Local meeting

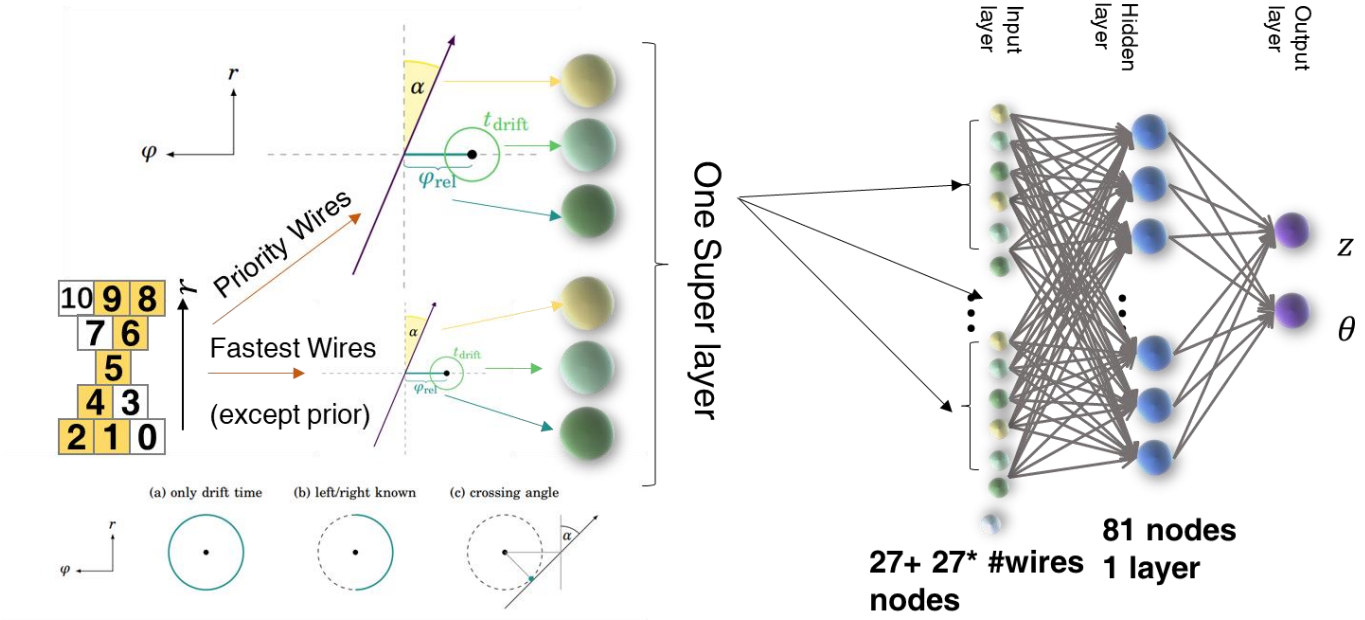
Yuxin Liu

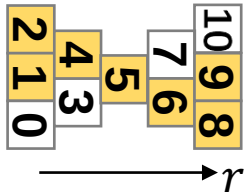
2023/01/31

OUTLINE

1. Extra location information for extra wires.

2. Optuna perfomance





Fastest Wires

(except prior)

?

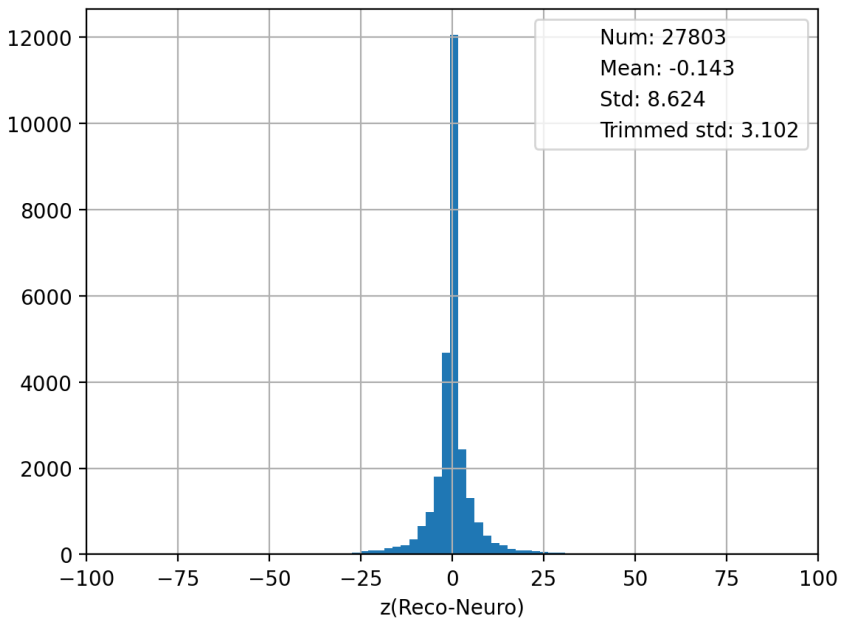
Location of Extra wire?

1. R_wire: precise, do have certain physics mean, can not distinguish (4,5) (7,6)(0,1,2)…

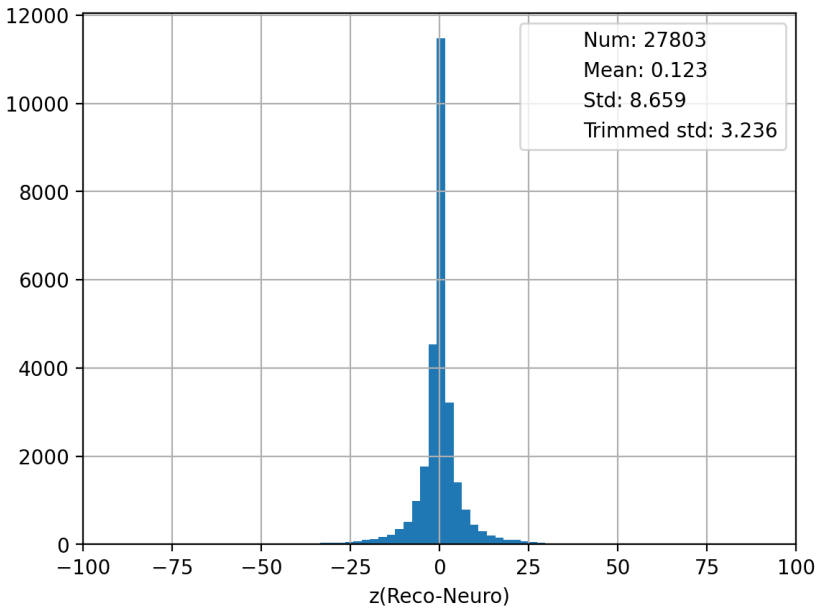
2. Related No. : **n/11**, no physics mean, large gradient .

Use 2 for test first.

Controll group



Using related location



Optuna

First searching hyperparams:

```
n_layers: 1 - 10
n_nodes : 10 - 300,step=10
lr : 1e-4 - 1e-2 (log search)
activate : relu or tanh/2
batch_size :[216,512,1024,2048,4096]
```

Optimization value : val loss from MSELoss

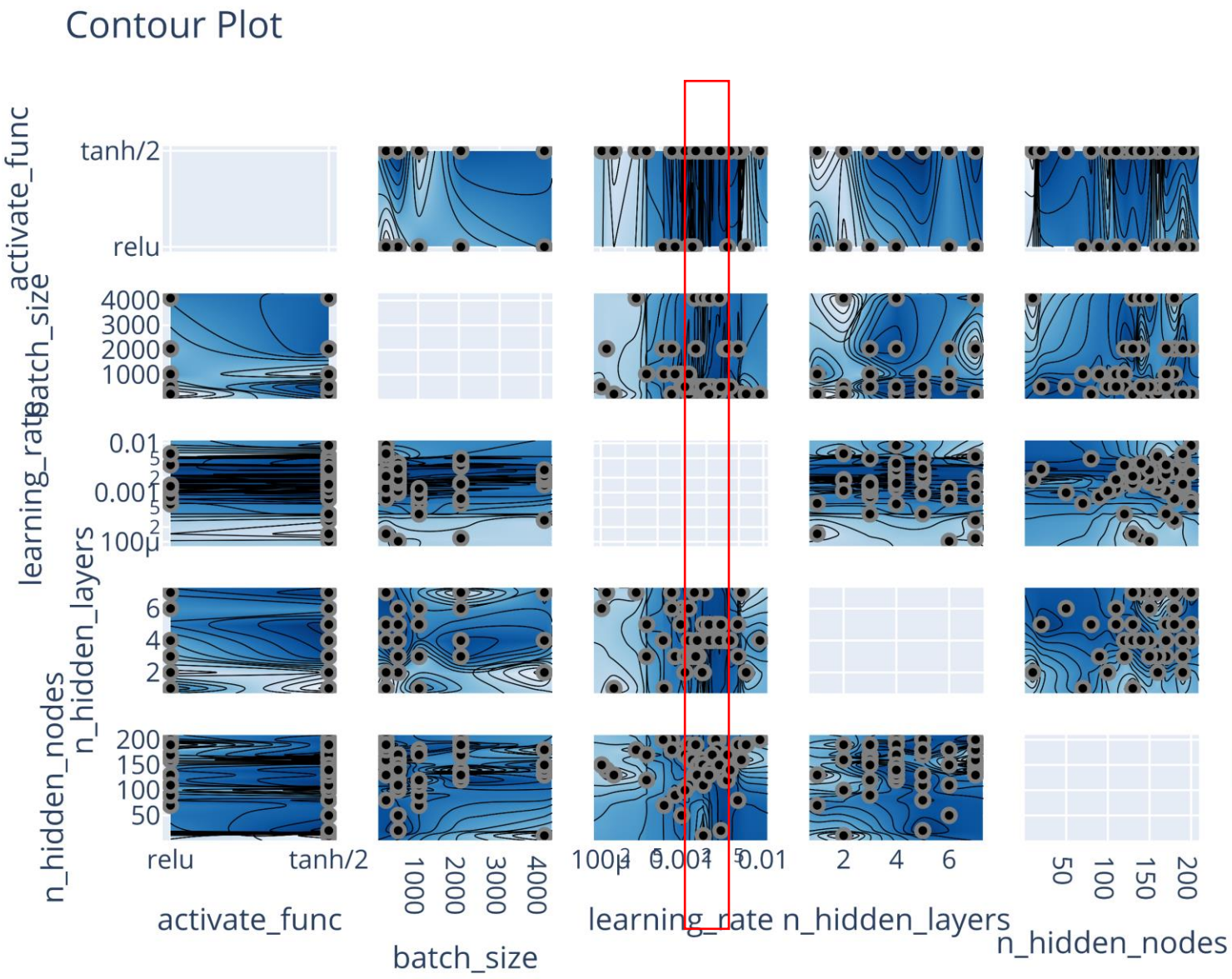
Dataset: 1/10 of total train set in exp26run1756 -run1780
Val Dataset: full exp26run96

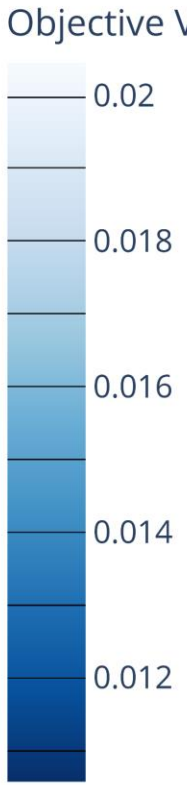
Epoch: 100 (almost enough for generally 3 hidden layer case)

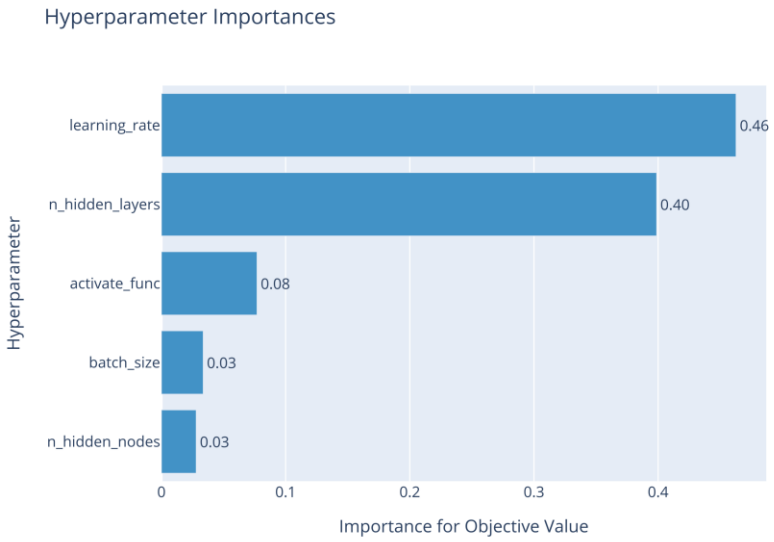
Model: Extrawire1 HD 3 w/o extra L/R

Trials (# NN trained) 50 → Actually should be larger but with kekcc, ~60 will used up memory, try to solve this problem…

Optuna







Learning rate is extremely important for training with limited epoch

Choosing the best value in 50 trials: `0.0021476556109654 46`

Optuna

Next searching hyperparams:

```
n_layers: 1 - 10
n_nodes : 10 - 300,step=10
lr : 0.00214765561096544 6
activate : relu or tanh/2
batch_size :[216,512,1024,2048,4096]
```

Optimization value : val loss from MSELoss

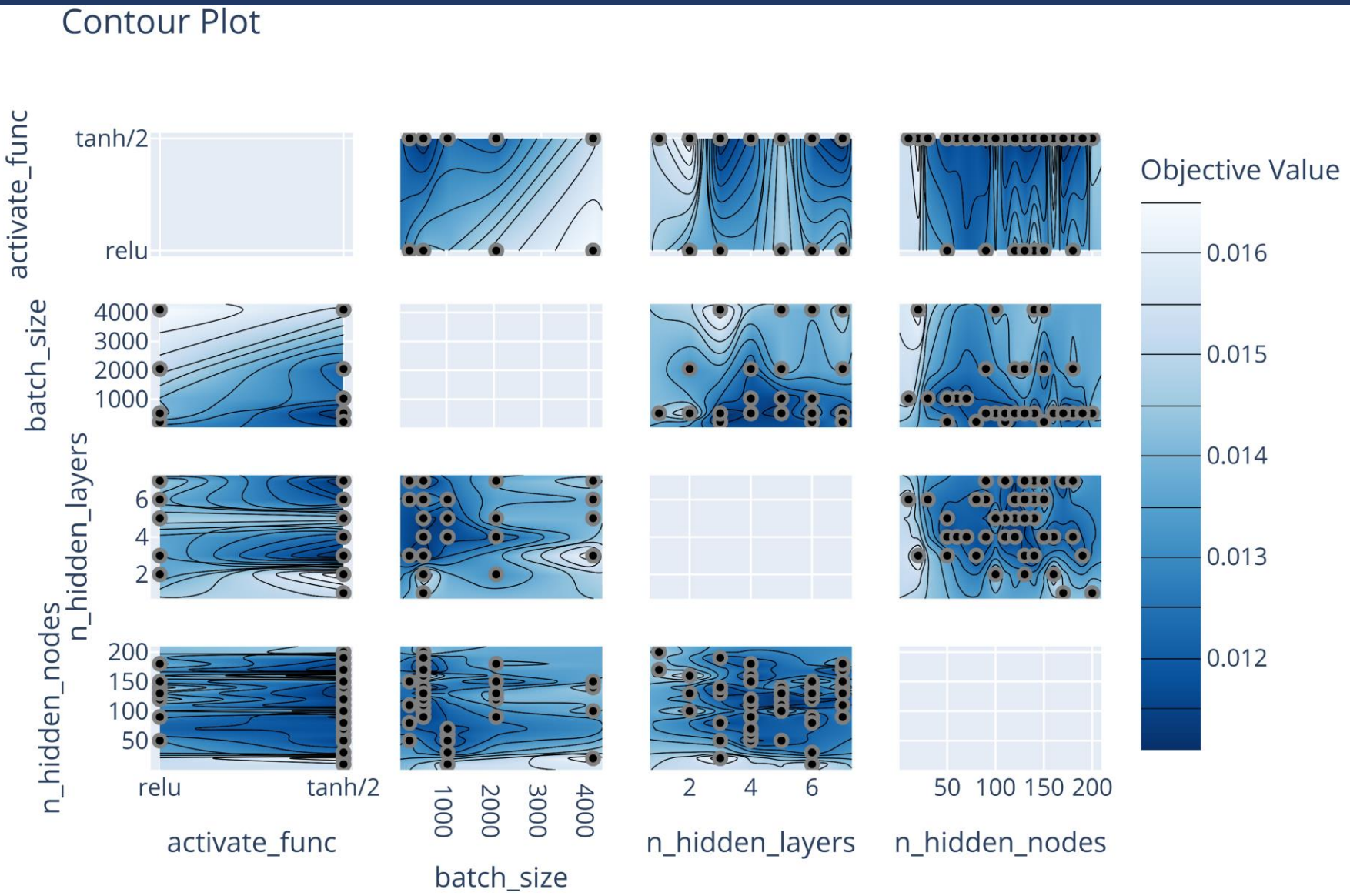
Dataset: 1/10 of total train set in exp26run1756 -run1780
Val Dataset: full exp26run96

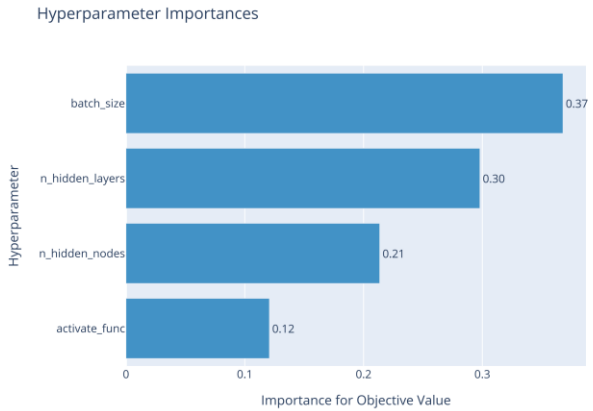
Epoch: 100 (almost enough for generally 3 hidden layer case)

Model: Extrawire1 HD 3 w/o extra L/R

Trials (# NN trained) 50 → Actually should be larger but with kekcc, ~60 will used up memory, try to solve this problem…

Optuna





Batch size first this time

Fixed as 512

Optuna

Third searching hyperparams:

```
n_layers: 1 - 10
n_nodes : 10 - 300,step=10
lr : 0.00214765561096544 6
activate : relu or tanh/2
batch_size: 512
```

Optimization value : val loss from MSELoss

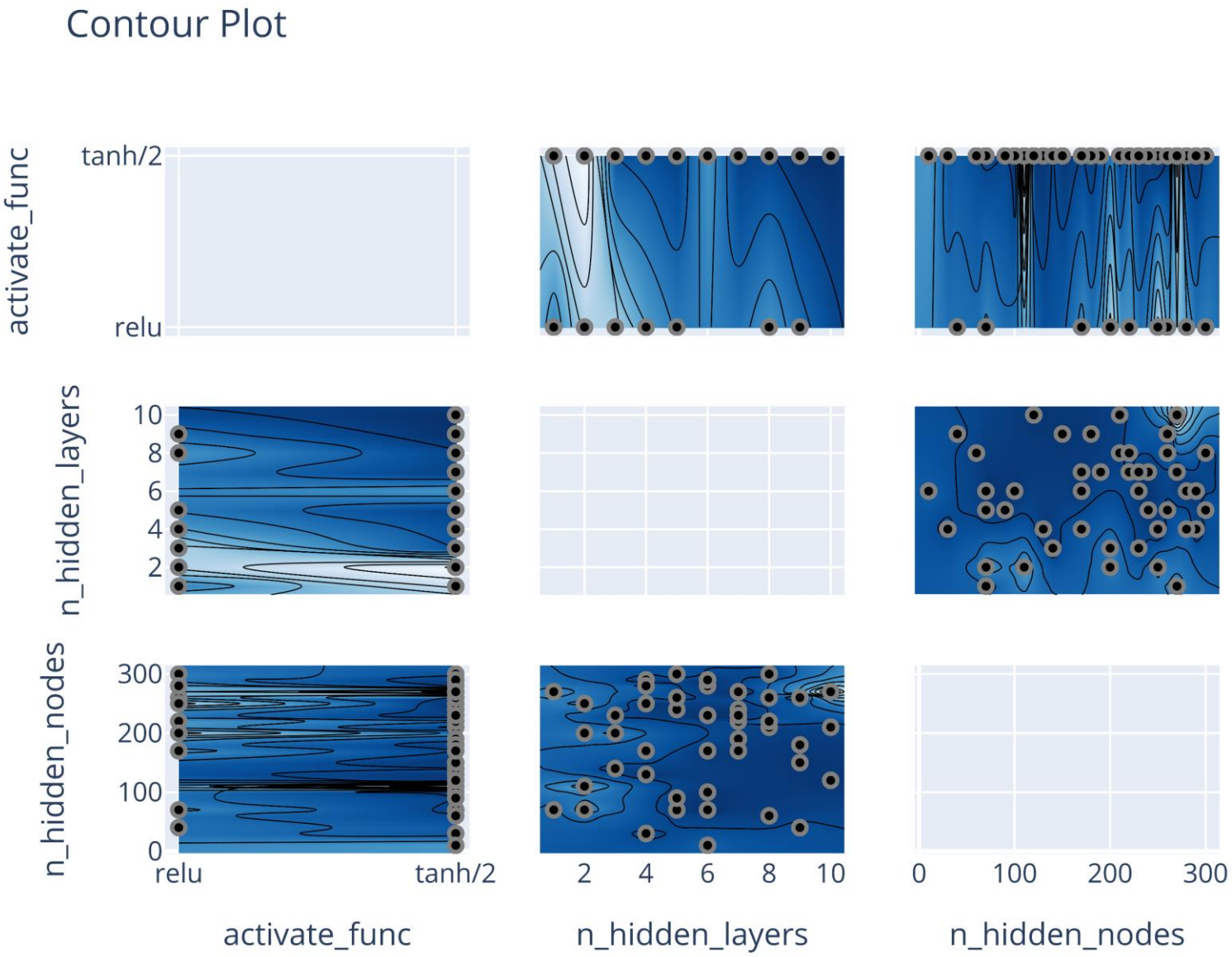
Dataset: 1/10 of total train set in exp26run1756 -run1780
Val Dataset: full exp26run96

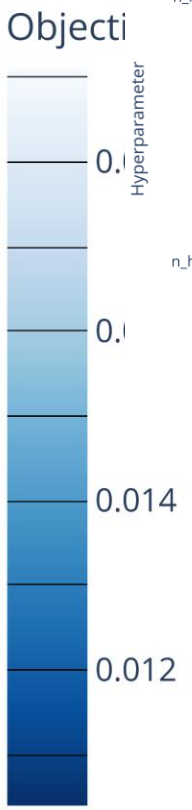
Epoch: 100 (almost enough for generally 3 hidden layer case)

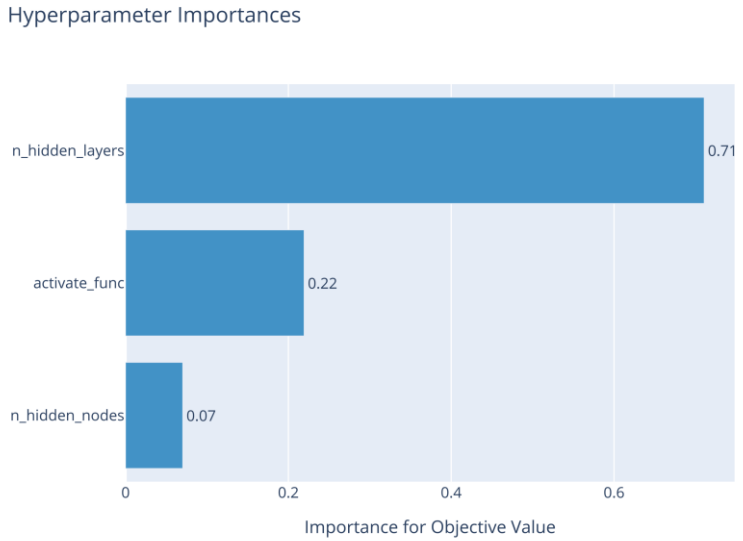
Model: Extrawire1 HD 3 w/o extra L/R

Trials (# NN trained) 50 → Actually should be larger but with kekcc, ~60 will used up memory, try to solve this problem…

Optuna







Hidden layers > activate function > hidden nodes

Summary & Plan

Summary

a) Extra related location do not work…

b) Optuna result meet previous result, will try deeply optimization with more trial

（From Tree-Structured Parzen Estimator, it said at least 200 for one hyperparams…）

END

Thanks for your listening and attention!

BACK UP

Categorization exp26 run 1780

Full reconstructed & Require relationship with CDCDedxTrack → built track in CDC

Trg Type	Event type	#Event	2D Track (new CDC 2D and nth = 16)		3D NN Track (w/o ADC LUT)		3D NN Track (w/ ADC LUT)	
			#2D track > 0	# 2D track ==0	#NN track > 0	#NN track ==0	#NN track > 0	#NN track ==0
STT:	large Z	5822	5390	432	5158	232	4801	589
	fake track	2514	1919	595	1615	304	1181	738
	Signal	7244	7014	230	6873	141	6638	376
FFY:	large Z	4745	4385	360	4158	227	3883	502
	fake track	1982	1550	432	1291	259	1004	546
	Signal	4168	4048	120	3965	83	3870	178
FYO:	large Z	4541	4317	224	4159	158	3973	344
	fake track	1214	979	235	849	130	689	290
	Signal	3623	3519	104	3457	62	3382	137

large Z : #offline track > 0 AND #offline track with |z|<1 ==0

fake track #offline track ==0

Signal: #offline track > 0 AND #offline track with |z|<1 >0

Categorization exp26 run 1261

Full reconstructed & Require relationship with CDCDedxTrack → built track in CDC

Trg Type	Event type	#Event	2D Track (new CDC 2D and nth = 16)		3D NN Track (w/o ADC LUT)		3D NN Track (w/ ADC LUT)	
			#2D track > 0	# 2D track ==0	#NN track > 0	#NN track ==0	#NN track > 0	#NN track ==0
STT:	large Z	4492	3624	868	3416	208	3121	503
	fake track	3833	1848	1985	1464	384	1071	777
	Signal	4457	4007	450	3868	139	3695	312
FFY:	large Z	4116	3536	580	3325	211	3068	468
	fake track	2926	1760	1166	1414	346	1119	641
	Signal	2497	2321	176	2252	69	2168	153
FYO:	large Z	3846	3410	436	3274	136	3083	327
	fake track	1792	1149	643	974	175	782	367
	Signal	2135	1995	140	1938	57	1864	131

large Z : #offline track > 0 AND #offline track with |z|<1 ==0

fake track #offline track ==0

Signal: #offline track > 0 AND #offline track with |z|<1 >0

Categorization Check consistence exp26 run 1261

Full reconstructed & Require relationship with CDCDedxTrack → built track in CDC

Trg Type	Event type	#Event from Full reconstructed (w/ relation to CDC	#Event from sudo scripts (CDC reconstructed only)
STT:	large Z	80	77
	fake track	95	105
	Signal	55	48
FFY:	large Z	67	67
	fake track	73	78
	Signal	24	19
FYO:	large Z	53	60
	fake track	40	40
	Signal	21	14

large Z : #offline track > 0 AND #offline track with |z|<1 ==0

fake track #offline track ==0

Signal: #offline track > 0 AND #offline track with |z|<1 >0

Different input…

Pick up best n wire though all steoro SL → Not work at all

Reason:

$$\frac{z_{cross} - z_B}{Z_F - Z_B} = \frac{\phi_{cross} - \phi_B}{\phi_F - \phi_B}$$

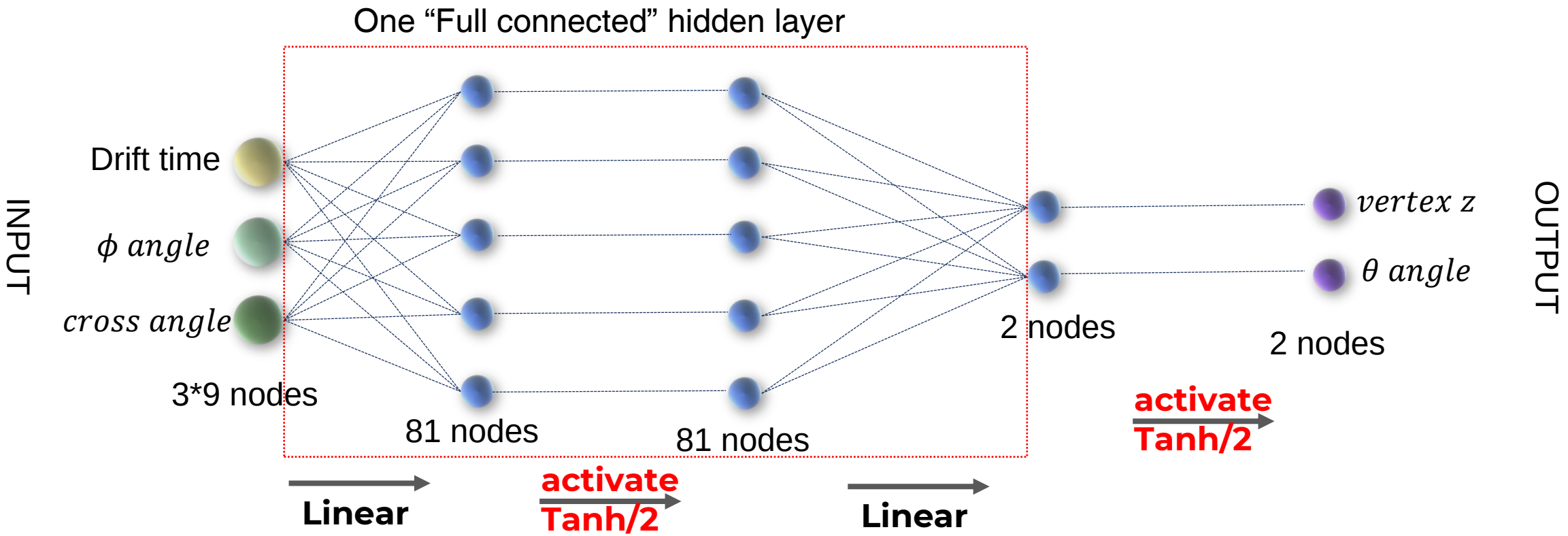
$$z_0 = z_{cross} - \cot\theta_0 \frac{2\alpha}{\omega}$$

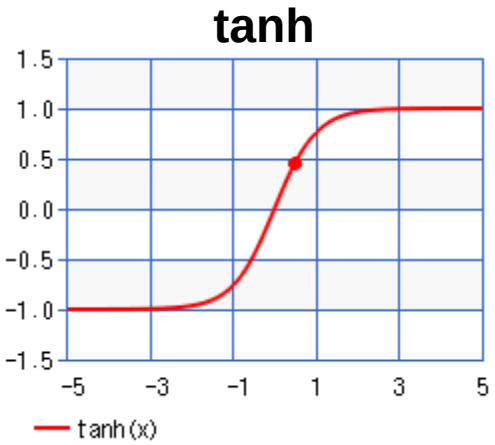
Many const are related to SL Z_F Z_B…

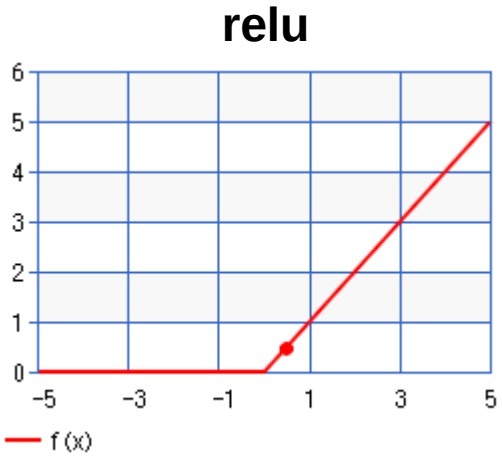
Try to input r_wire or z_F Z_b --> not work

→Next: try to directly input z cross? (with out drift time correction)

Activate function

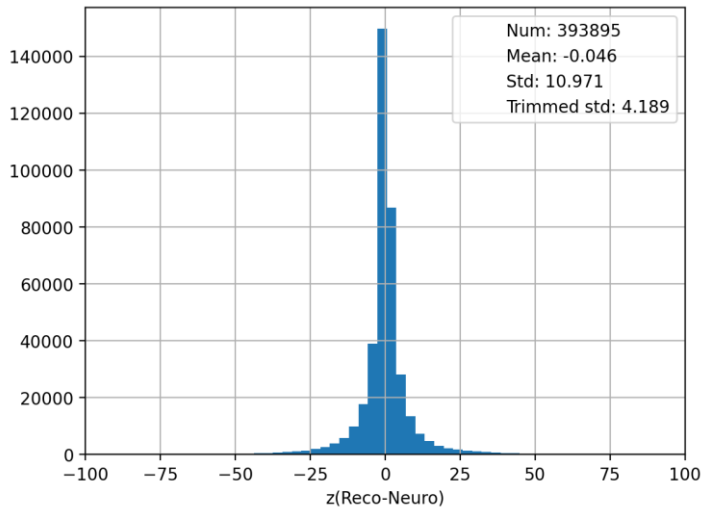


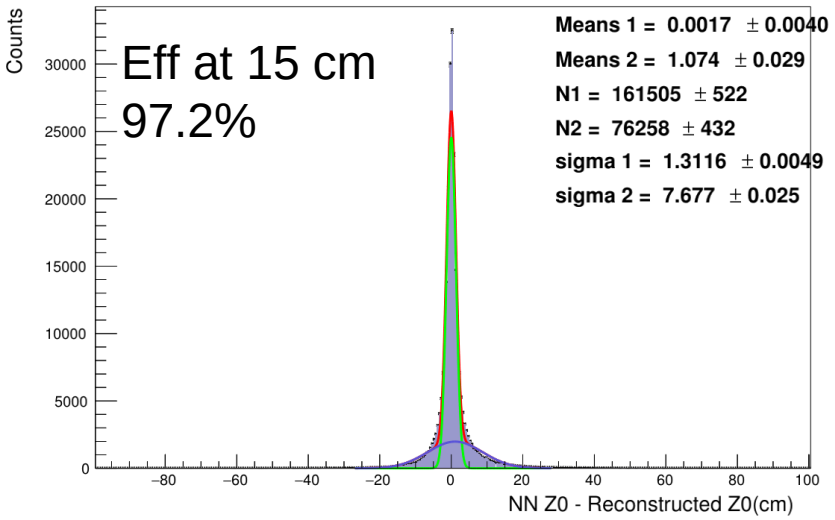


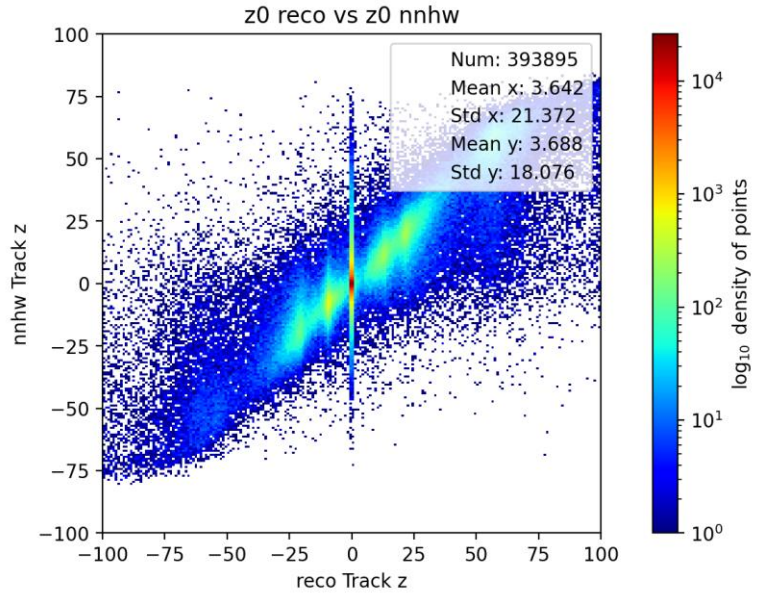


Activate function -removing

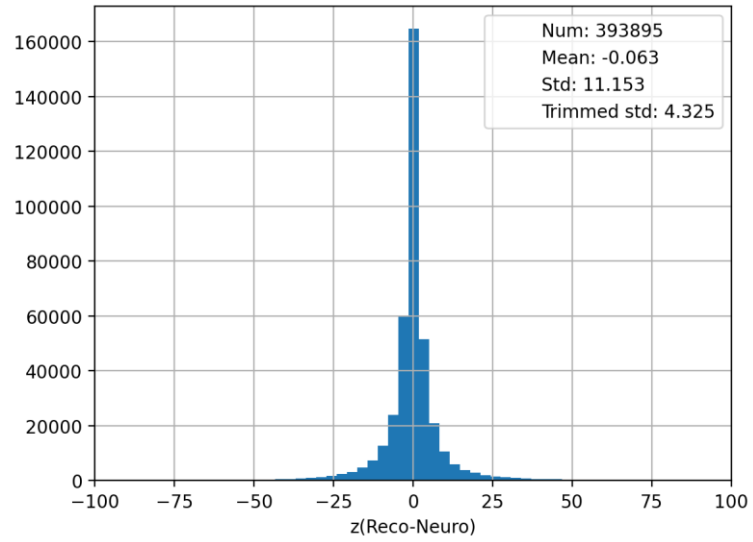
Extra wires 1 HD 3 (control group)

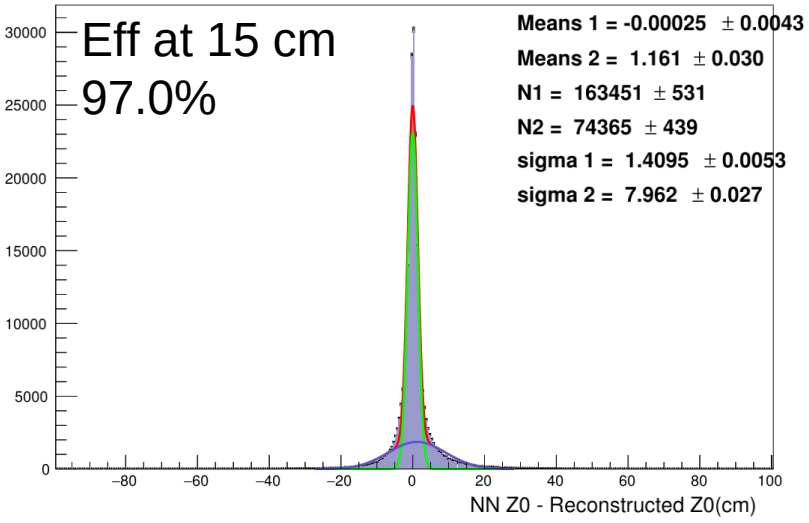


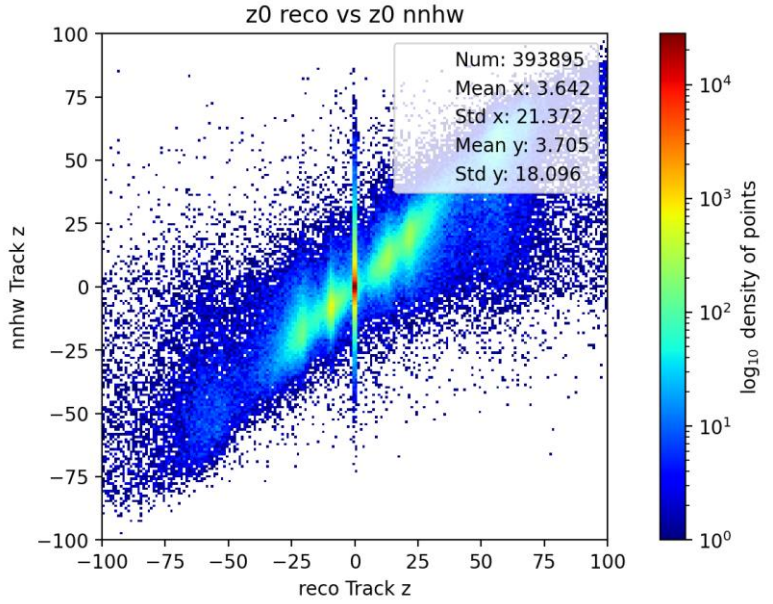




Remove final act func

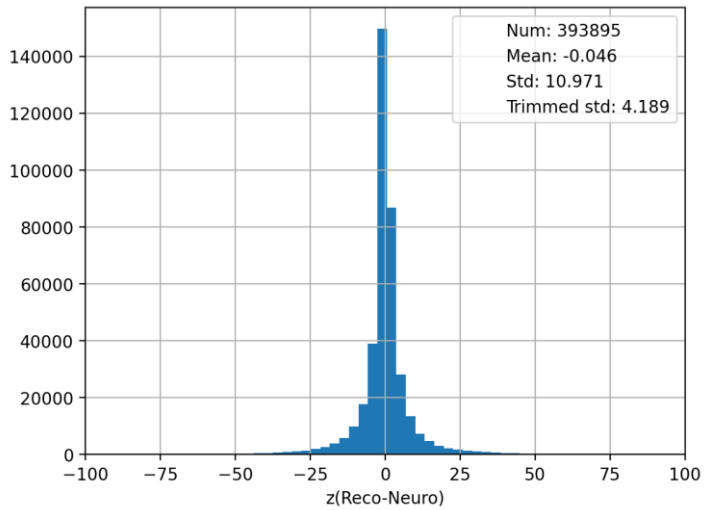


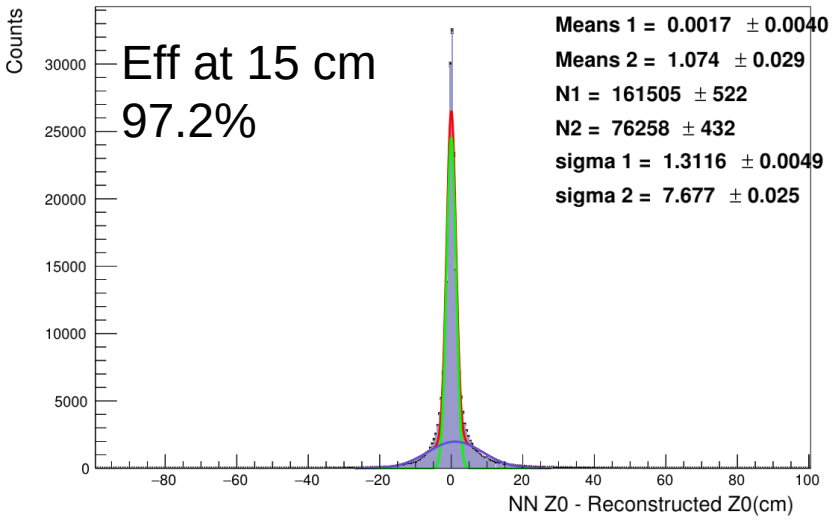


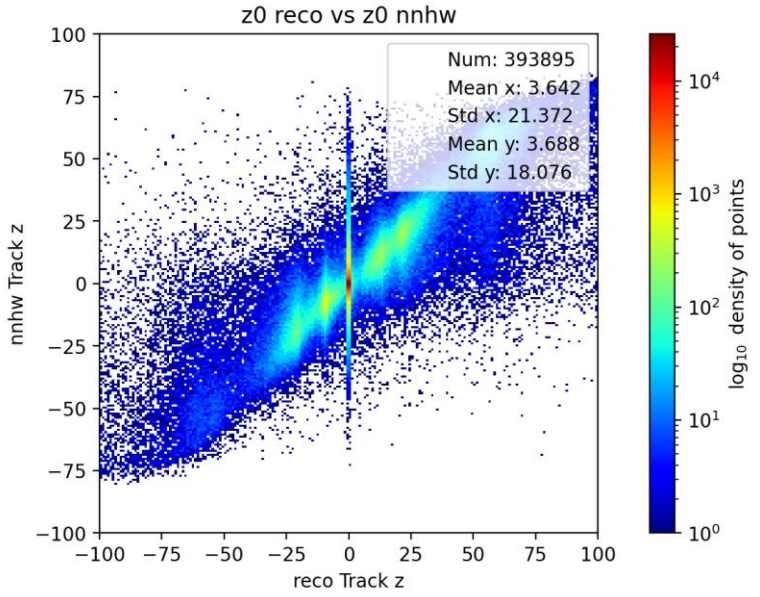


Activate function -relu

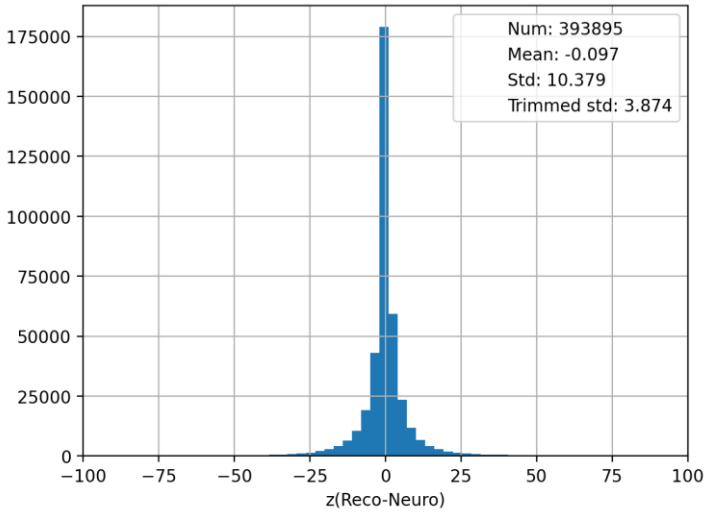
Extra wires 1 HD 3 (control group)

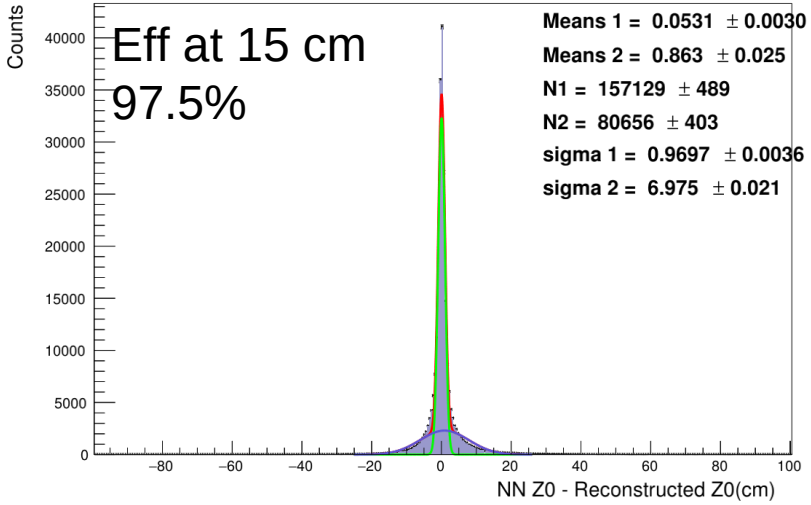


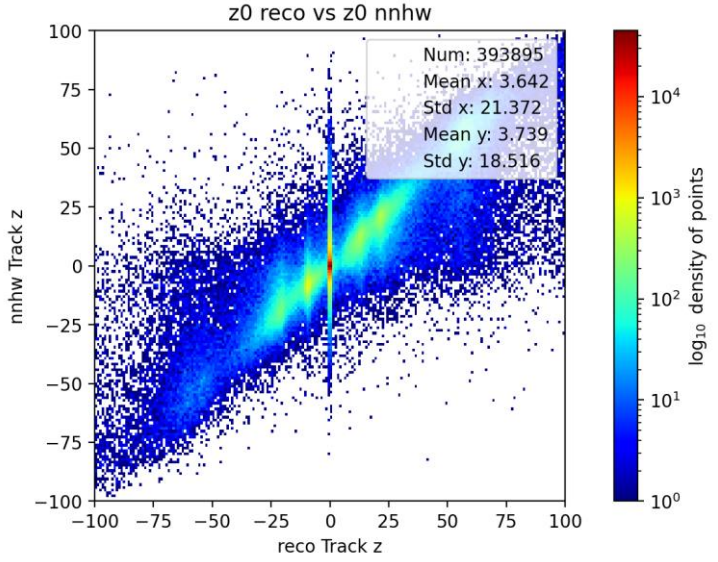




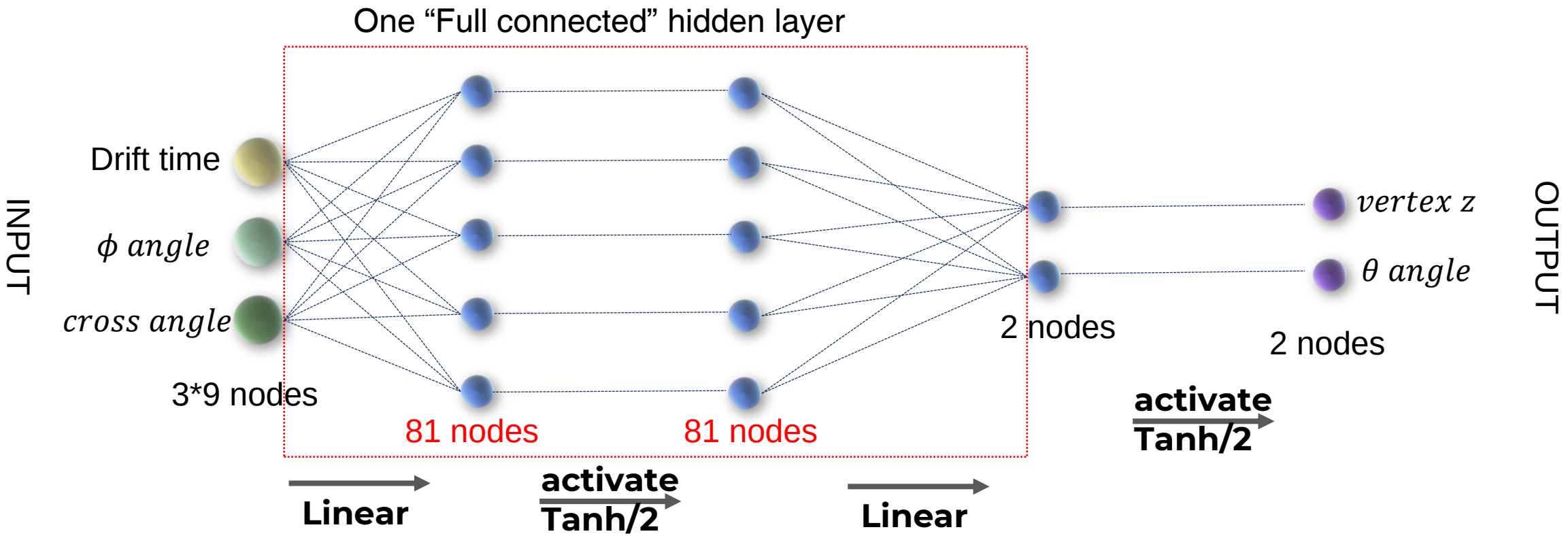
Remove final act func & Using relu instead





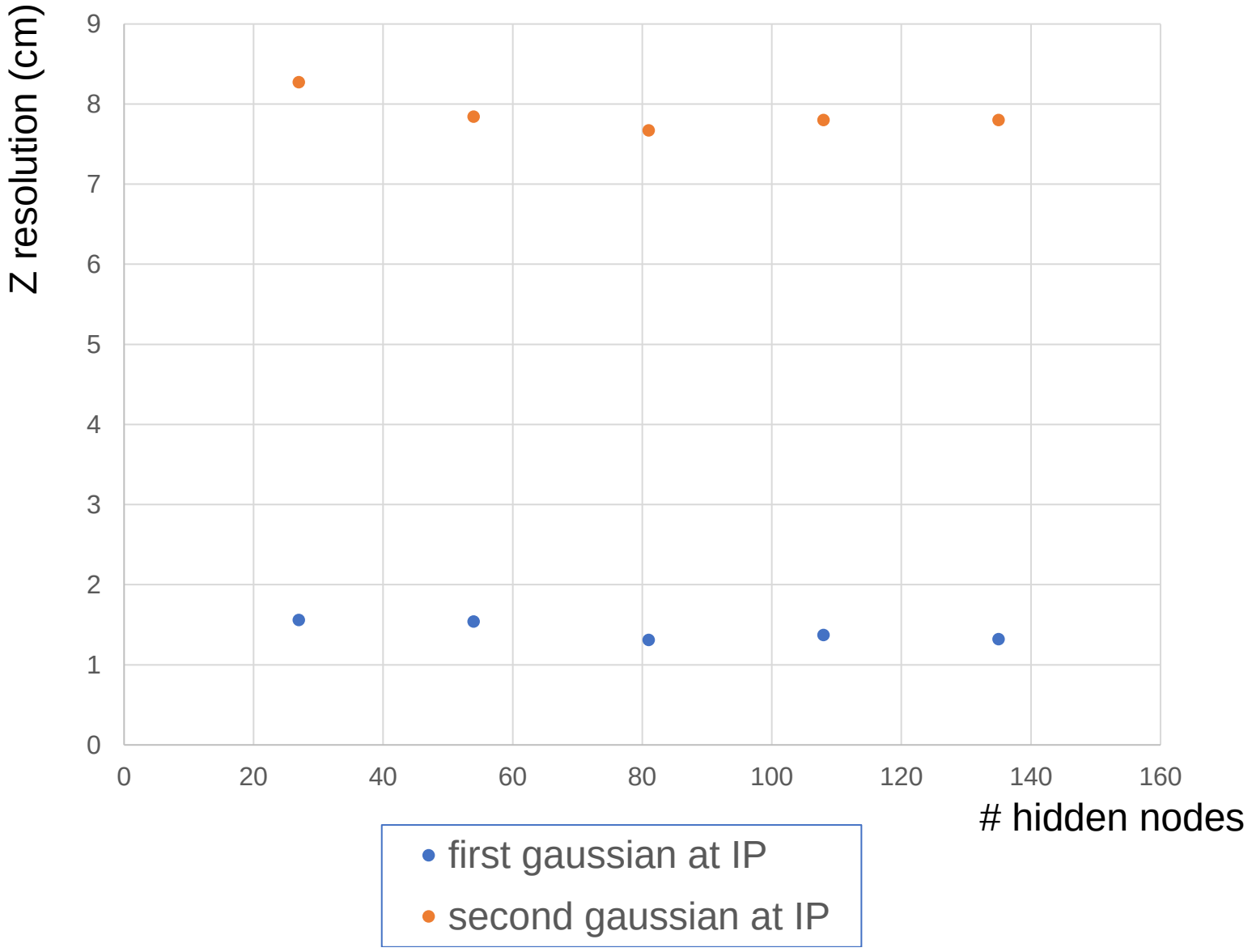


Activate function



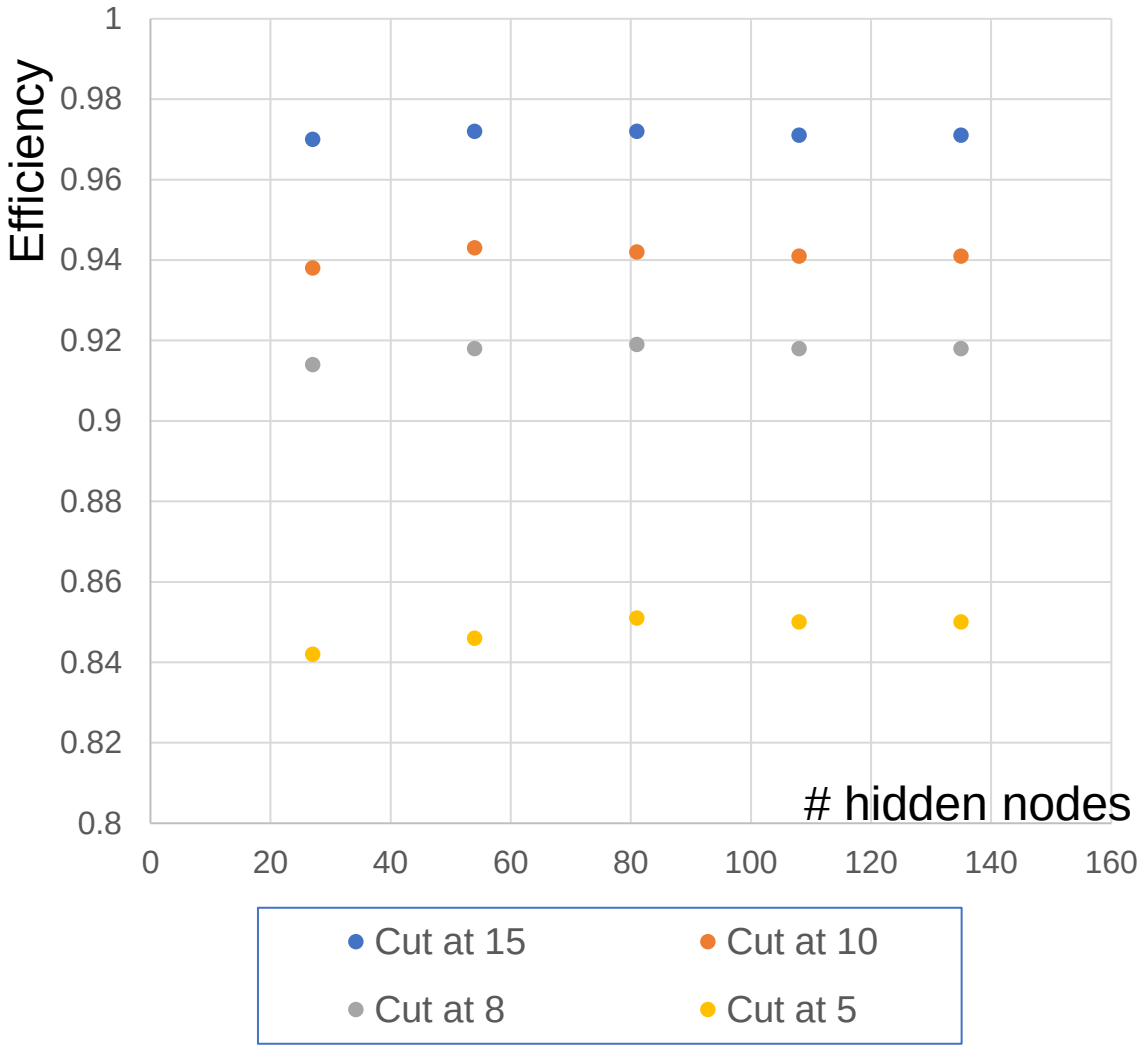
Change the hidden nodes for every hidden layer. For simplifying keep it the same

