Generative Models for Event Simulation

F. Ratnikov
Yandex / HSE University
LHCb Collaboration

ML4Sim Meeting, June 3, 2021

Detector Simulation Context

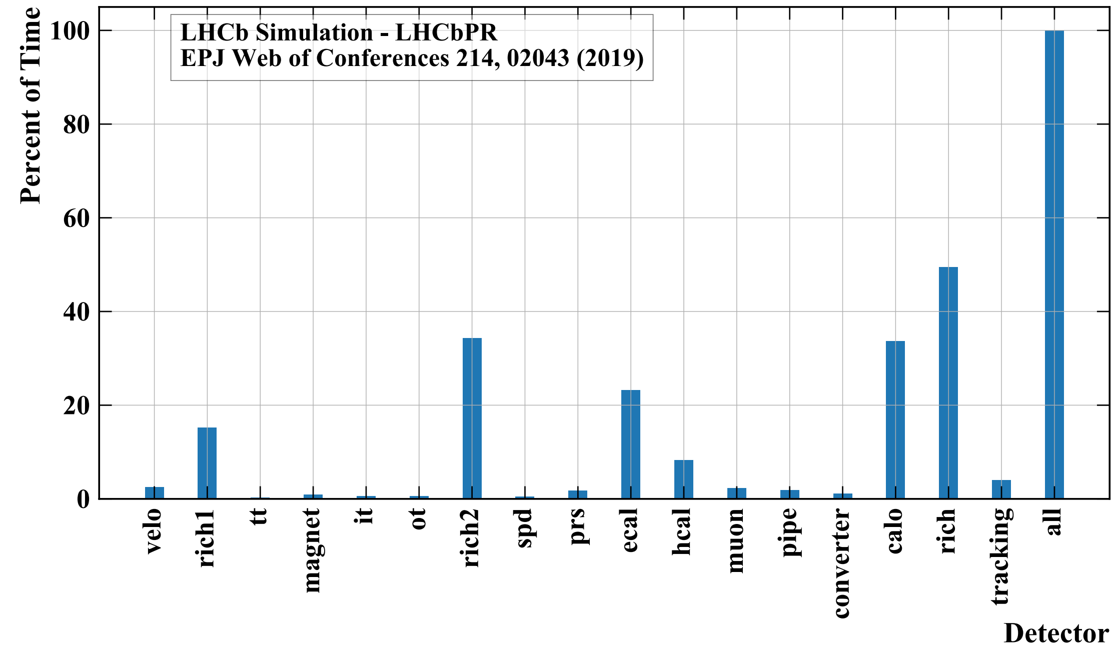
Two primary use cases

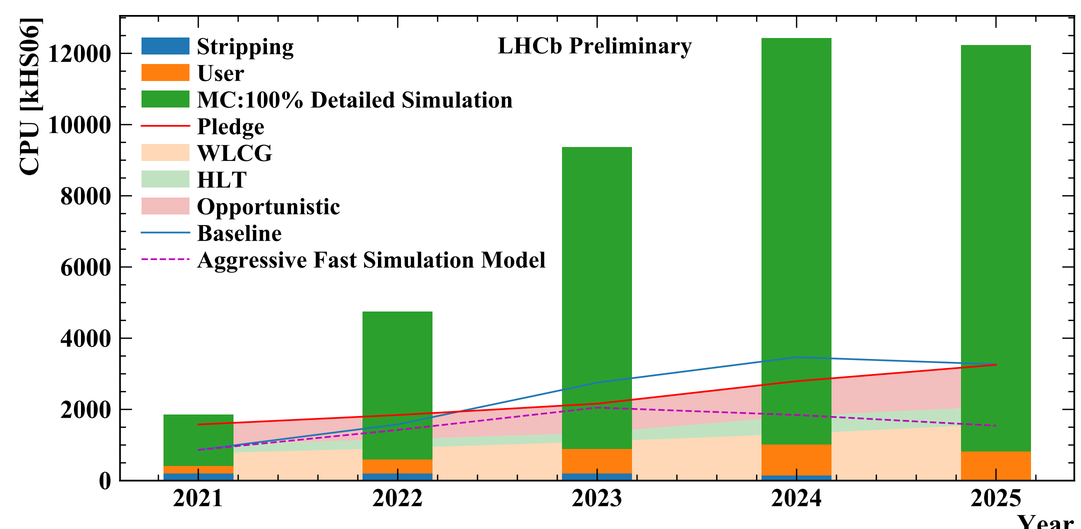
- detailed detector simulation

  GEANT

  - GEN → SIM
  - need following DIGI and RECO steps
  - most accurate, slow

  ML generative models may help to speed up bottlenecks





- fast simulation

  e.g. DELPHES

  - target individual simulation chain steps.

    extreme - GEN→RECO in single aggregated step

  - fast, can be less accurate (parametrized models)

  ML generative models may improve aggregated detector response model

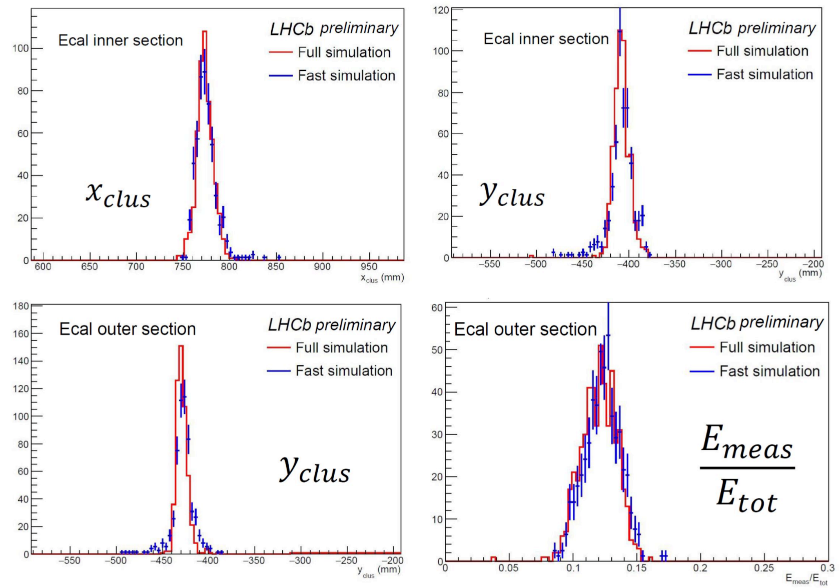
Library vs Generative Approach

Reference dataset is necessary to train generative model

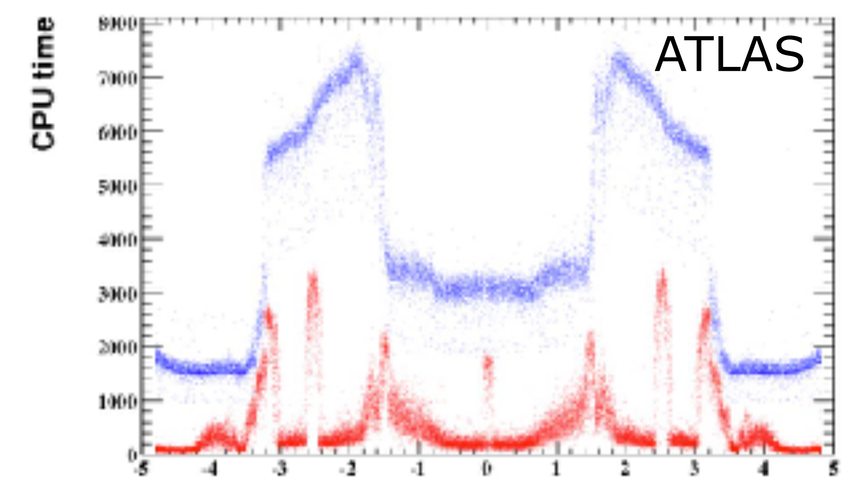
Reference dataset may be used to sample objects directly

- approach accommodated by CMS, ATLAS, LHCb
- PRO library approach comparing to generative models
  - aggregated distributions are guaranteed by construction
- PRO generative models comparing to library approach
  - discreetness of events
    - partly compensated by energy scaling
  - speed
    - massive matrix operations vs massive object search
  - size
    - both transient and persistent

