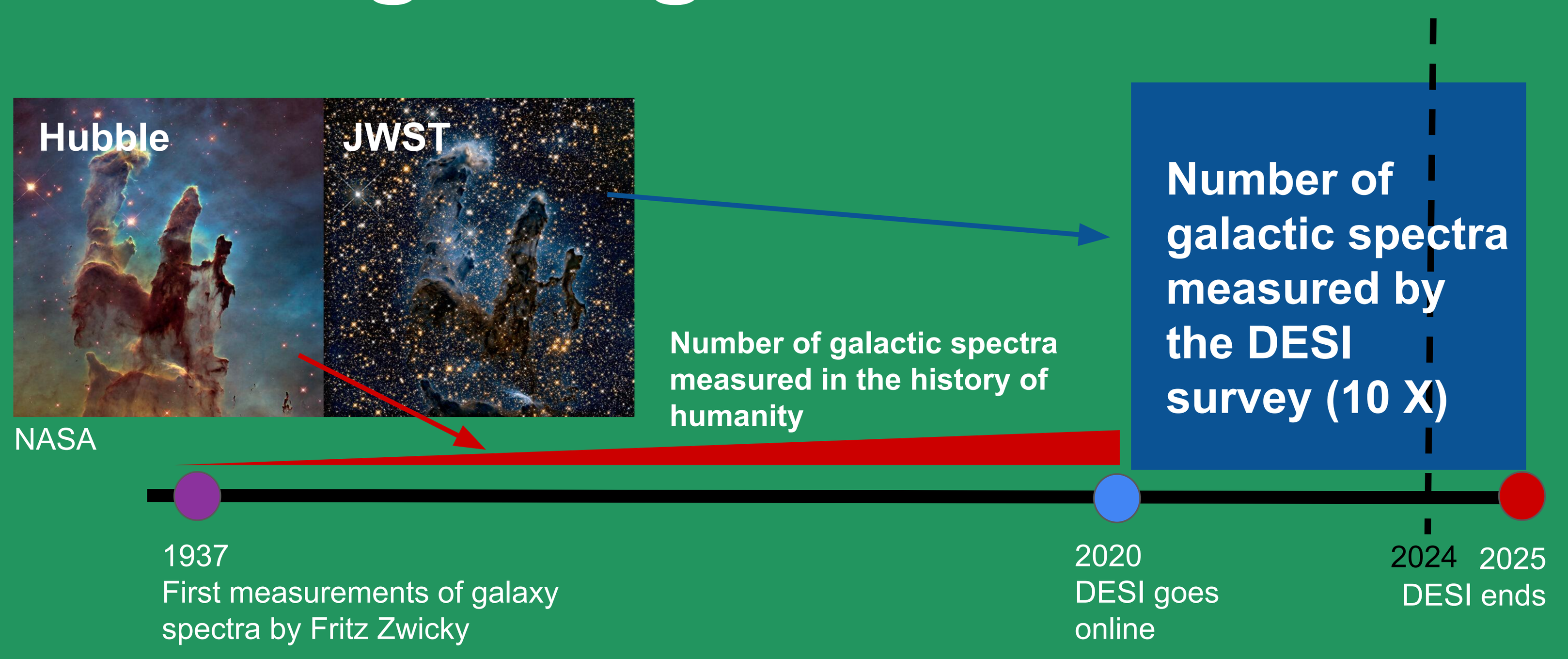
LimberJack.jl

Auto-differentiable methods for Cosmology in Julia

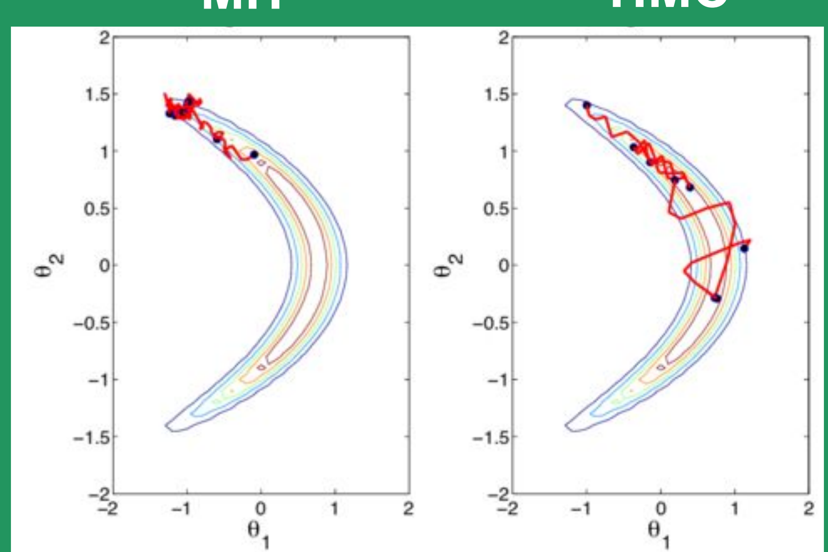
The problem: Cosmology is getting *a lot* of data



