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

Matteo Barbetti[1,2] **on behalf of the LHCb Simulation Project**

[1] Department of Information Engineering, University of Florence, via Santa Marta 3, Firenze (FI), Italy

[2] Istituto Nazionale di Fisica Nucleare, Sezione di Firenze, via G. Sansonse 1, Sesto Fiorentino (FI), Italy

E-mail: Matteo.Barbetti@fi.infn.it

Abstract. About 90% of the computing resources available to the LHCb experiment has been spent to produce simulated data samples for Run 2 of the Large Hadron Collider at CERN. The upgraded LHCb detector will be able to collect larger data samples, requiring many more simulated events to analyze the data to be collected in Run 3. Simulation is a key necessity of analysis to interpret signal vs background and measure efficiencies. The needed simulation will far exceed the pledged resources, requiring an evolution in technologies and techniques to produce these simulated data samples. In this contribution, we discuss Lamarr, a Gaudi-based framework to speed-up the simulation production parametrizing both the detector response and the reconstruction algorithms of the LHCb experiment. Deep Generative Models powered by several algorithms and strategies are employed to effectively parametrize the high-level response of the single components of the LHCb detector, encoding within neural networks the experimental errors and uncertainties introduced in the detection and reconstruction phases. Where possible, models are trained directly on real data, statistically subtracting any background components through weights application. Embedding Lamarr in the general LHCb Gauss Simulation framework allows to combine its execution with any of the available generators in a seamless way. The resulting software package enables a simulation process completely independent of the Detailed Simulation used to date.

1. Introduction

The LHCb detector [1, 2], originally designed to study particles containing b and c quarks produced at the Large Hadron Collider (LHC), is a single-arm forward spectrometer covering the pseudorapidity range $2 < \eta < 5$. The detector includes a high-precision tracking system providing measurements of the momentum p of charged particles and the minimum distance of a track to a primary vertex (PV), namely the impact parameter (IP). LHCb is also equipped with a highly performing particle identification (PID) system capable of distinguishing photons, electrons, long-lived hadrons, and muons, combining the response of two ring-imaging Cherenkov (RICH) detectors, the calorimeter system, and the MUON system.

The simulation of high-energy collisions, of the decays of the generated particles, and of the physics processes occurring within the detector by the decay products are a key necessity of analysis, typically for separating the signal from background sources or for efficiency studies. The simulation software of the LHCb experiment is built upon two main projects named Gauss and Boole [3], both based on the Gaudi framework [4]. The Gauss framework implements the so-called generation and simulation phases, while the Boole application is responsible for the digitization phase. The first step of any simulation production is the *generation* phase in which the high-energy collisions are simulated with Monte Carlo generators such as Pythia8 [5] and EvtGen [6]. The output of the generation phase is the set of long-lived particles able to traverse partially or entirely, depending on the particle species, the LHCb spectrometer. The radiation-matter interactions occurring within the detector by the traversing long-lived particles are reproduced during the *simulation* phase that aims to compute the energy deposited in the active volumes relying on Geant4 [7]. Lastly, during the *digitization* phase, the energy deposits are converted into raw data mimicking the data format used in the LHCb Data Acquisition pipeline.

During the LHC Run 2, the simulation of physics events at LHCb has taken more than 80% of the distributed computing resources available to the experiment, namely the pledged CPU time. The experiment has just resumed data taking after a major upgrade and will operate with higher luminosity and trigger rates collecting data samples at least one order of magnitude larger than in the previous LHC runs. Meeting the foreseen needs in Run 3 conditions using only the traditional strategy for simulation, namely *detailed simulation*, will far exceed the pledged resources. Hence, the LHCb Collaboration is making great efforts to modernize the simulation software stack [8, 9] and develop novel and faster simulation options [10, 11, 12, 13, 14].

2. The fast and ultra-fast simulation paradigms

The *detailed simulation* of the dynamics of the hadron collisions and the interaction of all primary and secondary particles with the detector materials is extremely expensive in terms of CPU time. It is therefore no surprise that the computation of energy deposits performed by Geant4 consumes more than 90% of the CPU resources spent by LHCb for simulation.

Several strategies have been developed to reduce the computational cost of the simulation phase based on resampling techniques [15] or parameterizations of energy deposits [10, 12, 13]. These options offer cheaper alternative solutions to reproduce the low-level response of the LHCb detector and are typically named *fast simulation* strategies. The fast simulation options do not modify the traditional data processing flow described in Figure 1 (top), but rather allow to speed up the simulation phase up to a factor 20 with respect to the *detailed simulation*.