LHCb



doi 10.22323/1.340.0162

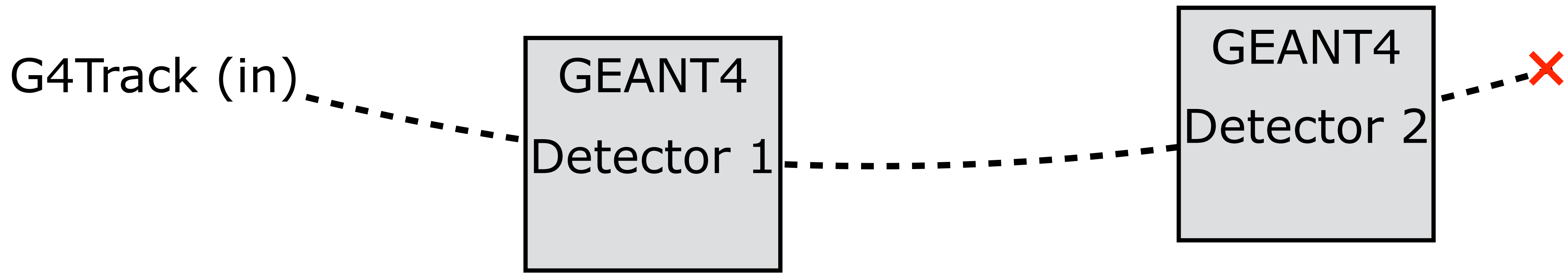


E Barberio *et al* 2008 *J. Phys.: Conf. Ser.* **119** 032008

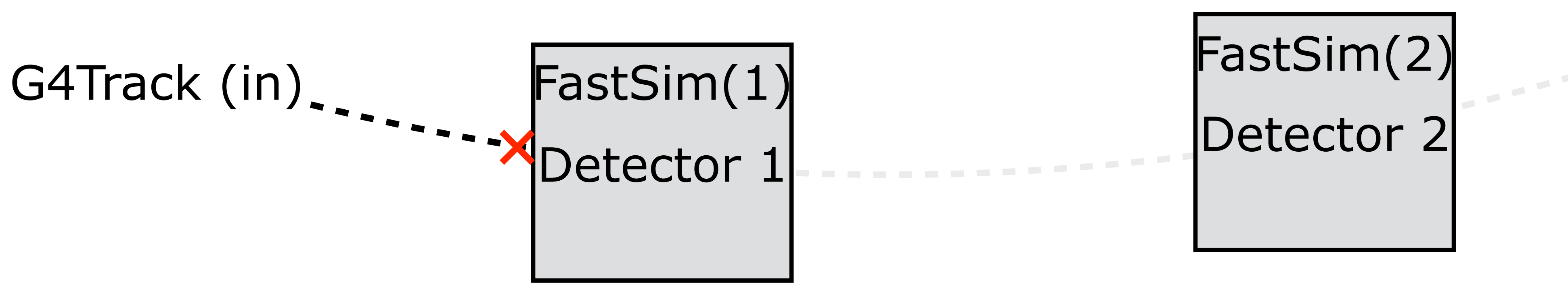
From technical perspective, library-based and ML-based modules have very similar interfaces for both gathering train data and inferencing objects

Operation Scheme

To speed up G4 we need to intercept G4Track in front of the detector, generate detector response, fill DetHits structures



$$\text{G4Track(in)} \xrightarrow[\text{GEANT4}]{} \text{Detector1Hits, G4Tracks} \xrightarrow[\text{GEANT4}]{} \text{Detector2Hits}$$



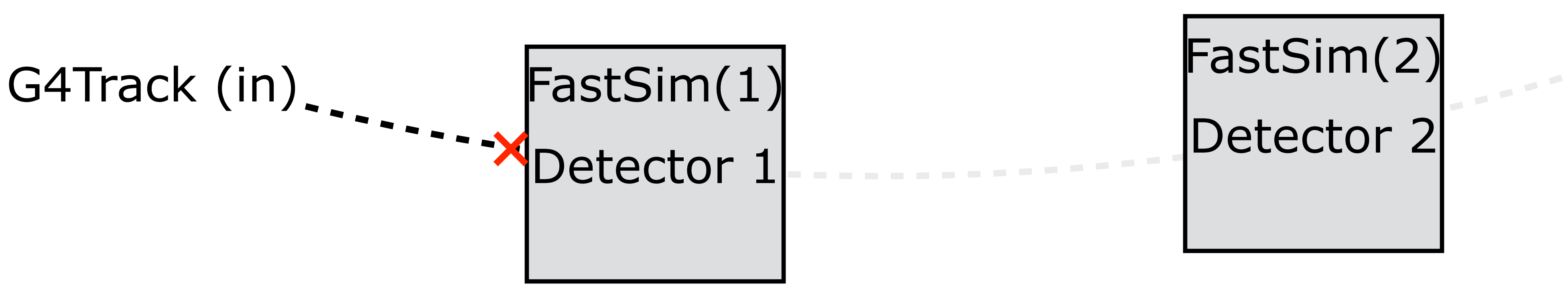
$$\text{G4Track(in)} \xrightarrow[\text{FastSim}]{} \text{Detector2Hits} \oplus \text{Detector2Hits}$$

Is partly addressed by by GEANT4 FastSim interface

Operation Scheme

Modular structure for detector processing with full and fast approaches is required: $\text{G4Track(in)} \xrightarrow[\text{GEANT4 or FastSim}]{} \text{DetectorHits}$

$$\text{G4Track(in)} \xrightarrow[\text{GEANT4}]{} \text{Detector1Hits, G4Tracks} \xrightarrow[\text{GEANT4}]{} \text{Detector2Hits}$$



$$\text{G4Track(in)} \xrightarrow[\text{FastSim}]{} \text{Detector2Hits} \oplus \text{Detector2Hits}$$

Pipeline

Collect train data

- run $\text{G4Track} \xrightarrow[\text{GEANT4}]{} \text{DetectorHits}$
- collect train dataset (G4Track, DetectorHits)

Train generative model (GAN)

