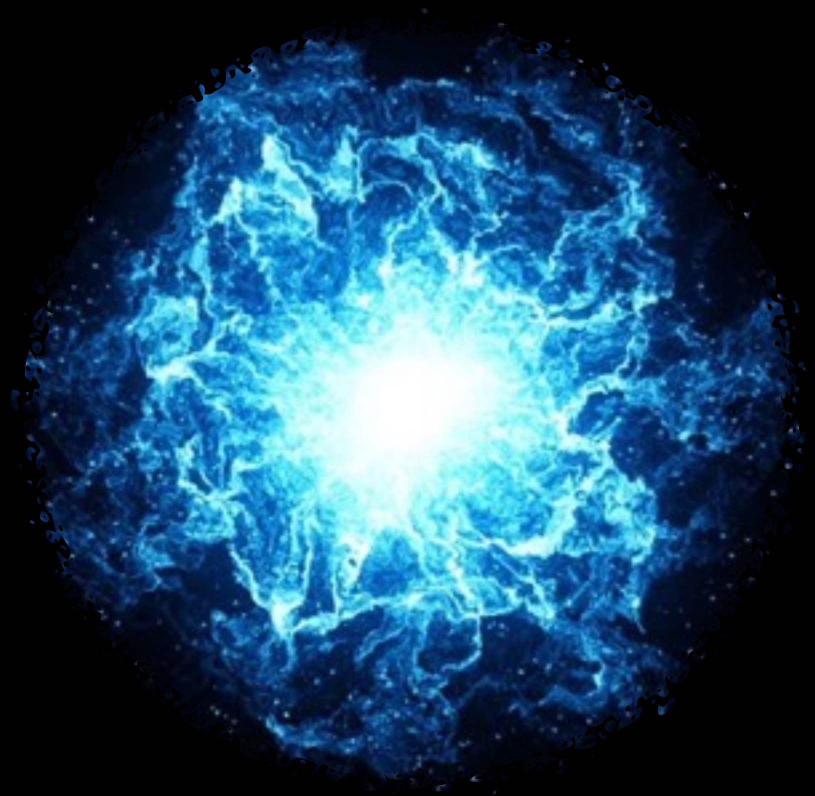
# Hierarchical modelling of galaxy clusters for Cosmology

Maggie Lieu, Will Farr, Michael Betancourt, Graham Smith, Mauro Sereno, Ian McCarthy, Paul Giles, Justin Alsing.

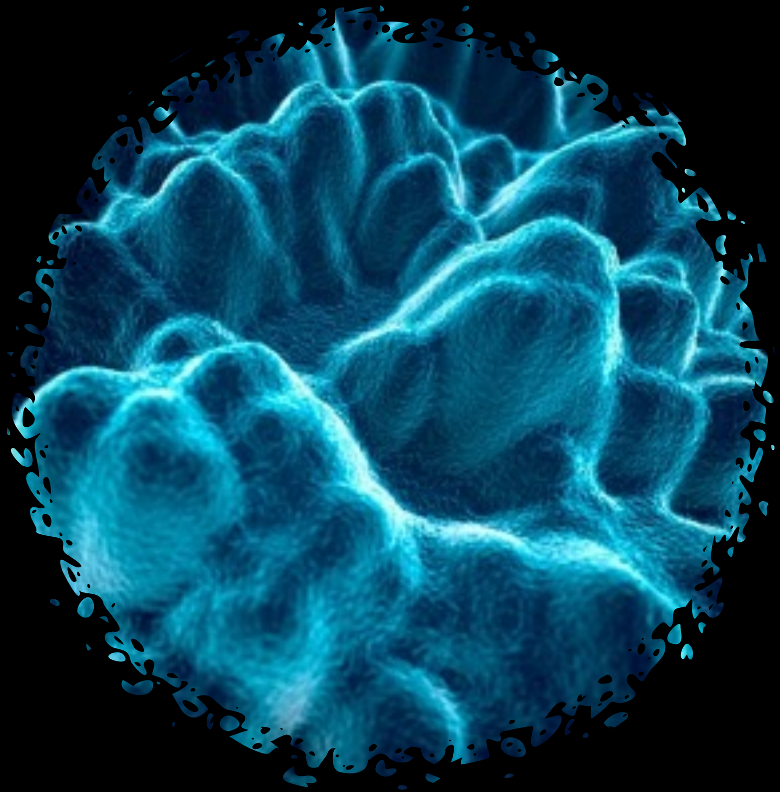
@space_mog

The early Universe



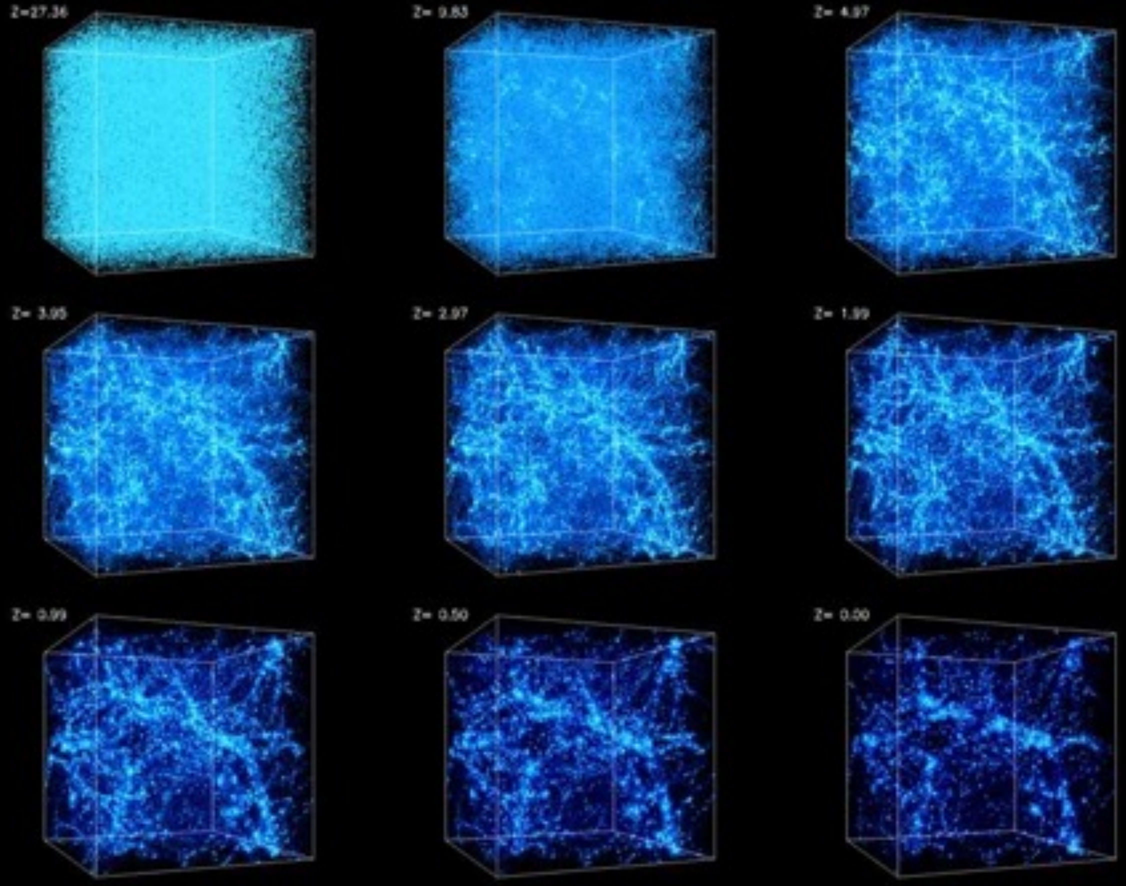
was **hot** and **dense**

a photon-baryon plasma

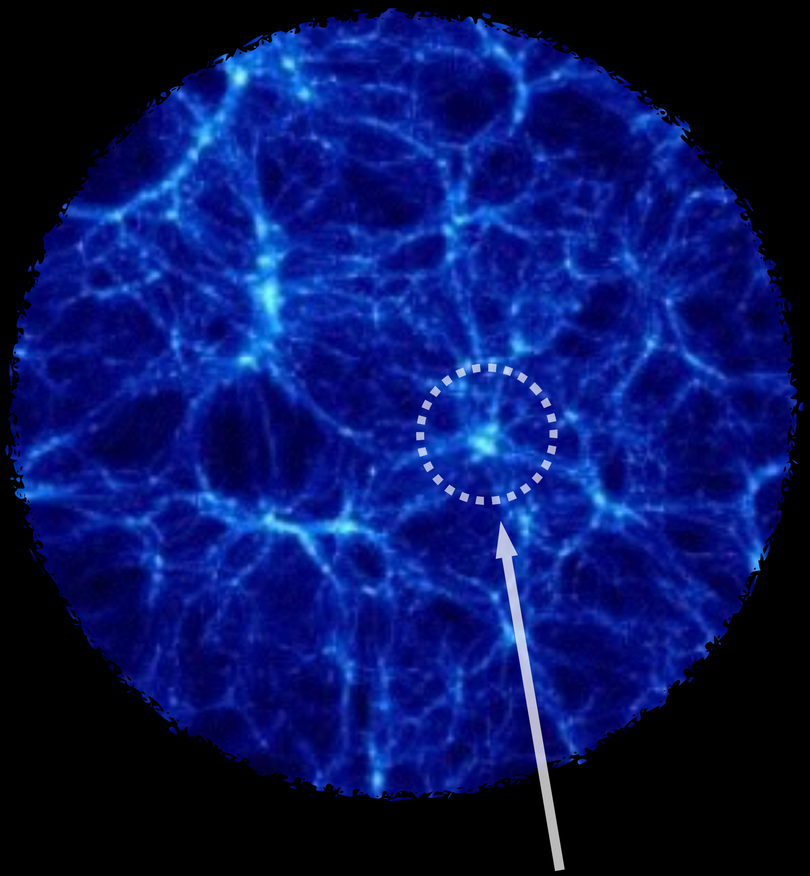


with quantum **density fluctuations**

as the Universe **expanded** with time,



these fluctuations **grew** into structures that we call the large scale structure or **cosmic web**

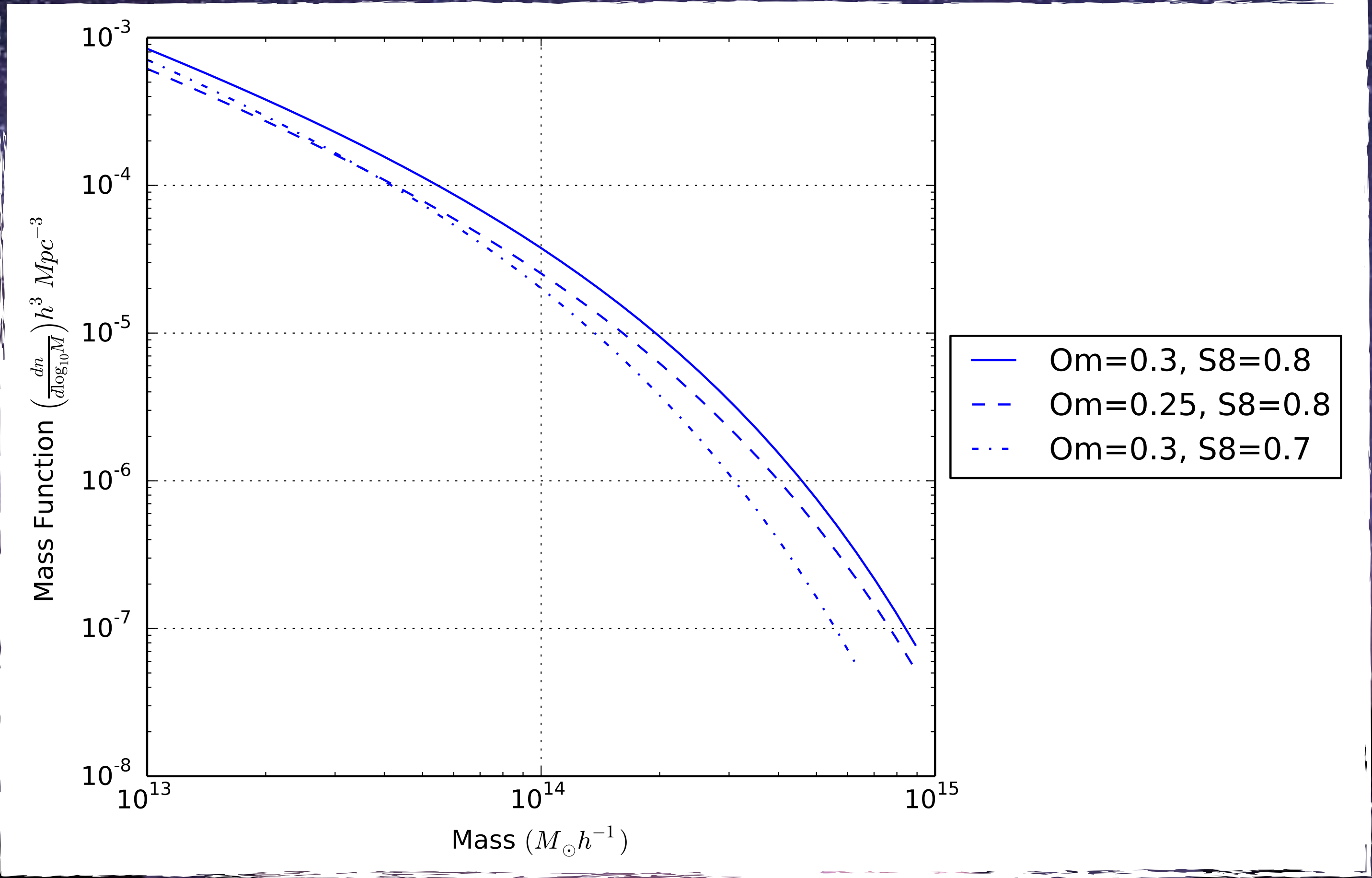


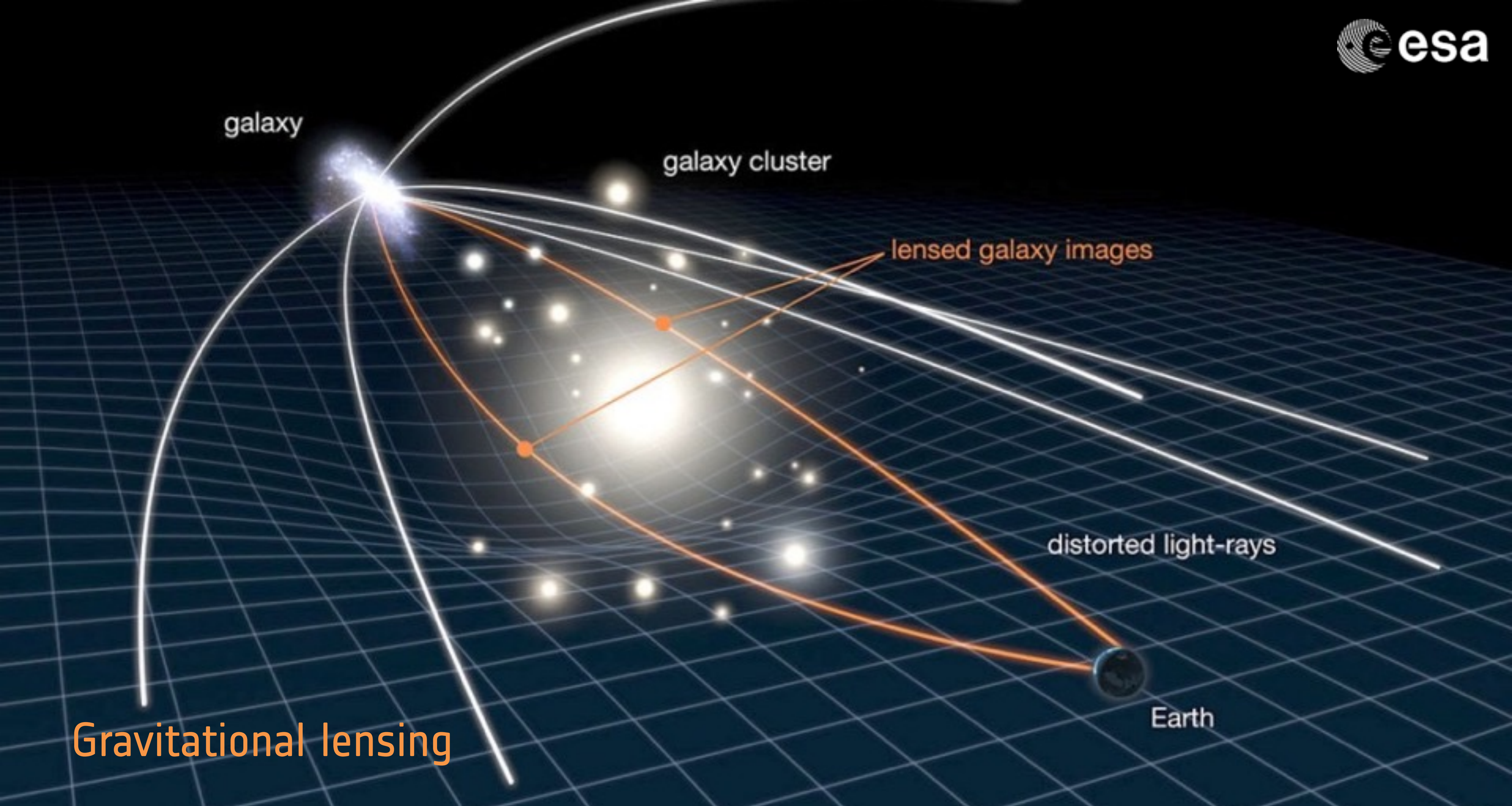
**Galaxy clusters** reside at the **nodes**


