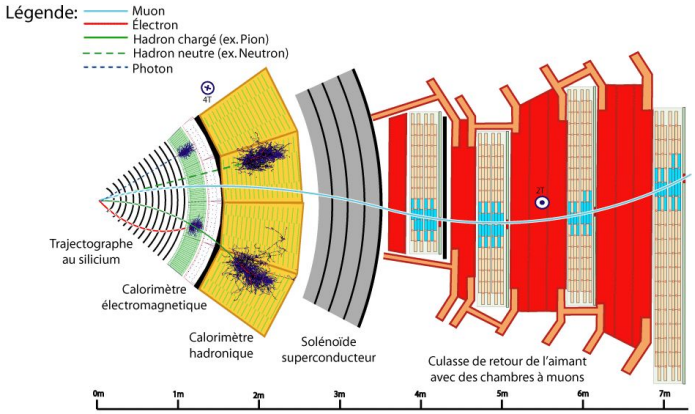
Brandeis UNIVERSITY

Update on Particle ID

Dowling Wong, Patrick McCormack

Aug.2023

The ultimate goal for our work is to build a particle ID like tagging system of particle flows at CMS, that use tracker, calorimeter, and hodoscope information to reconstruct and tag particles in each single event. In our plan, it will end up with two separate algorithms, one is for particles that have full track so that E/p, and the other is for only stations after st3_tracklet.



A considerable part of our work is exploratory, there might exist flaws , verbosities and errors in our idea and code.

Developed on previous work

Exploratory work, developing freshly

v1.0

The basic

v2.1

NN ID for single particle, with track and EMCal

v2.2

V2.1, but for multi particles.

Cut-based ID

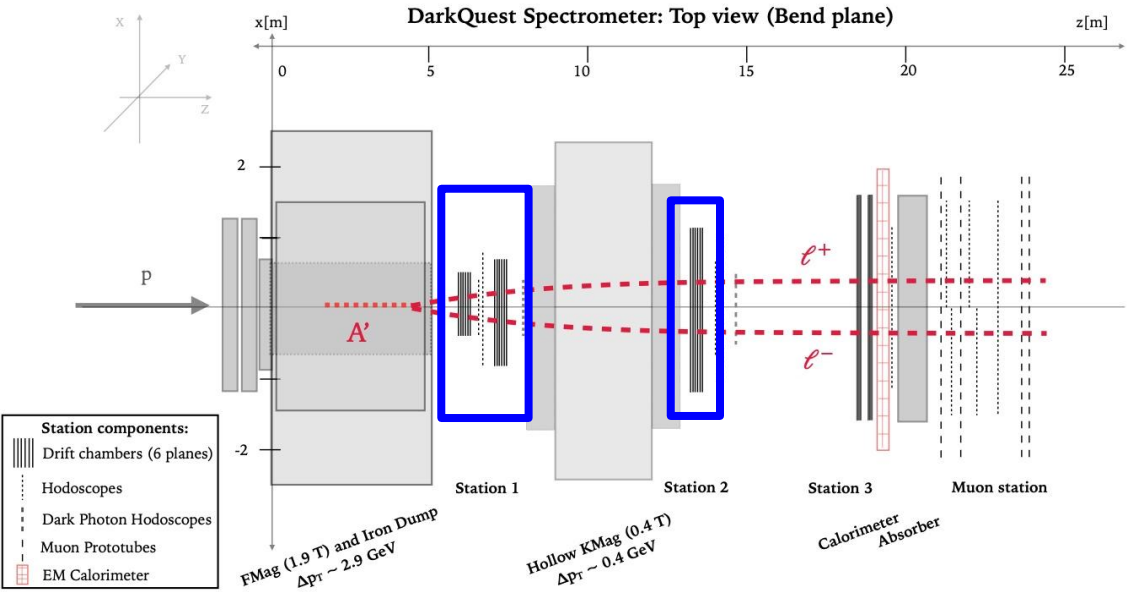
NN-based ID

v2.0

NN ID for single particle, within only EMCal

v2.3

Extend lifetime, while trying to keep high efficiency. 2.3.1 and 2.3.2 working in parallel.



ID v1.0

ID v1.0 is a cut based ID for e+e-,

Works for single particle gun with full track(tracklet 1,2&3 to measure momentum). All quantities are measured for the leading cluster.

W_ew (x+y)	Width (x+y)	Dist betw cts st3_emcal (x+y)

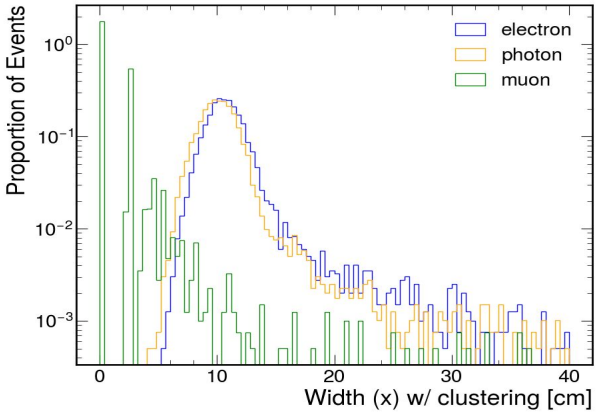
—

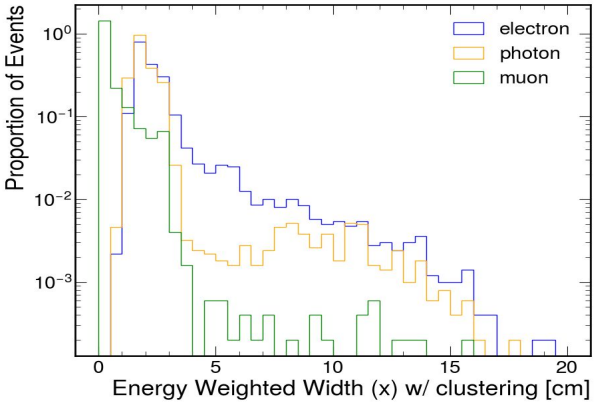
methods

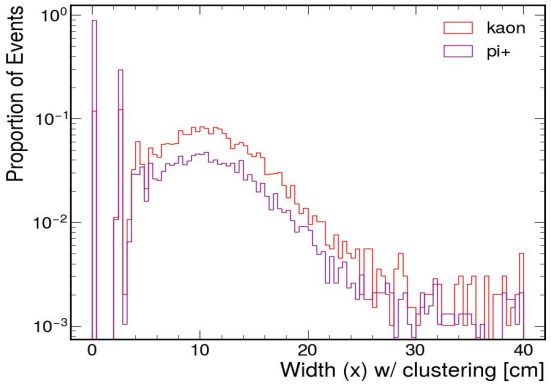
This particle ID using 10,000 single particle gun samples for μ-, π+, π-, π0, γ, K^0_L, e+, e- to construct a cut-based ID for μ-, π+, π-, π0, γ, K^0_L V.S. e+, e-.

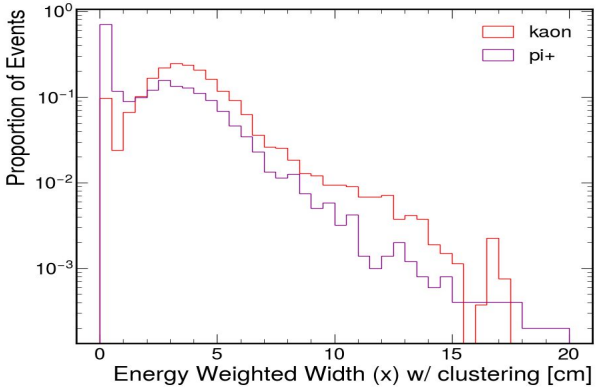
By manually adjust cuts on distribution plot(next page) to find the best choice. It's kinda like a trade off, depends on needs.

