New RooFit Developments to Speed up your Analysis

Zef Wolffs (Nikhef), Carsten Burgard (DESY), Jonas Rembser (CERN),
Lorenzo Moneta (CERN), Wouter Verkerke (Nikhef),
Patrick Bos (Netherlands eScience Center)

ICHEP 2022, 08-07-2022

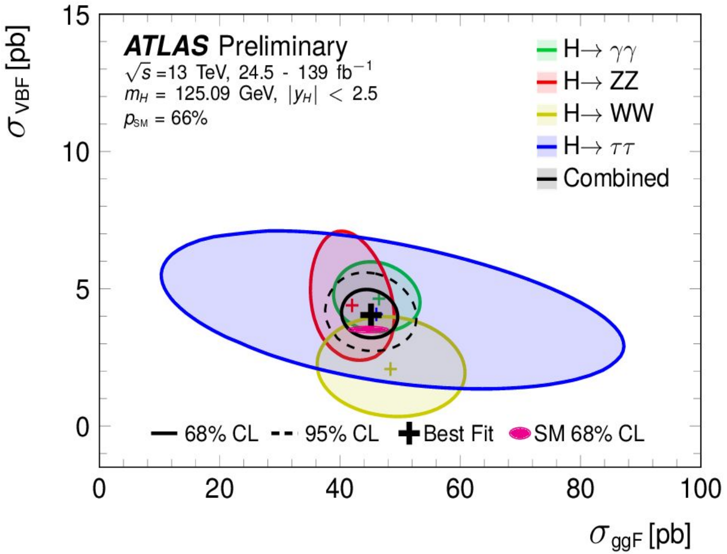
Introduction

- Physics analyses continuously generate increasingly complex likelihood models to describe their data
  - O(1000) parameters
  - O(100) likelihood components
  - O(100) datasets

- Just to name a few
  - **Higgs combination fits**
  - EFT interpretations

- It is certain models will increase in complexity in the foreseeable future with the first LHC run 3 data coming in soon

- RooFit needs to accommodate for these fits
  - From hours fit time down to minutes, i.e. work-day to coffee break



Recent ATLAS Higgs combination fit result [1]. This fit took about five hours to complete.

[1] ATLAS Collaboration. (2020). A combination of measurements of Higgs boson production and decay using up to 139 fb^{-1} of proton–proton collision data at $\sqrt{s}$ = 13 TeV collected … (see last slide)

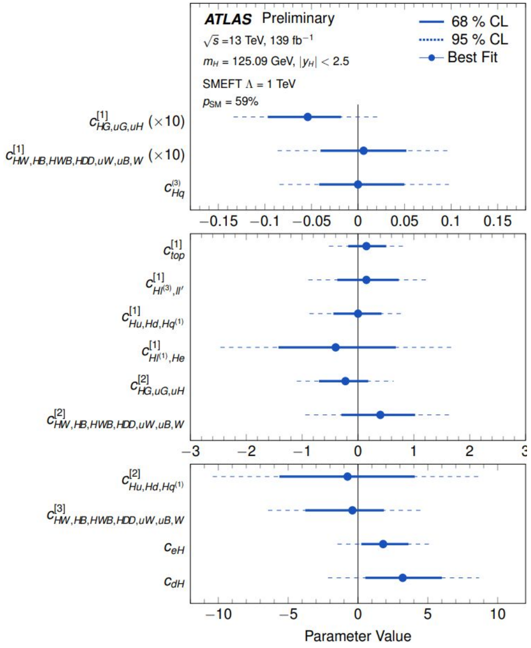
Introduction

- Physics analyses continuously generate increasingly complex likelihood models to describe their data
  - O(1000) parameters
  - O(100) likelihood components
  - O(100) datasets

- Just to name a few
  - Higgs combination fits
  - **EFT interpretations**

- It is certain models will increase in complexity in the foreseeable future with the first LHC run 3 data coming in soon

- RooFit needs to accommodate for these fits
  - From hours fit time down to minutes, i.e. work-day to coffee break

