Added Value of New Methods

Lennart Balkenhol (IAP)

CosmoForward (12/2/26)

NEU CosmoS

European Research Council
Established by the European Commission

SORBONNE UNIVERSITÉ

Overview

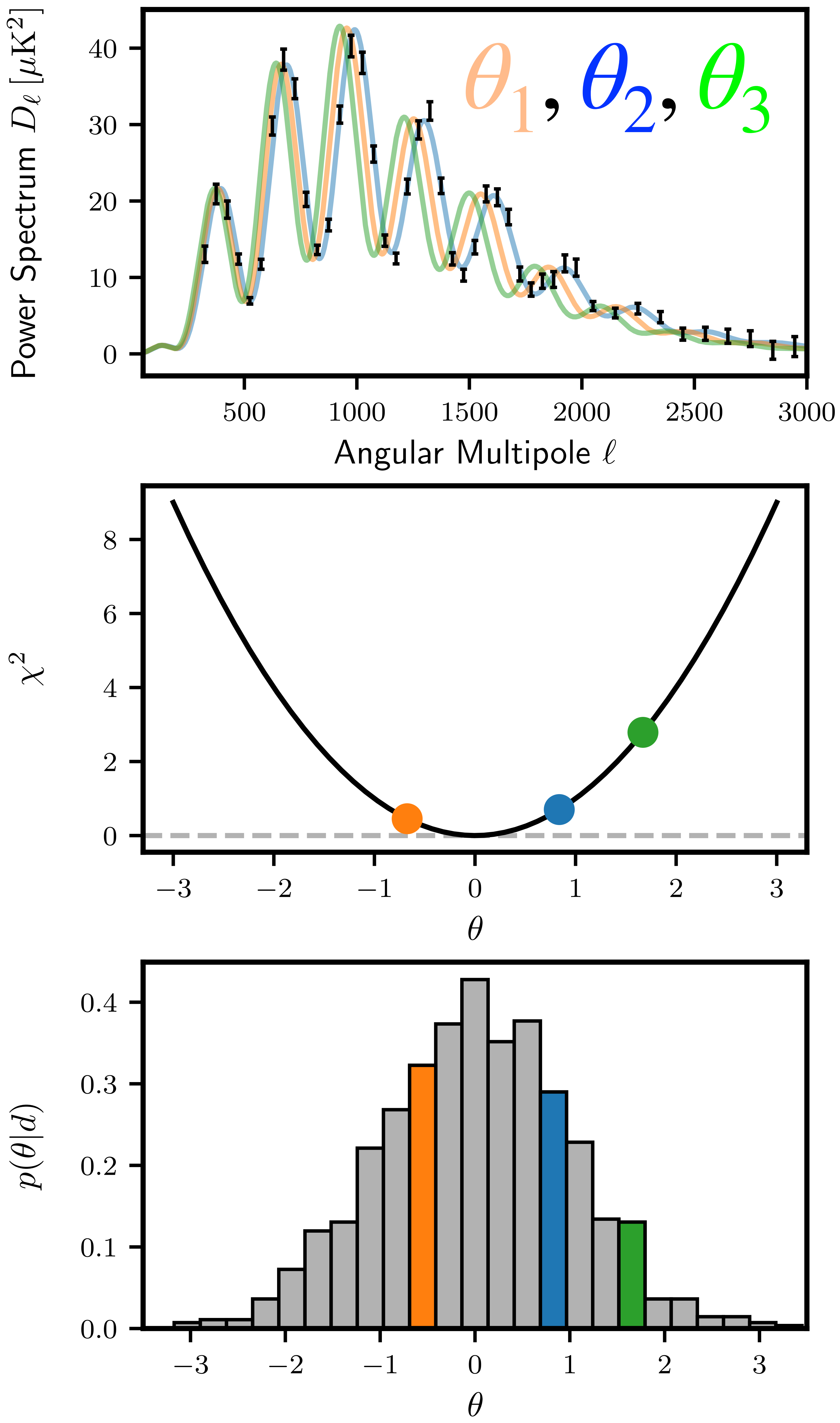
- CMB inference
  - Applications of differentiability
- Broad landscape of tools
  - incl. my personal opinions
- Conclusions

CMB Likelihood Analysis
aka line fitting

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

- **Standard solutions**
  - Boltzmann solvers (CAMB, CLASS)
  - Purpose-written likelihood functions
  - Samplers (Cobaya, MontePython, CosmoSIS, …)
- **Why bother changing things?**
  - Slow
  - Rigid
  - Stagnant for decades…

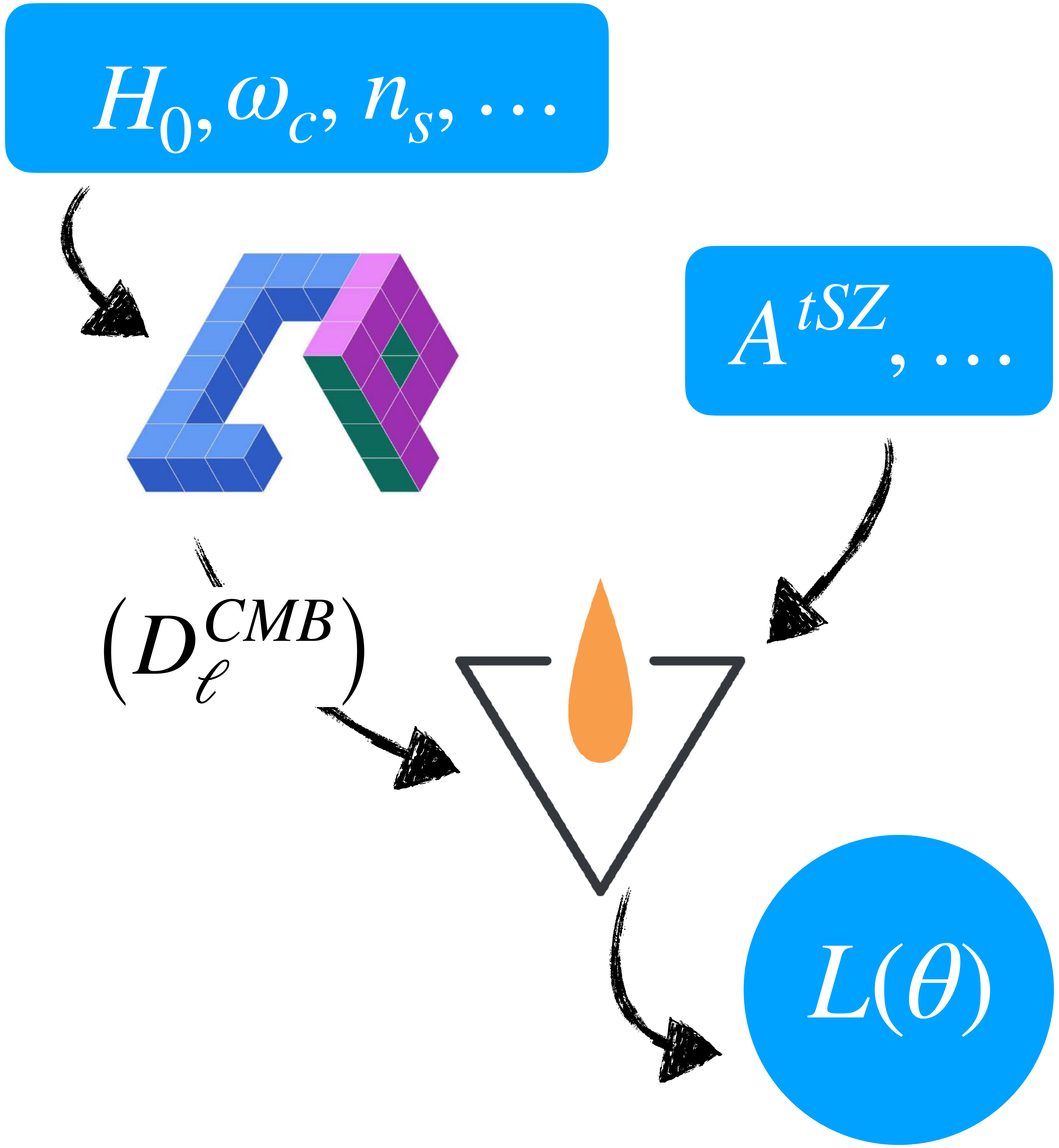
New data deserves better!



Differentiable CMB Pipeline

- **Use JAX autodiff magic with**
  - **CosmoPower-JAX:** Boltzmann emulator [Piras+23]
  - **candl:** differentiable CMB likelihoods (SPT, ACT, Planck) [Balkenhol+24]

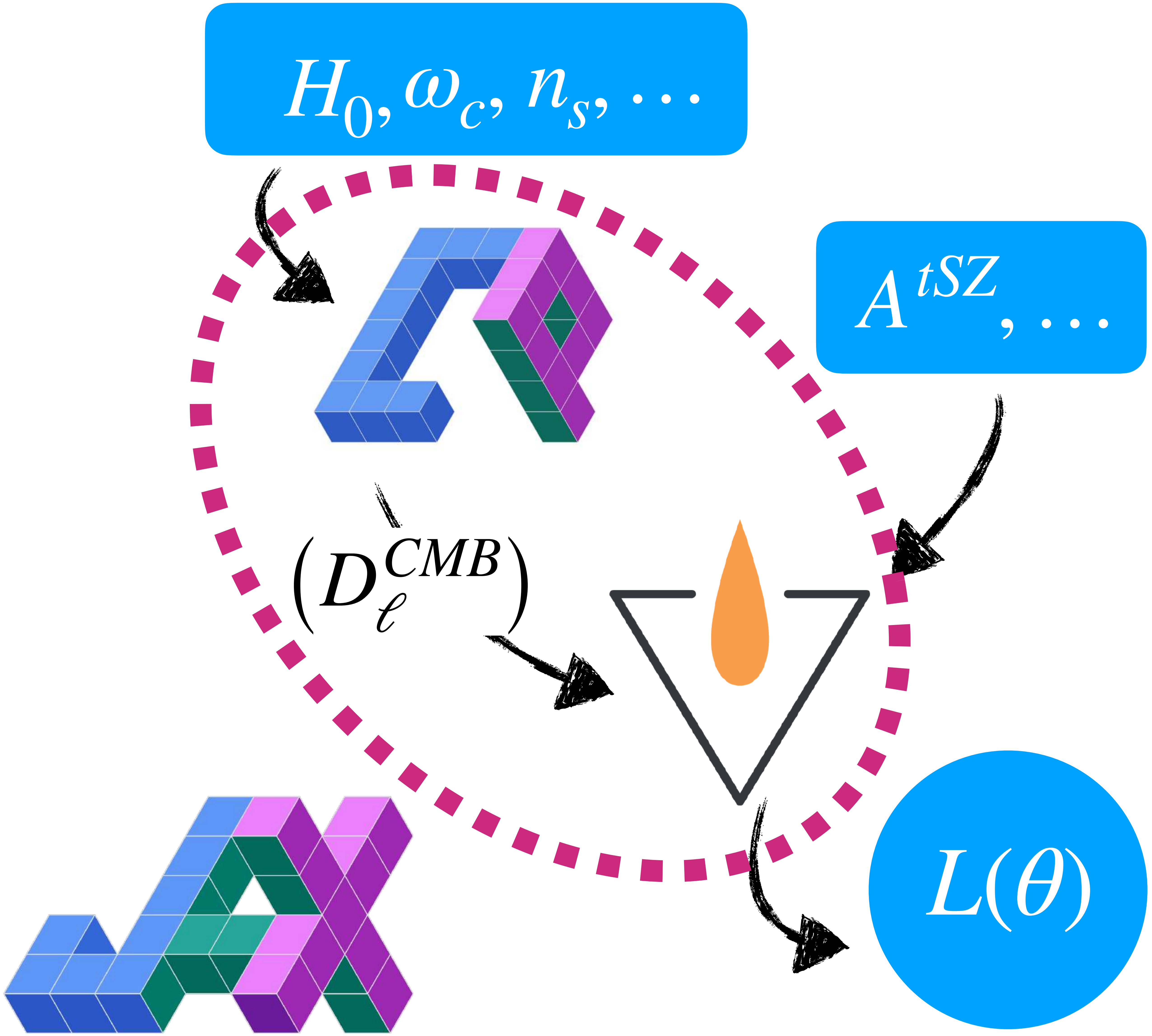
```
def L(theta):
    # candl+CosmoPower
    return logl

import jax
dL_dtheta = jax.grad(L)
d2L_dtheta2 = hessian(L)
```



Differentiable CMB Pipeline

- **Use JAX autodiff magic with**
  - **CosmoPower-JAX:** Boltzmann emulator [Piras+23]
  - **candl:** differentiable CMB likelihoods (SPT, ACT, Planck) [Balkenhol+24]

