NN development status

Trg meeting

Yuxin Liu

2023/03/01

A Bug Fixed

If we use ETF in Neuro Trigger :

```
m_T0 = m_eventTime->getBinnedEventT0(Const::CDC);
```

L 859 in CDCTriggerHoughtrafoForETF.cc:

```
 int ft   = associatedTSHitsList[iTrack][iHit]->fastestTime() * 2; // 2ns -> ns
```

But we use m_T0 as:

```
 int t = (m_hasT0) ? priot - m_T0 : 0; // priority in unit of (2ns) while m_T0 in unit of (ns)
```

Not matched at all…

Modified:

```
m_T0 = m_eventTime->getBinnedEventT0(Const::CDC)/2;
```

(Discovered by Sudo-san)

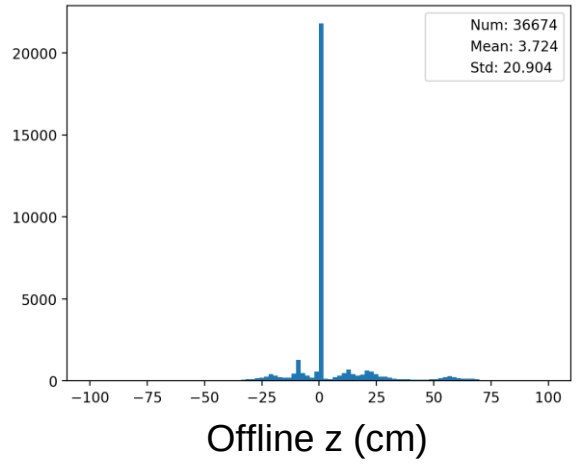
Motivation from …

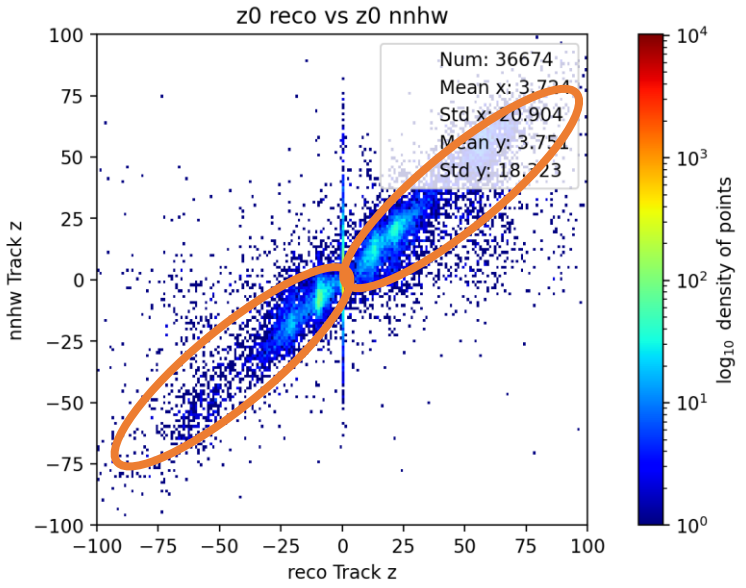
Track from large z seems have different feature comparing with the one from IP and such feature should already included in origin input

→ Why not use NN to directly predict if the tracks are from IP or not?

