ICHEP 2022 BOLOGNA

Istituto Nazionale di Fisica Nucleare
SEZIONE DI FIRENZE

Lamarr

the LHCb Ultra-Fast Simulation Framework

Lucio Anderlini,
on behalf of the LHCb Simulation Project

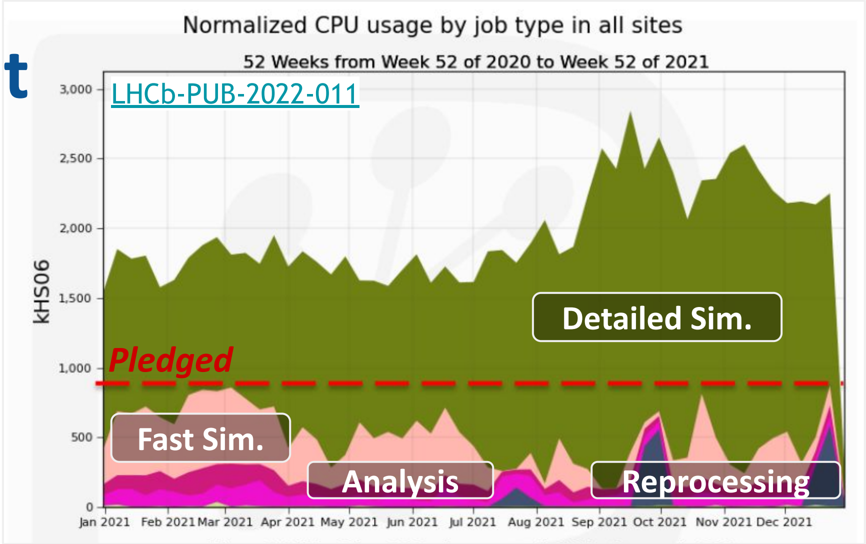
International Conference of High Energy Physics 2022
July, 8th   Bologna, Italy

Simulating the LHCb experiment

Detailed Simulation

Centralized MC productions. Interactions of particles with detector material is simulated with Geant4, and converted into *hits*.

Same trigger & reconstruction algorithms are used as for real data.



LHCb-PUB-2022-011

Fast Simulation

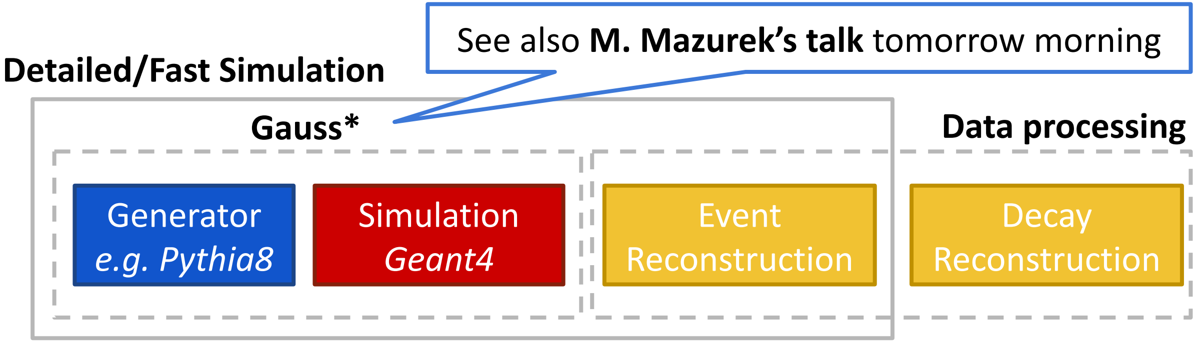
Replace parts of the simulation with models, e.*g.*

underlying event → ReDecay

[Eur. Phys. J. C 78 (2018) 1009]

calorimeter deposits → CaloGAN

[arXiv:1812.01319]



