Changing the way we analyse CMB data with

candl

L. Balkenhol*
with C. Trendafilova, K. Benabed, and S. Galli
*lennart.balkenhol@iap.fr

IAP, 19/11/2024, GDR Cоф Tools

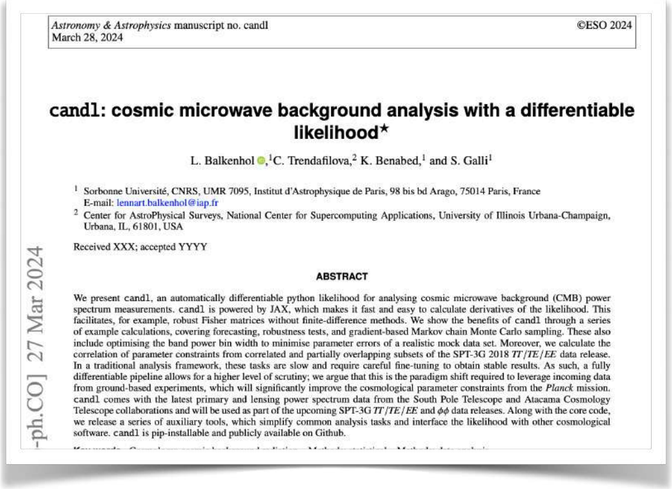
European Research Council
Established by the European Commission

Many Thanks To My Collaborators!

C. Trendafilova (UIUC)

K. Benabed (IAP)

S. Galli (IAP)



Astronomy & Astrophysics manuscript no. candl
March 28, 2024
©ESO 2024

candl: cosmic microwave background analysis with a differentiable likelihood*

L. Balkenhol,[1] C. Trendafilova,[2] K. Benabed,[1] and S. Galli[1]

[1] Sorbonne Université, CNRS, UMR 7095, Institut d'Astrophysique de Paris, 98 bis bd Arago, 75014 Paris, France
E-mail: lennart.balkenhol@iap.fr
[2] Center for AstroPhysical Surveys, National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA

Received XXX; accepted YYYY

ABSTRACT

We present candl, an automatically differentiable python likelihood for analysing cosmic microwave background (CMB) power spectrum measurements. candl is powered by JAX, which makes it fast and easy to calculate derivatives of the likelihood. This facilitates, for example, robust Fisher matrices without finite-difference methods. We show the benefits of candl through a series of example calculations, covering forecasting, robustness tests, and gradient-based Markov chain Monte Carlo sampling. These also include optimising the band power bin width to minimise parameter errors of a realistic mock data set. Moreover, we calculate the correlation of parameter constraints from correlated and partially overlapping subsets of the SPT-3G 2018 $TT/TE/EE$ data release. In a traditional analysis framework, these tasks are slow and require careful fine-tuning to obtain stable results. As such, a fully differentiable pipeline allows for a higher level of scrutiny; we argue that this is the paradigm shift required to leverage incoming data from ground-based experiments, which will significantly improve the cosmological parameter constraints from the *Planck* mission. candl comes with the latest primary and lensing power spectrum data from the South Pole Telescope and Atacama Cosmology Telescope collaborations and will be used as part of the upcoming SPT-3G $TT/TE/EE$ and $\phi\phi$ data releases. Along with the core code, we release a series of auxiliary tools, which simplify common analysis tasks and interface the likelihood with other cosmological software. candl is pip-installable and publicly available on Github.

-ph.CO] 27 Mar 2024

arXiv:2401.13433, published in A&A

E. Hivon (IAP)

F. Guidi (IAP)

A. R. Khalifeh (IAP)

E. Camphuis (IAP)

A. Vitrier (IAP)

Feedback from the SPT-3G Collaboration

Overview

1. Background: CMB
2. Traditional CMB Likelihood Analysis, Recent Developments
3. **candl** and Applications of a Fully Differentiable Pipeline
4. Getting Started & Conclusions

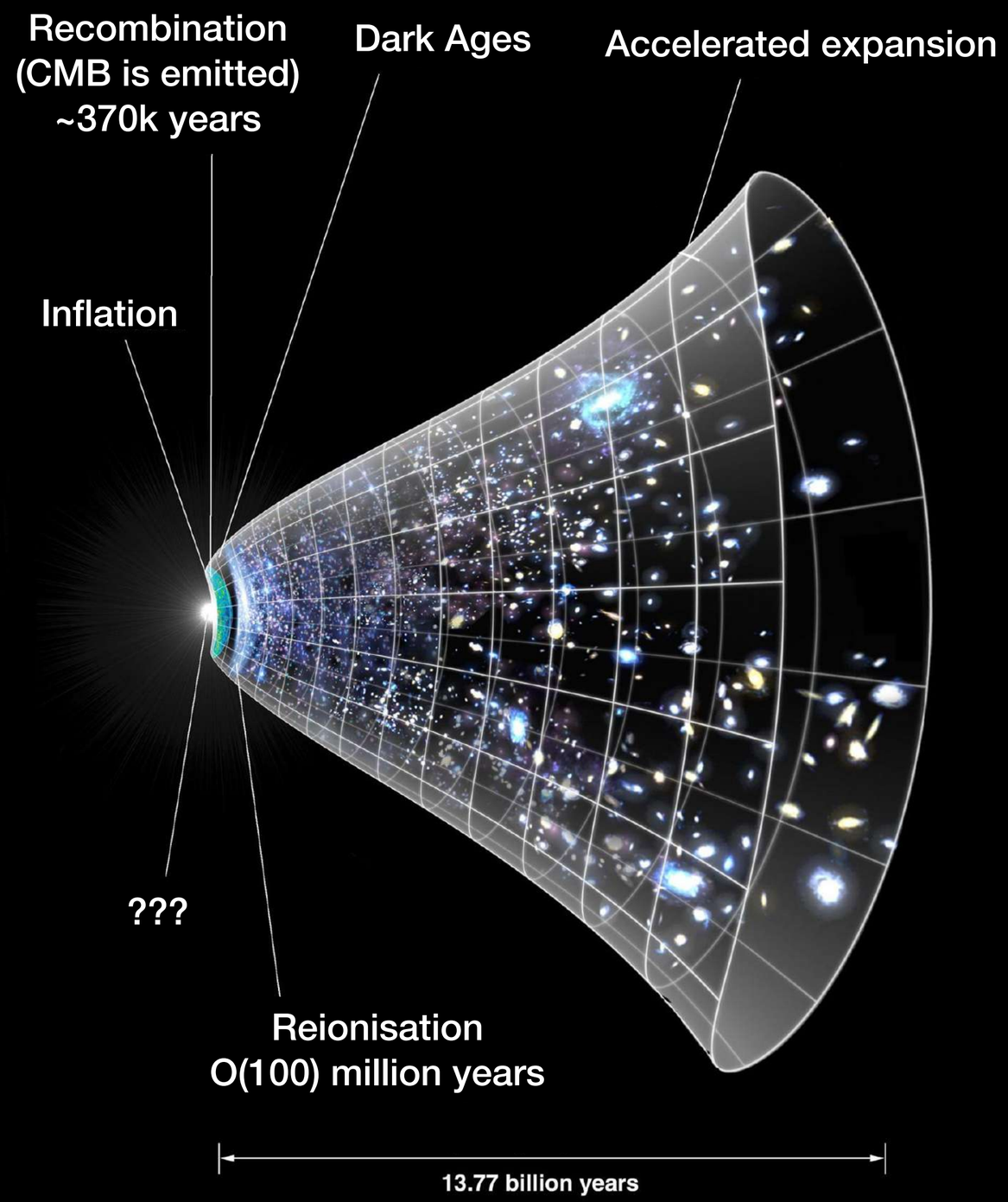
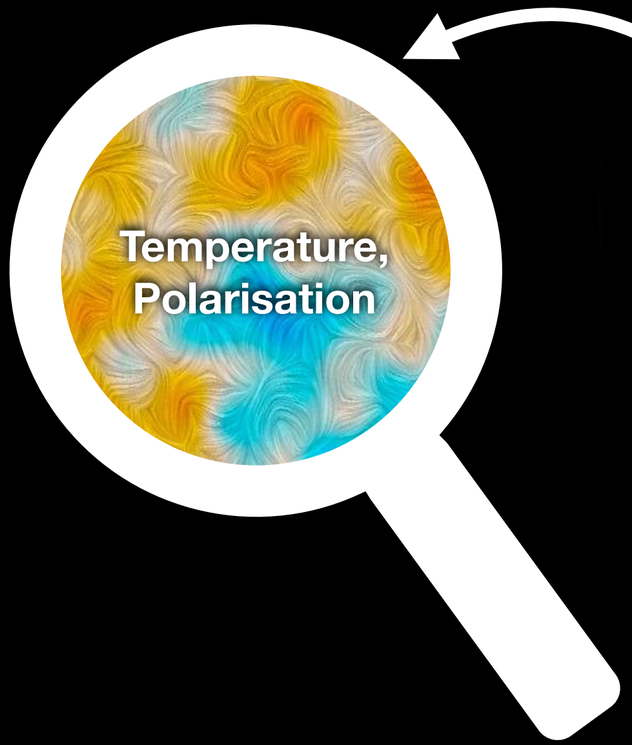
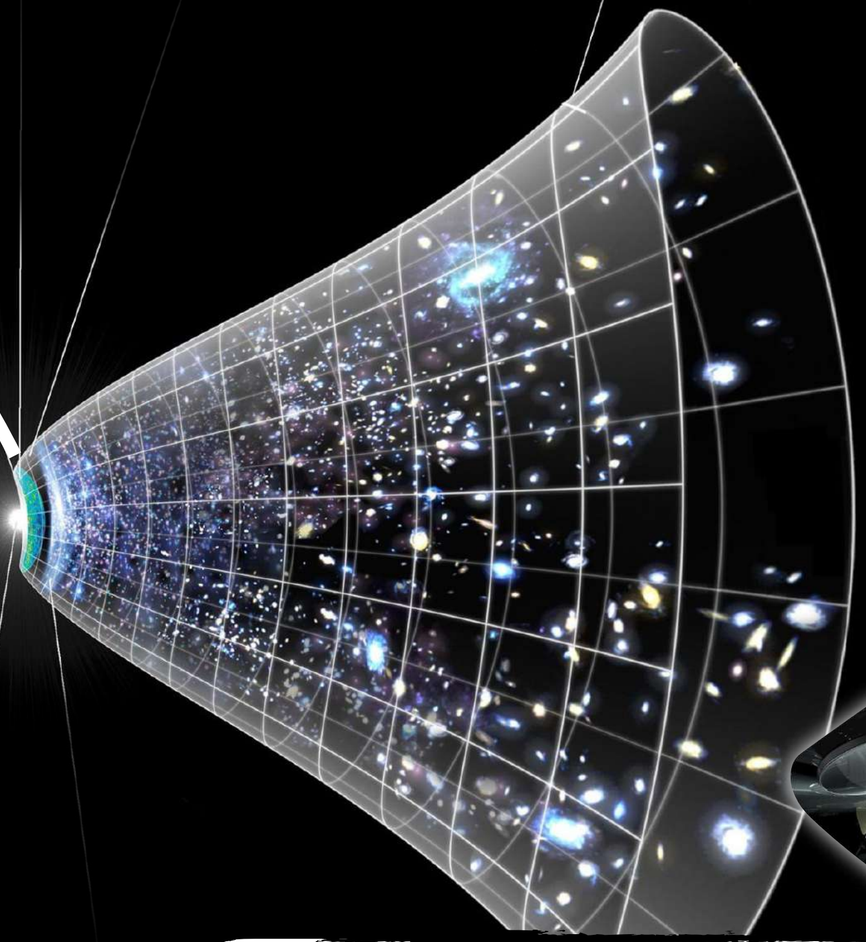
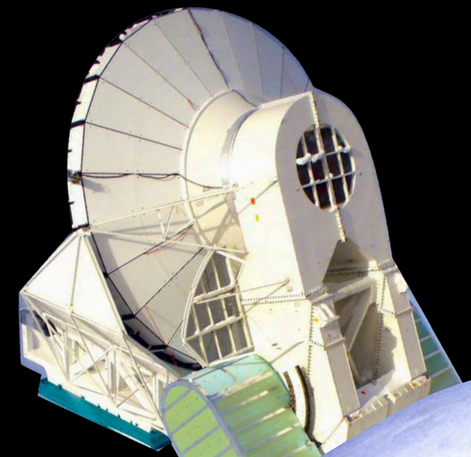


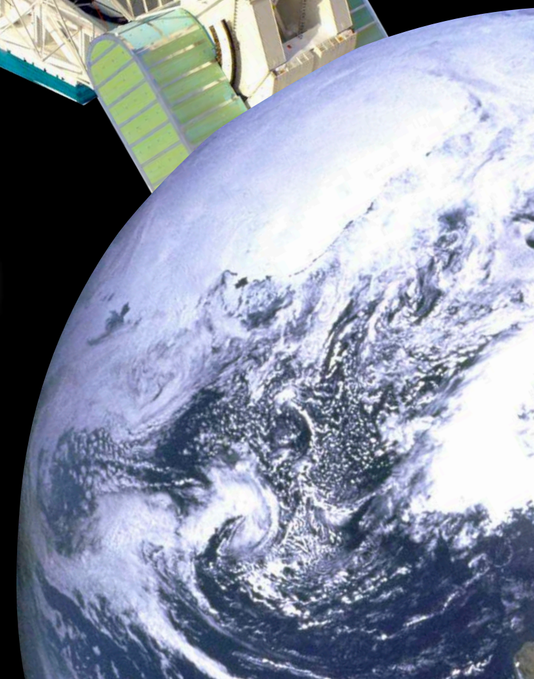
Image Credit:
NASA/WMAP Science Team
Art by Dana Berry

South Pole Telescope,
~16k detectors,
IAP (NEUCosmos) with leading contribution to different analyses







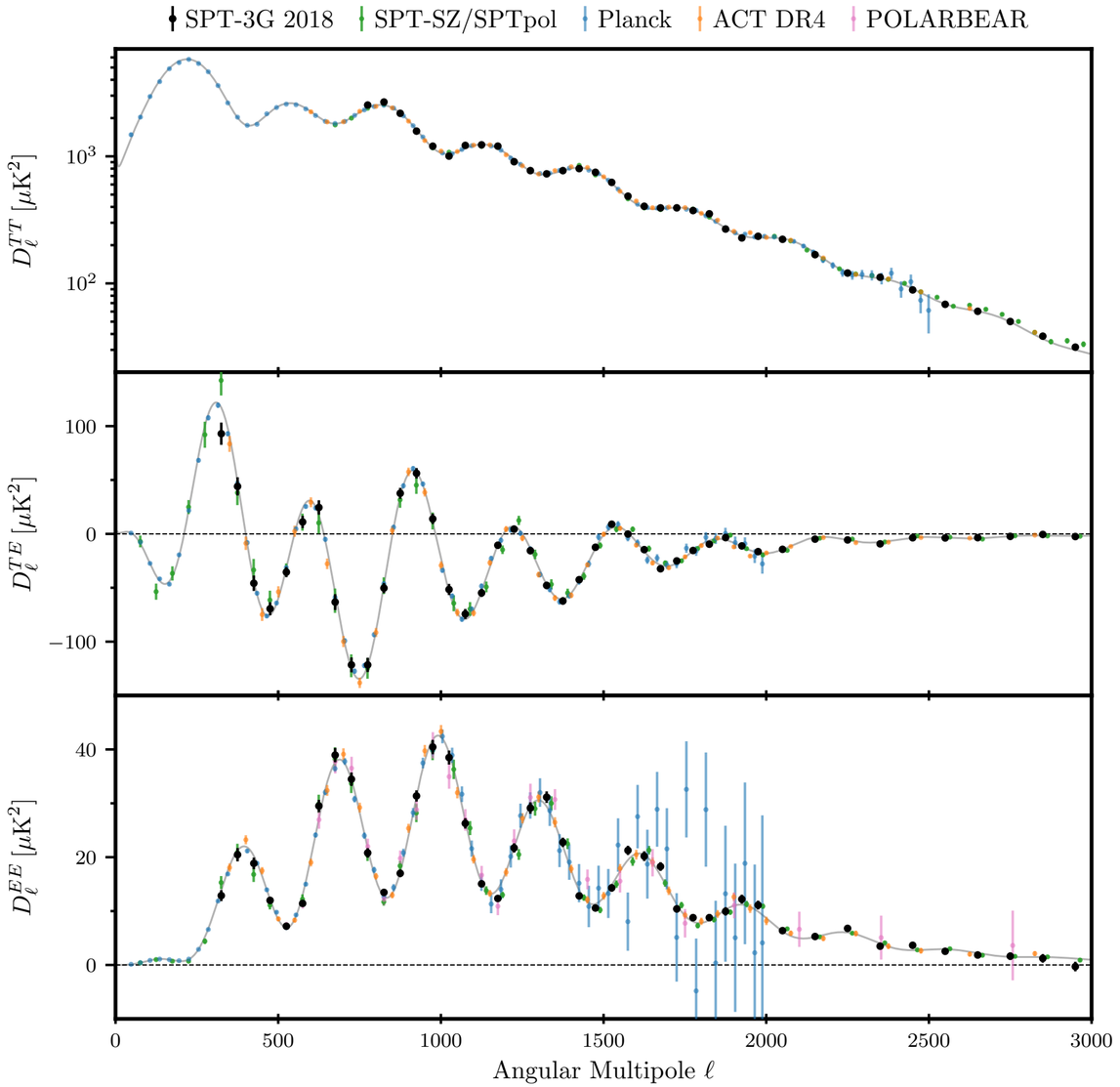


Planck satellite,
<100 bolometers,
2018 release “state-of-the-field”

Image Credit:
NASA/WMAP Science Team
Art by Dana Berry

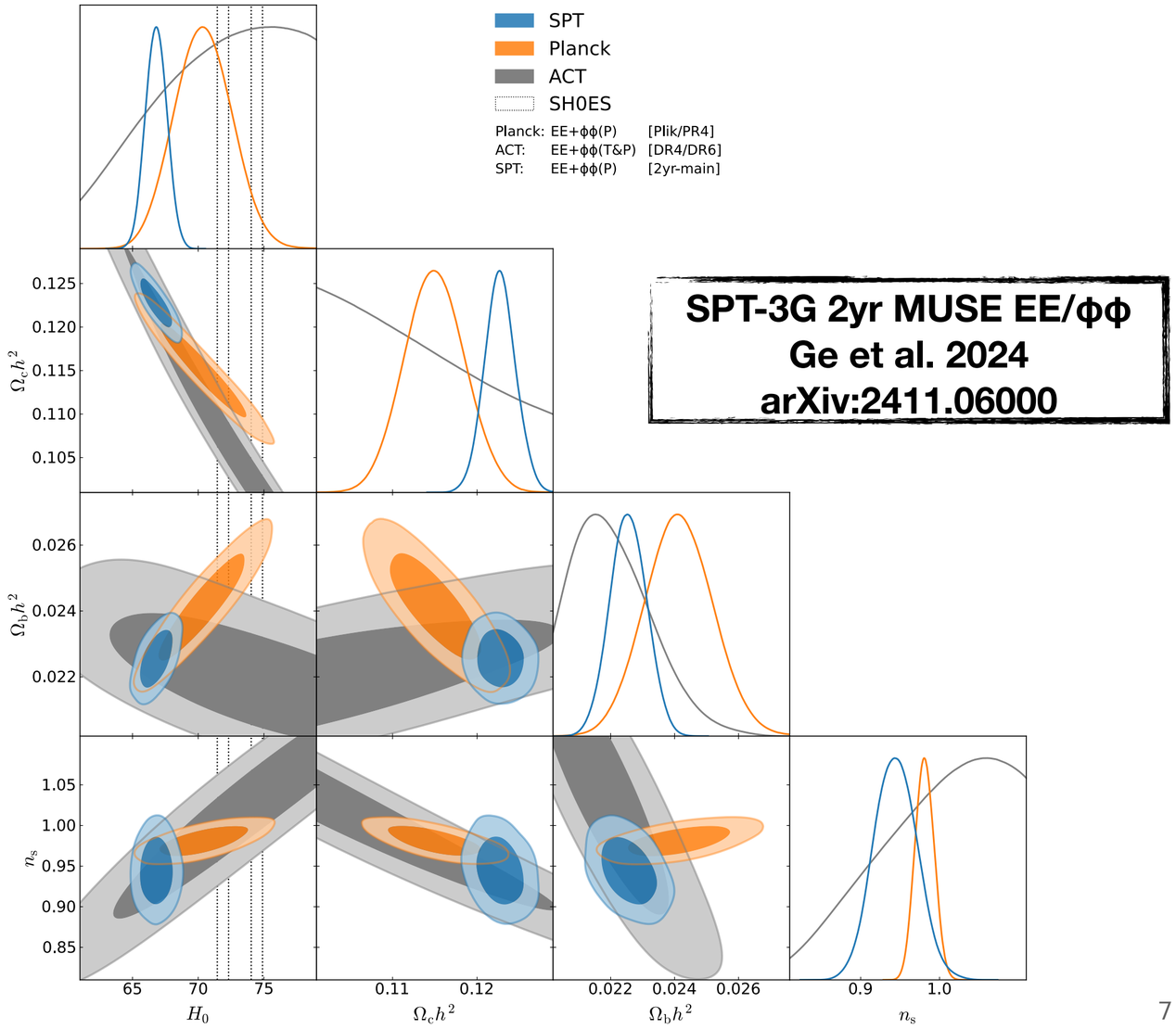
The CMB Power Spectrum

- Characteristic CMB power spectrum: acoustic peaks and small-scale damping
- Polarisation described in vector-basis of E-modes and B-modes
- Polarisation contains more information than temperature (Galli et al. 14) and less foreground contaminated (Reichardt et al. 2020, Henning et al. 2019)
- Polarised CMB is not fully characterised yet - measuring new modes!

