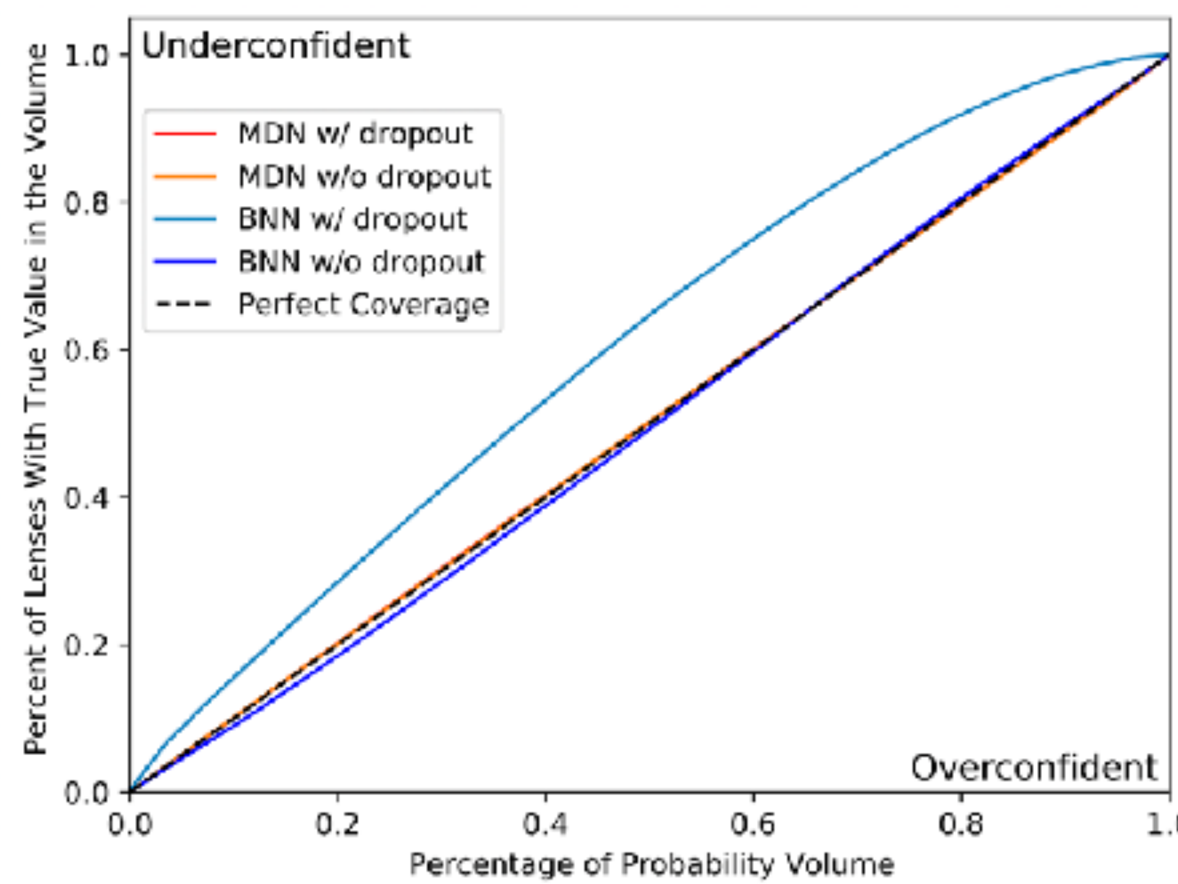
UNCERTAINTY ESTIMATION WITH SIMULATION-BASED INFERENCE METHODS

Ronan Legin



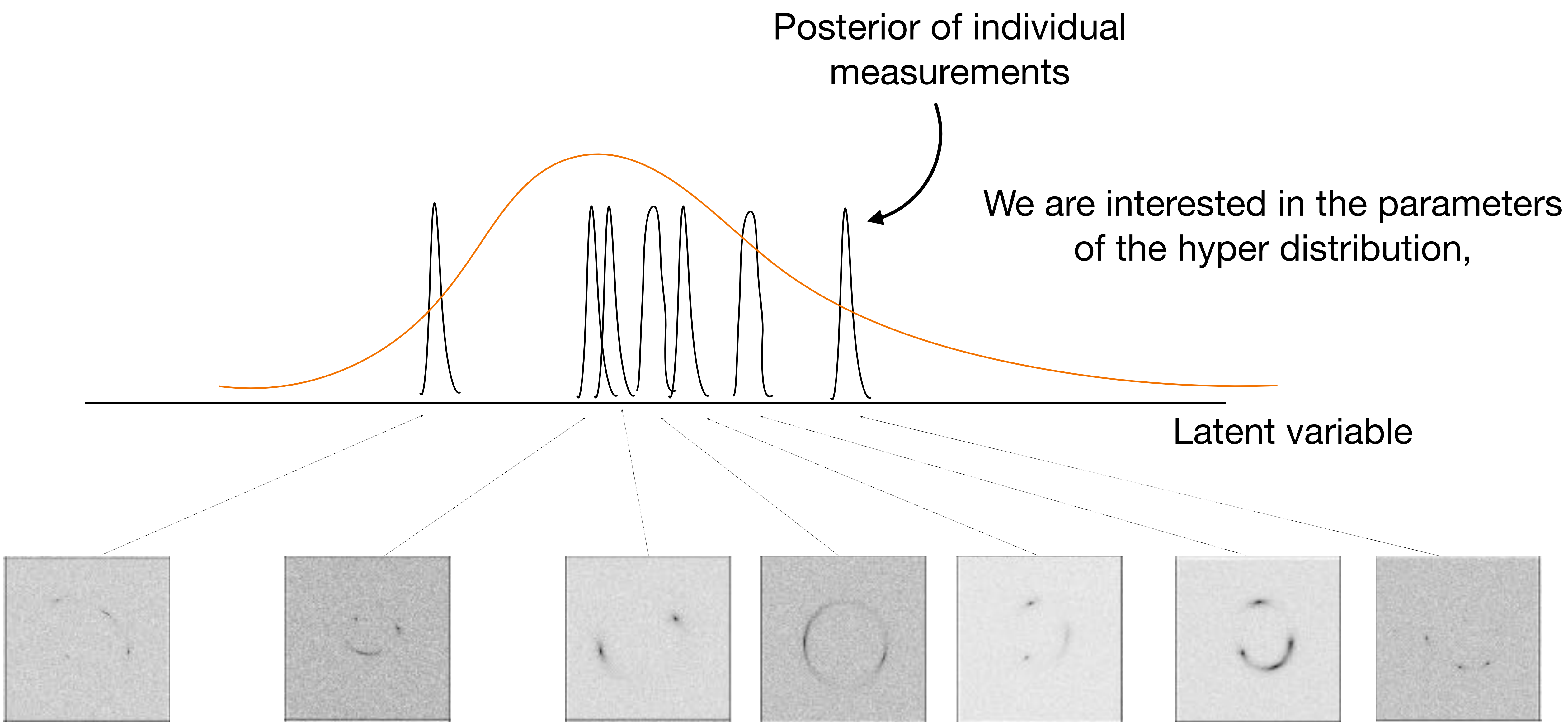
An example of the inference of the posterior of foreground variables

Coverage probabilities

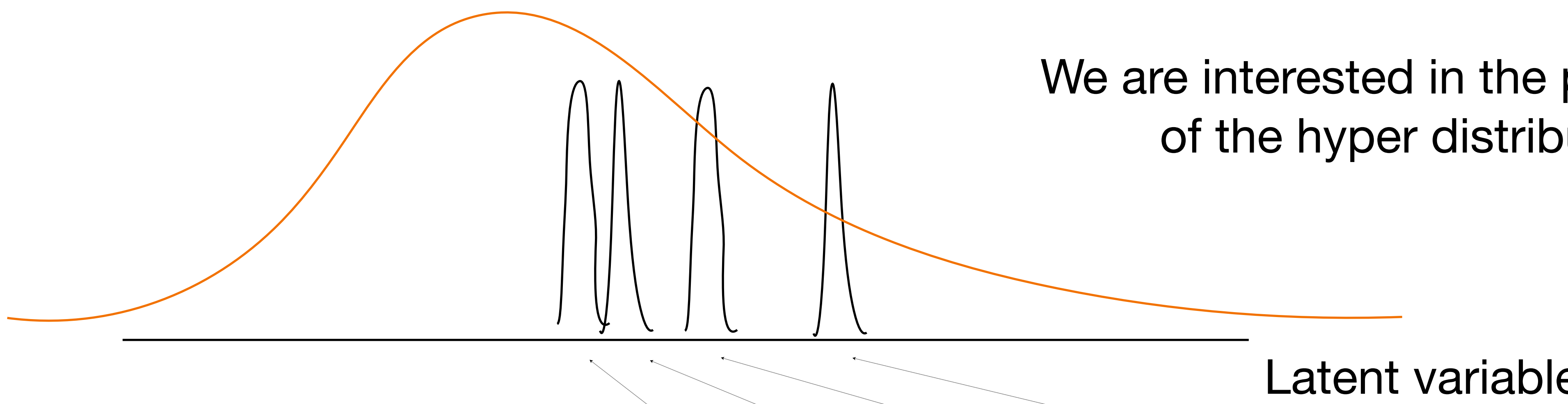


R. Legin, Y. Hezaveh, L. Perreault Levasseur, B. Wandelt, ApJ 2022

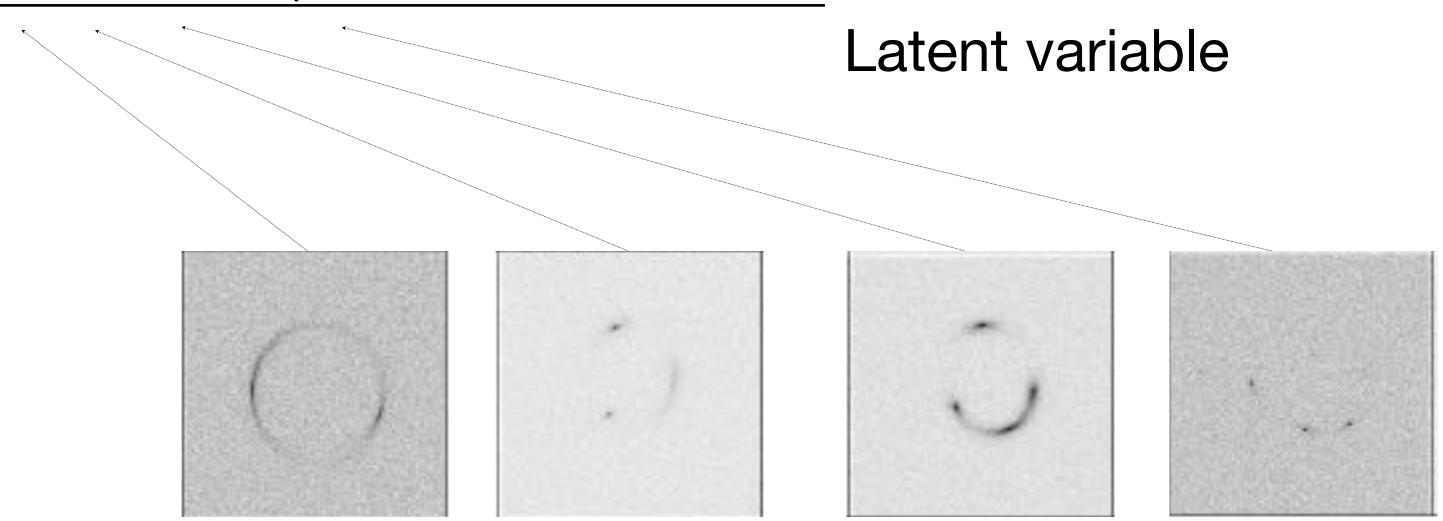
Hierarchical Bayesian inference



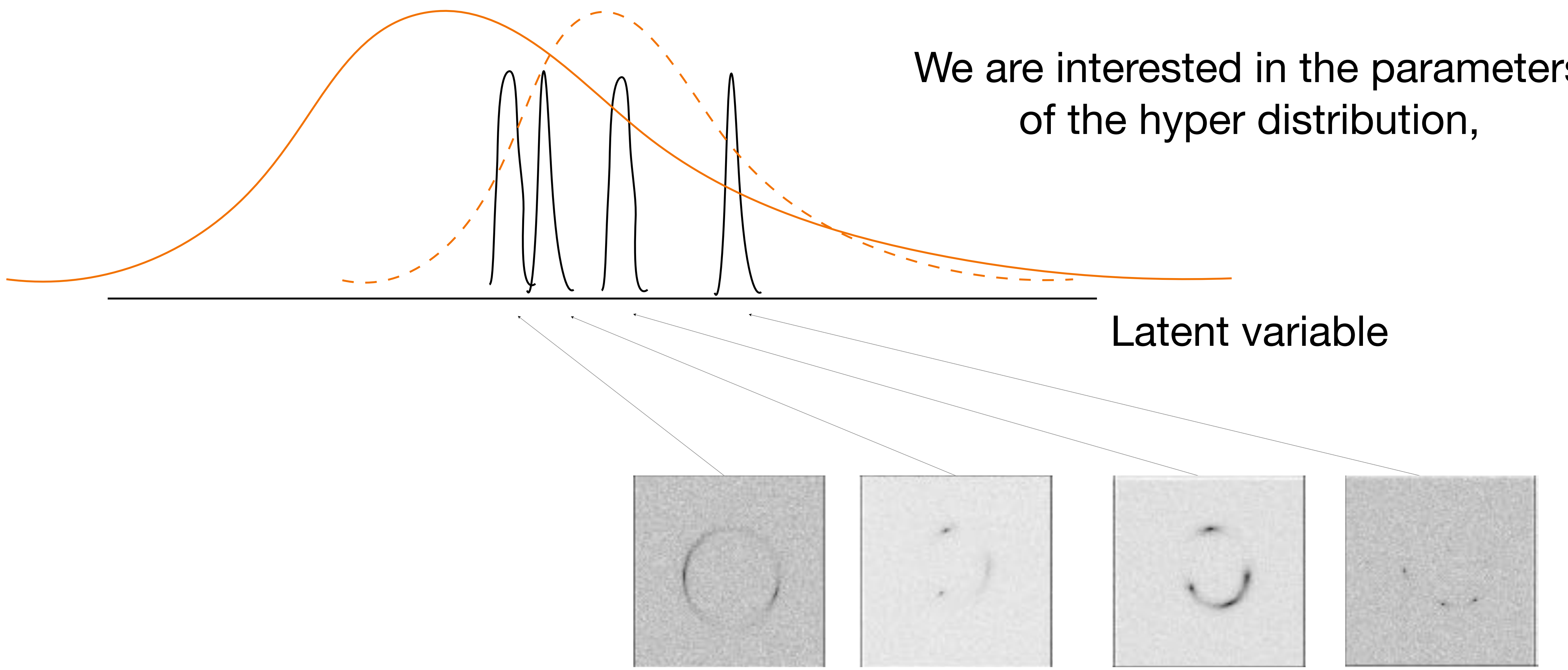
Hierarchical Bayesian inference



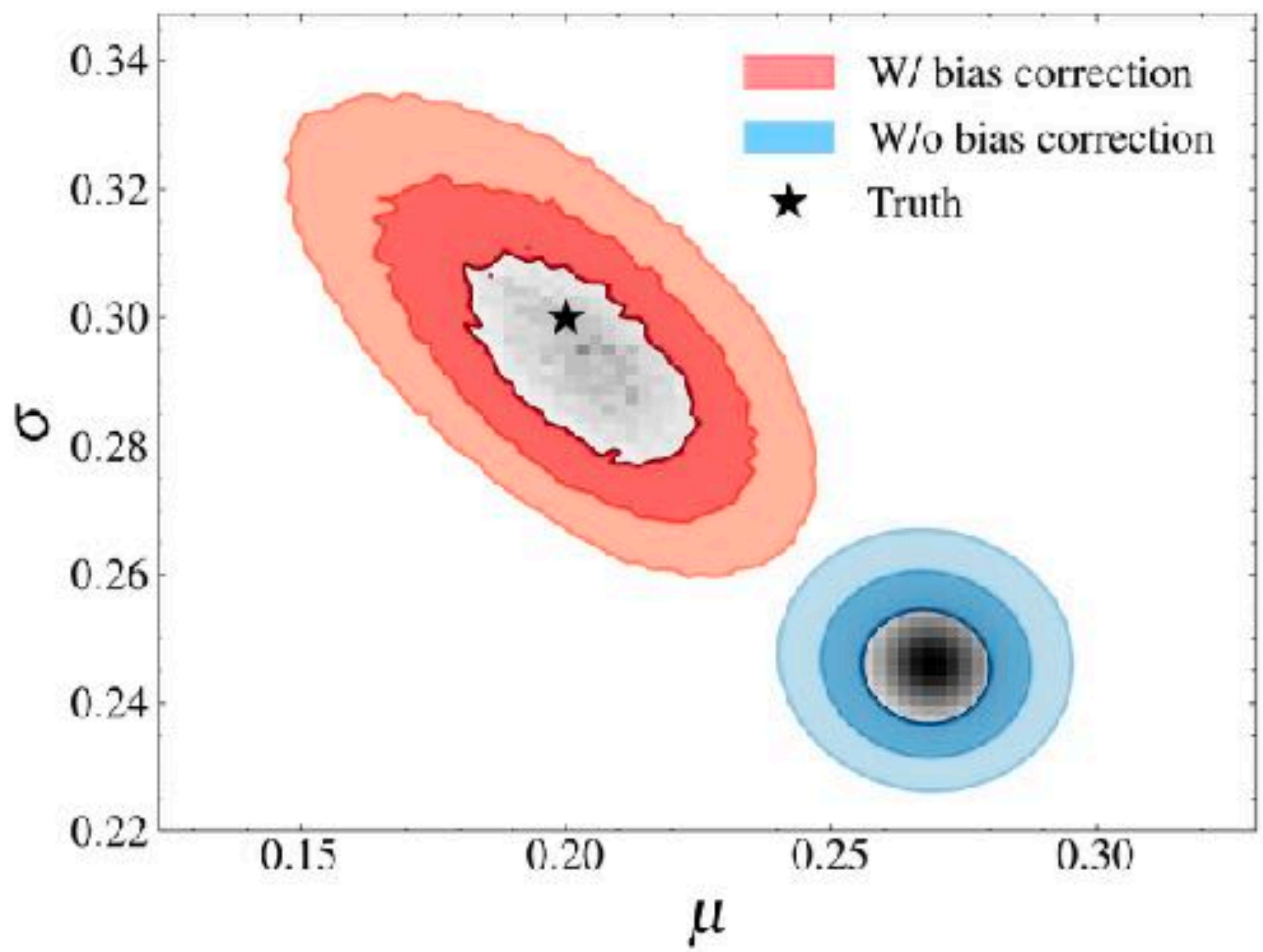
We are interested in the parameters of the hyper distribution,



Hierarchical Bayesian inference



Hierarchical Bayesian inference



Ronan Legin

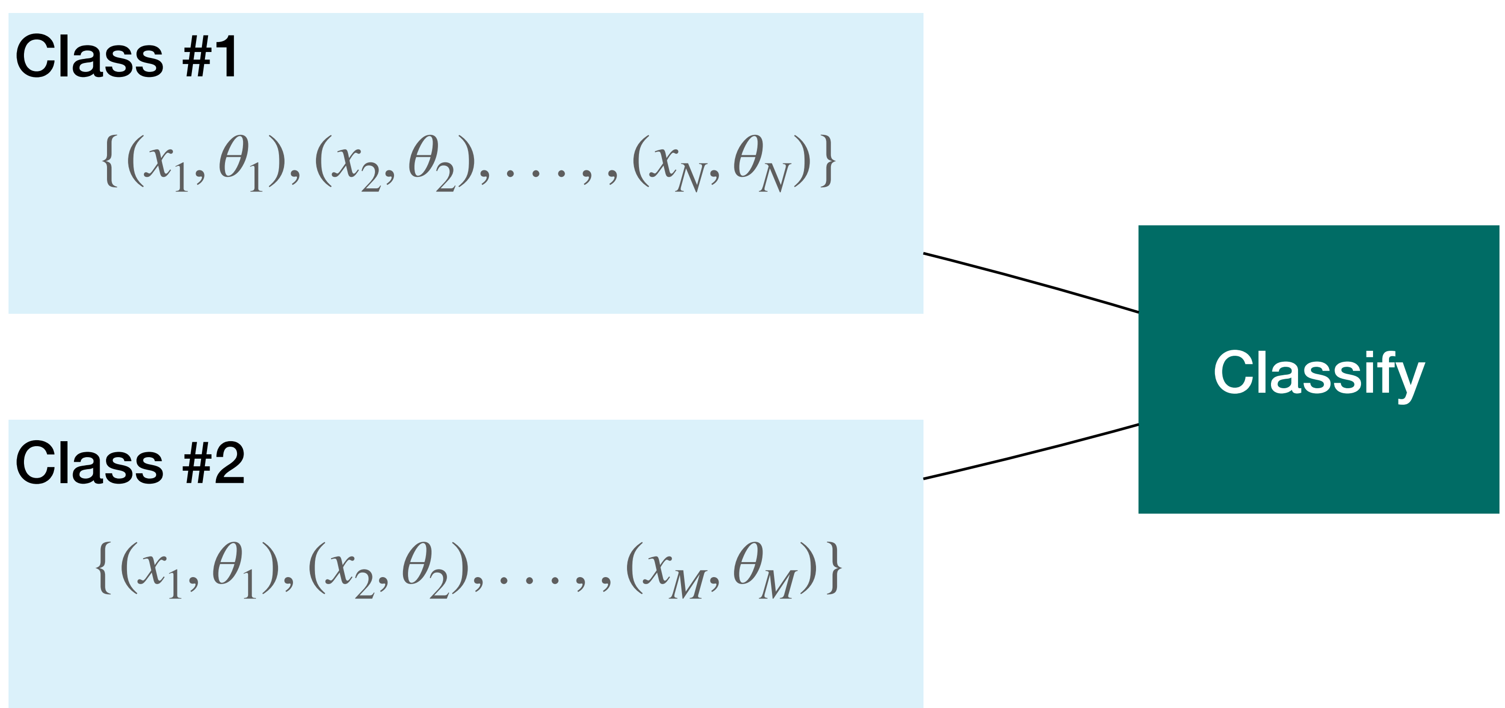
Connor Stone

Legin, Stone, Hezaveh, Perreault Levasseur,
ICML 2022 - ML4Astro Workshop arXiv:2207.04123

