zfit

scalable pythonic fitting

Jonas Eschle

jonas.eschle@cern.ch

In collaboration with
A. Puig, R. S. Coutinho, N. Serra

University of Zurich[UZH]

FNSNF
Swiss National Science Foundation

What is `zfit`?

zfit
scalable pythonic fitting

Fitting for HEP
(~RooFit Core)

Fresh API, codebase
(inspired by others)

scalable pythonic fitting

Scalable

zfit
scalable pythonic fitting

- TensorFlow **hidden** backend, uses graphs
  - numpy-like syntax
  - parallelization on CPU/GPU, analytic gradient,...
- Writing functions simple for users *and* developers
  - No Cython, MPI, CUDA,... for *state-of-the-art performance*
  - No low-level maintenance required!
- Used in multiple physics libraries and analyses

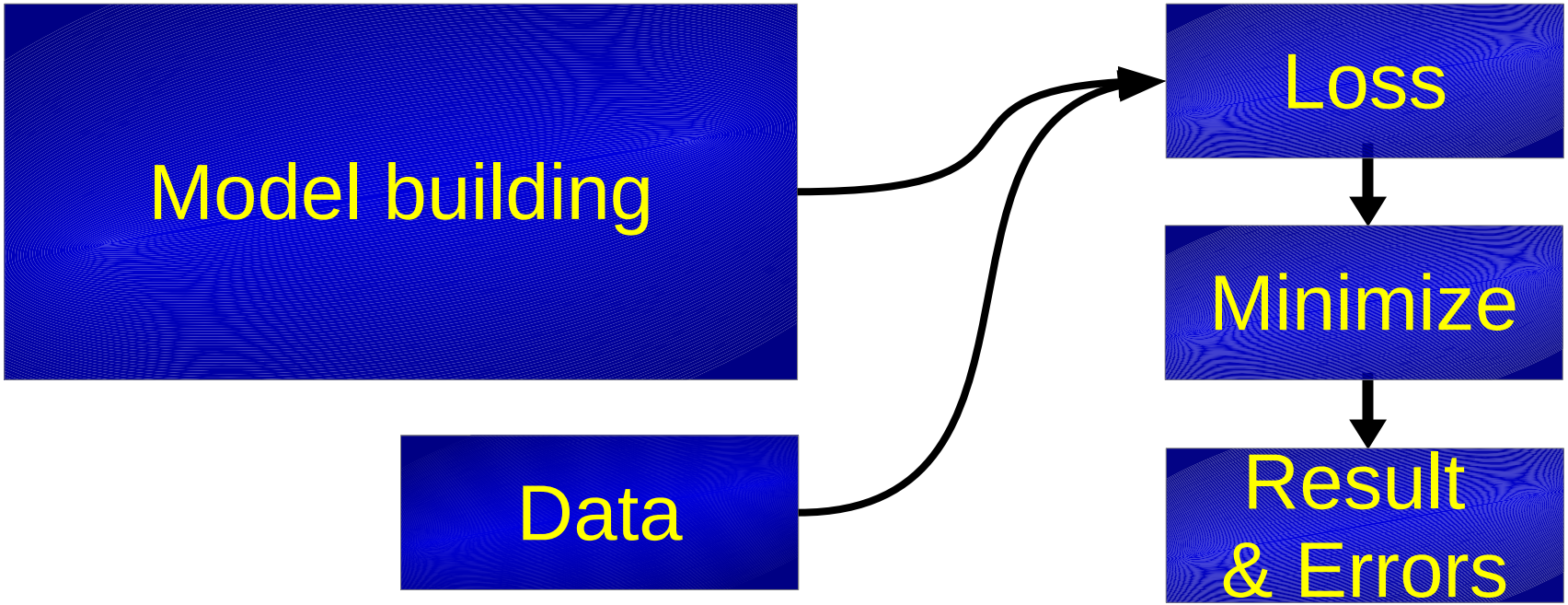
Pythonic

zfit
scalable pythonic fitting

- Pure Python
- Integrated into python ecosystem
  - Focus *only* on one thing
  - Integrate seamlessly with data, plotting and statistic packages
    - Uses uproot for ROOT files (no ROOT dependency!)
    - Used by lauztat for statistics
    - Can be used by phasespace for resonance modeling
- Easily extendable
  - e.g. custom PDF

