Particle Flow for Dual Read-Out Calorimeter Status Report

Michela Biglietti[1], Adelina D'Onofrio[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

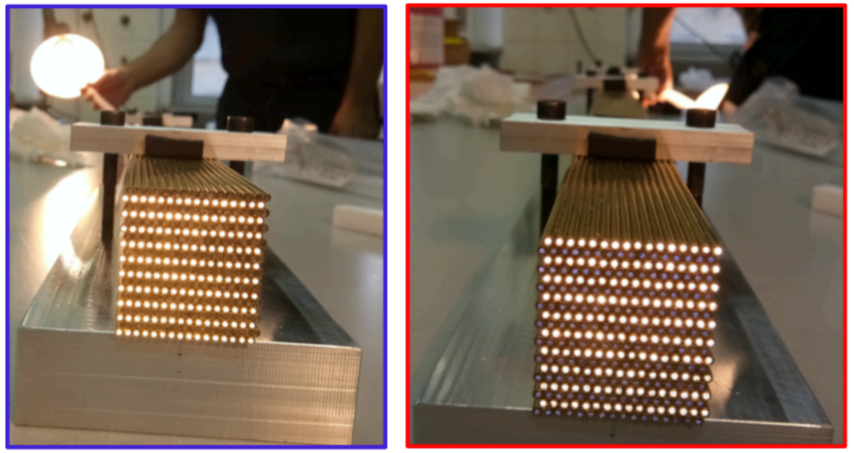
12th October 2022

AIDA innova

INFN Istituto Nazionale di Fisica Nucleare SEZIONE DI ROMA TRE

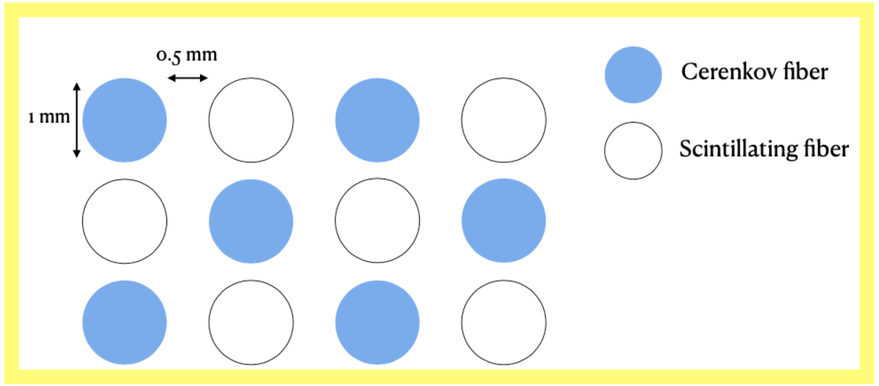
Dual Read-out Calorimeter for the IDEA detector

