Towards Reliable Neural Generative Modeling of Detectors

Sergey Mokhnenko, Lucio Anderlini, Matteo Barbetti, Denis Derkach, Nikita Kazeev, Artem Maevskiy

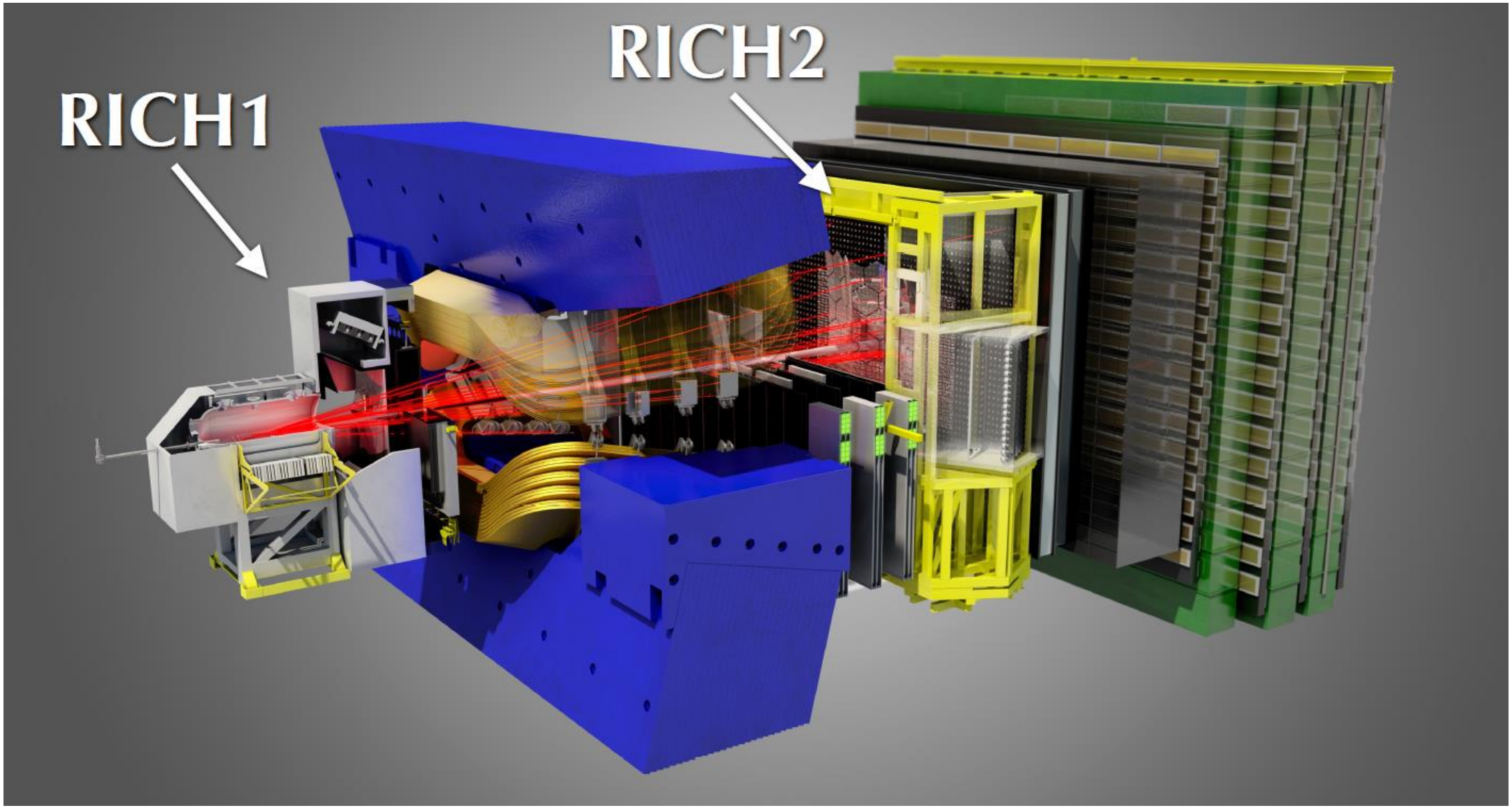
On behalf of the LHCb collaboration

20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research
ACAT 2021
29 Nov - 3 Dec 2021
Seoul, South Korea