Width and W_EW for the leading cluster



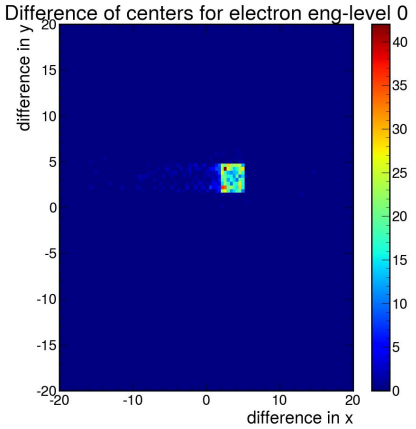


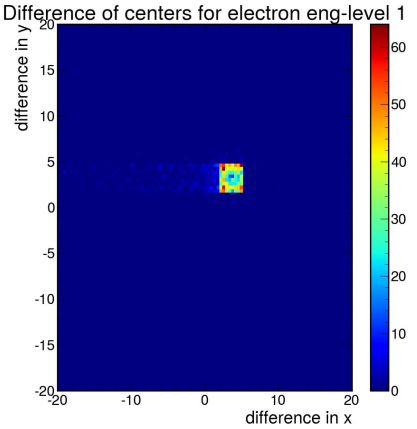


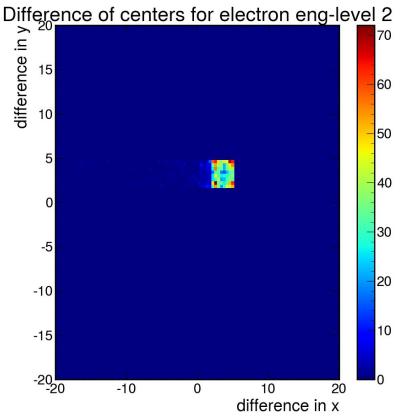


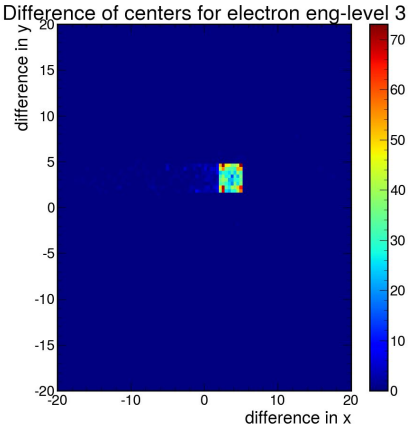
Distance between EMCal leading cluster center and extrapolated point on EMCal from st3_tracklet highly depends on valid points on both EMCal and track, valid rate are listed below.

	valid comparison
muon	99.60%
electron	95.70%
positron	96%
photon	3.70%
pi+	95.10%
pi-	96.50%
klong	0.60%
pi0	4.50%

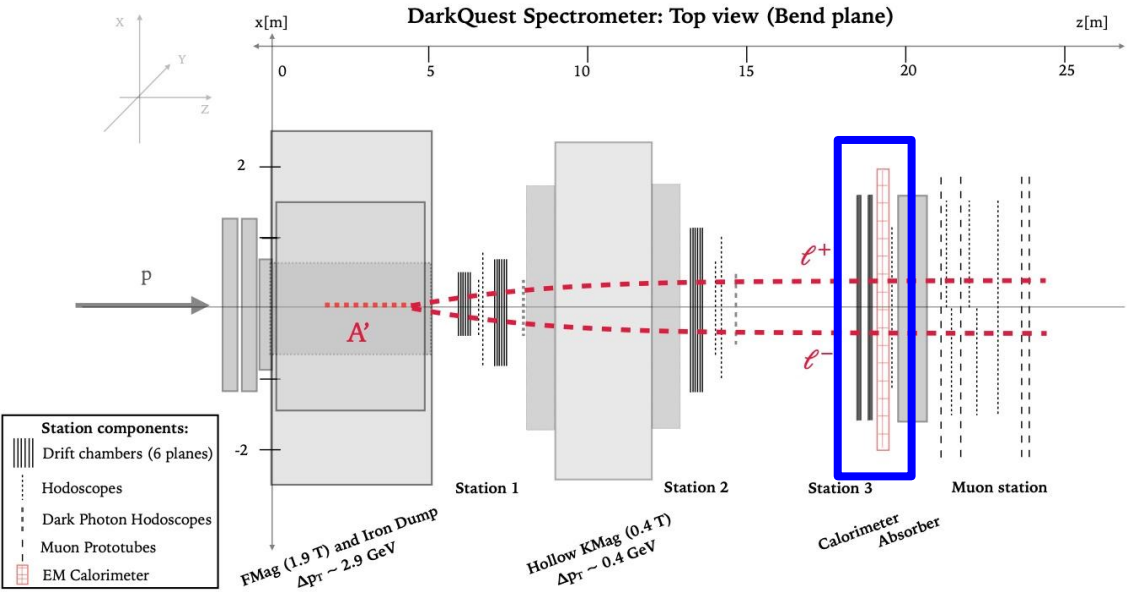








	(x+y)wew+diff+wid: pass rate
muon	0
electron	59.50%
positron	55.70%
photon	0.40%
pi+	3.40%
pi-	3.80%
klong	0
pi0	0.20%

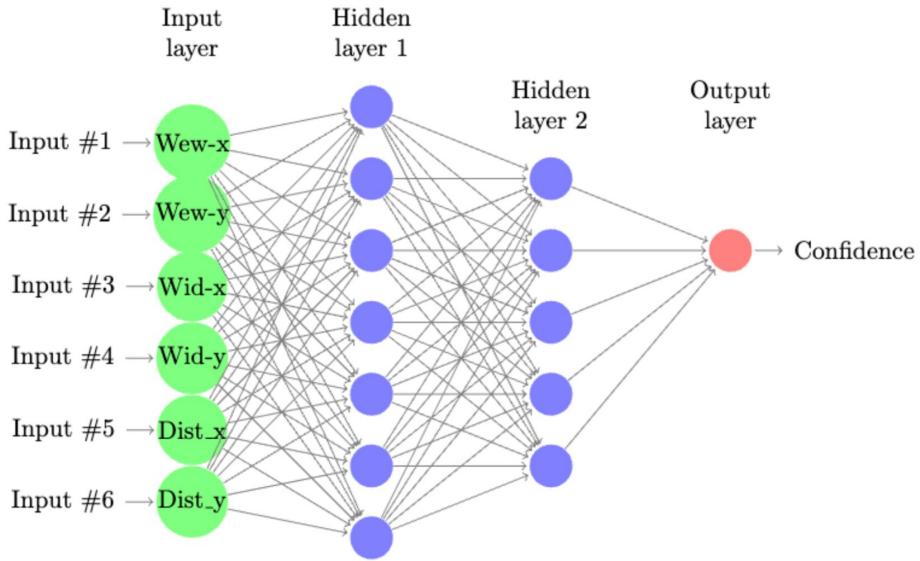


ID v2.0

ID v2.0 has the same input as v1.0. Mostly focused on data processing , to make it readable as input.

W_ew (x+y)	Width (x+y)	Dist betw ctrs st3_emcal (x+y)

I have implemented two Deep Neural Network in both pytorch and Keras, they have been implemented with identical configuration.



Used sigmoid as forward function and hidden layers of 50, 25 neurons. Activationfunctions are ReLU, ReLU and sigmoid to the output layer. The output is a normalized confidence level between 0 to 1.

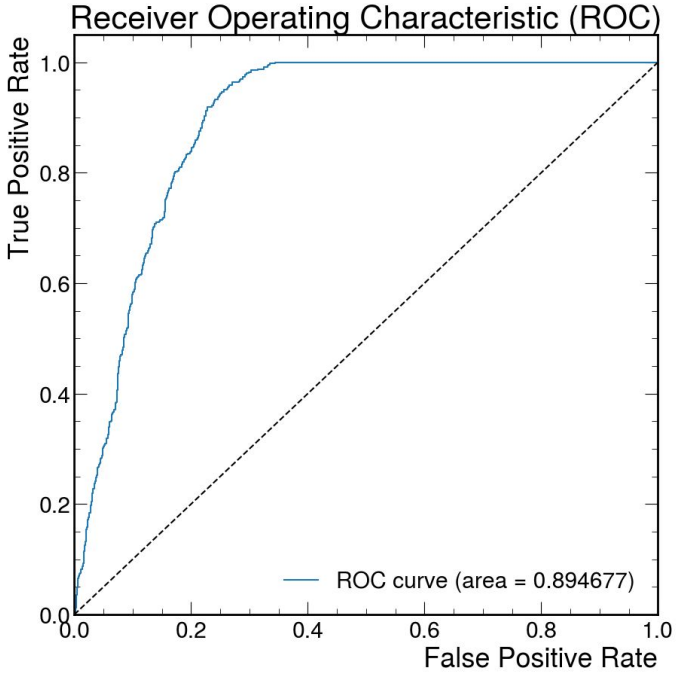
Pytorch

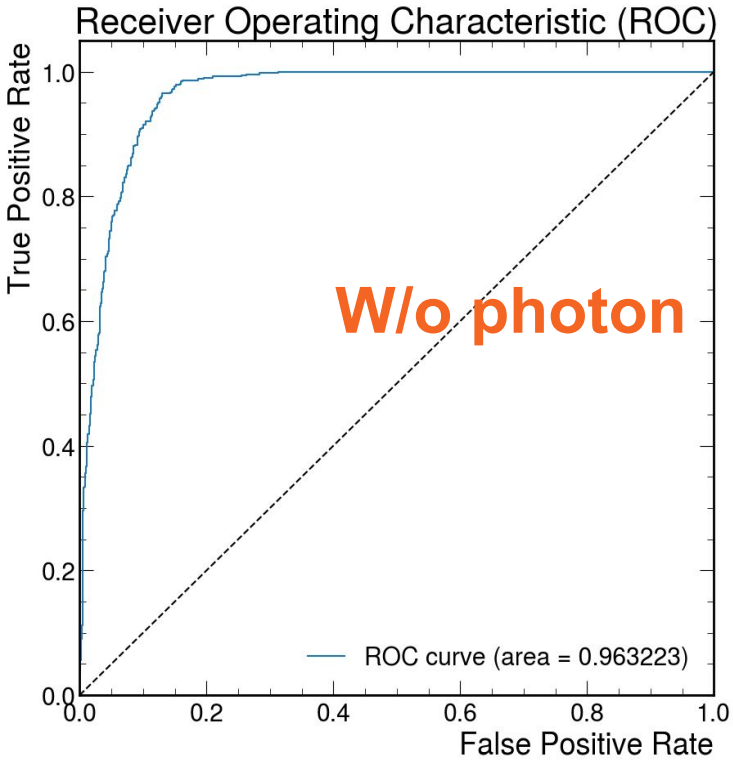
```
class Net(nn.Module):
  def __init__(self, in_count, out_count):
    super(Net, self).__init__()
    self.fc1 = nn.Linear(in_count, 50)
    self.fc2 = nn.Linear(50, 25)
    self.fc3 = nn.Linear(25, out_count)

  def forward(self, x):
    x = F.relu(self.fc1(x))
    x = F.relu(self.fc2(x))
    return torch.sigmoid(self.fc3(x))
```