MORE DATA → **MORE PARAMETERS** → **THAT WE CANNOT CONSTRAIN**

- Traditional inference methods such as Metropolis hastings (MH) can constrain **tens** of parameters
- Up coming cosmological analysis will have **hundreds** of parameters to account for extra systematic effects

The Solution: Hamiltonian Monte Carlo (HMC)

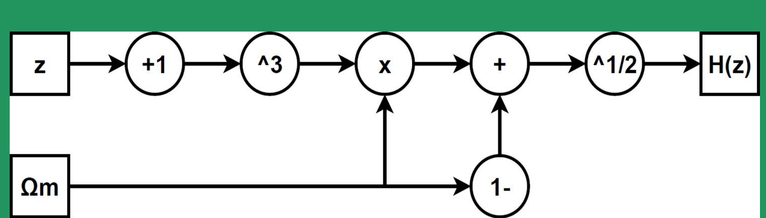


Comparison between MH and HMC exploring a 2D parameter space by Lagrangian Dynamical Monte Carlo (Lan et al, 2012)

- HMC simulates hamiltonian trajectories to explore large parameter spaces very efficiently
- In order to do so, HMC requires the **gradient** of our theory predictions with respect to said parameters

How to get cheap gradients: Auto-differentiation

$$H(z) = \sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)}$$



Symbolic representation of a computer program generated by auto-differentiation

- Auto-differentiation allows computers to chain rule through computer programs by building a symbolic representation of the code.

The Julia ecosystem

Turing.jl

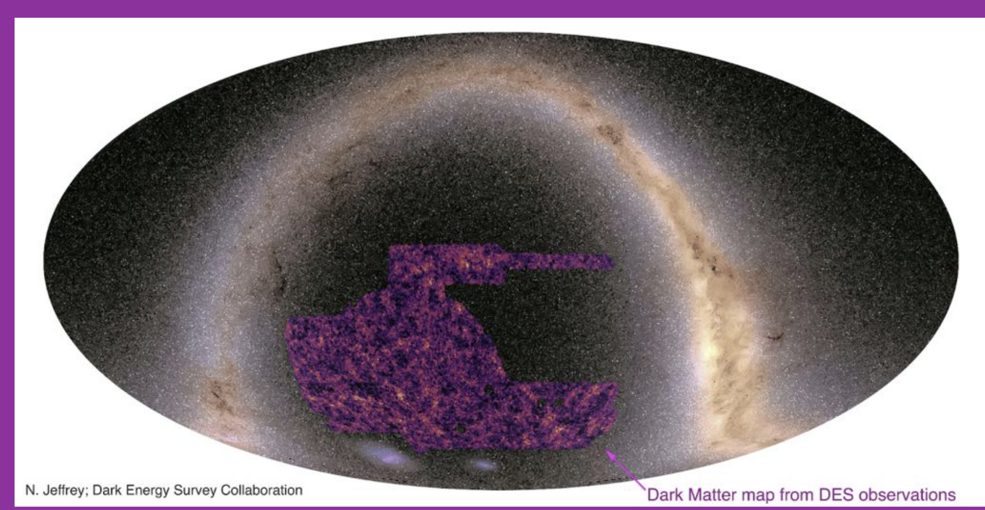
LimberJack.jl

- **Turing.jl:** provides an statistical inference framework with a native HMC implementation
- **LimberJack.jl:** provides auto-differentiable theory predictions for cosmological observables.
- Combining the two high-dimensional cosmological analyses are possible

An Example: Auto-differentiable analysis of angular correlation functions

The Data:

the DES-Y1 data is composed of 300 million galaxies across 500 square degrees on the sky



