Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

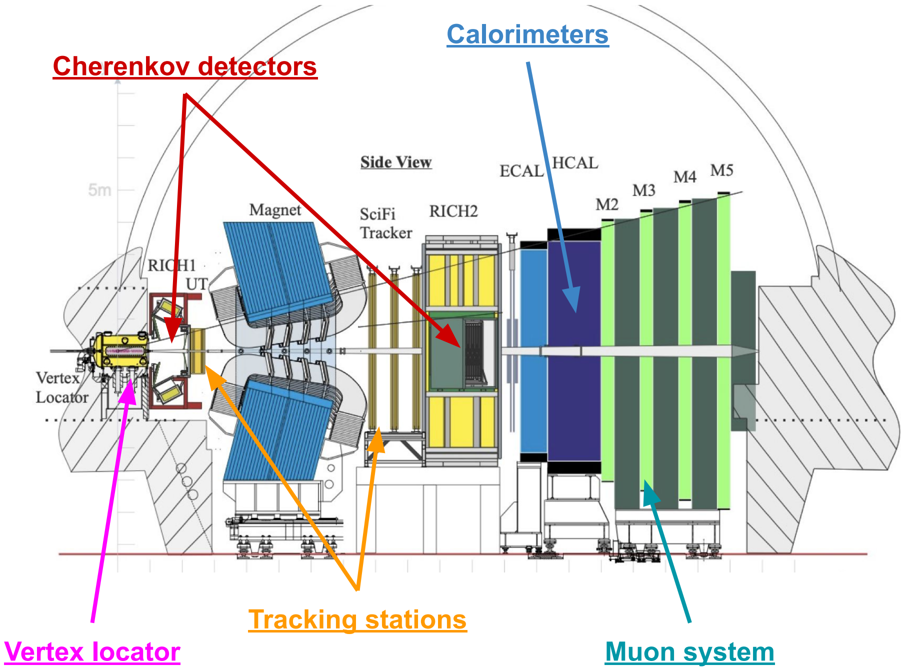
ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

Lamarr: implementing the flash-simulation paradigm at LHCb

M. Barbetti (INFN CNAF) on behalf of the LHCb Simulation Project

HSF Detector Simulation Working Group | 3 June 2024

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 finally complete. What's new?

- replacement of readout electronics
- new full software trigger system

fully software trigger system	**x 5** instantaneous luminosity	**x 2** selection efficiency

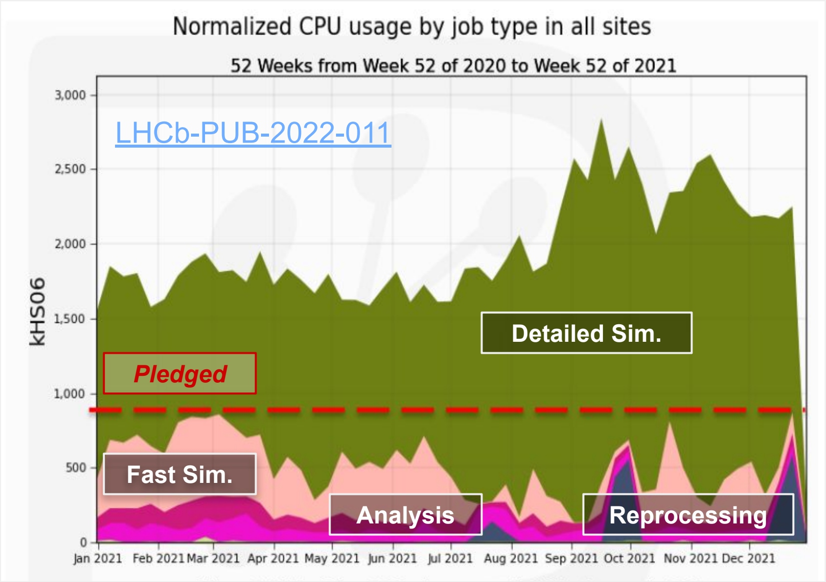
Computing requirements for simulation

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

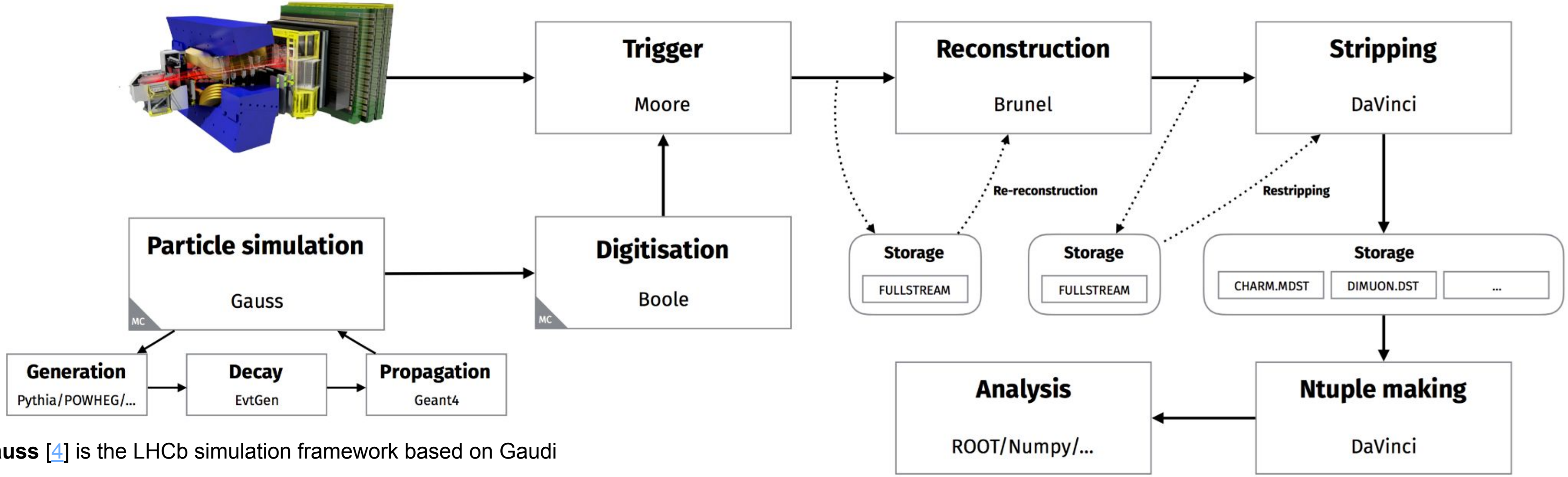
- simulation of all radiation-matter interactions
- simulated hits in detectors processed as real data
- **high CPU cost** (more than 90% [3] used during LHC Run 2)
- unsustainable in the long term (*i.e.*, LHC Run 3 and those to come next)

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.

