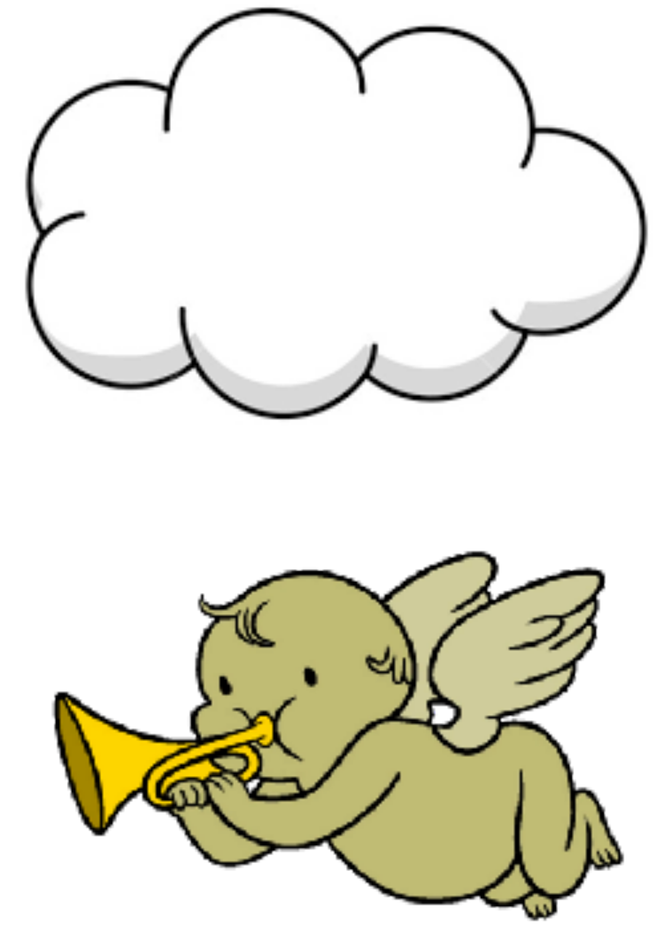
IDENTIFYING OBJECTS WITH NEURAL NETS

MATT ZHANG

My Co-Authors:

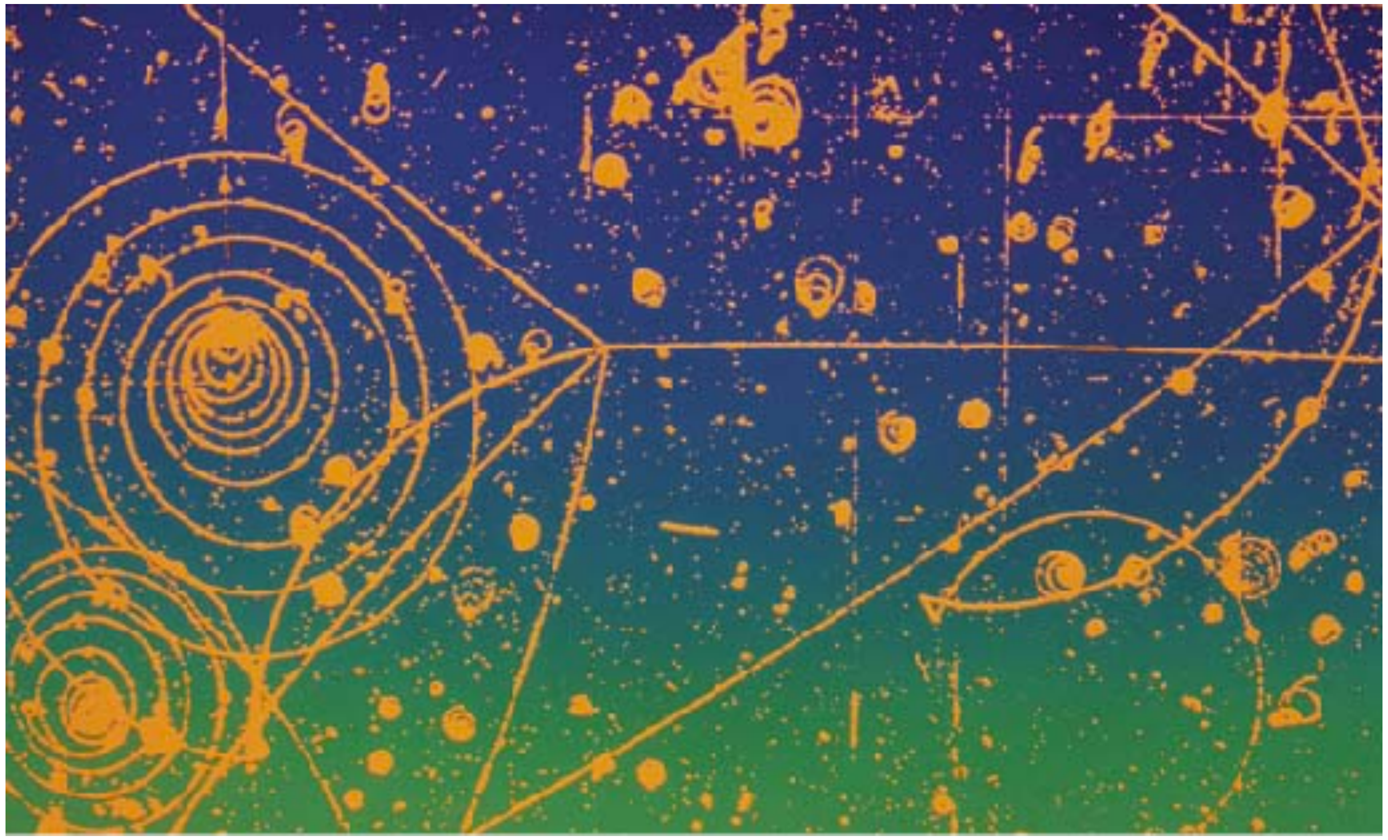
Amir Farbin
Ben Hooberman
Junze Liu
Maurizio Pierini
Ryan Reece
Jean-Roch Vlimant
William Wei



Finding New Physics

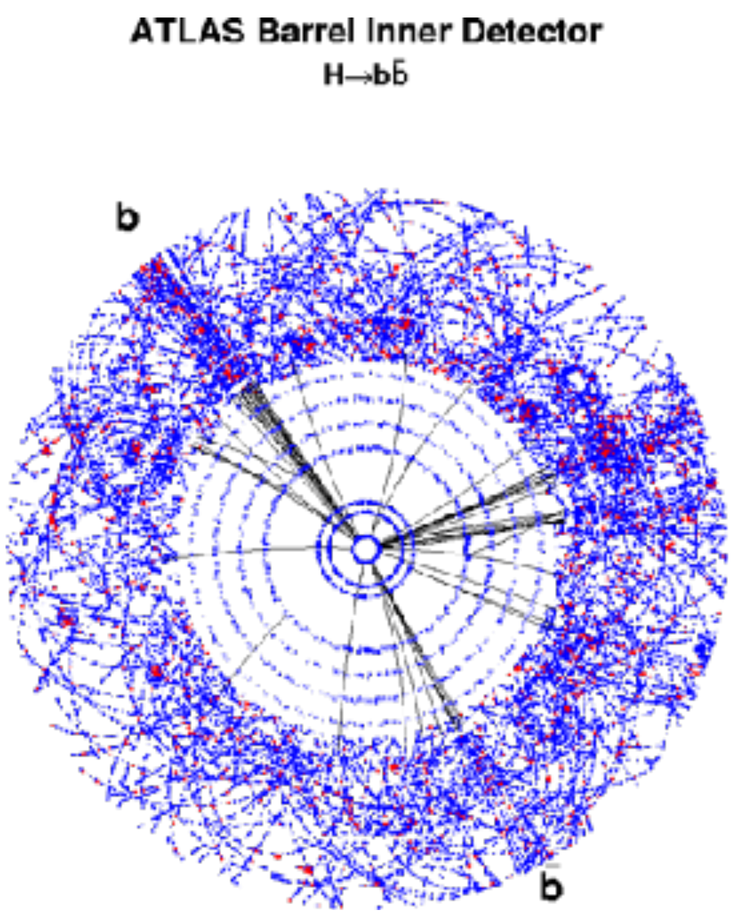
How to improve physics searches?

