# candl

## Forecasting made easy

and other applications of a differentiable CMB likelihood

L. Balkenhol
with C. Trendafilova, K. Benabed, S. Galli
CMB-S4 Remote Collaboration Meeting, 26/03/2024

SORBONNE UNIVERSITÉ

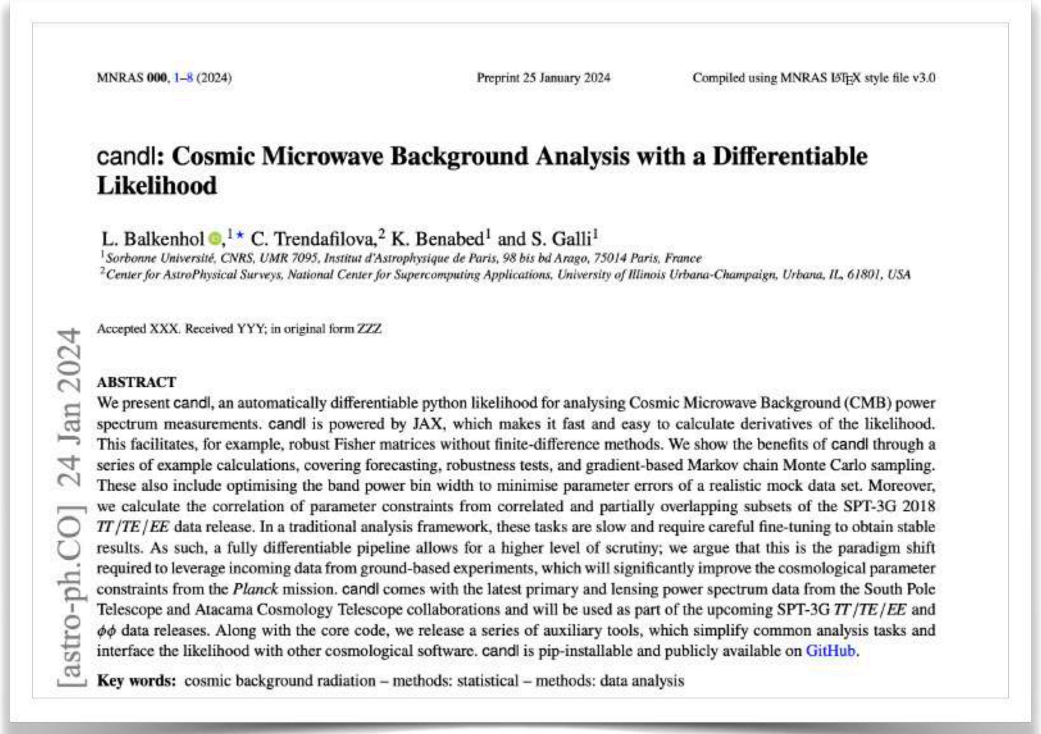
European Research Council
Established by the European Commission

# Many Thanks To My Collaborators!

*C. Trendafilova (UIUC)*

*K. Benabed (IAP)*

*S. Galli (IAP)*



MNRAS **000**, 1–8 (2024) Preprint 25 January 2024 Compiled using MNRAS LaTeX style file v3.0

**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*
[2] *Center for AstroPhysical Surveys, National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA*

Accepted XXX. Received YYY; in original form ZZZ



**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.

**Key words:** cosmic background radiation – methods: statistical – methods: data analysis

*arXiv:2401.13433, accepted 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:***

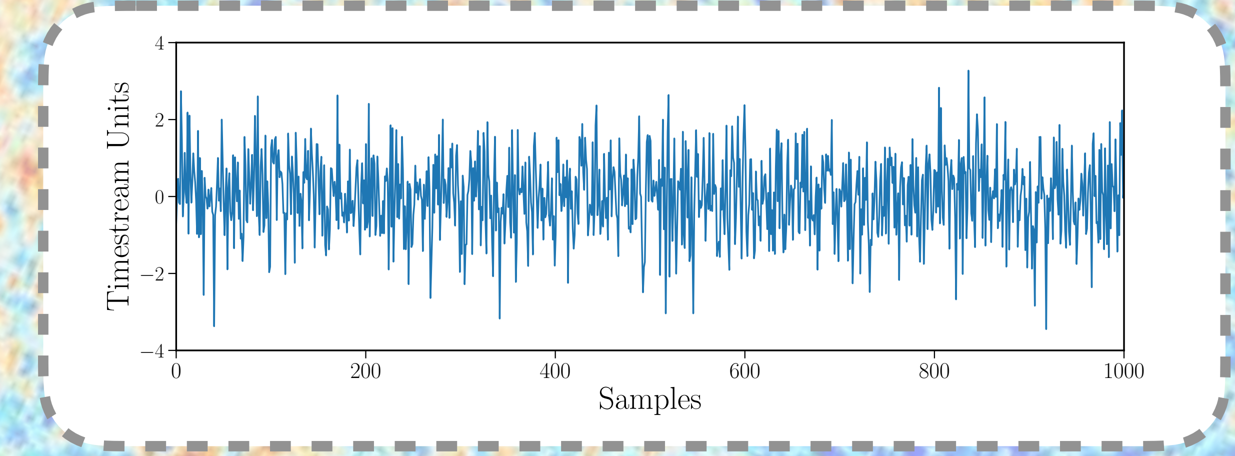
1. CMB Analysis
2. JAX, Automatic Differentiation
3. Application of a Differentiable Likelihood
4. Conclusions

