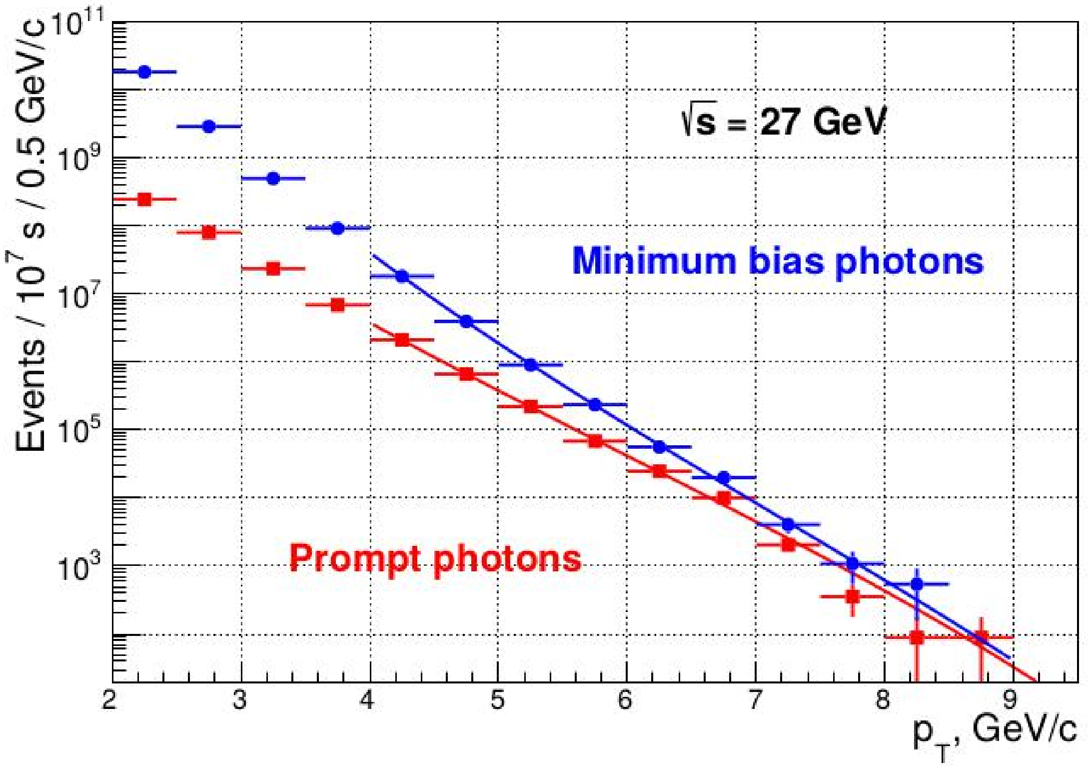
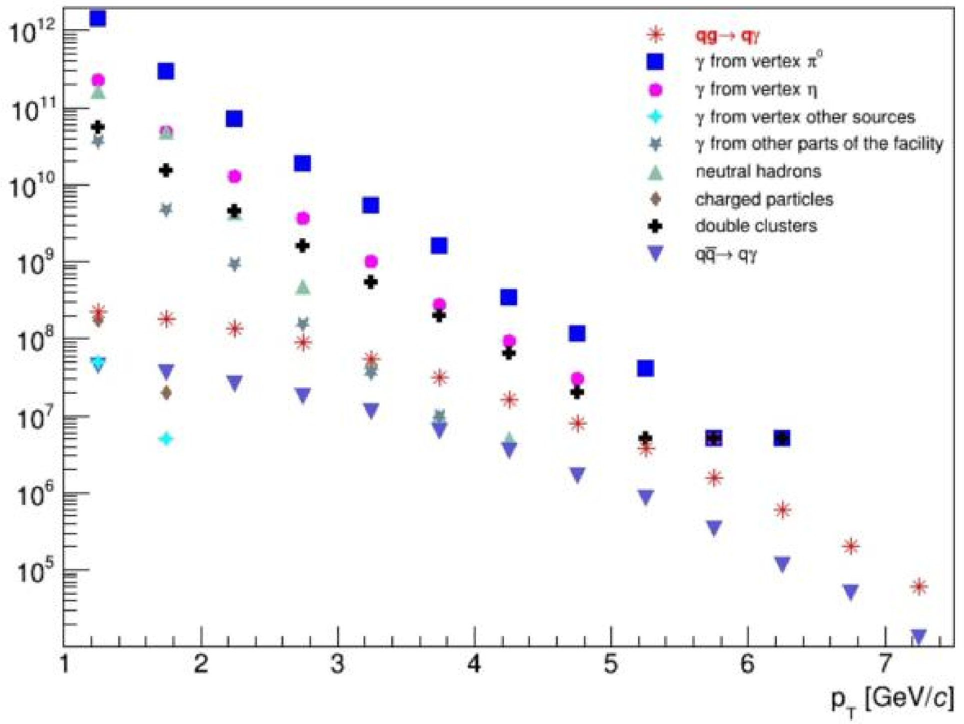
π^0/γ separation in SPD ECAL using machine learning approach

Andrei Maltsev
JINR, Dubna

SPD Physics & MC meeting
08.09.2021

Measurements with prompt photons





$$N_{prompt} = N_\gamma - k \times N_{\pi^0}$$

k: ratio of undetected $\pi^0/\eta/\ldots$ decays (from MC) giving “fake” propmpt photons

Important to have good π^0/γ separation:

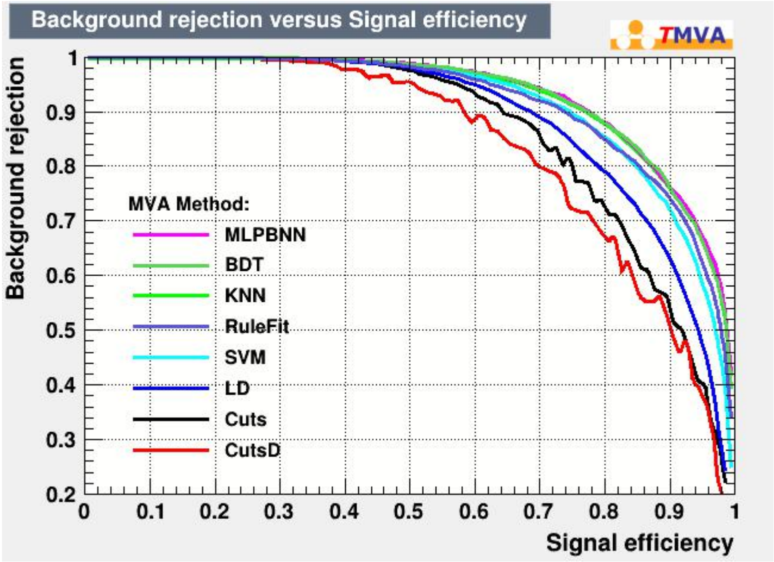
- fewer undetected π^0s: less uncertainty on k
- most importantly: larger N_γ (errors dominated by statistics at high energies!)