Galaxy clusters are the cosmic giants of our universe

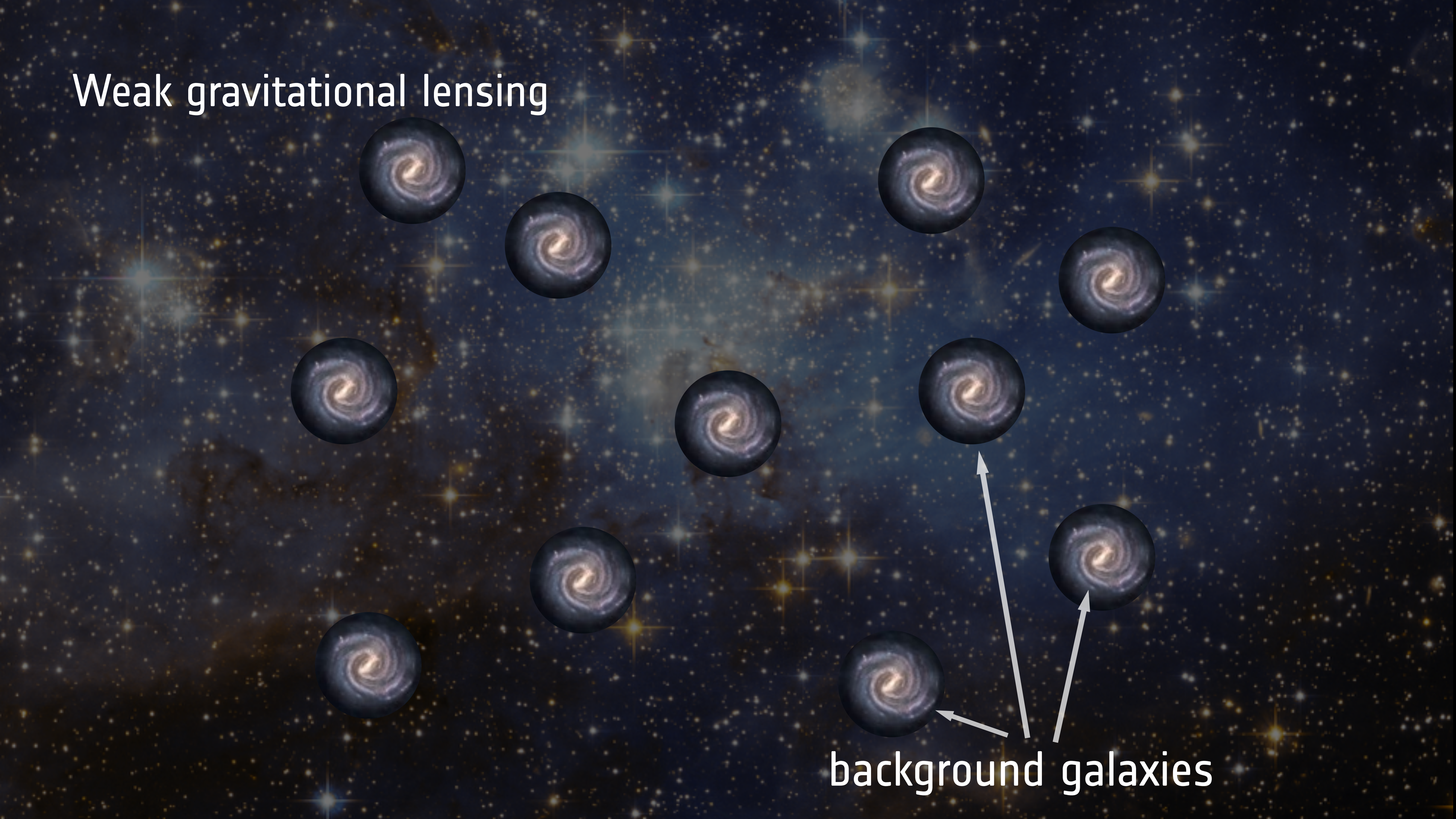
www.spacetelescope.org

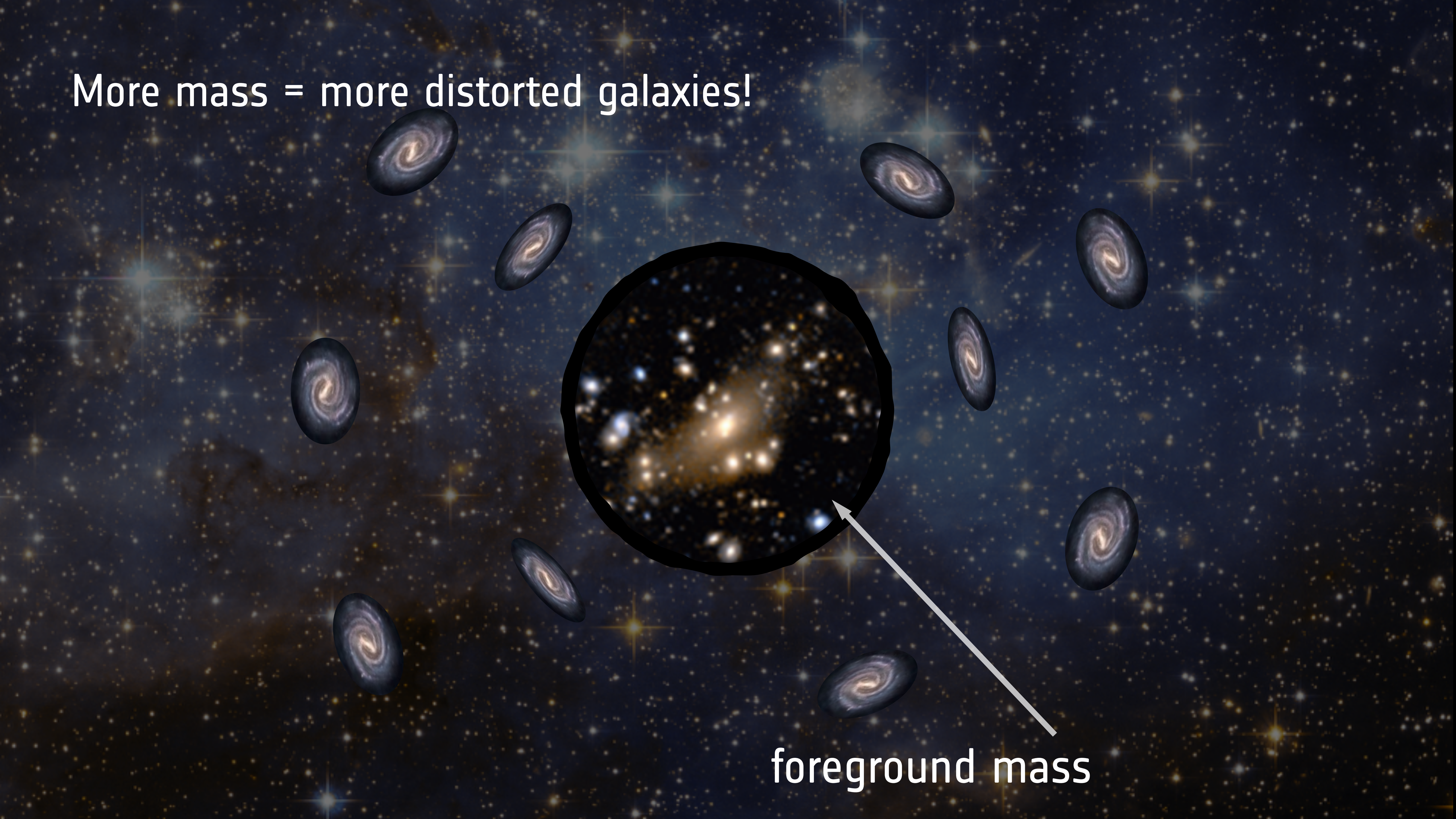
# Galaxy clusters are **powerful cosmological probes**






# Gravitational lensing


Weak gravitational lensing
background galaxies


More mass = more distorted galaxies!
foreground mass

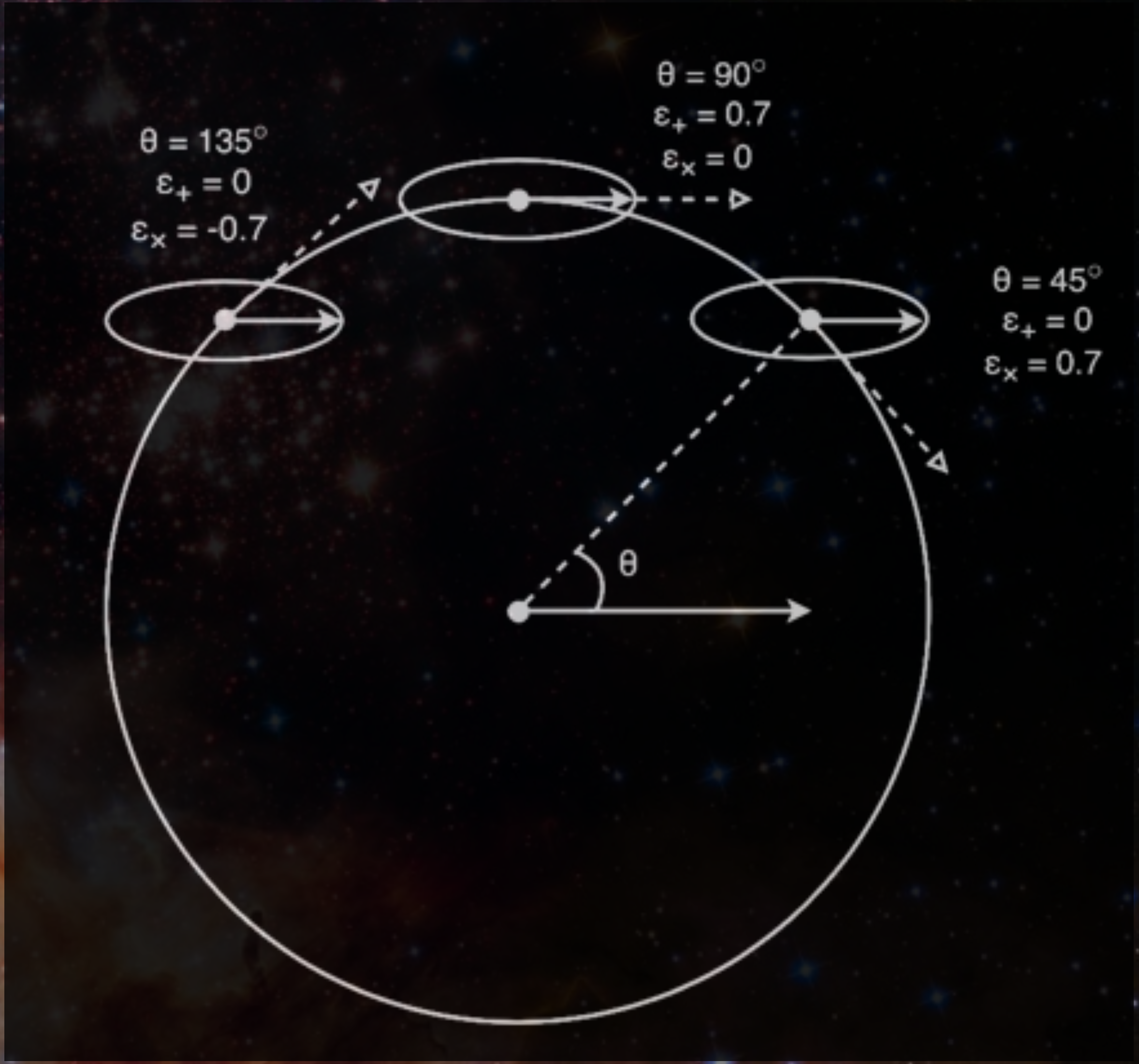
# Weak lensing:

$$\begin{pmatrix} \epsilon_+ \\ \epsilon_\times \end{pmatrix} = \begin{pmatrix} -\Re\left(\epsilon e^{-2i\theta}\right) \\ -\Im\left(\epsilon e^{-2i\theta}\right) \end{pmatrix}$$

$$\langle \gamma_+(M,c) \rangle \approx \langle \epsilon_+ \rangle$$

If no lensing: $\langle \gamma_+ \rangle = 0$

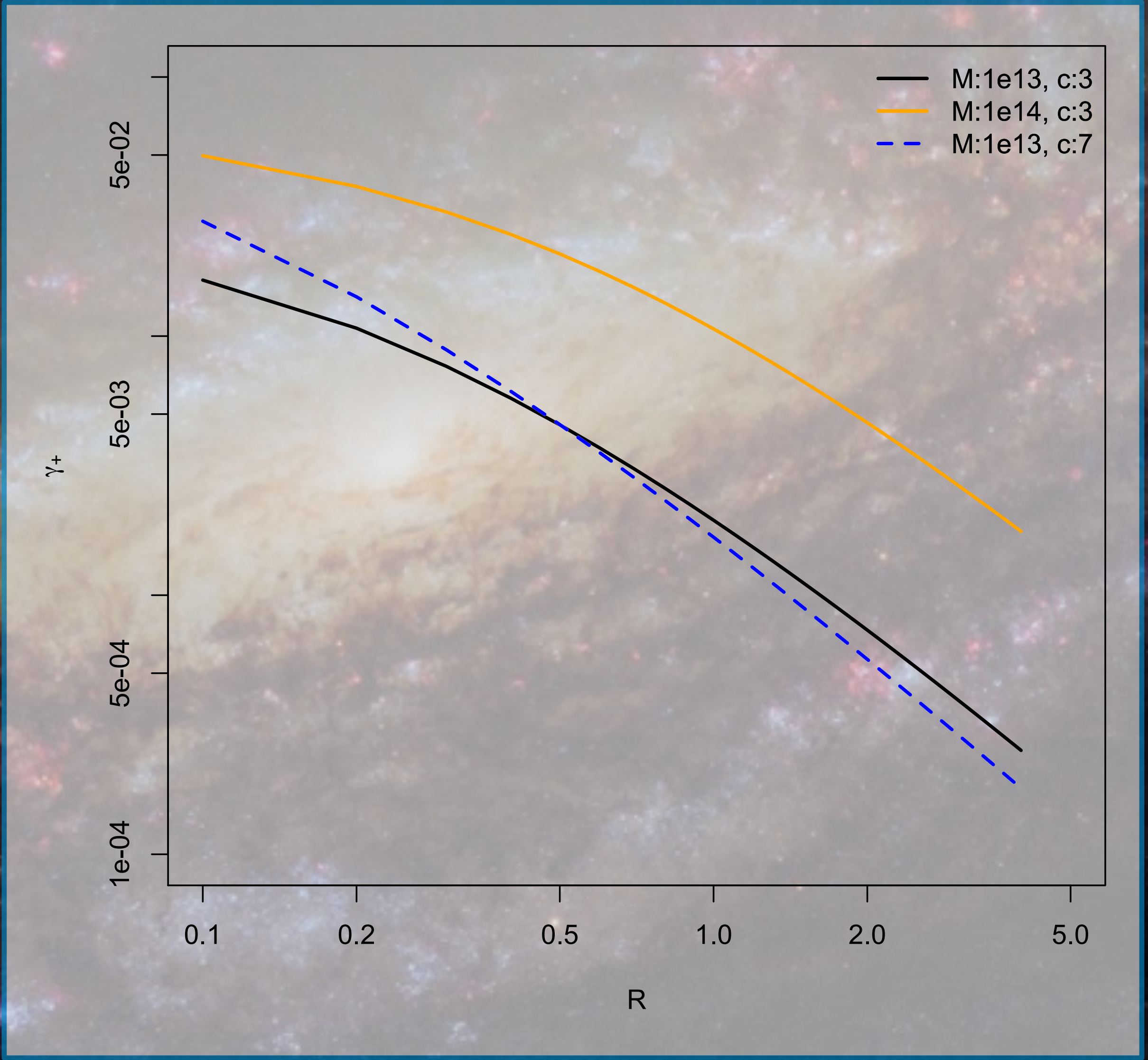
Systematics check: $\langle \gamma_\times \rangle = 0$



# NFW profile (Navarro, Frenk & White 1996)

Universal density profile from numerical simulations:

$$\rho(r) = \frac{\rho_0}{(r/r_s)(1 + r/r_s)^2}$$



# Bayes theorem:

Likelihood

Prior

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Posterior

Marginal likelihood

Typically:

$$P(M, c|\hat{\gamma}(r)) \propto P(\hat{\gamma}(r)|M, c)P(M)P(c)$$

Typically:

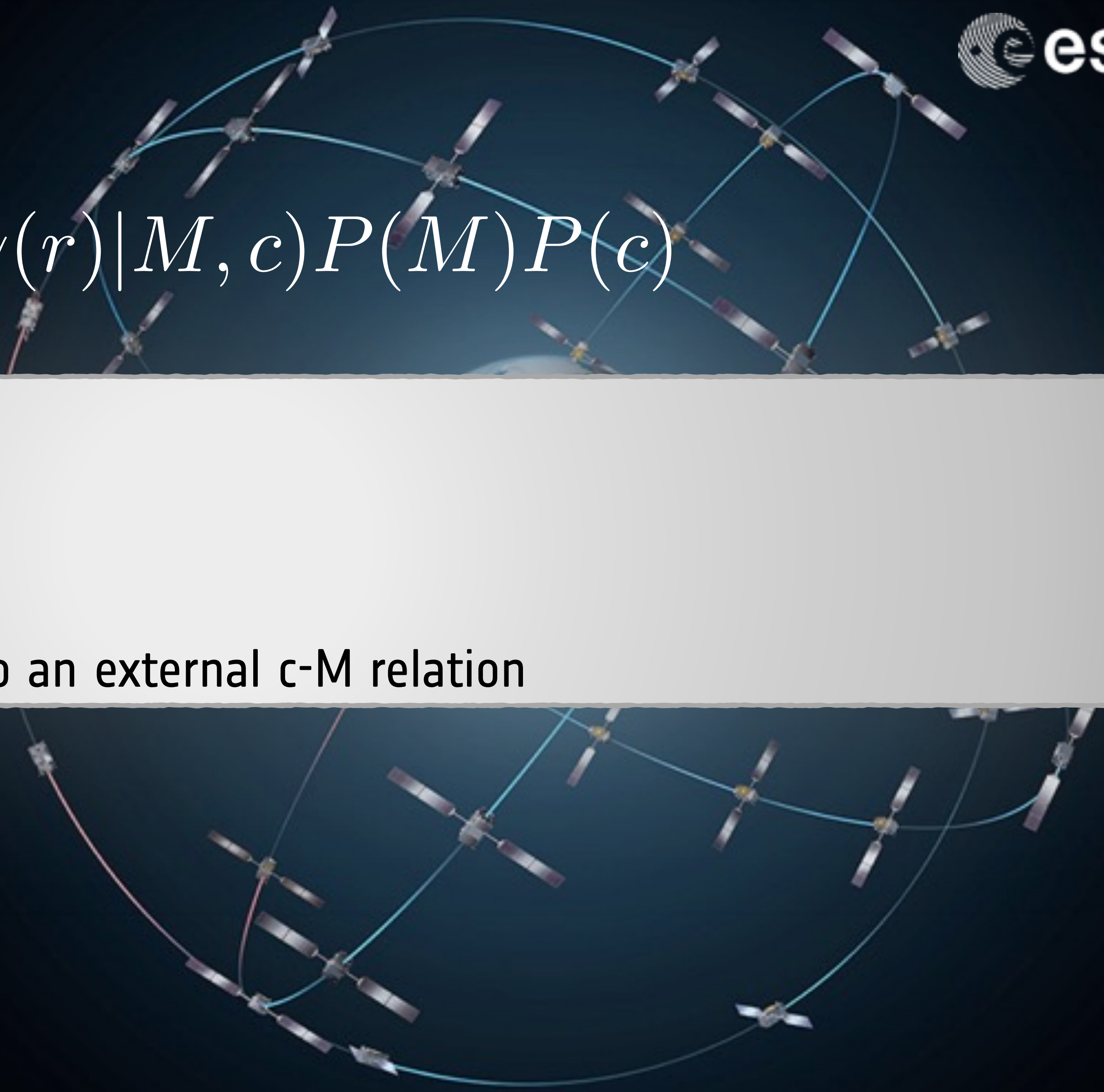
$$P(M, c|\hat{\gamma}(r)) \propto P(\hat{\gamma}(r)|M, c)P(M)P(c)$$

