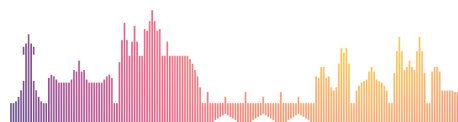
Fast Simulation at LHCb

Adam Morris, on behalf of the LHCb Collaboration

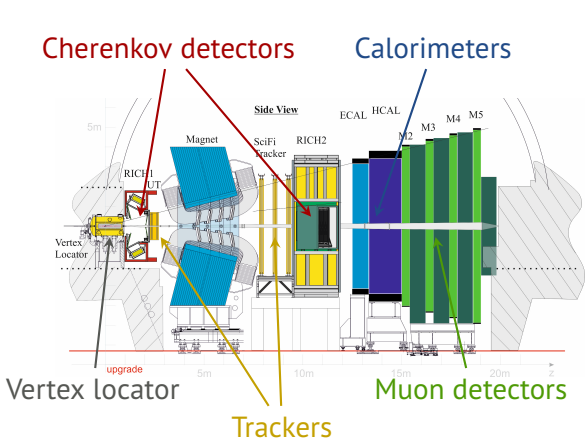
HISKP, University of Bonn

40th International Conference on High Energy Physics
Prague, 29th July 2020

UNIVERSITÄT BONN

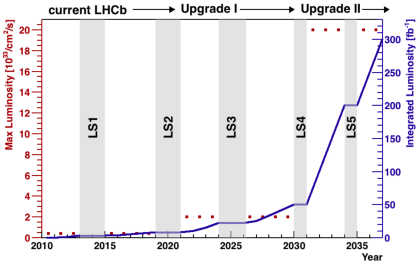


LHCb and its upgrade



Detector paper: [JINST 3 (2008) S08005]

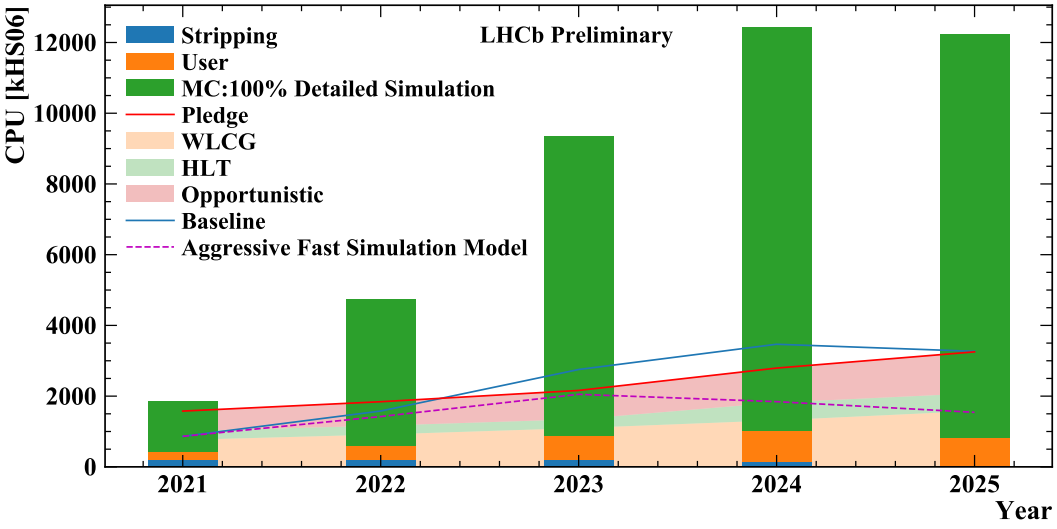
[Talk on upgrade by F. Alessio]



[LHCb-PUB-2018-009]

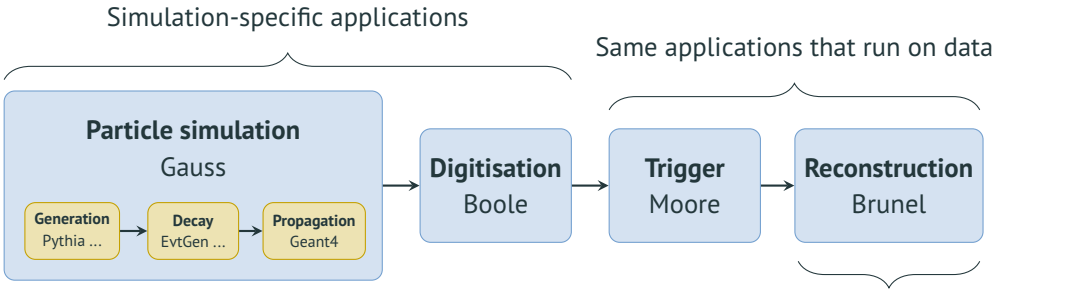
- Already a need for large MC samples for certain Run1+2 analyses.
- Increased luminosity in Run 3: more numerous and complex events.

