Simulation-based Inference
Cosmology from Strong Lensing

Becky Nevin and DeepSkies Lab

December, 2023
COFI AI/ML Winter School

Bayesian Inference

Given an observation x, we want to infer the parameters θ of the model

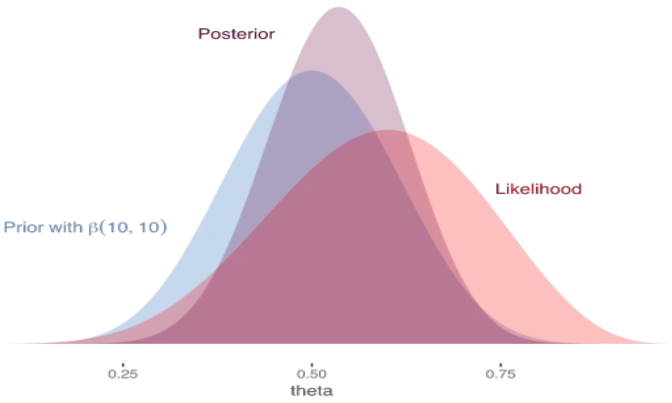
$$x \sim \text{Model}\ (\theta)$$

Likelihood

Prior

Posterior

$$P(\theta|x) = \frac{P(x|\theta)\ P(\theta)}{\int d\theta'\ P(x|\theta')\ P(\theta')}$$

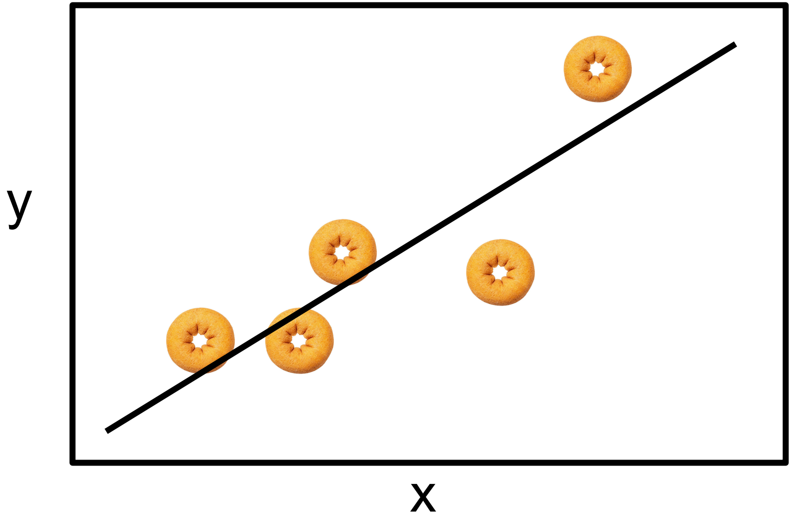


(Hidden slide) Idea

Reframe previous equation as whatever / p(x) and then expand on p(x) and likelihood and talk about how they can be intractable under certain conditions

Simple example

Task: predict parameters
ϑ::m,b given data::x,y



$y = mx + b$

$y = f(x) + \varepsilon$

Given these data,
what is p(theta|x)?

Bayesian Inference

- The model / simulator can be complicated with high dimensional parameters and latent variables

- Intractable Likelihood:

$$P(x \mid \theta) = \int dz \; P(x, z \mid \theta)$$

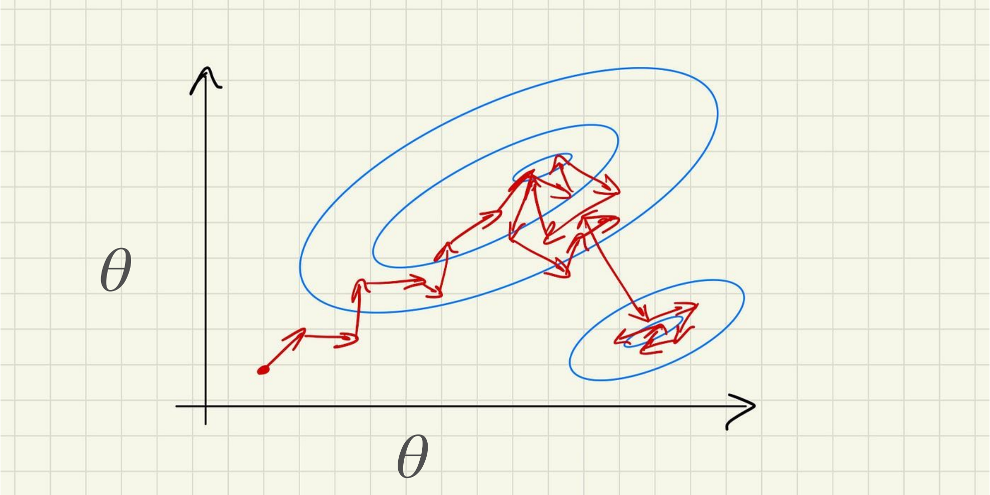
Quick MCMC, not so fast

$y = mx + b$

$y = f(x) + \varepsilon$

Likelihood (under a Gaussian assumption):