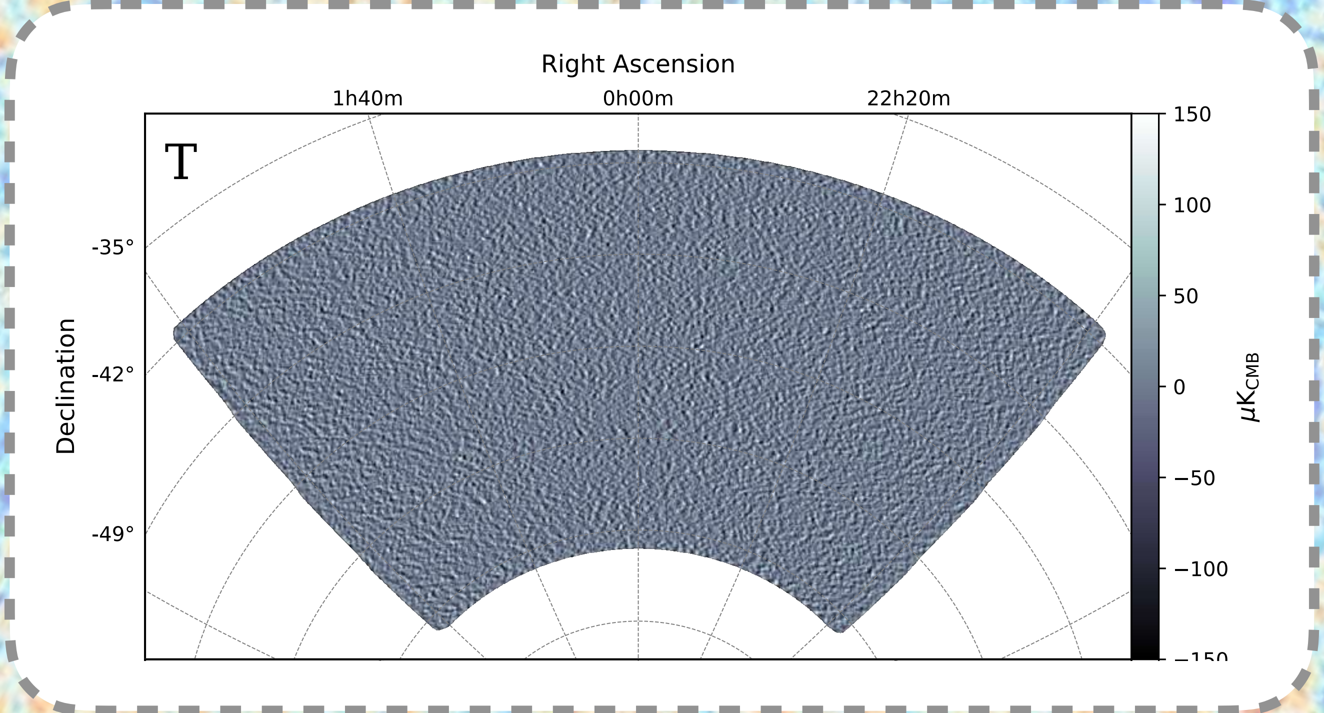
***Takeaways:***

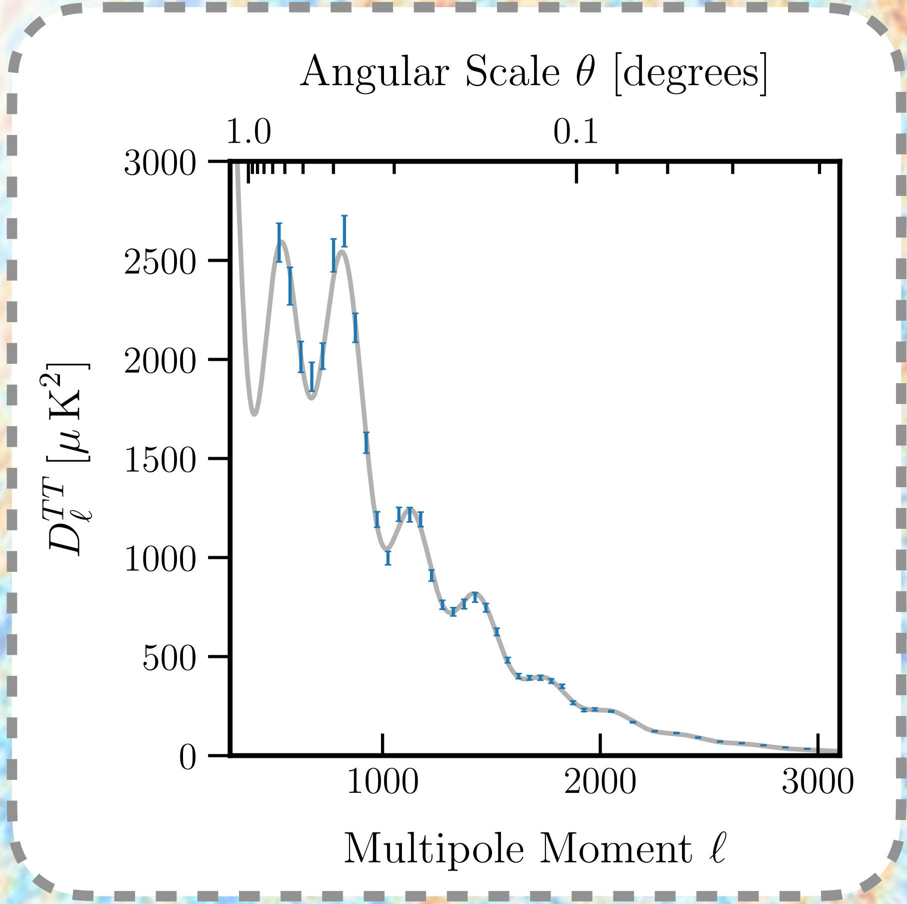
1. Forecasting in $\Lambda$CDM (and simple extensions) made fast and robust
2. Accessible python likelihood, easy access to data, comparison with SPT, ACT