NATIONAL RESEARCH UNIVERSITY

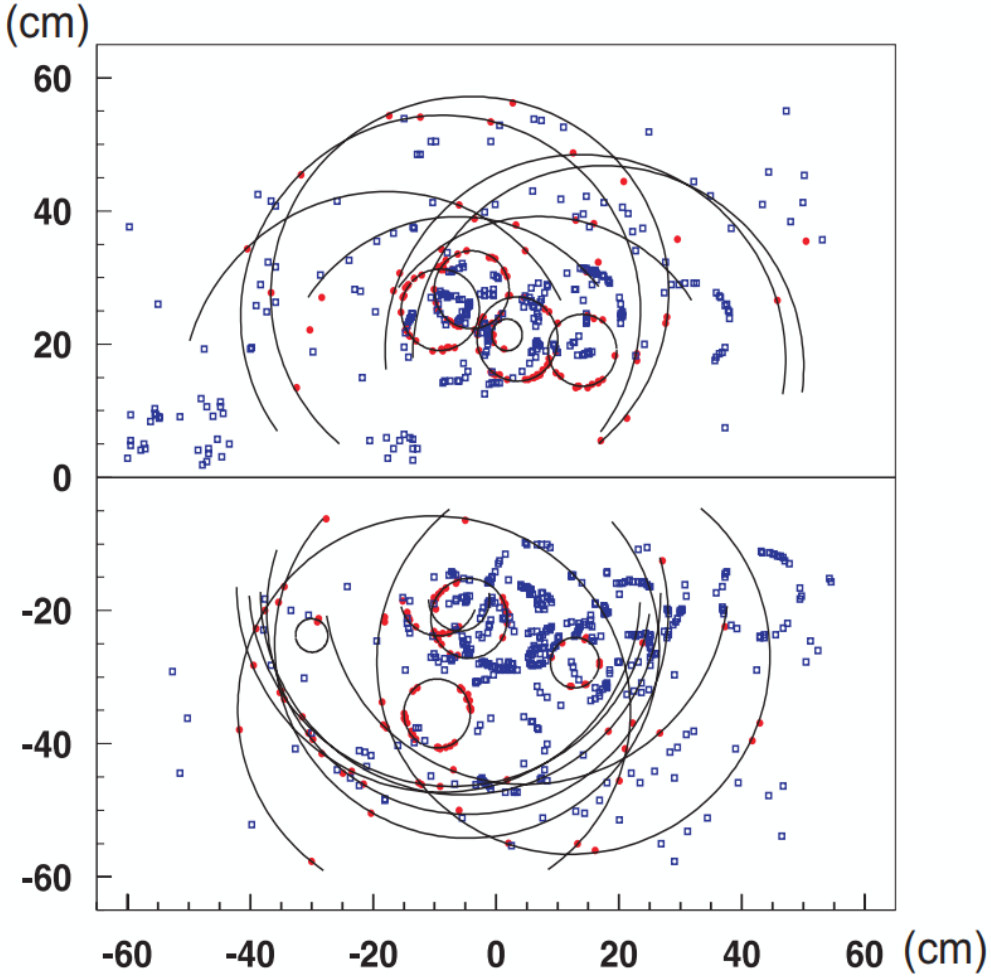
INFN
Istituto Nazionale di Fisica Nucleare

Cherenkov detectors at LHCb



Cherenkov detectors at LHCb

- A particle at a velocity higher than the speed of light in the medium emits Cherenkov photons
- Ring Imaging Cherenkov (**RICH**) detectors make use of the Cherenkov effect to identify particles
- The photons are emitted in a cone whose spread angle is a function of the particle's velocity
- Measuring this angle and momentum allows to constrain the mass of the particle

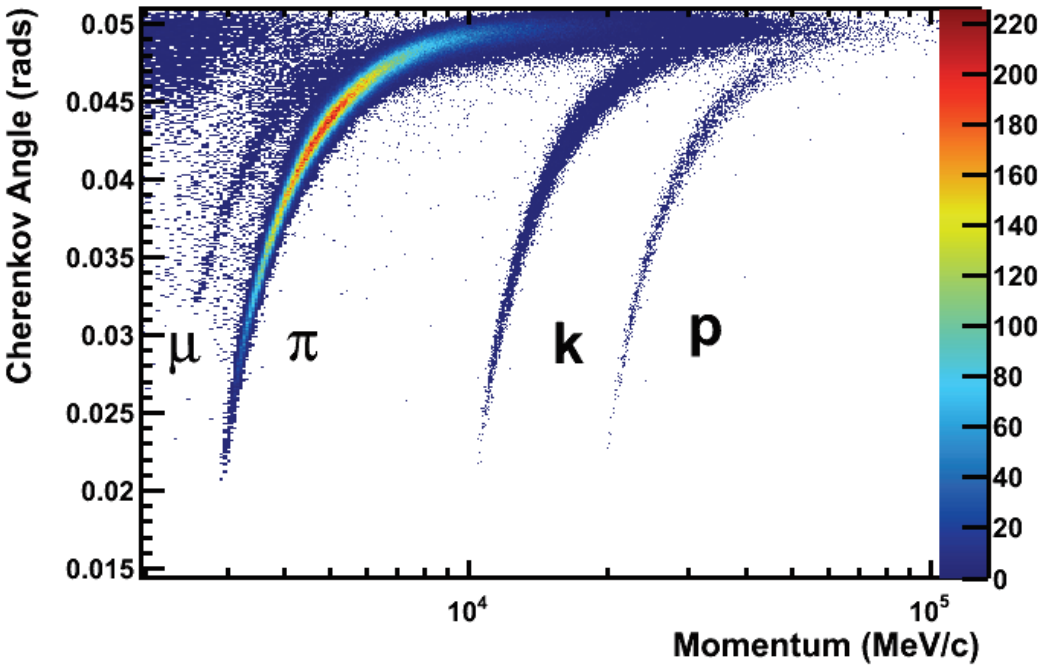


Picture from:
The LHCb Collaboration et al 2008 JINST 3 S08005

Cherenkov detectors at LHCb

- Particle identification (PID) is done with the maximum likelihood method
  - $\mathcal{L}(t_1, \dots, t_N)$ – likelihood to observe a given picture, as a function of all charged particle types (t_i- hypothesized particle type for track i)
  - A hypothesis $(t'_1, \dots, t'_N)$ maximizing $\boldsymbol{\mathcal{L}}$ is searched for
- For each track i, for each x ∈ {K, μ, e, p, below threshold}, quantities DLLx are then calculated as:

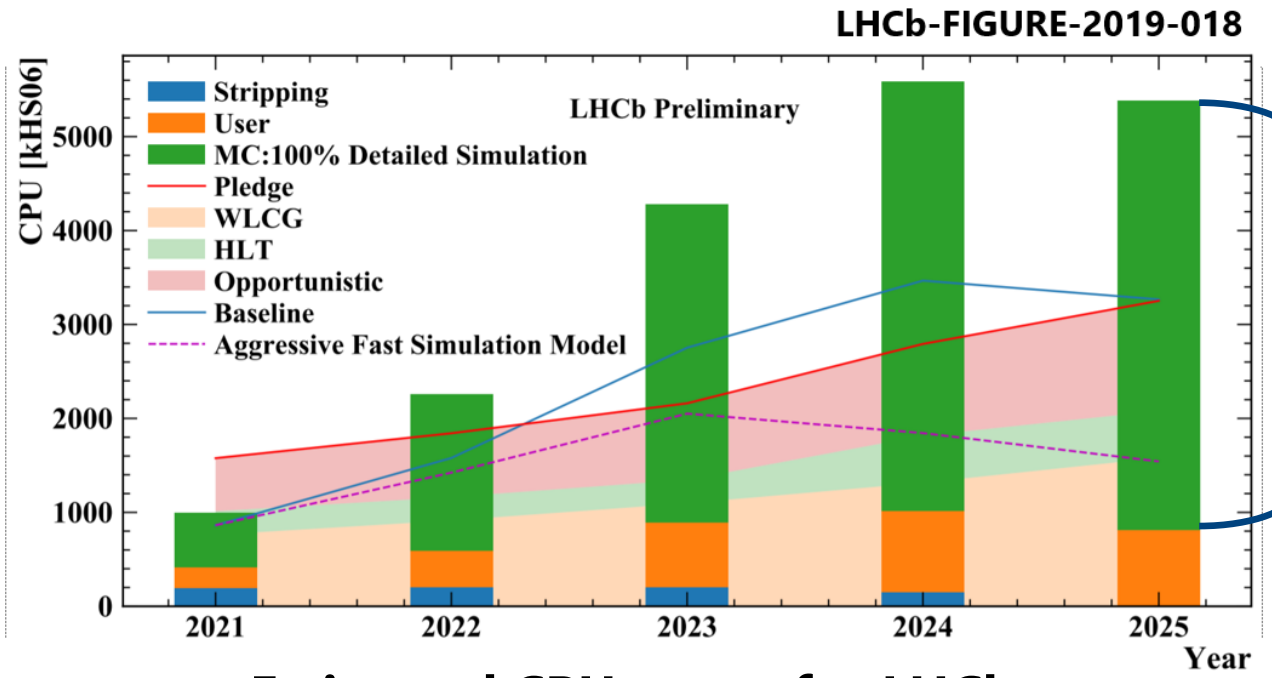
$$DLLx = log\mathcal{L}\left(t_i = x, t_{(-i)} = t'_{(-i)}\right) - log\mathcal{L}(t_i = \pi, t_{(-i)} = t'_{(-i)})$$



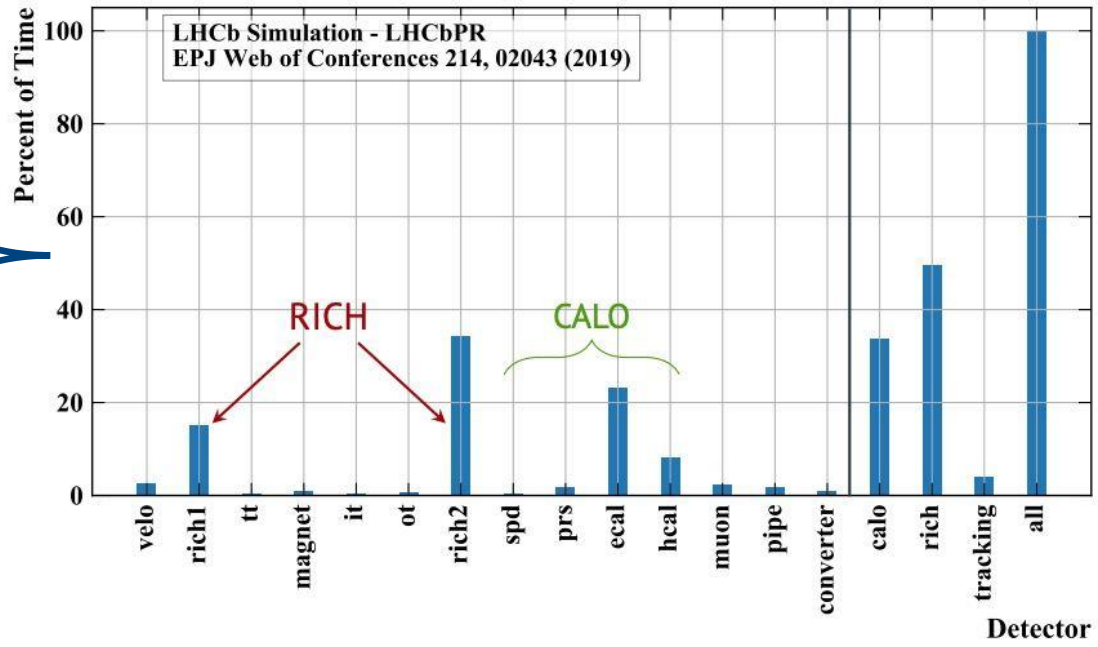
Picture from:
The LHCb RICH Collaboration, Adinolfi, M., Aglieri Rinella, G. et al. Eur. Phys. J. C (2013) 73: 2431

RICH fast simulation problem

- The task of fully simulating a RICH detector is computationally complex
- This problem will get worse with increasing luminosity in Run 3 of the LHC



