Fast Simulation Using Generative Adversarial Networks in LHCb

Lucio Anderlini, Maksim Artemev, Denis Derkach, Nikita Kazeev, Ruslan Khaidurov, Artem Maevskiy, Fedor Ratnikov, Andrey Ustyuzhanin, Egor Zakharov
On behalf of the LHCb collaboration

3rd IML Machine Learning Workshop
15-18 April 2019
CERN

NATIONAL RESEARCH UNIVERSITY
SCHOOL OF DATA ANALYSIS
SAPIENZA UNIVERSITÀ DI ROMA
INFN Istituto Nazionale di Fisica Nucleare
Skoltech Skolkovo Institute of Science and Technology

Outline

- Typical simulation workflow
  - Where GANs can be used?
- Fast Simulation with GANs in context of LHCb
  - GANs for calorimeter
  - GANs for the Cherenkov detectors

Typical simulation workflow

Event Gen → **GEANT4** → **Digitization** → **Reconstruction** →

- One may imagine any part of this chain to be replaced by GAN

Typical simulation workflow

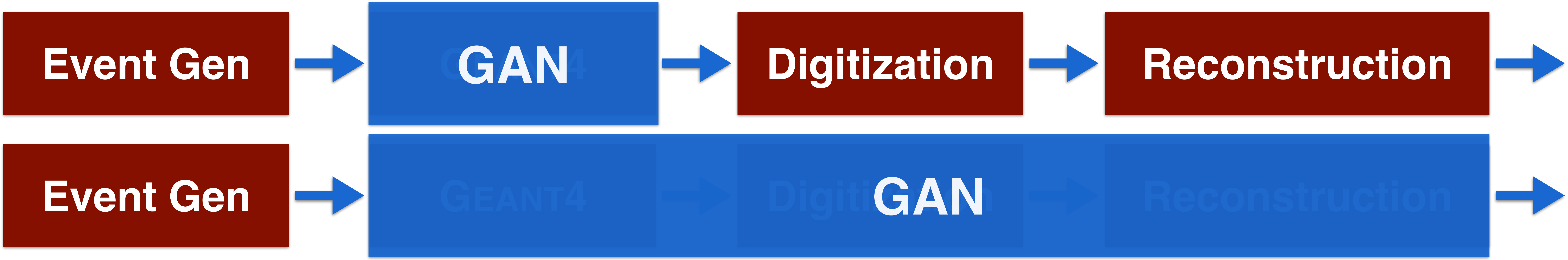
Event Gen → **GEANT4** → **Digitization** → **Reconstruction** →

- One may imagine any part of this chain to be replaced by GAN
- Here we demonstrate two approaches:

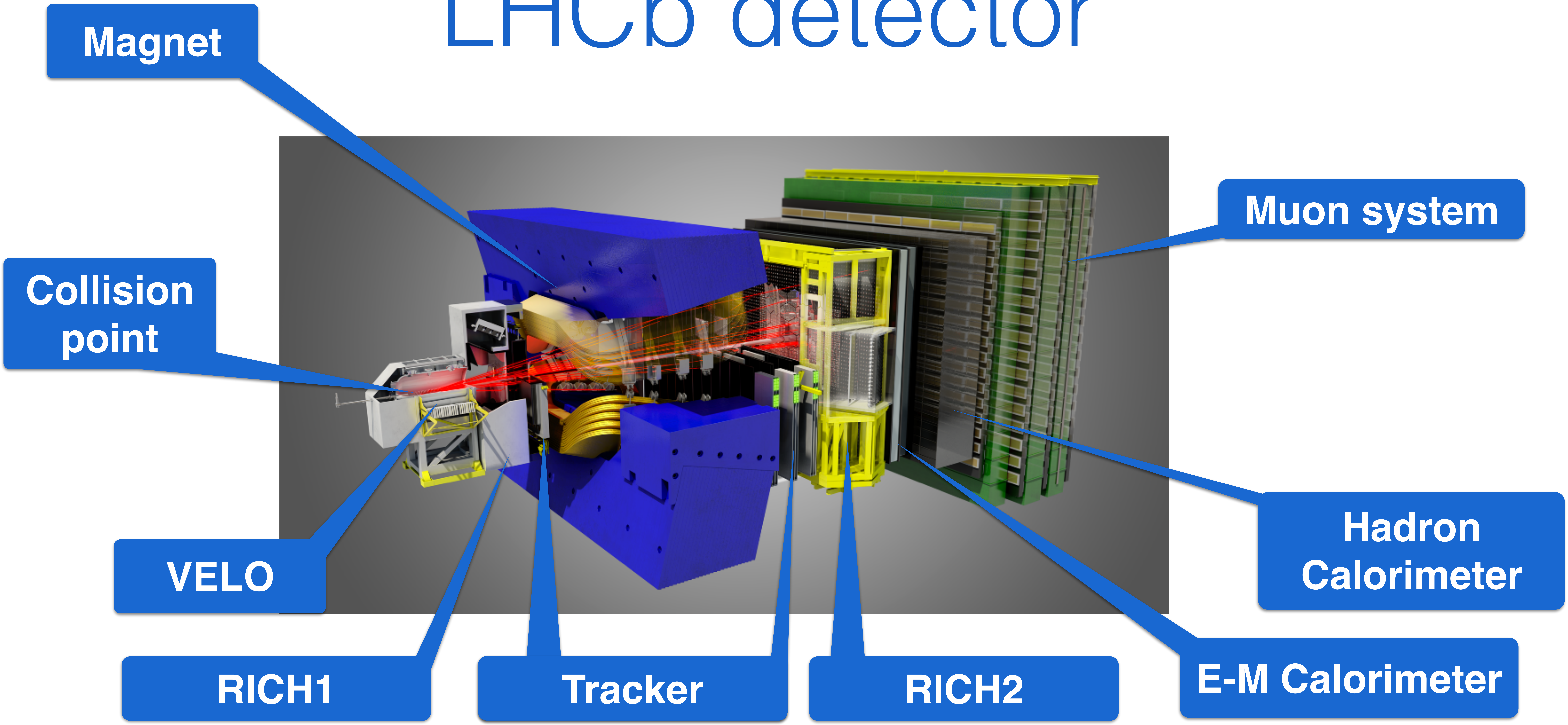
Event Gen → **GAN** → **Digitization** → **Reconstruction** →

Typical simulation workflow

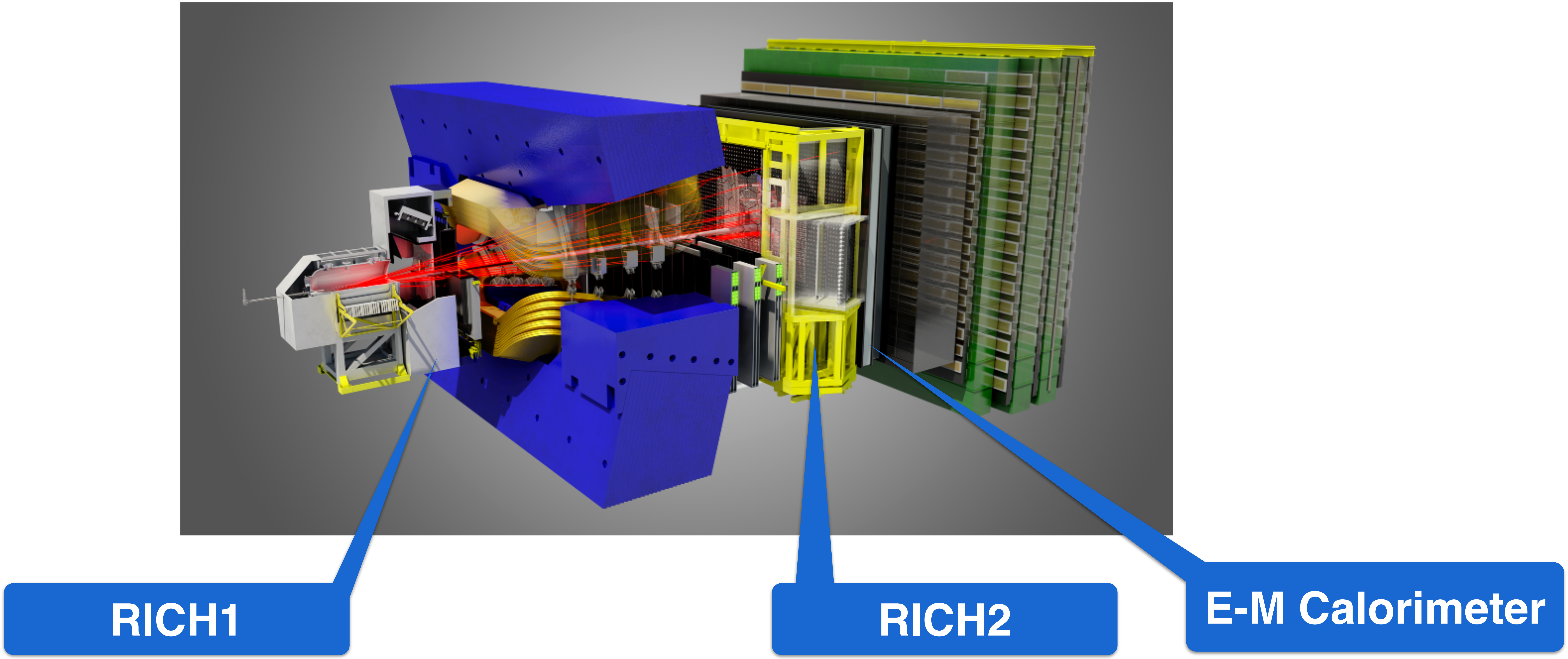
- One may imagine any part of this chain to be replaced by GAN
- Here we demonstrate two approaches:



LHCb detector



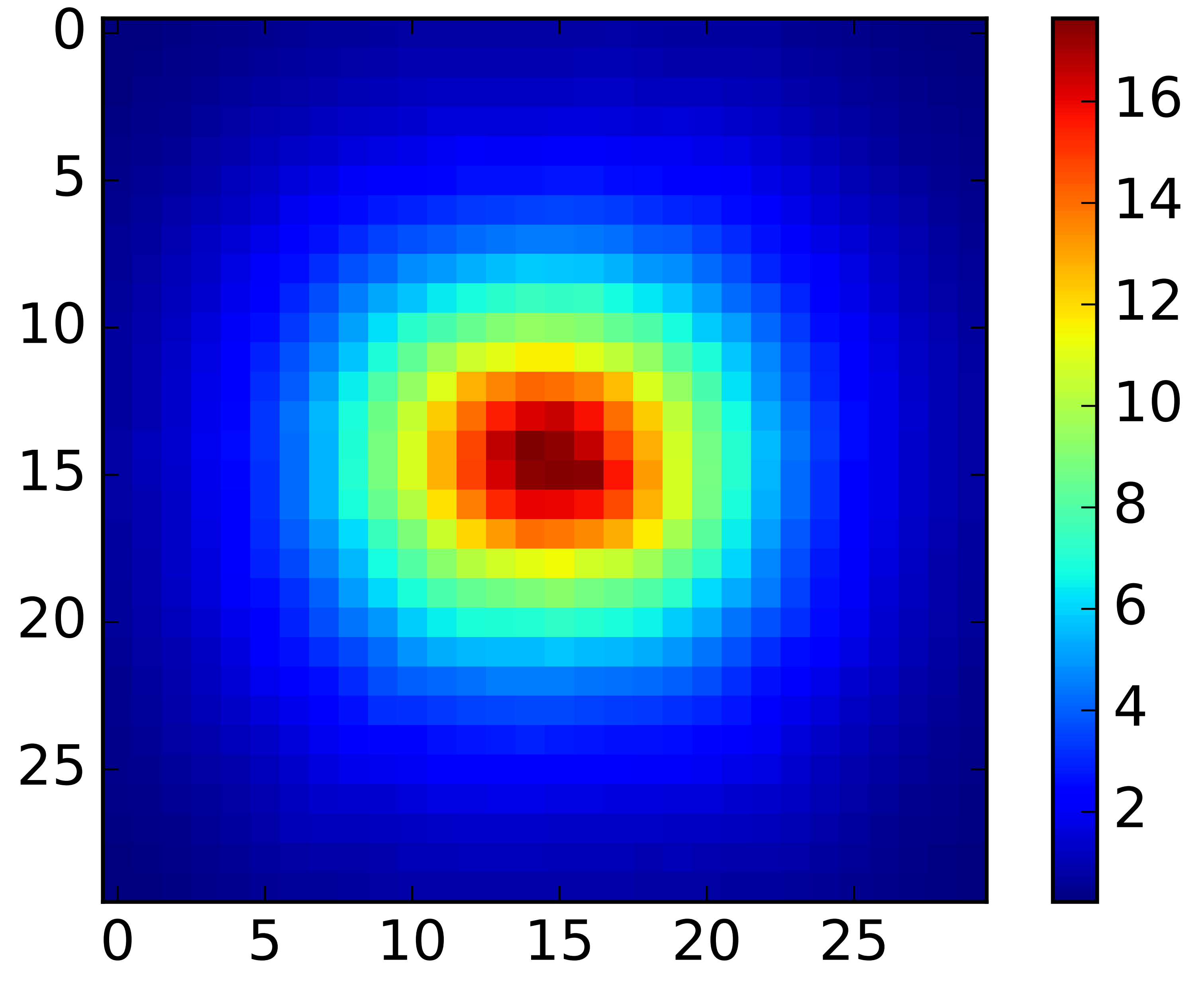
LHCb detector



FAST CALORIMETER SIMULATION

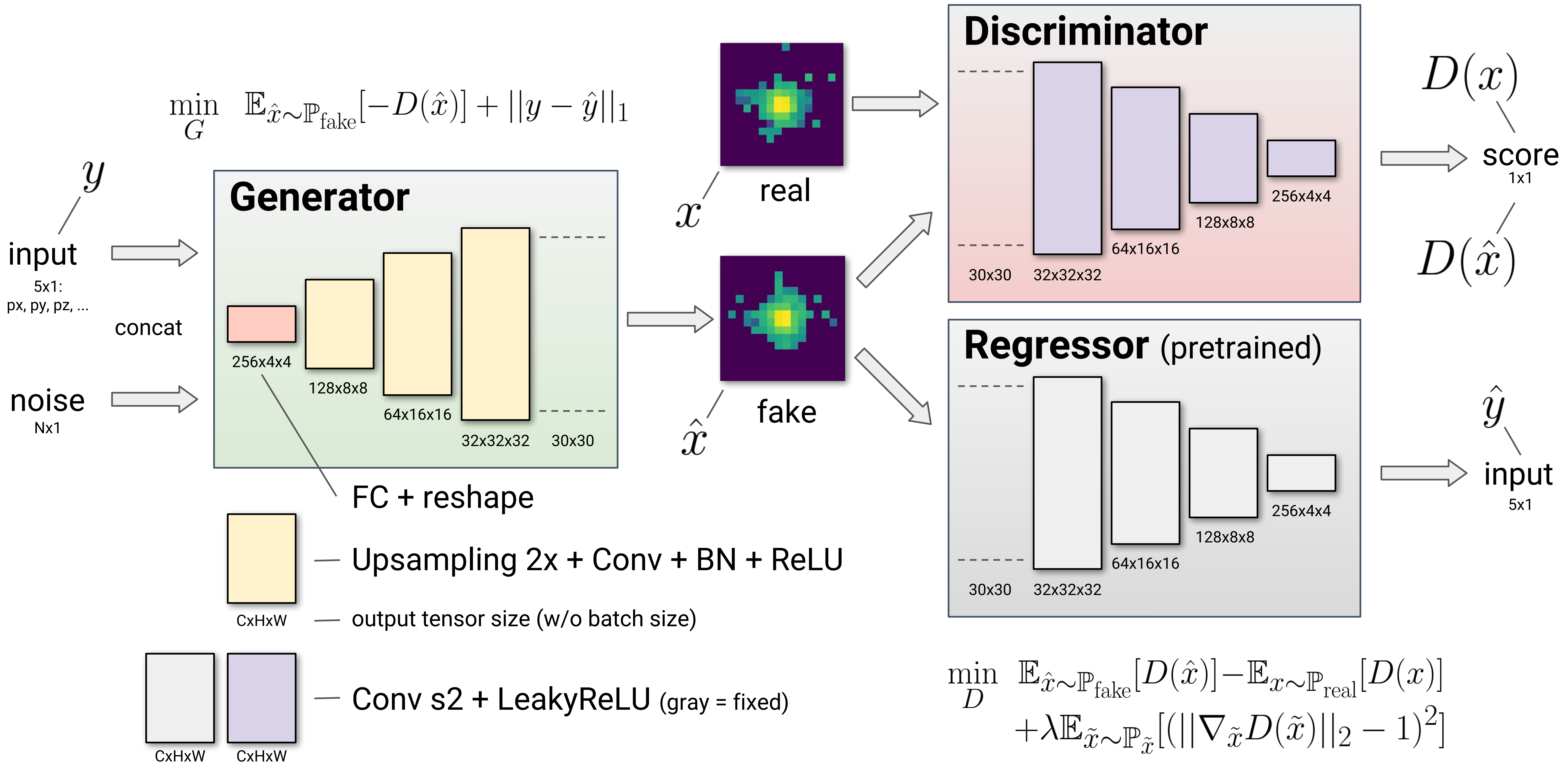
Setup

- LHCb inspired calorimeter in GEANT4
  - 30x30 cells
- 5 conditional parameters per particle
  - 3D momentum
  - 2D coordinate
- Electrons from particle gun shot at 1x1 cm square at the center of the calorimeter face

