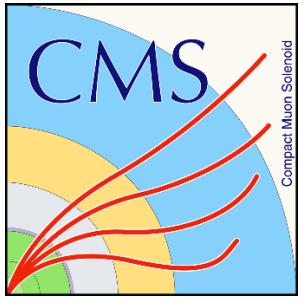
Deep Learning for the Level-1 ME0 Trigger in the CMS Experiment

HEO WooHyeon

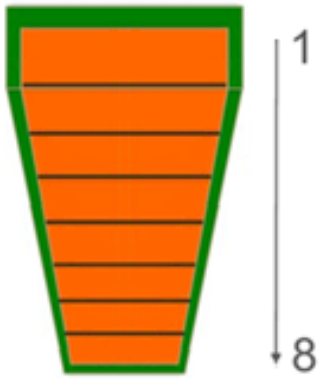
1st Korea HEP-FPGA Firmware Developers' Forum 2025

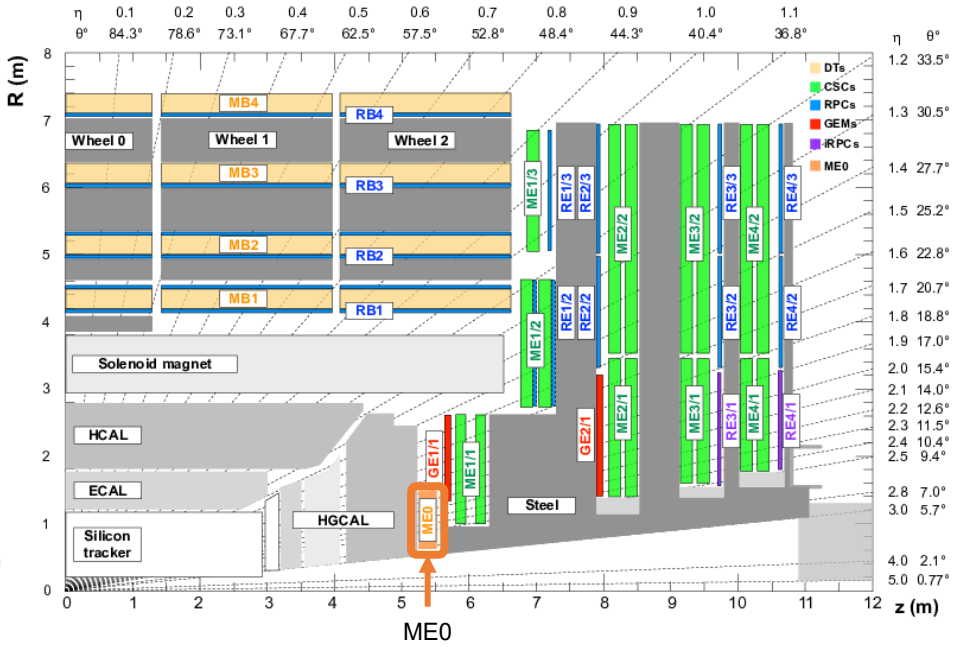
25th August 2025

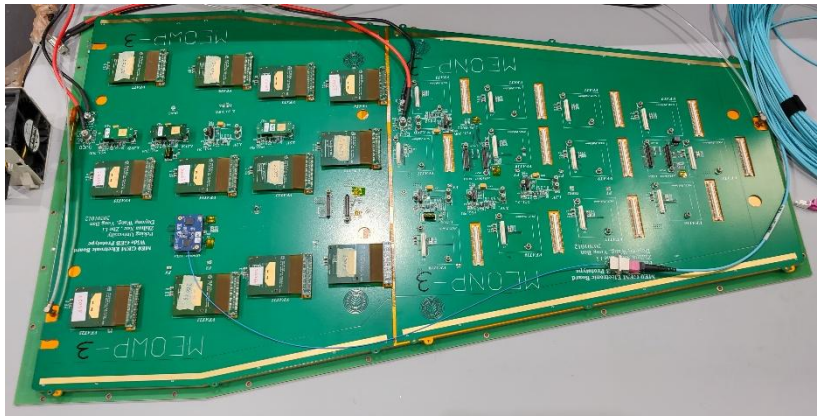


ME0

- In phase-2 upgrade of CMS, ME0 will be installed at the endcap as a part of the Muon system
- Feature[1]:
  - 6-layers of triple Gas Electron Multiplier (GEM) Chamber
    → Stack
  - 18 Stacks will be installed for each disk
  - Inner radius ≈ 0.6 m / Outer radius ≈ 1.5 m
  - Covers **2.0 < |η| < 2.8**, Δφ = 20°
    → The only muon detector above |η| = 2.4
  - Consists of 8 partitions along the η direction (iη) and 384 strips (374 for iη = 1) along the φ direction
- Due to the high background environment of ME0, it is important to trigger on proper targets



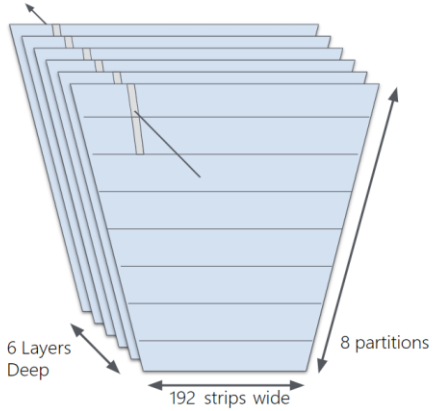


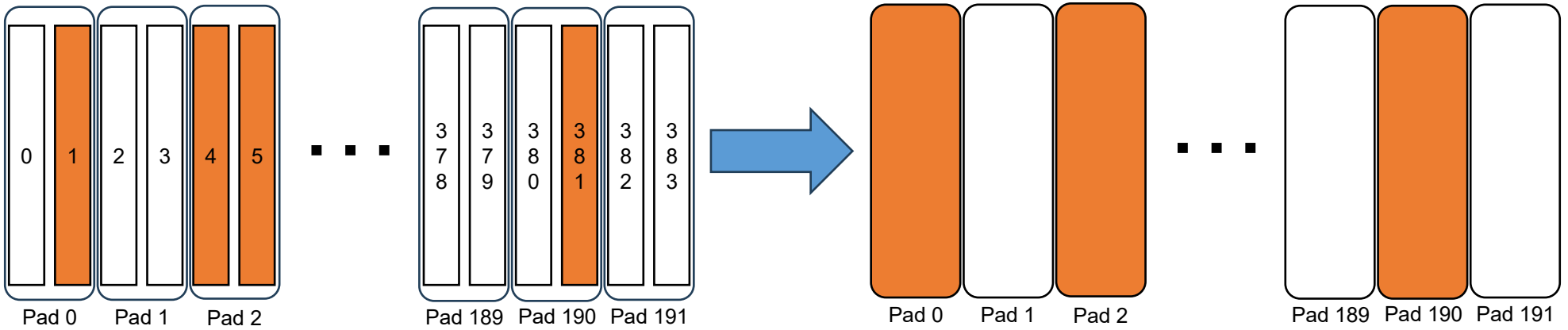


[1] “The Phase-2 Upgrade of the CMS Muon Detectors”, CMS Report (2017)

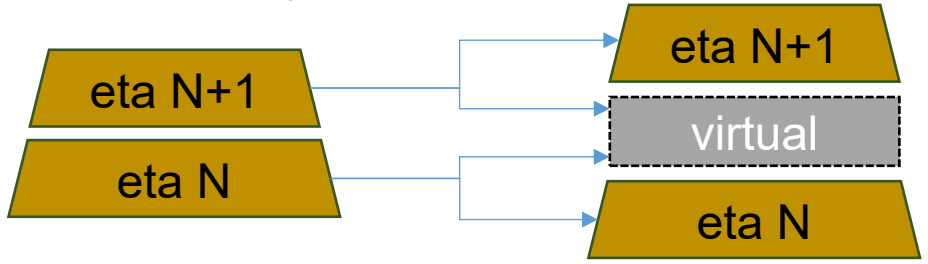
ME0 Stub Finder

1. Pre-Processing Data
   1) Pad Strip : PadStrip(N) <= Strip(2N) Or Strip(2N+1)
      → Reducing the processing time





   2) Combined eta Partition :
      Original 8 eta Partitions + 7 virtual Partitions (combined data of two adjacent eta partition)
      → Able to detect a track crossing two eta partition

