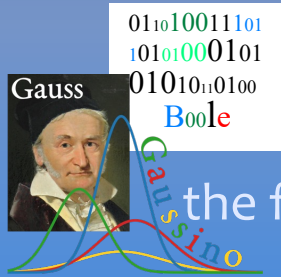


Community input
on the European Strategy for Particle Physics Update
25 – 27 March 2024



LHCb View

the future of simulation and simulation for the future

Gloria Corti, CERN

Michal Kreps, University of Warwick

Mark Whitehead, University of Glasgow



How will we do simulation in 2030+

LHCb U2 data taking is Run5



$$\begin{aligned} \text{Run3 } \mathcal{L}_{\text{leveled (fixed)}} &= 2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1} \\ \text{Run5 } \mathcal{L}_{\text{leveled (peak)}} &= 1.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1} \end{aligned}$$

What will we need simulation for ?



- Finalize detector and data processing
- Detector commissioning
- [Prepare for] Physics analysis
 - *Different needs in term of physics modeling accuracy, statistics and timescale turn-around*
 - *Different requirements for the varied LHCb U2 physics program*
 - *s, c, b physics, [B]SM and spectroscopy*
 - *QEE*
 - *Heavy Ion*
 - *Fixed Target*

Provide the **underlying simulation framework** to **explore different solutions** and promote their seamless **integration**, while continuing to support the **immediate needs of the experiment**

LHCb Simulation

Facilitate the use, validation and tuning of new features in the LHCb simulation

Integration of new technologies in full experimental software and computing infrastructure

HEP

Common software, e.g. Geant4 optimization, hooks for ML

Prototyping of new technologies with stand-alone sample use cases



LHCb simulation landscape today

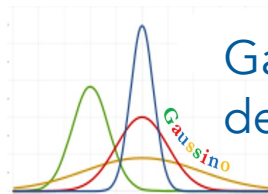
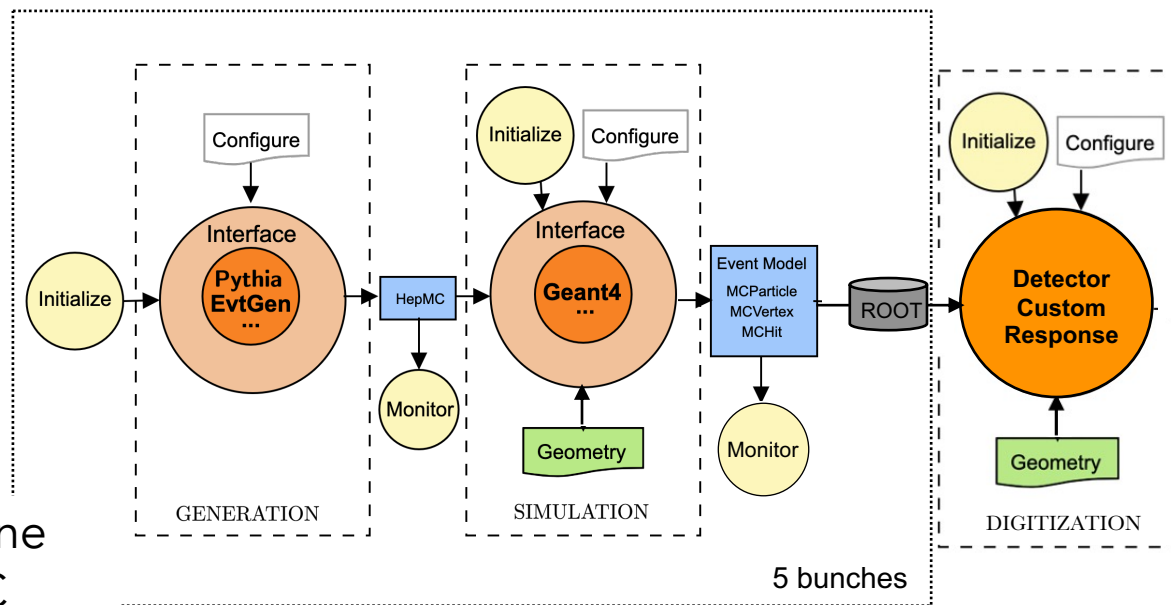
Pythia + EvtGen are the LHCb workhorses

We use other generators for HI, double heavy baryons, CEP, EW, Higgs, ...

