RooFit in LHCb

A brief overview of RooFit in LHCb physics analysis

Jamie Gooding (*TU Dortmund*)

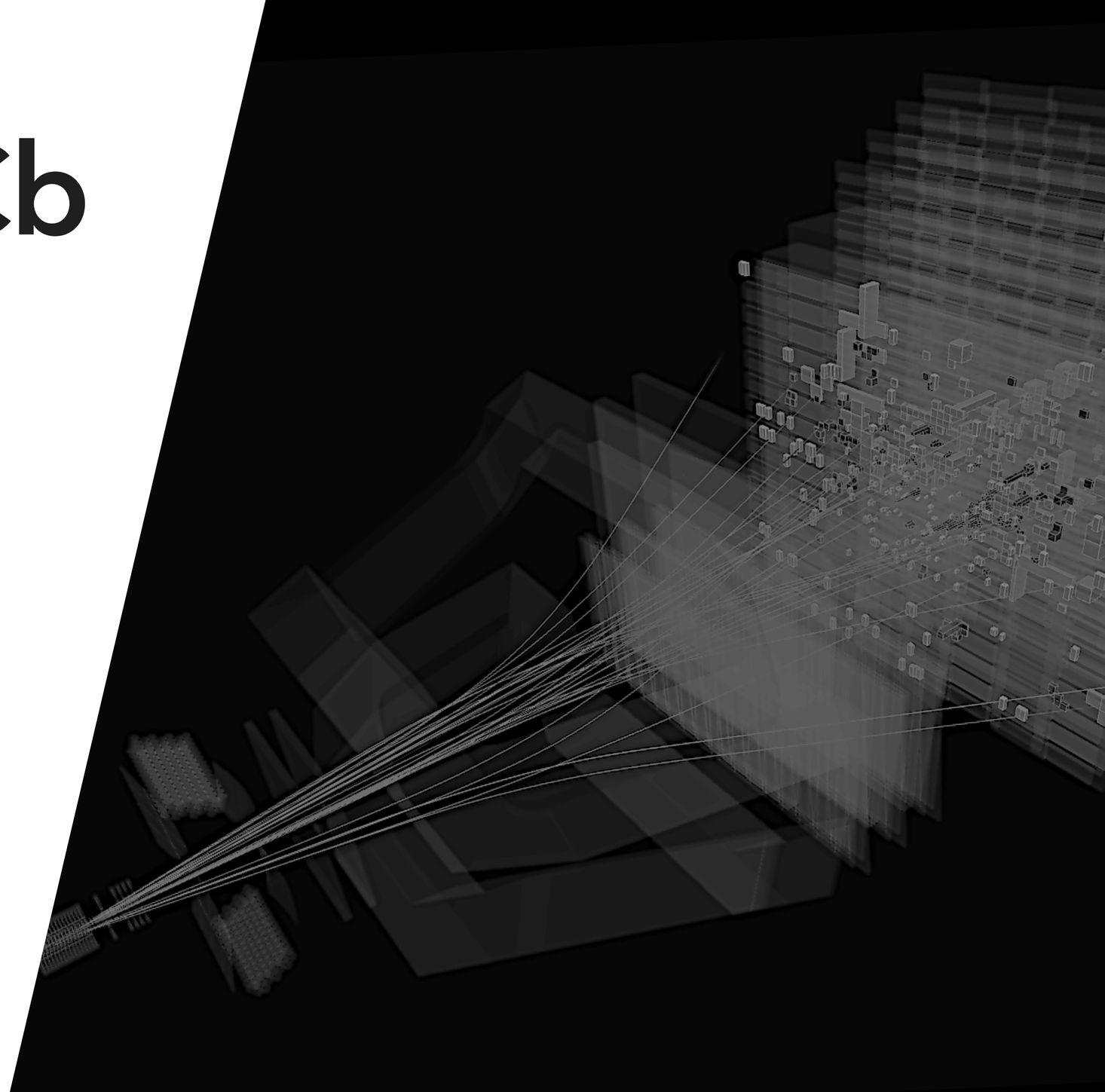
On behalf of the LHCb Experiment

Roofit Workshop 2024
12th April 2024

technische universität dortmund

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

MARIE CURIE ACTIONS

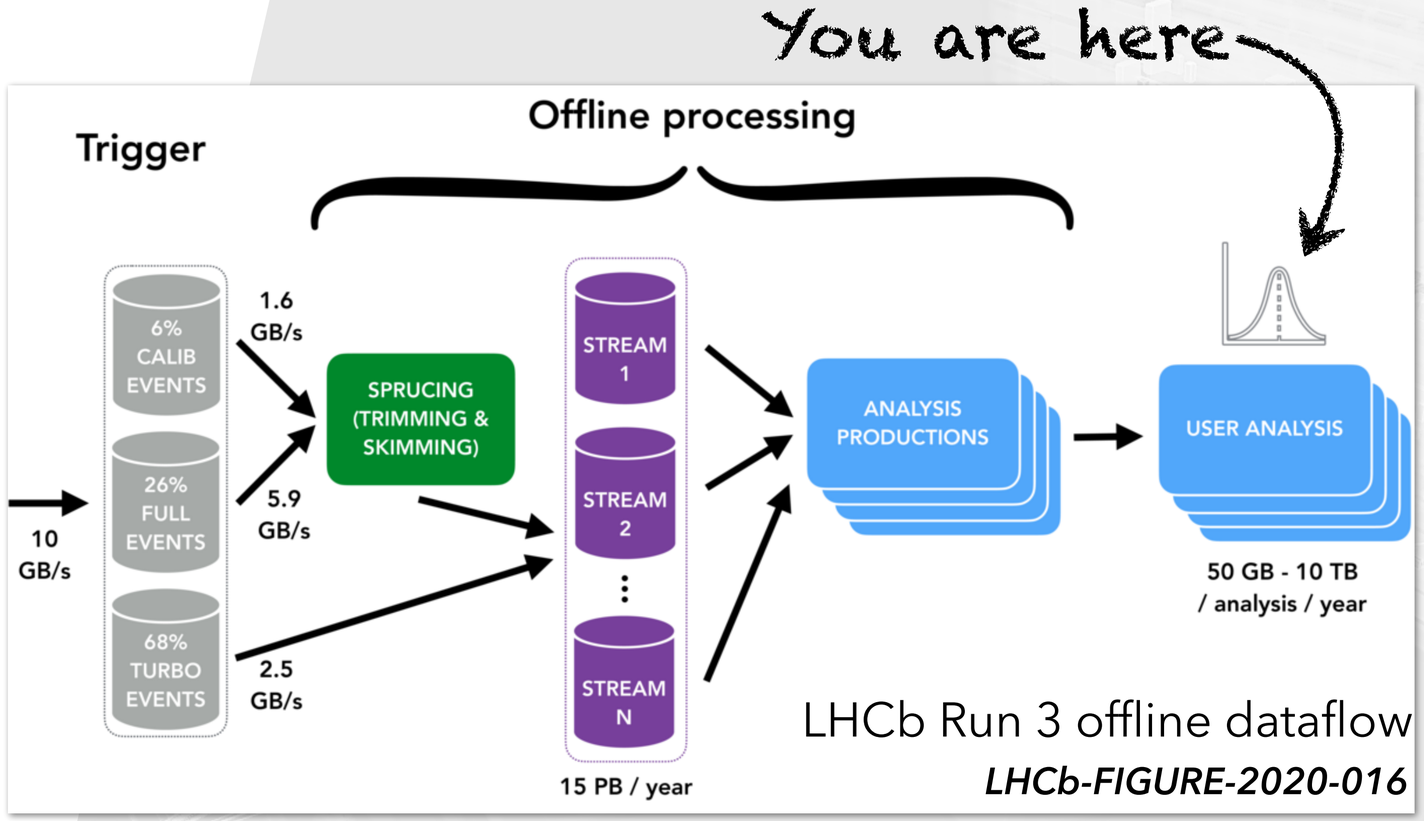


Introduction

- RooFit has long been popular amongst LHCb users
  - RooFit data structures match well to LHCb use case (e.g., `RooDataSet` matches “1 entry per event” structure)
  - RooFit supports many probability density functions used by LHCb (e.g., `RooCrystalBall`, `RooHypatia`, `RooJohnson`)
  - Other popular tools used include `zfit` and `iminuit`
- About me
  - LHCb analyst (and RooFit user) of ~3 years
  - Using RooFit in multiple working groups

LHCb physics analysis

- LHCb analyses typically*
  - $\lesssim \mathcal{O}(10)$ participants
  - $\mathcal{O}(1\ \text{TB})$ datasets
  - Python script-based
  - Use distributed resources
- Often centred around a main fit
  - Fit stability/bias/sensitivity commonly investigated
  - Fitting also used in additional studies (e.g., systematics)



LHCb Run 3 offline dataflow

LHCb-FIGURE-2020-016

*This is a broad generalisation!