A more radical approach is the one followed by the *ultra-fast simulation* strategies which aim to parameterize directly the high-level response of the LHCb detector [11, 14]. The core idea is to develop parameterizations able to transform generator-level particles information into reconstructed physics objects as schematically represented in Figure 1 (bottom). Such parameterizations can be built using *deep generative models* that have proven to succeed in describing the response of the LHCb detector at different levels [16] and in offering reliable synthetic simulated samples [17, 18].

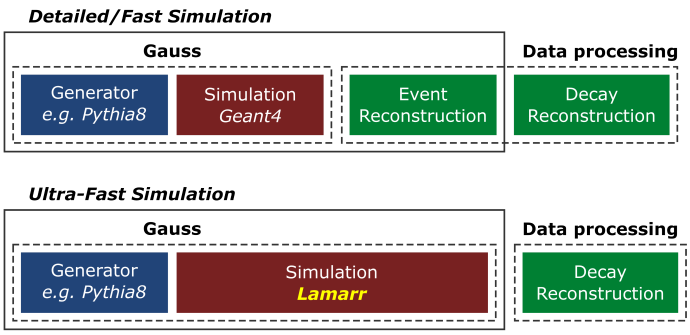

Figure 1. Schematic representation of the data processing flow in *detailed* and *fast simulation* (top), and in *ultra-fast simulation* (bottom).

3. Lamarr and its machine-learning-based parameterizations

Lamarr [14] is a novel LHCb simulation framework implementing the *ultra-fast simulation* paradigm. The Lamarr framework consists of a pipeline of modular parameterizations designed to take as input the particles generated by the event generators and provide as output high-level quantities representing the particles successfully reconstructed by LHCb. Lamarr is integrated with Gauss and disposes of a dedicated interface to the physics generators for selecting those particles that need to be propagated through the detector, splitting them into charged and neutral particles. The remainder of this document is devoted to discuss the implementation (this Section) and validation (Section 4) of the pipeline currently provided by Lamarr for charged particles.

Most of the parameterizations used by Lamarr rely on machine learning algorithms that we can split into two main classes. The first class of models uses *Gradient Boosted Decision Trees* (GBDT) to parameterize efficiencies learning the fraction of candidates that are in acceptance, that have been successfully reconstructed or that have been selected as muons. The second family of parameterizations is made up of *Generative Adversarial Networks* (GAN) [19] trained to reproduce the distributions of high-level physics quantities, typically conditioned [20] by the kinematics of the particles traversing a specific LHCb sub-detector. Additional algorithms to define detector parameterizations are being explored, but currently are not part of the Lamarr pipeline [21, 22].

Once taken the charged particles from physics generators, the first step performed by Lamarr is their propagation through the magnetic field following a trajectory approximated as two rectilinear segments with a single point of deflection (*single p_T kick* approximation). Then, the tracking acceptance and reconstruction efficiency are computed using GBDT models trained taking as input geometrical and kinematic features of the track. The resulting tracks still have information at generator-level. The promotion to high-level quantities, namely the application of the resolution effects due to, for example, multiple scattering phenomena, is carried out by GAN systems trained with *binary cross-entropy* as loss function and equipped with *skip connections* [23]. A similar GAN-based architecture is used to provide the correlation matrix obtained from the Kalman filter adopted in the reconstruction algorithm to define the position, slope and curvature of each track.

The LHCb PID system is parameterized using GAN-based models. The high-level response of the RICH and MUON systems are reproduced using the particles kinematic information provided by the Lamarr tracking modules and a description of the detector occupancy, for example based on the total number of tracks traversing the detector. The loss function adopted to train the PID-GAN models is the *Wasserstein distance* where the Lipschitz constraint on the discriminator is enforced explicitly using a method called *Adversarial Lipschitz Penalty* (ALP) regularization [24], resulting in WGAN-ALP models. GlobalPID classifiers, obtained in real data by combining RICH and MUON responses with information from the calorimeter system and features of the reconstructed tracks, are parameterized using similar GAN-based architectures that take as input what produced by the RICH-GAN and MUON-GAN models. Lastly, the efficiency of a binary muon-identification criterion, available since the earlier stage of data processing via a FPGA-based implementation, is parameterized with GBDT models.

Combining stacks of GBDT and GAN models, Lamarr provides the high-level response of the LHCb tracking and PID systems. To validate the *ultra-fast simulation* approach adopted machine-learning-based models are trained on detailed simulated samples and the output of Lamarr is compared to the reference distributions as described in Section 4. An extension of the training procedure allows to train the PID models directly on real data (in particular on calibration samples [25]), statistically subtracting any background components through weights application [26]. The trained models are deployed through a transcompilation approach using the `scikinC` toolkit and dynamically linked to the Gauss application to ease the development and prototyping of new parameterizations [27].