Geant4 is the corner stone of simulation for the LHC

We have **Fast** and **Flash** simulations under development

We heavily rely on **fast simulation techniques**



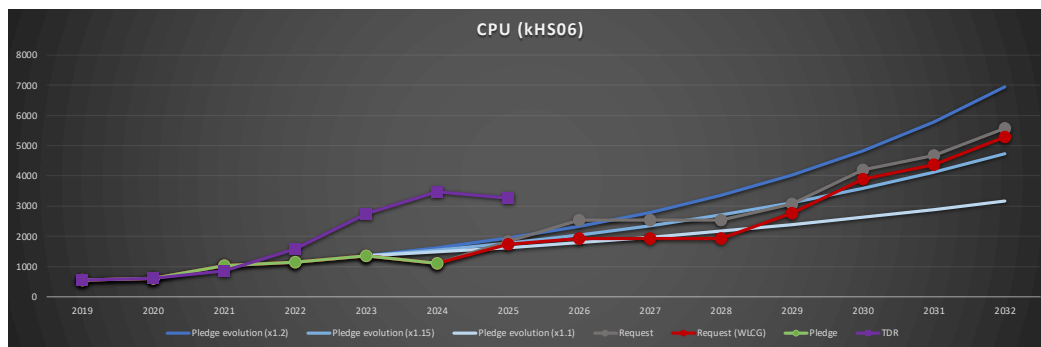
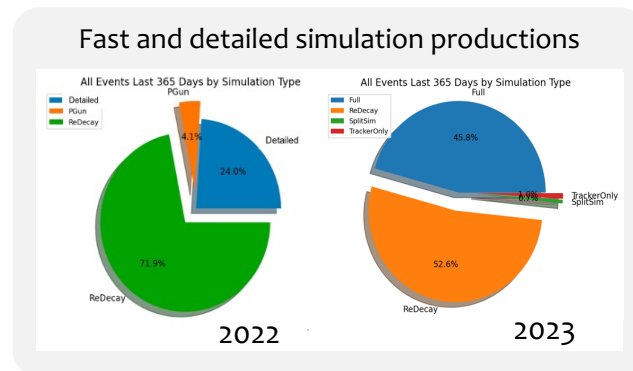
Gaussino is the ideal test-bed for new developments

minimal functionality in stand-alone mode

Monte Carlo productions dominates ($\geq 90\%$) the LHCb distributed computing CPU time

- Run everywhere
- Run continuously
- No input data
- Detector simulation dominates in most cases

... BUT for some type of events the generation is more time-consuming due to aggressive filtering and how we extract signal from minimum bias



Forecast of CPU resources for Run3 & Run4, c.f. C. Bozzi

A complex distributed ecosystem with **DIRAC** as LHCb standard for **workload** and **data** management

Can use pledged and opportunistic resources

- **Access to HPCs** will expand
- Support for **non-x86** architectures
 - ARM, GPUs ...

Until the start of Run3 heavily exploited the LHCb CPU online farm outside data-taking periods, i.e. LS, [YE]TS and occasional downtimes

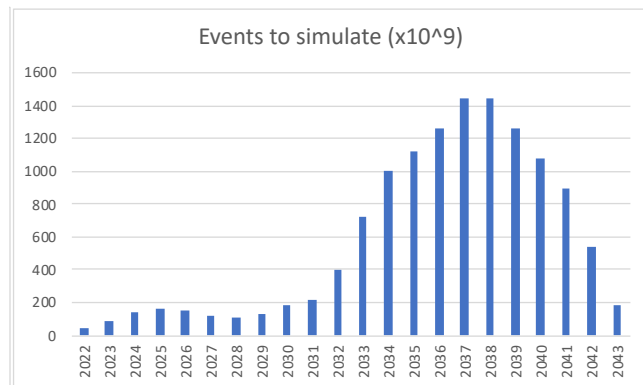
In Run3 we are using the LHCb HLT2 farm the same way, but currently we cannot use the HLT1 GPU farm for simulation