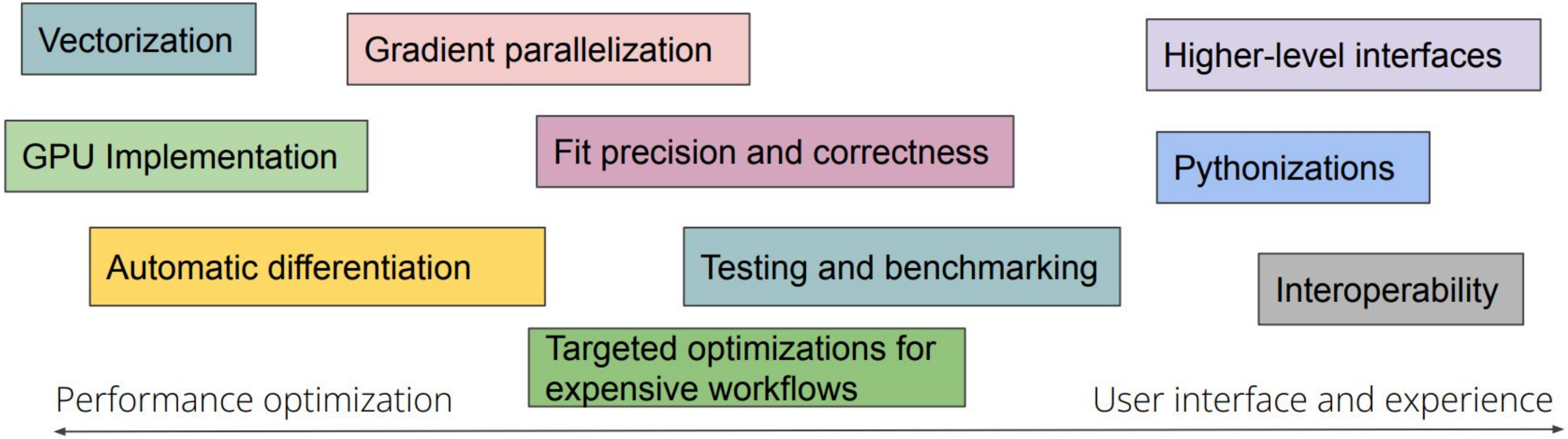


Recent ATLAS SMEFT fit result [2], this fit took about 10 hours to complete without nuisance parameter pruning.

[2] ATLAS Collaboration. (2021). Combined measurements of Higgs boson production and decay using up to 139 fb^{-1} of proton-proton collision data at $\sqrt{s}$= 13 TeV collected with the … (see last slide)

Introduction

In which areas does RooFit evolve?



Introduction

In which areas does RooFit evolve?

Vectorization

Gradient parallelization

Higher-level interfaces

GPU Implementation

Fit precision and correctness

Pythonizations

Automatic differentiation

Testing and benchmarking

Interoperability

Targeted optimizations for expensive workflows

Performance optimization ⟷ User interface and experience

These are the focus of my talk!

Gradient Parallelization

CPU parallelization - serial MIGRAD

$$\frac{f(x+h)-f(x)}{h}$$

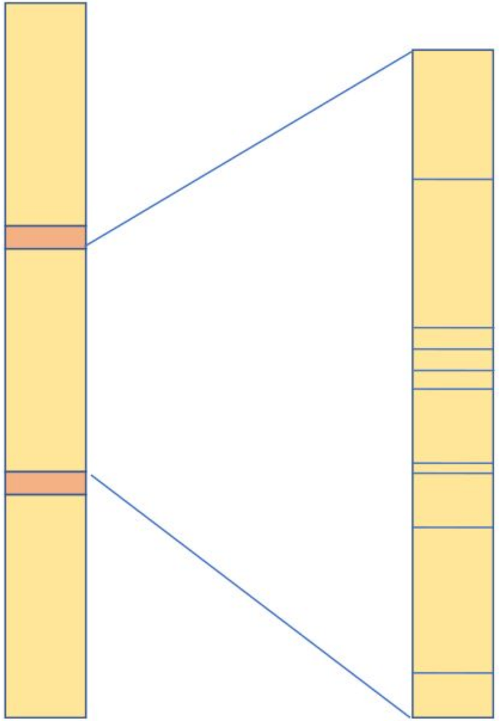
Gradient
N partial derivatives,
2N likelihood evals

