シミュレーションデータベースと統計学

吉田直紀, 西道啓博, 大木平

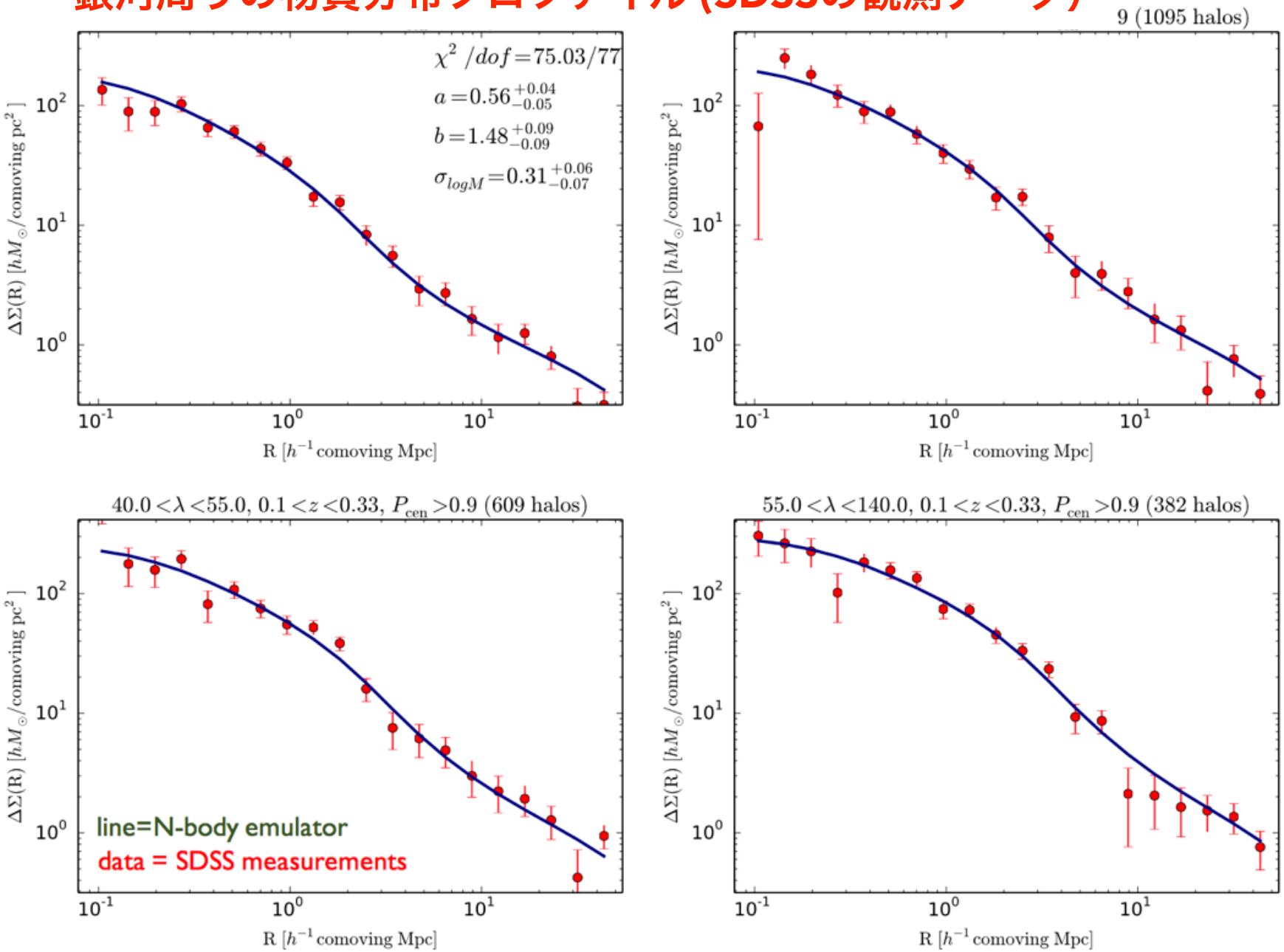
(KAVLI IPMU)

- Cosmology with Subaru HSC survey
- A large number of cosmological N-body simulations
- Parameter space exploration and the Gaussian process

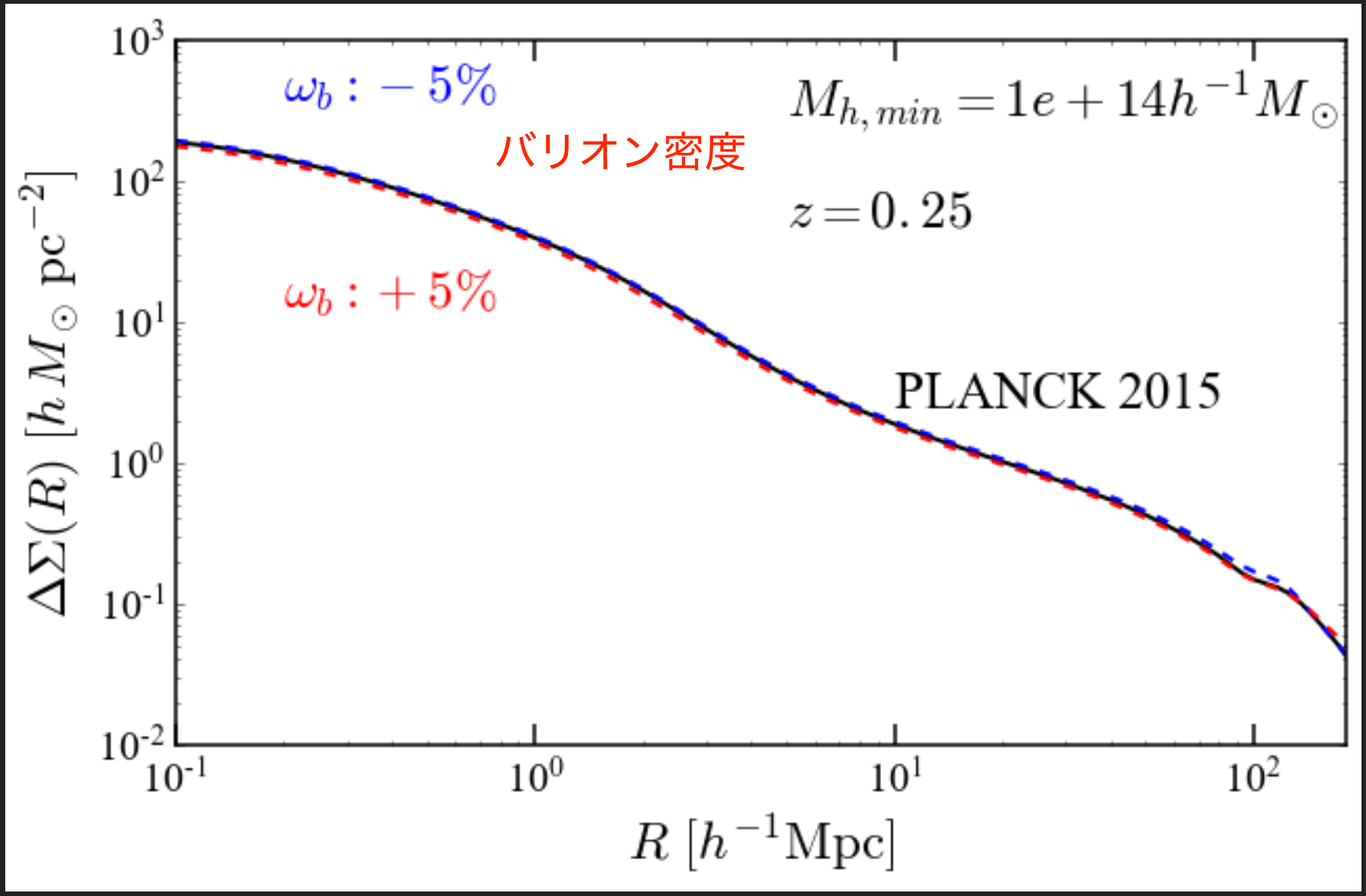
Observational data and forward modeling

銀河周りの物質分布プロファイル (SDSSの観測データ)

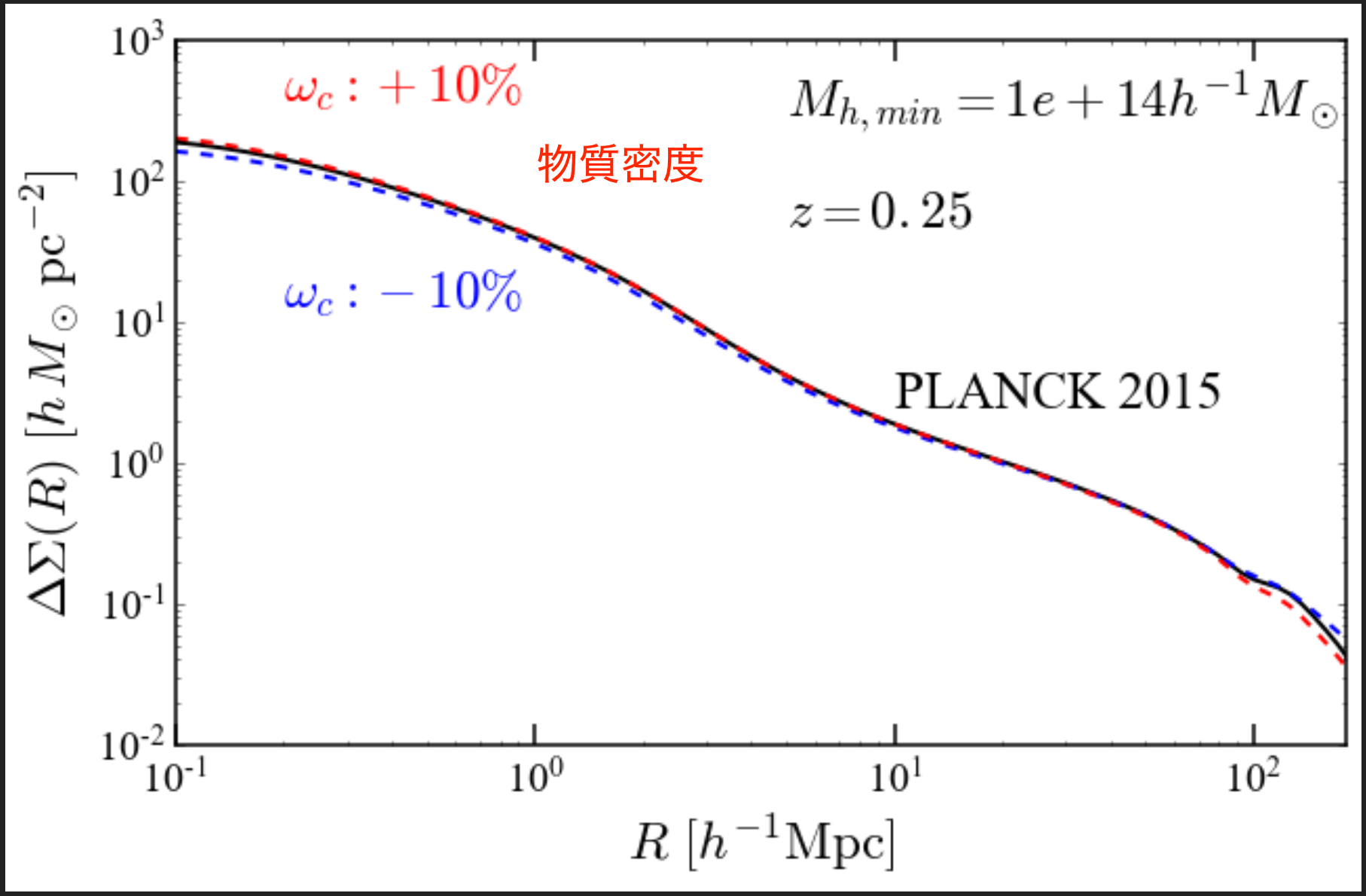
Average DM distribution around massive clusters



