Implementation of π/γ separation in SPDROOT

Andrei Maltsev (JINR)

SPD Physics & MC Meeting, 25.01.2023

Outline

Motivation

Prompt photons:

- interested in $p_T > 3\text{-}4$ GeV, reject background from π^0

Online polarizability measurement:

- select high-purity π^0 sample

Idea:

- separate clusters produced by a photon and neutral pion, where the two decay photon showers merged into one cluster, based on the shape of the cluster;
- here, cluster means a group of neighboring cells, where in each cell the deposited energy is above threshold.

Input parameters

- X/Y for endcaps or Z/ϕ for barrel, inputs shown in red

Energy distribution

- S_1, M_2 – cells with first and second largest energies
- S_9, S_{25} – sum of energies in 3×3, 5×5 regions around cell with highest energy
- S_6 – maximum energy in 3×2 region containing both first and second largest energy cells
- $\frac{S_1}{S_9}, \frac{S_9-S_1}{S_{25}-S_1}, \frac{M_2+S_1}{S_4}, \frac{S_6}{S_9}, \frac{M_2+S_1}{S_9}, \frac{S_4}{S_{25}}$

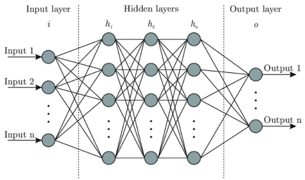
Size/shape

- $|x_{cog}|_{25} = |\frac{\sum_{i=1}^{25} E_i X_i^{rel}}{S_{25}}|$, $|y_{cog}|_{25} = |\frac{\sum_{i=1}^{25} E_i Y_i^{rel}}{S_{25}}|$
- $S_{\alpha\beta} = \frac{\sum_{i=1}^{N} e_i(\alpha_i-\alpha_c)(\beta_i-\beta_c)}{\sum_{i=1}^{N} e_i}$, $\alpha, \beta : X, Y$
- $\rightarrow S_{XX}, S_{YY}, S_{XY}$
- $r2 =< r^2 >= S_{XX} + S_{YY} = \frac{\sum_{i=1}^{N} e_i((x_i-x_c)^2+(y_i-y_c)^2)}{\sum_{i=1}^{N} e_i}$
- $\kappa = \sqrt{1 - 4\frac{S_{XX}S_{YY}-S_{XY}^2}{(S_{XX}+S_{YY})^2}} = \sqrt{1 - 4\frac{\det S}{\mathrm{Tr}^2 S}}$
- Angle θ of incidence

Topology of the network

Attempt at using a more complex NN

Inspired by the work of Dimitrije Maletic (thanks!) and https://cds.cern.ch/record/2042173



$O_i = f(W_{i0} + \sum_{j=1}^{N} W_{ij} O_j) \rightarrow$ weighted sum + bias for each node

- f: **ReLU** $f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$ sigmoid for output: $f(x) = \frac{1}{1+e^{-x}}$

- **Dropout** (p=0.1),
- **batchnorm** for each layer (before activation)

- Binary cross entropy loss **(BCE)**:

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- Optimizer: **Adam**
(stochastic gradient descent +
adaptive moment estimation)
(lr = 0.001, β_1 = 0.9, β_2 = 0.999, ε=1e-8

2 hidden layers, 64 neurons each

SpdEcalRCParticle

- SpdEcalRCParticle class $\rightarrow$ fPID field: 22 (γ)/111 (π^0)

SpdEcalPiGammaSeparator

- idea: create a separate class which contains:
  - function to predict particle ID from pre-calculated input parameters (implementation of the NN)
  - array of parameters of the NN as static members of the class
  - interface that accepts SpdEcalRCCluster
- also needs some geometry parameters as input (number of cells in barrel/endcap)
- single instance is stored in SpdEcalRCMaker (recontruction algorithm)

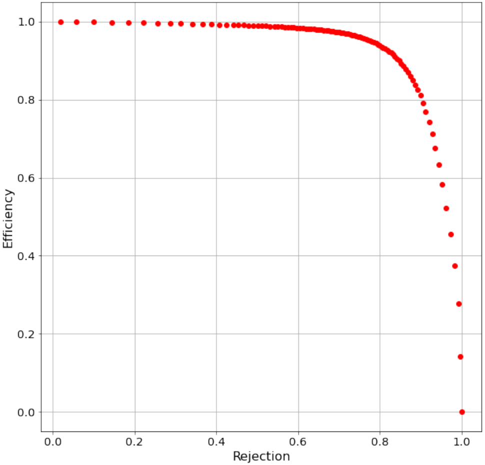
SpdEcalRCMaker

Class for cluster reconstruction is mostly unchanged, except:

- initializes SpdEcalPiGammaSeparator in the Init() step
- added one more function to predict particle ID (γ / π^0)

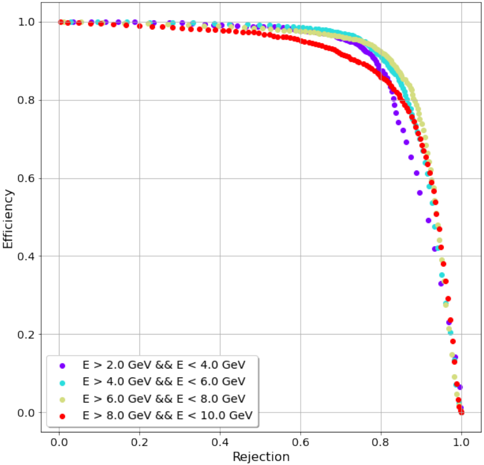
Efficiency and rejection

All angles and energies



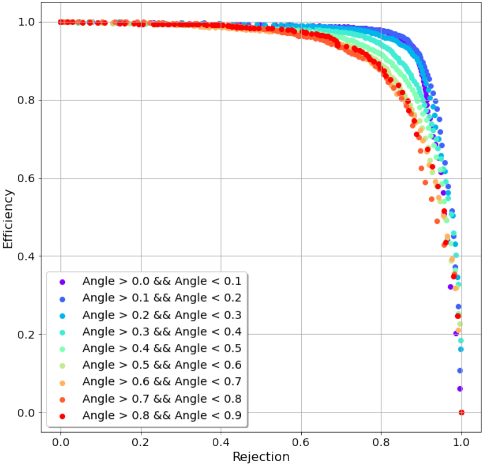
Efficiency and rejection

Efficiency/rejection for different energies



Efficiency and rejection

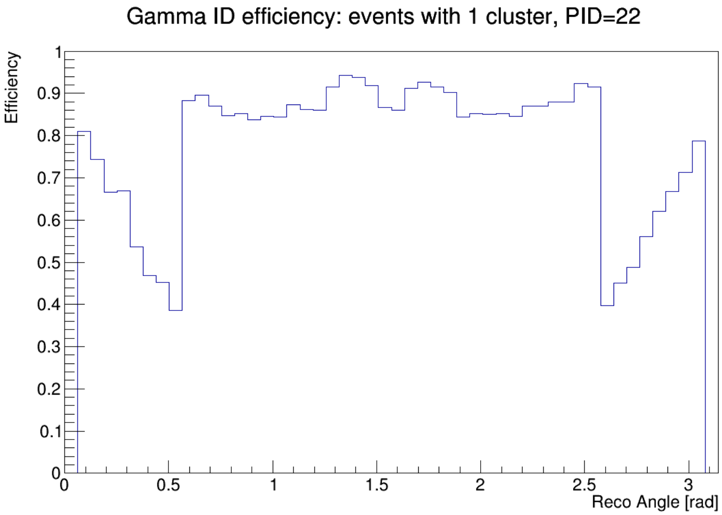
Efficiency/rejection for different angles



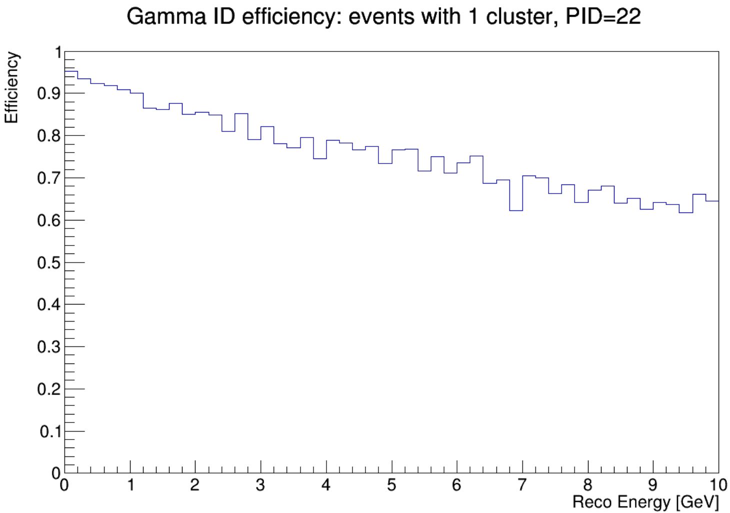
Test conditions

- π^0/γ generated isotropically;
- energies of particles from 0 to 10 GeV (discrete distribution);
- cell energy threshold is 20 MeV;
- identified as a photon if both conditions are true:
  - only one cluster in event;
  - the cluster is tagged as a photon;
- identified as a pion if either:
  - two clusters in event;
  - one cluster, tagged as a pion.