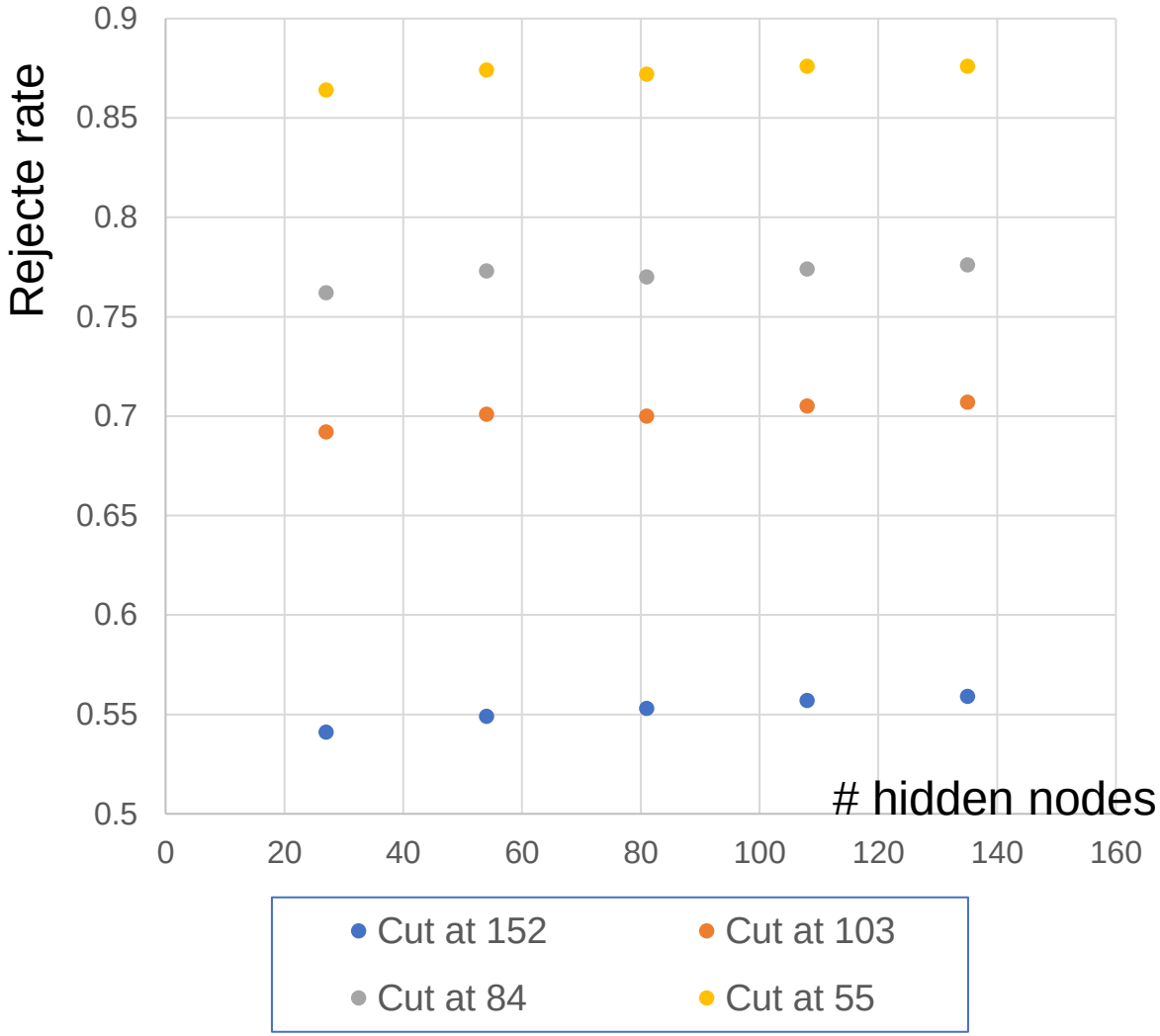
Training result – z resolution at IP



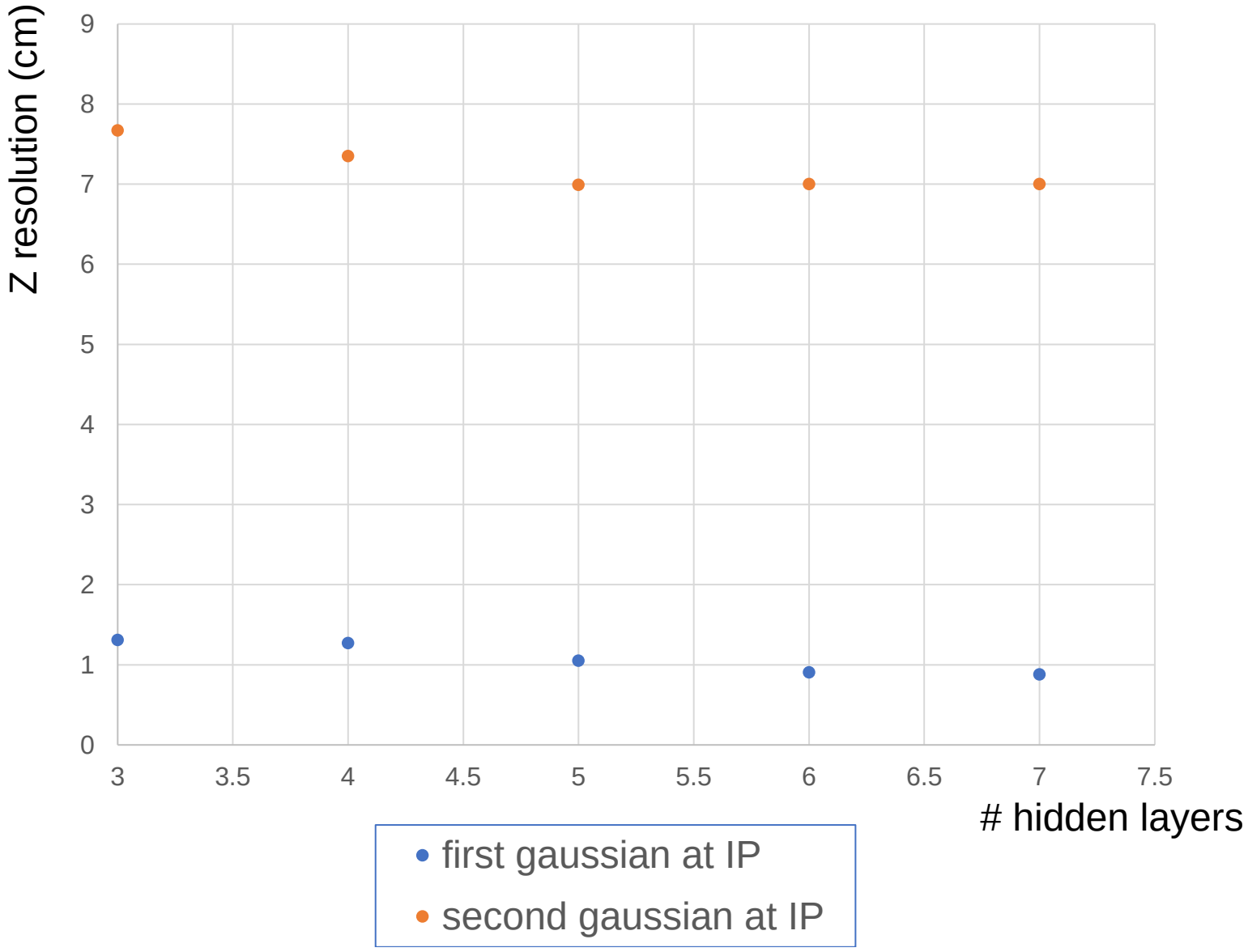
81 nodes should be best ?

Training result – z resolution at IP



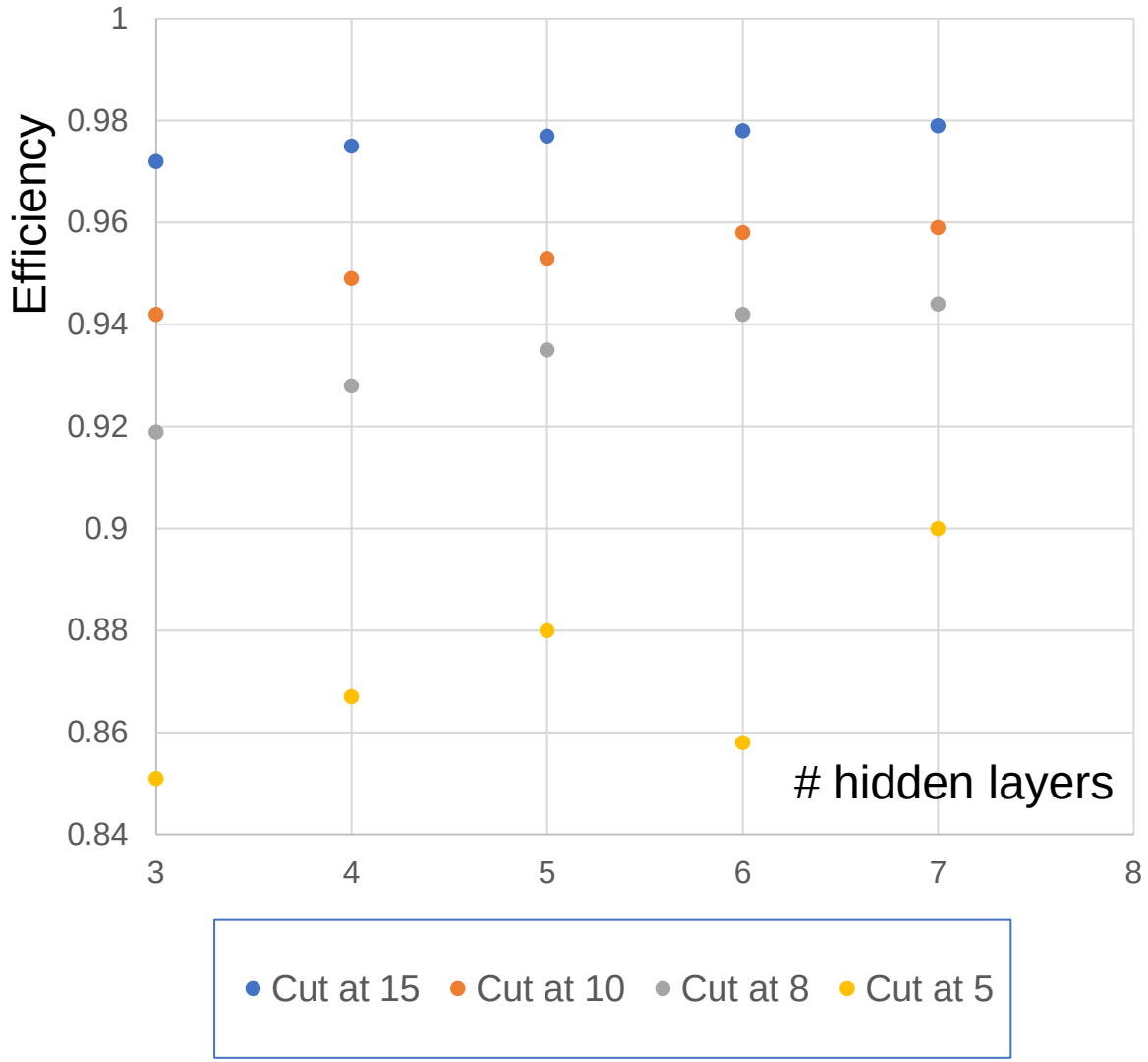


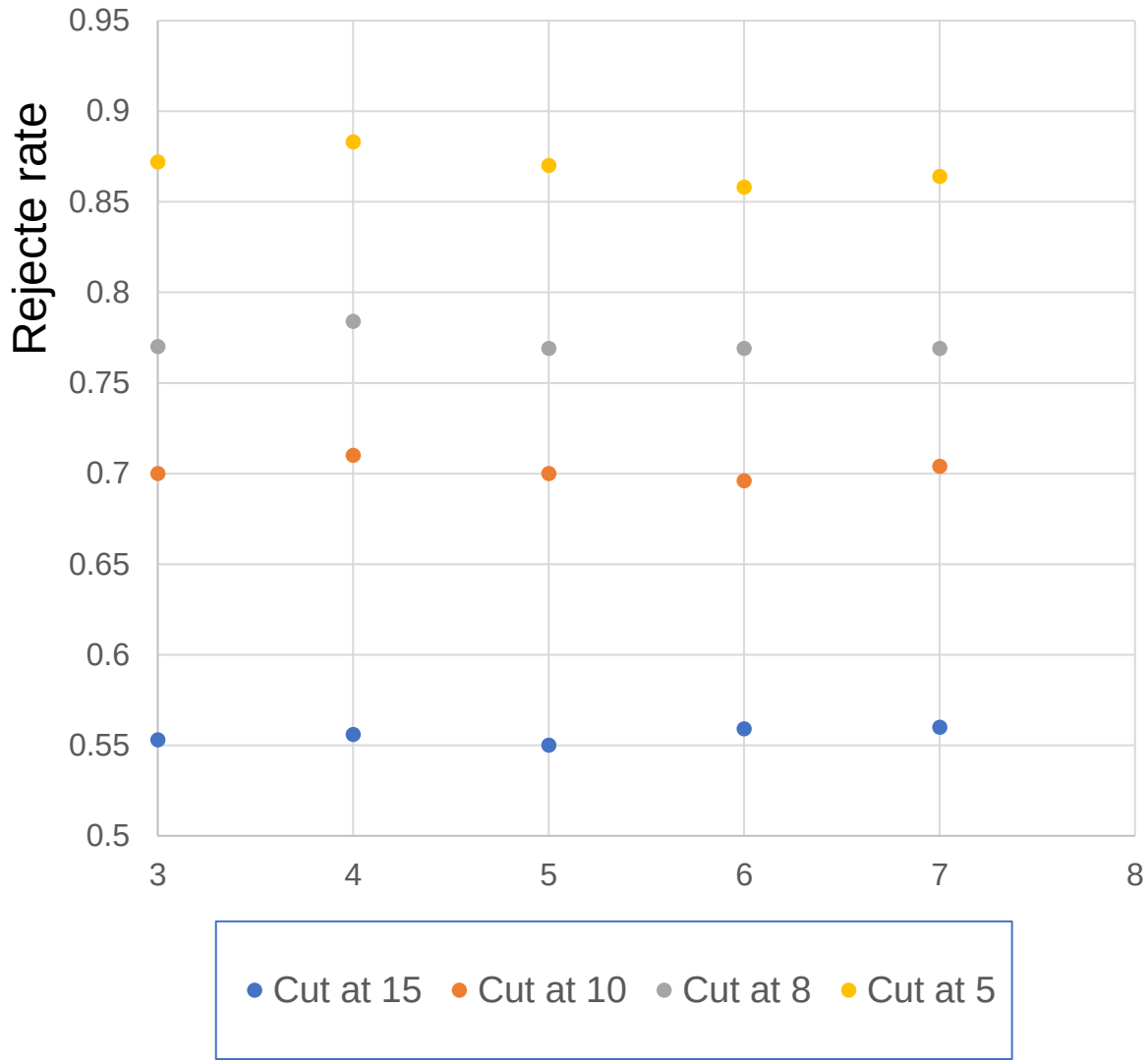
More More hidden layer?



Still improve?

More More hidden layer?

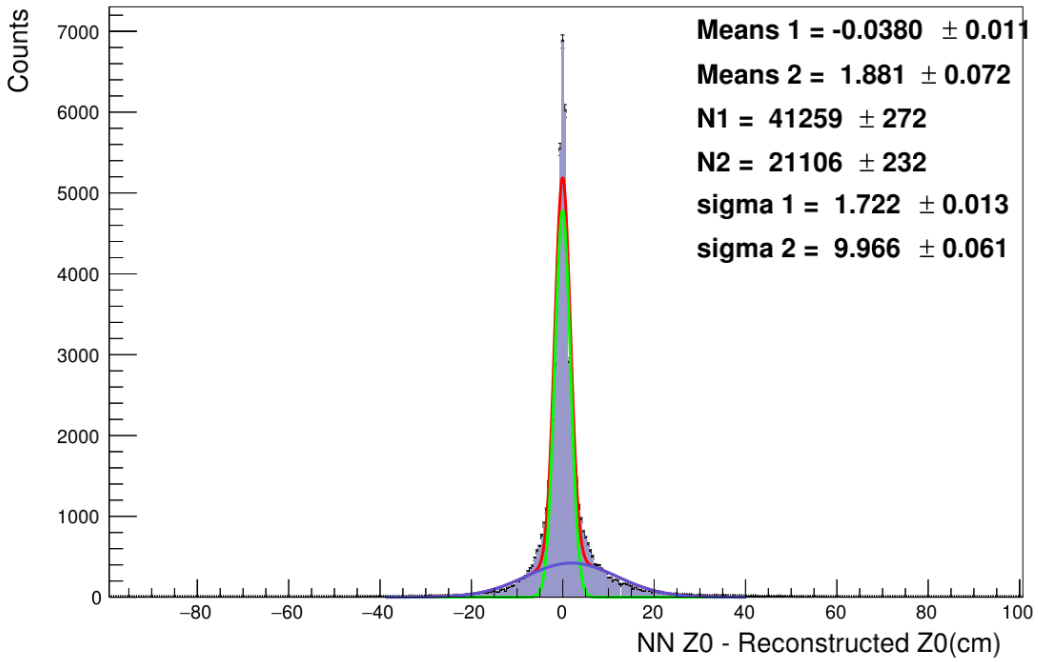




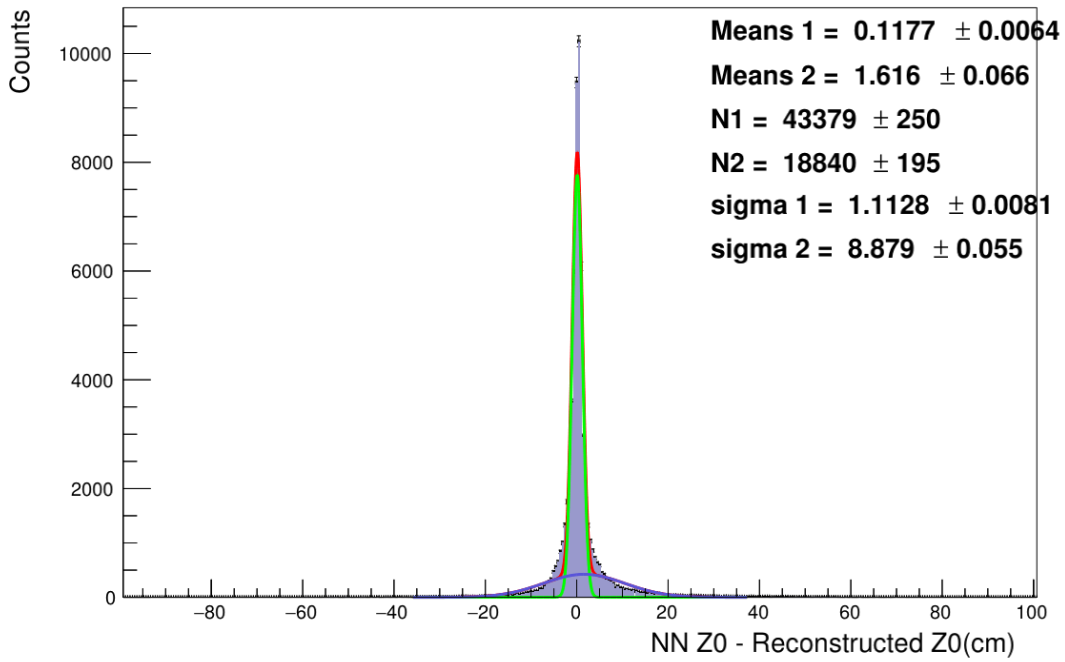
Overfitting?

Exp26 run96

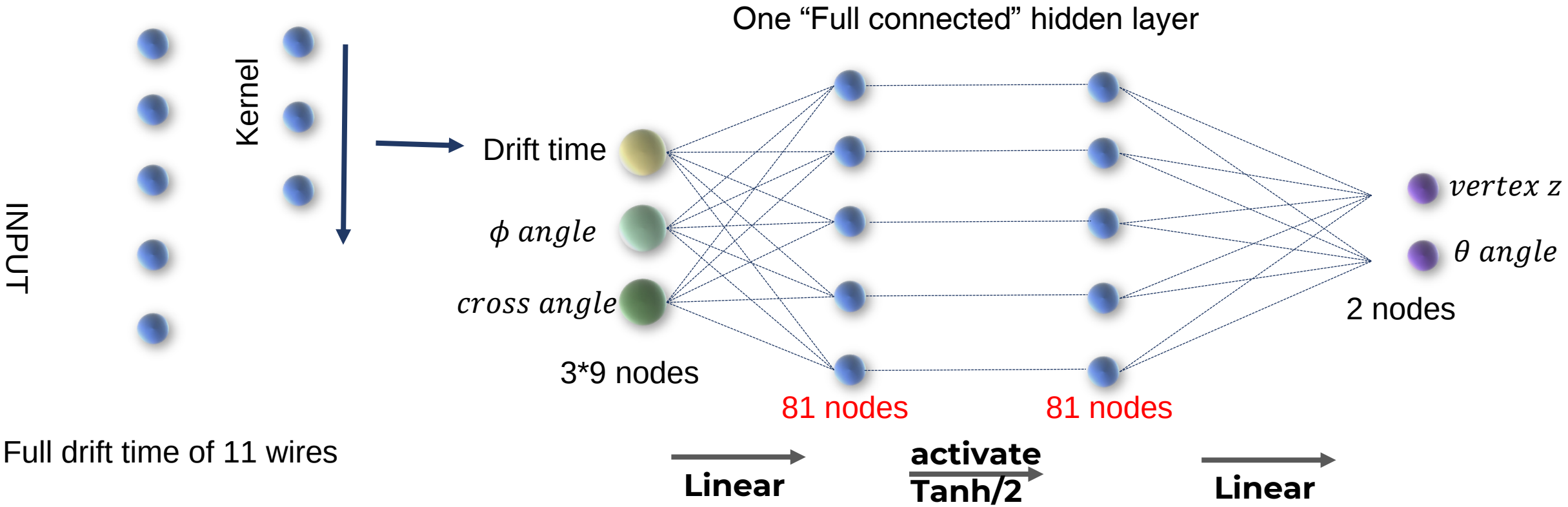
HD3



HD7



Different input...



Full drift time of 11 wires

Different input…

Pick up best n wire though all steoro SL

Not Bkg -> Not bad drift time -> has L/R -> fastest

L/R unknown is high

1. L/R unknown rate is high for extra wires

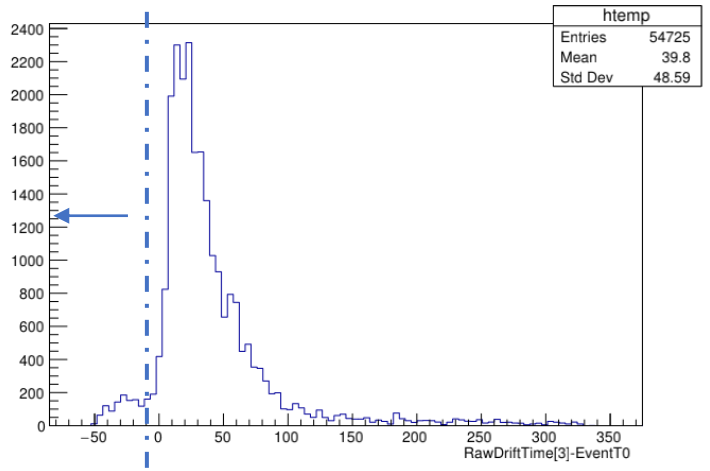
in Exp26 run1780:

	L/R known	L/R unknow	Bad drift time*
Extra wire 1	76%	19%	5%
Extra wire 2	70%	26%	4%
Extra wire 3	48%	46%	6%

1/5 to 1/2 extra wires had unknowed L/R

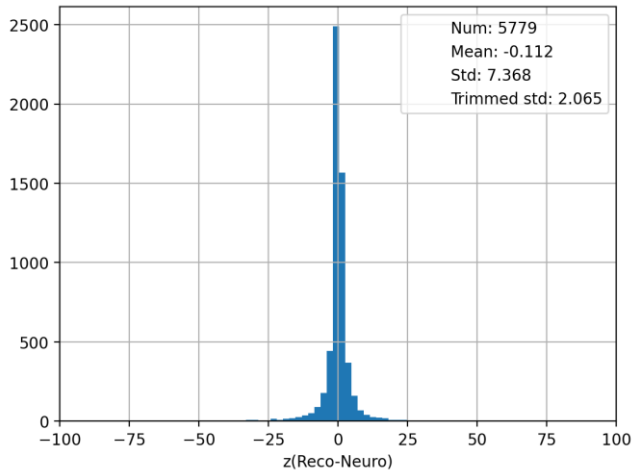
Bad drift time:

Drift time – Event T0

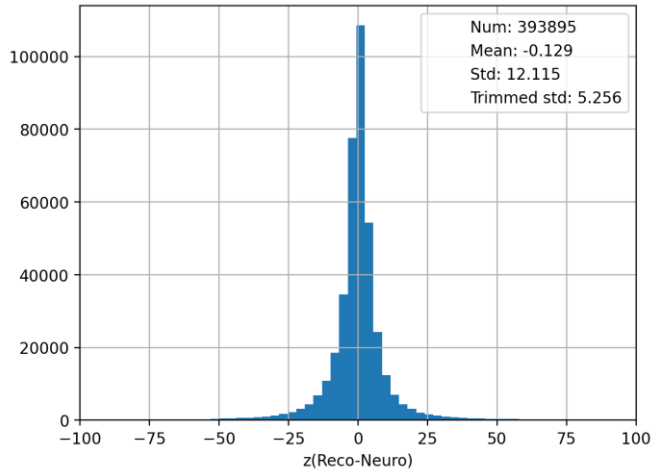


Event T0 from Event Time Finder simulation

If we only use data with known L/R:



Using all data



Only 1.4% data have known L/R for every wires

What if not use LR with more hidden layer?

Selection: Do not require LR, fastest & drift time >-10 (reject bad drift time)

Input change dirft time (-1~1) to (0,1)

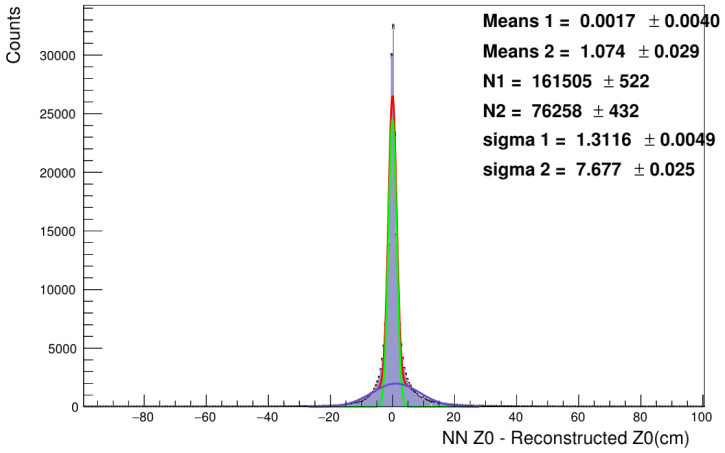
Modified batch size to 1024 from 2048 (This made improvement, so I keep it)

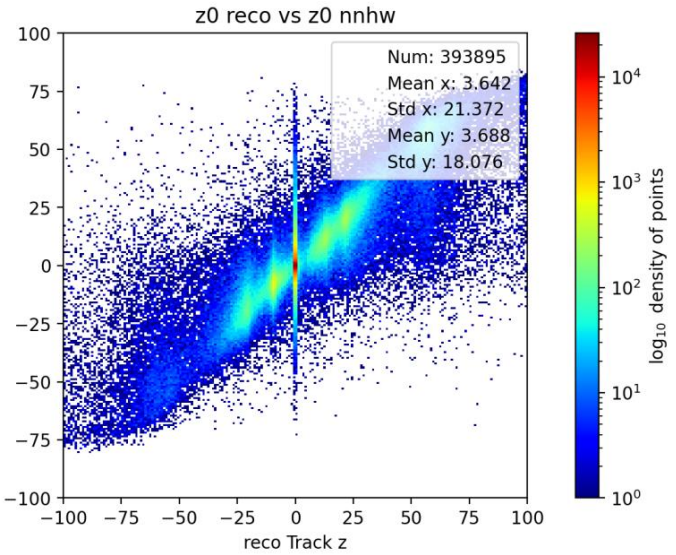
Remove batchnorm first, to see if we could train it to same result

What if not use LR with more hidden layer?

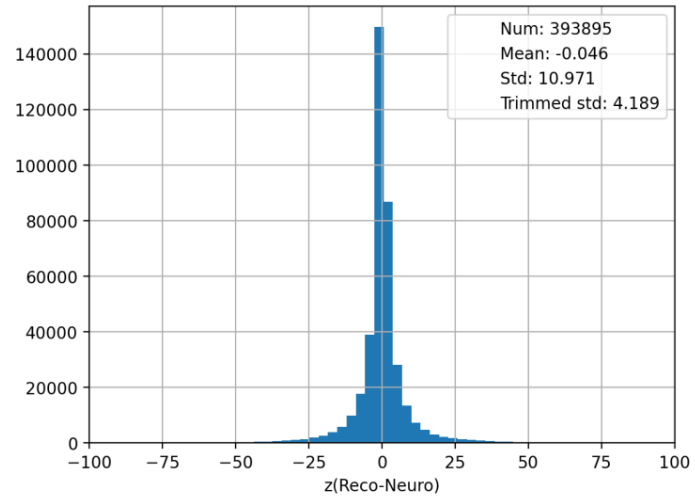
Extrawire1 HD3 w/o LR

IP resolution



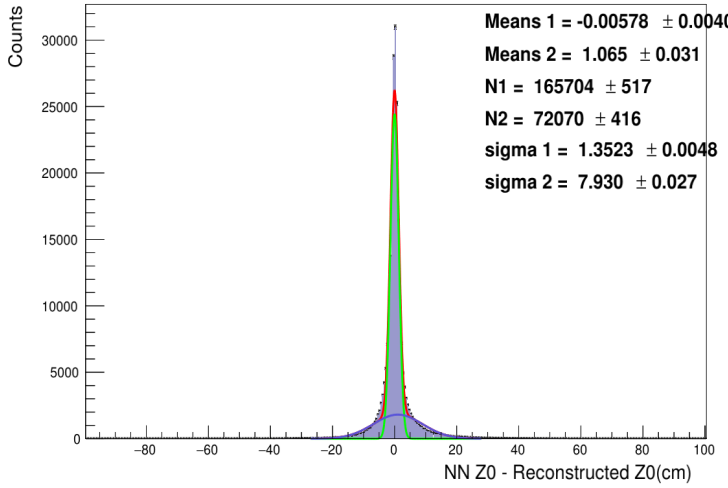


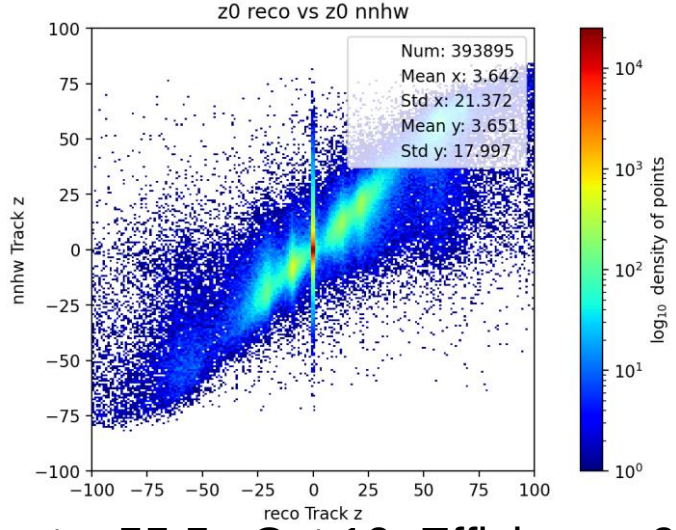
Total z diff

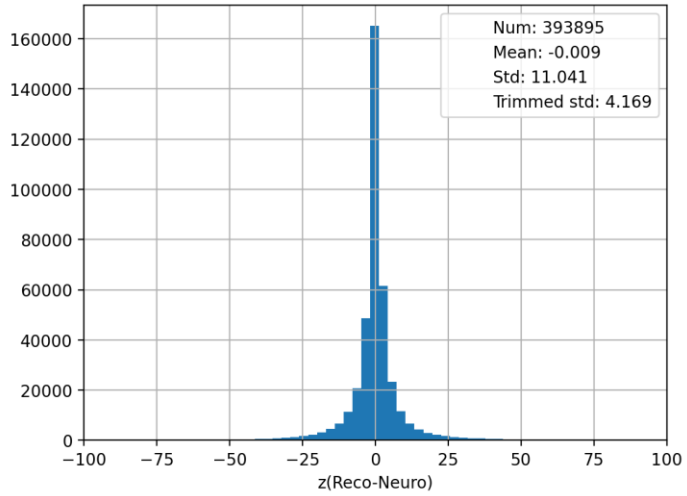


Cut 15: Efficiency:97.2 Rejected rate :55.3 Cut 10: Efficiency:.94.2 Rejected rate :70.0

Extrawire1 HD3 w/ LR

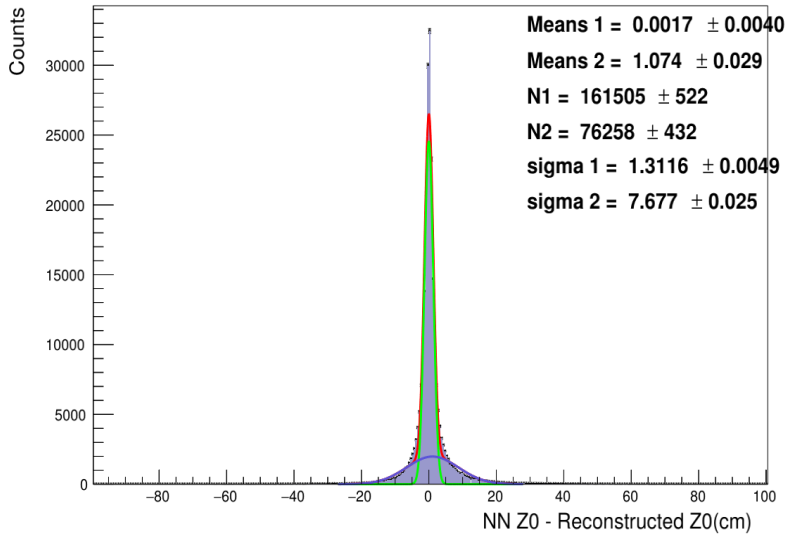




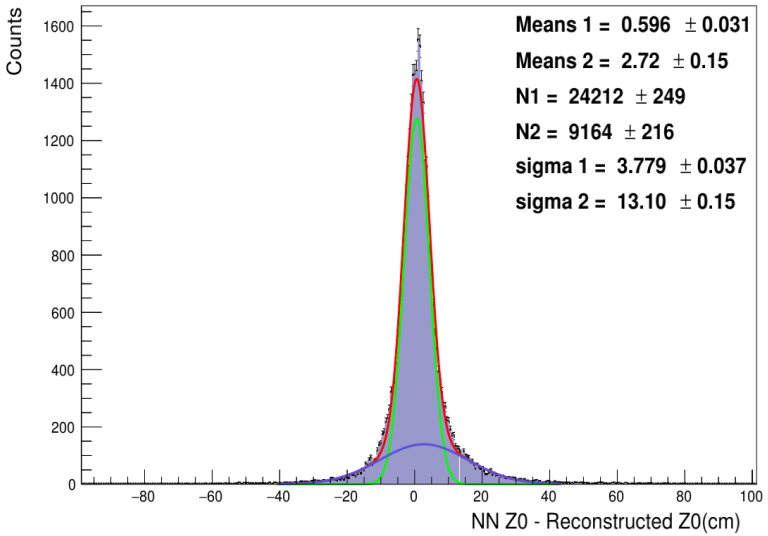


Cut 15: Efficiency:97.1 Rejected rate :55.5 Cut 10: Efficiency:.94.2 Rejected rate :70.6

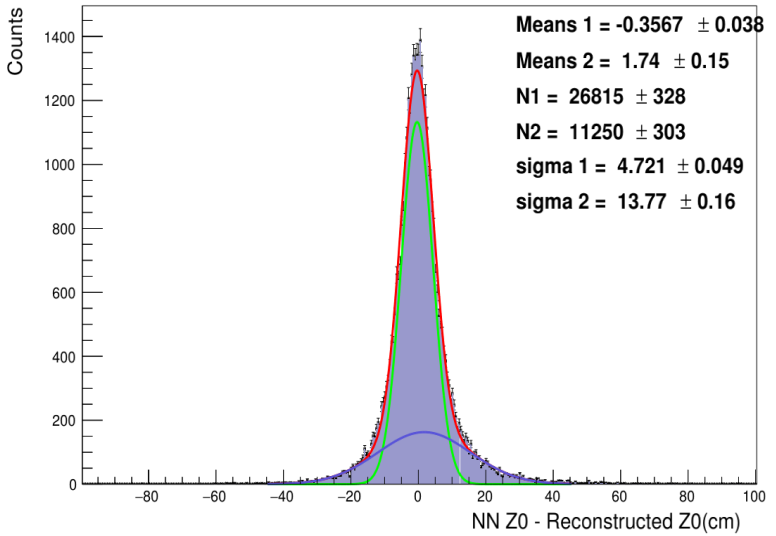
Extrawire1 HD3 w/o LR



Z<1

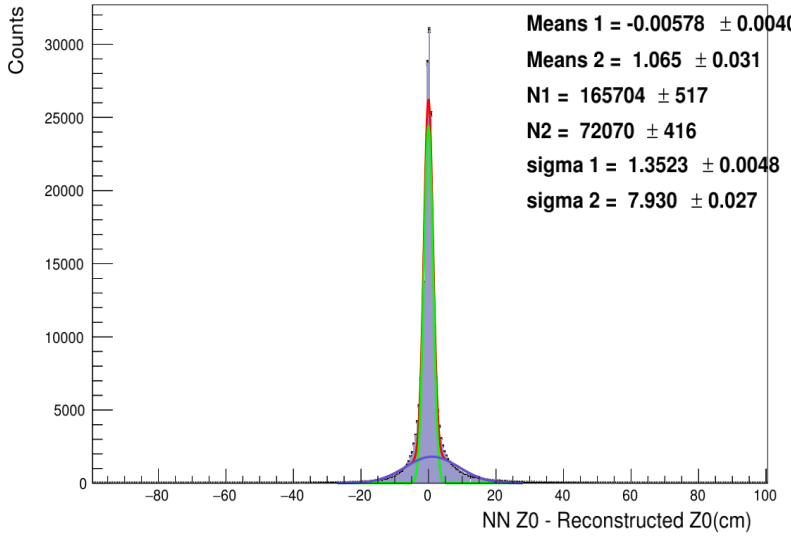


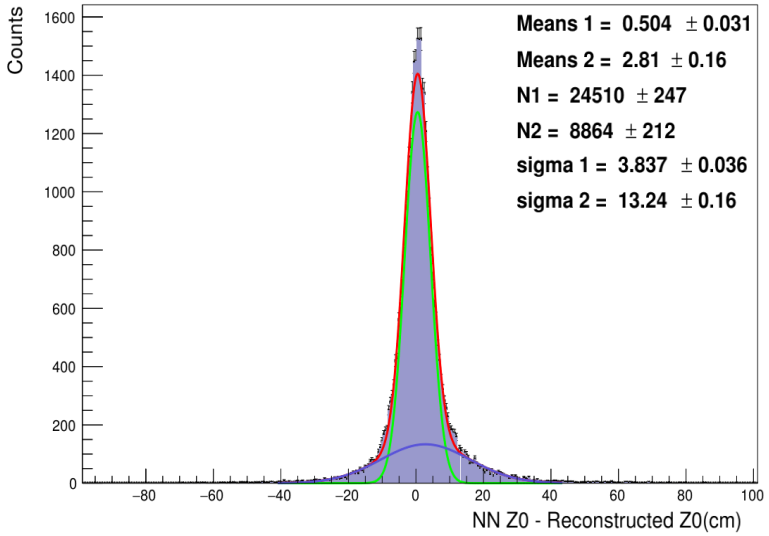
1<Z<10

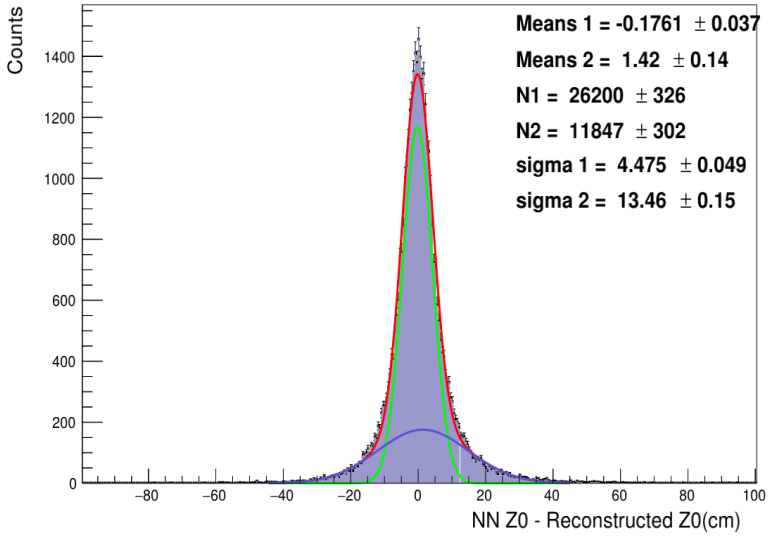


10<Z<20

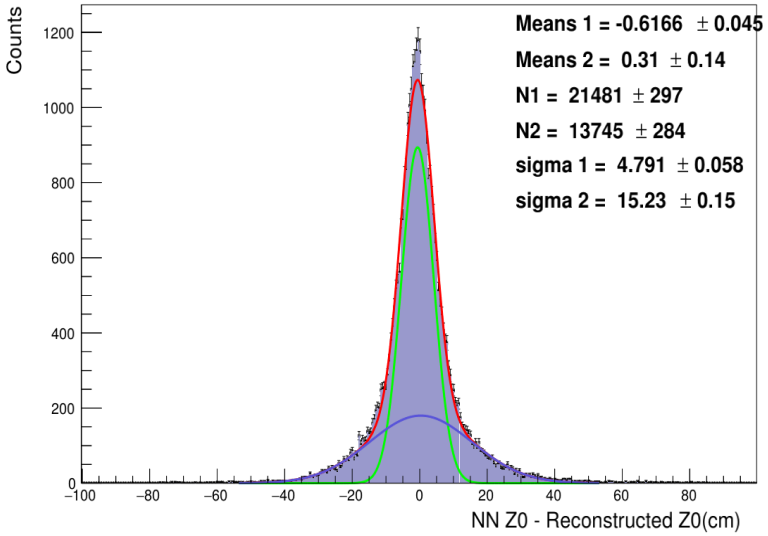
Extrawire1 HD3 w/ LR



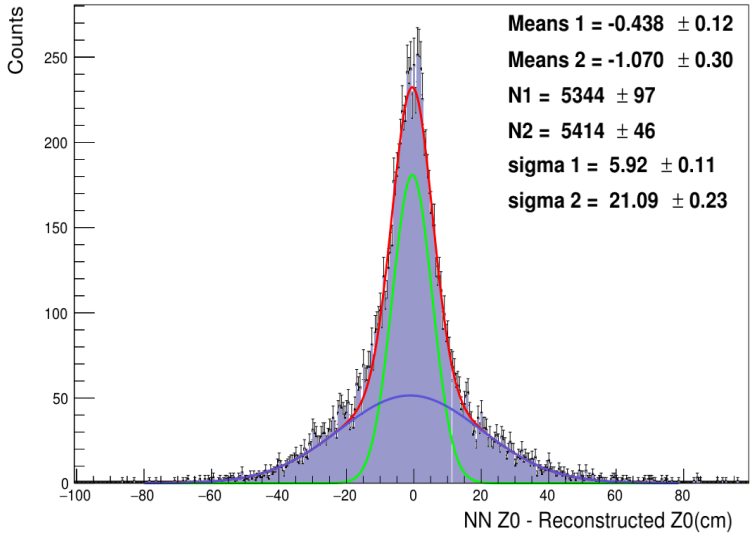




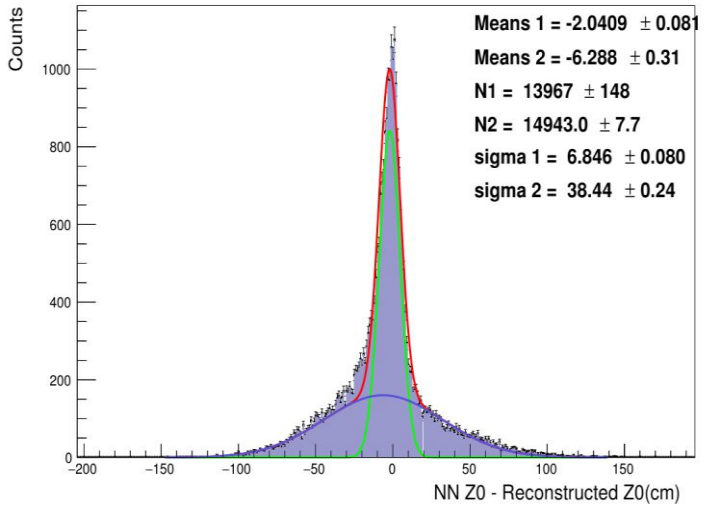
Extrawire1 HD3 w/o LR



20<Z<30

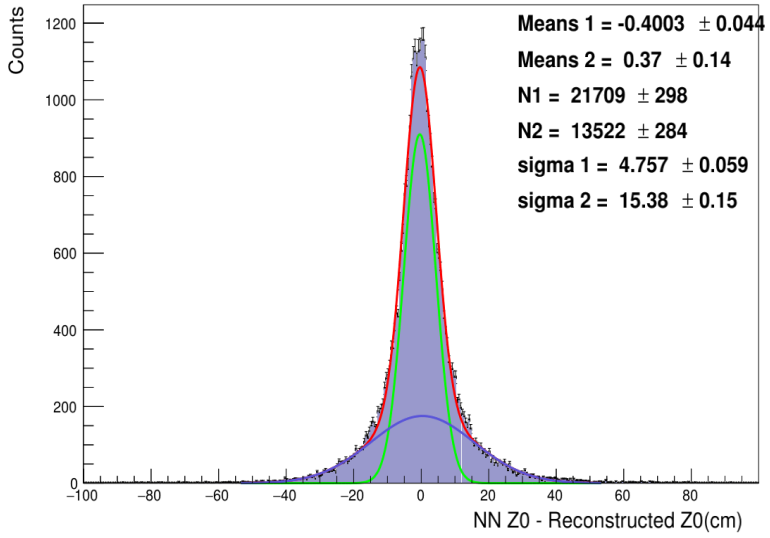


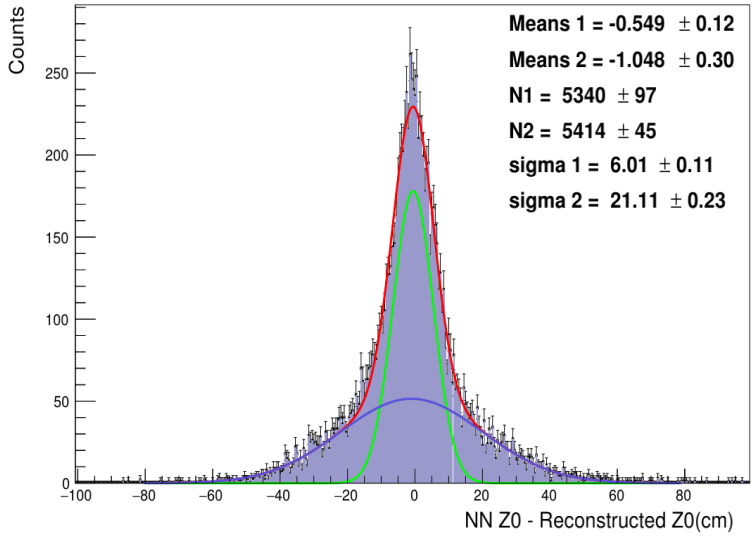
30<Z<40

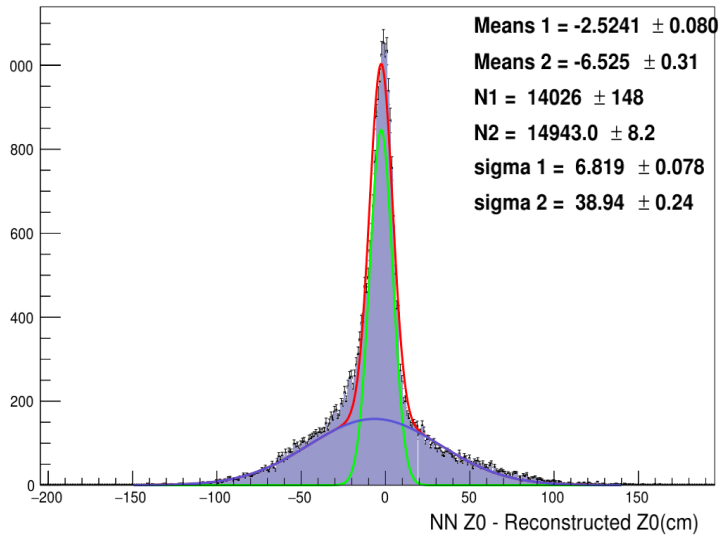


40<Z

Extrawire1 HD3 w/ LR







Trg Type	Event type	Event ratio in exp26run1780	Event ratio in exp26run1261
STT: Signal Track Trigger	Bkg from large Z	30.8 %	6078 47.6%
	Bkg from fake track	7.2 %	1947 15.2%
	Signal	62.0%	4757 37.2%
FFY: 2 2D track & 1 3D track	Bkg from large Z	36.6 %	4110 43.1%
	Bkg from fake track	9.0 %	1471 15.4%
	Signal	54.4%	3958 41.5%
FYO: 1*2D track & 1*3D track with angle >90	Bkg from large Z	39.2 %	3546 45.6%
	Bkg from fake track	4.4 %	869 11.2%
	Signal	56.4%	3358 43.2%

Training result

Training based on ADC cut sample – Not rejected

NN Type	Std (cm)	σ at IP (cm)	Efficiency (cut at 15/ 10/ 8 / 5) (%)	Reject rate (cut at 15/ 10/ 8 / 5) (%)
1 Extra Wire w/o LR + 3 hidden layer	11.0	1.3 | 7.8	97.2/ 94.2/ 91.9/ 85.1	55.3/ 70.0/ 77.0/ 87.2
1 Extra Wire w/ LR + 3 hidden layer	11.0	1.3 | 7.9	97.1/ 94.2/ 91.8/ 85.3	55.5/ 70.6/ 77.5/ 87.4
2 Extra Wire w/o LR + 3 hidden layer	10.7	1.3 | 7.7	97.3/ 94.5/ 92.3/ 85.9	56.1/ 70.7/ 77.8/ 88.0
2 Extra Wire w/ LR + 3 hidden layer(old)	10.9	1.7 | 8.0	97.3/ 94.4/ 92.0/ 84.7	55.8/ 72.2/ 78.9/ 87.9
3 Extra Wire w/o LR + 3 hidden layer	10.6	1.3 | 7.6	97.4/ 94.6/ 92.4/ 86.0	55.8/ 70.7/ 77.8/ 87.8
3 Extra Wire w/ LR + 3 hidden layer(old)	10.7	1.7 | 8.0	97.3/ 94.4/ 92.1/ 85.0	55.8/ 72.0/ 78.6/ 87.7
Standard	13.2	2.7|11.0	94.7/ 89.5/ 85.7/ 74.2	54.8/ 71.3/ 78.1/ 87.4
Standard +batchnorm	12.3	2.6 | 9.8	96.5/ 92.6/ 89.5/ 79.1	52.8/ 69.0/ 75.8/ 85.7
Standard +batchnorm+2 hidden layers	11.8	2.2 | 9.3	96.6/ 93.0/ 90.1/ 81.3	53.6/ 70.1/ 77.1/ 86.8
Standard +batchnorm+3 hidden layers	11.5	1.7 | 8.3	96.9/ 93.6/ 90.1/ 82.9	54.6/ 70.0/ 76.6/ 86.2

At least, for 3 hidden layers, L/R is not needed?