$p(x|\theta)$ = Normal(evaluated at each theta)

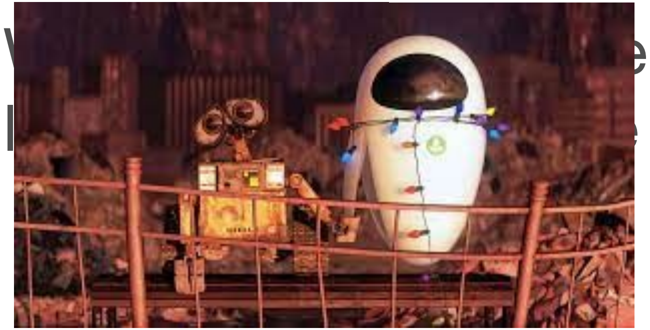


Given these data, what is $p(\theta|x)$?

(Hidden slide) What does it mean for evidence or likelihood to be intractable?

(Hidden slide) Introduction to Bayesian computa

Marriage of Bayesian inference and computation



W… e was a way to do this without the
l… e access to an efficient simulation…

Bayesian Inference

- The model / simulator can be complicated with high dimensional parameters and latent variables

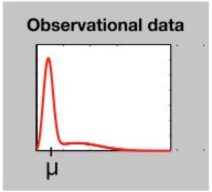
- Intractable Likelihood

$$P(x \mid \theta) = \int dz \, P(x, z \mid \theta)$$

Simulation Based Inference (SBI)

- Statistical Inference in the case of intractable likelihood (Likelihood Free Inference)

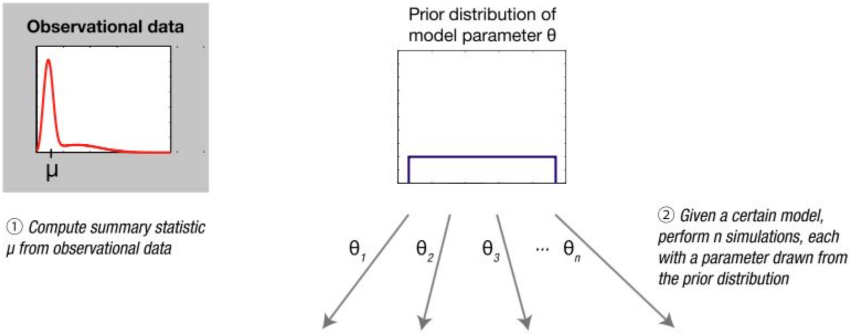
Approximate Bayesian Computation



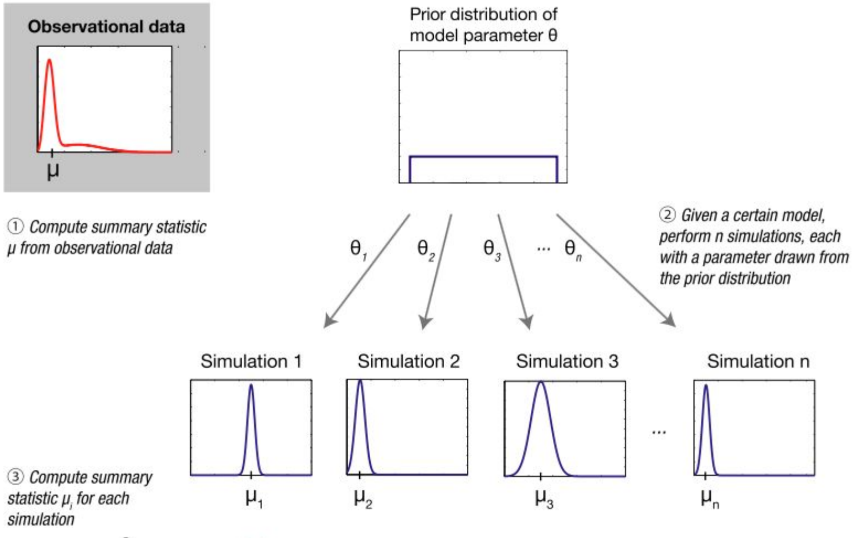
① *Compute summary statistic μ from observational data*

[Miles Cranmer Blog]

