Particle Flow for Dual Read-Out Calorimeter

Adelina D'Onofrio[1], Michela Biglietti[1], Biagio Di Micco[2], Roberto Di Nardo[2], Ada Farilla[1] Iacopo Vivarelli[3] , Sofia Vallecorsa[4] , Patrizia Azzi[5]

[1]INFN - Roma Tre

[2]Roma Tre University

[3]University of Sussex

[4]CERN

[5]INFN - Padova

FCC Italy-France Workshop - Lion

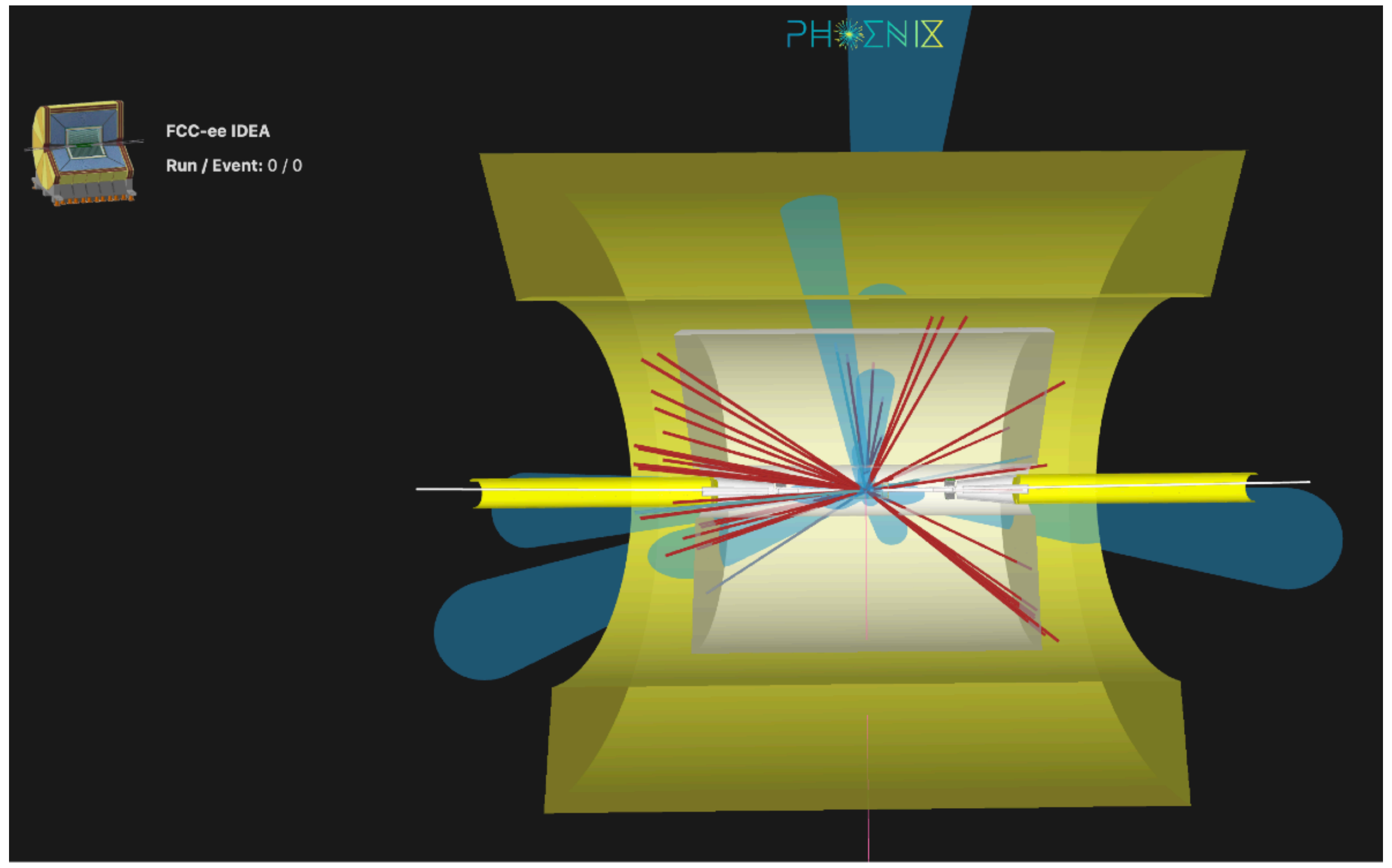
22nd November 2022

AIDA innova

INFN Istituto Nazionale di Fisica Nucleare SEZIONE DI ROMA TRE

Outline

- Highlights about dual read-out calorimeter for the IDEA detector
  - Single, dual-readout calorimeter for EM and **HAD** calorimetry
    - But option to have a **crystal, dual-readout** EM section
  - Detector layout
  - Energy measurement
- Particle Flow algorithm for dual-read-out calorimeters
  - Software implementation
  - Neural Network approach used for particle identification and jet energy reconstruction
  - Preliminary results



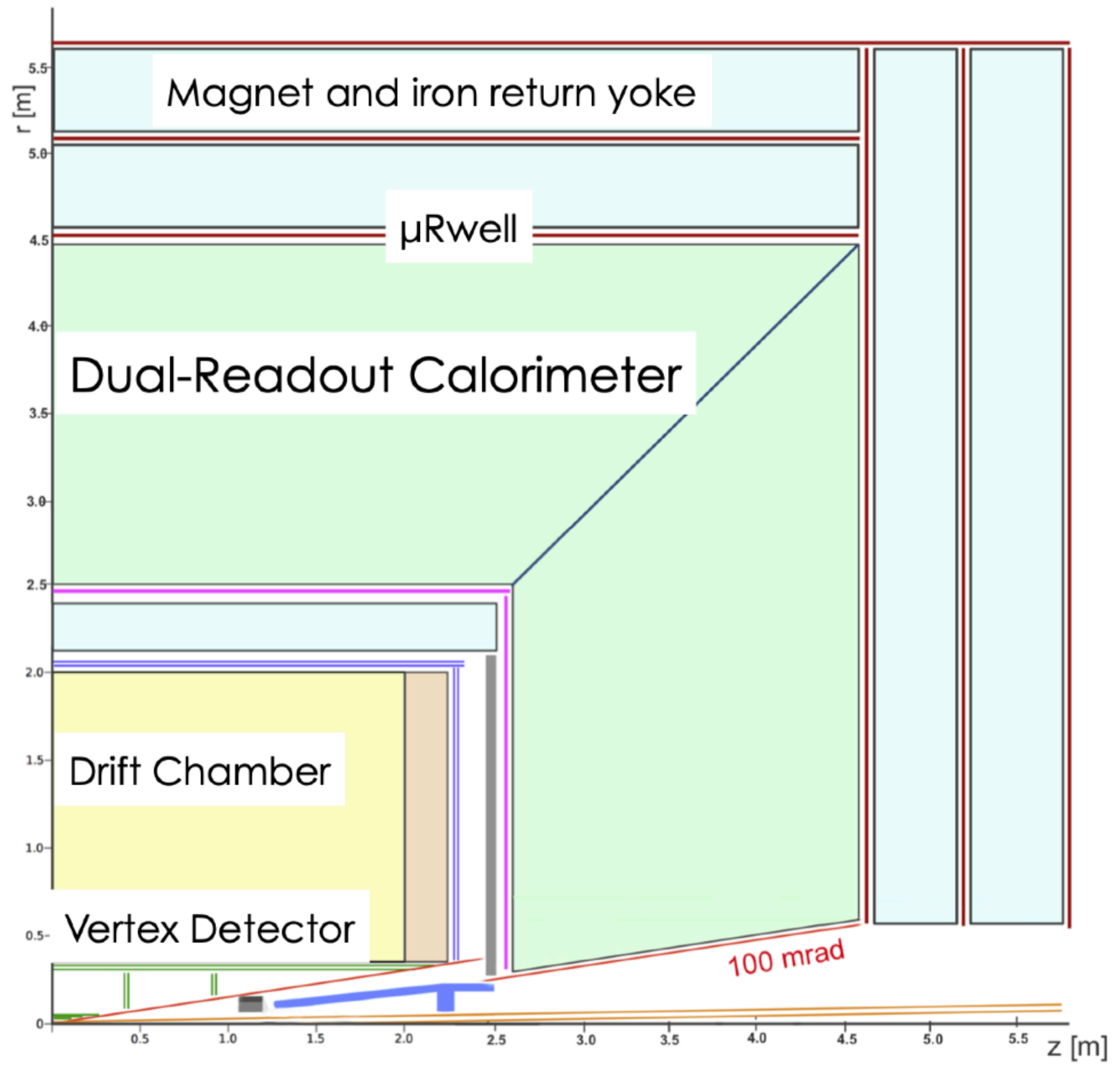
Physics Process	Measured Quantity	Critical Detector	Required Performance
$ZH \to \ell^+\ell^- X$	Higgs mass, cross section	Tracker	$\Delta(1/p_T) \sim 2 \times 10^{-5}$
$H \to \mu^+\mu^-$	$\mathrm{BR}(H \to \mu^+\mu^-)$		$\oplus 1 \times 10^{-3}/(p_T \sin\theta)$
$H \to b\bar{b},\ c\bar{c},\ gg$	$\mathrm{BR}(H \to b\bar{b},\ c\bar{c},\ gg)$	Vertex	$\sigma_{r\phi} \sim 5 \oplus 10/(p \sin^{3/2}\theta)\ \mu\mathrm{m}$
$H \to q\bar{q},\ VV$	$\mathrm{BR}(H \to q\bar{q},\ VV)$	ECAL, HCAL	$\boxed{\sigma_E^{\mathrm{jet}}/E \sim 3-4\%}$
$H \to \gamma\gamma$	$\mathrm{BR}(H \to \gamma\gamma)$	ECAL	$\sigma_E \sim 16\%/\sqrt{E} \oplus 1\%$ (GeV)

Desired jet energy resolution

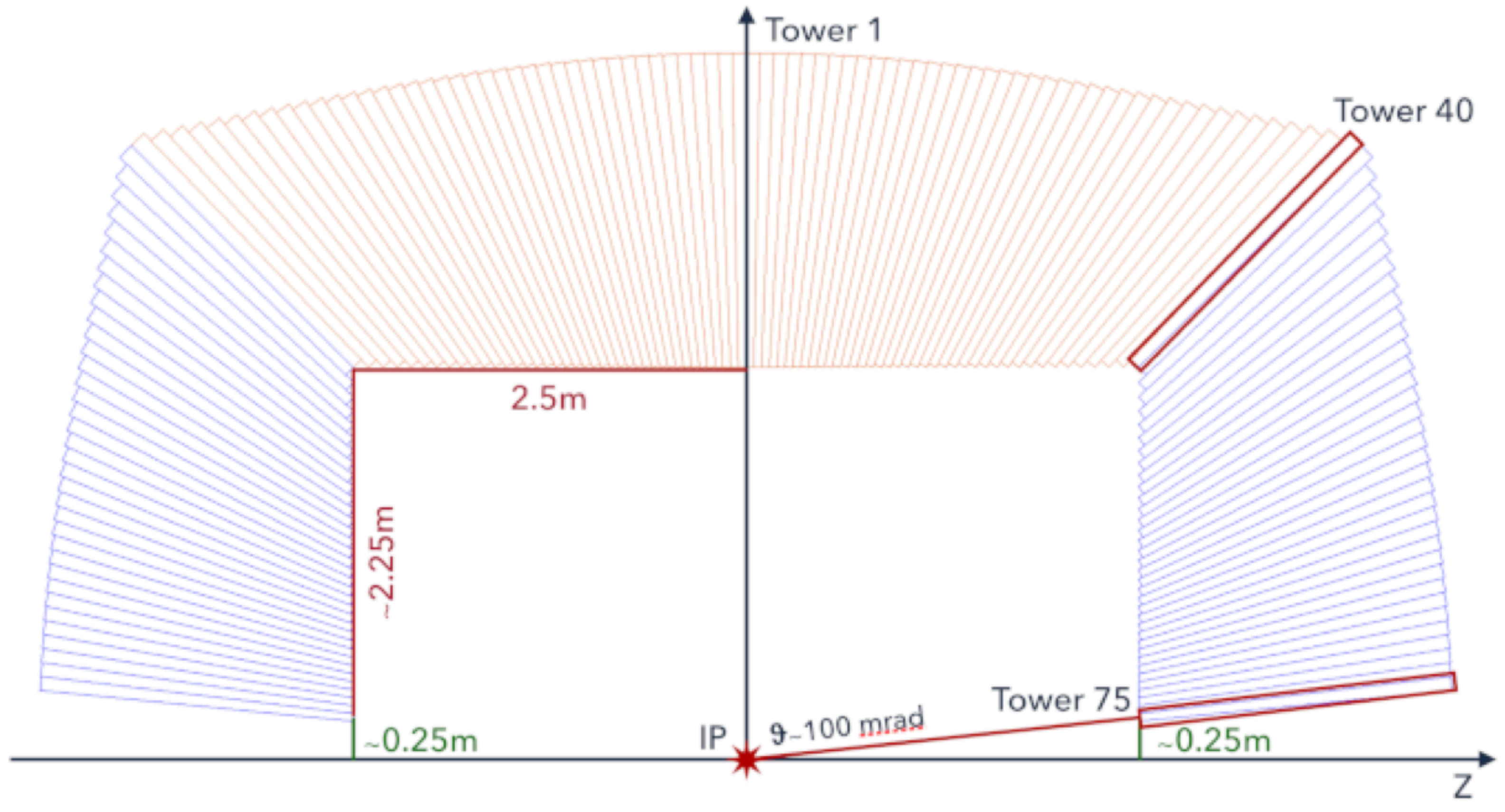
ParticleFlow Project - Status Report

Dual Read-out Calorimeter for the IDEA detector

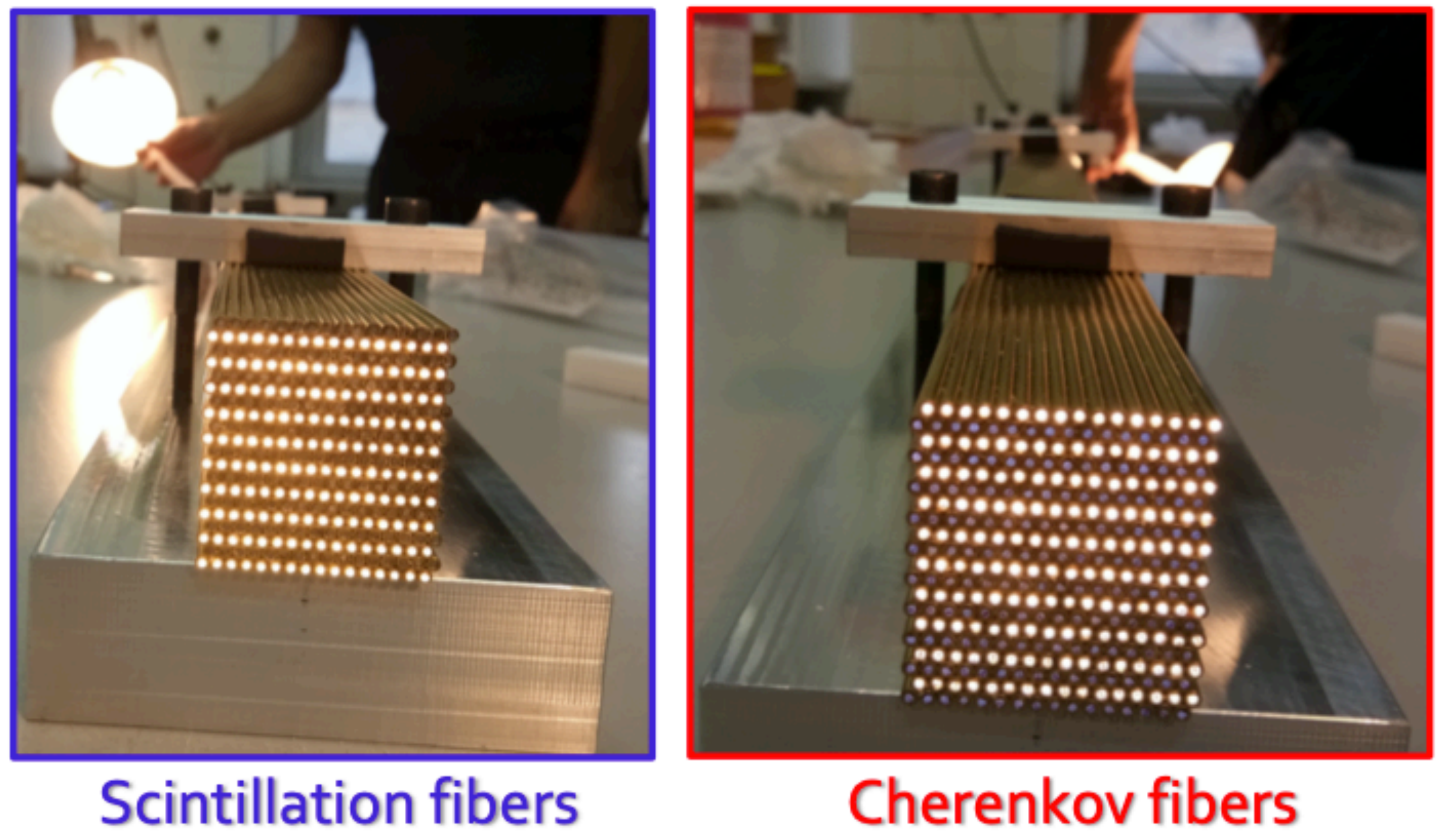
- Dual readout calorimeters aim at improving the energy resolution of hadronic calorimeters
  - Generally driven by the fluctuations between the electromagnetic and the hadronic component of showers
- Measure the hadronic component and the electromagnetic component (dual readout) of the showers separately, to derive proper correction factors to be applied to each component to reconstruct the energy of the impinging hadrons
- Exploit a passive/material - fibre layout where two type of fibres, one sensitive to the usual scintillation process, a second type of fibre producing Cherenkov light when ultra-relativistic particles cross with a speed higher than the speed of light in that fibre



A. IDEA detector transversal view



B. Sketch of a single slice of the IDEA calorimeter



C. Current R&Ds

The Particle Flow Project

- In general, particle flow algorithms applied to dual read-out calorimeters provide limited performance on the energy resolution of the electromagnetic component of the jets
  - Scenarios with EM calorimeter added in front of the IDEA dual read-out calorimeter —> Cons: calorimeter non-compensation
  - Ongoing R&D for crystal dual read-out calorimeters to fix the compensation issue
- The aim of the project is to build a Neural Network based algorithm that, from a given collection of energy deposits in the calorimeter, is able to completely reconstruct a jet in the detector
- Our goal: maximise the energy resolution of the dual read-out calorimeter exploiting NNs and taking as input all the available kinematic variables
  - NN based particle identification: use as basis a particle flow approach, which aims at identifying each single particle inside a jet
  - NN based jet reconstruction: construct a regression algorithm for particle-jet assignment and jet energy reconstruction

Overview of the Project

1 Geant4 simulation of a jet

2 Extract particle info at the input layer: position, momentum, particle type

Extract calorimeter info: fiber position, fiber type, collected light by the fiber

3 Clustering ?

4 NN based particle identification: charged: $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, $K^{\pm}$, p, neutral: γ, π^{0}, $K_{L,S}$, n, Λ

NN based jet reconstruction:
starting from particle lists and their momenta, build a jet

Work in progress

Software Implementation

Input from detector simulation (EDM4HEP) format

↓

Reading using KEY4HEP *code*

↓

Dumping algorithm, input variables for NN training

↓

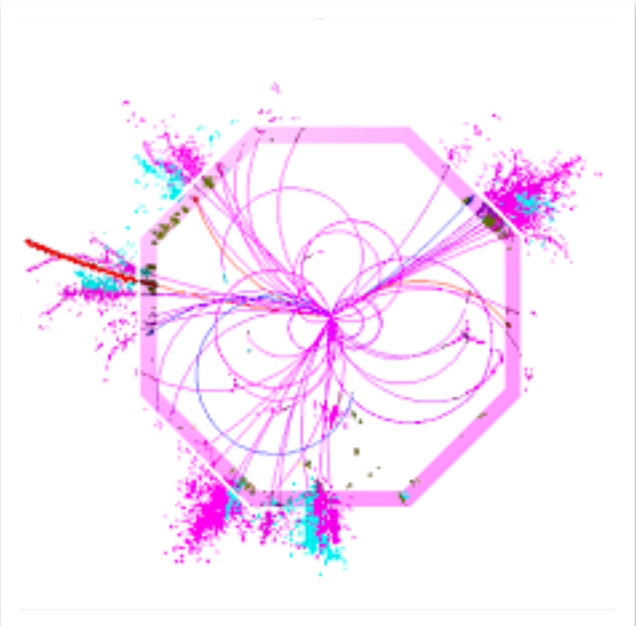
NN training using *Tensorflow* on CPU/GPU

