Neural networks and reconstruction in MC

Andrei Maltsev, JINR

SPD Collaboration Meeting
13.12.2021

Plan

1. ECAL simulation and reconstruction workflow
2. Particle reconstruction in ECAL: current status and outlook
3. π/γ separation in ECAL using neural networks

ECAL simulation workflow

1) Simulation (SPDROOT)

Geant4 simulation for particle propagation through matter (could be changed without changing the workflow)

input: geometry parameters (cell size, ECAL radius etc.)

output: points (energy depositions) in ECAL

→

2) Reconstruction (SPDROOT)

Hit making, track/cluster finding, particle and event reconstruction

input: points in ECAL

output: reconstructed particles in ECAL: energies, positions, etc.

→

3) Analysis (ROOT)

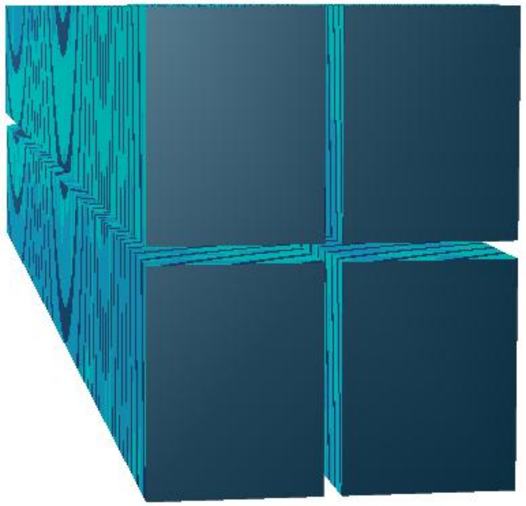
Statistical analysis of many events for physical results, calibration, debugging etc.

input: event-per-event information on reconstructed particles

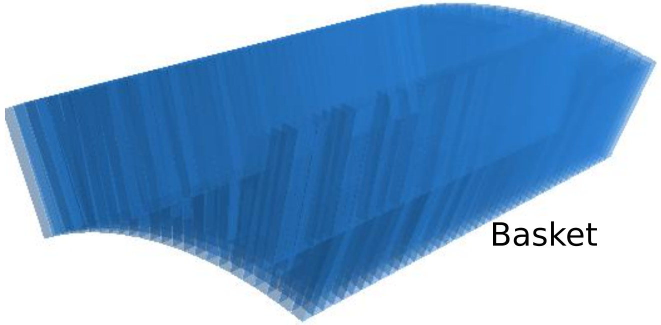
output: user-defined histograms, graphs, etc.

1) Simulation step

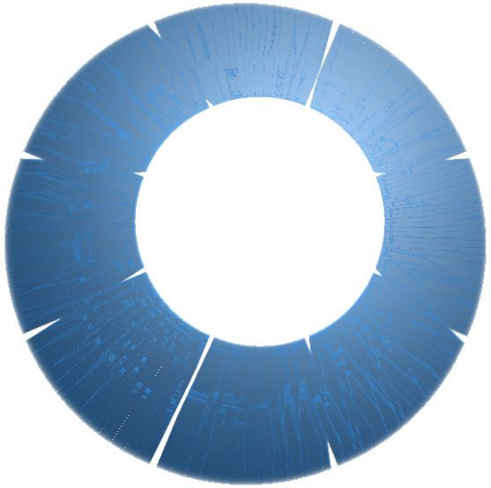
- **Geometry hierarchy:**
  - barrel part: cells → modules (2×2 cells) → baskets (Z/φ slices) → barrel
  - endcaps: cells → modules (2×2 cells) → endcap
- **Modifiable geometry:**
  - number of sectors (baskets in φ, Z)
  - gaps between sectors/baskets/modules/cells
  - cell sizes
  - number of layers, material thickness



Module = 2×2 cells
(gaps artificially enlargened)



Basket



Barrel
(gaps artificially enlargened)

2) Reconstruction step

Consists of two steps:

1) clustering
2) reconstruction of particles based on clusters

- Different interchangeable options for reconstruction step (linear weighting, log.weighting, multi-shower fit, etc.)
- Steps are performed separately for ECAL and barrel
- External parameters: clustering distance, cell energy threshold