**PROBLEM:**

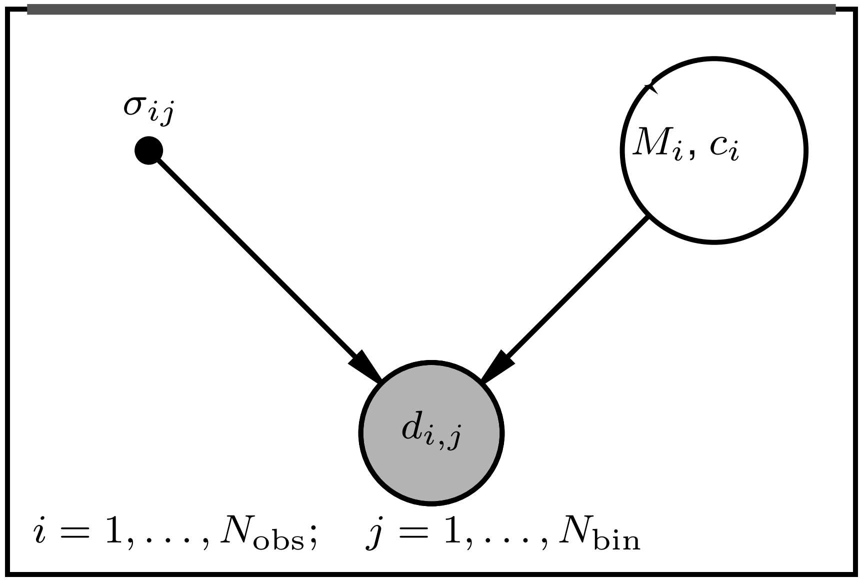
How do we choose P(M) P(c)?

Typical choices are

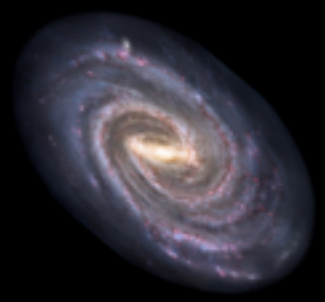
- $\log_{10}(M)$ ~ *uniform*(13,16)
- c ~ *uniform*(1,10) or c **fixed** to an external c-M relation



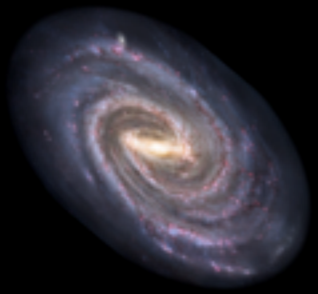
# STAN code:



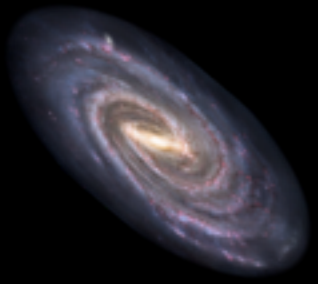

```
functions {
    real[] nfw(real[] r, real m, real c) {
        ...
        return shear;
    }
}

data {
  int<lower=0> Nr;
  real<lower=0> rs[Nr]; // radius in Mpc
  real gamma[Nr];
  real<lower=0> sigma_gamma[Nr];
}

parameters {
  real logM;
  real c;
}

transformed parameters {
    real model_gamma[Nr];
    real M;

    M = exp(logM);
    model_gamma = nfw(rs, M, c);
}

model {
    logM ~ uniform(28,35);
    c ~ uniform(1, 10);
    gamma ~ normal(model_gamma, sigma_gamma);
}
```



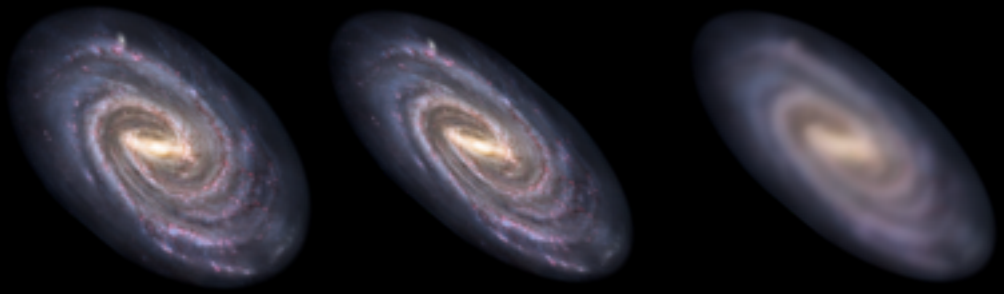
real galaxy



real galaxy



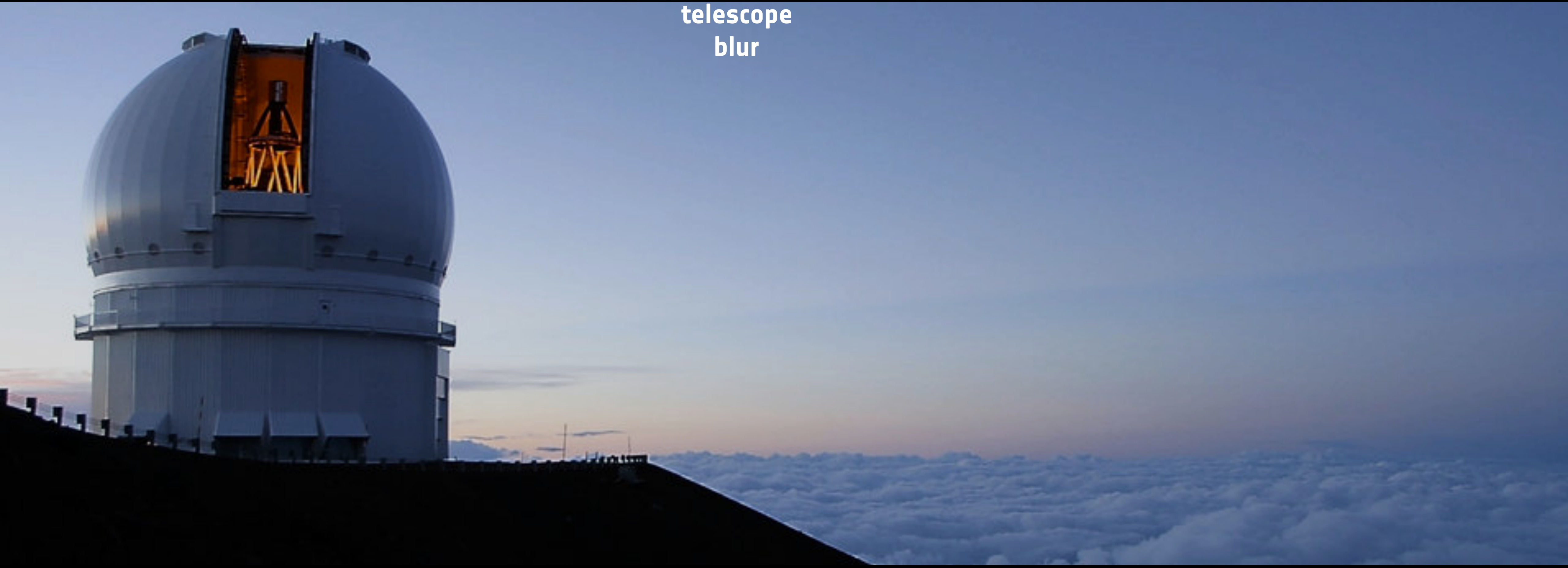
shear

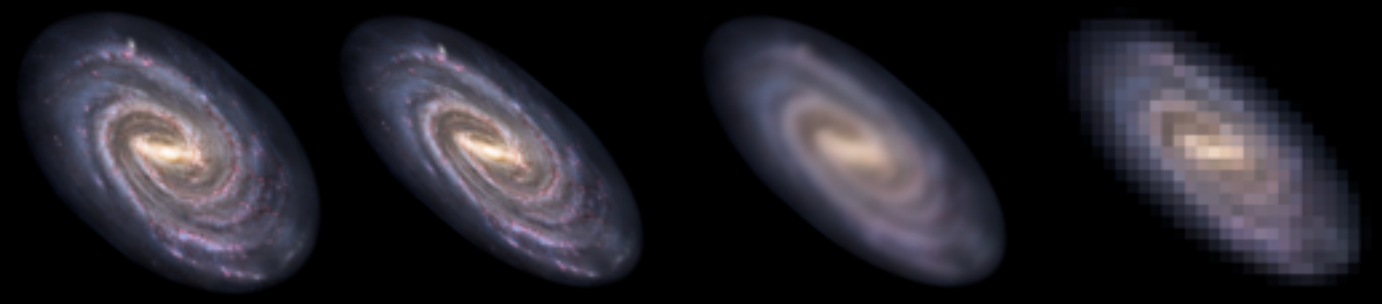


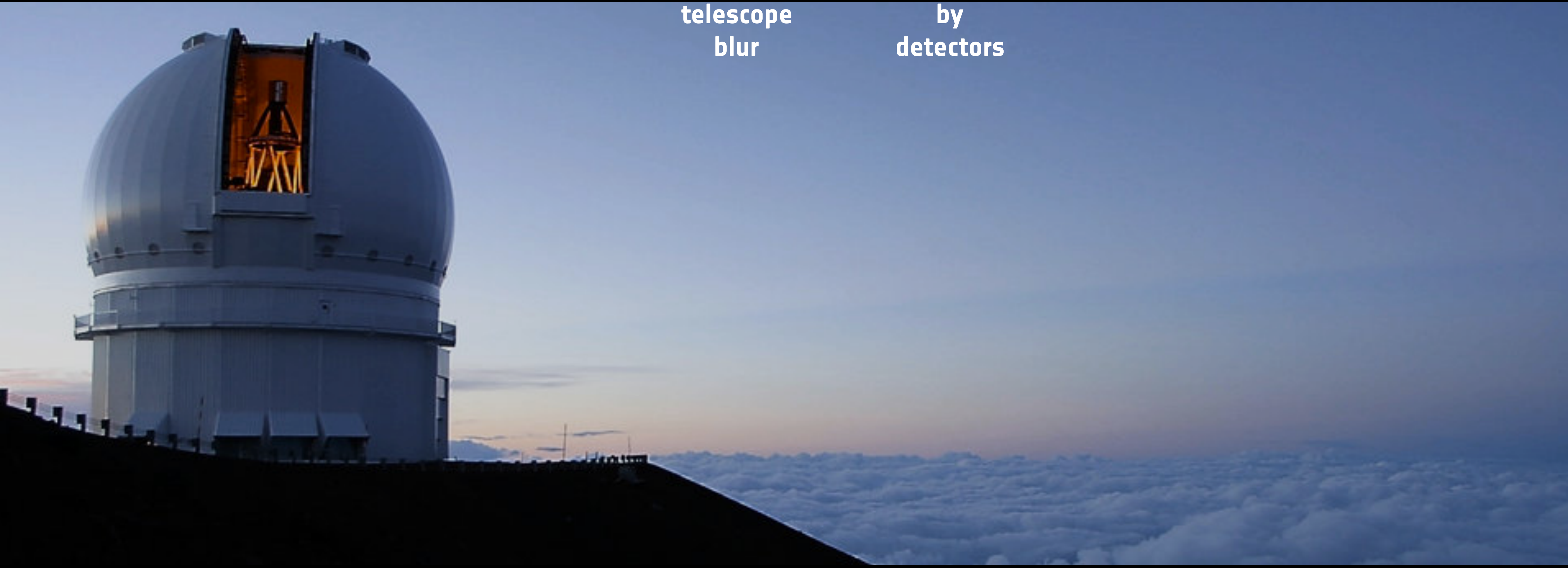
real galaxy

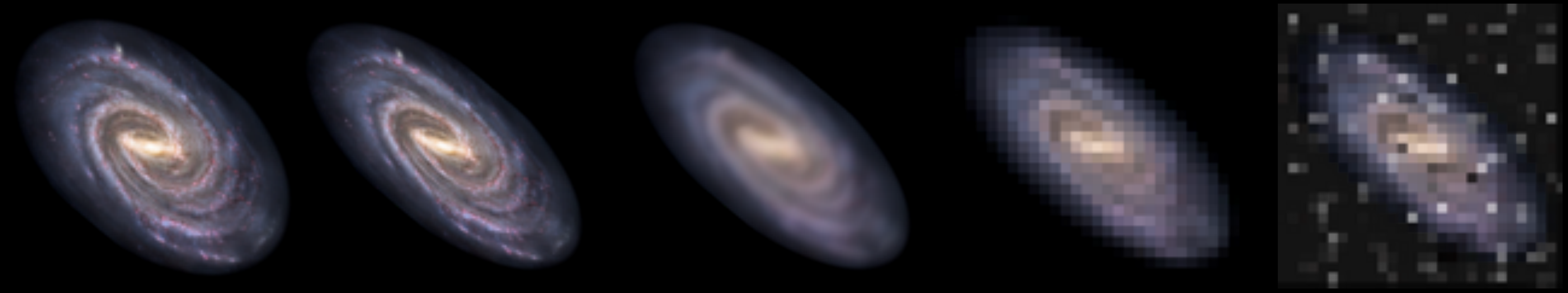
shear

atmosphere & telescope blur

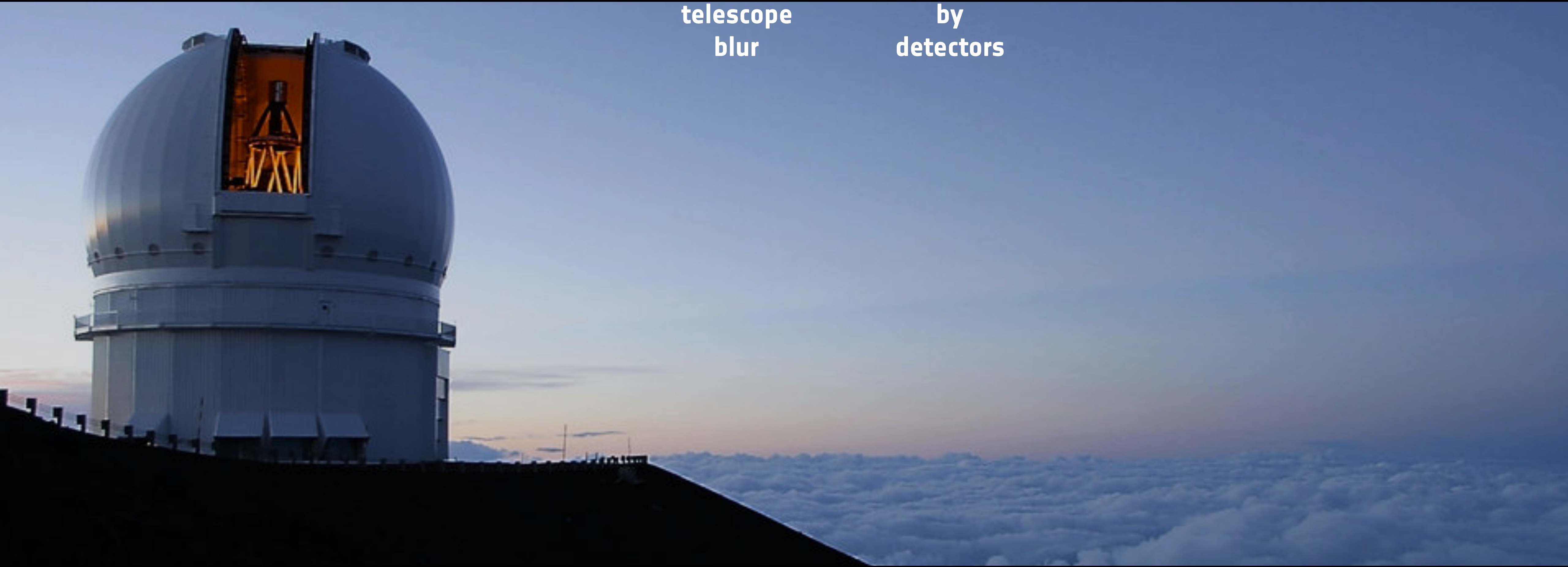



real galaxy
shear
atmosphere & telescope blur
pixelised by detectors

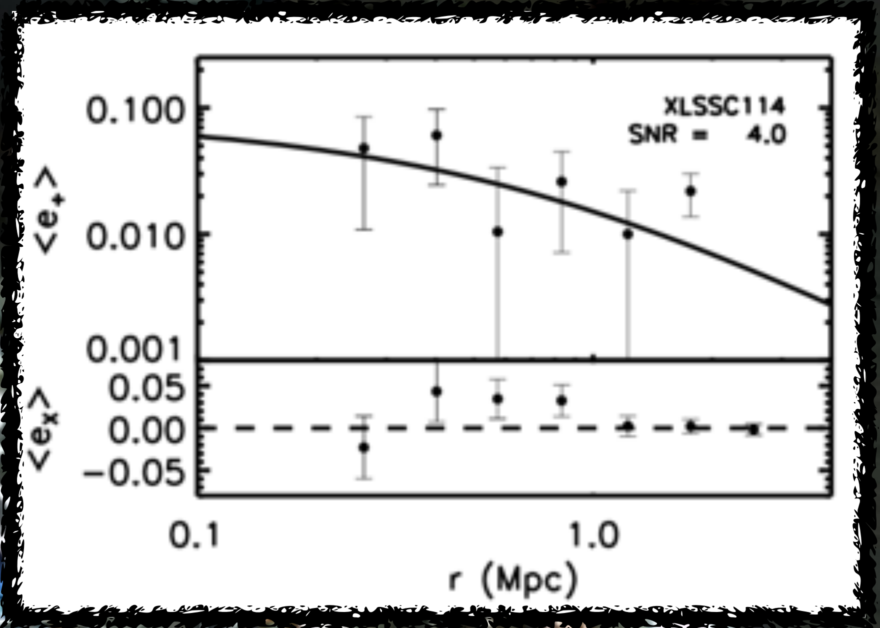




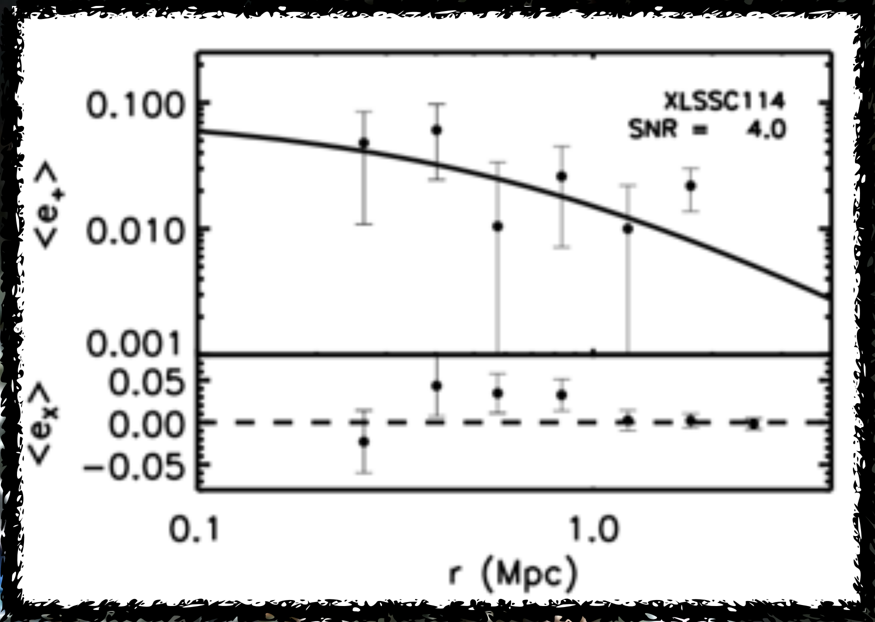
real galaxy | shear | atmosphere & telescope blur | pixelised by detectors | noise