Estimated CPU usage for LHCb

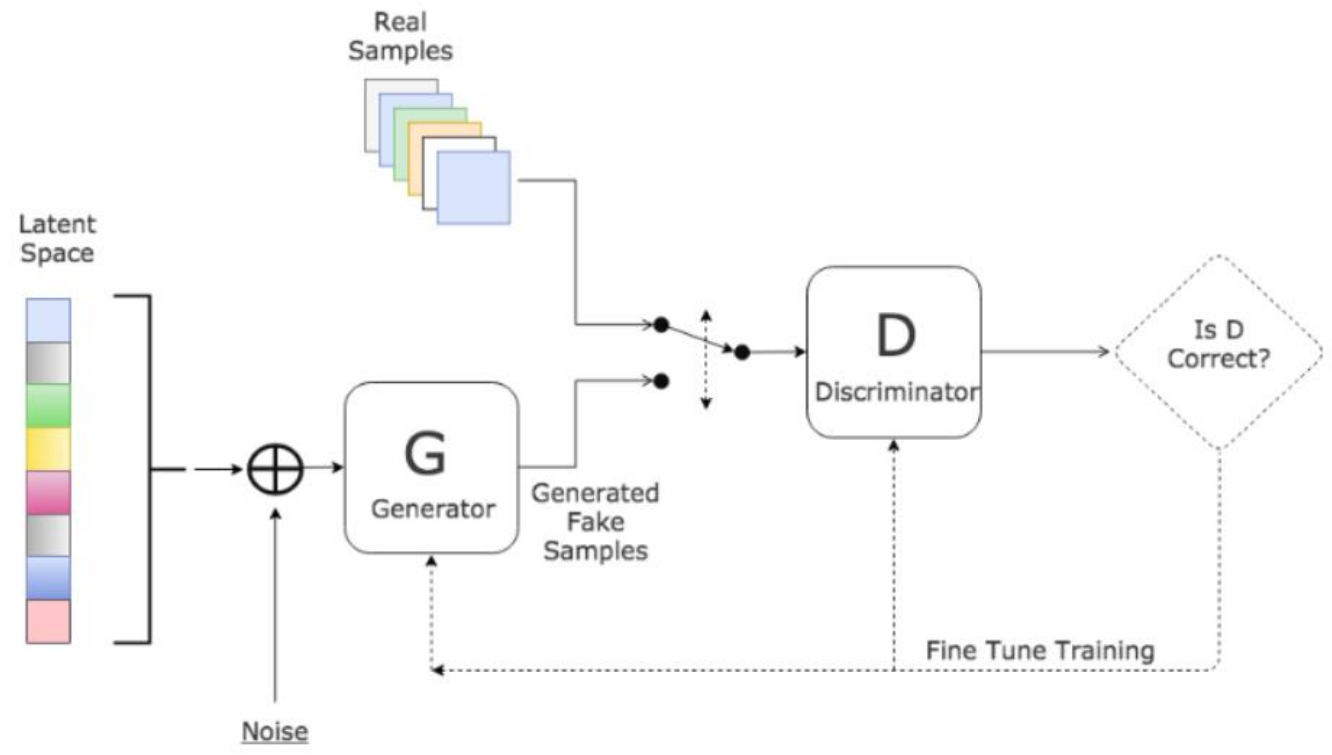


RICH fast simulation problem

- One possible solution is to develop a **data-driven approach** based on generative adversarial neural networks.
- Our global goal:
  - train a **model for fast generation** of PID parameters (RICH DLLs), given particle type and
    track characteristics;
- Current goal:
  - use **simulated data** (MC) to test the generalizability of our model
- Details:
  - Separate model trained for each of the particle types
  - 3 input variables: (P – momentum, η – pseudorapidity, nSPDHits – number of hits in the Scintillating Pad Detector)
  - 5 output variables (RichDLLx, $x \in e, \mu, k, p$, below threshold)

Generative adversarial networks (GANs)

- Two neural networks: **generator**, creates samples, **discriminator**, distinguishes correct samples from those created by generator
- These two neural networks learn to compete with each other in a zero-sum game



Picture from:
Serawork Wallelign. An Intelligent System for Coffee Grading and Disease Identification

Our model architecture (Cramér GAN)

