Towards reconstructing the halo clustering and halo mass function of N-body simulations using neural ratio estimation

Androniki Dimitriou, Christoph Weniger, Camila Correa

arXiv:2206.11312

Dark Matters
1 December 2022, Brussels

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

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

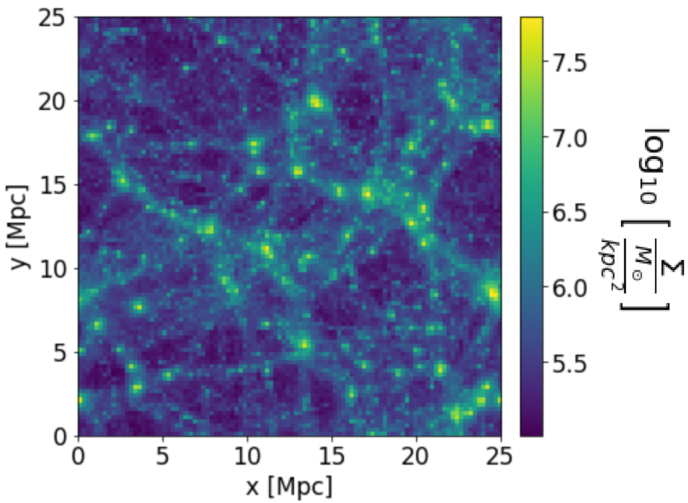
- **Approach:** direct calibration of **a analytical halo** model/simulator **based on a toy implementation of two body correlation functions** on DM-only **simulations**

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

- **Approach:** direct calibration of **a analytical halo** model/simulator **based on a toy implementation of two body correlation functions** on DM-only **simulations**

- **The EAGLE project**

(25 Mpc)^3 box with 376^3 particles

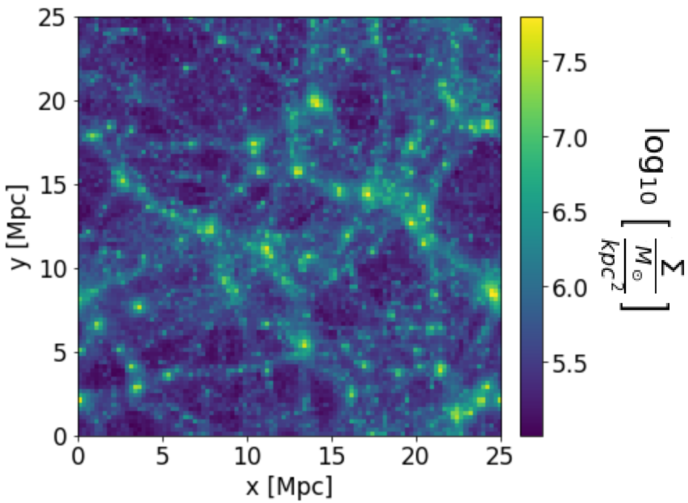


- **Goal:** reconstruction of **halo clustering** and **halo mass function** of DM-only cosmological simulations generated by the **EAGLE project**

- **Approach:** direct calibration of **a analytical halo** model/simulator **based on a toy implementation of two body correlation functions** on DM-only **simulations**

- **The EAGLE project**

(25 Mpc)^3 box with 376^3 particles



FOF halo finder

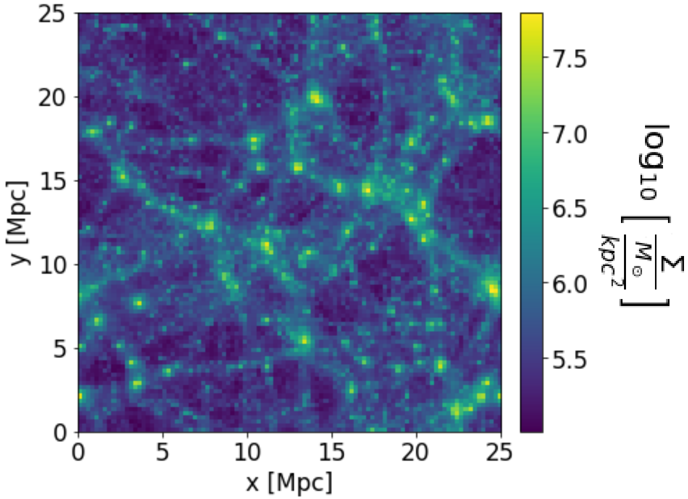
Particle data

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

- **Approach:** direct calibration of **a analytical halo** model/simulator **based on a toy implementation of two body correlation functions** on DM-only **simulations**

- **The EAGLE project**

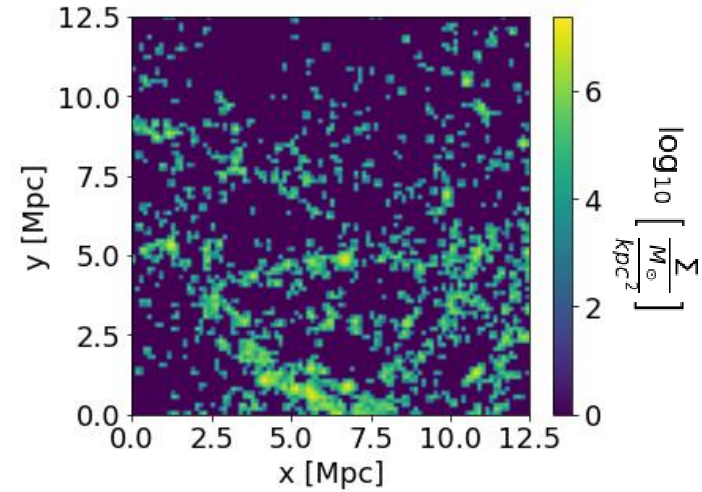
(25 Mpc)^3 box with 376^3 particles



FOF halo finder

Particle data

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

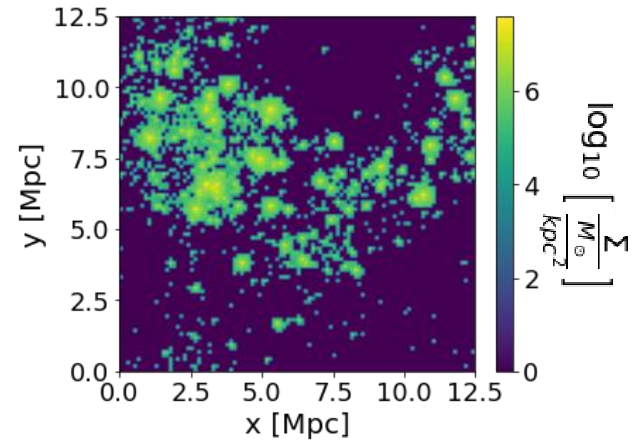


- **The toy halo model**

physical parameters

$\boldsymbol{\theta}$ =(N , α , ε , n)

Model/ simulator

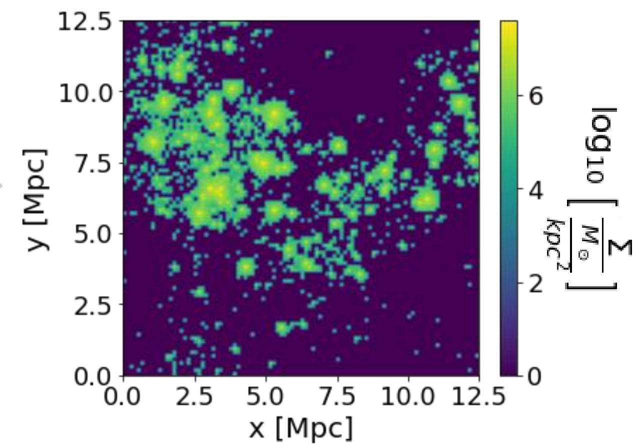


- **The toy halo model**

$\boldsymbol{\theta} = (N, a, \varepsilon, n)$

Number of DM haloes

Model/ simulator



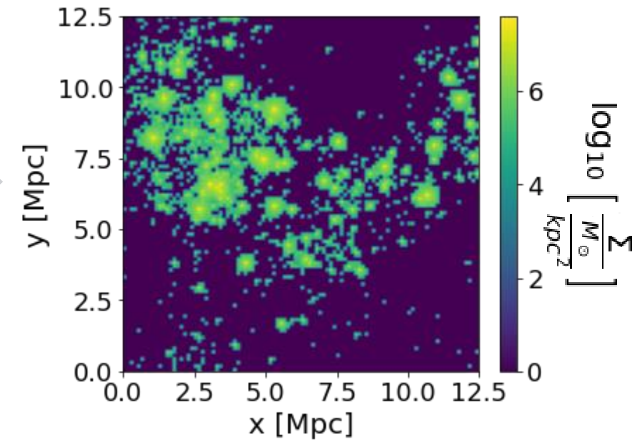
- **The toy halo model**

Slope of HMF: $\frac{dn}{dM} \propto M^{-a}$, where $a \cong 1.9$ for low masses

$\boldsymbol{\theta} = (N, a, \varepsilon, n)$

Number of DM haloes

Model/ simulator



- **The toy halo model**

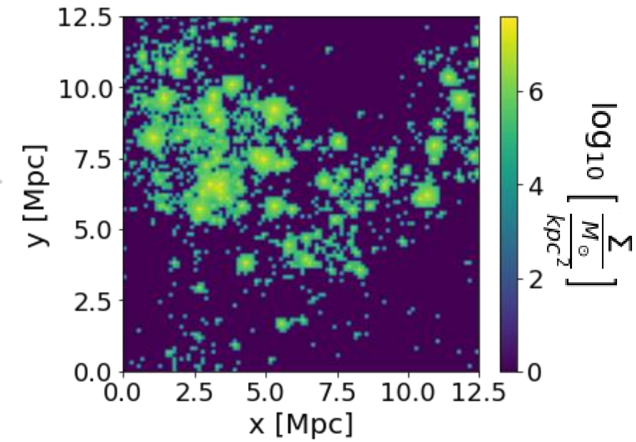
Slope of HMF: $\frac{dn}{dM} \propto M^{-a}$, where $a \cong 1.9$ for low masses

