COMET Phase-I実験オンライントリガーシステムの開発

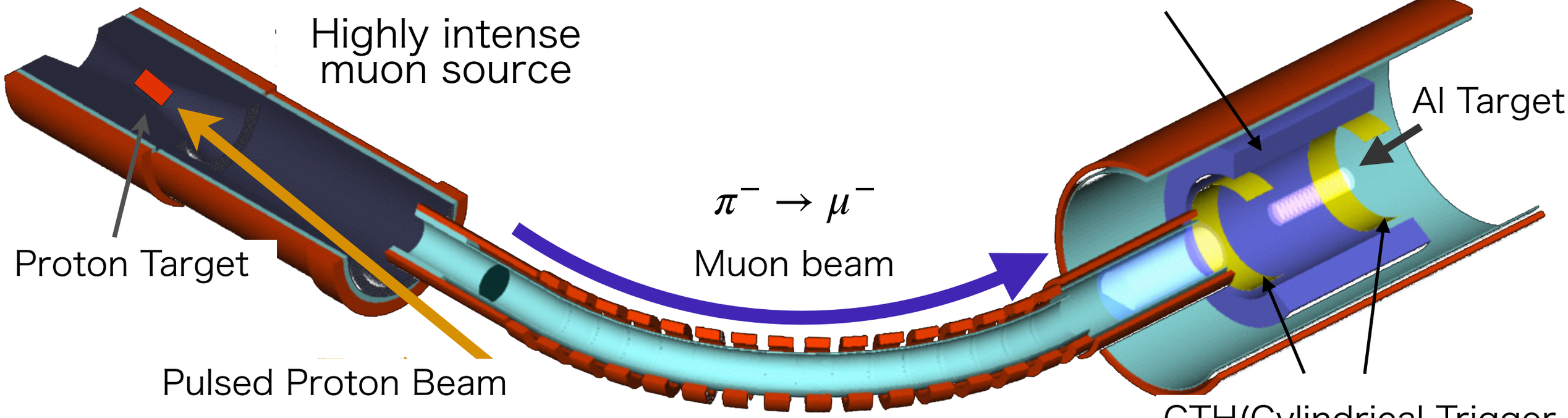
M1 宮滝　雅己

大阪大学 理学研究科 物理学専攻 青木研究室

m-miyataki@epp.phys.sci.osaka-u.ac.jp

COMET Phase-I experiment

COMET Phase-I Layout



- **Purpose**:**Search for μ-e conversion in an Al target**
  - **Signal : monoenergtic 105MeV electron**
- **Single event sensitivity**: 3.0×10^{-15}(100 times the current sensitivity)
- **Detector**:**Cylindrical detector system**
  - electron momentum and timing measurement

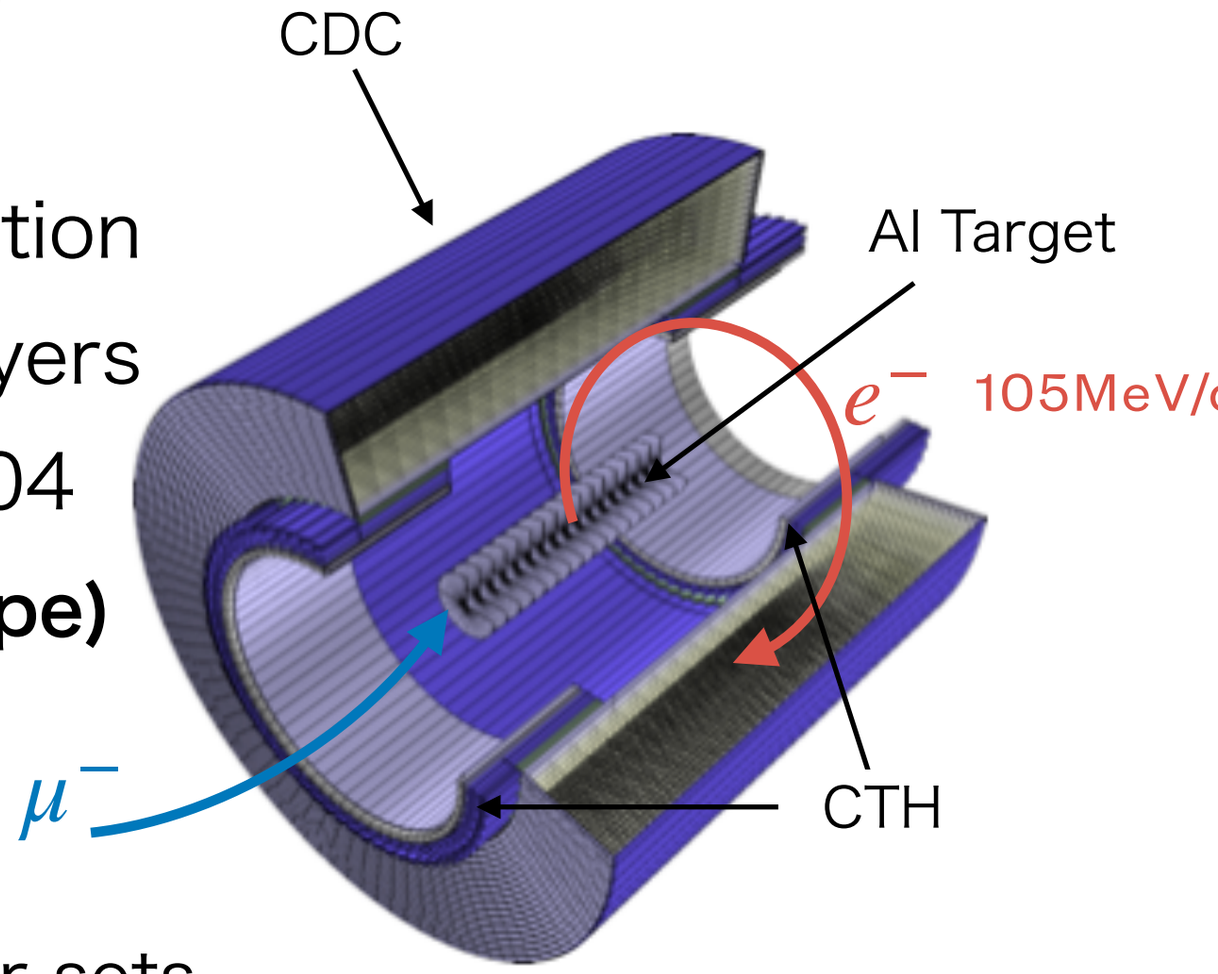
Cylindrical detector system

CDC (Cylindrical Drift Chamber)

