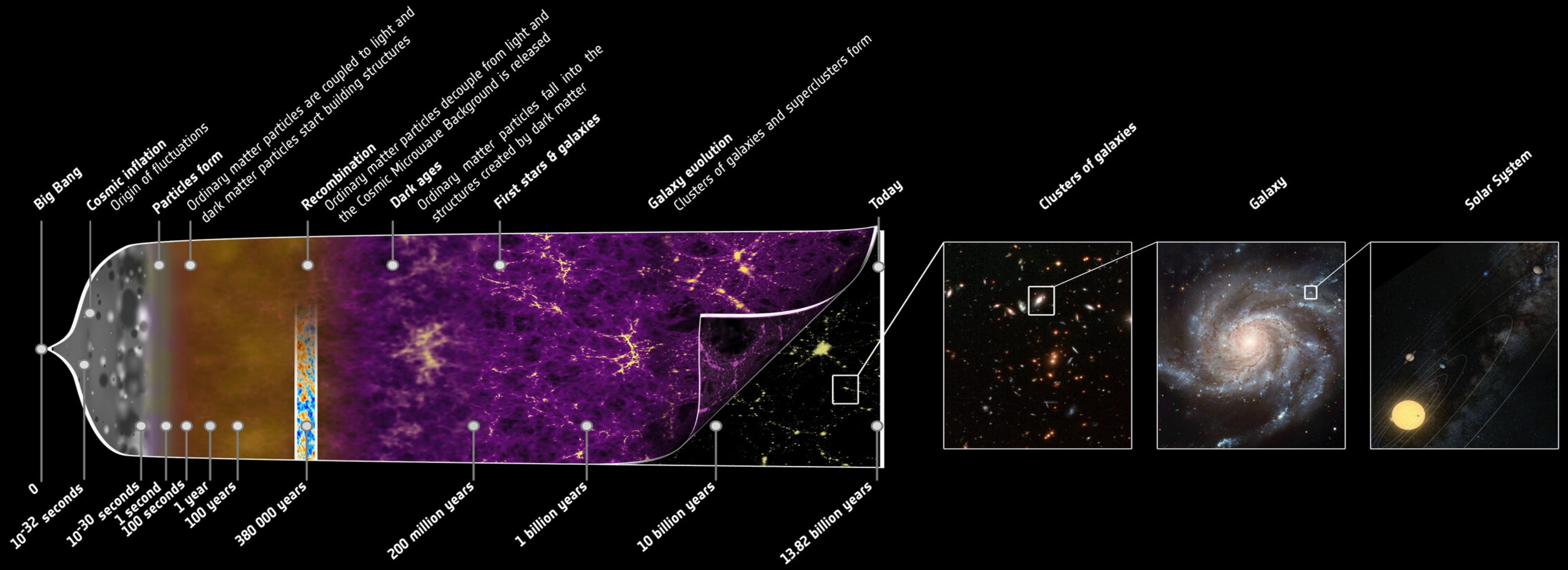
Likelihood- free Inference for Large Forward Models in Astrophysics/ Cosmology

Androniki Dimitriou– IFIC Valencia

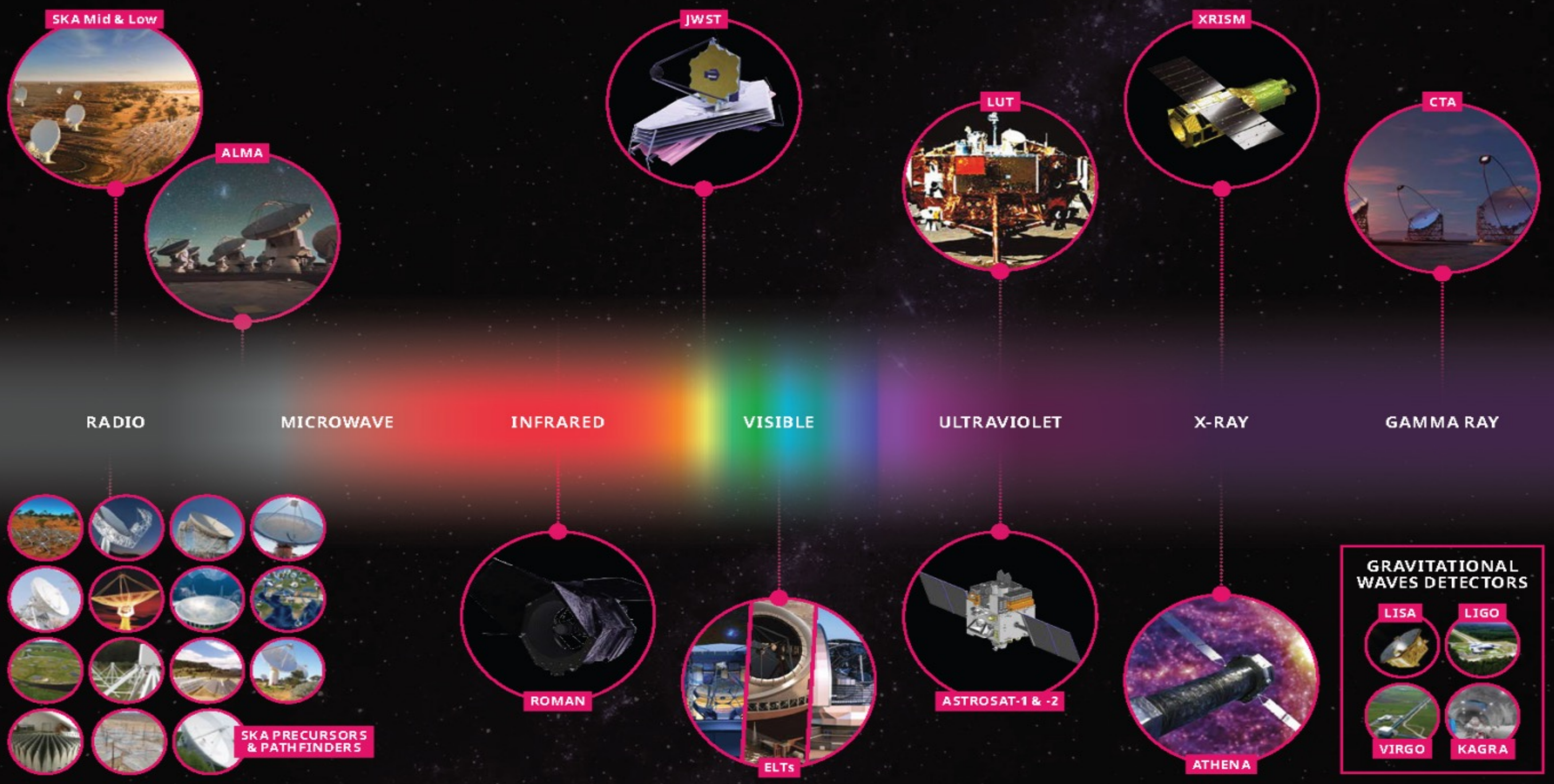
Fermilab, April 17, 2024

Timeline of the Universe



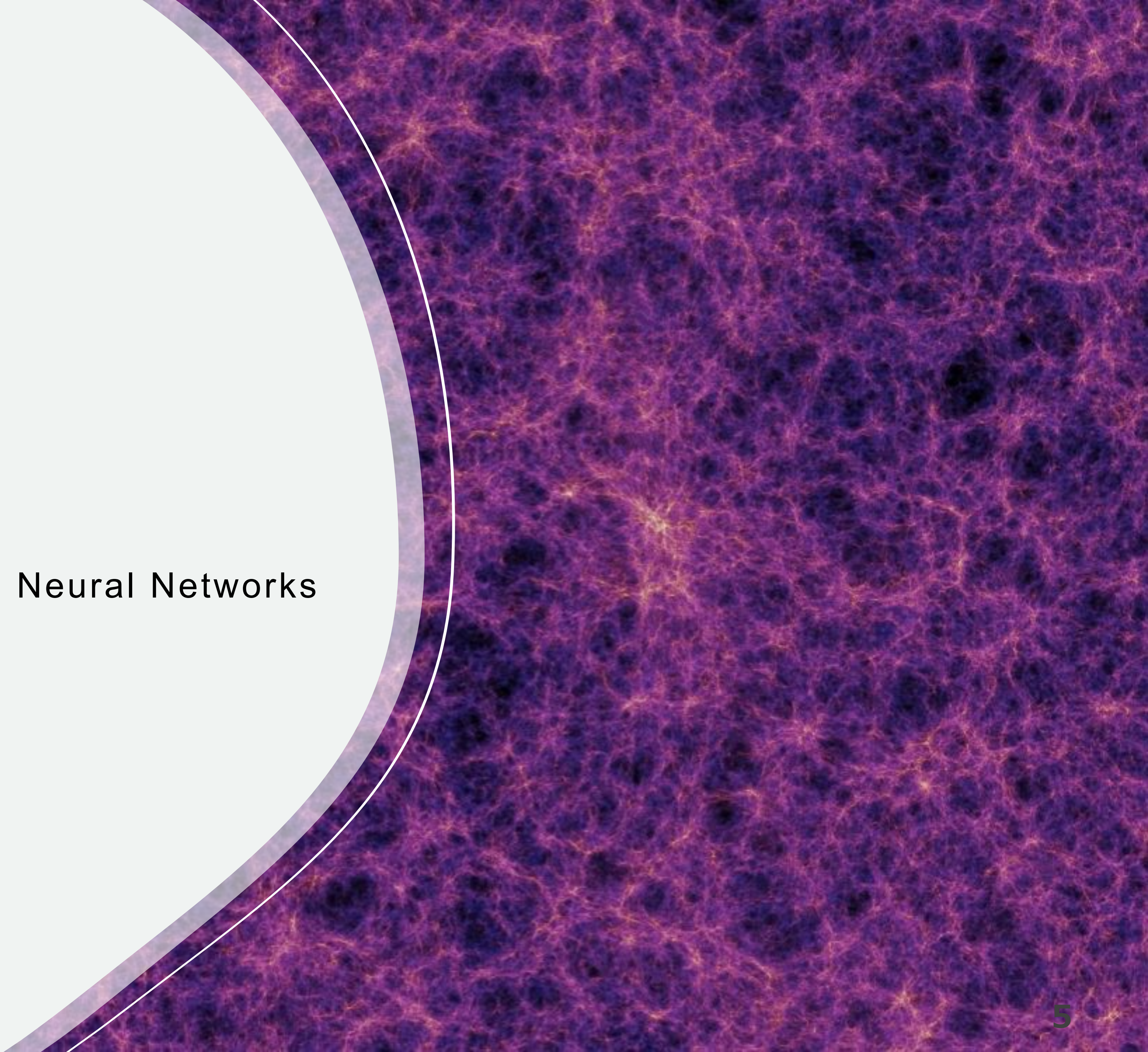
Structure Formation in the Universe

21st century cosmology

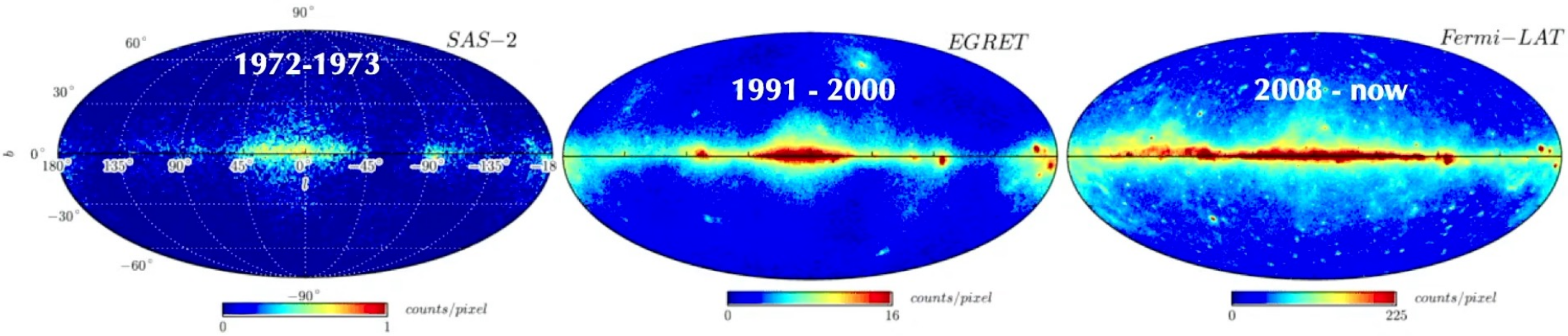


Overview

- Large Forward Models
- Inverse Problems
- Solving inverse problems with Neural Networks
- Examples



Large Forward Models



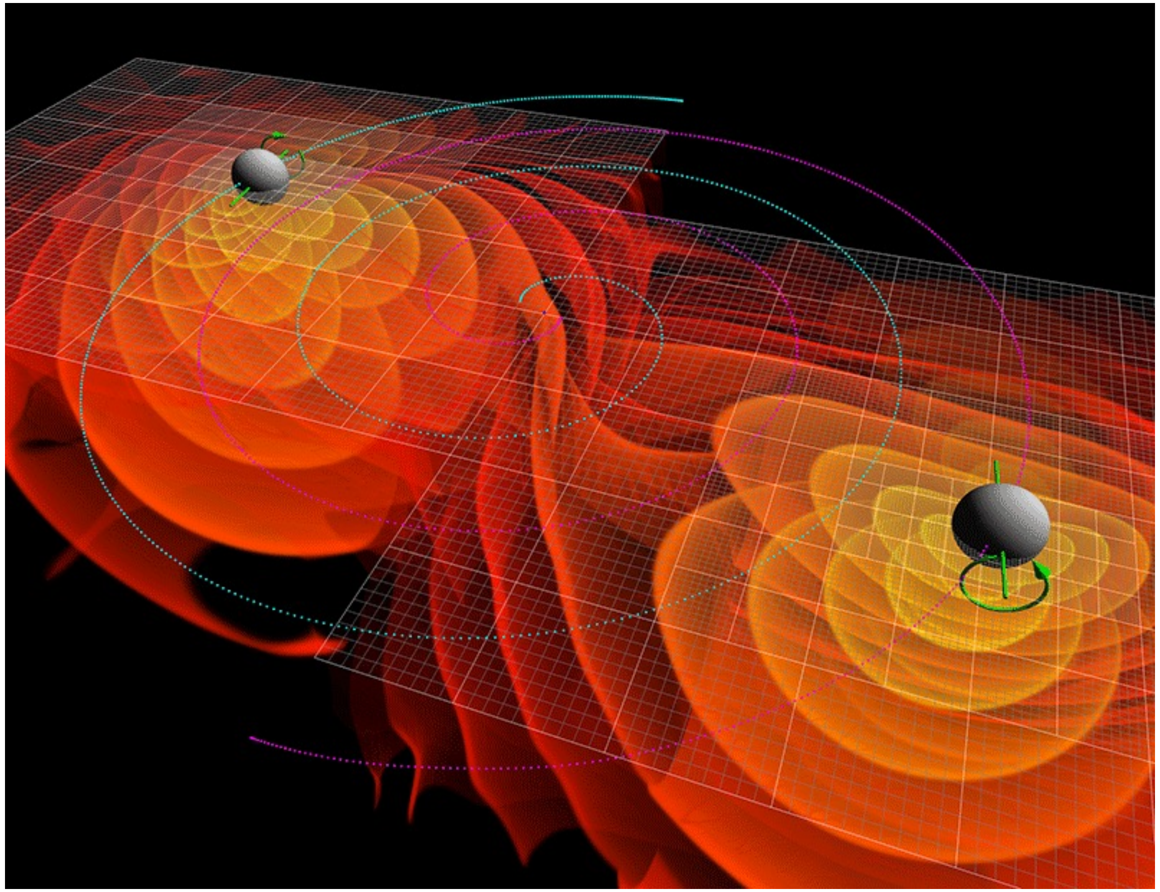
- ~ 6 Gamma-ray sources
- Diffuse emission