Will try 1 or 2 hidden layer also

If extra wire really used?

w/o L/R case: weights for first input extra wire

[-0.24813604354858398, -0.10292977094650269, -0.17695829272270203, 0.00966159999370575, 0.1597103327512741, -0.2265278398990631, 0.31949785351753235, 0.09929633140563965, -0.18960373103618622, 0.18001483380794525, 0.46470117568969727, -0.22326022386550903, 0.4799828827381134, -0.029258545488119125, -0.04485153779387474, 0.15445296466350555, 0.05755236744880676, -0.04450396075844765, 0.0548921637237072, -0.09542621672153473, 0.02211681753396988, -0.03317129239439964, -0.17439308762550354, -0.08265548944473267, -0.20804476737976074, 0.006292895879596472, -0.1899934858083725, 0.03650810569524765, 0.07716267555952072, -0.29153352975845337, 0.05761154741048813, 0.020245296880602837, -0.1758495271205902, 0.12332288920879364, 0.19053097069263458, -0.21652907133102417, -0.08756624162197113, -0.020610634237527847, -0.07745809853076935, -0.2680374085903168, 0.20523694157600403, -0.19775904715061188, -0.3678823411464691, 0.16456829011440277, -0.10474462807178497, -0.06493524461984634, 0.008172446861863136, -0.07382170110940933, -0.18212975561618805, -0.5558742880821228, 0.5700501203536987, 0.285658597946167, 0.21900387108325958, 0.7400479316711426, -0.06684859097003937, -0.058145735412836075, 0.5379076600074768, -0.004072423093020916, -0.07591185718774796, 0.46645450592041016, 0.3436751663684845, 0.05206478387117386, -0.1091272160410881, 0.06431704014539719, 0.14220617711544037, -0.14595340192317963, -0.0917854905128479, 0.08792682737112045, -0.09119176864624023, -0.12810276448726654, 0.10797716677188873, -0.15082700550556183, -0.22428378462791443, -0.03472667932510376, -0.17347513139247894, 0.11283260583877563, -0.012019463814795017, -0.18513266742229462, -0.03671427071094513, 0.115293487906456, -0.1858721673488617, -0.10385676473379135, 0.0563088022172451, -0.1440611183643341, 0.6056587100028992, 0.11828280240297318, -0.16134104132652283, 0.4791528284549713, -0.042354732751846313, -0.19736701250076294, 0.35837510228157043, -0.06116437539458275, -0.09977502375841141, 0.30060163140296936, -0.02834823913872242, -0.11343704164028168, -0.0929100438952446, 0.01150817982852459, 0.20117467641830444, -0.0036697620525956154, 0.2314114272594452, 0.21343819797039032, -0.10605911165475845, 0.19504447281360626, 0.18565228581428528, -0.30705079436302185, 0.09346219897270203, 0.2430061548948288],

w/ L/R case: weights for first input extra wire

(It just said, not zero or very small value→ used)

[0.22343716025352478, -0.0938759595155716, 0.04122266173362732, -0.09648856520652771, 0.12756474316120148, 0.06424813717603683, 0.07470588386058807, 0.07257045060396194, 0.04113960638642311, -0.0399806834757328, -0.09422793984413147, 0.04587939754128456, 0.43344205617904663, 0.011563792824745178, -0.048001546412706375, 0.38240548968315125, -0.01461311336606741, 0.03518868237733841, 0.18670646846294403, 0.08883973956108093, 0.021211545914411545, 0.2103162407875061, -0.05419749394059181, -0.011362993158400059, -0.06486336141824722, -0.16127929091453552, 0.22180652618408203, -0.0849444642663002, -0.09436224400997162, 0.21658092737197876, -0.0094265416264534, -0.04000578820705414, 0.23730874061584473, -0.1105792224407196, -0.021637504920363426, 0.22088268399238586, 0.6184473633766174, 0.028800634667277336, -0.09913826733827591, 0.5335654616355896, 0.1273764967918396, -0.0919310674071312, 0.3105182349681854, 0.06525144726037979, -0.08297815173864365, 0.24419409036636353, 0.12163866311311722, -0.07329908758401871, -0.10020250827074051, -0.12976931035518646, 0.14274625480175018, -0.09672066569328308, 0.023003214970231056, 0.1417037695646286, -0.06654255837202072, 0.05499771237373352, 0.10206054151058197, 0.024930626153945923, 0.047063443809747696, 0.12508001923561096, -0.10276258736848831, -0.08443999290466309, -0.1050499826669693, -0.19503402709960938, -0.15189209580421448, -0.06640048325061798, -0.11852992326021194, 0.07945716381072998, -0.12189128994941711, -0.15803803503513336, 0.020767970010638237, -0.08703223615884781, -0.03631749376654625, 0.09463648498058319, 0.05772719159722328, 0.01925610937178135, 0.03465835005044937, 0.08929714560508728, 0.05752383545041084, 0.03377871960401535, 0.09248676151037216, 0.09663037955760956, 0.018963659182190895, 0.04112373664975166, -0.10039466619491577, 0.017538560554385185, -0.07717432081699371, 0.05747659504413605, 0.034616414457559586, 0.0056767831556499004, 0.010696266777813435, 0.07134998589754105, -0.07596204429864883, -0.028239743784070015, 0.04906168207526207, -0.14625610411167145, -0.006308637093752623, 0.011202408000826836, -0.0012855567038059235, -0.07635674625635147, -0.2064904272556305, -0.044596150517463684, -0.0559067502617836, -0.0038232163060456514, 0.016832474619150162, -0.02043667621910572, 0.01332989614456892, 0.005045489873737097],

Event categorization

			Exp26 Run1756 @peak luminosity $=2.3 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$			Exp26 Run1261 @peak luminosity $=4.6 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$	
		Definition	Ratio (%)	Trigger Rate (kHz)		Ratio (%)	Trigger Rate (kHz)
Triggered by CDC trigger (ffy OR fyo OR stt)	1. Signal	Not BG, fake track	15.2%	0.61		18.7%	2.13
	2. BG from large z	#(offline with |z0|>1cm) > 0 AND #(offline with |z0|<1cm) == 0	4.3%	0.17		8.3%	0.95
	3. Fake track	#(offline) == 0	1.8%	0.07		14.7%	1.67
Triggered by other trigger			78.7%	3.18		58.4%	6.66
	total		100%	4.04		100%	11.4

※ # of sampled events = 10000
※ HLT filtering is OFF in both runs.

Motivation

Sudo-san

Event categorization

			Exp26 Run1756 @peak luminosity $=2.3 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$			Exp26 Run1261 @peak luminosity $=4.6 \times 10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$	
		Definition	Ratio (%)	Trigger Rate (kHz)		Ratio (%)	Trigger Rate (kHz)
	1. Signal	Not BG, fake track	15.2%	0.61		18.7%	2.13
Triggered by CDC trigger (ffy OR fyo OR stt)	**2. BG from large z**	#(offline with |z0|>1cm) > 0 AND #(offline with |z0|<1cm) == 0	4.3%	0.17		8.3%	0.95
	3. Fake track	#(offline) == 0	1.8%	0.07		14.7%	1.67
Triggered by other trigger			78.7%	3.18		58.4%	6.66
	total		100%	4.04		100%	11.4

What new NN can reduce

What we plan to reduce with new NN

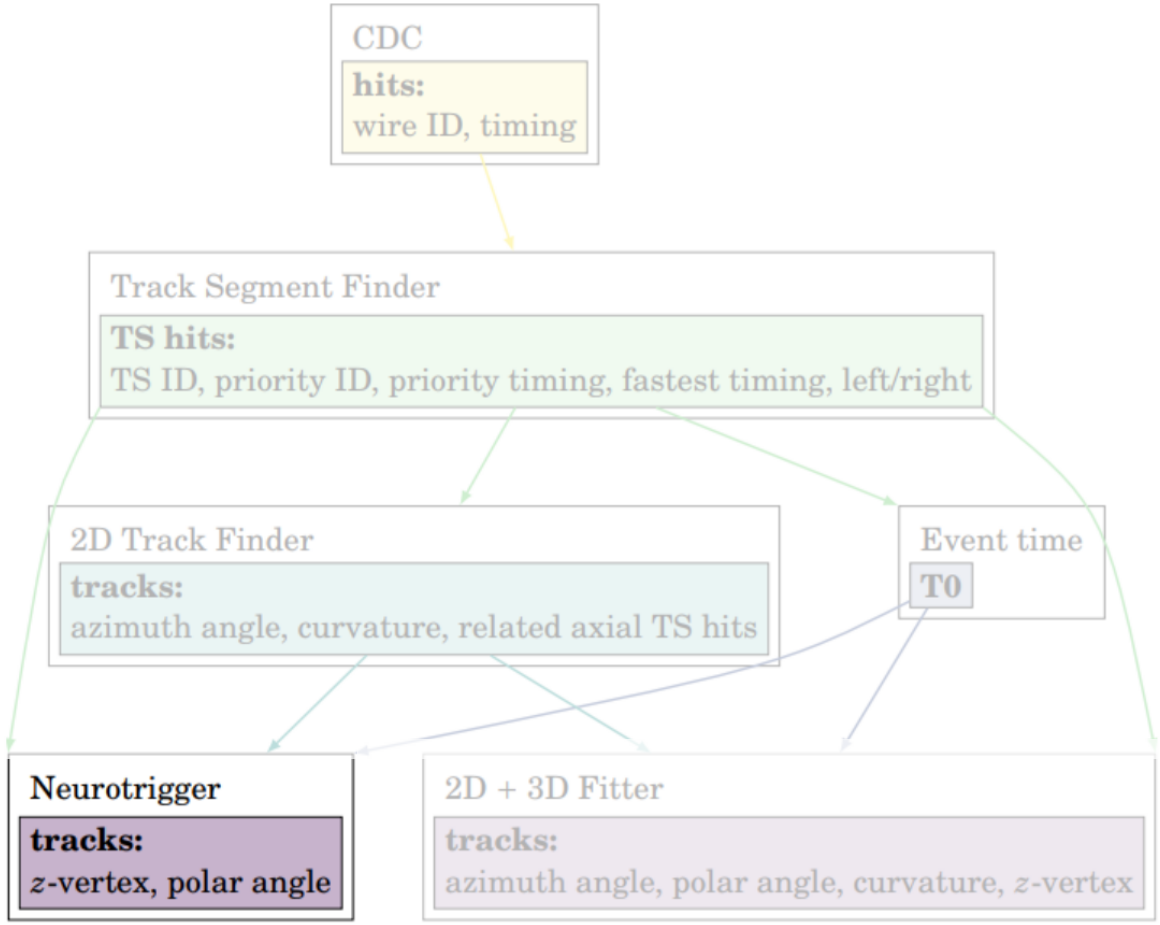
※ # of sampled events = 10000
※ HLT filtering is OFF in both runs.

CDC TRG rate contribute large part of total L1 trigger

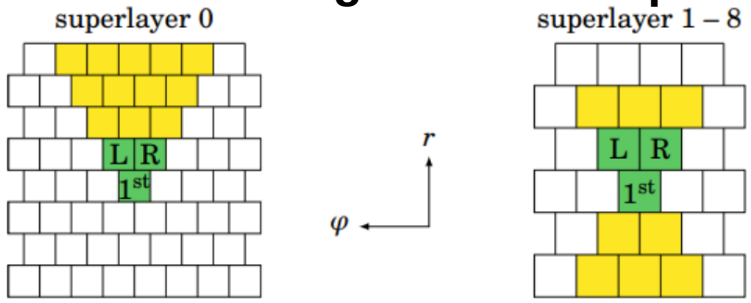
We want to reduce it by **50%**

Main idea for Neural Network: Improve vertex z resolution and rejecting background with large z

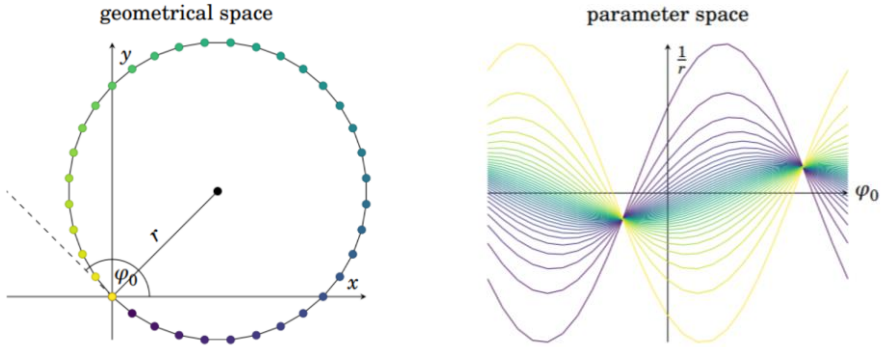
Introduction-CDC first level TRG



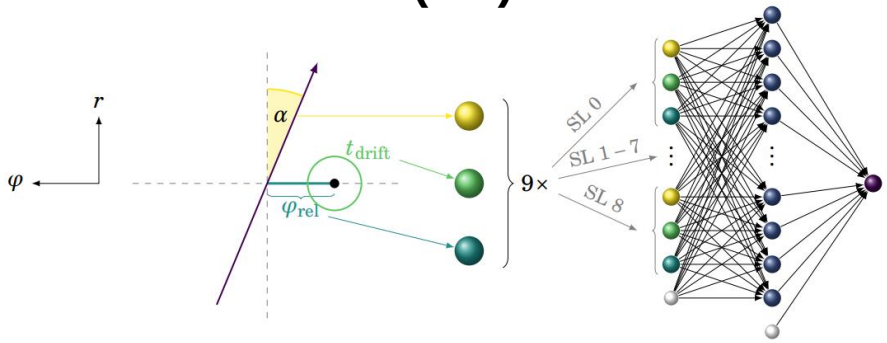
Track Segment build up



2D Track reconstructed (using hough transformation



3D Neural Network (NN) to calculated z / θ



New Training process with pytorch –A single epoch

input

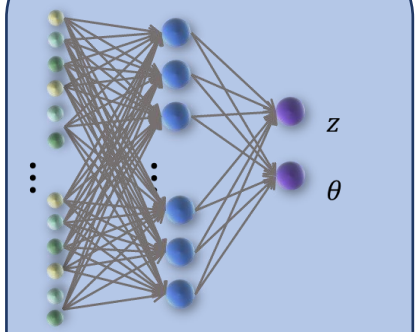
Input data (drift time, ϕ and α)

Input target (z and θ)

Validation process

Validation data → Validation Loss → If no decreasing ~n epoch → Stop training

Input batches (a subset of data)



Neuro Network

Predicted outputs (z and θ)

Loss

$J_t = \Sigma\left(z^i_{predict} - z^i_{target}\right)^2/N$

(Mean square loss function)

gradient

$g_t(w_i) = \nabla J(w_i)$

optimization

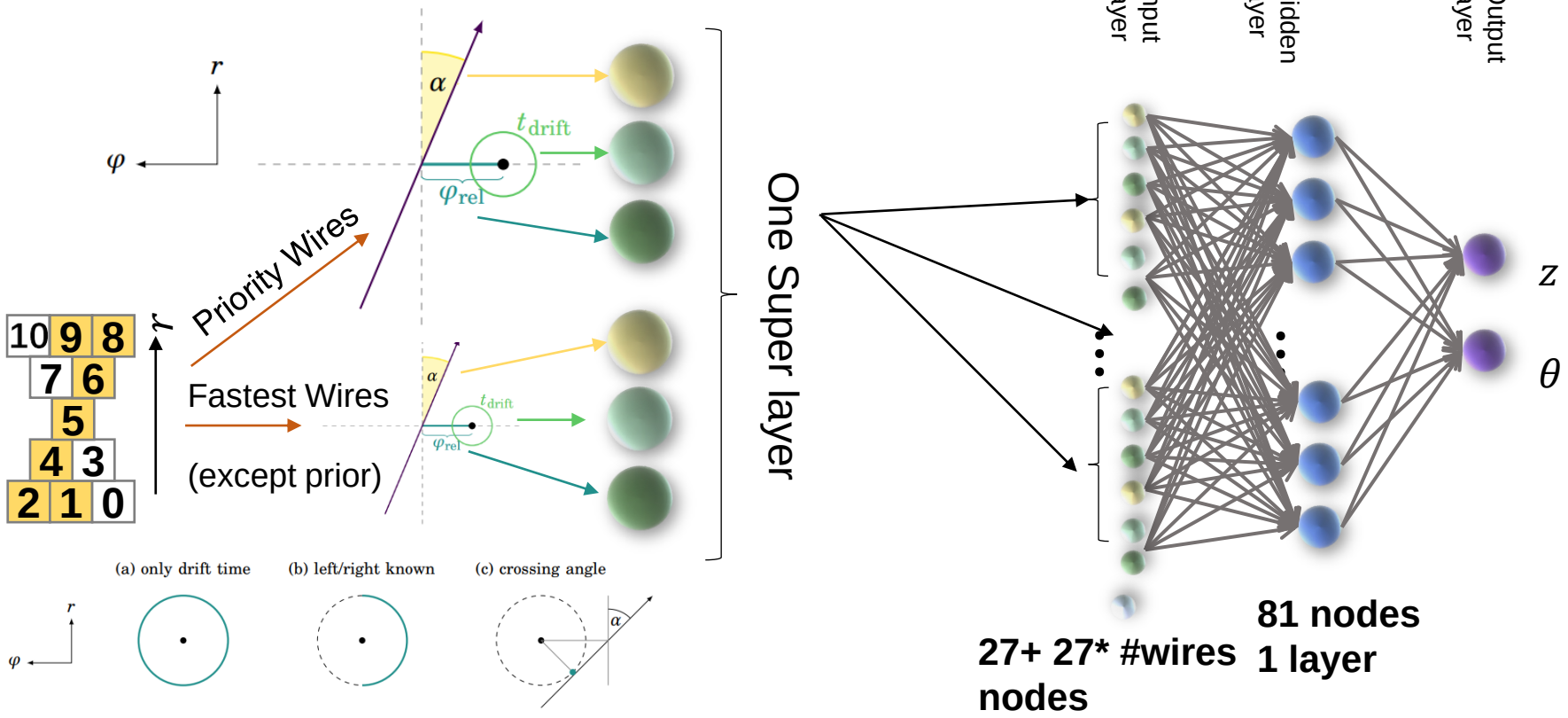
$w_i = w_i - \eta Optim(g_t)$

η: $learning\ rate$

Feed back to NN

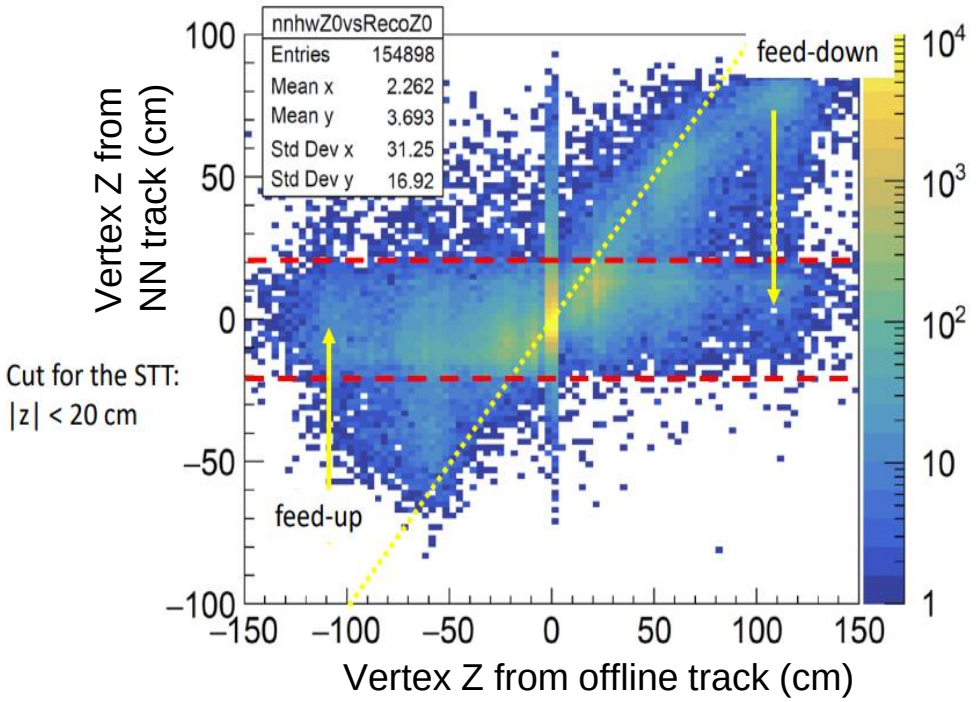
Optim = Adam for our NN

First attempt: Use extra wire(s) with full information

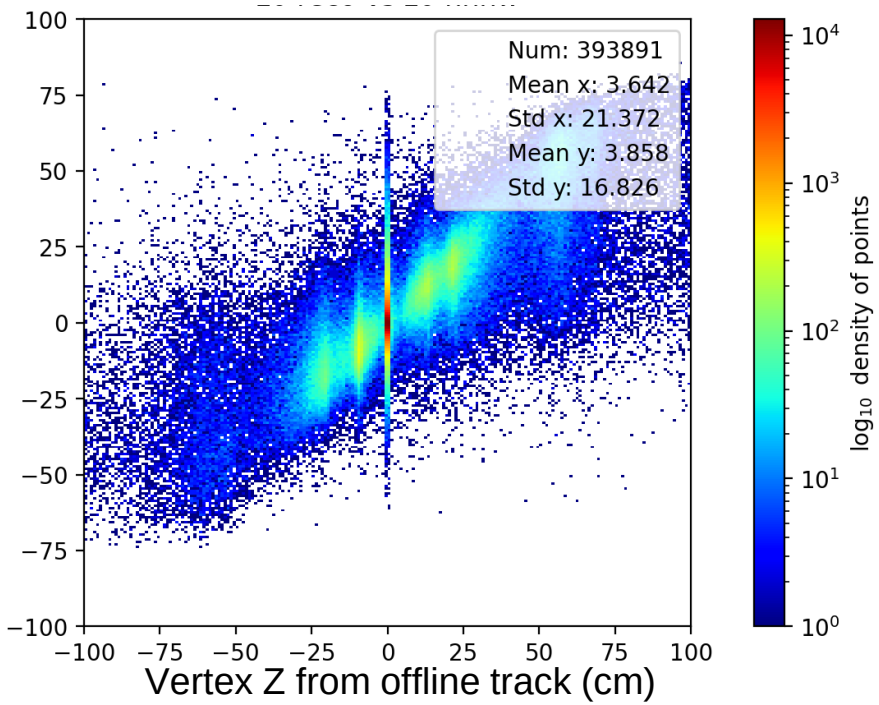


Choose the 1(2,3) wire(s) w*/* **L/R know first** and fastest t_{drift} as extra input of NN (we call such model “n Extra wires”)

New NN TRG performance in simulation



Currently hardware NN TRG



Re-Trained NN with 1 extra wire
& new training method

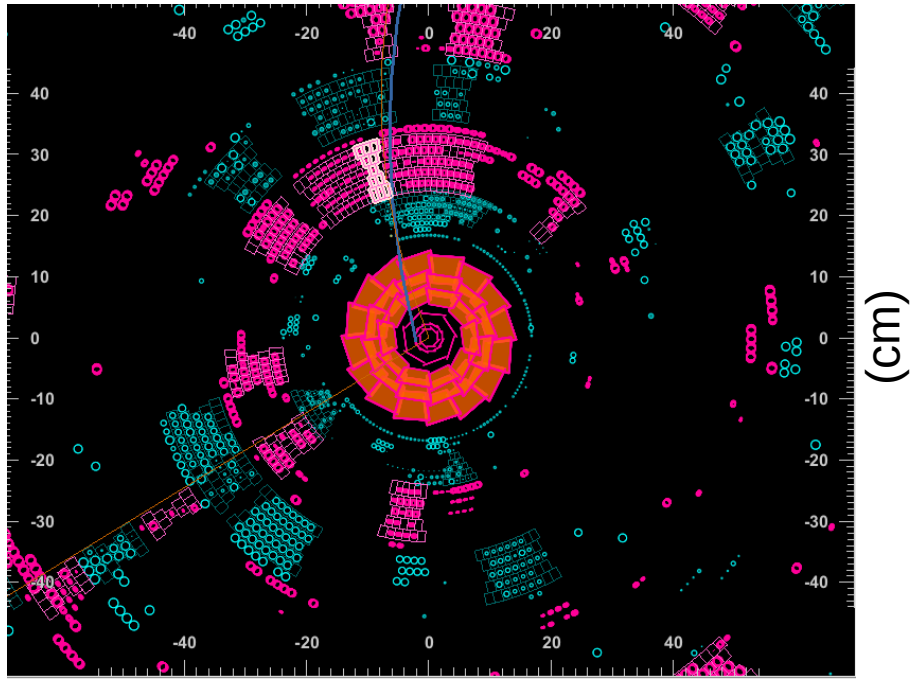
Feed down and feed up almost disappeared, resolution improved by factor 2

Problem encountered

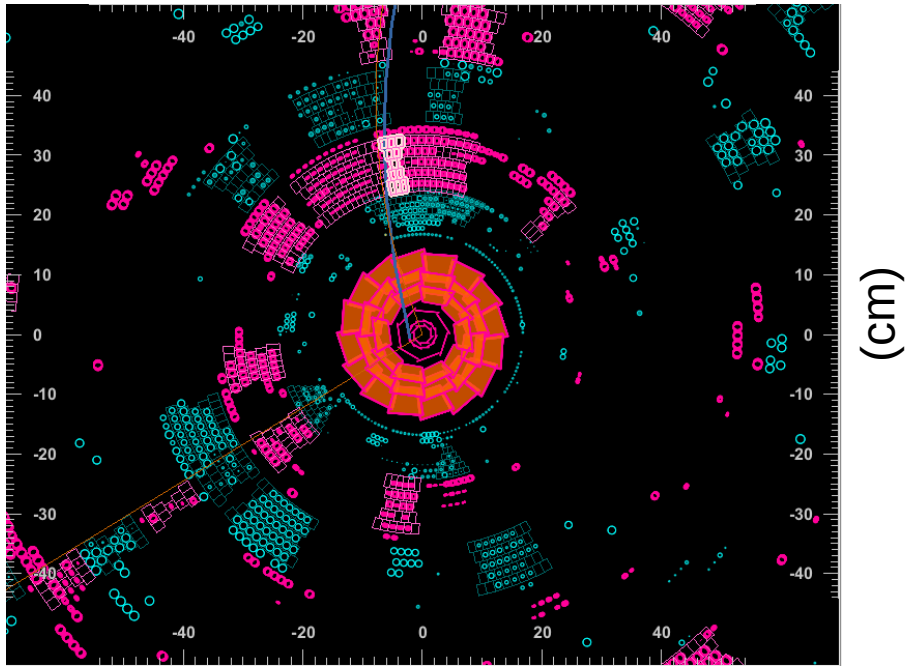
1. Track segment (TS) selection

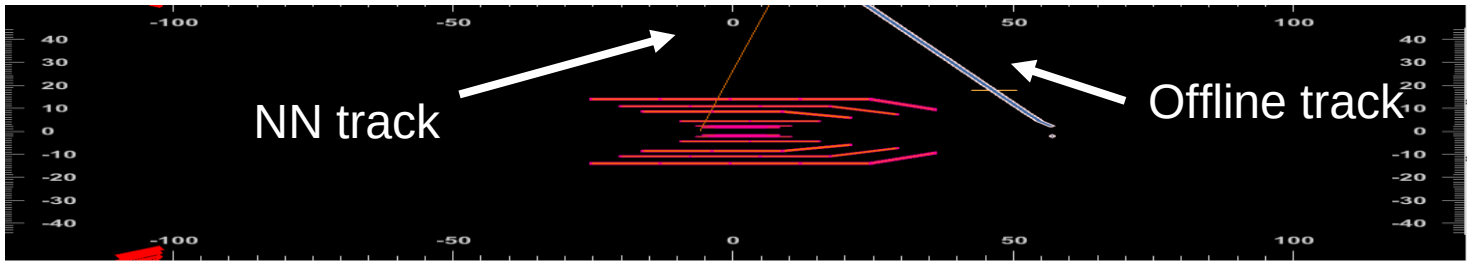
Exp26 run1780 event 22631185

Track Segment choosing by NN



Track Segment using in offline track

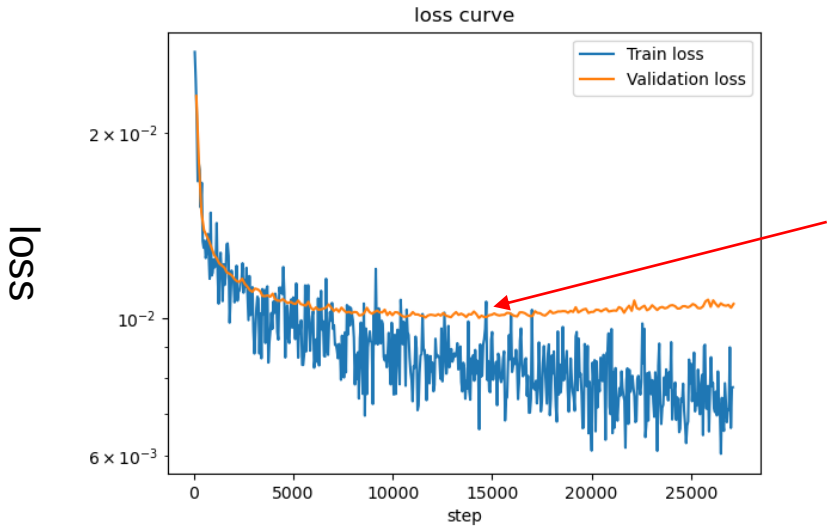




Large background caused trouble for TS choosing

Problem encountered

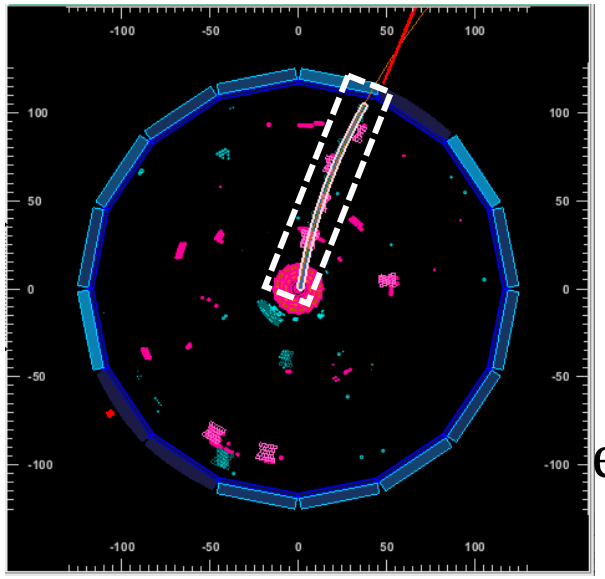
2. Validation loss didn't decrease further with more training (some time even increasing → overfitting)



*loss calculated the average difference between NN output and target value

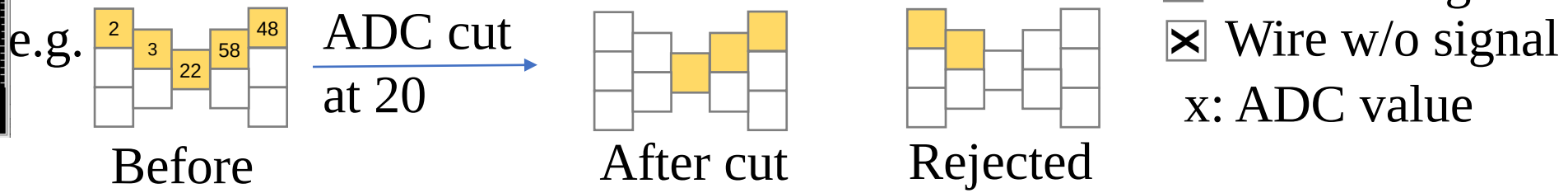
Best validation point

Generate Track Segment ADC pattern & LUT

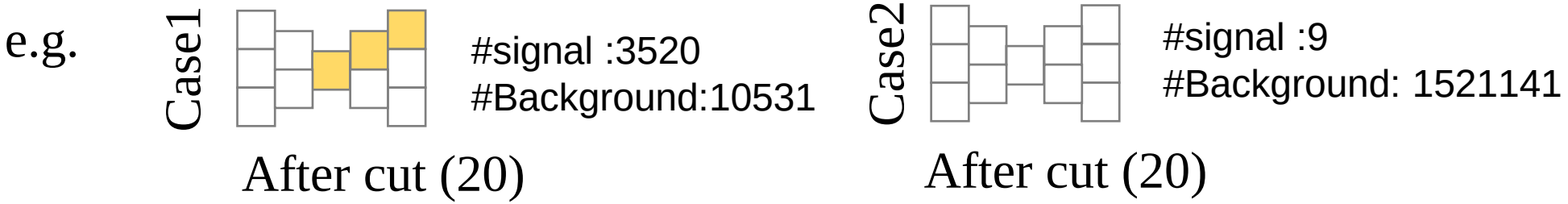


1. Tagging Track Segment related to an offline track as a “signal” other Track Segment as “background”

2. Apply ADC cut for every wires in TS; Get the new pattern after cut and rejected by cut



3. Count the "signal" and background” track segment corresponding to each pattern over exp26run1756-1780

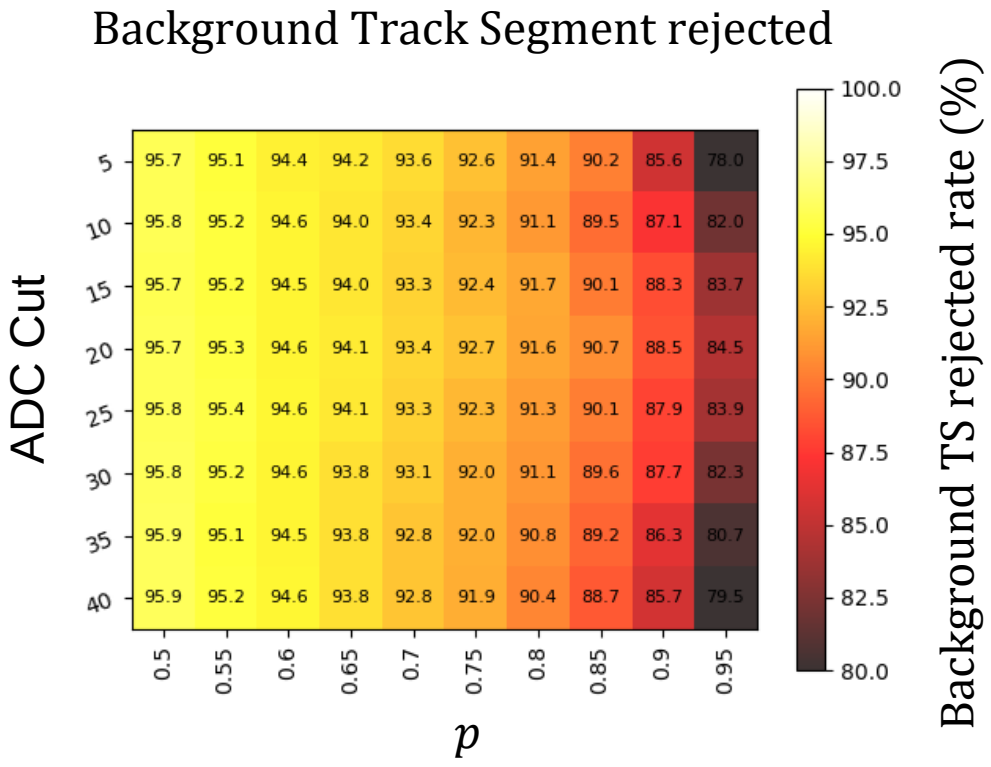


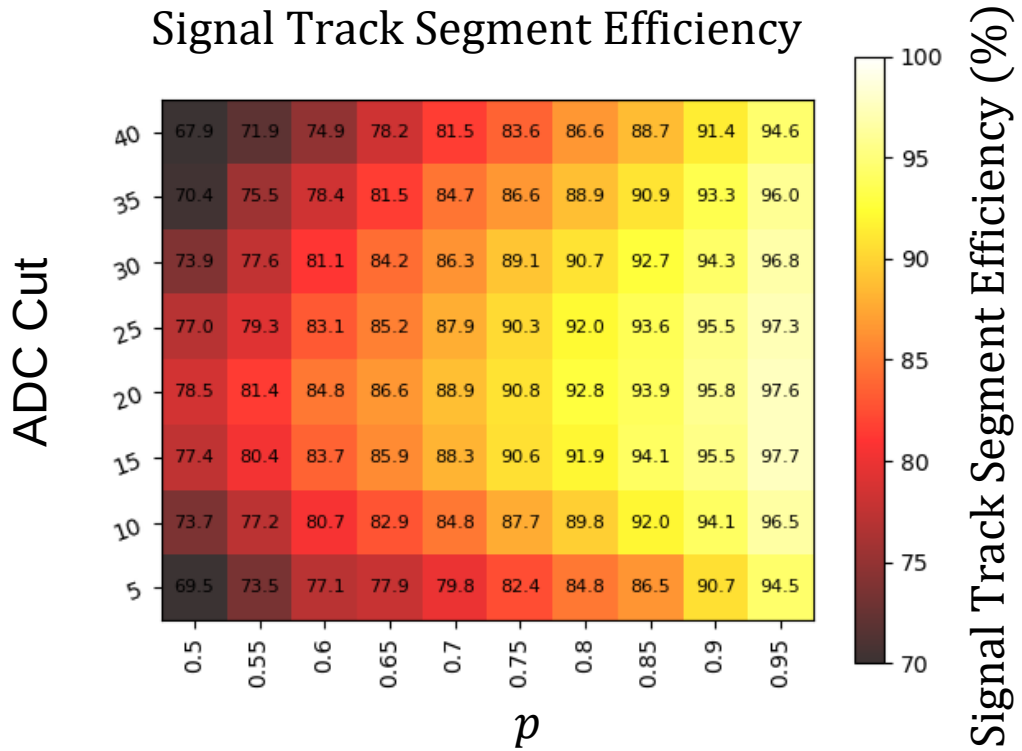
4. $TrackSegment = \begin{cases} backgound & if\ \#Bkg > p(\#Bkg + \#signal) + 3\sigma \\ undeterminded & otherwise \end{cases}$

If p = 0.9
Case 1: undeterminded
Case 2: Background

$\sigma = \sqrt{(\#Bkg + \#signal)p(1-P)}$ is the width of a binomial distribution with probability p.

Track Segment ADC pattern –Cut and P choose





To keep best efficiency, we select:

ADC cut = 15 ; p = 0.95

For TS selection in NN, Try two different strategies:

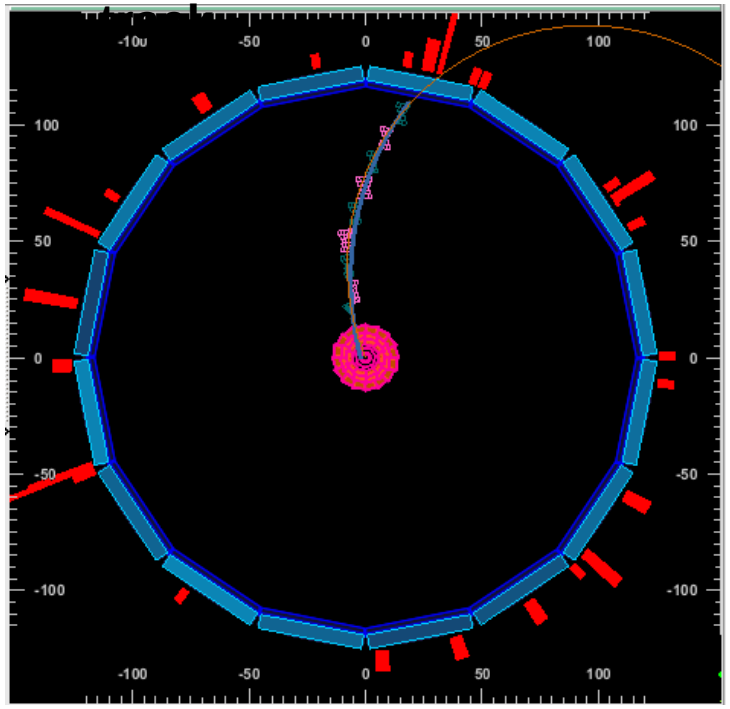
(1) Rejected: Only using TS not marked as background

(2) Not Rejected: Giving highest priority to “not background” tag but do not reject those with “background” tag

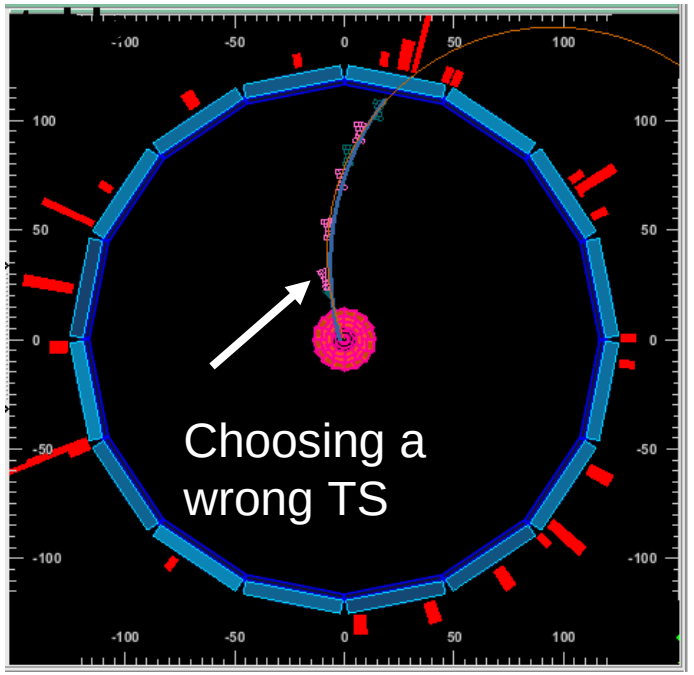
Some example with ADC LUT table

Exp26 run1780 event 22631185

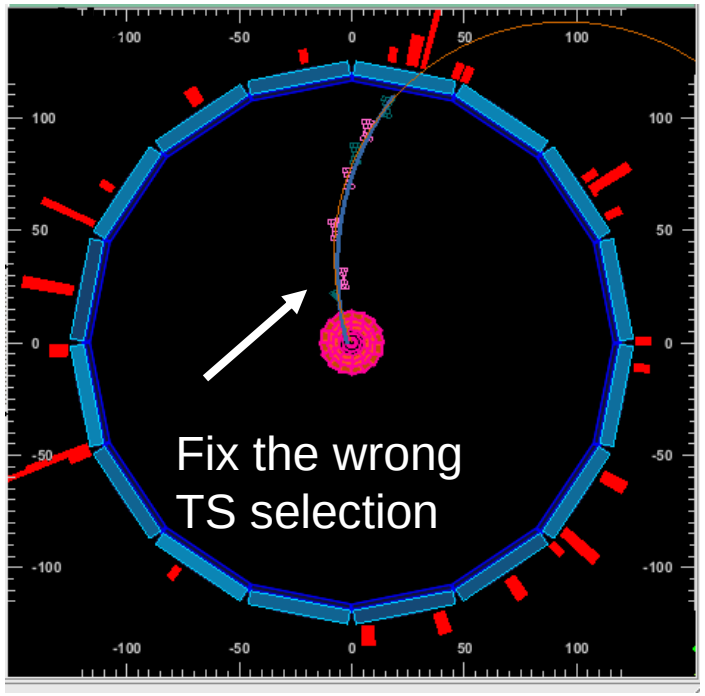
TS using in offline

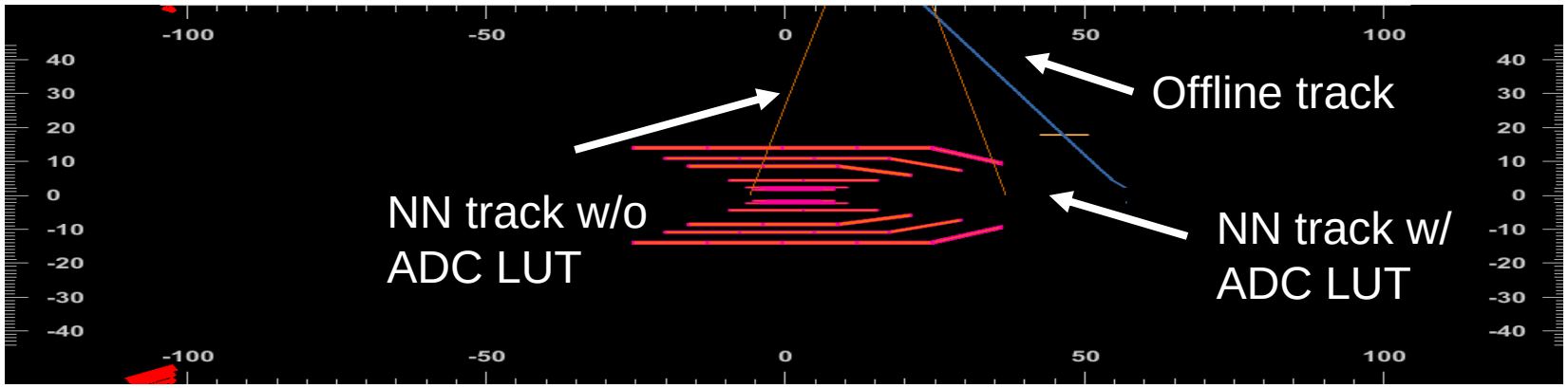


TS choosing w/o ADC LUT

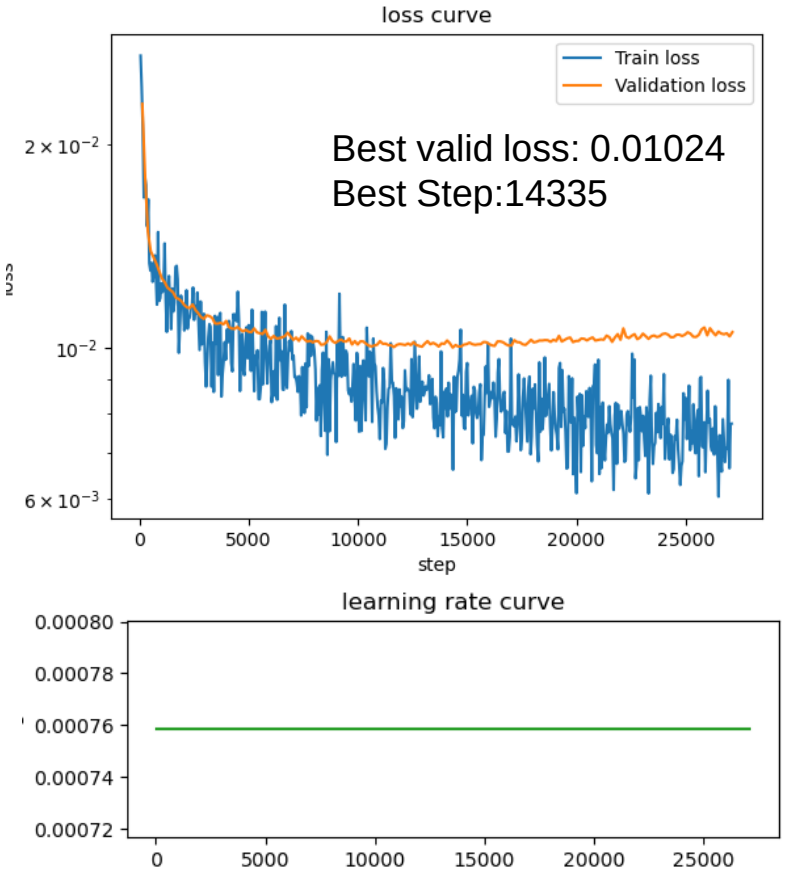


TS choosing w/ ADC LUT

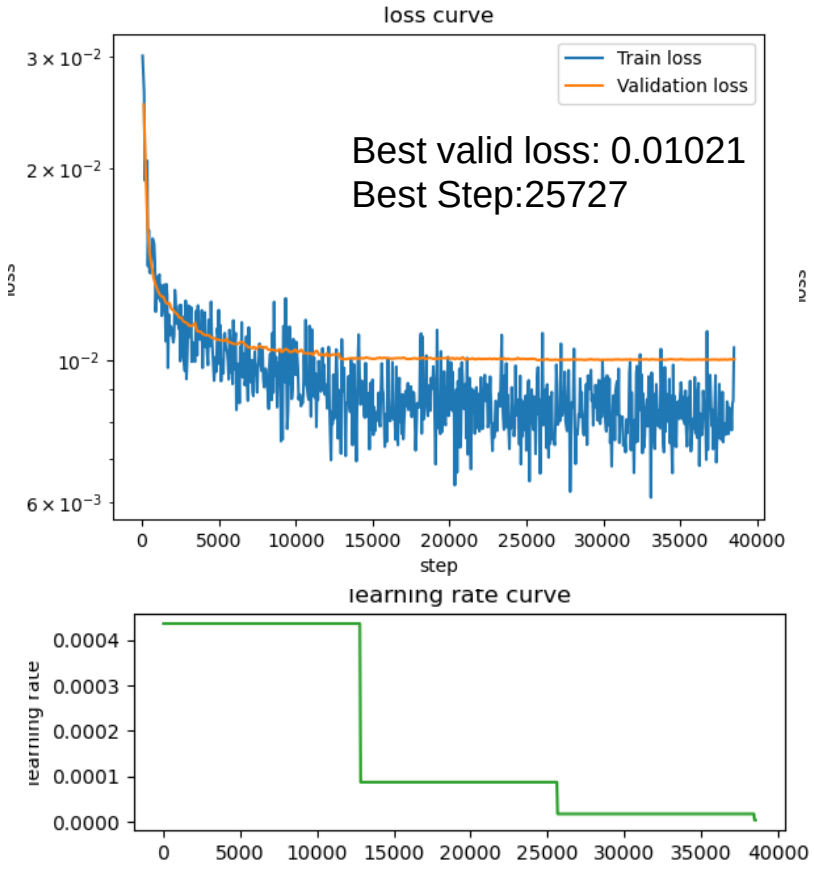




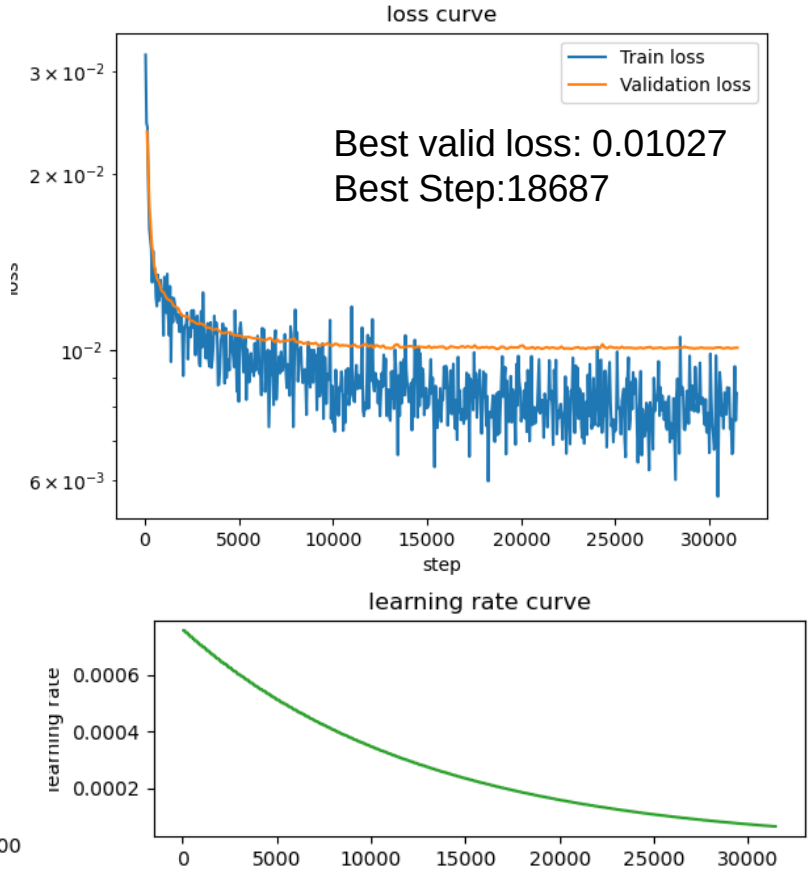
Learning rate adjustment



Constant learning rate



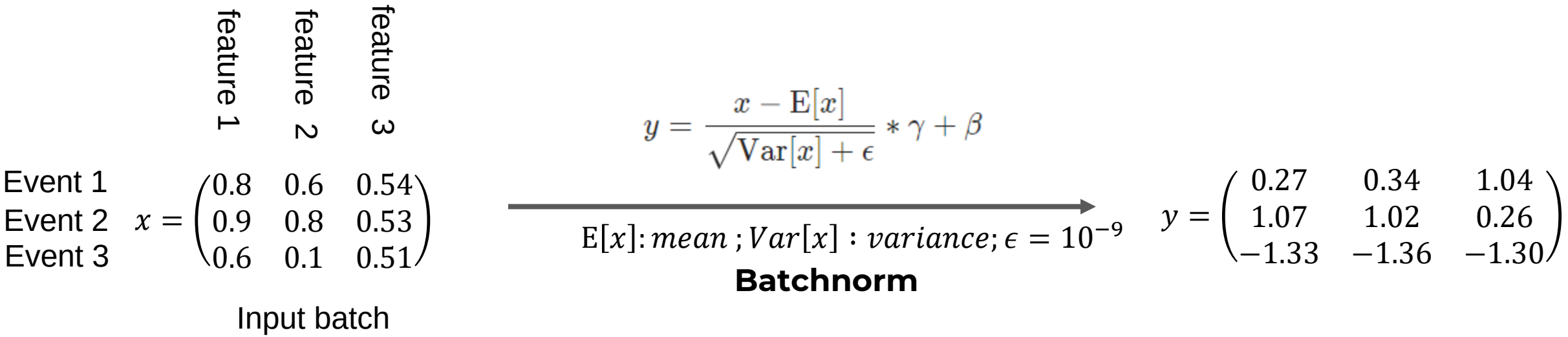
scheduled learning rate : Decrease lr when we find no improvement in validation loss after 100 epoch



Exponentially learning rate

Using scheduled learning rate for further training

Batchnorm –To speed up training

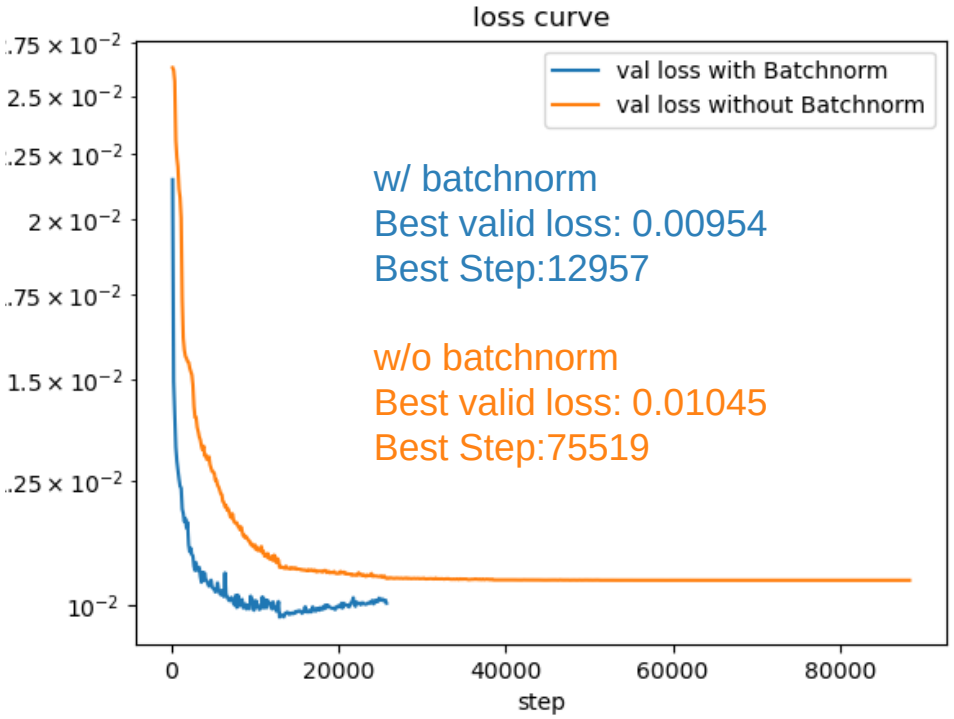


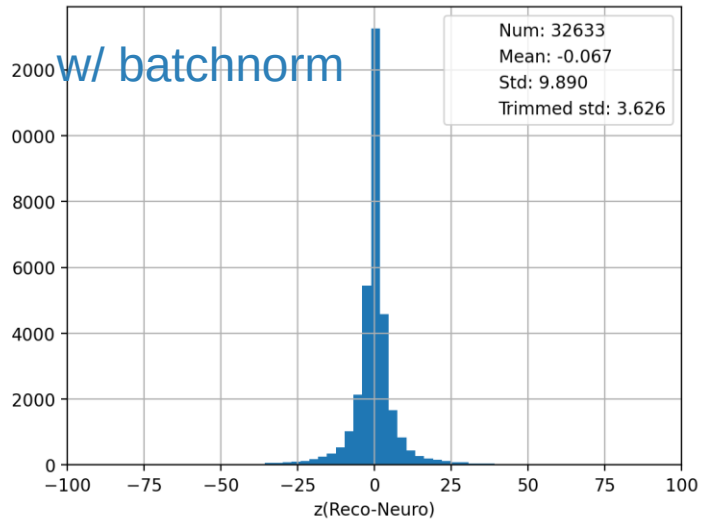
$*\ \mathrm{E}[x]\ and\ Var[x]\ will\ calculated\ over\ every\ batch\ with\ certain\ momnetum$

