DATA-DRIVEN STRONG GRAVITATIONAL LENSING ANALYSIS IN THE ERA OF LARGE SKY SURVEYS

Laurence Perreault-Levasseur
**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.

- Distance ~ 5 Billion light years
- Distance ~ 12 Billion light years
Two images of a background quasar

Lensing galaxy
SCIENCE MOTIVATIONS FOR STRONG LENSING

1 - 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.

![Image of lensing phenomenon](image)
Matter power spectrum

- CMB/LSS
- Clusters
- Galactic
- Satellite galaxies/substructures

DM constrained to be CDM

Small Scales: unconstrained

$\Delta^2(k)$

$10^{18} \quad 10^{16} \quad 10^{14} \quad 10^{12} \quad 10^{10} \quad 10^{8} \quad 10^{6} \quad 10^{4}$

$k \quad [\text{Mpc}^{-1}]$

$10^{-2} \quad 10^{-1} \quad 10^{0} \quad 10^{1} \quad 10^{2}$

Ferreira et al., A&A Review, 2021
2 - 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

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

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

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

\[ y = L(p)x + n \]

Data

Source Parameters (linear)

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.
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

UNDISTORTED IMAGE OF THE BACKGROUND SOURCE

delensed image of background source?
UNDISTORTED IMAGE OF THE BACKGROUND SOURCE

\[ S = \left( L^T C_N^{-1} L + C_p^{-1} \right)^{-1} L^T C_N^{-1} D \]
UNDISTORTED IMAGE OF THE BACKGROUND SOURCE WITH THE RECURRENT INFEERENCE MACHINE (RIM)
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
orientation angle

ellipticity
SIMULATED GALAXIES GENERATED WITH A VARIATIONAL AUTOENCODER

COSMOS

VAE

TNG100

VAE

Alexandre Adam
TRAINING ON HYDRODYNAMICAL SIMULATIONS

**Ground Truth**
- Background
- Foreground
- Lensed Image

**Prediction**
- Background
- Foreground
- Lensed Image

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

Uncertainty estimation with approximate Bayesian neural networks

Perreault-Levasseur et al., 2017
Morningstar et al., 2018
Variational Inference to Approximate Bayesian Neural Networks

Pros:
- Amortized.
- Requires few hundred forward passes at evaluation time (to collect samples). Still very fast.
- Marginalizes implicitly over parameters we do not wish to explicitly model.
- With good coverage probabilities, one can use importance sampling of the output distribution to get an unbiased posterior. (Provided one can actually write this posterior)

Caveats:
- The variational distributions (Bernoulli) are extremely simplistic, therefore even if we attempt to use them to approximate the true weight distributions, that approximation could be bad and yield inaccurate uncertainties.

Same problem remains regardless of the variational distribution used: there is no way of quantifying how well we approximate the true weight distributions.
UNCERTAINTY ESTIMATION WITH SIMULATION-BASED INFEERENCE METHODS

An example of the inference of the posterior of foreground variables

Coverage probabilities

Hierarchical Bayesian inference

We are interested in the parameters of the hyper distribution, the posterior of individual measurements.
Hierarchical Bayesian inference

We are interested in the parameters of the hyper distribution,

Latent variable
Hierarchical Bayesian inference

![Hierarchical Bayesian inference diagram](image)

Ronan Legin
Connor Stone

Legin, Stone, Hezaveh, Perreault Levasseur,
ICML 2022 - ML4Astro Workshop arXiv:2207.04123
**UNCERTAINTY ESTIMATION WITH SIMULATION-BASED INFERENCE METHODS**

**Pros:**
- Use the **power of ML to find a compress statics**, and even if it is biased, we can get unbiased error estimate, the only drawback would be sub-optimal precision. (Provided the simulation pipeline is accurate!)
- A well-defined statistical framework that can: be relatively **fast**, deal with **complex distributions**, model **joint posteriors**.
- Use a neural density estimator to get the joint distribution $p(\text{data, parameter})$, no need for the epsilon parameter in ABC.
- Can **change the prior** from data point to data point without retraining the ML compressor.
- Once we have the posterior, can **generate samples** that are consistent with data (this is really important for ‘interrogating the black box’)