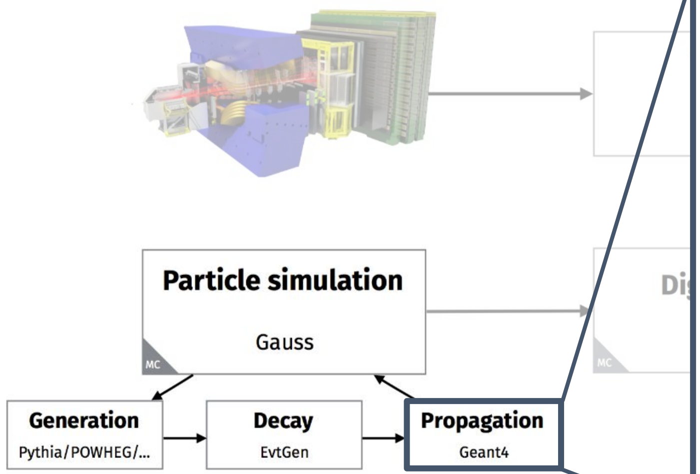


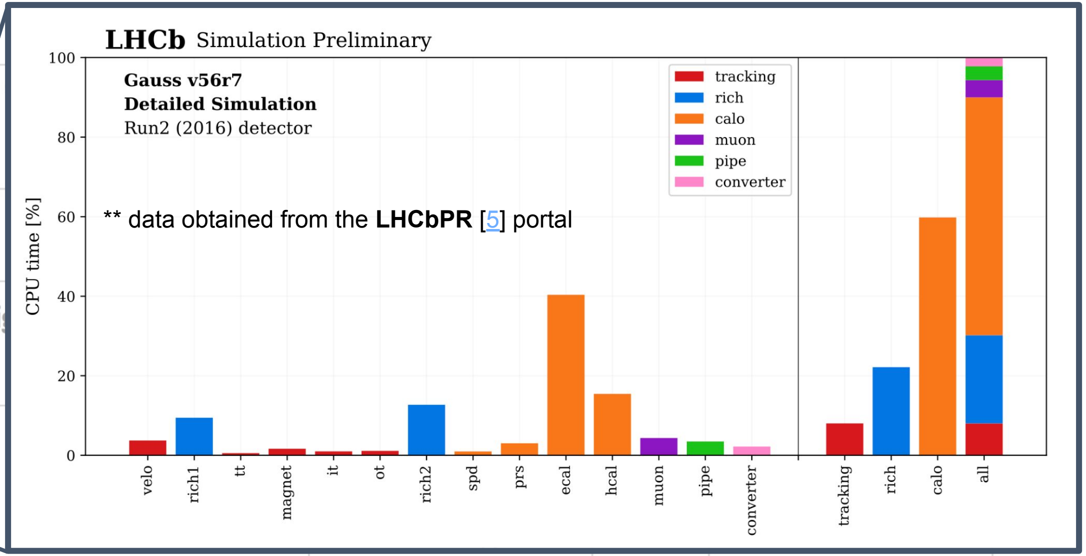
LHCb data processing



* **Gauss** [4] is the LHCb simulation framework based on Gaudi

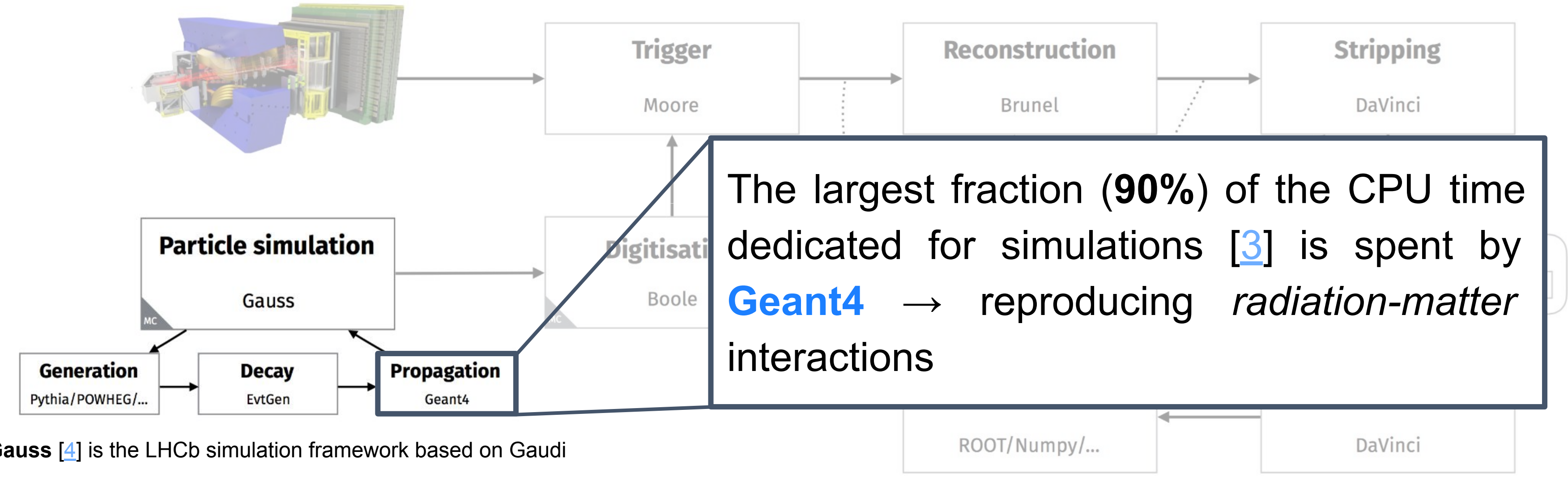
LHCb data processing





* **Gauss** [4] is the LHCb simulation framework based on Gaudi

LHCb data processing

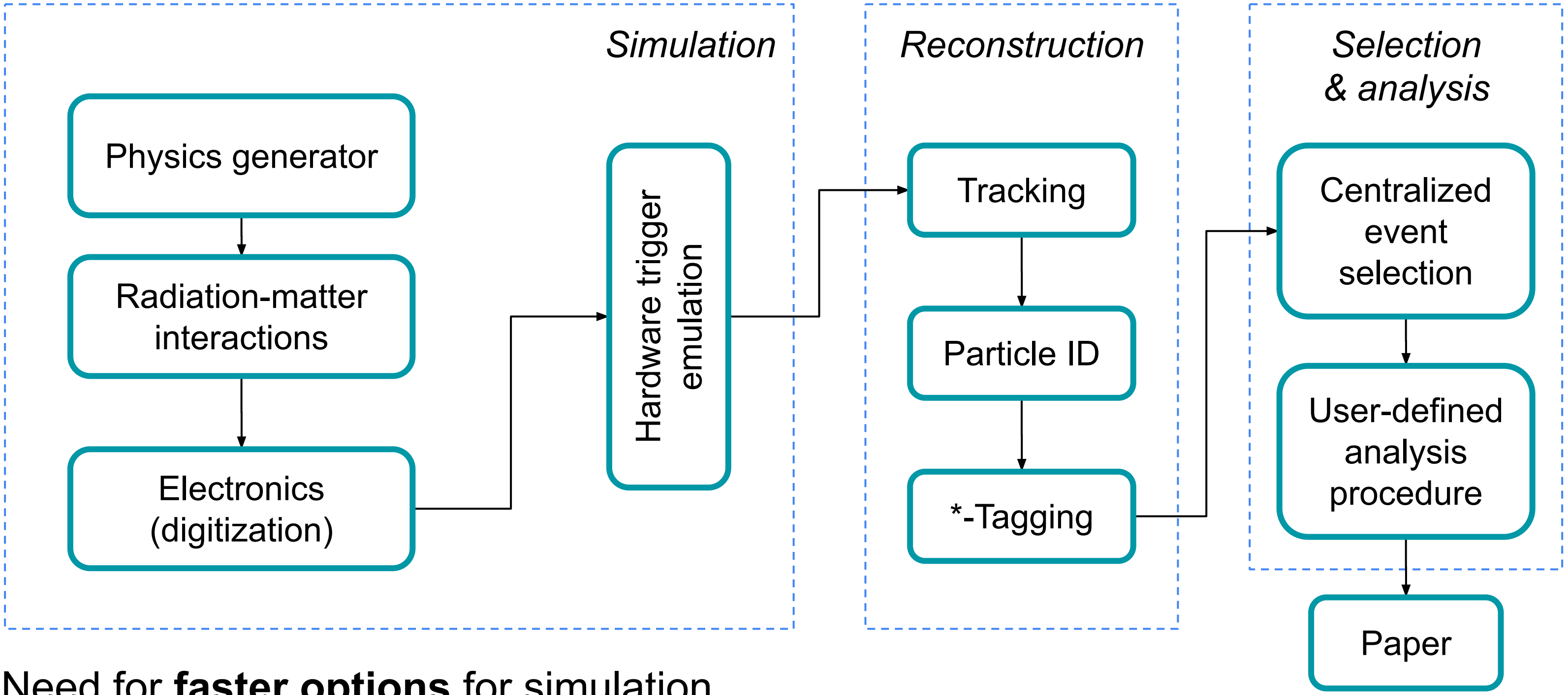


The largest fraction (**90%**) of the CPU time dedicated for simulations [3] is spent by **Geant4** → reproducing *radiation-matter* interactions

* **Gauss** [4] is the LHCb simulation framework based on Gaudi

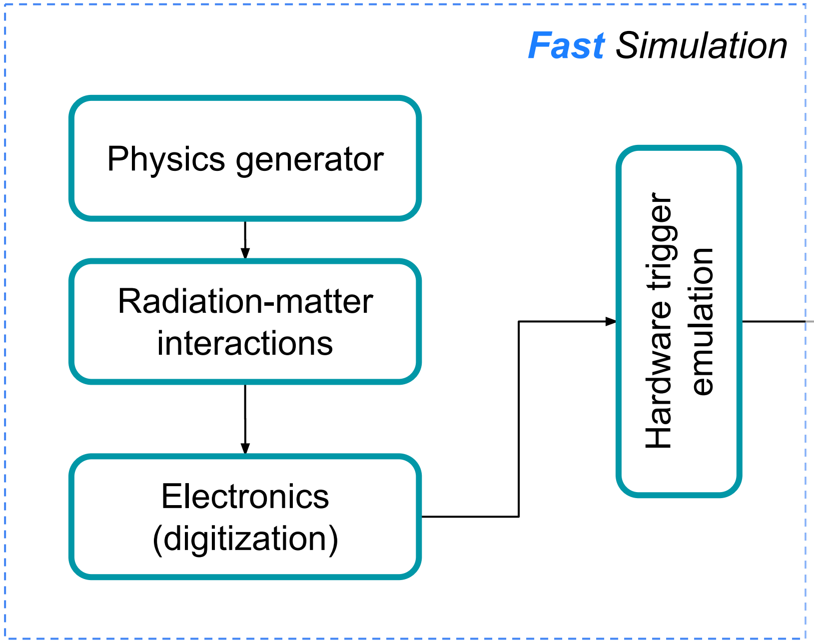
➪ Need for **faster options** for simulation

General data processing



➩ Need for **faster options** for simulation

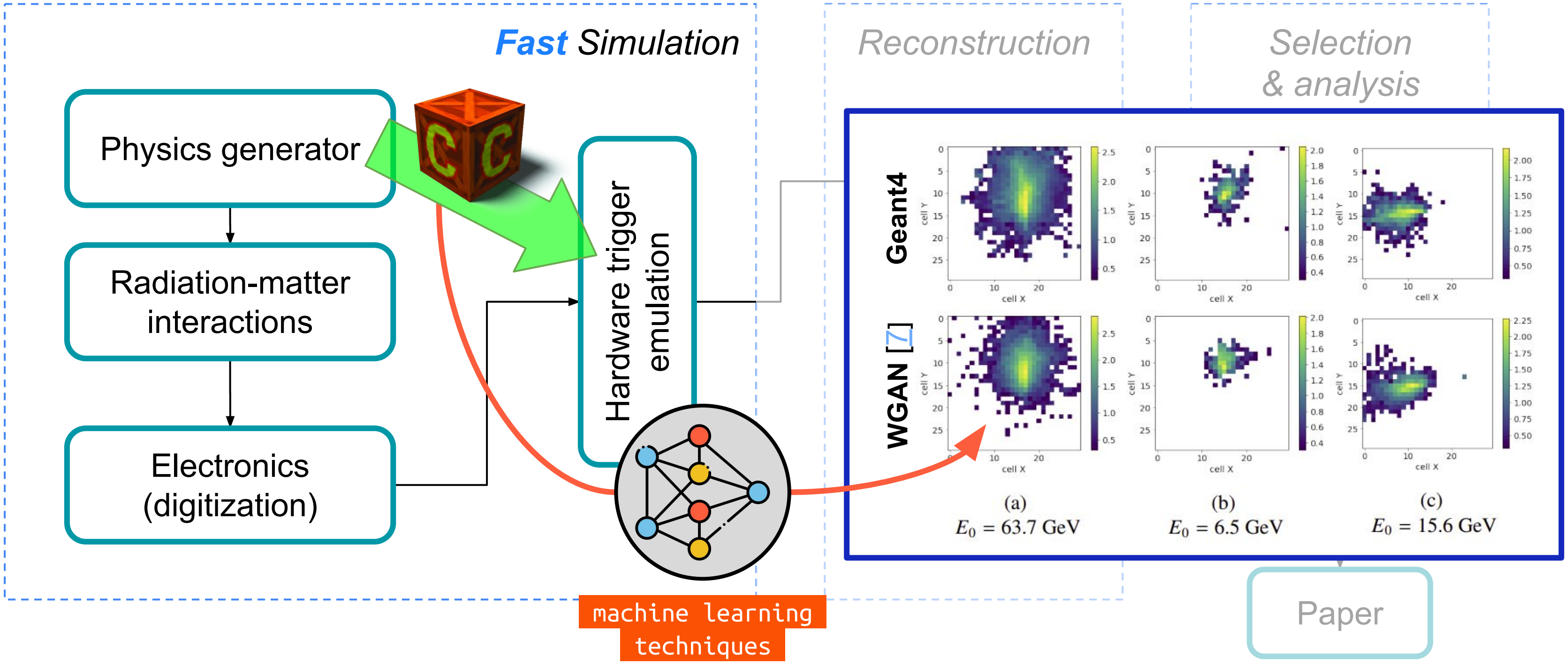
Fast detector simulation



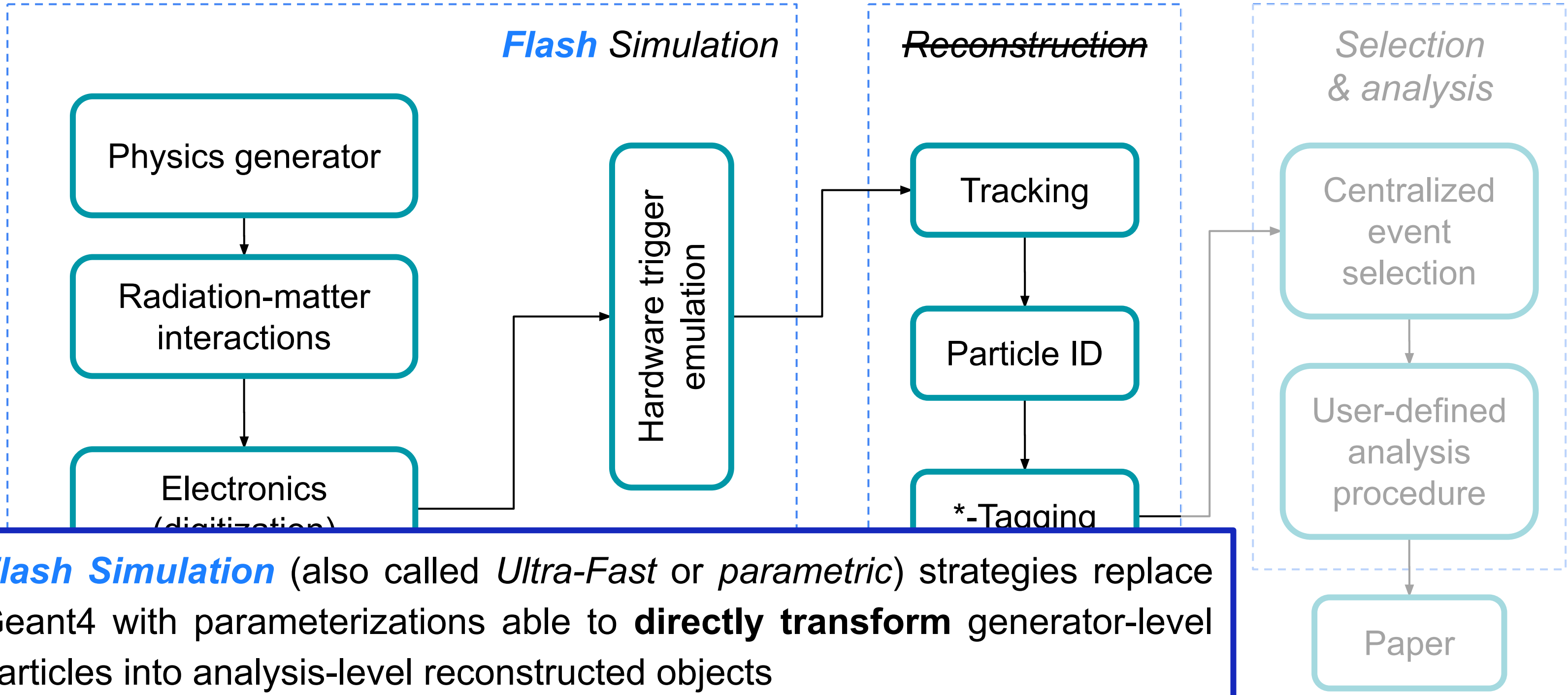
Fast Simulation techniques aim to speed up the Geant4-based simulation production:

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

Fast detector simulation

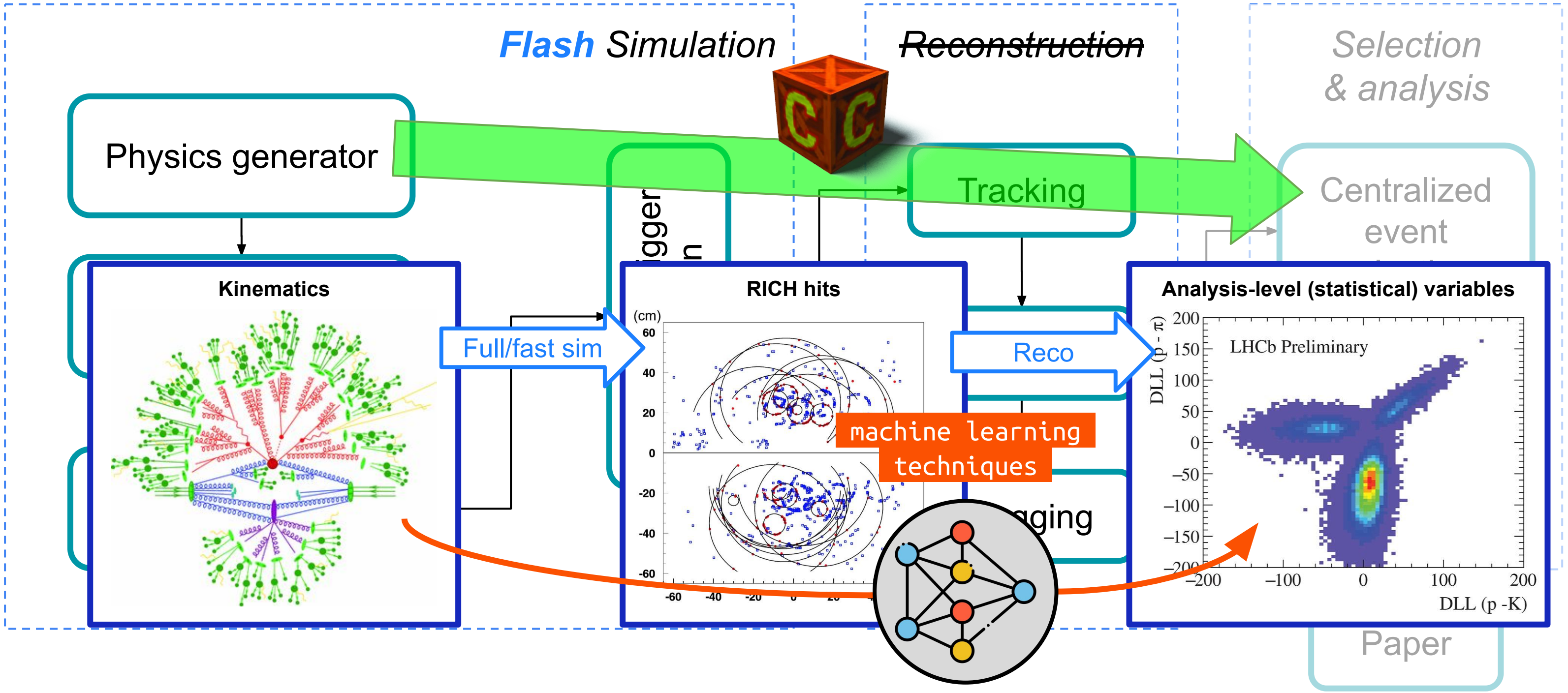


Flash detector simulation



Flash Simulation (also called *Ultra-Fast* or *parametric*) strategies replace Geant4 with parameterizations able to **directly transform** generator-level particles into analysis-level reconstructed objects

Flash detector simulation



Priorities for flash-simulated samples

The simulation production is driven by the LHCb physics program, *i.e.* **heavy hadron decays**

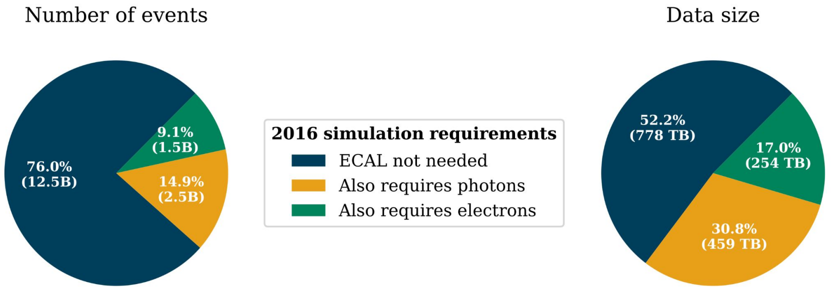
- most of the events (76%) don't require ECAL
- photons and electrons are less requested

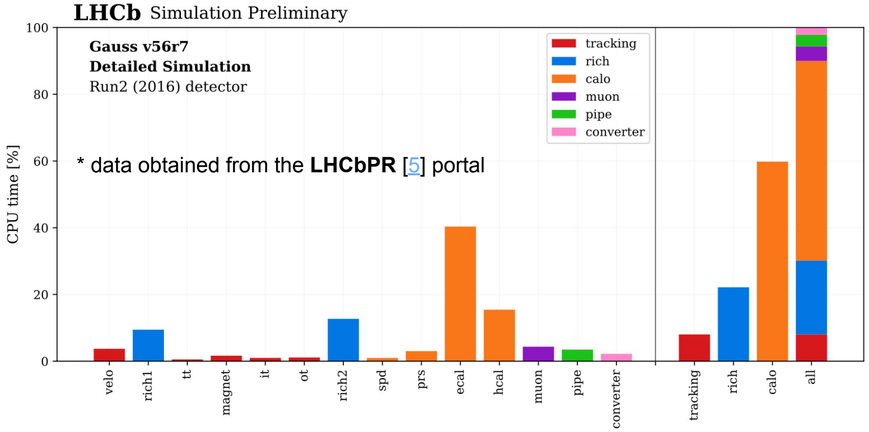
The simulation cost is driven by Geant4

- simulating secondary particles is expensive
- RICH and calorimeter systems dominate the cost
- parameterizing the detector response allows to **save a lot of computing resources**

Priorities for LHCb flash-simulation:

1. tracking + PID → most of charged particles (no electrons)
2. ECAL → photons
3. tracking + PID → **specialized treatment** for electrons

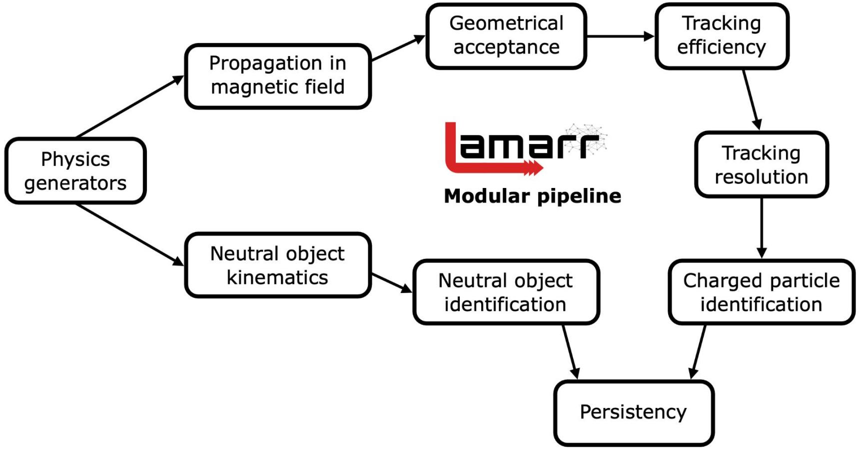




* data obtained from the **LHCbPR** [5] portal

Lamarr: the LHCb flash-simulation option

Lamarr [9-11] 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.

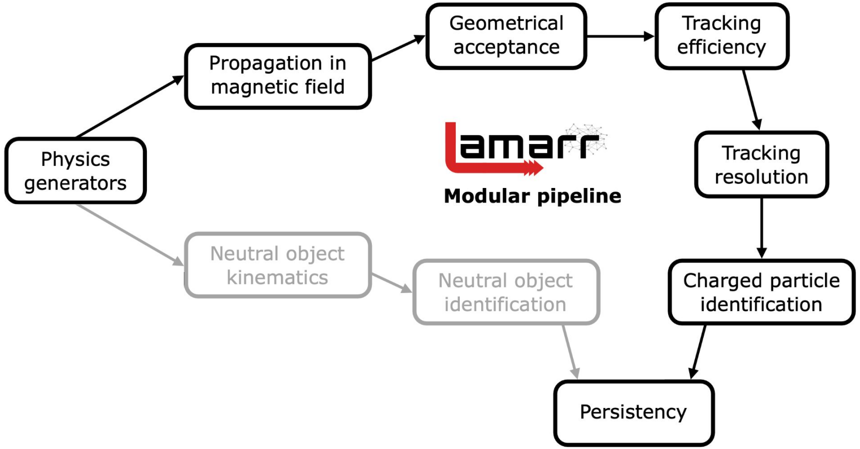


The Lamarr pipeline can be split in two branches:

1. a branch treating **charged particles** relying on tracking and particle identification (RICH + MUON + GPID) parameterizations
2. a branch facing the *particle-to-particle correlation* problem innate in the **neutral objects** (ECAL clusters) reconstruction

Lamarr: the LHCb flash-simulation option

Lamarr [9-11] 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.

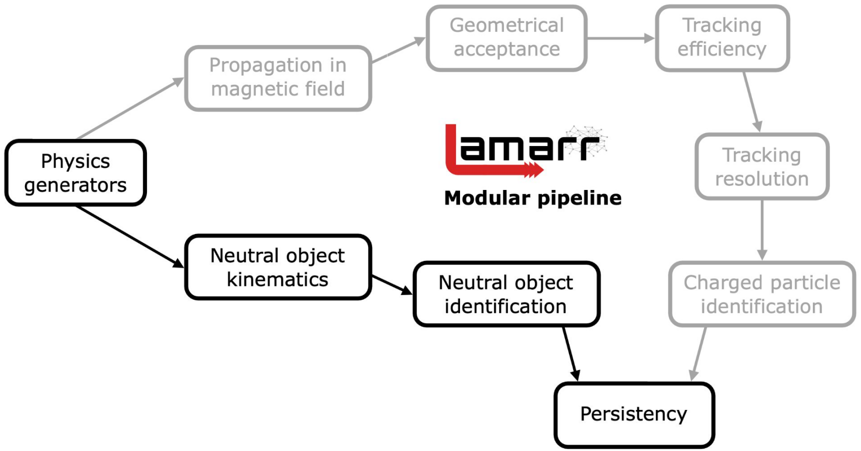


The Lamarr pipeline can be split in two branches:

1. a branch treating **charged particles** relying on tracking and particle identification (RICH + MUON + GPID) parameterizations
2. a branch facing the *particle-to-particle correlation* problem innate in the **neutral objects** (ECAL clusters) reconstruction

Lamarr: the LHCb flash-simulation option

Lamarr [9-11] 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.

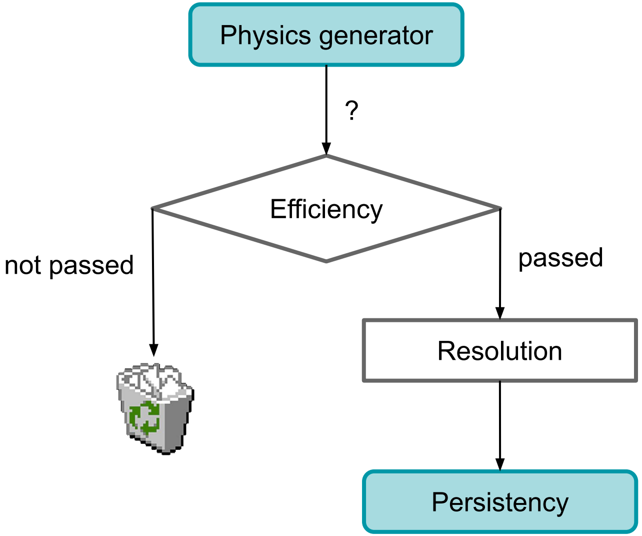


The Lamarr pipeline can be split in two branches:

1. a branch treating **charged particles** relying on tracking and particle identification (RICH + MUON + GPID) parameterizations
2. a branch facing the *particle-to-particle correlation* problem innate in the **neutral objects** (ECAL clusters) reconstruction

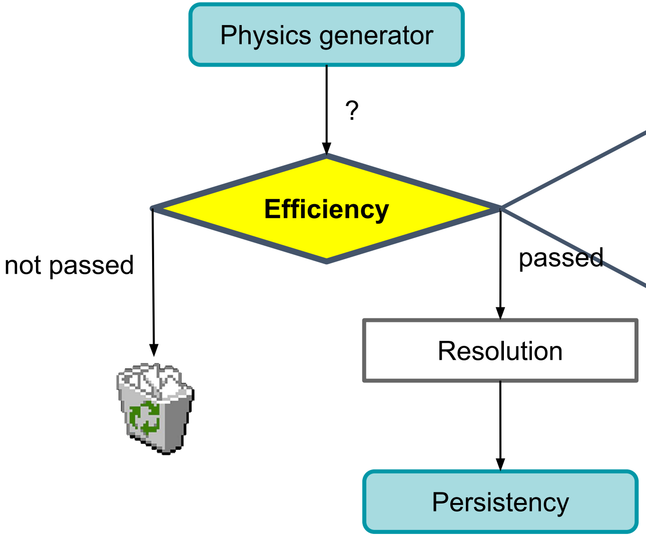
The k-to-k detector problem

Unambiguous *k*-to-*k* relation between generated particles and reconstructed objects → detector response modeled in terms of **efficiency** and **"resolution"** (e.g., high-level quantities)



The k-to-k detector problem

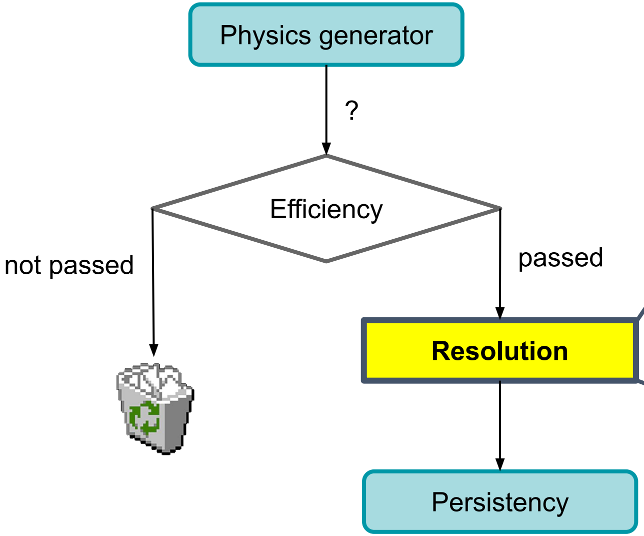
Unambiguous *k*-to-*k* relation between generated particles and reconstructed objects → detector response modeled in terms of **efficiency** and **"resolution"** (e.g., high-level quantities)



Deep Neural Networks (alternatively, *Gradient Boosted Decision Trees*) trained on detailed simulated samples to predict the fraction of **"good" candidates** (e.g., accepted, reconstructed, or selected) as a function of generator-level quantities.