Neural Ratio Estimators

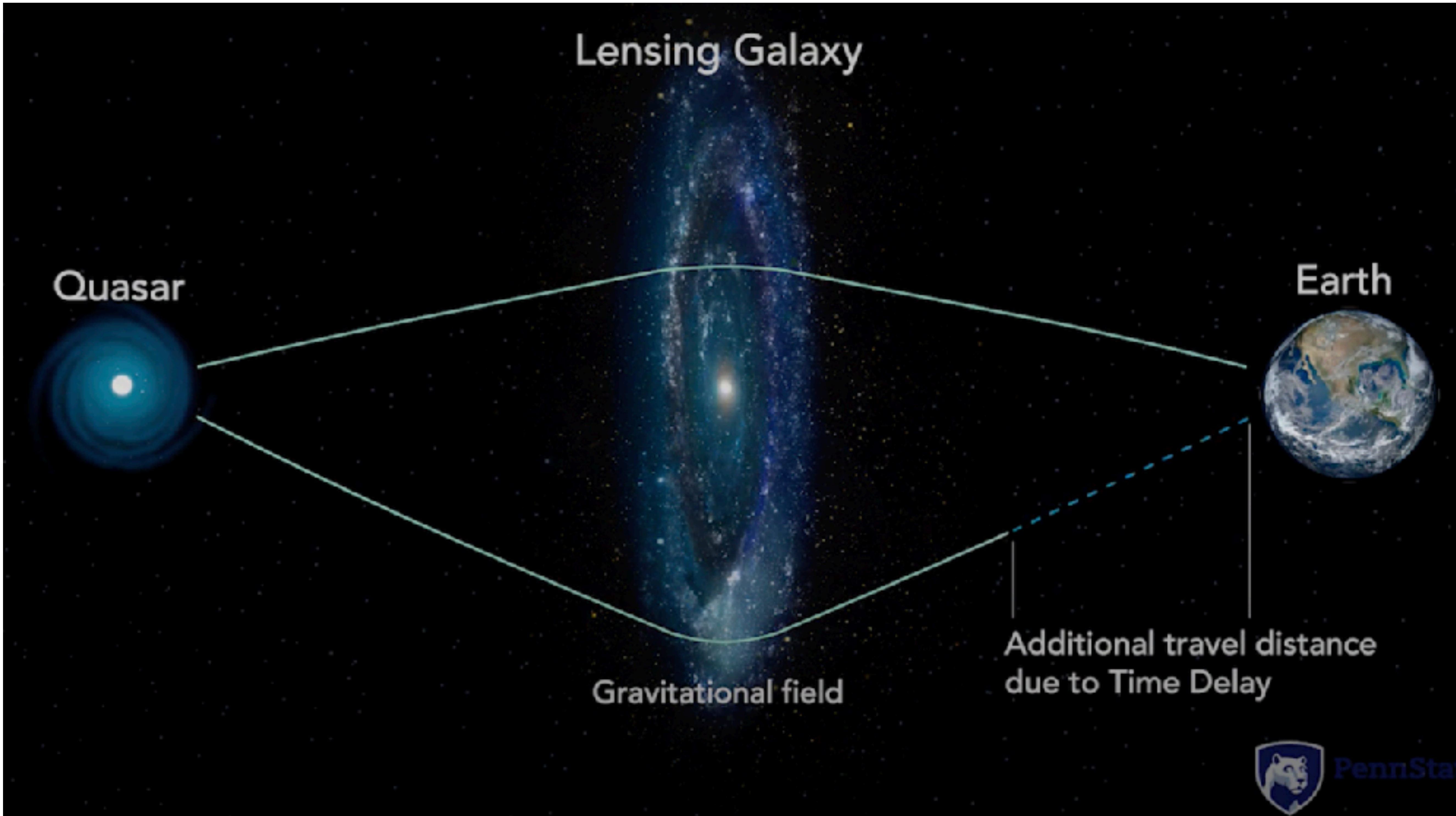
$P(x, \theta)$

$P(x)P(\theta)$



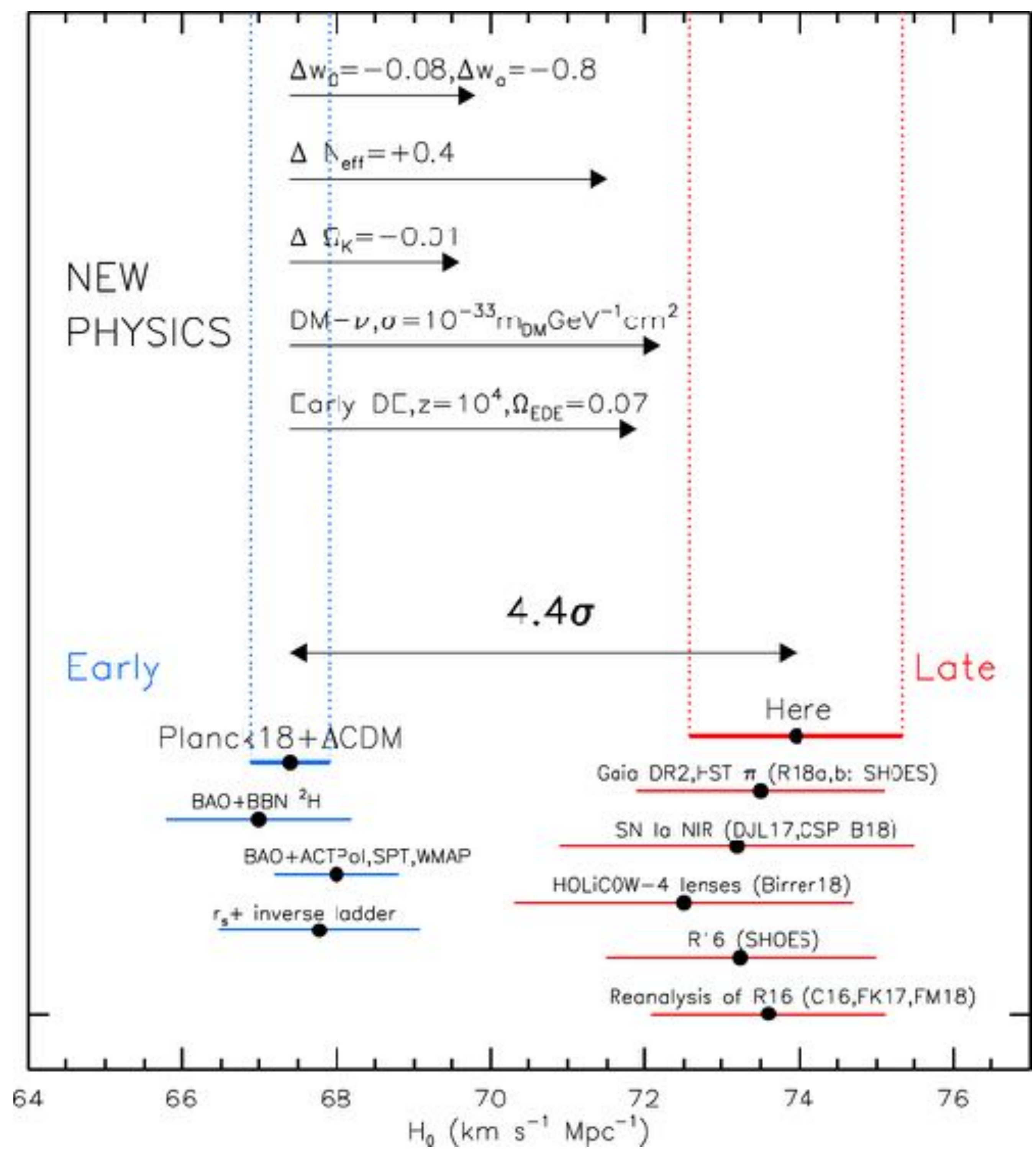
$$r(x, \theta) = \frac{p(x, \theta)}{p(x)p(\theta)} = \frac{p(\theta \,|\, x)}{p(\theta)}$$

H0 Inference with Time Delay Cosmography



THE HUBBLE CONSTANT

DISCREPANCY BETWEEN MEASUREMENTS



Adam G. Riess *et al* 2019
ApJ **876** 85

H0 Inference with Neural Ratio Estimators

$P(\theta, x)$

$p($ 

$,$  $)$

$P(\theta)P(x)$

$p($ 

$)p($  $)$

Class #1

$\{(x_1, \theta_1), (x_2, \theta_2), \ldots, , (x_N, \theta_N)\}$

Class #2

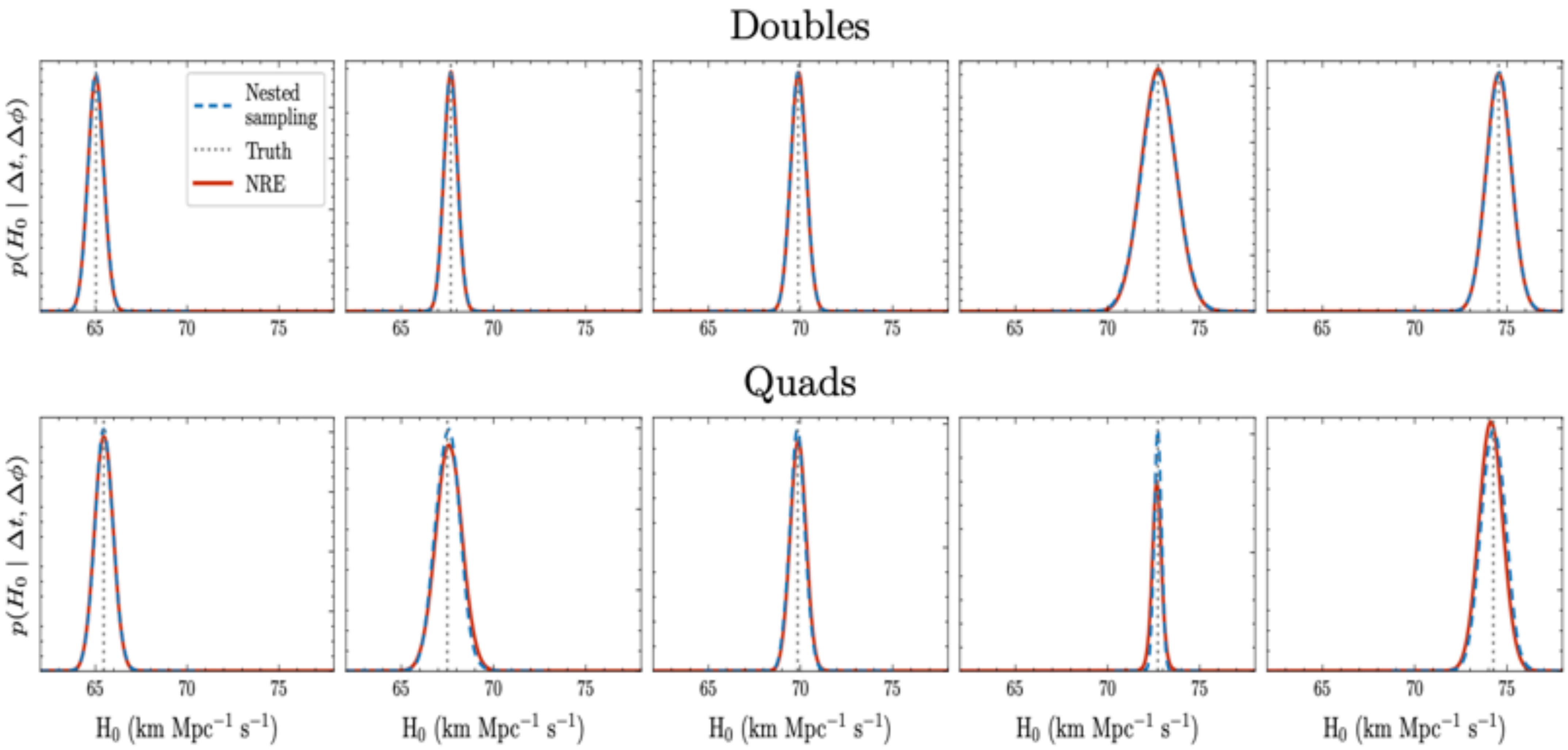
$\{(x_1, \theta_1), (x_2, \theta_2), \ldots, , (x_M, \theta_M)\}$

Classify

$$r(x, \theta) = \frac{p(x, \theta)}{p(x)p(\theta)} = \frac{p(\theta \,|\, x)}{p(\theta)}$$

