Forward modelling pipeline of tSZ maps

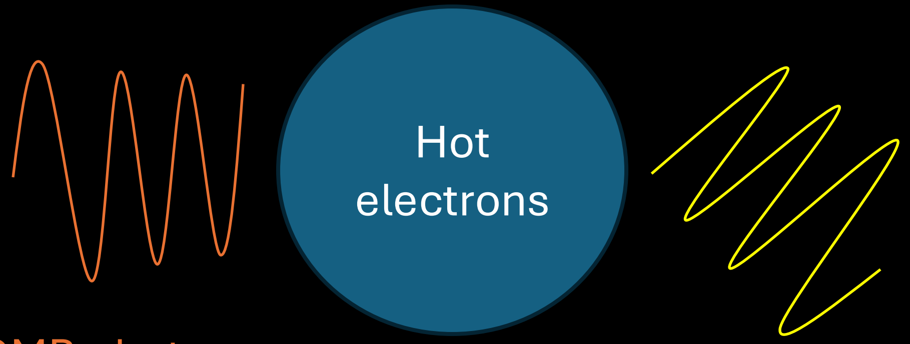
Licong Xu, Íñigo Zubeldia, James Alvey, Boris Bolliet, and Anthony Challinor

Cambridge-LMU meeting, Sep. 18, 2025

The thermal Sunyaev-Zel'dovich effect

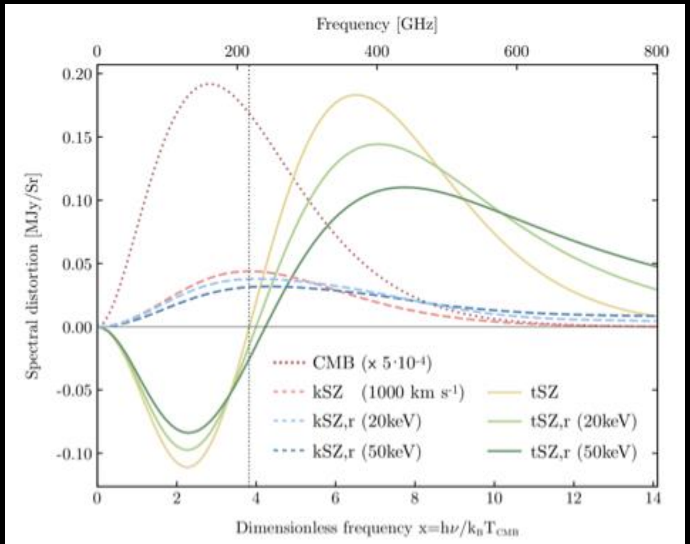
$$\frac{\Delta T_{CMB}}{T_{CMB}} = g(\nu)y \qquad y(M,z,\theta) = \frac{\sigma}{m_e c^2} P_e\left(\sqrt{l^2 + d_A^2\theta^2}\right) dl$$

Spectral distortion of the CMB sensitive to integrated pressure of profiles of clusters along line of sight

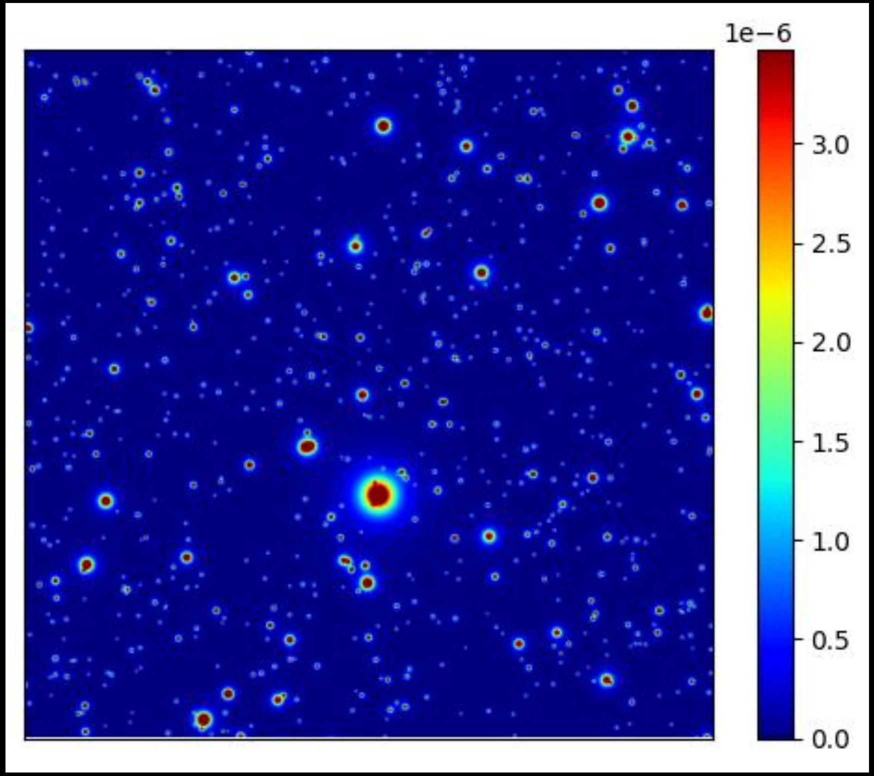


CMB photons

Scattered CMB photons



"Compton-y maps"



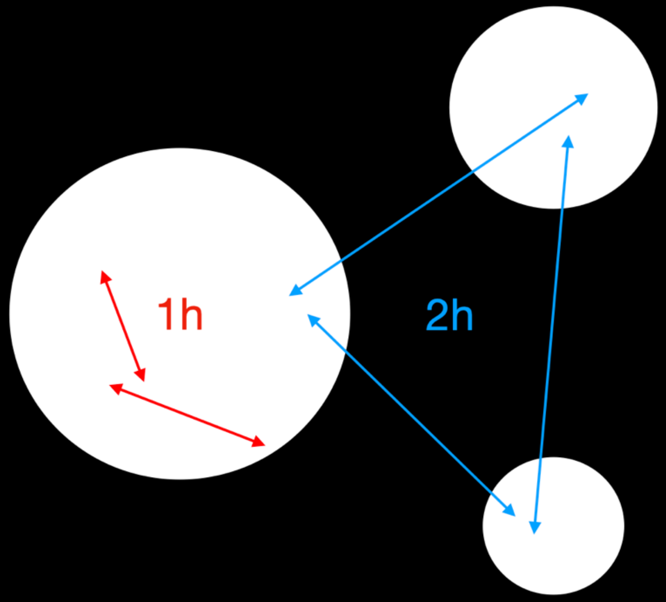
Licong Xu, lx256@cam.ac.uk

The tSZ power spectrum

Halo Model Formalism (Cooray & Sheth, 2002)

$$C_\ell^{yy} = C_\ell^{yy,1h} + C_\ell^{yy,2h}$$

"within each halo" "across halos"



$$C_\ell^{yy,1h} = \int_0^{z_{max}} dz \frac{dV}{dz} \int_{M_{min}}^{M_{max}} dM \frac{dN}{dM\,dV} |y_\ell(M,z)|^2$$

HMF FT of halo pressure profile

"Sum of (FT of) pressure profile of all the halos in the catalogue, averaged over ensemble halo population"

$C_\ell^{yy,2h}$: correlations between spatial positions of halos (clustering)

Note:

- the 2h halo term is not significant if we include massive clusters
- but **might be** non-negligible if we remove massive clusters
- We neglect 2h term in our current analysis