Control group

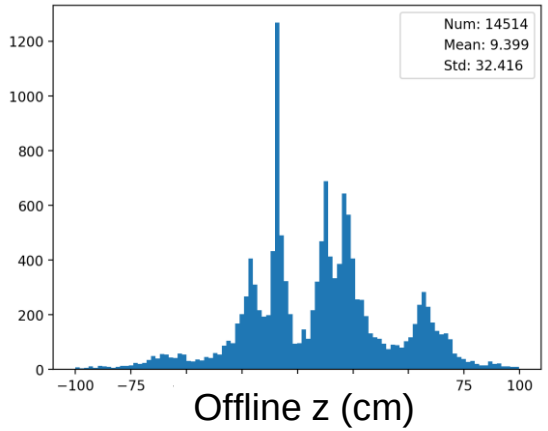
Origin dataset training

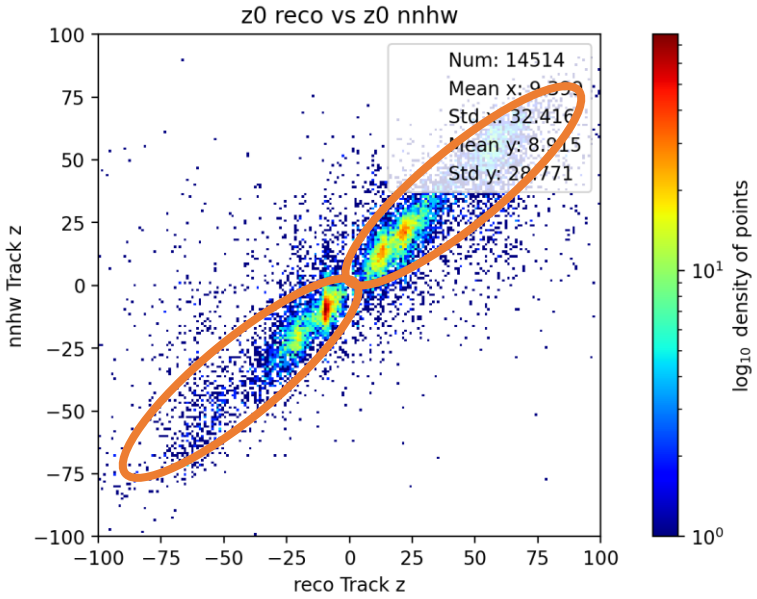




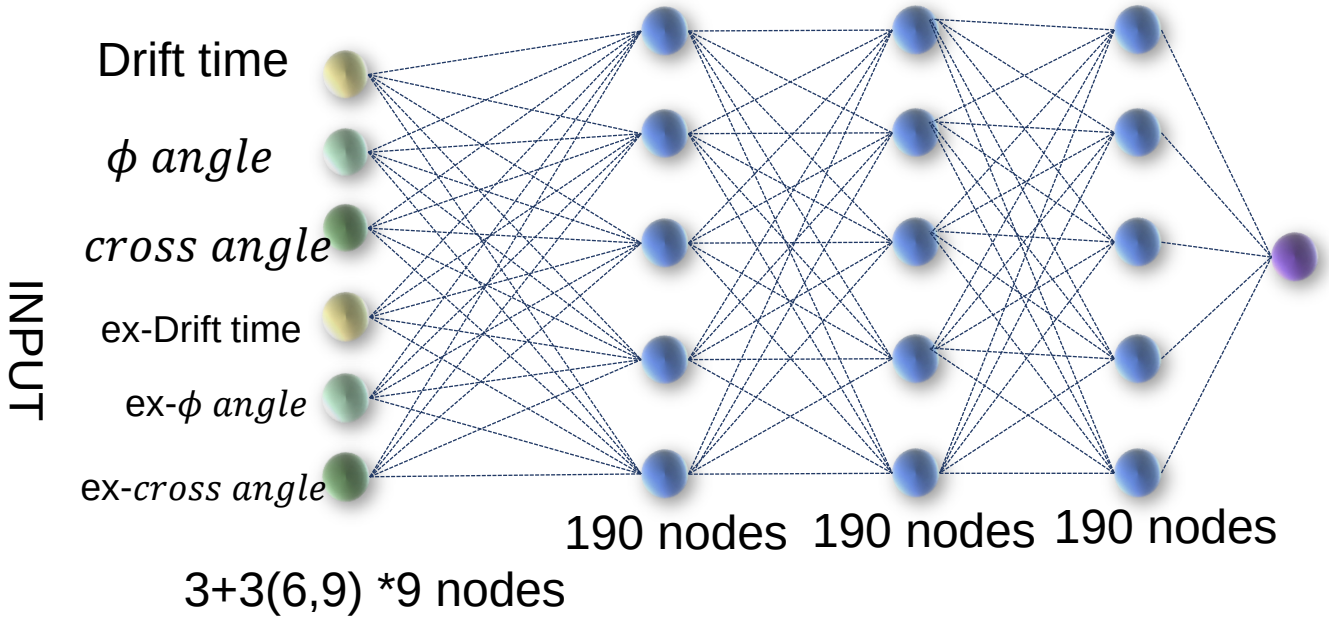
Large z

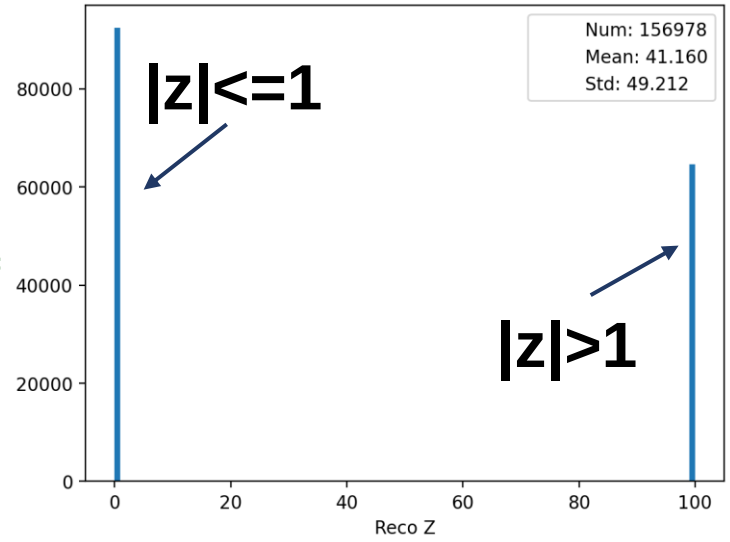
Remove |Z| <1 events training
(left 1/3 events, enough for training)





Start from a simple model





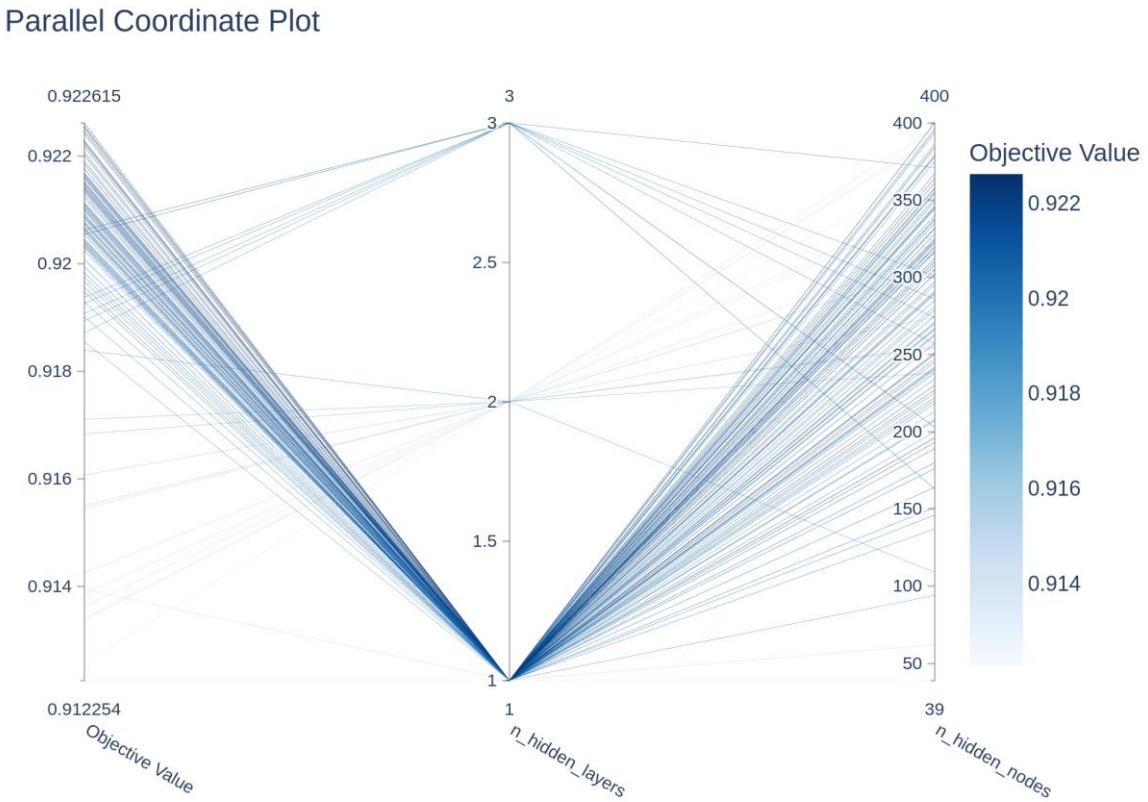
Target distribution

Train & validation set enlarged to full dataset from 1756 -1780 (~ 1.4 M training data and 0.3M validation date, 1781 full data for test

Try only one output first

4

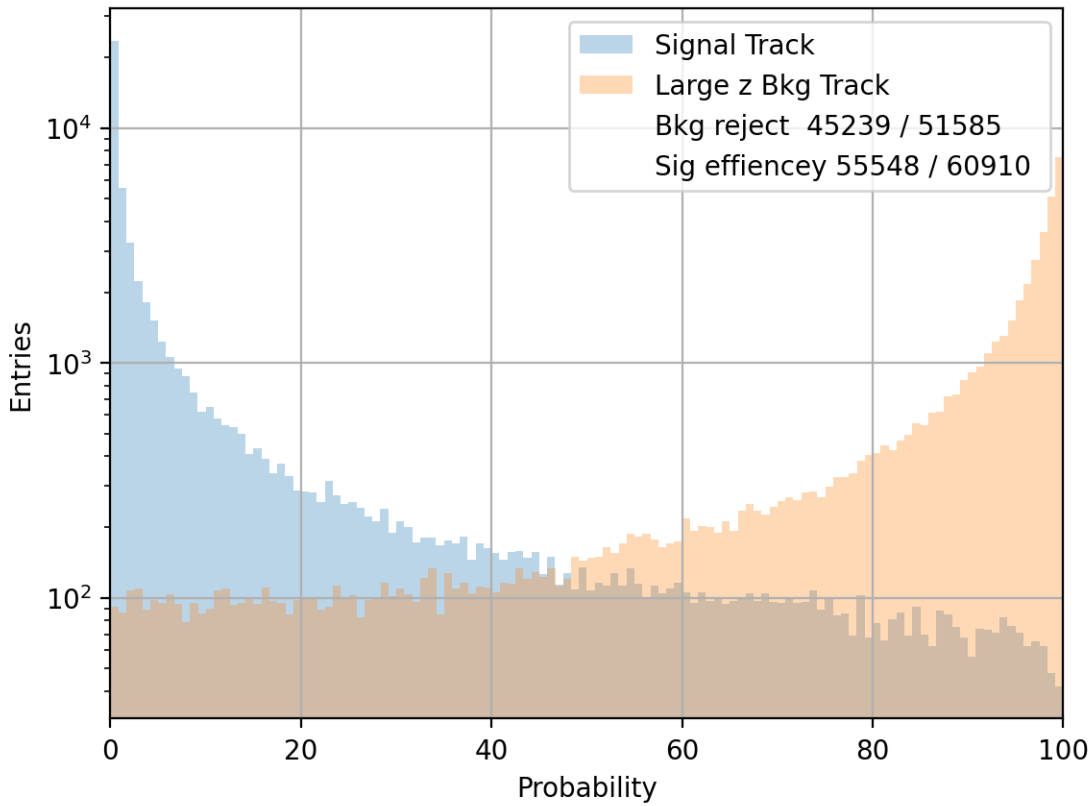
Optimized with optuna



Objective is accuracy (0-1)