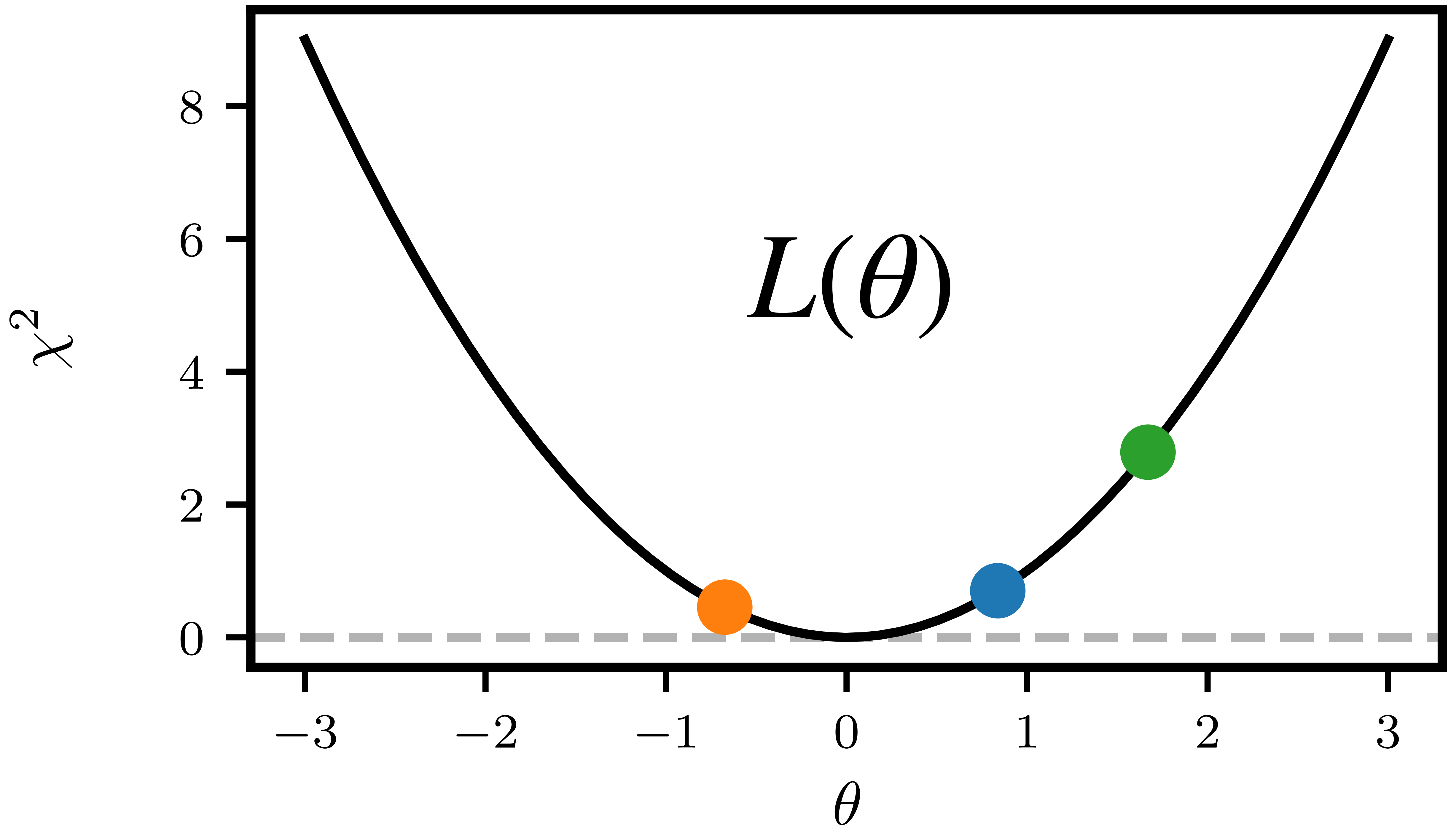
```
def L(theta):
    # candl+CosmoPower
    return logl

import jax
dL_dtheta = jax.grad(L)
d2L_dtheta2 = hessian(L)
```



Differentiable CMB Pipeline

- **Use JAX autodiff magic with**
  - **CosmoPower-JAX:** Boltzmann emulator [Piras+23]
  - **candl:** differentiable CMB likelihoods (SPT, ACT, Planck) [Balkenhol+24]

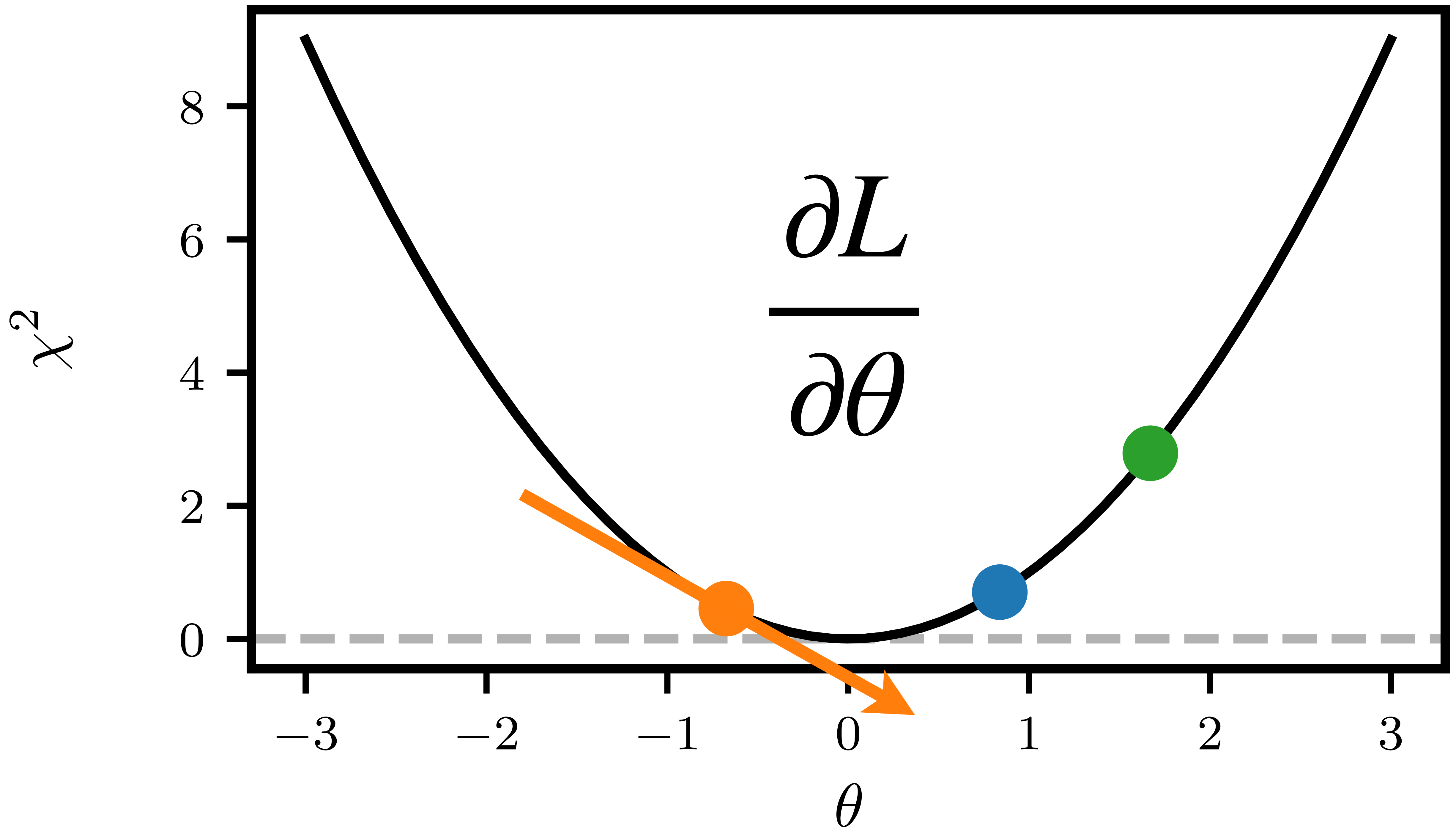
```
def L(theta):
    # candl+CosmoPower
    return logl

import jax
dL_dtheta = jax.grad(L)
d2L_dtheta2 = hessian(L)
```



Differentiable CMB Pipeline

- **Use JAX autodiff magic with**
  - **CosmoPower-JAX:** Boltzmann emulator [Piras+23]
  - **candl:** differentiable CMB likelihoods (SPT, ACT, Planck) [Balkenhol+24]

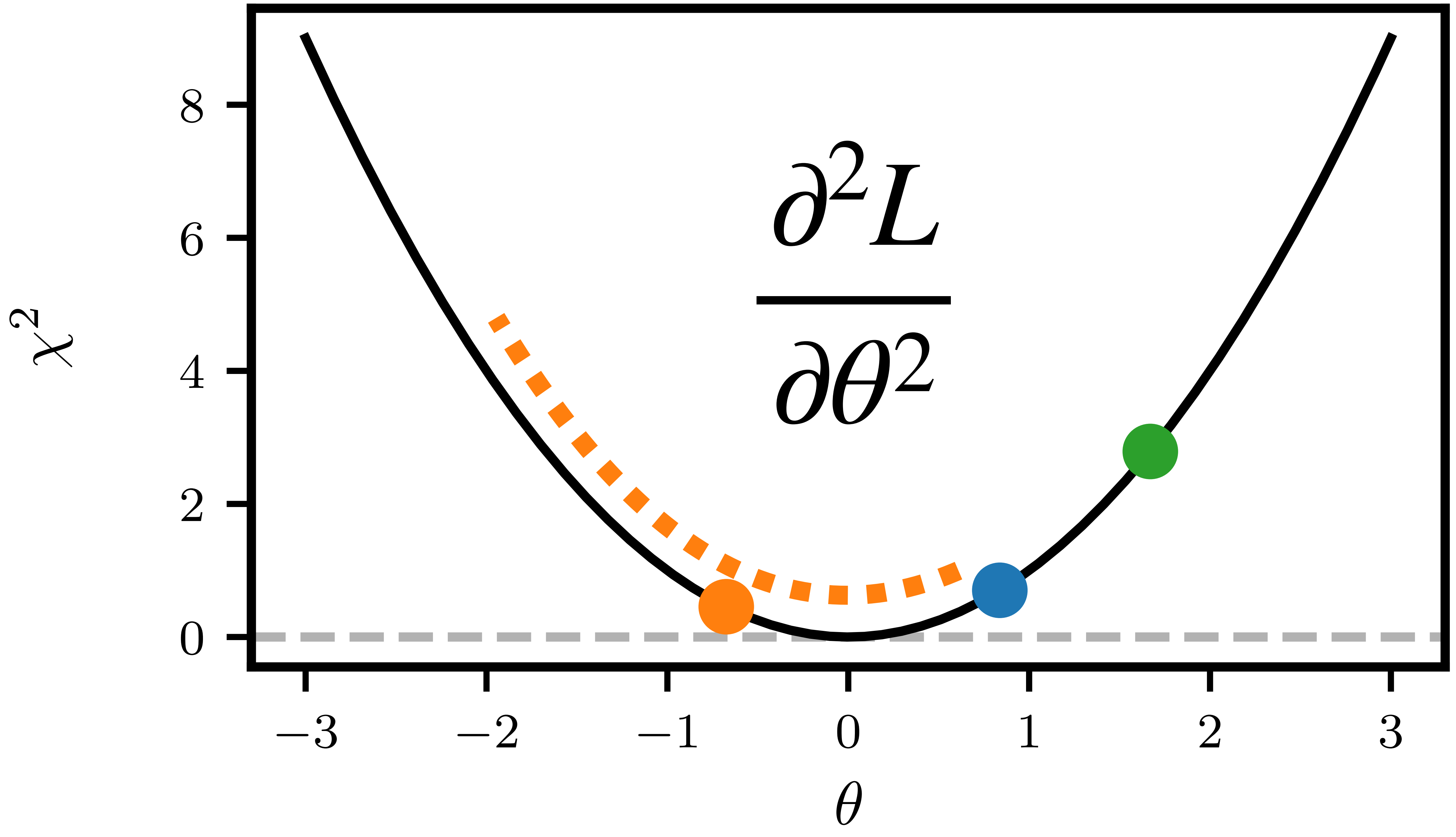
```
def L(theta):
    # candl+CosmoPower
    return logl

import jax
dL_dtheta = jax.grad(L)
d2L_dtheta2 = hessian(L)
```



Differentiable CMB Pipeline

- **Use JAX autodiff magic with**
  - **CosmoPower-JAX:** Boltzmann emulator [Piras+23]
  - **candl:** differentiable CMB likelihoods (SPT, ACT, Planck) [Balkenhol+24]