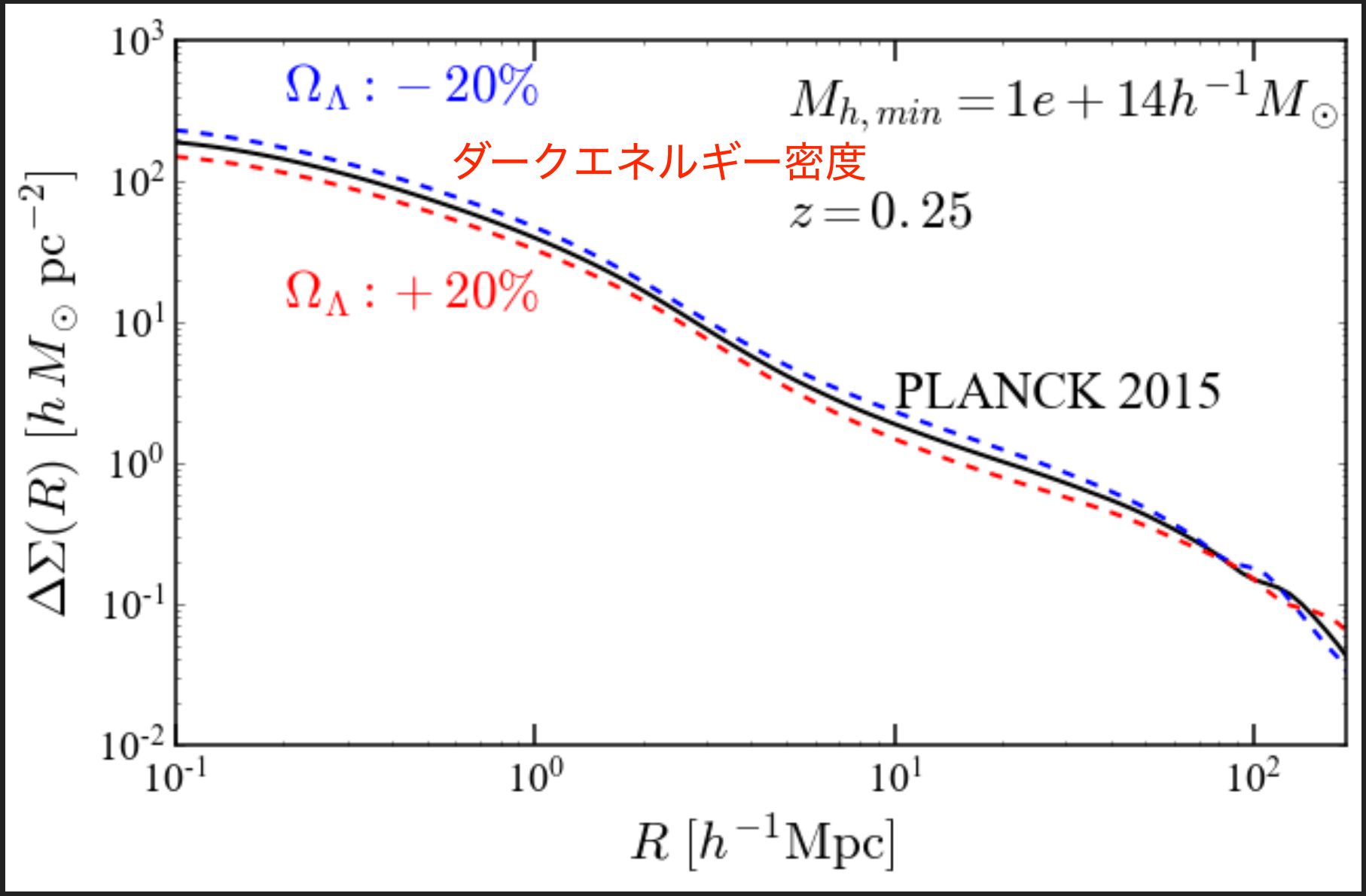
COSMOLOGICAL PARAMETER DEPENDENCE



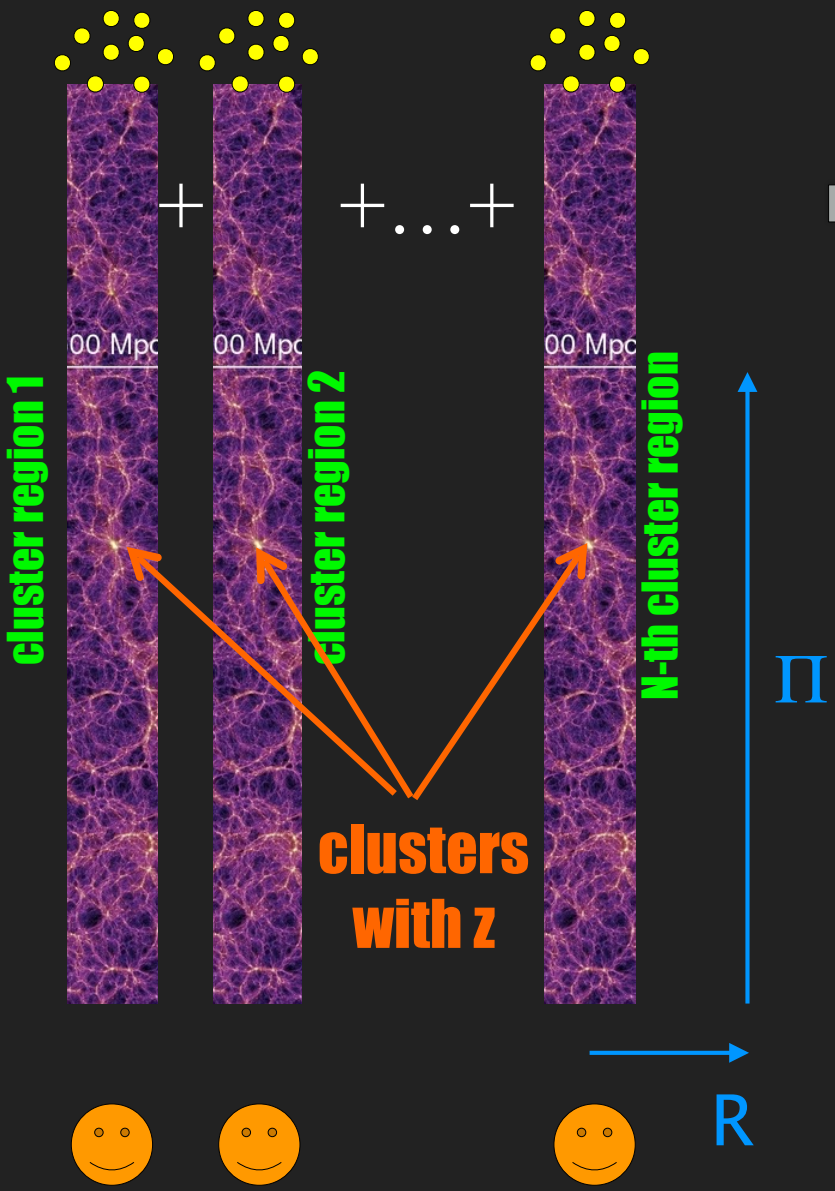
COSMOLOGICAL PARAMETER DEPENDENCE

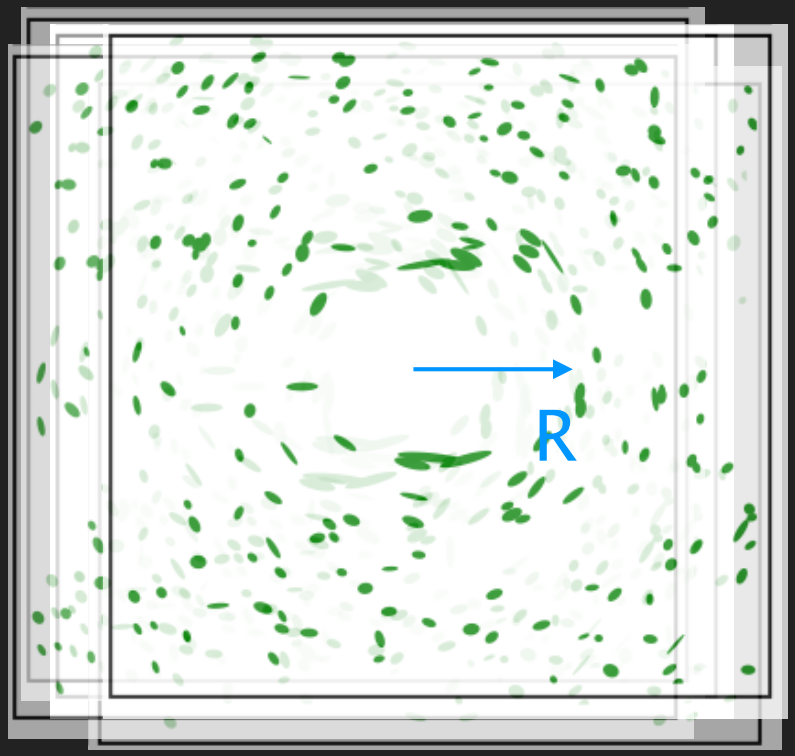


COSMOLOGICAL PARAMETER DEPENDENCE

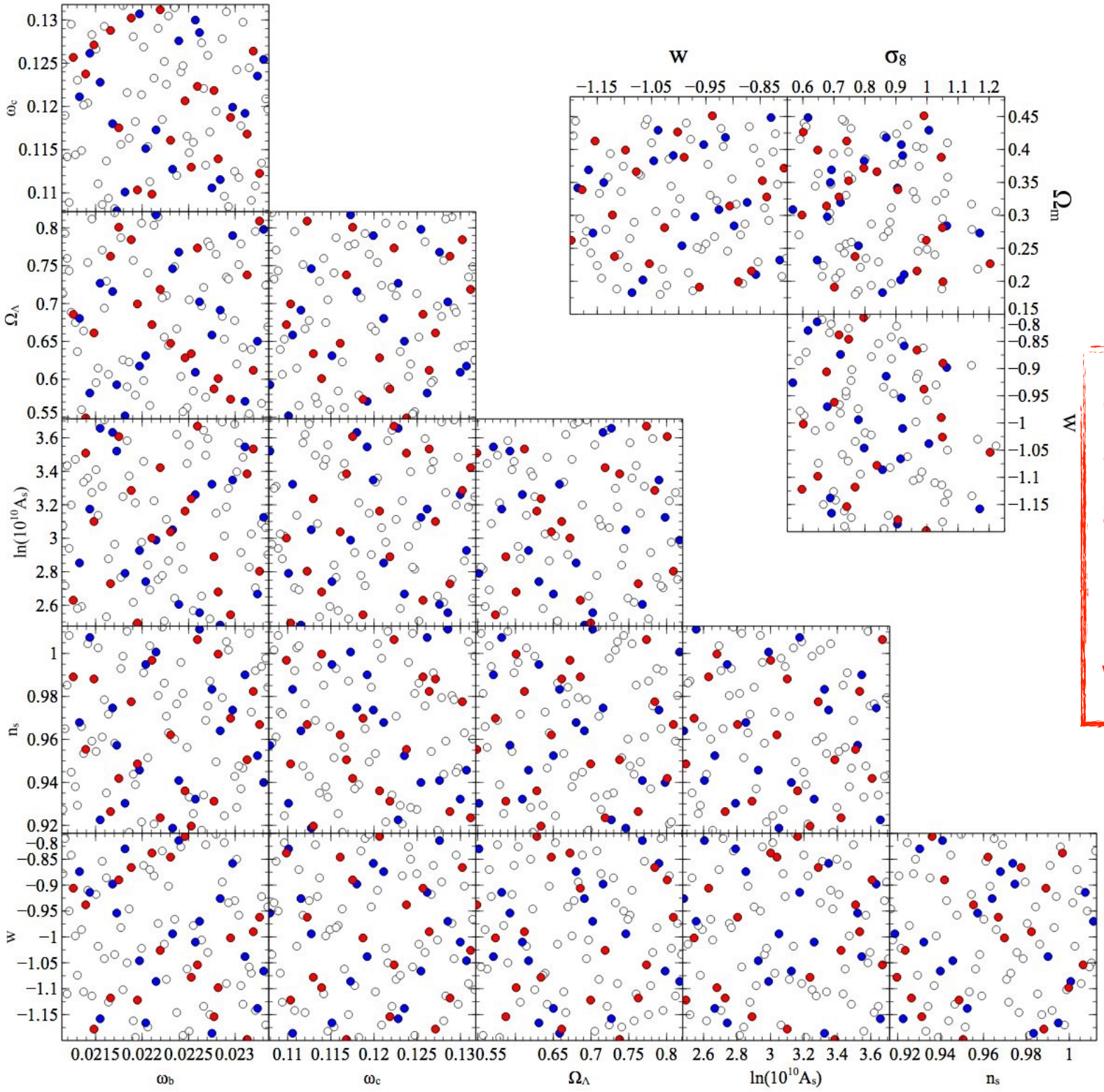


HSC GALAXY-GALAXY LENSING





- Known lens objects
  - host halo properties well known
  - photo-z accuracy not important
  - can measure Σ as a function of scale R instead of angle on the sky



$\omega_b = \Omega_b h^2$: ±5%

$\omega_c = \Omega_c h^2$: ±10%

Ω_Λ: ±20%

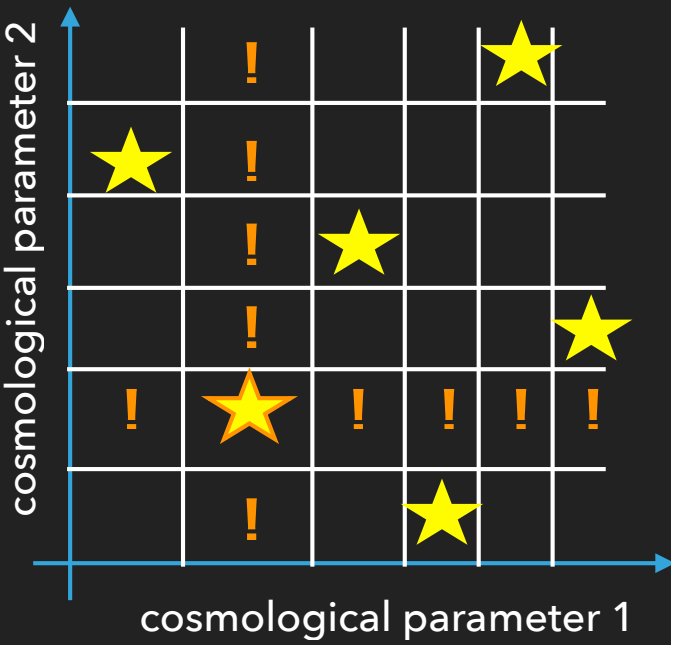
$\ln(10^{10}\, A_s)$: ±20%

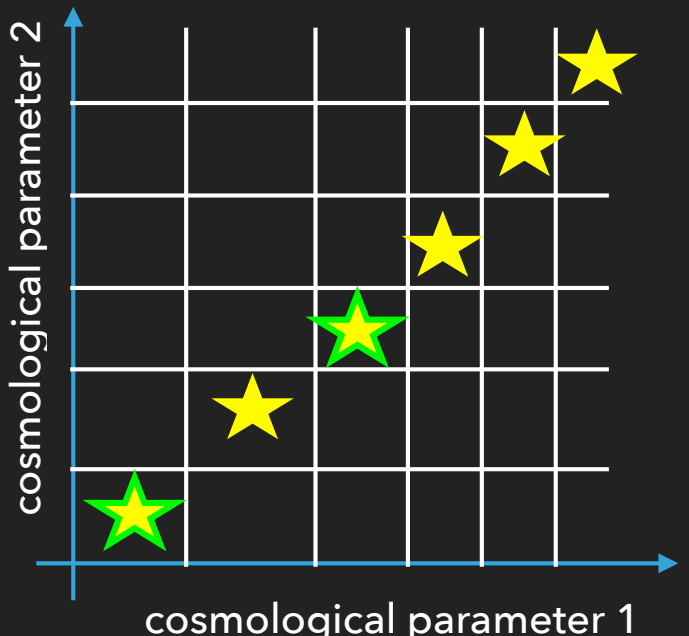
n_s: ±5%

