Digital Twins for Nuclear and Particle Physics – **NPTwins 2024**
Museo Diocesano, Genova, 16–18 December 2024

Simulating the LHCb experiment with Machine Learning

Lucio Anderlini, INFN
on behalf of the LHCb Simulation Project

Genova, 16 December 2024

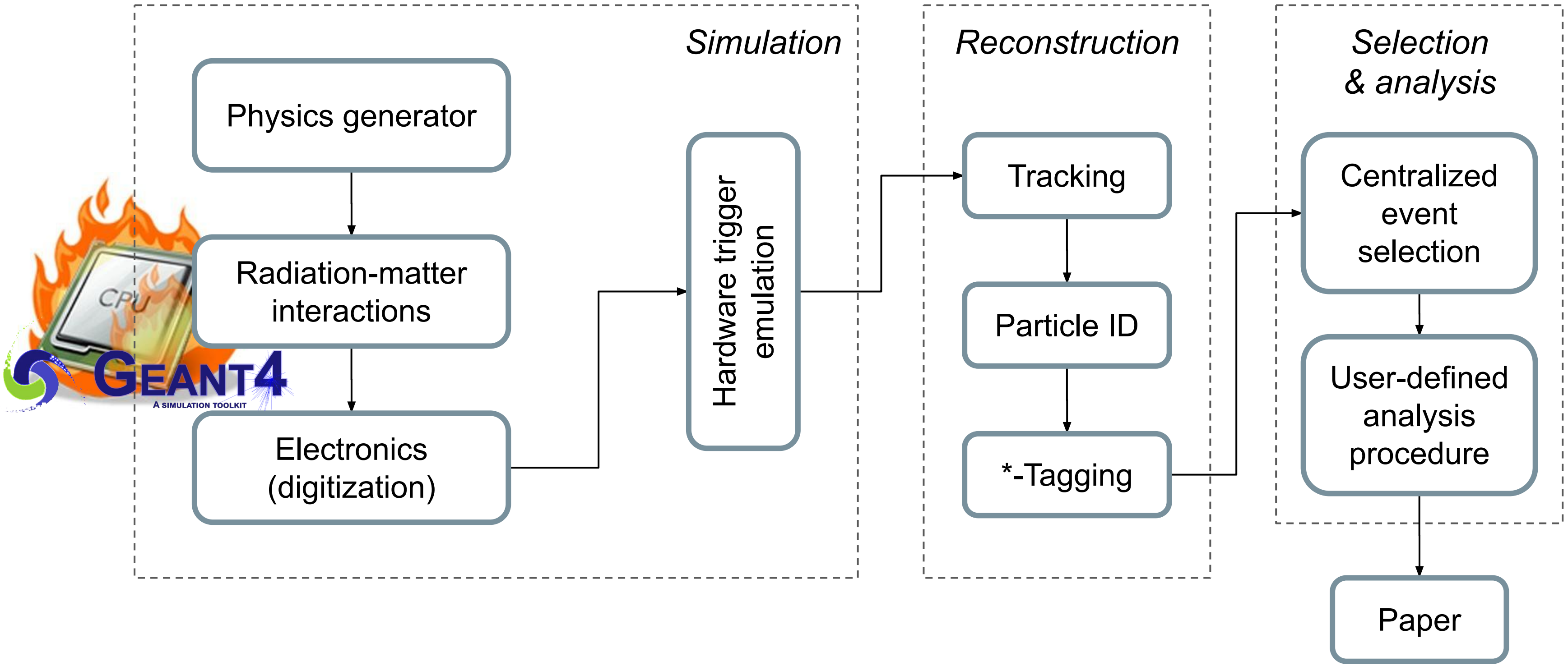
INFN FIRENZE

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

AI_INFN
Artificial Intelligence technologies
for INFN research

LHCb

Generic simulation processing in High Energy Physics



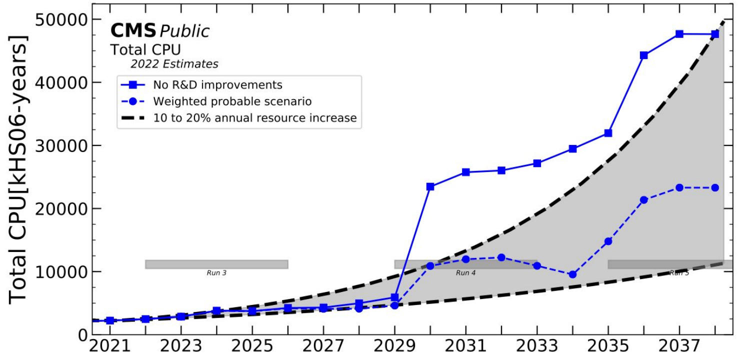
The simulation challenge

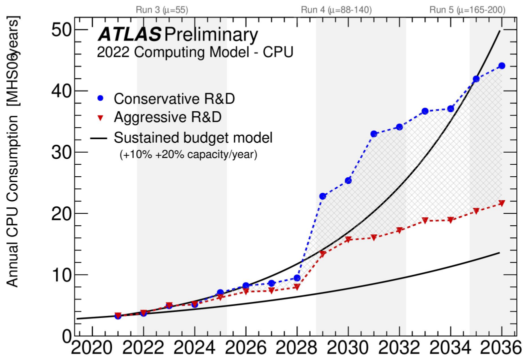
The next-generation experiments aim to operate with a significant increased statistics that will put **severe pressure** on the CPU resources available

In particular, the future request for simulated samples will far **exceed the pledged resources**, even considering an optimistic increasing of the budget intended for computing

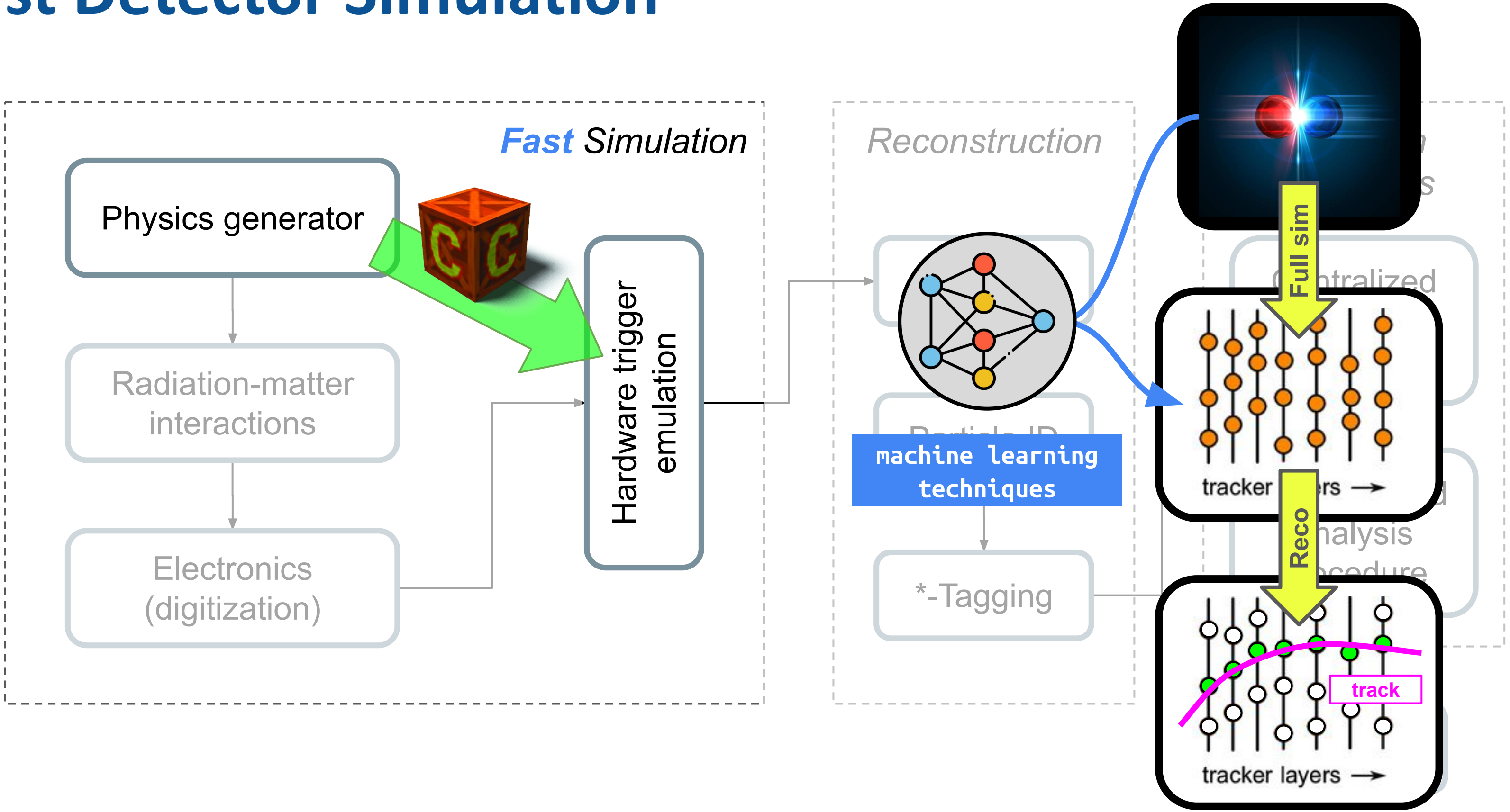
Evolving the simulation technologies is then mandatory to meet the future request for simulated samples:

- the ***Fast Simulation*** paradigm
- the ***Flash Simulation*** paradigm

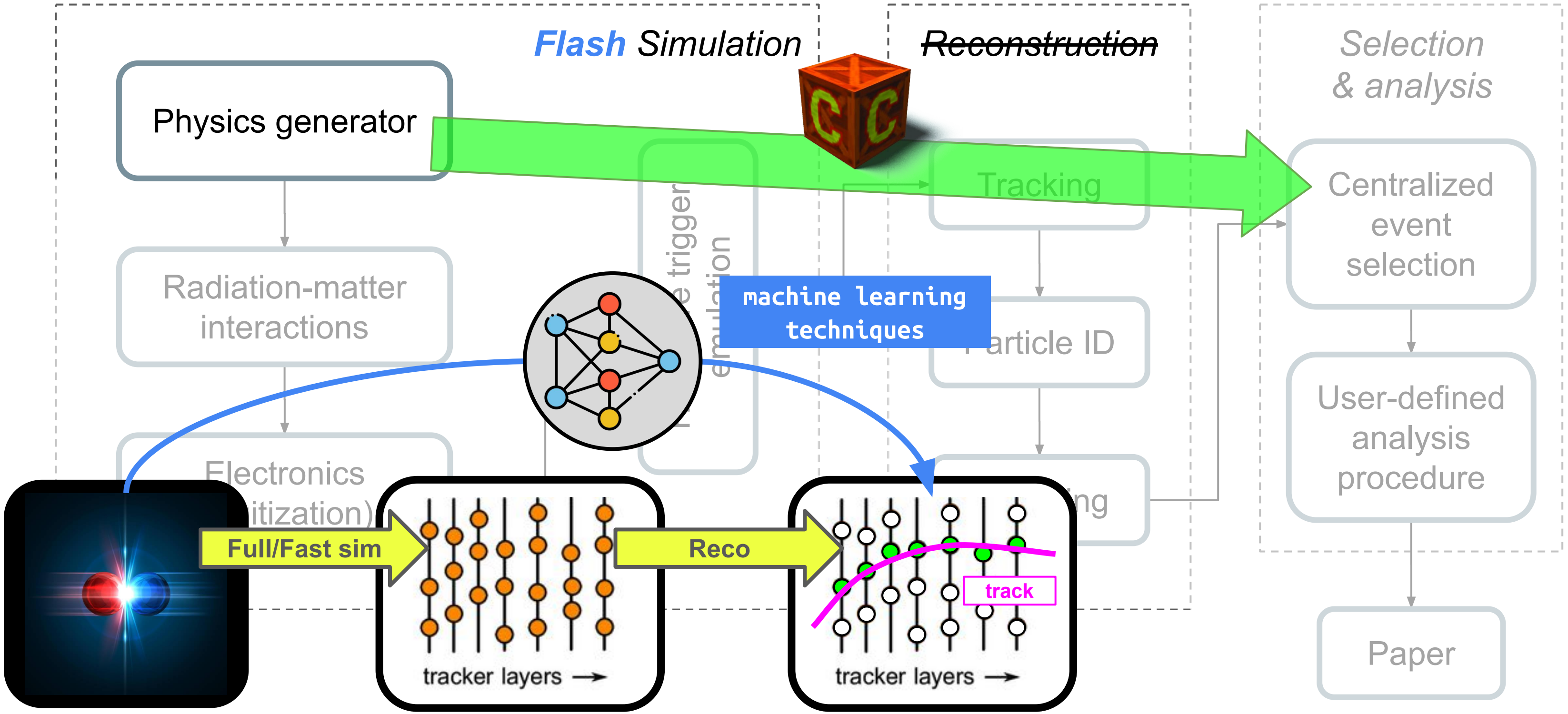




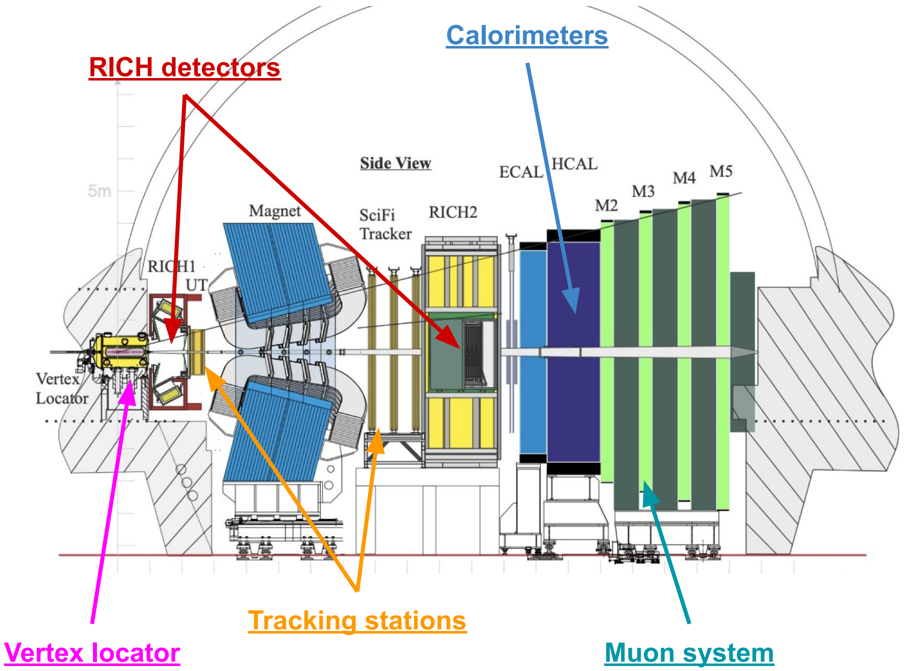
Fast Detector Simulation



Flash detector simulation



The LHCb experiment

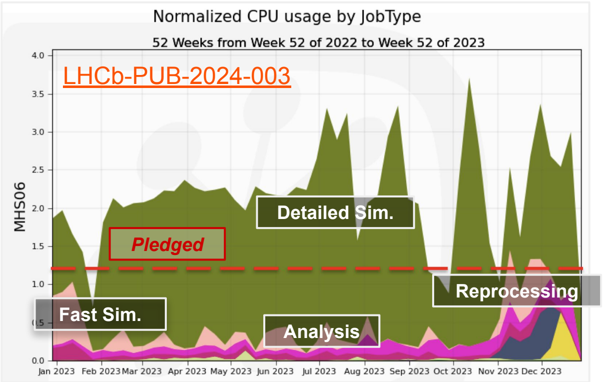


The **LHCb detector** is a single-arm forward spectrometer designed to study particles containing _b and c quarks_.

The **Upgrade I** of the LHCb experiment 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



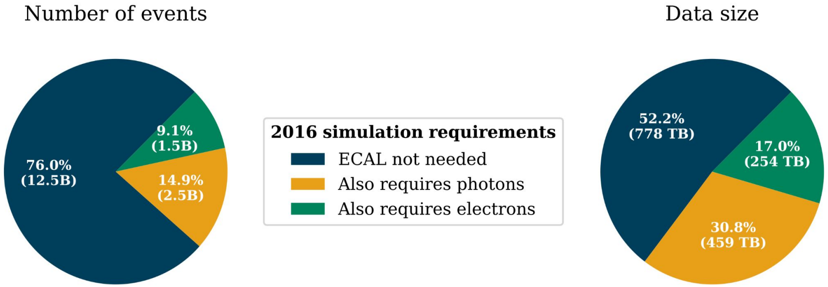
What costs in the LHCb Simulation

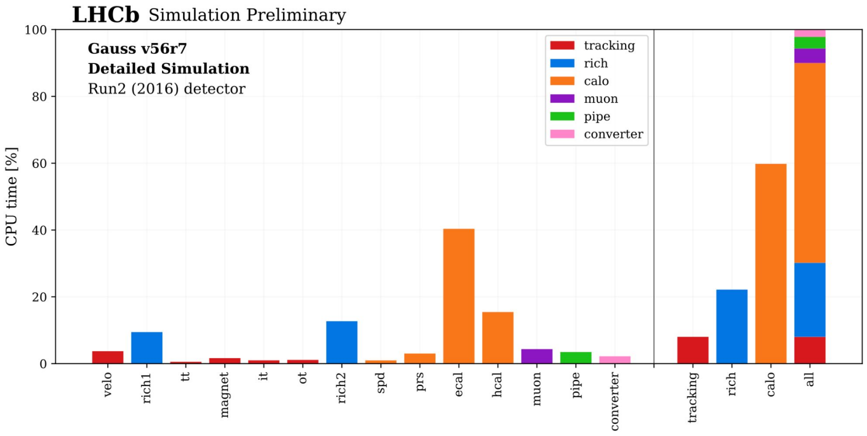
The simulation production is driven by the LHCb physics program, *i.e.* **heavy hadron decays**

- most of the analyses don't require neutral reco in 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





Fast Simulation

CaloChallenge-compatible fast simulation of the LHCb Calorimeter

2022

Geant4 launches an initiative for developing ML models for Calorimeter simulation

[website] [workshop]

2023

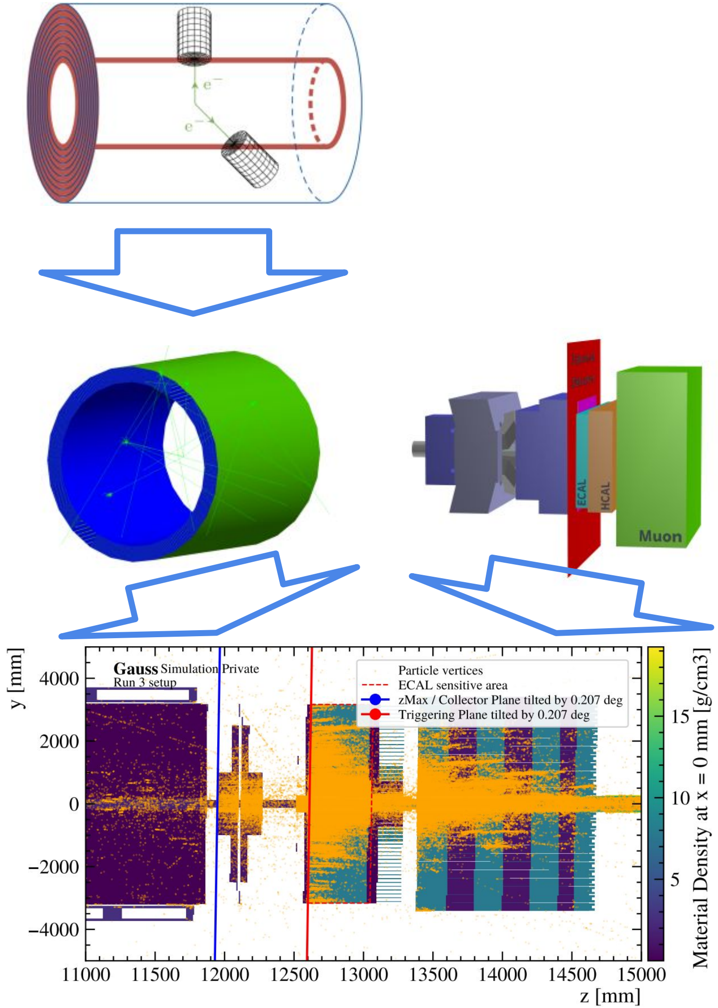
LHCb extends Gaussino to support ML-based parametrizations

[CHEP 2023 slides]

2024

CaloChallenge models integrated in Gaussino and LHCb geometry for training & validation

[ACAT 2024 poster]



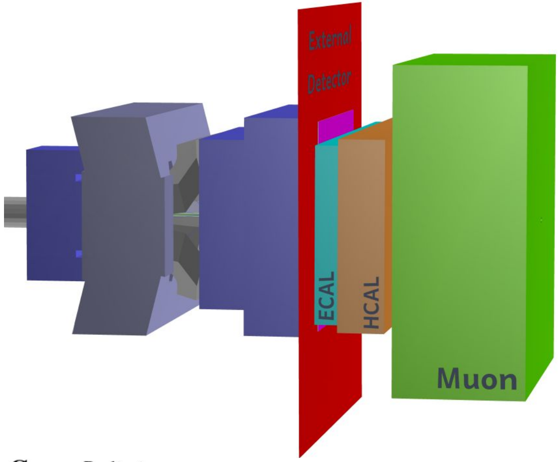
Training

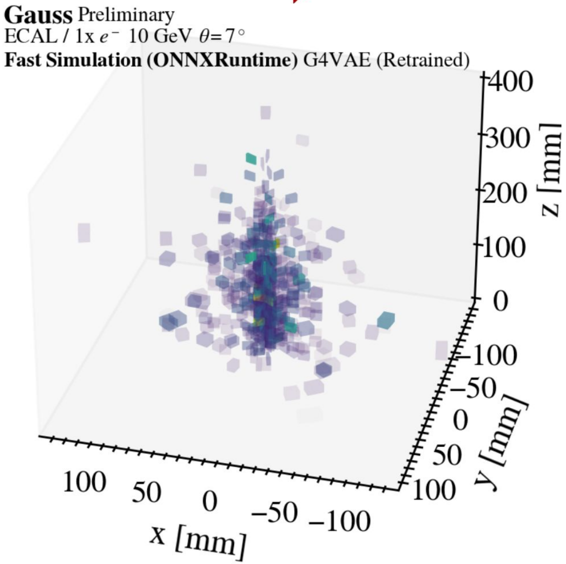
Geometry of the LHCb detector "modified":
→ add a **thin active volume** upstreams the calorimeters.

No effect on the physics:
it collects "***conditions***" for each impinging particle.

Collected conditions are used for **training** a model and at **deployment** time, when the model replaces the calorimeter simulation for *electrons* and *photons*.

Conditions are turned into clusters using a **Variational AutoEncoder** (G4VAE) with custom resampling.



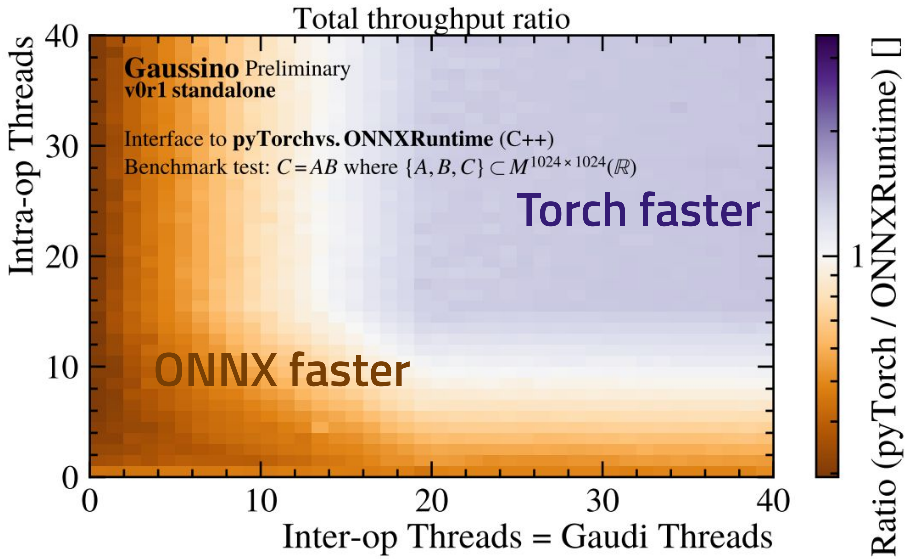


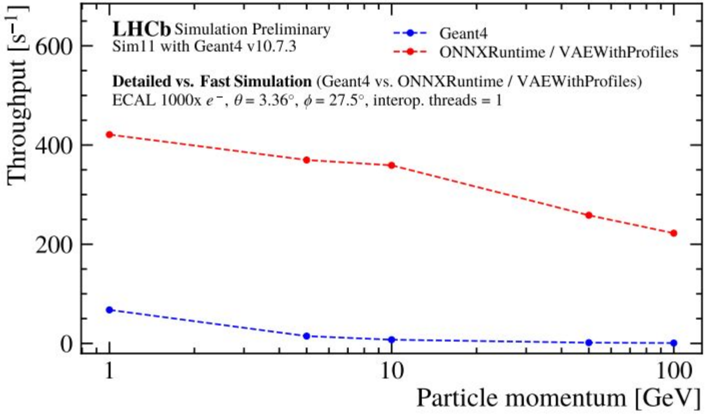
M. Mazurek ACAT 2024 poster

Deployment

torchlib and ONNX Runtimes integrated in Gaussino and performance compared.
ONNX selected for lower thread multiplicity.

Significant throughput improvement over Geant4.

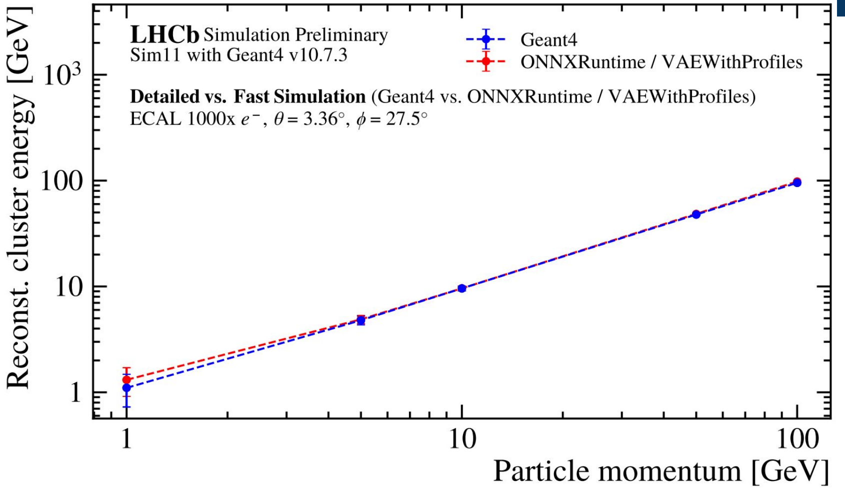


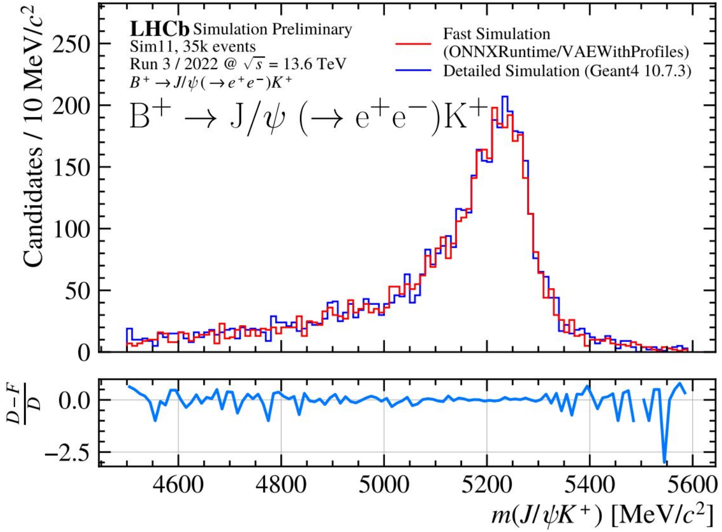


M. Mazurek
ACAT 2024 poster

Validation

- One single model used for both electrons and photons, with a **400× speedup**
- 1−4% accuracy in the **reconstructed energy**
- Verified quality for physics analysis, **simulating *B⁺* decay channels**
- The output data format is identical to **Detailed Simulation**
  → can be used for *e.g.* reco. studies



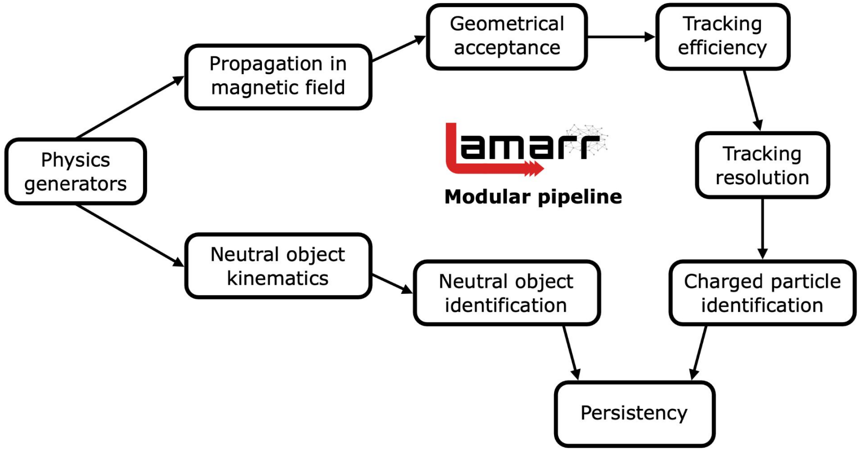


M. Mazurek [ACAT 2024 poster]

Flash Simulation *with Lamarr*

Lamarr: *a pipeline of parametrizations*

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



The Lamarr pipeline can be split in two 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

L. Anderlini, MB, *et al.*, "Lamarr: the ultra-fast simulation option for the LHCb experiment", PoS **ICHEP2022** (2022) 233
LHCb Simulation Project, M. Barbetti, "Lamarr: LHCb ultra-fast simulation based on machine learning models deployed within Gauss", in ACAT 2023, arXiv:2303.11428
LHCb Simulation Project, L. Anderlini, MB, *et al.*, "The LHCb ultra-fast simulation option, Lamarr design and validation", EPJ Web Conf. **295** (2024) 03040, arXiv:2309.13213

Two kind of parametrizations

- Acceptance/Reconstruction/Selection **efficiencies**:
  $\longrightarrow$ *deep neural networks trained as binary classifiers*
- **Reconstructed features** (*e.g. smeared momenta, or PID variables)*
  $\longrightarrow$ *Generative Adversarial Networks (GANs)*

Main design limitation:

Each physics particle corresponds at most to one reconstructed object

- Encapsulate occupancy-effects in the single-particle paramerization
- **Obtain the same reconstructed features using Pythia or Particle Guns**

A spin-off: systematic collection of GAN algorithms

Significant effort devoted to study the rich literature on strategies and best-practices to **regularize the GAN training** and increase the generator descriptive capabilities

M. Barbetti (INFN) developed a Python package designed to simplify the use of GAN algorithms and employ best training practices.

The **PIDGAN** package offers several ready-to-use GAN implementations and state-of-the-art tricks for training.

PIDGAN

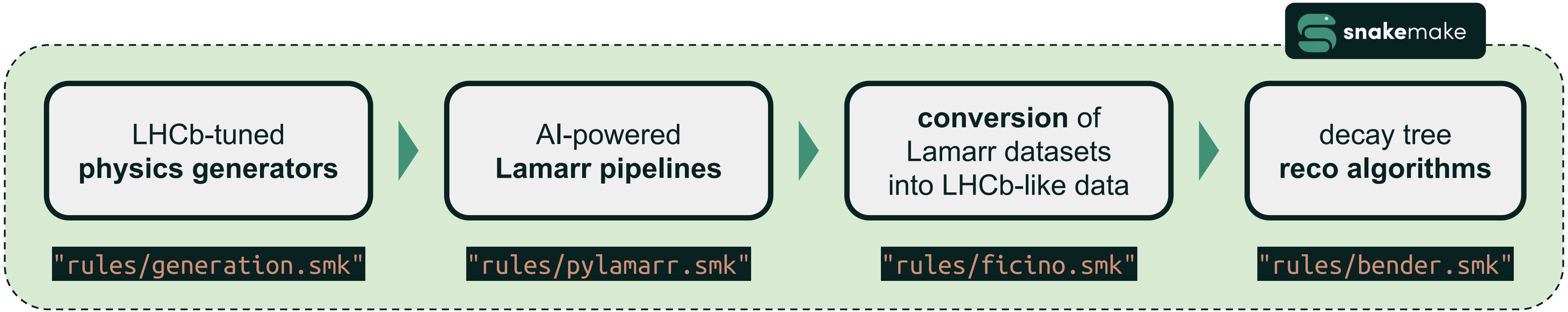
Algorithms*	Source	Avail	Test	Lipschitz**	Refs	Tutorial
GAN	k2 / k3	✅	✅	❌	2, 10, 11	Open in Colab
BceGAN	k2 / k3	✅	✅	❌	4, 10, 11	Open in Colab
LSGAN	k2 / k3	✅	✅	❌	5, 10, 11	Open in Colab
WGAN	k2 / k3	✅	✅	✅	6, 11	Open in Colab
WGAN-GP	k2 / k3	✅	✅	✅	7, 11	Open in Colab
CramerGAN	k2 / k3	✅	✅	✅	8, 11	Open in Colab
WGAN-ALP	k2 / k3	✅	✅	✅	9, 11	Open in Colab
BceGAN-GP	k2 / k3	✅	✅	✅	4, 7, 11	Open in Colab
BceGAN-ALP	k2 / k3	✅	✅	✅	4, 9, 11	Open in Colab

Lamarr workflow

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

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**



Porting the workflow in Cloud

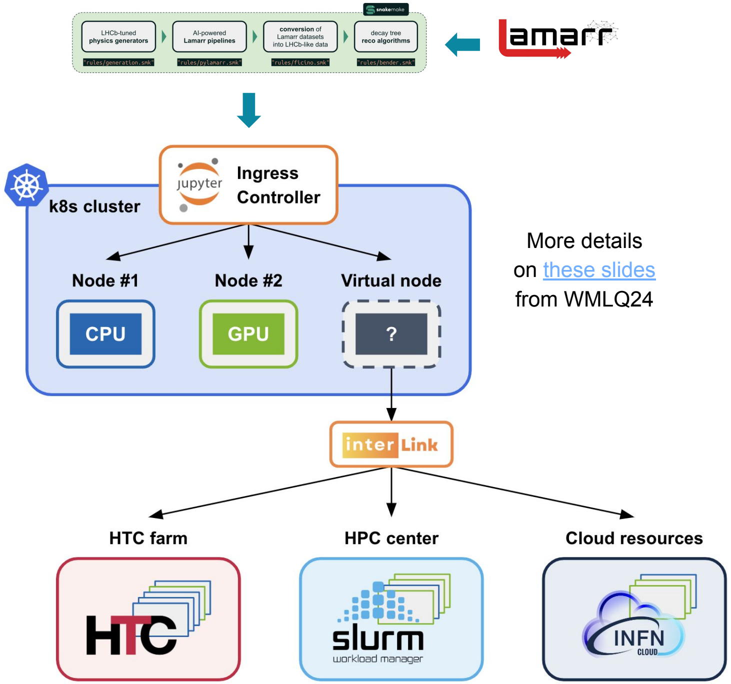
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

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 **multiple different computing sites** scattered across Italy (CNAF-Tier1, CINECA Leonardo, 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**



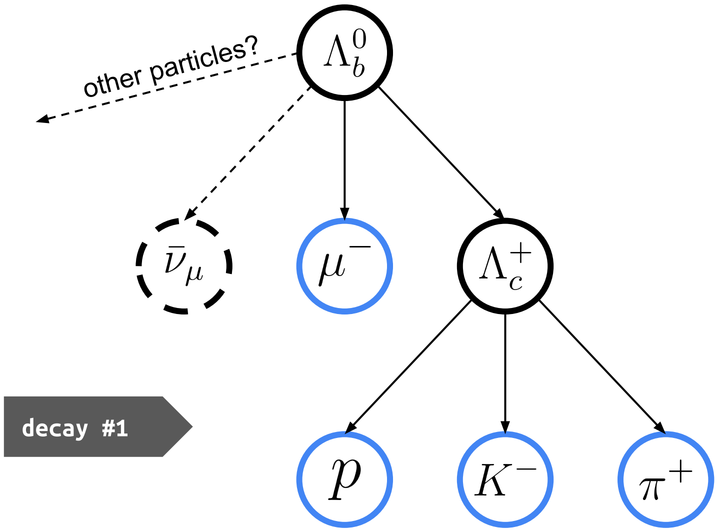
More details on these slides from WMLQ24

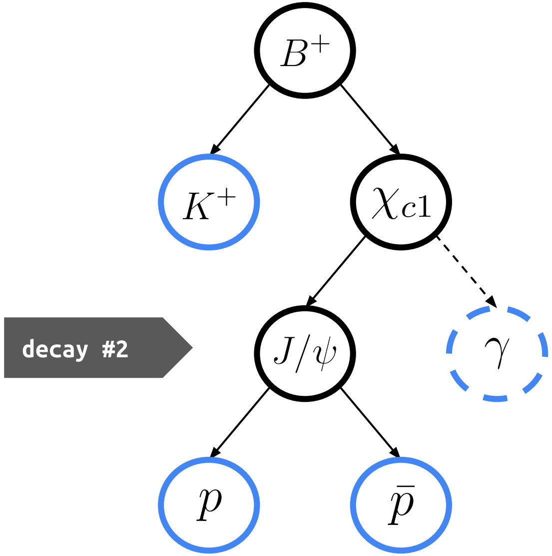
[M. Barbetti, ICHEP 2024]

[M. Barbetti, ICHEP 2024]

Validation

To **test the validity** of both the Lamarr framework and the underlying *flash-simulation* philosophy, several validation campaigns have been performed:





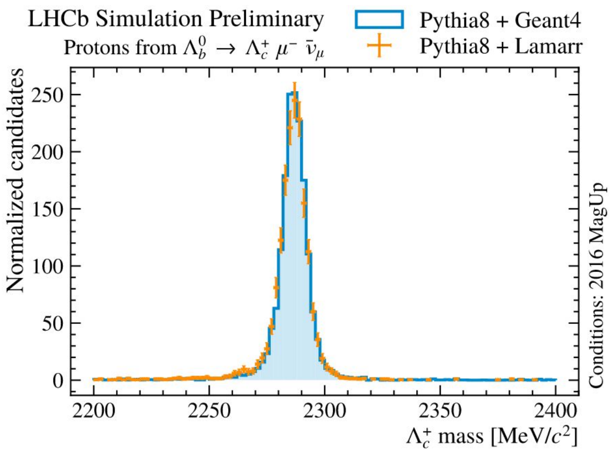
[M. Barbetti, ICHEP 2024]

Results for $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$

The parameterizations and the workflow defined within the Lamarr framework succeeds in **reproducing the errors** introduced in the detection and reconstruction steps by the LHCb experiment. How?