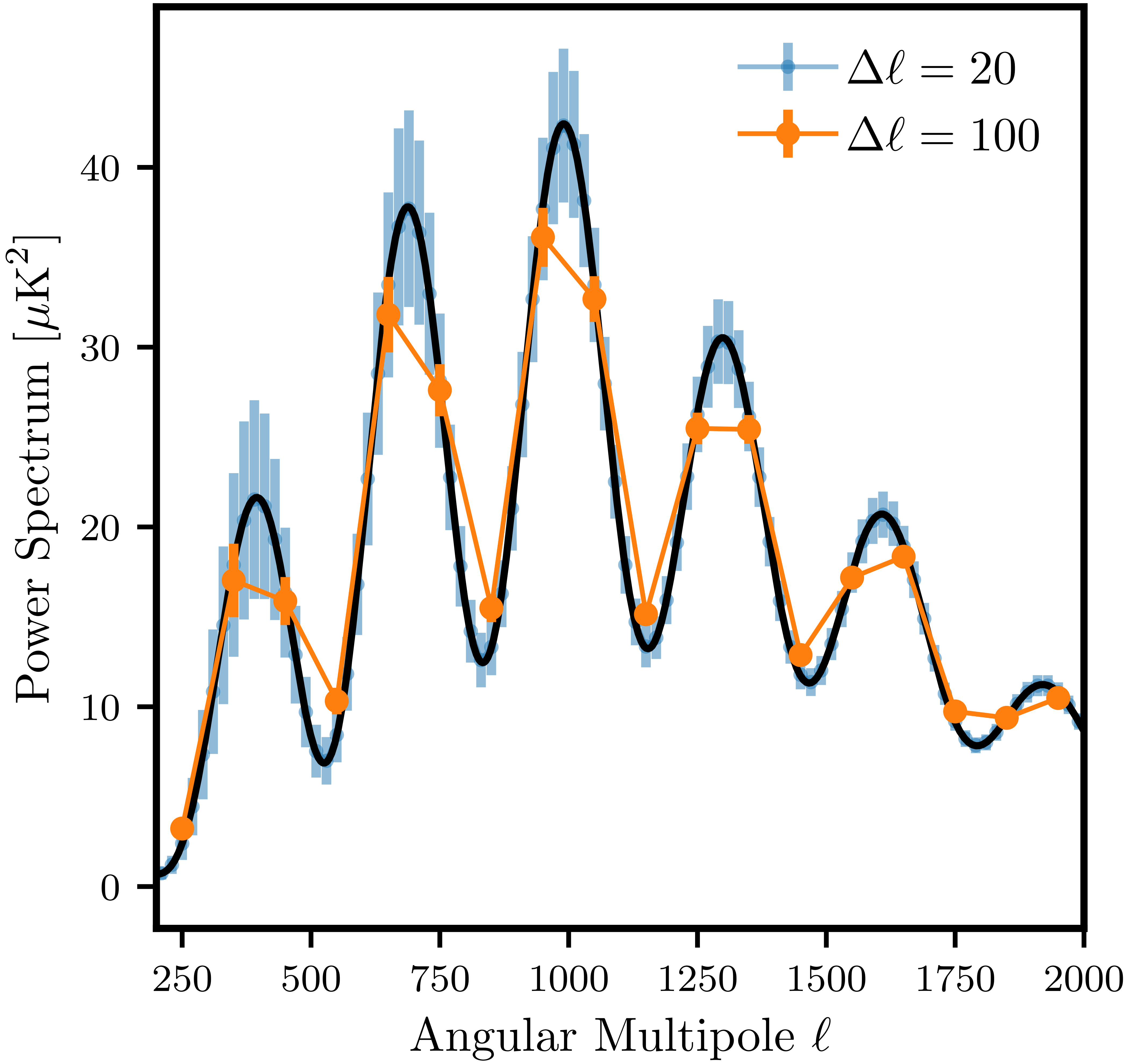
```
def L(theta):
    # candl+CosmoPower
    return logl

import jax
dL_dtheta = jax.grad(L)
d2L_dtheta2 = hessian(L)
```



Applications

“How should I bin my data?”

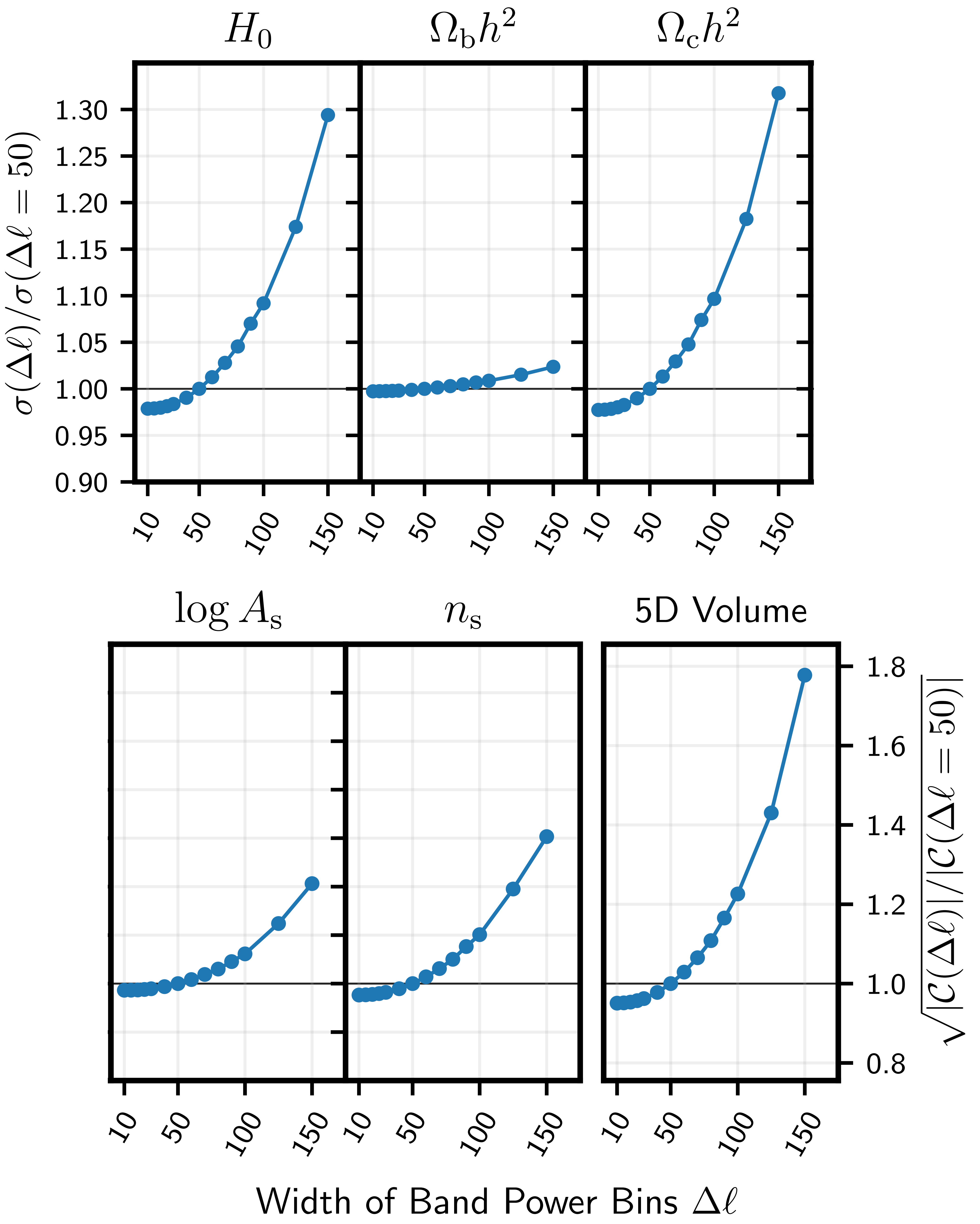
- **<u>Quick, easy, reliable Fisher matrices:</u>**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **<u>Smart exploration of the likelihood:</u>**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling



Balkenhol et al. 2024

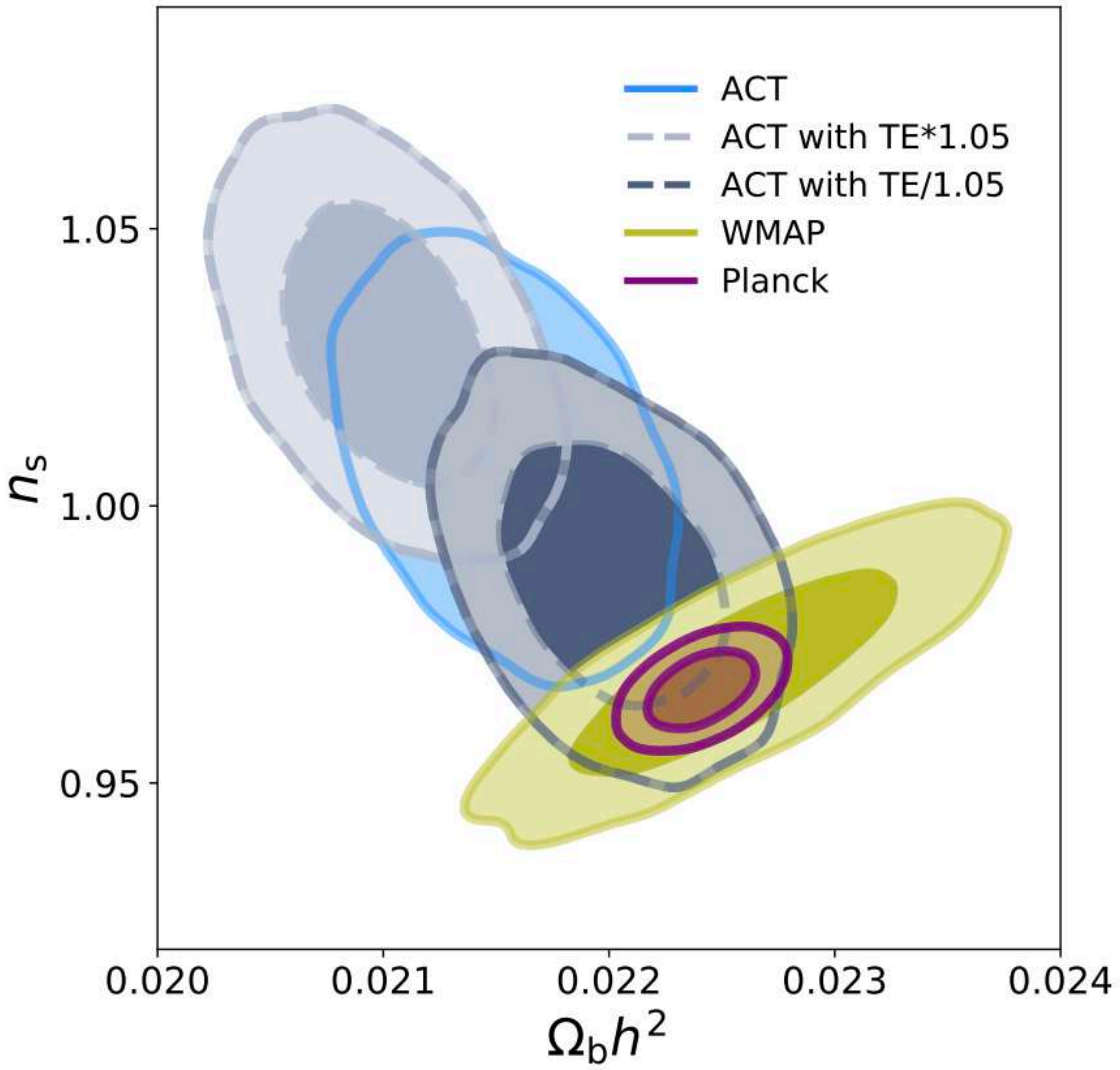
Applications

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling



Applications

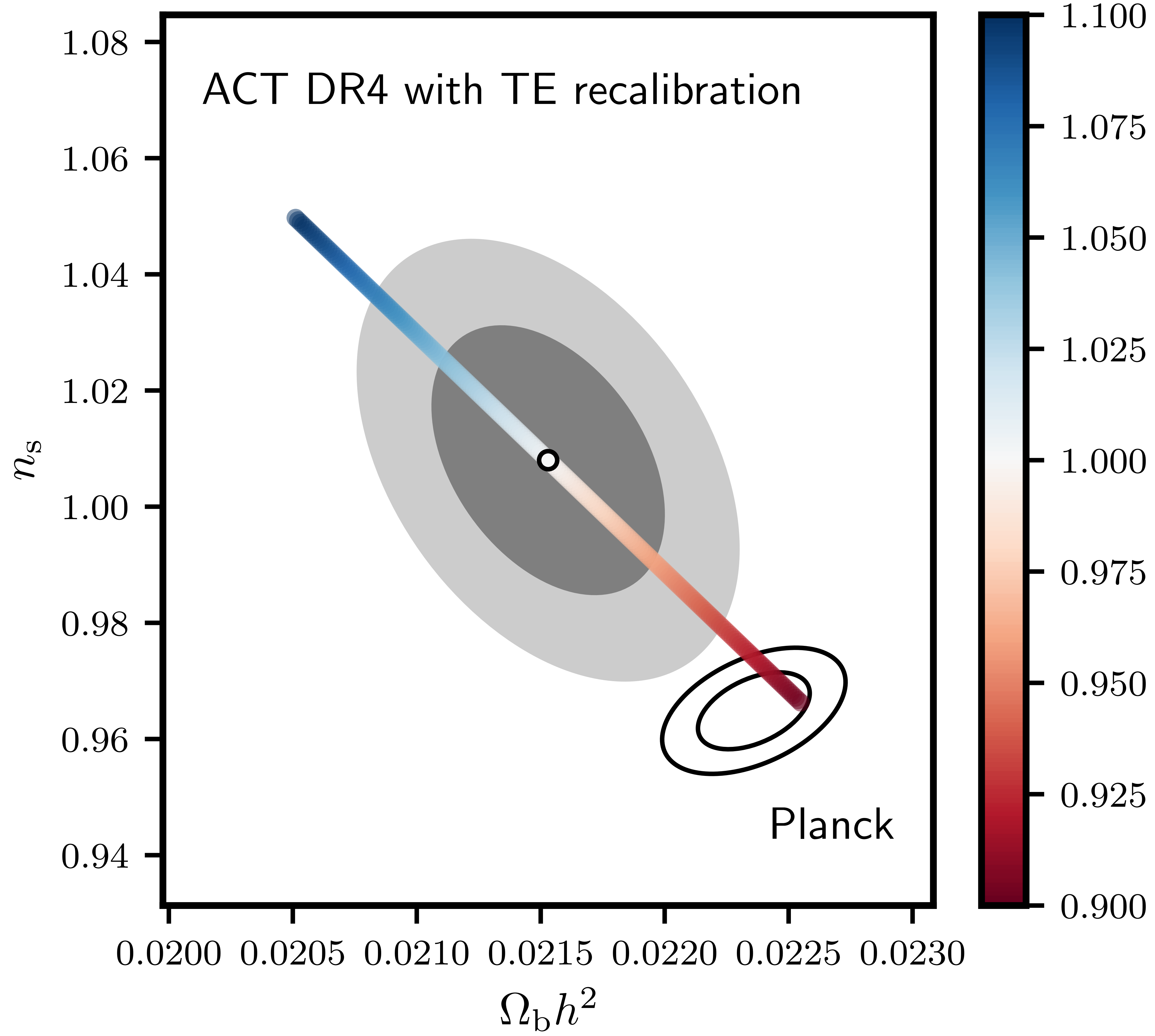
- **<u>Quick, easy, reliable Fisher matrices:</u>**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **<u>Smart exploration of the likelihood:</u>**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling



Aiola et al. 2020

Applications

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling



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

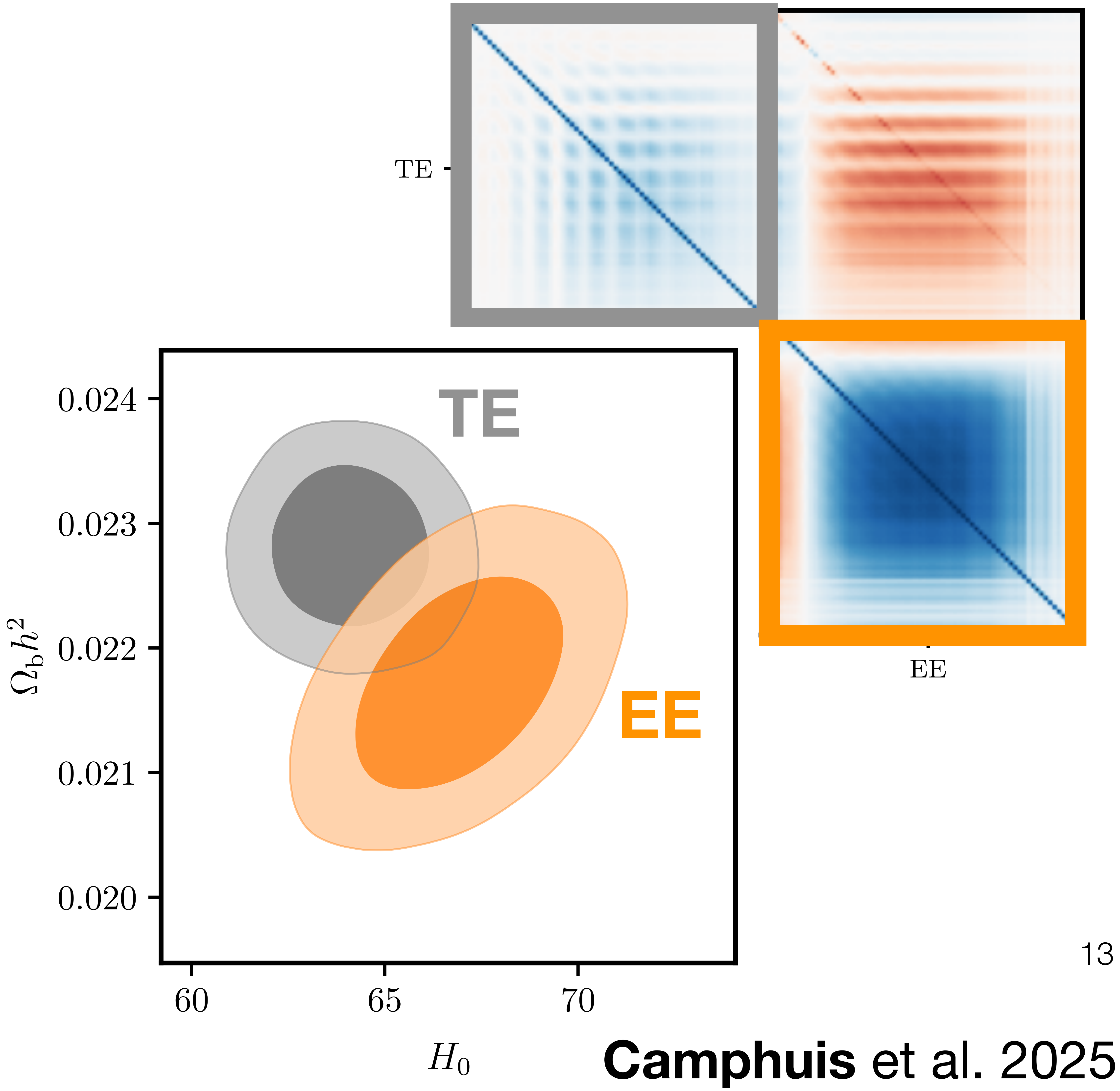
to parameters

from band powers

Applications

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

- **<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
  - Approximating MCMC chains
  - Gradient-powered sampling



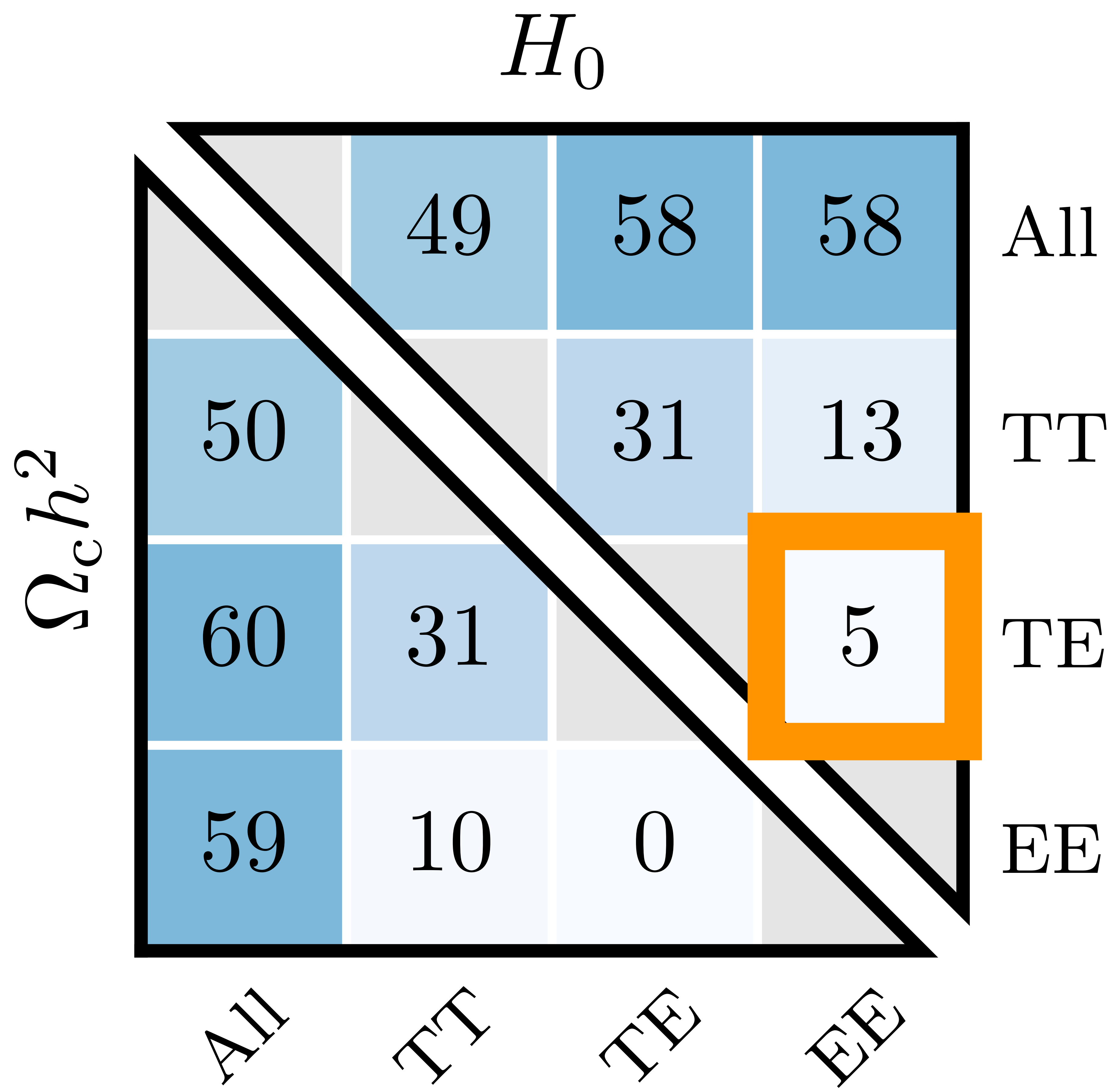
Camphuis et al. 2025

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

Applications

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling

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



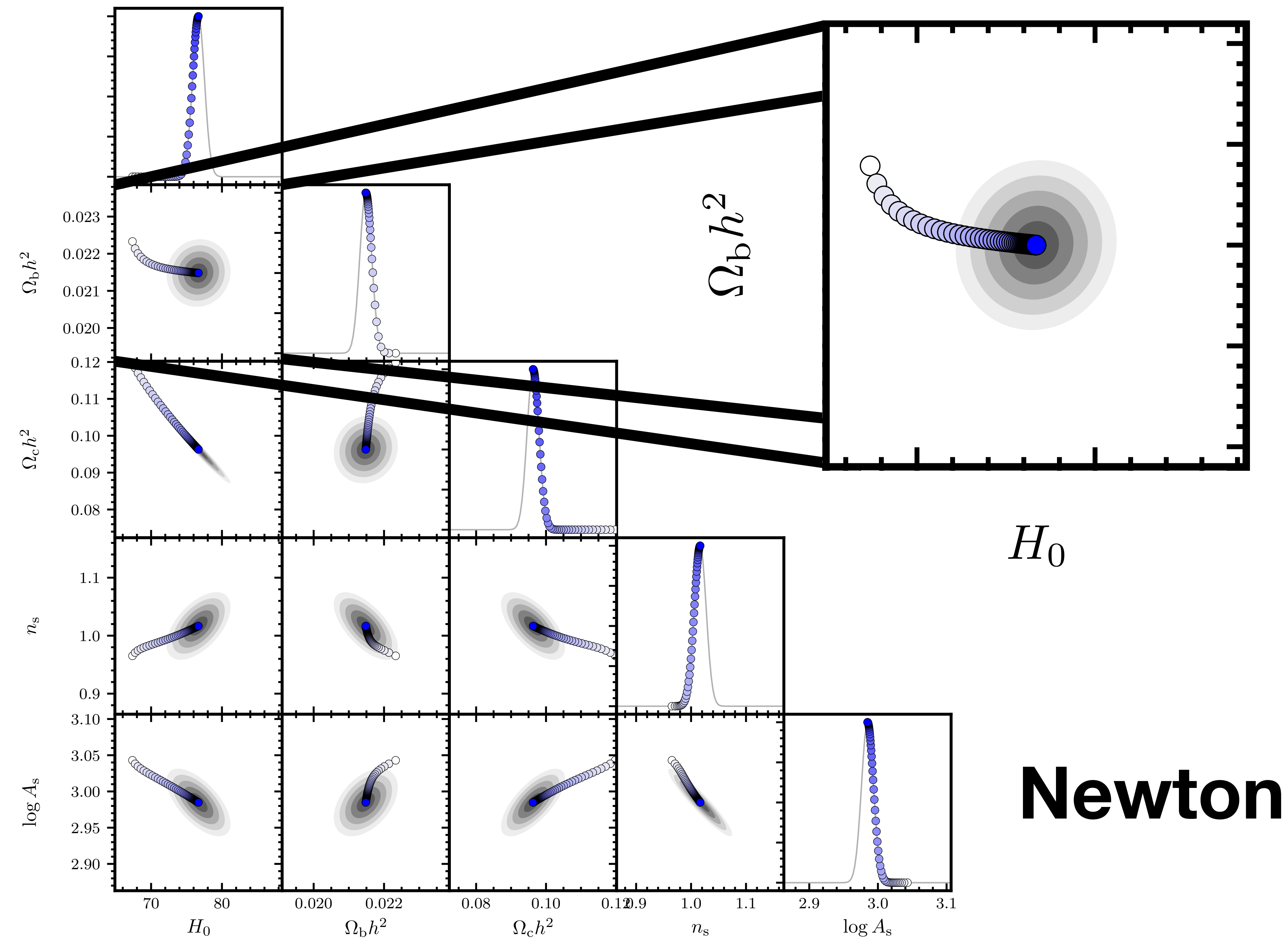
$$\rho(H_0^{TE}, H_0^{EE}) = f(F^{TE}, F^{EE}, \partial L/\partial\theta)$$

