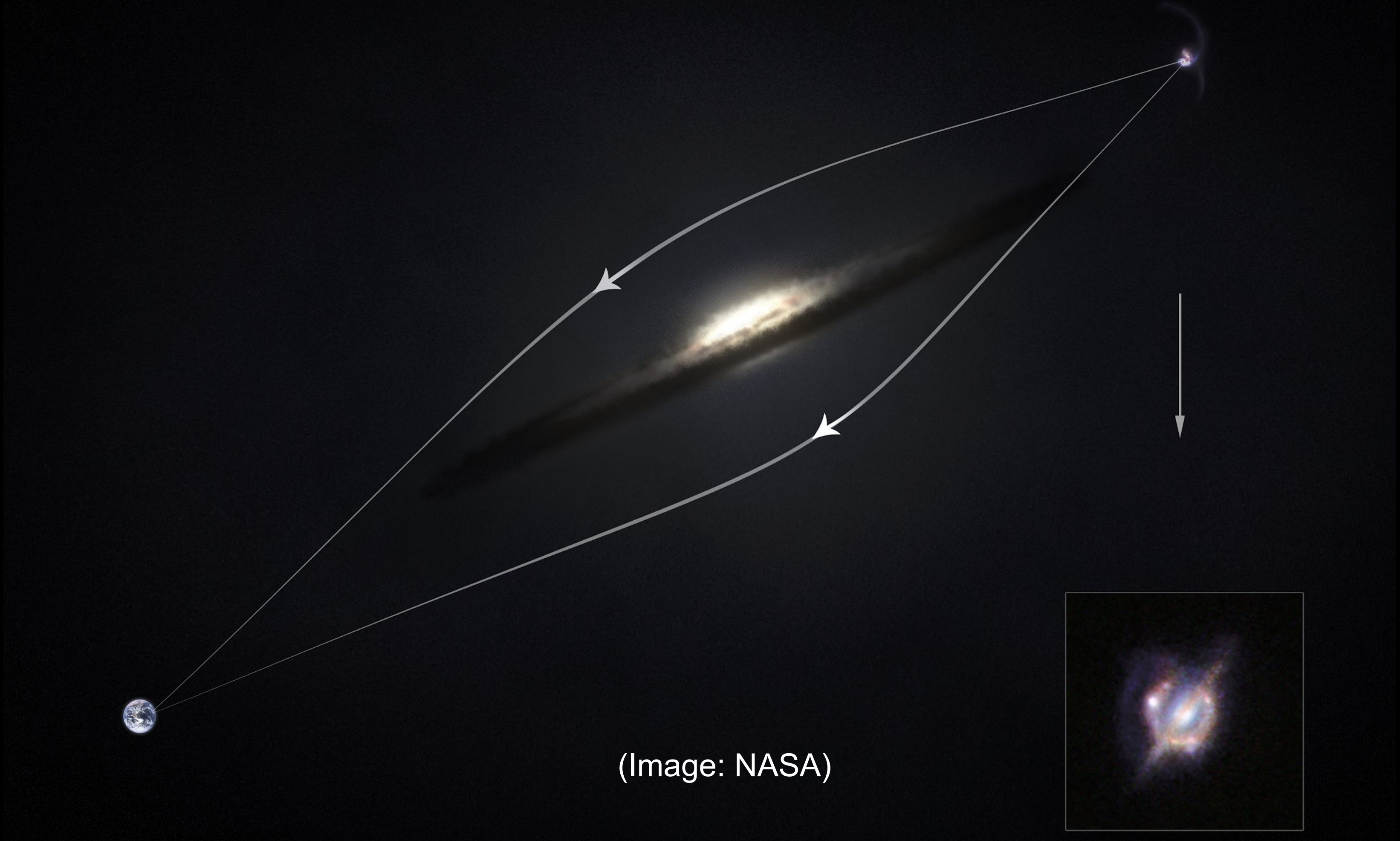
FERMILAB-SLIDES-22-078-SCD

Automated Lens Parameter Estimation using Simulation-Based Inference Methods

Jason Poh (he/him) (University of Chicago)
With A. Samudre, A. Ciprijanovic, B. Nord, J. Frieman, D. Tanoglidis, G. Khullar
Deep Skies Lab
June 22 2022
New Perspectives 2022