1 hidden layer and 319 nodes got best result

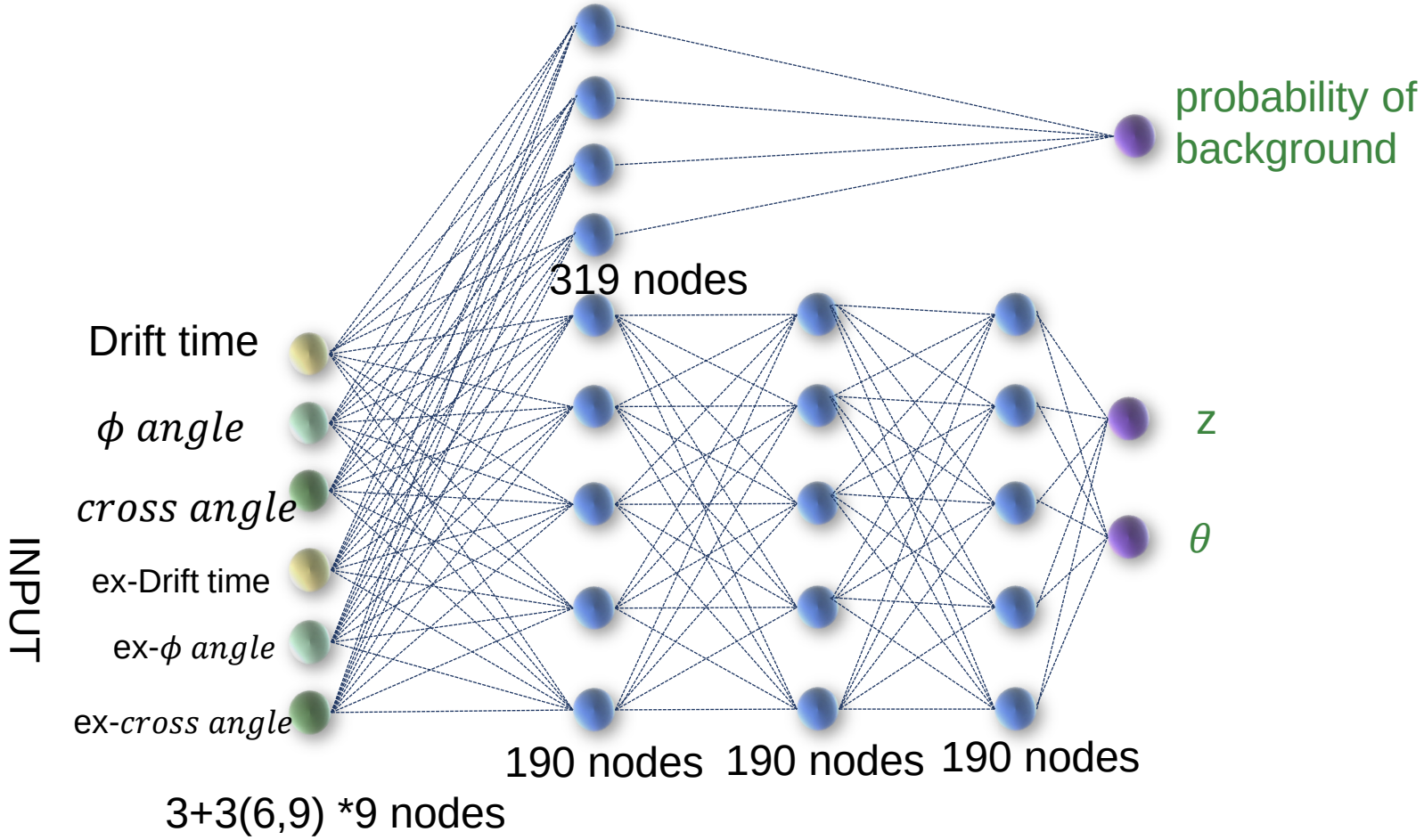
We do not need to increase hidden layer anymore with this.



With cut at 50 we have 92% efficiency & 88% reject rate

5

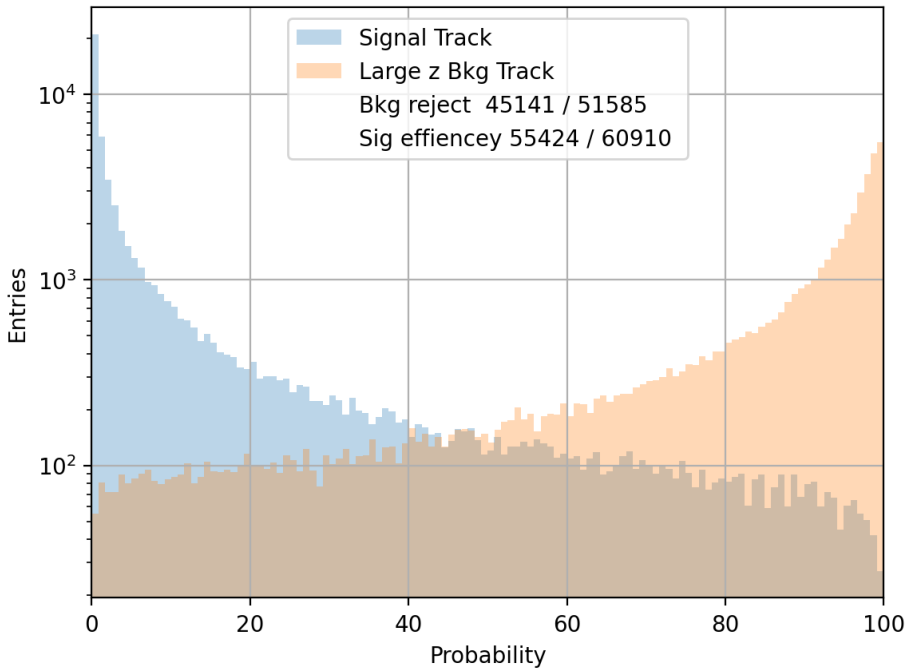
Add the probability output together with origin model