- Geant4-based simulations of the IDEA detector for *e, π, K* with energy and angular uniform distribution (thanks for the inputs!)
- An algorithm was developed that reads KEY4HEP format and produces an output to perform a Neural Network training
  - To be done: interface between KEY4HEP and Pandora
- Target: build a NN able to reconstruct the energy and the position of the impinging particles and identify them
  - Regression and classification (to discriminate *e, π, K*) algorithms implemented in a single NN
- State of the art:
  - NN studies performed on an input sample containing electrons with uniform distributions (in energy and position), training performed for energy regression

Software Stack

The project will be developed in KEY4HEP and Pandora —> interface Pandora with KEY4HEP

- key4HEP: general software framework developed for many experiments
  https://github.com/key4hep
  - GEANT4 implementation in KEY4HEP already started



- Pandora Particle Flow Algorithm *https://github.com/PandoraPFA*
  - Collection of pattern recognition algorithm, the idea is to insert our algorithm inside Pandora and compare its performance with already existing algorithms. Algorithm already present: several clustering algorithms, non NN based Particle Flow algorithms, NN based reconstruction algorithm for LAr TPC for the DUNE experiment
  - Training outside Pandora, use interfaces to produce input to the training data format and inference

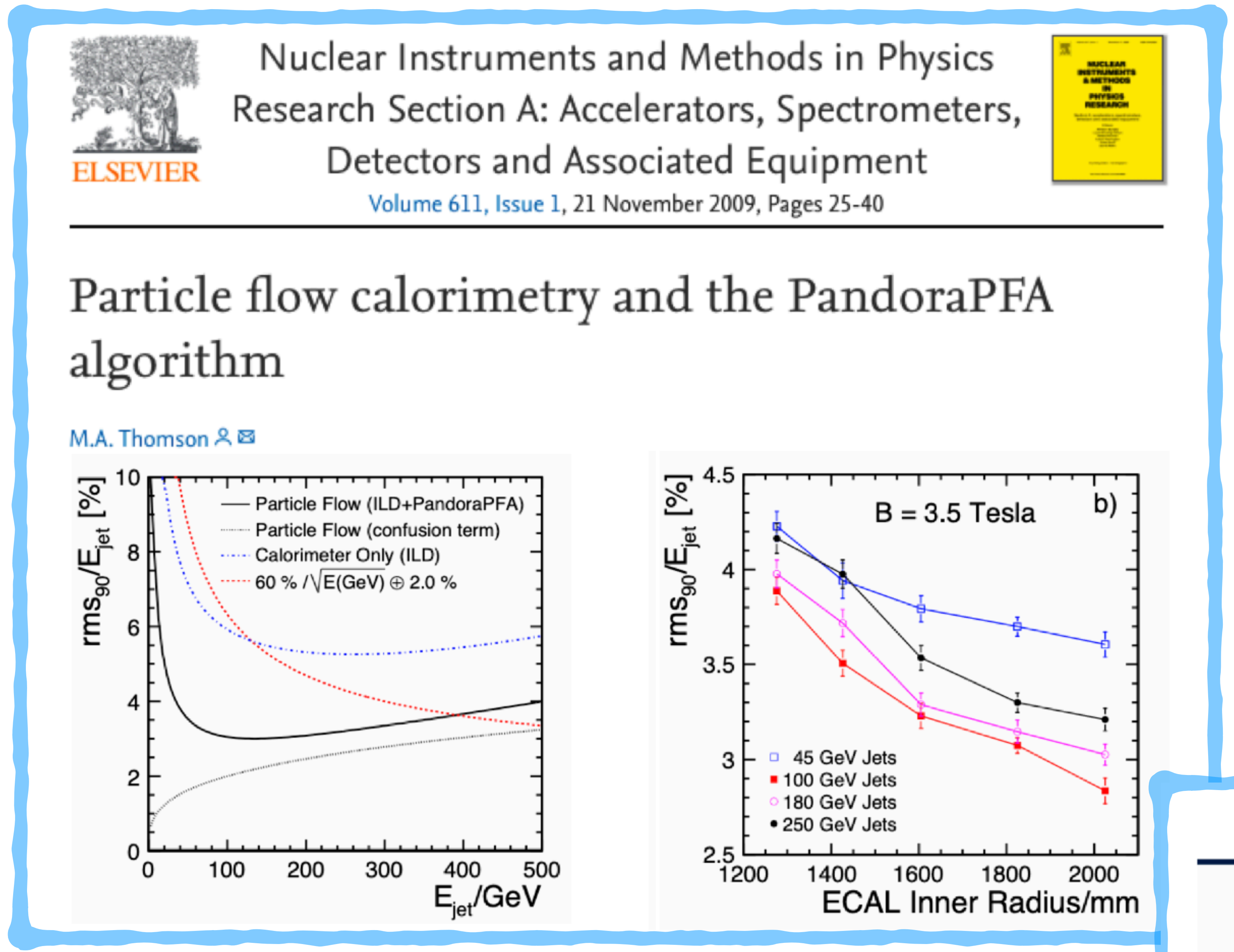
Pandora Interface to NN training

Training outside Pandora (using GPU for example)

↓

Inference inside Pandora

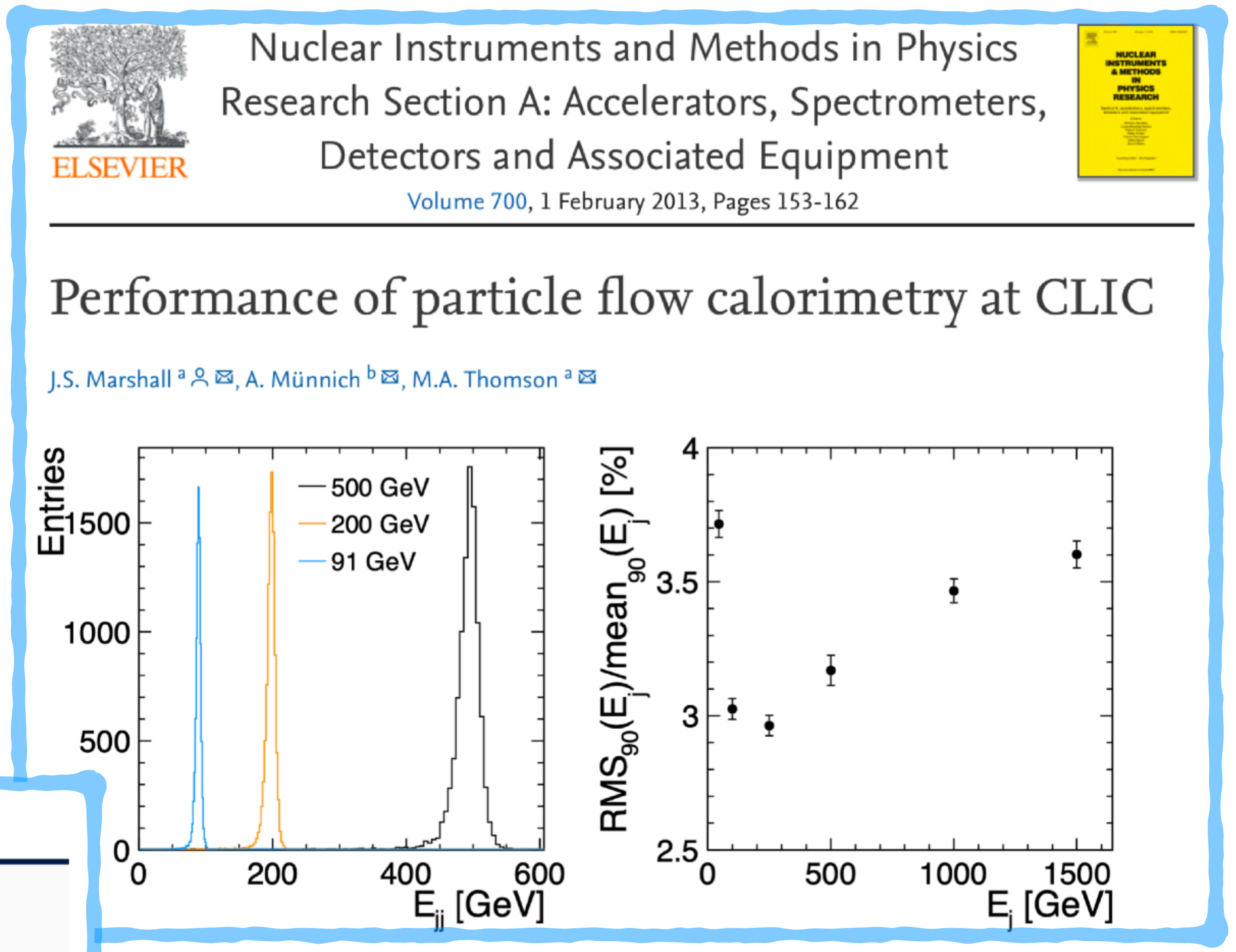
Pandora: examples at ILC/CLIC


Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment
Volume 611, Issue 1, 21 November 2009, Pages 25-40

Particle flow calorimetry and the PandoraPFA algorithm

M.A. Thomson

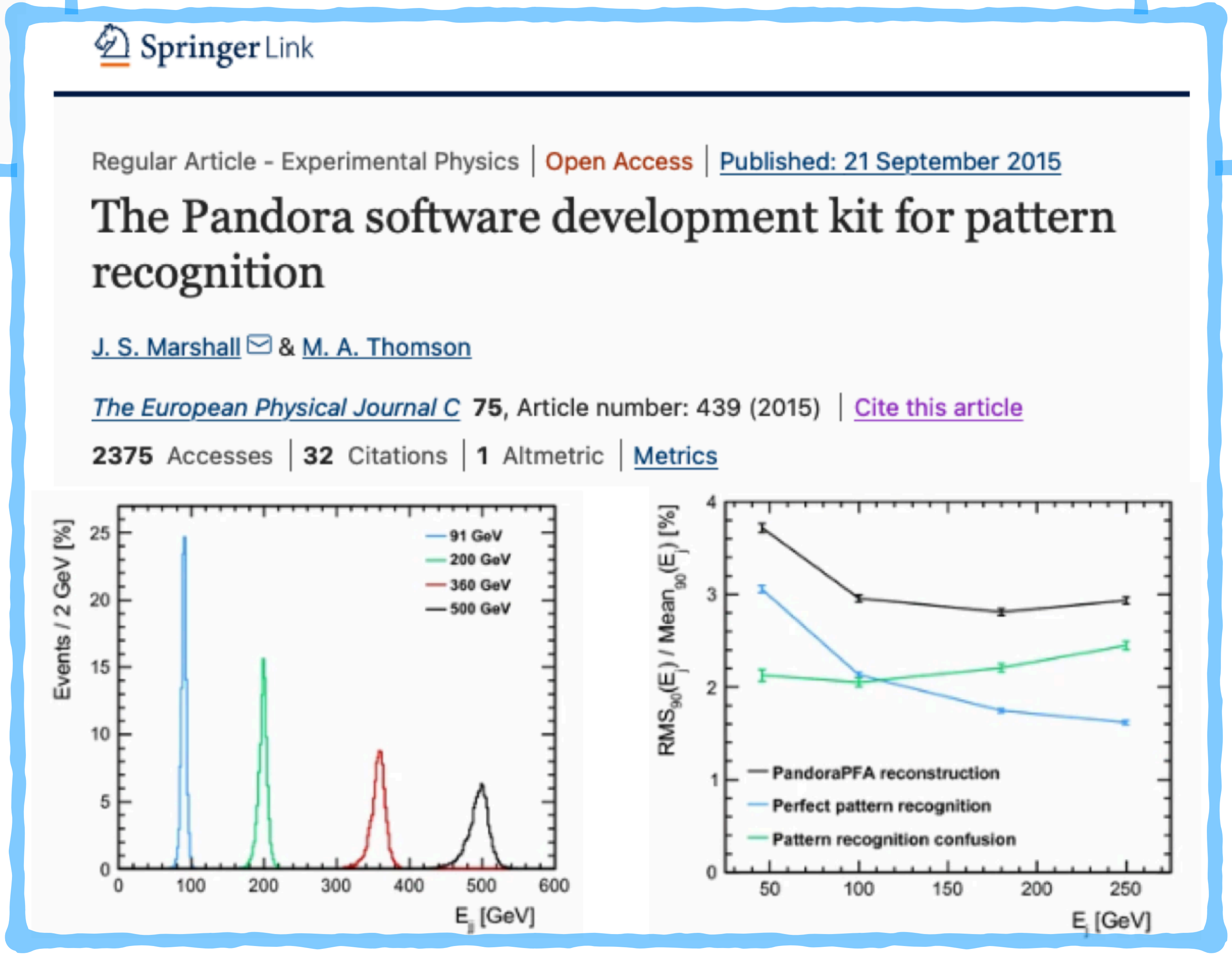
Nucl. Instrum. Meth.A611, 25 (2009)


Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment
Volume 700, 1 February 2013, Pages 153-162

Performance of particle flow calorimetry at CLIC

J.S. Marshall [a], A. Münnich [b], M.A. Thomson [a]

Nucl. Instrum. Meth.A700, 153 (2013)


Springer Link

Regular Article - Experimental Physics | Open Access | Published: 21 September 2015

The Pandora software development kit for pattern recognition

J. S. Marshall & M. A. Thomson

The European Physical Journal C 75, Article number: 439 (2015) | Cite this article

2375 Accesses | 32 Citations | 1 Altmetric | Metrics

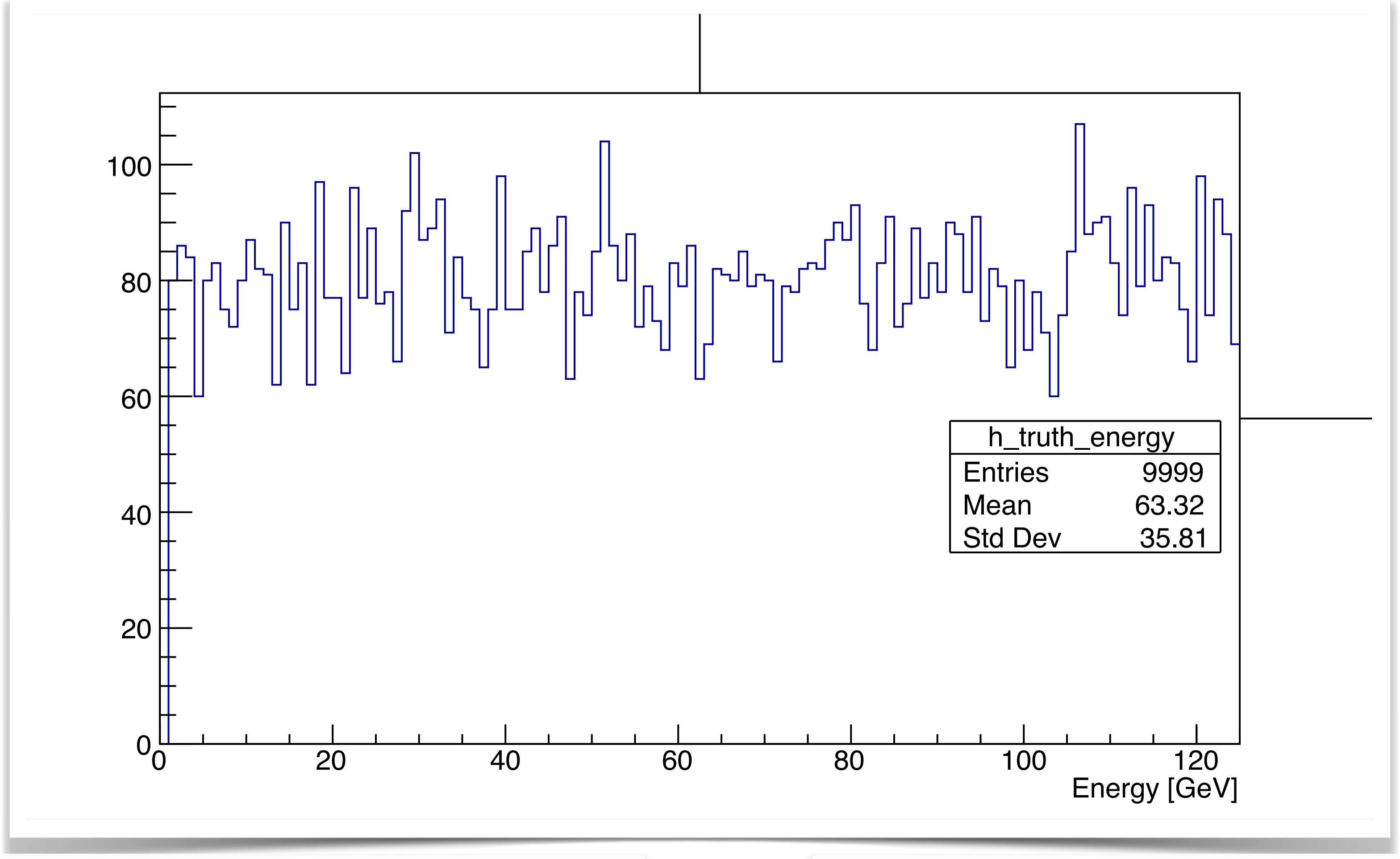
Eur. Phys. J. C 75, 439 (2015)