- Higher energy collisions
- More luminosity
- Improving analysis techniques

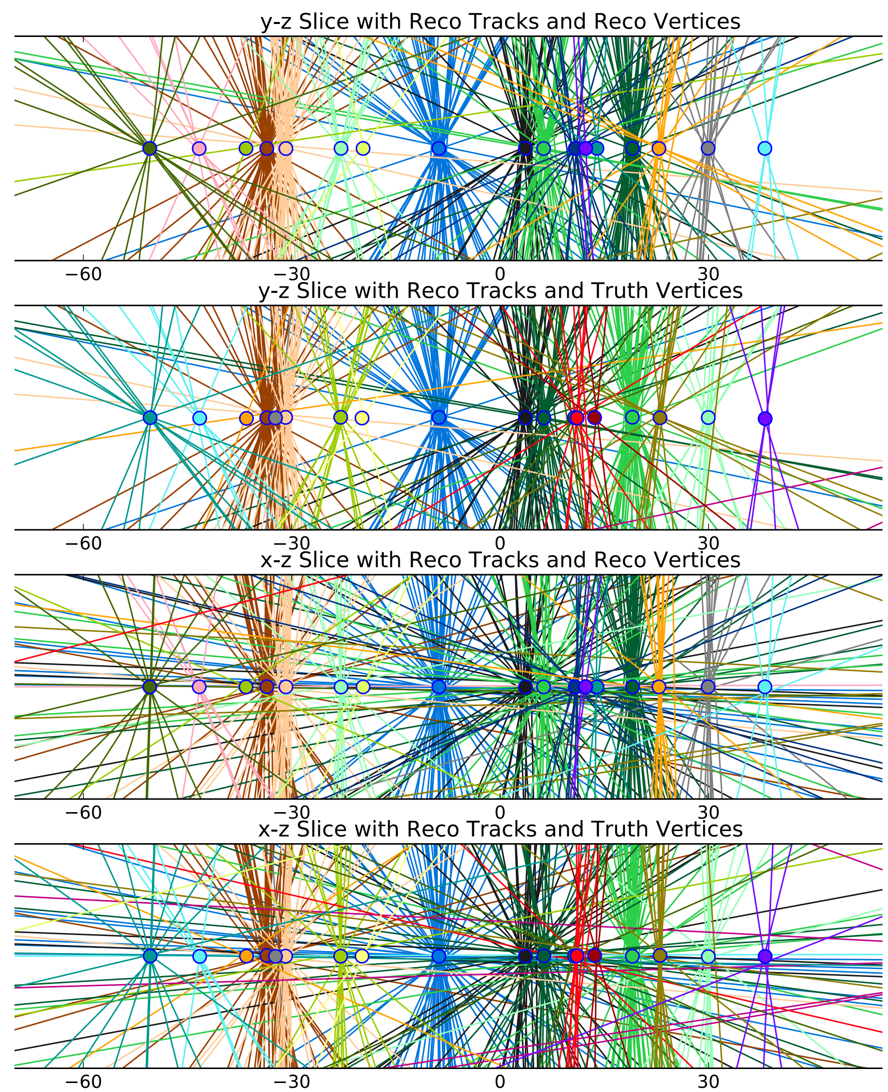


More accurate event reconstruction and signal/background discrimination through machine learning has an equivalent effect to adding more luminosity

Machine learning techniques are being explored for many HEP applications:



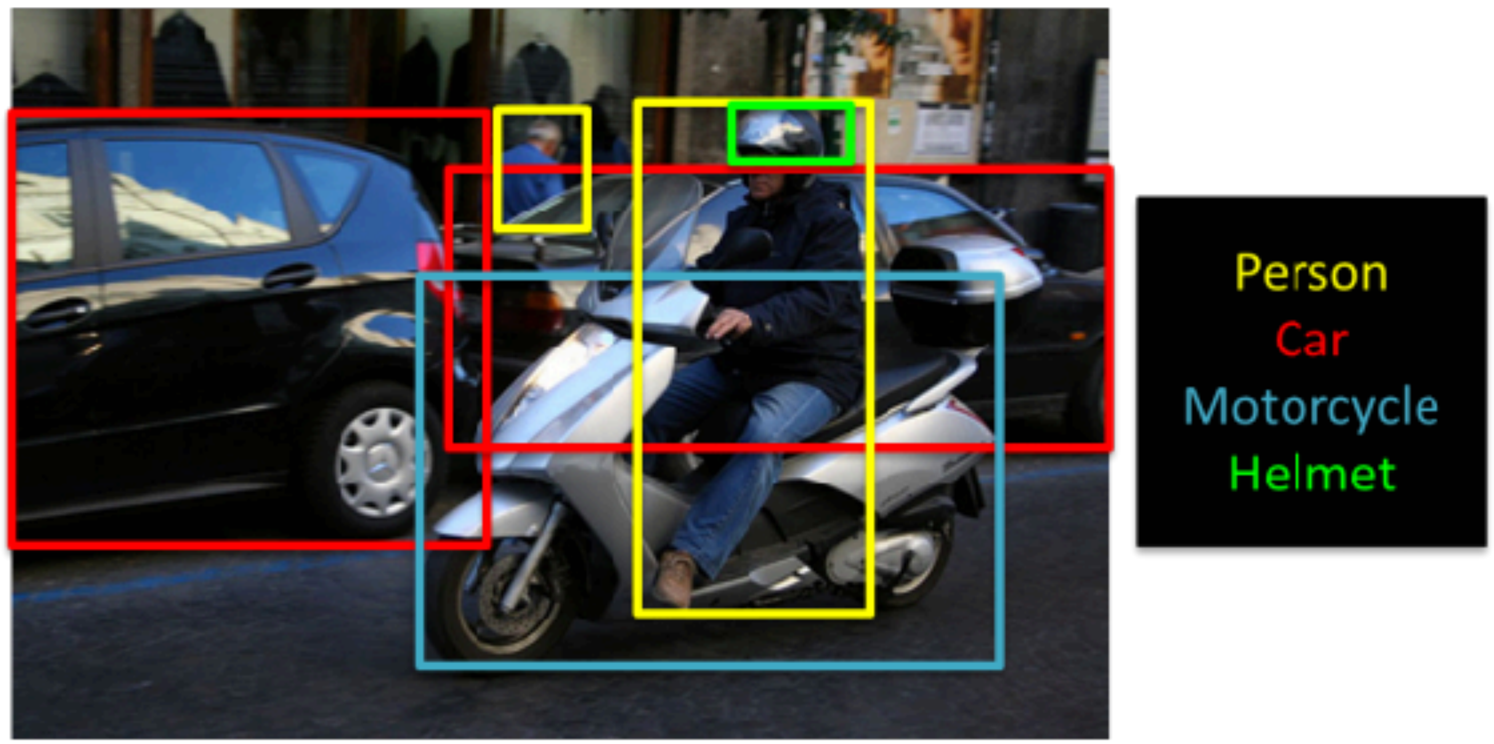
Track reconstruction



Vertex finding

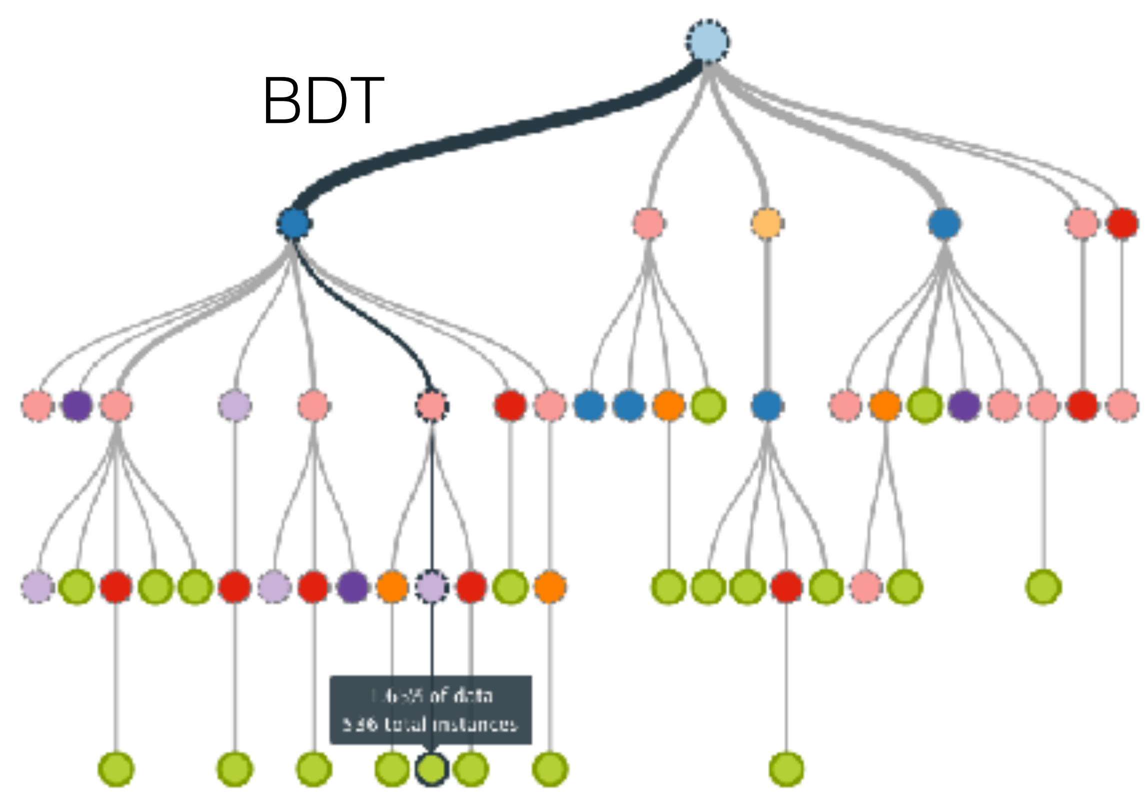
Jet identification

convolutional neural net



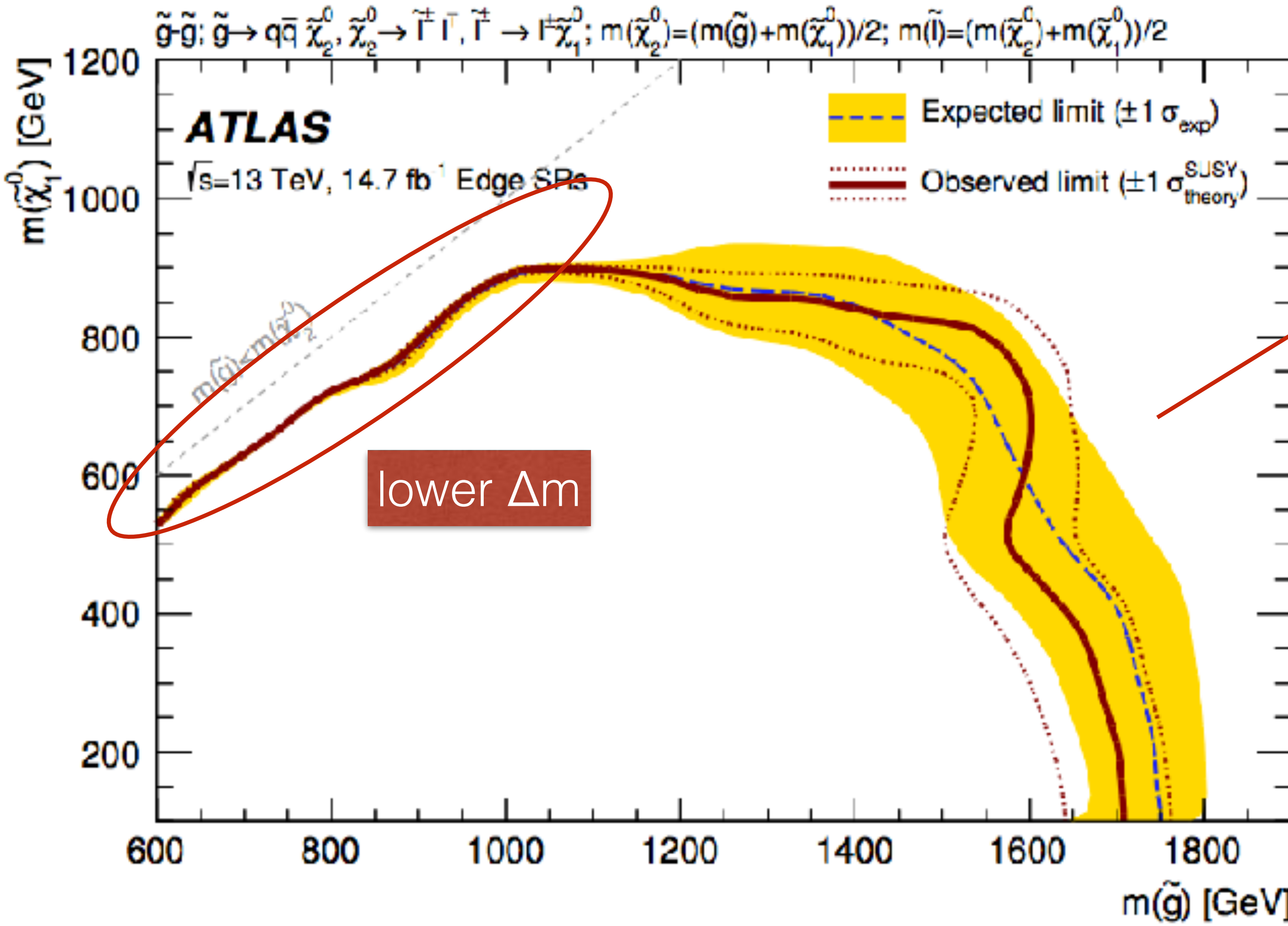
Traditional analysis techniques often rely on calculating features, then performing simple cuts and inequalities to identify signals and backgrounds.

BDT

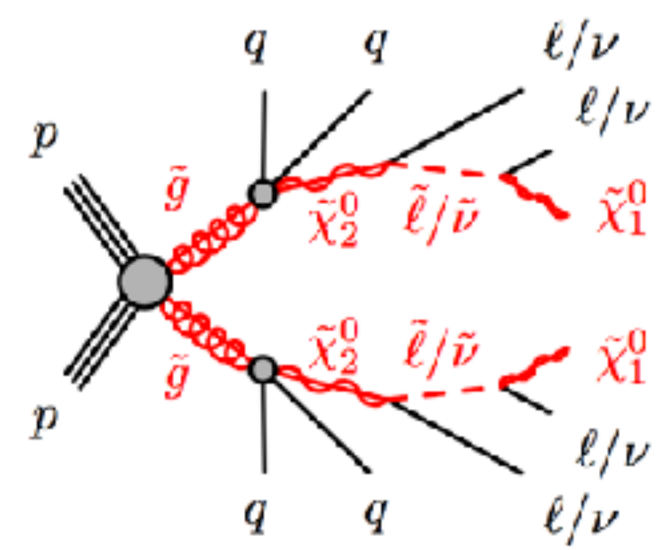


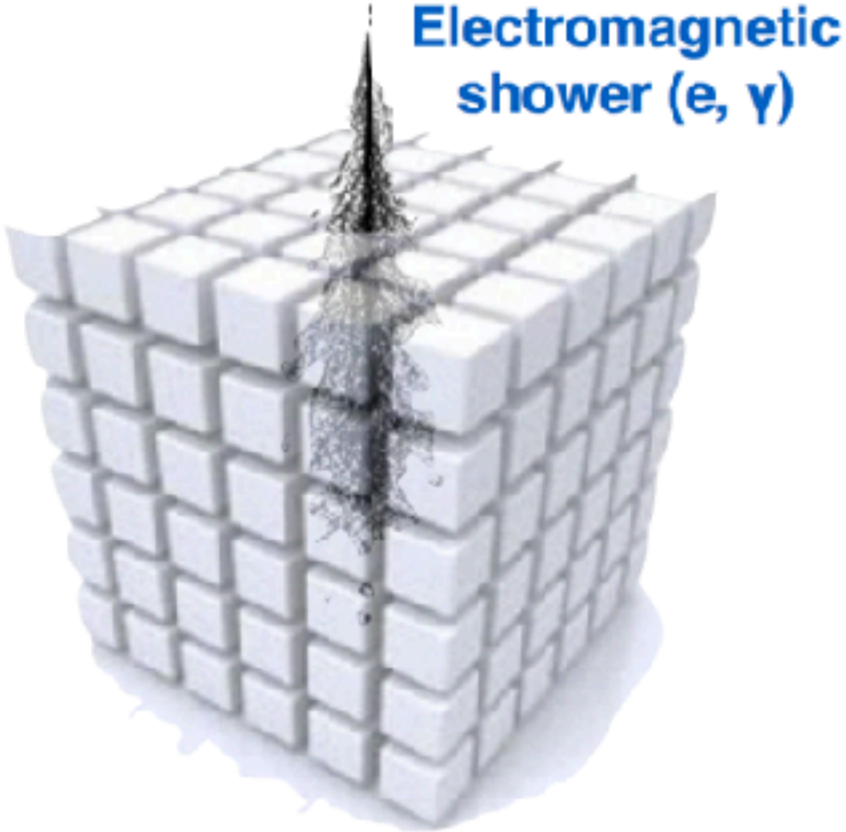
ML techniques like decision trees and neural nets can either find deeper relations between features, or analyze image-like data where the complexity of data makes feature-based analysis difficult.

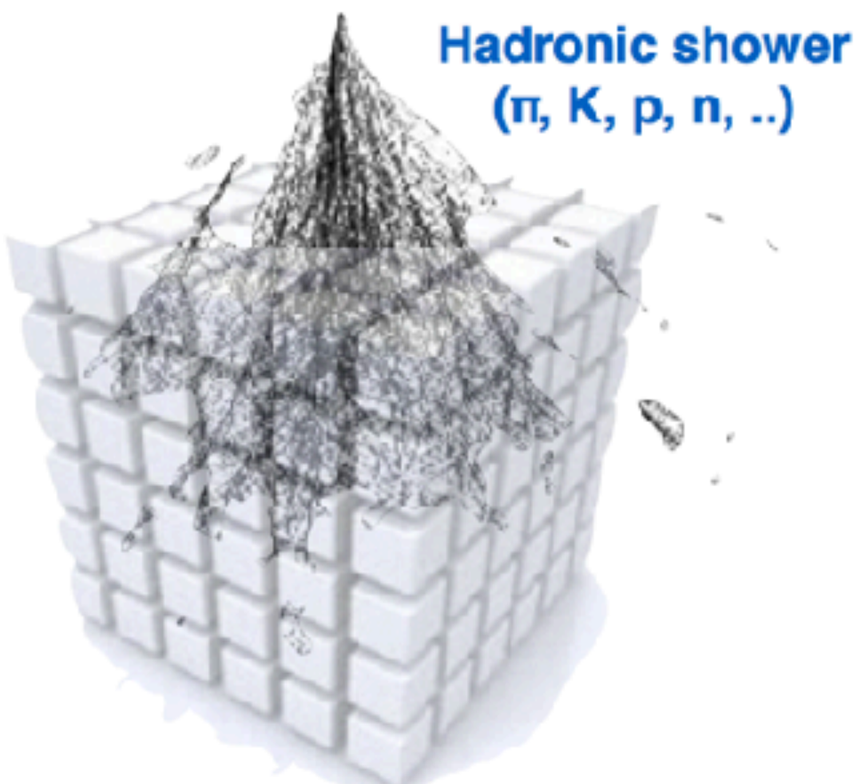
Accurate identification of low-pT objects (such as soft leptons) can push analyses further into the low Δm range



higher energy







The deposition of energy in calorimeters is a good example of an image-recognition problem

Our aim is to identify objects and their energies from depositions in the ECAL and HCAL.