Computing resource projection



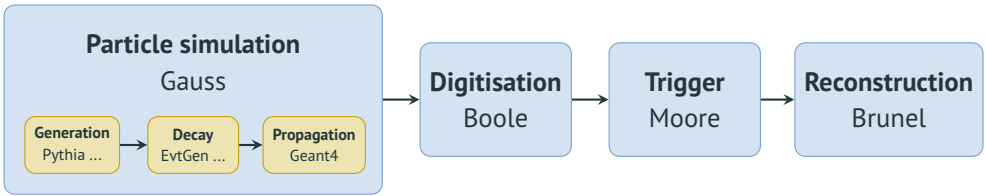
[LHCb-FIGURE-2019-018]

Simulation overview



[Talk by R. Quagliani]

Simulation overview



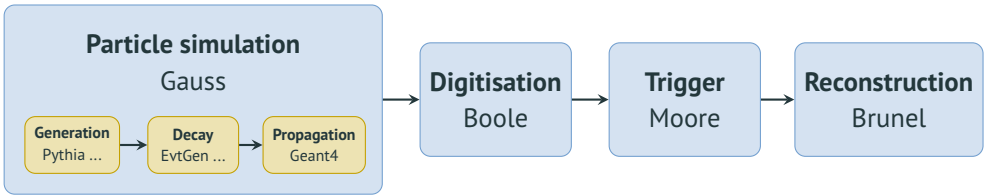
95 – 99% of CPU time for whole chain spent in Geant4

How can we speed this up?

- Code optimisation
- Multithreading
- ...

Simulation overview

Selective simulation
(ReDecay, PGun, SplitSim)

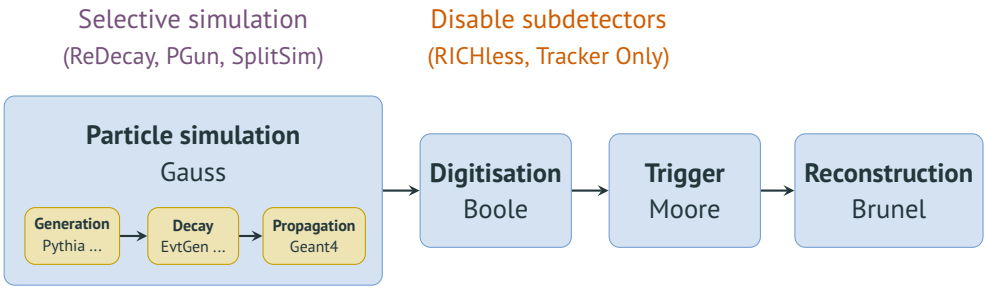


95 – 99% of CPU time for whole chain spent in Geant4

How can we speed this up?

- Code optimisation
- Multithreading
- Simulate fewer particles

Simulation overview

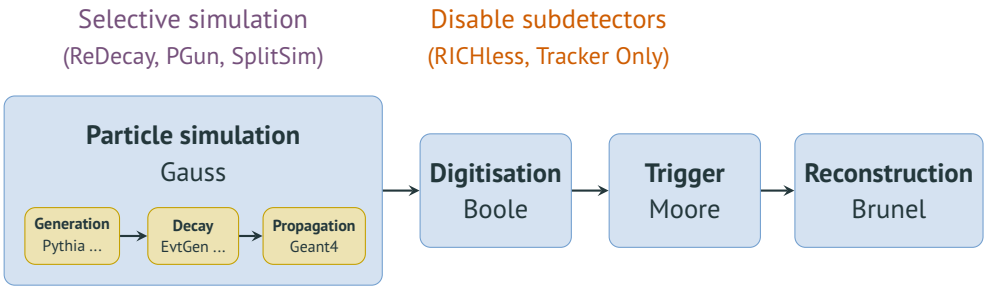


95 – 99% of CPU time for whole chain spent in Geant4

How can we speed this up?

- Code optimisation
- Multithreading
- Simulate fewer particles
- Simulate less of the detector

Simulation overview



How can we speed this up?