Sustainability has emerged as a major concern and may dictate type of resources available

this has a human resources cost that needs to be taken into account

More beam data requires **more simulated data**

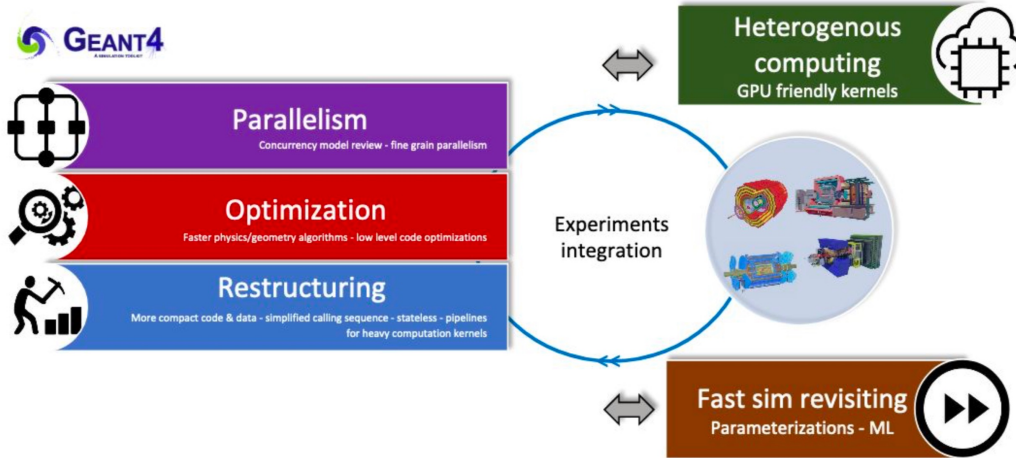
- Simulation scales with the event rate
- Extreme pressure on the computing budget
- How we run today will not be affordable in Run5



U2 Workshop May 2021, C. Bozzi

*Simulations need to be **faster**,
without sacrificing relevant
physics accuracy*

- Refactoring and internal improvements
 - Optimisation of current Geant4 code to run faster
 - Mostly work internal to Geant4

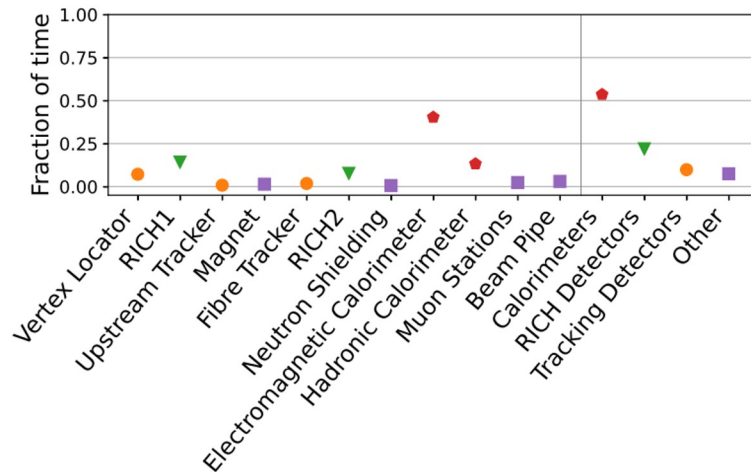


All aspects in **Geant4 R&D** activities
Combined with physics improvements

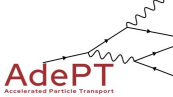

HL-LHC Computing Review: Common Tools and Community Software

- Hardware (R)Evolution
 - GPUs are more and more available
 - AdEPT and Celeritas R&D is maturing nicely
 - How can we effectively use them for a complete detector simulation?
 - What about NPUs, TPUs?
- Fast and Flash Simulation
 - Replace detailed particle tracking models with different methods
 - Long tradition of parametric response implementations
 - Machine Learning is the cool kid on the block

- In LHCb two processes are responsible for most of the CPU usage in the detector simulation:
 - Electromagnetic showers inside the calorimeters
 - Optical photons transportation in the RICH detectors

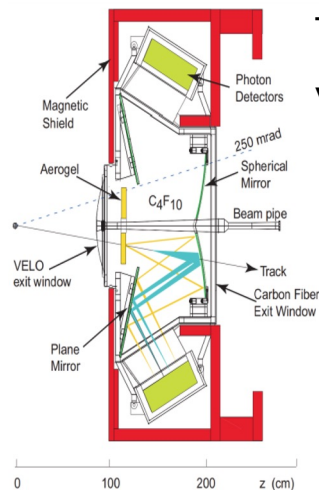


Investigate a hybrid workflow scheme with the use of GPUs to tackle the issue for EM showers and optical photons leveraging on HEP R&D

