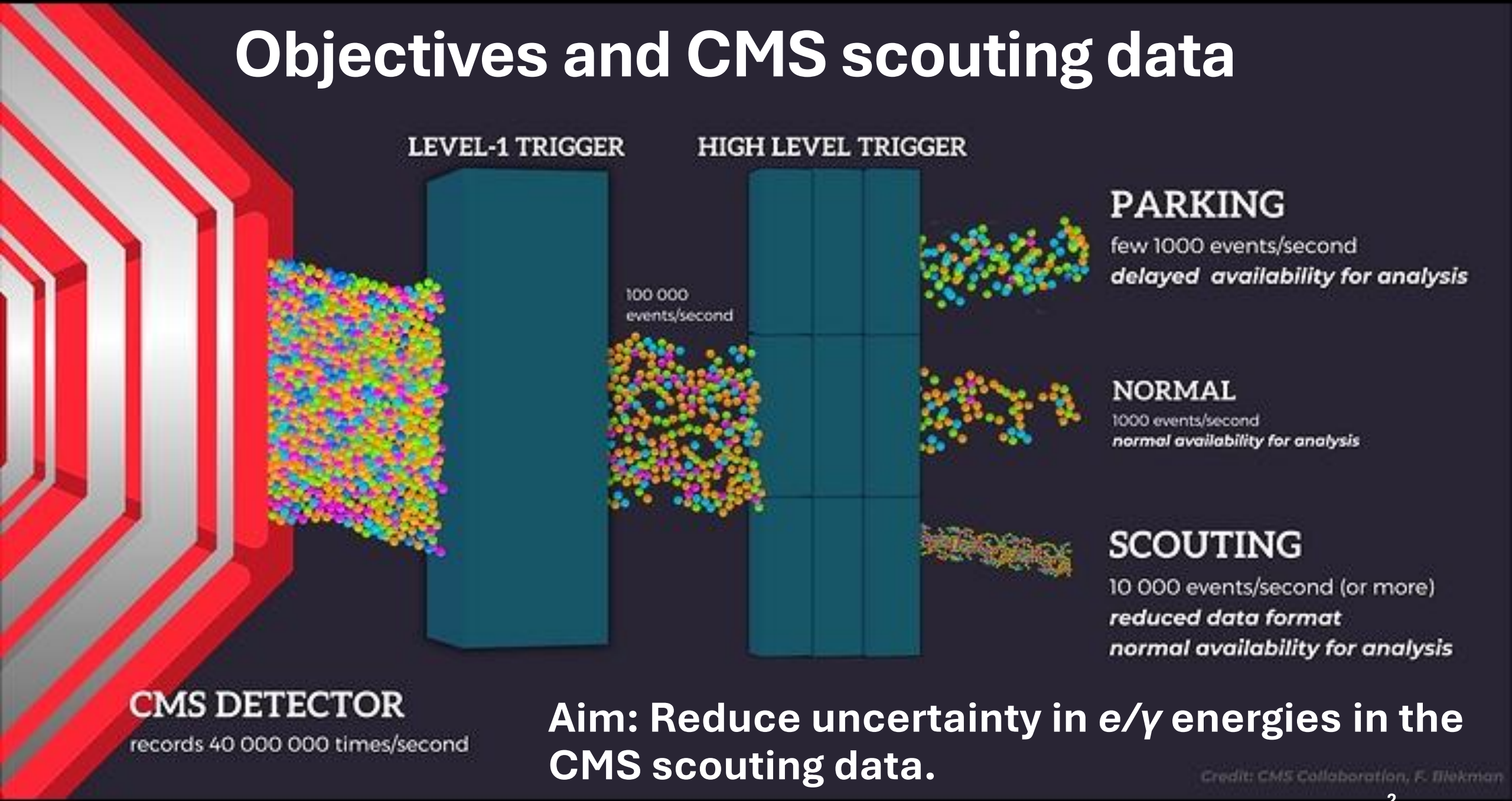
Exploring machine learning to reconstruct electron and photon energies in the CMS scouting data

Bartosz Dlubak

Supervisor: A.R.Sahasransu

Objectives and CMS scouting data

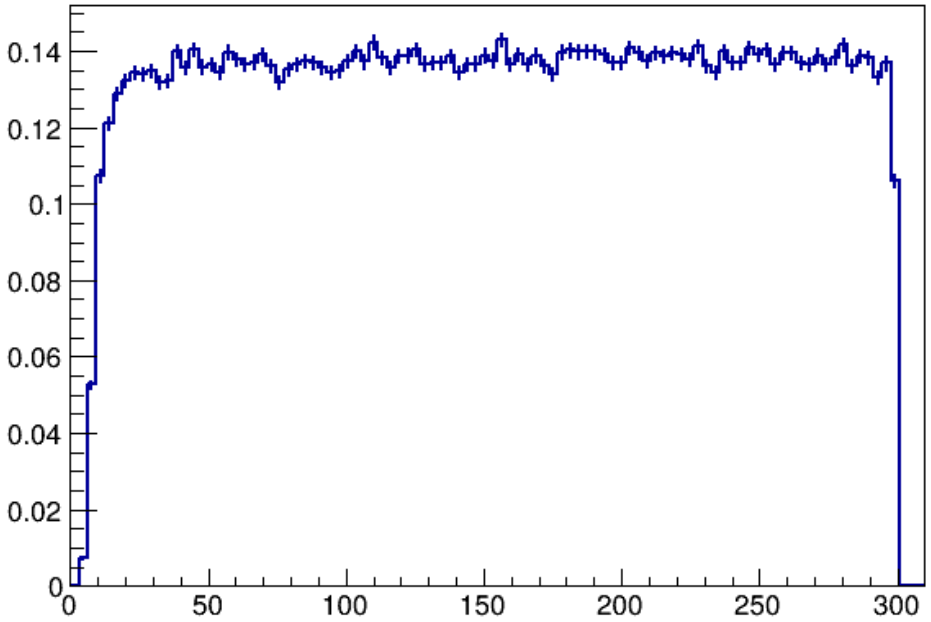


Aim: Reduce uncertainty in *e/γ* energies in the CMS scouting data.

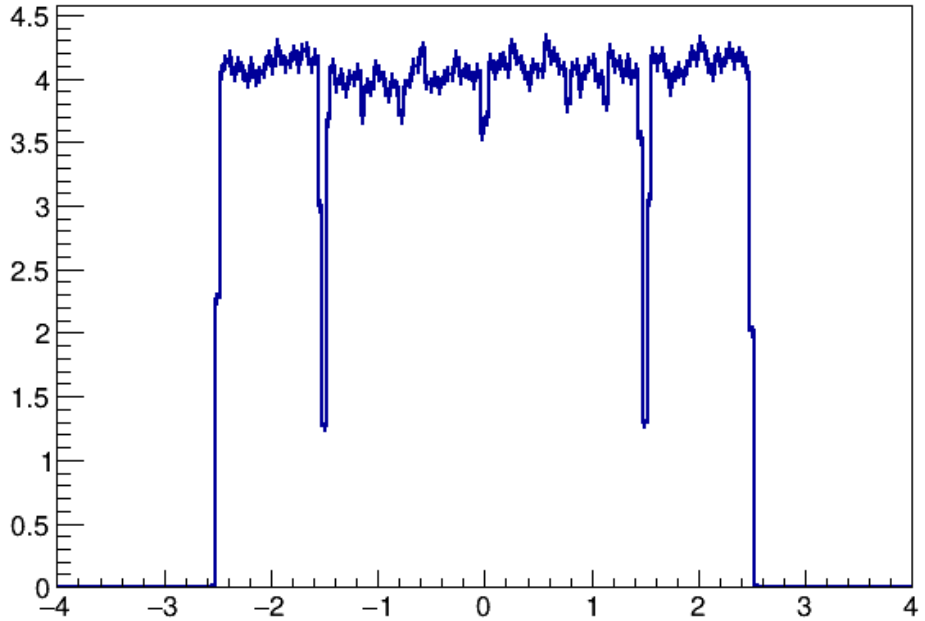
Credit: CMS Collaboration, F. Blekman

Simulated data

- Uniform distribution over MC electron (p_T, η, ϕ).
- Dips in MC η correspond to the barrel – endcap transition region.

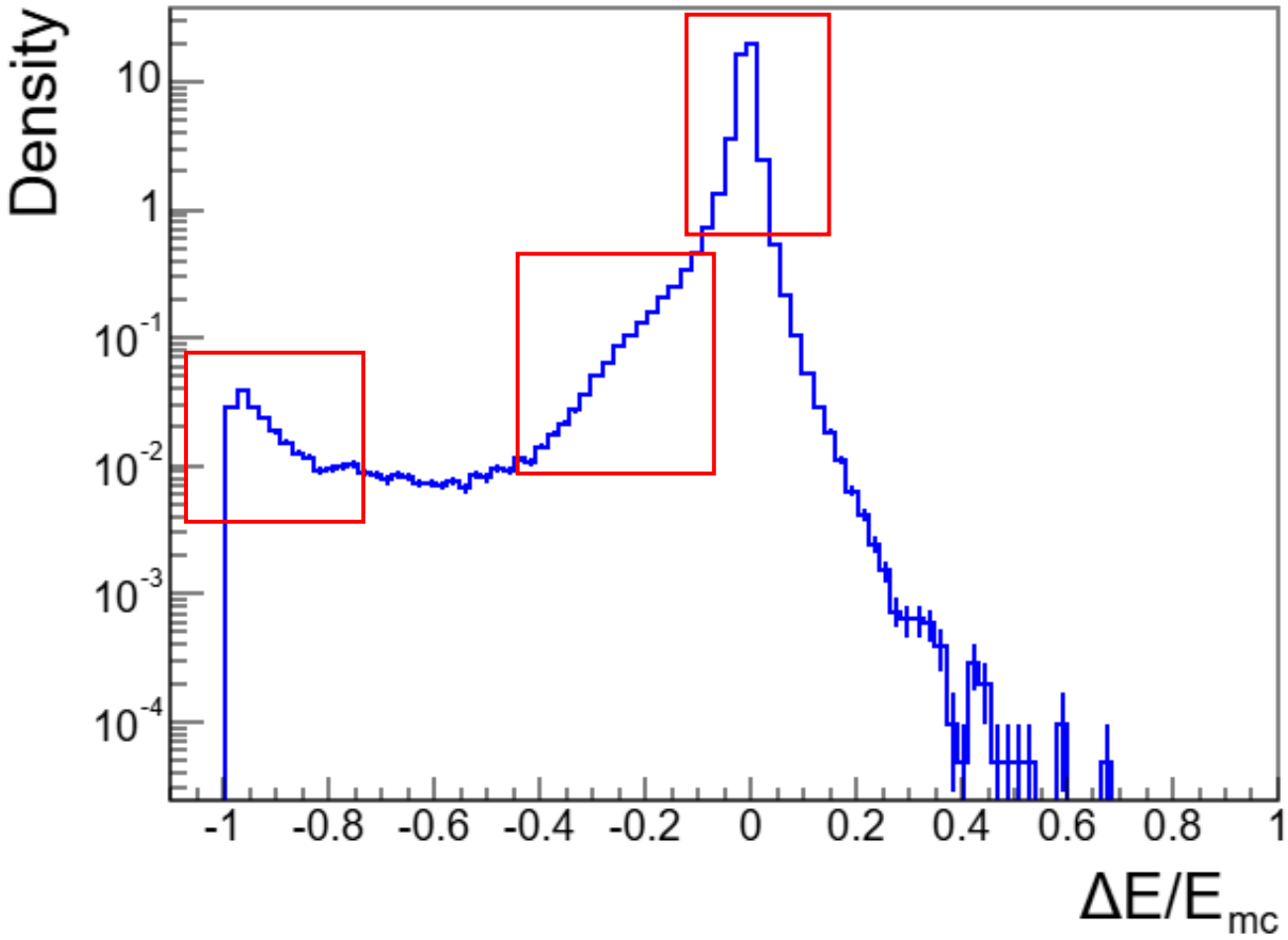


MC p_T



MC η

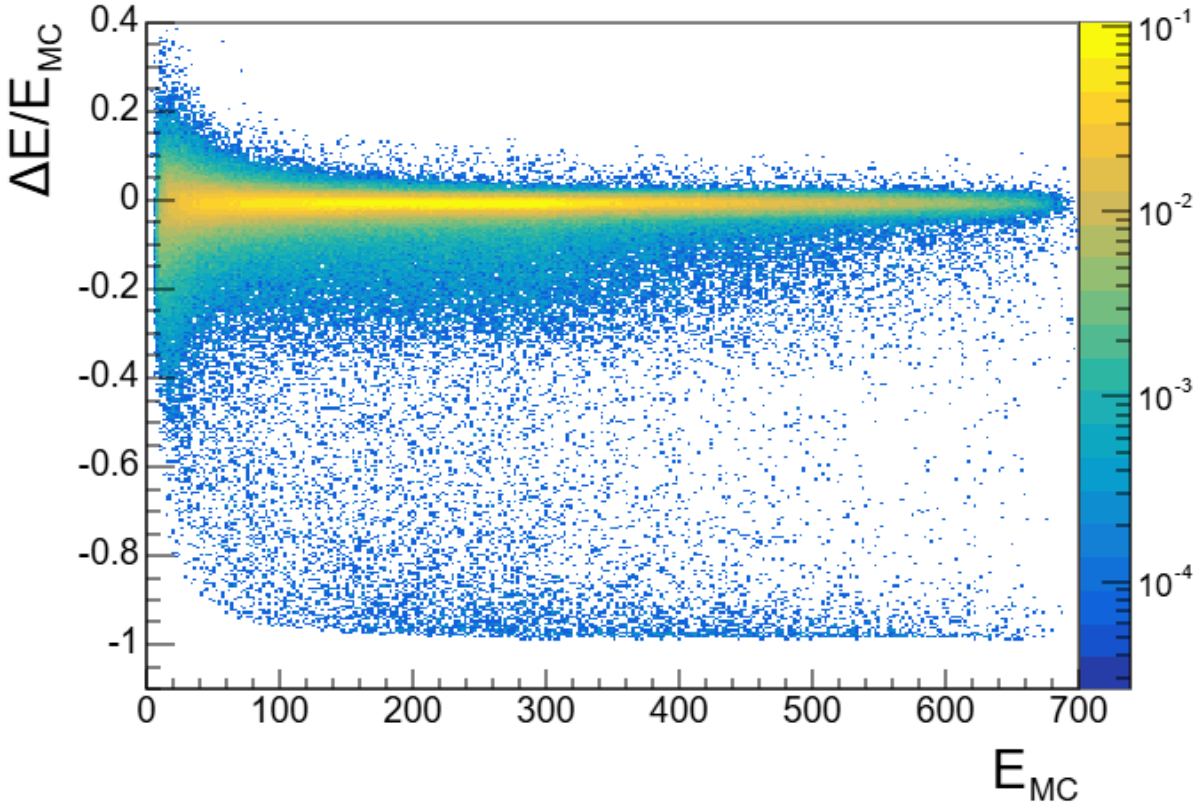
Uncertainty in scouting electron energy

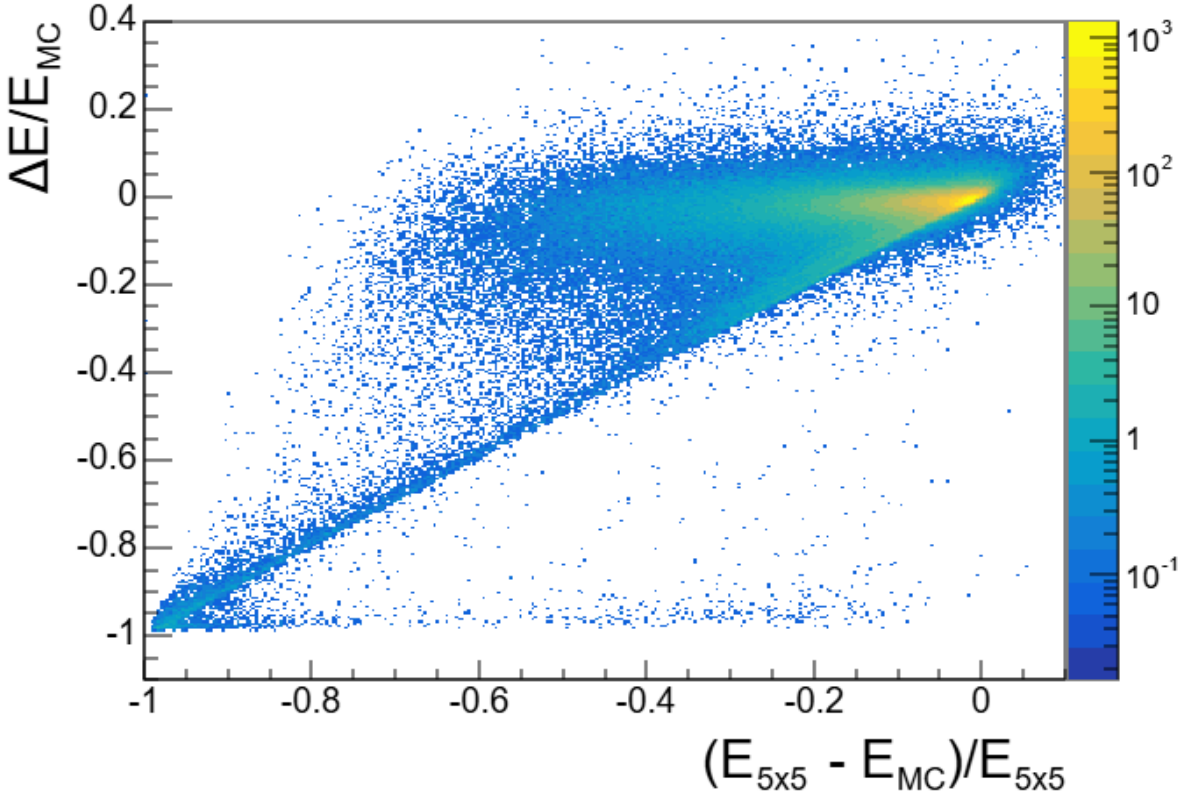


- E_{reco}: calibrated ECAL energy
- $\Delta E/E_{MC} = ((E_{reco} - E_{MC})/E_{MC})$
- Peak observed ~ 0.0,
- Shoulder ~ -0.2 and
- Peak at -1.0

Correlations between features and uncertainty

- Shoulder contribution has a linear dependence on shower containment in a 5x5 crystal block





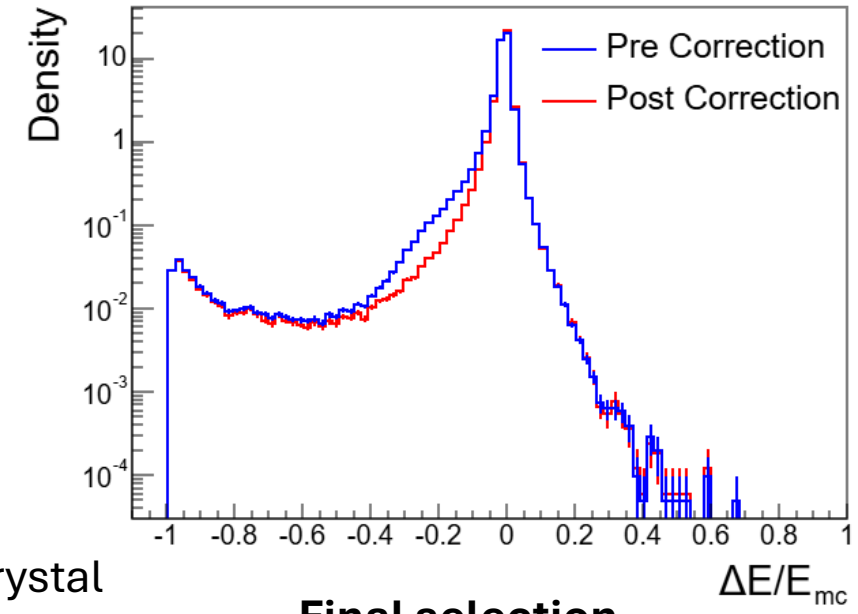
Identifying good e (Barrel)

Two known noisy effects due to:

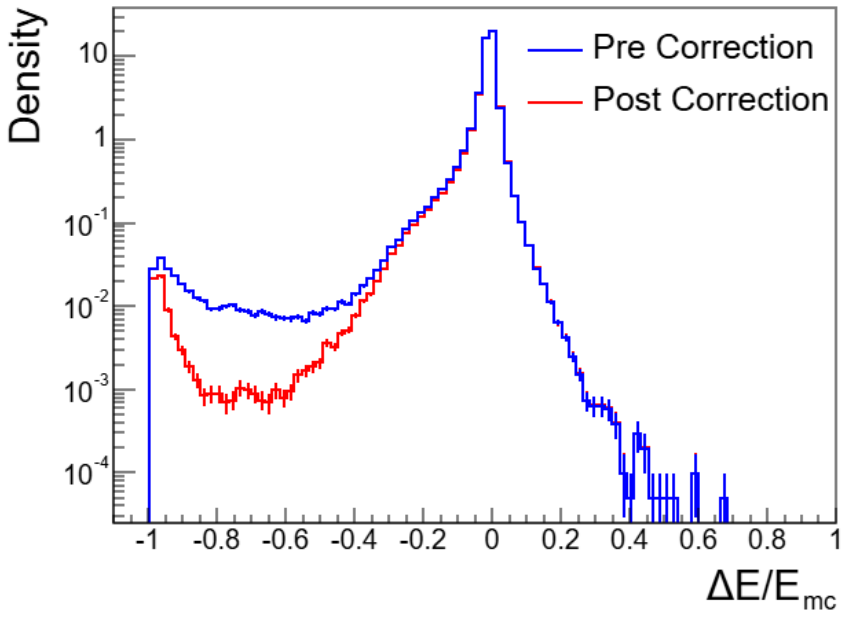
- Super modules and
- Dead crystals

Can be identified in the (η, ϕ) map.

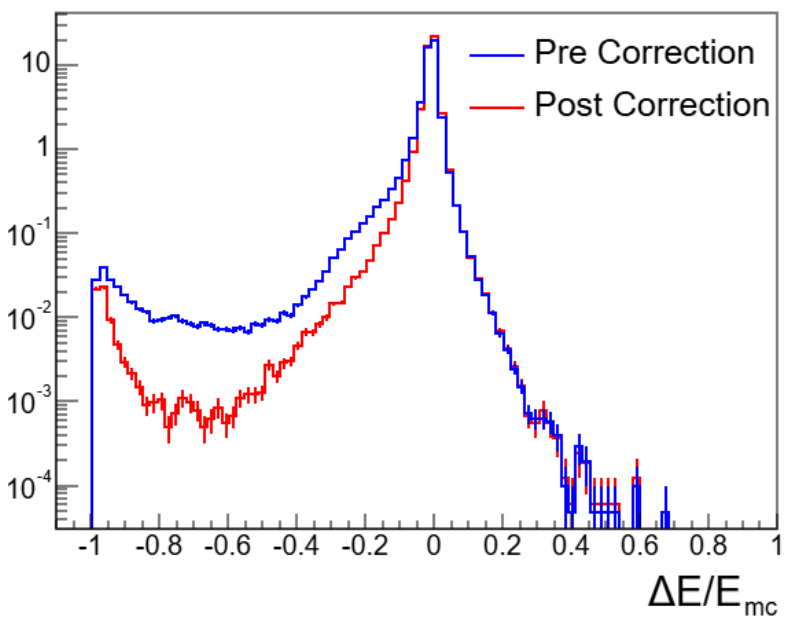
Super module



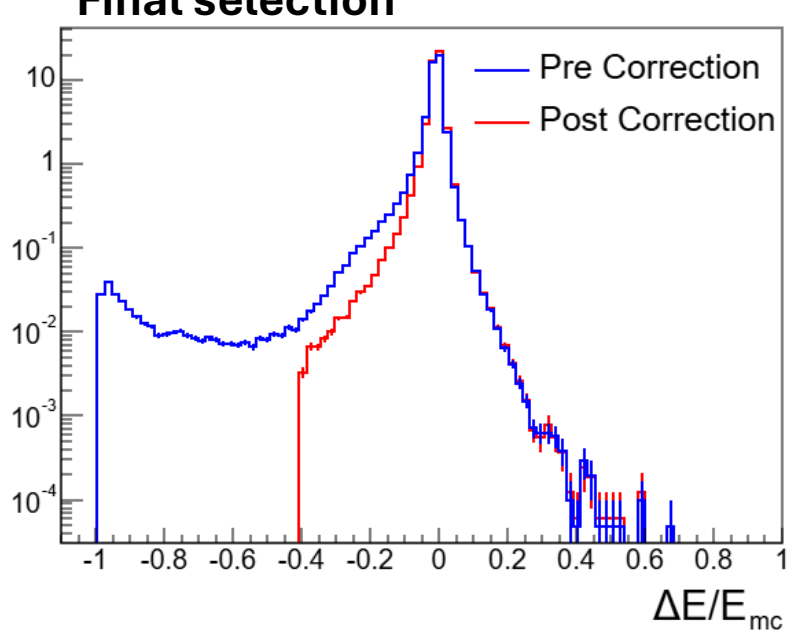
Dead crystal



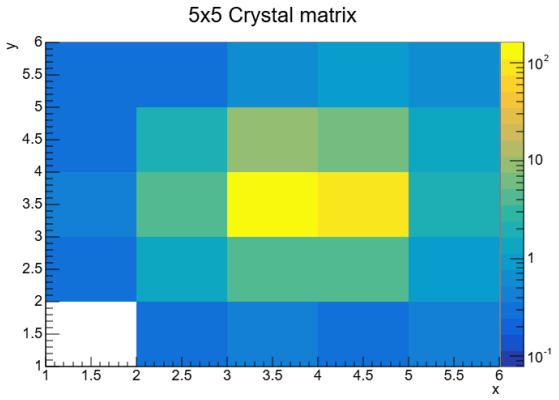
Super module + Dead crystal



Final selection



ML Models: BDT and NN



Trying two different approaches:

Shower shapes

(p_T, η, ϕ)