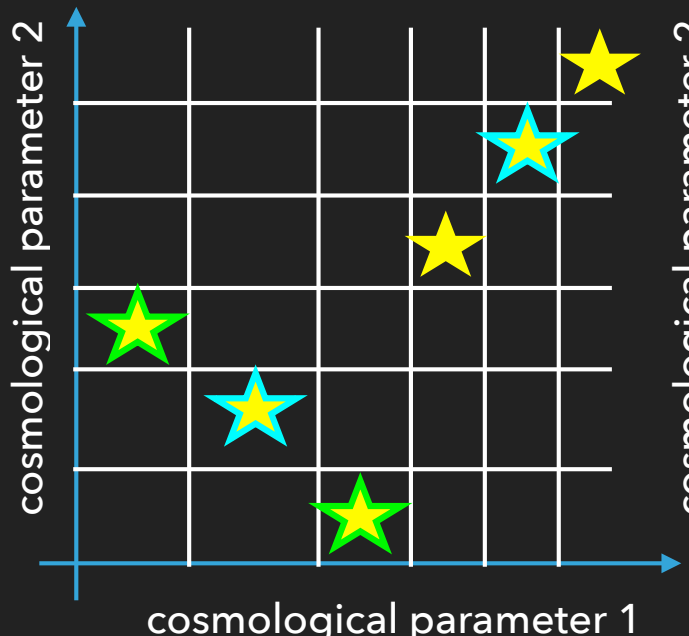
w: ±20%

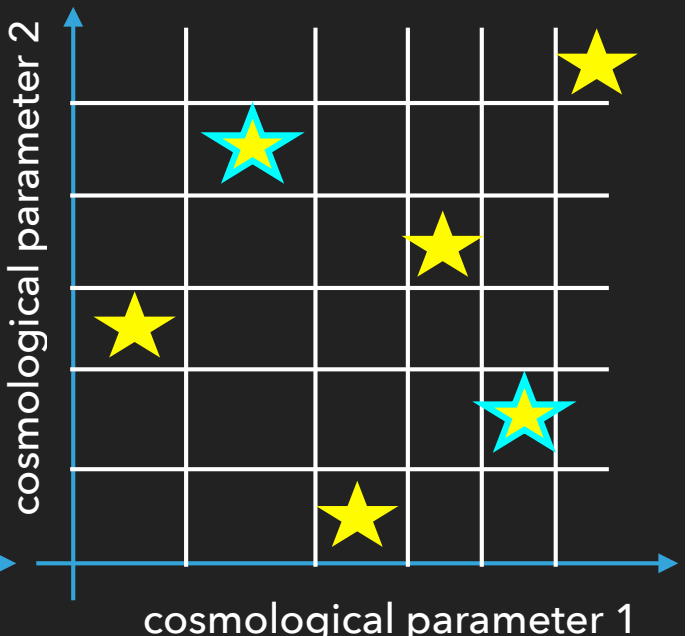
EFFICIENT SAMPLING IN MULTI DIMENSIONAL SPACE: LATIN HYPERCUBE

- Each sample is the only one in each axis-aligned hyperplane containing it
  - One can find many realizations of such design (ex. diagonal design)
  - Impose additional condition such as "the sum of the distances to the nearest design point is maximal" (maximin distance)









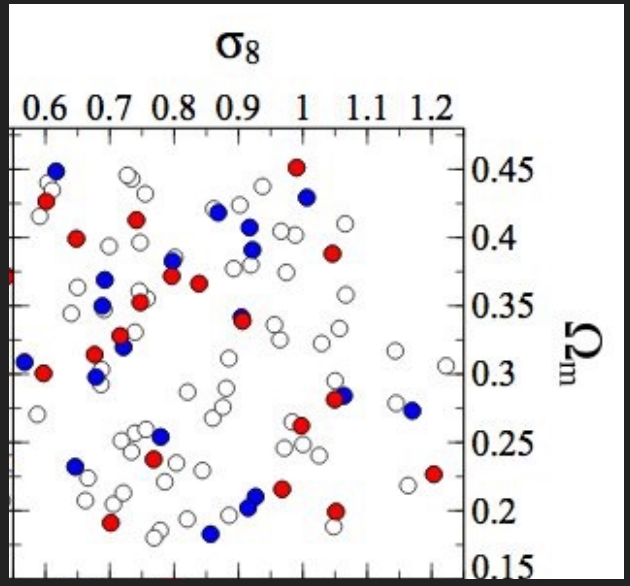
EFFICIENT SAMPLING IN MULTI DIMENSIONAL SPACE: LATIN HYPERCUBE

Simulation spec

- ✓ N of particles: 2048^3
- ✓ Size: 5 billion lightyears
- ✓ 21 outputs per model
- ✓ 5 Tbyte data per model

