ML-INFN meeting ▪ 29/05/2023

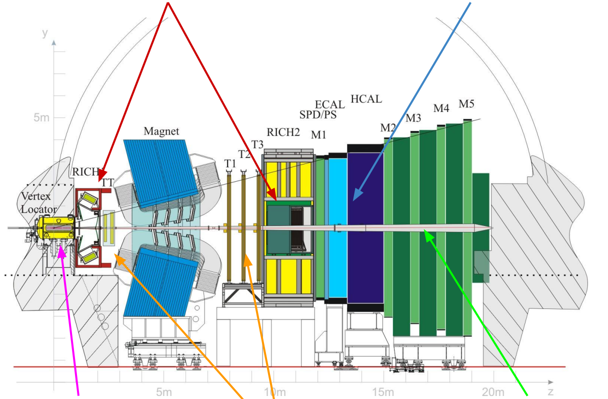
THE LAMARR FRAMEWORK

LHCb ultra-fast simulation based on deep generative models deployed within Gauss

Matteo Barbetti

on behalf of the LHCb Simulation Project

The LHCb experiment and its upgrades



The **LHCb detector** [1] is a single-arm forward spectrometer designed to study particles containing *b* and *c* quarks.

The **Upgrade I** of the LHCb experiment [2] is currently in commissioning. What's new?

- replacement of readout electronics
- new full software trigger system

The new detector will be able to collect datasets at least **one order of magnitude larger** thanks to an increased instantaneous luminosity (x5) and a more performant selection algorithm (x2).

fully software trigger system

x 5 instantaneous luminosity

x 2 selection efficiency

x 10 data sample size

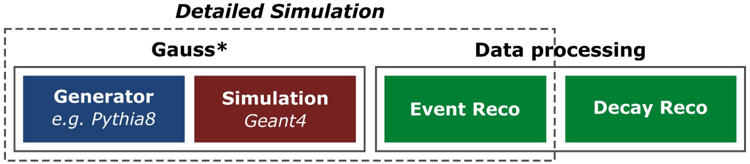
Simulating the LHCb experiment

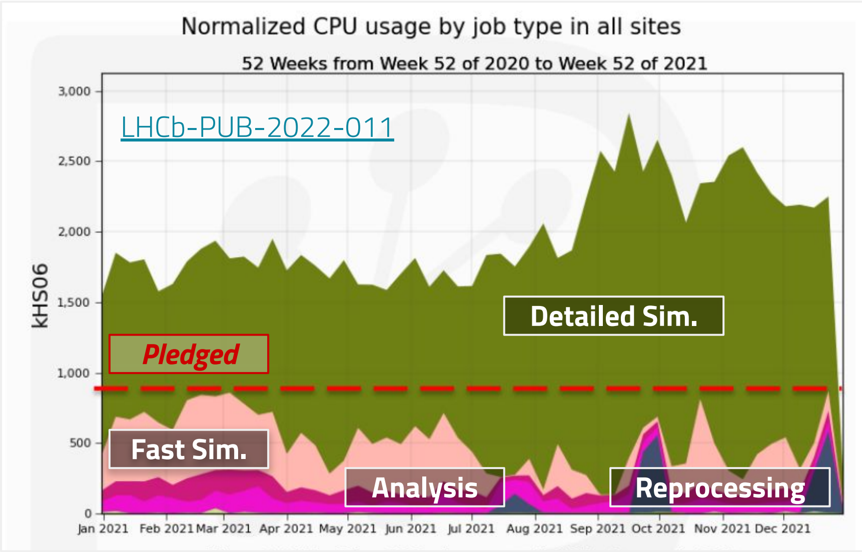
The standard for simulation at LHCb is ***Detailed Simulation***:

- simulation of all radiation-matter interactions
- simulated hits processed as real data
- extremely expensive in terms of CPU time (**more than 90%** used during LHC Run 2)
- unsustainable in the long term (*i.e.*, LHC Run 3 and those to come next)

Using Detailed Simulation only for LHC Run 3 needs will **far exceed the pledged resources** of LHCb.

Developing ***faster* simulation** strategies is mandatory to meet the upcoming and future requests for simulated data samples.





* **Gauss** is the LHCb simulation framework based on Gaudi [3]

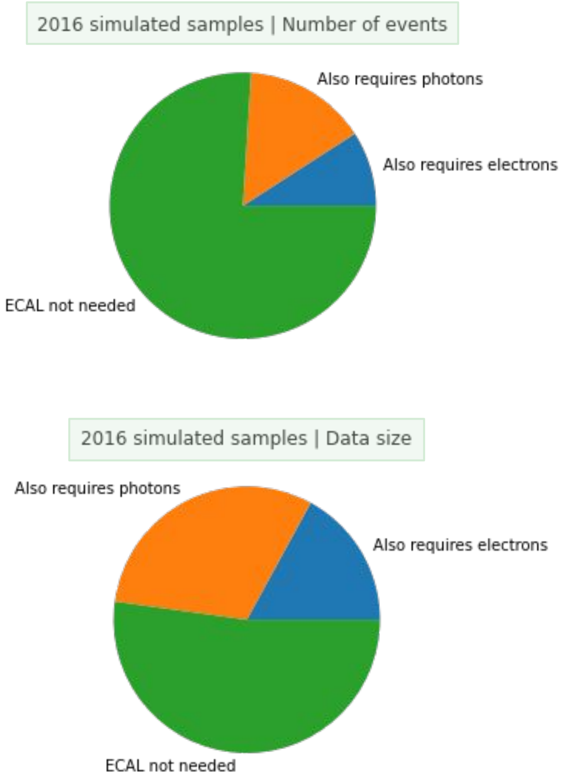
How does LHCb simulate events?

Simulations production driven by the LHCb physics program, *i.e.* most of the simulated decay modes are **heavy hadron decays**.

The detector will provide very "similar response" to, *e.g.*, a kaon from any source as long as with the same kinematics and detector conditions.

We could **save a lot of computing resources** by parameterizing the detector (low-level) response to that kaon and applying it to whatever decay model.

Analyses involving $h^{\pm}$ and $\mu^{\pm}$ only, often **drop** simulated raw detector information immediately → parameterizing directly the **high-level response** of the detector allows to save even more computing resources.



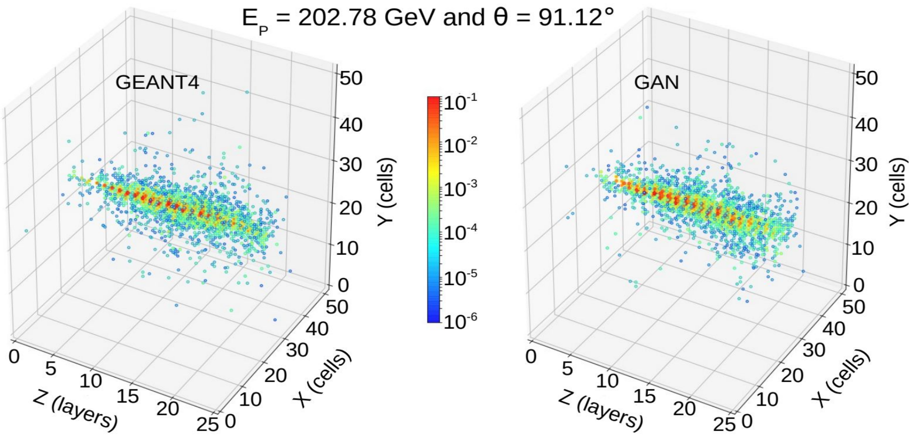
Machine learning for fast simulation

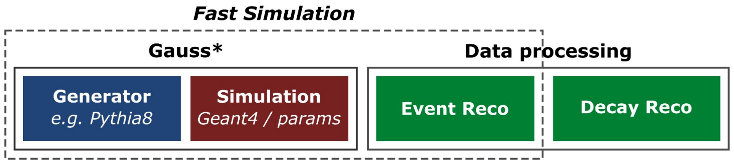
Machine learning models, such as ***generative models*** (*e.g.*, GAN, VAE, normalizing flows, diffusion models), can be trained to parameterize the detector response.