The k-to-k detector problem

Unambiguous *k*-to-*k* relation between generated particles and reconstructed objects → detector response modeled in terms of **efficiency** and **"resolution"** (e.g., high-level quantities)



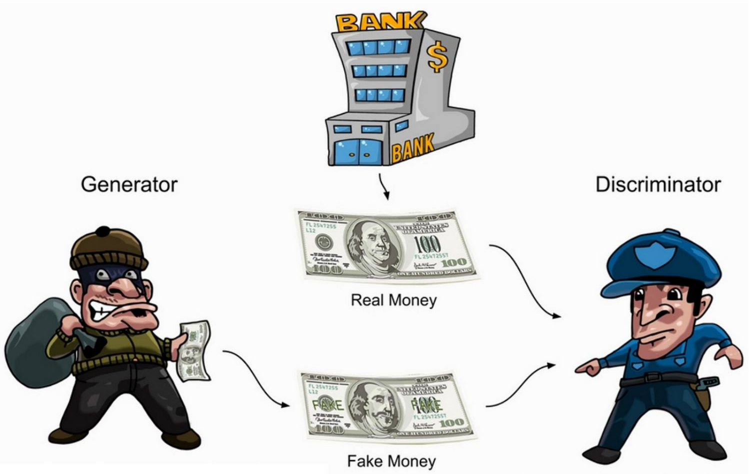
Conditional ***Generative Adversarial Networks*** (or *Deep Generative Models*, in general) trained on detailed simulated samples to parameterize the **high-level response** of LHCb detector (e.g., reconstruction errors, differential log-likelihoods, multivariate classifier output).

Generative Adversarial Networks

Generative Adversarial Networks (GAN) [12] rely on the simultaneous training of two neural nets:

- the **discriminator network** (D) is trained by a classification task to separate the generator output from the reference dataset;
- the **generator network** (G) is trained by a simulation task to reproduce the reference dataset trying to fake the discriminator.

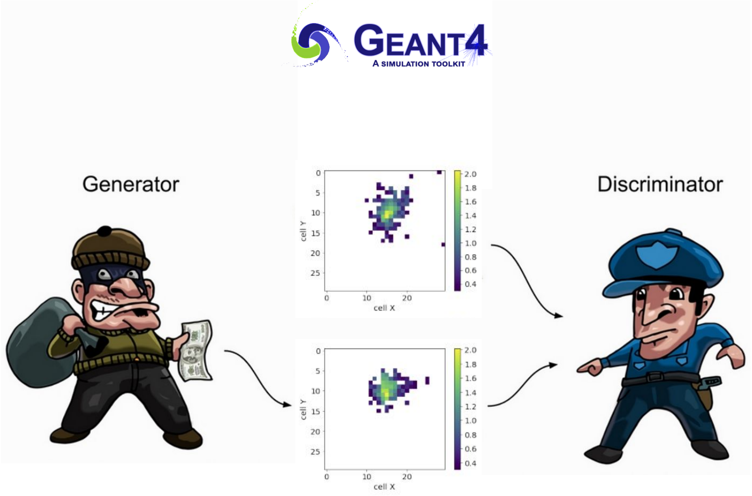
This framework corresponds to a **minimax two-players game**



Generative Adversarial Networks

Generative Adversarial Networks (GAN) [12] rely on the simultaneous training of two neural nets:

- the **discriminator network** (D) is trained by a classification task to separate the generator output from the reference dataset;
- the **generator network** (G) is trained by a simulation task to reproduce the reference dataset trying to fake the discriminator.

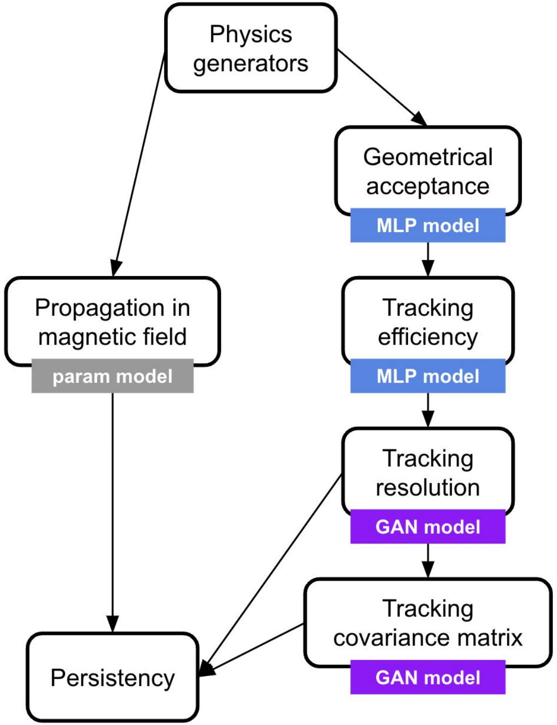


This framework corresponds to a **minimax two-players game**

Charged particle pipeline: the tracking system

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

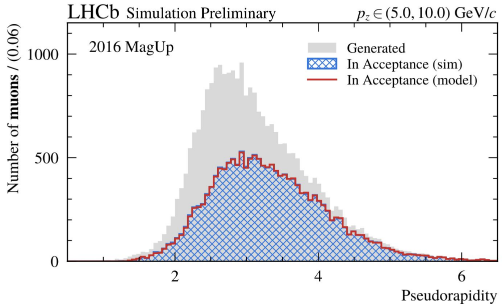
- **propagation** → approximates the trajectory of a charged particles through the dipole magnetic field
- **acceptance** → predicts which of the generated tracks lay within a sensitive area of the detector
- **efficiency** → predicts which of the generated tracks in acceptance are properly reconstructed by the detector
- **resolution** → parameterizes the errors introduced by the reconstruction algorithms to the track parameters
- **covariance** → parameterizes the uncertainties assessed by the Kalman filter procedure

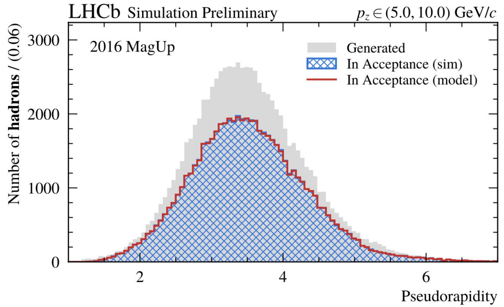


Charged particle pipeline: the tracking system

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

- **propagation** → approximates the trajectory of a charged particles through the dipole magnetic field
- **acceptance** → predicts which of the generated tracks lay within a sensitive area of the detector
- **efficiency** → predicts which of the generated tracks in acceptance are properly reconstructed by the detector
- **resolution** → parameterizes the errors introduced by the reconstruction algorithms to the track parameters
- **covariance** → parameterizes the uncertainties assessed by the Kalman filter procedure



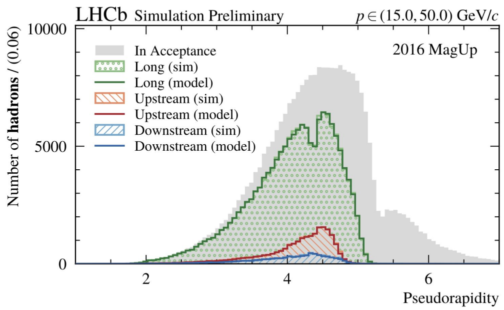


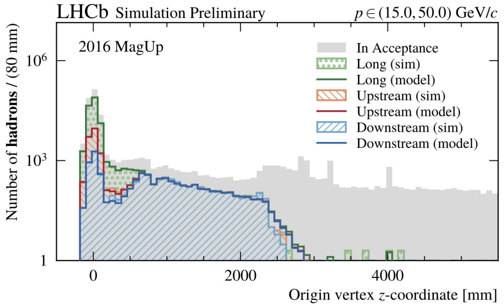
Efficiency (DNN model)

Charged particle 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 a charged particles through the dipole magnetic field
- **acceptance** → predicts which of the generated tracks lay within a sensitive area of the detector
- **efficiency** → predicts which of the generated tracks in acceptance are properly reconstructed by the detector
- **resolution** → parameterizes the errors introduced by the reconstruction algorithms to the track parameters
- **covariance** → parameterizes the uncertainties assessed by the Kalman filter procedure



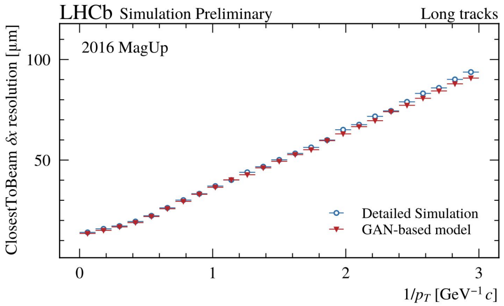


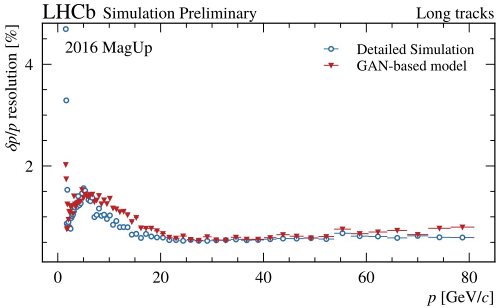
Efficiency (DNN model)

Charged particle 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 a charged particles through the dipole magnetic field
- **acceptance** → predicts which of the generated tracks lay within a sensitive area of the detector
- **efficiency** → predicts which of the generated tracks in acceptance are properly reconstructed by the detector
- **resolution** → parameterizes the errors introduced by the reconstruction algorithms to the track parameters
- **covariance** → parameterizes the uncertainties assessed by the Kalman filter procedure





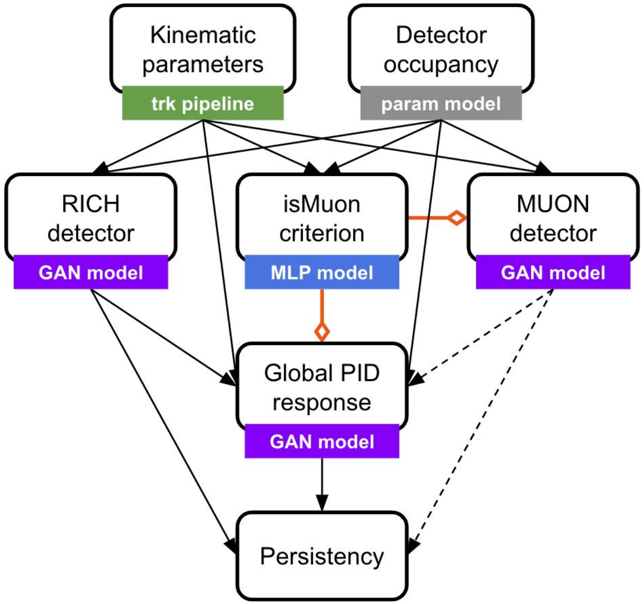
Resolution (GAN model)

Charged particle 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
- **MUON** → parameterizes likelihoods resulting from the MUON system
- **isMuon** → parameterizes the response of a FPGA-based criterion for muon loose boolean selection
- **Global PID** → parameterizes the global high-level response of the PID system, consisting of CombDLLs and ProbNNs