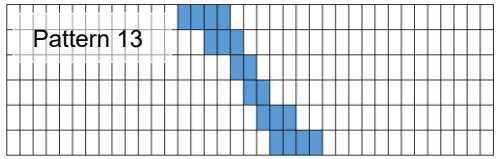


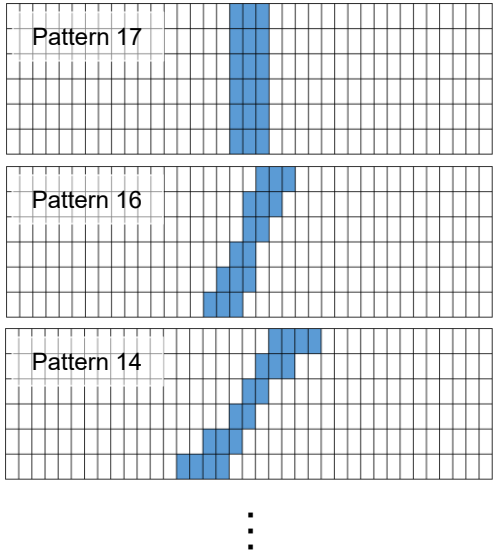
ME0 Stub Finder

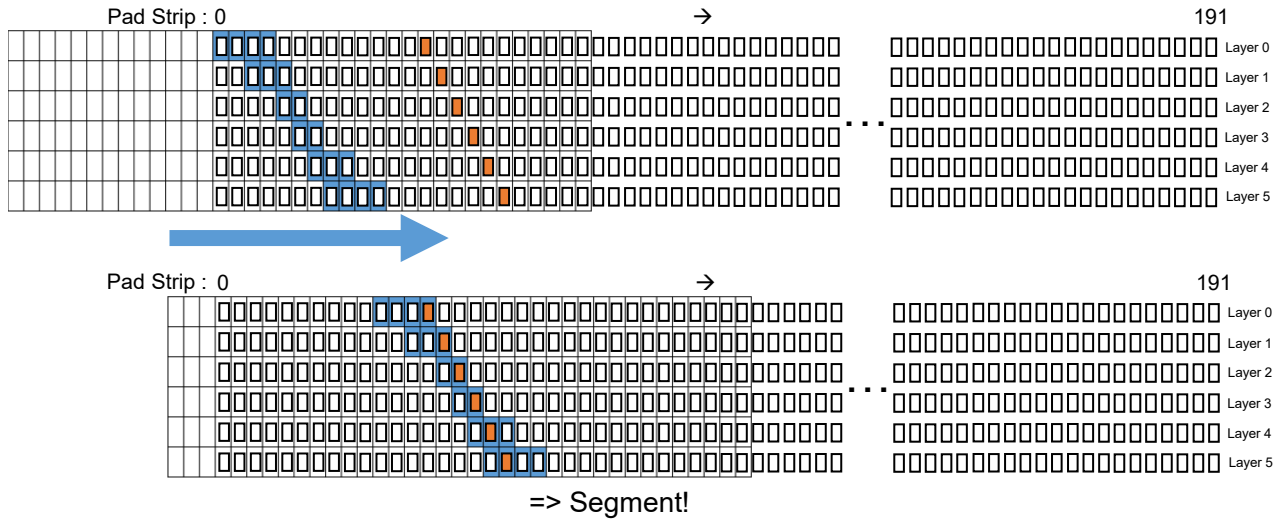
2. Scanning Pad Data with pattern masks

Pattern 1, 3, 5, … , 15
are mirrored patterns of
Pattern 2, 4, 6, … , 16

ex)







- Segments that satisfy certain conditions are sent to the Endcap Muon Track Finder (EMTF)
  - A segment must have hits in at least 4 layers (Minimal requirement)
  - 27 Bits per segment (4 : Eta / 10 : Phi / 9: Bending Angle / 4 : Quality)
  - Position and Bending Angle are obtained by linear regression
- Simulation result with minimal requirement:
  - Efficiency = 99.19 %
  - Minbias rate per chamber = 179.7 MHz

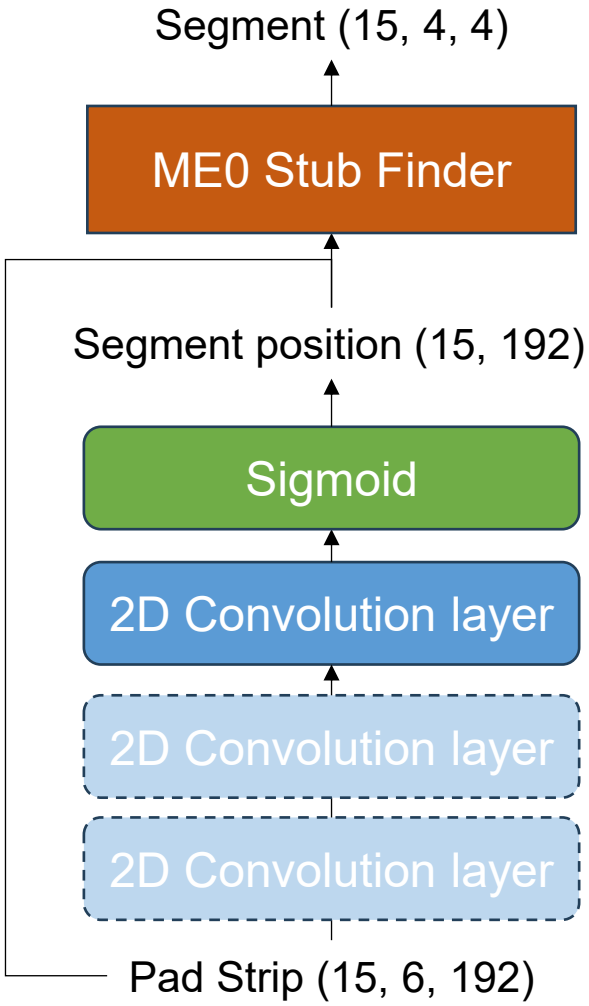
$$\text{Muon Efficiency} = \frac{(\#\text{ of matched muon track})}{(\#\text{ of total muon track})}$$

$$\text{Minbias rate per chamber} = \frac{(\#\text{ of unmatched segment})}{(36\text{ chambers}) \times (25\text{ ns})} \times (\text{Fill Factor})$$

Concerning number of segments to send EMTF / most of them are effect of Pile-up → need to filter the segments from Pile-Ups

