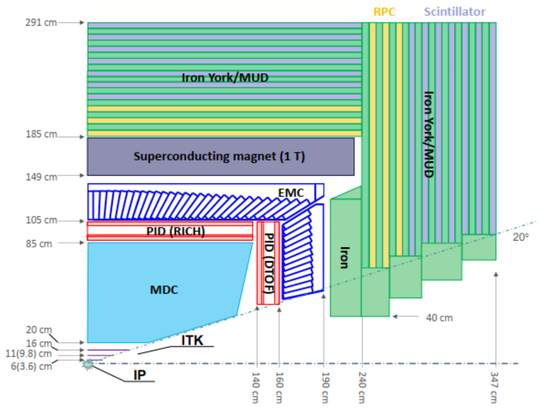
A 3D Reconstruction Algorithm for the Pre-research of STCF MDC L1 Trigger

Yidi Hao[1,2], Changqing Feng[1,2], Wenhao Dong[1,2], Zixuan Zhou[1,2], Zhujun Fang[1,2], Hang Zhou[1,2], Shubin Liu[1,2]

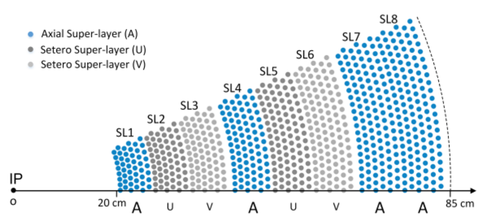
Introduction

STCF



- CME $2 \sim 7GeV$
- L $\sim 0.5 \times 10^{35}$ $cm^{-2}s^{-1}$
- Event rate ~400kHz

MDC



Cross-section view

- 8 superlayers
- 11520 sense wires

Preprocessing

Track Segment Finding

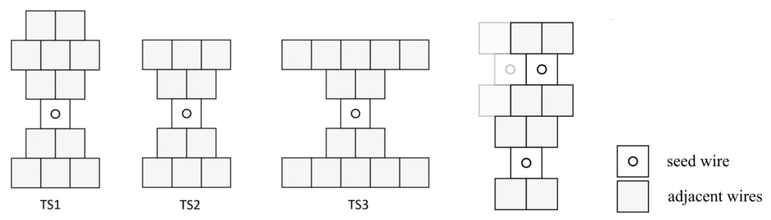
TS finding—the initial step commonly used in track finding algorithms of L1 trigger systems.

- More TSs (larger N) -> better resolution of z-vertex

$$\Delta z_0|_{res} = \frac{2\sigma_z}{\sqrt{(N+1)(N+2)}}\sqrt{(N+\frac{1}{2}) + \frac{3Nr_0}{L_0} + \frac{3Nr_0^2}{L_0^2}}$$

So we use

- 1 TS for each axial superlayer
- 2 TSs for each stereo superlayer



(a) Axial superlayers.

(b) Stereo superlayers.

Track Finding

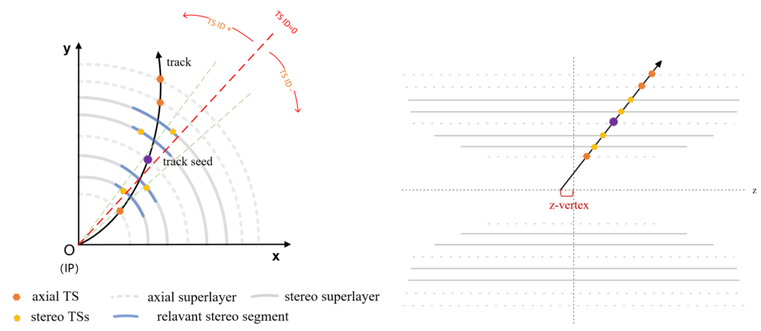
Track Info from 2D tracking algorithm:

- Axial TSs and p_t
- Azimuthal angles of track in stereo superlayers

Additional Info for 3D reconstruction algorithm:

- Stereo TSs near azimuthal angles calculated by 2D algorithm
- TDC timings of seed wires

Sectorization and Normalization



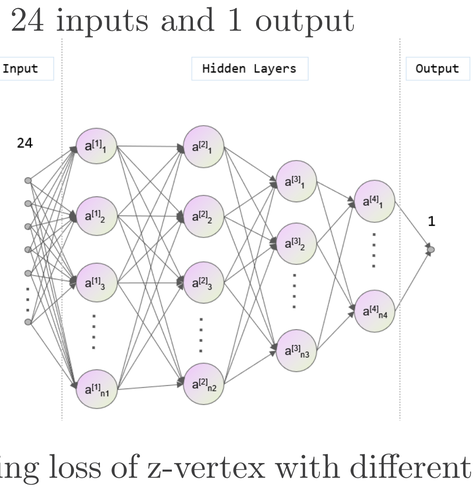
(a) XOY plane.

(b) XOZ plane.