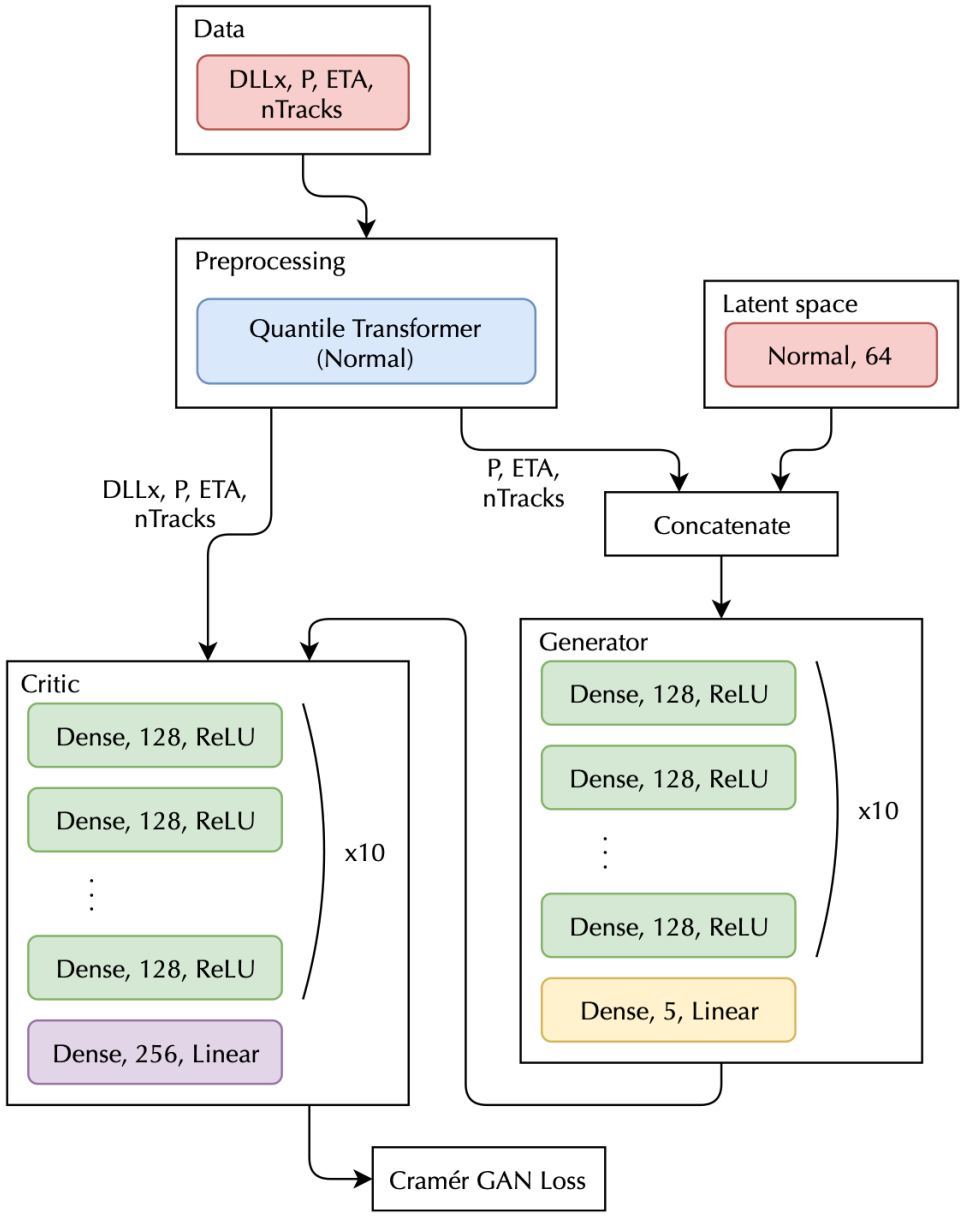
- One of the most common GAN architectures is WGAN-GP due to nice theoretical properties of the Wasserstein distance (arXiv:1704.00028)
- We use Cramér GAN (arXiv:1705.10743) which is one further improvement on top of WGAN, because it gives the best empirical agreement
- Cramér distance between distributions P and Q:

$$l_2^2(P,Q) := \int_{-\infty}^{\infty} (F_P(x) - F_Q(x))^2 \mathrm{d}x$$

F_P and F_Q are CDFs

- This is (1/2 times) the 1-dimensional case of the Energy distance: $X, X' \sim P$ and $Y, Y' \sim Q$

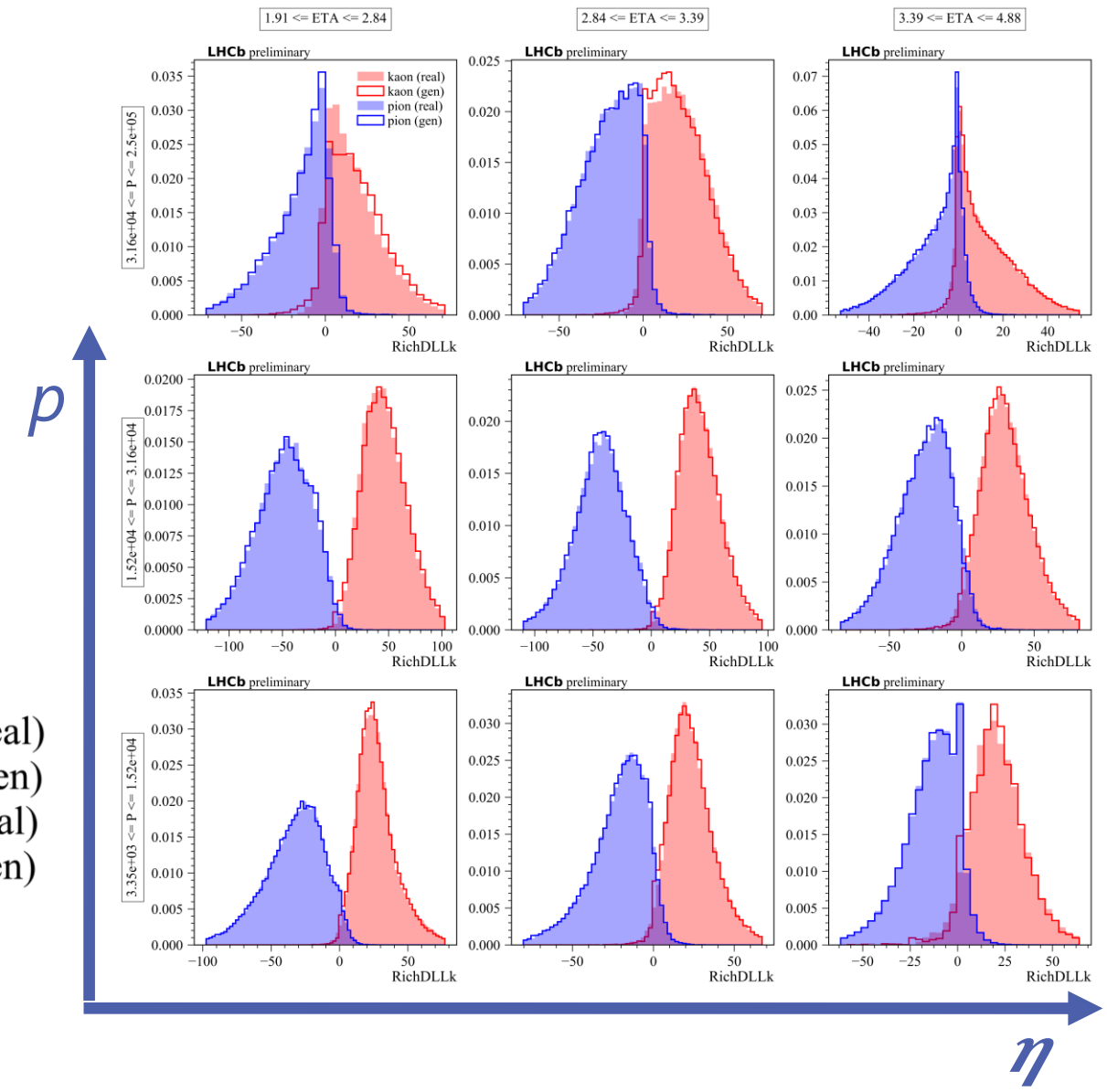
$$\mathcal{E}(X,Y) := 2\,\mathbb{E}\,\|X-Y\|_2 - \mathbb{E}\,\|X-X'\|_2 - \mathbb{E}\,\|Y-Y'\|_2$$