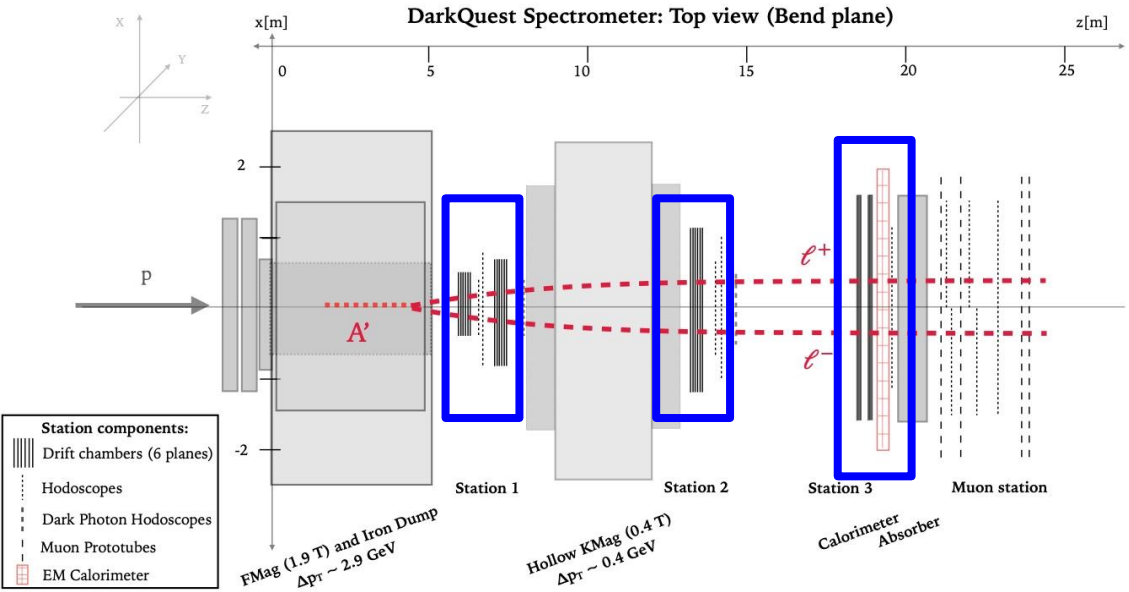
Keras:

```
model = Sequential()
model.add(Dense(100, input_dim=x.shape[1], activation='relu',
        kernel_initializer='random_normal'))
model.add(Dense(50,activation='relu',kernel_initializer='random_normal'))
model.add(Dense(25,activation='relu',kernel_initializer='random_normal'))
model.add(Dense(1,activation='sigmoid',kernel_initializer='random_normal'))
model.compile(loss='binary_crossentropy',
      optimizer=tensorflow.keras.optimizers.Adam(),
      metrics =['accuracy'])
```





This is the ROC curve for particle ID simply EMCal and st3_tracklet. It turns out not even good as cut based. Guess because absence of full track make photon hard to exclude.



ID v2.1

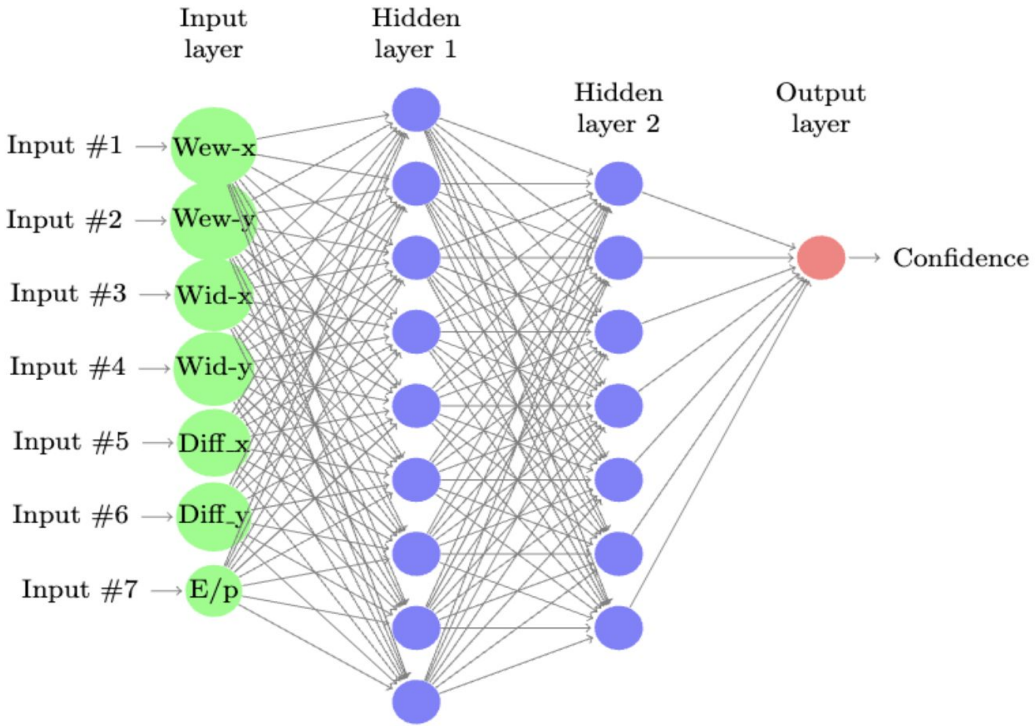
ID v2.1 is a NN based ID for e+e-,

Works for single particle gun with full track(tracklet 1,2&3 to measure momentum).

W_ew (x+y)	Width (x+y)	E/p	Dist betw cts st3_emcal (x+y)

	label	wew_x	wew_y	wid_x	wid_y	diff_x	diff_y	E/p
0	1.0	1.584334	1.465552	11.281222	9.981665	3.620048	3.764283	0.989164
1	1.0	2.007739	1.524296	8.507692	8.338515	2.570766	2.901524	1.033287
2	1.0	3.126703	2.135596	10.663046	10.803723	3.276891	1.865097	1.179087
3	1.0	5.345500	1.823037	12.140334	12.363477	-3.691071	2.315703	2.588458
4	1.0	2.276824	1.549370	11.703530	10.924367	2.272942	2.970442	1.035610
...	...	...	...	...	...	...	...	...
7995	0.0	2.525472	2.587707	10.404633	6.879240	-9999.000000	-9999.000000	-1.000000
7996	0.0	3.554478	4.037831	11.389580	13.993940	-9999.000000	-9999.000000	-1.000000
7997	0.0	-1.000000	-1.000000	-1.000000	-1.000000	-9999.000000	-9999.000000	-1.000000
7998	0.0	6.758660	6.096007	9.880163	11.339236	-9999.000000	-9999.000000	-1.000000
7999	0.0	-1.000000	-1.000000	-1.000000	-1.000000	-9999.000000	-9999.000000	-1.000000