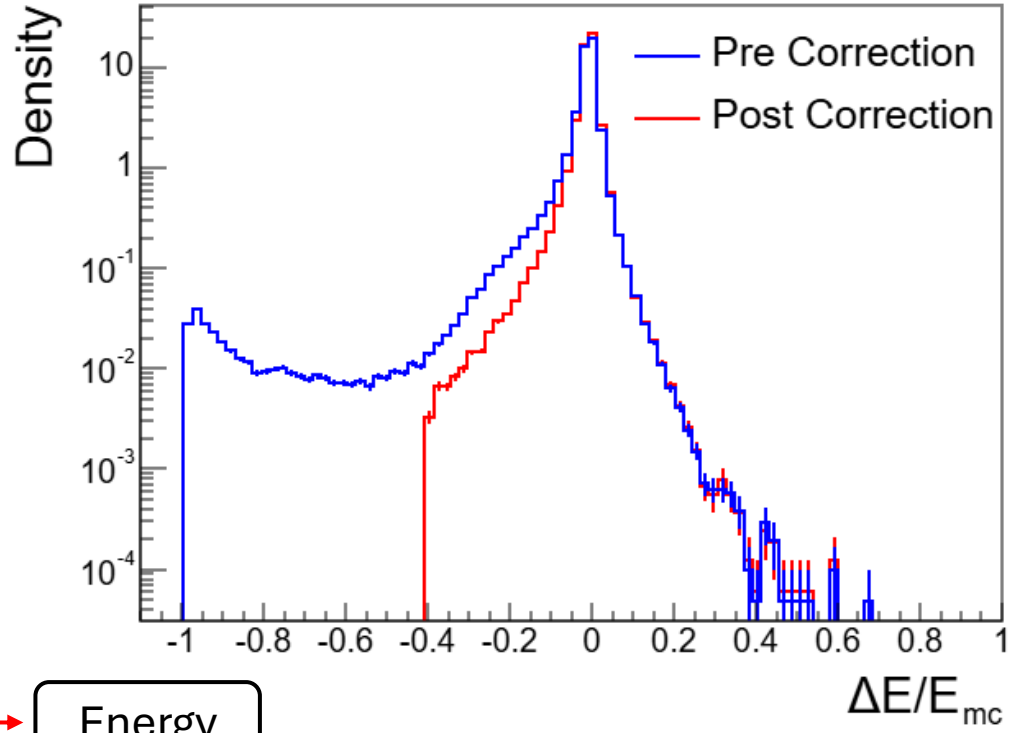
Tracker data

NN

BDT

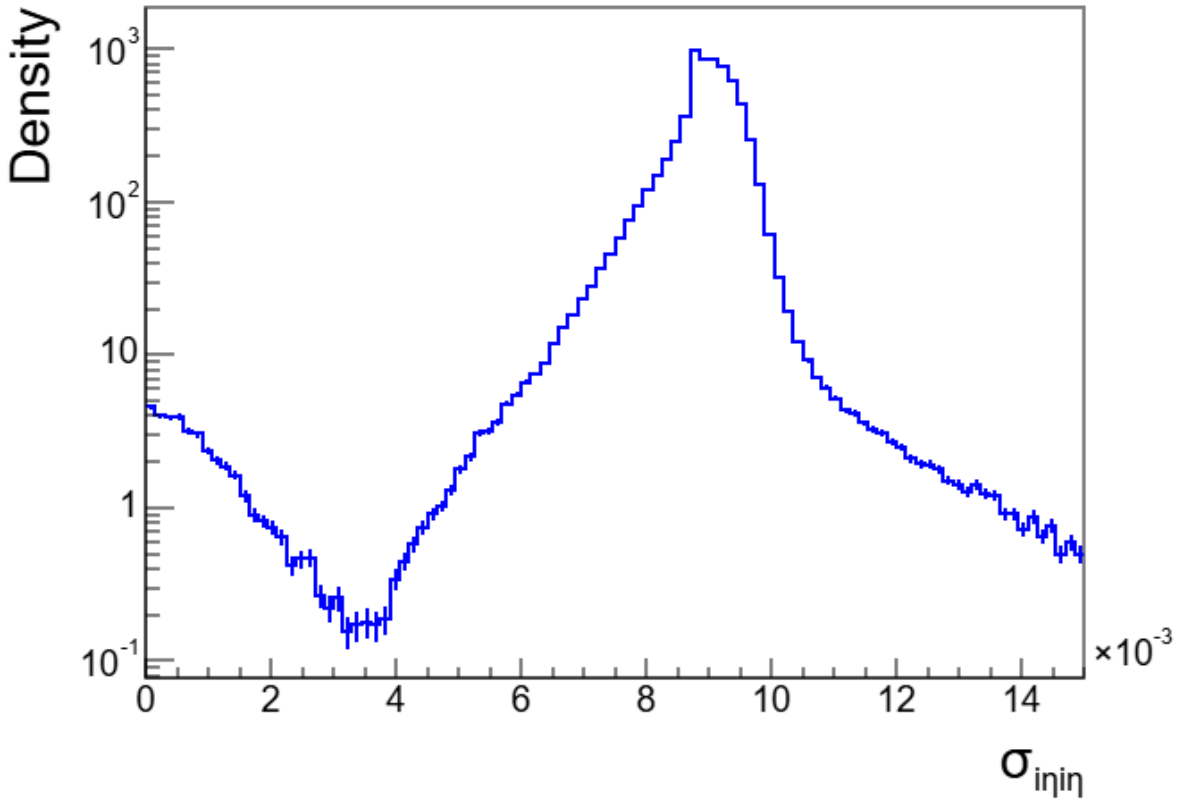
Energy

Correction factor



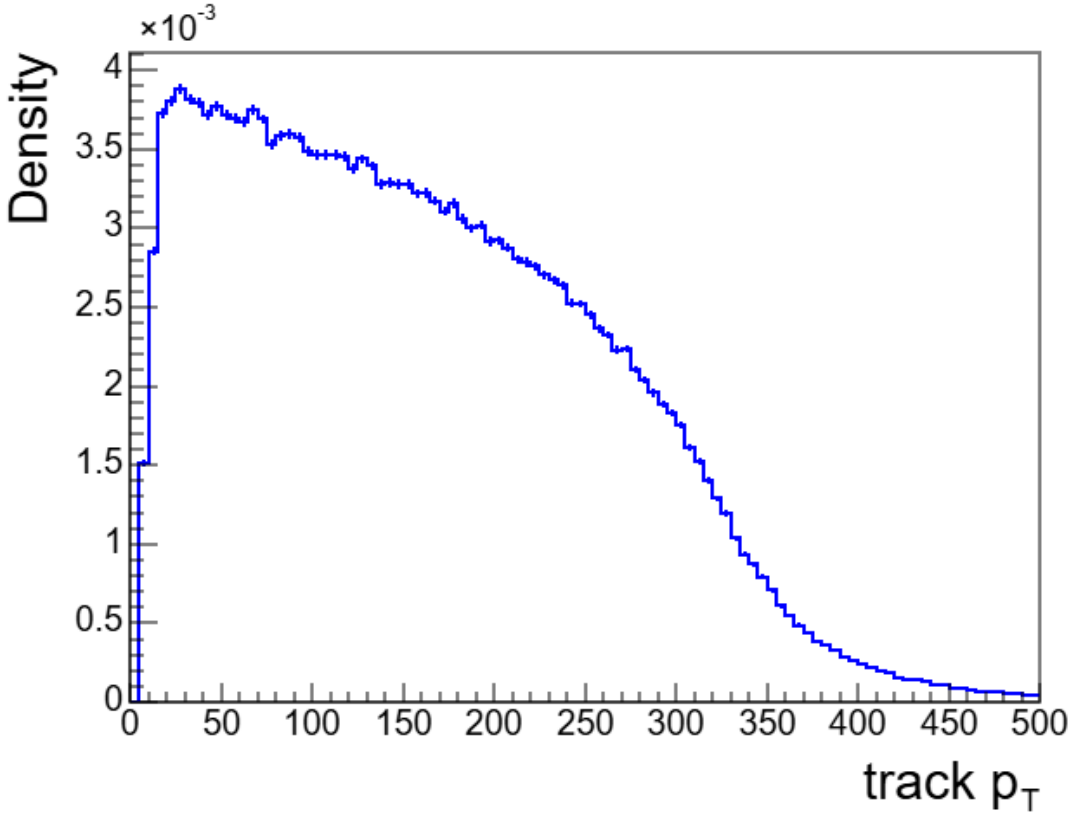
Use multivariate analysis models to reduce the uncertainty on the energy reconstruction for *e/γ*.

Input features



$$\sigma_{i\eta i\eta} = \sqrt{\frac{\sum_i^{5\times5} w_i(\eta_i - \bar{\eta}_{5\times5})^2}{\sum_i^{5\times5} w_i}}$$

$$w_i = \max(0, 4.7 + \ln(E_i/E_{5\times5}))$$



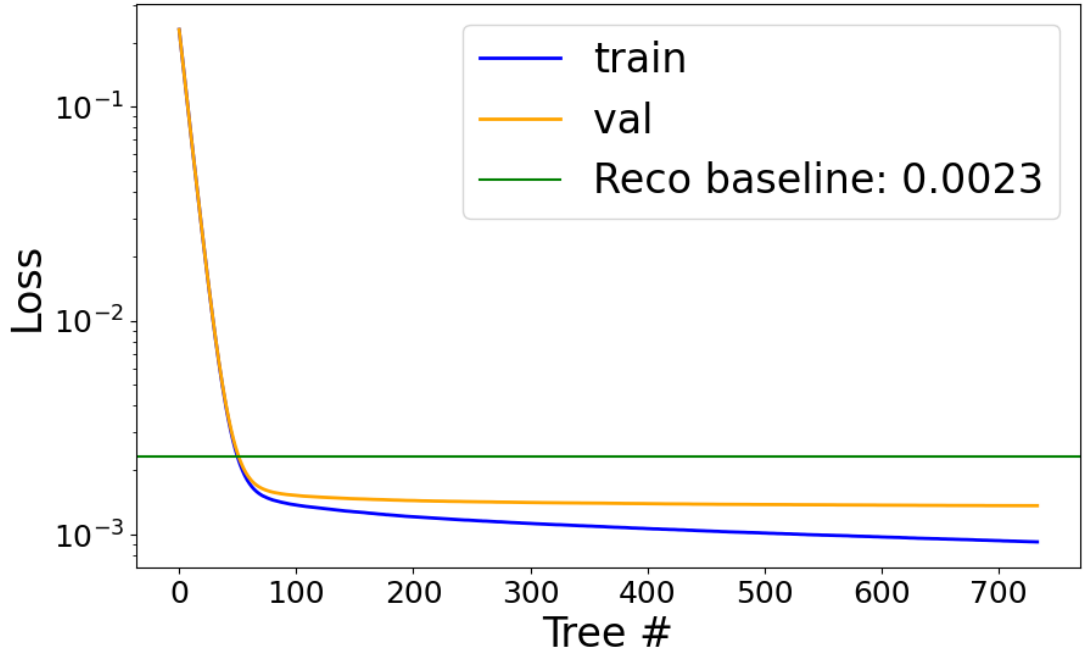
Two distinct category of features;

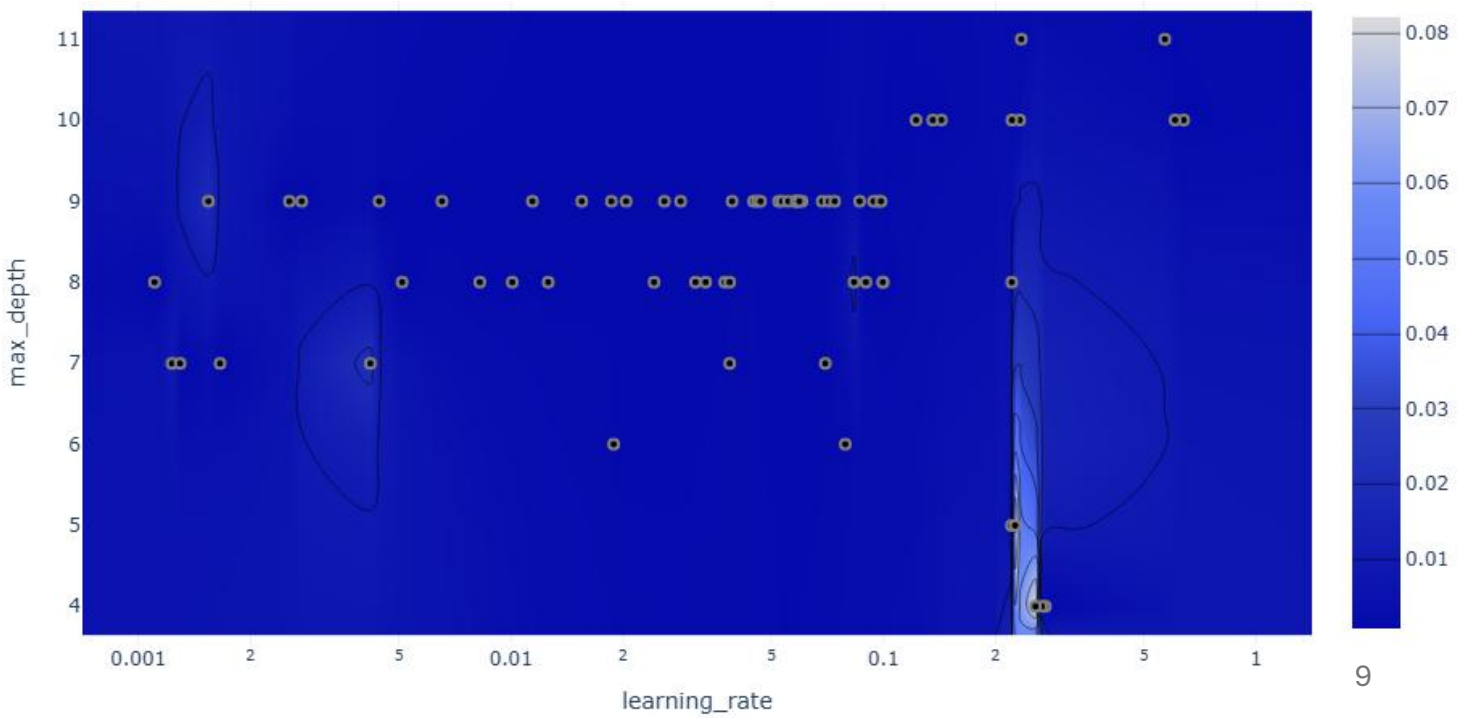
- ECAL only for γ
- Additional track parameters for e

BDT Training

- Targets:
  - $y_{true} = E_{MC}/E_{reco}$
- Loss:
  - $= ((E_{pred} - E_{MC})/E_{MC})^2$
  - $= (\frac{y_{pred}}{y_{true}} - 1)^2$

- Used Optuna for hyperparameter optimisation.
  - https://optuna.org/

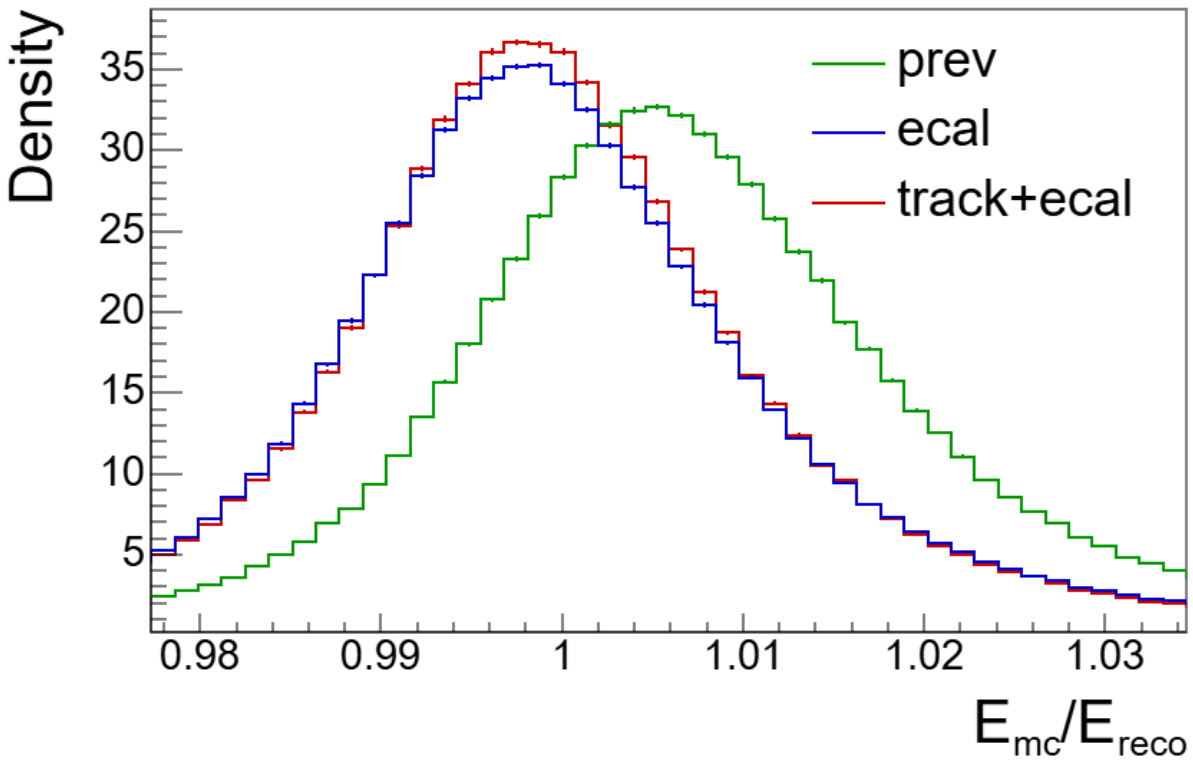




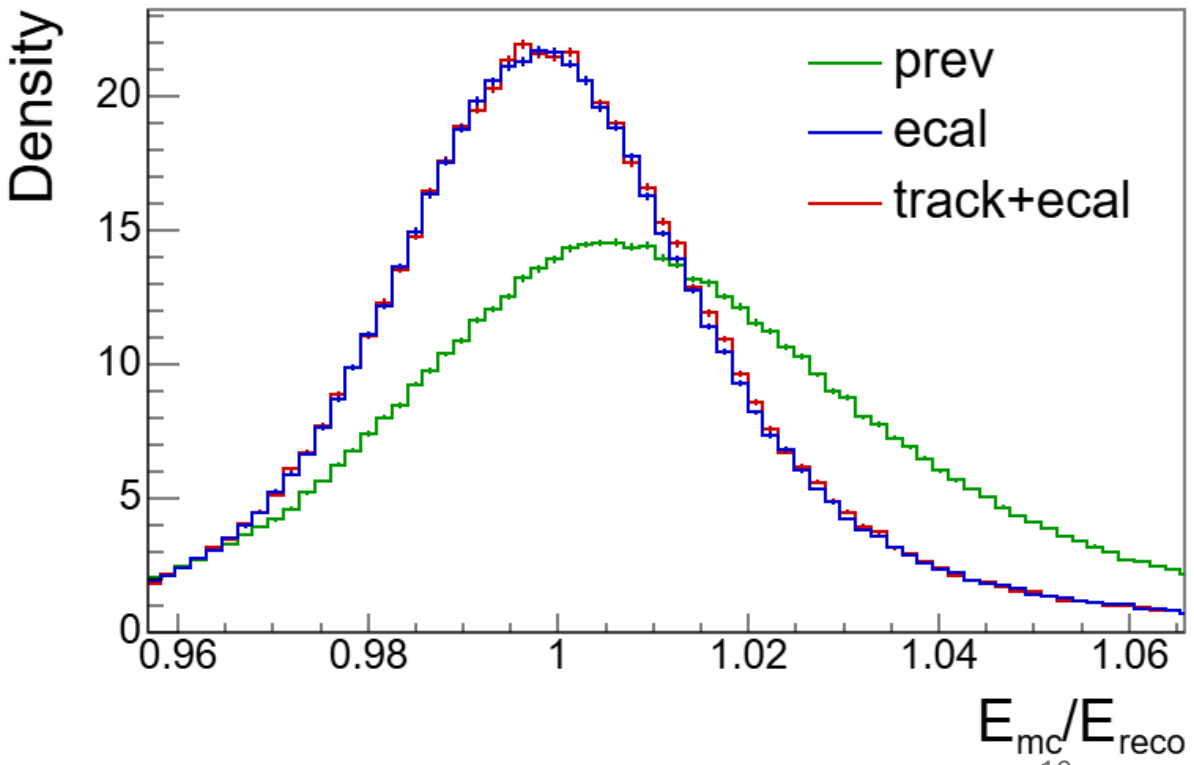
BDT Results

- Peak shifted closer to one and lower spread in both ECAL barrel and endcap.

Barrel (EB)

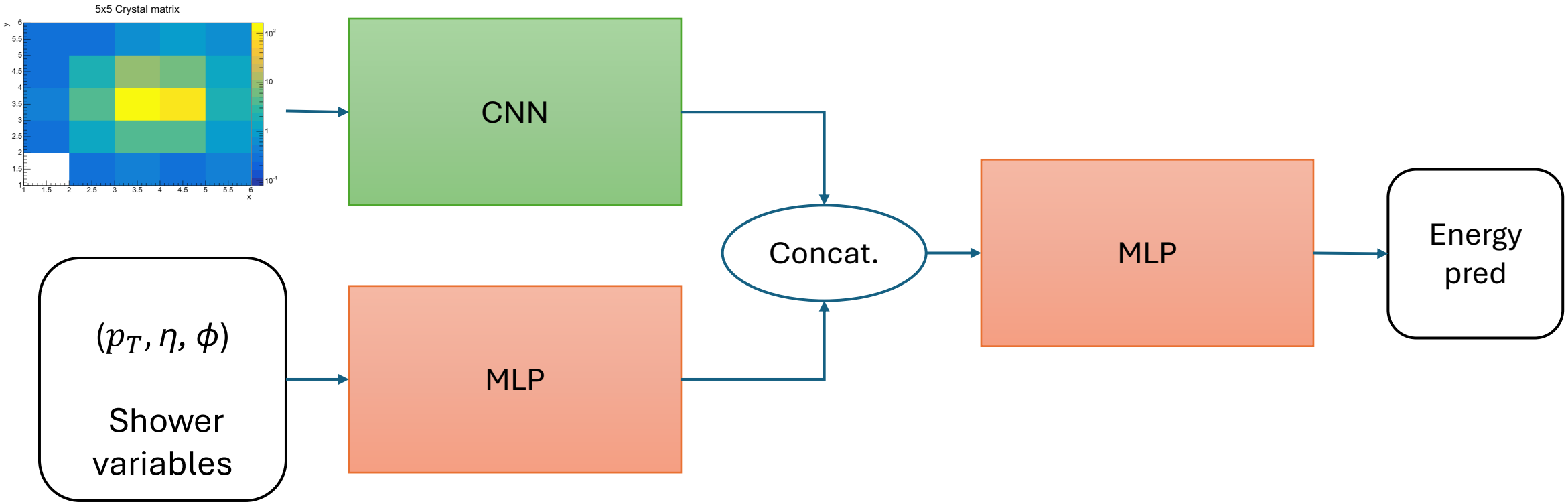


Endcaps (EE)



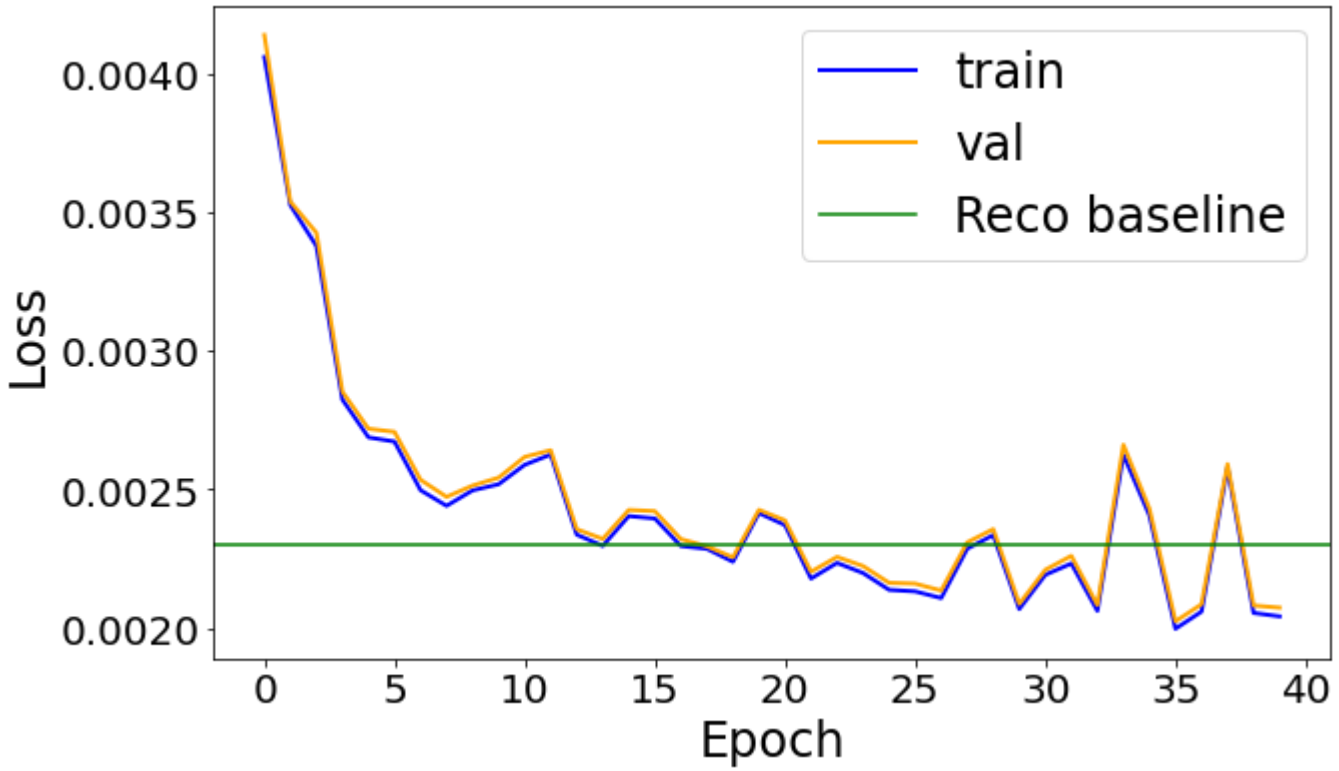
Deep Learning: CNN based architecture

- $Loss := ((E_{pred} - E_{MC})/E_{MC})^2$



NN Training

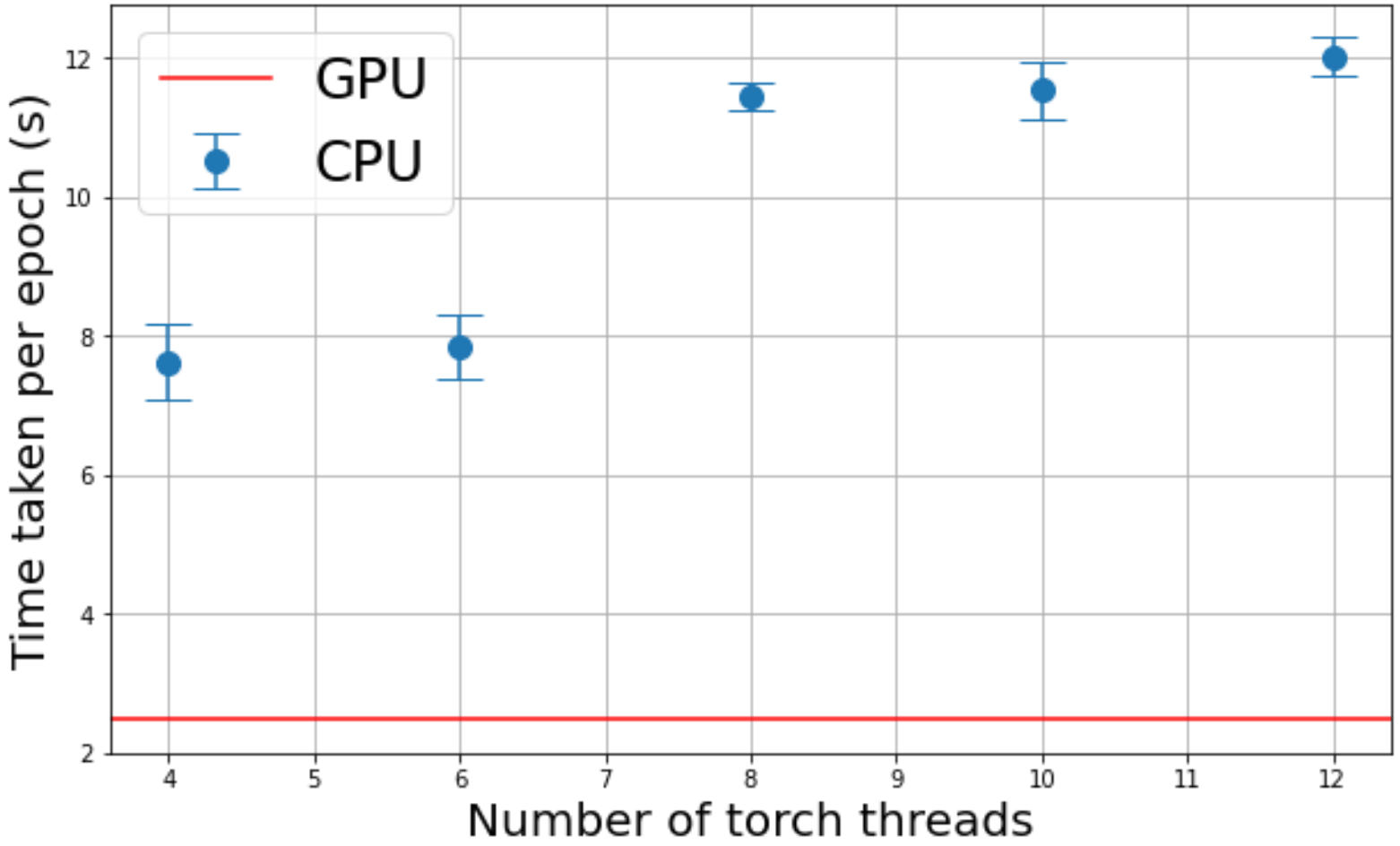
- Validation and train loss curve are very similar
  - =>Train and validation come from a similar distribution (large enough datasets)