Generative Adversarial Nets (GAN) [4, 5] rely on the simultaneous training of two neural nets:

- *discriminator* → classification task
- *generator* → simulation task

GAN-based models can be effectively used to **replace the Geant4 simulation phase** of most of the HEP experiments [6, 7]. With these models the reconstruction step is the same as for real data (and detailed simulation).





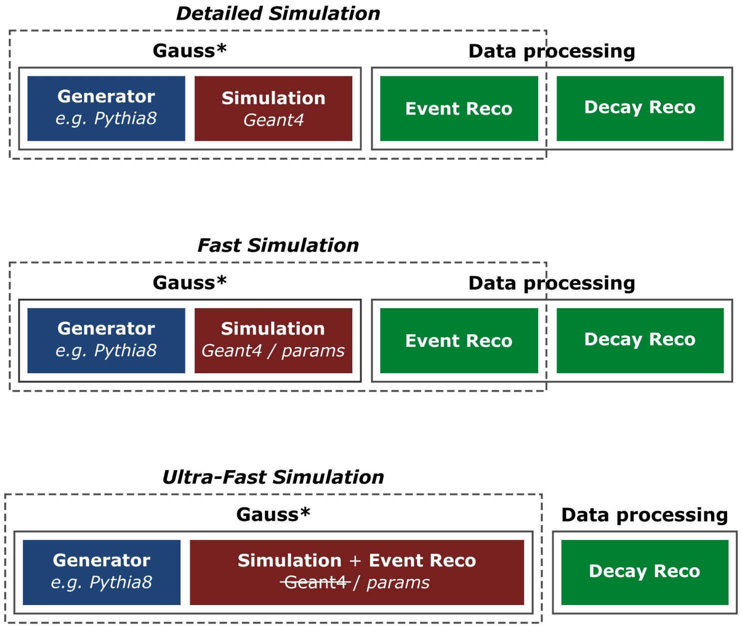
* **Gauss** is the LHCb simulation framework based on Gaudi [3]

Fast simulation VS. ultra-fast simulation

Fast Simulation techniques aim to speed up the Geant4-based simulation production:

- Simulation framework upgrade
- Reducing the detector geometry (*e.g.*, track-only sim)
- Reuse of the underlying events, **ReDecay** [8]
- Parameterizing **energy deposits** instead of relying on Geant4 (*e.g.*, shower libraries [9] or GANs [6, 7])

Ultra-Fast Simulation strategies replace Geant4 with parameterizations able to transform generator-level particles into analysis-level reconstructed objects [10].



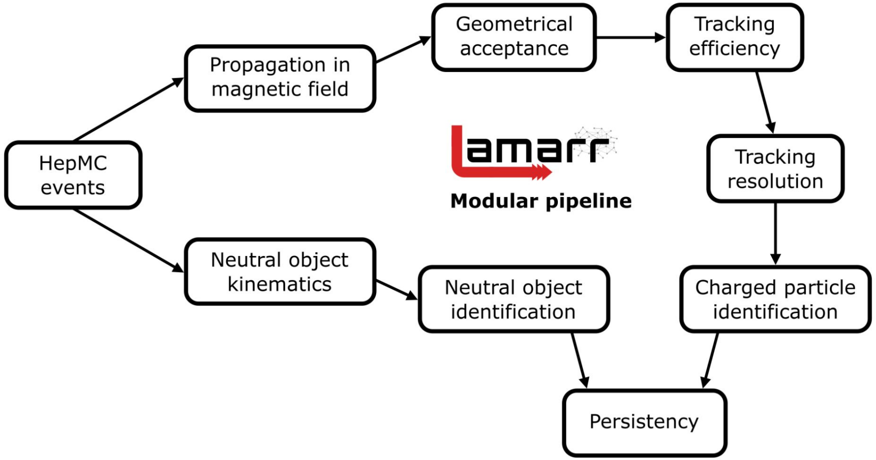
* **Gauss** is the LHCb simulation framework based on Gaudi [3]

Lamarr: the LHCb ultra-fast simulation option

Lamarr is the novel ultra-fast simulation framework of LHCb, able to offer the fastest options for simulation. Lamarr consists of a **pipeline of** (ML-based) **modular parameterizations** designed to replace both the simulation and reconstruction steps [11, 12].

Lamarr is integrated with the LHCb simulation framework:

- compatibility with all the **LHCb-tuned generators**
- compatible with the **distributed computing** middleware (LHCbDirac) and production environment
- able to provide datasets in the same format used for analysis

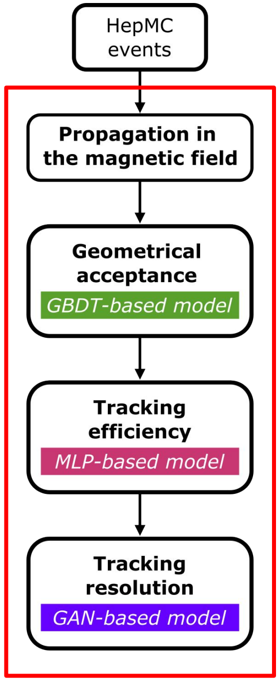


Tracking system models

Lamarr parameterizes the **LHCb Tracking system** mostly relying on a set of (ML-based) modules:

- **acceptance** → predict which of the generated tracks fall in the geometrical acceptance of the experiment
- **efficiency** → predict which of the generated tracks in acceptance are properly reconstructed by the detector
- **resolution** → convert the generator-level parameters of the reconstructed tracks into analysis-level ones, including track-quality features

A major effort is ongoing to model correctly the Tracking system in function of the **type of tracks**.



Lamarr Tracking pipeline

Geometrical acceptance

`model  :` **Gradient Boosted Decision Tree**

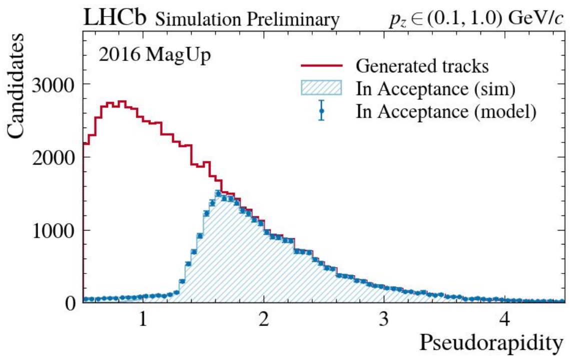
`loss   :` Binary Cross Entropy

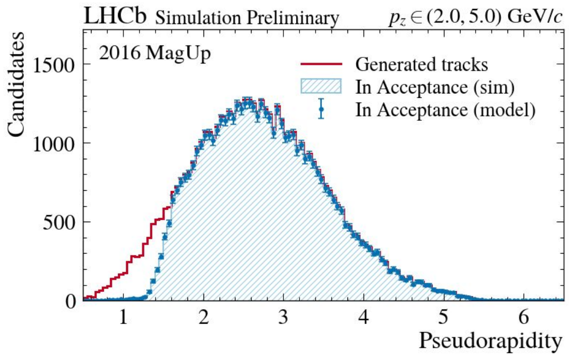
`input  :` generator-level position and slope of tracks

`output :` in acceptance [True , False]

Training performed on **Detailed Simulation**

The **probability** of having a track in acceptance that outputs from the GBDT model is used to weight the sample of generated tracks.