- integration with LHCb-tuned generators → good generated kinematics ✓
- efficiency models → correct "good candidates" selection ✓
- tracking pipeline → smearing effects and reconstruction uncertainties ✓
- PID pipeline → protons and kaons PID variables ✓

NOTE As Detailed Simulation, also flash-simulated datasets can be **further processed** (at decay-level) by using the standard reconstruction algorithms → invariant mass computation



LHCb-FIGURE-2022-014

[M. Barbetti, ICHEP 2024]

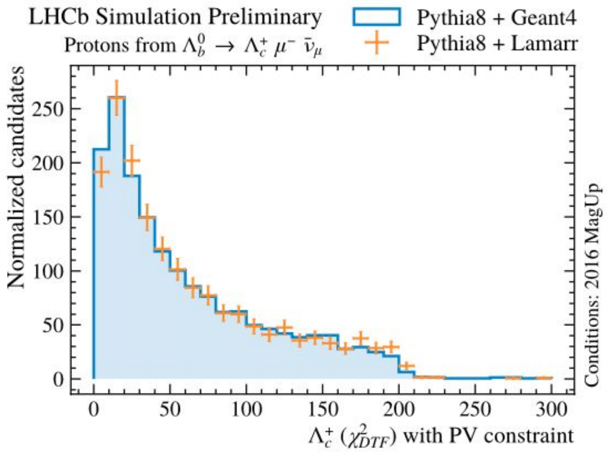
Results for $\Lambda_b^0 \to \Lambda_c^+ \mu^- \bar{\nu}_\mu$

The parameterizations and the workflow defined within the Lamarr

DECAY TREE FITTER

The `DecayTreeFitter` (DTF) **algorithm** defines a least squares fit that extracts all parameters in a decay chain simultaneously. It allows to correct the momenta of the final state particles to account for the constraints (e.g., PV, masses) of the decay chain of interest.

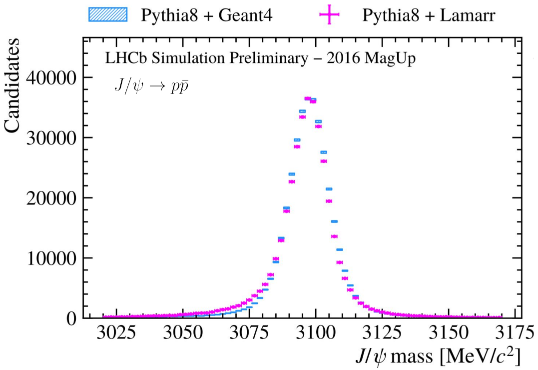
NOTE As Detailed Simulation, also flash-simulated datasets can be **further processed** (at decay-level) by using the standard reconstruction algorithms → DTF χ^2 computation

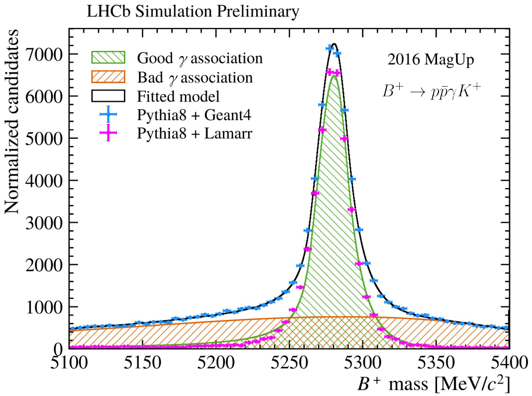


LHCb-FIGURE-2022-014

[M. Barbetti, ICHEP 2024]

Results for photons in $B^+ \to p\bar{p}\gamma K^+$





The parameterizations and the workflow defined within the Lamarr framework succeeds in **reproducing the errors** introduced in the detection and reconstruction steps by the LHCb experiment. How?

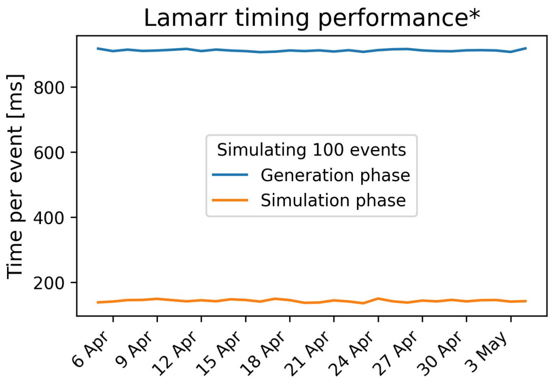
- integration with LHCb-tuned generators → good generated kinematics ✓
- efficiency models → correct "good candidates" selection ✓
- tracking pipeline → smearing effects and reconstruction uncertainties ✓
- PID pipeline → protons and kaons PID variables ✓
- ECAL pipeline → "efficiency" and smearing effects ≈

NOTE As Detailed Simulation, also flash-simulated datasets can be **further processed** (at decay-level) by using the standard reconstruction algorithms → invariant mass computation

[M. Barbetti, ICHEP 2024]

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**
- The generation of *b*-baryons is expensive → Pythia8 is the new **major CPU consumer**
- The Particle Gun approach drops to **almost zero** the cost of the Generation phase → (PGun + Lamarr) allows to reach **three-order-of-magnitude** speed-up



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

Detailed simulation: Pythia8 + Geant4
1M events @ 2.5 kHS06.s/event ≈ 80 HS06.y

Flash simulation: Pythia8 + Lamarr
1M events @ 0.5 kHS06.s/event ≈ 15 HS06.y

Flash simulation: Particle Gun + Lamarr
1M events @ 1 HS06.s/event ≈ 0.04 HS06.y

Cost for **1B events** production?

~ 800000 €

~ 150000 €

~ 400 €

Conclusion

Simulation cost is increasing to an unsustainable level.

LHCb simulation stack is **similar** to other LHC → benefits from CaloChallenge
LHCb usage of simulation is **different**: most simulated samples are analysis-specific
→ Flash Simulation (**Lamarr**) may apply to a large fraction of analyses.

Successful effort on the Machine Learning algorithms → it is now the "easy bit"
Active research on
→ Modeling particle-to-particle
→ Predicting uncertainties of GANs

Integration in the experiment software stack is the challenge:
most proof-on-concepts have successeeded → effort needed to reach production