$\omega_b = \Omega_b h^2$: ±5%
$\omega_c = \Omega_c h^2$: ±10%
Ω_Λ: ±20%
$\ln(10^{10} A_s)$: ±20%
n_s: ±5%
w: ±20%

84 sims are already available

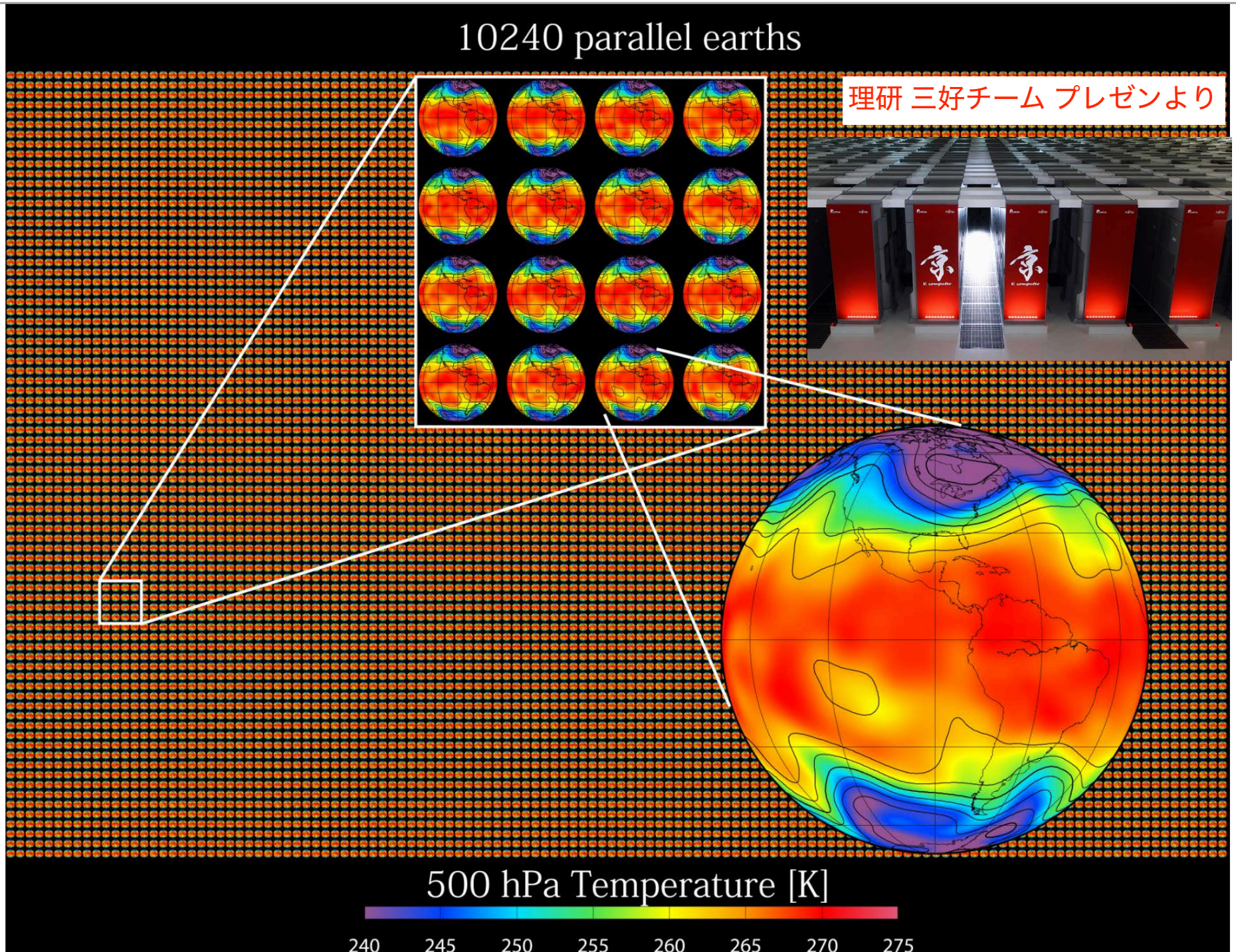


parameter space sweep

- "sliced" LH design (Ba, Brenneman & Myers '15)
- generate >100 sample eventually
- maxi-min distance LH design for each 20 model set (e.g., red/blue points)

- two suites of runs
  - keep the initial random number seed (20 done)
  - different seeds (40)
    - **red: emulator**
    - **blue: validation**

SUPERCOMPUTING AND BIG DATA

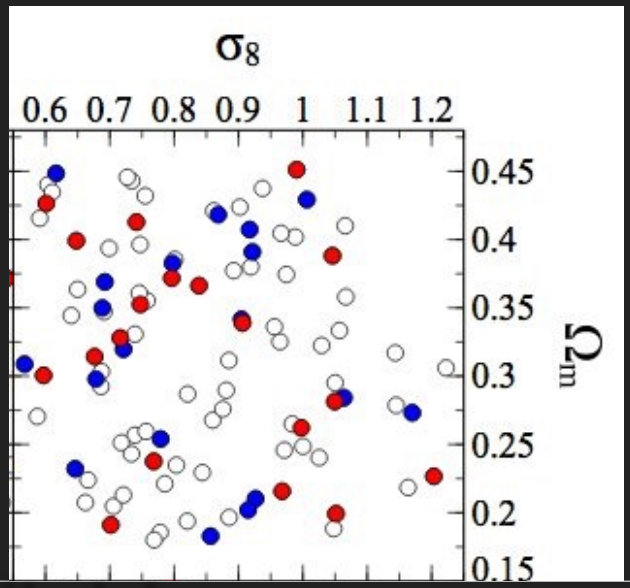


EFFICIENT SAMPLING IN MULTI DIMENSIONAL SPACE: LATIN HYPERCUBE

fiducial model

- PLANCK15 (flat ΛCDM)
- 24 realizations done
- assess statistical error
- emulator accuracy check

$\omega_b = \Omega_b h^2$: ±5%
$\omega_c = \Omega_c h^2$: ±10%
Ω_Λ: ±20%
$\ln(10^{10} A_s)$: ±20%
n_s: ±5%
w: ±20%



119 sims are already available

parameter space sweep

- "sliced" LH design (Ba, Brenneman & Myers '15)
- generate >200 sample eventually
- maxi-min distance LH design for each 20 model set (e.g., red/blue points)
- two suites of runs
  - keep the initial random number seed (20 done)
  - different seeds (40)
    - **red: emulator**
    - **blue: validation**

SIMULATION SPEC

- ✓ N of particles: 2048^3
- ✓ box size: $1h^{-1}$Gpc

  resolve a 10^{12} $h^{-1}M_{solar}$ halo with ~100 particles

- ✓ 2nd-order Lagrangian PT initial condition @ $z_{in}=59$

  (vary slightly for different cosmologies to keep the RMS displacement about 25% of the inter-particle separation)

- ✓ Tree-PM force by L-Gadget2 (w/ 4096^3 PM mesh)

- ✓ 21 outputs in $0 \leqq z \leqq 1.5$ (equispaced in linear growth factor)

- ✓ Data compression (256GB -> 48GB par snapshot)
  - ✓ positions -> displacement (16 bits par dimension; accuracy ~$1h^{-1}$kpc)
  - ✓ velocity: discard after halo identification
  - ✓ ID: rearrange the order of particles by ID and then discard
  - ✓ already consuming ~200TB in half a year (~observational data)

SIMULATION PIPELINE

~2TB, 2+2days / 1 run

on-the-fly analysis stream

prediction

density estimate

final catalog

data analysis

time evolution

mass/profile determination

Gadget2

rockstar

IC generator

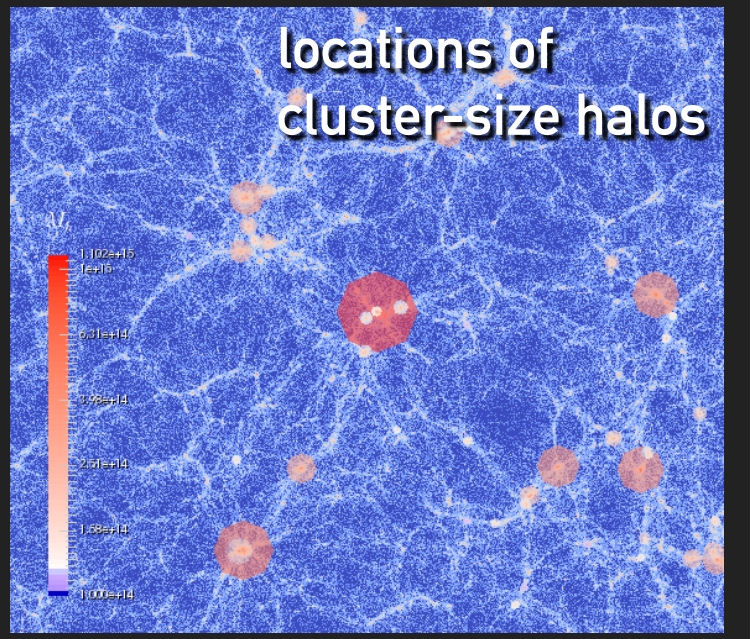
subfind

create "universes"

FOF (on the fly)

object identification

on supercomputer
(XC30 @NAOJ; using 648 cores)

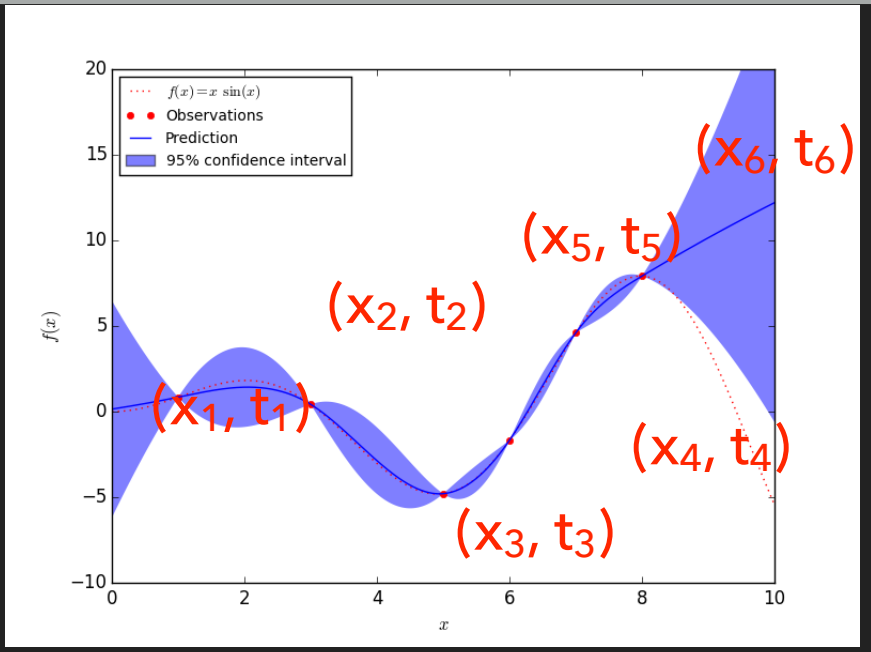


GAUSSIAN PROCESS

- A machine-learning technique to do inference in function space
  - non-parametic Bayesian inference
  - nonlinear regression analysis
- Basic quantities
  $f(x) \sim P[\mu(x), k(x, x')]$
  - mean function (cf. mean)
  - covariance function (cf. variance)
- covariance function is characterized by a simple function with several hyper parameters

ex. $$C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \theta_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{I}\frac{(x_i - x_i')^2}{r_i^2}\right] + \theta_2.$$