- Code optimisation
- Multithreading
- Simulate fewer particles
- Simulate less of the detector
- Faster subdetector simulation

Fast simulation overview

Growing menu of fast simulation options

	Step sped-up					
Method	Generation	Decay	Propagation	Digitisation	Trigger	Reconstruction
ReDecay	✓	✓	✓			
PGun	✓	✓	✓			
SplitSim	✓		✓			
RICHless			✓			
TrackerOnly			✓			
Lamarr			✓	✓	✓	✓
FastCALO*			✓			

* [Separate talk by M. Rama]

Particle gun

In production

- Only generate the signal particles without underlying event
- Primary vertex smearing applied in reconstruction

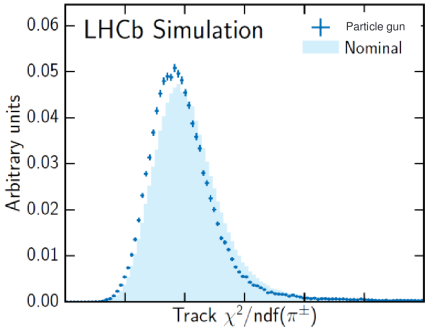
✓ Roughly 50 – 100× faster

✓ About 1% of the disk usage

- Useful for quick checks, but limited suitability for physics analysis
  - ✓ Good for Central Exclusive Production

✗ No information from underlying event

✗ Poorly reproduces variables affected by local occupancy



[LHCb-TALK-2018-302]

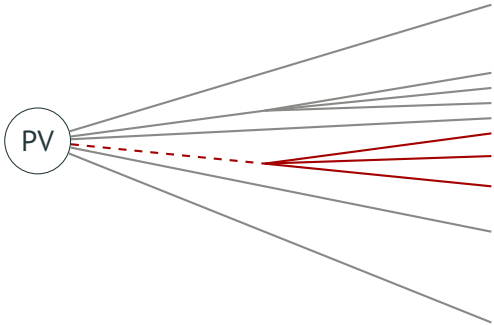
[EPJC (2018) 78:1009]

In production

Idea: use the same underlying event for many signal decays.

Method:

- Generate a full event
- Delete signal decay but save production kinematics
- Generate *N* signal decays and merge with underlying event



Configurable:

- *N* decays (typically 100)
- What to re-decay (signal only, heaviest ancestor, all heavy flavour, …)

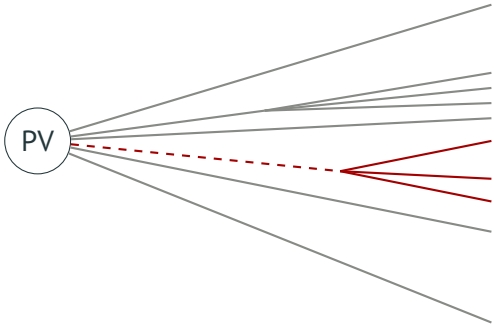
[EPJC (2018) 78:1009]

In production

Idea: use the same underlying event for many signal decays.

Method:

- Generate a full event
- Delete signal decay but save production kinematics
- Generate *N* signal decays and merge with underlying event



Configurable:

- *N* decays (typically 100)
- What to re-decay (signal only, heaviest ancestor, all heavy flavour, …)

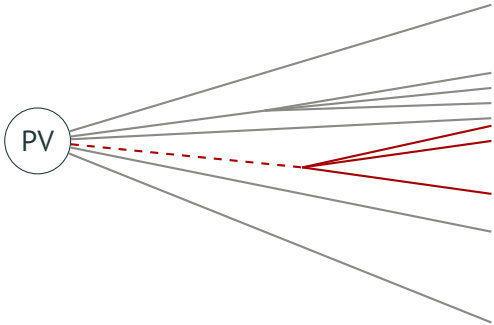
[EPJC (2018) 78:1009]

In production

Idea: use the same underlying event for many signal decays.