LHCb-FIGURE-2022-004

Tracking efficiency

`model` : **Multi-Layer Perceptron** (+ skip connections)

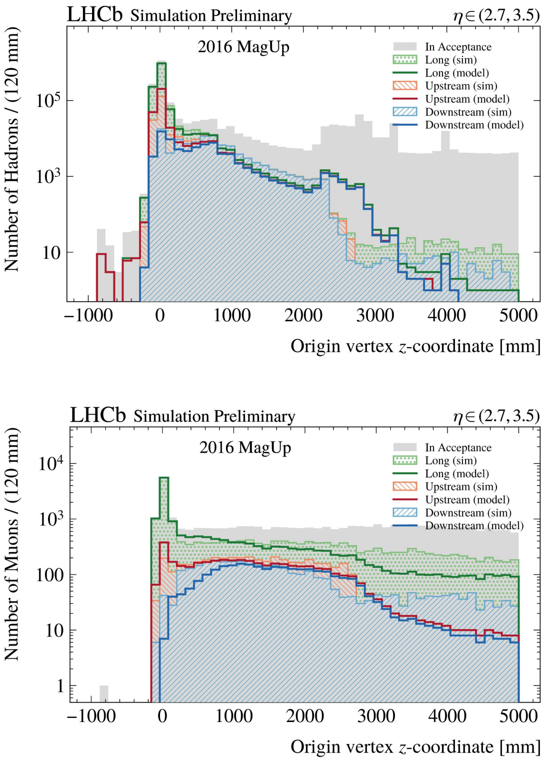
`loss`   : Categorical Cross Entropy

`input` : generator-level position, slope and momentum of tracks, eta, phi, particle species (*i.e.*, h, mu, e)

`output` : track classification as [*long* , *upstream* , *downstream* , *non-reconstructed*]

Training performed on **Detailed Simulation**

The **class probabilities** of reconstructing a track as long/upstream/downstream that outputs from the MLP model is used to weight the sample of generated tracks.



LHCb-FIGURE-2023-008

Tracking resolution

`model  :` **Generative Adversarial Networks**

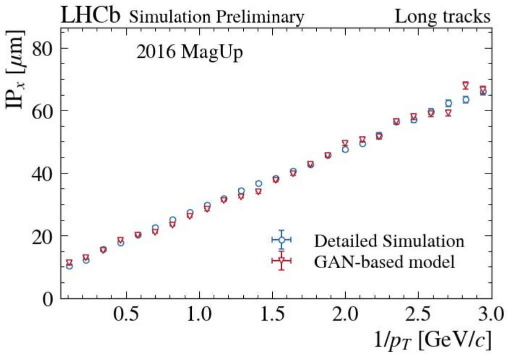
`loss   :` Binary Cross Entropy

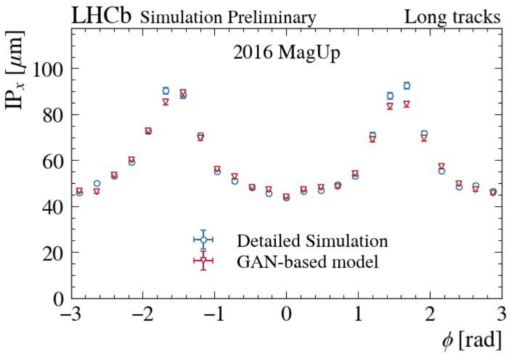
`input  :` generator-level position, slope and momentum of tracks

`output :` reconstructed tracks information

Training performed on **Detailed Simulation**

GAN-based model succeeds in parameterizing the x-projection of the *Impact Parameter* of tracks originated from the *Primary Vertex* even if **neither the transverse momentum nor the phi angle are used for training**.



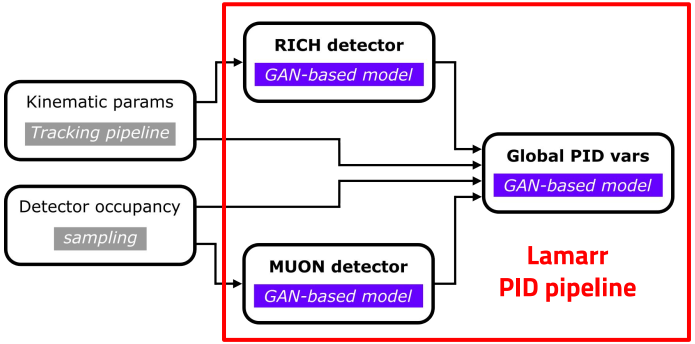


LHCb-FIGURE-2022-004

Particle identification system models

The high-level response of the **LHCb PID system** mostly relies on GAN-based models that can be trained on either detailed simulated samples or real data (more details here).

Lamarr provides RICH and MUON models for **muon**, **pion**, **kaon** and **proton** tracks based on the kinematics of the reconstructed tracks and a description of the detector occupancy.



This information alone aren't enough to parameterize the **Global PID variables**, that also need the response of the RICH and MUON systems. Hence, Lamarr provides the higher-level response of the PID system relying on a **stack of GAN-based models**.

RICH and MUON systems response

`model  :` **Generative Adversarial Networks**

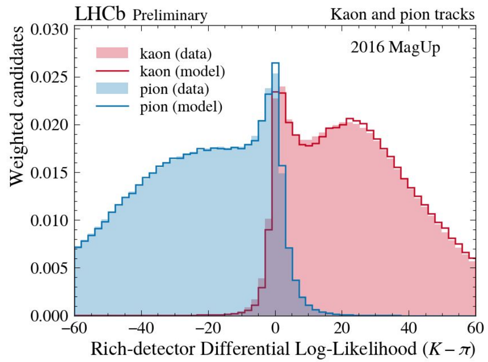
`loss   :` Wasserstein distance

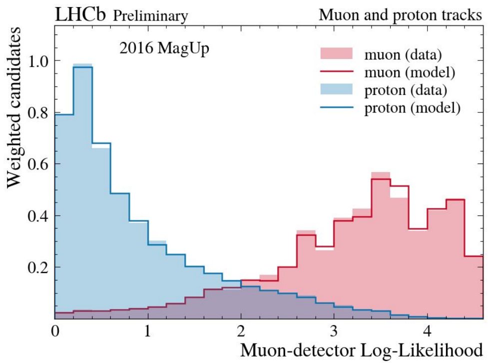
`input  :` analysis-level track kinematic parameters,
detector occupancy and particle species (*i.e.*, μ, π, K, p)

`output :` high-level response of the RICH detector
or the MUON system

Training performed on **Calibration Samples**

Lamarr provides one model per particle specie and per detector (x8). Training these parameterizations on real data needs for removing **any residual background sources**.