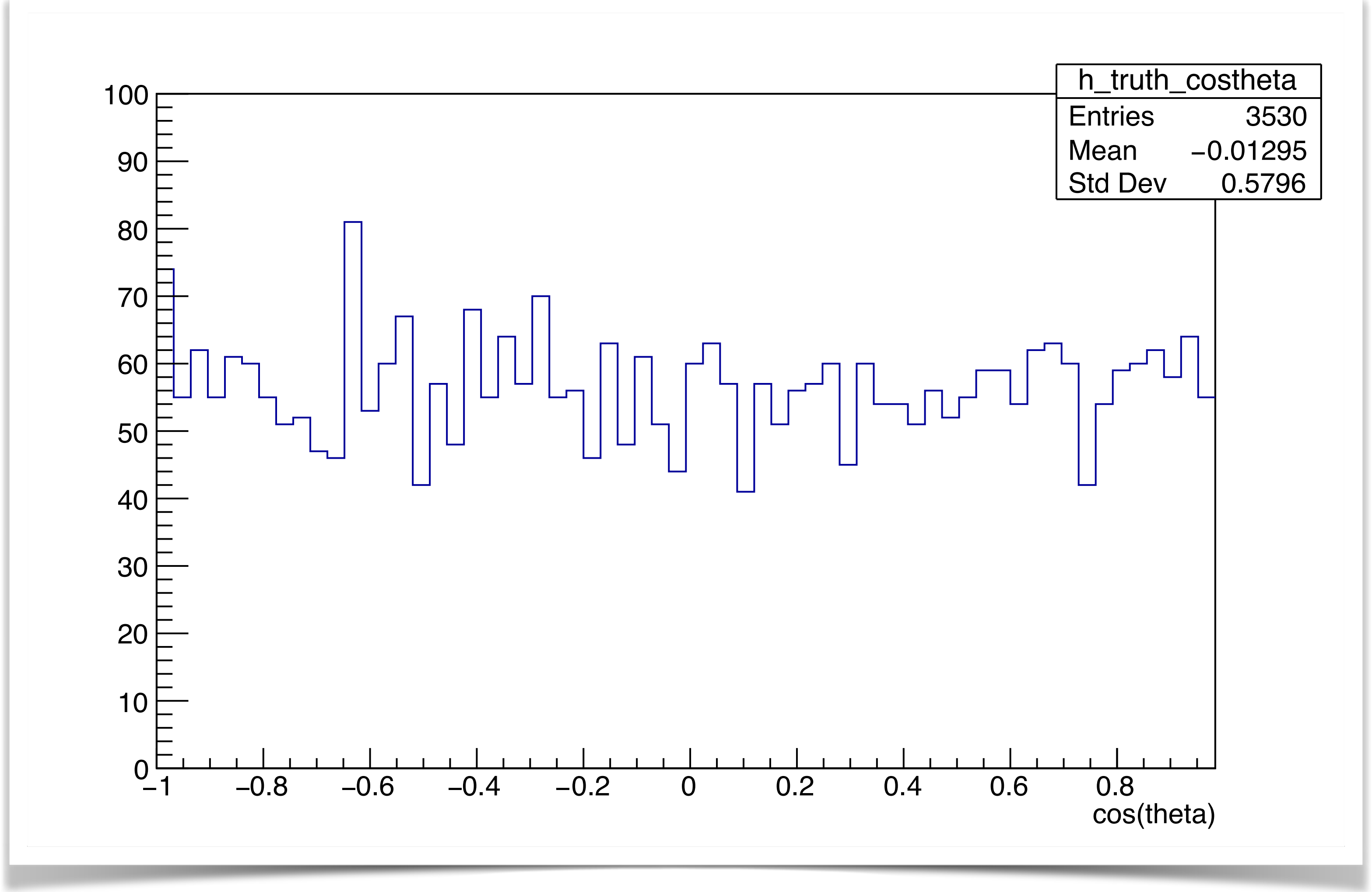
Kinematic distributions - Electrons

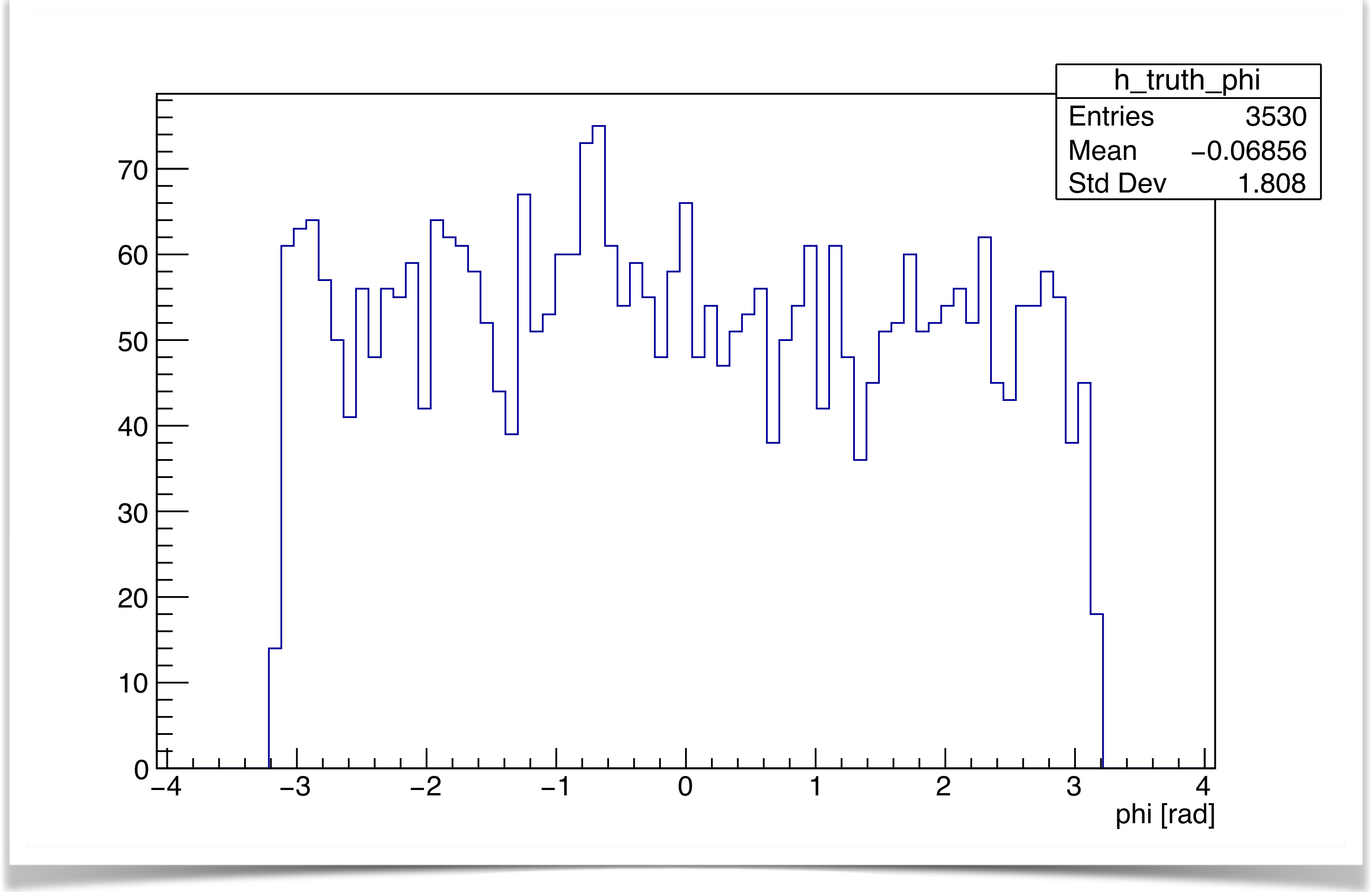
All events used (#10000)

- Position and energy collected in the scintillating (S) **and** Cerenkov (C) fibres in 10000 events **simulating** impinging electrons of uniform energy, in the range [0-125] GeV, and uniform angular distributions

Dumper Algorithm output







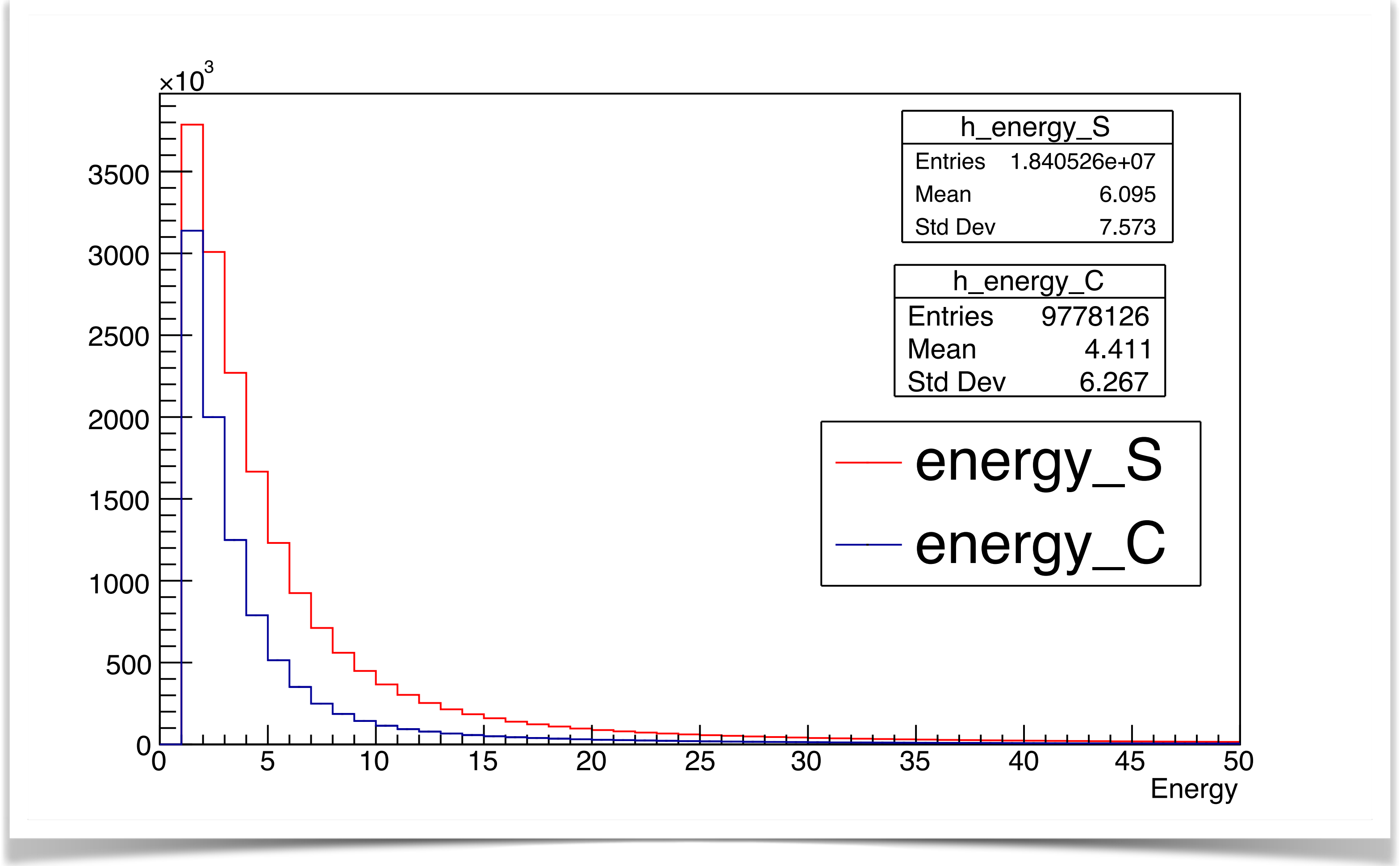
Truth level Variables

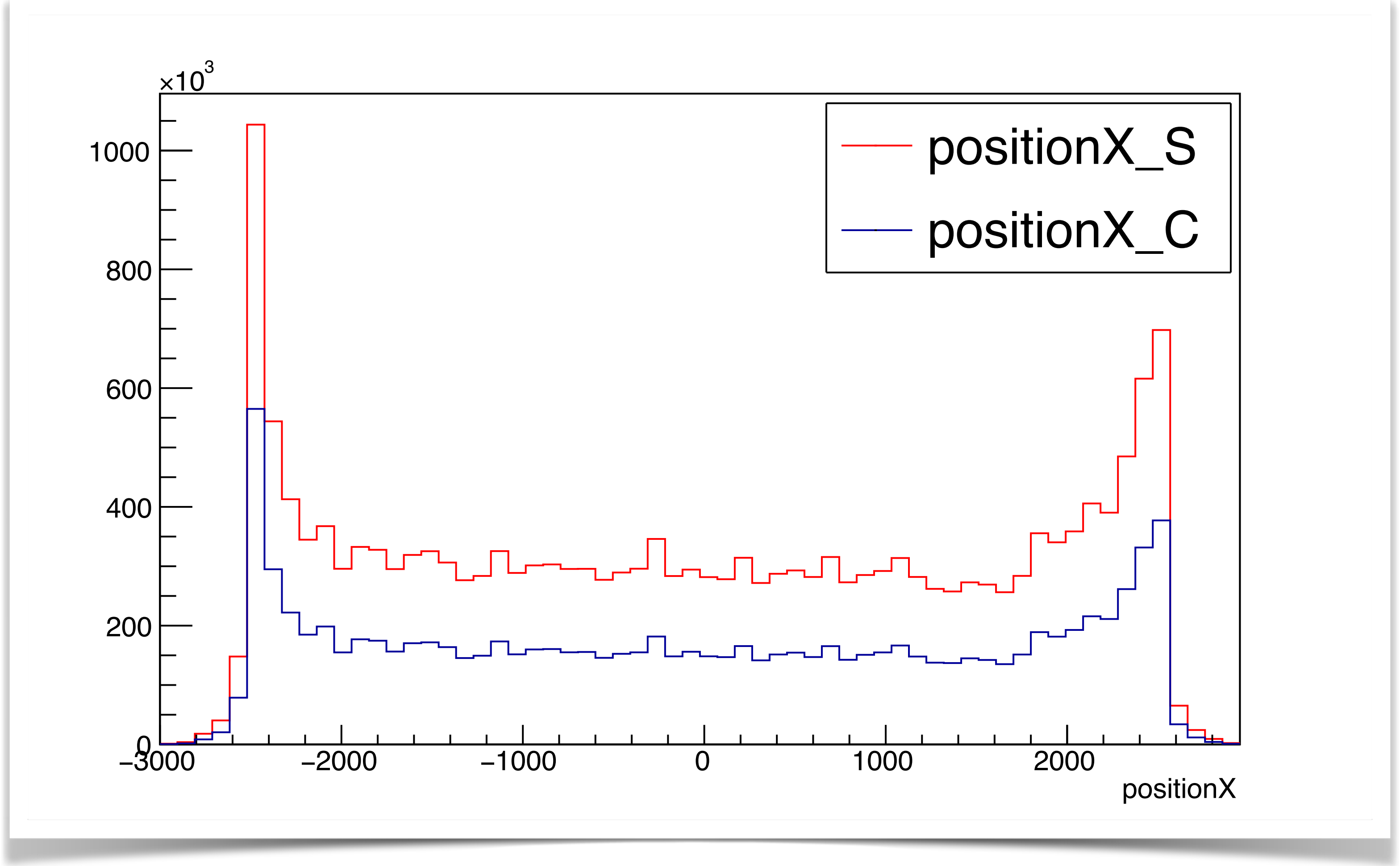
Kinematic distributions - Electrons

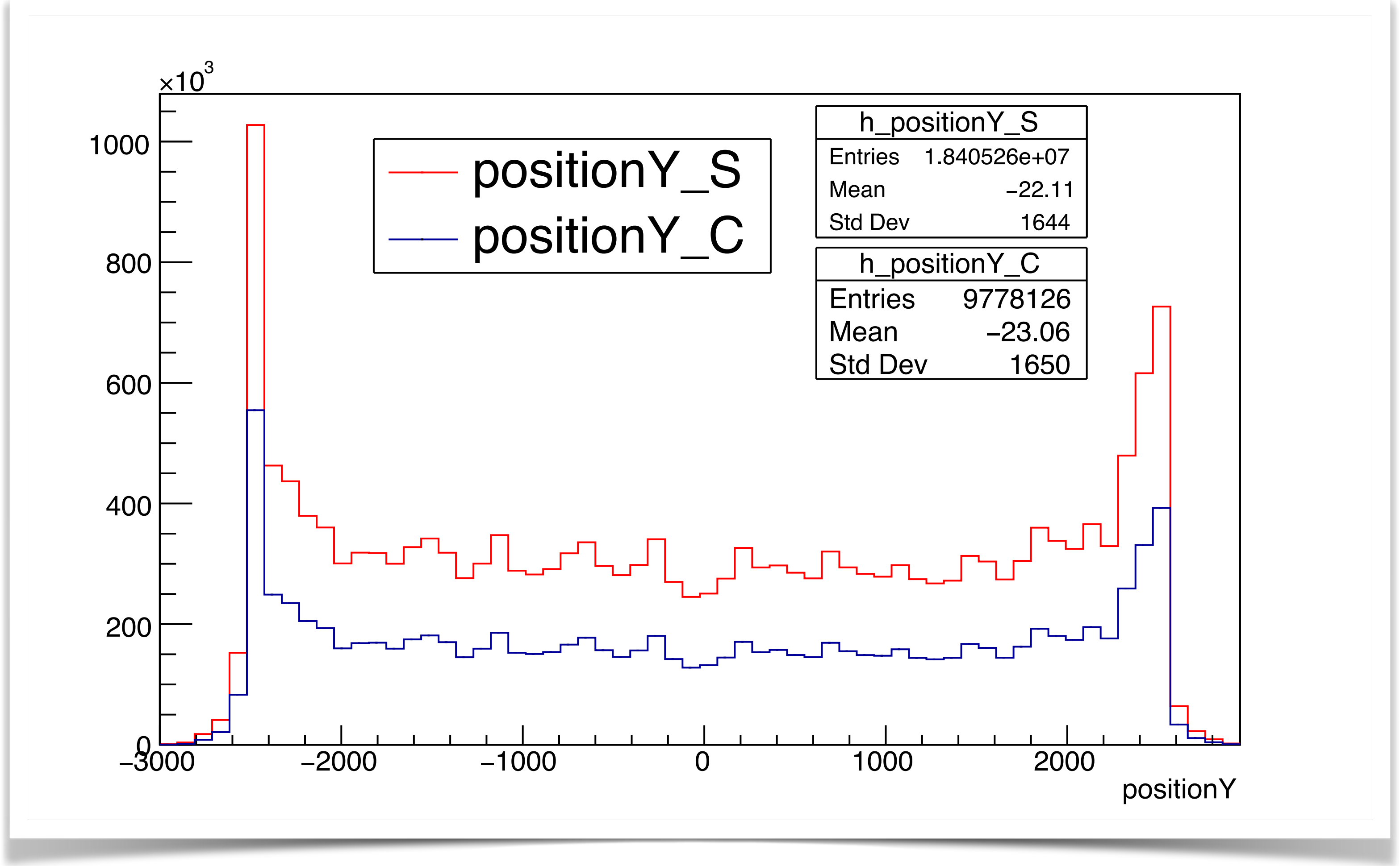
All events used (#10000)

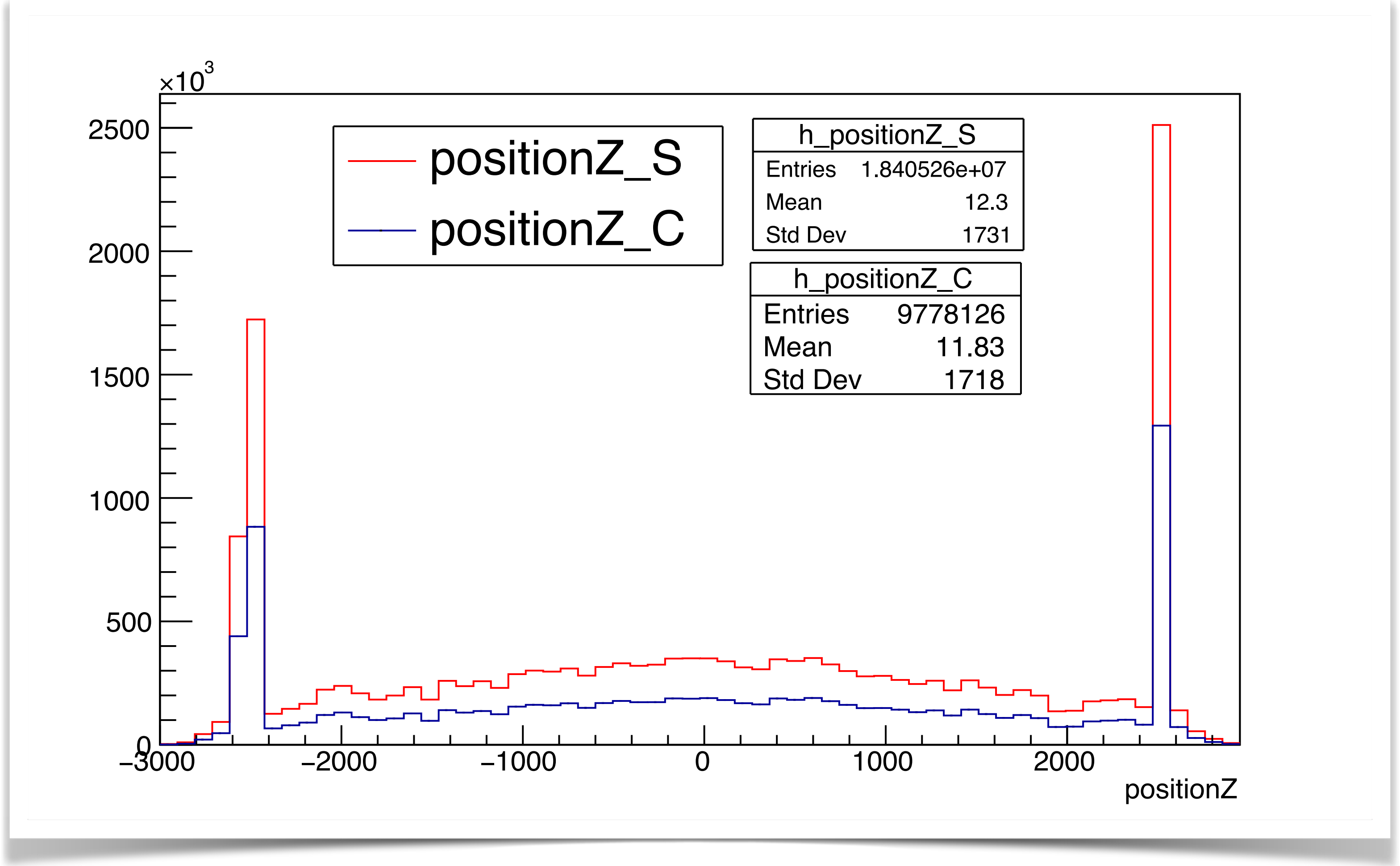
- Position and energy collected in the scintillating (S) **and** Cerenkov (C) fibres in 10000 events **simulating** impinging electrons of uniform energy, in the range [0-125] GeV, and uniform angular distributions

