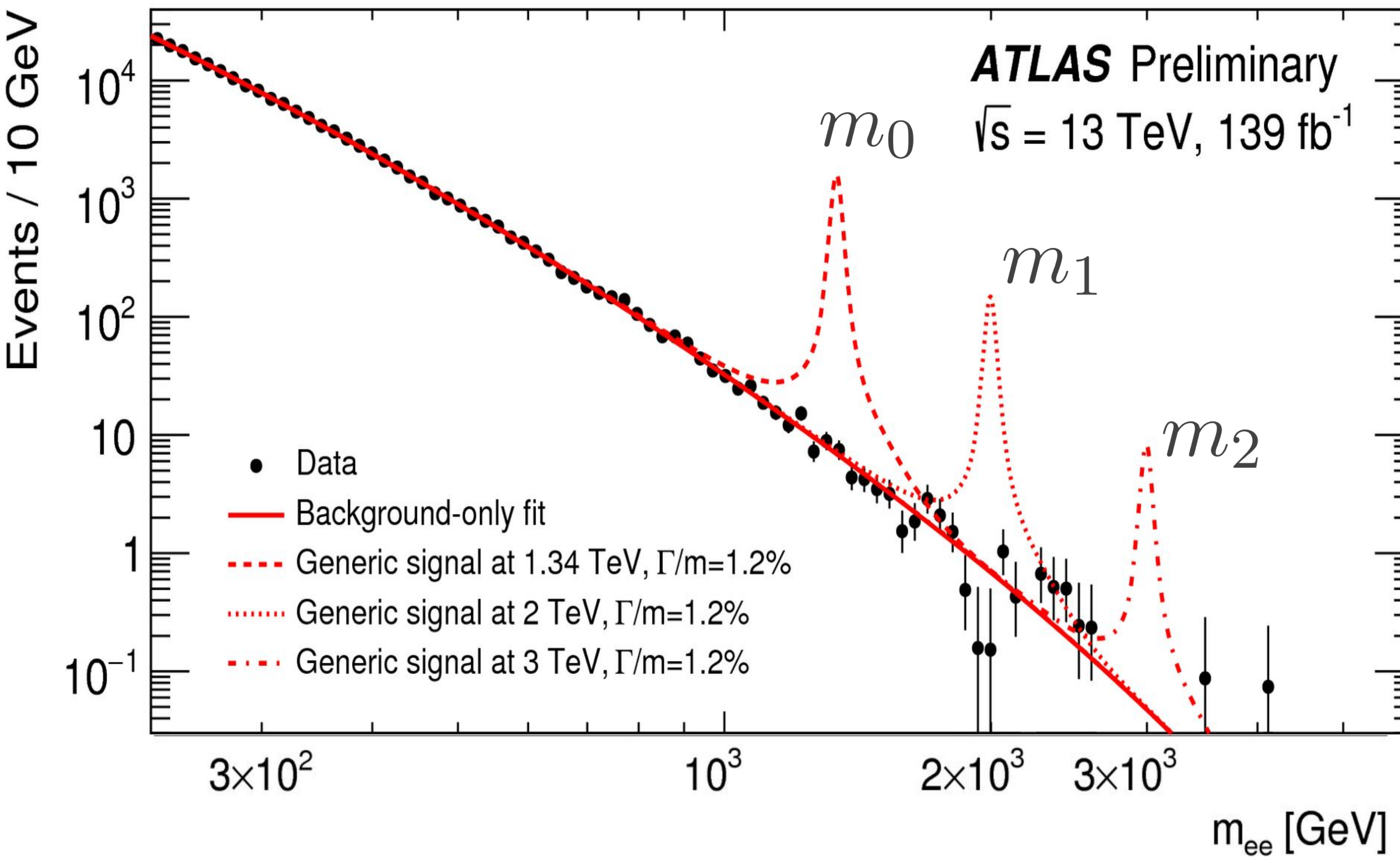
Simulation Based Inference(SBI) in RooFit

Robin Syring

Supervisor: Jonas Rembser

Student Sessions - 09/08/2024



ATLAS Preliminary
$\sqrt{s}$ = 13 TeV, 139 fb^{-1}
m_0
m_1
m_2
Data
Background-only fit
Generic signal at 1.34 TeV, Γ/m=1.2%
Generic signal at 2 TeV, Γ/m=1.2%
Generic signal at 3 TeV, Γ/m=1.2%
Events / 10 GeV
m_{ee} [GeV]
10^4
10^3
10^2
10
1
10^{-1}
3×10^2
10^3
2×10^3
3×10^3