- 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



Scintillation fibers

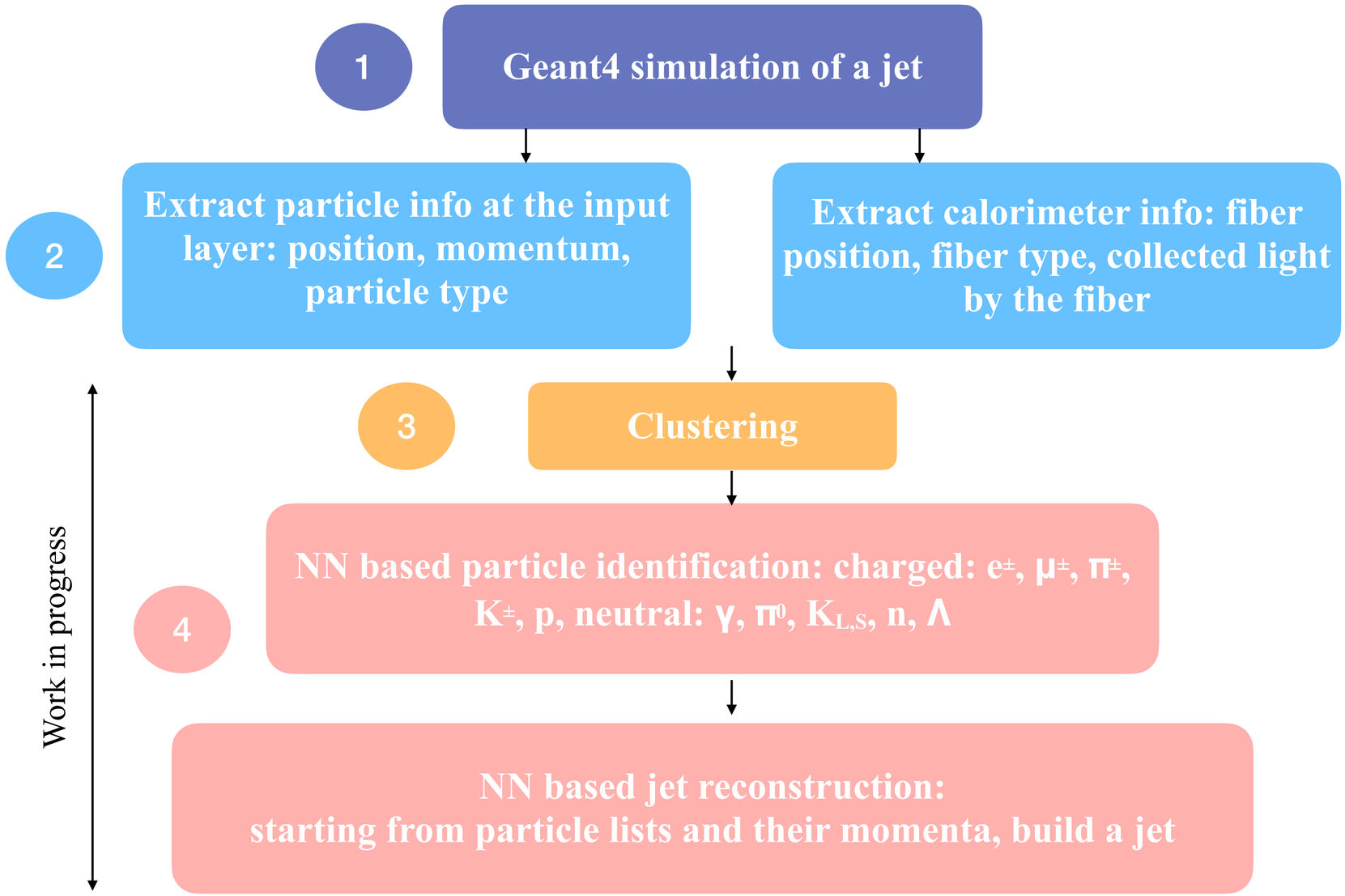
Cherenkov fibers



Goal of the project

- 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
- 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
- Our goal: maximise the energy resolution of the dual read-out calorimeter exploiting NNs and taking as input all the available kinematic variables

Overview of the Project



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*

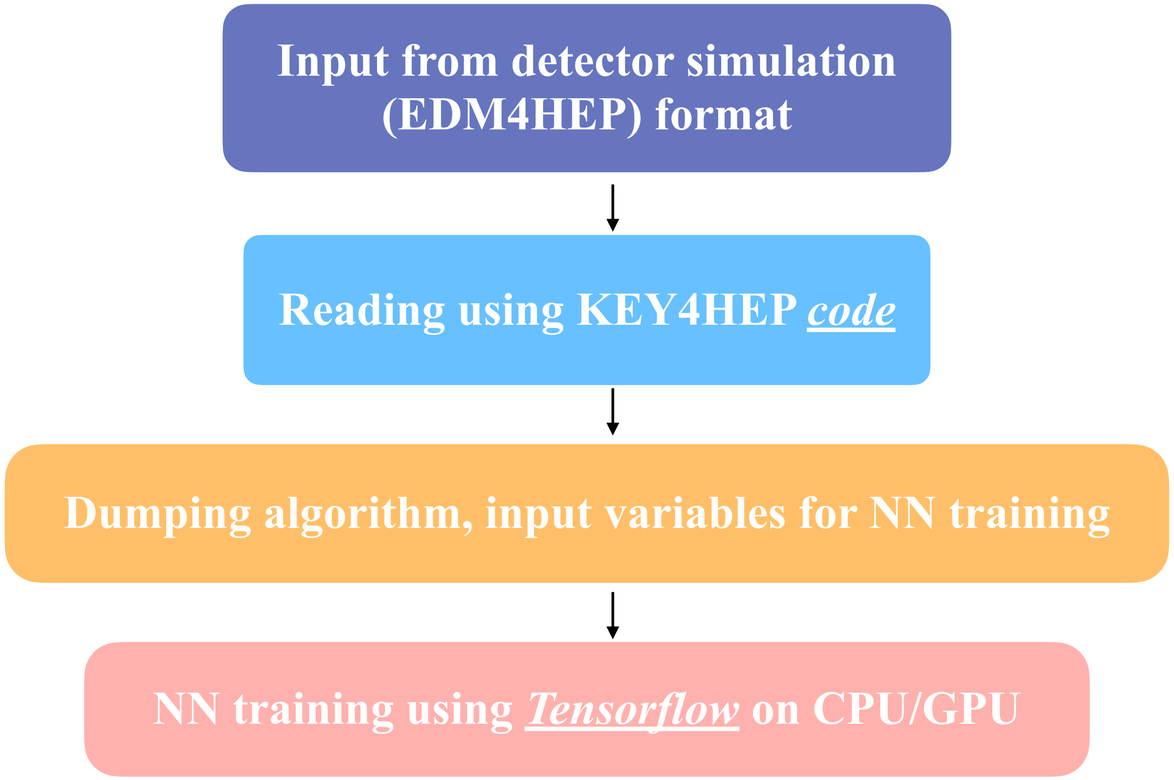
(3) Clustering: several clustering algorithms already on the market, *i.e.* NN based reconstruction algorithm for LAr TPC for the DUNE experiment, with interfaces to run Pandora using Torch Data format —> Collaboration in progress with DUNE team