The tSZ power spectrum

$$C_\ell^{yy,1h} = \int_0^{z_{max}} dz \frac{dV}{dz} \int_{M_{min}}^{M_{max}} dM \frac{dN}{dMdV} |y_\ell(M,z)|^2$$

Cosmology (Ω_m, σ_8) at large scale

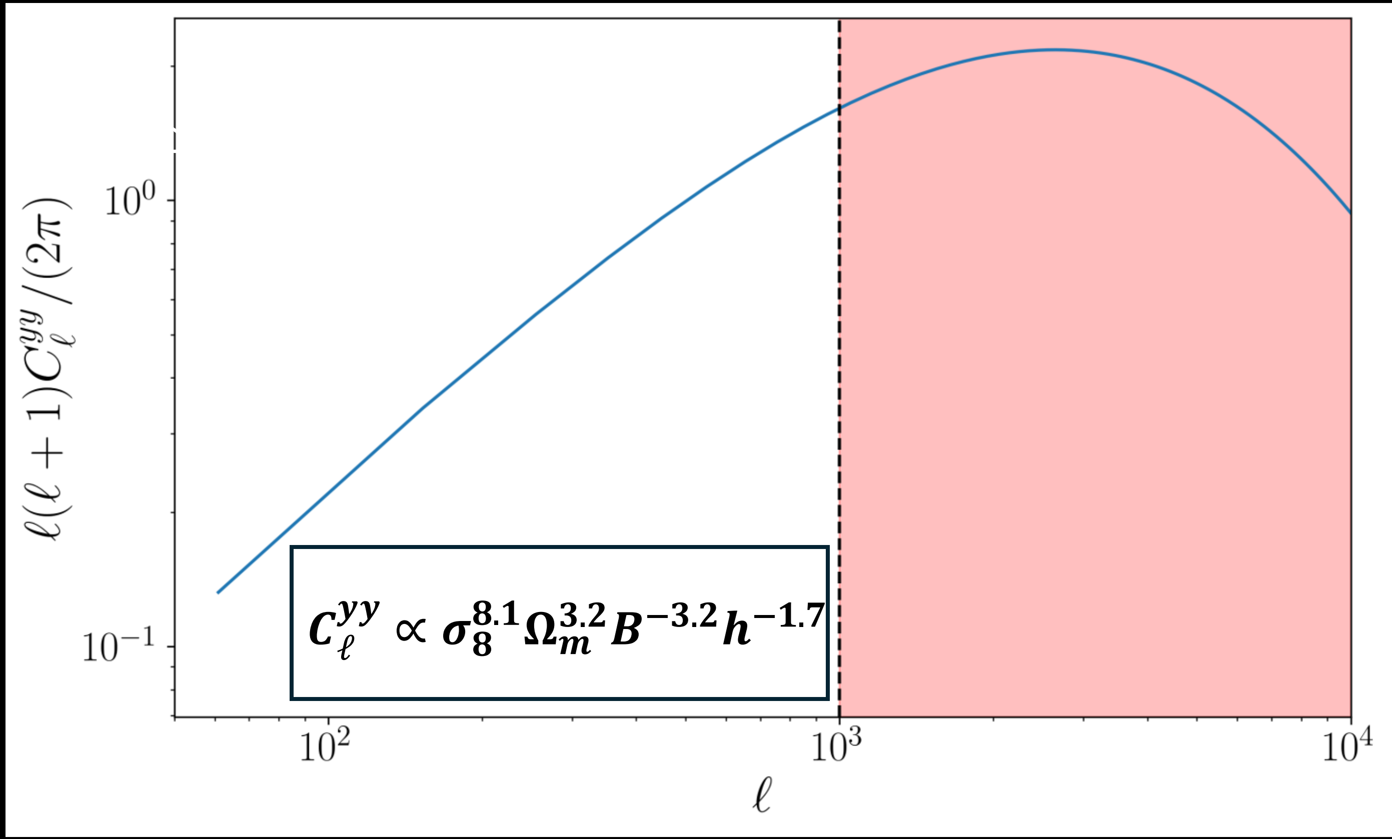
Gas physics at small scale

Assume Generalized NFW pressure profile:

$$P_e(x) = C \times P_0 (c_{500} x)^{-\gamma} [1 + (c_{500} x)^\alpha]^{(\gamma - \beta)/\alpha}$$

$$x = r/r_{500}$$

Use Arnauld 2010 parameters:
$\{\gamma, \alpha, \beta, P_0, c_{500}\}$ are constants



Likelihood analysis

- Clusters follow **Poisson** Statistics
- But people assume a **Gaussian** likelihood

$$\chi^2 = (\hat{\boldsymbol{C}} - \boldsymbol{C})^T \boldsymbol{M}^{-1} (\hat{\boldsymbol{C}} - \boldsymbol{C})$$

$$C_\ell^{yy,tot} = C_\ell^{tSZ,1h} + A_{CIB} C_\ell^{CIB} + A_{IR} C_\ell^{IR} + A_{RS} C_\ell^{RS} + A_{CN} C_\ell^{CN}$$

Signal FG Residual

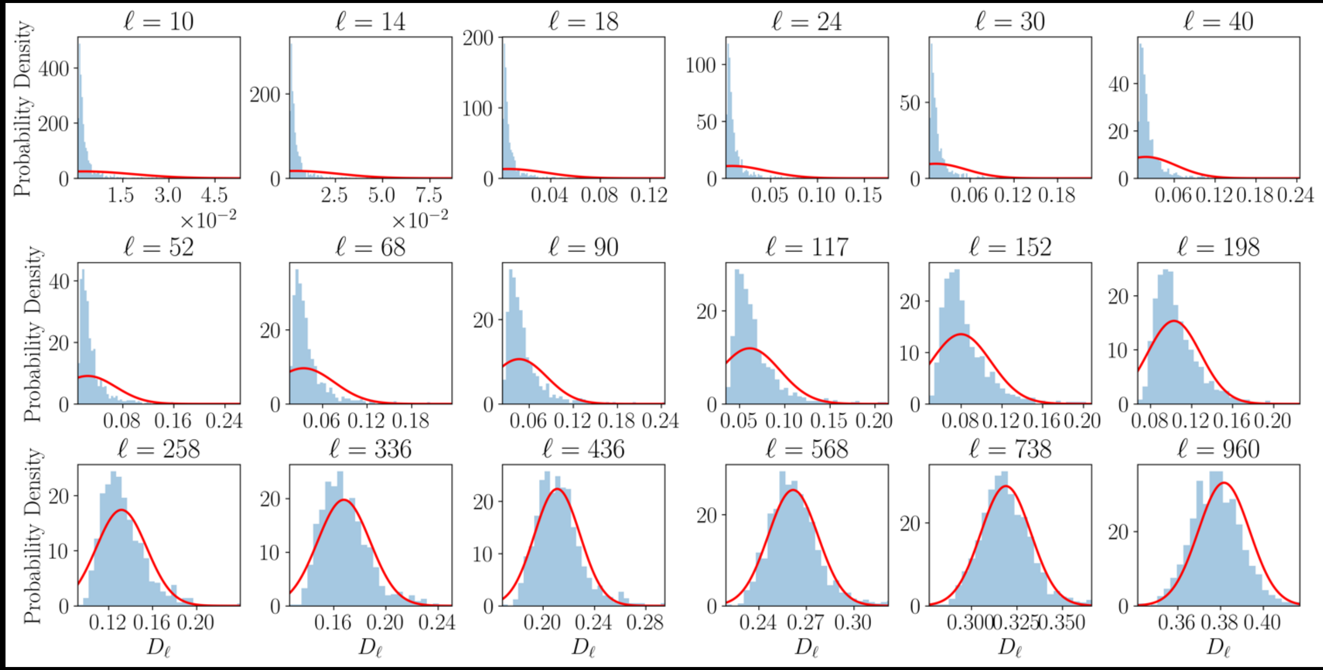
- Include the **trispectrum** term in the covariance matrix

$$T_{\ell\ell'} = \int_0^{z_{max}} dz \frac{dV}{dz d\Omega} \int_{M_{min}}^{M_{max}} dM \frac{dn(M,z)}{dM} |y_\ell(M,z)|^2 |y_{\ell'}(M,z)|^2 \longrightarrow$$ "Covariance of the power spectrum"

- Sensitive to four posterior-driven parameters:
  - $F = \sigma_8 \left(\frac{\Omega_m}{B}\right)^{0.40} h^{-0.21}, A_{CIB}, A_{IR}, A_{RS}$

Why we do SBI?