$\boldsymbol{\theta} = (N, a, \varepsilon, n)$

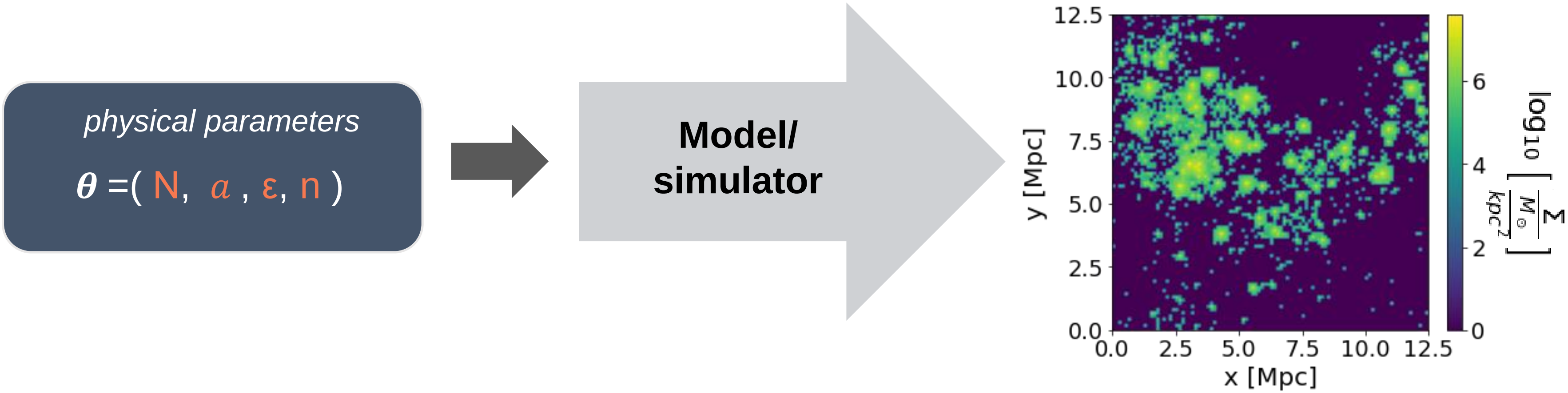
Number of DM haloes

Effective parameters of a clustering toy model

Model/ simulator



- **The toy halo model**



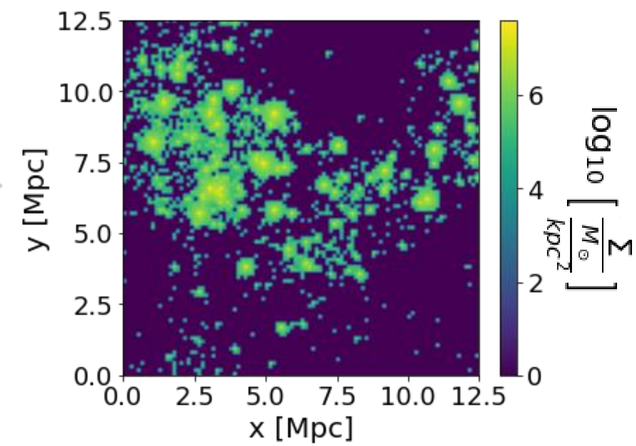
- Given an image, we want to know which parameters, ***θ***, generated it

- **The toy halo model**

physical parameters

$\boldsymbol{\theta}$ = (N, α , ε, n)

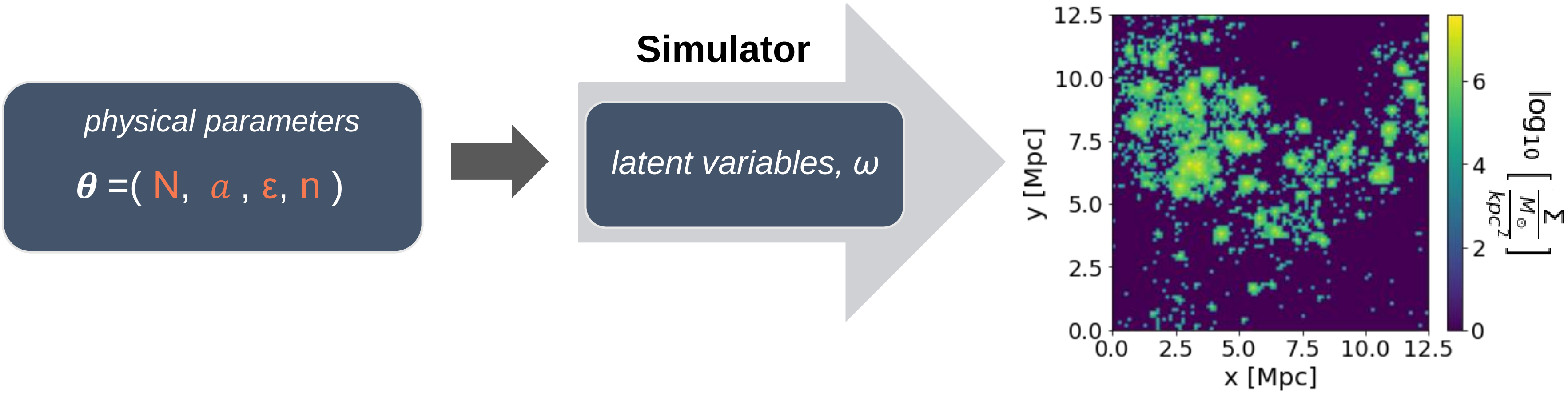
Model/ simulator



- Given an image, we want to know which parameters, $\boldsymbol{\theta}$, generated it

- **What is inside the simulator?**

- **The toy halo model**



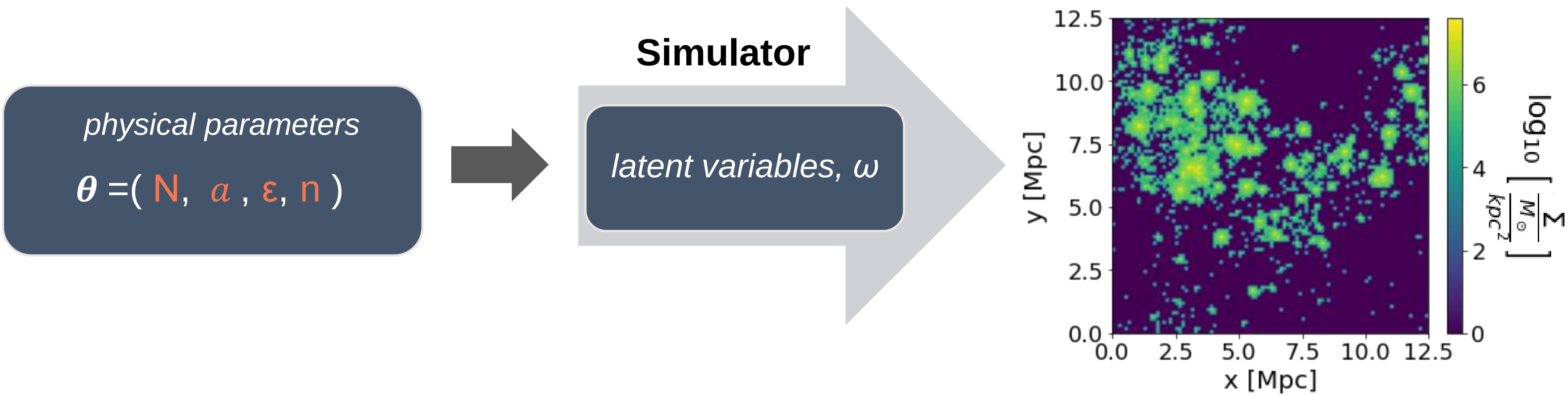
- Given an image, we want to know which parameters, $\boldsymbol{\theta}$, generated it

- **What is inside the simulator?**

latent variables

- **The toy halo model**

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



- Given an image, we want to know which parameters, $\boldsymbol{\theta}$, generated it

- **What is inside the simulator?**

latent variables

- Given some simulation $\boldsymbol{x}$, and a model with parameters $\boldsymbol{z}$, what can we learn about $\boldsymbol{z}$?

- Given some simulation $\boldsymbol{x}$, and a model with parameters $\boldsymbol{z}$, what can we learn about $\boldsymbol{z}$?

Bayes Theorem

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

- Posterior: $p(\boldsymbol{z}|\boldsymbol{x})$
- Likelihood: $p(\boldsymbol{x}|\boldsymbol{z})$
- Prior: $p(\boldsymbol{z})$
- Evidence: $p(\boldsymbol{x}) = \int d\boldsymbol{z}\;\; p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

- Given some simulation $\boldsymbol{x}$, and a model with parameters $\boldsymbol{z}$, what can we learn about $\boldsymbol{z}$?

Bayes Theorem

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

- Posterior: $p(\boldsymbol{z}|\boldsymbol{x})$
- Likelihood: $p(\boldsymbol{x}|\boldsymbol{z})$
- Prior: $p(\boldsymbol{z})$
- Evidence: $p(\boldsymbol{x}) = \int d\boldsymbol{z}\;\; p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

- Bayes theorem, gives the **joint posterior** for the full parameter space

- Given some simulation $\boldsymbol{x}$, and a model with parameters $\boldsymbol{z}$, what can we learn about $\boldsymbol{z}$?

