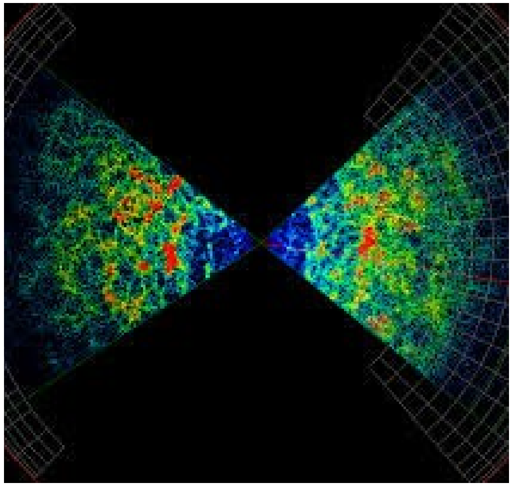
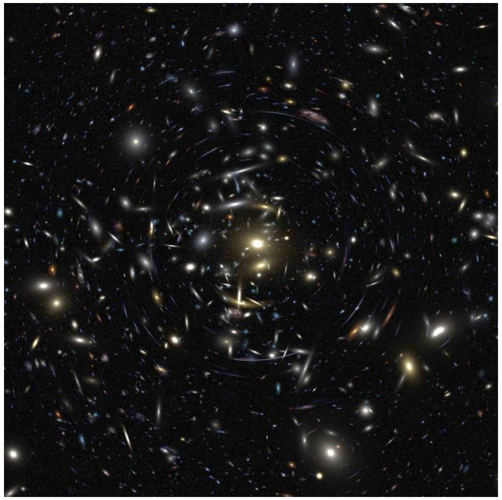
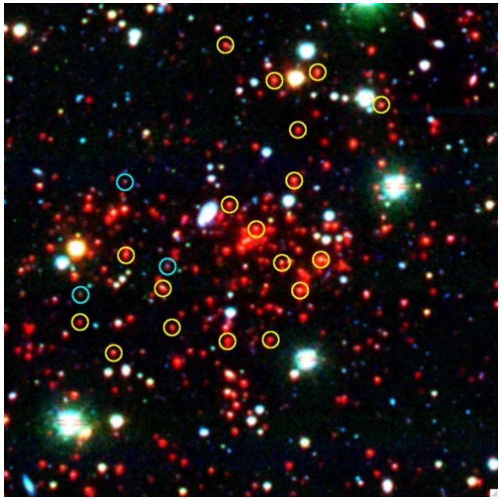
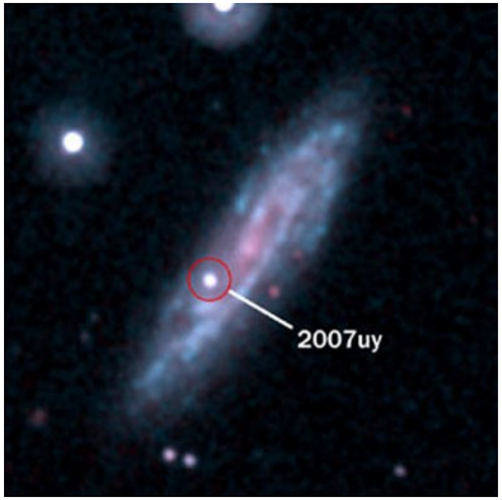
# Analysis challenges and tools for next-generation LSS surveys










*David Alonso*
*STFC Ernest Rutherford Fellow*
*Cardiff University*

*South American Workshop on Cosmology in the LSST Era*
*Dec 19th 2018*

# Analysis challenges and tools for next-generation LSS surveys

**Google "Oxford December"**


**Google "Sao Paulo December"**

*David Alonso*
*STFC Ernest Rutherford Fellow*
*Cardiff University*

*South American Workshop on Cosmology in the LSST Era*
*Dec 19th 2018*

# How will we go about it?

**Parameters**

$$A_s \quad n_s$$
$$r \quad f_{\mathrm{NL}}$$

$$\Omega_{\mathrm{DE}} \quad \Omega_{\mathrm{M}}$$
$$w_a \quad w_0$$

Initial conditions
Energy components
Background evolution

# How will we go about it?

**Parameters**

$$A_s \quad n_s$$
$$r \quad f_{\mathrm{NL}}$$

$$\Omega_{\mathrm{DE}} \quad \Omega_{\mathrm{M}}$$
$$w_a \quad w_0$$

Initial conditions
Energy components
Background evolution

**Observables**

$$\Delta(\mathbf{k}, t)$$
$$\langle |\Delta(\mathbf{k}, t)|^2 \rangle$$
$$\Downarrow$$
$$P(k, t)$$

Matter fluctuations
Power spectrum

# How will we go about it?

**Parameters**

$$\boxed{\begin{matrix} A_s & n_s \\ r & f_{\rm NL} \end{matrix}}$$

$$\boxed{\begin{matrix} \Omega_{\rm DE} & \Omega_{\rm M} \\ w_a & w_0 \end{matrix}}$$

Initial conditions
Energy components
Background evolution

**(Un)observables**

$$\Delta(\mathbf{k},t)$$

$$\langle|\Delta(\mathbf{k},t)|^2\rangle$$

$$\Downarrow$$

$$P(k,t)$$

Matter fluctuations
Power spectrum

**Observables**

$$\Delta^\alpha(\theta,\phi,\lambda)$$

$\alpha$=:
- CMB temperature
- CMB polarisation
- Galaxy density
- Galaxy shapes
- Ly$\alpha$ absorption
- 21cm flux

...

# How will we go about it?

| Parameters | (Un)observables | Observables | Observations |
|---|---|---|---|
| $A_s \quad n_s$ <br> $r \quad f_{\mathrm{NL}}$ <br> $\Omega_{\mathrm{DE}} \quad \Omega_{\mathrm{M}}$ <br> $w_a \quad w_0$ | $\Delta(\mathbf{k},t)$ <br> $\langle\lvert\Delta(\mathbf{k},t)\rvert^2\rangle$ <br> $\Downarrow$ <br> $P(k,t)$ | $\Delta^\alpha(\theta,\phi,\lambda)$ | $\Delta^\alpha_{\mathrm{obs}} \equiv \Delta^\alpha + N^\alpha + S^\alpha$ |
| Initial conditions <br> Energy components <br> Background evolution | Matter fluctuations <br> Power spectrum | α=: <br> CMB temperature <br> CMB polarisation <br> Galaxy density <br> Galaxy shapes <br> Lyα absorption <br> 21cm flux <br> ... | Object catalogues <br> Intensity maps <br> Spectra |

# How will we go about it?

**Parameters** | **(Un)observables** | **Observables** | **Observations**

Theoretical uncertainties

Astrophysical uncertainties

Instrumental noise
Inst. systematics
Selection effects

$$A_s \quad n_s$$
$$r \quad f_{\rm NL}$$

$$\Omega_{\rm DE} \quad \Omega_{\rm M}$$
$$w_a \quad w_0$$

Initial conditions
Energy components
Background evolution

$$\Delta(\mathbf{k}, t)$$
$$\langle|\Delta(\mathbf{k}, t)|^2\rangle$$
$$\Downarrow$$
$$P(k, t)$$

Matter fluctuations
Power spectrum

$$\Delta^\alpha(\theta, \phi, \lambda)$$

α=:
- CMB temperature
- CMB polarisation
- Galaxy density
- Galaxy shapes
- Lyα absorption
- 21cm flux
- ...

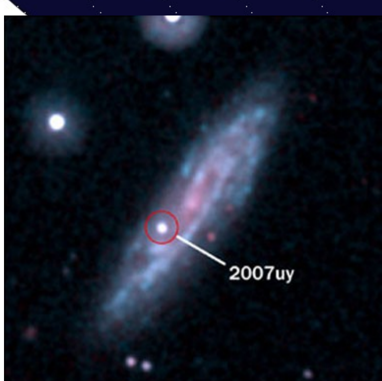
$$\Delta^\alpha_{\rm obs} \equiv \Delta^\alpha + N^\alpha + S^\alpha$$

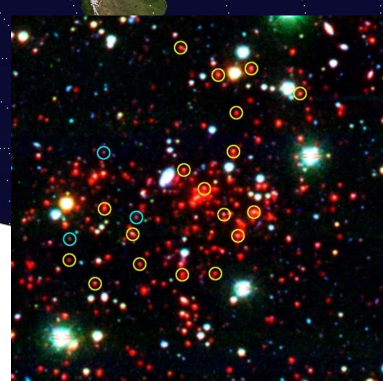
Object catalogues
Intensity maps
Spectra

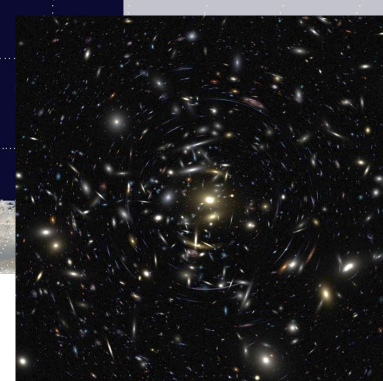
# Example: the LSST

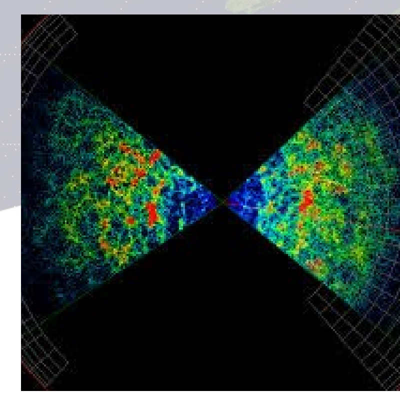

**Outstanding numbers:**
- World's largest imager
  8.4 m, 9.6 sq-deg FOV
- Wide: 20K sq-deg
- Deep: r~27
- Fast: ~100 visits per year
- Big data: ~15 TB per day

**Dark Energy Science Collaboration:**
- Supernovae
- Cluster science
- Strong lensing
- Weak lensing
- Large-scale structure



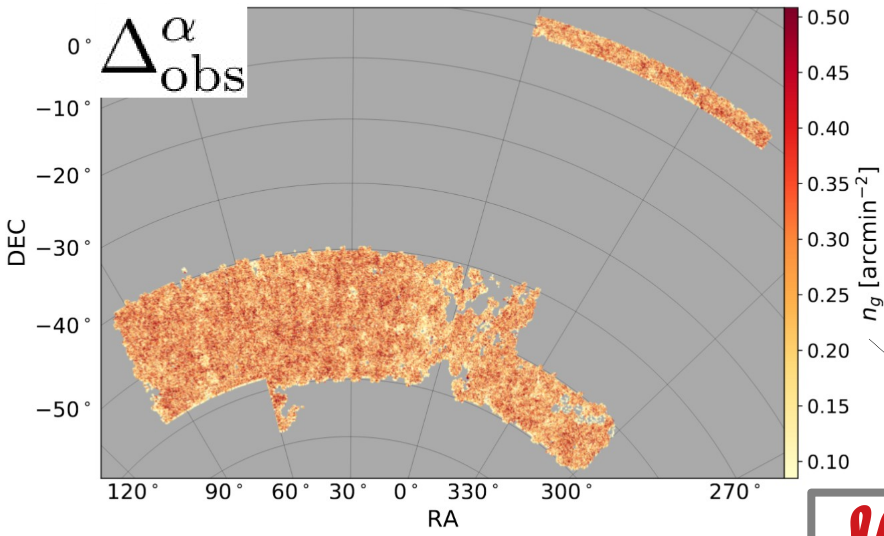







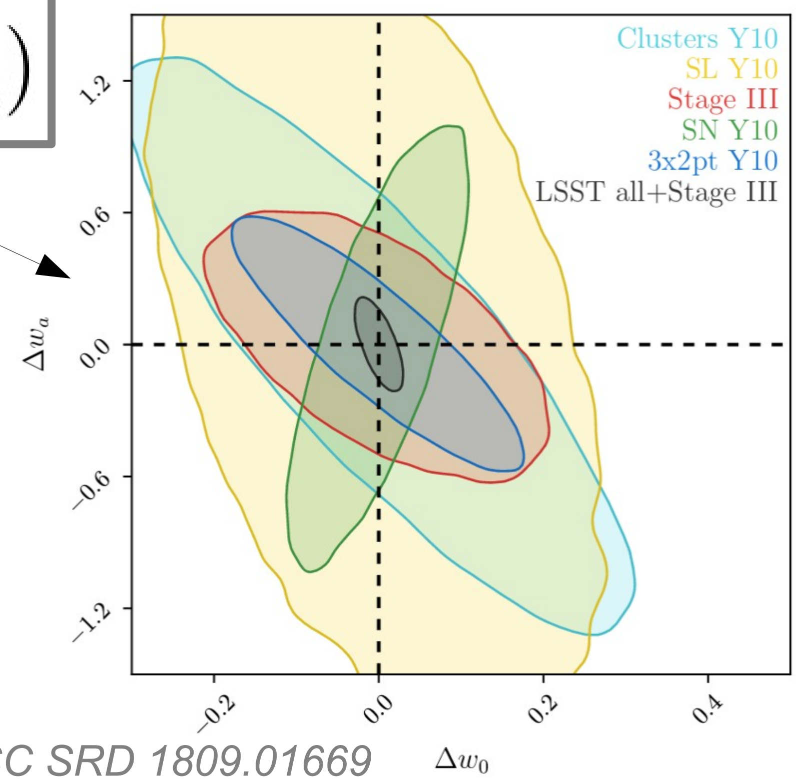
*LSST Coll. et al. 0912.0201*

# Ideal analysis pipeline



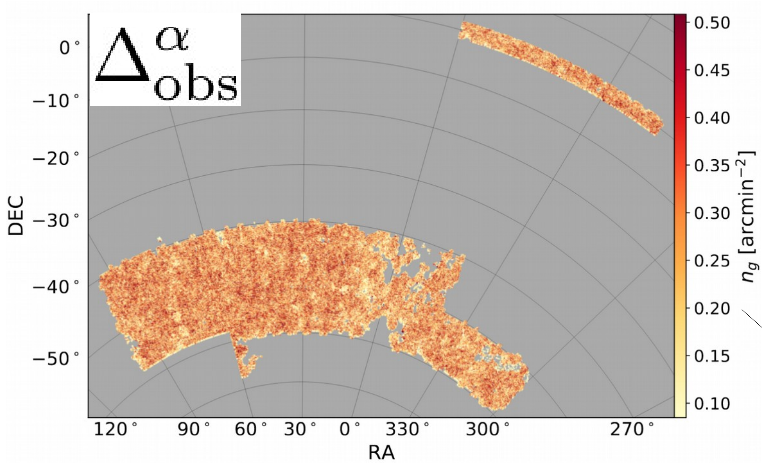

Magic

$$p(w|\Delta^{\alpha}_{\mathrm{obs}})$$



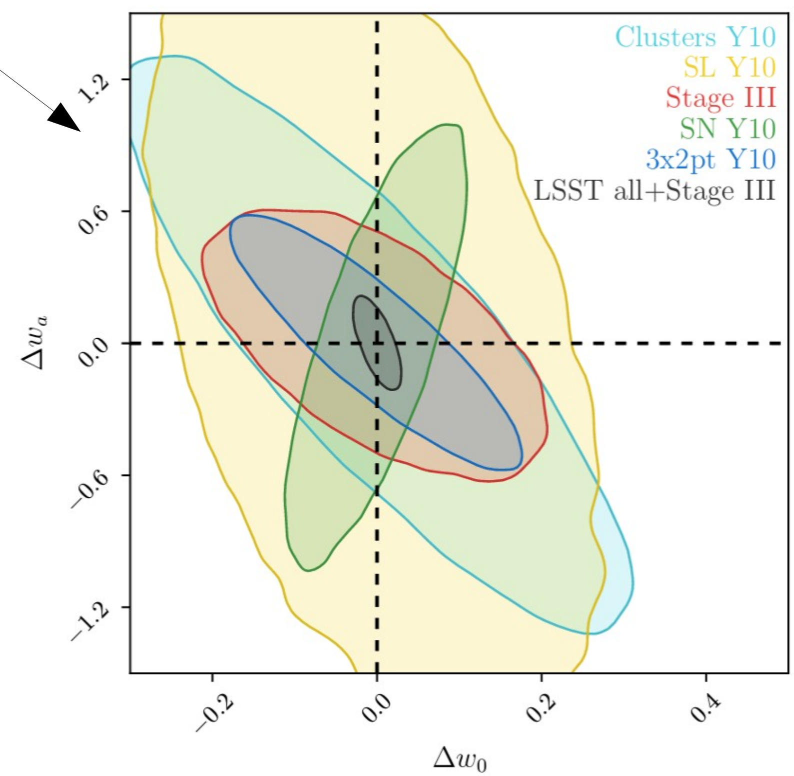

*DESC SRD 1809.01669*

# Ideal analysis pipeline

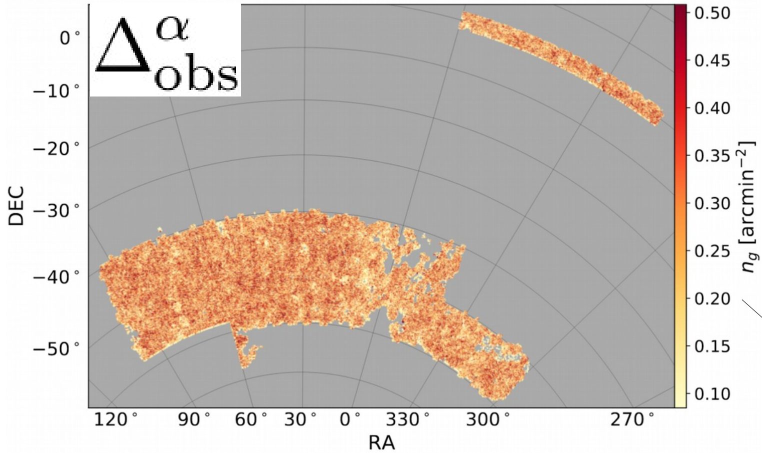



$$p(w|\Delta^{\alpha}_{\mathrm{obs}}) = \int \mathcal{D}\Delta^{\alpha}\,\mathcal{D}\Delta\,\mathcal{D}P_k\,p(w|P_k)\,p(P_k|\Delta)\,p(\Delta|\Delta^{\alpha})\,p(\Delta^{\alpha}|\Delta^{\alpha}_{\mathrm{obs}})$$

- Cosmological model
- Structure formation model
- Astrophysical model
- Instrument/noise model



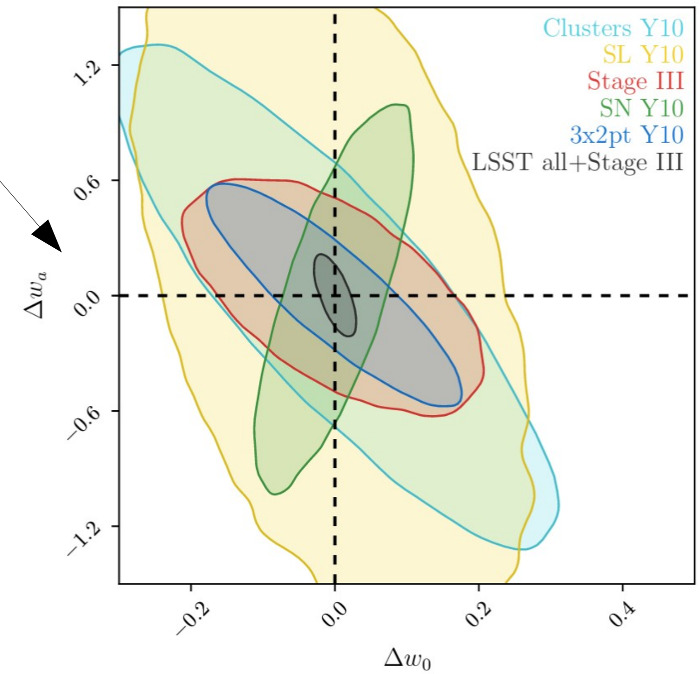
# Ideal analysis pipeline




**BORG:**
*Porqueres et al. 1812.05113*
*Kodi Ramanah et al. 1808.07496*
*Jasche & Lavaux 1806.11117*
*Lavaux & Jasche 1509.05040*
*Jasche & Wandelt 1306.1821*

$$p(w|\Delta^\alpha_{\rm obs}) = \int d\mathbf{s}_{\rm th}\, d\mathbf{s}_{\rm ast}\, d\mathbf{s}_{\rm inst}\, \mathcal{D}\Delta^\alpha\, \mathcal{D}\Delta\, \mathcal{D}P_k\, p(\mathbf{s}_{\rm th}, \mathbf{s}_{\rm ast}, \mathbf{s}_{\rm inst})$$
$$p(w|P_k, \mathbf{s}_{\rm th})\, p(P_k|\Delta, \mathbf{s}_{\rm th})\, p(\Delta|\Delta^\alpha, \mathbf{s}_{\rm th}, \mathbf{s}_{\rm ast})\, p(\Delta^\alpha|\Delta^\alpha_{\rm obs}, \mathbf{s}_{\rm ast}, \mathbf{s}_{\rm inst})$$

- Cosmological model
  - modelling uncertainties
- Structure formation model
  - non-linearities
  - baryonic effects
- Astrophysical model
  - galaxy biasing
  - intrinsic alignment
  - mass-observable relation
- Instrument/noise model
  - photo-z
  - depth variations
  - shape measurement
  - ...

