Strong Gravitational Lensing Systems with Machine Learning: first results on Dark Matter substructures

Marco Chianese

Advanced Workshop on Accelerating the Search for Dark Matter with Machine Learning

8-12 April 2019, Trieste

UNIVERSITY OF AMSTERDAM

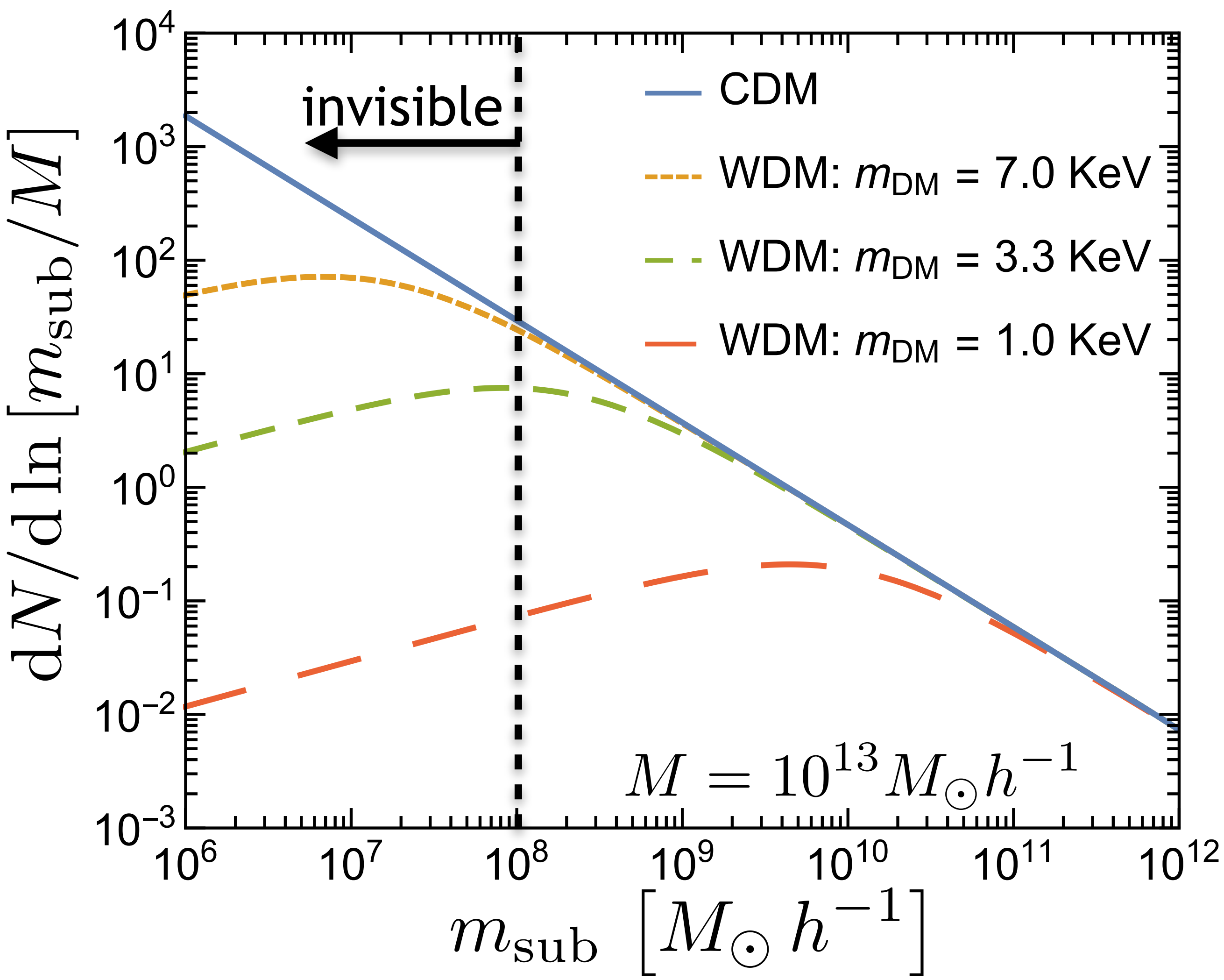
GRAPPA

GRavitation AstroParticle Physics Amsterdam

Dark Matter substructures

In ΛCDM model, structures are formed hierarchically from the collapse of small-scale fluctuations. Their formation depends on the ***coldness of dark matter*** that provides a halo mass cut-off:

$$M_{\rm cut} = 10^{10} \left(\frac{m_{\rm MD}}{1\,{\rm KeV}} \right)^{-3.33} M_\odot h^{-1}$$



A key quantity is the subhalo mass function:

$$\frac{dN}{d\ln\xi} = (1+z)^{1/2} A_M\, \xi^\alpha \exp\left(-\beta\, \xi^3\right) \left(1 + \frac{M_{\rm cut}}{M}\xi\right)^\gamma \quad \text{where} \quad \xi = m_{\rm sub}/M$$

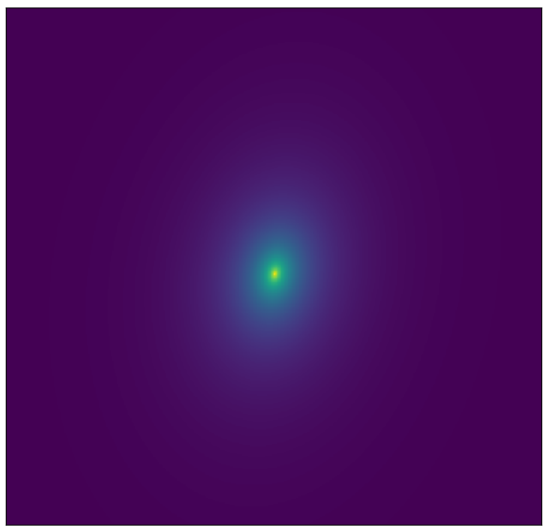
Cold DM **Warm DM**

Gao et al., MNRAS 455 (2004)
Giocoli et al., MNRAS 404 (2010)
Gao et al., MNRAS 410 (2011)
Schneider et al., MNRAS 424 (2012)
Lovell et al., MNRAS 439 (2014)
Han et al., MNRAS 457 (2016)
Despali and Vegetti, MNRAS 469 (2017)
Gilman et al., arXiv:1901.11031
Ando et al., arXiv:1903.11427

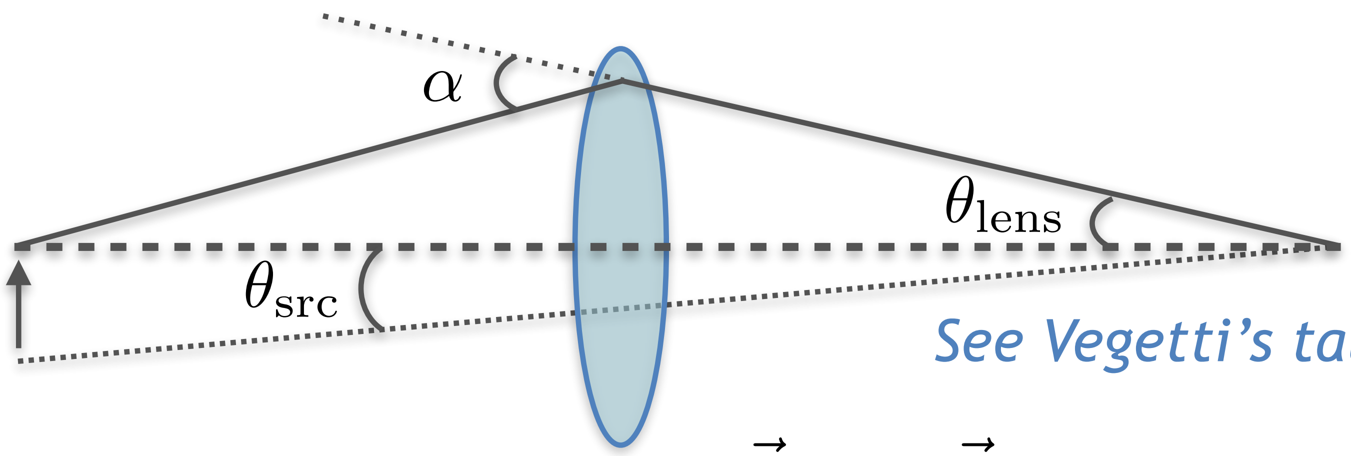
Strong Gravitational Lensing

We want to infer at the same time the structure of the background source galaxy and the total mass distribution (gravitational potential) of the foreground lens galaxy.