- Evaluate particle momentum
- Measurement of particle hit position
  - 4986 sense wires,18 stereo layers
- Readout electronics : RECBE x104

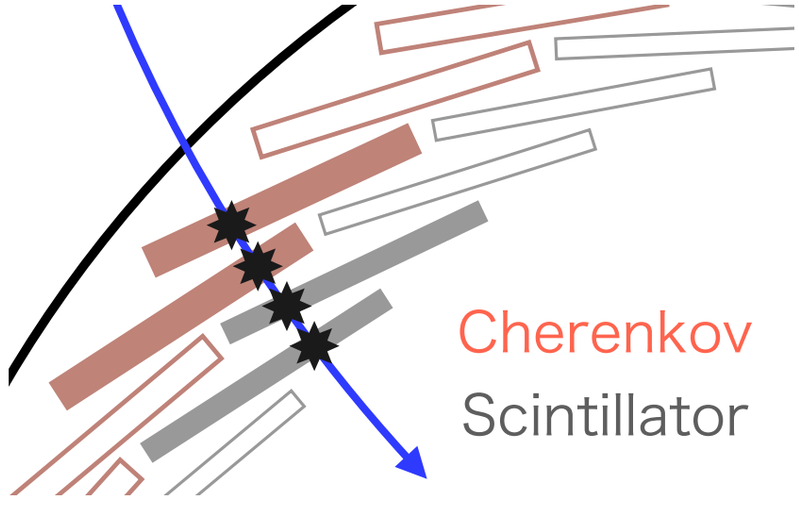
CTH (Cylindrical Trigger Hodoscope)

- Measure event timing
- Trigger counter
  - Cherenkov & Scintillation counter sets



CTH trigger rate > 90 kHz

- 4 fold coincidence
- Accidental coincidence & low-E electron dominant



Trigger system

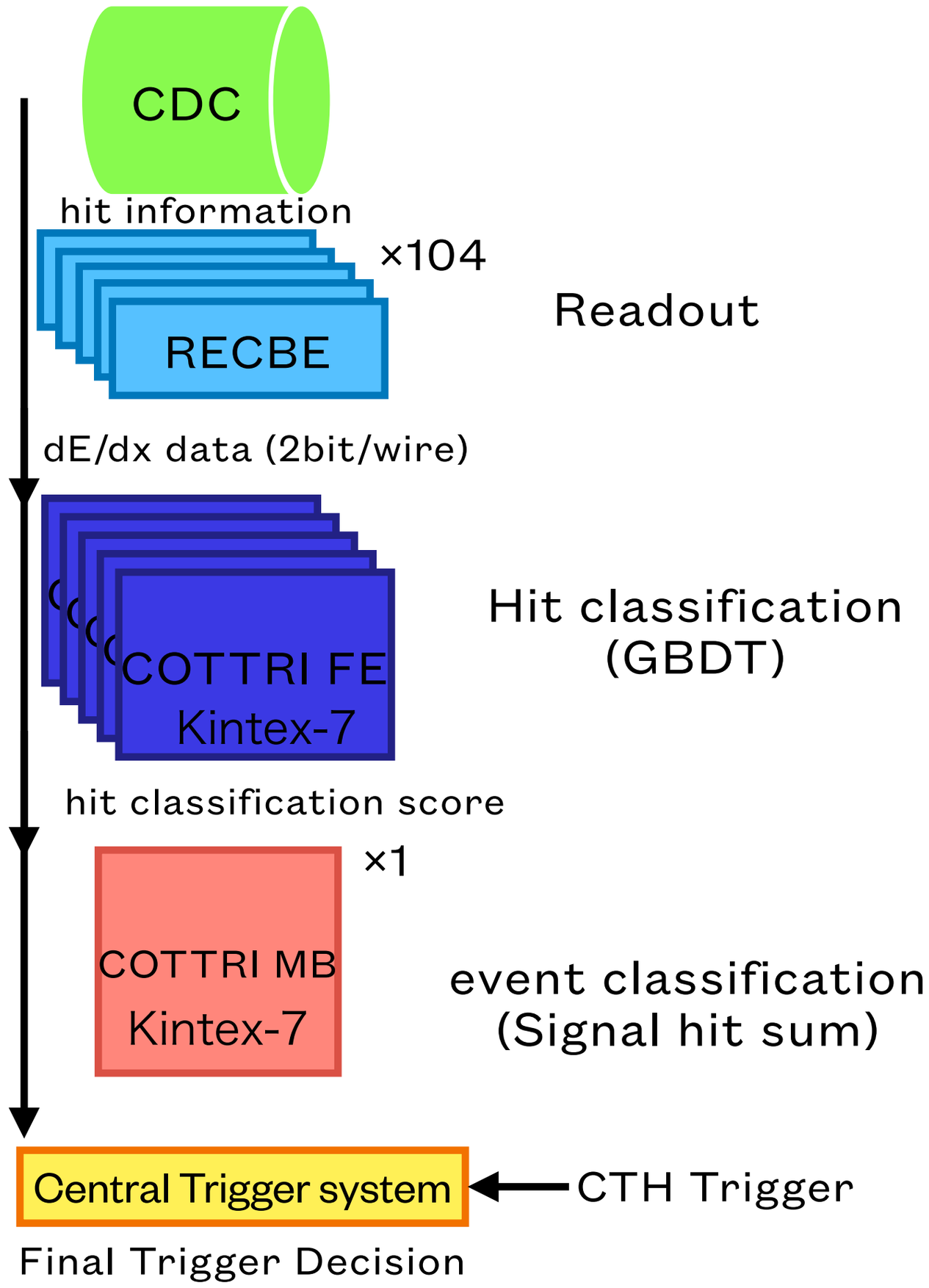
Requirement

- Trigger rate 13 kHz
  - constraints from DAQ
  - CTH trigger : 90kHz
- Latency <7 μs
  - Constraint by RECBE buffer time

Expected performance

- Signale efficiency 96%
  - w/ the required trigger rate
- Latency 3.2 μs

Improve
by Neural Network!