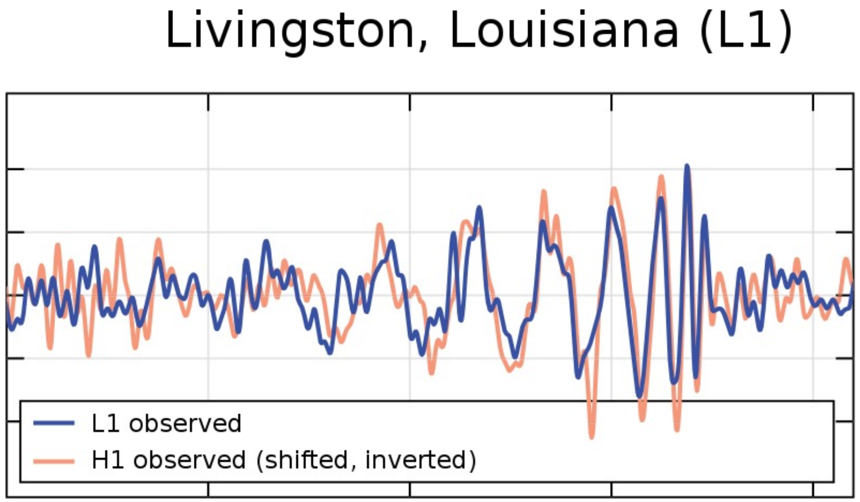
- 188 Gamma-ray sources
- Extended point sources
- Resolved gas emission

- 6658 Gamma-ray sources
- Highly resolved gas
- Many extended sources
- Fermi bubbles
- GeV excess
- Dark gas
- …

More data leads to more objects.
More objects lead to more parameters.
More parameters lead to more complicated statistical analyses.

Binary black hole merger ➡ Gravitational waves





- Computationally expensive numerical relativity simulations.
- Many parameters (masses or merging objects, positions in the sky etc…)

Characteristics of Large Forward Models

A forward model generates samples $\boldsymbol{x}, \boldsymbol{z} \sim \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z})\boldsymbol{p}(\boldsymbol{z})$. Why it could be large?

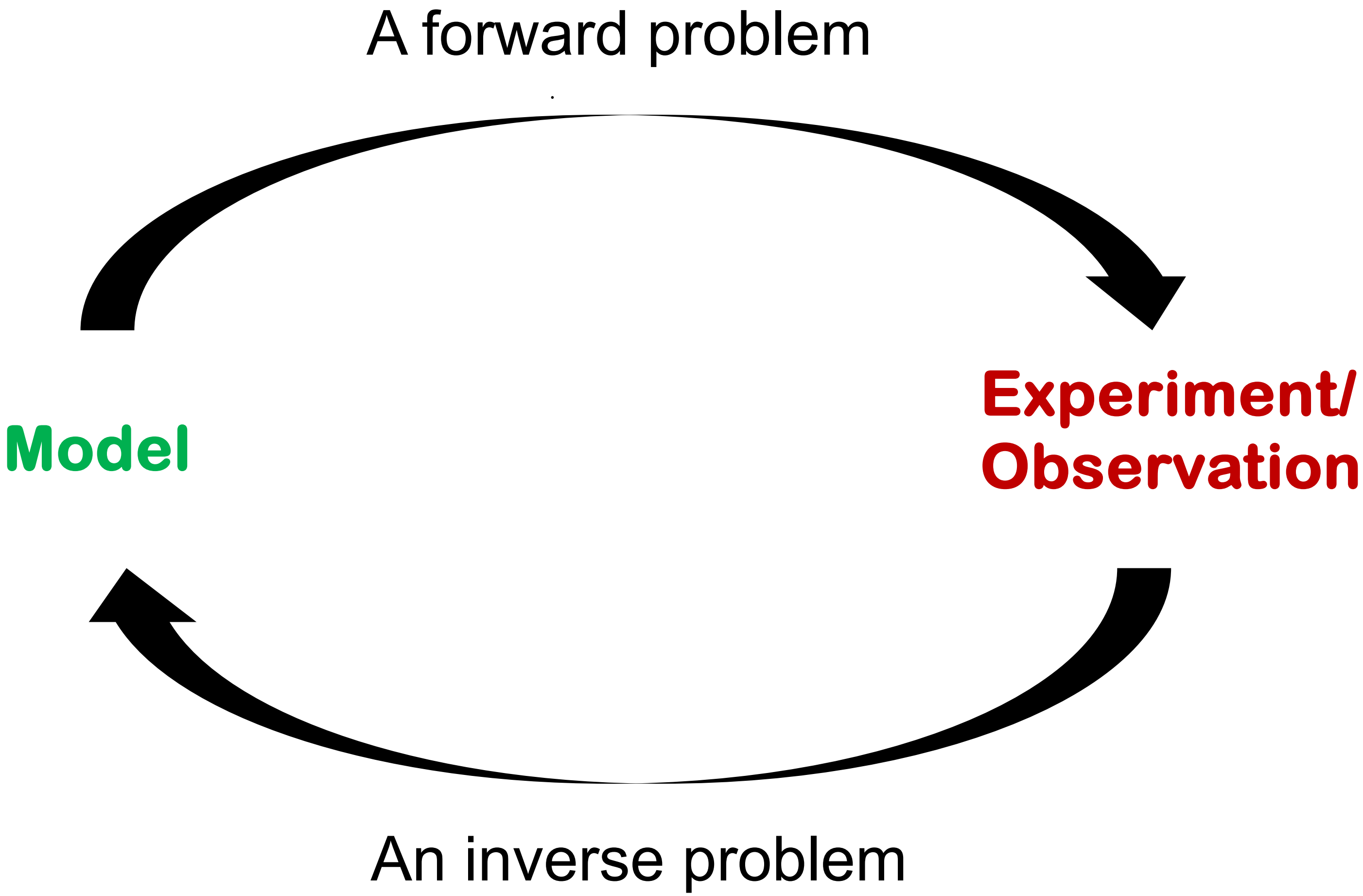
simulator

1. **Model Complexity:** Number of parameters with complex interactions between them
2. **Amount of Data:** Different data sources, volume of data
3. **Computational Recourses:** Computational power, memory or time to run

- **In Physics and Astronomy**

Forward models are usually principled. They implement specific physical laws.
Parameters $\boldsymbol{z}$ typically have an interpretation in terms of specific physical quantities.

Inverse Problems



Bayes Theorem

Likelihood

Posterior

Prior

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}$$

where $p(\boldsymbol{x}) = \int d\boldsymbol{z}\; p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

Evidence

Likelihood-based Inference

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}$$

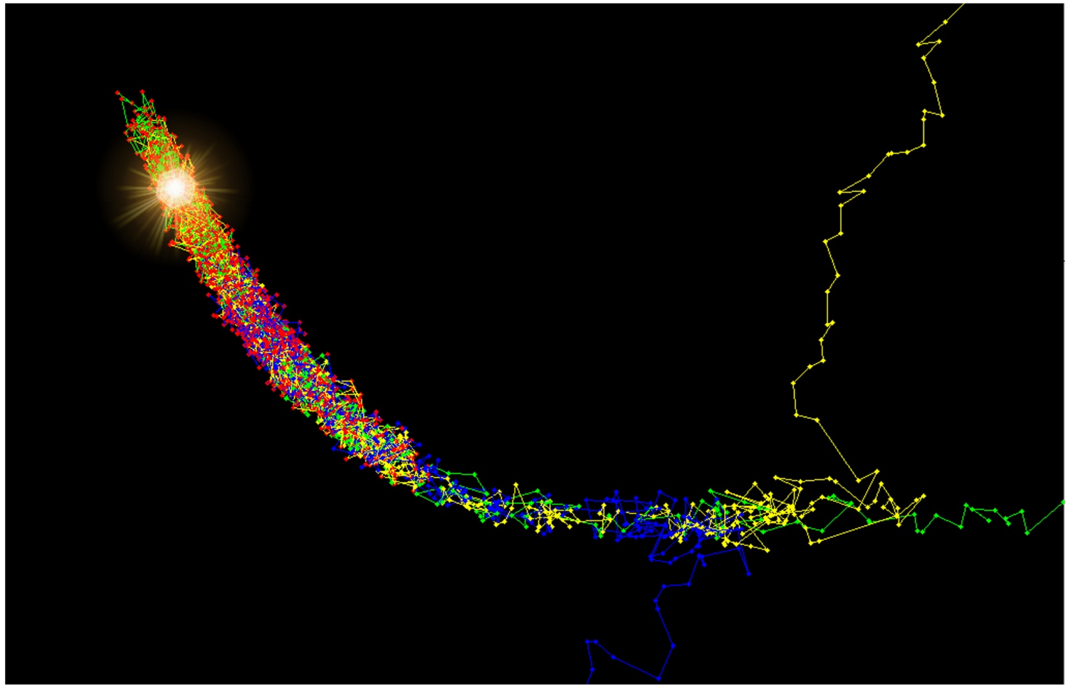


Image credit: Wikipedia

Ex: Metropolis Hastings Algorithm

- MC methods sample from the **joint high-dimensional posterior for all parameters**

$$\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{x}), \boldsymbol{z} \in \mathbb{R}^{\mathrm{D}}$$

D: Number of parameters

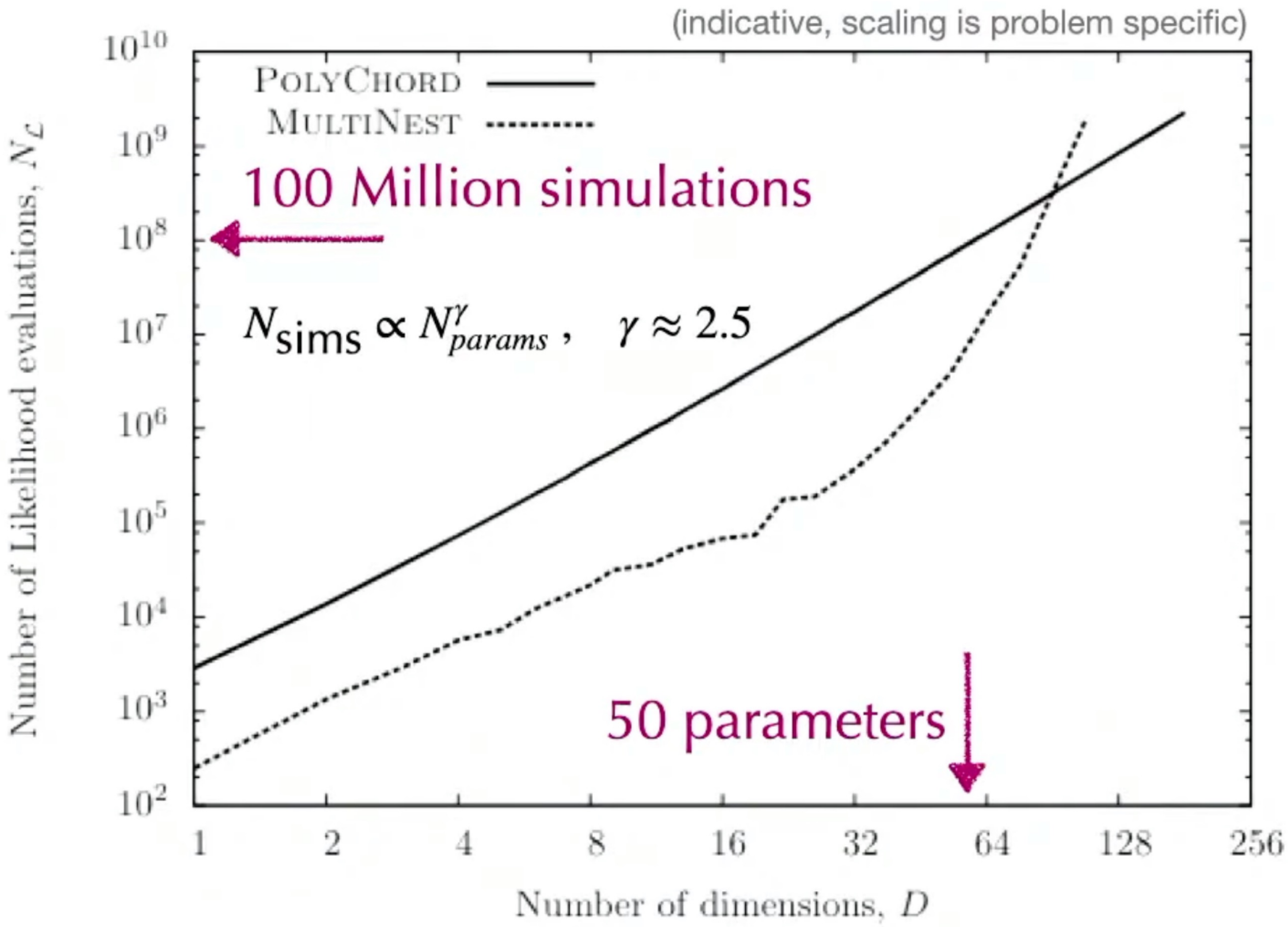
- **Obtain marginals by projection onto parameters of interest**

$$\boldsymbol{z} \equiv (z_1, z_2, \ldots, z_D)^T \rightarrow (zi, zj)^T \in \mathbb{R}^2$$

Expensive!