Ultimately, we want to apply these techniques to ATLAS searches, but we begin by looking at single-particle events in a detector with simple geometry.

The eventual goal is to perform object recognition using complicated ATLAS detector geometry, in high-pileup events.

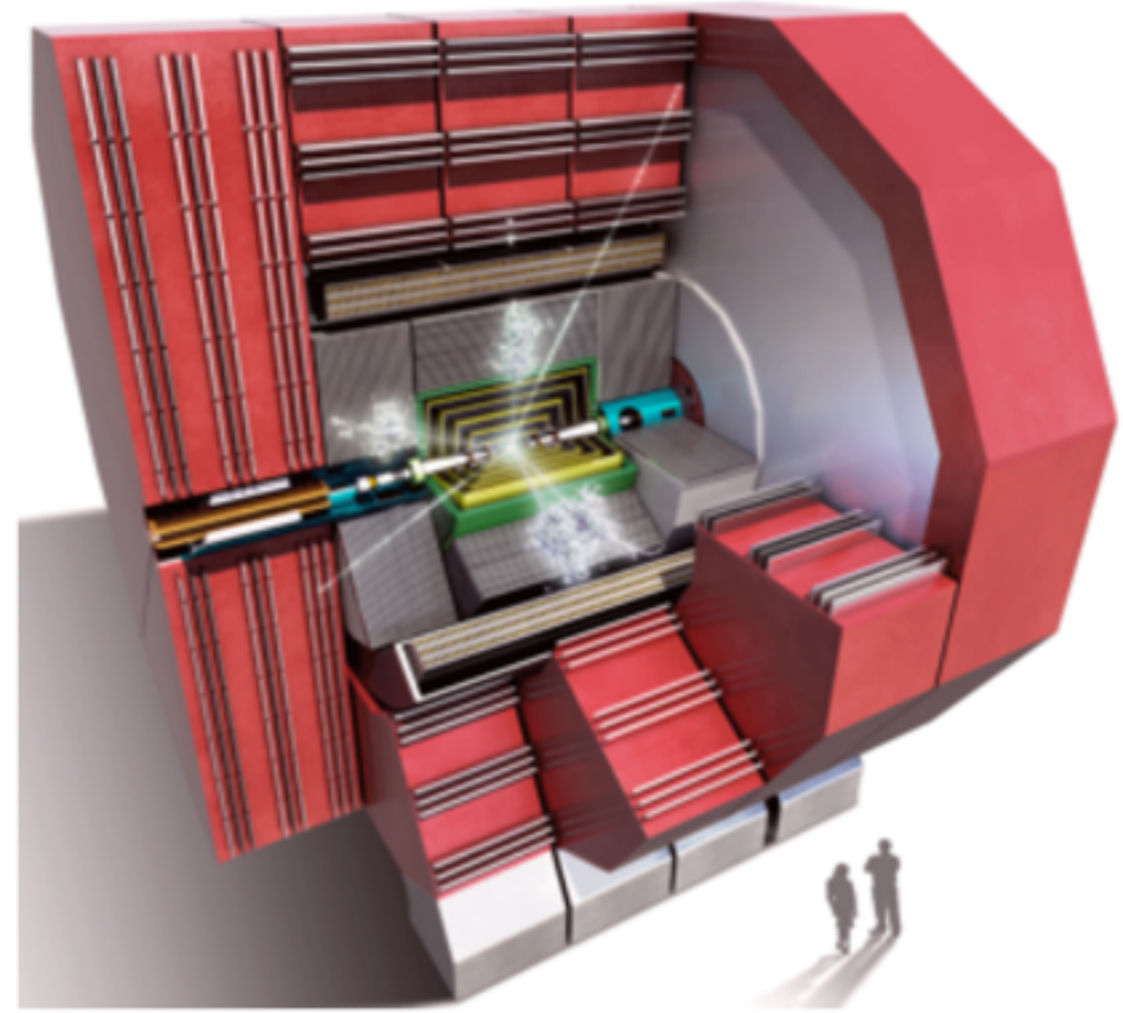
Dataset Generation

- Our data is generated using geometry for the LCD detector concept for the proposed CLIC linear electron-positron collider

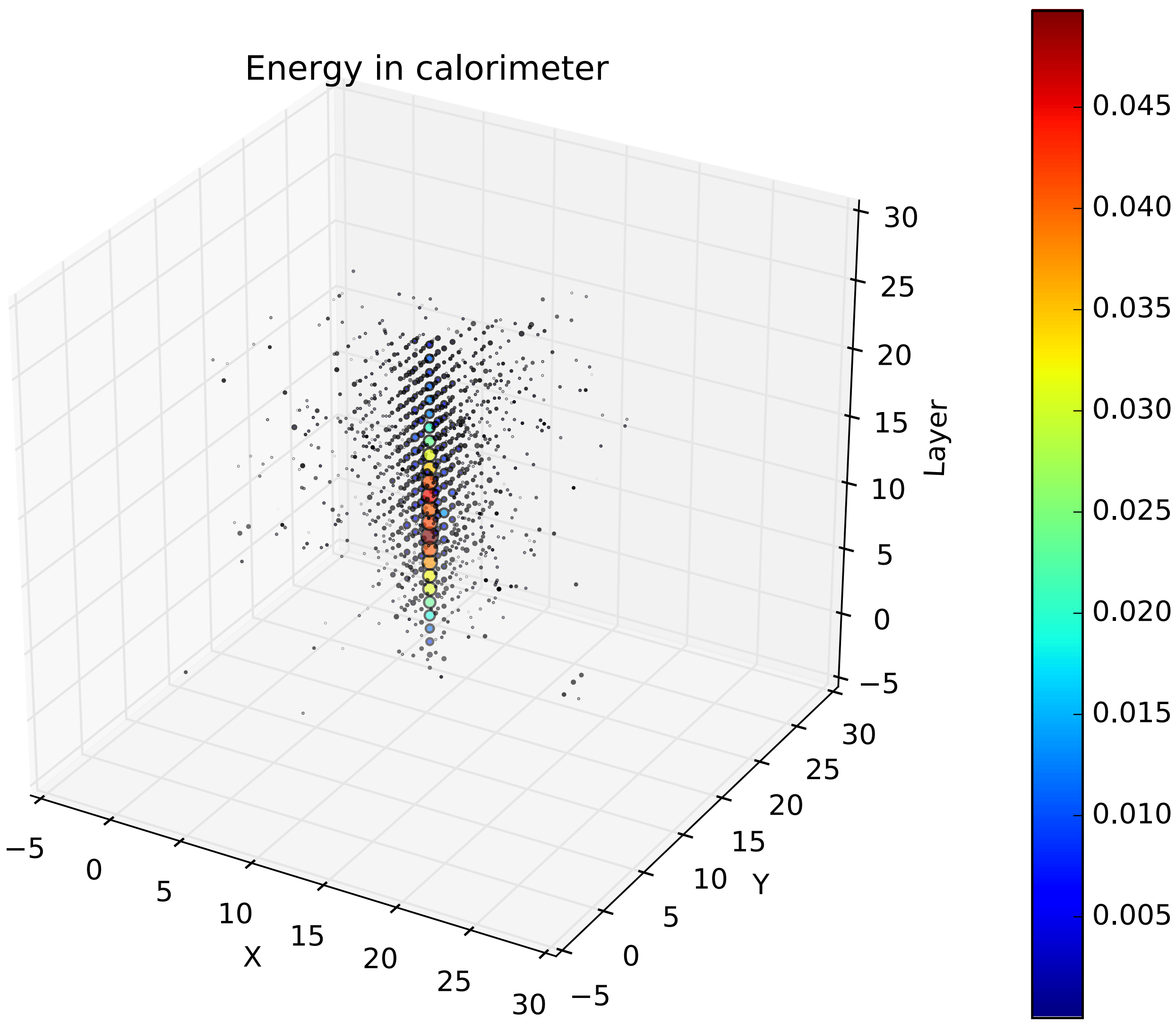
- A 25x25x25 ECAL slice and a 5x5x60 HCAL slice is cut around each object interacting with the calorimeters

- 4 classes of objects at various energies are generated - neutral pion, charged pion, electron/position, and photon

