1 Dimensional CNN

2020.03.11

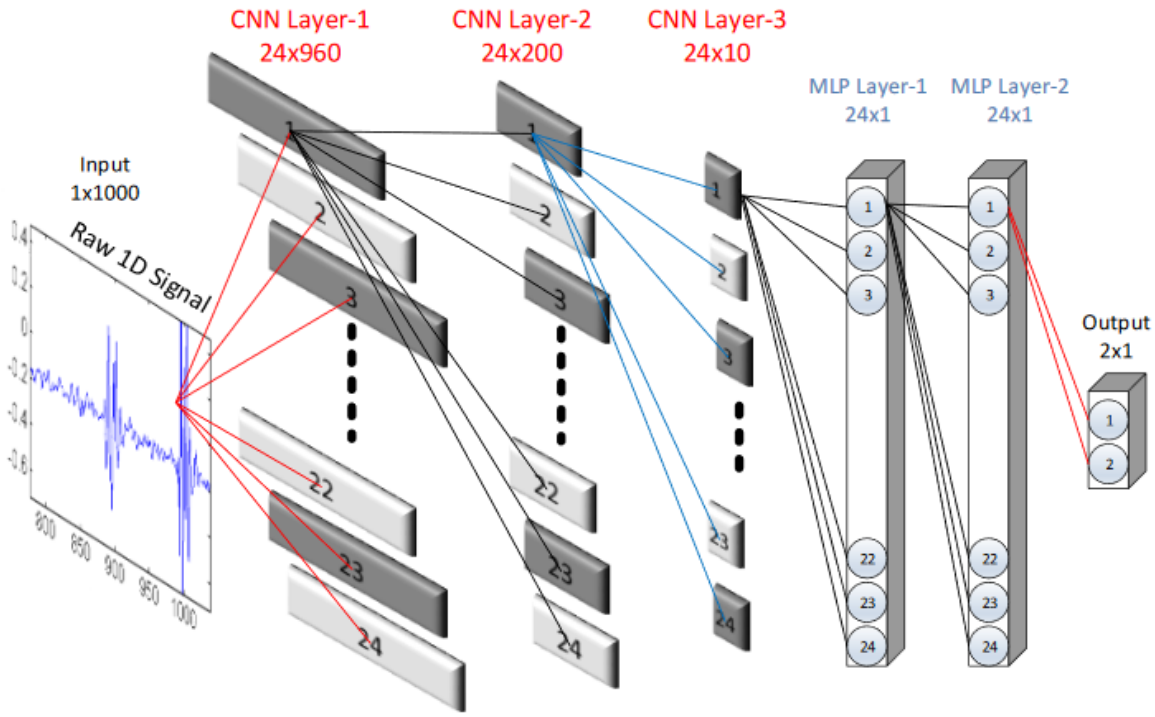
Seungmok Lee

First CNN for Cosine

- I implemented and ran CNN for Cosine data!
  - Used data having energy 6~10 keV, from crystal 3.
  - Events from calibration run (runnum 1765, 1601) having coincidence were tagged as ‘Signal’.
    - 2540 events
  - Events from physics run (runnum 1858, 1859) without coincidence were tagged as ‘Background’.
    - 63221 events
- I designed my networks imitating two famous networks.