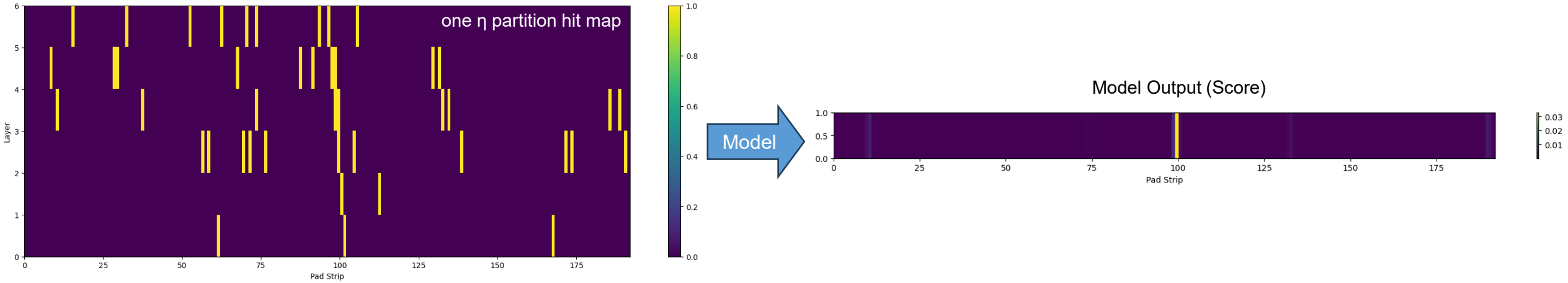
ME0 Stub Finder with ML

- Study for ME0 Stub Finder (ME0SF) using CNN
  - CNN show great performance in pattern finding problem
- Model :
  - 1-3 layers of 2D Convolution filter + ReLU between CNNs
  - Set to be light to meet the maximum processing time of ME0 Trigger system
- Input Data : (15, 6, 192)
  - Pad Strip data for 15 η partitions (8 original η partitions + 7 virtual partitions)
  - Training data : muons with p_T = 1-10 GeV and average 200 Pile-Up (~100,000 events)



ME0 Stub Finder with ML

- Output Data : (15, 192)
  - Segment strip position
    - 15 vectors corresponding to strip-wise segment position for 15 partition
    - Each vector has 192 dimensions and n^{th} dimension correspond to n^{th} pad strip
    - The score of 0 to 1 will be given representing how a segment is likely be at that pad strip
  - ME0SF will only run on the strips specified by the model, while the standard algorithm scans all strips



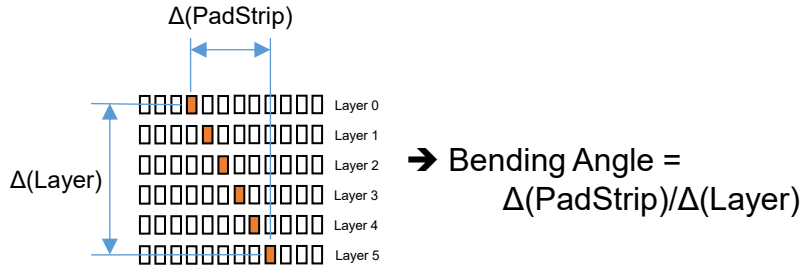
Performance

- Performance of ME0SF with 1-, 2-, and 3-layer Models
- Sample
  - For efficiency : 50,000 events, each containing 8 randomly generated muons with uniform p_T=1–200 GeV and |η|=2.0–2.8, along with an average of 200 additional pile-up collisions per bunch crossing (BX)
  - For Minbias rate : 50,000 events, only pile-up collisions, with an average of 200 per BX
- Matching rule :
  - A segment is considered matched with a muon track if

    Eta position match : $\left|(\eta\ Partition)_{MuonTrack} - (\eta\ Partition)_{segment}\right| \leq 1$

    Strip position match : $\left|(PadStrip)_{MuonTrack} - (PadStrip)_{segment}\right| \leq 5$

    Bending angle match : $\left|(Bending\ Angle)_{MuonTrack} - (Bending\ Angle)_{segment}\right| \leq 0.4$

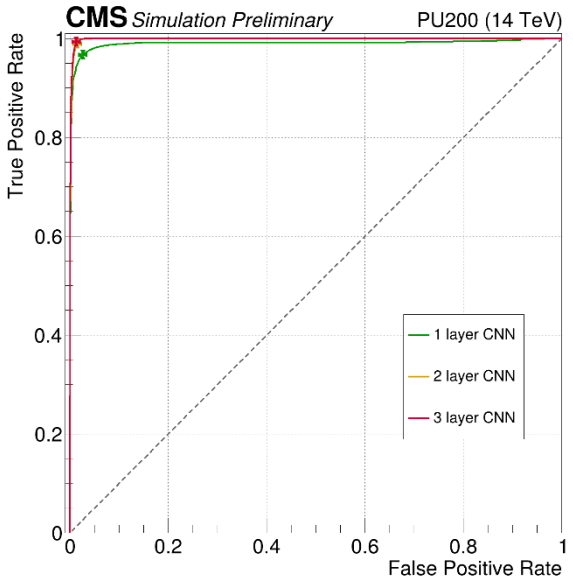


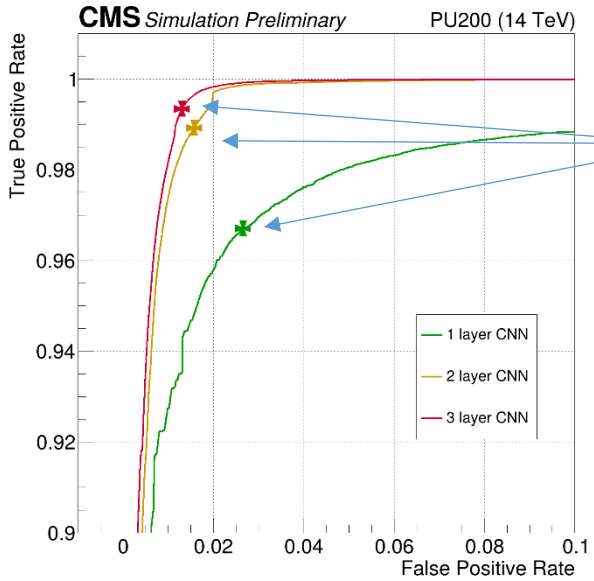
➔ Bending Angle = Δ(PadStrip)/Δ(Layer)

- Muon Efficiency $= \dfrac{\text{(\# of matched muon track)}}{\text{(\# of total muon track)}}$
- Minbias rate per chamber $= \dfrac{\text{(\# of unmatched segment)}}{\text{(36 chambers)} \times \text{(25 ns)}} \times$ (Fill Factor)