Method:

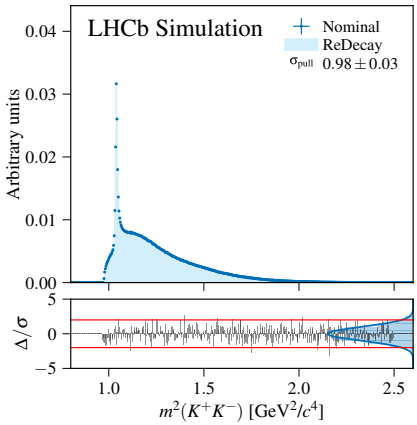
- Generate a full event
- Delete signal decay but save production kinematics
- Generate *N* signal decays and merge with underlying event



Configurable:

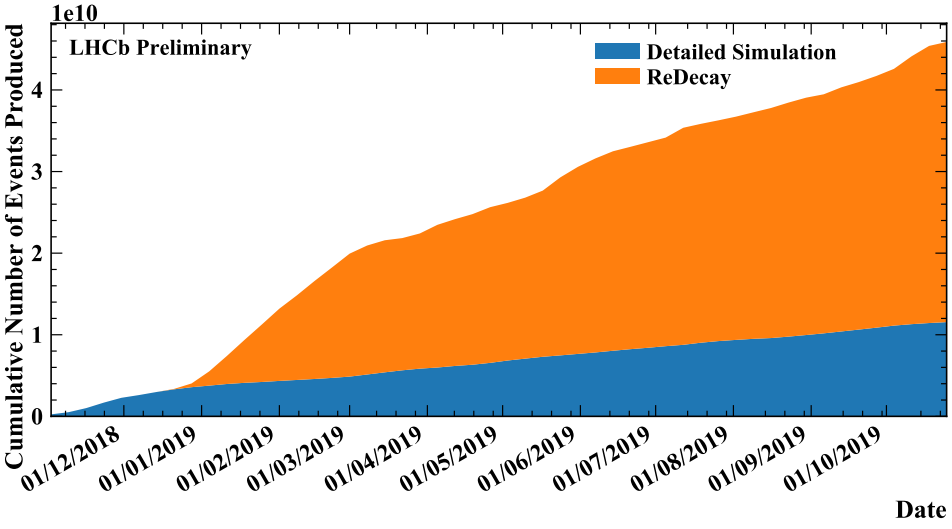
- *N* decays (typically 100)
- What to re-decay (signal only, heaviest ancestor, all heavy flavour, …)

- ✓ Independent of choice of generator
- ✓ Can be combined with other fast simulation options
- ✓ Speed-up simulation by 10× to 50×
- ⚠ Loss of statistical independence between events
  - Statistical uncertainties $\geq \sqrt{N}$ depending on type of variable
  - ✓ LHCb usually deals with properties of signal decays re-generated for each event
  - Proper statistical treatment with block-bootstrapping
- ✗ Only suitable for generating specific decays (not *e.g.* min bias)



[EPJC (2018) 78:1009]

ReDecay adoption

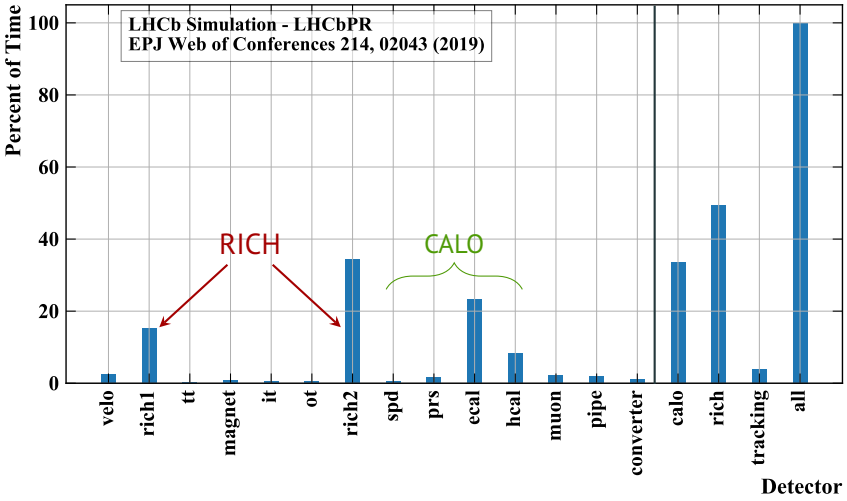


[LHCb-FIGURE-2019-018]

In production

- Simulate part of the event before deciding whether to simulate the rest of it
- Similar splitting/merging technique to ReDecay
- More effcient filtering on material interactions or particles decayed by Geant4
  - Converted photons $\gamma \rightarrow e^+e^-$
  - Rare decays of K^0_S

Timing of detector simulation



Detector simulation dominated by Cherenkov (RICH) detectors and calorimeters (CALO)

In production

- Disable generation of optical photons in the RICH
- Interaction with RICH material still simulated
  - ✓ No impact on downstream detectors