- $Loss := ((E_{pred} - E_{MC})/E_{MC})^2$

Acceleration with GPUs for NNs

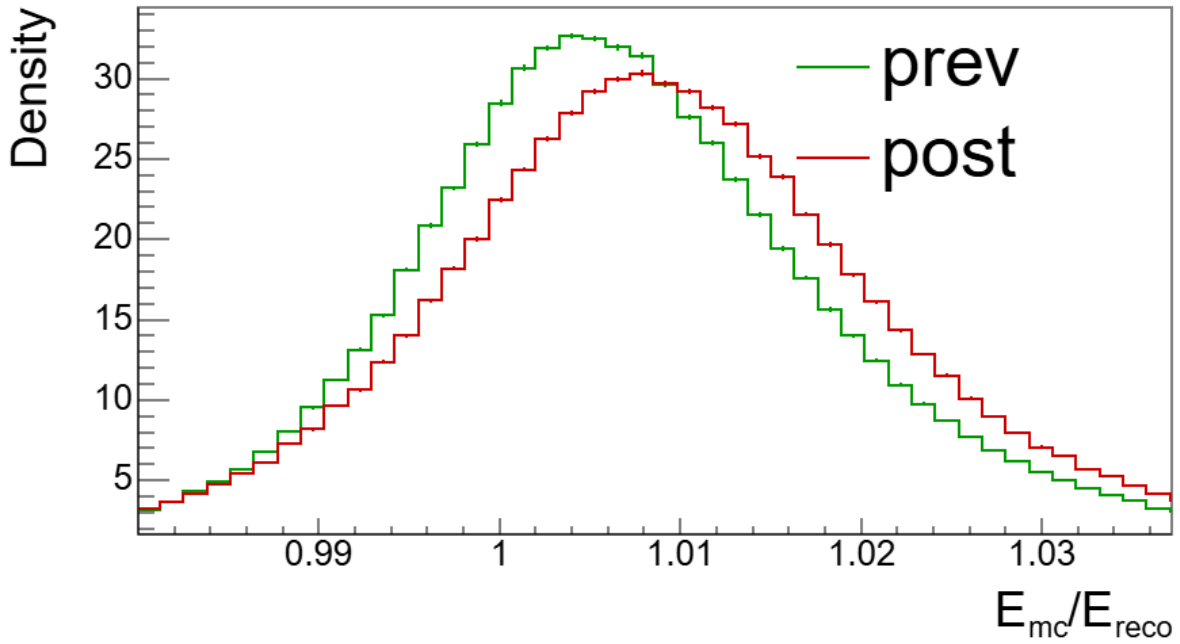
- GPU 4-6x faster than all CPU setting



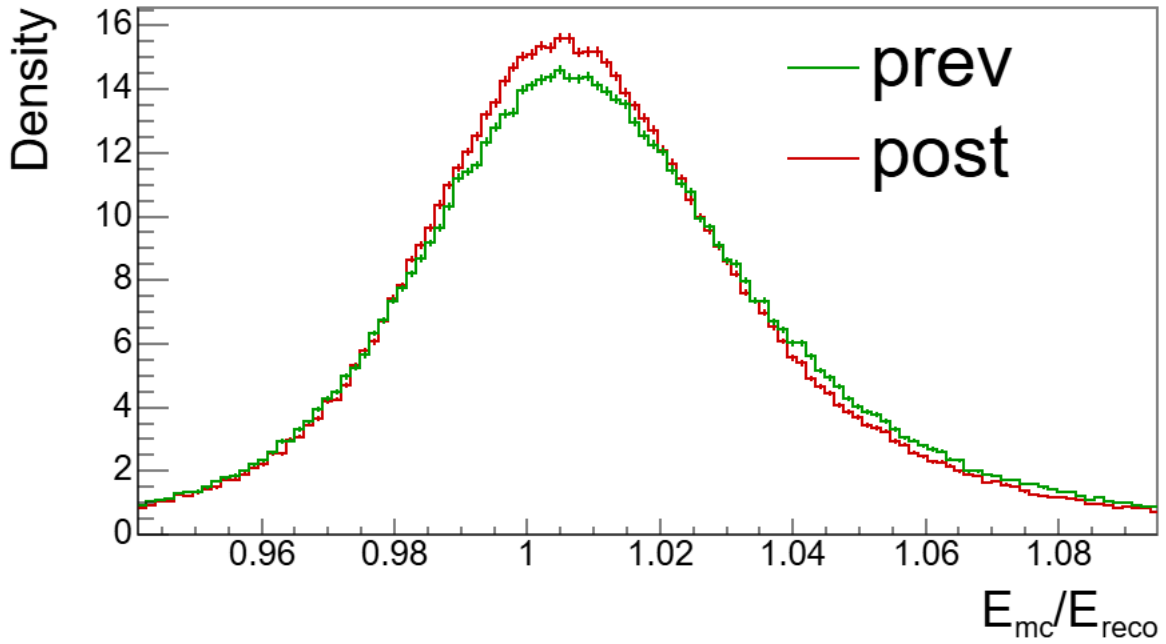
Results of NN

- CNN needs to be further optimised

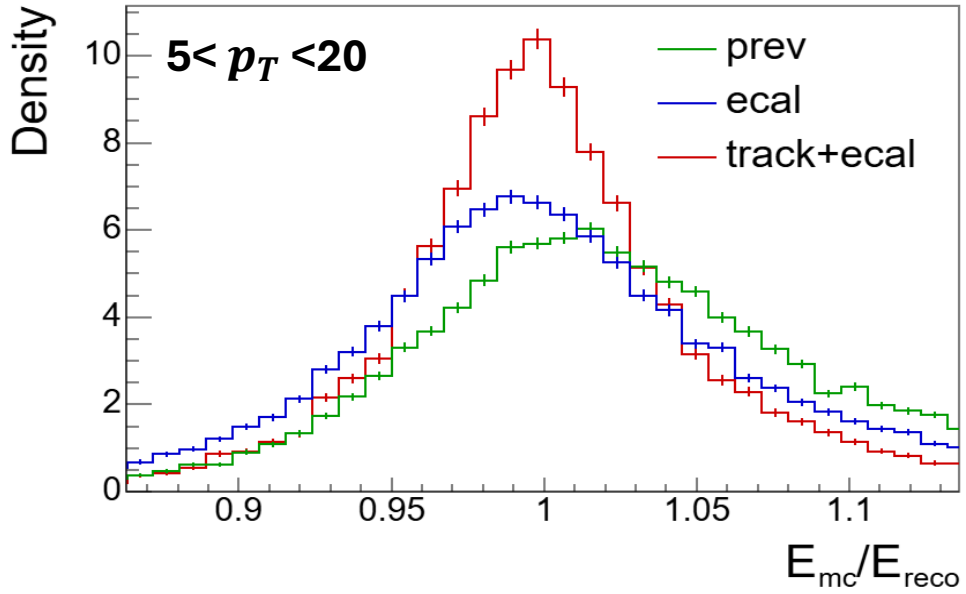
Barrel (EB)

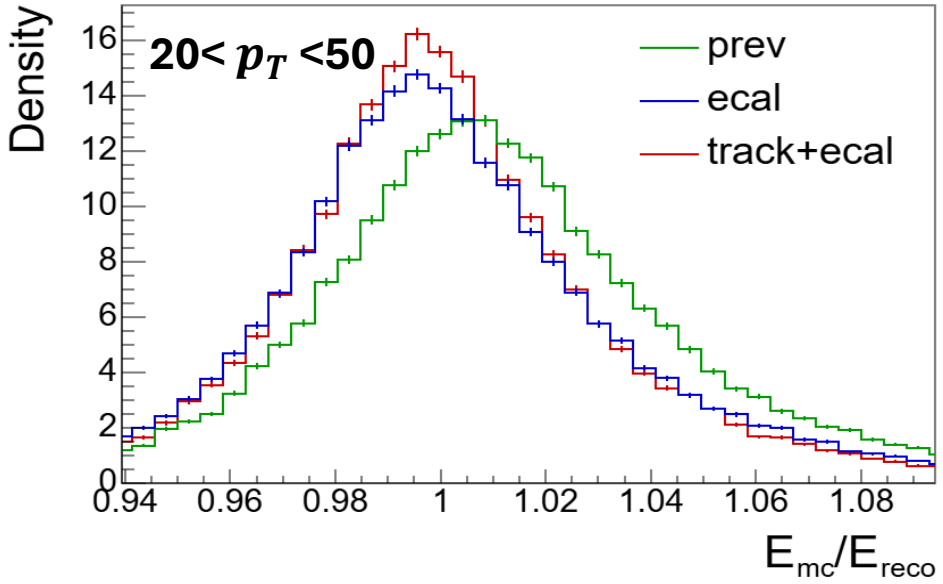


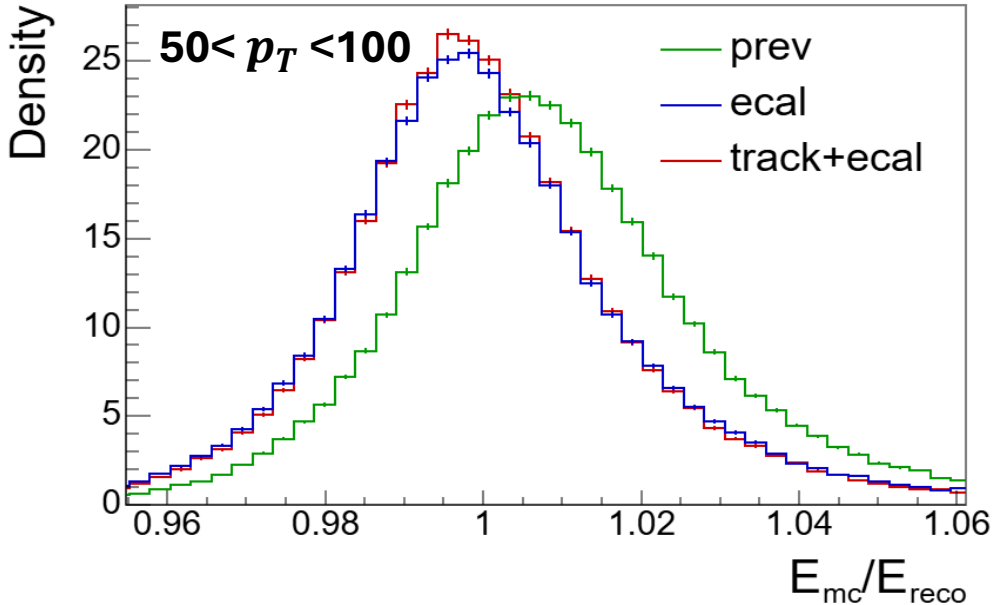
Endcaps (EB)

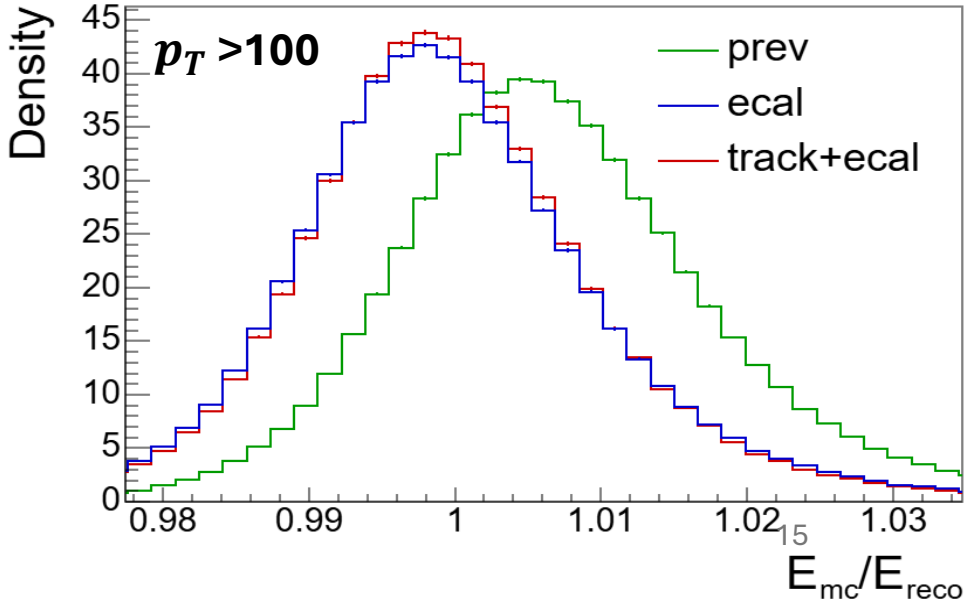


Lowest uncertainty results (BDT)









Conclusions and outlook

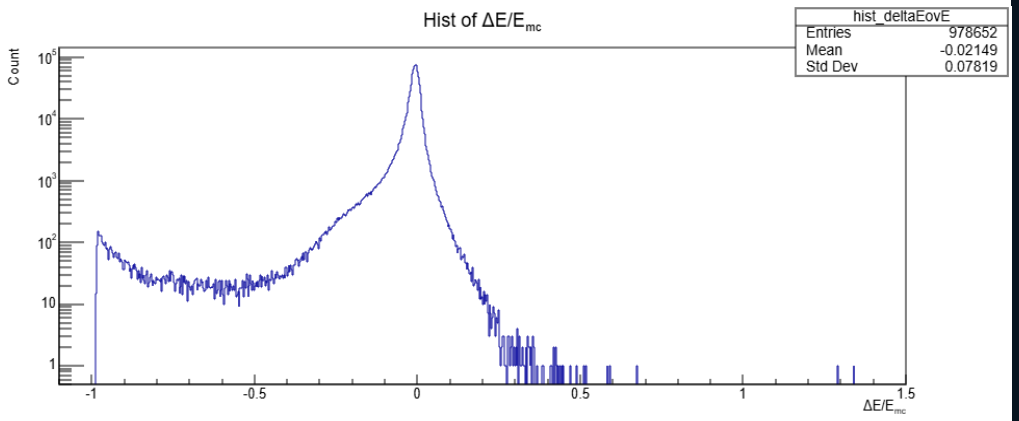
- Analysed MC e/γ sample and used ML to reduce the uncertainty on scouting data.
- Largest reduction in uncertainty observed for electrons in 5< p_T <20 GeV.
- Consistent reduction in energy uncertainty observed for electrons and photons through all energies.
- Electron/photons incident on super module gaps need to be addressed separately.
- CNN approach needs to be further optimised
  - Shower shape variables, e.g. *sieie,* seemed to outperform the CNN.

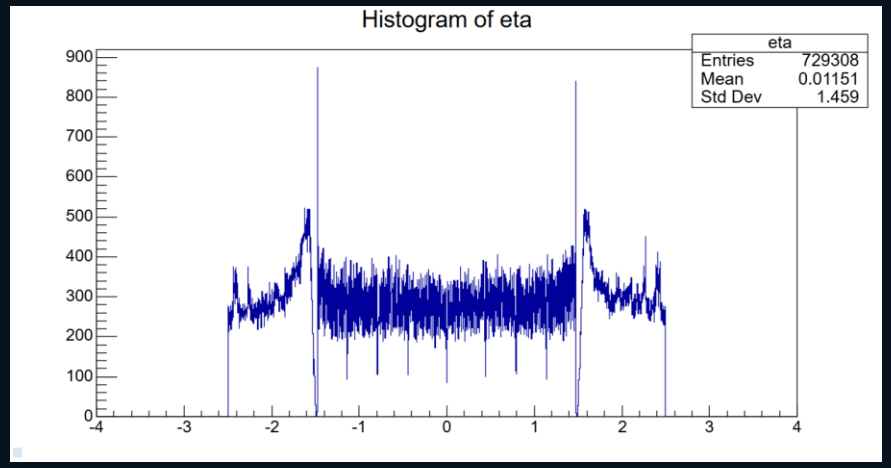
BACKUPS SLIDES

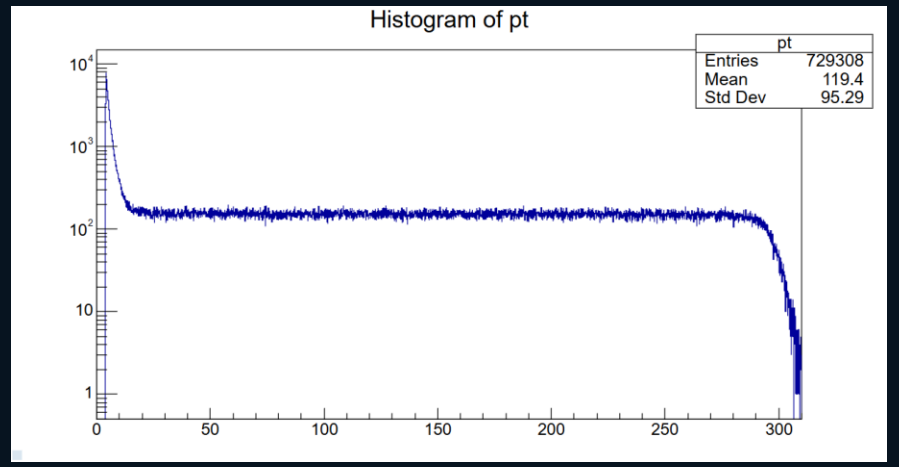
Target data

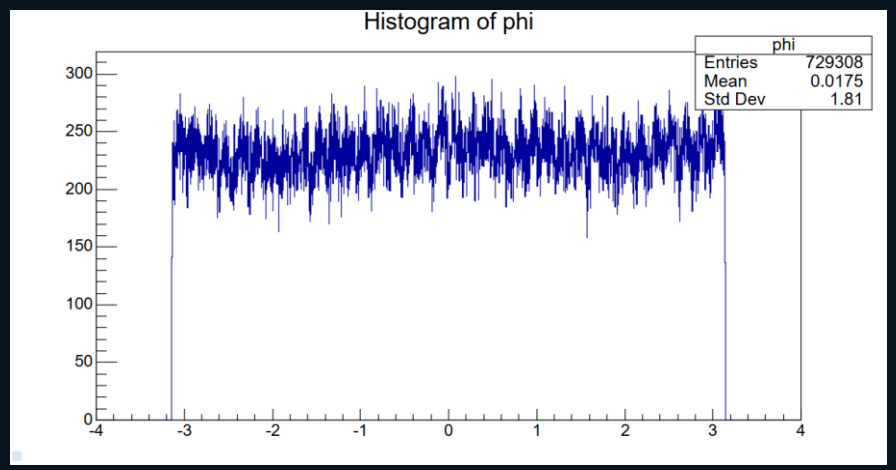
- Monte Carlo simulations produce generated particles
- Each particle consists of three kinematic variables:
  - ϕ : Azimuthal angle [-π, π]
  - η : Pseudo rapidity
  - p_T : Transverse momentum
- Can combine p_T and η to get energy:
  - $E_{MC} = \sqrt{(p_T \cosh \eta)^2 + m^2}$, the Monte-Carlo simulated energy

Preliminary predictions





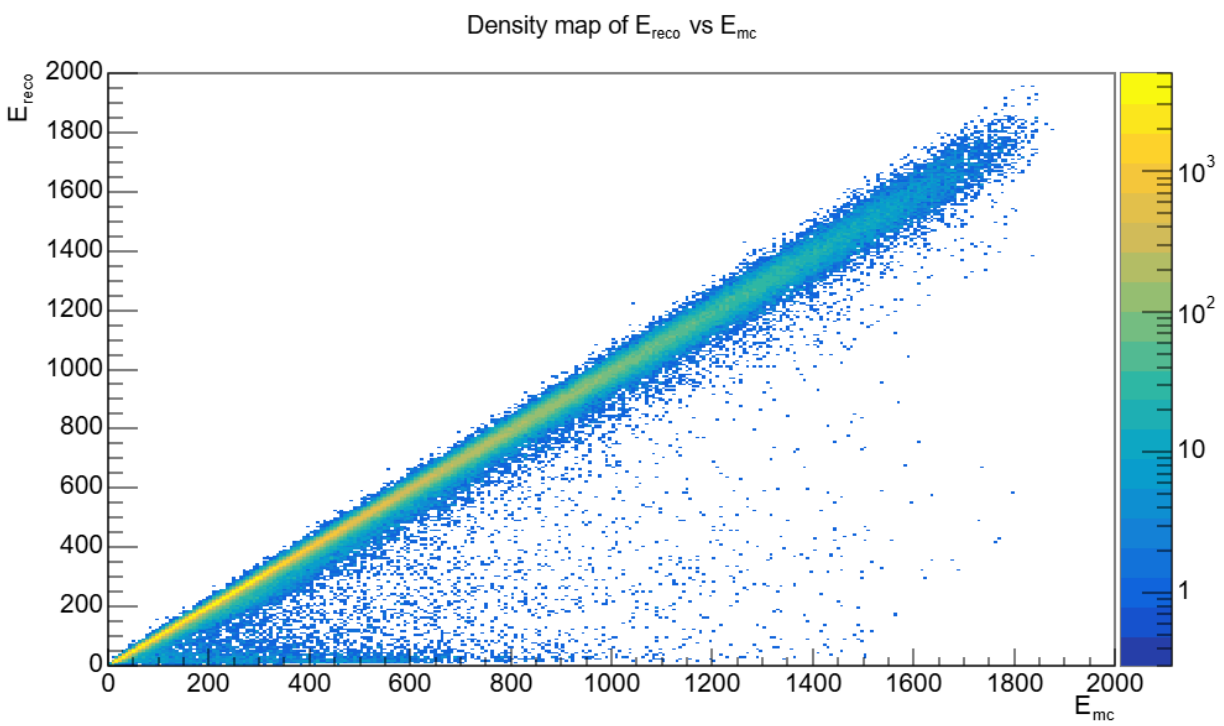


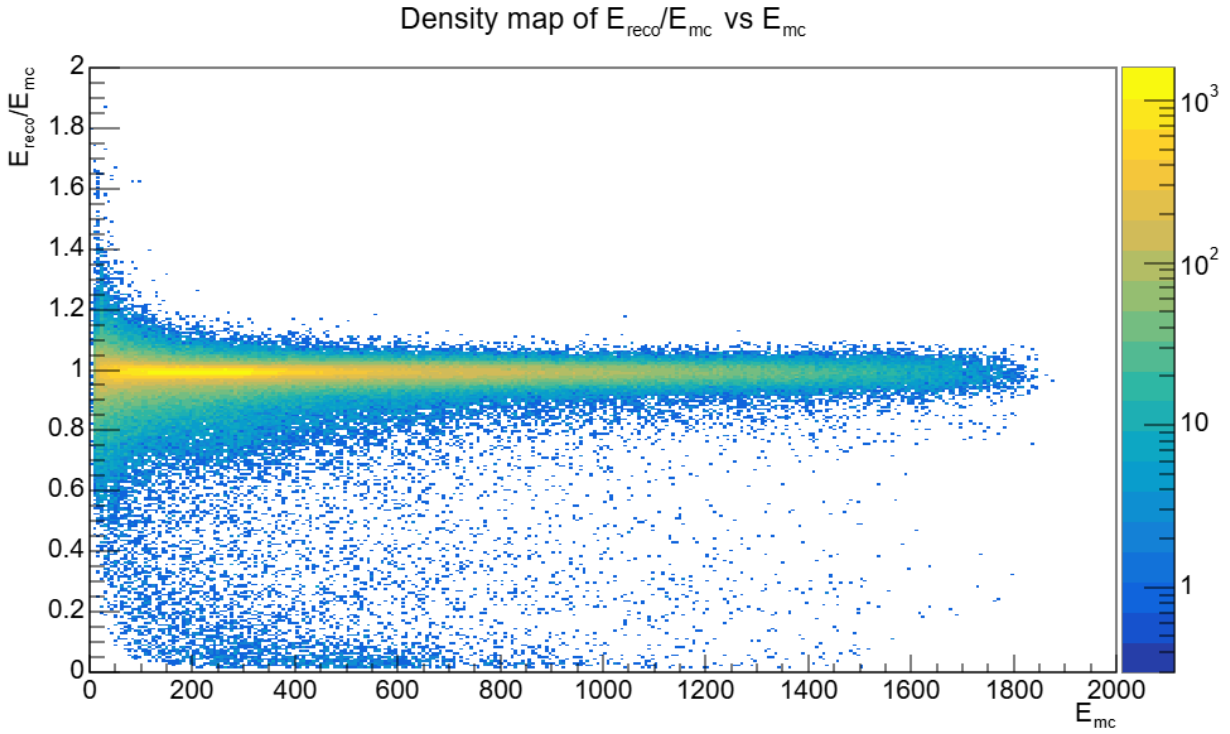


Investigating large error regions

- Instead of decaying, the left tail of the $\Delta E/E_{MC}$ distribution grows significantly.
- Suggests $E_{reco} = 0$, as $\frac{0-E_{MC}}{E_{MC}} = -1$.
- Critical to investigate the origin of this effect, as it may introduce systematic biases and degrade ML model performance.