Bayes Theorem

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

- Posterior: $p(\boldsymbol{z}|\boldsymbol{x})$
- Likelihood: $p(\boldsymbol{x}|\boldsymbol{z})$
- Prior: $p(\boldsymbol{z})$
- Evidence: $p(\boldsymbol{x}) = \int d\boldsymbol{z}\ \ p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

- Bayes theorem, gives the **joint posterior** for the full parameter space
- Almost always, we want to calculate **marginal posteriors** of the parameters of interest, $\boldsymbol{\vartheta}$

$$p(\boldsymbol{\vartheta}|\boldsymbol{x}) = \int d\boldsymbol{\eta}\, p(\boldsymbol{\eta}, \boldsymbol{\vartheta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})}{p(\boldsymbol{x})}$$

nuisance parameters

- Given some simulation $\boldsymbol{x}$, and a model with parameters $\boldsymbol{z}$, what can we learn about $\boldsymbol{z}$?

Bayes Theorem

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

- Posterior: $p(\boldsymbol{z}|\boldsymbol{x})$
- Likelihood: $p(\boldsymbol{x}|\boldsymbol{z})$
- Prior: $p(\boldsymbol{z})$
- Evidence: $p(\boldsymbol{x}) = \int d\boldsymbol{z}\ \ p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

- Bayes theorem, gives the **joint posterior** for the full parameter space
- Almost always, we want to calculate **marginal posteriors** of the parameters of interest, $\boldsymbol{\vartheta}$

$$p(\boldsymbol{\vartheta}|\boldsymbol{x}) = \int d\boldsymbol{\eta}\, p(\boldsymbol{\eta}, \boldsymbol{\vartheta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})}{p(\boldsymbol{x})}$$

nuisance parameters

Intractable due to high dimensionality

This problem appears in the context of likelihood-based inference methods, e.g., MCMC

Marginal Neural Ratio Estimation (MNRE)

arXiv: 2107.01214

Marginal Neural Ratio Estimation (MNRE)

arXiv: 2107.01214

Estimates posteriors through a binary classification problem:

"Given a (parameter: $\boldsymbol{\vartheta}$, image: $\boldsymbol{x}$) pair, is the image, $\boldsymbol{x}$, actually generated by the parameter $\boldsymbol{\vartheta}$?"

Class 1:	**Class 2:**
$(\boldsymbol{x}, \boldsymbol{\vartheta}) \sim p(\boldsymbol{x}\|\boldsymbol{\vartheta})\, p(\boldsymbol{\vartheta})$	$(\boldsymbol{x}, \boldsymbol{\vartheta}') \sim p(\boldsymbol{x})p(\boldsymbol{\vartheta})$

Marginal Neural Ratio Estimation (MNRE)

arXiv: 2107.01214

Estimates posteriors through a binary classification problem:

“Given a (parameter: $\boldsymbol{\vartheta}$, image: $\boldsymbol{x}$) pair, is the image, $\boldsymbol{x}$, actually generated by the parameter $\boldsymbol{\vartheta}$?”

Class 1:	**Class 2:**
$(\boldsymbol{x}, \boldsymbol{\vartheta}) \sim p(\boldsymbol{x}\|\boldsymbol{\vartheta})\, p(\boldsymbol{\vartheta})$	$(\boldsymbol{x}, \boldsymbol{\vartheta}') \sim p(\boldsymbol{x})p(\boldsymbol{\vartheta})$

Train a classifier with mock data to directly estimate:

$$r(\boldsymbol{x}, \boldsymbol{\vartheta}) \cong \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})}{p(\boldsymbol{x})}$$

Marginal Neural Ratio Estimation (MNRE)

arXiv: 2107.01214

Estimates posteriors through a binary classification problem:

"Given a (parameter: $\boldsymbol{\vartheta}$, image: $\boldsymbol{x}$) pair, is the image, $\boldsymbol{x}$, actually generated by the parameter $\boldsymbol{\vartheta}$?"

Class 1:

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

Class 2:

$(\boldsymbol{x}, \boldsymbol{\vartheta}') \sim p(\boldsymbol{x})p(\boldsymbol{\vartheta})$

Train a classifier with mock data to directly estimate:

$$r(\boldsymbol{x}, \boldsymbol{\vartheta}) \cong \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})}{p(\boldsymbol{x})} \overset{\text{bayes theorem}}{=} \frac{p(\boldsymbol{\vartheta}|\boldsymbol{x})}{p(\boldsymbol{\vartheta})}$$

Marginal Neural Ratio Estimation (MNRE)

arXiv: 2107.01214

Estimates posteriors through a binary classification problem:

"Given a (parameter: $\boldsymbol{\vartheta}$, image: $\boldsymbol{x}$) pair, is the image, $\boldsymbol{x}$, actually generated by the parameter $\boldsymbol{\vartheta}$?"

Class 1:	**Class 2:**
$(\boldsymbol{x}, \boldsymbol{\vartheta}) \sim p(\boldsymbol{x}\|\boldsymbol{\vartheta})\, p(\boldsymbol{\vartheta})$	$(\boldsymbol{x}, \boldsymbol{\vartheta}') \sim p(\boldsymbol{x}) p(\boldsymbol{\vartheta})$

Train a classifier with mock data to directly estimate:

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

Once we have trained the network, we can **estimate the posterior**:

$$p(\boldsymbol{\vartheta}|\vec{\boldsymbol{x}}) = r(\boldsymbol{x}, \boldsymbol{\vartheta}) p(\boldsymbol{\vartheta})$$

Training with `swyft`

arXiv:2107.01214

Training with `swyft`

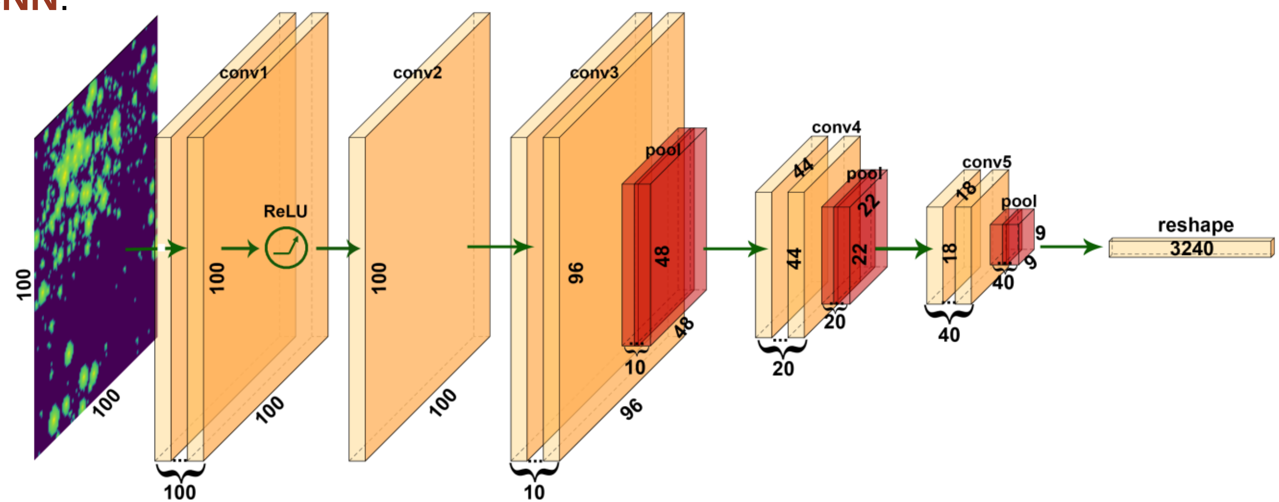
arXiv:2107.01214

- **Physical parameters:**
  - **N**: Number of halos, where N ϵ (100, 2100)
  - **a** : Inner slope of the halo mass function, where a ϵ (1, 3)
  - **ε**: Exponent of the density field, where ε ϵ (0,2)
  - **n**: Slope of the power spectrum, where n ϵ (0,10)

Training with `swyft`

arXiv:2107.01214

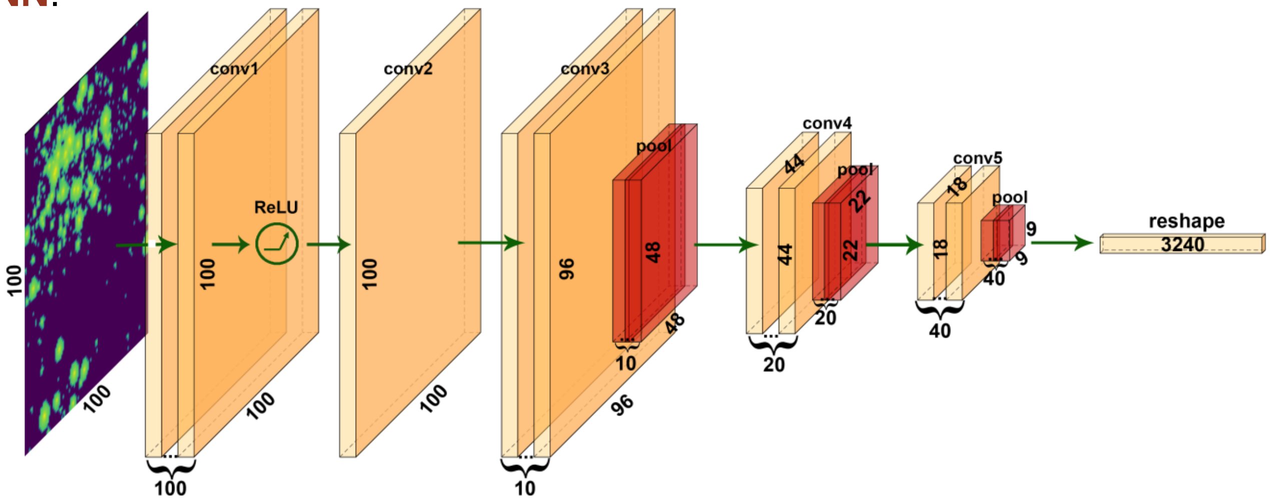
- **Physical parameters:**
  - **N**: Number of halos, where N ϵ (100, 2100)
  - **a** : Inner slope of the halo mass function, where a ϵ (1, 3)
  - **ε**: Exponent of the density field, where ε ϵ (0,2)
  - **n**: Slope of the power spectrum, where n ϵ (0,10)
- We define a **CNN**:



Training with `swyft`

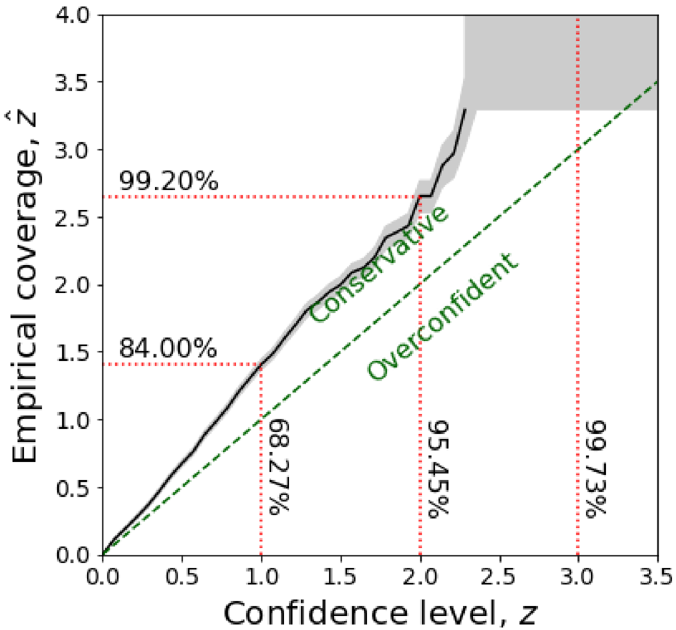
arXiv:2107.01214

- **Physical parameters:**
  - **N**: Number of halos, where N ϵ (100, 2100)
  - **a** : Inner slope of the halo mass function, where a ϵ (1, 3)
  - **ε**: Exponent of the density field, where ε ϵ (0,2)
  - **n**: Slope of the power spectrum, where n ϵ (0,10)
- We define a **CNN**:

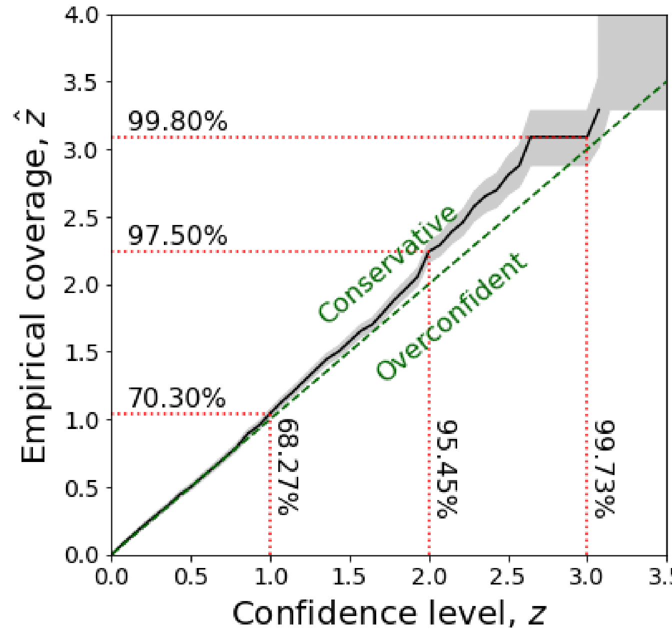


- We **train** using 200.000 mock images

Results on mock data



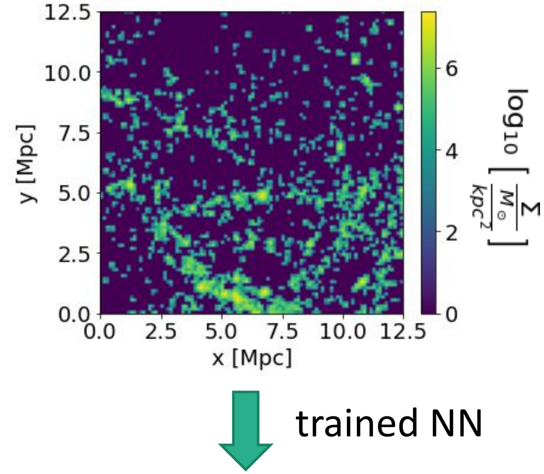
number of halos (N)



slope of power spectrum (n)