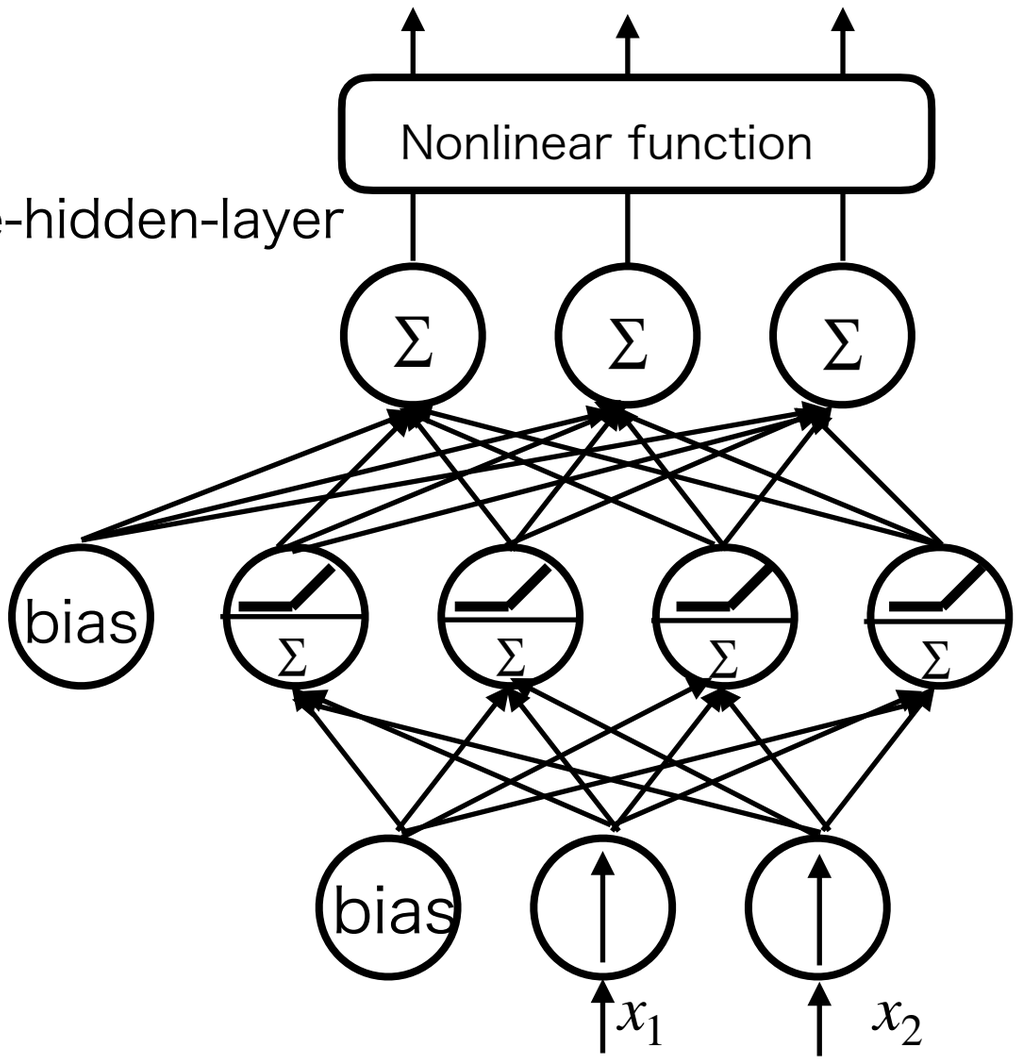
*GBDT(Gradient Boosted Decision Tree) is one of the Machine Learning Algorithm

Why Neural Network?

- Efficient finding of signal electron
  - Neural networks learn relationships between neurons in the form of weights
- FPGAs are a good match for NN calculations
  - The calculation of a neural network is essentially a non-linear function, an iterative inner product calculation. High energy efficiency hardware implement and massively parallel computing are heavily demand.
- Tools for building machine learning models on FPGAs have become available in recent years.

  -> later in this slide

Ex)One-hidden-layer



$$\mathbf{x}_m = g_m\left(\mathbf{W}_{m,m-1}\mathbf{x}_{m-1} + \mathbf{b}_m\right)$$

hls4ml

Hit classification

- Remove wires with multiple hits

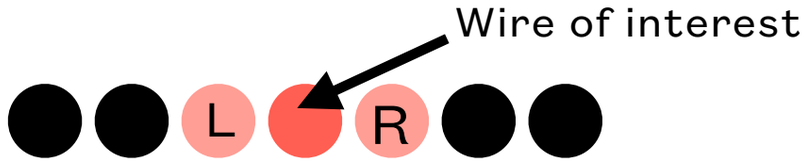
  to eliminate low energy electrons staying in the same cell

- Machine learning to score hit information for each wire based on energy loss and local patterns