(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

Software Implementation - Block Scheme



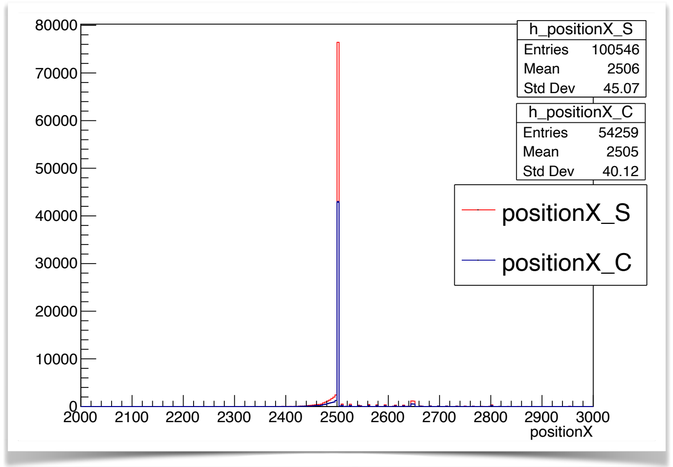
- Geant-based simulations of the IDEA detector for *e, π, K* with energy and angular uniform distribution (thanks for the inputs!)
- Target: build a NN able to reconstruct the energy and the position of the impinging particles and identify them
  - Regression and classification (to discriminate *e, π, K*) algorithms implemented in a single NN
- State of the art:
  - NN studies performed on an input sample containing 20 GeV electrons, training performed for energy regression

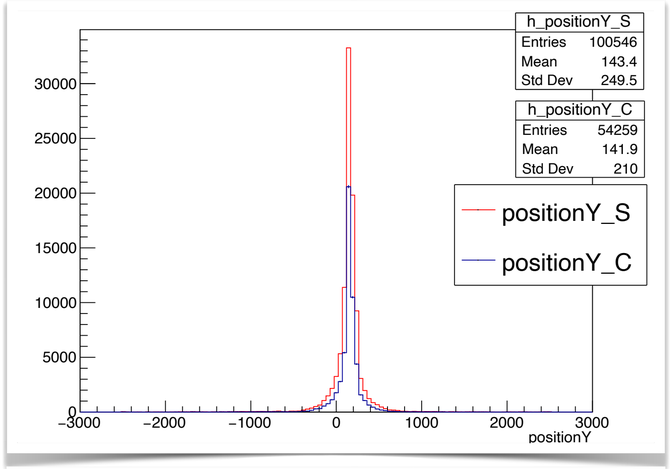
Kinematic distributions - 20 GeV electrons

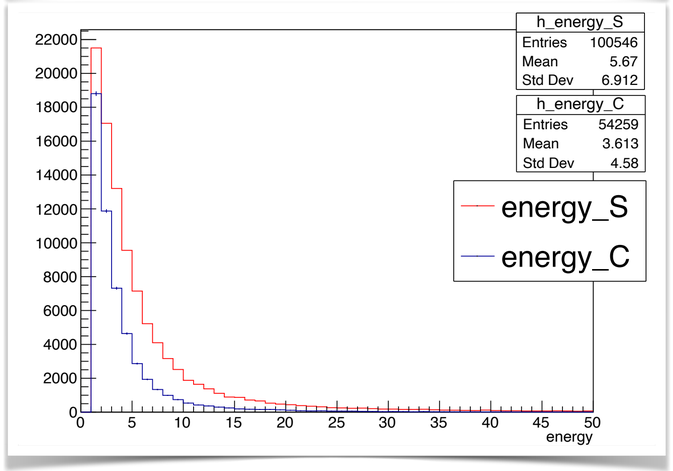
All events used (#102)