- tSZ is non-Gaussian and analytical likelihood difficult to write, SBI does not need likelihood



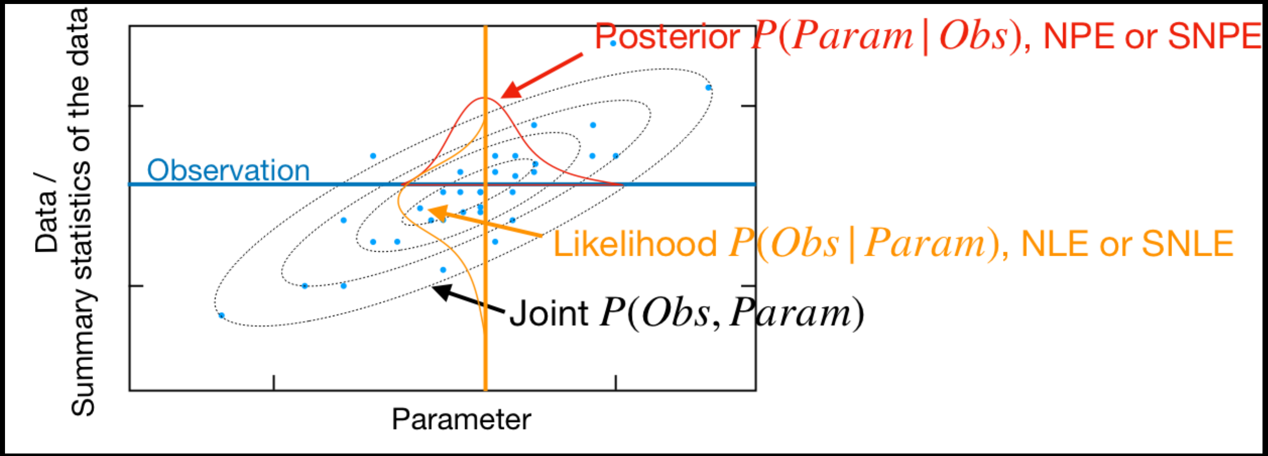
- No need to model covariance if you have a good forward model

SBI – A brief introduction

SBI is

- Produce lots of simulations with parameter-data pairs: $\{\theta_i, d_i\}$
- Density estimation task, three ways: $p(\theta|d),\ p(d|\theta),\ p(\theta, d)$

Comment: this is the same as Generative AI



Many ways to do that:

- Simply estimate the KDE
- Neural networks

Examples of neural networks:

- Simple Gaussian Mixture Model (GMM):

$$p(\boldsymbol{d}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{\theta}; \boldsymbol{w}) N[(\boldsymbol{d}|\mu_k(\theta; w), \boldsymbol{C_k}(\boldsymbol{\theta}, \boldsymbol{w})]$$

- **Autoregressive models:**

$$p(\boldsymbol{d}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(d_i|d_1, d_2, \ldots, d_{i-1}; \theta) = \prod_{i=1}^{n} p(d_i|\boldsymbol{d}_{<i}; \boldsymbol{\theta})$$

→ **Masked Autoencoder Density Estimation (MADE)**
Masked Autoregressive Flow (MAF)

SBI – NPE & NLE

Given data $\boldsymbol{x}$, compress the data into some summary statistics $\boldsymbol{t}$

Neural Posterior Estimation (NPE)

- Learn the posterior $p(\boldsymbol{\theta}|\boldsymbol{t})$
- So the learned posterior should be similar to true posterior $p^*(\boldsymbol{\theta}|\boldsymbol{t})$
- So weights $\boldsymbol{w}$ of Neural Net should be tuned to minimize

$$\mathbb{E}_{\boldsymbol{p}(\boldsymbol{x})}[\mathbb{KL}((\boldsymbol{p}^*(\boldsymbol{\theta}|\boldsymbol{t})||\boldsymbol{p}(\boldsymbol{\theta}|\boldsymbol{t};\boldsymbol{w}))]$$

Avg. over data Similarity

$$Loss = -\frac{1}{N_{samples}}\sum_{i=1}^{N_{samples}} \ln p(\boldsymbol{\theta}_i|\boldsymbol{t}_i, \boldsymbol{w})$$

Neural Likelihood Estimation (NLE)

- Learn the likelihood $p(\boldsymbol{t}|\boldsymbol{\theta})$
- So the learned likelihood should be similar to true likelihood $p^*(\boldsymbol{t}|\boldsymbol{\theta})$
- So weights $\boldsymbol{w}$ of Neural Net should be tuned to minimize

$$\mathbb{E}_{\boldsymbol{p}(\boldsymbol{\theta})}\left[\mathbb{KL}\left(\left(\boldsymbol{p}^*(\boldsymbol{t}|\boldsymbol{\theta})\right|\left|\boldsymbol{p}(\boldsymbol{t}|\boldsymbol{\theta};\boldsymbol{w})\right)\right]$$

Avg. over params Similarity

$$Loss = -\frac{1}{N_{samples}}\sum_{i=1}^{N_{samples}} \ln p(\boldsymbol{t}_i|\boldsymbol{\theta}_i, \boldsymbol{w})$$

SBI – Likelihood based

~~Given data $\boldsymbol{x}$~~, compress the data into some summary statistics ~~$\boldsymbol{t}$~~ C_ℓ^{yy}


tszpower: Cosmological Parameter Inference with tSZ Power Spectrum

Python 3.8+ | JAX | License Apache 2.0

A repository for computing the thermal Sunyaev-Zel'dovich (tSZ) power spectrum and performing cosmological parameter inference using both **Markov Chain Monte Carlo (MCMC)** and **Simulation-Based Inference (SBI)**. Built with JAX for accelerated computation and automatic differentiation.

$$\chi^2 = \left(\hat{\boldsymbol{C}} - \boldsymbol{C}\right)^T \boldsymbol{M}^{-1} \left(\hat{\boldsymbol{C}} - \boldsymbol{C}\right)$$ ➡ "Data are Gaussian with Covariance $\boldsymbol{M}$"

$$C_\ell^{yy} = C_\ell^{yy,signal} + N_\ell^{yy}$$

Theoretical tSZ PS given cosmo params

$\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$ (Cholesky decomposition)

$\boldsymbol{N} = \boldsymbol{L}\boldsymbol{v}$ (transform from standard Gaussian)

Likelihood benchmark:

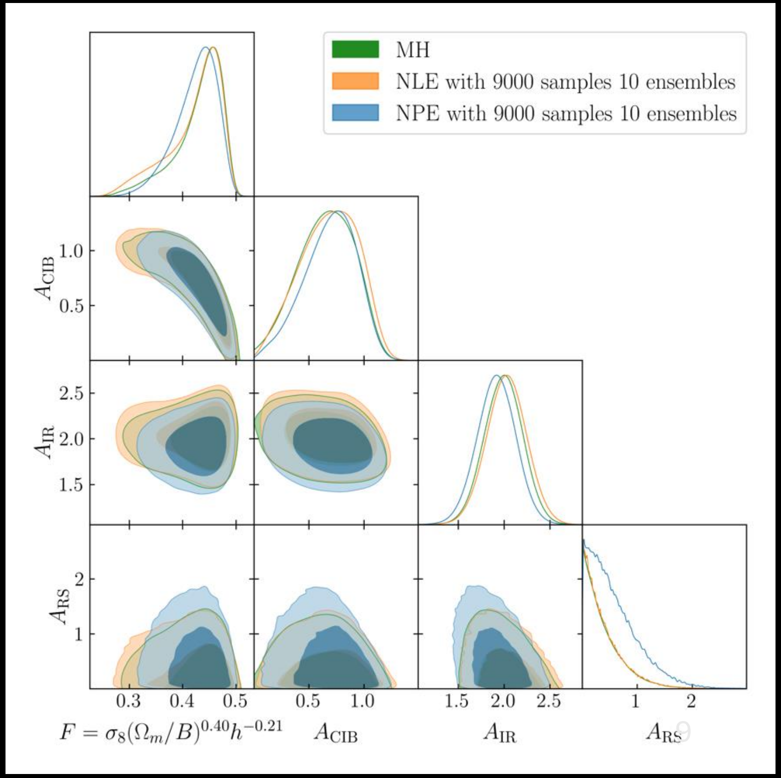
- Cobaya with Metropolis-Hastings algorithm
- $z_{min} = 0.005, z_{max} = 3.0$
- $M_{min} = 10^{10} M_\odot h^{-1}$, $M_{max} = 3.5 \times 10^{15} M_\odot h^{-1}$

SBI details:

- 9000 parameter-data pairs
- Train:Validation split = 80:20
- Averaging over ensemble of 10 networks: reduce intrinsic bias of NN

Conclusion:

- NLE: all within 0.2σ
- NPE: worst within 0.33σ (A_{IR})



Licong Xu, lx256@cam.ac.uk

SBI – Halo based

"C_ℓ^{yy} non-Gaussian at low ℓ so Gaussian likelihood is not correct"

tSZ maps $\quad C_\ell^{yy}$

Given data $\boldsymbol{x}$, compress the data into some summary statistics $\boldsymbol{t}$

1. Generating cluster catalogues

cosmocnc

cosmocnc is a Python package for evaluating the number count likelihood of galaxy cluster catalogues in a fast, flexible and accurate way. It is based on the use of Fast Fourier Transform (FFT) convolutions in order to evaluate some of the likelihood integrals. The code was introduced in Zubeldia & Bolliet (2024), where the likelihood formalism and implementation are described in detail. If you use the code, please cite the paper.

- Sample N clusters from the HMF
- Assign random sky coordinate (no clustering)
- For each cluster in the catalogue:
  - (z, M, lon, lat)

2. Painting clusters onto sky

XGPaint.jl

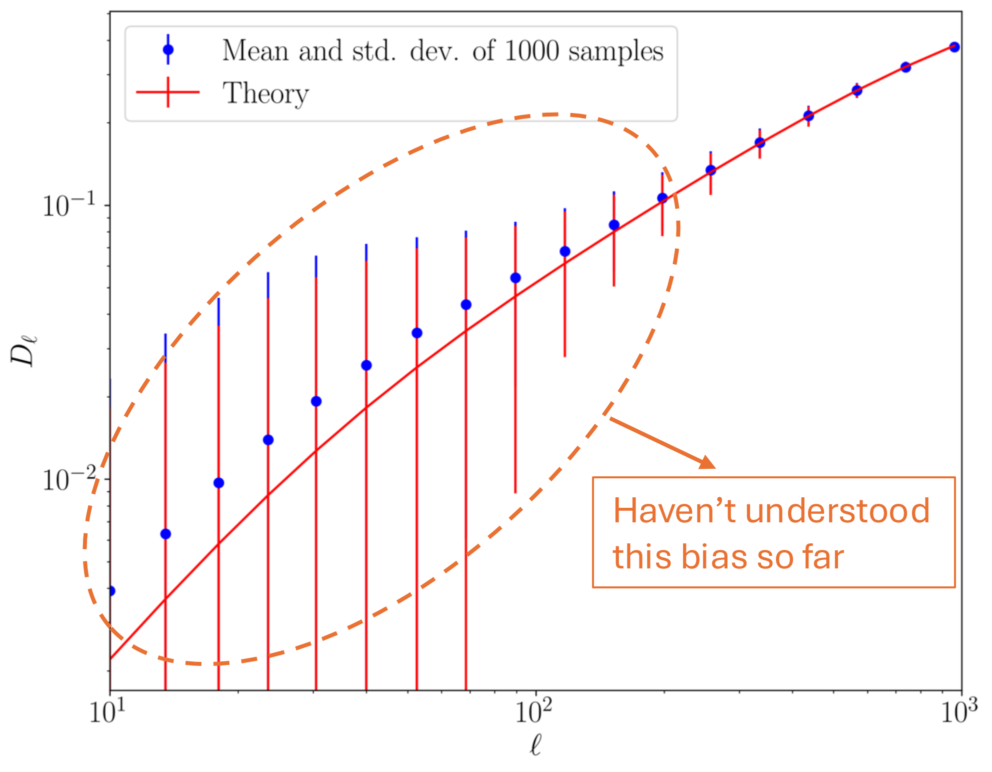
docs dev | CI passing | codecov 23%

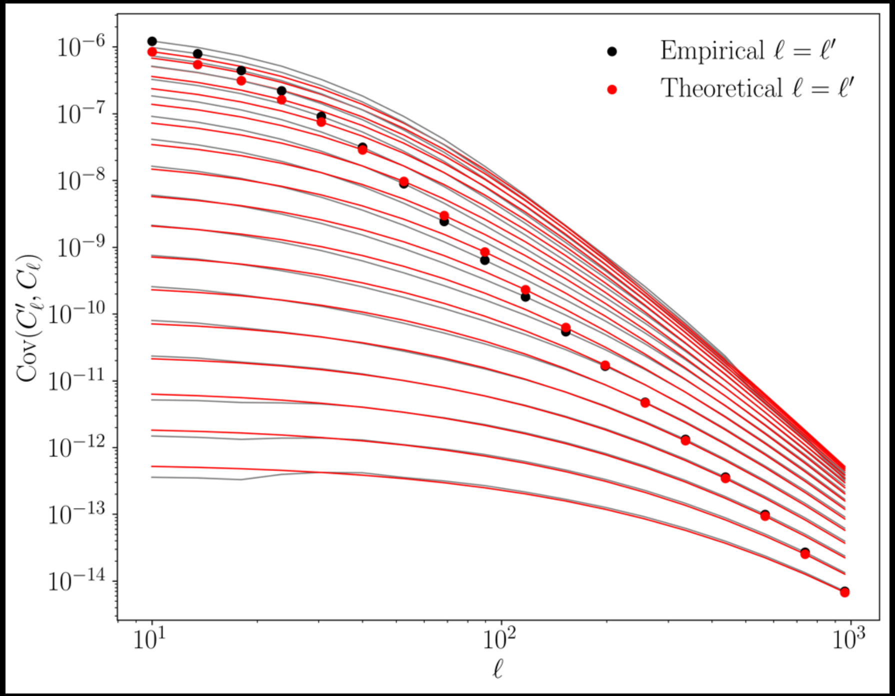
`XGPaint.jl` paints maps of extragalactic foregrounds using halo catalogs. Please read the documentation.