- What level of changes are required for experiment's production frameworks using GPUs toolkit?
- Both  and  to be tried out in Gaussino
- Necessary 'glue code' in Gaussino developed to compare AdePT with Geant4 using CaloChallenge geometry exploited for ML
 - Work ongoing in Gauss-on-Gsino to understand it for the full LHCb needs
- Fruitful synergy between R&D projects and LHCb
 - provide feedback and requirements to AdePT [and Celeritas teams], e.g. may use selective offload as speed up depends on particle multiplicity, missing MC truth,...
- Need to understand how to cope with an heterogenous production framework in a production environment!

- OPTICKS, that provides an interface between Geant4 and the NVIDIA OptiX™ ray tracing engine to simulate photon propagation while maintaining the simulation of other particles on CPU

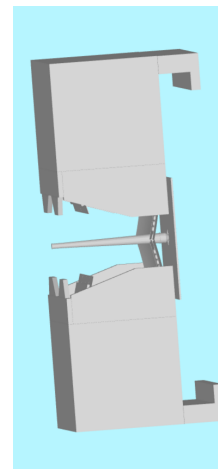
- Mitsuba open-source physics-based renderer to simulate optical photons as an **alternative**



Test with a simplified version of LHCb RICH1

Integration of OPTICKS into Gaussino is challenging

Issue with synch with NVIDIA new versions



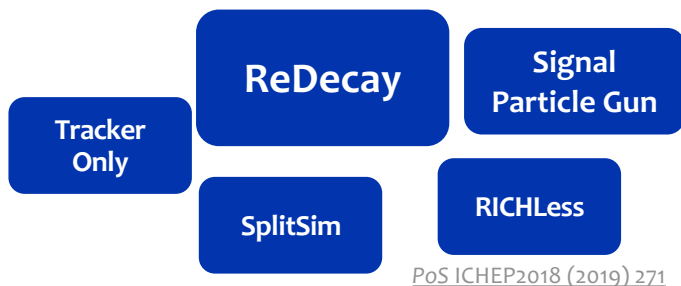
RICH1 geometry in Mitsuba

Apply to RICH Cherenkov photons in test setup

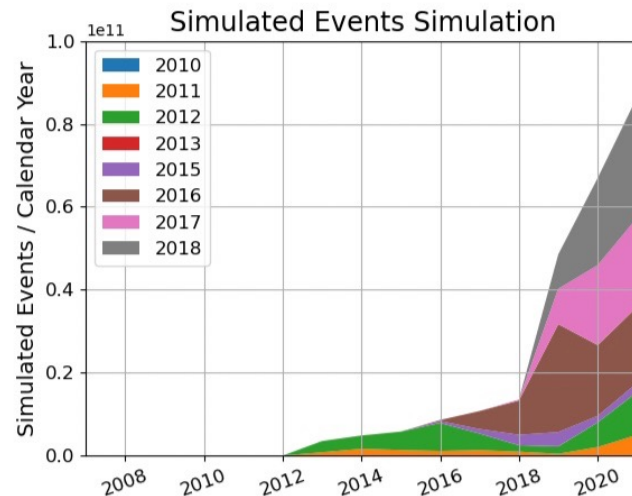
Development within UK cross-experiment SWIFT-HEP project and LHCb Upgrade 2 project

Fast simulation techniques

- LHCb has already been quite successful in producing factors more events without a corresponding increase in computing resources



