Lamarr: implementing a flash-simulation paradigm at LHCb

in *22nd International Workshop on Advanced Computing and Analysis Techniques in Physics Research* (ACAT 2024)

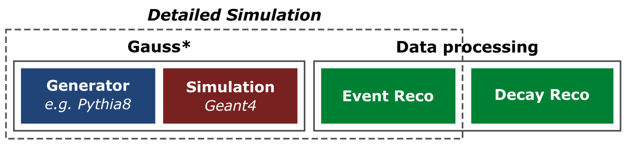
M. Mazurek[a] on behalf of the LHCb Simulation Project

[a]National Centre for Nuclear Research (NCBJ), Poland

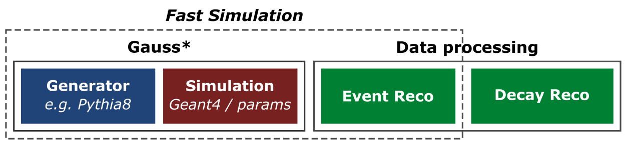
1. Motivation

The **detailed simulation** of the interaction between the traversing particles and the LHCb active volumes is the major consumer of CPU resources. During the LHC Run2, the LHCb experiment has spent **more than 90% of the pledged CPU time** to produce simulations. Matching the upcoming and future demand for simulated samples means that the development of **faster simulation options** is critical.

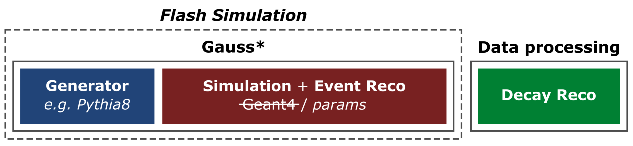
2. Fast simulation VS. flash simulation



Detailed simulation relies on Geant4 to reproduce the radiation-matter interactions that are computed within Gauss*, the LHCb simulation software.



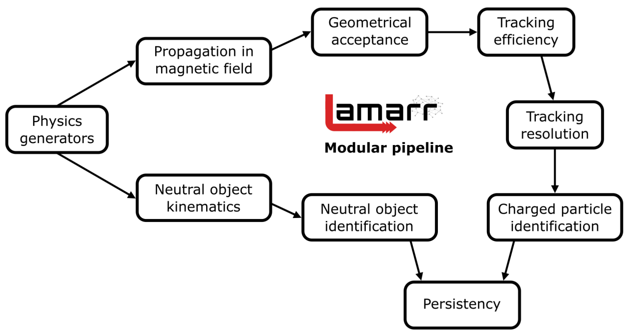
Fast simulation techniques aim to speed up the Geant4-based simulation production by parameterizing the energy deposits instead of relying on physics.



Flash (or *Ultra-Fast*) ***simulation*** strategies aim to directly transform generator-level particles into analysis-level reconstructed objects.

3. What is Lamarr?

Lamarr is the novel flash-simulation framework of LHCb, able to offer the fastest option for simulation. Lamarr consists of a **pipeline of** (ML-based) **modular parameterizations** designed to replace both the simulation and reconstruction steps.



The Lamarr pipeline can be split in two chains:

1. a branch treating **charged particles** relying on tracking and particle identification models;
2. a branch facing the *particle-to-particle correlation* problem innate in the **neutral objects** reconstruction.

4. Models under the k-to-k hypothesis

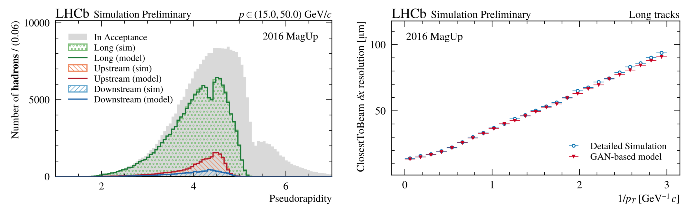
Assuming the existence of an **unambiguous** (k-to-k) **relation** between generated particles and reconstructed objects, the high-level detector response can be modeled in terms of **efficiency** and **"resolution"** (i.e., analysis-level quantities):

- **Efficiency:** *Deep Neural Networks* (DNN) trained to perform classification tasks so that they can be used to parameterize the fraction of "good" candidates (e.g., accepted, reconstructed, or selected).
- **Resolution:** Conditional *Generative Adversarial Networks* (GAN) trained on detailed simulated samples to parameterize the high-level response of LHCb detector (e.g., reconstruction errors, differential log-likelihoods, or multivariate classifier output).