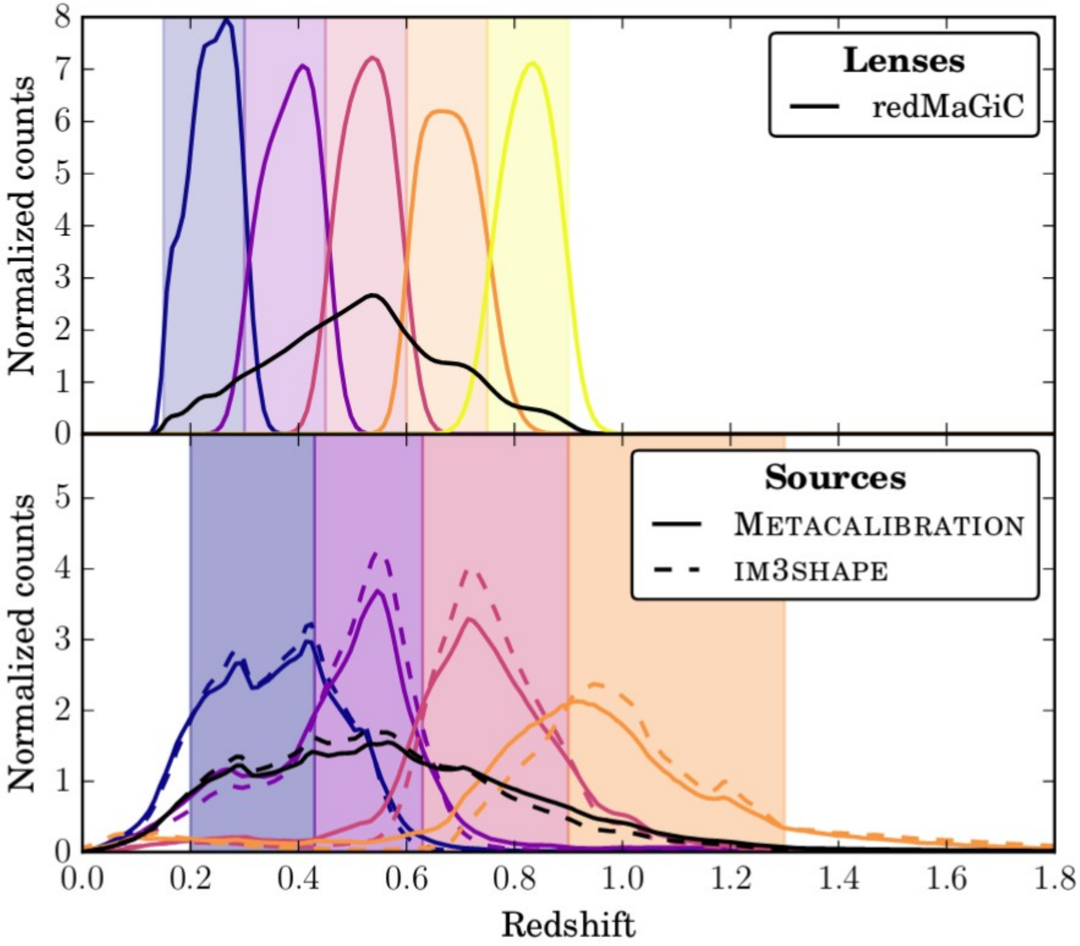


# Simplified 2-point pipeline

- Idea: reduce the dimensionality of your data vector by using only two-point correlations, including all tracer cross-correlations.
  - − Disregard information in higher-order moments, but...
  - + … if the observables are close to Gaussian, all the information is in the one- and two-point cumulants.
  - + Two-point functions are averages over equivalent but independent modes → Gaussian statistics may be a good approximation (CLT).
  - + Lower number of data vector elements.
- Model everything at the summary statistic level
  - − Arguably less optimal systematics marginalization
  - + Fewer effective nuisance parameter.
  - + Possibly less sensitive to modelling uncertainties.

# Example: tomographic analysis

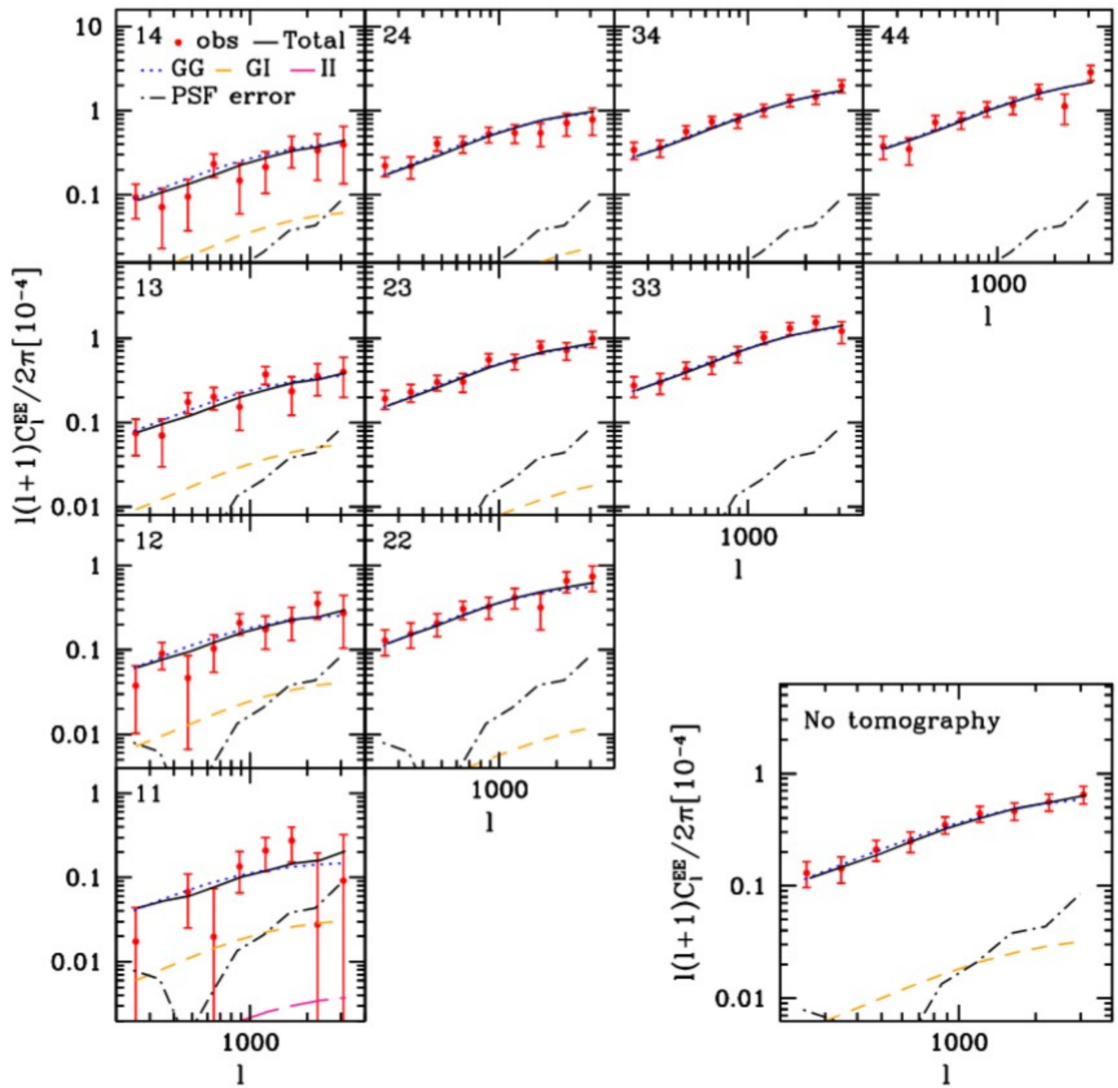
- Photo-zs are complicated.
- Bunch galaxies up into photo-z bins and project onto the sphere.




*DES Y1 data*

# Example: tomographic analysis

- Photo-zs are complicated.
- Bunch galaxies up into photo-z bins and project onto the sphere.
- Compute all possible two-point cross-correlations (different bins, different observables).
- Model them and use them (all or some) to get cosmological parameters.




*HSC Y1 data*

# Example: tomographic analysis

Photo-z estimation

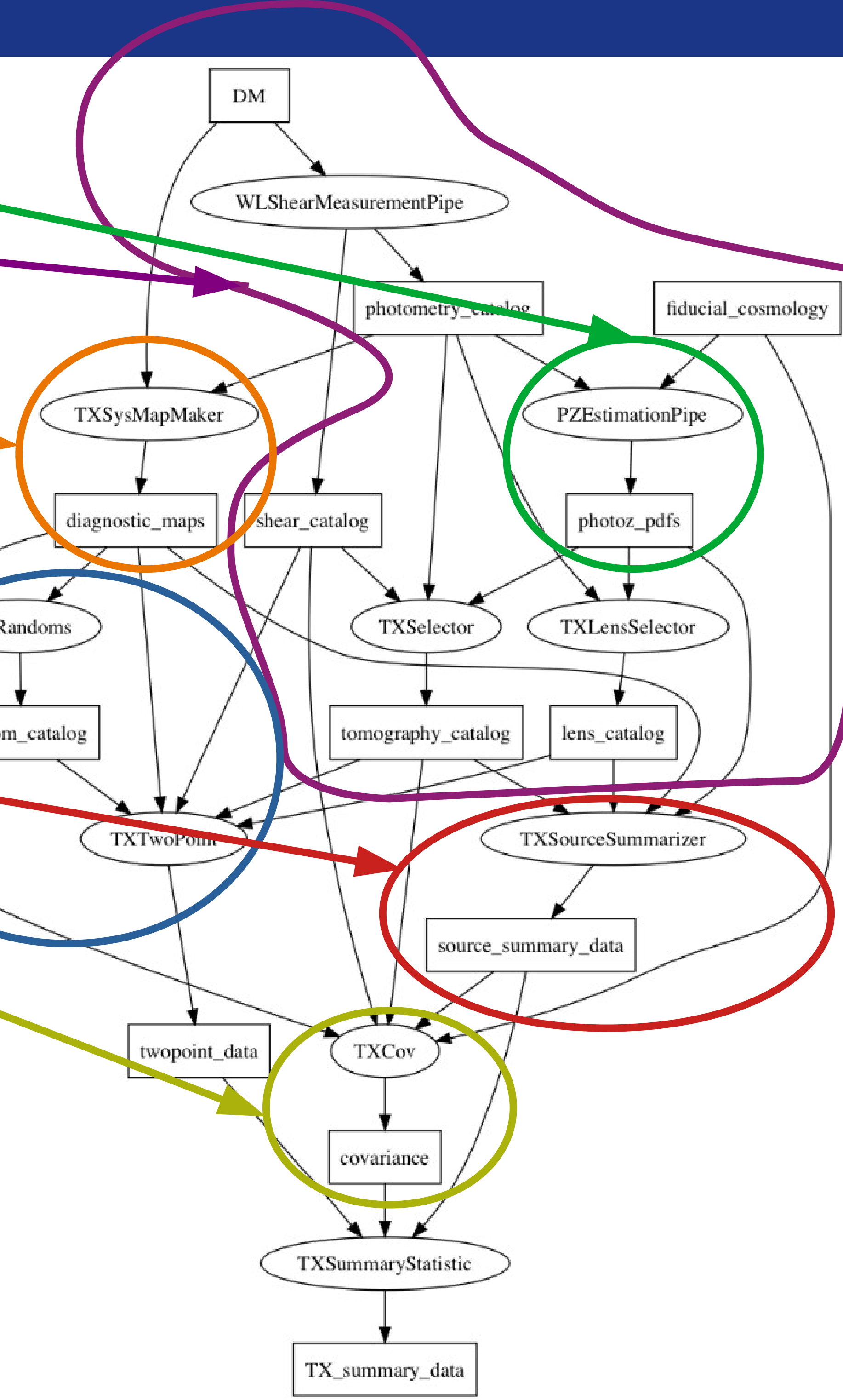
Sample selection (lenses, sources, tomographic bins)

Survey geometry (mask), depth maps, sky systematics

Estimate two-point functions $C^{ij}(\ell)$, $\xi^{ij}(\theta)$

Estimate redshift distributions

Estimate covariance matrix




**Gaussian likelihood**

$$-2 \log P(d|\theta) = (d-t(\theta))^T C^{-1} (d-t(\theta)) + L_0$$

Vector of cross-correlations

Theory prediction

# Example: tomographic analysis

Photo-z estimation

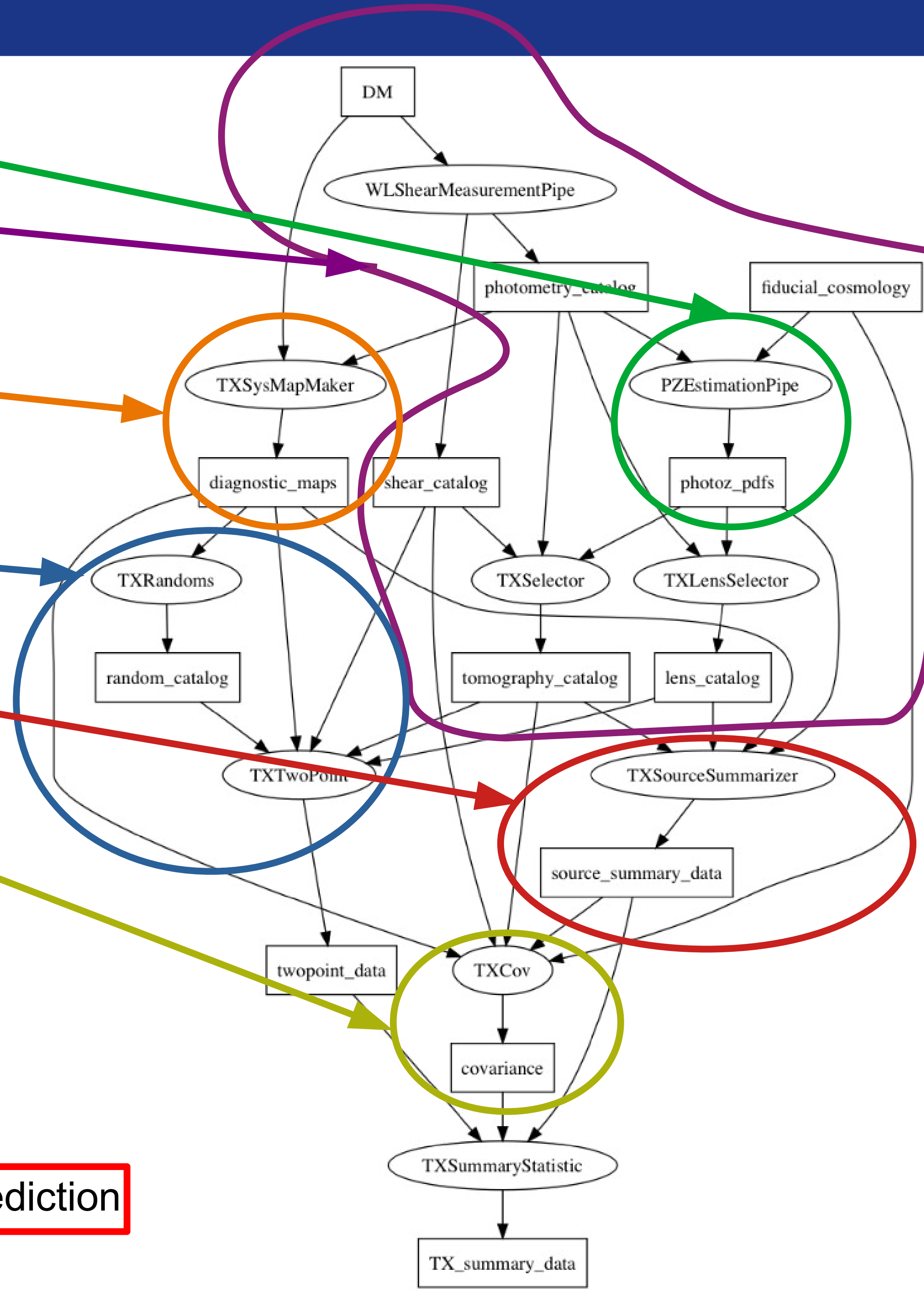
Sample selection (lenses, sources, tomographic bins)

Survey geometry (mask), depth maps, sky systematics

Estimate two-point functions $C^{ij}(\ell)$, $\xi^{ij}(\theta)$

Estimate redshift distributions

Estimate covariance matrix




**Gaussian likelihood**

$$-2 \log P(d|\theta) = (d-t(\theta))^T C^{-1} (d-t(\theta)) + L_0$$

Vector of cross-correlations

Theory prediction

*A unified pseudo-$C_\ell$ estimator*

DA, F.J. Sanchez, A. Slosar

arXiv:1809.09603

# Why power spectra?

- Power spectrum cleanly separates theoretically well-understood large-scales from small, non-linear scales:
  - k-cuts have clear interpretation
  - No Hankel transforms, no hand-waving about linear biasing

- Covariance matrix of power spectrum measurements is much more diagonal that correlation function
  - Can do $\chi^2$ by eye
  - Arguably need fewer MC samples if calculating/checking covariance from mocks

- Better scaling performance:
  - Scales $\sim N^{3/2}$ with good prefactor after coupling matrix has been calculated
  - Naive pair-counting scales as $N^2$ (can be improved to $\sim N$ for tree-codes, but not in all cases)

# Power spectrum estimation

Optimal quadratic estimation ($B_j$ below):

$$\overbrace{\mathsf{F}_{ij}}^{\text{Mode-coupling}} B_j = \frac{1}{2}\underbrace{\mathbf{a}^\dagger \tilde{\mathsf{C}}^{-1}\,\mathsf{P}_i\,\overbrace{\tilde{\mathsf{C}}^{-1}\mathbf{a}}^{\text{Inverse-variance-weight data}}}_{\text{Fourier-transform and square}} - \frac{1}{2}\underbrace{\mathrm{Tr}\left[\tilde{\mathsf{C}}^{-1}\mathsf{P}_i\tilde{\mathsf{C}}^{-1}\,\mathsf{N}\right]}_{\text{Noise bias}}$$

In the simplest scenario (full sky, homogeneous/no noise) this corresponds simply to:

$$C_\ell = \frac{1}{2\ell+1}\sum_{m=-\ell}^{\ell} |a_{\ell m}|^2$$

However, in any realistic scenario, this estimator implies inverting $N_{pix}$ x $N_{pix}$ matrices, which can be horribly slow for high-resolutions (even using smart methods).

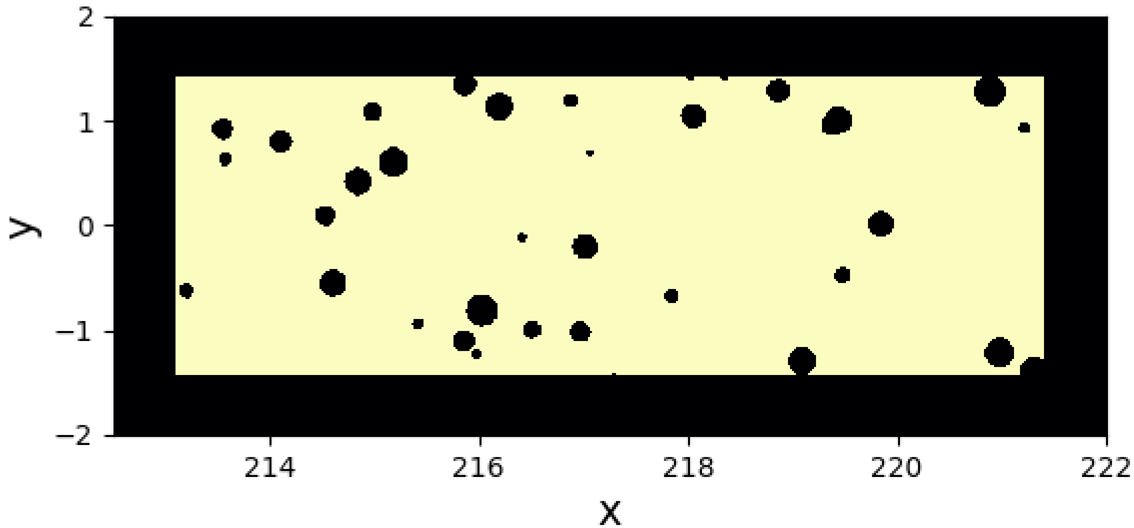
# The pseudo-$C_\ell$ estimator

The PCL estimator attempts to use the simplest scenario ("SHT, square and sum") in a real-world one:

1. Mask your field. $\mathbf{a}^v \equiv v(\hat{\boldsymbol{\theta}})\mathbf{a}(\hat{\boldsymbol{\theta}})$

   Minimally, this mask "v" includes knowledge about which regions you have observed (v=1) and which ones you haven't (v=0).