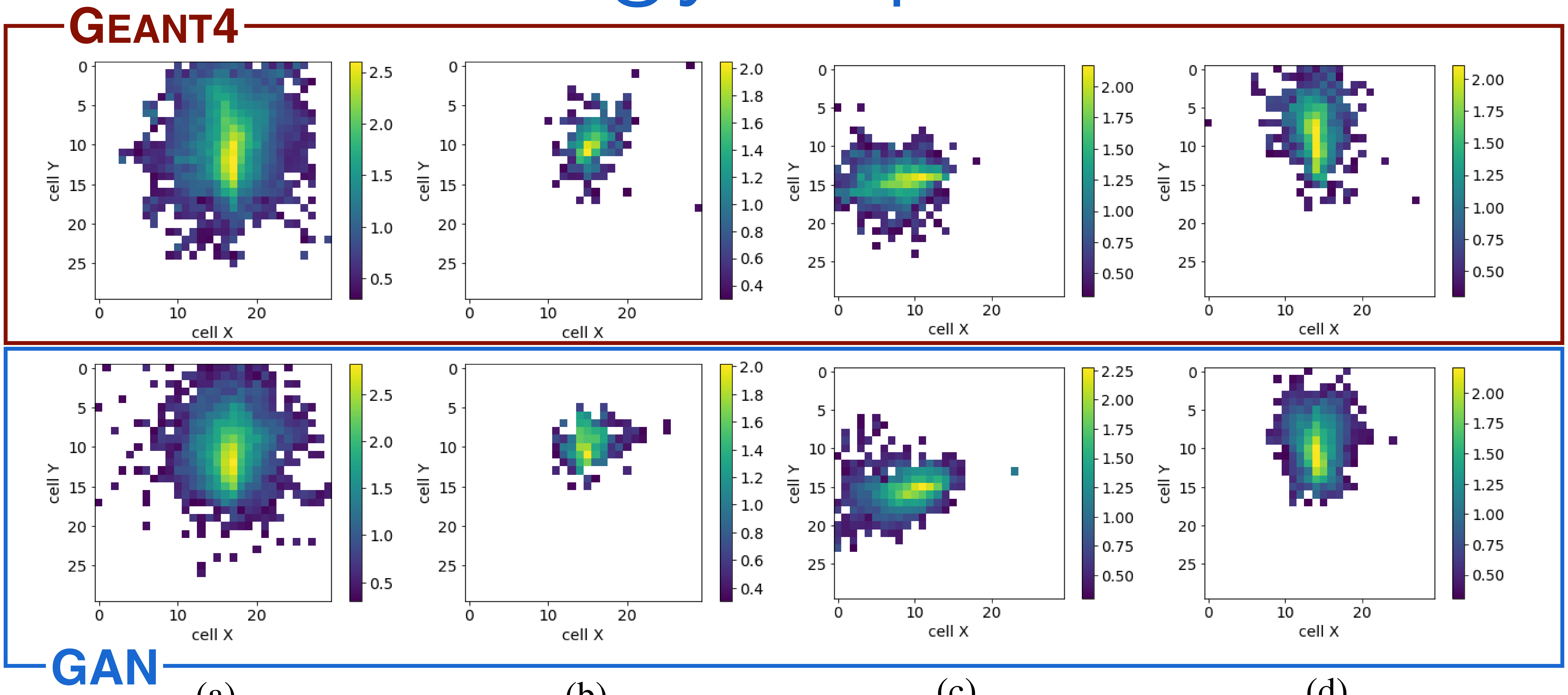


Average cluster

Model architecture



Energy deposits



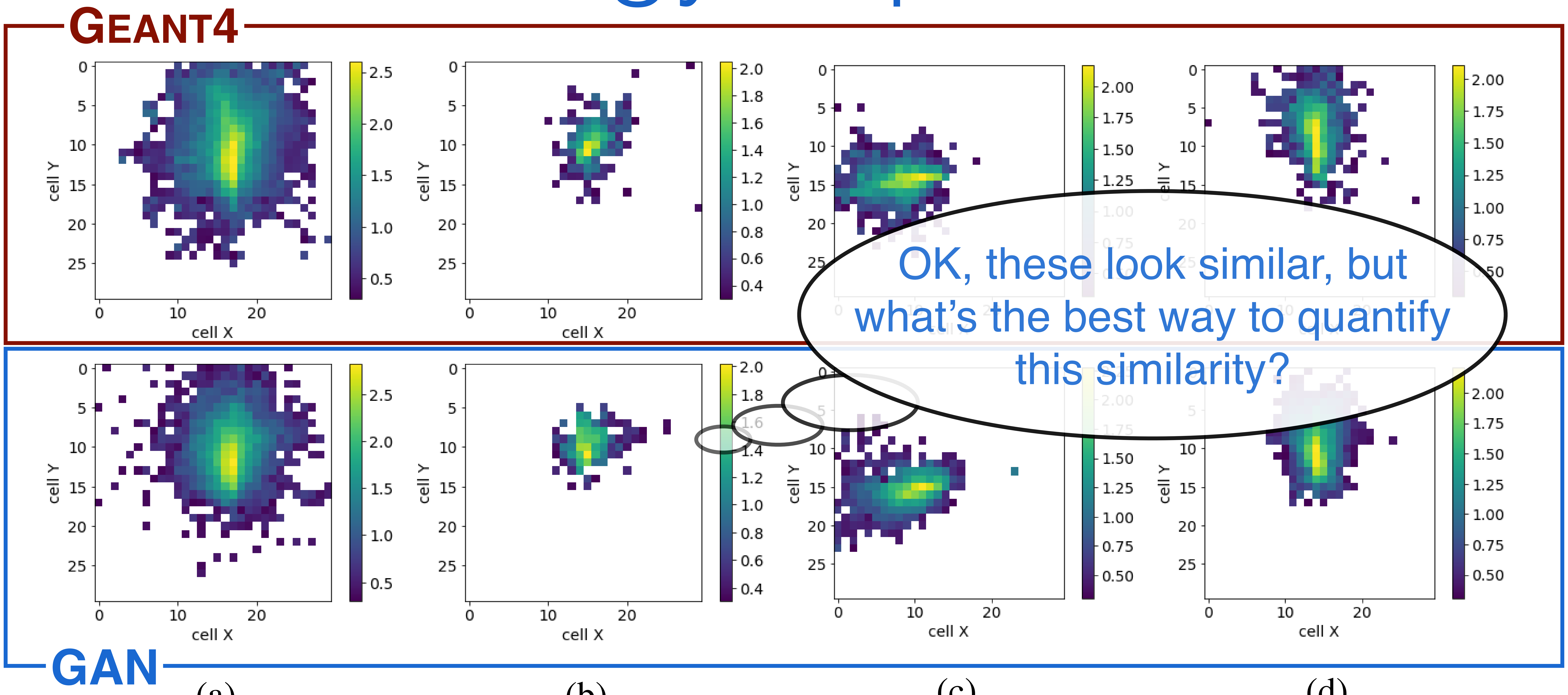
(a) $E_0 = 63.7$ GeV

(b) $E_0 = 6.5$ GeV

(c) $E_0 = 15.6$ GeV

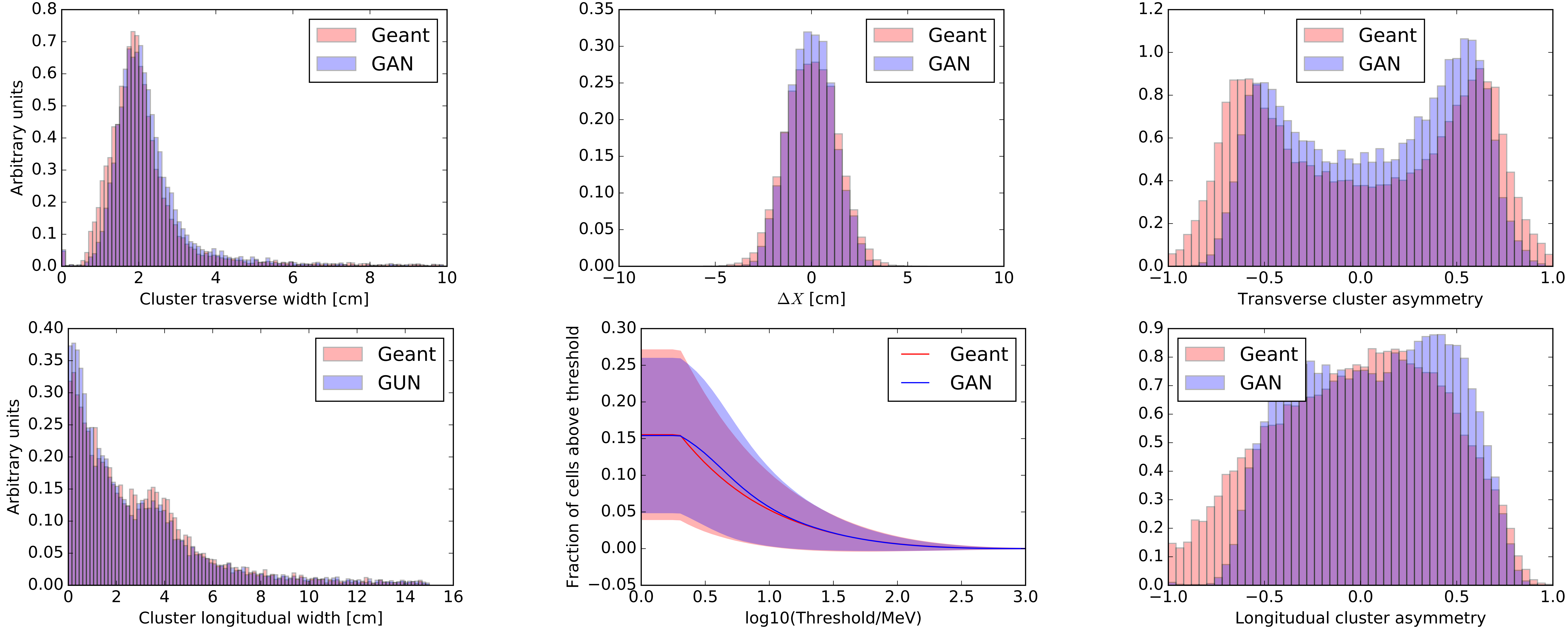
(d) $E_0 = 15.9$ GeV

Energy deposits



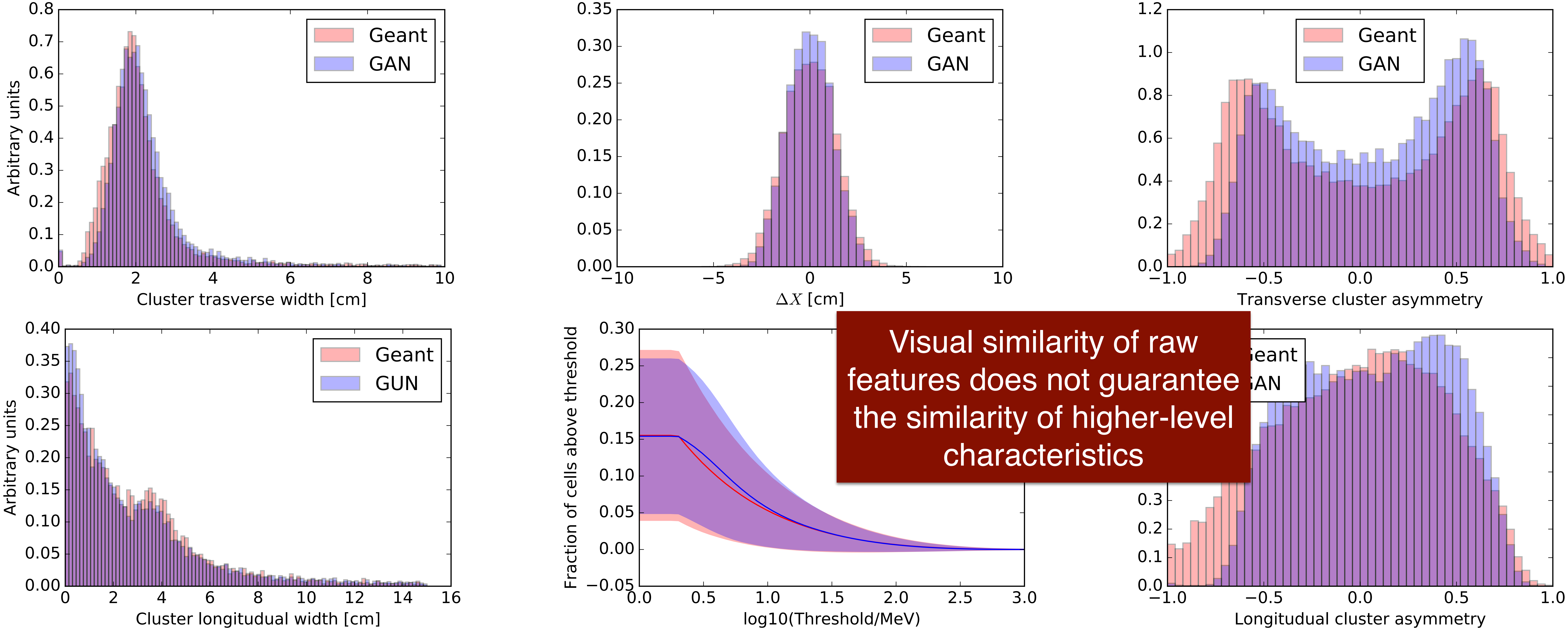
(a) $E_0 = 63.7$ GeV (b) $E_0 = 6.5$ GeV (c) $E_0 = 15.6$ GeV (d) $E_0 = 15.9$ GeV

Physics-motivated characteristics



- Idea: convert each cluster to a meaningful single value
  - Then compare real vs generated distributions of such values

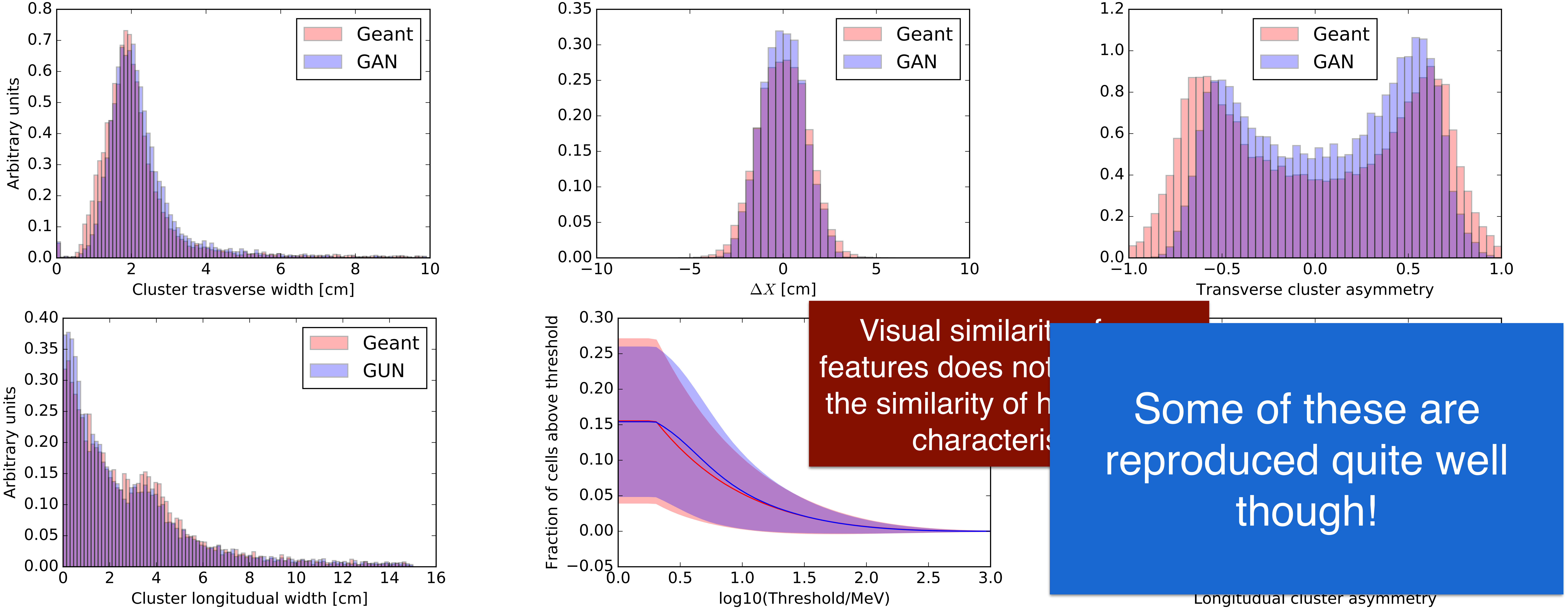
Physics-motivated characteristics



Visual similarity of raw features does not guarantee the similarity of higher-level characteristics

