HiDRa Simulation Updates

Andrea Pareti - 06/02/2024

π^0/γ Separation

EventDisplay-like plots with 8 fibre grouping binning
(Remember HidraSim is rotated by 90º)
Try to separate π^0 produced 2 metres before calorimeter front face from gammas
Define some variables for a simple discrimination

The method for γ/π^0 separation is mainly based on the broader shape of the calorimeter cluster of a merged π^0 with respect to that for a single photon.

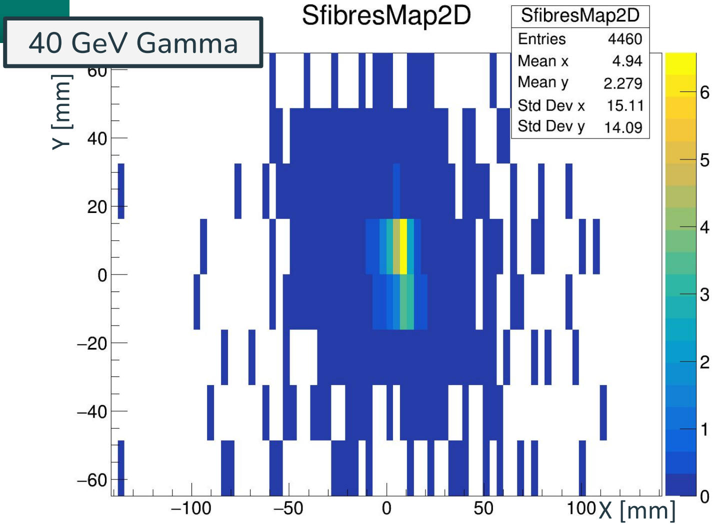
For a given cluster of N cells[1], being (x_i, y_i) the position and e_i the energy of each cell $(i = 1, ..., N$ cells), we define the position spread matrix elements as:

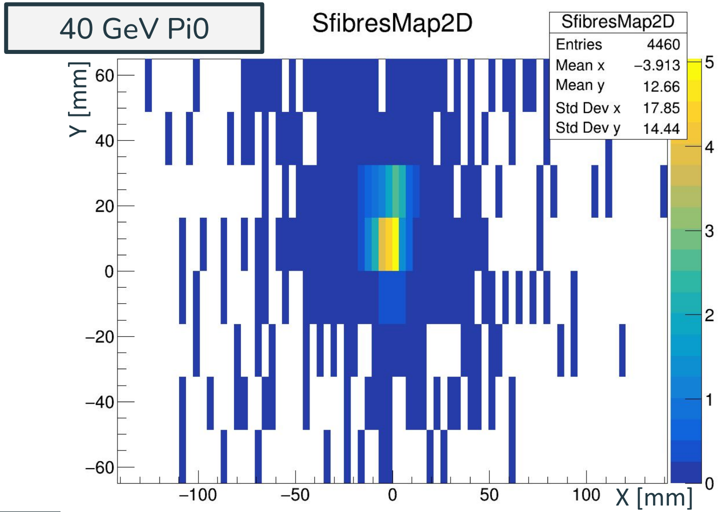
$$S_{XX} = \frac{\sum_{i=1}^{N} e_i (x_i - x_c)^2}{\sum_{i=1}^{N} e_i}, \quad S_{YY} = \frac{\sum_{i=1}^{N} e_i (y_i - y_c)^2}{\sum_{i=1}^{N} e_i}, \tag{1}$$

$$S_{XY} = S_{YX} = \frac{\sum_{i=1}^{N} e_i (x_i - x_c)(y_i - y_c)}{\sum_{i=1}^{N} e_i}, \tag{2}$$

where (x_c, y_c) is the position of the cluster, computed as $x_c = \frac{\sum_i e_i x_i}{\sum_i e_i}$ and $y_c = \frac{\sum_i e_i y_i}{\sum_i e_i}$.