   More generally, the mask can be thought of as a local inverse-variance weight $v \propto 1/\sigma^2$ (e.g. infinite noise if you haven't observed a given pixel).



# The pseudo-$C_\ell$ estimator

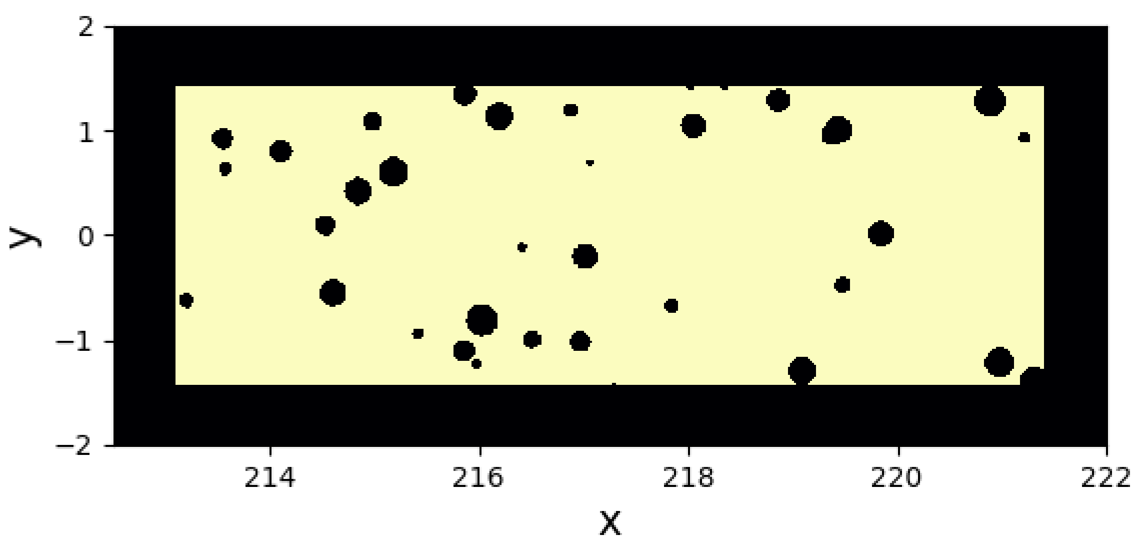
The PCL estimator attempts to use the simplest scenario ("SHT, square and sum") in a real-world one:

1. Mask your field. $\mathbf{a}^v \equiv v(\hat{\boldsymbol{\theta}})\mathbf{a}(\hat{\boldsymbol{\theta}})$

   Minimally, this mask "v" includes knowledge about which regions you have observed (v=1) and which ones you haven't (v=0).

   More generally, the mask can be thought of as a local inverse-variance weight $v \propto 1/\sigma^2$ (e.g. infinite noise if you haven't observed a given pixel).

2. Fourier/Harmonic-transform the masked field, square and average over m.



# The pseudo-$C_\ell$ estimator

The PCL estimator attempts to use the simplest scenario ("SHT, square and sum") in a real-world one:

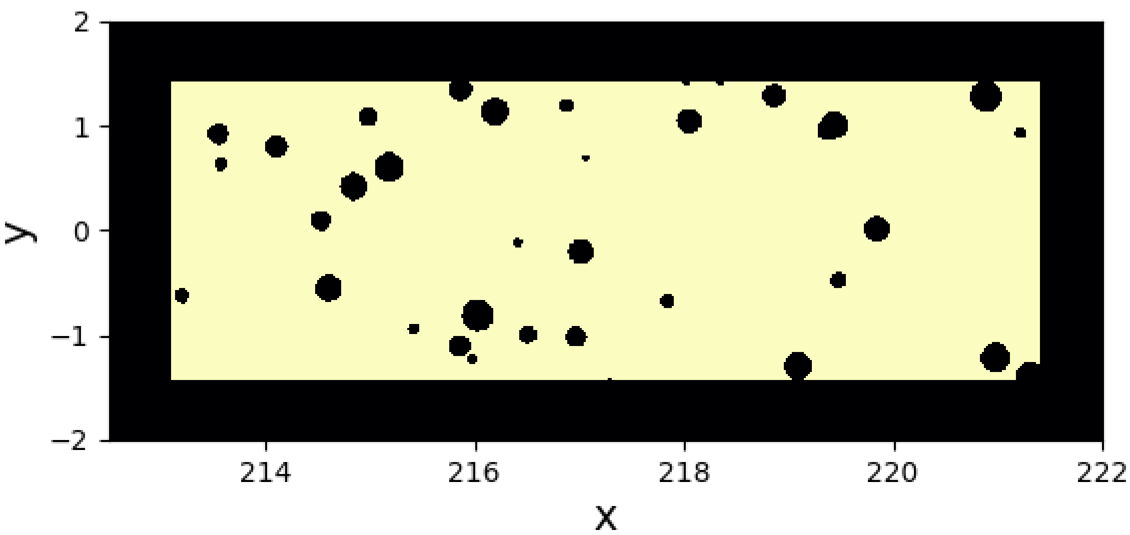
1. Mask your field. $\mathbf{a}^v \equiv v(\hat{\boldsymbol{\theta}})\mathbf{a}(\hat{\boldsymbol{\theta}})$

   Minimally, this mask "v" includes knowledge about which regions you have observed (v=1) and which ones you haven't (v=0).

   More generally, the mask can be thought of as a local inverse-variance weight $v \propto 1/\sigma^2$ (e.g. infinite noise if you haven't observed a given pixel).

2. Fourier/Harmonic-transform the masked field, square and average over m.
3. Figure out mode coupling induced by masking.This can be done analytically!

The PCL is then significantly faster, with an $\propto N_{pix}^{3/2}$ ($\ell^3_{max}$) scaling.



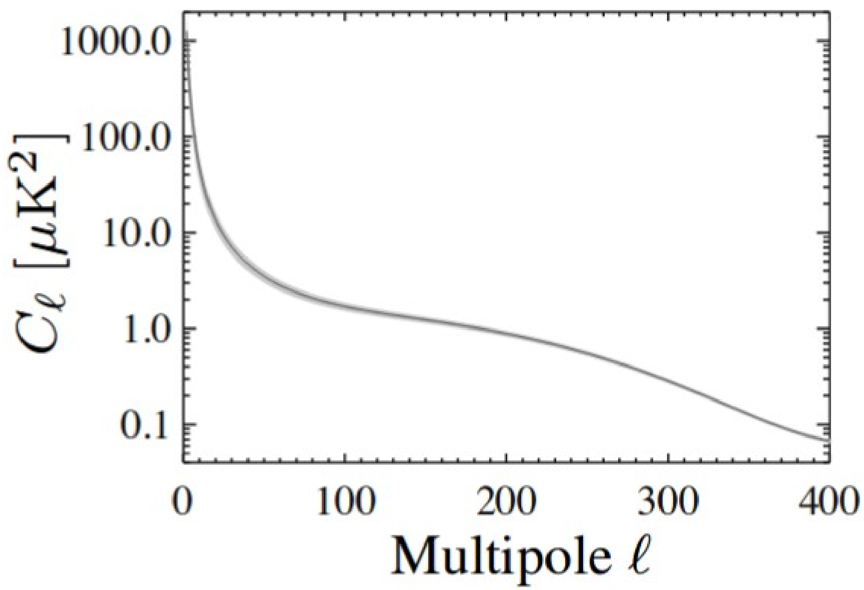
# The pseudo-$C_\ell$ estimator

The PCL estimator can be thought of in **two ways**:

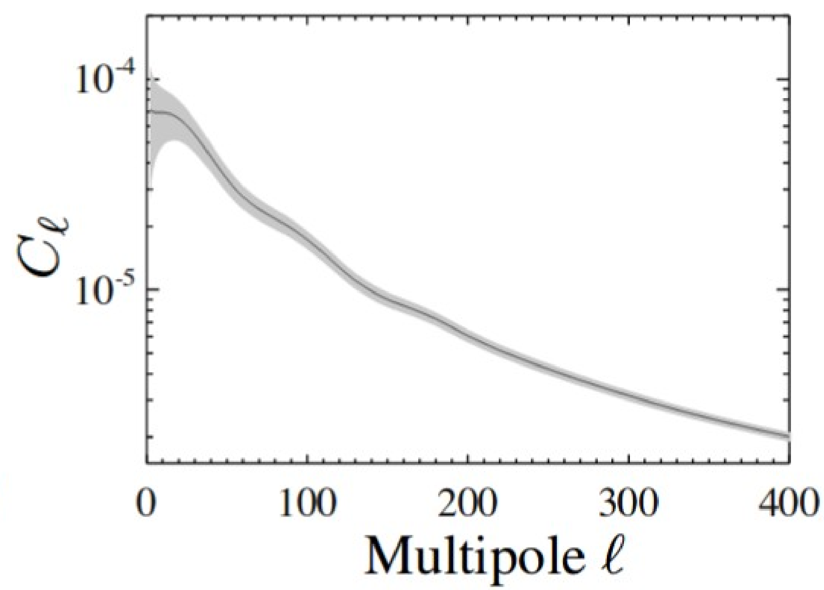
- It is what one would intuitively do:
  - Fourier transform, square
  - Correct for the fact that you shouldn't be doing that

- It is an approximation to the maximum likelihood solution that approximates the covariance matrix as diagonal for the purpose of weighting:
  This will work, when this is a good approximation:
  - Full sky data
  - Flat underlying power-spectrum
  - Noise domination (e.g. shot noise is perfectly flat)

*Leistedt et al. 1306.0005*

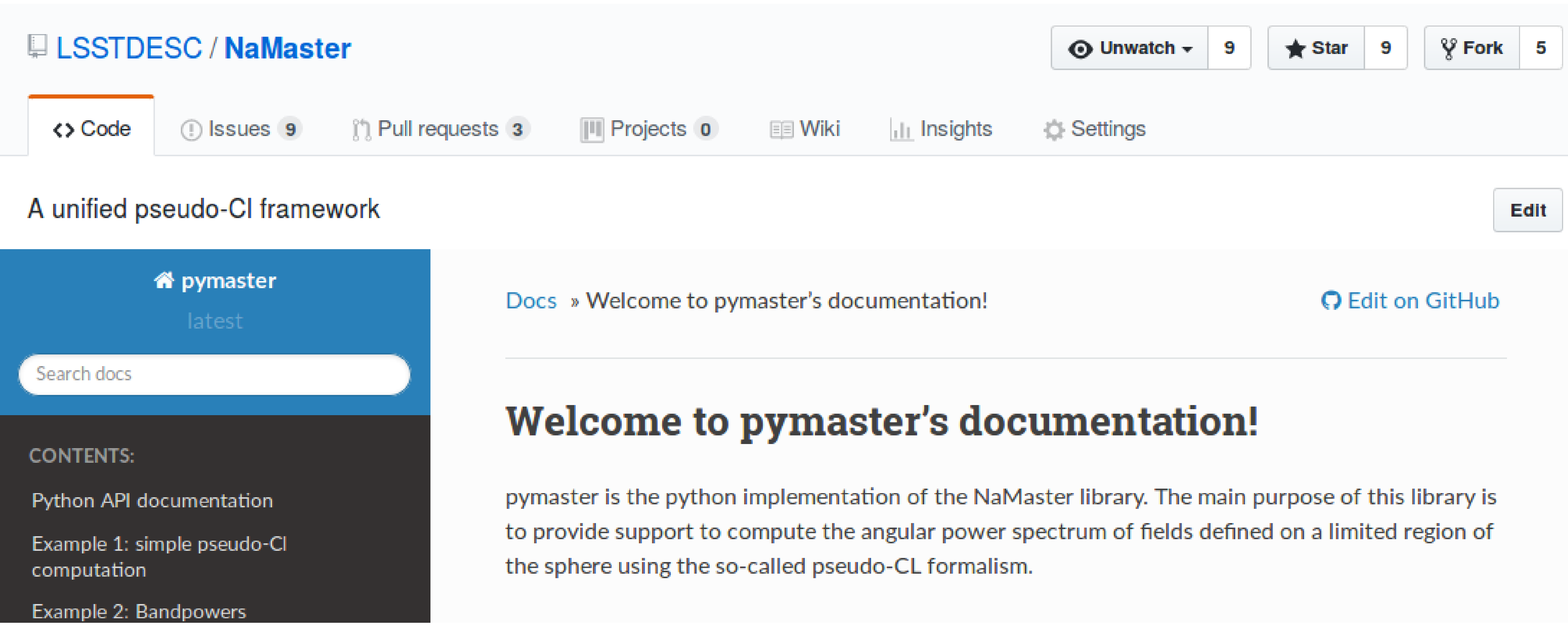



(a) CMB spectrum




(b) CMASS spectrum

# A unified pseudo-$C_\ell$ code




**Code:** https://github.com/LSSTDESC/NaMaster
**Docs:** https://namaster.readthedocs.io/en/latest/index.html

# A unified pseudo-$C_\ell$ code

## Example 5: Using workspaces

This sample script showcases the use of the NmtWorkspace class to speed up the computation of multiple power spectra with the same mask. This is the most general example in this suite, showing also the correct way to compare the results of the MASTER estimator with the theory power spectrum.

```
import numpy as np
import healpy as hp
import matplotlib.pyplot as plt
import pymaster as nmt

#This script showcases the use of NmtWorkspace objects to speed up the
#computation of power spectra for many pairs of fields with the same masks.

#HEALPix map resolution
nside=256

#We start by creating some synthetic masks and maps with contaminants.
#Here we will focus on the cross-correlation of a spin-2 and a spin-1 field.
#a) Read and apodize mask
mask=nmt.mask_apodization(hp.read_map("mask.fits",verbose=False),1.,apotype="Smooth")
#b) Read maps
mp_t,mp_q,mp_u=hp.read_map("maps.fits",field=[0,1,2],verbose=False)
#c) Read contaminants maps
tm_t,tm_q,tm_u=hp.read_map("temp.fits",field=[0,1,2],verbose=False)
#d) Create contaminated fields
#   Spin-0
f0=nmt.NmtField(mask,[mp_t+tm_t],templates=[[tm_t]])
#   Spin-2
f2=nmt.NmtField(mask,[mp_q+tm_q,mp_u+tm_u],templates=[[tm_q,tm_u]])
#e) Create binning scheme. We will use 20 multipoles per bandpower.
b=nmt.NmtBin(nside,nlb=20)
```

**Code:** https://github.com/LSSTDESC/NaMaster
**Docs:** https://namaster.readthedocs.io/en/latest/index.html

LSSTDESC / Star 9 Fork 5
Code
A unified pseudo
Edit
pym
lat
Search docs
Edit on GitHub
CONTENTS:
Python API documen
Example 1: simple ps computation
Example 2: Bandpow
se of this library is
a limited region of

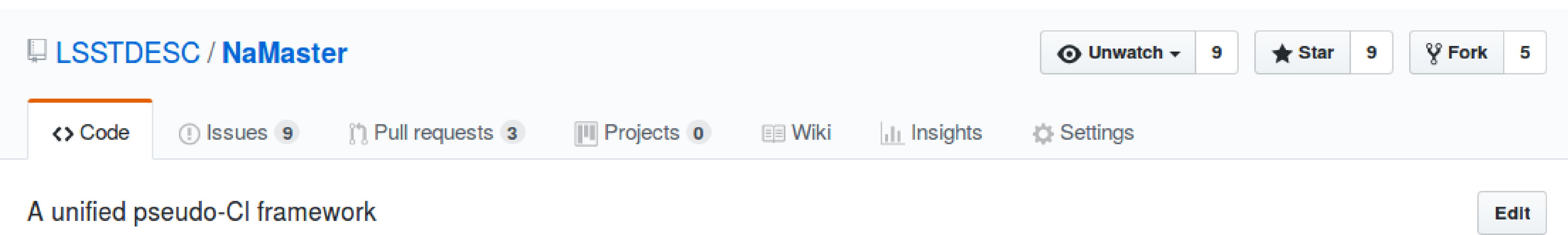
# Why another code?

There are many public codes to measure power spectra, e.g.:

- Xpol (https://gitlab.in2p3.fr/tristram/Xpol)
- PolSpice (http://www2.iap.fr/users/hivon/software/PolSpice/)
- Xpure (https://gitlab.in2p3.fr/tristram/Xpure)
- Many more. Sorry if you don't see yours here!

All of them have some features that we need, none of them has <u>all the features</u>.
We needed code we understand and can become a standard toolkit:

- Have a wide range of convenience features (next slide)
- Validated
- Documented
- Continuously supported
- Easy to install and use

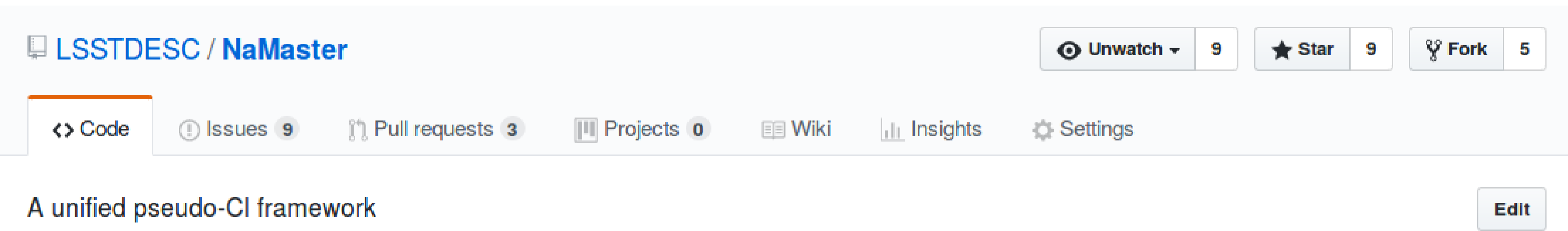



**Code:** https://github.com/LSSTDESC/NaMaster
**Docs:** https://namaster.readthedocs.io/en/latest/index.html

# Why another code?

What **features** does it implement?

- Calculate PCL power spectra (including coupling matrix, etc.)
- Capable of doing both:
  - Full spherical case using spherical transforms
  - Flat-sky patches using 2D FFT
- Capable of doing both:
  - Spin-0 fields (density, CMB temperature)
  - Spin-2 fields (shear, CMB polarization)
  - Cross-correlations
- Bells and whistles:
  - Mode deprojection
  - E/B mode purification

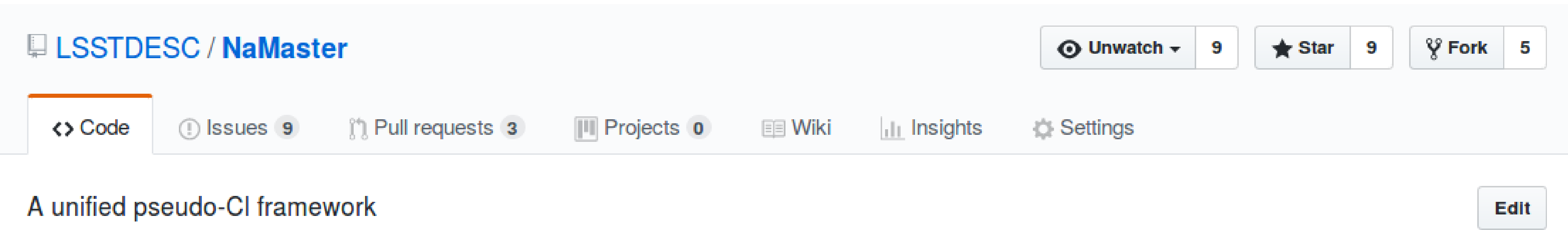



**Code:** https://github.com/LSSTDESC/NaMaster
**Docs:** https://namaster.readthedocs.io/en/latest/index.html

# Why another code?

What **features** does it implement?

