**pyhf**

pure Python implementation of HistFactory

Matthew Feickert\(^1\), Lukas Heinrich\(^2\), Giordon Stark\(^3\), Kyle Cranmer\(^4\)

\(^1\) University of Illinois at Urbana-Champaign, \(^2\) CERN, \(^3\) University of California Santa Cruz, \(^4\) New York University

![PyPi Package](https://pypi.org/project/pyhf/0.1.2)

DOI: 10.5281/zenodo.1169739

**HistFactory**

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

Standard Model  
Supersymmetry  
Exotics

**Declarative binned likelihoods**

\[
f(n, a | \phi, \chi) = \prod_{c \in \text{channels}} \prod_{b \in \text{bins}} \text{Pois}(n_{cb} | n_{cb}(\eta, \chi)) \prod_{\chi \in \chi} \text{c}_\chi(a | \chi)
\]

Primary Measurement:
- Multiple disjoint “channels” (e.g. event observables) each with multiple bins of data
- 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

![Performance Graph](https://example.com/performance.png)

Competitive with traditional C++ implementation — often faster

**Hardware Acceleration**

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

![Scaling Graph](https://example.com/scaling.png)

**Automatic Differentiation**

Tensor libraries from machine learning frameworks provide exact gradients of computational graphs (likelihood). Useful for minimization!

**Optimizer**

Support optimizers that implement Newton’s method as well as **scipy.optimize** and MINUIT. More advanced optimizers to come soon!