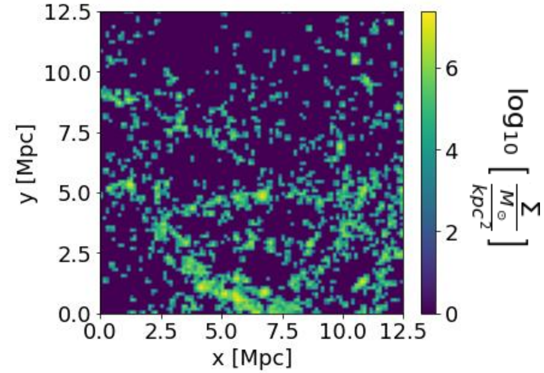
Results on actual N body simulations

- one simulation box



Results on actual N body simulations

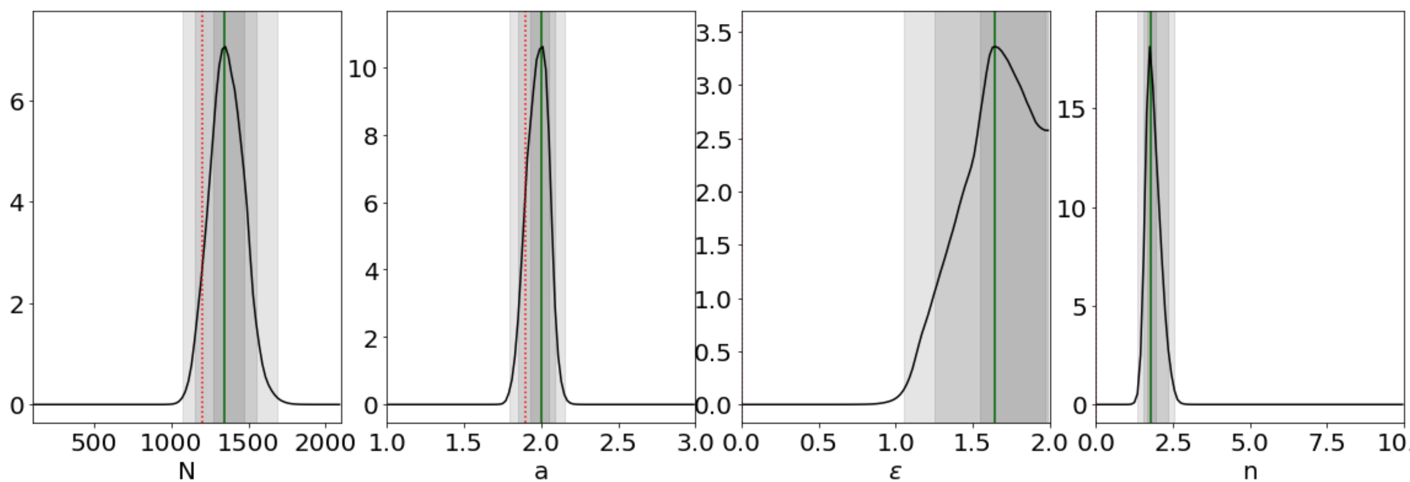
- one simulation box



trained NN

mode

correct value

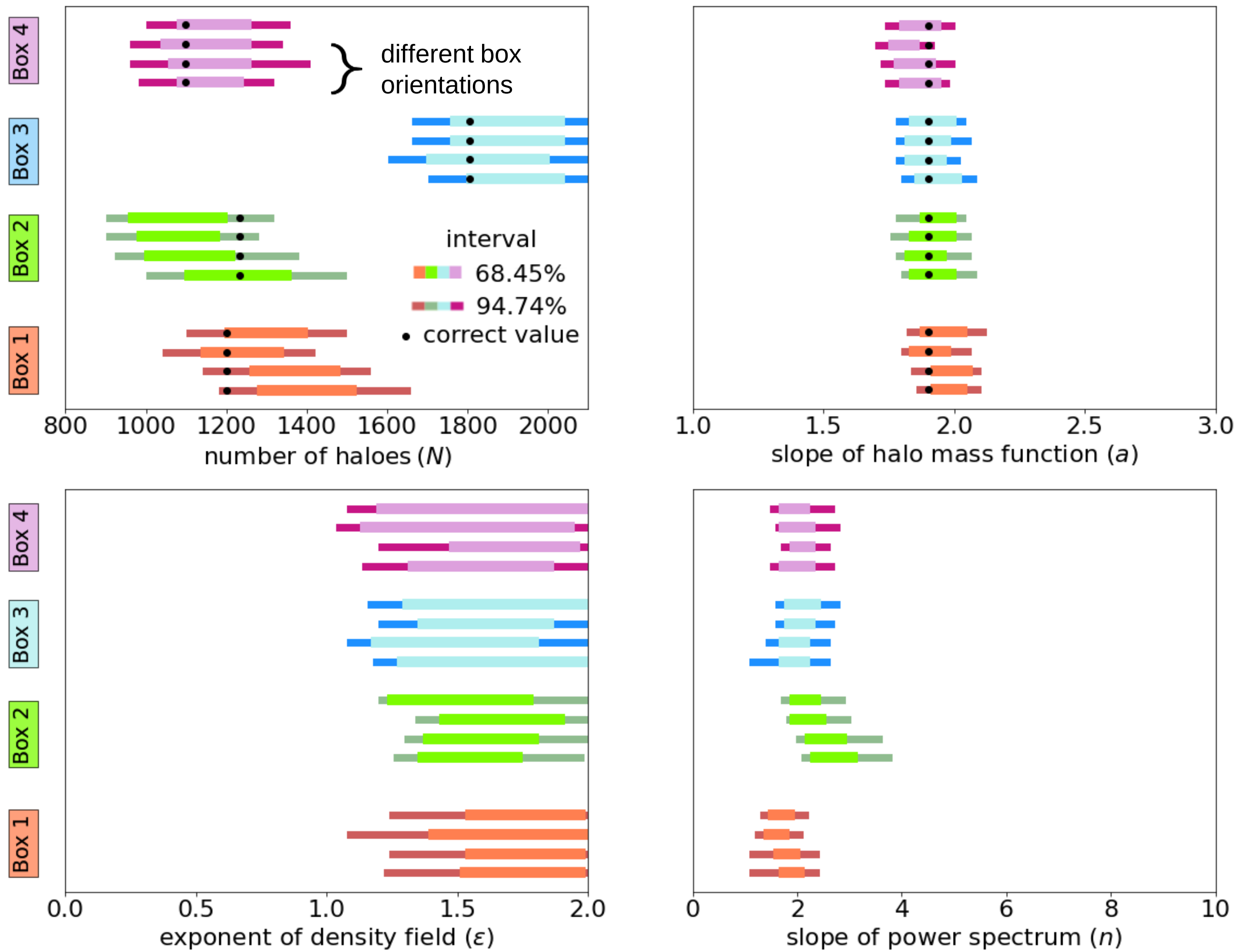


1D Posteriors

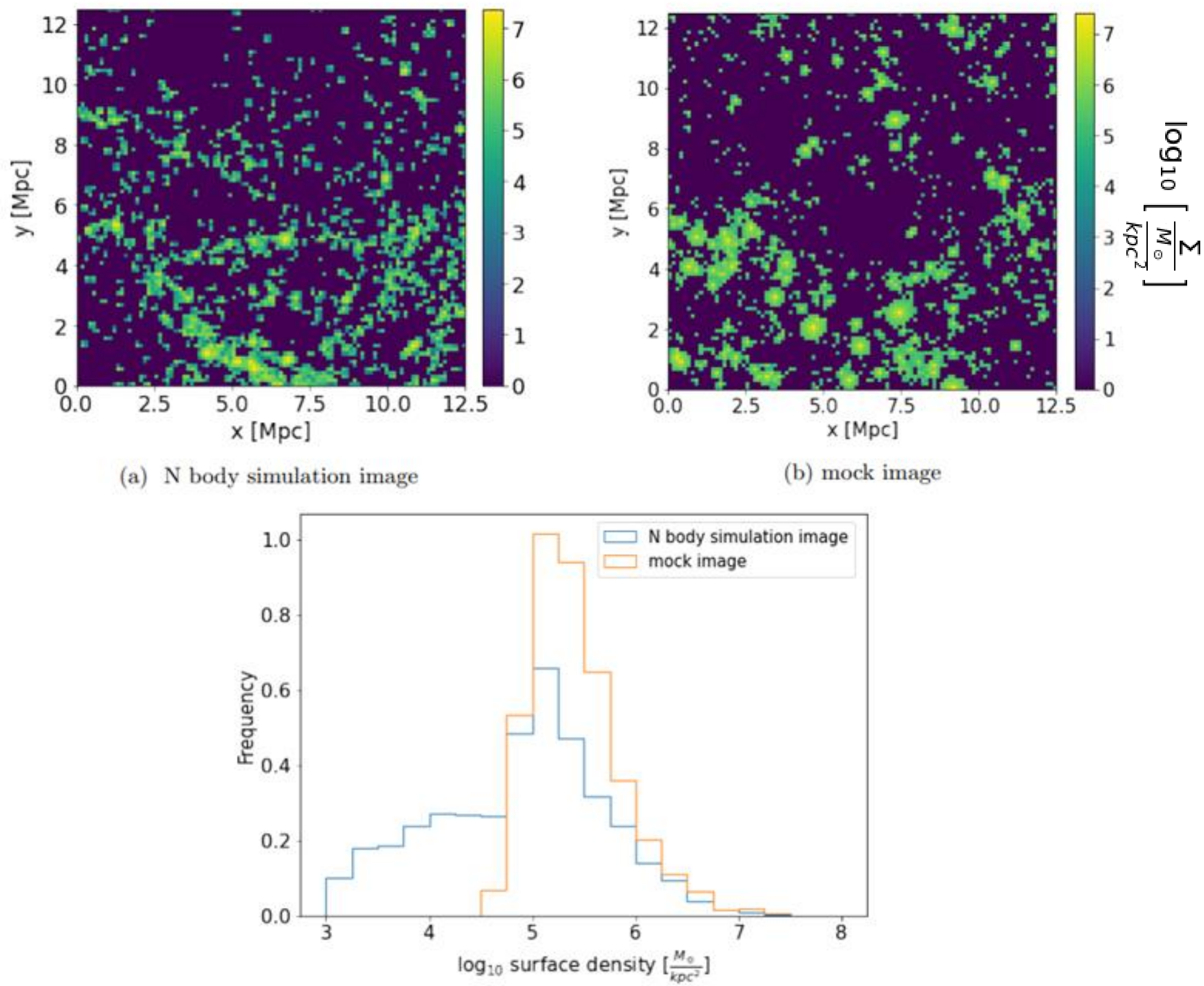
Results on actual N body simulations

- 4 simulation boxes and 3 rotations of them

Accurate marginal posteriors!



Comparison of a N-body simulation and a mock image



(a) N body simulation image

(b) mock image

- **Using a toy halo model and MNRE**
    - we reconstructed the halo mass function
    - we generated images similar to DM-only N-body simulations

- **Using a toy halo model and MNRE**
    - we reconstructed the halo mass function
    - we generated images similar to DM-only N-body simulations

- **Long-term goal**

a analytical model for haloes, subhaloes, clustering and baryonic matter that generates actual N-body simulations

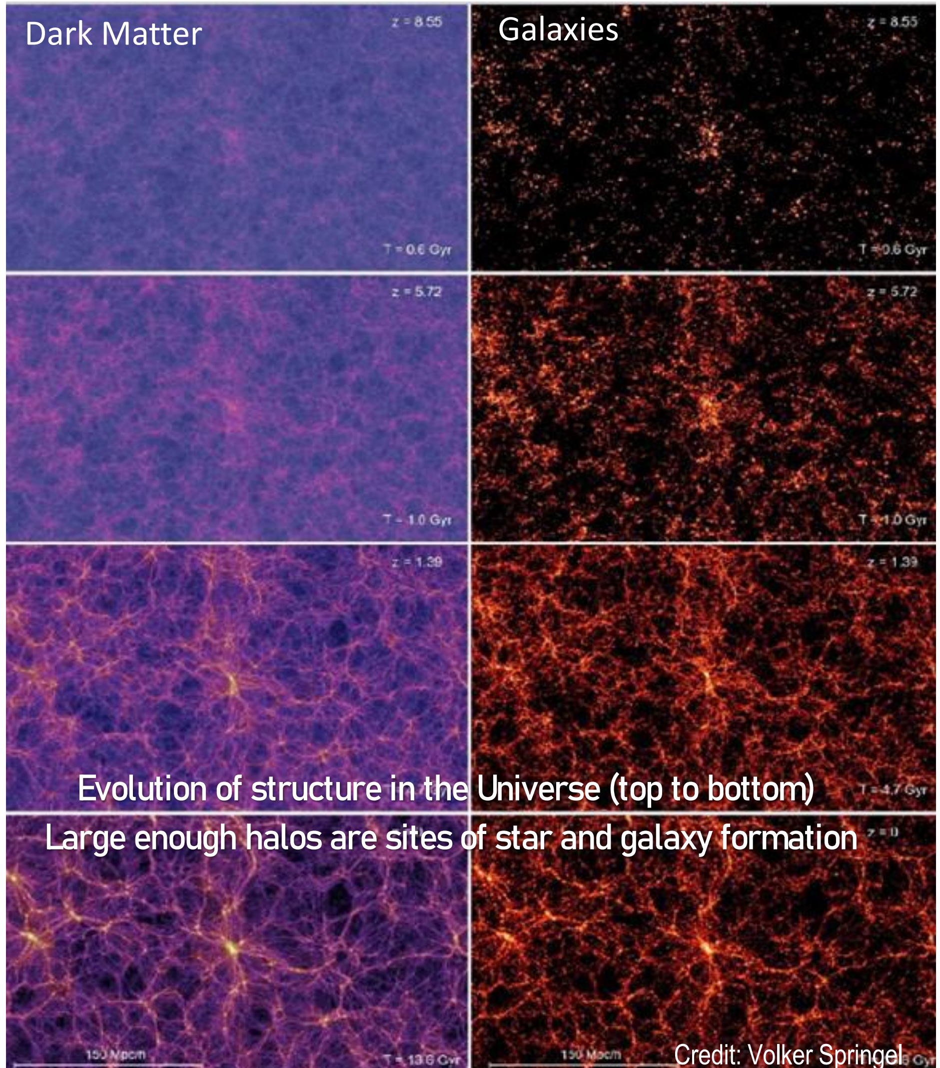
- **Using a toy halo model and MNRE**
  - we reconstructed the halo mass function
  - we generated images similar to DM-only N-body simulations

- **Long-term goal**

a analytical model for haloes, subhaloes, clustering and baryonic matter
that generates actual N-body simulations

Thank you!

Backup slides



Evolution of structure in the Universe (top to bottom)

Large enough halos are sites of star and galaxy formation

Identifying Dark Matter Haloes

FOF Halo Finders

Connect particles that are close to each other

↓

Haloes with arbitrary shapes, i.e., it's difficult to assign a mass to them

↓

Define a boundary: the virial radius r_h within which the mean internal matter density is: $\rho_h = 200\rho_c^0$

↓

The mass of the halo, M_h, is defined as the total mass contained within the radius r_h

The Eagle Project

- **Using the FOF halo finder:**
  - Read halo masses.
  - Pick haloes with masses that belong to a specific mass range, e.g. $(10^9, 10^{12})\ M_{\odot}$.
  - Define sub volumes of the 25 Mpc^3 box, e.g., 12.5 Mpc^3, 12.5x 12.5x 5 Mpc^3 boxes.
  - Read the centers of potential (x, y, z coordinates) of those haloes.
  - Pick the haloes that belong to those boxes and count their number.