- Calculate PCL power spectra (including coupling matrix, etc.)
- Capable of doing both:
  - Full spherical case using spherical transforms
  - Flat-sky patches using 2D FFT
- Capable of doing both:
  - Spin-0 fields (density, CMB temperature)
  - Spin-2 fields (shear, CMB polarization)
  - Cross-correlations
- Bells and whistles:
  - Mode deprojection
  - E/B mode purification




**Code:** https://github.com/LSSTDESC/NaMaster
**Docs:** https://namaster.readthedocs.io/en/latest/index.html

# Mode deprojection

*A. Slosar: "The greatest thing since sliced bread"*

- **Masking:** if I have a bad pixel, I make sure it doesn't get used.
- **Mode deprojection** is the extension of this idea into an arbitrary linear combination of pixels.
  Imagine contaminating your data field as

True map

Observed map

$$\delta_i^c = \delta_i + \alpha m_i$$

Contaminant template (e.g. dust map)

A proper analysis would marginalize over $\alpha$.

*Leistedt et al. 1306.0005*
*Elsner et al. 1609.03577*

# Mode deprojection

*A. Slosar: "The greatest thing since sliced bread"*

- **Masking:** if I have a bad pixel, I make sure it doesn't get used.
- **Mode deprojection** is the extension of this idea into an arbitrary linear combination of pixels.
  Imagine contaminating your data field as

True map

Observed map

$$\delta_i^c = \delta_i + \alpha m_i$$

Contaminant template (e.g. dust map)

A proper analysis would marginalize over $\alpha$.

If you do the maths, in PCL this amounts to:

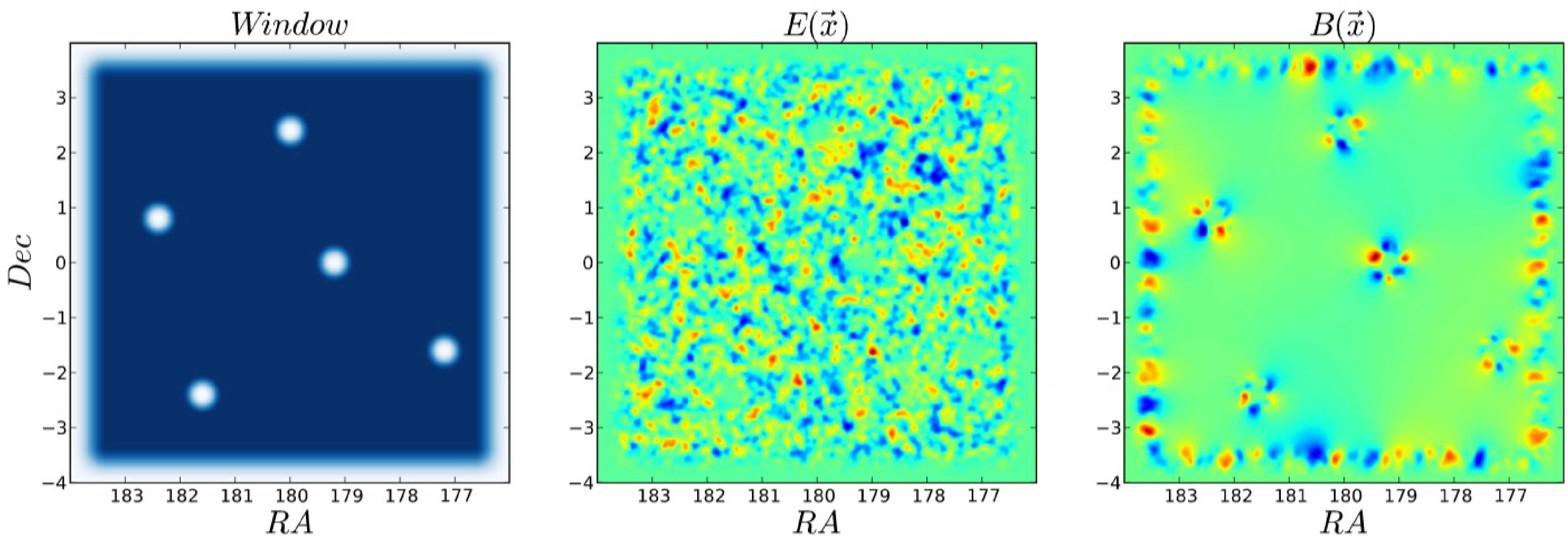
- Finding the best fit value of $\alpha$.
- Subtracting a contaminant map from the data using this $\alpha$
- Calculate the PCL estimates and **correct for the bias** this subtraction has produced
- Multiply by the inverse of the mode-coupling matrix

*Leistedt et al. 1306.0005*
*Elsner et al. 1609.03577*

# E/B purification

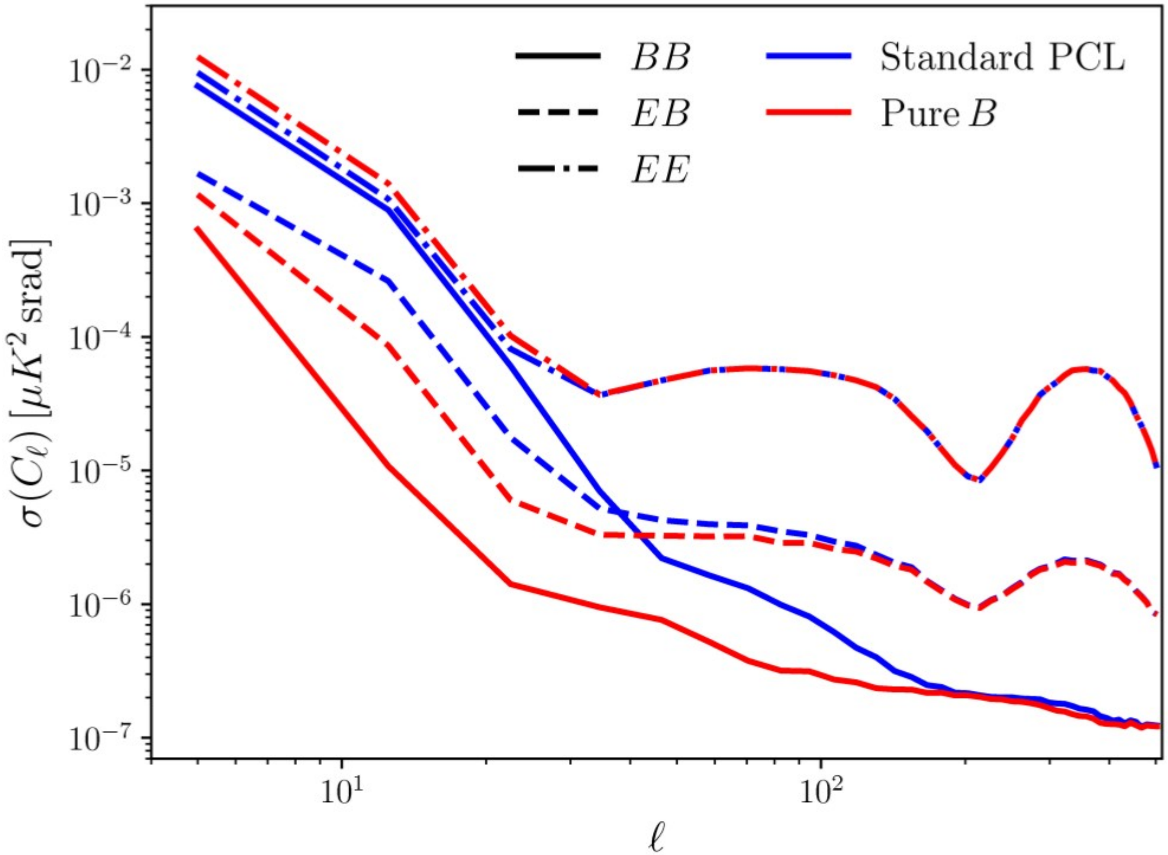
- A sky mask mixes E and B modes. Effectively, it generates ambiguous modes.
- A standard pseudo-Cl algorithm, by construction will give you an unbiased estimate of the power spectrum. It will separate E and B at the level of the power spectrum.
- However, if E>>B, the contamination of E in the B map leaks into the variance of the estimator, making it very suboptimal.
- E/B purification consists of projecting out all ambiguous E or B modes at the map level. Effectively we lose a bit of signal, but it pays off in terms of estimator signal-to-noise.

*Louis et al. 1306.6692*



# E/B purification

- A sky mask mixes E and B modes. Effectively, it generates ambiguous modes.
- A standard pseudo-Cl algorithm, by construction will give you an unbiased estimate of the power spectrum. It will separate E and B at the level of the power spectrum.
- However, if E>>B, the contamination of E in the B map leaks into the variance of the estimator, making it very suboptimal.
- E/B purification consists of projecting out all ambiguous E or B modes at the map level. Effectively we lose a bit of signal, but it pays off in terms of estimator signal-to-noise.
- This is vital for CMB B-mode searches. But it's also useful to quantify lensing systematics.



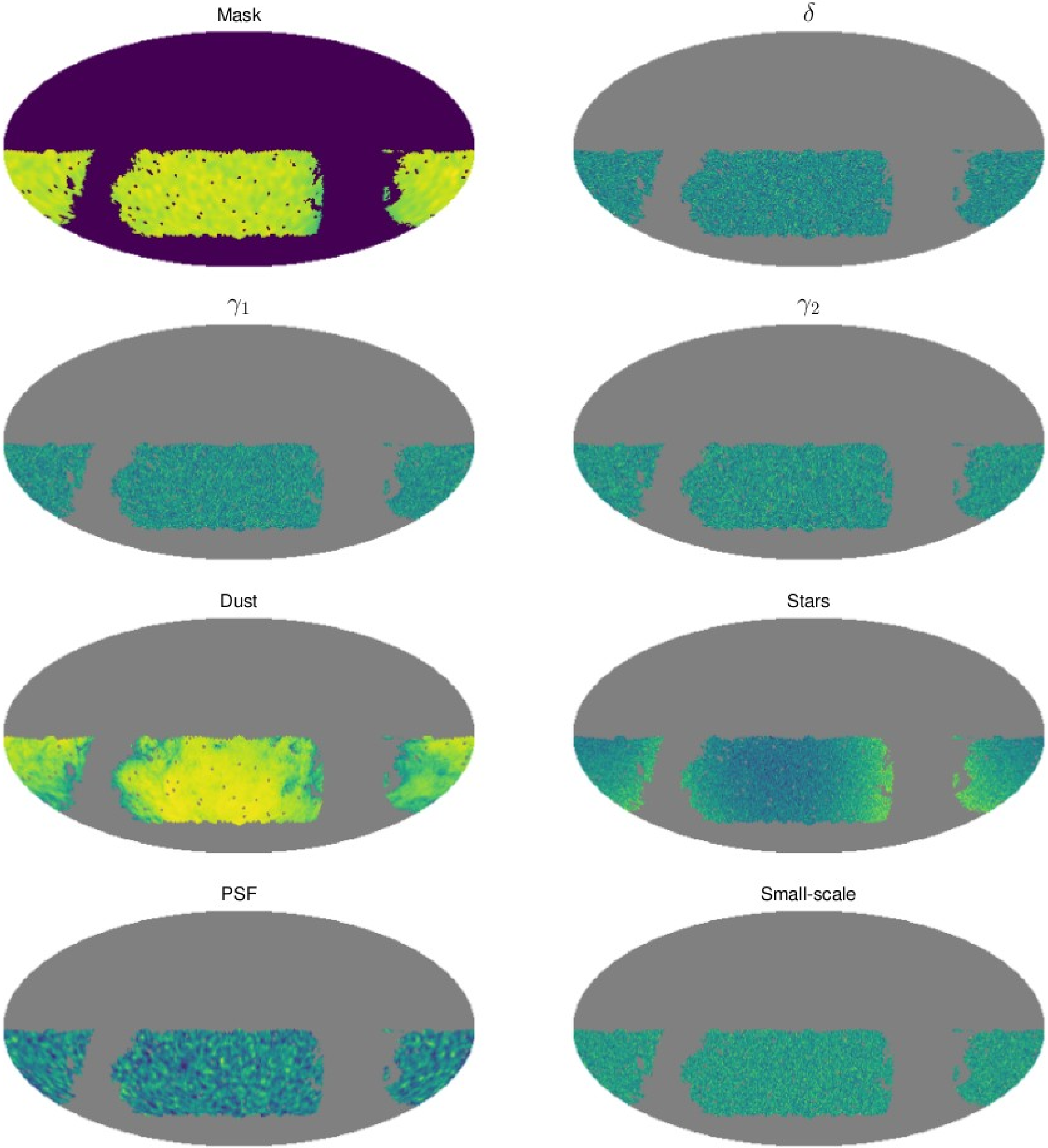
# Code validation

2 validation suites:

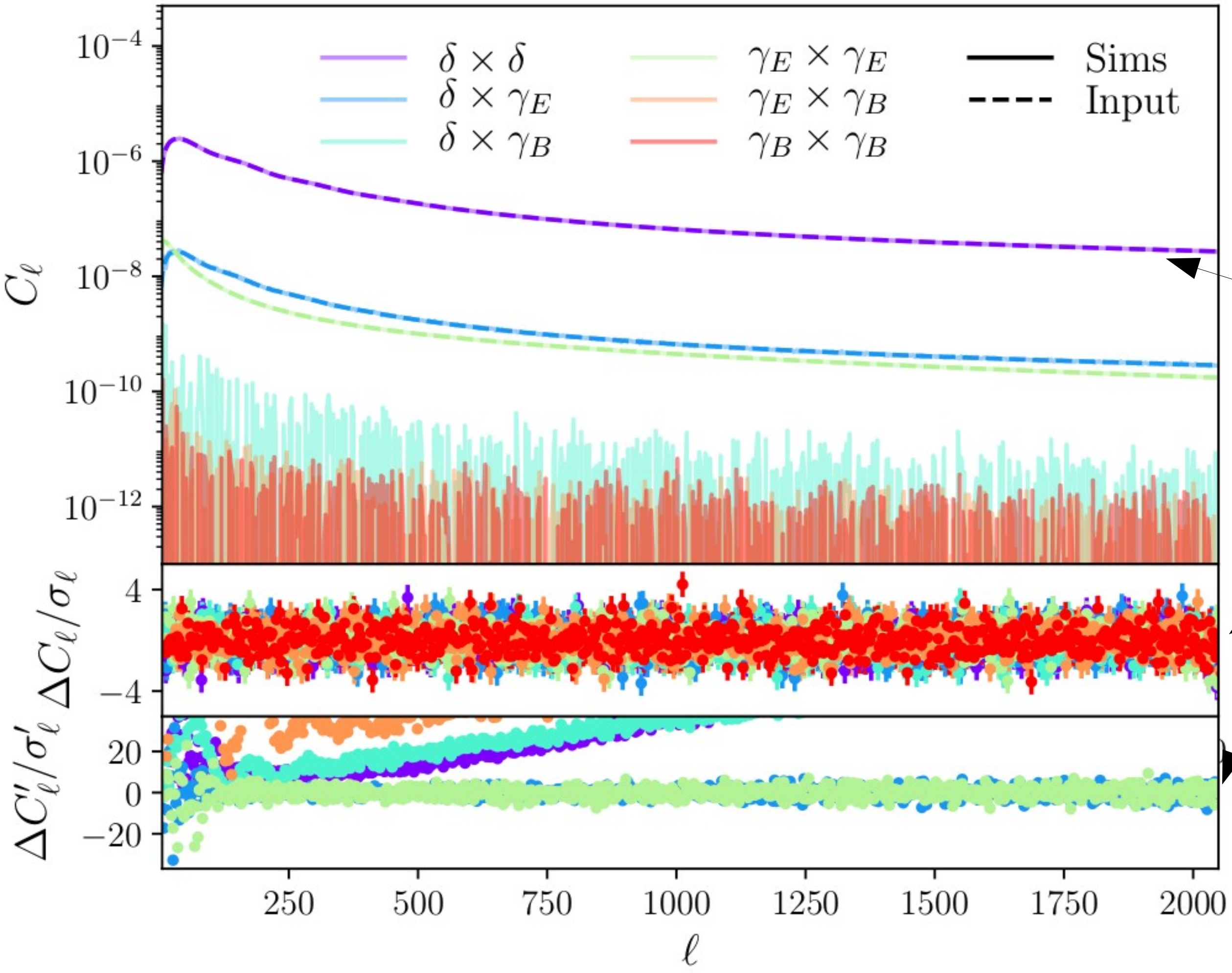
- **LSS:** galaxy clustering and lensing with a large set of contaminants.
- **CMB:** B-mode and lensing experiments with foreground contamination.

1000 Gaussian simulations

- w./w.o. contaminant deprojection
- w./w.o. E/B purification.
- curved and flat skies.



# Code validation

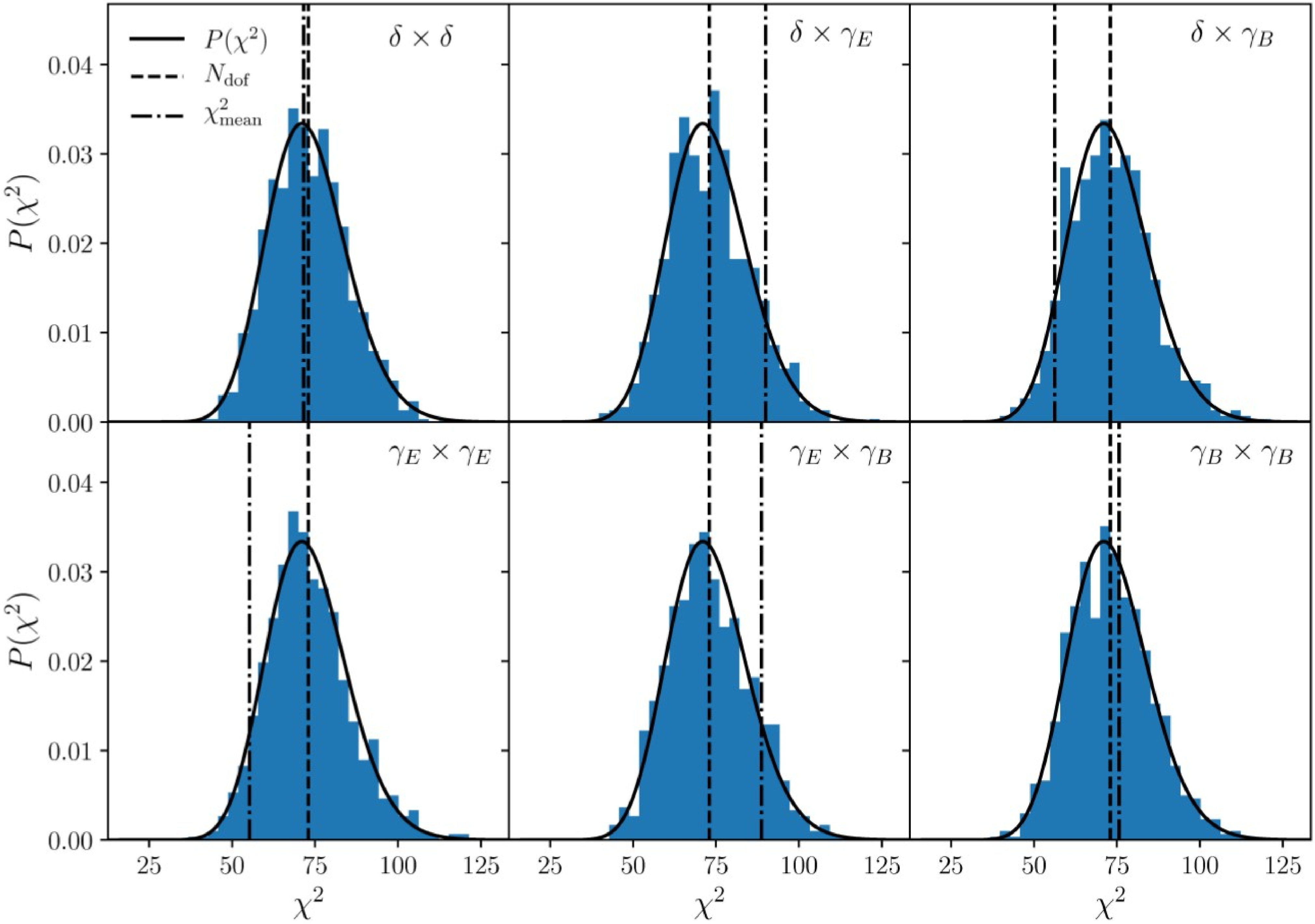



Input power spectra recovered.