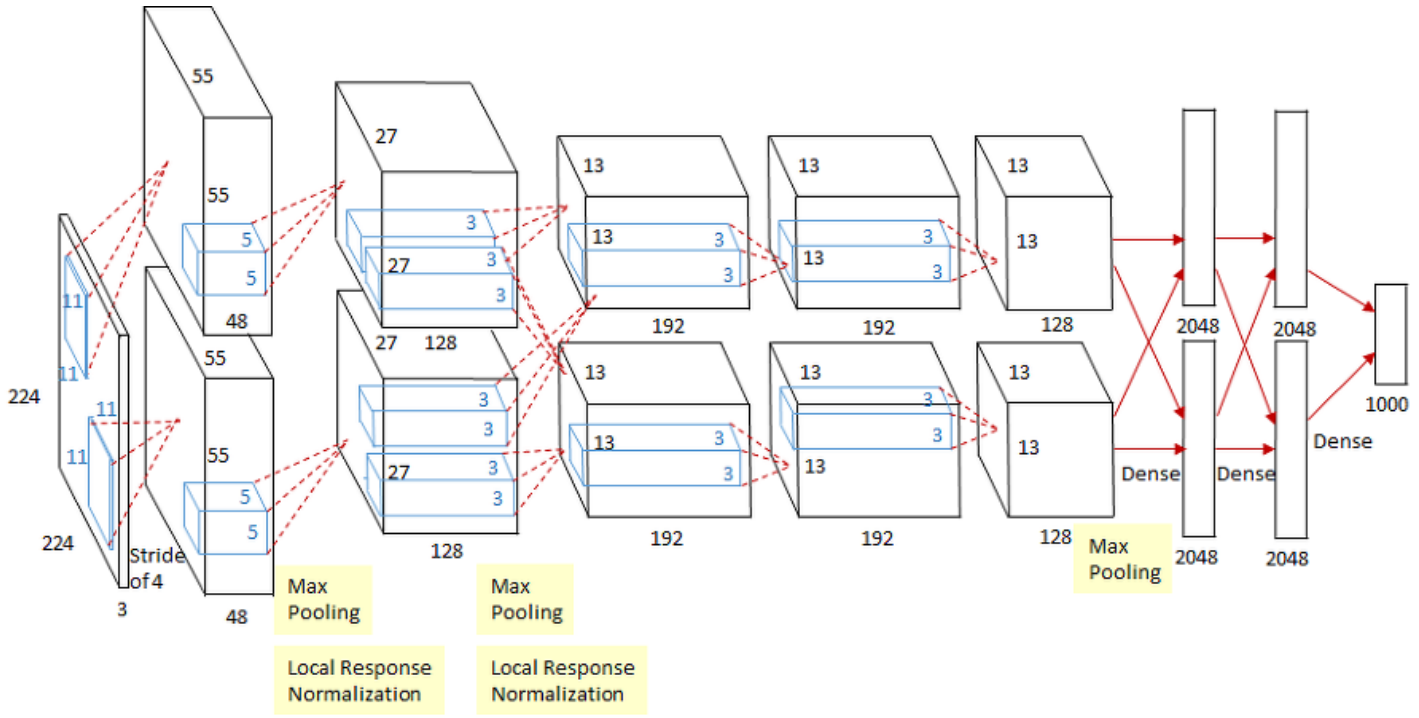
2 Famous CNN Networks

- Kiranyaz Network (2017, 1D Electrocardiogram Classification)
  - Kernel size = 41
  - Pooling size = 4



2 Famous CNN Networks

- AlexNet (2012, 2D Image Classification)
  - Kernel size getting narrower.
  - Stride applied for wide filter.
  - So many filters.