line search O(3) evals

Gradient
N partial derivatives,
2N likelihood evals

line search O(3) evals

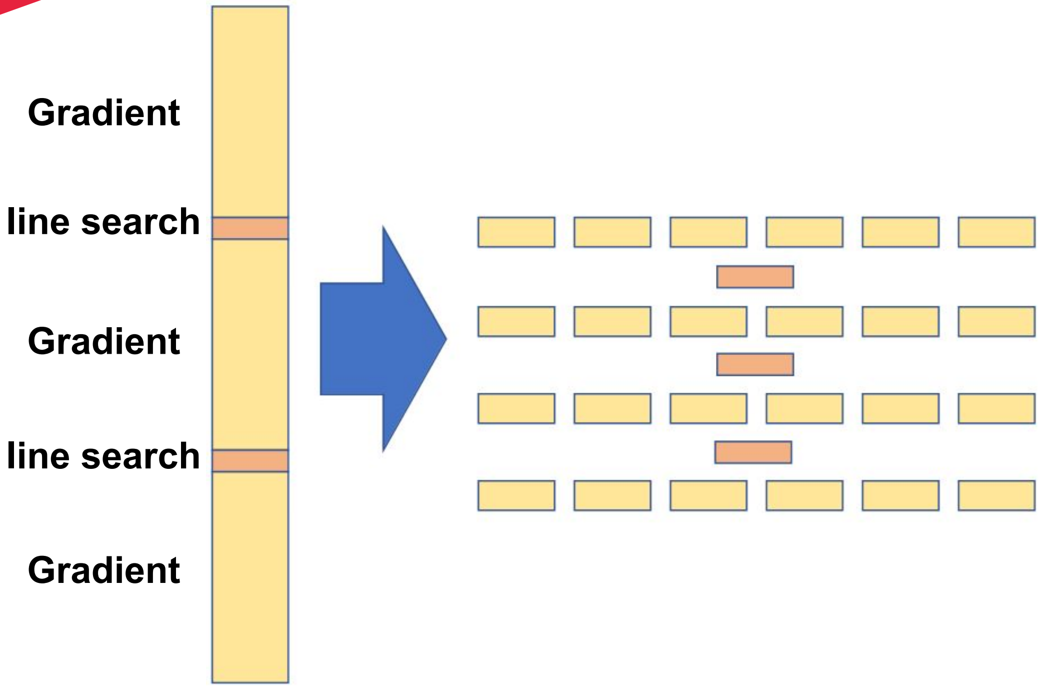
Gradient
N partial derivatives,
2N likelihood evals



CPU time for partial derivative varies strongly with parameter:

- Some parameters only in subset of likelihood components, so most components do not always need re-evaluation
- Complexity of component likelihood calculation varies