Dumper Algorithm output









Reco level Variables

Energy deposits - 20 GeV electrons

Dumper Algorithm output

Electron deposits

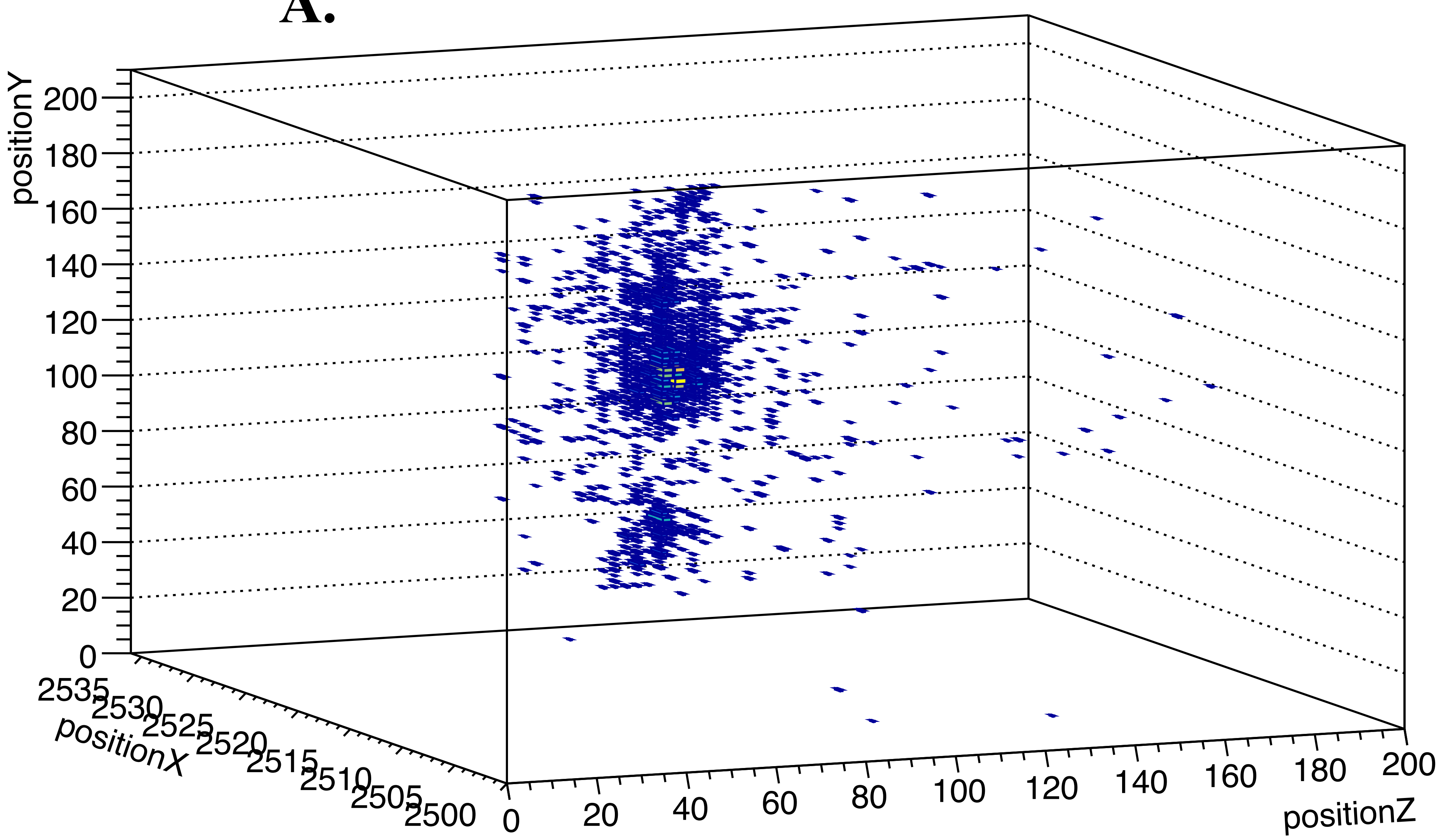
- Energy collected in the scintillating (S) **and** Cherenkov (C) fibres in 100 events **simulating** impinging electrons of 20 GeV

A. Energy deposited in the detector, projected in the *(x,y,z)* space —> combined fibres

B. Energy deposited in the scintillating fibres, polar coordinates

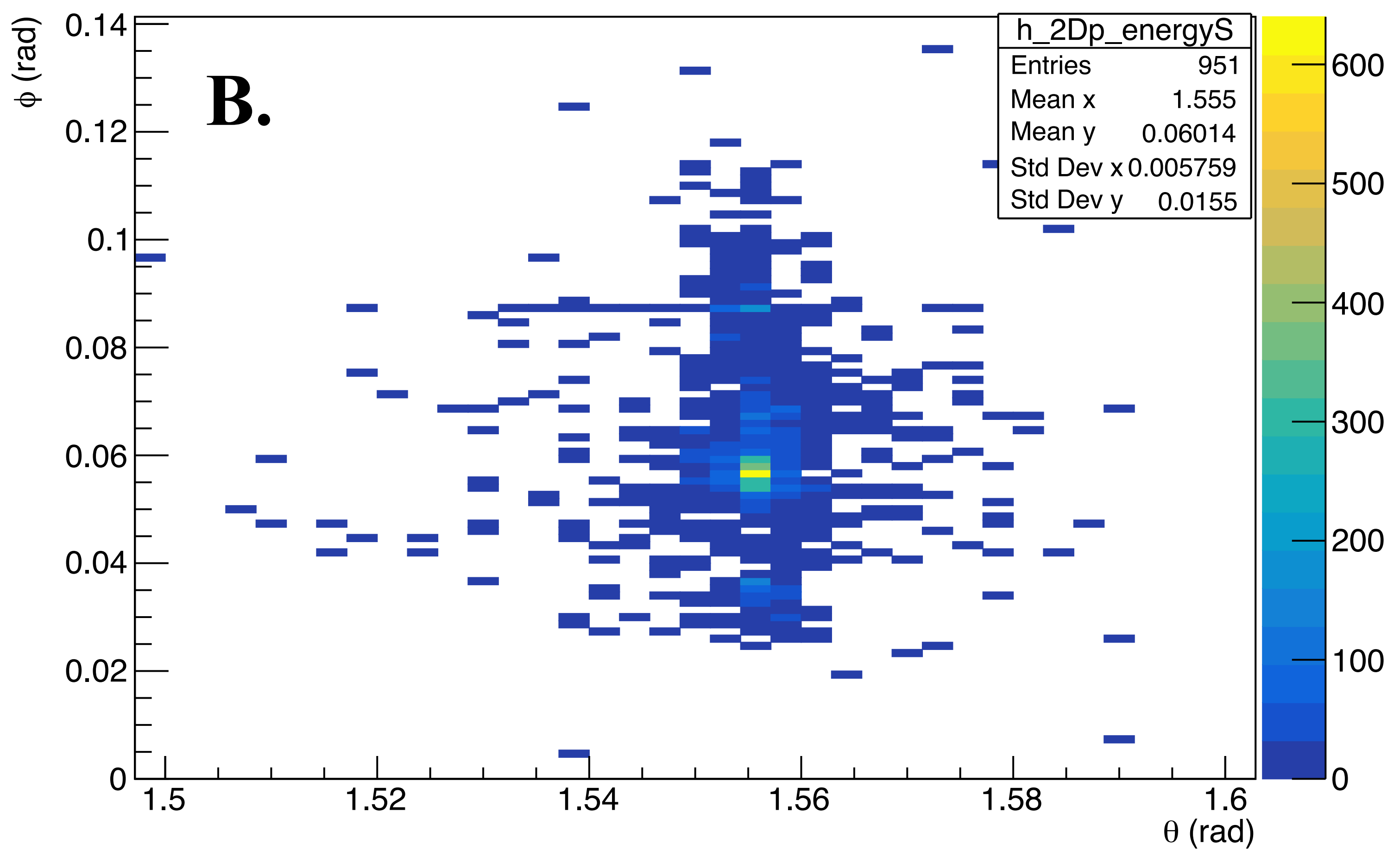
C. Energy deposited in the Cherenkov fibres, polar coordinates

Combined fibres

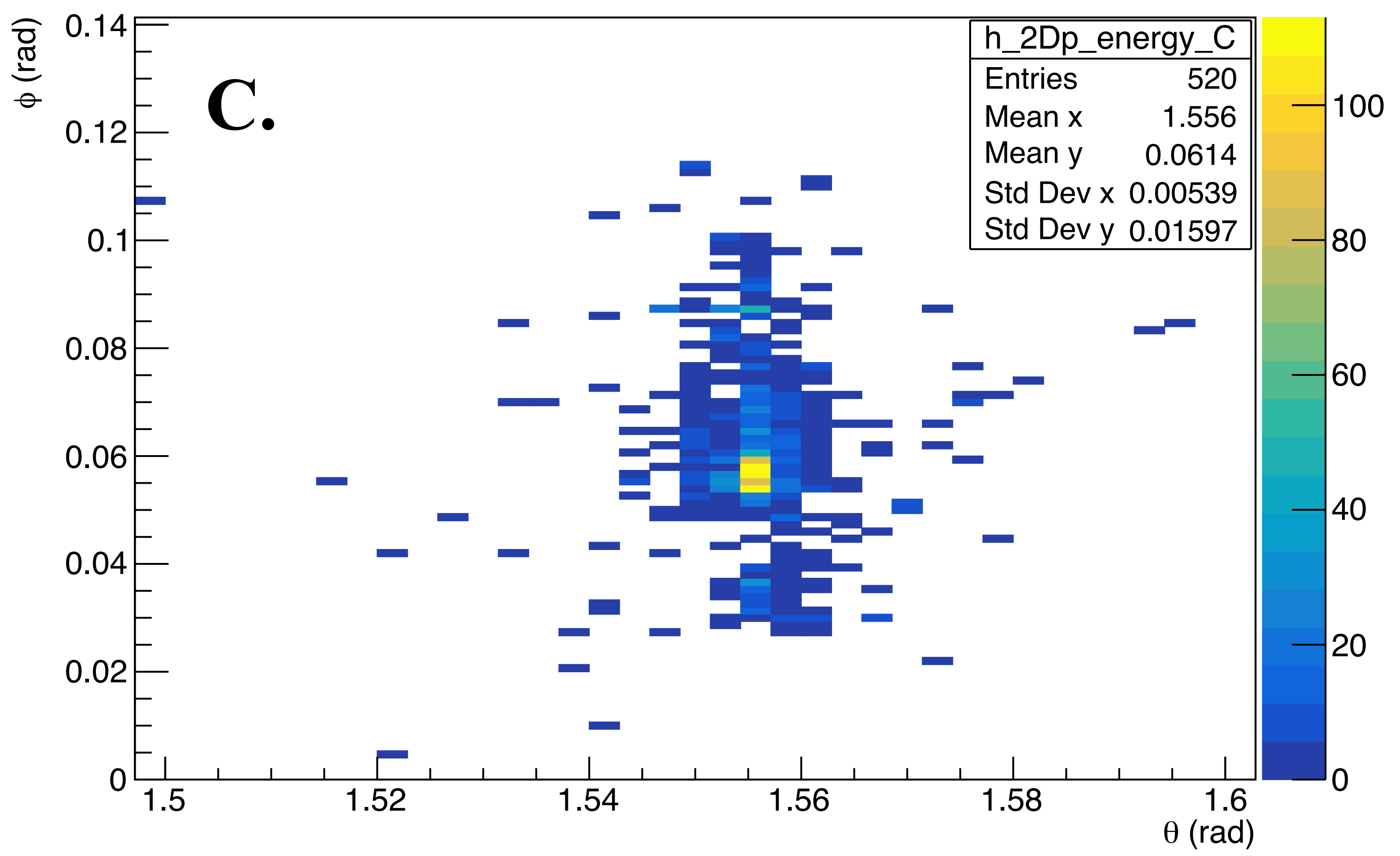


3D plot, cartesian coordinates

Scintillating fibres



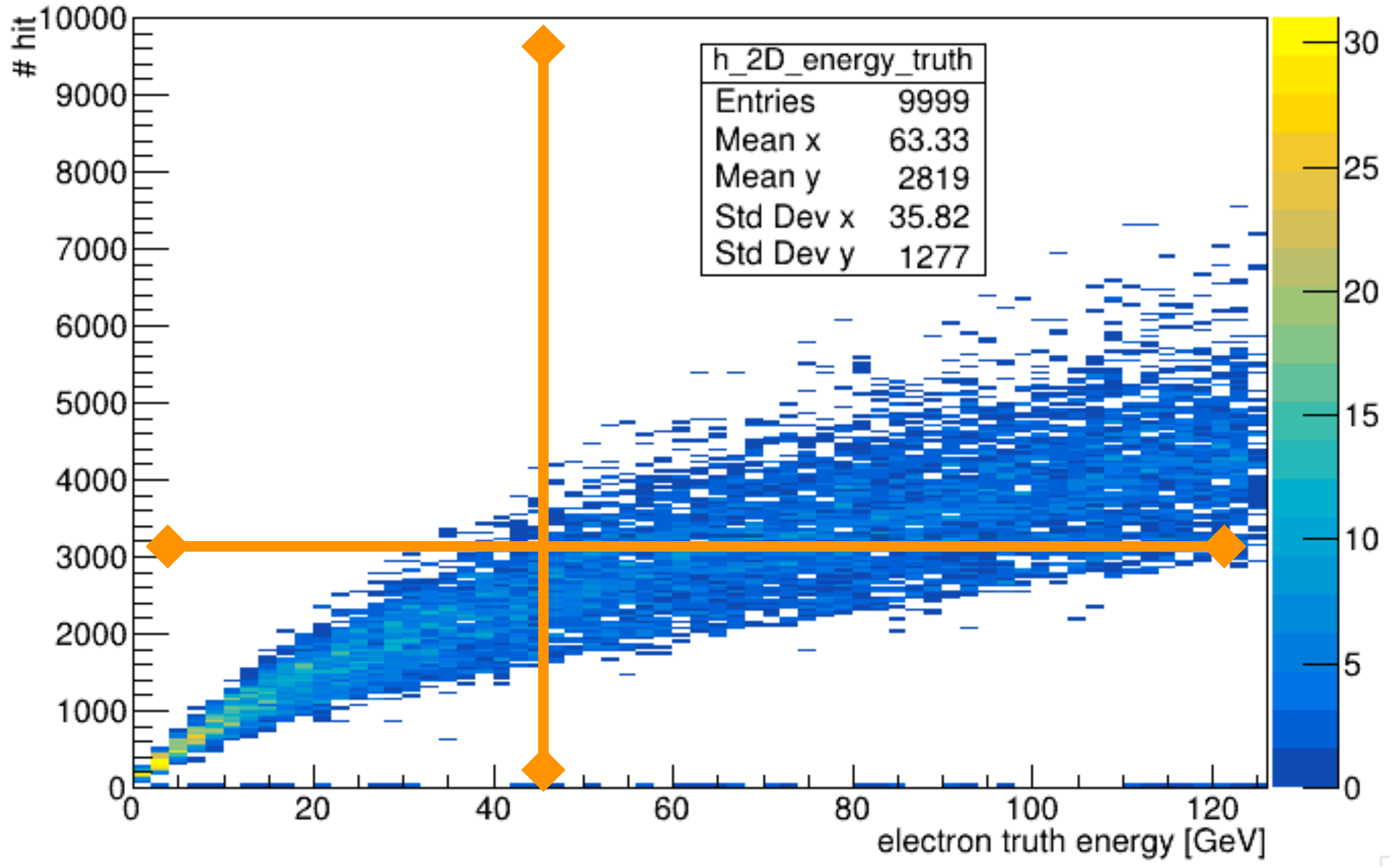
Cherenkov fibres

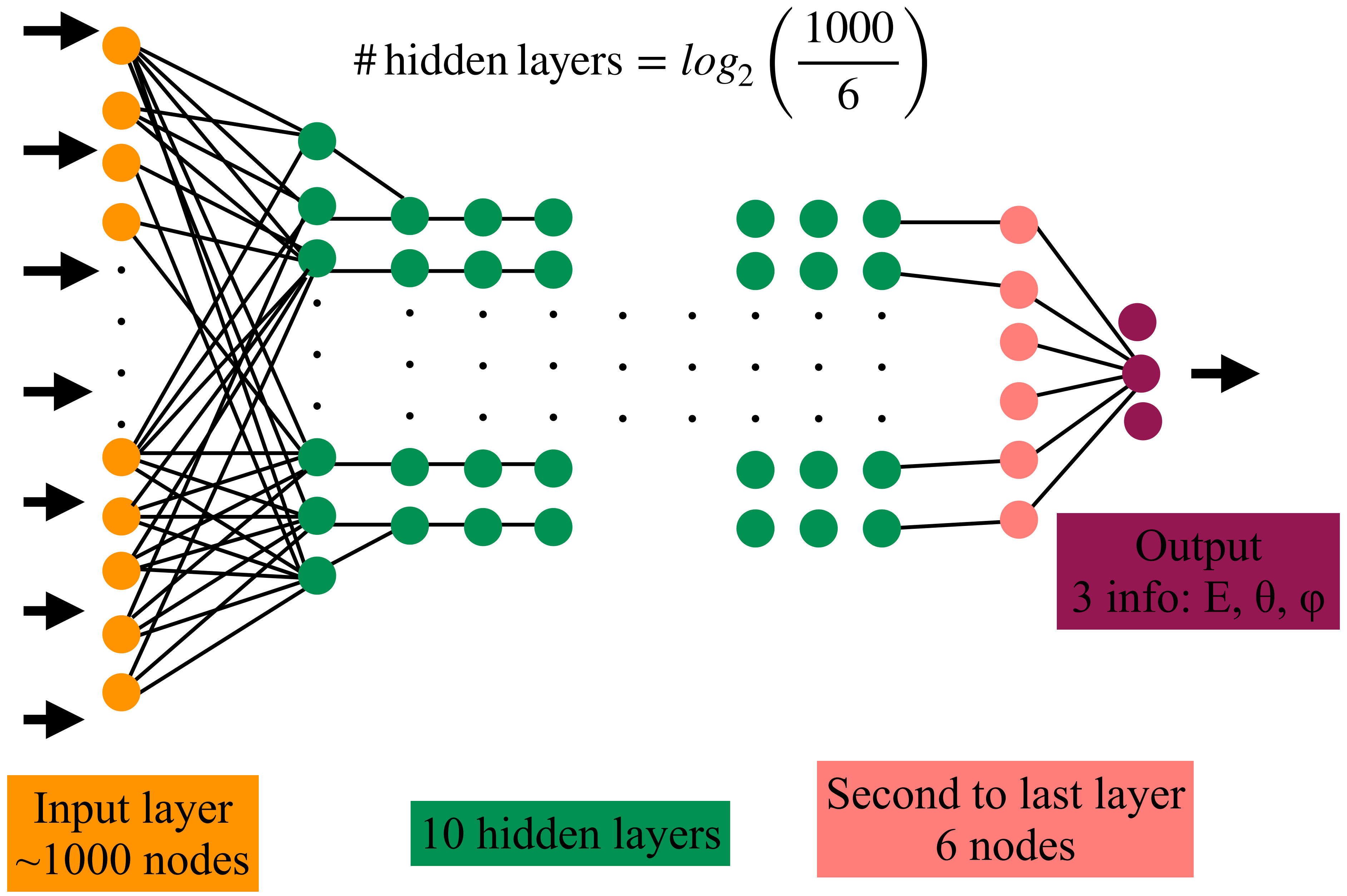


2D plots, polar coordinates

NN training using Tensorflow on GPUs

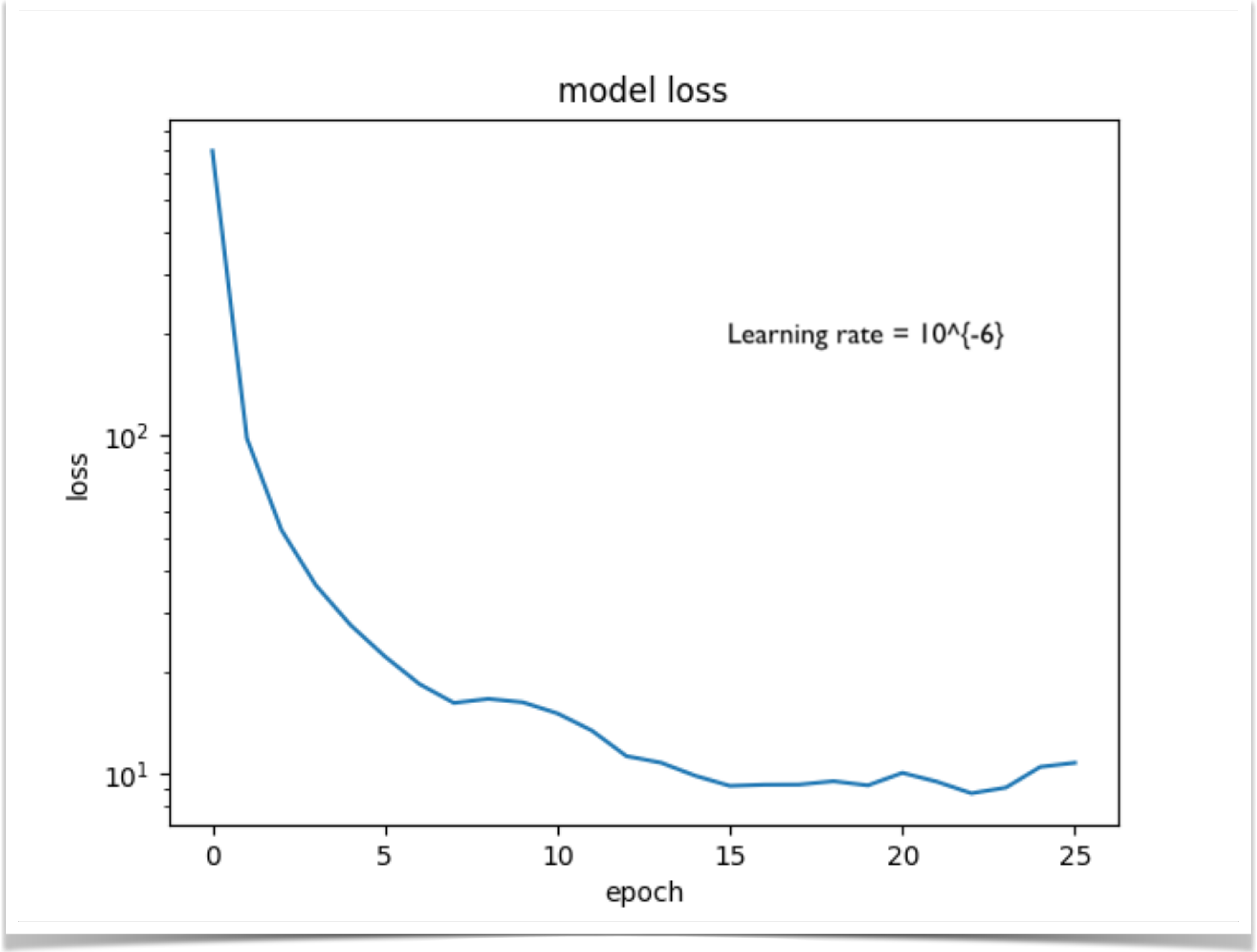
- Tensorflow, interfaced with Keras, is used to build and train a NN on GPUs
- Inputs: energy and position of each hit in the shower generated by the impinging electron and recorded in both S&C fibres—> NN input: 6 kinematic variables (*E, x, y, z, t, flag*) times hit multiplicity (~ 45000 info per event, 10000 simulated events used)
  - Maximum hit multiplicity: ~2200 per electron
  - Zero padding approach: if the number of hits in the event is less than the max hit multiplicity, set to zero the remaining positions in the array
- # initial nodes = # input info
  - Exploit the average hit multiplicity * 6 kinematic variables as #initial nodes to reduce the complexity of the problem
- # hidden layers = 10
- At each layer the number of nodes halves
- Need to reduce the number of inputs due to GPU memory issues
- Speed of the algorithm:
  ~10 minutes on GPU
  ~2 hours on CPU