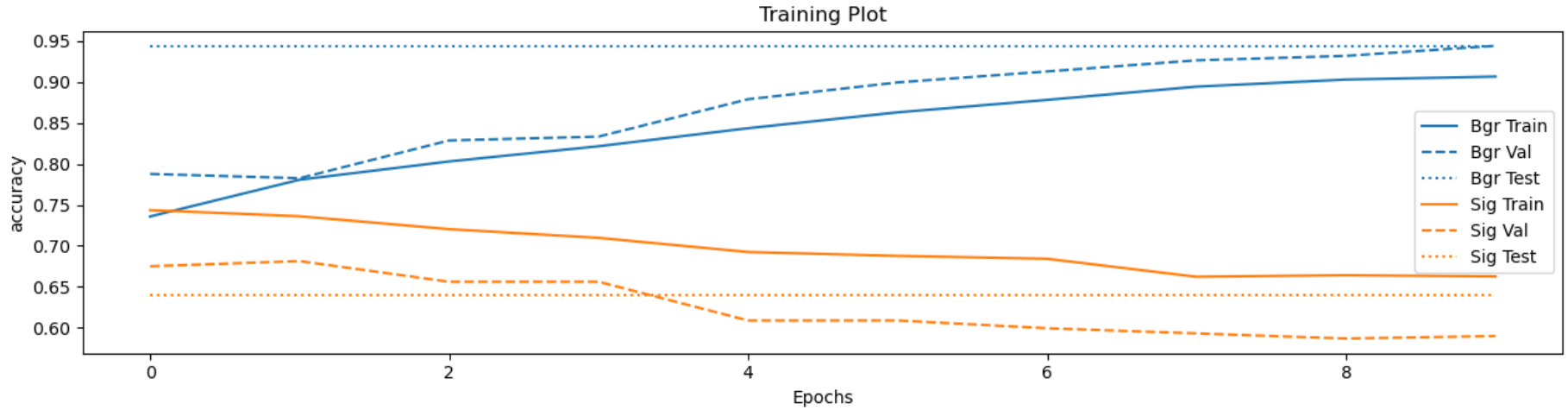


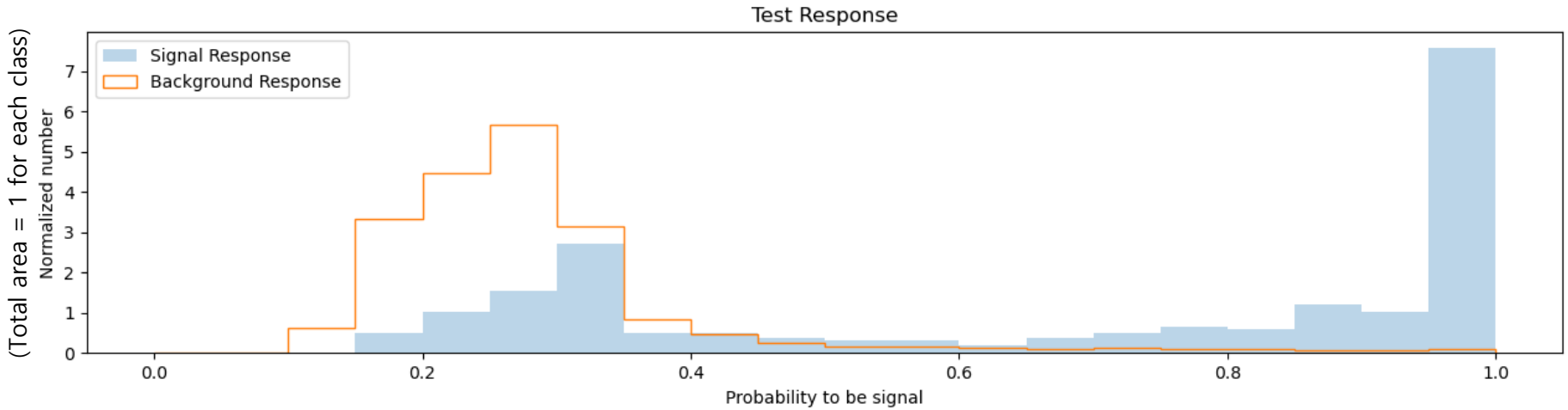
First CNN for Cosine

Input shape	Layer	Output shape
(4080, 2)	Conv1D(filters=64, kernel_size=81)	(4000, 24)
(4000, 24)	MaxPool1D(pool_size=4)	(1000, 24)
(1000, 24)	Conv1D(filters=24, kernel_size=41)	(960, 24)
(960, 24)	MaxPool1D(pool_size=4)	(240, 24)
(240, 24)	Conv1D(filters=24, kernel_size=41)	(200, 24)
(200, 24)	MaxPool1D(pool_size=4)	(50, 24)
(50, 24)	Conv1D(filters=24, kernel_size=41)	(10, 24)
(10, 24)	Flatten()	240
240	Dense(24)	24
24	Dense(24)	24
24	Dense(2)	2

Motivated from Kiranyaz ECG Network.
Batch normalization, Dropout, ReLU activation applied.