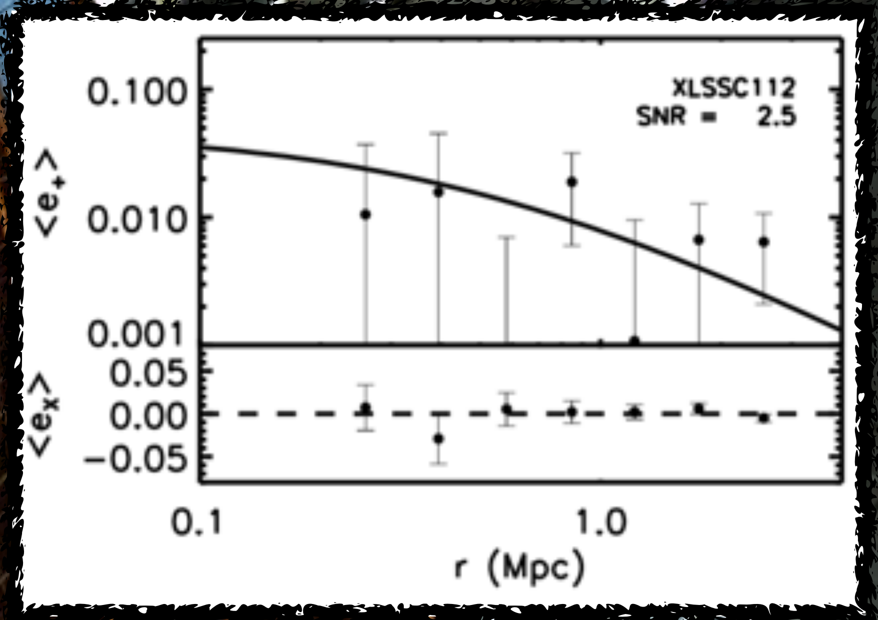
**Fine for those high SNR** clusters where the data alone can constrain parameters



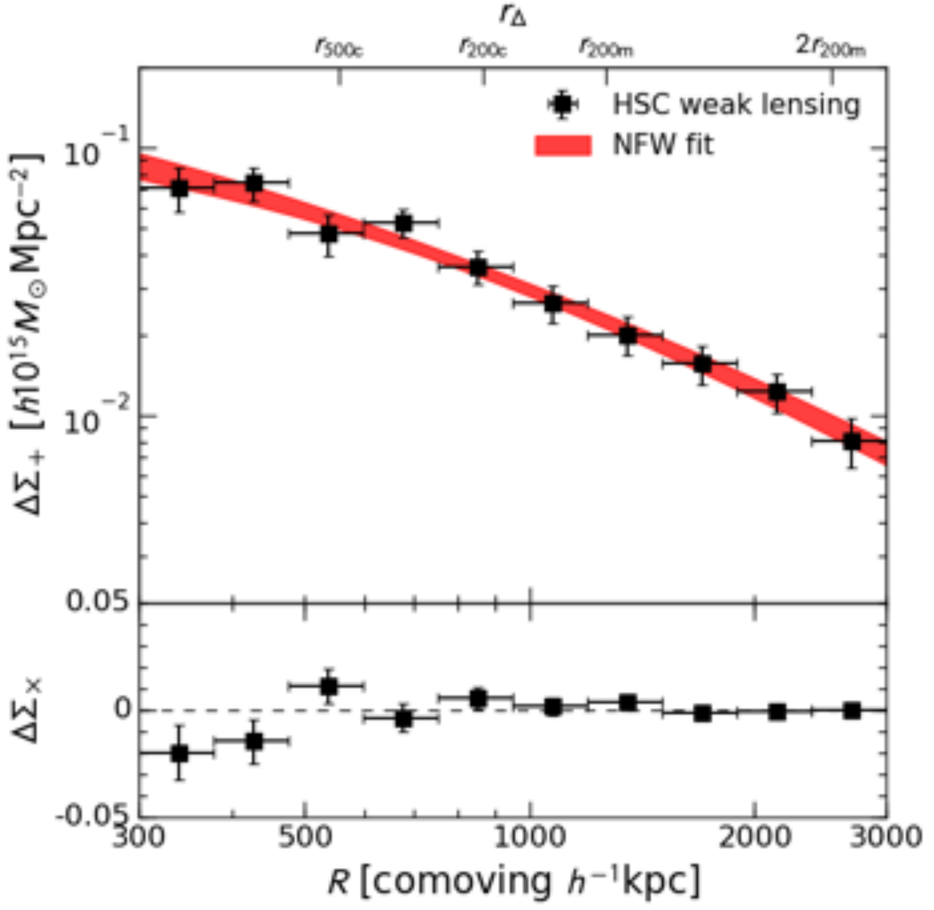
**Fine for those high SNR** clusters where the data alone can constrain parameters




**Bad for low SNR** clusters that become dominated by the prior!



Umetsu et al. (in prep)




## Stacking

- How to stack?
- Assumes all clusters have same mass
- Misrepresentation of outliers

# Typical route:

1

Fit galaxy shapes


caveats:
calibration, errors

# Typical route:

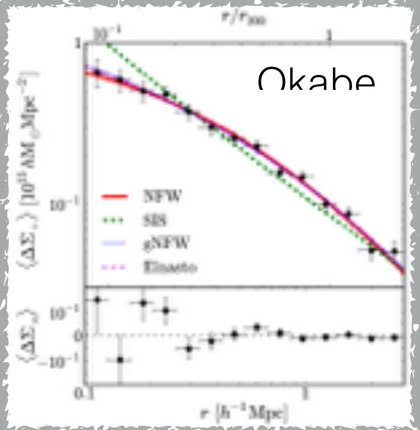
1

Fit galaxy shapes

caveats: calibration, errors

2

Fit shear profiles



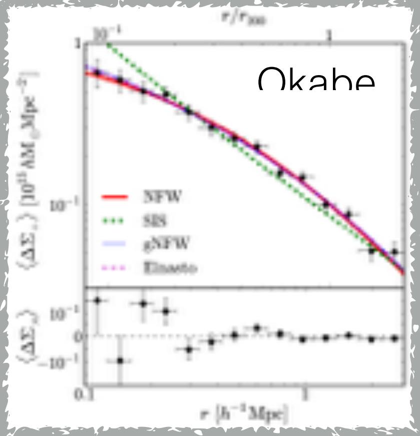

caveats: priors, binning, stacking, centering, errors

# Typical route:

## 1
### Fit galaxy shapes

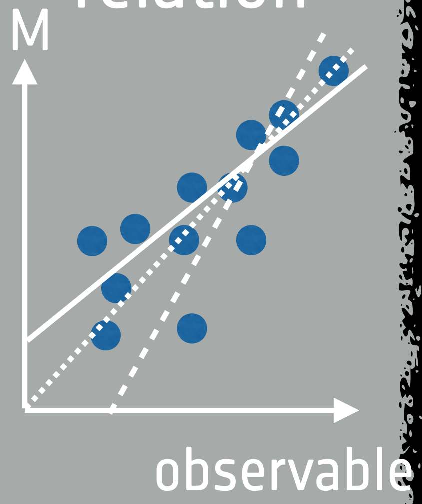
caveats: calibration, errors

## 2
### Fit shear profiles




caveats: priors, binning, stacking, centering, errors

## 3
### Fit scaling relation




caveats: fit type, biases, selection effects, errors

# Typical route:

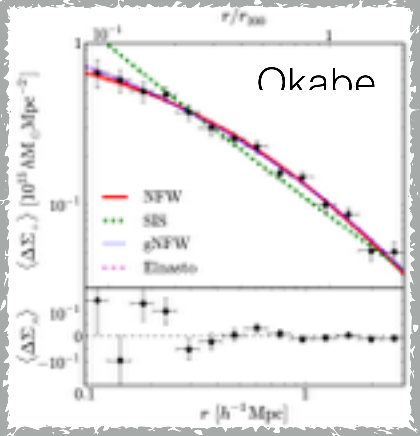
## 1

Fit galaxy shapes

caveats: calibration, errors

## 2

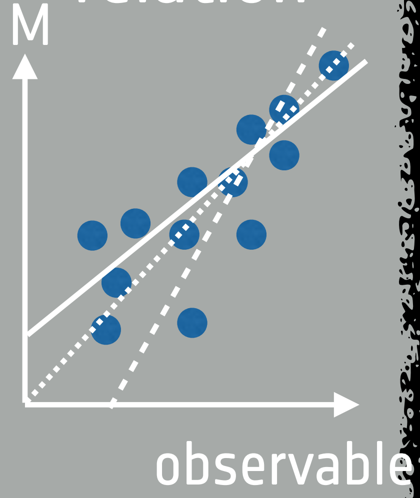
Fit shear profiles




caveats: priors, binning, stacking, centering, errors

## 3

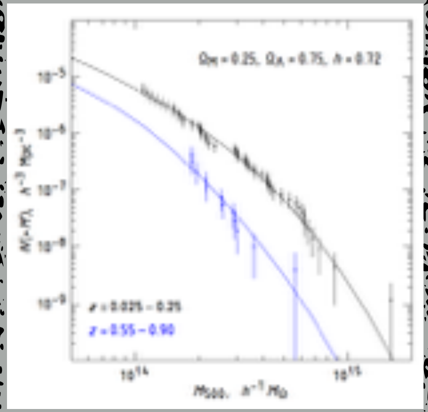
Fit scaling relation




caveats: fit type, biases, selection effects, errors

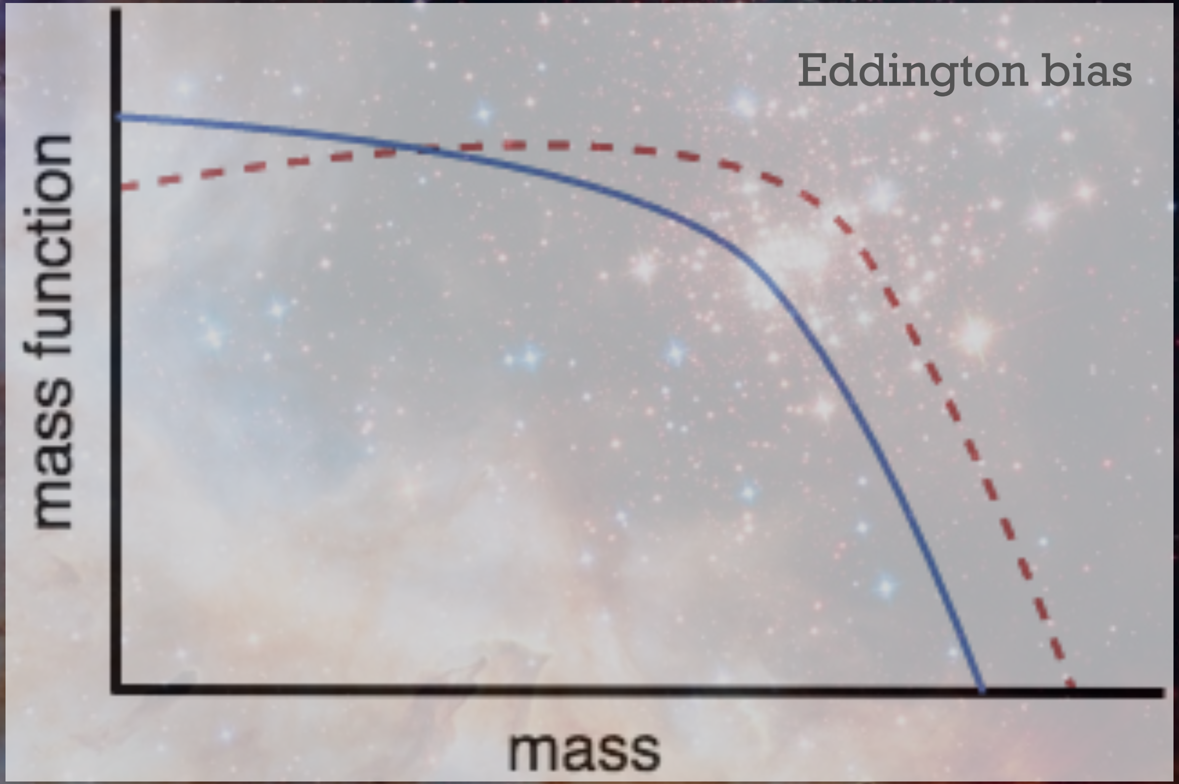
## 4

Fit cluster mass function



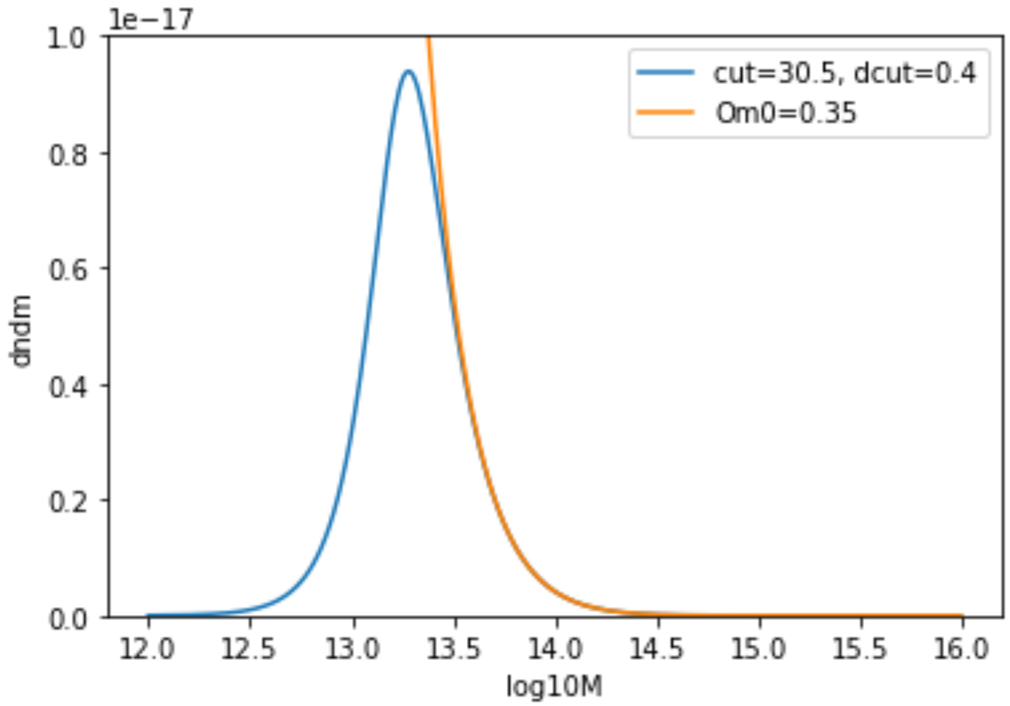
caveats: propagation of errors

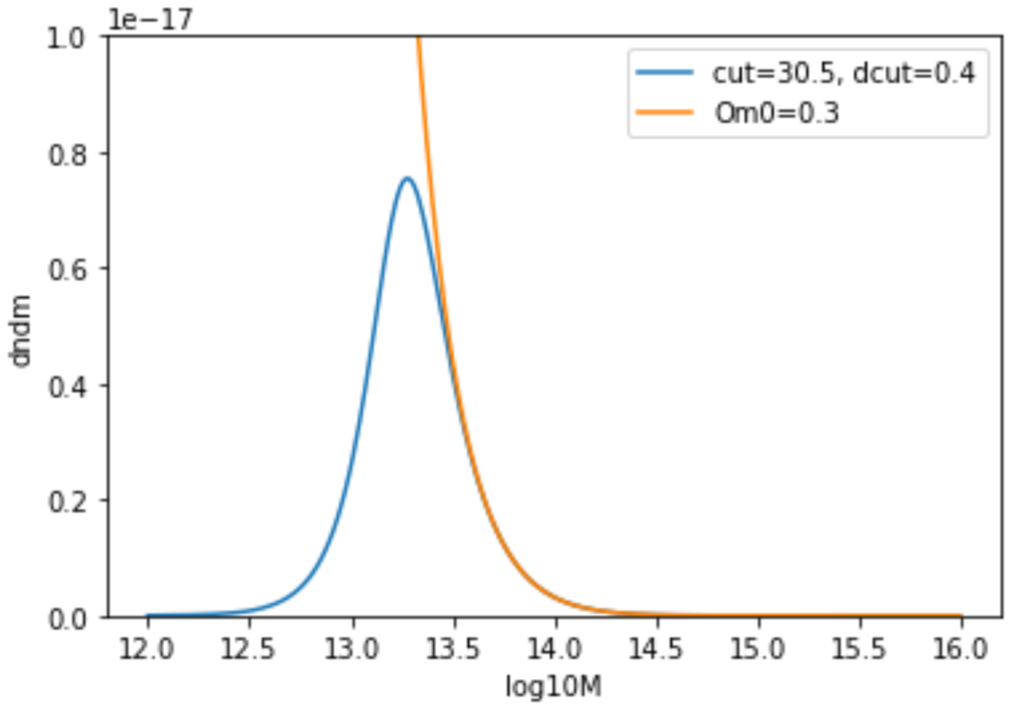
# Difficult to measure masses both **precisely** and **accurately**