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

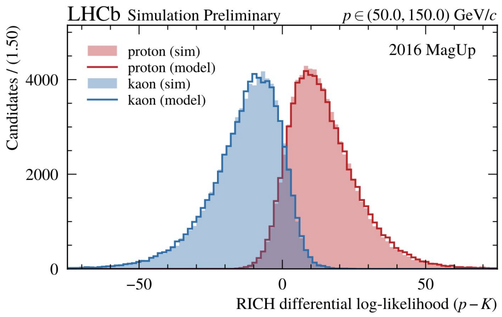


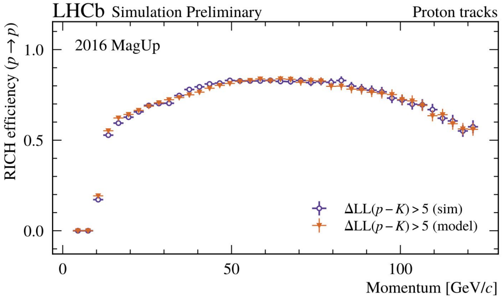
Charged particle pipeline: the PID system

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

- **RICH** → parameterizes DLLs resulting from the RICH detectors
- **MUON** → parameterizes likelihoods resulting from the MUON system
- **isMuon** → parameterizes the response of a FPGA-based criterion for muon loose boolean selection
- **Global PID** → parameterizes the global high-level response of the PID system, consisting of CombDLLs and ProbNNs

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





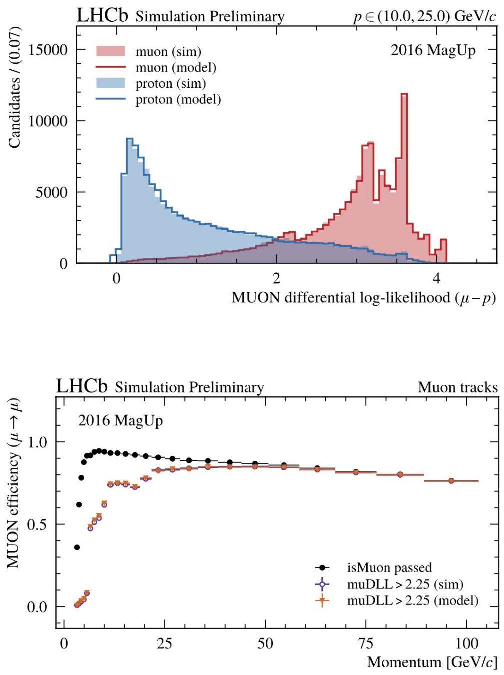
"Resolution" (GAN model)

Charged particle 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
- **MUON** → parameterizes likelihoods resulting from the MUON system
- **isMuon** → parameterizes the response of a FPGA-based criterion for muon loose boolean selection
- **Global PID** → parameterizes the global high-level response of the PID system, consisting of CombDLLs and ProbNNs

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



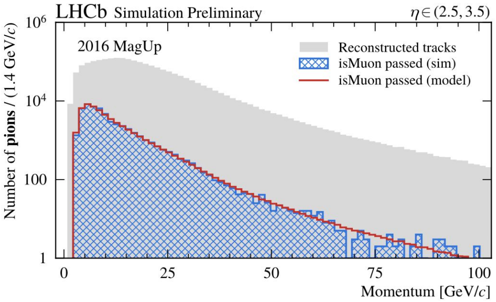
"Resolution" (GAN model)

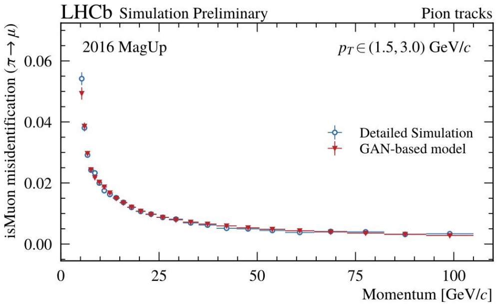
Charged particle pipeline: the PID system

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

- **RICH** → parameterizes DLLs resulting from the RICH detectors
- **MUON** → parameterizes likelihoods resulting from the MUON system
- **isMuon** → parameterizes the response of a FPGA-based criterion for muon loose boolean selection
- **Global PID** → parameterizes the global high-level response of the PID system, consisting of CombDLLs and ProbNNs

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





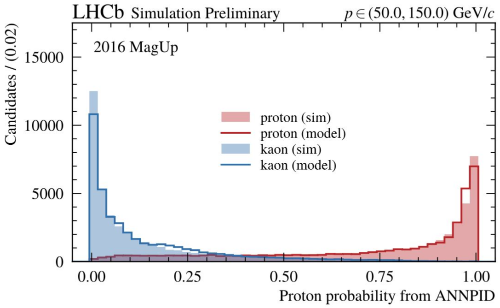
Efficiency (DNN model)

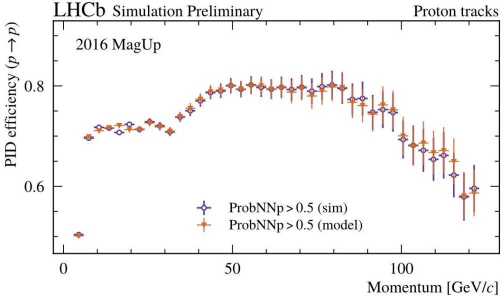
Charged particle pipeline: the PID system

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

- **RICH** → parameterizes DLLs resulting from the RICH detectors
- **MUON** → parameterizes likelihoods resulting from the MUON system
- **isMuon** → parameterizes the response of a FPGA-based criterion for muon loose boolean selection
- **Global PID** → parameterizes the global high-level response of the PID system, consisting of CombDLLs and ProbNNs

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



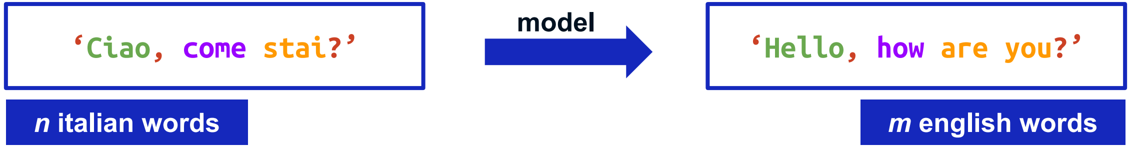


"Resolution" (GAN model)

The n-to-m detector problem

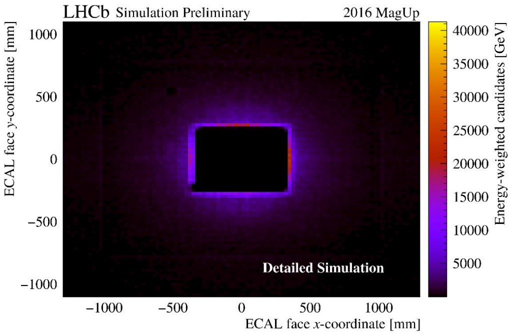
The flash-simulation of the ECAL detector is a non-trivial task:

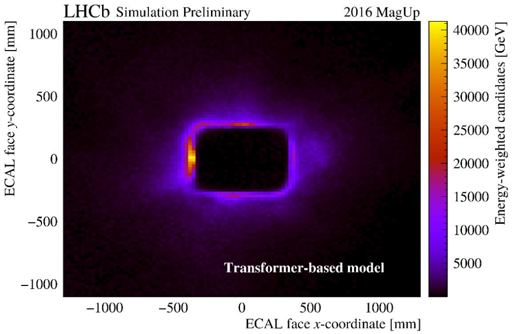
- **bremsstrahlung radiation**, **converted photons**, or **merged π^0** may lead to have *n* generated particles responsible for *m* reconstructed objects (in general, with $n \neq m$)
- the ***particle-to-particle correlation problem*** limits the validity of strategies used for modeling the unambiguous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



In our case, the source (italian) sentence is a **sequence of *n* generated photons** and the target (english) sentence is the corresponding **sequence of *m* reconstructed clusters**.

Neutral particles pipeline: the ECAL detector





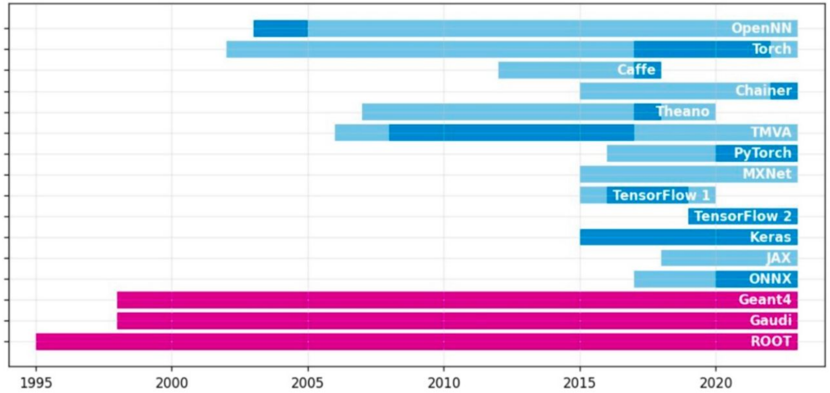
Two different approaches are currently under investigation:

- **signal photons** → for photons included in the decay modes under study
  - *k*-to-*k* condition enforced via **geometrical and energetic matching**
  - ECAL response described in terms of efficiency and resolution
- **seq2seq approach** → for photons produced by secondary process
  - ECAL event-level description inspired by translation problems
  - data sorted by energy → ***Transformer*** [13] + discriminator
  - graph topology → ***Graph Neural Network*** [14] + discriminator

Integration with the LHCb software stack

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 analysis



Most of the Lamarr parameterizations are ML-based:

- need for a **fast development cycle** (new architectures or training strategies easily outperform predecessors)
- AI community **extremely versatile** in terms of software technologies (no decades tradition of HEP community)

Models deployment → **transcompilation approach** [15,16]

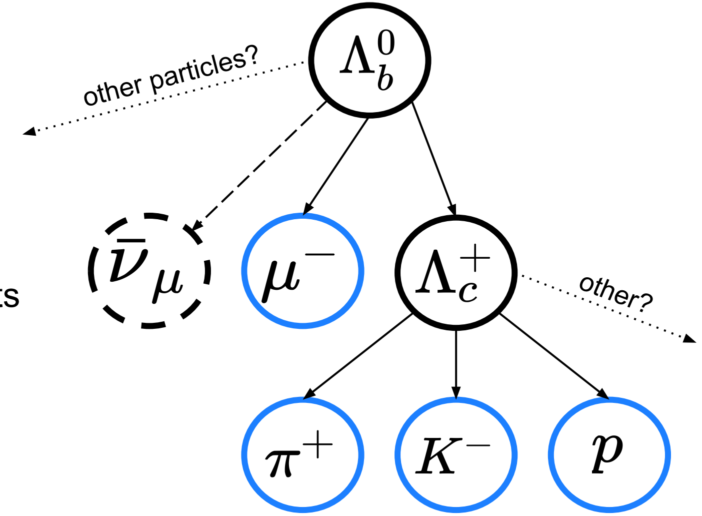
- compatibility with the `scikinC` package
- models compiled as shared library and **dynamically linked** to the main application (Gauss)
- distribution through WLCG nodes via `cvmfs`
- dynamic links **avoid to recompile** the main application for model updates → fast development cycle

Lamarr validation campaign

Lamarr provides the LHCb high-level response by relying on a **pipeline of ML-based modules**

To validate the ***flash-simulation* philosophy**, we employ the following decay mode:

- non-trivial semileptonic decay mode
  - crucial interface with LHCb-tuned physics generators
- **muons**, **pions**, **kaons**, and **protons** in a single decay
  - all particle species for which Lamarr provides models
- Lamarr-based samples, detailed simulated samples, and plots obtained from the **LHCb analysis software**
  - testing the integration with the current version of Gauss
- models training based on a **cocktail** of heavy flavour decays
  - $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ represents a negligible fraction of the sample

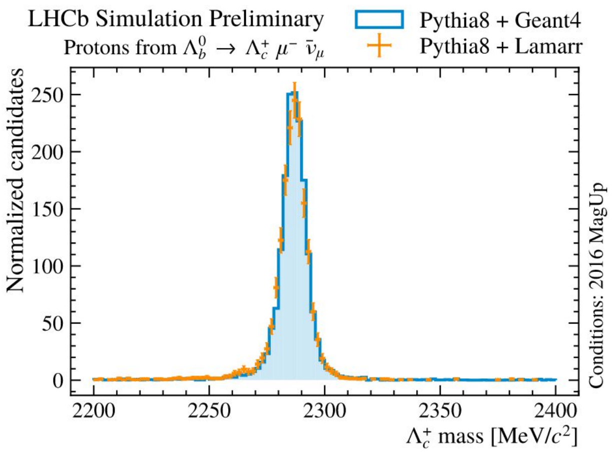


Lamarr validation campaign

Lamarr provides the LHCb high-level response by relying on a **pipeline of ML-based modules**