Not Satisfying Result





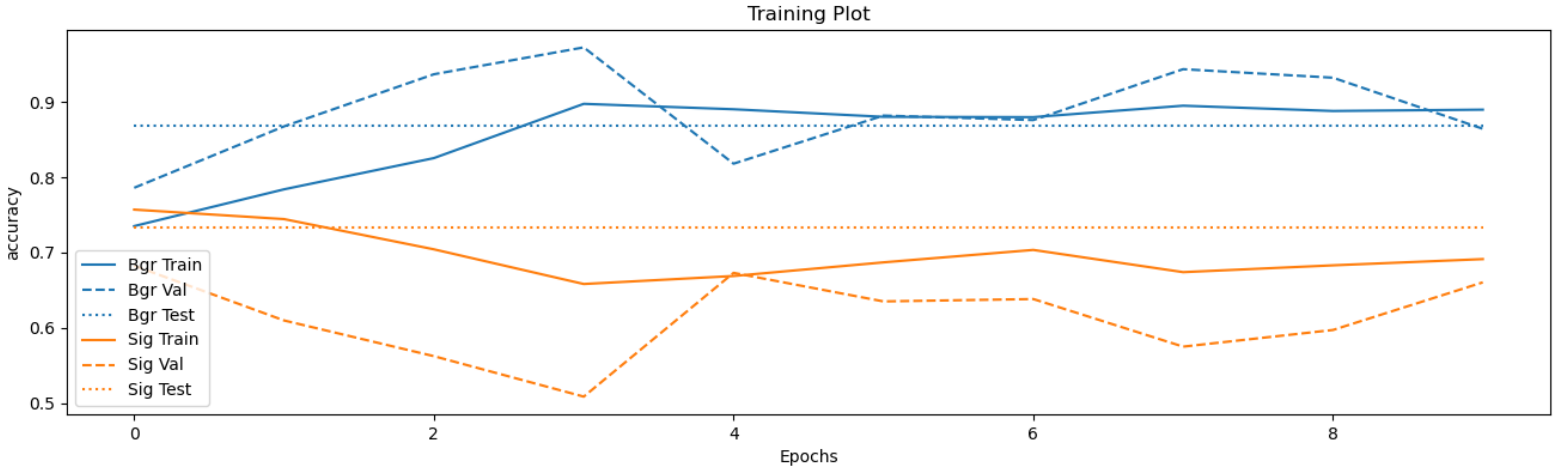
Note! Two graphs have different color configuration for sig / bg.