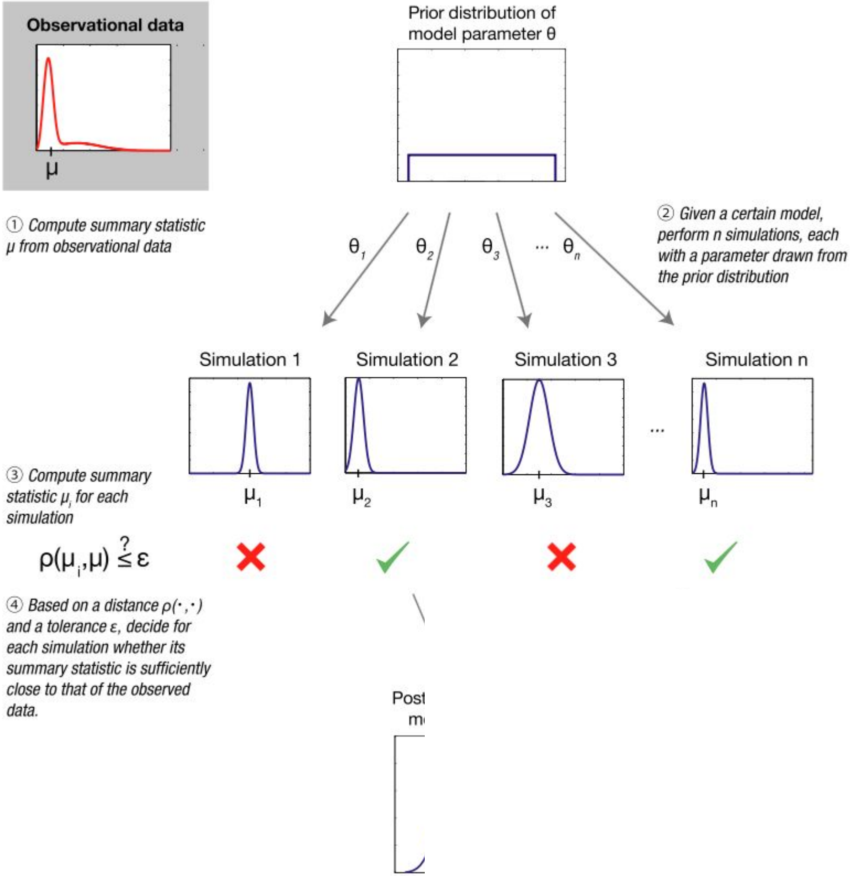
Approximate Bayesian Computation



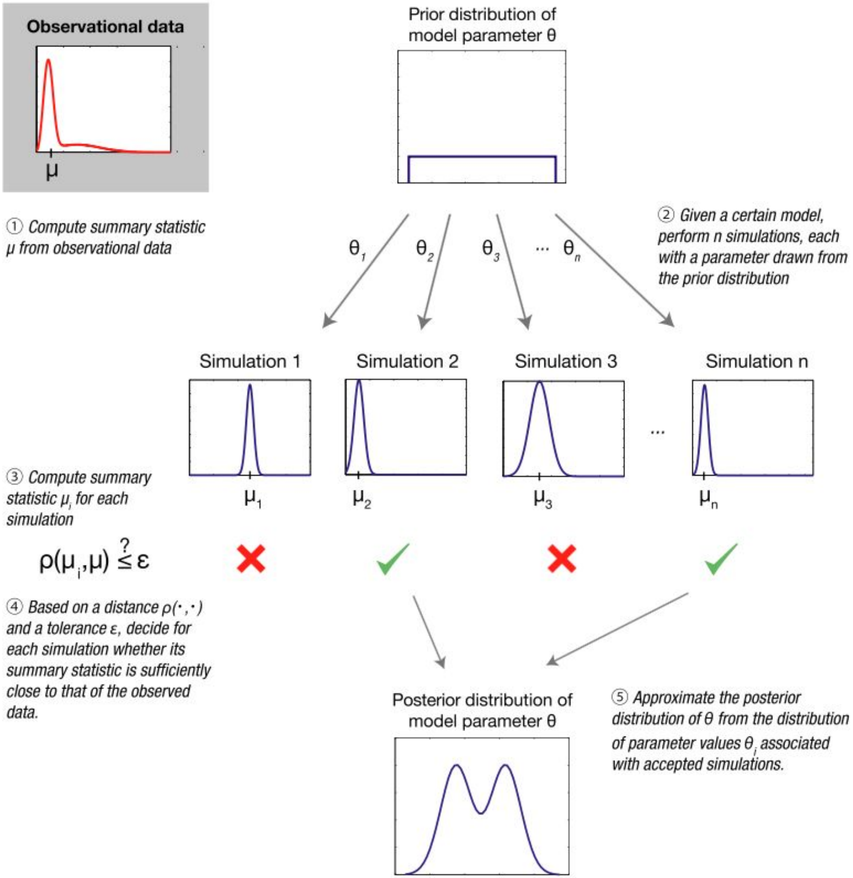
Approximate Bayesian Computation



Approximate Bayesian Computation



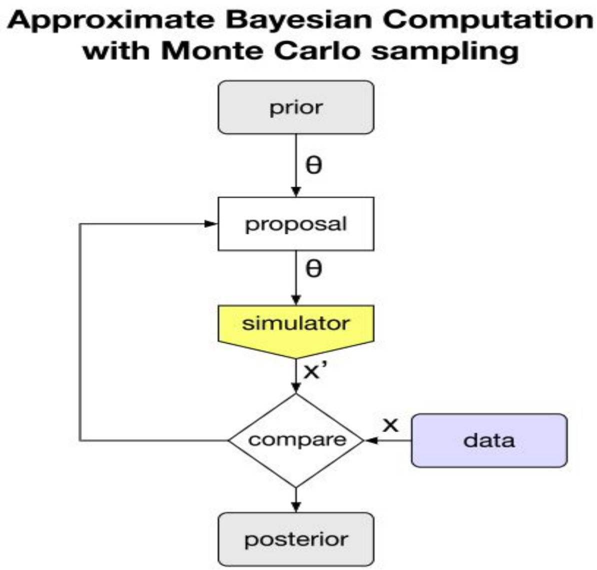
Approximate Bayesian Computation



Approximate Bayesian Computation

Challenges:

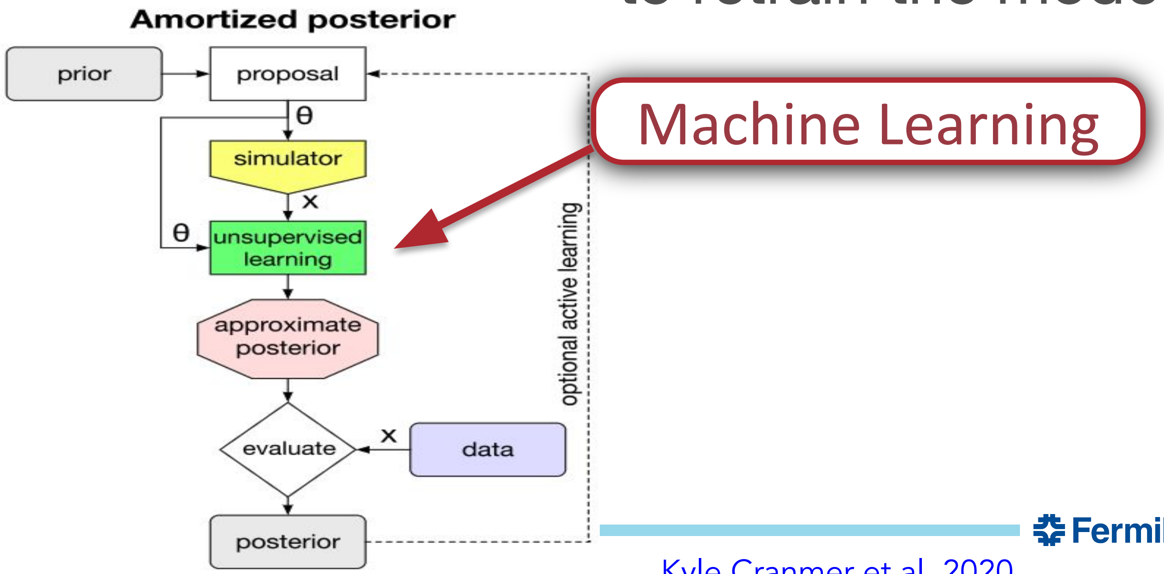
- Many simulations needed
- Sampling efficiency (Monte Carlo) scales poorly with high dimensional data
- Inference for new observations requires running the entire inference



[Kyle Cranmer et al. 2020]

SBI using Machine Learning

- Computationally expensive upfront simulation to get
- Use ML to learn the posterior, likelihood, or likelihood ratio
- Can evaluate new data without having to retrain the model!



Kyle Cranmer et al. 2020

Neural Density Estimators

- An estimate of the exact posterior $p(\theta|x)$ can be learnt from neural density estimators $q_\phi(\theta|x)$
- Maximize the averag

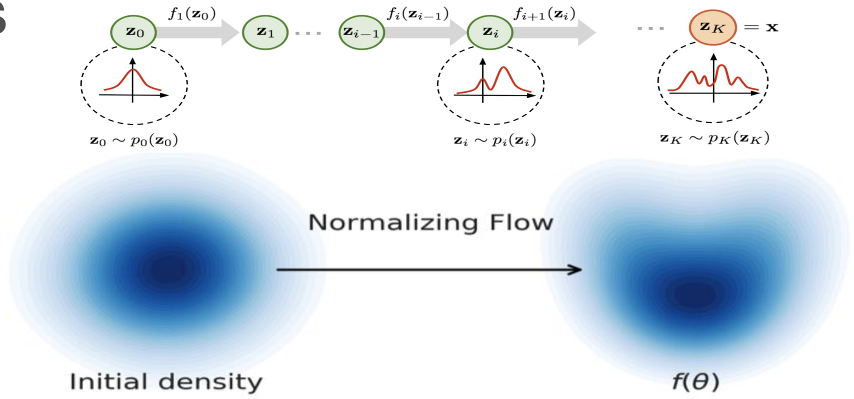
$$\frac{1}{N}\sum_n log\ q_\phi(\theta_n|x_n)$$