Division to clustering and reconstruction is not final:

- plans to apply CNN also at the level of reconstruction
- suboptimal bridging of particles in barrel/endcap

3) Analysis step

Using event-by-event output with reconstructed particles and their parameters:

- reconstruction position, energy
- “MC truth”: which MC particles contributed to the reconstructed particle, how much energy each of them contributed

Next release: charged track association

Particle reconstruction algorithms

<u>Currently implemented algorithm:</u>

- Initial values of position: log.weighting
- Empirical functions as interpolation over grid points;
- Values in the grid points obtained via dedicated MC

$$x_c = \frac{\Sigma_i W_i(E_i) x_i}{\Sigma_i W_i(E_i)} \qquad W_i^{(linear)}(E_i) = E_i,$$

$$W_i^{(log)}(E_i) = Max\{0, a_0 + ln(E_i) - ln(E_{total})\}.$$

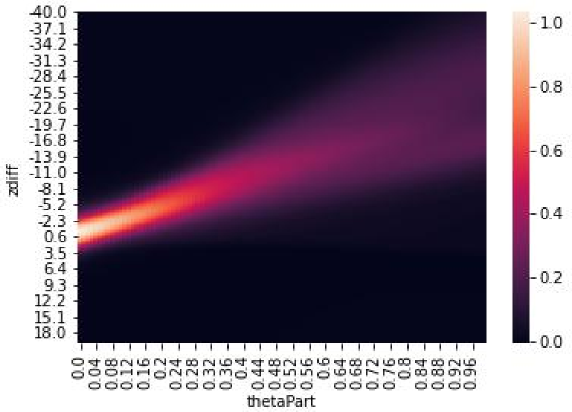
Advantages:
angle is implicitly taken into account via the empirical corrections → rough reconstruction of angled showers
easy to understand and debug

Drawbacks:
have to change the empirical parameters each time we change the geometry (cell size, module thickness)
each cluster corresponds to one particle → bad efficiency for high-energy π0 (prompt photon background)

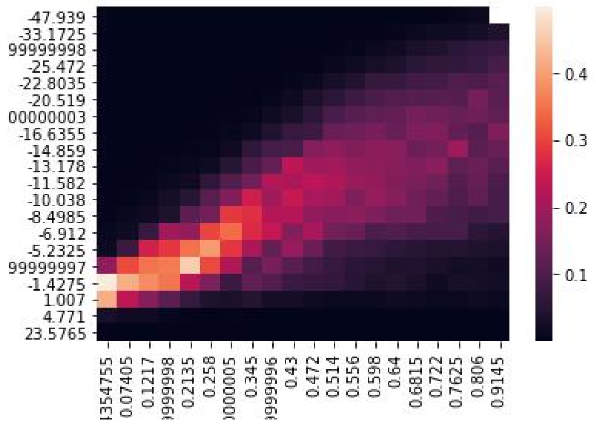
Particle reconstruction algorithms

Alternative algorithm:

- each cluster → multiple particles
- particle parameters (position, energy) taken from fit with shower shapes
- shower shapes have to dependent on the incidence angle
- latest attempt: predict shower shape for different angles based on output of neural network, trained on MC samples



NN output



MC input

Advantages:

- higher π0 reconstruction efficiency,

especially for high angles

Drawbacks:

- large computation time
- fit may be sensitive to initial variables

→ one needs to test the fit convergence and quality

- need analytical shower shape or additional NN for higher angles

Particle reconstruction algorithms

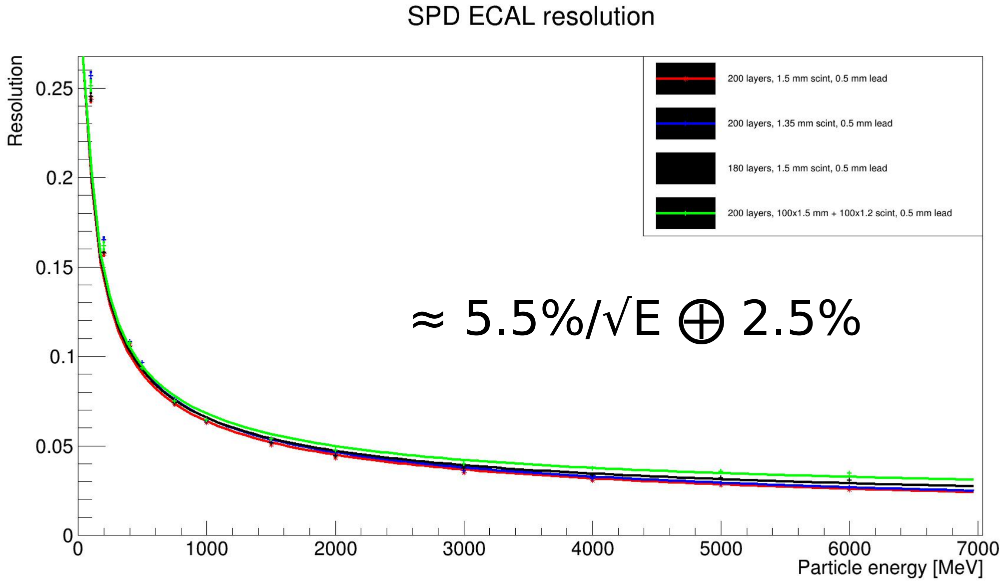
Convolutional neural network (see Dimitrije's talk):

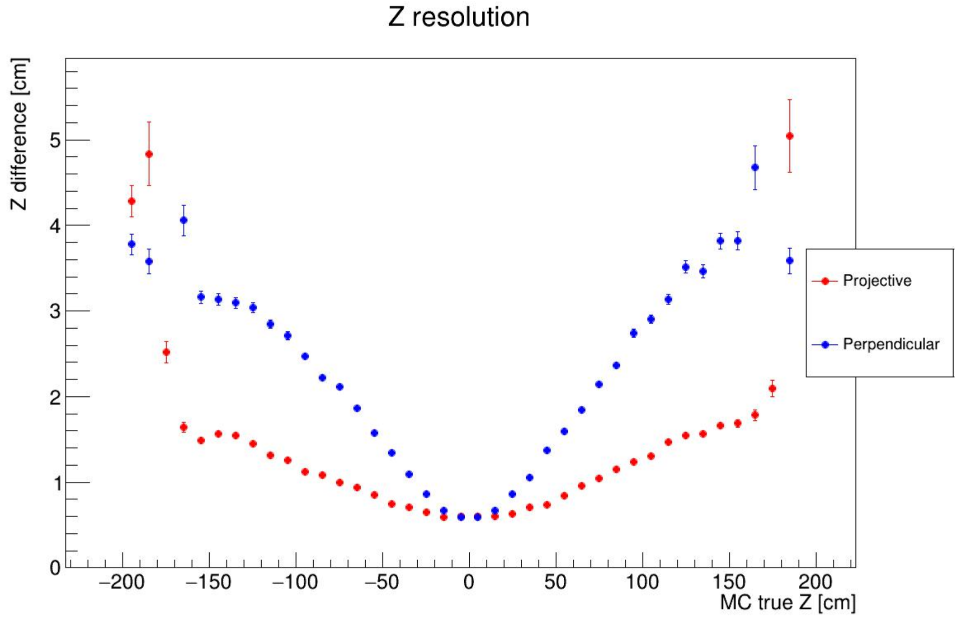
- inputs: cell energies or time slices (ADC counts)
- outputs: particle energies, positions, types ($e^+/e^-/\gamma$/hadron/μ)