See also **M. Mazurek's talk** tomorrow morning

Detailed/Fast Simulation

Gauss*

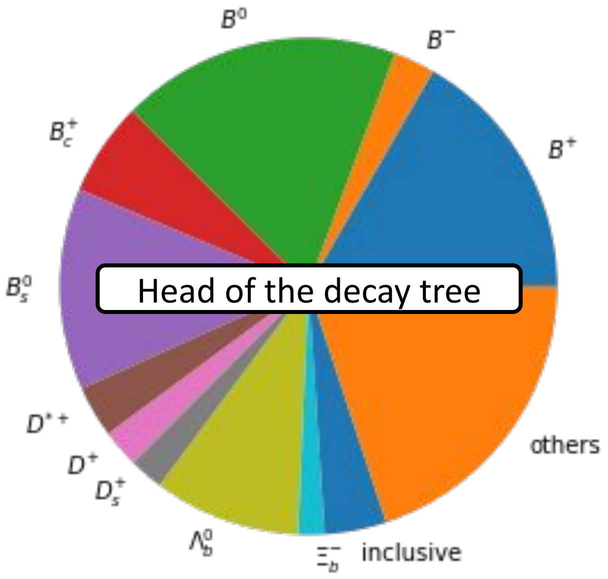
Data processing

* *Gauss is the LHCb simulation framework based on Gaudi* [J. Phys. Conf. Ser. 331 032023]

How does LHCb simulate events?

What should we parametrize first?

10^4 MC datasets generated with 2016 nominal conditions



Head of the decay tree

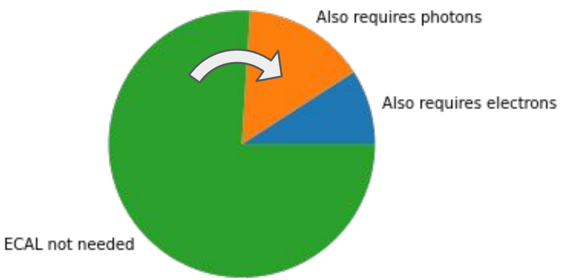
Most simulated decay modes are heavy hadron decays.

The detector will provide very similar *“response”* to, *e.g.*, a *kaon* from either a B^+ or a B_c^+.

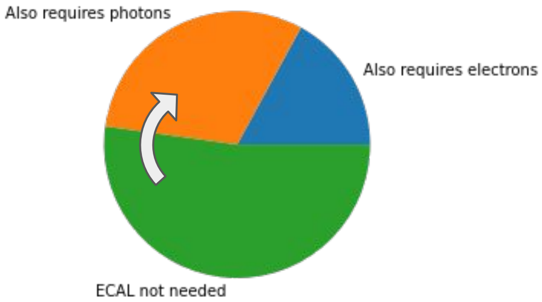
We could save a lot of computing resources by parametrizing the **detector response** to that *kaon* and applying it to whatever decay model.