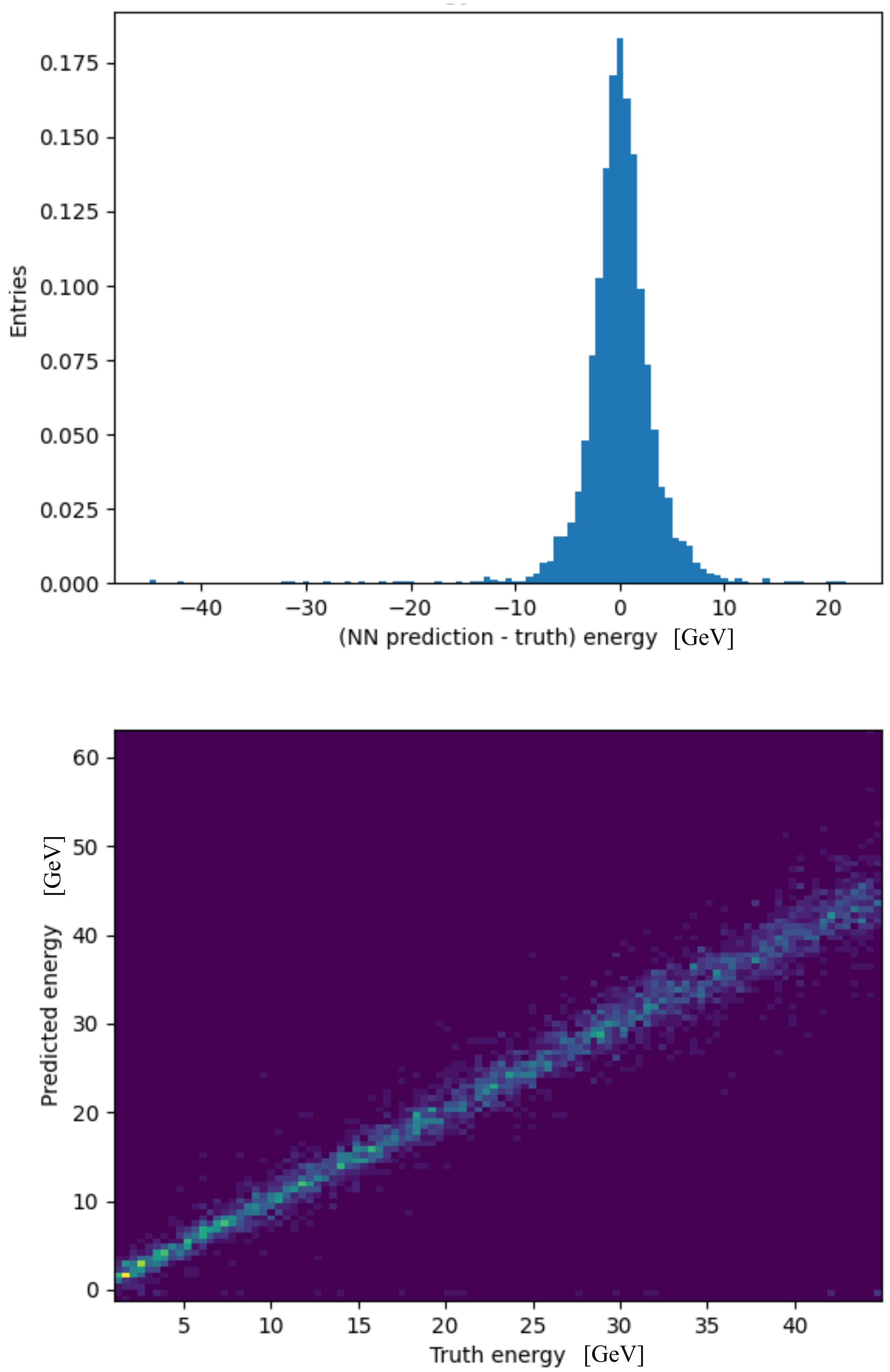


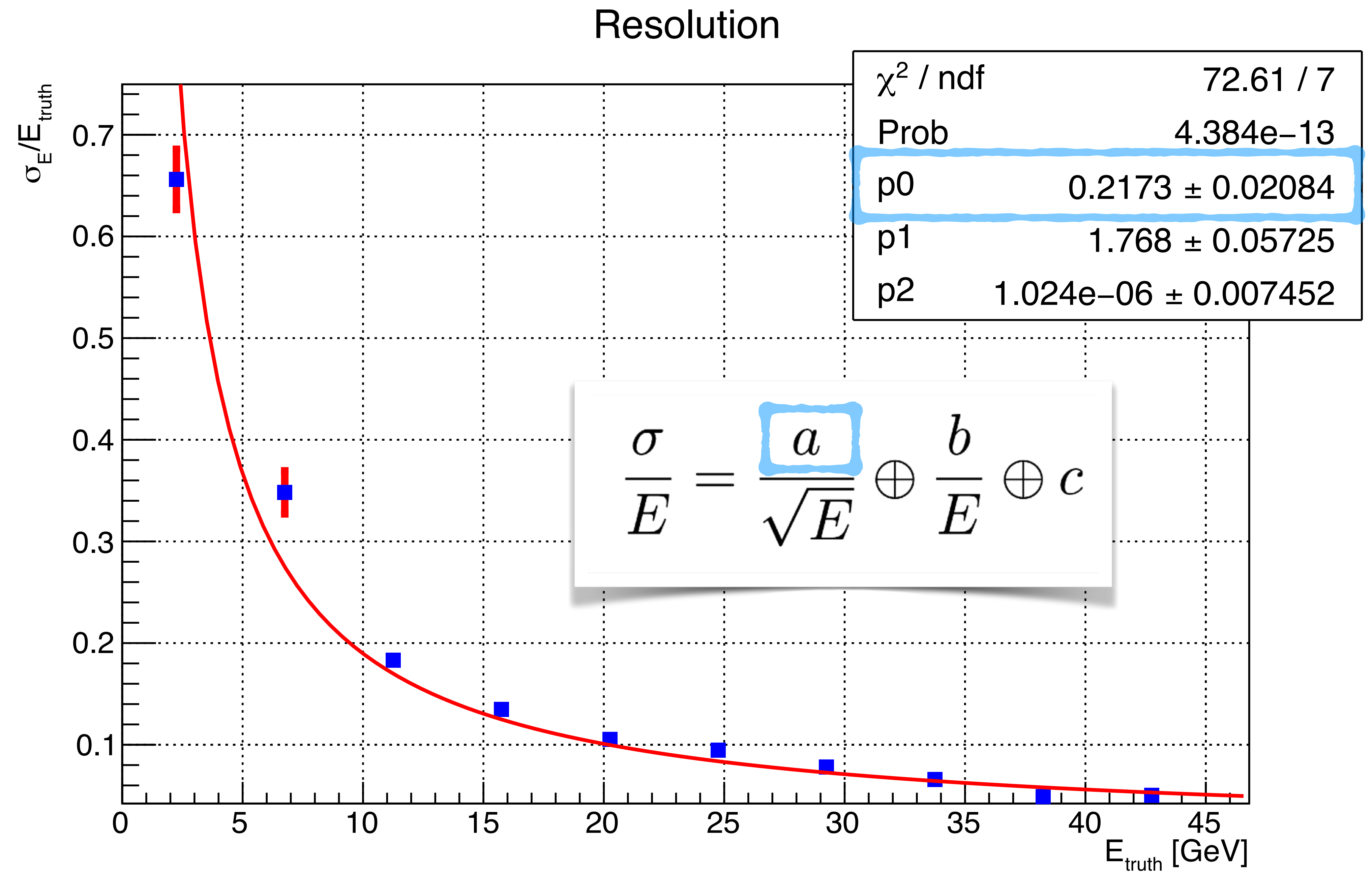
NN training using Tensorflow - Model

- Model loss: MeanSquaredError(), $\frac{1}{n}\sum_{i=1}^{n}\left(y_{\text{true}} - y_{\text{pred}}\right)^2$, optimised with respect to the simulated energy of the incoming electrons
- Adam, a stochastic optimiser, is used as optimiser to minimise the loss [Reference]
- Testing different START learning rate



Preliminary results on electron energy resolution





- Very preliminary results
- NN configurations might be under-performing
  - Too easy architecture? **Work in progress**

Conclusions and Next steps

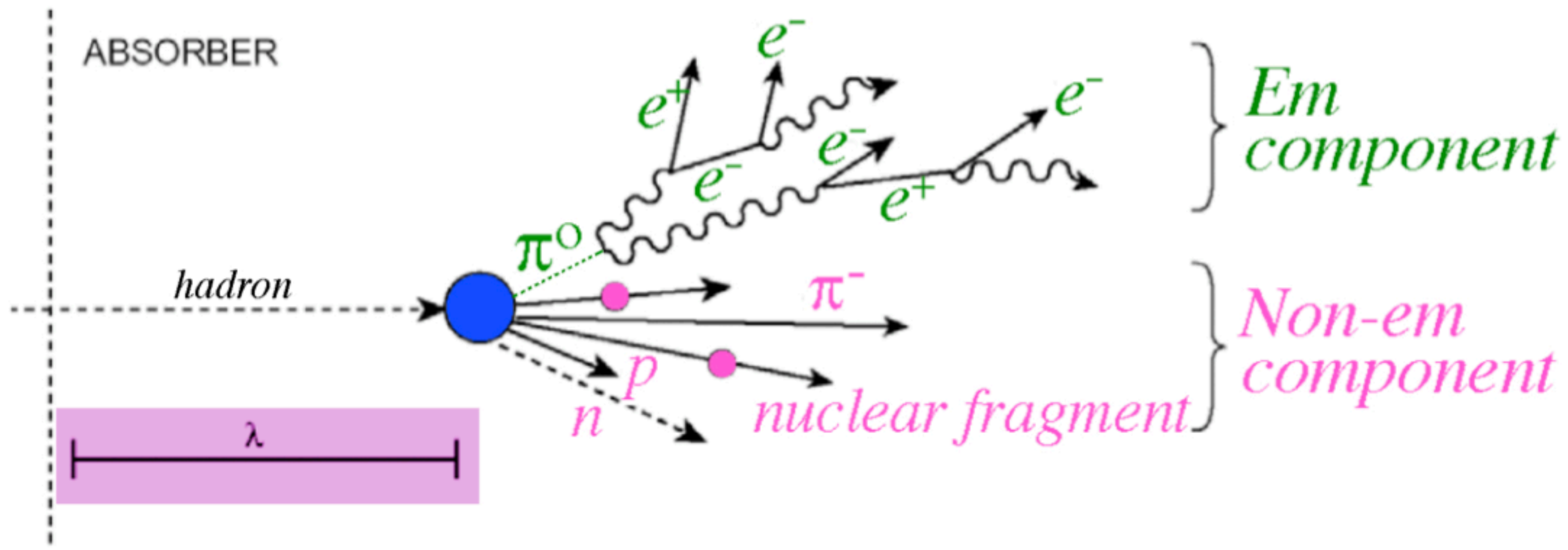
- The simulation works, the machinery for the PFA is in place and the first distributions are reasonable
- To do: tests for overtraining, test other NN architectures
- Train the NN on a newly simulated sample containing electrons with uniform energy (up to 125 GeV) and angular distributions, and info about the initial spatial coordinates of the impinging electrons at truth level —> **Simulation work in progress**
    - Perform the hyper parameter optimisation (*i.e.*: #layers, #epochs)
- Determine the energy and position resolution from NN, for electrons
- Repeat the above procedure also for π, K, μ, γ
- Plan to move to Pytorch for better optimisation with Pandora
- Long term goal: NN-based particle identification and jets reconstruction

Thanks a lot for listening!

Back-Up

Energy Measurement

- **Non-compensating calorimeters**: response to EM part different from that to non-EM part.
- The response ratio for electrons and charged hadrons is: $\frac{h}{e} = \eta < 1$
- The EM fraction of the shower, $< f_{em} >$, is energy dependent $\Rightarrow$ Non-linear calorimeter **response** to hadrons
- $< f_{em} >$ fluctuations largely determine energy **resolution** $\Rightarrow$ sampling the hadronic shower with two calorimeters with different *e/h* **boosts energy resolution**

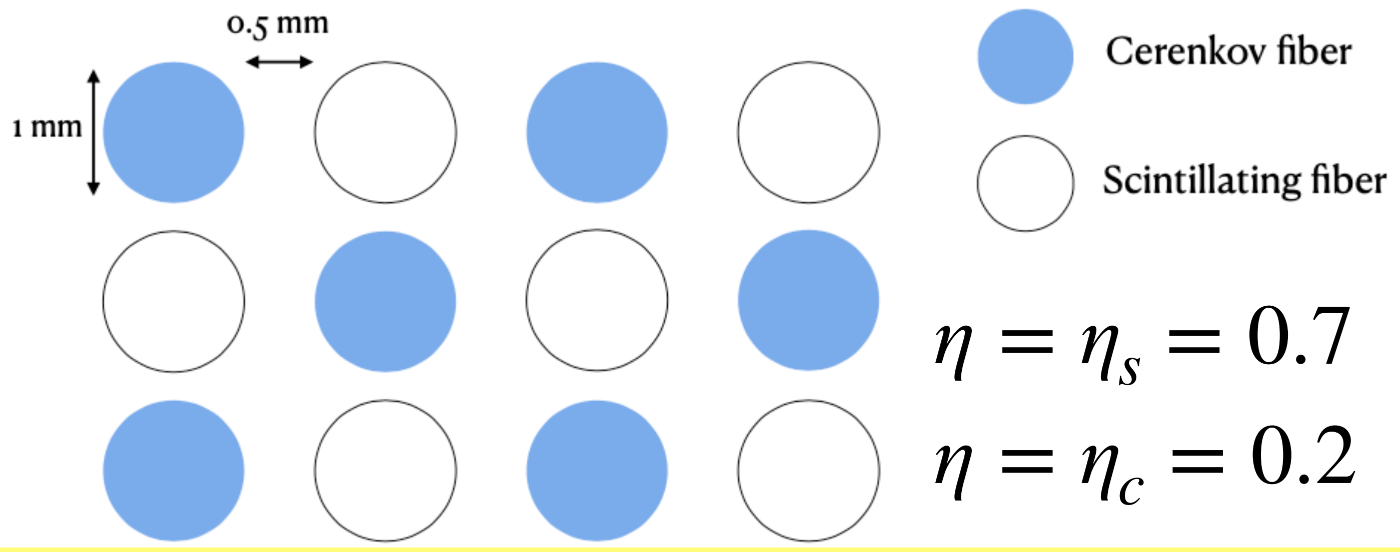


Hadron shower

$$S = E\left[f_{em} + \eta_s \cdot (1 - f_{em})\right]$$

$$C = E\left[f_{em} + \eta_c \cdot (1 - f_{em})\right]$$

$$\Rightarrow \quad \frac{C}{S} = \frac{\left[f_{em} + \eta_c \cdot (1 - f_{em})\right]}{\left[f_{em} + \eta_s \cdot (1 - f_{em})\right]}$$