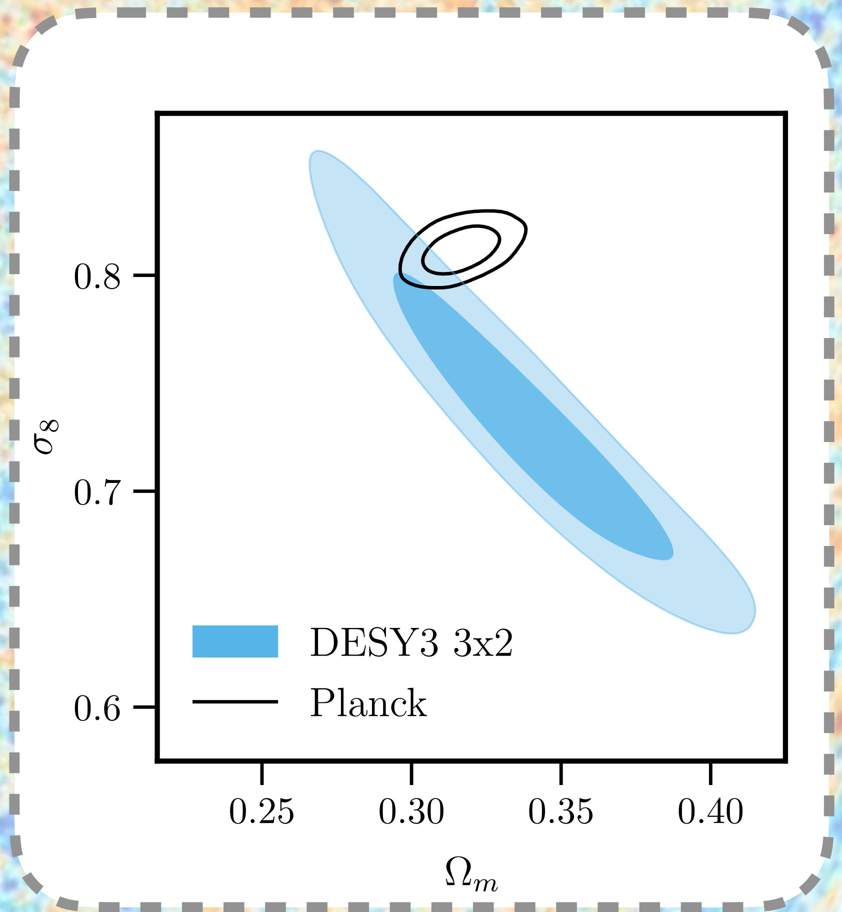
# CMB Analysis

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



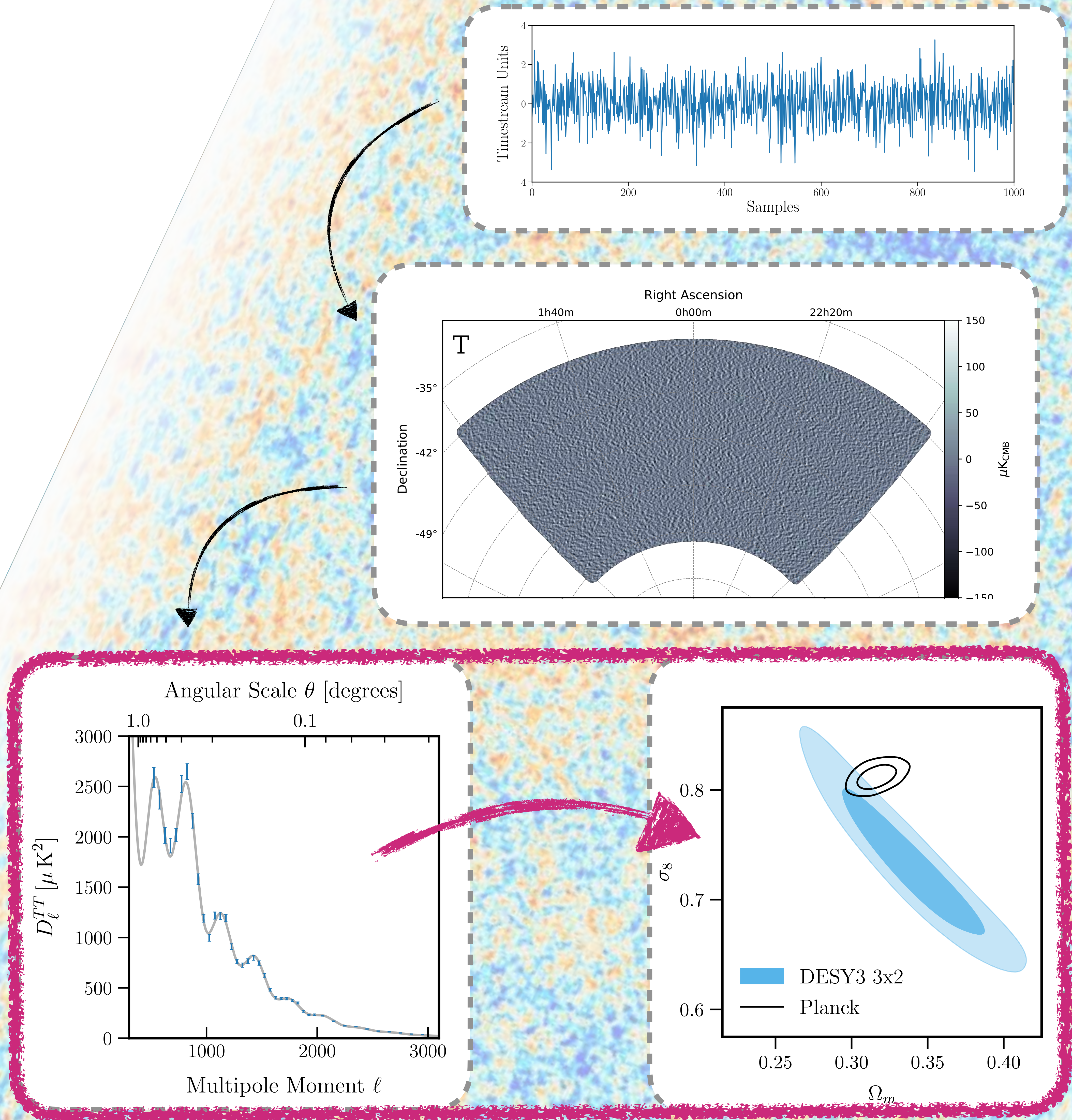



# CMB Analysis

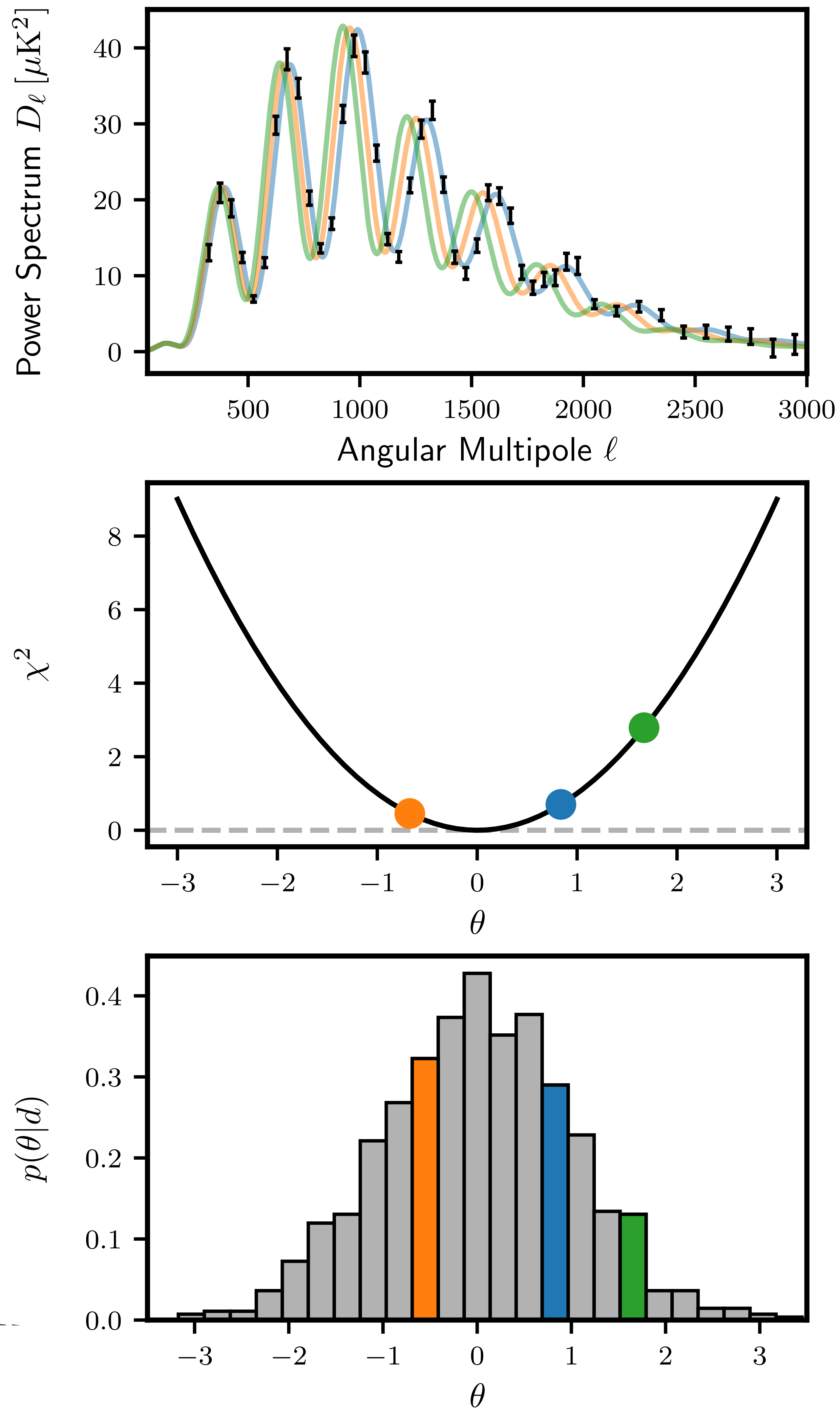
1) Telescope records time-ordered-data
2) Turn time-ordered-data into maps
3) Calculate power spectrum
4) Produce cosmological constraints
5) Interpret Results