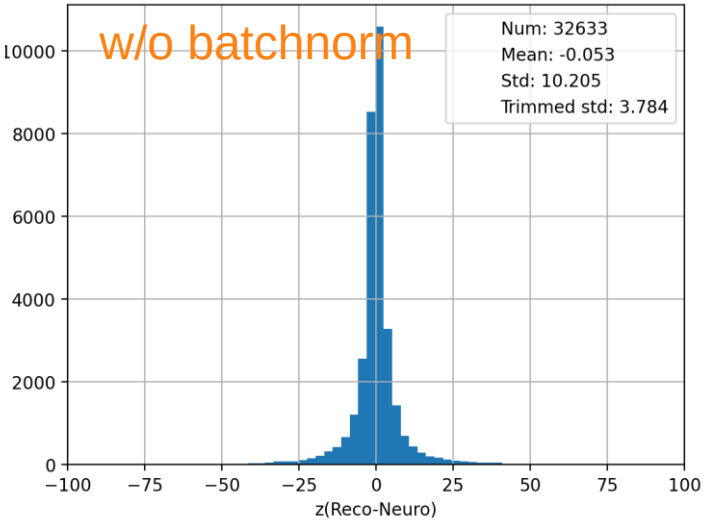
- **Using Batchnorm after every linear layer**
- **Keeping origin feature of each parameter while enlarge the gradient to speed up training**
- $E[x]\ and\ Var[x]$ **in training will be saved for evaluation**

Batchnorm –To speed up training

Validation loss for 3 hidden layers NN training



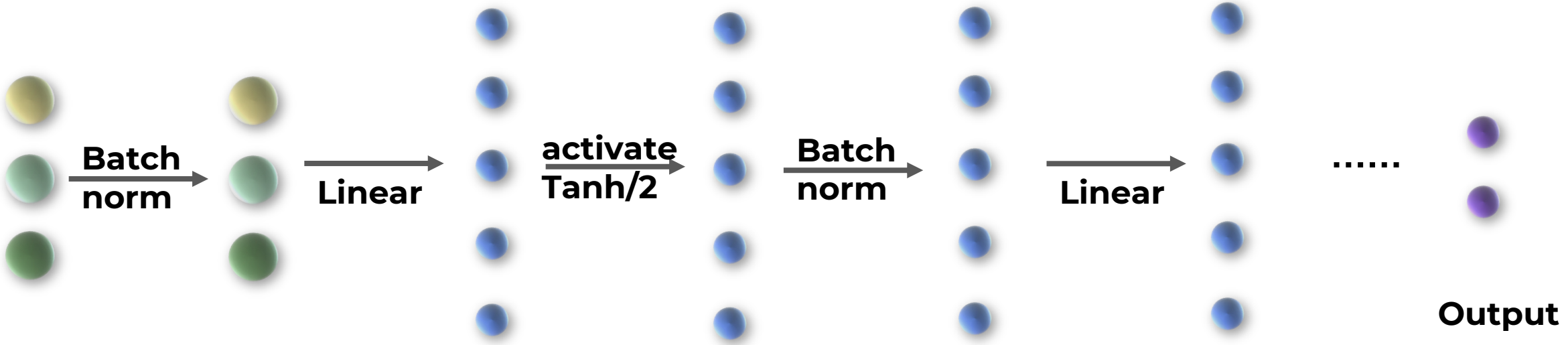




Batchnorm makes training faster and smaller final loss

Training time reduced to ¼ and validation loss reduced by 10%

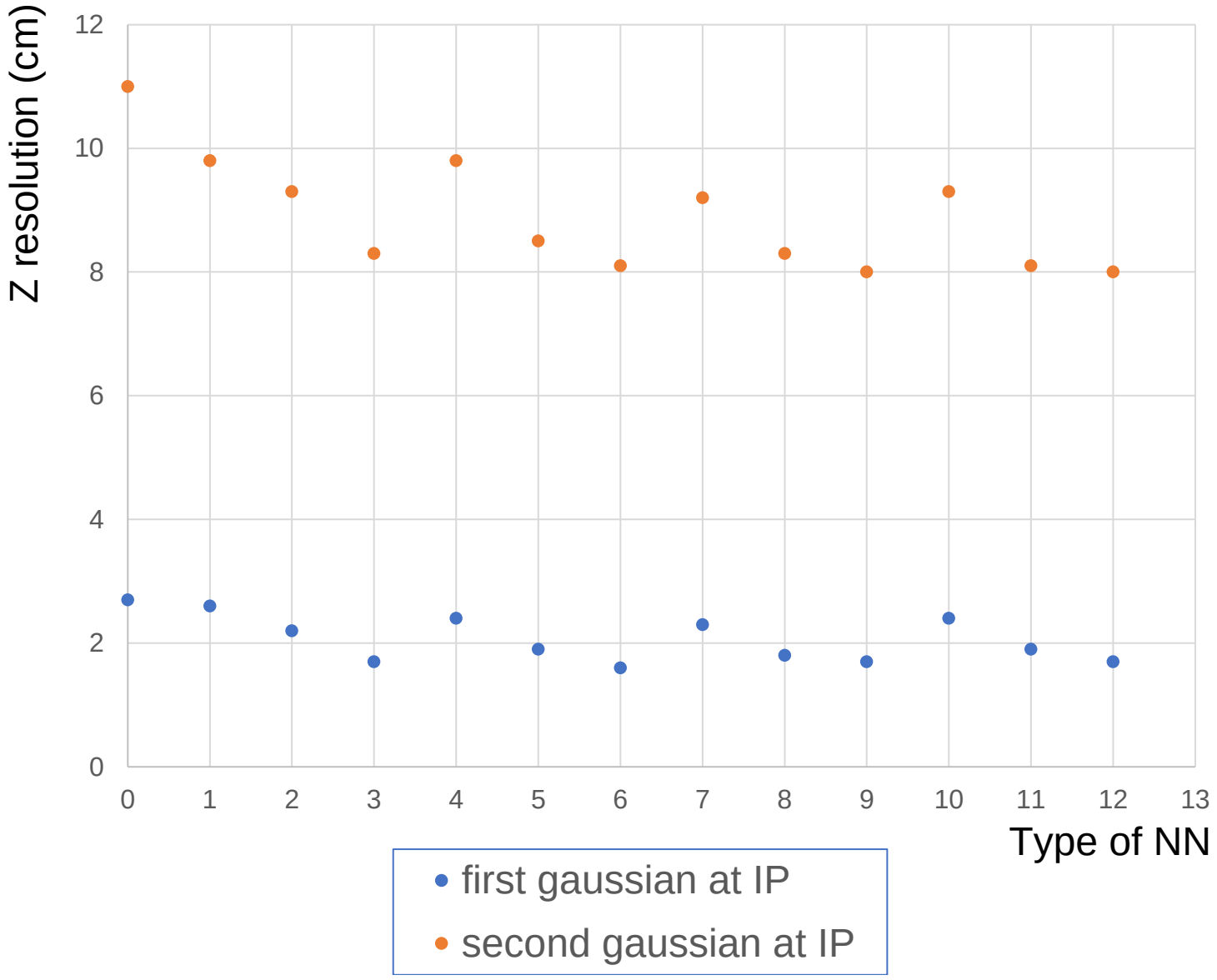
More hidden layer with Batchnorm



Try to add hidden layer with same nodes

Using Batchnorm after every linear layer

Training result – z resolution at IP

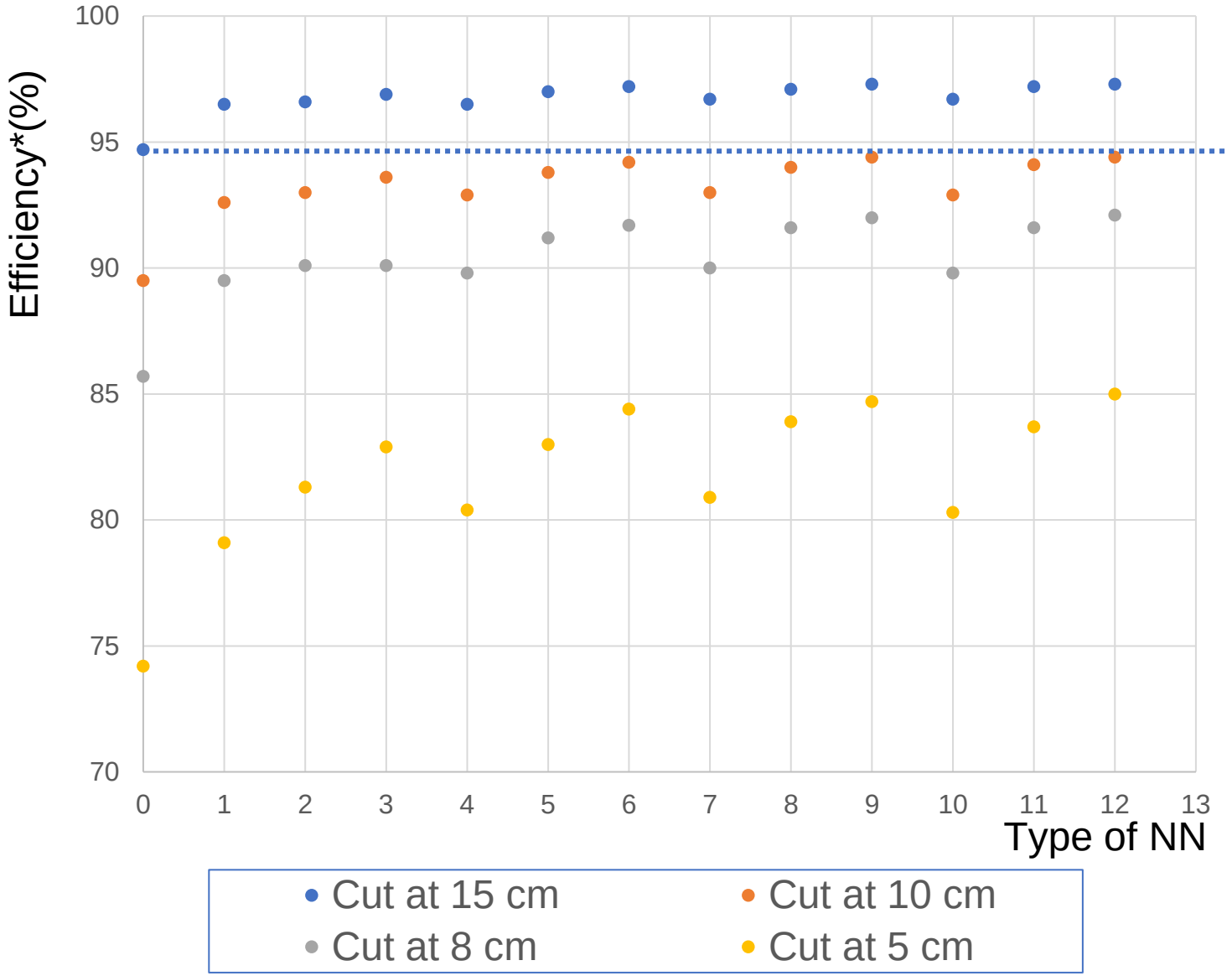


Type of NN:

0. Standard
1. Standard +batchnorm
2. Standard +batchnorm+2 hidden layers
3. Standard +batchnorm+3 hidden layers
4. 1 Extra Wire +batchnorm
5. 1 Extra Wire +batchnorm + 2 hidden layers
6. 1 Extra Wire +batchnorm + 3 hidden layers
7. 2 Extra Wires +batchnorm
8. 2 Extra Wires +batchnorm +2 hidden layers
9. 2 Extra Wires +batchnorm +3 hidden layers
10. 3 Extra Wires +batchnorm
11. 3 Extra Wires +batchnorm +2 hidden layers
12. 3 Extra Wires +batchnorm +3 hidden layers

IP Resolution highly depending on #hidden layers but almost same for varied #extra wires

Training result – Efficiency



Standard NN efficiency
at cut 15 cm

*Efficiency:

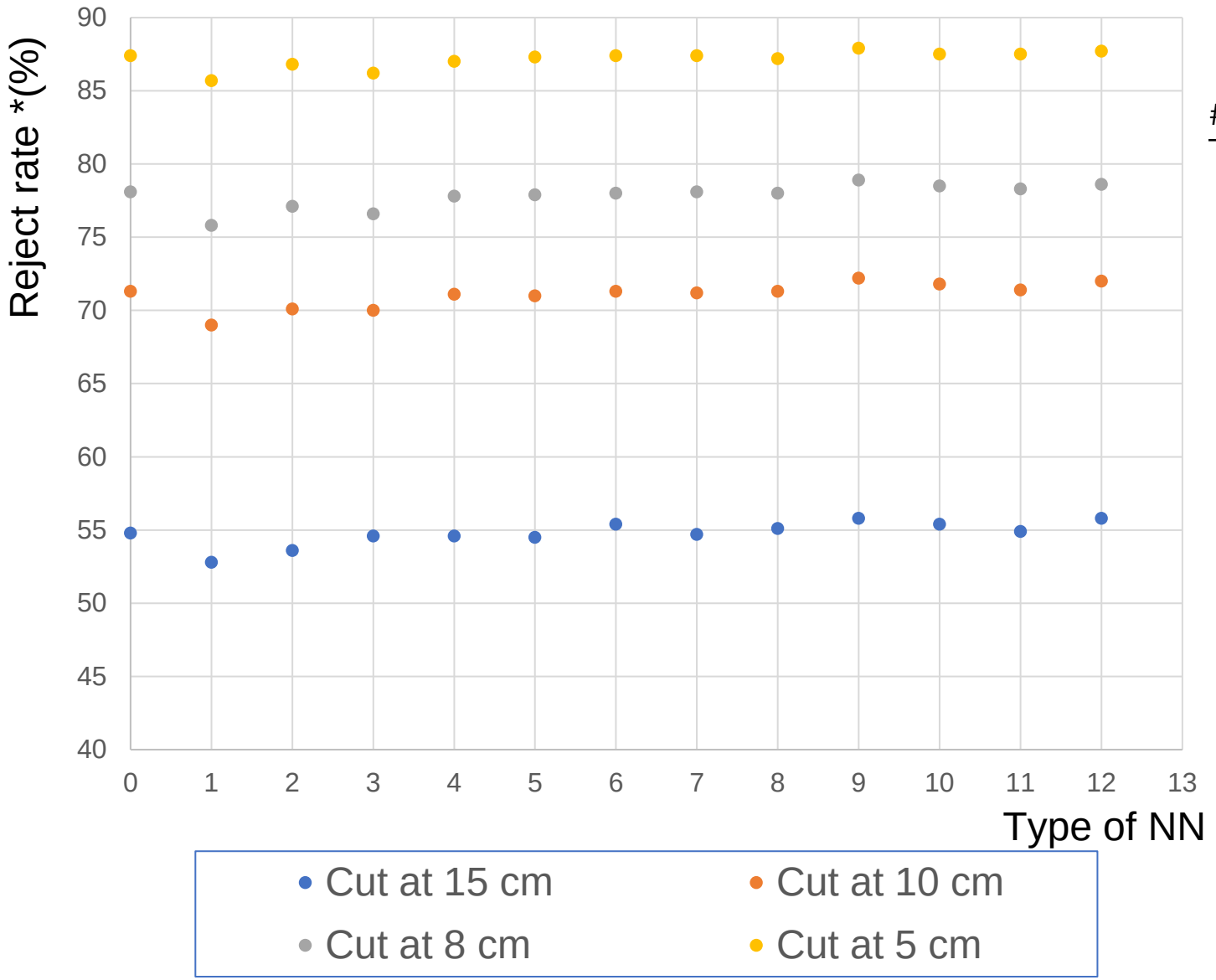
$$\frac{\#offline\ Tracks \text{ with } (|z_0| < 1\ \ AND\ \ |z_0| \text{from related NN } < \text{cut})}{\#offline\ Tracks\ with\ (\ |z_0|\ < 1 \text{ AND } \#\text{related NN } Track \geq 1)}$$

Only consider built NN track

Efficiency from NN selection and 2D Track Will be considered later

With more hidden layer and extra 1(2 or 3) wires, we could keep same efficiency with cut at 10 cm comparing with standard one with cut at 15cm

Training result – Background reject rate



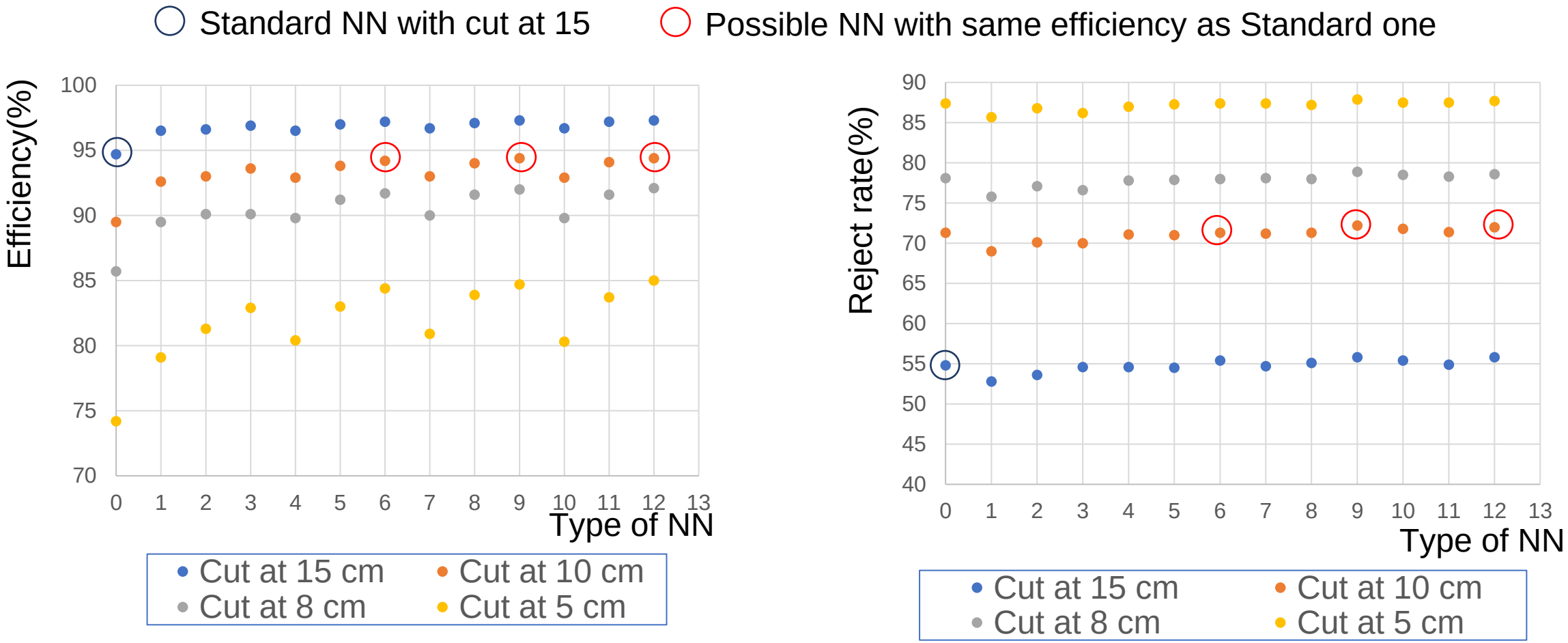
*Reject rate:

$$\frac{\#offline\ Tracks \text{ with } (|z_0| > 1\ \ AND\ \ |z_0| \text{from related NN} > \text{cut})}{\#offline\ Tracks\ with\ (\ |z_0| > 1 \text{ AND } \#\text{related NN } Track \geq 1)}$$

Only consider built NN track also

Rejected highly depending on cut

Training result –Efficiency and Background reject rate



With 1(2 or 3) Extra Wire(s) +3 hidden layers, we could keep same efficiency while tightening cut to 10 cm

Reject rate: $55\% \rightarrow 72\%$ reducing ~40% of background from large z

Summary & Plan

Summary

a) With 3 hidden layers NN, we could tighten cut to 10 and reduce ~40% background from large z

b) ADC pattern LUT could help build better NN track

Plan

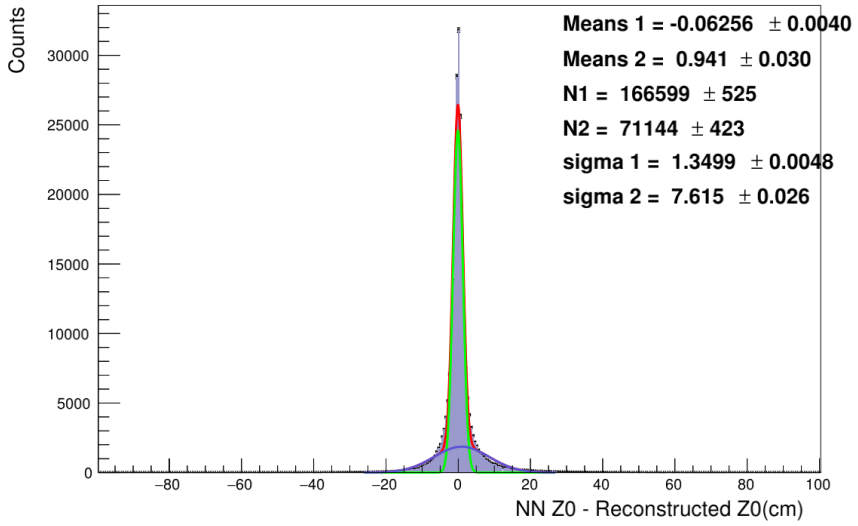
a) Apply ADC pattern LUT with “Reject” mode to see if fake track rate could be reduced

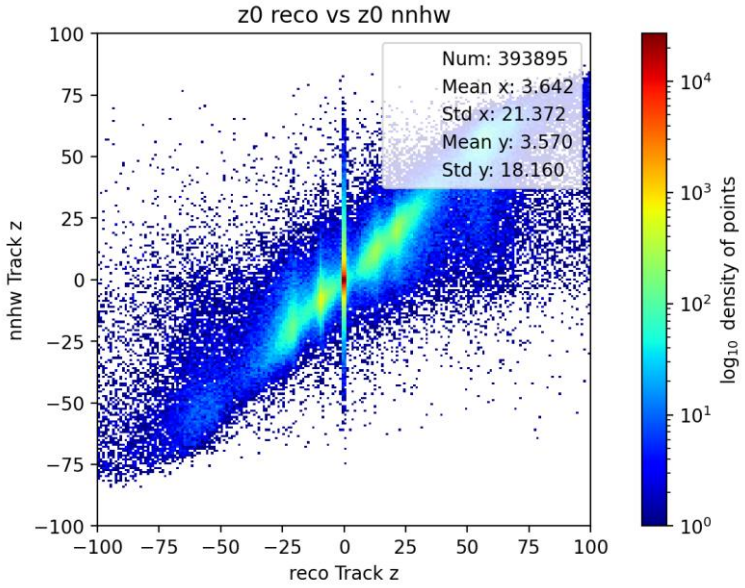
b) Try “group convolution” for the extra wires as input.

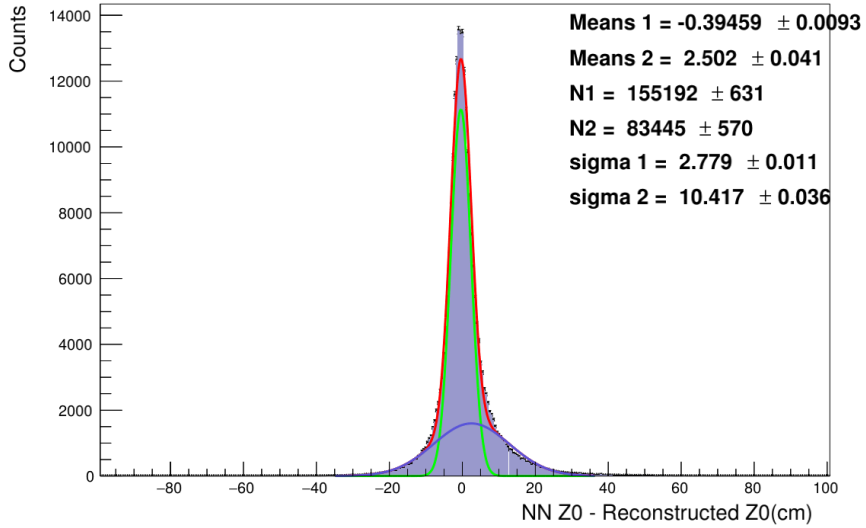
c) Try different hidden nodes

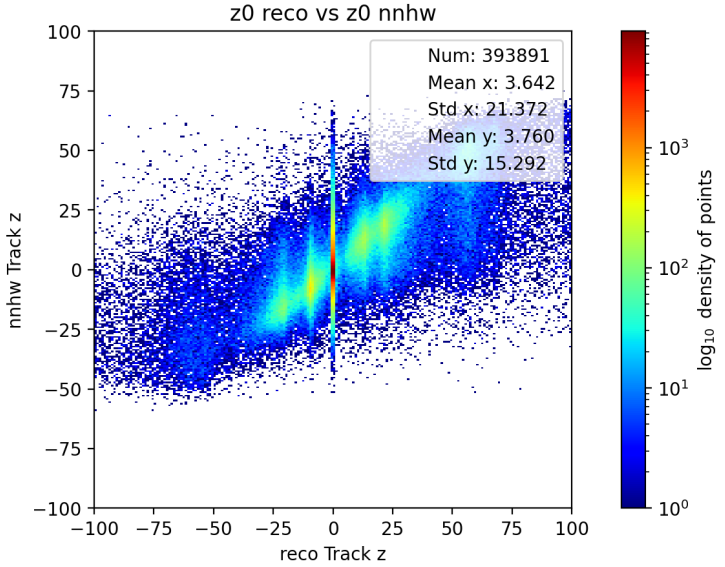
END

Best result


Counts
Means 1 = -0.06256 ± 0.0040
Means 2 = 0.941 ± 0.030
N1 = 166599 ± 525
N2 = 71144 ± 423
sigma 1 = 1.3499 ± 0.0048
sigma 2 = 7.615 ± 0.026
NN Z0 - Reconstructed Z0(cm)


z0 reco vs z0 nnhw
Num: 393895
Mean x: 3.642
Std x: 21.372
Mean y: 3.570
Std y: 18.160
nnhw Track z
reco Track z
log10 density of points


Counts
Means 1 = -0.39459 ± 0.0093
Means 2 = 2.502 ± 0.041
N1 = 155192 ± 631
N2 = 83445 ± 570
sigma 1 = 2.779 ± 0.011
sigma 2 = 10.417 ± 0.036
NN Z0 - Reconstructed Z0(cm)

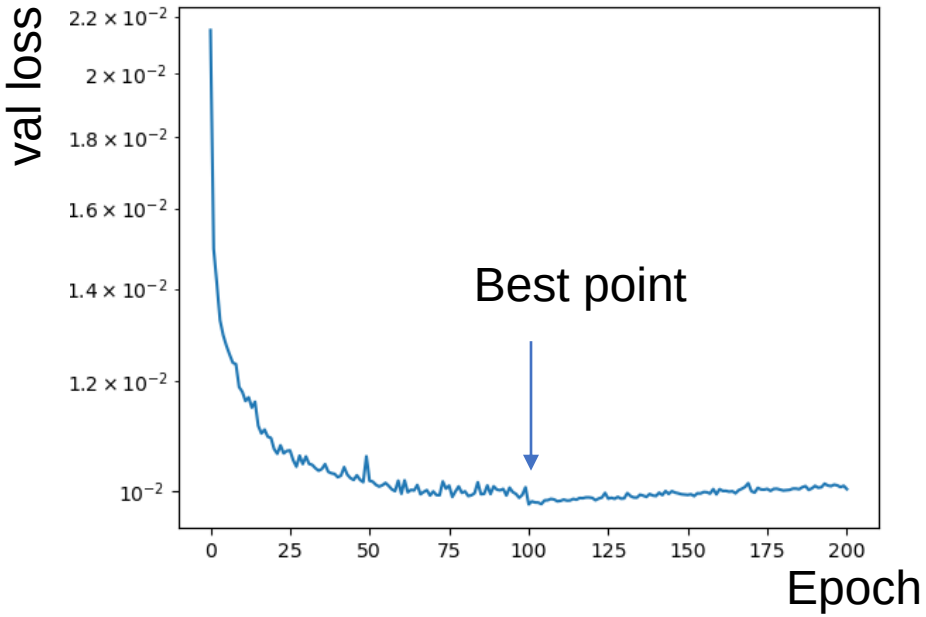

z0 reco vs z0 nnhw
Num: 393891
Mean x: 3.642
Std x: 21.372
Mean y: 3.760
Std y: 15.292
nnhw Track z
reco Track z
log10 density of points

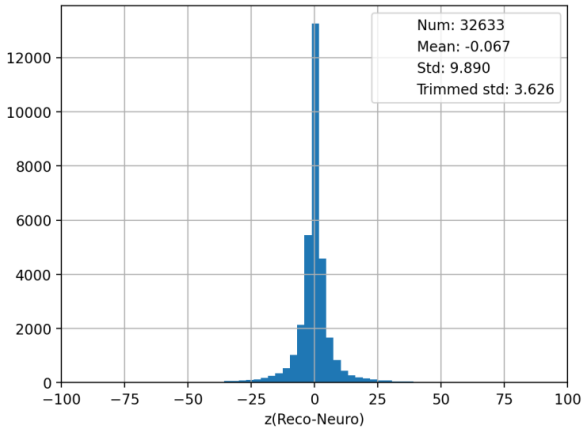
What if we not use batch norm in ExtraWire case?

For 3 Extra Wires +3 hidden layers case:

Using Batch norm:

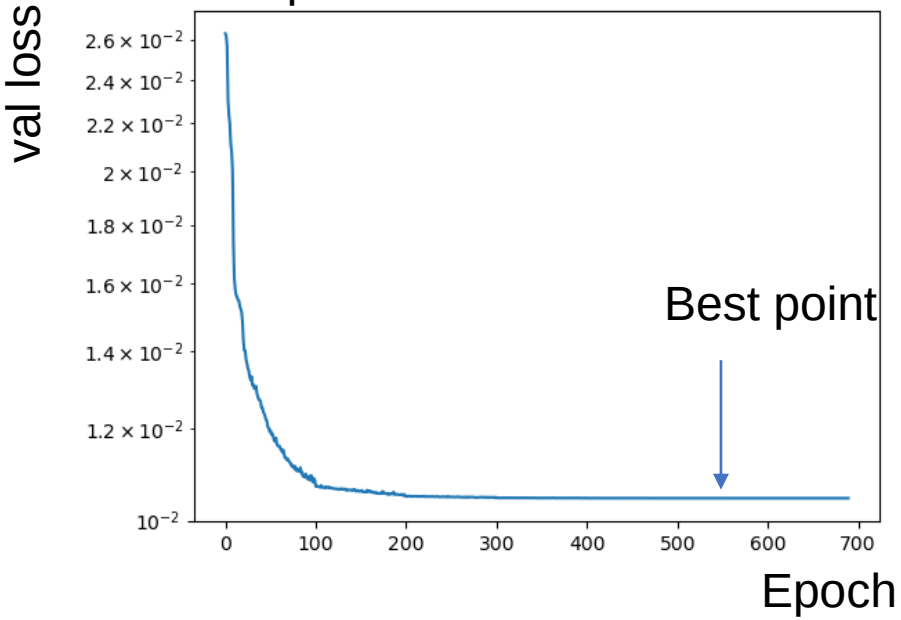
Final epoch: 100

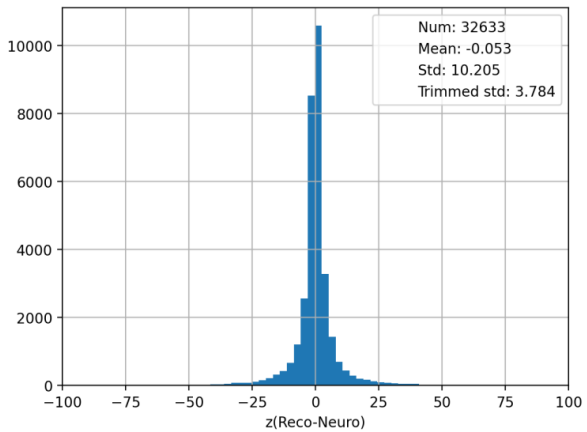




Not using Batch norm:

Final epoch: 589





Training result

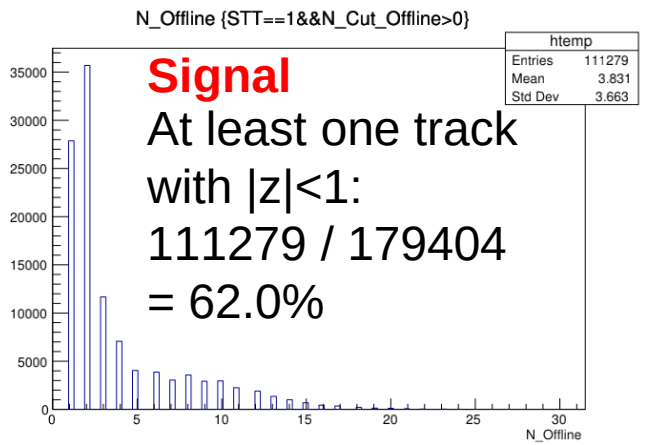
Training based on ADC cut sample – Not rejected

NN Type	Std (cm)	σ at IP (cm)	Efficiency (cut at 15/ 10/ 8 / 5) (%)	Reject rate (cut at 15/ 10/ 8 / 5) (%)
Standard	13.2	2.7|11.0	94.7/ 89.5/ 85.7/ 74.2	54.8/ 71.3/ 78.1/ 87.4
Standard +batchnorm	12.3	2.6 | 9.8	96.5/ 92.6/ 89.5/ 79.1	52.8/ 69.0/ 75.8/ 85.7
Standard +batchnorm+2 hidden layers	11.8	2.2 | 9.3	96.6/ 93.0/ 90.1/ 81.3	53.6/ 70.1/ 77.1/ 86.8
Standard +batchnorm+3 hidden layers	11.5	1.7 | 8.3	96.9/ 93.6/ 90.1/ 82.9	54.6/ 70.0/ 76.6/ 86.2
1 Extra Wire	12.0	2.4 | 9.8	96.5/ 92.9/ 89.8/ 80.4	54.6/ 71.1/ 77.8/ 87.0
1 Extra Wire + 2 hidden layers	11.3	1.9 | 8.5	97.0/ 93.8/ 91.2/ 83.0	54.5/ 71.0/ 77.9/ 87.3
1 Extra Wire + 3 hidden layers	11.1	1.6 | 8.1	97.2/ 94.2/ 91.7/ 84.4	55.4/ 71.3/ 78.0/ 87.4
2 Extra Wires	11.9	2.3 | 9.2	96.7/ 93.0/ 90.0/ 80.9	54.7/ 71.2/ 78.1/ 87.4
2 Extra Wires +2 hidden layers	11.2	1.8 | 8.3	97.1/ 94.0/ 91.6/ 83.9	55.1/ 71.3/ 78.0/ 87.2
2 Extra Wires +3 hidden layers	10.9	1.7 | 8.0	97.3/ 94.4/ 92.0/ 84.7	55.8/ 72.2/ 78.9/ 87.9
3 Extra Wires	11.9	2.4 | 9.3	96.7/ 92.9/ 89.8/ 80.3	55.4/ 71.8/ 78.5/ 87.5
3 Extra Wires +2 hidden layers	11.2	1.9 | 8.1	97.2/ 94.1/ 91.6/ 83.7	54.9/ 71.4/ 78.3/ 87.5
3 Extra Wires +3 hidden layers	10.7	1.7 | 8.0	97.3/ 94.4/ 92.1/ 85.0	55.8/ 72.0/ 78.6/ 87.7

Generate Track Segment ADC pattern & LUT

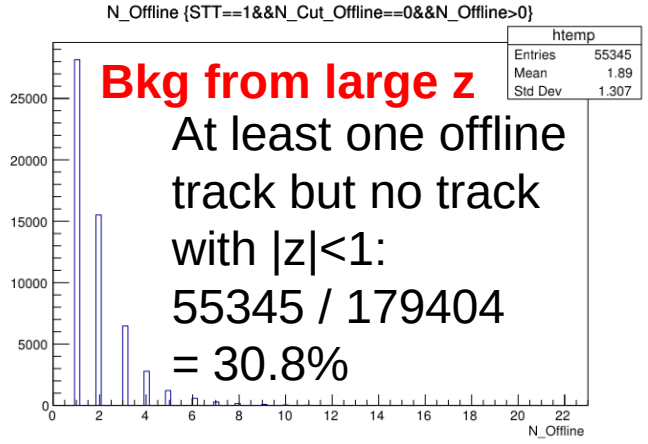
Using data in exp26run1756-1780 (56 58 61 66 69 72 73 74 77 80):

STT: Total 179404



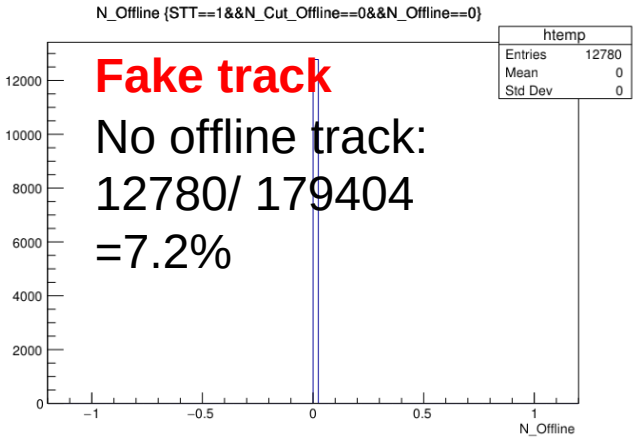
Signal

At least one track with |z|<1:
111279 / 179404
= 62.0%



Bkg from large z

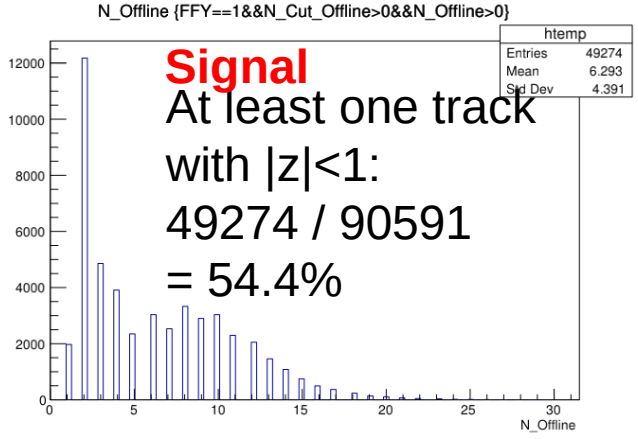
At least one offline track but no track with |z|<1:
55345 / 179404
= 30.8%



Fake track

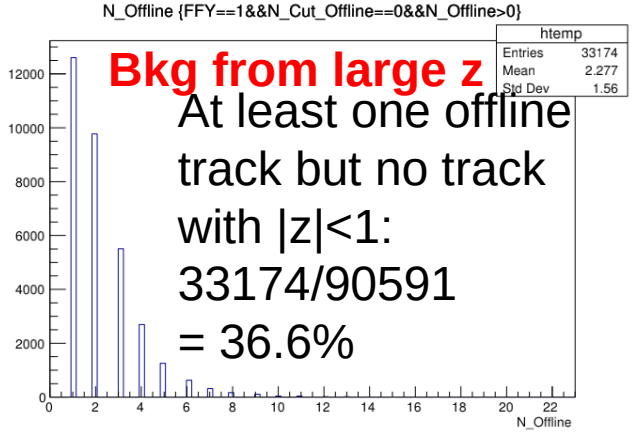
No offline track:
12780/ 179404
=7.2%

FFY: Total 90591



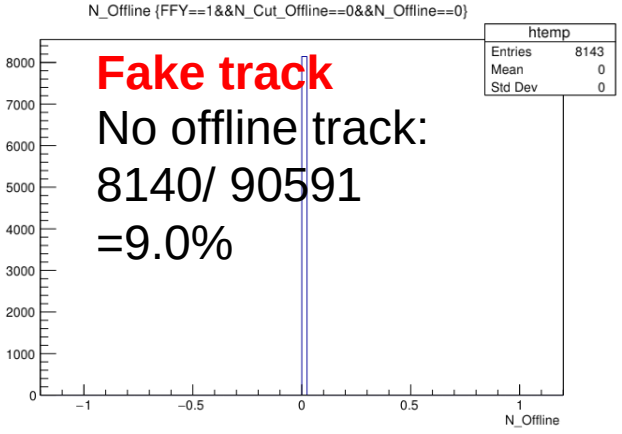
Signal

At least one track with |z|<1:
49274 / 90591
= 54.4%



Bkg from large z

At least one offline track but no track with |z|<1:
33174/90591
= 36.6%



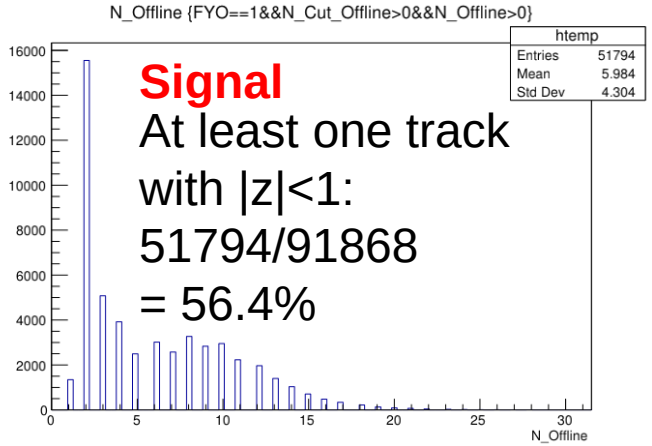
Fake track

No offline track:
8140/ 90591
=9.0%

Generate Track Segment ADC pattern & LUT

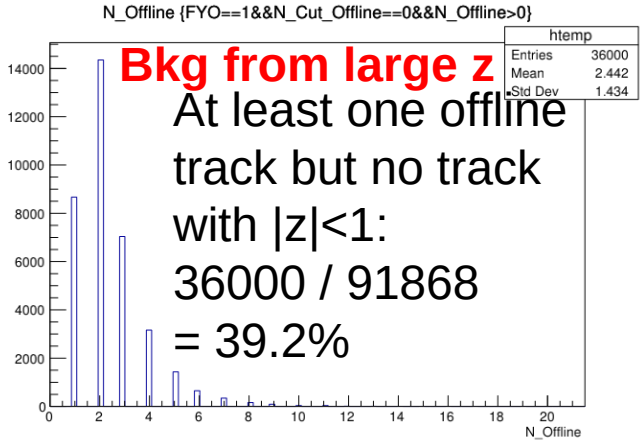
Using data in exp26run1756-1780 (56 58 61 66 69 72 73 74 77 80):

FYO: Total 91868



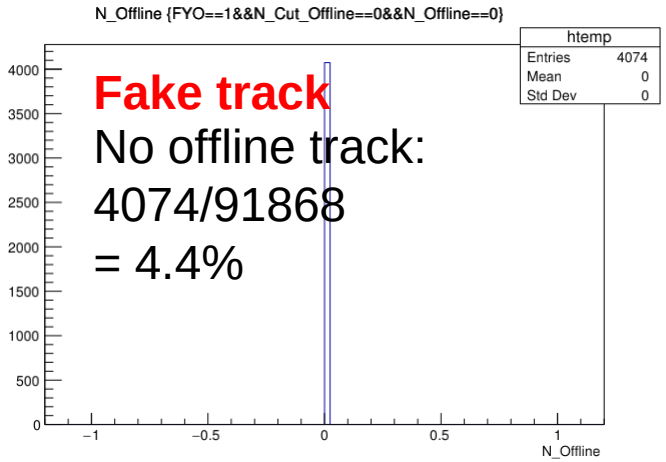
Signal

At least one track with |z|<1:
51794/91868
= 56.4%



Bkg from large z

At least one offline track but no track with |z|<1:
36000 / 91868
= 39.2%



Fake track

No offline track:
4074/91868
= 4.4%

Fake track rate is low comparing with Bkg from large z in this data set

Motivation

Trigger menu and rate @ 2022/6/9, exp26r1261

T.Koga-san

-Total L1 rate=~11.5kHz, Luminosity=~4.5 × 10^34

-Rate of standard bits (ffy+fyo+c4+hie) = 4.7kHz: need to keep until end of BelleII

-Others are 6.8kHz

event triggered by upper bits are excluded in lower bits in table

Category	Bit name and condition	Raw rate (kHz)	Exclusive rate (kHz)
CDC ~~B physics~~ standard bits	**ffy**: #full track>=3, |z|<20cm	2.18	2.18
	fyo: #full track>=2, Δϕ>90deg, |z|<20cm	1.77	0.73
ECL ~~B physics~~ standard bits	**c4**: #cluster>=4	0.47	0.26
	hie: Energy sum>1GeV	2.02	1.54
Subtotal		**4.7**	**4.7**
KLM τ/dark	**klmb2b, eklmb2b, beklm**: Back to back sector hits	0.51	0.46
	cdcklm, seklm, ecleklm: #CDC-KLM, ECL-KLM matching>=1	1.11	0.83
CDC τ/dark	**stt**: #full track>=1, |z|<15cm, p>0.7GeV	2.93	1.37
	syo: #full track>=1, #short track>=1, Δϕ>90deg, |z|<20cm	1.93	0.63
	fy30: #full track>=2, Δϕ>30deg, |z|<20cm	2.59	0.22
ECL τ/dark	**lml**: several combination of #cluster and energy	3.92	2.18
	eclmumu: back to back low energy hit	0.63	0.01
Calibration with prescale>1	PID (two photon)	0.35	0.16
	Other (Bhabha, γγ, random, trg)	1.00	0.60
Total L1	**OR of all bits**	**11.5**	**11.5**

CDC TRG rate contribute large part of total L1 trigger

We want to reduce it by 50%

CDC TRG :

CDCTRG Modified Module	Required rate reduction to achieve 50% (status)
CDCFE crosstalk filter, ADC	~10% (not yet)
CDCTRG 2D	~20% (achieved by simulation)
CDCTRG NN, 3D, 3DHough	~30% (not yet)
Total	~50%
(CDC-ECL matching)	30~50% (achieved by data, not used)

What I working on

Need events with |z|<1

→Cut at 20/15cm

→ Better resolution for z could restrict the cut

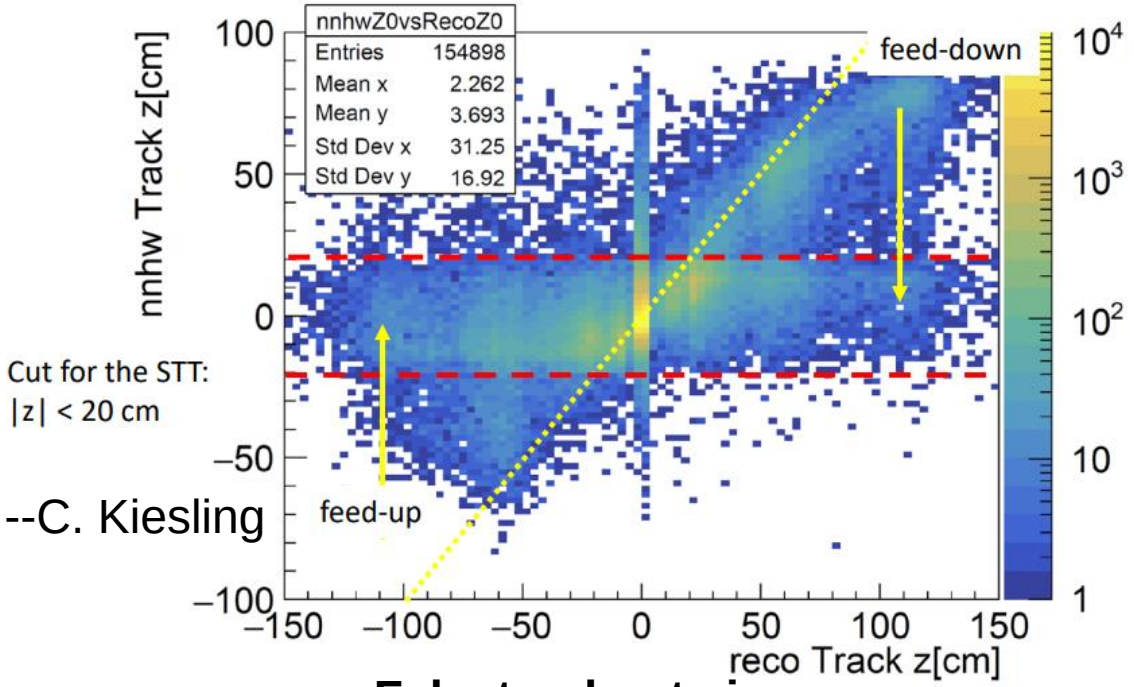
→ reject more Bkg while keeping efficiency

Main idea for NN: Improve z resolution

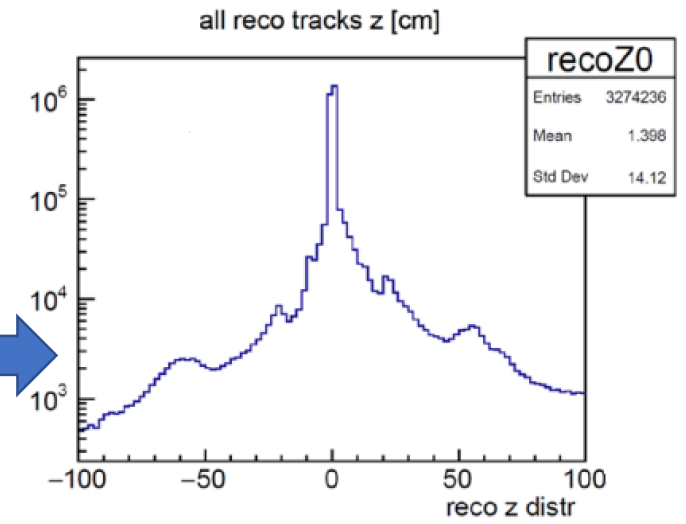
4

Motivation –Problem for CDCNN TRG

“Feed Up” and Feed down in large reco track z area



--C. Kiesling

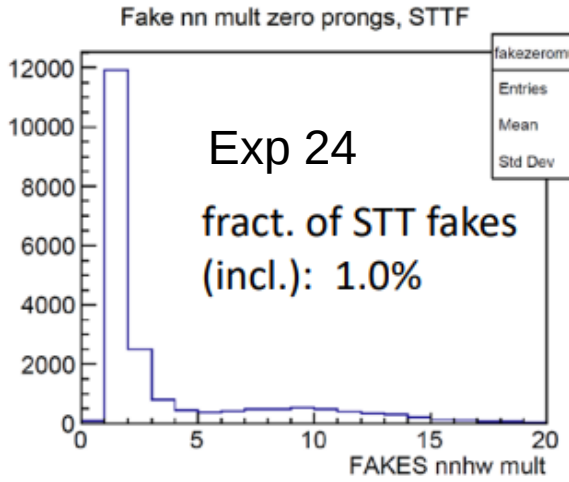


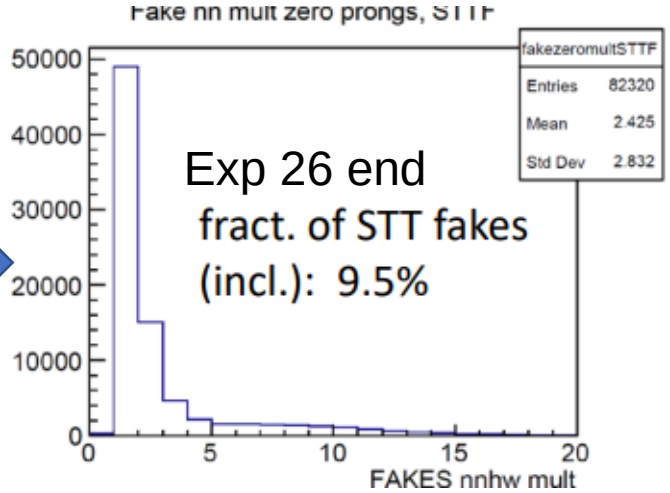
May due to the not proper z distribution for Training set

(Give large weigh on track at IP)

- Could be reduce by better Training set (Or put more weigh on large z area)
- Add extra input may also improve it

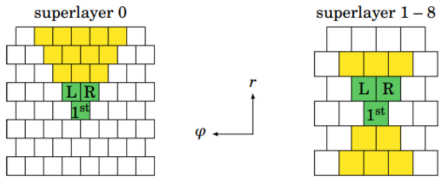
Fake track rate increase



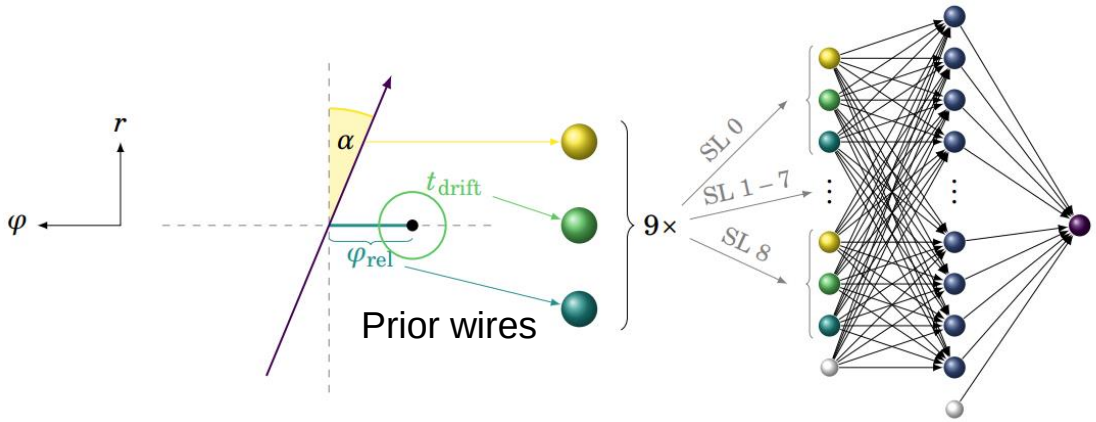


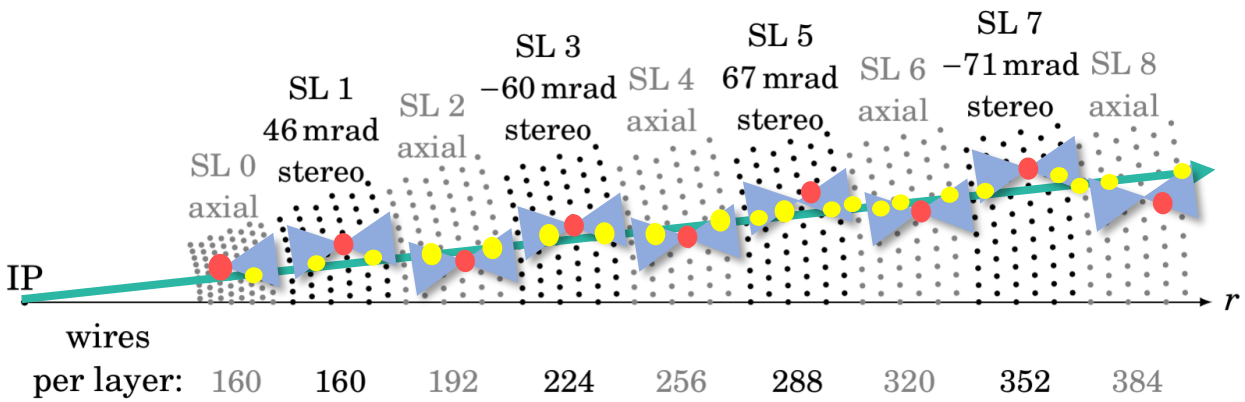
- Not try yet… Will perform detail categorizing for recent data
- Will try to use ADC inform to reject some “Fake Track Segment“ and reduce Fake Track

Motivation



"Track Segment" as a minimal unit CDC TRG





Wires hit in a TS -- Plan to add

Priority wires -- Already used

Present 3D NN use only one prior wire per every Track Segment.