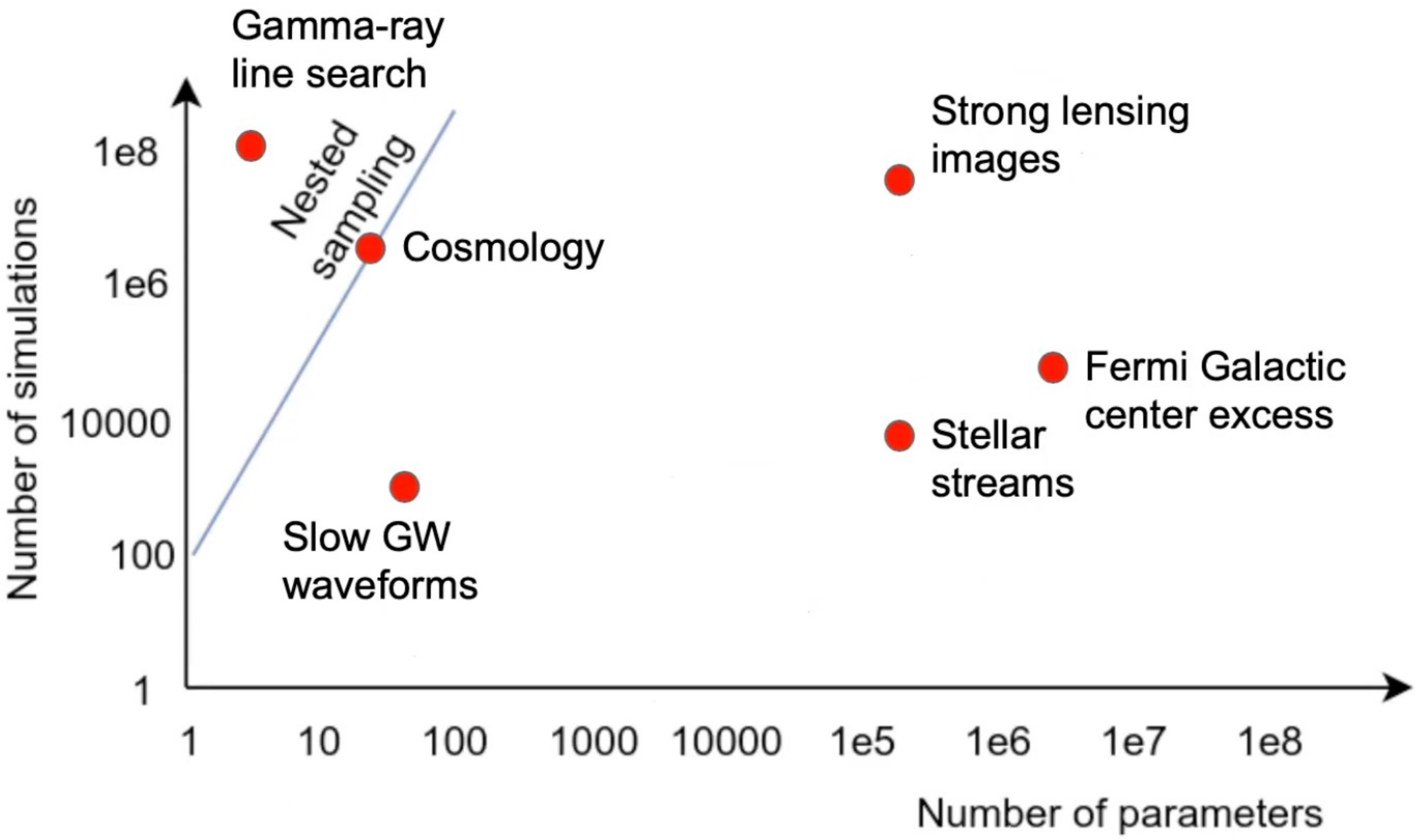
- Require an explicit likelihood

Likelihood-based Inference



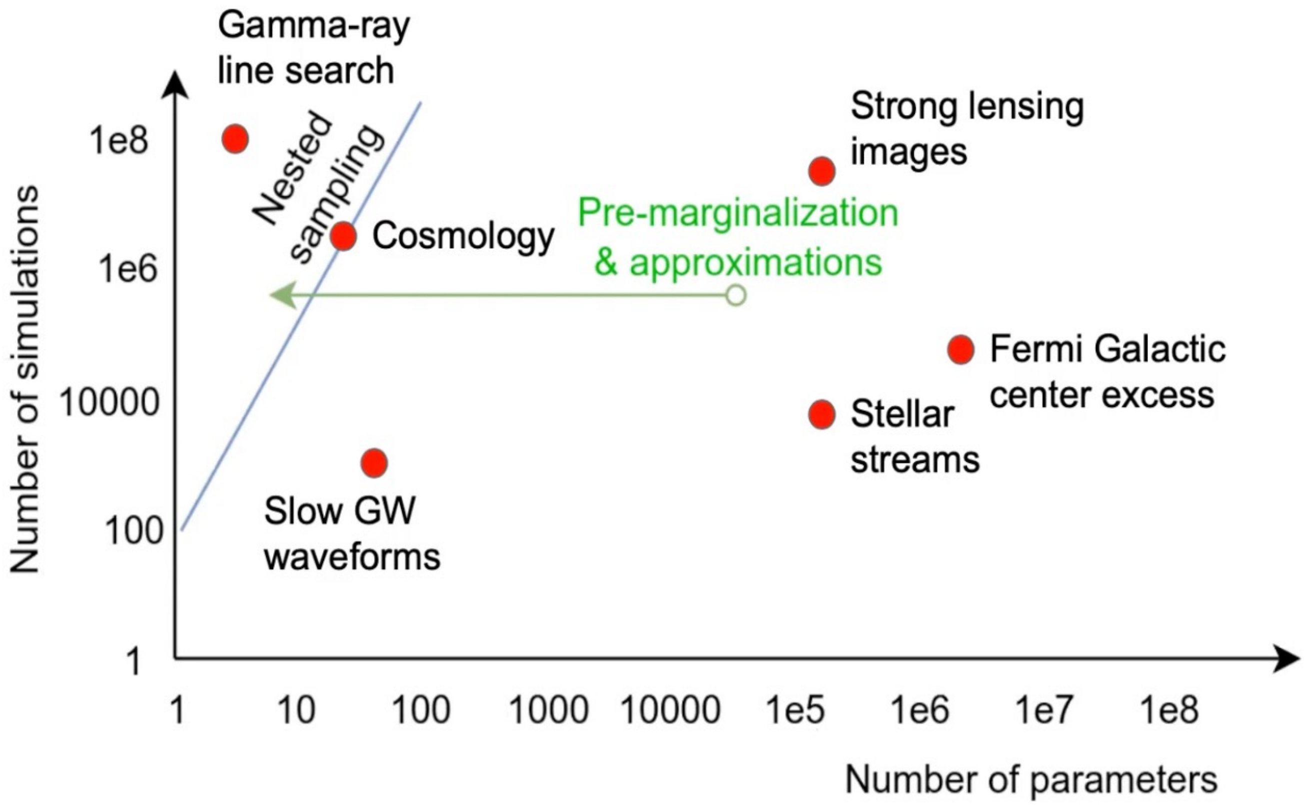
[Handley, Hosbon, Lasenby 1911.01429]

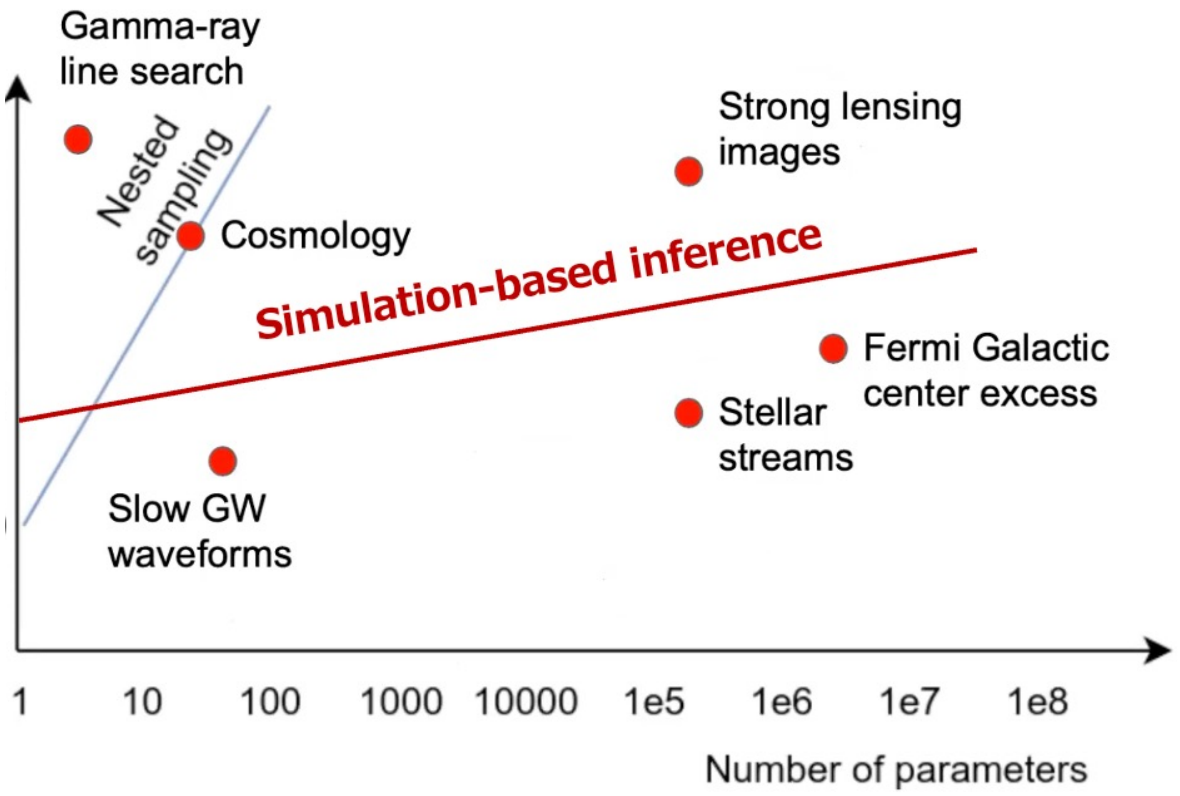
Likelihood-based Inference vs high dimensional models



Slide credit: Christoph Weniger

Making problems tractable





Slide credit: Christoph Weniger

A thought experiment

Simulated images:

1, 3, 2, **1**, 5, 4, 3, 1, 6, 7, 9, …

6, 2, 5, **8**, 6, 8, 4, 3, 2, 1, 3, …

3, 4, 2, **3**, 1, 7, 8, 9, 5, 3, 2, …

4, 2, 1, **4**, 6, 8, 6, 4, 3, 2, 4, …

1, 3, 2, **9**, 5, 4, 3, 1, 6, 7, 9, …

6, 2, 5, **8**, 6, 8, 4, 3, 1, 3, 4, …

2, 3, 4, **1**, 1, 7, 8, 9, 5, 3, 2, …

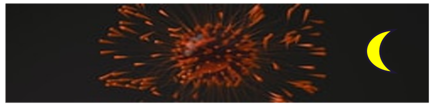
4, 2, 1, **2**, 6, 8, 6, 4, 3, 2, 4, …

1, 3, 2, **4**, 5, 4, 3, 1, 6, 7, 9, …

Red (position of moon):
parameter of interest

Black:
Nuisance parameters
(parametrizing *all* possible background images)

Observed data:



?

A thought experiment

Simulated images:

1, 3, 2, **1**, 5, 4, 3, 1, 6, 7, 9, …

6, 2, 5, **8**, 6, 8, 4, 3, 2, 1, 3, …

3, 4, 2, **3**, 1, 7, 8, 9, 5, 3, 2, …

4, 2, 1, **4**, 6, 8, 6, 4, 3, 2, 4, …

1, 3, 2, **9**, 5, 4, 3, 1, 6, 7, 9, …

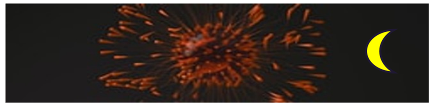
6, 2, 5, **8**, 6, 8, 4, 3, 1, 3, 4, …

2, 3, 4, **1**, 1, 7, 8, 9, 5, 3, 2, …

4, 2, 1, **2**, 6, 8, 6, 4, 3, 2, 4, …

1, 3, 2, **4**, 5, 4, 3, 1, 6, 7, 9, …

Observed data:



–, –, –, **8**, –,–, –, –, –, –, –,…

Red (position of moon):
parameter of interest

Black:
Nuisance parameters
(parametrizing *all* possible background images)

A thought experiment

→ **MCMC** →

After a Hubble time*

→ **Your brain as a Neural network** →

~ 8
After less than a second

You learned from a few examples to recognize the pattern related to the parameter of interest.

Simulation efficient: Just 8 examples are enough!

How can we achieve efficiency and precision?

Neural simulation-based inference (SBI)

Numerous methods exist with various pros/cons

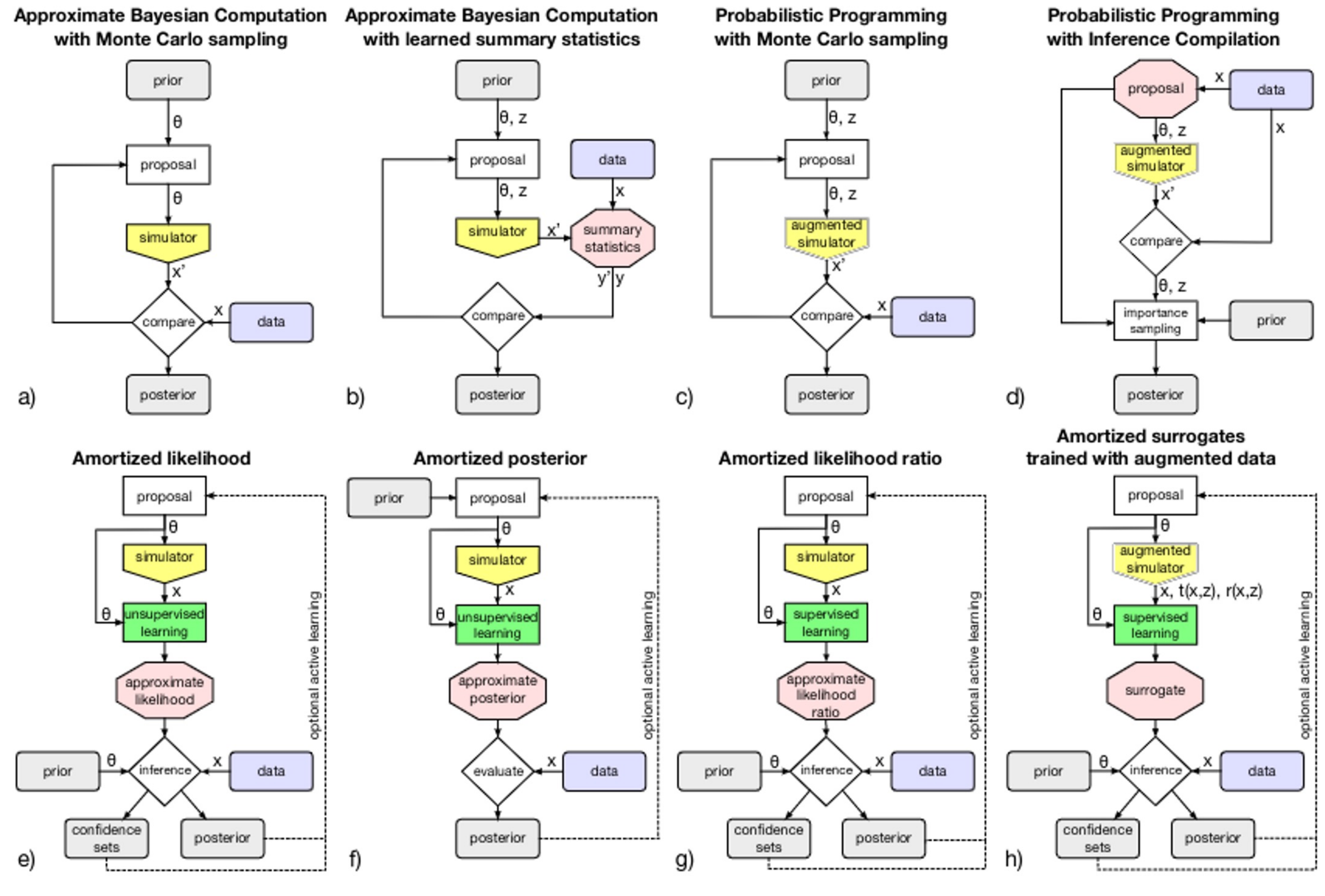


Fig. 3. Overview of different approaches to simulation-based inference.