H0 Inference with Neural Ratio Estimators

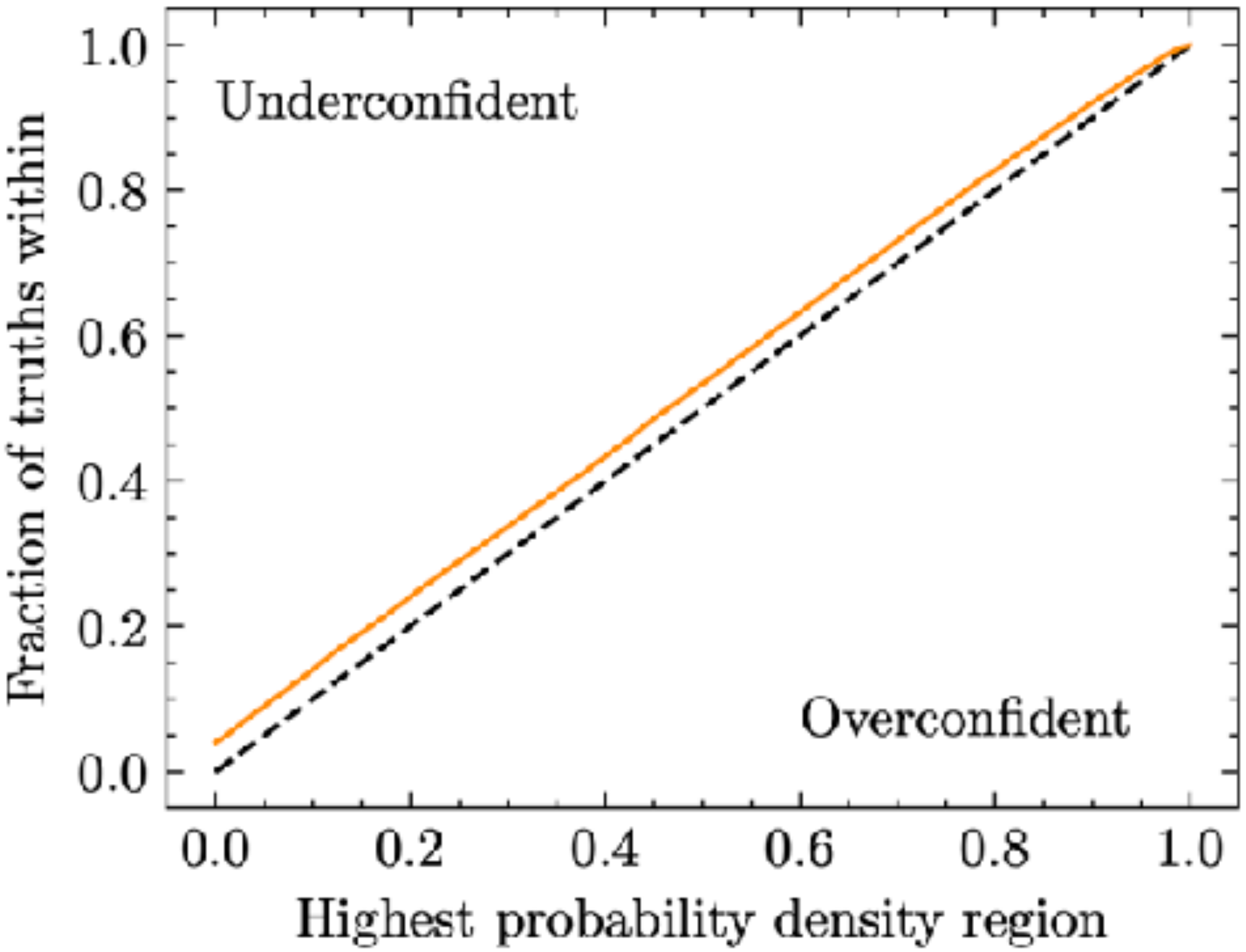
Ève Campeau-Poirier



Campeau-Poirier et al. ICML 2023 ML4Astro Workshop, arXiv:2309.16063

H0 Inference with Neural Ratio Estimators

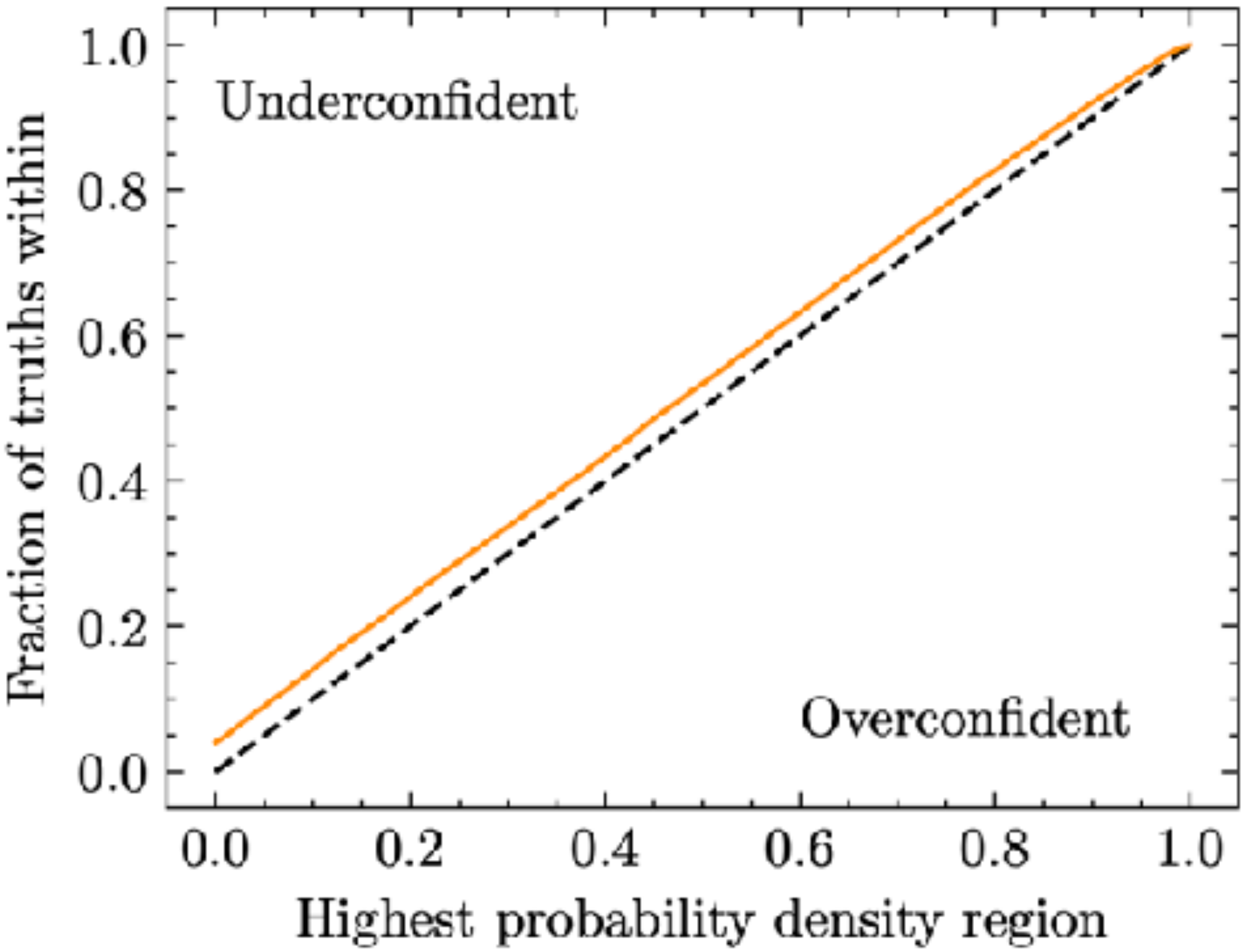
Ève Campeau-Poirier

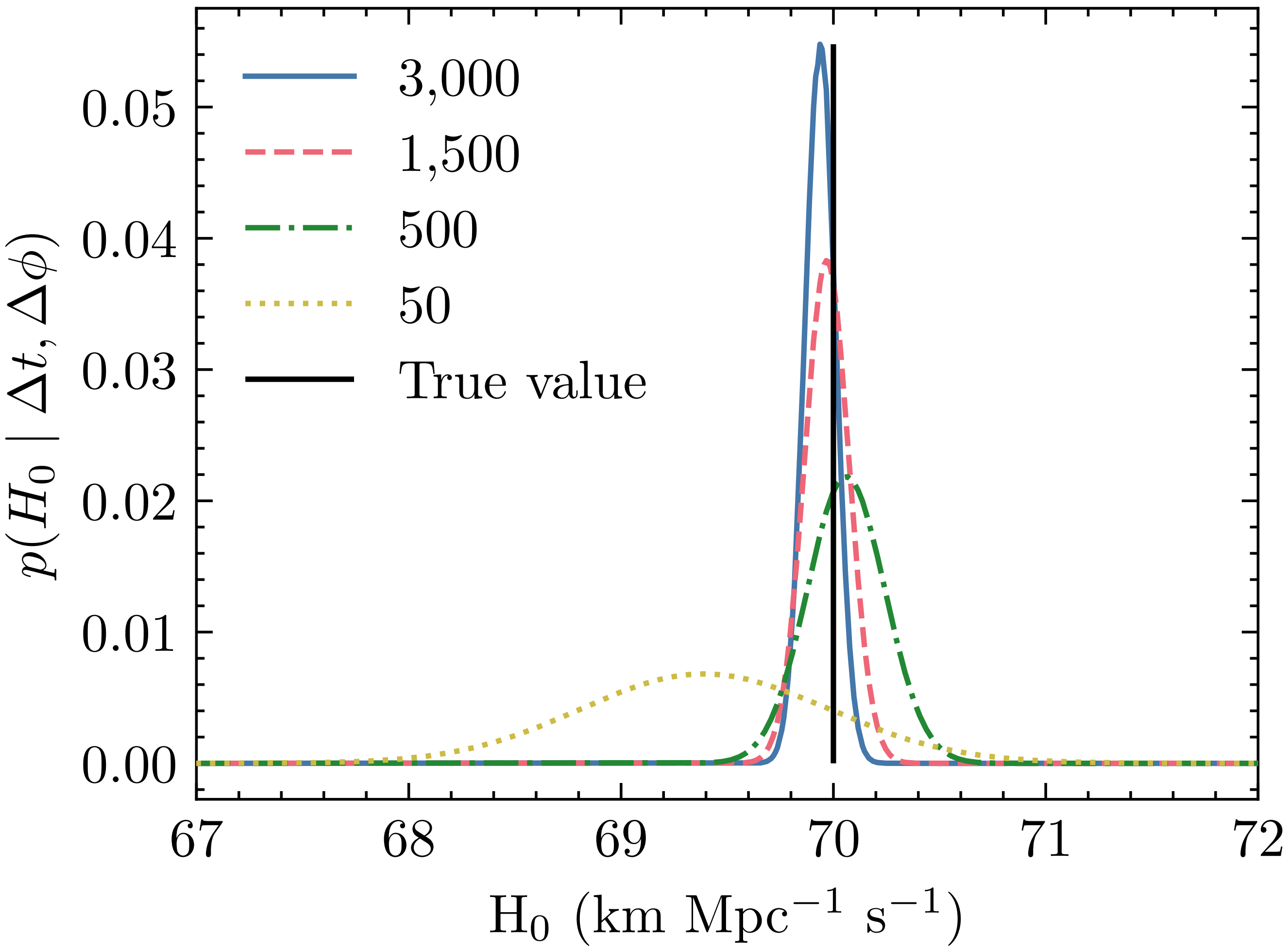


Campeau-Poirier et al. ICML 2023 ML4Astro Workshop, arXiv:2309.16063

H0 Inference with Neural Ratio Estimators

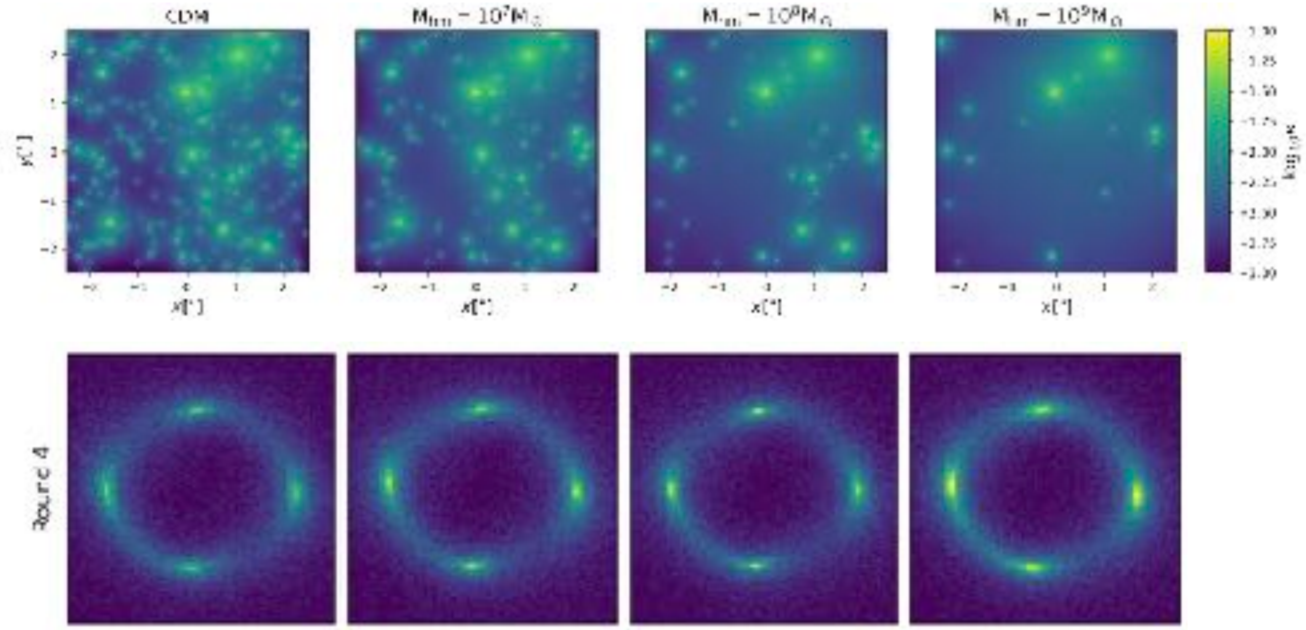
Ève Campeau-Poirier

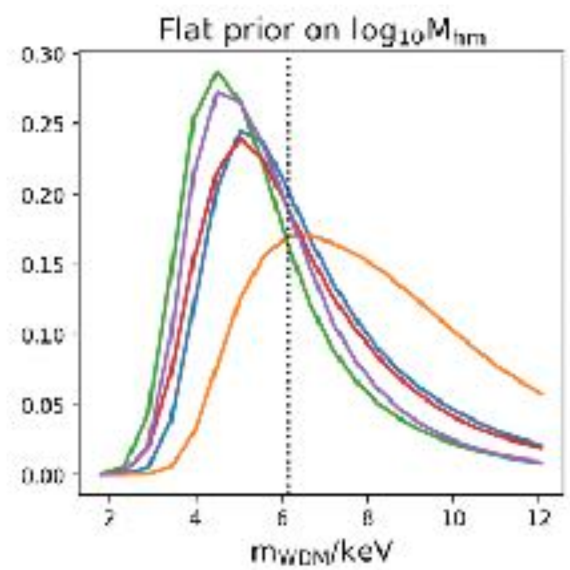




Campeau-Poirier et al. ICML 2023 ML4Astro Workshop, arXiv:2309.16063

Estimating the dark matter particle temperature with Neural Ratio Estimators





Adam Coogan

Anau Montel, Coogan et al. 2022, arXiv:2205.09126

Coogan et al. , NeurIPS 2020 ML4PS Workshop

Tackling an unsolved problem: high dimensional inference

How do we infer the posteriors of high-dimensional parameters (e.g., an image or spectra)?

Obstacles:

1) How do we encode complex priors

2) How we sample such high-dimensional posteriors (even if we could compute them)

LEARNING THE PRIOR EXPLICITLY

Can we learn our high-dimensional prior explicitly from data?
i.e. can we learn a generative model that will produce samples from that distribution?

Learning the Prior Explicitly

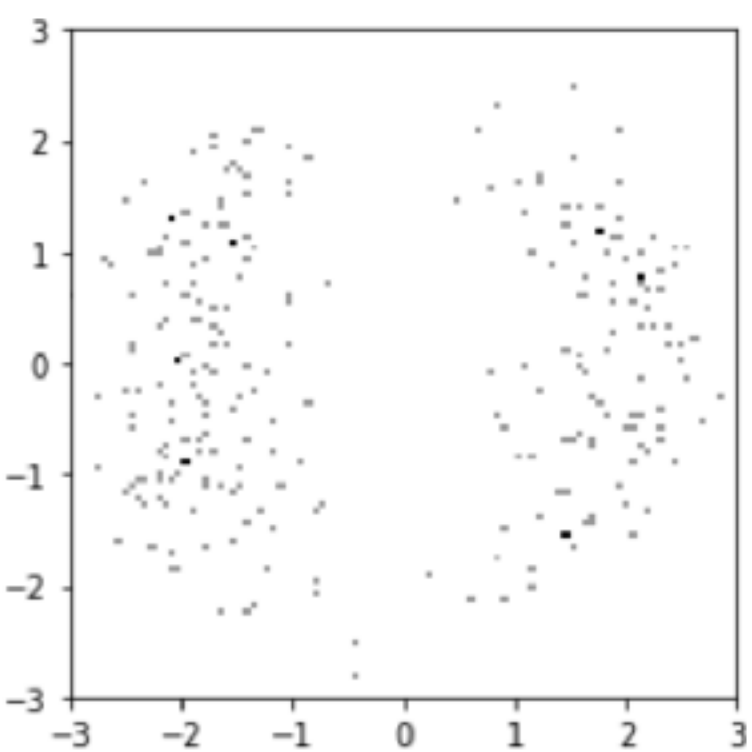
Can we learn our high-dimensional prior explicitly from data?
i.e. can we learn a generative model that will produce samples from that distribution?

How can we do this from samples (e.g. data)? Modeling the density?

LEARNING THE PRIOR EXPLICITLY

Can we learn our high-dimensional prior explicitly from data?
i.e. can we learn a generative model that will produce samples from that distribution?

How can we do this from samples (e.g. data)? Modeling the density?



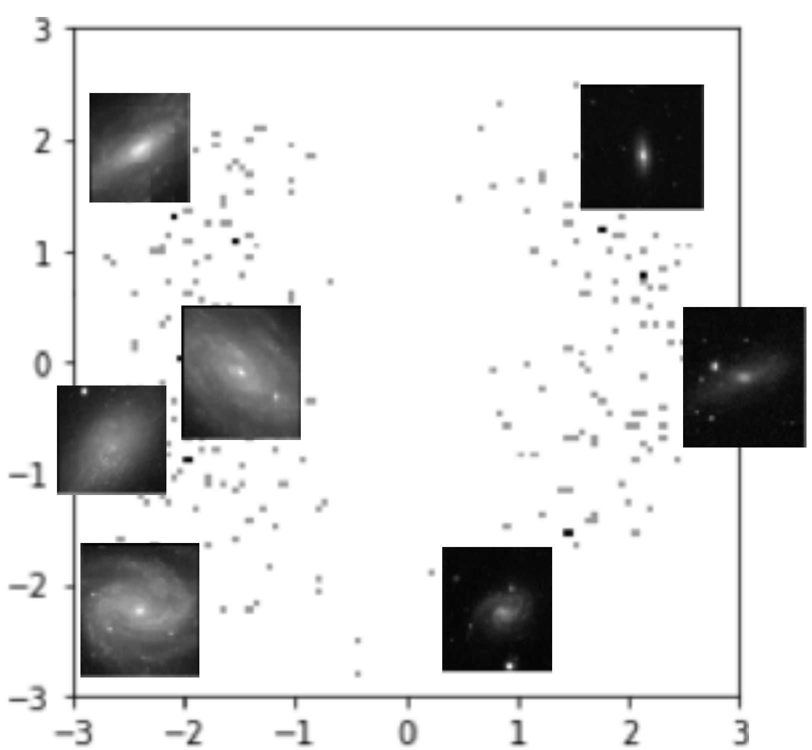
Training data

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_n,\} \overset{iid}{\sim} \pi_{data}(\mathbf{x})$$

LEARNING THE PRIOR EXPLICITLY

Can we learn our high-dimensional prior explicitly from data?
i.e. can we learn a generative model that will produce samples from that distribution?

How can we do this from samples (e.g. data)? Modeling the density?



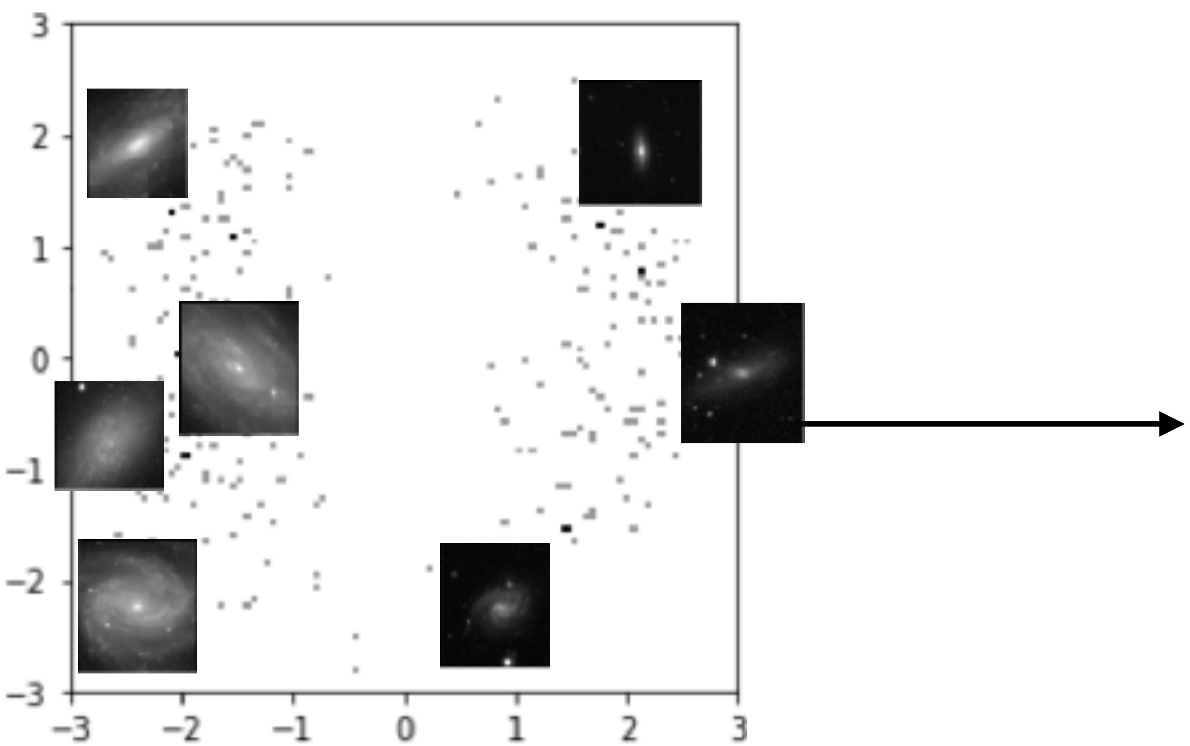
Training data

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_n,\} \overset{iid}{\sim} \pi_{data}(\mathbf{x})$$

LEARNING THE PRIOR EXPLICITLY

Can we learn our high-dimensional prior explicitly from data?
i.e. can we learn a generative model that will produce samples from that distribution?

How can we do this from samples (e.g. data)? Modeling the density?



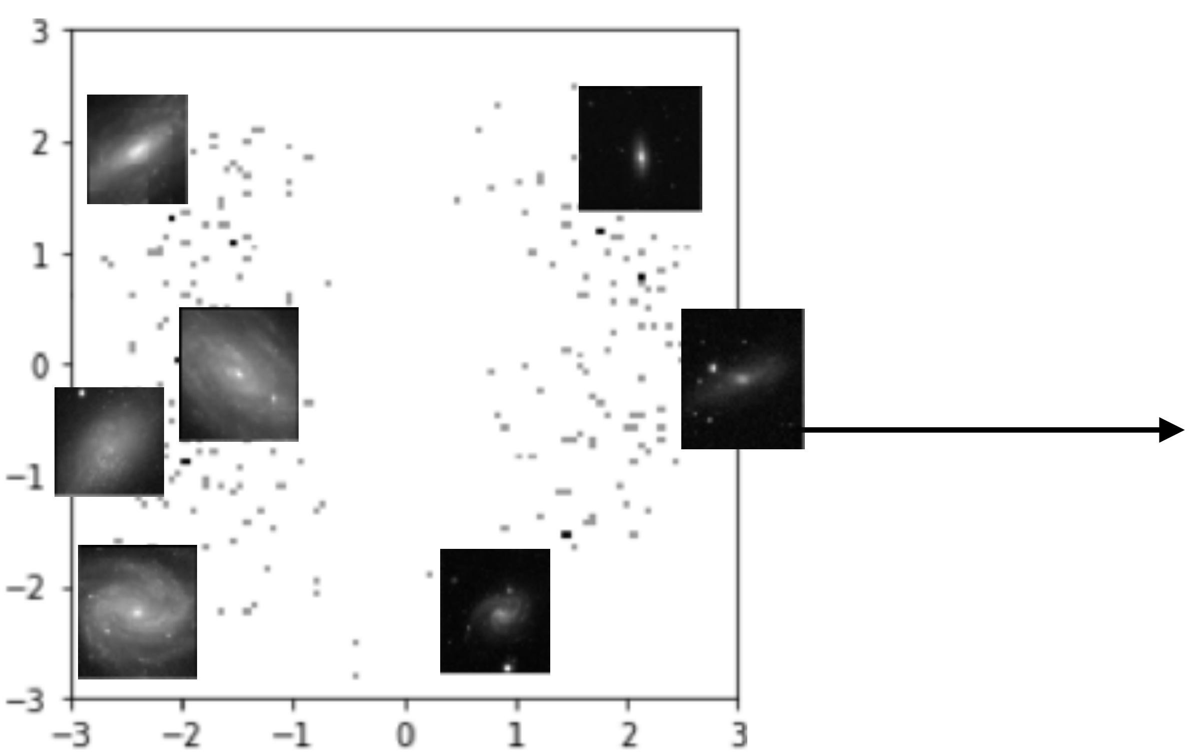
Training data

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_n,\} \overset{iid}{\sim} \pi_{data}(\mathbf{x})$$

LEARNING THE PRIOR EXPLICITLY

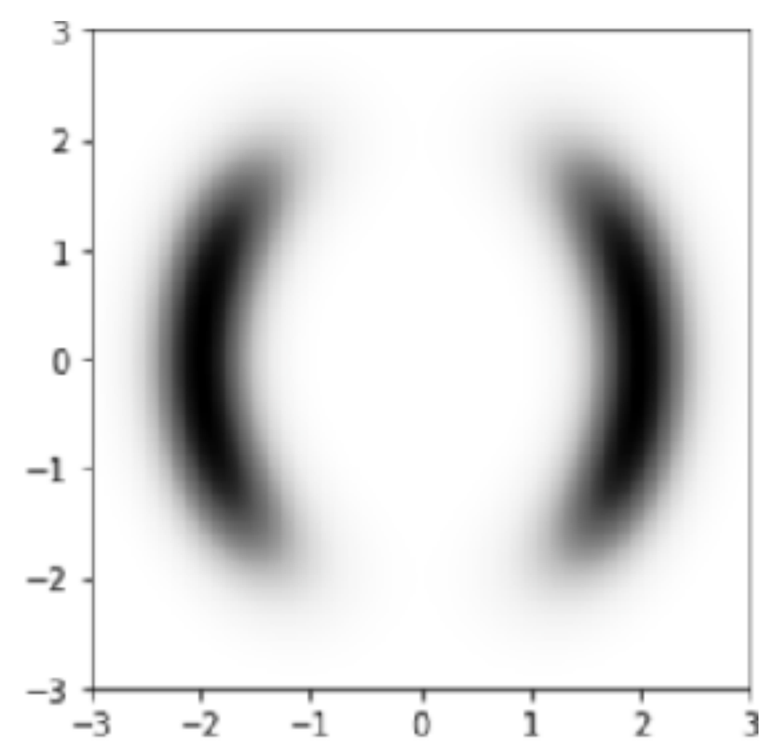
Can we learn our high-dimensional prior explicitly from data?
i.e. can we learn a generative model that will produce samples from that distribution?

How can we do this from samples (e.g. data)? Modeling the density?