Masses are expensive and **scaling relations** derived from different samples suffer **different selection effects and biases**

# Selection function + halo mass function ≅ gaussian




$$P(M_{det}|M) = \frac{1}{2}\left(1 + \tanh\left(\frac{\ln M - \ln M_{cut}}{dcut}\right)\right)$$

# Simple hierarchical approach:

$$P(M), P(c) \quad \rightarrow \quad \ln M, \ln c \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = (\mu_{lnM}, \mu_{lnc})$$

$$\Sigma = \begin{pmatrix} \sigma^2_{\ln M} & \sigma_{\ln M \ln c} \\ \sigma_{\ln M, \ln c} & \sigma^2_{\ln c} \end{pmatrix}$$

Simple hierarchical approach:

$$P(M), P(c) \quad \rightarrow \quad \ln M, \ln c \sim \mathcal{N}(\mu, \Sigma)$$

$$\mu = (\mu_{lnM}, \mu_{lnc})$$

$$\Sigma = \begin{pmatrix} \sigma^2_{\ln M} & \sigma_{\ln M \ln c} \\ \sigma_{\ln M, \ln c} & \sigma^2_{\ln c} \end{pmatrix}$$

Joint probability distribution of all clusters:

$$P(\mu, \Sigma, \mathbf{M}, \mathbf{c} | \hat{\gamma}, \boldsymbol{\sigma}_\gamma) =$$
$$P(\hat{\gamma} | \gamma(\mathbf{M}, \mathbf{c}), \boldsymbol{\sigma}_\gamma) P(\mathbf{M}, \mathbf{c} | \mu, \Sigma) P(\mu) P(\Sigma)$$

# Hyperpriors

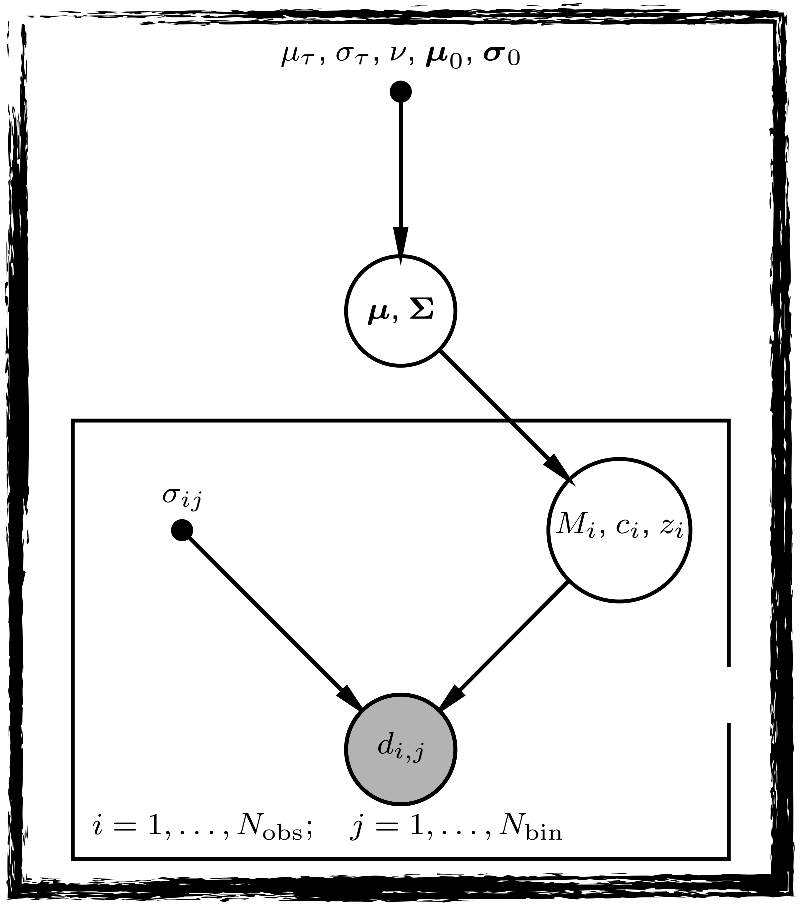
$$\mu \sim \mathcal{N}\left(\begin{bmatrix} 32 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}\right)$$

Optimise covariance matrix sampling using Cholesky factorisation:

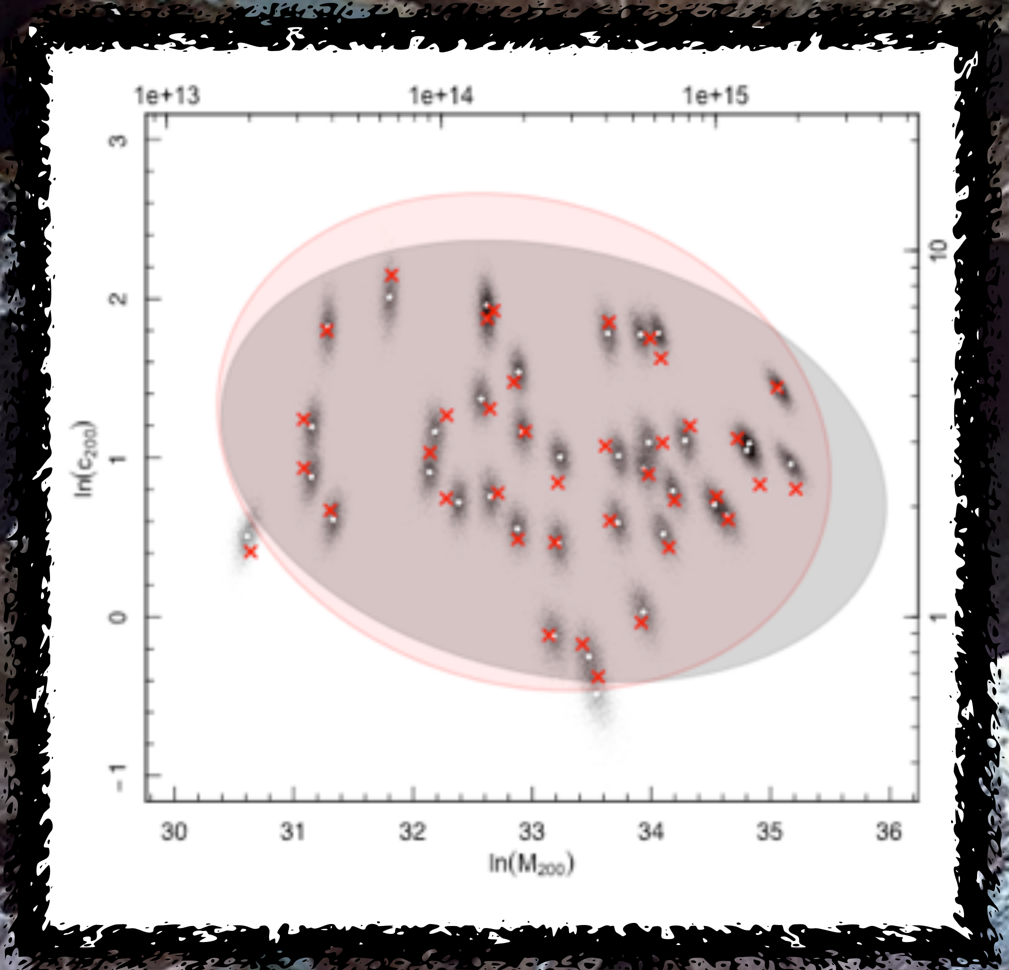
$$\Sigma = \mathrm{diag}(\tau)\Omega\mathrm{diag}(\tau)$$
$$\Sigma = \mathrm{diag}(\tau)L_\Omega L_\Omega^\intercal\mathrm{diag}(\tau)$$
$$\tau \sim \mathcal{N}(0, 2)$$
$$L_\Omega \sim LKJ(10)$$

# STAN code:

```
data {
  int<lower=0> Nc; //number of clusters
  int<lower=0> Nr; //number of radial bins
  vector<lower=0>[Nr] rs[Nc]; // radius in Mpc
  vector[Nr] gammas[Nc];
  vector<lower=0>[Nr] sigma_gammas[Nc];
  vector[2] mu0; // hyperpriors [lnM, lnc]
  vector[2] sigma0;
}

parameters {
    vector[2] mu;
    cholesky_factor_corr[2] L_Omega;
    vector<lower=0>[2] tau; //prior scale
    vector[2] log_params[Nc];
}

transformed parameters {
    vector<lower=0>[2] cl_params[Nc];
    vector[Nr] model_gammas[Nc];
    cholesky_factor_cov[2] L; //cholesky factor of covariance matrix

    L = diag_pre_multiply(tau, L_Omega);
    for(i in 1:Nc){
        cl_params[i] = exp(log_params[i]);
        model_gammas[i] = nfw(rs[i], cl_params[i][1], cl_params[i][2]);
    }
}

model {
    mu ~ normal(mu0, sigma0);
    tau ~ normal(0,2);
    L_Omega ~ lkj_corr_cholesky(10);
    log_params ~ multi_normal_cholesky(mu, L);
    for (i in 1:Nc){
        gammas[i] ~ normal(model_gammas[i], sigma_gammas[i]);
    }
}
```

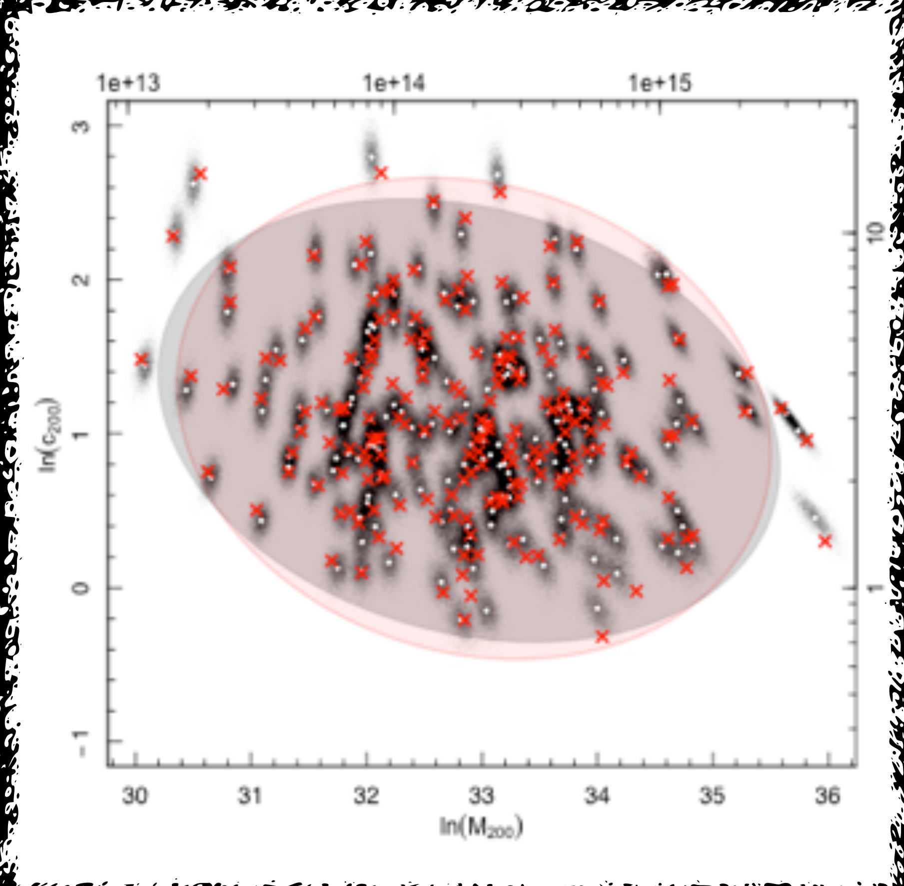


# Toy simulations

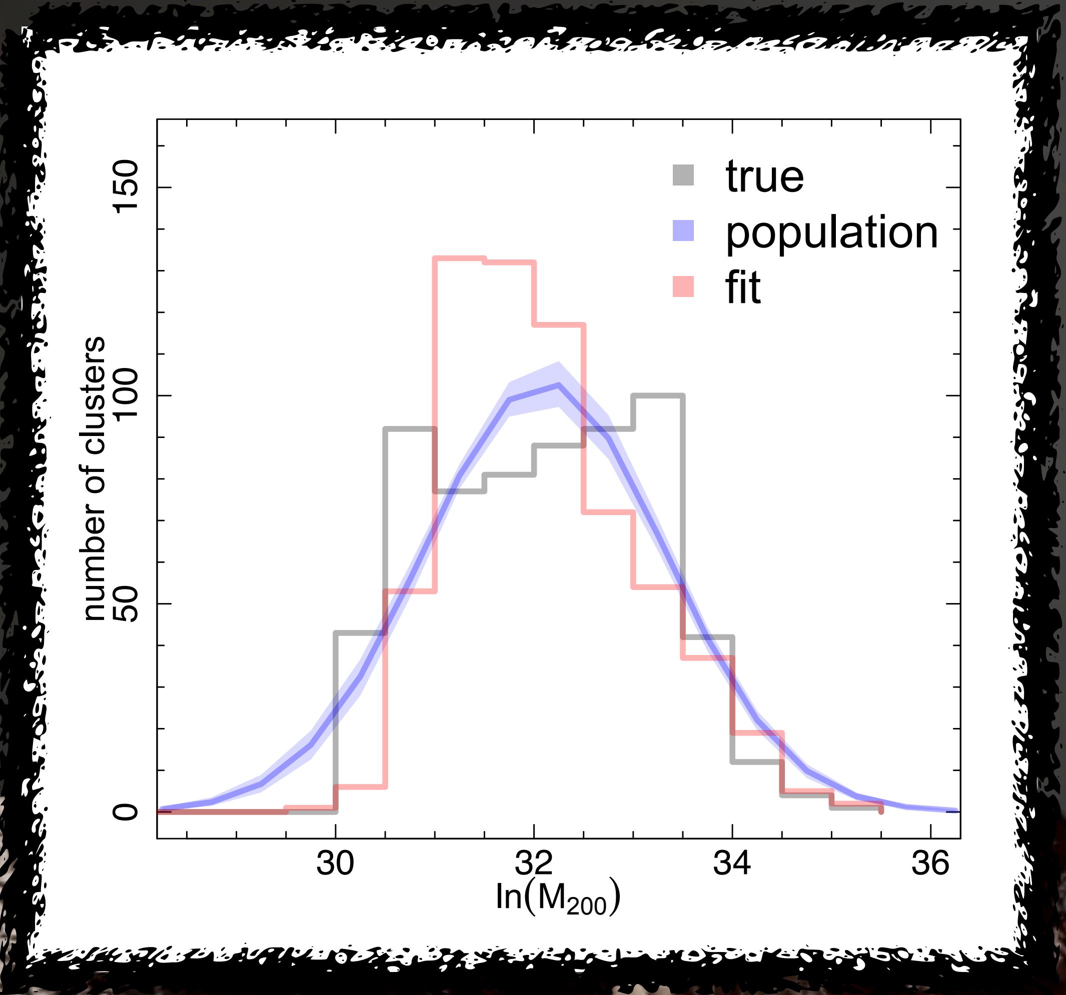



Test on **38** simulated clusters
- **10%** shear uncertainty
- average parameters recovered to **2%** w/ exception of $\mu_c$

# Toy simulations



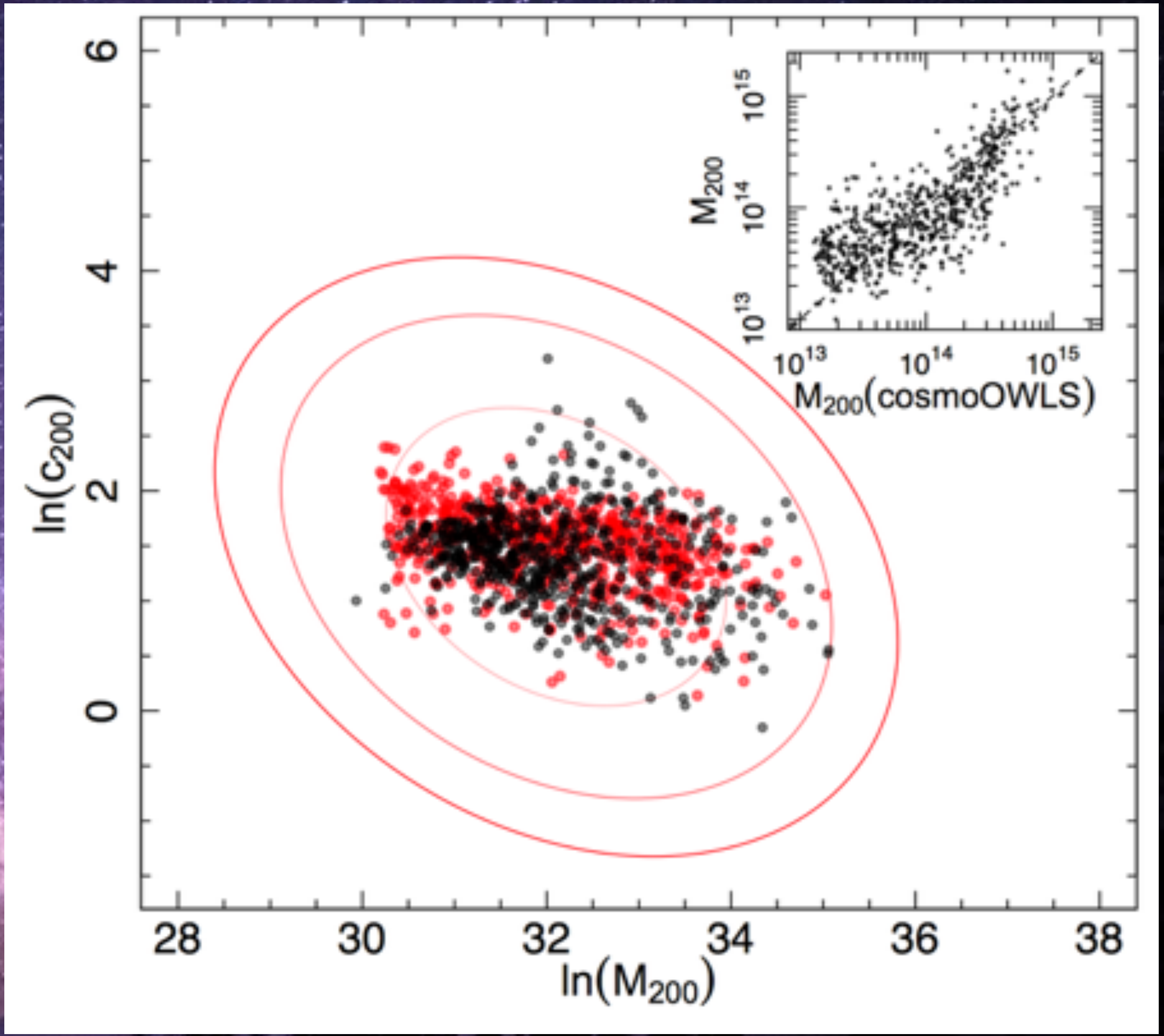

Test on **200** simulated clusters
- **10%** shear uncertainty
- All parameters recovered to **<2%** on average