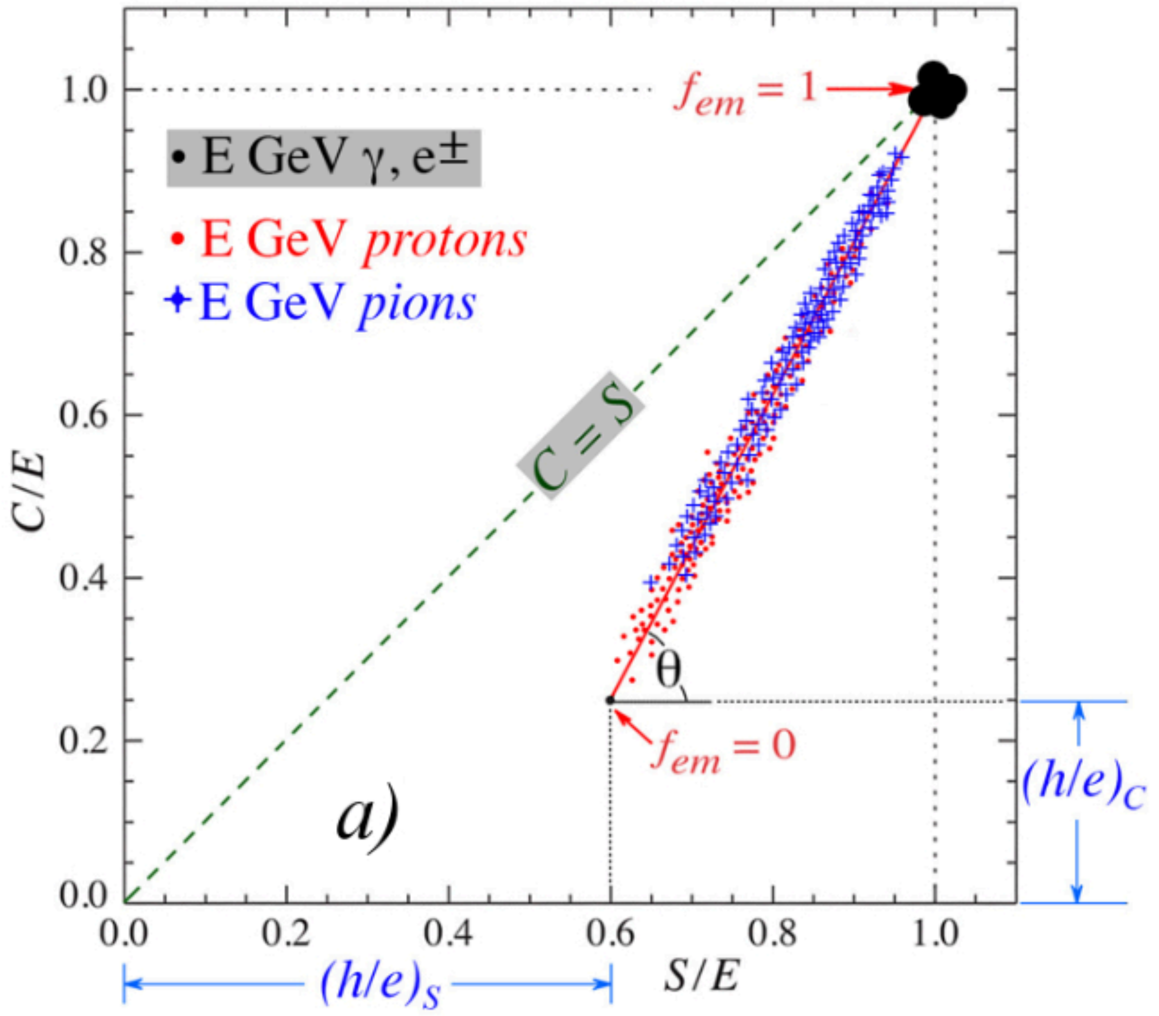


$$\eta = \eta_s = 0.7$$

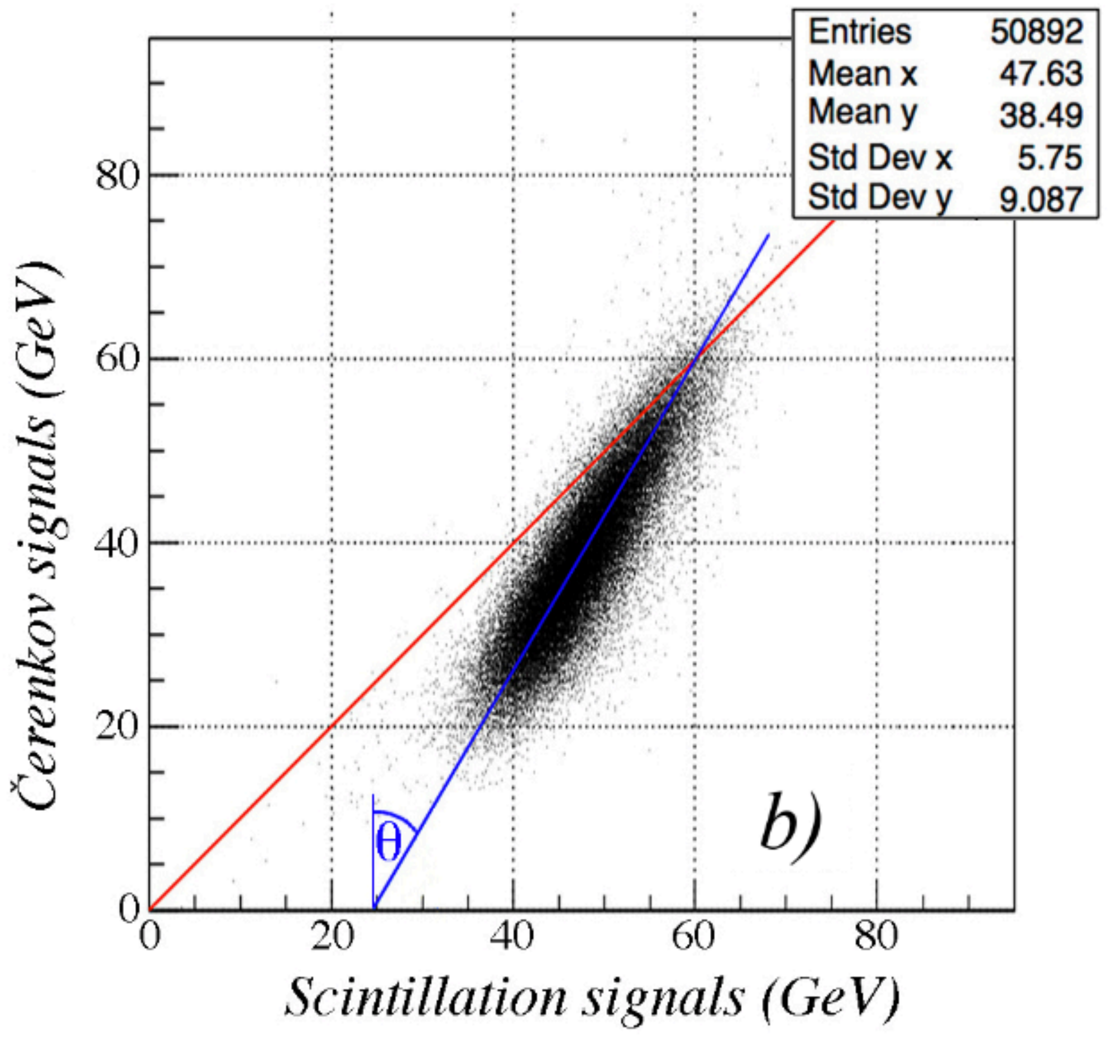
$$\eta = \eta_c = 0.2$$

$$\chi = \frac{1 - \eta_s}{1 - \eta_c} = \cot(\theta)$$

$$E = \frac{S - \chi C}{1 - \chi}$$



a) **Scatter plot of C/E versus S/E in a dual-readout calorimeter for p and π**



b) **Scatter plot of C and S signals for 60 GeV pions in the RD52 lead-fiber calorimeter**

Software Implementation

(1) Geant4 jets simulation: outside the scope of this project, provided by Iacopo and his team in KEY4HEP format

(2) Extract particle/calorimeter info from simulations

- **New code** in *IDEADetectorSIM git repo*: *https://github.com/HEP-FCC/IDEADetectorSIM/tree/master/ParticleFlow_k4pandora*
- It is an algorithm that reads KEY4HEP format and produces an output to perform a Neural Network training
- **Preliminary plots** of electrons and photons kinematic variables in the *next slides*

(3) Clustering: several clustering algorithms already on the market, *i.e.* NN based reconstruction algorithm for LAr TPC for the DUNE experiment, with interfaces to run Pandora using Torch Data format —> Collaboration in progress with DUNE team

(4) NN based particle identification: use as basis a particle flow approach, which aims at identifying each single particle inside a jet

- Machine Learning with *TensorFlow*
- CPU & GPU installation performed on Roma Tre cluster
  - The site is equipped with about 50 server (mainly based on Blade technology) with a total amount of cores available (or VCPU) of about 1500 interconnected with Infiniband (DDR 20Gbps e QDR 40Gbps)
  - The site has also 2 Graphical Processor Unit (GPU) K 80 (4 in total: 2 x K40), where jobs can be parallelised if needed
  - There is a storage system present in the cluster for a total amount of about 700TB
  - **Extensive innovation next year**, in order to double the CPU and storage system

(4) NN based jet reconstruction: construct a regression algorithm for particle-jet assignment and jet energy reconstruction

Overview of the Project

3

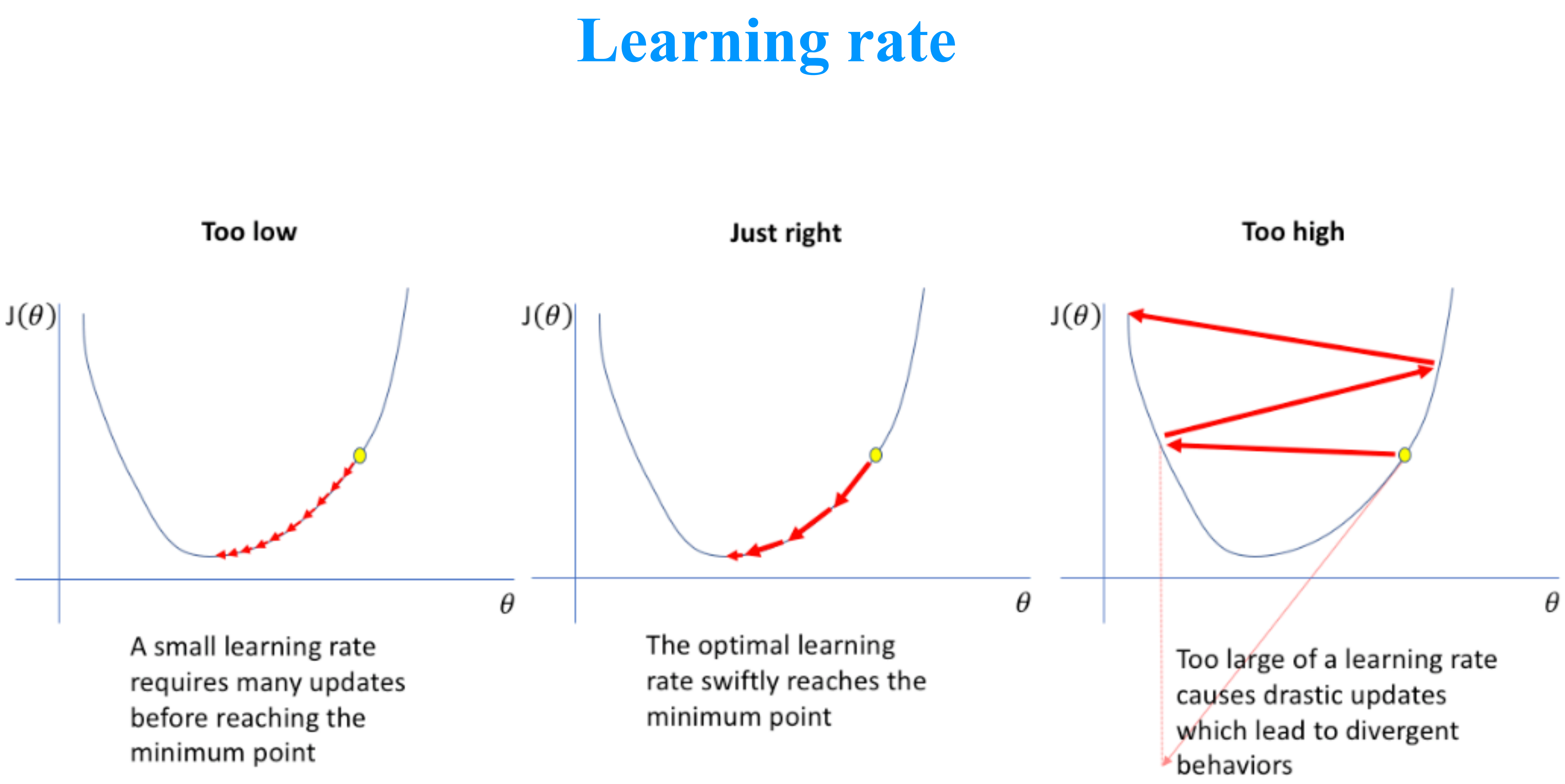
Clustering ?

- **clustering seems the obvious way to simplify conceptually the algorithm**

1. Identify energy deposits released by a single particle, collect them, associate the cluster to a track, apply PID at cluster level;

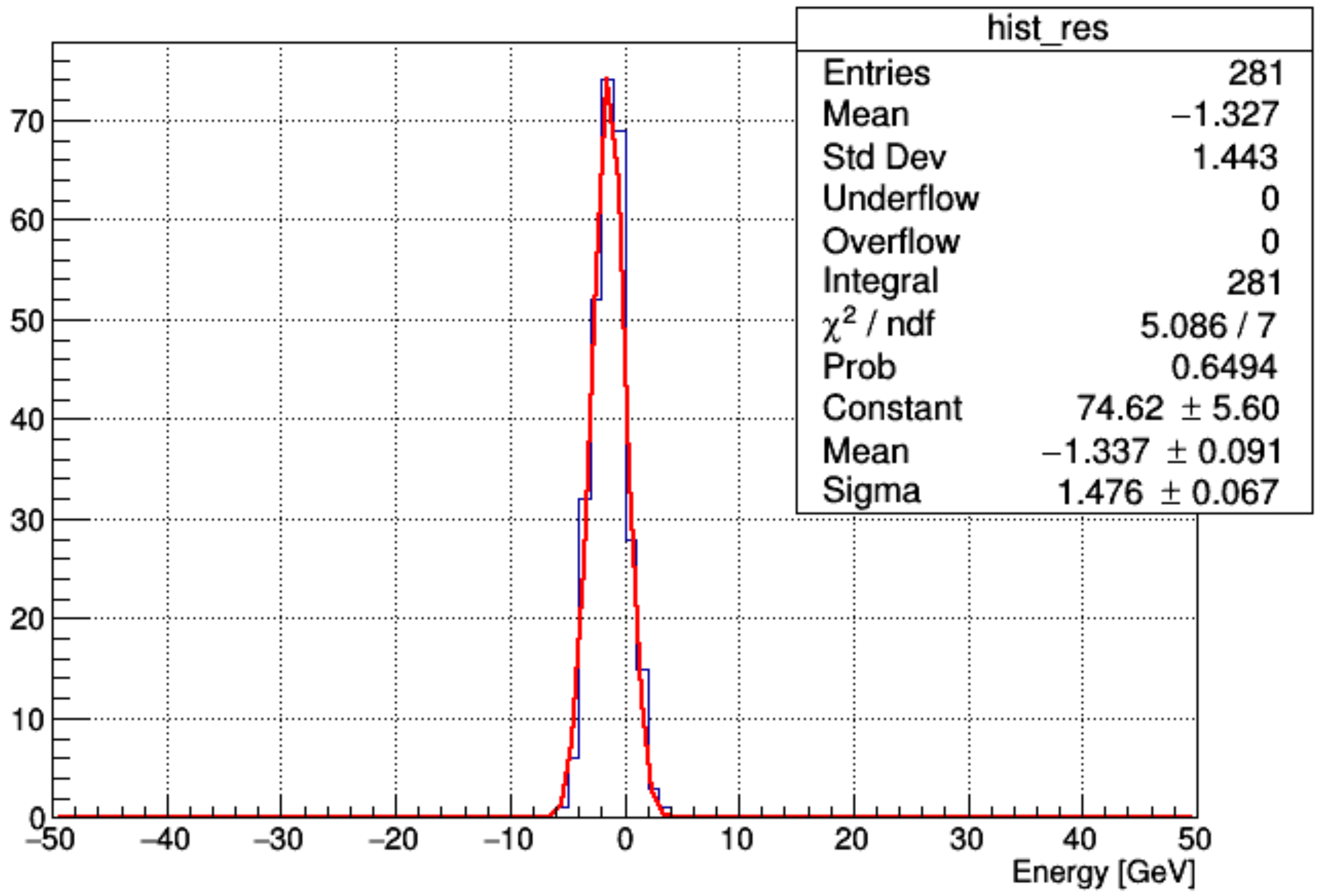
2. Draw back: the NN could miss information outside the cluster, how to minimize the risk ? NN based clustering using track informations ? We need parent energy deposit information to make a NN based clustering