Fitting

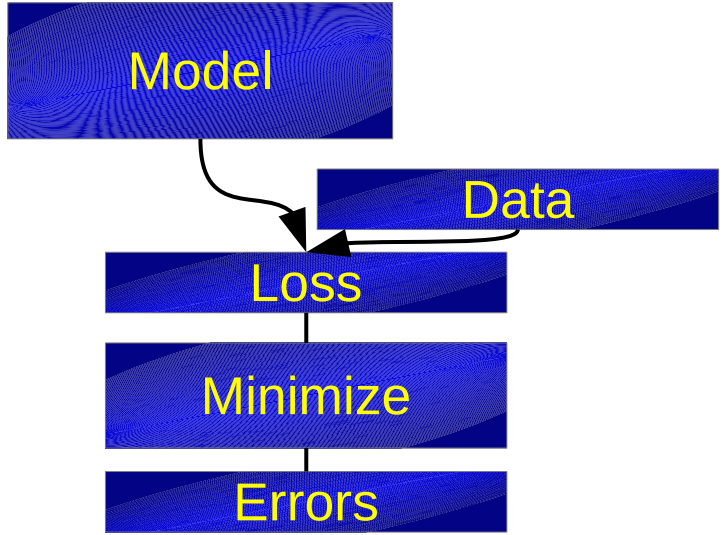
zfit
scalable pythonic fitting



Complete fit

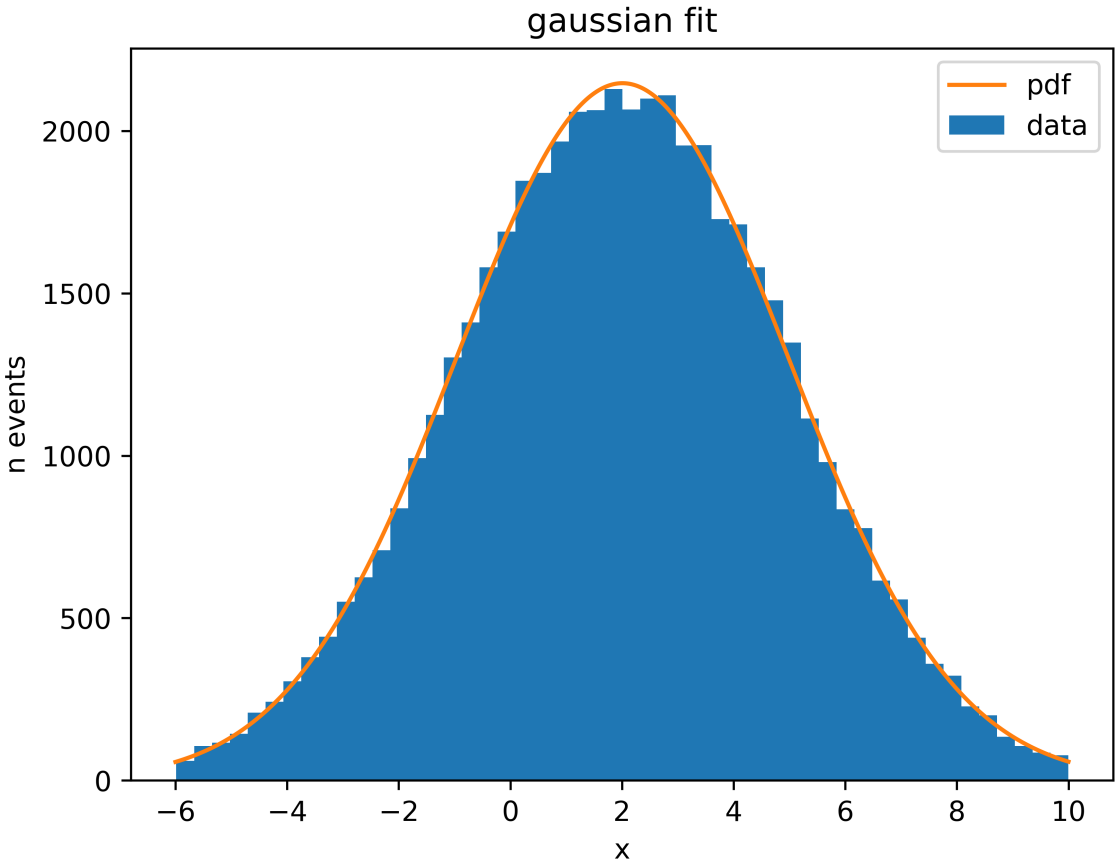
zfit
scalable pythonic fitting

```
import zfit

normal_np = np.random.normal(loc=2., scale=3., size=10000)

obs = zfit.Space("x", limits=(-10, 10))

mu    = zfit.Parameter("mu",    1., -5,  5)
sigma = zfit.Parameter("sigma", 3.,  1, 10)
gauss = zfit.pdf.Gauss(mu=mu, sigma=sigma, obs=obs)

data = zfit.data.Data.from_numpy(obs=obs, array=normal_np)

nll = zfit.loss.UnbinnedNLL(model=gauss, data=data)

minimizer = zfit.minimize.MinuitMinimizer()
result = minimizer.minimize(nll)

param_errors = result.error()
```