Previous results

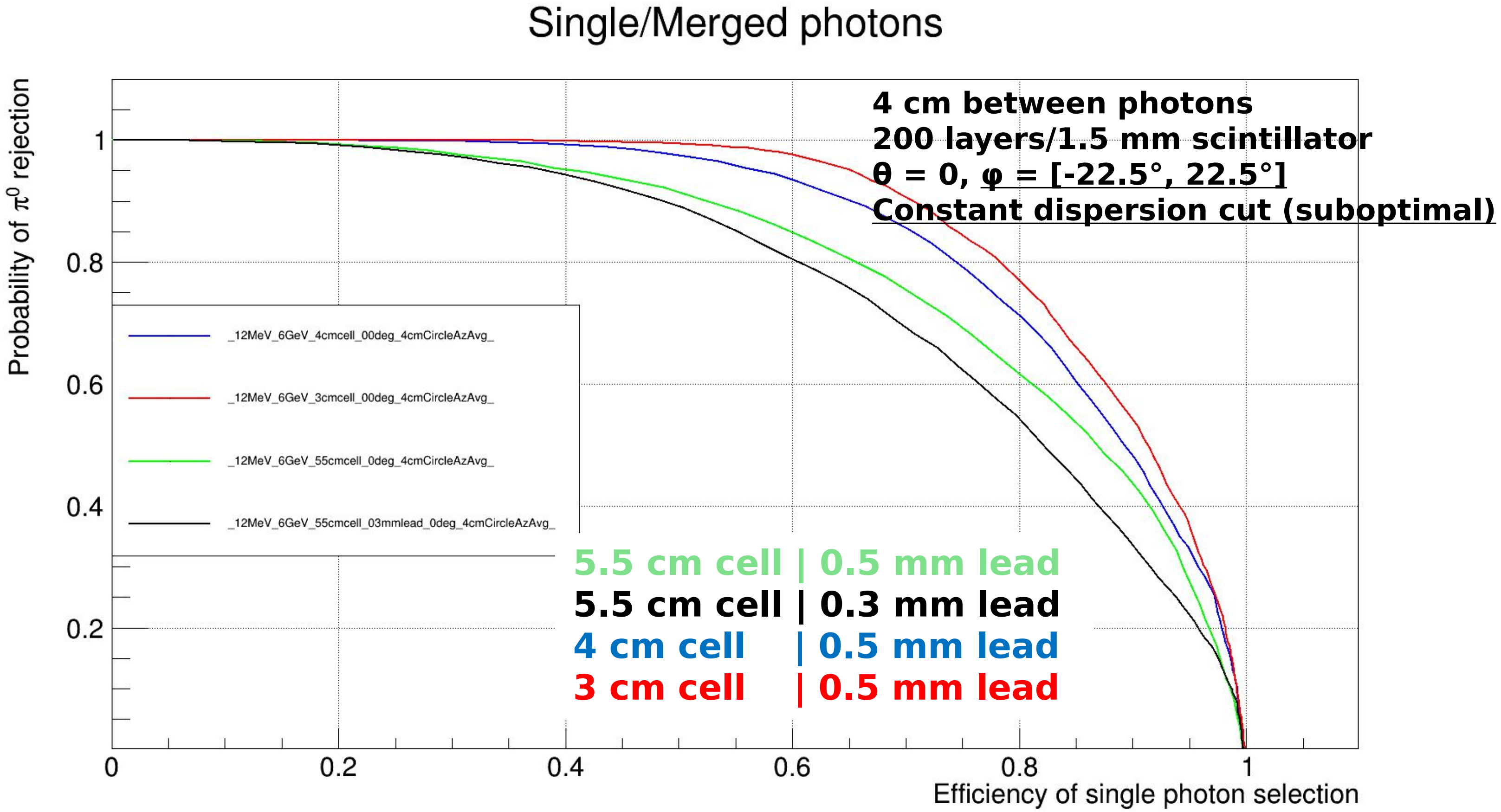
SPD Physics & MC meeting

07.10.2020

φ = [-22.5°,22.5°], θ = 0,
Four parameters: dispersion, r2r4, κ, asym



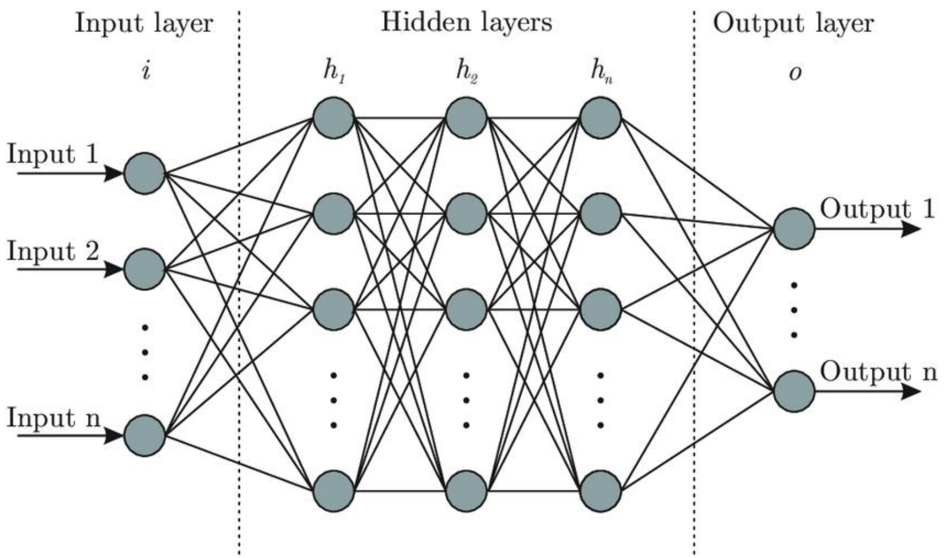
“ideal” conditions (θ = 0°), 4 cm between photons, E_γ = 3 GeV



cell size ~ distance between photons → 80-90% rejection @ 80% efficiency
Can we improve it?

Attempt at using a more complex NN

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



$O_i = f(W_{i0} + \sum_{j=1}^{N} W_{ij} O_j)$ → 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

Inputs

Variables describing moments

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

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

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

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

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

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

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

Energy distribution

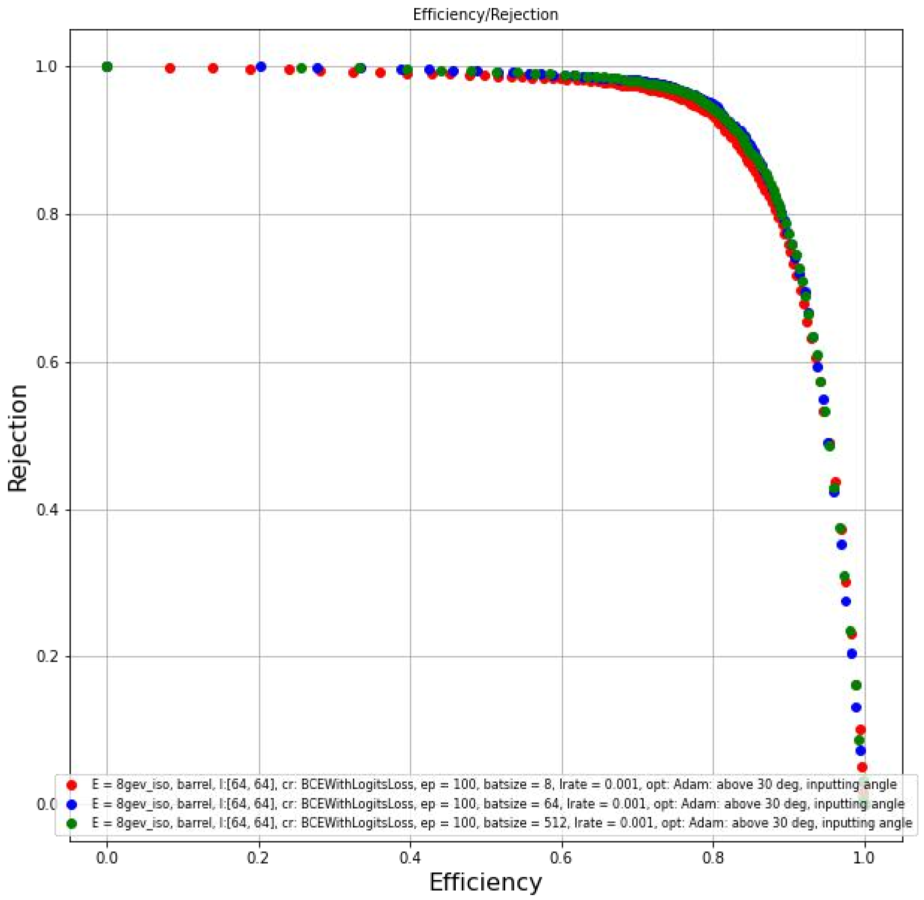
$$\frac{S_1}{S_9} \qquad \frac{S_9 - S_1}{S_{25} - S_1} \qquad \frac{M_2 + S_1}{S_4} \qquad \frac{S_6}{S_9} \qquad \frac{M_2 + S_1}{S_9}$$

