# pyhf

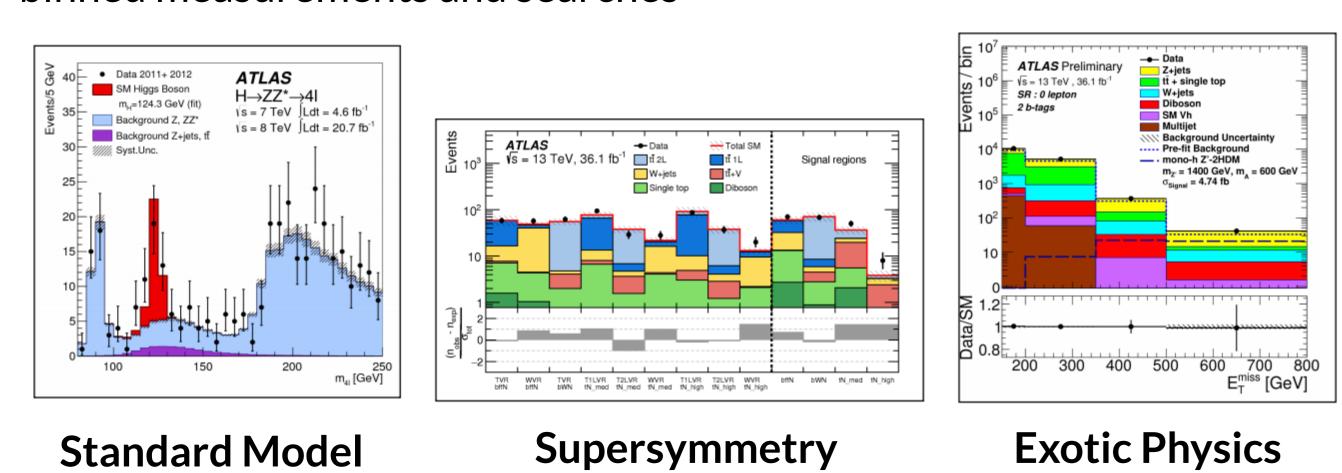
### pure Python implementation of HistFactory

Matthew Feickert<sup>1</sup>, Lukas Heinrich<sup>2</sup>, Giordon Stark<sup>3</sup>, Kyle Cranmer<sup>4</sup>

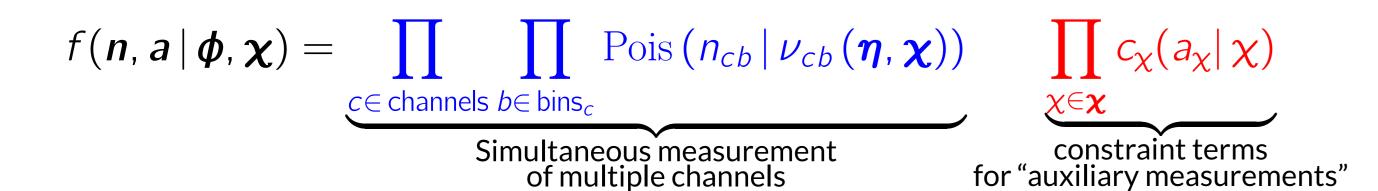
1 Southern Methodist University, 2 CERN, 3 University of California Santa Cruz, 4 New York University

# HistFactory

One of the most widely used statistical models in high energy physics for binned measurements and searches