E_{reco} vs E_{MC}

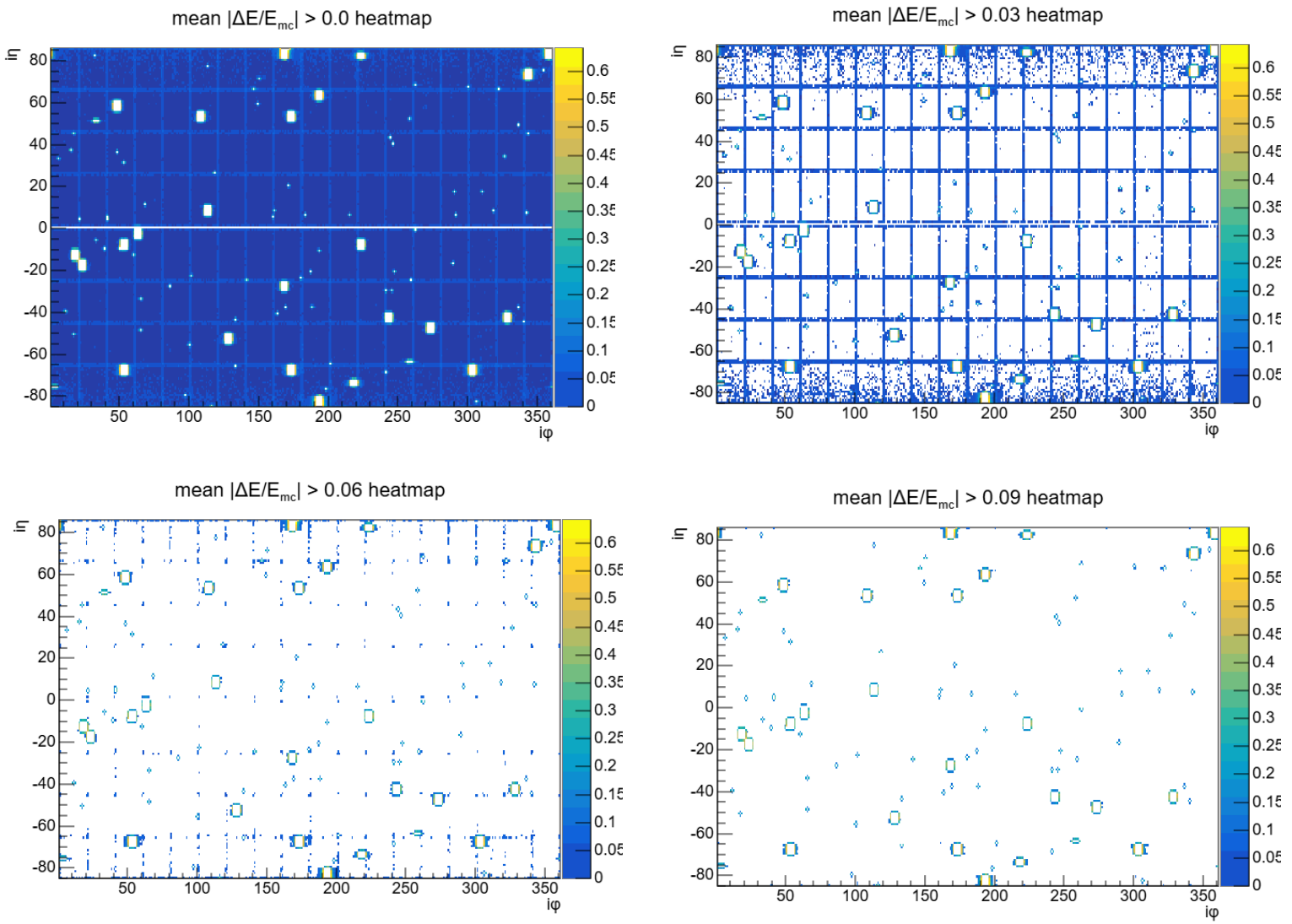

Density map of E_reco vs E_mc
E_reco
E_mc
0
200
400
600
800
1000
1200
1400
1600
1800
2000
1
10
10^2
10^3


Density map of E_reco/E_mc vs E_mc
E_reco/E_mc
E_mc
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
200
400
600
800
1000
1200
1400
1600
1800
2000
1
10
10^2
10^3

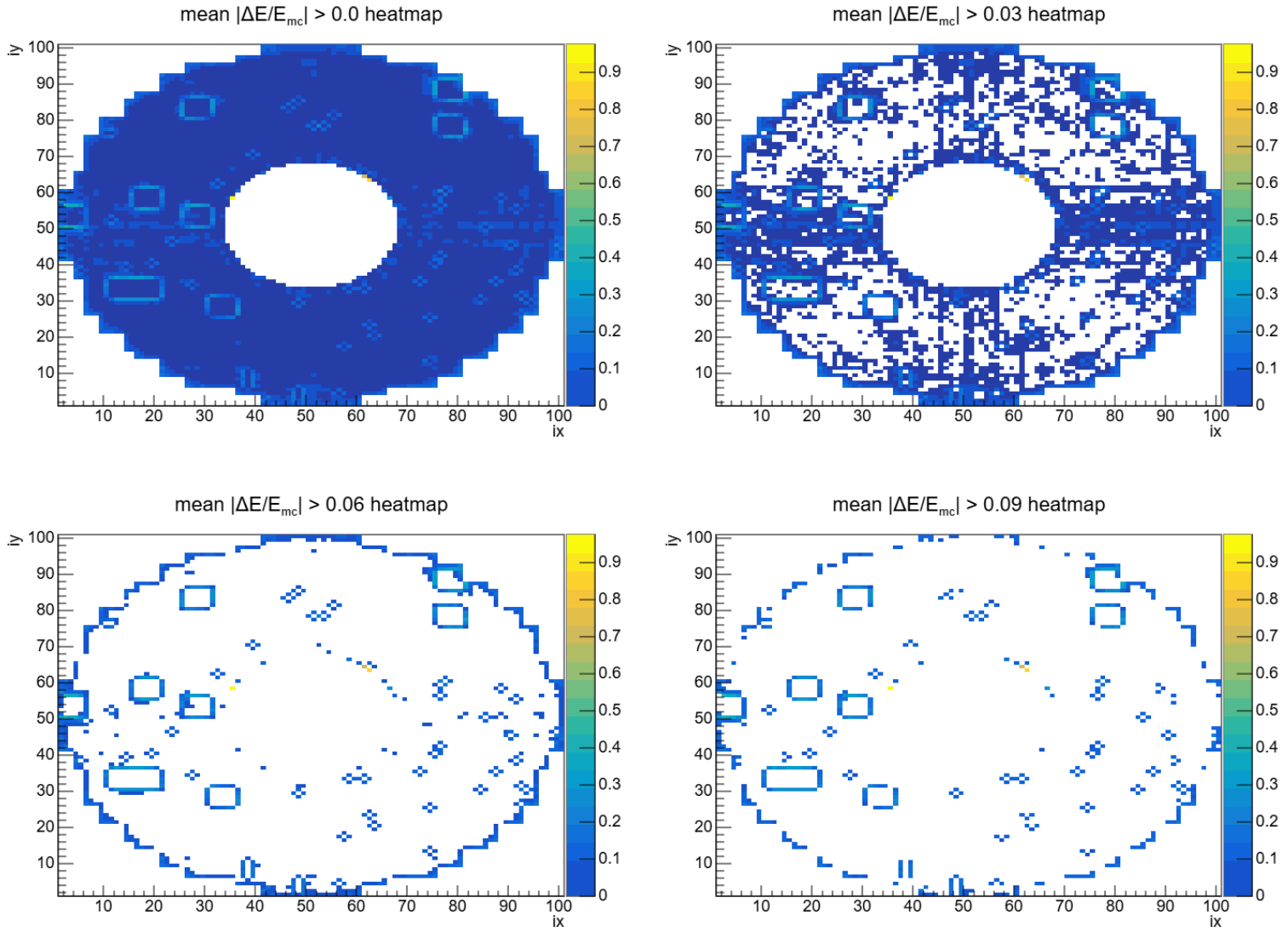
Error vs shower containment

- $(E_{5X5} - E_{MC})/E_{MC}$ plots
- 4 distinct regions
  - 1. Gaussian like distribution at higher containments
  - 2 . $y = x$ line region, i.e. E_{MC} = E_{5X5}
  - 3. High density region near (-1,-1)
  - 4. Spread of points with high error along the x-axis, irrespective of x values
- Region 1 is what we want the ML model to focus on; thus, cuts should be applied to eliminate regions 2-4.

Barrel (EB)

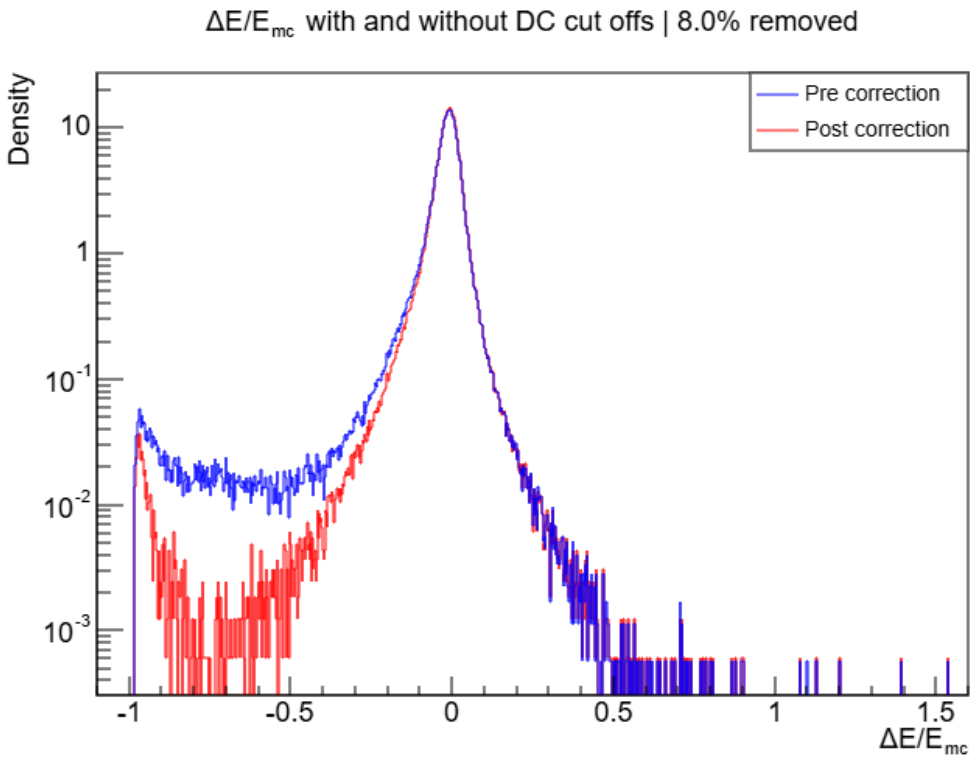

mean |ΔE/E_mc| > 0.0 heatmap
mean |ΔE/E_mc| > 0.03 heatmap
mean |ΔE/E_mc| > 0.06 heatmap
mean |ΔE/E_mc| > 0.09 heatmap
iη
iφ

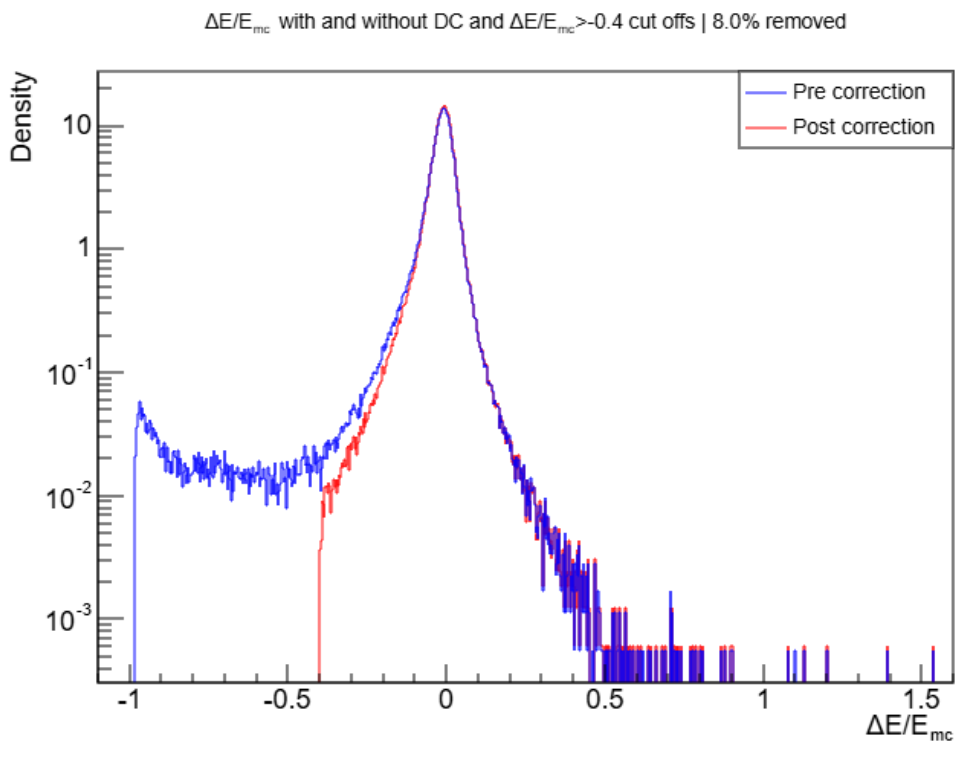
Endcaps (EE)


mean |ΔE/E_mc| > 0.0 heatmap
mean |ΔE/E_mc| > 0.03 heatmap
mean |ΔE/E_mc| > 0.06 heatmap
mean |ΔE/E_mc| > 0.09 heatmap
iy
ix

Endcap cutoffs

- Cuts in EE can be applied by:
  - Removing points near dead crystals and outer edges of the EE



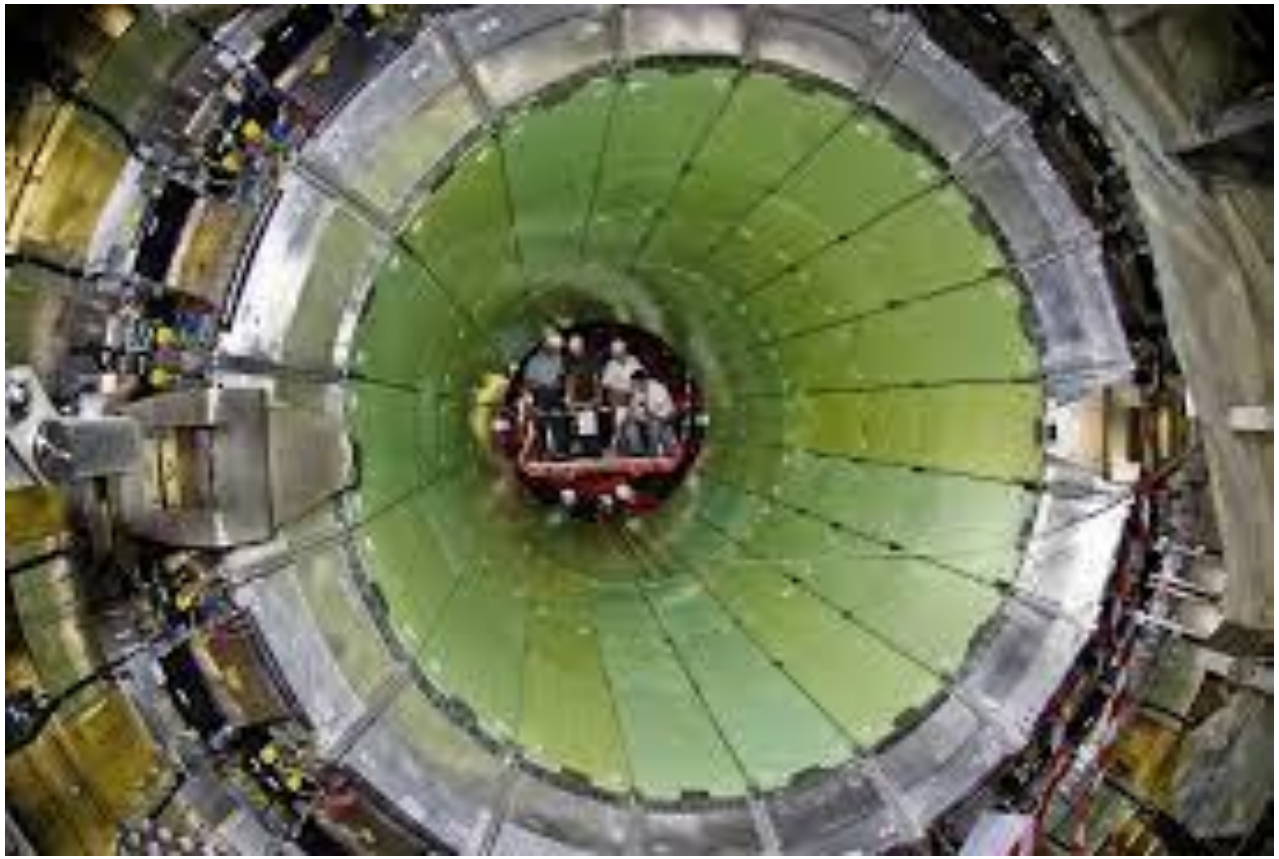


ML model – Data settings

- Data is separated into a training, validation and testing dataset:
  - Training: X samples
  - Validation: X samples
  - Testing: X samples
- Corrections applied:
  - Near-dead crystal pixels cutoff
  - Super-module transition in EB and outer EE edge cutoff
  - Sharp cutoff of: $-0.4 < \Delta E/E_{MC}$
- Separate models created for EE and EB

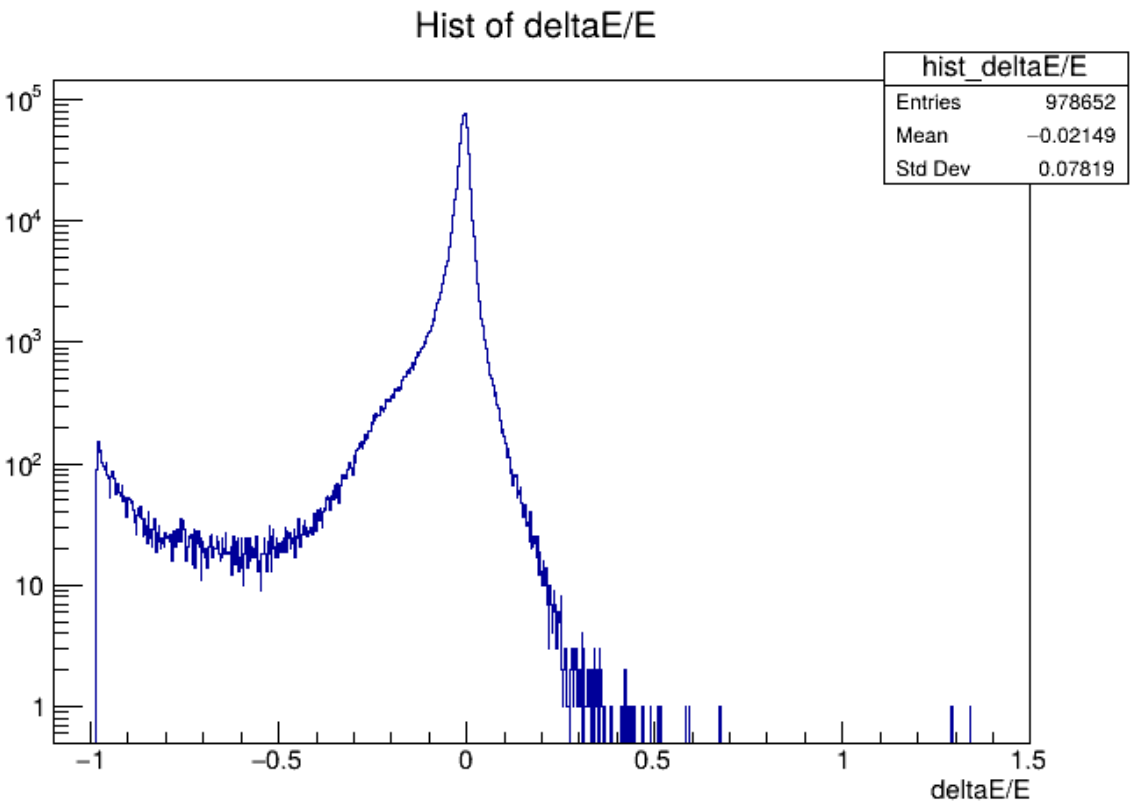
XGBoost (BDT)

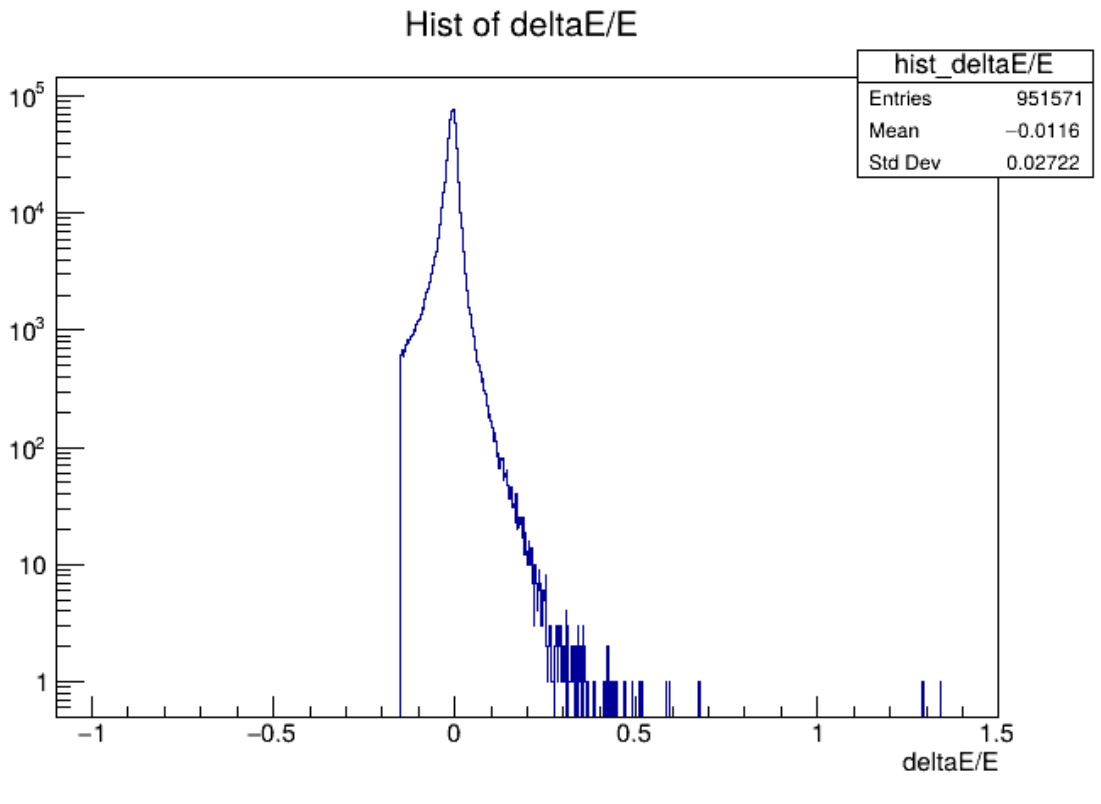
- List of input features:
  - Tracker:
    - detain
    - dphiin
    - fbrem
    - ooemoop
    - trkpt
    - trketa
    - trkphi
    - trkq
  - ECAL:
    - e1x5
    - e2nd
    - e2x5B, e2x5L, e2x5M,
    - e2x5R, e2x5T
    - e5x5
    - eB, eL, eR, eT
    - eMax
    - eta ieta
    - iphi phi
    - pse
    - pt
    - r9
    - sieie
- Labels:
  - $y_{true} = E_{MC}/E_{reco}$
- Loss:
  - $= (E_{pred} - E_{MC})/E_{MC})^2$
  - $= (\frac{y_{pred}}{y_{true}} - 1)^2$
- No normalization (not necessary for BDT)