- conditioned by track parameters
- (G4Track, DetectorHits) → GAN

  $\text{G4Track} \xrightarrow[\text{GAN}]{} \text{DetectorHits}$

Speedup simulation

- $\text{G4Track} \xrightarrow[\text{GAN}]{} \text{DetectorHits}$

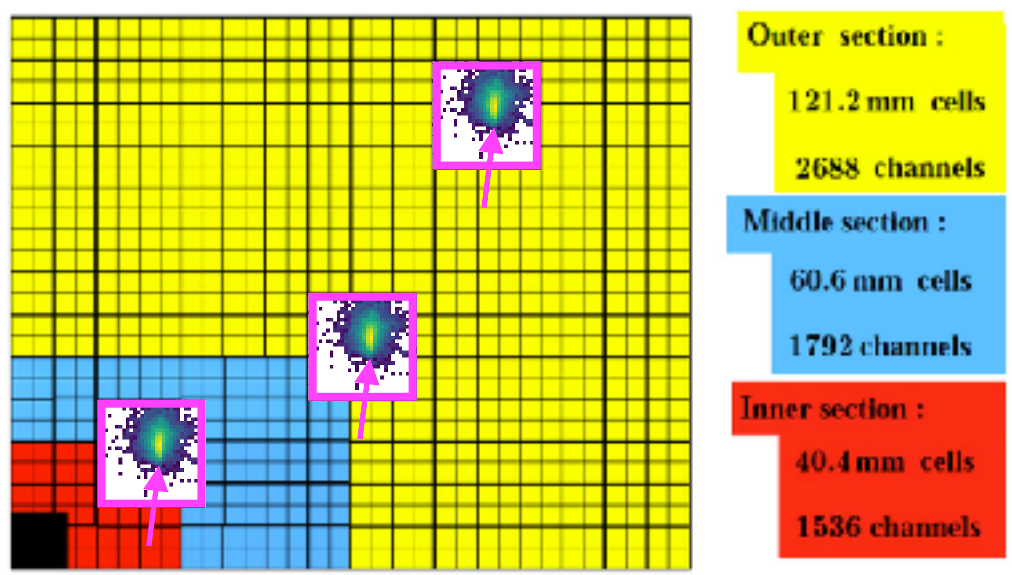
Dimensionality Reduction

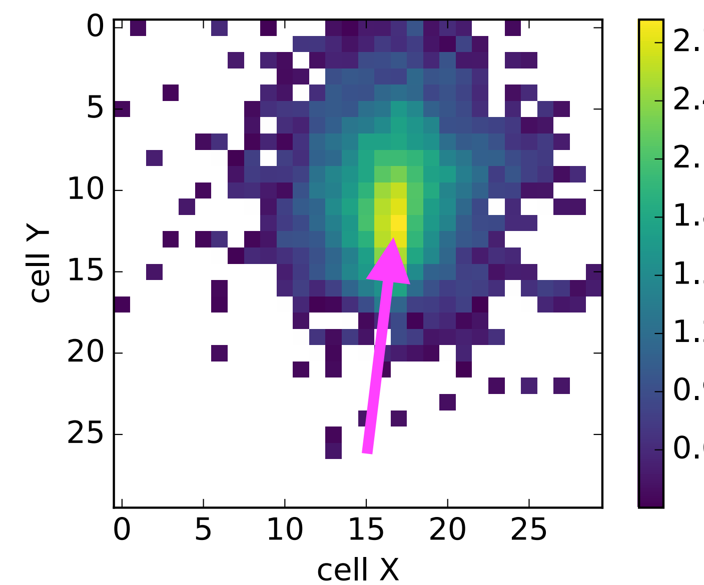
We can hardly build generative model for the full detector

- many channels - high dimensional objects

Response of the impact particle is usually local

- can limit generated object to local area of the response

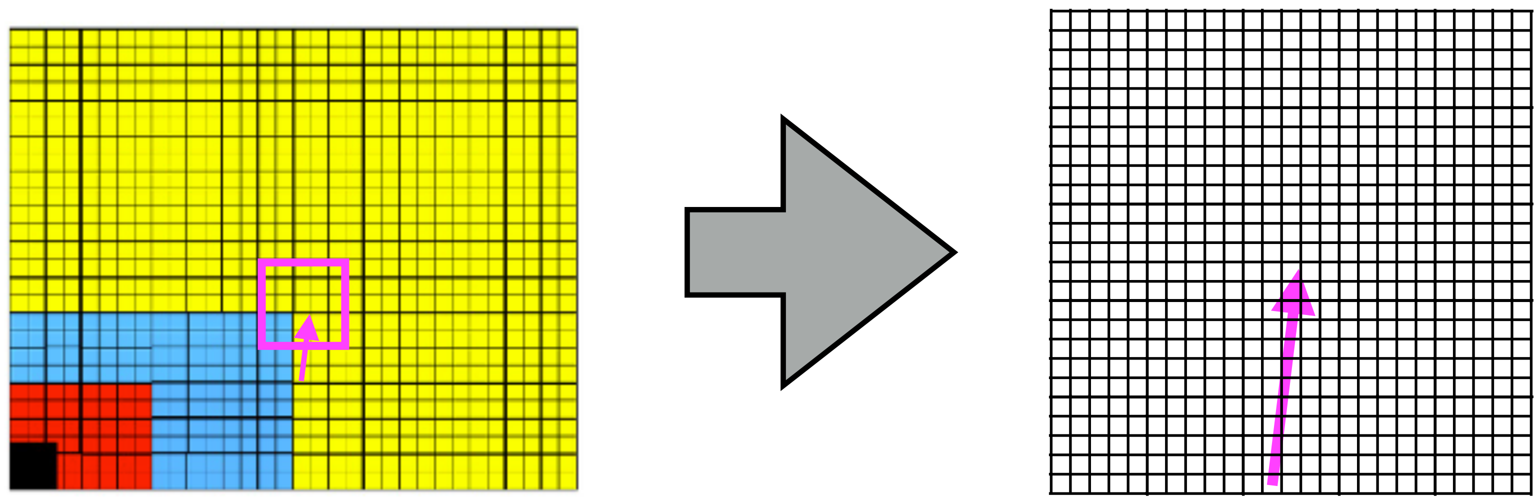




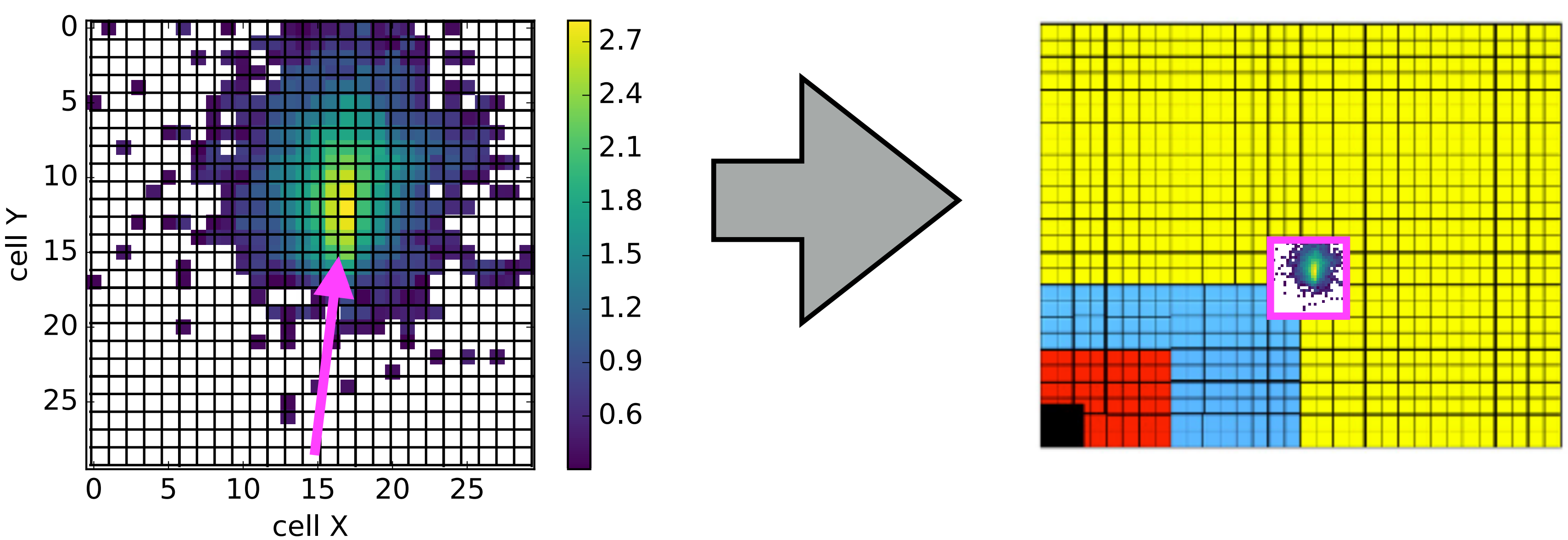
30×30 matrix of 20.2 mm cells is a proxy to 40.4, 60.6, 121.2 mm cells in any combination

Global - Local Transformations

Need interface to convert global SIM geometry to local ML model geometry and back