Or, with **Parametric Simulation.**

2016 simulated samples | Number of events



2016 simulated samples | Data size

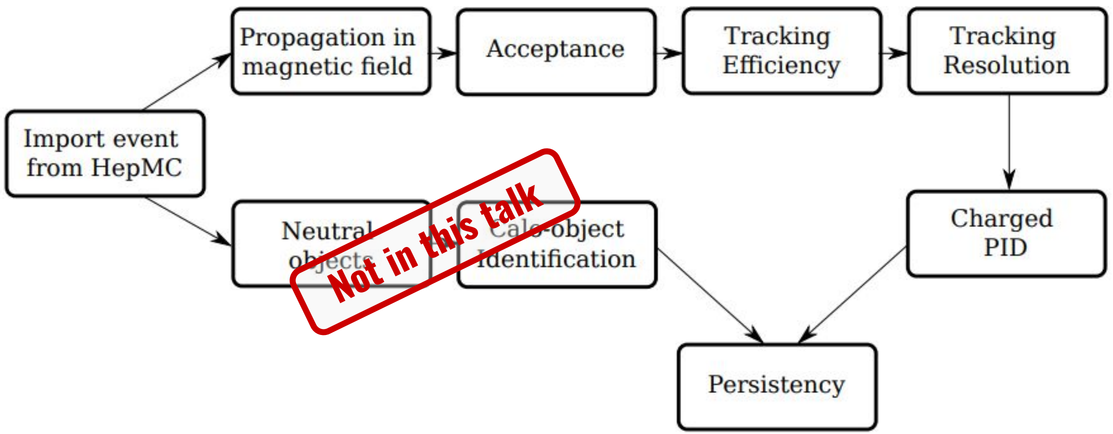


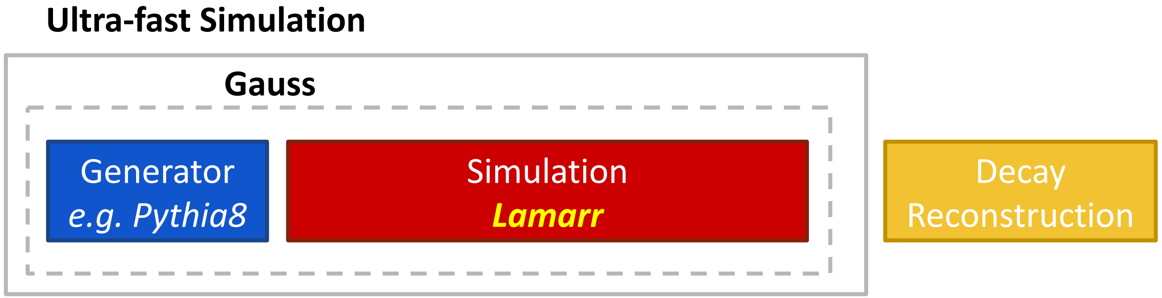
Analyses involving $h^\pm$ and $\mu^\pm$, only, often **drop simulated raw detector information** immediately.

Lamarr: *a pipeline of parameterizations embedded in Gauss*

Lamarr is a pipeline of **modular parametrizations**, integrated with the LHCb analysis framework:

- compatibility of the same, LHCb-tuned, **generators**
- compatibility with the **distributed computing** middleware (LHCbDirac) and production environment
- producing datasets with same **persistency** format





Ultra-fast Simulation

Gauss

Generator
e.g. Pythia8

Simulation
Lamarr

Decay Reconstruction

Machine Learning parametrizations: two families

see the dedicated presentation: M. Barbetti "Simulating the LHCb experiment with Generative Models" at *Learning To Discover*, April 2022, Orsay

Efficiencies

Gradient Boosted Decision Trees (GBDTs) trained on simulated data with *Binary* or *Categorical Cross Entropy* to predict the fraction of "*good**" candidates, *i.e.* the "efficiency"of a specific step as a function of generator-level quantities.

- GBDTs are robust and easy to train
- Almost no preprocessing is needed

* either "accepted", "reconstructed", "selected"... depending on the context

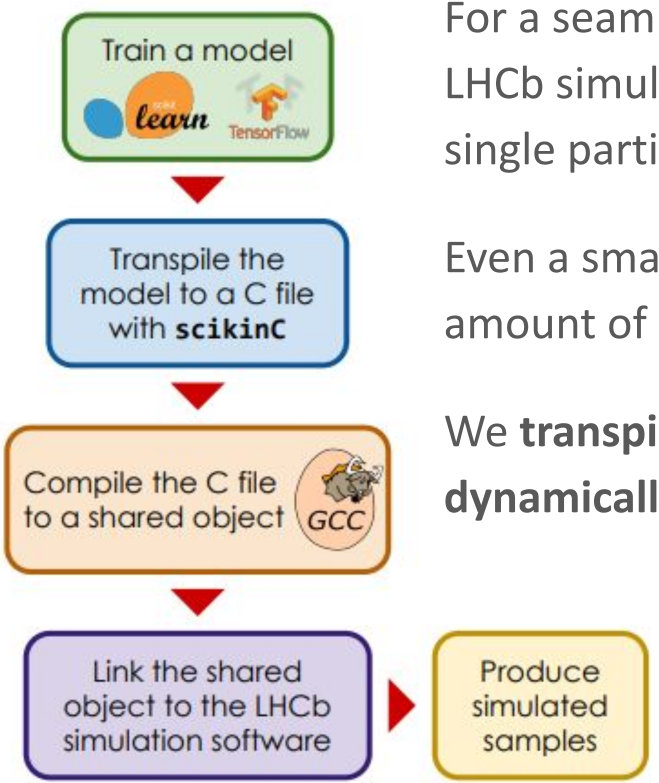
Reconstructed quantities

Conditional **Generative Adversarial Networks** trained on either simulated or calibration data.

Various GAN flavours adopted for different parameterizations balancing between accuracy and robustness.

Training is performed on **opportunistic GPU resources** provided to the Collaboration.

Deployment of ML models in Gauss



Train a model

Transpile the model to a C file with `scikinC`

Compile the C file to a shared object

Link the shared object to the LHCb simulation software

Produce simulated samples

For a seamless integration of the trained parameterizations in the LHCb simulation framework models have to be applied to each single particle → **thousands of independent calls per event**.

Even a small latency (*e.g. context switching*) wastes unacceptable amount of CPU resources .