Neural Density Estimators

- An estimate of the exact posterior can be learnt from neural density estimators
- Maximize the averag

$$\frac{1}{N}\sum_{n} log\ q_{\phi}(\theta_n | x_n)$$

- Conditional density estimators are flexible with invertible transformations

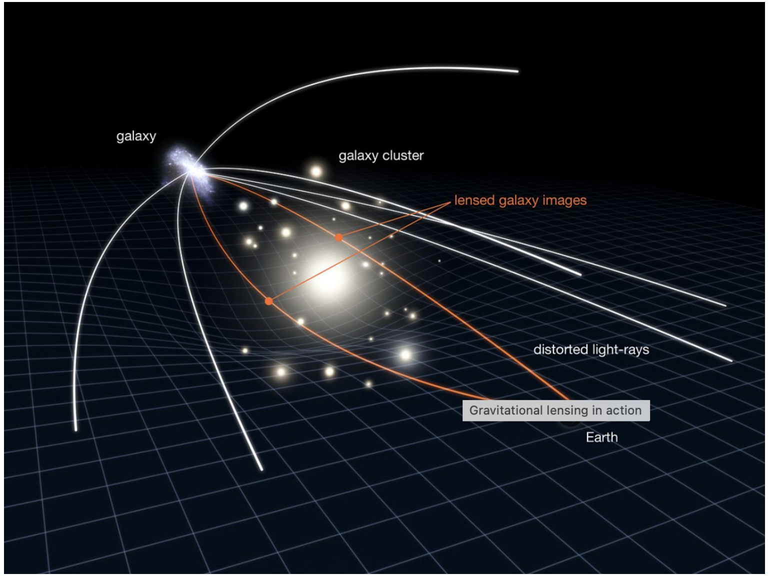


Cosmology from Strong Gravitational Lensing using SBI

- Cosmology constraints from simulated lensed images

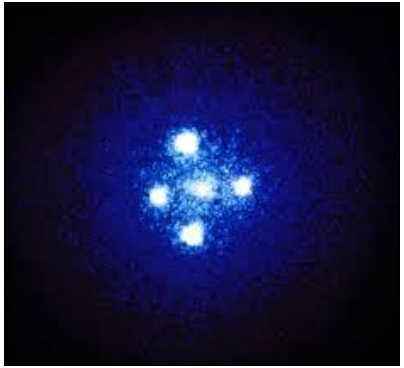
Cosmology from Strong Gravitational Lensing using SBI

- Cosmology constraints from simulated lensed images
- Strong Lensing
    - Lensed by massive elliptical galaxy or galaxy cluster



Cosmology from Strong Gravitational Lensing using SBI

- Cosmology constraints from simulated lensed images
- Strong Lensing
  - Lensed by massive elliptical galaxy or galaxy cluster
  - Arcs and multiple images



Cosmology from static strong lenses

- Constrain on matter density Ω_m and dark energy equation of state parameter w through distance ratio
- Cosmology independent of Hubble constant H_0

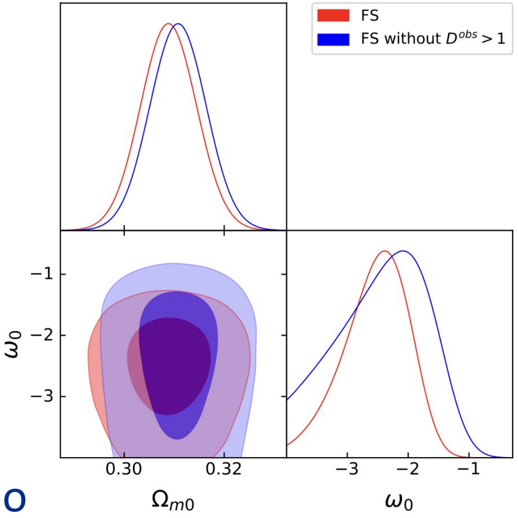
$$D = f(H_0, \Omega_m, w)$$

Velocity dispersion

$$\theta_E = 4\pi \frac{\sigma^2_{SIS}}{c^2} \frac{D_{ls}}{D_s}$$

Einstein radius

Distance ratio



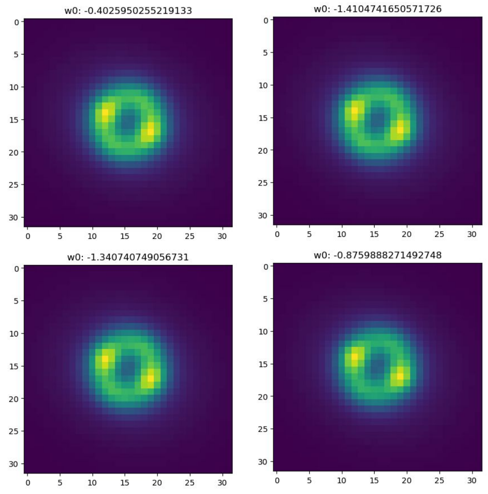
Amante et al. 2020

SBI setup

- Python package :

https://www.mackelab.org/sbi/

- Inputs to the SBI package
  - Simulated strong lens images : Deeplenstronomy
  - Priors on the parameters



- Masked Autoregressive Flows [Papamakarios et al. 2018] as Neural Posterior Estimator