5. Charged particles pipeline: the tracking system

Lamarr parameterizes the high-level response of the **LHCb tracking system** relying on the following models:

- **propagation:** approximates the trajectory of charged particles through the dipole magnetic field (parametric model);
- **geometrical acceptance:** predicts which of the generated tracks lay within a sensitive area of the detector (DNN model);
- **tracking efficiency:** predicts which of the generated tracks in acceptance are properly reconstructed by the detector (DNN model);
- **tracking resolution:** parameterizes the errors introduced by the reconstruction algorithms to the track parameters (GAN model);
- **covariance matrix:** parameterizes the uncertainties assessed by the Kalman filter procedure (GAN model).



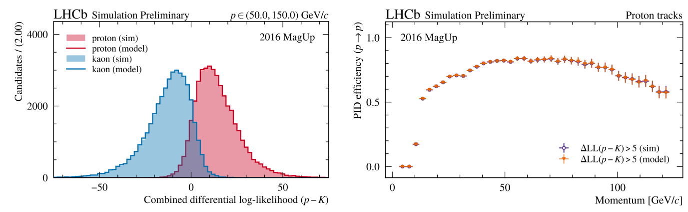
Validation plots for the DNN-based model of the tracking efficiency (left) and the GAN-based model of the spatial tracking resolution (right).

6. Charged particles pipeline: the PID system

Lamarr parameterizes the high-level response of the **LHCb PID system** relying on the following models:

- **RICH:** parameterizes DLLs resulting from the RICH detectors (GAN model);
- **MUON:** parameterizes likelihoods resulting from the MUON system (GAN model);
- **isMuon:** parameterizes the response of a FPGA-based criterion for muon loose boolean selection (DNN model);
- **Global PID:** parameterizes the global high-level response of the PID system, consisting of CombDLLs and ProbNNs (GAN model).

Lamarr provides separated models for **muons**, **pions**, **kaons**, and **protons** for each PID set of variables.



Validation plots for the proton-kaon separation parameterized with the GAN-based models of the Global PID response in terms of distributions (left) and proton selection efficiency (right).

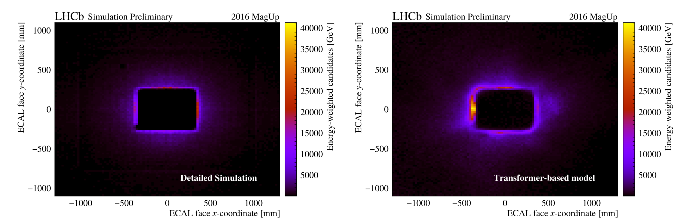
7. Neutral particles pipeline: the ECAL detector

The flash simulation of the LHCb ECAL detector is a non-trivial task:

- bremsstrahlung radiation, converted photons, or merged π^0 may lead to have n **generated particles** responsible for m **reconstructed objects** (in general, with $n \neq m$);
- the ***particle-to-particle correlation*** **problem** limits the validity of strategies used for modeling the unambiguous k-to-k detector response.

To parameterize a generic n-to-m response of the ECAL detector, solutions inspired by the natural language **translation problem** are currently under investigation:

- the aim is to define an **event-level description** of the ECAL response;
- assuming ordered sequences of photons/clusters, the problem can be modeled with a ***Transformer*** model;
- complying with the problem topology, the ECAL response can be modeled with a ***Grapha Neural Network*** (GNN) model



Validation plots for the (x, y)-position of the ECAL clusters as reconstructed by detailed simulation (left) and a Transformer-based model (right). Each bin entry is properly weighted to include also the energy signature.