"length scale" r_i for each x_i



✓ infer hyper parameters θ from training data (x_i, t_i)

✓ Given a point x_{N+1}, infer t_{N+1} from θ and (x_i, t_i)

$$P(t_{N+1} \mid \mathbf{t}_N) \propto \exp\left[-\frac{1}{2}\left[\mathbf{t}_N\ t_{N+1}\right]\mathbf{C}_{N+1}^{-1}\begin{bmatrix}\mathbf{t}_N \\ t_{N+1}\end{bmatrix}\right]$$

answer:

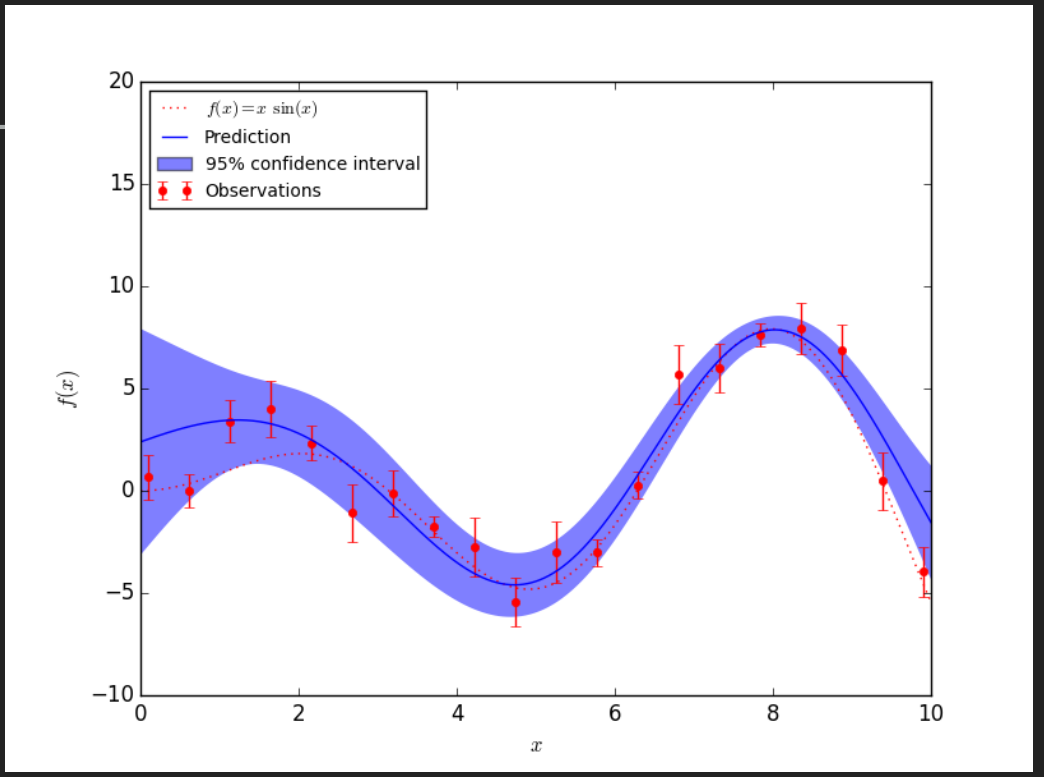
$$\begin{aligned}\hat{t}_{N+1} &= \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{t}_N \\ \sigma_{\hat{t}_{N+1}}^2 &= \kappa - \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{k}.\end{aligned}$$

GAUSSIAN PROCESS

- A machine-learning technique to do inference in function space
  - non-parametic Bayesian inference
  - nonlinear regression analysis
- Basic quantities
  $f(x) \sim P[\mu(x), k(x, x')]$
  - mean function (cf. mean)
  - covariance function (cf. variance)
- covariance function is characterized by a simple function with several hyper parameters

ex. $$C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \theta_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{I}\frac{(x_i - x_i')^2}{r_i^2}\right] + \theta_2.$$

"length scale" r_i for each x_i



✓ infer hyper parameters θ from training data (x_i, t_i)

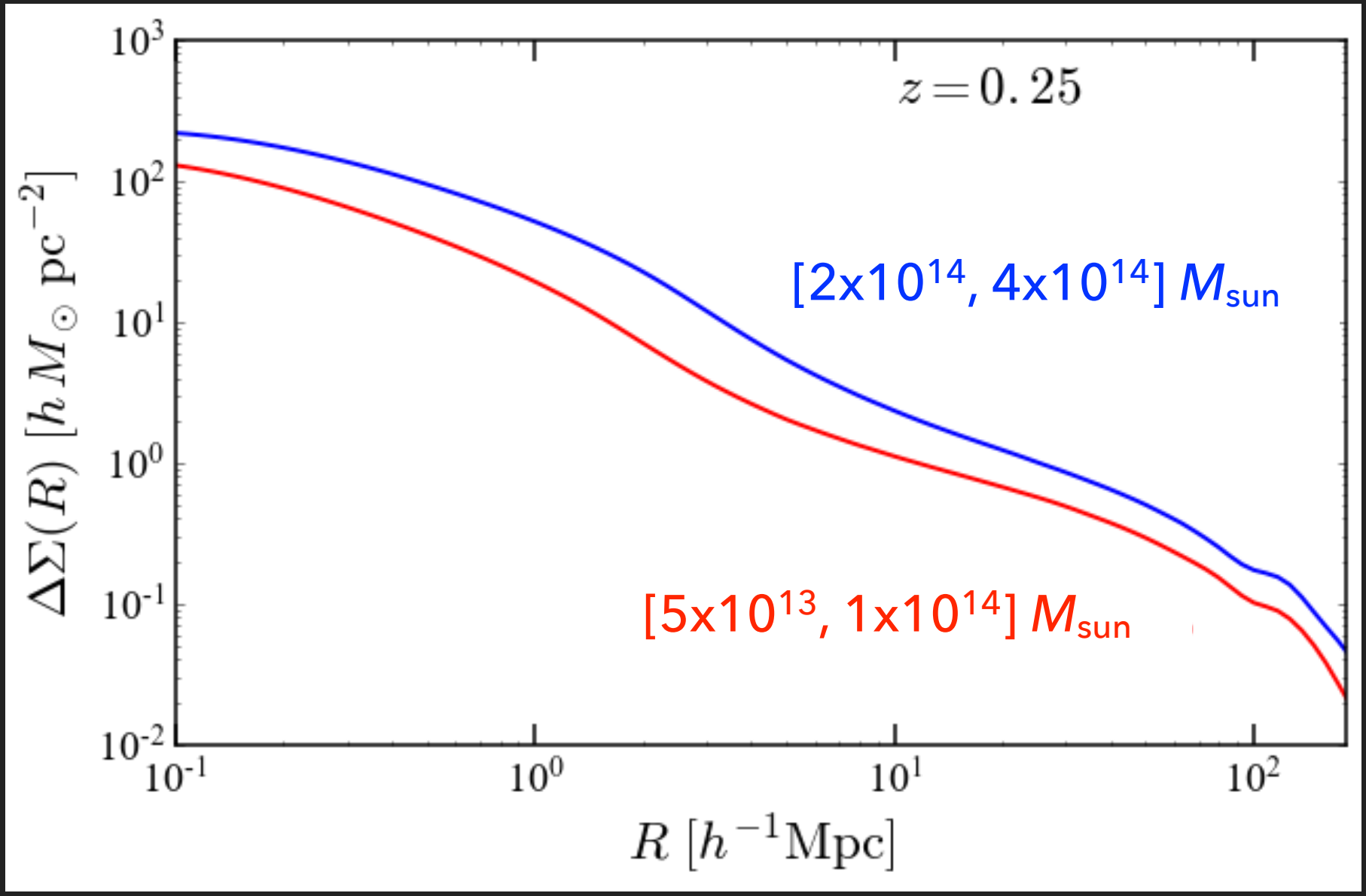
✓ Given a point x_{N+1}, infer t_{N+1} from θ and (x_i, t_i)

$$P(t_{N+1} \mid \mathbf{t}_N) \propto \exp\left[-\frac{1}{2}\left[\mathbf{t}_N\ t_{N+1}\right]\mathbf{C}_{N+1}^{-1}\begin{bmatrix}\mathbf{t}_N \\ t_{N+1}\end{bmatrix}\right]$$

answer:

$$\begin{aligned}\hat{t}_{N+1} &= \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{t}_N \\ \sigma_{\hat{t}_{N+1}}^2 &= \kappa - \mathbf{k}^T\mathbf{C}_N^{-1}\mathbf{k}.\end{aligned}$$