- Cluster counts of **individual** masses **underestimates at low M** and **over estimates at mean**
- **Population** estimation gives **good fit** to truth **at all M scales**

# Tests on hydrodynamical simulations

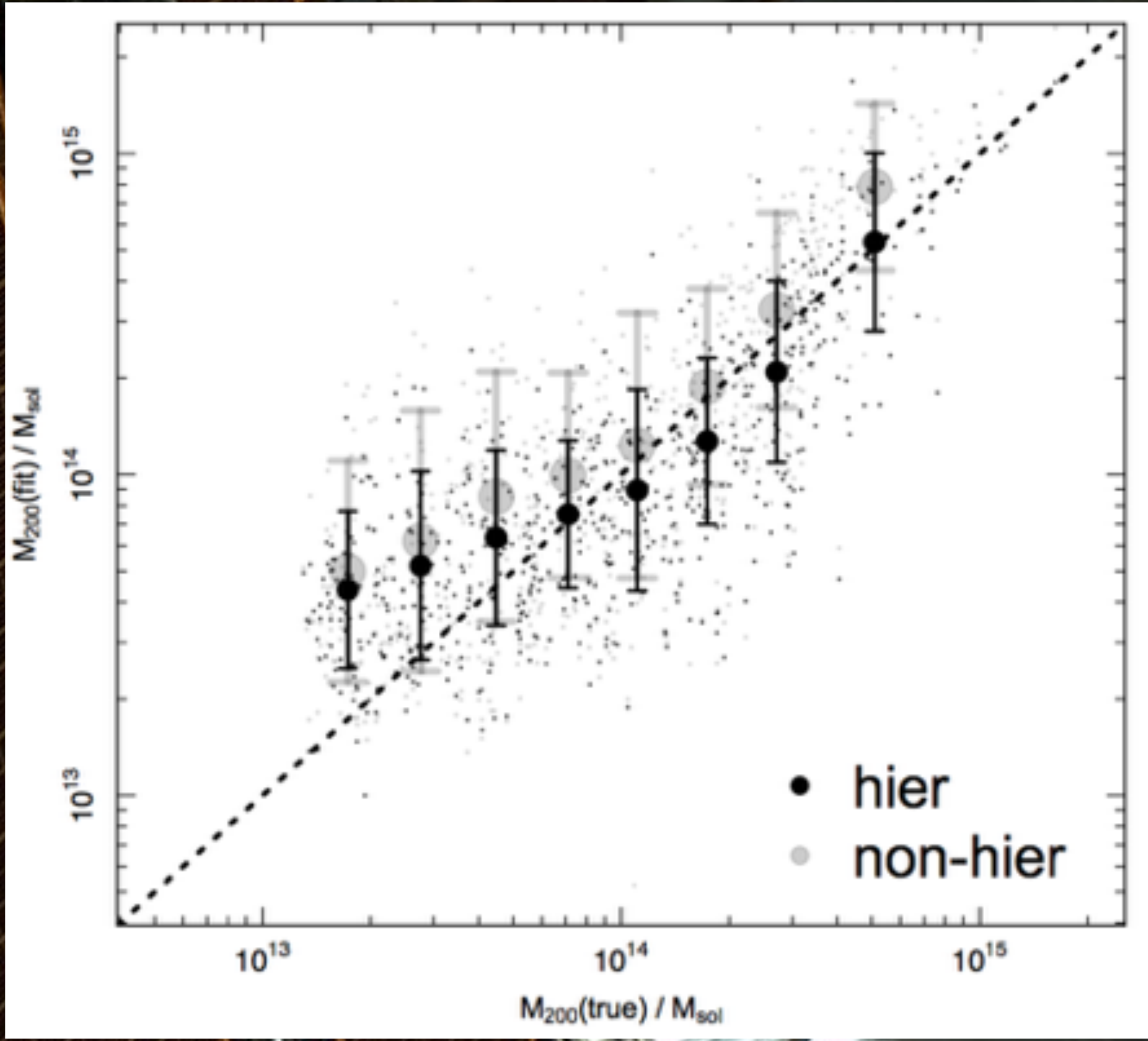
- **COSMO-OWLS** simulations (*Le Brun +2015*)
- DM only run
- WL shear from ray-tracing of source
- **632 clusters** selected as 100 systems in 8 log-spaced mass bins **13 < $\log_{10}(M_{500})$ <15** where possible
- shear profiles constructed from source density **5 galaxies arcmin$^{-2}$, with shape noise**



# How much are we influenced by the high S/N clusters?

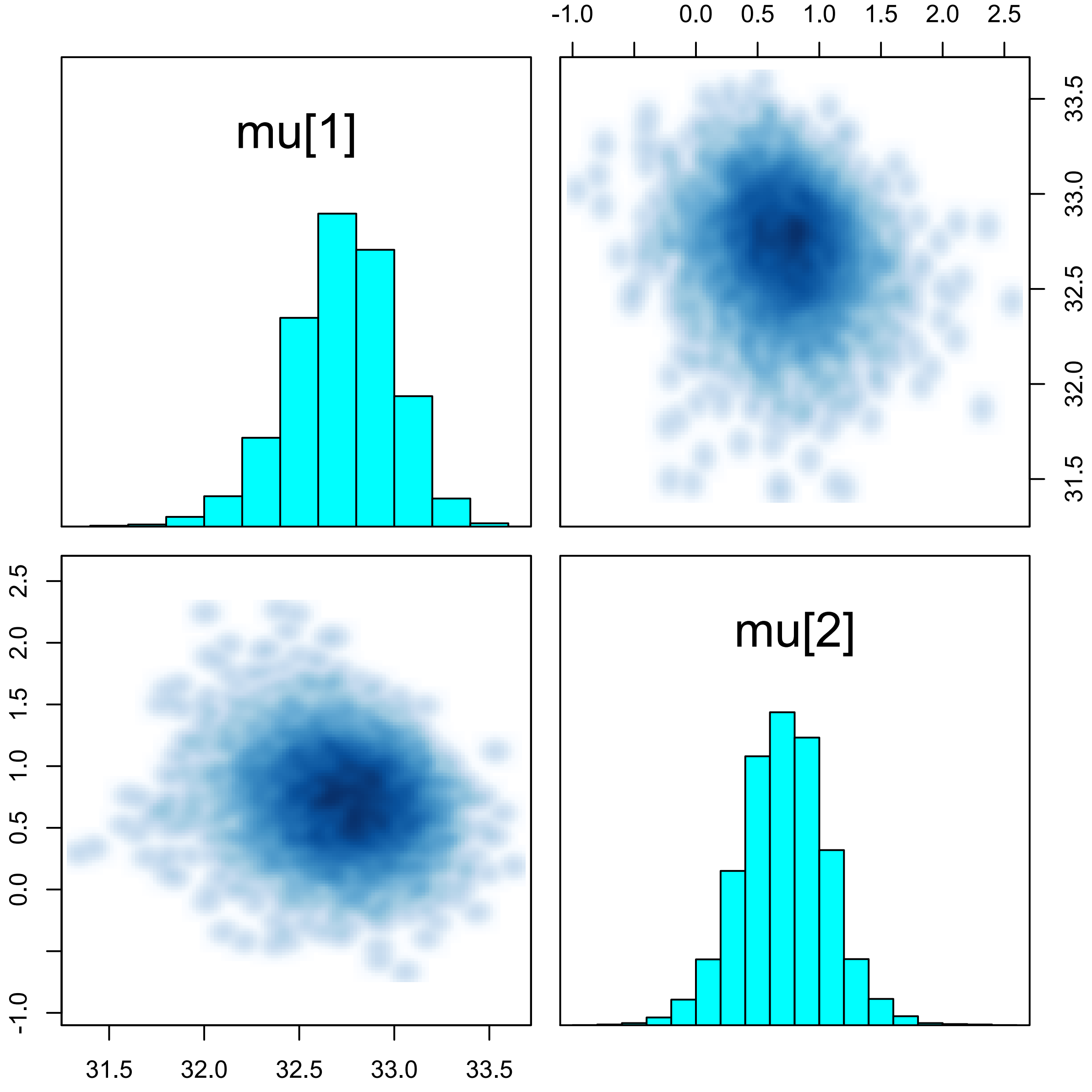
High M clusters do **NOT** dominate the fit!

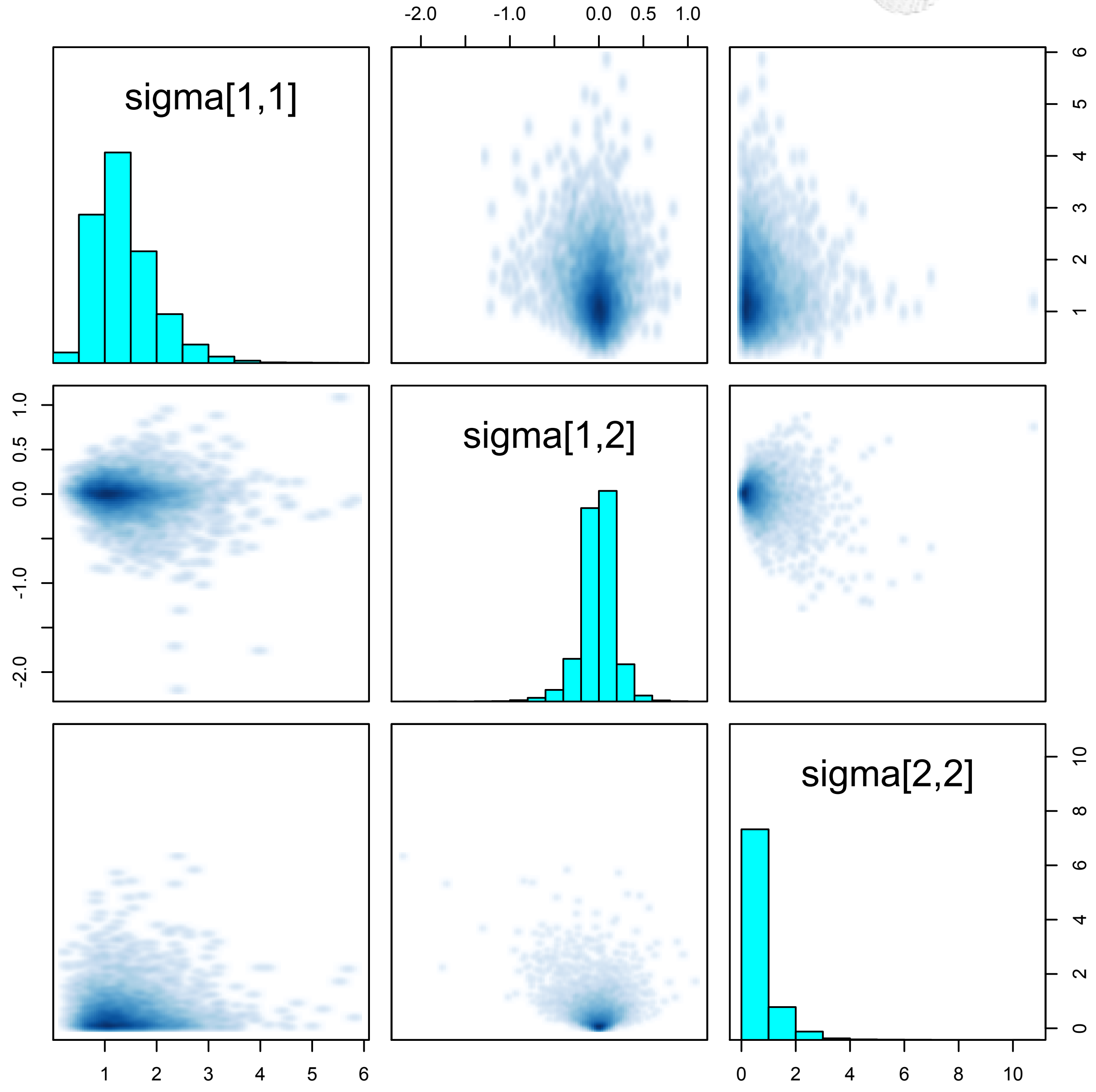
- **More low M** groups
- **High M bin is moved down more** than low M bin moved up



# Application to real data

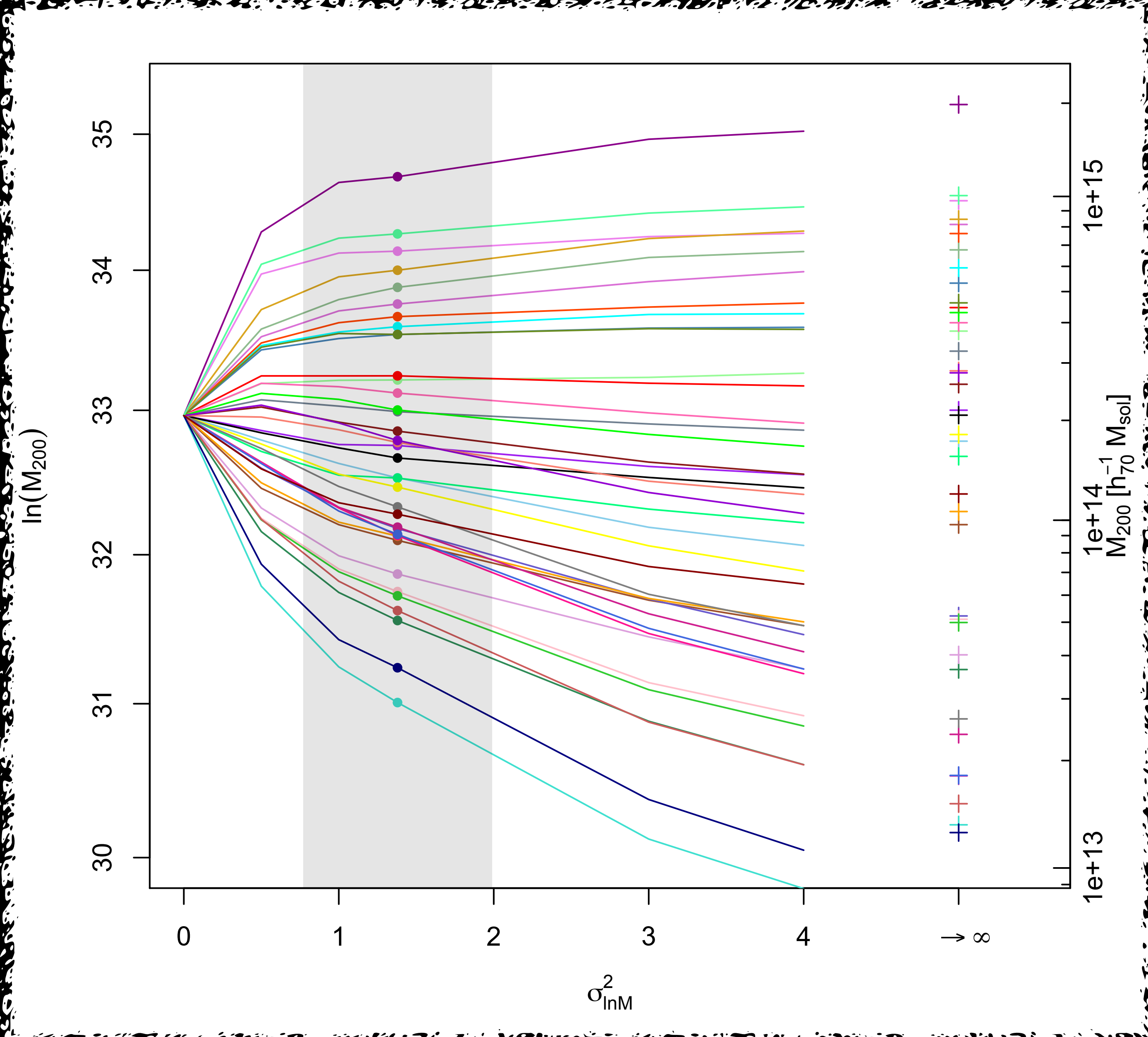
- XXL survey 2 x 25 $deg^2$
- Brightest 100 clusters (Pacaud+2016)
- Overlap with weak lensing survey CFHTLenS
- nearby clusters $z<0.6$ & $T_X$ (Giles+2016)
- 38 galaxy clusters


mu[1]
mu[2]
-1.0 0.0 0.5 1.0 1.5 2.0 2.5
33.5 33.0 32.5 32.0 31.5
31.5 32.0 32.5 33.0 33.5