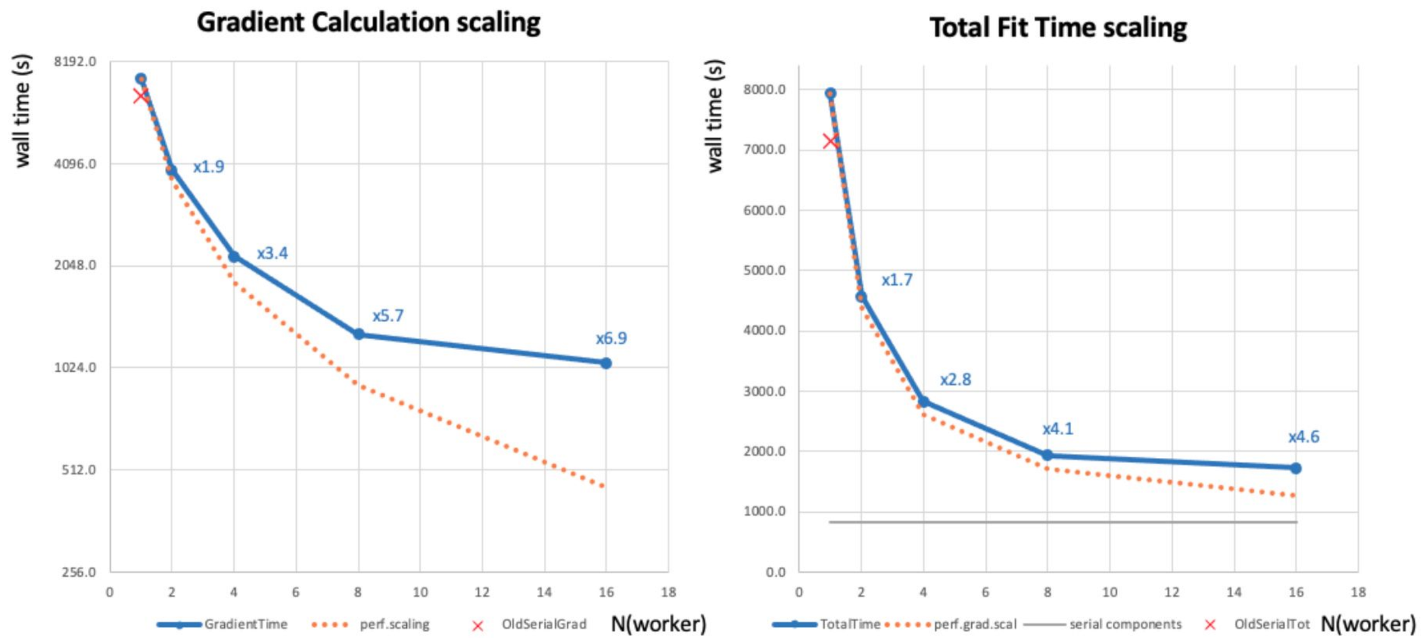
CPU parallelization - parallel MIGRAD



- Parallelize at gradient calculation level
- Dynamic load balancing over workers through random work stealing algorithm
- Designed to have maximum speed impact of complex fits with many parameters
- Line search has limited impact on scaling, but this was investigated further

[3] Bos, E. G. P., Burgard, C. D., Croft, V. A., Hageboeck, S., Moneta, L., Pelupessy, I., ... & Verkerke, W. (2020). Faster RooFitting: Automated parallel calculation of collaborative … (see last slide)

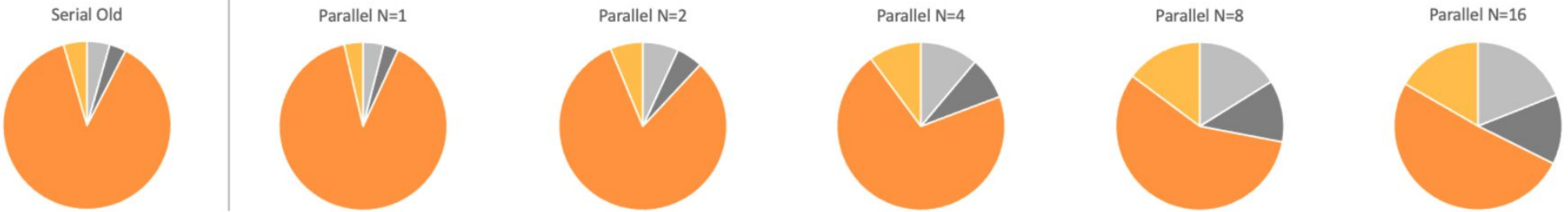
CPU parallelization - Higgs combination



- Walltime decrease in Higgs combination fit from from **2h12m26s → 28m52s**
- Serial time expenditure close to parallel time expenditure with single worker
- Fit validated to conform to serial run, all parameters agree within 1% of estimated uncertainty

CPU parallelization - Higgs combination

	serial old	parallel N=1	parallel N=2	parallel N=4	parallel N=8	parallel N=16
roofit_setup	313	314	315	315	312	327
migrad_seed	230	231	231	231	231	231
migrad_gradient	6289	7102	3734	1997	1107	879
migrad_descent	323	287	287	287	287	287



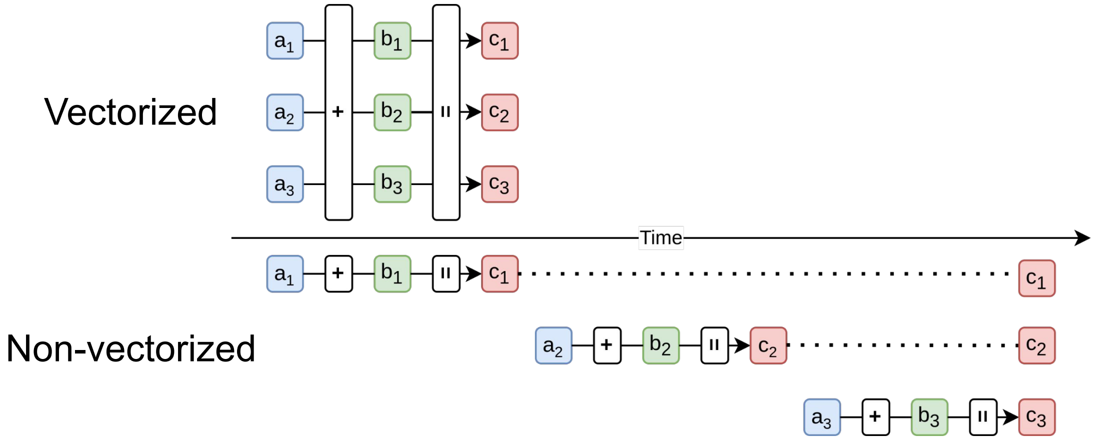
- Relative time expenditures show that serial components start playing increasingly important role in total walltime when using more workers