LHCb-FIGURE-2022-004

Higher-level PID response

`model  :` **Generative Adversarial Networks**

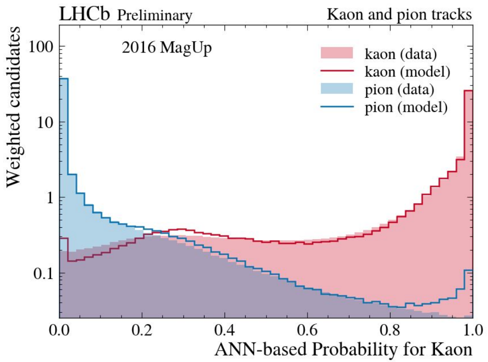
`loss   :` Wasserstein distance

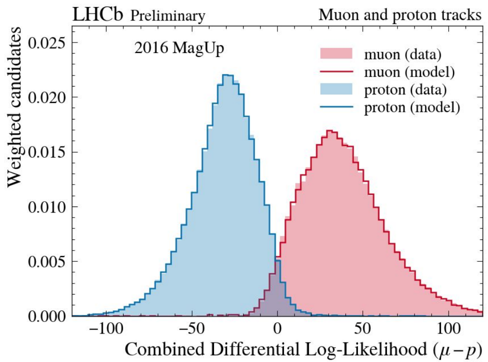
`input  :` analysis-level track kinematic parameters, detector occupancy, particle species (*i.e.*, μ, π, K, p), high-level response of the RICH detector and high-level response of the MUON system

`output :` global PID variables

Training performed on **Calibration Samples**

Lamarr provides one model per particle specie and per family of higher-level PID variables (x8).





LHCb-FIGURE-2022-004

Electromagnetic calorimeter model

Parameterization of the **LHCb calorimeter** available in Lamarr designed for detector studies → not suitable for simulation production

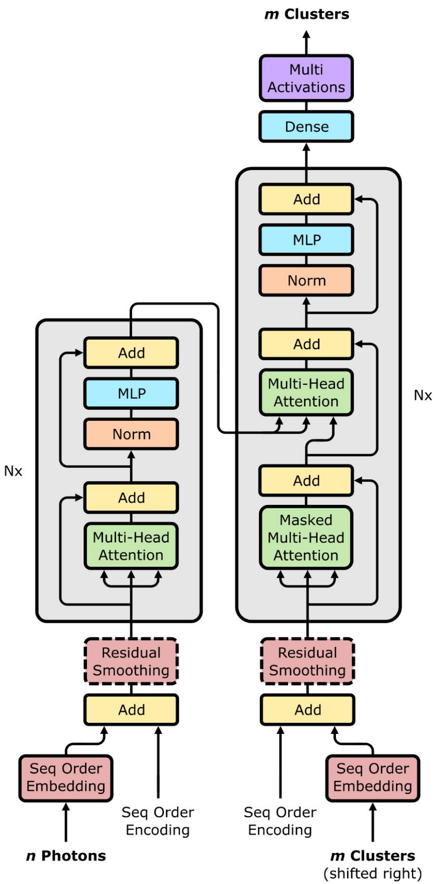
Improving the ECAL models is a necessary step if we want that Lamarr provides reliable parameterizations also for **photons** and **electrons**.

Simulating ECAL with an ultra-fast approach requires to face the **particle-to-particle correlation problem**:

- sequence of n generated photons → sequence of m reconstructed clusters (in general, with $n \neq m$)
- approached as a **translation problem**

Two strategies are currently under investigation:

- ***Graph Neural Networks*** (GNN) [13, 14]
- ***Transformer*** [15, 16]



GNN-based ECAL model

`model  :` **Graph Neural Networks** (heterogeneous graph)

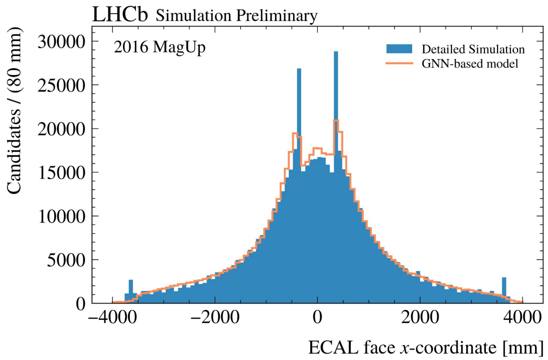
`loss   :` Weighted Mean Square Error + adversarial term

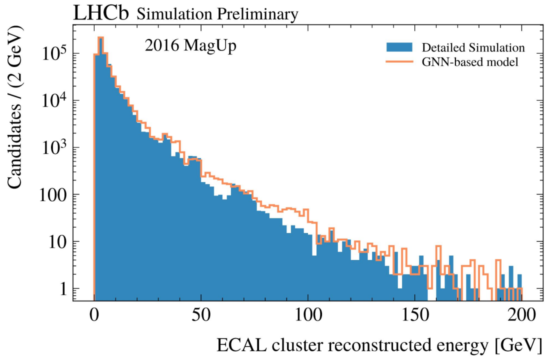
`input  :` position on ECAL face, slope, momentum and position of origin vertex of generated photons

`output :` position on ECAL face and total energy of reconstructed clusters

Training performed on Detailed Simulation

GNN-based model can process events with different number of photons/clusters by design (**no padding**). The loss function is weighted to enforce that *geometrically matched* clusters are correctly reproduced.





LHCb-FIGURE-2023-008

Transformer-based ECAL model

`model` : **Transformer** (encoder-decoder model)

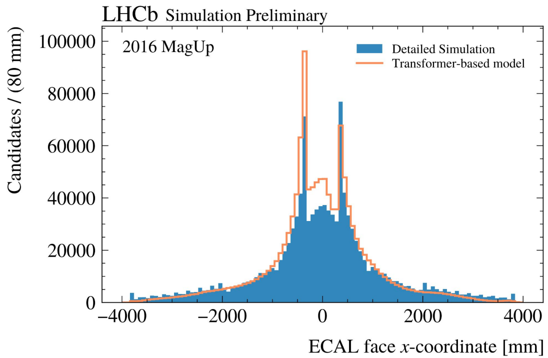
`loss` : Weighted Mean Square Error + adversarial term

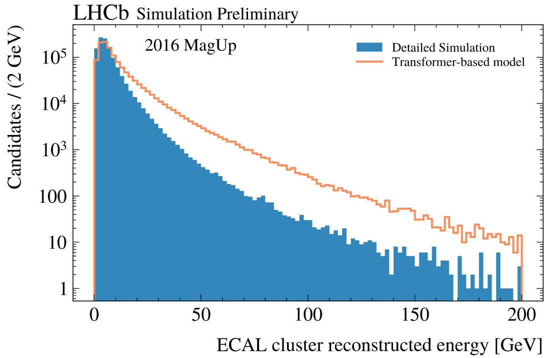
`input` : position on ECAL face, slope, momentum and position of origin vertex of generated photons

`output` : position on ECAL face and total energy of reconstructed clusters

Training performed on **Detailed Simulation**

To treat events with different number of photons/clusters, the Transformer needs a **padded training set**. The loss function is weighted to enforce that *geometrically matched* clusters are correctly reproduced.

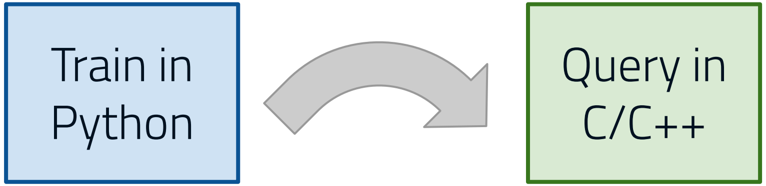




LHCb-FIGURE-2023-008

Deploying trained models in Gauss

Using trained ML models in C++ applications is wider and more general issue.