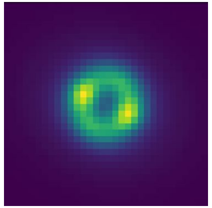
- Keeping all parameters except fixed

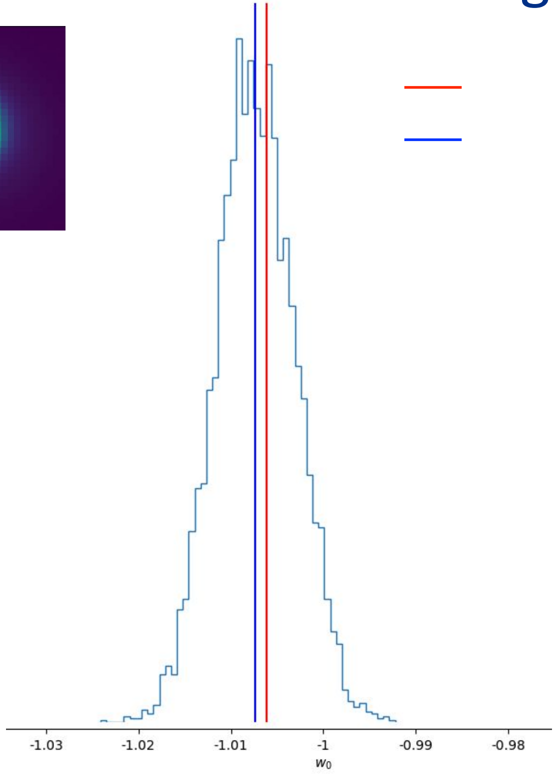
- Train on 100k images

Fermilab

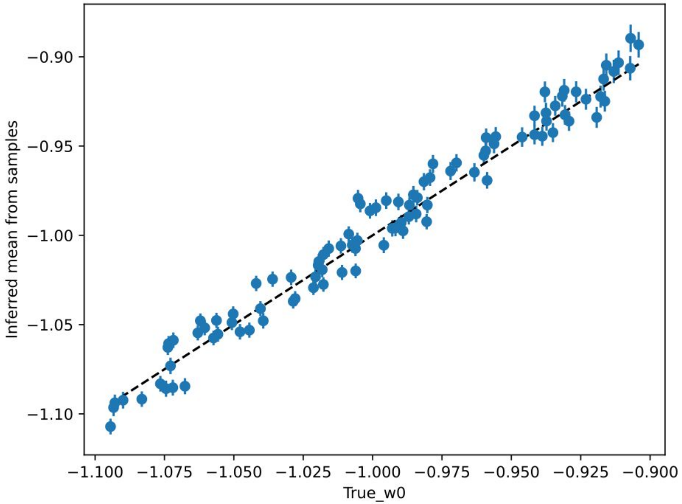
Dark Energy equation of state parameter

Inference from a single lens



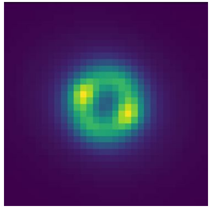