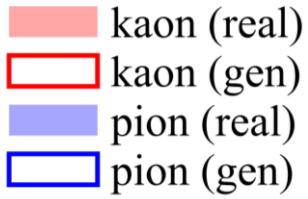


RichDLLk (π vs K)

- Training and testing is performed on real data
- Several clean data samples are selected
- Caveat: the quality metrics only shows the description of particular decays

3x3 bin plot over full P-ETA range





Picture from:
A Maevskiy et al 2020 J. Phys.: Conf. Ser. 1525 012097

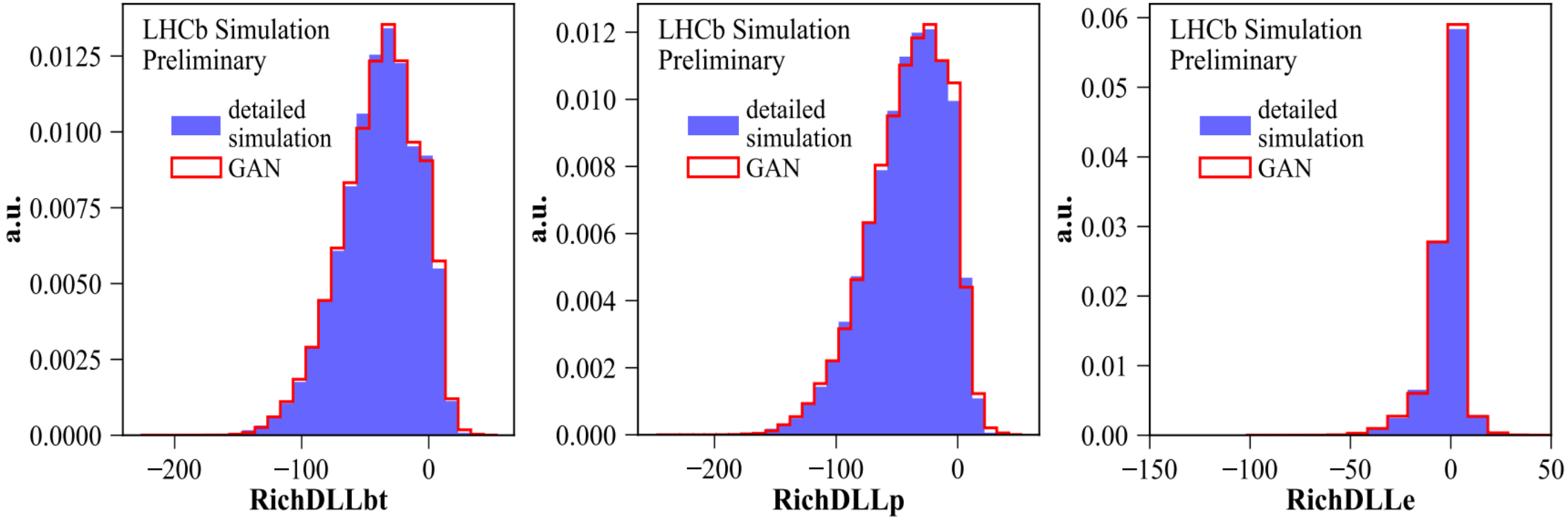
Results on decays into muons

- Training on muons from a mixture of simulated events:
  - Inclusive J/ψ
  - $B^{\pm} \to J/\psi(\mu^{+}\mu^{-})K^{\pm}$

- Evaluating on simulated events for $B^{\pm} \to K^{*\pm}\mu^{+}\mu^{-}$

GAN Simulation Results

- The distributions of DLLx variables for detailed simulation and GAN generated data are in good agreement



- $B^{\pm} \to K^{*\pm}\mu^{+}\mu^{-}$
- But we do not want to compare global distributions, but rather understand how our GANs work in specific cases

Evaluation metric

- We measure the **efficiency** of RichDLLx cuts at various quantiles of the RichDLLx distribution:

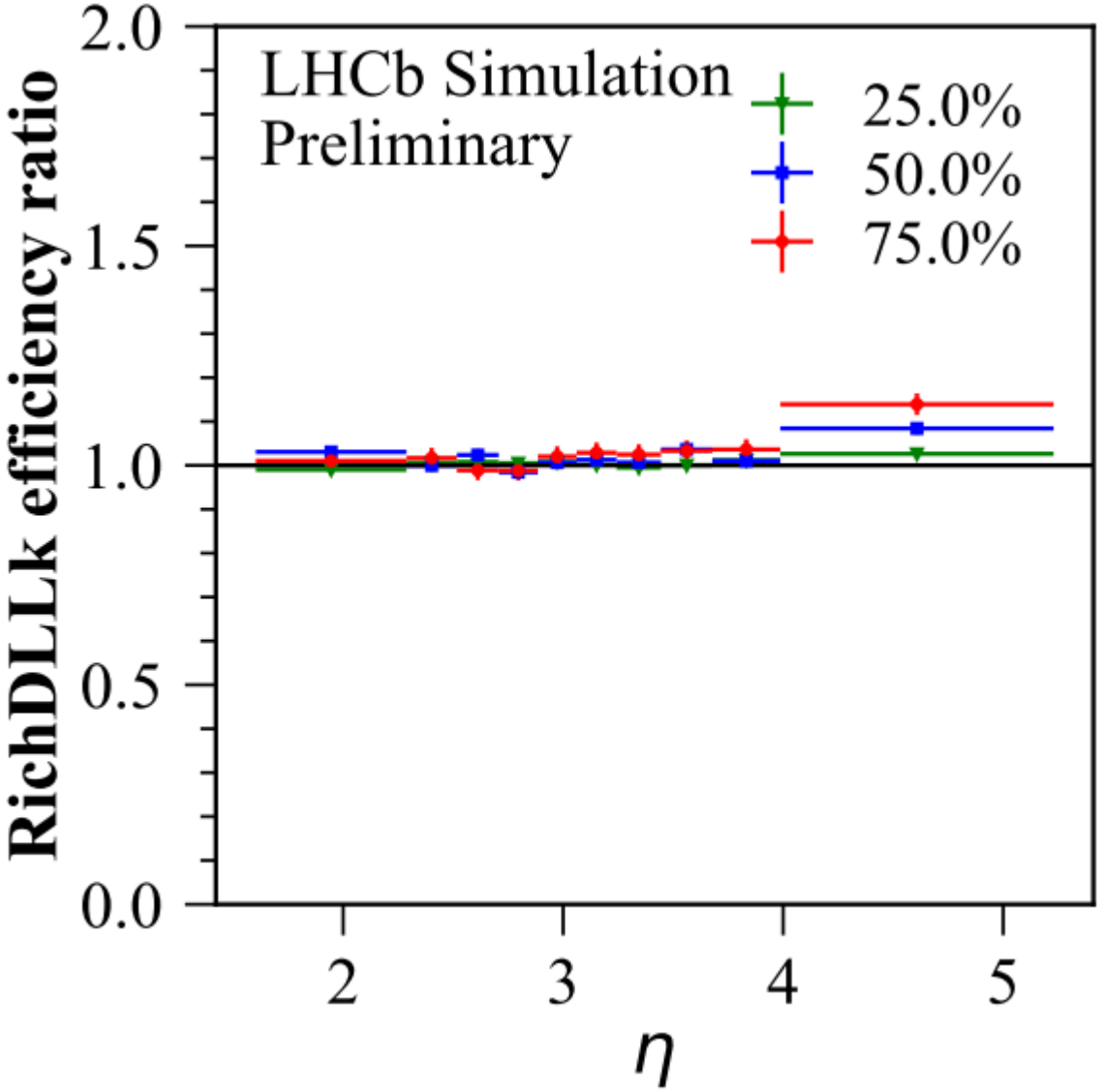
$$\varepsilon = \frac{number\ of\ tracks\ above\ x\%\ threshold}{total\ number\ of\ tracks}$$

- Do this as a function of the input variables:

$$\varepsilon(P, \eta, nSPDHits)$$

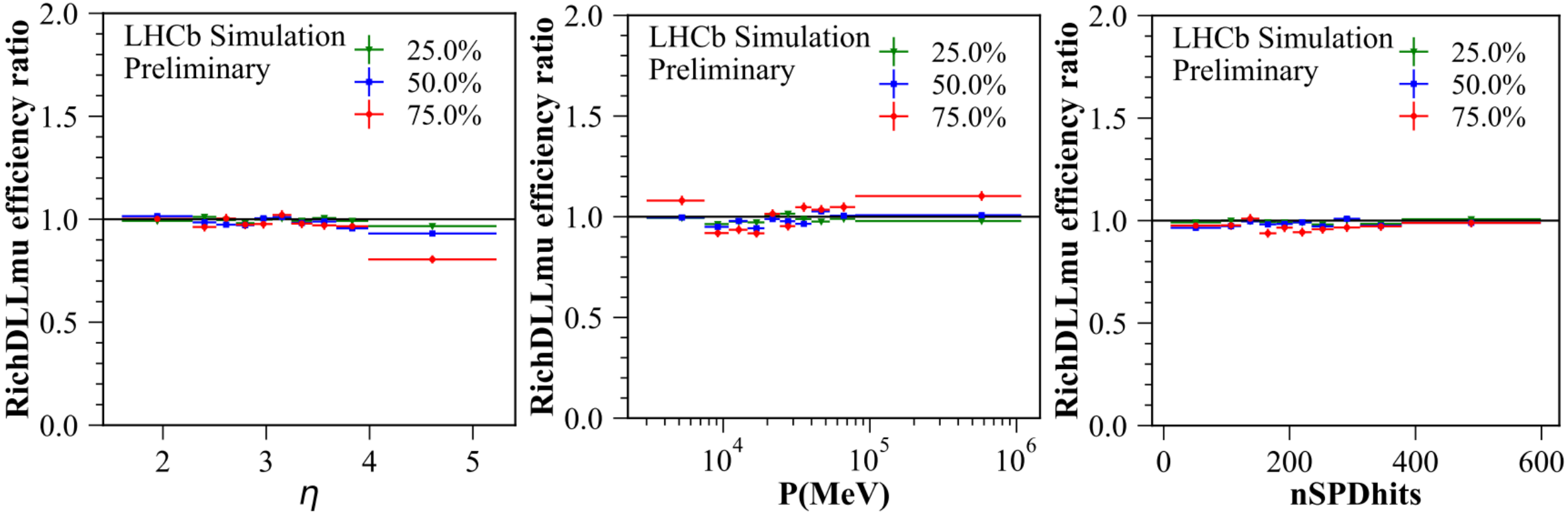
- Calculate the **efficiency ratio** between GAN predictions and simulated events (in bins of a variable):