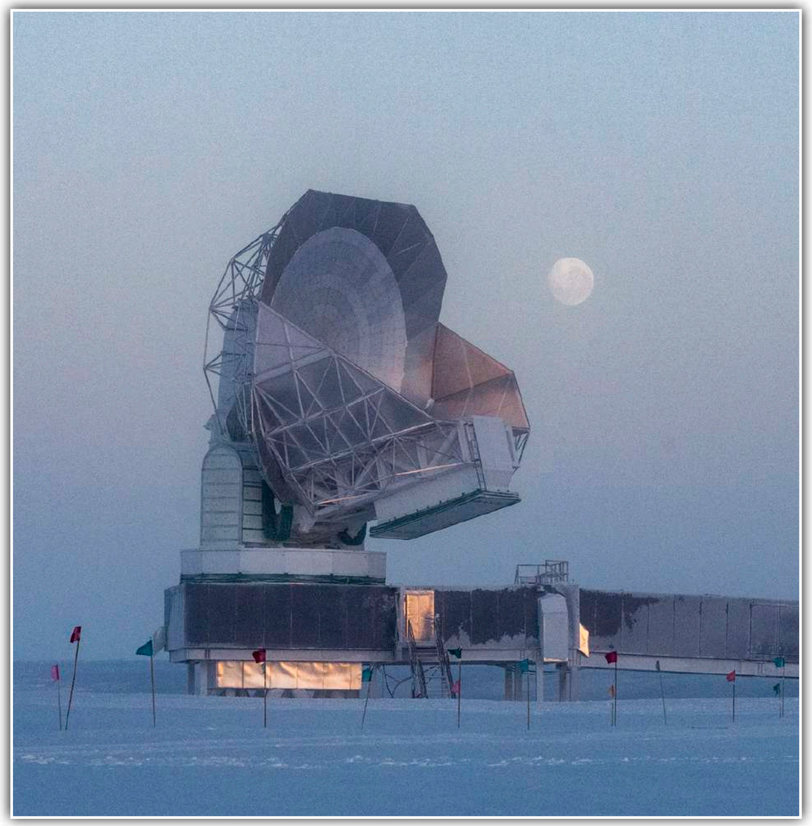


Pushing Past Planck Precision

- Contemporary CMB experiments (SPT/ACT/SO) are allowing us to push past Planck precision
  - Hot off the press: Ge et al. 2024 (2411.06000)
- Detailed understanding of systematic uncertainties and biases
- Thorough exploration of theoretical model space

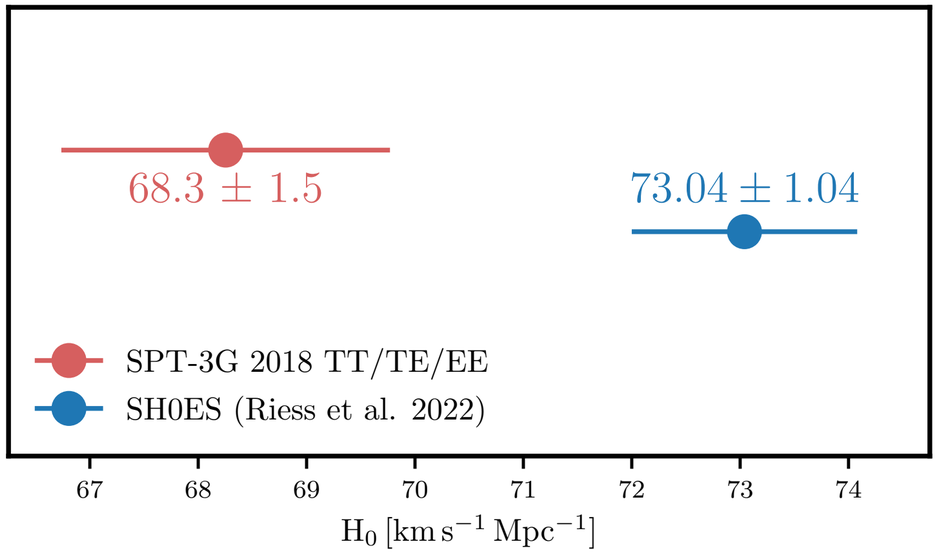


How Do We Analyse CMB Data?



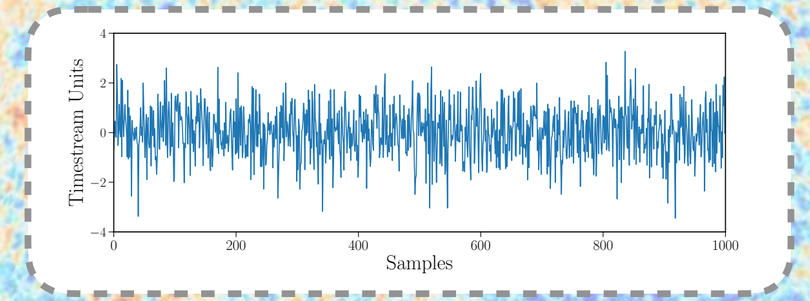
?

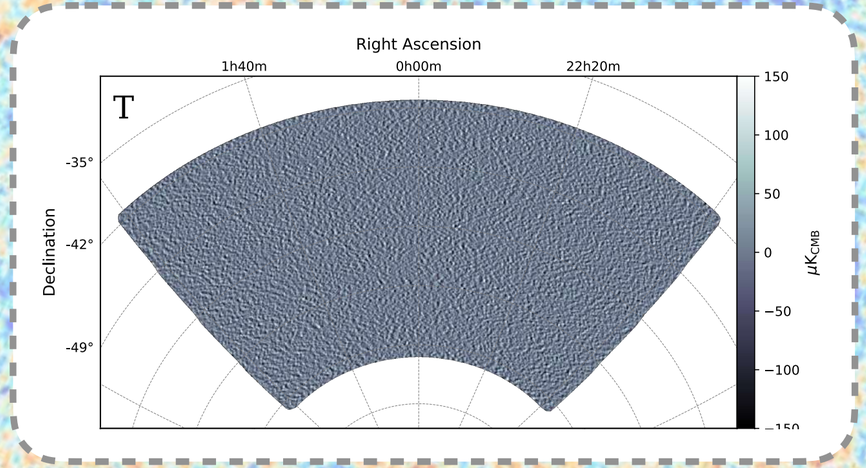
Expansion Speed of the Universe

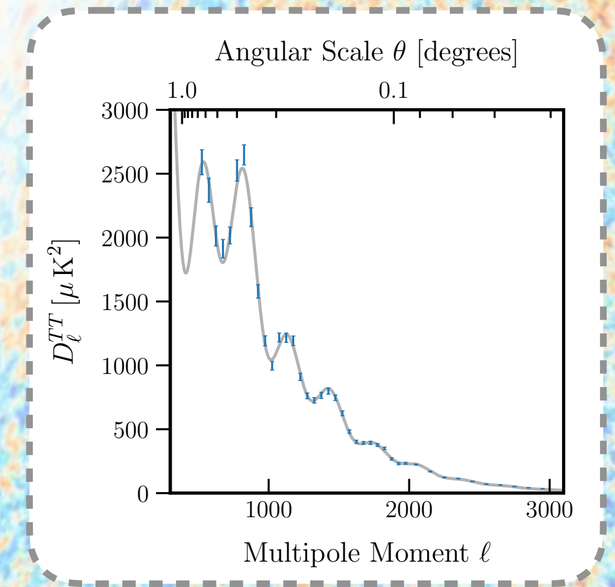


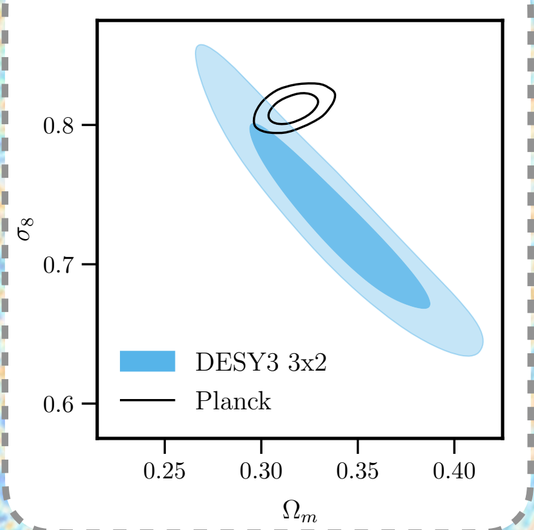
CMB Analysis

1) Telescope records time-ordered-data
2) Turn time-ordered-data into maps
3) Calculate power spectrum
4) Produce cosmological constraints
5) Interpret Results



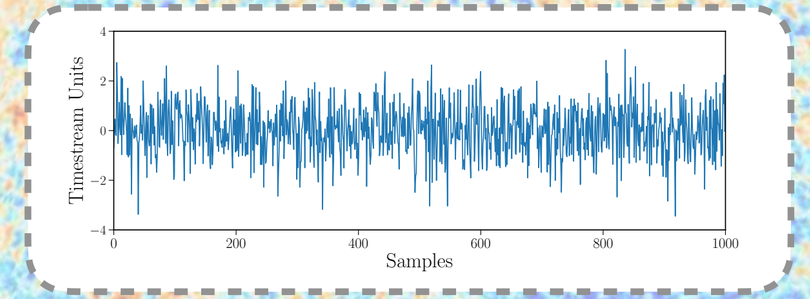


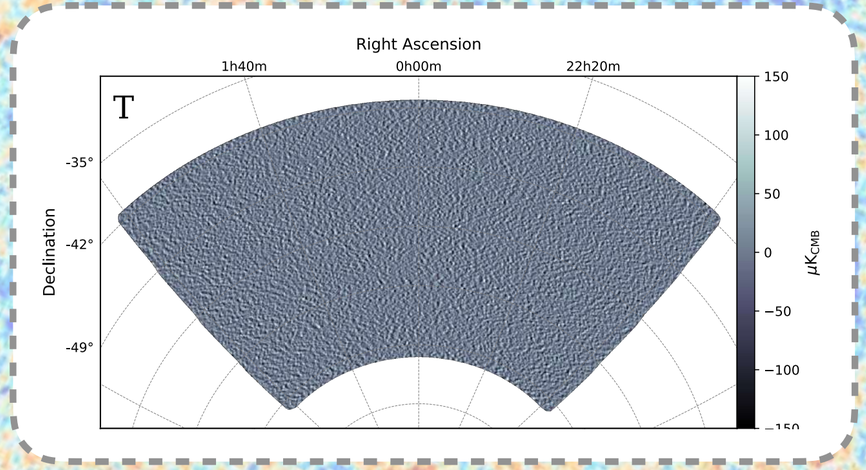


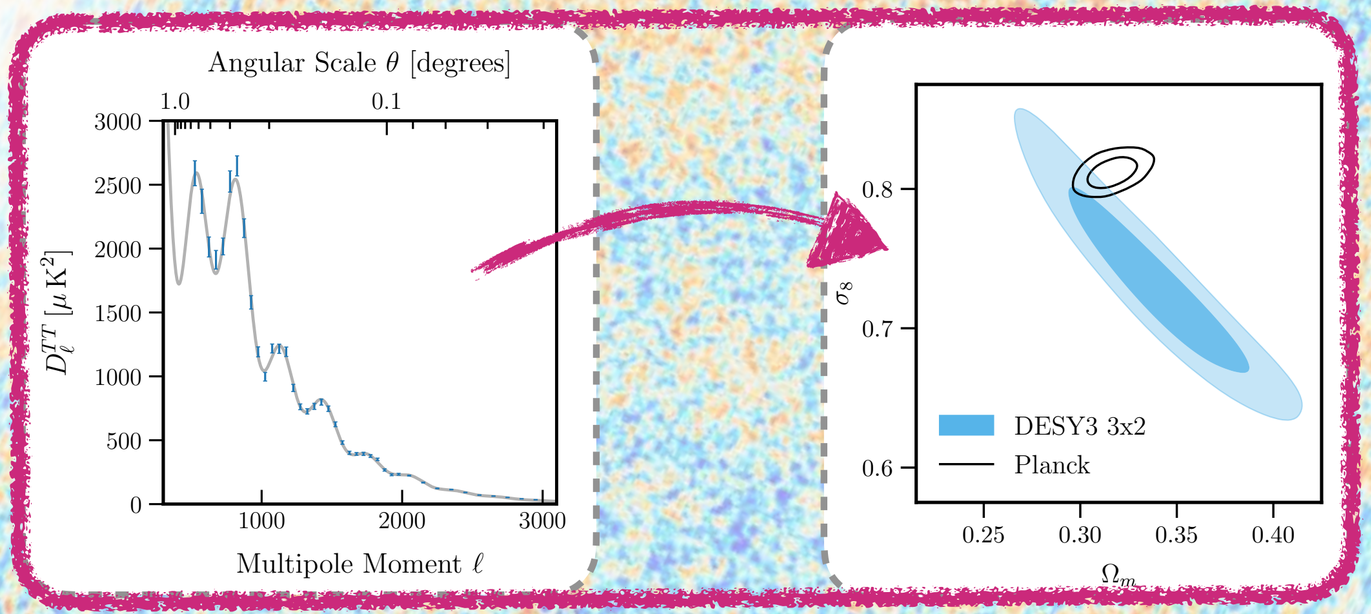


CMB Analysis

1) Telescope records time-ordered-data
2) Turn time-ordered-data into maps
3) Calculate power spectrum
4) Produce cosmological constraints
5) Interpret Results







CMB Likelihood Analysis

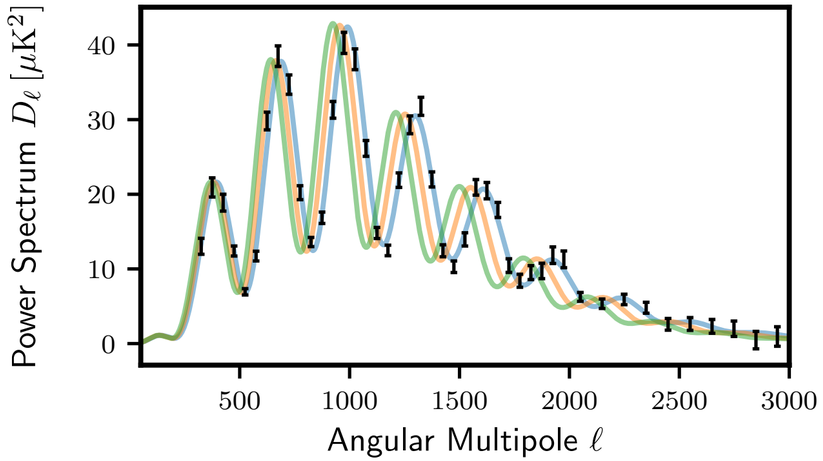
… is line fitting

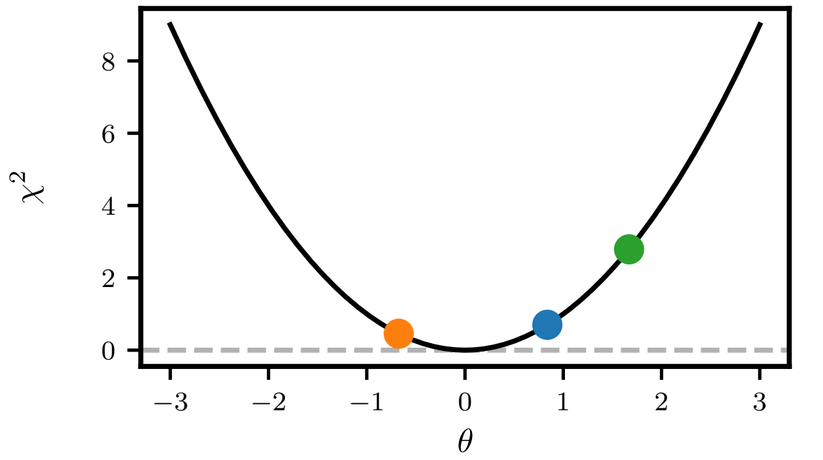
Problem: *Given measured data, find posterior distrubution of paramters for a certain model.*

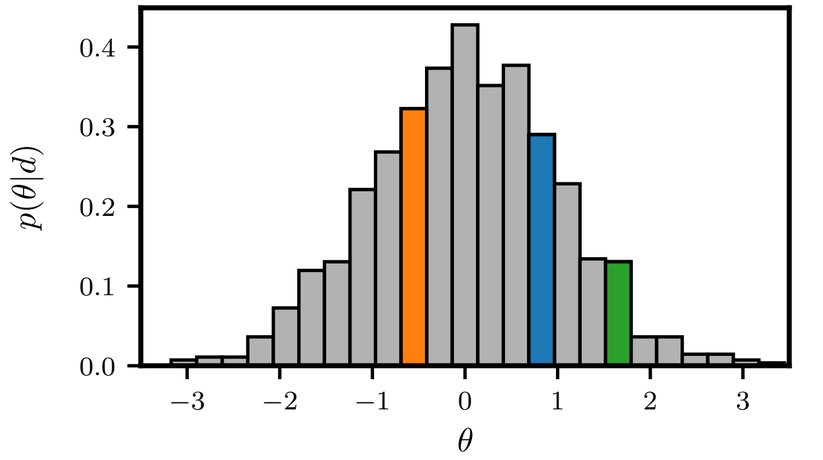
- Gaussian Likelihood form:

$$\chi^2 = \left(\text{Data} - \text{Model}\right)^T \text{Covariance}^{-1} \left(\text{Data} - \text{Model}\right)$$

- Confront data with theory predictions, explore parameter space with Markov chain Monte Carlo sampling
  - Typically 100-1000 data points, 5-50 parameters



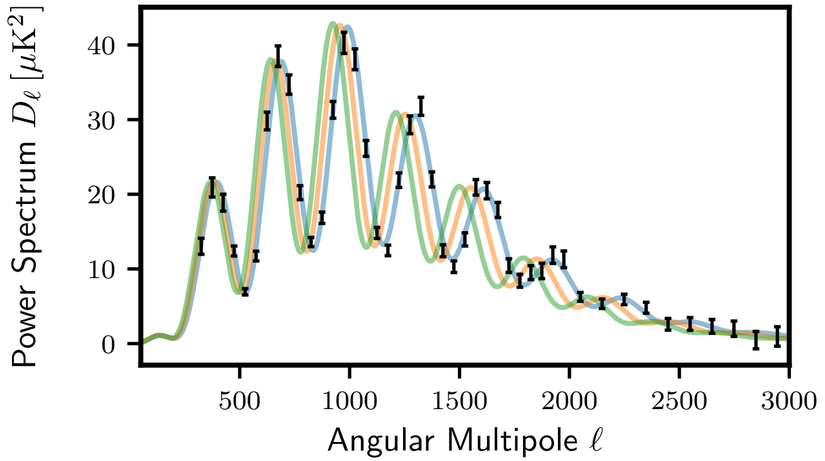


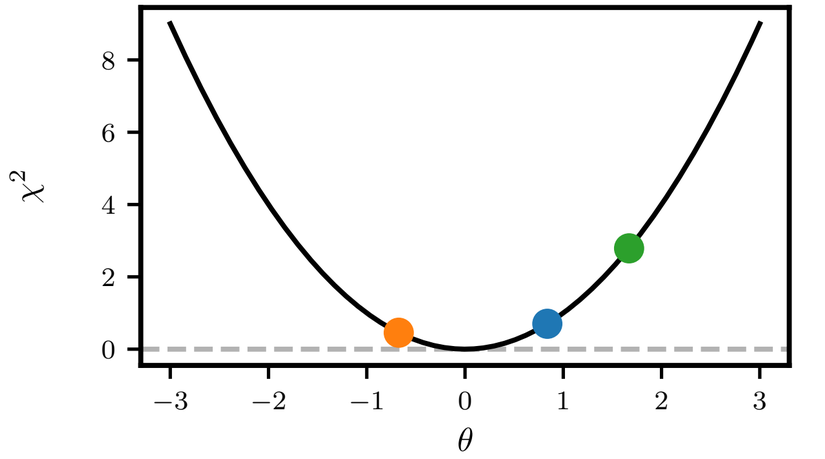


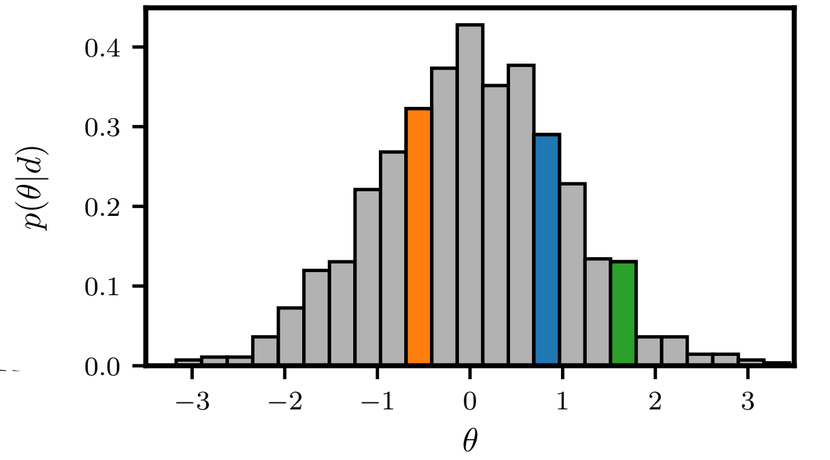
CMB Likelihood Analysis
The Traditional Approach

- Boltzmann Code - supplies CMB theory spectra
  - CLASS (arXiv:1408.4788), CAMB (arXiv:9911177)
- Likelihood - adapts theory spectra for data model and compares to data
  - Custom code in fortran or python
- Sampler - explores the parameter space
  - CosmoMC (arXiv:0205436), MontePython (arXiv:1804.07261), Cobaya (arXiv:2005.05290)

Inference is rigid and takes O(days) for a given model!