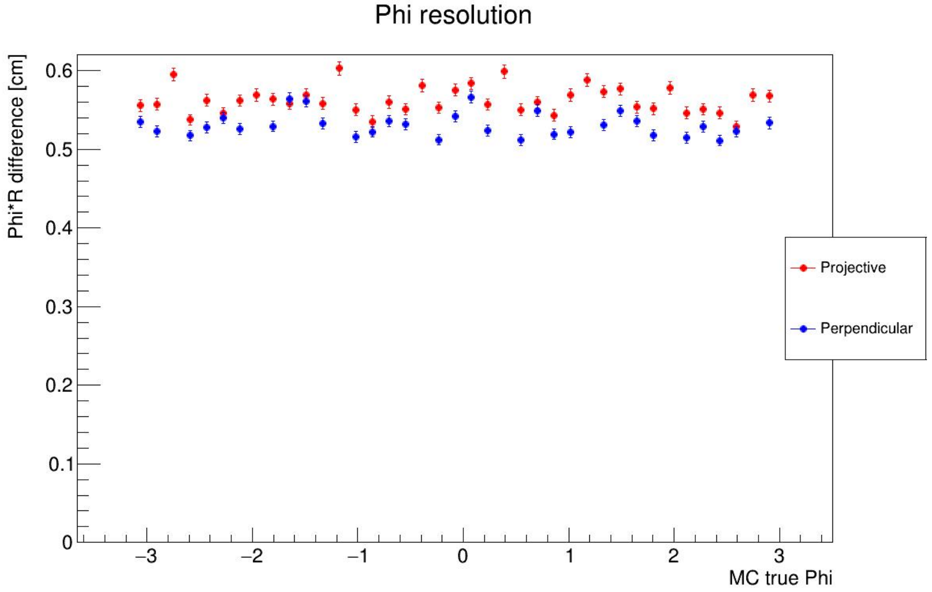
Advantages:
fast

Drawbacks:
need to control the stability in real data conditions

Performance of the current algorithm







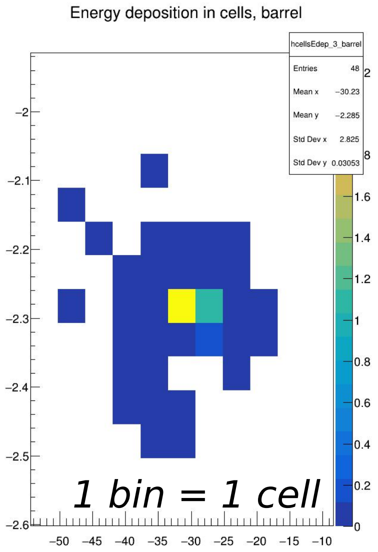
For high angles,
there is contribution not only from transverse fluctuations,
but also from longitudinal → resolution worse than
cell size/sqrt(12)

*Projective geometry implemented in [forked SPDROOT repository]

π/γ separation in ECAL

General idea:

Select 5×5 region of cells in ECAL



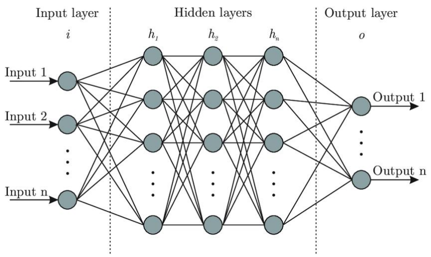
Form characteristic input variables

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

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

...

Use variables as input to a multilayer perceptron (MLP)



One output: π⁰/γ flag

*Not yet implemented into SPDROOT

Inputs

θ/φ moments:

$$|x_{cog}|_{25} = \left| \frac{\sum_{i=1}^{25} E_i X_i^{rel}}{S_{25}} \right|$$

$$|y_{cog}|_{25} = \left| \frac{\sum_{i=1}^{25} E_i Y_i^{rel}}{S_{25}} \right|$$

Correlation:

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

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

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

Importance of tails:

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

Shape variable:

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

Energy distribution

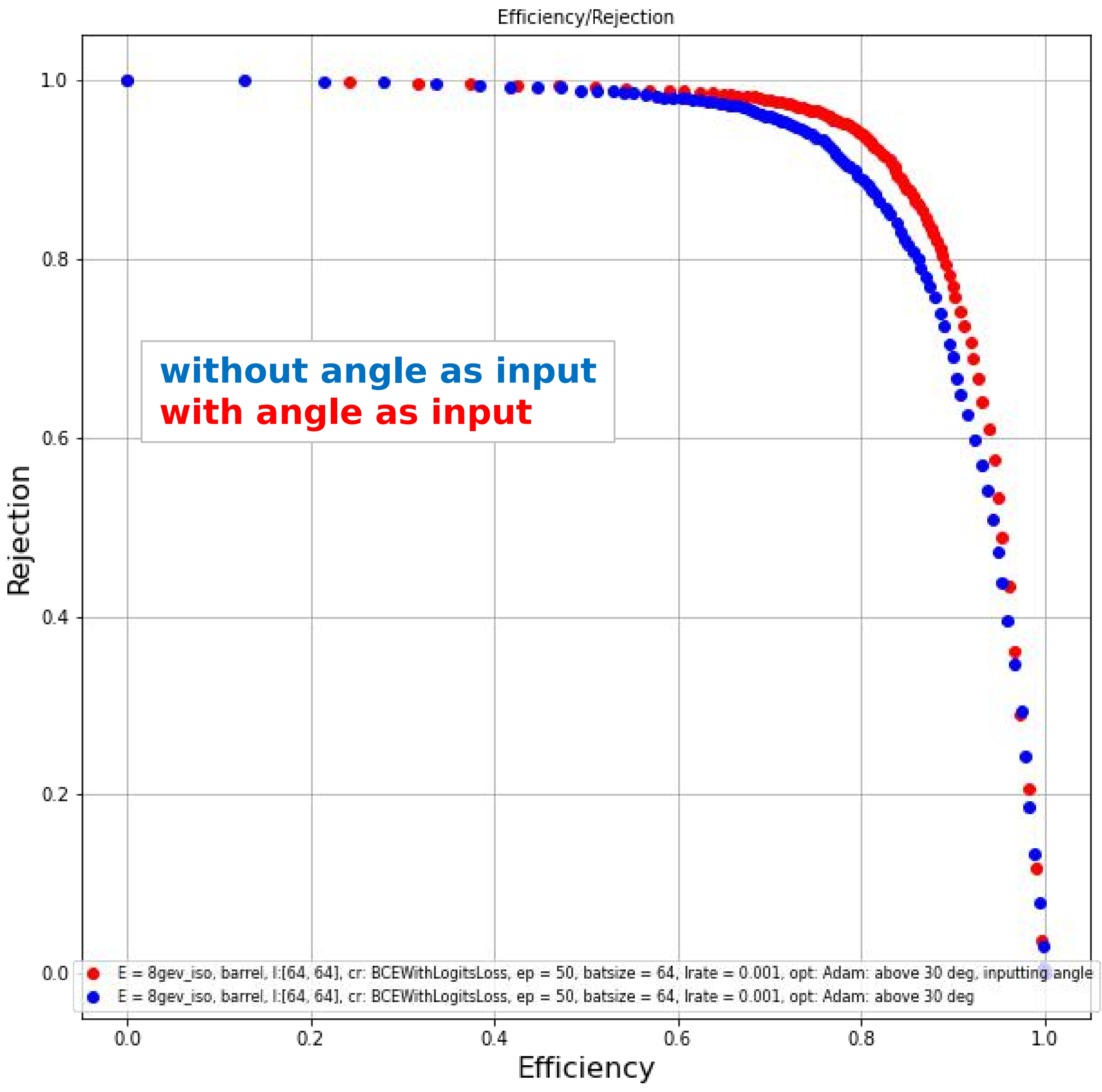
$$\frac{S_1}{S_9} \qquad \frac{S_9 - S_1}{S_{25} - S_1} \qquad \frac{M_2 + S_1}{S_4} \qquad \frac{S_6}{S_9} \qquad \frac{M_2 + S_1}{S_9}$$