- Idea: convert each cluster to a meaningful single value
  - Then compare real vs generated distributions of such values

Physics-motivated characteristics



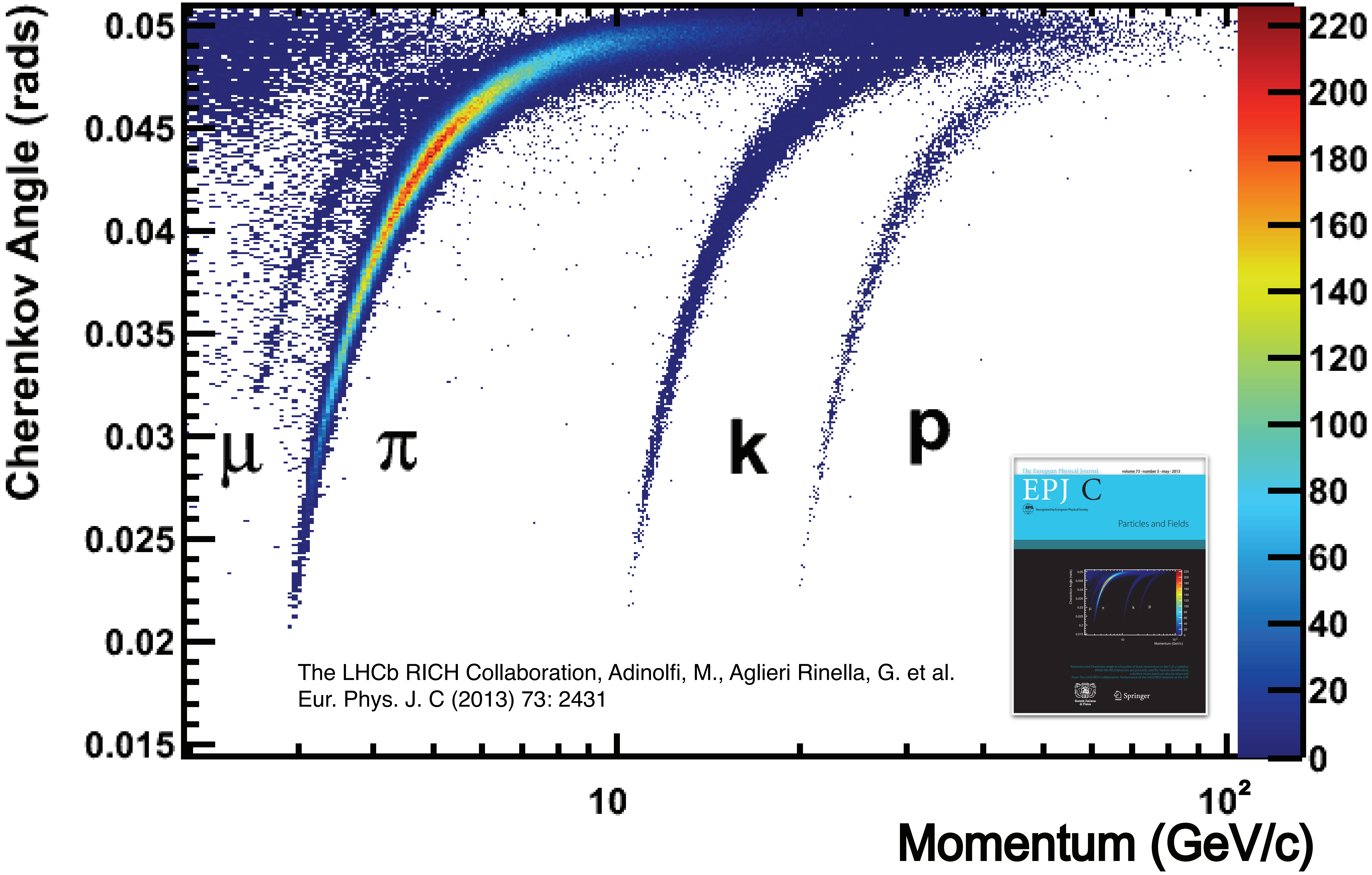
Visual similarit... features does not... the similarity of h... characteris...

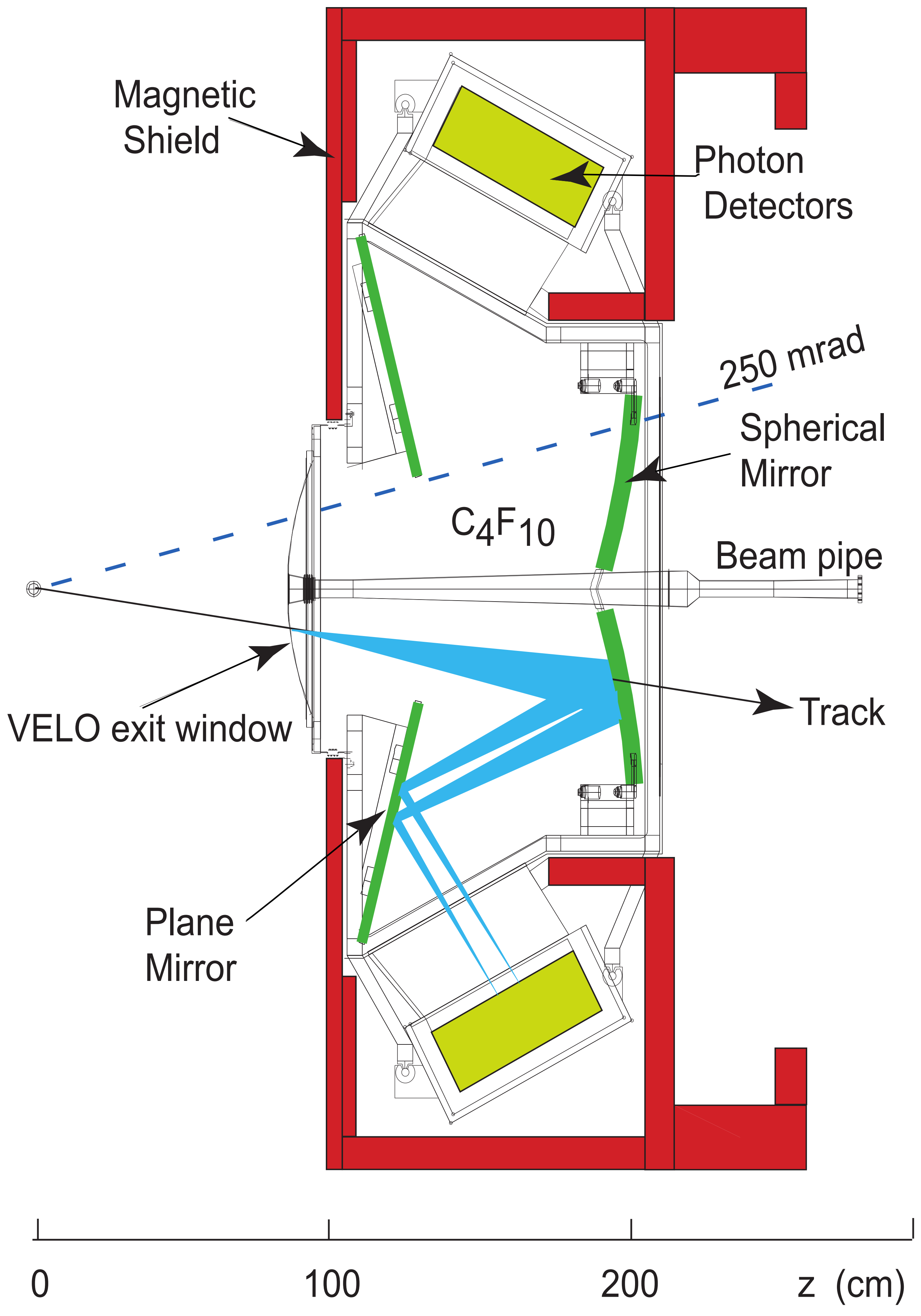
Some of these are reproduced quite well though!

- Idea: convert each cluster to a meaningful single value
  - Then compare real vs generated distributions of such values

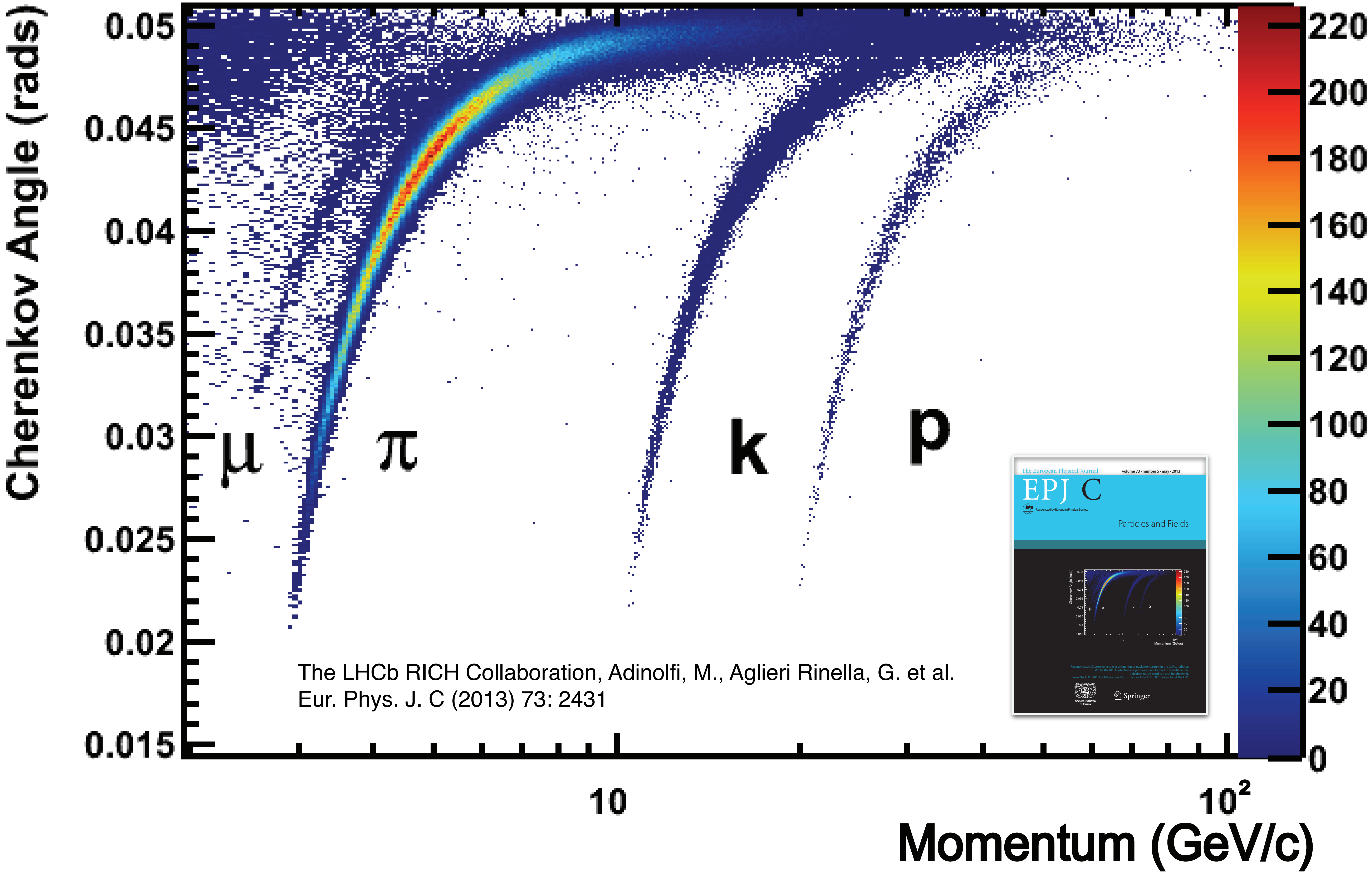
FAST CHERENKOV DETECTOR SIMULATION

Ring Imaging Cherenkov Detectors (RICH)