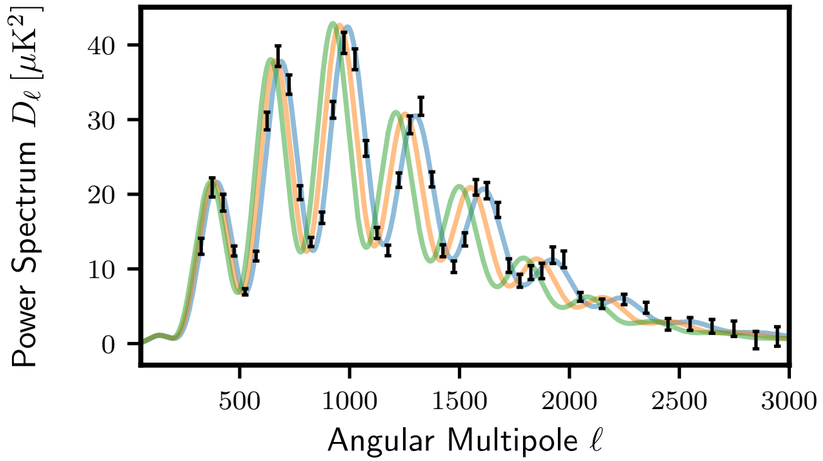


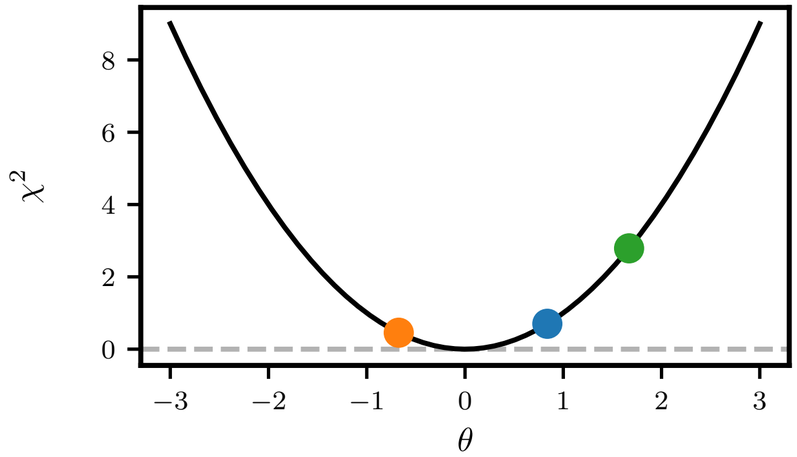


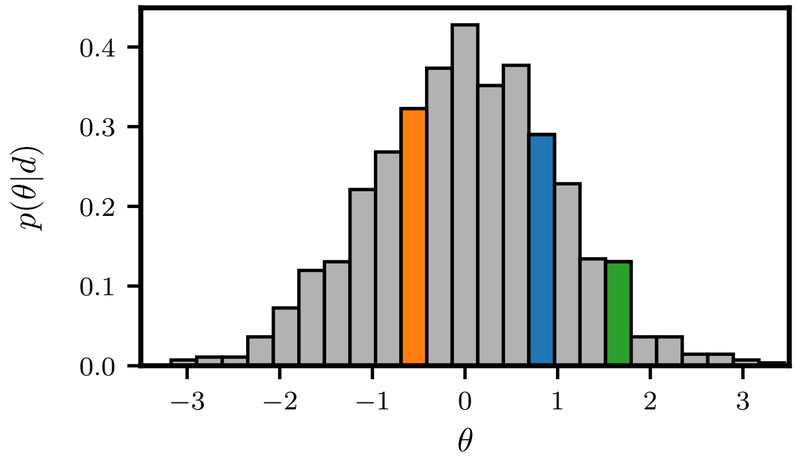


Developments in Likelihood Analysis

- **<u>Emulators replacing Boltzmann solvers:</u>**
  - Neural network emulators:
    CosmoPower (arXiv:2106.03846, arXiv:2305.06347),
    Capse.jl (arXiv:2307.14339),
    CONNECT (arXiv:2205.15726),
    COSMICNET (arXiv:1907.05764, arXiv:2207.05707)
  - Gaussian process emulators:
    OLÉ (arXiv:2307.01138, Günther et al. in prep.),
    GPry* (arXiv:2211.02045)
- **<u>Accelerating sampling with gradient information:</u>**
  - HMC, NUTS, Microcanocial HMC (arXiv:2212.08549)
  - Laplace approx. (arXiv:2301.11895)

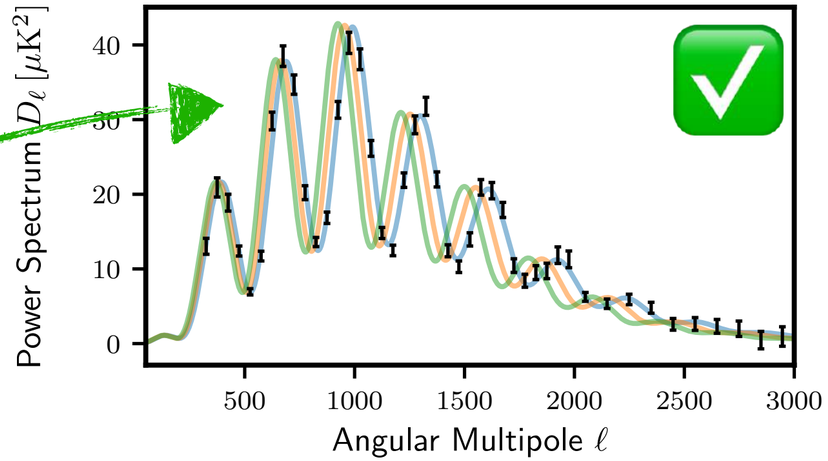


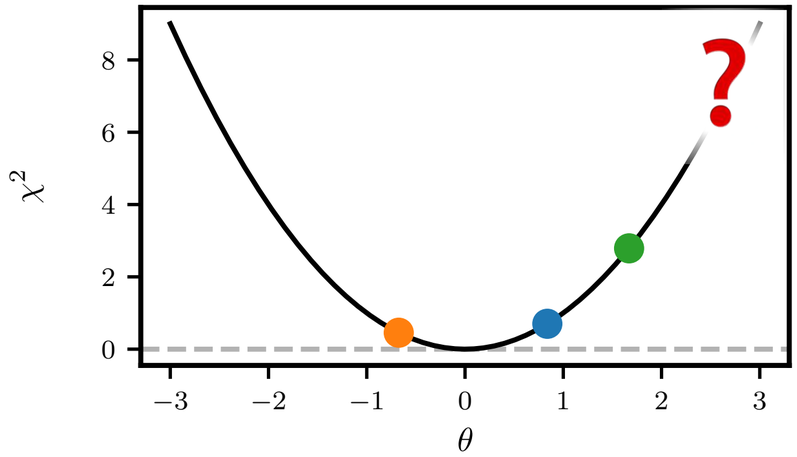


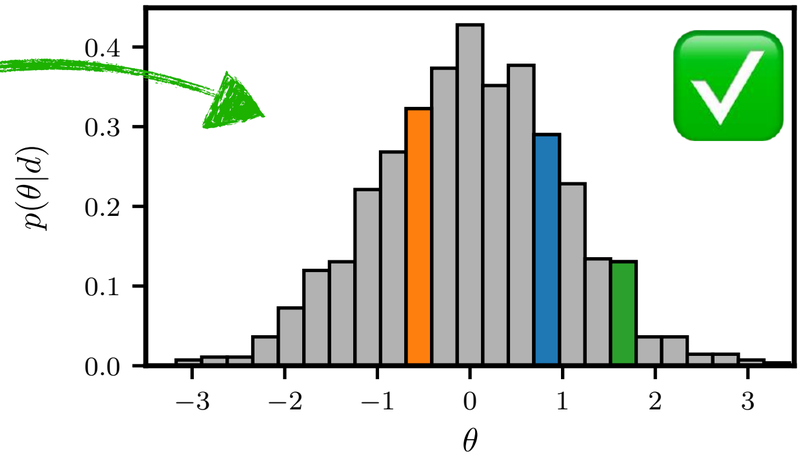


Developments in Likelihood Analysis

- **<u>Emulators replacing Boltzmann solvers:</u>**
  - Neural network emulators:
    CosmoPower (arXiv:2106.03846, arXiv:2305.06347),
    Capse.jl (arXiv:2307.14339),
    CONNECT (arXiv:2205.15726),
    COSMICNET (arXiv:1907.05764, arXiv:2207.05707)
  - Gaussian process emulators:
    OLÉ (arXiv:2307.01138, Günther et al. in prep.),
    GPry* (arXiv:2211.02045)
- **<u>Accelerating sampling with gradient information:</u>**
  - HMC, NUTS, Microcanocial HMC (arXiv:2212.08549)
  - Laplace approx. (arXiv:2301.11895)







New CMB Data Deserves Better

- Ground-based CMB experiments (SPT-3G ’20 and ACT DR6 data) achieving Planck precision imminently

“With great data comes great responsibility”

- **Need**: light, flexible pipeline to power consistency checks and robustness tests
  - Unchartered territory: claims of new physics need confidence
  - Distribution: Data and tools need to be as accessible to the community as possible
- **Opportunity:**
  - Field is transitioning to Python, embracing developments from previous slide
  - Increased use of JAX (see JAX-COSMO for why arXiv:2302.05163)

CMB **An**alysis with a **D**ifferentiable **L**ikelihood

candl

General, stand-alone, python-based likelihood

TT/TE/EE/BB/ϕϕ

Doesn't require Cobaya, Montepython, CosmoMC, …

Intuitive access to data, pip-installable

- Easy, straightforward interface with Cobaya, MontePython, CAMB, CLASS, CosmoPower, PyCapse
- Latest SPT-3G ('18 TT/TE/EE, ϕϕ) and ACT (DR4 TT/TE/EE, 6 ϕϕ) data sets available
- Manipulating data and adding new data sets is easy
- **Optionally uses JAX** (https://jax.readthedocs.io/en/latest/index.html)
  - Automatically differentiable -> w/ diff. theory code (e.g. CosmoPower emulators) easy Fisher matrices
  - Just-in-time compilation, GPU optimisation, automatic vectorisation, …

```python
import candl
import candl.data

candl_like = candl.Like(candl.data.SPT3G_2018_TTTEEE)
candl_like.data_bandpowers,
candl_like.covariance,
candl_like.effective_ells

candl_like.log_like({"Dl": ..., "nuisance_1": ...})
candl_like.chi_square({"Dl": ..., "nuisance_1": ...})

candl_like.data_model, candl_like.priors
```

Understand the Data and the Model!

```python
import candl
import candl.data

candl_like = candl.Like(candl.data.SPT3G_
candl_like.data_bandpowers,
candl_like.covariance,
candl_like.effective_ells

candl_like.log_like({"Dl": ..., "nuisance
candl_like.chi_square({"Dl": ..., "nuisan

candl_like.data_model, candl_like.priors
```

```
Successfully initialised candl likelihood 'SPT-3G 2018 TT/TE/EE (Balkenhol et al.
2023)' (type: <class 'candl.likelihood.Like'>).
Data loaded from '/opt/homebrew/Cellar/python@3.10/3.10.14/envs/jax_env/lib/
python3.10/site-packages/candl/data/SPT3G_2018_TTTEEE_v0/'.
Functional likelihood form: gaussian_beam_detcov
--------------------------------------------------------------------------------
It will analyse the following spectra:

TT 90x90     (35 bins, bin centres spanning ell = 774.5 - 2946.3)
TE 90x90     (44 bins, bin centres spanning ell = 325.8 - 2946.0)
EE 90x90     (44 bins, bin centres spanning ell = 325.2 - 2945.8)
…
TT 220x220   (30 bins, bin centres spanning ell = 1024.4 - 2946.3)
TE 220x220   (44 bins, bin centres spanning ell = 325.7 - 2946.1)
EE 220x220   (44 bins, bin centres spanning ell = 324.9 - 2945.9)
--------------------------------------------------------------------------------
A data model consisting of 11 transformations has been initialised.
The following transformations will be applied to the theory spectra in this order:

(1) Name: Super-Sample Lensing
    Type: <class 'candl.transformations.common.SuperSampleLensing'>
(2) Name: Aberration
    Type: <class 'candl.transformations.common.AberrationCorrection'>
(3) Name: Poisson Power
    Type: <class 'candl.transformations.common.PoissonPower'>
(4) Name: CIB clustering
    Type: <class 'candl.transformations.common.CIBClustering'>
…
(11) Name: Calibration
     Type: <class 'candl.transformations.common.CalibrationCross'>
--------------------------------------------------------------------------------
```

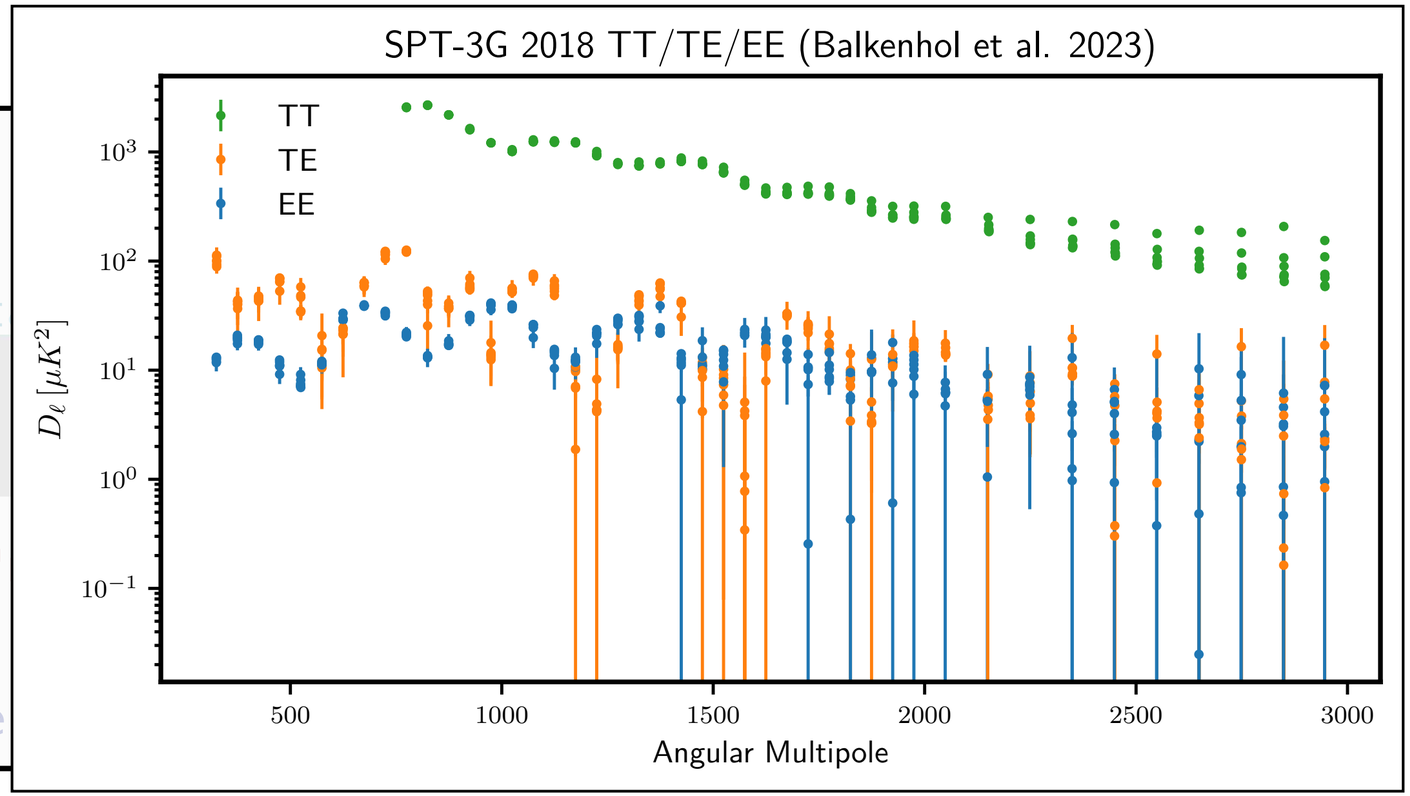
Easily Select Parts of the Data!

```
import candl
import candl.data

candl_like = candl.Like(candl.data.SPT3G_2018_TTTEEE)
candl_like.data_bandpowers,
candl_like.covariance,
candl_like.effective_ells

candl_like.log_like({"Dl": ..., "nuis
candl_like.chi_square({"Dl": ..., "nu

candl_like.data_model, candl_like.priors
```

```
data_selection = "TT only", "ell>2000 remove", …
```

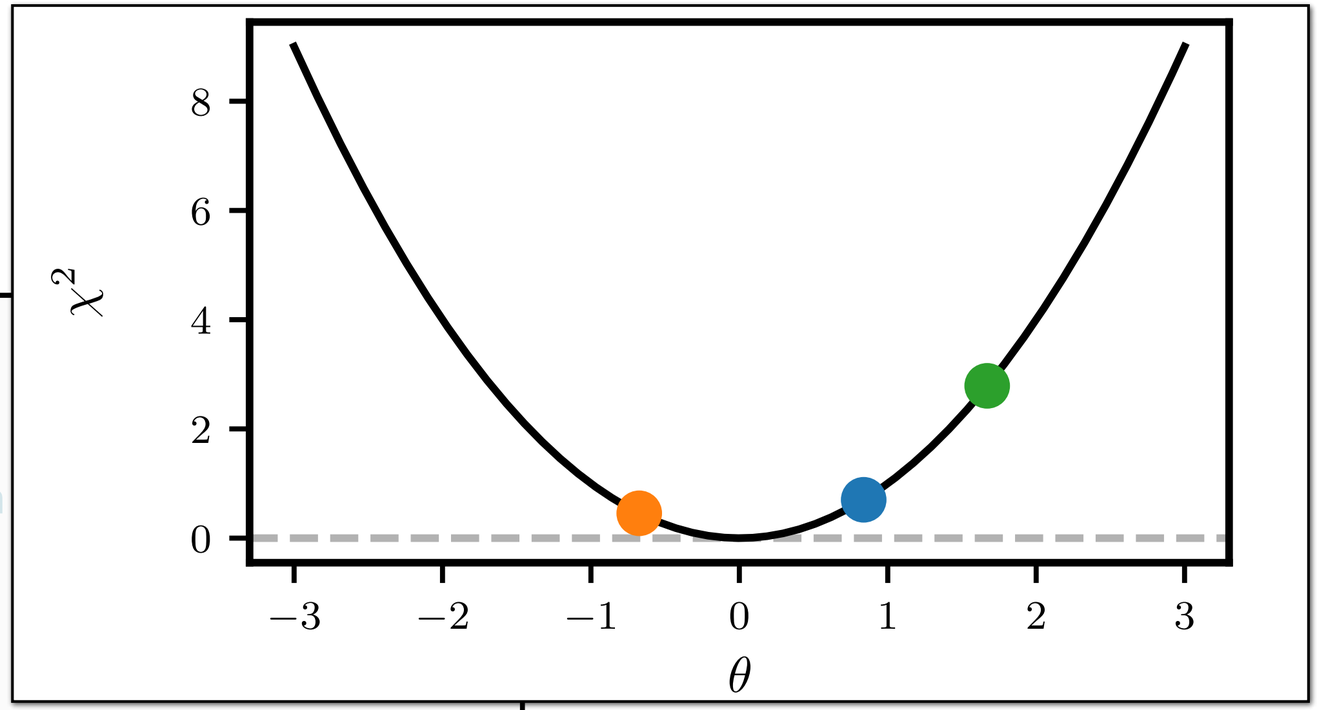
Easy Access to All Aspects of the Data!

```
import candl
import candl.data

candl_like = candl.Like(candl.dat
candl_like.data_bandpowers,
candl_like.covariance,
candl_like.effective_ells

candl_like.log_like({"Dl": ..., "
candl_like.chi_square({"Dl": ...,

candl_like.data_model, candl_like
```




```
import candl.plots
candl.plots.plot_band_powers(candl_like)
```

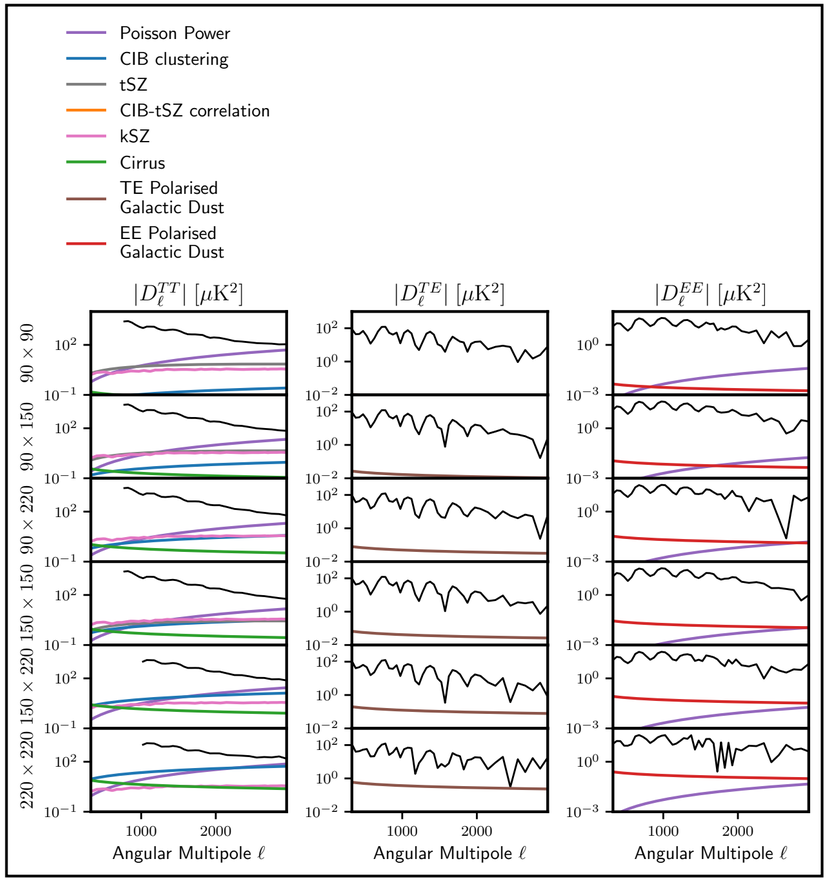
```python
import candl
import candl.data

candl_like = candl.Like(candl.data
candl_like.data_bandpowers,
candl_like.covariance,
candl_like.effective_ells

candl_like.log_like({"Dl": ..., "nuisance_1": ...})
candl_like.chi_square({"Dl": ..., "nuisance_1": ...})

candl_like.data_model, candl_like.priors
```



Evaluate the likelihood easily!