Source

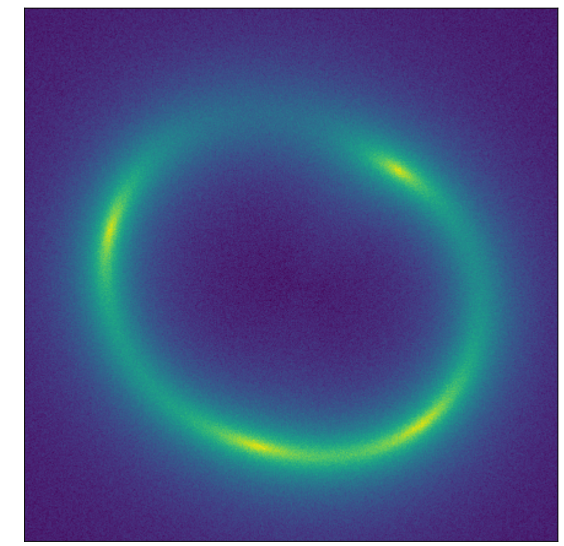


Main Lens



See Vegetti's talk

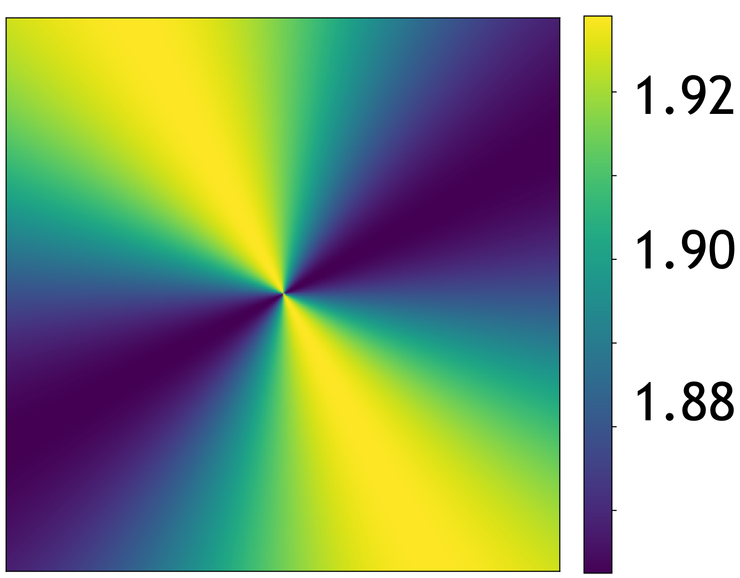
What we observe



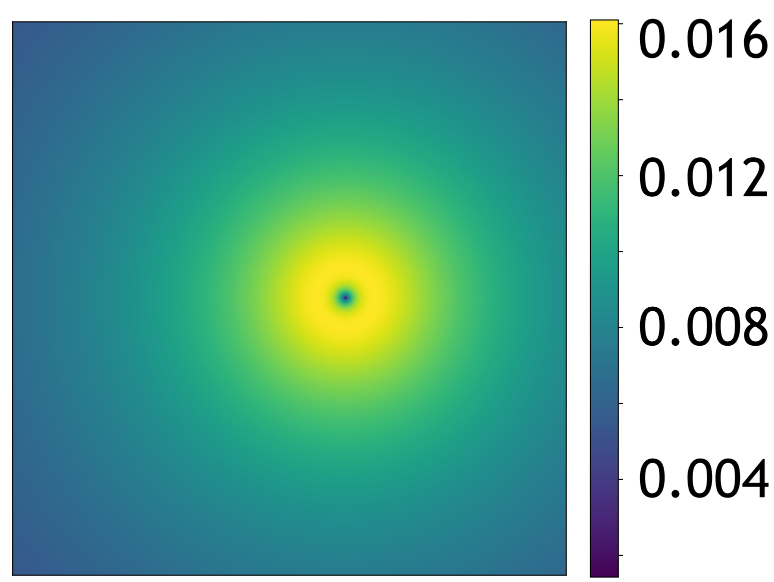
Solve the lens equation: $\vec{\theta}_{\rm src} = \vec{\theta}_{\rm lens} - \vec{\alpha}$

The effect of dark matter subhaloes is subdominant and localized w.r.t. the main halo.

Deflection due to main halo



Deflection due to subhalo

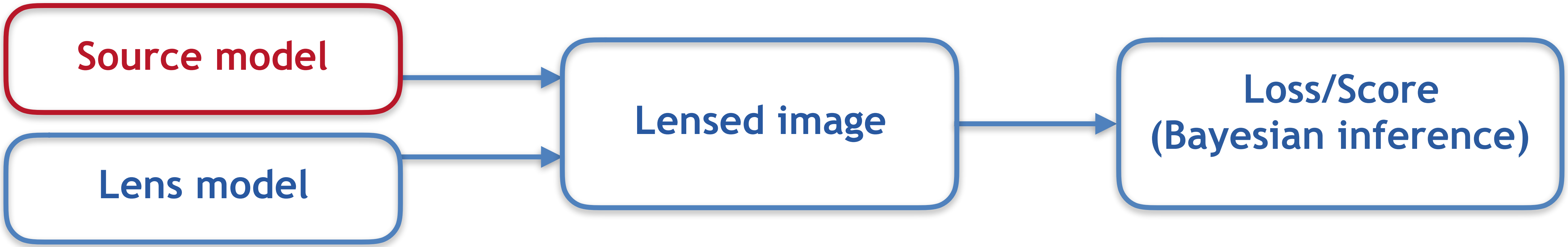


$$m_{\rm sub} = 10^{10}\, M_\odot$$

$$\downarrow$$

$$\Delta\alpha \sim 0.9\%$$

Analysis pipeline



Two methods have been exploited so far:

Parametric models

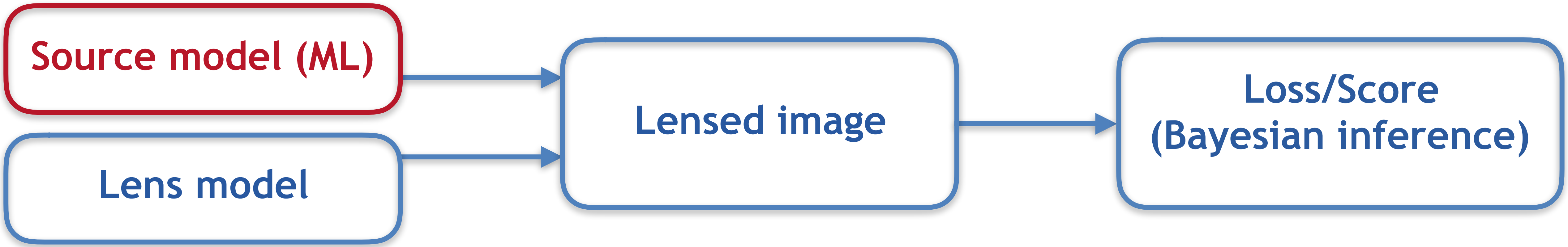
- **Pros:** analytical models that depend on a set of parameters (Sersic profile).
- **Cons:** oversimplified models that do not cover all the realistic galaxies with complex morphology (bulge, disk, …).

Non-parametric models

- **Pros:** more freedom due to image reconstruction pixel by pixel.
- **Cons:** no *a priori* information from real galaxies and regularization according to some criteria (like smoothness).

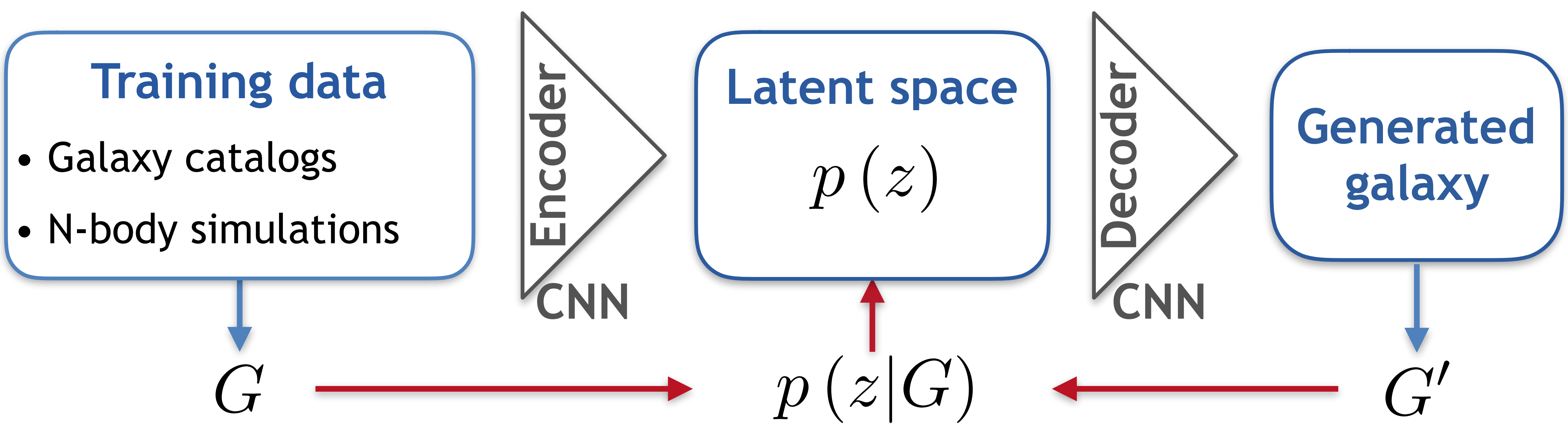
Koopmans, MNRAS 363 (2005)
Vegetti and Koopmans, MNRAS 392 (2009)

Analysis pipeline



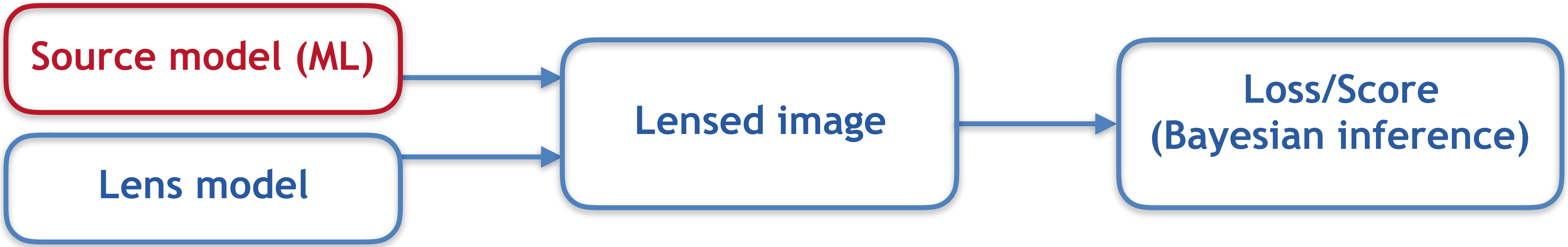
We adopt **generative models** as **Variational AutoEncoder** to model the surface brightness of the source galaxy.