With UT4 Module, more input and larger NN is possible for CDCTRG NN

For extra wire even with $\sigma_{t_{drift}} \sim 32ns$

$$z_0 = z_{cross} - \cot\theta_0 \frac{2\alpha}{\omega}$$

$$\Delta z_{cross} = \frac{r_{wire}}{\sin\psi}\sqrt{(\Delta\phi_{cross})^2 + (\Delta\phi_B)^2}$$

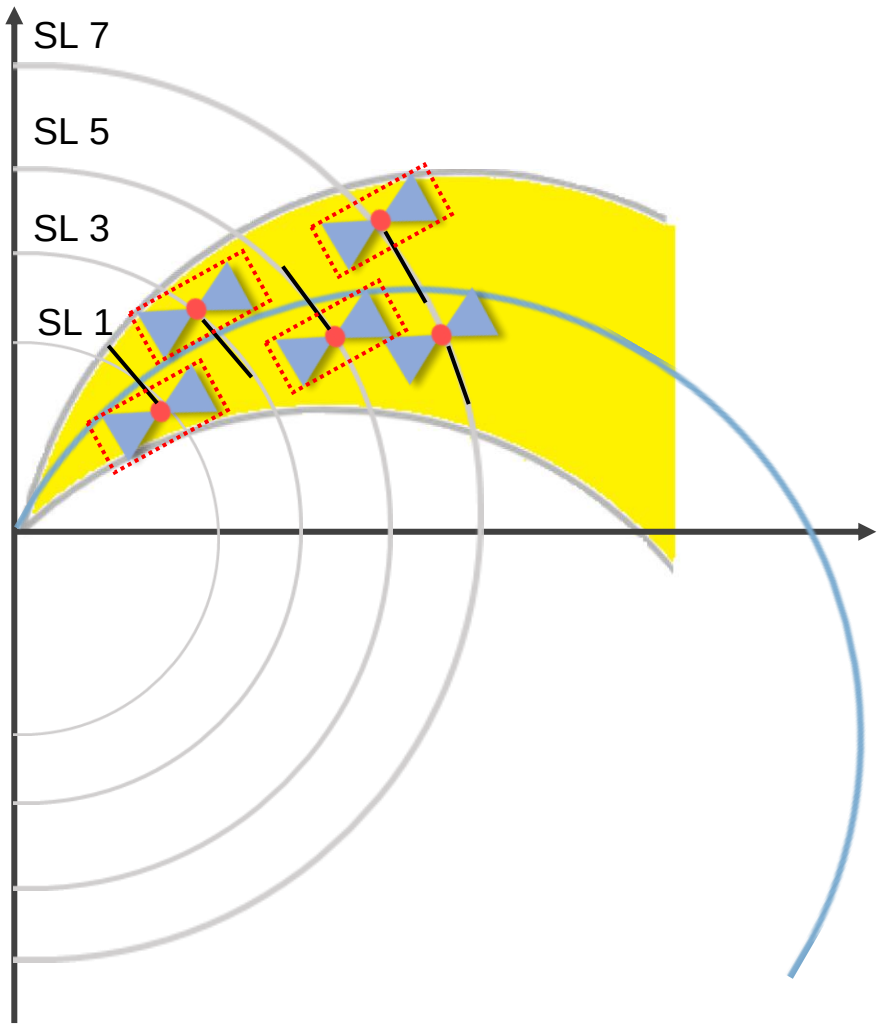
The Δz_{cross} calculated by a single wire is ($P_t > 0.4GeV$)

$$\Delta z_{cross} \sim 2.0 \text{ cm to } 3.4 \text{ cm}$$

In the same order of prior wire (0.4cm ~ 1.4cm)

Can be used to improve the resolution of NN.

How CDCNN TRG work



CDC cross-sectional schematic
(Only stereo layer showed)

1. Get 2D Track from CDC2DHough (Calculate from all axial layer)

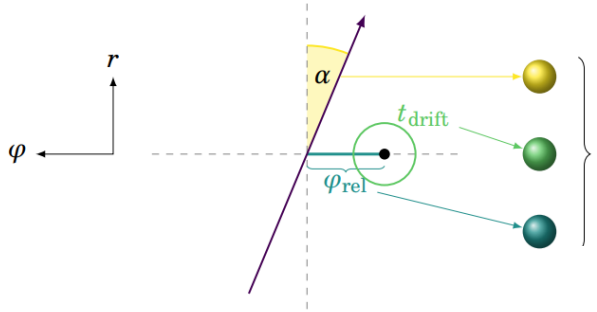
2. Set relative range based on previous RecoTracks

3. Find out all TS in relative range

4. Select out one TS for every SL (first with left/right known; next with smallest drift time)

`if n TS selected >3`

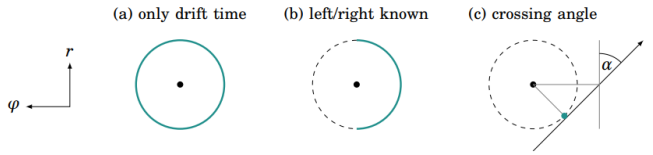
5.1 Use parameter from selected TS and 2D track as input for 5 different NN (for varied missing SL):



$$\phi_{rel} = \phi_{wire} - \phi_{cross}$$

$$\alpha: from\ 2D\ track\ and\ crossing\ SL$$

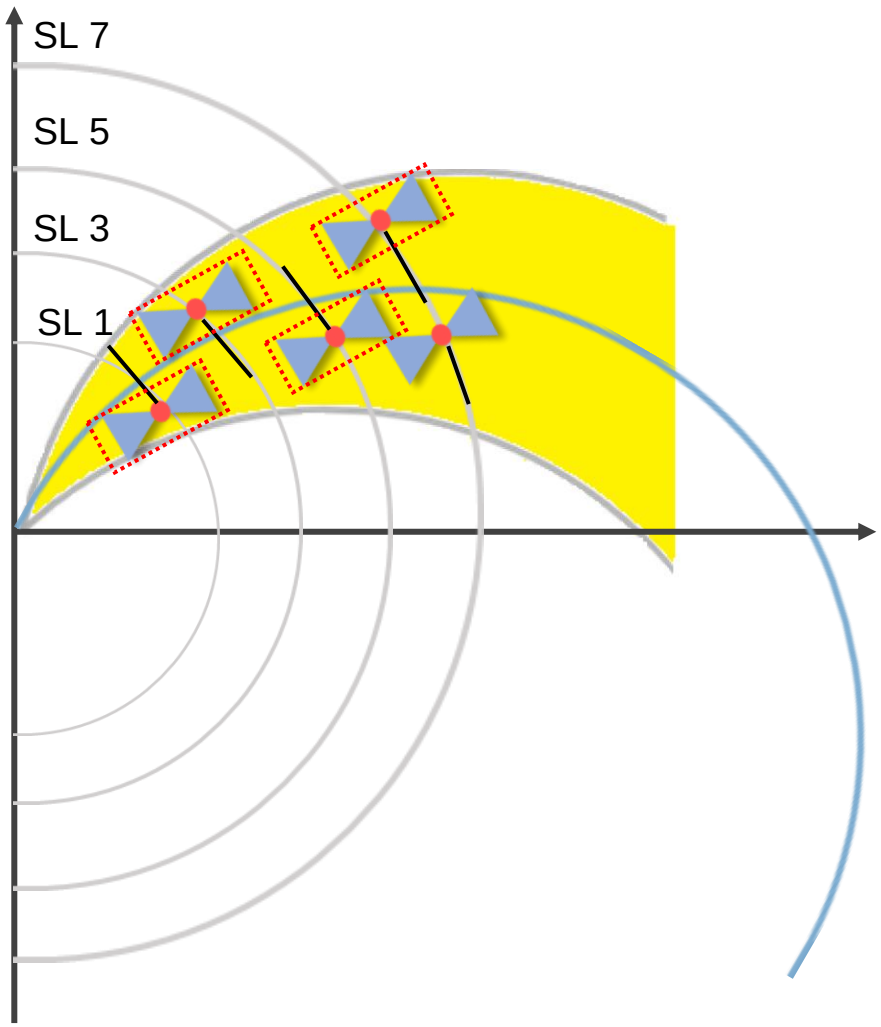
$$t_{drift} = (t_{wire} - t_0) * L/R$$



`else`

5.2 Not build 3D NN track

What we want to add



CDC cross-sectional schematic
(Only stereo layer showed)

1. Get 2D Track from CDC2DHough (Calculate from all axial layer)

2. Set relative range based on previous RecoTracks

3. Find out all TS in relative range

4. Select out one TS for every SL (first with left/right known; next with smallest drift time) Might be improved

```
if n TS selected >3
```

5.1 Use parameter from selected TS and 2D track as input for 5 different NN (for varied missing SL):

5.1.2 Select out 1/2/3 wire(s) for every TS (not prior one; first with left/right known; next with smallest drift time)

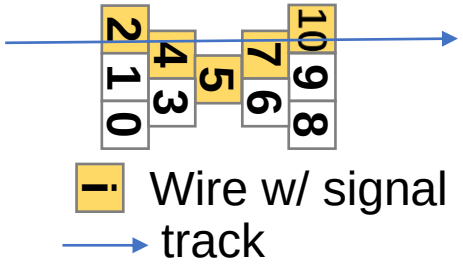
5.1.3 Use parameter from selected TS, 2D Track and selected extra wires

```
else
```

5.2 Not build 3D NN track

Check the L/R impact for NN

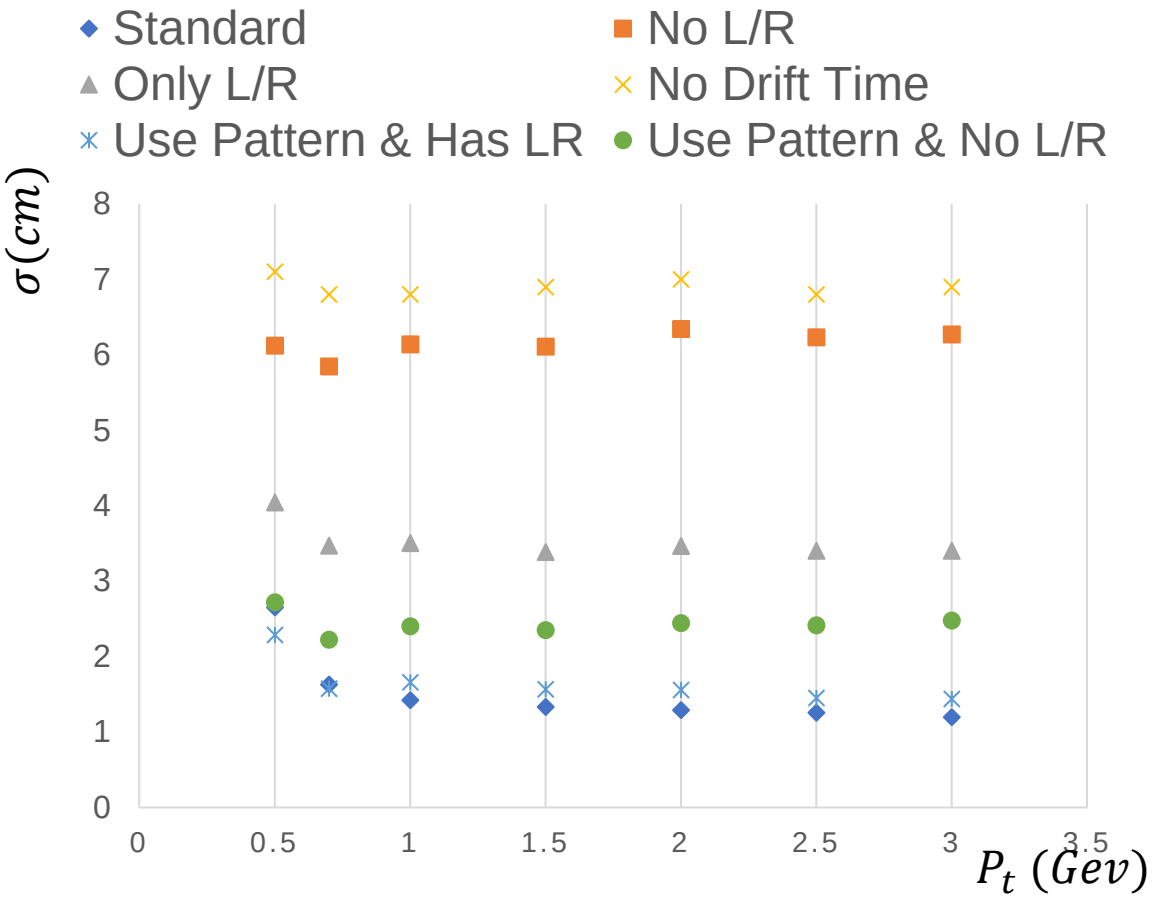
How to determine Left/Right:



A certain pattern →Count over a large number of events and find the most likely L/R

Only for **prior wire,** not available for extra wires

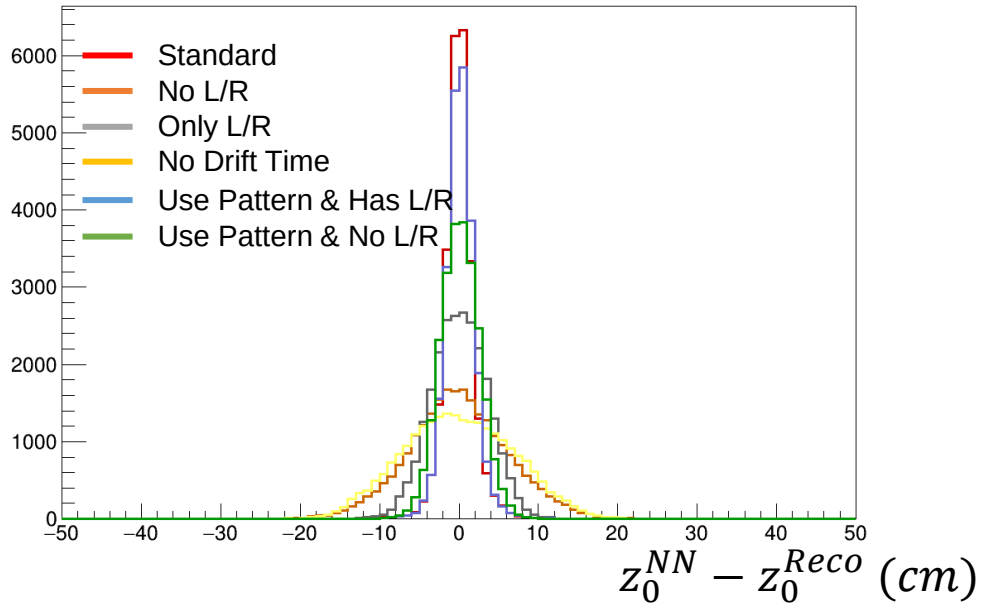
Build a “Full LUT” for every wires?



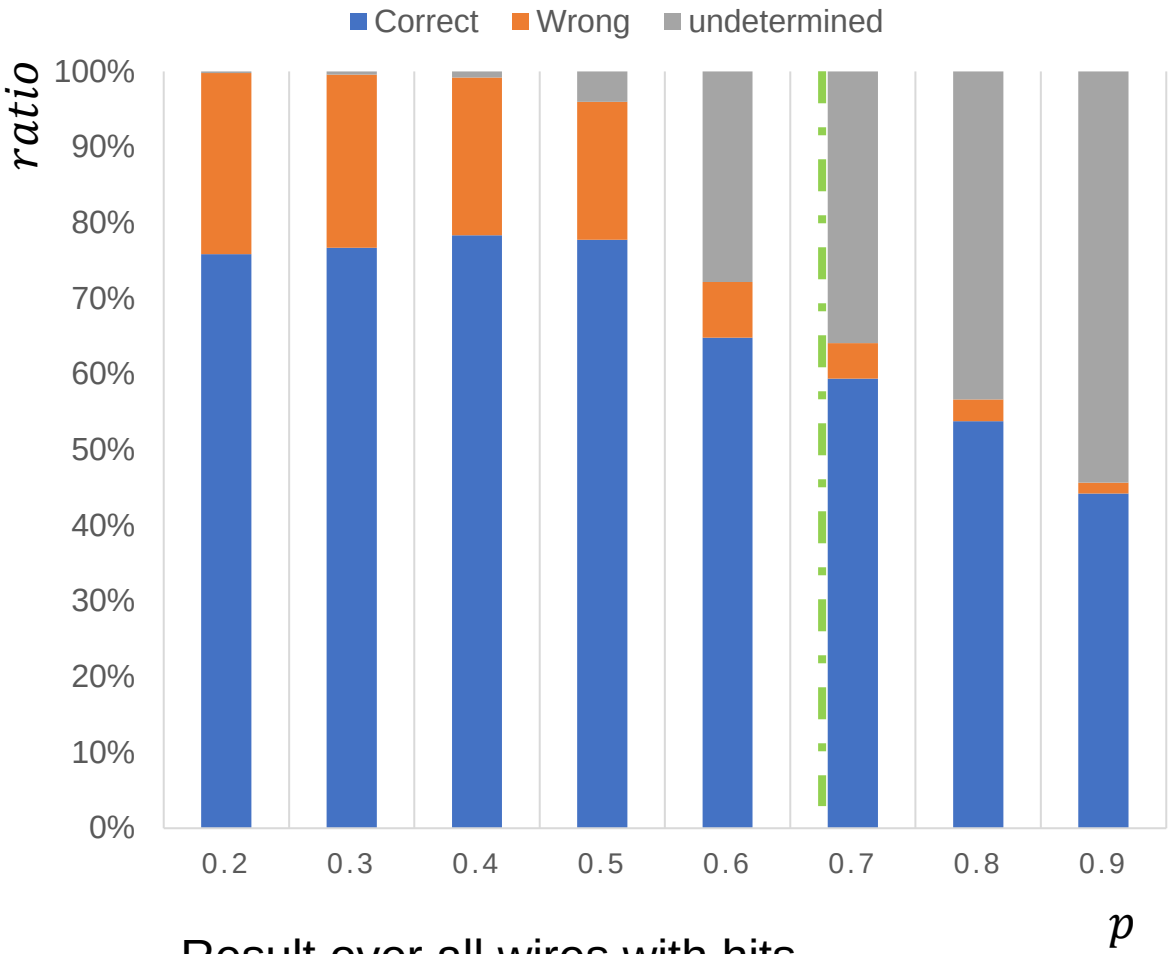
Trained NN based on single track MC w/ fann

L/R is extremely important for currently NN

Pattern input can not fully replace L/R. Even with both pattern and L/R, no improvement for the standard one



Build L/R LUT table for every wires in TS



Result over all wires with hits (exclude prior one)

Following the old way to build up a LUT for **every wires in TS**

Use MC without Bkg first

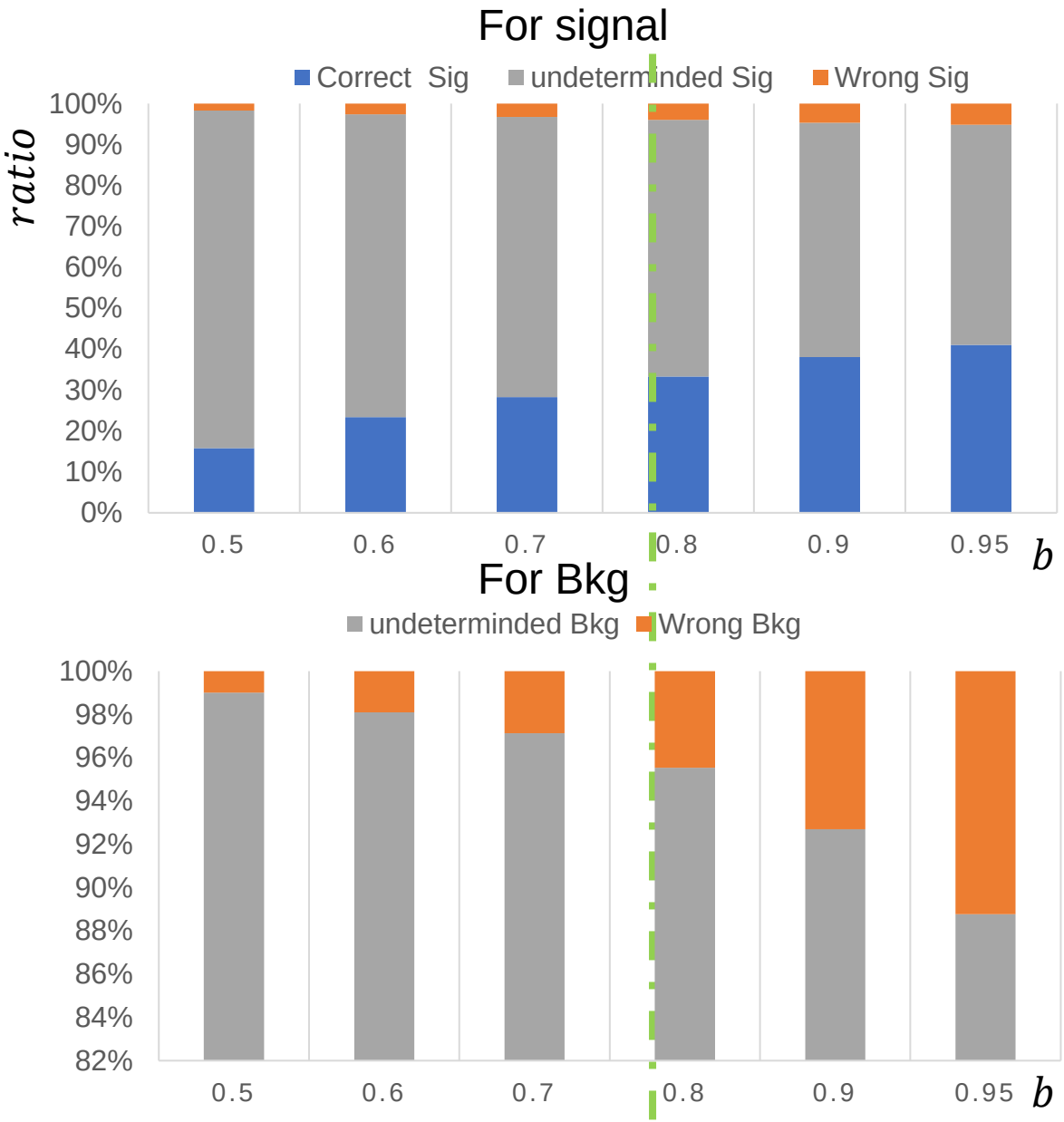
$$L/R\ state = \begin{cases} left & if\ n_L > p(n_L + n_R) + 3\sigma \\ right & if\ n_R > p(n_L + n_R) + 3\sigma \\ undecide & otherwise \end{cases}$$

$$\sigma = \sqrt{(n_L + n_R)p(1-P)}$$

Choose P = 0.7 for LUT.

Since undetermined rate is high, for more wires (>1) case, undetermined events increase

Build L/R LUT table for every wires in TS



build up a LUT include Bkg

Use MC with early Phase III Bkg

$$L/R\ state(Bkg) = \begin{cases} signal: & otherwise \\ Bkg: & if\ n_b > b(n_{Total}) \end{cases}$$

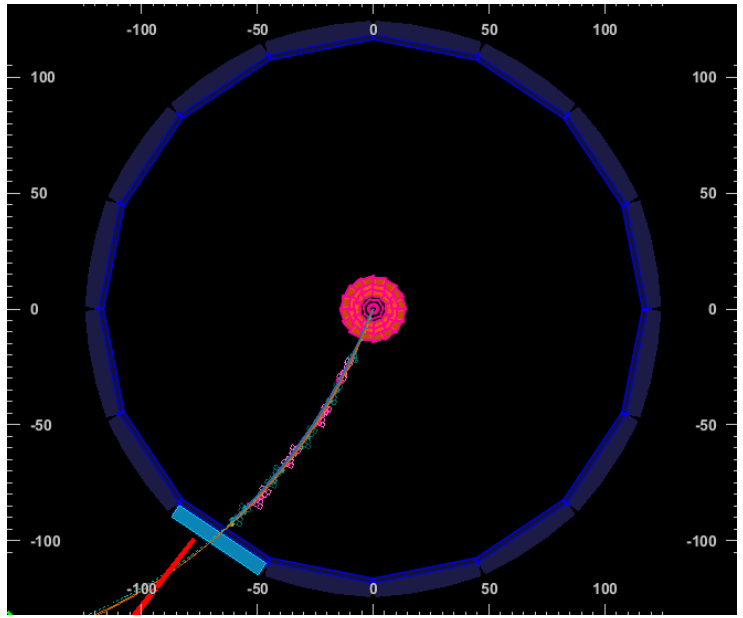
Want to got correct sig & undetermined Bkg (Which will have a low priority for Bkg hits) as much as possible

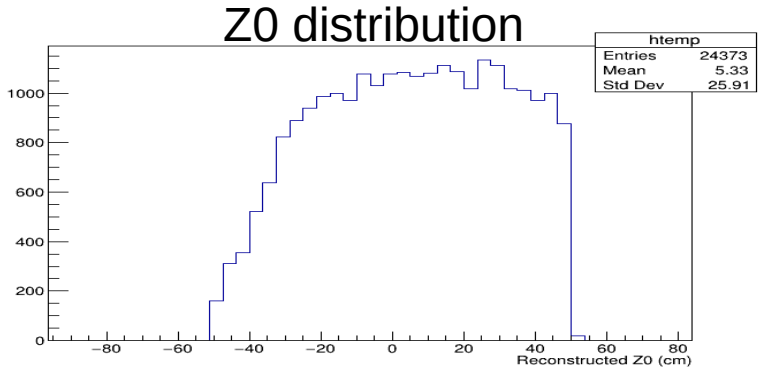
Choose b = 0.8 for LUT. To make sure that for a TS, we have large possibility to find at least one extra wires with L/R known

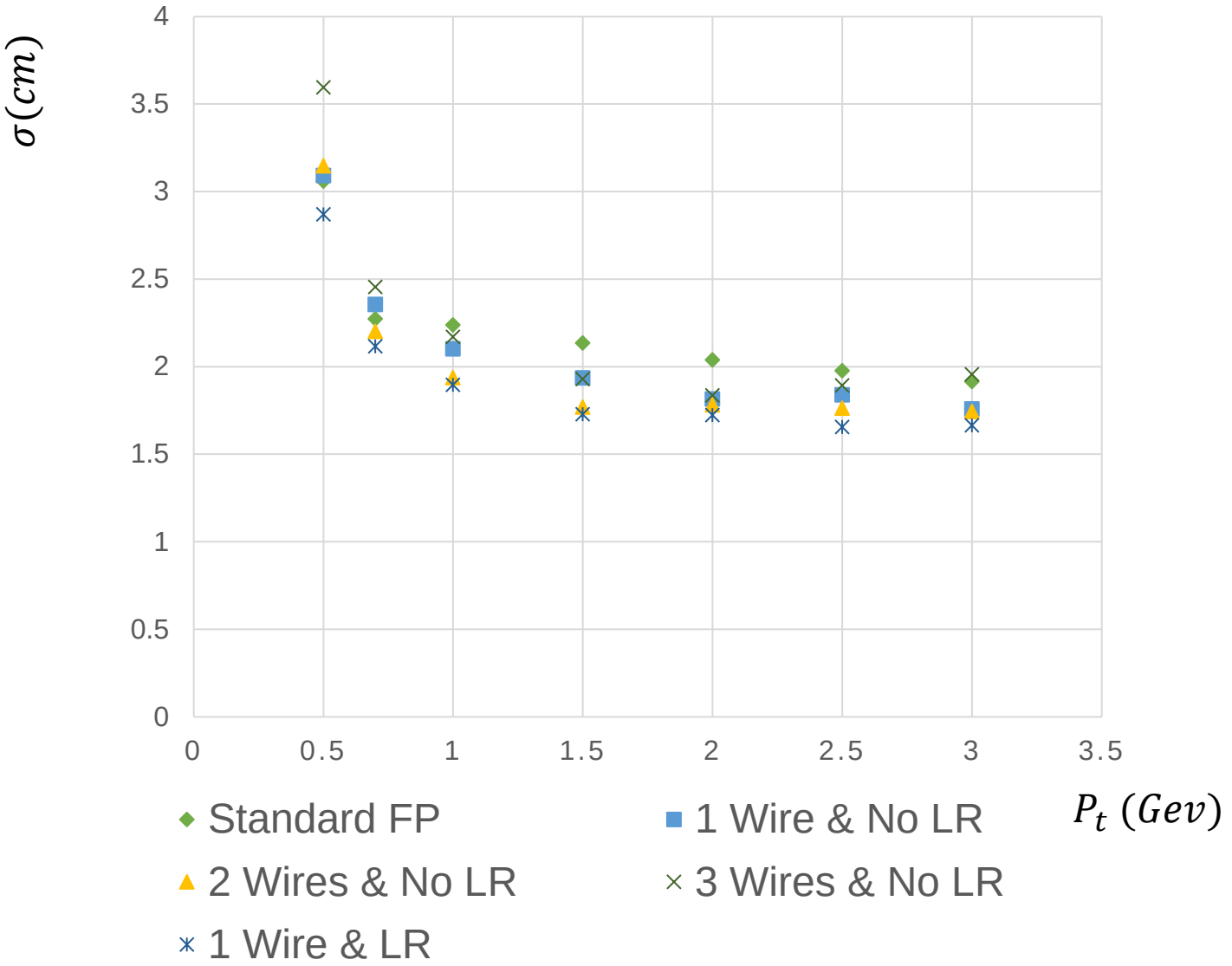
Will generated LUT with Recotrack soon

MC Test

MC :
Single track w/o Bkg;
uniform Pt,Φ, θ and vertex z







Not large but could see improvement with L/R known extra wires

Pytorch training with real data

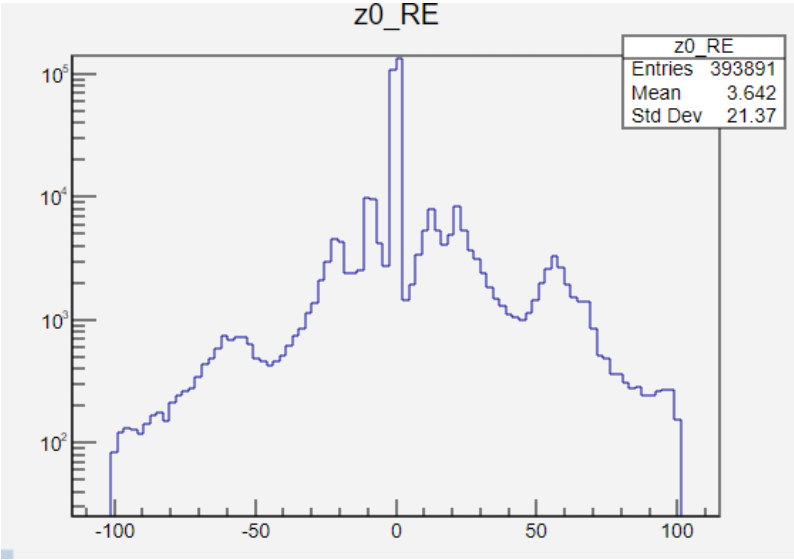
Data: exp26run1756-1780 (w/ beam reco monitor) (random separated to two set)

Generate training data with Extra 3 wires with LUT

Change training method to pytorch → faster convergence and better optimization

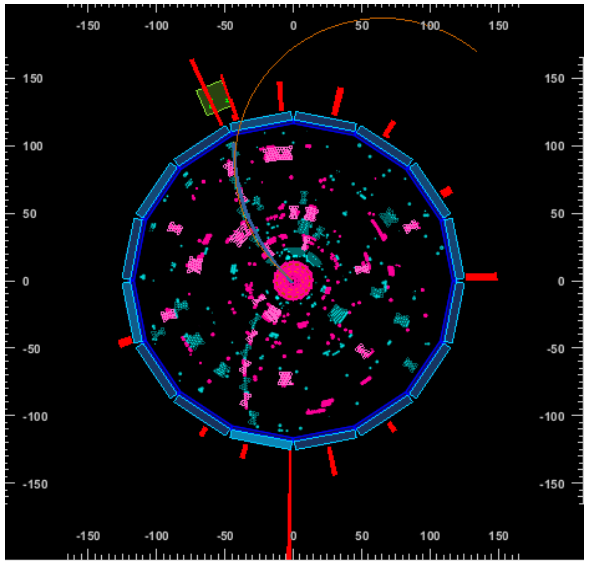
Using simulated ETFHough

Reco Z distribution for data

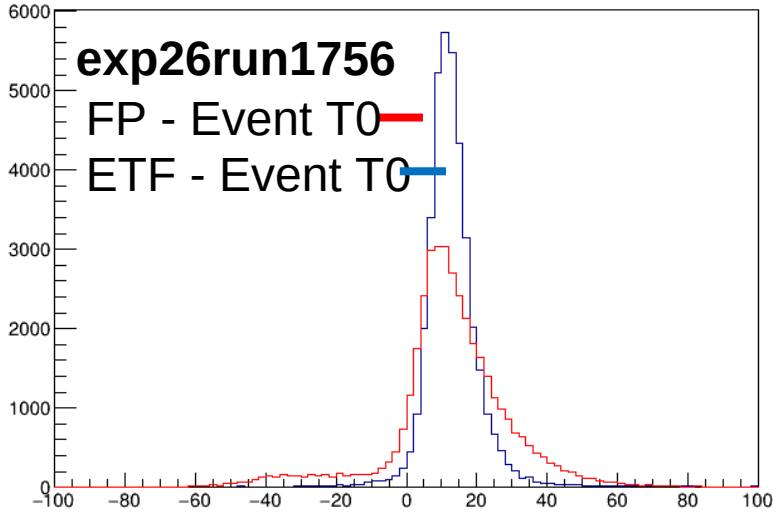


z_{RE} (cm)

Event display



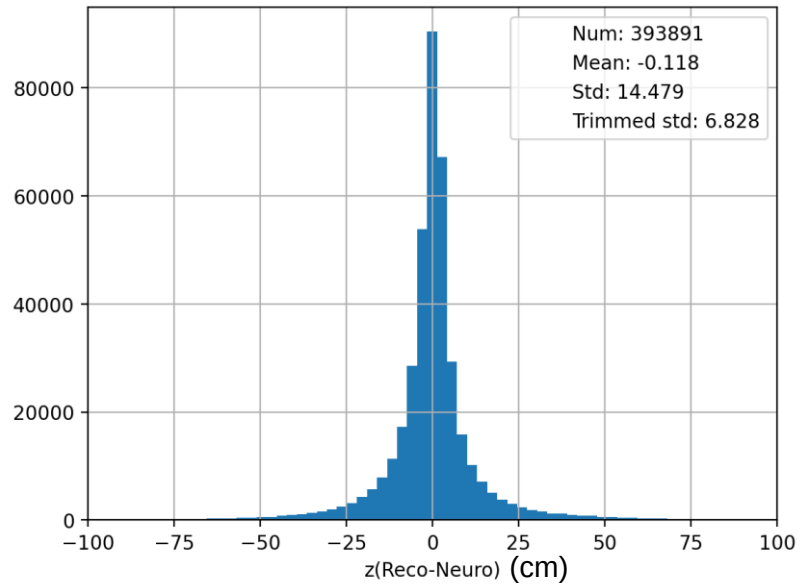
ETF compare with FP

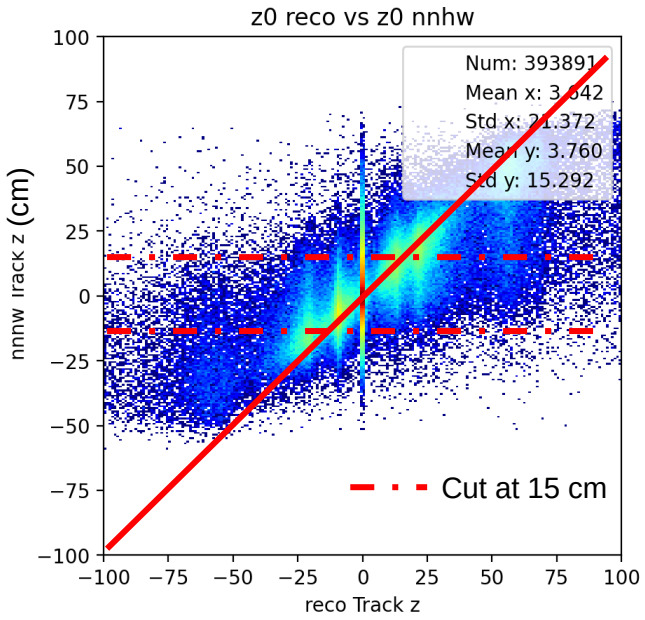


Pytorch training with real data

(Sum over all fives experts)

Standard method





$$\text{Trg efficiency} = \frac{\#|z_0| \text{ from RecoTracks} < 1 \;\&\&\# \; |z_0| \text{ from CDCNNTrack} < \text{cut}}{\#|z_0| \text{ from RecoTracks} < 1}$$

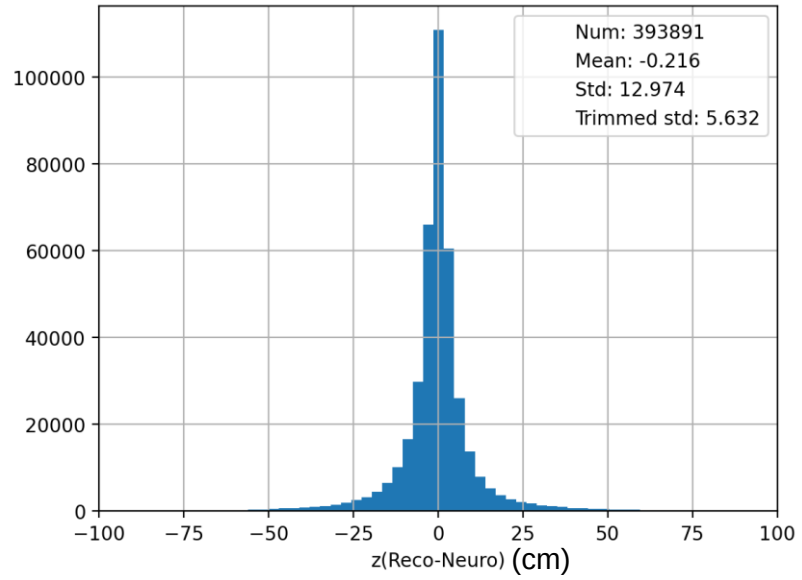
$$\text{Rejected rate} = \frac{\#|z_0| \text{ from RecoTracks} > 1 \;\&\&\# \; |z_0| \text{ from CDCNNTrack} > \text{cut}}{\#|z_0| \text{ from RecoTracks} > 1}$$

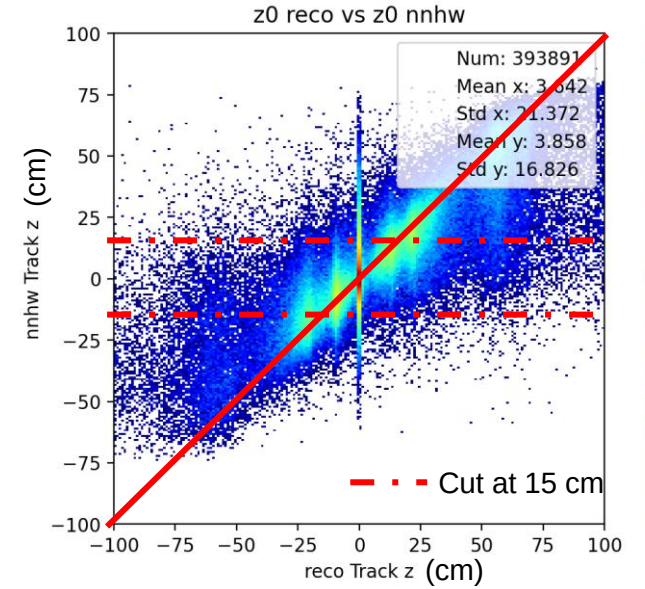
Regarding a needed events as |z0|<1 and z0 cut at 15 cm

TRG Efficiency: 94.6%

Rejected rate: 50.8%

Extra one wire





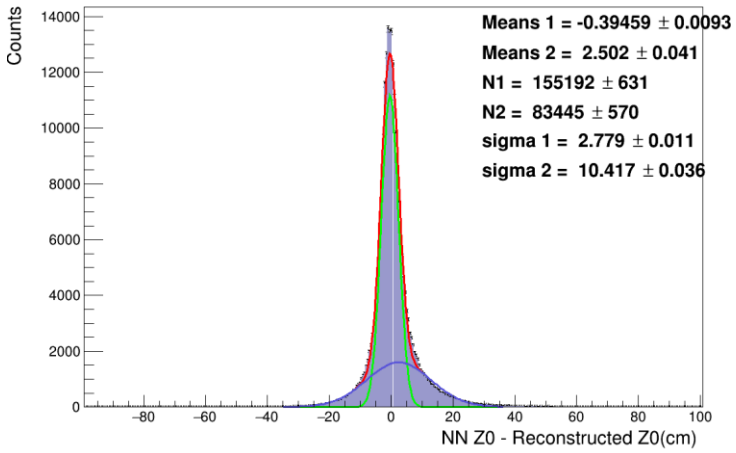
TRG Efficiency: 96.2%

Rejected rate: 53.5%

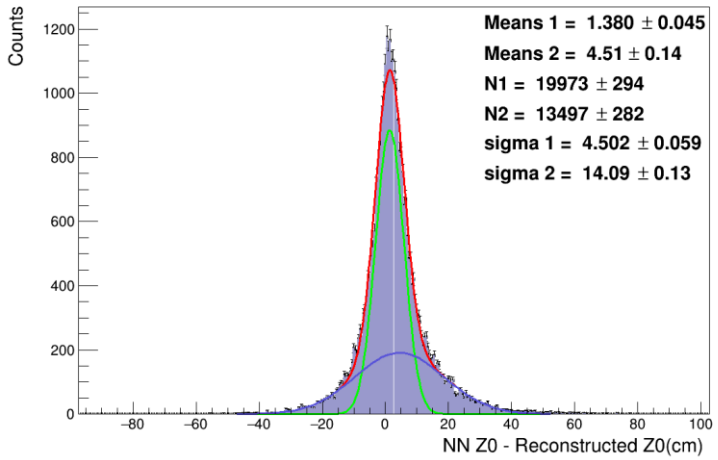
Better performance comparing with standard one, especially for large z0

Details performance at different z0

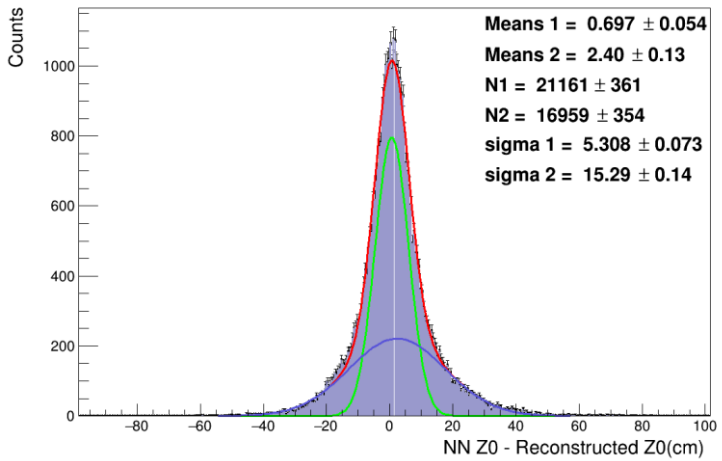
Standard method



Z<1

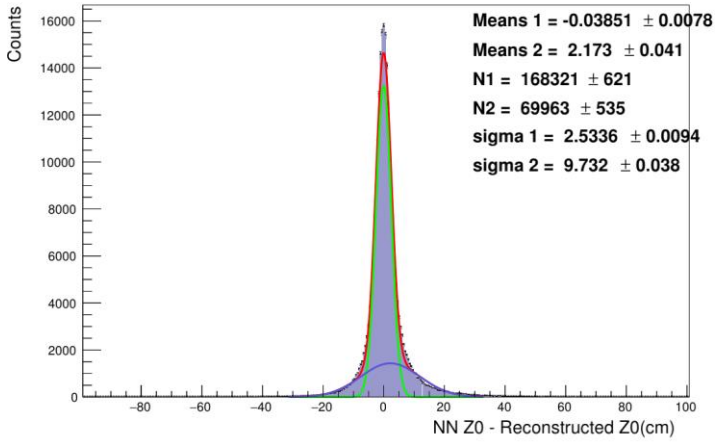


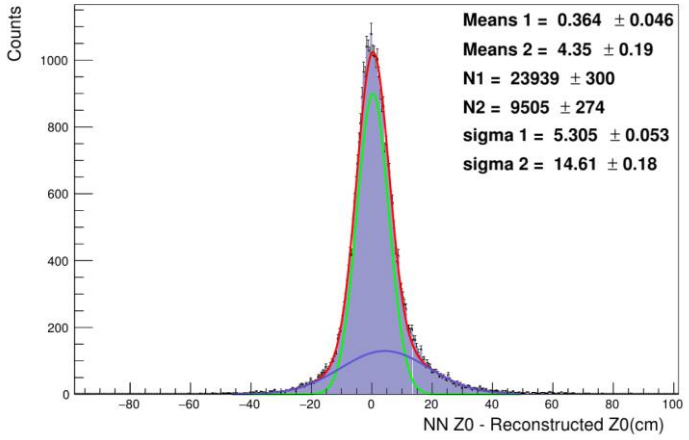
1<Z<10

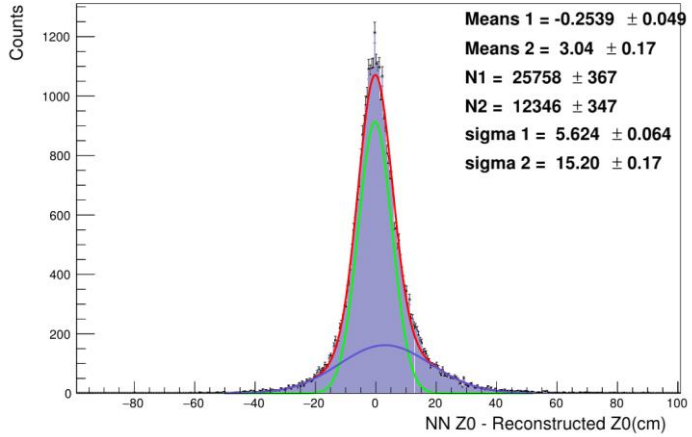


10<Z<20

Extra one wire

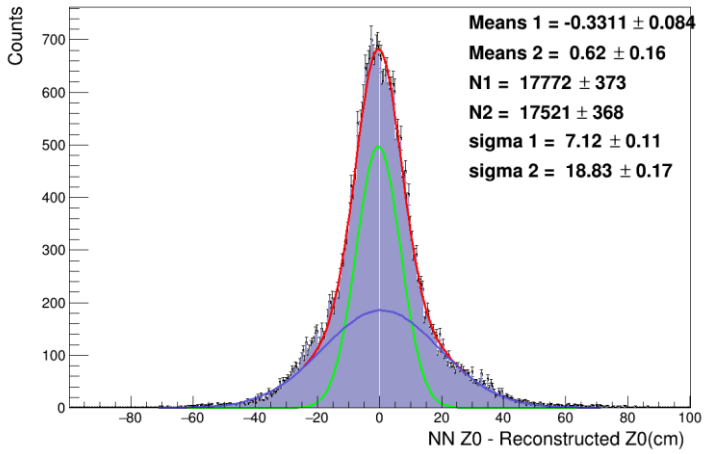




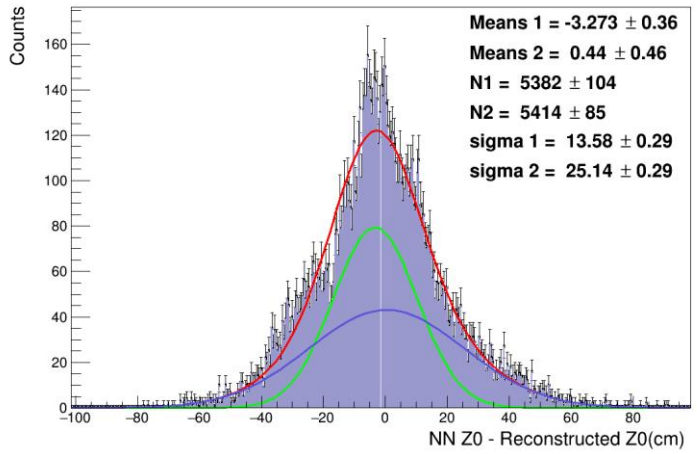


Details performance at different z0

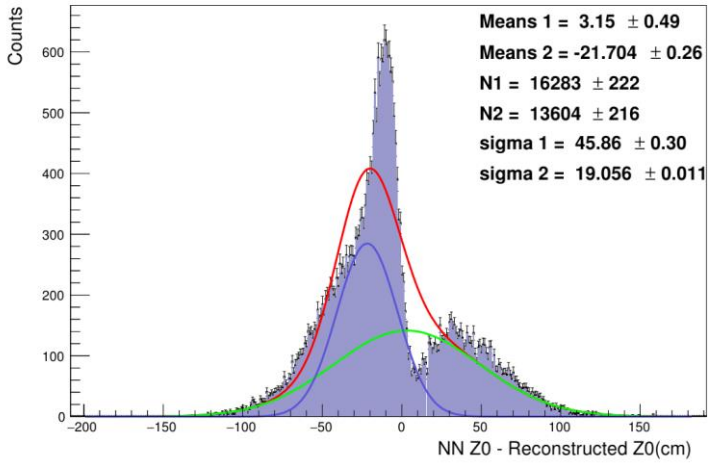
Standard method



20<Z<30

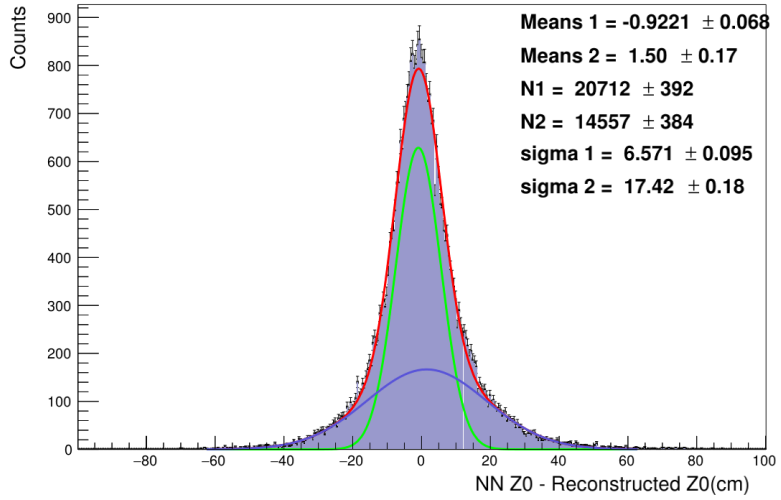


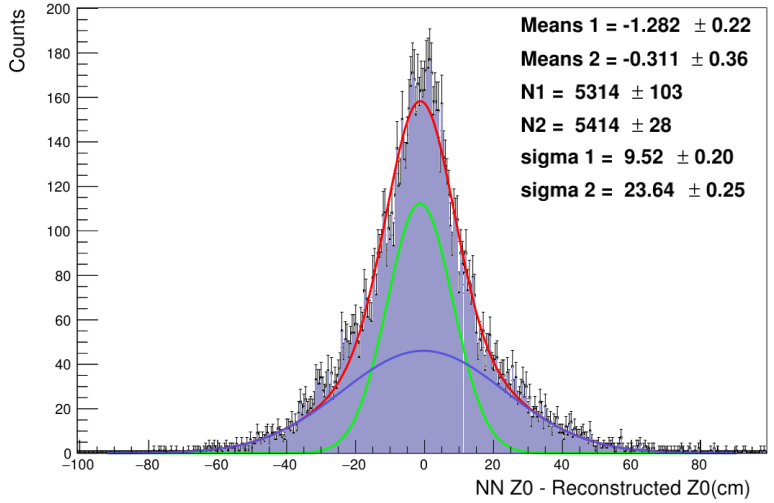
30<Z<40

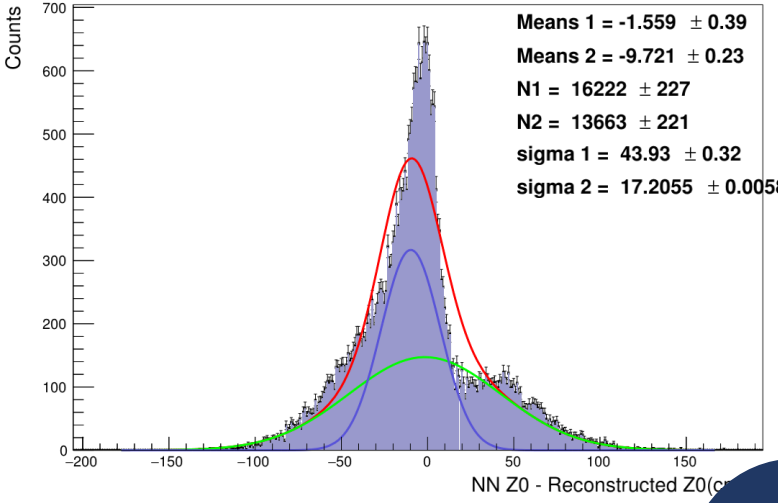


40<Z

Extra one wire



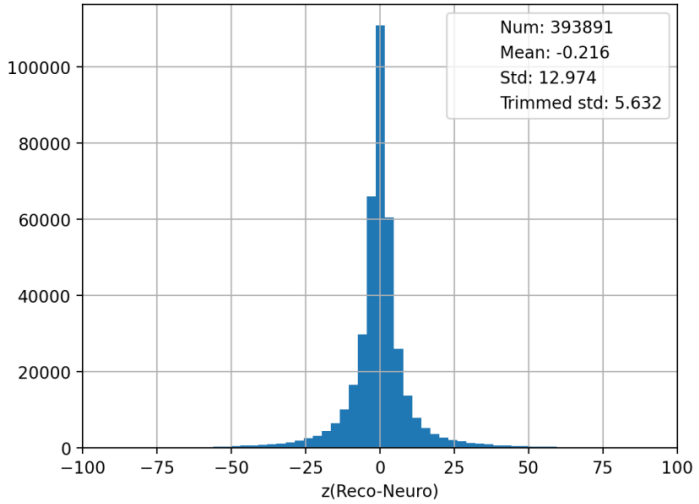


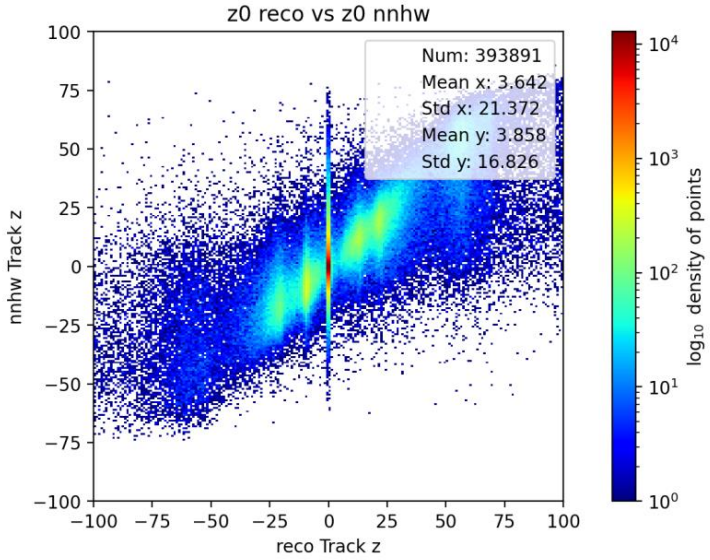


Still some “feed down” and feed up → leakage of training data?