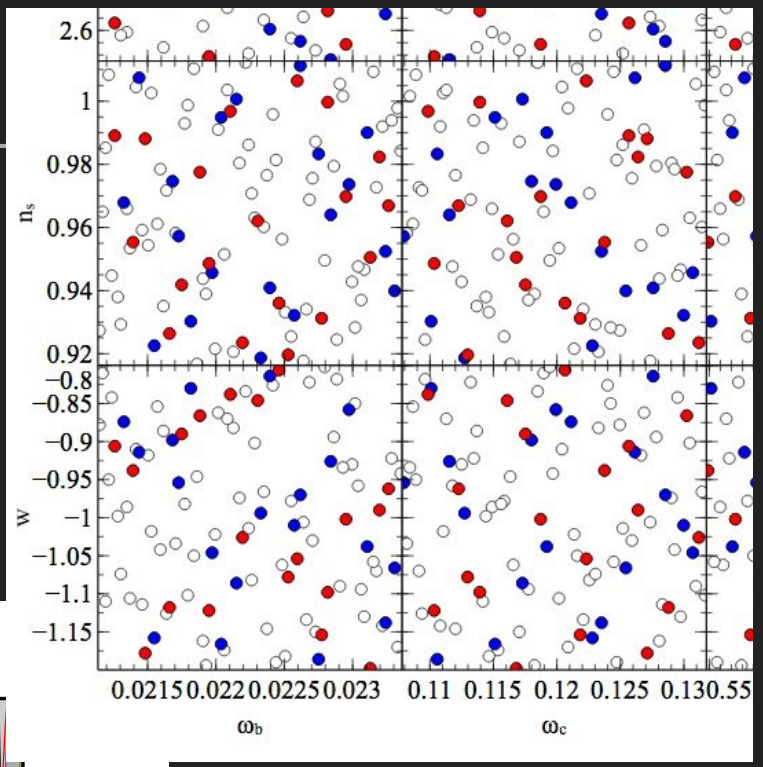
OUR ΔΣ EMULATOR DEMONSTRATIONS



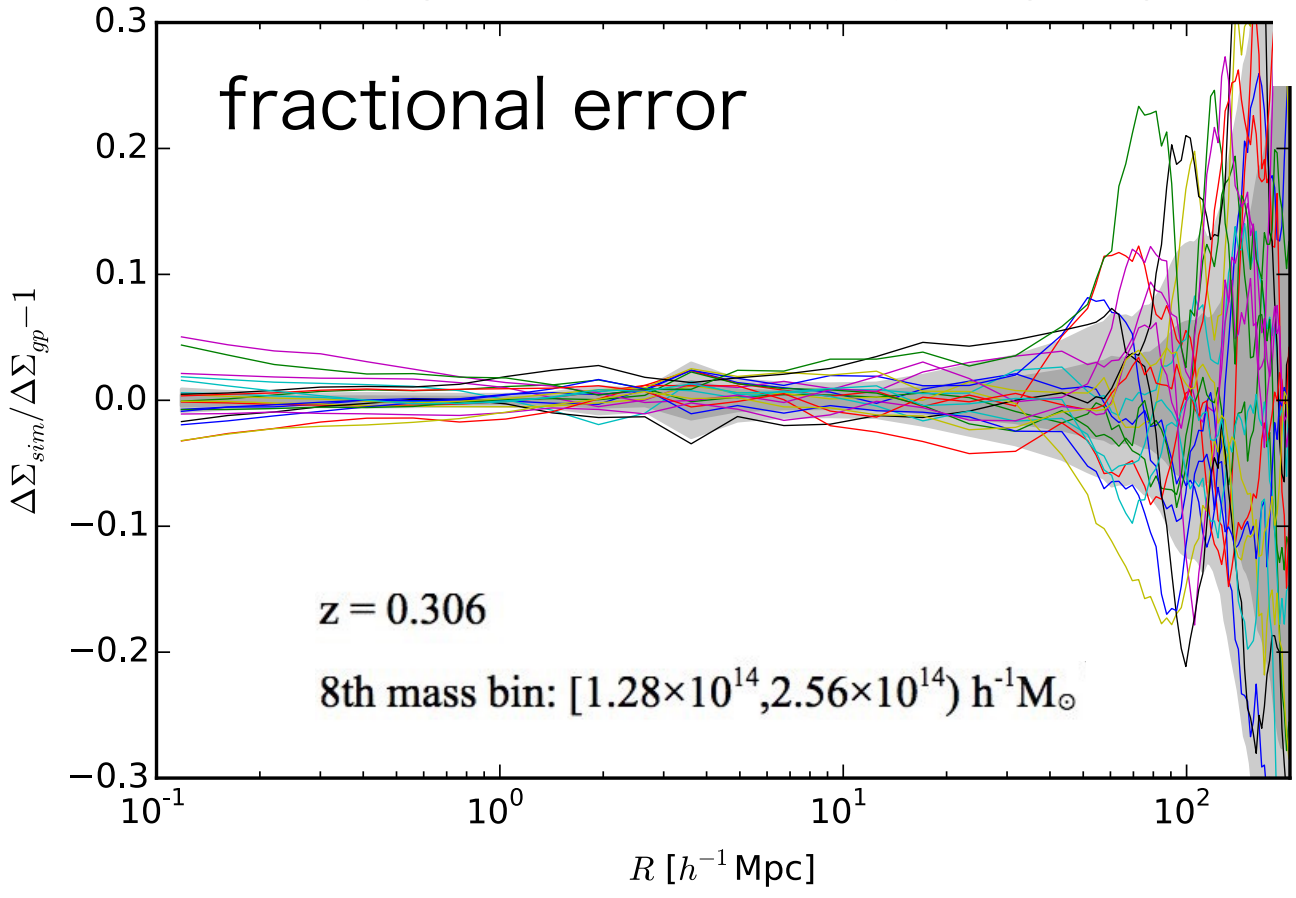
GAUSSIAN PROCESS ACCURACY

training with 20 models (red)

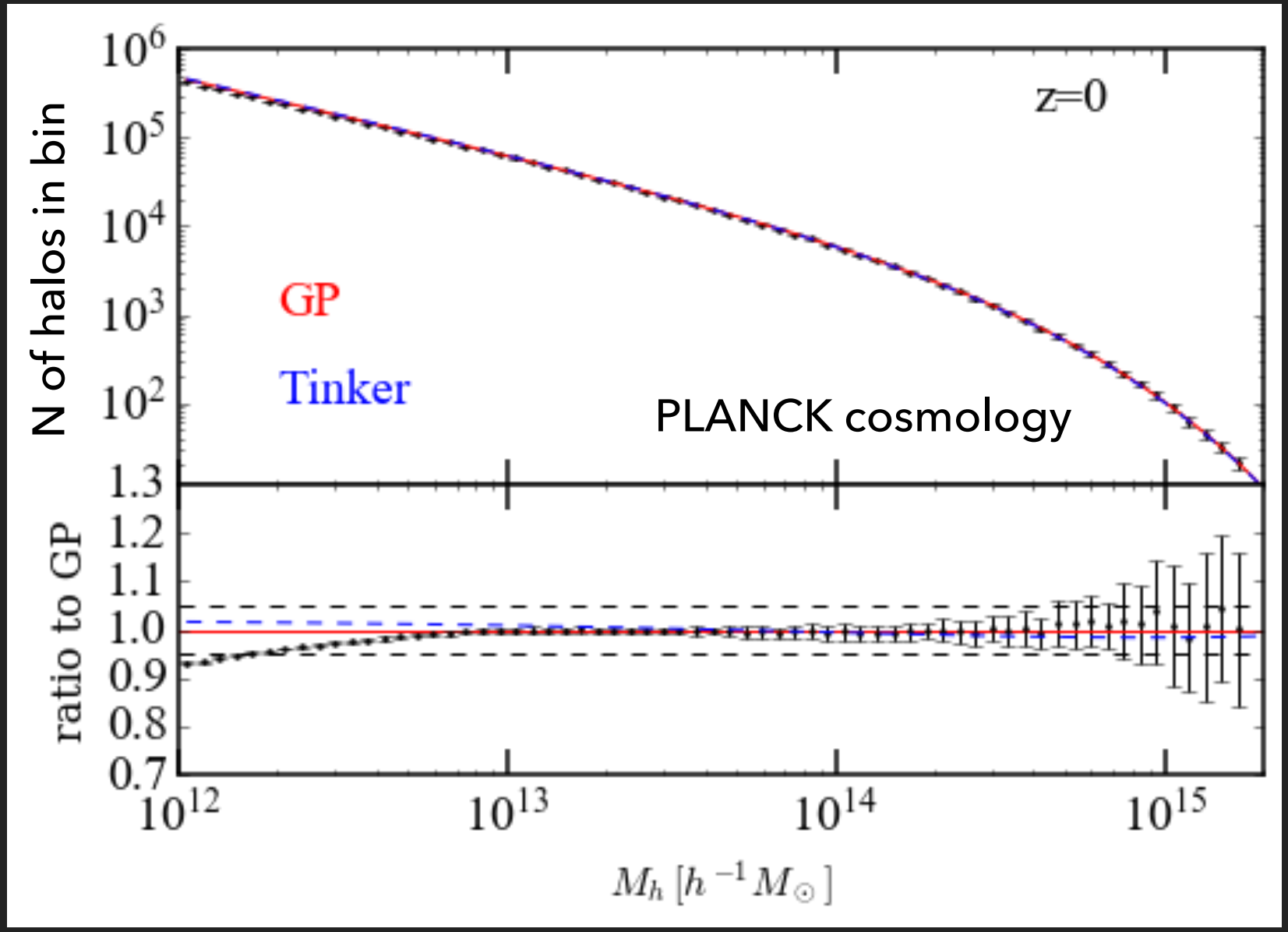
validation with 20 other models (blue)



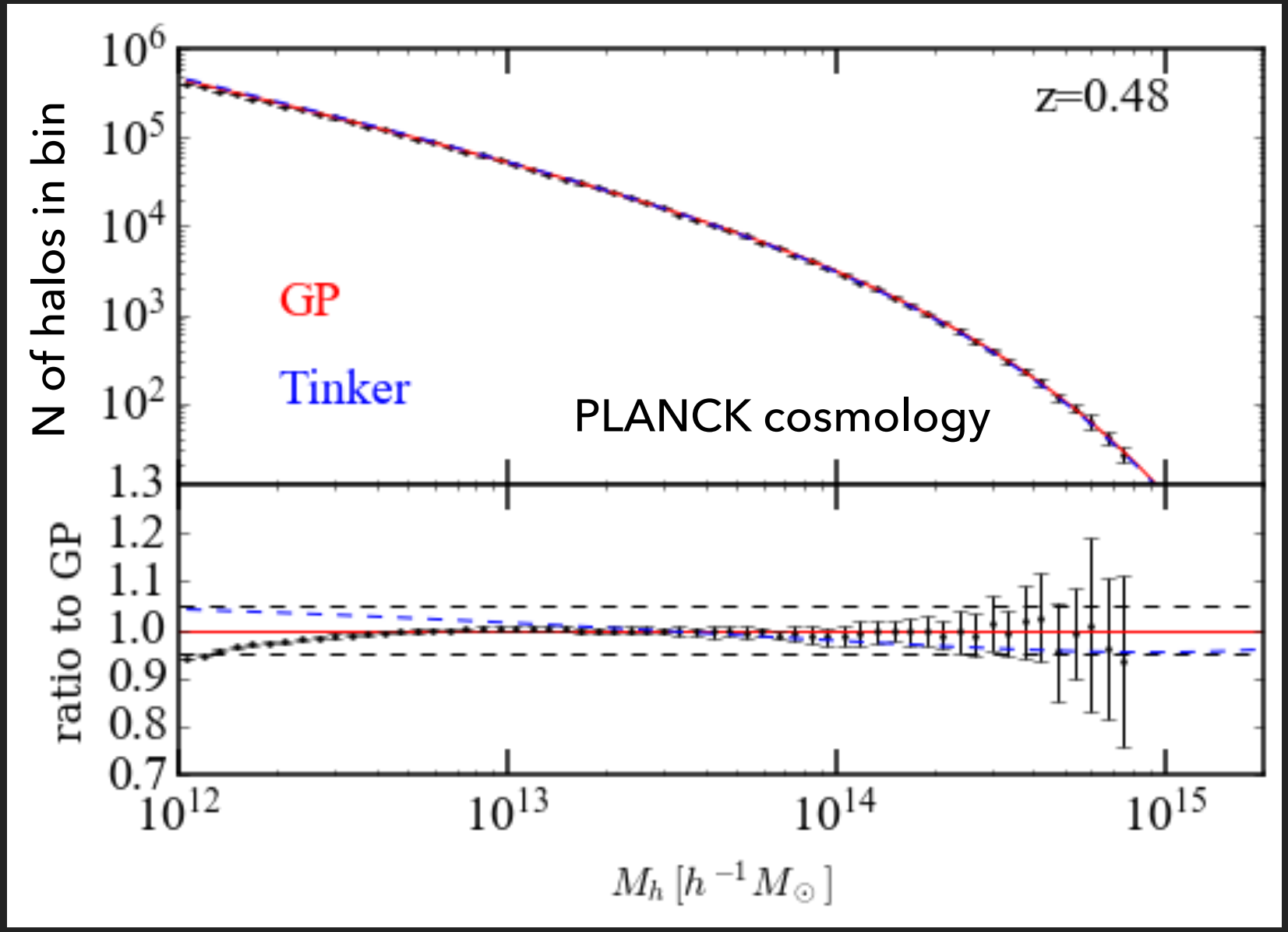
realizations with different random # seeds