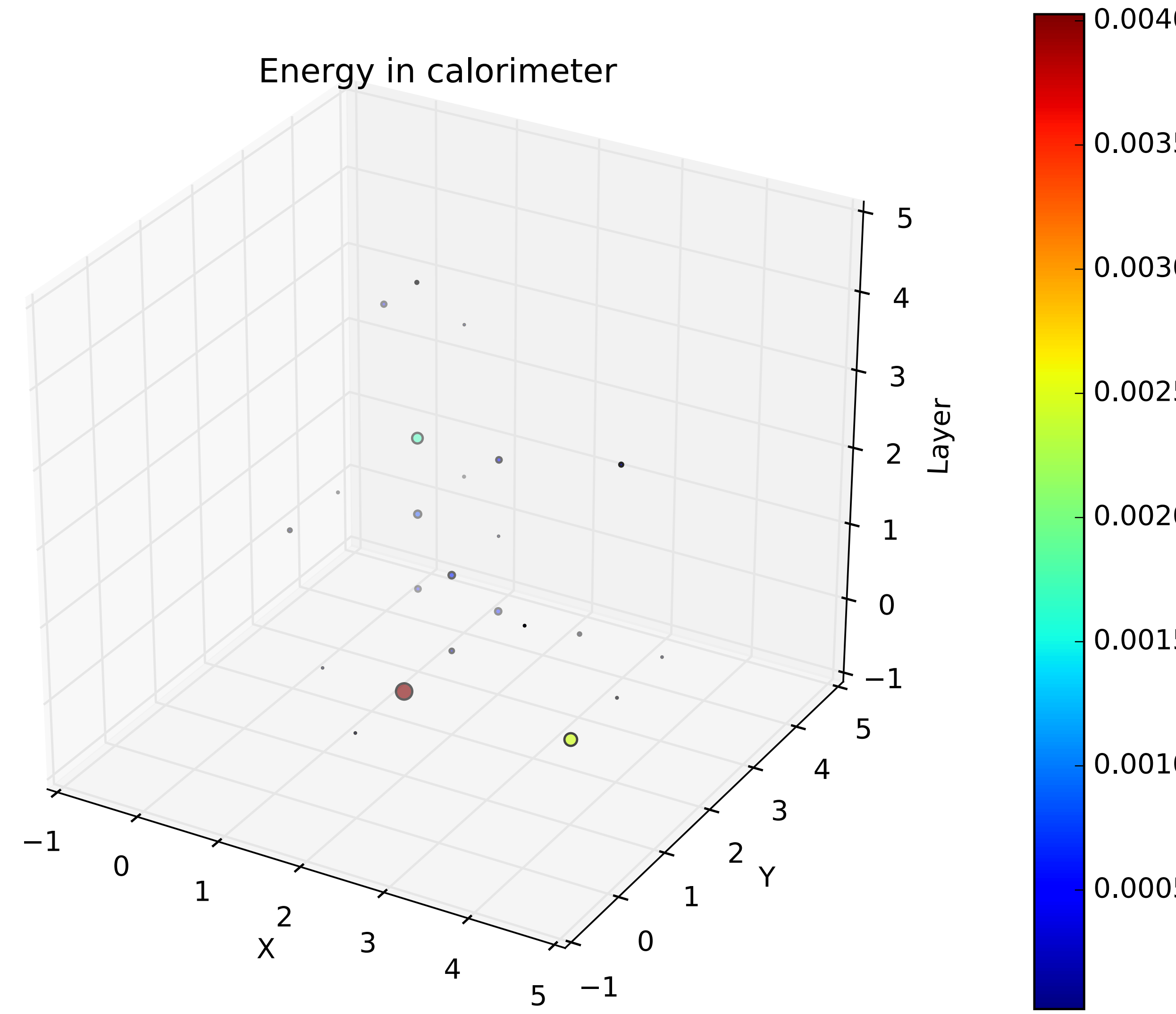
- Datasets and help provided by Jean-Roch Vlimant and Maurizio Pierini