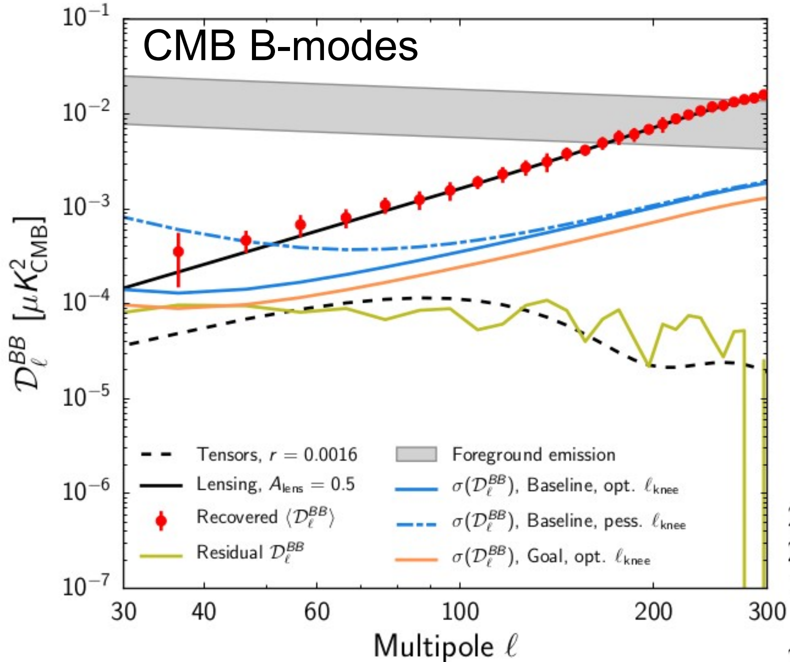
Deprojection is important!

Residuals are as expected.

# Code validation

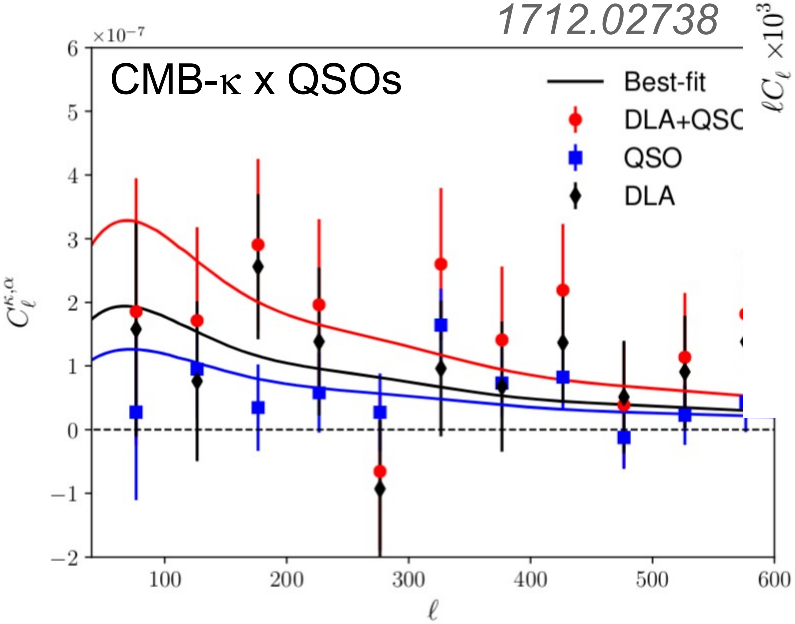


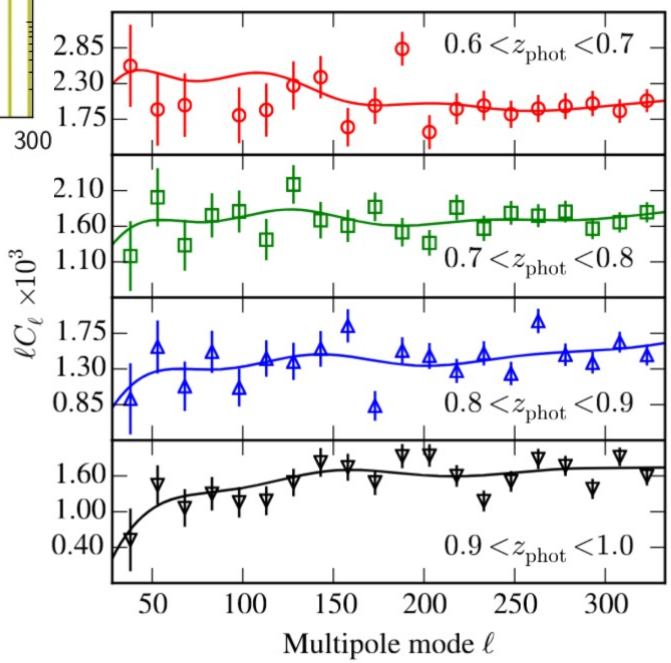
# NaMaster




*1808.07445*

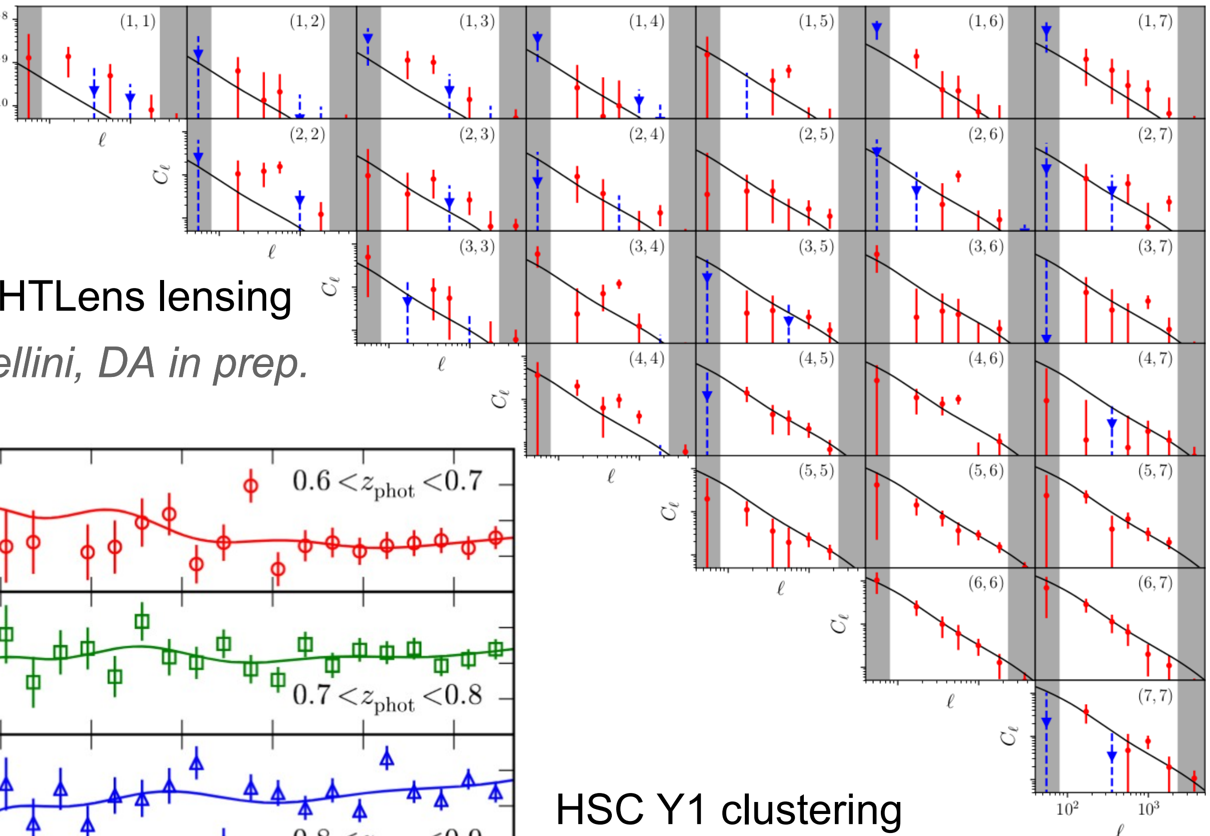
*1712.02738*



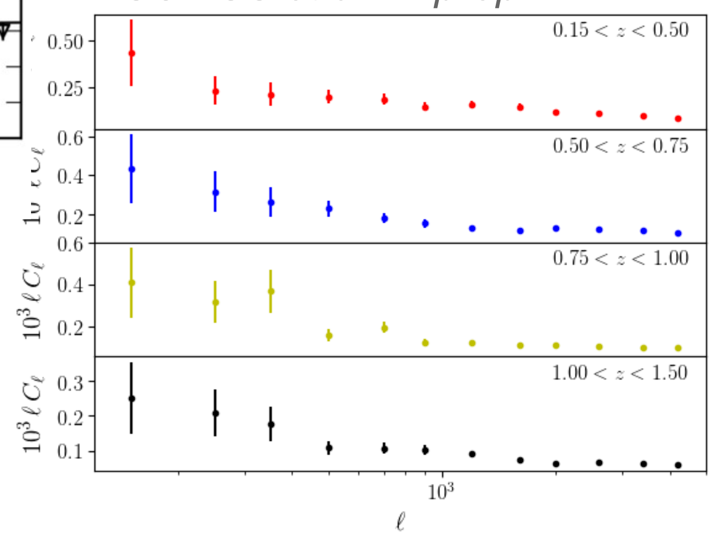

DES Y1 clustering

*1807.10163*



CFHTLens lensing

*Bellini, DA in prep.*




HSC Y1 clustering

*DESC LSS et al. in prep.*

**Code:** https://github.com/LSSTDESC/NaMaster

**Docs:** https://namaster.readthedocs.io/en/latest/index.html

*Science-driven 3D data compression*

DA, arXiv:1707.08950

*Cosmic shear with 2 modes*

E. Bellini, DA
in preparation

# Covariance matrices and data compression

A tomographic two-point function analysis already compresses the initial data vector significantly:

Catalogue with ~billions of objects and >5 quantities per object

$\downarrow$

A number of cross-correlations between sub-samples of these

What is the actual number of cross correlations?

# Covariance matrices and data compression

A tomographic two-point function analysis already compresses the initial data vector significantly:

Catalogue with ~billions of objects and >5 quantities per object

$\downarrow$

A number of cross-correlations between sub-samples of these

What is the actual number of cross correlations?

Let's take an ideal LSST as an example:
10 redshift bins for lensing. 10 bins for clustering. 15 angular bins.

$$N_d = N_\theta \, N_{bin} \, (N_{bin}+1) / 2 = 3150$$

Compression factor: $\sim 3 \times 10^6$ → pretty good!

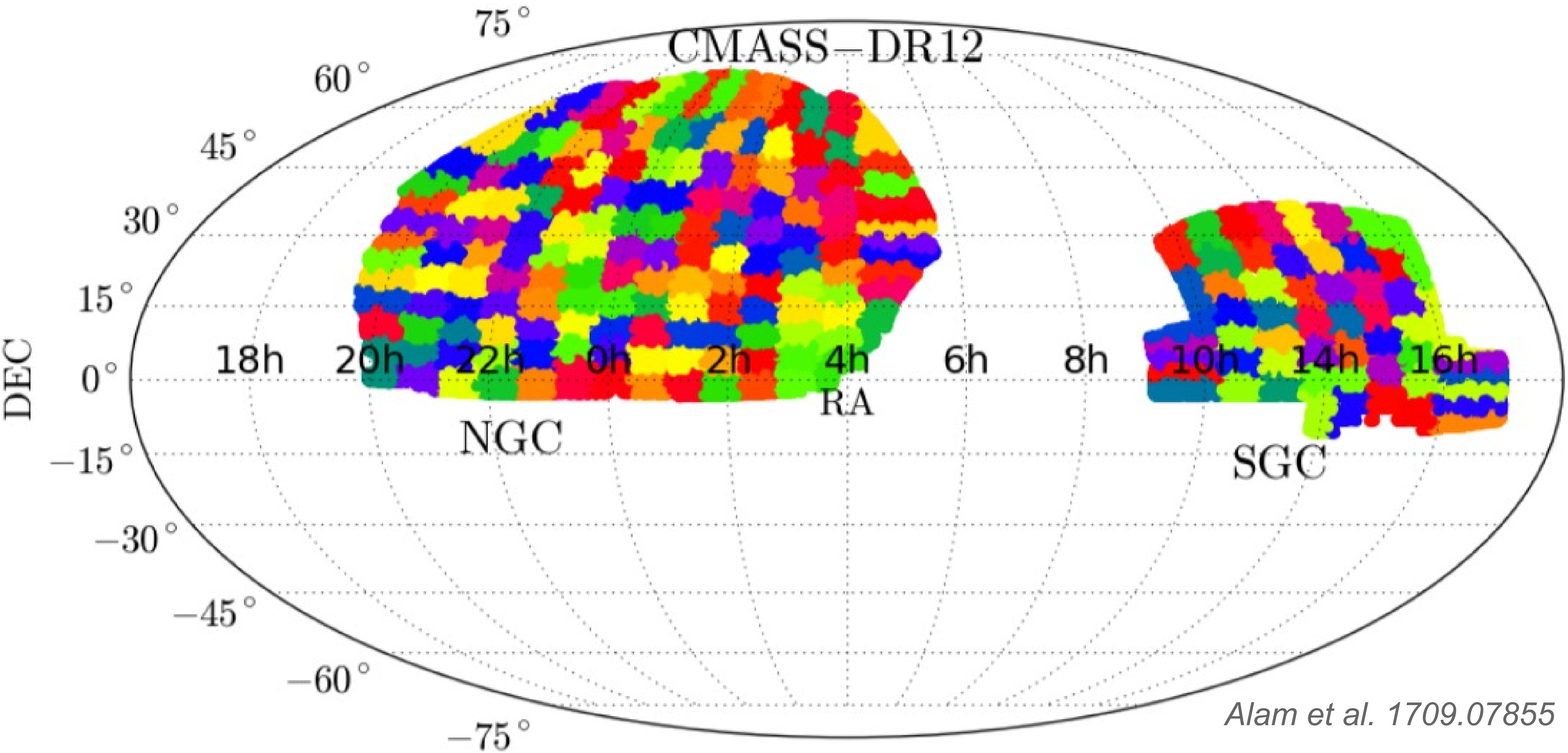
Achieved by:
- Selecting only the most informative summary statistic.
- Averaging over equivalent modes (e.g. using statistical isotropy).

However, now we need to compute the data **covariance matrix**.

# Computing the covariance matrix

Different methods:

- Jackknife/bootstrap: use sub-samples of your own data.

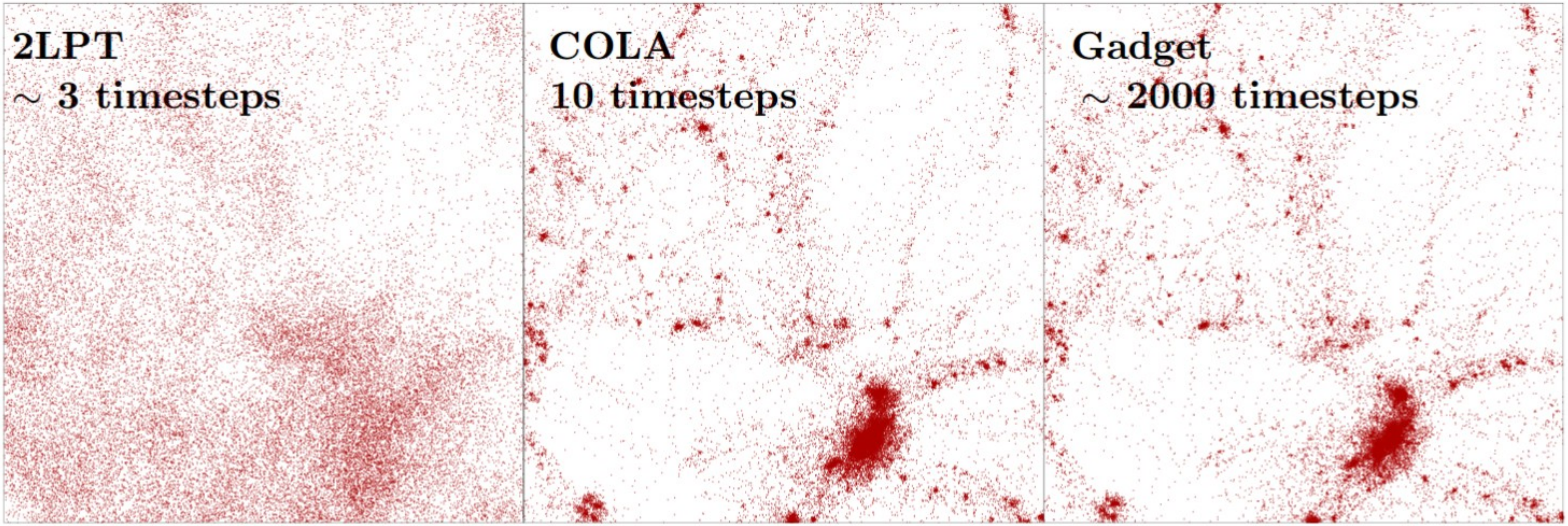



*Alam et al. 1709.07855*

# Computing the covariance matrix

Different methods:

- Jackknife/bootstrap: use sub-samples of your own data.
- Mock catalogues: based on N-body sims or fast methods (Gaussian, FLASK, 2LPT, COLA, PINOCHIO, PTHALOS, QuickPM …)




*Tassev et al. 1301.0322*

For both of these, rule of thumb is $N_{samples}$ > 10 x (size of data vector).
Then, $O(3 \times 10^4)$ mocks/JKs are needed (covering the same volume as LSST).

# Computing the covariance matrix

Different methods:

- Jackknife/bootstrap: use sub-samples of your own data.
- Mock catalogues: based on N-body sims or fast methods (Gaussian, lognormal, 2LPT, COLA, PINOCHIO, PTHALOS, QuickPM …)
- Analytical covariance matrix:
  Gaussian connected part:

*Krause & Eifler 1601.05779*

$$\mathrm{Cov}^{\mathrm{G}}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \frac{4\pi\delta_{l_1 l_2}}{\Omega_s(2l_1+1)\Delta l_1}\left[\left(C^{ik}_{AC}(l_1)+\delta_{ik}\delta_{AC}N^i_A\right)\left(C^{jl}_{BD}(l_2)+\delta_{jl}\delta_{BD}N^j_B\right)+\left(C^{il}_{AD}(l_1)+\delta_{il}\delta_{AD}N^i_A\right)\left(C^{jk}_{BC}(l_2)+\delta_{jk}\delta_{BC}N^j_B\right)\right]$$

SSC

$$\mathrm{Cov}^{\mathrm{SSC}}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \int \mathrm{d}\chi \, \frac{q^i_A(\chi)q^j_B(\chi)q^k_C(\chi)q^l_D(\chi)}{\chi^4} \, \frac{\partial P_{AB}(l_1/\chi, z(\chi))}{\partial \delta_b} \frac{\partial P_{CD}(l_2/\chi, z(\chi))}{\partial \delta_b} \sigma_b(\Omega_s; z(\chi))$$

Relevant connected parts

$$\mathrm{Cov}^{\mathrm{NG},0}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \frac{1}{\Omega_s}\int_{|\mathbf{l}|\in l_1}\frac{d^2\mathbf{l}}{A(l_1)}\int_{|\mathbf{l}'|\in l_2}\frac{d^2\mathbf{l}'}{A(l_2)}\int \mathrm{d}\chi \, \frac{q^i_A(\chi)q^j_B(\chi)q^k_C(\chi)q^l_D(\chi)}{\chi^6} \, T^{ijkl}_{ABCD}(\mathbf{l}/\chi, -\mathbf{l}/\chi, \mathbf{l}'/\chi, -\mathbf{l}'/\chi; z(\chi))$$

+ double Hankel transform if you work in real space
+ probably worry about survey geometry (mode coupling)

$$\left\langle \Delta\tilde{C}^{ab}_{\ell}\Delta\tilde{C}^{cd}_{\ell'}\right\rangle = \sum_{mm'}\sum_{\mathbf{l}_1\mathbf{l}_2}\left(C^{ac}_{\ell_1}C^{bd}_{\ell_2}W^a_{\mathbf{l}\mathbf{l}_1}W^b_{\mathbf{l}\mathbf{l}_2}W^c_{\mathbf{l}\mathbf{l}_1}W^d_{\mathbf{l}\mathbf{l}_2} + C^{ad}_{\ell_1}C^{bc}_{\ell_2}W^a_{\mathbf{l}\mathbf{l}_1}W^b_{\mathbf{l}\mathbf{l}_2}W^c_{\mathbf{l}\mathbf{l}_2}W^d_{\mathbf{l}\mathbf{l}_1}\right)$$

Computation scales very bad: $O(N_\theta^2 \, N_{bin}^4)$

# Computing the covariance matrix

Different methods:

- Jackknife/bootstrap: use sub-samples of your own data.
- Mock catalogues: based on N-body sims or fast methods (Gaussian, lognormal, 2LPT, COLA, PINOCHIO, PTHALOS, QuickPM …)
- Analytical covariance matrix:
  Gaussian connected part:

*Krause & Eifler 1601.05779*

$$\mathrm{Cov}^{\mathrm{G}}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \frac{4\pi\delta_{l_1 l_2}}{\Omega_s(2l_1+1)\Delta l_1}\left[\left(C^{ik}_{AC}(l_1)+\delta_{ik}\delta_{AC}N^i_A\right)\left(C^{jl}_{BD}(l_2)+\delta_{jl}\delta_{BD}N^j_B\right)+\left(C^{il}_{AD}(l_1)+\delta_{il}\delta_{AD}N^i_A\right)\left(C^{jk}_{BC}(l_2)+\delta_{jk}\delta_{BC}N^j_B\right)\right]$$

SSC

$$\mathrm{Cov}^{\mathrm{SSC}}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \int \mathrm{d}\chi\, \frac{q^i_A(\chi)q^j_B(\chi)q^k_C(\chi)q^l_D(\chi)}{\chi^4}\, \frac{\partial P_{AB}(l_1/\chi, z(\chi))}{\partial\delta_b}\frac{\partial P_{CD}(l_2/\chi, z(\chi))}{\partial\delta_b}\sigma_b(\Omega_s; z(\chi))$$

Relevant connected parts

$$\mathrm{Cov}^{\mathrm{NG},0}\left(C^{ij}_{AB}(l_1), C^{kl}_{CD}(l_2)\right) = \frac{1}{\Omega_s}\int_{|\mathbf{l}|\in l_1}\frac{d^2\mathbf{l}}{A(l_1)}\int_{|\mathbf{l}'|\in l_2}\frac{d^2\mathbf{l}'}{A(l_2)}\int \mathrm{d}\chi\, \frac{q^i_A(\chi)q^j_B(\chi)q^k_C(\chi)q^l_D(\chi)}{\chi^6}\; T^{ijkl}_{ABCD}(\mathbf{l}/\chi, -\mathbf{l}/\chi, \mathbf{l}'/\chi, -\mathbf{l}'/\chi; z(\chi))$$

+ double Hankel transform if you work in real space
+ probably worry about survey geometry (mode coupling)

NaMaster can do this.
Stay tuned!

$$\left\langle \Delta\tilde{C}^{ab}_{\ell}\Delta\tilde{C}^{cd}_{\ell'}\right\rangle = \sum_{mm'}\sum_{\mathbf{l}_1\mathbf{l}_2}\left(C^{ac}_{\ell_1}C^{bd}_{\ell_2}W^a_{\mathbf{l}\mathbf{l}_1}W^b_{\mathbf{l}\mathbf{l}_2}W^c_{\mathbf{l}\mathbf{l}_1}W^d_{\mathbf{l}\mathbf{l}_2} + C^{ad}_{\ell_1}C^{bc}_{\ell_2}W^a_{\mathbf{l}\mathbf{l}_1}W^b_{\mathbf{l}\mathbf{l}_2}W^c_{\mathbf{l}\mathbf{l}_2}W^d_{\mathbf{l}\mathbf{l}_1}\right)$$

Computation scales very bad: $O(N_\theta^2\, N_{bin}^4)$

# Computing the covariance matrix

Different methods:

- Jackknife/bootstrap: use sub-samples of your own data.
- Mock catalogues: based on N-body sims or fast methods (Gaussian, lognormal, 2LPT, COLA, PINOCHIO, PTHALOS, QuickPM …)
- Analytical covariance matrix:

**All of these cases would benefit massively from reducing the size of the data vector.**

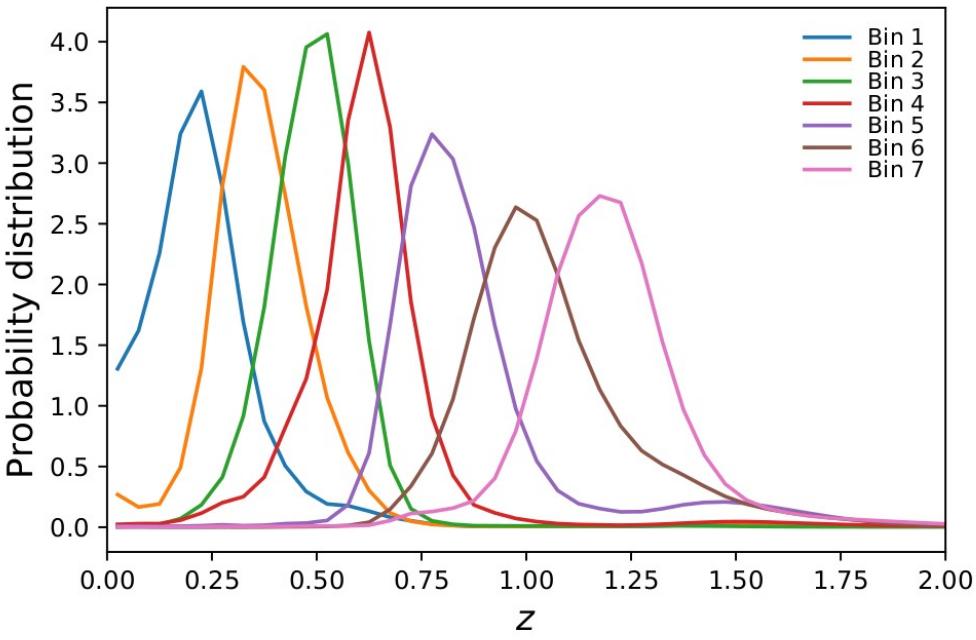
**<u>Can we compress further?</u>**

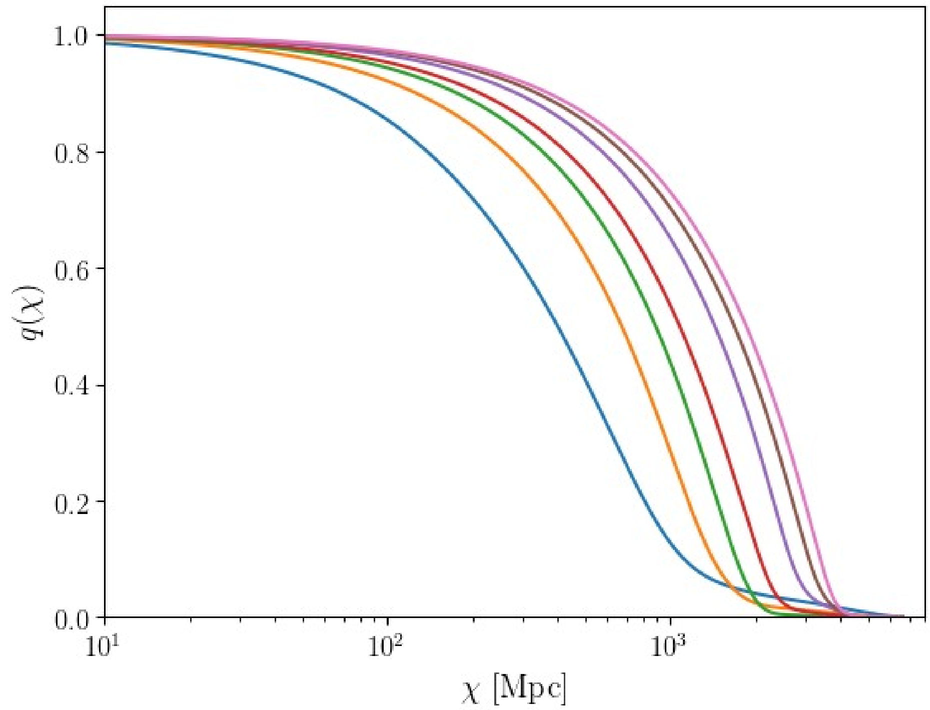
# Data compression

**Example:** cosmic shear

$$C^{ij}_{AB}(l) = \int d\chi \frac{q^i_A(\chi) q^j_B(\chi)}{\chi^2} P_{AB}(l/\chi, z(\chi))$$

$$q^i_\kappa(\chi) = \frac{3H_0^2\Omega_m}{2c^2} \frac{\chi}{a(\chi)} \int_\chi^{\chi_h} d\chi' \frac{n^i_{\rm source}(z(\chi'))dz/d\chi'}{\bar{n}^i_{\rm source}} \frac{\chi' - \chi}{\chi'}$$



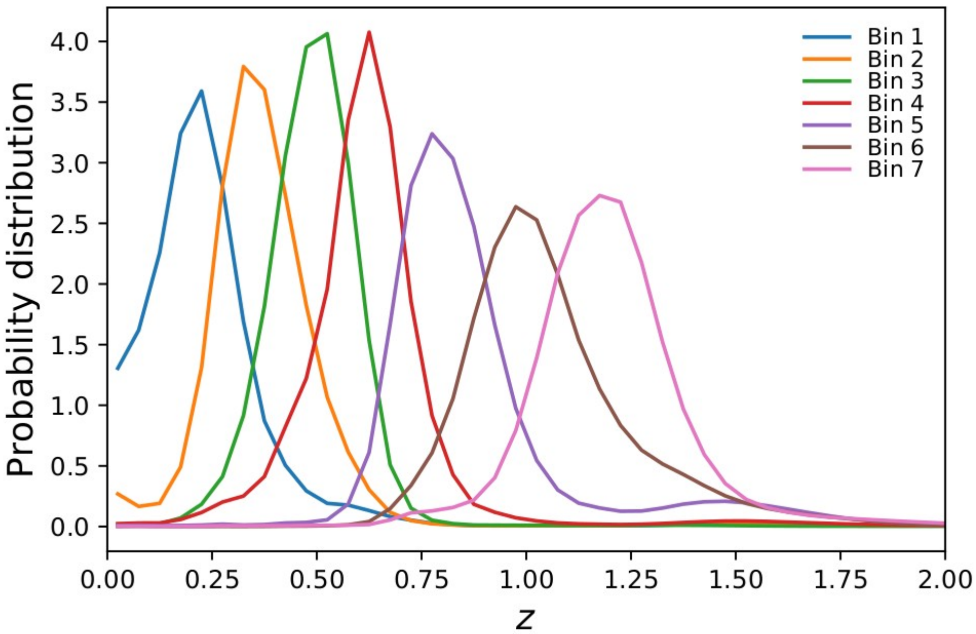



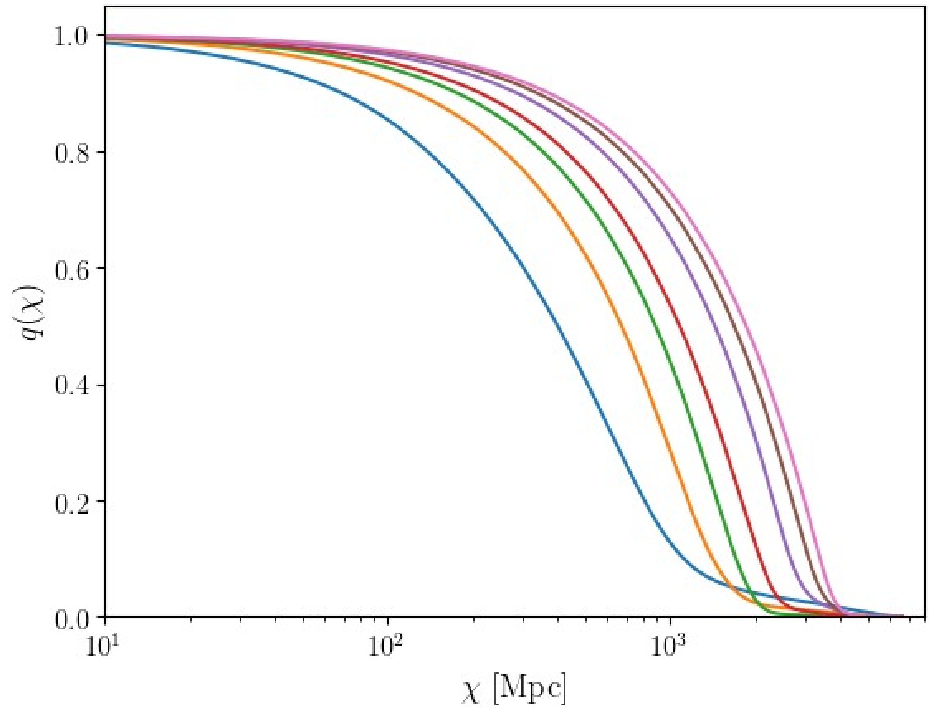
# Data compression

**Example:** cosmic shear

$$C^{ij}_{AB}(l) = \int d\chi \frac{q^i_A(\chi) q^j_B(\chi)}{\chi^2} P_{AB}(l/\chi, z(\chi))$$

$$q^i_\kappa(\chi) = \frac{3H_0^2\Omega_m}{2c^2} \frac{\chi}{a(\chi)} \int_\chi^{\chi_h} d\chi' \frac{n^i_{\rm source}(z(\chi'))dz/d\chi'}{\bar{n}^i_{\rm source}} \frac{\chi' - \chi}{\chi'}$$




Different bins are very correlated.
Correlation → you have fewer *d.o.f.*s than you think.
You **can** compress further!

# The Karhunen-Loeve transform

**Idea:** find the linear combinations of your data that contain most of the information about a given parameter θ.

$$y_p \equiv \mathbf{e}_p^\dagger \, \mathbf{x}$$

**Data:** **x** → maps/$a_{\ell m}$s of a given set of tomographic observables (e.g. galaxy overdensity or shear in a set of redshift bins).

The linear coefficients **e** can be found as the eigenvectors of a generalized eigenvalue equation:

$$\partial_\theta \mathsf{C}\, \mathbf{e}_p = \lambda_p \mathsf{C}\, \mathbf{e}_p$$

Covariance of **x**

One generic parameter we could optimize for is the overall S/N amplitude. Maximizing this should provide us with most of the information about any parameter in most cases.

In this case, the eigenvalue equation reads:

Noise covariance

Signal covariance

$$(\mathsf{S} + \mathsf{N})\mathbf{e}_p = \lambda_p \mathsf{N}\mathbf{e}_p$$

Resulting modes $y_p$ are uncorrelated and contain the maximum amount of information (info($y_0$) > info($y_1$) > …).

# The Karhunen-Loeve transform

**Example:** galaxy clustering with spectroscopic redshifts.

**x** $\rightarrow$ galaxy overdensity in an infinitesimal redshift bin.

C $\rightarrow$ all possible cross-power spectra between bins (noise + signal)
N $\rightarrow$ flat, diagonal shot-noise power spectrum

The solution to the generalized eigenvalue equation (KL modes) is

$$e_{k,\ell}(z) \propto j_\ell\left(k\chi(z)\right)$$

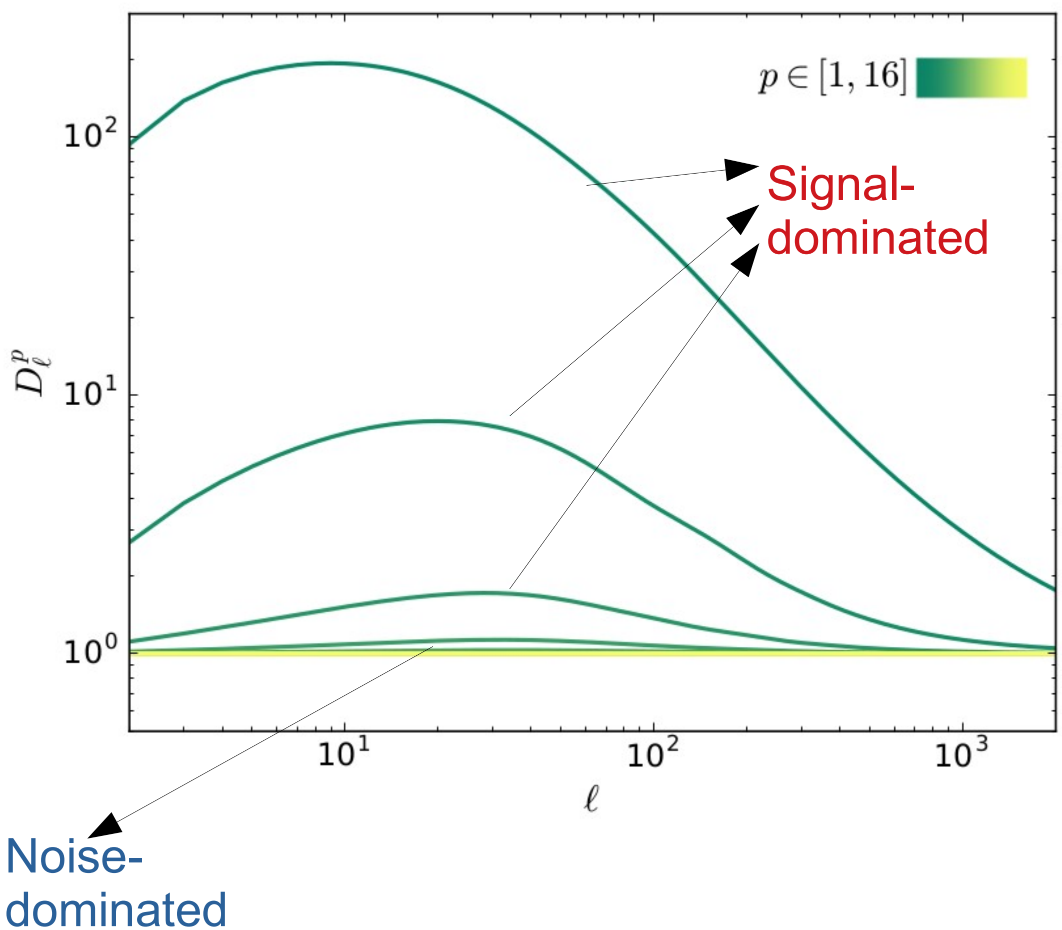
i.e. KL transform in this case is the harmonic-Bessel transform.

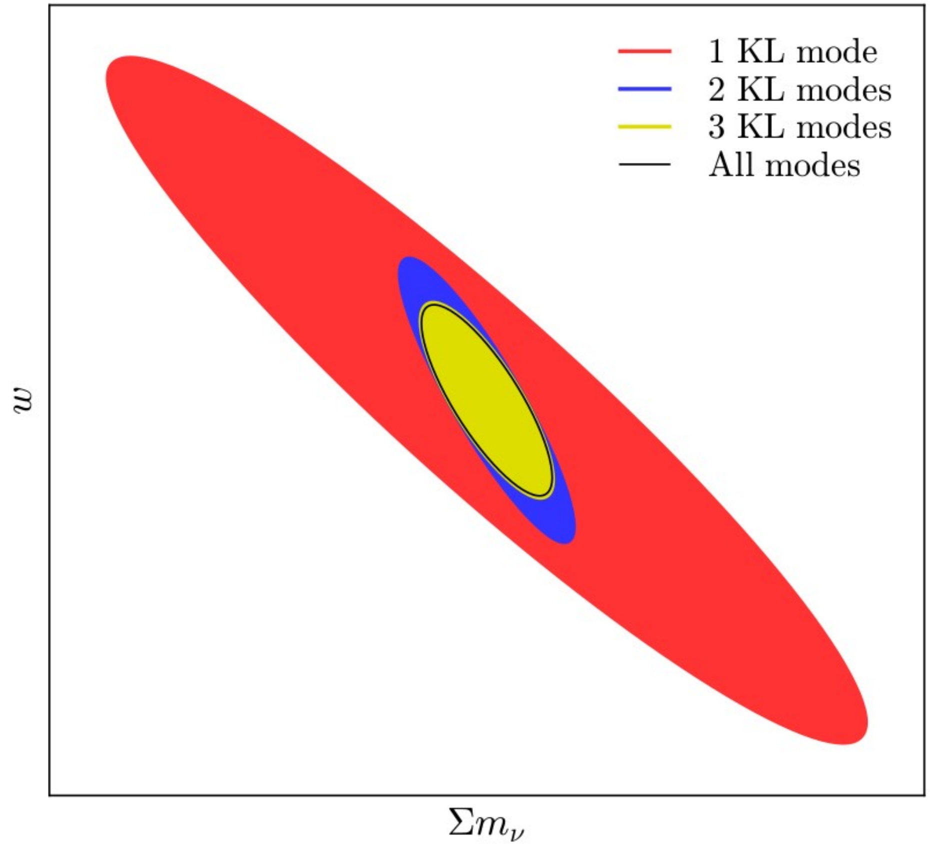
The covariance of the resulting KL modes is

$$\lambda_{k,\ell} \propto P(k)$$

i.e. in this case the KL transform tells you to just compute the Fourier transform and estimate the 3D power spectrum (as expected!).