  Input feature : ΔE @ wire-of-interest

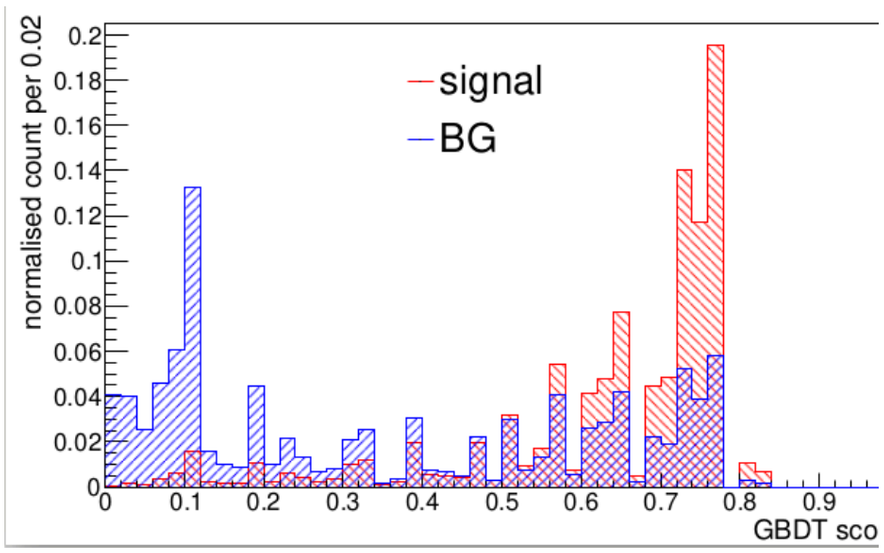
  ΔE @ wire-of-interest

  ΔE @ wire-of-interest

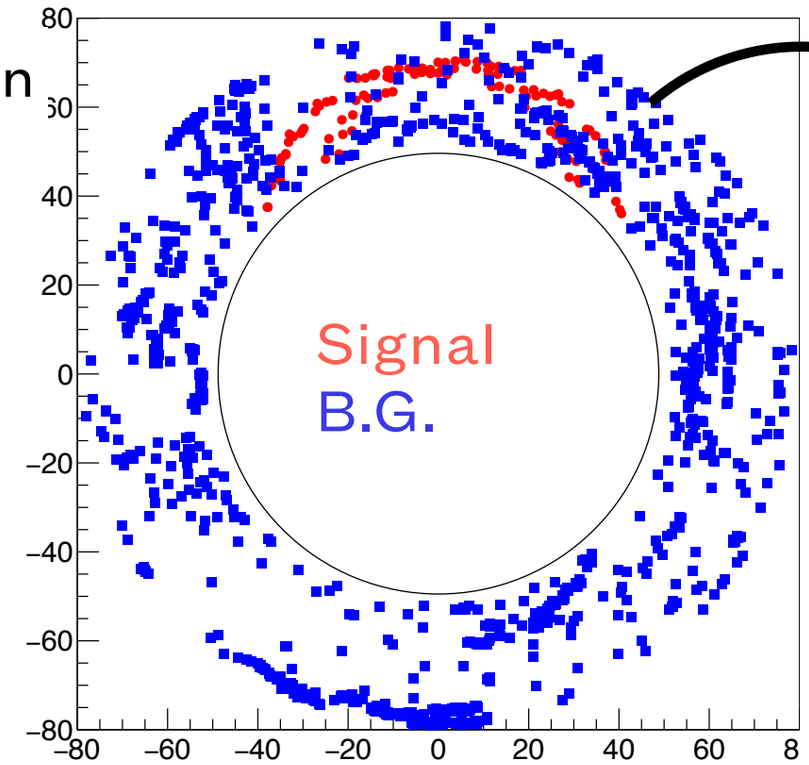


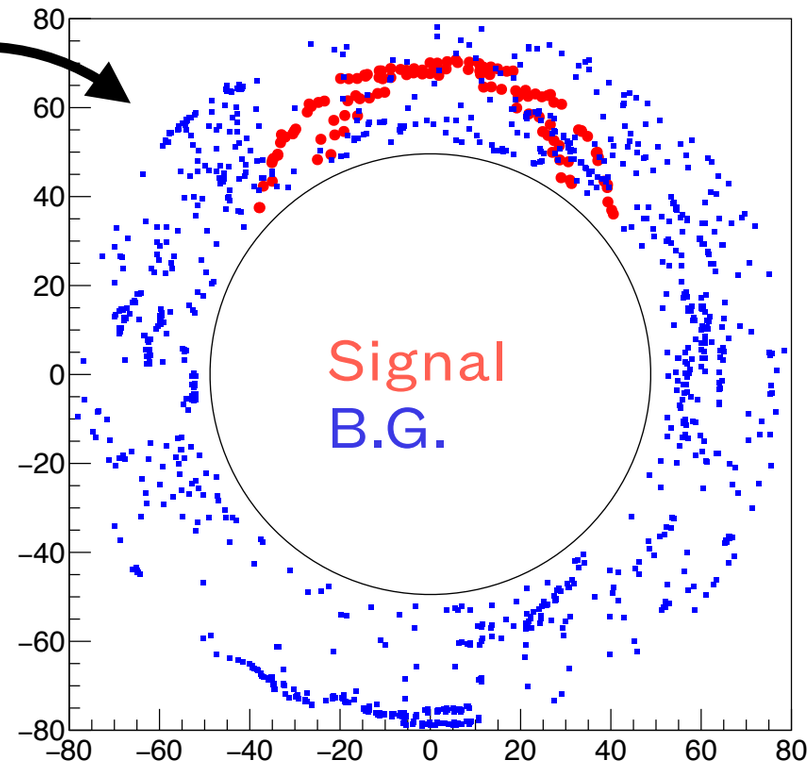
Layer ID of wires

GBDT output score distribution



Scoring by LUTs





Event Classification Algorithm

Previous algorithm

・Hit flag

・hit classification score

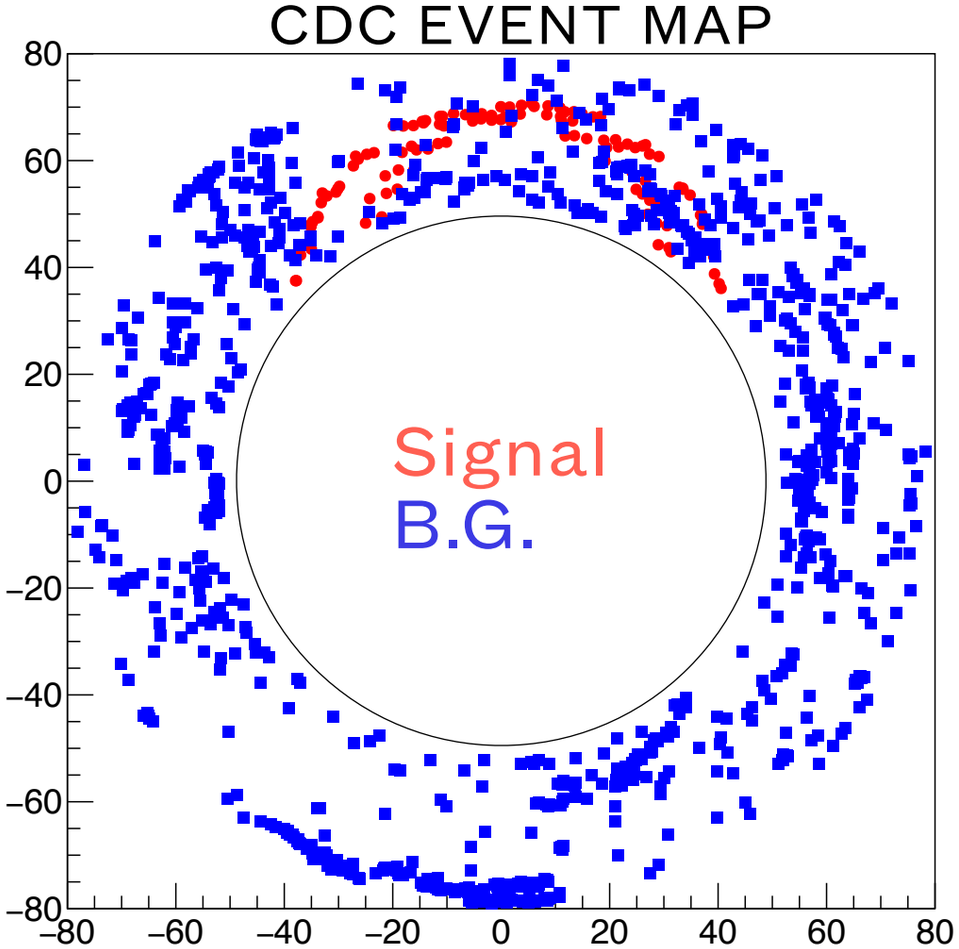
B.G.-like < $S_{threshold}$ < Signal-like