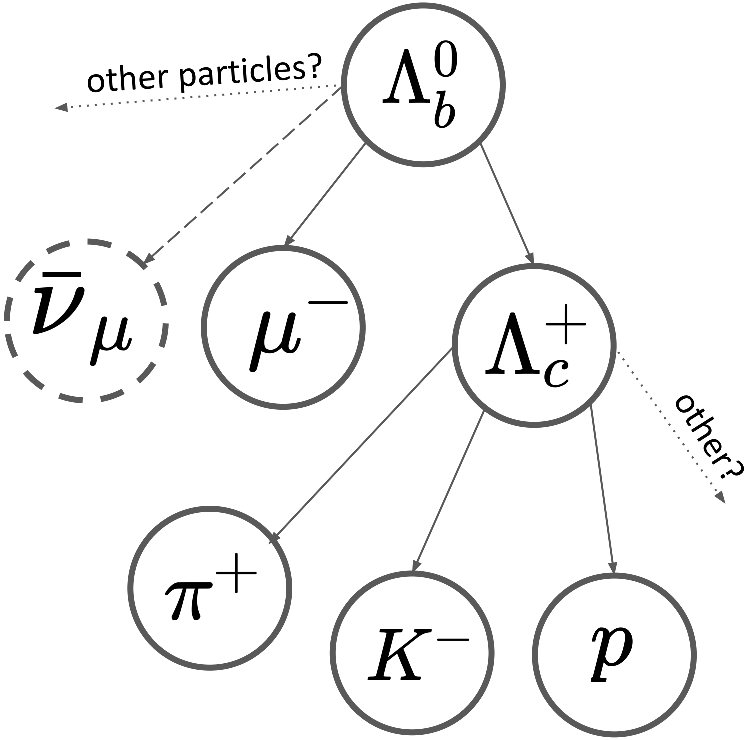
We **transpile our models in C** and compile them to binaries, **dynamically linked** at runtime.

LHCb tool: **scikinC** [PoS(CompTools2021)034, accepted]

Possible partial migration to **keras2c** [J.Eng.App.AI, (2021) 104182]

Lamarr validation: $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ with $\Lambda_c^+ \to p K^- \pi^+$

- Abundant decay in LHCb, widely studied to measure, *e.g.*, beauty baryon production
  - *e.g. see* JHEP 10 (2021) 060, PHYS. REV. D100 (2019) 032001, PHYS. REV. D96 (2017) 112005, …
- It is part of the Particle Identification Calibration samples [EPJ TI 2019 6:1];
- It is described by a complex decay model including many feed-down modes;
- It provides examples for **muons**, **pions**, **kaons** and **protons** in a single decay mode.

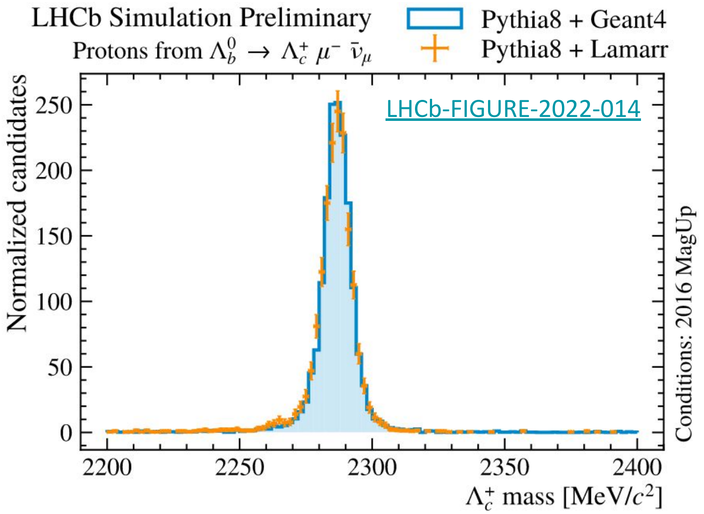


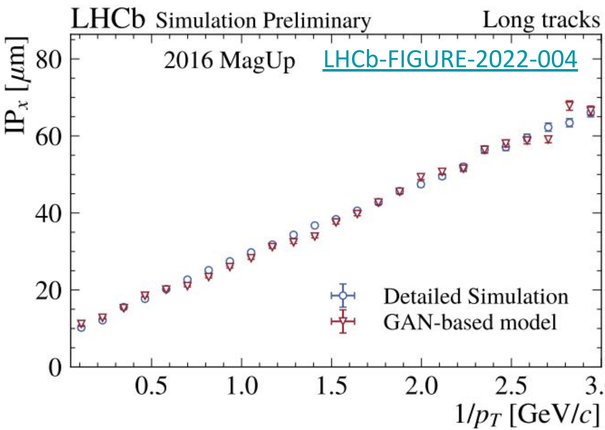
The training of the models is based on a cocktail of heavy flavour decays, where $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ represents a negligible fraction.