G4Track: global → local ML



DetectorHits: local ML → global

Alternative Dimensionality Reduction Approach

Aggregate fine grain SIM information into RECO level observables

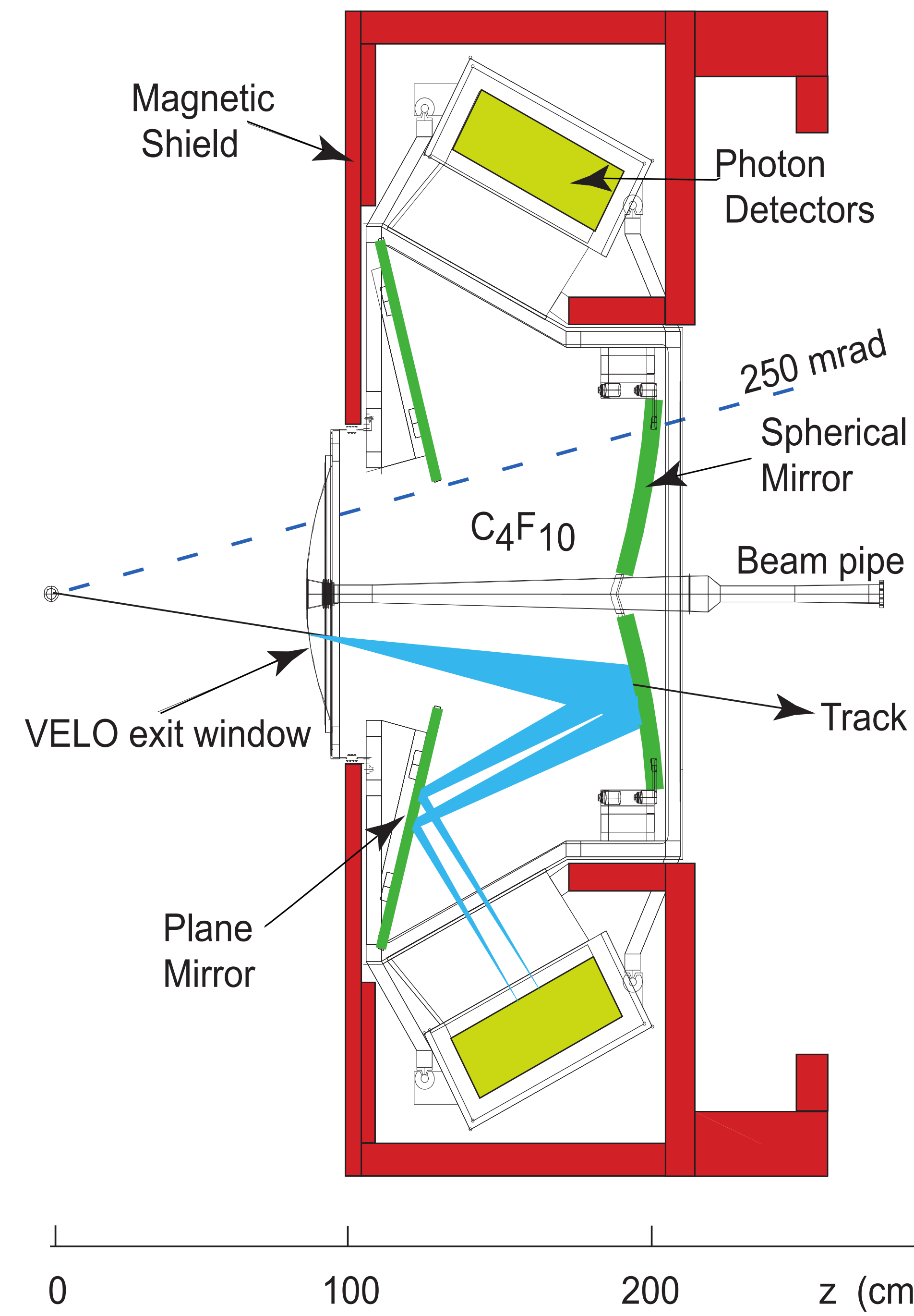
Accurate LHCb RICH simulation involves:

- tracing the particles through the radiators
- Cherenkov light generation
- photon propagation, reflection, refraction and scattering
- Hybrid Photon Detector (photo-cathode + silicon pixel) simulation

These require significant computing resources

Besides:

- quality of obtained simulated ID variables is not as good as we wish, when comparing to calibration data samples

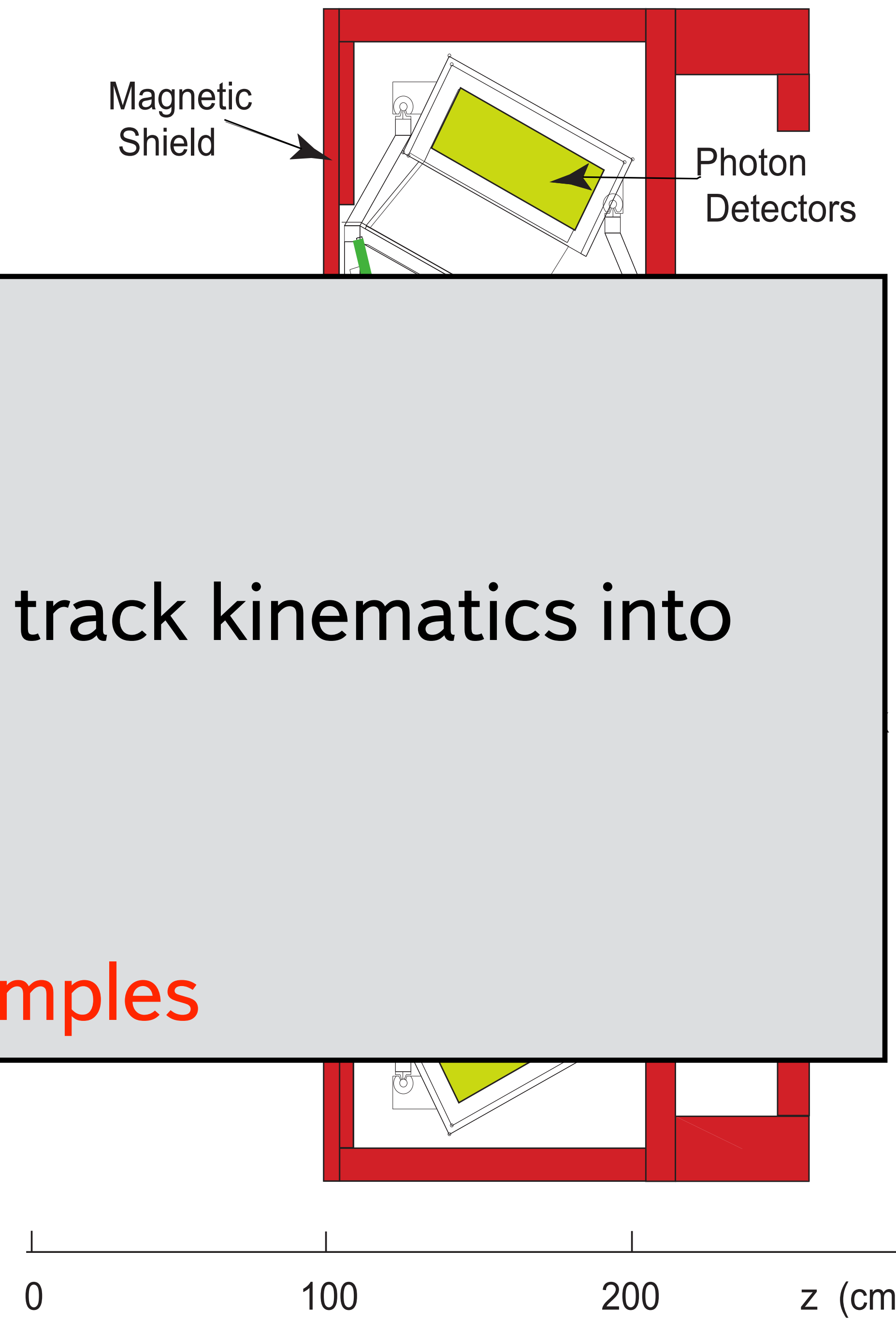


Alternative Dimensionality Reduction Approach

Aggregate fine grain SIM information into RECO level observables

Accurate LHCb RICH simulation involves:

- tracing the particles through the radiators

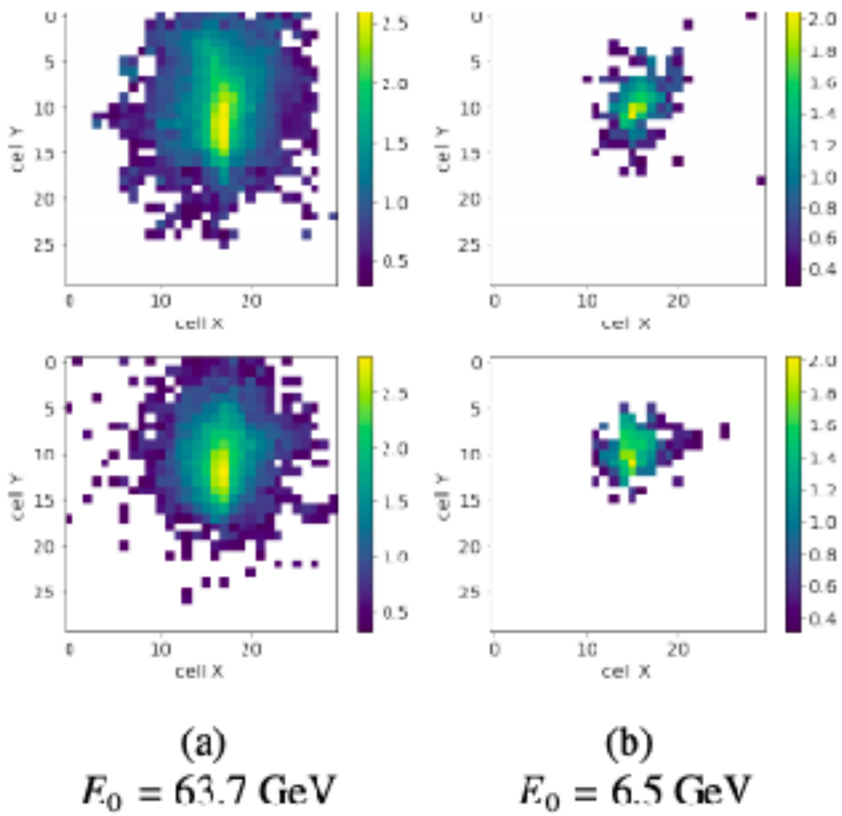


- ◇ RICH is used for particle ID only
- ◇ Let's use ML:
  - ◇ train generative model to directly convert track kinematics into PID variables
    - ◇ 3→5 generative model
    - ◇ can train directly on calibration data samples

- quality of obtained simulated ID variables is not as good as we wish, when comparing to calibration data samples

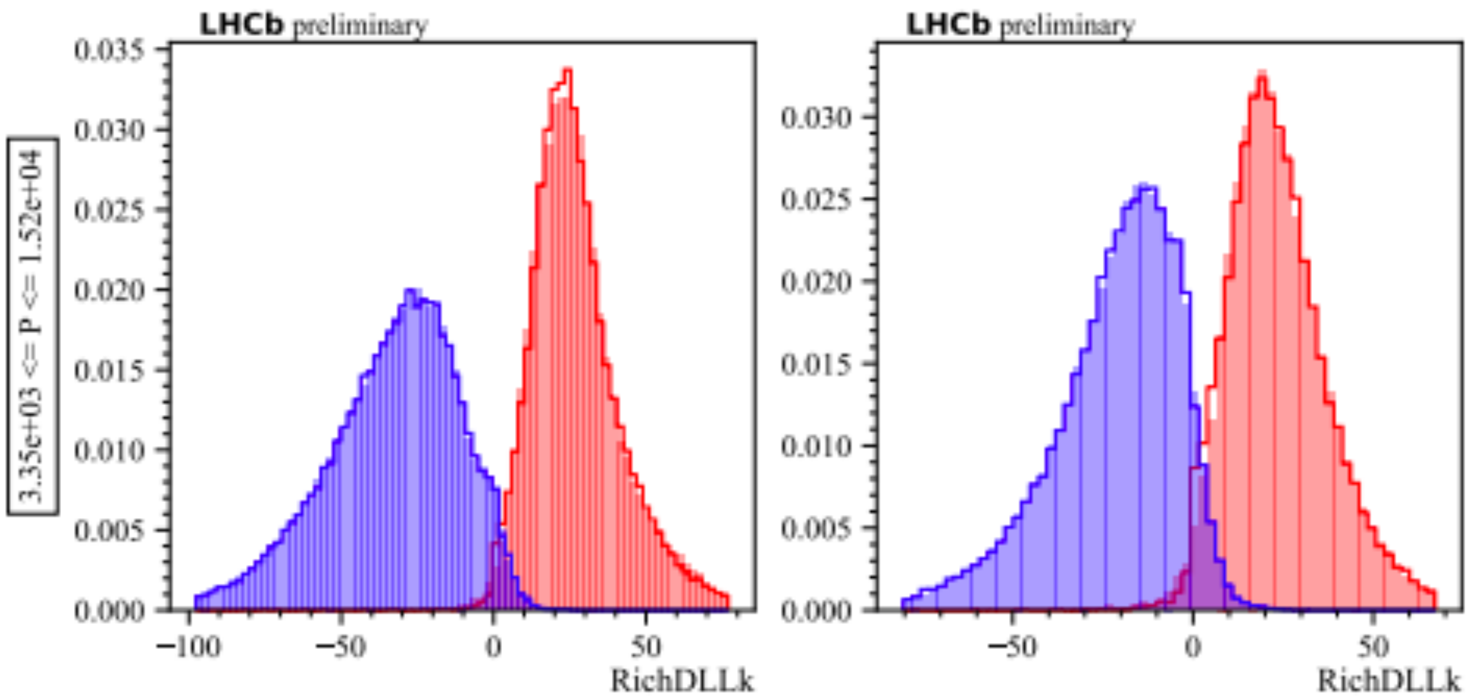
Generative Models in LHCb

Direct simulation of calorimeter responses



V. Chekalina et al. EPJ WoC: 214, 02034 (2019)

Simulation of reconstruction output for RICH and Muon ID



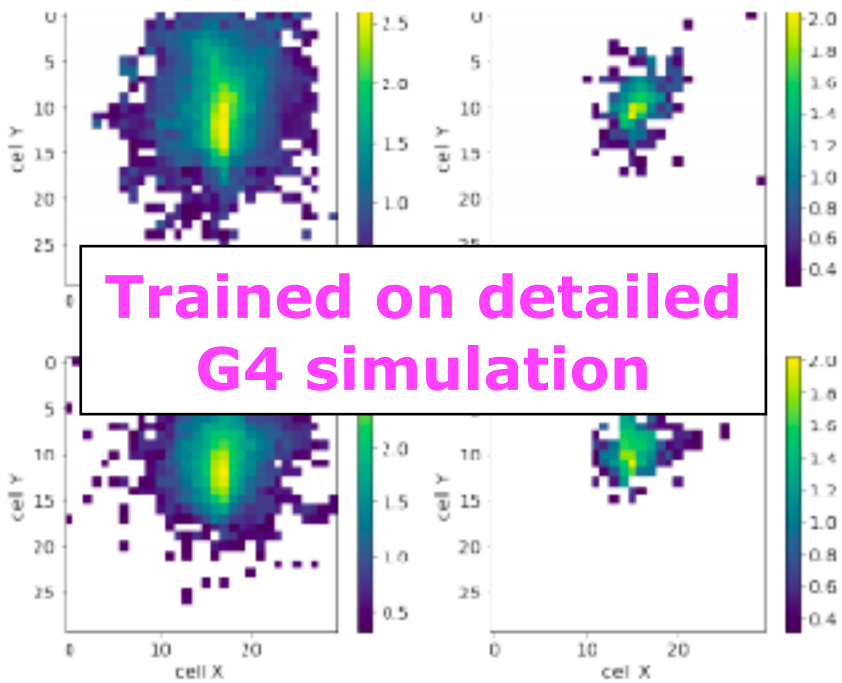
A. Maevskiy et al., ML4PHYS@Neurips 2019