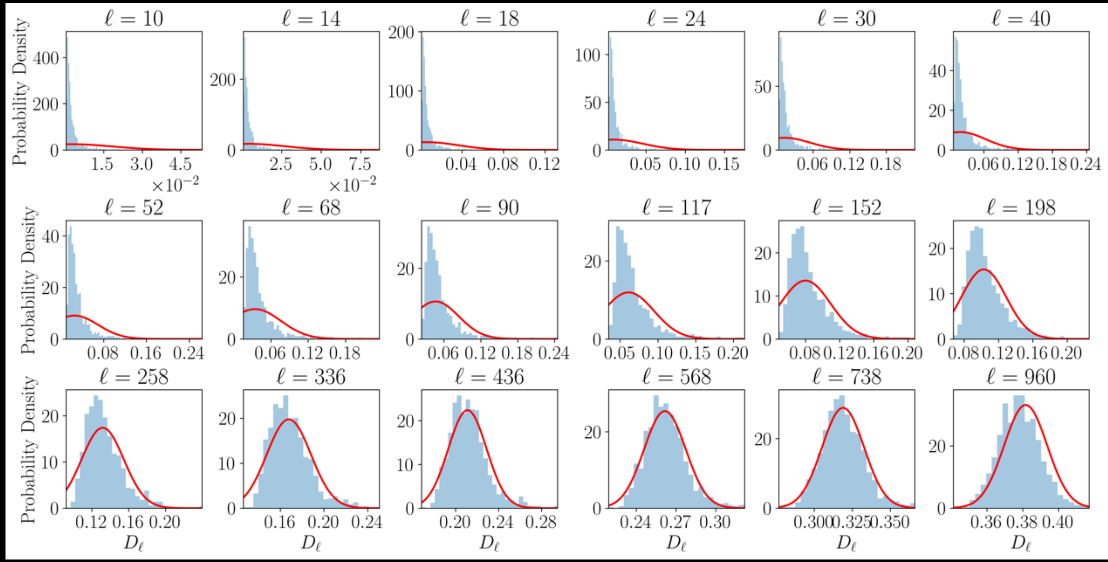
- Assign beam convolved (FWHM=10') pressure profile for each cluster
- Paint clusters by sky positions
- Healpix workspace with $N_{side} = 1024$

SBI – Halo based

- 1000 full sky maps, fixed cosmology
- $z_{min} = 0.005, z_{max} = 3.0$
- $M_{min} = 10^{14} M_{\odot} h^{-1}$, $M_{max} = 10^{16} M_{\odot} h^{-1}$







SBI – Halo based

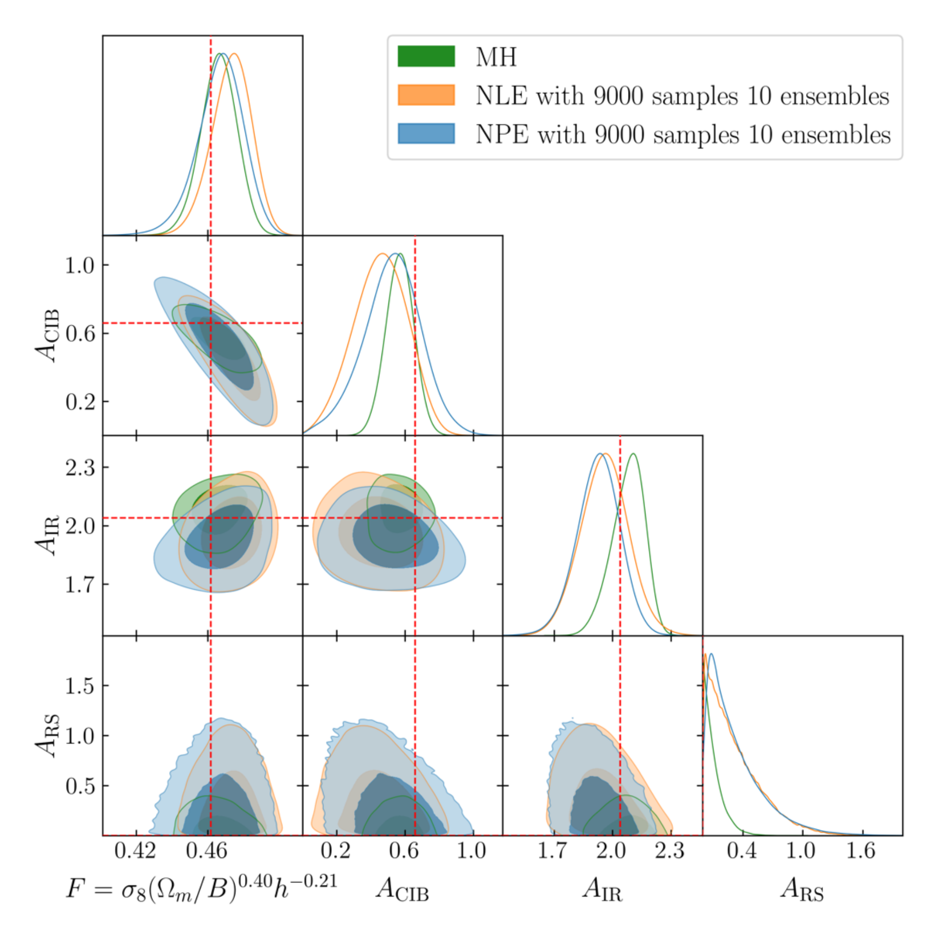
Simulation details:

- Painting halos from halo model
- FG residuals painted as Gaussian random field (GRF) from template FG power spectrum
- No instrumental noise

SBI details:

- 9000 **full-sky maps**
- $z_{min} = 0.005, z_{max} = 3.0$
- $M_{min} = 10^{14} M_{\odot} h^{-1}$, $M_{max} = 10^{16} M_{\odot} h^{-1}$
  - C_ℓ^{yy} at low ℓ saturate if we decrease M_{min} further
  - Fewer clusters, so save computational time