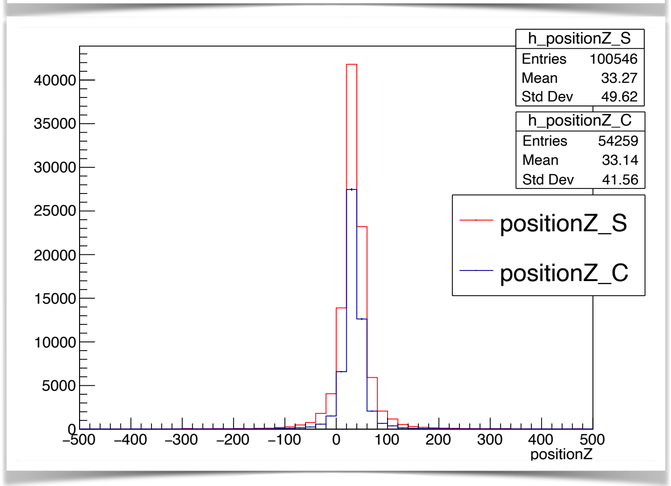
- Position and energy collected in the scintillating (S) **and** Cerenkov (C) fibres in 100 events **simulating** impinging electrons of 20 GeV

Dumper Algorithm output









Energy deposits - 20 GeV electrons

Dumper Algorithm output

Electron deposits

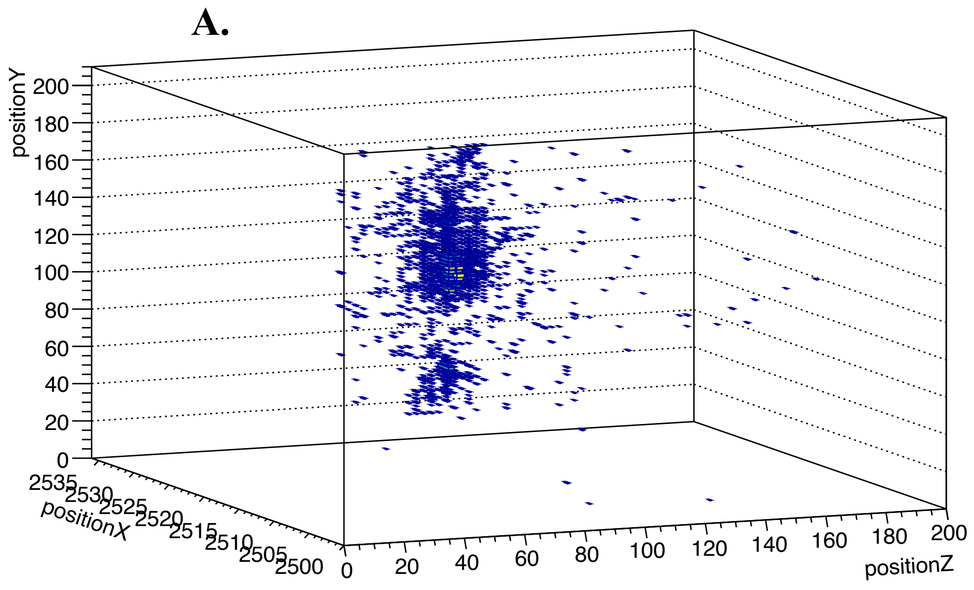
- Energy collected in the scintillating (S) **and** Cerenkov (C) fibres in 100 events **simulating** impinging electrons of 20 GeV

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

B. Energy deposited in the scintillating fibres, polar coordinates

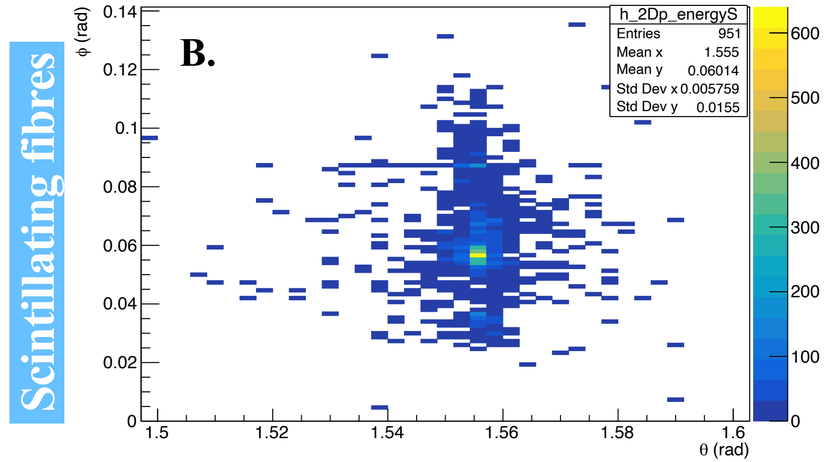
C. Energy deposited in the Cherenkov fibres, polar coordinates