$$efficiency\ ratio = \frac{\varepsilon_{GAN}}{\varepsilon_{simulated}}$$



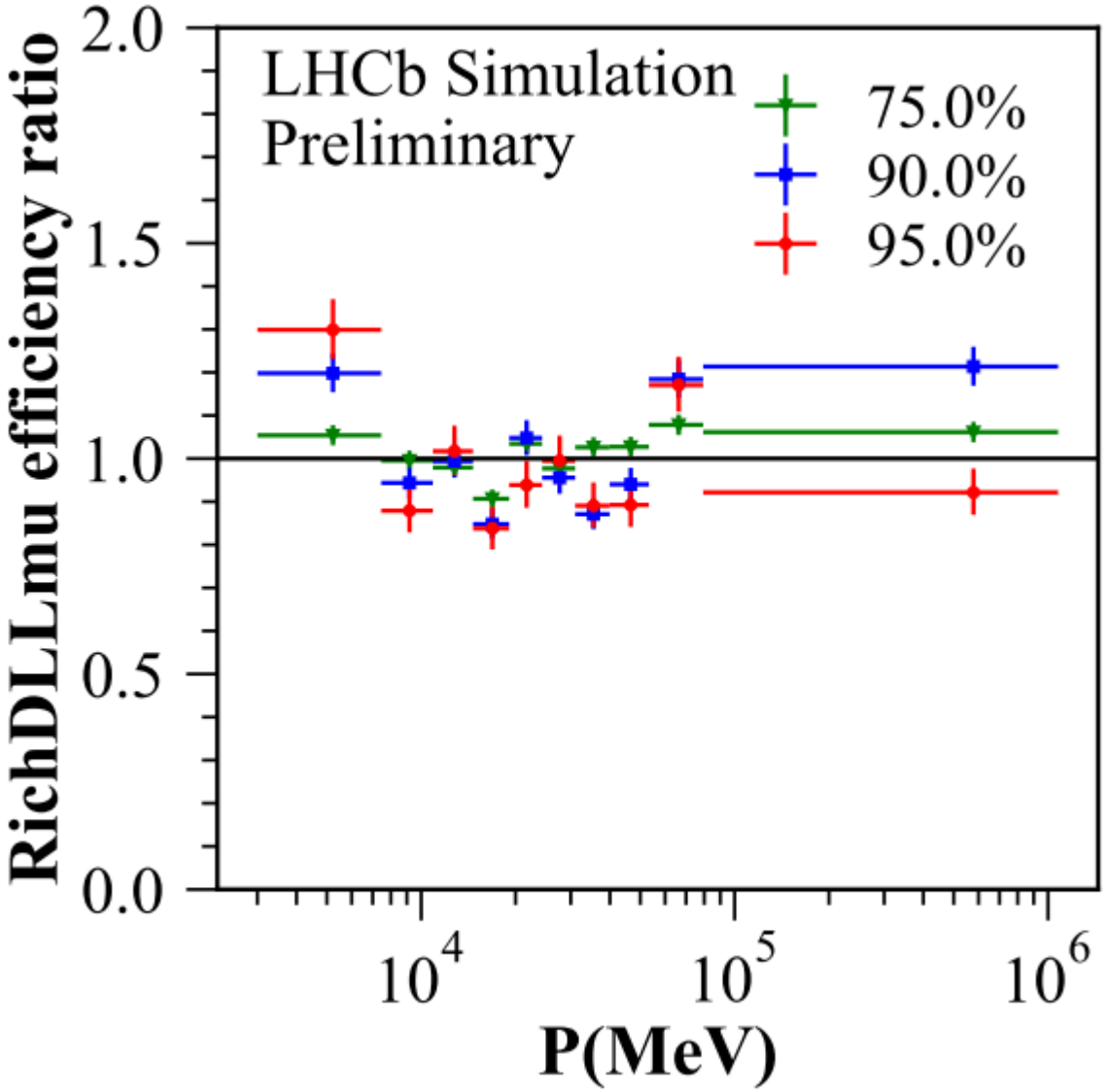
Efficiency Ratio Result (RichDLLmu)

- We trained GAN on simulated data for decays: Inclusive J/ψ and $B^{\pm} \to J/\psi(\mu^{+}\mu^{-})K^{\pm}$
- The ratio of efficiencies between GAN predictions and simulated events for decay $B^{\pm} \to K^{*\pm}\mu^{+}\mu^{-}$ is presented



Tails of distributions

- The quality of agreement degrades on tails of distributions
- We expect the problem to be less pronounced with the growth of training statistics and complexity of the model
- Tail problems affect small amount of events



Conclusion

- The model trained on some decay channels works on other decay channels
- The systematic effects are well under control – we have good efficiency ratios (some tail effects can be seen)
- GAN-based simulation is robust for the kinematic characteristics change
- Our implementation boosts the simulation speed by at least a factor of x100 wrt detailed simulation, depending on the production hardware
- We are working towards neural-network driven description of LHCb detector