The Eagle Project

- **Using the Particle data:**

  - Read particle group numbers
  - Select particles that belong to the haloes that we picked before from the FOF halo finder
  - Read the coordinates of those particles (x,y,z coordinates)
  - Construct a heatmap with 100 bins by projecting the z direction
  - Multiply the counts of the histogram with the mass of the particles: $1.15 \cdot 10^7 M_\odot$ and divide with xedge · yedge of the histogram i.e., $125 \cdot 10^2\ \mathrm{kpc}^2$

The Model

- The **first physical parameter** 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: $\frac{dn}{dM} = bM^a$.

- The **second physical parameter** is the slope, a, of the halo mass function, while:

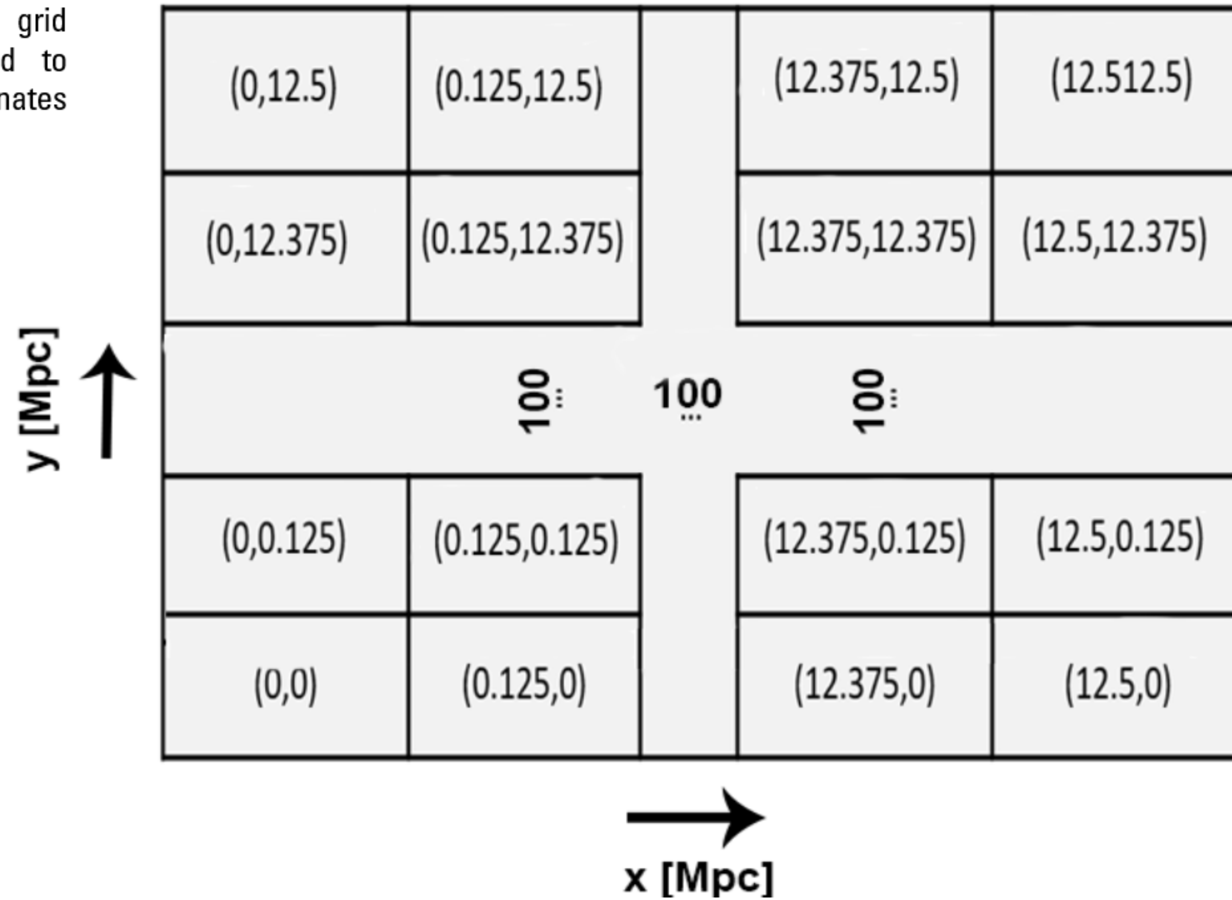
$$b = (1-a) \cdot \frac{N}{\left((10^{12})^{1-a} - (10^9)^{1-a}\right)V}$$

- From the masses of the haloes, we can calculate their concentrations c:

$$\log_{10} c = 1.4986 - 0.02499 \log_{10}(M/M\odot)[1 + 0.00565(\log_{10}(M/M\odot)^2)]$$ (Correa et al., 2015b)

- Now, we want to place the haloes in the 2D sky:

We construct a 100x100 grid whose values correspond to pairs of x and y coordinates where (x,y) ∈ (0,12.5) Mpc.



- Now, we want to place the haloes in the 2D sky:

  - ➢ We construct a 100x100 grid whose values correspond to pairs of x and y coordinates where $(x, y) \in (0,12.5)$ Mpc.

Adding Clustering to the Model

- Will sample the positions according to distributions generated from 2D realizations of gaussian random fields on an 100x100 grid.

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

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

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

Constructing the Realizations of the Gaussian Fields

- We generate position space realization of a white noise field, φ_{ab}, with unit amplitude, on a 100x100 grid, i.e., a,b ∈ {0,..,99}.

- We Fourier transform the white noise realization: $\varphi_{ab} \rightarrow \varphi_{k_a k_b}$, where $k_a, k_b \in \frac{2\pi}{N}$ {0,..,99}.

- We want to multiply $\varphi_{k_a k_b}$ with $\sqrt{P(k)}$ to get $\delta_{k_a k_b}$.

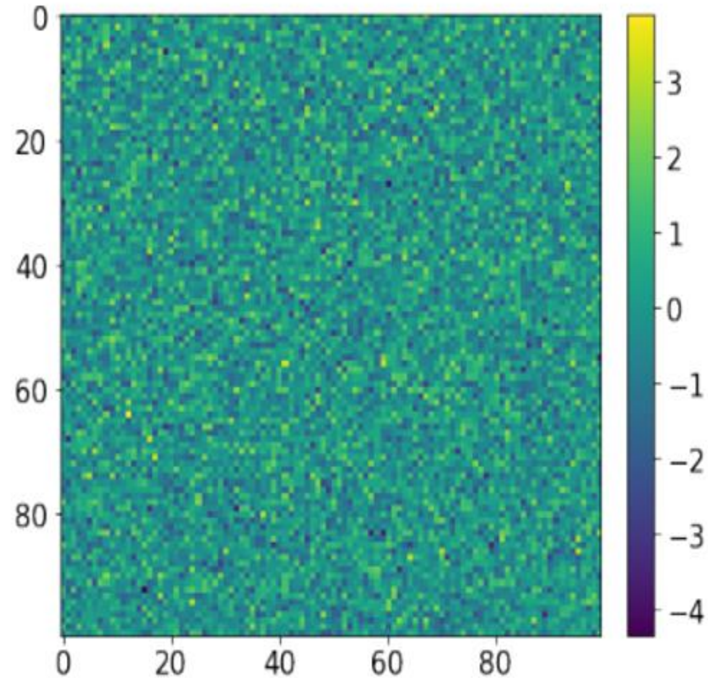
- Naive way: Calculate P(k) at points $k = \sqrt{k_a^2 + k_b^2}$ ➡ leads to imaginary fields

- Alternatively: Calculate P(k) at points $k = \sqrt{k_a'^2 + k_b'^2}$, where $k'_a, k'_b \in \frac{2\pi}{N}$ {0,..,50,−49, ... , −1}.

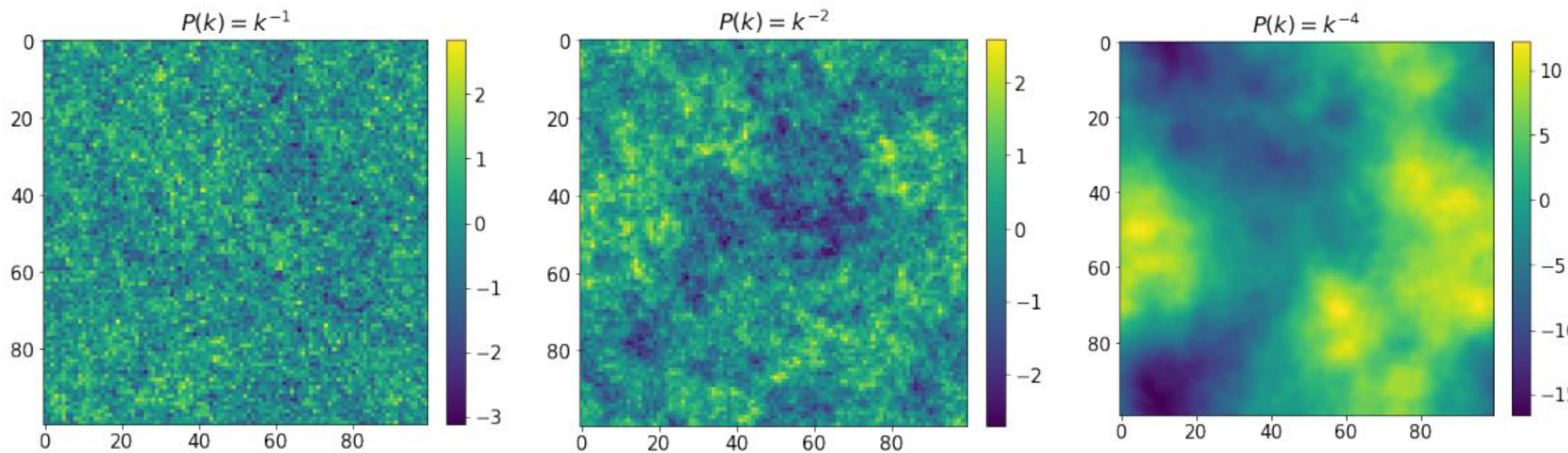
- $\delta_{k_a k_b} = \sqrt{P(k)}\,\varphi_{k_a k_b}$