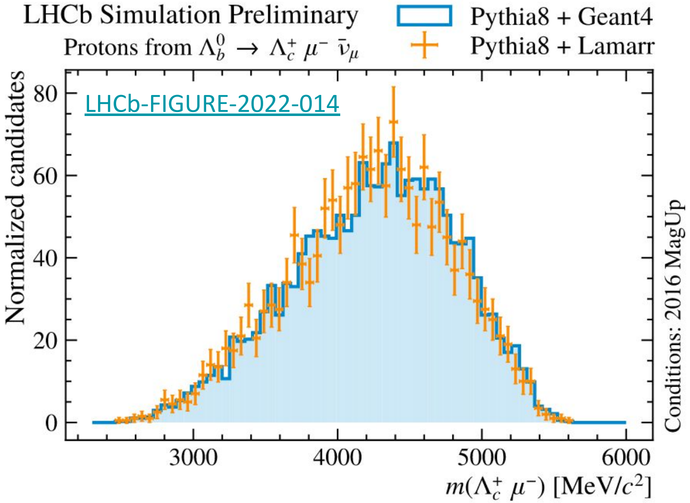
Track smearing

The momentum and point of closest approach to the beams of the generated particles **get smeared**: a GAN predicts effects as *multiple scattering*, imperfections of alignment, calibration…

Reconstructed masses and **impact parameters** are then computed on the smeared quantities.





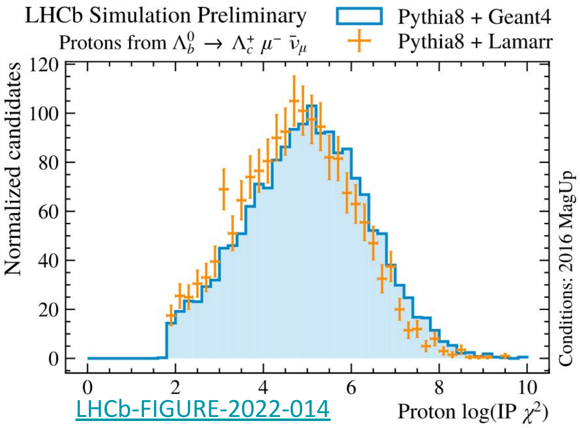


Tracking uncertainties

A GAN is used to predict the **uncertainties associated to the track reconstruction**.

Track uncertainties are crucial in LHCb to define the **consistency of trajectories with vertices**.

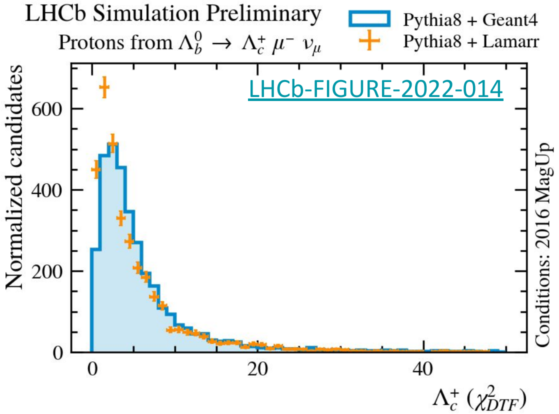
For example, the **impact parameter χ^2** is a measure of inconsistency of a trajectory with a PV.



LHCb-FIGURE-2022-014

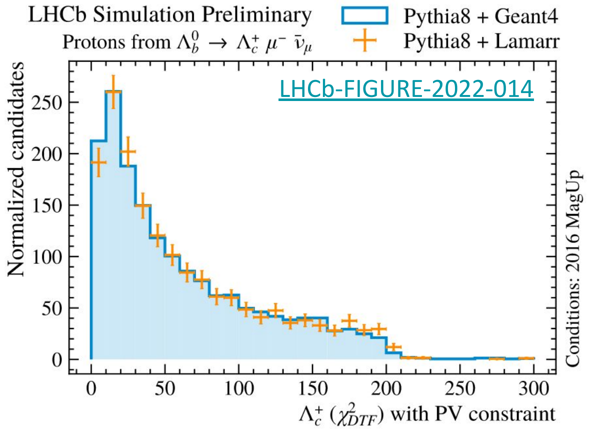
The whole decay tree can be fitted at once with a ***Decay Tree Fitter*** [NIMA 552 (2005) 566]

Constraining the Λ_c^+ to be produced in the PV, the χ^2 explodes.