60 GeV Photon Sample

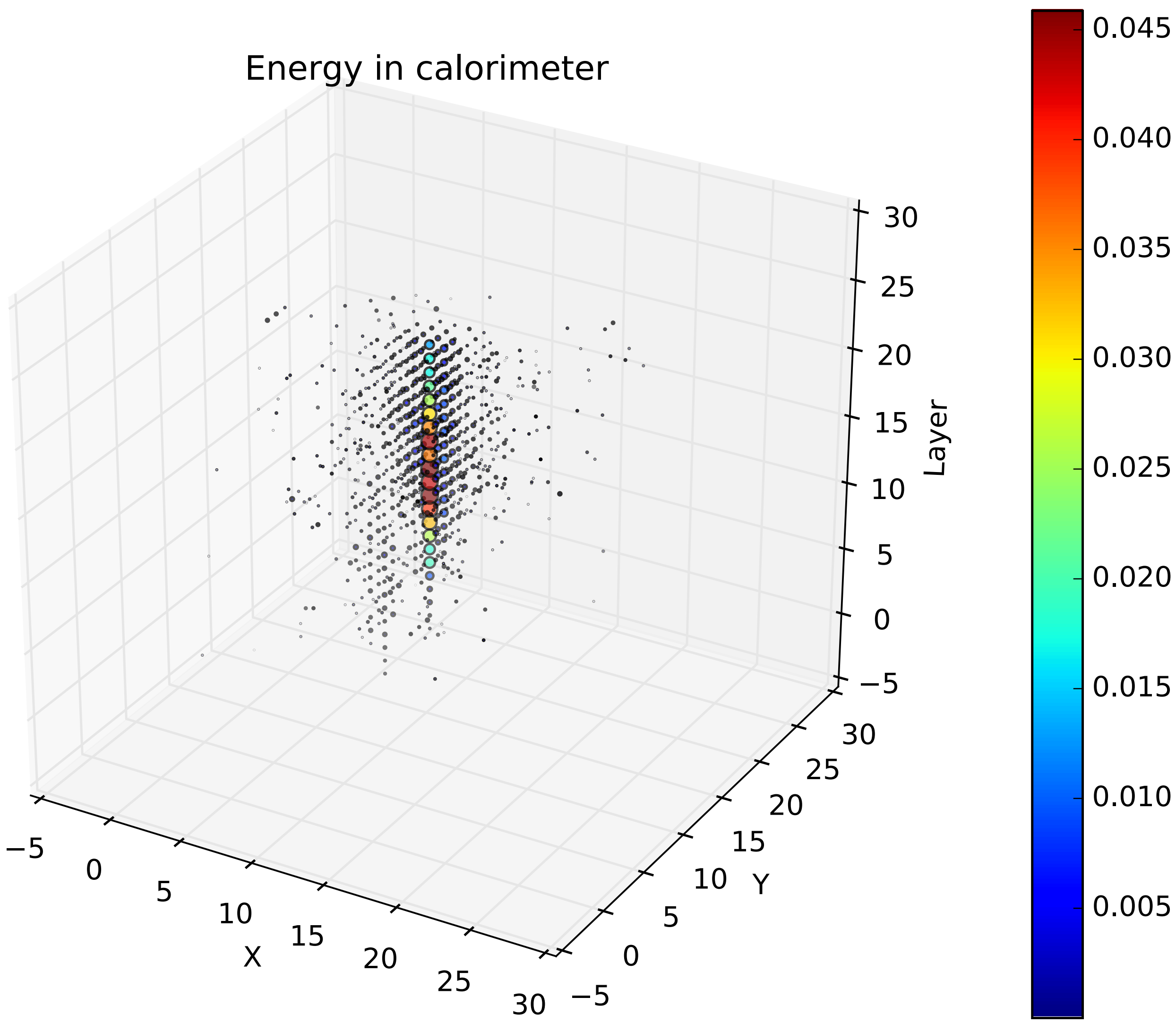


ECAL

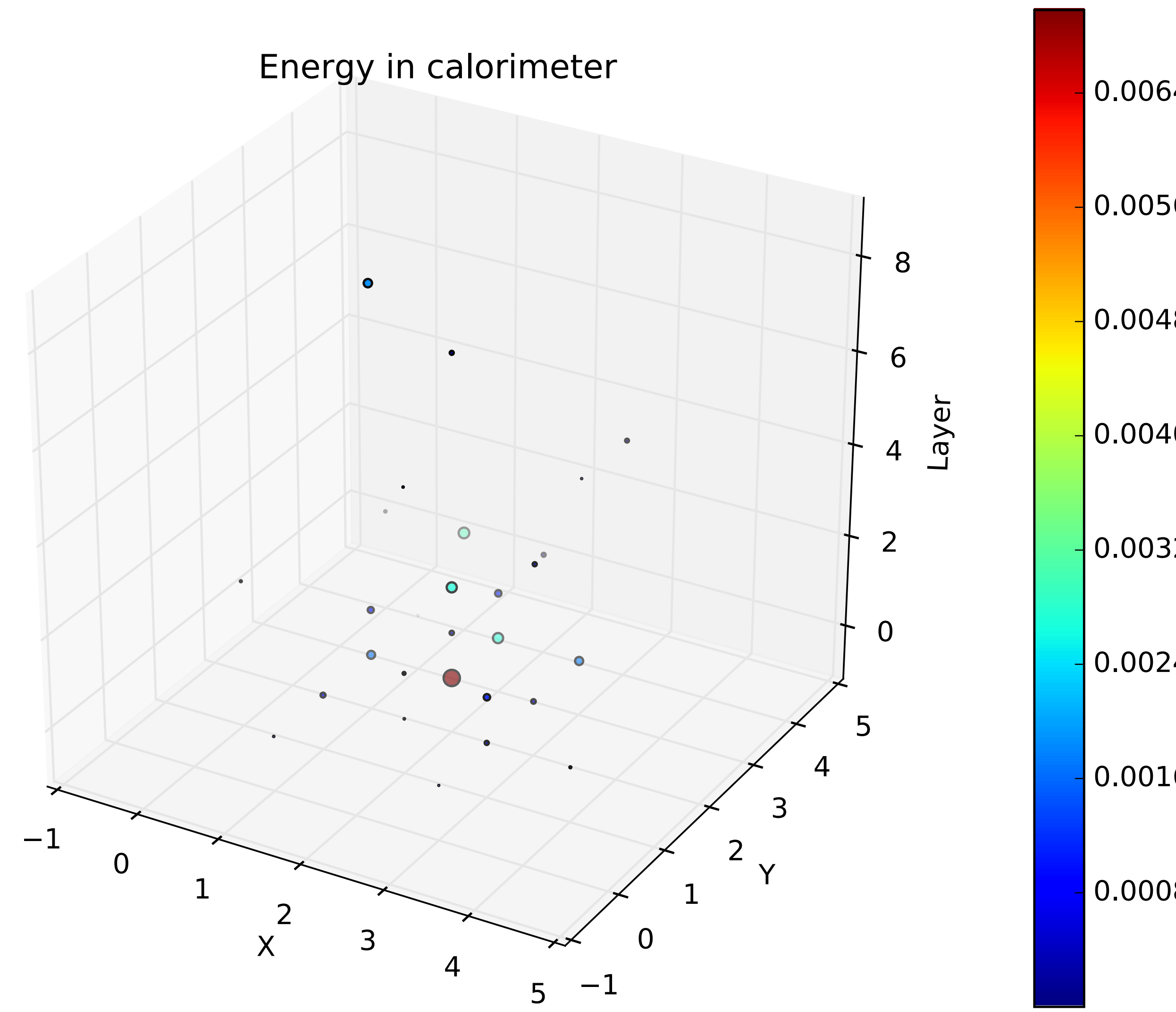


HCAL

60 GeV $\pi_0 \rightarrow \gamma\gamma$ Sample w/ Opening Angle < 0.01 rad



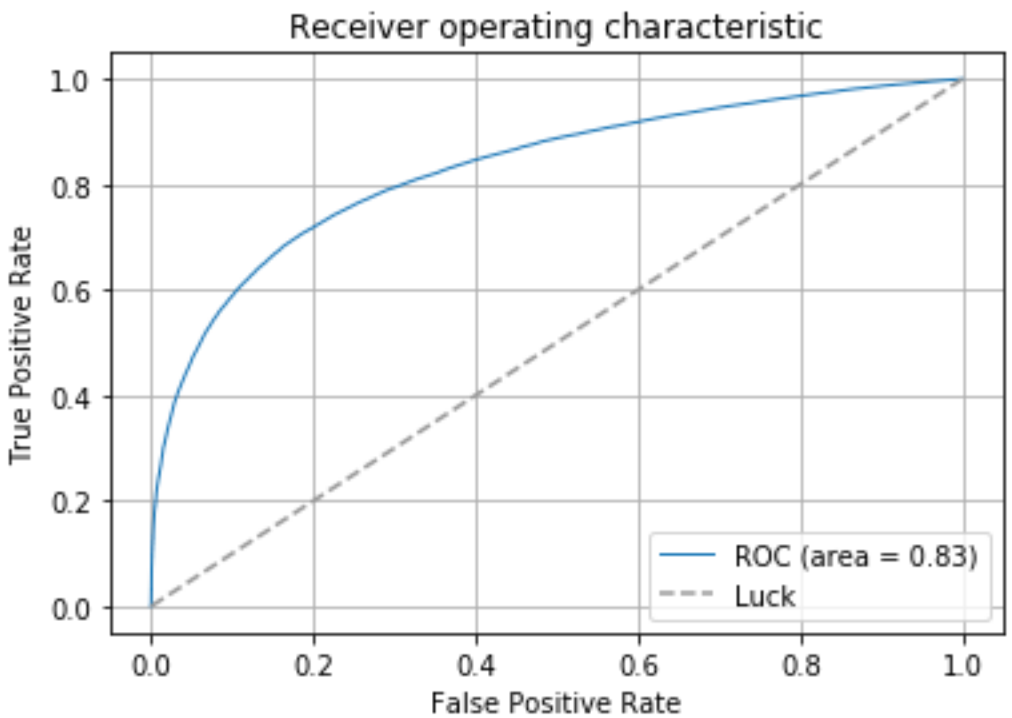
ECAL

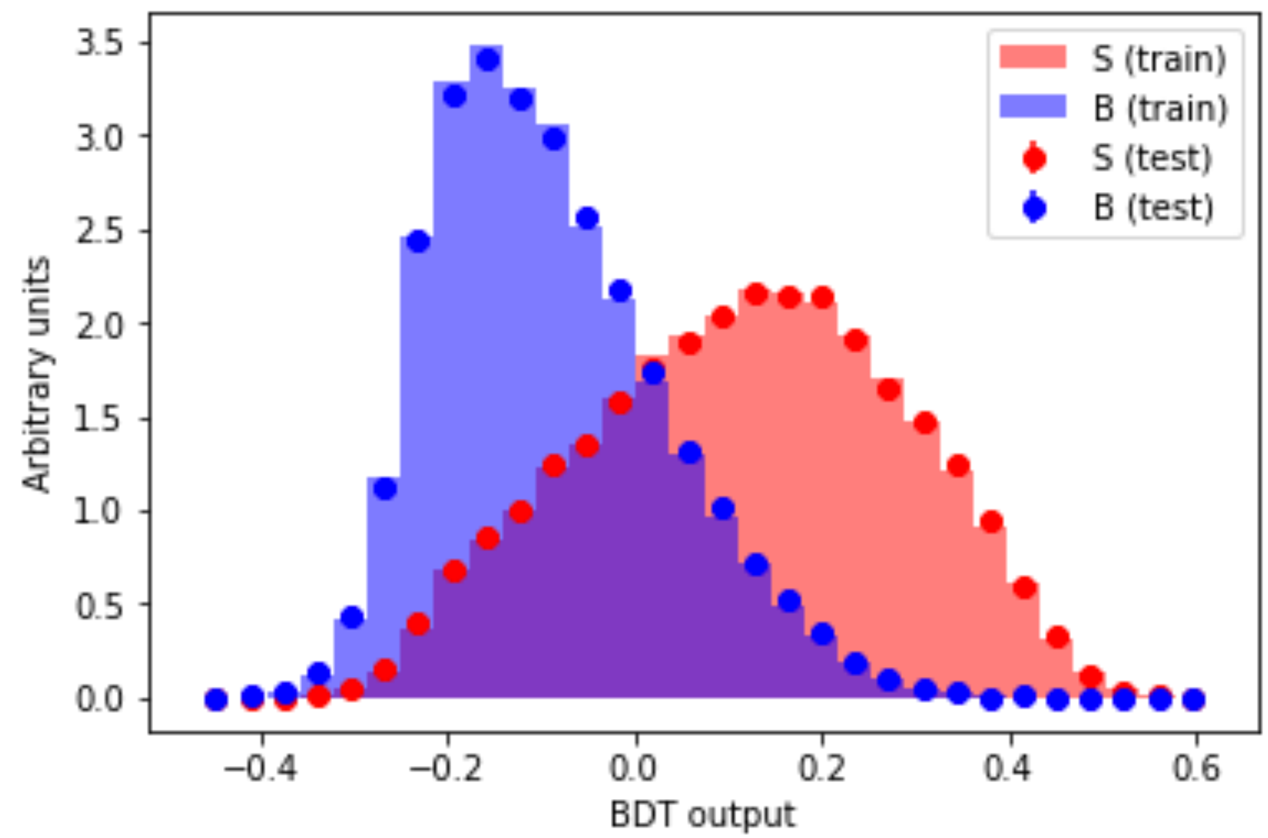


HCAL

Now we compare different methods for discriminating between 60 GeV γ and π_0, selecting π_0 events which decay to two photons with an opening angle less than 0.01 rad. We use 300,000 events with an even mix of γ and π_0.

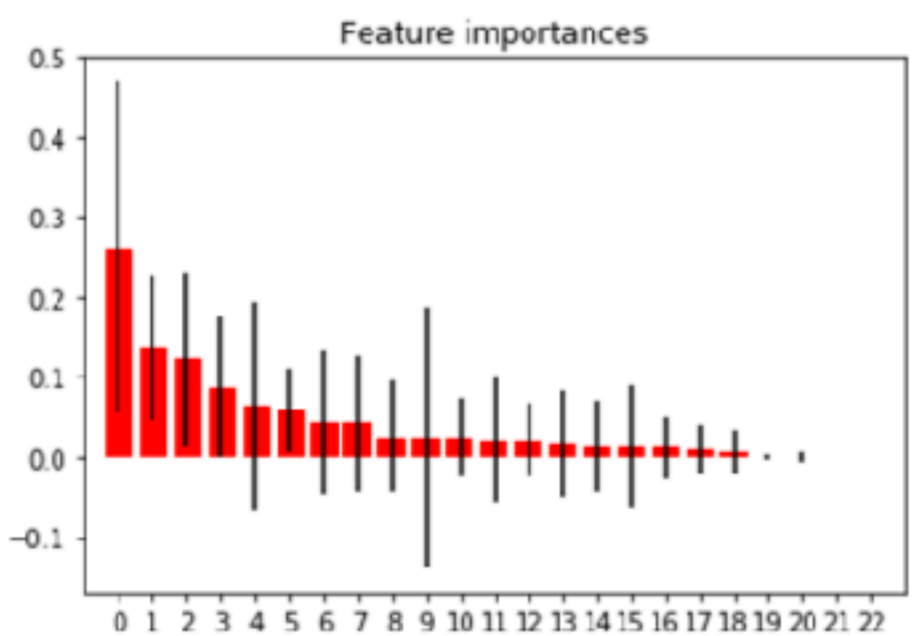
```
             precision    recall  f1-score   support

     photon       0.75      0.78      0.76     44400
   electron       0.77      0.74      0.75     44040

avg / total       0.76      0.76      0.76     88440

Area under ROC curve: 0.8311
```