sigma[1,1]
sigma[1,2]
sigma[2,2]
-2.0 -1.0 0.0 0.5 1.0
1 2 3 4 5 6
0 2 4 6 8 10

# Shrinkage

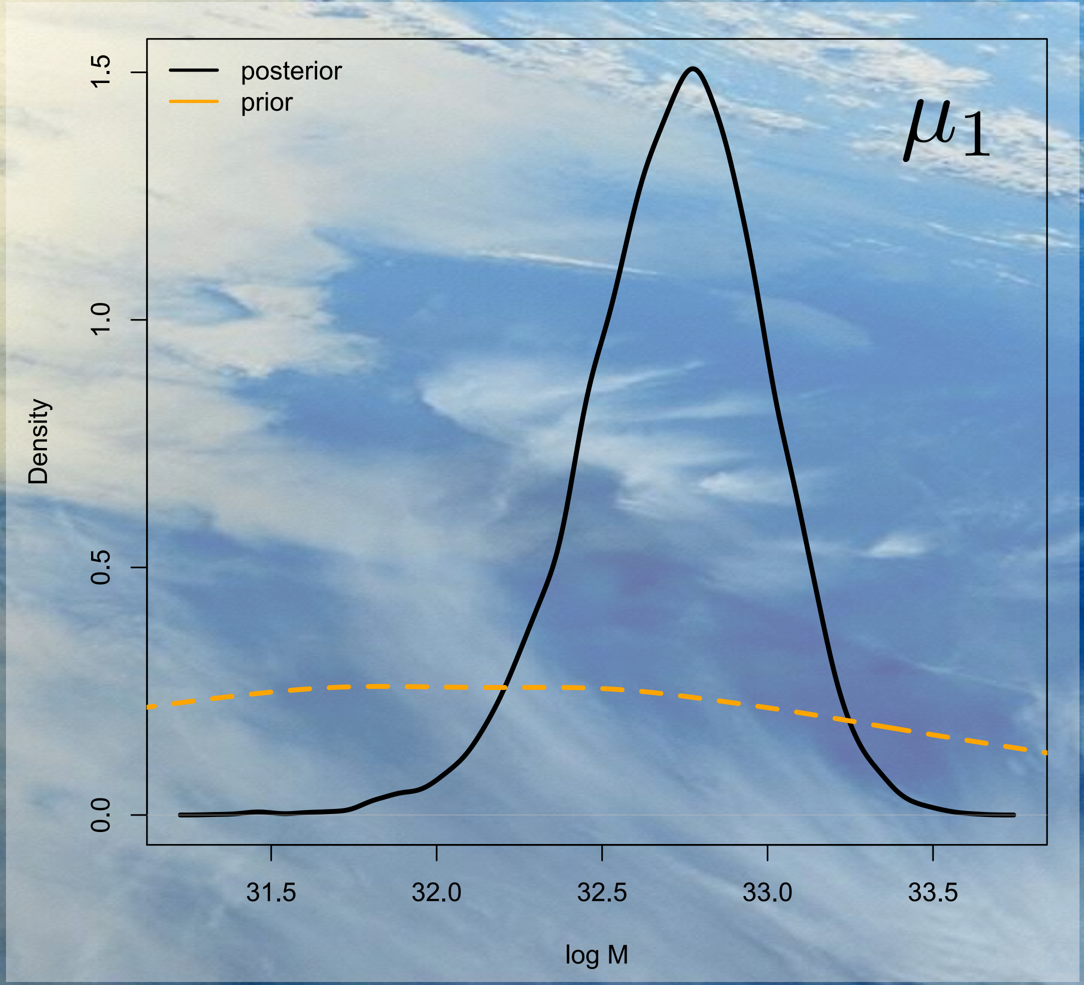
**Hierarchical approach is quasi-stacking!** - clusters shrink towards mean but strength of <u>shrinkage is determined completely by the data</u>!



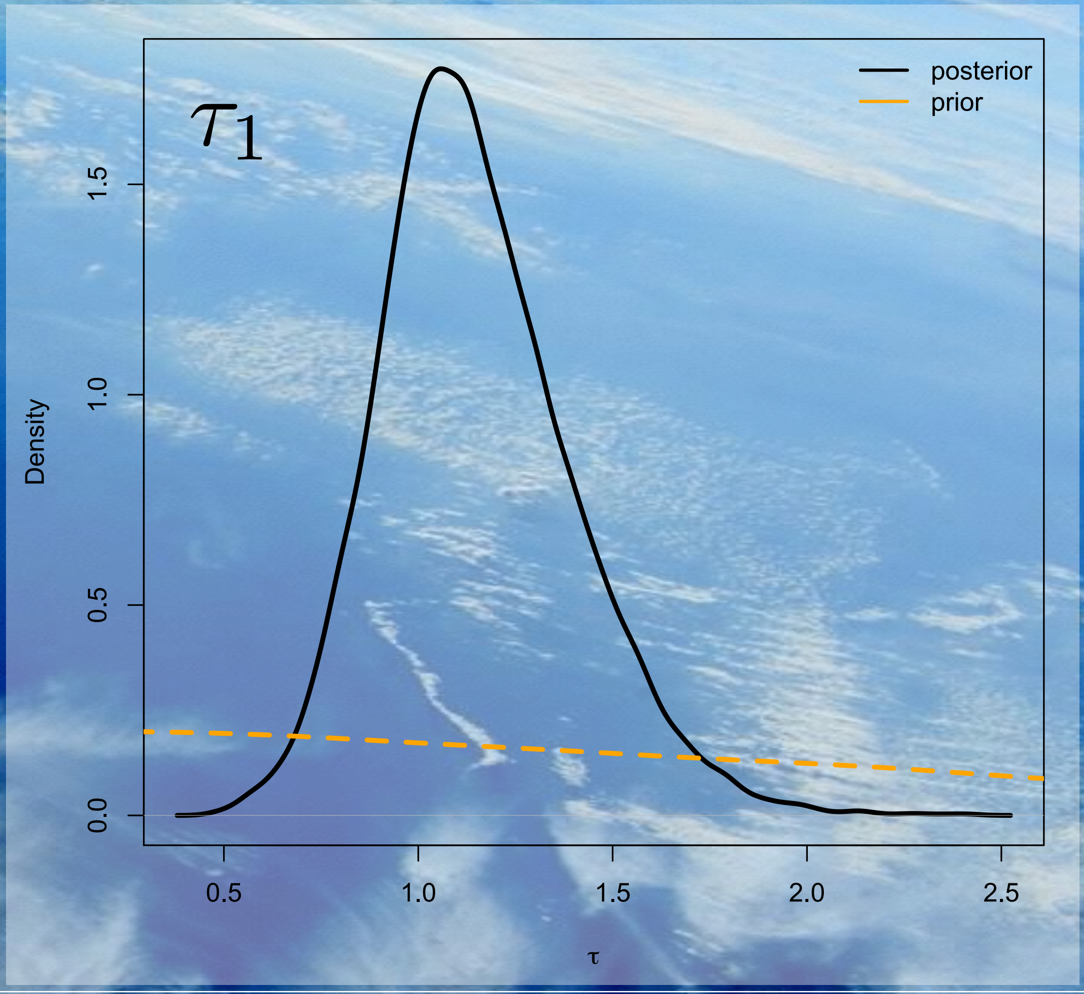
# Hyperpriors are very weak!

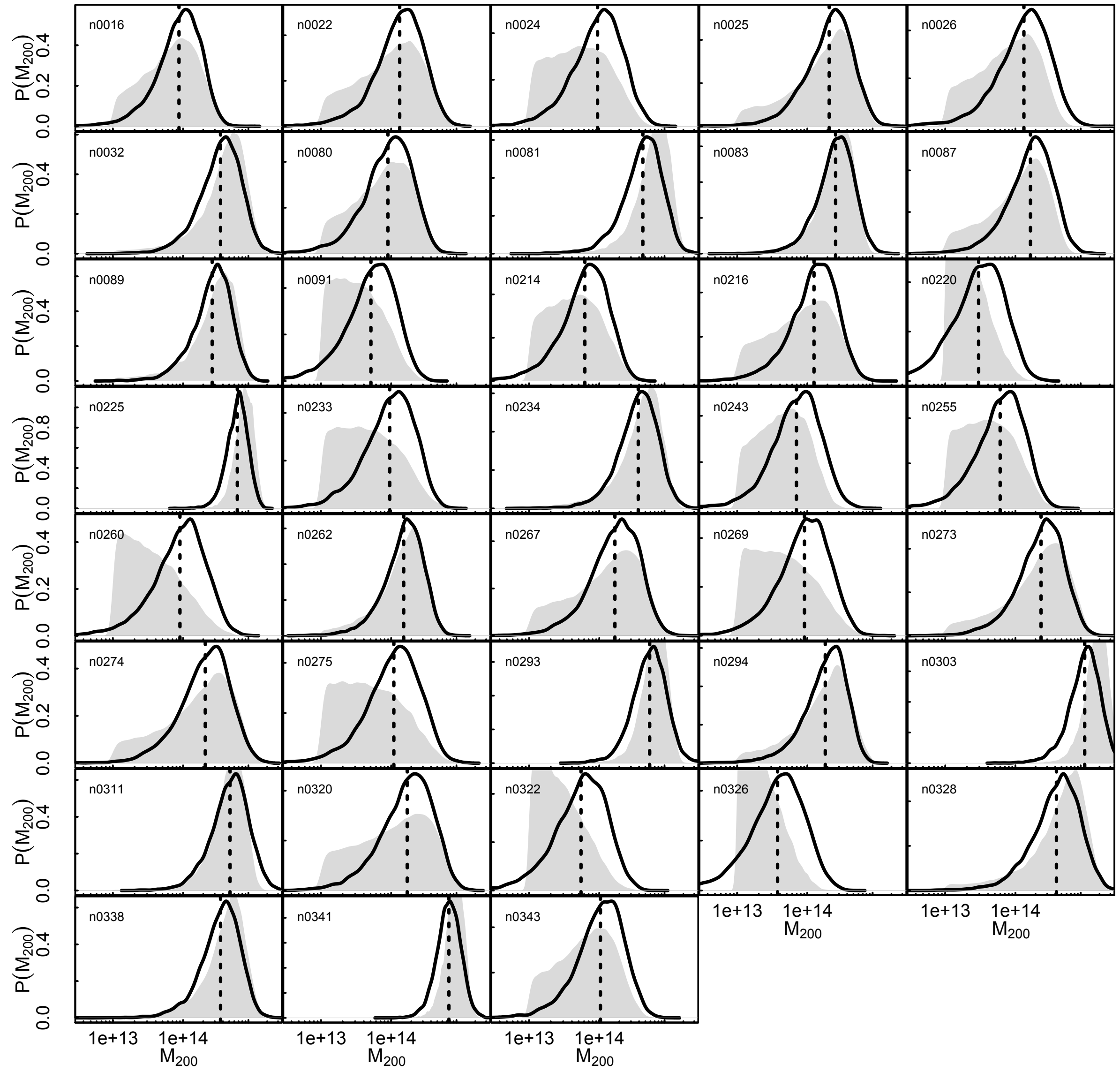
$$\text{shrinkage} = 1 - \frac{\sigma^2_{posterior}}{\sigma^2_{prior}}$$

shrinkage = 0.92




shrinkage = 0.98






# non hierarchical vs hierarchical

- Posteriors in good agreement
- **No upper limit** measurements w/ hierarchical method

# Hierarchical cluster Cosmology

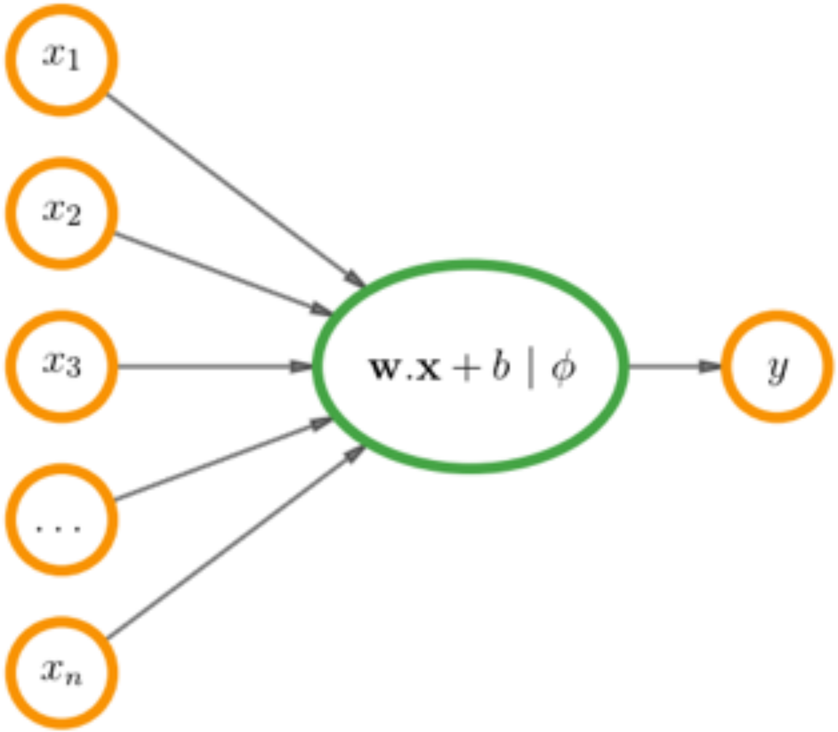
Sampling cluster masses from the true halo mass function and not gaussian approximation

# Hierarchical cluster Cosmology

Sampling cluster masses from the true halo mass function and not gaussian approximation

Problem: **halo mass function** is **nasty**!!!
Lots of **multi-dimensional improper integrals**!

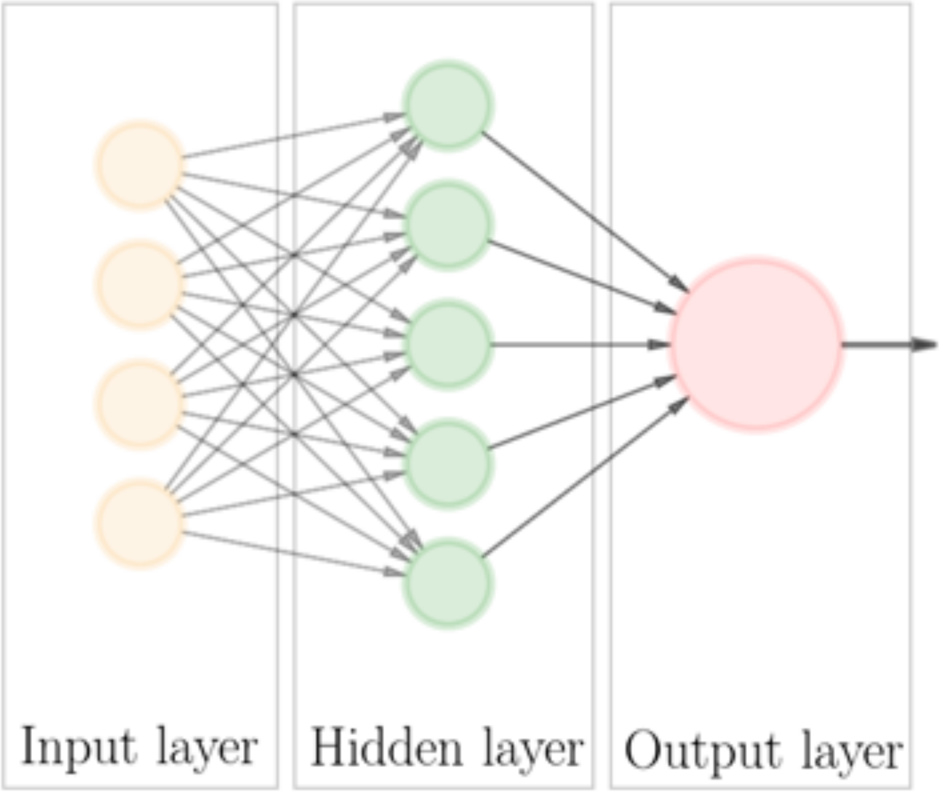
# PDF emulator with mixture density network.