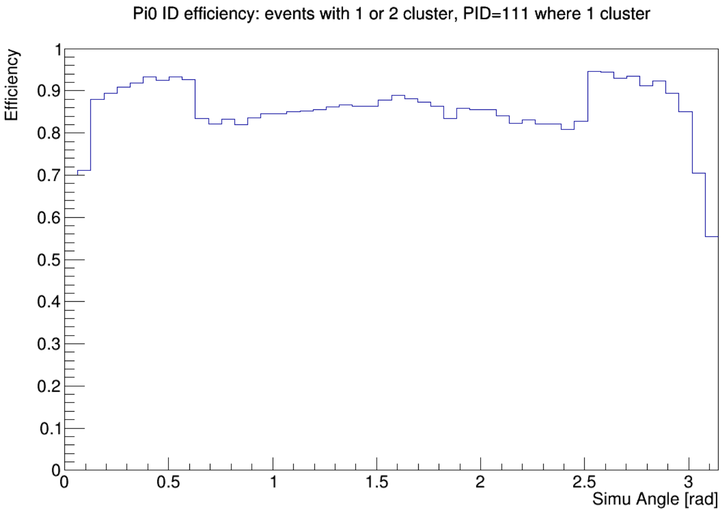
γ identification



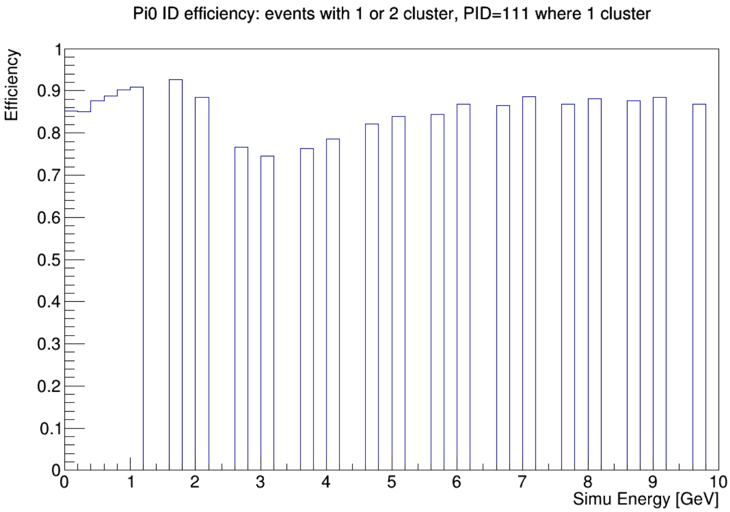
γ identification



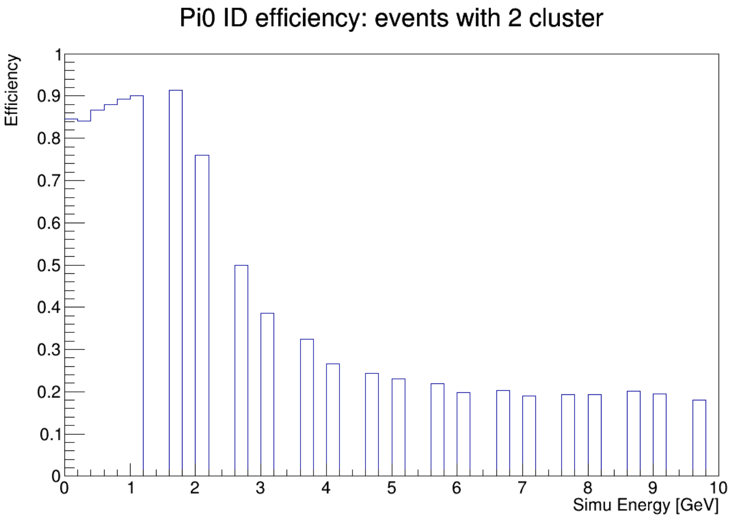
π^0 identification



π^0 identification



π^0 identification



Conclusions

- After being implemented into SPDROOT, the separation network gives 70-90% photon detection efficiency (depending on energy) and 80-90% pion detection efficiency
- Bad performance in the vicinity of the gap between barrel and endcap

Future steps

- Add the macros to produce pi/gamma database and train the network to git
- Bridge clusters in barrel and endcap
- Investigate the possibility of using R-CNN for cluster reconstruction