Several options for deployment exist, but come with some practical limitation. For example:

- Require **external dependencies** sometimes difficult to integrate in the build system of large HEP applications
- Geant4-based simulations are **hardly described** by ML typical computing graphs
- Introduce limits in the interplay between the **preprocessing** and **algorithmic** steps
- Often require **compiling with the framework** large part of the algorithm

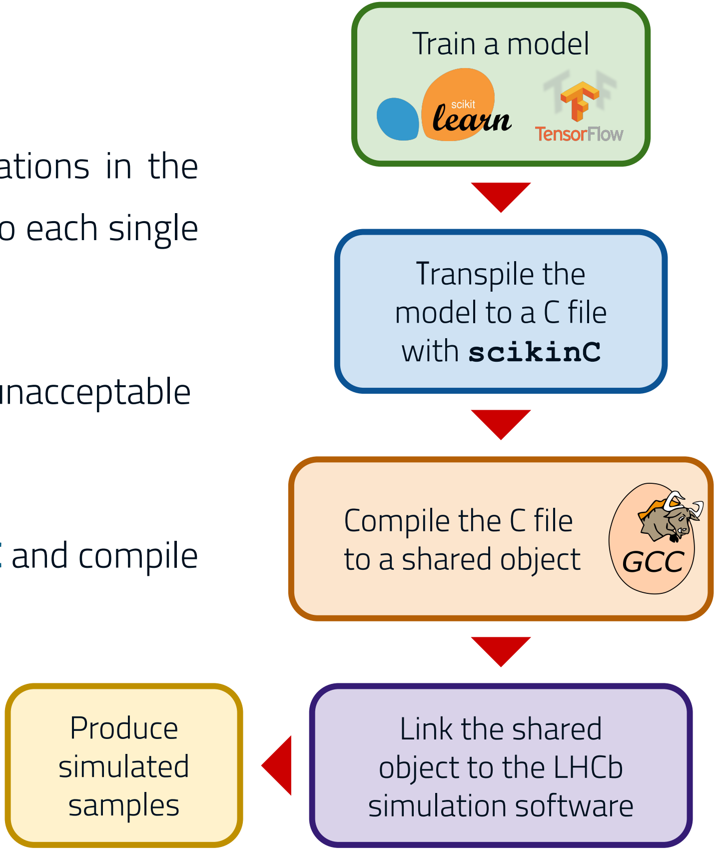
The transpiling approach

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.

Lamarr solution → we **transpile the trained models in C** and compile them to binaries, **dynamically linked** at runtime

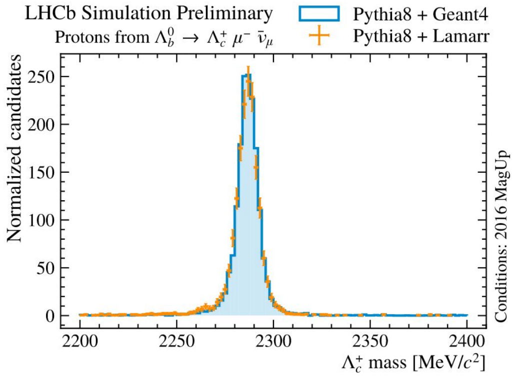
- LHCb tool: `scikinC` [17]
- Possible partial migration to: `keras2c` [18]

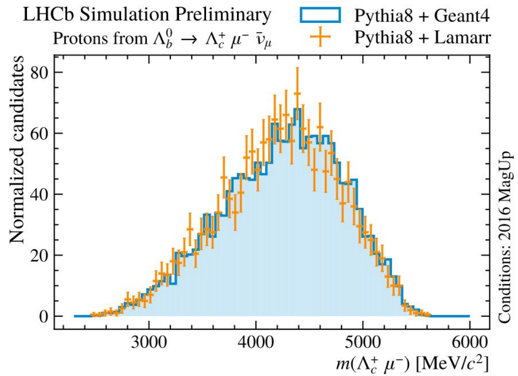


Lamarr validation campaign

Lamarr is currently under validation, comparing the distributions of the **analysis-level reconstructed quantities** parameterized with what obtained from Detailed Simulation.

- semileptonic decay mode: $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ with $\Lambda_c^+ \to pK^-\pi^+$
  - crucial the interface with LHCb-tuned generators
- **muons**, **pions**, **kaons** and **protons** in a single decay
  - all particle species for which Lamarr provides parameterizations
- Lamarr-based samples, detailed simulated samples and plots obtained from the **LHCb analysis software**
  - testing the integration with the current version of Gauss





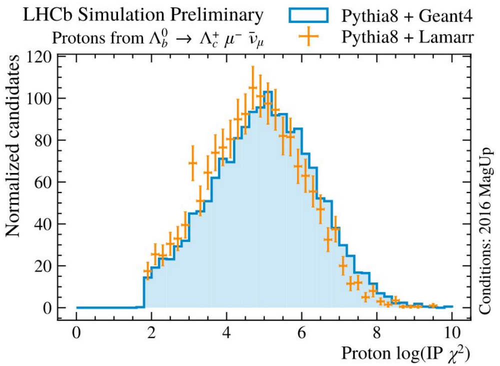
LHCb-FIGURE-2022-014

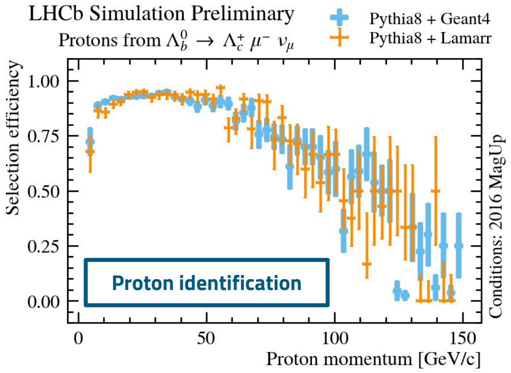
Some validation results

Smeared kinematic parameters of the generated particles are used to compute the **reconstructed masses**, the **impact parameters** and the **PID variables**.

Lamarr also provides information on *uncertainties* associated to the track reconstruction. For example, the **impact parameter χ^2** is a measure of inconsistency of a track trajectory with the PV.

Lamarr simulates the distribution of the detector response. But it's also crucial assessing that the **selection efficiencies** in function of the kinematic parameters and detector conditions for the parameterized quantities are well reproduced.





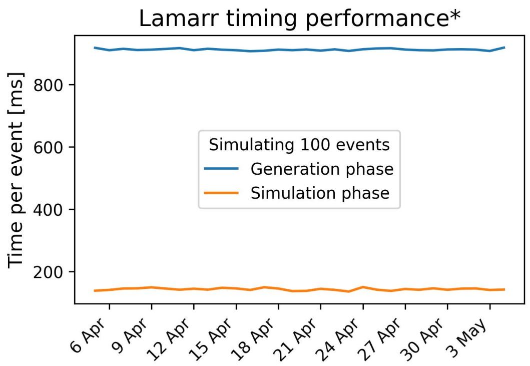
LHCb-FIGURE-2022-014

Preliminary timing studies

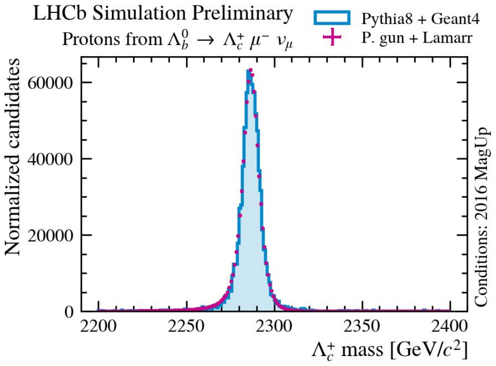
Comparing the normalized CPU spent for Geant4-based and Lamarr simulations of $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ decays we estimate a CPU reduction of **two-order-of-magnitude** for the Simulation phase.