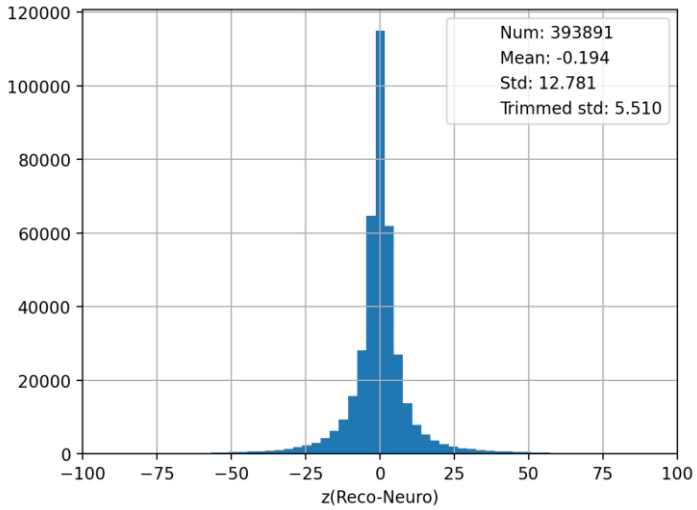
More Extra wires?

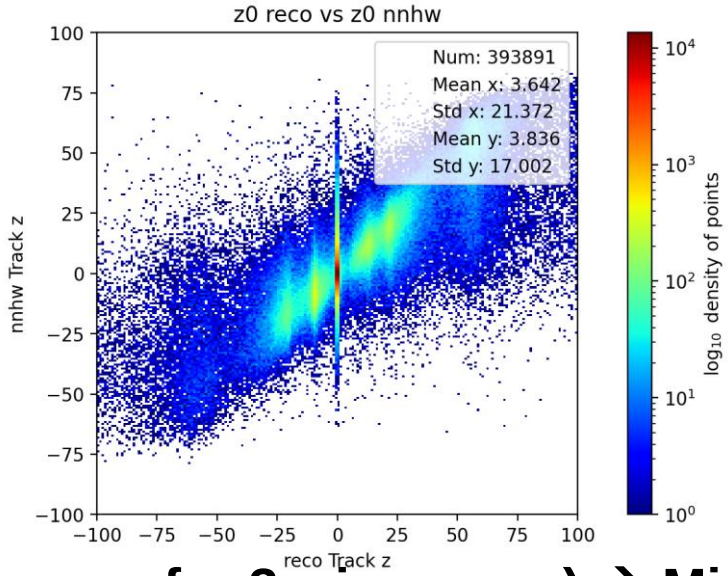
Extra one wire



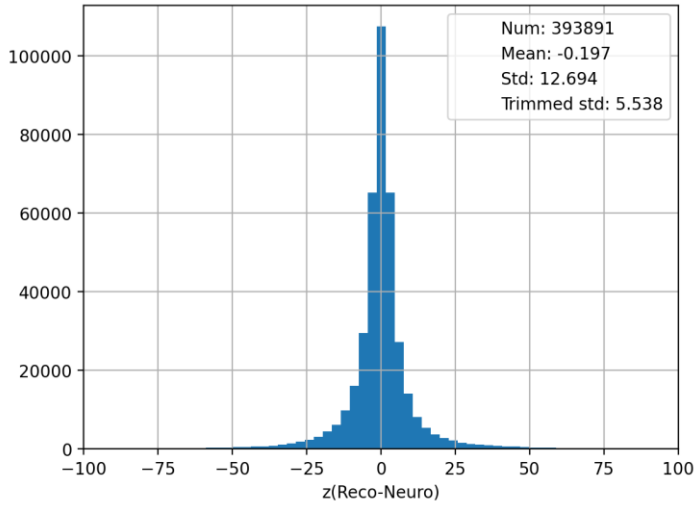


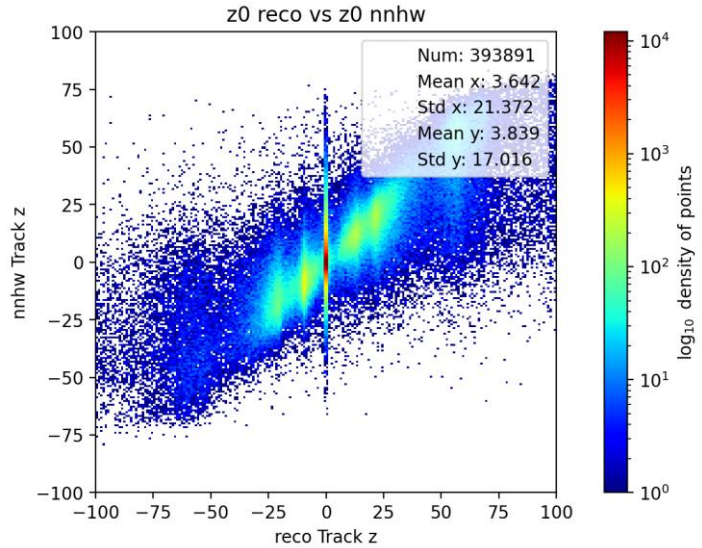
Extra two





Extra three

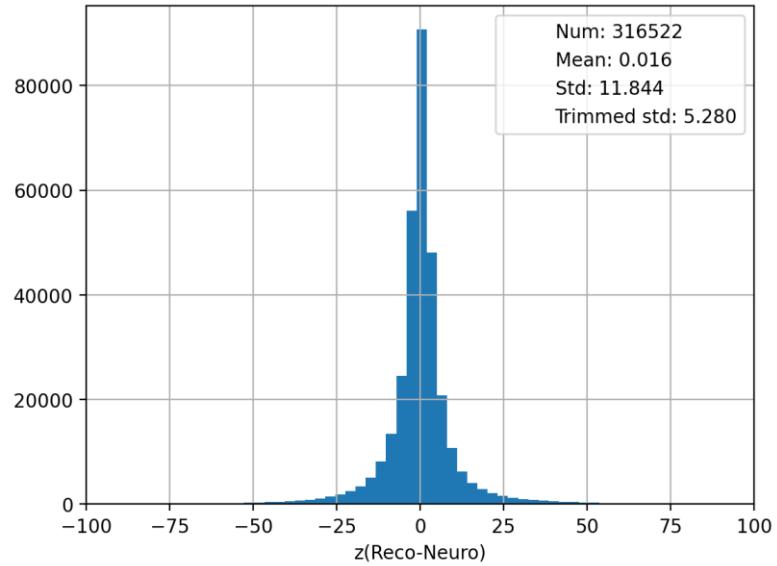




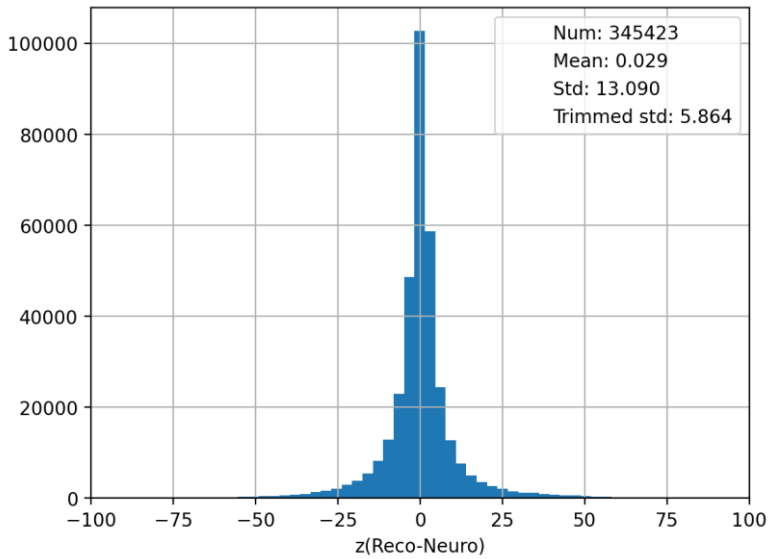
No large improvement (even worse for 3 wires case) → Might due to the L/R undetermined wires or not enough hidden layer/ hidden nodes

Difference between experts for extra 1 wire case

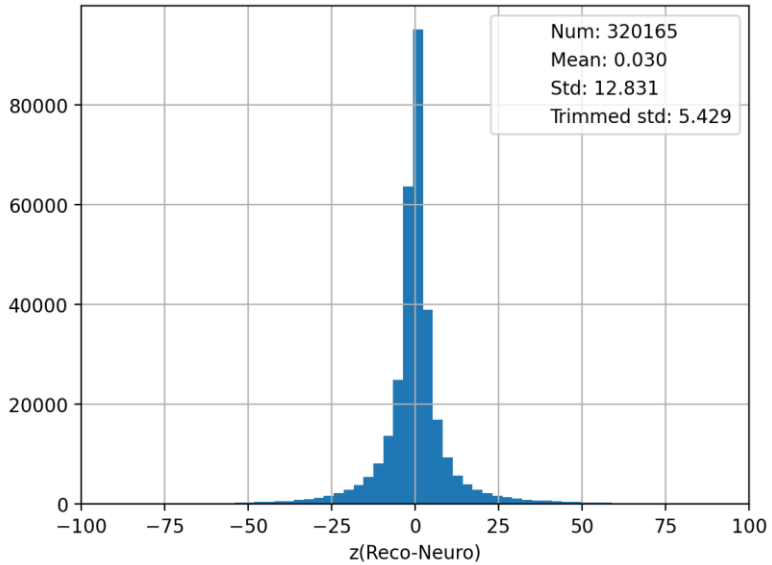
All SL have TS



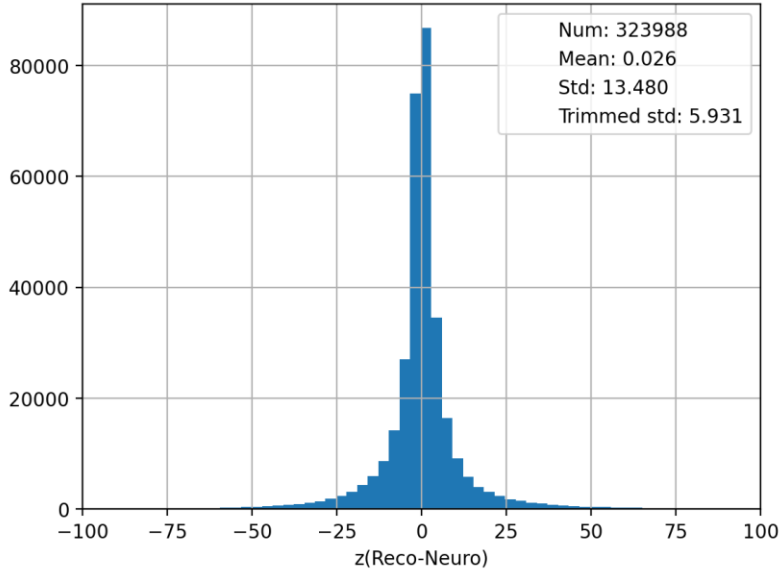
Missing SL 7



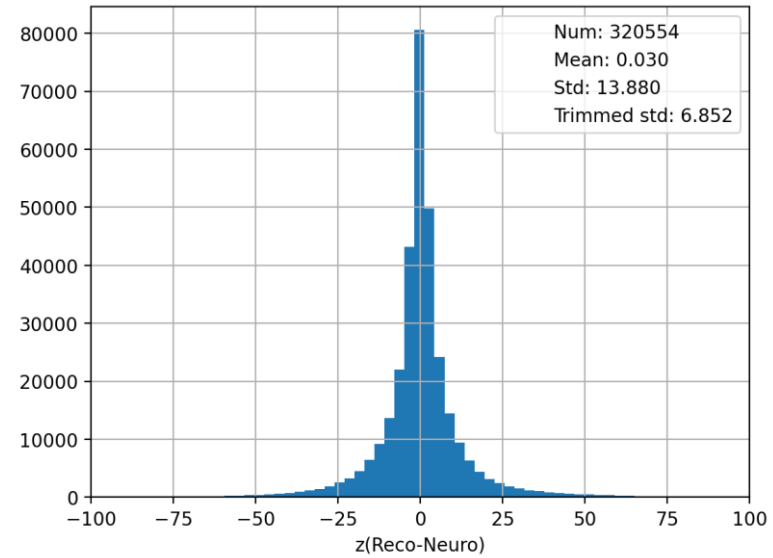
Missing SL 5



Missing SL 3



Missing SL1



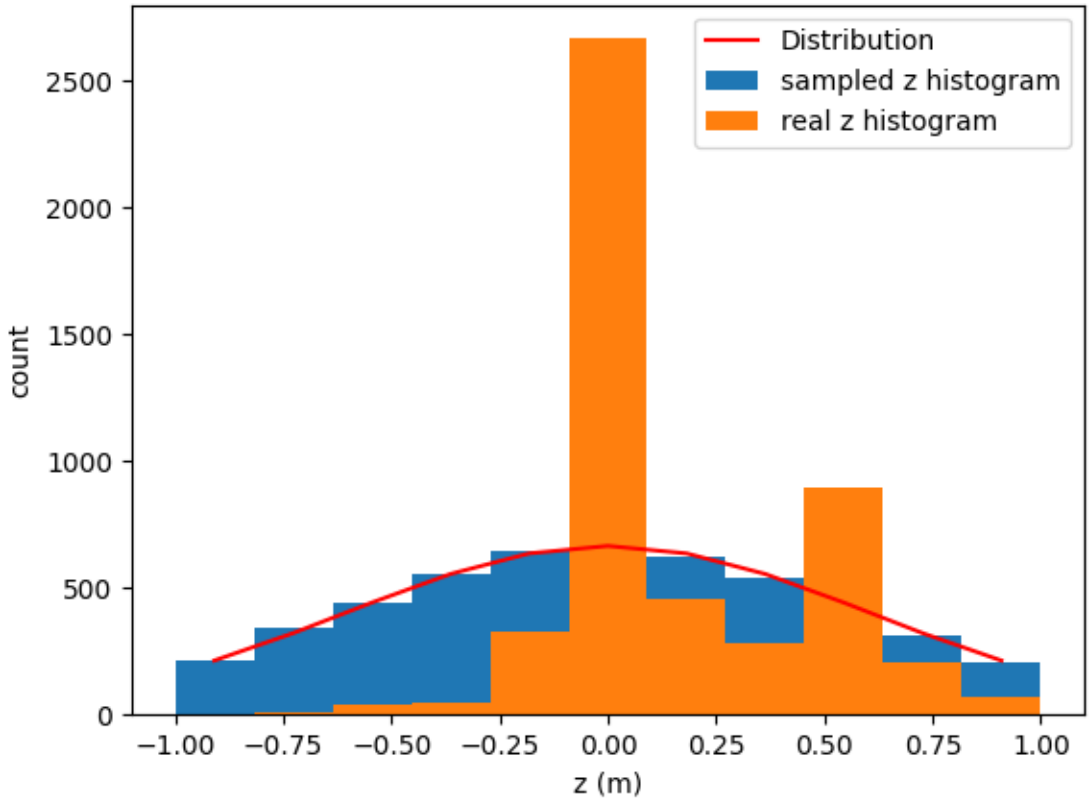
w/ missing SL1 still got worst result.

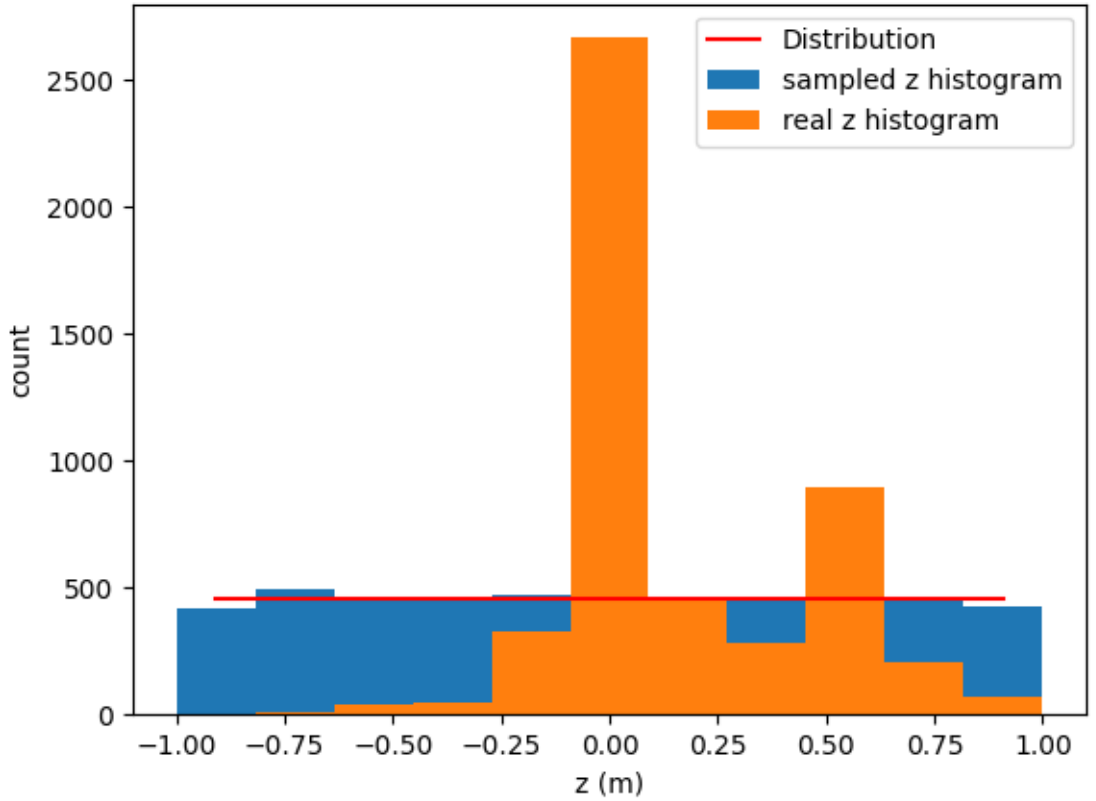
(However, expert 0 case dominate the events in exp26run1756-1780 >80%)

Reshape the training set?

We could reshape the data set to **randomly (uniform of follow gaus dist.)** pick up events at different z ↓

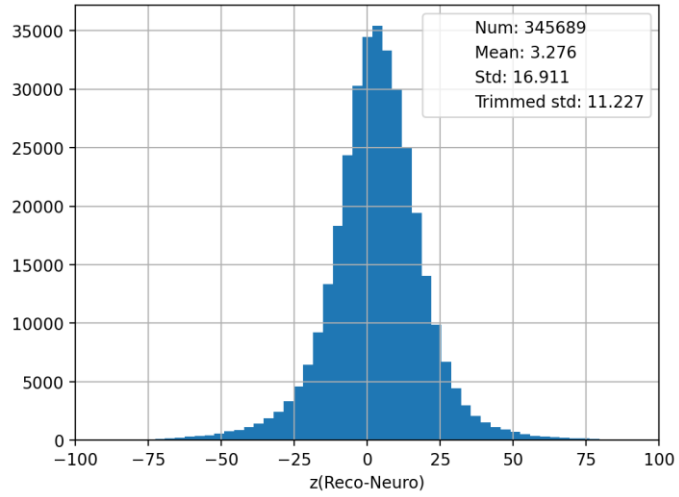
Still want to keep good resolution at IP→ not modified validation set

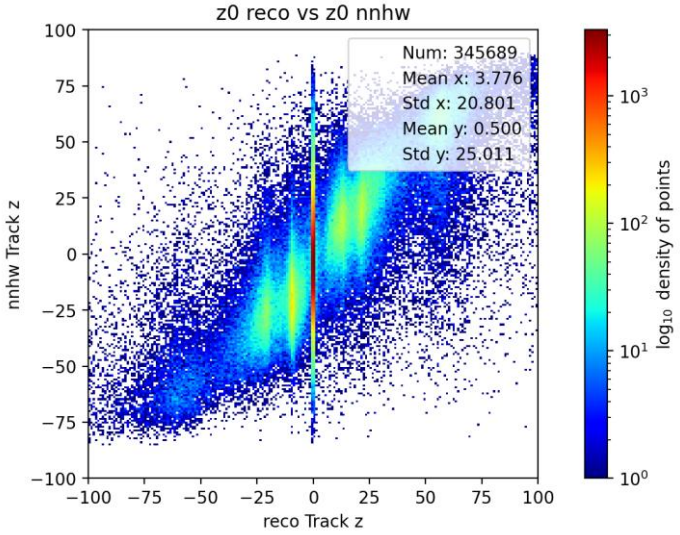




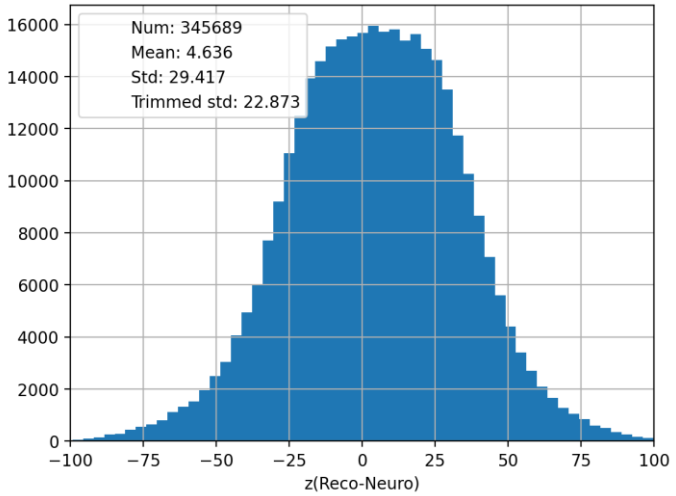
Reshape the training set?

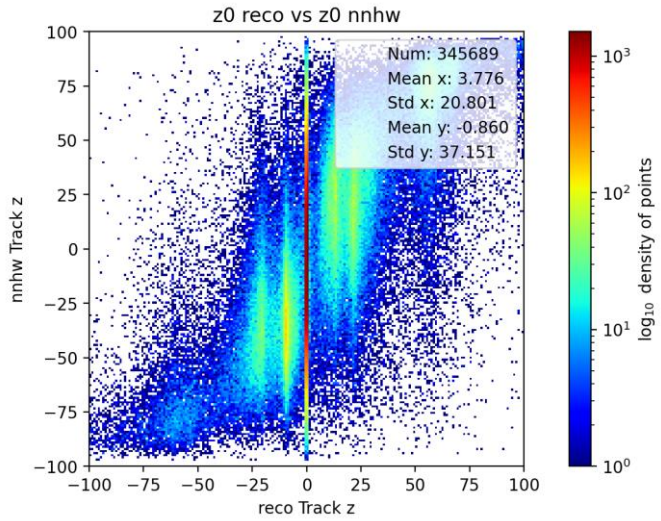
Norm distribution



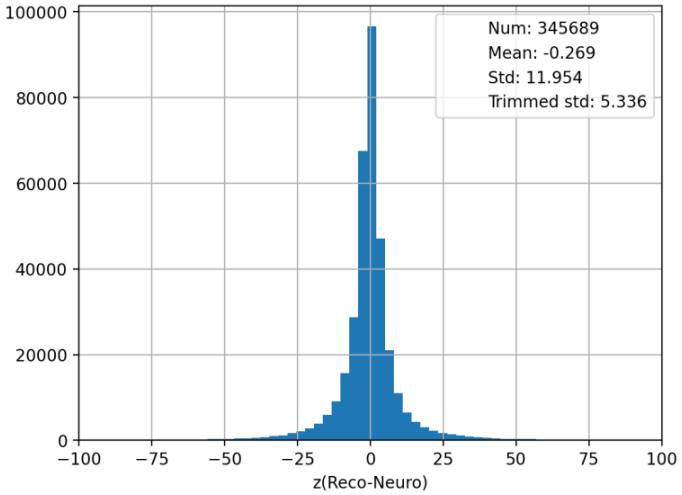


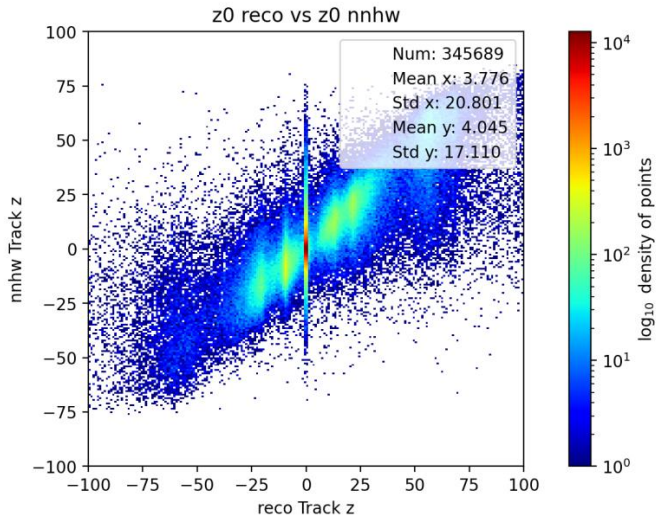
Uniform distribution





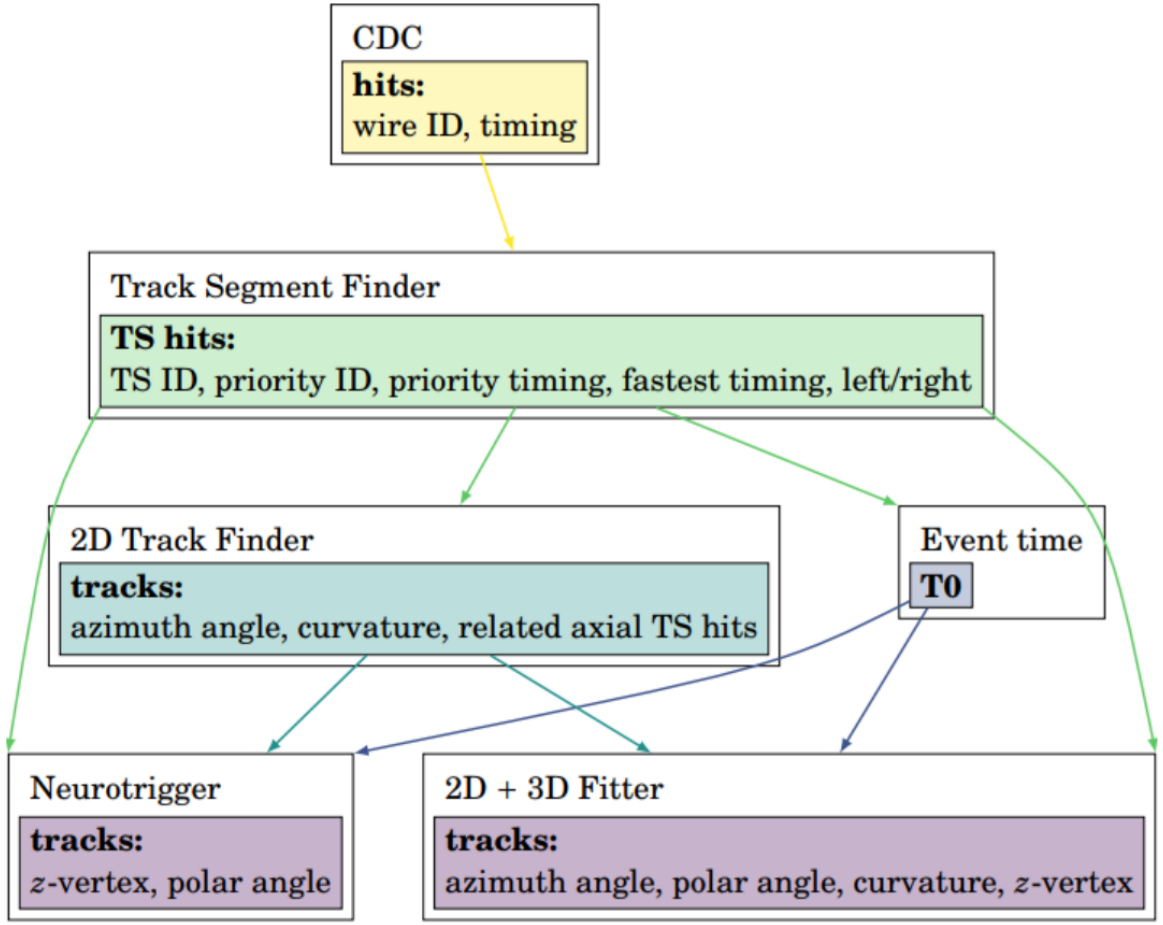
Norm Extra 1 Wires



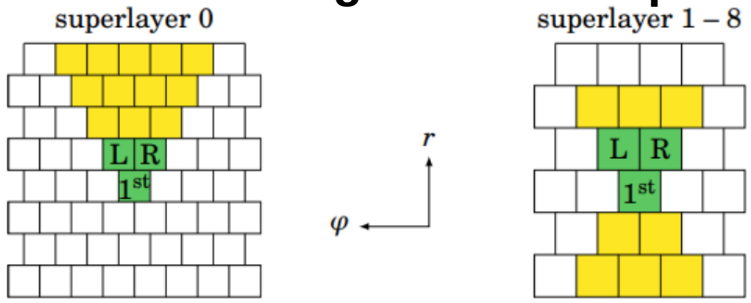


Converge soon due unexpected increasing error in validation set…

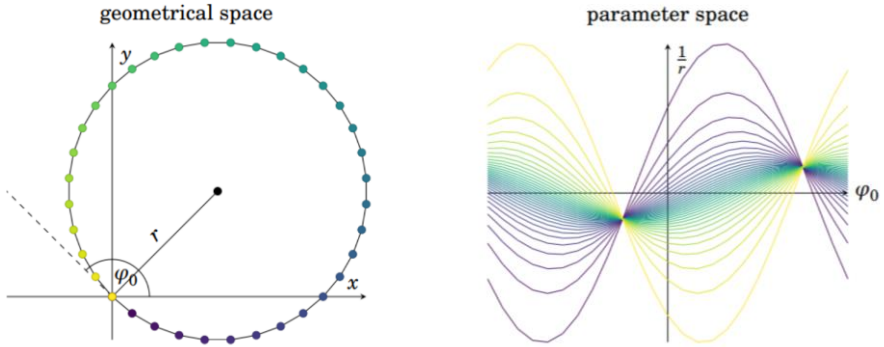
Introduction-CDC first level TRG



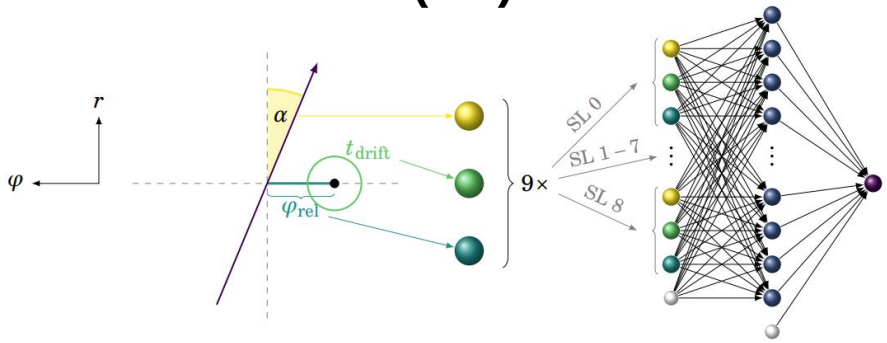
Track Segment build up



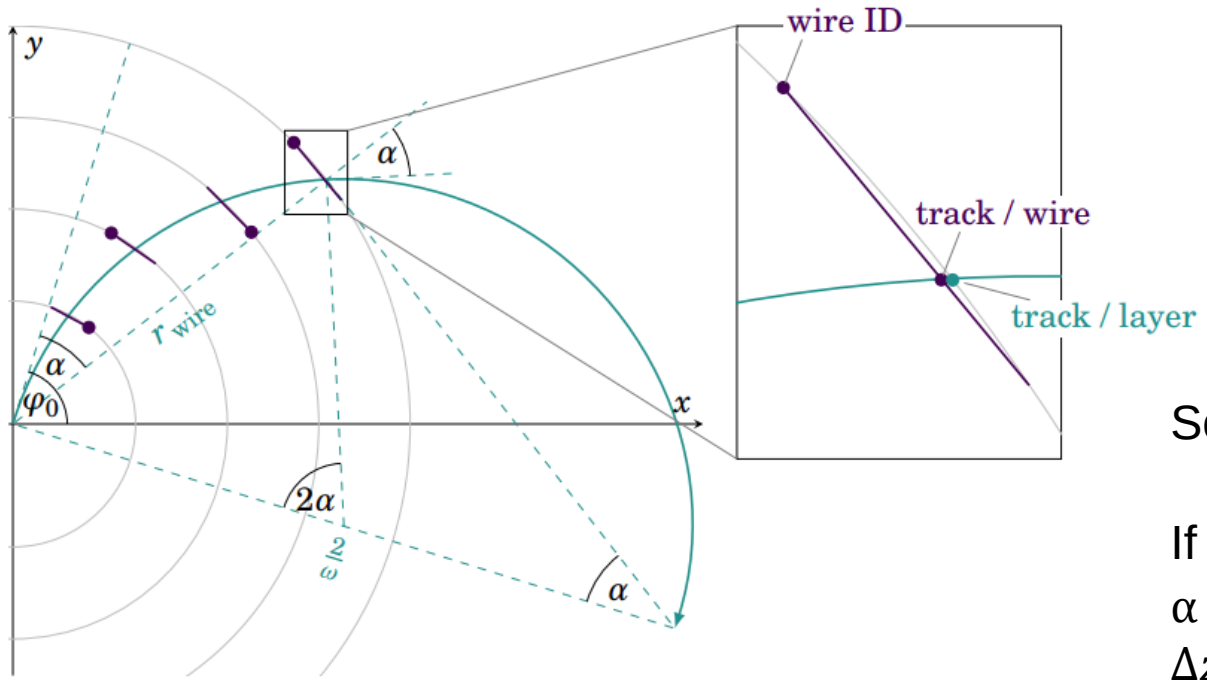
2D Track reconstructed (using hough transformation



3D Neural Network (NN) to calculated z / θ



How to calculate out z0&z0 uncertainty



With direct cross stereo wire:

$$\phi_{cross} \sim \phi_0 - \arcsin\left(\frac{1}{2} r_{wire}\omega\right) \equiv \phi_0 - \alpha(r,\omega)$$

$$\frac{z_{cross} - z_B}{Z_F - Z_B} = \frac{\phi_{cross} - \phi_B}{\phi_F - \phi_B}$$

$$z_0 = z_{cross} - \cot\theta_0 \frac{2\alpha}{\omega}$$

Drift time would influence:

$$\phi_{hit} = \phi_{wire} \pm \arcsin\left(\frac{v_{drift} t_{drift} cos\alpha}{r_{wire}}\right)$$

$$r_{hit} = r_{wire} \pm v_{drift} t_{drift} sin\alpha$$

So the δt_{drift} would influence ϕ_{cross} and r_{wire}

If we ignore r_{wire} comparing with δt_{drift}, (with small α and large P_t)
Δz_0 could be consist of Δz_{cross} (from 3D Fitter /NN)
And $\Delta(\cot\theta_0 \frac{2}{\omega})$ (From 2D track)

And: $$\Delta z_{cross} = \frac{r_{wire}}{\sin\psi}\sqrt{(\Delta\phi_{cross})^2 + (\Delta\phi_B)^2}$$

Still, ignore r_{wire} comparing with δt_{drift},

$$\Delta\phi_{cross} \times \sim 0.03° - 0.08° \ (varied\ from\ r_{wire})$$

$$\Delta\phi_B \sim \frac{v_{drfit} cos\alpha}{r_{wire}} \Delta t_{drift}$$

MC Test

MC :

Train Sample

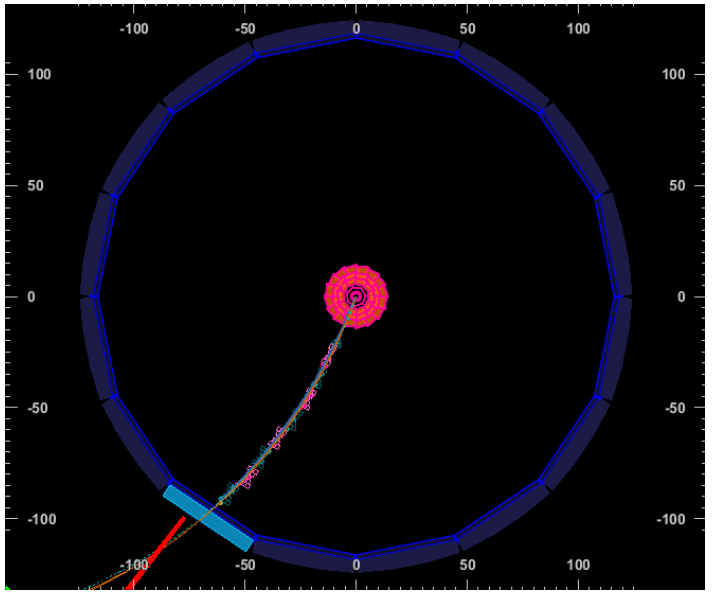
- **Particle gun:**
- **muons; single tracks;**
- **Pt :[0.3 GeV,3 GeV], uniform;**
- **Φ: [0, 360],uniform;**
- **θ: [0,170], uniform;**
- **Vertex z0: [-50, 50], uniform;**
- **N events: 300k**

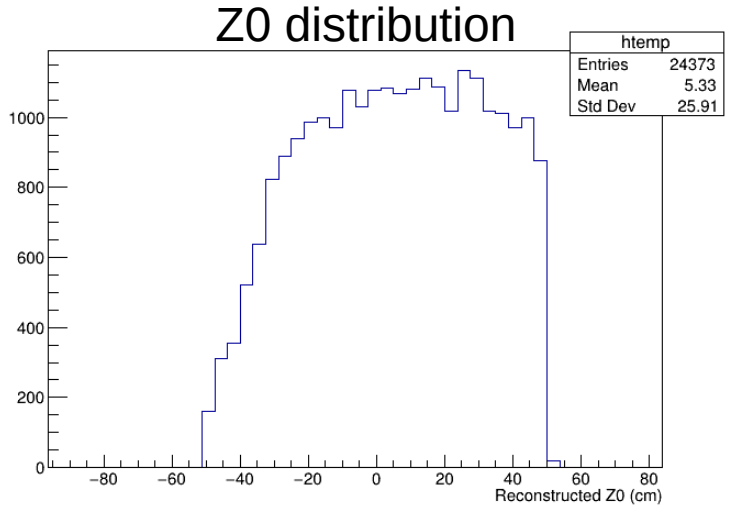
Validation Sample:

- **Same config;**
- **N events: 20k**

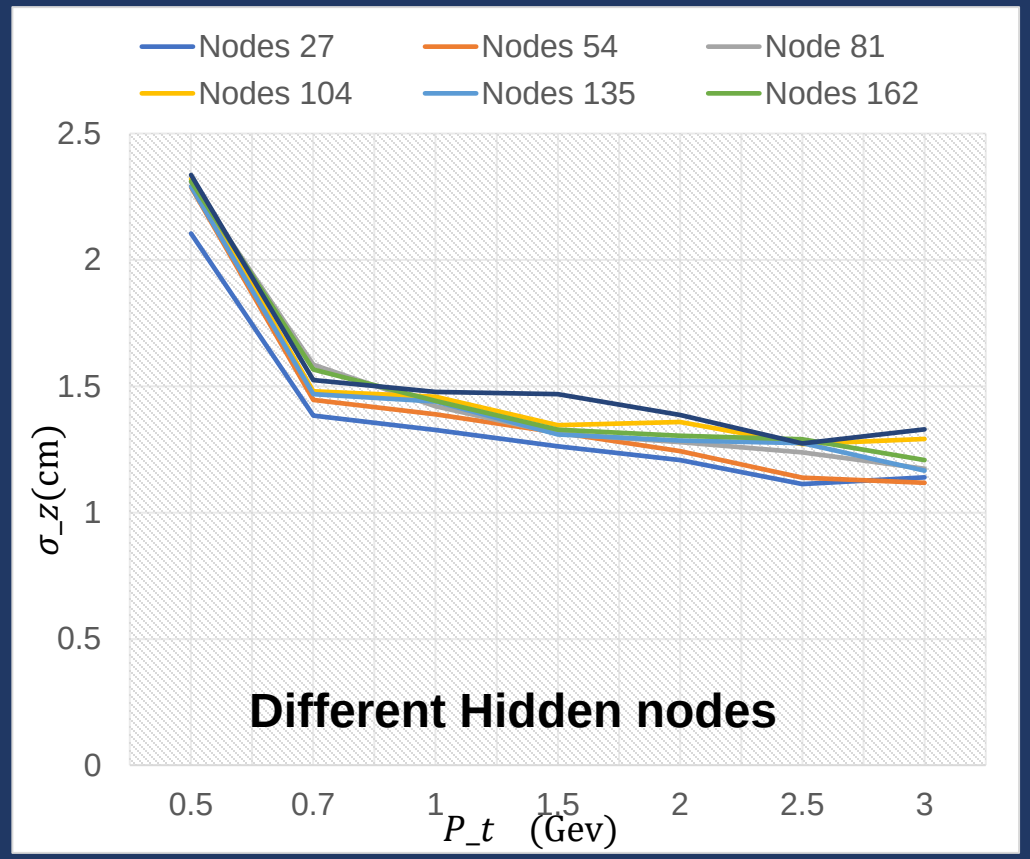
Test Sample:

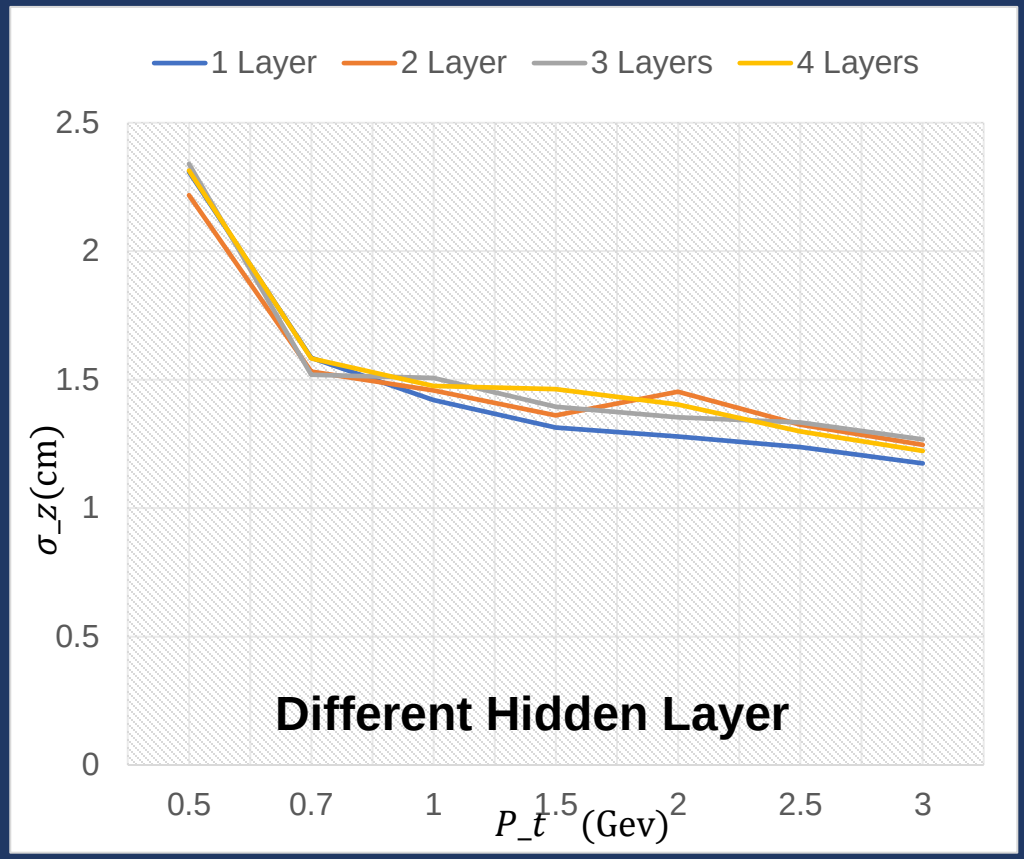
- **Same config;**
- **N events: 50k**





Hidden Layer



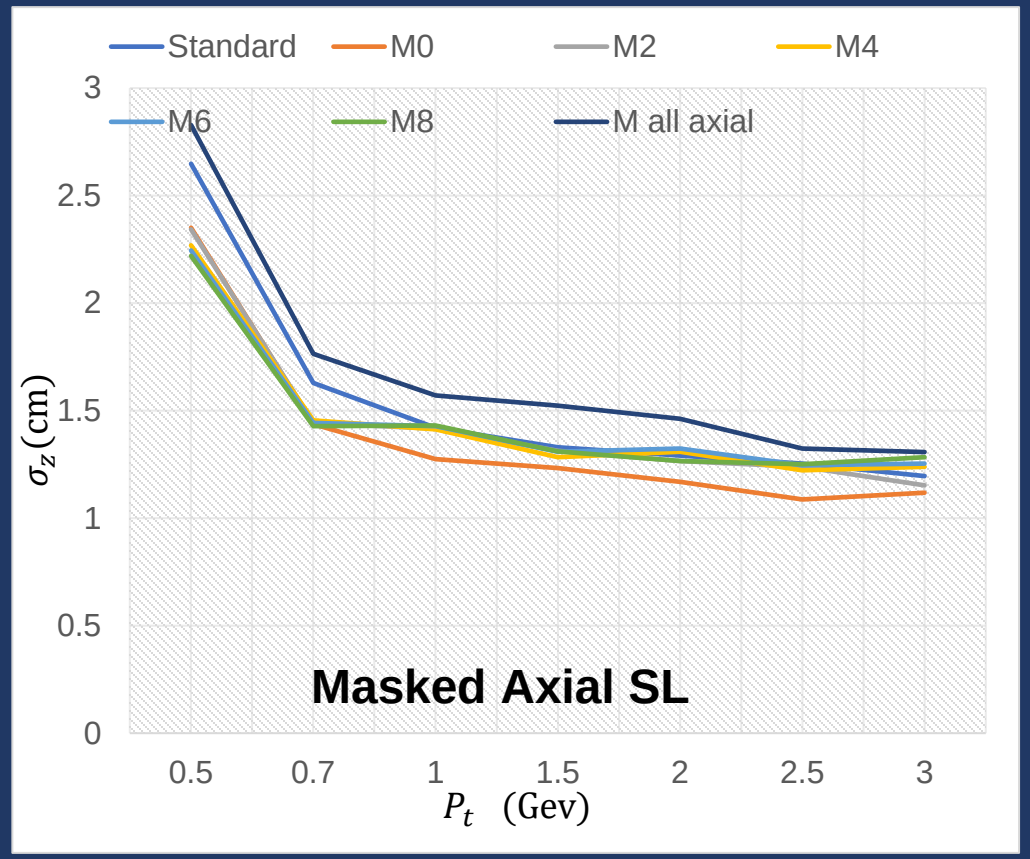


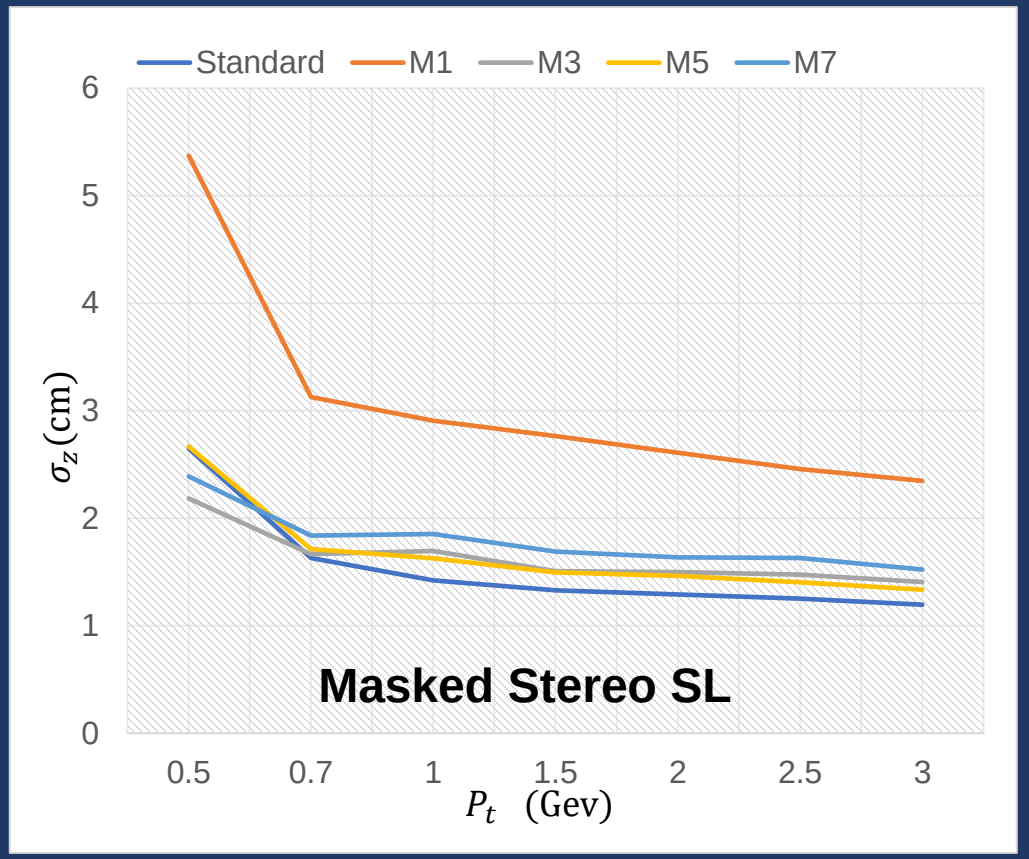
Trained with MC, event t0 =0.

Different hidden Layers / nodes do not make large difference in standard model

Add more wires do not induce other relationship, keep hidden layer as before.

Masked Super Layer

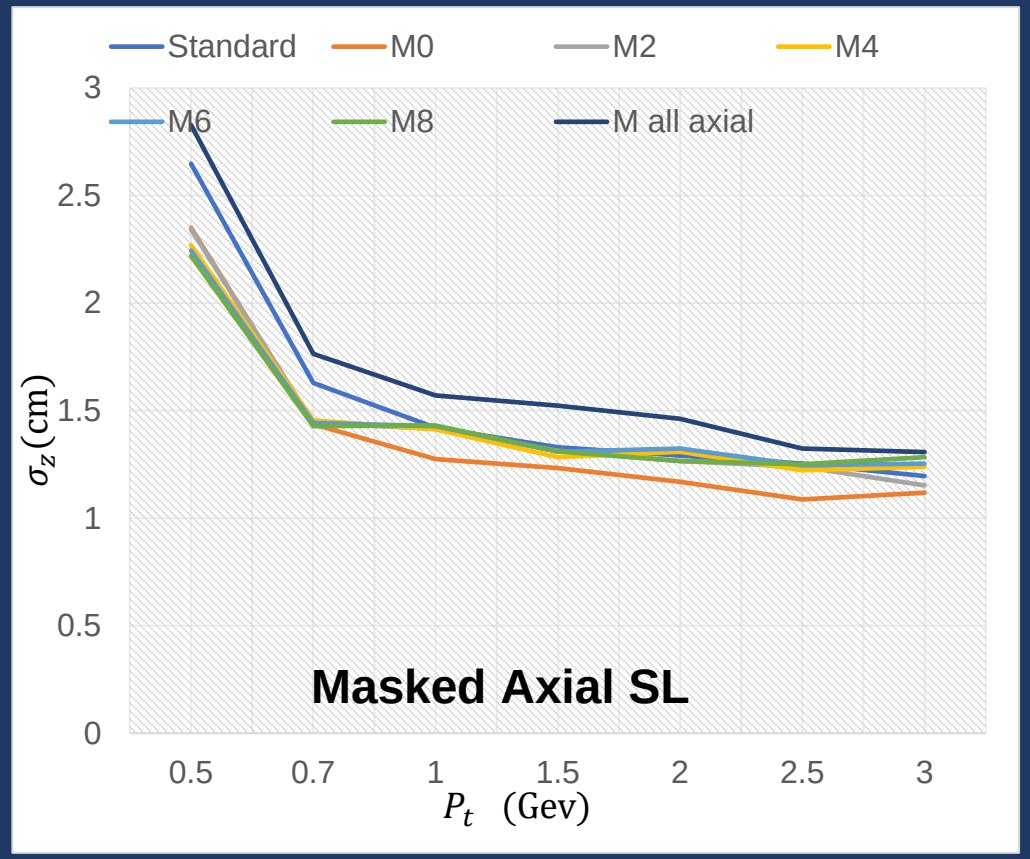


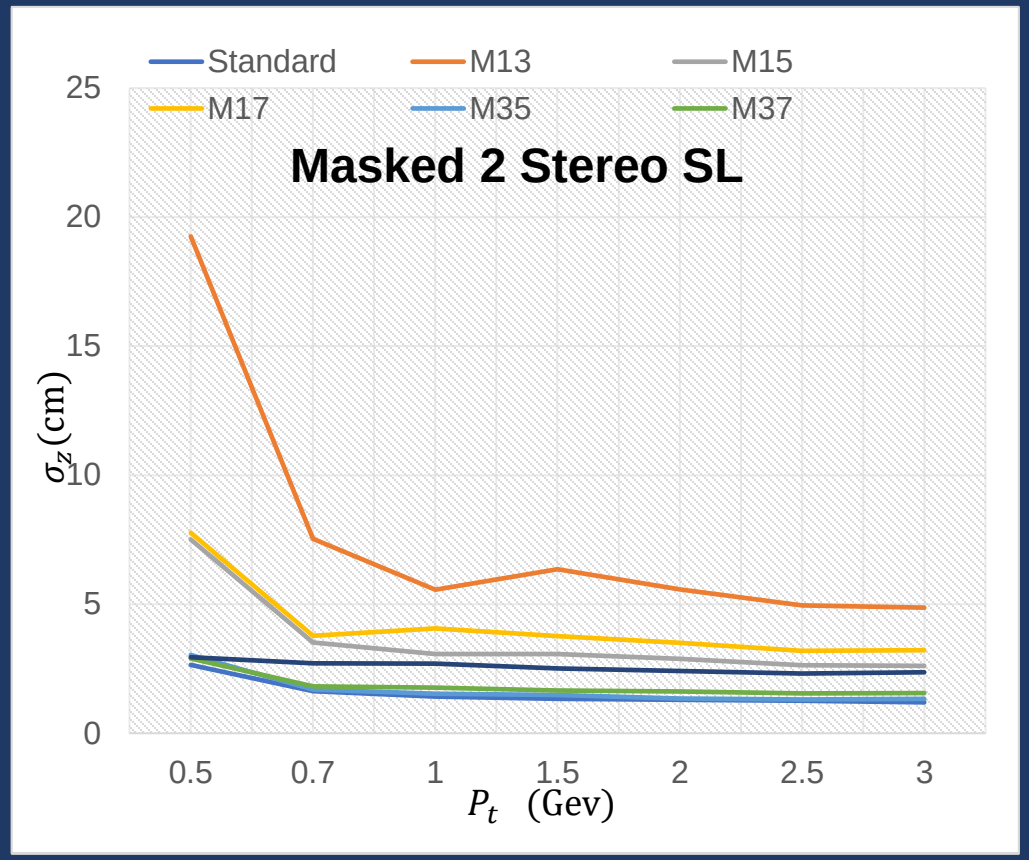


To see the importance of each super layer, masked each one for the training & testing for NN.