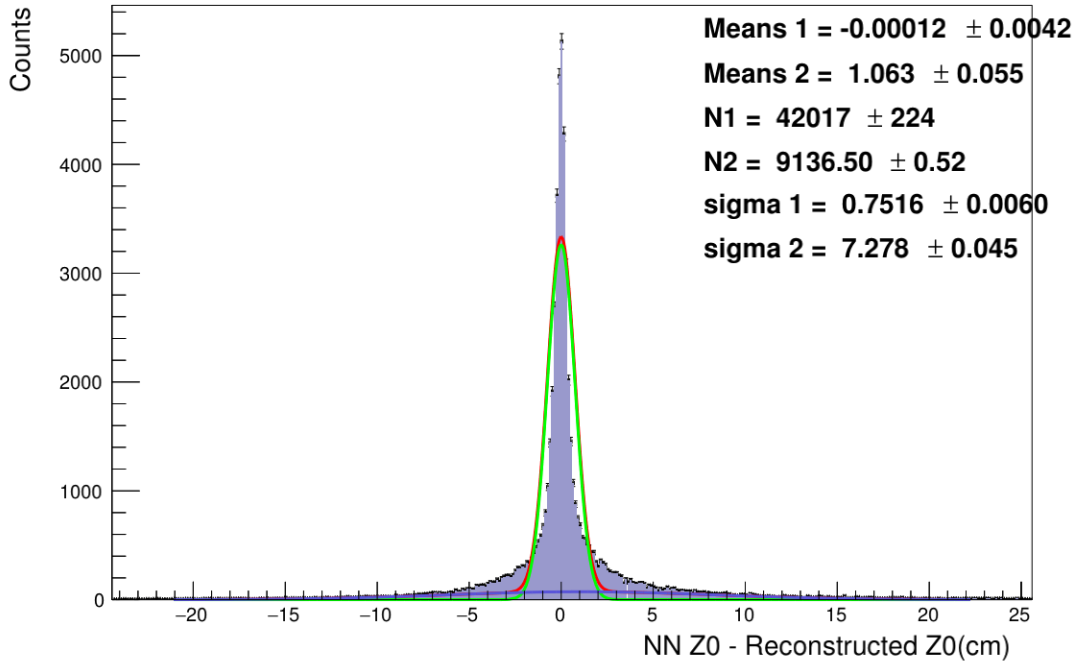
Two nets were independent actually(which I find got better result in training) but using same input and trained together
Using the best config for both (3 hidden layers with 190 nodes for z and θ; 1 hidden layer and 319 nodes for P

6

Result

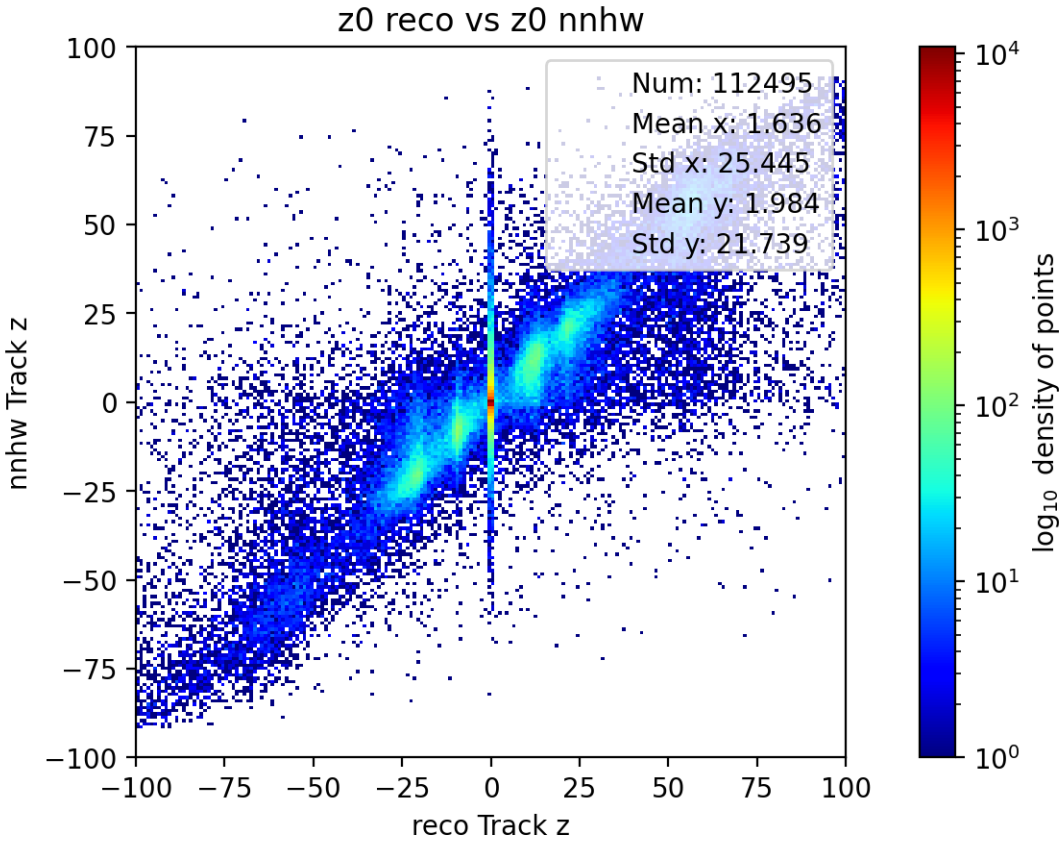


Probability got same result as the only P output network (as expected)