4. Validation campaigns powered by $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays

As mentioned in the previous Section, the validation of the ultra-fast philosophy of LAMARR is based on the comparison between the distributions obtained from models trained on detailed simulation and the ones resulting from standard simulation strategies. In particular, we discuss here the validation studies performed using simulated $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays with $\Lambda_c^+ \to pK^-\pi^+$. We are dealing with a semileptonic Λ_b^0 decay whose dynamics is not trivial and needs a faithful reproduction, highlighting the importance of interfacing to dedicated generators, in this case EVTGEN. This decay channel is being widely studied by LHCb, at the point that it is part of the calibration samples designed to provide data-driven corrections to the simulated PID efficiencies for proton candidates [25]. Interestingly, this Λ_b^0 decay includes in its final state the four charged particle species parameterized in the current version of LAMARR, namely muons, protons, kaons and pions.

The validation of LAMARR tracking modules is reported in Figure 2 (top) where a comparison between the distributions of the proton impact parameter χ^2 (top left) and of the Λ_c^+ invariant mass (top right) are shown. IP χ^2 represents a measure of the inconsistency of the proton track with the PV obtained executing the same analysis algorithm both on LAMARR output and detailed simulated samples. The agreement between the two invariant mass distributions proves that the decay dynamics is well reproduced and the resolution effects correctly parameterized. To show the performance of the LAMARR PID parameterizations, the distribution of the Combined

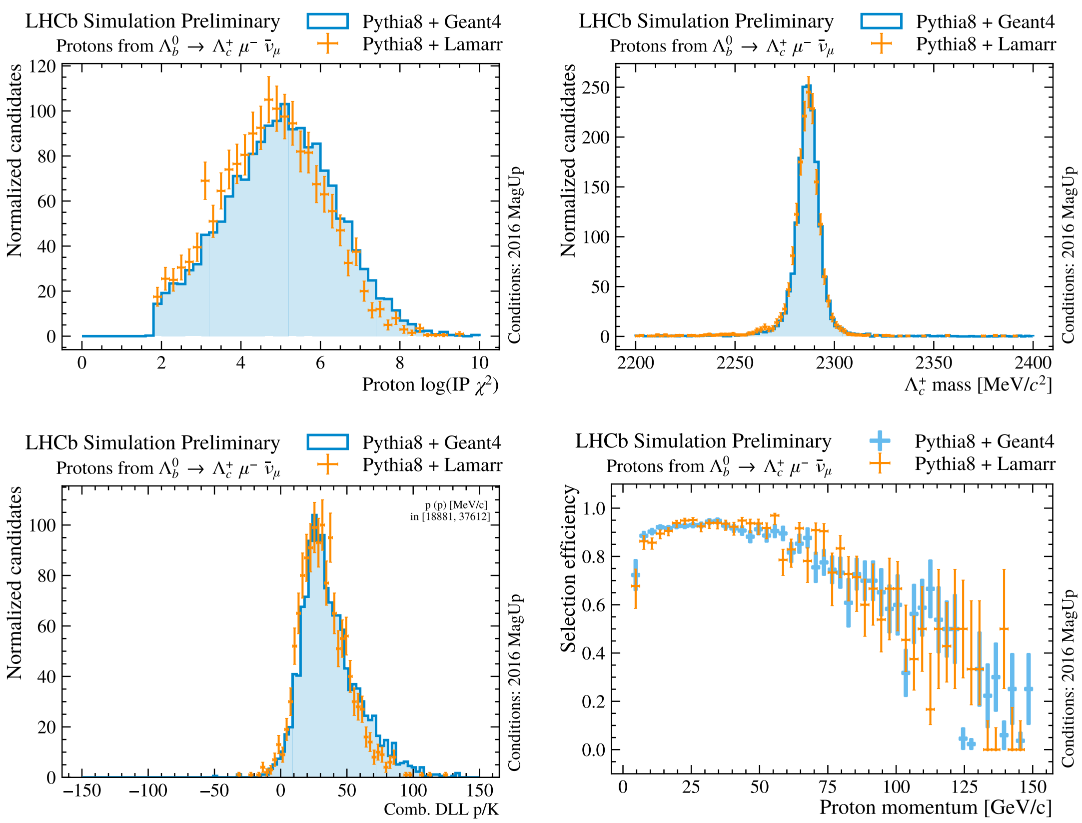

Figure 2. Validation plots for $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays with $\Lambda_c^+ \to pK^-\pi^+$ simulated with PYTHIA8, EVTGEN and LAMARR (orange markers) and compared with *detailed simulation* samples relying on PYTHIA8, EVTGEN and GEANT4 (cyan shaded histogram). Reproduced from LHCB-FIGURE-2022-014.