Axial layer contribute little to the NN, even masked all, resolution decrease little

Masked Super Layer

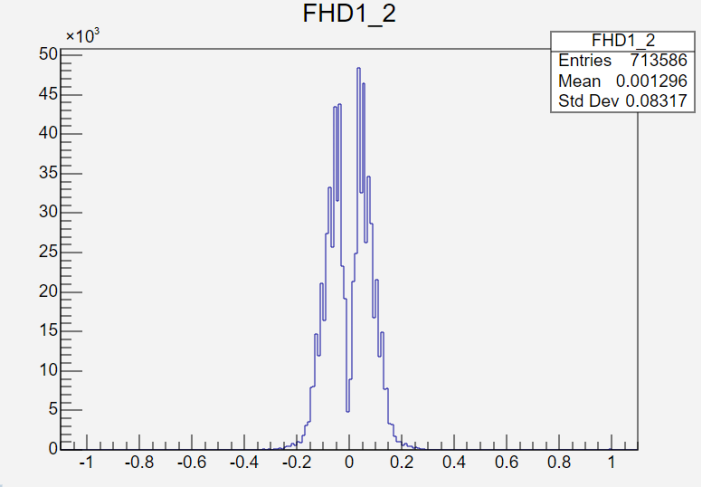




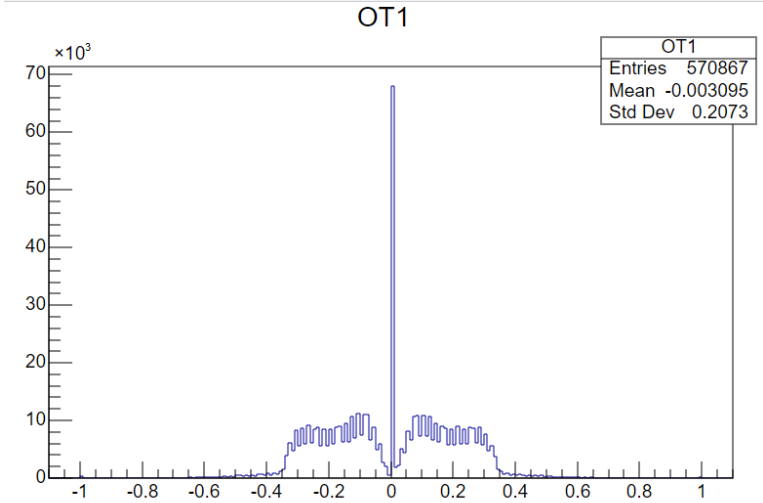
To see the importance of each super layer, masked each one for the training & testing for NN.

Axial layer contribute little to the NN, even masked all, resolution decrease little

Input Parameters

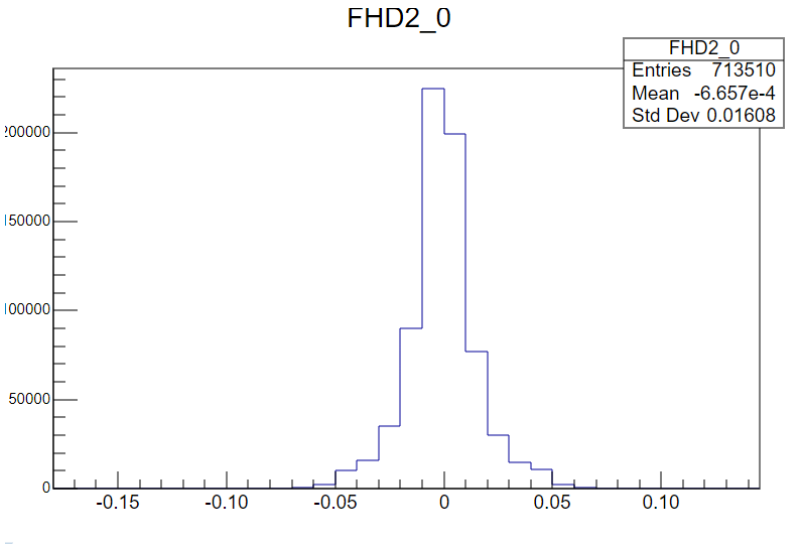


Extra T

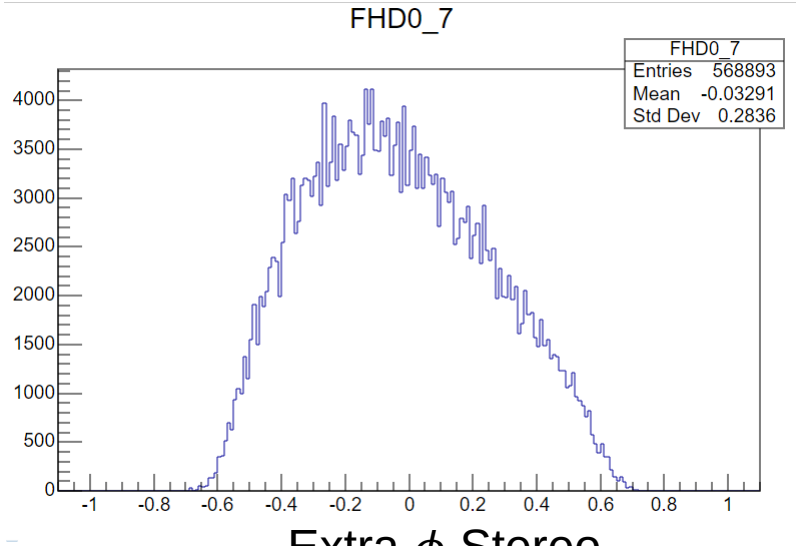


Origin T

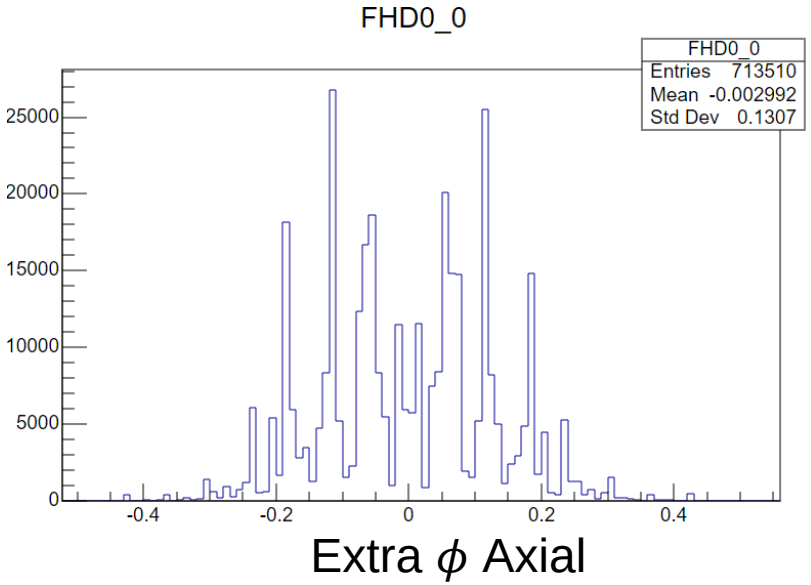
For exp26run1771



Extra α



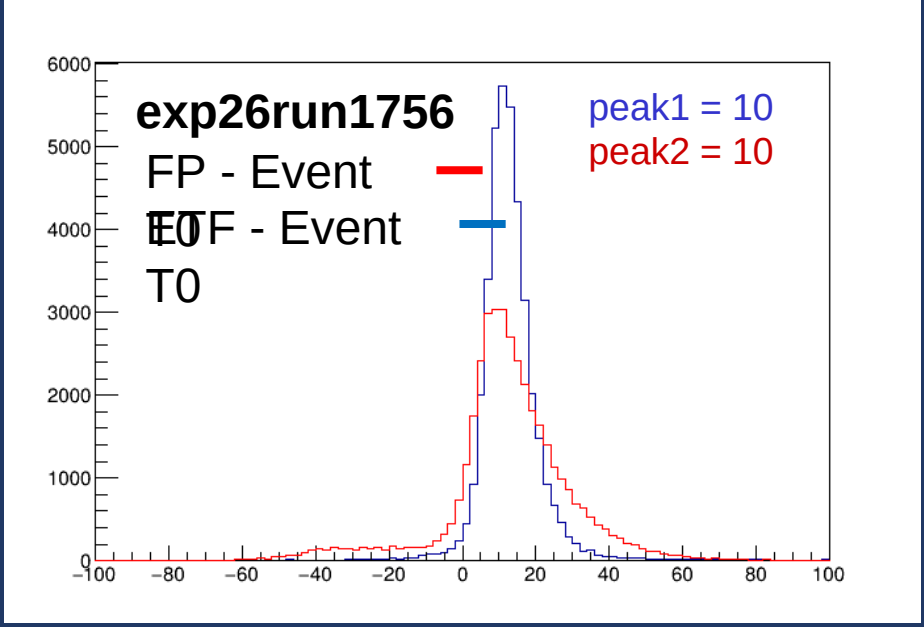
Extra ϕ Stereo



Extra ϕ Axial

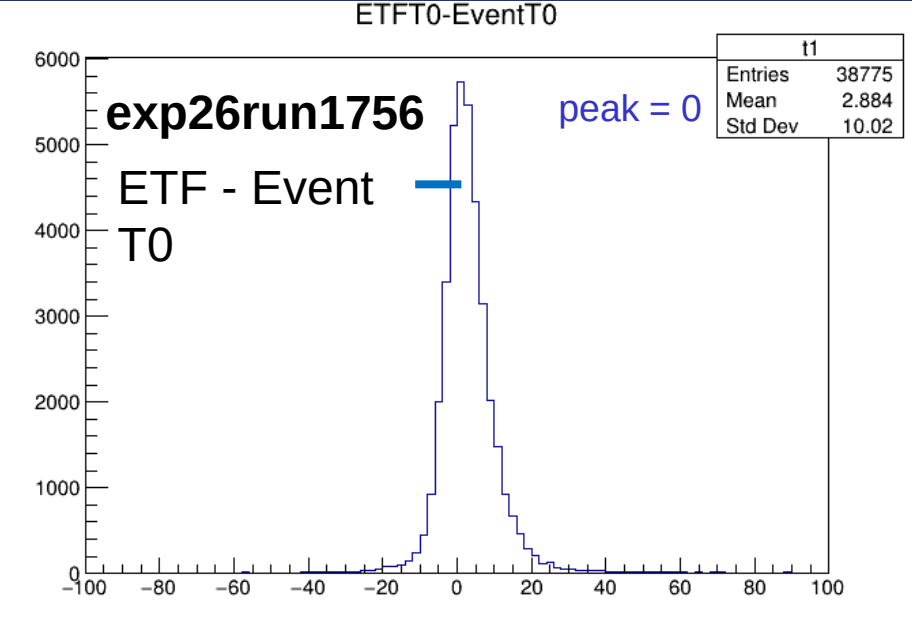
ETF-offset

```
addParam("offset", m_offset,
         "Set certain time offset for ETFHough simulation"
         "Default as 0", 0);
```



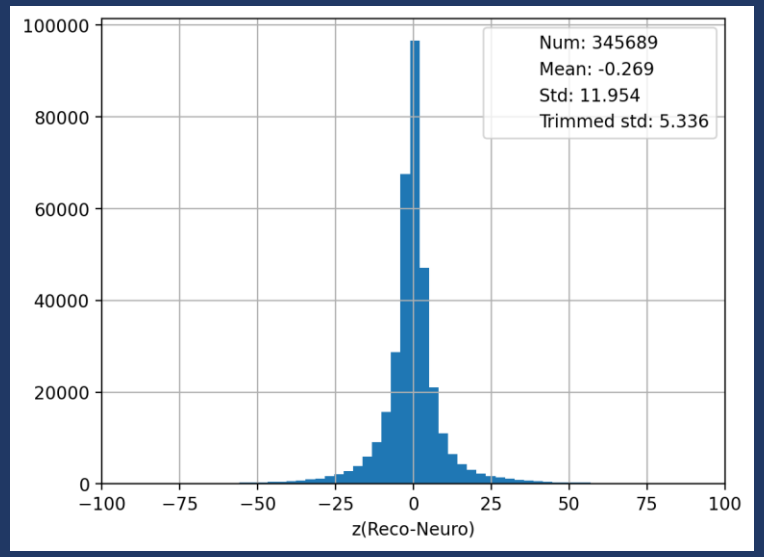
After

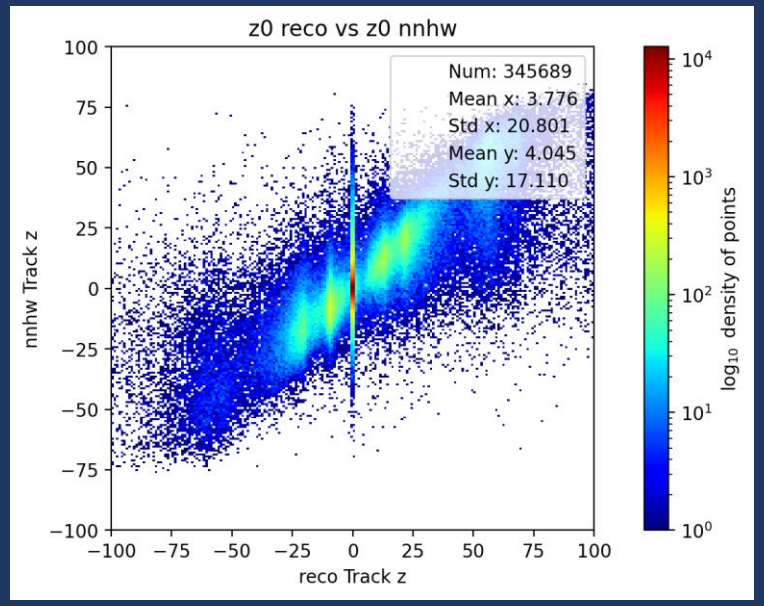
Offset = -10 ns



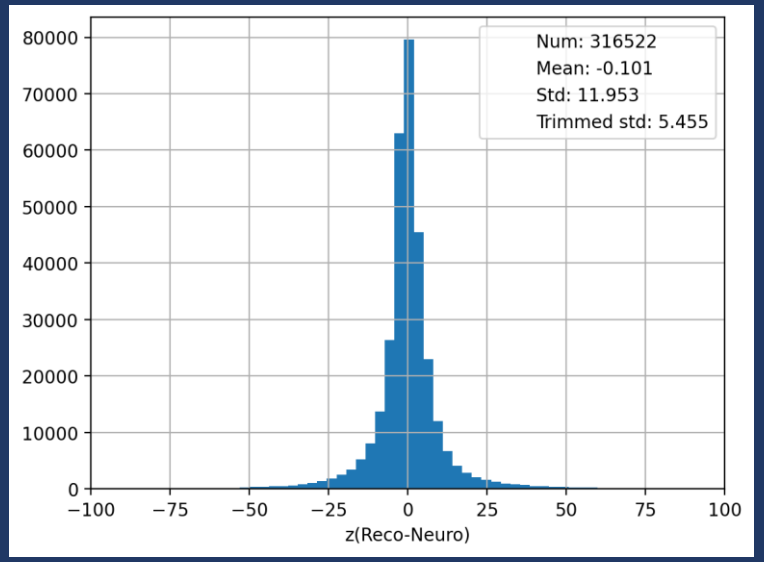
After ETF offset 1 wires expert 0

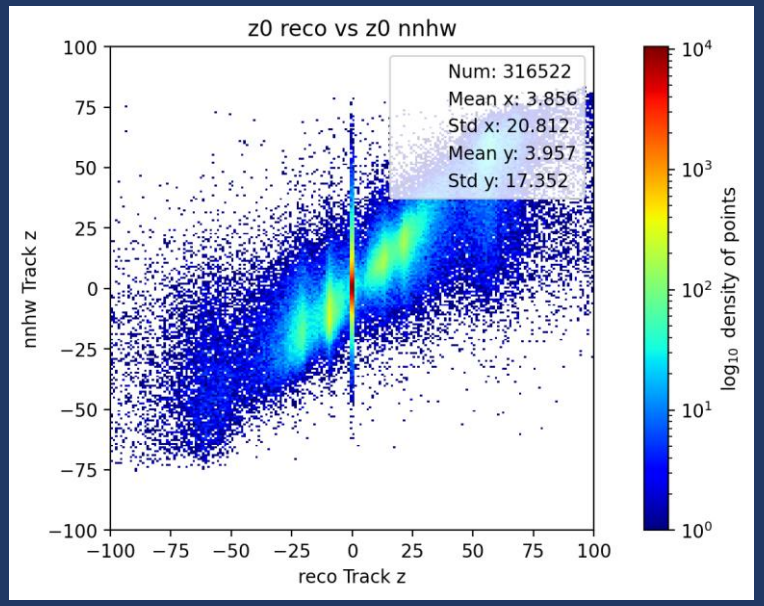
Before offset





After offset





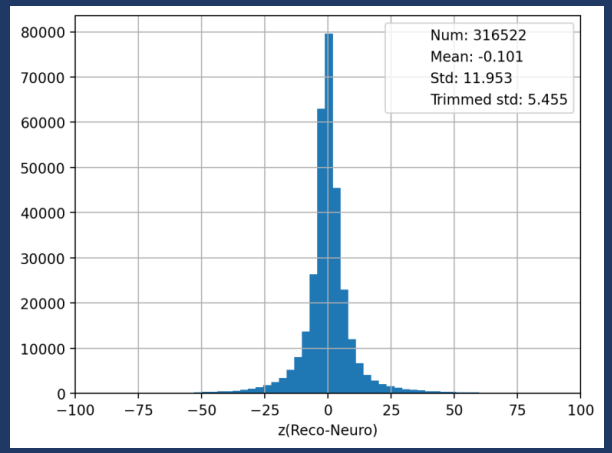
No difference
--As expected: NN could learn the offset

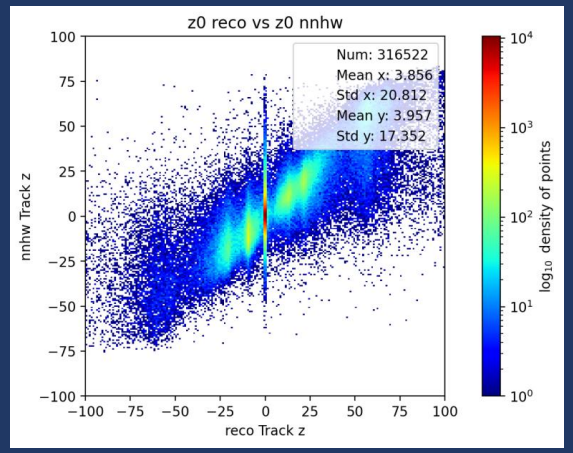
Step learning rate?

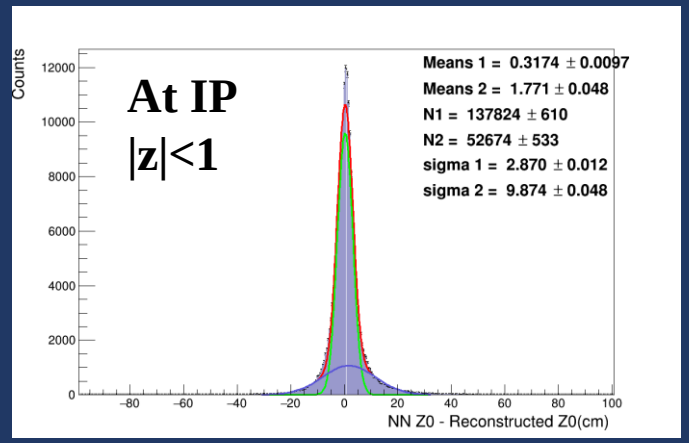
Question: Training error will start to oscillate after a few hundreds epoch → try to adjust learning rate to improve it more deeply.

First attempt: learning rate * 0.2 at every 200 epoch

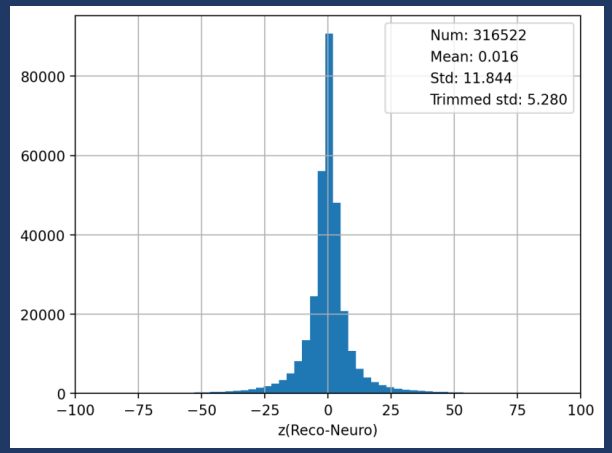
before

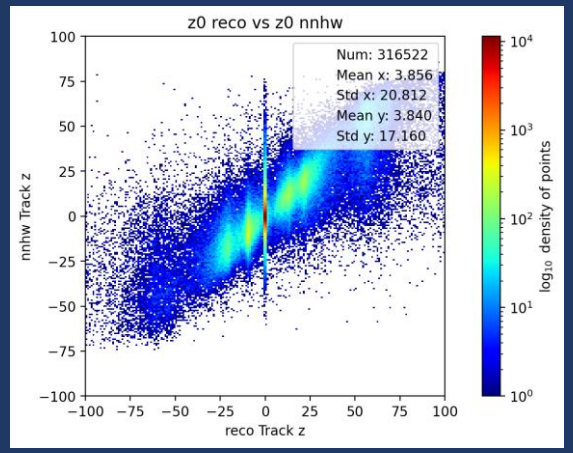


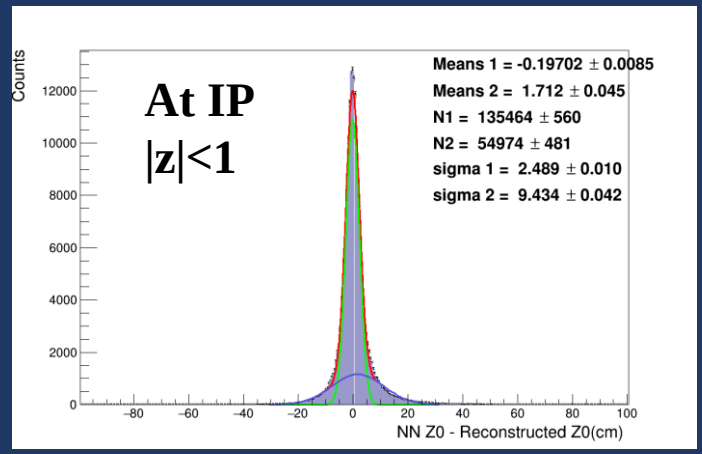




After

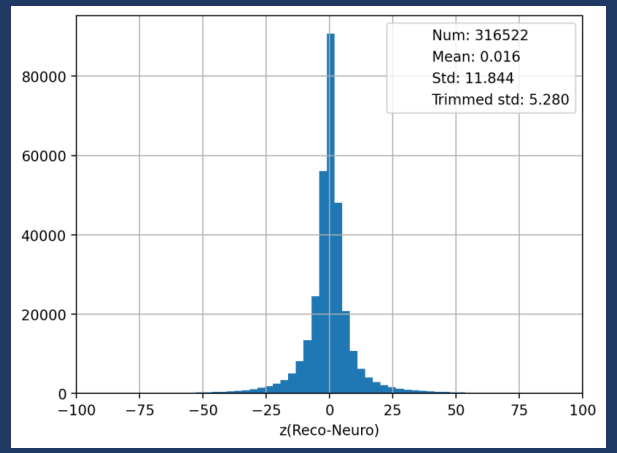


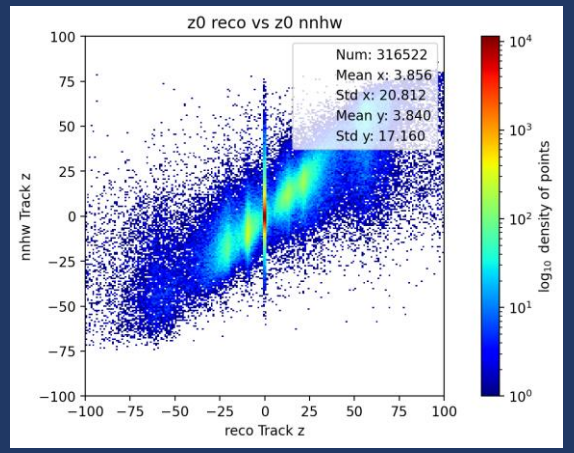


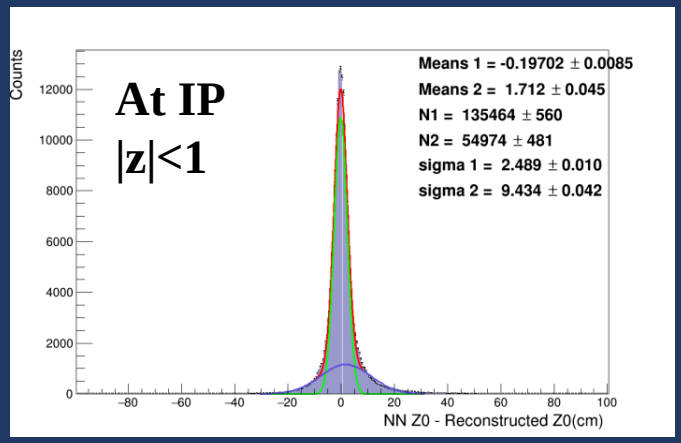


More wires?

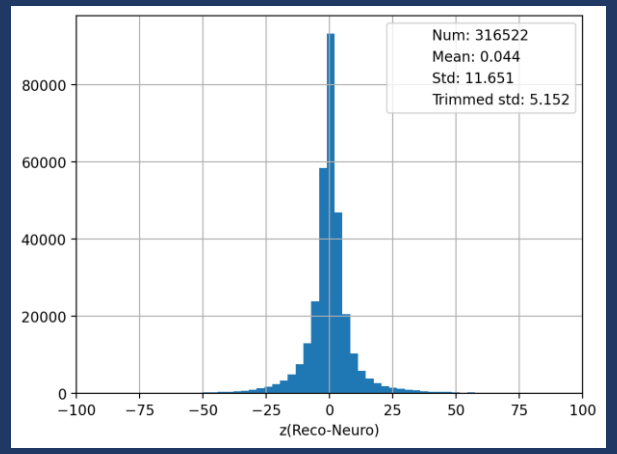
Wire 1

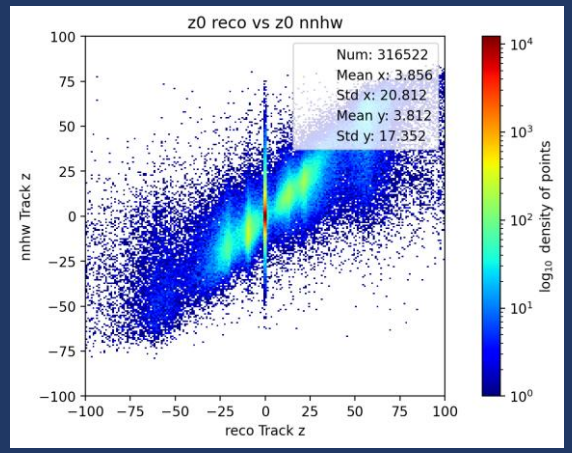


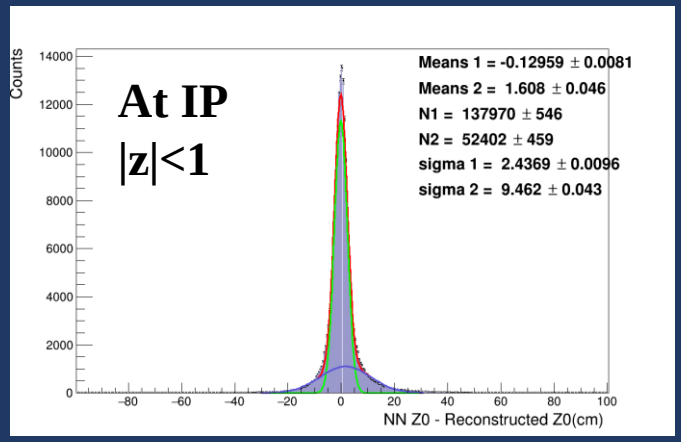




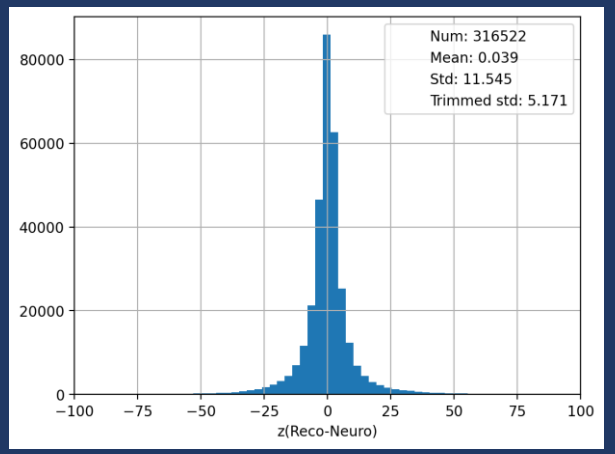
Wire 2

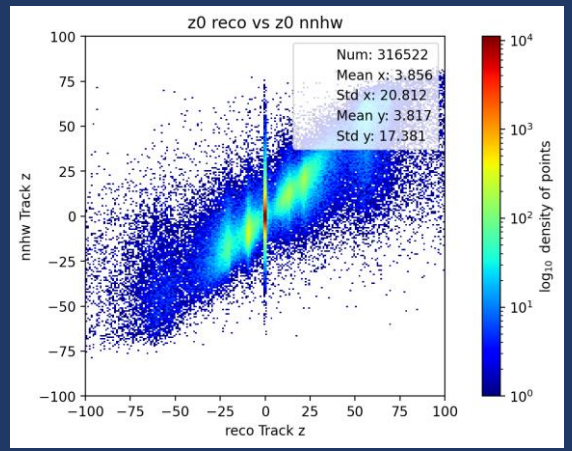


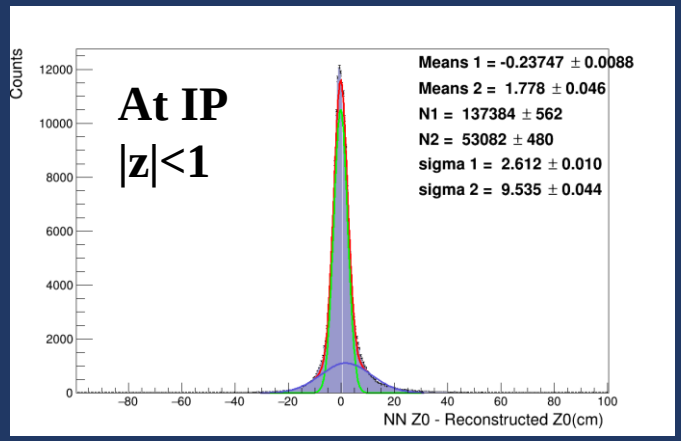




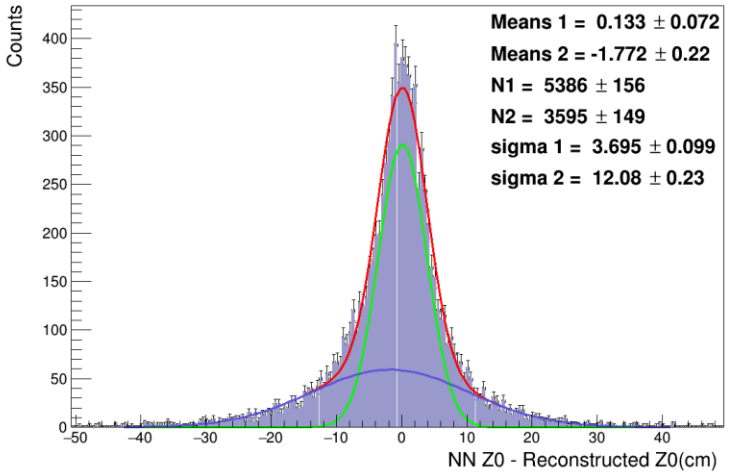
Wire 3

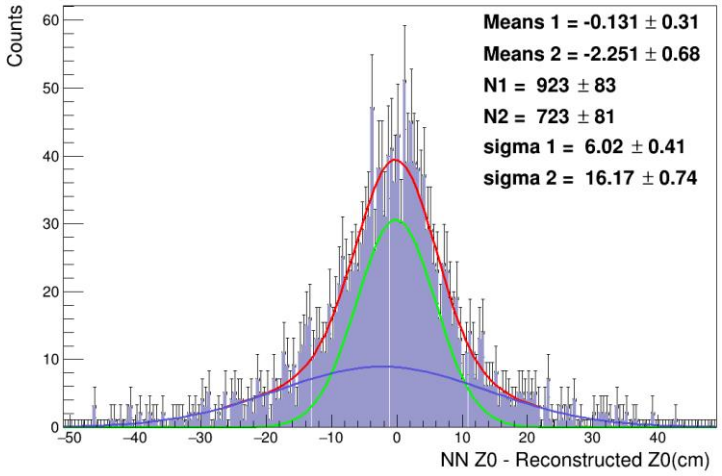


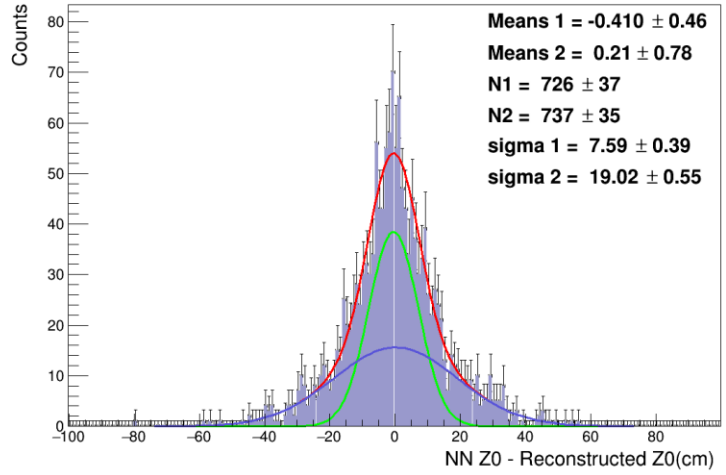




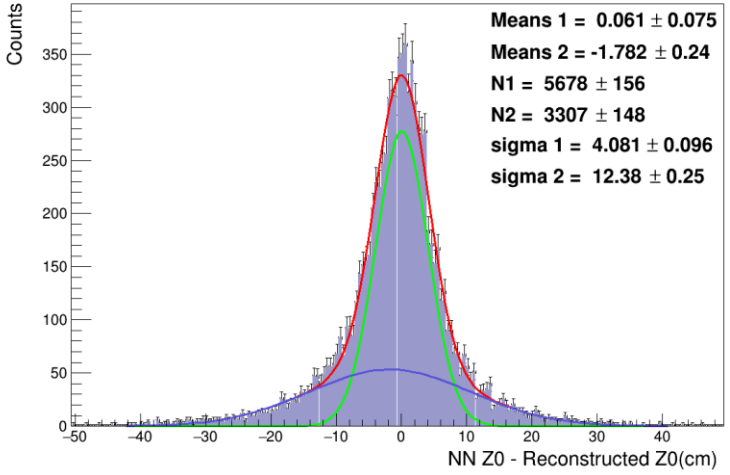
Detail result compare with z0 -- OLD FANN

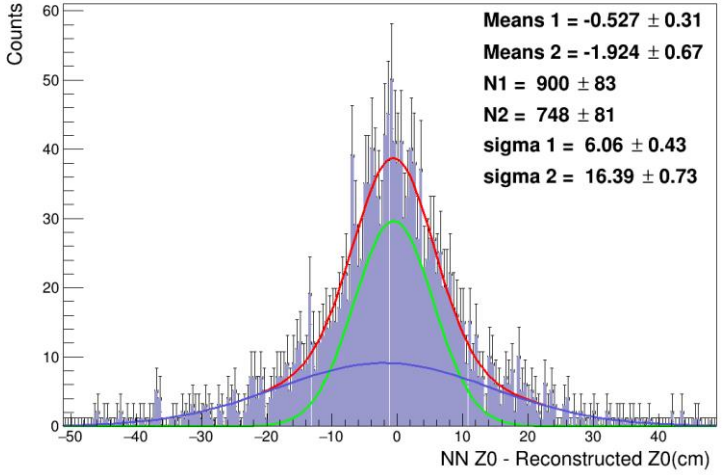


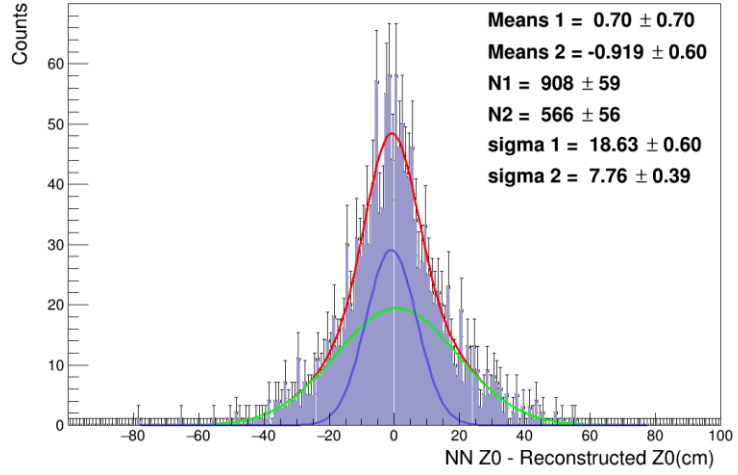




Wire1







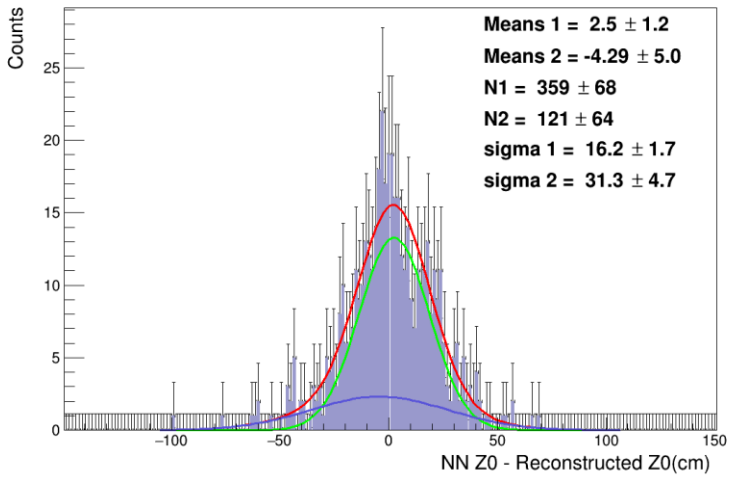
Standard

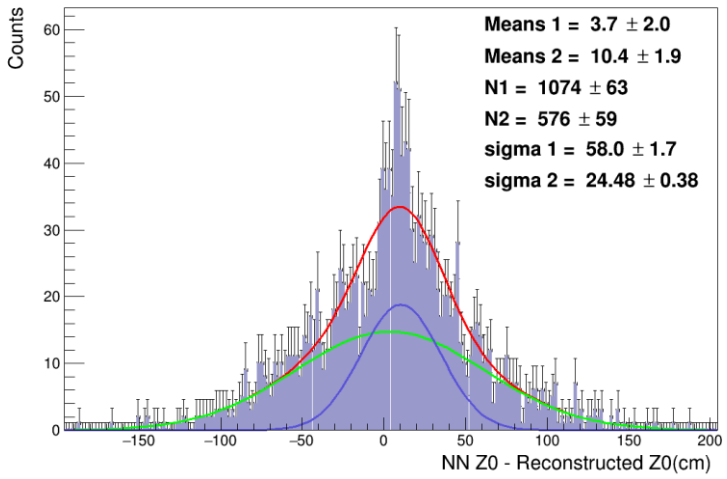
0-10

10-20

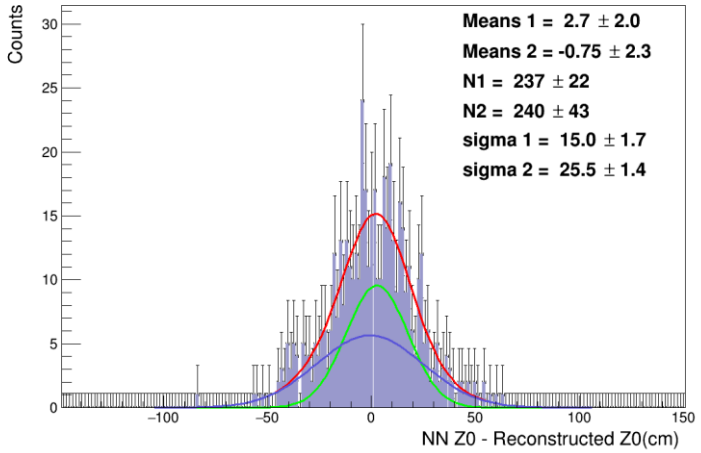
20-30

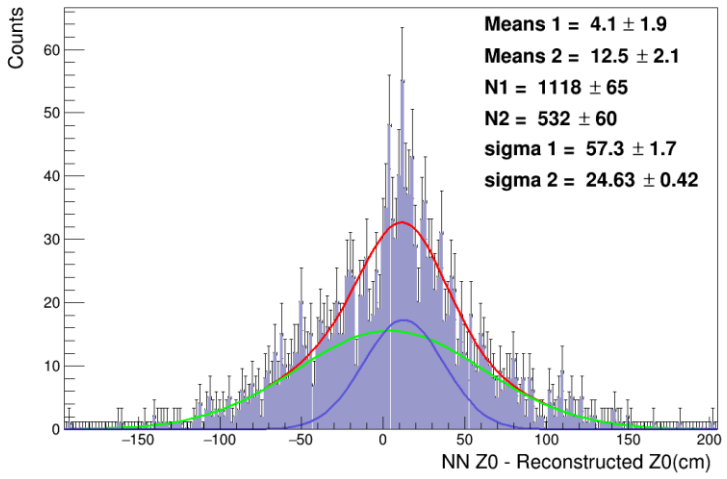
Detail result compare with z0-- OLD FANN





Wire1





Standard

30-40

40+

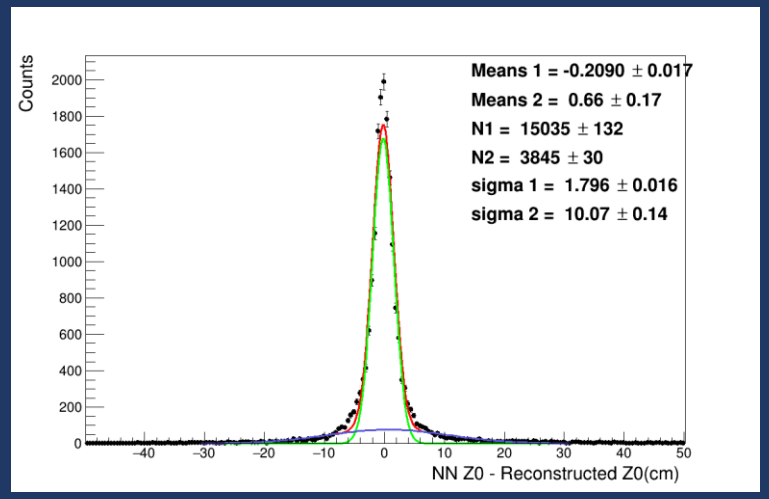
Real data test –Train with exp24 run 2004

Train NN with exp24 run2004 and exp26r1968

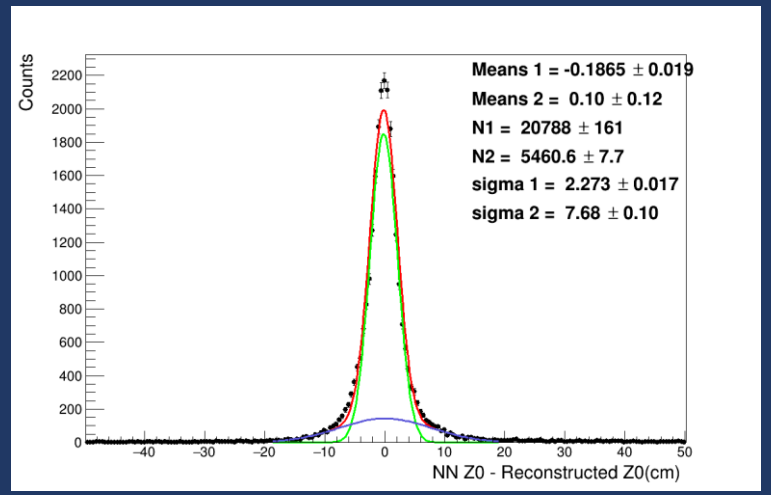
Test with exp24 run2004(sorry not unpacked another one for test)
And exp26 run 1777 (which use for trigger study with z0 in any range)

exp24 run2004

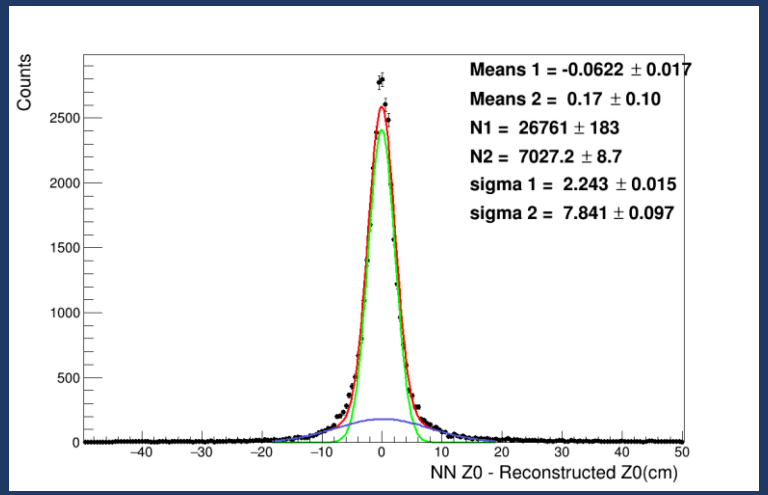
Origin with ETFHough



Wire 1 with ETFHough



Wire 2 with ETFHough



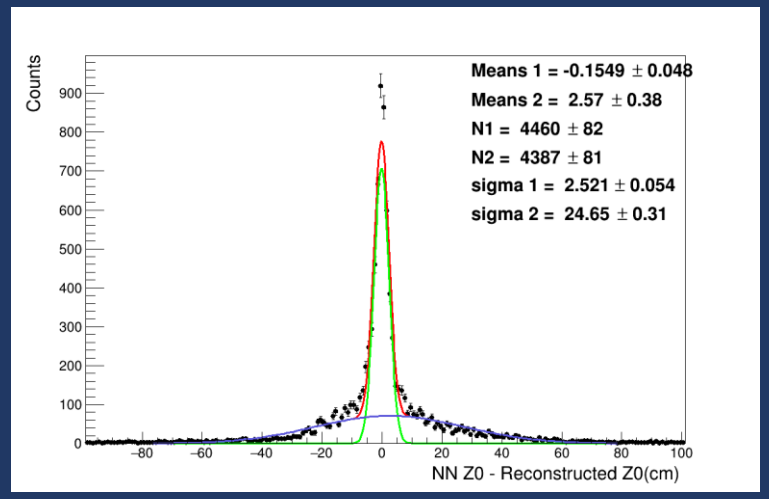
Real data test –Train with exp24 run 2004

Train NN with exp24 run2004 and exp26r1968

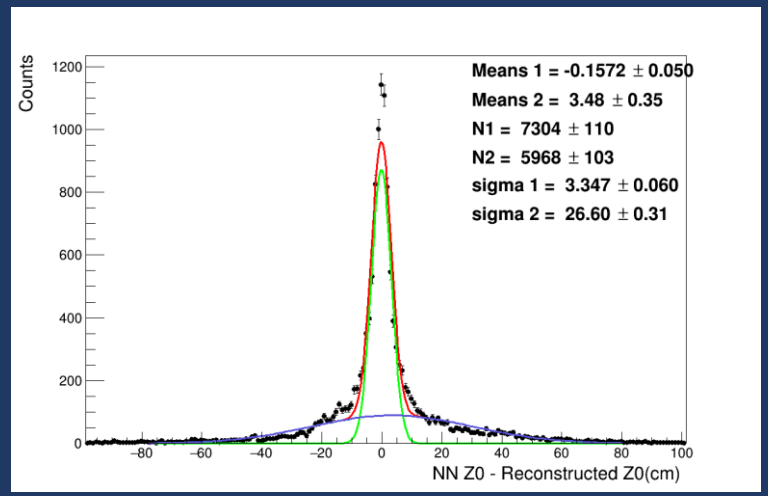
Test with exp24 run2004(sorry not unpacked another one for test)
And exp26 run 1777 (which use for trigger study with z0 in any range)

exp26 run 1777

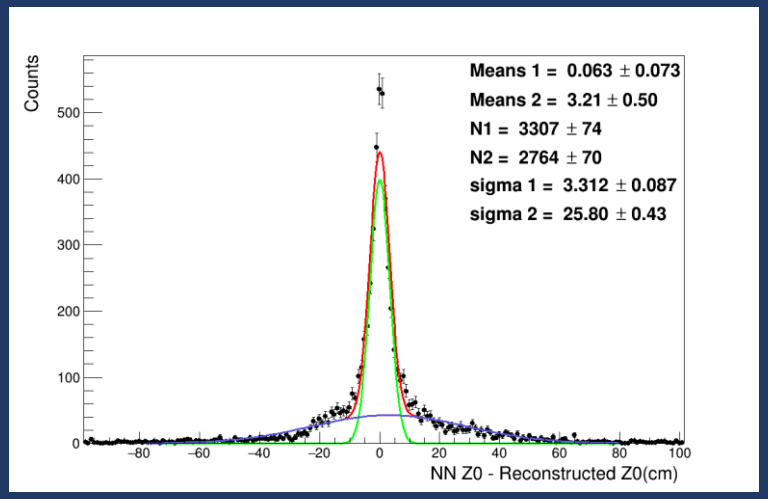
Origin with ETFHough



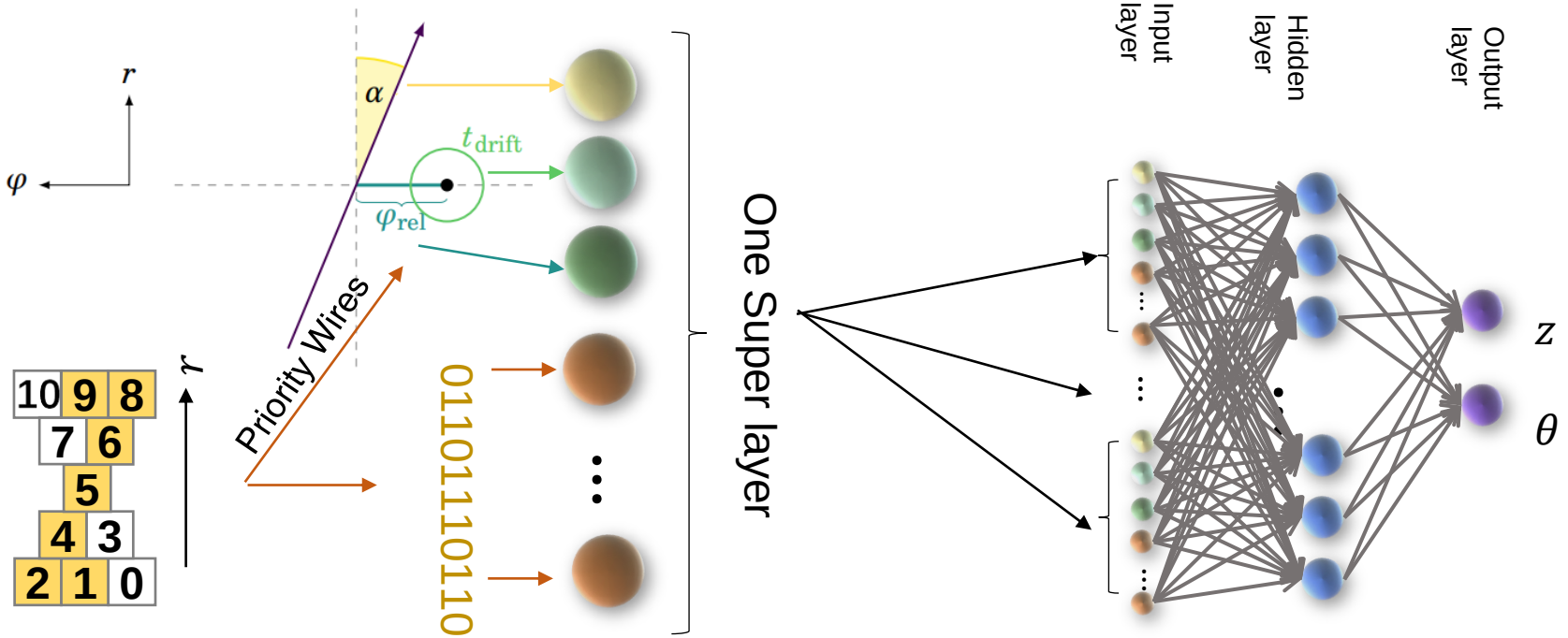
Wire 1 with ETFHough



Wire 2 with ETFHough



First attempt: Directly use TS pattern as input



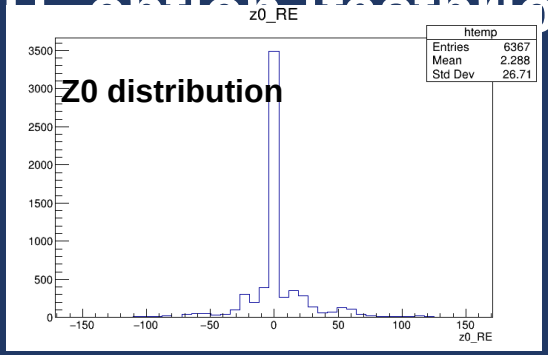
Directly use 11/15 bits pattern as input

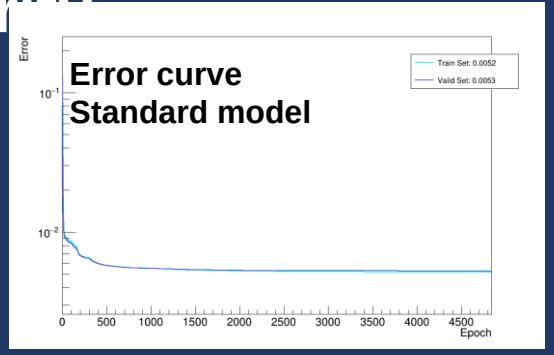
Since L/R information are got from pattern, hoping could replace L/R with it

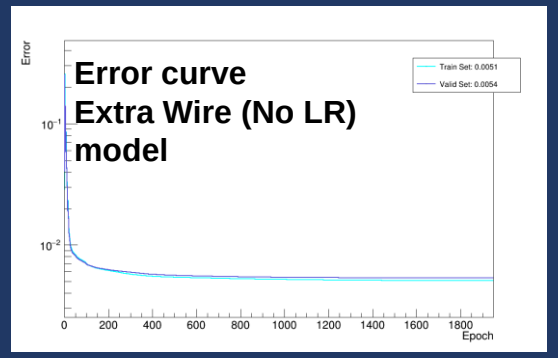
Train with real data

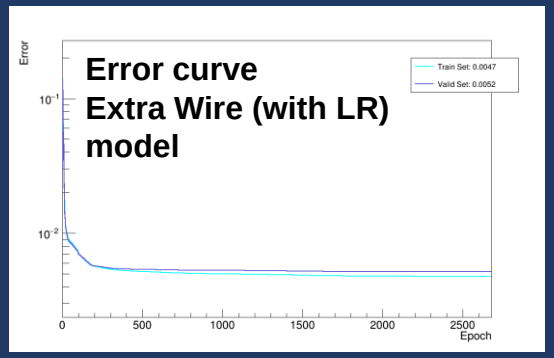
Train Standard model and one extra wire model (with/without LR) with exp26 run1771 & exp26 run1762; beam-reco-monitor.