[Cranmer, Brehmer, Louppe 1911.01429]

General goal: obtain ***neural network approximator*** for one of the following:

- Posterior* $\boxed{p(\boldsymbol{z}|\boldsymbol{x})}$
- Likelihood* $p(\boldsymbol{x}|\boldsymbol{z})$
- Ratios of posteriors and priors = ratios of likelihood and evidence

$$\boxed{\frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})} = \frac{p(\boldsymbol{x}|\boldsymbol{z})}{p(\boldsymbol{x})}}$$

- Various variations of the above quantities …

Neural Ratio Estimation (NRE)

Estimating the joint-vs-marginal ratio estimates the posterior

Bayes theorem

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

Strategy: We train a neural network $d_\varphi(\boldsymbol{x},\boldsymbol{z})$ as a binary classifier to be able to distinguish

- dependent sample-parameter pairs: $\boldsymbol{x},\boldsymbol{z} \sim p(\boldsymbol{x}|\boldsymbol{z})\, p(\boldsymbol{z})$ with class label Y=1 from
- independent ones: $\boldsymbol{x},\boldsymbol{z} \sim p(\boldsymbol{x})\, p(\boldsymbol{z})$ with class label Y=0.

[Cranmer, Begy, Louppe 1903.04057]

Neural Ratio Estimation (NRE)

We want the output of the NN to be the probability of class Y=1, i.e.,

$$\boxed{d_\varphi(\boldsymbol{x},\boldsymbol{z})} = p(\boldsymbol{y} = \mathbf{1}|\boldsymbol{x},\boldsymbol{z}) = \frac{p(\boldsymbol{x},\boldsymbol{z}|Y=1)p(Y=1)}{p(\boldsymbol{x},\boldsymbol{z}|Y=1)p(Y=1)+p(\boldsymbol{x},\boldsymbol{z}|Y=0)p(Y=0)} = \boxed{\frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x})p(\boldsymbol{z})}}$$

The corresponding loss function is the binary cross-entropy loss function:

$$\mathrm{L}[d_\varphi(\boldsymbol{x},\boldsymbol{z})] = -\int d\boldsymbol{x}\, d\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z})[-\ln(d_\varphi(\boldsymbol{x},\boldsymbol{z})]) + p(\boldsymbol{x})p(\boldsymbol{z})[\ln(1 - d_\varphi(\boldsymbol{x},\boldsymbol{z}))]$$

z={cats, donkeys}

$\boldsymbol{x},\boldsymbol{z} \sim p(\boldsymbol{x},\boldsymbol{z}) \propto p(\boldsymbol{x}|\boldsymbol{z})\, p(\boldsymbol{z})$

Draw labeled images

$\boldsymbol{x},\boldsymbol{z} \sim p(\boldsymbol{x})\, p(\boldsymbol{z})$

Draw images with random labels

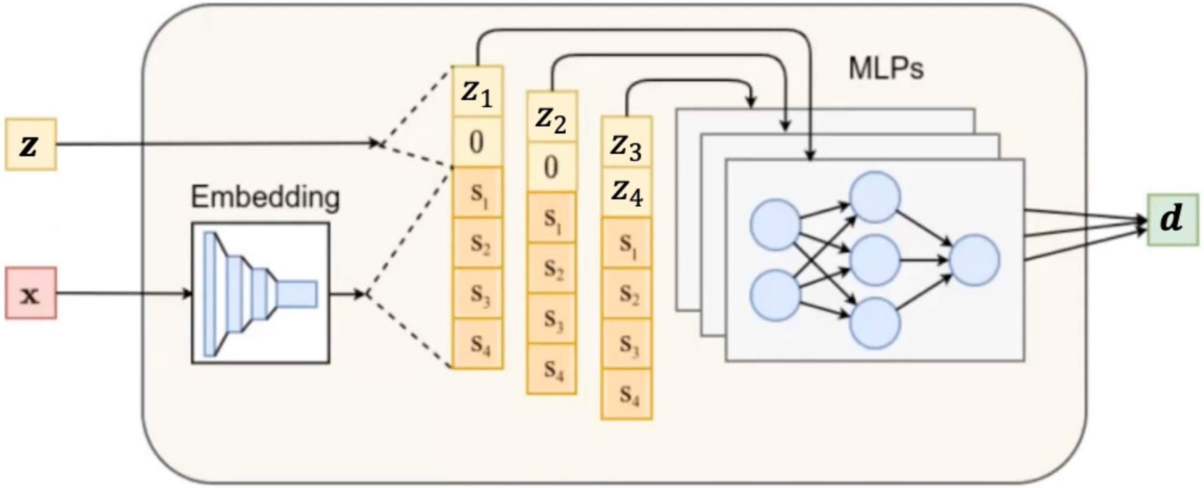
Minimizing this function w.r.t. the network parameters φ yields

$$d^*_\varphi(\boldsymbol{x},\boldsymbol{z}) = \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{x},\boldsymbol{z}) + p(\boldsymbol{x})\,p(\boldsymbol{z})} \quad \text{and thus} \quad r(\boldsymbol{x},\boldsymbol{z}) = \frac{d^*_\varphi(\boldsymbol{x},\boldsymbol{z})}{d^*_\varphi(\boldsymbol{x},\boldsymbol{z}) - 1}$$

[Cranmer, Brehmer, Louppe 1911.01429]

Using the result of the NN we have access to posteriors

$$p(\boldsymbol{z}|\boldsymbol{x}) = r(\boldsymbol{x},\boldsymbol{z}) \cdot p(\boldsymbol{z})$$



- Estimates: $p(z_1|\boldsymbol{x}), p(z_2|\boldsymbol{x}), p(z_3, z_4|\boldsymbol{x})$
- Does **not** estimate $p(z_1, z_2|\boldsymbol{x}), p(z_1, z_2, z_3|\boldsymbol{x})$

Pros

- **Implicit likelihood → Cherry-picking marginal posteriors!**
- No need to retrain to look at a different observation (**Amortization**)
- Ability to focus on a single observation through truncation
- Easy network architecture

Cons

- Difficult to sample from high-dim posteriors

Example 1:

A toy model for N-body simulations

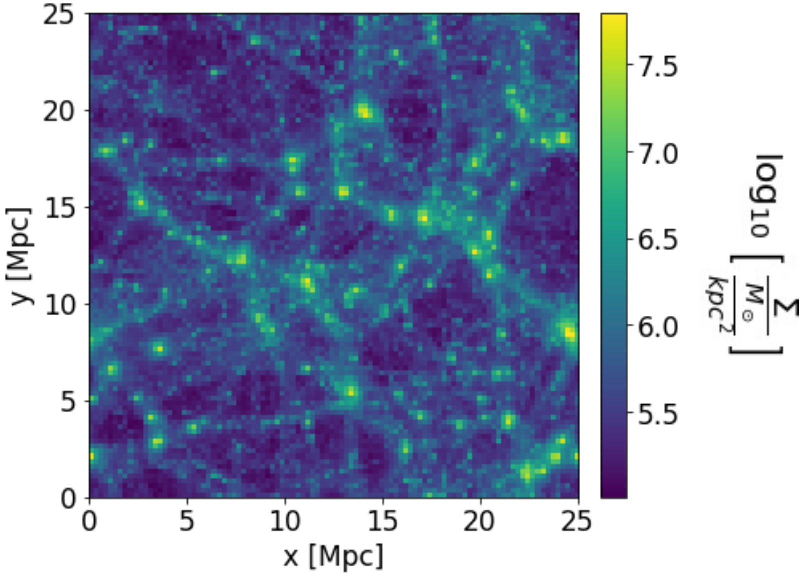
[**AD**, Camila Correa,Christoph Weniger 2206.11312]

- **Goal:** reconstruction of **halo clustering** and **halo mass function** of DM-only cosmological simulations generated by the **EAGLE project**

- **Approach:** an **analytical halo model** based **on a toy implementation** of two body correlation functions

- **The EAGLE project**

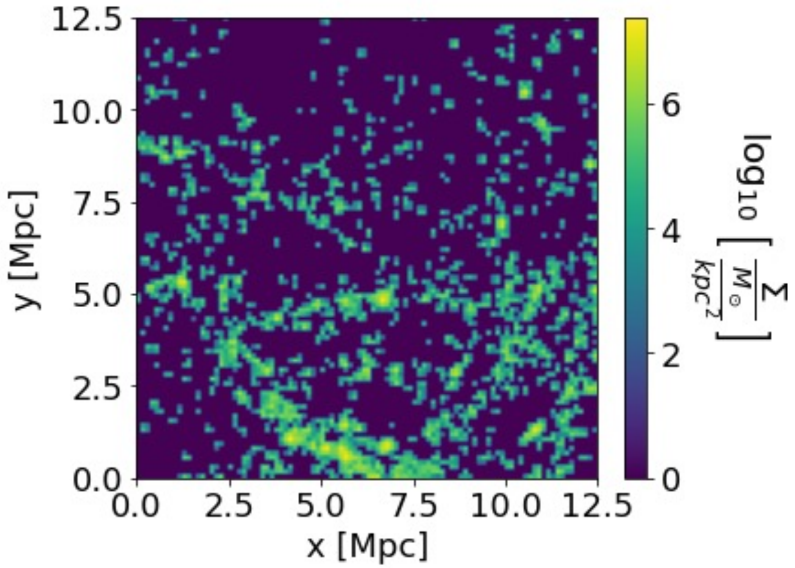
$(25\ \text{Mpc})^3$ box with 376^3 particles



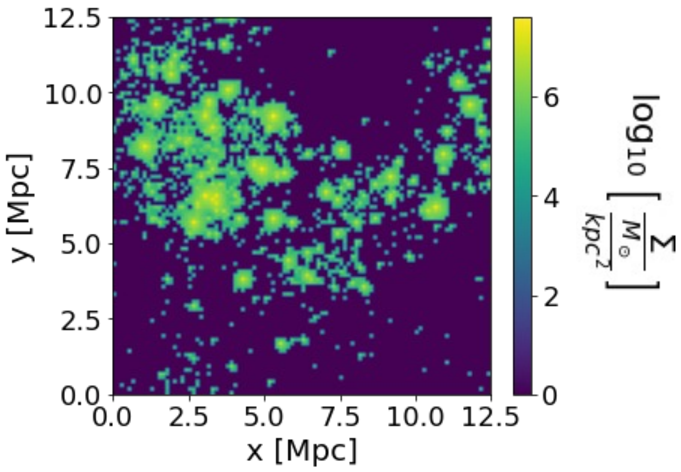
FOF halo finder

Particle data

$M_h \in (10^9, 10^{12})\ M_\odot,\ (12.5\ \text{Mpc})^3$ box



- **The toy halo model**



- The **first parameter of interest** is the number, N, of the haloes

- The masses of the haloes, $M_h \in (10^9, 10^{12})\ M_\odot$, can be sampled from a halo mass function

$$\frac{dn}{dM} \propto M^{-a}$$

- The **second parameter of interest** is the slope, a, of the halo mass function

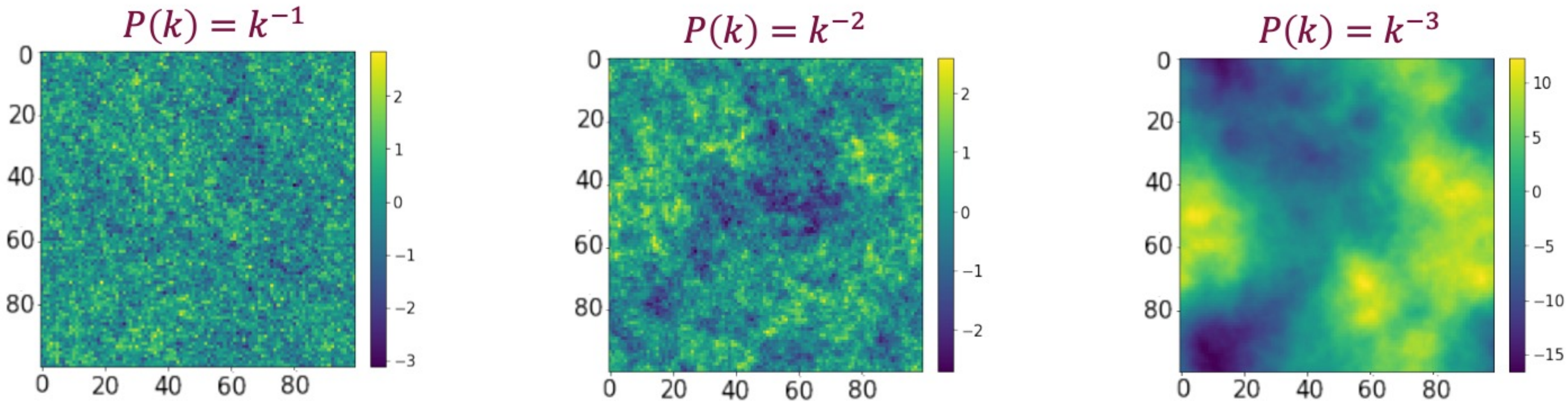
- **Adding Clustering to the Model**

  - We will sample the positions according to distributions generated from 2D realizations of gaussian random fields, $\boldsymbol{\delta}$