LHCb-FIGURE-2022-014

Include PV constraint



LHCb-FIGURE-2022-014

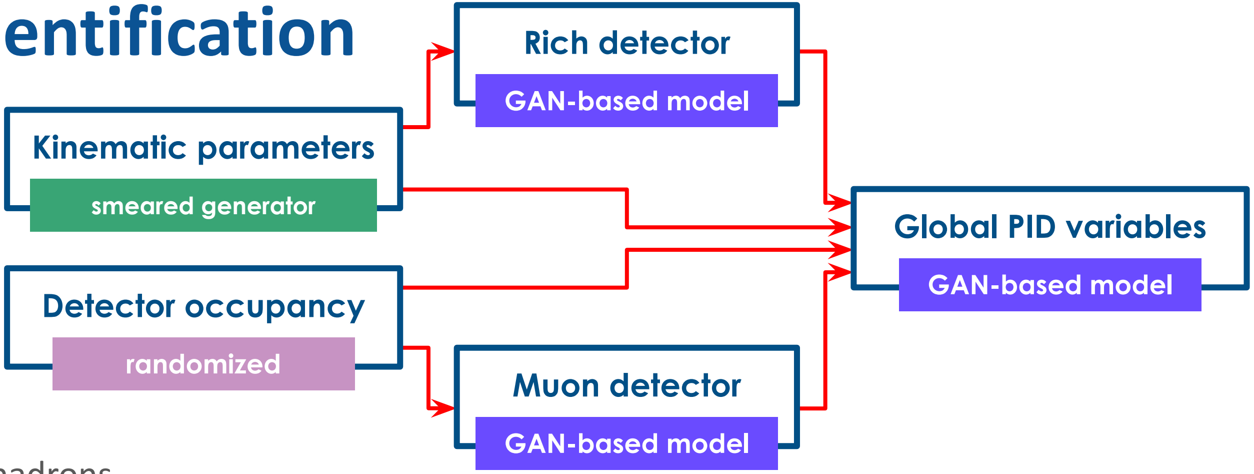
Charged Particle Identification

Particle ID at LHCb:

- Two **RICH** detectors
  - mainly for hadron identification
- A **MUON** detector
  - to distinguish muons from hadrons
- An **ECAL** (Electromagnetic Calorimeter)
  - mainly for electrons and photons

Each of these detectors provides a classifier.

Higher level, multivariate classifiers have been developed to combine effectively the response of each system.



Kinematic parameters
smeared generator

Detector occupancy
randomized

Rich detector
GAN-based model

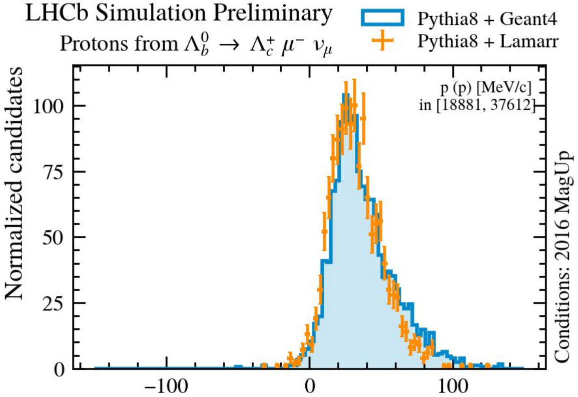
Muon detector
GAN-based model

Global PID variables
GAN-based model

The adopted stacked GAN structure is designed to simulate both single-system **detector response** and **higher-level classifiers**, enabling analysts to define new higher level classifiers based on the underlying basic quantities.

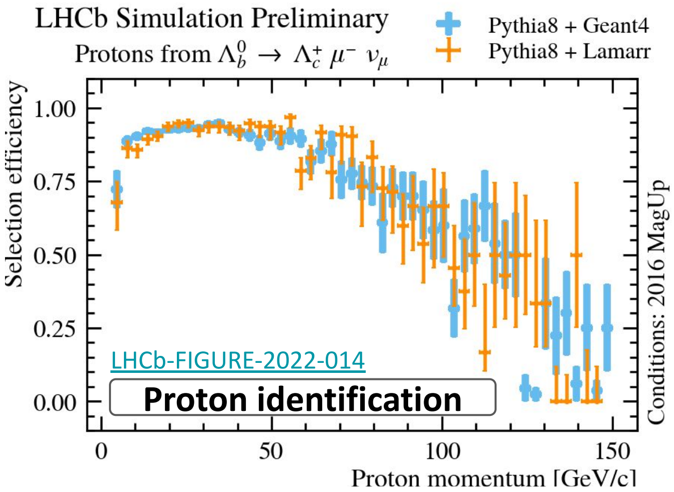
Proton identification

Lamarr simulates the distribution of the detector response. Analysts often inject the detector response in some analysis-specific classifier.