  Fill Factor ≈ 0.7710

Overall performance





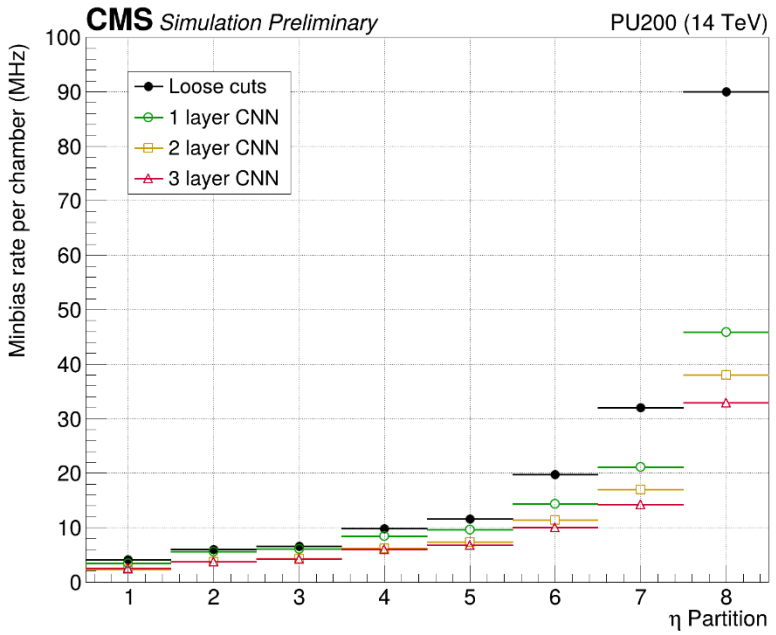
Best Discriminator Points

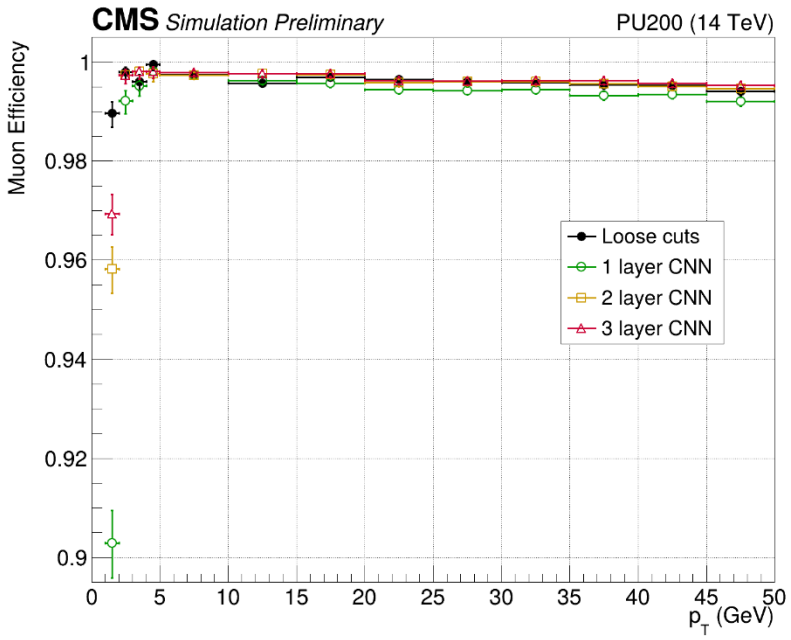
	Loose cut*	1 layer CNN	2 layers CNN	3 layers CNN
Efficiency	99.19 %	98.70 %	99.04 %	99.21 %
Minbias rate per chamber	179.7 MHz	114.3 MHz	90.10 MHz	80.32 MHz

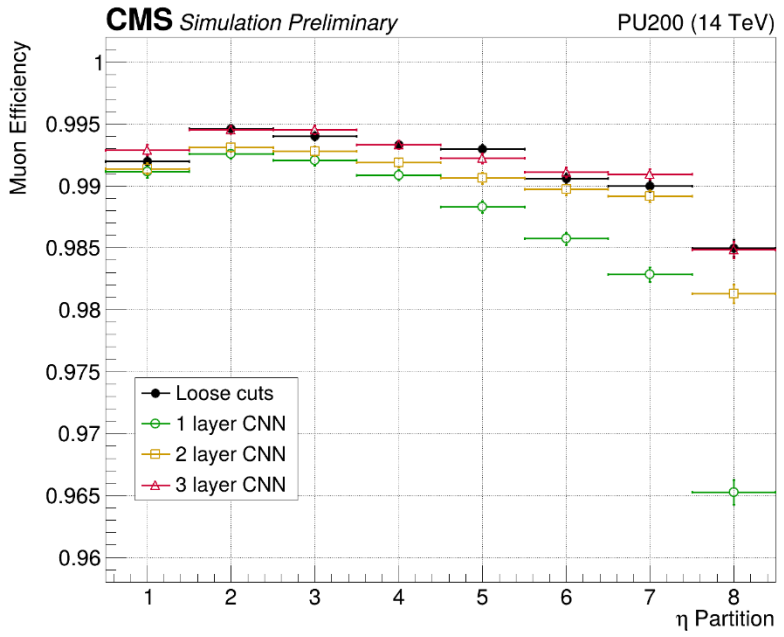
For every case of CNN, Minbias rate is reduced by 1/3 to 1/2 of the one from the standard ME0SF while preserving ~99% of Efficiency, even for the 1-layer CNN

* “Loose cut” indicate the standard ME0SF implementation with a minimal segment requirement, defined as having hits in at least 4 layers. The “loose cut” is also applied to CNN assisted ME0SF

Performance vs η partition & p_T







- Significant reduction in the Minbias rate when using CNN especially in high η
- Maintained >95% efficiency after applying CNN to ME0SF even for the low p_T of <5 GeV or high η of 2.8

$$\text{Muon Efficiency} = \frac{(\#\text{ of matched muon track})}{(\#\text{ of total muon track})}$$

$$\text{Minbias rate per chamber} = \frac{(\#\text{ of unmatched segment})}{(36\text{ chambers}) \times (25\text{ ns})} \times (\text{Fill Factor}$$

Quantization

$$y = W \cdot x + b$$

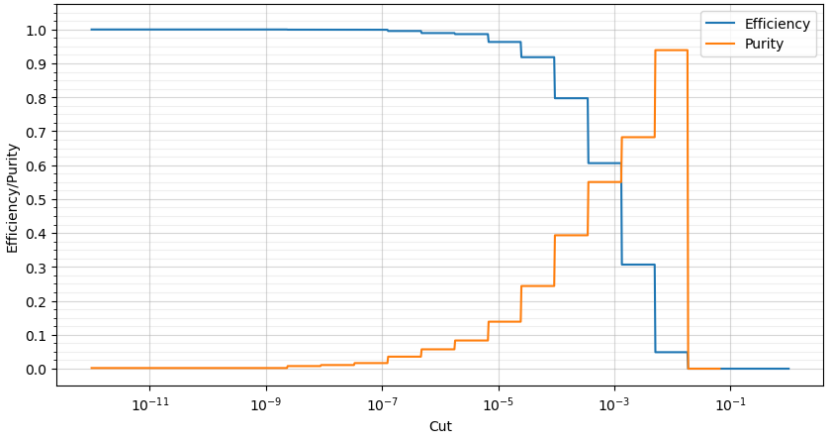
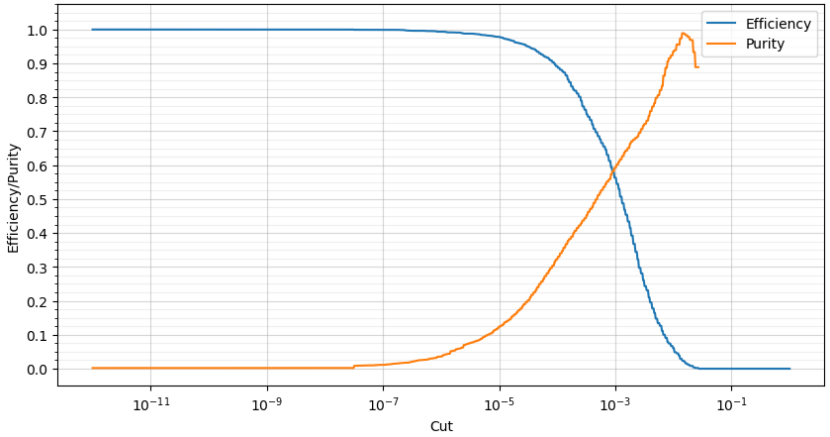
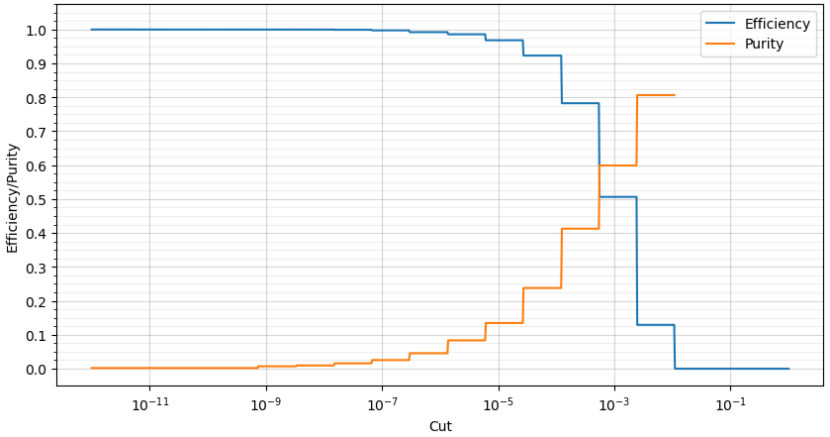
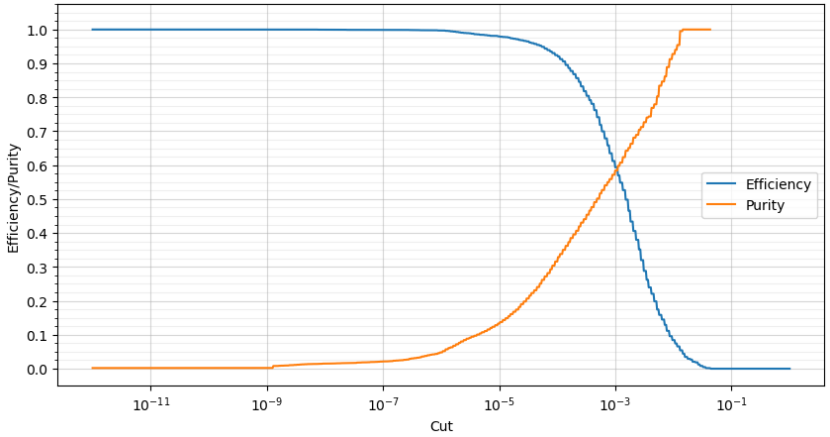
$$y = s_W(W_q - z_W) \cdot s_x(x_q - z_x) + b$$