- $\delta_{k_a k_b} \rightarrow \delta_{ab}$

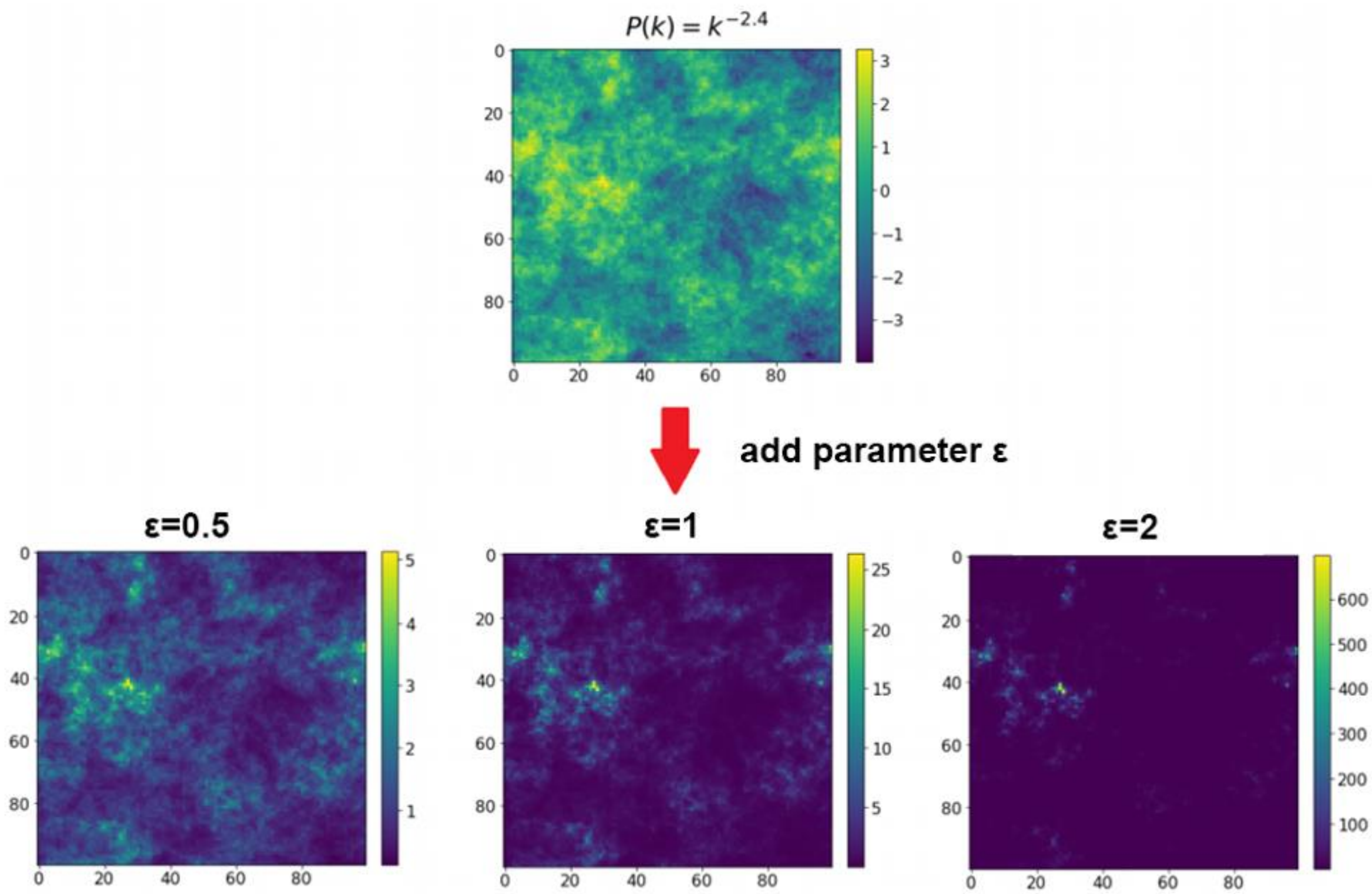
Realization of White Noise Field

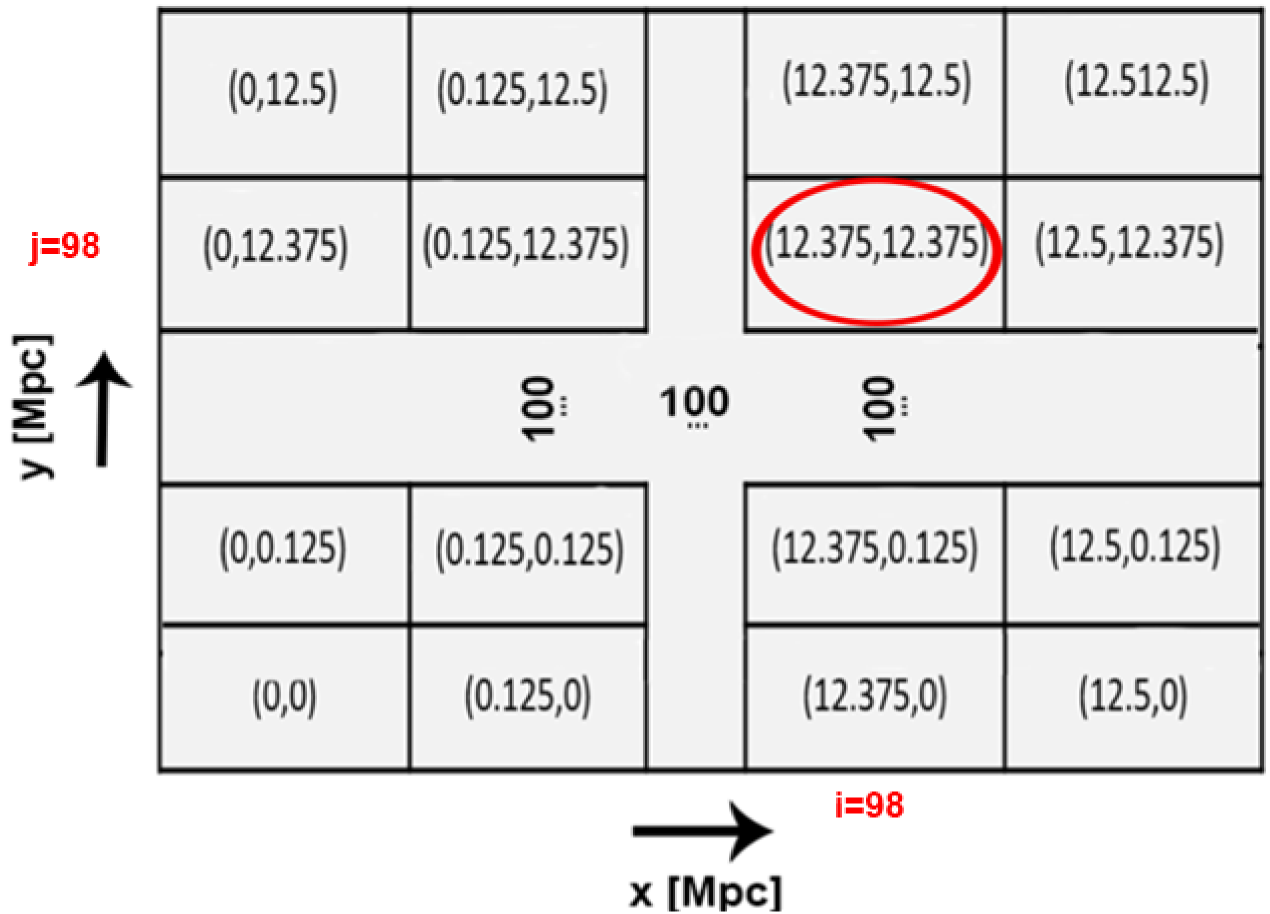


Realizations of the Gaussian Fields



The effect of parameter ε

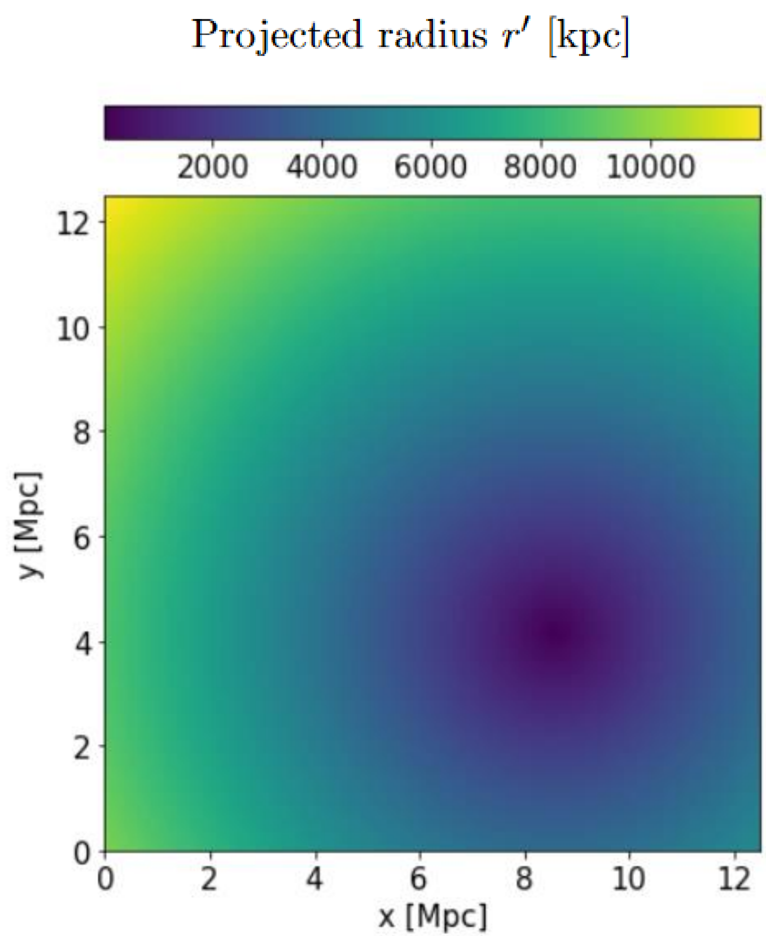




(0,12.5)
(0.125,12.5)
(12.375,12.5)
(12.512.5)
j=98
(0,12.375)
(0.125,12.375)
(12.375,12.375)
(12.5,12.375)
y [Mpc]
100
100
100
(0,0.125)
(0.125,0.125)
(12.375,0.125)
(12.5,0.125)
(0,0)
(0.125,0)
(12.375,0)
(12.5,0)
i=98
x [Mpc]