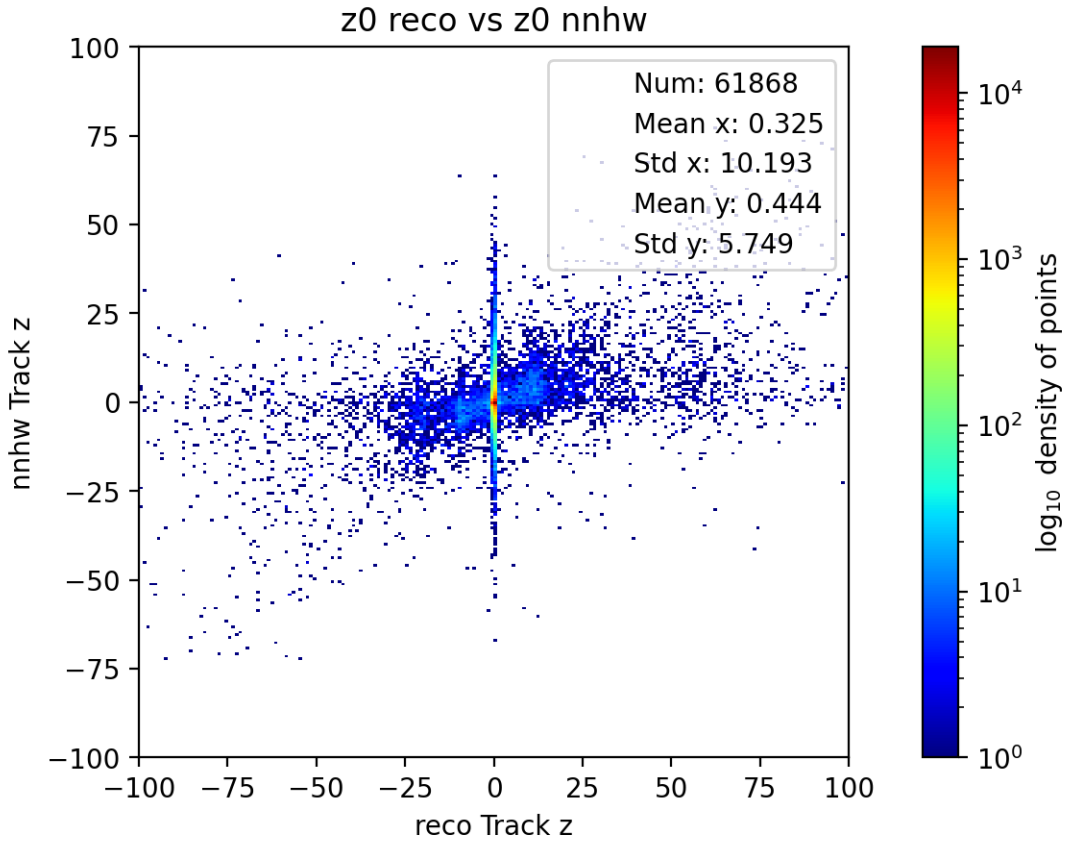


IP resolution same as before also

Result



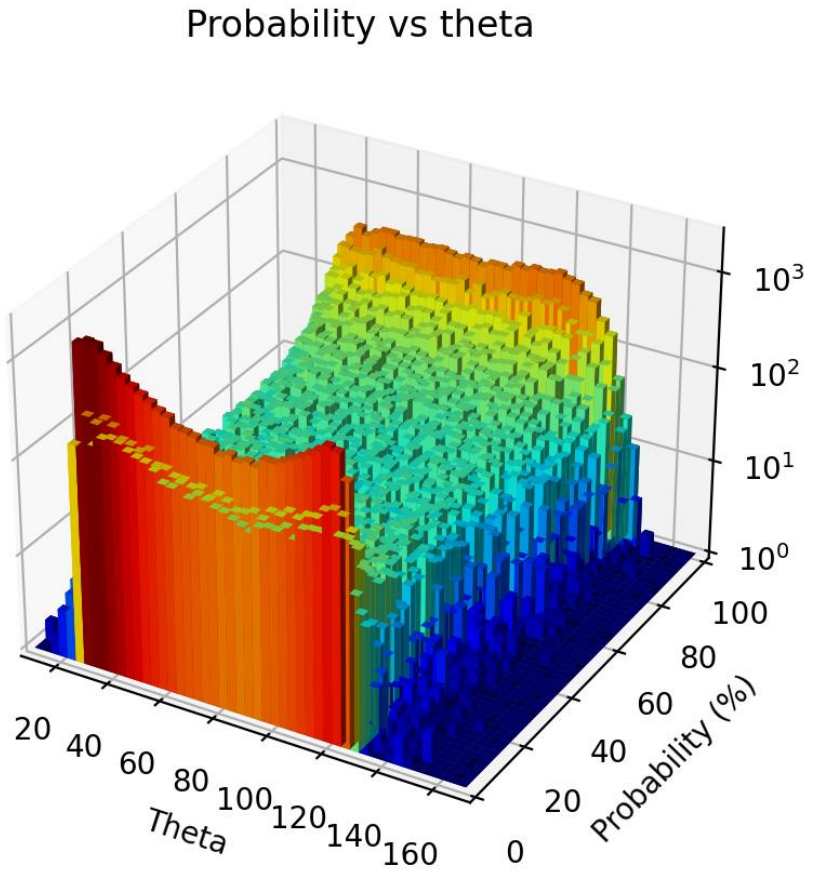
Before P cut



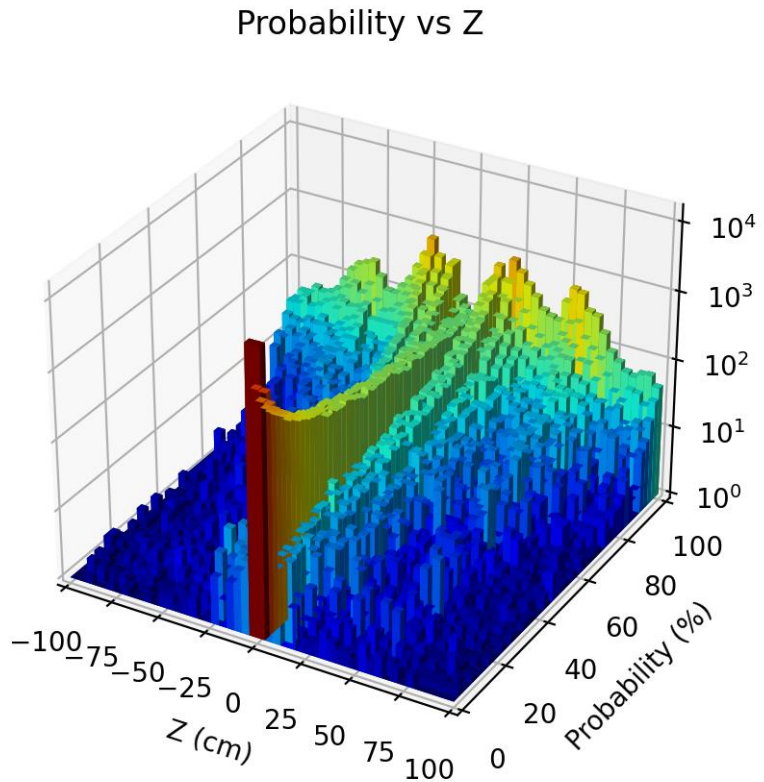
After P cut at 50

IP tracks were well kept even with large z in NN while rejected most background

Result

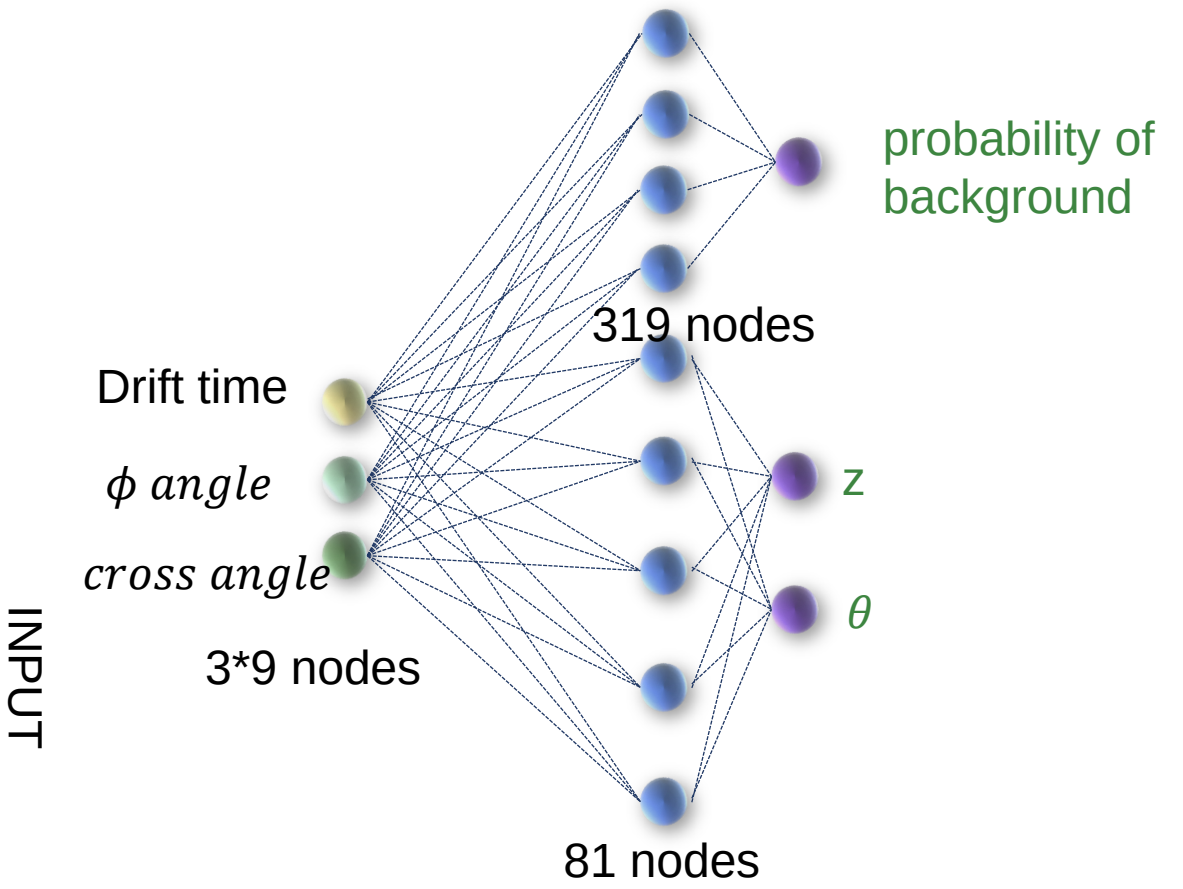


P versu theta
at P ==1 , theta has certain distribution while at P ==0 it is flat



P versu z
As expected, at P==1, z is almost 0 and with increasing P z enlarged also