  - The gaussian fields will be specified by a power-law power spectrum

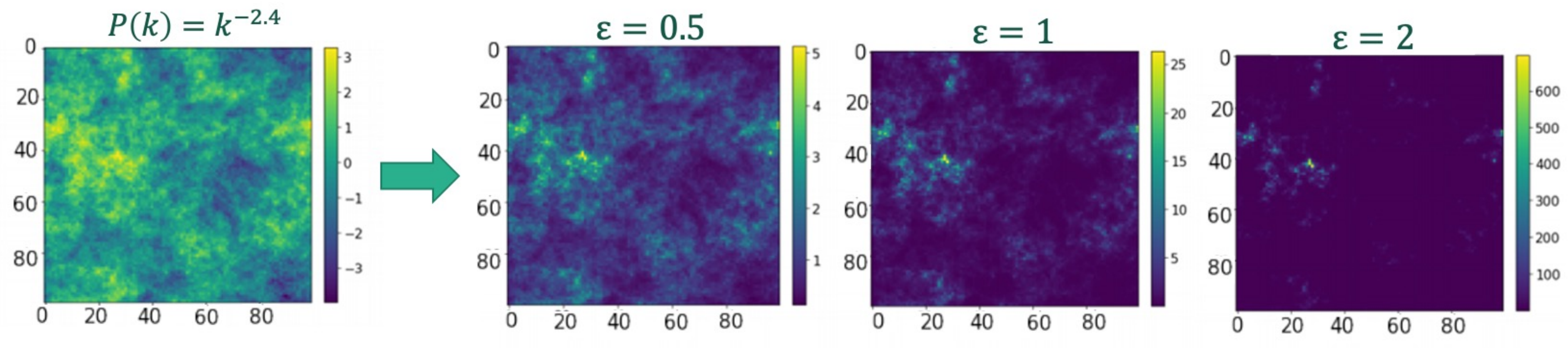
$$P(k) \propto \frac{1}{k^n}$$

  - The slope of the power spectrum, n, is the **third parameter of interest** of our model



- **Adding Clustering to the Model**

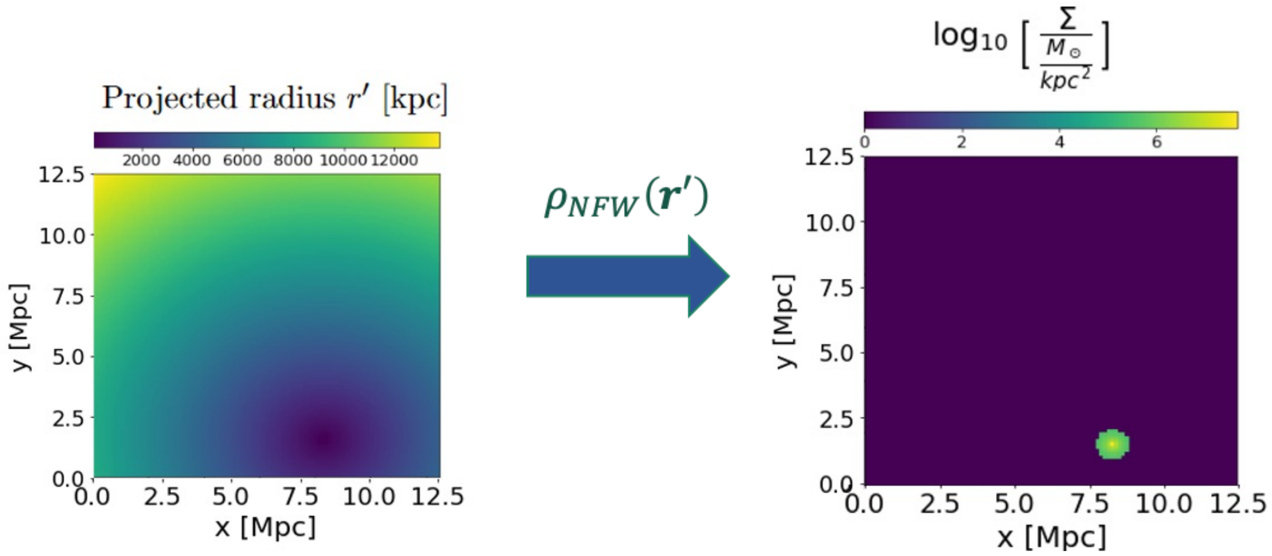
  - We transform the field $\boldsymbol{\delta}$ to a probability distribution function to sample from it:
    - We first multiply $\boldsymbol{\delta}$ with a **fourth parameter of interest** ε,
    - We exponentiate $\boldsymbol{\delta} \cdot \varepsilon$,
    - We normalize the field $\boldsymbol{f} = e^{\boldsymbol{\delta}\cdot\varepsilon}$, s.t., its values sum to 1



  - We sample the positions of the haloes according to this distribution

- ## The toy halo model

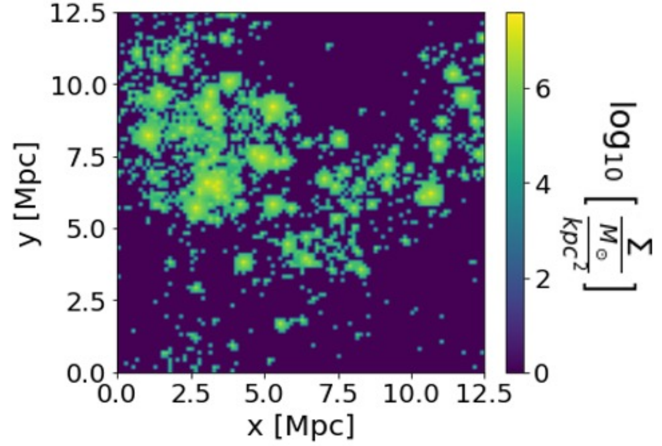
➢ We calculate the logarithmic surface density



➢ We add all the images of the individual haloes together to obtain the total surface density field

mock data

➢ We add poisson noise to the final image



Thousands of nuisance parameters: positions of haloes, masses

Training NRE

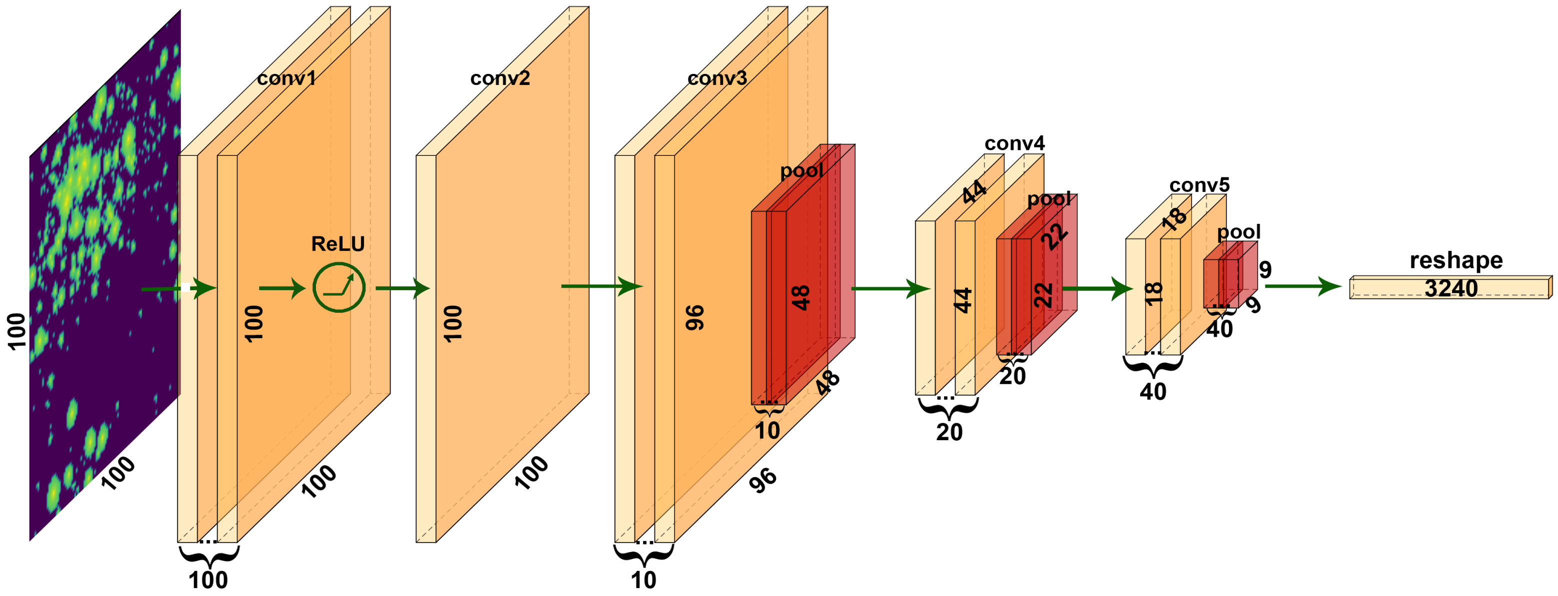
Parameters of interest:

- N: Number of halos, where $N \in (100, 2100)$
- a : Inner slope of the halo mass function, $a \in (1, 3)$

ε: Exponent of the density field, where $\varepsilon \in (0,2)$

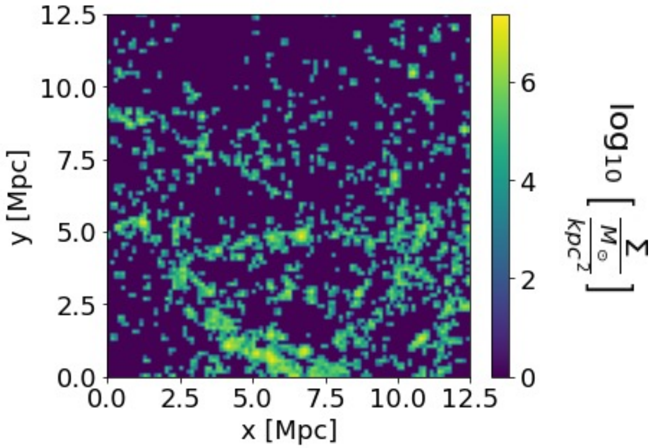
n: Slope of the power spectrum, where $n \in (0,10)$

- We define a **CNN**:

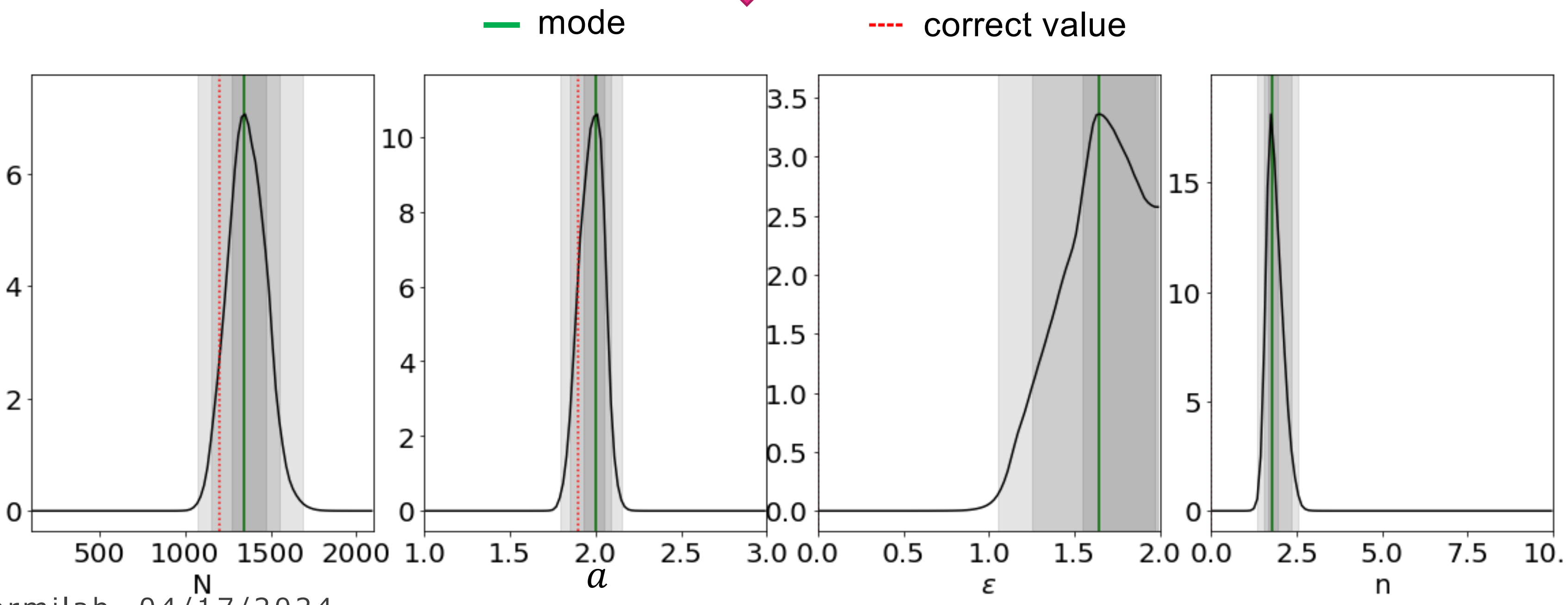


- We **train** using 200.000 mock images

Results on actual N body simulations

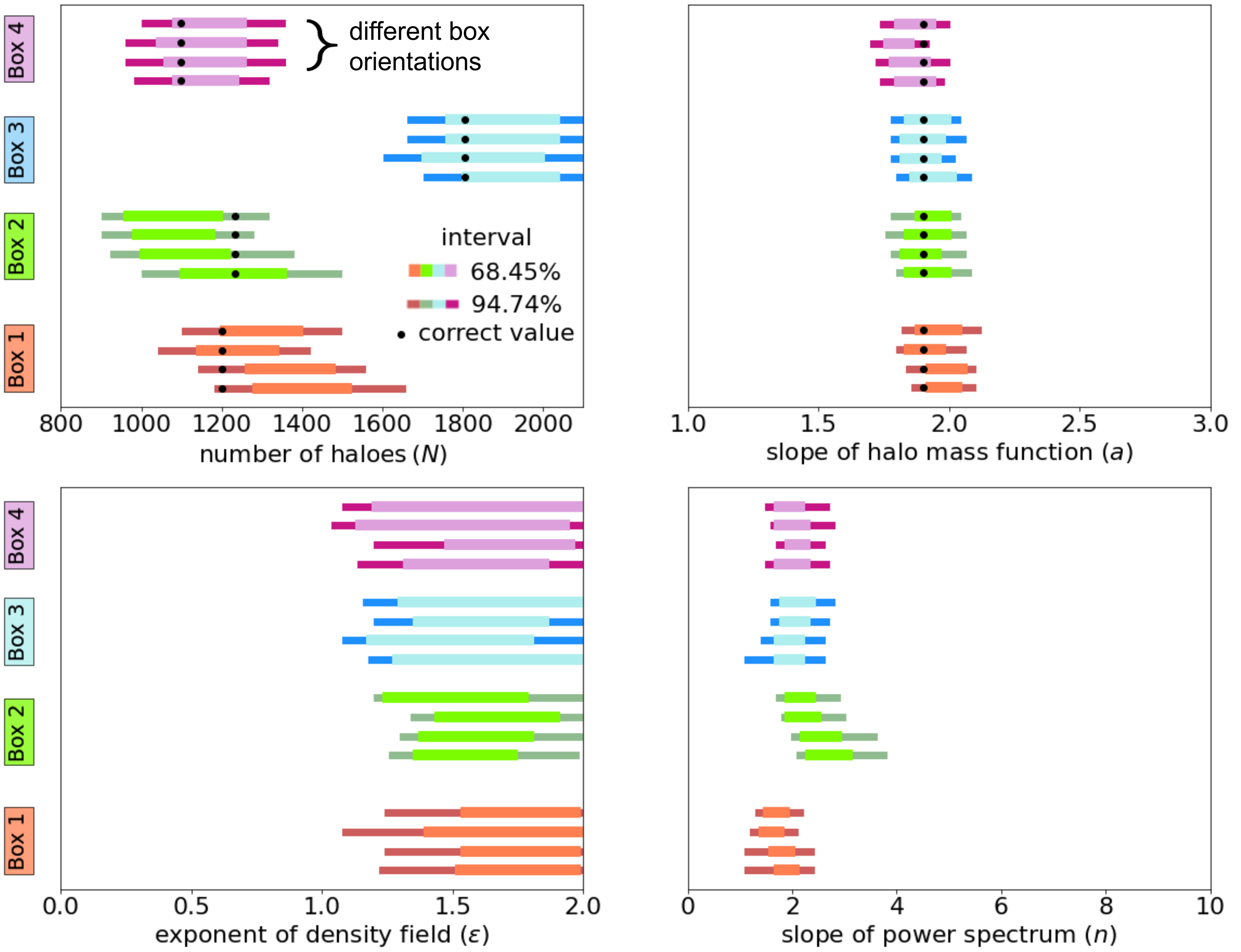


trained NN

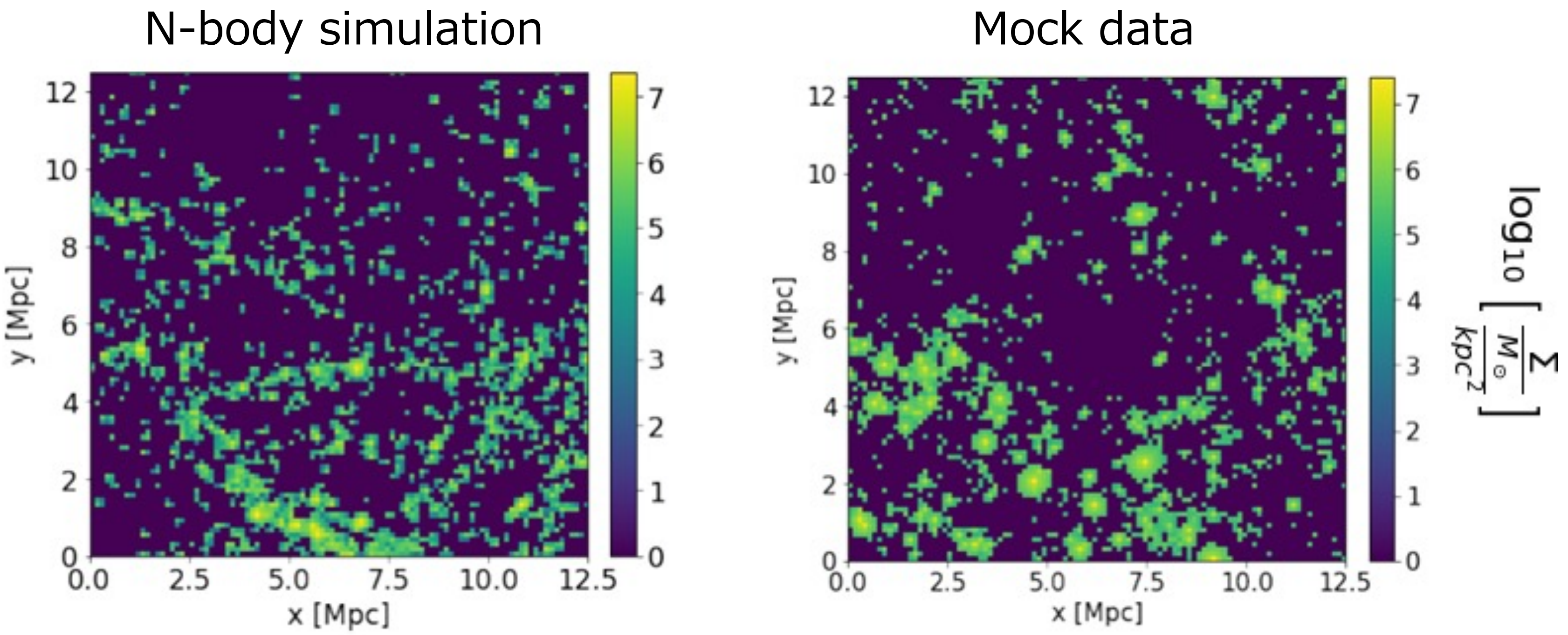


Results on actual N body simulations

- 4 simulation boxes and 3 rotations of them



N-body simulation image vs mock image



Neural Posterior Estimation (NPE)

Approximate $p(\boldsymbol{z}|\boldsymbol{x_0})$ with a NN $q_\varphi(\boldsymbol{z}|\boldsymbol{x_0})$. How can we compare the two pdfs?

$$D_{KL}(p||q_\phi) = \int d\boldsymbol{z}\; p(\boldsymbol{z}|\boldsymbol{x}_0)\ln\left[\frac{p(\boldsymbol{z}|\boldsymbol{x}_0)}{q_\varphi(\boldsymbol{z}|\boldsymbol{x}_0)}\right] = \overbrace{\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x}_0)}}^{\text{intractable}}\ln\left[\frac{p(\boldsymbol{z}|\boldsymbol{x}_0)}{q_\varphi(\boldsymbol{z}|\boldsymbol{x}_0)}\right] \text{ intractable}$$

Problem: How to avoid sampling from the intractable posterior?

Solution: Amortize (average) over simulation data. The loss is now the expected KL divergence, where we have exploited Bayes theorem, i.e., $p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})= p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})$.

$$\underbrace{\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}}_{\text{tractable}}\underbrace{\left[D_{KL}(p||q_\phi)\right]}_{\text{intractable}} = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}}_{\text{tractable}}\underbrace{\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{x})}}_{\text{intractable}}\left[\ln \overbrace{\frac{p(\boldsymbol{z}|\boldsymbol{x})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}}^{\text{intractable}}\right] = \overbrace{\mathbb{E}_{\boldsymbol{x},\boldsymbol{z}\sim p(\boldsymbol{x},\boldsymbol{z})}}^{\text{tractable}}\left[\ln \overbrace{\frac{p(\boldsymbol{z}|\boldsymbol{x})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}}^{\text{intractable}}\right]$$

$$\boldsymbol{x},\boldsymbol{z}\sim p(\boldsymbol{x},\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})\mathrm{p}(\boldsymbol{z})$$

Neural Posterior Estimation (NPE)

tractable | intractable but constant

$$\mathbb{E}_{x\sim p(x)}\left[D_{KL}(p||q_\phi)\right] = \mathbb{E}_{x,z\sim p(x,z)}\left[\ln \frac{p(\mathbf{z}|\mathbf{x})}{q_\varphi(\mathbf{z}|\mathbf{x})}\right] = \underbrace{-\mathbb{E}_{x,z\sim p(x,z)} \ln q_\varphi(\mathbf{z}|\mathbf{x})}_{\text{tractable}} + \underbrace{\mathbb{E}_{x,z\sim p(x,z)} \ln p(\mathbf{z}|\mathbf{x})}_{\text{intractable but constant}}$$

Training inference networks with forward KL leads to posteriors that minimize the posterior entropy:

$$\mathcal{L} = -\mathbb{E}_{\mathbf{x},\mathbf{z}\sim p(\mathbf{x},\mathbf{z})} \ln q_\varphi(\mathbf{z}|\mathbf{x}) \longrightarrow \varphi^* \text{ (optimum parameters)}$$

[Papamakarios, Murray 1605.06376]

Pros

- **Implicit likelihood → Cherry-picking marginal posteriors!**
- No need to retrain to look at a different observation **(Amortization)**
- Ability to focus on a single observation with multi-round inference
- Obtain samples and evaluate the posterior

Cons

- q_φ should be a PDF, i.e. normalizing flow or Gaussian mixture model, **sometimes difficult to train**

Example 2:
Reconstruction of Gravitational Wave Backgrounds

[**AD**, Daniel G. Figueroa, Bryan Zaldivar 2309.08430]

- **Goal:** Reconstructing the **spectral shape** $\Omega_{GW}(f)$ **of an unknown Stochastic Gravitational Wave Background (SGWB)** with **LISA:**

  - will consist of a constellation of **three satellites** forming a nearly equilateral triangle with ~ 2.5 million km length arms

  - will perform **three correlated interferometric measurements** (X, Y and Z data streams)

  - X, Y and Z data streams can be **transformed into three uncorrelated ones** (A, E, and T), which diagonalise the signal and noise covariance matrices

  - will cover **frequencies** from $3{\times}10^{-5}$Hz to 0.5 Hz

- **Approach: A model for data** as **close** as possible **to the real data-taking procedure of LISA**

  - We assume that the LISA mission will last 4 years.
  - The data will be collected 75% of the time and will be clean of noise glitches and transient signals.
  - Due to the need for regular operational breaks, the data will be collected in 94 time chunks (N_c).

  - **Data**

α, β: given channels

GW signal **Noise**

$$D_{i,j}^{\alpha\beta} = \boxed{S_{i,j}} + \boxed{\mathcal{N}_{i,j}^{\alpha\beta}}$$

$i \in \{1, 2, ..., N_c\}$
time chunks

$f_j \in \{3\times10^{-3}, 0.5\}$

GW signal Noise

- **Data** $$D^{\alpha\beta}_{i,j} = S_{i,j} + \mathcal{N}^{\alpha\beta}_{i,j}$$ where

$$S_{i,j} = \left| \frac{G_1\left(0, \sqrt{\Omega_{\rm GW}(f_j)}\right) + iG_2\left(0, \sqrt{\Omega_{\rm GW}(f_j)}\right)}{\sqrt{2}} \right|^2$$

Gs samples from Gaussian distributions

$$\mathcal{N}^{\alpha\beta}_{i,j} = \left| \frac{G_3\left(0, \sqrt{\Omega^{\alpha\beta}_{\rm noise}(f_j)}\right) + iG_4\left(0, \sqrt{\Omega^{\alpha\beta}_{\rm noise}(f_j)}\right)}{\sqrt{2}} \right|^2$$

- For each frequency f_j , we obtain the mean across time chunks for each channel

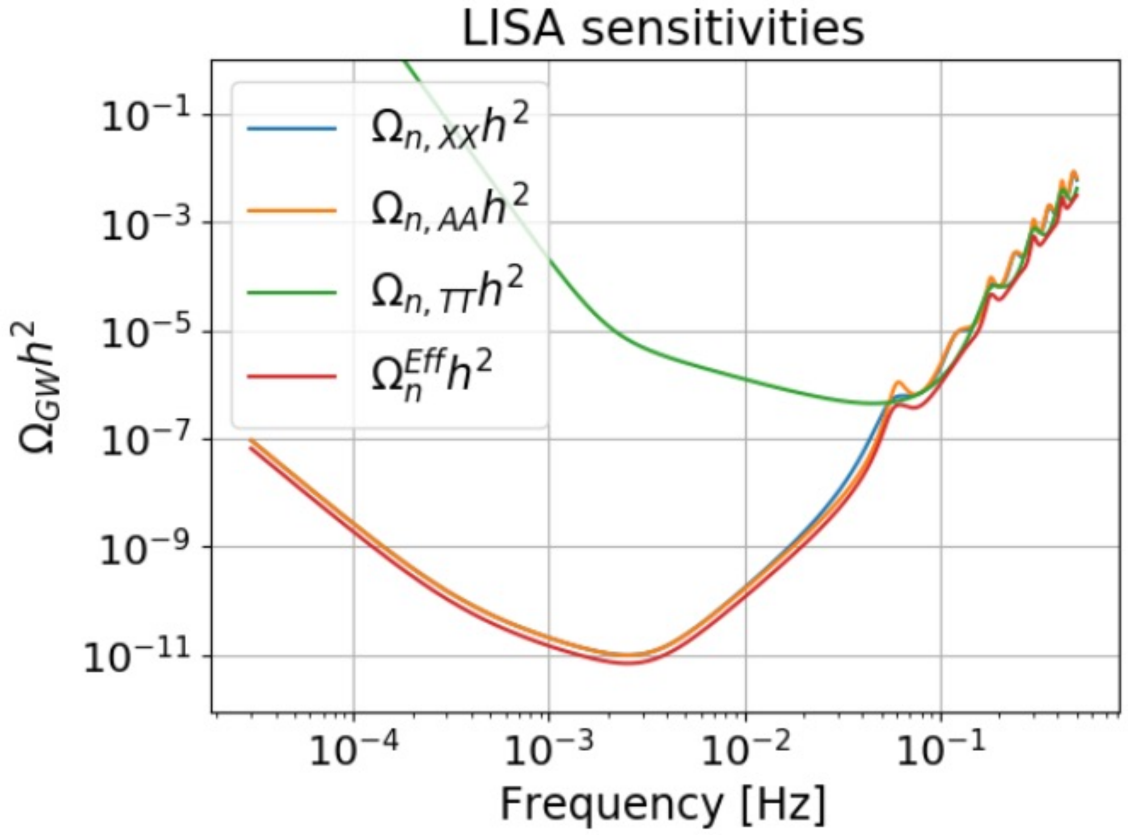
$$\bar{D}^{\alpha\beta}_{j} = \frac{1}{N_c}\sum_{i=1}^{N_c} D^{\alpha\beta}_{i,j}$$

- **Noise power spectra**

  - We model the LISA noise spectra in the A, E, and T channel basis,

$$\Omega^{\alpha\beta}_{noise}\left(f, \boxed{A_P, A_{acc}}\right) \begin{cases} \Omega^{AA}_{noise}\left(f, A_P, A_{acc}\right) = \Omega^{EE}_{noise}\left(f, A_P, A_{acc}\right) \\ \\ \Omega^{TT}_{noise}\left(f, A_P, A_{acc}\right) \end{cases}$$

[Flauger et al. 2009.11845]



- **GW power spectrum**

➢ We define a GW signal with an unknown frequency profile by approximating it **in a piece-wise** manner as **power-laws** in 27 intervals (27 *bins*).