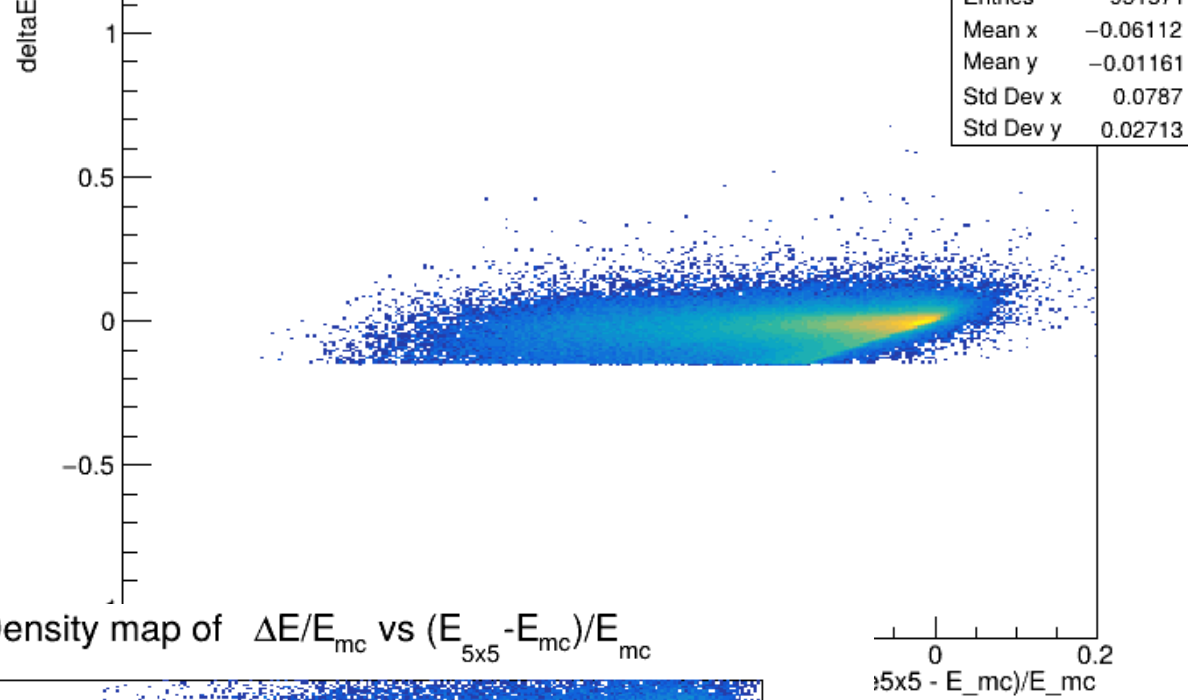
BACKUPS ON BACKUPS

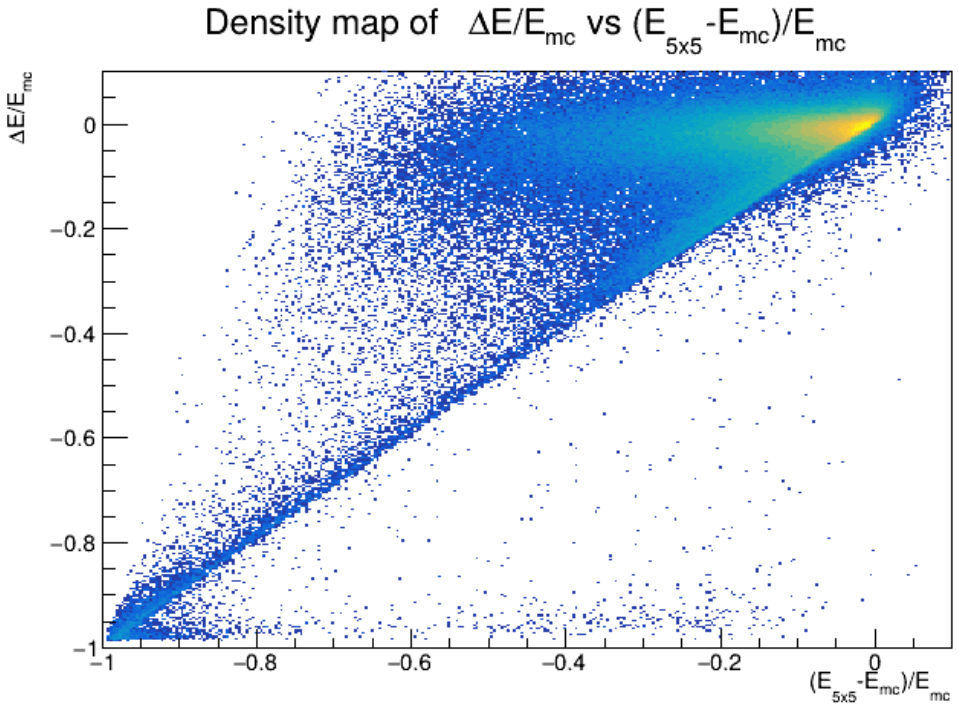
Pre vs post corrections

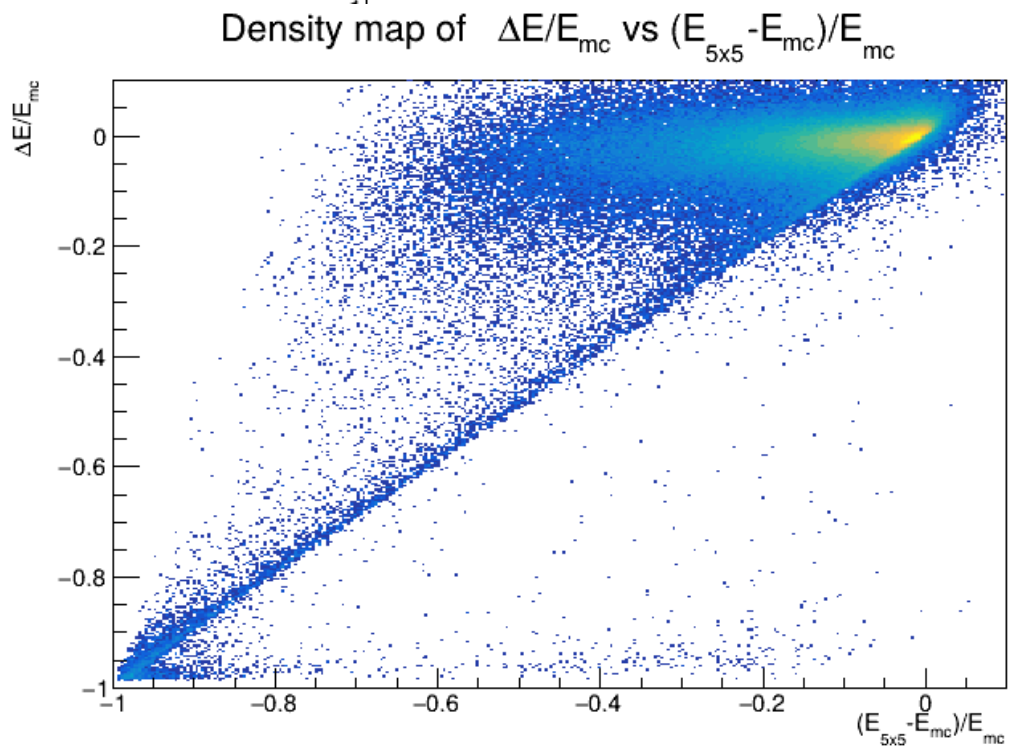




Pre vs post corrections

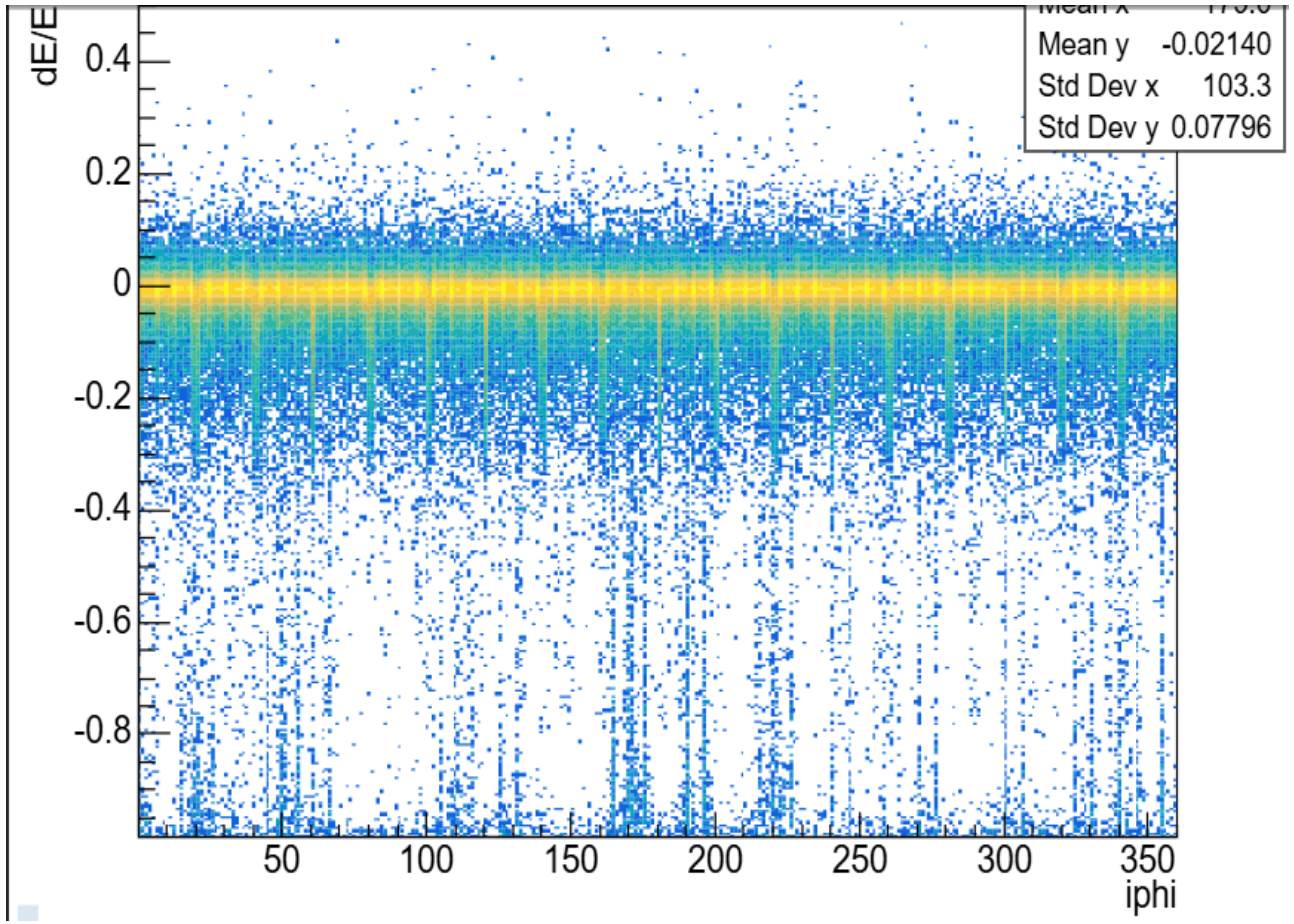


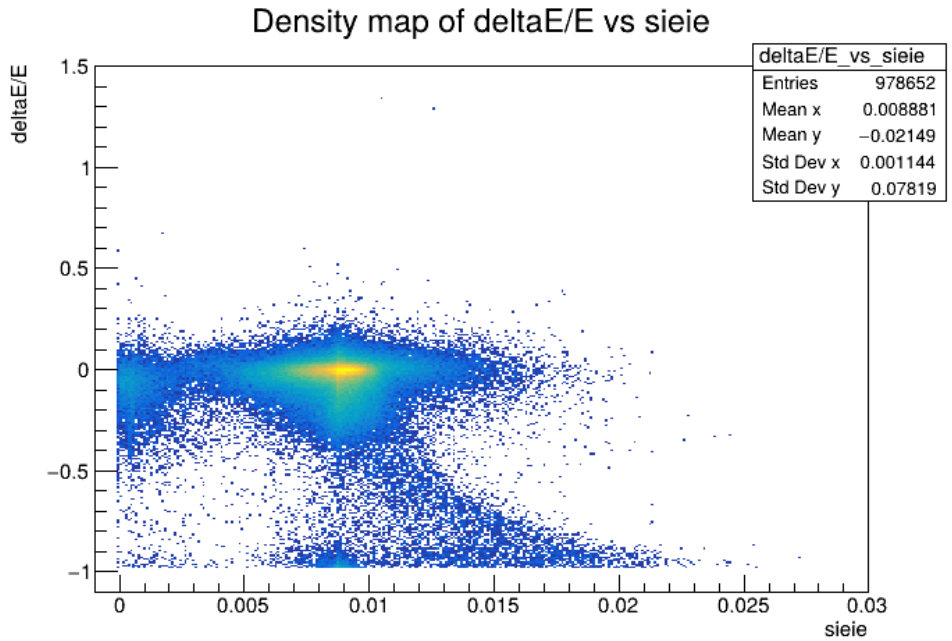


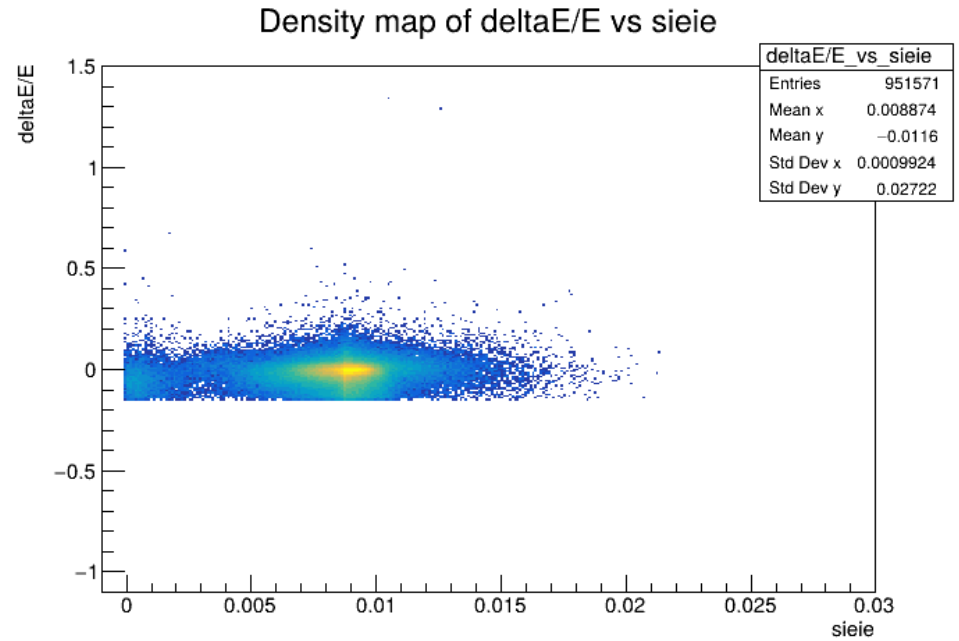


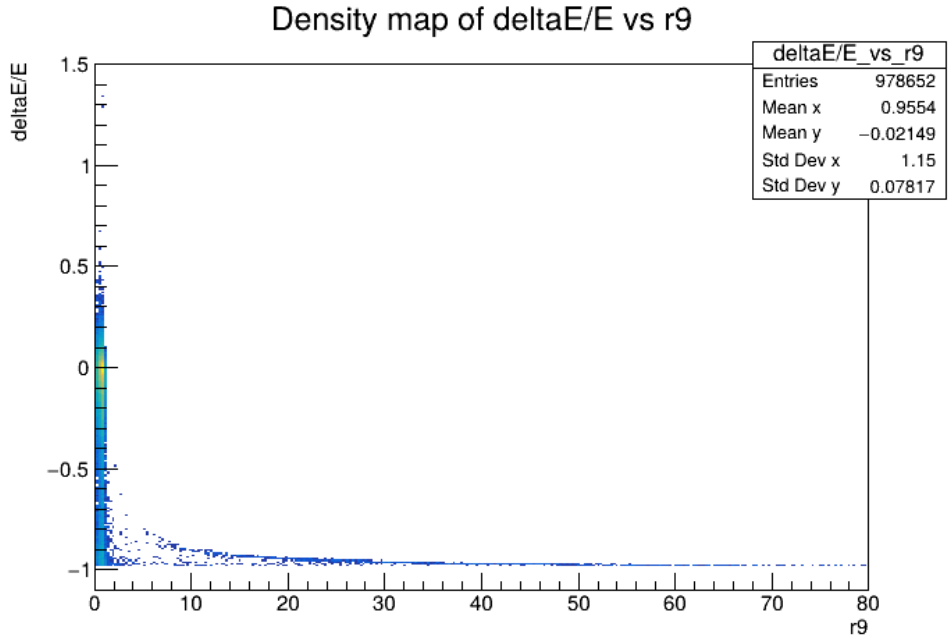
Investigating errors

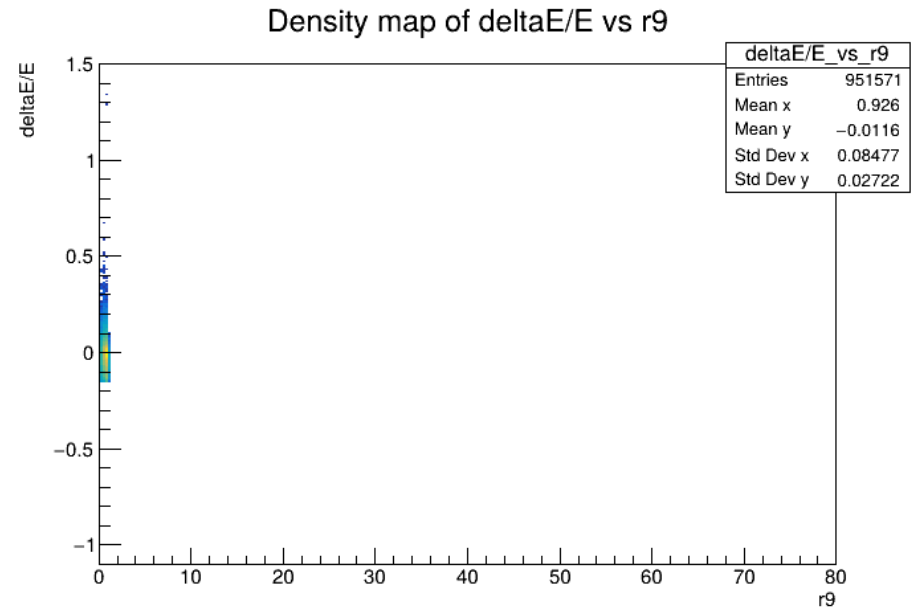
- Plot of err vs iphi:
  - 18 Fringes – 18 Super modules

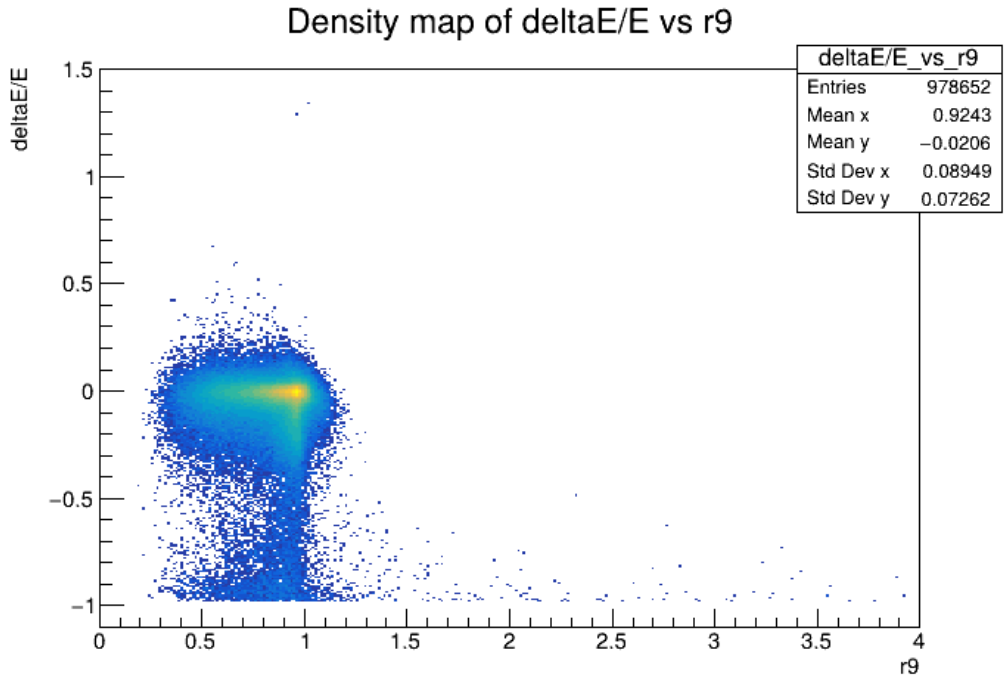


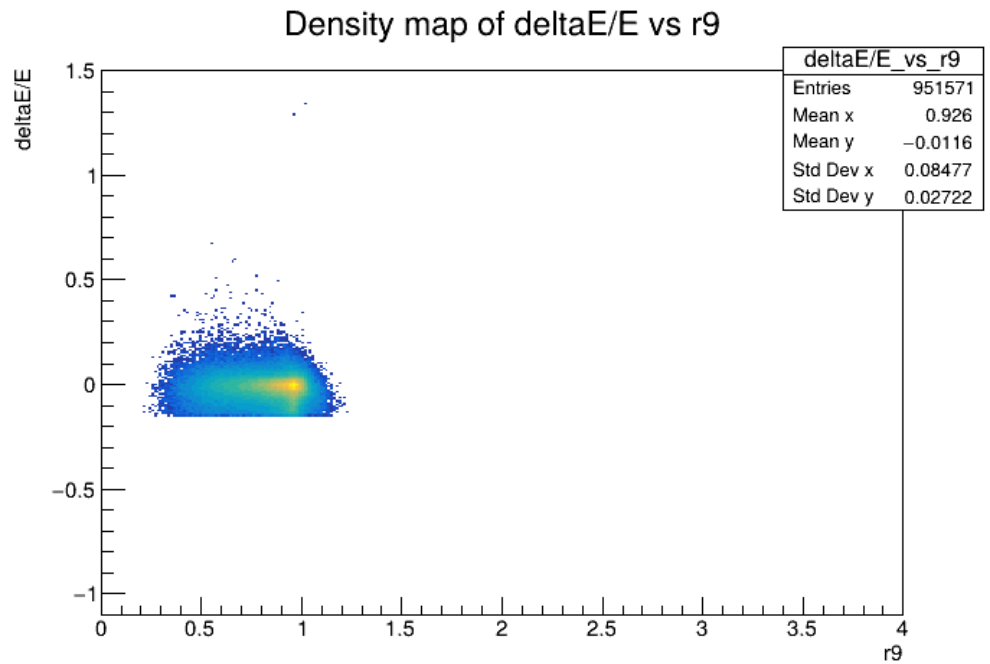

Density map of deltaE/E vs sieie
deltaE/E_vs_sieie
Entries 978652
Mean x 0.008881
Mean y −0.02149
Std Dev x 0.001144
Std Dev y 0.07819
deltaE/E
sieie

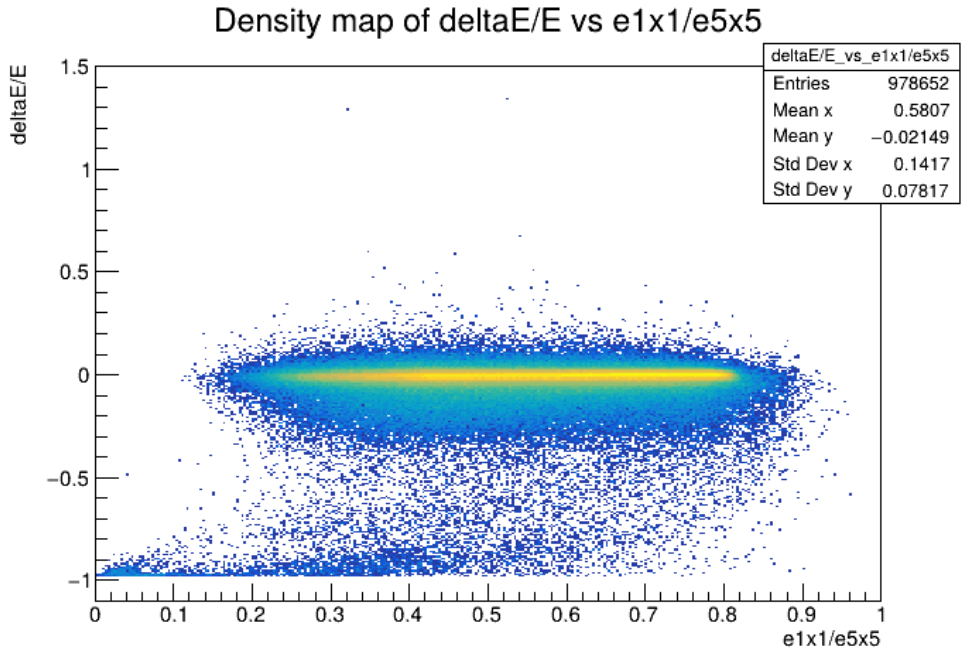

Density map of deltaE/E vs sieie
deltaE/E_vs_sieie
Entries 951571
Mean x 0.008874
Mean y −0.0116
Std Dev x 0.0009924
Std Dev y 0.02722
deltaE/E
sieie

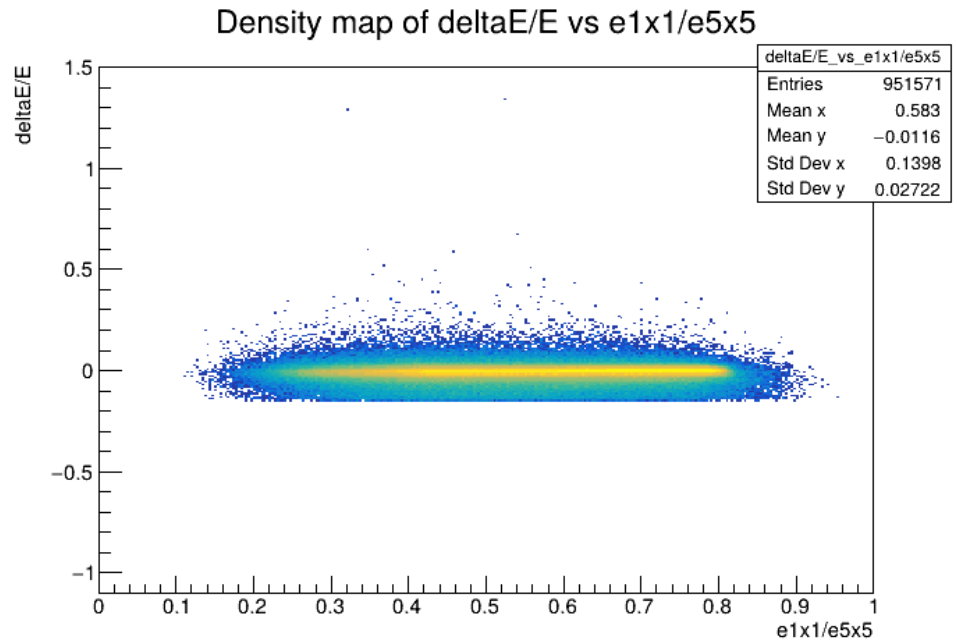

Density map of deltaE/E vs r9
deltaE/E_vs_r9
Entries 978652
Mean x 0.9554
Mean y −0.02149
Std Dev x 1.15
Std Dev y 0.07817
deltaE/E
r9