Since the generation of *b*-baryons is exceptionally expensive, Pythia8 becomes the **major consumer of CPU** for simulation in the ultra-fast paradigm.

A **further speed-up** can be reached reducing the cost for generation, for example simulating only the signal particles (*i.e.* with ***Particle Guns***) and avoiding at all the simulation of the *pp* collisions, not needed since Lamarr models the detector occupancy.



* data obtained from the **LHCbPR** portal [19]

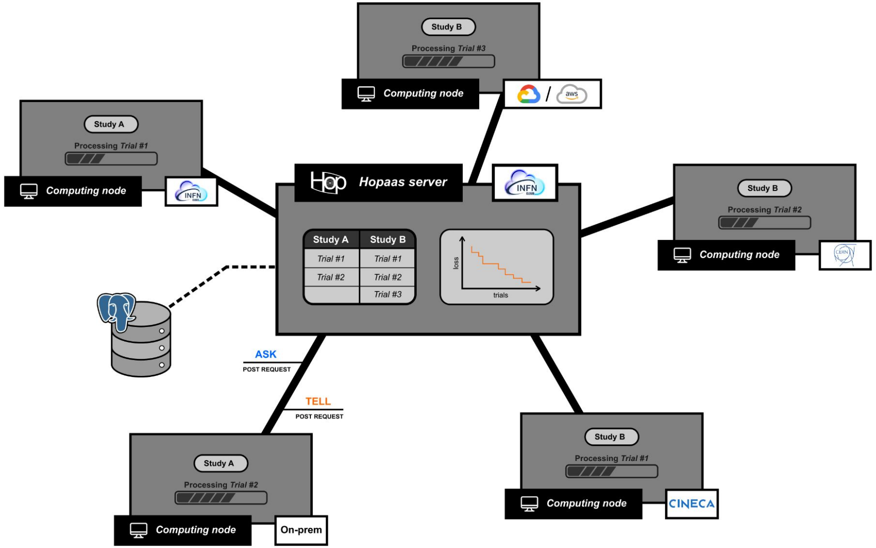


LHCb-FIGURE-2022-014

Distributed hyperparameter optimization

The quality of adversarial-driven models benefit from **massive hyperparameter optimization** (HPO) **campaigns**.

To enable using opportunistic resources we need a **centralized service for managing HPO campaigns**, independent of the resource provider [20].



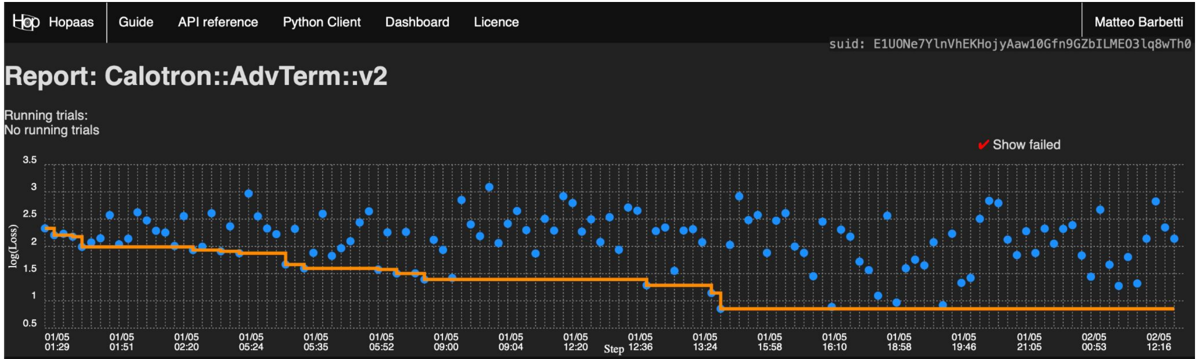
https://hopaas.cloud.infn.it

Web-based service hosted by INFN Cloud accessed via ***REST APIs*** *and* ***token authentication***

Optimization strategies and dashboard

Hopaas (*Hyperparameter OPtimization As A Service*) allows to orchestrate optimization studies powered by ***Bayesian techniques*** (more details here) across multiple computing instances.

Set up the training procedure and defined the **quality metric** (*e.g.*, BCE, KSD, EMD), the status of the optimization campaign can be monitored via the web dashboard provided by the service.



Trial id	N steps	Loss	Created	Stopped	Status	d_ds	alpha	t_ds	t_lr0	d_lr0
73	1	2.350	2023-05-01 15:23:19.156067+00:00	2023-05-01 15:23:19.305745+00:00	complete	70000	9.3e-01	150000	3.2e-04	1.0e-05
85	1	2.432	2023-05-01 16:22:29.780180+00:00	2023-05-01 16:22:29.921850+00:00	complete	70000	6.1e-01	150000	3.5e-04	1.1e-05
103	1	2.512	2023-05-01 19:37:29.566648+00:00	2023-05-01 19:37:29.741542+00:00	complete	20000	5.3e-01	170000	2.7e-04	3.4e-05
96	1	2.637	2023-05-01 19:22:05.748789+00:00	2023-05-01 19:22:05.898986+00:00	complete	200000	5.7e-01	150000	2.9e-04	1.6e-05
95	1	2.989	2023-05-01 18:48:15.552821+00:00	2023-05-01 18:48:15.725905+00:00	complete	80000	5.4e-01	150000	2.8e-04	1.0e-05

SUMMARY AND CONCLUSION

- The Lamarr framework offers to LHCb the **fastest option for simulation** needed to meet the upcoming and future requests for simulated samples
- Lamarr is integrated with the LHCb analysis framework and Lamarr-based simulation can be produced centrally using **LHC Grid resources**
- Great effort on improving the quality of the parameterizations (through massive optimization campaigns) and developing a new parameterization for the calorimeter able to face successfully the **particle-to-particle problem**

Lamarr will never replace the Geant4-based simulation, but may provide soon a precious tool to reduce the pressure on CPU of Detailed Simulation. Lamarr is designed to meet most of the needs for simulation of physics groups, from **designing selection** strategies, **training multivariate classifiers**, to **studying systematics** or **correlation effects**.

Flash advertising

The **Beyond Vision: Physics meets AI** workshop is organized in conjunction with the *22nd International Conference on Image Analysis and Processing* (ICIAP 2023).

Two main tracks:

1. Nuclear & other Physics-based Imaging technologies
2. Generative Models & other disruptive Deep Learning methods for Physical Sciences

THANKS!

Any questions or comments?