- Take "true data" to be one realization of mock data



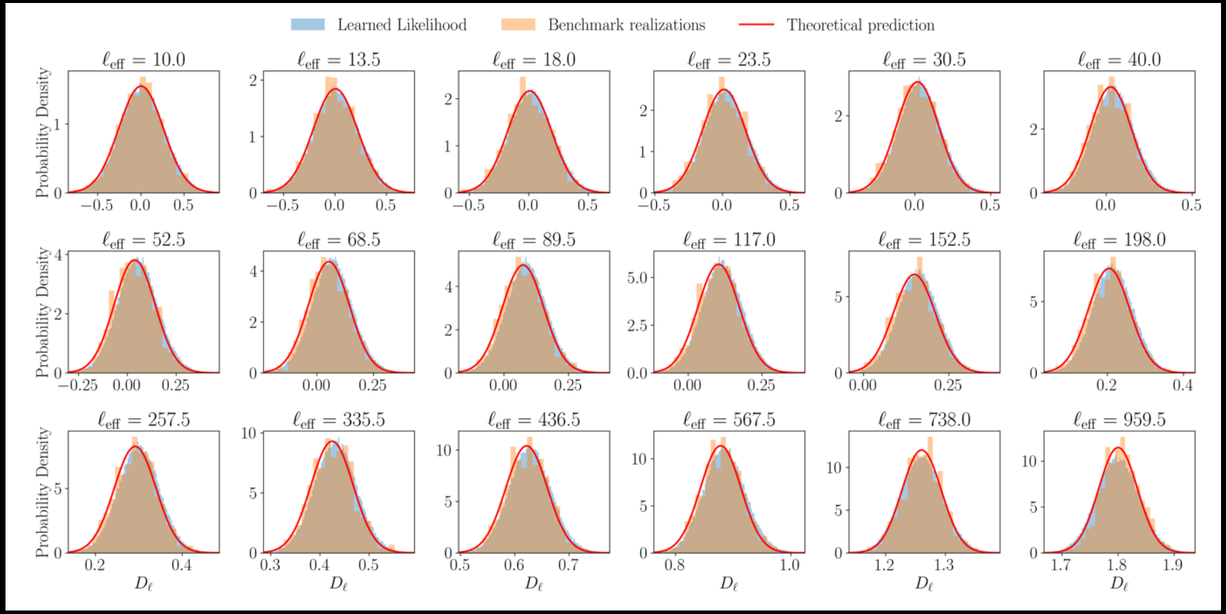
Conclusion:

- SBI and likelihood agrees on F (cosmology)
- SBI predicts wider posteriors for foreground parameters ⟶ Haven't come up with an explanation yet

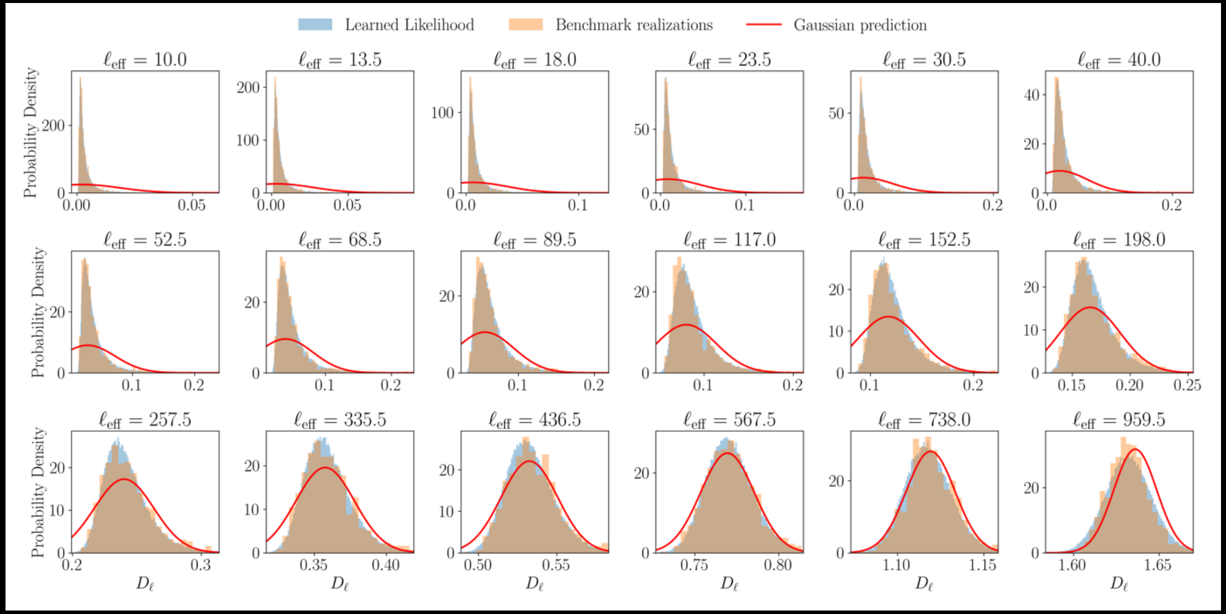
SBI – NLE Validation

- NLE learns the likelihood
- Can look at the learned likelihood at some input parameters (since SBI = generative AI)
- Meanwhile, we can generate many realizations of tSZ maps at those input parameters
- If SBI is learning correctly, these two should agree