Learning rate

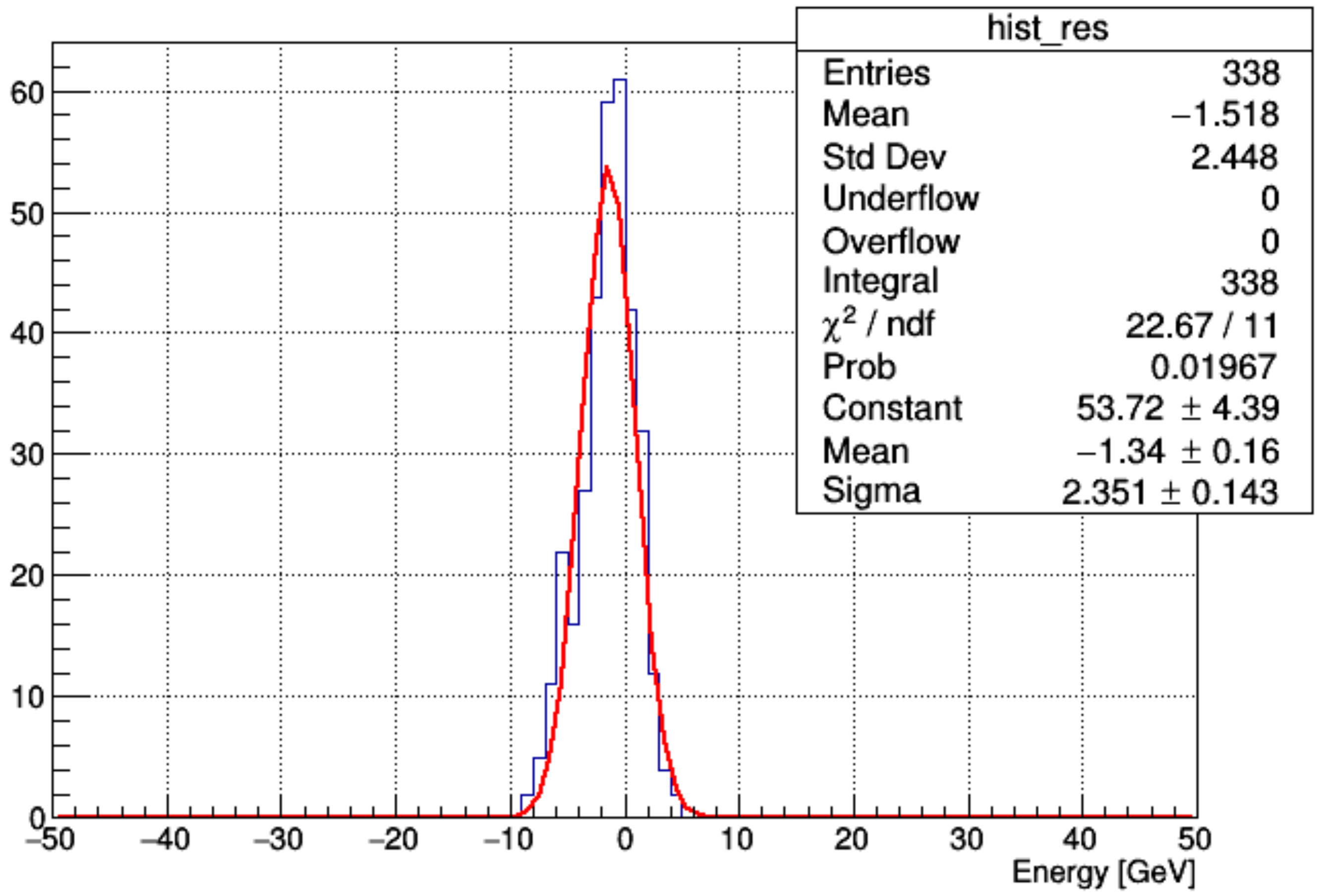


Fit range = all range

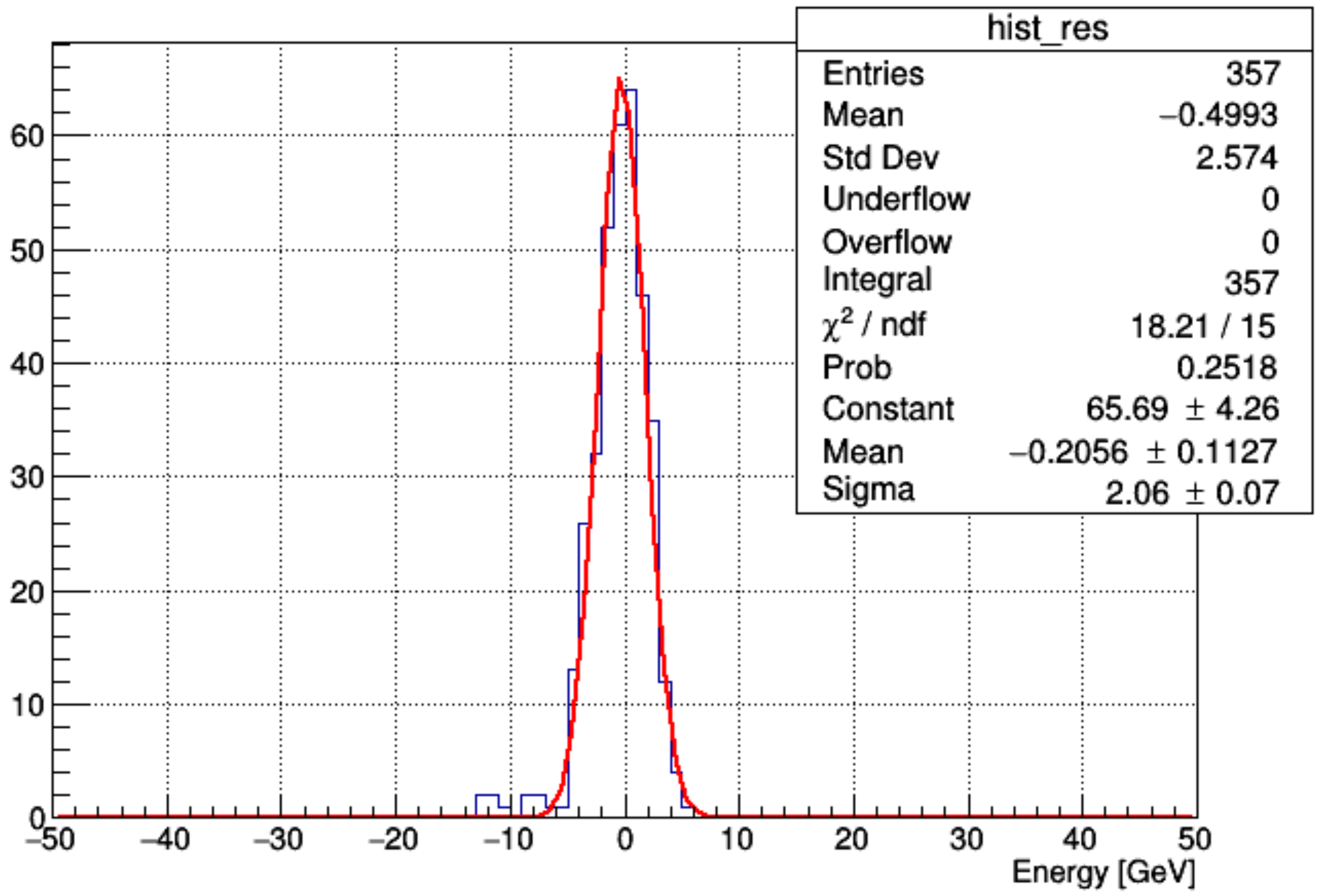
0 GeV < E_{truth} < 4.5 GeV



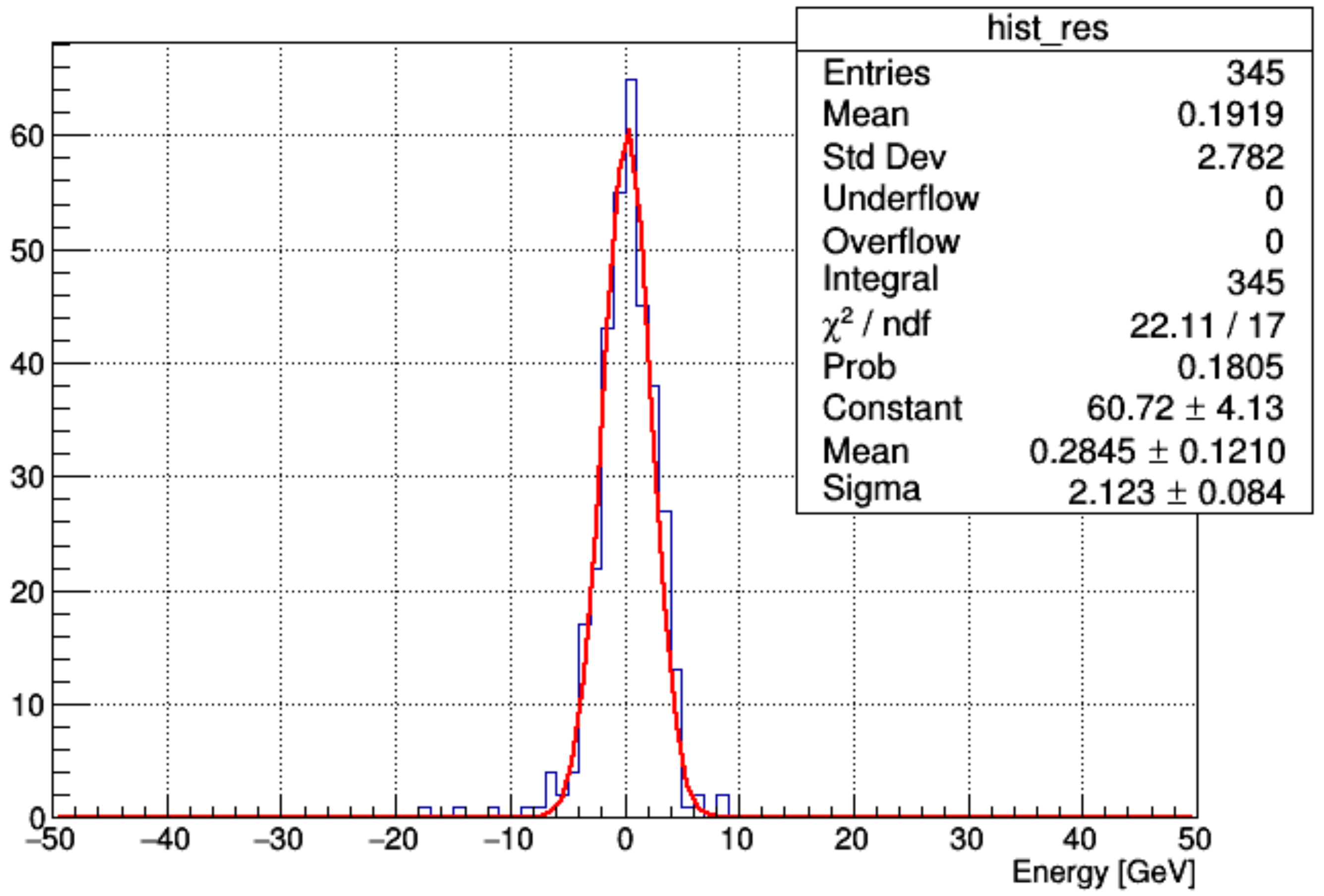
4.5 GeV < E_{truth} < 9 GeV



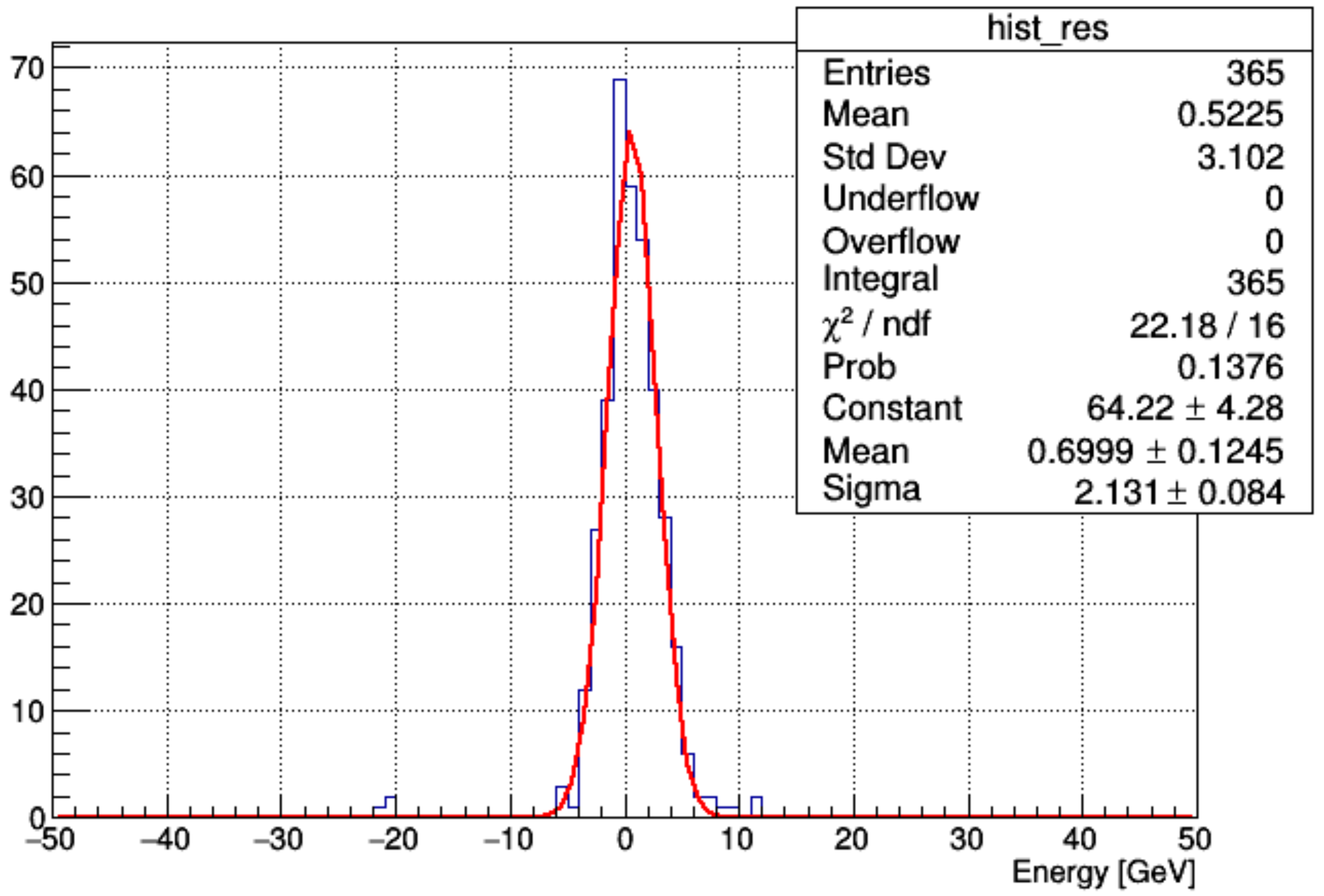
9 GeV < E_{truth} < 13.5 GeV



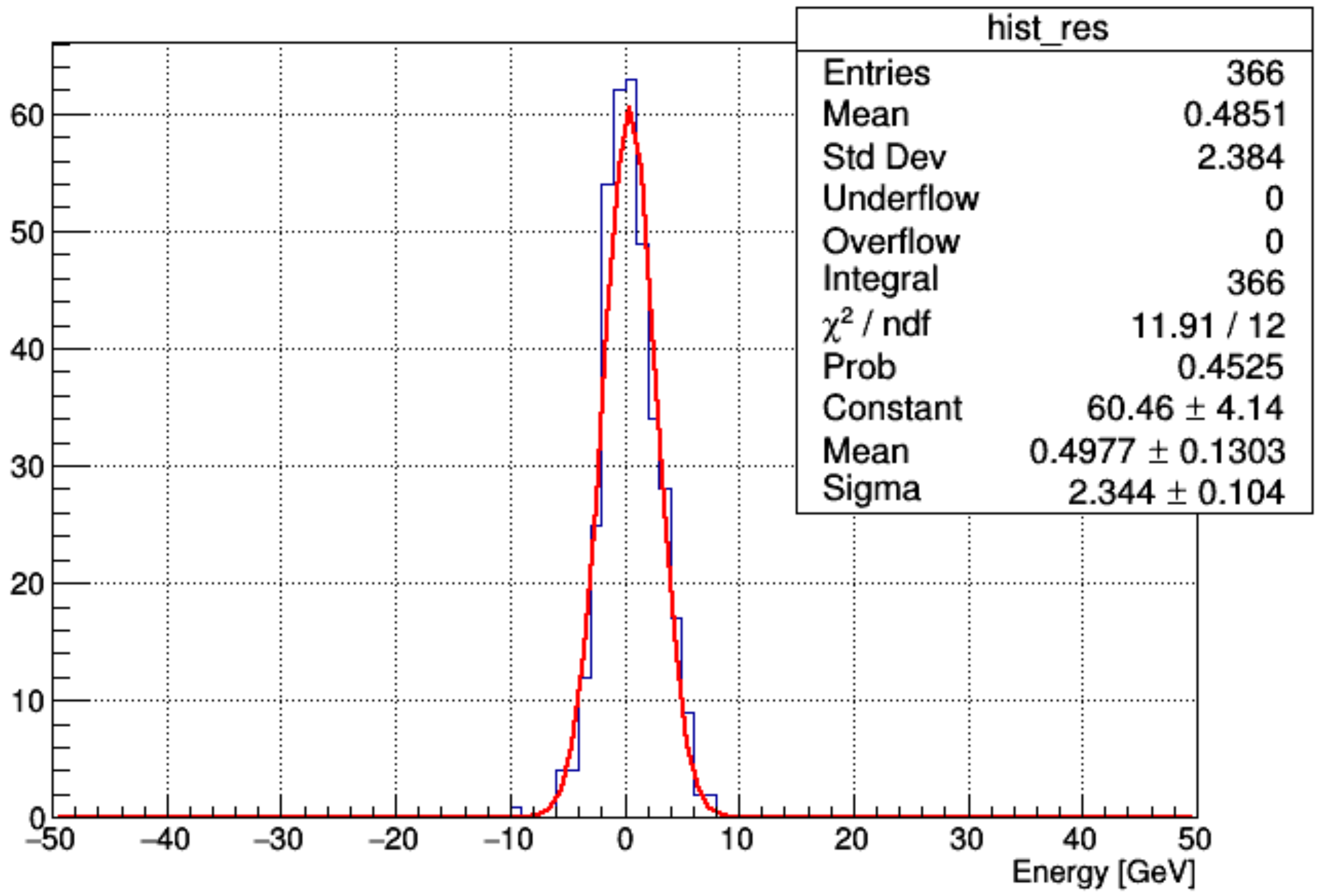
13.5 GeV < E_{truth} < 18 GeV



18 GeV < E_{truth} < 22.5 GeV

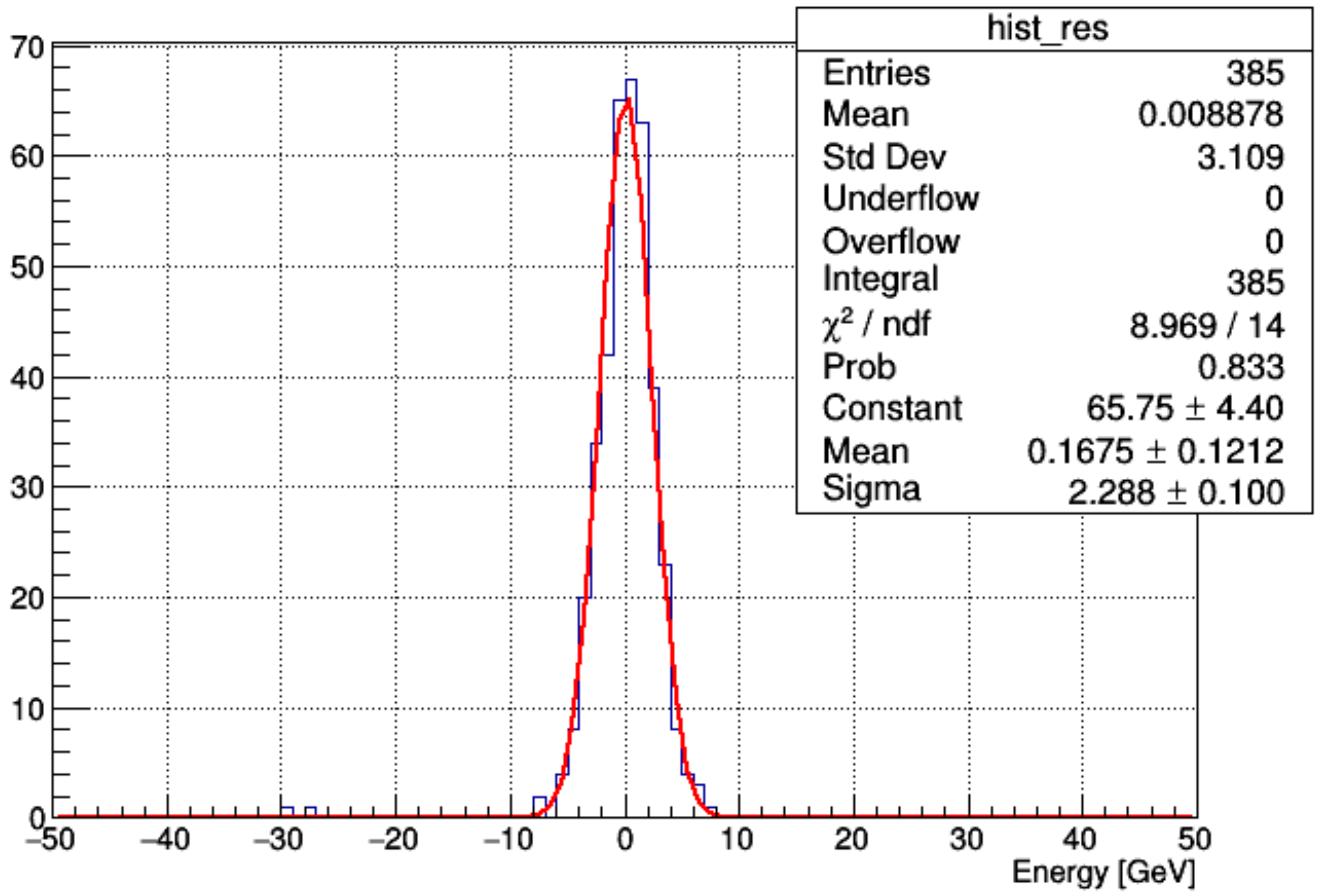


22.5 GeV < E_{truth} < 27 GeV

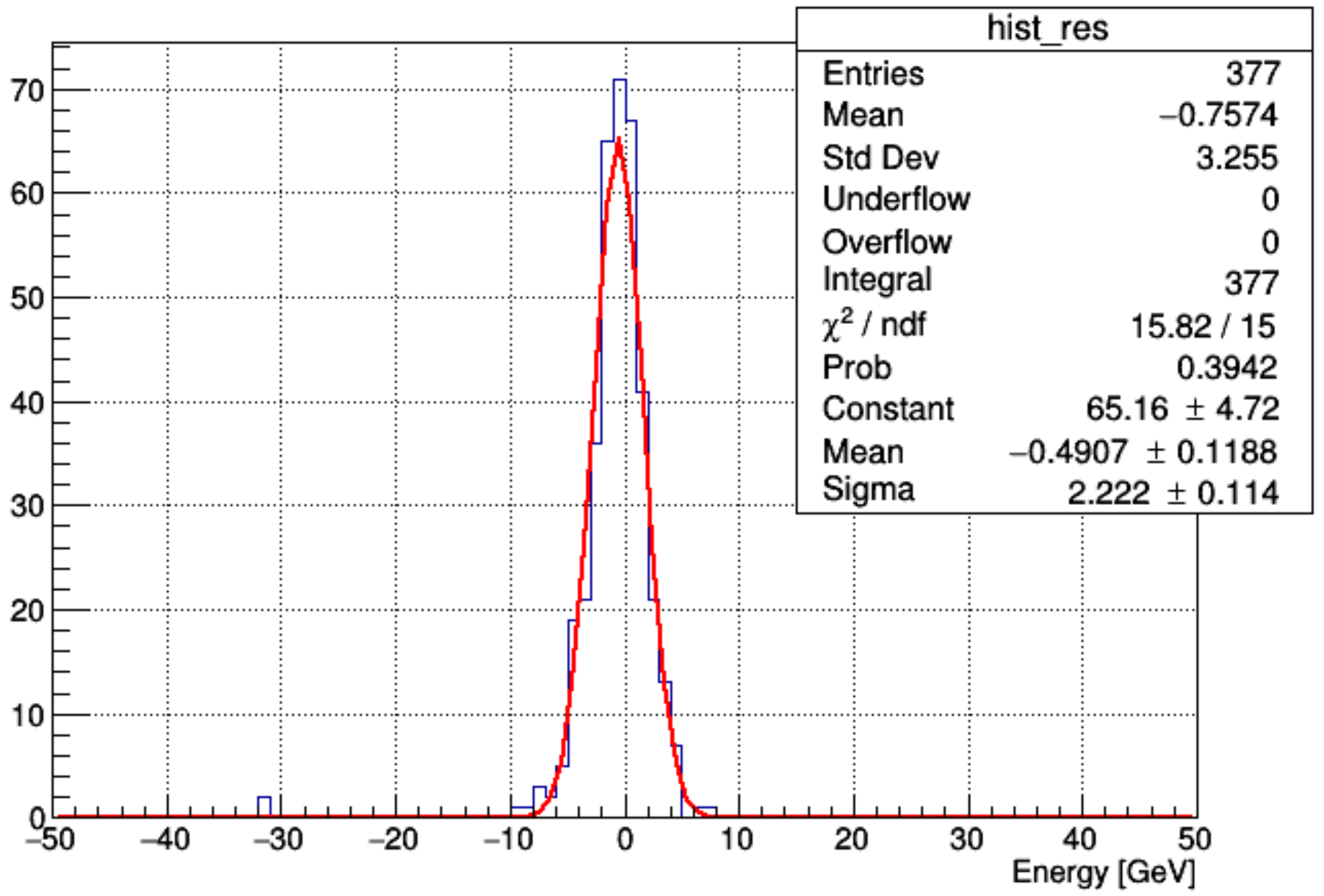


Fit range = all range

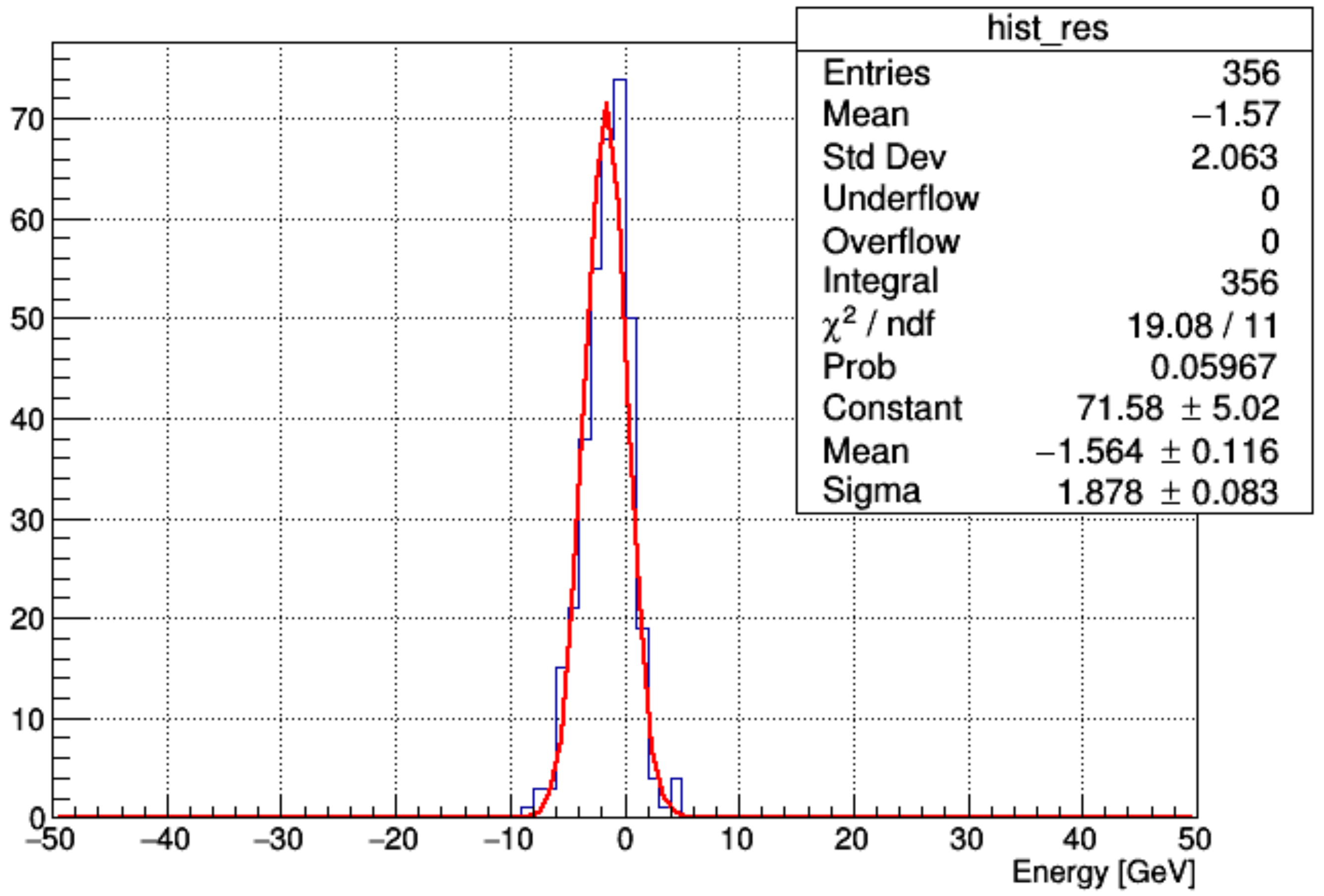
27 GeV < E_{truth} < 31.5 GeV



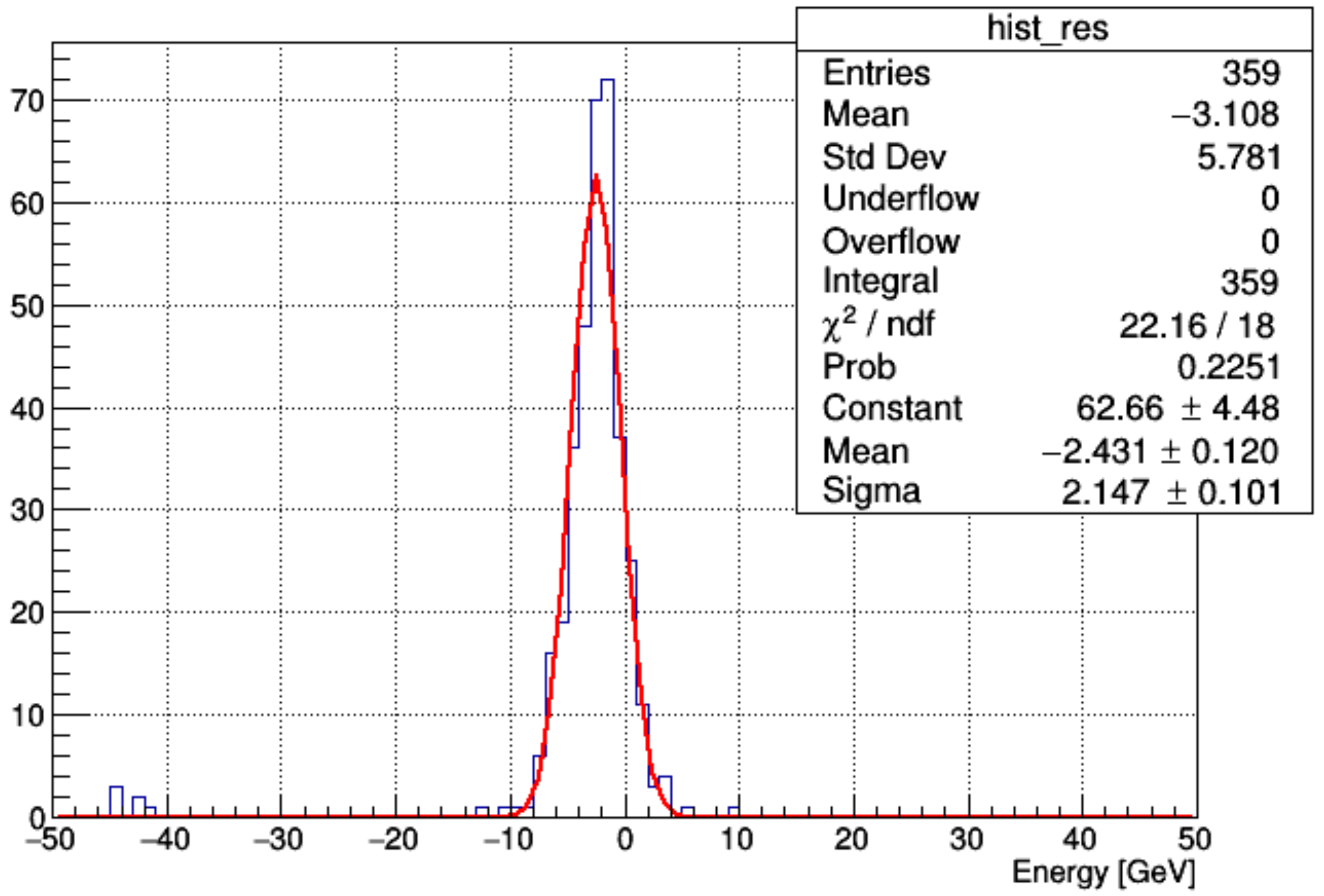