Scrutinise the Likelihood Components!

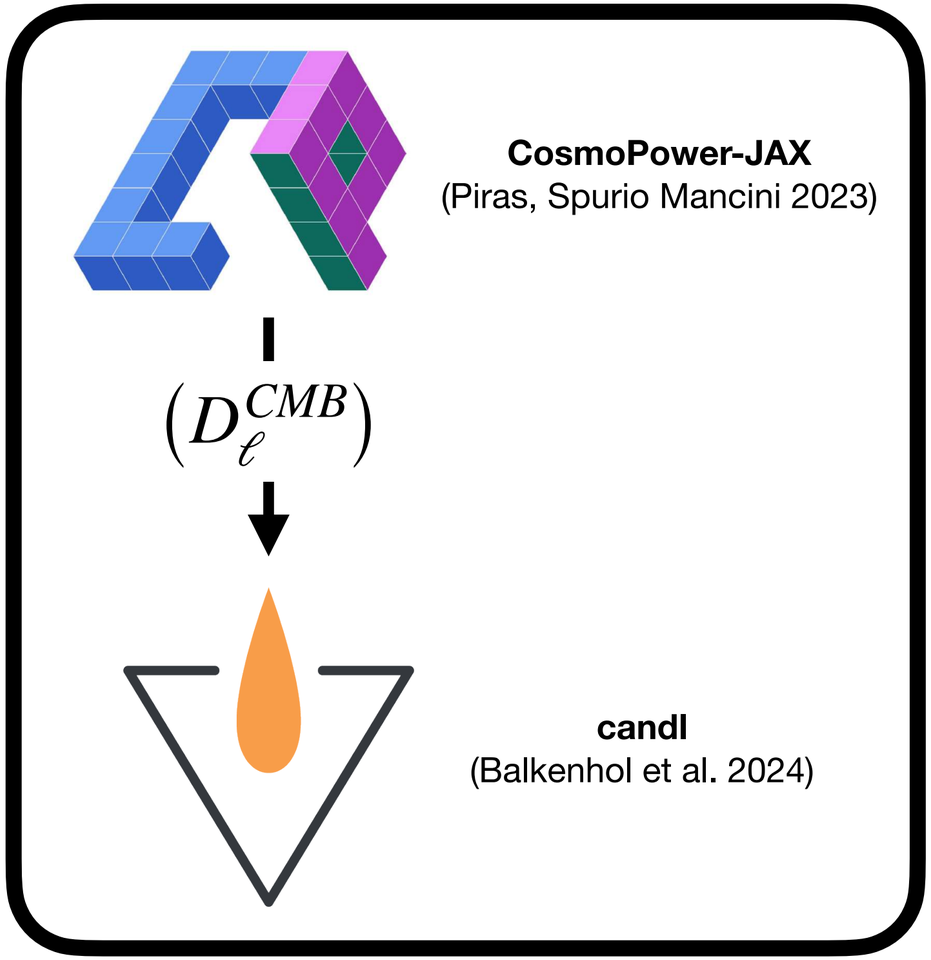
```
import candl
import candl.data

candl_like = candl.Like(candl.data.SPT3G_2018_TTTEEE)
candl_like.data_bandpowers,
candl_like.covariance,
candl_like.effective_ells

candl_like.log_like({"Dl": ..., "nuisance_1": ...})
candl_like.chi_square({"Dl": ..., "nuisance_1": ...})

candl_like.data_model, candl_like.priors
```

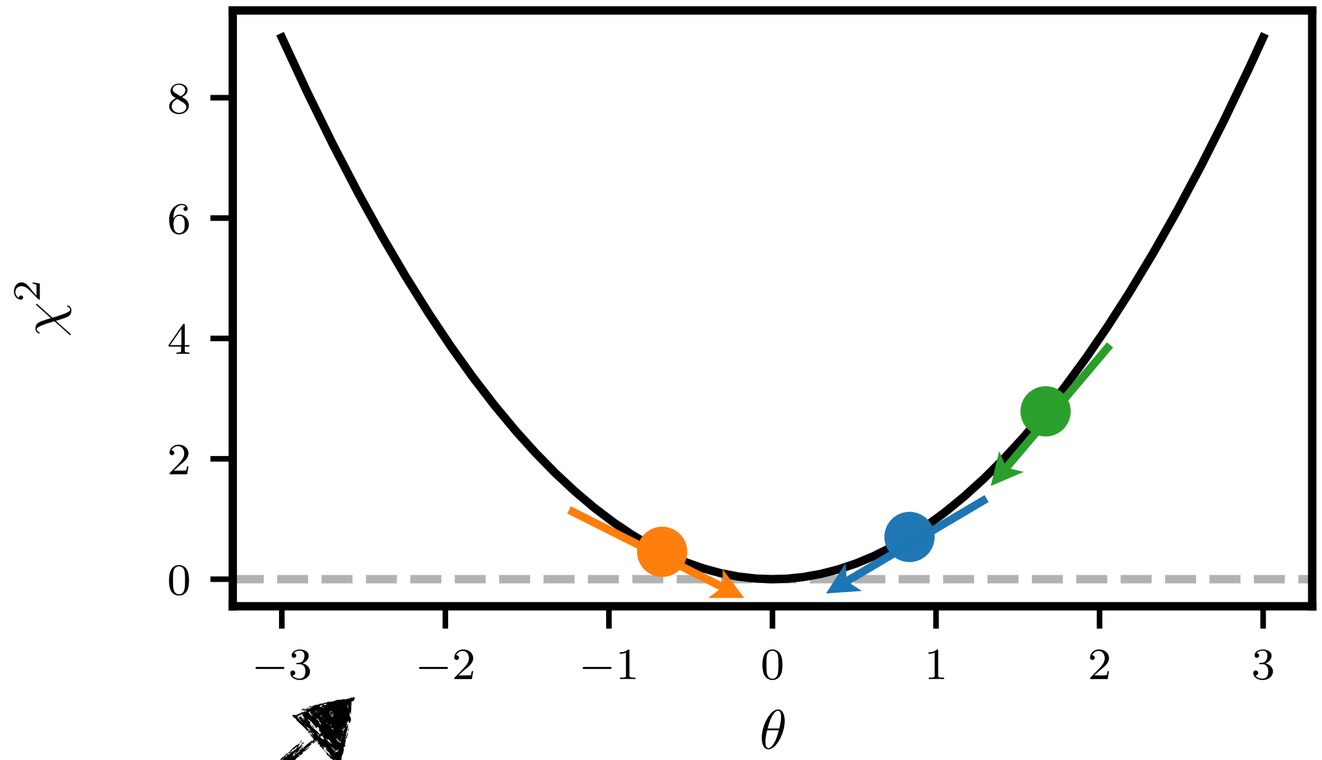



```
fg_dict = candl.tools.get_foreground_contributions(candl_like, base_fid_pars)
candl.plots.plot_foreground_components(candl_like, fg_dict)
```

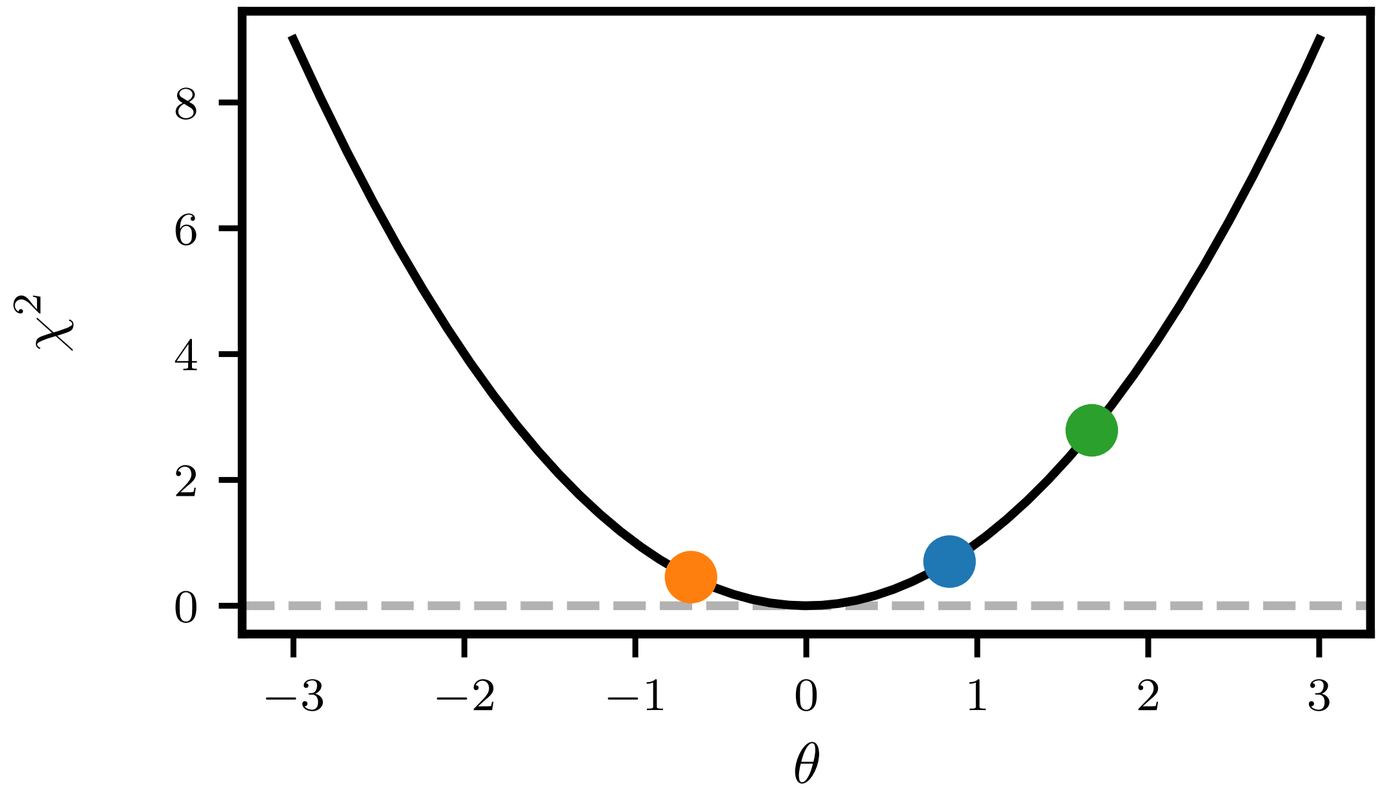
Building a Fully Differentiable Pipeline

$H_0, \omega_c, n_s, \cdots$

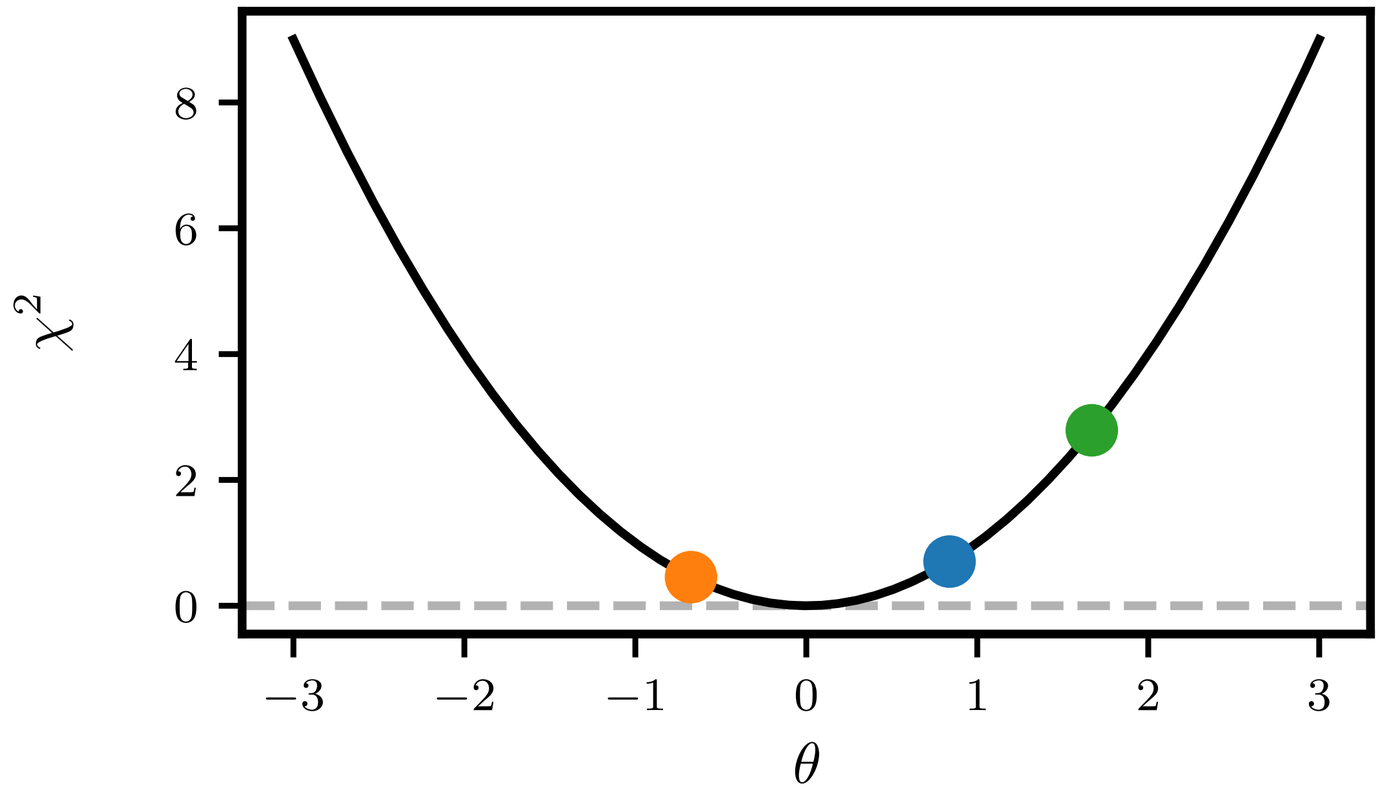




```python
import candl.tools
cp_emus = {"TT": "cmb_spt_TT_NN", "TE": "cmb_spt_TE_PCAplusNN", "EE": "cmb_spt_EE_NN"}
pars_to_theory_specs = candl.interface.get_CosmoPowerJAX_pars_to_theory_specs_func(cp_emus)

pars_to_logl = candl.tools.get_params_to_logl_func(candl_like, pars_to_theory_specs)
pars_to_logl_deriv = jax.jacrev(pars_to_logl)
candl.tools.get_fisher_matrix(pars_to_theory_specs, candl_like, {"H0": ...})
```

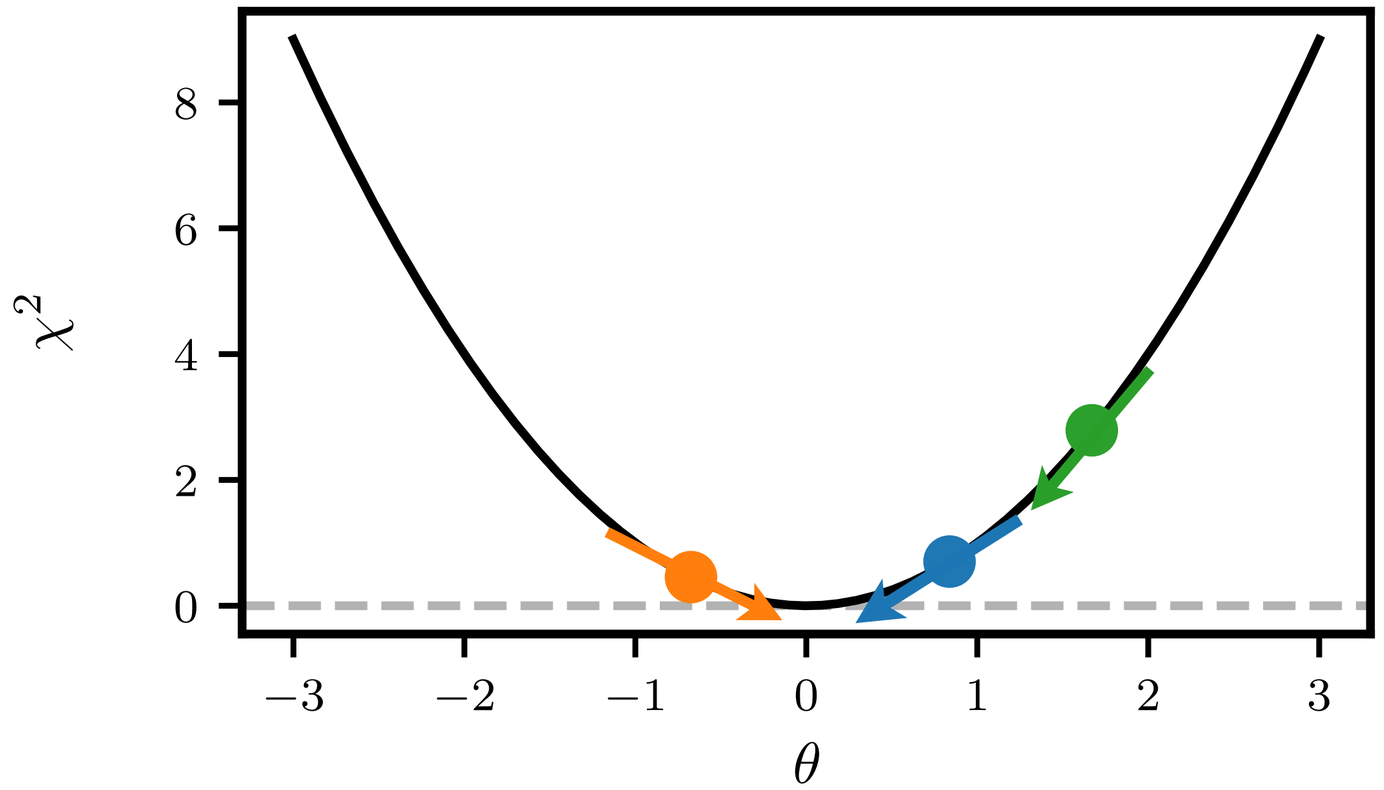


```
import candl.tools
cp_emus = {"TT": "cmb_spt_TT_NN", "TE": "cmb_spt_TE_PCAplusNN", "EE": "cmb_spt_EE_NN"}
pars_to_theory_specs = candl.interface.get_CosmoPowerJAX_pars_to_theory_specs_func(cp_emus)

pars_to_logl = candl.tools.get_params_to_logl_func(candl_like, pars_to_theory_specs)
pars_to_logl_deriv = jax.jacrev(pars_to_logl)
candl.tools.get_fisher_matrix(pars_to_theory_specs, candl_like, {"H0": ...})
```