To validate the ***flash-simulation* philosophy**, we employ the following decay mode:

- non-trivial semileptonic decay mode
  - crucial interface with LHCb-tuned physics generators
- **muons**, **pions**, **kaons**, and **protons** in a single decay
  - all particle species for which Lamarr provides models
- Lamarr-based samples, detailed simulated samples, and plots obtained from the **LHCb analysis software**
  - testing the integration with the current version of Gauss
- models training based on a **cocktail** of heavy flavour decays
  - $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ represents a negligible fraction of the sample



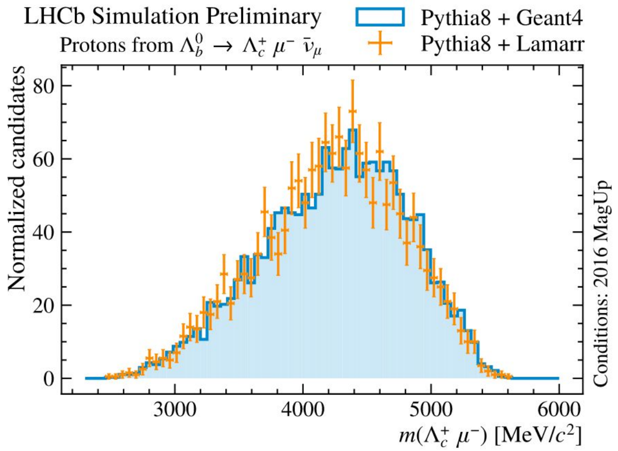
LHCb-FIGURE-2022-014

Lamarr validation campaign

Lamarr provides the LHCb high-level response by relying on a **pipeline of ML-based modules**

To validate the ***flash-simulation* philosophy**, we employ the following decay mode:

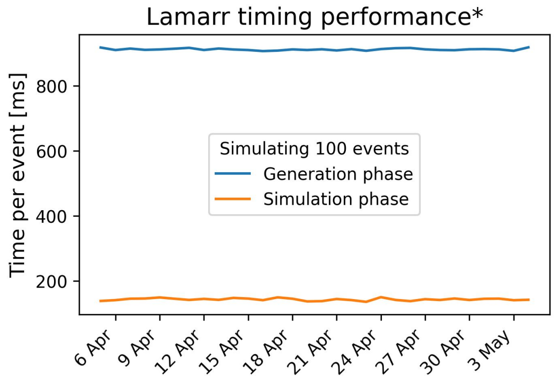
- non-trivial semileptonic decay mode
  - crucial interface with LHCb-tuned physics generators
- **muons**, **pions**, **kaons**, and **protons** in a single decay
  - all particle species for which Lamarr provides models
- Lamarr-based samples, detailed simulated samples, and plots obtained from the **LHCb analysis software**
  - testing the integration with the current version of Gauss
- models training based on a **cocktail** of heavy flavour decays
  - $\Lambda_b^0 \to \Lambda_c^+ \mu^- X$ represents a negligible fraction of the sample



LHCb-FIGURE-2022-014

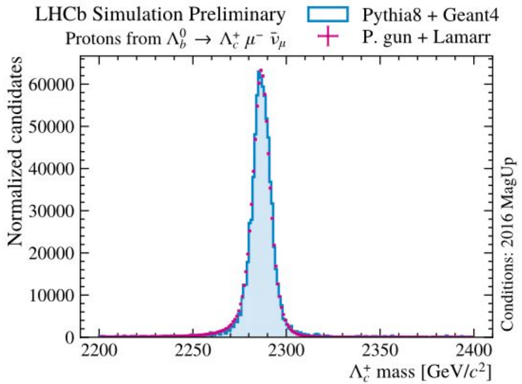
Preliminary timing studies

- Geant4-based simulations are expensive in terms of CPU
- Lamarr allows to reduce the CPU cost for the simulation phase of (at least) **two-order-of-magnitude**
- Pythia8 is the new **major CPU consumer** → the generation of *b*-baryons is expensive



* data obtained from the **LHCbPR** portal (2023/05)

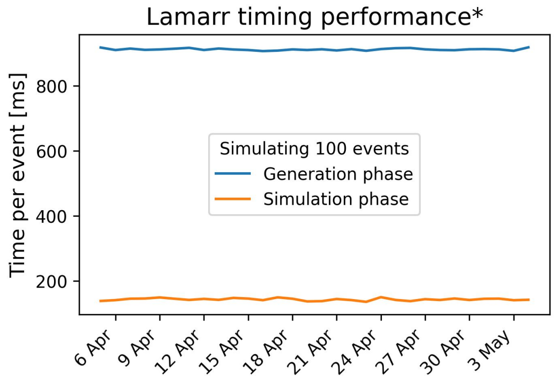
LHCb-FIGURE-2022-014



- Lamarr derives **high-quality** distributions from particle-guns
- The particle-gun approach drops to **almost zero** the cost of the Generation phase
- PGun + Lamarr → **three-order-of-magnitude** speed-up

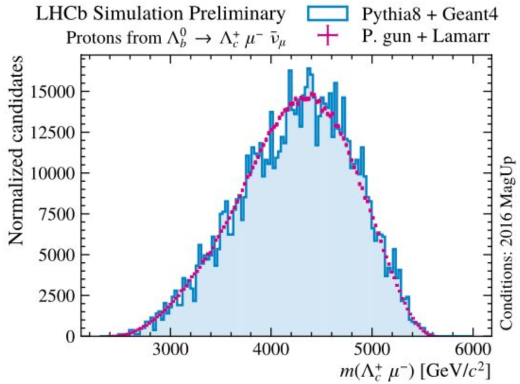
Preliminary timing studies

- Geant4-based simulations are expensive in terms of CPU
- Lamarr allows to reduce the CPU cost for the simulation phase of (at least) **two-order-of-magnitude**
- Pythia8 is the new **major CPU consumer** → the generation of *b*-baryons is expensive



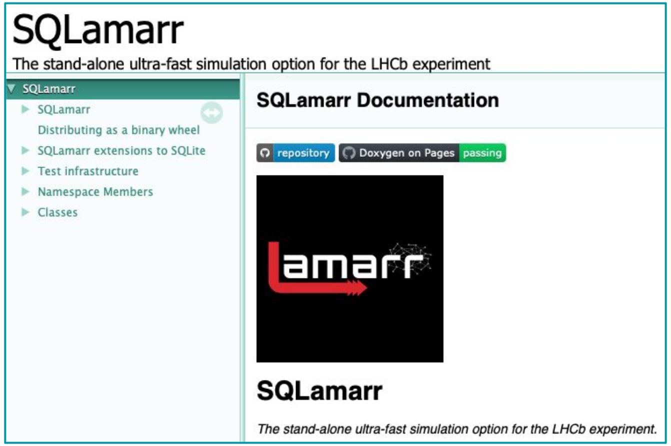
* data obtained from the **LHCbPR** portal (2023/05)

LHCb-FIGURE-2022-014



- Lamarr derives **high-quality** distributions from particle-guns
- The particle-gun approach drops to **almost zero** the cost of the Generation phase
- PGun + Lamarr → **three-order-of-magnitude** speed-up

Future of the Lamarr project



Integration of Lamarr within Gauss

- **mandatory** to work with the LHCb software ecosystem (*e.g.*, Gaudi, generators, DIRAC, Bender)
- **unappealing** for researches outside of the LHCb community

Efforts to decouple Lamarr from Gaudi → ***SQLamarr***

- LHCb Event Model mimic with a **SQLite database**
- **set of APIs** for loading data from physics generators and defining pipelines from models compiled as shared libraries

The development of SQLamarr moves in two different and complementary directions

- minimal dependencies, thread-safe database engine, pipeline configuration → **integration with Gauss-on-Gaussino** [17], the newer version of Gauss (based on Gaussino)
- providing a ***stand-alone* flash-simulation framework** → high-quality description of the LHCb experiment relying on ML-based parameterizations and particle-gun generated samples

The role of ICSC for flash-simulation

The **lifecycle** of a generic flash-simulation model includes:

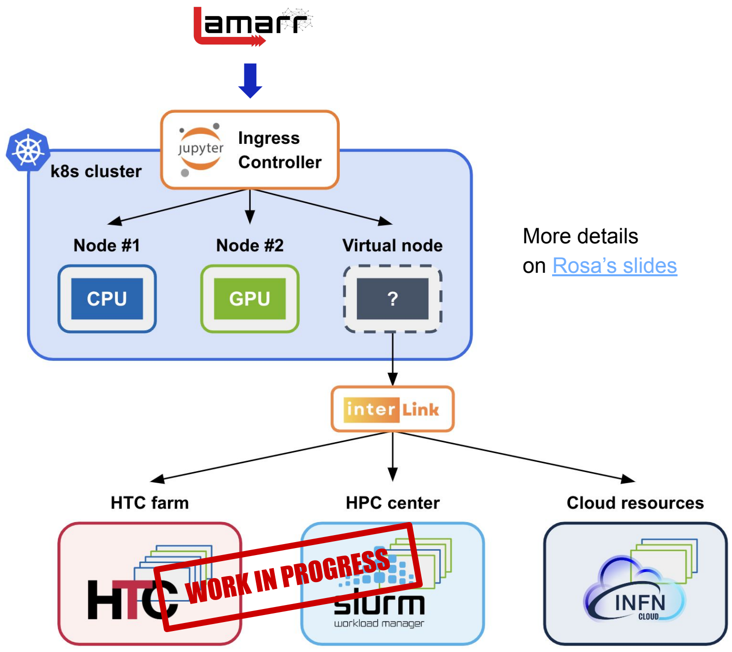
- designing
- training
- optimization [18]
- deployment [15,16]
- validation

⇨ development steps involve **GPU nodes** (**HPC paradigm**)

⇨ validation relies on the **production environment** (**HTC paradigm**)

The aim of **ICSC** (*Italian Center for SuperComputing*) is to create the national digital infrastructure for research and innovation, leveraging existing HPC, HTC and Big Data infrastructures and evolving towards a **cloud data-lake model**.

Lamarr is pioneering such **hybrid workloads** on **distributed and federated resources**



More details on Rosa's slides

Conclusions

- The ***Lamarr*** framework offers to LHCb the **fastest option for simulation** needed to meet the upcoming and future requests for simulated samples
- GAN-based models succeed in reproducing the errors introduced in the **detection** and **reconstruction** steps of both the tracking and PID systems of the LHCb experiment
- Transformers and GNNs powered by the **attention mechanism** and an **adversarial-driven training** are under investigation to parameterize the event-level response of ECAL to traversing photons
- A preliminary validation campaign demonstrates that a **pipeline of subsequent ML-based models** succeeds in reproducing high-quality reconstructed quantities, at a **few-percent fraction of the cost**
- Great effort to adapt the whole Lamarr workloads on **distributed and federated resources** taking the most from **all the computing paradigms** (HTC, HPC, and Cloud)

Thanks!

Any questions or comments?