Kavli Institute for Cosmological Physics at The University of Chicago

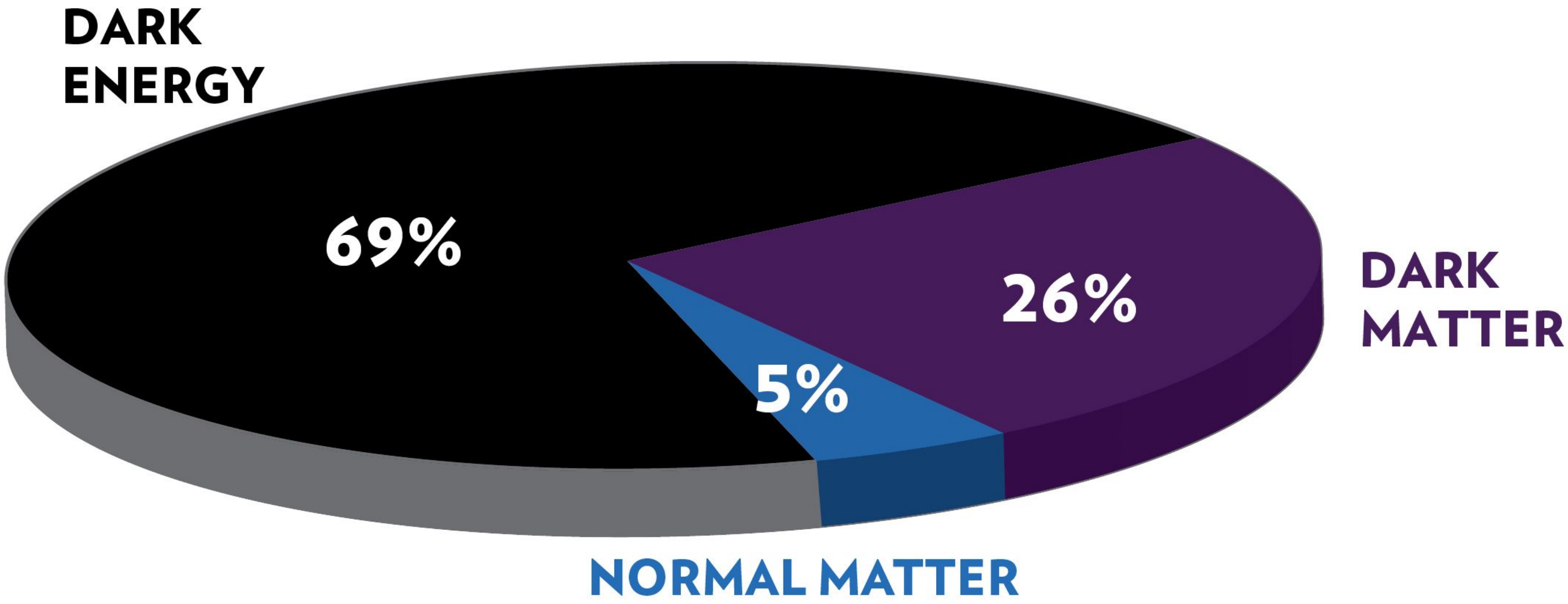
DEEP SKIES
Bringing Artificial Intelligence to Astrophysics



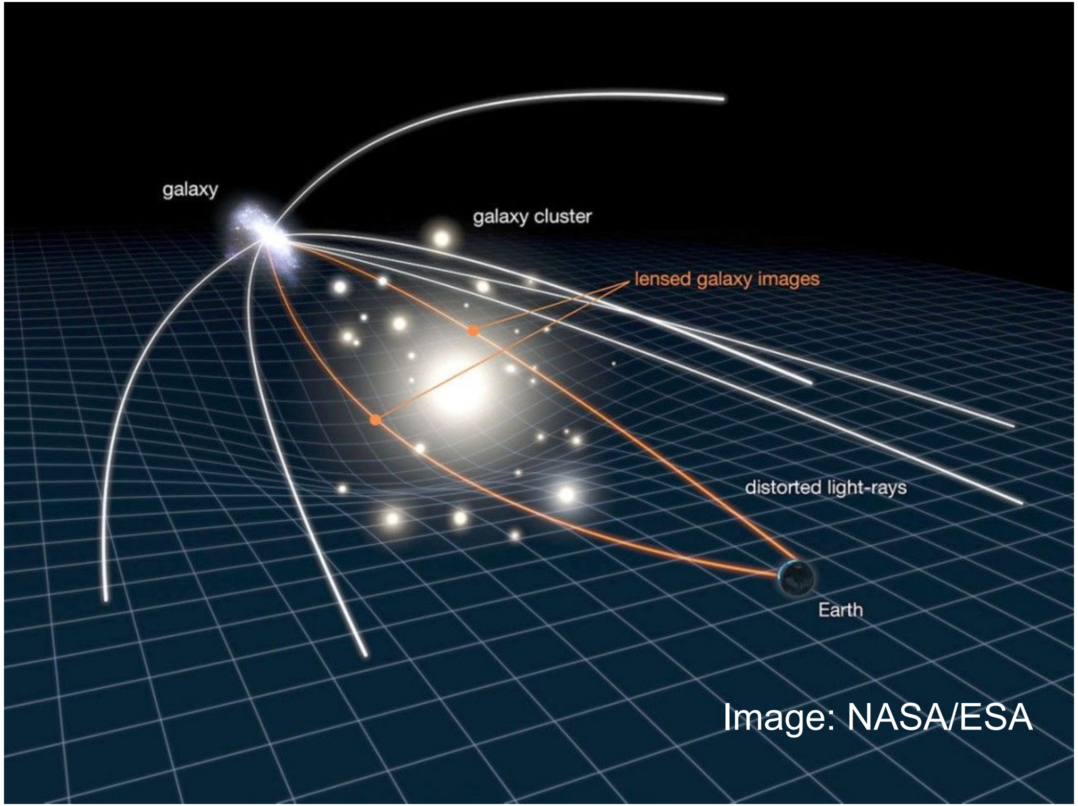
(Image: NASA)

Why study gravitational lenses?

ENERGY DISTRIBUTION OF THE UNIVERSE



Strong Lenses for Dark Matter



Future Predicted Lens Populations

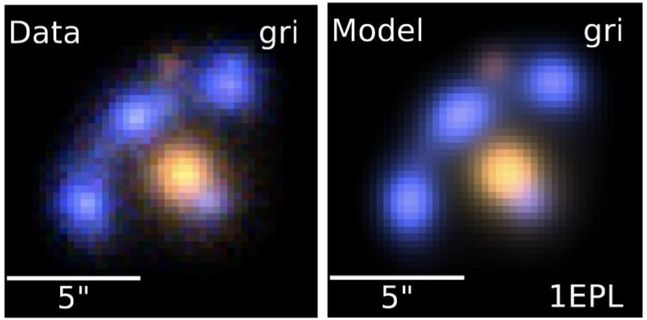
Today	1,000
DES	2,400
LSST	120,000

(Moustakas, 2012, Nord+ 2016, Collett 2015, Oguri & Marshall 2010)

~700 lens candidates from DES so far.
(Diehl et al, 2017, Jacobs 2018, Diehl et al, 2022)

Accurate and precise modeling of lens systems are needed for inferring its underlying astrophysics

Modeling a simple galaxy-galaxy lens system



~6 parameters

Source/Lens Light Model

Lens Mass Model

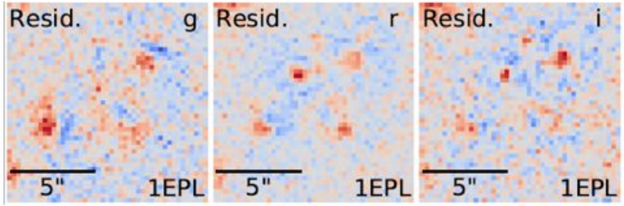
~6 parameters

Solve lens equation in Image plane $\vec{\beta} = \vec{\theta} - \vec{\alpha}(\vec{\theta})$

Convolve with PSF

Construct Model Image

Calculate residuals by subtracting observed image brightness from model image brightness in each pixel and divide by pixel noise.



Calculate Likelihood from sum of residuals

Maximize likelihood using MCMC

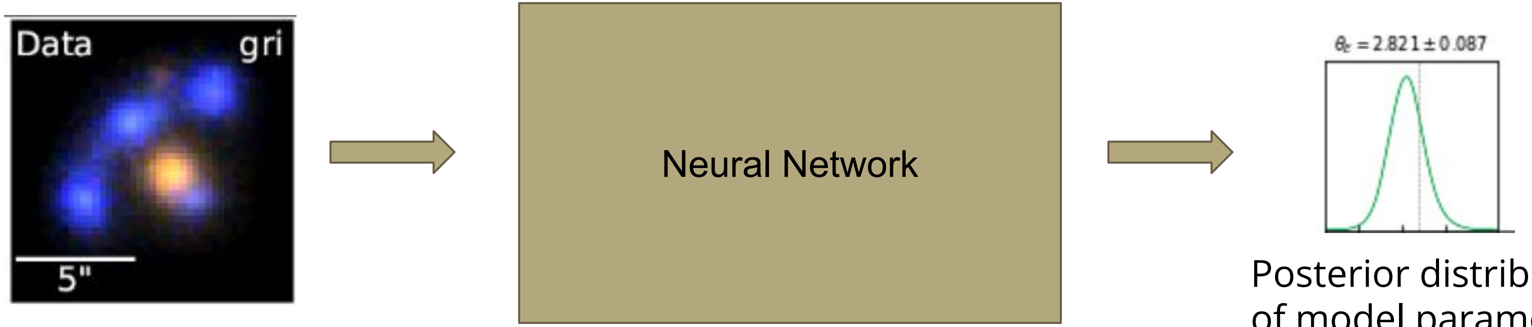
Challenges for conventional lens modeling methods

- Does not easily scale.
- More complex models can have up to hundreds of parameters.
- Computationally expensive, perhaps prohibitively so.

This necessitates the development of **automated** inference methods.

Simulation-Based Inference (SBI)

- Family of methods that use **simulators** to circumvent the need to calculate **explicit likelihoods** for bayesian inference.
- We train a **neural network** to predict the **posterior probability** of the **lens model parameters** we want to infer from **some given data**.