Baseline model with P output

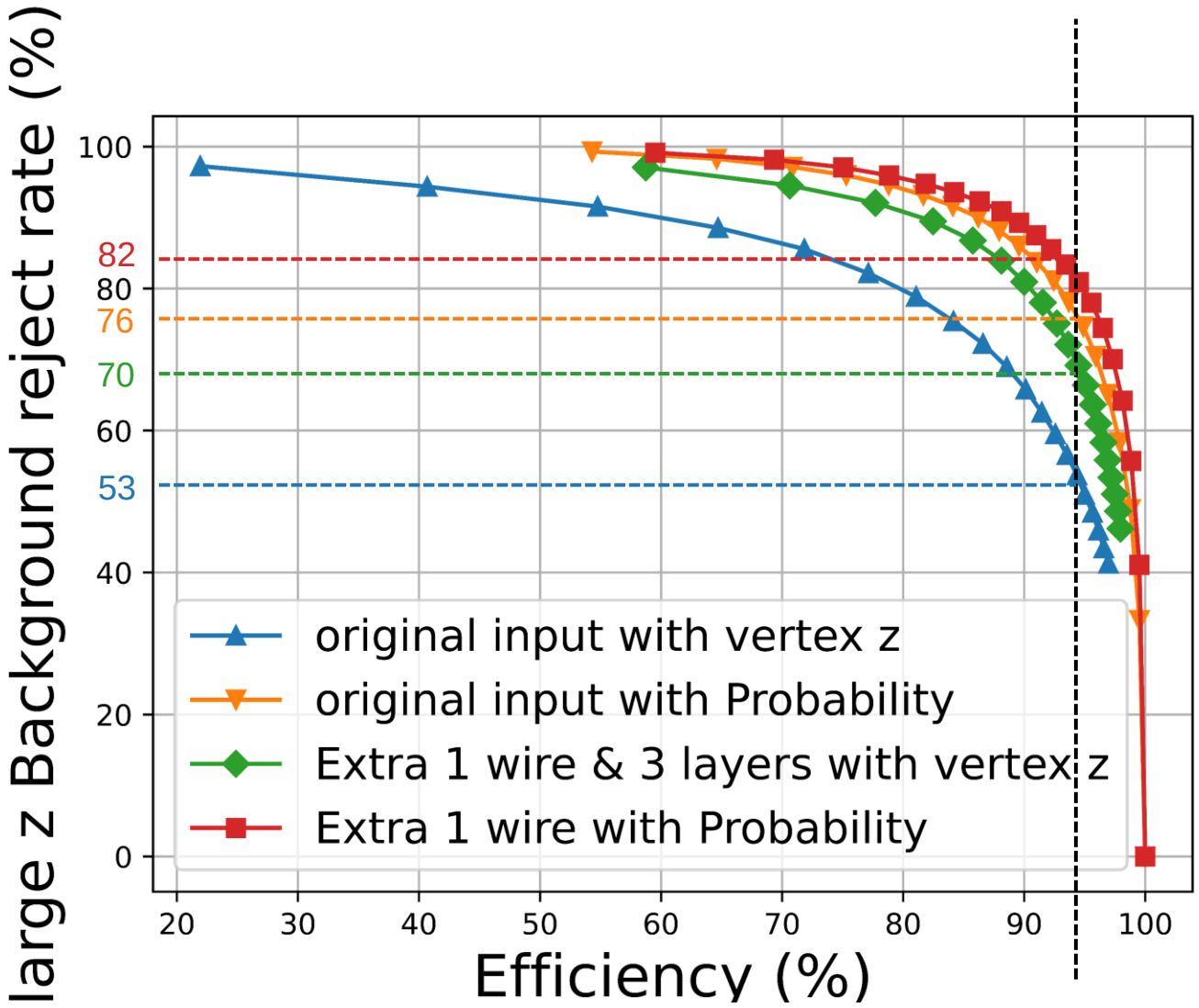


Also try to add P output with current input parameter (no extra wires here)

ADC cut is still applied since it is at data generated level

10

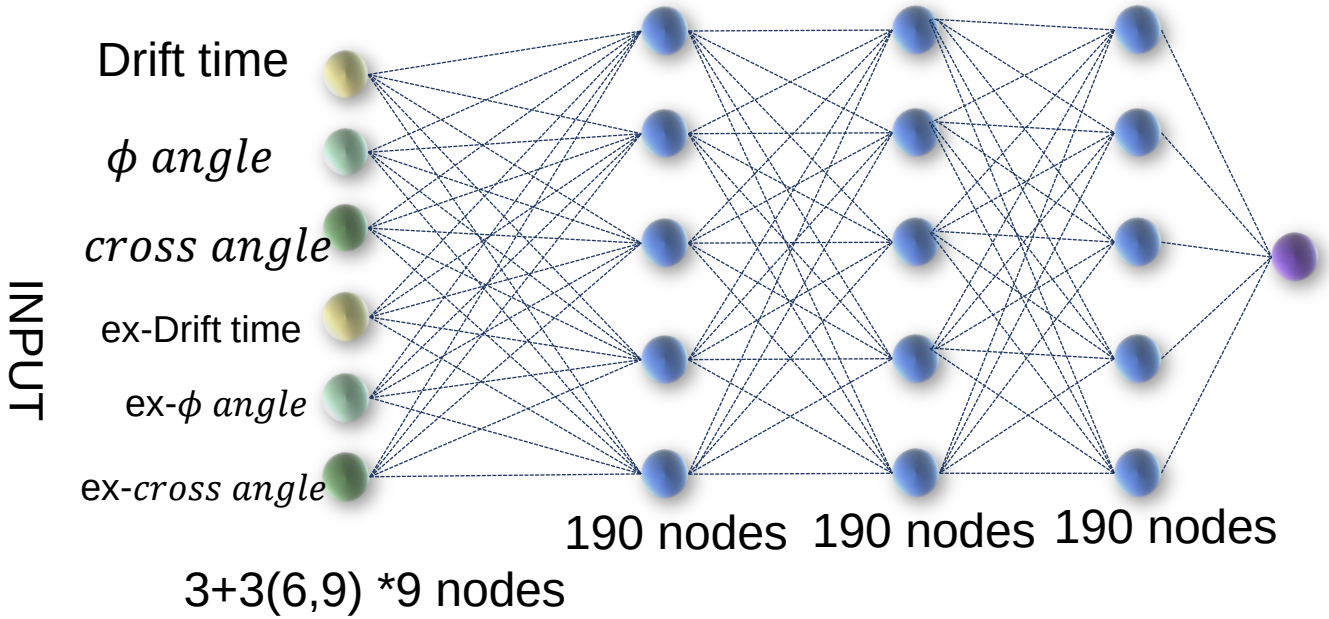
Comparison between different model

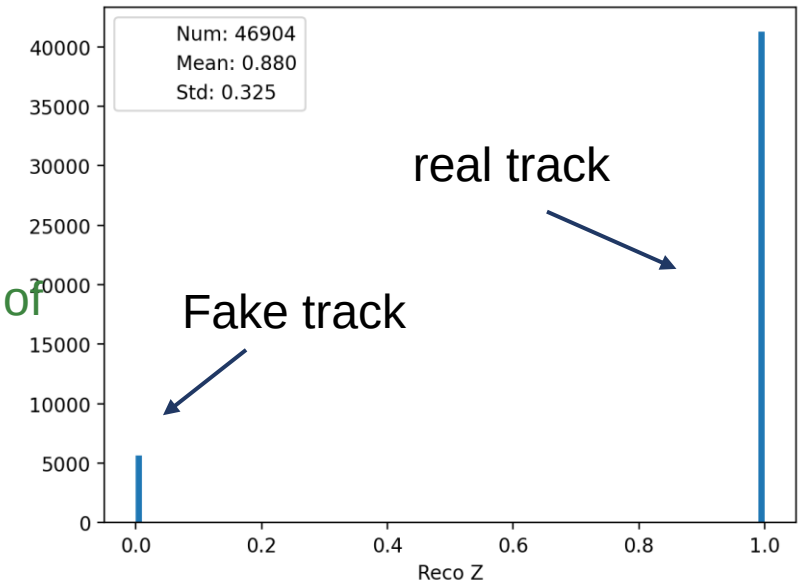


Black line:
Efficiency When we apply z cut 15 cm for origin NN

Even with origin input we could see large improvement in background reject rate
With extra input it could be improved further (For 1781 case 53% → 82%)