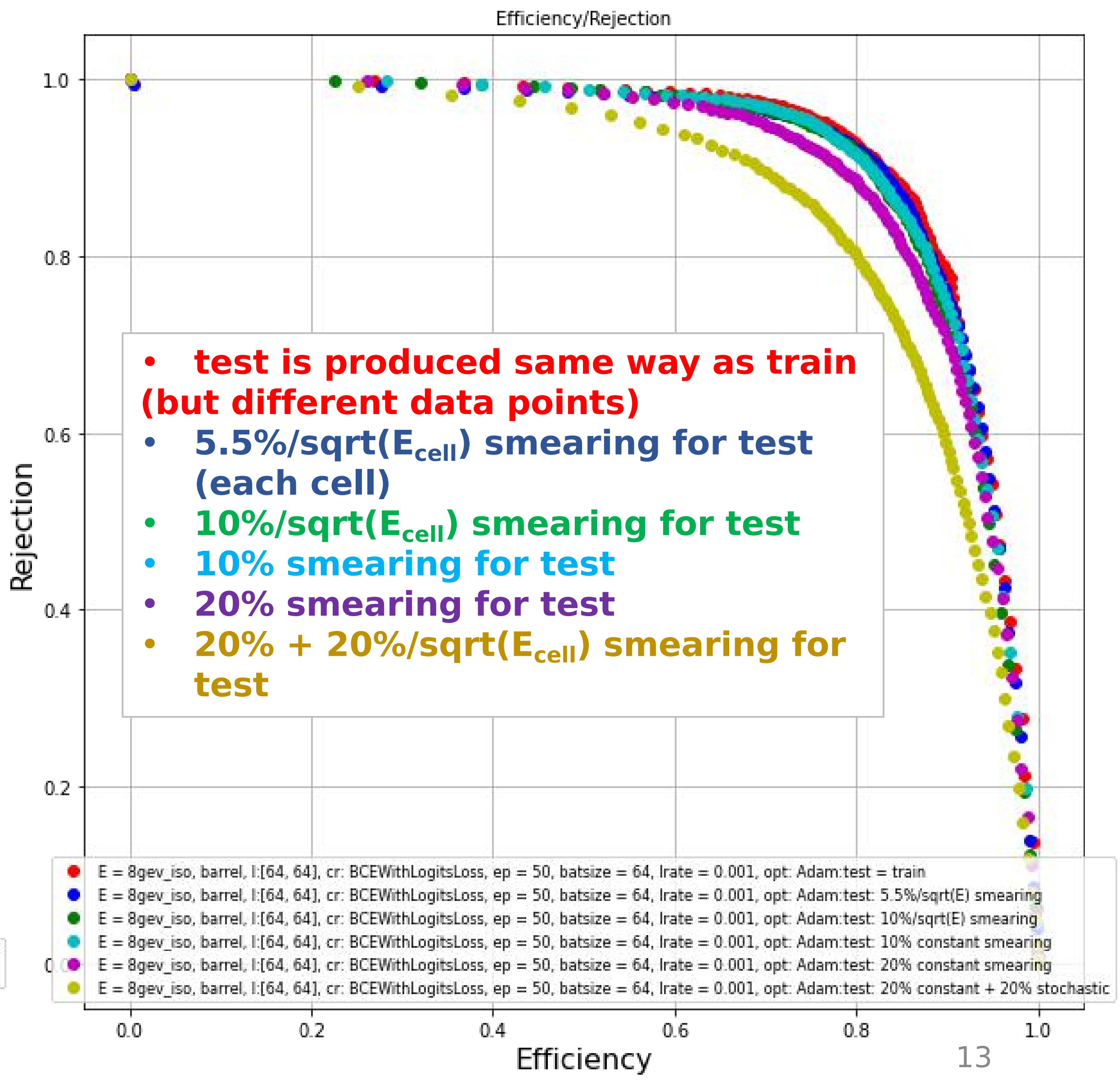
- Angle θ as an input variable
(improves separation at high energies)
- Total energy

X,Y ~ θ,φ
S_1, M_2 - 1st and 2nd largest energies
S_9, S_{25},S_6 - energy in 3×3, 5×5, 3×3 region