- Now, we want to place the haloes in the 2D sky:

  - ➢ We construct a 100x100 grid whose values correspond to pairs of x and y coordinates where $(x, y) \in (0,12.5)$ Mpc.
  - ➢ For each one of the N haloes, we sample its coordinates X ,Y coordinates from distributions generated from 2D realizations of gaussian random fields.
  - ➢ We subtract the coordinates of each halo (as pairs of X,Y values) from the values x, y of the grid and we end up with N grids.
  - ➢ For each grid, we calculate the root sum square of the two values in each one of its cells, i.e., the projected radius of the halo.


Projected radius r' [kpc]
2000
4000
6000
8000
10000
y [Mpc]
x [Mpc]
0
2
4
6
8
10
12

- Now, we want to place the haloes uniformly in the 2D sky:

➢ For each halo we calculate the surface density:

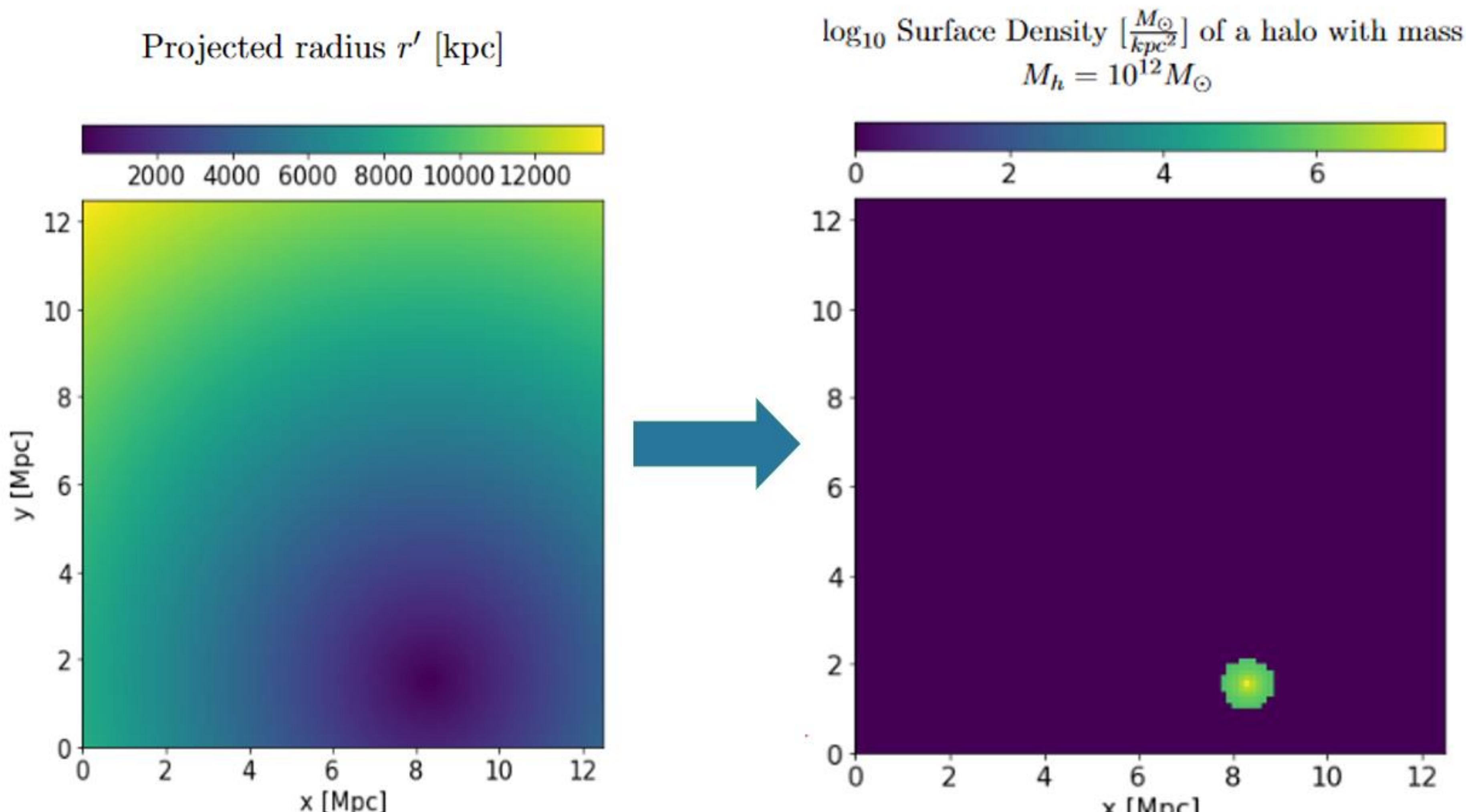
$$f(r') = \frac{2r_s\left[r'\cdot(r_s - r'^2) + 2r_s\cdot r'\cdot \arctan\left(\frac{r'\cdot\sqrt{r'-r_s}}{r'\cdot\sqrt{r'+r_s}}\right)\cdot\sqrt{r'^2 - r_s^2}\right]}{r'\cdot(r_s^4 - 2r_s^2 r'^2 + r'^4)}$$

where: $$r_s = \frac{(3M_h)^{1/3}}{(800\pi\rho_c^0)^{1/3}c}$$

➢ We set the values of the pixels that correspond to $r'>2.7r_h$ equal to 1.

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

➢ We add poisson noise to the final image.



Projected radius r' [kpc]
2000 4000 6000 8000 10000 12000
y [Mpc]
x [Mpc]
$\log_{10}$ Surface Density $[\frac{M_\odot}{kpc^2}]$ of a halo with mass $M_h = 10^{12} M_\odot$
0 2 4 6
y [Mpc]
x [Mpc]

Joint vs marginal samples

1) Examples for H_0, jointly sampled from $(\boldsymbol{x},\boldsymbol{\vartheta}) \sim p(\boldsymbol{x}|\boldsymbol{\vartheta})\, p(\boldsymbol{\vartheta})$

2) Examples for H_1, marginally sampled from $(\boldsymbol{x},\boldsymbol{\vartheta}) \sim p(\boldsymbol{x})\, p(\boldsymbol{\vartheta})$

Data: $\boldsymbol{x}$=Image, Label: $\boldsymbol{\vartheta} \in$ {Cat, Donkey}

Credit: C. Weniger

Loss function

- **Strategy:** We train a neural network $d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta}) \in [0,1]$ as binary classifier to estimate the probability of hypothesis H_0 or H_1. The Network output can be interpreted, for a given input pair $\boldsymbol{x}$ and $\boldsymbol{\vartheta}$, as probability that H_0 is true.

  - H_0 is true: $d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta}) \simeq 1$
  - H_1 is true: $d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta}) \simeq 0$

- The corresponding loss function is the so-called "binary cross-entropy":

$$L[d(\boldsymbol{x},\boldsymbol{\vartheta})] = -\int d\boldsymbol{x} d\boldsymbol{\theta} \, [p(\boldsymbol{x},\boldsymbol{\vartheta}) \ln(d(\mathbf{x},\boldsymbol{\vartheta})) + p(\boldsymbol{x})p(\boldsymbol{\vartheta}) \ln(1 - d(\boldsymbol{x},\boldsymbol{\vartheta}))]$$

- Minimizing that function w.r.t the network parameters φ yields:

$$d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta}) \approx \frac{p(\boldsymbol{x},\boldsymbol{\vartheta})}{p(\boldsymbol{x},\boldsymbol{\vartheta}) + p(\boldsymbol{x})p(\boldsymbol{\vartheta})}$$

Credit: C. Weniger

Likelihood-to-evidence ratio

Training binary classification networks yield true Bayesian posterior estimates!

- With a bit of math one can show that:

$$r(\boldsymbol{x},\boldsymbol{\vartheta}) \equiv \frac{d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta})}{d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta})-1} \approx \frac{p(\boldsymbol{x}|\boldsymbol{\vartheta})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{\vartheta}|\boldsymbol{x})}{p(\boldsymbol{\vartheta})}$$

- Once we have trained the network $d_\varphi(\boldsymbol{x},\boldsymbol{\vartheta})$, we can **estimate the posterior**:

$$p(\boldsymbol{\vartheta}|\boldsymbol{x}) \simeq r(\boldsymbol{x},\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})$$

- Swyft: a flexible and powerful tool for efficient marginal posterior estimation using NN, designed by B. Miller et al. (2020, 2021).

Credit: C. Weniger

- **Until now, using our model and Swyft we were able:**
  - to reconstruct the halo mass function
  - and to produce images like those of N body simulations!

- As a **next step** we can test if we can identify the lowest mass of the haloes in these images.

To do that:

- Instead of sampling haloes with masses: $M_h \in (10^9, 10^{12})\ M_\odot$,
  we will sample masses: $M_h \in (10^c, 10^{12})\ M_\odot$,
- where c is **a new parameter** of our model.
- We set the values of the parameters a, ε and n equal to the modes of their combined posteriors.

Results on actual N body simulations