Try similar method for fake track



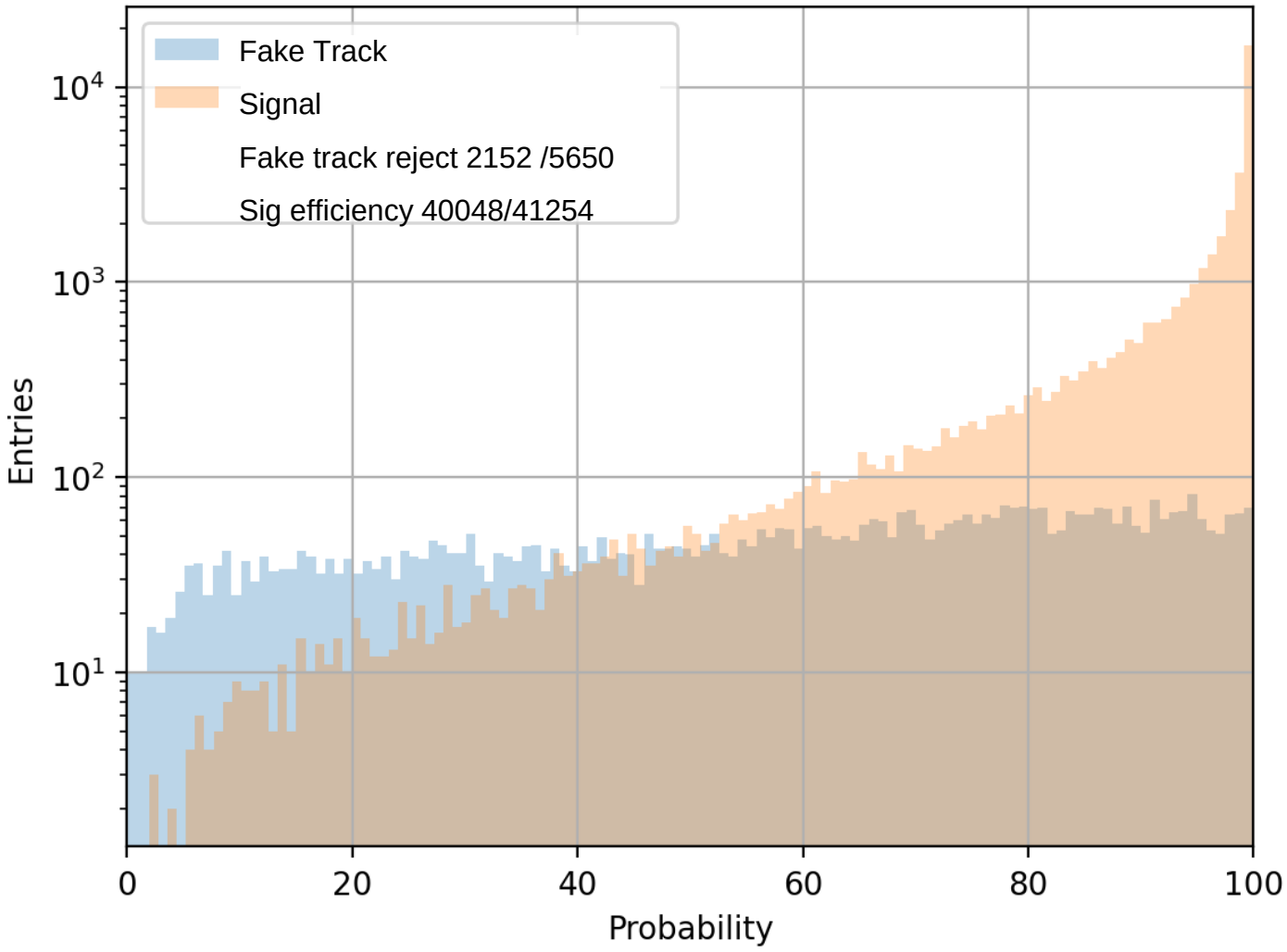


Target distribution

Only Train & validation with partial dataset from 1756 -1780 (~ 200k training data and 50k validation

NN Track related to RecoTracks will be marked as real track, others as Fake Track (~10% fake track same as Christian's result