Training data

$\{\mathbf{x}_1, \ldots, \mathbf{x}_n,\} \overset{iid}{\sim} \pi_{data}(\mathbf{x})$



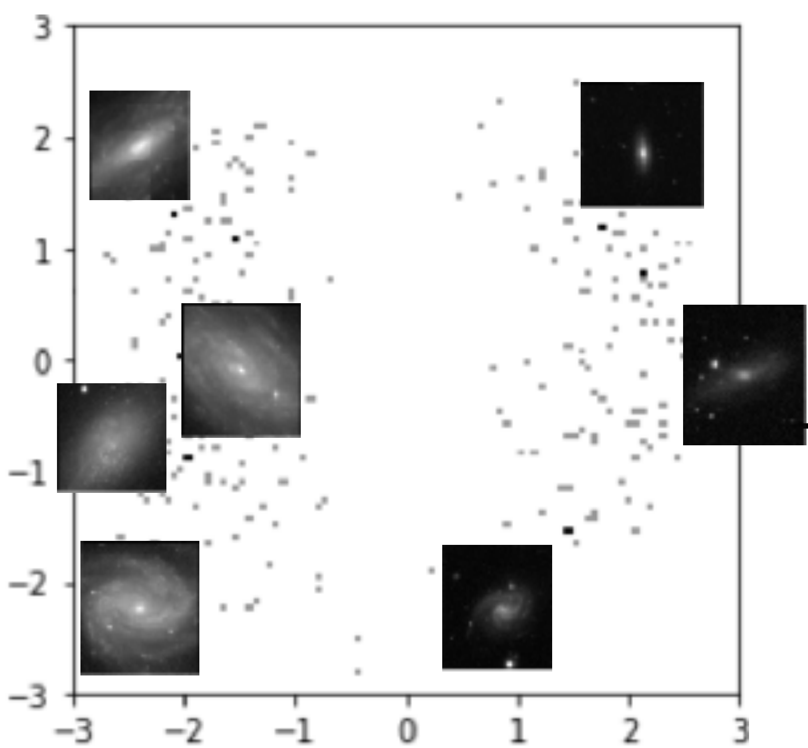
Density function

$p_\theta(\mathbf{x}) \approx \pi_{data}(\mathbf{x})$

SCORE MODELING

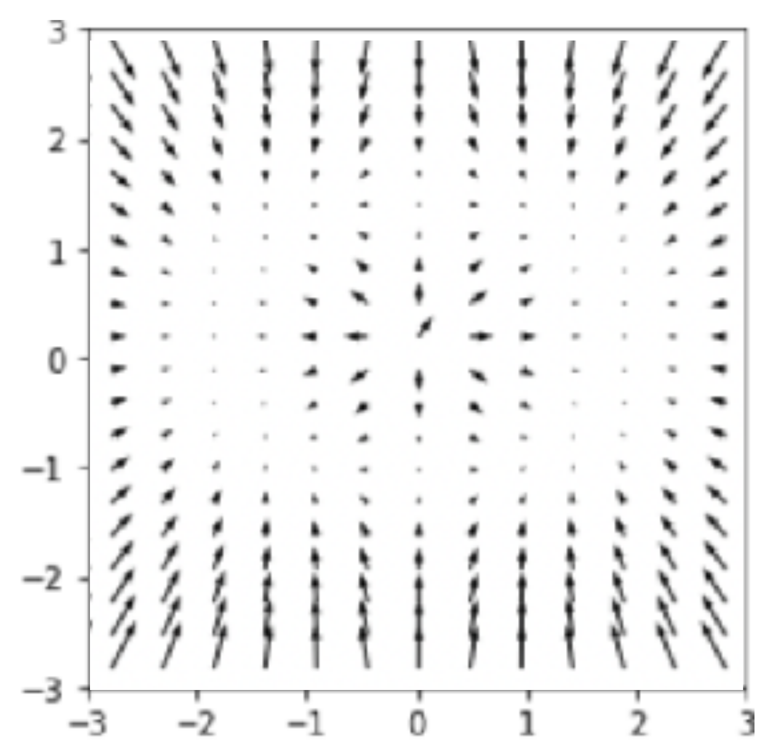
Turns out that if I want to sample a distribution, the only thing I need to learn is its ***score***, which does not include the normalization constant and only uses local information

$$\mathbf{s}(\mathbf{x}) = \nabla_{\mathbf{x}} \log(\pi(\mathbf{x}))$$



Training data

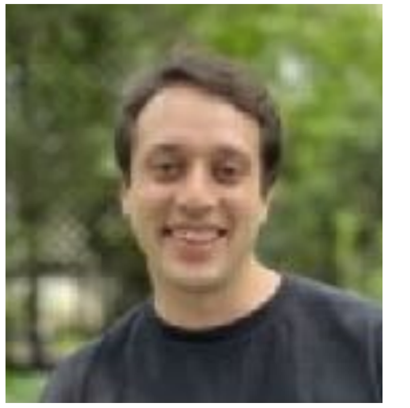
$$\{\mathbf{x}_1, \ldots, \mathbf{x}_n,\} \overset{iid}{\sim} \pi_{data}(\mathbf{x})$$



Score function

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla \log(\pi(\mathbf{x}))$$

SCORE-BASED MODELING



Alexandre Adam

We model the score of the prior

$$s_\theta(x) \equiv \nabla_x \log p_\theta(x)$$

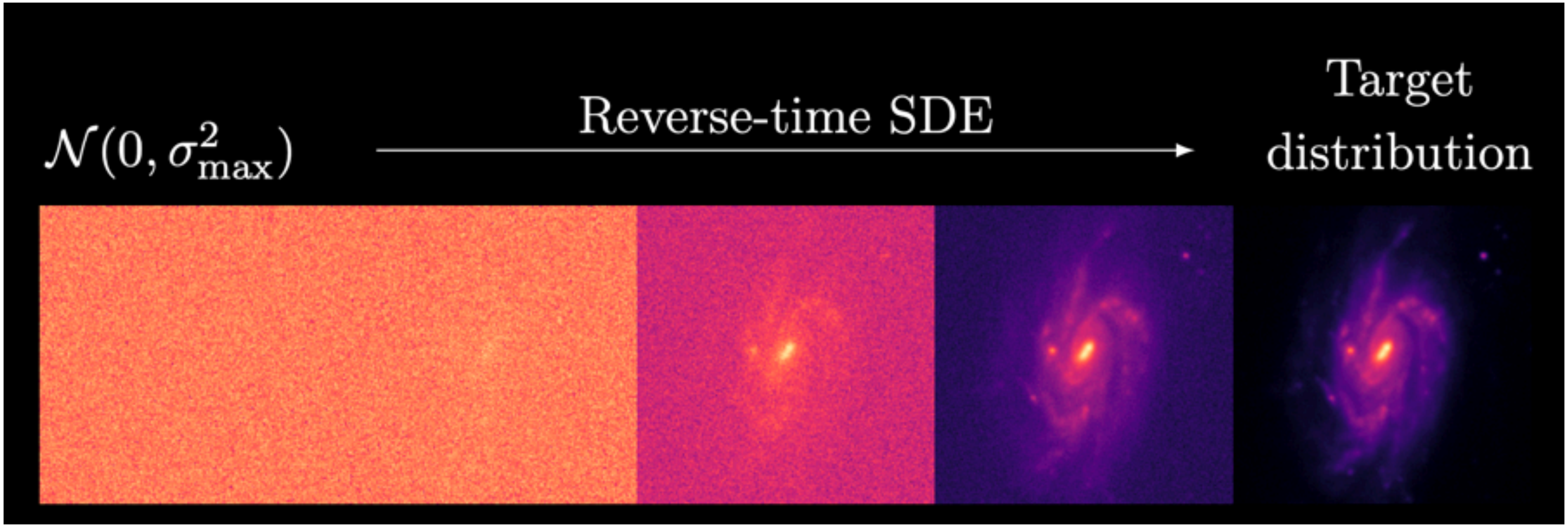
Adam et al. NeurIPS 2022 ML4PS workshop, arXiv:2211.03812

Score-Based Modeling

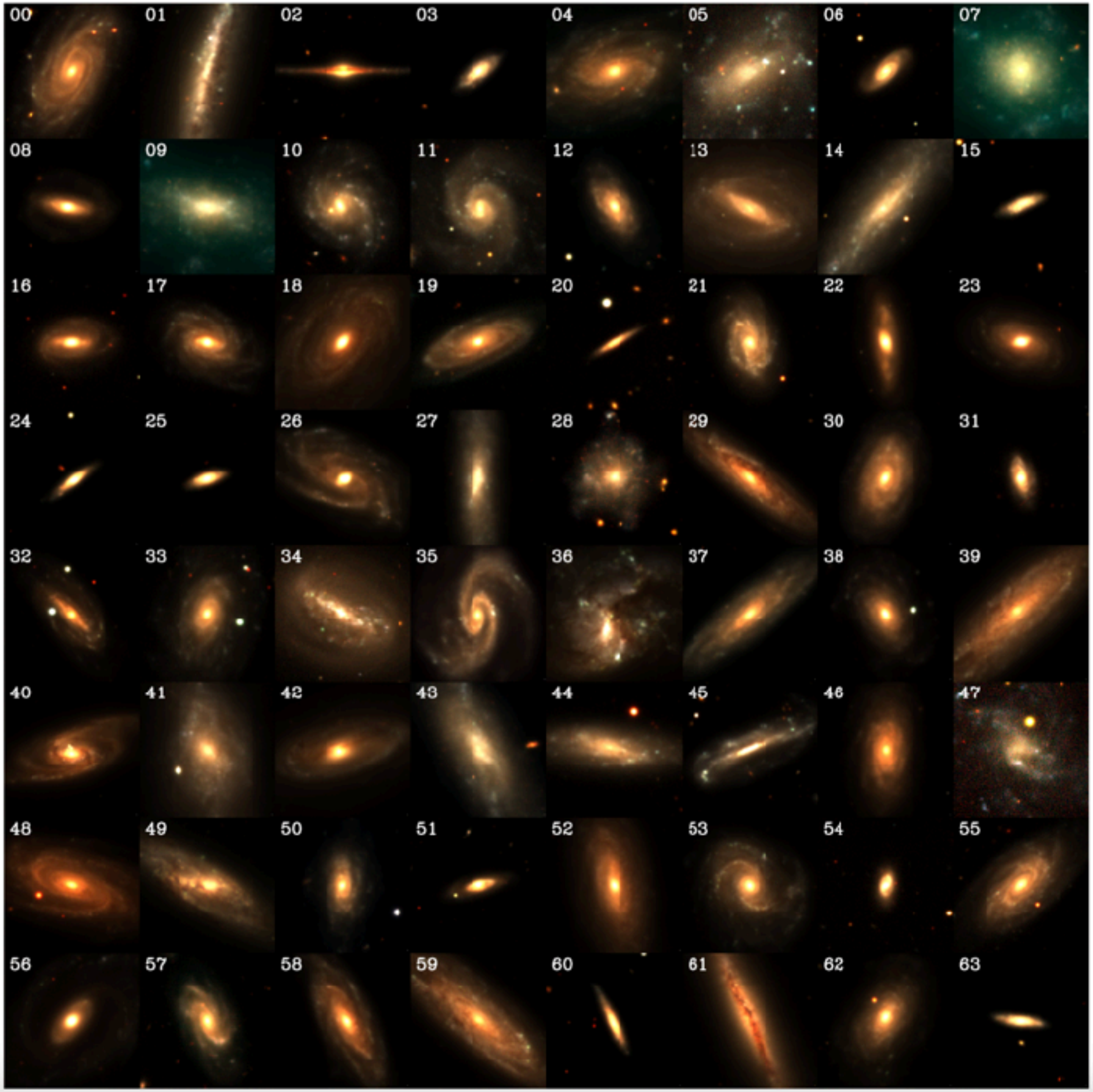

Alexandre Adam

We model the score of the prior

$$s_\theta(x) \equiv \nabla_x \log p_\theta(x)$$



Adam et al. NeurIPS 2022 ML4PS workshop, arXiv:2211.03812

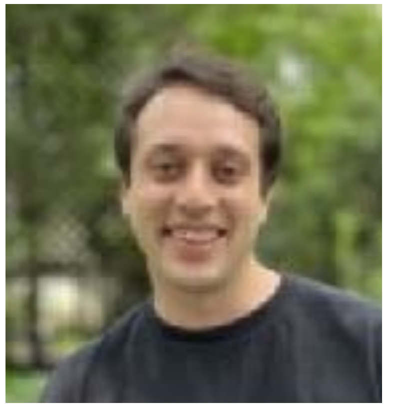


Connor Stone

http://www.mjjsmith.com/thisisnotagalaxy/

Smith et al. arXiv:2111.01713

SCORE-BASED MODELING

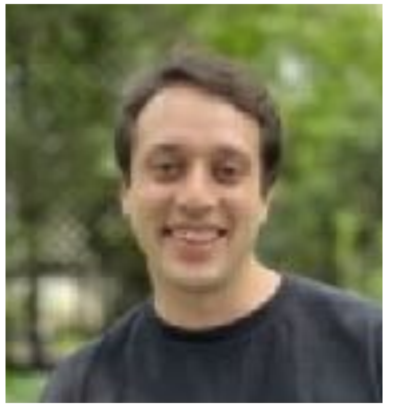


Alexandre Adam

Now if we want to sample from the posterior, its score is all we need:

$$\nabla_x \log p(x \mid y)$$

Score-based Modeling



Alexandre Adam

Now if we want to sample from the posterior, its score is all we need:

$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p_\theta(x)$$

Score-based Modeling

Alexandre Adam

Now if we want to sample from the posterior, its score is all we need:

$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p_\theta(x)$$

This is the prior score we learnt from the training data

Score-based Modeling


Alexandre Adam

Now if we want to sample from the posterior, its score is all we need:

$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p_\theta(x)$$

To a good approximation, we can calculate the likelihood score analytically if we assume it's Gaussian and we know the lensing matrix.

This is the prior score we learnt from the training data

Score-based Modeling


Alexandre Adam

Now if we want to sample from the posterior, its score is all we need:

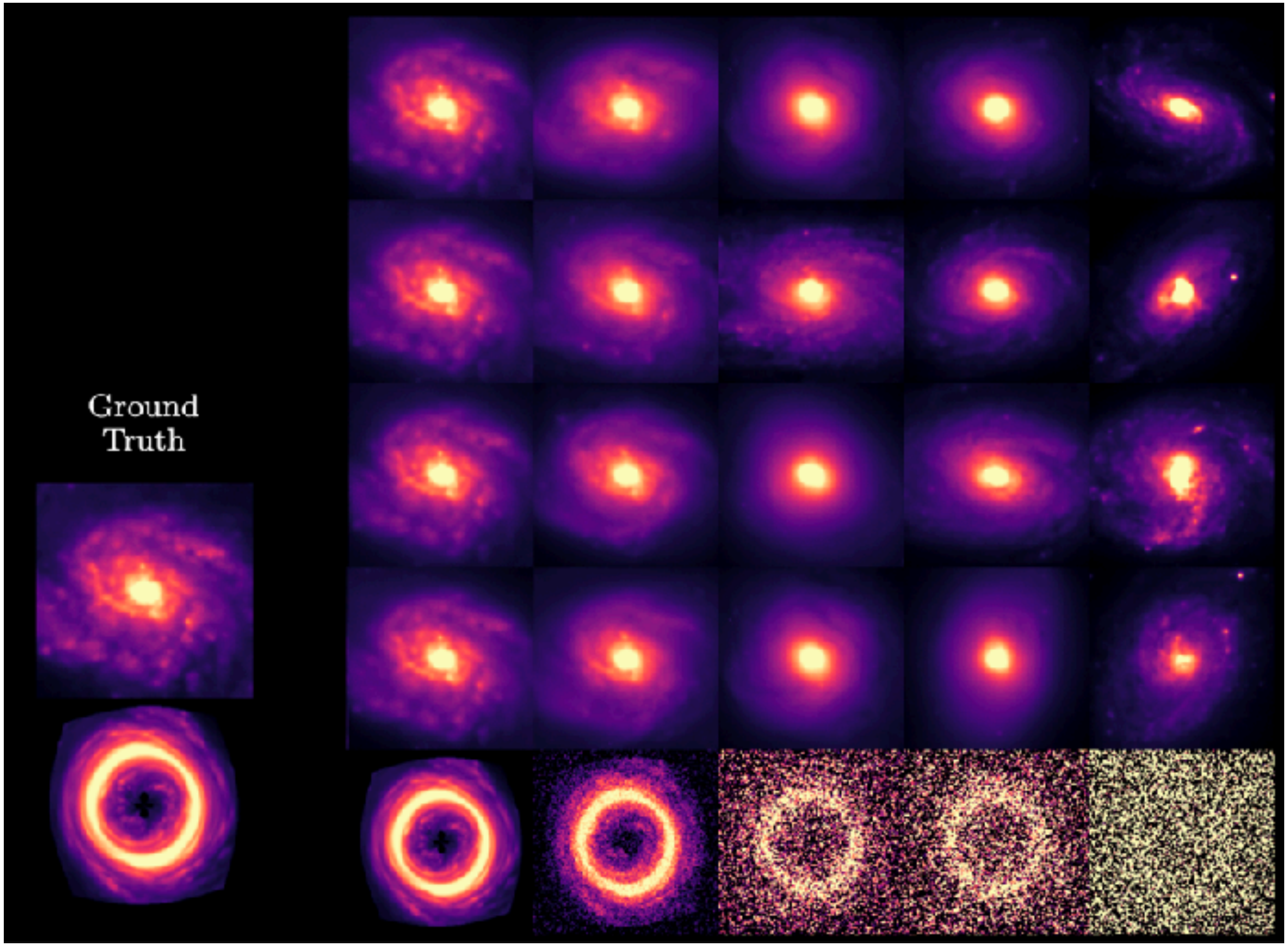
$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p_\theta(x)$$

Score-based Modeling

Alexandre Adam

Now if we want to sample from the posterior, its score is all we need:

$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p_\theta(x)$$

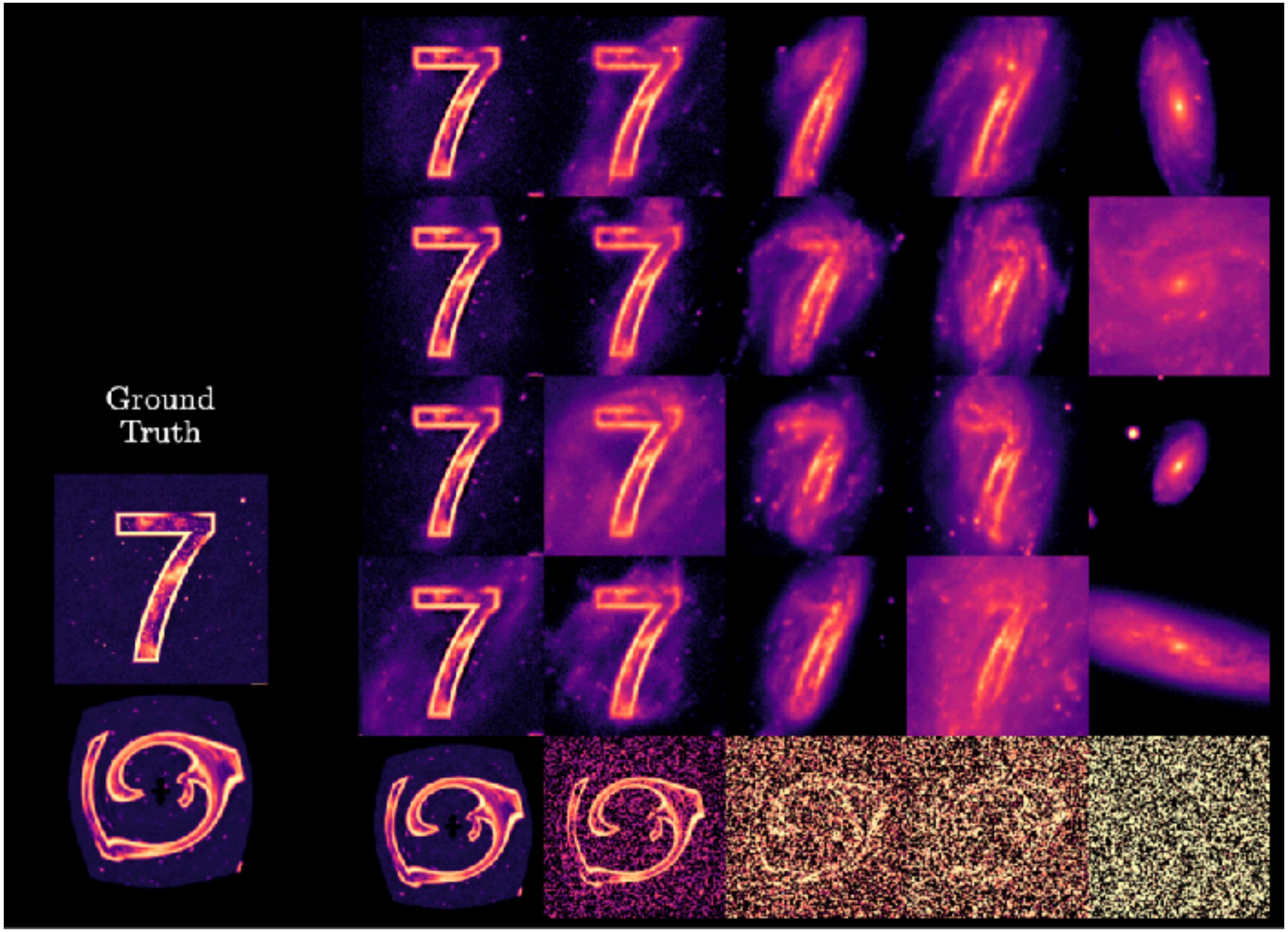


Out of Distribution Tests

Alexandre Adam

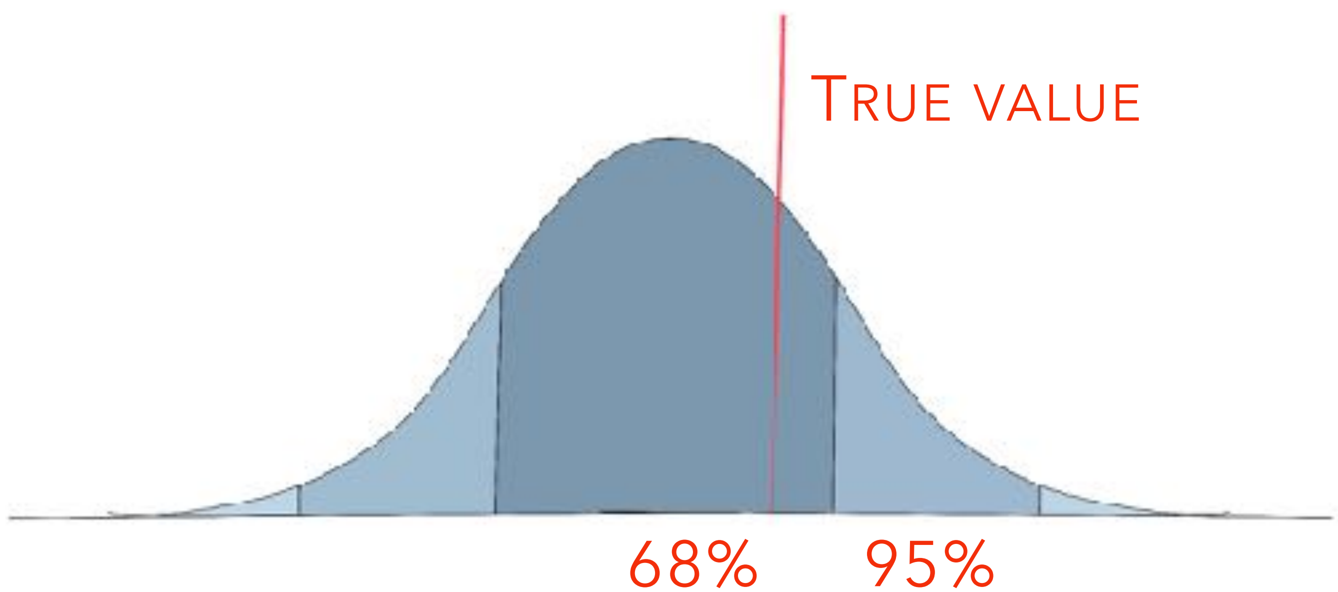
Now if we want to sample from the posterior, its score is all we need:

$$\nabla_x \log p(x \mid y) = \nabla_x \log p(y \mid x) + \nabla_x \log p_\theta(x)$$



ARE THESE UNCERTAINTIES ACCURATE?

