Lamarr: LHCb ultra-fast simulation based on machine learning models

in *21st International Workshop on Advanced Computing and Analysis Techniques in Physics Research* (ACAT 2022)

M. Barbetti[a,b] on behalf of the LHCb Simulation Project

[a]University of Florence, [b]INFN Firenze

1. Motivation

During the LHC Run 2, the LHCb experiment has spent **more than 80% of the pledged CPU time** to produce simulated samples. Run 3 CPU resource needs will far exceed the computing resources available to the LHCb Collaboration, that is spending huge efforts in developing **faster options for simulation**, like the new Lamarr framework.

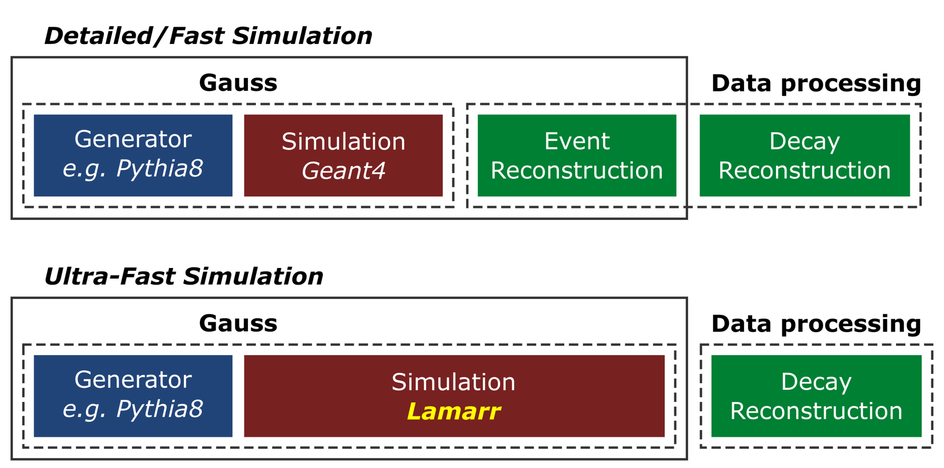
2. What is Lamarr?

The new *ultra-fast simulation* framework for LHCb is named **Lamarr**[1] and is embedded within the LHCb simulation framework Gauss. Lamarr consists of a pipeline of (ML-based) **modular parameterizations** designed to replace both the physics simulation and the reconstruction steps.

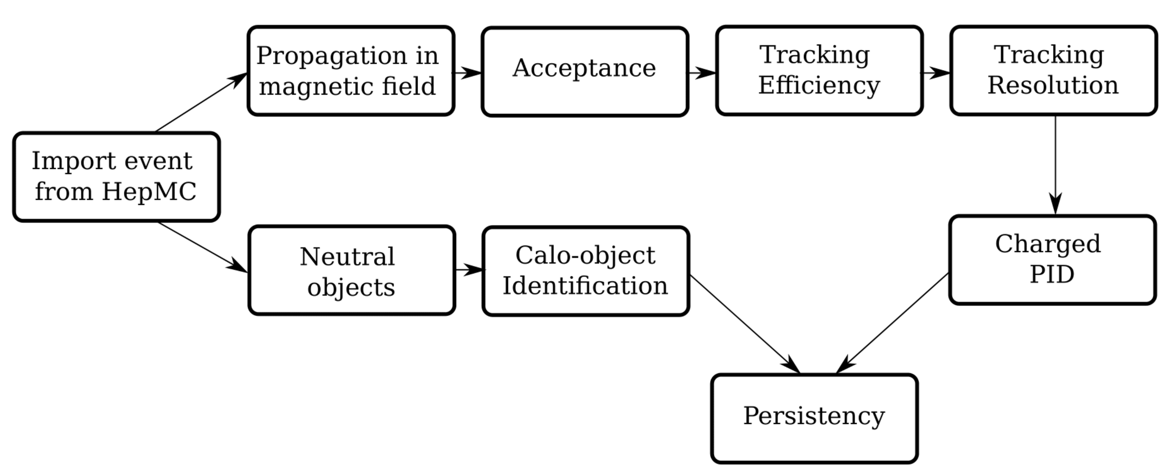
- Compatibility with LHCb-tuned **generators** (e.g. Pythia8, Particle Guns);
- Promotion of generator-level particles to successfully **reconstructed** candidates;
- Possibility of submitting *Lamarr jobs* through the LHCb **distributed computing** middleware Dirac;
- Capability of producing datasets with the same **persistency** format as the LHCb physics analysis framework DaVinci.

[1] The framework name comes from Hedy Lamarr, that was an Austrian-born American film actress and inventor. Read more on Wikipedia.

3. Pipeline of modular parameterizations



Schematic representation of the data processing flow in *detailed* and *fast simulation* (top), and in *ultra-fast simulation* (bottom).



Schematic representation of the modular pipeline provided by Lamarr to transform information from generators into high-level quantities.