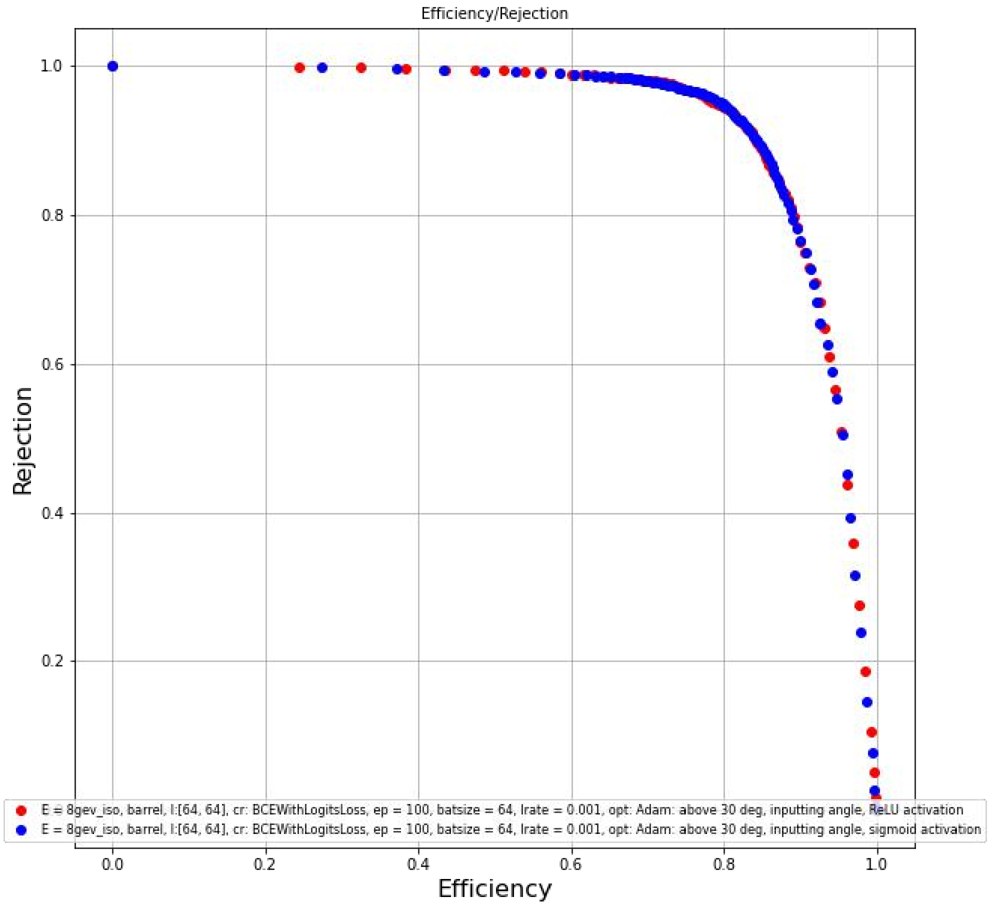
Angle θ as an input variable
(improves separation at high energies)
Total energy

X,Y ~ θ,φ
S_1, M_2 - 1st and 2nd largest energies
S_9, S_{25} - energy in 3x3, 5x5 region
S_6 - maximum energy in 3x2 region containing S_1 and M_2

14 inputs
Dataset: 2/3 → train, 1/3 → test

Hyperparameters (sanity check)