### Declarative binned likelihoods



### **Primary Measurement:**

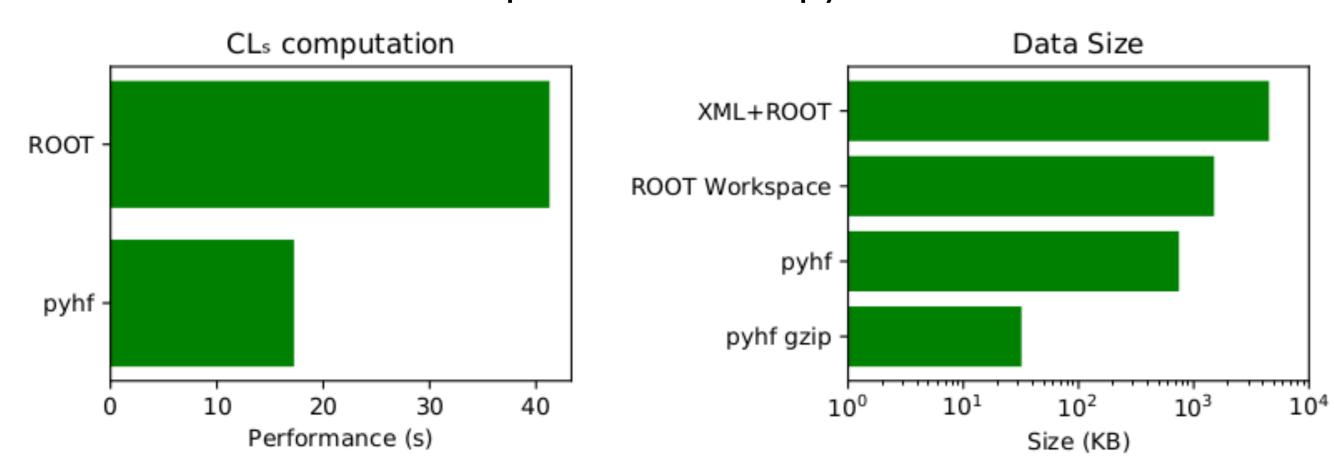
- Multiple disjoint "channels" (e.g. event observables) each with multiple bins of data
- ullet Example parameter of interest: strength of physics signal,  $\mu$

#### Auxiliary Measurements:

- Nuisance parameters (e.g. in-situ measurements of background samples)
- Systematic uncertainties (e.g. normalization, shape, luminosity)

### Performance

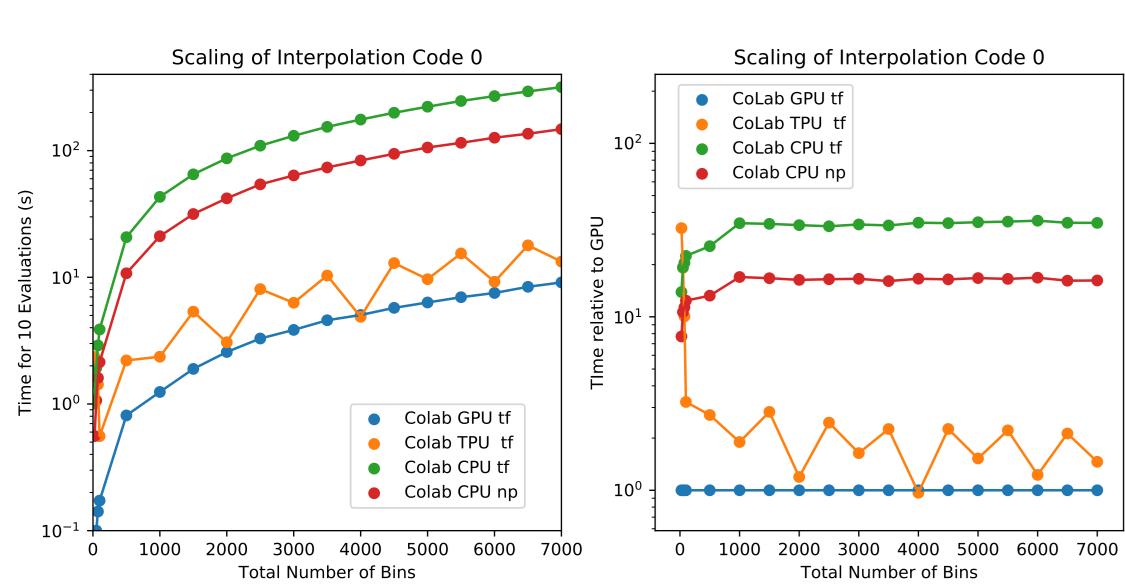
Efficient use of tensor computation makes pyhf fast



Competitive with traditional C++ implementation — often faster

### Hardware Acceleration

For ML-library tensor backends the computational graph can be transparently placed on hardware accelerators: GPUs and TPUs for order of magnitude speed-up in computation