**Caveats:**
- Hard to marginalize implicitly over parameters, we need to **explicitly** model them.
- We don’t model the uncertainty of the density estimator itself. (But it’s a fairly simple ML model, and except for very pathological problems it’s reasonable to expect that we are in interpolation mode).
- Limited to **low-dimensional posteriors** (10s maximum).
- Requires an **accurate simulation pipeline**.
Neural Ratio Estimators

\[ P(x, \theta) \]

Class #1
\[ \{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_N, \theta_N)\} \]

Class #2
\[ \{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_M, \theta_M)\} \]

\[ r(x, \theta) = \frac{p(x, \theta)}{p(x)p(\theta)} = \frac{p(\theta | x)}{p(\theta)} \]
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**

\[ P(\theta, x) \]

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

Class #1
\[ \{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_N, \theta_N)\} \]

Class #2
\[ \{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_M, \theta_M)\} \]
H0 INFEERENCE WITH NEURAL RATIO ESTIMATORS

Campeau-Poirier et al. ICML 2023 ML4Astro Workshop, arXiv:2309.16063
H0 Inference with Neural Ratio Estimators

Ève Campeau-Poirier

Fraction of truths within the highest probability density region

Underconfident

Overconfident

\( p(H_0 | \Delta t, \Delta \phi) \)

- 3,000
- 1,500
- 500
- 50
- True value

H_0 (km Mpc\(^{-1}\) s\(^{-1}\))
Estimating the dark matter particle temperature with Neural Ratio Estimators

Anau Montel, Coogan et al. 2022, arXiv:2205.09126
Coogan et al., NeurIPS 2020 ML4PS Workshop
ESTIMATING THE SENSITIVITY OF LSST TO THE BREAK AND SLOPE OF THE DARK MATTER MASS FUNCTION

Andreas Filipp
TESTING FOR ROBUSTNESS TO MODEL MISSPECIFICATION

Training on SIS (spherically symmetric lenses)...

And testing on galaxies with slight ellipticities, $q=0.9\%$

Andreas Filipp
**Ratio estimation methods**

**Pros:**
- Can marginalize implicitly over large number of nuisance parameters

**Caveats:**
- Because we have marginalized, we’ve lost the capability to generate samples consistent with the observations.
- From experiments, it seems easy to find examples where the NN is very brittle and sensitive to model misspecification.
- So far: no real way of quantifying the uncertainty of the ratio estimator itself. All the guarantees are in terms of convergence to a specific ratio in the limit of perfect training. Is this always realistic?
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)
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?

\{x_1, \ldots, x_n\} \overset{iid}{\sim} \pi_{data}(x)

\begin{align*}
p_\theta(x) &\approx \pi_{data}(x)
\end{align*}
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

$$s(x) = \nabla_x \log(\pi(x))$$

Training data

$$\{x_1, \ldots, x_n\} \overset{iid}{\sim} \pi_{data}(x)$$

Score function

$$s_\theta(x) \approx \nabla \log(\pi(x))$$
Score-Based Modeling

We model the score of the prior

\[ s_\theta(x) \equiv \nabla_x \log p_\theta(x) \]

\[ \mathcal{N}(0, \sigma_{\text{max}}^2) \quad \text{Reverse-time SDE} \quad \text{Target distribution} \]
**Score-based Modeling**

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

\[ \nabla_x \log p(x \mid y) \]

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

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

$$\nabla_x \log p(x | y) = \nabla_x \log p(y | x) + \nabla_x \log p_\theta(x)$$
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.
Coverage Test for Accuracy

Simulation 1 ($x_1^* \sim p(\cdot | \theta_1^*)$)
- $\theta \sim \tilde{p}(\theta | x^*)$
- $\theta^*$
- $\theta_r$

30% in ball

Simulation 2 ($x_2^* \sim p(\cdot | \theta_2^*)$)

40.6% in ball

Simulation 3 ($x_3^* \sim p(\cdot | \theta_3^*)$)