You can find me at: matteo.barbetti@fi.infn.it

ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

This work is partially supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU

References

1. LHCb Collaboration, A. Augusto Alves Jr *et al.*, JINST **3** (2008) S08005
2. LHCb Collaboration, R. Aaij *et al.*, arXiv:2305.10515
3. LHCb Collaboration, M. Clemencic *et al.*, J. Phys.: Conf. Ser. **331** (2011) 032023
4. I. J. Goodfellow *et al.*, arXiv:1406.2661
5. D. Teljék, arXiv:1907.05681
6. V. Chekalina *et al.*, EPJ Web Conf. **214** (2019) 02034
7. G. R. Khattak *et al.*, Eur. Phys. J. C **82** (2022) 386
8. D. Müller *et al.*, Eur. Phys. J. C **78** (2018) 1009
9. M. Rama and G. Vitali, EPJ Web Conf. **214** (2019) 02040
10. LHCb Collaboration, L. Anderlini, arXiv:2110.07925
11. L. Anderlini *et al.*, PoS **ICHEP2022** 233
12. M. Barbetti, arXiv:2303.11428
13. F. Scarselli *et al.*, IEEE Trans Neural Netw **20** (2009) 61
14. P. Velickovic *et al.*, arXiv:1710.10903
15. A. Vaswani *et al.*, arXiv:1706.03762
16. A. Dosovitskiy *et al.*, arXiv:2010.11929
17. L. Anderlini, M. Barbetti, PoS **CompTools2021** 034
18. R. Conlin *et al.*, J. Eng. App. AI **100** (2021) 104182
19. D. Popov, EPJ Web Conf. **214** (2019) 02043
20. M. Barbetti and L. Anderlini, arXiv:2301.05522

BACKUP

Long, upstream and downstream tracks

Upstream tracks

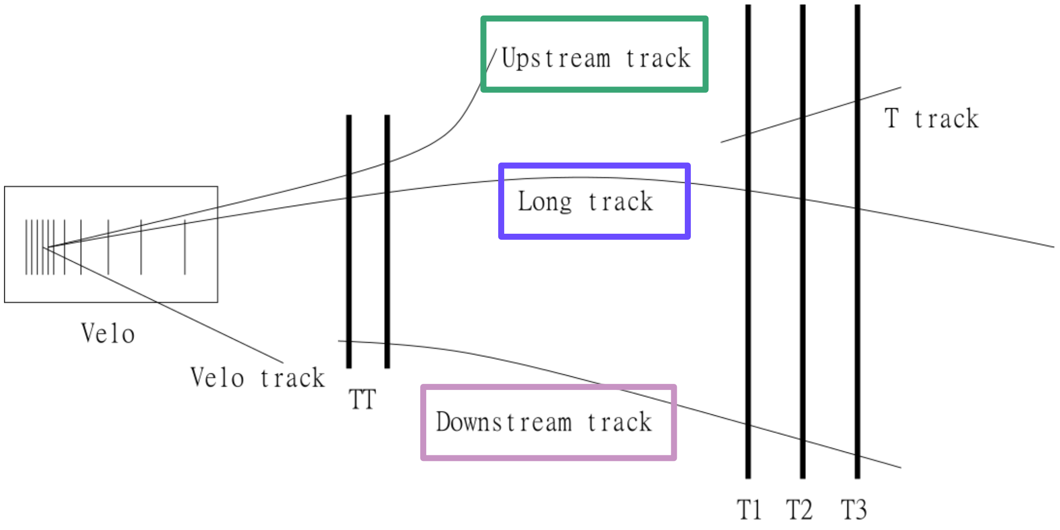
Tracks reconstructed by the tracking stations upstream the magnet.

Long tracks

Tracks reconstructed after having traversed the whole detector.

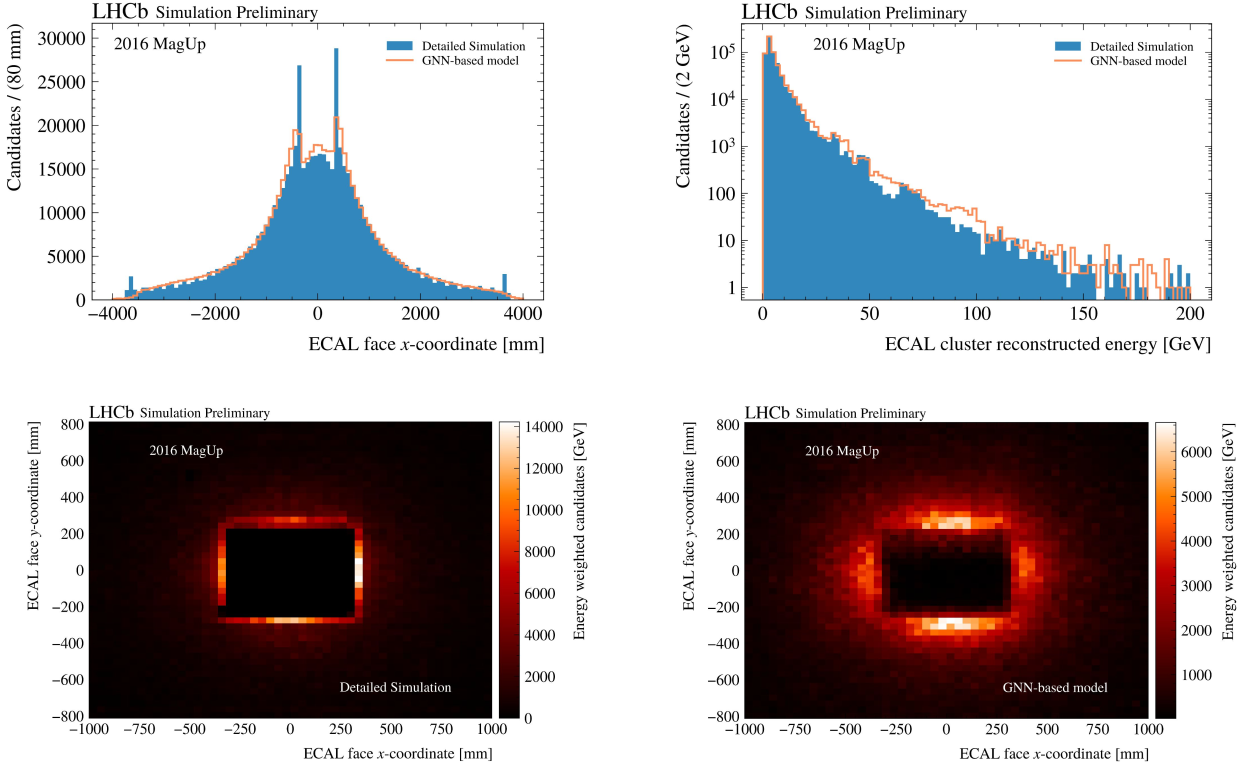
Downstream tracks

Tracks reconstructed by the tracking stations, but no hits found in the VELO.