31.5 GeV < E_{truth} < 36 GeV



36 GeV < E_{truth} < 40.5 GeV



40.5 GeV < E_{truth} < 45 GeV



Simulated events: e^+e^- —>$Z(\nu\nu)H(\gamma\gamma)$

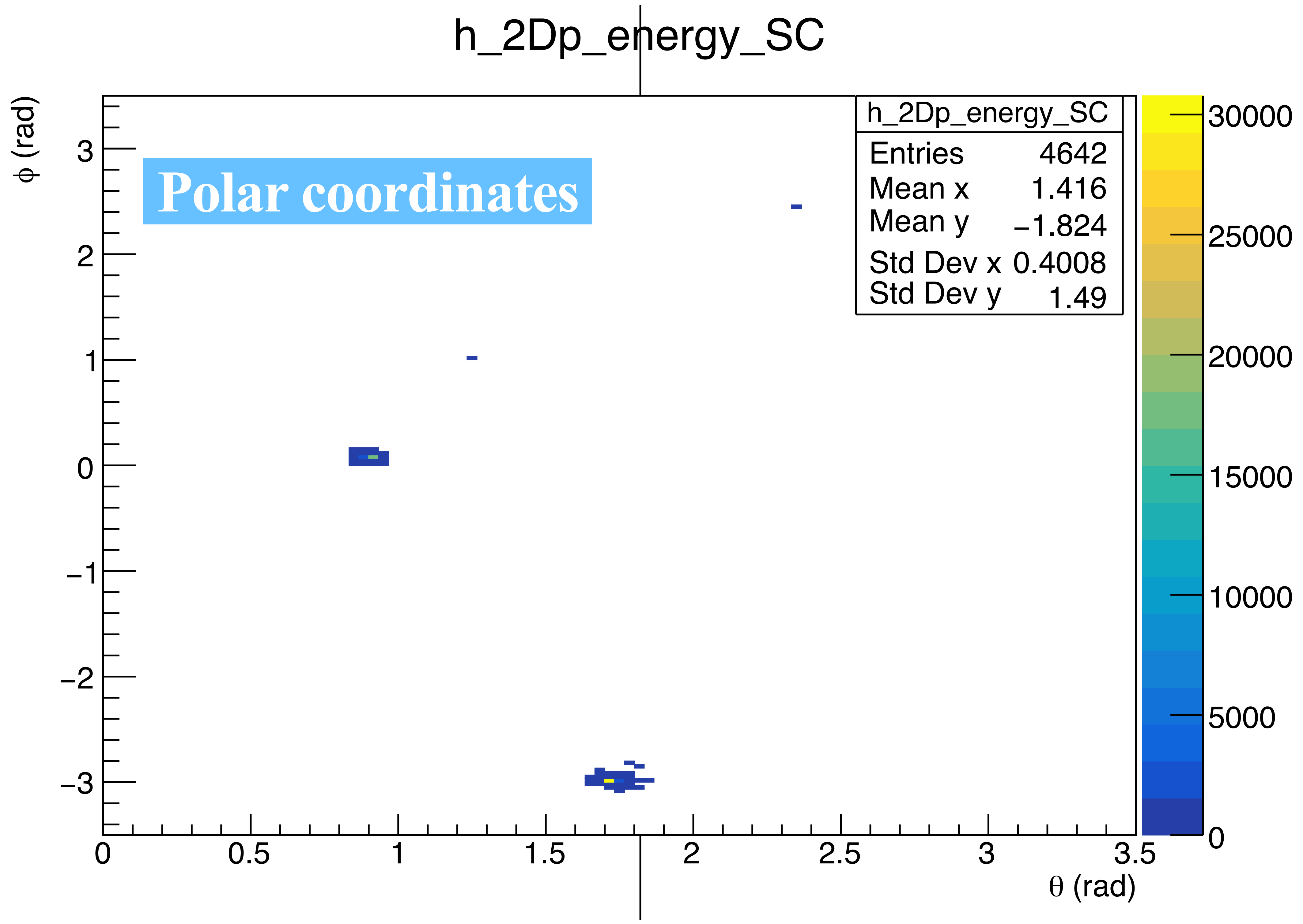
Energy deposits - e^+e^- —>$Z(\nu\nu)H(\gamma\gamma)$

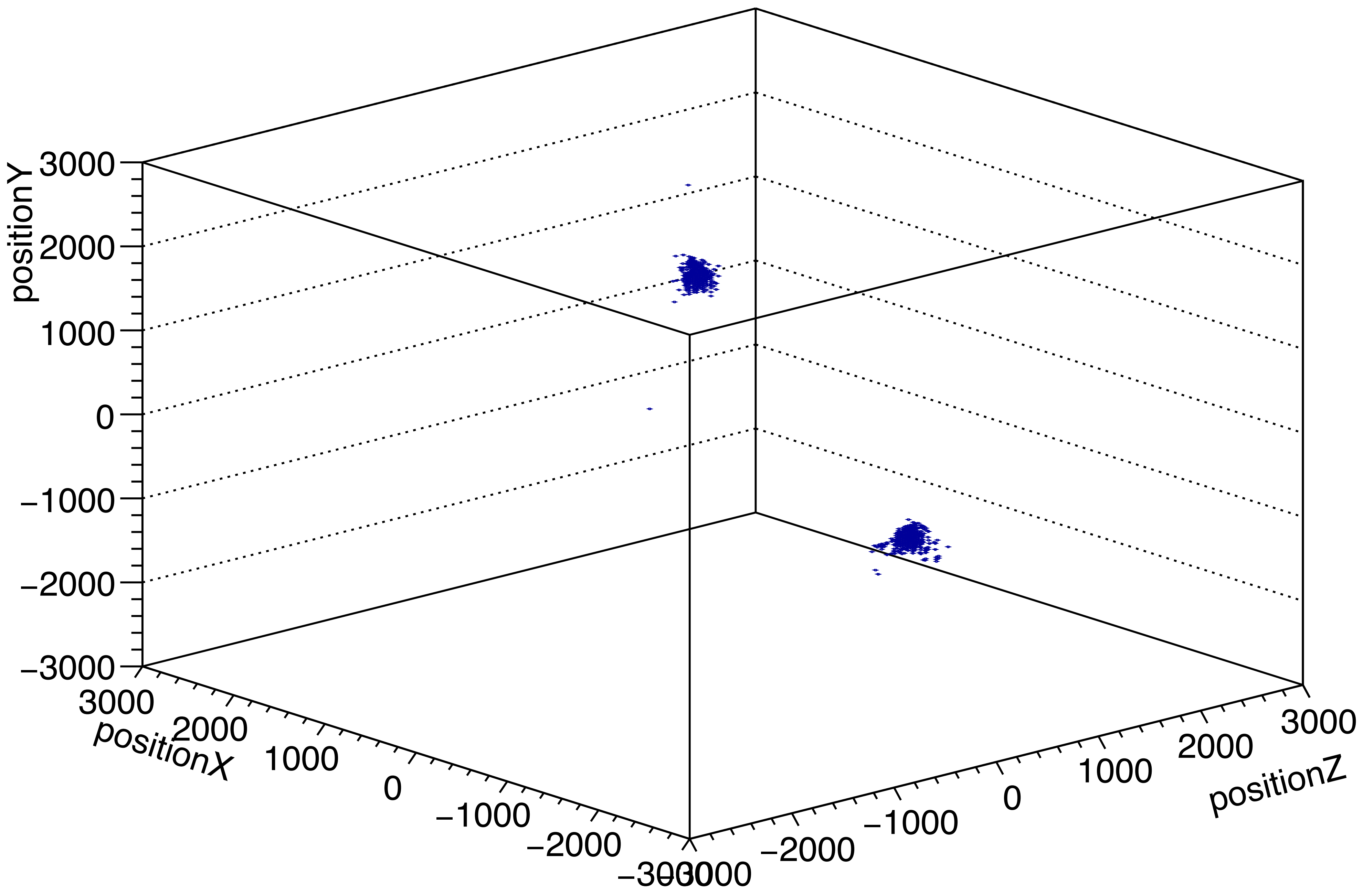
Dumper Algorithm output

Combined fibres

Photon deposits

Input file: EDMOutput_Higgs.root



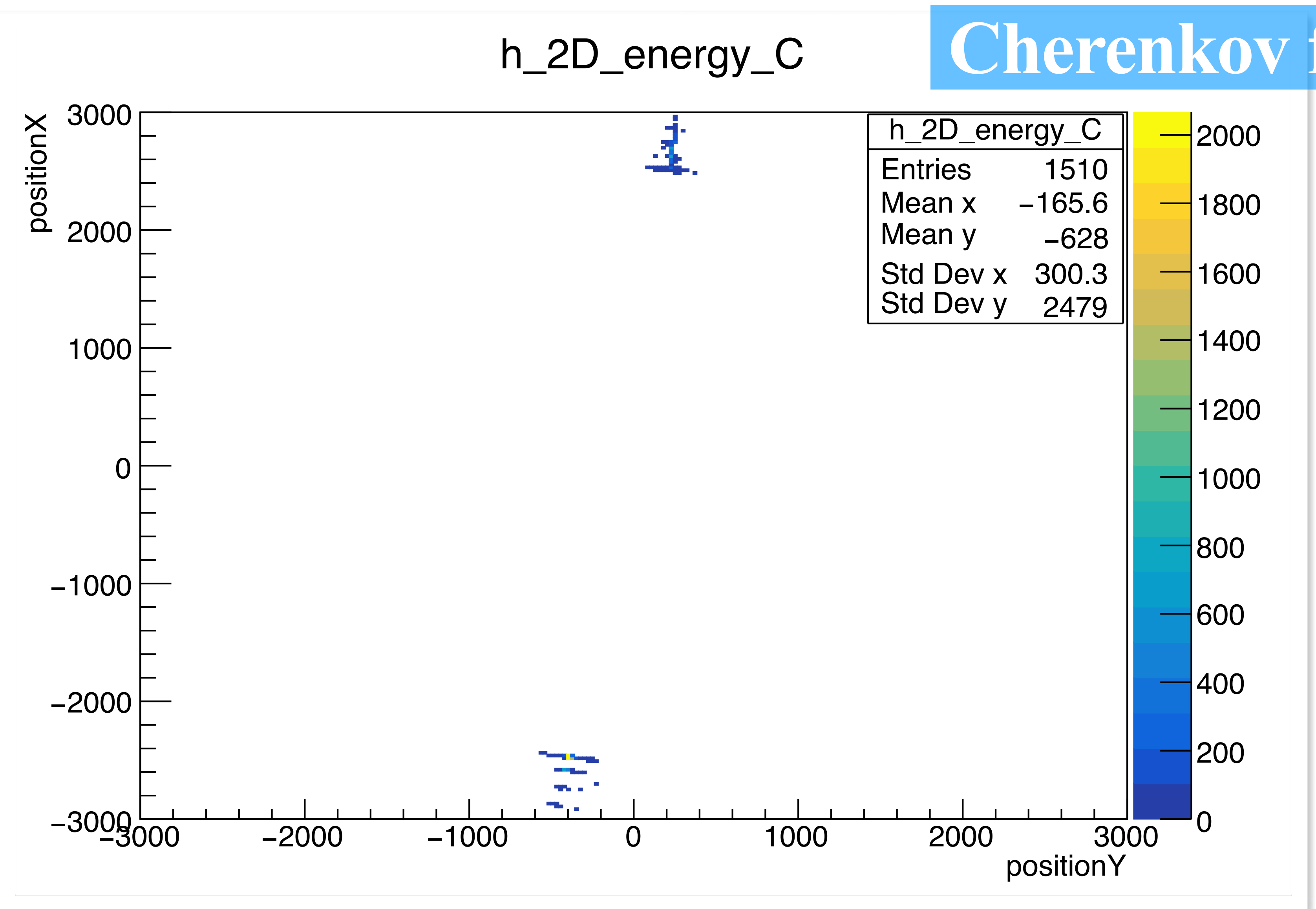


3D plot, cartesian coordinates

Energy deposits - e+e- —>Z(νν)H(γγ)

Dumper Algorithm output

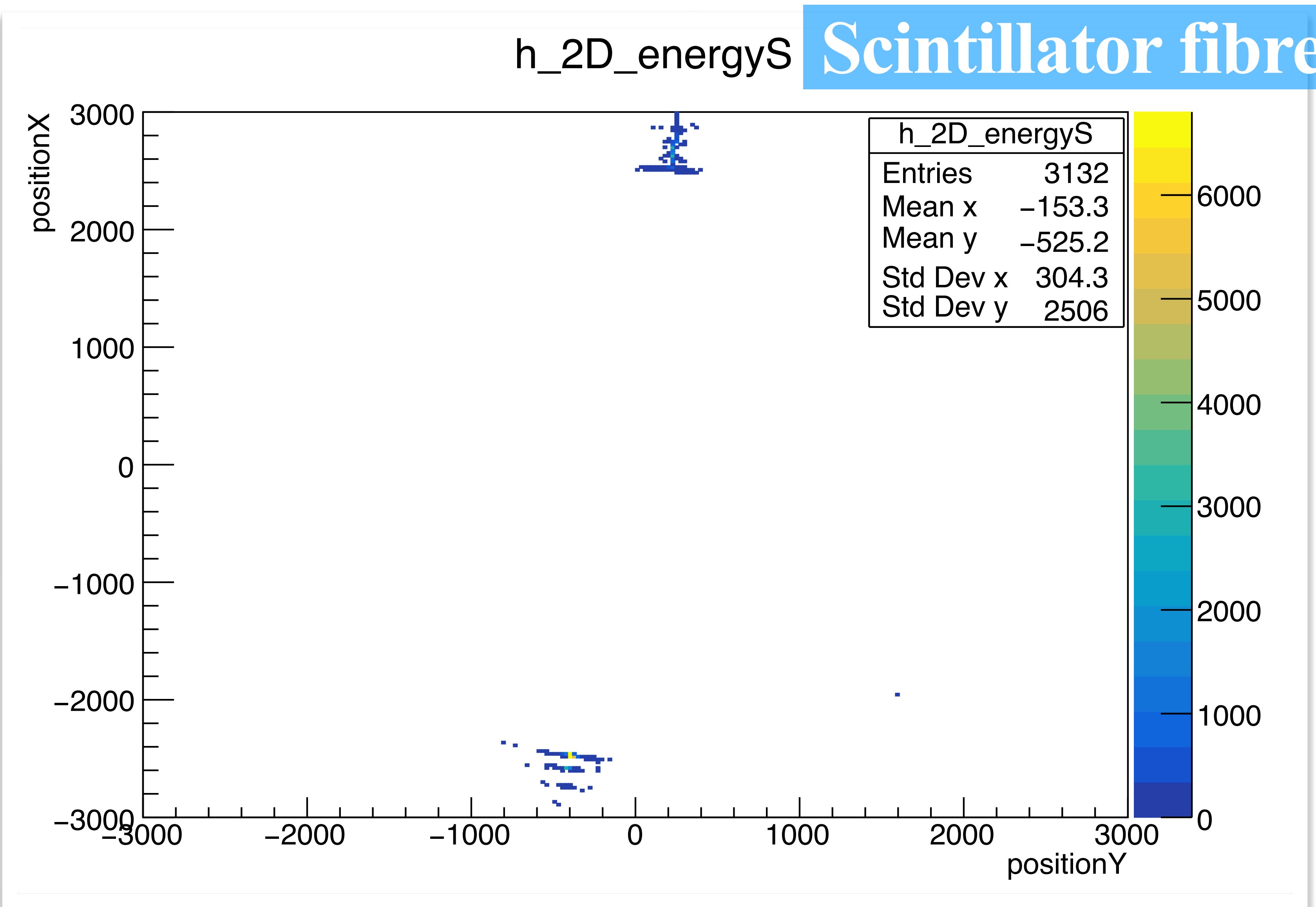
Cherenkov fibres



Photon deposits

Input file: EDMOutput_Higgs.root

Scintillator fibres



Combined fibres