Complete fit: plot

zfit
scalable pythonic fitting



Gaussian PDF fitted to gaussian distributed sample

Model building

zfit
scalable pythonic fitting

```
obs = zfit.Space("x", limits=(-10, 10))

mu =    zfit.Parameter("mu",          1,  -4,  6)
sigma = zfit.Parameter("sigma",       1, 0.1, 10)
lambd = zfit.Parameter("lambda",     -1,  -5,  0)
frac =  zfit.Parameter("fraction", 0.5,   0,  1)

gauss = zfit.pdf.Gauss(mu=mu, sigma=sigma, obs=obs)
exponential = zfit.pdf.Exponential(lambd, obs=obs)

sum_pdf = zfit.pdf.SumPDF([gauss, exponential], fracs=frac)

sum_pdf = frac * gauss + exponential
```

parameters

models

equivalent

Custom PDF

zfit
scalable pythonic fitting

```
from zfit import ztf

class CustomPDF(zfit.pdf.ZPDF):
    _PARAMS = ['alpha']

    def _unnormalized_pdf(self, x):
        data = x.unstack_x()
        alpha = self.params['alpha']

        return ztf.exp(alpha * data)
```

implement custom function

```
custom_pdf = CustomPDF(obs=obs, alpha=0.2)

integral = custom_pdf.integrate(limits=(-1, 2))
sample   = custom_pdf.sample(n=1000)
prob     = custom_pdf.pdf(sample)
```

use functionality of model

Simultaneous fit

zfit
scalable pythonic fitting

```
mu_shared = zfit.Parameter("mu_shared", 1., -4, 6)
sigma1 = zfit.Parameter("sigma_one", 1., 0.1, 10)
sigma2 = zfit.Parameter("sigma_two", 1., 0.1, 10)

gauss1 = zfit.pdf.Gauss(mu=mu_shared, sigma=sigma1, obs=obs)
gauss2 = zfit.pdf.Gauss(mu=mu_shared, sigma=sigma2, obs=obs)
```