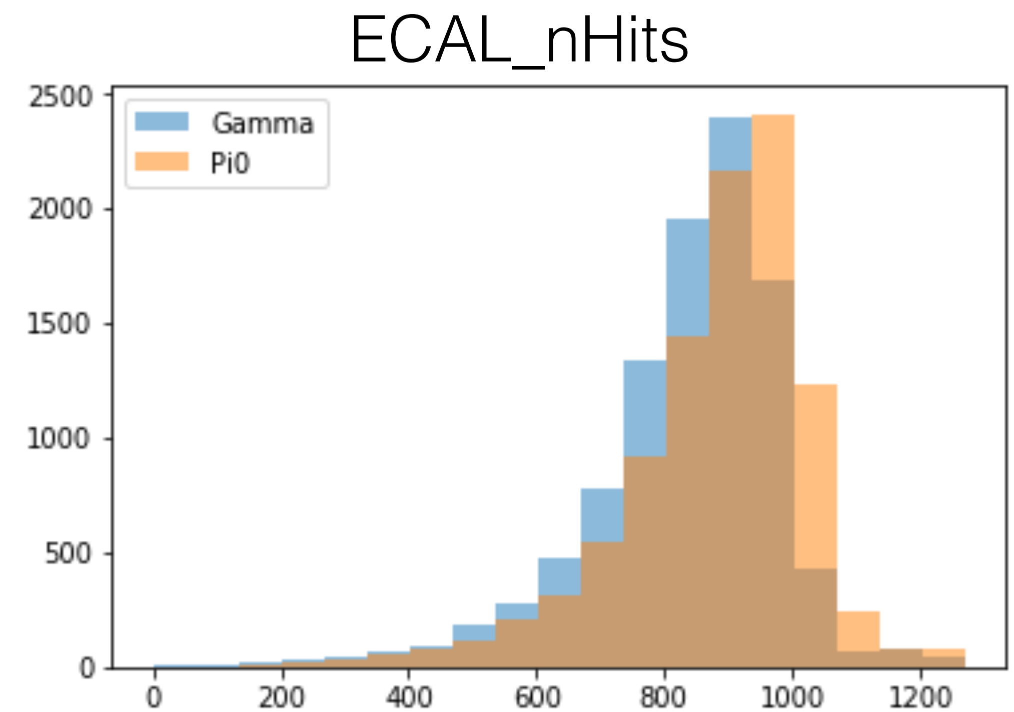


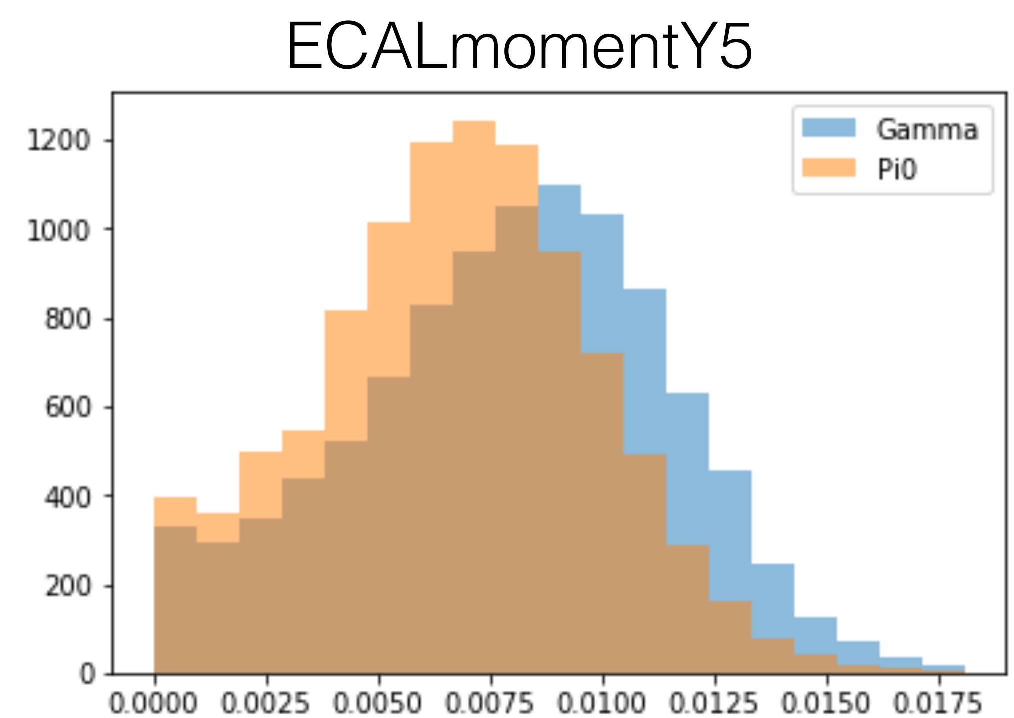


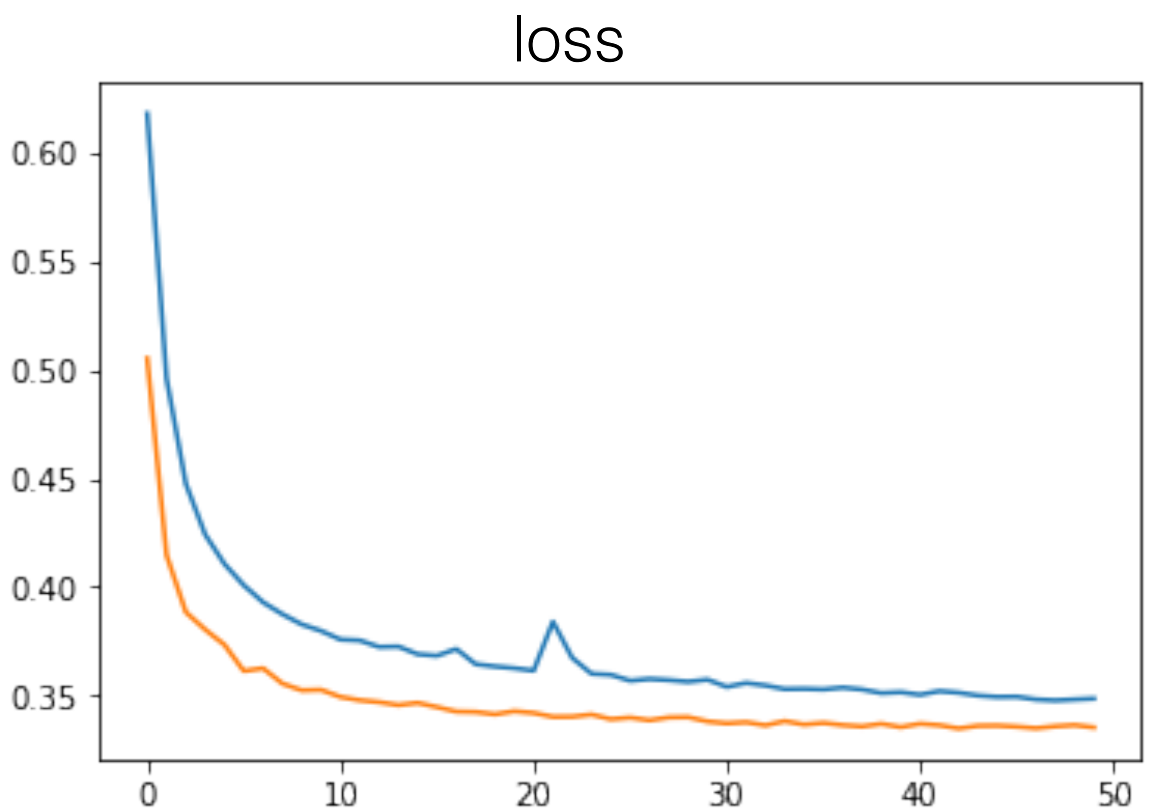
BDT is a feature-based method, and most closely matches what is used in standard analysis. Calculated features include energy and number of hits deposited in each calorimeter, as well energy ratios within layers and between the two calorimeters, and the moments in X, Y, and Z of the deposition patterns.