8000 rows × 8 columns

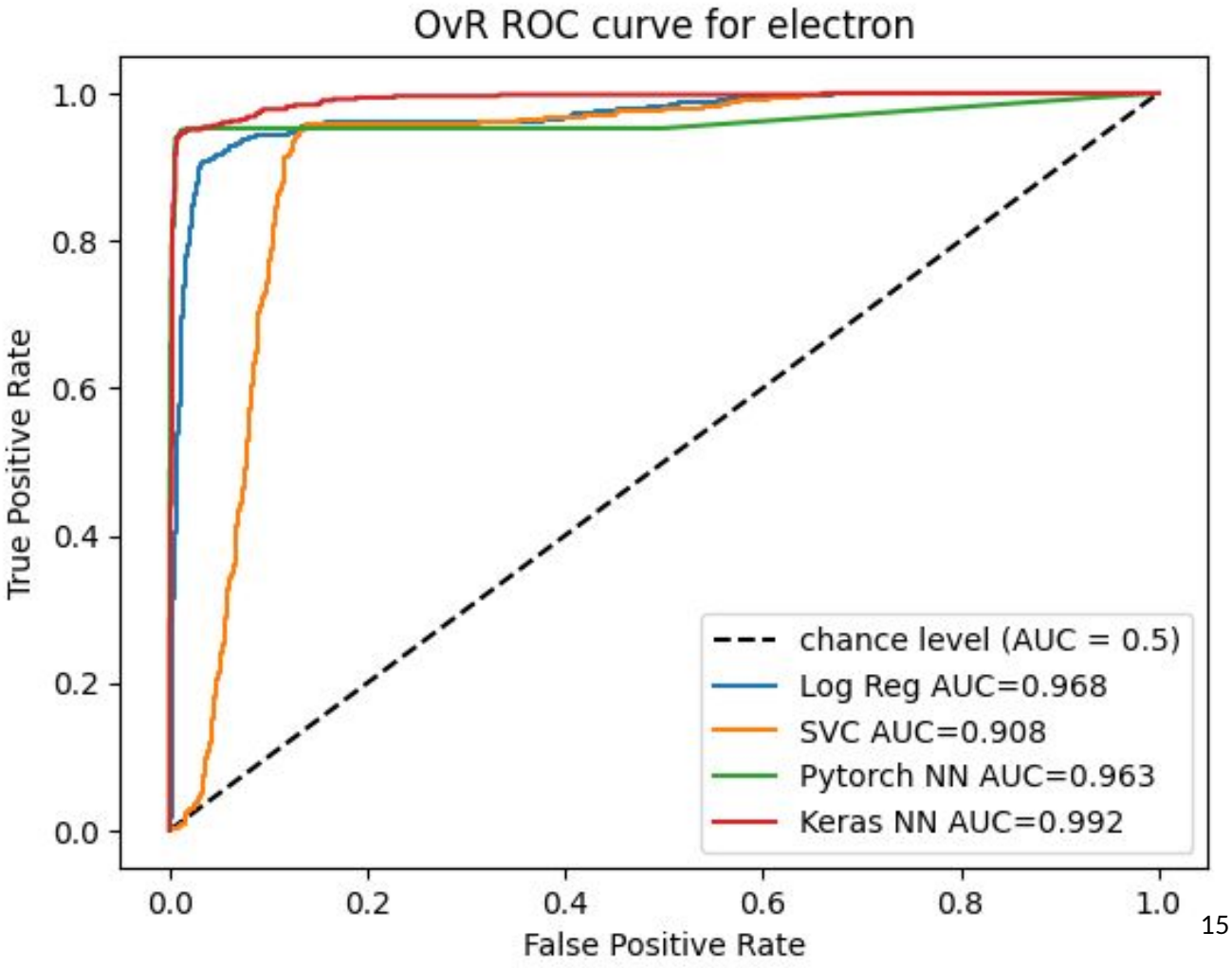


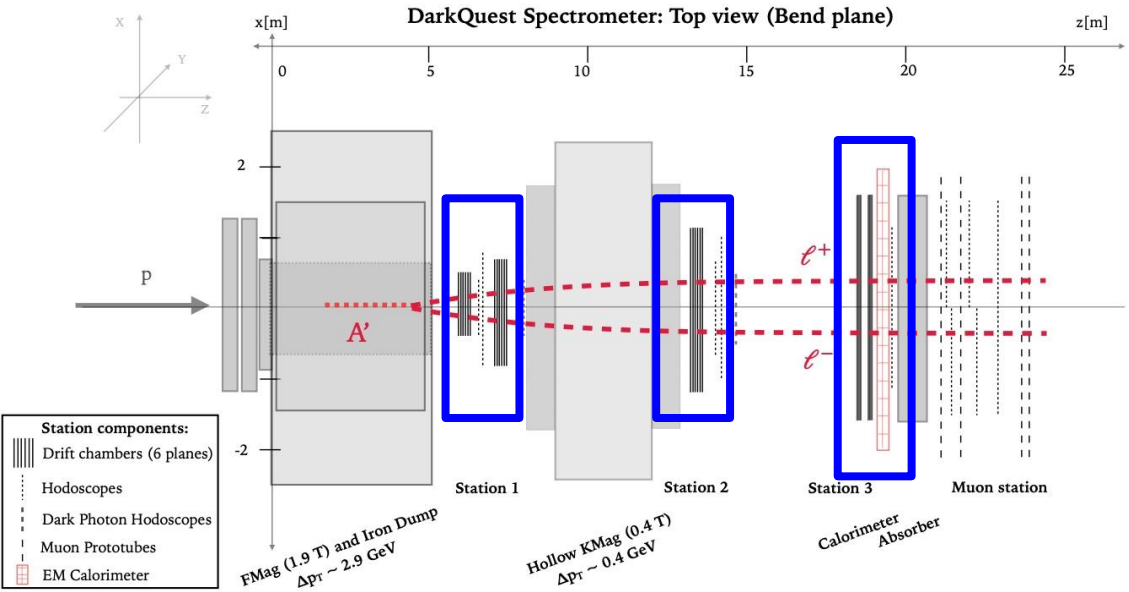
Used sigmoid as forward function and hidden layers of 50, 25 neurons. Activationfunctions are ReLU, ReLU and sigmoid to the output layer. The output is a normalized confidence level between 0 to 1.

You may noticed -9999 in Wew, width and -1 in Diff. They are place holders for evt that doesn't have EMCal record and tracklet record.

Looks pretty good? But actually has several flaws:

1. Only work for single particle gun.
2. Need full track.
3. Only for dielectron but not dimuon.



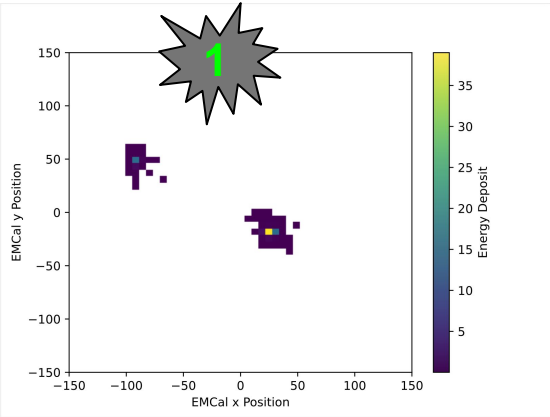


ID v2.2

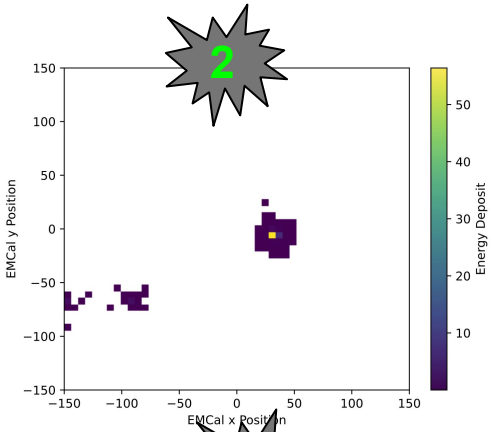
ID v2.2 has basically the same feature as v2.1. But modified to fit multi-cluster situation. Testing with A' -> ee samples.

W_ew (x+y)	Width (x+y)	E/p	Dist betw cts st3_emcal (x+y)

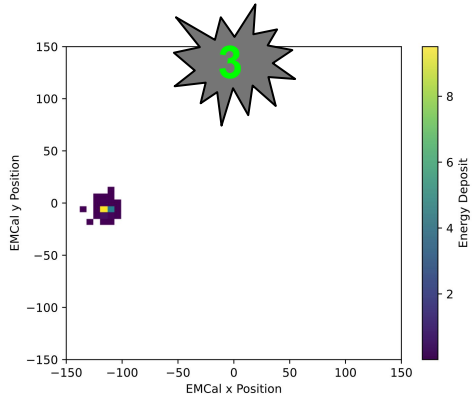
Difficulty been exponentially increased on aspects of clustering and matching up, depends on complexity of EMCal.



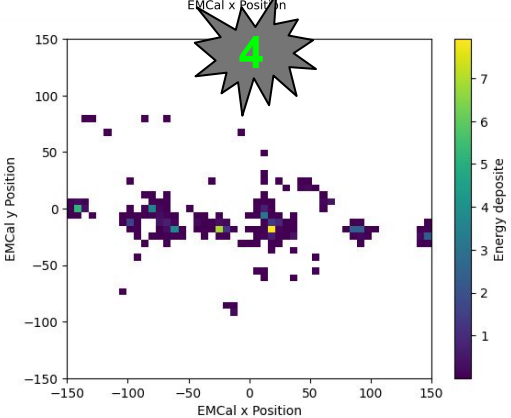
Pretty 2 cluster.



Decayed, but one at the edge.



Decayed, but only captured one

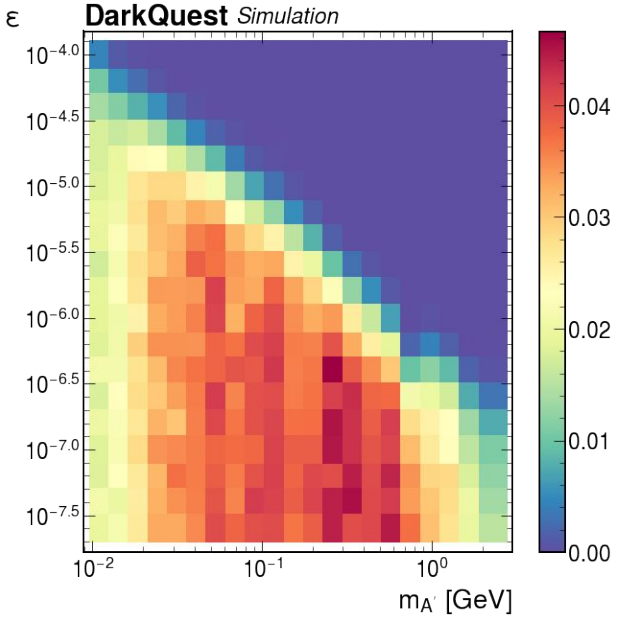


Decayed but brem showered. Common on low coupling.