Combined fibres

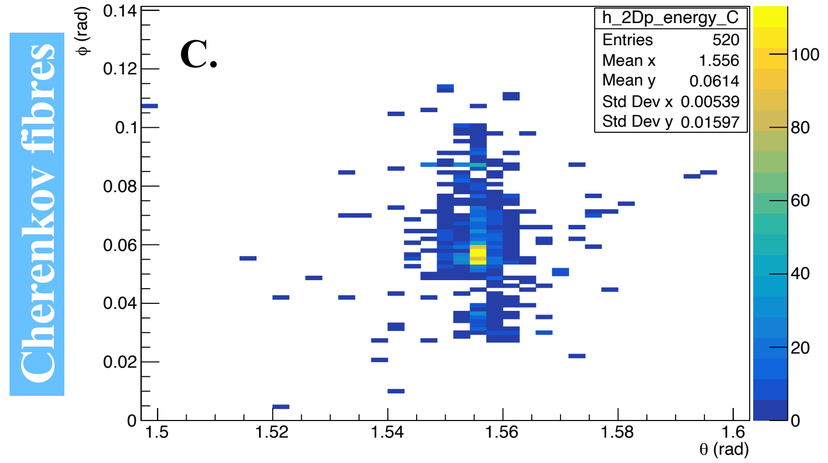


3D plot, cartesian coordinates

Scintillating fibres



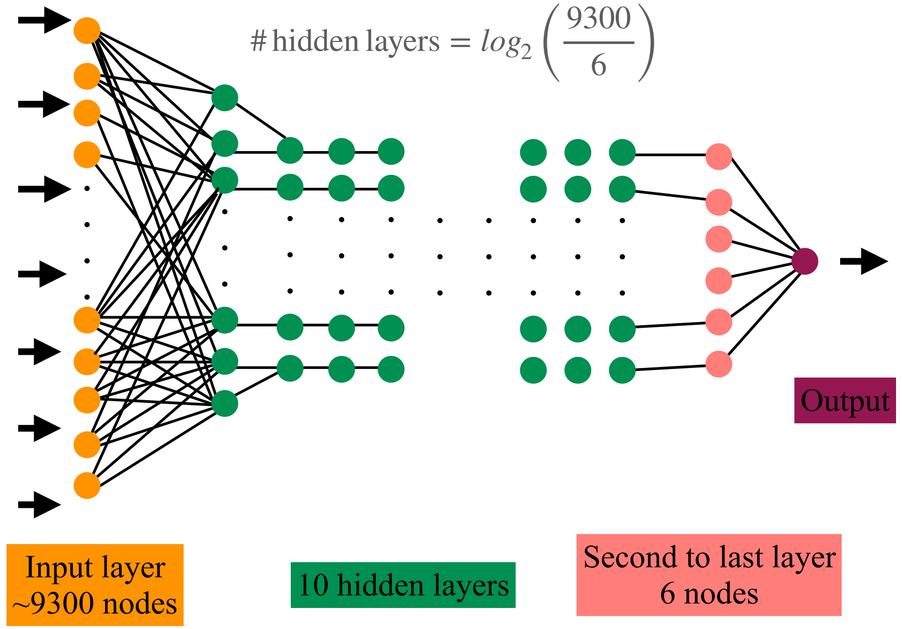
Cherenkov fibres



2D plots, Polar coordinates

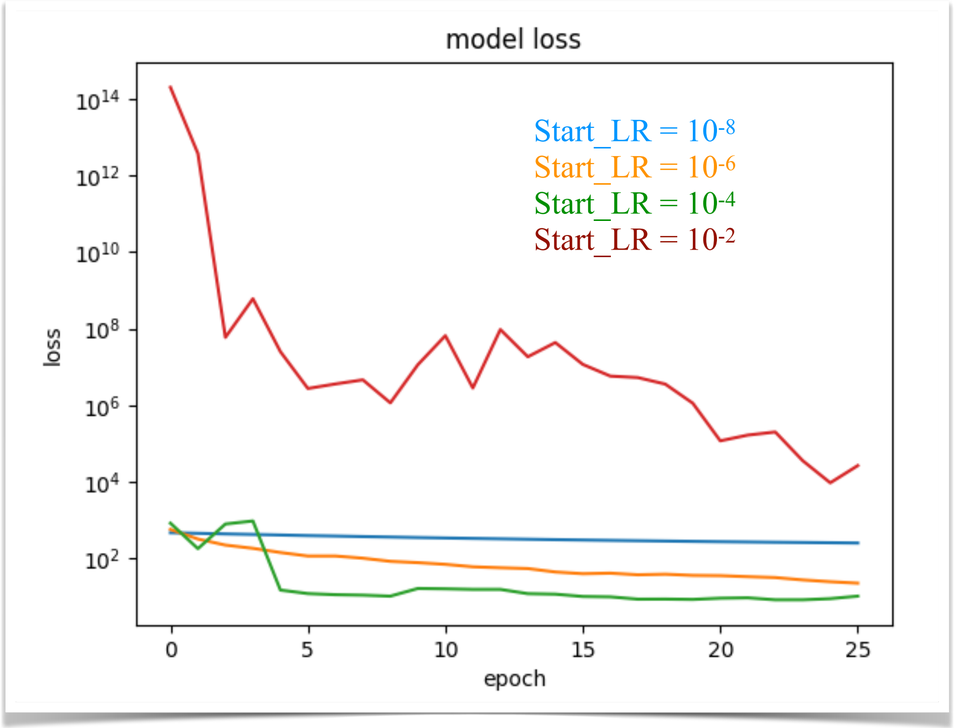
NN training using Tensorflow on GPUs

- 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 (~ 9300 info per event, 100 simulated events used)
  - Maximum hit multiplicity: ~1500 per event