Link





π^0/γ Separation

- The first one is related to the size of the cluster, referred to as $r2$ or shower shape. It is simply the second order momenta:

$$r2 = < r^2 > = S_{XX} + S_{YY} = \frac{\sum_{i=1}^{N} e_i((x_i - x_c)^2 + (y_i - y_c)^2)}{\sum_{i=1}^{N} e_i} \quad (3)$$

- The second variable informs about the importance of the tails, here referred to as $r2r4$:

$$r2r4 = 1 - \frac{< r^2 >^2}{< r^4 >} \quad (4)$$

- The following is known as κ and is related to the ratio of the eigenvalues of the matrix S, which is $(1+\kappa)/(1-\kappa)$. Thus, it relates to the major and minor semiaxes of an ellipse.

$$\kappa = \sqrt{1 - 4\frac{S_{XX}S_{YY} - S_{XY}^2}{(S_{XX} + S_{YY})^2}} = \sqrt{1 - 4\frac{\det S}{\mathrm{Tr}^2 S}} \quad (5)$$

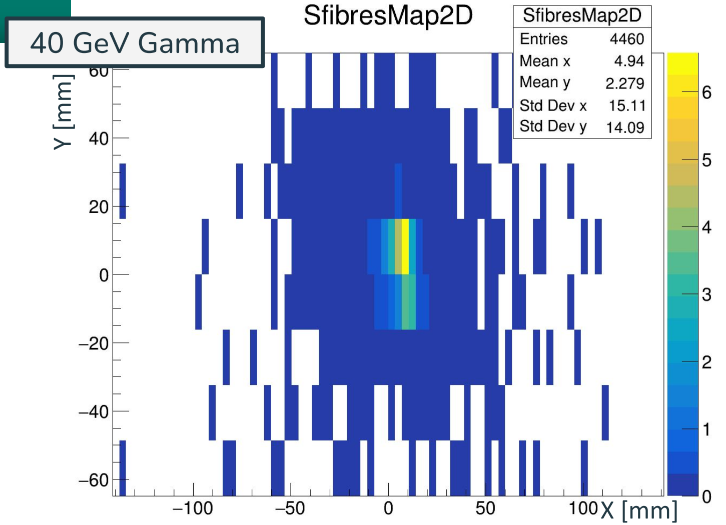
- The last one, referred to as *asym*, provides information about the orientation of the ellipse or correlation between X and Y coordinates.

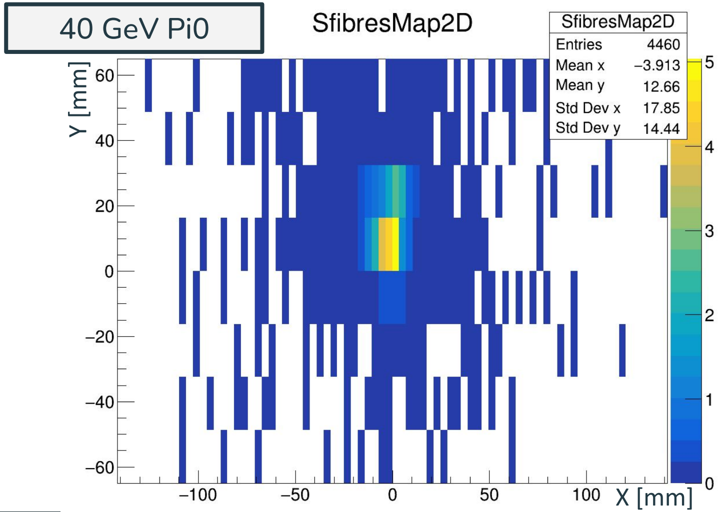
$$\text{asym} = \frac{S_{XY}}{\sqrt{S_{XX}S_{YY}}} \quad (6)$$

The energy of the seed cell (the most energetic cell of the cluster), E_{seed}, and of the second most energetic cell of the cluster, E_{2nd}, are also studied. In particular, the ratios

$$\frac{E_{\text{seed}}}{E_{\text{cl}}} \quad \text{and} \quad \frac{(E_{\text{seed}} + E_{2nd})}{E_{\text{cl}}}$$

are finally used.