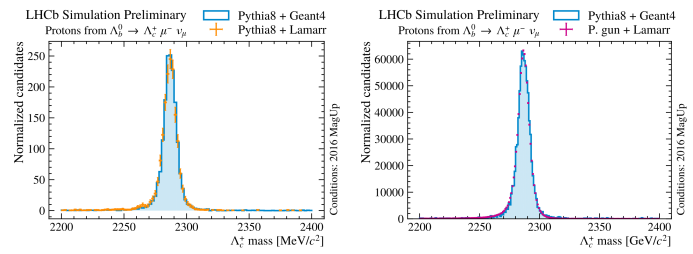
8. Validation campaign

Lamarr provides the high-level response of the LHCb detector by relying on a **pipeline of** (subsequent) **ML-based modules**. To validate the charged particles chain, the distributions of a set of **analysis-level** reconstructed quantities resulting from Lamarr have been compared with that obtained from detailed simulation for $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ decays with $\Lambda_c^+ \to pK^-\pi^+$.

The deployment of the ML-based models follows a ***transcompilation approach*** based on **`scikinC`**. The models are translated to C files, compiled as *shared objects*, and then dynamically linked to the LHCb simulation software (Gauss).

The integration of Lamarr with Gauss unlocks:

- interface with all the **LHCb-tuned physics generators** (e.g., Pythia8, EvtGen);
- compatibility with the **distributed computing middleware** and production environment;
- providing **ready-to-use datasets** for centralized analysis.



Validation plots for the Λ_c^+ mass obtained from Pythia8 (left) and particle-gun (right) generators by Lamarr against detailed simulation. Reproduced from LHCB-FIGURE-2022-014.

9. Preliminary timing studies

Overall time needed for producing simulated samples has been analyzed for fully detailed simulation (Geant4-based propagation) and Lamarr. When Lamarr is employed, the particle generation (in particular, Pythia8) becomes the new **major CPU consumer**.

Lamarr allows to reduce the CPU cost for the simulation phase of (at least) **two-order-of-magnitude**. Further timing improvements can be achieved by generating only the signal of interest (i.e., particle-gun approach).

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 effort is ongoing to put a **fully parametric simulation** of the LHCb experiment into production, aiming to reduce the pressure on computing resources.

DNN-based and GAN-based models succeed in describing the high-level response of the LHCb tracking and PID detectors for **charged particles**, while work is still required to parameterize the response of the ECAL detector due to the **particle-to-particle correlation problem**.

The future development of Lamarr looks to design a flash-simulation framework that, although integrated within the LHCb software stack, can also be run as **standalone**.

References

1. V. Chekalina *et al.*, *Generative Models for Fast Calorimeter Simulation: the LHCb case*, EPJ Web Conf. **214** (2019) 02034, arXiv:1812.01319
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 and M. Barbetti, *scikinC: a tool for deploying machine learning as binaries*, PoS **CompTools2021** (2022) 034
4. A. Rogachev and F. Ratnikov, *GAN with an Auxiliary Regressor for the Fast Simulation of the Electromagnetic Calorimeter Response*, J. Phys. Conf. Ser. **2438** (2023) 012086, arXiv:2207.06329
5. L. Anderlini *et al.*, *Lamarr: the ultra-fast simulation option for the LHCb experiment*, PoS **ICHEP2022** (2023) 233
6. M. Barbetti, *Lamarr: LHCb ultra-fast simulation based on machine learning models deployed within Gauss*, arXiv:2303.11428
7. F. Vaselli *et al.*, *FlashSim prototype: an end-to-end fast simulation using Normalizing Flow*, CERN-CMS-NOTE-2023-003
8. L. Anderlini *et al.*, *The LHCb ultra-fast simulation option, Lamarr: design and validation*, arXiv:2309.13213
9. M. Barbetti, *The flash-simulation paradigm and its implementation based on Deep Generative Models for the LHCb experiment at CERN*, PhD thesis, University of Firenze, 2024