  - 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



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, is used as optimiser to minimise the loss *Refence*
- Testing different START learning rate



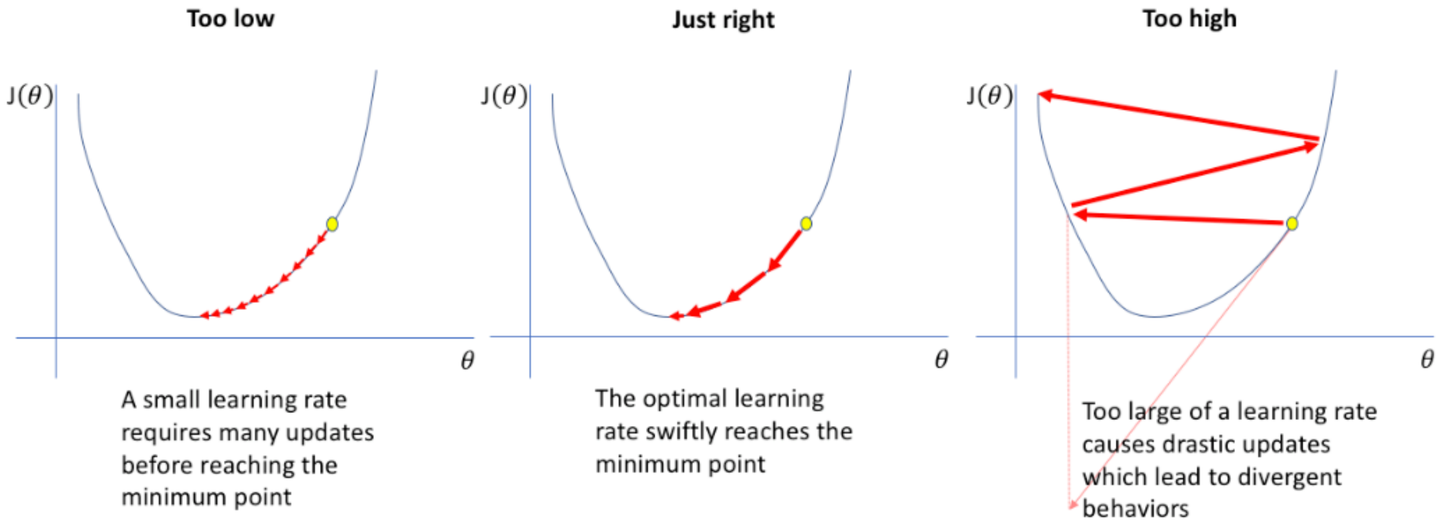
Conclusions and Next steps

- Update NN plugging in angular variables
- 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 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 Slides