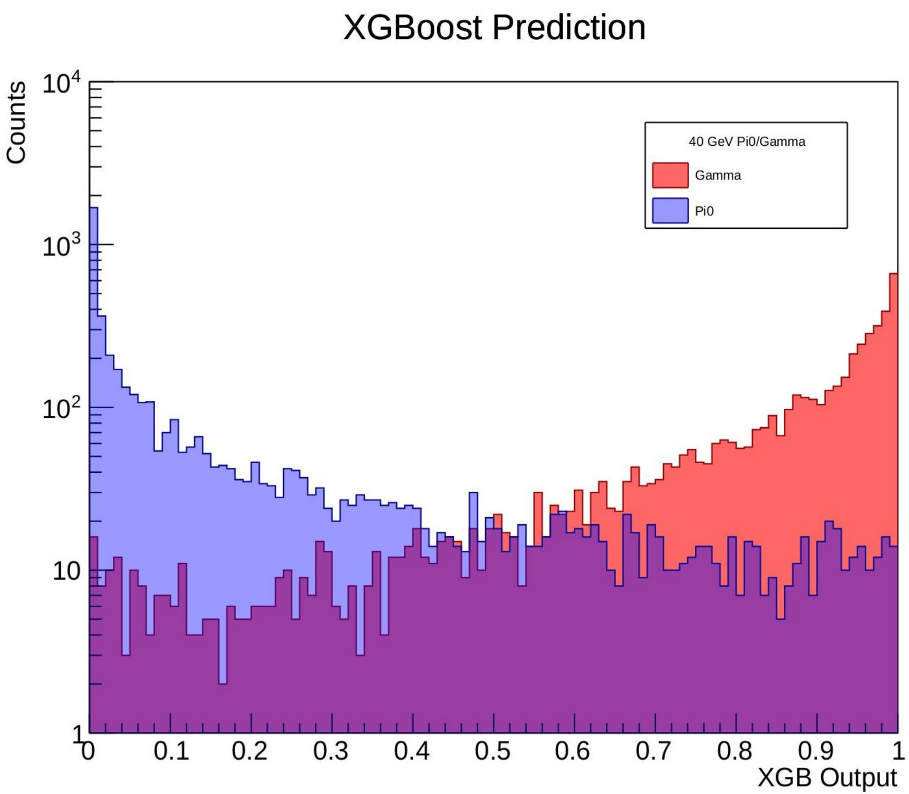


π^0/γ Separation

Boosted Decision Tree trained on ~10k events (5k gammas + 5k pi0)
Output is the probability of belonging to the "gamma" class for each event, given by XGBoost

This is just a very simple test with some generic variables (for the LHCb calorimeter):

- ➢ Can we define some smarter variables, based on the HiDRa geometry?