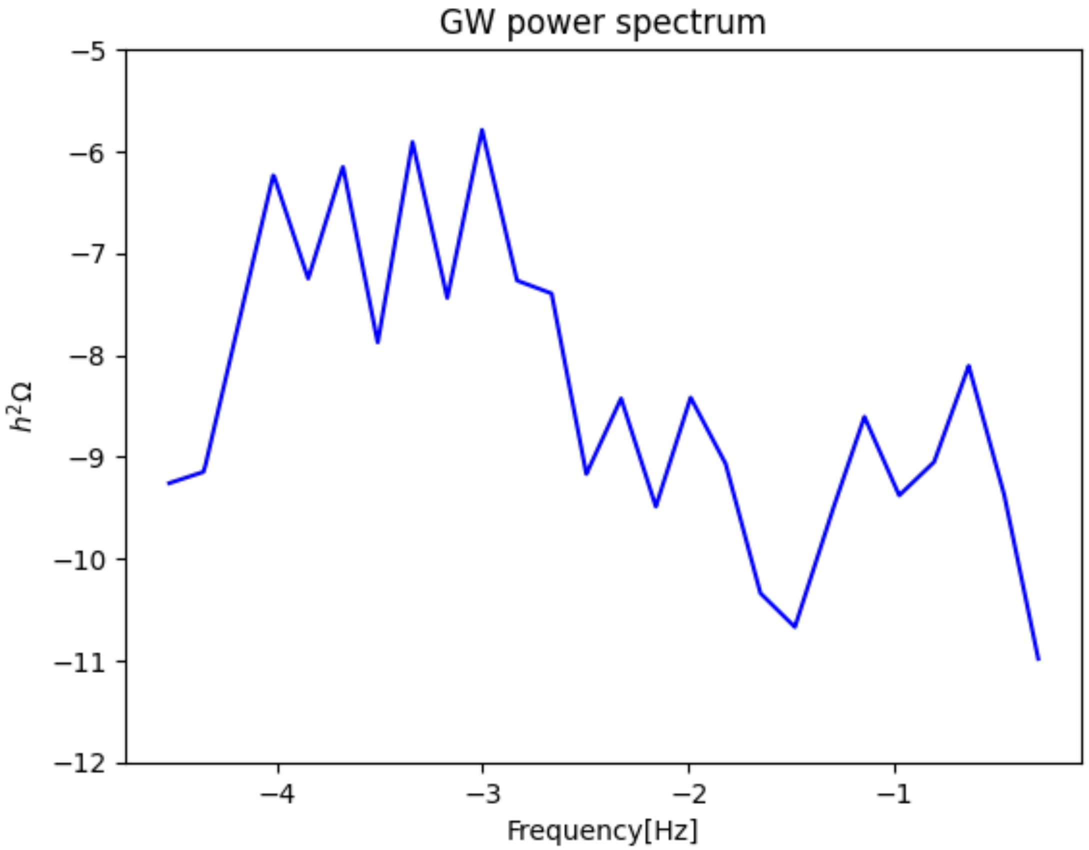
$$\Omega_{\rm GW}(f;\mathbf{s}) = \Omega_p^{(*)} \cdot \begin{cases} \vdots \\ F_{-2}(f) & \text{if } f_{-2} \le f < f_1 \\ F_{-1}(f) & \text{if } f_{-1} \le f < f_0 \\ F_0(f) & \text{if } f_0 \le f < f_1 \\ F_{+1}(f) & \text{if } f_1 \le f < f_2 \\ F_{+2}(f) & \text{if } f_2 \le f < f_3 \\ \vdots \end{cases}$$

with

$$F_0(f) \equiv \left(\frac{f}{f_0}\right)^{\gamma_0},$$

$$F_{-1}(f) \equiv F_0(f_0)\left(\frac{f}{f_0}\right)^{\gamma_{-1}}, \quad F_{-2}(f) \equiv F_{-1}(f_{-1})\left(\frac{f}{f_{-1}}\right)^{\gamma_{-2}}, \quad \ldots$$

$$F_{+1}(f) \equiv F_0(f_1)\left(\frac{f}{f_1}\right)^{\gamma_1}, \quad F_{+2}(f) \equiv F_{+1}(f_2)\left(\frac{f}{f_2}\right)^{\gamma_2}, \quad \ldots$$



1 Amplitude: $\Omega_p(*)$ and 27 slopes: …, γ_{-1}, γ_0, γ_1, …

➢ Given a dataset $\boldsymbol{D_0}=\{\boldsymbol{D_o}^{AA}, \boldsymbol{D_o}^{EE}, \boldsymbol{D_o}^{TT}\}$ we want to reconstruct and plot,

$$\Omega_{\rm GW}(f;\mathbf{s}) = \Omega_p^{(*)} \cdot \begin{cases} \vdots \\ F_{-2}(f) & \text{if } f_{-2} \leq f < f_1 \\ F_{-1}(f) & \text{if } f_{-1} \leq f < f_0 \\ F_0(f) & \text{if } f_0 \leq f < f_1 \\ F_{+1}(f) & \text{if } f_1 \leq f < f_2 \\ F_{+2}(f) & \text{if } f_2 \leq f < f_3 \\ \vdots \end{cases} \quad \text{with} \quad \begin{array}{l} F_0(f) \equiv \left(\frac{f}{f_0}\right)^{\gamma_0}, \\ F_{-1}(f) \equiv F_0(f_0)\left(\frac{f}{f_0}\right)^{\gamma_{-1}}, \quad F_{-2}(f) \equiv F_{-1}(f_{-1})\left(\frac{f}{f_{-1}}\right)^{\gamma_{-2}}, \quad \dots \\ F_{+1}(f) \equiv F_0(f_1)\left(\frac{f}{f_1}\right)^{\gamma_1}, \quad F_{+2}(f) \equiv F_{+1}(f_2)\left(\frac{f}{f_2}\right)^{\gamma_2}, \quad \dots \end{array}$$

i.e., we need to sample from p$(\Omega_p(*),\dots, \gamma_{-1}, \gamma_0, \gamma_1,\dots, A_P, A_{acc} \mid \boldsymbol{D_0})$

- **30 parameters of interest:** 2 noise + 28 GW parameters
- **thousands of nuisance parameters**: G's
- **non-gaussian likelihood**

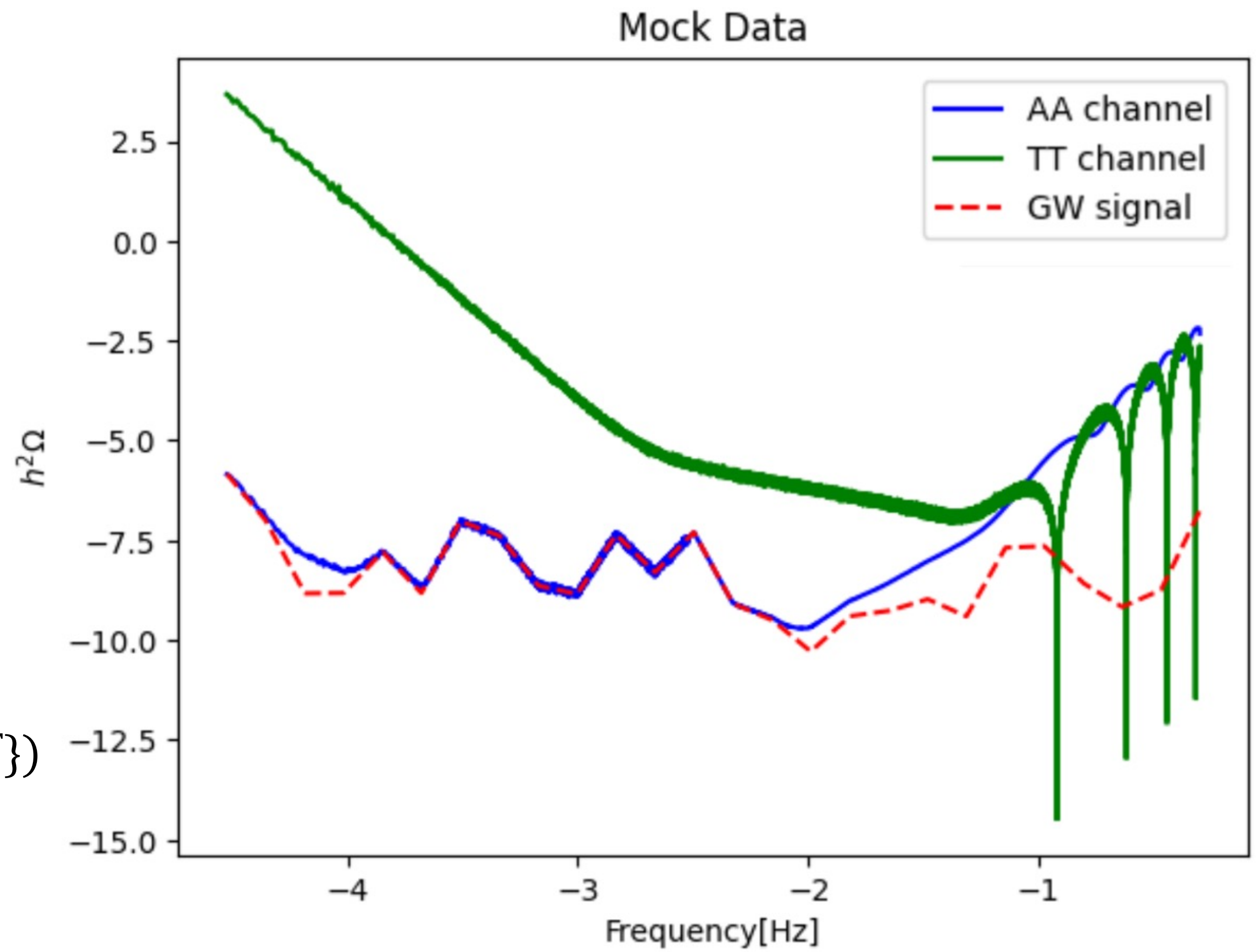
GWBackFinder with NPE

GWBackFinder pipeline

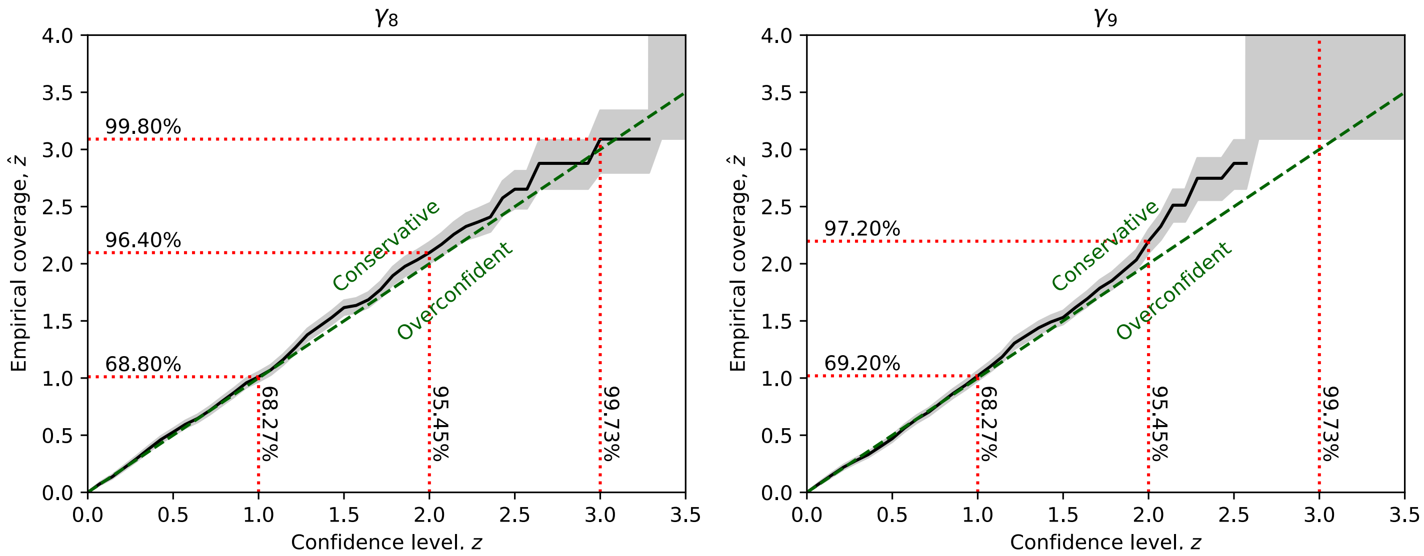
- Generate training data in the 3 channels by varying the parameters of interest ($\mathcal{O}(10^5)$ training data)

- Train a normalizing flow to approximate **instantly** the posterior $p(\Omega_p(*), \ldots, \gamma_{-1}, \gamma_0, \gamma_1, \ldots, A_P, A_{acc} \mid \{\boldsymbol{D}^{AA}, \boldsymbol{D}^{EE}, \boldsymbol{D}^{TT}\})$ of every possible observation $\{\boldsymbol{D}^{AA}, \boldsymbol{D}^{EE}, \boldsymbol{D}^{TT}\}$.

Amortized!