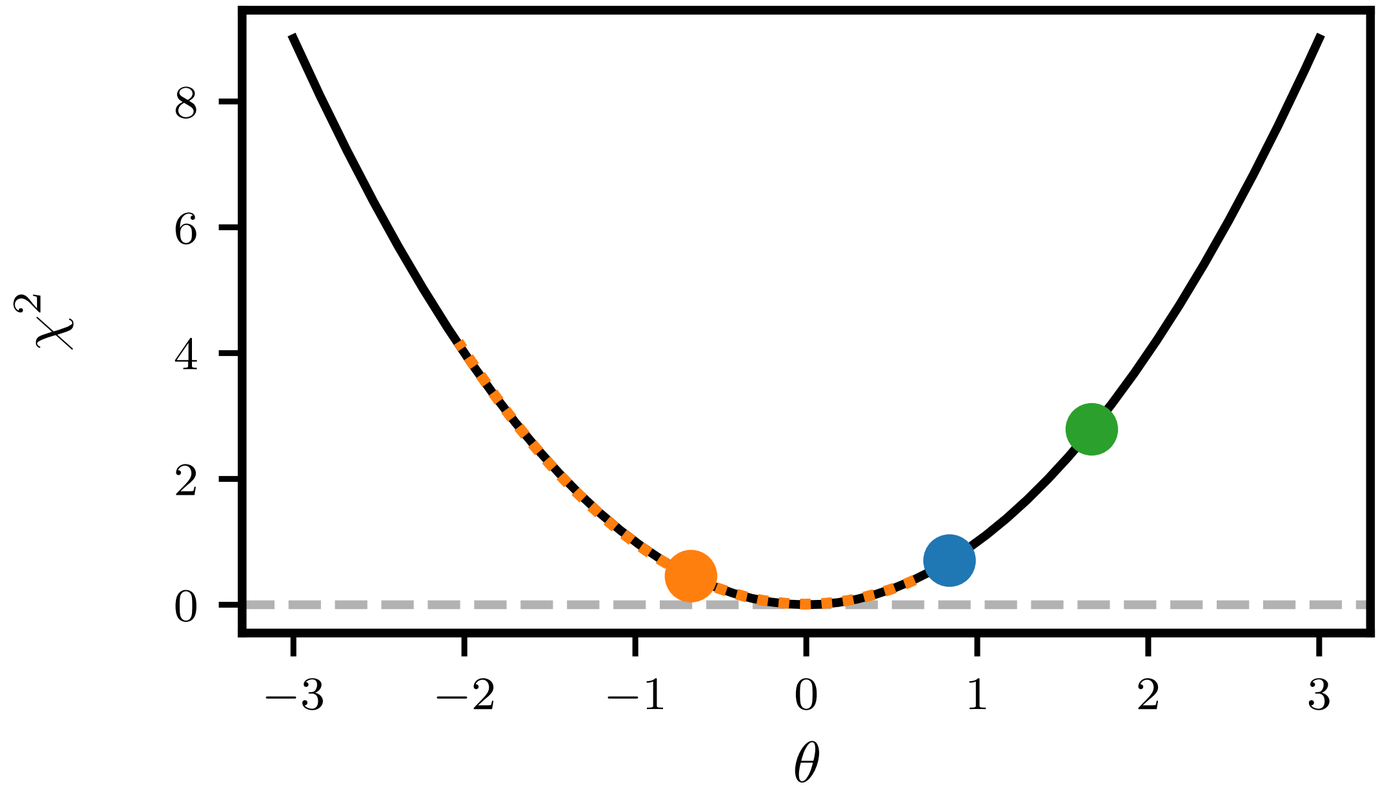
$L(\theta)$

```
import candl.tools
cp_emus = {"TT": "cmb_spt_TT_NN", "TE": "cmb_spt_TE_PCAplusNN", "EE": "cmb_spt_EE_NN"}
pars_to_theory_specs = candl.interface.get_CosmoPowerJAX_pars_to_theory_specs_func(cp_emus)

pars_to_logl = candl.tools.get_params_to_logl_func(candl_like, pars_to_theory_specs)
pars_to_logl_deriv = jax.jacrev(pars_to_logl)
candl.tools.get_fisher_matrix(pars_to_theory_specs, candl_like, {"H0": ...})
```



$$\frac{\partial L}{\partial \theta}$$

```
import candl.tools
cp_emus = {"TT": "cmb_spt_TT_NN", "TE": "cmb_spt_TE_PCAplusNN", "EE": "cmb_spt_EE_NN"}
pars_to_theory_specs = candl.interface.get_CosmoPowerJAX_pars_to_theory_specs_func(cp_emus)

pars_to_logl = candl.tools.get_params_to_logl_func(candl_like, pars_to_theory_specs)
pars_to_logl_deriv = jax.jacrev(pars_to_logl)
candl.tools.get_fisher_matrix(pars_to_theory_specs, candl_like, {"H0": ...})
```



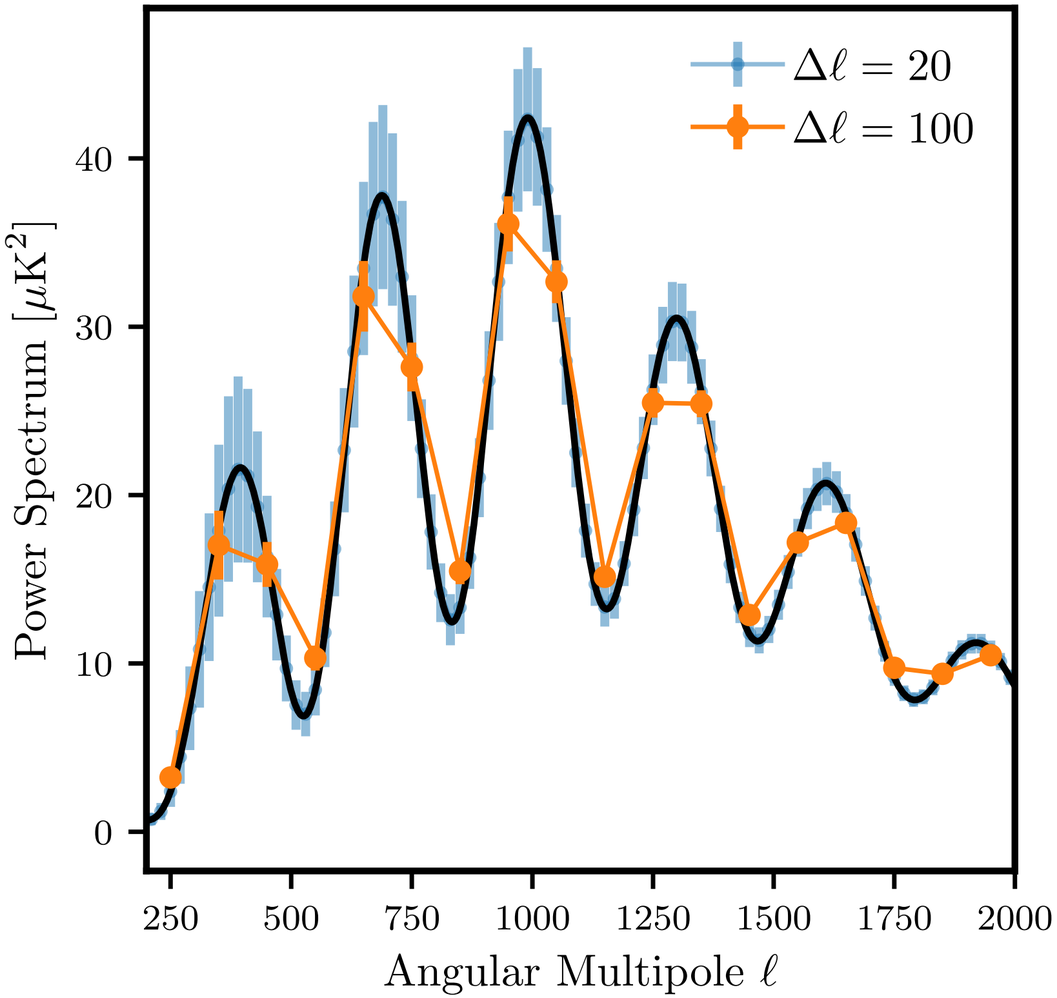
$$\left(-\frac{\partial^2 L}{\partial \theta^2}\right)^{-1}$$

Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **<u>Quick, easy, reliable Fisher matrices:</u>**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **<u>Smart exploration of the likelihood:</u>**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!

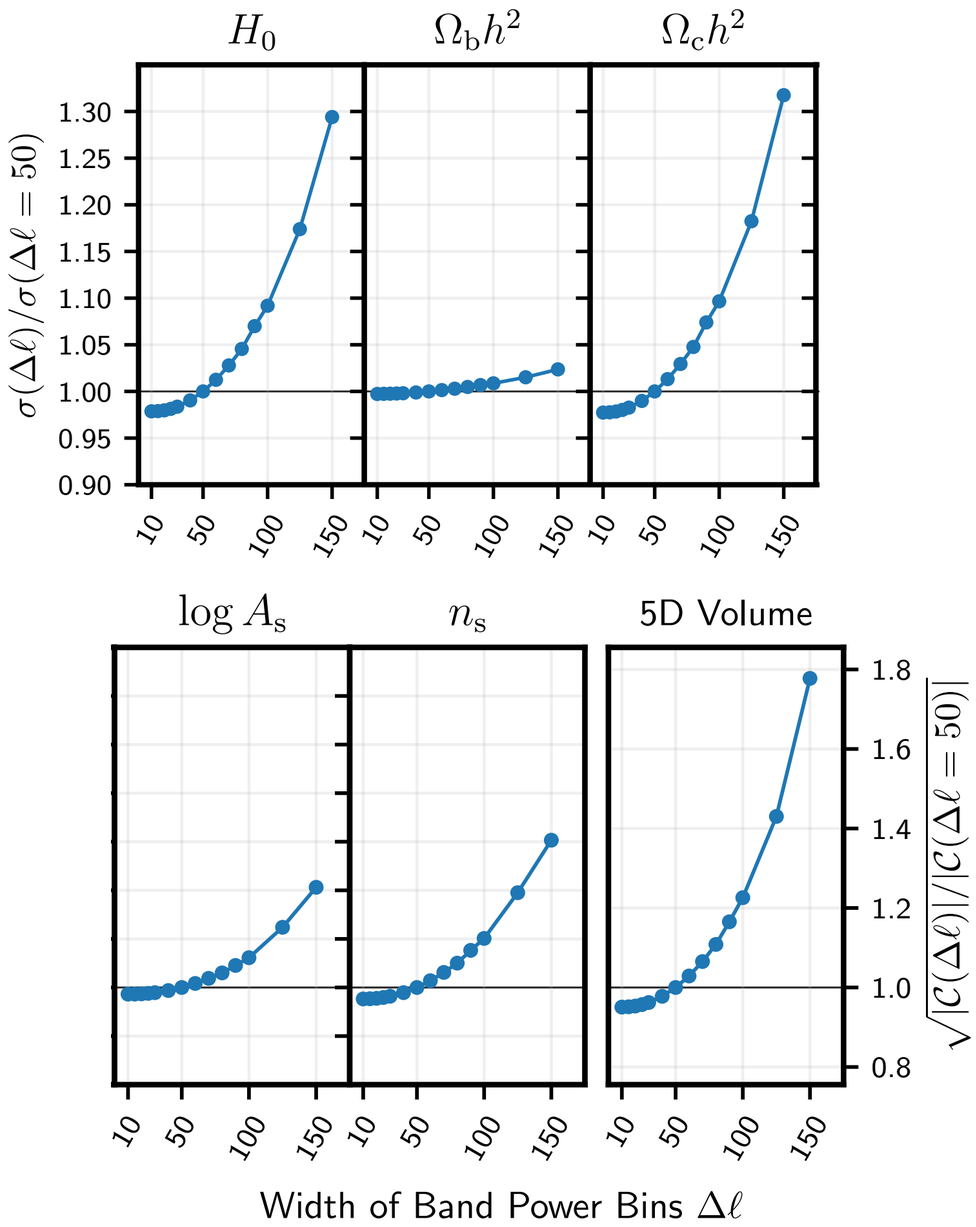
“How fine should I bin my data?”



Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

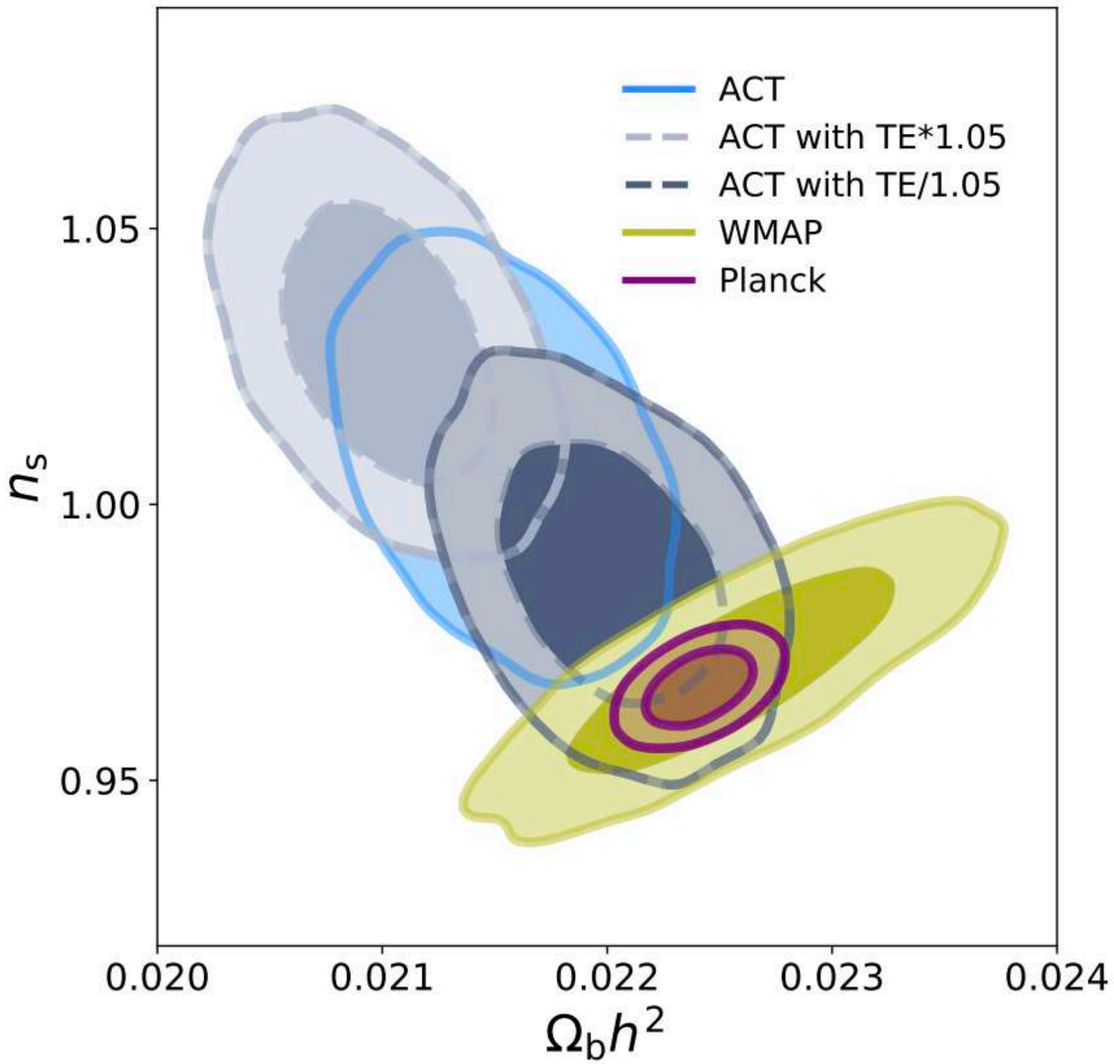
- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!



Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **<u>Quick, easy, reliable Fisher matrices:</u>**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **<u>Smart exploration of the likelihood:</u>**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!

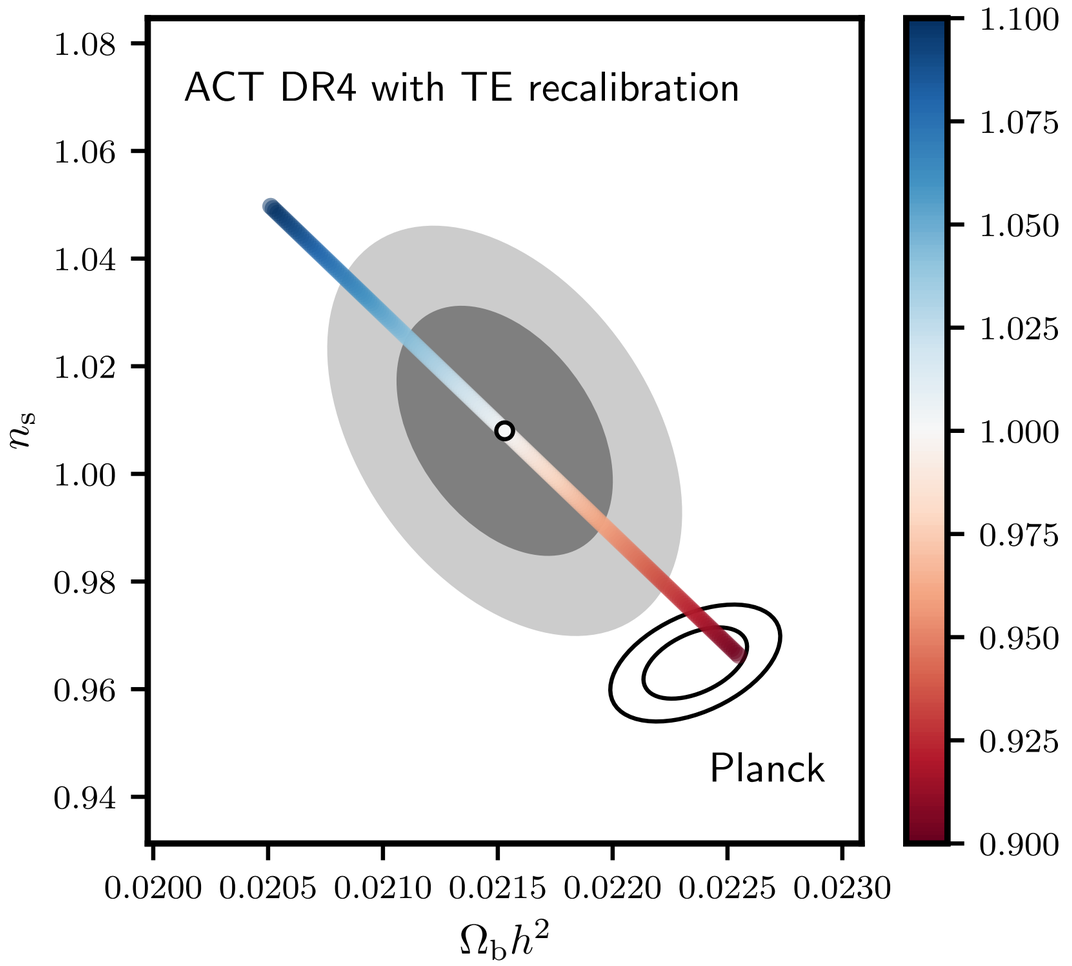


Aiola et al. 2020, Fig. 14

Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!



$$\Delta\theta = [-H]^{-1} \left.\frac{\partial D}{\partial \theta}\right|_{\text{fid}} C^{-1}\delta D$$

to parameters

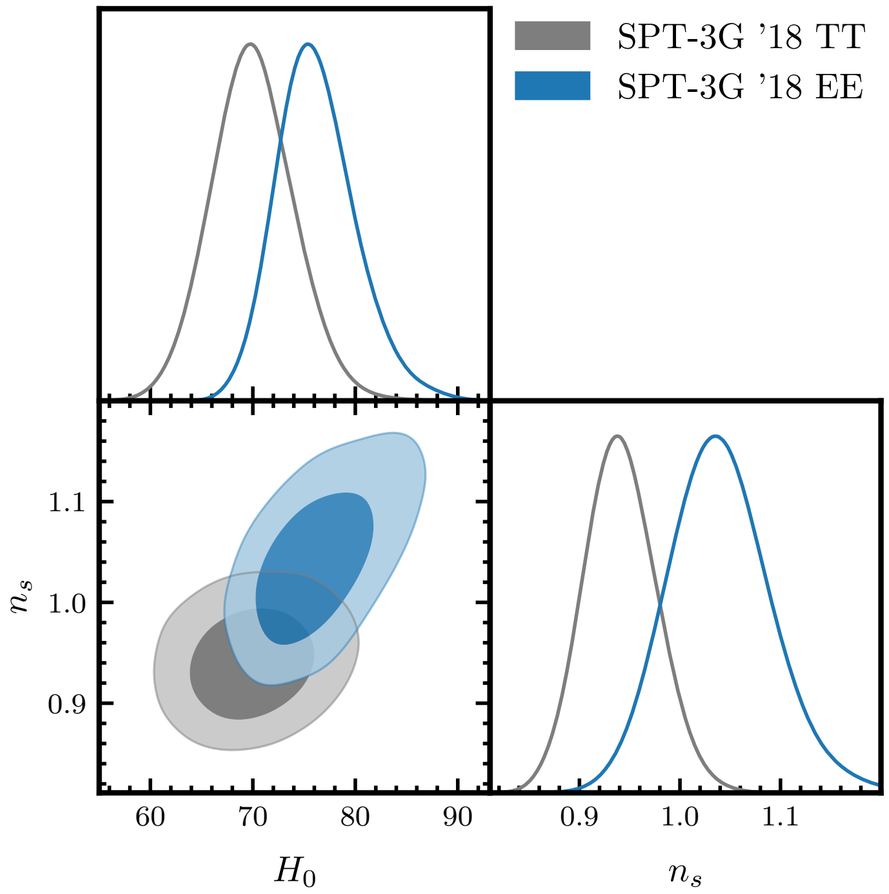
from band powers

Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!