- Each neuron node consists of:
- weights
- bias
- activation : e.g. tanh, ReLU etc

# PDF emulator with mixture density network.

model
parameters




gaussian mixture
components

# PDF emulator with mixture density network.

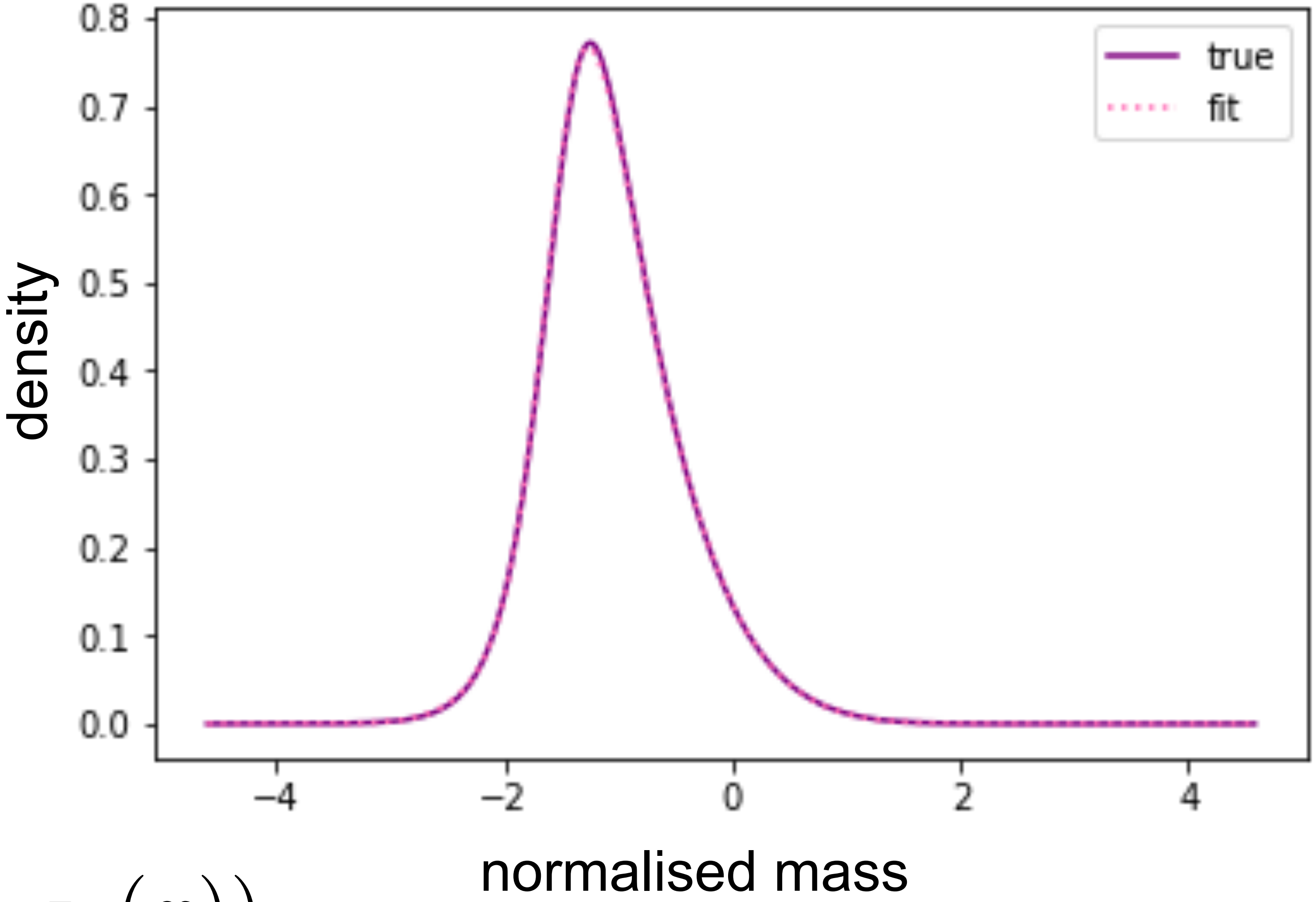
1. **Sample cosmological parameters**, cluster redshift and selection function parameters
2. **Generate lots of simulations** of the normalised PDF: halo mass function + selection function
3. Use tensorflow to **train neural network** by minimising the loss between the true PDF and the sum of the gaussian mixtures
4. **Save neural network weights** to use STAN

- 20000 simulations

$$\phi = [\Omega_M, \sigma_8, \ln M_{cut}, d\ln M_{cut}, z]$$

- 8 Mixture components
- 2 hidden layers
- each 30 nodes

$$P(Y=y|X=x) = \sum_{k=1}^{K} \pi_k(x)\mathcal{N}(y, \mu_k(x), \sigma_k(x))$$



# STAN code:

```
data{
    int<lower=0> N; //number of clusters
    int<lower=0> Nr; //number of radial bins

    vector[Nr] rs[N];
    vector[Nr] obs_shear[N];
    vector[Nr] sigma_shear[N];
    real<lower=0> zls[N];

    //nn weights
    int<lower=0> nout; //size of neural output
    int<lower=0> kmix; //number of mixtures
    int<lower=0> nparam; //number of parameters
    int<lower=0> nn; //number of neurons
    matrix[nparam,nn] wh;
    row_vector[nn] bh;
    matrix[nn,nn] wh1;
    row_vector[nn] bh1;
    matrix[nn, kmix*3] wo;
    row_vector[kmix*3] bo;
}
```

```
parameters{
    real<lower=0, upper=1> s8;
    real<lower=0, upper=1> Om;
    real<lower=-10, upper=10> lnMc;
    real<lower=0, upper=10> dlnM;
    vector[N] logM;
    real Bcm;
    real Acm;
}

transformed parameters{
    vector[Nr] true_shear[N];
    vector[N] mass;
    vector[N] conc;

    for(i in 1:N){
        mass[i] = exp(logM[i]+log(1e14));
        conc[i] = exp(Bcm*(logM[i] + log(1e14) - log(2e12)) + Acm;
        true_shear[i] = nfw(rs[i], mass[i], conc[i], zls[i], Om);
    }

}
```

# STAN code:

```
model{
    vector[kmix] theta;
    vector[kmix] sigma;
    vector[kmix] mu;
    real lps[kmix];

    //priors
    s8 ~ normal(0.7,0.2);
    Om ~ normal(0.4,0.2);
    lnMc ~ normal(-0.1,0.15);
    dlnM ~ normal(0, 0.2);
    Bcm ~ normal(0,0.05);
    Acm ~ normal(1.6,0.1);
```

```
    for (n in 1:N){

        {
        vector[kmix*3] output;
        vector[kmix*3] out_params;
        row_vector[nparam] x = [s8, Om, lnMc, dlnM, zls[n]];

        output = get_nn_output(x, nn, nparam, nout, wh, bh, wh1, bh1, wo, bo );
        out_params = get_mix_coeff(output, kmix);
        theta = out_params[1:kmix];
        sigma = out_params[(kmix+1):(2*kmix)];
        mu = out_params[(2*kmix+1):(3*kmix)];
        }
       for (k in 1:kmix){
           lps[k] = log(theta[k]) + normal_lpdf(logM[n] | mu[k], sigma[k]);
        }
        target += log_sum_exp(lps);
        obs_shear[n] ~ normal(true_shear[n], sigma_shear[n]);
    }
}
```

# Tests on toy simulations:

- 557 galaxy clusters

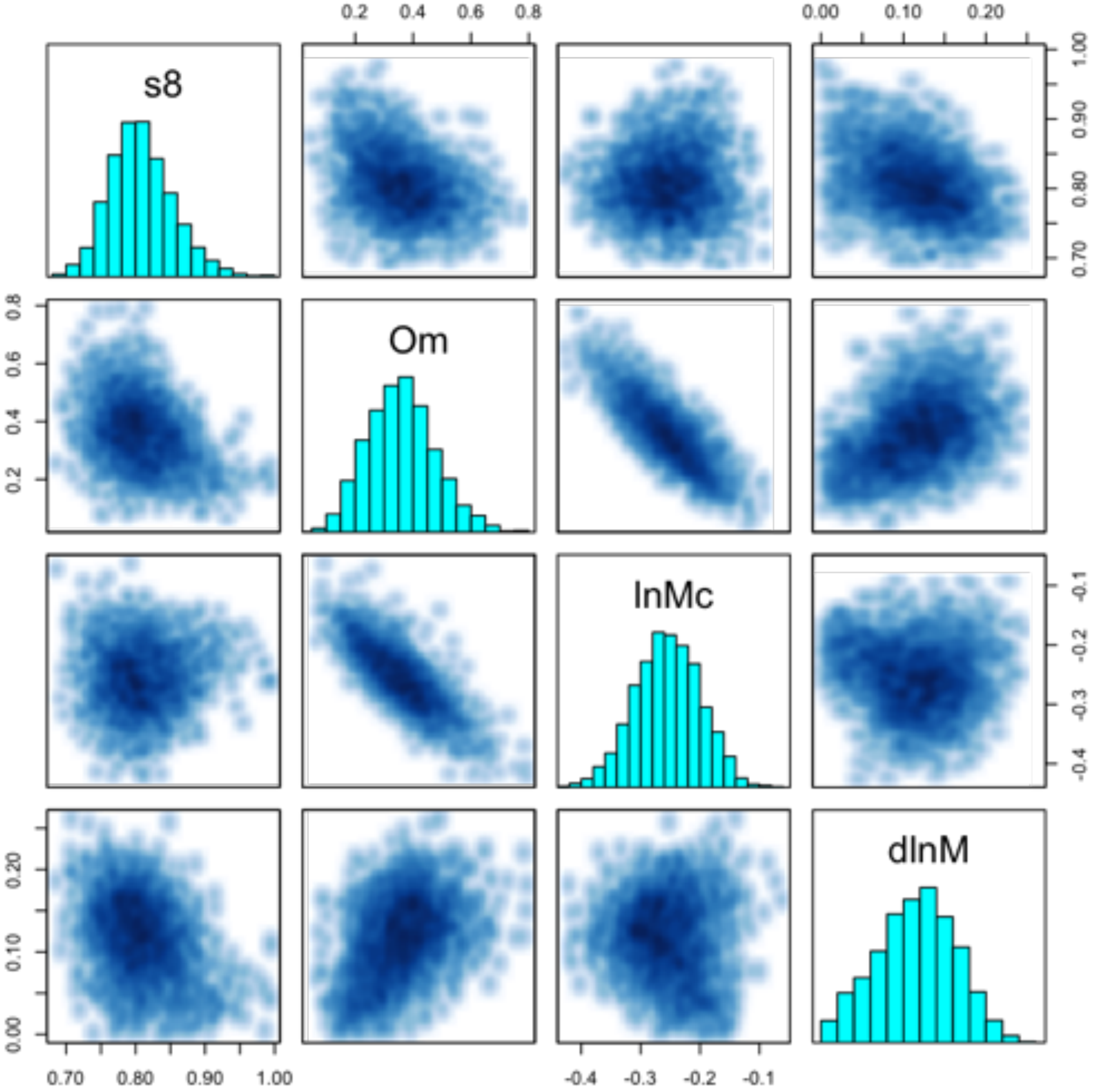
Truth:

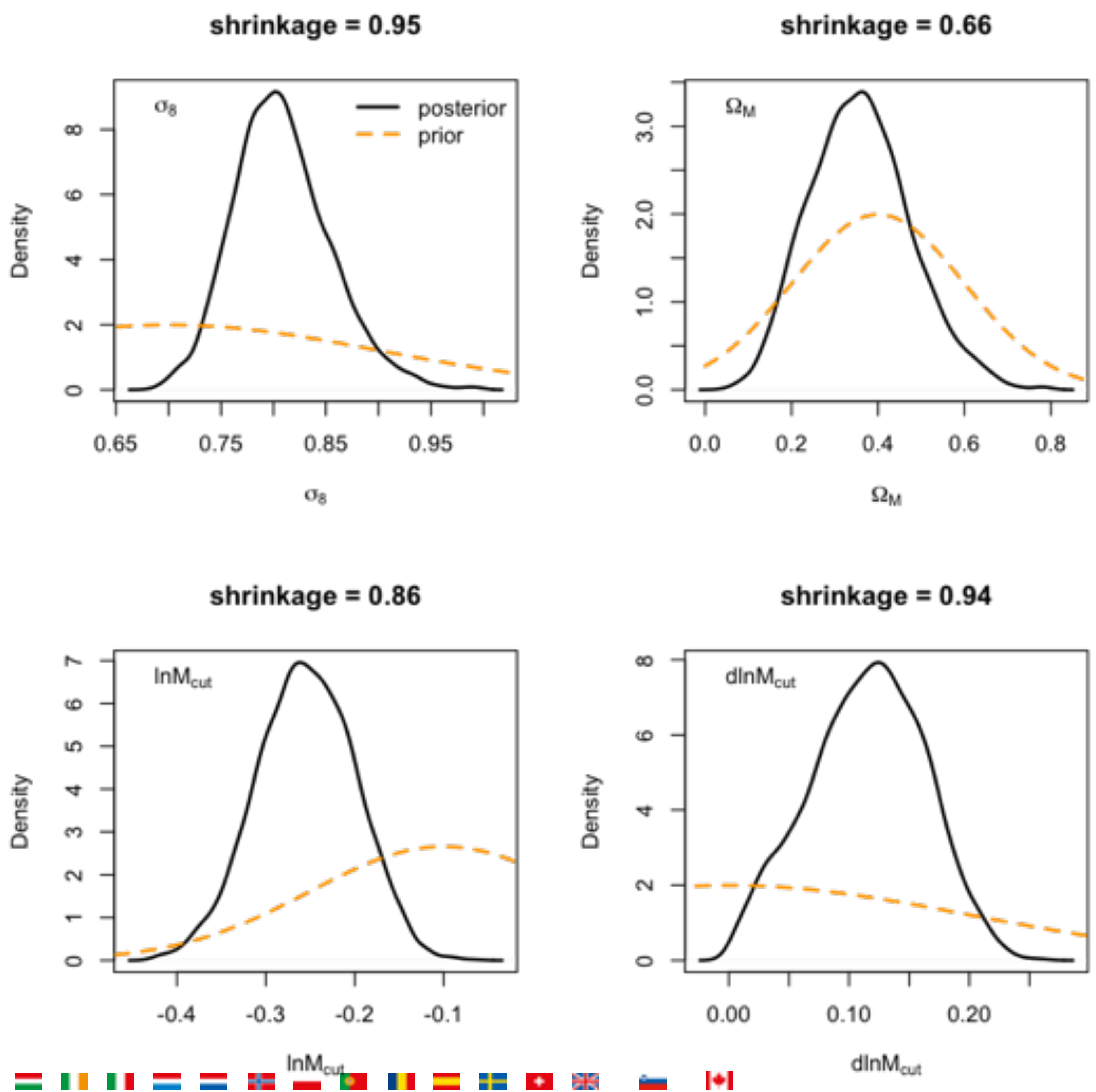
$$\sigma_8 = 0.83$$
$$\Omega_M = 0.3$$
$$\ln M_{cut} = -0.23$$
$$d \ln M_{cut} = 0.1$$

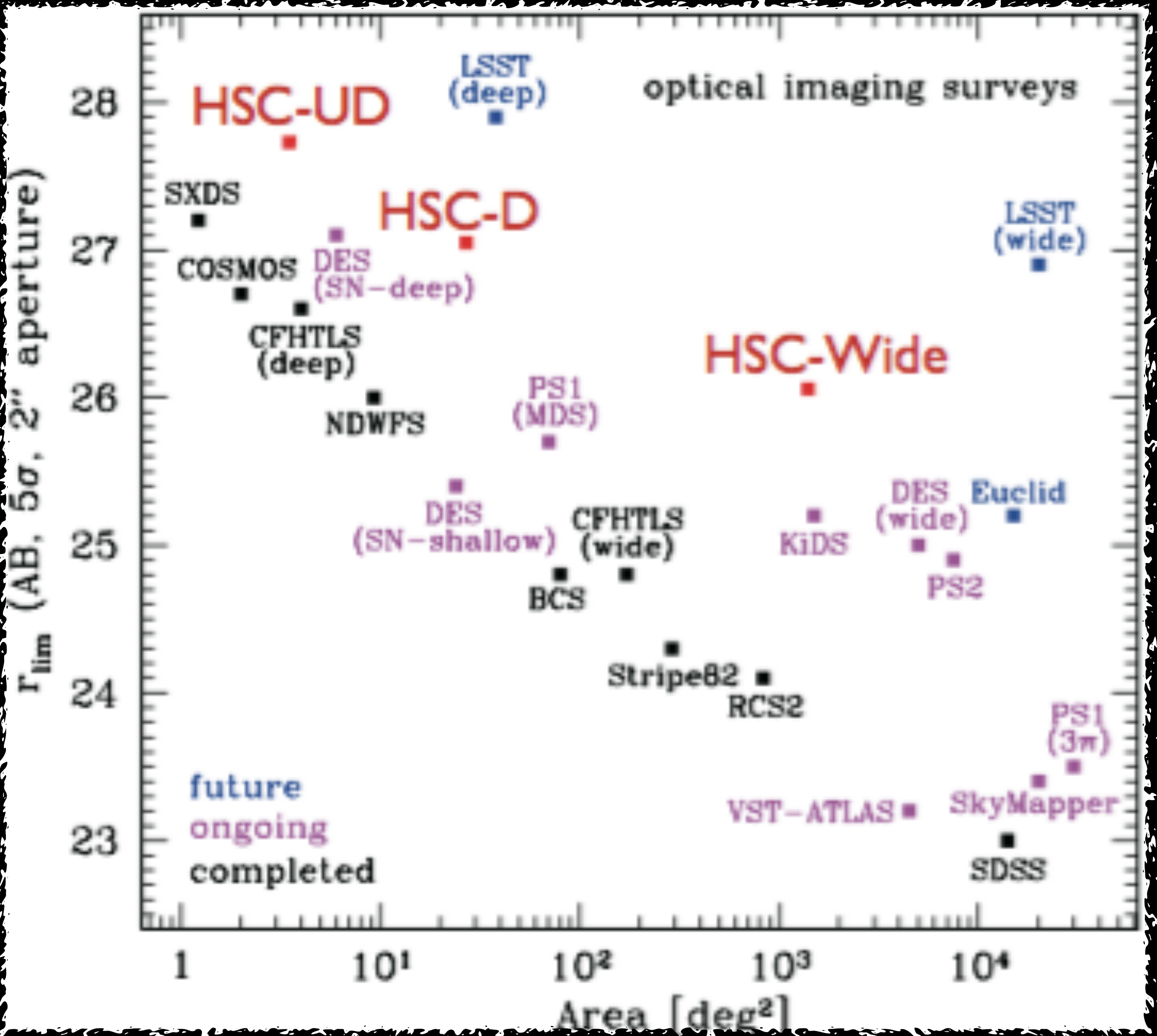
Inference for Stan model: stan_NFW_hierwl_cosmo.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

| | mean | se_mean | sd | 2.5% | 25% | 50% | 75% | 97.5% | n_eff | Rhat |
|---|---|---|---|---|---|---|---|---|---|---|
| s8 | 0.81 | 0.00 | 0.05 | 0.73 | 0.78 | 0.80 | 0.84 | 0.91 | 617 | 1.00 |
| Om | 0.36 | 0.01 | 0.12 | 0.15 | 0.28 | 0.36 | 0.43 | 0.61 | 104 | 1.03 |
| lnMc | -0.25 | 0.00 | 0.06 | -0.37 | -0.29 | -0.25 | -0.22 | -0.15 | 125 | 1.02 |
| dlnM | 0.12 | 0.00 | 0.05 | 0.02 | 0.08 | 0.12 | 0.15 | 0.20 | 229 | 1.02 |

Samples were drawn using NUTS(diag_e) at Wed Aug 29 08:46:07 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).


s8
Om
lnMc
dlnM
0.2 0.4 0.6 0.8
0.00 0.10 0.20
0.70 0.80 0.90 1.00
-0.4 -0.3 -0.2 -0.1


shrinkage = 0.95
shrinkage = 0.66
shrinkage = 0.86
shrinkage = 0.94
posterior
prior
Density
$\sigma_8$
$\Omega_M$
$\ln M_{cut}$
$d\ln M_{cut}$
0.65
0.75
0.85
0.95
0.0
0.2
0.4
0.6
0.8
1.0
2.0
3.0
-0.4
-0.3
-0.2
-0.1
0.00
0.10
0.20




- Upcoming WL will be wide and deep
- **LOTS MORE DATA!**
- need to make optimal use of information available

Euclid will detect **2x10⁵** (0.2 ≤ z ≤ 2) clusters!

# Take away message

Cosmology with clusters is difficult - the data are noisy!

Hierarchical inference can help you:

Correctly propagate uncertainties in analyses

Improve constraints on noisy data

Self -consistently determine the amount of stacking