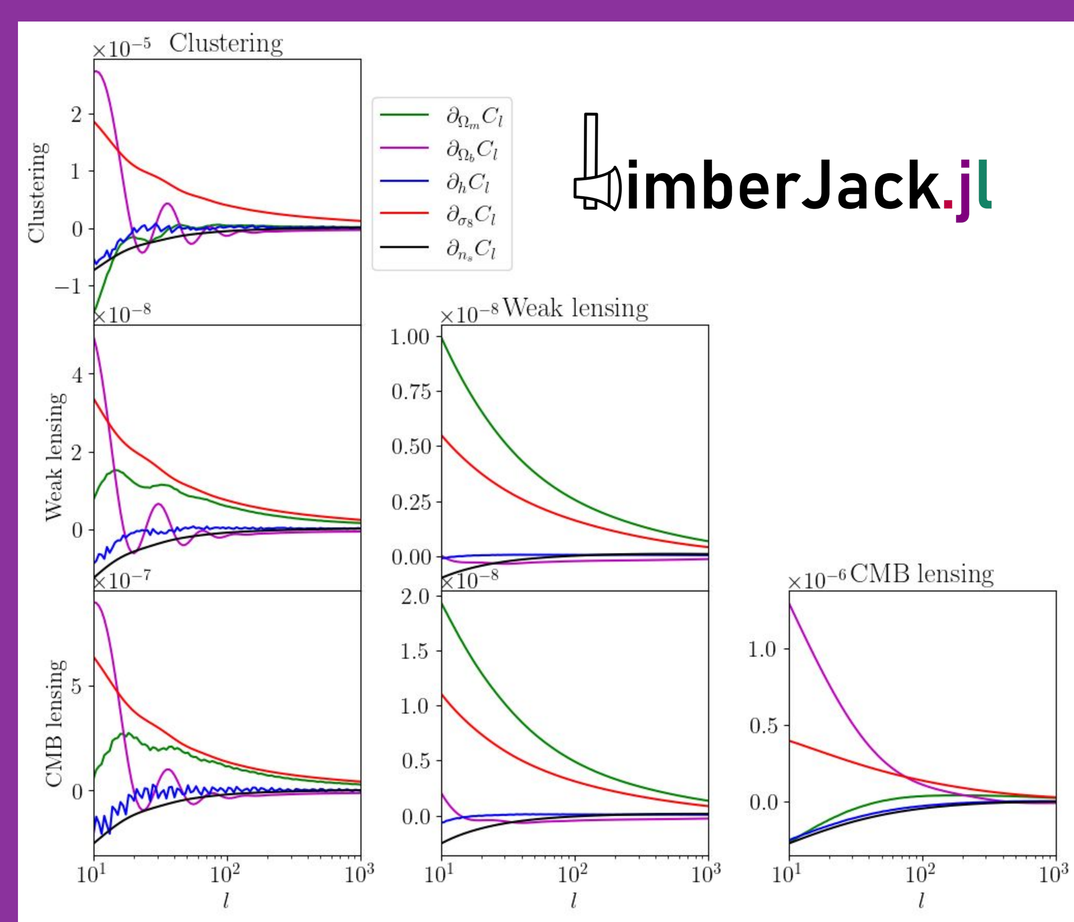
Footprint of the DES Y1 survey compared to our view of the milky way

The Theory:

$$C_\ell^{UV} = \int \frac{d\chi}{\chi^2} q_U(\chi) q_V(\chi) P_{UV}\left(k = \frac{\ell + 1/2}{\chi}, z(\chi)\right)$$

We predict the angular (on the sky) 2 point correlation function of these galaxies and their properties.