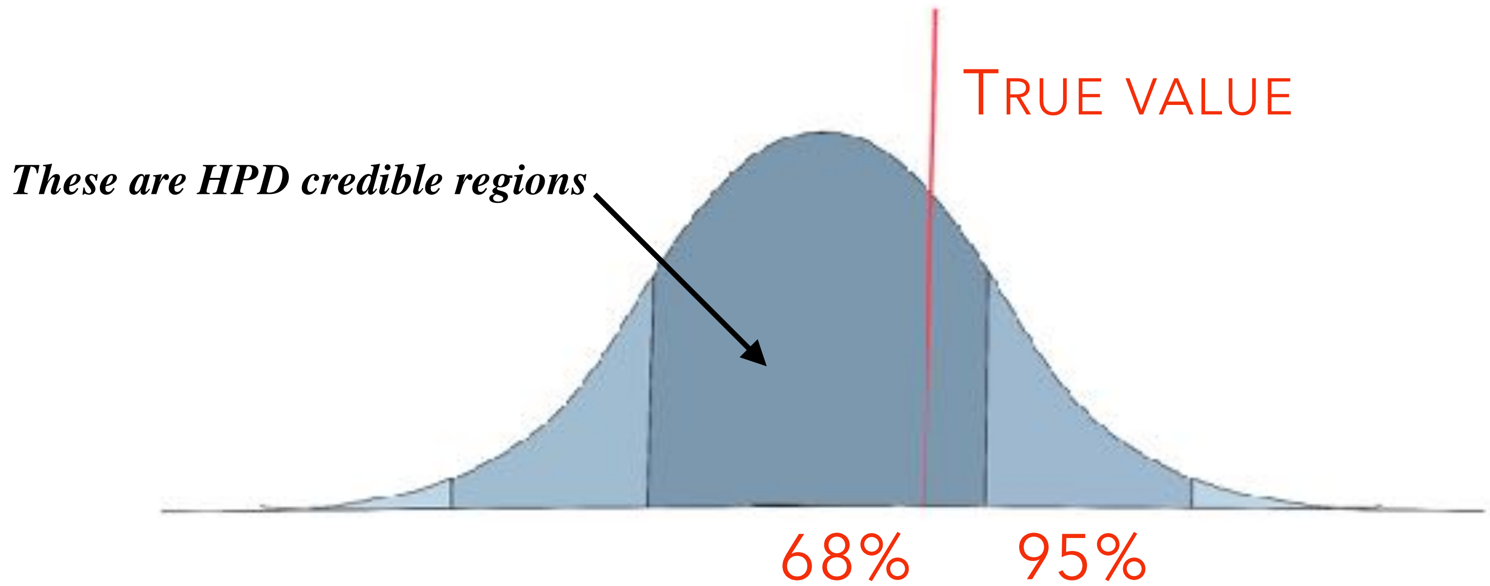
The expected coverage probability of a credible region is the proportion of the time that the region contains the true value of interest.



For an accurate posterior estimator, the expected coverage probability is equal to the probability mass of the credible region.

ARE THESE UNCERTAINTIES ACCURATE?

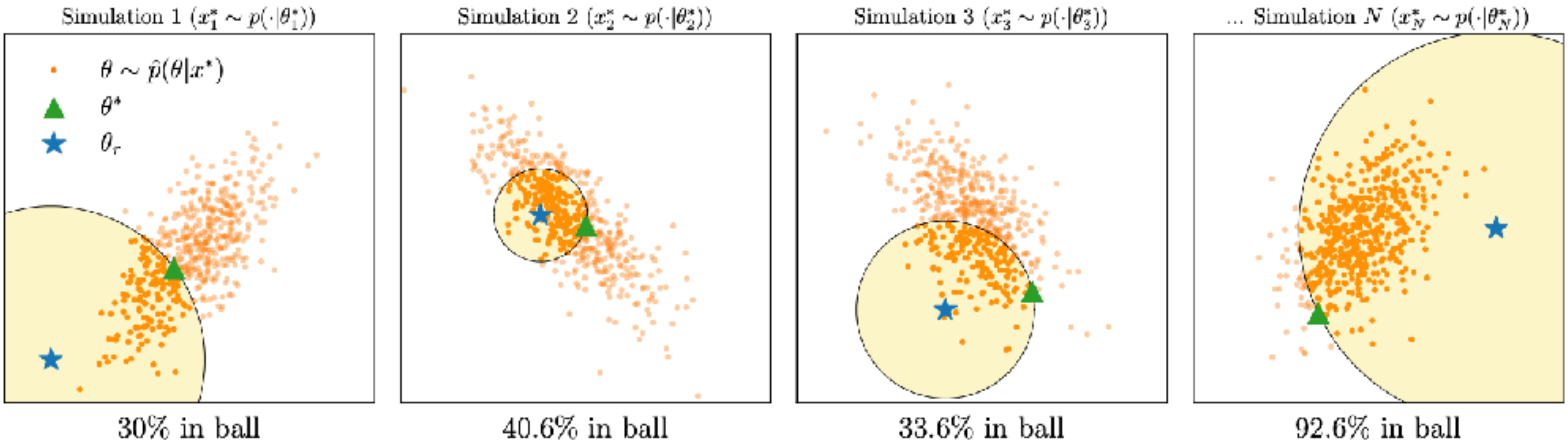
The expected coverage probability of a credible region is the proportion of the time that the region contains the true value of interest.



For an accurate posterior estimator, the expected coverage probability is equal to the probability mass of the credible region.

Coverage Test for Accuracy

Pablo Lemos

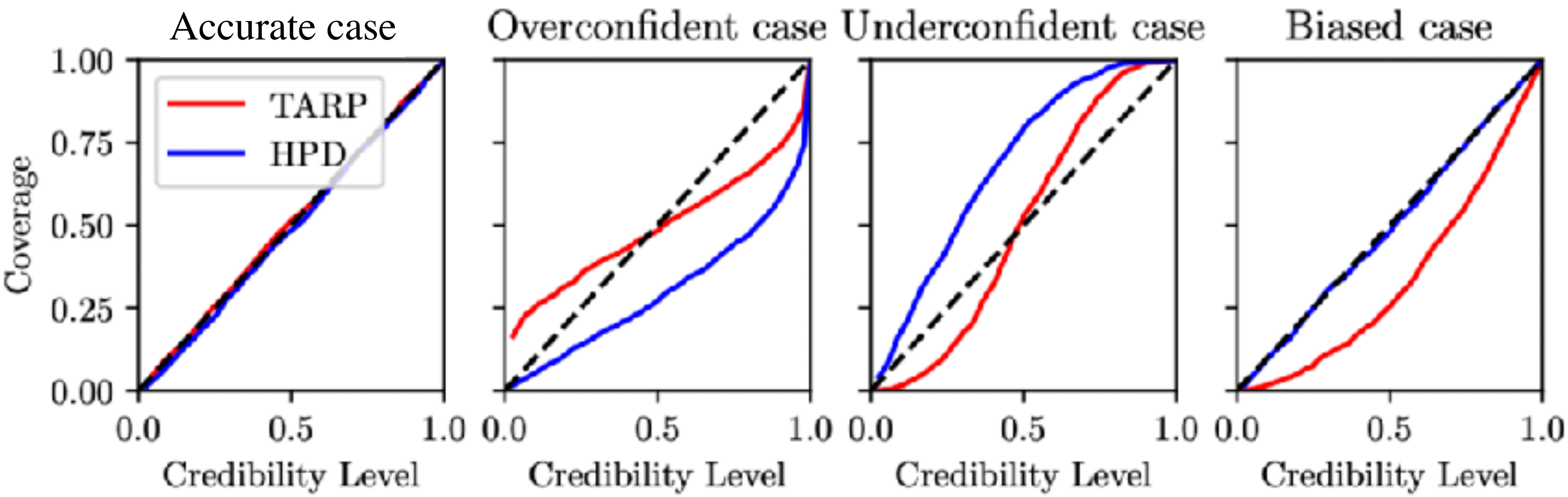


pip install tarp

Lemos, et al. ICML 2023, 2302.03026

COVERAGE TEST FOR ACCURACY

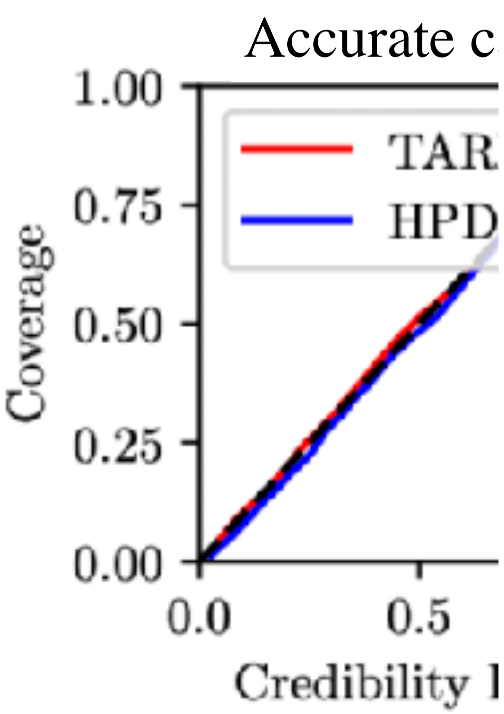
Pablo Lemos

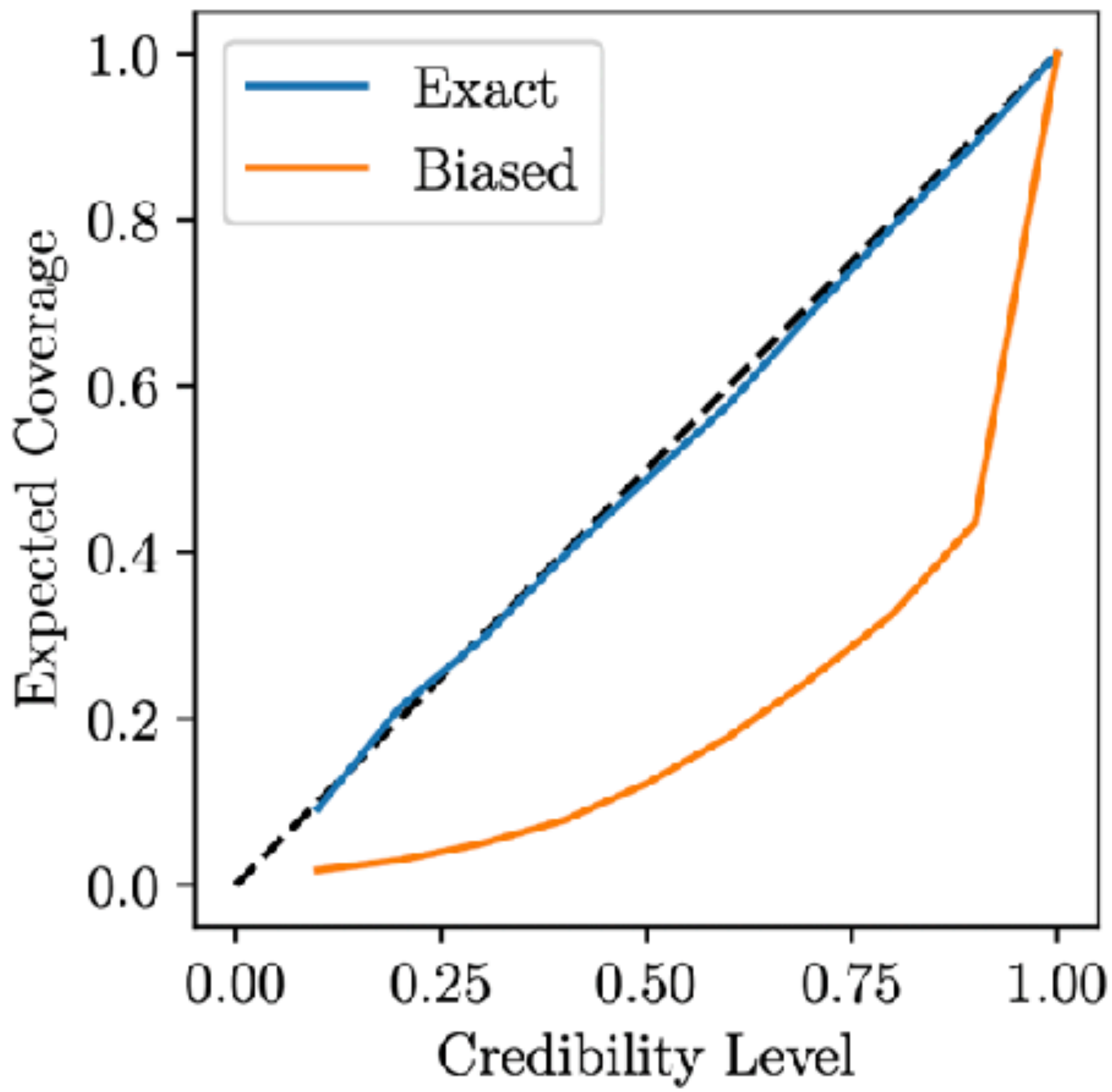


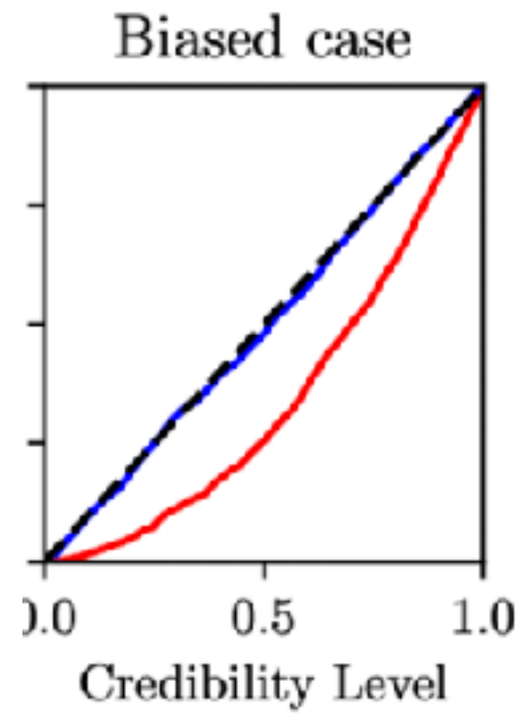
Lemos, et al. ICML 2023, 2302.03026

COVERAGE TEST FOR ACCURACY

Pablo Lemos





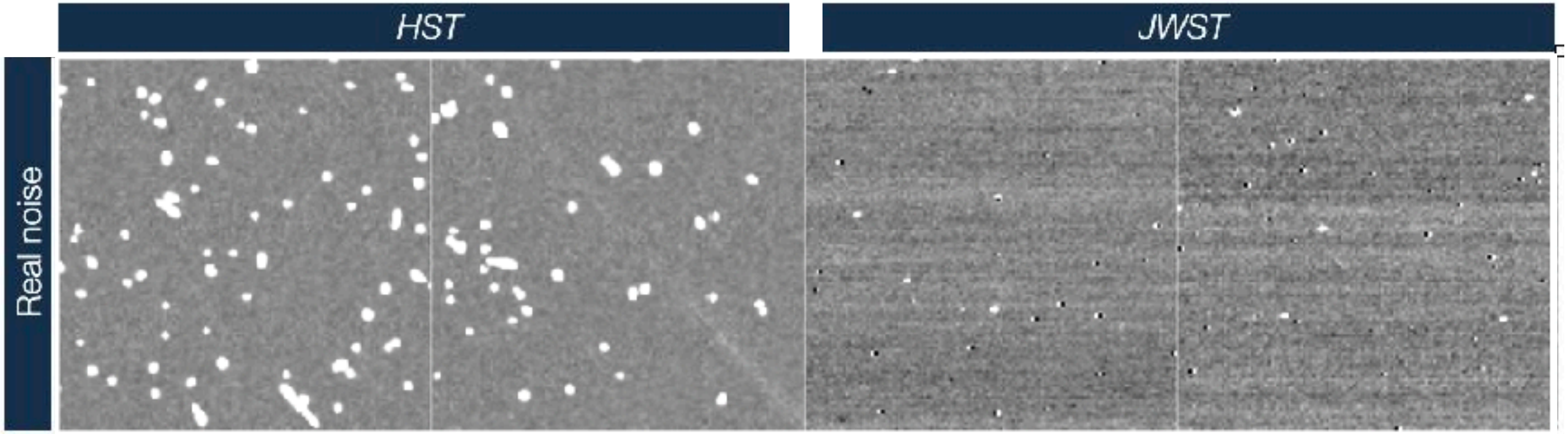


Lemos, et al. ICML 2023, 2302.03026

DEALING WITH REALISTIC NOISE: BEYOND GAUSSIANITY

Alexandre Adam

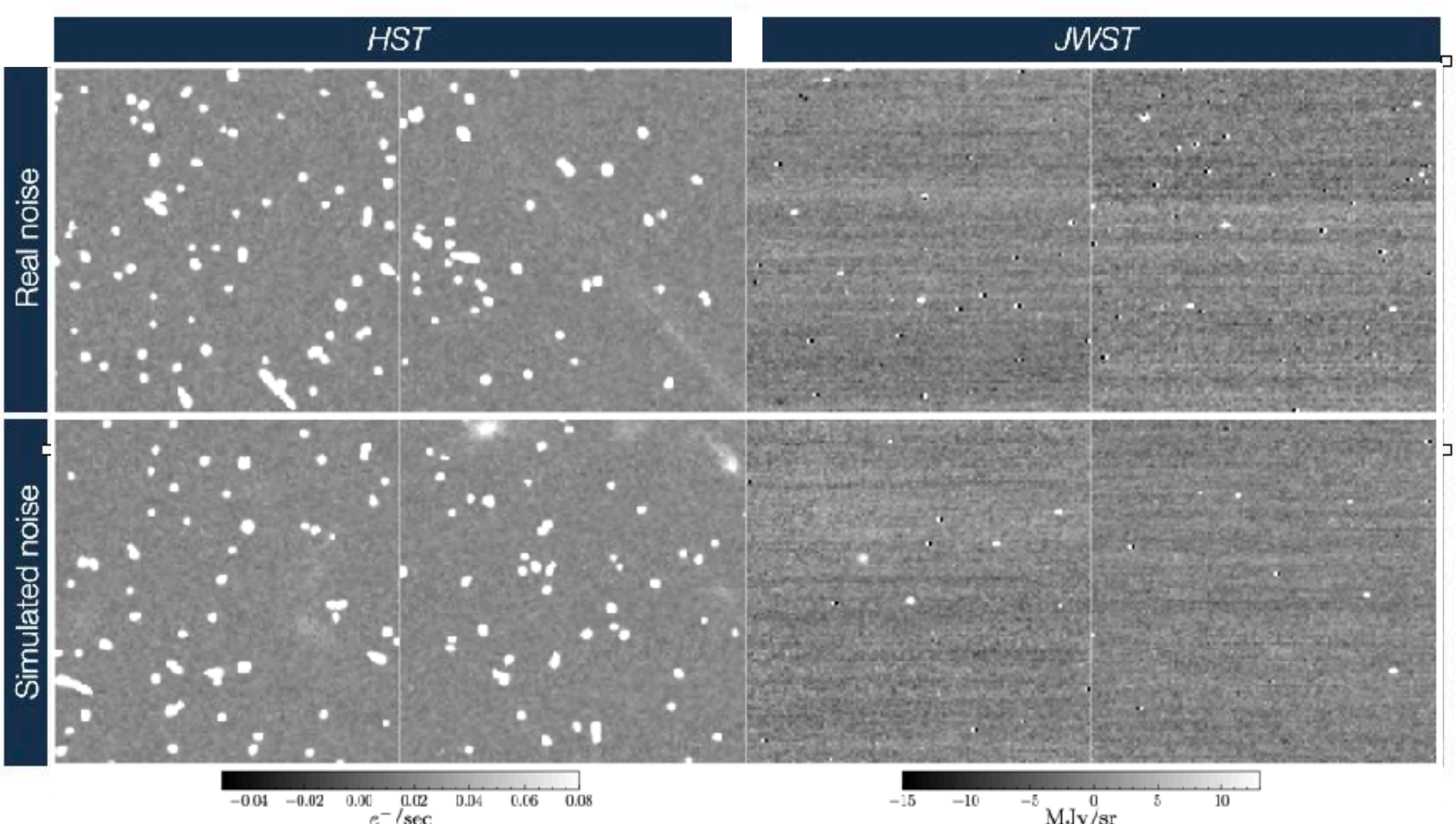
Ronan Legin



Legin, Adam, et al. ApJ Letters 2023, 2302.03046

DEALING WITH REALISTIC NOISE: BEYOND GAUSSIANITY
SLIC: SCORE-BASED LIKELIHOOD CHARACTERIZATION

Since we have learnt a generative model of the additive noise, it can now be used in a simulation pipeline to get new, independent realizations of noise:

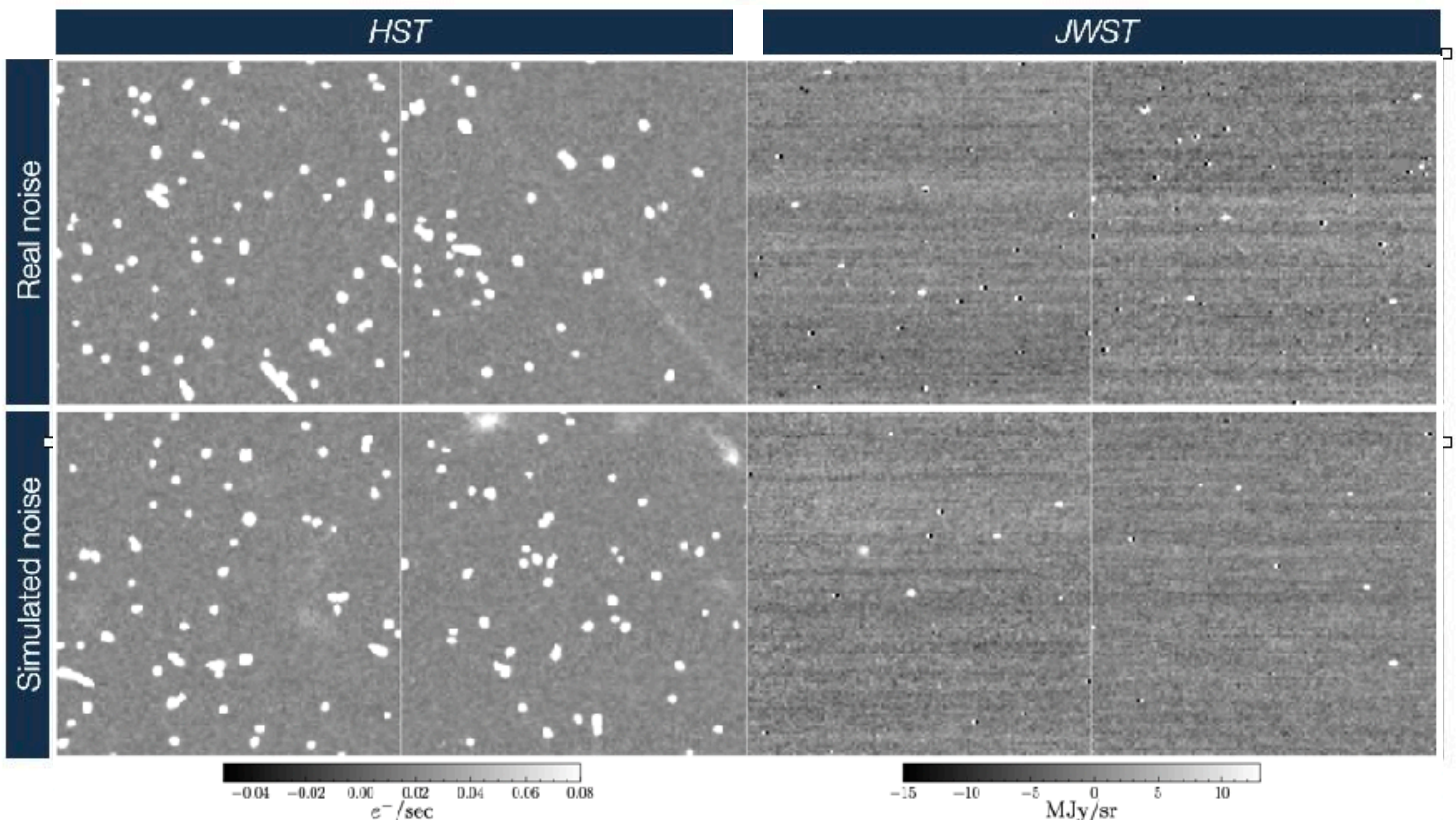


Legin, Adam, et al. ApJ Letters 2023, 2302.03046

DEALING WITH REALISTIC NOISE: BEYOND GAUSSIANITY
SLIC: SCORE-BASED LIKELIHOOD CHARACTERIZATION

Since we have learnt a generative model of the additive noise, it can now be used in a simulation pipeline to get new, independent realizations of noise:

$$P(\mathbf{x}_{\mathrm{O}}|\eta) = Q(\mathbf{x}_{\mathrm{O}} - \mathbf{M}(\eta))$$



Legin, Adam, et al. ApJ Letters 2023, 2302.03046

Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

Legin, Adam, et al. ApJ Letters 2023, 2302.03046

Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

$$\mathbf{s}(\mathbf{x}_0) = \partial \log Q(\mathbf{x}_0)/\partial \mathbf{x}$$

Legin, Adam, et al. ApJ Letters 2023, 2302.03046

Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

$$\mathbf{s}(\mathbf{x}_0) = \partial \log Q(\mathbf{x}_0)/\partial \mathbf{x}$$

$$P(\mathbf{x}_\mathrm{O}|\eta) = Q(\mathbf{x}_\mathrm{O} - \mathbf{M}(\eta))$$

Legin, Adam, et al. ApJ Letters 2023, 2302.03046

Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

$$\mathbf{s}(\mathbf{x}_0) = \partial \log Q(\mathbf{x}_0)/\partial \mathbf{x}$$

$$P(\mathbf{x}_\mathrm{O}|\eta) = Q(\mathbf{x}_\mathrm{O} - \mathbf{M}(\eta))$$

$$\eta_{\iota+1} = \eta_\iota + \tau \boldsymbol{\nabla}_\eta \log Q(\mathbf{x}_\mathrm{O} - \mathbf{M}(\eta)) + \sqrt{2\tau}\xi_\iota$$

Legin, Adam, et al. ApJ Letters 2023, 2302.03046

Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

$$\mathbf{s}(\mathbf{x}_0) = \partial \log Q(\mathbf{x}_0)/\partial \mathbf{x}$$

$$P(\mathbf{x}_\mathrm{O}|\eta) = Q(\mathbf{x}_\mathrm{O} - \mathbf{M}(\eta))$$

$$\eta_{i+1} = \eta_i + \tau\, \boldsymbol{\nabla}_\mathbf{x} \log Q(\mathbf{x}_\mathrm{o} - \mathbf{M}(\eta)) \boldsymbol{\nabla}_\eta M(\eta_i) + \sqrt{2\tau}\xi$$

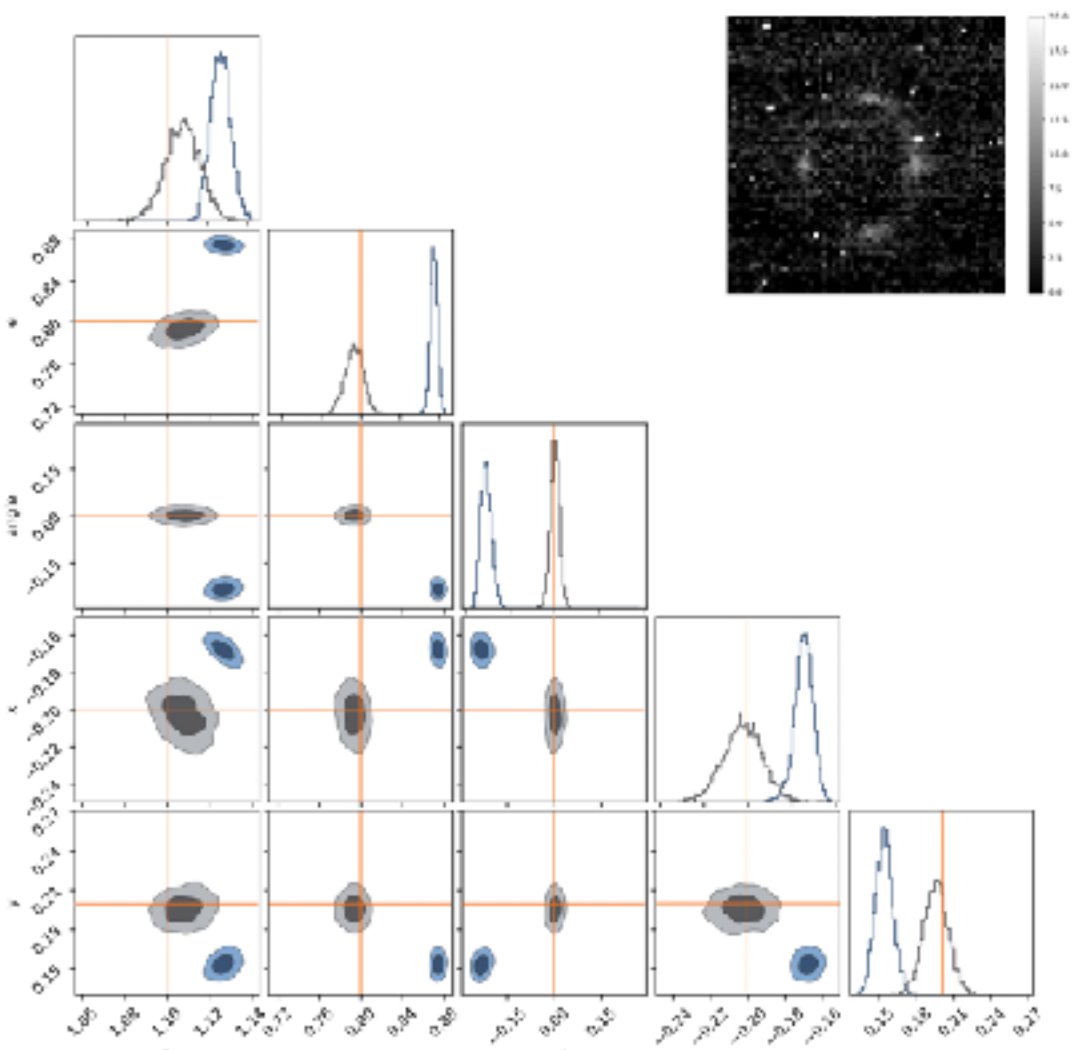
Legin, Adam, et al. ApJ Letters 2023, 2302.03046

Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

$$\mathbf{s}(\mathbf{x}_0) = \partial \log Q(\mathbf{x}_0)/\partial \mathbf{x}$$

$$P(\mathbf{x}_O|\eta) = Q(\mathbf{x}_O - \mathbf{M}(\eta))$$

$$\eta_{i+1} = \eta_i + \tau\, \boldsymbol{\nabla}_{\mathbf{x}} \log Q(\mathbf{x}_o - \mathbf{M}(\eta)) \boldsymbol{\nabla}_\eta M(\eta_i) + \sqrt{2\tau}\xi$$



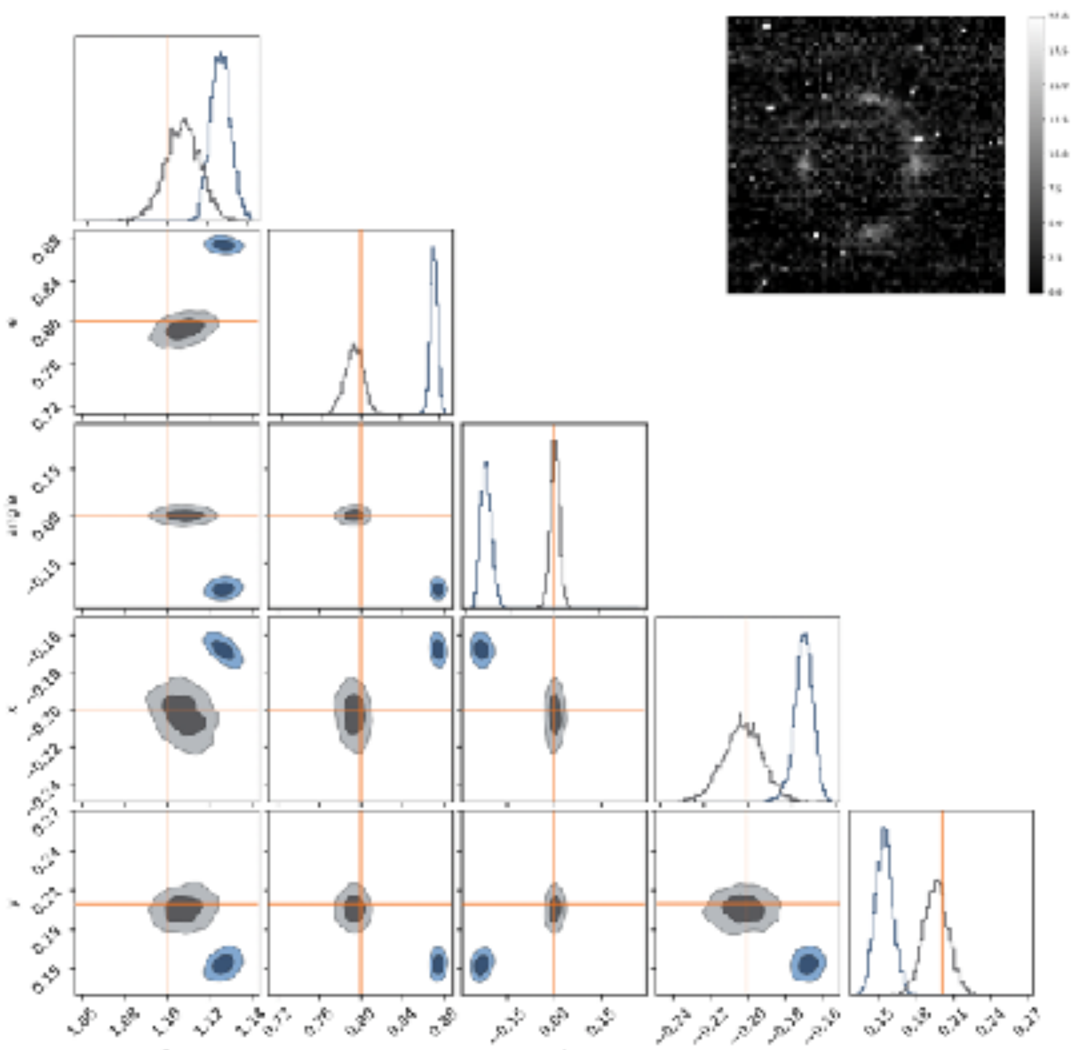
Legin, Adam, et al. ApJ Letters 2023, 2302.03046

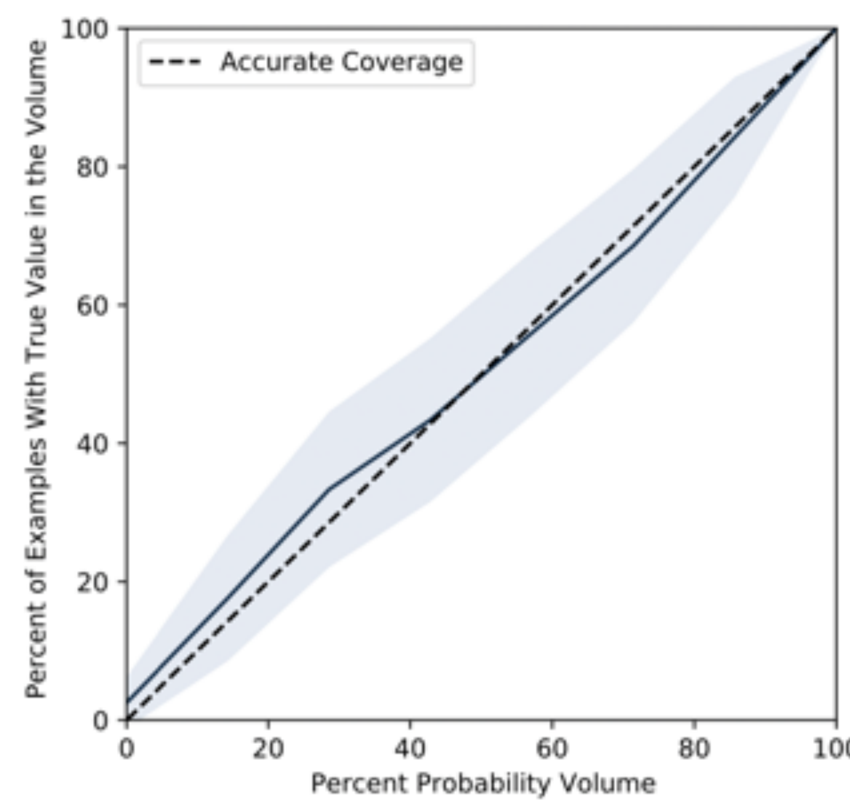
Dealing With Realistic Noise: Beyond Gaussianity
SLIC: Score-based LikelIhood Characterization

$$\mathbf{s}(\mathbf{x}_0) = \partial \log Q(\mathbf{x}_0)/\partial \mathbf{x}$$

$$P(\mathbf{x}_O|\eta) = Q(\mathbf{x}_O - \mathbf{M}(\eta))$$

$$\eta_{i+1} = \eta_i + \tau \boldsymbol{\nabla}_{\mathbf{x}} \log Q(\mathbf{x}_o - \mathbf{M}(\eta)) \boldsymbol{\nabla}_{\eta} M(\eta_i) + \sqrt{2\tau}\xi$$

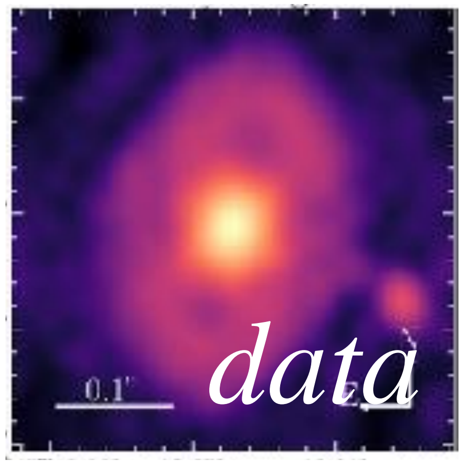




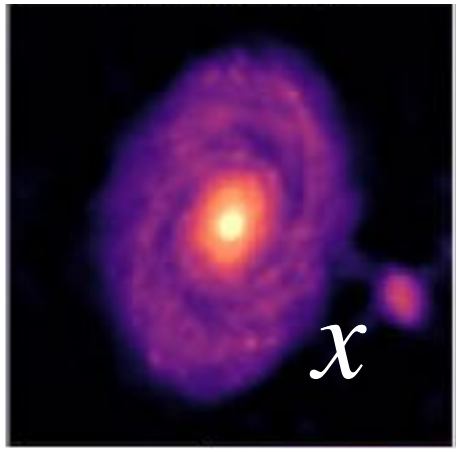
Legin, Adam, et al. ApJ Letters 2023, 2302.03046

PSF-DECONVOLUTION

Alexandre Adam



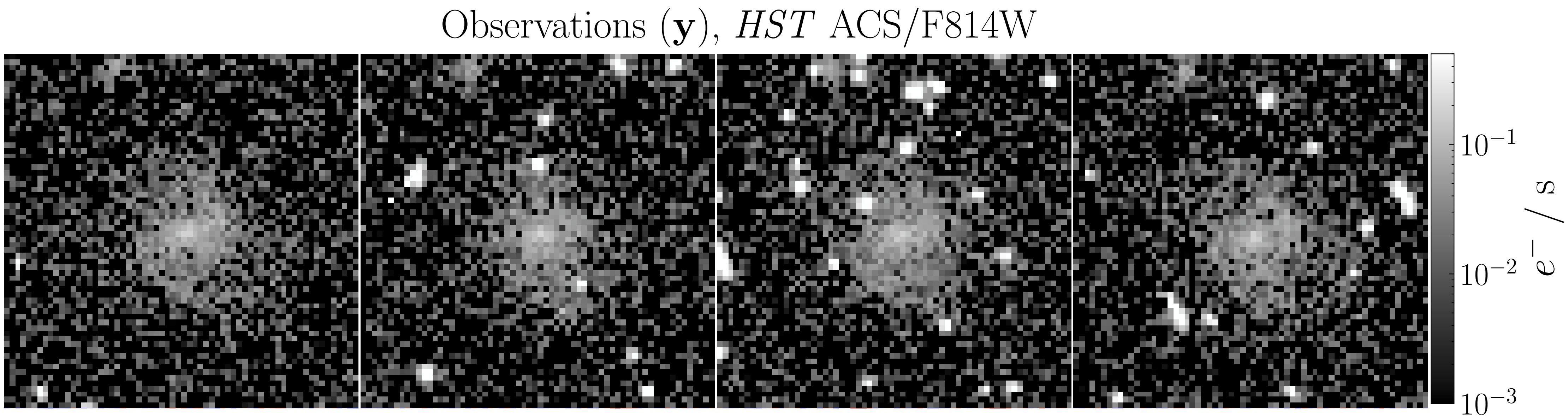
$= PSF \Big($



$\Big) + n$

PSF-DECONVOLUTION (FOR HST)