Work in both directions

- speed up G4 bottlenecks
- direct simulation of RECO observables

Generative Models in LHCb

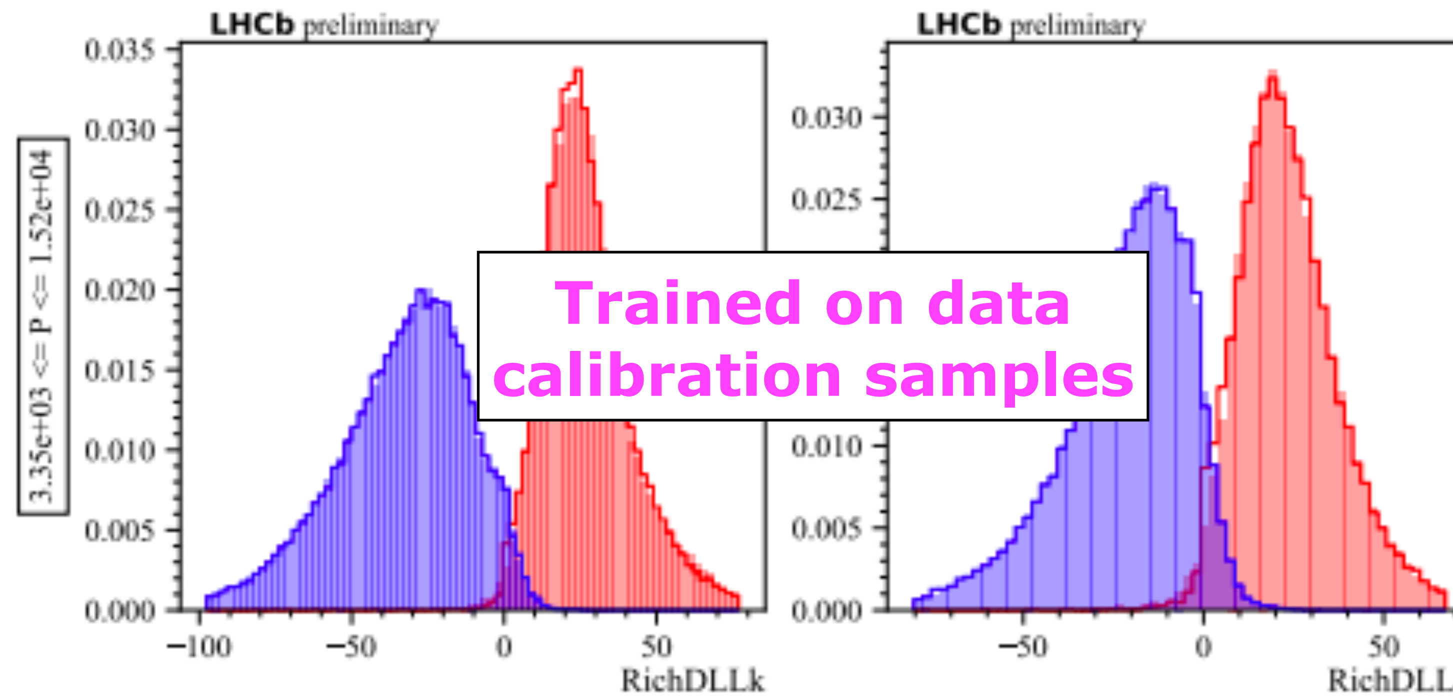
Direct simulation of calorimeter responses



(a) $E_0 = 63.7$ GeV (b) $E_0 = 6.5$ GeV

V. Chekalina et al. EPJ WoC: 214, 02034 (2019)

Simulation of reconstruction output for RICH and Muon



A. Maevskiy et al., ML4PHYS@Neurips 2019

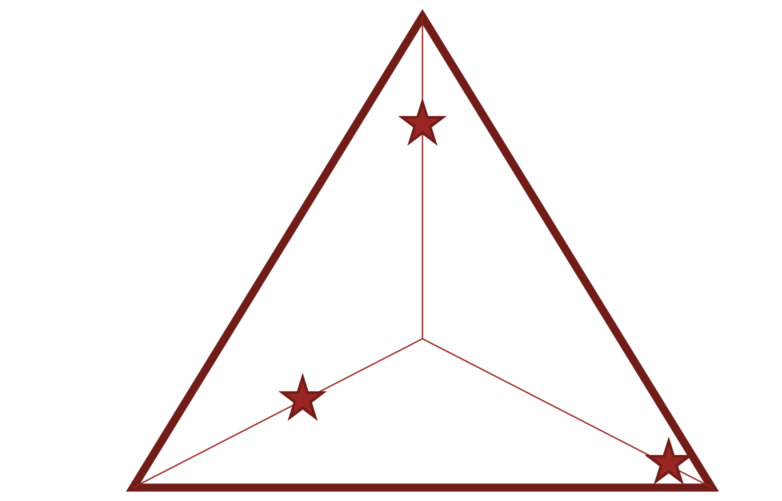
Work in both directions

- speed up G4 bottlenecks
- direct simulation of RECO observables

Generative Models Characteristics

Fast Sampling

- much faster than detailed MC
- models can get complicated
- current RICH simulation speed ~70 ms



Very Fast training

- retrain can be done very fast
- train process still should be periodically controlled
- current RICH model trains ~2 days using GPU