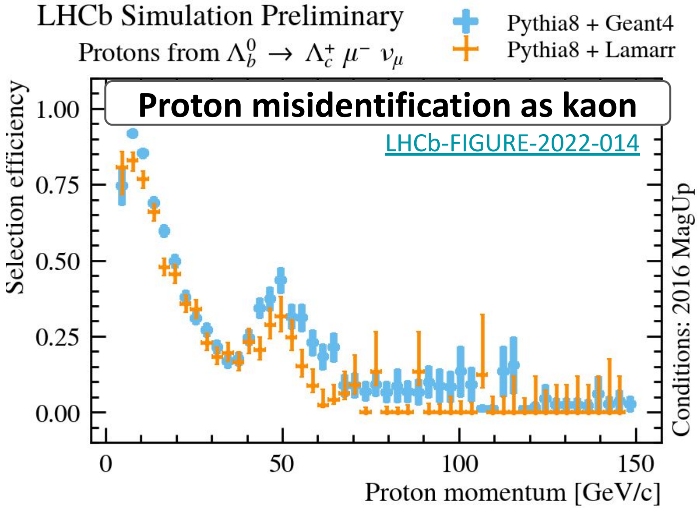
LHCb-FIGURE-2022-014

Here, we define cuts to visualize the ability of the trained models to describe the **dependence of the detector response on occupancy and kinematics**.



LHCb-FIGURE-2022-014

Proton identification



Proton misidentification as kaon

LHCb-FIGURE-2022-014

One more word on timing

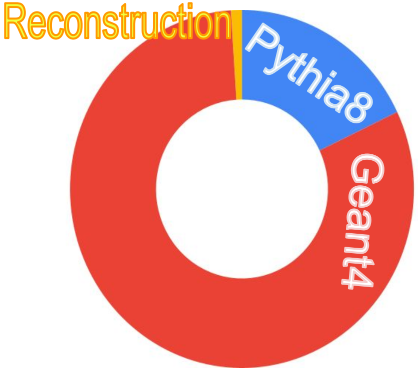
Comparing the normalized CPU spent for Geant4-based and Lamarr simulations of $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ decays we estimate a CPU reduction of 98.3 % for the ***Simulation* phase**.

Generation of *b*-baryons is exceptionally expensive: here Pythia 8 largely dominates the CPU consumption.

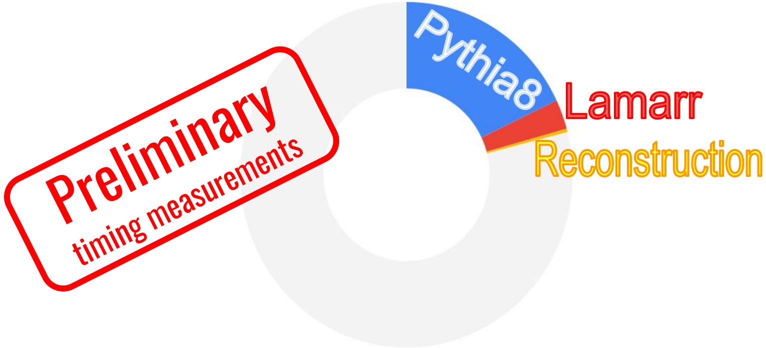
Generation of *b*-mesons requires 5% of less of the overall Simulation time.