Ring Imaging Cherenkov Detectors (RICH)



- PID with RICH is done with a global log-likelihood method
- PID information encoded in log-likelihood differences (DLL) between particle type hypotheses

RICH fast simulation

- A possible solution:
  - Bypass all accurate simulation steps from Cherenkov light generation up to the high-level likelihood parameters (DLLs)
  - Learn the distribution of DLLs for given track parameters and sample from it, P(DLLs | <track params>)

Derkach et al, NIMA 2019 (01) 031

RICH fast simulation

- Number of output features:
  - 5 DLLs
- Number of input features:
  - track momentum and pseudorapidity (+2)
  - total number of tracks in that event (+1)

To account for occupancy-related effects

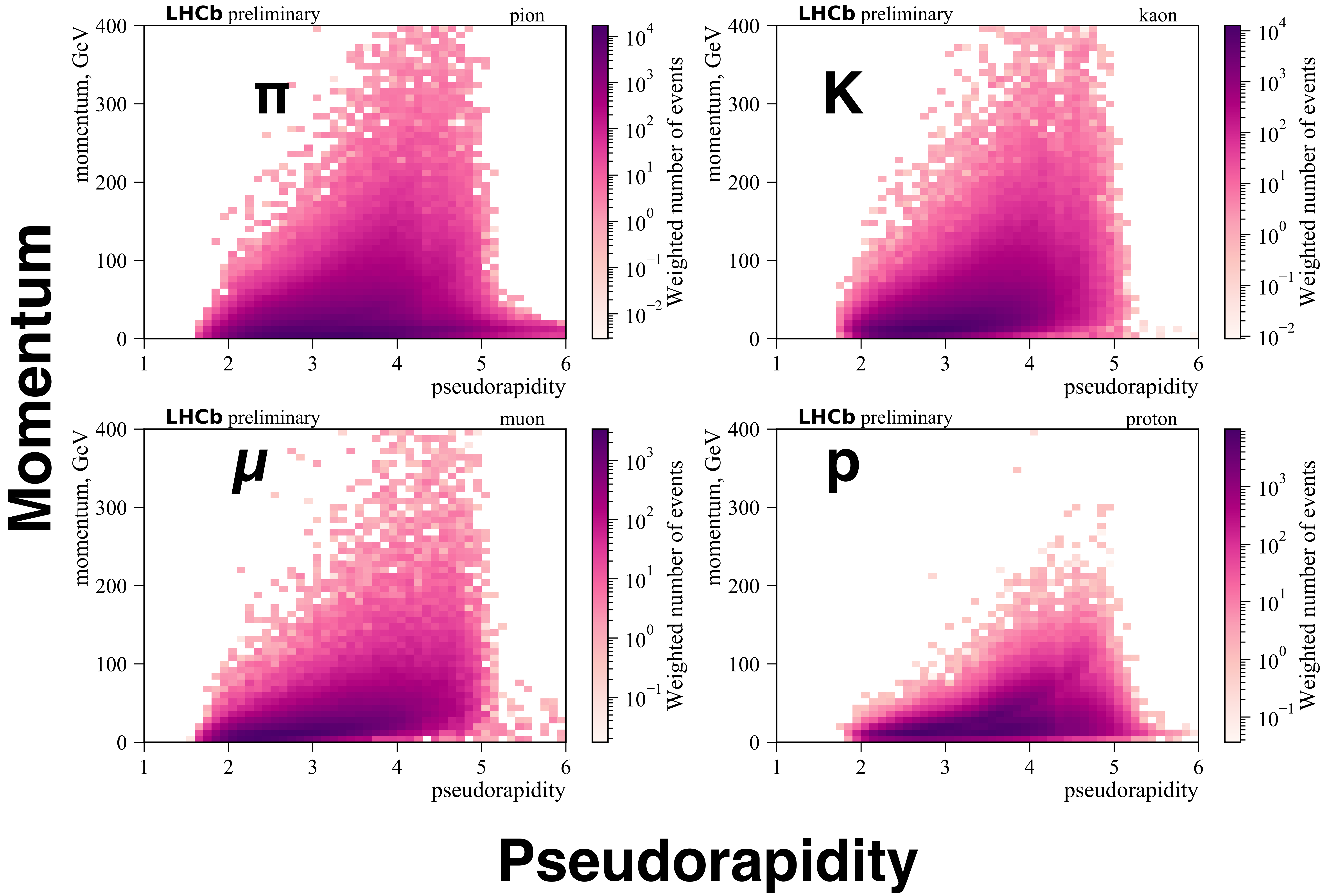
RICH fast simulation

- Number of output features:
  - 5 DLLs
- Number of input features:
  - track momentum and pseudorapidity (+2)
  - total number of tracks in that event (+1)
- Training on real data (calibration channels)
  - using sPlot technique[1] to extract signal distributions
    ⇒ loss function is weighted
    ⇒ some of the weights are negative

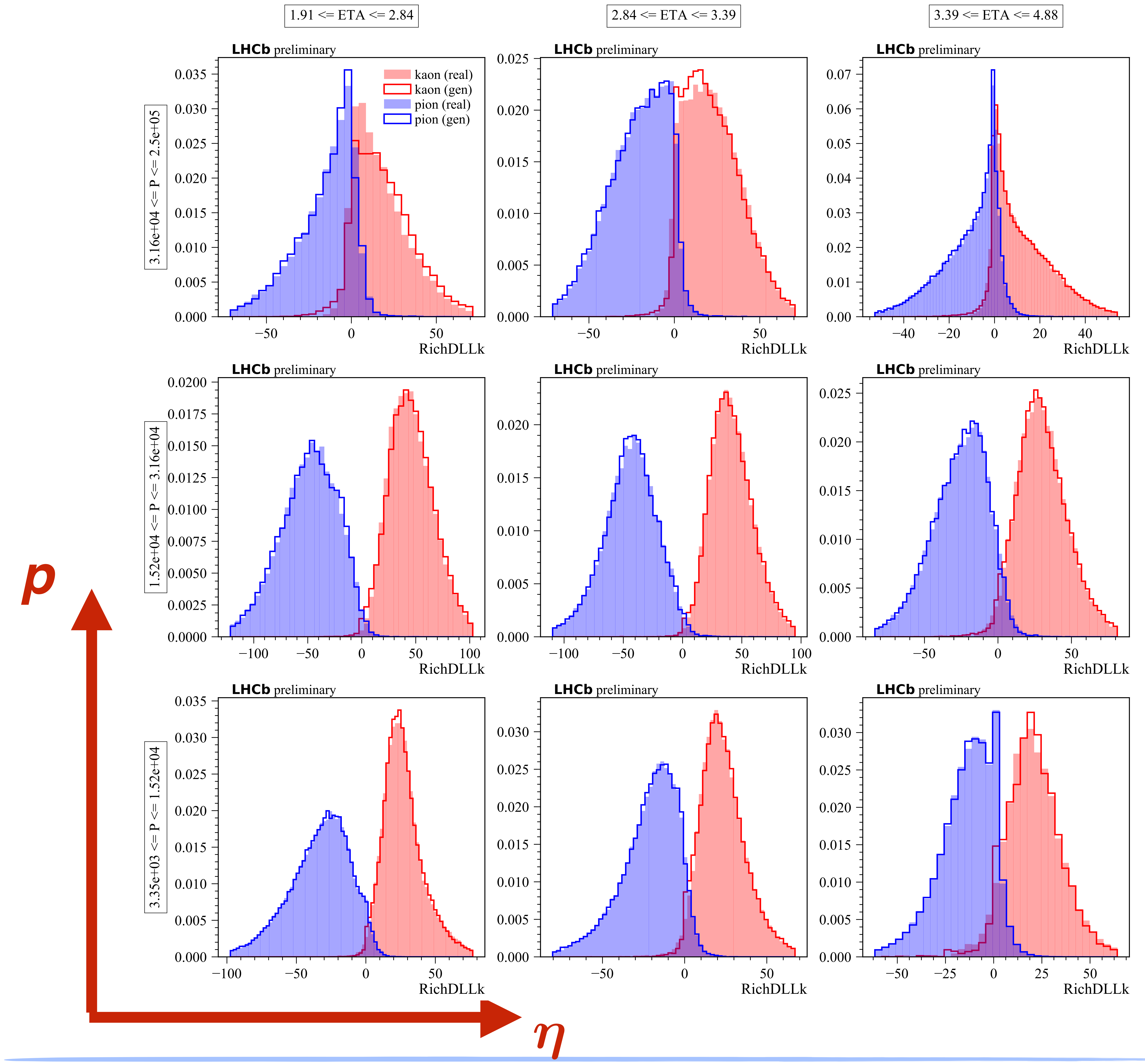
[1]Pivk, Muriel et al. Nucl.Instrum.Meth.A 555 (2005) 356-369

Implementation details and input parameter distributions

- Optimizing the Cramér metric (energy distance), arXiv: 1705.10743
- 10 hidden fully-connected layers for both generator and discriminator
  - 128 neurons each
  - ReLU activation
- 64-dimensional latent space (noise shape)
- 256-dimensional discriminator output
- 15 discriminator updates per 1 generator update
- RMSProp optimizer, exp decaying learning rate