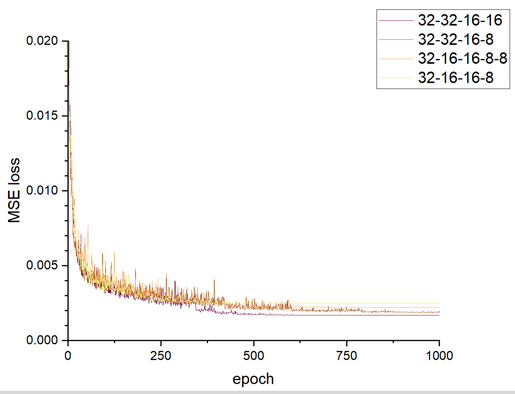
MLP Training

Basic Structure

MLP: 24 inputs and 1 output



Training loss of z-vertex with different MLP structures



Activation Function

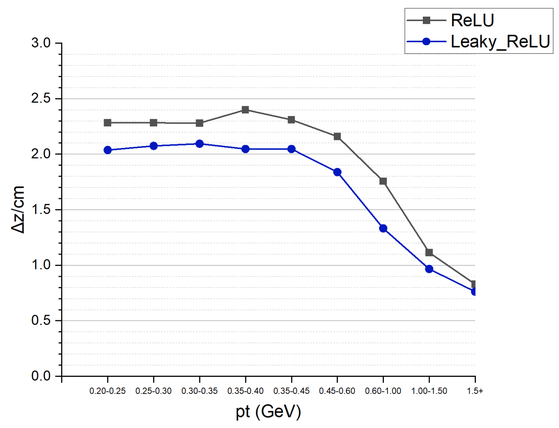
ReLU & LeakyReLU

- Highly suitable for FPGA implementation when choosing parameter "a" appropriately

$$f(y) = \begin{cases} y & y \geq 0 \\ ay & y < 0 \end{cases}$$

Z-vertex resolution

- MLP with identical structure but different parameters for each p_t interval
- Training: 60k tracks Test: 100k tracks



Qkeras

Quantization aware training:

- Train neural networks with ***fixed-point*** numbers
- Implement in FPGA without further precision loss

bitwidth*	8_1	12_4	16_6	20_8
Δz/cm	2.93	2.84	2.53	2.51

*W_I represents '$ap\_fixed\langle W, I\rangle$', indicating a W-bit fixed-point number with I integer bits (including one sign bit).

Pruning

Set less salient parameters to zero:

- Significantly reduces the model's size with minor accuracy loss
- ***Large-sparse*** Models outperform small-dense ones
- Resolution of z-vertex ranges from 1.1cm to 2.5cm in different p_t intervals

Structure	**Sparsity**	**NNZ Params**	**Δz/cm**
A[1]	0	2.53k	2.43
	0.2	2.04k	2.46
	0.4	1.52k	2.56
	0.6	1.06k	2.90
	0.8	0.58k	5.05
B[2]	0.4	2.11k	2.31

[1]24-32-32-16-8-1.
[2]24-48-32-16-8-1.

Implementation

Hls4ml

Hls4ml utilizes High-Level Synthesis (HLS) to convert neural networks trained by Python into the models described in HDL.

- A short development period
- Automatically optimize the logic design based on pruned structure and clock frequency

Structure	**Resource Utilization**				
	Sparsity	***BRAM***	***DSP***	***FF***	***LUT***
A[1]	0	0	610(22%)	80769(12%)	87014(26%)
	0.4	0	369(13%)	58293(8%)	52650(15%)
B[2]	0.4	0	503(18%)	77026(11%)	72034(21%)

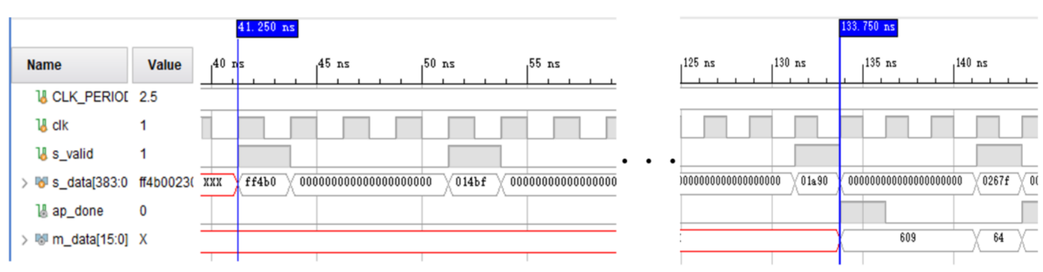
[1]24-32-32-16-8-1.
[2]24-48-32-16-8-1.

FPGA Resource Utilization

FPGA: XCKU060
clk: 400MHz

Simulation

Dead time: 4 clks
Latency: 37 clks (unpruned structure A); 30 clks (pruned structure B)



Conclusion

- The resolution of z-vertex is better than 3cm in all p_t intervals and the algorithm is supposed to reject beam background with a $\pm 3\sigma$ interval (about 10cm).
- With Qkeras and Pruning, the fixed-point MLP achieved comparable resolution with fewer parameters, thus reducing the consumption of FPGA resources.
- The MLP IP generated by hls4ml and HLS perfectly meets our need with a latency less than 100ns.
- The algorithm still needs further optimization to implement all required networks into a single FPGA.

[1]*State Key Laboratory of Particle Detection and Electronics, USTC, Hefei 230026, China*
[2]*Department of Modern Physics, USTC, Hefei 230026, China*
E-mail: ydhao@mail.ustc.edu.cn