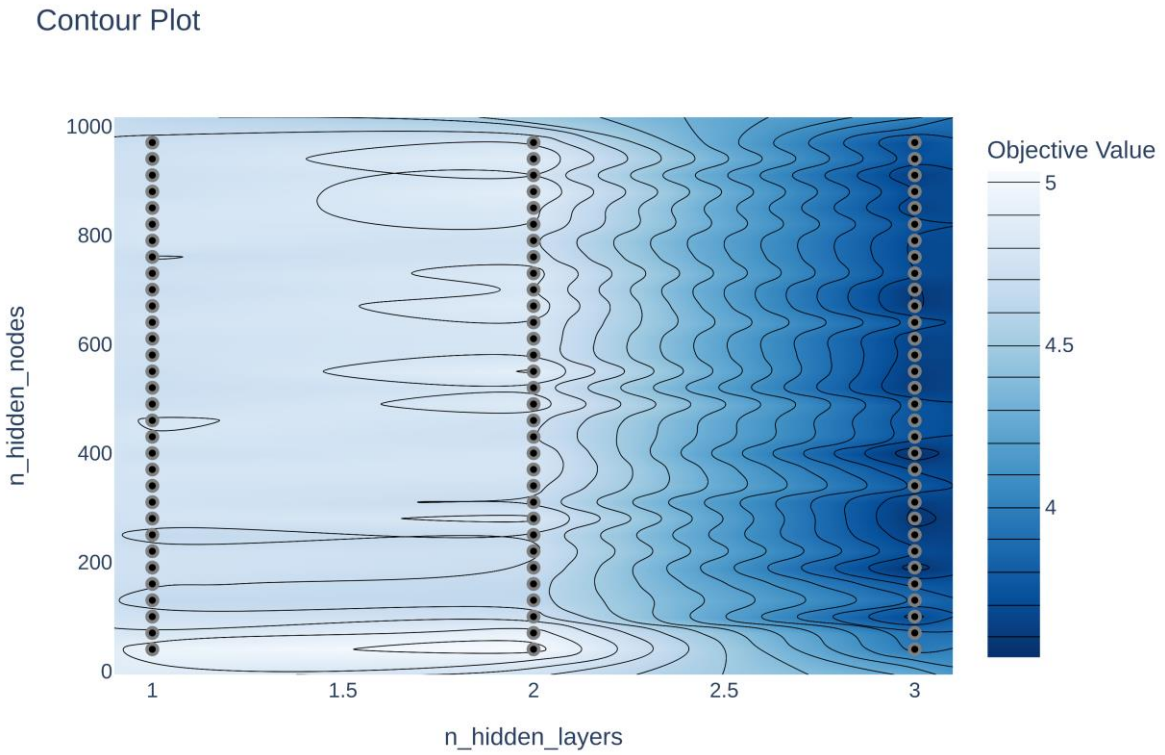
Try similar method for fake track



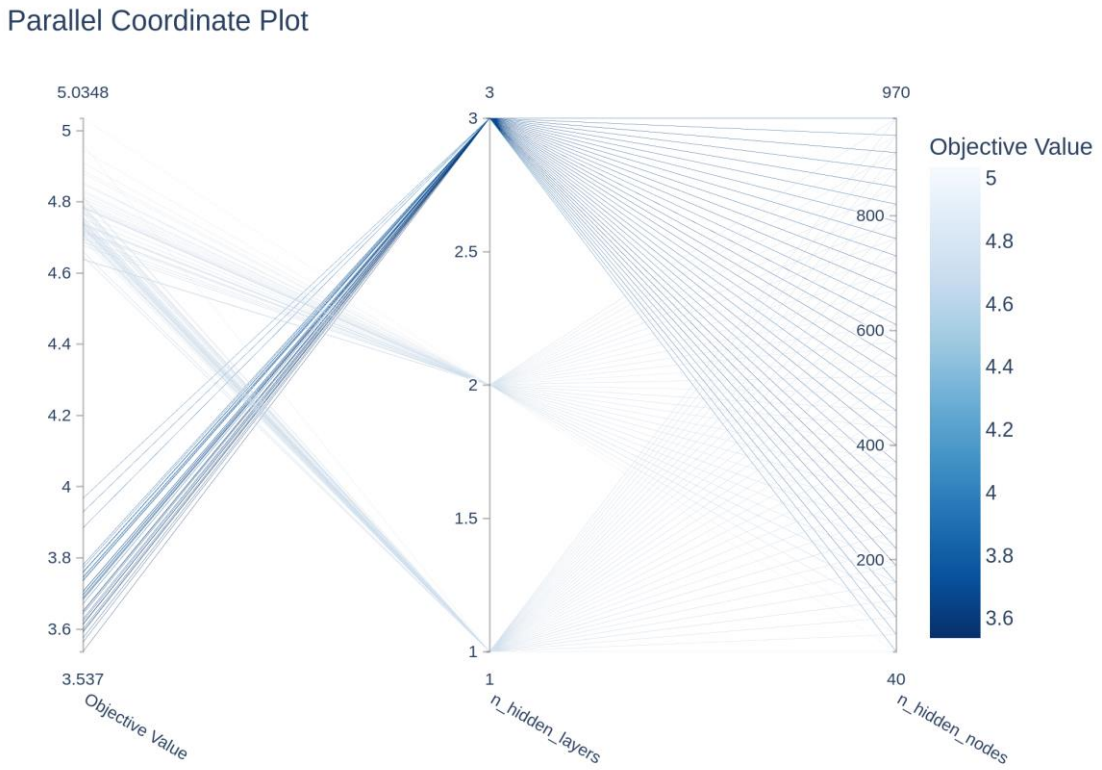
(in validation set) with efficiency ~ 97%, it could reduce ~ 38% fake track

If fake track rate continue increase, I will try to improve this NN.

A mistake fix and a reply



Only Black dots in this plot are a certain trials, other is interpolated from the exist trails… Origin method will focus on good performance area and not precise at 1 hidden layer case.
Perform a grid search see detail in 1 or 2 hidden layer(s).



Hidden nodes can never replace hidden layers in vertex z case.

Summary & Plan

Summary

a) Directly use NN to predict Signal/Background looks more powerful than fit z and set cut on z (Do we have a better ways to tag a track as signal or background instead of |z| <1?)

b) Similar model could also used for fake track reject

c) For z resolution improvement, more hidden layers are necessary

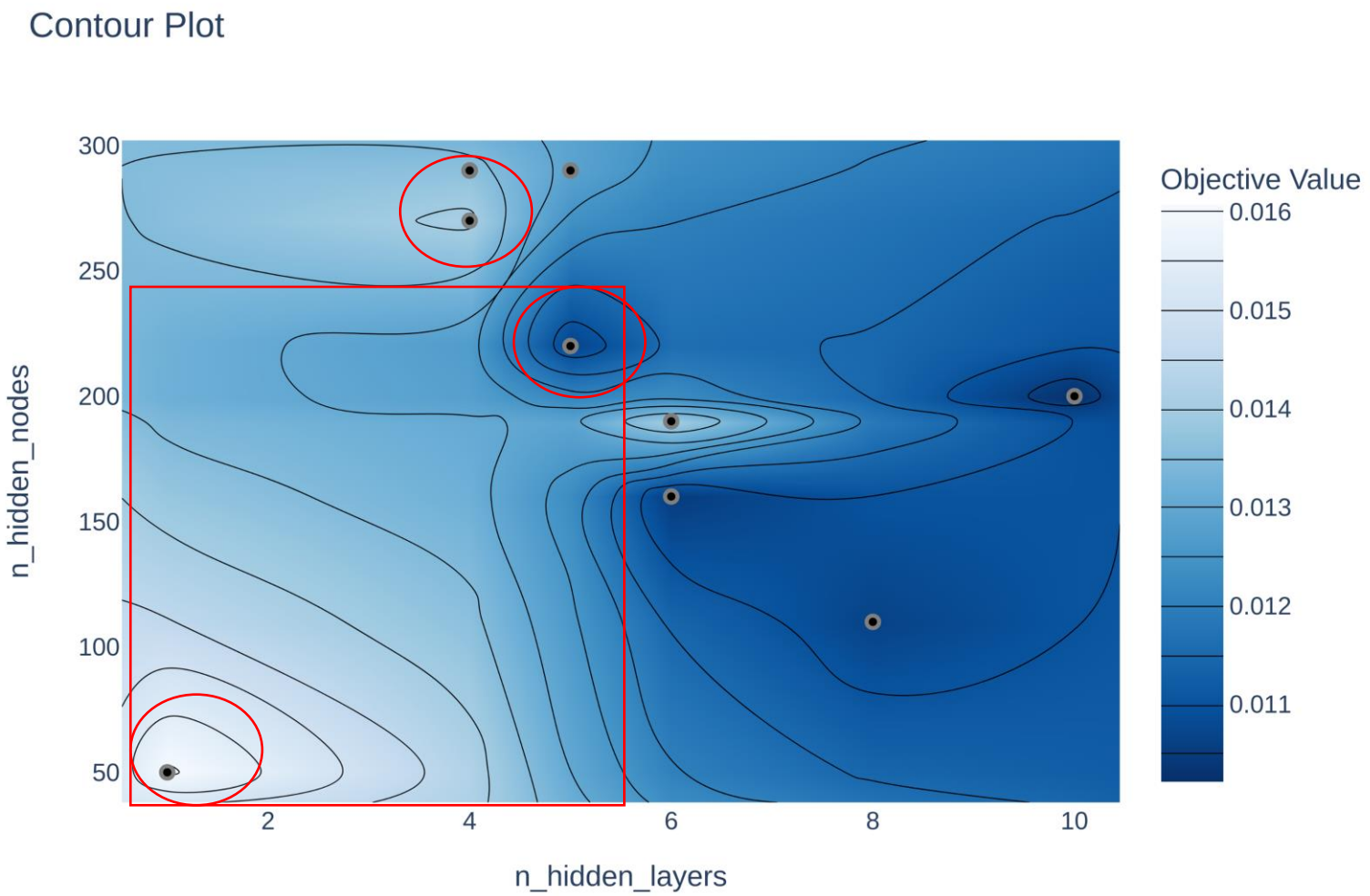
END

Thanks for your listening and attention!

END

Back UP

A mistake fix and a reply



For this case, with prune, only two or three trials are used to generate contour line in between….