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

LHCb

Generative models and seq2seq techniques for the flash-simulation of the LHCb experiment

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

42nd International Conference on High Energy Physics (ICHEP) | 20 July 2024

Computing requirements for simulation

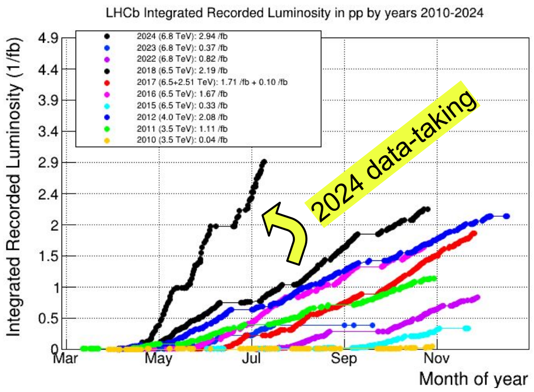
The baseline for simulation at LHCb is ***Detailed Simulation***:

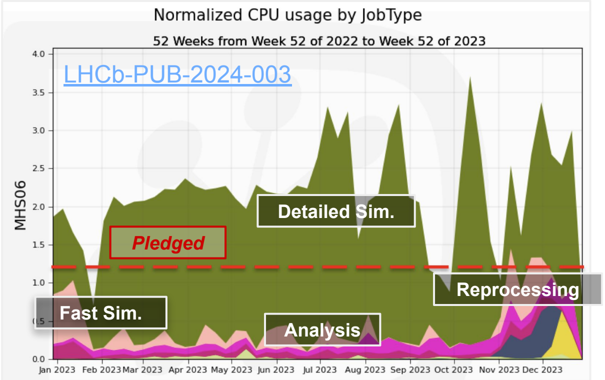
- simulation of all radiation-matter interactions
- simulated hits in detectors processed as real data
- **high CPU cost** (more than 90% used during LHC Run 2)
- unsustainable in the long term (*i.e.*, LHC Run 3+)

The **new Run 3 LHCb detector** (more details in the G. Tuci's parallel talk and Y. Ahmis' plenary talk) is collecting an increased amount of data. This puts **severe pressure on the CPU resources** to meet the upcoming and future requests for simulated samples.

Relying only on Detailed Simulation will **far exceed the computing budget** (pledge) of the experiment → **evolving the simulation technologies** is mandatory!

* **Gauss** [1] is the LHCb simulation framework based on Gaudi





Fast simulation vs. Flash simulation

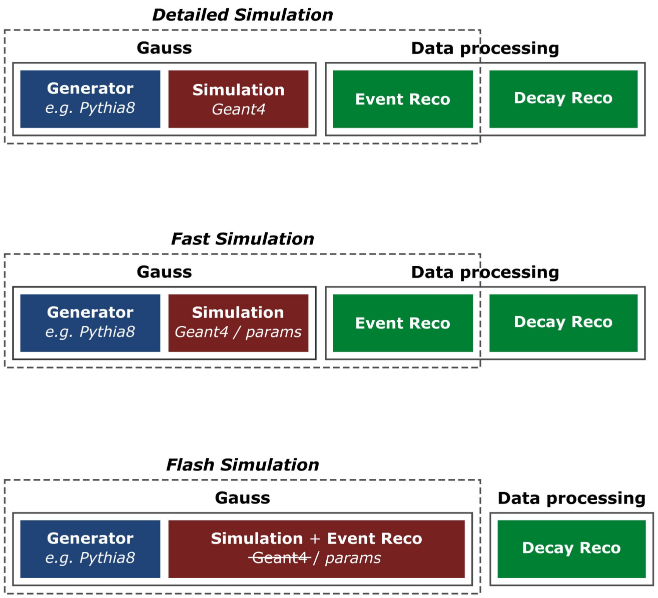
Methods to **speed up** the Geant4-based simulation productions:

- upgrade of the simulation framework (including multi-threading)
- leveraging GPU-acceleration (*e.g.*, use AdEPT, Celeritas)
- reuse of the not-signal part of the event, **ReDecay** [2]

Fast Simulation techniques to parameterize the detector low-level response without relying on Geant4:

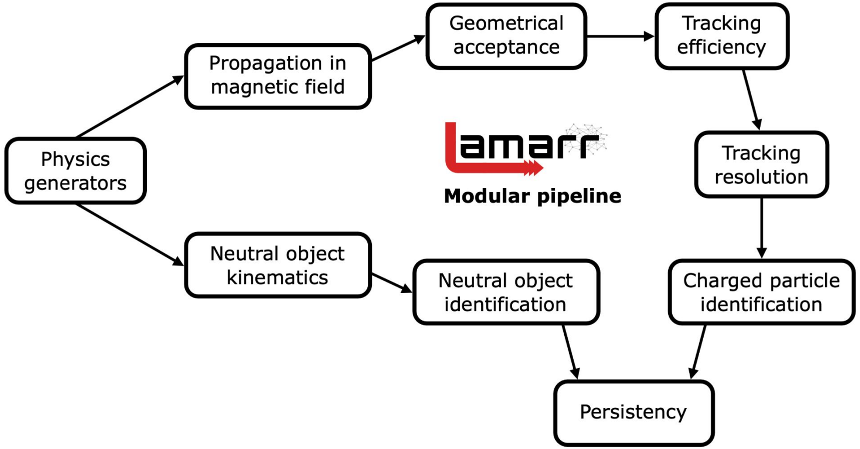
- **Point library** for Calorimeters energy deposits [3]
- **Generative Models** (*e.g.*, GAN, VAE) for Calorimeters energy deposits [4]

Flash Simulation (also called *Ultra-Fast* or *parametric*) defines a more radical approach by replacing Geant4 and reconstruction with parameterizations able to **directly transform** generator-level particles into analysis-level reconstructed objects



Lamarr: the LHCb flash-simulation option

Lamarr [5-7] 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.