Differential Log-Likelihood (CombDLL) between the proton hypothesis and the kaon one on proton tracks is reported in Figure 2 (bottom left) against what expected from detailed simulated samples. A comparison between the selection efficiencies for a tight requirement on proton identification against pion hypothesis (bottom right) is also shown in Figure 2 (bottom right).

5. Conclusion

Developing new simulation techniques is an unavoidable requirement for LHCb to tackle the demand for simulated samples expected for Run 3 and those will follow. The *ultra-fast simulation* approach is a viable solution to reduce the pressure on pledged CPU resources and succeeds in describing the uncertainties introduced in the detection and reconstruction steps through the use of *deep generative models*. Such parameterization are provided to the LHCb software stack via the novel LAMARR framework, in which statistical models for tracking and charged particle identification have been deployed and validated with satisfactory results on $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$ decays. Preliminary studies show that LAMARR is able to speed up the simulation production up to a factor 1000 with respect to *detailed simulation* [14]. Improvements on the quality of the parameterizations currently provided have been planned, relying on intense optimization campaigns on distributed computing resources [28]. Further development of the neutral particles pipeline is one of the major ongoing activities with the purpose of enhancing the variety of physics analyses that can benefits from LAMARR.

Acknowledgments

We acknowledge access to MARCONI100 GPUs provided to INFN by CINECA to accomplish this study.

References

[1] Alves Jr A A *et al.* (LHCb) 2008 *JINST* **3** S08005
[2] Aaij R *et al.* (LHCb) 2015 *Int. J. Mod. Phys. A* **30** 1530022 (*Preprint* 1412.6352)
[3] Clemencic M *et al.* (LHCb) 2011 *J. Phys. Conf. Ser.* **331** 032023
[4] Barrand G *et al.* 2001 *Comput. Phys. Commun.* **140** 45–55
[5] Sjostrand T, Mrenna S and Skands P Z 2008 *Comput. Phys. Commun.* **178** 852–867 (*Preprint* 0710.3820)
[6] Lange D J 2001 *Nucl. Instrum. Meth. A* **462** 152–155
[7] Allison J *et al.* 2006 *IEEE Trans. Nucl. Sci.* **53** 270
[8] Corti G *et al.* 2023 *J. Phys. Conf. Ser.* **2438** 012108
[9] Mazurek M, Clemencic M and Corti G 2022 *PoS* **ICHEP2022** 225
[10] Rama M and Vitali G (LHCb) 2019 *EPJ Web Conf.* **214** 02040
[11] Maevskiy A *et al.* (LHCb) 2020 *J. Phys. Conf. Ser.* **1525** 012097 (*Preprint* 1905.11825)
[12] Ratnikov F and Rogachev A 2021 *EPJ Web Conf.* **251** 03043
[13] Rogachev A and Ratnikov F 2023 *J. Phys. Conf. Ser.* **2438** 012086 (*Preprint* 2207.06329)
[14] Anderlini L *et al.* 2022 *PoS* **ICHEP2022** 233
[15] Müller D *et al.* 2018 *Eur. Phys. J. C* **78** 1009 (*Preprint* 1810.10362)
[16] Ratnikov F *et al.* 2023 *Nucl. Instrum. Meth. A* **1046** 167591
[17] Anderlini L *et al.* (LHCb) 2023 *J. Phys. Conf. Ser.* **2438** 012088 (*Preprint* 2210.09767)
[18] Anderlini L *et al.* (LHCb) 2023 *J. Phys. Conf. Ser.* **2438** 012130 (*Preprint* 2204.09947)
[19] Goodfellow I *et al.* 2014 NeurIPS'14 pp 2672–2680 (*Preprint* 1406.2661)
[20] Mirza M and Osindero S 2014 (*Preprint* 1411.1784)
[21] Graziani G *et al.* 2022 *JINST* **17** P02018 (*Preprint* 2110.10259)
[22] Mariani S *et al.* 2023 *J. Phys. Conf. Ser.* **2438** 012107
[23] He K *et al.* 2016 CVPR'16 pp 770–778 (*Preprint* 1512.03385)
[24] Terjék D 2020 ICLR'20 (*Preprint* 1907.05681)
[25] Aaij R *et al.* 2019 *EPJ Tech. Instrum.* **6** 1 (*Preprint* 1803.00824)
[26] Borisyak M and Kazeev N 2019 *JINST* **14** P08020 (*Preprint* 1905.11719)
[27] Anderlini L and Barbetti M 2022 *PoS* **CompTools2021** 034
[28] Barbetti M and Anderlini L 2023 ACAT'22 (*Preprint* 2301.05522)