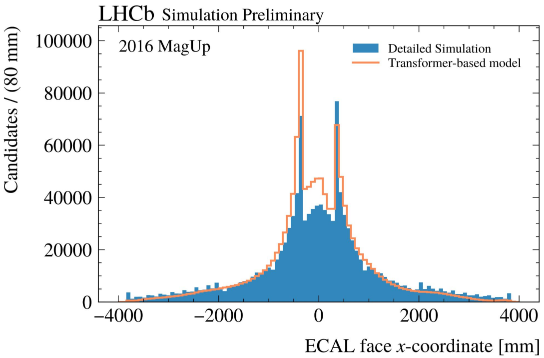
More GNN results for calorimeter

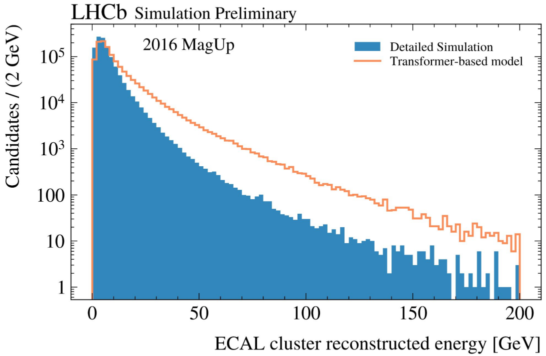
LHCb-FIGURE-2023-008

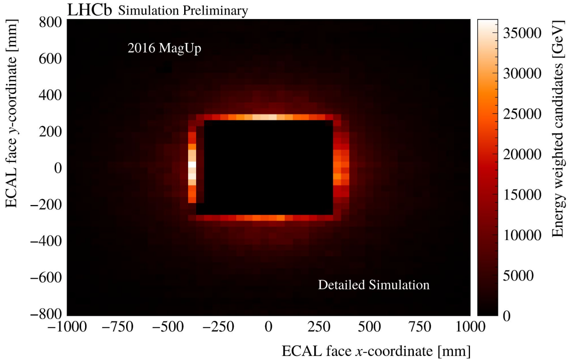


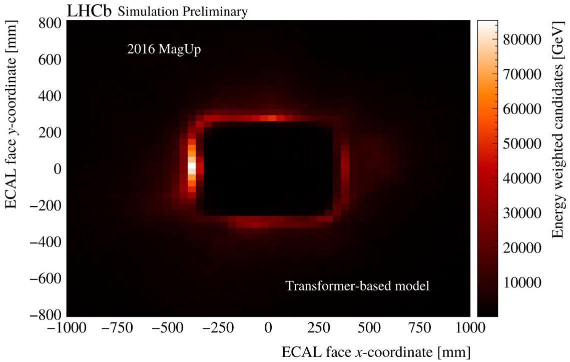
More Transformer results for calorimeter

LHCb-FIGURE-2023-008









Implementing Lamarr with the future Gauss

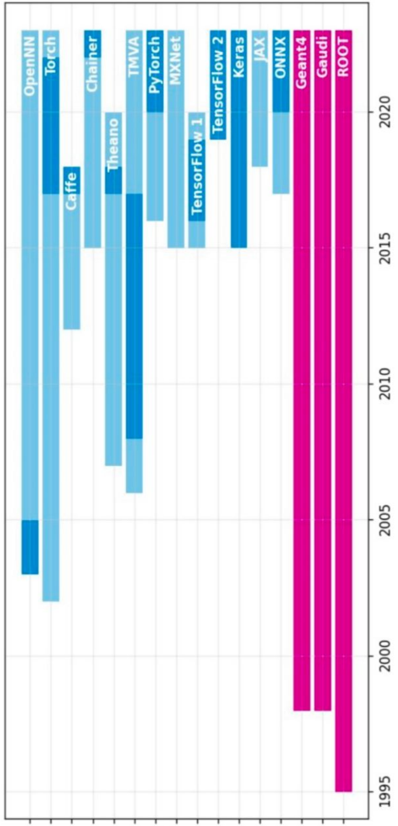
Developing an integration pipeline for ML-based models we should take account of:

- enable developments on **short time scales**
- put into production takes time (production quality studies)
- want to use in production for **many years**

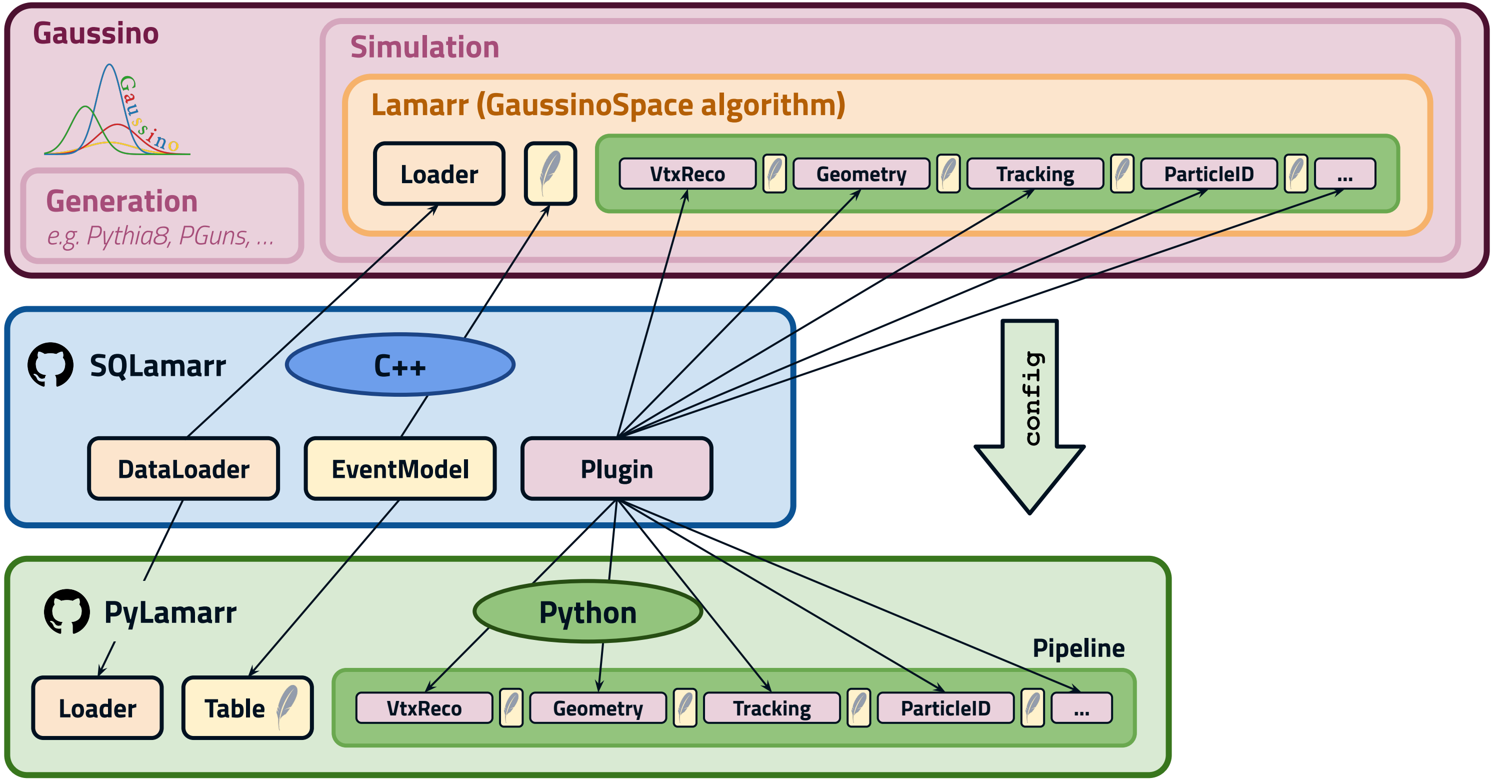
A solution is to deploy the trained models as C file (with a transpiling approach, *e.g.* scikinC [17]) to compile them to binaries, **dynamically linked** at runtime.

Integration of Lamarr with Gauss-on-Gaussino (more details here) is currently under development:

- **SQLamarr** (repo, docs) is a C++ library based on SQLite3 that defines classes and interfaces for loading data and managing parameterizations
  - **hard dependency policy** to be compiled within Gaussino (more details here)
  - stand-alone application provided
- **PyLamarr** (repo) is a pure-python project designed to configure pipelines
  - based on SQLamarr
  - pipelines can be executed in stand-alone mode



Using Lamarr within Gaussino



Hopaas: client-server system