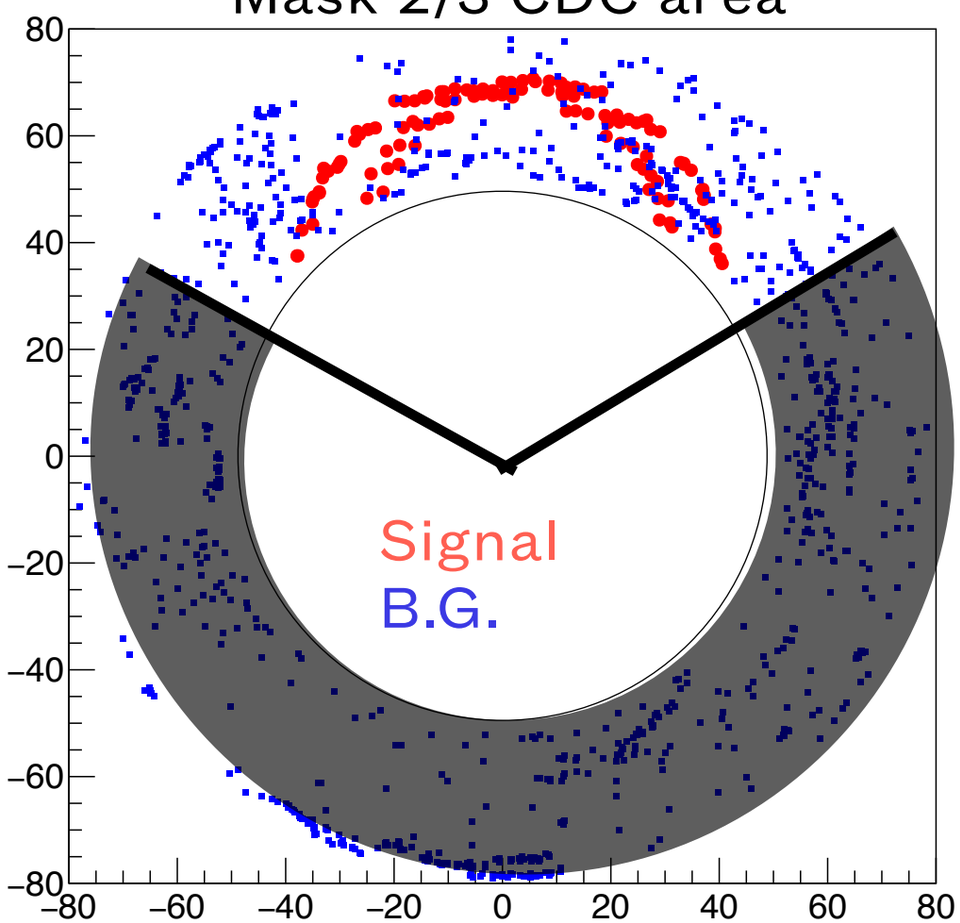
New algorithm

・2bits or more score information

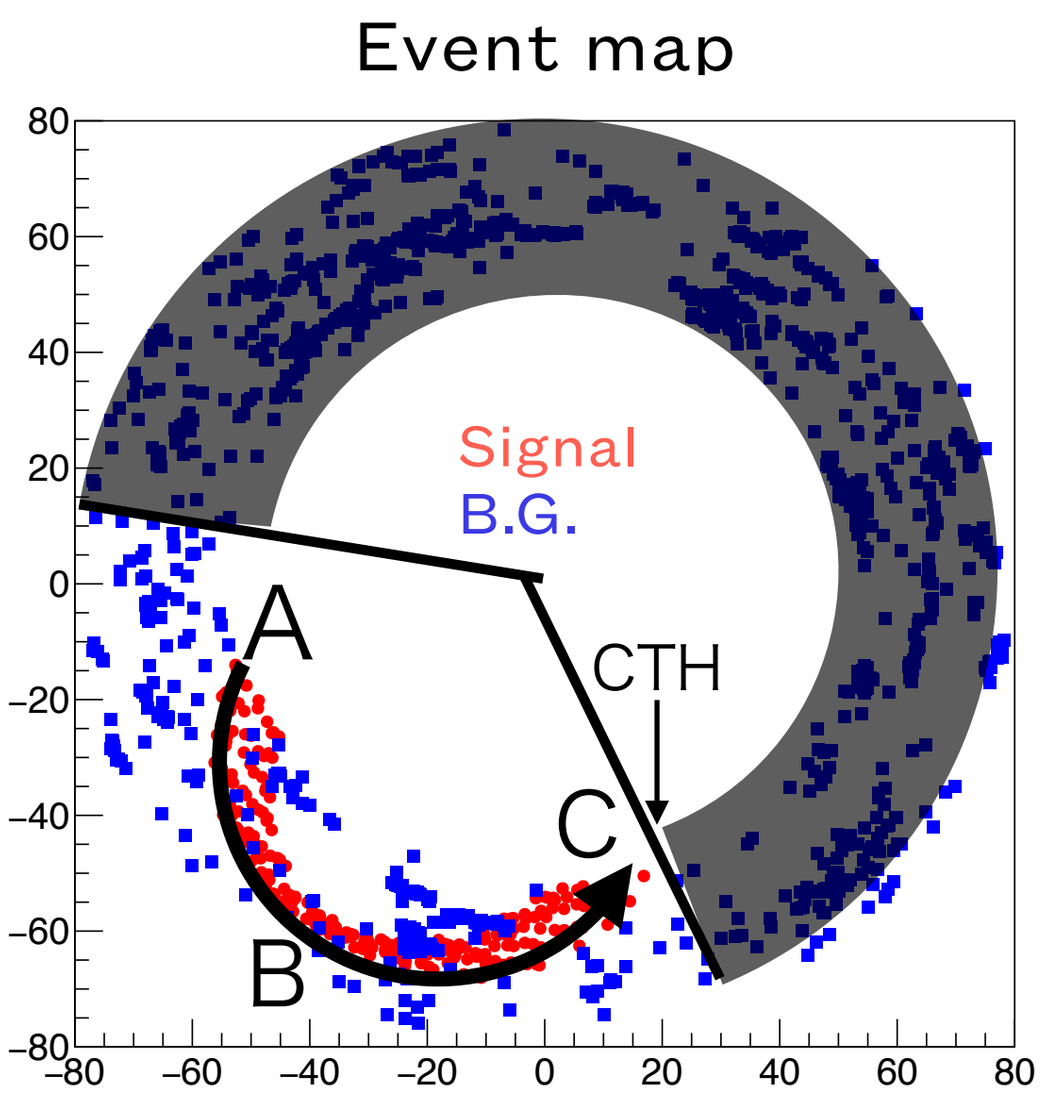
Coincidence on
signal hit sum and CTH flag