Alexandre Adam

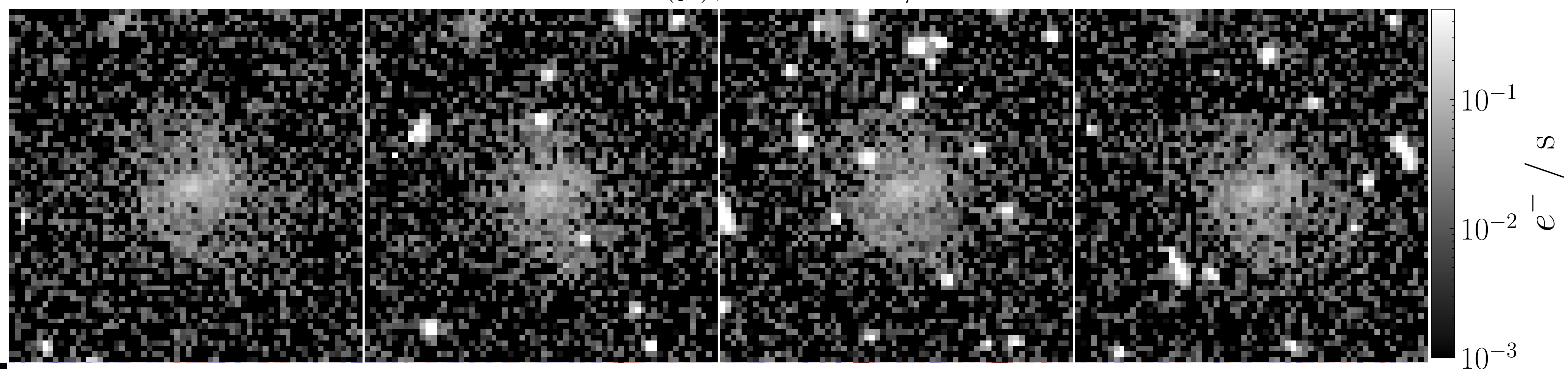


Adam et al. NeurIPS 2023 ML4PS workshop

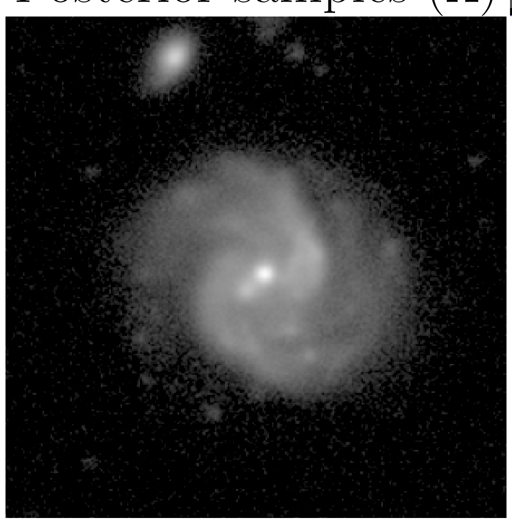
PSF-DECONVOLUTION (FOR HST)

Alexandre Adam

Observations ($\mathbf{y}$), *HST* ACS/F814W



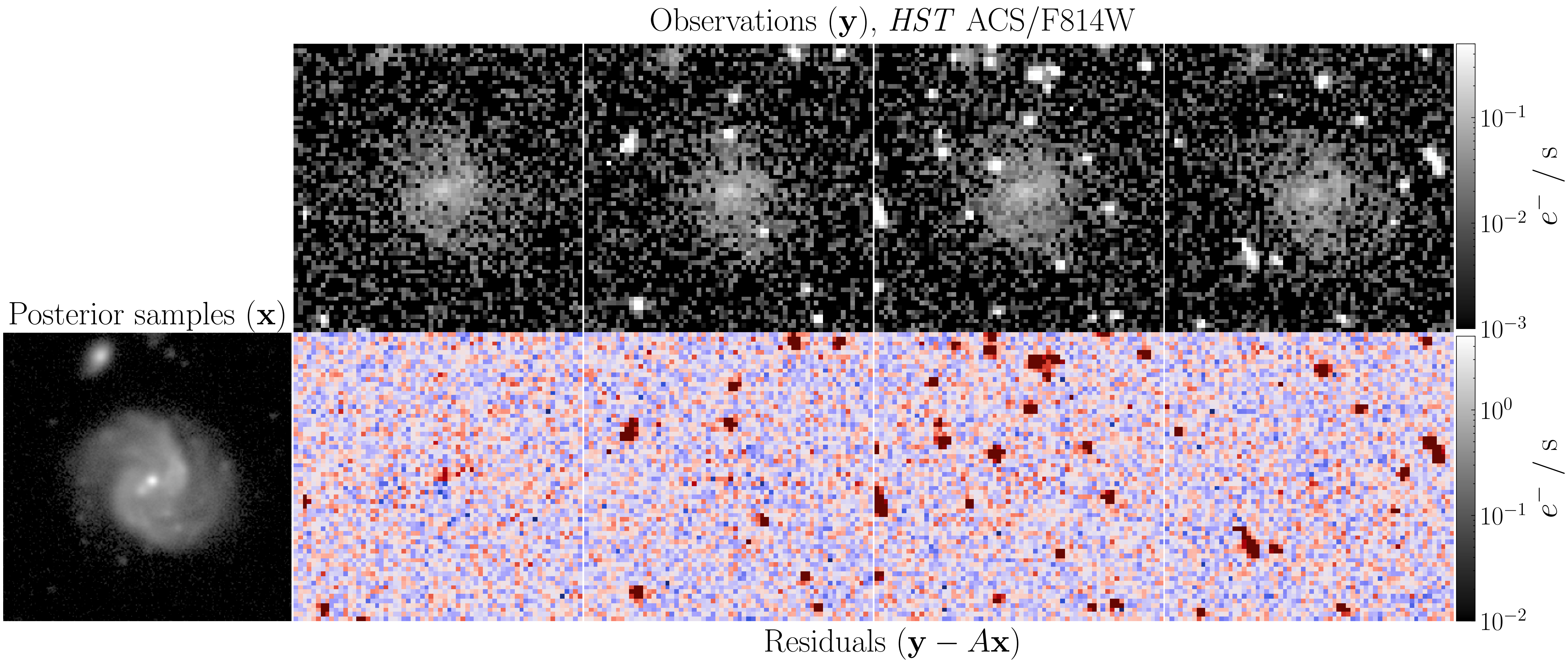
Posterior samples ($\mathbf{x}$)



Adam et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION (FOR HST)

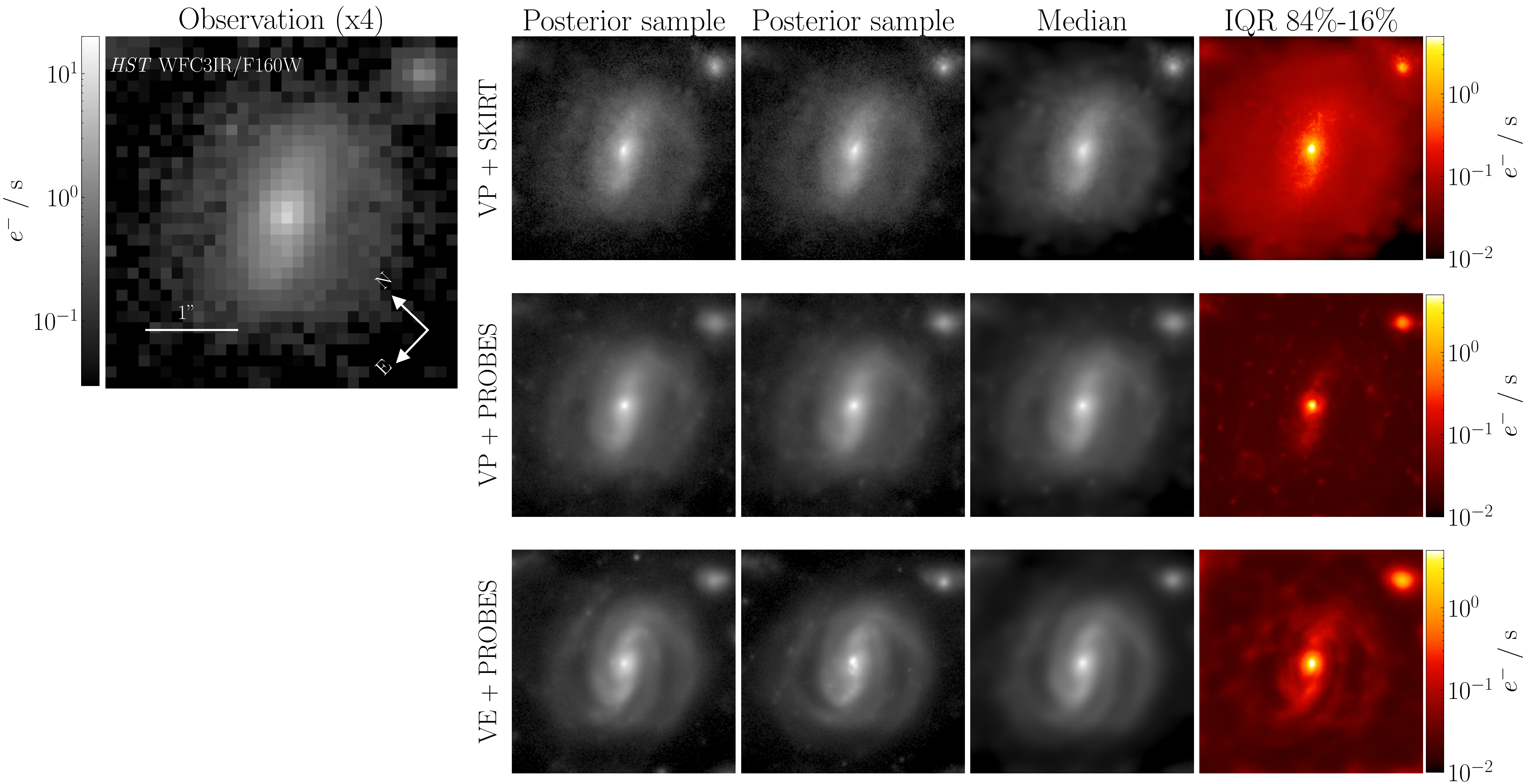
Alexandre Adam



Adam et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION (FOR HST)

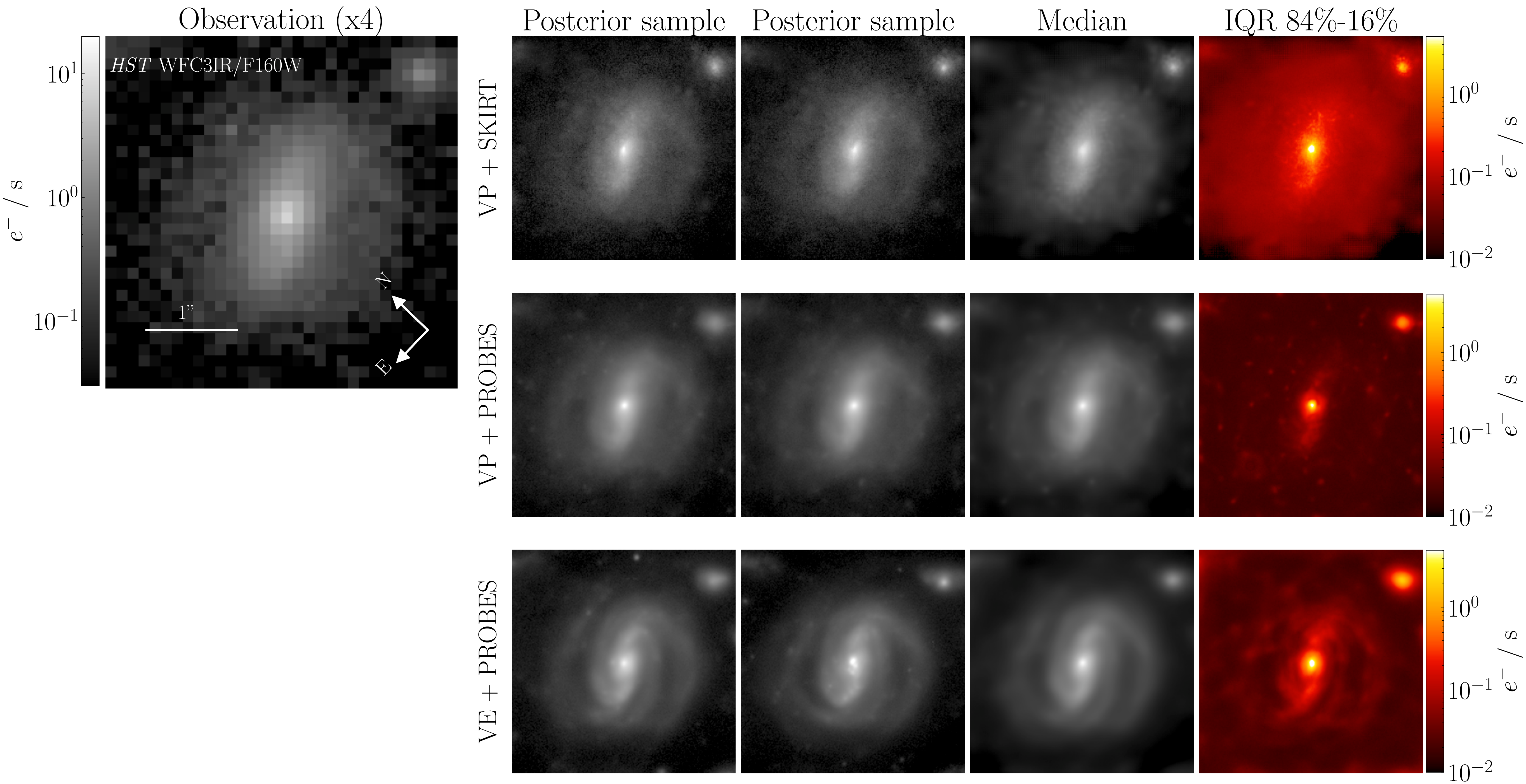
Alexandre Adam



Adam et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION (FOR HST)

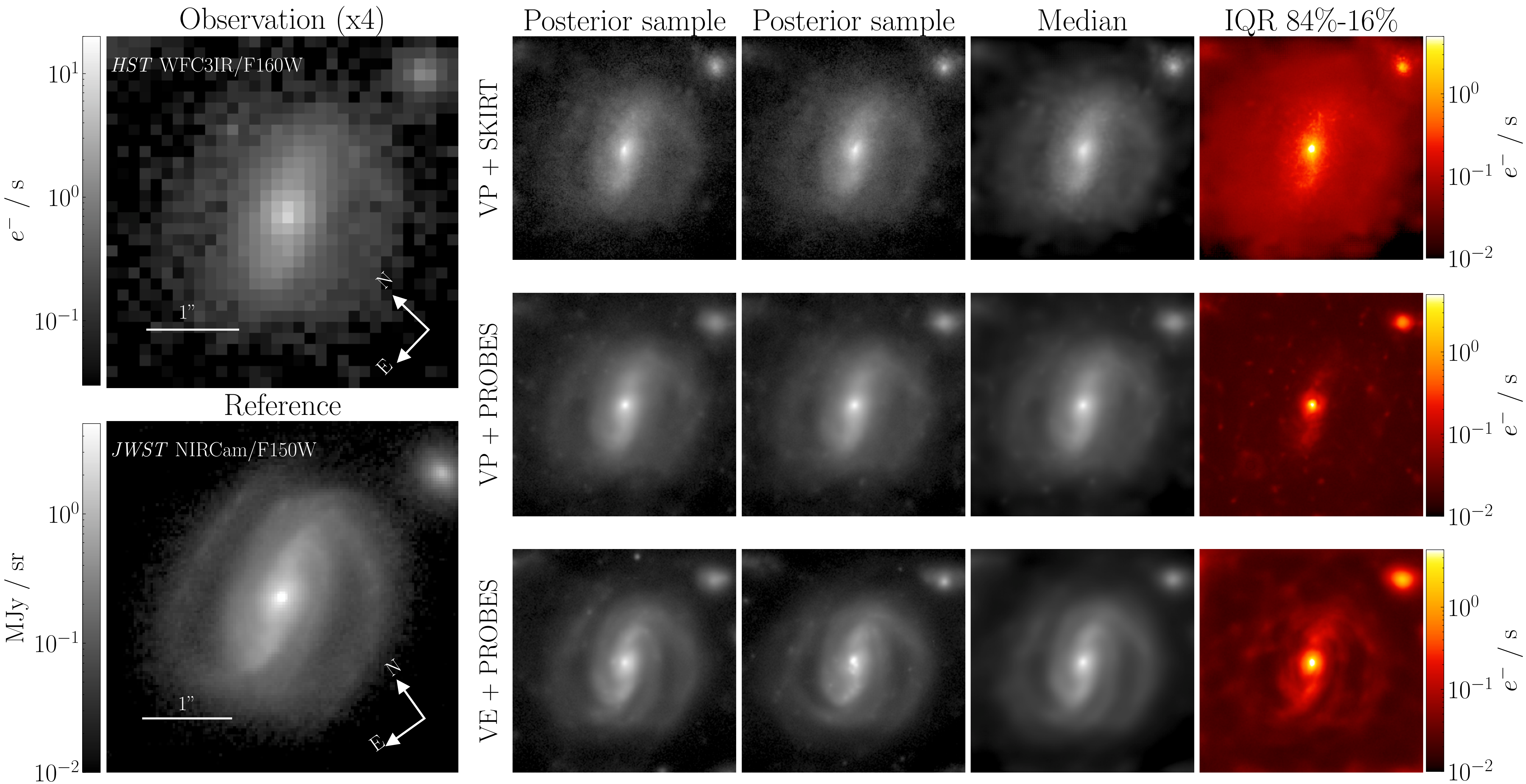
Alexandre Adam



Adam et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION (FOR HST)

Alexandre Adam



Adam et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Sky emission

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Observed visibilities → $\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$

Sky emission (→ $\mathbf{x}$)

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Observed visibilities → $\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$

Dense, unitary, Fourier operator

Sky emission

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Mask, subsampling function

Observed visibilities

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Dense, unitary, Fourier operator

Sky emission

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Mask, subsampling function

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Observed visibilities

Gaussian noise

Dense, unitary, Fourier operator

Sky emission

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Mask, subsampling function

Primary beam

Observed visibilities

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Gaussian noise

Dense, unitary, Fourier operator

Sky emission

Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

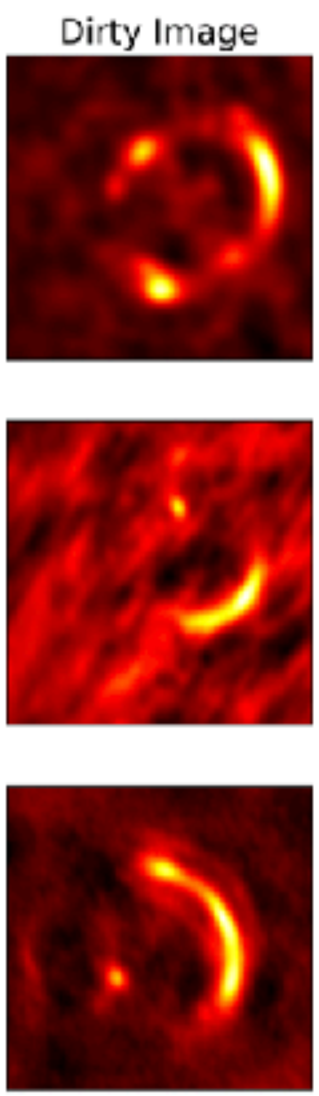
Mask, subsampling function

Primary beam

Observed visibilities → $\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$ ← Gaussian noise