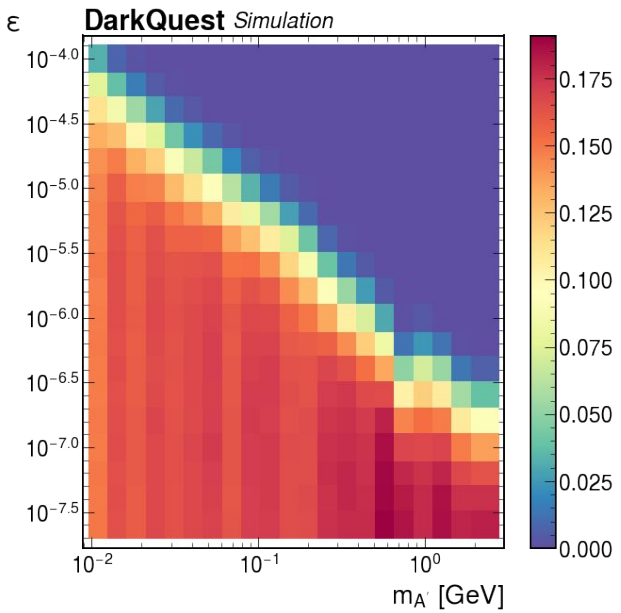
*From Duc

Track & Trigger acceptance (all 200-600cm)

Track acceptance (events with 2 tracks)



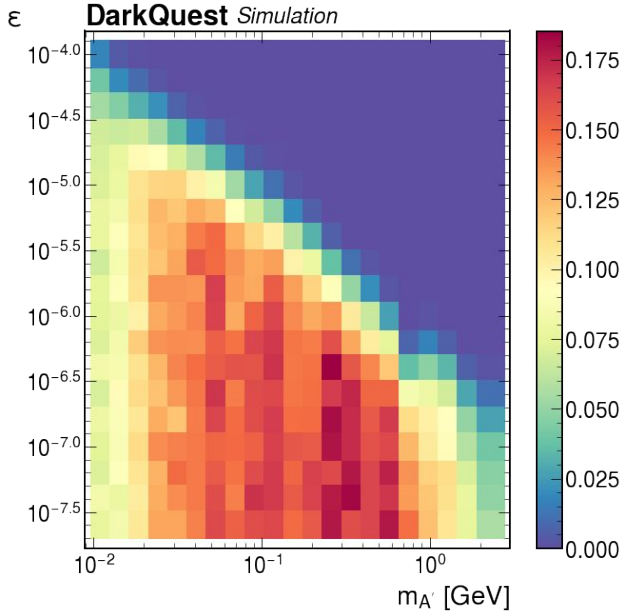
Trigger acceptance: (ECAL Trigger (sum > 2GeV))



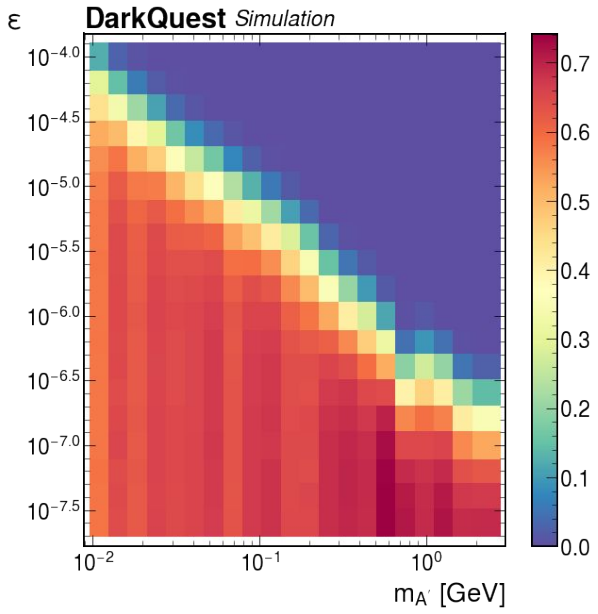
*From Duc

Track & Trigger acceptance (all 500-600cm)

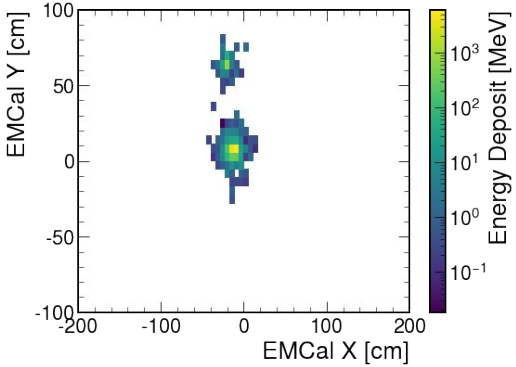
Track acceptance (events with 2 tracks)

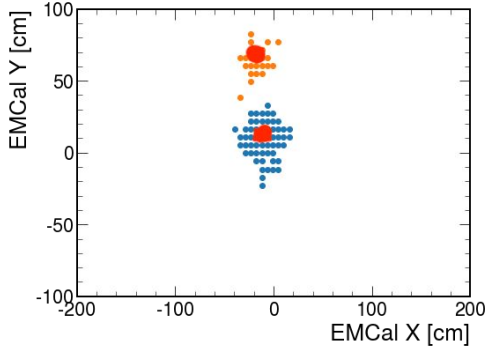


Trigger acceptance: (ECAL Trigger (sum > 2GeV))



1. Semi-customized Clustering





Depends on the decay z, mass and coupling, Aprime may has very different pattern on EMCal.

Seed energy highly depends on the energy level of A'. Use a low threshold of eng and filter by radius.

Once get array of seeds, #of seeds passed as # of clusters for Kmeans, and an input for NN.*

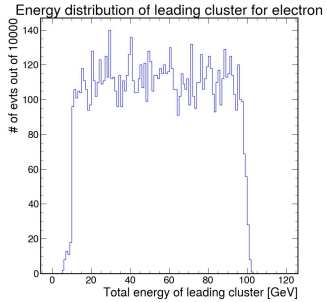
Find all hits above 0.8 GeV, sort by eng decrease order, calc dist

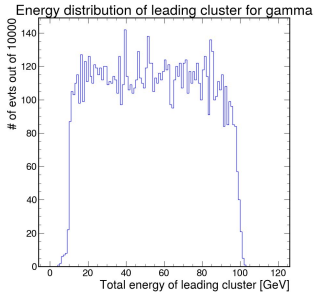
min_energy = 0.8 GeV

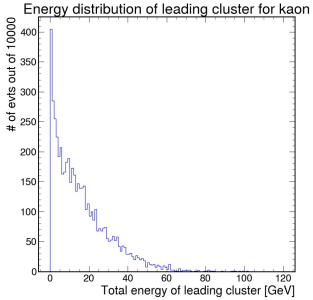
seed_radius = 15 cm

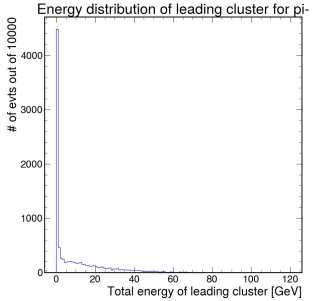
With this setup even situation 4 could be handled.

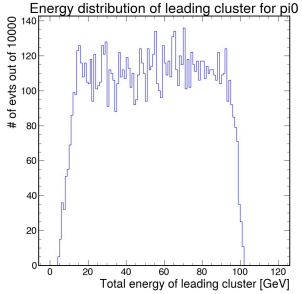
Energy distribution of single particle gun: importance of setting low eng threshold:

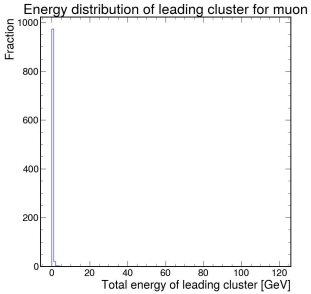


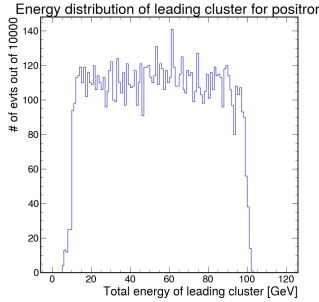


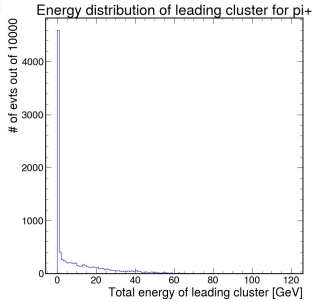




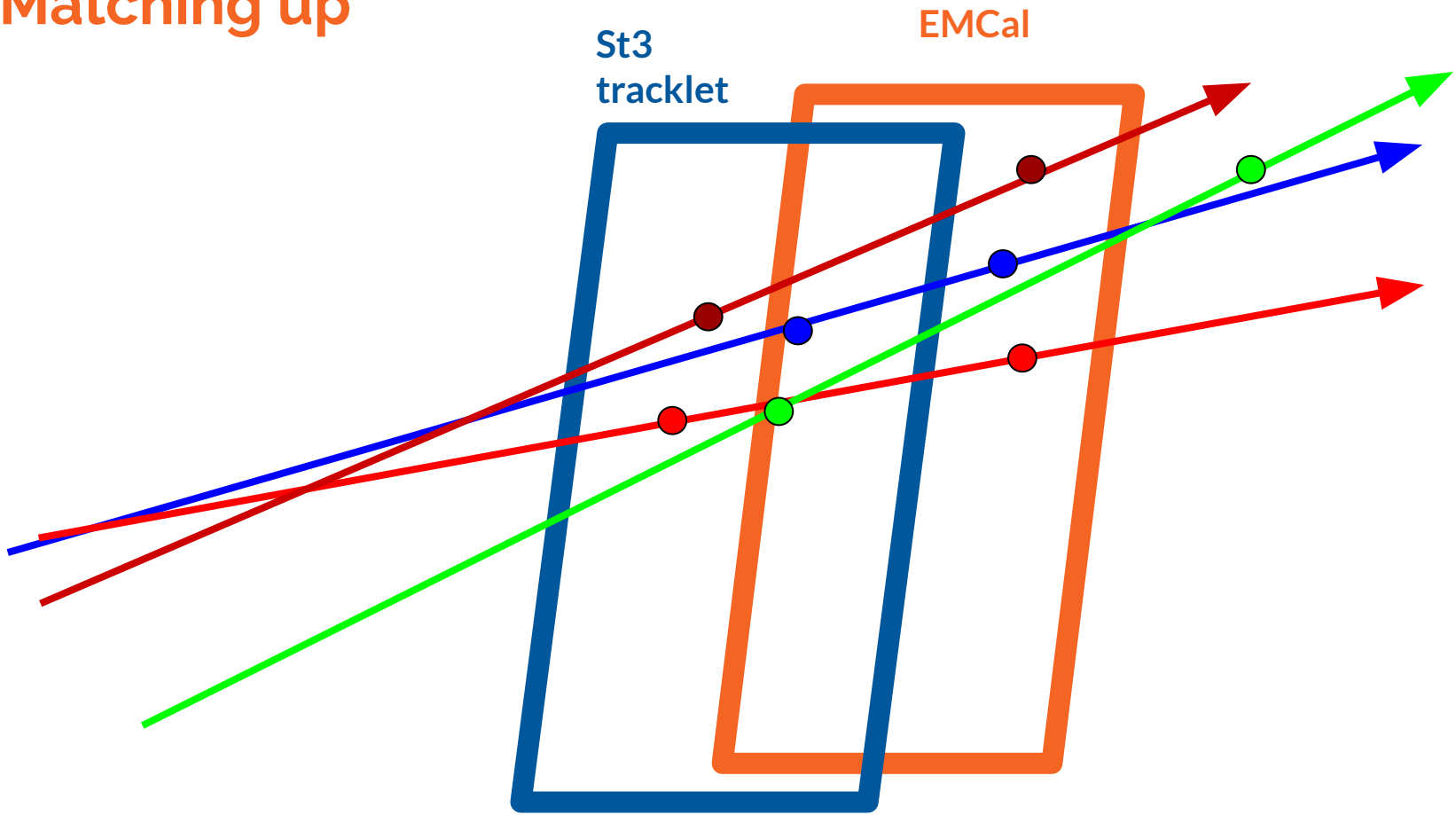


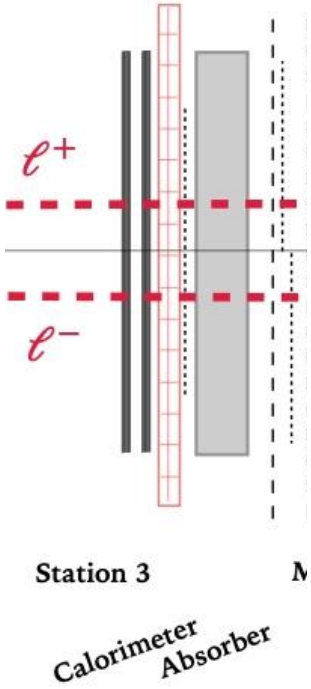


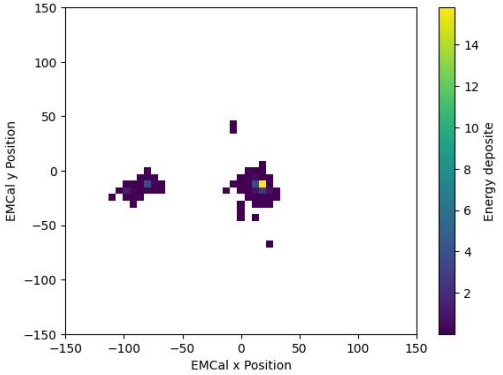


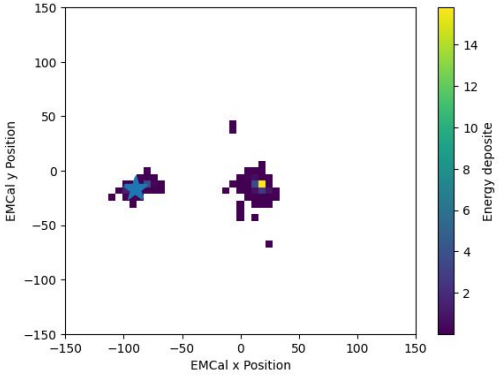


2. Matching up





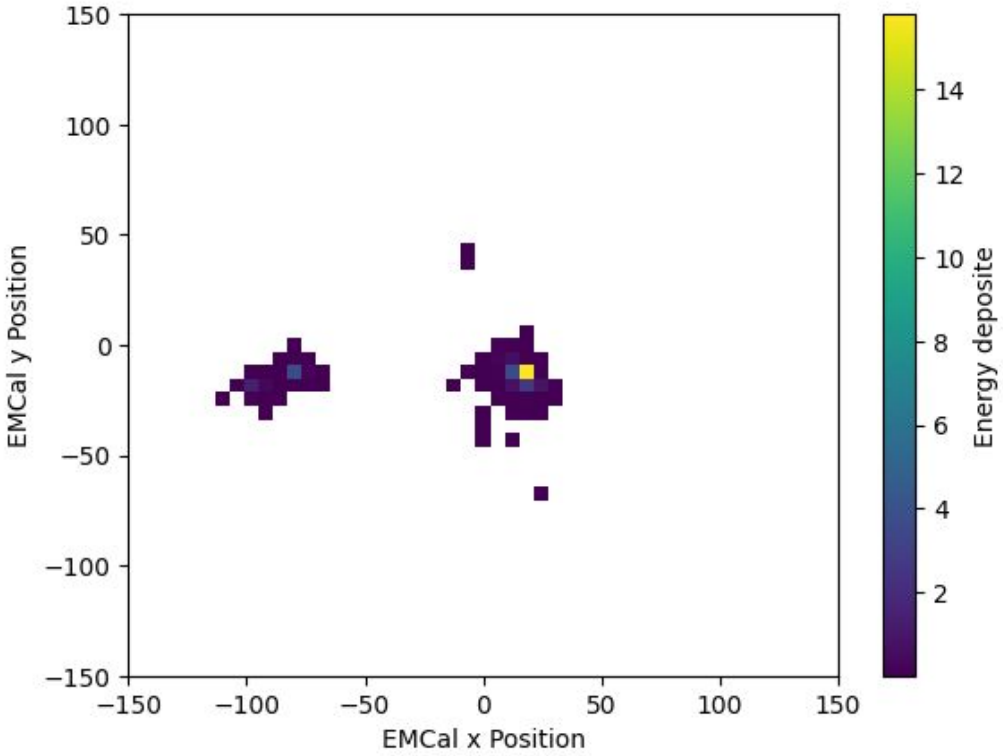




Double_t track_x_CAL = track_x_st3 + (track_px_st3 / track_pz_st3) * (1930. - track_z_st3);
Double_t track_y_CAL = track_y_st3 + (track_py_st3 / track_pz_st3) * (1930. - track_z_st3);

- EMCal is still the primary information we have
- Sort clusters by energy decrease order
- Match the closest tracklet-extrapolation point, if <15cm, then match tracklet with cluster. Delete this tracklet from array.