# CMB Likelihood Analysis
## The Traditional Approach

- Boltzmann Code - supplies CMB theory spectra
    - CAMB (arXiv:9911177), CLASS (arXiv:1408.4788)
- 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), Cobaya (arXiv:2005.05290), MontePython (arXiv:1804.07261)

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



# New CMB Data Deserves Better

- Ground-based CMB experiments (SPT-3G, ACT, SO, S4) pushing past Planck precision

***<u>"With great data comes great responsibility"</u>***

- **<u>Need</u>**: 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
- **<u>Opportunity:</u>**
  - Field is transitioning to Python
  - Field is embracing differentiable emulators: CosmoPower (arXiv:2106.03846), Capse (arXiv:2307.14339), Günther 2023 (arXiv:2307.01138), …
  - Increased use of JAX: Campagne et al. 2023 (arXiv:2302.05163), Piras & Spurio Mancini 2023 (arXiv:2305.06347)

# 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

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

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

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

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, …

# Applications of a Fully Differentiable Pipeline

$H_0, \omega_c, \omega_b, n_s, A_s, \tau, \ldots$

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

**CosmoPower-JAX**
(Piras and Spurio Mancini 2023)

**candl**
(Balkenhol et al. 2024)

$\rightarrow \chi^2$

# Applications

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

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

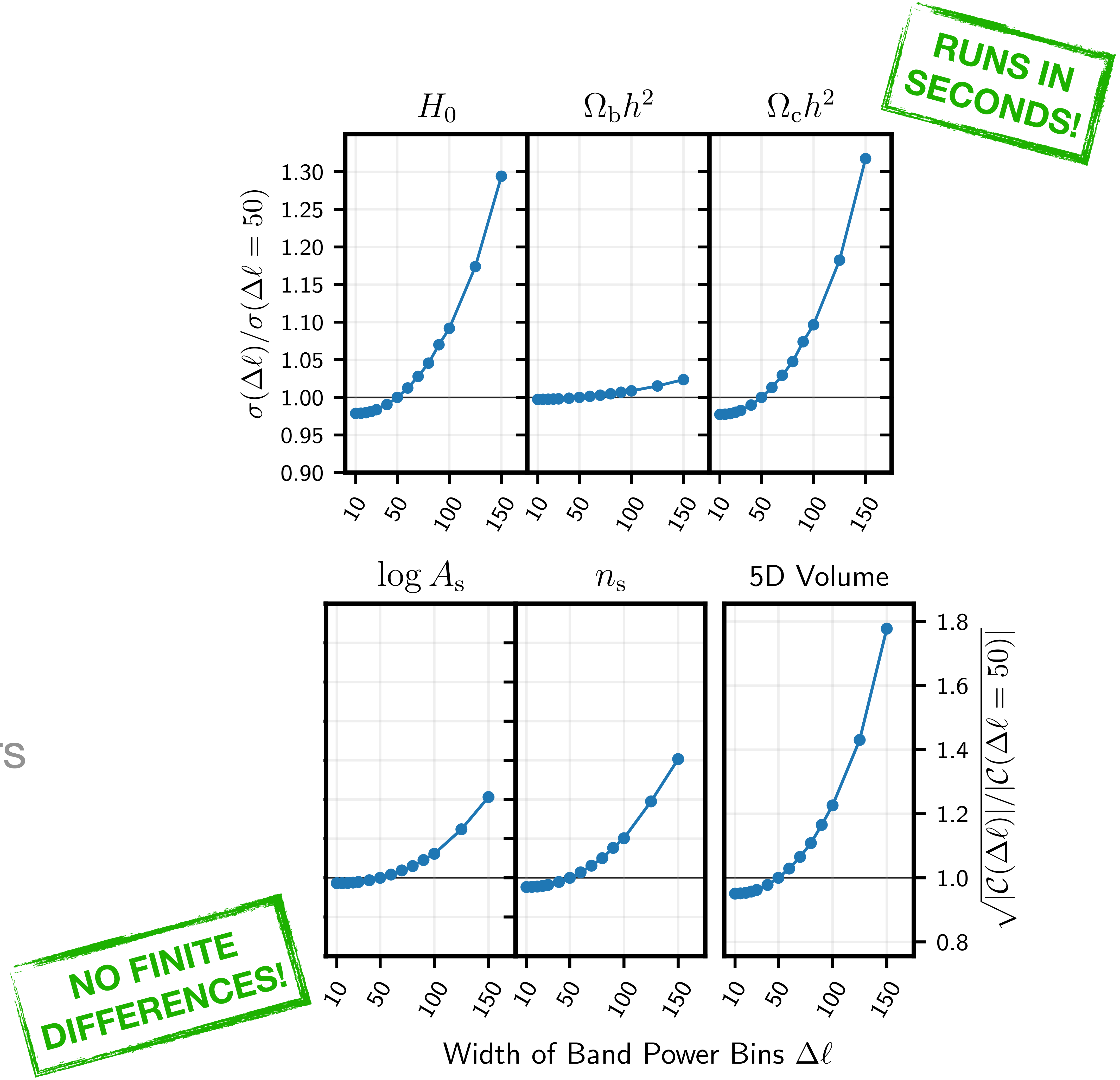
```
candl.tools.get_fisher_matrix()
```

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

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

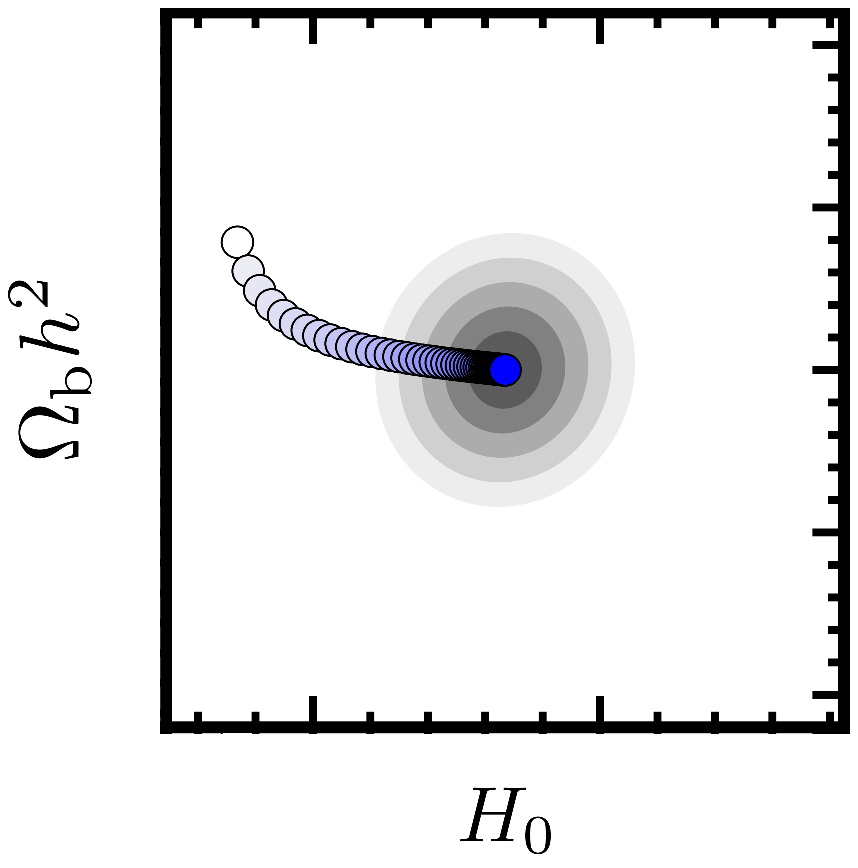
# Applications

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

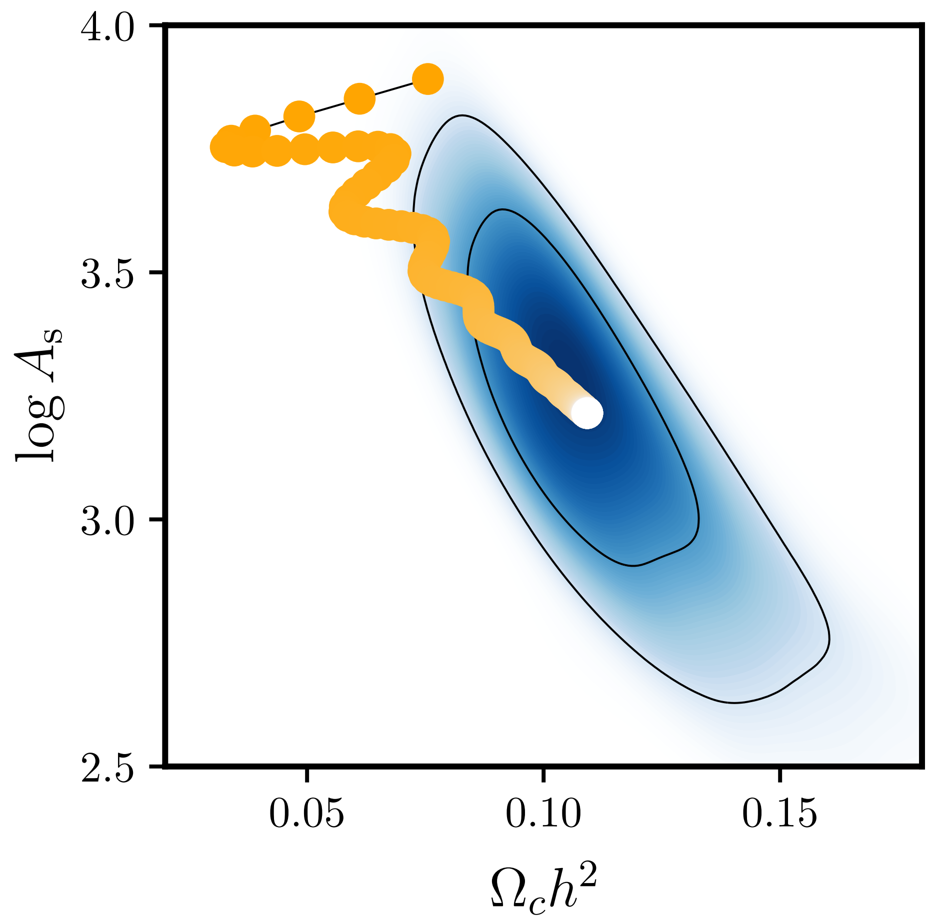


# Applications

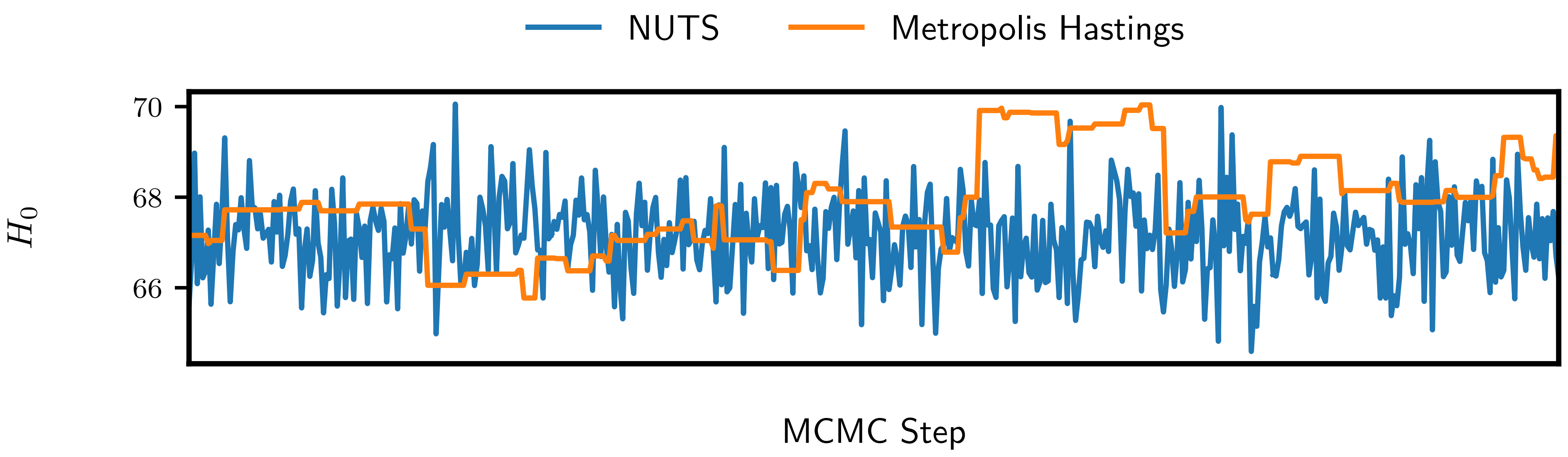
- <u>Fisher forecasting made easy:</u>
  - E.g. optimal band power bin width
- **<u>Other applications:</u>**
  - Gradient-based minimisers/samplers
  - Correlation between subsets