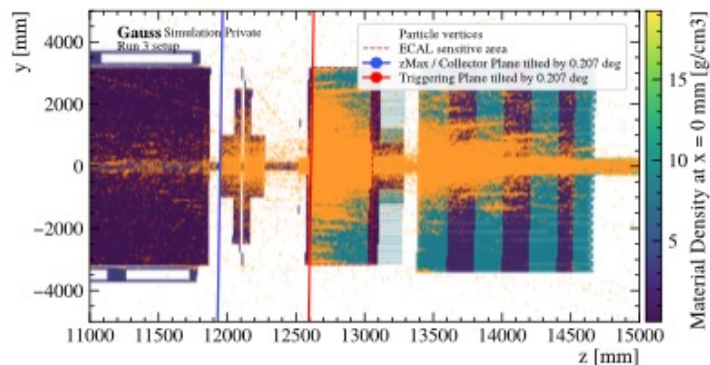
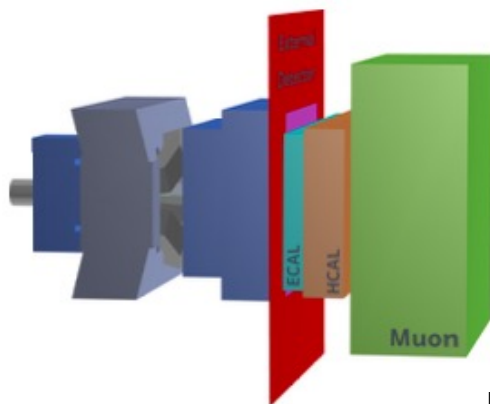
We should continue to exploit and explore new techniques as they best fit the physics



Year	Simulated events (10^9)	Stored events (10^9)	Ratio	CPU work kHS06.y	CPU per event kHS06.s	LFS TB
2017	10.3	4.2	40.3%	817	2.50	640
2018	12.0	3.0	25.3%	1009	2.65	550
2019	45.0	6.9	15.2%	1290	0.90	1110
2020	67.0	16.8	31.7%	1357	0.81	2010
2021	80.0	11.1	13.9%	1815	0.72	2030

Optimise turn around – Fast simulations

- Parameterize the detector low-level response without relying on Geant4 and exploiting the Gaussino custom physics infrastructure
 - Point library for Calorimeters energy deposits
 - Generative Models (e.g., GAN, VAE) for Calorimeters energy deposits



Example from CaloChallenge exercise,

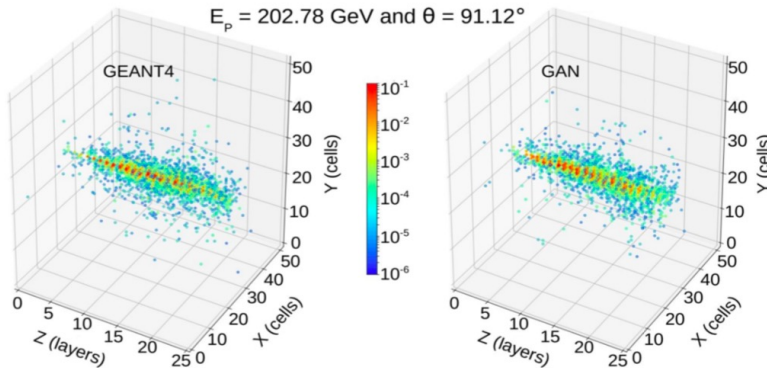
- The issue is to choose best model ahead and train it

New challenges for machine learning in fast simulation

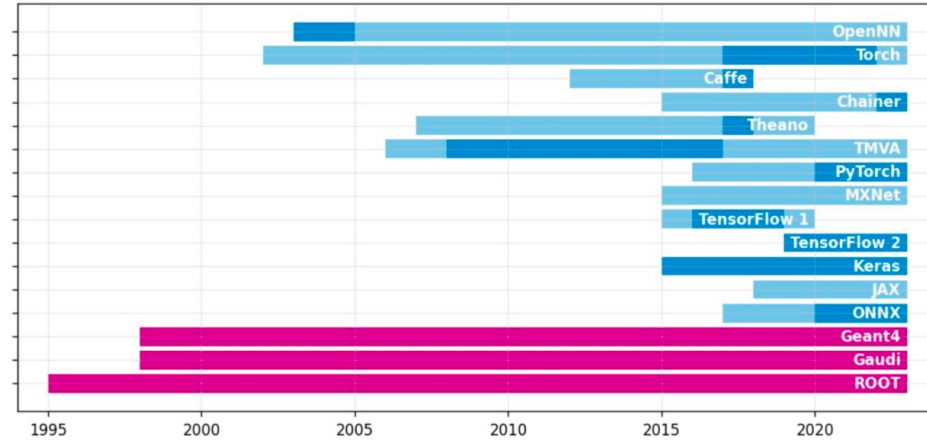
- Simulate time-evolution of ECAL showers

Showers develop in $O(ns)$, becoming relevant for $O(ps)$ -resolution timing detectors.

Recently succeeded training 3D GANs for full spatial correlations [[Khattak \(2021\)](#)]. 4D was never attempted.