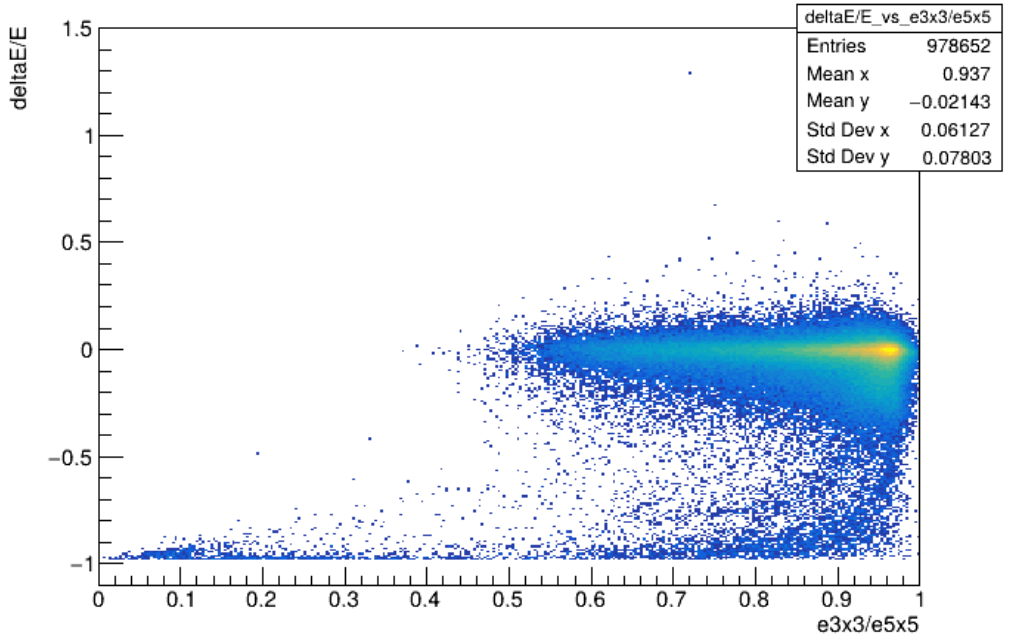

Density map of deltaE/E vs r9
deltaE/E_vs_r9
Entries 951571
Mean x 0.926
Mean y −0.0116
Std Dev x 0.08477
Std Dev y 0.02722
deltaE/E
r9

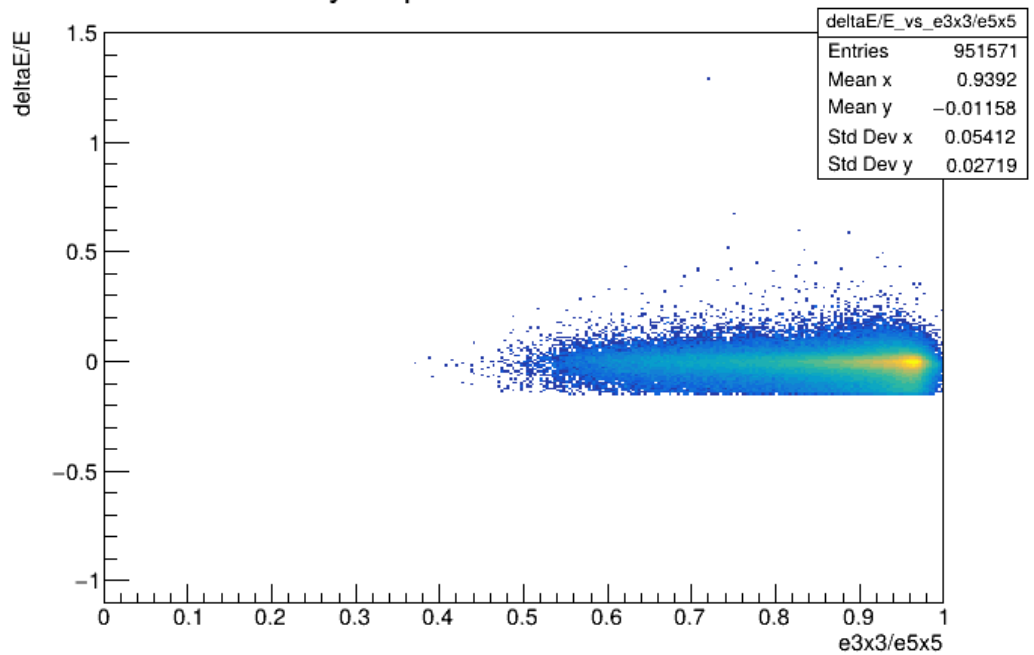

Density map of deltaE/E vs r9
deltaE/E_vs_r9
Entries 978652
Mean x 0.9243
Mean y −0.0206
Std Dev x 0.08949
Std Dev y 0.07262
deltaE/E
r9

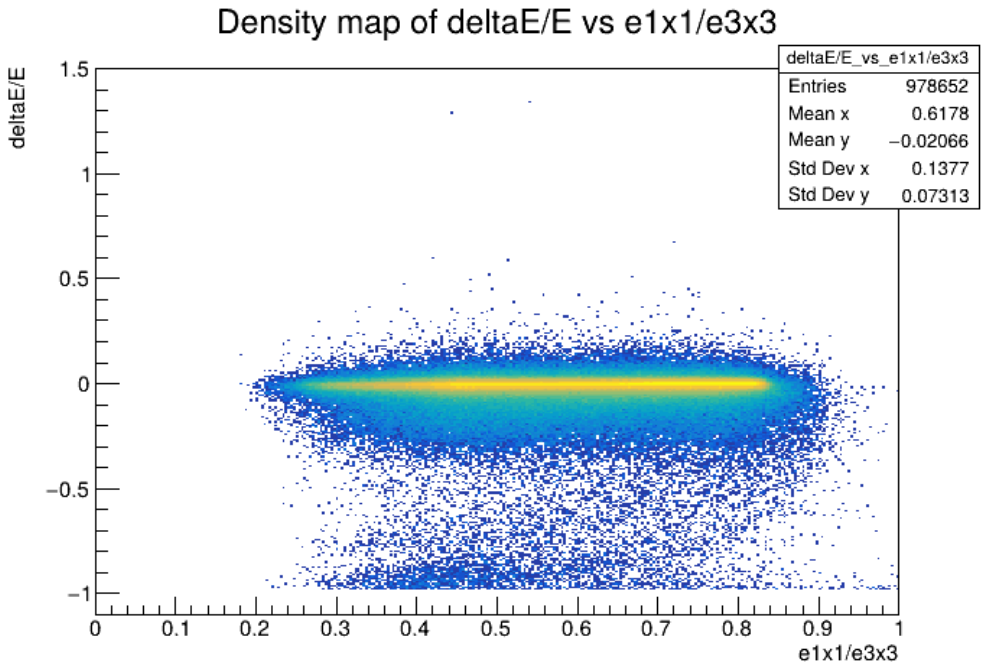

Density map of deltaE/E vs r9
deltaE/E_vs_r9
Entries 951571
Mean x 0.926
Mean y −0.0116
Std Dev x 0.08477
Std Dev y 0.02722
deltaE/E
r9

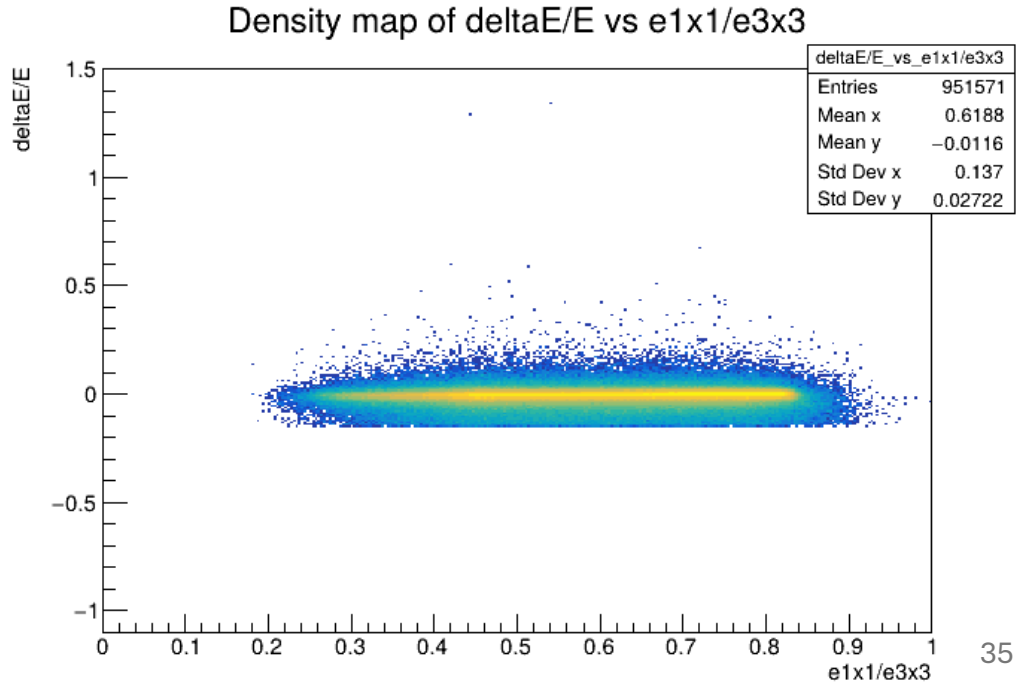

Density map of deltaE/E vs e1x1/e5x5
deltaE/E_vs_e1x1/e5x5
Entries 978652
Mean x 0.5807
Mean y −0.02149
Std Dev x 0.1417
Std Dev y 0.07817
deltaE/E
e1x1/e5x5


Density map of deltaE/E vs e1x1/e5x5
deltaE/E_vs_e1x1/e5x5
Entries 951571
Mean x 0.583
Mean y −0.0116
Std Dev x 0.1398
Std Dev y 0.02722
deltaE/E
e1x1/e5x5


Density map of deltaE/E vs e3x3/e5x5
deltaE/E_vs_e3x3/e5x5
Entries 978652
Mean x 0.937
Mean y −0.02143
Std Dev x 0.06127
Std Dev y 0.07803
deltaE/E
e3x3/e5x5

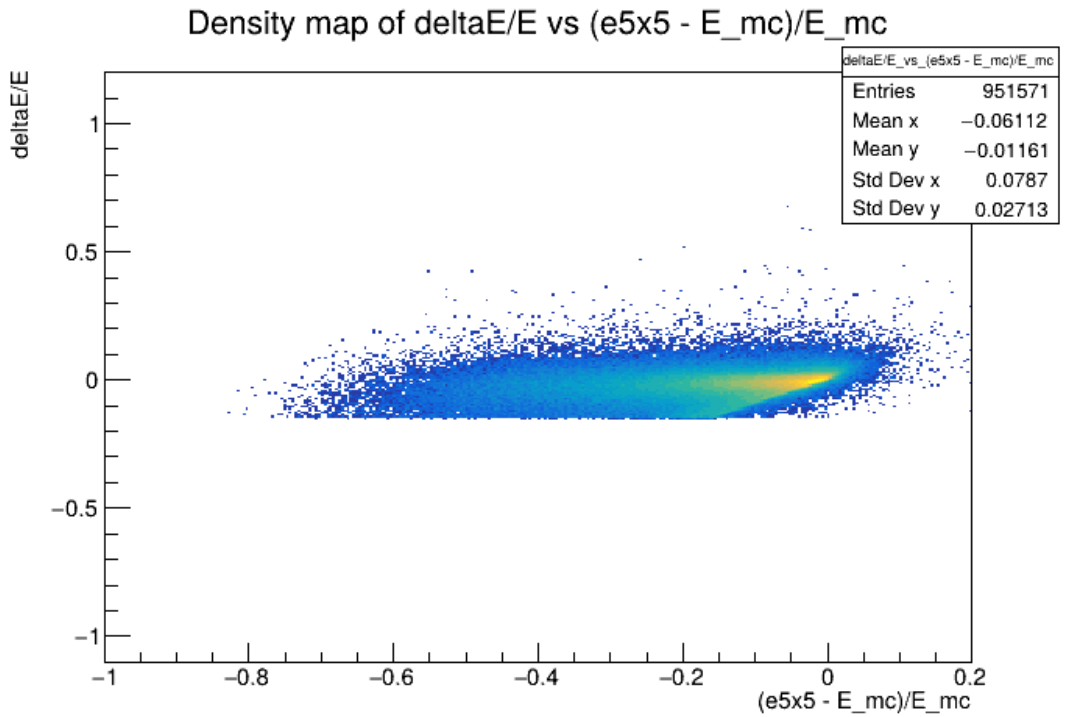

Density map of deltaE/E vs e3x3/e5x5
deltaE/E_vs_e3x3/e5x5
Entries 951571
Mean x 0.9392
Mean y −0.01158
Std Dev x 0.05412
Std Dev y 0.02719
deltaE/E
e3x3/e5x5

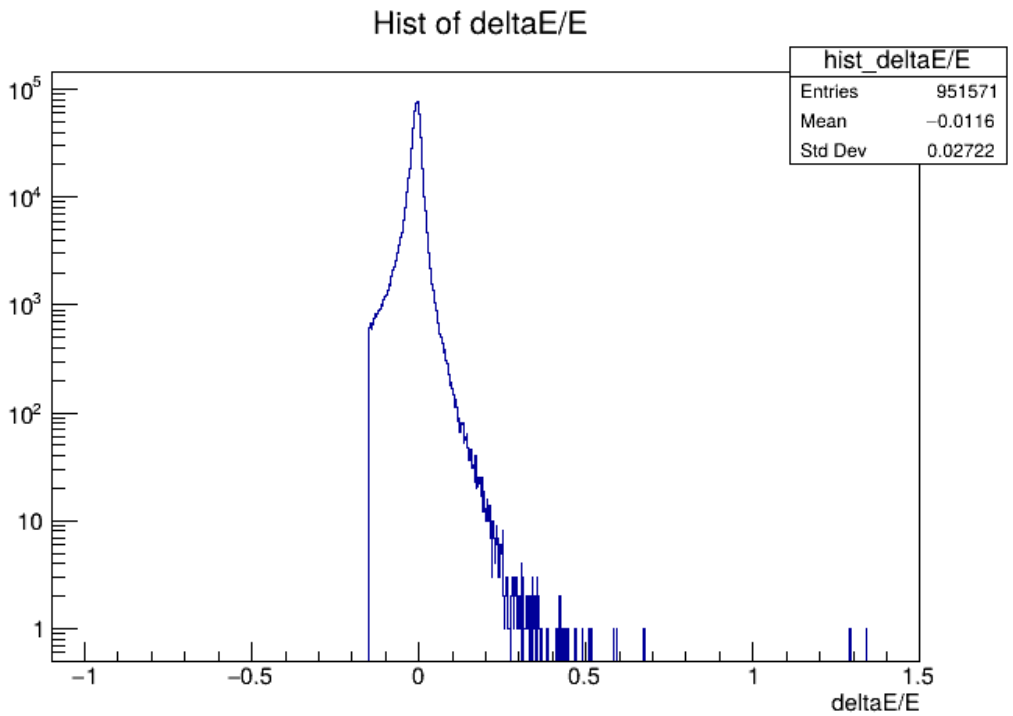

Density map of deltaE/E vs e1x1/e3x3
deltaE/E_vs_e1x1/e3x3
Entries 978652
Mean x 0.6178
Mean y −0.02066
Std Dev x 0.1377
Std Dev y 0.07313
deltaE/E
e1x1/e3x3


Density map of deltaE/E vs e1x1/e3x3
deltaE/E_vs_e1x1/e3x3
Entries 951571
Mean x 0.6188
Mean y −0.0116
Std Dev x 0.137
Std Dev y 0.02722
deltaE/E
e1x1/e3x3

Post Correction 1:

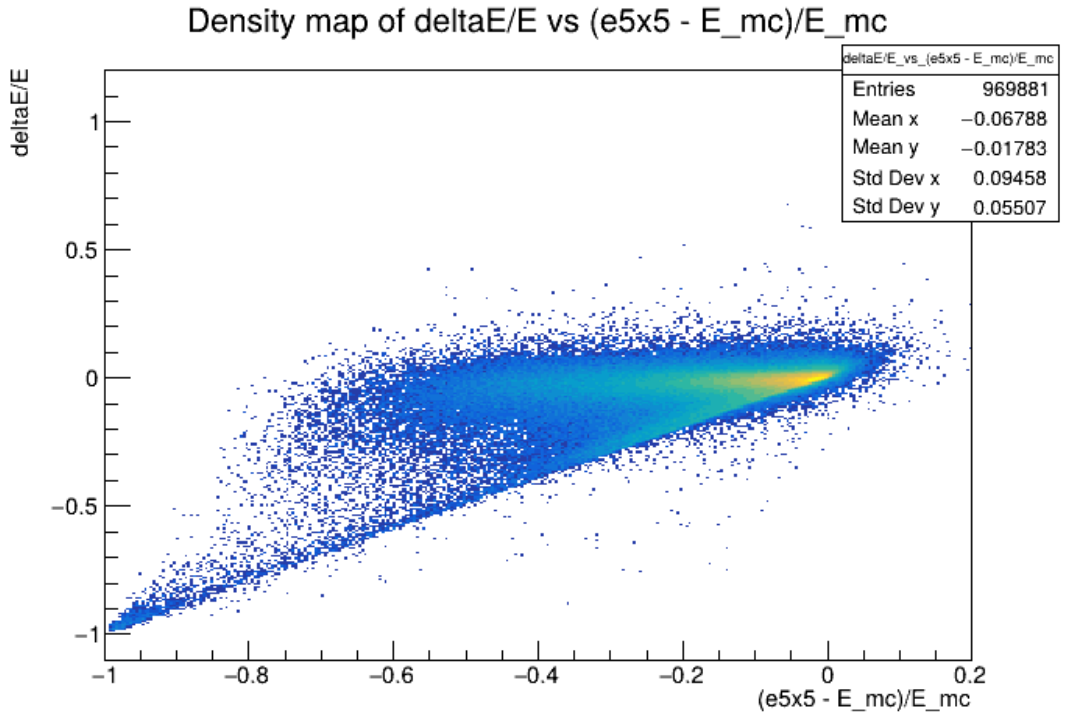
- E_reco > 0.85 * E_mc

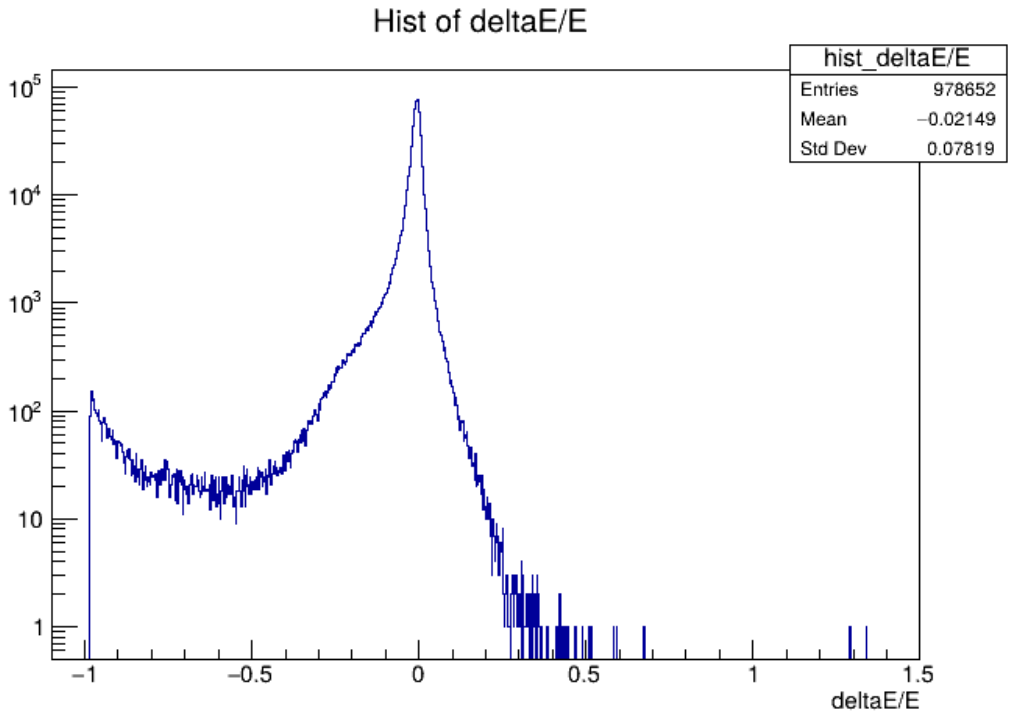


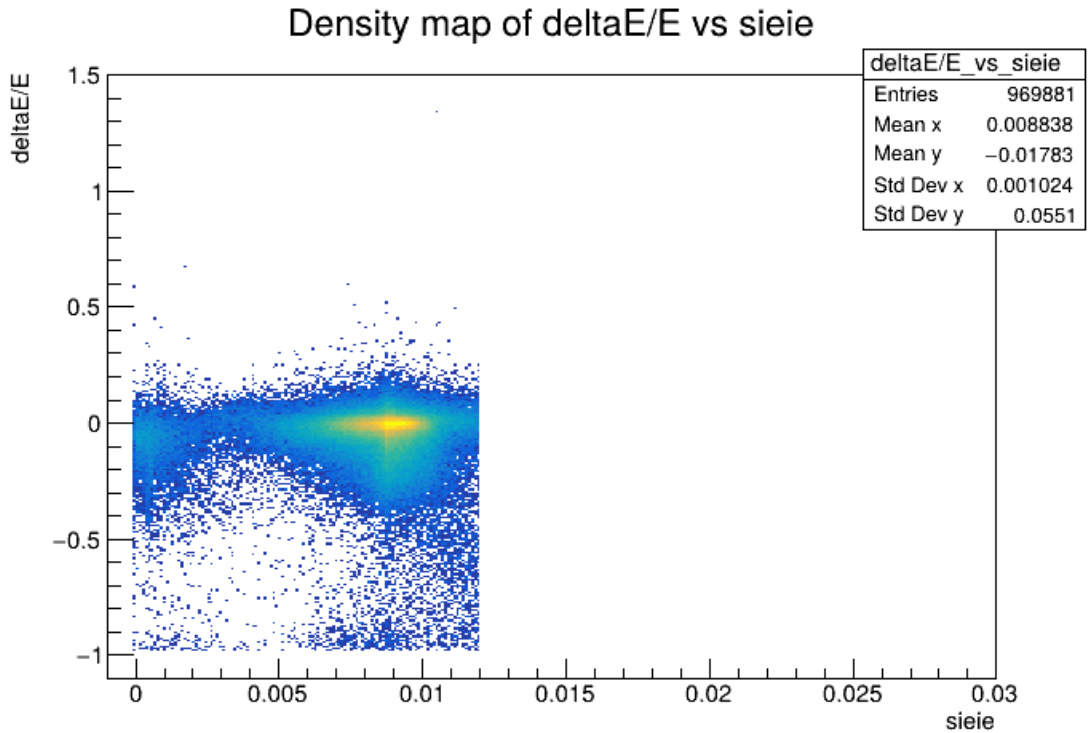


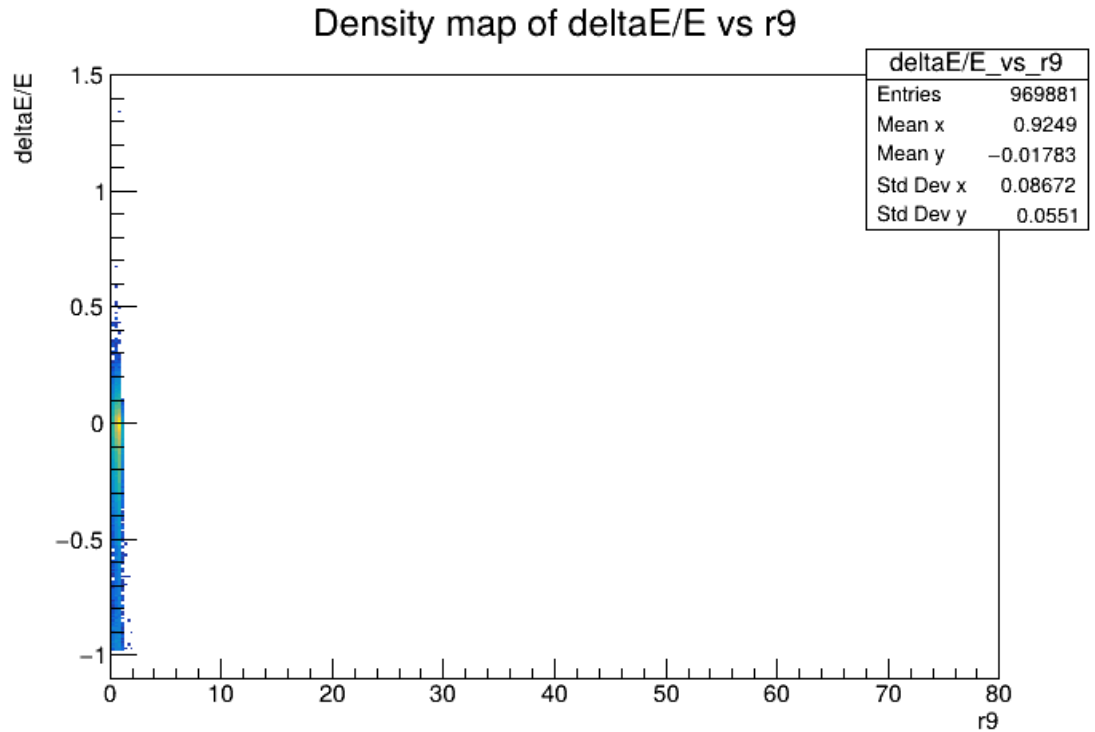
Post Correction 2:

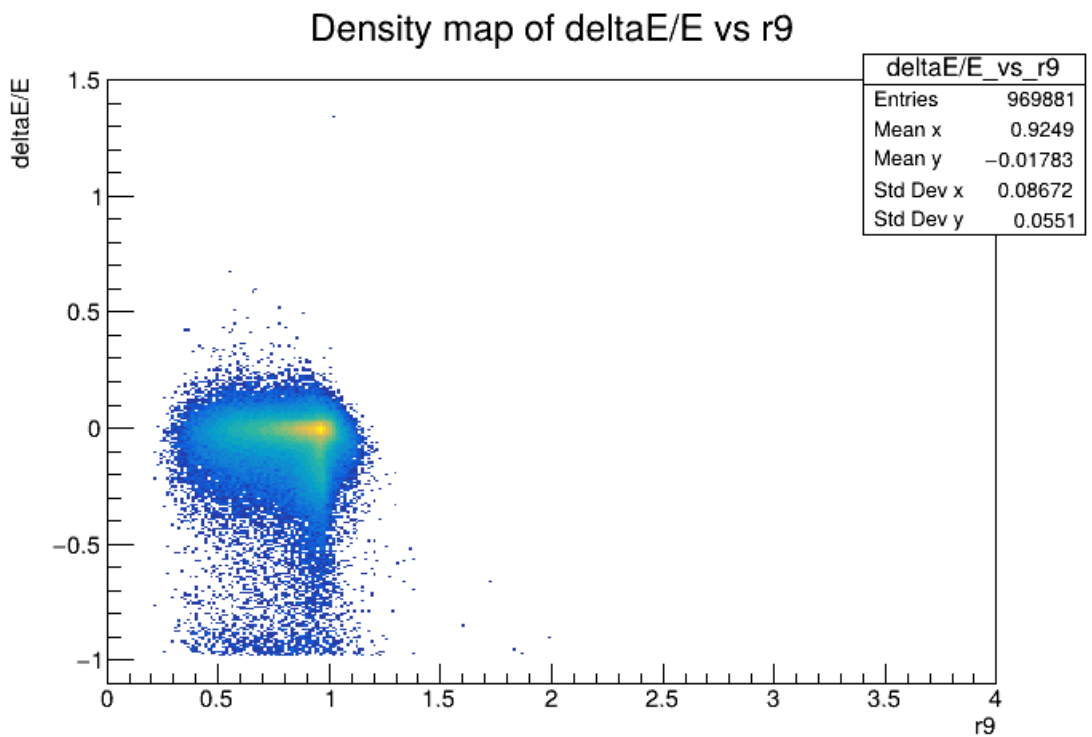
- e3x3/e5x5 > 0.5,
- emax/e3x3 < 1,
- r9 < 2,
- sieie < 0.012

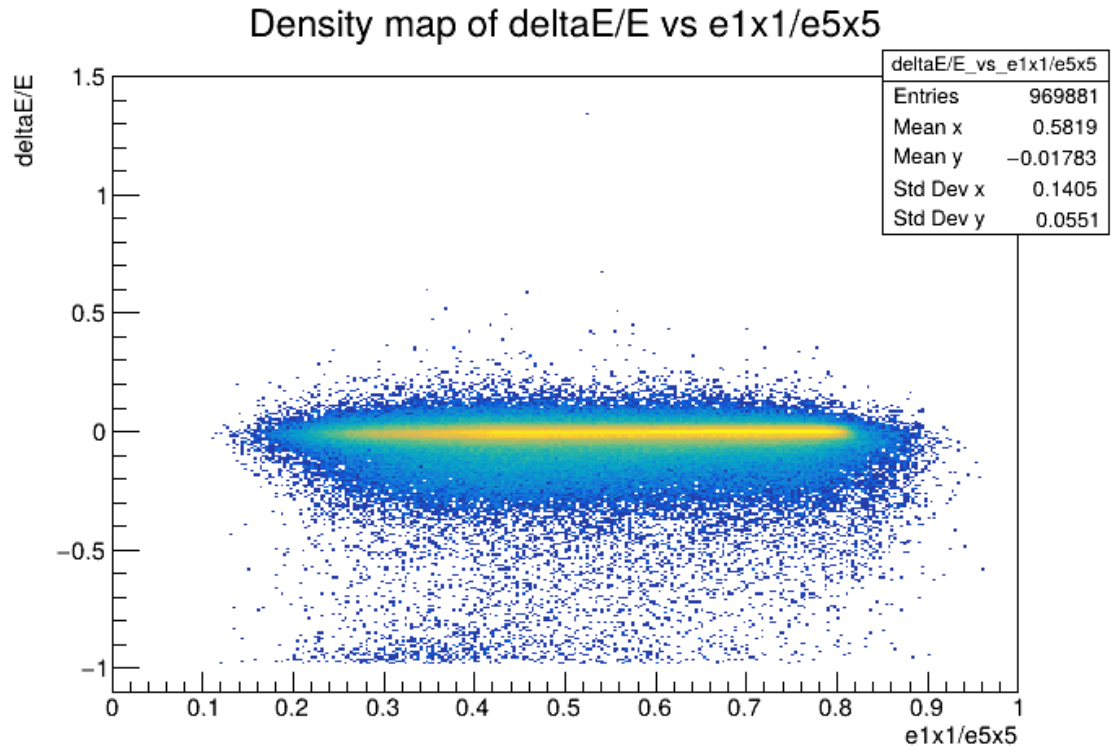


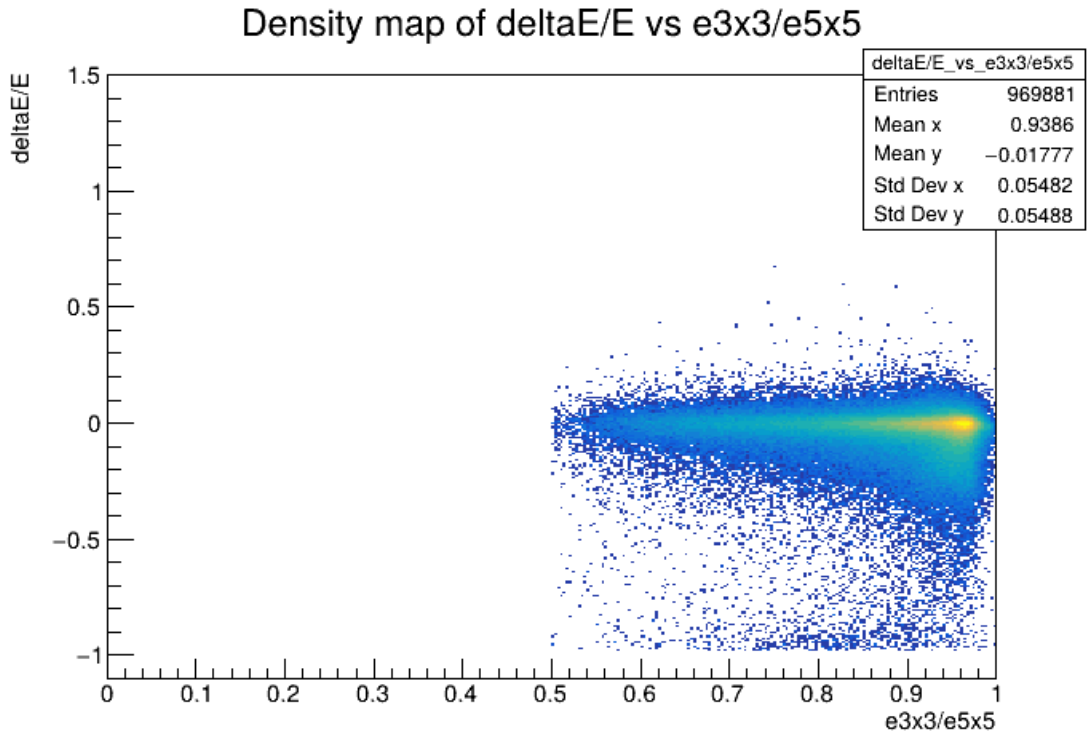


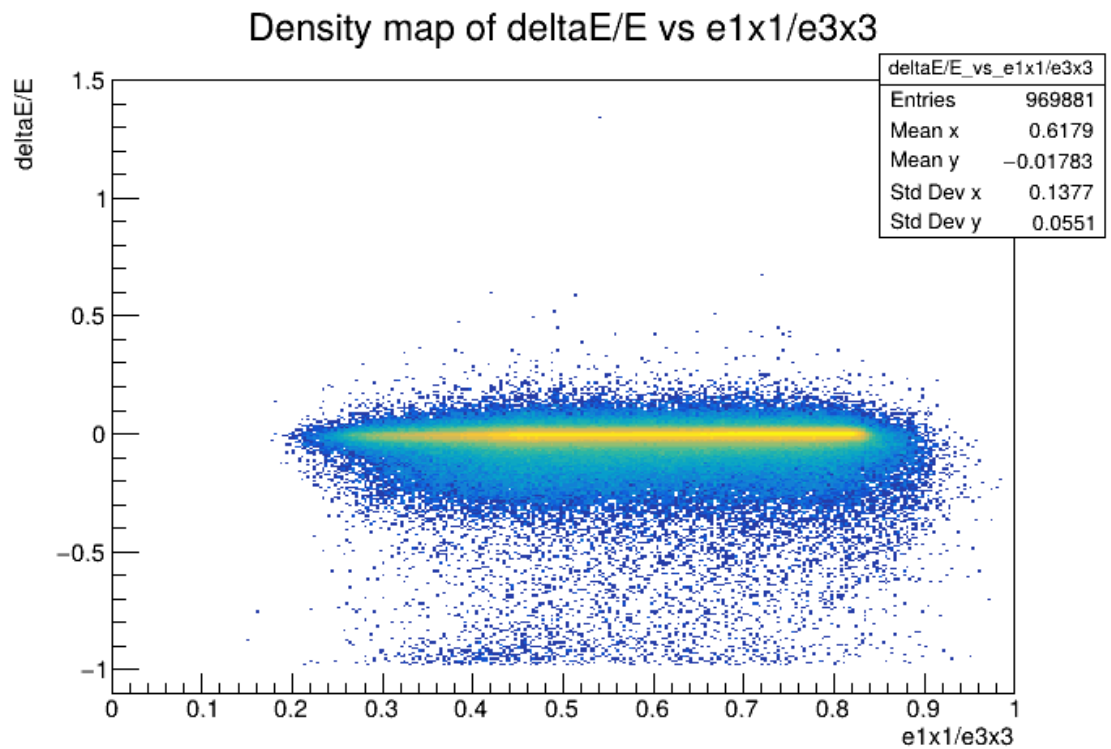

Density map of deltaE/E vs sieie
deltaE/E_vs_sieie
Entries 969881
Mean x 0.008838
Mean y −0.01783
Std Dev x 0.001024
Std Dev y 0.0551
deltaE/E
sieie
1.5
1
0.5
0
−0.5
−1
0
0.005
0.01
0.015
0.02
0.025
0.03

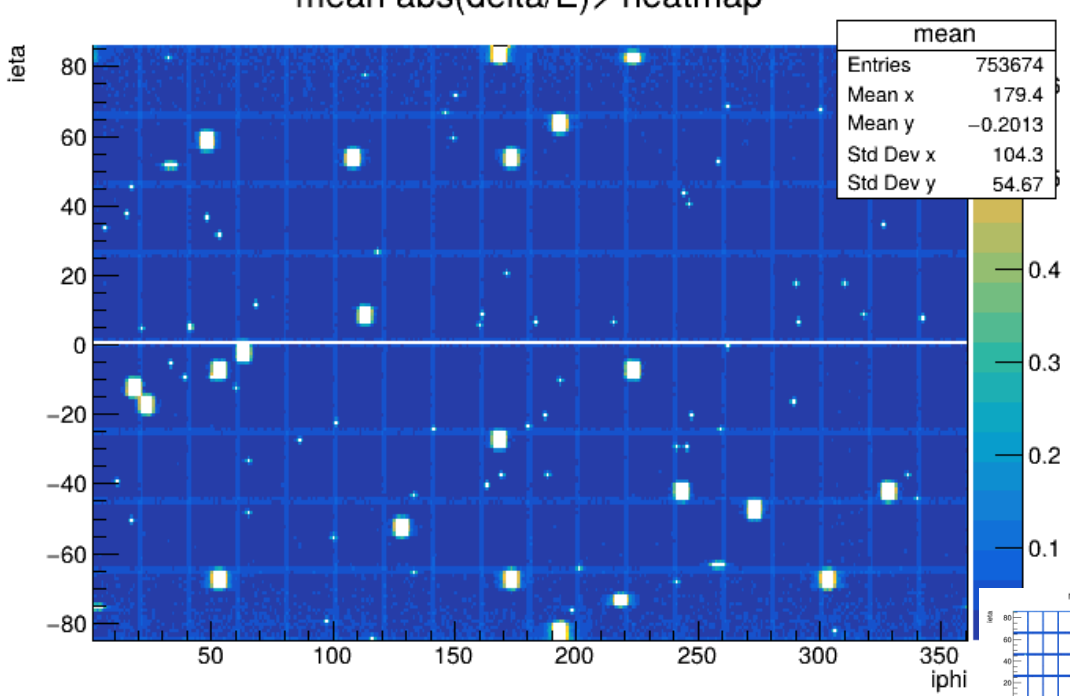

Density map of deltaE/E vs r9
deltaE/E_vs_r9
Entries 969881
Mean x 0.9249
Mean y −0.01783
Std Dev x 0.08672
Std Dev y 0.0551
deltaE/E
r9
1.5
1
0.5
0
−0.5
−1
0
10
20
30
40
50
60
70
80

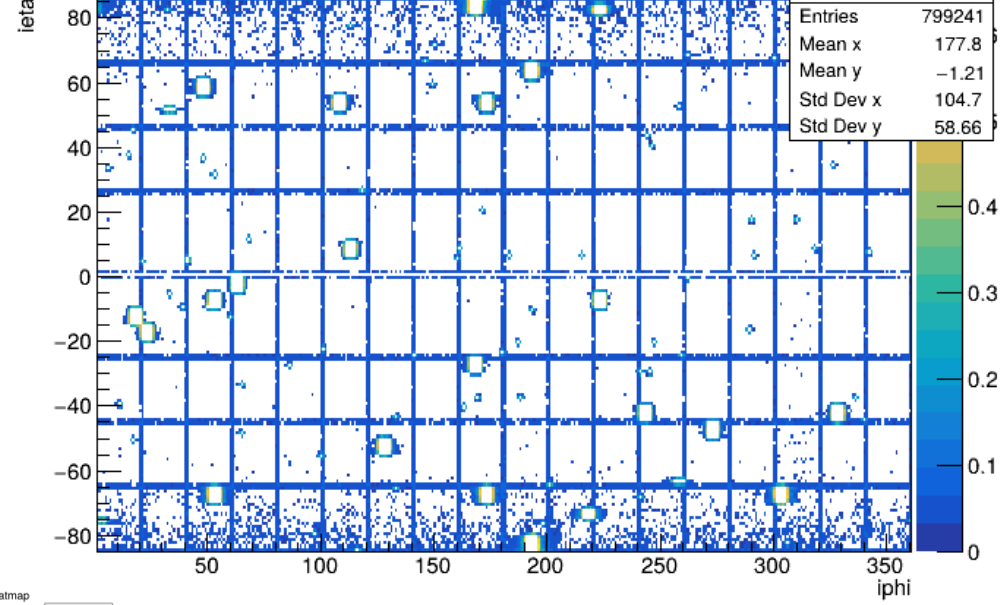

Density map of deltaE/E vs r9
deltaE/E_vs_r9
Entries 969881
Mean x 0.9249
Mean y −0.01783
Std Dev x 0.08672
Std Dev y 0.0551
deltaE/E
r9
1.5
1
0.5
0
−0.5
−1
0
0.5
1
1.5
2
2.5
3
3.5
4

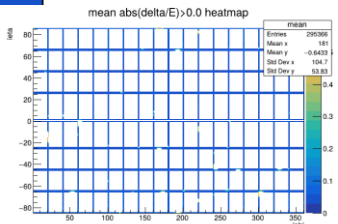

Density map of deltaE/E vs e1x1/e5x5
deltaE/E_vs_e1x1/e5x5
Entries 969881
Mean x 0.5819
Mean y −0.01783
Std Dev x 0.1405
Std Dev y 0.0551
deltaE/E
e1x1/e5x5
1.5
1
0.5
0
−0.5
−1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

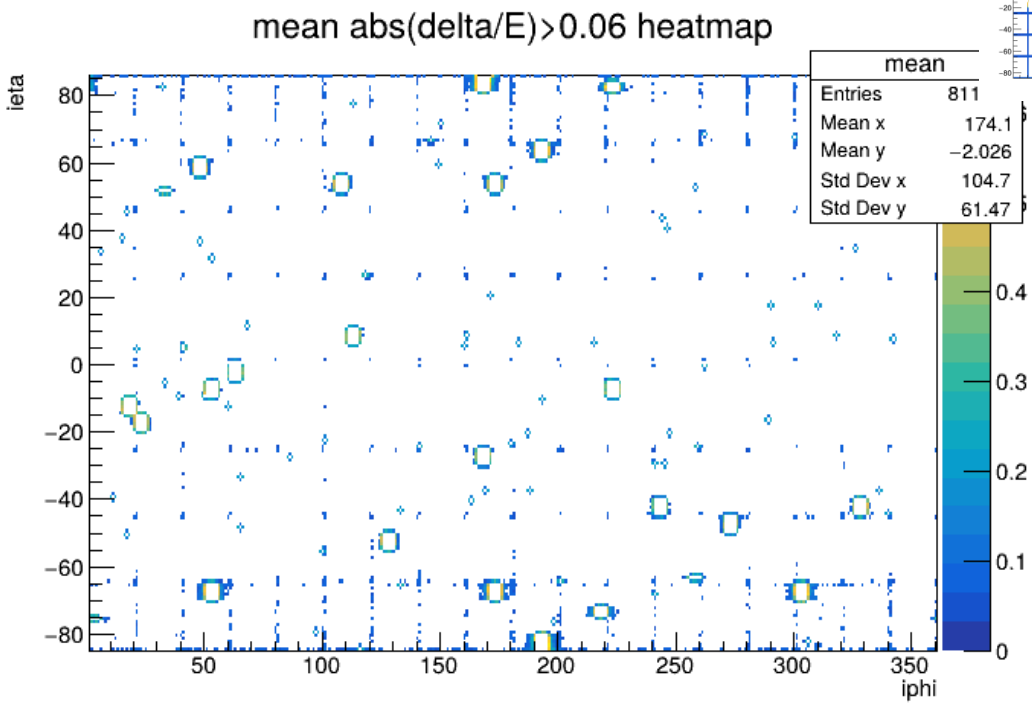

Density map of deltaE/E vs e3x3/e5x5
deltaE/E_vs_e3x3/e5x5
Entries 969881
Mean x 0.9386
Mean y −0.01777
Std Dev x 0.05482
Std Dev y 0.05488
deltaE/E
e3x3/e5x5

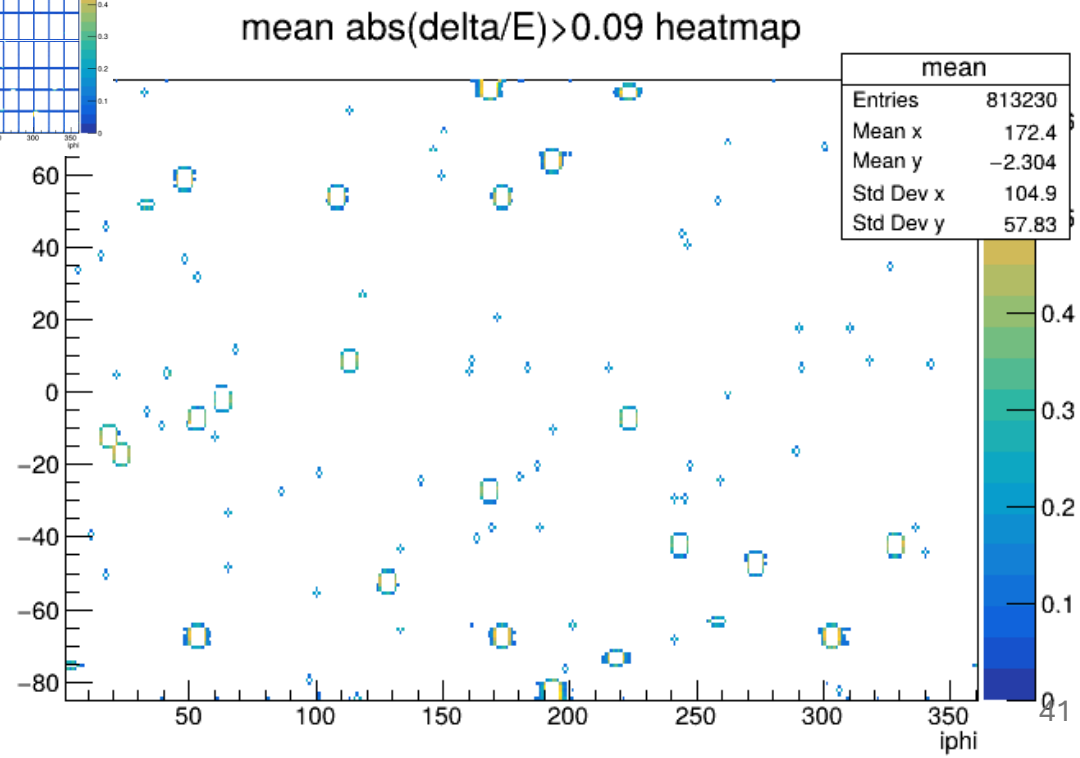

Density map of deltaE/E vs e1x1/e3x3
deltaE/E_vs_e1x1/e3x3
Entries 969881
Mean x 0.6179
Mean y −0.01783
Std Dev x 0.1377
Std Dev y 0.0551
deltaE/E
e1x1/e3x3

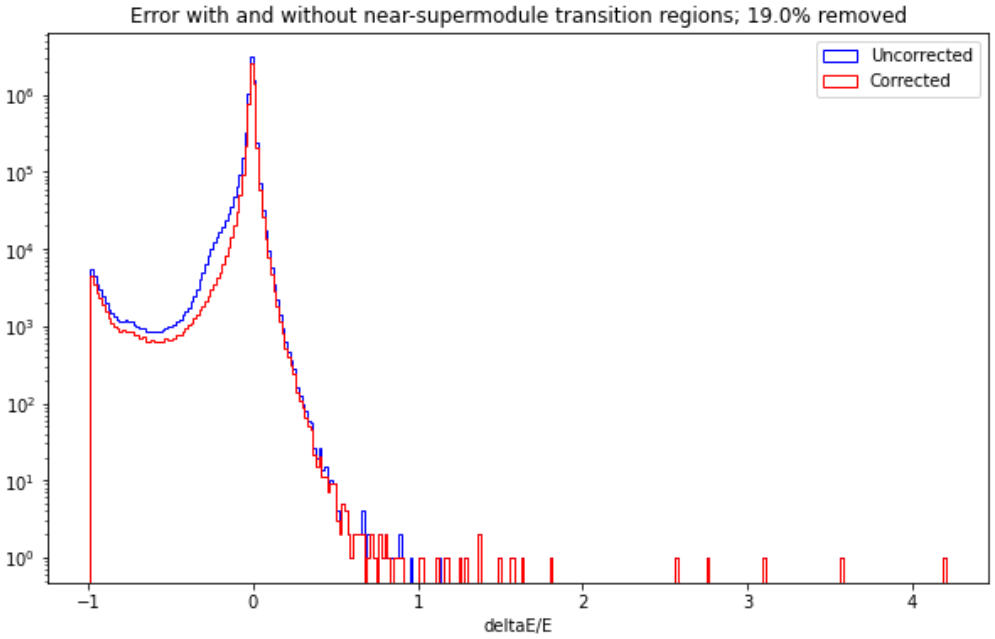

mean
Entries 753674
Mean x 179.4
Mean y −0.2013
Std Dev x 104.3
Std Dev y 54.67
ieta
iphi

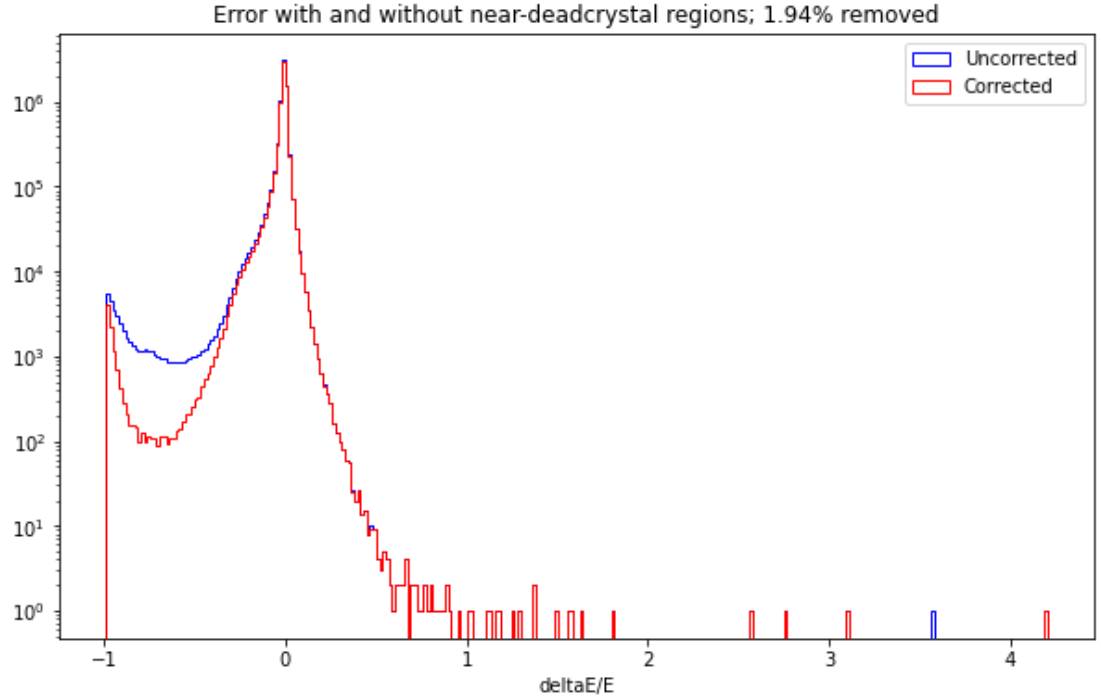

Entries 799241
Mean x 177.8
Mean y −1.21
Std Dev x 104.7
Std Dev y 58.66
ieta
iphi