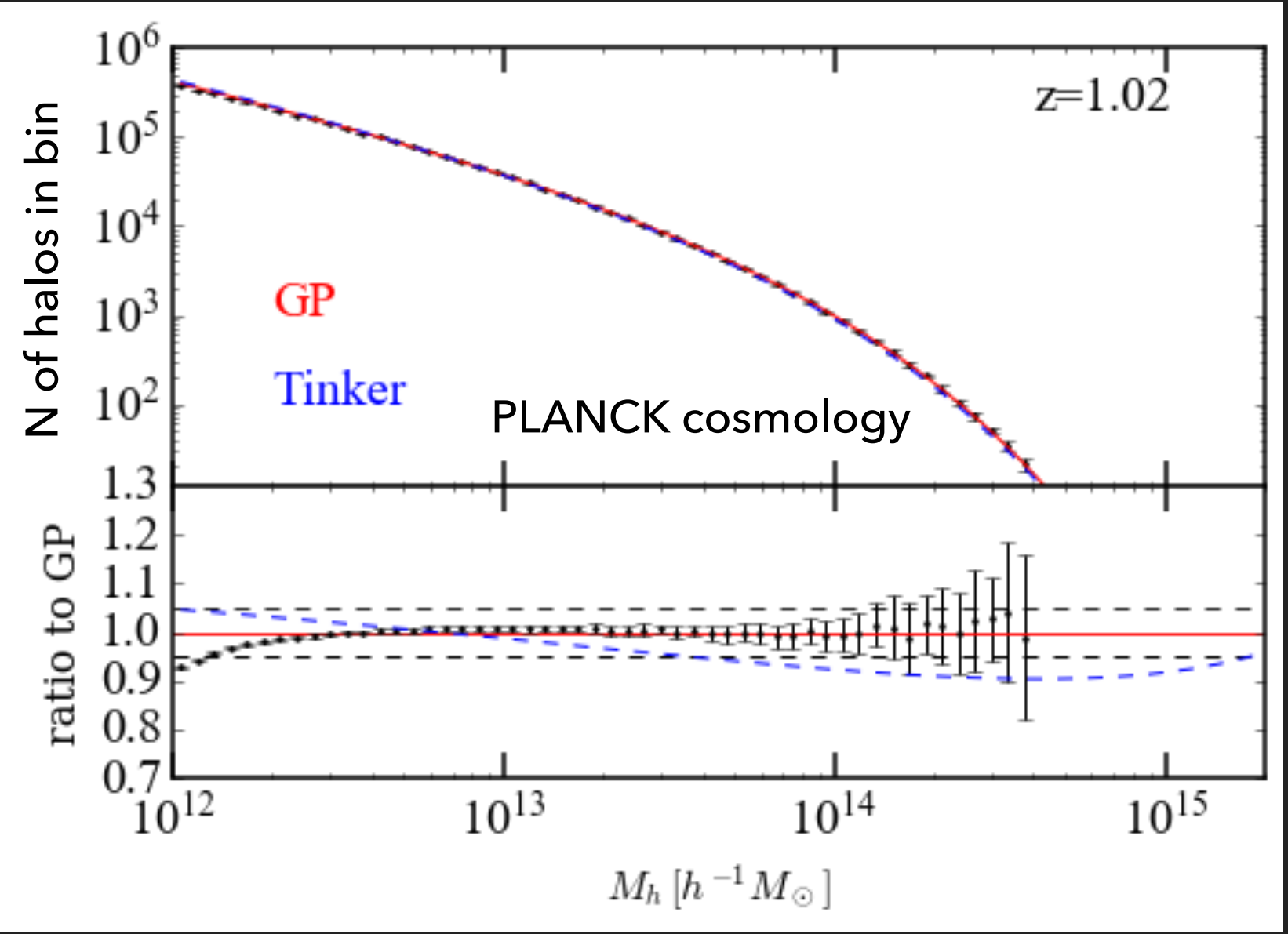
HMF EMULATOR PERFORMANCE



HMF EMULATOR PERFORMANCE



HMF EMULATOR PERFORMANCE



jupyter dsigma_demo Last Checkpoint: 3 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar

Read out pre-computed PCA basis function + GP

```
In [3]: basis = [[joblib.load('basis/S%03d_dens%02d_basis.pkl' % (snap,ndens)) for ndens in range(0,10)] for snap in range(0,21)]
        dsig_gp = [[joblib.load('gp/S%03d_dens%02d_gp.pkl' % (snap,ndens)) for ndens in range(0,10)] for snap in range(0,21)]

In [96]: def prep_dsigma_template(params):
             global d_gp
             d_gp = np.zeros((21,10,81))
             for snap in range(0,21):
                 for ndens in range(0,10):
                     d_gp_pca=dsig_gp[snap][ndens].predict(params)
                     d_gp[snap,ndens,] = basis[snap][ndens].inverse_transform(d_gp_pca)[0,]
```

Prepare a table for 3D spline

Inputs:

- scale factor
- number density (or halo mass)
- projected distance

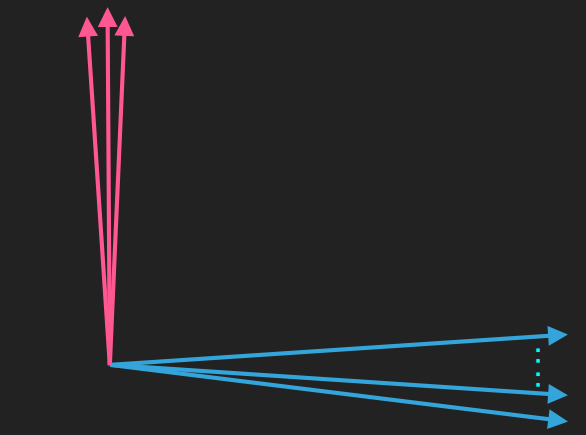
```
In [59]: z=0.25
         ascale = 1./(1+z)
         Rplot = np.logspace(-1,2.3,100) # in h^{-1}Mpc
         Mmin = 1e14 # in h^{-1}M_\odot
```

```
# PLANCK 2015
cparam = np.array([[0.02225,0.1198,0.6844,3.094,0.9645,-1]])
set_cosmo(cparam)        give your cosmological params ~5s
set_redshift(z)          and redshifts ~600ms; HMF GP called inside
lognh = mh_to_logdens(Mmin)    convert M_min to n_h ~50μs
plt.loglog(Rplot,get_dsigma(ascale, lognh, Rplot),lw=2,color='red')
```

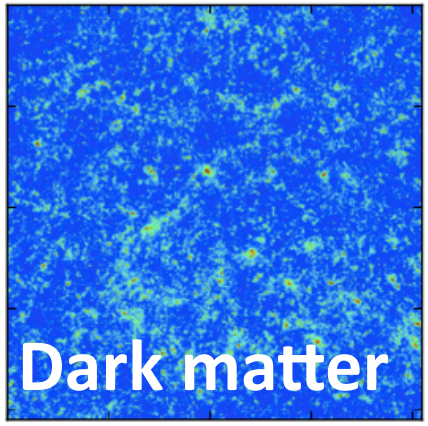
Evaluate !! ~1ms

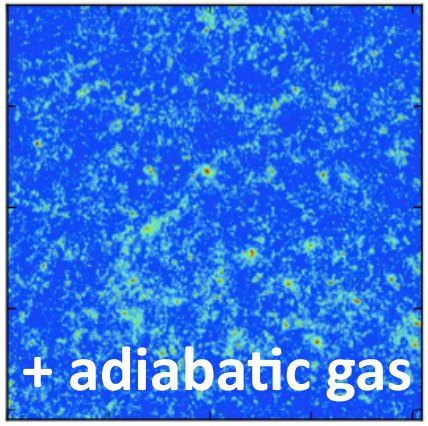
SUMMARY + FUTURE

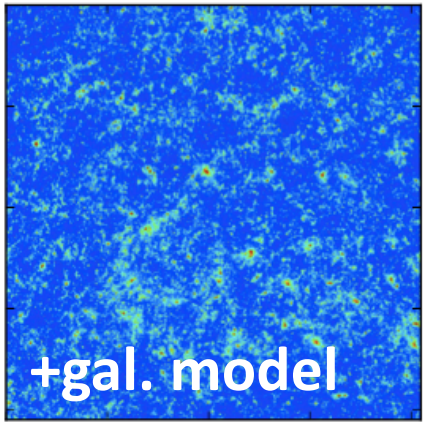
- Modeling the halo mass function and galaxy-galaxy lensing signal
  - Latin hypercube design + fitting/GP/spline
  - handy emulator in python ready
  - accuracy test undergoing, aimed at 5% accuracy
- To come
  - scikit-learn → george
  - RSD emulator to combine g-g lensing and 3D clustering (needs bigger volume)
  - further extension under discussion
    - e.g., non-flat, w0-wa cosmologies