Likelihood vs Probability

- hypothesis mass m_0
- measured data with mass m

$$\mathcal{L}(m_0) = p(m|m_0)$$

- Likelihood $\mathcal{L}$
- probability density function p

Likelihood Function in Practice

$$\mathcal{L}_{ratio}(m_0) = \prod_i \frac{p(m_i|m_0)}{p_{ref}(m_i)}$$

$$p(m_i|m_0) = \int dz_d \int dz_s \int dz_p p(m_i, z_d, z_s, z_p|m_0)$$

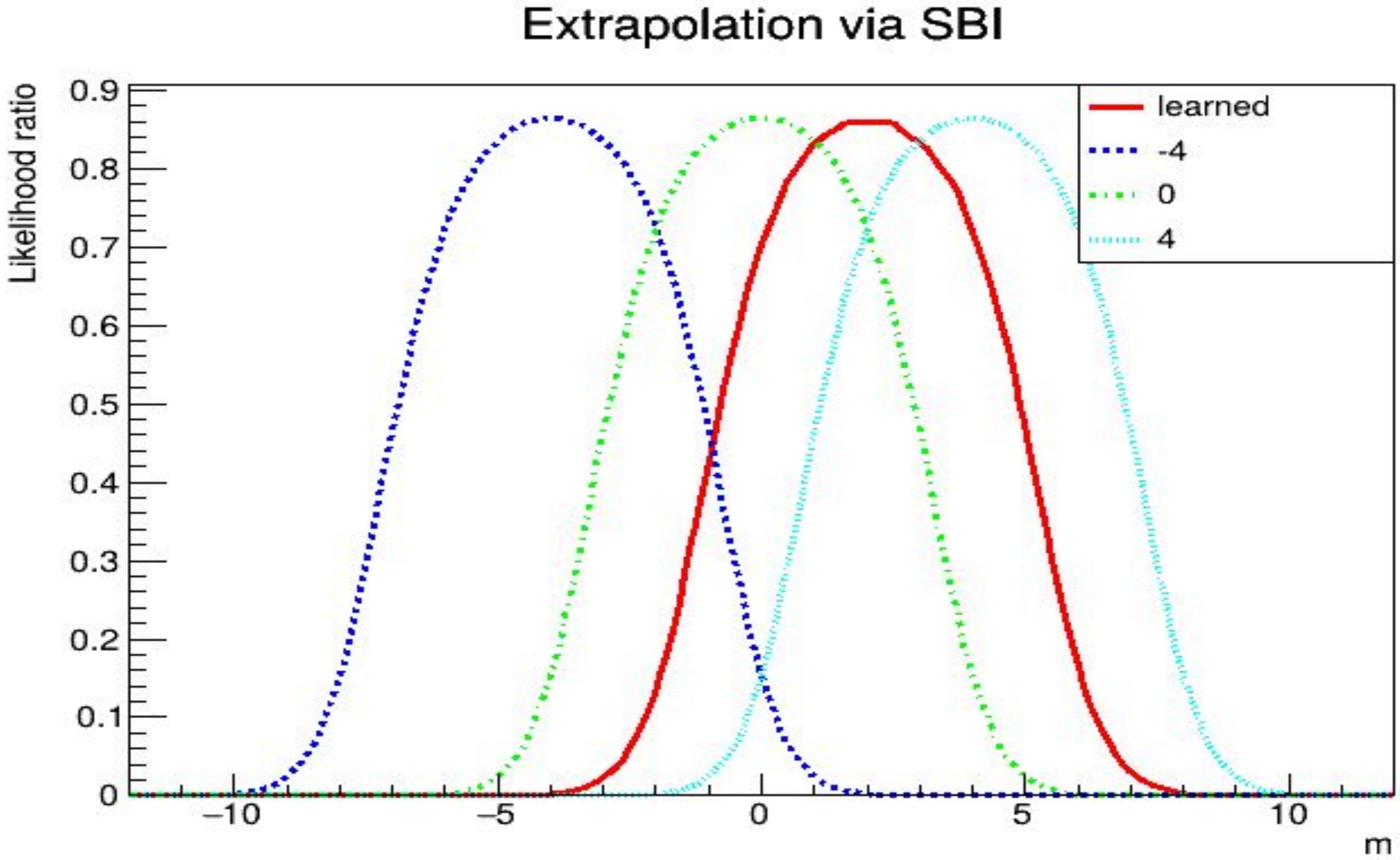
- Integrals cannot evaluated directly
  - Detector simulations take time