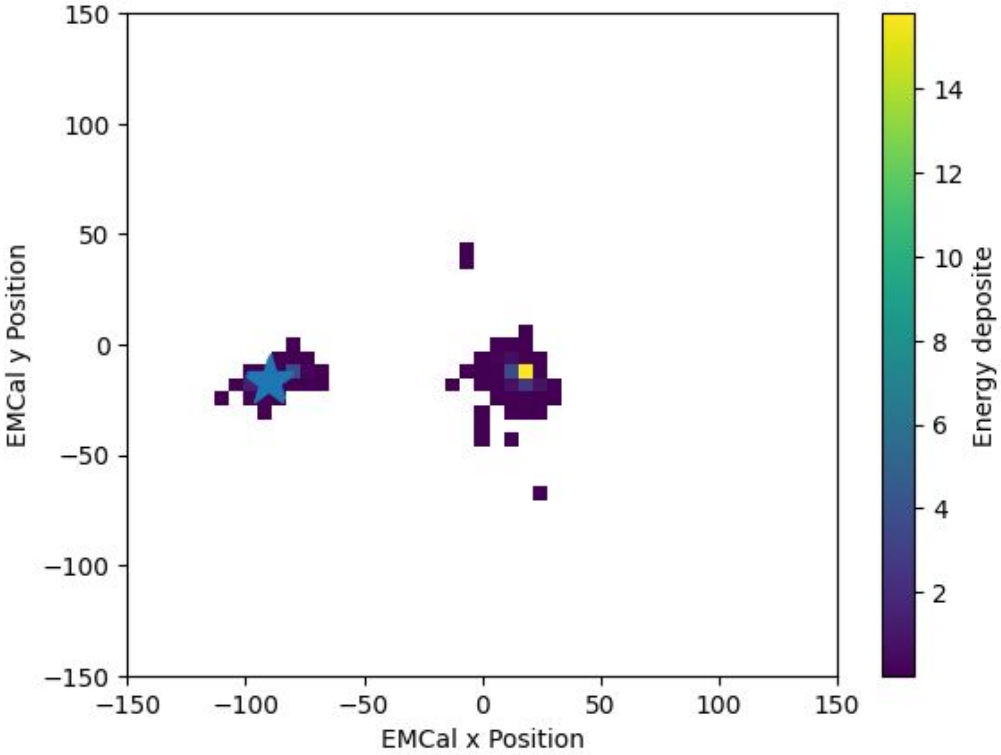
Let’s take look on example.



Our primary criterion is still EMCal, using number of clusters(seeds), to determine how many particles we need to reconstruct.

Higher energy cluster has higher priority for choosing track. If no track matched, then just append a dummy variable.

Let's take look on example.



Larger Energy Seeds:
P1: (15.67, -11.51, 31.03)
P2: (-78.33, -11.51, 3.67)
P3: (-94.92, -17.04, 2.69)

St3_tracklet extap to EMCal:
***Labeled with blue star**
S1: [-89.85, -15.58]
S2: [-90.23, -16.90]

Then, P2 matched to E1, P3 matched to E2.
P1 may recognized to be a photon

—

Components of V2.2 has implemented, integrating and testing with various A'->ee sample. Kinda a hard march, since making it accurate as well as time/RAM efficient.

And yes, z-500-600 cm with all mass and epsilon, currently looping to generate on spinquest01.

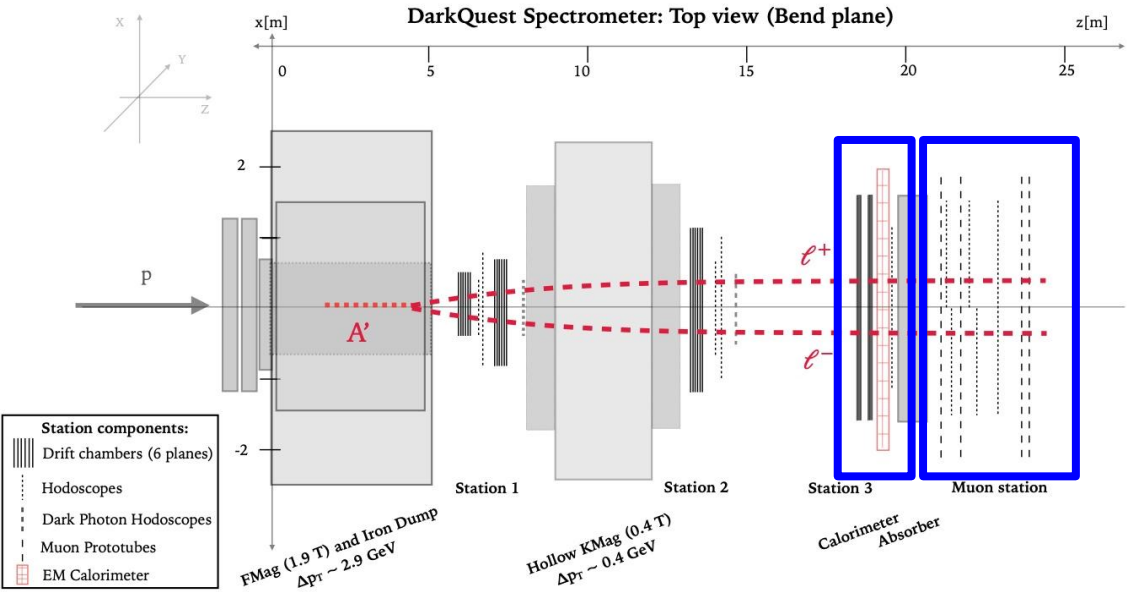
Note from 8.17:
Version 2.2 put into work

Find 27% samples with 2 seeds on EMCal, under mass=2.3, eps=e-3.3

Manually seen approx 32% evts has beautiful 2 clusters.

Number of seeds highly depends on eng_min and radius. Let's say st23_tracklet extends 651 evts to EMCal.

Then we adjust emin and radius to make the evt has at least one seed approx 651.