4. ML-based parameterizations

Efficiencies: *Gradient Boosted Decision Trees* (GBDT) trained on simulated data to predict the fraction of accepted / reconstructed / selected candidates.

High-level quantities: Conditional *Generative Adversarial Networks* (GAN) trained on either simulated or calibration data to synthetize the high-level response of LHCb sub-detectors.

5. Model deployment within Gauss

Best-performing parameterizations can easily replace specific modules without recompiling the whole pipeline using the deployment tool `scikinC`.

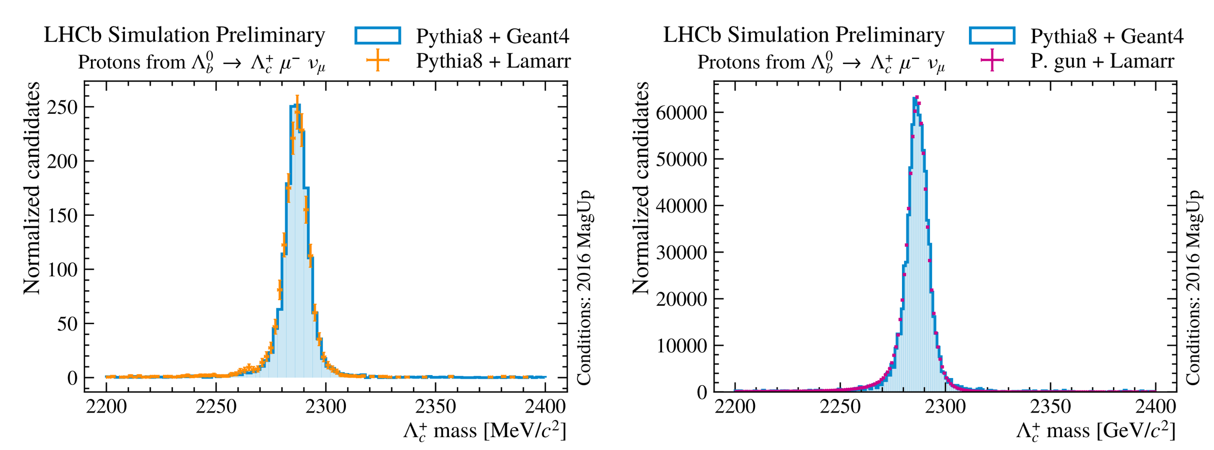
`scikinC` translates ML-based models to be dynamically linked to the main application (Gauss). In this way, parameterizations can be developed and released **independently**.

- 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**.

6. Validation campaign

Lamarr is currently under validation, comparing the distributions of the **analysis-level reconstructed** quantities parameterized with what obtained from *detailed simulation* for $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ decays with $\Lambda_c^+ \to pK^-\pi^+$.

- Decay abundantly produced in the LHCb acceptance, widely studied, and also utilized as *PID calibration sample*;
- 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.



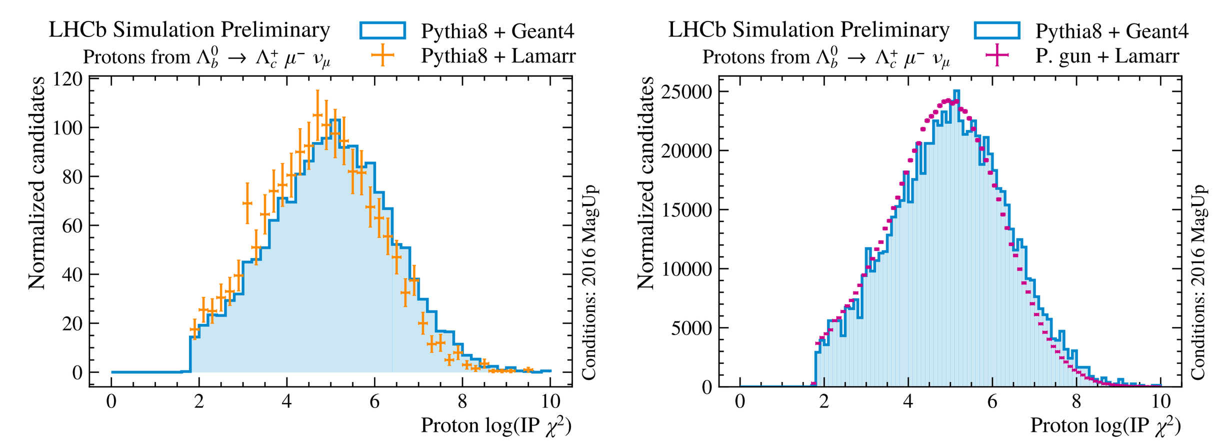
Λ_c^+ mass obtained from Pythia8 (left) and Particle Gun (right) generators by Lamarr against *detailed simulation*. Reproduced from LHCB-FIGURE-2022-014.