- For workspaces with many component likelihoods, parallelizing the linesearch could also prove beneficial

Batched Computation

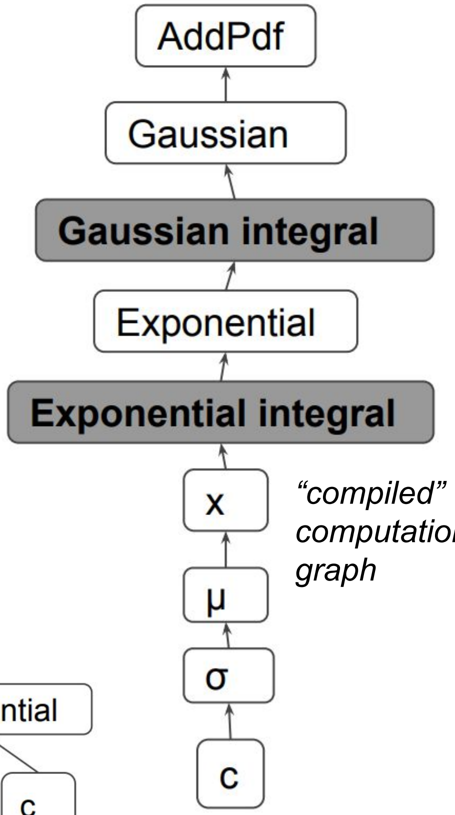
Batched Computation - Motivation

- Batched computation in RooFit refers to the principle of computing batches of events simultaneously, this has the following benefits:
  - GPU parallelization: Simple operations applied on each of the O(1000)+ CUDA cores on the GPU in parallel
  - Vectorization: Most modern processors are equipped with instruction sets which apply the same operation simultaneously to multiple pieces of data, or “batches”

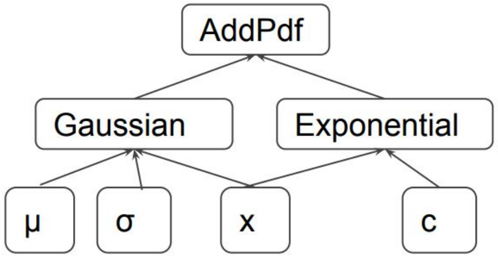


Batched Computation - Compute graph

- Major restructure of RooFit computational back-end to allow for batched computation
  - Previous implementation evaluated the computational graph of likelihood components on event-by-event basis
  - Newly developed implementation restructures computation graph as sequence of functions to allow for vectorized evaluation of all (relevant) events per computational graph node
  - Regardless of vectorization, this new strategy reduces computation times due to improved CPU caching