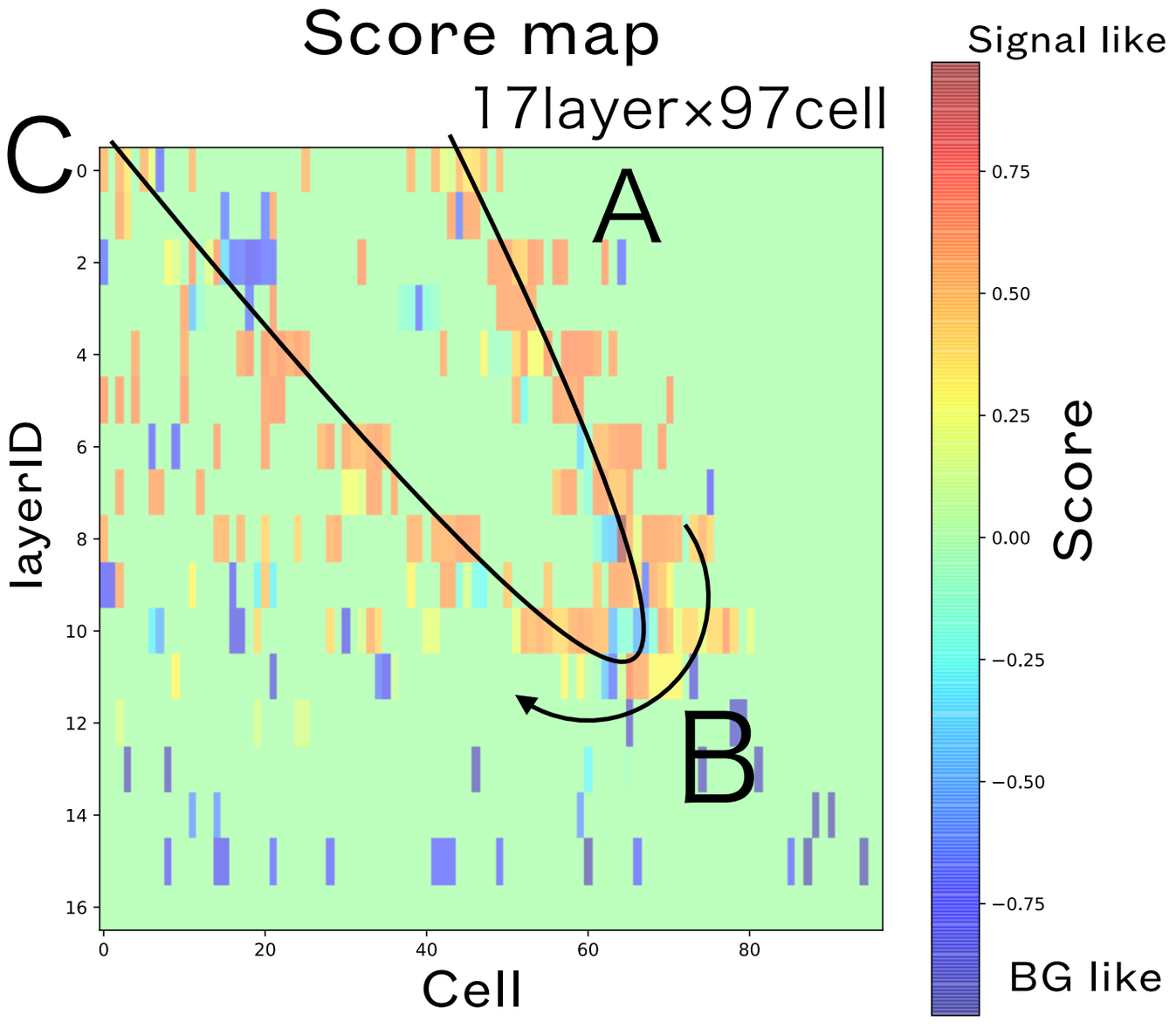
・Classification by Neural Network





Concept of the event-classification





- Now event classification problem changes into image classification problem.
  - Pattern recognition of the signal electron trajectory with hit score.
- Neural network design study is ongoing.
  - ->Feasibility check

How to implement your NN model on FPGA?