Learning rate



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

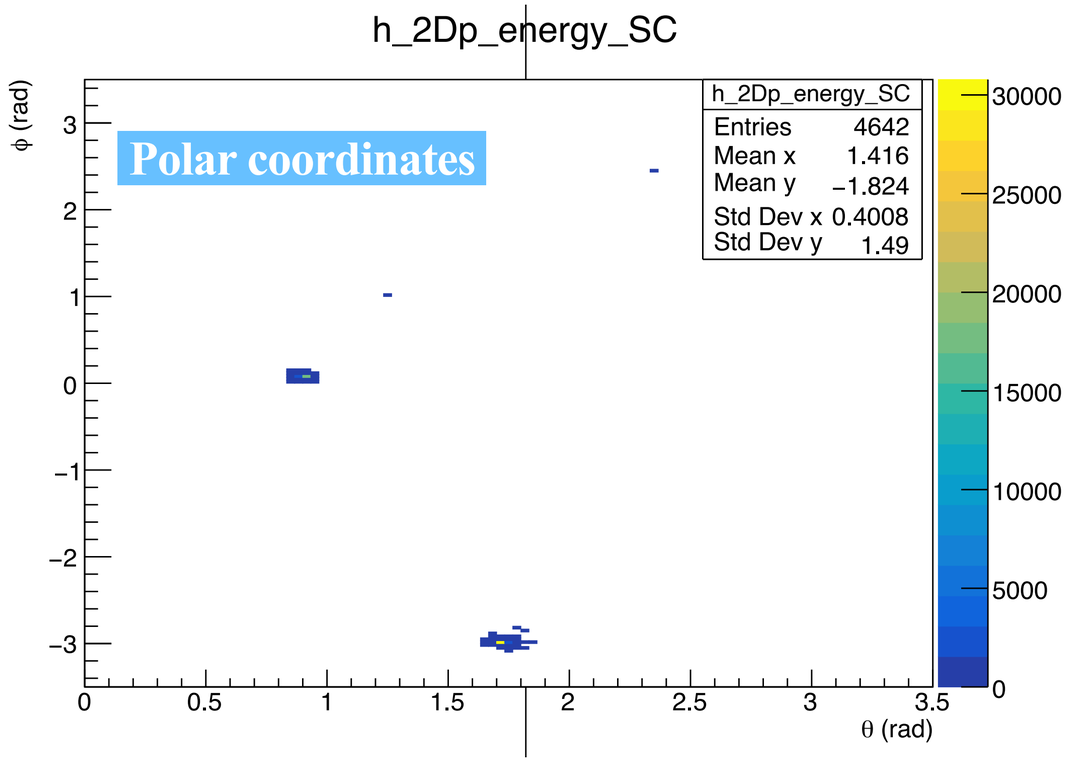
Energy deposits - $e^+e^- \longrightarrow Z(\nu\nu)H(\gamma\gamma)$