GWBackFinder pipeline



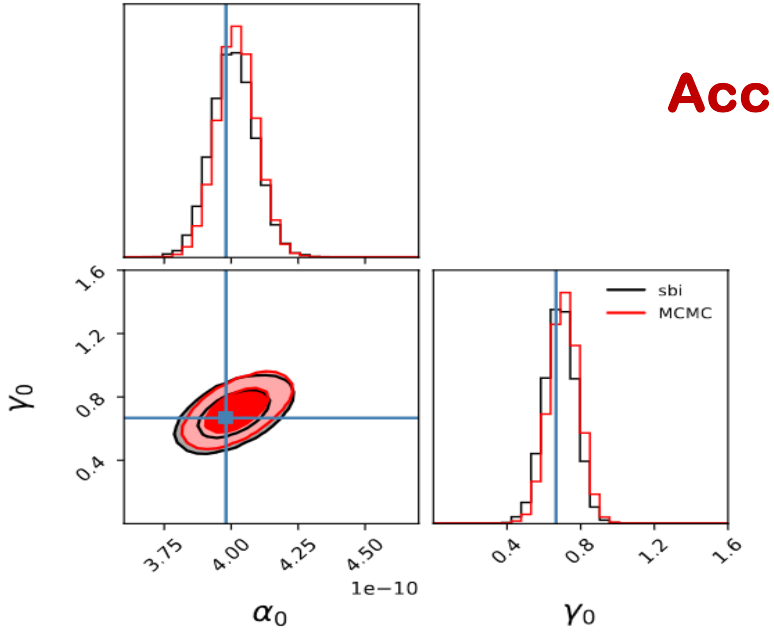
- Test coverage on mock data

- Compare analysis with MCMC for mock data using a non-gaussian likelihood:

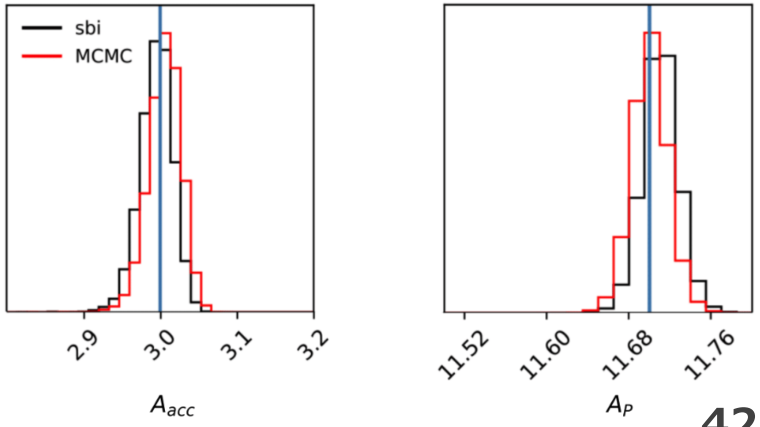
$$\ln \mathcal{L} = \frac{1}{3} \ln \mathcal{L}_G + \frac{2}{3} \ln \mathcal{L}_{LN}$$

$\mathcal{O}(10^5)$ likelihood evaluations to obtain the posterior of only one observation

slow & NOT Amortized !

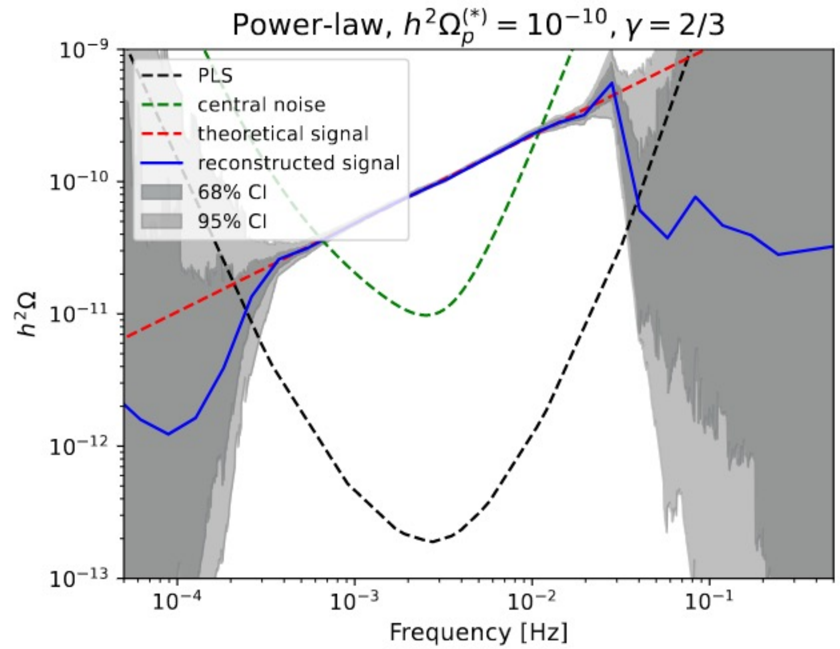


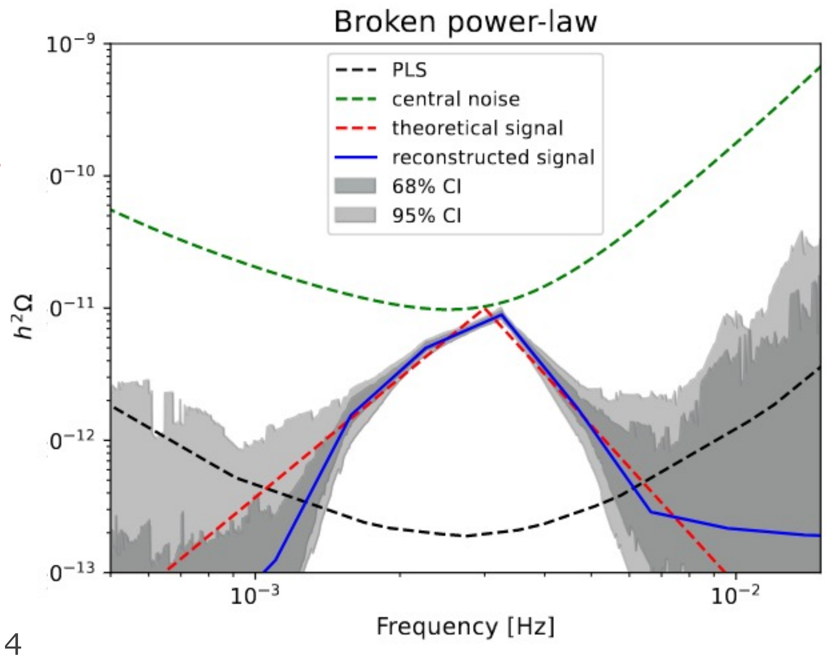
Accurate posteriors!

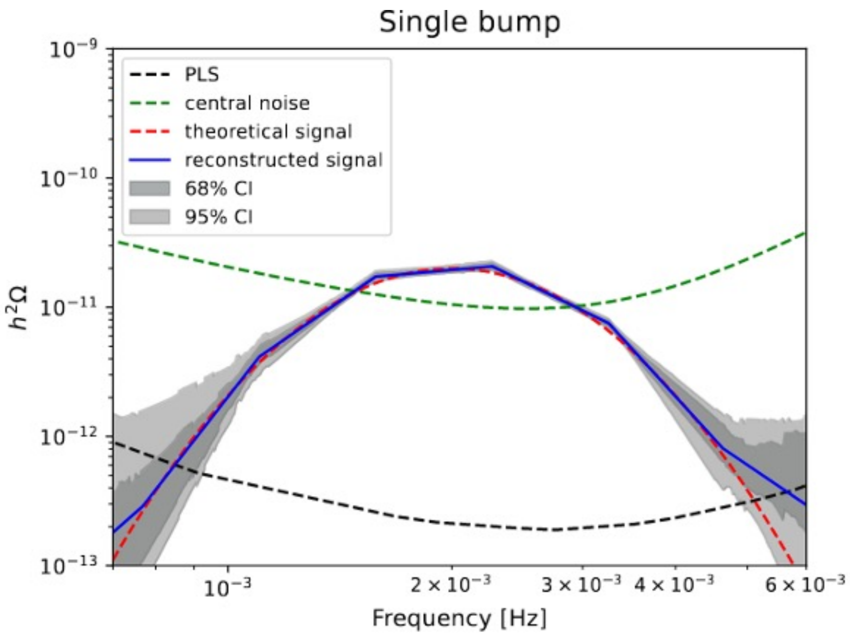


GWBackFinder pipeline

- Reconstruct **template** signals, e.g.,
  - ➢ single power-law
  - ➢ broken power-law
  - ➢ single bump signal







Accurate & fast reconstruction!

GWBackFinder pipeline

- Reconstruct **template** signals in the presence of **foregrounds**:

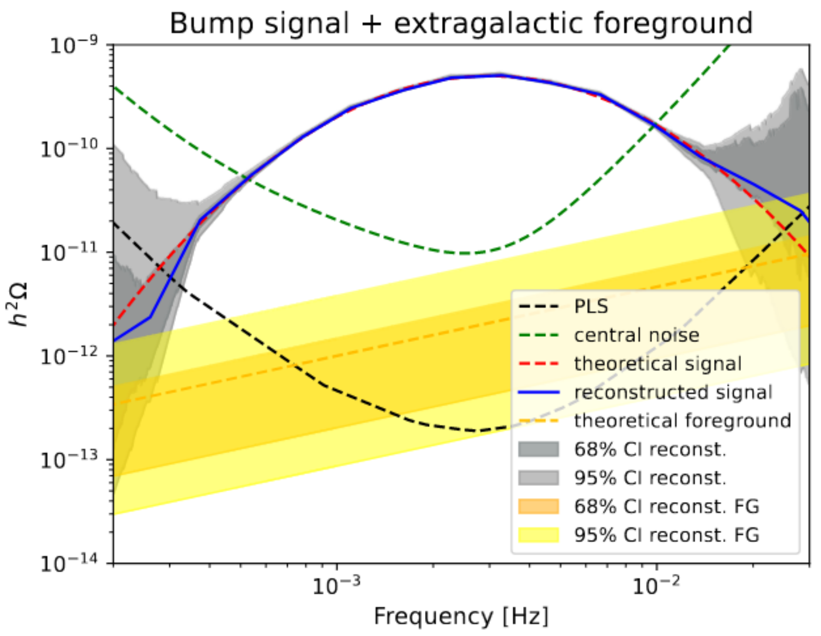
  ➢ Extragalactic:

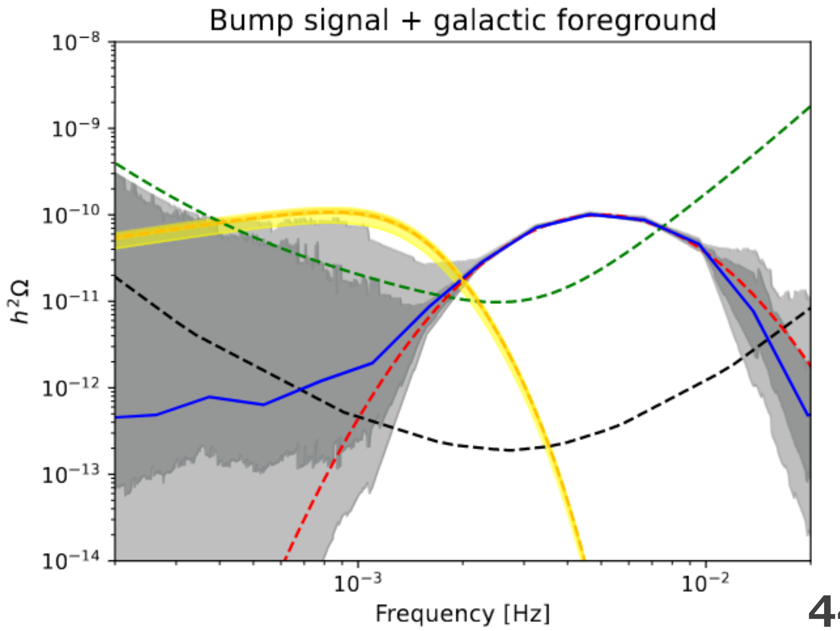
$$\Omega_{\rm EF}(f) = 10^{\alpha_{\rm EF}} \left(\frac{f}{0.001\,{\rm Hz}}\right)^{2/3}$$

  ➢ Galactic:

$$\Omega_{\rm GF}(f) = 10^{\alpha_{\rm GF}} f^{2/3} \exp\Big(-f^{a_1} - a_2 f \sin(a_3 f)\Big) \cdot \{1+\tanh[a_4(f_k - f)]\}$$

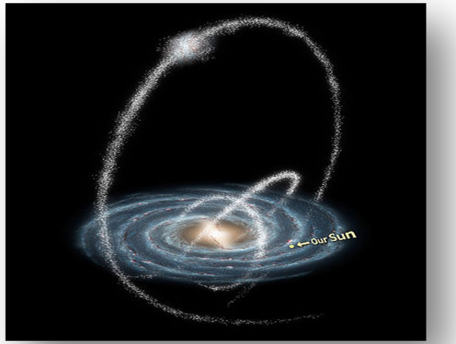
Accurate & fast reconstruction!





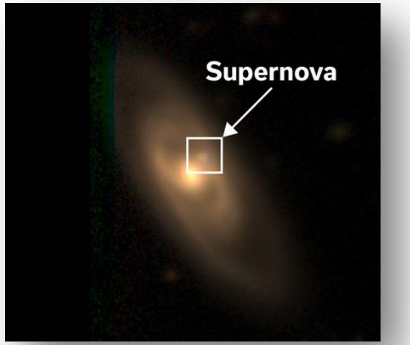
SBI appears to be broadly applicable

Stellar streams



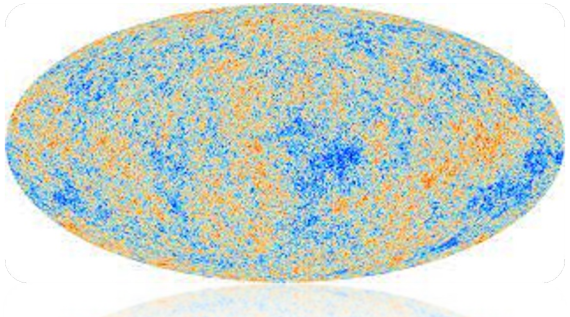
arxiv: 1911.01429

Supernova cosmology



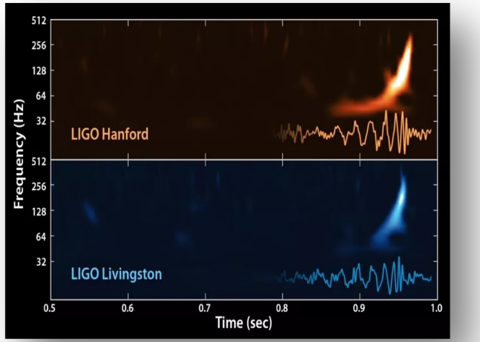
arxiv: 2209.06733

CMB physics



arxiv: 2111.08030

Gravitational waves



arxiv: 2308.06318

Strong Lensing

arxiv: 2211.04365

Conclusions

- Large forward models are omnipresent in physics/astronomy and pose challenges for analysing current and future data.

- Deep learning opens many new powerful ways for tackling these inference problems.

- There is a large variety of different methods and algorithms, both simulation- and likelihood-based, which differ in applicability, scaling and requirements for the forward model.

- There are many open research questions, both in physics/ astronomy and computer science.

Thank you!