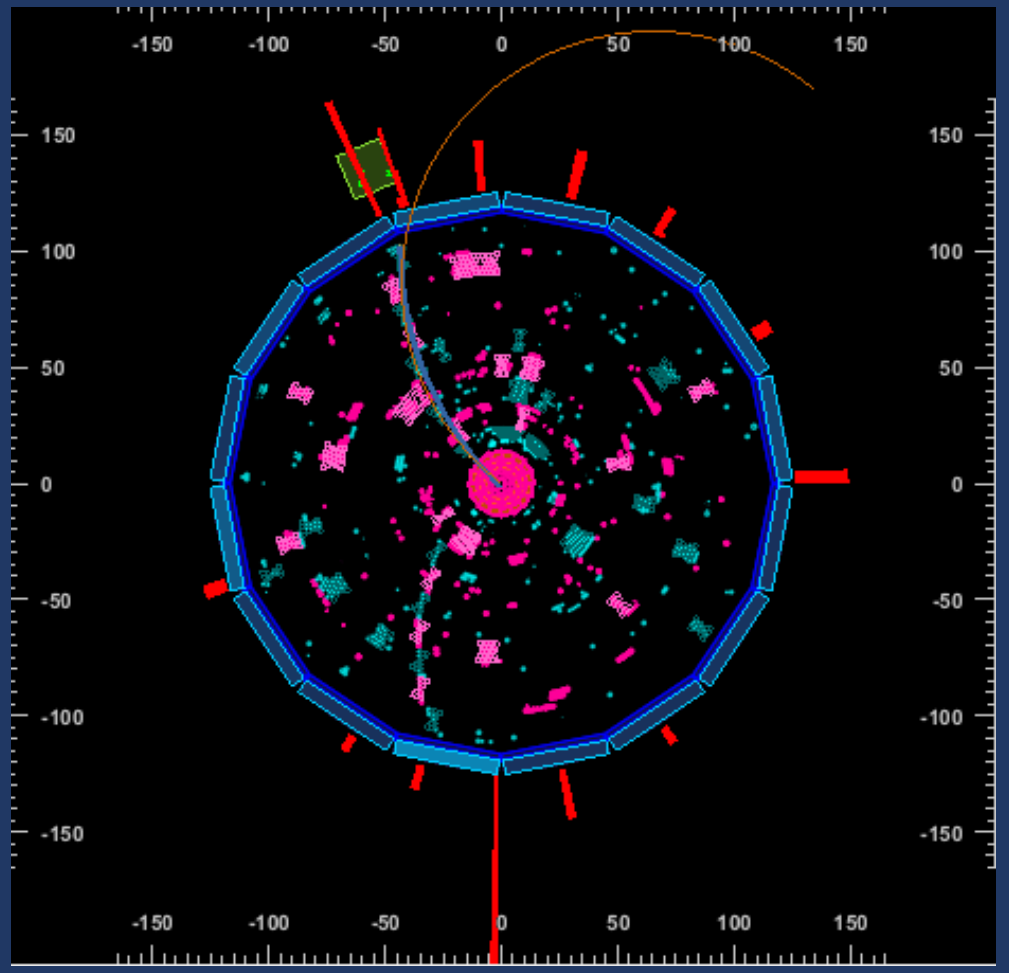
ETF option: footprint





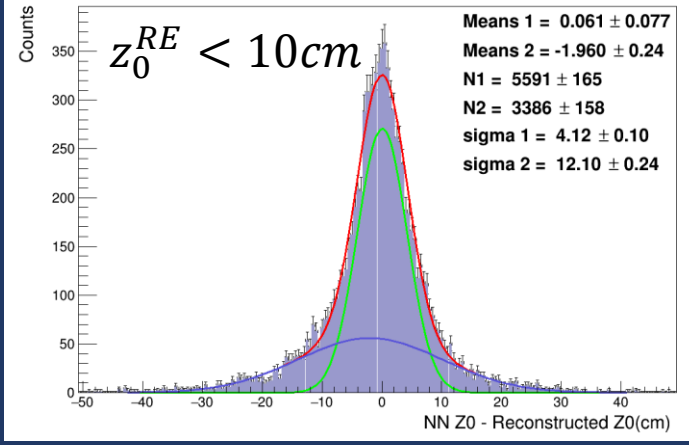


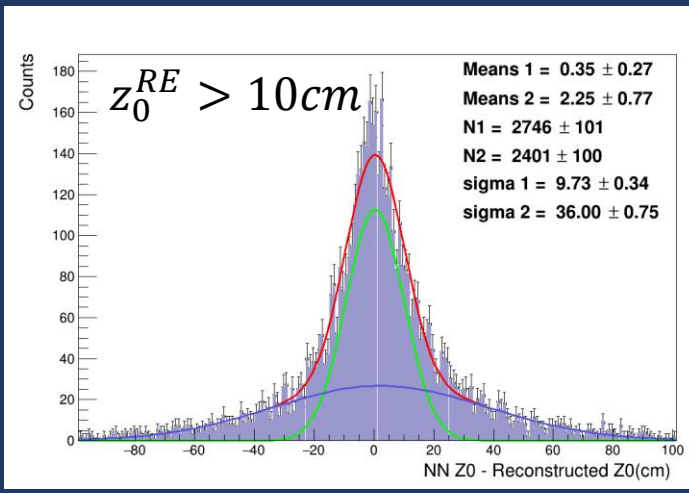




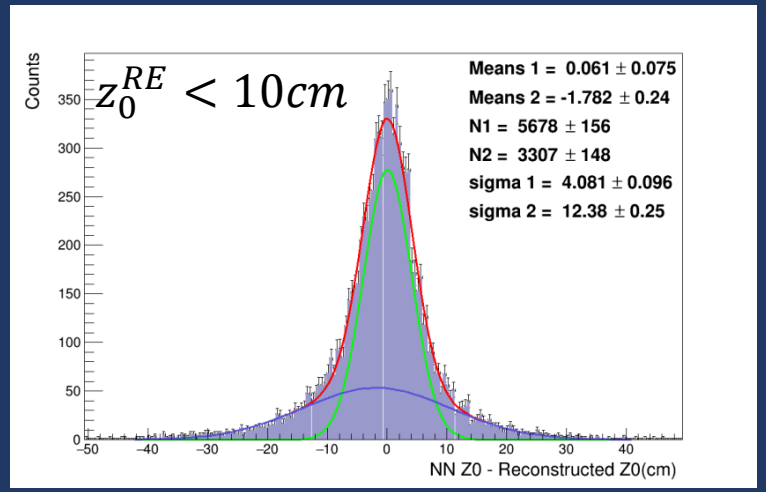
Test with real data exp26 run1771

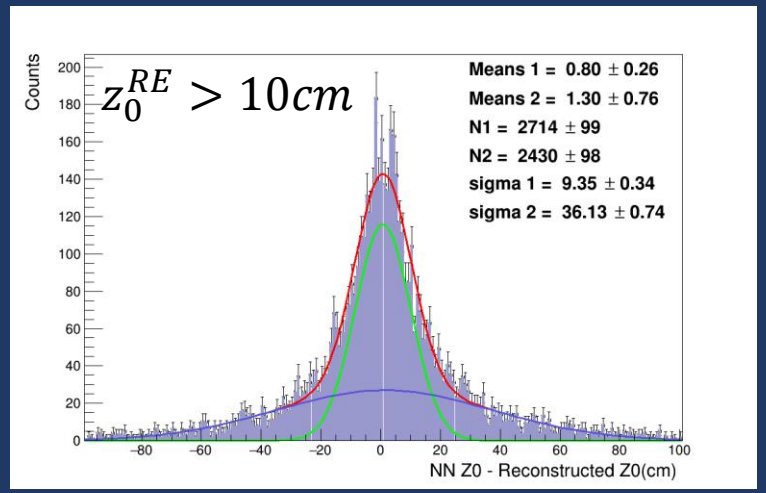
Standard Model



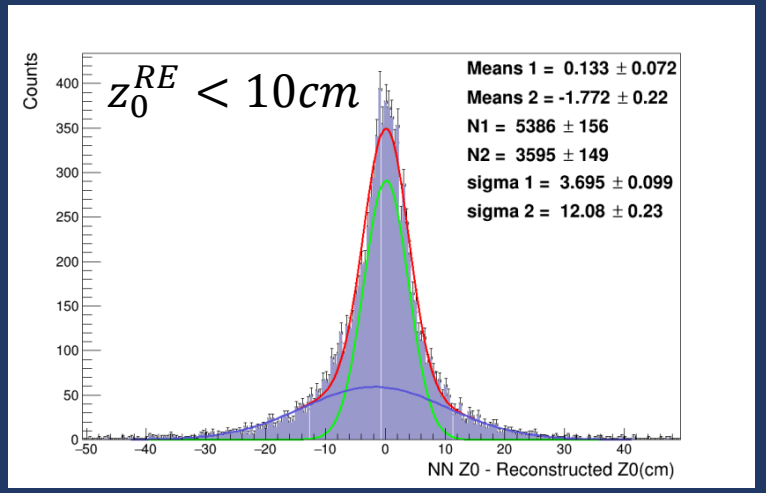


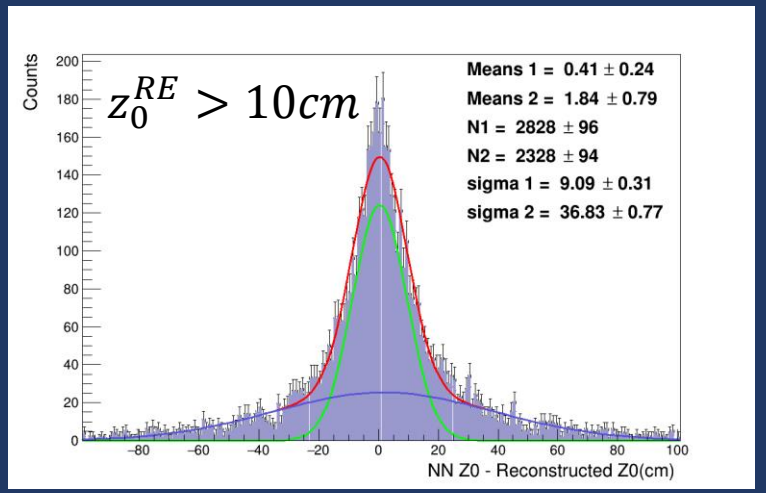
Extra Wire 1 No L/R





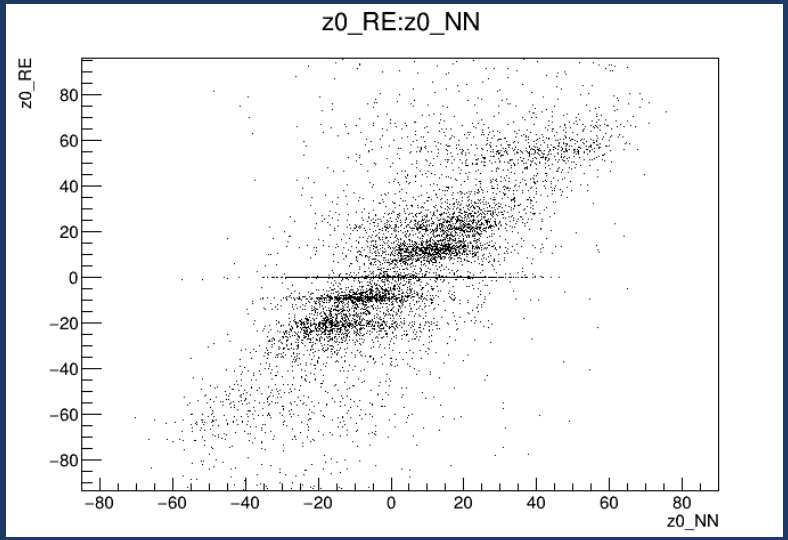
Extra Wire 1 with L/R



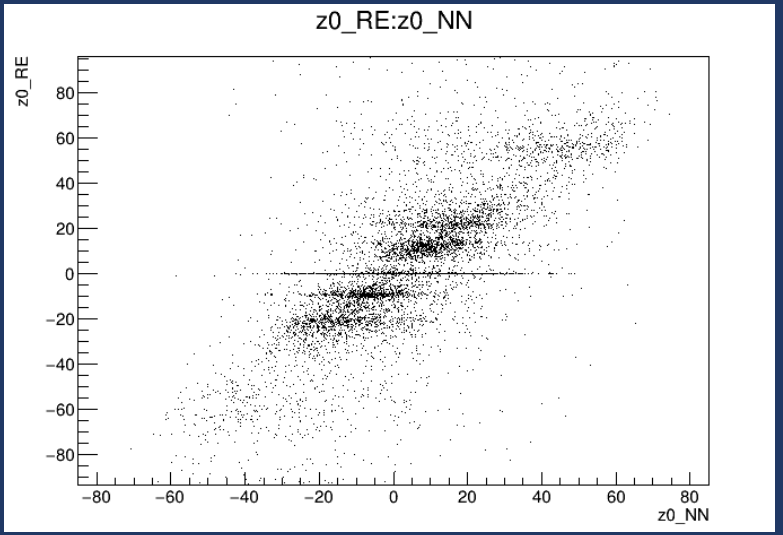


Test with real data exp26 run1771

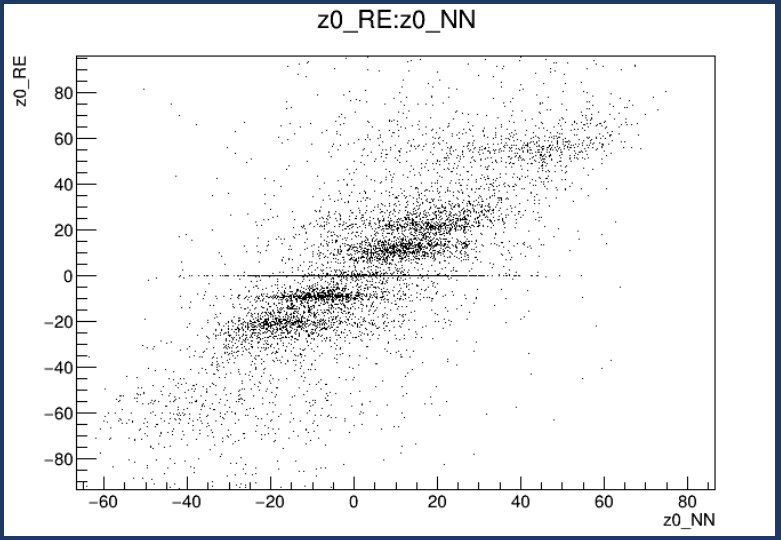
Standard Model



Extra Wire 1 No L/R



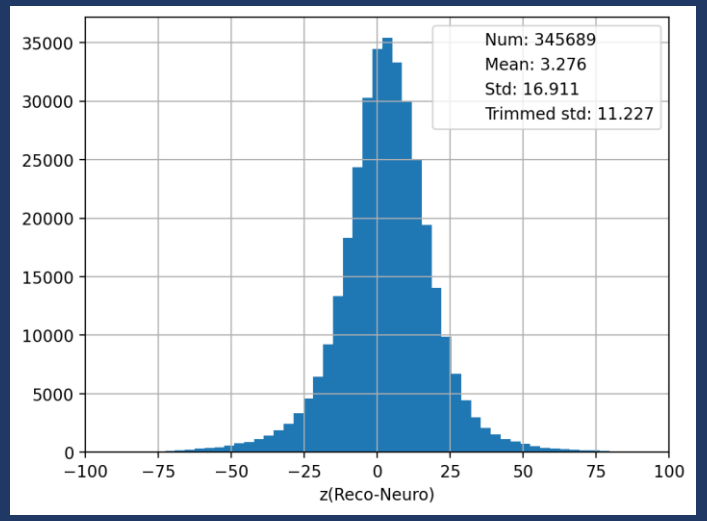
Extra Wire 1 with L/R

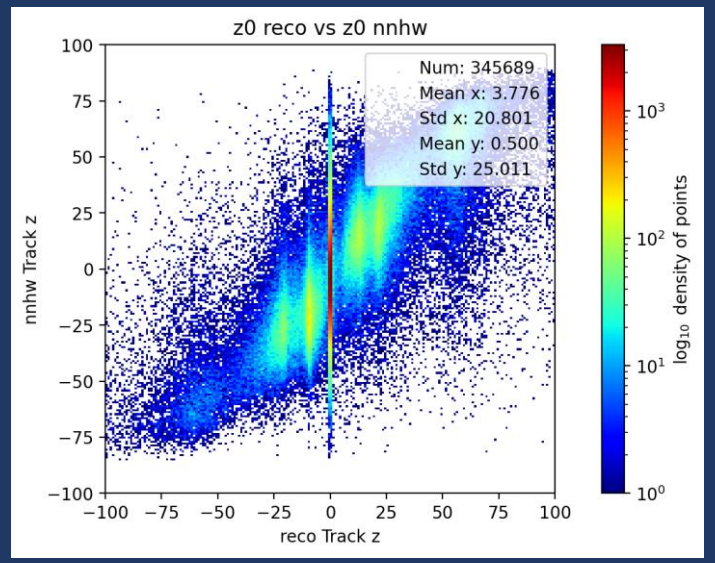


Training still need to be improved.

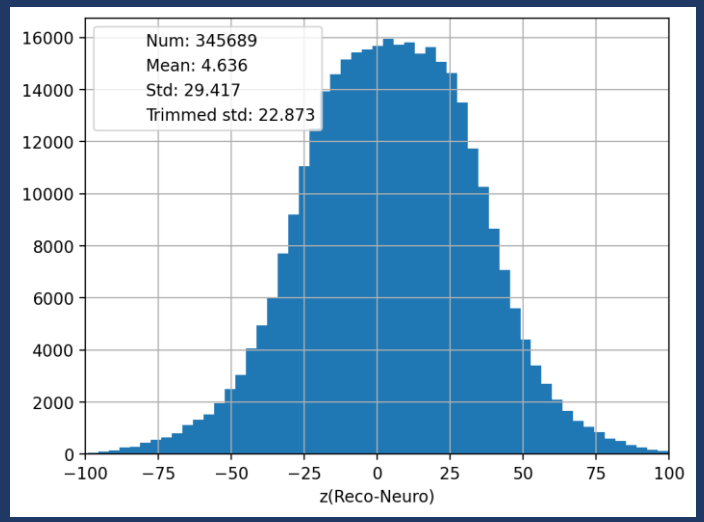
Pytorch training result -Uniform / norm distribution

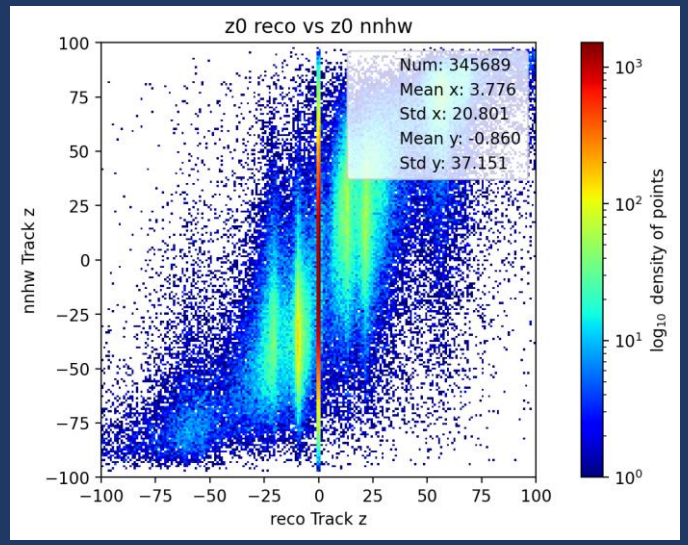
Norm



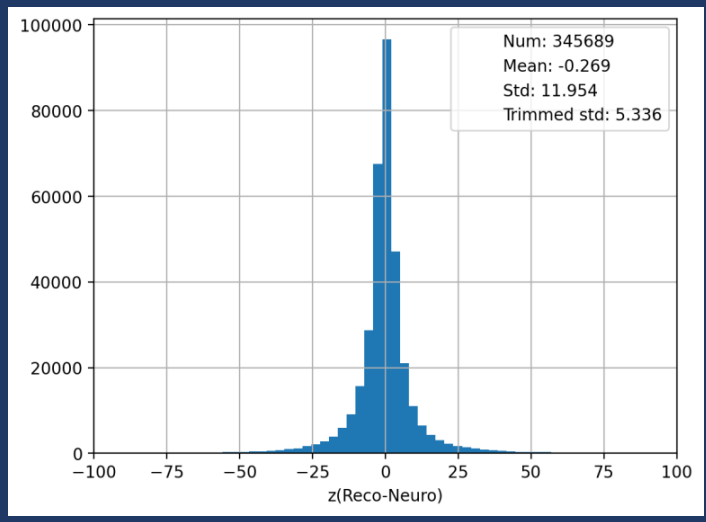


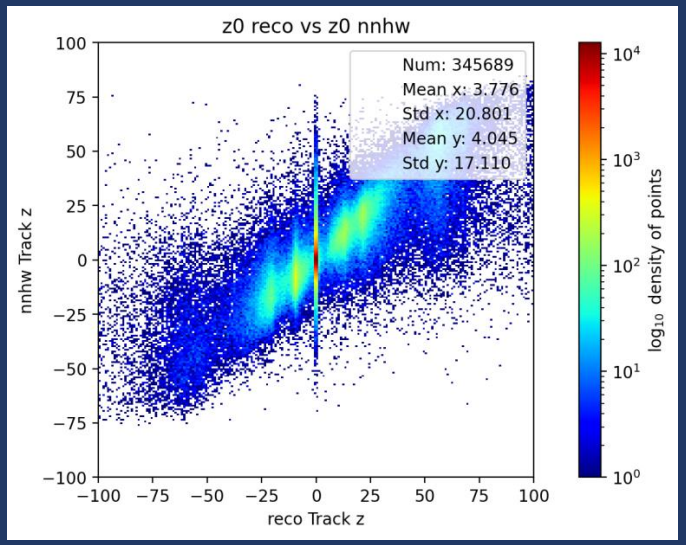
Uniform



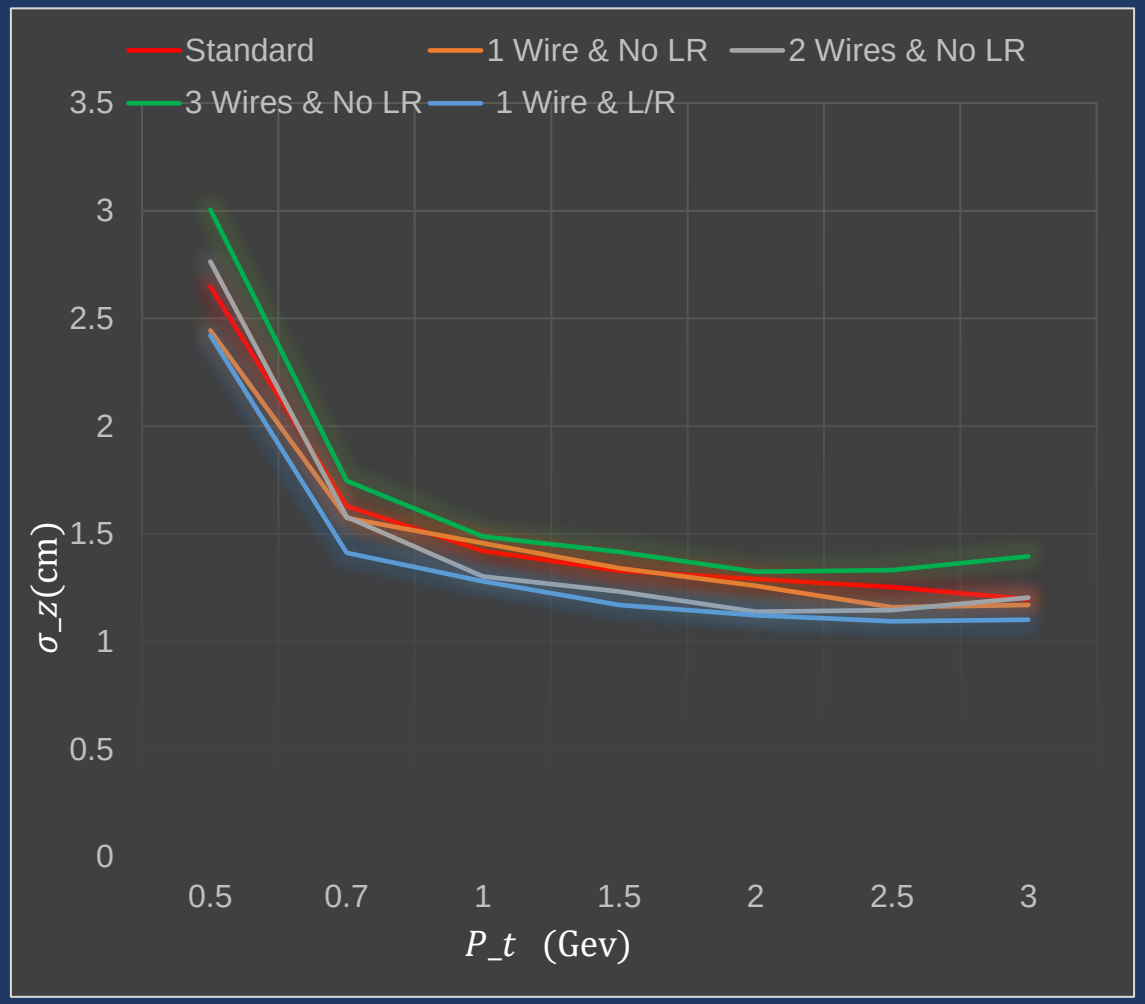


ExtraWires





Extra wire as input

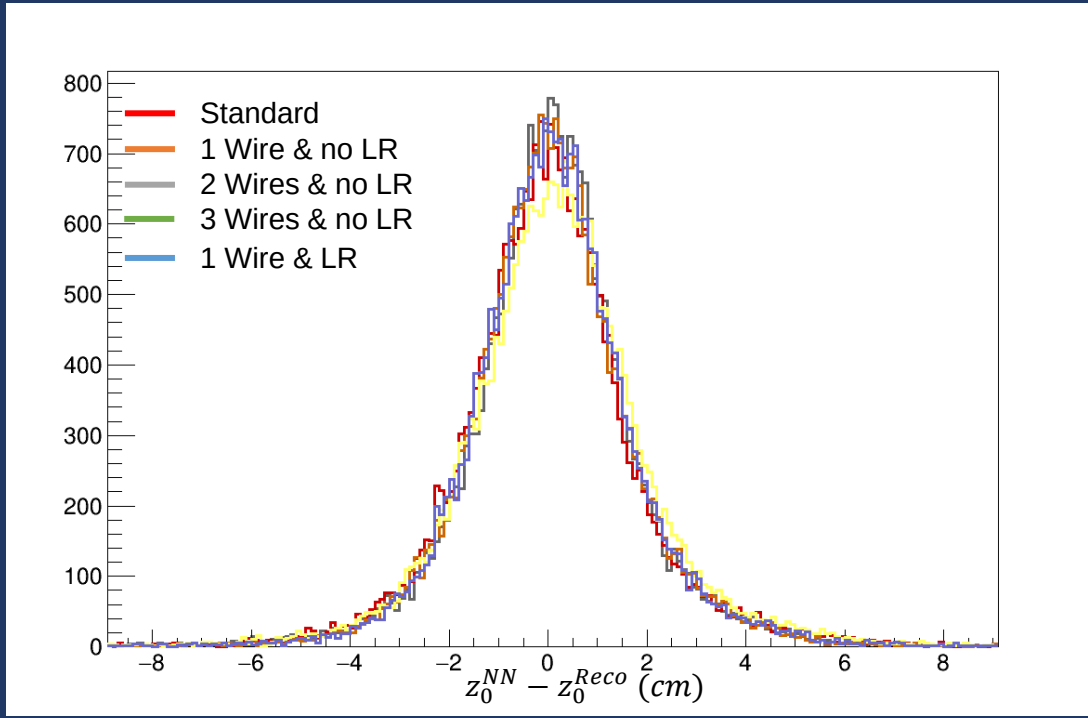


Trained NN based on MC

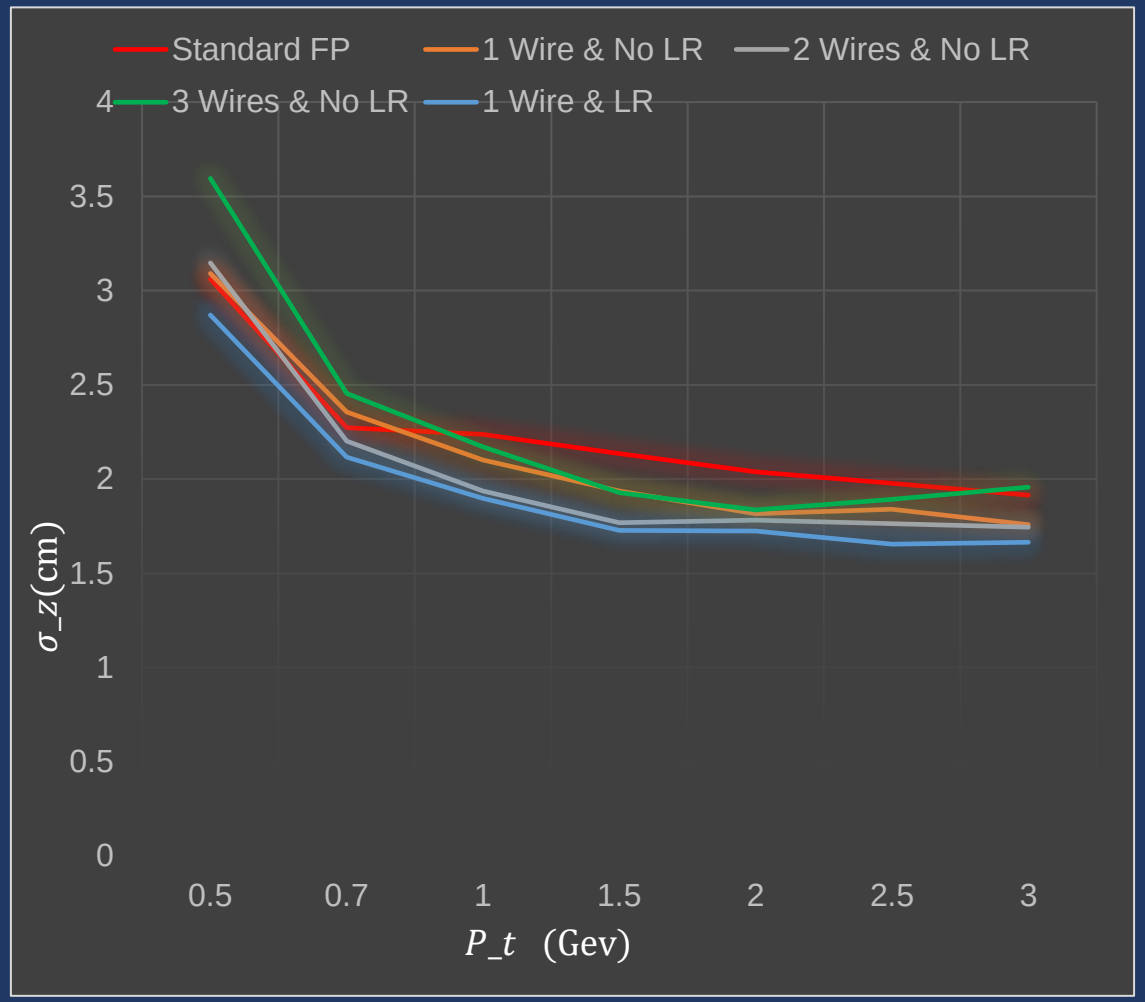
ETF : Set Event T0 as zero for precise t_{drift}

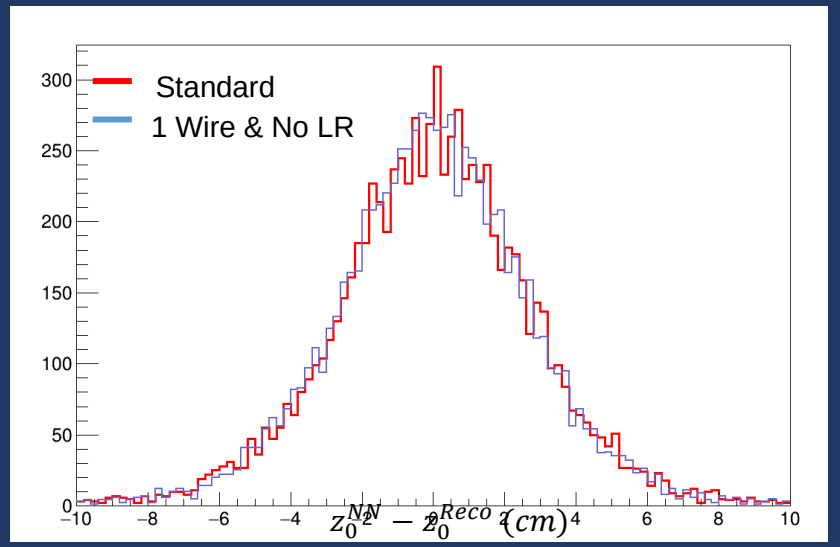
Add More wires without L/R make little improvement

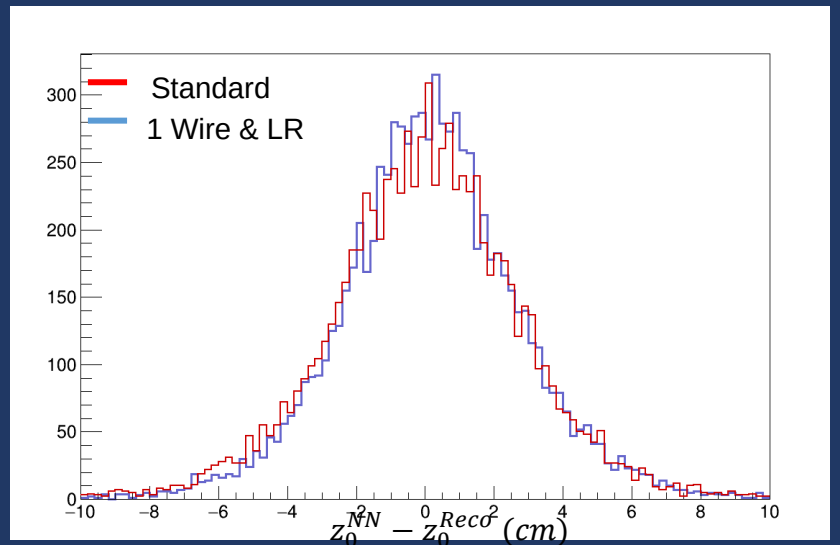
Add wire with L/R could make slightly difference in MC



Extra wire as input -- fastpriority

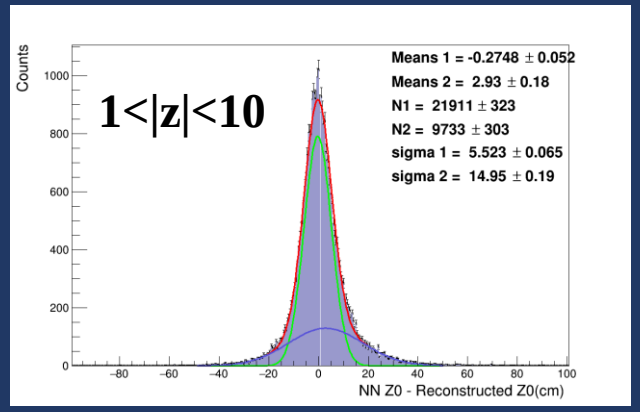


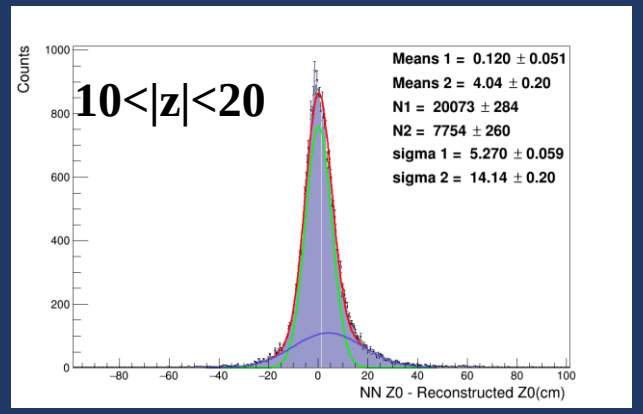


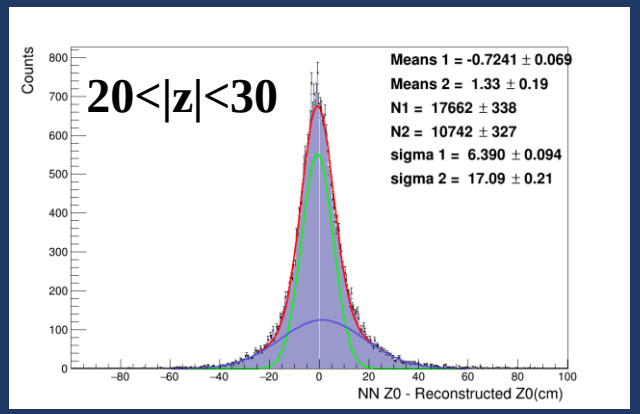


More wires?

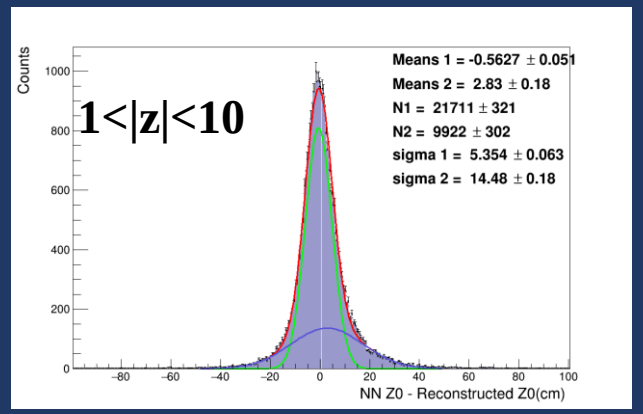
Wire 1

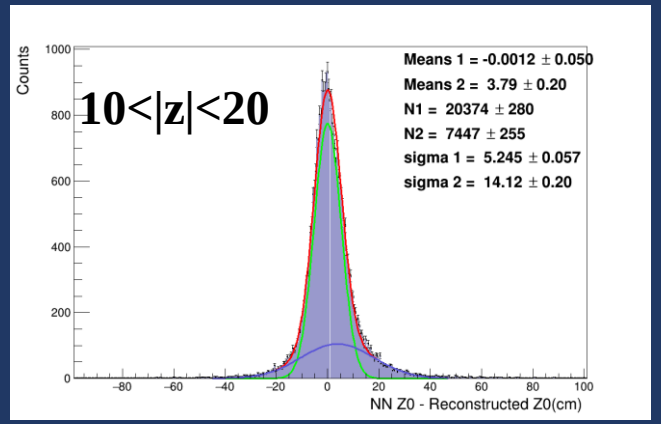


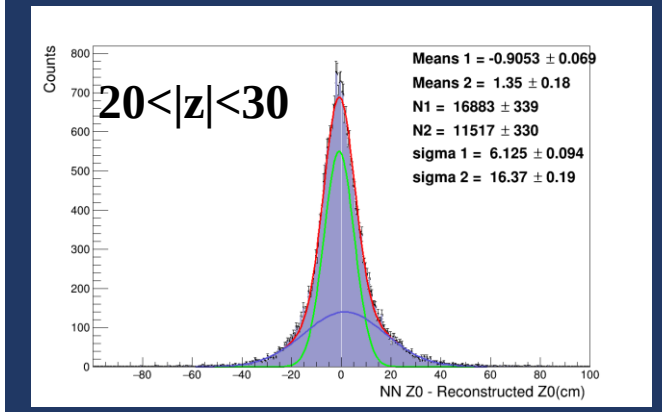




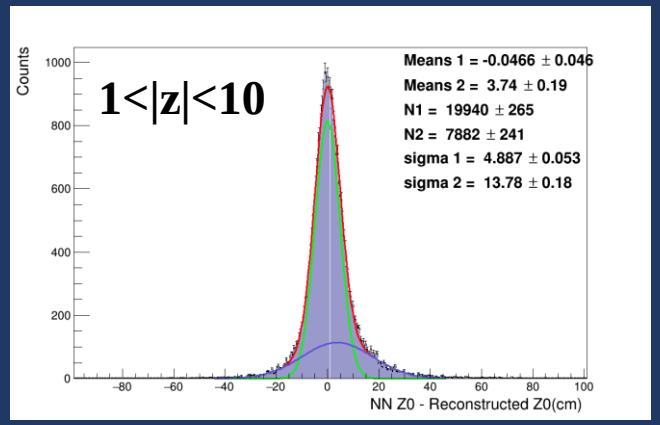
Wire 2

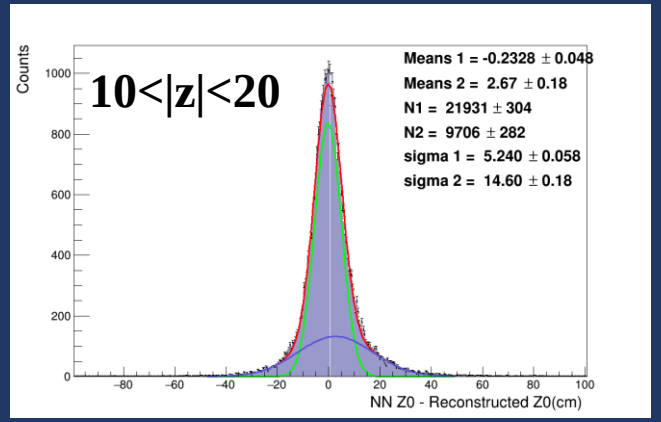


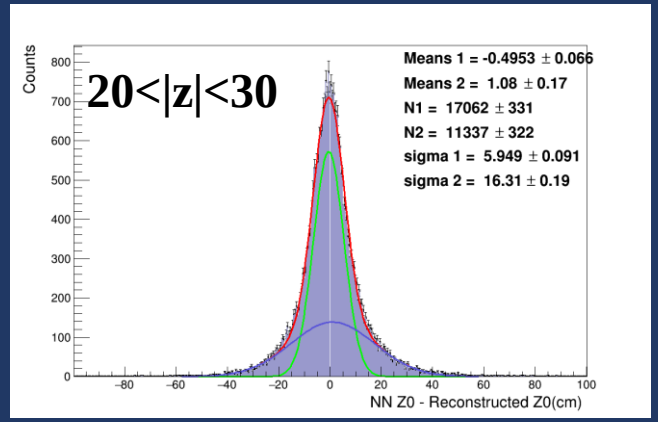




Wire 3

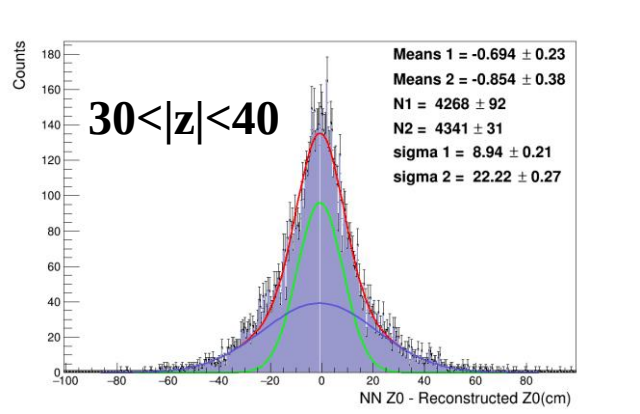


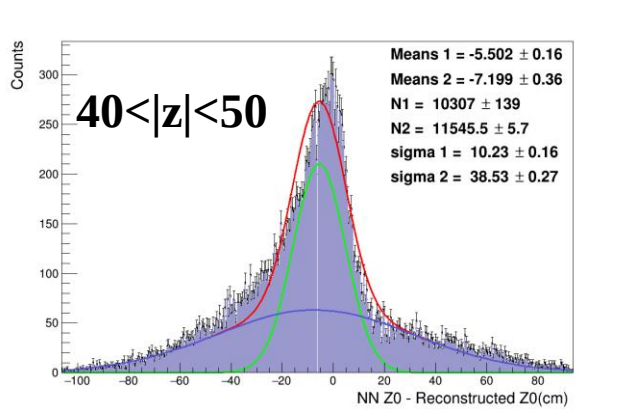




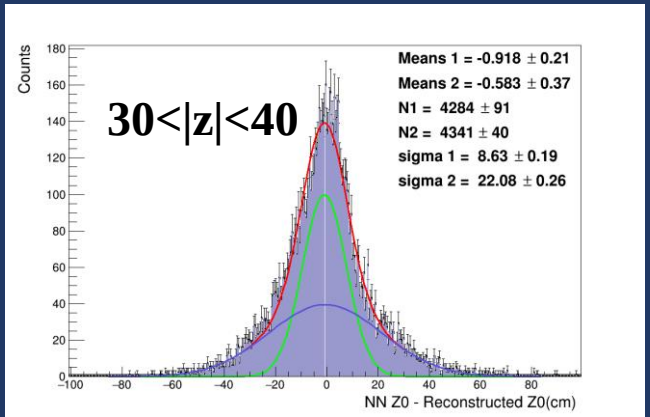
More wires?

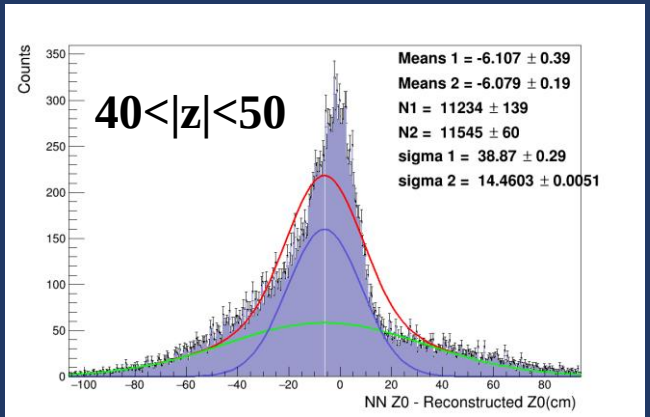
Wire 1



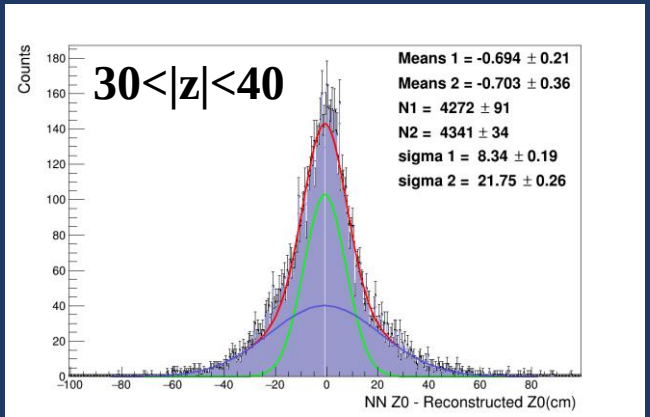


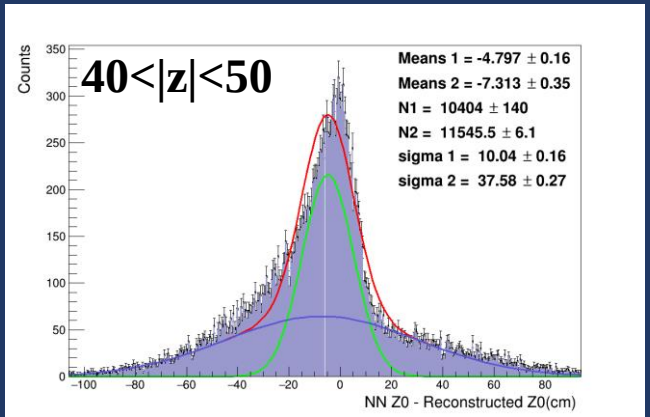
Wire 2





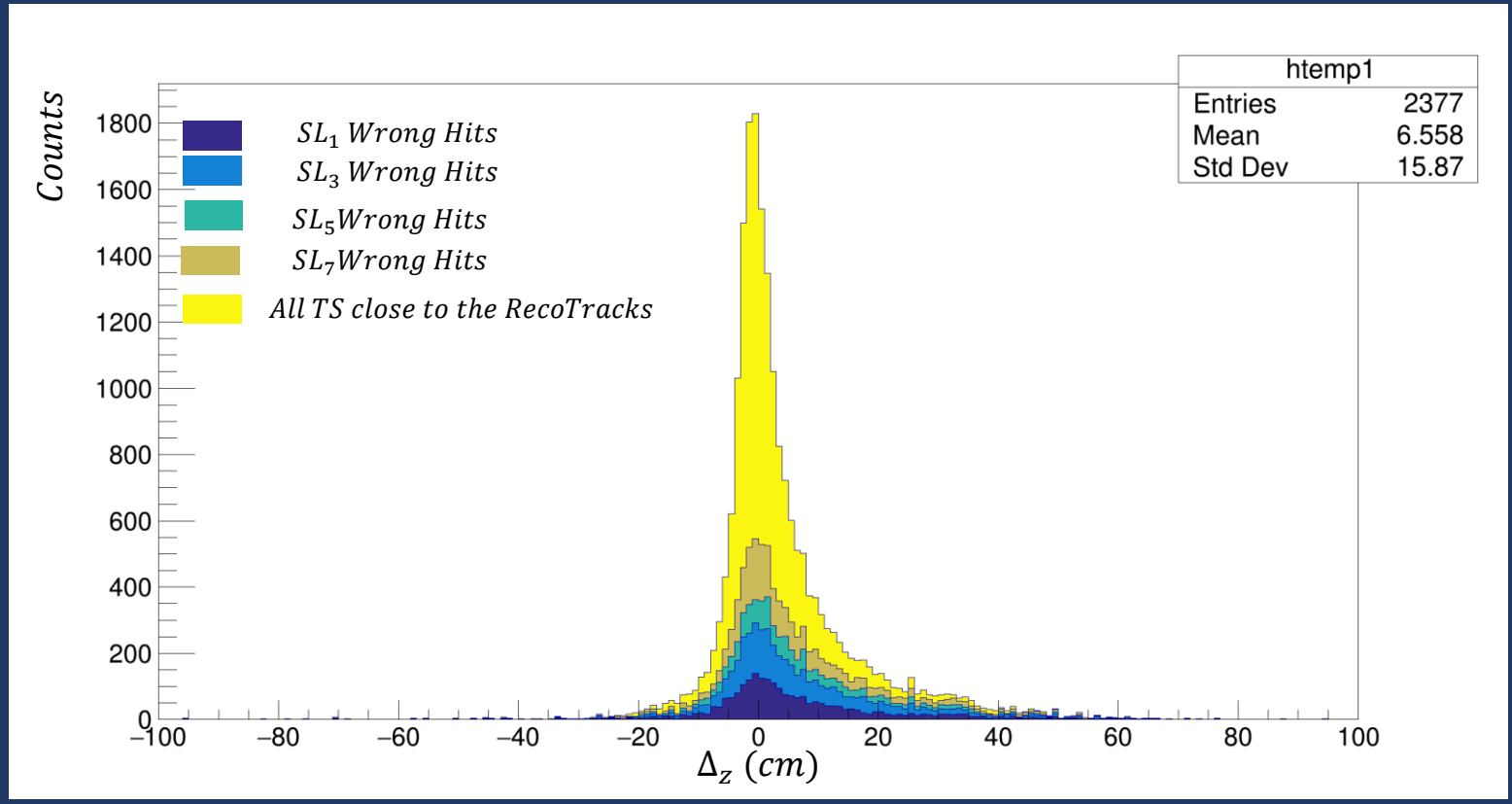
Wire 3





Track Segment ID

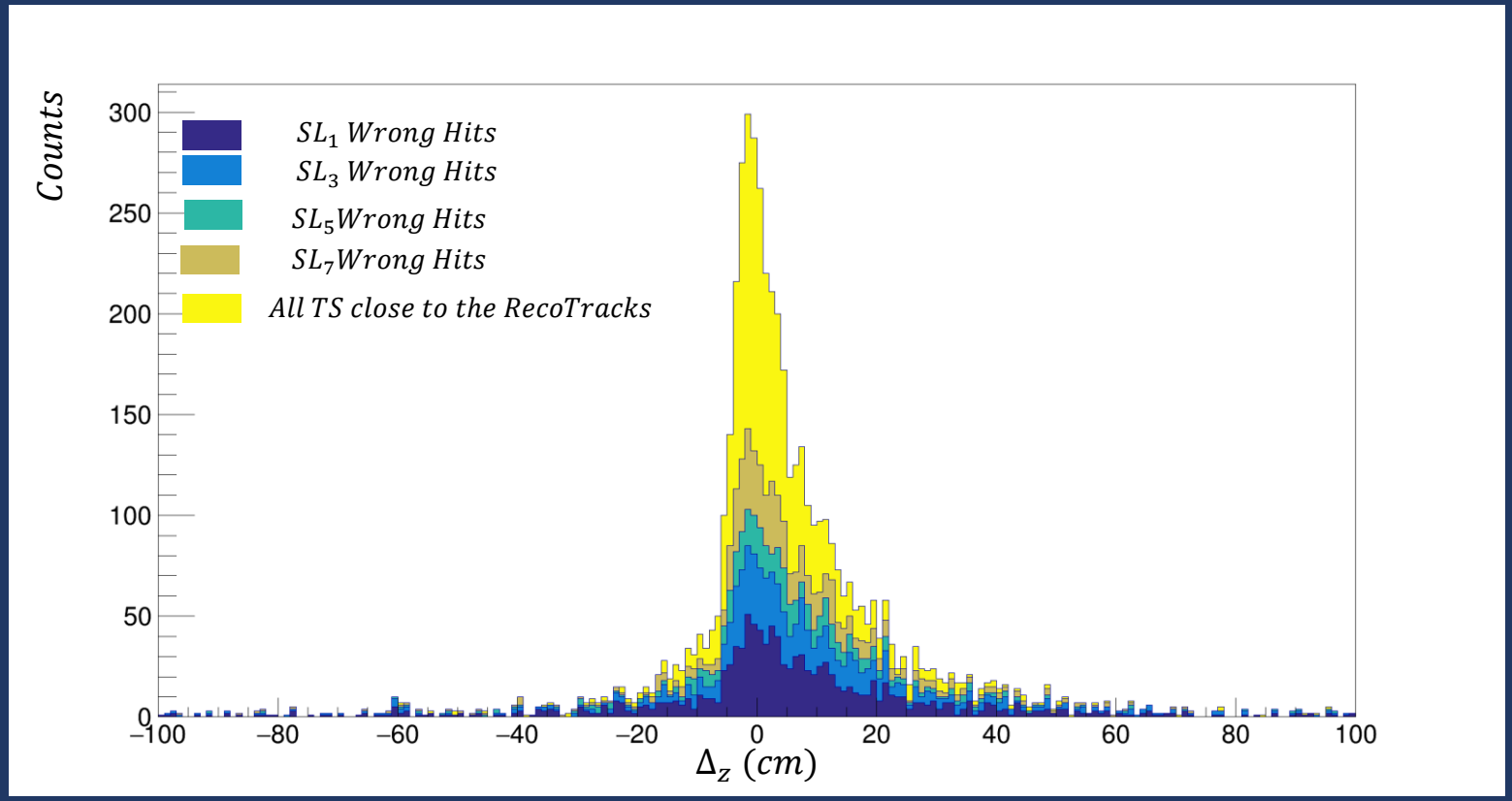
Consider the Delta Z distribution of those NN choose wrong hit



exp024run2004

Track Segment ID

Consider the Delta Z distribution of those NN choose wrong hit



exp026run1777