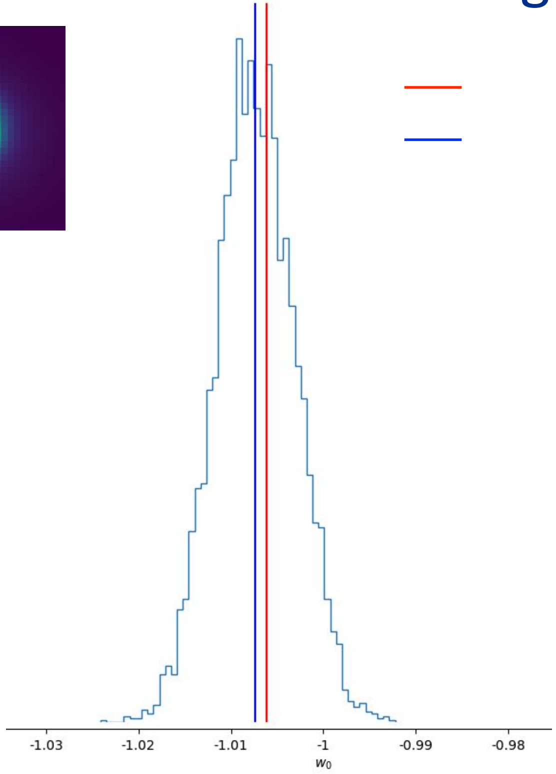
Inference from 100 lens



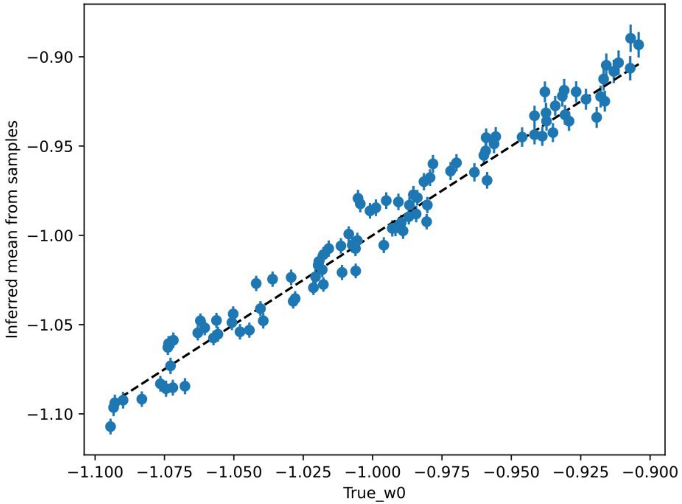
Dark Energy equation of state parameter

Inference from a single lens





Inference from 100 lens



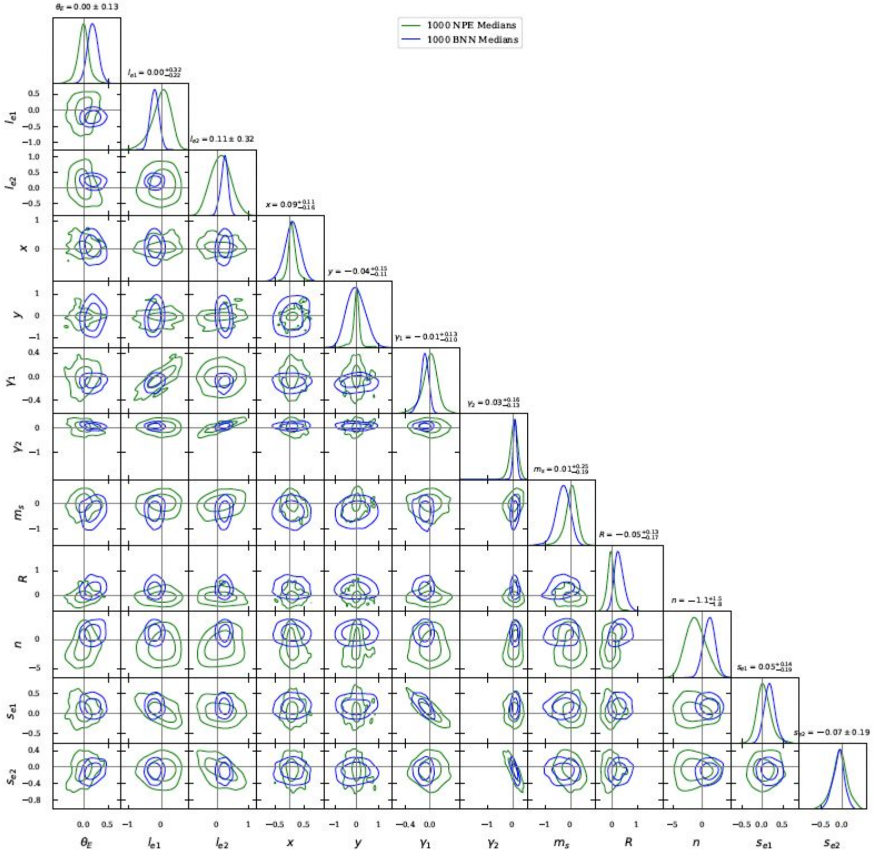
Next steps:

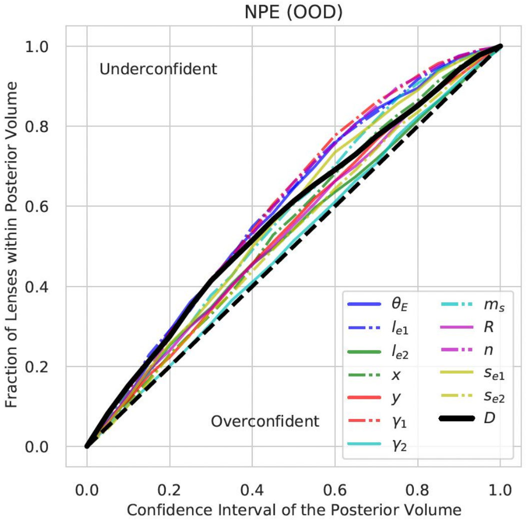
- astrophysics + cosmology
- Joint inference