Posterior distribution of model parameters

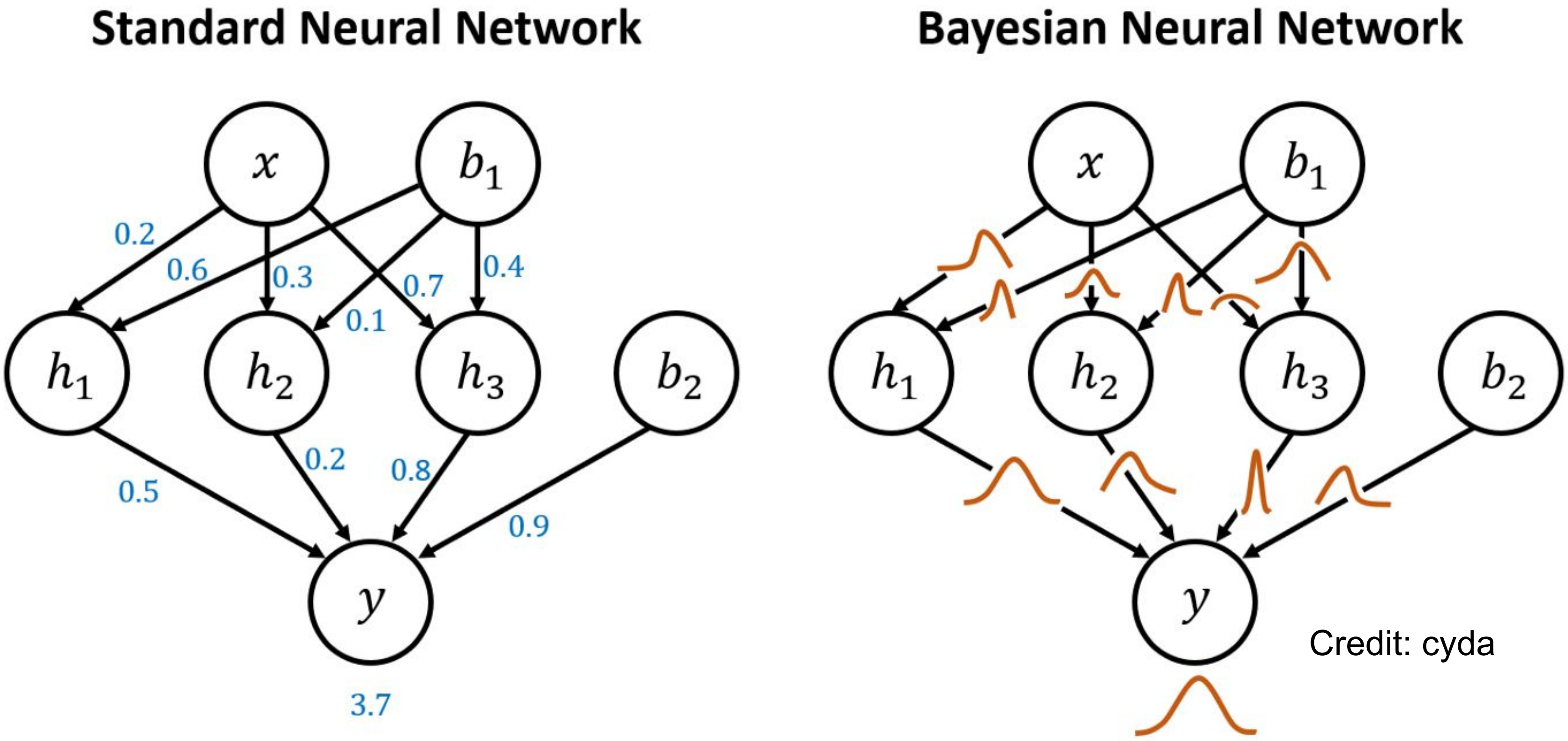
Advantages of SBI over conventional methods.

- Computation is **amortized** (i.e. after the upfront cost of training the density estimator, inference is cheap).
- **Scales** to large datasets (unlike ABC and MCMC where individual inference chains are needed for each observation).
- Posterior probability as model output allow for **interpretable uncertainty estimates**.
- **Robust validation methods** such as simulation-based calibrations (SBC).

This Work

- We use Masked Autoregressive Flows (MAF) (Papamakarios+2017) as the **neural posterior estimator (NPE)** as our SBI algorithm of choice.
    - Mackelab sbi python library (Alvaro Tehero-Cantero+2020)
- We compare our results with a **bayesian neural network (BNN)** trained on the same training dataset.
- BNNs are extensions of standard neural networks where weights and output are treated as probability distributions.

Tangent: Bayesian neural networks (BNN)



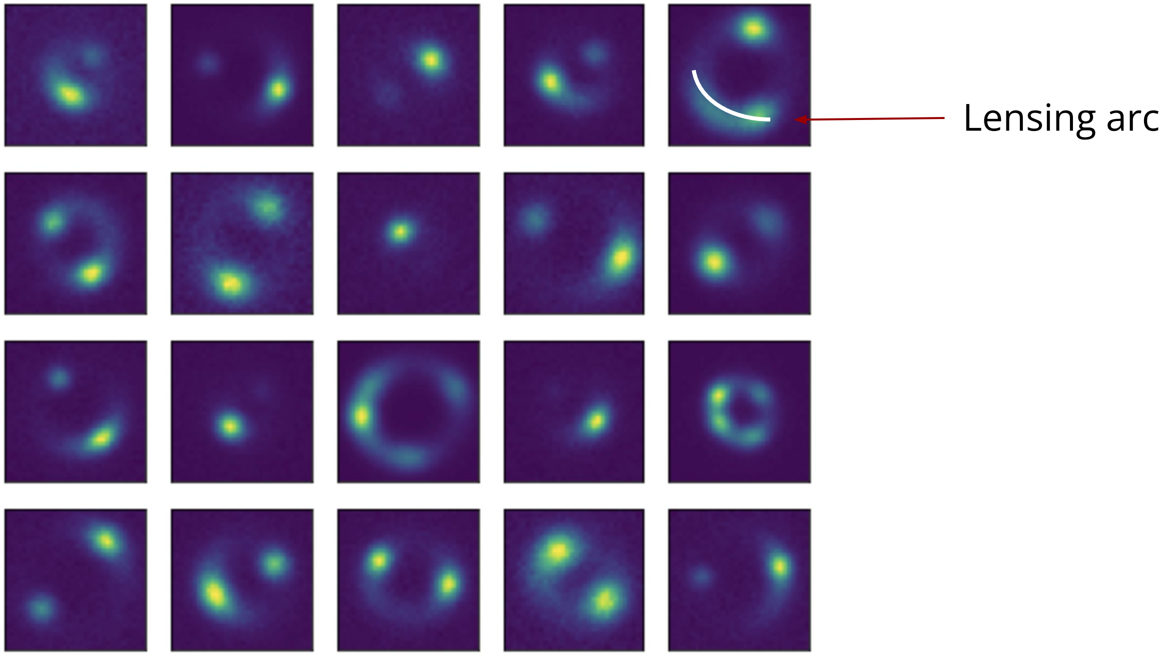
Credit: cyda

Dataset and Models

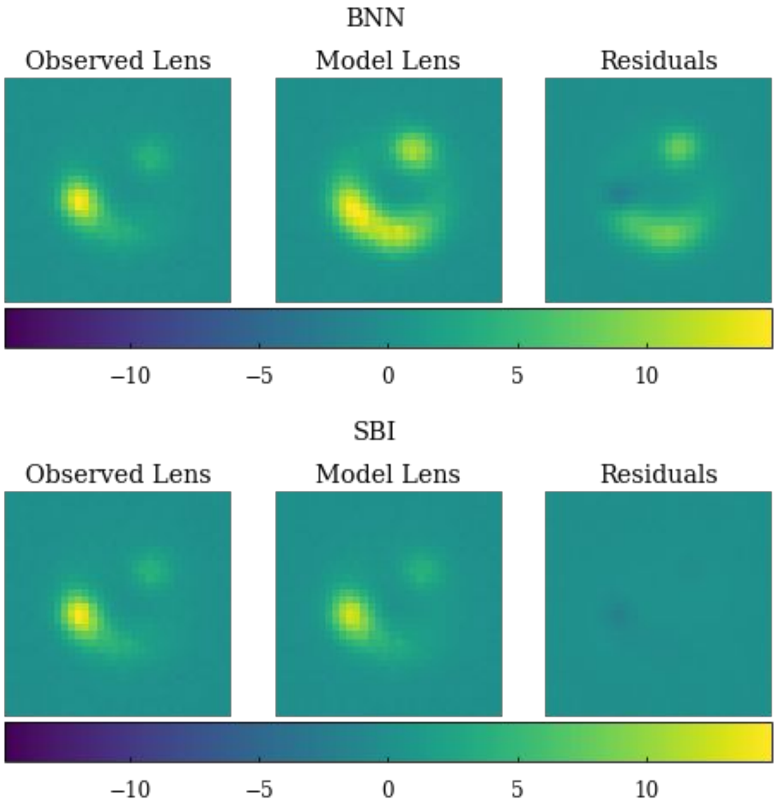
- Training Set: 200k Dark Energy Survey-like single band images with lens light subtracted.
  - deeplenstronomy (Morgan+2021) and lenstronomy (Birrer+2015,2018,2021) libraries
- Data: 1000 images not from the training set, to be modeled.
- Models:
  - **5 parameter singular isothermal ellipsoid (SIE) for the lens mass**
  - *12 parameter - SIE mass model (5) + sersic source light model (5) + external shear (2) (in progress)*

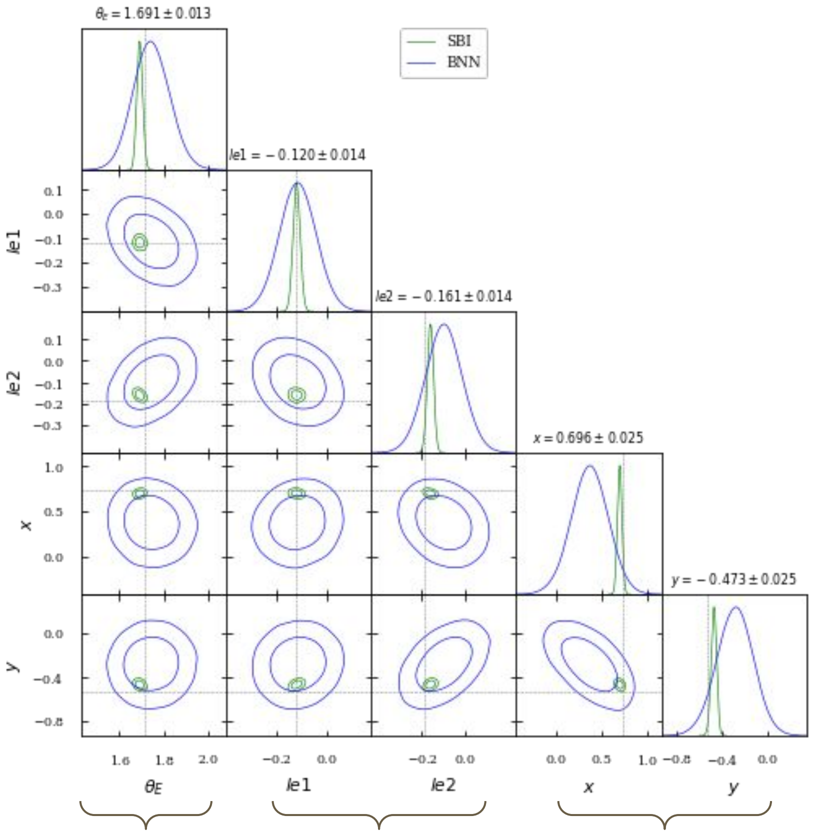
Simulated Images

20 randomly-selected images in the simulated “observed” data



Results - 5-parameter single lens system

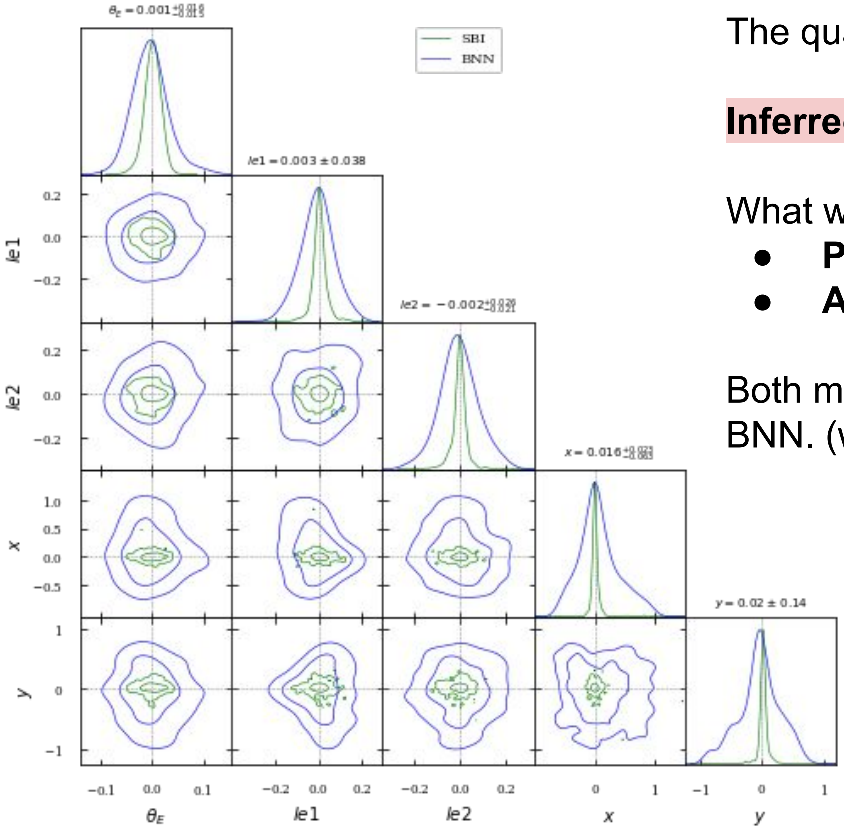




Size

Position

Results - 5-parameter model, 1000 images



The quantities in this corner plot are:

Inferred best-fit value from posteriors - ‘true’ value from sim

What we want**:**

- **Precise**: contours are appropriately small in size
- **Accurate**: contours are centered around origin

Both methods are accurate, but NPE method is more precise than BNN. (we’re trying to understand why)

Conclusion and Work in Progress

- Expand simulation complexity (full 12 parameter mass+light model).
  - Other lens models?
- Continue to optimize performance of our SBI methods.
  - Increase training set size (200k to 1mil)
  - Optimize hyperparameters of neural density estimator.
  - More diagnostics like simulation-based calibration (SBC), coverage probability plots, etc, to increase confidence in the results.

Thanks!

jasonpoh@uchicago

Backup slides

BNN and SBI Architectures

Table A.2. The architecture of the embedding network used in our SBI method. In the parameters column, 'k' denotes the kernel size and 's' denotes the stride, in_ft denotes the input features for the Linear (Dense) layer.

Layer	Output shape	Parameters
Conv2d	[-1, 8, 32, 32]	k=3, s=1
BatchNorm2d	[-1, 8, 32, 32]	k=3, s=1
Conv2d	[-1, 16, 32, 32]	k=3, s=1
BatchNorm2d	[-1, 16, 32, 32]	k=3, s=1
MaxPool2d	[-1, 16, 16, 16]	k=2, s=2
Conv2d	[-1, 32, 16, 16]	k=3, s=1
BatchNorm2d	[-1, 32, 16, 16]	k=3, s=1
Conv2d	[-1, 32, 16, 16]	k=3, s=1
BatchNorm2d	[-1, 32, 16, 16]	k=3, s=1
MaxPool2d	[-1, 32, 8, 8]	k=2, s=2
Conv2d	[-1, 64, 8, 8]	k=3, s=1
BatchNorm2d	[-1, 64, 8, 8]	k=3, s=1
Conv2d	[-1, 128, 8, 8]	k=3, s=1
BatchNorm2d	[-1, 128, 8, 8]	k=3, s=1
MaxPool2d	[-1, 128, 4, 4]	k=2, s=2
Linear	[-1, 24]	in_ft=128*4*4

Table A.3. The Bayesian Neural Network architecture. In the parameters column, 'k' denotes the kernel size and 's' denotes the stride. Conv2dFlipout denotes the 2D convolution layer with Flipout estimator (Wen et al., 2018) and DenseFlipout denotes the Dense layer with Flipout estimator. MultivariateNormalTriL denotes the multivariate normal distribution

Layer	Output shape	Parameters
Conv2dFlipout	[-1, 8, 32, 32]	k=5, s=1
MaxPool2d	[-1, 8, 16, 16]	k=2, s=2
Conv2dFlipout	[-1, 16, 16, 16]	k=3, s=1
Conv2dFlipout	[-1, 16, 16, 16]	k=3, s=1
MaxPool2d	[-1, 16, 8, 8]	k=2, s=2
Flatten	[-1, 1024]	-
DenseFlipout	[-1, 512]	-
Dense	[-1, 20]	-
MultivariateNormalTriL	[-1, 5]	-

Uncertainties

Table 1. True and inferred lens parameter values for the single lens image in Figure 2. T.V. denotes the True Values. The SBI and BNN uncertainties are marginalized 68% posterior uncertainties.

Parameter	T.V.	SBI	BNN
θ_{ein}('')	1.712	1.691 ± 0.013	1.74 ± 0.08
le_2	-0.125	-0.120 ± 0.014	-1.16 ± 0.07
le_2	-0.185	-0.161 ± 0.014	-0.10 ± 0.08
x ('')	0.731	0.696 ± 0.025	0.37 ± 0.20
y ('')	-0.530	-0.473 ± 0.025	-0.29 ± 0.16

Table A.4. Average 68% scatter in the difference between the best-fit lens parameters from for 1000 lenses for SBI and BNN lens parameters

Parameter	SBI	BNN
θ_{ein}('')	$0.001^{+0.016}_{-0.015}$	$-0.003^{+0.030}_{-0.037}$
le_1	0.003 ± 0.038	-0.006 ± 0.081
le_2	$-0.002^{+0.026}_{-0.021}$	$-0.003^{+0.078}_{-0.087}$
x ('')	$0.016^{+0.027}_{-0.040}$	$0.07^{+0.27}_{-0.40}$
y ('')	0.02 ± 0.14	$-0.04^{+0.41}_{-0.28}$

Strong Lenses for Dark Energy - Time Delay Lenses

Time delay

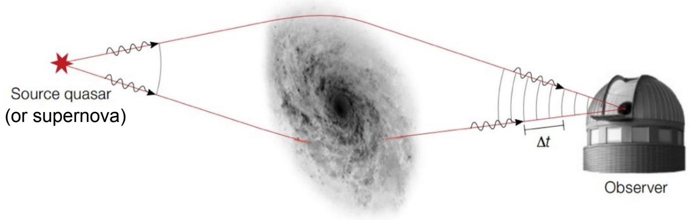
Geometric time-delay

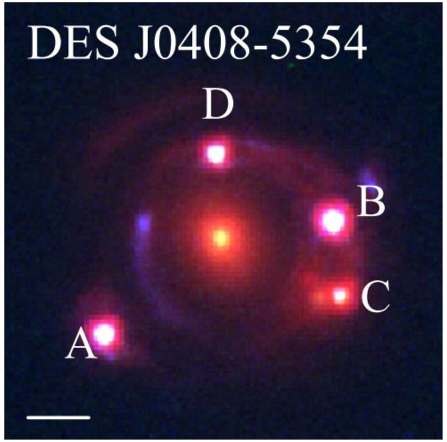
Lens potential

$$\Delta\tau = \frac{(1+z_L)}{c}\mathcal{D}\left[\frac{1}{2}\left|\boldsymbol{\theta}-\boldsymbol{\beta}\right|^2 - \phi\right]$$

$$\mathcal{D} \equiv \frac{D_L D_S}{D_{LS}}$$

Time-delay distance





(Shajib et al. 2018)