Camphuis et al. 2025

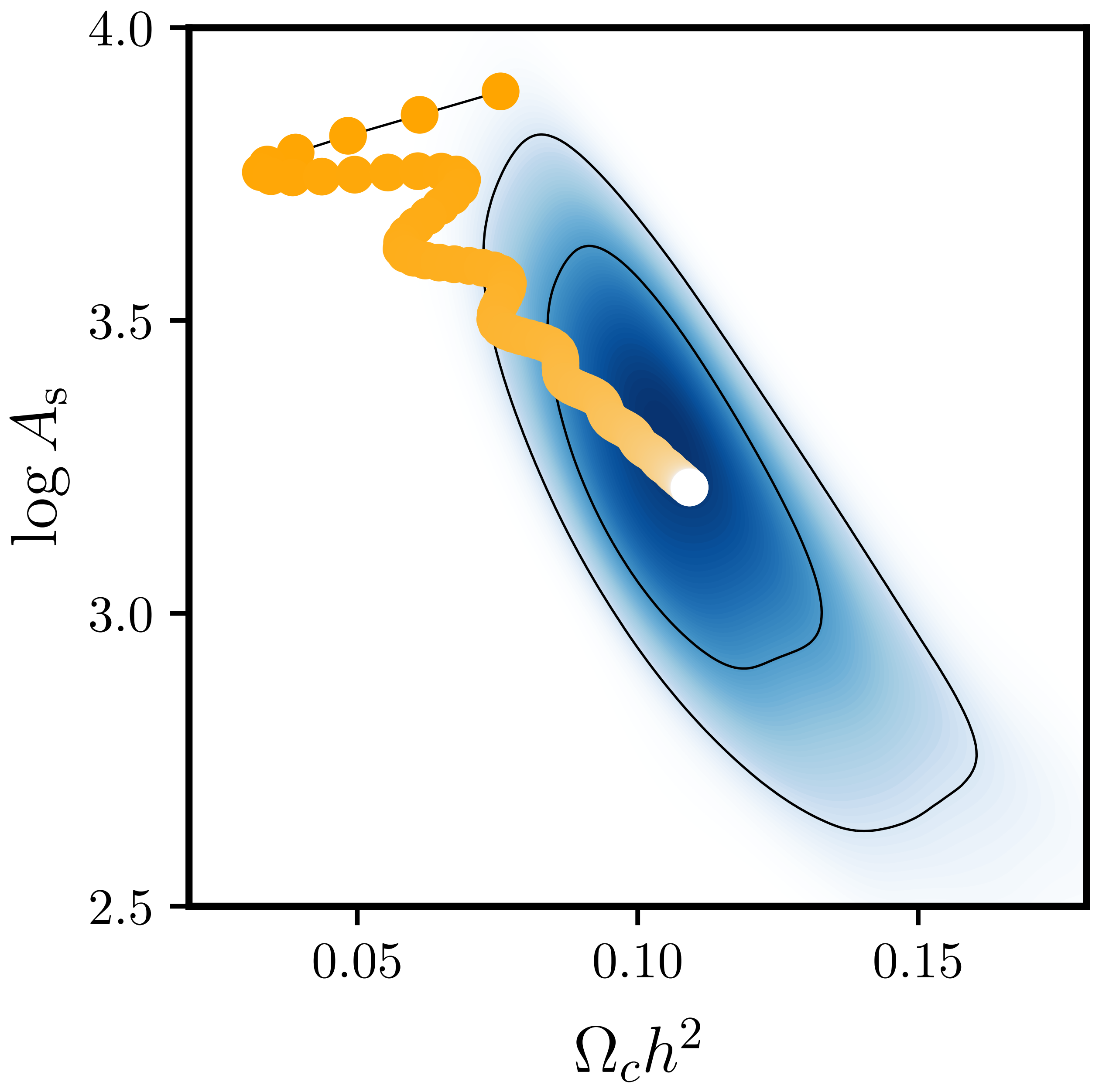
Background Kable, Addison, Bennett 2019 (arXiv:1908.01626)

Applications

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling



Newton-Raphson



ADAM (arXiv:1412.6980)

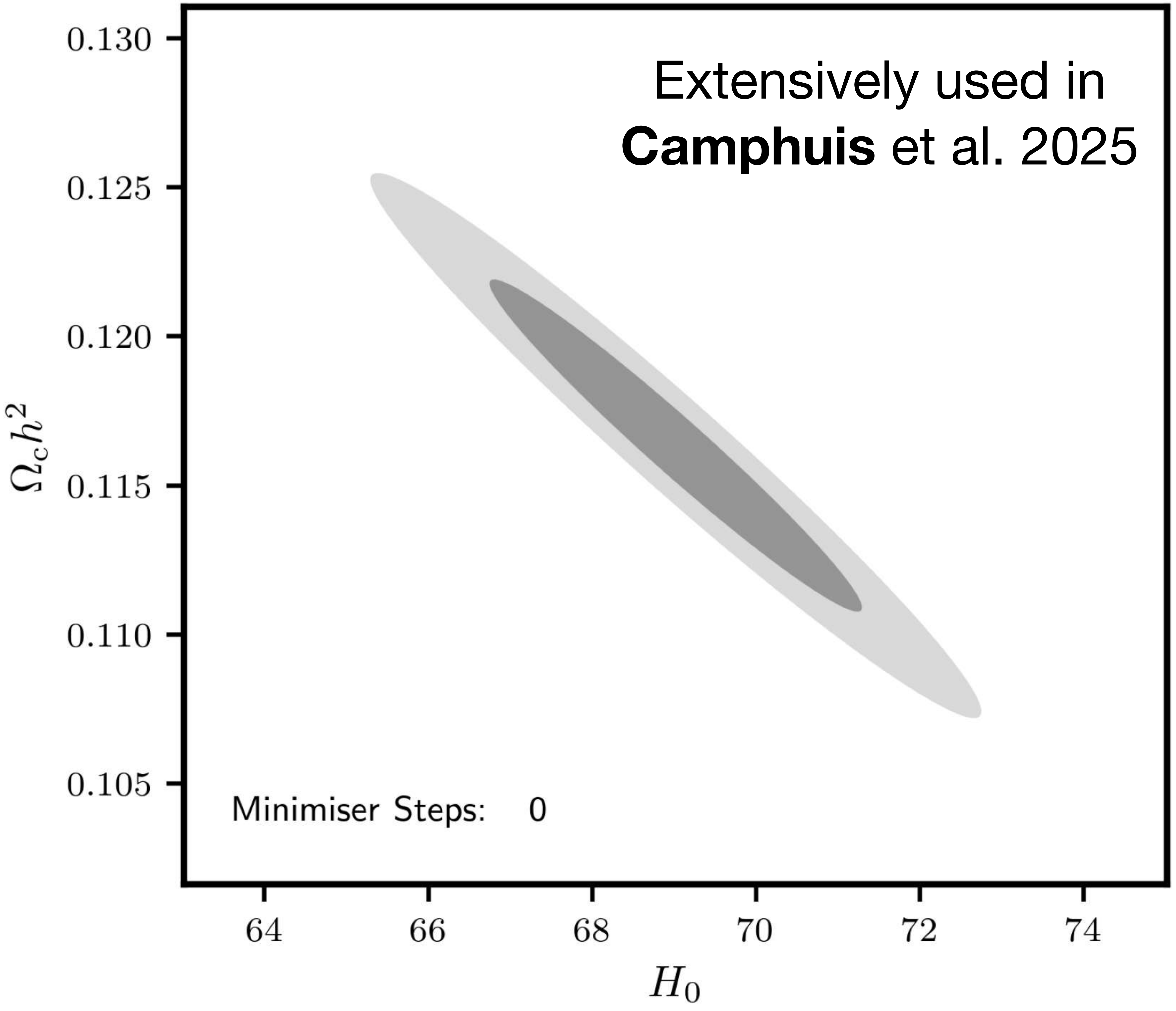
https://github.com/deepmind/optax

Applications

(1): Minimise using gradient information

*(2): Approximate covariance with Fisher matrix**

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling

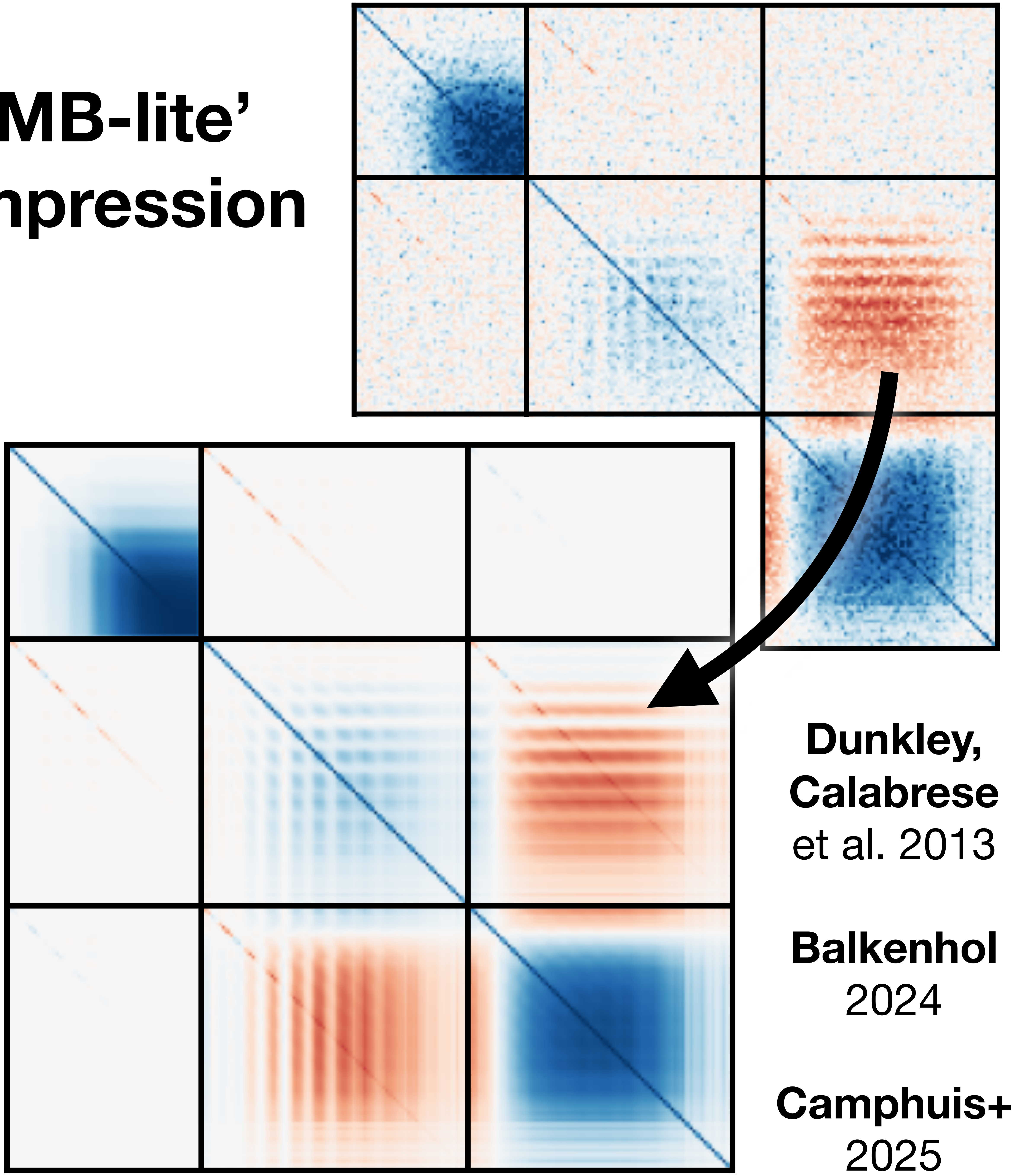


*use DALI for higher order corrections (Sellentin+14)