**TT/TE/EE Likelihood Newton-Raphson minimiser**




**ΦΦ Likelihood ADAM minimiser w/ Optax**
(arXiv:1412.6980, https://github.com/deepmind/optax)

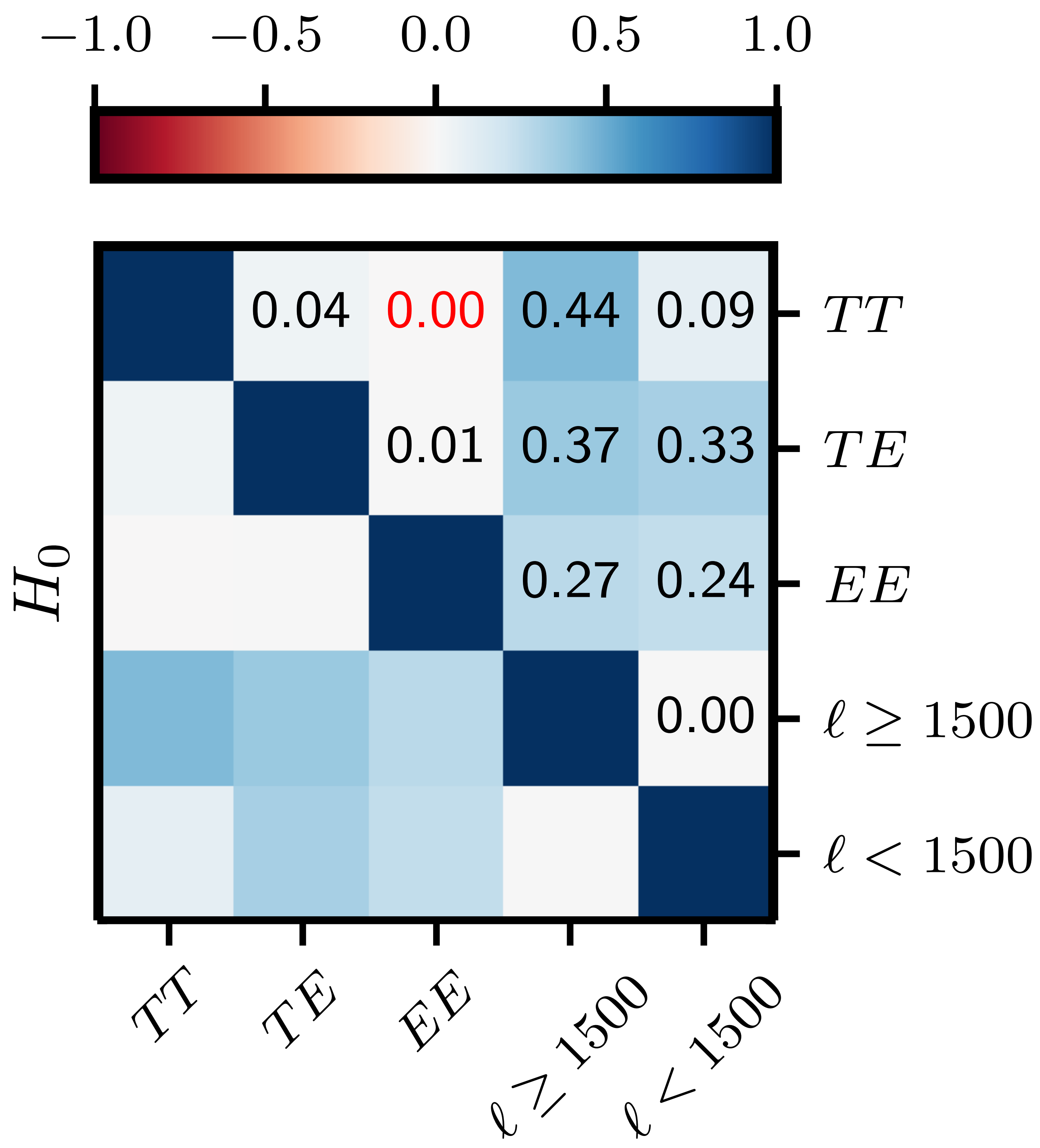



**TT/TE/EE Likelihood, NUTS sampling w/ BlackJAX**
(arXiv:1111.4246, https://github.com/blackjax-devs/blackjax)

# Applications

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

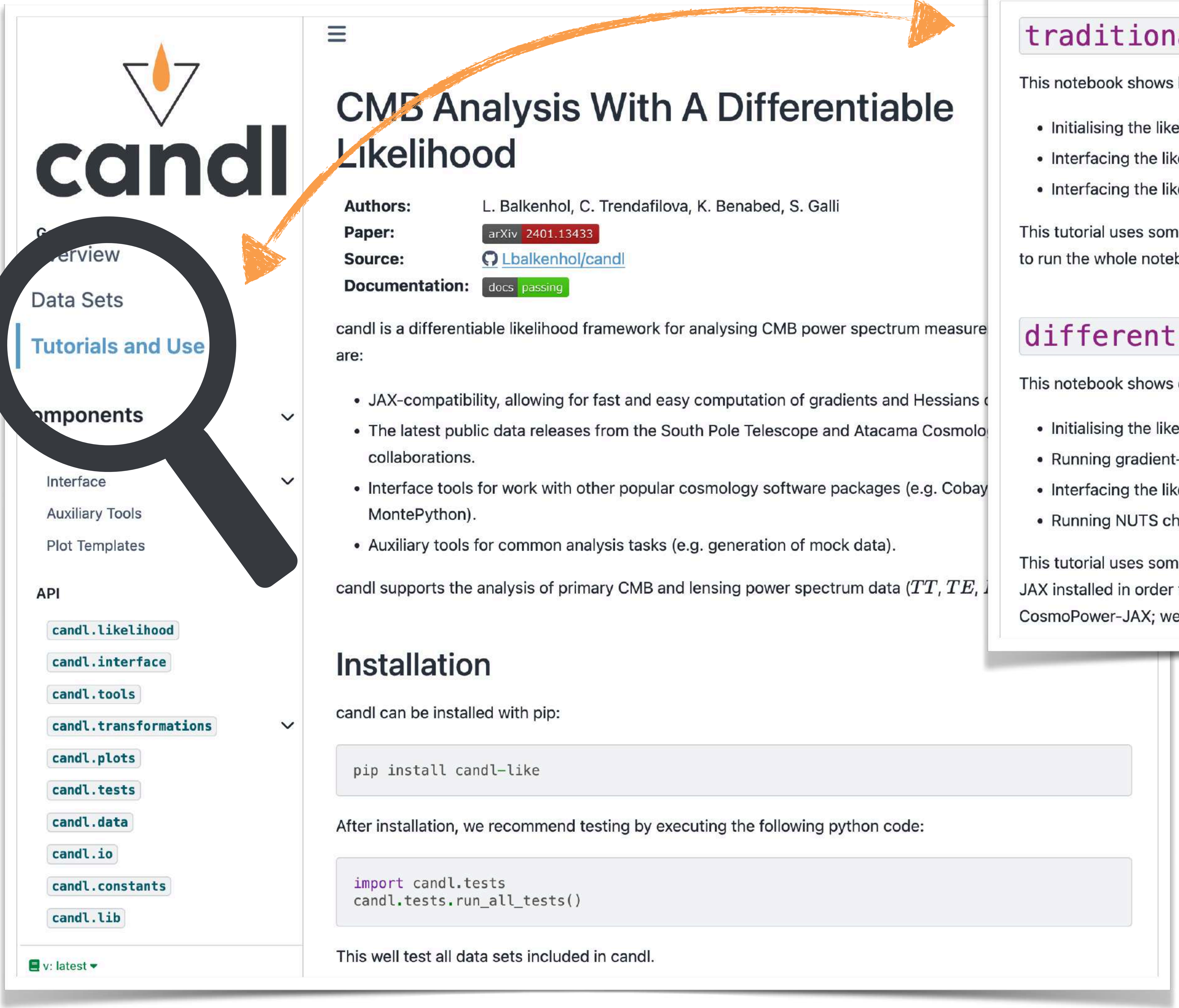
*"How correlated are $H_0$ constraints from different parts of the data?"*




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

# Why Should CMB-S4 Care?

1. *Forecasting in ΛCDM (and simple extensions) made fast and robust*
    - Easily propagate instrumental design, analysis choices through to parameters
    - No finite differences: no multiple likelihood evulation points, no tuning of step sizes, …no numerical jibberish!
2. *Accessible python package with intuitive access to data products*
3. *Easy comparison with SPT and ACT data*



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 spectrum measure
are:

- JAX-compatibility, allowing for fast and easy computation of gradients and Hessians o
- The latest public data releases from the South Pole Telescope and Atacama Cosmolo collaborations.
- Interface tools for work with other popular cosmology software packages (e.g. Cobay 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 ($TT, TE,$

## Installation

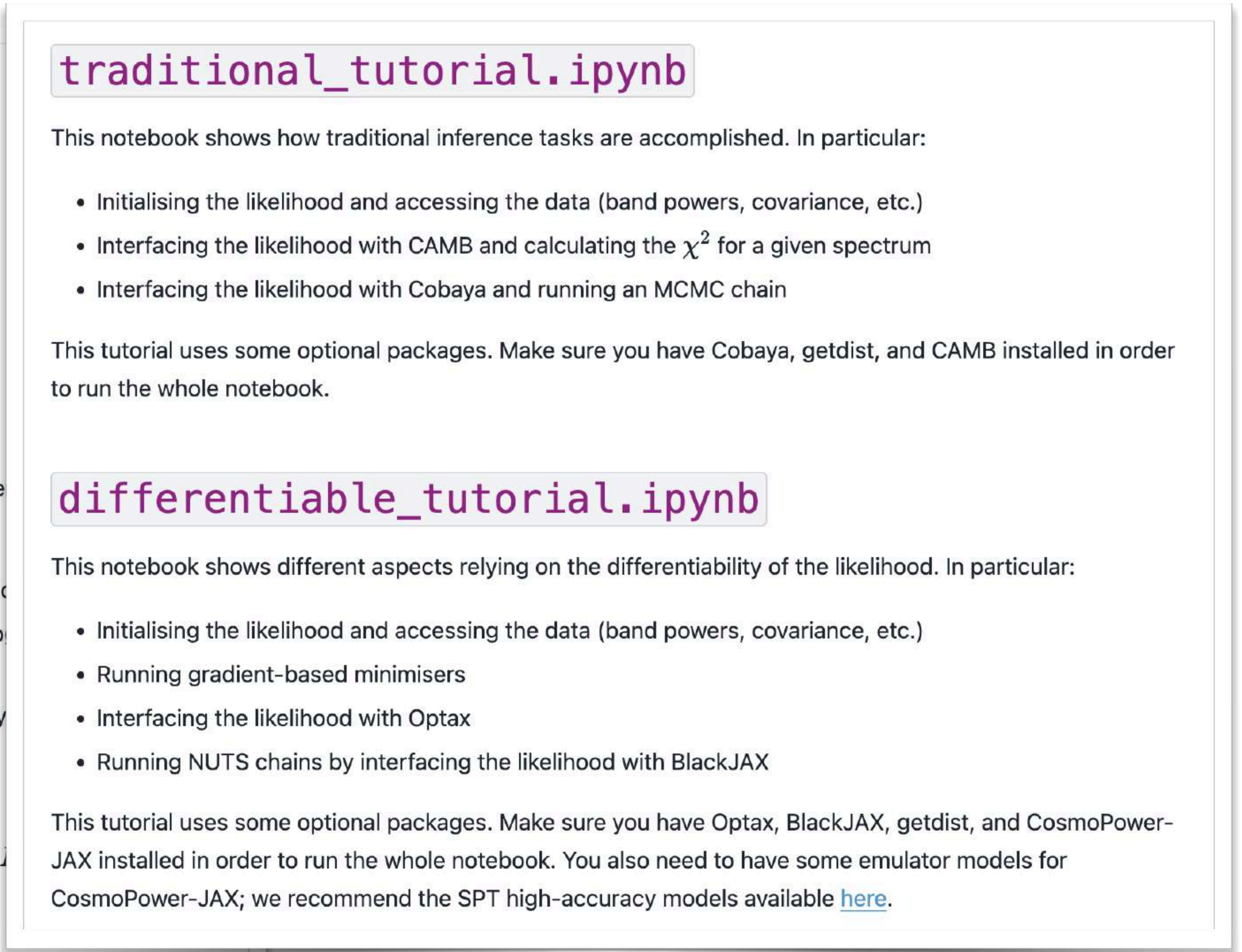
candl can be installed with pip:

```
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.



`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 $\chi^2$ 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.

Extensive Documentation and Tutorials:

https://candl.readthedocs.io/en/latest/

# Conclusions

- Upcoming CMB data need fast, efficient, robust tools
- **candl** is a python-based, stand-alone, CMB likelihood
- **candl** comes with extensive documentation and tutorials
- **candl** is differentiable giving easy access to Fisher matrices
- **candl** can easily be interfaced with Cobaya, Montepython, CosmoPower, Optax, BlackJAX, …