“How correlated are constraints from different parts of the data?”



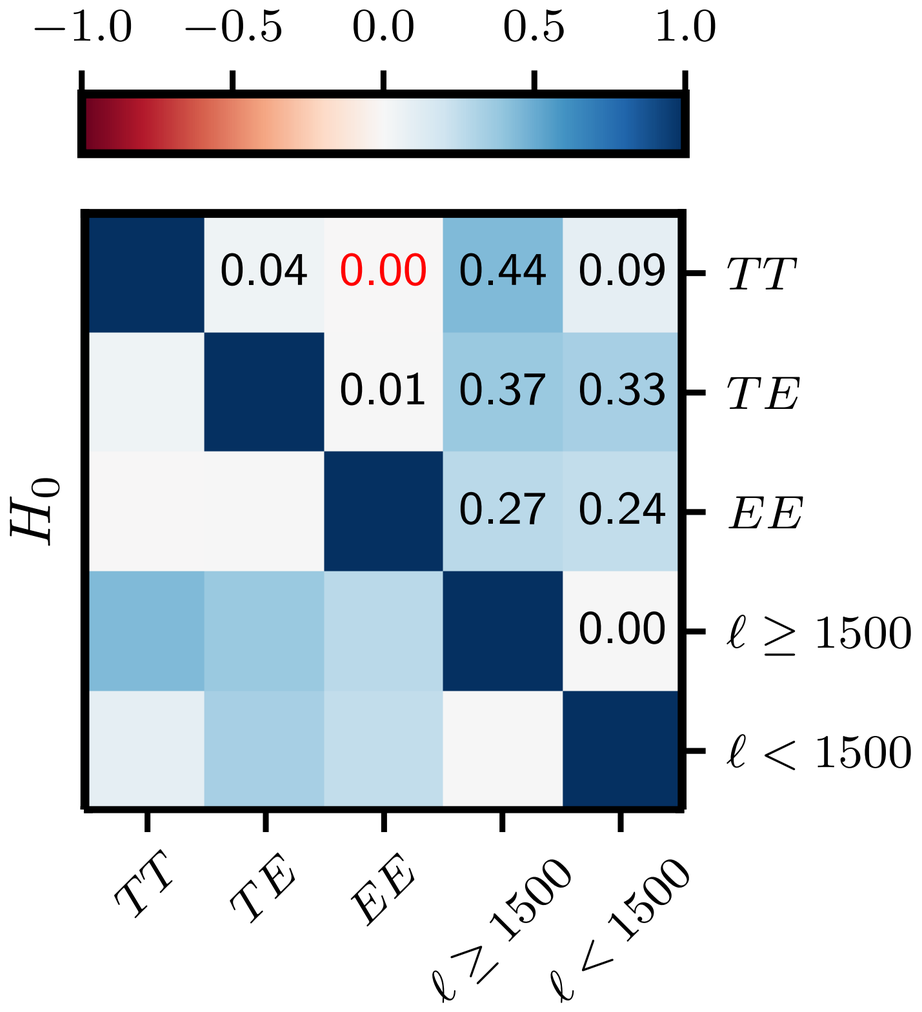
For background see Kable, Addison, Bennett 2019 (arXiv:1908.01626)

Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **<u>Quick, easy, reliable Fisher matrices:</u>**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **<u>Smart exploration of the likelihood:</u>**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!

“How correlated are constraints from different parts of the data?”

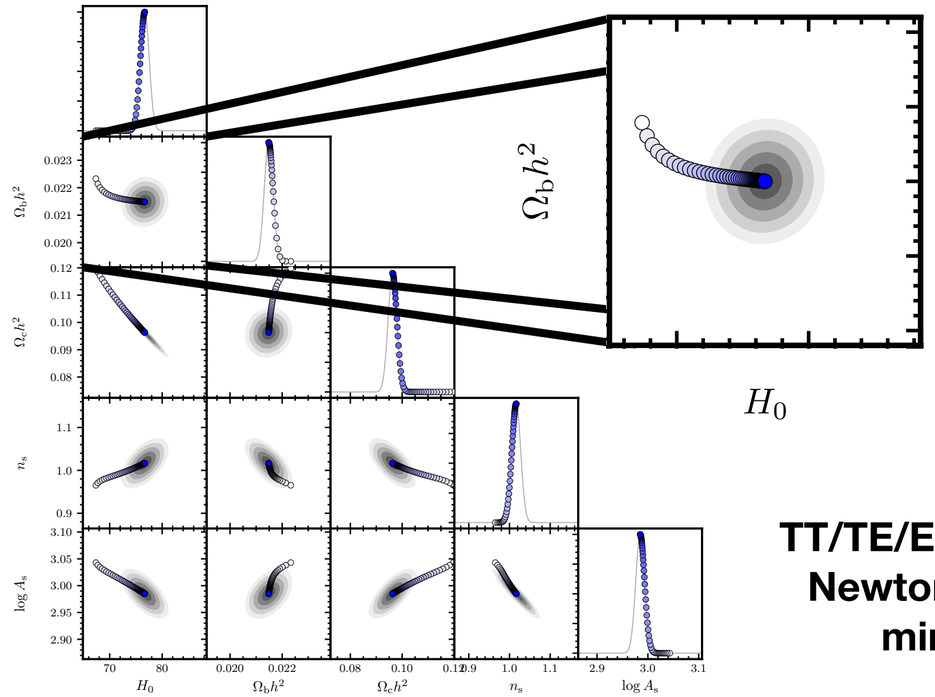


For background see Kable, Addison, Bennett 2019 (arXiv:1908.01626)

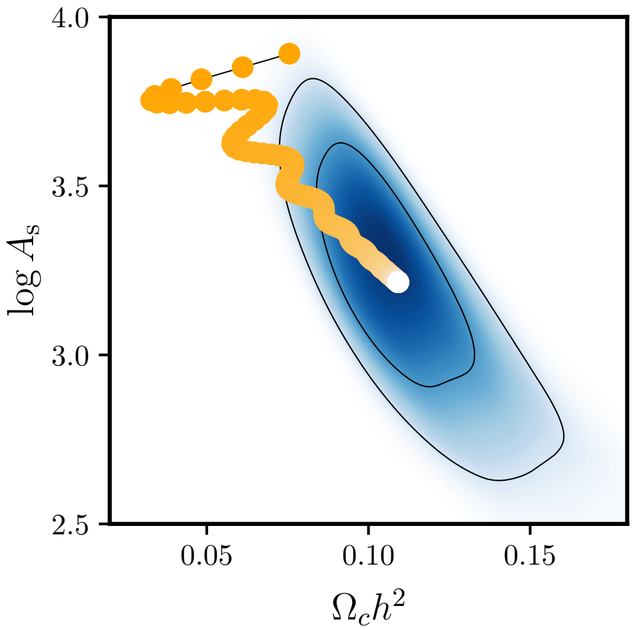
Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

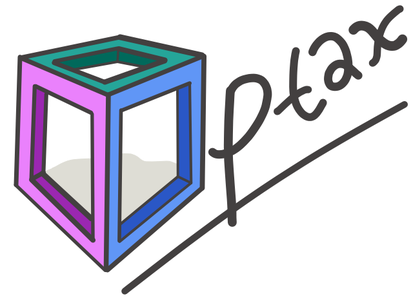
- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!



TT/TE/EE Likelihood
Newton-Raphson minimiser



ΦΦ Likelihood
ADAM minimiser
(arXiv:1412.6980)

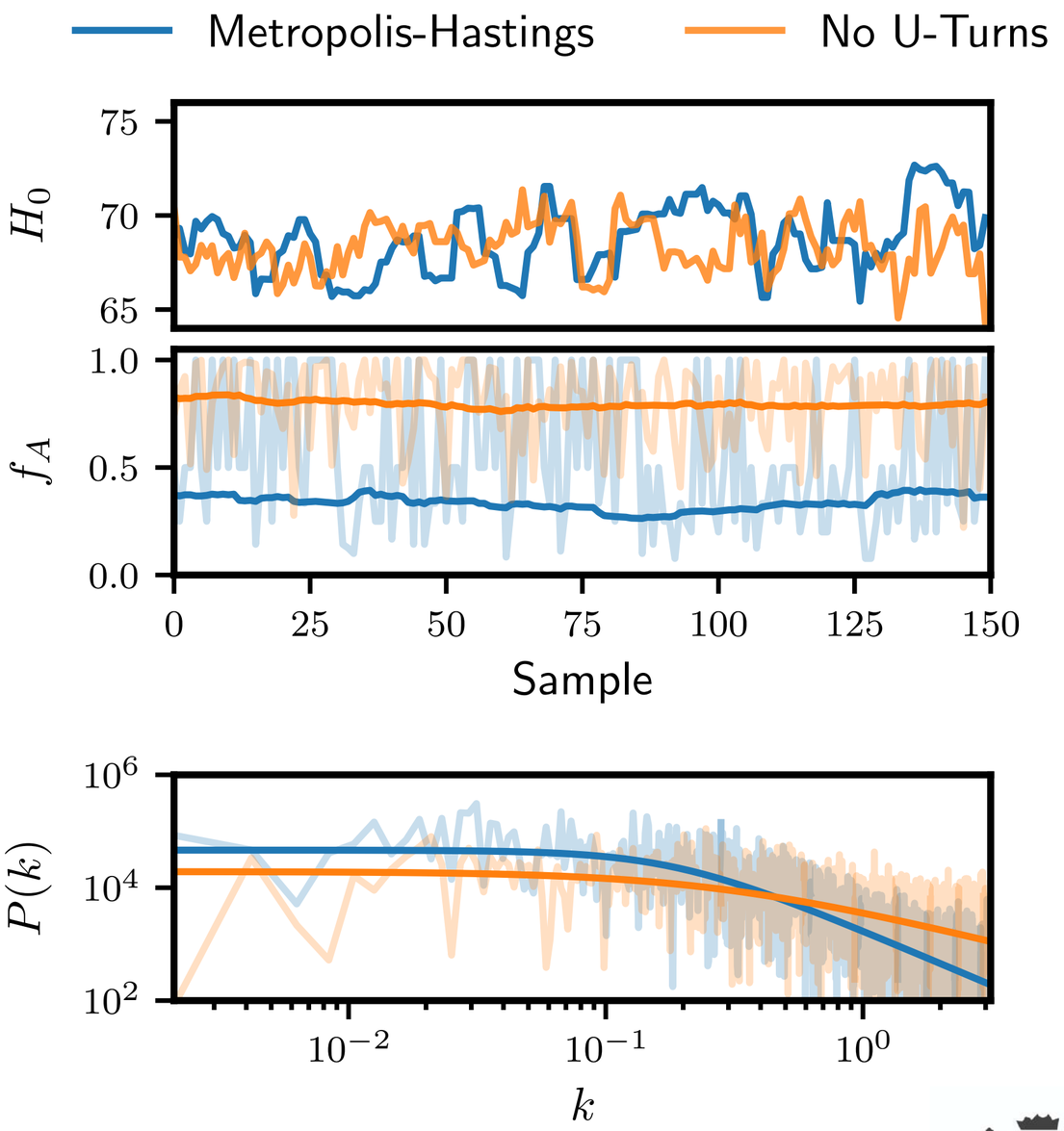


https://github.com/deepmind/optax

Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!

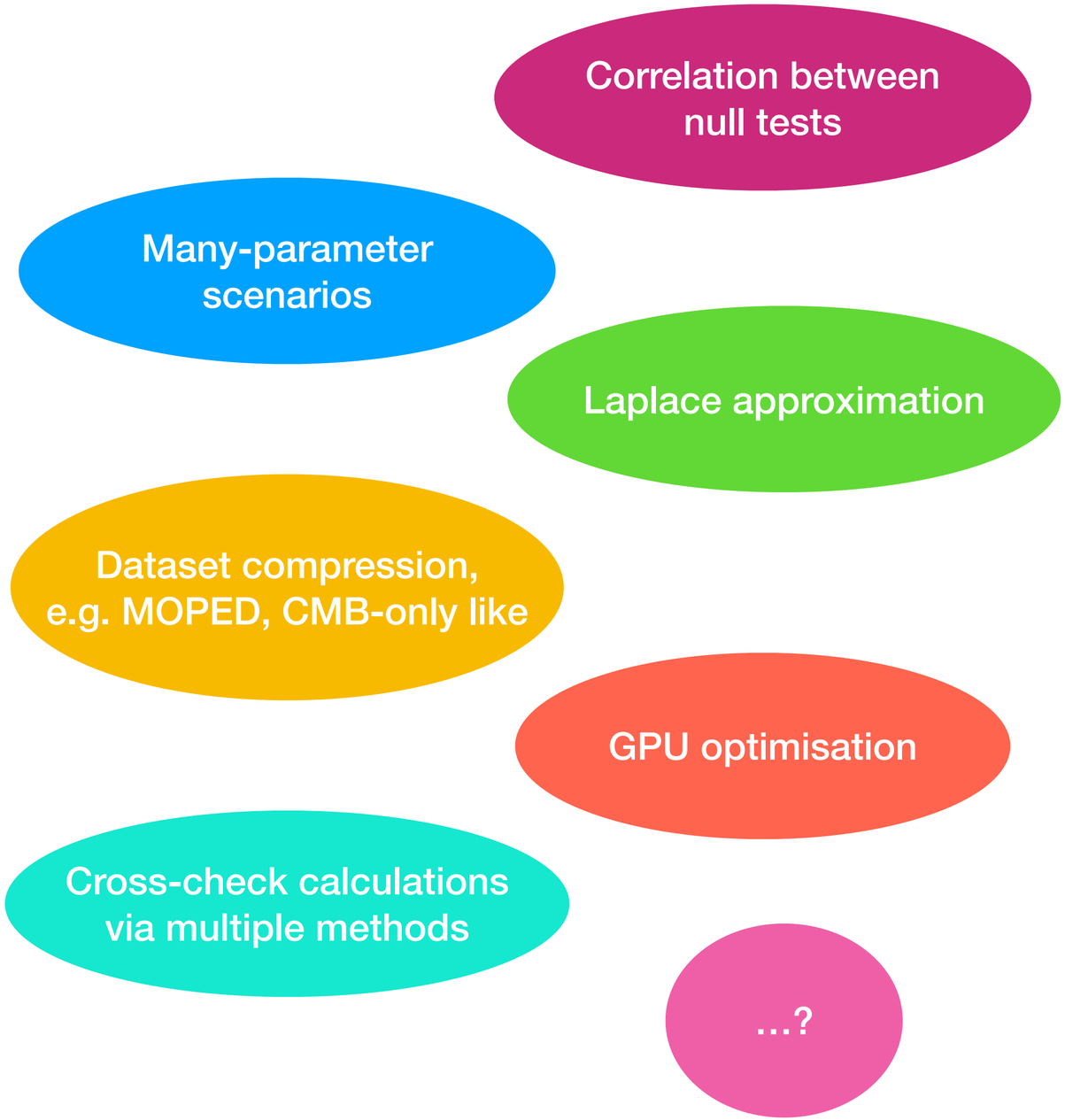


Using BlackJAX: https://github.com/blackjax-devs/blackjax

Applications

w/ CosmoPower-JAX (Piras and Spurio Mancini 2023)

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - HMC/NUTS sampling
- … and more!



candl

Getting Started

- Overview
- Data Sets
- Tutorials and Use

Components

- Likelihood Code
- Transformations
- Interface
- Auxiliary Tools
- Plot Templates

API

- candl.likelihood
- candl.interface
- candl.tools
- candl.transformations
- candl.plots
- candl.tests
- candl.data
- candl.io
- candl.constants
- candl.lib

v: latest

CMB Analysis With A Differentiable Likelihood

Authors: L. Balkenhol, C. Trendafilova, K. Benabed, S. Galli

Paper: arXiv 2401.13433

Source: Lbalkenhol/candl

Documentation: docs passing

candl is a differentiable likelihood framework for analysing CMB power spectrum measurements. Key features are:

- JAX-compatibility, allowing for fast and easy computation of gradients and Hessians of the likelihoods.
- The latest public data releases from the South Pole Telescope and Atacama Cosmology Telescope collaborations.
- Interface tools for work with other popular cosmology software packages (e.g. Cobaya and MontePython).
- Auxiliary tools for common analysis tasks (e.g. generation of mock data).

candl supports the analysis of primary CMB and lensing power spectrum data (TT, TE, EE, BB, $\phi\phi$, $\kappa\kappa$).

Installation

candl can be installed with pip:

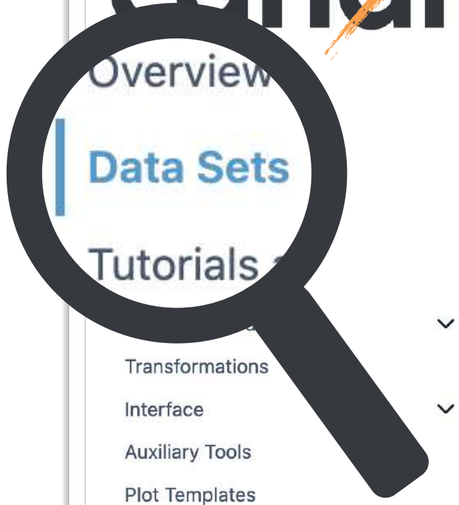
```
pip install candl-like
```

After installation, we recommend testing by executing the following python code:

```
import candl.tests
candl.tests.run_all_tests()
```

This well test all data sets included in candl.

Extensive Documentation & Tutorials Available



candl

Overview

Data Sets

Tutorials

Transformations

Interface

Auxiliary Tools

Plot Templates

API

candl.likelihood

candl.interface

candl.tools

candl.transformations

candl.plots

candl.tests

candl.data

candl.io

candl.constants

candl.lib

v: latest

CMB Analysis With A Differentiable Likelihood

Authors:
Paper:
Source:
Documentation:

candl is a differenti
are:

- JAX-compatibi
- The latest publ
collaborations.
- Interface tools
MontePython).
- Auxiliary tools

candl supports the

Installatio

candl can be install

```
pip install ca
```

After installation, w

```
import candl.te
candl.tests.ru
```

This well test all da

ACT DR4 TT/TE/EE

Paper(s): S. Aiola, E. Calabrese, L. Maurin, S. Naess, B. L. S
arXiv 2007.07288
S. K. Choi, M. Hasselfield, S.-P. P. Ho, B. Koopma
arXiv 2007.07289

Type: Primary power spectrum measurement (TT/TE
LAMBDA: NASA archive
Short cut(s): `candl.data.ACT_DR4_TTTEEE`
Latest version: `v0`

Note: This is the CMB-only, foreground marginalised version of the l
deep data as `dxd` and the wide data as `wxw`.

ACT DR6 PP

Paper(s): M. S. Madhavacheril, F. J. Qu, B. D. Sherwin, N. MacCrann, Y. Li et al. (ACT Collaboration)
arXiv 2304.05203
F. J. Qu, B. D. Sherwin, M. S. Madhavacheril, D. Han, K. T. Crowley et al. (ACT Collaboration)
arXiv 2304.05202

Type: Lensing power spectrum measurement ($\phi\phi$)
Website: Github
Short cut(s): `candl.data.ACT_DR6_Lens`, `candl.data.ACT_DR6_Lens_and_CMB`
Latest version: `v0`

Note: this data set uses the lensing power spectrum in $\kappa\kappa$. For the ACT + Planck lensing combination see also Carron, Mirmelstein, Lewis 2023. Use `candl.data.ACT_DR6_Lens` when only working with lensing data, use `candl.data.ACT_DR6_Lens_and_CMB` when combining lensing and primary CMB data.