See Coogan's talk!



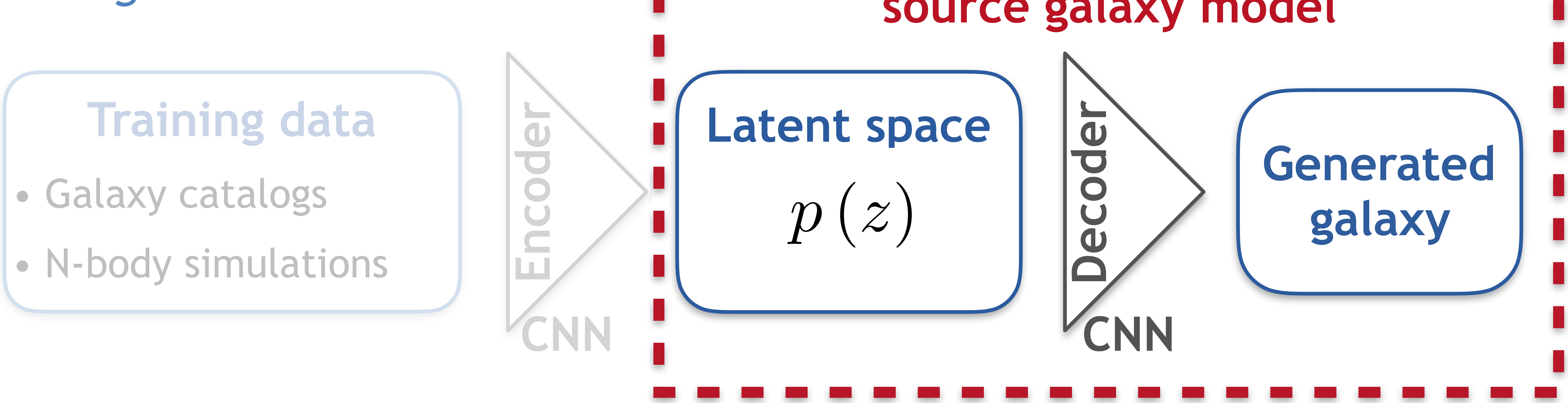
See also Hendriks' talk!

Analysis pipeline



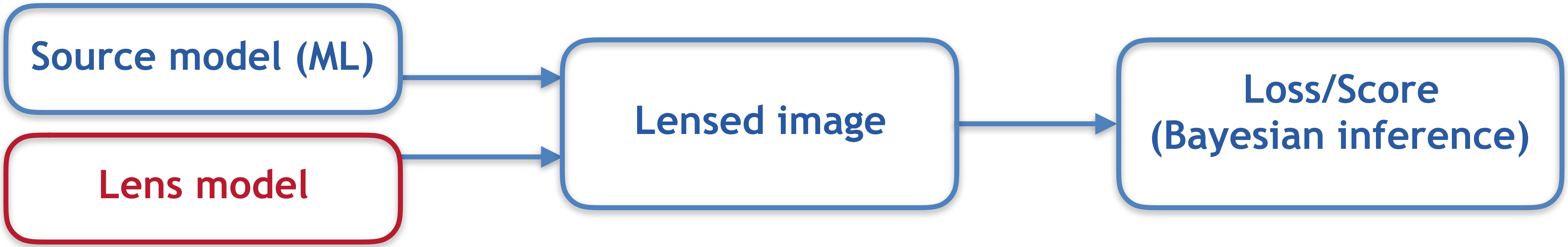
We adopt **generative models** as **Variational AutoEncoder** to model the surface brightness of the source galaxy.

See Coogan's talk!



See also Hendriks' talk!

Analysis pipeline



1. Define a pixel grid according to the observed image

2. Compute the total projected mass (main lens + subhaloes)

$$\kappa_{ij} = \kappa_{\rm ml}\left(\theta^x_{ij}, \theta^y_{ij}\right) + \kappa_{\rm sub}\left(\theta^x_{ij}, \theta^y_{ij}\right) + \ldots$$

3. Compute the displacement field via a 2-dimensional convolution

$$\alpha(\vec{\theta}) = \int \kappa(\vec{\theta}')\, F(\vec{\theta} - \vec{\theta}')\, d^2\vec{\theta}'$$

with $$F(\vec{\theta} - \vec{\theta}') = \frac{1}{\pi} \frac{\vec{\theta} - \vec{\theta}'}{|\vec{\theta} - \vec{\theta}'|^2}$$

Discrete convolution $$\alpha_{ij} = \sum_{i'j'} F\left[i - i', j - j'\right] \kappa\left[i', j'\right]$$

4. Add the external shear contribution

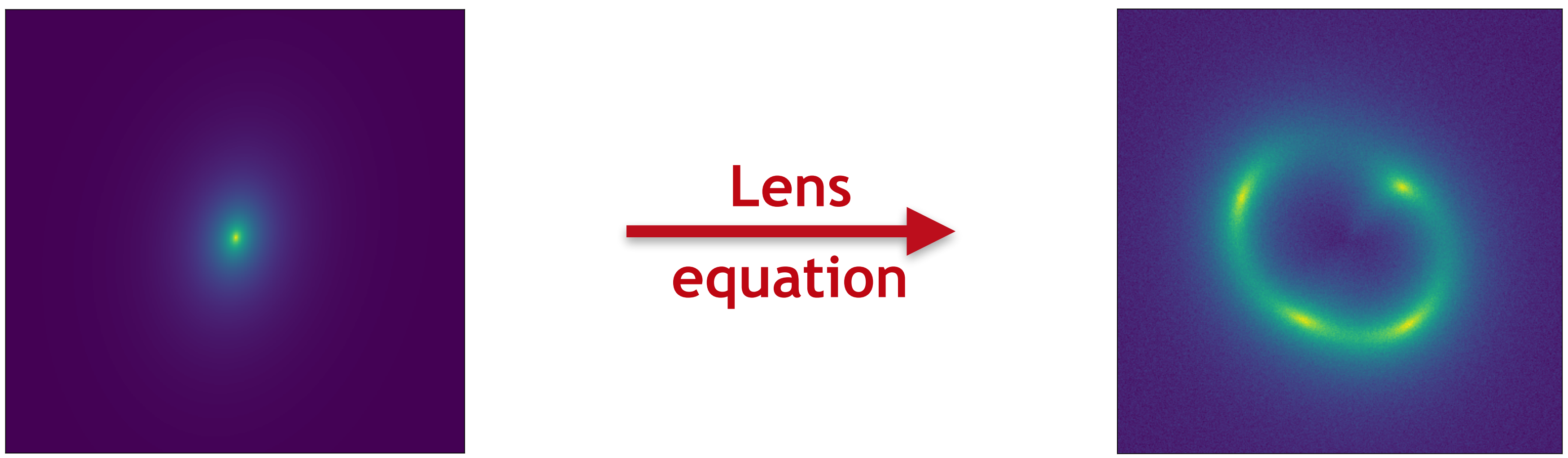
$$\alpha^x_{ij} = \alpha_{ij} + \alpha^{\rm ext}_{ij}$$
$$\alpha^y_{ij} = \alpha_{ij} + \alpha^{\rm ext}_{ij}$$

→ **Total displacement field**

Analysis pipeline

1. Compute the brightness in the lens plane

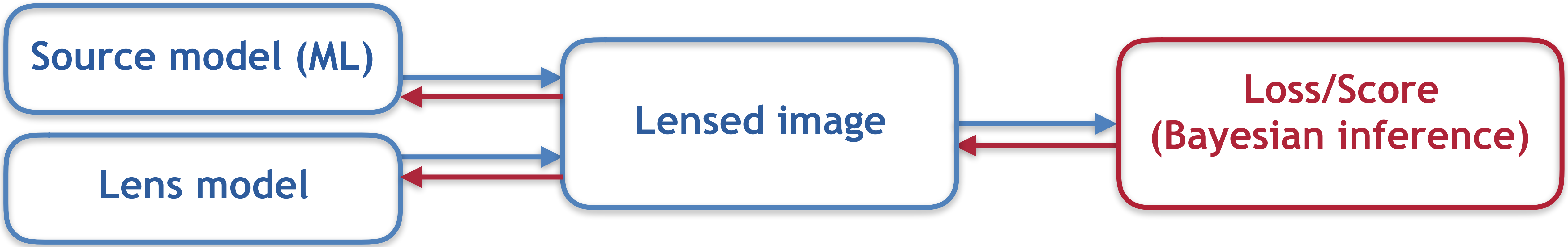
$$f_{\rm src}(\tilde{\theta}^x_{ij},\, \tilde{\theta}^y_{ij}) \qquad f_{\rm src}(\theta^x_{ij} - \alpha^x_{ij},\, \theta^y_{ij} - \alpha^y_{ij})$$