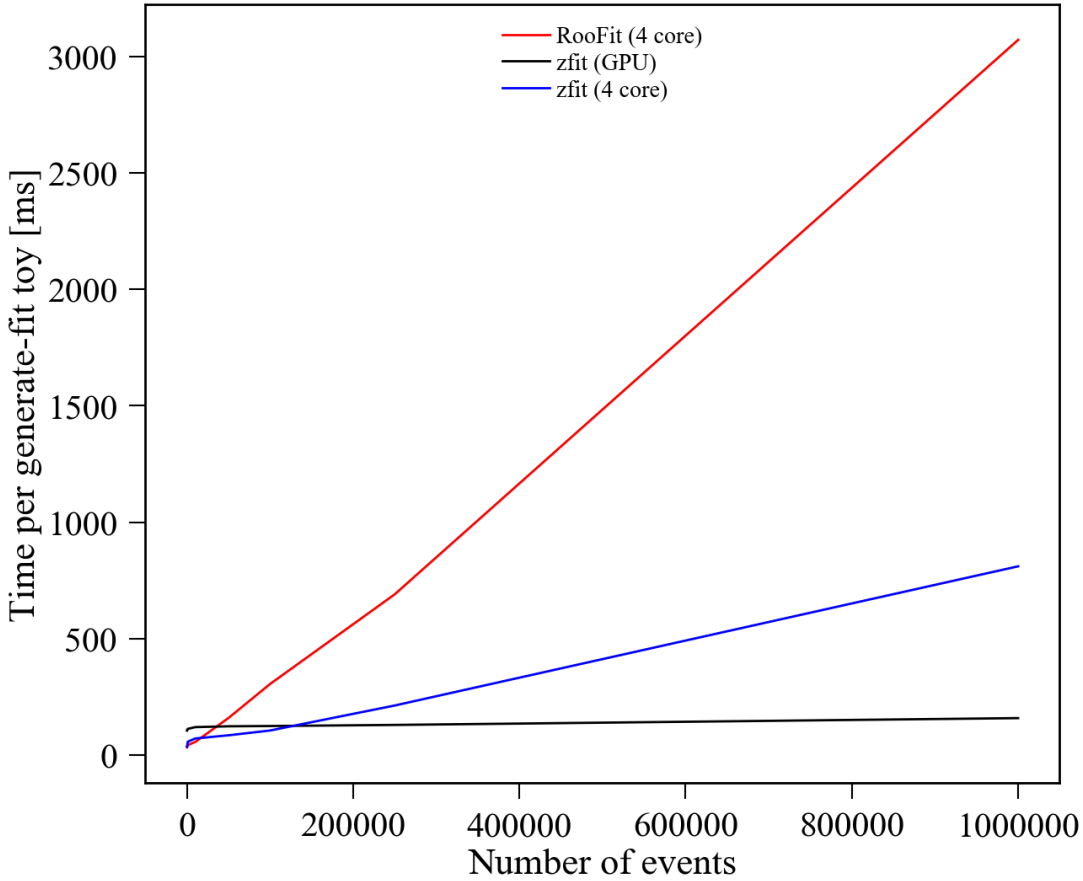
shared parameters

```
nll_simultaneous = zfit.loss.UnbinnedNLL(model=[gauss1, gauss2],
                                         data=[data1, data2])

nll1 = zfit.loss.UnbinnedNLL(model=gauss1, data=data1)
nll2 = zfit.loss.UnbinnedNLL(model=gauss2, data=data2)
nll_simultaneous2 = nll1 + nll2
```

Completely equivalent

Simple toy fit: speed

zfit
scalable pythonic fitting



Multiple sample and fit

Comparison
RooFit (4 cores)
zfit (4 cores)
zfit (GPU)

Summary

zfit
scalable pythonic fitting

- Beta stage, usable! (already used in LHCb analyses)
  - Not feature complete, but API stabilizing
- Contributions in form of *feedback and criticism* very welcome
  - API, use-cases, bugs,…
  - Any crazy idea!

It's about a reliable library

not about

«we need to get it working fast»

https://zfit.github.io/zfit/

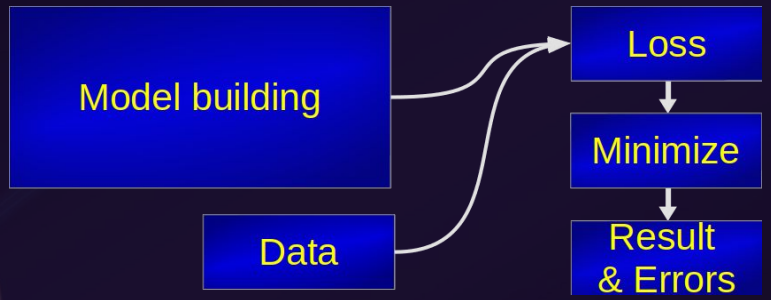
zfit@GitHub

Gitter channel

zfit@physik.uzh.ch

Join the discussion!