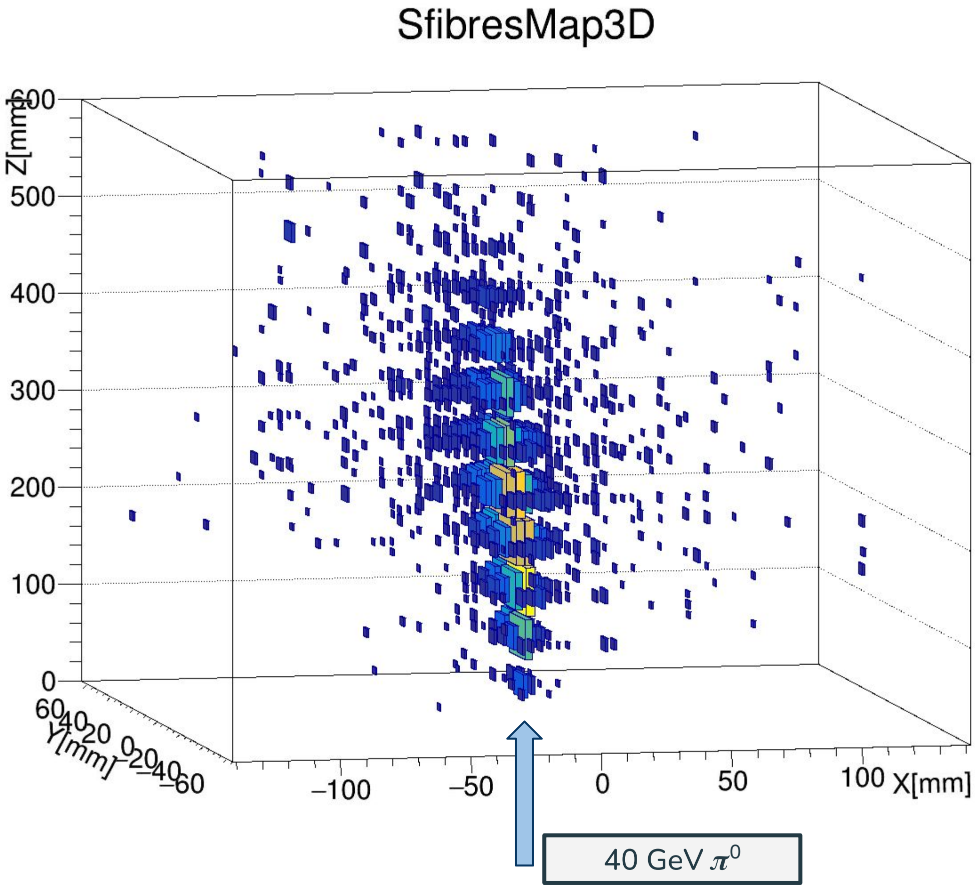
π^0/γ Separation

5 cm binning on Z axis
(box size depends on number of entries in the bin)

Boosted Decision Tree trained on ~10k events (5k gammas + 5k pi0)
Output is the probability of belonging to the "gamma" class for each event, given by XGBoost

This is just a very simple test with some generic variables (for the LHCb calorimeter):

- ➢ Can we define some smarter variables, based on the HiDRa geometry?
- ➢ Try using some CNN/GNN
- ➢ How to use C/S fibres information?



Towards HidraSim++

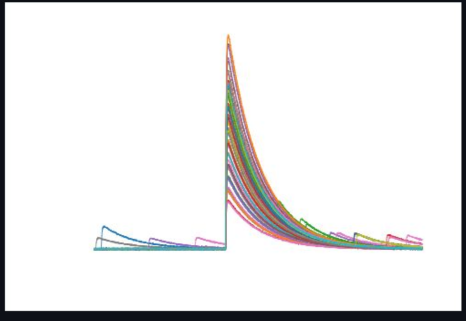
- ★ Started having a look at Edoardo's SiPM simulation
  - Input is the timestamp of optical photons upon each SiPM
  - output is the produced waveform
  - At first I would start using this simulation as a “post-processing” from the already existing HidraSim output
  - Then try to include it in my own HidraSim fork
  - Have to check with Romualdo the SiPM features to input to the sim configuration

Introduction

SimSiPM is a simple and easy to use C++ library providing a set of object-oriented tools with all the functionality needed to describe and simulate Silicon PhotonMultipliers (SiPM) sensors. The main goal of SimSiPM is to include the response of SiPM sensors, along with noise and saturation effects, in the description of a generic detector in order to have a more detailed simulation. It can also be used to perform optimization studies considering different SiPMs models allowing to choose the most suitable product available on the market.

SimSiPM code mostly follows FCCSW rules and guidelines concerning C++. SimSiPM has been developed especially for high-energy physics applications and particle physics experiments, however its flexibility allows to simulate any kind of experiments involving SiPM devices.

SimSiPM does not have any major external dependency making it the perfect candidate to be used in an already existing environment (Geant4 or DD4HEP) or as "stand-alone".



Towards HidraSim++

- ★ Started having a look at Edoardo's SiPM simulation
  - Input is the timestamp of optical photons upon each SiPM
  - output is the produced waveform
  - At first I would start using this simulation as a “post-processing” from the already existing HidraSim output
  - Then try to include it in my own HidraSim fork
  - Have to check with Romualdo the SiPM features to input to the sim configuration

- ★ Import updates from TB2023 simulation:
  - move calorimeter position within macro card
  - Attenuation in fibres
  - …anything else on the shop list?