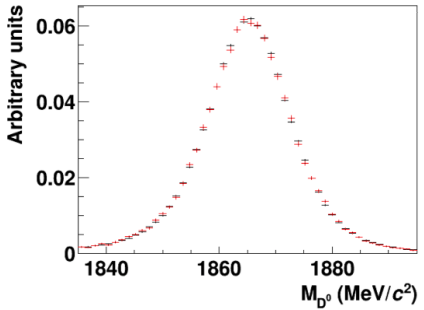
✓ Roughly 30% speedup

✓ Slight reduction in disk usage

⚠ No simulated hadron PID

- Existing parametric techniques (*e.g.* PIDCalib[LHCb-PUB-2016-021])

✗ Requires special trigger and offline selection



[LHCb-TALK-2018-302]

Tracker-only simulation

In production

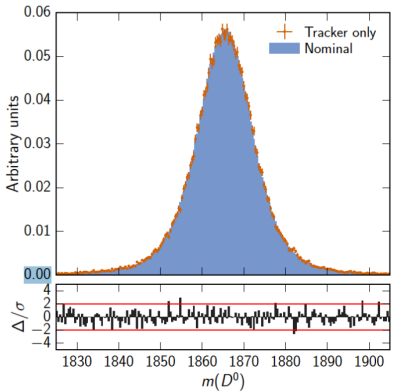
- Disable RICH optical photons
- Remove CALO and MUON entirely

✓ Roughly 10× speedup

✓ Roughly 25% disk use

✗ No PID at all

✗ No hardware trigger simulation



[LHCb-TALK-2018-302]

Fully parametrised detector simulation

Replace propagation, digitisation and reconstruction with fully-parametrised detector.

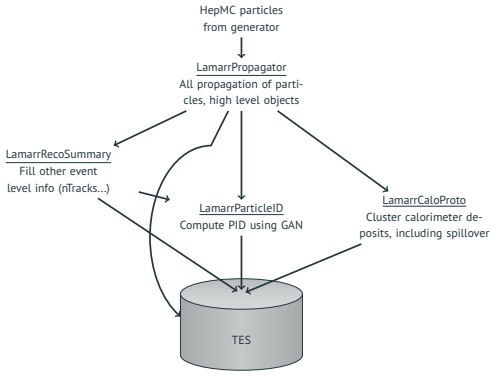
Existing tool: Delphes[JHEP 02 (2014) 057]

- LHCb geometry not natively supported
  - Implemented dipole magnet
  - Implemented cartesian calorimeter segmentation
- Interfacing with Gauss required extra steps
- Review found duplications and complications of event processing frameworks
- Decided to proceed with in-house parametrised detector simulation: **Lamarr**
- LHCb implementation will be pushed upstream to Delphes for use in HEP community

Lamarr tracking and clustering

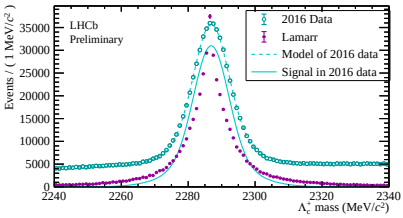
In development

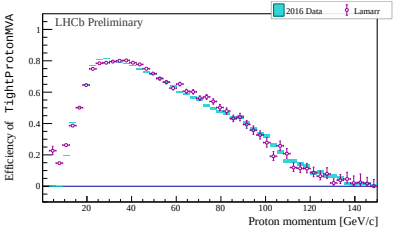
- Propagate MC particle to all points of interest, then smear and apply efficiencies
- Sample track info with Inverse Cumulative Method
- Calorimeter parameterisation using simple loops on geometries, inspired by studies with Delphes
- Stacked GANs for PID
- Output is high-level objects used in physics analysis



Lamarr physics validation

- ‘Out-of-the-box’ output compared with 2016 data
- Example: $\Lambda_b^0 \to \Lambda_c^+ \mu^- \overline{\nu}_\mu$, $\Lambda_c \to pK^-\pi^+$
- Further refinements ongoing