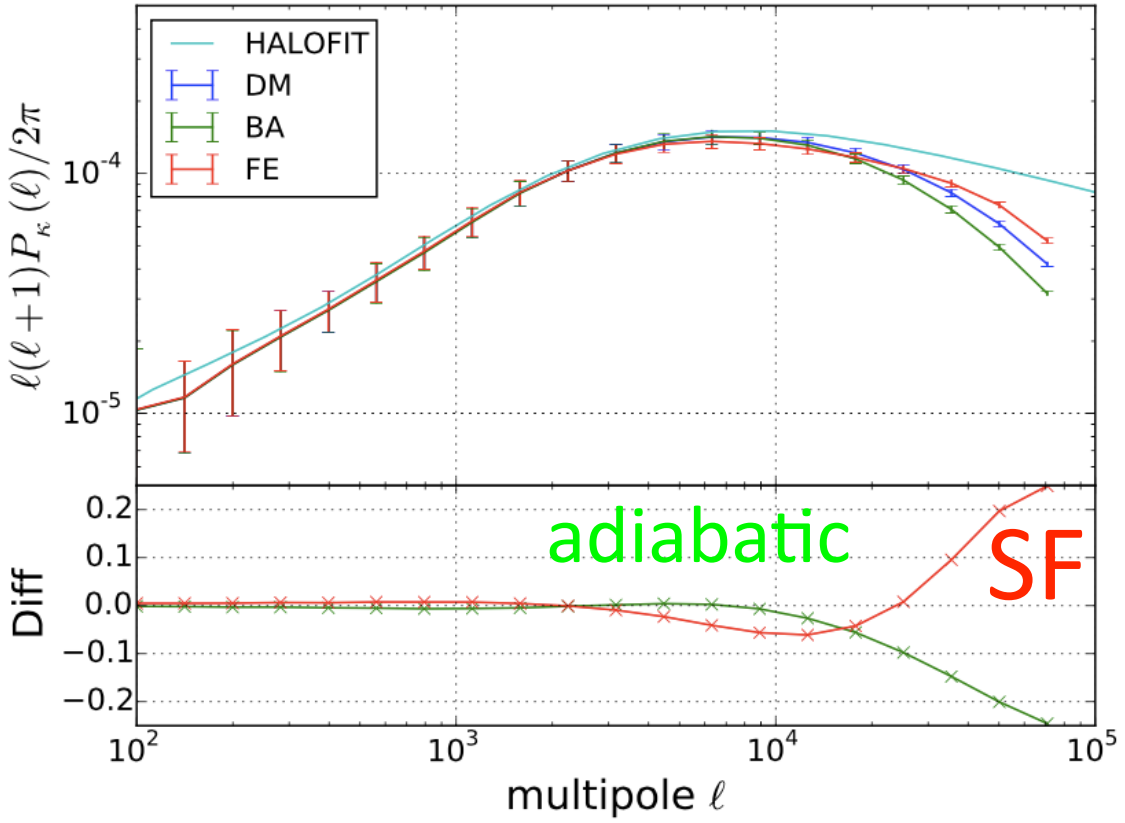


Don't forget, however, the effect of baryons

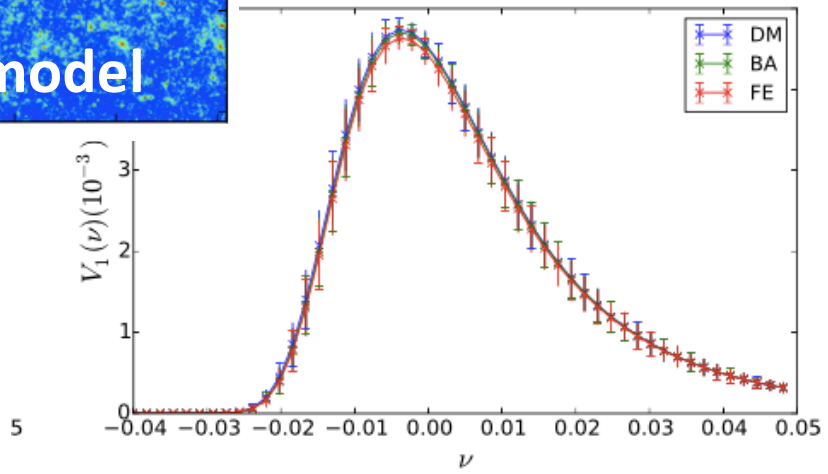


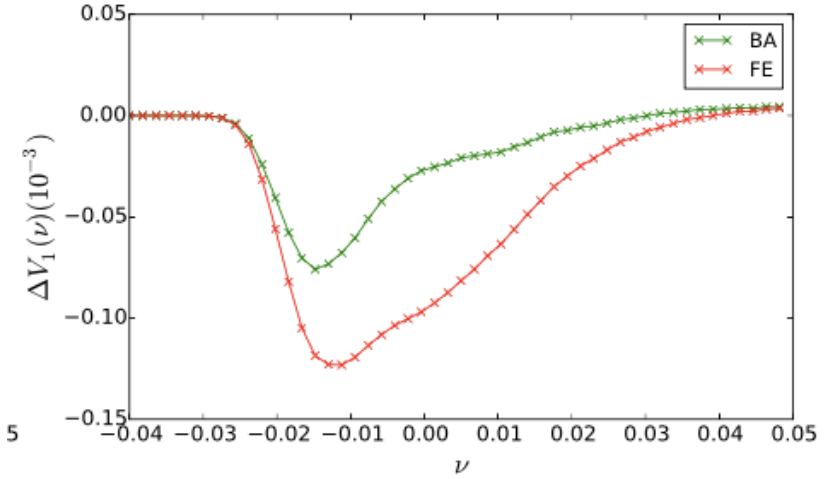






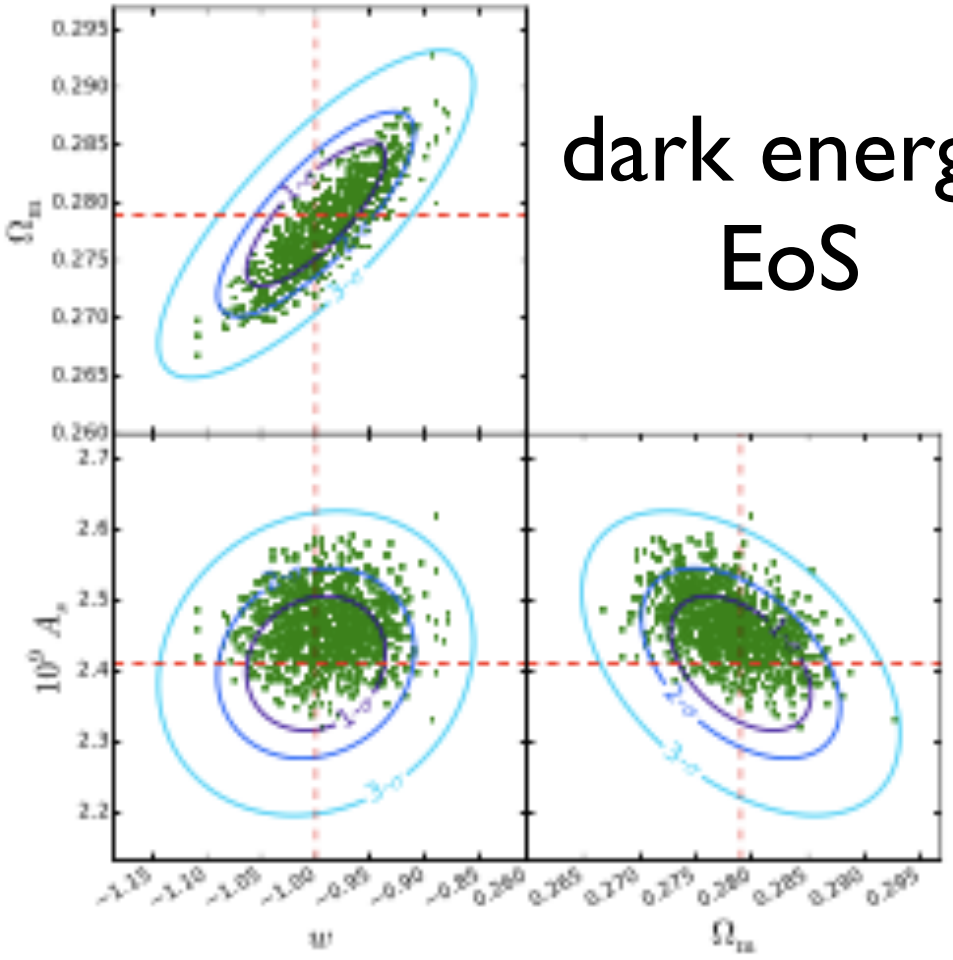
MF V_0



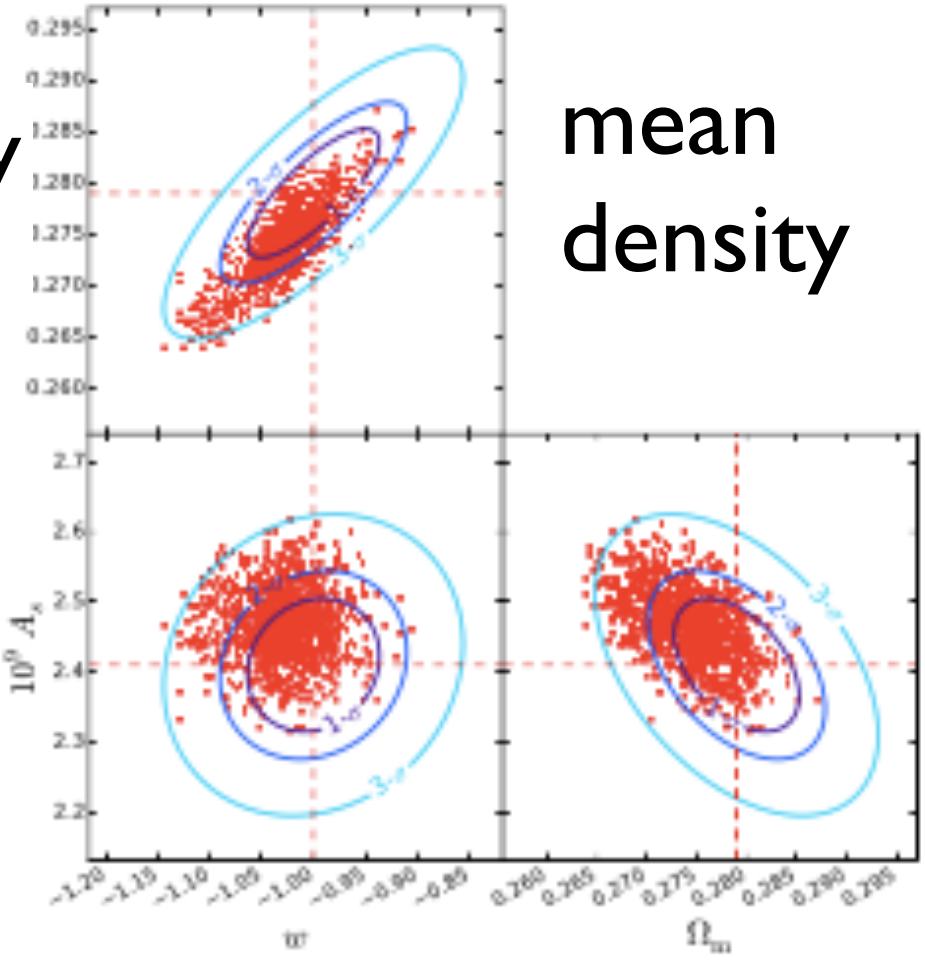


Osato, Shirasaki, NY, 2015

Need for accurate templates



dark energy EoS



mean density

Parameter bias
with respect to naive estimate
from dark matter only simulations

Osato, Shirasaki, NY, 2015