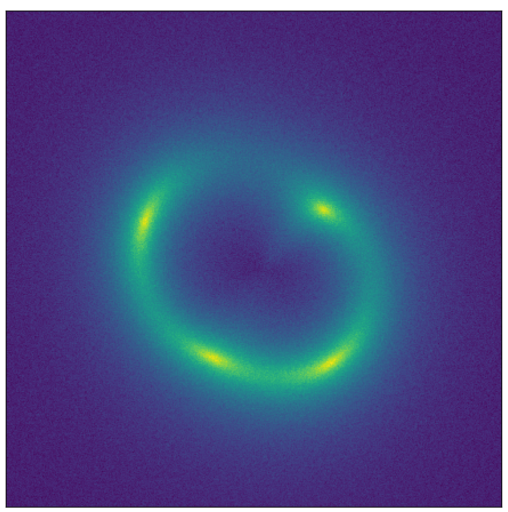
2. Apply Point Spread Function (PSF)

3. Add Gaussian noise

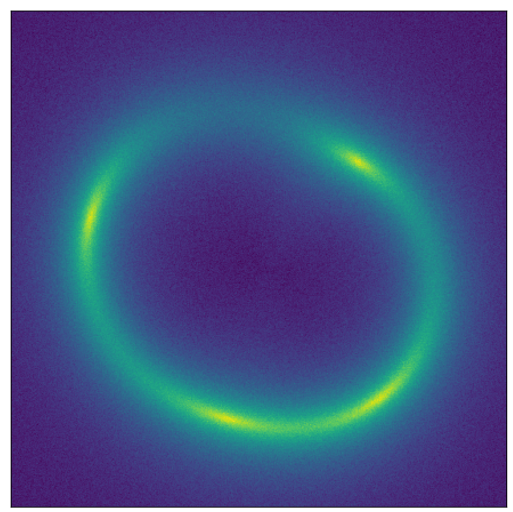
Analysis pipeline



Loss/Score



(model)



(data)

Posteriors

$$p\,(x|D) \sim p\,(D|x)\;p\,(x)$$

Priors: $p(x) = p(x_{\rm src})\,p(x_{\rm lens})$

Sampling techniques with gradient descent

- Hamiltonian Monte Carlo (HMC)
- Stochastic Variational Inference (SVI)

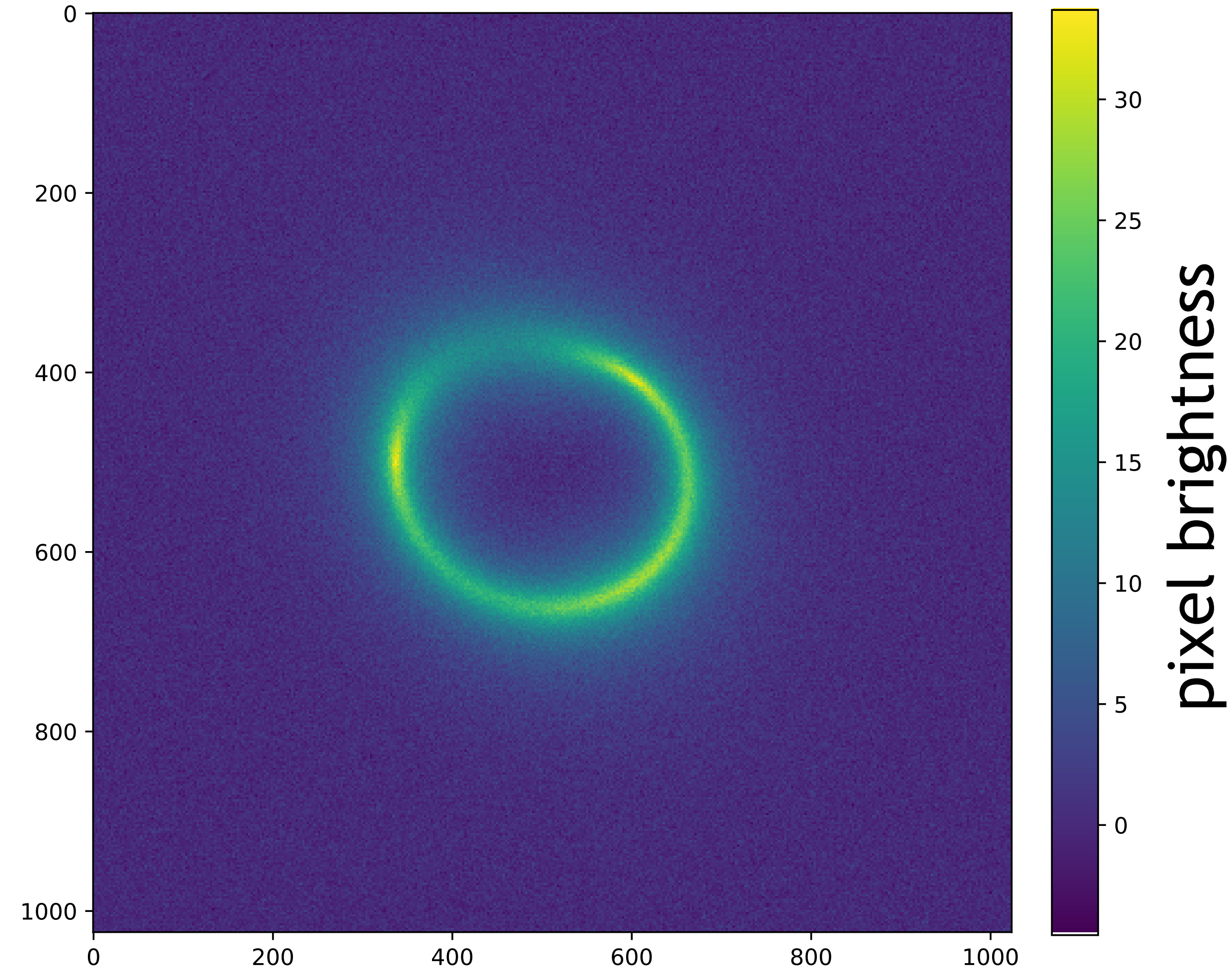
PYTORCH + Pyro

See Thomas and Weniger's talks

Pipeline test

In order to test our Bayesian inference pipeline, we generate and analyze mock data.

- **Optical image**: area of 10x10 arcsec^2 with 1024x1024 pixels.
- **Source**: analytical Sersic profile.
- **Main lens**: Singular Power-Law Ellipsoid plus external shear.
- **Subhaloes**: 1 truncated NFW.
- **Total number of parameters**: 13 + 4 (subhalo) = 17



Due to the presence of several local minima in the 17-dimensional space, we perform:

- **Optimization**: find the best initial values for the parameters by means of SVI with guide

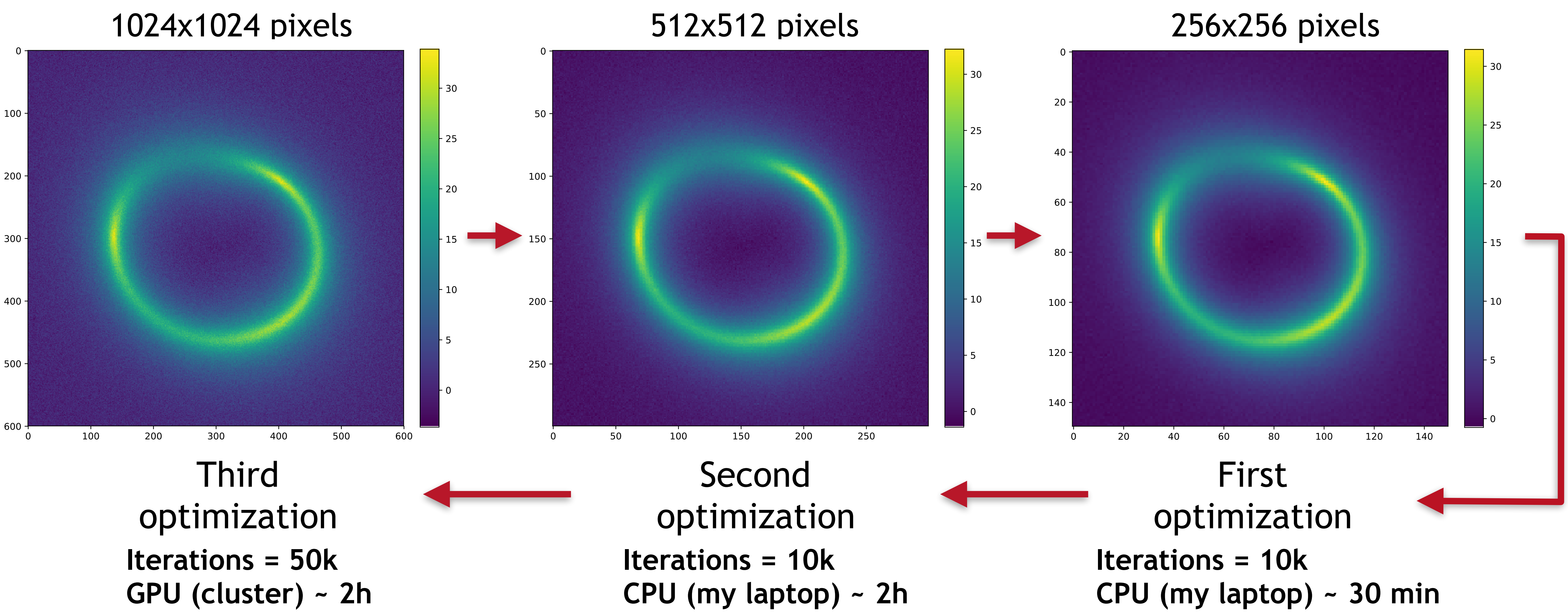
$$q_\theta\left(x\right) = \delta\left(x - \theta\right)$$