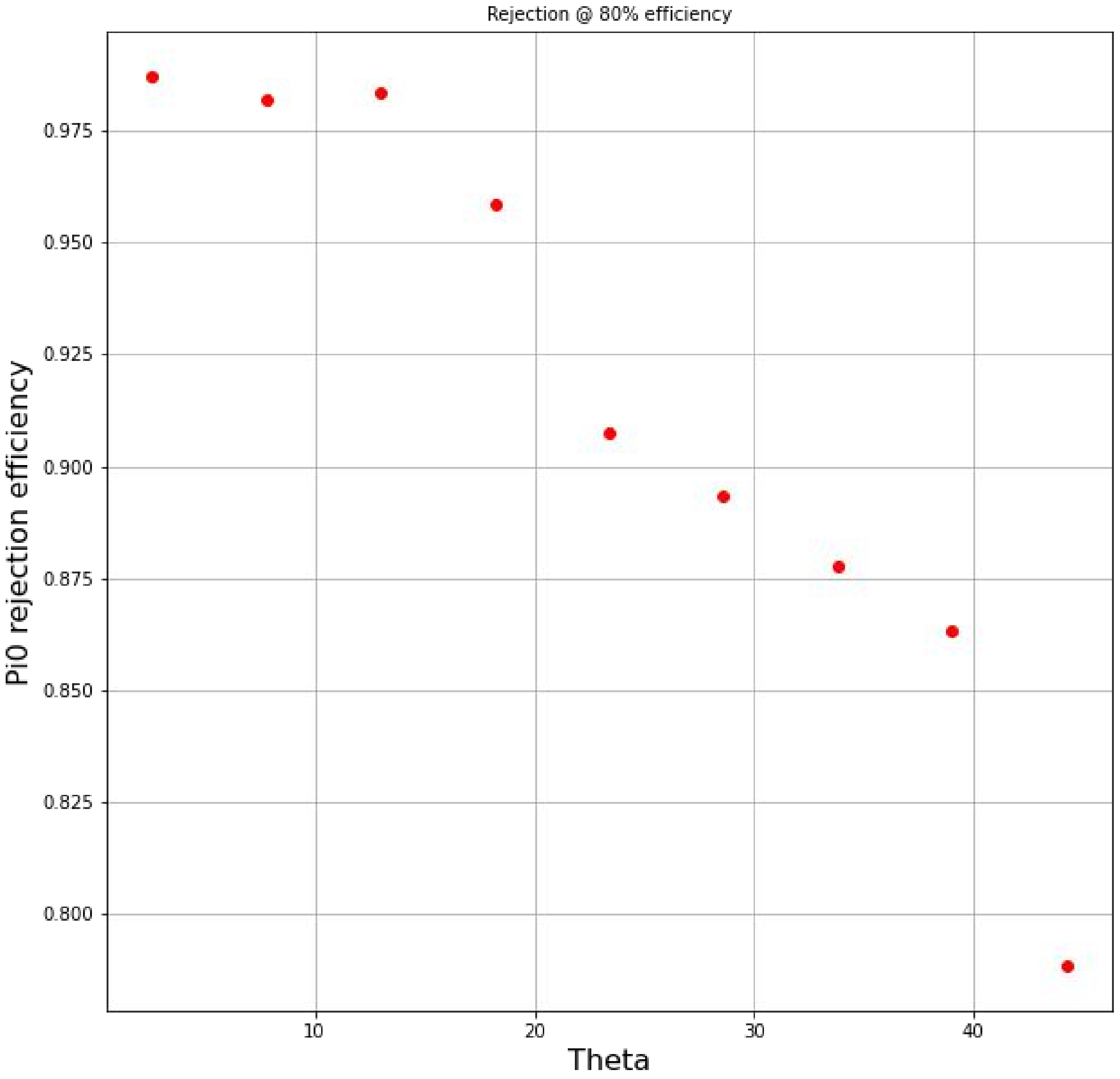


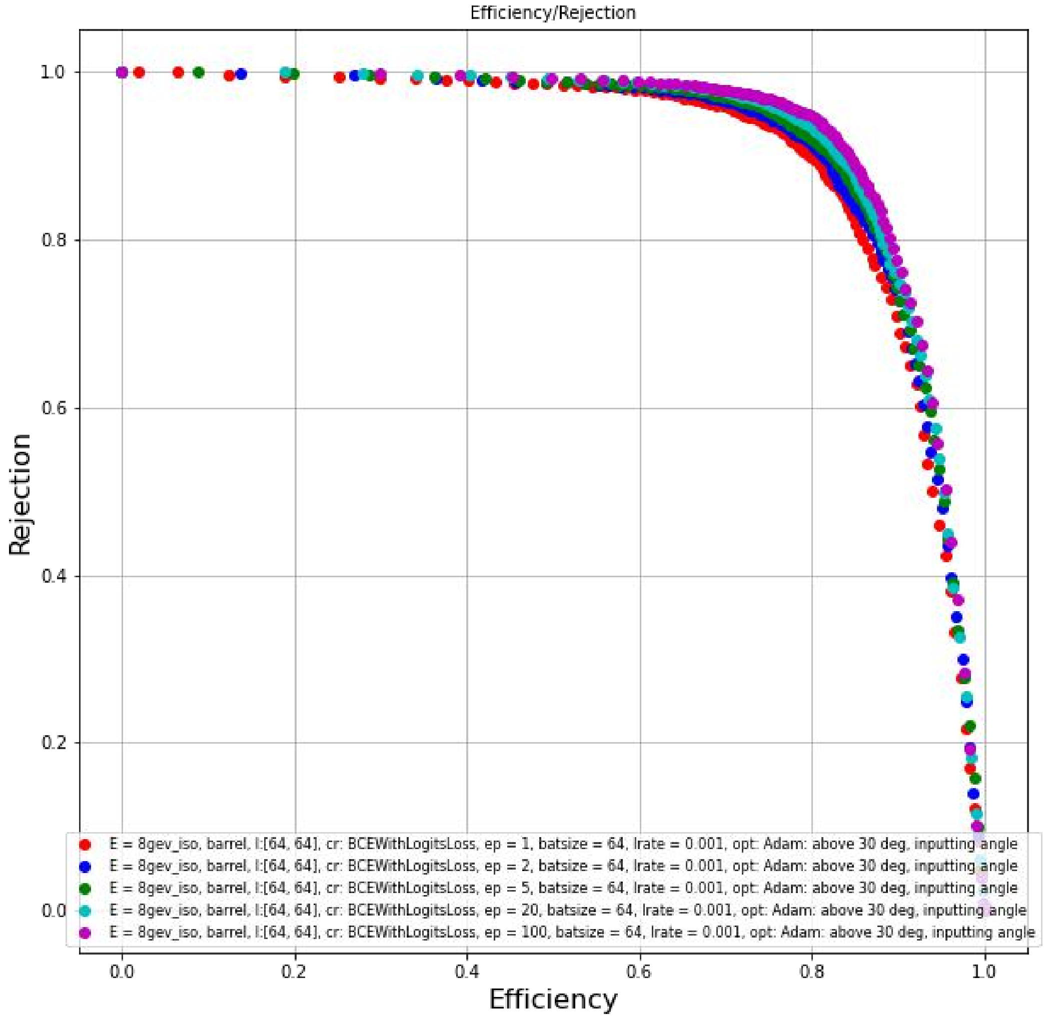


Checked different hyperparameters/network settings:

- batch size
- number of training epochs
- activation function (ReLU/sigmoid)
- learning rate
- loss function (BCE/MSE)
- number of layers and neurons