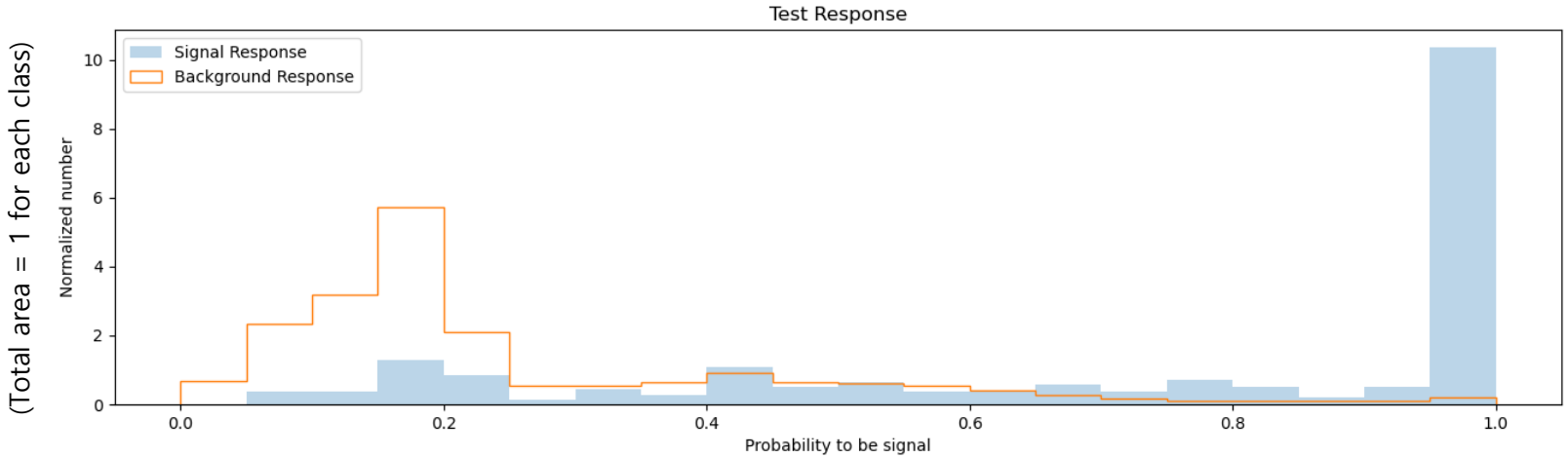
Second CNN for Cosine

Input shape	Layer	Output shape
(4080, 2)	Conv1D(filters=96, kernel_size=39, strides=3)	(1348, 96)
(1348, 96)	MaxPool1D(pool_size=4)	(337, 96)
(337, 96)	Conv1D(filters=256, kernel_size=23)	(315, 256)
(315, 256)	MaxPool1D(pool_size=3)	(105, 256)
(105, 256)	Conv1D(filters=384, kernel_size=9)	(97, 384)
(97, 384)	Conv1D(filters=384, kernel_size=9)	(89, 384)
(89, 384)	Conv1D(filters=256, kernel_size=9)	(81, 256)
(81, 256)	MaxPool1D(pool_size=3, strides=2)	(40, 256)
(40, 256)	Flatten()	10240
10240	Dense(512)	512
512	Dense(512)	512
512	Dense(512)	512
512	Dense(512)	512
512	Dense(2)	2

Motivated from AlexNet. Batch normalization, ReLU activation applied.