- **Bayesian inference**: obtain the parameters' posteriors by means of HMC

Optimization

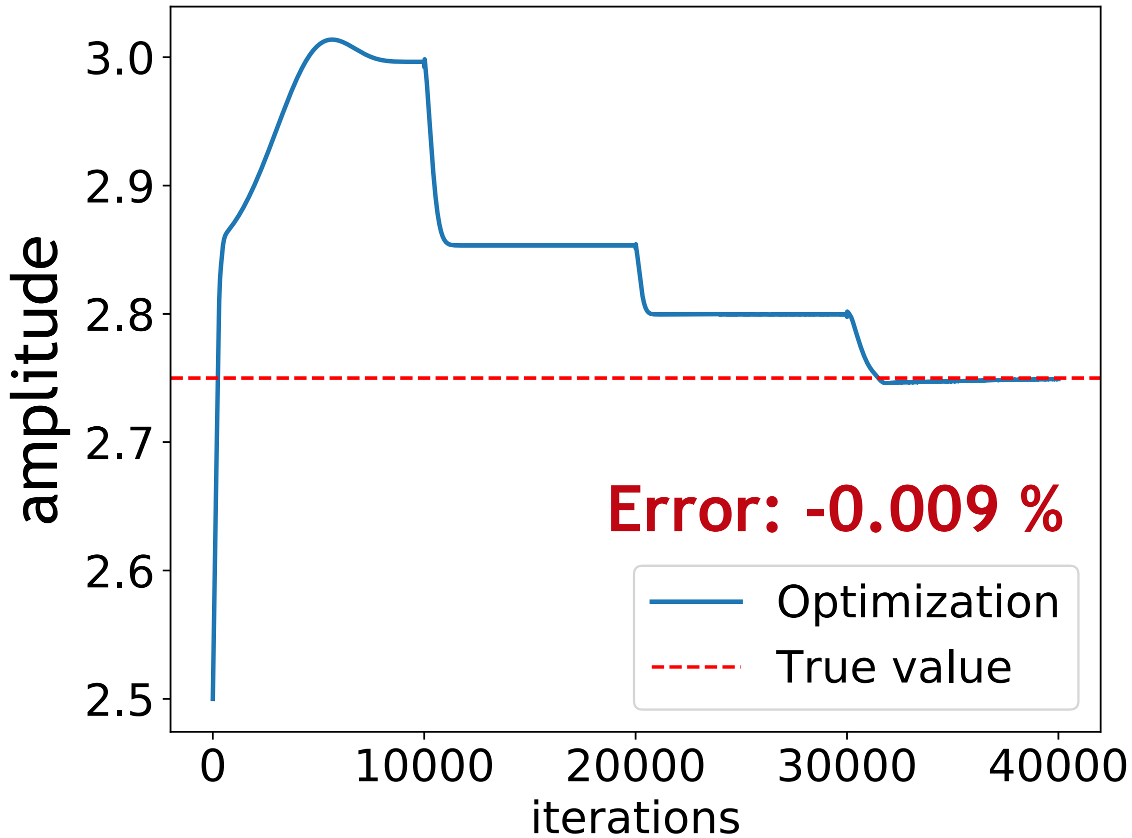
The optimization is based on the following steps:

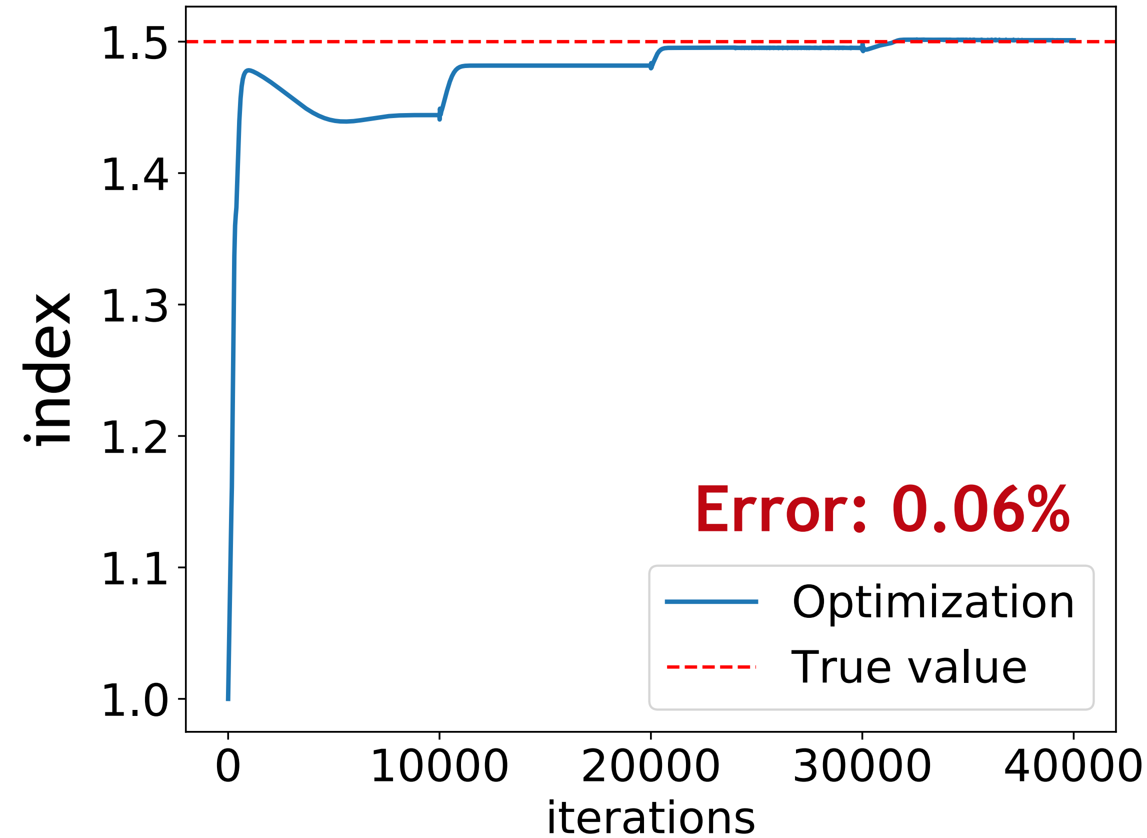
1. **Hierarchical modeling**: start with a lensing model without including the substructure.
2. **Down-sampling**: consider a smaller image.



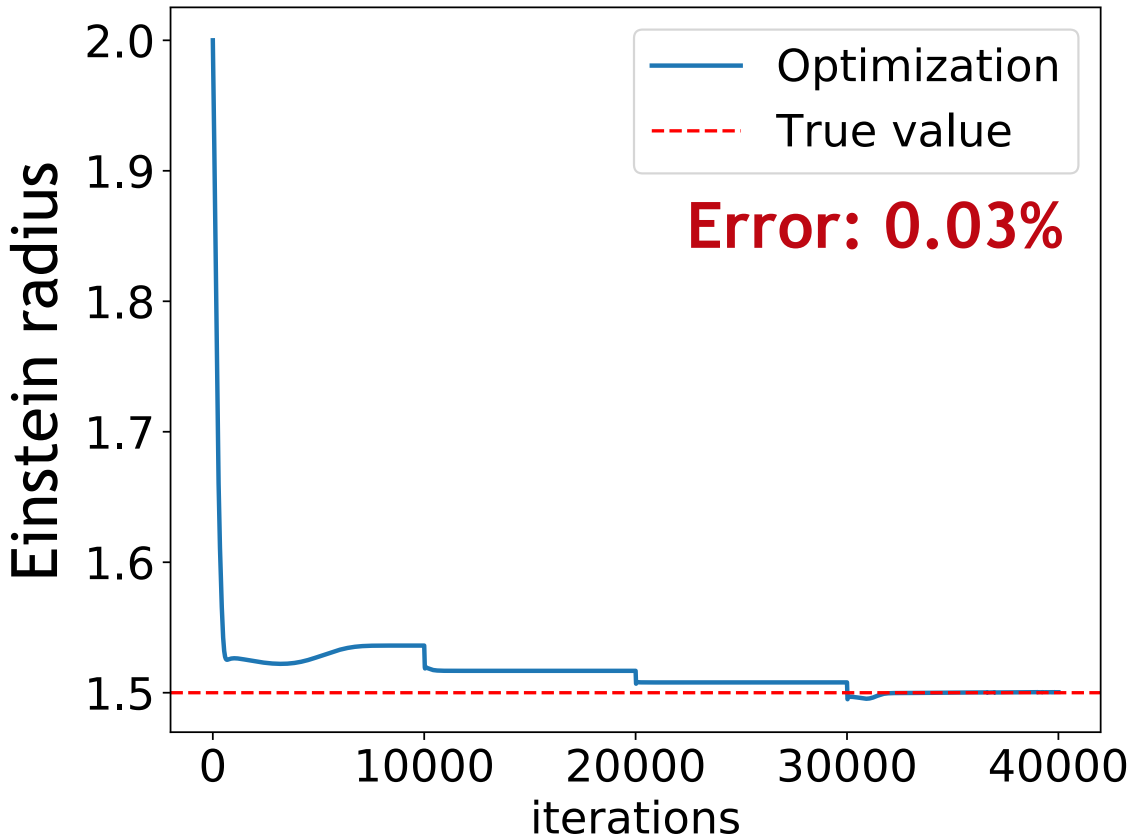
3. **Including the substructure**: perform the last optimization with the substructure.