scalable pythonic fitting



zfit

It's about a reliable library

Backup Slides

Scalable: TensorFlow

zfit
scalable pythonic fitting

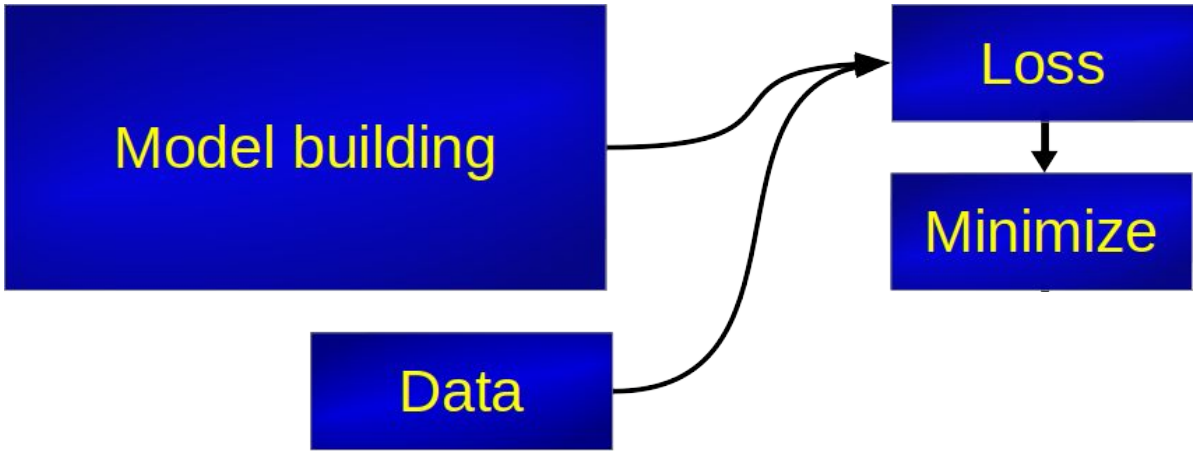
- Deep Learning framework by Google
- Modern, declarative graph approach
- Built for highly parallelized, fast communicating CPU, GPU, TPU,… clusters
- Built to use «Big Data»

TensorFlow

Scalable: TensorFlow

zfit
scalable pythonic fitting

- Machine learning in a nutshell:
  - Build a model (a lot of matrix multiplications with simple non-linear functions in between) with 100k+ free parameters
  - Create a loss function (see how good/bad the predictions are)
  - Minimize it



Pythonic: statistics tool «lauztat»

zfit
scalable pythonic fitting

- Author: Matthieu Marinangeli
- WIP, pre-beta
- Python statistics tool for limits, significance etc.
  (~ RooStats)
- lauztat on Github with example notebooks using `zfit`