Applications

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling

'CMB-lite' Compression



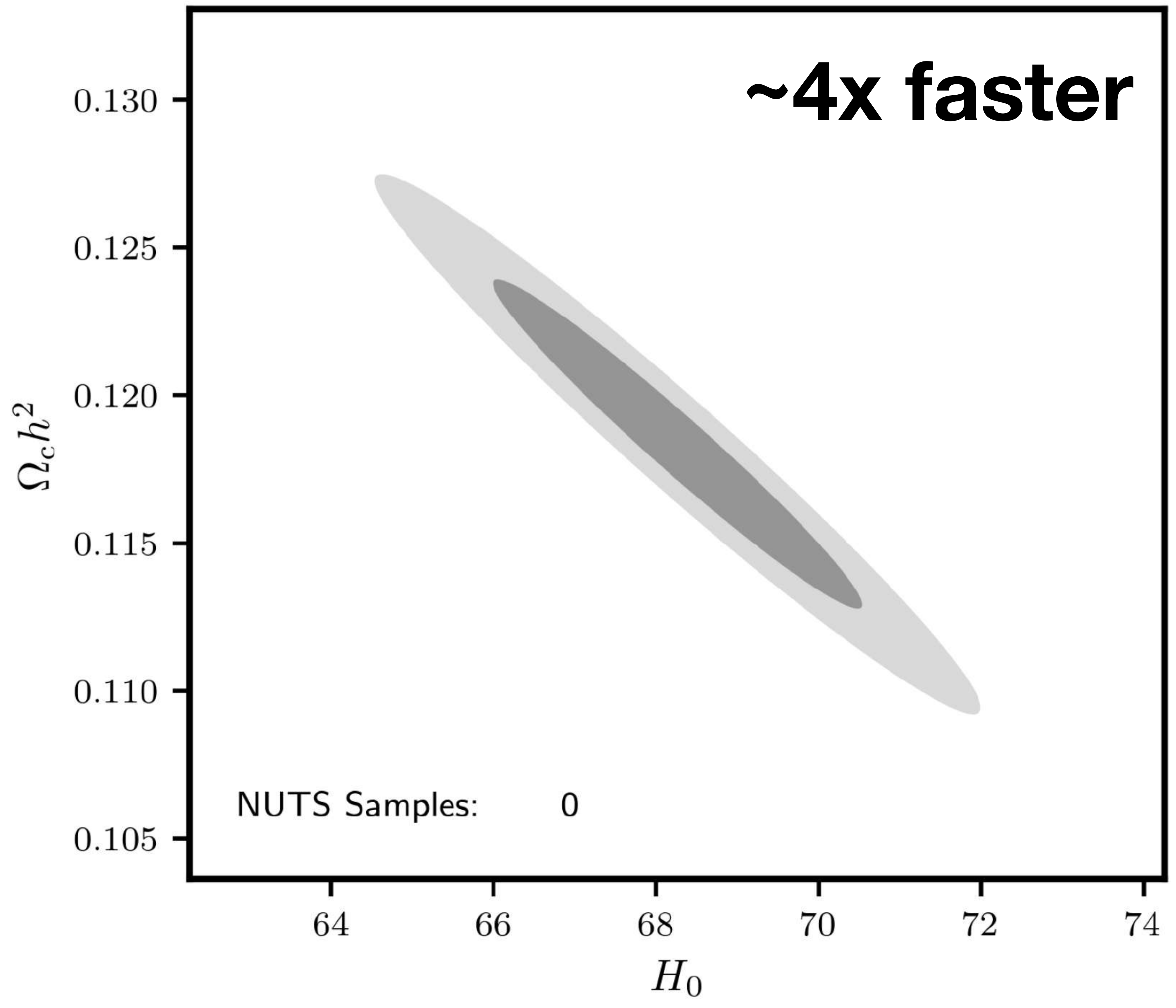
Dunkley, Calabrese et al. 2013

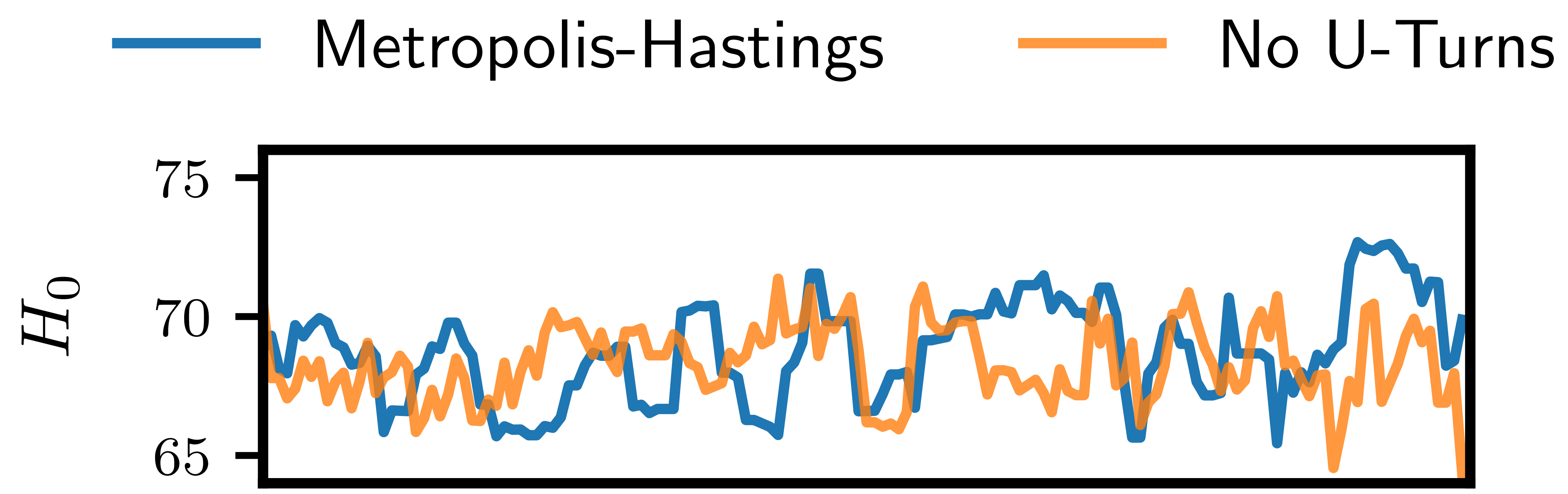
Balkenhol 2024

Camphuis+ 2025

Applications

- **Quick, easy, reliable Fisher matrices:**
  - Forecasting
  - Propagating biases to parameters
  - Correlation between subsets
- **Smart exploration of the likelihood:**
  - Gradient-based minimisers
  - Approximating MCMC chains
  - Gradient-powered sampling





Balkenhol et al. 2024

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

Applications

- **Quick, easy, reliable Fisher matrices:**
    - Forecasting
    - Propagating biases to parameters
    - Correlation between subsets
- **Smart exploration of the likelihood:**
    - Gradient-based minimisers
    - Approximating MCMC chains
    - Gradient-powered sampling

Faster
More flexible
More in depth
More accurate

Methods are mature!

All used in real analyses,
game-changer for SPT-3G D1 analysis
Camphuis et al. 2025

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:

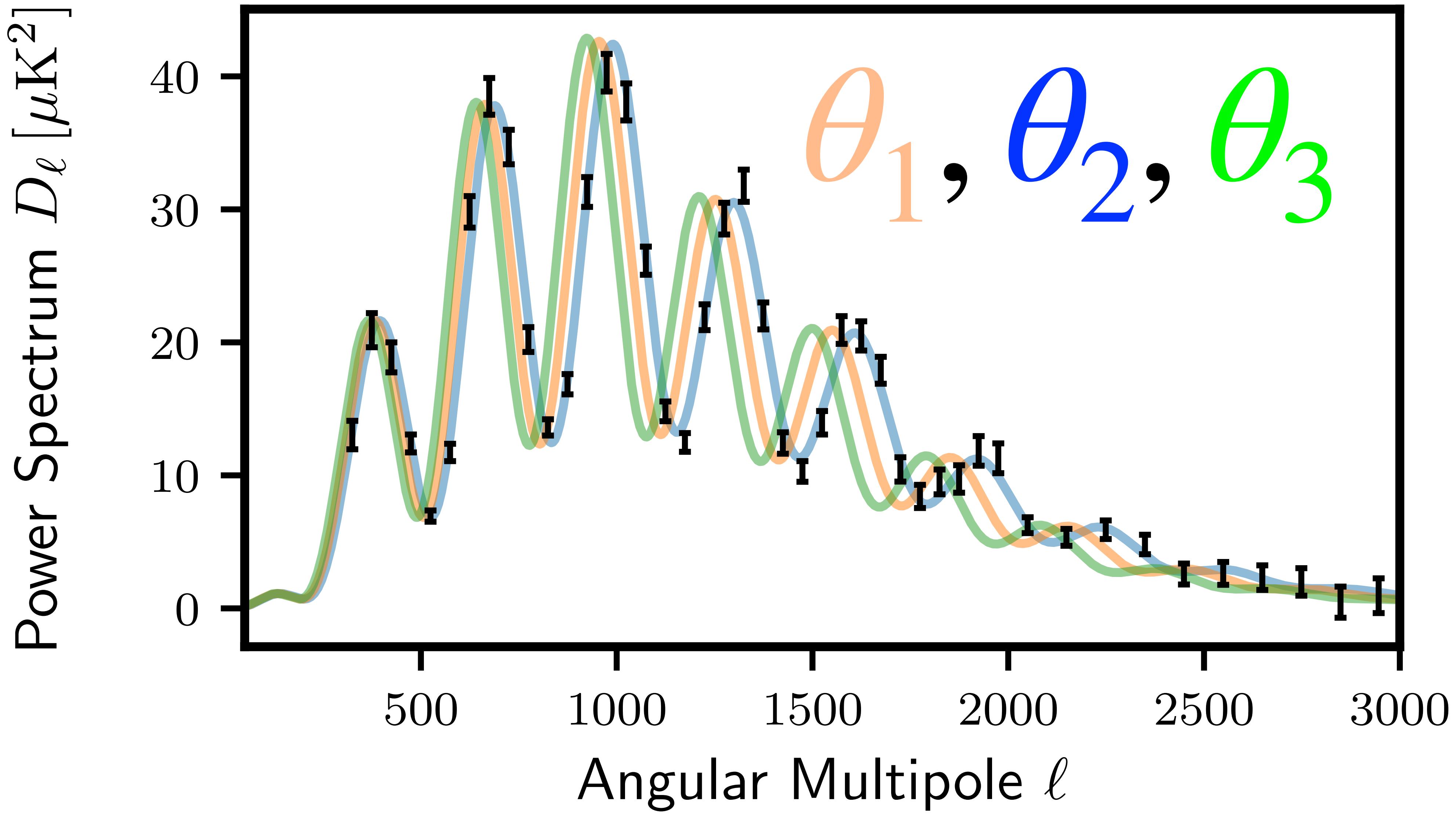
```
import candl.tests
candl.tests.run_all_tests()
```

This well test all data sets included in candl.

Extensive Documentation & Tutorials Available

Theory Calculators

- **Pre-trained emulators:**
  - **CosmoPower** (arXiv:2106.03846, 2305.06347, 2405.07903) [CMB, LSS | NN]
  - PICO (arXiv:0606709) [CMB | poly]
  - Capse.jl (arXiv:2307.14339) [CMB | NN]
  - CosmoNet (arXiv:0608174) [CMB | NN]
  - COMET (arXiv:2208.01070) [LSS | GP]
  - Matryoshka (arXiv:2109.15236, 2202.07557) [LSS | NN]
  - EmulateLSS (arXiv:2112.05889) [LSS | NN]
  - Lazanu (arXiv:2506.07514) [LSS | NN]
  - Coyote (arXiv:1304.7849) [LSS | GP]
  - EuclidEmulator(2) (arXiv:1809.04695, 2010.11288) [LSS | PCA]
  - Aricò+ (arXiv:2104.14568) [LSS | NN]
  - Mira-Titan (arXiv:2207.12345) [LSS | GP]
  - Bartlett+ (arXiv:2510.18749) [LSS | SR]

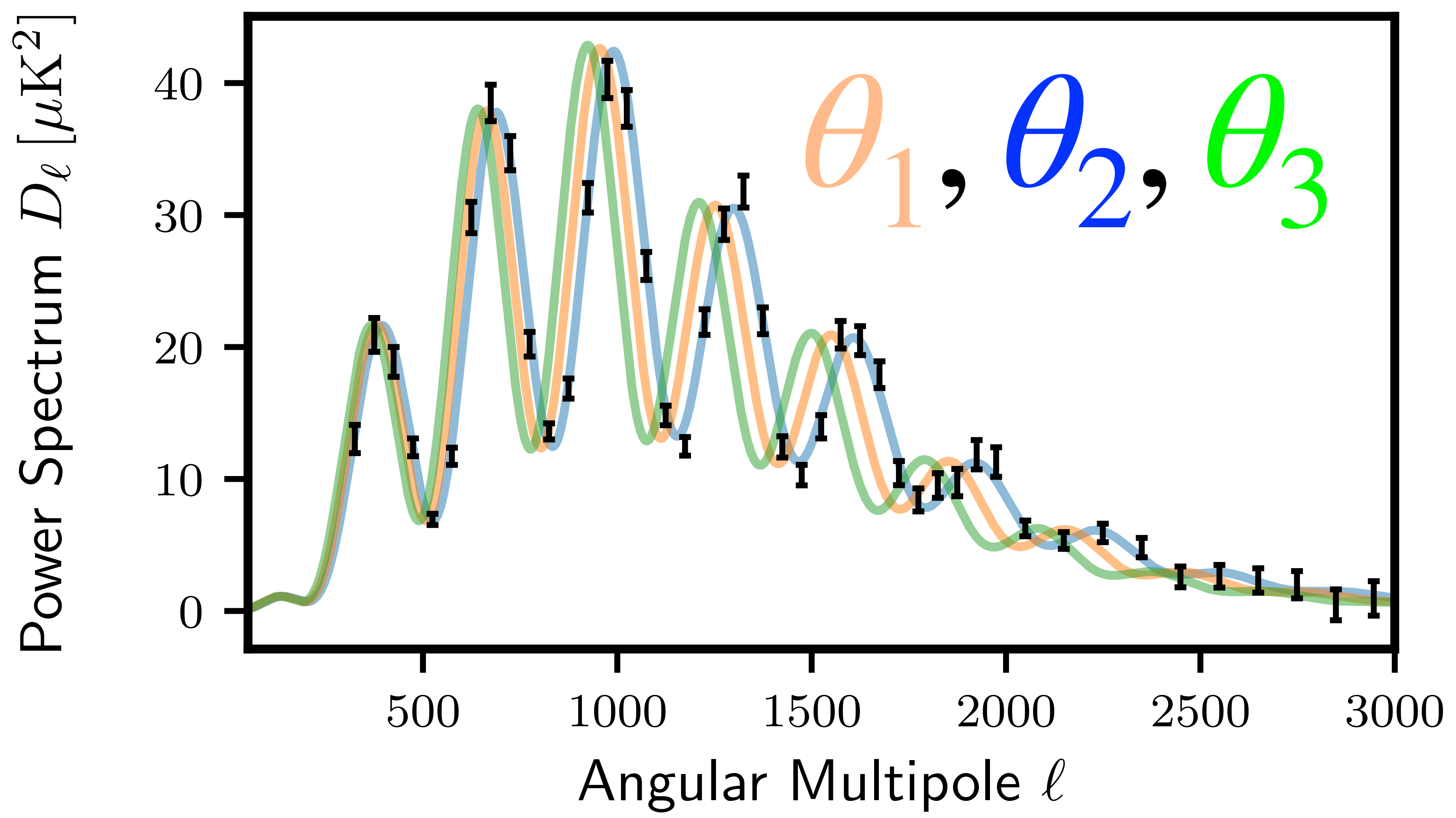


+
Fast
Reusable
Differentiable

−
Substantial Training
Not all models available
Not all observables

... and more!