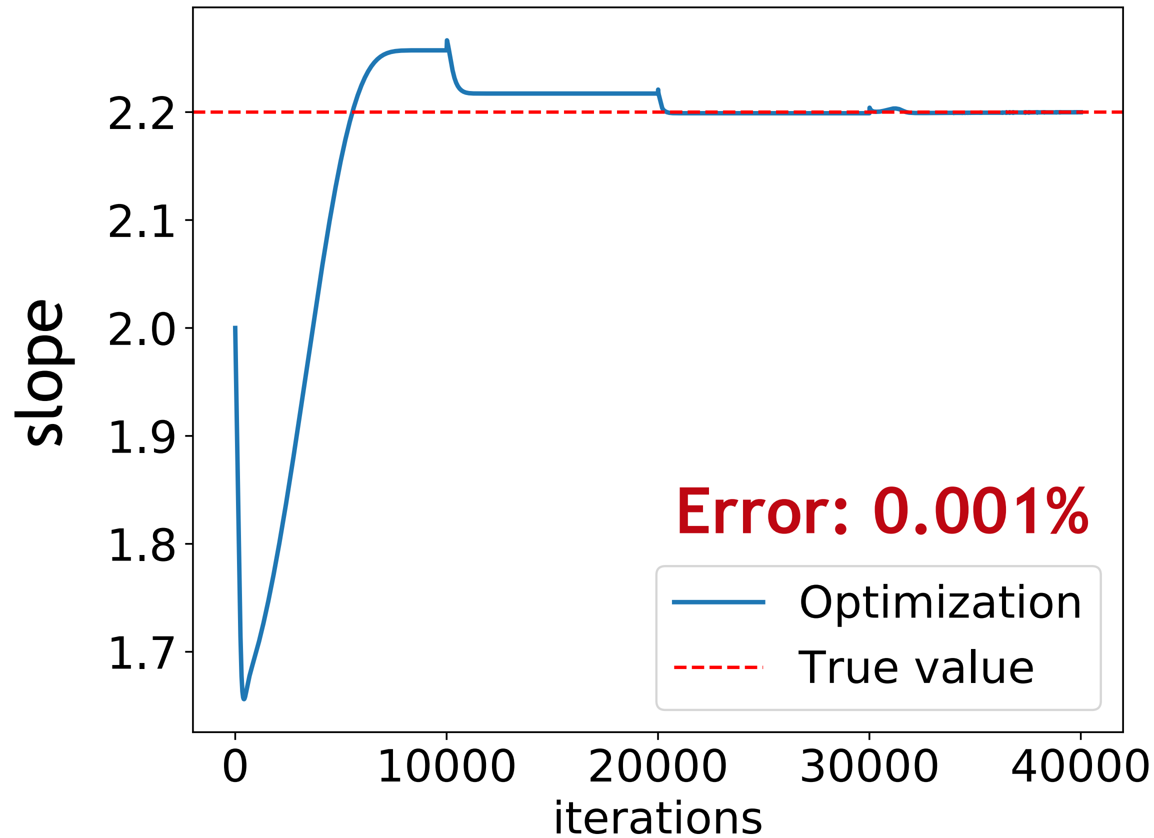
Optimization: results





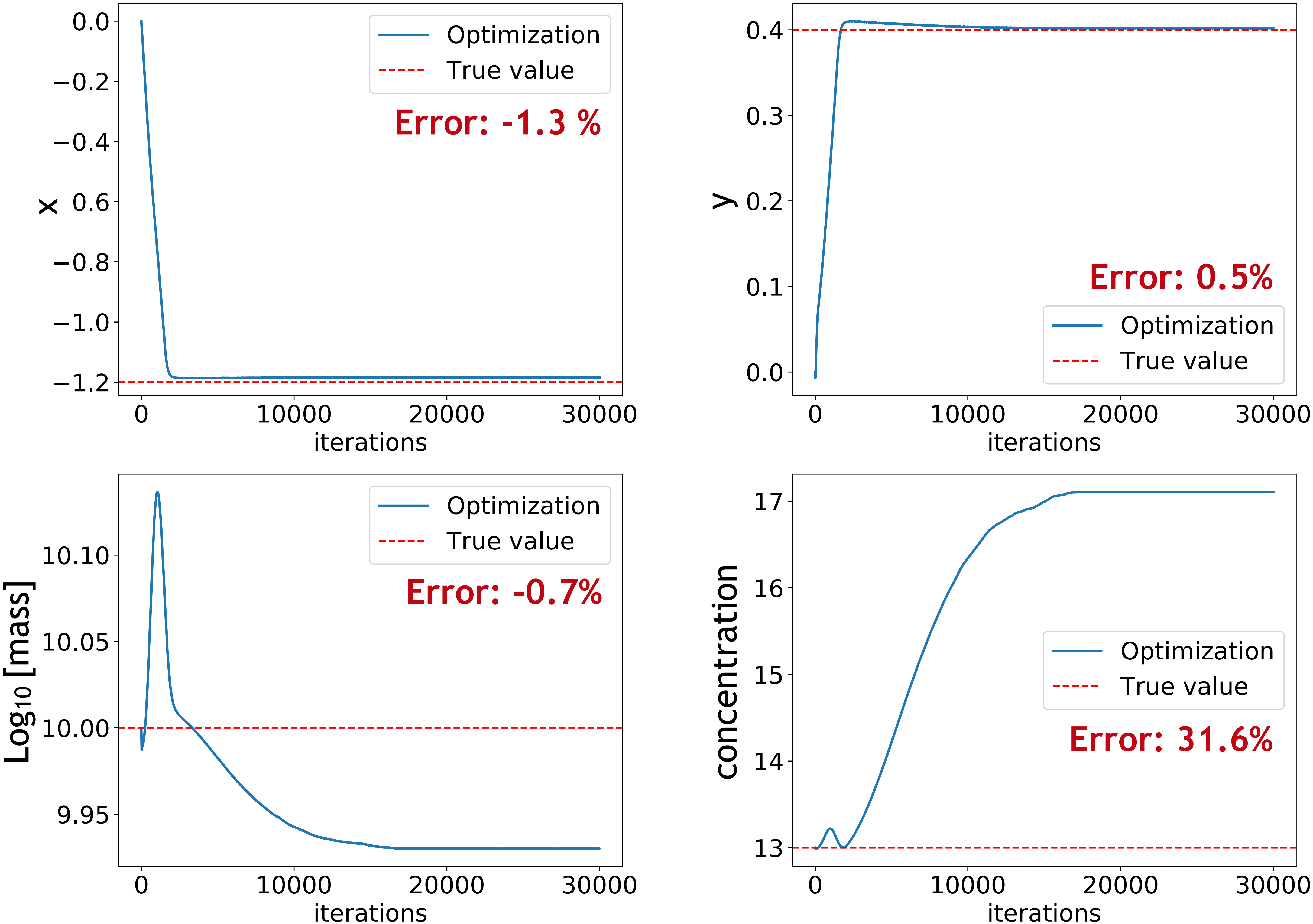
Source





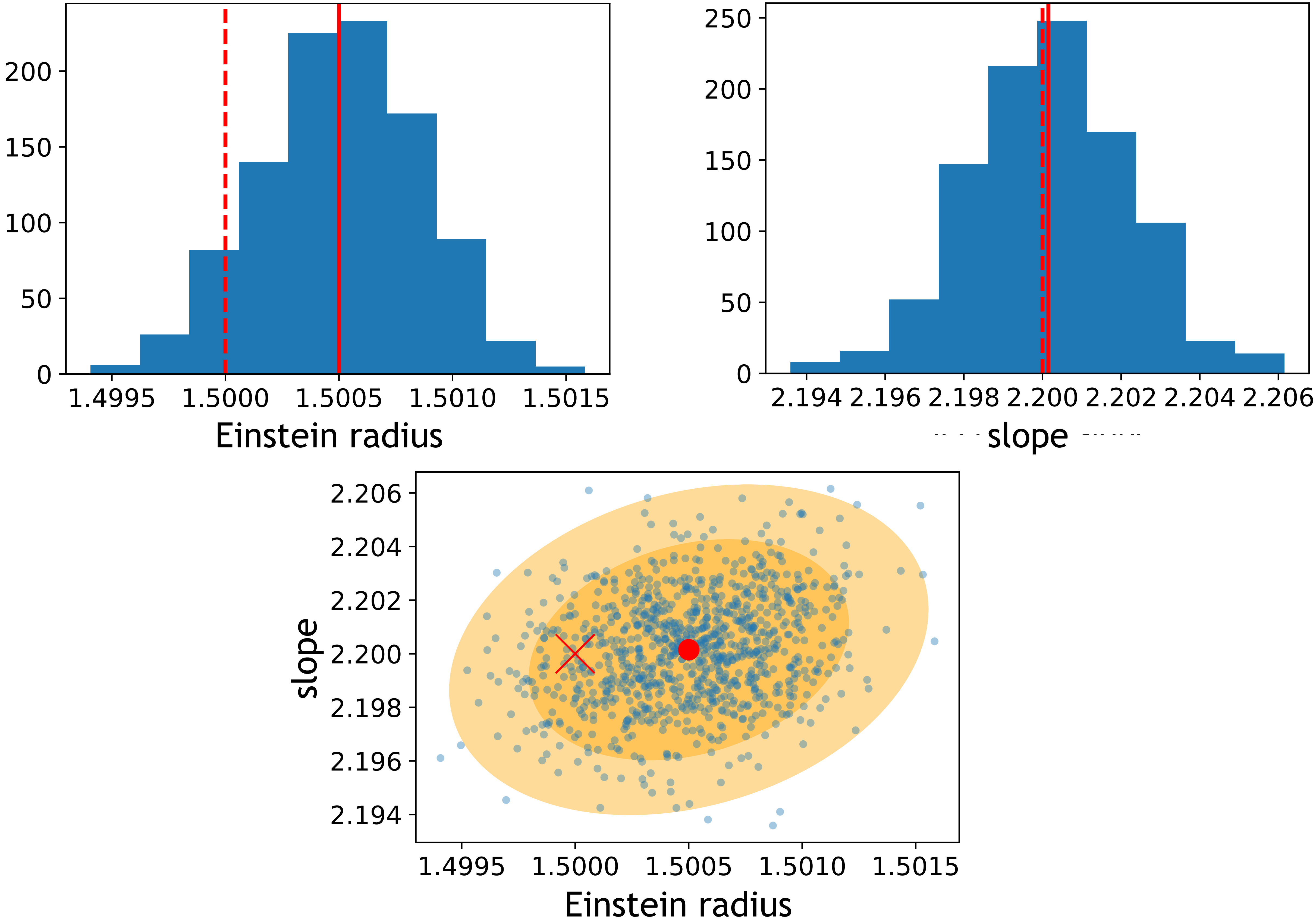
Main Lens

Optimization: subhalo



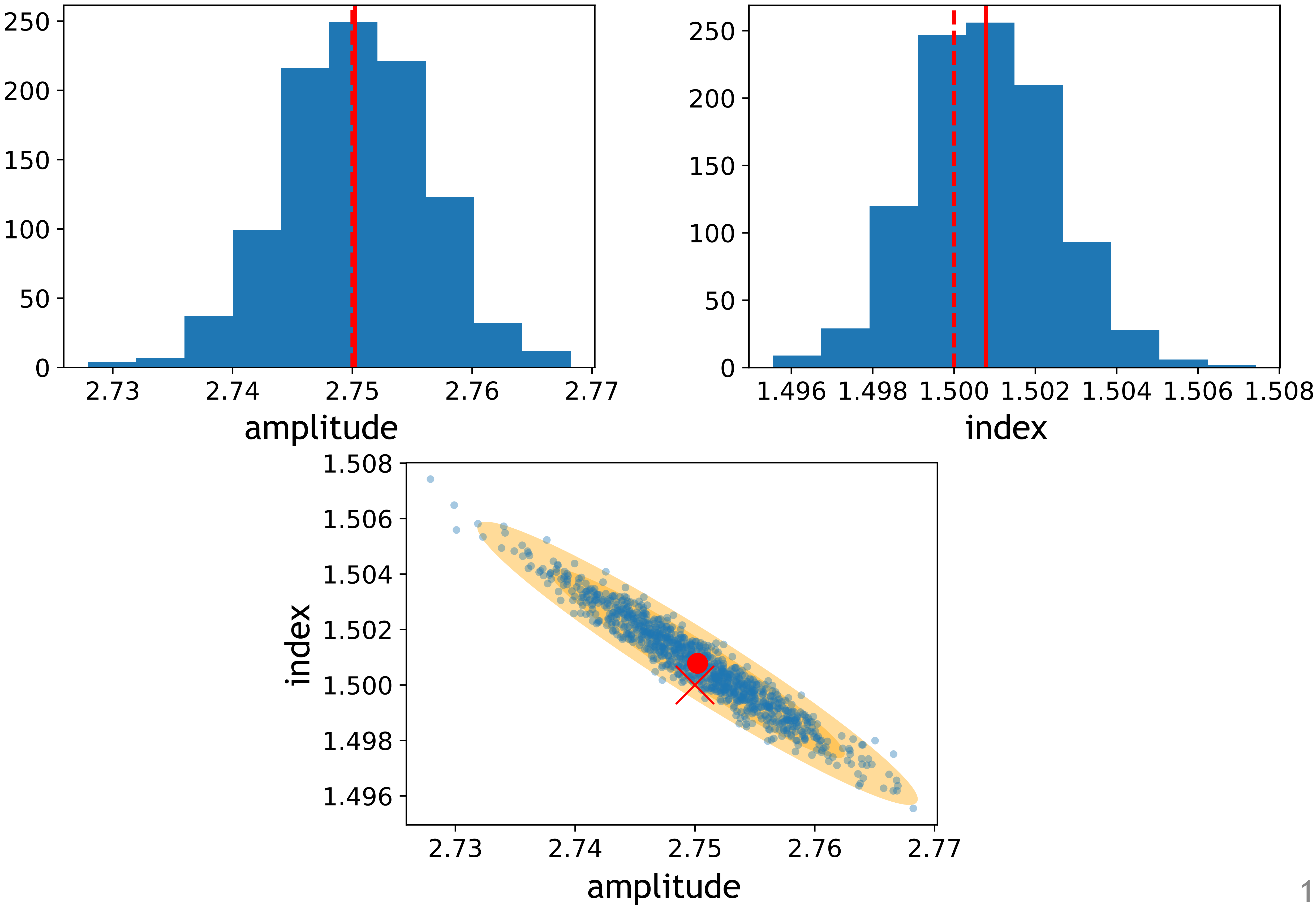
truncated NFW subhalo

HMC: lens