Fit complexity at LHCb

Time-integrated measurements

$\mathcal{O}(1)$ parameters

"Simple" probability density functions

$\lesssim$ 5-dimensional

$\mathcal{O}(10)$ parameters

Adds decay time dimensions (convolution w/ acceptance+resolution)

Time-dependent fits

$\mathcal{O}(100)$ parameters

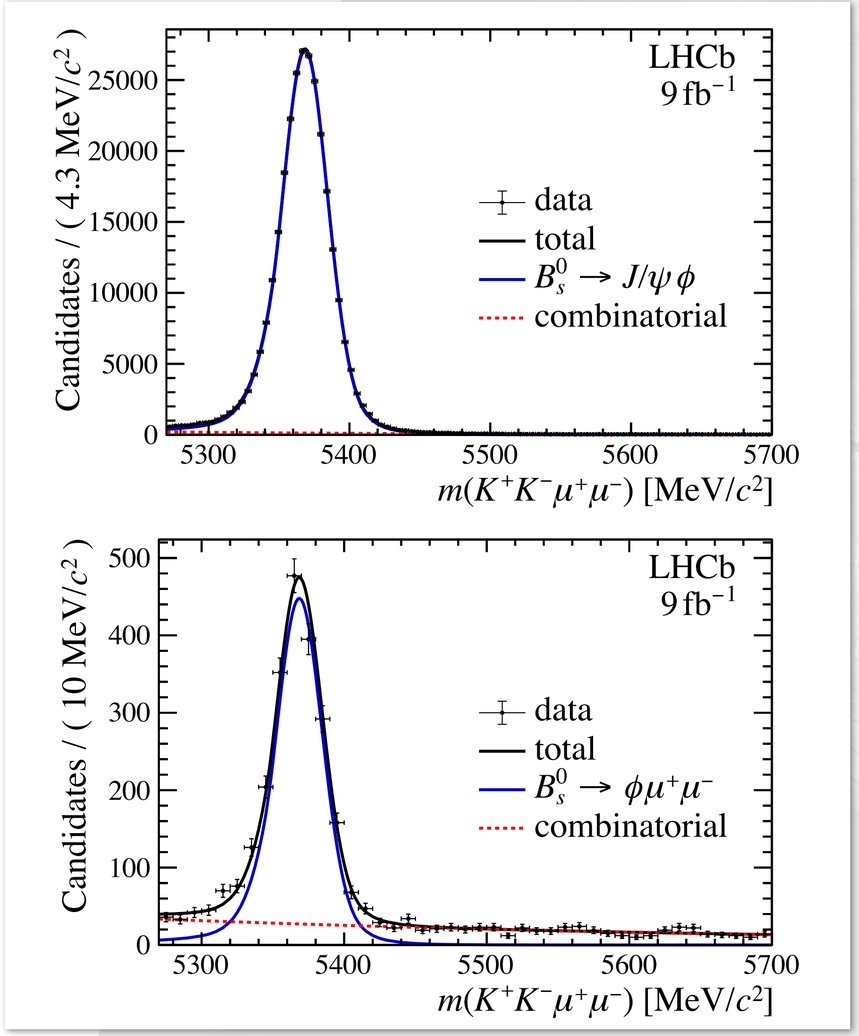
Models formed of many components

$\gtrsim$ 2-dimensional

Amplitude analyses

Time-integrated fits

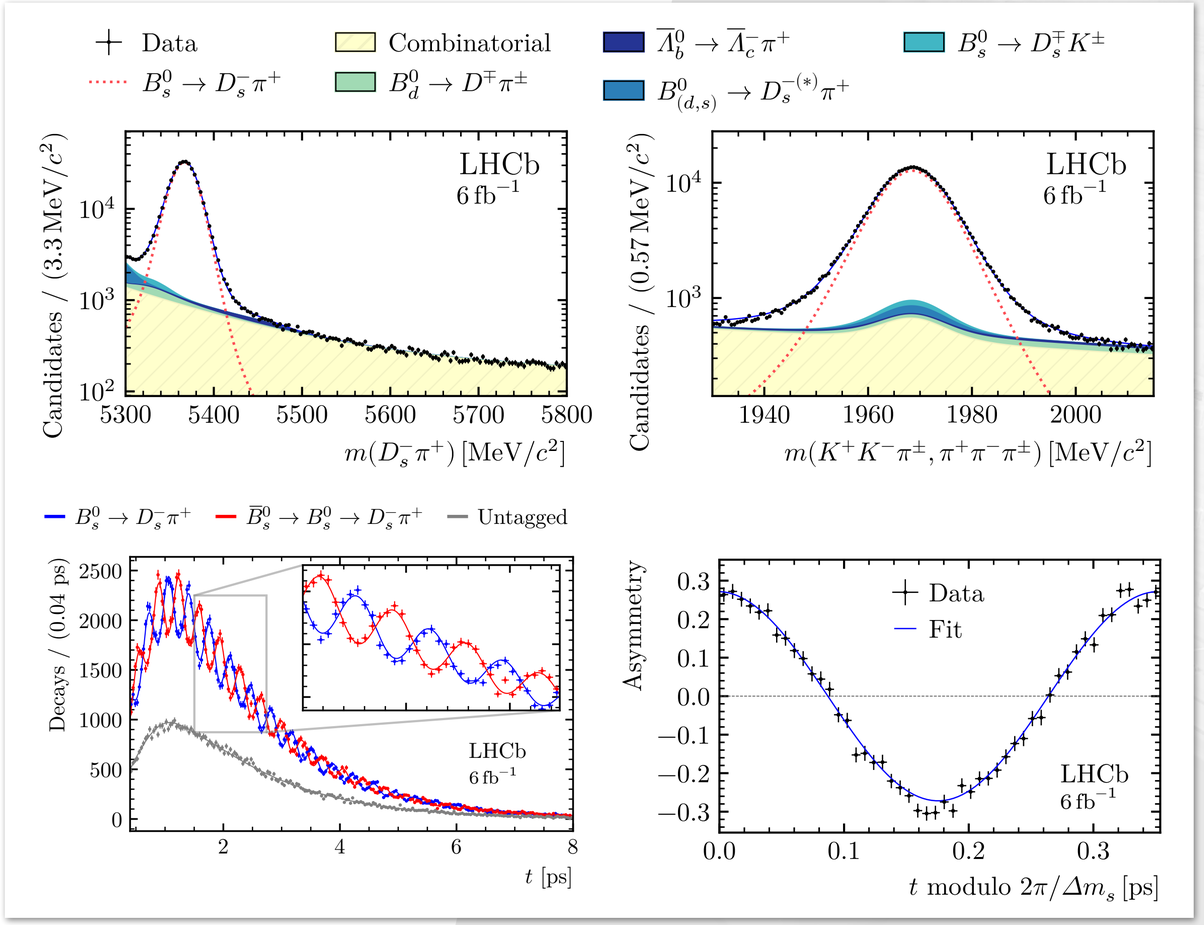
- Time-integrated fits → broad category, typically involve fits to kinematic distributions
- Fits can be (and often are) multi-dimensional, e.g., in invariant masses of different particle combinations
- Often fits are simultaneous, common cases include
  - Control of parameters through control/normalisation channels
  - Charge conjugate final states
  - Bins in other variables (e.g., kinematic bins, MVA output)
- Fits also used elsewhere in analyses, e.g., to provide weights via the *sPlot* method



Phys. Rev. Lett. 127 (2021) 151801

Time-dependent fits

- Time-dependent fits necessity for measurements involving mixing
  - Meson oscillations accounted for
- Fit model *must* account for decay time acceptance and resolution
  - Can be implemented as convolution with decay time
  - Add to existing dimensions required in time-integrated fit
- Complexity requires user-defined classes/ functionality e.g., B2DXFitters in Urania



Nature Phys. 18 (2022) 1-5

Amplitude analysis fits

- RooFit not well-suited to complexity and required functionality
- Other frameworks generally used:
  - On CPU:
    - Laura++/Mint/AmpGen (C++-based)
  - On GPU:
    - GooFit (C++-based)
    - Impanema-β (Python-based)
  - On both:
    - zfit (Python-based)
- Associated fits (e.g., mass fits) do frequently use RooFit

Largely based on talk by Albert Puig Navarro

Fit validation

- Fit validation can take a number of forms:
  - Measurement of fit bias in parameters
  - Verification of fit stability
  - Systematic effects of model template parameters
- Often evaluated through toy studies
  1. Fit is performed
  2. “Toy” Monte Carlo (MC) sample generated from fit
  3. Fit performed on toy sample
  4. Repeat $N(many)$ times
  5. Inspect variation across toy fits

Future requirements

- Speed already a crucial factor for users
  - LHCb analyses will only get larger (LHCb plans to take $50\ \text{fb}^{-1}$ in Run 3+4, $8.7\ \text{fb}^{-1}$ for Run 1+2 for context)
  - Functionality from ROOT 6.30 provides big improvement
    - GPU backend opens up many possibilities for fits with many floating parameters
    - Alternatives, e.g., iminuit+numba still provide similar speed [ScikitHEP iminuit benchmark]
- Usability a key focus → lots of recent progress
  - Stability of pythonisations significant factor in LHCb

Conclusion

- RooFit remains popular in LHCb (though challengers also present)
- Points for discussion:
  - How can RooFit keep pace with the scale of data in Run 3 and beyond?
  - What can LHCb provide to the developer community to encourage further collaboration?
  - How do we promote new user uptake, particular among Python-focused users?

Thanks for your attention

Backup

The LHCb Detector