Still Not Satisfying



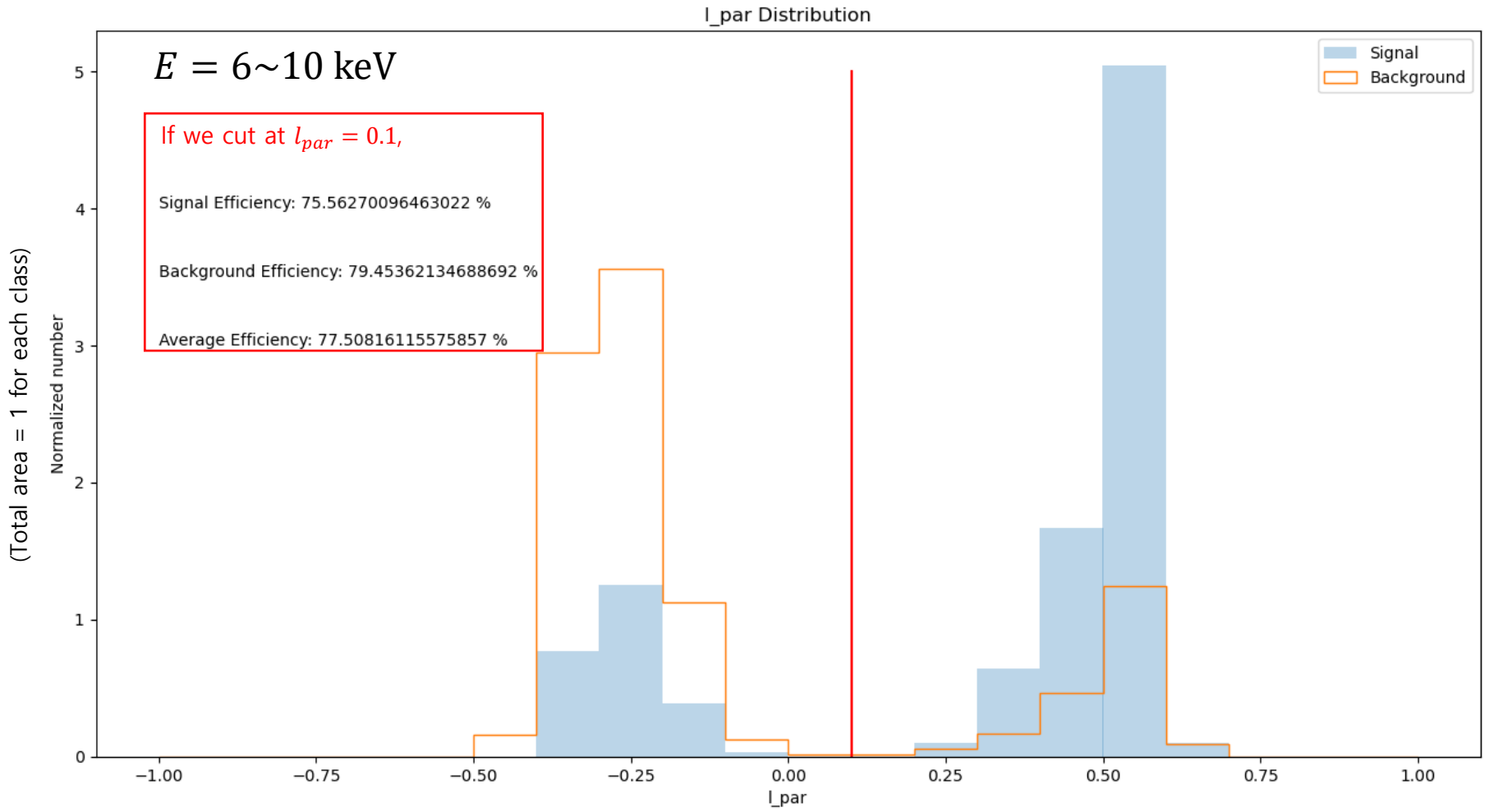


Note! Two graphs have different color configuration for sig / bg.

Data Impurity : Comparison With lpar

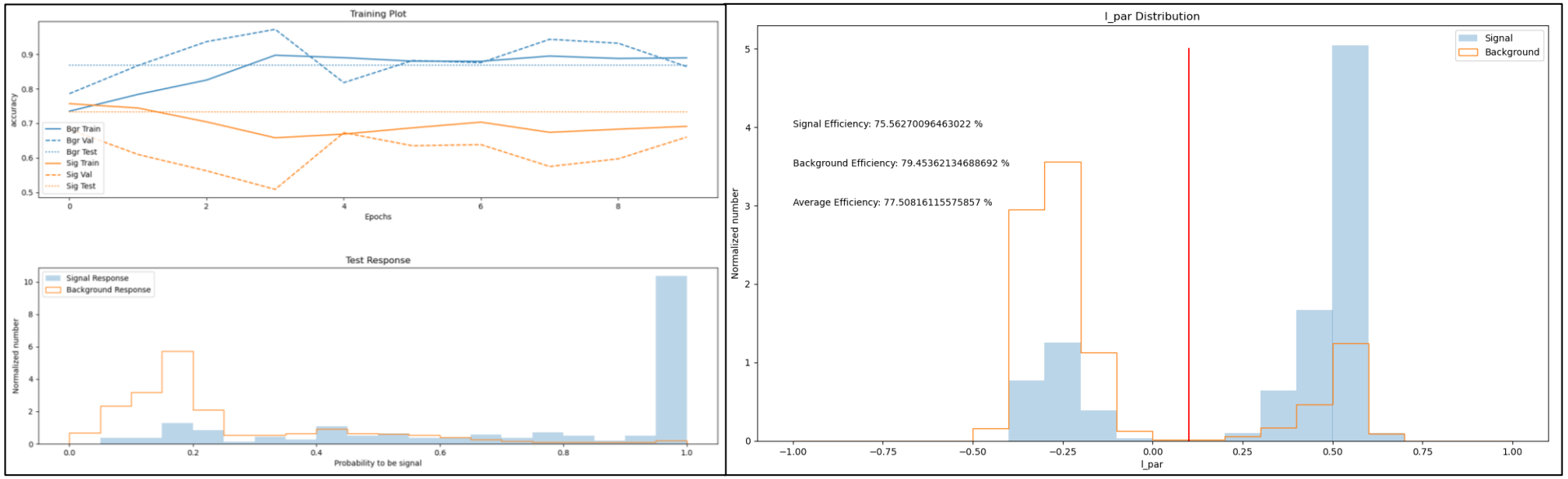
- It was awkward that CNN did not improve the performance.
- To check the impurity of my data, I observed the lpar distribution.