Fermilab

SBI to infer astrophysics parameters from Strong Lensing

Jason Poh et al. and DeepSkies





SBI to infer galaxy properties from spectra

Khullar et al. 2022 and DeepSkies

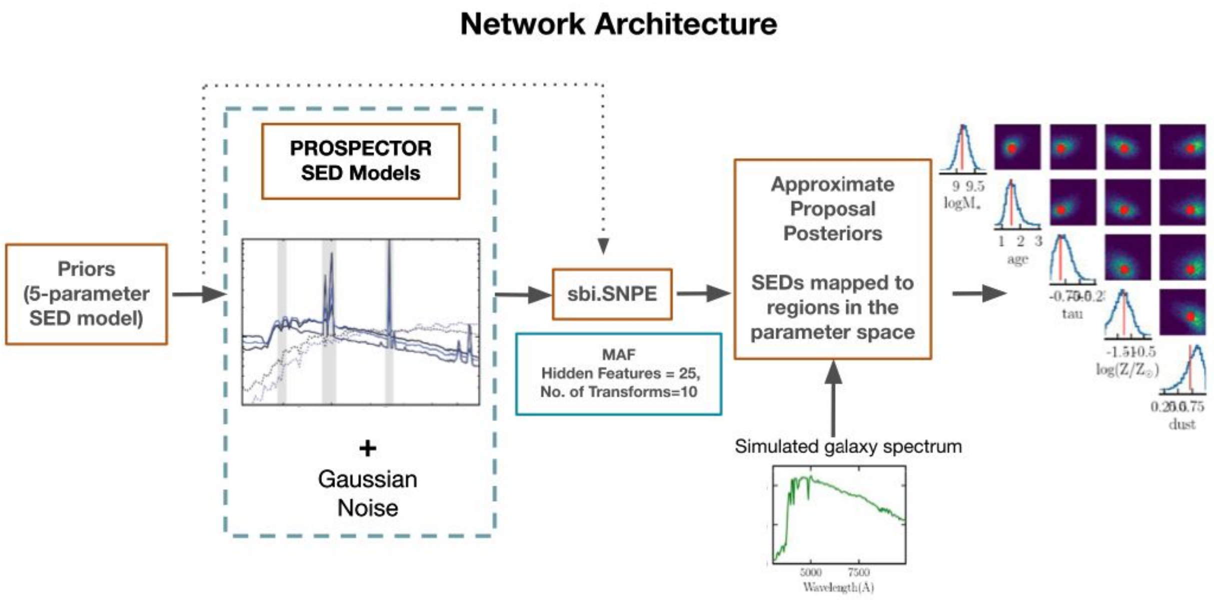


Figure 2. The architecture used in this work to infer galaxy SED properties with spectroscopic data. We use a five-parameter model and a training set with realistic spectra, that is trained by an NPE to generate approximate posteriors.

Tutorial

- Go to colab link

So where do we go from here?

- Hierarchical paradigm: retrieve population and astrophysics parameters

- Other formats such as graph neural networks

- The question of uncertainty

Summary

- Simulation Based Inference
  - Intractable likelihood
  - Machine learning methods are amortized i.e. can evaluate posterior from new data without retraining the model
  - Computationally efficient than MCMC based methods
- Can constrain astrophysics and cosmology parameters from strong lens images using SBI

EXTRAS

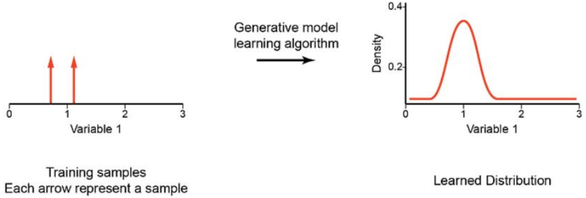
Density Estimators (Thanks to Jason!)

- Unsupervised method of getting structure from data:
  - Given , what is ?
  - Given , what is ?
- Gaussian Density Model

Model: $$q_{\phi}(\mathbf{x}) = \frac{1}{|\det(2\pi\boldsymbol{\Sigma})|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \quad \text{where} \quad \boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}.$$

Training: Parametric density models are typically estimated by maximum likelihood. Given a set of training datapoints $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ that have been independently and identically generated by a process with density $p(\mathbf{x})$, we seek a setting of the model's parameters $\boldsymbol{\phi}$ that maximize the average log likelihood on the training data:

$$L(\boldsymbol{\phi}) = \frac{1}{N}\sum_n \log q_{\boldsymbol{\phi}}(\mathbf{x}_n). \tag{2.13}$$



From Papamakarios (2019)

Neural Density Estimators

- Neural Networks to parameterize density model

$$L(\phi) = \frac{1}{N}\sum_n \log q_\phi(\mathbf{x}_n) = \frac{1}{N}\sum_n f_\phi(\mathbf{x}_n).$$

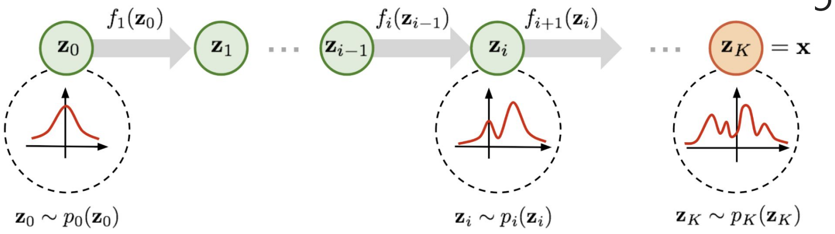
- The parameters of the Neural network are updated through gradient descent

$$\nabla_\phi \hat{L}(\phi) = \frac{1}{M}\sum_m \nabla_\phi f_\phi(\mathbf{x}_{n_m}).$$

- SBI has 4 built-in density estimators:
  - Masked Autoregressive Flow (MAF)
  - Neural Spline Flow (NSF)
  - Masked Autoencoder for Distribution Estimation (MADE)
  - Mixture Density Network (MDN)

MAF

- Normalizing Flow of Autoregressive Models
- Normalizing flow
  - base density to target density though invertible transformation



- Autoregressive models
  - Decompose Target de nditionals and models of conditionals

$$p(\boldsymbol{x}) = p(x_1, x_2, \ldots, x_n)$$

$$p(\boldsymbol{x}) = p(x_1)p(x_2) \ldots p(x_n)$$

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i)$$

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1})$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels