COSMOPOWER:

Deep Learning – accelerated cosmological inference from next-generation surveys

Alessio Spurio Mancini

with D. Piras, J. Alsing, B. Joachimi, M. Hobson, A. Pourtsidou

UCL
MULLARD SPACE SCIENCE LABORATORY

Based on
ASM+, MNRAS 511, 2022
ASM & Pourtsidou, MNRAS Letters 512, 2022

COSMOPOWER

ASM+, MNRAS 511, 2022

We introduce a suite of
neural cosmological power spectrum emulators
covering both CMB (temperature, polarization and lensing),
and large-scale structure power spectra

MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini

EMULATION

- Boltzmann solvers: computational **bottleneck** for 2pt stats analysis

- ***emulation*** speeds up theory predictions

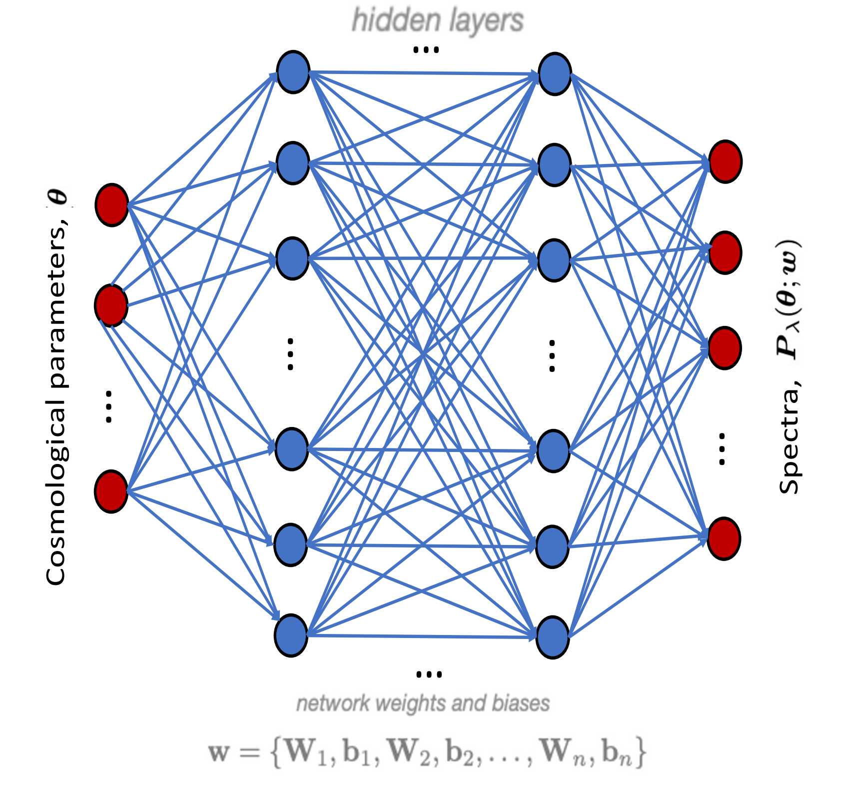
(Fendt & Wandelt 07, …, Aricò+22, Mootoovaloo+22)

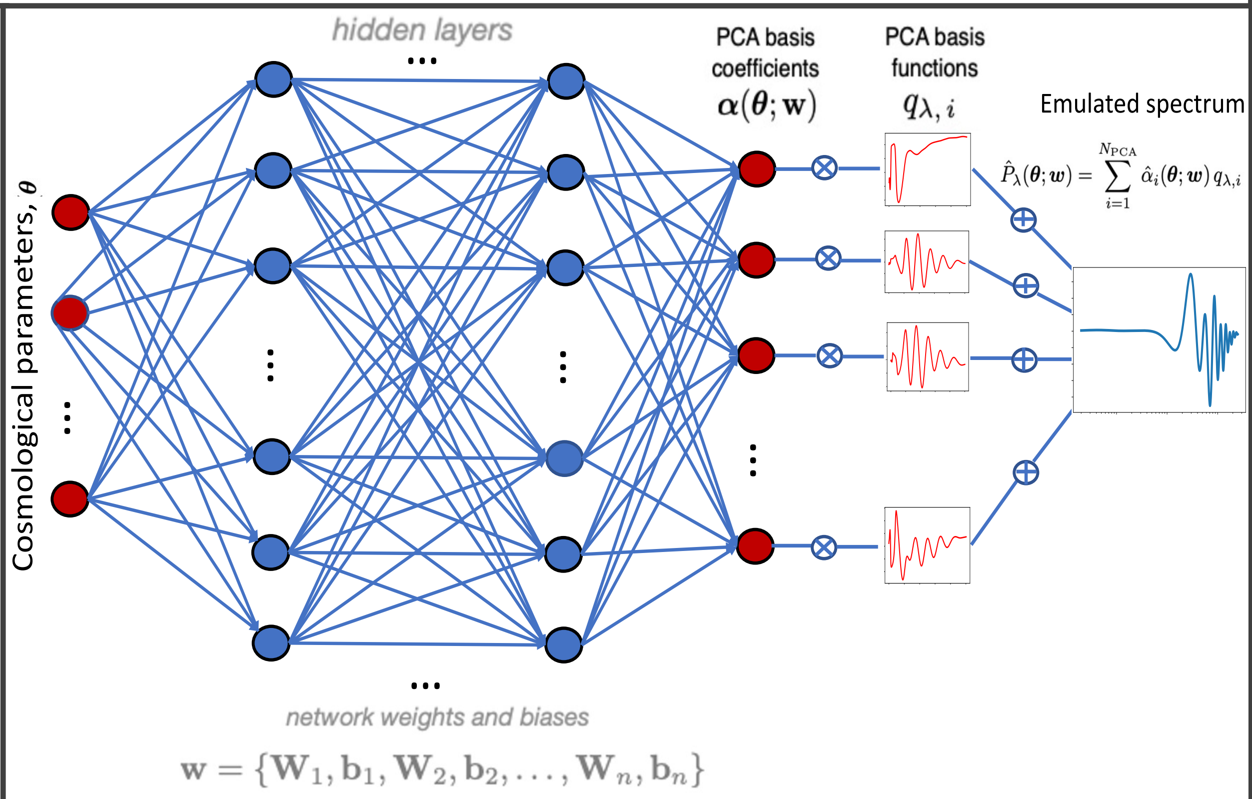
- a "simple" Machine Learning problem …
- … with exceptionally high levels of **accuracy** required!

→ need **flexible** and **well tested** tool. CosmoPower emulates

- P(k,z) (LIN+NL BOOST)
- CMB TT, TE, EE, $\phi\phi$

METHODS





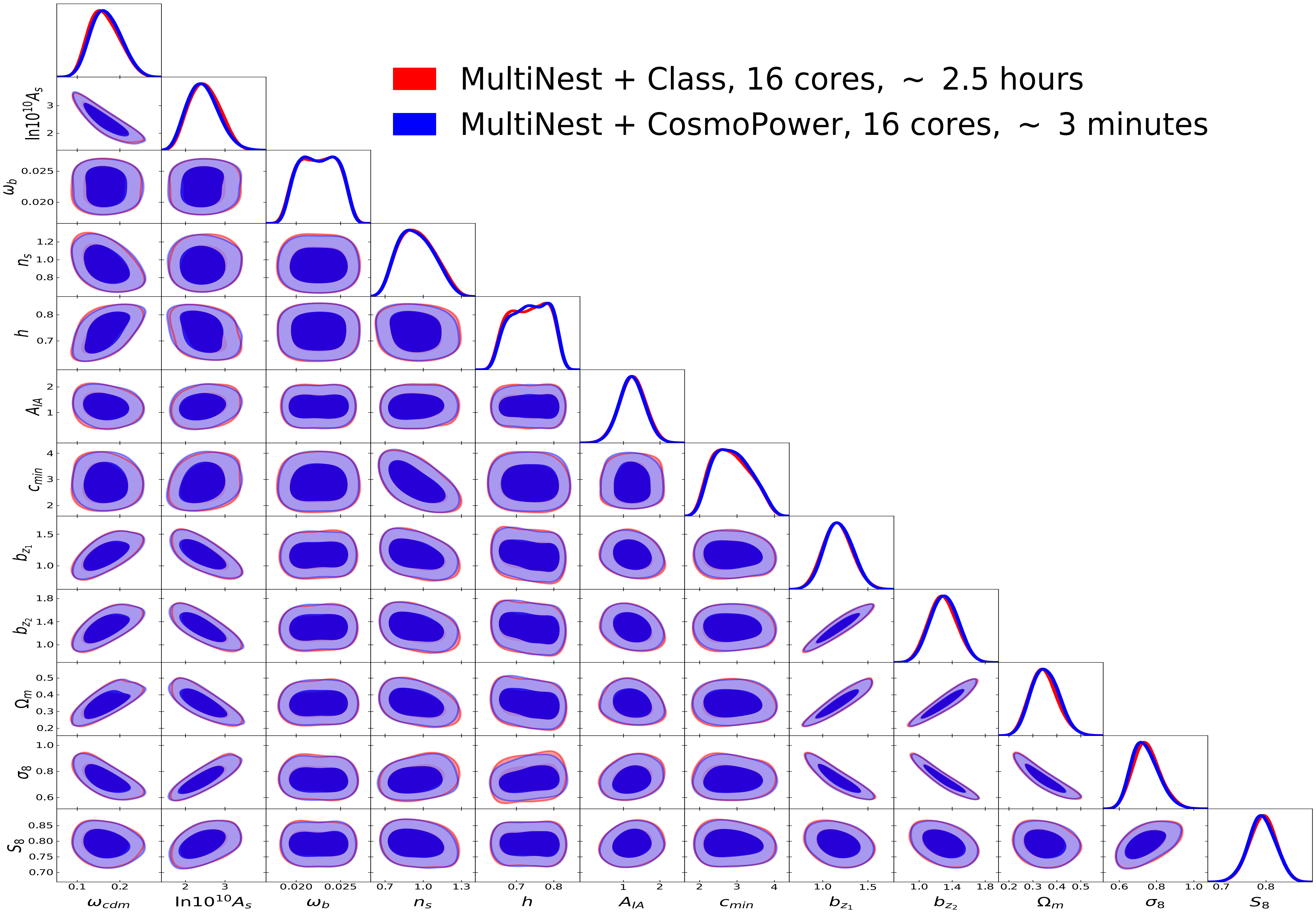
NEURAL NETWORK

NEURAL NETWORK + PCA

KIDS-450+GAMA 3X2PT

UCL
MULLARD SPACE
SCIENCE LABORATORY

A. Spurio
Mancini



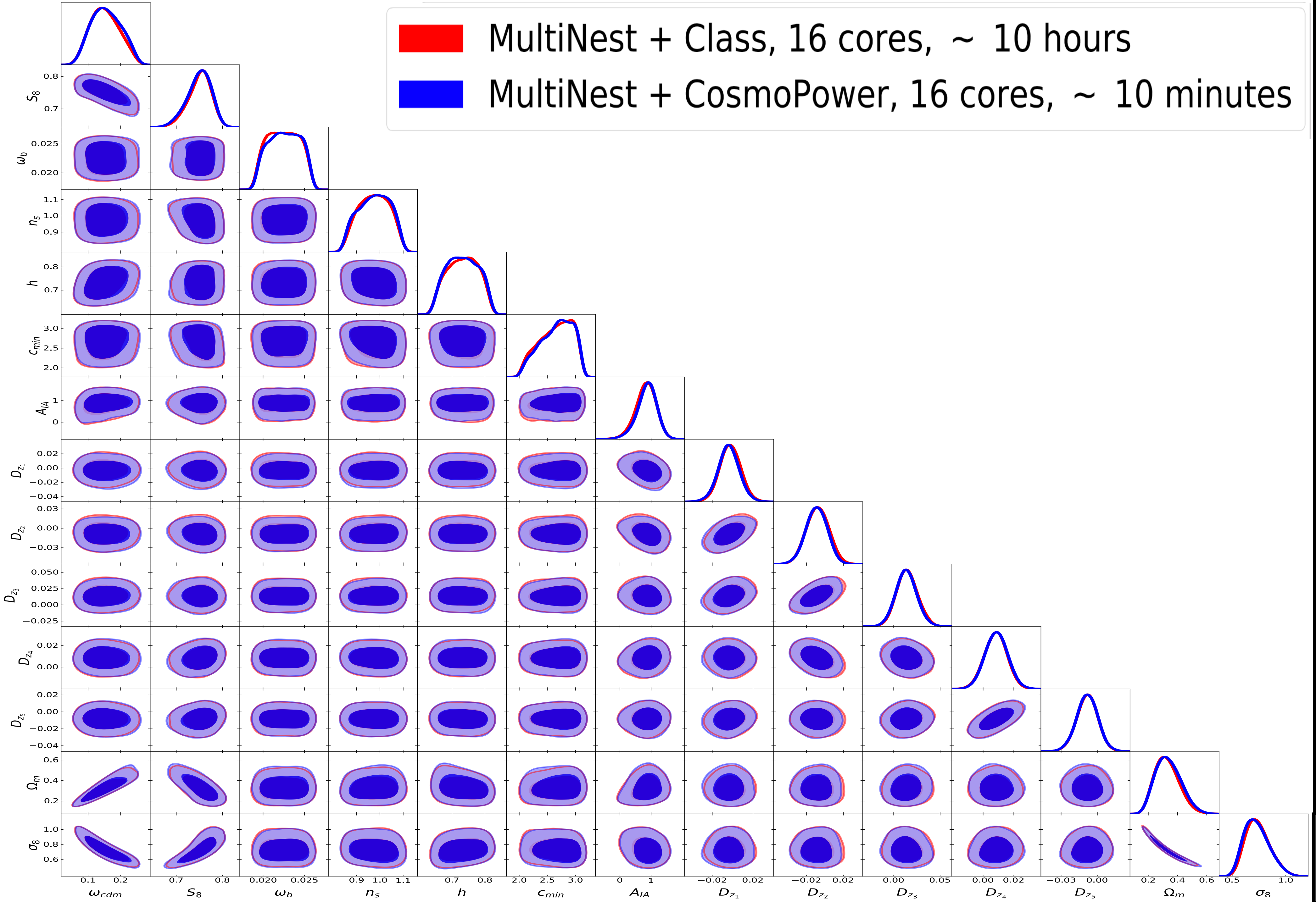
MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini

KIDS-1000 COSMIC SHEAR

MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini



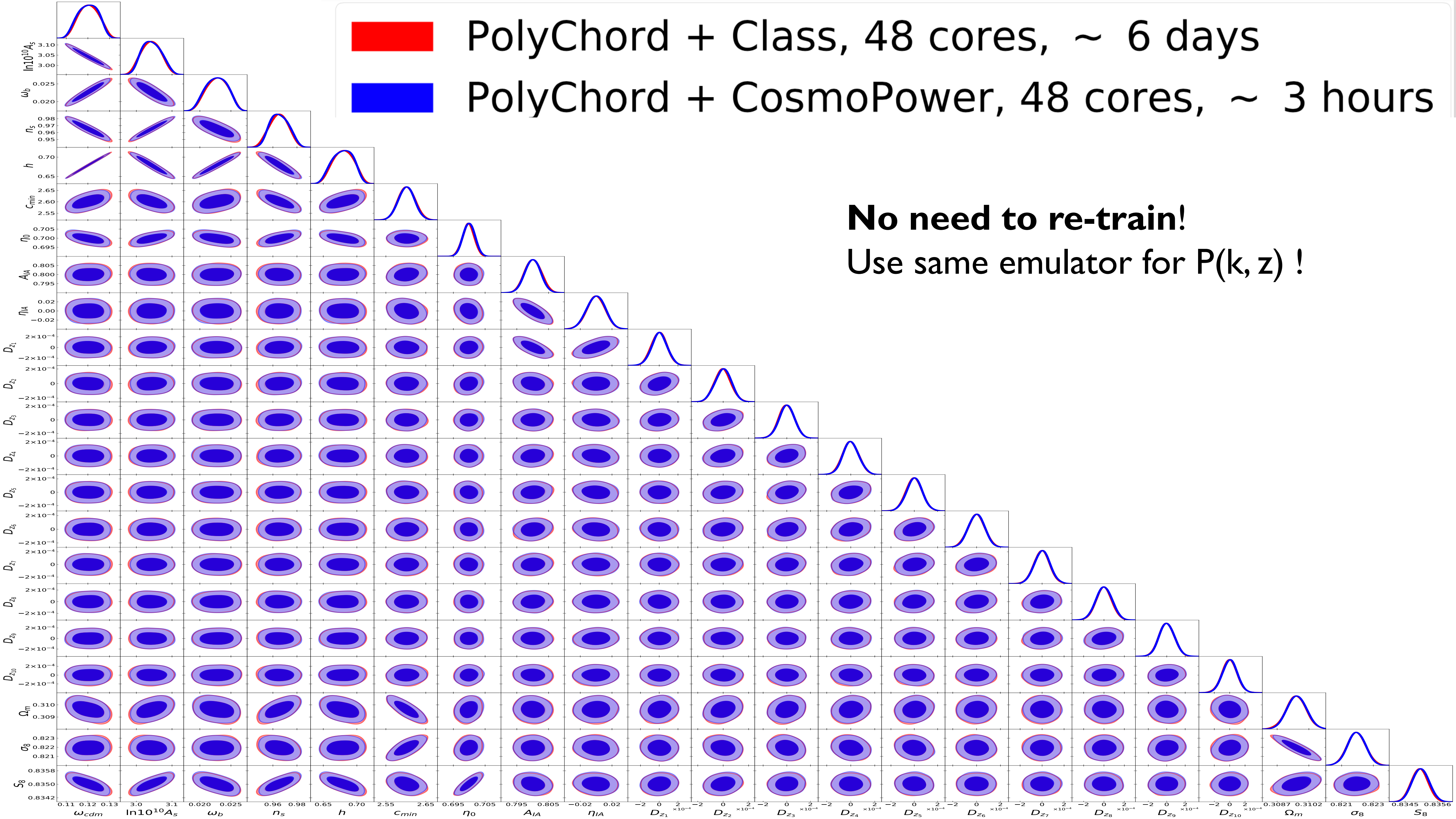
UCL
MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini

EUCLID-LIKE COSMIC SHEAR

MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini



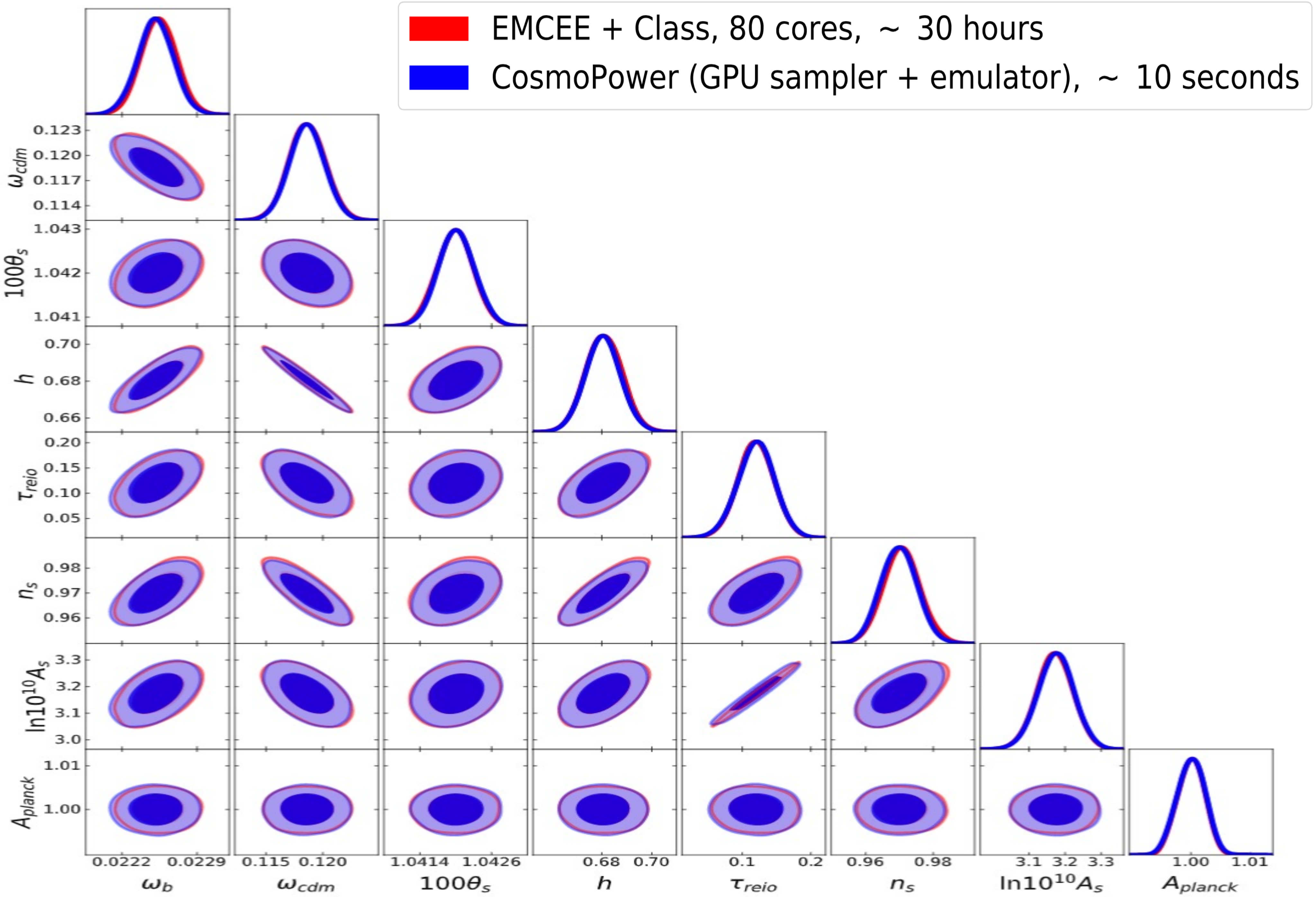
No need to re-train!

Use same emulator for P(k, z) !

PLANCK 2018 TTTEEE

MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini



UCL
MULLARD SPACE
SCIENCE LABORATORY

A. Spurio
Mancini

BEYOND ΛCDM

MULLARD SPACE
SCIENCE LABORATORY

A. Spurio
Mancini

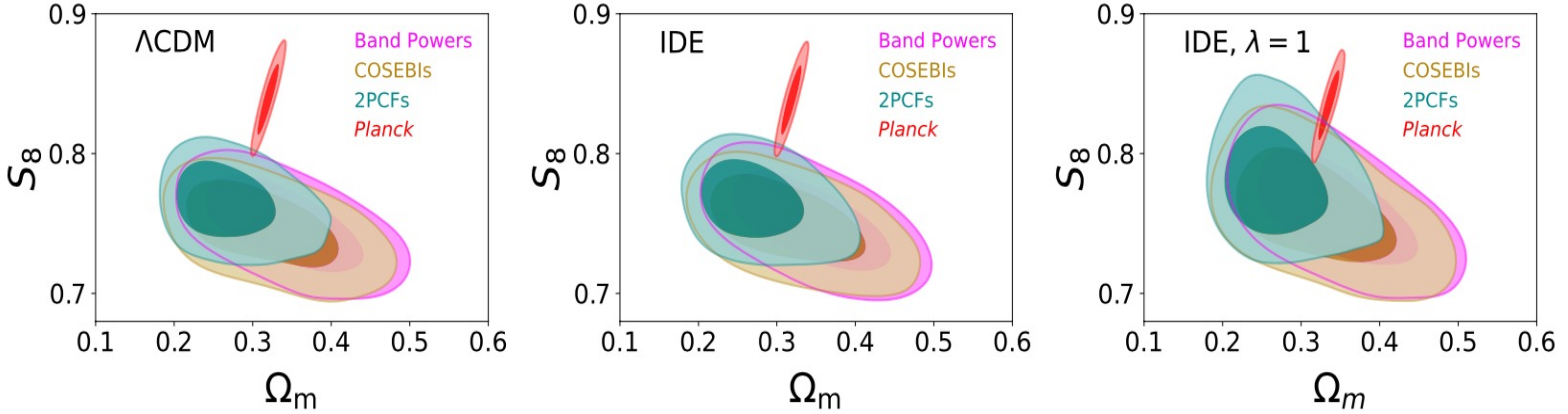
KIDS-1000 COSMOLOGY: MACHINE LEARNING - ACCELERATED CONSTRAINTS ON **INTERACTING DARK ENERGY** WITH COSMOPOWER

$$S_\phi = \int dt\, d^3x\, a^3 \left[\frac{1}{2}(1-2\beta)\dot{\phi}^2 - \frac{1}{2}|\vec{\nabla}\phi|^2 - V(\phi)\right]$$

$$V(\phi) = V_0 e^{-\lambda\phi}$$

Pourtsidou, Skordis & Copeland (2013)
Skordis, Pourtsidou & Copeland (2015)
Pourtsidou & Tram (2016)

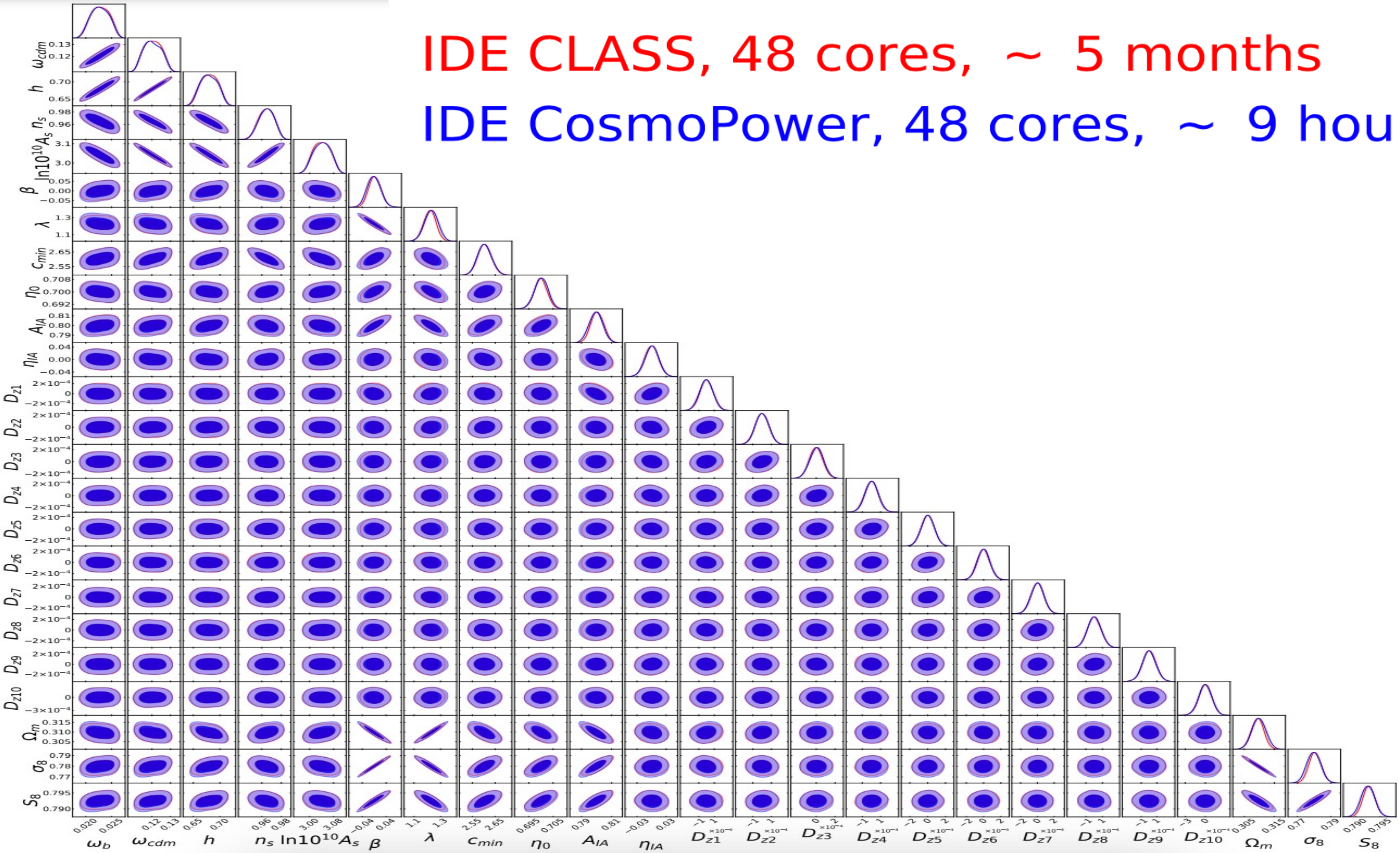
KIDS-1000 COSMOLOGY: MACHINE LEARNING - ACCELERATED CONSTRAINTS ON **INTERACTING DARK ENERGY** WITH COSMOPOWER



ASM & Pourtsidou, MNRAS Letters 512, 2022

MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini



IDE CLASS, 48 cores, $\sim$ 5 months

IDE CosmoPower, 48 cores, $\sim$ 9 hours

COSMOPOWER: MAIN FEATURES

- **Tested** at inference level + evidence, on **large parameter ranges**
- Massive **speed-up**: up to $O(10^4)$ on inference (even more beyond ΛCDM!)
- **Flexible**, no need to re-train e.g. for different n(z)'s
- Compatible with COBAYA, COSMOSIS, MONTEPYTHON…
- **GPU**/TPU: additional speed-up
- Fully **differentiable**

COSMOPOWER: GITHUB REPOSITORY

README.md

Python | Tensorflow | License GPLv3 | Author Alessio Spurio Mancini | Installation pip install cosmopower

Overview • Documentation • Installation • Getting Started • Training • Trained Models • Likelihoods • Support • Citation

Overview

`CosmoPower` is a library for Machine Learning - accelerated Bayesian inference. While the emphasis is on building algorithms to accelerate Bayesian inference in *cosmology*, the interdisciplinary nature of the methodologies implemented in the package allows for their application across a wide range of scientific fields. The ultimate goal of `CosmoPower` is to solve *inverse* problems in science, by developing Bayesian inference pipelines that leverage the computational power of Machine Learning to accelerate the inference process. This approach represents a principled application of Machine Learning to scientific research, with the Machine Learning component embedded within a rigorous framework for uncertainty quantification.

In cosmology, `CosmoPower` aims to become a fully *differentiable* library for cosmological analyses. Currently, `CosmoPower` provides neural network emulators of matter and Cosmic Microwave Background power spectra. These emulators can be used to replace Boltzmann codes such as CAMB or CLASS in cosmological inference pipelines, to source the power spectra needed for two-point statistics analyses. This provides orders-of-magnitude acceleration to the inference pipeline and integrates naturally with efficient techniques for sampling very high-dimensional parameter spaces. The power spectra emulators implemented in CosmoPower, and first presented in its release paper, have been applied to the analysis of real cosmological data from experiments, as well as having been tested against the accuracy requirements for the analysis of next-generation cosmological surveys.