L. Anderini



- The deployment of these models in our framework can hardly be achieved without a **solid integration of third-party machine learning software.**

Which one? It's a bet.

Exploit common R&D and new initiatives for ML

Community driven idea: CaloChallenge

ÖAW

The Fast Calorimeter Simulation Challenge 2022
— ML4jets at DESY Hamburg, Germany —

Claudius Krause
Institute of High Energy Physics (IHEP), Austrian Academy of Sciences (OAW)
November 9, 2023

Facco Giordelli, Gergo Karasik, Ben Nachman, Dhruv Iskanter, David Witt, and Anna Zabonova
<https://calochallenge.github.io/homepage/>

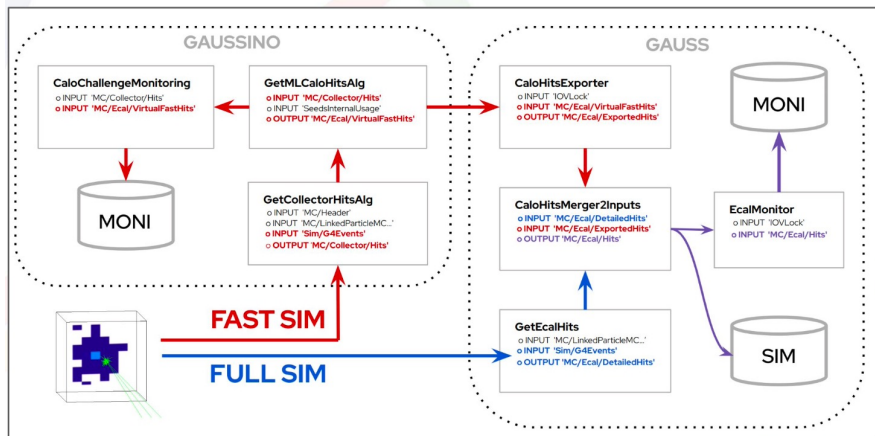
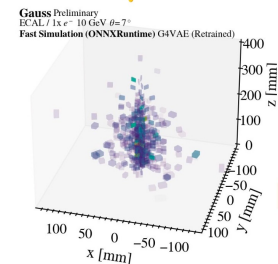
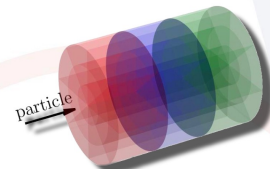
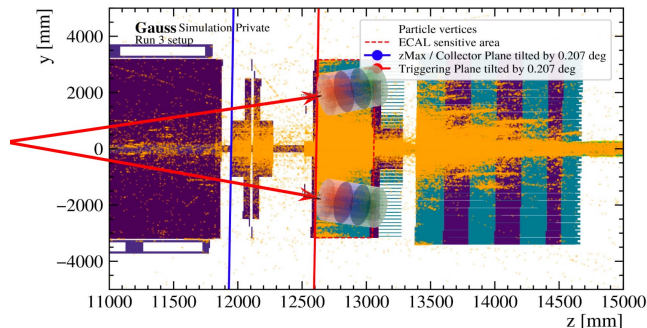
Build a cylinder of virtual hits around a particle shower

3 goals

- Compare models objectively
- Adapt easily by retraining
- Easy conversion to the target geometry