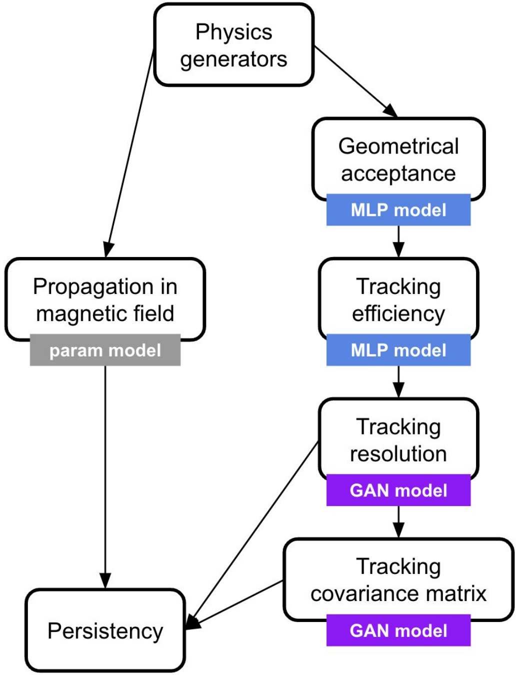
The Lamarr pipeline can be split in two branches:

1. a branch treating **charged particles** relying on tracking and particle identification (RICH + MUON + GPID) parameterizations
2. a branch treating **neutral particles** that require an accurate parameterization of the ECAL detector

The Lamarr pipeline for the tracking system

Lamarr relies on the following models to parameterize the high-level response of the **LHCb tracking system** for a set of quasi-stable charged particles (electrons, muons, and hadrons):

- **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

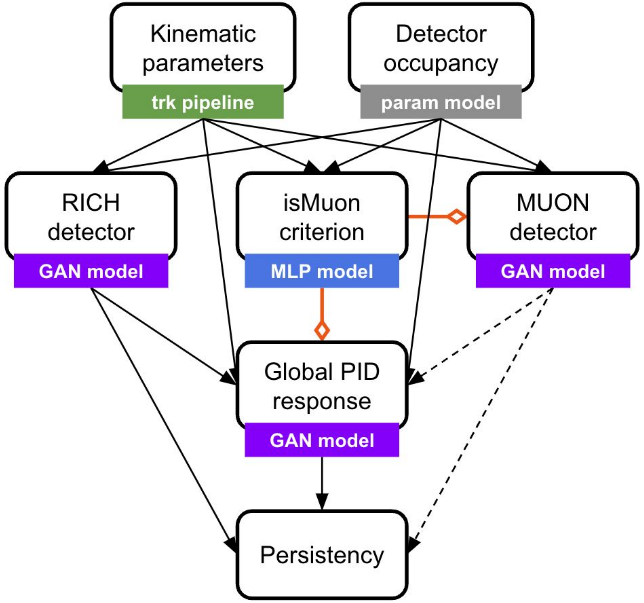


The Lamarr pipeline for the PID system

Lamarr relies on the following models to parameterize the high-level response of the **LHCb particle identification system**:

- **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 [8]

Models specialized for **muons**, **pions**, **kaons**, and **protons** are provided by Lamarr for each set of PID variables.



Parameterizations for charged particles

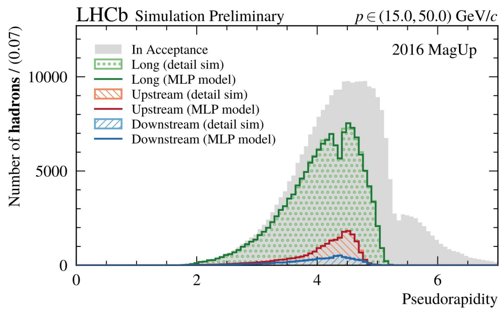
Unambiguous relation between *k* generated particles and *k* (<=) reconstructed tracks → 2 families of models

Efficiency model

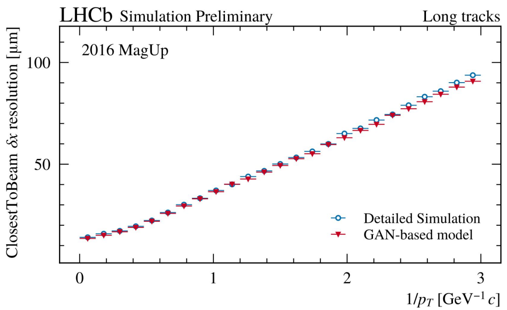
Neural network trained to perform a classification task to capture the fraction of **"good" candidates** (*e.g.*, accepted, reconstructed, selected) as a function of generator-level quantities

"Resolution" model

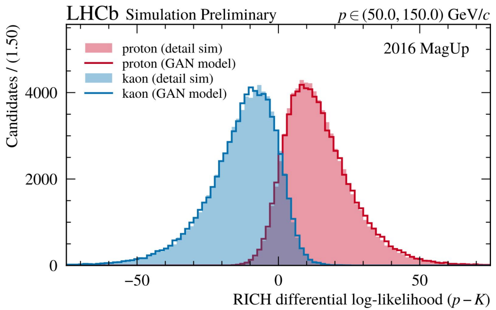
GAN [9] trained to learn the probability distributions of **variables at reconstructed-level** (*e.g.*, reconstructed errors, PID likelihoods, classification probabilities) considering generator-level conditions



trk reco efficiency



trk reco resolution



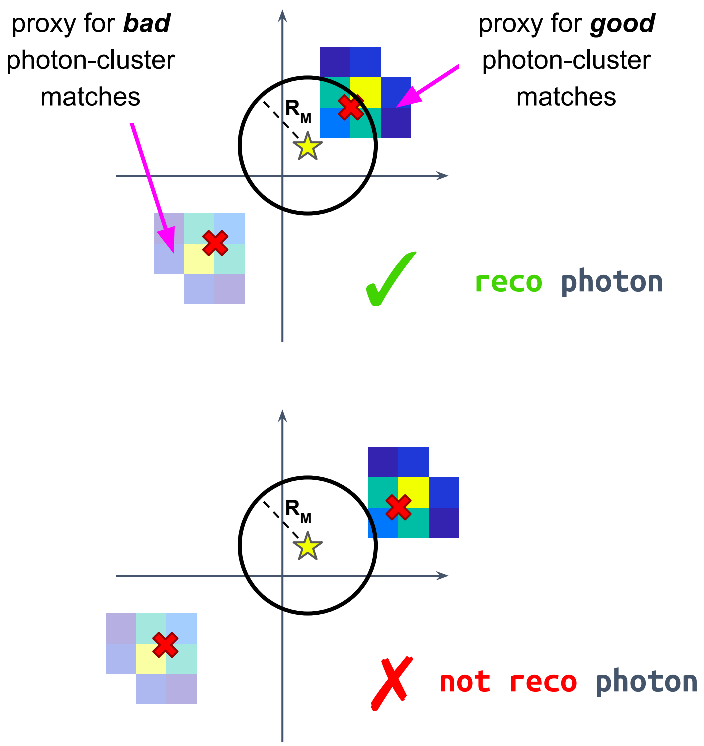
RICH DLL model

The ECAL detector in Lamarr

The flash-simulation of the ECAL detector is a <u>more complex task</u> due to **secondary processes** that lead to have *n* generated particles responsible for *m* reconstructed calorimetric clusters (with $n \neq m$)

While for <u>background</u> modeling the *n*-to-*m* relations cannot be evaded, in case of **signal photons** Lamarr is investigating a strategy similar to the one employed for charged particles (track + PID) and eligible once a **photon-cluster matching rule** is defined

Geometric rules allow to define <u>*reconstructed* photons</u> by enforcing **unambiguous *k*-to-*k* relations**