Dense, unitary, Fourier operator

Sky emission



Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Mask, subsampling function

Primary beam

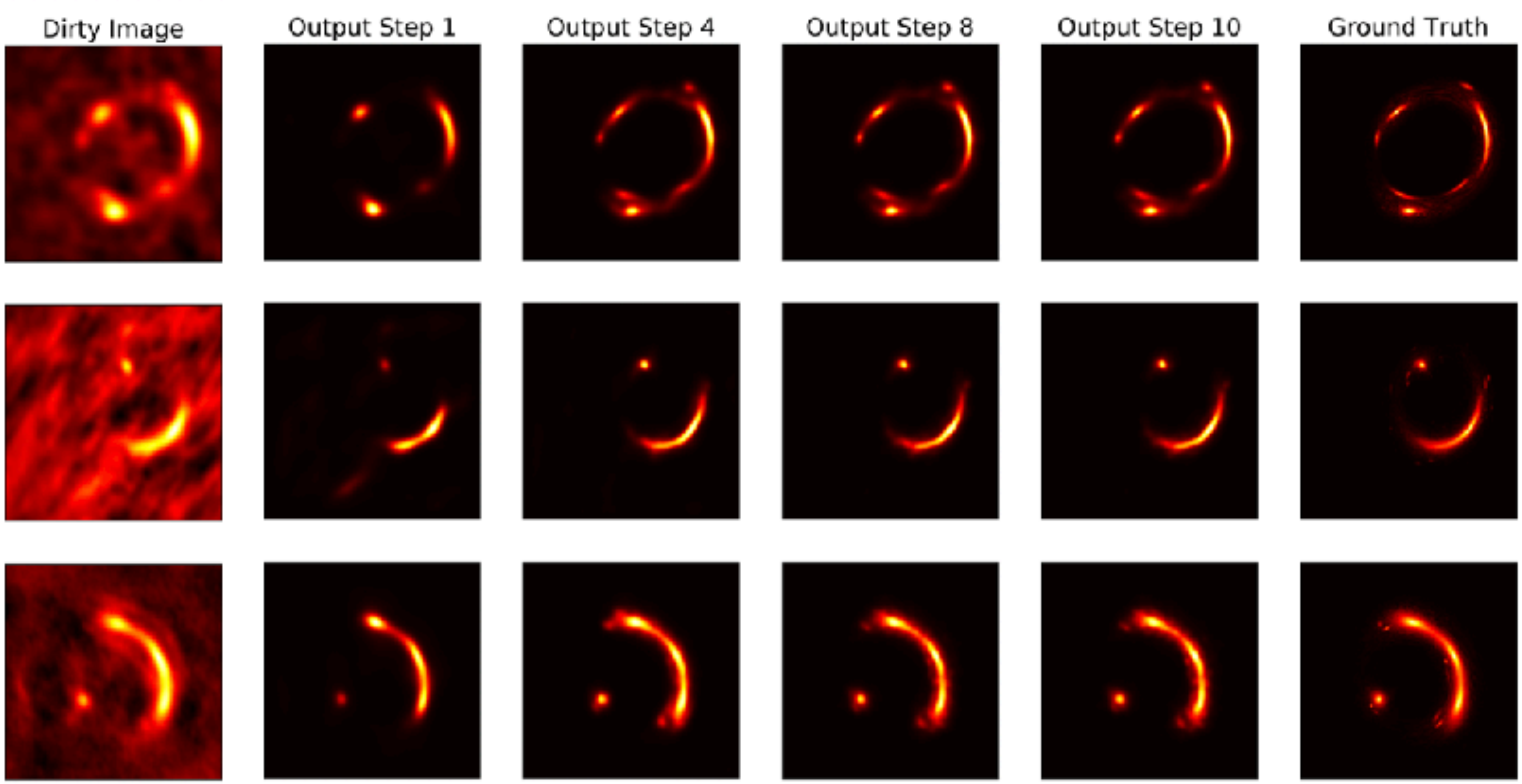
Observed visibilities

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Gaussian noise

Dense, unitary, Fourier operator

Sky emission



Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Mask, subsampling function

Primary beam

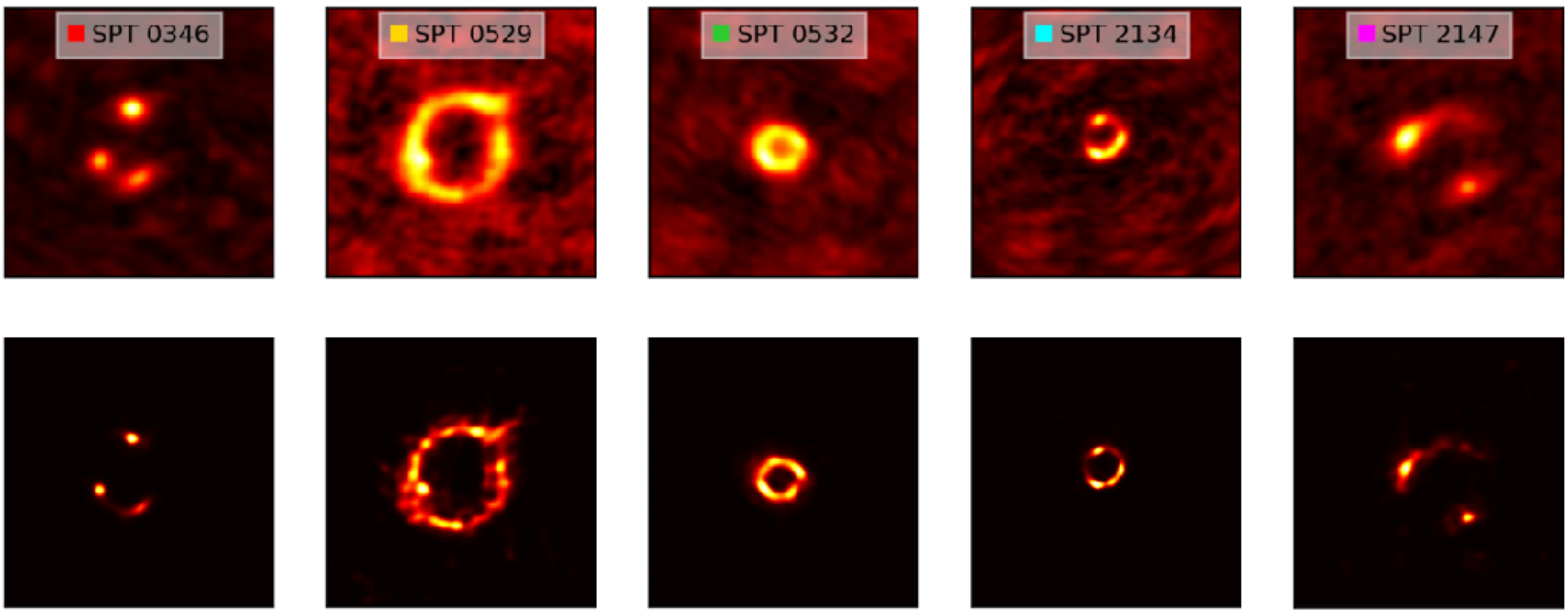
Observed visibilities

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Gaussian noise

Dense, unitary, Fourier operator

Sky emission



Morningstar et al. 2019

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Mask, subsampling function

Primary beam

Observed visibilities

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Gaussian noise

Dense, unitary, Fourier operator

Sky emission

Noé Dia

Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop

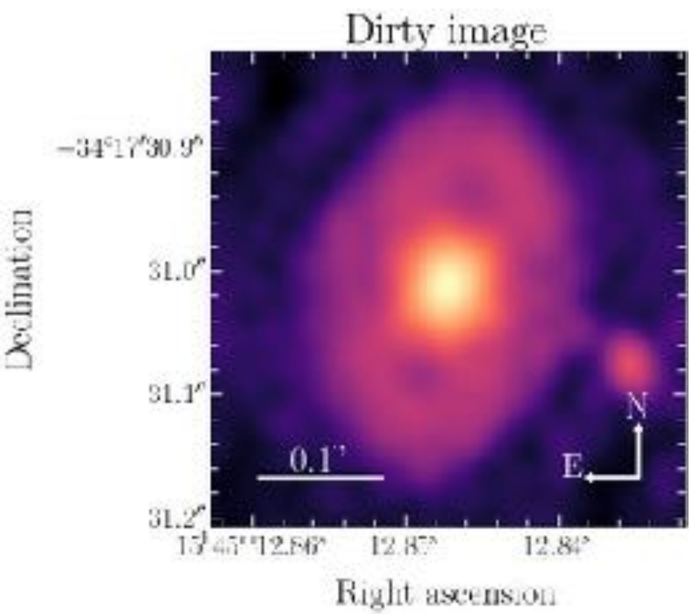
PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

- Observed visibilities → $\mathcal{V}$
- Mask, subsampling function → S
- Dense, unitary, Fourier operator → $\mathcal{F}$
- Primary beam → P_{beam}
- Sky emission → $\mathbf{x}$
- Gaussian noise → $\boldsymbol{\eta}$

Noé Dia



Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop

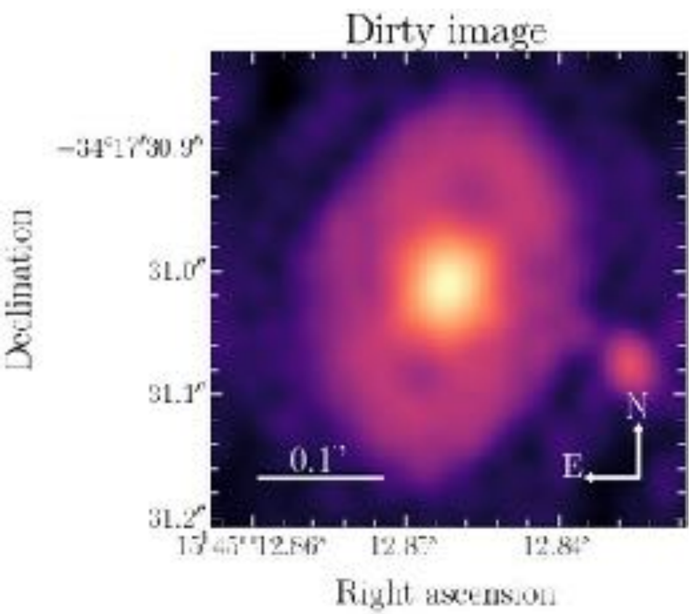
PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

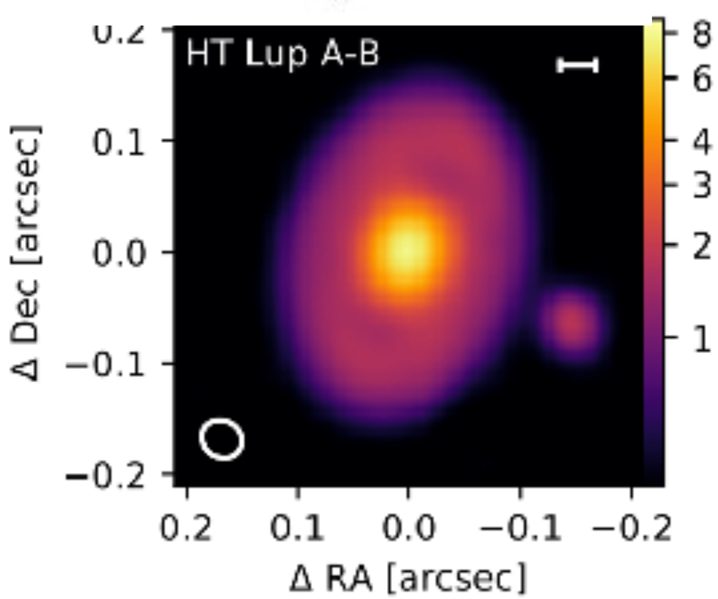
With Score-Based Diffusion Models

Noé Dia

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

- Observed visibilities → $\mathcal{V}$
- Mask, subsampling function → S
- Dense, unitary, Fourier operator → $\mathcal{F}$
- Primary beam → P_{beam}
- Sky emission → $\mathbf{x}$
- Gaussian noise → $\boldsymbol{\eta}$





Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

Noé Dia

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Observed visibilities → $\mathcal{V}$

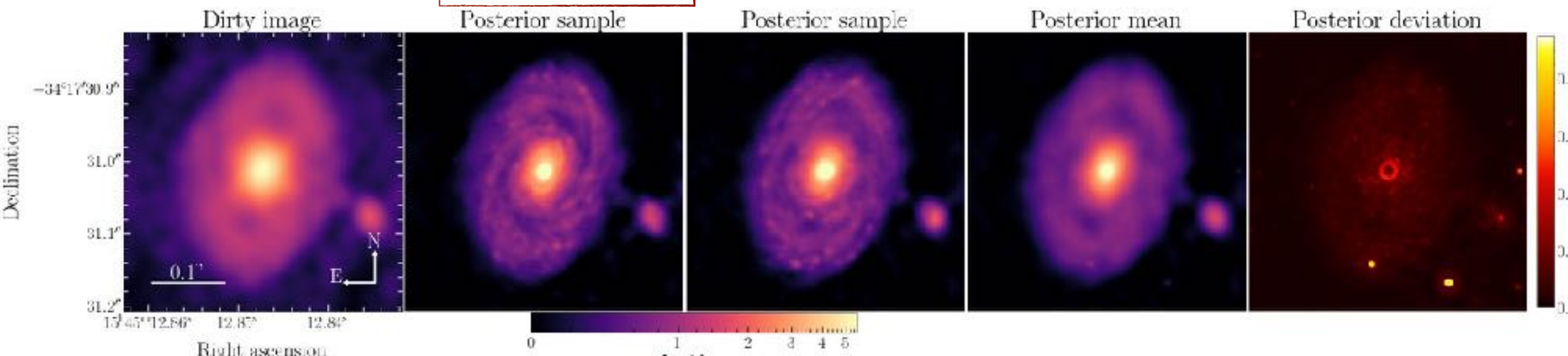
Mask, subsampling function → S

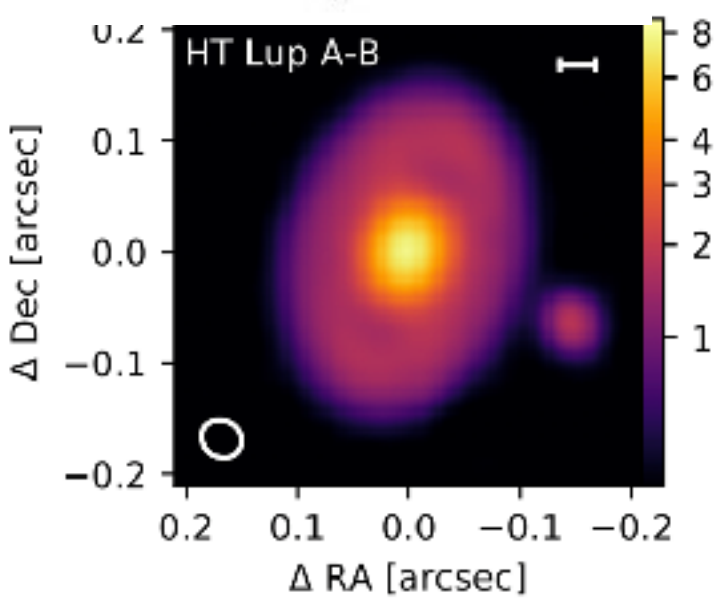
Dense, unitary, Fourier operator → $\mathcal{F}$

Primary beam → P_{beam}

Sky emission → $\mathbf{x}$

Gaussian noise → $\boldsymbol{\eta}$





Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop

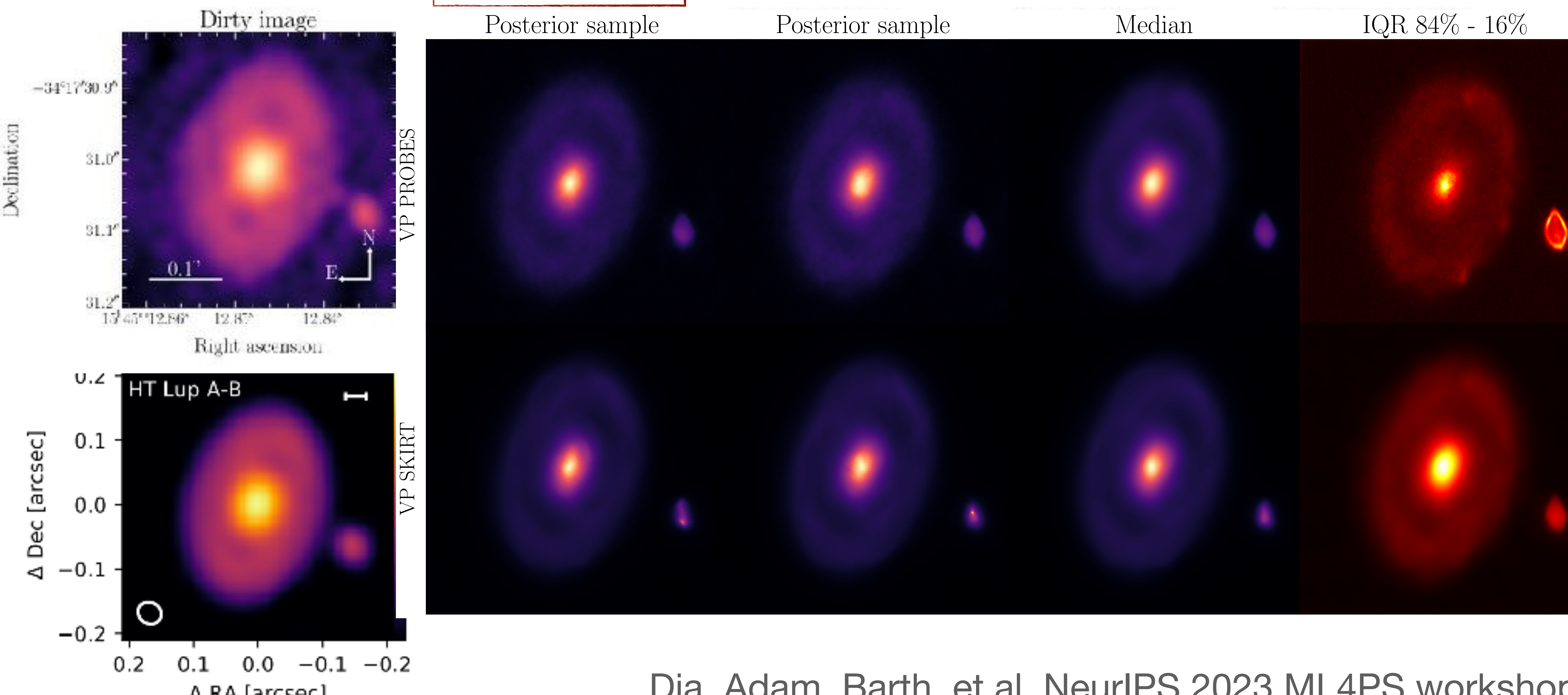
PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

- Observed visibilities → $\mathcal{V}$
- Mask, subsampling function → S
- Dense, unitary, Fourier operator → $\mathcal{F}$
- Primary beam → P_{beam}
- Sky emission → $\mathbf{x}$
- Gaussian noise → $\boldsymbol{\eta}$

Noé Dia



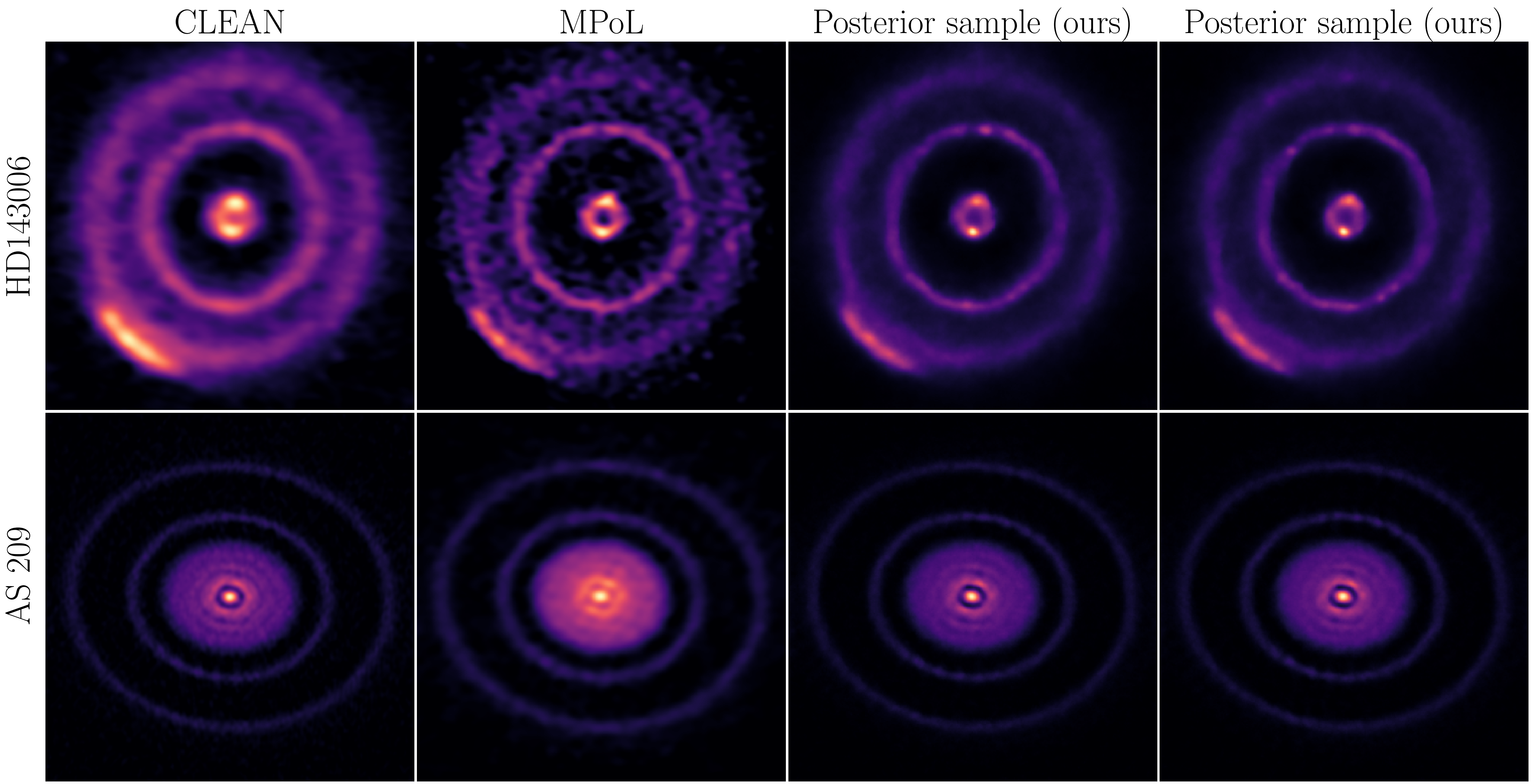
Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop

PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

$$\mathcal{V} = S\mathcal{F}P_{\text{beam}}\mathbf{x} + \boldsymbol{\eta}$$

Noé Dia



Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop

Strong lensing simulation pipeline: Caustic

A fast, AI-empowered, differentiable, extremely modular simulation pipeline for all your strong lensing needs.

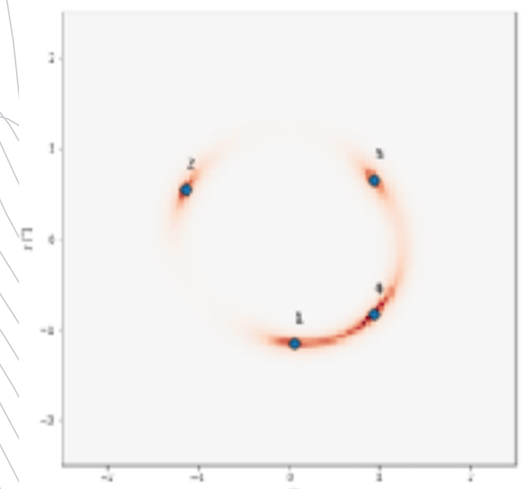
1) Lens and source from analytic profiles or pixelated images/densities

2) Multiplane lensing

3) Line of sight mass distributions

4) Fast microlensing simulations

5) Time-delays




Connor Stone


Adam Coogan


Andi Filipp


Alex Adam


Misha Barth


Charles Wilson

https://github.com/Ciela-Institute/caustic

https://ciela-institute.github.io/caustic/BasicIntroduction.html