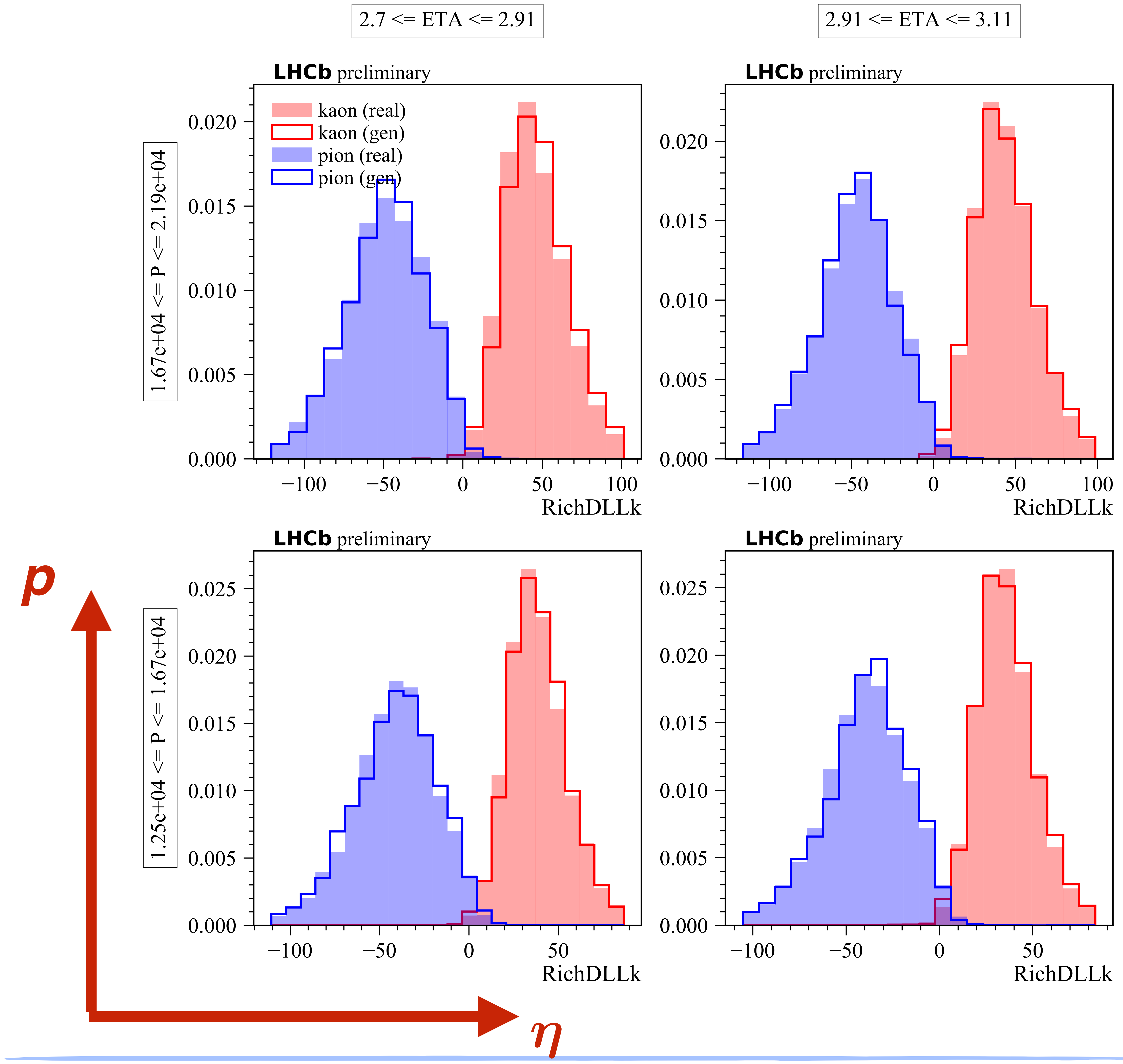
RichDLLk (π vs K)



kaon (real)
kaon (gen)
pion (real)
pion (gen)

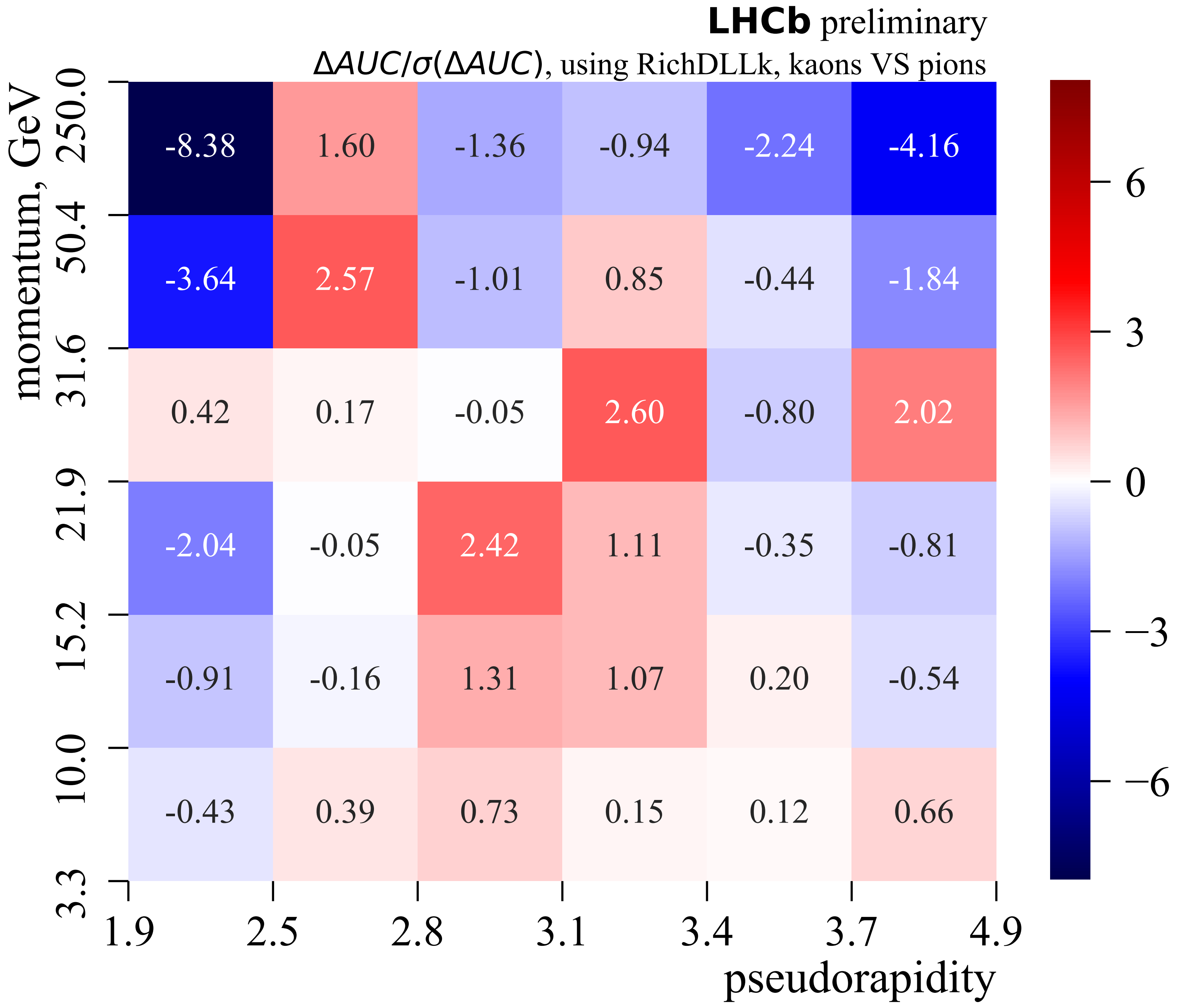
3x3 bin plot over full P-ETA range

RichDLLk (π vs K)