Lucio Anderlini (INFN Firenze)
email: lucio.anderlini@cern.ch

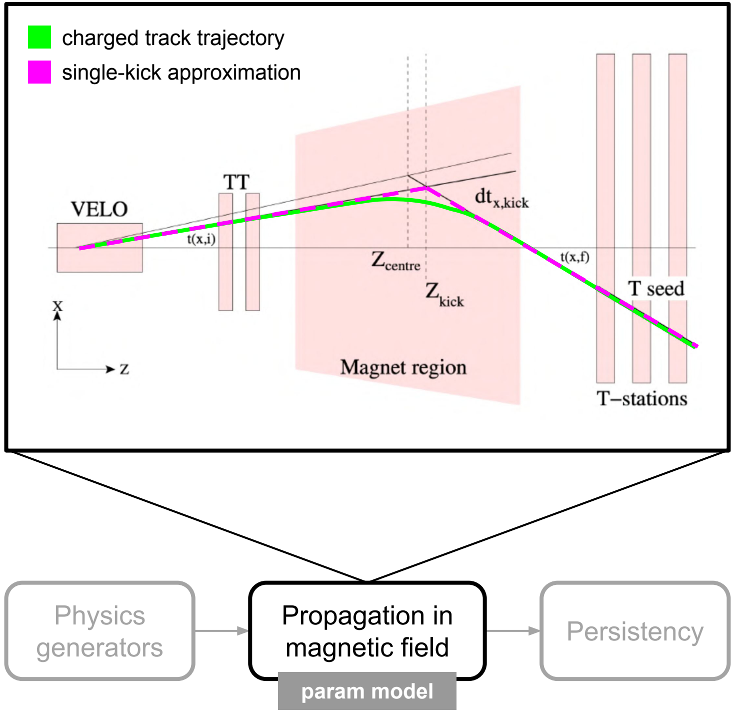
Matteo Barbetti (INFN CNAF)
email: matteo.barbetti@cern.ch

References

1. LHCb collaboration, A. Augusto Alves Jr., JINST **3** (2008) S08005
2. LHCb collaboration, R. Aaij *et al.*, JINST **19** (2024) P05065, arXiv:2305.10515
3. LHCb collaboration, LHCB-TDR-017, 2018
4. LHCb collaboration, M. Clemencic *et al.*, J. Phys.: Conf. Ser. **331** (2011) 032023
5. LHCb collaboration, D. Popov, EPJ Web Conf. **214** (2019) 02043
6. D. Müller *et al.*, Eur. Phys. J. C **78** (2018) 1009, arXiv:1810.10362
7. V. Chekalina *et al.*, EPJ Web Conf. **214** (2019) 02034, arXiv:1812.01319
8. LHCb collaboration, M. Rama *et al.*, EPJ Web Conf. **214** (2019) 02040
9. L. Anderlini *et al.*, PoS **ICHEP2022** (2022) 233
10. LHCb Simulation Project, M. Barbetti, arXiv:2303.11428
11. LHCb Simulation Project, L. Anderlini *et al.*, EPJ Web Conf. **295** (2024) 03040, arXiv:2309.13213
12. I. J. Goodfellow *et al.*, arXiv:1406.2661
13. A. Vaswani *et al.*, arXiv:1706.03762
14. F. Scarselli *et al.*, IEEE Trans Neural Netw **20** (2009) 61
15. L. Anderlini and M. Barbetti, PoS **CompTools2021** (2022) 034
16. R. Conlin *et al.*, Eng. Appl. Artif. Intell. **100** (2021) 104182
17. M. Mazurek, M. Clemencic and G. Corti, PoS **ICHEP2022** (2022) 225
18. M. Barbetti and L. Anderlini, arXiv:2301.05522

BACKUP

Propagation in magnetic field [1/2]



Crossing position and momentum coordinates don't require to know exactly the trajectory of particles

- tracking stations in region with **no-magnetic field** → straight track segments
- **no-dissipative effects** → constant variation of the momentum along the bending axis

The trajectory of a particle can be modeled by a single change of direction of the momentum vector in the xz-plane → **single-kick dipole approximation** requires only two parameters:

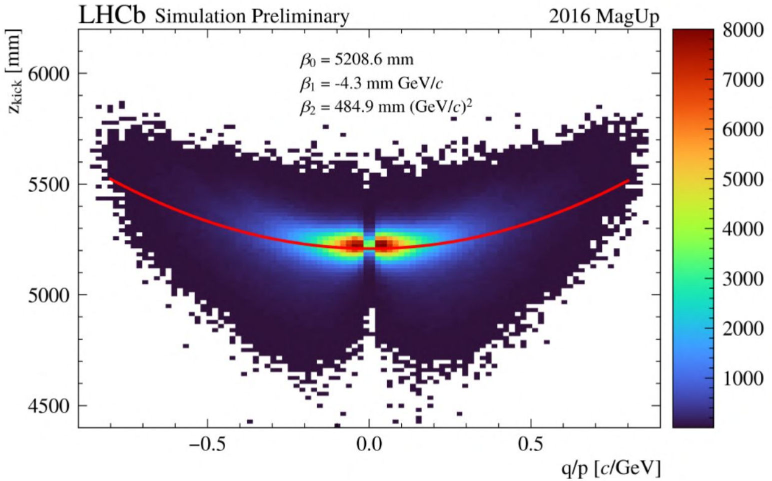
- Δp_x – constant variation of momentum p along bending axis
- z_{kick} – coordinate of point where magnet effect is condensed

Propagation in magnetic field [2/2]

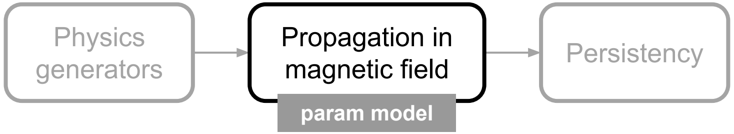
An expression for the z_{kick} coordinate follows from trivial trigonometric formulas

- by considering **negligible** Δp_y
- by requiring the momentum **conservation law**

$$z_{\text{kick}} = \frac{x' - x + z \cdot t_x - z' \cdot t'_x}{t_x - t'_x} \qquad \text{where} \qquad \begin{cases} t_x = p_x/p_z \\ t'_x = p'_x/p'_z \end{cases}$$

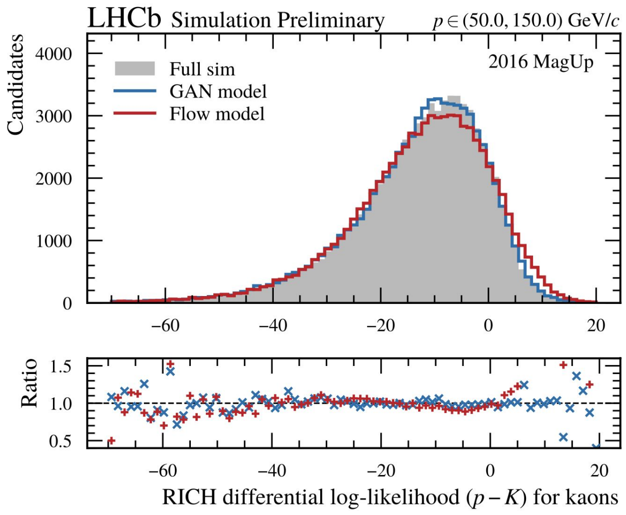


By fitting q/p versus z_{kick} with a **parabolic parametric function**, we are able to infer the crossing position and momentum coordinates simply relying on trigonometry.



RICH detectors: *alternative solution*

Recent developments in deep generative models reveal the effectiveness of using **Normalizing Flows** for fast detector simulation → promising results obtained by CMS with its *FlashSim* application



Preliminary study for LHCb flash simulations → RICH system

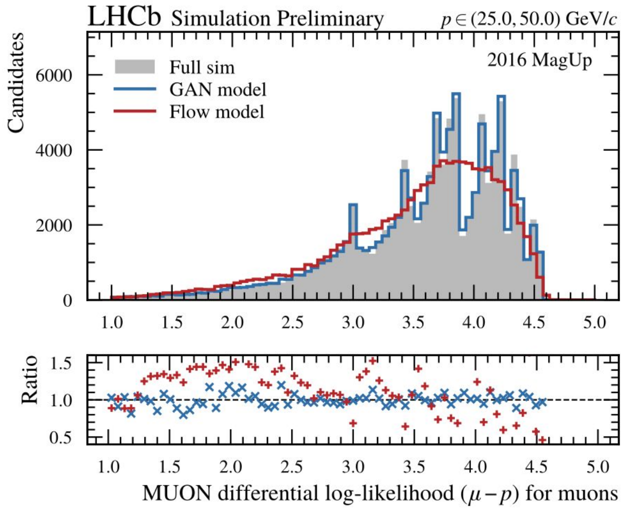
- same input/output of GAN-based models
- conditioned pdf directly learned by Flow-based models
- ***Masked Autoregressive Flows*** (MAF) used for these studies

Promising results in proton-kaon separation

- as for GANs, `RichDLLpK` **not included** in input conditions
- GAN performance benefits from the **auxiliary training process** (`RichDLLpK` only used by the discriminator)
- MAF-based models obtain **good results** even without the auxiliary training process

MUON detectors: *alternative solution*

Recent developments in deep generative models reveal the effectiveness of using **Normalizing Flows** for fast detector simulation → promising results obtained by CMS with its *FlashSim* application



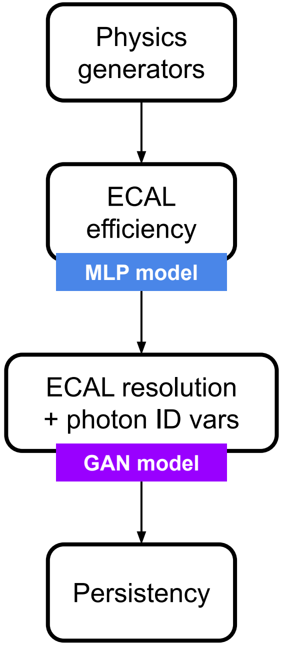
Preliminary study for LHCb flash simulations → MUON system

- same input/output of GAN-based models
- conditioned pdf directly learned by Flow-based models
- ***Masked Autoregressive Flows*** (MAF) used for these studies

Unsatisfactory results in muon-proton separation

- as for GANs, `muDLL` **not included** in input conditions
- GAN performance strongly benefits from the **auxiliary training process** (`muDLL` only used by the discriminator)
- MAF-based models **fail to reproduce the peaked structures** of the `muDLL` distribution without relying on the auxiliary procedure

Signal photons: *definition*



Photons included in the studied decay modes → accurate model of ECAL

- **unambiguous relation** between photons and (*matching*) clusters → *k*-to-*k* system
- **standard strategies** can be employed to describe the ECAL high-level response

The *k*-to-*k* relation follows from **geometrical and energetic constraints**:

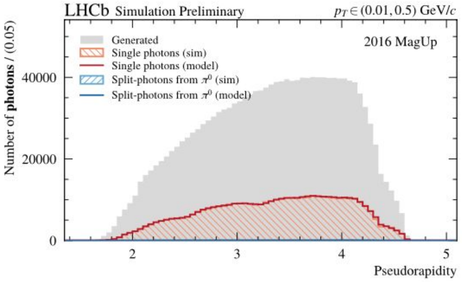
$$\sqrt{(x_{\text{photon}} - x_{\text{cluster}})^2 + (y_{\text{photon}} - y_{\text{cluster}})^2} < R_M \qquad |E_{\text{photon}} - E_{\text{cluster}}| < 2\,\sigma_E$$

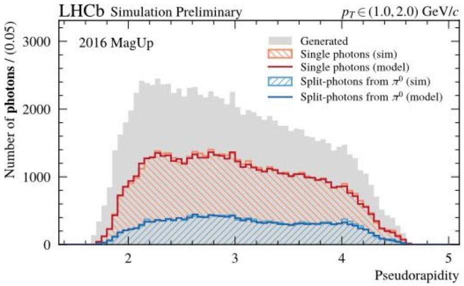
under these conditions a photon is considered reconstructed

The ***k*-to-*k* relation** enables to parameterize the ECAL high-level response by using the same techniques employed for tracking and PID models

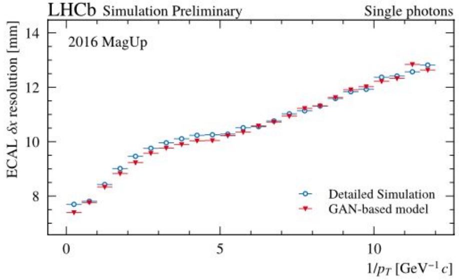
- **efficiency** → MLP-based model trained to predict the reconstruction probability
- **resolution** → GAN-based model trained to infer high-level quantities from generator-level information

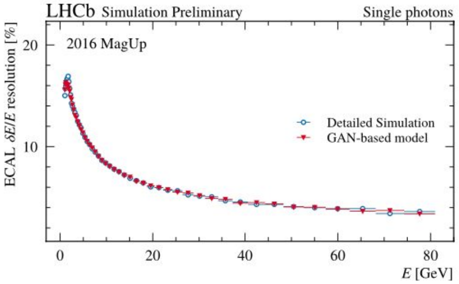
Signal photons: *validation*



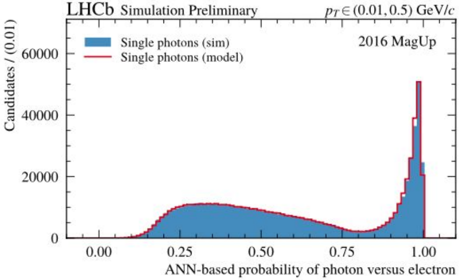


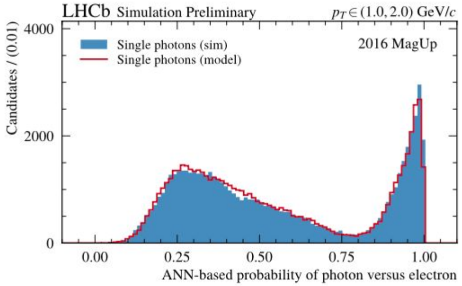
efficiency





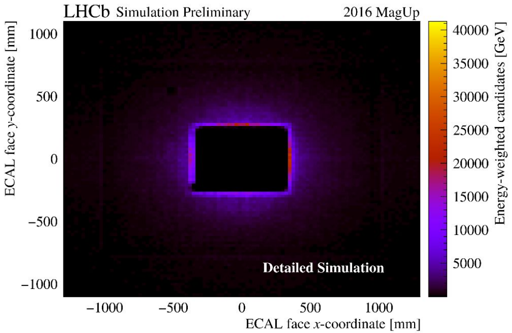
resolution

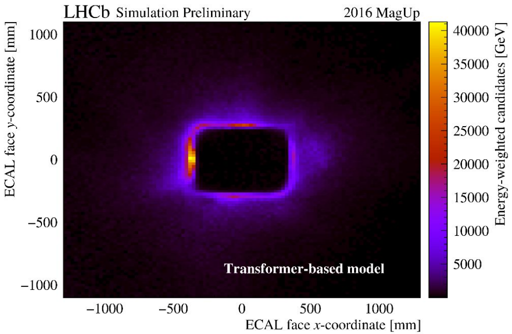




photon ID

Seq2seq approach





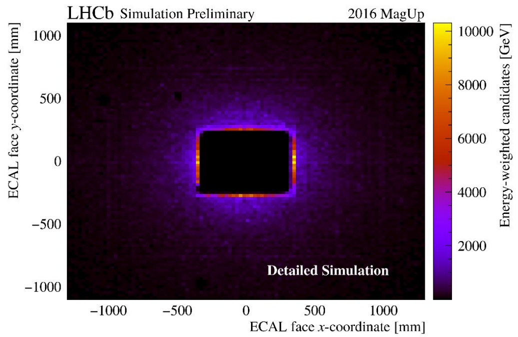
Aiming to directly facing the particle-to-particle correlation problem, the ECAL response can be described as a **translation problem**