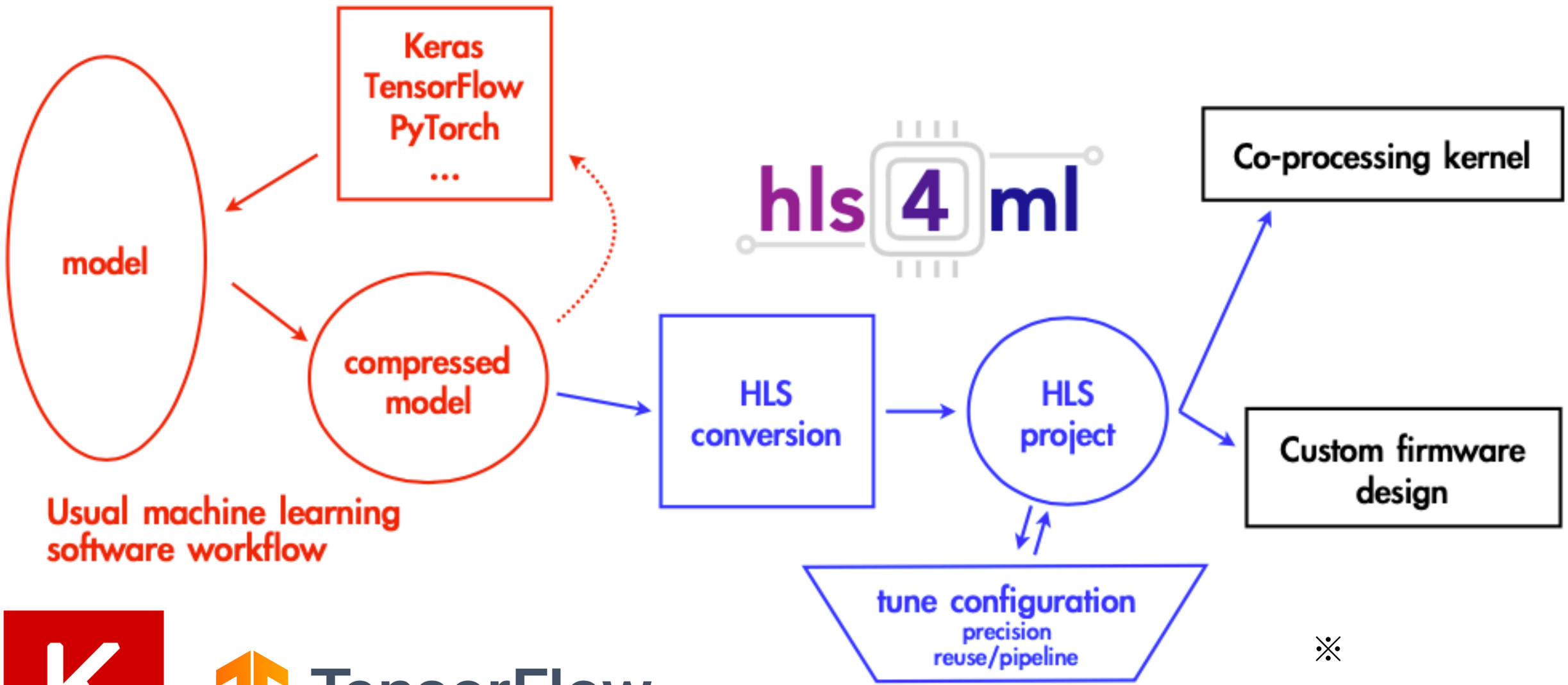
&

Feasibility Check

- How long latency ?
- How large resource?

HLS4ML concept

Work flow to translate a model into a FPGA implementation using hls4ml



※

※Fast inference of deep neural networks in FPGAs for particle physics
arXiv:1804.06913v3 [physics.ins-det] 28 Jun 2018

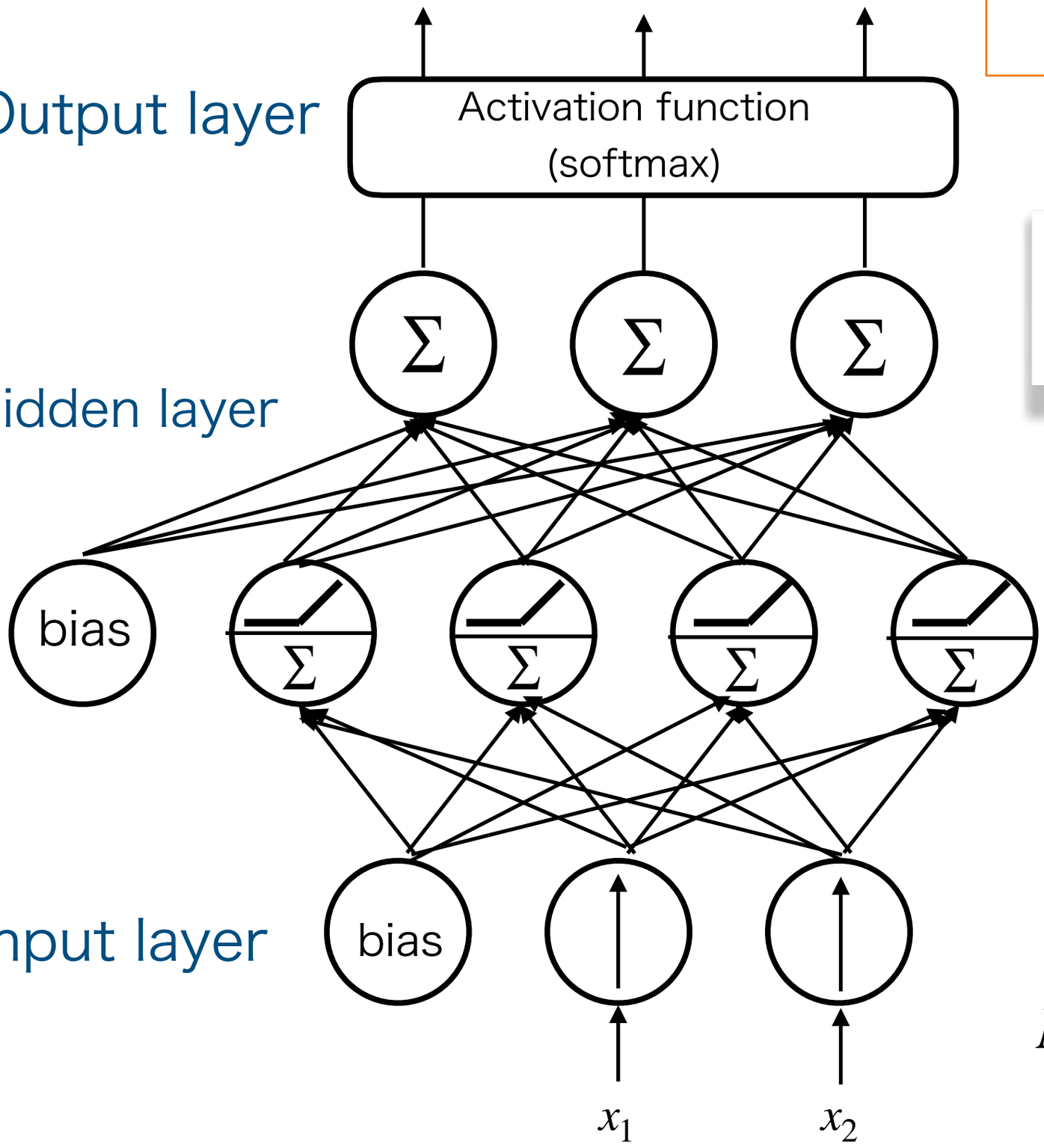
Key metrics for an FPGA implementation

- **Latency**
  - The total time required for the algorithm to complete
- **Resource usage**
  - BRAM : Block RAM
    - Hardened RAM resource
  - DSPs : Digital Signal Processor
    - Performs multiplication and other arithmetic in the FPGA
  - FF : Flip Flops
    - Register data in time with the clock pulse
  - LUTs : Look Up Table(Logic)
    - Generic functions on small bit width inputs.