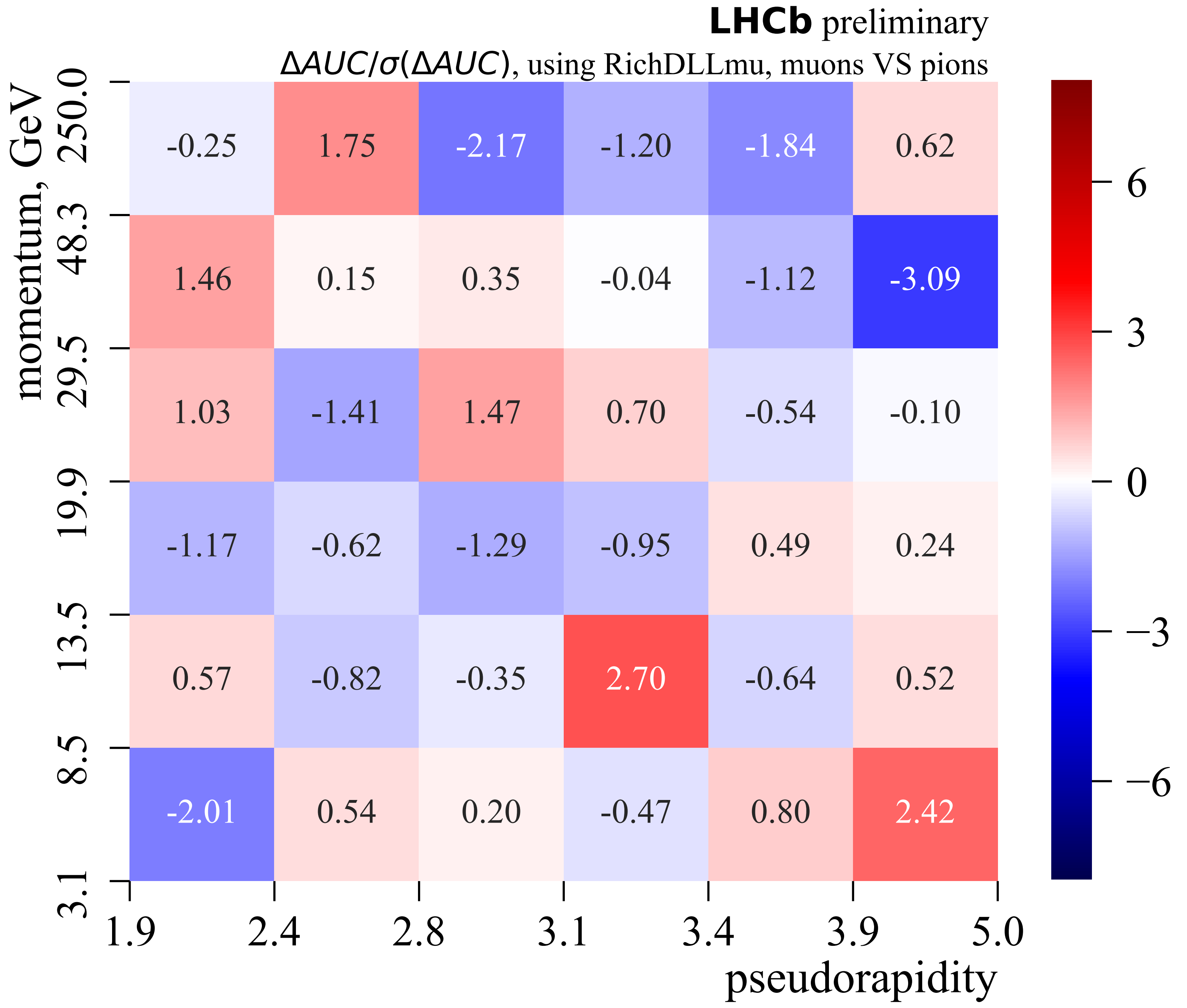


zoomed in to the most populated region

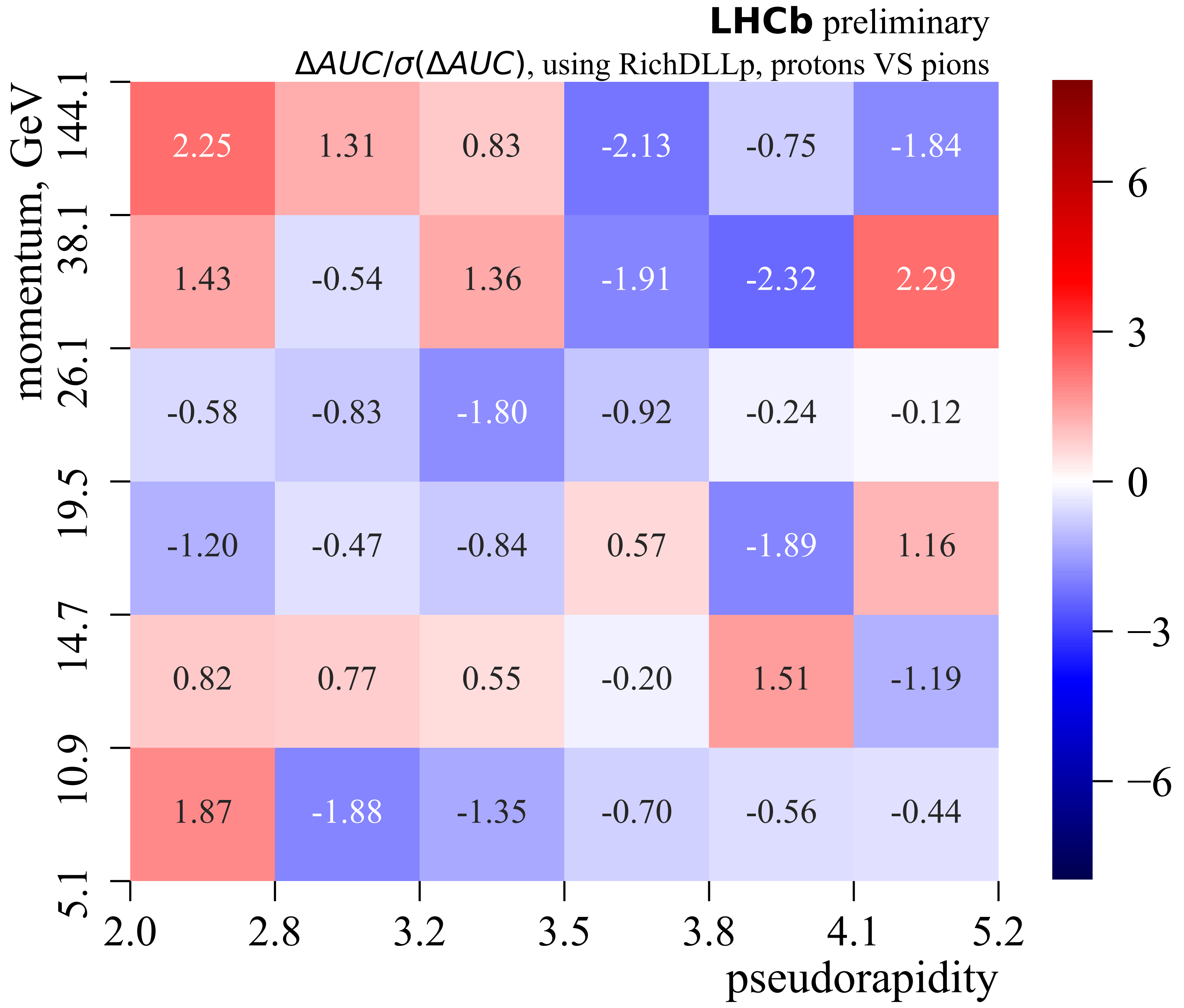
Differences between AUCs for real and generated samples (divided by uncertainty)



K vs π, using RichDLLk



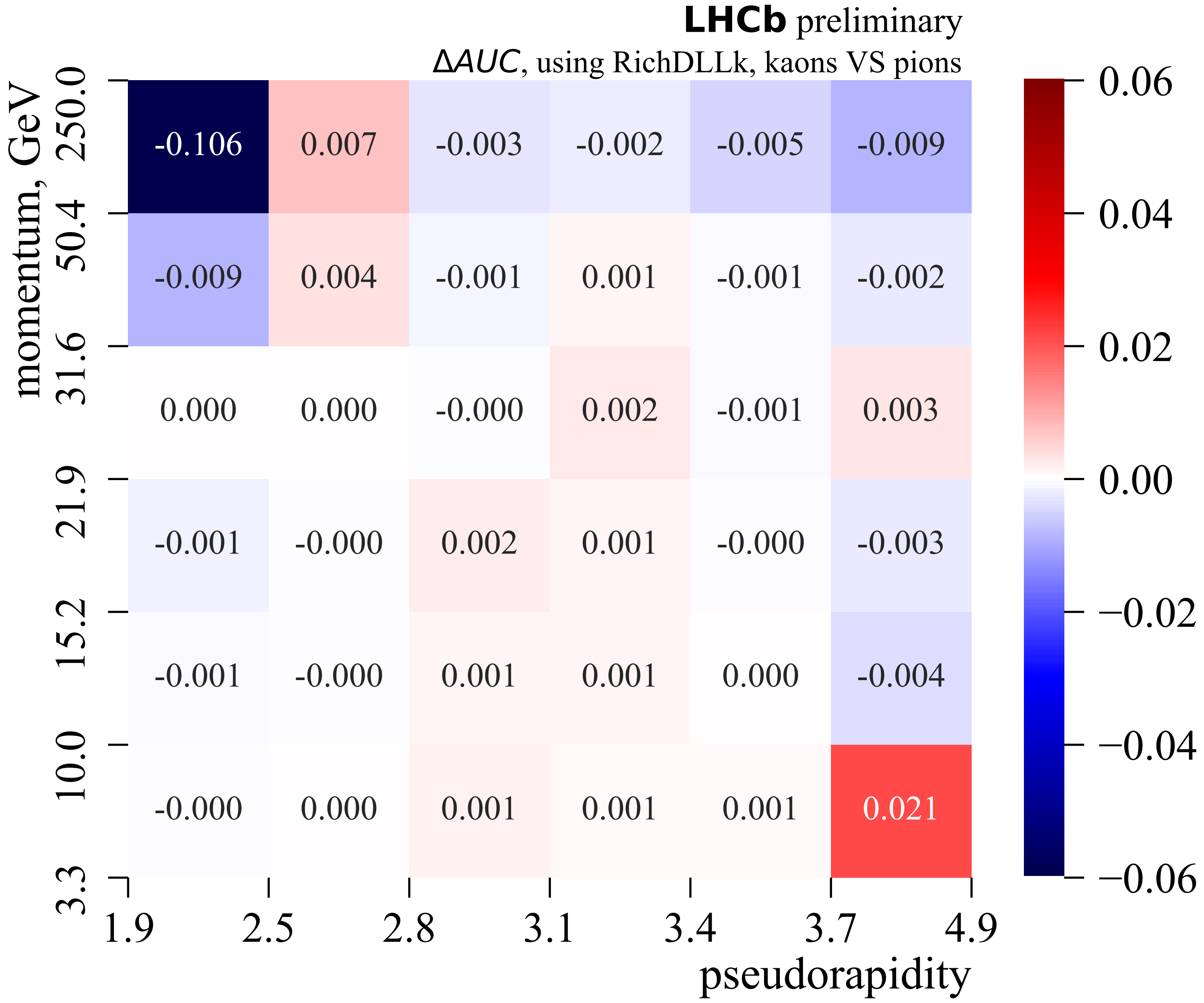
μ vs π, using RichDLLmu



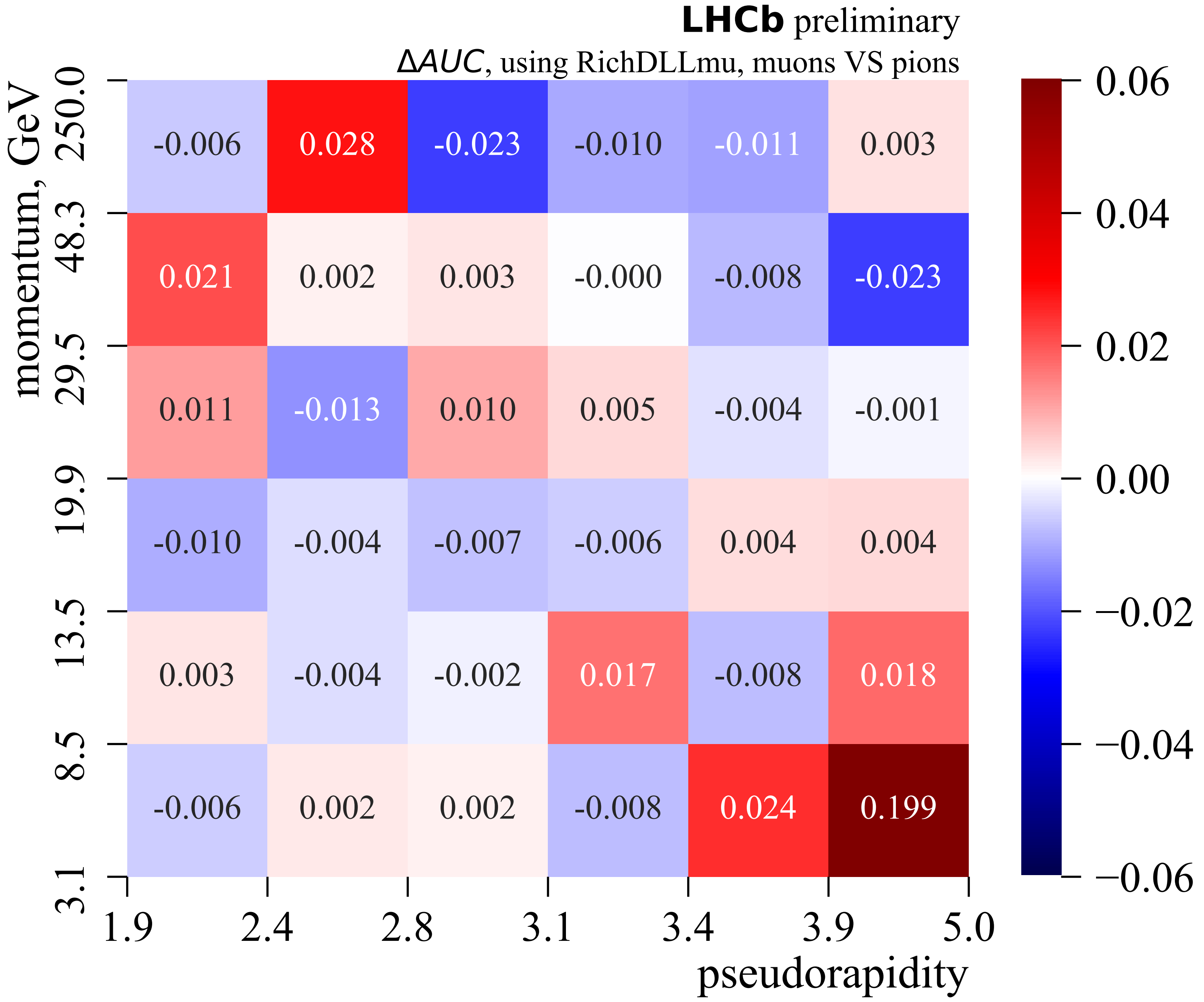
p vs π, using RichDLLp

- Uncertainties estimated using bootstrap technique
- Most differences are within just a few sigmas, larger deviations at low-stat regions