$$y_{int} = (W_q - z_W) \cdot (x_q - z_x) + \left\lfloor \frac{b}{s_W s_x} \right\rfloor$$

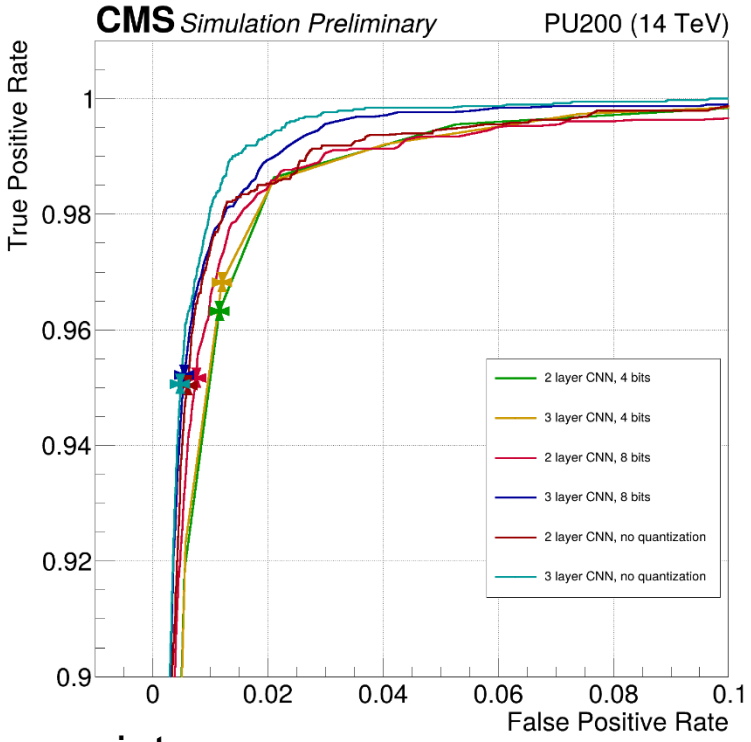
- Technique to simplifies the computation of deep learning model reducing memory usage and increasing processing speed
  - Reducing the number of bits in weight
  - Pruning multiple layer of network
- To fit in the required processing time, quantizing the model is essential

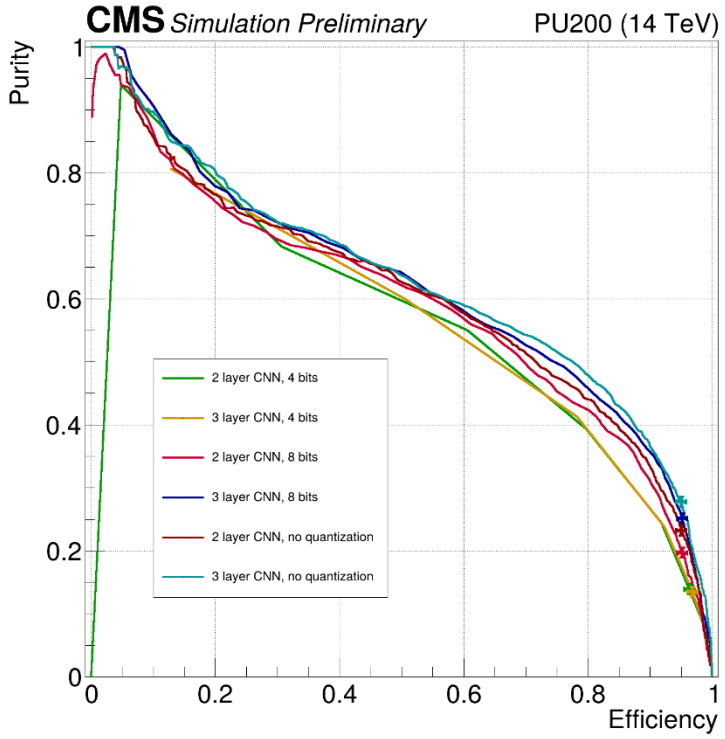
Quantization-aware Training

- Quantization-aware training for 2 and 3 layers CNN model with 4 and 8bits quantization
- Kernel size = (3,3) for intermediate layer and (6,3) for final layer, $N_{kernel} = 5$

	4bits Quantization	8bits Quantization
2 Layers CNN		
3 Layers CNN		

Quantization-aware Training





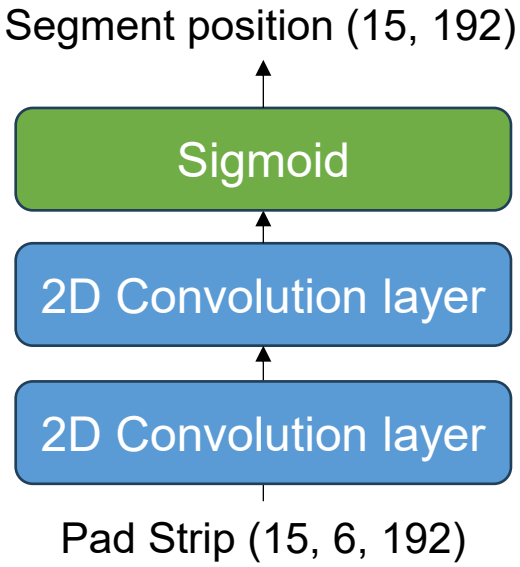
Using 95% efficiency point

	2 layer, 4bits	3 layer, 4bits	2 layer, 8bits	3 layer, 8bits	2 layer (no quantization)	3 layer (no quantization)
Efficiency	0.9632	0.9682	0.9515	0.9522	0.9504	0.9506
Purity	**0.1385**	**0.1346**	**0.1965**	**0.2508**	**0.2323**	**0.2777**

hls4ml

- “hls4ml” is a python package for implementation of ML model to FPGAs
- Supports ML frameworks of Keras, Pytorch and ONNX
- Supports Vivado HLS and Vitis HLS
- hls4ml translate the pre-trained model to c++ that is convertible to the hardware language like Verilog or VHDL

hls4ml




```
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  |  3.33 ns|  2.388 ns|     0.90 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+------+------+----------+
    |  Latency (cycles) |  Latency (absolute) |   Interval  | Pipeline |
    |   min   |   max   |    min   |    max   | min  |  max |   Type   |
    +---------+---------+----------+----------+------+------+----------+
    |     1361|     1363|  4.536 us|  4.543 us|  1159|  1160|  dataflow|
    +---------+---------+----------+----------+------+------+----------+
```