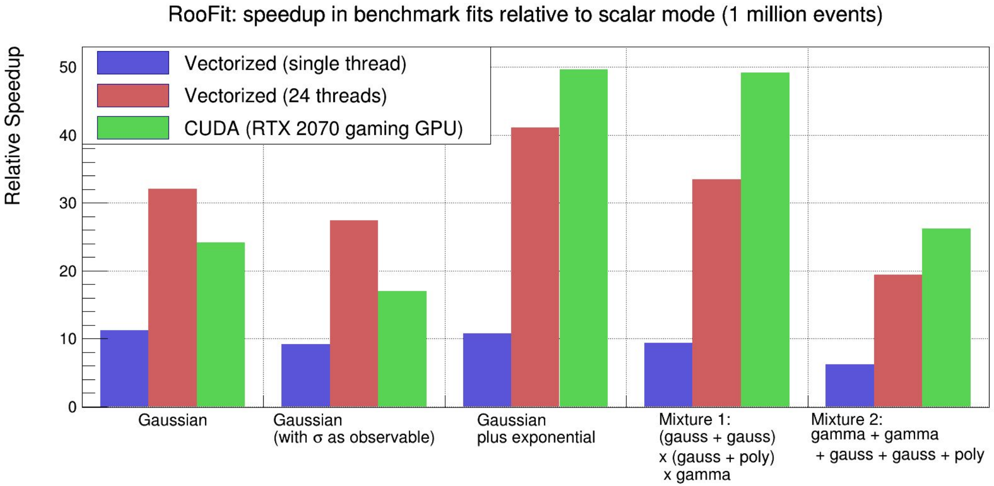
“compiled” computation graph



Original computation graph

Batched Computation - Results

- For unbinned fits with large numbers of events we see a huge speedup using batched computation
  - Vectorization alone effective, but with also multithreading and GPU parallelization (CUDA) we record even larger speedups
- Speedup grows with the size of the dataset that is being evaluated
  - Especially true for GPU due to relatively large overhead of data transfer to GPU cores

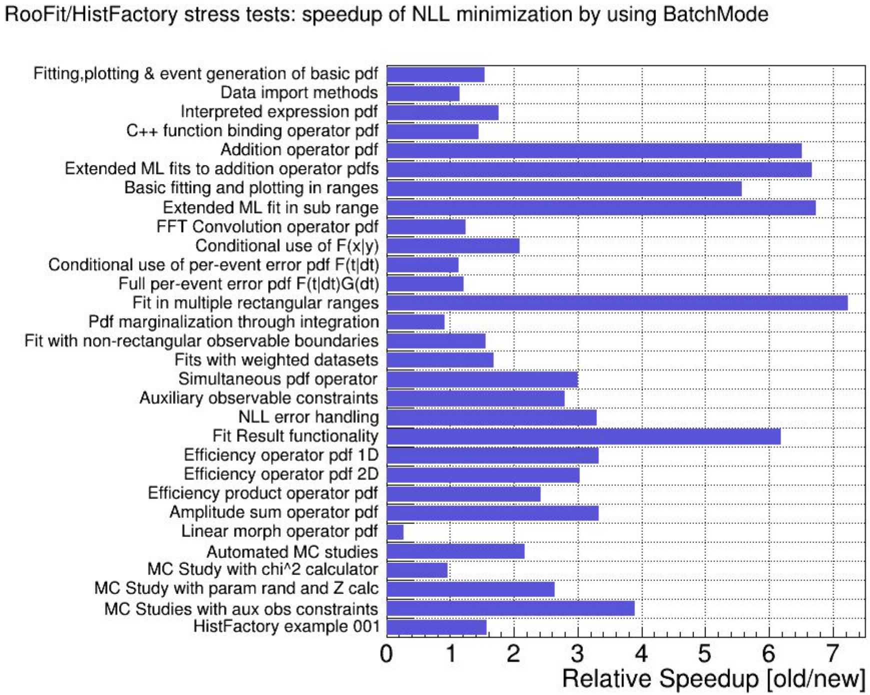


Batched Computation - Results

- Plot on the right shows relative time spent in the minimization of the likelihood with Batch mode on vs. off

- Speedup for virtually all tests
  - Even for tests which do not involve large numbers of events, restructuring of the computational graph results in faster minimization

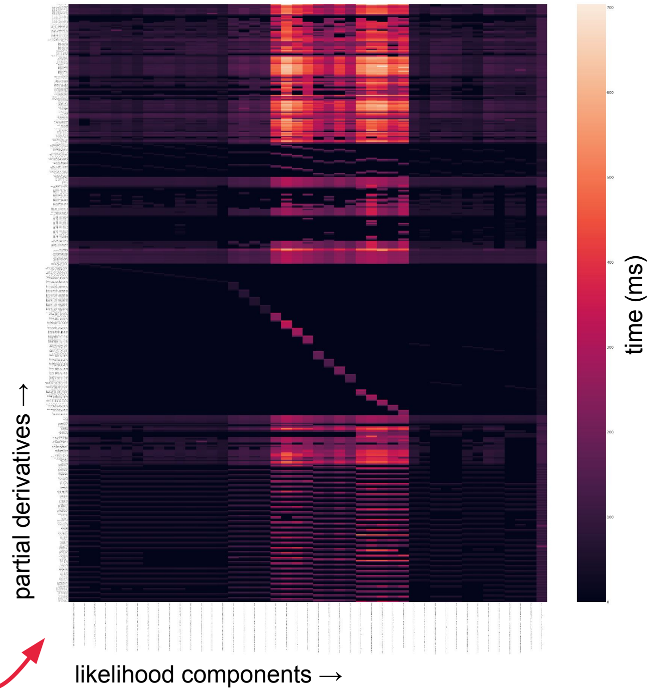
- Speedup appears very dependent on the problem