Theory Calculators



- **Online learning emulators:**
  - CONNECT (arXiv:2205.15726) [NN]
  - OLÉ (arXiv:2307.01138, 2503.13183) [PCA, GP]

\+ No pre-training
Somewhat reusable
Differentiable

− Not fully reusable
Slower due to training

- **Boltzmann solvers++:**
  - COSMICNET (arXiv:1907.05764, 2207.05707)
  - DISCO-DJ (arXiv:2311.03291)
  - LimberJack.jl (arXiv:2310.08306)

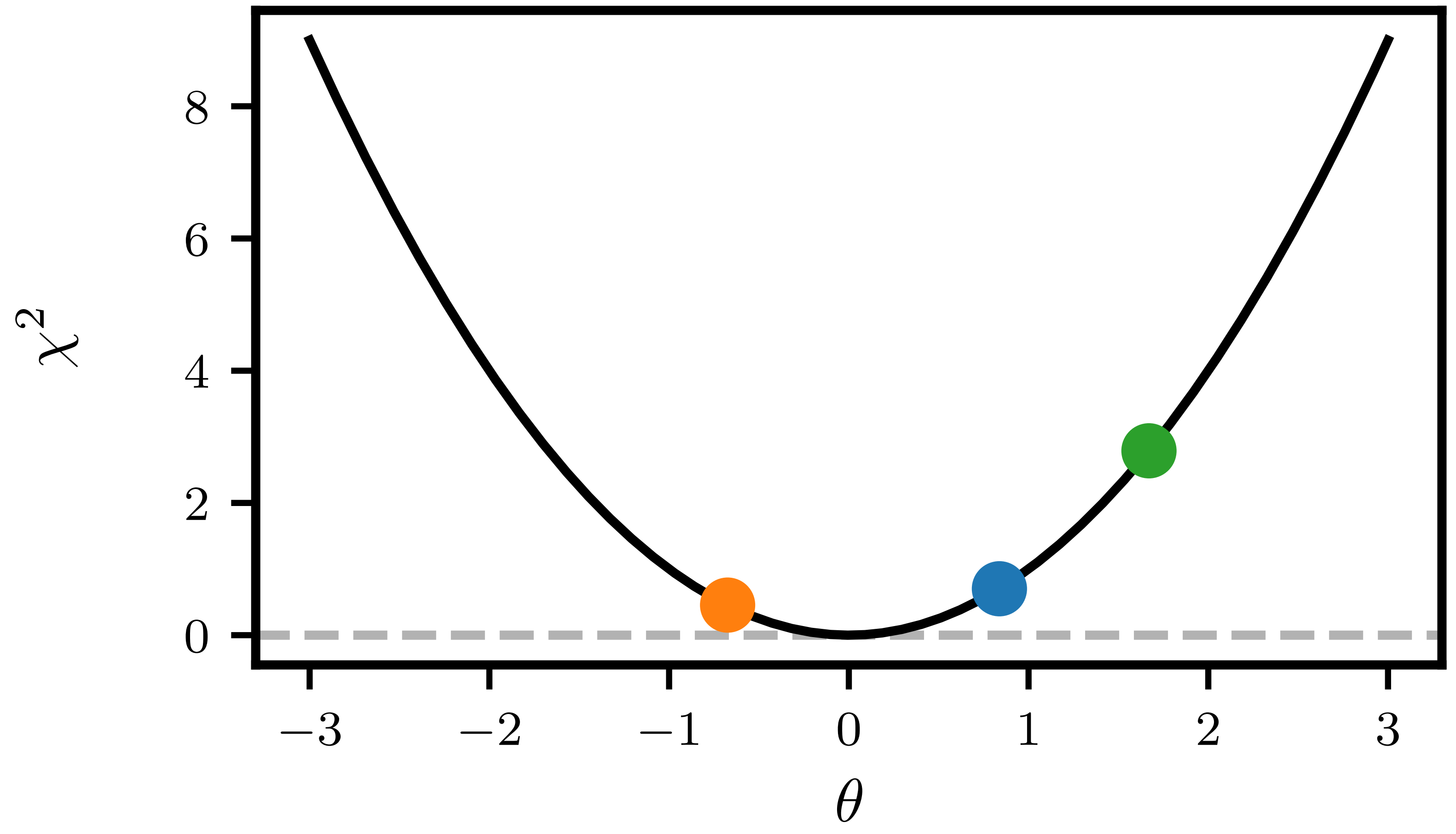
\+ Completely flexible
Most accurate

− Slow
Not always differentiable
Not all observables

Likelihoods

- **CMB with derivatives:**
  - candl (arXiv:2401.13433) [SPT, ACT]
  - clipy (github.com/benabed/clipy/tree/main/clipy) [Planck]
- **LSS with derivatives:**
  - desilike (github.com/cosmodesi/desilike)
  - cloe (github.com/cloe-org)
  - …probably more?
- **Generic surrogate likelihoods**
  - GPry (arXiv:2211.02045)
  - CLiENT (arXiv:2512.17509)

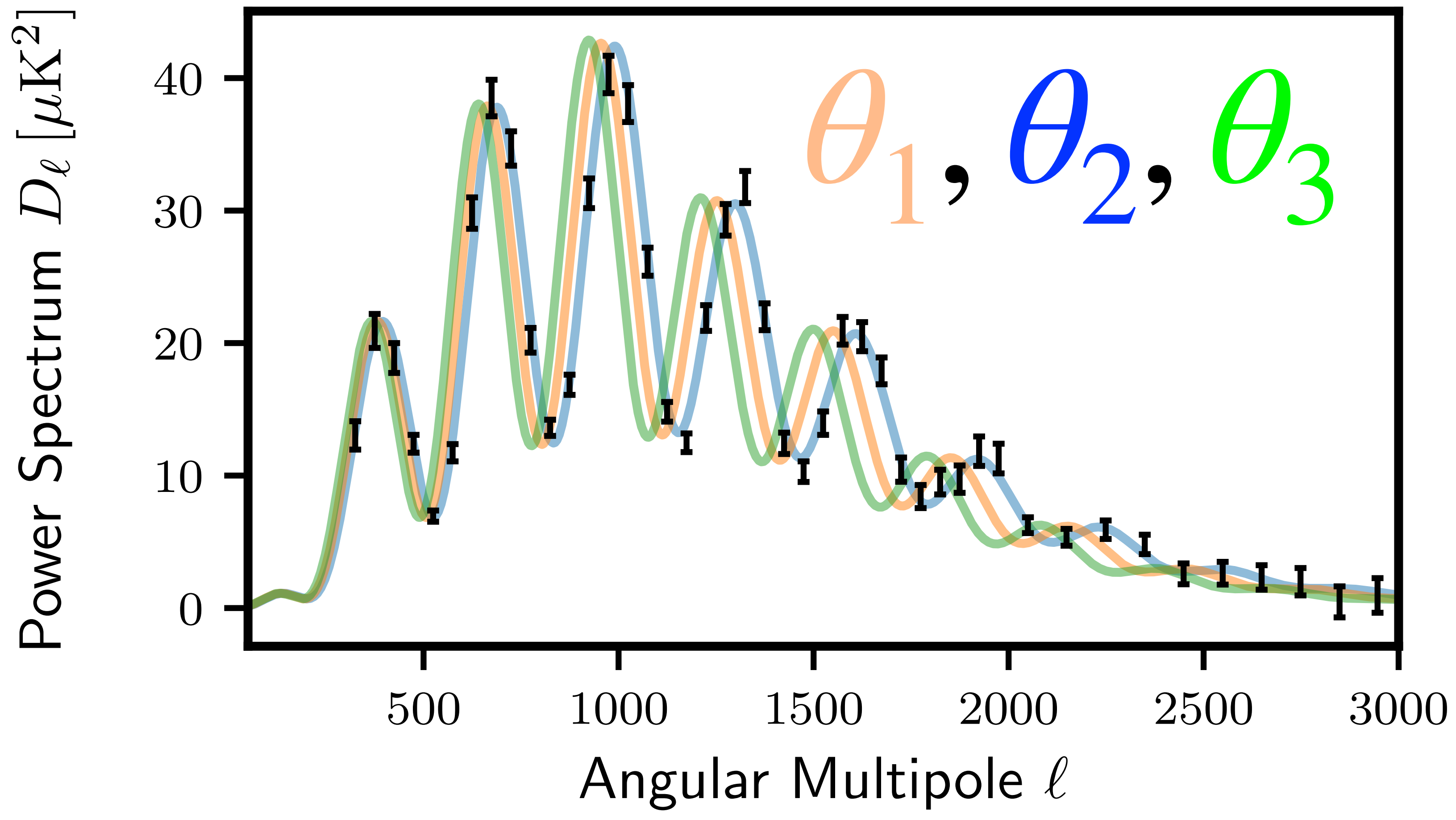
High flexibility
Full accuracy
Differentiable

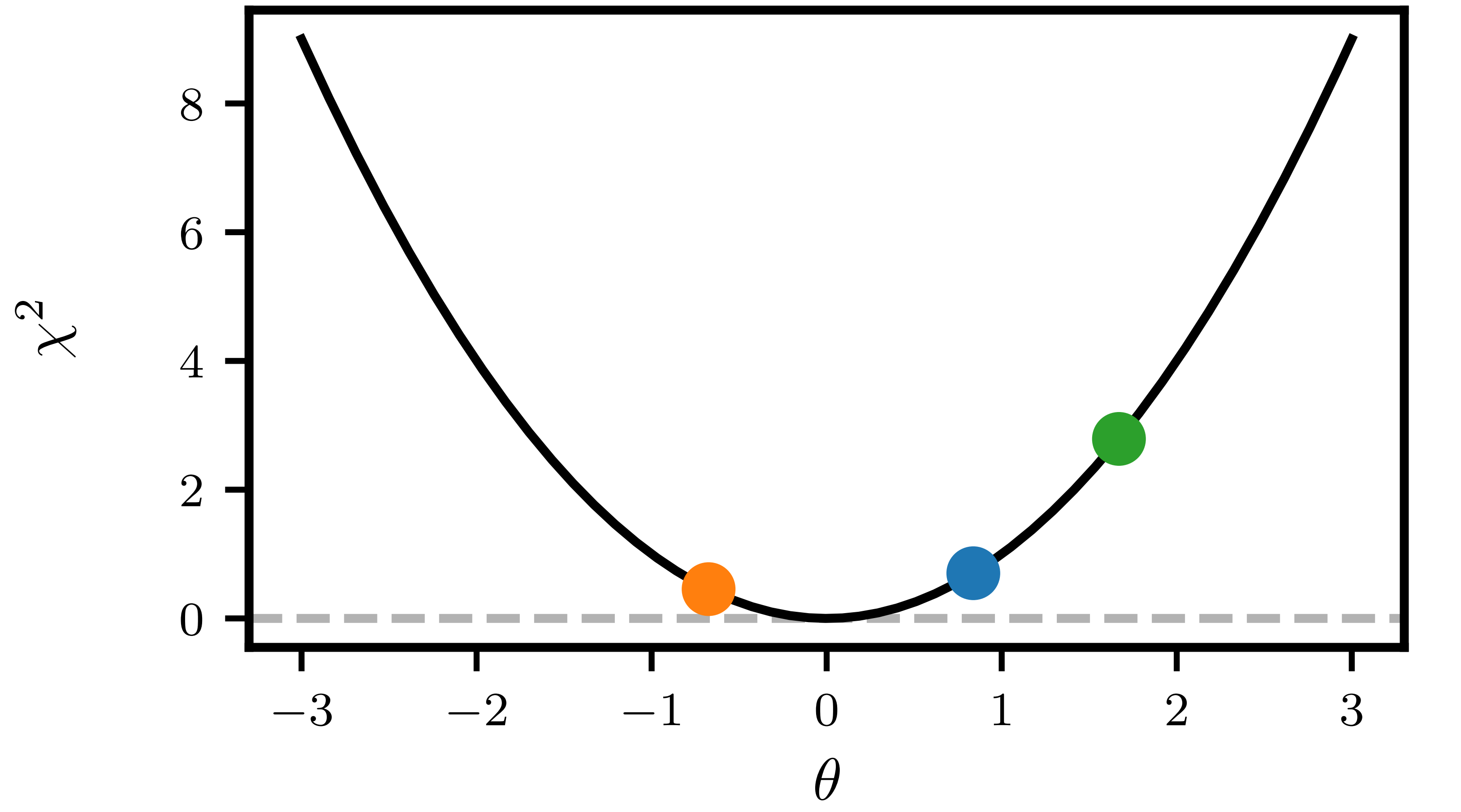


For diff. need diff. theory code

Likelihoods

- **CMB with derivatives:**
  - candl (arXiv:2401.13433) [SPT, ACT]
  - clipy (github.com/benabed/clipy/tree/main/clipy) [Planck]
- **LSS with derivatives:**
  - desilike (github.com/cosmodesi/desilike)
  - …probably more?
- **Generic surrogate likelihoods**
  - GPry (arXiv:2211.02045)
  - CLiENT (arXiv:2512.17509)