Performance report of the model

Conclusion

- CNN Models with different depth of 1, 2 and 3 are trained to filter the pile-up induced segments in ME0SF
  - The models allow ME0SF to run only on strips specified by the model

    → Potential to decrease the processing time for ME0SF
- CNN Models effectively reduced the Minbias rate while preserving efficiency even for high η or low p_T
- Quantize the model to reach much accessible processing time and resource

Next Plan

- Find optimal number of kernel or kernel size or stride comparing with performance and processing time

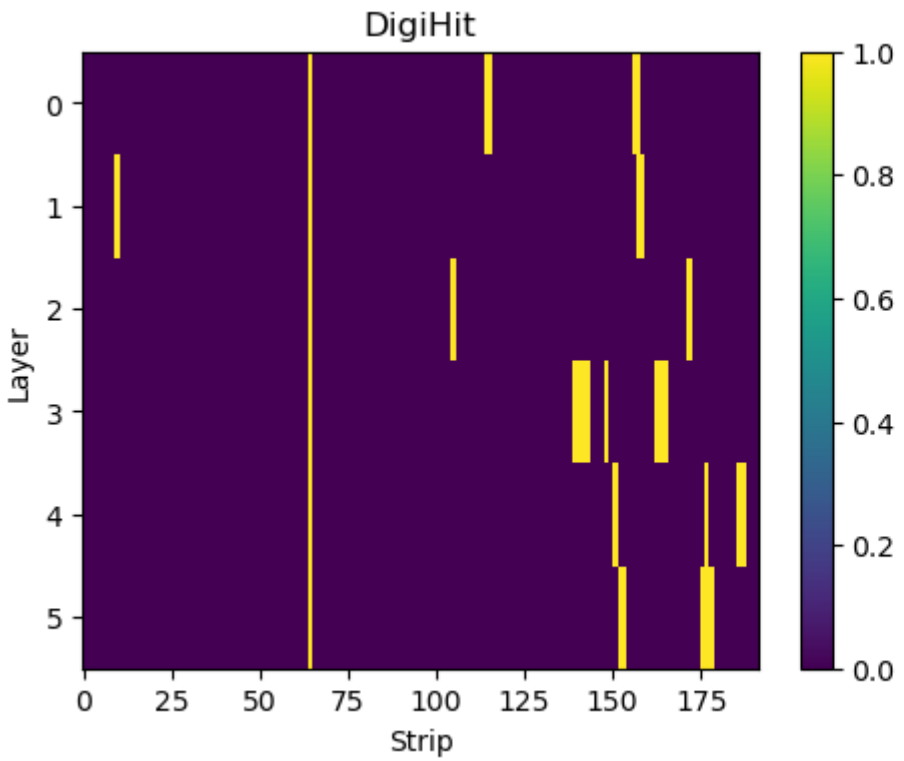
Back Up

Processing Latency

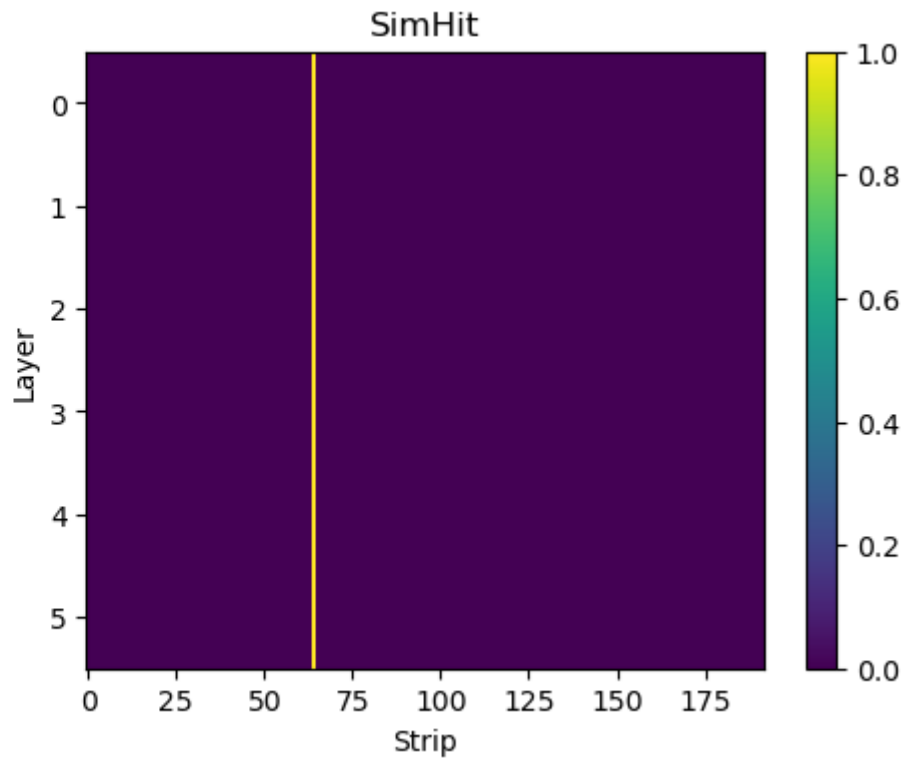
- Processing Latency result from Vitis-hls synthesis
  - Target Device = xcu250-figd2104-2L-e (Alveo250)
  - Target Clock Period = 3.333 ns
  - Minimum Requirement : Max Latency < 4 μs

Model	Estimated Clock	Max Latency (cycle)	Max Latency (absolute)
1 layer CNN	2.514 ns	34	0.113 μs
2 layer CNN (3 kernel)	2.411 ns	1357	4.523 μs
2 layer CNN (7 kernel)			