ML4jets 2023

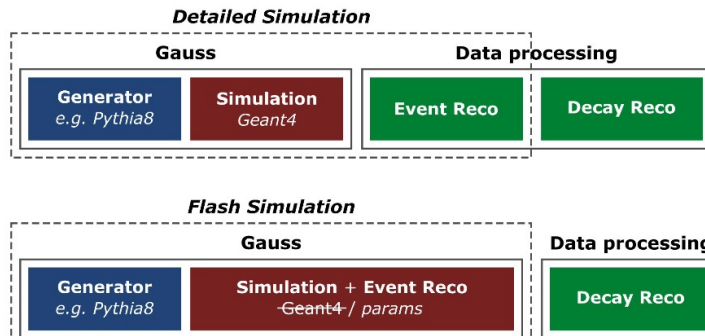
Generic showers in LHCb



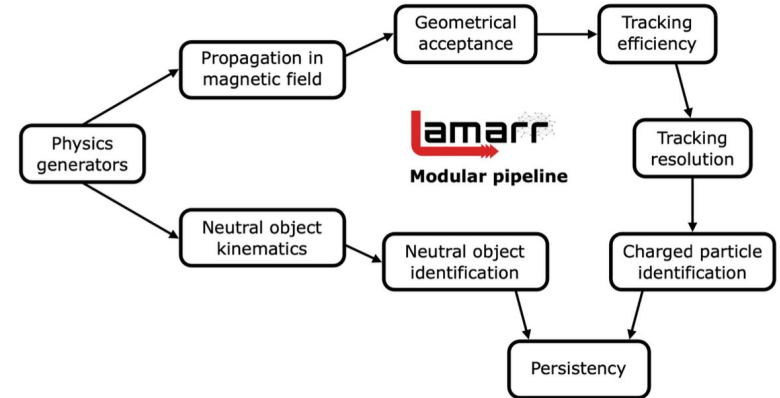
- Interplay with ML expert developing new models
- Evaluate models easily
- Maintenance of models
- Training frameworks and ML libraries
- Profit from proposed AI initiatives

The more radical approach of Flash simulation

- Replace Geant4 and reconstruction with parametrizations able to directly transform generator quantities into analysis-level reconstructed object



Lamarr is the LHCb flash simulation



it consists of a pipeline of (ML-based) modular parameterizations with two separate branches for charged and neutral particles

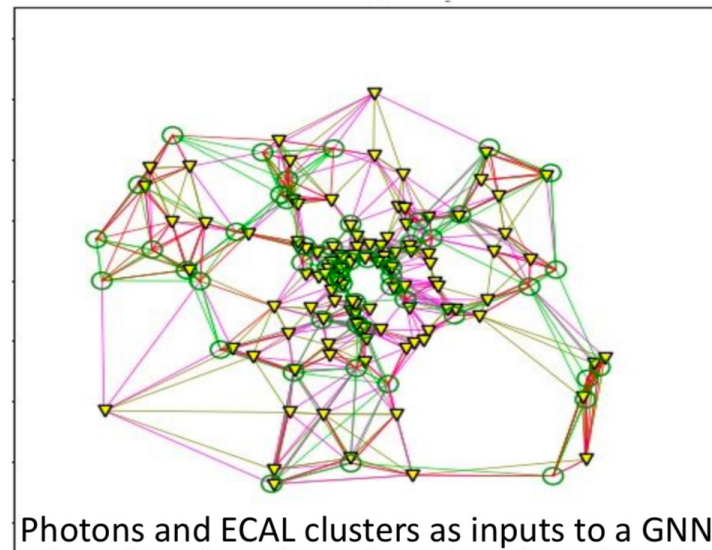
By relying on a containerised Snakemake-based workflow the Lamarr pipeline was successfully validated on three different and combined Cloud computing sites scattered across Italy

Ultra-fast (or *parametric*) simulation, today relies on the assumption that **each particle can be processed separately**, accounting *statistically* for the rest of the event.

Treating particle-to-particle correlations:

- will become critical with higher multiplicities
- will extend the application of ultra-fast simulation to more applications

Modern architectures such as **GNNs** and **Transformers** provide tools for describing complex relations between *spatially-correlated objects*, and applications in HEP are just dawning.



Several challenges have emerged, already, involving many different aspects

- Support for **multiple Event Models** for reconstructed quantities
- Interactions between **Python and C++ frameworks** used for training and deployment, respectively;
- Assess **uncertainties** arising from the adoption of generative models
- Access to **important, distributed resources** for training the models, extremely difficult to provide and account as part of WLCG pledges
- Overall, a novelty for the HEP field, creating new opportunities of **inter-experiment collaborations**.

- The simulation framework is the key
 - Use Gaussino as a test bed
- Ideas of
 - Mix of fast / ultra-fast / detailed simulation for different collisions in same 'event'
 - Simulation of partial detector
 - Simulate only events interesting, à la SplitSim.... More complicated if interest is based on reconstructed quantities
- Docker container to exploit clouds and HPC...

- We need to invest in R&D and be open to new strategies
 - Exploring different “dimensions” of simulation
 - Exploit heterogeneous architectures
- Need combination of deep software expertise and physics knowledge
- Flexibility is key
- Try out new idea, learn from experience, use early, share knowledge and developments