SBI = Use of ML to learn the likelihood function

Likelihood trick

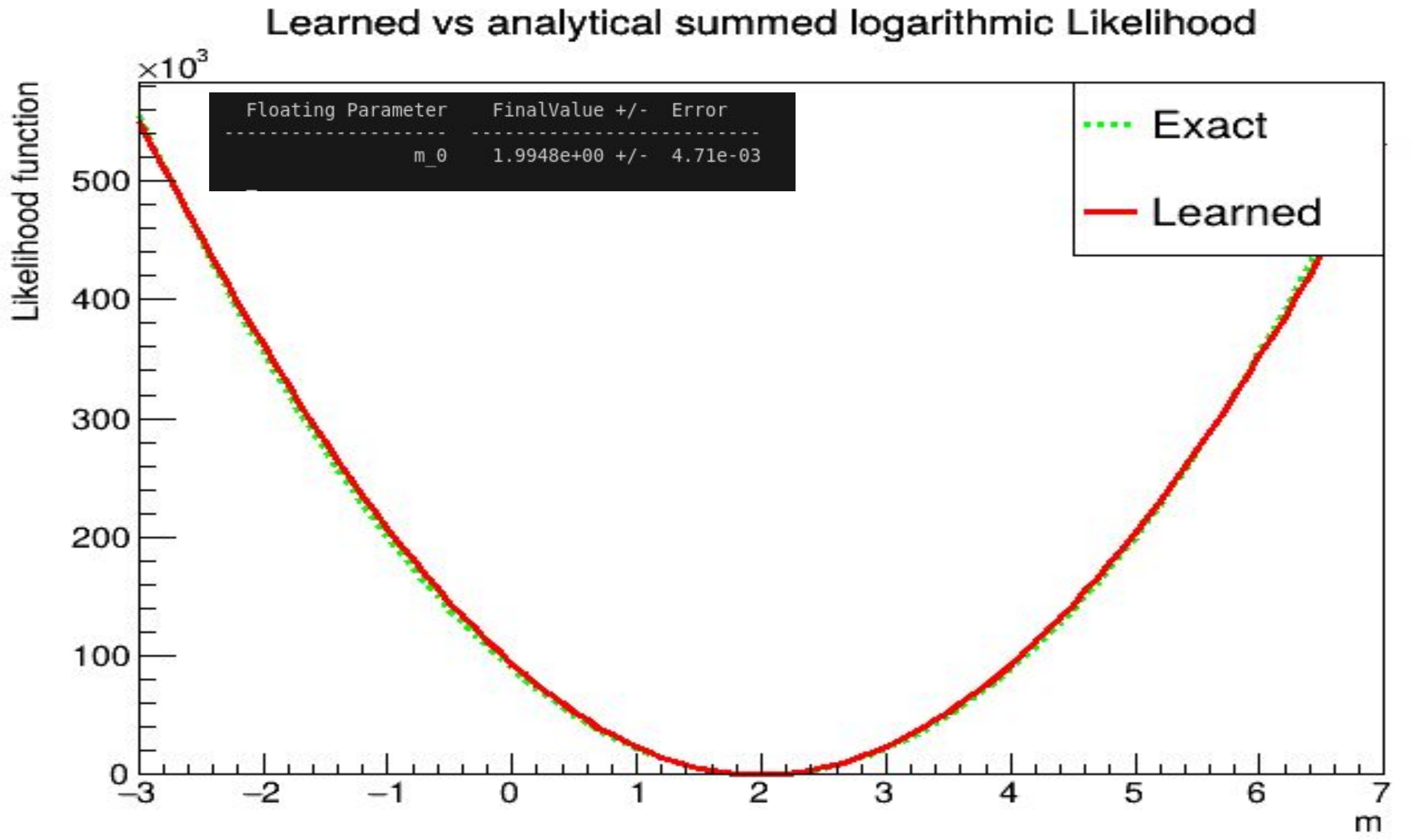
- train classifier to
  - discriminate between samples $m_i \sim p(m_i|m_0) \quad m_i \sim p_{ref}(m_i)$
  - transform output of classifier $\hat{s}(m_i|m_0)$ to estimator of likelihood ratio function by

$$\mathcal{L}_{ratio}(m_0) = \prod_i \frac{1-\hat{s}(m_i|m_0)}{\hat{s}(m_i|m_0)}$$


Extrapolation via SBI
Likelihood ratio
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−10
−5
0
5
10
m
learned
-4
0
4

Power of this method

- Model generalizes from ‘seen’ values to ‘unseen’
- reduction of necessary data samples i.e. detector simulations



Learned vs analytical summed logarithmic Likelihood
Floating Parameter FinalValue +/- Error
m_0 1.9948e+00 +/- 4.71e-03
Exact
Learned
Likelihood function
×10^3
m

Implementation

- **ROOT** for High Energy Physics:
  - Powerful data analysis framework for HEP
  - Developed at CERN for processing, analysis, and visualization
- **RooFit**:
  - ROOT's library for statistical analysis in HEP
  - Provide tools for parameter estimation as likelihoods

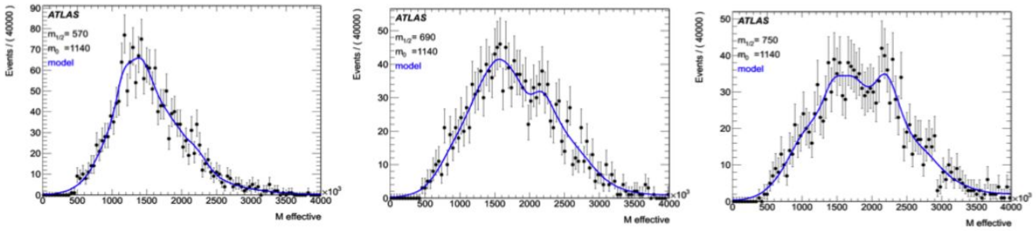
C++ Implementation

- wrapper to call arbitrary python functions in ROOT
- For showcase: MLP from Sklearn
- integrate declaration and definition of class in running codebase

Next Steps

Open tasks

- Provide test/use case for high dimensional problems
- comparison to currently used methods (template histograms)
    - evaluate proposed generalization behavior
- Final result: rf615_learned_likelihoods.py



Thank you!