33.6% in ball

... Simulation $N$ ($x_N^* \sim p(\cdot | \theta_N^*)$)

92.6% in ball

pip install tarp
COVERAGE TEST FOR ACCURACY WITH RANDOM POINTS (TARP)

Step 1: Generate Simulations

Step 2: Estimate Credibility Level

Step 3: Estimate Coverage Probability

Step 4: Repeat over multiple truth in test set

Step 5: Calculate and plot expected coverage probability curve
COVERAGE TEST FOR ACCURACY
DEALING WITH REALISTIC NOISE: BEYOND GAUSSIANITY

Alexandre Adam
Ronan Legin

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:

\[ P(x|\eta) = Q(x - M(\eta)) \]
Dealing With Realistic Noise: Beyond Gaussianity

SLIC: Score-based Likelihood Characterization

\[ s(x_0) = \partial \log Q(x_0) / \partial x \]

\[ P(x_0 | \eta) = Q(x_0 - M(\eta)) \]

\[ \eta_{i+1} = \eta_i + \tau \nabla_x \log Q(x_0 - M(\eta)) \nabla_\eta M(\eta_i) + \sqrt{2\tau} \xi \]

PSF-deconvolution

\[ \text{data} = \text{PSF} \left( \begin{array}{c} x \\ \end{array} \right) + n \]
PSF-DECONVOLUTION (FOR HST)

Observations (y), \( HST \ ACS/F814W \)
**PSF-DECONVOLUTION (FOR HST)**

Observation (x4)

**HST WFC3IR/F160W**

VP + SKIRT

Posterior sample

Posterior sample

Median

IQR 84%-16%

VP + PROBES

VP + PROBES

VE + PROBES

Adam et al. NeurIPS 2023 ML4PS workshop
PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

\[ \mathbf{v} = S F P_{\text{beam}} \mathbf{x} + \eta \]

Observed visibilities

Mask, subsampling function

Primary beam

Dense, unitary, Fourier operator

Sky emission

Gaussian noise

Noé Dia

Dia, Adam, Barth, et al. NeurIPS 2023 ML4PS workshop
PSF-DECONVOLUTION FOR INTERFEROMETRIC DATA

With Score-Based Diffusion Models

\[ \mathbf{v} = SFP_{\text{beam}} \mathbf{x} + \mathbf{\eta} \]

CLEAN  MPoL  Posterior sample (ours)  Posterior sample (ours)

HD143006

AS 209

Noé Dia

\[ k(x, R) \sim \mathcal{B}(n, \mathbb{P}_p(R)) \]

\[ \hat{N}_{x,i} \equiv n\hat{p}_{R_i}, \quad \hat{N}_{y,i} \equiv m\hat{p}_{R_i}, \]

\[ \hat{p}_{R_i} \equiv \frac{k(x, R_i) + k(y, R_i)}{n + m} \]

\[ \chi^2_{\text{PQM}} \equiv \sum_{i=1}^{n_R} \left[ \frac{(k(x, R_i) - \hat{N}_{x,i})^2}{\hat{N}_{x,i}} + \frac{(k(y, R_i) - \hat{N}_{y,i})^2}{\hat{N}_{y,i}} \right] \]

Given any sampling distribution, or generative model, if two sets of samples are generated from the same distribution, then the statistic $\chi^2_{PQM}$ follows a chi-square distribution with $n_R - 1$ degrees of freedom.

\[
\begin{align*}
41853553 & \quad 37249815 & \quad 8408016 \\
\end{align*}
\]

\[
\begin{align*}
\chi^2_{PQMaSS} & \quad \text{Epoch} \\
0 & \quad 100 \\
100 & \quad 0 \\
10^2 & \quad 10^4 \\
10^3 & \quad \text{VAE Model 1 Run 1} & \quad \text{VAE Model 1 Run 2} \\
& \quad \text{Diffusion Model 1 Run 1} & \quad \text{Diffusion Model 1 Run 2} \\
& \quad \text{Ideal} \\
\end{align*}
\]