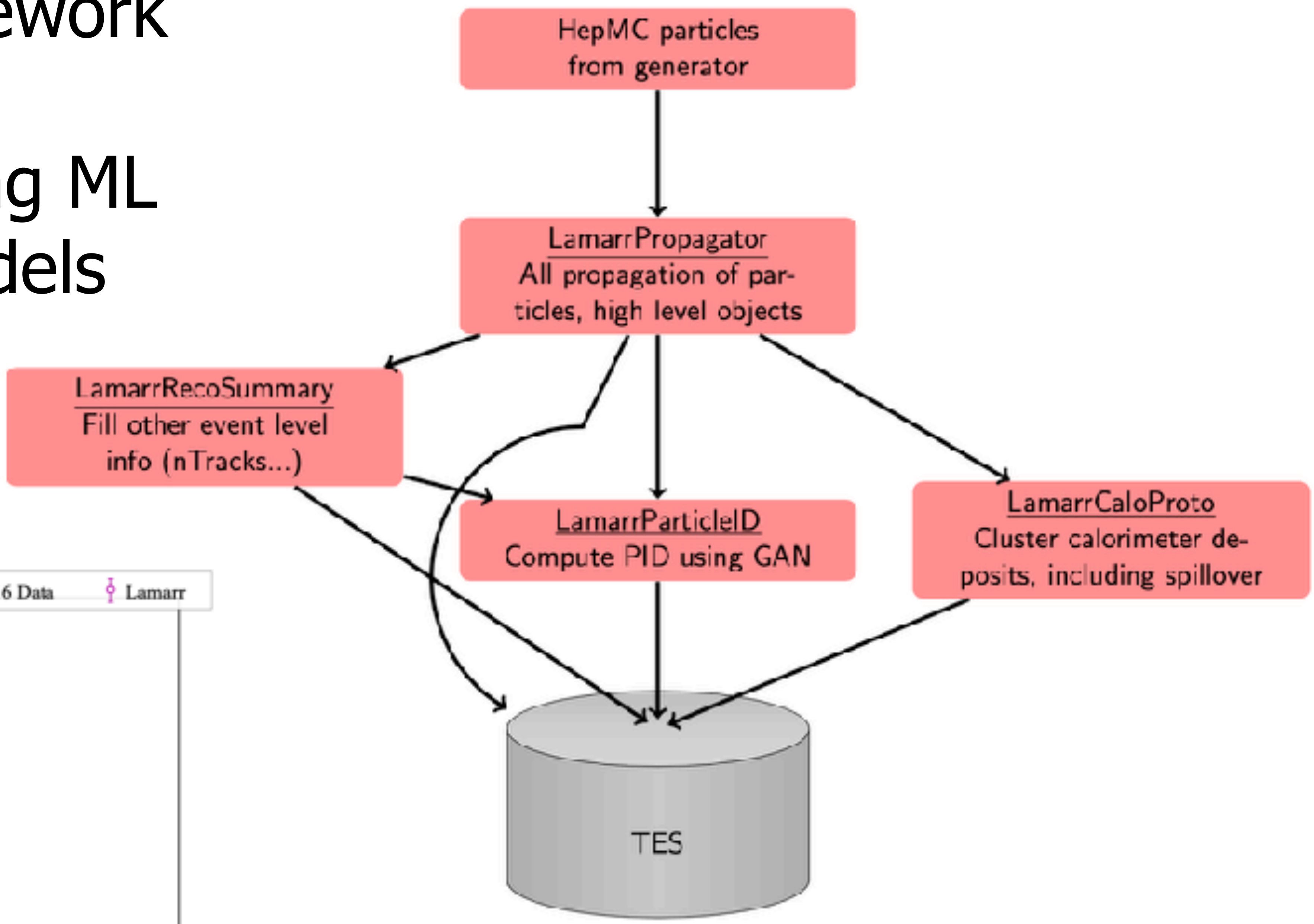
Good Precision

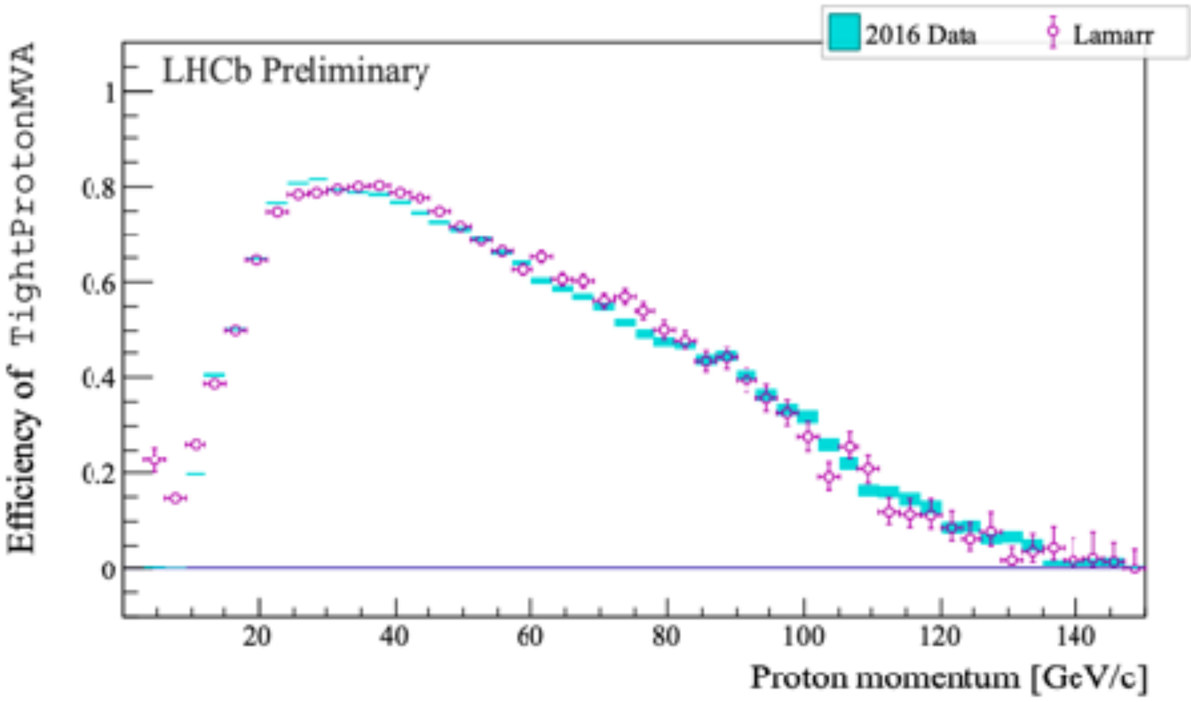
- complicated models can be quite precise
- precision is controlled by train sample statistics

current RICH precision is available in ROC AUC scores (0.52)

Fast Simulation for LHCb: Lamarr

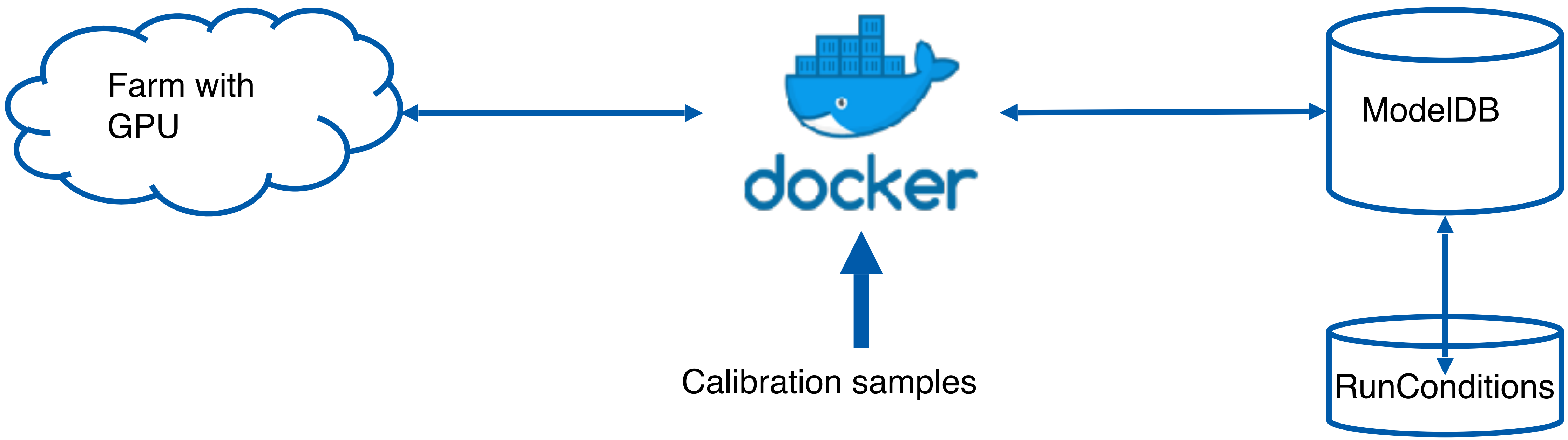
Lamarr FastSim framework allows plugins to parametrize data using ML based generative models