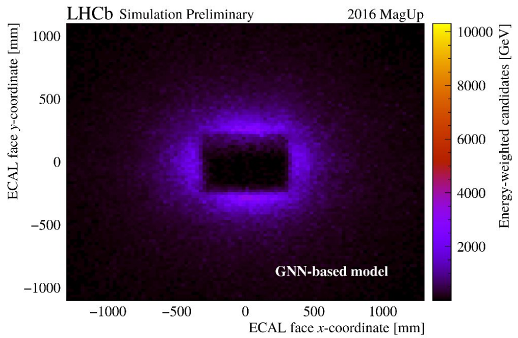
- source: sequence of *n* generated photons
- target: sequence of *m* reconstructed clusters

***Transformer*-based model** investigated to describe this *n*-to-*m* system

- *encoder-decoder* architecture powered by ***attention mechanism***
- *encoder* designed to process the source sequence (*i.e.*, generated photons), and parameterize photon-to-photon correlations
- *decoder* designed to process the target sequence (*i.e.*, reconstructed clusters), and parameterize both cluster-to-cluster and photon-to-cluster correlations
- training driven by a **regression task** → event-level ECAL description
- convergence trick → **adversarial-powered training** relying on DeepSets

Graph2graph approach



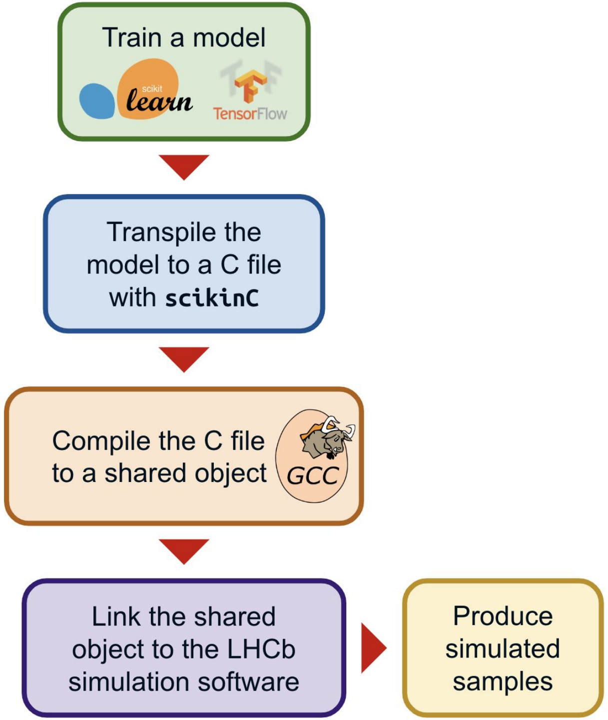


Relaxing the sorting statement at the basis of the seq2seq approach, we end up with the fact the graphs better describe the *topology* of calorimeter simulations → **graph2graph approach**

***GNN*-based model** investigated to describe this *n*-to-*m* system

- *heterogeneous graph* composed of two families of nodes (photon/cluster)
- photon edges follow a geometrical criteria in the (*x*, *y*, *E*)-space
- cluster edges randomly initialized to finite number of photon/cluster nodes
- message passing procedure powered by the **attention mechanism**
  - immutable photon features and updatable photon hidden states
  - updatable cluster features and updatable cluster hidden states
- training driven by a **regression task** → event-level ECAL description
- convergence trick → **adversarial-powered training** relying on DeepSets

The transpiling approach



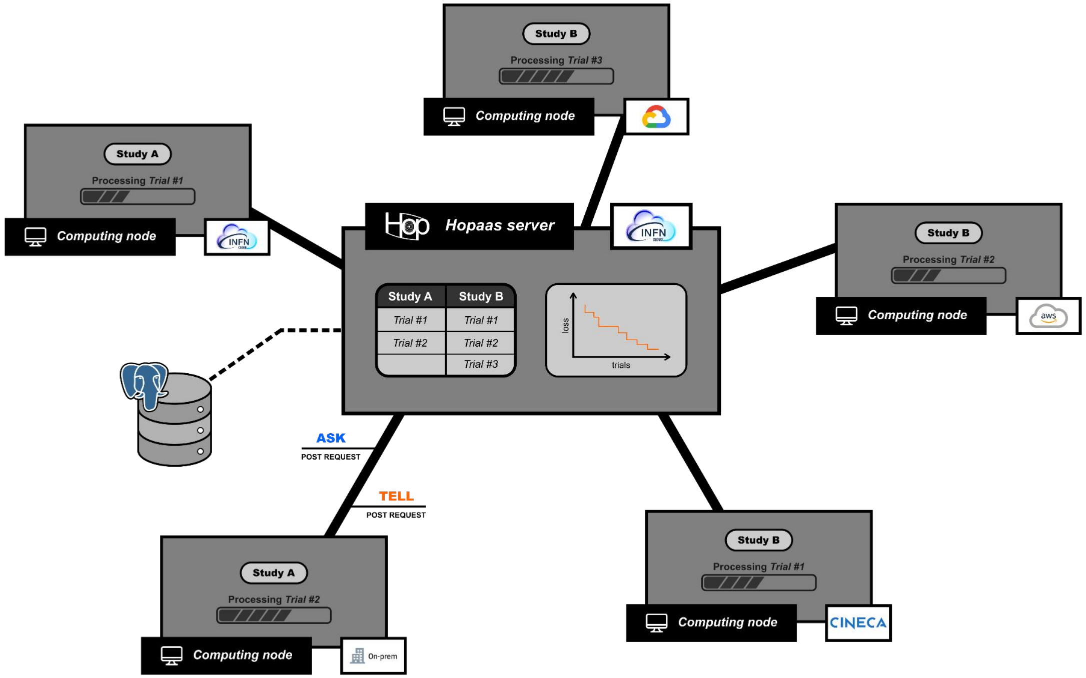
For a seamless integration of the trained parameterizations in the LHCb simulation framework models have to be applied to each single particle → **thousands of independent calls per event**

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

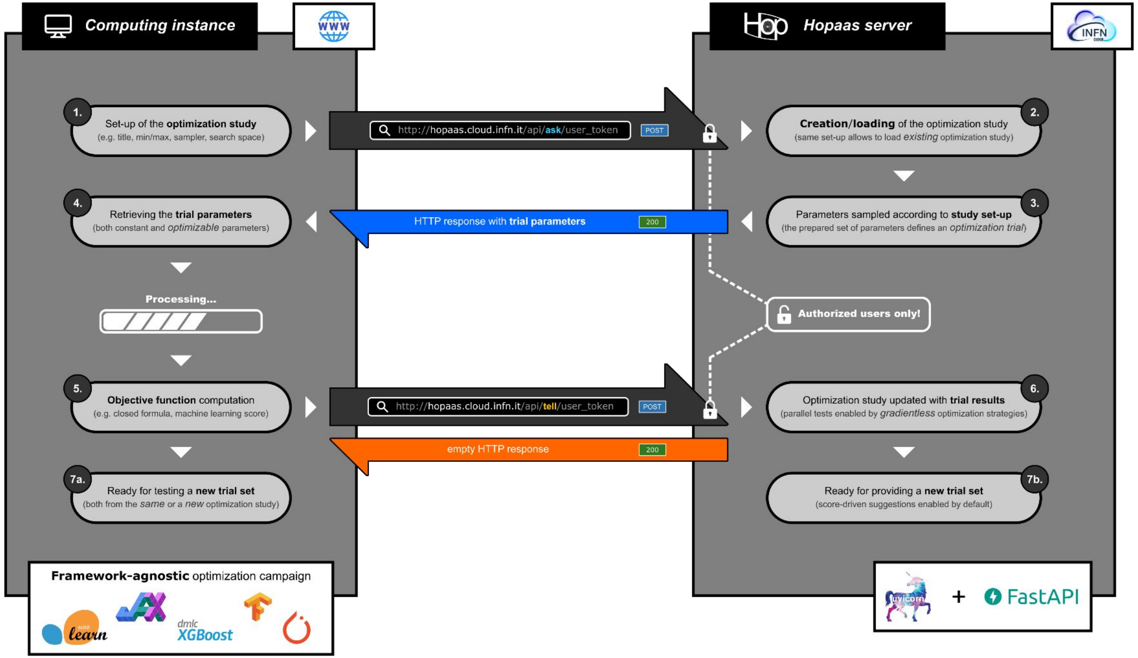
- LHCb tool: `scikinC` [15]
- Possible partial migration to: `keras2c` [16]

Hopaas: *multi-site optimization campaigns*



source:
https://hopaas.cloud.infn.it

Hopaas: *client-server system*



source:
https://hopaas.cloud.infn.it

Hopaas: *web dashboard*

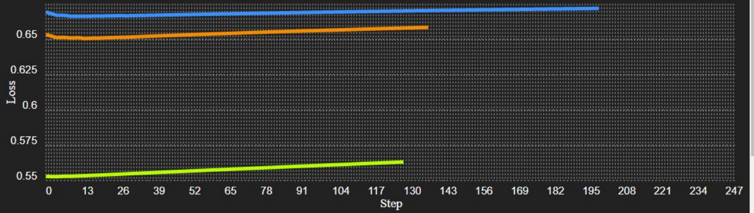
Hopaas | Guide | API reference | Python Client | Dashboard | Licence | Matteo Barbetti

suid: xHwHVl__ewNIyiuAUgLVyv-Jj1Ome0nbTZe3971pkmU

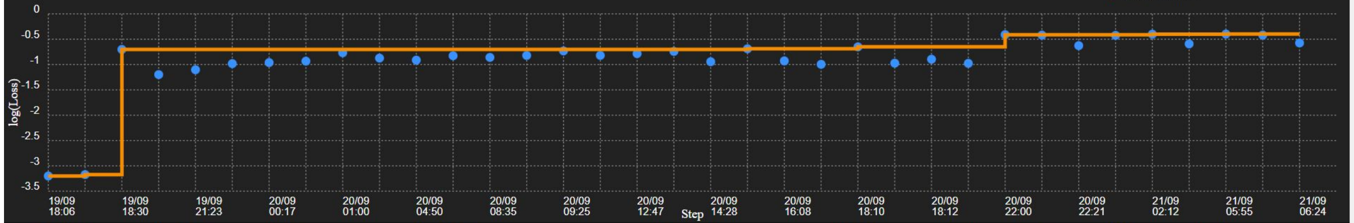
Report: Rich_Pion_2016-MagUp-simu_Study2-nw-opt

Running trials:

	Trial id	N steps	Created	Waited for	Duration	Declare
(blue)	34	202	2022-09-21 05:55:23	43	10912	FAILED
(orange)	35	140	2022-09-21 06:07:06	47	10205	FAILED
(green)	36	131	2022-09-21 06:24:13	37	9188	FAILED



✓ Show failed



Trial id	N steps	Loss	Created	Stopped	Status	d_nl	g_nl	d_lr	duxb	g_lr
34	202	0.672	2022-09-21 05:55:23	2022-09-21 08:57:15	running	9	5	5.7e-05	3	3.6e-05
32	250	0.670	2022-09-21 02:12:28	2022-09-21 05:53:38	complete	9	5	9.0e-05	3	3.2e-05

source:
https://hopaas.cloud.infn.it