1. Gaussian likelihood SBI



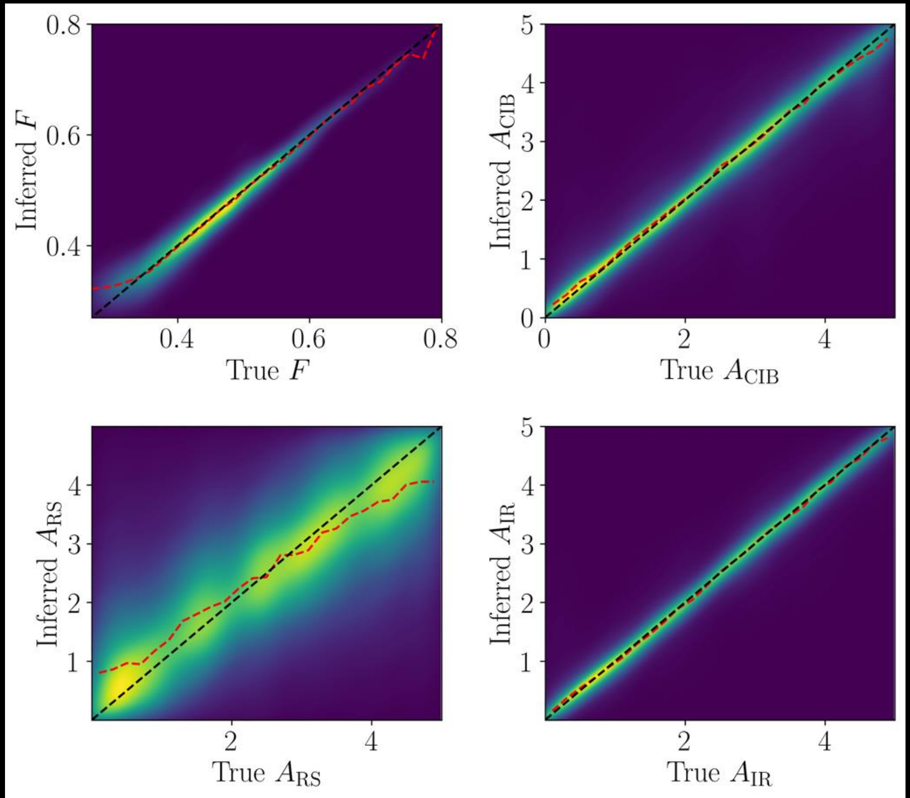
2. Halo based SBI from painting



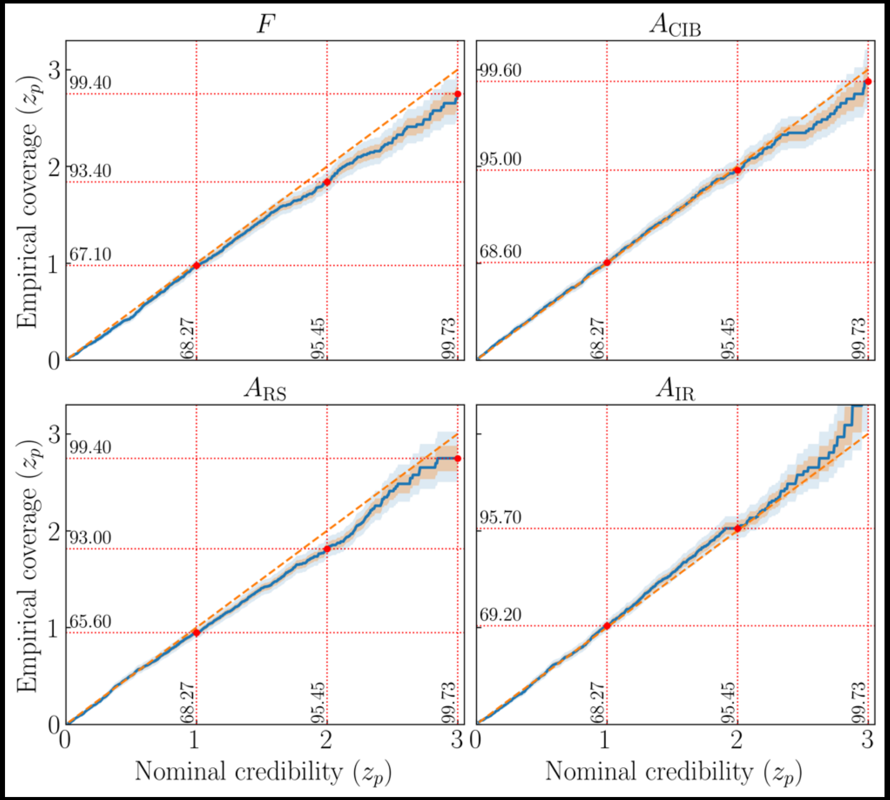
SBI – NPE Validation

- To test SBI is unbiased across parameter space of interest
- SBI is “amortized” (since SBI = generative AI), only need to train once
- Sampling posteriors at different mock observational data points is fast

Coverage plot



Probability-Probability (P-P) plot



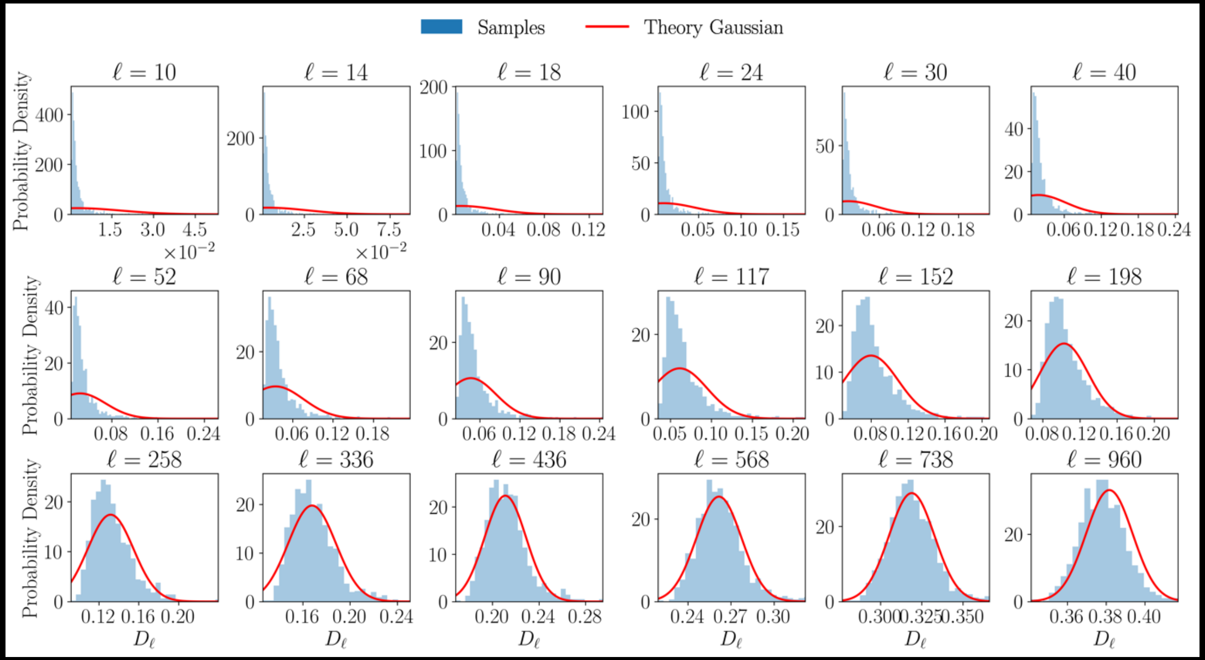
Is PS really the optimal thing to look at?

tSZ maps

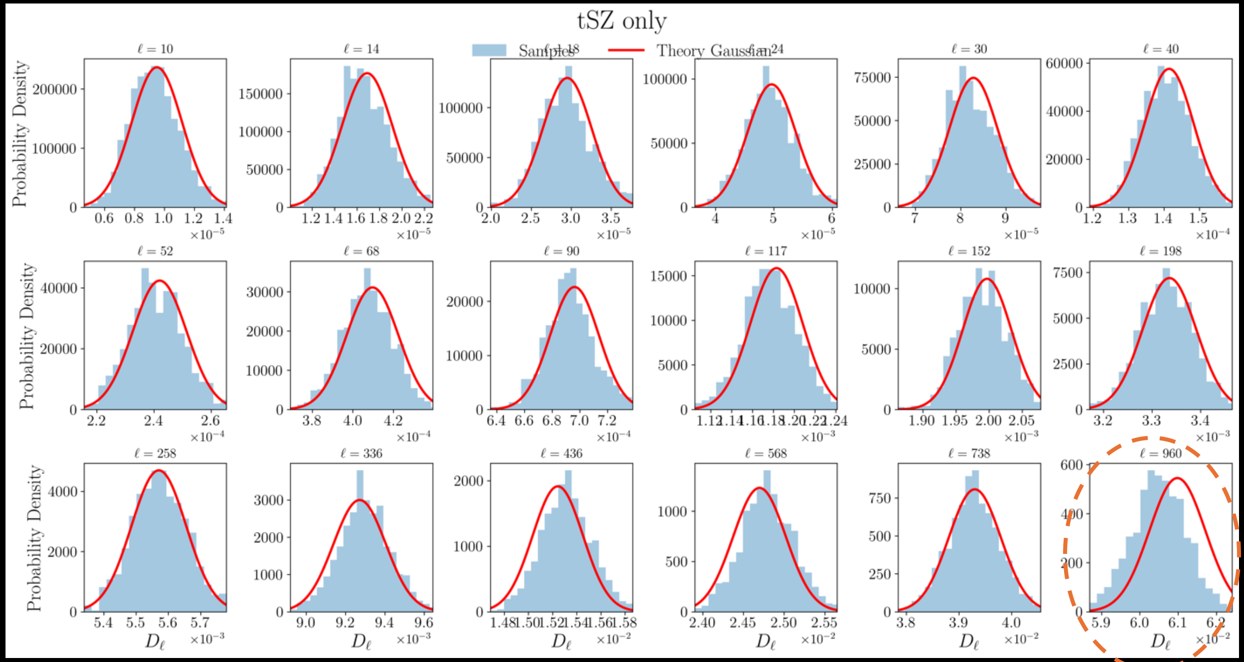
???

Given data x, compress the data into some summary statistics t

$z_{min} = 0.005, z_{max} = 3.0$
$M_{min} = 10^{14} M_{\odot} h^{-1}$, $M_{max} = 10^{16} M_{\odot} h^{-1}$



$z_{min} = 0.5, z_{max} = 3.0$
$M_{min} = 10^{14} M_{\odot} h^{-1}$, $M_{max} = 10^{15} M_{\odot} h^{-1}$



"Removing local clusters makes the low ℓ more Gaussian"

The masked part are resolved clusters!