SPT-3G 2018 TT/TE/EE

Paper(s): L. Balkenhol, D. Dutcher, A. Spurio Mancini, A. Dussot, K. Benabed, S. Galli et al. (SPT-3G Collaboration)
arXiv 2212.05642

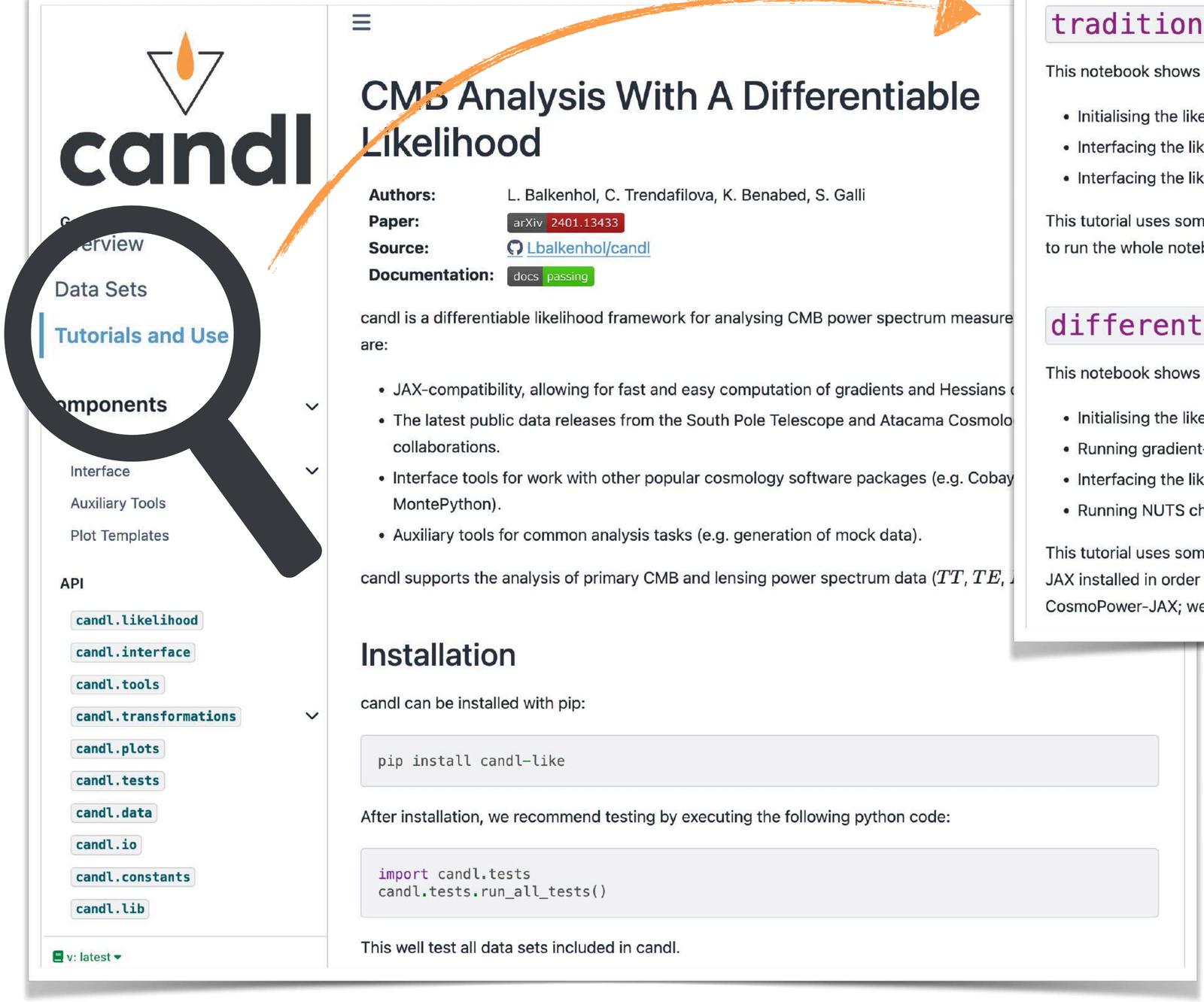
Type: Primary power spectrum measurement ($TT/TE/EE$)
Website: SPT-3G Website
LAMBDA: NASA Archive
Short cut(s): `candl.data.SPT3G_2018_TTTEEE`
Latest version: `v0`

SPT-3G 2018 PP

Paper(s): Z. Pan, F. Bianchini, W. L. K. Wu et al. (SPT-3G Collaboration)
arXiv 2308.11608

Type: Lensing power spectrum measurement ($\phi\phi$)
Short cut(s): `candl.data.SPT3G_2018_Lens`, `candl.data.SPT3G_2018_Lens_and_CMB`
Latest version: `v0`

Note: this data set uses the lensing power spectrum in $\phi\phi$. Use `candl.data.SPT3G_2018_Lens` when only working with lensing data, use `candl.data.SPT3G_2018_Lens_and_CMB` when combining lensing and primary CMB data.



`traditional_tutorial.ipynb`

This notebook shows how traditional inference tasks are accomplished. In particular:

- Initialising the likelihood and accessing the data (band powers, covariance, etc.)
- Interfacing the likelihood with CAMB and calculating the χ^2 for a given spectrum
- Interfacing the likelihood with Cobaya and running an MCMC chain

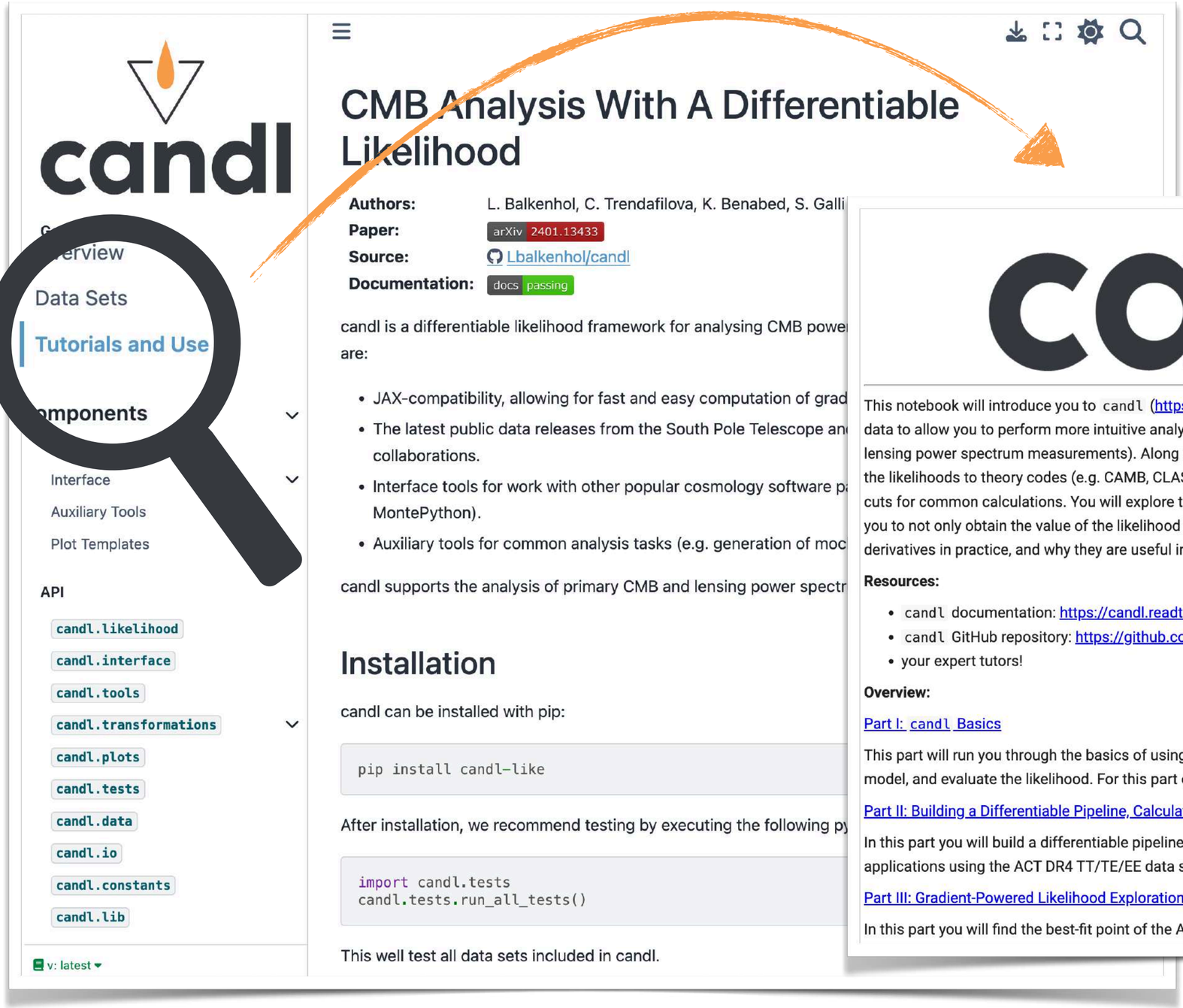
This tutorial uses some optional packages. Make sure you have Cobaya, getdist, and CAMB installed in order to run the whole notebook.

`differentiable_tutorial.ipynb`

This notebook shows different aspects relying on the differentiability of the likelihood. In particular:

- Initialising the likelihood and accessing the data (band powers, covariance, etc.)
- Running gradient-based minimisers
- Interfacing the likelihood with Optax
- Running NUTS chains by interfacing the likelihood with BlackJAX

This tutorial uses some optional packages. Make sure you have Optax, BlackJAX, getdist, and CosmoPower-JAX installed in order to run the whole notebook. You also need to have some emulator models for CosmoPower-JAX; we recommend the SPT high-accuracy models available here.



Joint SPT & ACT
CMB Analysis Summer School 2024

candl

This notebook will introduce you to `candl` (https://arxiv.org/abs/2401.13433). `candl` is a likelihood code that provides easy access to CMB data to allow you to perform more intuitive analysis. Out of the box, `candl` comes with the latest SPT and ACT data installed (primary CMB and lensing power spectrum measurements). Along with the likelihood code, `candl` features an interface module that allows you to easily connect the likelihoods to theory codes (e.g. CAMB, CLASS) and samplers (e.g. Cobaya, MontePython) as well as a tools module that provides short cuts for common calculations. You will explore these in part 1 of this notebook. Moreover, `candl` likelihoods are differentiable, i.e. they allow you to not only obtain the value of the likelihood function, but also its slope. You will learn what this means in detail, how to work with derivatives in practice, and why they are useful in parts 2 and 3 of this notebook.

Resources:

- `candl` documentation: https://candl.readthedocs.io/en/latest/
- `candl` GitHub repository: https://github.com/Lbalkenhol/candl
- your expert tutors!

Overview:

Part I: `candl` Basics

This part will run you through the basics of using `candl`: how to initialise a likelihood, access different data products, understand the data model, and evaluate the likelihood. For this part of the notebook you will be using the SPT-3G 2018 TT/TE/EE data set.

Part II: Building a Differentiable Pipeline, Calculating Derivatives

In this part you will build a differentiable pipeline from cosmological parameters all the way to the likelihood value. We will [illegible] applications using the ACT DR4 TT/TE/EE data set.

Part III: Gradient-Powered Likelihood Exploration

In this part you will find the best-fit point of the ACT DR4 TT/TE/EE data set using traditional and gradient-pow[illegible]

…email me for solutions ;)

Joint SPT & ACT
CMB Analysis Summer School 2024

candl

Overview

Data Sets

Tutorials and Use

Components

Interface

Auxiliary Tools

Plot Templates

API

- candl.likelihood
- candl.interface
- candl.tools
- candl.transformations
- candl.plots
- candl.tests
- candl.data
- candl.io
- candl.constants
- candl.lib

v: latest

CMB Analysis With A Differentiable Likelihood

Authors: L. Balkenhol, C. Trendafilova, K. Benabed, S. Galli

Paper: arXiv 2401.13433

Source: Lbalkenhol/candl

Documentation: docs passing

candl is a differentiable likelihood framework for analysing CMB power spectra. Features are:

- JAX-compatibility, allowing for fast and easy computation of gradients
- The latest public data releases from the South Pole Telescope and Atacama Cosmology Telescope collaborations.
- Interface tools for work with other popular cosmology software packages (Cobaya, MontePython).
- Auxiliary tools for common analysis tasks (e.g. generation of mock data).

candl supports the analysis of primary CMB and lensing power spectra.

Installation

candl can be installed with pip:

```
pip install candl-like
```

After installation, we recommend testing by executing the following python script:

```
import candl.tests
candl.tests.run_all_tests()
```

This well test all data sets included in candl.

PART I: `candl` Basics

Let's learn the basics of `candl`, how to access different parts of the data, evaluate the likelihood, and use transformations.

Background:

Generally, CMB likelihoods can be written as:

$$-2\log L(\theta) = \left\{\hat{D} - T\left[D^{\rm CMB}(\theta), \theta\right]\right\}^T C^{-1} \left\{\hat{D} - T\left[D^{\rm CMB}(\theta), \theta\right]\right\},$$

where L is the likelihood, θ are the (cosmological and nuisance) parameters, $D^{\rm CMB}$ is the CMB power spectrum, C the band power covariance matrix and T the data model. The data model represents a series of transformations that need to be applied to the CMB power spectrum so that it can be compared to the data; this includes accounting for e.g. foreground contamination or calibration uncertainty.

`candl` likelihoods take a dictionary populated with (cosmological and nuisance) parameters, as well as CMB spectra as an input and return the negative log-likelihood value. The data model is implemented as a series of transformations that are applied to the theory spectra in a specific order. These transformations are specified in a `.yaml` file, along with all other necessary information (such as information where to find the band powers, what kind of spectra are being supplied, etc.).

We'll start by initialising a candl likelihood for the SPT-3G 2018 TT/TE/EE data set. Programatically, `candl` likelihoods are python classes and their `log_like` function implements the equation above. Execute the cell below and inspect the output produced by the initialisation paying attention to the information on spectra and the data model.

```
[ ]  # Initialise the SPT-3G 2018 TT/TE/EE likelihood
     candl_like = candl.Like(candl.data.SPT3G_2018_TTTEEE)
```

Data Access

`candl` likelihoods have a lot of useful attributes that allow us to directly access different parts of the data set. Let's ge… information that is available.

Exercise:

…email me for solutions ;)

Joint SPT & ACT
CMB Analysis Summer School 2024

candl

Overview

Data Sets

Tutorials and Use

Components

Interface

Auxiliary Tools

Plot Templates

API

- candl.likelihood
- candl.interface
- candl.tools
- candl.transformations
- candl.plots
- candl.tests
- candl.data
- candl.io
- candl.constants
- candl.lib

v: latest

CMB Analysis With A Differentiable Likelihood

Authors: L. Balkenhol, C. Trendafilova, K. Benabed, S. Galli

Paper: arXiv 2401.13433

Source: Lbalkenhol/candl

Documentation: docs passing

candl is a differentiable likelihood framework for analysing CMB power spectra. Key features are:

- JAX-compatibility, allowing for fast and easy computation of gradients
- The latest public data releases from the South Pole Telescope and Atacama Cosmology Telescope collaborations.
- Interface tools for work with other popular cosmology software packages (Cobaya, MontePython).
- Auxiliary tools for common analysis tasks (e.g. generation of mock data).

candl supports the analysis of primary CMB and lensing power spectra.

Installation

candl can be installed with pip:

```
pip install candl-like
```

After installation, we recommend testing by executing the following python script:

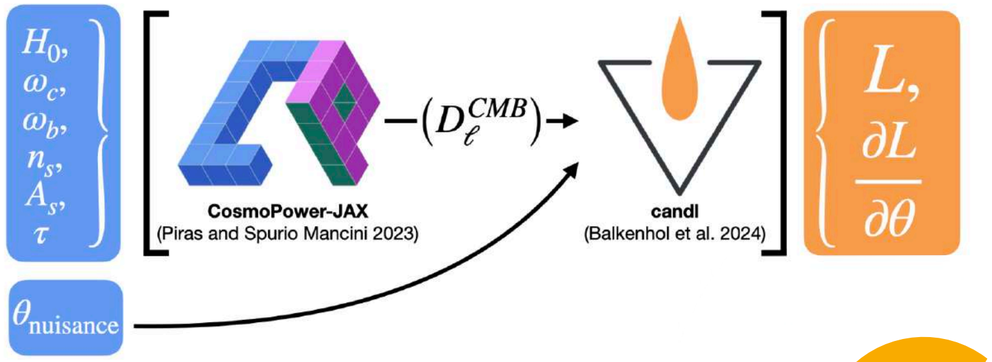
```
import candl.tests
candl.tests.run_all_tests()
```

This well test all data sets included in candl.

Automatic differentiation, or "auto-diff", is a computer science concept that is seeing more and more applications in scientific computing and in particular cosmology (see e.g. https://arxiv.org/abs/2302.05163). It is a way to obtain derivatives of functions you write with respect to their input parameters - without resorting to finite difference methods. How auto-diff works is beyond the scope of this school, though if you are interested take a look at https://jax.readthedocs.io/en/latest/automatic-differentiation.html.

`CosmoPower` (https://arxiv.org/abs/2106.03846) is a framework for neural network-based emulators of the CMB power spectrum. You can think of it as a short-cut for evaluating Boltzmann solvers, such as CAMB. `CosmoPower` models are trained on a series of pre-calculated CMB spectra and perform effectively an interpolation between the known points in order to avoid slow CAMB evalutations. Not only do they provide a speed-up, but `CosmoPower` models are also differentiable. Below, we will use `CosmoPower-JAX` (https://arxiv.org/abs/2305.06347) to stay within the same code eco-system and use models trained on high-precision CAMB spectra for the SPT-3G 2018 TT/TE/EE analysis (https://arxiv.org/abs/2212.05642, https://github.com/alessiospuriomancini/cosmopower/tree/main/cosmopower/trained_models/SPT_high_accuracy).

The diagram belows how the different codes work together to create a pipeline that is differentiable. We start out on the left with our input parameters: some cosmological (top blue box) and some nuisance parameters (bottom blue box). The cosmological parameters are fed into `CosmoPower`, which produces a CMB power spectrum (D_ℓ^{CMB}). This power spectrum in return is then passed to `candl` along with any nuisance parameters and `candl` calculates the likelihood value, L. Since all codes in the black brackets are differentiable, we can also calculate the derivate of the likelihood with respect to any of the input parameters (cosmological and nuisance), $\partial L/\partial\theta$.



...email me for solutions ;)

Joint SPT & ACT
CMB Analysis Summer School 2024

candl

Overview
Data Sets
Tutorials and Use

Components
Interface
Auxiliary Tools
Plot Templates

API
candl.likelihood
candl.interface
candl.tools
candl.transformations
candl.plots
candl.tests
candl.data
candl.io
candl.constants
candl.lib

v: latest

CMB Analysis With A Differentiable Likelihood

Authors: L. Balkenhol, C. Trendafilova, K. Benabed, S. Galli
Paper: arXiv 2401.13433
Source: Lbalkenhol/candl
Documentation: docs passing

candl is a differentiable likelihood framework for analysing CMB power
are:

- JAX-compatibility, allowing for fast and easy computation of grad
- The latest public data releases from the South Pole Telescope an
collaborations.
- Interface tools for work with other popular cosmology software p
MontePython).
- Auxiliary tools for common analysis tasks (e.g. generation of moc

candl supports the analysis of primary CMB and lensing power spectr

Installation

candl can be installed with pip:

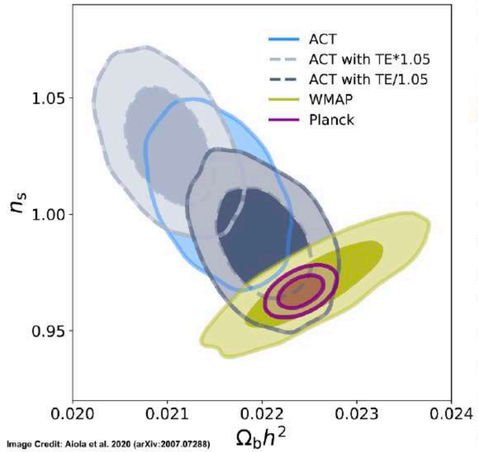
```
pip install candl-like
```

After installation, we recommend testing by executing the following py

```
import candl.tests
candl.tests.run_all_tests()
```

This well test all data sets included in candl.

ACT DR4 TE Artificial Recalibration



Background:

The image on the left shows constraints placed by the ACT DR4 TT/TE/EE data in $\Lambda\mathrm{CDM}$ in the $n_s - \Omega_b h^2$ plane (blue contours) (taken from https://arxiv.org/abs/2007.07288). The authors note a preference for lower $\Omega_b h^2$ and higher n_s data compared to Planck and WMAP results (purple and yellow/green contours, respectively).

The size of the difference of the ACT results w.r.t. Planck is compatible with a statistical fluctuation. Furthermore, the authors see no evidence for such a re-calibration of their TE power spectrum in their data. Nevertheless, they show how their results shift when artificially multiplying their TE band powers by 5% (light and dark blue contours) and can in such a way be brought to a closer agreement with the satellite data.

In order to produce these results, new MCMC runs with the TE power spectrum artifically multiplied by 0.95 and 1.05 are typically required. This is computationally expensive, especially bearing in mind that there may be multiple effects at the band power level that project along the same axis in parameter space. Without a good intuition, you can find yourself applying many different offsets to the band powers with varying amplitudes and running an MCMC analysis for each case.

With a differentiable pipeline, you can investigate how band power biases project to parameter space more efficiently. Consider the following formula (https://arxiv.org/abs/1908.01626):

$$\Delta\theta = F^{-1} \frac{\partial D_\ell}{\partial \theta} C^{-1} \Delta \hat{D_\ell},$$

where $\Delta\theta$ is the shift to best-fit parameters caused by a displacement $\Delta\hat{D_\ell}$ of the band powers, C the band power covariance matrix, F is the Fisher matrix and $\partial D_\ell / \partial \theta$ the derivative of the model spectra with respect to parameters.