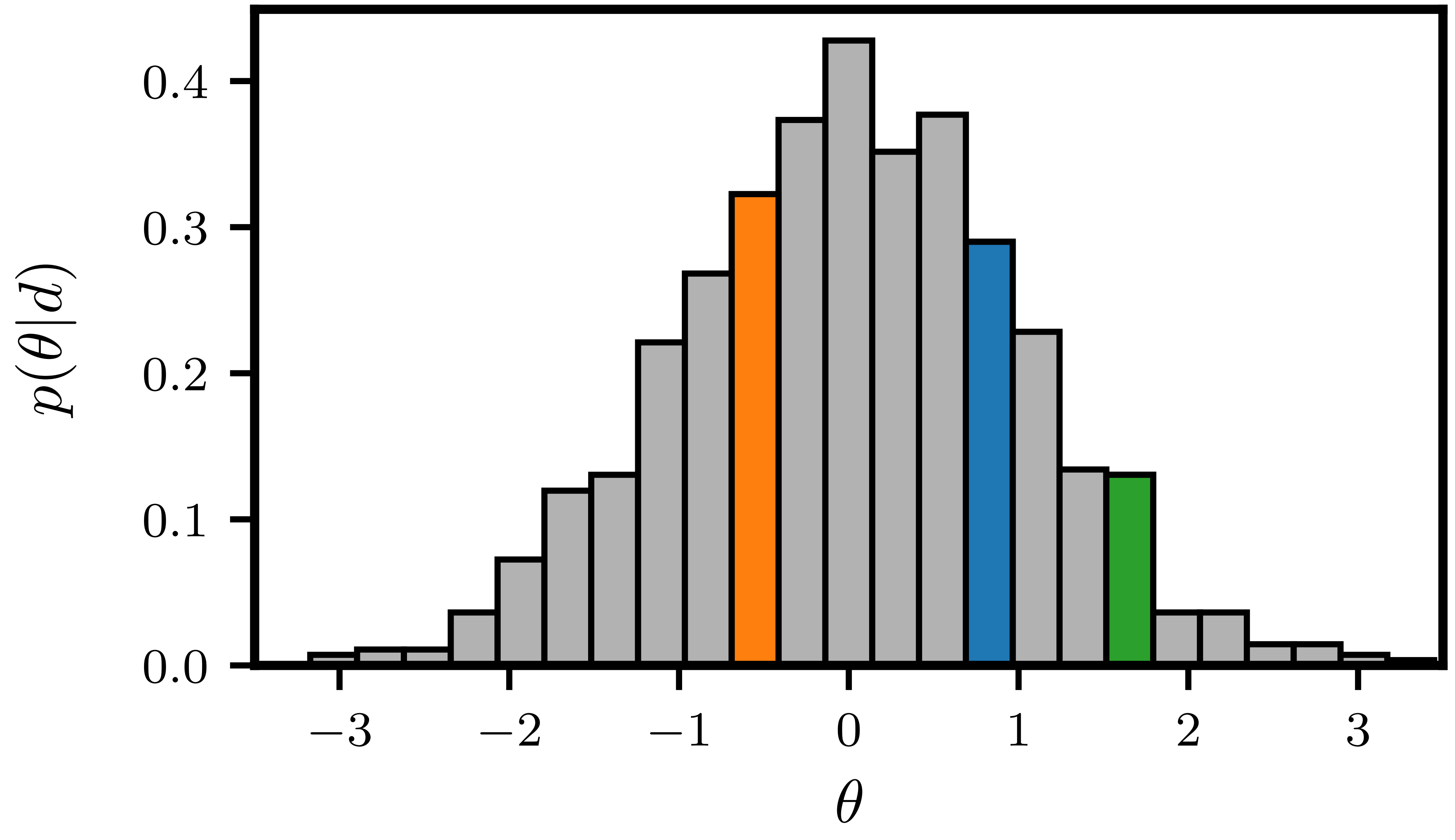




Generic
Differentiable

Low reusability
Training needed

Samplers



- **Cosmology frameworks:**
  - Cobaya (github.com/CobayaSampler/cobaya)
  - MontePython (github.com/brinckmann/montepython_public)
  - CosmoSIS (github.com/cosmosis-developers)

+ Speak our language

− Not differentiable
Only standard techniques

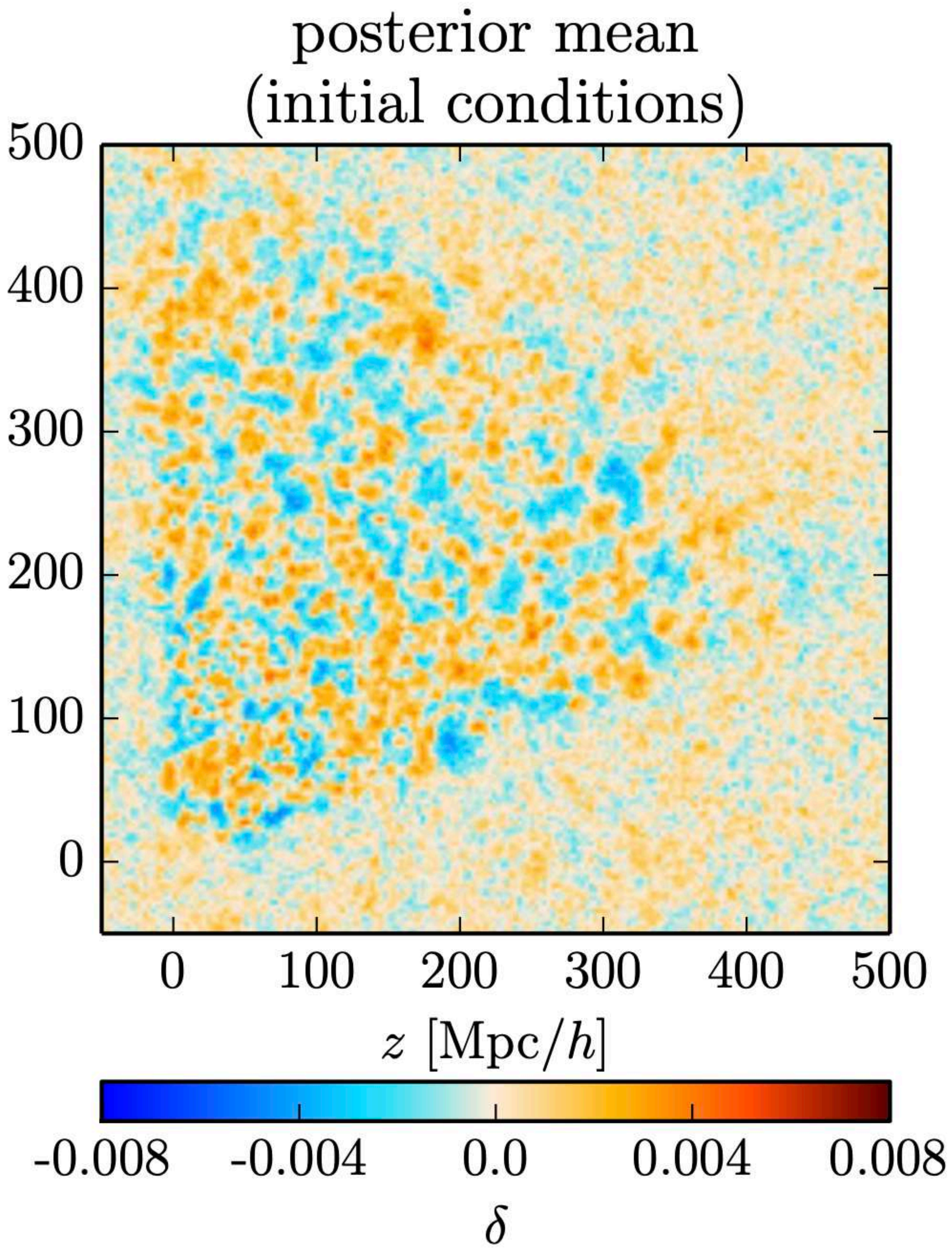
- **Generic frameworks:**
  - BlackJAX (github.com/blackjax-devs/blackjax)
  - Optax (github.com/google-deepmind/optax)
  - PyTorch (pytorch.org)
  - PyMC (github.com/pymc-devs)
  - Many more!

+ Industry codes
Many samplers available

− Interface learning curve

Beyond Line Fitting

- **<u>Field level inference</u>**
  - Forward modelling
  - Can be simulation-based, implicit likelihood
  - CMB e.g.: MUSE (Millea, Seljak, Ge, ++)
  - LSS e.g.: BORG (Lavaux, Jasche, ++), DISCO-DJ (Hahn, ++)



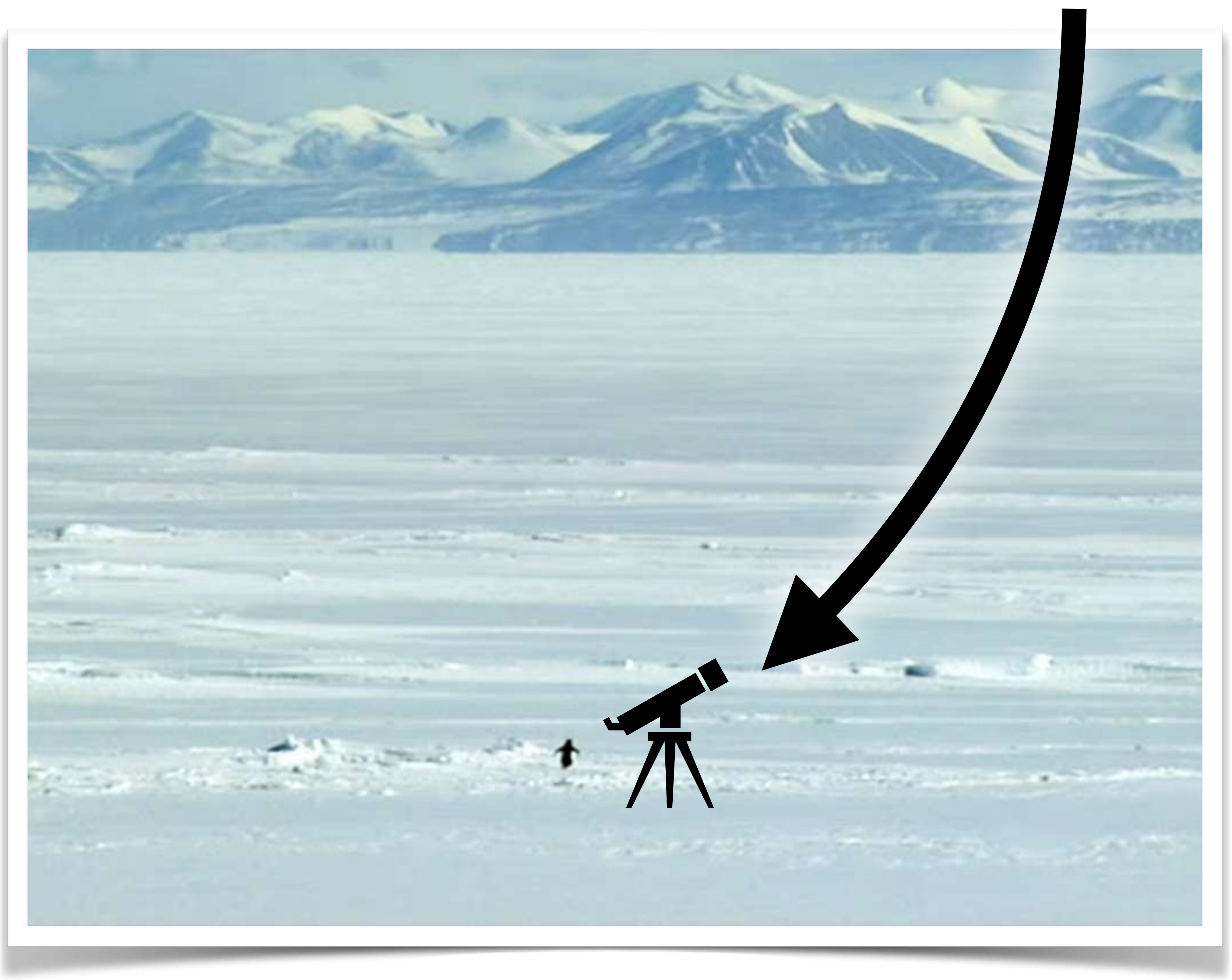
Use all information
Neatly fold in systematics
Framework for x-correlations

Computationally expensive
Use ALL information

Leclercq+ (arXiv:1512.02242)

Conlcusions

Prototype (if you can)!



Wish list:

- 1 pre-trained emulator to rule them all
- All new likelihoods differentiable
- Gradient-powered samplers in Cobaya, MontePython
- A differentiable Boltzmann solver for CMB T&E spectra (in JAX)
- ...