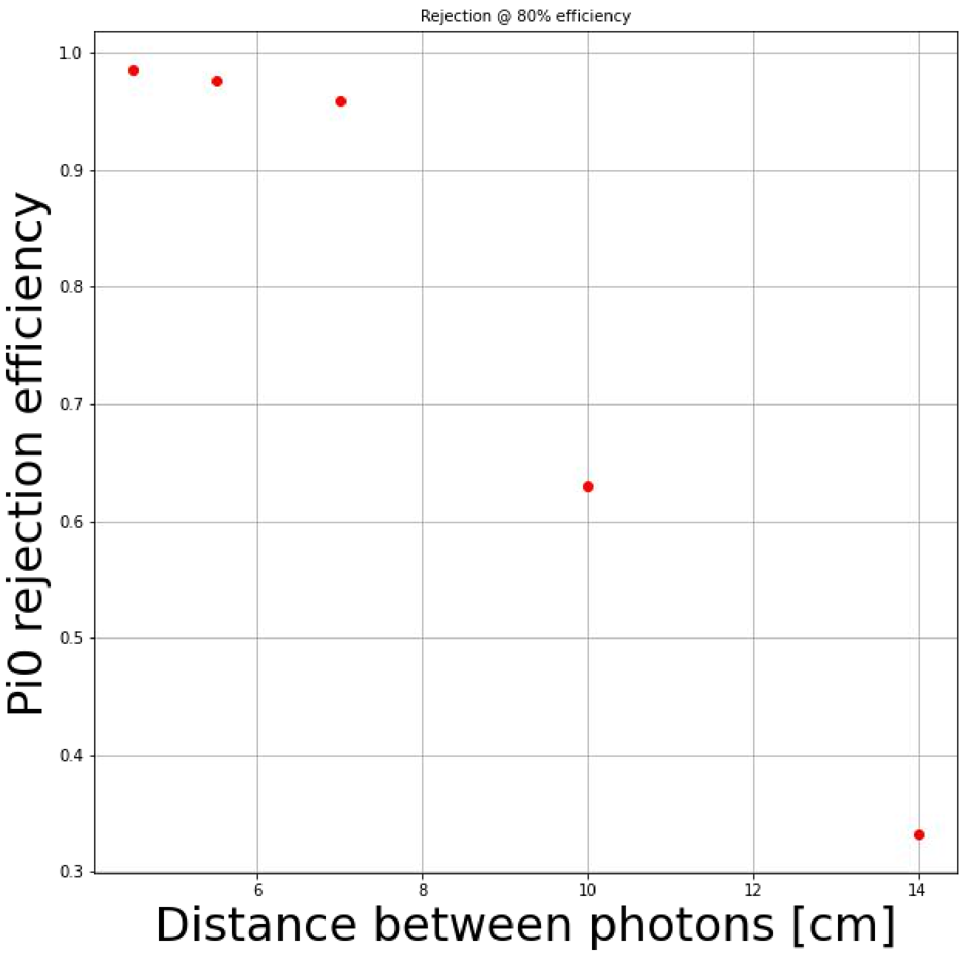
Results **(barrel)**

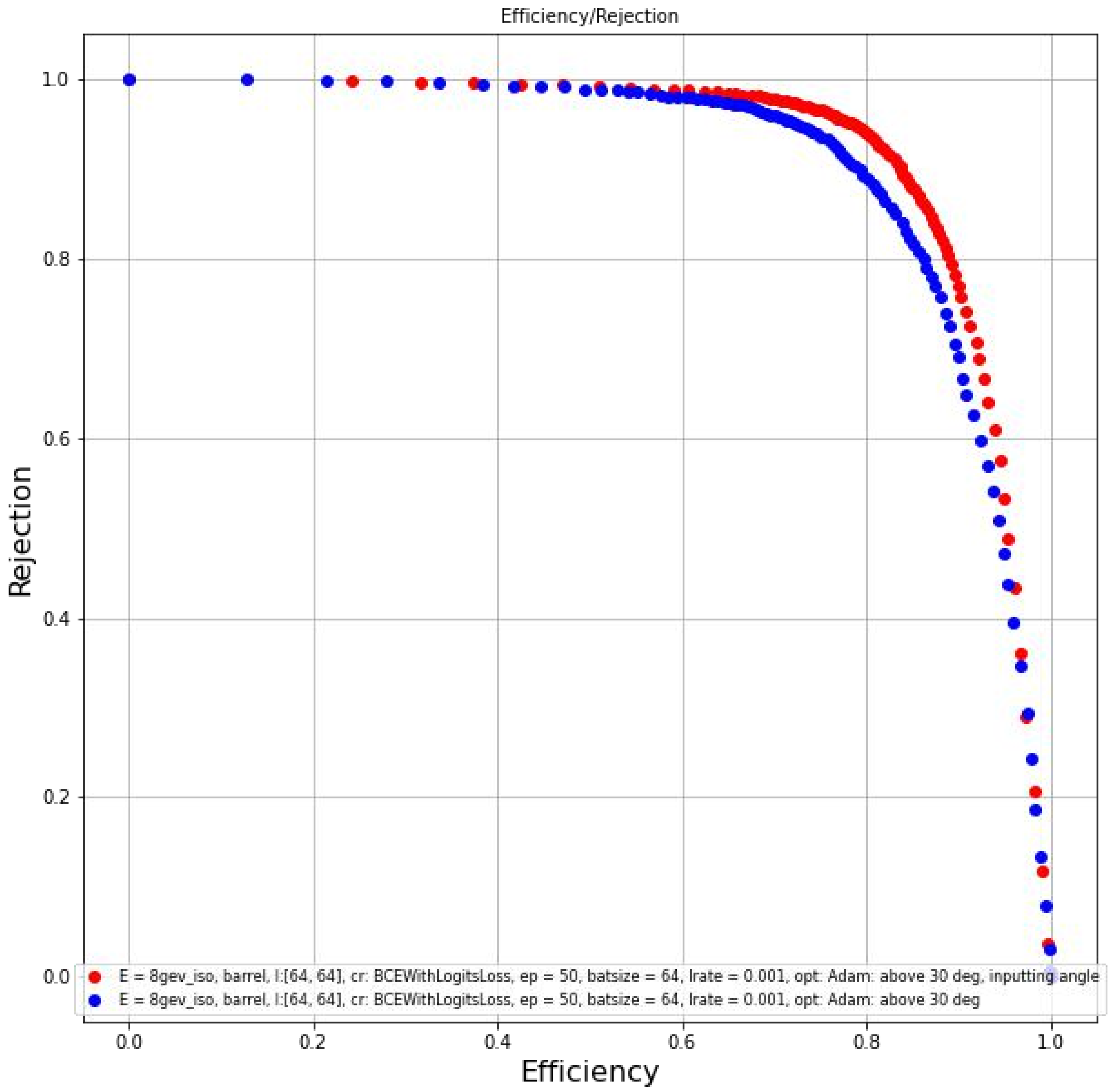




- ~98% π^0 rejection (at 80% γ efficiency) at small angles
- ~80% for larger angles

Results **(barrel)**





π^0 detection inefficiency attributed to soft photon, or a photon hitting barrel edge on the border with the endcap