`CosmoPower` is written entirely in Python. Neural networks are implemented using the TensorFlow library.

Documentation

Comprehensive documentation is available here.

UCL
MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini

Installation

We recommend installing `CosmoPower` within a Conda virtual environment. For example, to create and activate an env
called `cp_env`, use:

```
conda create -n cp_env python=3.7 pip && conda activate cp_env
```

Once inside the environment, you can install `CosmoPower`:

- **from PyPI**

  ```
  pip install cosmopower
  ```

  To test the installation, you can use

  ```
  python3 -c 'import cosmopower as cp'
  ```

getting_started_with_cosmopower_NN.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

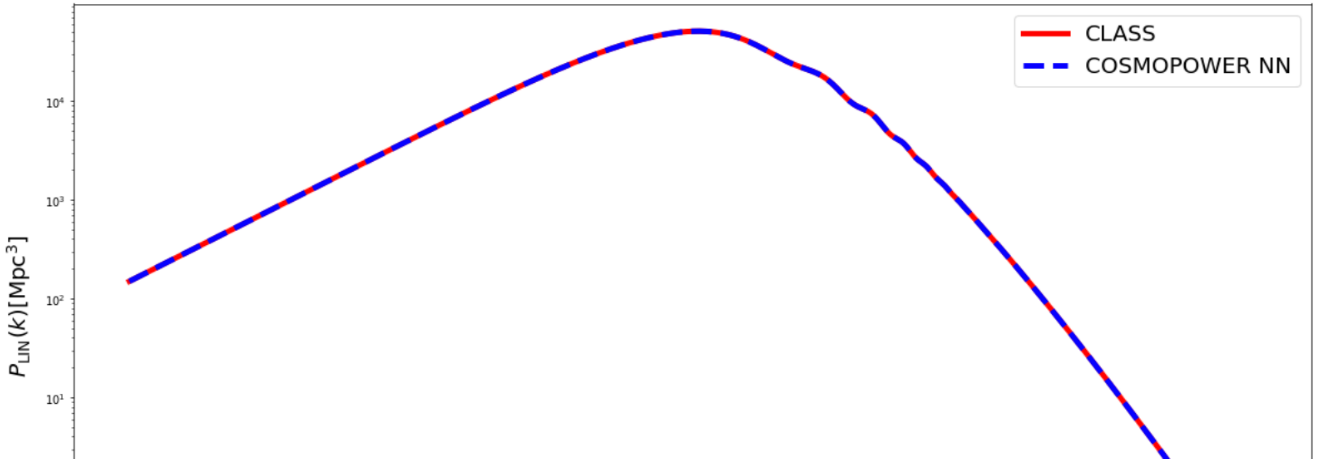
COMPARING with CLASS

```
k_modes = cp_nn.modes

cosmo = Class()

# Define your cosmology (what is not specified will be set to CLASS default parameters)
params = {'output': 'tCl mPk',
          'z_max_pk': 5,
          'P_k_max_1/Mpc': 10.,
          'nonlinear_min_k_max': 100.,
          'N_ncdm' : 0,
          'N_eff' : 3.046,
          'omega_b': 0.0225,
          'omega_cdm': 0.113,
          'h': 0.7,
          'n_s': 0.96,
          'ln10^{10}A_s': 3.07,
          }

# Set the parameters to the cosmological code
cosmo.set(params)
cosmo.compute()

z = 0.5

spectrum_class = np.array([cosmo.pk(ki, z) for ki in k_modes])
```

[ ]

```
pred = spectrum_cosmopower_NN
true = spectrum_class
fig = plt.figure(figsize=(20,10))
plt.loglog(k_modes, true, 'red', linewidth=5, label = 'CLASS')
plt.loglog(k_modes, pred, 'blue', label = 'COSMOPOWER NN', linewidth=5, linestyle='--')
plt.xlabel('$k$ [Mpc$^{-1}]$', fontsize=20)
plt.ylabel('$P_{\mathrm{LIN}}(k) [\mathrm{Mpc}^3]$', fontsize=20)
plt.legend(fontsize=20)
```

<matplotlib.legend.Legend at 0x7f25dd360fd0>



MULLARD SPACE SCIENCE LABORATORY

A. Spurio Mancini

COSMOPOWER: FUTURE WORK

A fully differentiable library for cosmology:

- Beyond ΛCDM
- Beyond-Limber
- Higher-order statistics
- Systematics
- … and more!

https://github.com/alessiospuriomancini/cosmopower

a.spuriomancini@ucl.ac.uk

THANK YOU!