Data Impurity : Comparison With lpar



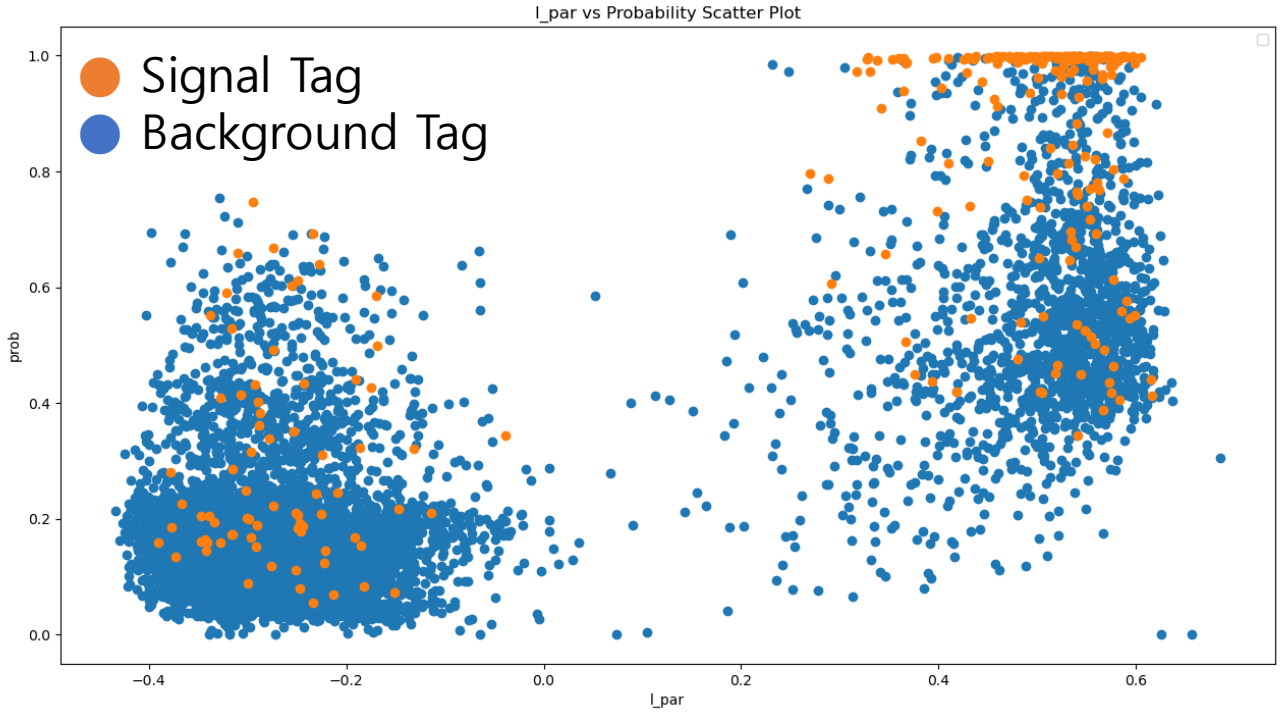
Data Impurity : Comparison With lpar

- Lpar was inconsistent with my tagging!
- My CNN have already reached lpar-level performance. Maybe this was the reason why I couldn't go beyond efficiency 75%