Training Perspective

LHCb Future Developments

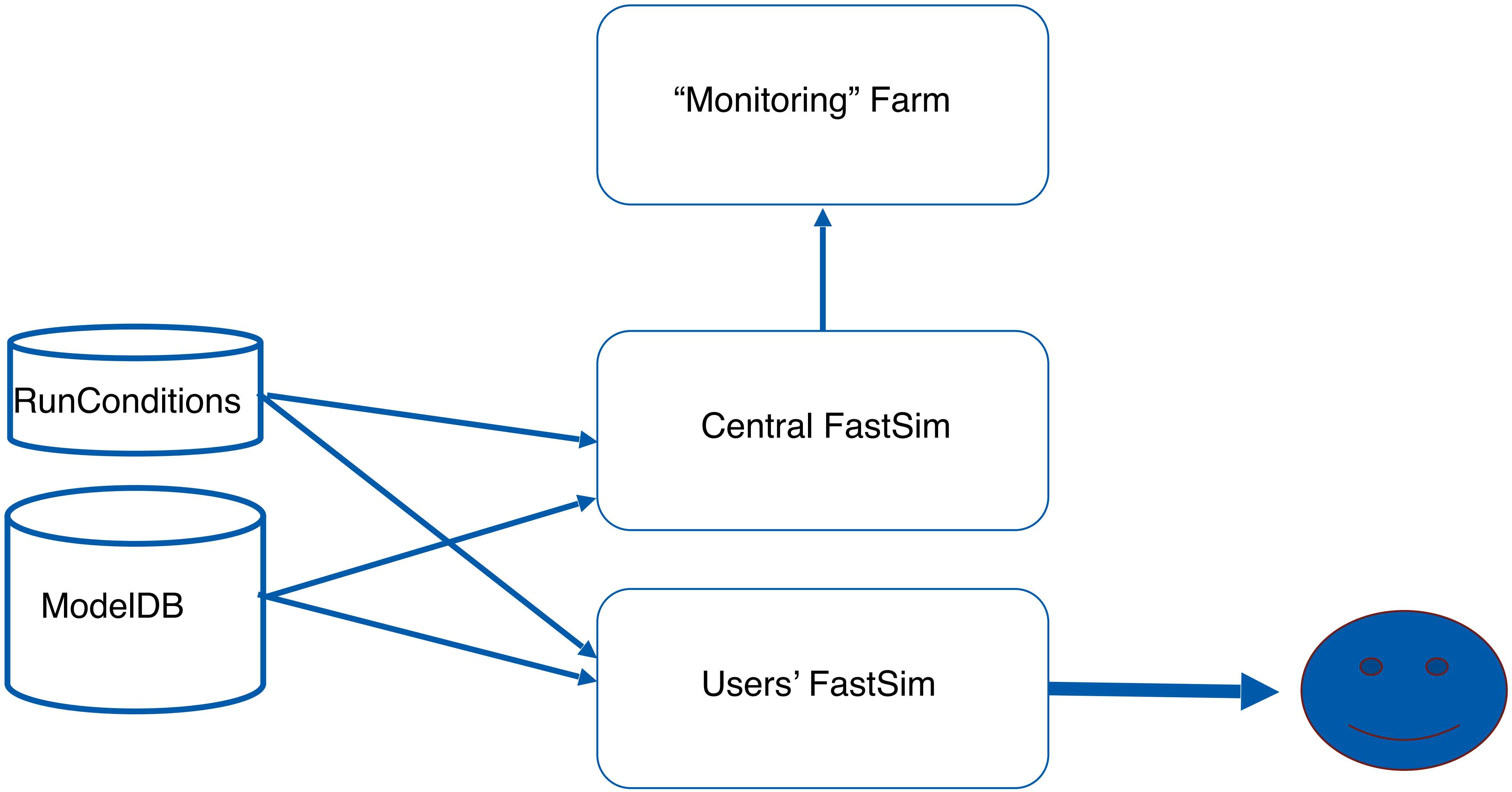


Calibration samples → generative model

Dockerized version can run anywhere

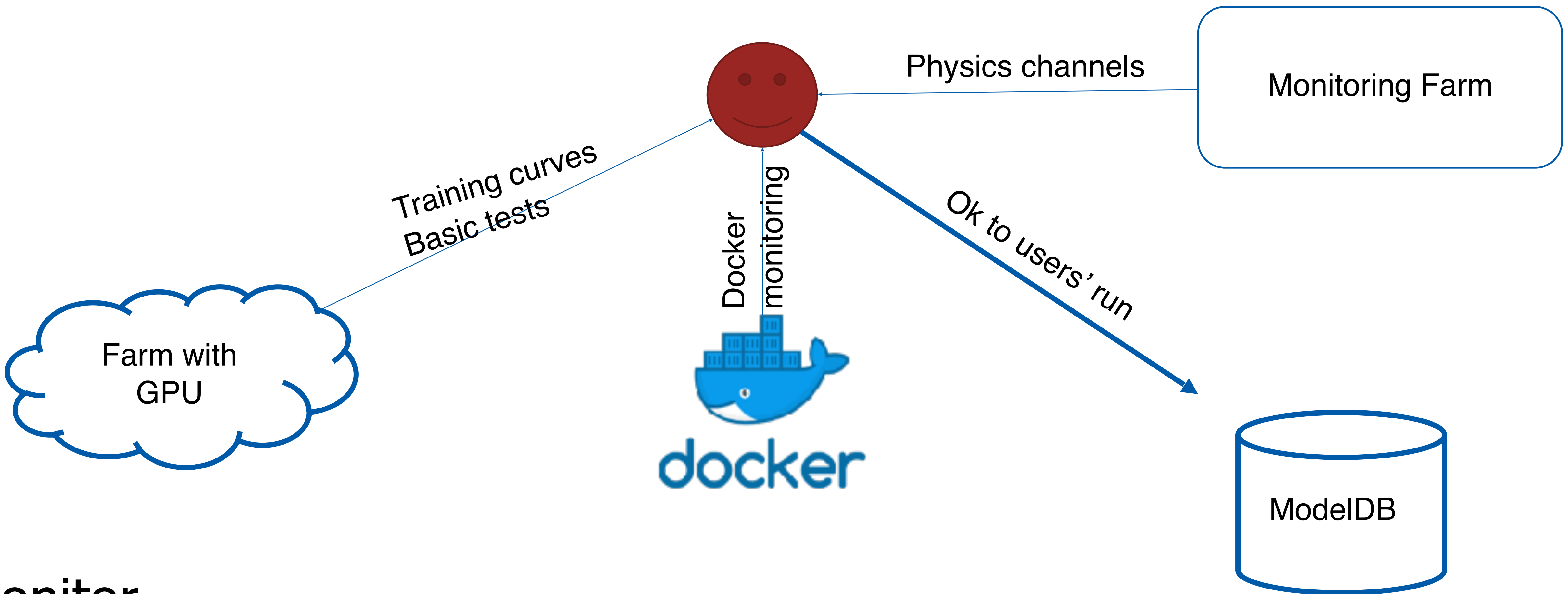
Inference Perspective

LHCb Future Developments



Monitoring Perspective

LHCb Future Developments

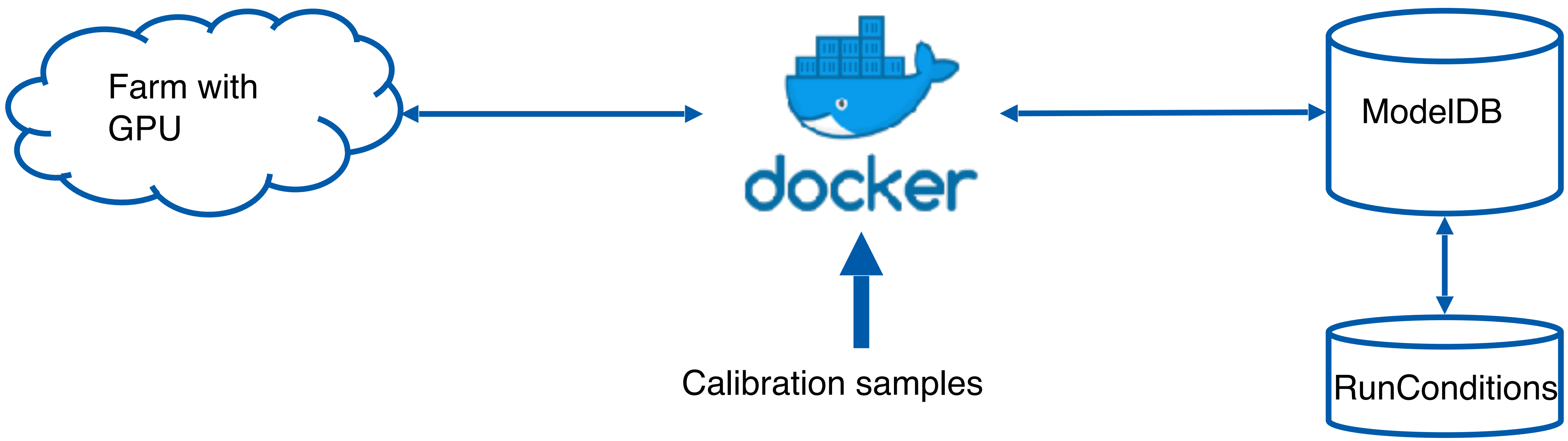


Monitor

- docker pipeline
- training procedures
- prompt - basic distributions
- physics channels

Re-training

LHCb Future Developments



Start from pre-trained models

- significant speedup speedup
- improved training stability

FastSim @ LHCb: Future Outlook

Work ongoing to form centralized system to train and test ML techniques for Calorimeter and RICH simulations

- this moves the training from individuals to a centralized system, improving reproducibility and ensuring high quality

Work ongoing to include ML-based FastSim models via GEANT4 FastSim interface

- presented ealrier in this series

  https://indico.cern.ch/event/1030029/contributions/4326642/attachments/2231623/3781422/ml4sim_at_lhcb_22_04_21.pdf

Conclusions

Infrastructure in the simulation framework is necessary to use ML based FastSim models in routine operation

- exchangeable GEANT-based and ML-based modules to convert GEANT track to detector response G4Track(in) → DetectorHits

  to collect training data

  to use fast generative model

- converter from global geometry to local model geometry

ML-based FastSim models aggregating SIM-DIGI-RECO may be trained on real data and bypass SIM-DIGI-RECO steps

They need even more operational efforts for training and validation

- closer connection between data taking run conditions and simulation
- established procedures for routine re-training models for new Run conditions
- infrastructure to plug newly trained models into operation stack