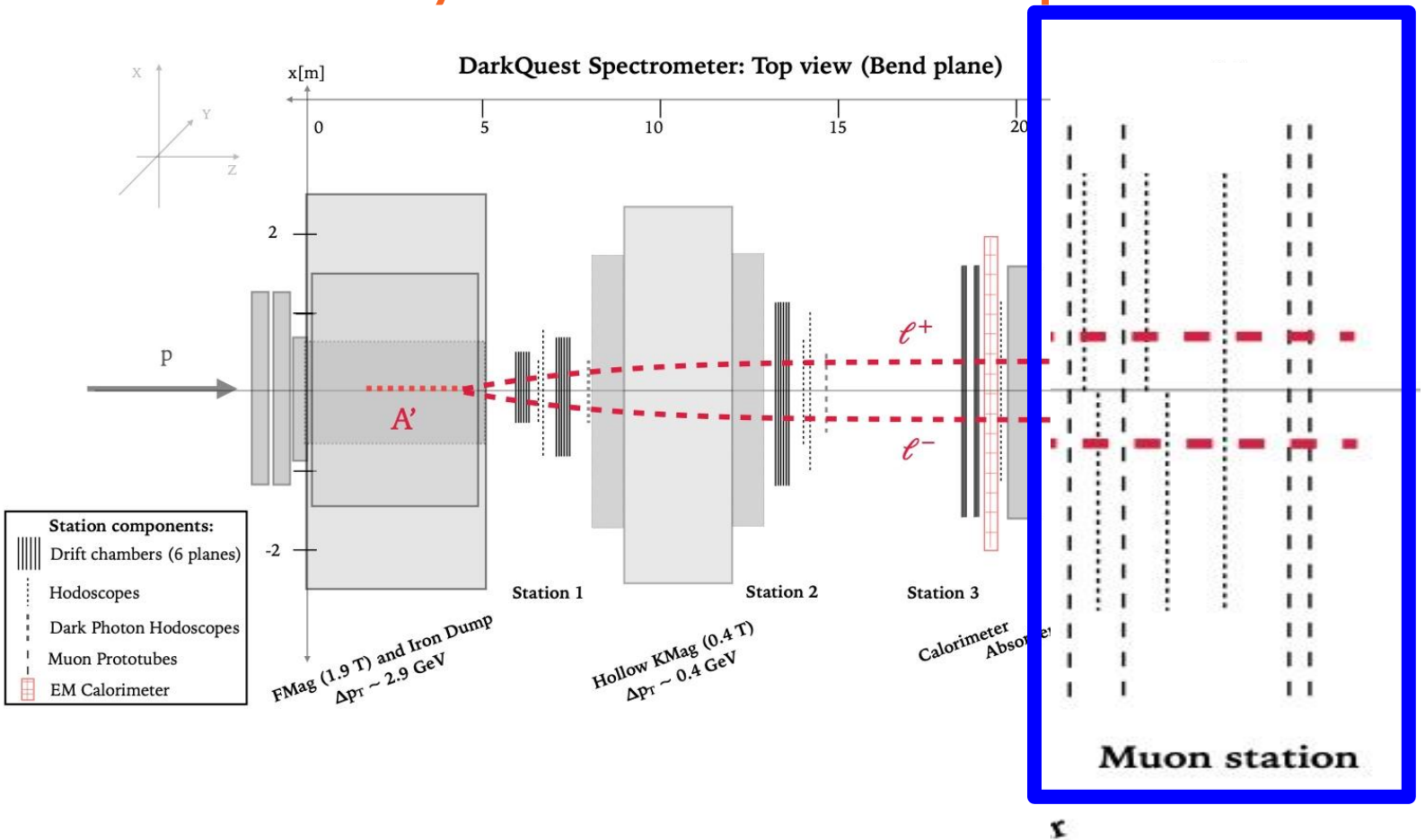


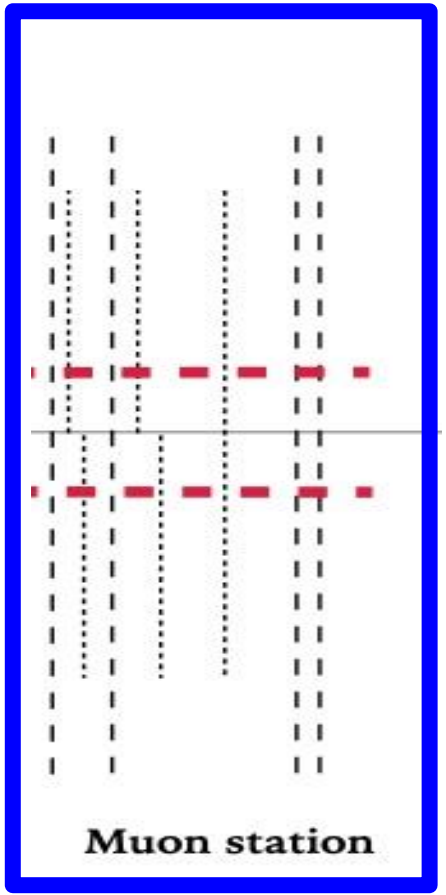
ID v2.3.1

ID v2.3.1 is a NN-based ID work for multi-particle w/o full track

W_ew (x+y)	Width (x+y)	# of tracklets at st3	# of clusters on EMCal	Dist betw cts st3_emcal (x+y)	# of hits on h4	Dist betw cts st3_h4 (x or y)

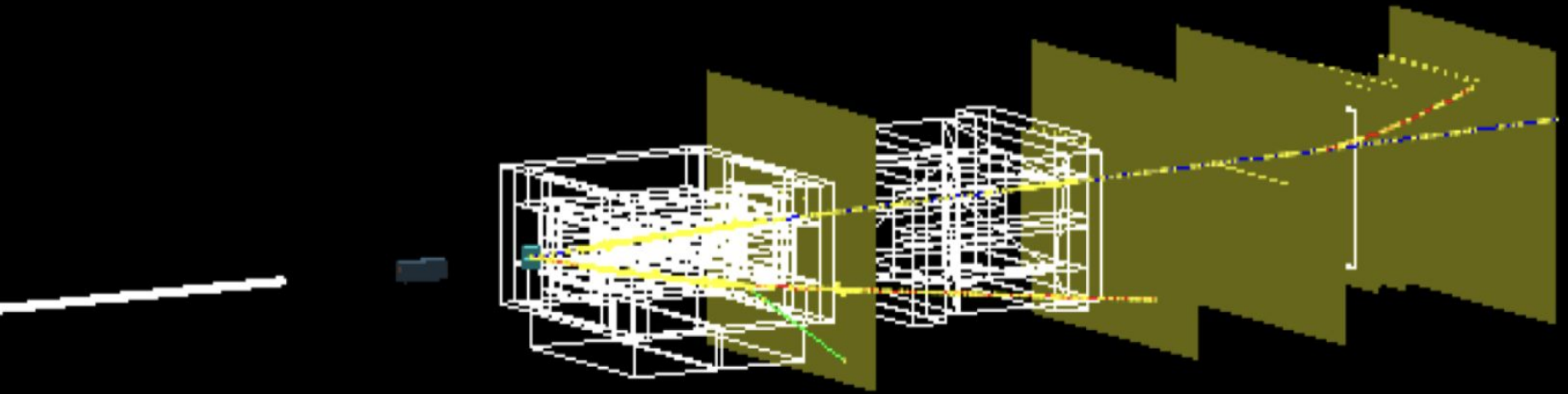
The main difficulty is at station 4 hodoscope

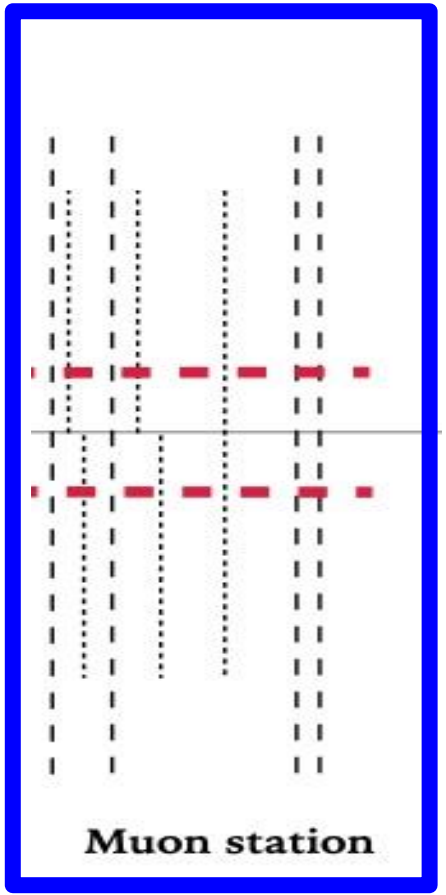




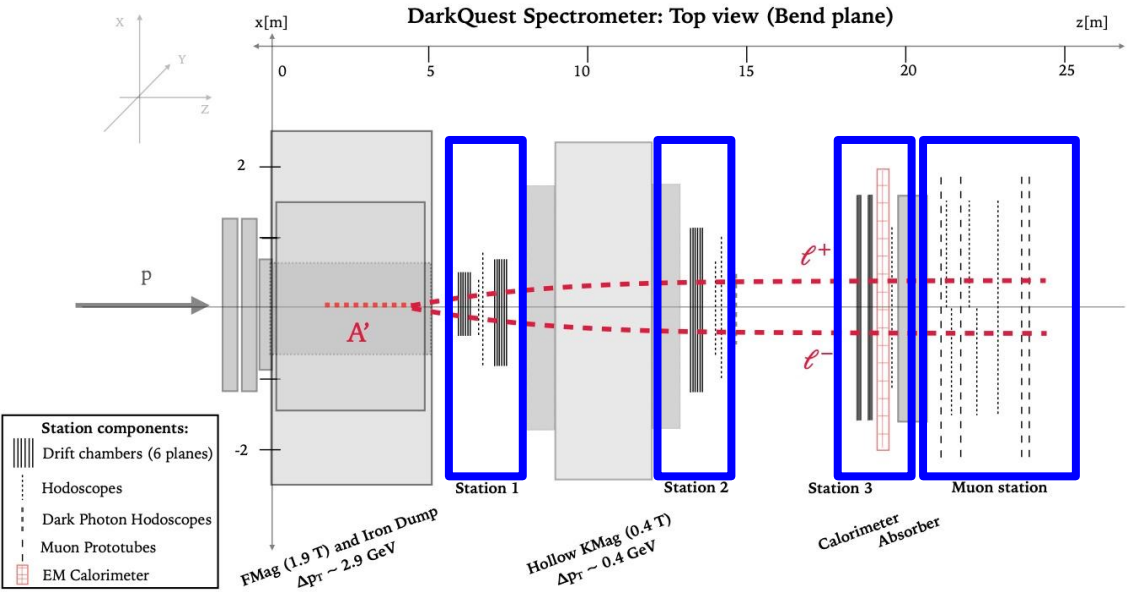
detID	name	z-pos	measures
41	H4Y1L	2130.27	y
42	H4Y1R	2146.45	y
43	H4Y2L	2200.44	y
44	H4Y2R	2216.62	y
45	H4B	2251.71	x
46	H4T	2234.29	x

Then, how to we match up points on 6-layer hodoscope?





- Some pions and muons can be captured by st4. So adding this st is mainly for recognizing muon.
- We may get multiple hits on multiple layers. We don't know whether they are left by the same particle.
- We still follow the same logic of matching st3tracklet points and EMCal clusters.
- Say if there's a hits on det42, then we extend st3_tracklets' ys to det42's z position. Calculate dist between points, assign the closest h4 to st3_tracklet.



ID v2.3.2

ID v2.3.2 is a NN-based ID work for multi-particle w/ full track

W_ew (x+y)	Width (x+y)	# of tracklets at st3	# of clusters on EMCal	Dist betw cts st3_emcal (x+y)	# of hits on h4	Dist betw cts st3_h4 (x or y)	E/p

Still using EMCal as primary criterion. Match up with track first, then extrapolate to h4 for match up muon station. Use dummy variable as placeholder, rarely has all information.

Added much more calculation, need to simplify current code before testing.

Milestones

v1.0

The most basic one

v2.1

NN ID for single particle, with track and EMCal

v2.2

V2.1, but for multi particles.

v2.0

NN ID for single particle, within only EMCal

v2.3

Extend lifetime, while trying to keep high efficiency. 2.3.1 and 2.3.2 working in parallel.

What people are saying

Please let me know if there's verbosity in ID algorithm.

Developer D, MA

Yes, this is the last slide.

Dwong, Cambridge

Thank you for listening.

Dowling, Waltham

Quotes for illustration purposes only