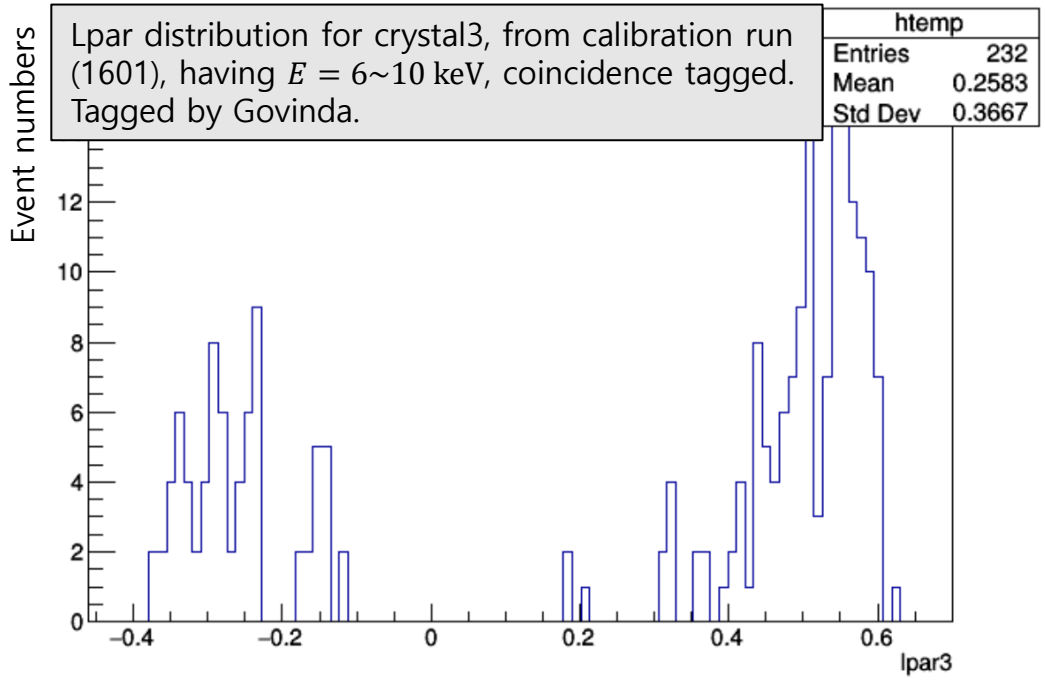
Data Impurity : Comparison With lpar

- The correlation between lpar and the probability (by my network) was weak but existed.
- The first thing I must do is data checking!



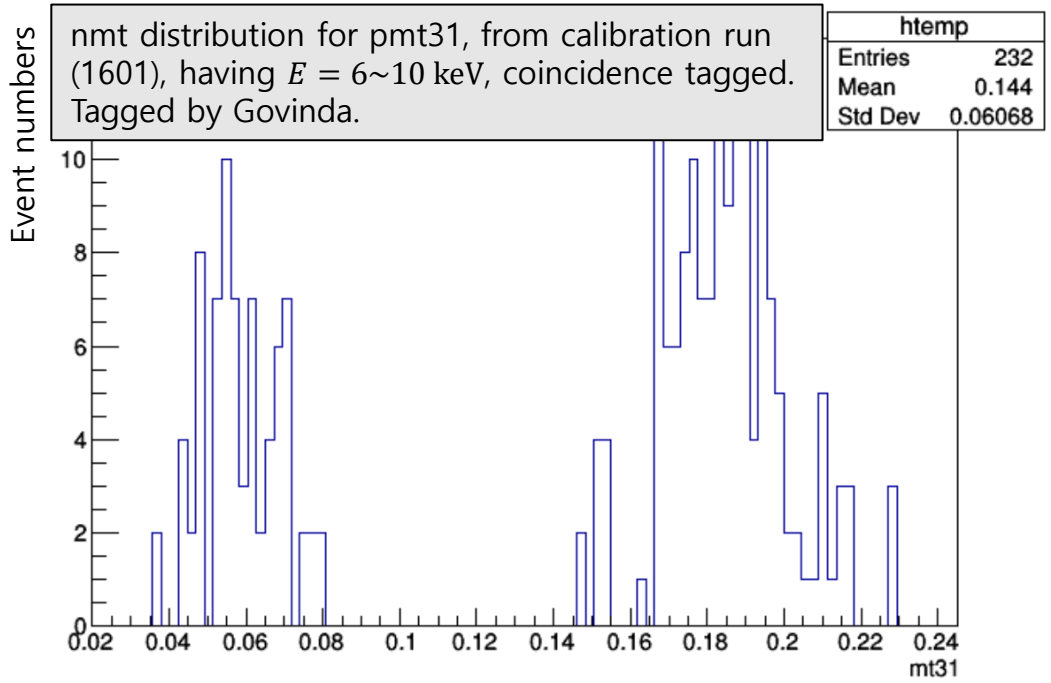
Is Coincidence Tagging Enough?

- I followed Govinda's tagging algorithm, but that seems not enough.
- Histogram below is data with coincidence tagged Govinda. It is also impure.



Is Coincidence Tagging Enough?

- nmt is also contaminated.
- Is there a better way to extract pure signal / background?



What is this?

- However, according to Govinda, lpar seems nice.
- What am I missing?