Conclusions and outlook

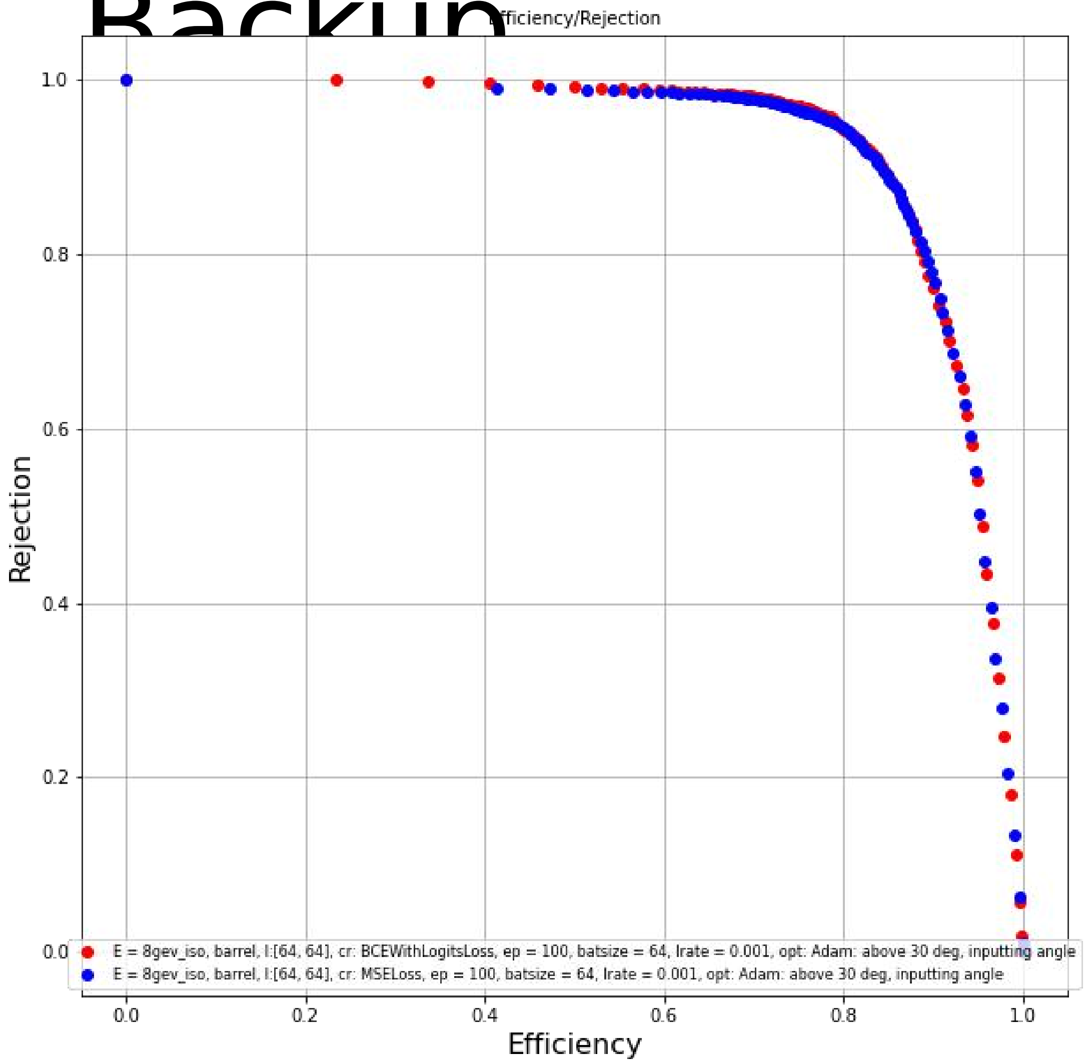
- ~98% π^0 rejection @ 80% efficiency for 8 GeV photons and low incident angles,

  ~80% π^0 rejection for high angles
- Some part of π^0 detection inefficiency attributed to soft photon hitting barrel edge: to be studied
- Efficiency could be overestimated due to fixed energy: to be studied

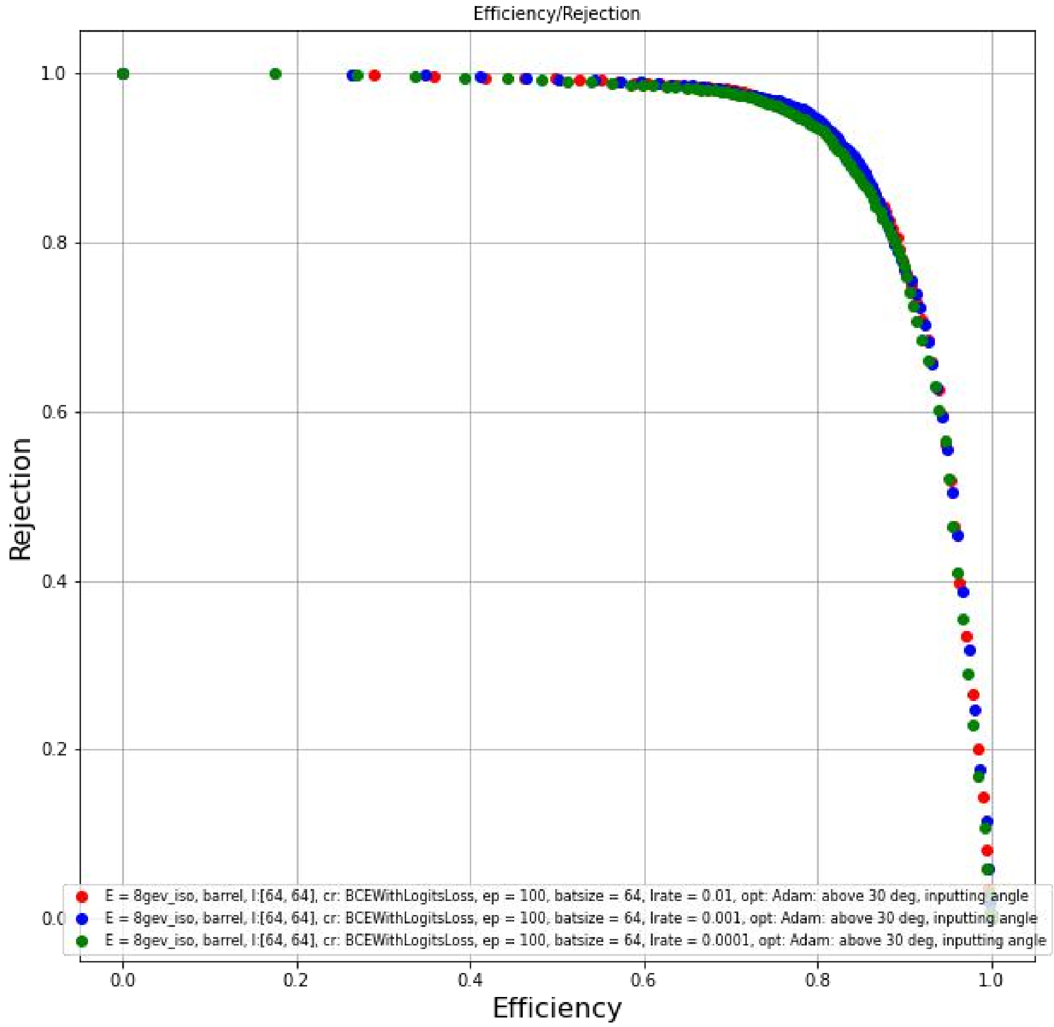
To do:

- repeat the analysis for endcaps (slightly different cell size)
- study dependence on energy of particle
- determine the best set of inputs
- determine a better metric?

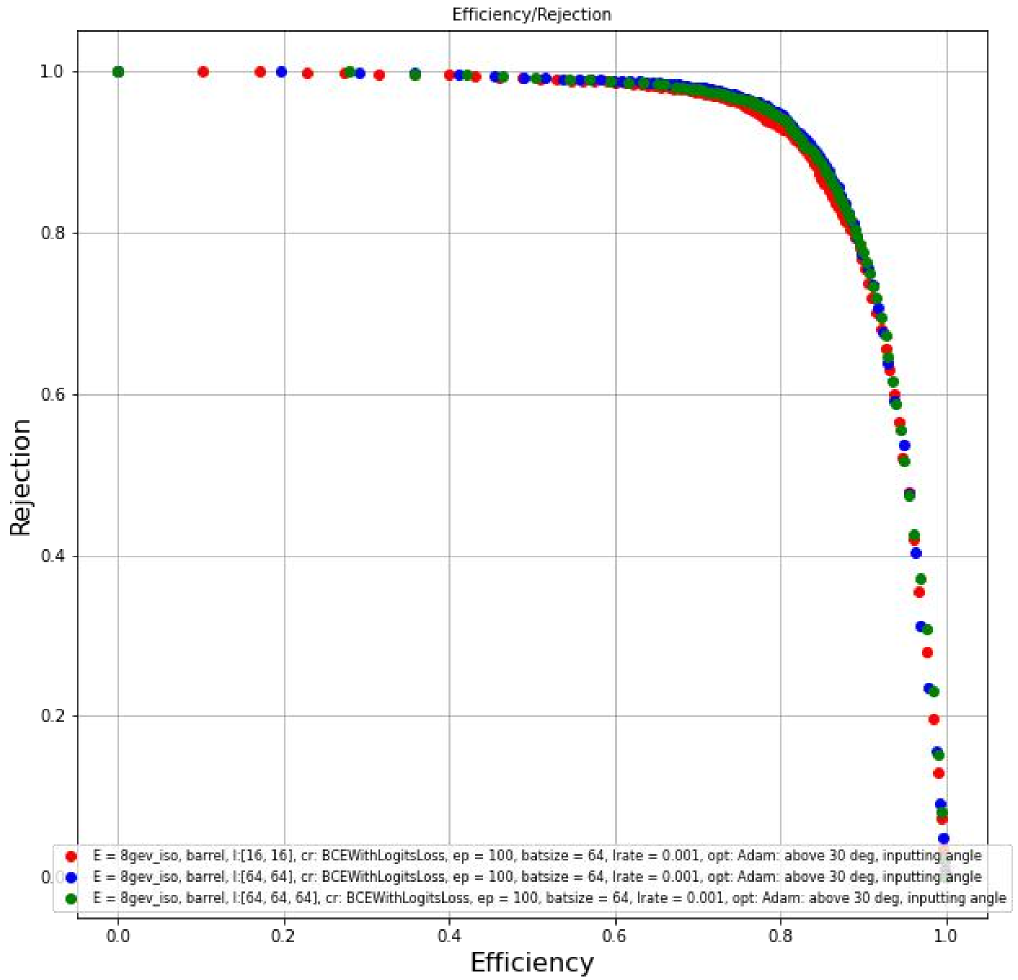
Backup



Efficiency/Rejection
Rejection
Efficiency
0.0
0.2
0.4
0.6
0.8
1.0
E = 8gev_iso, barrel, l:[64, 64], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.001, opt: Adam: above 30 deg, inputting angle
E = 8gev_iso, barrel, l:[64, 64], cr: MSELoss, ep = 100, batsize = 64, lrate = 0.001, opt: Adam: above 30 deg, inputting angle



Efficiency/Rejection
Rejection
Efficiency
0.0
0.2
0.4
0.6
0.8
1.0
E = 8gev_iso, barrel, l:[64, 64], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.01, opt: Adam: above 30 deg, inputting angle
E = 8gev_iso, barrel, l:[64, 64], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.001, opt: Adam: above 30 deg, inputting angle
E = 8gev_iso, barrel, l:[64, 64], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.0001, opt: Adam: above 30 deg, inputting angle


Efficiency/Rejection
Rejection
Efficiency
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0
E = 8gev_iso, barrel, l:[16, 16], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.001, opt: Adam: above 30 deg, inputting angle
E = 8gev_iso, barrel, l:[64, 64], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.001, opt: Adam: above 30 deg, inputting angle
E = 8gev_iso, barrel, l:[64, 64, 64], cr: BCEWithLogitsLoss, ep = 100, batsize = 64, lrate = 0.001, opt: Adam: above 30 deg, inputting angle