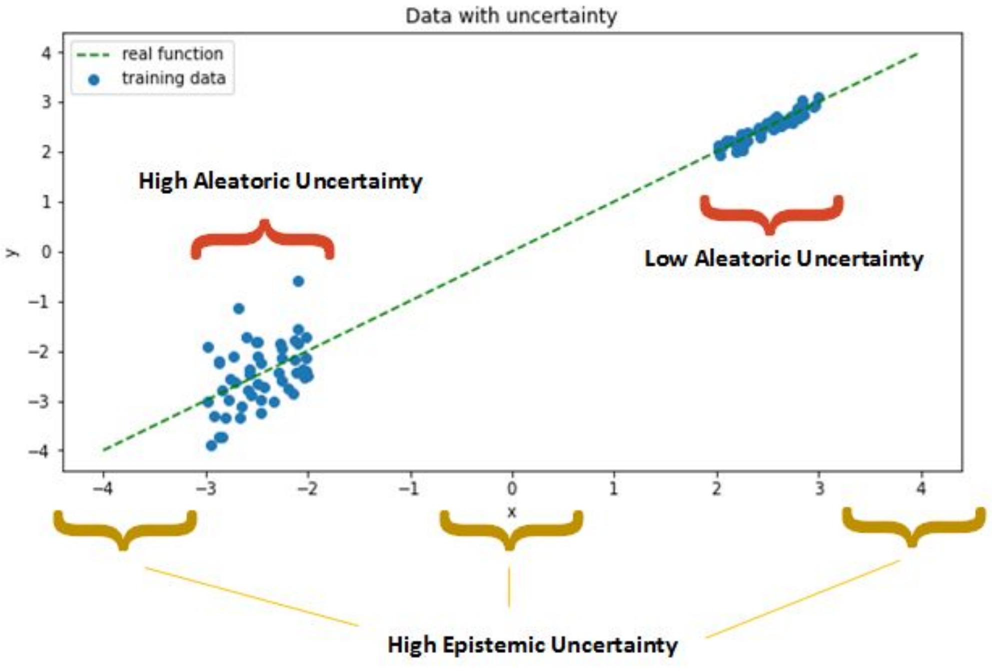
Uncertainties

Aleatoric: Error in the **data**

- Modeled by allowing output to be a distribution

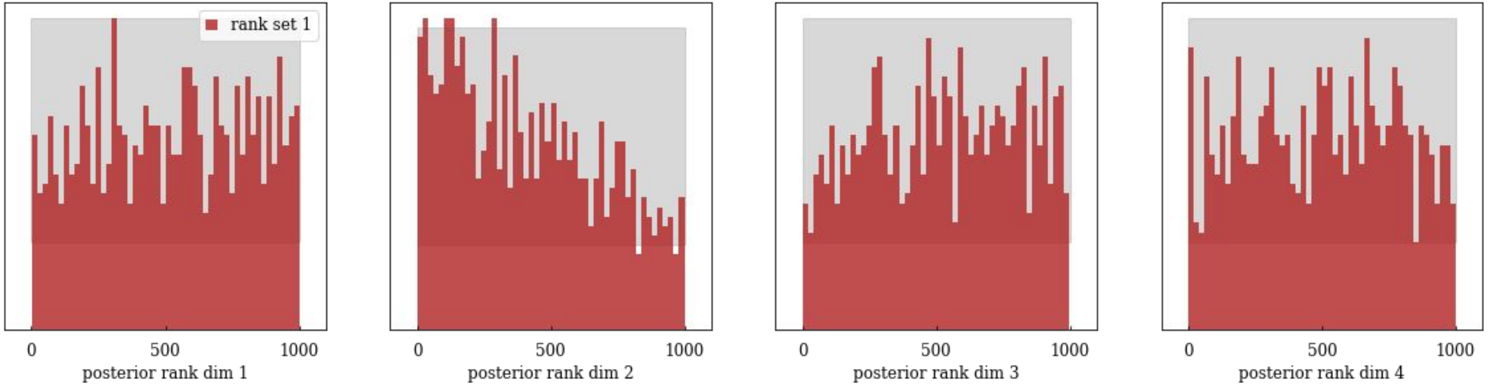
Epistemic: Error in the **model**

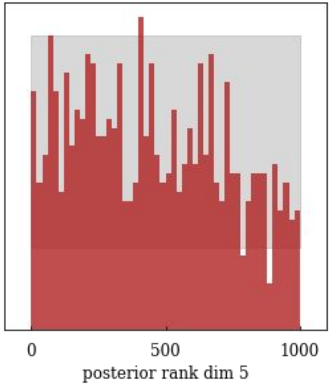
- Modeled by allowing weights be a distribution.

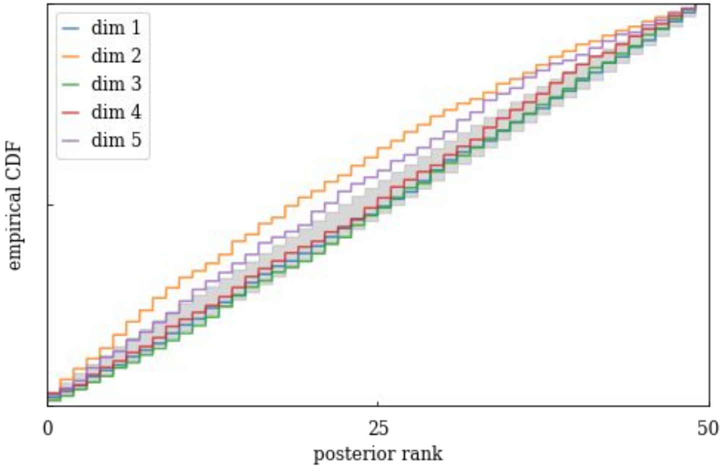


Credit: CampusAI

Simulation-based Calibrations








```
kolmogorov-smirnov p-values
check stats['ks pvals'] =
[2.2129539e-01 1.2099531e-13
2.3344547e-02 2.8580514e-01
3.7295149e-06]
c2st accuracies
check stats['c2st ranks'] = [0.572
0.5825 0.5975 0.585  0.589 ]
- c2st accuracies
check stats['c2st dap'] = [0.4945
0.522  0.5205 0.465  0.499 ]
```