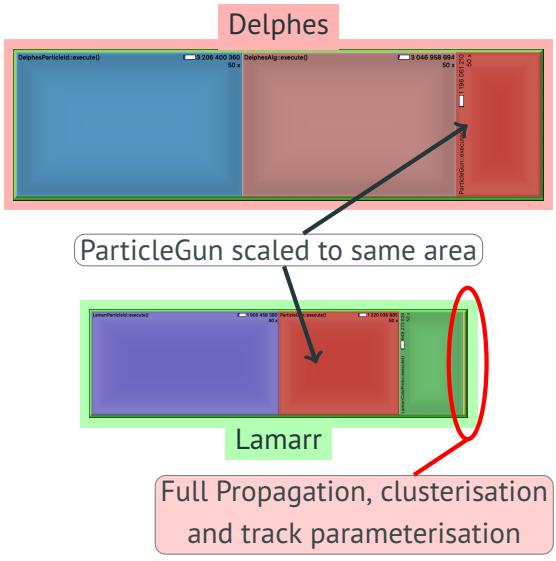




[LHCb-FIGURE-2019-017]

Lamarr performance

- Run Valgrind with Cachegrind on 50 $B^0 \to K^+K^-\pi^0(\to \gamma\gamma)$ events with both Delphes and Lamarr setup
- Propagation and high level particle making is tiny sliver on Lamarr graph
- Future improvements focus on:
  - TensorFlow memory management
  - Multithreaded random generators for calorimeter clustering



Summary and conclusion

⚠ Fast simulation an absolute necessity to meet future simulation demands

✓ Growing suite of techniques and improvements

- ✓ Complementary options that can be combined
- ✓ Successful adoption and widespread use of several techniques already
- ✓ Several more on the horizon

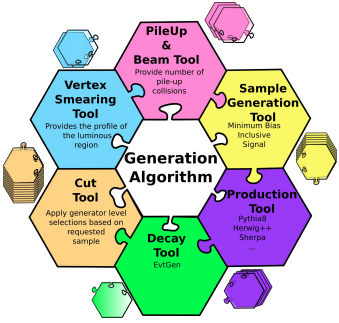
- Stay tuned for Matteo's talk on fast calorimeter simulation

Backup slides

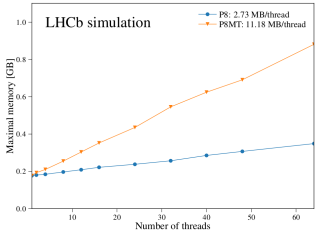
[LHCb-PROC-2019-010]

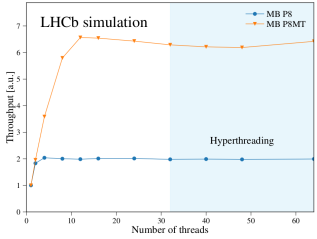
Experiment-independent core simulation framework

- Built on Gaudi using its task-based parallelism
- Modularity inspired by Gauss
- Multithreaded generation and simulation phases



In development





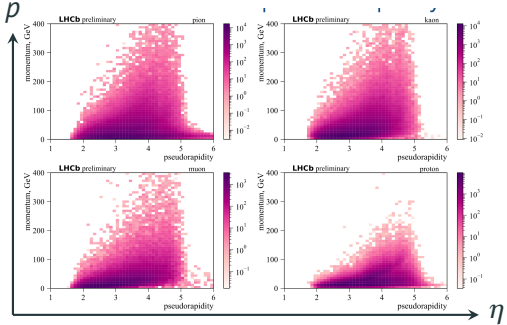
Thread-local Pythia8 configuration

Shared Pythia8 configuration

RICH response with generative adversarial network

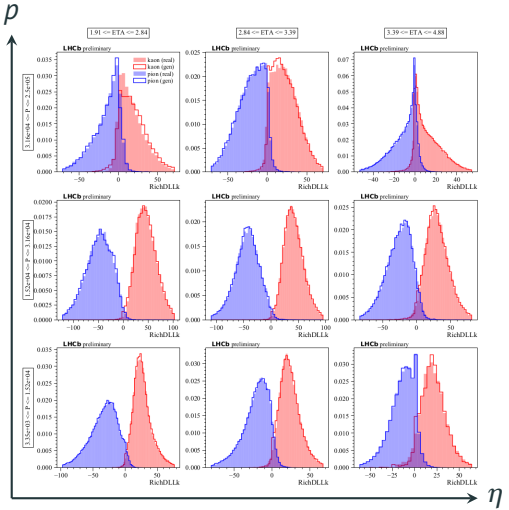
[CHEP 2019 proceedings]

- Learn PID response given only particle type and kinematics
- Based on Cramer GAN
- Trained on calibration data



Calibration data

In development



Generated vs actual $\Delta_{LL}K$ response for pions and kaons

Stacked GANs for Lamarr