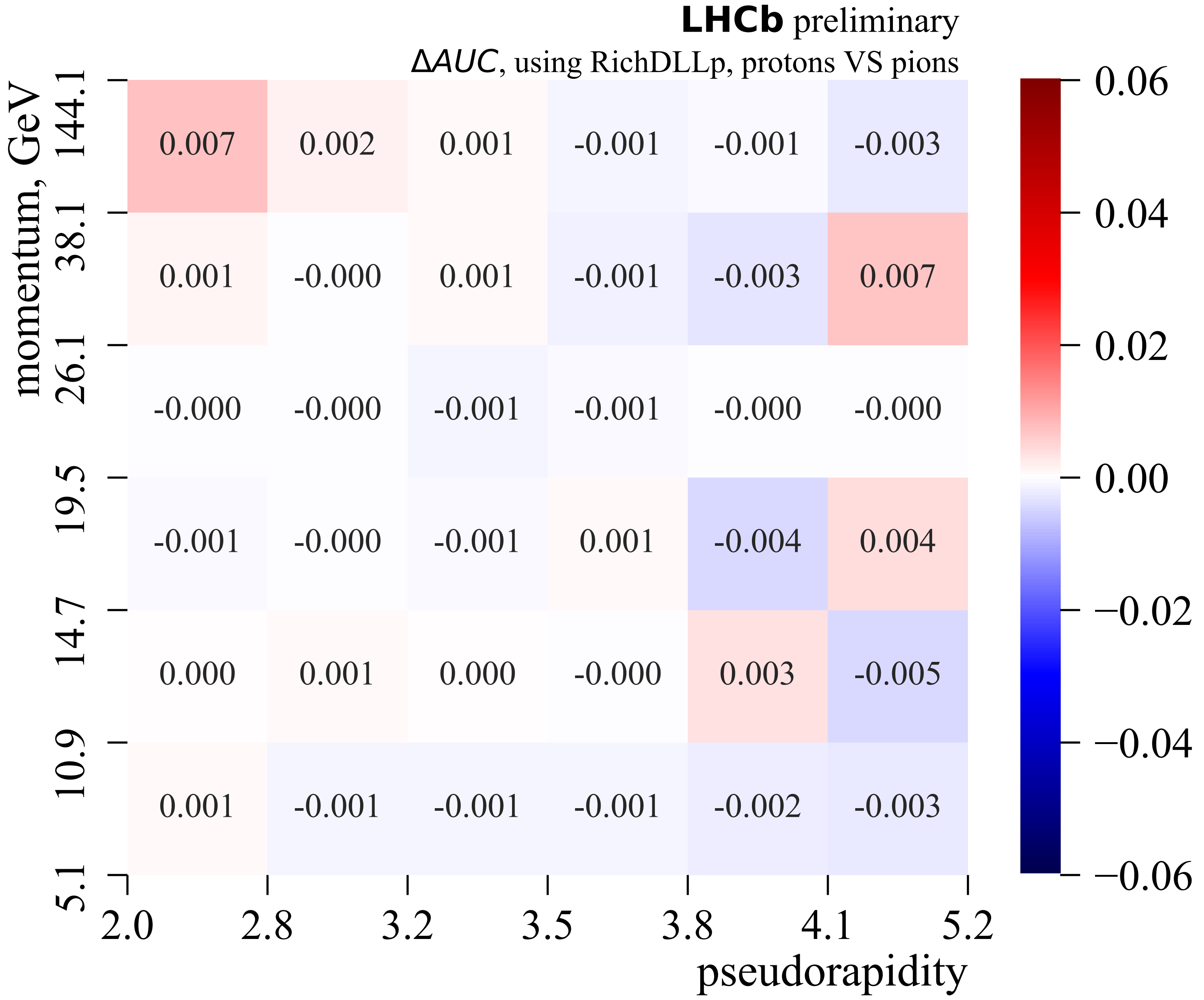
Differences between AUCs for real and generated samples (absolute, generated – real)



K vs π, using RichDLLk



μ vs π, using RichDLLmu



p vs π, using RichDLLp

- Absolute differences between AUCs are mostly in the 0.001-0.01 range

Summary

- **GANs** are a promising tool for fast simulation
  - Can be trained in background-contaminated environment
  - Two approaches shown:
    - generating either **low-**
    - or **high-level** parameters
- Evaluating a generative model performance is a challenge itself
  - DeltaAUC agreement with 0 – necessary but not sufficient criterion
  - For low-level case – important to compare physics-motivated parameters (real vs generated)
  - The 'ultimate' way: **test in a physics analysis environment**
    - **work in progress**

Backup

Generative Adversarial Networks (GANs)

x - input variables
y - target variables

P(DLLs | <track params>)

y x

- Random variable z
- Two deterministic functions (neural nets)
  - generator G(x, z)
  - discriminator D(x, y)
- Generator maps (x, z) to y^{gen}
- Discriminator distinguishes between (x, y^{gen}) and (x, y^{real})
- Training step («competition» between the two nets):
  - train discriminator to improve (x, y^{gen}) and (x, y^{real}) separation
  - train generator to increase the discriminator's error rate

Discriminator metric

- Some of the options for the discriminator metric:
    - Binary cross-entropy between the real and generated samples
        - Equilibrium when Jensen–Shannon divergence is minimized
        - Problematic for distributions with different support; mode collapse problems
    - Wasserstein (aka Earth Mover's) distance
        - Discriminator => «Critic» (evaluates the metric)
        - Naturally solves the non-equal support and mode collapse problems
        - Suffers from biased gradients

arXiv:1406.2661
arXiv:1701.07875

GAN (JS)

- Possible discriminator and generator losses – binary cross-entropy:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

real samples **discriminator NN** **noise samples** **generator NN**

arXiv:1406.2661

- This leads to equilibrium when Jensen-Shannon divergence between real and generated samples is minimized
- Problems:
  - vanishing gradients when discriminator too powerful
  - mode collapse (generating only a subset of the target distribution)

GAN (Wasserstein)

arXiv:1701.07875

- Another possible metric: Wasserstein distance (Earth Mover's distance)

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

- γ – «optimal transport plan»
- This should solve the mode collapse and vanishing gradients problems
- Solution may not be optimal due to biased gradients (see arXiv: 1705.10743)

GAN (Cramér / energy distance)

- 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:

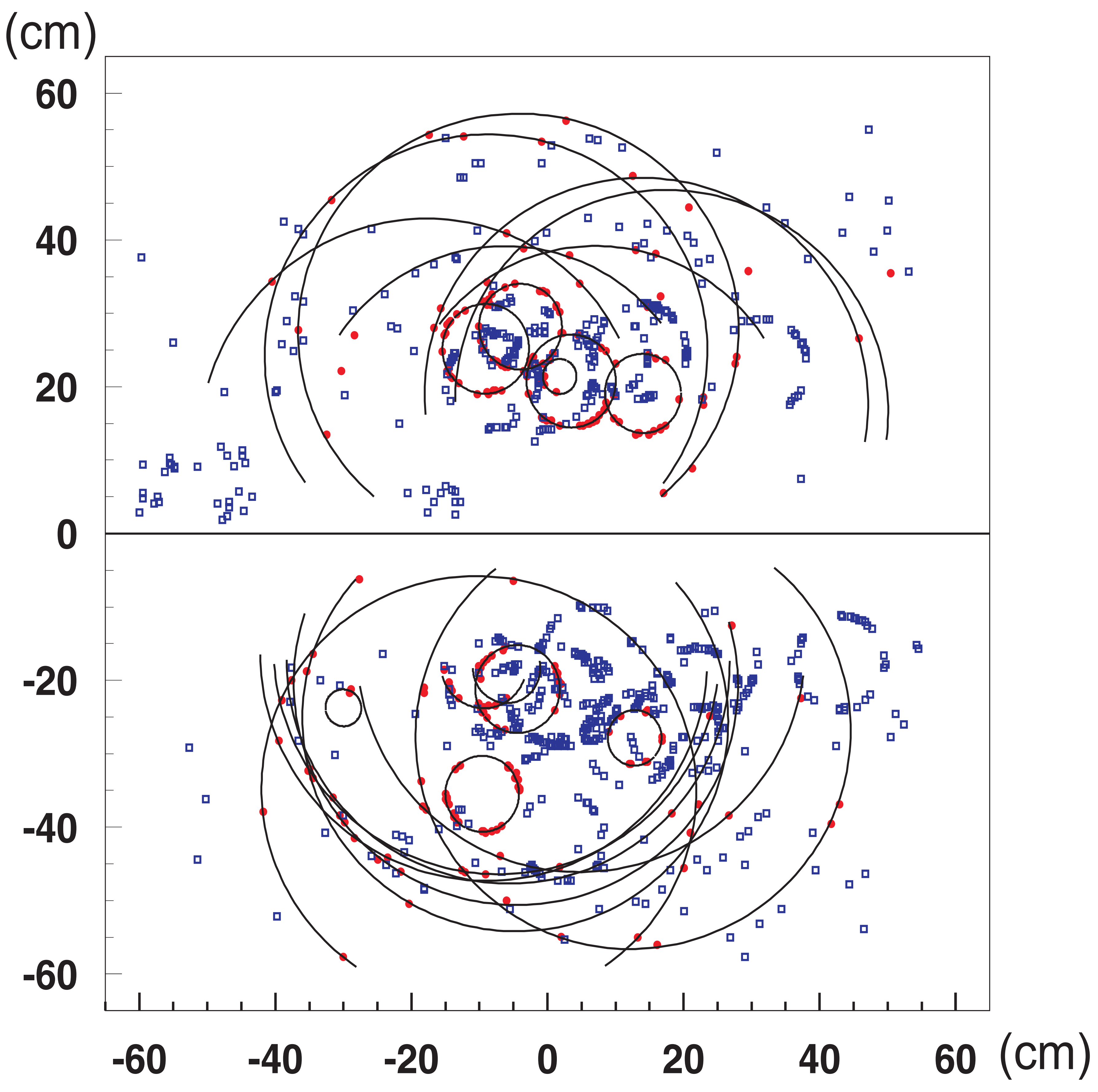
$$\mathcal{E}(X,Y) := 2\,\mathbb{E}\,\|X - Y\|_2 - \mathbb{E}\,\|X - X'\|_2 - \mathbb{E}\,\|Y - Y'\|_2$$

$X, X' \sim P$ and $Y, Y' \sim Q$

- A GAN using this metric preserves all the nice properties of Wasserstein GAN, while solving the biased gradients problem

arXiv:1705.10743

Information from RICH detectors



- PID with RICH is done with the maximum likelihood method
  - $\mathcal{L}(t_1, \ldots, t_N)$ – likelihood to observe a given picture, as a function of **all track PIDs**
    t_i – hypothesized particle type for track i
  - A hypothesis $(\underline{t}_1, \ldots, \underline{t}_N)$ maximizing $\mathcal{L}$ is searched for
- For each track i, for each x ∈ {K, μ, e, p, below threshold}, quantities **DLLx** are then calculated as:

$$\log\mathcal{L}(t_k=\underline{t}_k,\ k\neq i;\ t_i=\mathrm{x}) - \log\mathcal{L}(t_k=\underline{t}_k,\ k\neq i;\ t_i=\pi)$$

RICH simulation

- Accurate RICH simulation involves:
  - Tracing the particles through the radiators and delta-electron generation
    - Delta-electrons contribute to Cherenkov light emissions
  - Cherenkov light generation
  - Photon propagation, reflection, refraction and scattering
  - Hybrid Photon Detector (photo-cathode + silicon pixel) simulation
- All this takes time & resources
- Given the growing demand on the number of simulated events, accurate simulation becomes unfeasible