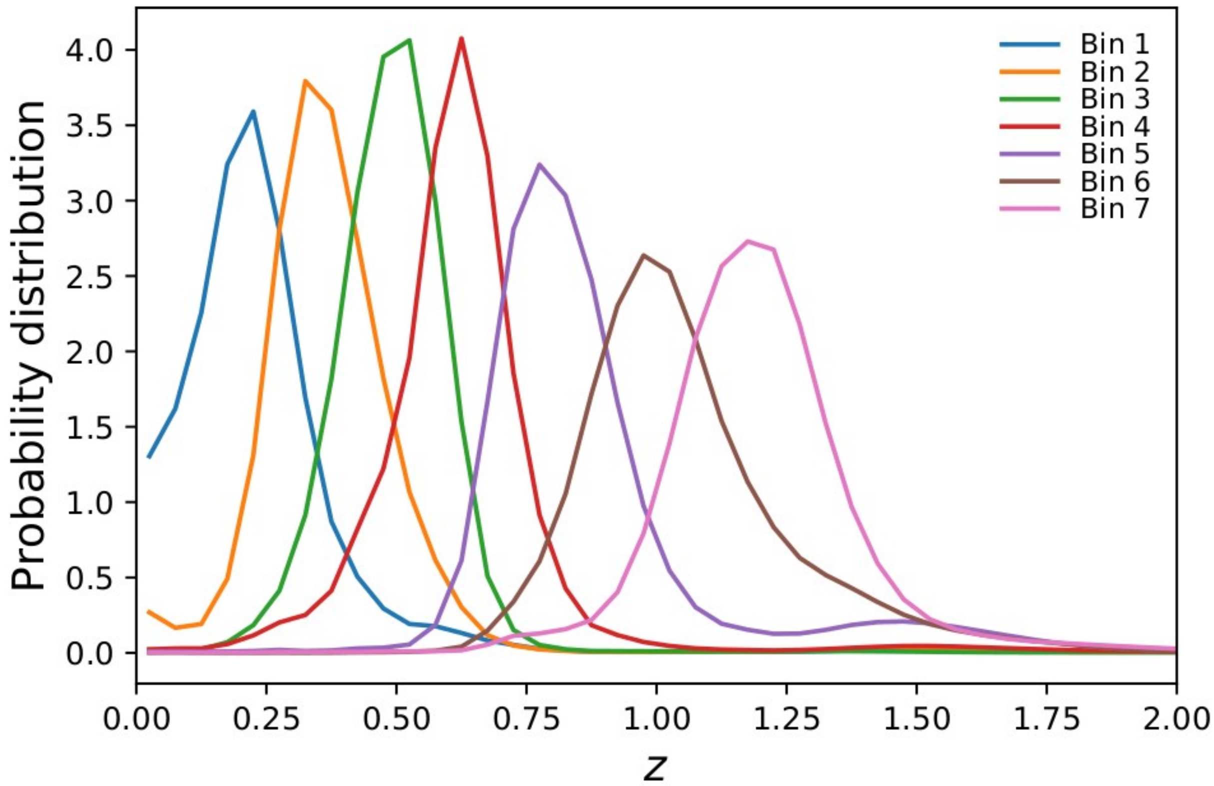
# The KL transform: cosmic shear



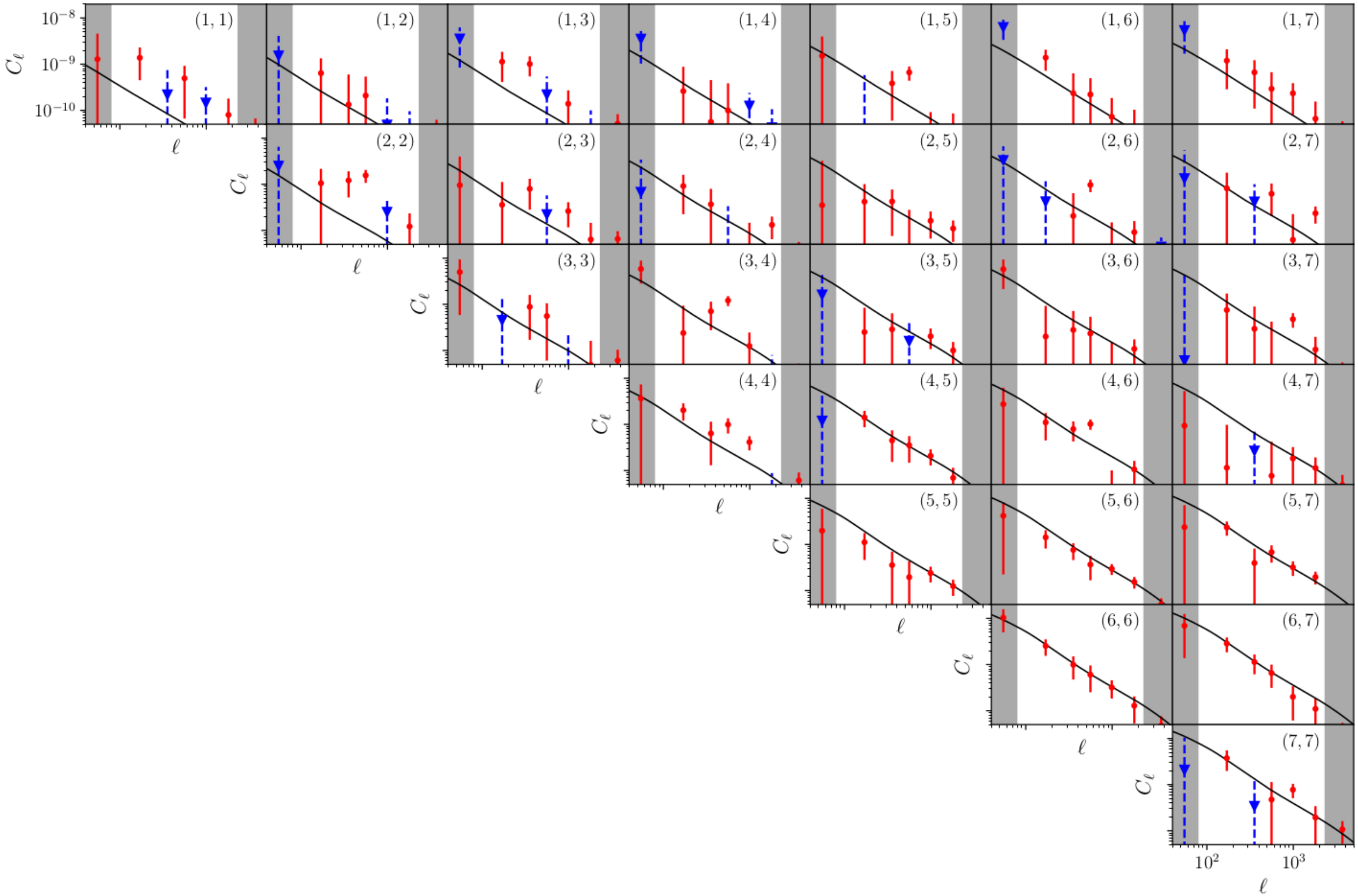

- Idealized LSST-like survey ($n_{gal}$ = 27 arcmin-2)
- First three modes contain all of the signal
- They are also able to recover the full constraining power.
- Formally speaking, this is the P(k) equivalent of tomographic shear analyses.

# The KL transform: application to CFHTLens



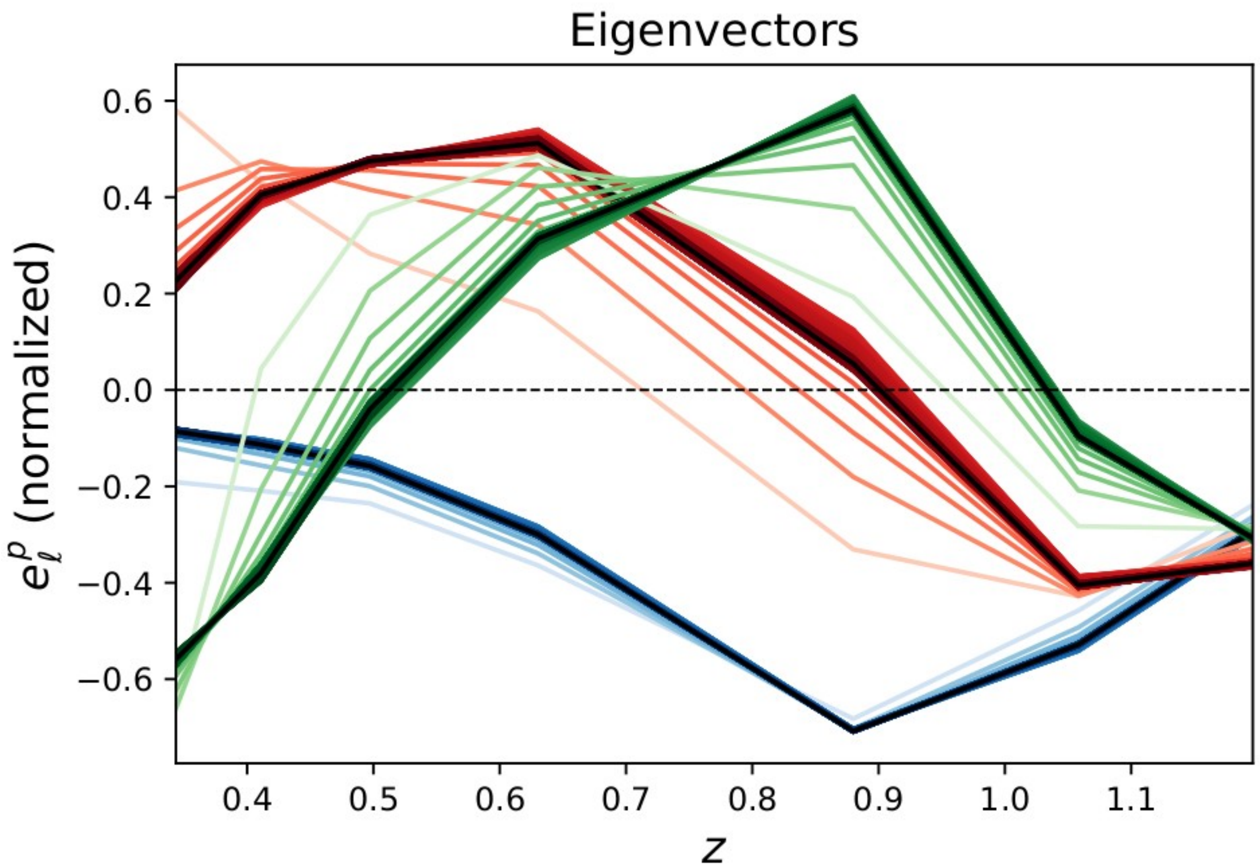

- Latest analysis (Joudaki et al. 2016) uses 7 tomographic bins in real space.

# The KL transform: application to CFHTLens



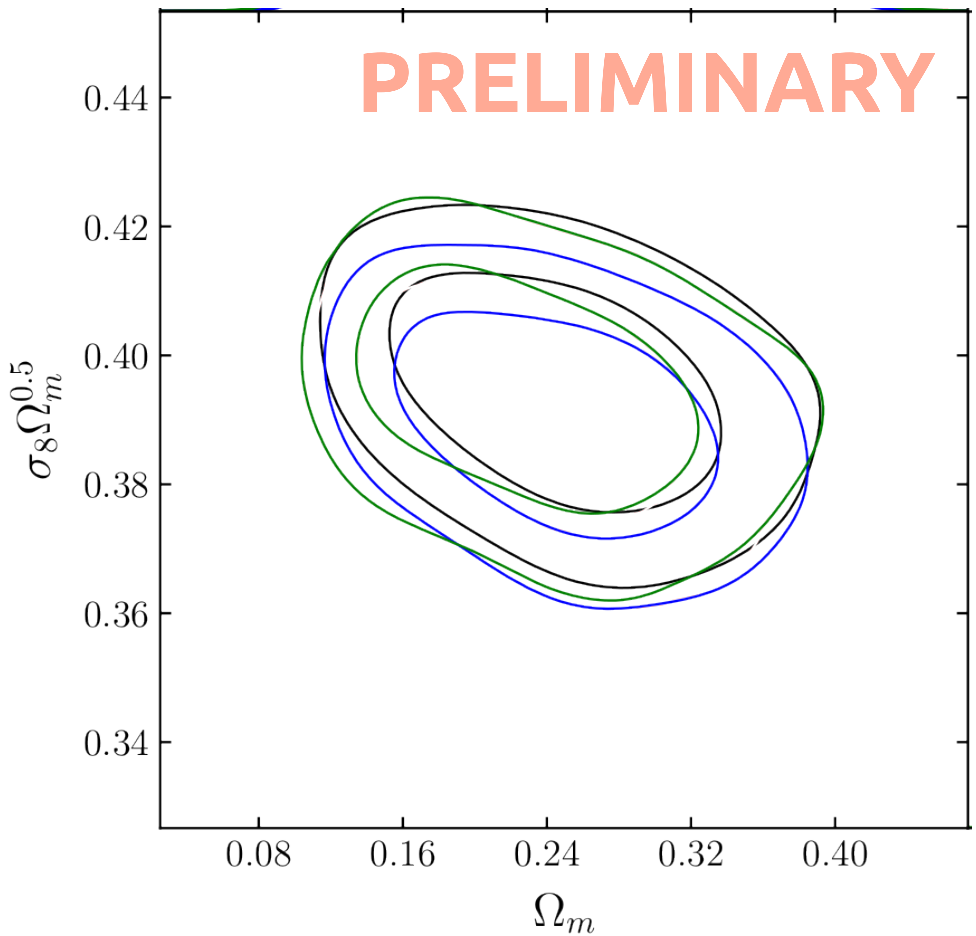

- Latest analysis (Joudaki et al. 2016) uses 7 tomographic bins in real space.
- Size of data vector: 2 x 5 x (7x8)/2 = 280 elements

# The KL transform: application to CFHTLens




- Latest analysis (Joudaki et al. 2016) uses 7 tomographic bins in real space.
- Size of data vector: 2 x 5 x (7x8)/2 = 280 elements.
- Eigenvectors close to scale-independent.
  Think of them as redshift-dependent galaxy weights.

# The KL transform: application to CFHTLens




**Black** → Fiducial constraints
**Blue** → 2 principal KL modes
**Green** → 2 modes, only auto-corr

- Latest analysis (Joudaki et al. 2016) uses 7 tomographic bins in real space.
- Size of data vector: 2 x 5 x (7x8)/2 = 280 elements.
- Eigenvectors close to scale-independent.
  Think of them as redshift-dependent galaxy weights.
- The first 2 modes are able to recover the full constraining power.
  **Compression factor ~19-30**!

# Data compression

**Other uses of the KL transform:**

- Large-scale effects: optimize fNL constraints.
- Systematics: remove modes that are most sensitive to e.g. intrinsic alignments, magnification …
  (basically put everything you don't like in the noise component)
- Foreground removal in 21cm experiments

**Extreme data compression:**

- *Alsing & Wandelt 1712.00012, Alsing et al. 1801.01497.*
- One summary statistic per free parameter.
- Can be made robust to systematics.
- Potentially more sensitive to modeling errors. Missing systematics may be more difficult to detect (KL at least gives you maps to inspect).

# Robust theory predictions

*Core Cosmology Library: precision cosmological predictions for LSST*

Chisari E., DA, E. Krause +27,
arXiv:1812.05995

# Robust theory predictions

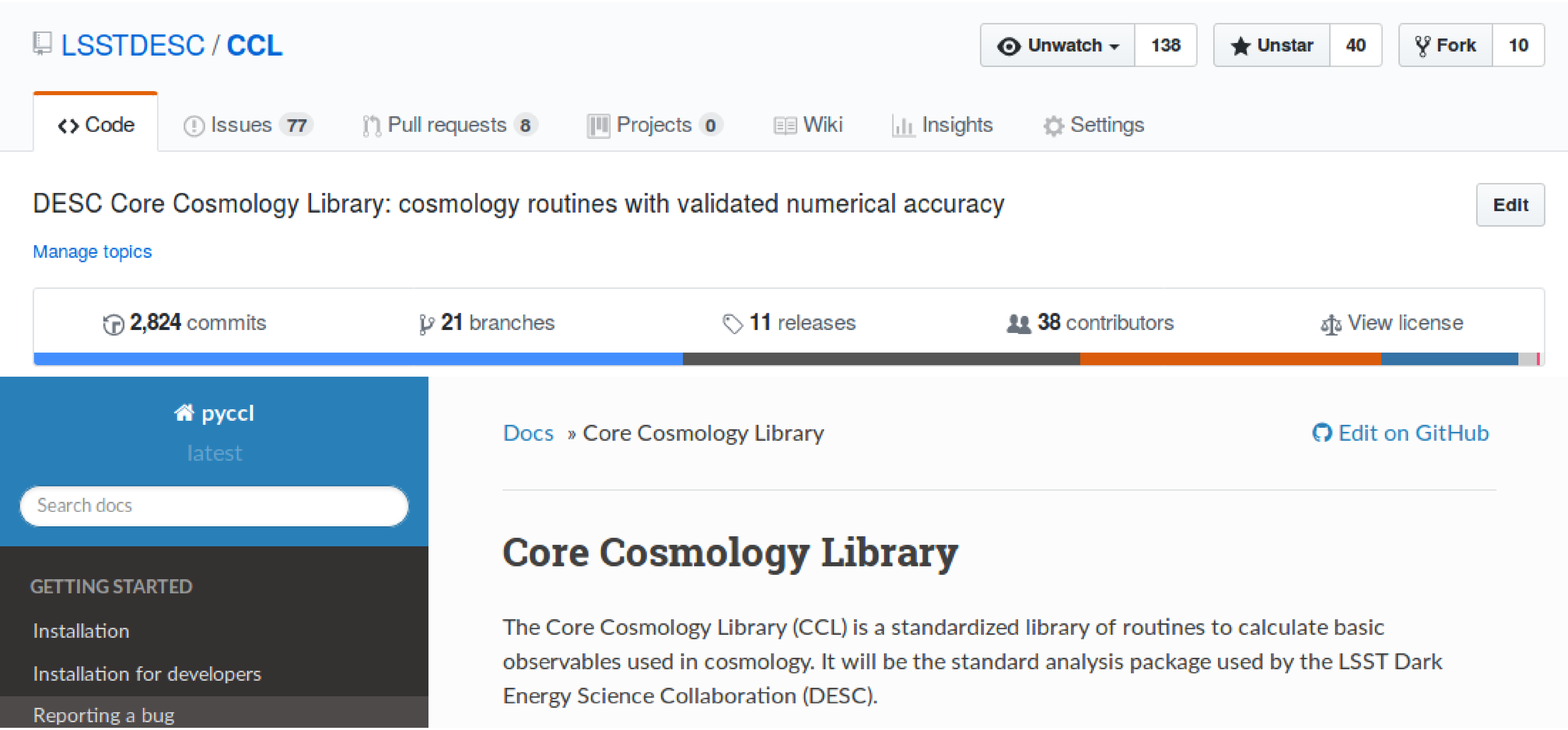
$$-2 \log P(d|\theta) = (d-t(\theta))^T C^{-1} (d-t(\theta)) + L_0$$

Having accurate models for $t(\theta)$ is vital.
The accuracy must be significantly higher than the statistical power.
LSST's statistical power will be awesome.

**Requirements for LSST:**

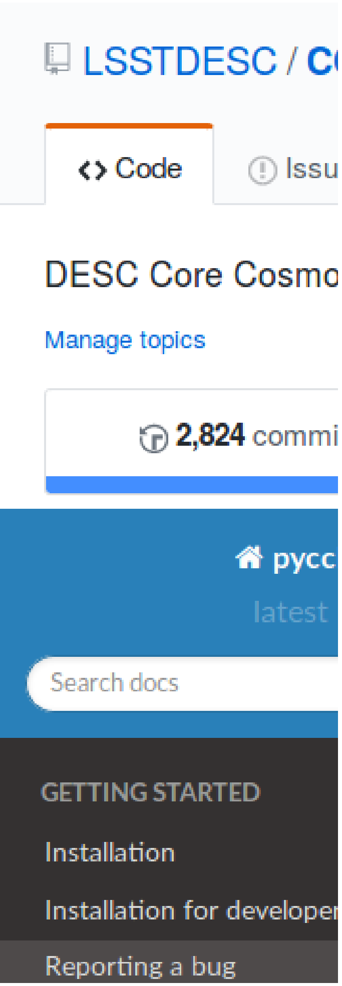
- Accuracy (errors well below statistical uncertainties)
- Robustness (thorough code validation and comparison)
- Flexibility (many observables, many cosmological models, ability to vary models and absorb systematics)
- Numerical performance (reasonable MCMC-ing time)

# The Core Cosmology Library




**Code:** https://github.com/LSSTDESC/CCL
**Docs:** https://ccl.readthedocs.io/en/latest/
**Latest release:** https://github.com/LSSTDESC/CCL/releases/tag/v1.0.0

# The Core Cosmology Library




## 3x2 correlations with CCL

With two samples of galaxies, there are three types of correlations we can perform. We could measure and model the auto-correlations of galaxy positions ("clustering"), the auto-correlations of galaxy shapes ("cosmic shear") and the cross-correlation between positions and shapes. Normally, to gain information on the expansion history of the Universe, we would split the sample into different redshift bins with a sufficient number of galaxies to have a significant measurement in each one. But for now, let's just take the full redshift distribution.