Parameterizations for signal photons

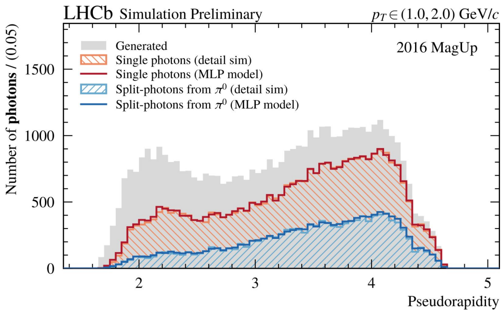
Forcing an **unambiguous relation** between *k* generated photons and *k* (<=) reconstructed clusters → 2 families of models

Efficiency model

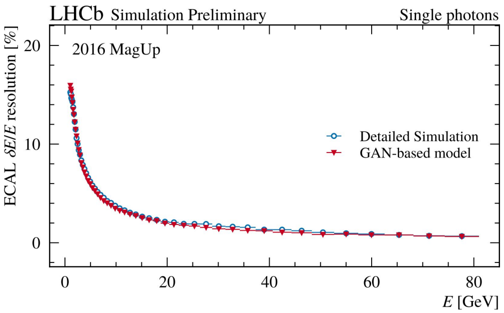
Neural network trained to perform a classification task to capture the fraction of ***reconstructed* signal photons** (photons that geometrically match with clusters) as a function of generator-level quantities

"Resolution" model

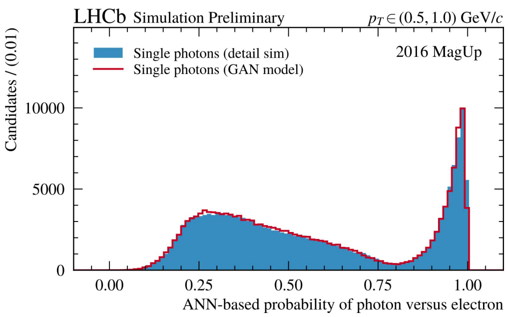
GAN [9] trained with conditions (generator-level quantities) to parameterize both the **resolution effects** and **photon identification variables** as obtained from reconstructed ECAL clusters



photon efficiency



photon resolution



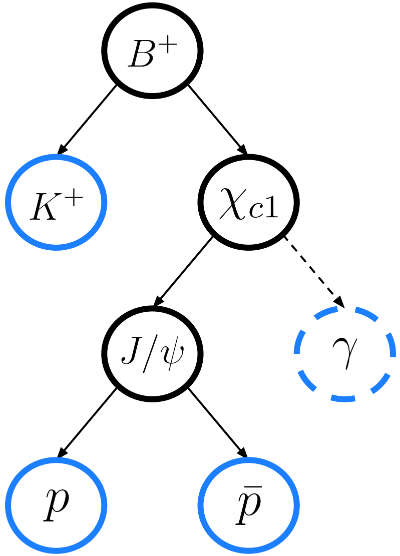
photon ID variable

Validation of the flash-simulation paradigm

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

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

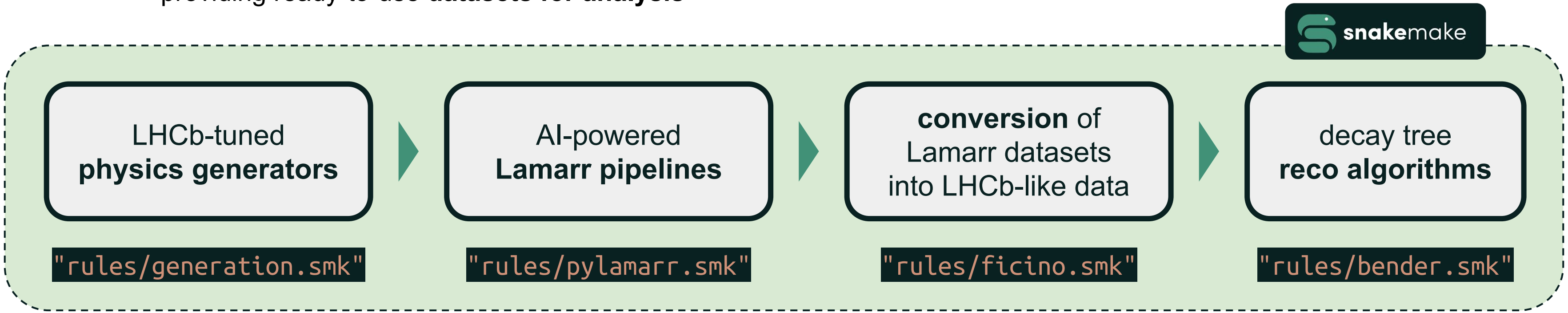
- interesting *B*-meson decay channel
  - crucial for *charmonium* spectrum studies (latest results in the parallel talks from V. Yeroshenko and L.M. Garcia Martin)
- **kaons**, **protons**, and **photons** in a single decay
  - most of the particle species for which Lamarr provides models
- Lamarr-based samples, detailed simulated samples, and plots obtained from the **LHCb analysis software**
  - enabling to test the integration with the LHCb software stack
- training samples made of a **cocktail** of heavy flavour decays
  - the validation decays represents a negligible fraction of the sample



The Lamarr software

Lamarr has been designed with dual capabilities:

- being a ***stand-alone* simulation framework**:
  - fast development cycle in Python environments as typical in machine learning projects
  - use of ML backend-agnostic models by relying on a ***trancompilation* approach** [10]
- being seamlessly **integrated with Gauss(-on-Gaussino)** [1,11]:
  - interface with all the **LHCb-tuned physics generators**
  - access to Grid distributed computing resources and production environment
  - providing ready-to-use **datasets for analysis**



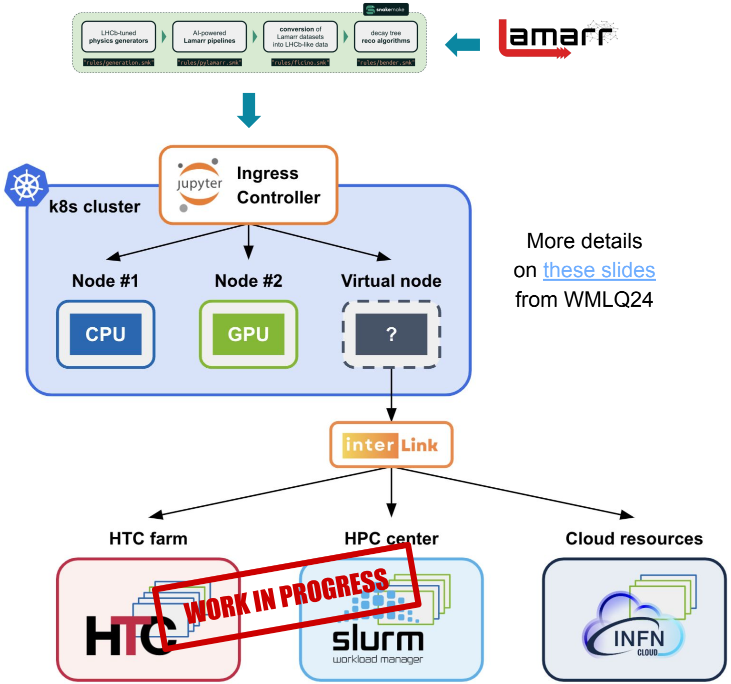
Exploiting Cloud resources for validation