Repeating the exercise **on minimum bias, CPU reduction exceeds 99%.**

Detailed simulation: Pythia8 + Geant4
1M events @ 2.5 kHS06.s/event ≃ 80 HS06.y



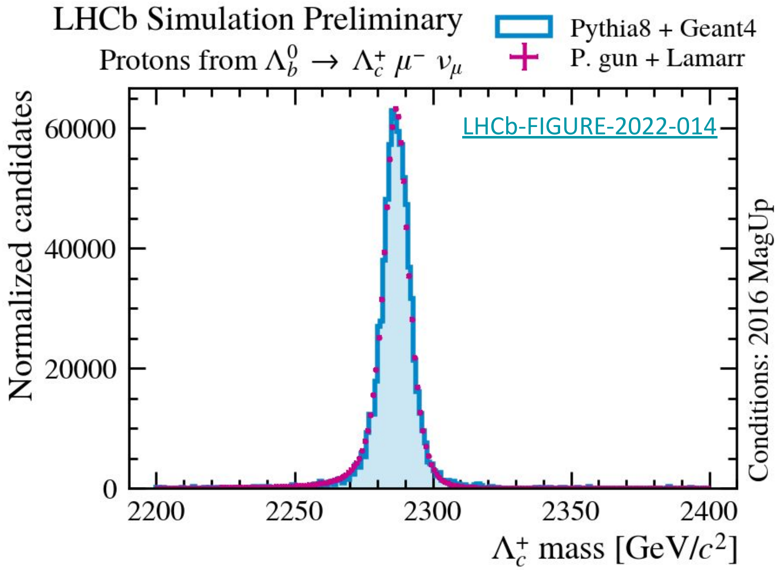
Ultra-fast simulation: Pythia8 + Lamarr
1 M events @ 0.5 kHS06.s/event ≃15 HS06.y

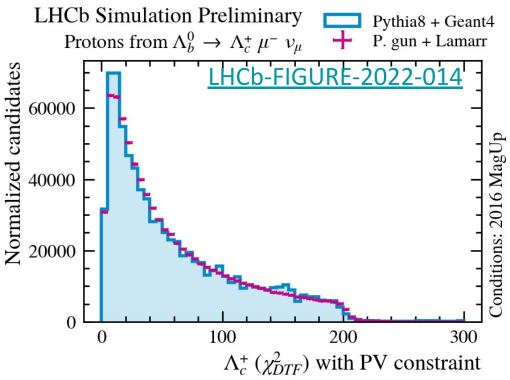


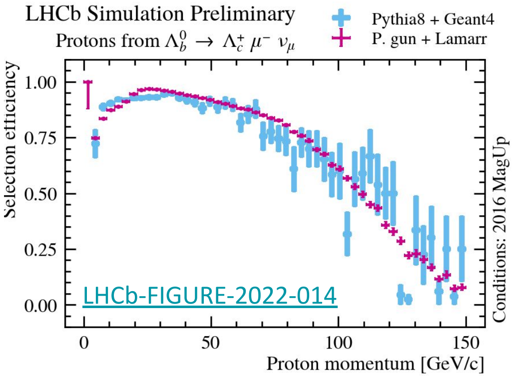
Saving more with Particle Guns

Detector occupancy is parametrized: one can achieve similar performance by **only simulating the signal particles** (*i.e.* with *Particle Guns*).

Production spectra are generated once-for-all with Pythia8 and then sampled.







Detailed simulation: Pythia8 + Geant4
1M events @ 2.5 kHS06.s/event ≃ 80 HS06.y

Ultra-fast simulation: Pythia8 + Lamarr
1 M events @ 0.5 kHS06.s/event ≃ 15 HS06.y

Ultra-fast simulation: Particle Gun + Lamarr
100 M events @ 1 HS06.s/event ≃ 4 HS06.y

Technology Readiness Level & Limitations

Private productions are currently run on the **LHCb Grid with Dirac** and could be made standard, centralized productions easily.

We are now **tuning models to compromise between accuracy and CPU performance**, focusing on 2016 datataking conditions. We plan to extend to 2015, 2017 and 2018 soon. Run1 support may come later.

Lamarr will never replace Detailed Simulation, but may provide soon a precious tool to **design selection** strategies, **train multivariate classifiers**, study kinematic-induced **correlation effects** in the analysis-level quantities, or in general when **theoretical uncertainties** on the decay model are large.

As of today, these use-cases are mostly covered with *Detailed Simulation*.

Conclusion and outlook

Great progress is being made on developing a **fully parametric simulation of the LHCb experiment**, aiming at reducing the pressure on the CPU computing resources.

Additional features requiring more **advanced AI techniques** are being investigated, *e. g.*

- ➢ *improved robustness of the tail description* [more in **S. Mokhnenko's talk at ACAT`21**]
- ➢ *prediction of the performance uncertainties due to GAN training* [more in **N. Kazeev's poster at ACAT`21**]
- ➢ *application of quasi-parametric, domain-specific density-based models* [**JINST 17 (2022) P02018**]

Model development, tuning and specialization will continue taking great advantage of **opportunistic GPU resources** made available to the Collaboration.