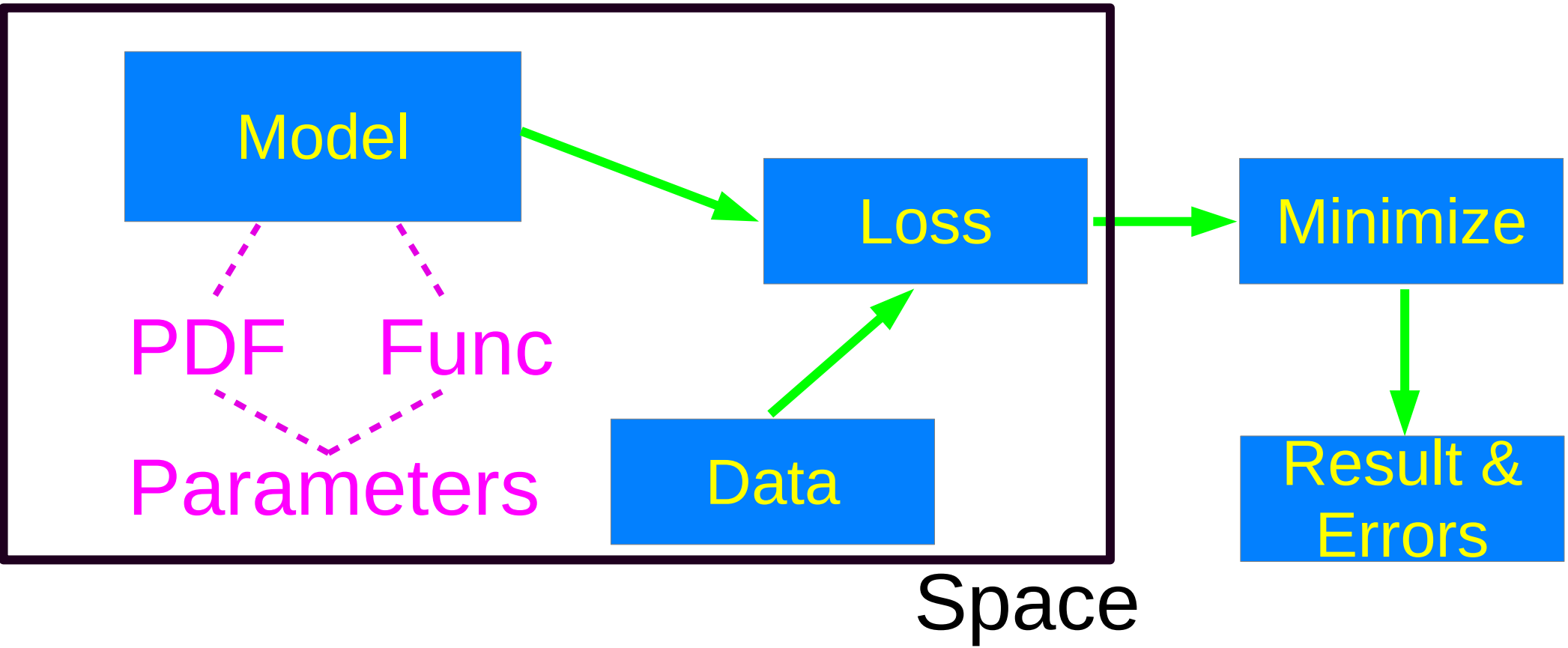
Pythonic: «phasespace»

zfit
scalable pythonic fitting

- Author: Albert Puig
- Python tool for n-body phasespace generation (~ TgenPhaseSpace)
- Uses TensorFlow as backend
- Can use zfit pdf for sampling

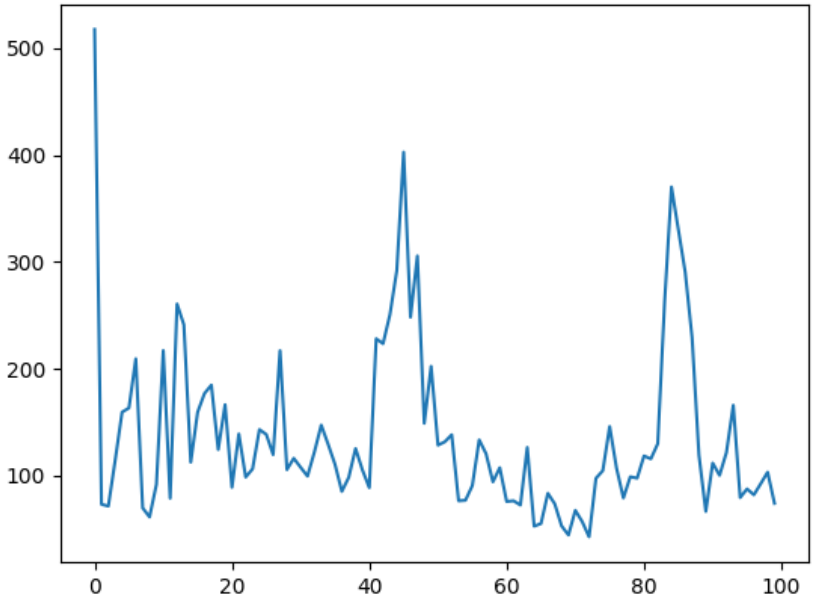
Fitting: complete structure

zfit
scalable pythonic fitting



Performance toy example

zfit
scalable pythonic fitting



Time per toy