Conclusion

Conclusion

- Optimization strategies currently under development
  - **Batched computations** (available now, ROOT 6.26)
  - **Gradient parallelization** (available soon, ROOT 6.28)
  - **Combination of the above**, allowing for combined speedup (available soon, ROOT 6.28)
  - **Automatic differentiation** (early stage)
    - Prototyping HistFactory implementation with automatic differentiation using Clad
- Also more tools under development to allow analysers to scrutinize workspace and optimize analysis computationally
- Reduced complex Higgs combination fit time by factor five: goal of reducing day-long fits to coffee break within reach



References

[1] ATLAS Collaboration. (2020). A combination of measurements of Higgs boson production and decay using up to 139 fb^{-1} of proton–proton collision data at $\sqrt{s}$ = 13 TeV collected with the ATLAS experiment. *ATLAS-CONF-2020-027.*

[2] ATLAS Collaboration. (2021). Combined measurements of Higgs boson production and decay using up to 139 fb^{-1} of proton-proton collision data at $\sqrt{s}$= 13 TeV collected with the ATLAS experiment. *ATLAS-CONF-2021-053.*

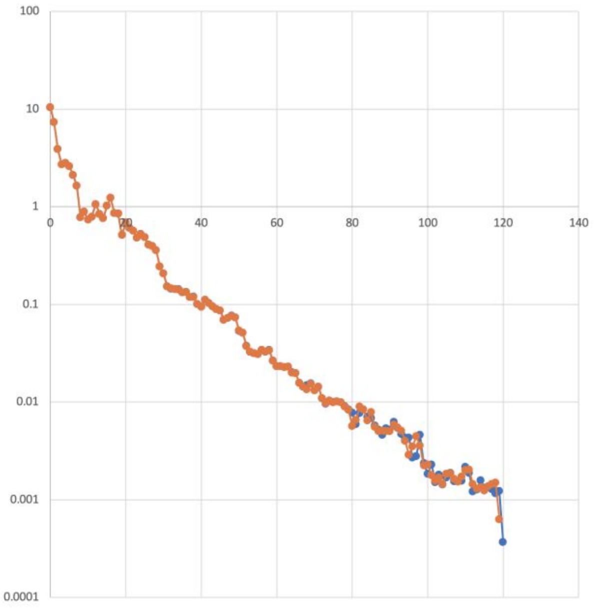
[3] Bos, E. G. P., Burgard, C. D., Croft, V. A., Hageboeck, S., Moneta, L., Pelupessy, I., ... & Verkerke, W. (2020). Faster RooFitting: Automated parallel calculation of collaborative statistical models. *EPJ Web of Conferences* (Vol. 245, p. 06027). EDP Sciences.

Backup

Parallel CPU fit validation

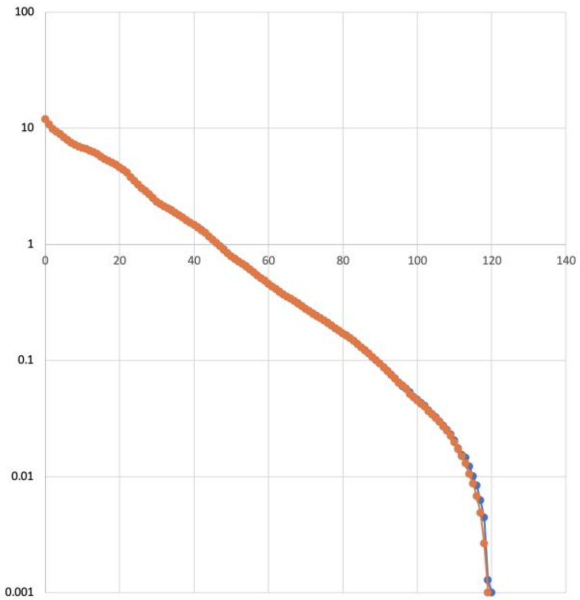
Hcomb workspace (120 VM steps, Npar=3105, Ncomp=334)

EDM vs VariableMetric step



-log(L) vs VariableMetric step

(L offset such that minimum is by definition at 0.001)



Convergence

99% of the 3105 parameters agree within 0.1% of estimated uncertainty
All parameters agree within 1% of the estimate uncertainty