Exercise:

Create your own version of the plot above, focussing only on the ACT data (forgetting about WMAP and Planck)

...email me for solutions ;)

Joint SPT & ACT
CMB Analysis Summer School 2024

candl

Overview

Data Sets

Tutorials and Use

Components

Interface

Auxiliary Tools

Plot Templates

API

- candl.likelihood
- candl.interface
- candl.tools
- candl.transformations
- candl.plots
- candl.tests
- candl.data
- candl.io
- candl.constants
- candl.lib

v: latest

CMB Analysis With A Differentiable Likelihood

Authors: L. Balkenhol, C. Trendafilova, K. Benabed, S. Galli

Paper: arXiv 2401.13433

Source: Lbalkenhol/candl

Documentation: docs passing

candl is a differentiable likelihood framework for analysing CMB power
are:

- JAX-compatibility, allowing for fast and easy computation of grad
- The latest public data releases from the South Pole Telescope an
collaborations.
- Interface tools for work with other popular cosmology software p
MontePython).
- Auxiliary tools for common analysis tasks (e.g. generation of moc

candl supports the analysis of primary CMB and lensing power spectr

Installation

candl can be installed with pip:

```
pip install candl-like
```

After installation, we recommend testing by executing the following py

```
import candl.tests
candl.tests.run_all_tests()
```

This well test all data sets included in candl.

PART III: Gradient-Powered Likelihood Exploration

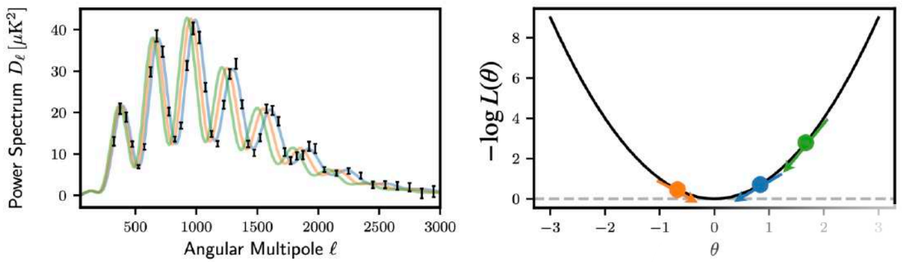
We will now see how gradient information can be used to explore the likelihood surface. In particular, we will code a "traditional" and a gradient-based algorithm to find the best-fit point of the data set and compare the performance of the two algorithms.

Background:

Finding the best-fit point of the data amounts to finding the global extremum (that is not at infinity) of the likelihood function. Conventions on the sign of the likelihood differ. Some codes, such as `candl`, return the log likelihood, which is typically negative and we want to find the maximum. However, we conventionally speak about **minimisation** when we want to find the best-fit point of the data.

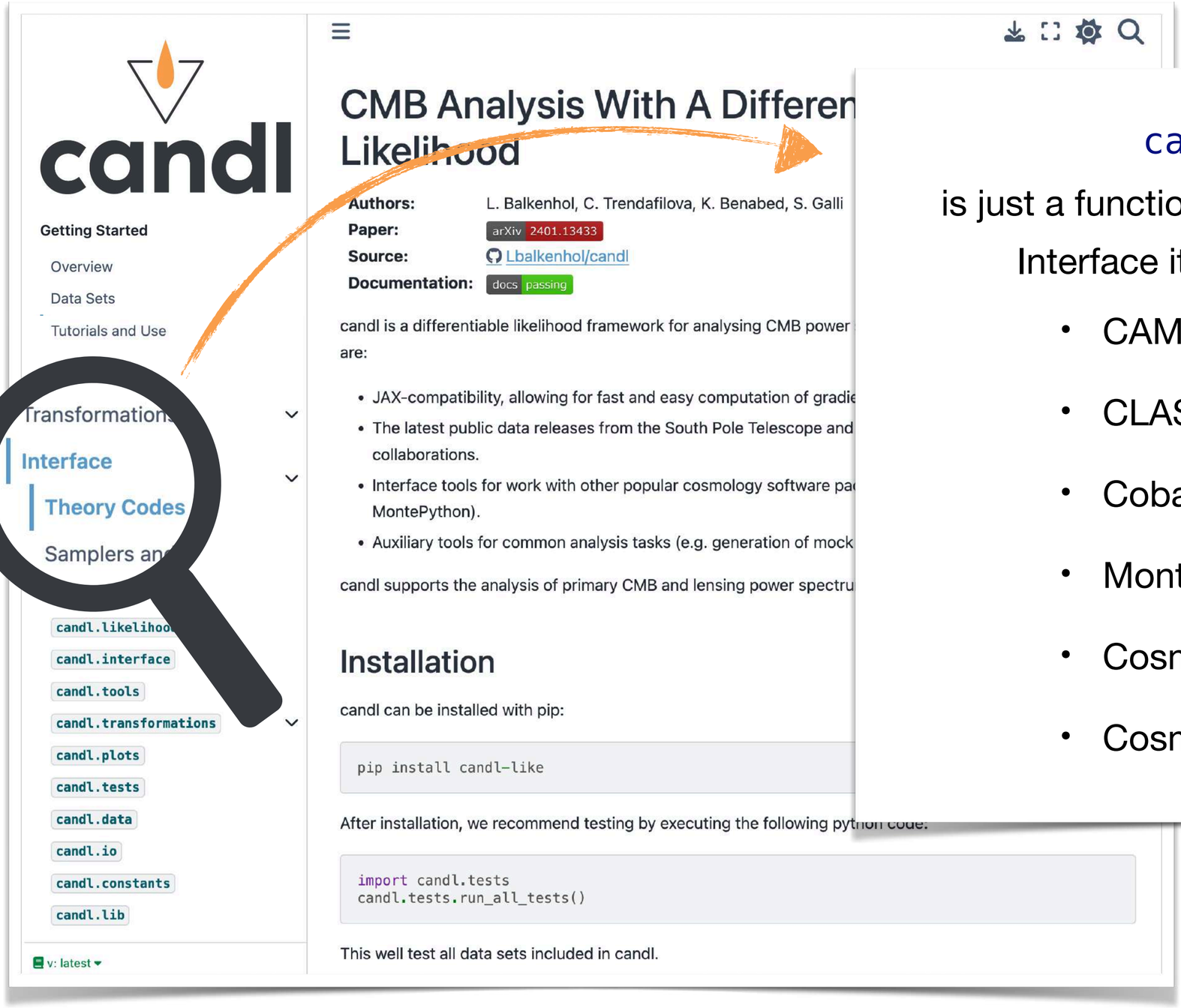
In principle, the likelihood surface can take a complicated shape that can be difficult to explore. However, modern primary CMB data has become quite constraining, such that for ΛCMB we can typically approximate it as a (multivariate) Gaussian. Hence, in the algorithms below we will assume that the likelihood surface has a single peak, which we want to get to as quickly as possible.

The diagram below illustrates the idea. Imagine our likelihood compares our band powers (black points, left panel) to some model predictions (coloured lines, left panel) as we vary some cosmological parameter θ. The (negative) log-likelihood is shown as a function of this parameter θ in the right panel. The best-fit point is the minimum of this function, and the goal of our minimisation algorithms will be to find it. If we only access the likelihood value, our information is limited to the function values (indicated by the coloured points in the right panel). However, if we also have gradient information we know the slope too (indicated by the coloured arrows in the right panel).



We will first write a minimisation algorithm that does not use gradient information, which we will run with CAMB and
codes. We will then develop an algorithm that uses gradient information and compare the performance of the di

...email me for solutions ;)



`candl_like.log_like()`

is just a function with a simple dictionary/vector input.

Interface it with whatever tools you like, e.g.

- CAMB
- CLASS
- Cobaya
- MontePython
- CosmoSIS
- CosmoPower
- Capse
- Optax
- scipy.minimize
- BlackJAX
- flowMC
- OLÉ

Differentiable Planck Likelihood

Sneak Peak

- **Full plik 2018 Likelihood wrapper**
  - TT: $30 \leq \ell < 2500$
  - TE/EE: $30 \leq \ell < 2000$
  - Including all foregrounds and nuisance parameters
  - Arbitrary data selection (eg TT only, $\ell > 1000$ only, …)
  - Already validated against official Planck likelihood
- Near future: low ell, full candl functionality, plik lite

K. Benabed (IAP)

COMING SOON*

Conclusions

candl

- Upcoming CMB data need fast, efficient, robust tools
- This is an exciting time for CMB parameter inference
  - Lots of development and excitement, many techniques are becoming viable
- **candl** is a python-based stand-alone, CMB likelihood, with (optional) JAX-powered differentiability

```
> pip install candl-like
```

https://github.com/Lbalkenhol/candl https://candl.readthedocs.io/en/latest/ https://pypi.org/project/candl-like/

Lennart Balkenhol, IAP, lennart.balkenhol@iap.fr

BACKUP SLIDES

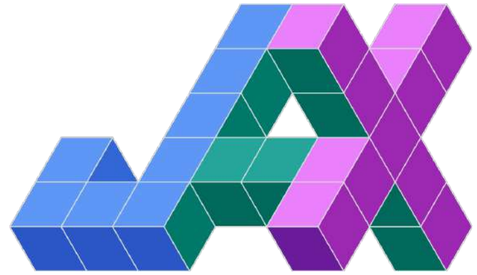
CosmoPower-JAX

- **Piras and Spurio Mancini 2023, CosmoPower-JAX, arXiv:2305.06347**
- Nerual-network based power spectrum emulator
- Written in JAX, compatible with vanilla CosmoPower models.
- High-accuracy models exist for:

$$\Lambda\mathrm{CDM}$$
$$\Lambda\mathrm{CDM} + A_\mathrm{L}$$
$$\Lambda\mathrm{CDM} + N_\mathrm{eff}$$
$$\Lambda\mathrm{CDM} + \Sigma m_\nu$$
$$w\mathrm{CDM}$$

https://github.com/alessiospuriomancini/cosmopower/tree/main/cosmopower/trained_models/SPT_high_accuracy
https://github.com/cosmopower-organization

What is JAX?



- *“JAX is NumPy (…) with great **automatic differentiation** for high-performance machine learning research”* [1]
- Google-developed Python library with:
  - Just-in-time compilation
  - CPU, GPU, and TPU optimisation
  - **Automatic differentiation**
  - Automatic vectorisation
  - Numpy API compliance

```python
import jax
def f(x):
    # super complicated function
    return x**2.0

dfdx = jax.grad(f)
# dfdx(x) gives 2.0*x
```

- See Campagne et al. 2023 (arXiv:2302.05163) for cosmology-specific information

[1] https://jax.readthedocs.io/en/latest/index.html

What is Autodiff?

- It is **NOT**:
  - Manual differentiation (deriviates it by hand)
  - Numerical differentiation (finite differences)
  - Symbolic differentiation (mathematica, sympy)
- What does Autodiff do?
  - Functions are combinations of basic operations -> known derivatives, apply chain rule
  - Goes through the implementation of the function and grabs intermediate variables
  - Helpful analogy: network graphs

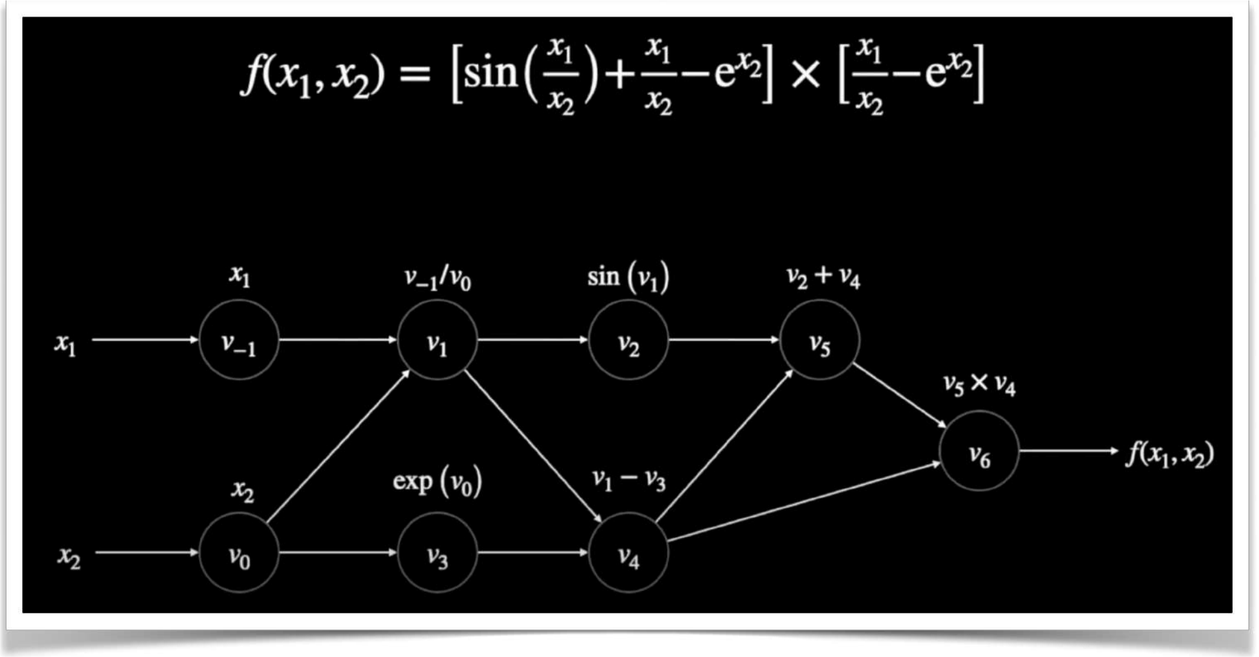


Image credit, and condensing info from JAX-endorsed video: https://www.youtube.com/watch?v=wG_nF1awSSY

JAX Autodiff vs Finite Differences

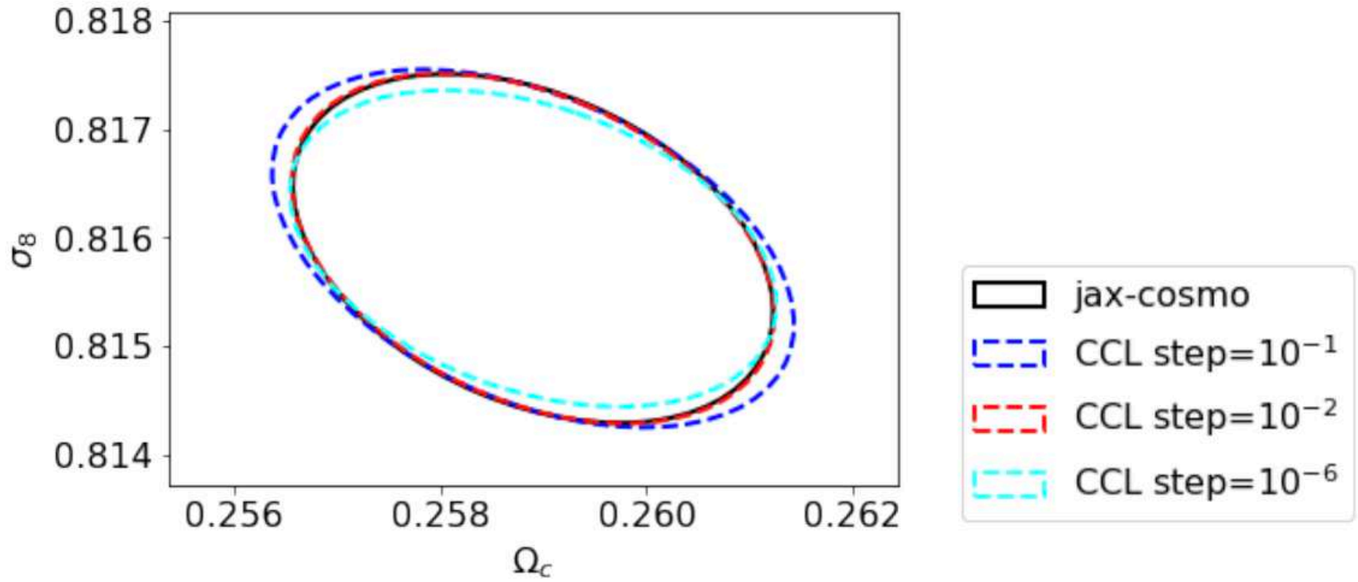


Figure 5. Fisher matrices (Eq.12) estimated using the CCL for the angular power spectra estimation and finite difference method to get the Jacobian with different spacing of parameters (10^{-6}, 10^{-2}, 10^{-1}). For comparison, the contour obtained with `jax-cosmo` is reproduced from Figure 4 in black.

Campagne et al. 2023, Fig. 5

[1] https://jax.readthedocs.io/en/latest/index.html
[2] https://arxiv.org/abs/2302.05163

Differentiable Theory Codes

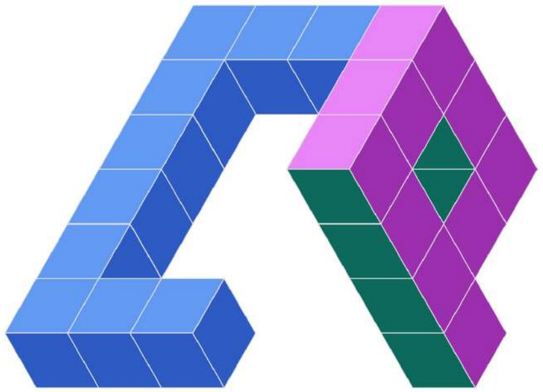
- **Piras and Spurio Mancini 2023, CosmoPower-JAX, arXiv:2305.06347**
  - Nerual-network based power spectrum emulator, high acc. models exist for*$\Lambda\mathrm{CDM}$, $\Lambda\mathrm{CDM} + A_\mathrm{L}$, $\Lambda\mathrm{CDM} + N_\mathrm{eff}$, $\Lambda\mathrm{CDM} + \Sigma m_\nu$, $w\mathrm{CDM}$
- **Bonici, Bianchini, Ruiz-Zapatero, Capse, arXiv:2307.14339**
  - Nerual-network based power spectrum emulator, written in Julia, python wrapper available
- ~~**COSMICNET, Albers+ arXiv:1907.05764, Günther+ arXiv:2207.05707**~~
  - ~~Emulate the most expensive operations of a Boltzmann solver, preserve flexibility~~
- **Hahn, List, Porqueres 2024, DISCO-DJ, arXiv:2311.03291**
  - Differentiable Boltzman solver written in JAX (but no CMB spectra yet…)
  - See also: Bolt.jl (Li and Sullivan in prep.) and LimberJack.jl (arXiv:2310.08306) in Julia
- **OLÉ (Günther et al. in prep.) Günther 2023, arXiv:2307.01138**
  - Train emulator “on-the-fly” with <200 CLASS calls, applied in Khalifeh+ arXiv:2312.09814

* https://github.com/alessiospuriomancini/cosmopower/tree/main/cosmopower/trained_models/SPT_high_accuracy
https://github.com/cosmopower-organization

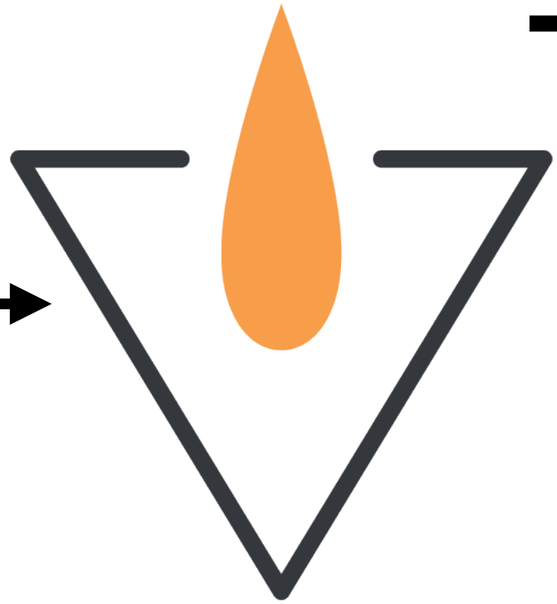
Differentiable

candl

H_0,
ω_c,
ω_b,
n_s,
A_s,
τ,
...



$-\left(D_\ell^{CMB}\right)\rightarrow$



$\rightarrow \chi^2$

CosmoPower-JAX
(Piras and Spurio Mancini 2023)

```
import candl.tools
cp_emus = {"TT": "cmb_spt_TT_NN", "TE": "cmb_spt_TE_PCAplusNN", "EE": "cmb_spt_EE_NN"}
pars_to_theory_specs = candl.interface.get_CosmoPowerJAX_pars_to_theory_specs_func(cp_emus)
pars_to_chisq = candl.tools.get_params_to_chi_square_func(candl_like, pars_to_theory_specs)
pars_to_chisq_deriv = jax.jacrev(pars_to_chisq)
```

Applications

- **Fisher forecasting made easy:**
  - E.g. optimal band power bin width
- **Other applications:**
  - Gradient-based minimisers/samplers
  - Correlation between subsets

$$F = \frac{\partial D_\ell}{\partial \theta}^T C^{-1} \frac{\partial D_\ell}{\partial \theta}$$

```
candl.tools.get_fisher_matrix()
```

$$F = -H(\mathscr{L})$$

```
jax.hessian(like)
```

Differentiable Planck Likelihood

Sneak Peak

- **Full plik 2018 Likelihood wrapper**
  - TT: $30 \leq \ell < 2500$
  - TE/EE: $30 \leq \ell < 2000$
  - Including all foregrounds and nuisance parameters
  - Arbitrary data selection (eg TT only, $\ell > 1000$ only, …)
  - Already validated against official Planck likelihood
- Near future: low ell, full candl functionality, plik lite

K. Benabed (IAP)