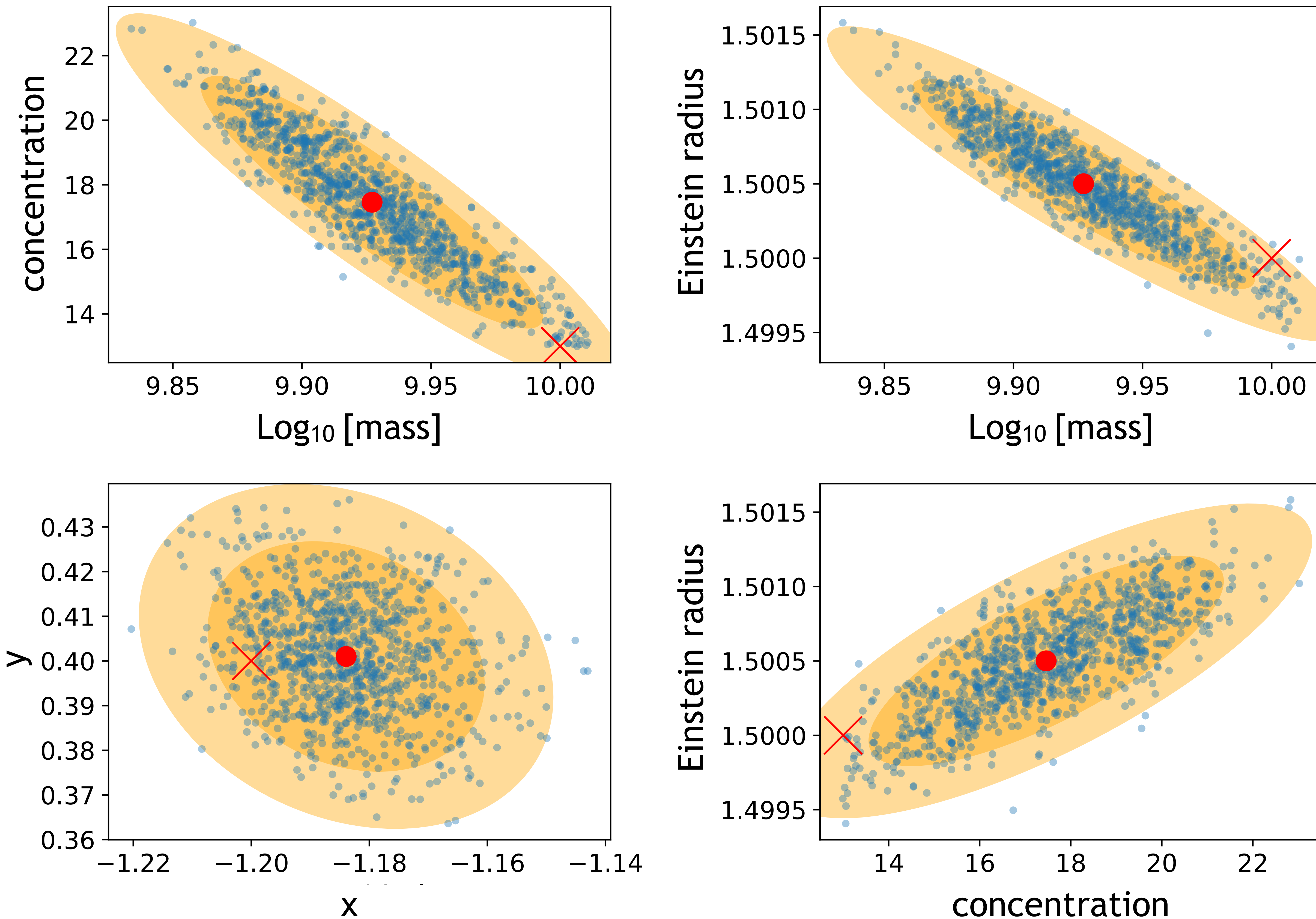
200
150
100
50
0
1.4995 1.5000 1.5005 1.5010 1.5015
Einstein radius
250
200
150
100
50
0
2.194 2.196 2.198 2.200 2.202 2.204 2.206
slope
2.206
2.204
2.202
2.200
2.198
2.196
2.194
slope
1.4995 1.5000 1.5005 1.5010 1.5015
Einstein radius