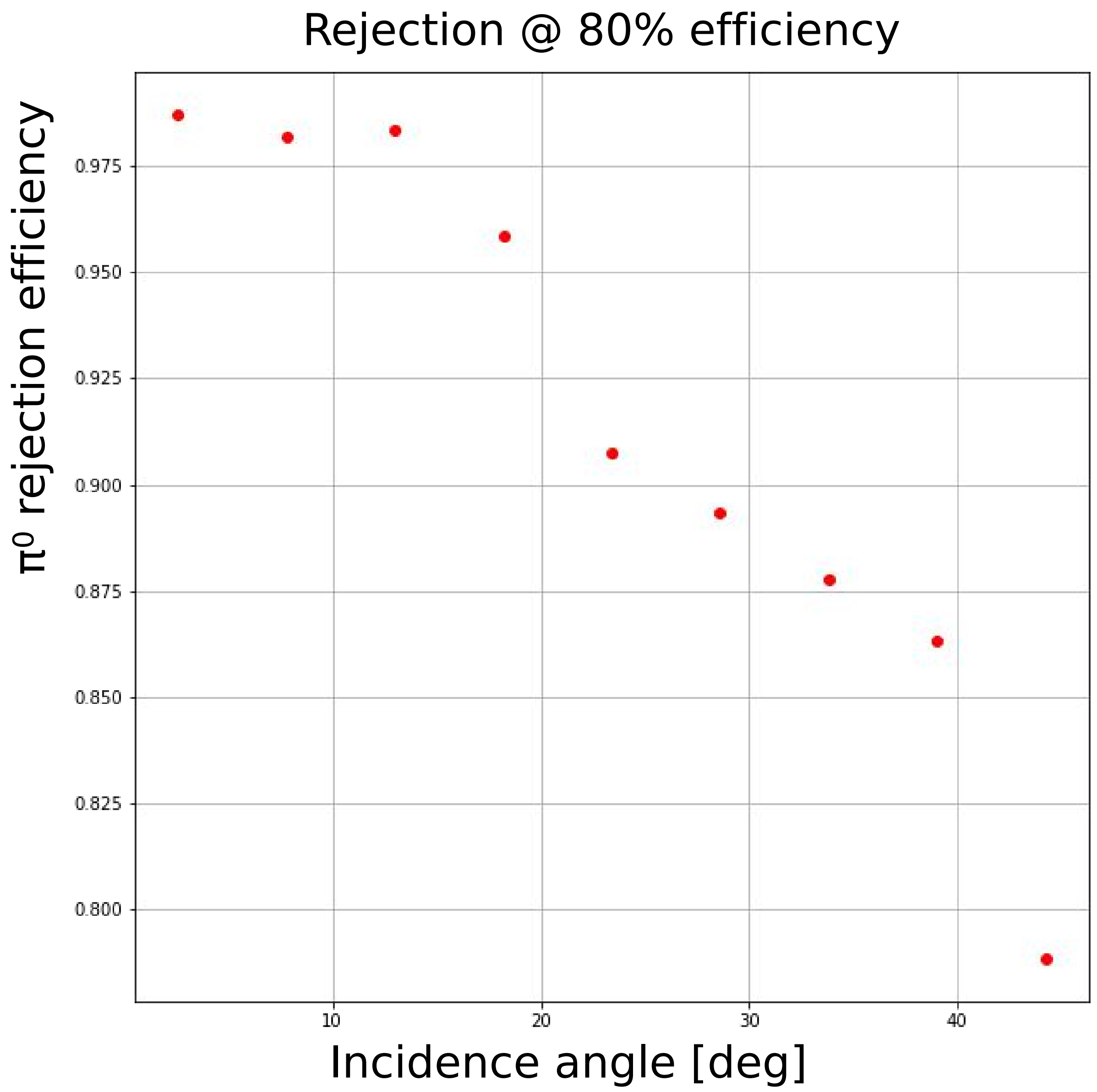
Network parameters:
- 2 hidden layers × 64 neurons, ReLU activation
- output normalized to [0,1] using sigmoid
- dropout (p=0.1), batchnorm
- BCE loss: $H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$
- Optimizer: **Adam**
(lr = 0.001, β_1 = 0.9, β_2 = 0.999, ε=1e-8