7. Results: Tracking system

The momentum and the point of closest approach to the beams at generator-level **get smeared**: GAN-based model is used to parameterize *multiple scattering* and residual detector effects (alignment, calibration).

Track reconstruction **uncertainties** rely on dedicated GAN-based model. Correct modeling track uncertainties is essential for LHCb analyses: e.g., the *impact parameter* (IP) is a common discriminator between prompt and displaced vertices.

Output quantities can be used within LHCb offline reconstruction to compute **higher-level quantities**, like the reconstructed mass.



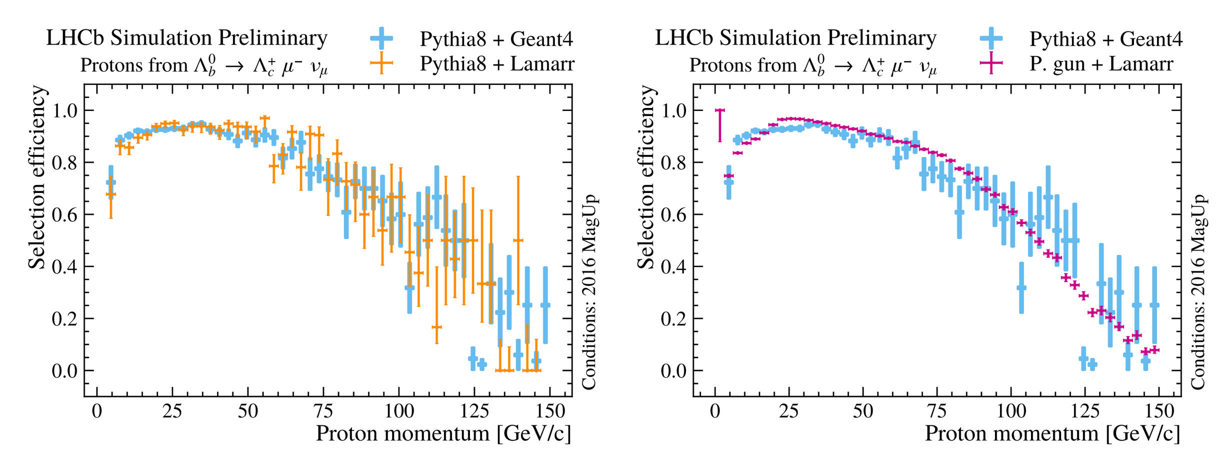
Proton impact parameter (IP) χ^2 obtained from Pythia8 (left) and Particle Gun (right) generators by Lamarr against *detailed simulation*. Reproduced from LHCB-FIGURE-2022-014.

8. Results: PID system

Smeared *track kinematics* and *detector occupancy* are used by two sets of GAN-based models to parameterize the **high-level response** of the RICH and MUON systems.

Further GAN-based models are trained to reproduce the **higher-level PID classifiers** typically used in physics analyses, relying only on the input and the output of RICH and MUON parameterizations.

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



Proton identification efficiency obtained from Pythia8 (left) and Particle Gun (right) generators by Lamarr against *detailed simulation*. Reproduced from LHCB-FIGURE-2022-014.

9. Timing performance

Overall time needed for producing simulated samples has been analyzed for fully *detailed simulation* (Geant4-based propagation) and Lamarr. Lamarr timing is dominated by **particle generation** (Pythia8).

Preliminary studies show that Lamarr ensure a **CPU reduction of at least 98%** for the physics simulation phase. Further improvement in timing can be achieved tacking the generation, as shown when using Particle Guns (e.g. only generating signal of interest).

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

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

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

10. Conclusions and outlook

Great progress has been made on developing a **fully parametric simulation** of the LHCb experiment, aiming to reduce the pressure on the CPU computing resources.

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

- Further speed improvements under study;
- *Thread safety* for **multithreaded Gaudi algorithms** under development.

References

1. M. Borisyak and N. Kazeev, *Machine Learning on data with sPlot background subtraction*, JINST **14** (2019) P08020, arXiv:1905.11719
2. A. Maevskiy *et al.*, *Fast Data-Driven Simulation of Cherenkov Detectors Using Generative Adversarial Networks*, J. Phys. Conf. Ser. **1525** (2020) 012097, arXiv:1905.11825
3. L. Anderlini, *Machine Learning for the LHCb Simulation*, arXiv:2110.07925
4. L. Anderlini *et al.*, *Towards Reliable Neural Generative Modeling of Detectors*, arXiv:2204.09947
5. C. Bozzi, *LHCb Computing Resource usage in 2021*, LHCb-PUB-2022-011
6. L. Anderlini and M. Barbetti, *scikinC: a tool for deploying machine learning as binaries*, PoS **CompTools2021** (2022) 034
7. L. Anderlini *et al.*, *Lamarr: the ultra-fast simulation option for the LHCb experiment*, PoS **ICHEP2022** 233 (in preparation)