HMC: source



250
200
150
100
50
0
2.73
2.74
2.75
2.76
2.77
amplitude
250
200
150
100
50
0
1.496
1.498
1.500
1.502
1.504
1.506
1.508
index
1.508
1.506
1.504
1.502
1.500
1.498
1.496
index
2.73
2.74
2.75
2.76
2.77
amplitude

HMC: subhalo



concentration
Log10 [mass]
Einstein radius
y
x
concentration
Einstein radius

Conclusions

We have developed a fast Bayesian analysis pipeline for strong lensing data in a deep probabilistic programming framework.

Optimization
Iterations: 50 000
Time: 2 hours (GPU)

→

Inference (HMC)
Iterations: 1 000
Time: 20 hours (GPU)

We have studied the feasibility of inference techniques with gradient descent (HMC and SVI) for systems with high-dimensional parameters space.

The next steps are:

- Systematic study of the sensitivity to substructures
- Inference on real strong lensing data
- Estimation of the Bayesian evidence (models discrimination)
- Couple to **generative models** for the background source galaxy *See Coogan's talk!*

Conclusions

We have developed a fast Bayesian analysis pipeline for strong lensing data in a deep probabilistic programming framework.

Optimization
Iterations: 50 000
Time: 2 hours (GPU)

→

Inference (HMC)
Iterations: 1 000
Time: 20 hours (GPU)

We have studied the feasibility of inference techniques with gradient descent (HMC and SVI) for systems with high-dimensional parameters space.

The next steps are:

- Systematic study of the sensitivity to substructures
- Inference on real strong lensing data
- Estimation of the Bayesian evidence (models discrimination)
- Couple to **generative models** for the background source galaxy *See Coogan's talk!*

Thank you for your attention

Backup slides

Pytorch and Pyro

In **Pytorch**, the key features are:

PYTORCH

- automatic *differentiation*

`M = model(x, ...)` with `x = torch.Tensor(..., requires_grad=True)`

`M.backward()` ⟷ $\partial M / \partial x$

- simple implementation on GPU with: `M.cuda()`

In **Pyro**, the basic units are primitive stochastic functions:

- double role as *probability distribution* and *random number generator*

```
x = pyro.sample("param", dist.Normal(mean, sigma))
image = pyro.sample("image", dist.Normal(M, noise))
```

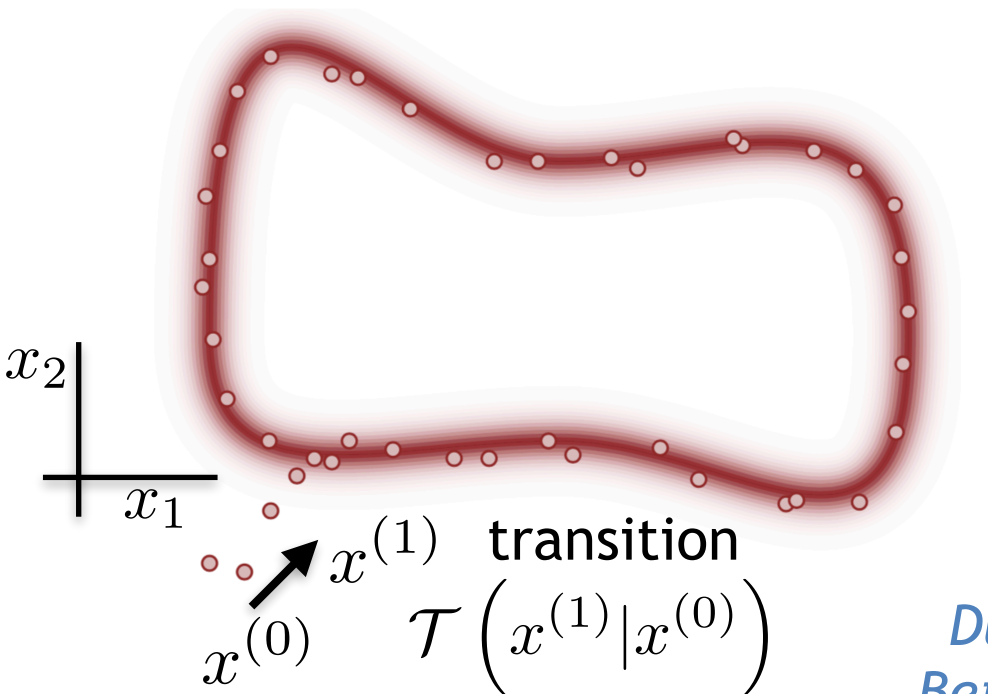
- can be conditioned on data

```
conditioned_image = pyro.condition(image, data={"obs": obs_image})
```

- used in *universal inference algorithms* (e.g. HMC, SVI)

HMC vs SVI

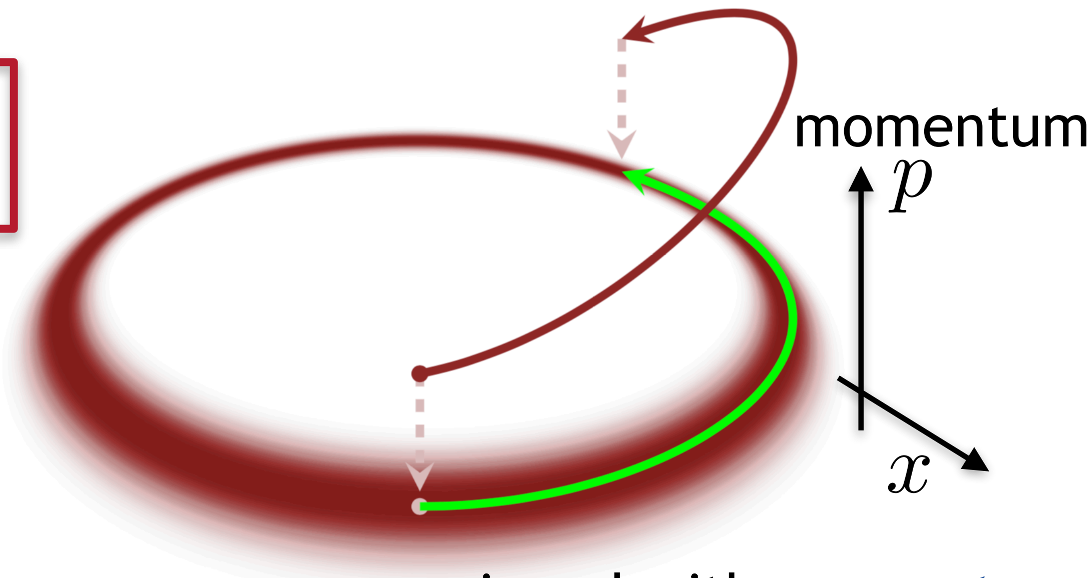
The **Hamiltonian Monte Carlo** (HMC) is a MCMC that exploits gradient information.



$$x \sim \pi\left(x\right)$$

HMC →

Duan et al., PLB 195 (1987)
Betancourt, arXiv:1701.02434



equipped with *momentum* and *Hamiltonian* $\mathcal{H}(x,p)$

The **Stochastic Variational Inference** (SVI) is a fast method for approximately posteriors.

- **Guide**: $q_\theta\left(x\right)$ → find θ so that $q_\theta\left(x\right) \sim p\left(x|D\right)$
- Maximize the **Evidence Lower BOund**

$$\mathrm{ELBO} \equiv \log p\left(D\right) - \mathrm{KL}\left(q_\theta\left(x\right), p\left(x|D\right)\right)$$

Kullback-Leibler divergence (measure of approximation error)

Bayesian evidence

Wingate and Weber, arXiv:1301.1299 *Ranganath, Gerrish and Blei, arXiv:1401.0118*