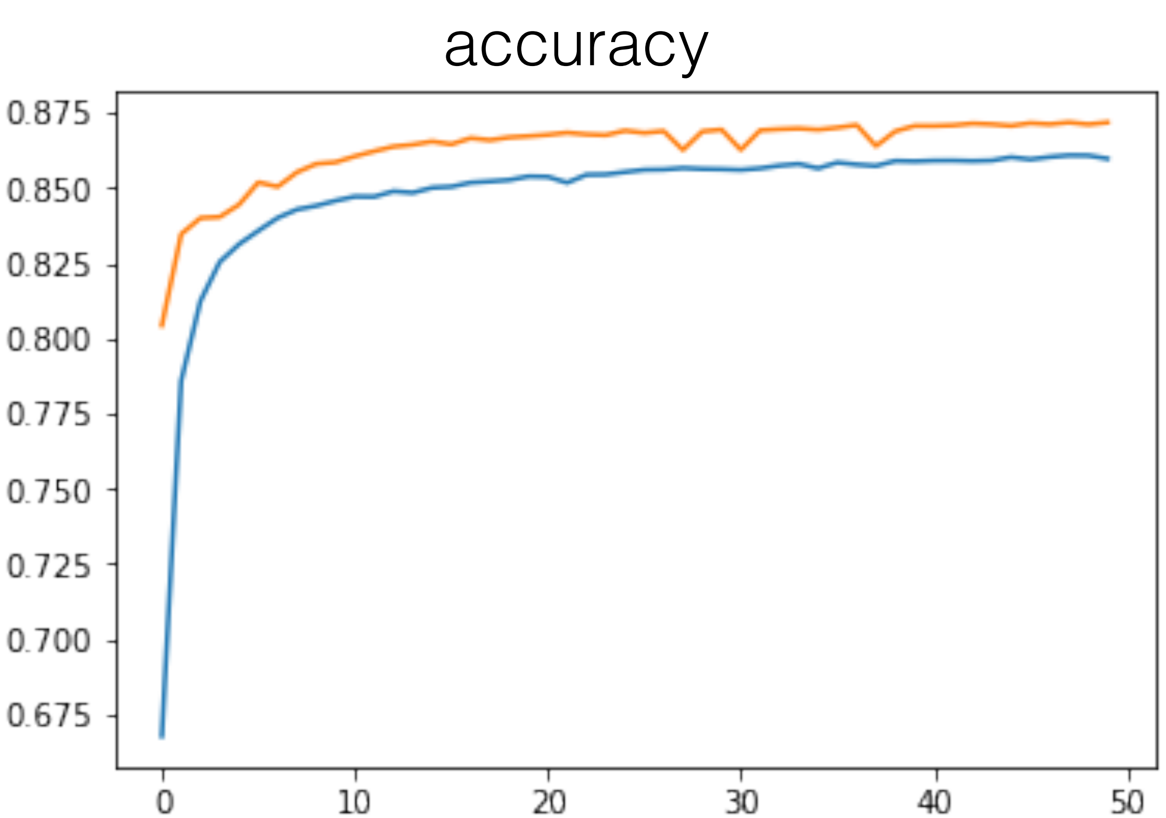
```
Feature ranking:
1. ECAL_nHits (0.262327)
2. ECALmomentY5 (0.136302)
3. ECAL_E (0.121541)
4. ECALmomentZ1 (0.087116)
5. ECALmomentY1 (0.063538)
6. ECALmomentX5 (0.058756)
7. HCAL_nHits (0.042656)
8. ECALmomentY4 (0.041717)
9. ECALmomentY2 (0.026004)
10. ECAL_ratioFirstLayerToTotalE (0.024097)
11. ECALmomentY3 (0.023396)
12. ECALmomentZ2 (0.021951)
13. ECALmomentX4 (0.021255)
14. ECALmomentX1 (0.016328)
15. ECALmomentX2 (0.013683)
16. HCAL_ratioFirstLayerToTotalE (0.012366)
17. HCAL_E (0.011821)
18. ECALmomentX3 (0.008615)
19. HCAL_ECAL_ERatio (0.006148)
20. ECALmomentZ3 (0.000224)
21. HCAL_ECAL_nHitsRatio (0.000159)
22. ECALmomentZ4 (0.000000)
23. ECALmomentZ5 (0.000000)
```









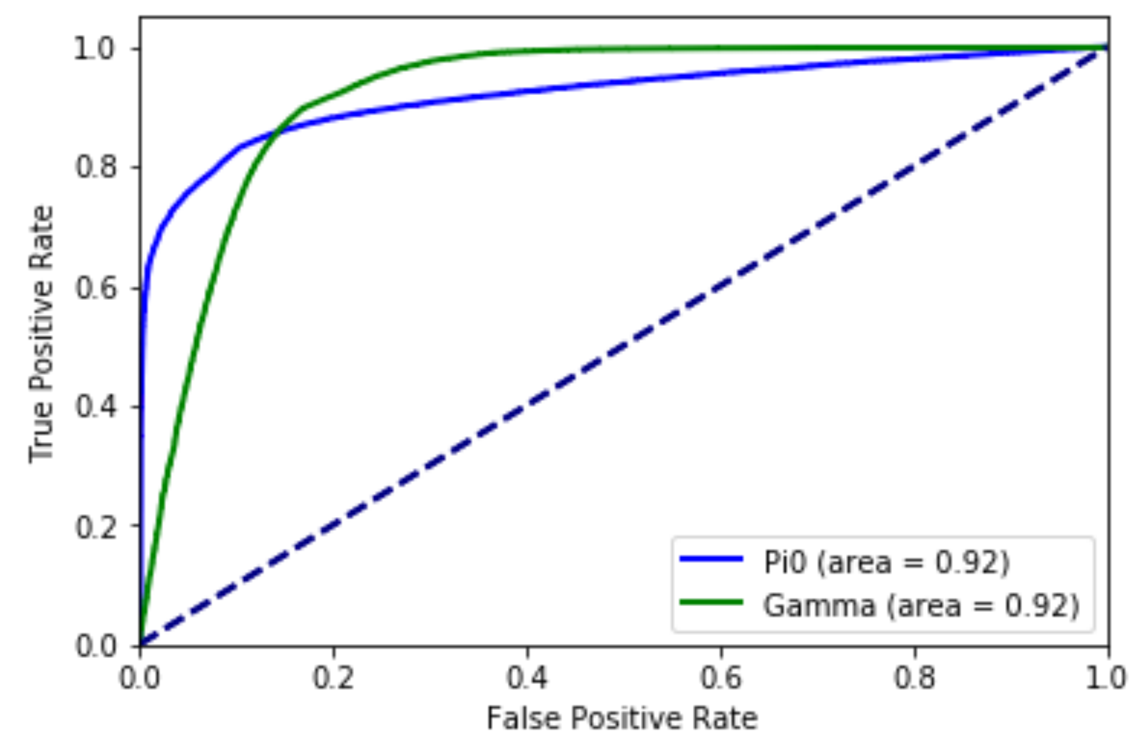


Here we train two **densely-connected neural nets**, with their respective outputs flattened and fed into a final densely connected layer.

blue = training
orange = validation

validation performs better than training due to the presence of two dropout layers

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 25, 25, 25)	0	
input_2 (InputLayer)	(None, 5, 5, 60)	0	
flatten_1 (Flatten)	(None, 15625)	0	input_1[0][0]
flatten_2 (Flatten)	(None, 1500)	0	input_2[0][0]
activation_1 (Activation)	(None, 15625)	0	flatten_1[0][0]
activation_4 (Activation)	(None, 1500)	0	flatten_2[0][0]
dense_1 (Dense)	(None, 64)	1000064	activation_1[0][0]
dense_3 (Dense)	(None, 32)	48032	activation_4[0][0]
activation_2 (Activation)	(None, 64)	0	dense_1[0][0]
activation_5 (Activation)	(None, 32)	0	dense_3[0][0]
dropout_1 (Dropout)	(None, 64)	0	activation_2[0][0]
dropout_3 (Dropout)	(None, 32)	0	activation_5[0][0]
dense_2 (Dense)	(None, 64)	4160	dropout_1[0][0]
dense_4 (Dense)	(None, 32)	1056	dropout_3[0][0]
activation_3 (Activation)	(None, 64)	0	dense_2[0][0]
activation_6 (Activation)	(None, 32)	0	dense_4[0][0]
dropout_2 (Dropout)	(None, 64)	0	activation_3[0][0]
dropout_4 (Dropout)	(None, 32)	0	activation_6[0][0]
concatenate_1 (Concatenate)	(None, 96)	0	dropout_2[0][0] dropout_4[0][0]
dense_5 (Dense)	(None, 2)	194	concatenate_1[0][0]



Currently training and optimizing two **convolutional neural nets**, with their respective outputs flattened and fed into a final densely connected layer.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 25, 25, 25, 1)	0	
input_2 (InputLayer)	(None, 5, 5, 60, 1)	0	
conv3d_1 (Conv3D)	(None, 12, 12, 12, 10	280	input_1[0][0]
conv3d_3 (Conv3D)	(None, 4, 4, 28, 10)	210	input_2[0][0]
dropout_1 (Dropout)	(None, 12, 12, 12, 10	0	conv3d_1[0][0]
dropout_3 (Dropout)	(None, 4, 4, 28, 10)	0	conv3d_3[0][0]
conv3d_2 (Conv3D)	(None, 5, 5, 5, 40)	10840	dropout_1[0][0]
conv3d_4 (Conv3D)	(None, 3, 3, 12, 40)	8040	dropout_3[0][0]
flatten_1 (Flatten)	(None, 5000)	0	conv3d_2[0][0]
flatten_2 (Flatten)	(None, 4320)	0	conv3d_4[0][0]
dropout_2 (Dropout)	(None, 5000)	0	flatten_1[0][0]
dropout_4 (Dropout)	(None, 4320)	0	flatten_2[0][0]
dense_1 (Dense)	(None, 2)	10002	dropout_2[0][0]
dense_2 (Dense)	(None, 2)	8642	dropout_4[0][0]
concatenate_1 (Concatenate)	(None, 4)	0	dense_1[0][0] dense_2[0][0]
dense_3 (Dense)	(None, 2)	10	concatenate_1[0][0]

Further Work

- Refine discrimination of objects at low-pT, and determine not only object type, but also energy

- Move to multi-object environments

- Apply methods to calorimeter geometries in ATLAS environment

- Use methods in low-Δm analysis