→ "cluster number count"

To be fixed

Combine with cluster number count (coming soon …)

tSZ maps $\{C_\ell^{yy,masked}, N_{clusters}\}$

Given data ~~x~~, compress the data into some summary statistics ~~t~~

$N(M, z) \rightarrow N(q, z)$ Converting mass M to observables q (e.g., in tSZ, $q = SNR = \frac{y_0}{\sigma_{y_0}}$)

$$\frac{dN}{dqdz} \propto p(q, z) = \int dM\, p(M, z) p(q|M, z)$$

HMF

Scaling relations

$$N \propto \int_{M,z} dM dz\, p(M, z) \int_{q_{min}}^{\infty} dq\, p(q|M, z)$$

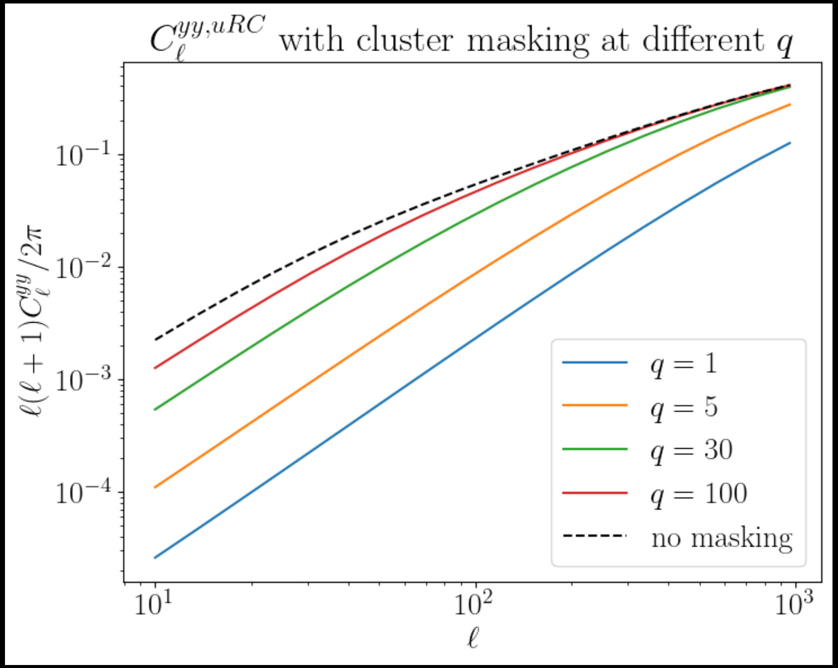
$\hat{\chi}$, completeness

Combine with cluster number count (coming soon …)

tSZ maps $\{C_\ell^{yy,masked}, N_{clusters}\}$

Given data x, compress the data into some summary statistics t

$$C_\ell^{yy,1h} = \begin{cases} \langle |y_\ell|^2 \;\; \hat{\chi} \rangle & \text{Resolved} \rightarrow N_{clusters} \\ \langle |y_\ell|^2 \;\; (1-\hat{\chi}) \rangle & \text{Unresolved} \end{cases}$$

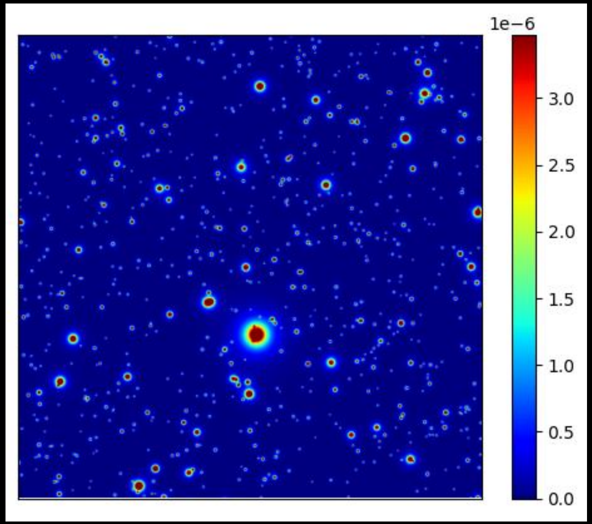


Note in SBI:

- No need to model covariance between $C_\ell^{yy,masked}$ & $N_{clusters}$
  - This covariance is not understood, can use SBI as test
- Can easily extend to full joint analysis of $C_\ell^{yy,full}$ & $N_{clusters}$ (where there is covariance but unclear how to model it)
- Things missing now: intrinsic & noise scattering

Exhausting all information in y-maps?

Data x



Summary statistics t

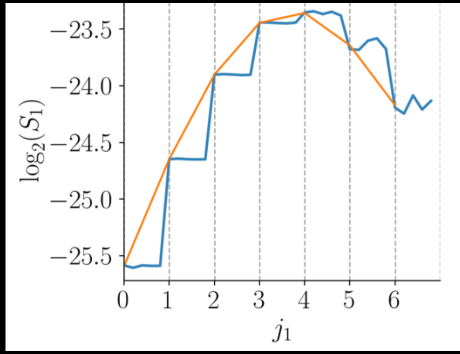
- C_ℓ^{yy}
- $N_{clusters}$

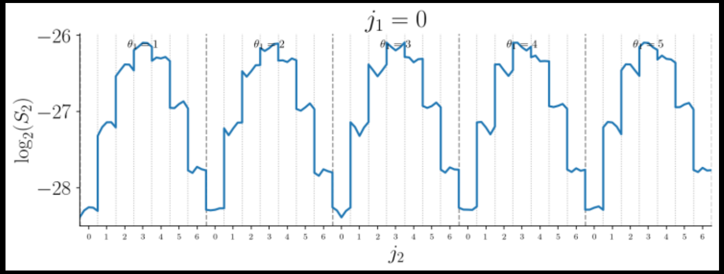
} This work

- Minkowski Functionals: V_0, V_1, V_2
- Peaks
- Minima
- Moments
  - $\sigma_0 = \sqrt{\langle y^2 \rangle}, \sigma_1 = \sqrt{\langle |\nabla y|^2 \rangle}, S_1 = \sqrt{\langle y^2 \nabla^2 y \rangle}, K_1 = \sqrt{\langle y^3 \nabla^2 y \rangle}$

} See Sabyr et al. (2025) for Gaussian likelihood analysis

- Wavelet scattering coefficients: S_0, S_1, S_2 ⟶ Currently working on it
- CNNs
- and many more ...





tszpower

- Fast computation of tSZ power spectrum using Python & JAX with cosmopower emulator
- 2 × faster than class_sz due to JIT compilation on tSZ PS, 10 × faster for trispectrum
- Customized SBI backend based on sbi and PyTorch, interfaced with wandb

XGPaint

- Fast cluster map painting code with Julia backend
- Support Healpix & CAR workspace
- Painting NOT expensive (can provide scaling laws if required)

Time & Space complexities of halo-based simulations

GNFW is 1D grid $y(\theta) = y_0(M,z)I\left(\frac{\theta}{\theta_{500}}\right)$ so building the interpolation grid is cheap

Time complexity is $O(N_{clusters}N_{side}^2)$

General pressure profile is 3D: $y(M,z,\theta)$
Interpolation time is more expensive: $O(N_M N_z N_\theta)$

On Macbook M2 Pro: painting $N_{side} = 1024$ with 300,000 clusters takes 90 s

Cluster catalogue validation