Great effort has been spent to integrate Lamarr with modern Cloud technologies, like Kubernetes (K8s):

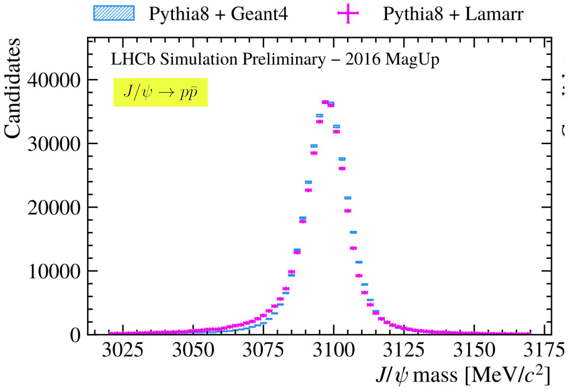
- access to Cloud computing resources
- hardware-aware workflows (on CPU and/or GPU)
- ***quasi-interactive* production environment** for simulations

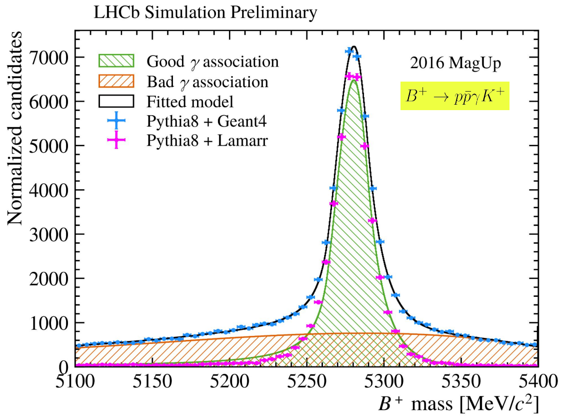
By relying on a K8s-powered snakemake-based workflow, the Lamarr validation campaign was successfully performed combining the resources provisioned by **three different Cloud computing sites** scattered across Italy (Cloud@CNAF, CloudVeneto, and Cloud@ReCaS-Bari)

The workload for validation was distributed among the 3 sites by relying on the *Virtual Kubelet* mechanism with interLink as provider allowing to expand K8s **beyond the local cluster nodes**



More details on these slides from WMLQ24





Lamarr sanity checks

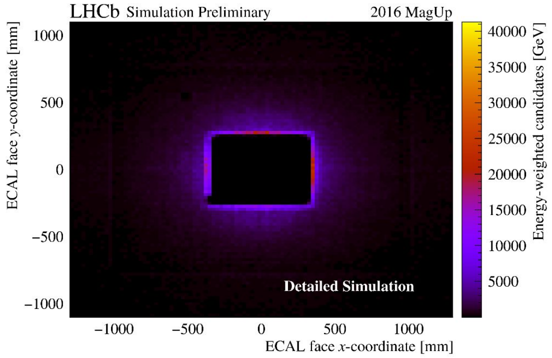
The described workflow allows to well reproduce the **invariant masses** of all the hadrons involved in the validation decay chain, demonstrating the capabilities of Lamarr to parameterize the high-level response of the LHCb experiment:

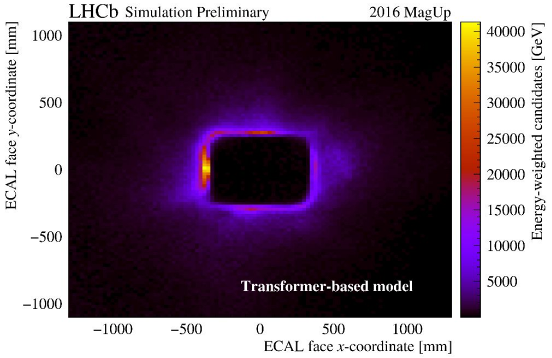
- integration with LHCb-tuned generators → good generated kinematics ✓
- tracking pipeline → smearing effects and reconstruction uncertainties ✓
- PID pipeline → protons and kaons PID variables ✓
- ECAL pipeline → “efficiency” and smearing effects ≈

Enforcing *k*-to-*k* relations, the pipeline chosen for signal photons has limited capabilities and fails to reproduce the plateau of wrongly associated photons shown in the *B* mass from Geant4 → need for an **evolution of the ECAL model**

NOTE As Detailed Simulation, also flash-simulated datasets can be **further processed** by using the standard combinatorial selection algorithms → invariant mass and decay vertexing

Facing the n-to-m ECAL problem





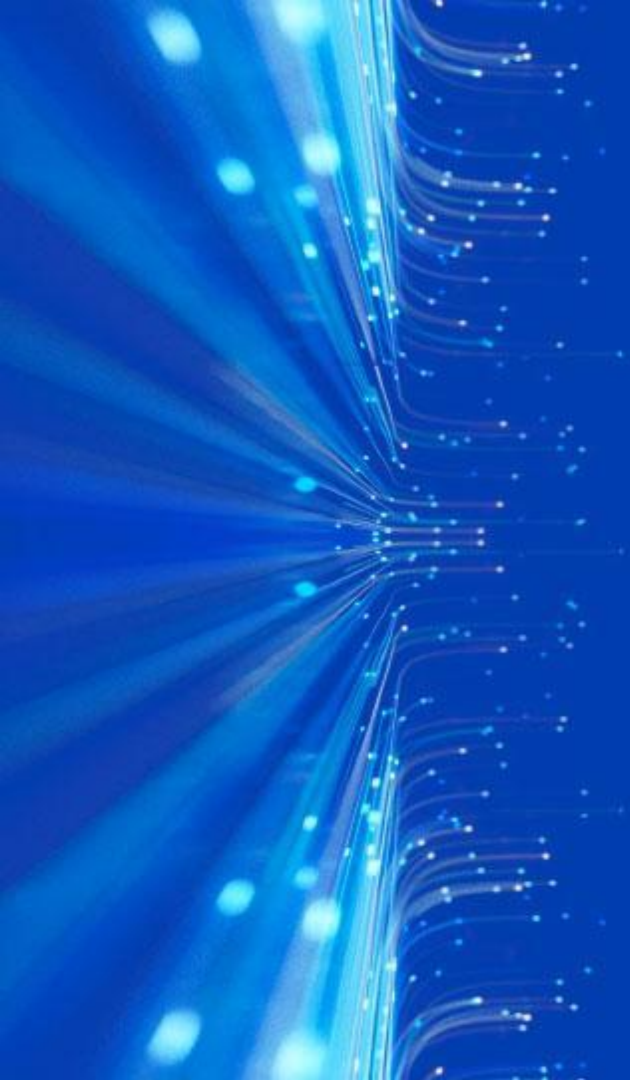
In general, a reconstructed cluster *matches* with a generated photon if any fraction of the energy deposited comes from that photon → **particle-to-particle correlation problem**

The correlations require that the ECAL flash-simulation relies on models able to describe the high-level response due to all the particles traversing the detector in an event at once → we need an **event-level parameterization** for ECAL

Lamarr is investigating the use of **seq2seq models**, like Transformers [12] and GNNs [13], to face the *n*-to-*m* problem and the capabilities of the **attention mechanism** to capture the correlations between generated particles and reconstructed objects

Conclusions

- ***Lamarr*** offers to LHCb the fastest option for simulation production
  - **pipelines of subsequent ML models** succeed in implementing the *flash-simulation* paradigm
- Lamarr is designed as a ***stand-alone*** simulation framework **well-integrated** with the **LHCb Simulation software**
  - the ***transcompilation* approach** allows to deploy the ML models as plugins within the framework
- Validation campaigns ongoing to test the validity of the flash-simulation philosophy and software implementation
  - test also for a ***quasi-interactive* production environment** for simulation on federated Cloud resources
- Validation studies have highlighted **pros** and **cons** of employing flash-simulation strategies
  - significant reduction of the simulation cost for high-quality simulated samples
  - work still needed to face the *n*-to-*m* relation problem for an **event-level description** of the detector response



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, M. Clemencic *et al.*, J. Phys.: Conf. Ser. **331** (2011) 032023
2. D. Müller *et al.*, Eur. Phys. J. C **78** (2018) 1009, arXiv:1810.10362
3. LHCb collaboration, M. Rama *et al.*, EPJ Web Conf. **214** (2019) 02040
4. V. Chekalina *et al.*, EPJ Web Conf. **214** (2019) 02034, arXiv:1812.01319
5. L. Anderlini *et al.*, PoS **ICHEP2022** (2022) 233
6. LHCb Simulation Project, M. Barbetti, arXiv:2303.11428
7. LHCb Simulation Project, L. Anderlini *et al.*, EPJ Web Conf. **295** (2024) 03040, arXiv:2309.13213
8. LHCb collaboration, R. Aaij *et al.*, Int. J. Mod. Phys. **A30** (2015) 1530022, arXiv:1412.6352
9. I. J. Goodfellow *et al.*, arXiv:1406.2661
10. L. Anderlini and M. Barbetti, PoS **CompTools2021** (2022) 034
11. M. Mazurek, M. Clemencic and G. Corti, PoS **ICHEP2022** (2022) 225
12. A. Vaswani *et al.*, arXiv:1706.03762
13. F. Scarselli *et al.*, IEEE Trans Neural Netw **20** (2009) 61
14. R. Conlin *et al.*, Eng. Appl. Artif. Intell. **100** (2021) 104182
15. M. Zaheer *et al.*, arXiv:1703.06114
16. F. Vaselli *et al.*, CERN-CMS-NOTE-2023-003
17. G. Papamakarios *et al.*, arXiv:1912.02762

BACKUP

Priorities for flash-simulated samples

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

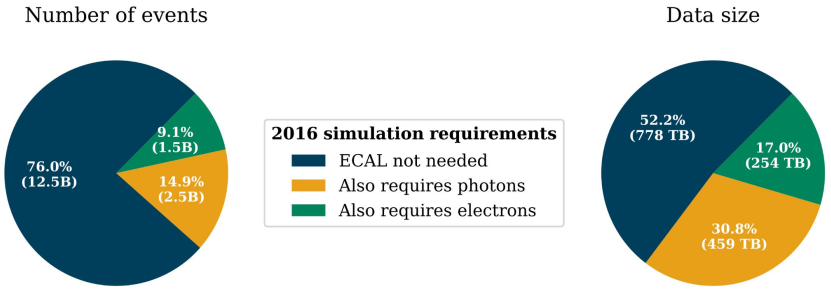
- most of the events (76%) don't require ECAL
- photons and electrons are less requested

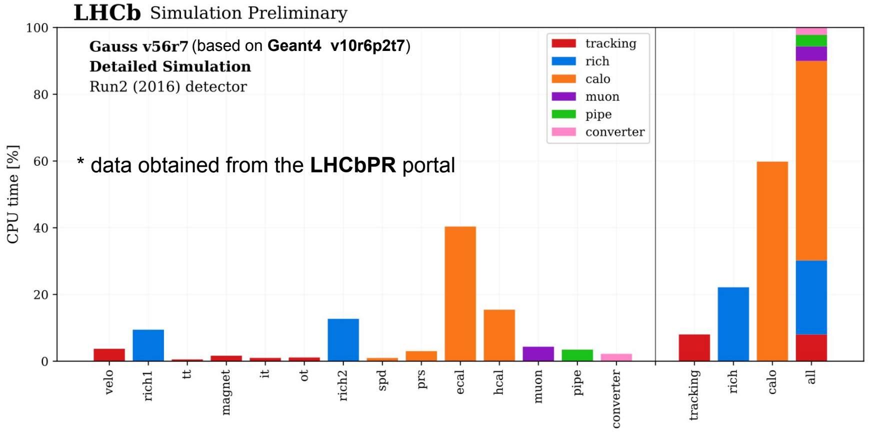
The simulation cost is driven by Geant4

- 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

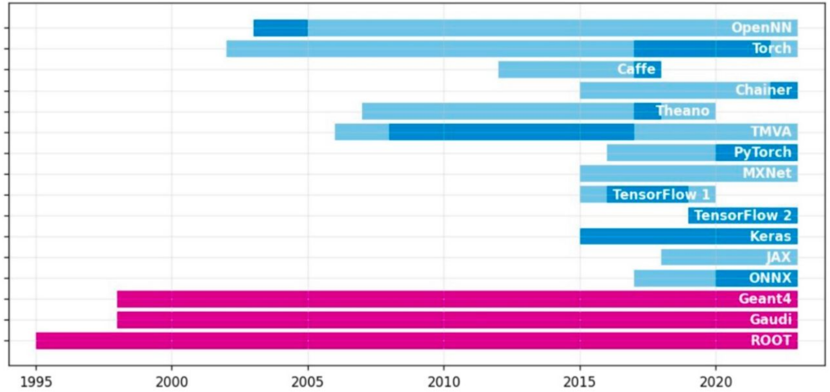




Integration with the LHCb software stack

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



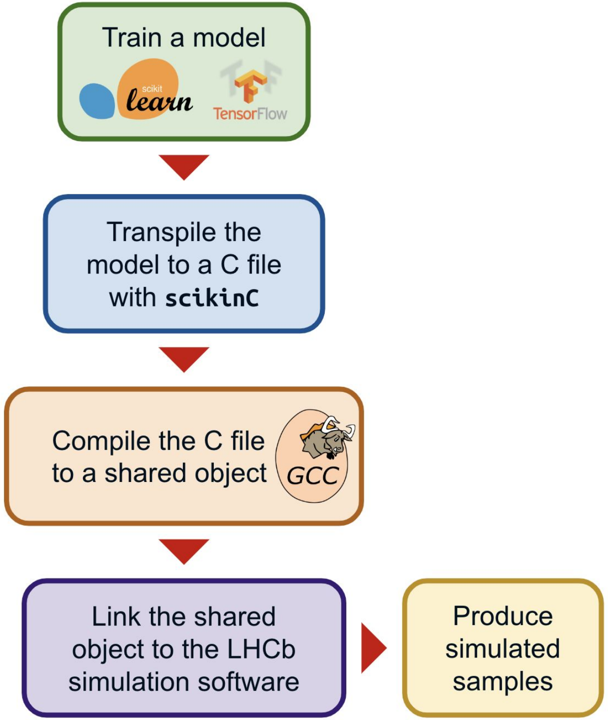
Most of the Lamarr parameterizations are ML-based:

- 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** [10,14]

- 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

The transcompilation approach



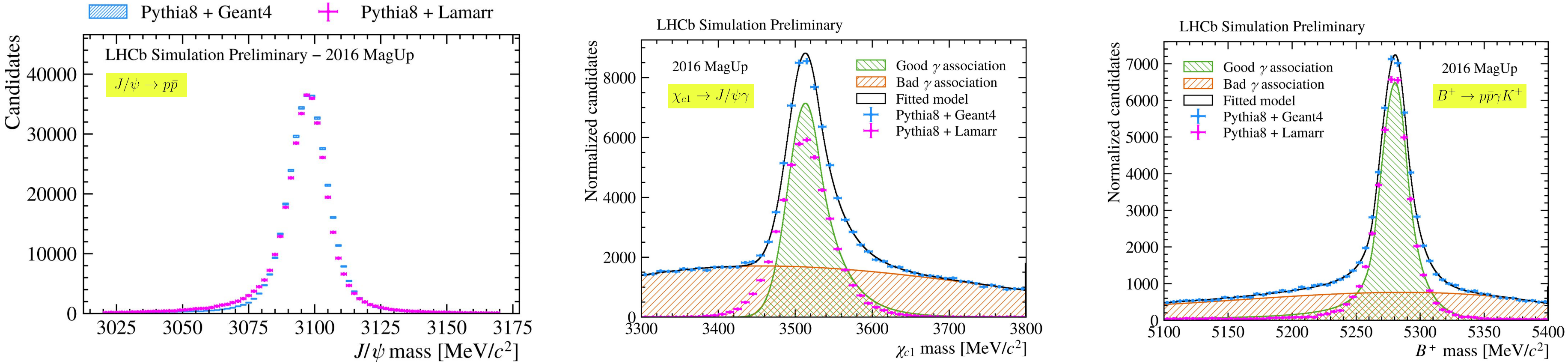
For a seamless integration of the trained parameterizations in the LHCb simulation framework models have to be applied to each single particle → **thousands of independent calls per event**

Even a small latency (*e.g.*, *context switching*) wastes unacceptable amount of CPU resources

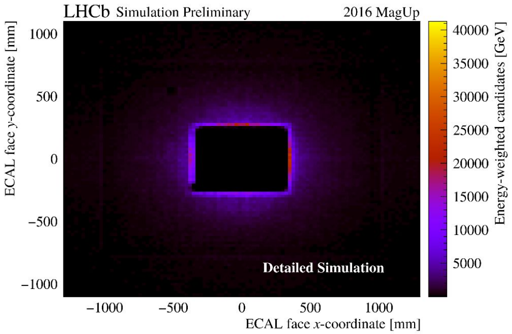
Lamarr solution → we **transpile the trained models in C** and compile them to binaries, **dynamically linked** at runtime

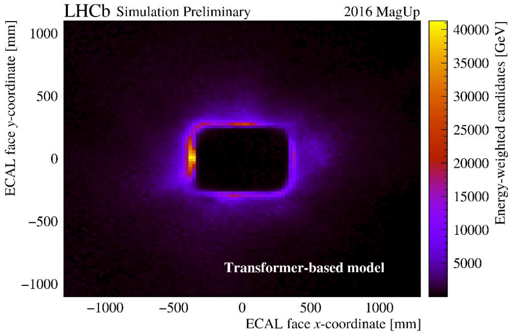
- LHCb tool: `scikinC` [10]
- Possible partial migration to: `keras2c` [14]

Photon-cluster association issues



Seq2seq approach





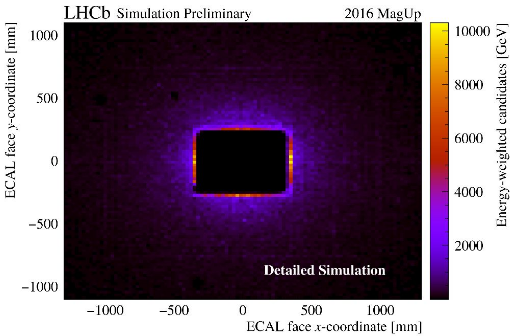
Aiming to directly facing the particle-to-particle correlation problem, the ECAL response can be described as a **translation problem**

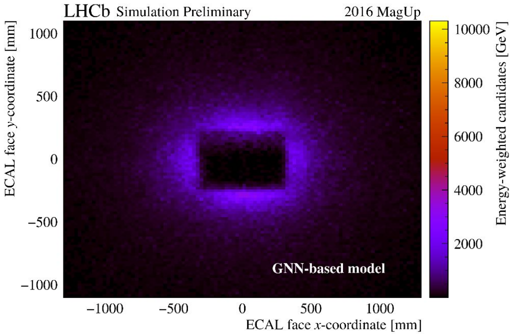
- source: sequence of *n* generated photons
- target: sequence of *m* reconstructed clusters

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

- *encoder-decoder* architecture powered by ***attention mechanism*** [12]
- *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 [15]

Graph2graph approach





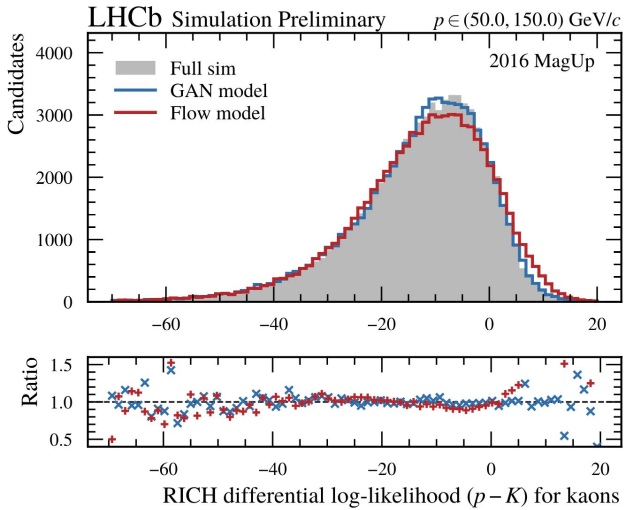
Relaxing the sorting statement at the basis of the seq2seq approach, we end up with the fact the graphs better describe the *topology* of calorimeter simulations → **graph2graph approach**

***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** [12]
  - 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 [15]

RICH detectors: *alternative solution*

Recent developments in deep generative models reveal the effectiveness of using **Normalizing Flows** for fast/flash detector simulation [16]



Preliminary study for LHCb flash simulations → RICH system

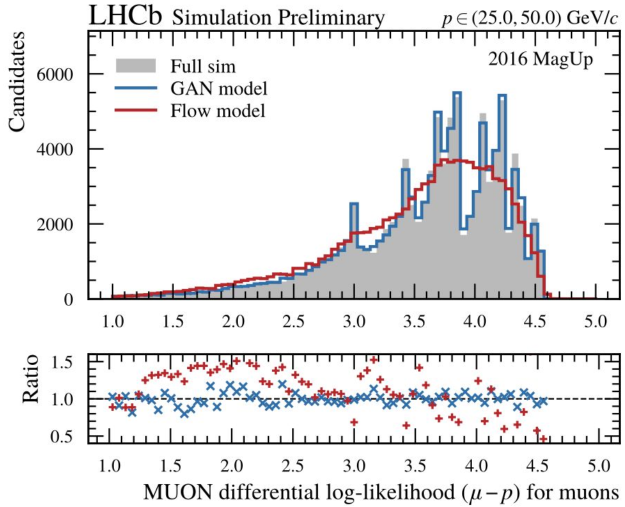
- same input/output of GAN-based models
- conditioned pdf directly learned by Flow-based models
- ***Masked Autoregressive Flows*** (MAF) [17] used for these studies

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/flash detector simulation [16]



Preliminary study for LHCb flash simulations → MUON system

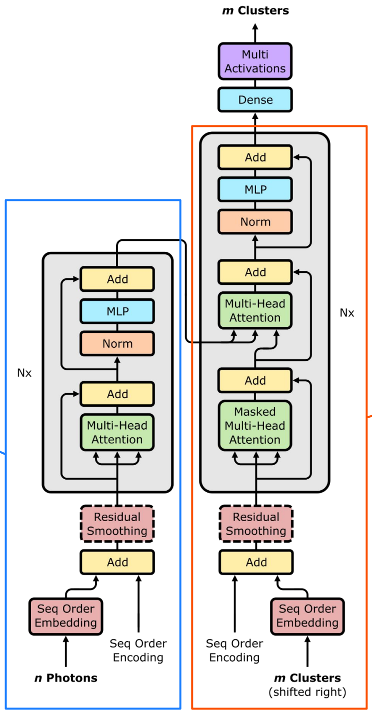
- same input/output of GAN-based models
- conditioned pdf directly learned by Flow-based models
- ***Masked Autoregressive Flows*** (MAF) [17] used for these studies

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

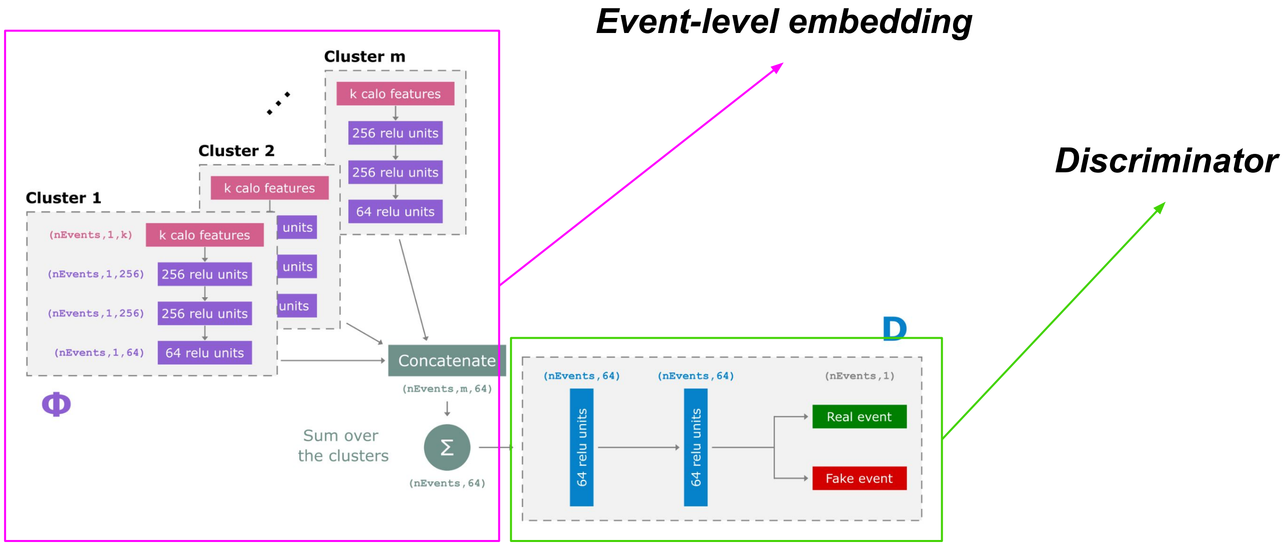
Transformer architecture

Photon sequence encoder

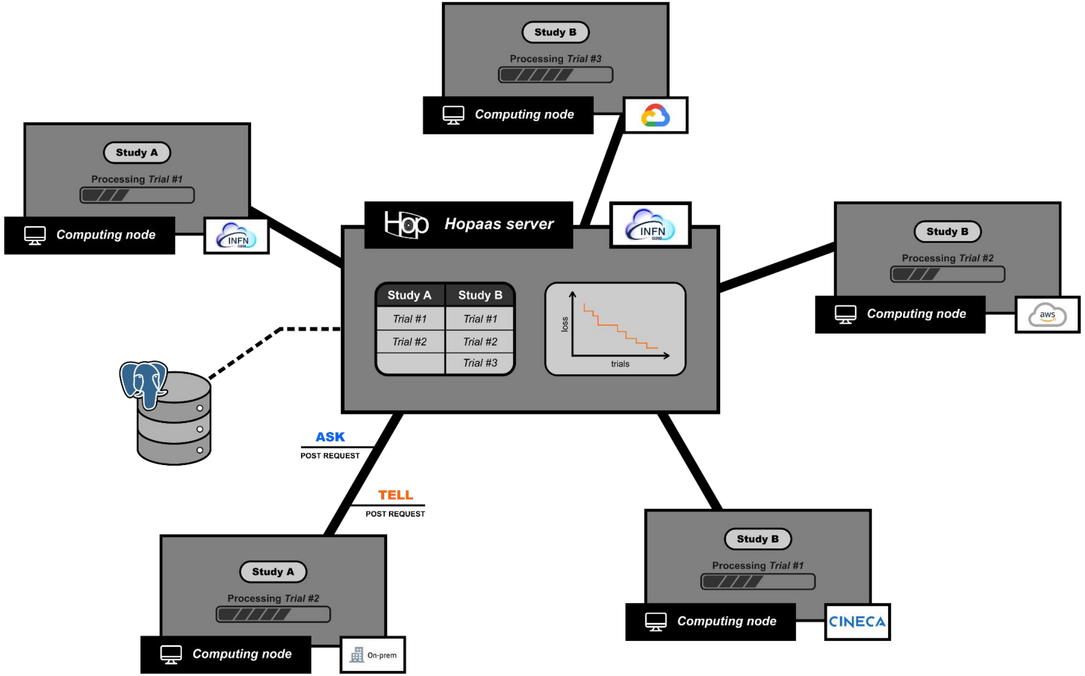


Cluster sequence decoder

Deep Sets architecture

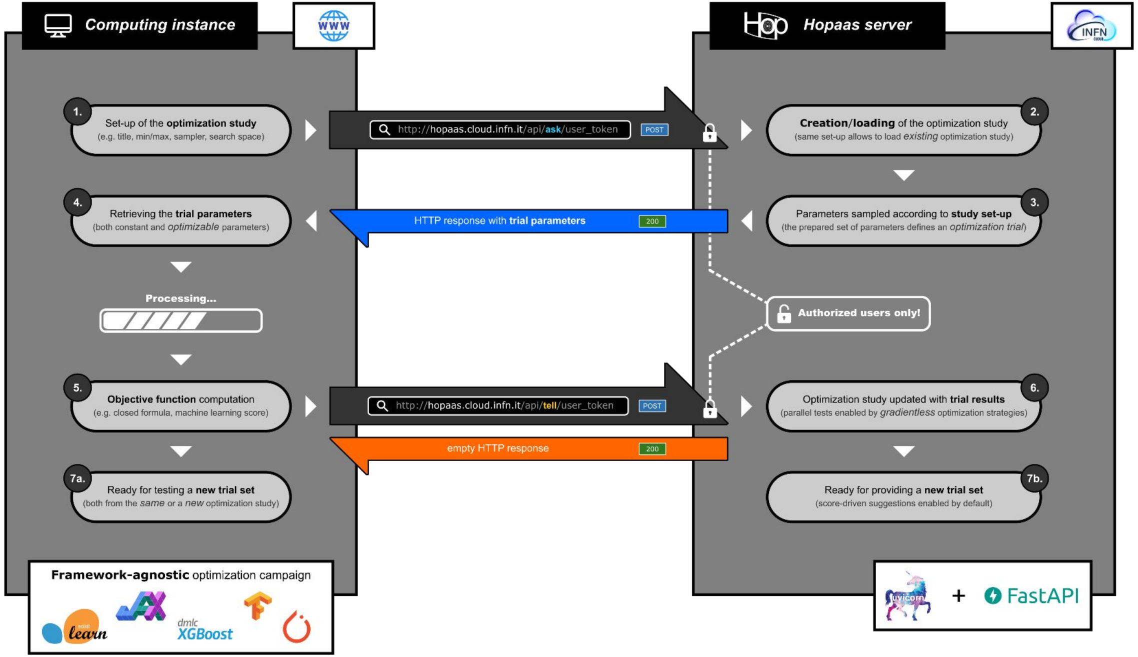


Hopaas: *multi-site optimization campaigns*



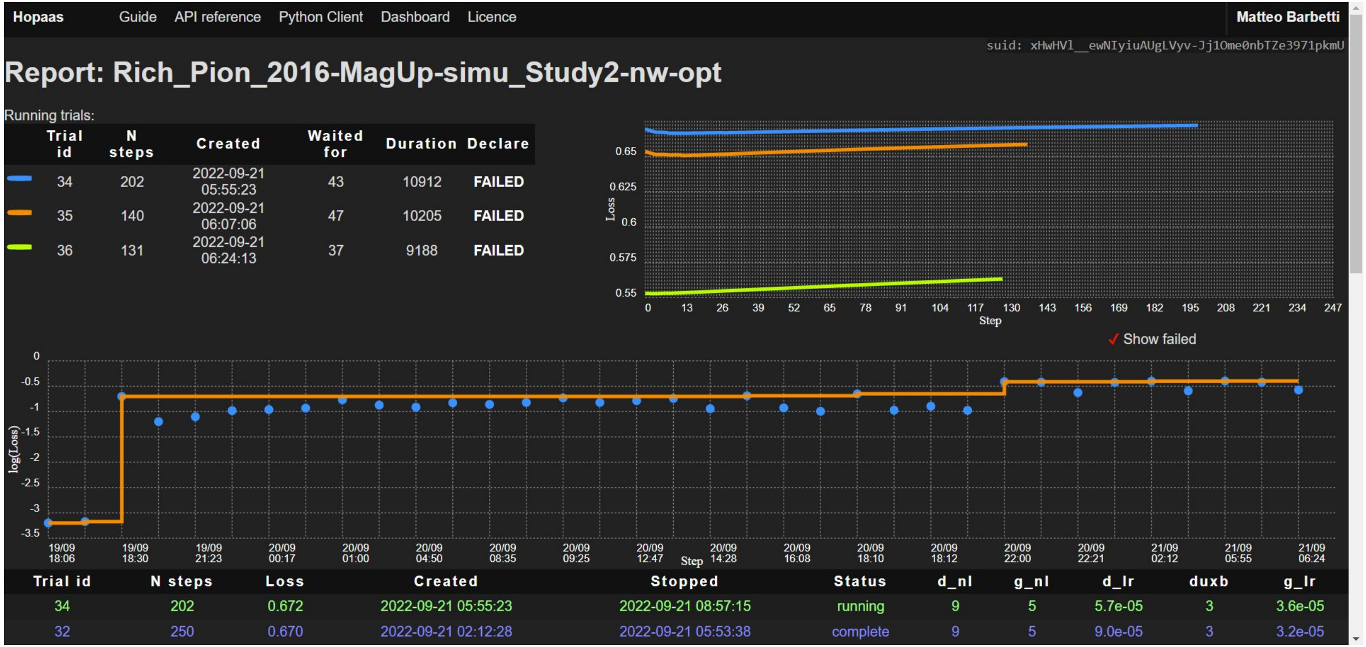
source:
https://hopaas.cloud.infn.it

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
	34	202	2022-09-21 05:55:23	43	10912	FAILED
	35	140	2022-09-21 06:07:06	47	10205	FAILED
	36	131	2022-09-21 06:24:13	37	9188	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