Correlations are expressed in different forms. Here, we are going to express them in terms of "angular power spectra", or C_ell. Imagine that we took the sphere of the sky and expanded any function of the coordinates of the sphere into a basis of spherical harmonics. Each harmonic would contribute to the expansion with a given amplitude. What we are going to plot is the square of that amplitude as a function of multipole index.

### Galaxy positions

To model galaxy positions we need to define a "bias" parameter. This parameter tells us how the galaxies are connected to the density field. To make it simple, we'll take a one-to-one relation. Galaxies are simply tracing the density field in this model:

In [9]:
```
bias_gal = np.ones(z.size)
```

We know need to define a convenience function, called a "tracer" which will store this information.

In [10]:
```
gal_pos = ccl.NumberCountsTracer(cosmo_fid, has_rsd=False, dndz=(z, dNdz_pos), bias=(z, bias_gal)
)
```

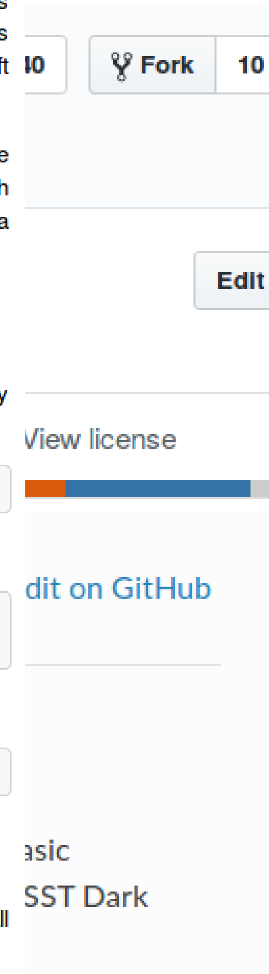
The "False" statements above control the modeling of potential contributions to the signal, like redshift-space distortions.

In [11]:
```
gal_shapes = ccl.WeakLensingTracer(cosmo_fid, dndz=(z, dNdz_shape))
```

### Angular power spectra

We are now ready to compute angular power spectra, C_ell. These are a function of multipole number, with high ell correponding to small scales on the sky and low ell, to large separations on the sky.

In [12]:
```
ell=np.arange(100,5000)
cls_auto_pos = ccl.angular_cl(cosmo_fid, gal_pos, gal_pos, ell)
cls_auto_shape = ccl.angular_cl(cosmo_fid, gal_shapes, gal_shapes, ell)
cls_pos_shape = ccl.angular_cl(cosmo_fid, gal_pos, gal_shapes, ell)
```

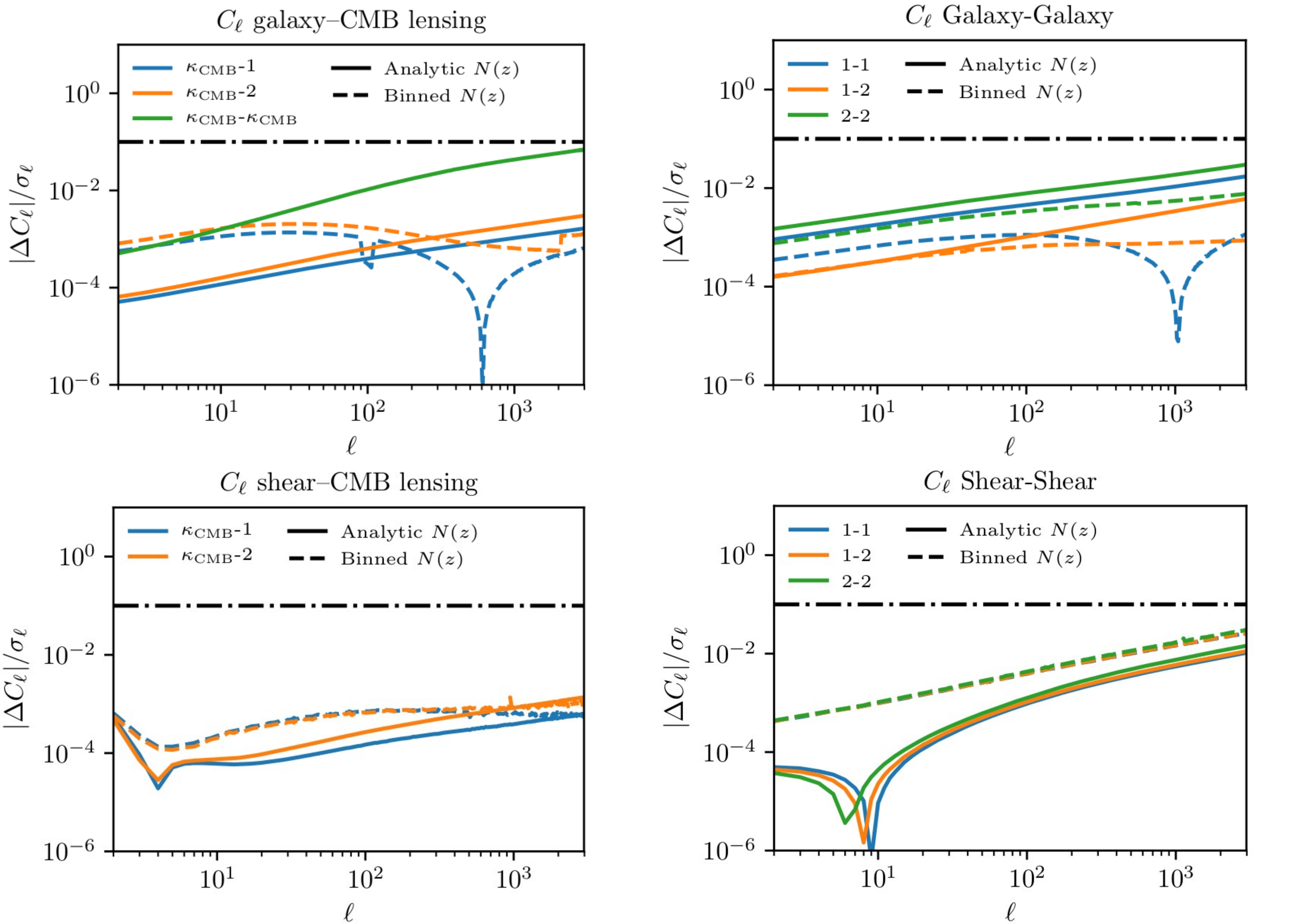



**Code:** https://github.com/LSSTDESC/CCL
**Docs:** https://ccl.readthedocs.io/en/latest/
**Latest release:** https://github.com/LSSTDESC/CCL/releases/tag/v1.0.0

## Strict code validation requirements

- All calculations are performed with at least one different independent code.
- Agreement must be found within well-motivated/crazy stringent requirements.
- Alternative calculations are kept as benchmarks.
- CCL is automatically compared against benchmarks whenever a new addition is made to the code.
- Unit tests for other types of functionality (error passing etc.) are also in place.

# Code validation



# Code validation

**Currently implemented:**

- Background quantities and linear growth.
- Matter power spectrum
  Links to CLASS, CosmicEmu, fast approximations (E&H, BBKS).
- Halo quantities:
  Mass function
  Bias
  Concentrations
  Profiles
  Halo model power spectra
- Angular power spectra
  Galaxy clustering, cosmic shear, CMB lensing
- Angular correlations functions
- 3D correlation functions

**Latest release:** https://github.com/LSSTDESC/CCL/releases/tag/v1.0.0

# Code validation

**Currently implemented:**

- Background quantities and linear growth.
- Matter power spectrum
  - Links to CLASS, CosmicEmu, fast approximations (E&H, BBKS).
- Halo quantities:
  - Mass function
  - Bias
  - Concentrations
  - Profiles
  - Halo model power spectra
- Angular power spectra
  - Galaxy clustering, cosmic shear, CMB lensing
- Angular correlations functions
- 3D correlation functions

**Ongoing/future work:**

- Flexibility: generalized input power spectra and radial kernels.
- Other observables: CMB observables (SZ, ISW), generalized halo models.
- Speed optimization.
- Integration into downstream pipelines.

**Latest release:** https://github.com/LSSTDESC/CCL/releases/tag/v1.0.0

*The effect on cosmological parameter estimation of a parameter-dependent covariance matrix*

Kodwani D., DA, P. Ferreira

arXiv:1811.11584

# Theory predictions and covariance matrices

$$-2 \log P(d|\theta) = (d-t(\theta))^T C^{-1} (d-t(\theta)) + L_0$$

# Theory predictions and covariance matrices

$$-2 \log P(d|\theta) = (d-t(\theta))^T C^{-1}(\theta) (d-t(\theta)) + L_0 \text{ ?}$$
$$-2 \log P(d|\theta) = (d-t(\theta))^T C_{fid}^{-1} (d-t(\theta)) + L_0 \text{ ?}$$

- Do we have to take into account the parameter dependence of the covariance matrix?
- I.e. do we need to compute a new covariance at every point in an MCMC chain?

# Theory predictions and covariance matrices

$$-2 \log P(d|\theta) = (d-t(\theta))^T \, C^{-1}(\theta) \, (d-t(\theta)) + L_0 \; ?$$

$$-2 \log P(d|\theta) = (d-t(\theta))^T \, C_{fid}^{-1} \, (d-t(\theta)) + L_0 \; ?$$

- Do we have to take into account the parameter dependence of the covariance matrix?
- I.e. do we need to compute a new covariance at every point in an MCMC chain?
- Carron 2016: for Gaussian fields it's not only unnecessary, it's incorrect.
- The galaxy overdensity and cosmic shear aren't Gaussian, so do we need to worry about this at all?

# The math

The information content of the covariance matrix can be quantified approximating the likelihood as Gaussian around the maximum (i.e. a la Fisher).

- Effect on parameter uncertainties:

$$\mathcal{F}_{\mu\nu} = \partial_\mu \mathbf{t}^T \Sigma^{-1} \partial_\nu \mathbf{t} + \frac{1}{2} \mathrm{Tr}\left(\Sigma^{-1} \partial_\mu \Sigma \Sigma^{-1} \partial_\nu \Sigma\right)$$

- Effect on parameter bias:

$$\Delta\theta_\mu = -\frac{1}{2} \mathcal{F}^{-1}_{\mu\nu} \, \mathcal{F}^{-1}_{\rho\tau} \, \partial_\rho \mathbf{t}^T \Sigma^{-1} \partial_\nu \Sigma \, \Sigma^{-1} \partial_\tau \mathbf{t}$$

# The math

The information content of the covariance matrix can be quantified approximating the likelihood as Gaussian around the maximum (i.e. a la Fisher).

- Effect on parameter uncertainties:

$$\mathcal{F}_{\mu\nu} = \partial_\mu \mathbf{t}^T \Sigma^{-1} \partial_\nu \mathbf{t} + \frac{1}{2} \mathrm{Tr}\left(\Sigma^{-1} \partial_\mu \Sigma \Sigma^{-1} \partial_\nu \Sigma\right)$$

- Effect on parameter bias:

$$\Delta\theta_\mu = -\frac{1}{2} \mathcal{F}^{-1}_{\mu\nu} \, \mathcal{F}^{-1}_{\rho\tau} \, \partial_\rho \mathbf{t}^T \Sigma^{-1} \partial_\nu \Sigma \, \Sigma^{-1} \partial_\tau \mathbf{t}$$

Let's examine the dependence on $f_{sky}$.

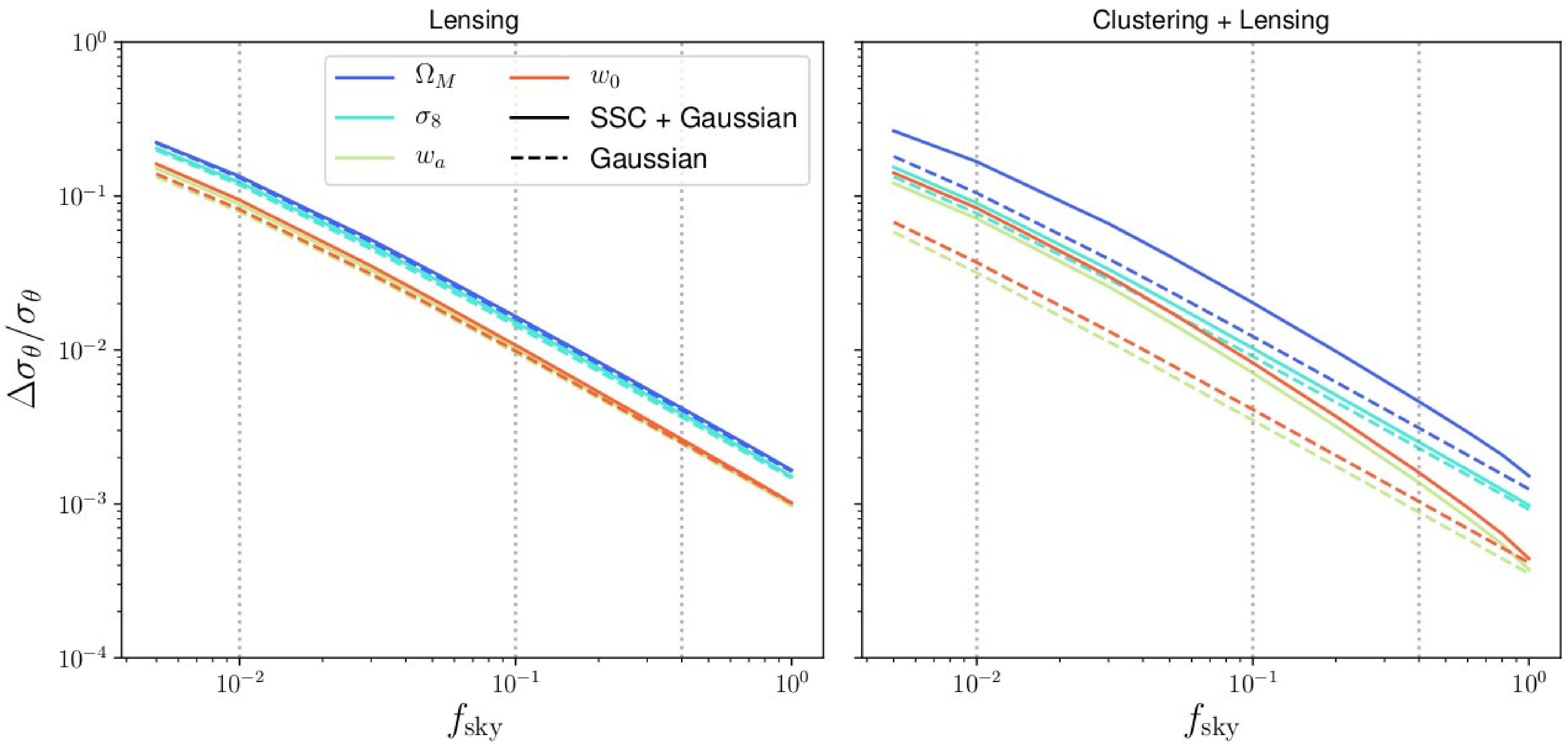
Roughly: $\Sigma \propto f_{\mathrm{sky}}^{-1}$

Then: $\Delta\theta \propto f_{\mathrm{sky}}^{-1} \quad , \quad \delta\sigma(\theta) \propto f_{\mathrm{sky}}^{-3/2}$

In general, the effects of a parameter-dependent covariance shrink with the number of modes in the analysis (same also with $\ell_{max}$).

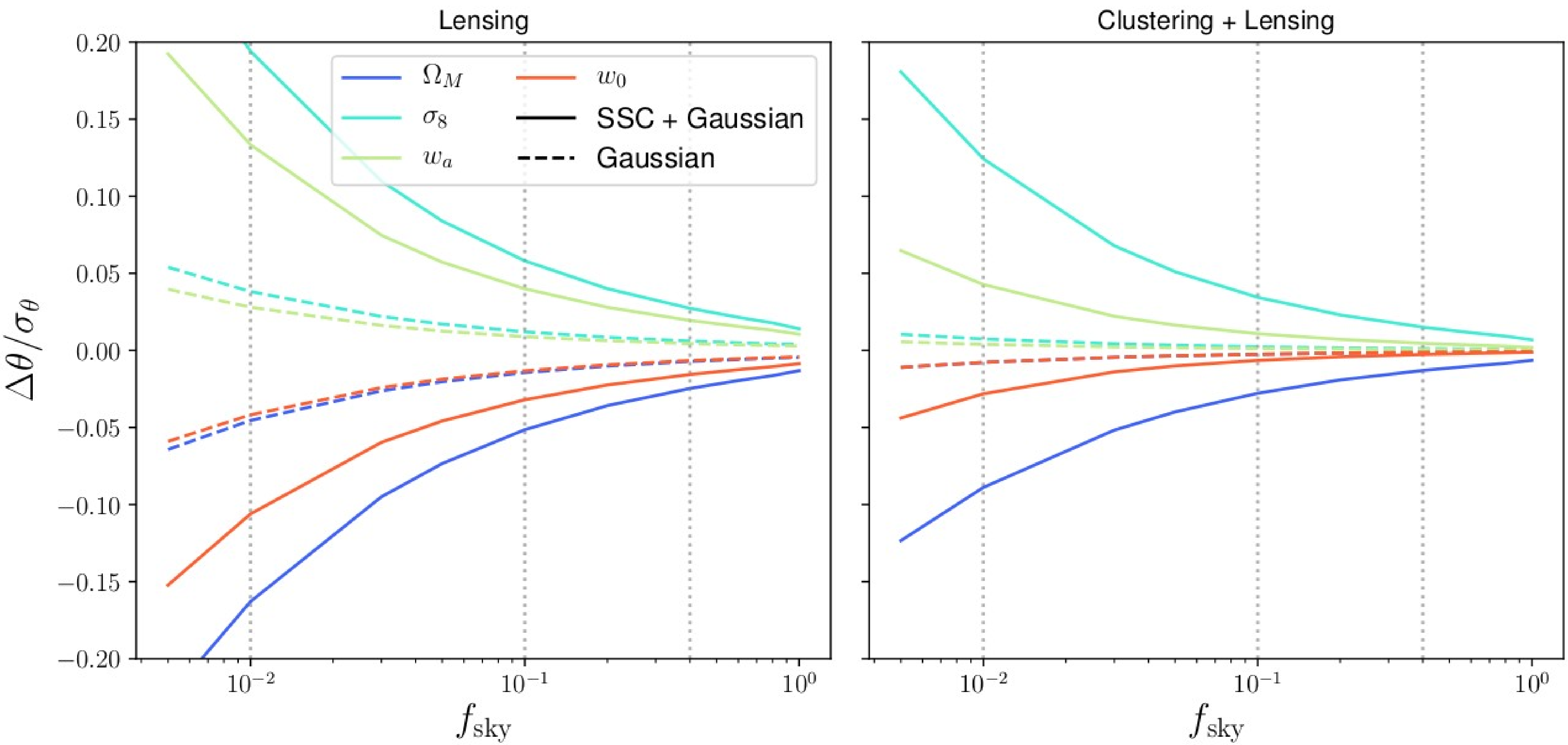
# Parameter-dependent covariances

**Results: parameter uncertainties**



# Parameter-dependent covariances

## Results: parameter uncertainties




The parameter dependence of the covariance is irrelevant in all cases.

# Summary

- **Two computational tools** for future large-scale structure experiments developed by the LSST DESC:
  - Compute power spectra with NaMaster:
    Arbitrary-spin quantities.
    Systematics deprojection.
    E/B purification.
    More work in progress.
  - Compute theory predictions with CCL:
    Background quantities.
    Halo-model quantities.
    Power spectra.
    Correlation functions.
    More work in progress.

- **Data compression** can help mitigate problems with covariance estimation (among other things).
  - Particularly true for cosmic shear due to strong inter-bin correlations.
  - Proof of concept: application to CFHTLens data (full constraints recovered with 2 modes).

- There is **no need to account for parameter dependence of the covariance matrix** in two-point analyses.

# Summary

- **Two computational tools** for future large-scale structure experiments developed by the LSST DESC:
  - Compute power spectra with NaMaster:
    Arbitrary-spin quantities.
    Systematics deprojection.
    E/B purification.
    More work in progress.
  - Compute theory predictions with CCL:
    Background quantities.
    Halo-model quantities.
    Power spectra.
    Correlation functions.
    More work in progress.

- **Data compression** can help mitigate problems with covariance estimation (among other things).
  - Particularly true for cosmic shear due to strong inter-bin correlations.
  - Proof of concept: application to CFHTLens data (full constraints recovered with 2 modes).

- There is **no need to account for parameter dependence of the covariance matrix** in two-point analyses.

**Obrigado!**