```
> 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

Differentiable

# candl

$$\left\{ H_0, \omega_c, \omega_b, n_s, A_s, \tau, \ldots \right\} \left[ \longrightarrow \left( D_\ell^{CMB} \right) \longrightarrow \right] \rightarrow \chi^2$$

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)
```

# candl

Stand-alone

Straightforward interface

```
import candl
import candl.data

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

import candl.interface
cobaya_dict = {‘likelihood':
  candl.interface.get_cobaya_info_dict_for_like(candl_like)}
```

Easy access to data

# can

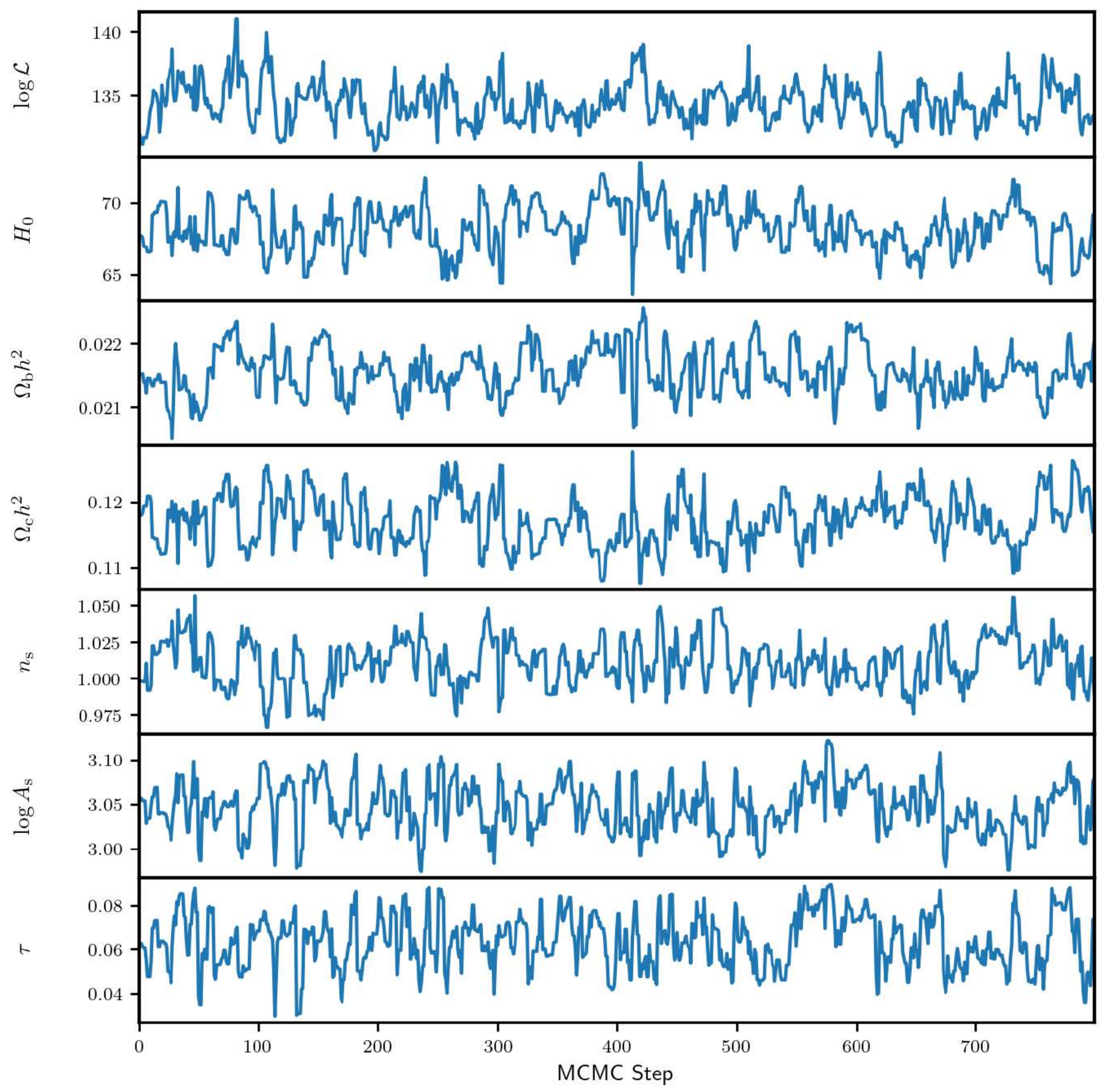
Stand-alone

Straightforward interface

```
import candl
import candl.data

candl_like = candl.Like(candl.data
# candl_like.data_bandpowers, canc

import candl.interface
cobaya_dict = {‘likelihood':
  candl.interface.get_cobaya_info_
```



# Related Work

- Increasing use of JAX and differentiable likelihoods more generally in Cosmology:
  - **Campagne et al 2023, JAX-COSMO, arXiv:2302.05163:**
    Library of different cosmological calculations.
    Examples showing the power of differentiable theory code and likelihood.
    Part of differentiable universe initiative.
  - **Piras and Spurio Mancini 2023, CosmoPower-JAX, arXiv:2305.06347:**
    Differentiable emulator written in JAX.
    Compatible with vanilla CosmoPower models.
  - Implicit likelihood inference (arXiv:2104.12992), MUSE (arXiv:2112.09354), …

# CMB Likelihood Analysis

## … is line fitting

***Problem:*** *Given measured data, find posterior distribution of parameters 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