ML Out Example

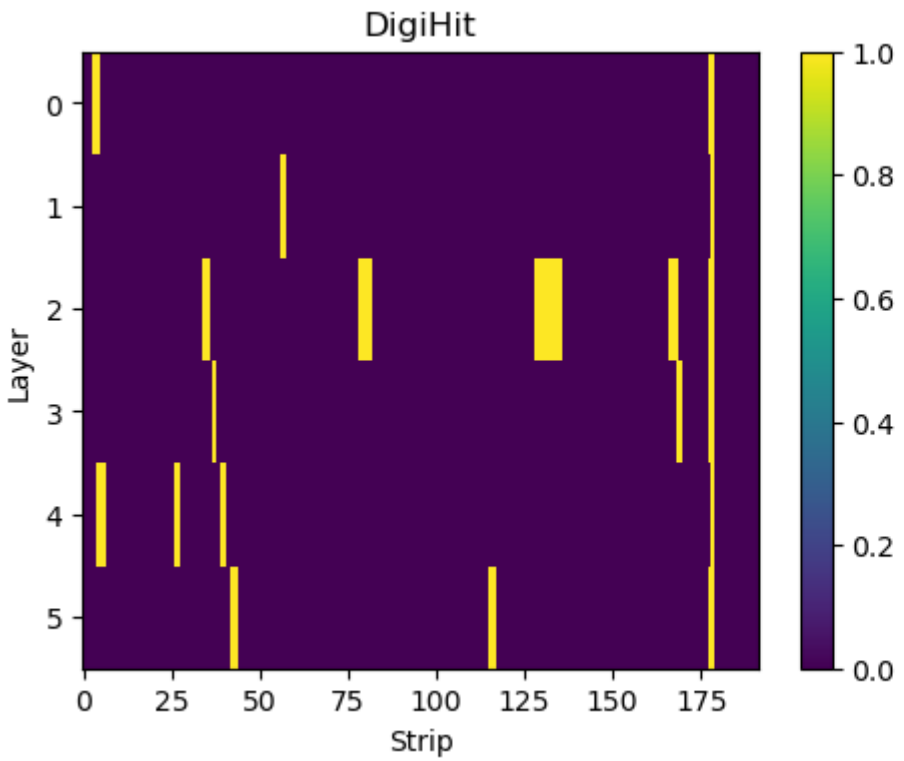


Seg Position from ML : 64, 65, 63

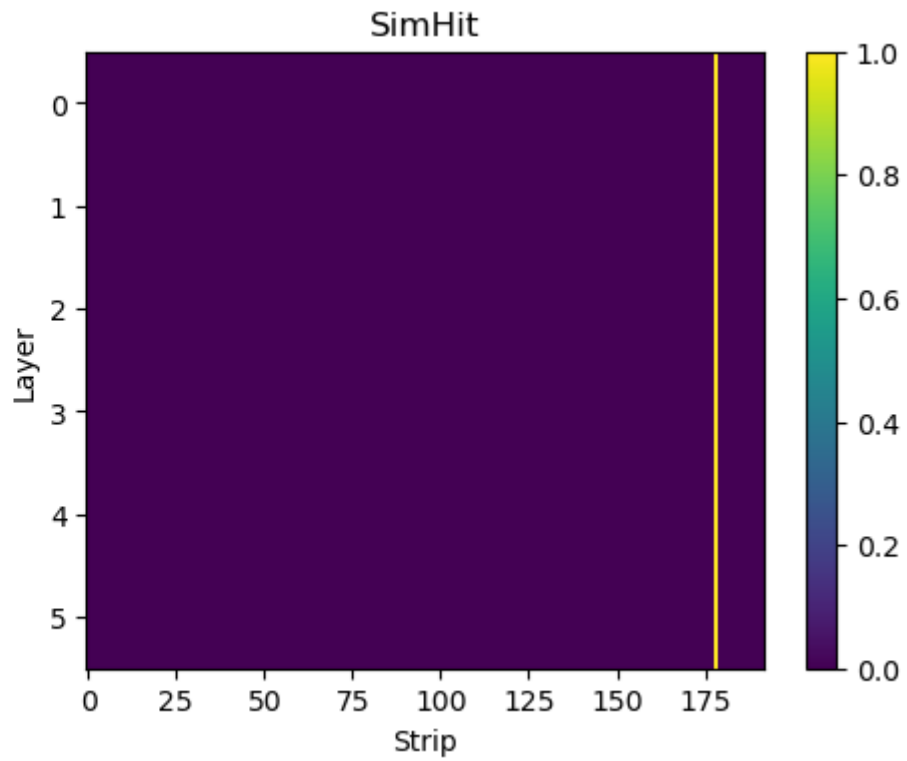


Real Track Position : 64

ML Out Example

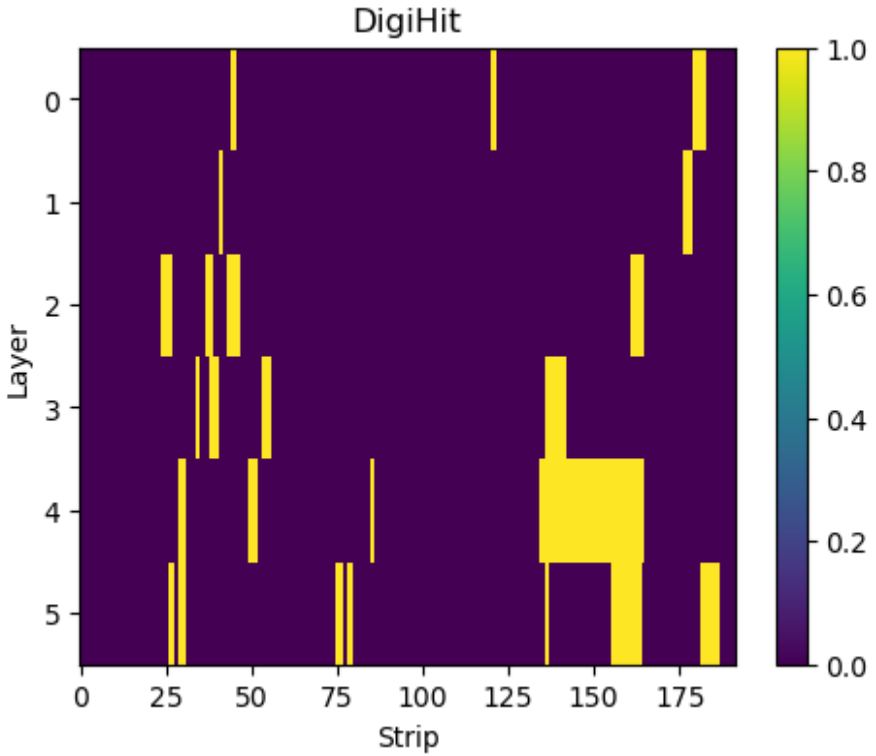


Seg Position from ML : 178, 177, 168

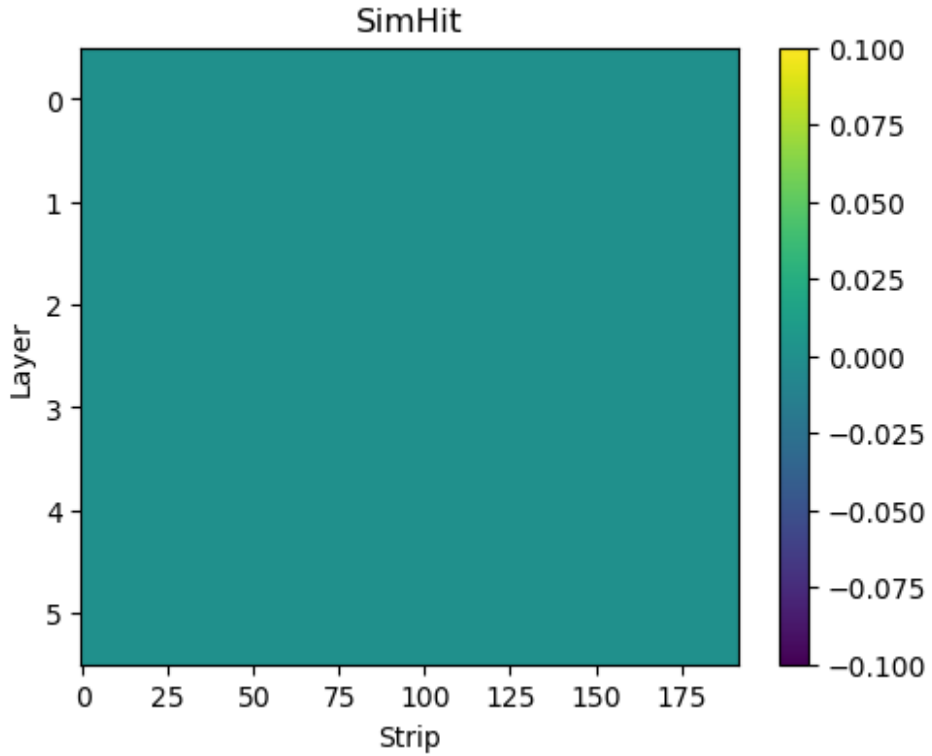


Real Track Position : 178

ML Out Example

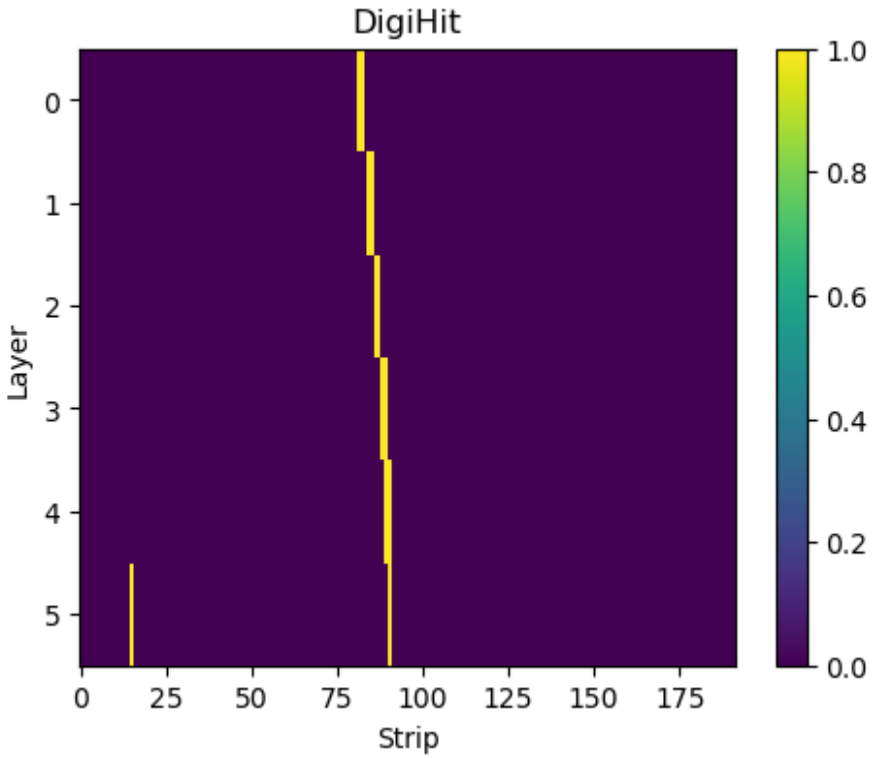


Seg Position from ML :

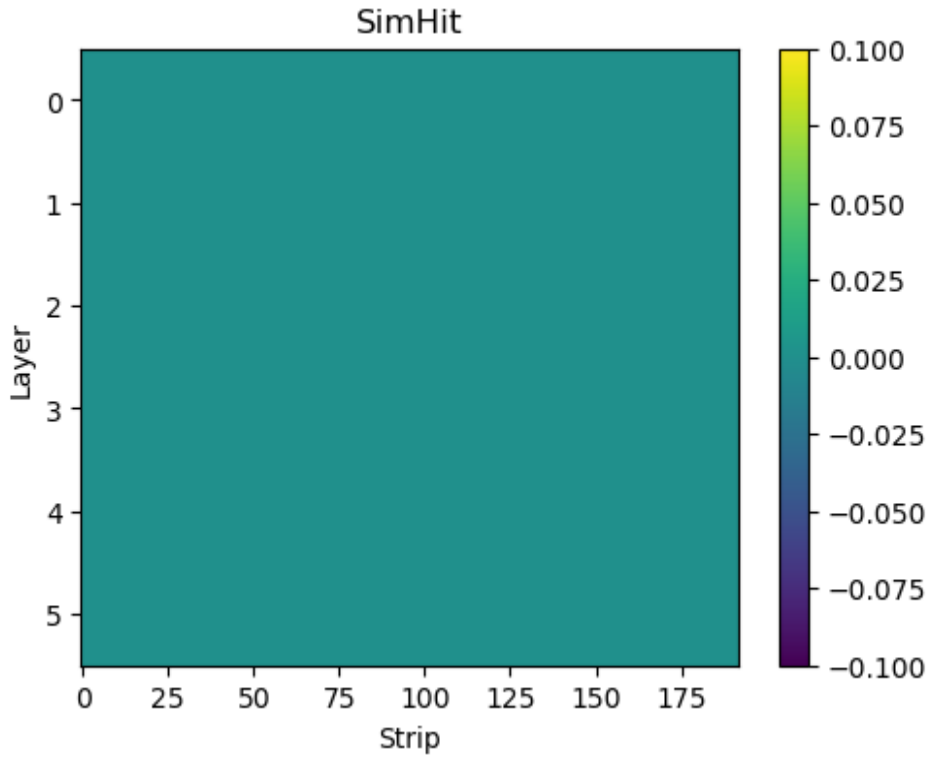


Real Track Position :

ML Out Example



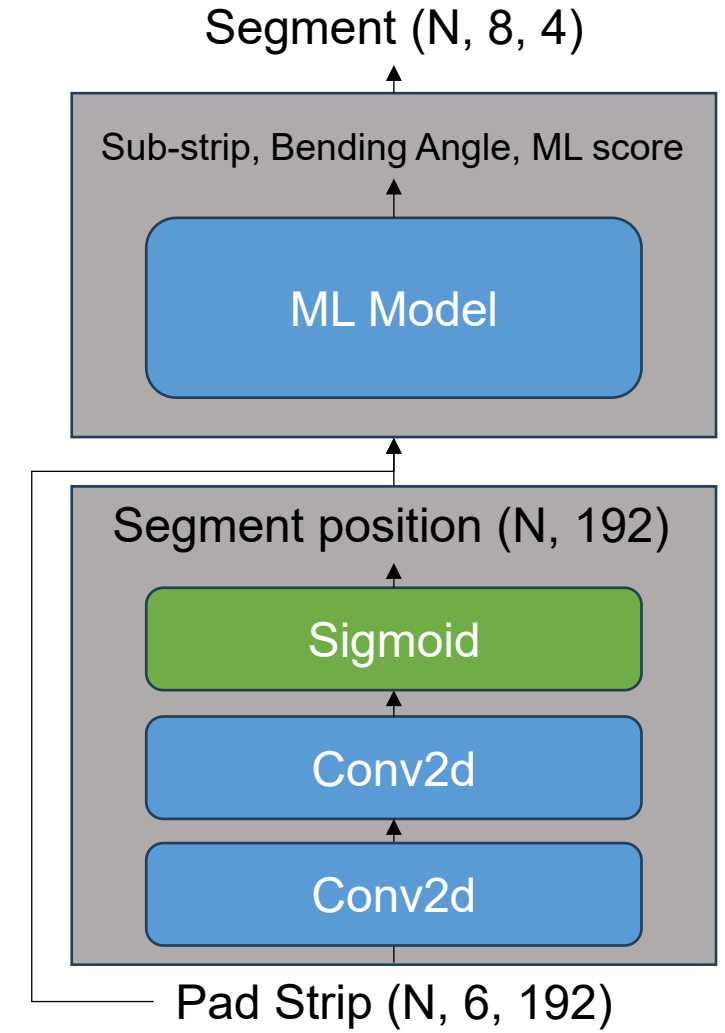
Seg Position from ML : 88, 87



Real Track Position :

Further Plan

- Full ML Algorithm for ME0 Stub Finder
- Model that produce “sub-strip”, “bending angle” and “ML score” at given position from “Position Finder”
  - ML score (= model loss) is used for the ghost cancelation or cross partition cancelation



Process Latency

Process Latency from hls4ml report

- Target device:
  xcu250-figd2104-2L-e (**Alveo250**)
- Total Latency : 8341 clock cycles
  - 27.801 μs for 3.33 ns clock
  - Mostly caused by Interval
    (the time to get next data)

What to do

- Change Configuration
  - PipelineStyle
    DataFlow → Pipeline (Failed)
  - Strategy
    Latency → Unrolled
  - Precision
    fixed<16,6> → lower the bits

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+---------+----------+------------+
    |  Clock |  Target | Estimated| Uncertainty|
    +--------+---------+----------+------------+
    |ap_clk  |  3.33 ns|  2.433 ns|     0.90 ns|
    +--------+---------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+-----------+-----------+------+------+----------+
    |  Latency (cycles) |   Latency (absolute)  |   Interval  | Pipeline |
    |   min   |   max   |    min    |    max    | min  |  max |   Type   |
    +---------+---------+-----------+-----------+------+------+----------+
    |     8341|     8341|  27.801 us|  27.801 us|  8322|  8322|  dataflow|
    +---------+---------+-----------+-----------+------+------+----------+
```