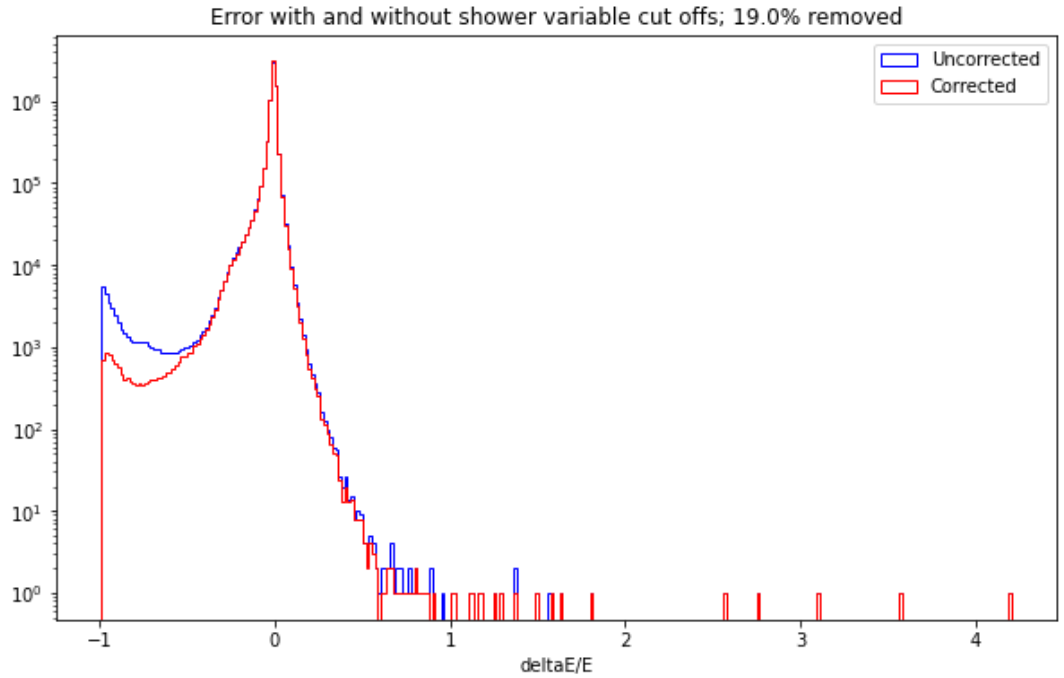

mean abs(delta/E)>0.0 heatmap

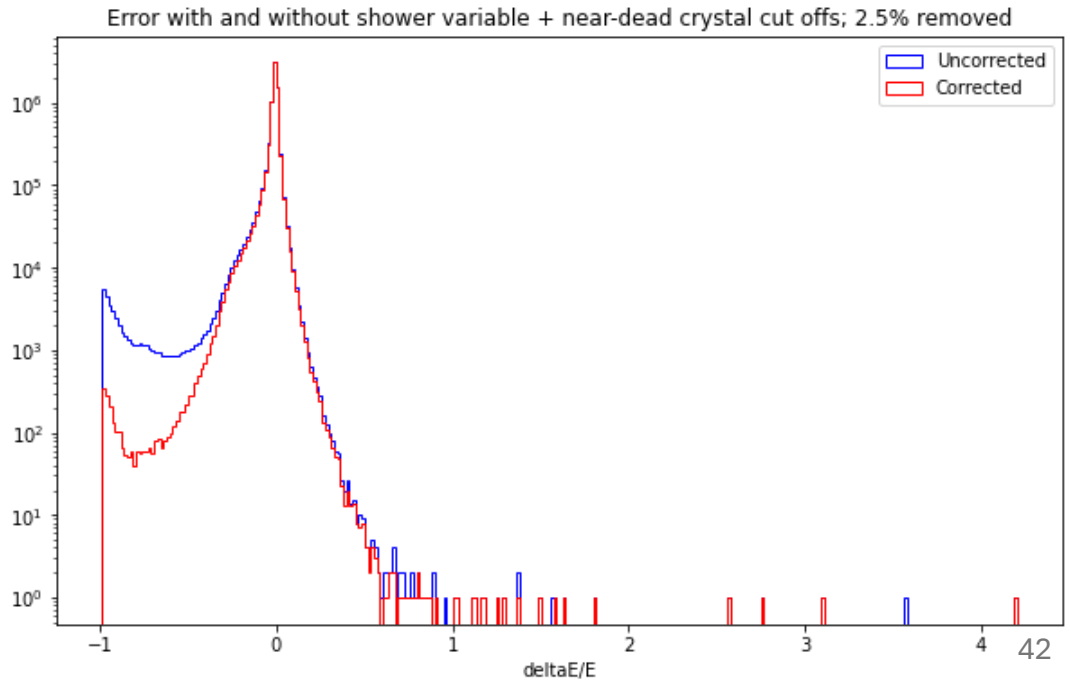

mean abs(delta/E)>0.06 heatmap
mean
Entries 811
Mean x 174.1
Mean y −2.026
Std Dev x 104.7
Std Dev y 61.47
ieta
iphi

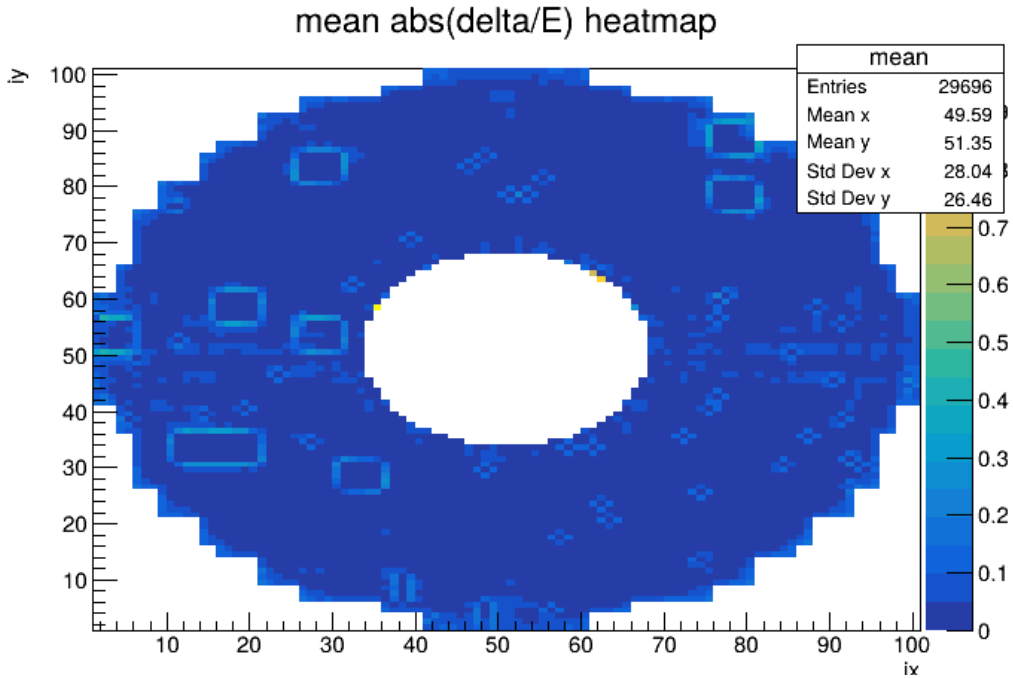

mean abs(delta/E)>0.09 heatmap
mean
Entries 813230
Mean x 172.4
Mean y −2.304
Std Dev x 104.9
Std Dev y 57.83
iphi

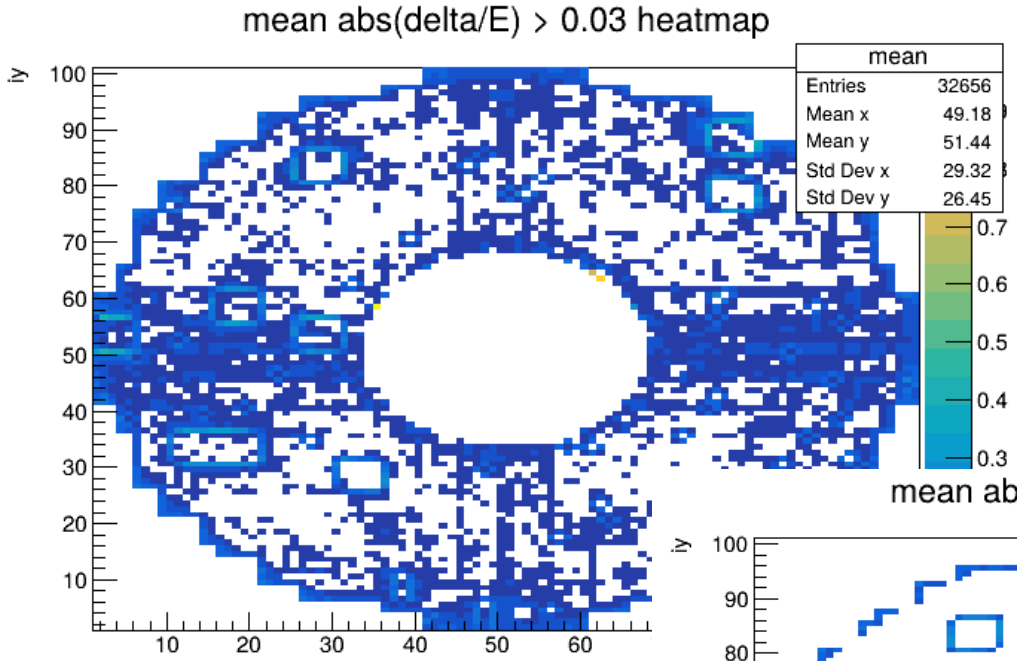

Error with and without near-supermodule transition regions; 19.0% removed
Uncorrected
Corrected
deltaE/E

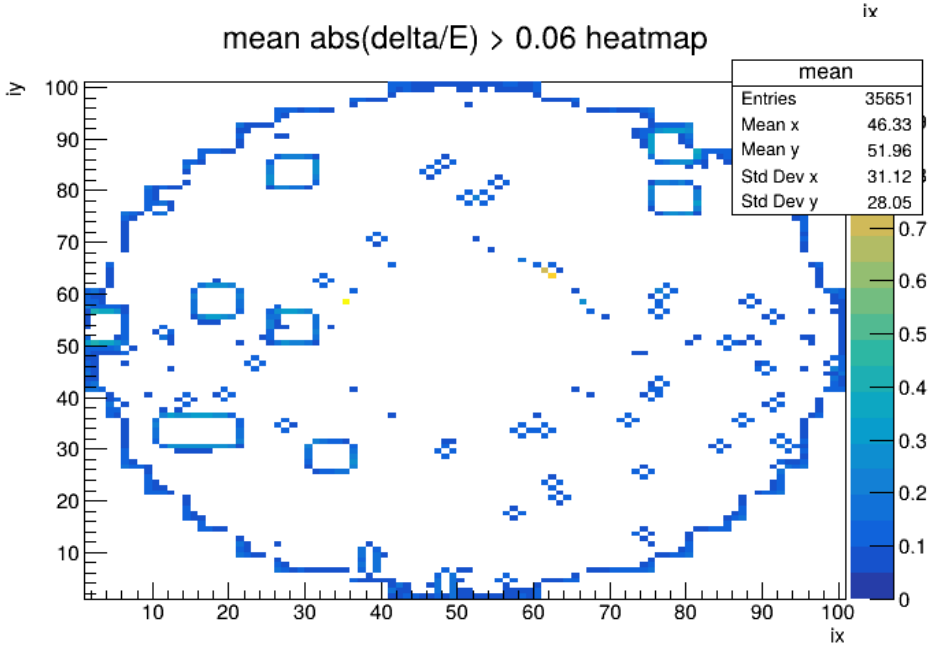

Error with and without near-deadcrystal regions; 1.94% removed
Uncorrected
Corrected
deltaE/E

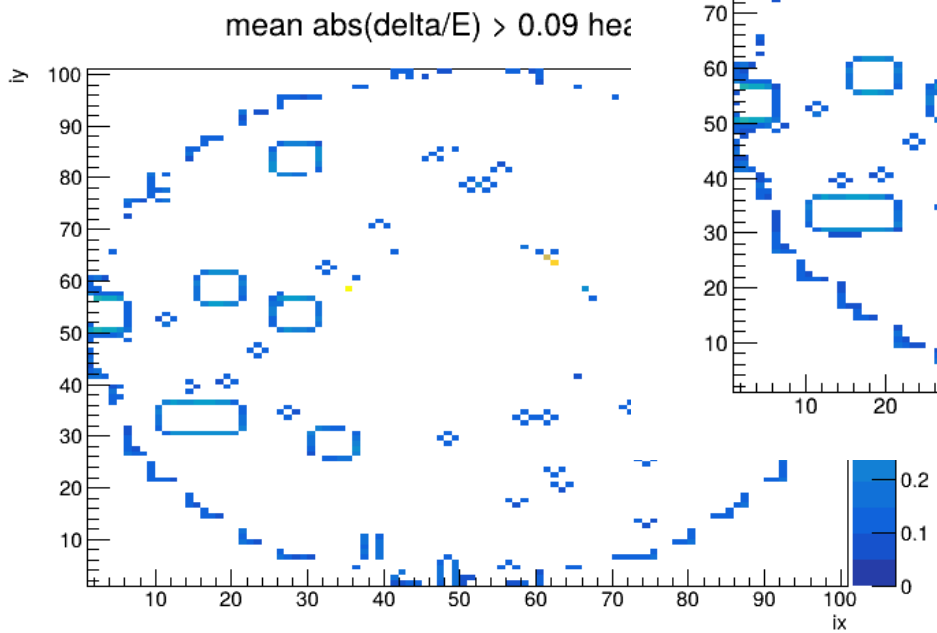

Error with and without shower variable cut offs; 19.0% removed
Uncorrected
Corrected
deltaE/E

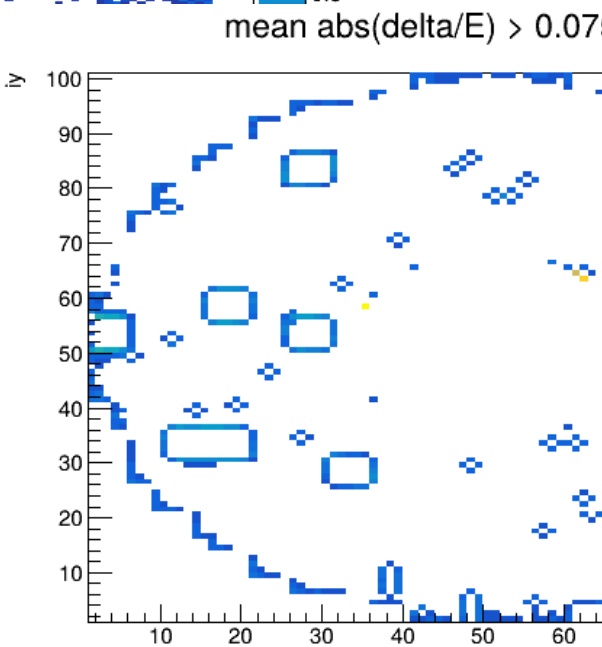

Error with and without shower variable + near-dead crystal cut offs; 2.5% removed
Uncorrected
Corrected
deltaE/E


mean abs(delta/E) heatmap
mean
Entries 29696
Mean x 49.59
Mean y 51.35
Std Dev x 28.04
Std Dev y 26.46
iy
ix


mean abs(delta/E) > 0.03 heatmap
mean
Entries 32656
Mean x 49.18
Mean y 51.44
Std Dev x 29.32
Std Dev y 26.45
iy


mean abs(delta/E) > 0.06 heatmap
mean
Entries 35651
Mean x 46.33
Mean y 51.96
Std Dev x 31.12
Std Dev y 28.05
iy
ix

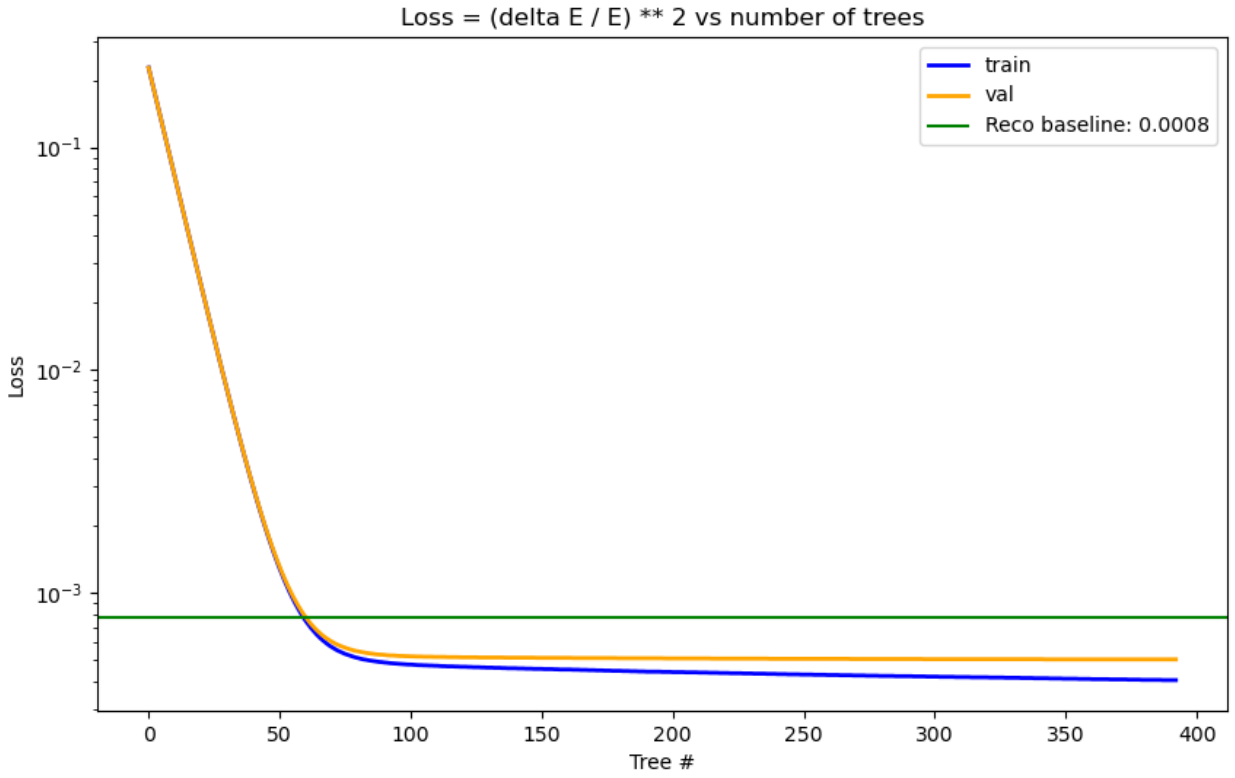

mean abs(delta/E) > 0.09 hea
iy
ix


mean abs(delta/E) > 0.07
iy

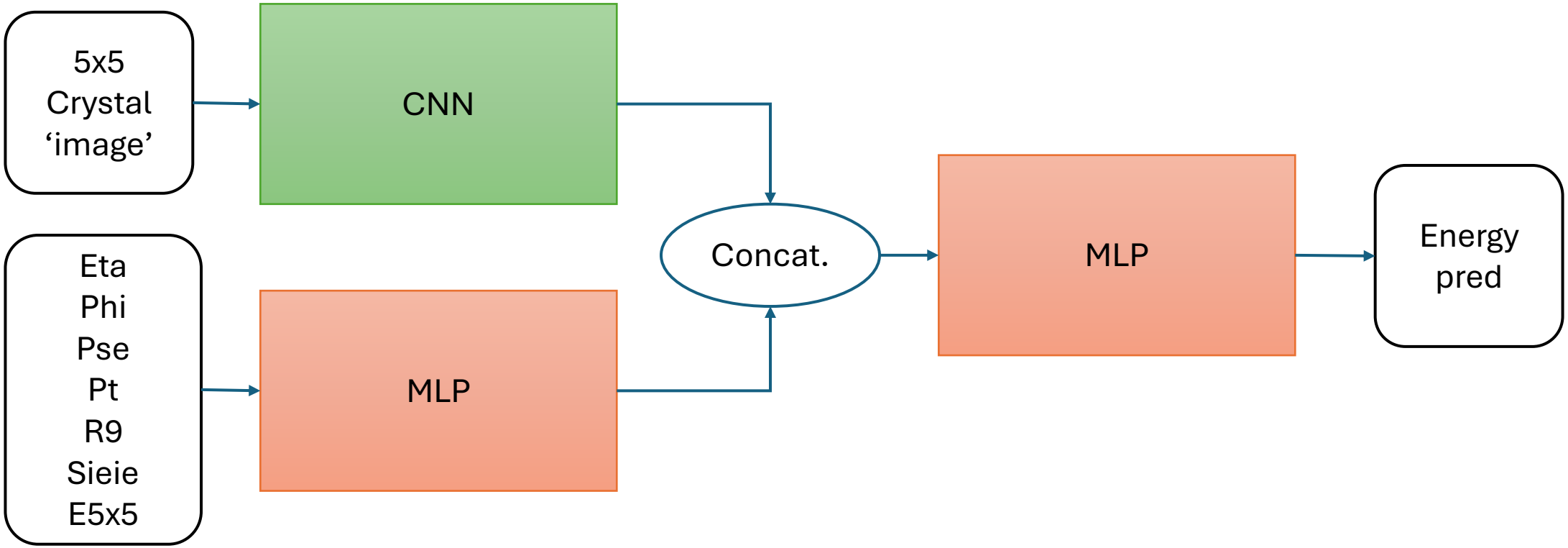
BDT

- XGBoost
  - Features:
    - e1x5
    - e2nd
    - e2x5B, e2x5L, e2x5M, e2x5R, e2x5T
    - e5x5
    - eB, eL, eR, eT
    - eMax
    - eta ieta
    - iphi phi
    - pse
    - pt
    - r9
    - sieie
  - Labels:
    - y_true = E_mc / E_reco
    - y_pred = prediction on y_true
  - Loss:
    - ((E_pred – E_mc)/E_mc) ** 2
    - = ((y_pred * E_reco - E_mc) / E_mc) ** 2
    - = ((y_pred – E_mc/E_reco) / (E_mc/E_reco)) ** 2
    - = ((y_pred – y_true)/y_true) ** 2
  - No normalization (not necessary for BDT)

BDT (XGBoost) – Best model



Deep Learning: CNN – MLP Hybrid



Attempted Normalizations

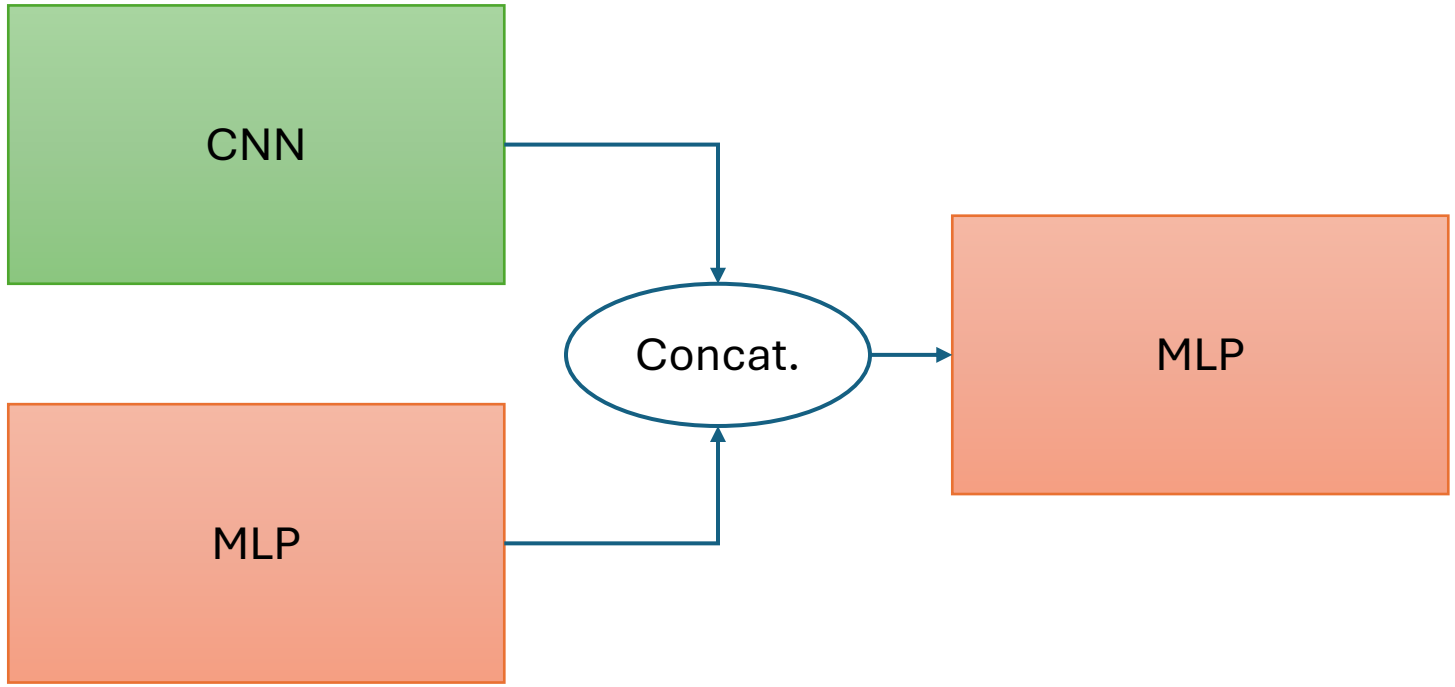
- Input image (X_img):
  - Normalise each image by its eMax
  - Log then normalise each image by its eMax
  - Sieie normalization: (max(0, 0.47 + log(E_i/E_5x5))
- Input features (X_tabular):
  - Mean std normalization
  - Transforming to [0,1] (dividing by scale factor)
  - Taking log then applying one of the two above
  - Sin-cos embedding for phi
- Output labels (y):
  - Unscaled
  - Transforming to [0,1] (dividing by scale factor)
  - Log
  - Y = E_mc / E_reco

Loss functions

- L = ((E_mc – E_reco / E_mc))**2
- L = (E_mc – E_reco) ** 2
- L = log(sig ** 2) + 0.5 * ((E_mc – mu) / sig)) ** 2 (predicting mu and sig)
- First loss performed best

More on architecture

2 3x3 Conv layers
Padding = 1
Channels = 16, 16



Dense (d = 200)

Train vs val for CNN