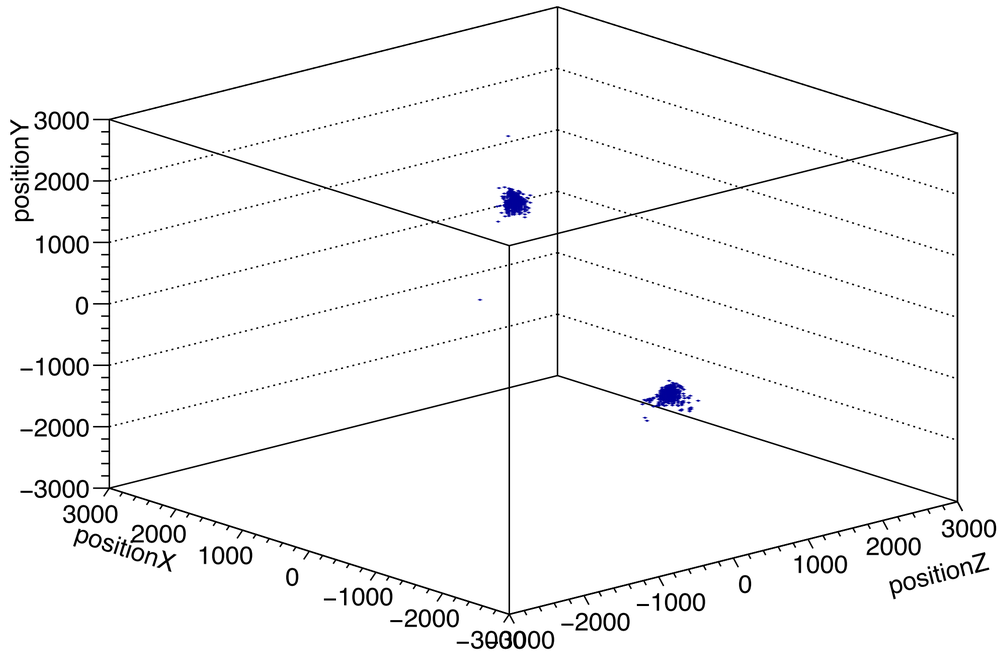
Dumper Algorithm output

Combined fibres

Photon deposits

Input file:
EDMOutput_Higgs.root



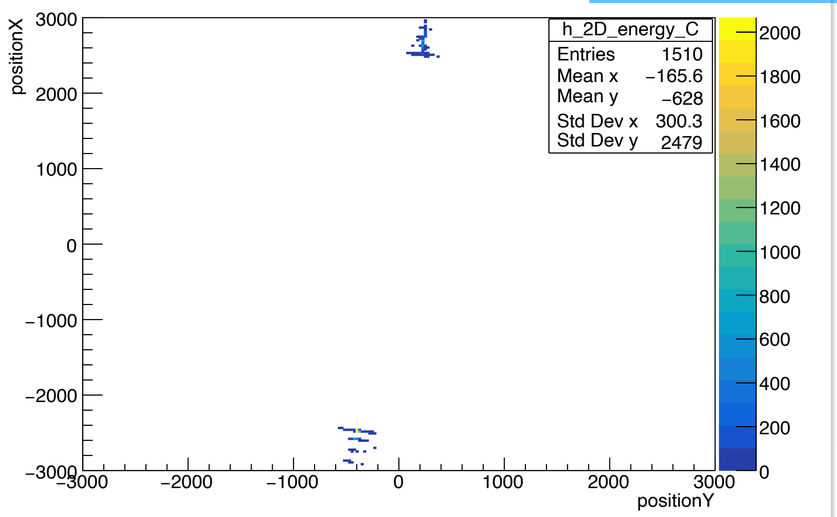


3D plot, cartesian coordinates

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

Dumper Algorithm output

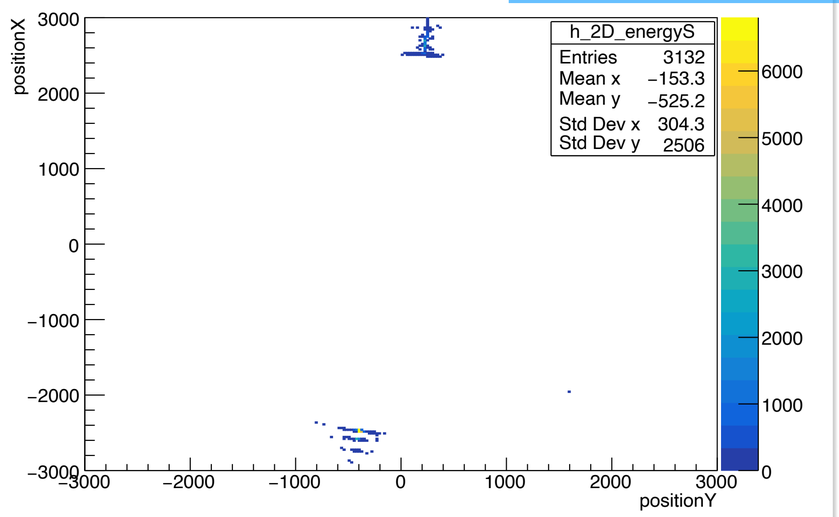
Cherenkov fibres



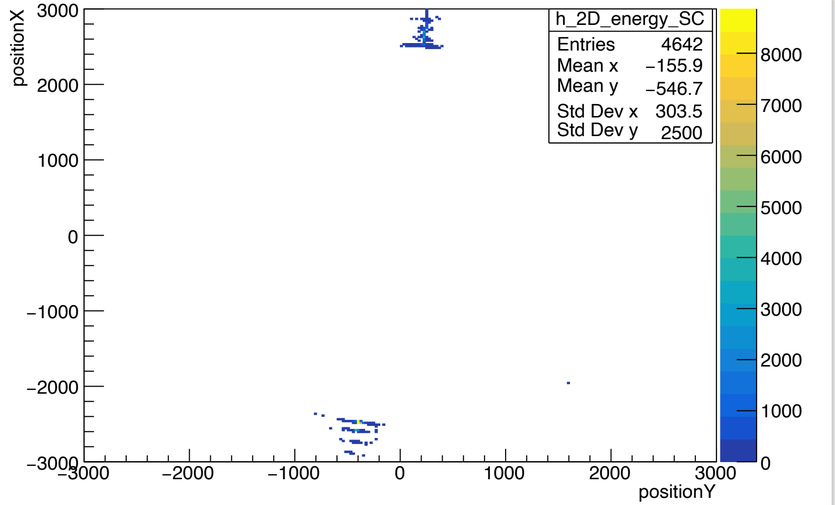
Photon deposits

Input file: EDMOutput_Higgs.root

Scintillator fibres



Combined fibres



ParticleFlow Project - Status Report