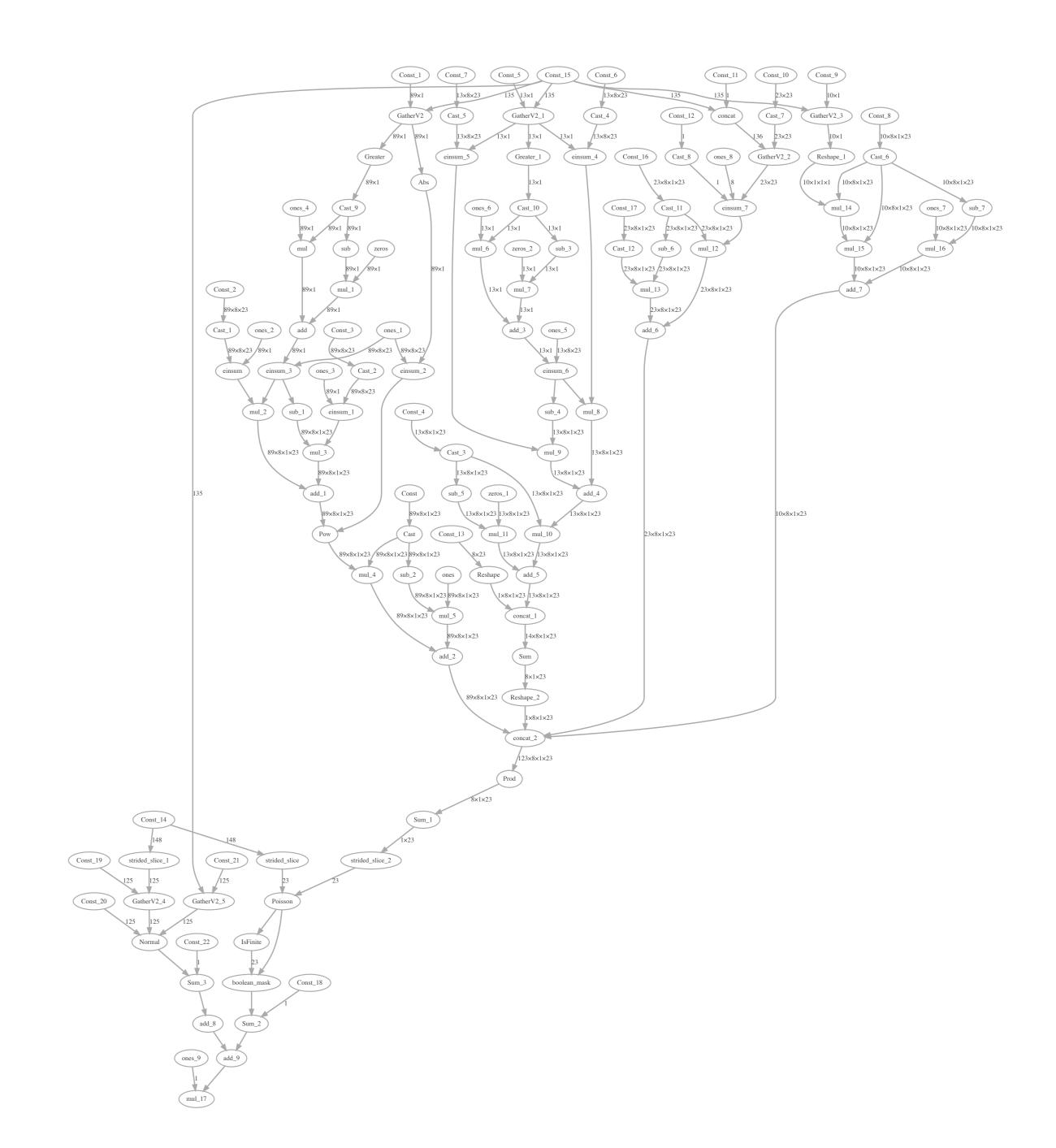


pyhf is a pure Python statistical fitting library that uses tensors and autograd to speed up physics analysis at the 

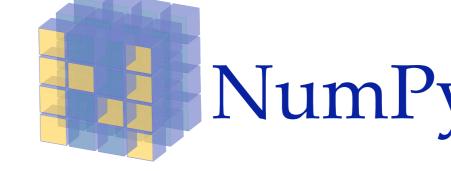




# Implementation



The computational graph of multidimensional array operations for likelihood function of a physics analysis defined through HistFactory







Use of n-dimensional array ("tensor") operations through a common API layer around high performance tensor libraries

# JSON Specification

The full likelihood can be expressed as a single JSON document Archive friendly for analysis presentation



**Example:** 2 binned single channel with 2 samples with 1 parameter of interest and 1 nuisance parameter ırl https://raw.githubusercontent.com/diana-hep/pyhf/master/examples/2-bin\_1-channel.json | pyhf cls "CLs\_exp": [ 0.07807427911686152 0.17472571775474582 0.35998495263681274 0.6343568235898907 0.8809947004472013 'CLs obs": 0.3599845631401913