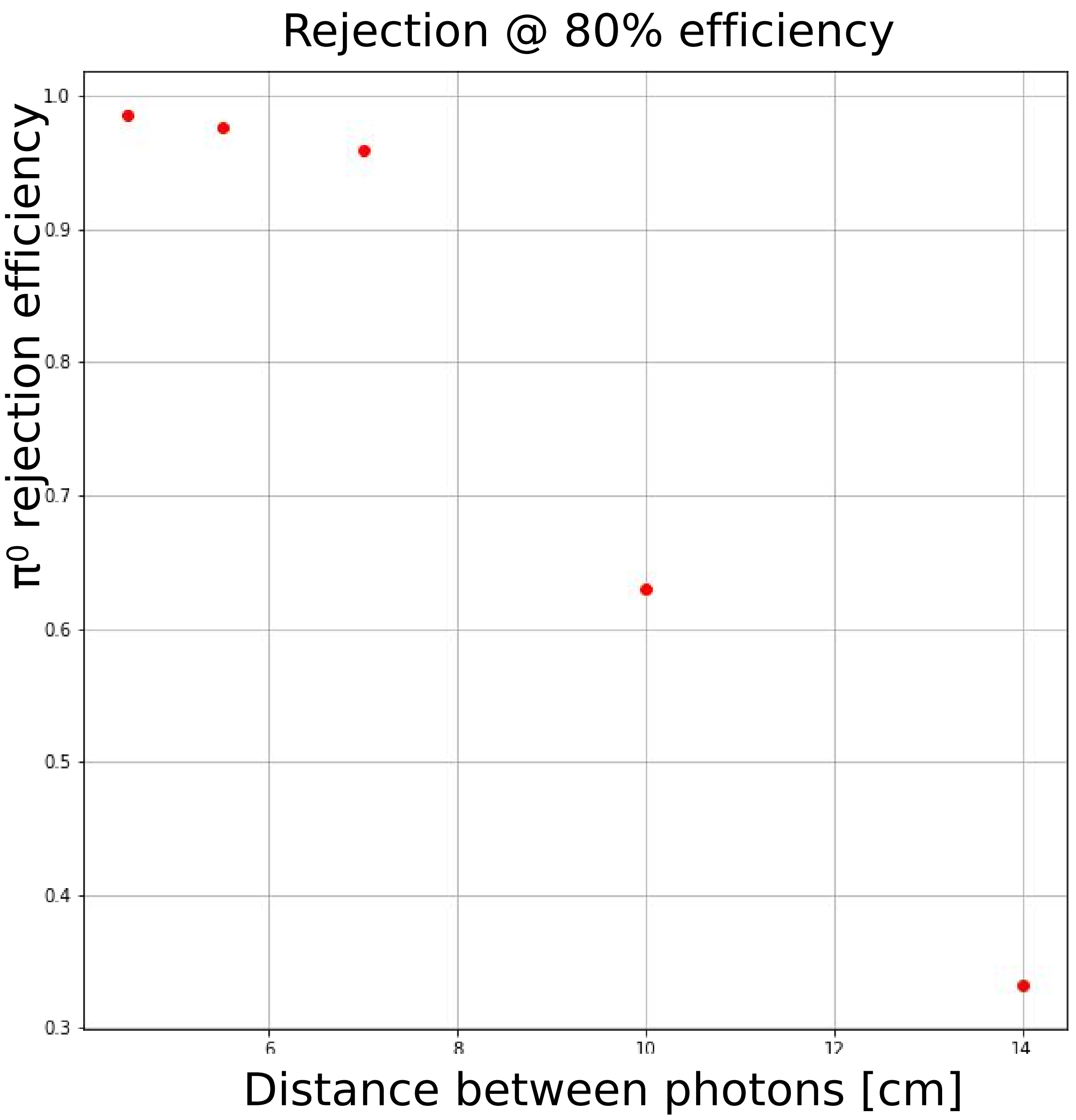
Performance of π/γ separation algorithm





Performance of π/γ separation algorithm





Conclusions and outlook

- Presently implemented particle reconstruction algorithm produces reasonable results in terms of energy and position resolution, but could be significantly improved in terms of speed and efficiency of π/γ separation
- With latest setup, one can expect about 90% π^0 rejection efficiency while selecting 80% of photons

Algorithms to be developed and implemented:

- multi-shower fit to reconstruct particles (to cross-check fast CNN reconstruction)
- CNN applicability to perform fast ECAL reconstruction
- π/γ separation and its integration into workflow, ideally into the CNN fast reconstruction