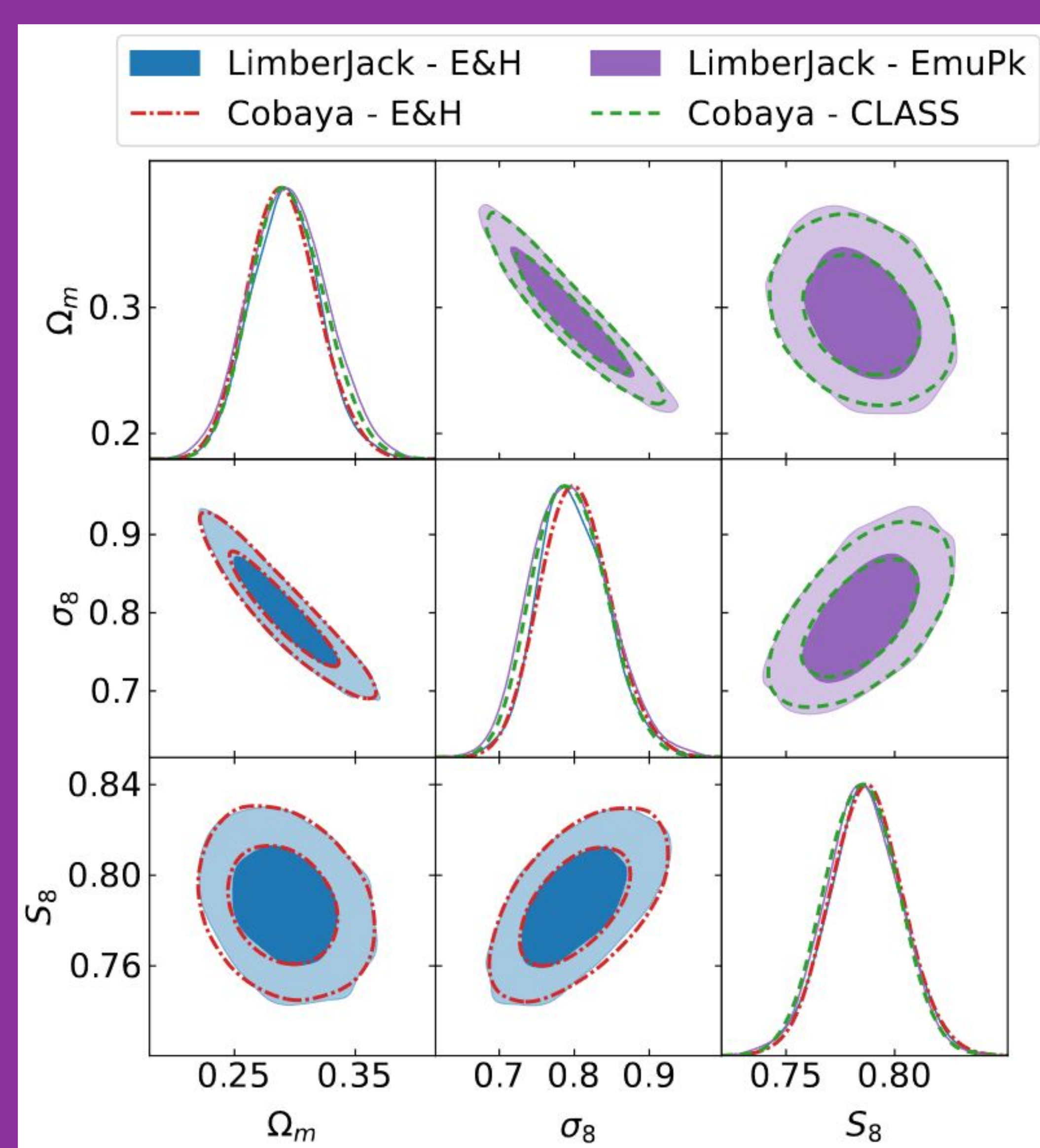
- LimberJack uses mix of emulation and auto-differentiable programming to evaluate these correlation functions and their gradients efficiently
- Emulation employs machine learning to build approximations to difficult computations



Gradients of the different 2-point correlation functions with respect the five main cosmological parameters computed by LimberJack

Results:

- Identical parameter constraints than state-of art-software (Cobaya)
- Order of magnitude faster gradients than finite differences
- Enables efficient high-dimensional Cosmological analyses
- Gradient-based inference methods not worth it for low dimensions



Lower triangle compares LimberJack to Cobaya when both use approximations to compute their theory.
Upper triangle shows comparisons when Limberjack employs emulation to match exact computations.
In this plot Cobaya uses MH and LimberJack HMC.

Jaime Ruiz Zapatero: jaime.ruiz-zapatero@ucl.ac.uk
RSE at UCL working on LSST and Euclid

Based of ***LimberJack.jl: auto-differentiable methods for angular power spectra analyses*** *by J. Ruiz-Zapatero, D. Alonso, C. García-García, A. Nicola, A. Mootoovaloo, J. M. Sullivan, M. Bonici, P. G. Ferreira*

Github

Arxiv