These limitations become constraints.

Neural Network on FPGA

Ex)One-hidden-layer



Vector of neuron output values at each layer

Matrix of weight

$$\mathbf{x}_m = g_m\left(\mathbf{W}_{m,m-1}\mathbf{x}_{m-1} + \mathbf{b}_m\right)$$

Activation Function
- Precomputed and Stored in BRAMs

Multiplication
- DSPs

Addition
Logic cells

$$N_{multiplication} = \sum_{m=1}^{M} N_{m-1} \times N_m$$

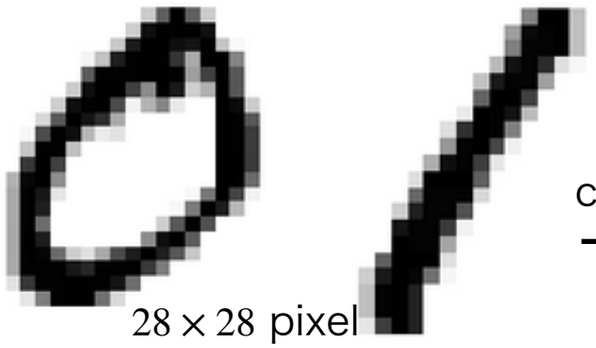
$\propto$DSPs

現行のCOTTRI MB*に対しどのくらいのサイズのネットワークが構築できそうか、MNIST手書き文字データセットで試してみた

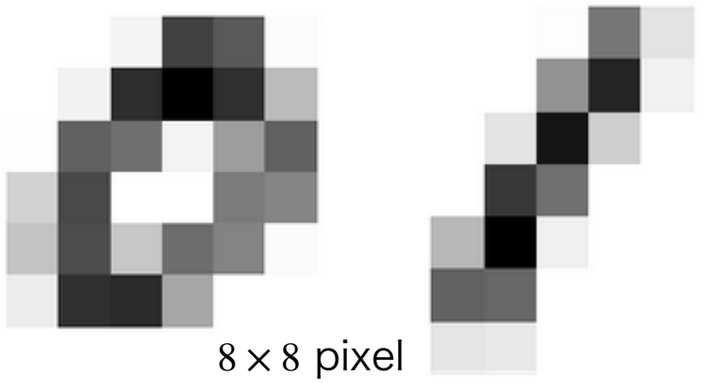
※Xilinx Kintex-7 FPGA with part number
xc7k355tffg901-1
is installed on COTTRI MB

MNIST handwritten character data test bench

Two classes of classification, 0 and 1



compress →



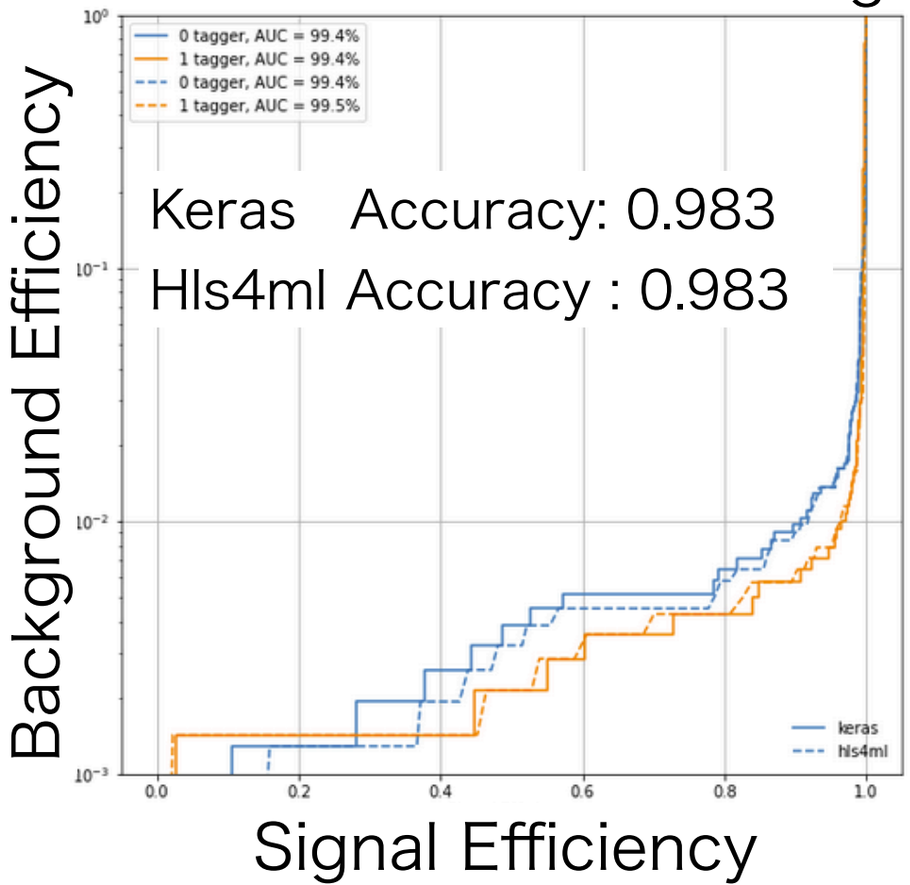
$$N_{multipulication} = \sum_{m=1}^{3} N_{m-1} \times N_m$$
$$= 2080 + 528 + 34$$
$$= 2642$$

∝Minimum DPSs required

3layer Dense MLP

64 inputs → 32 nodes, Activation: ReLU → 16 nodes, Activation: ReLU → 2 outputs, Activation: SoftMax

FPGA: Kintex-7 xc7k355t-ffg901-1



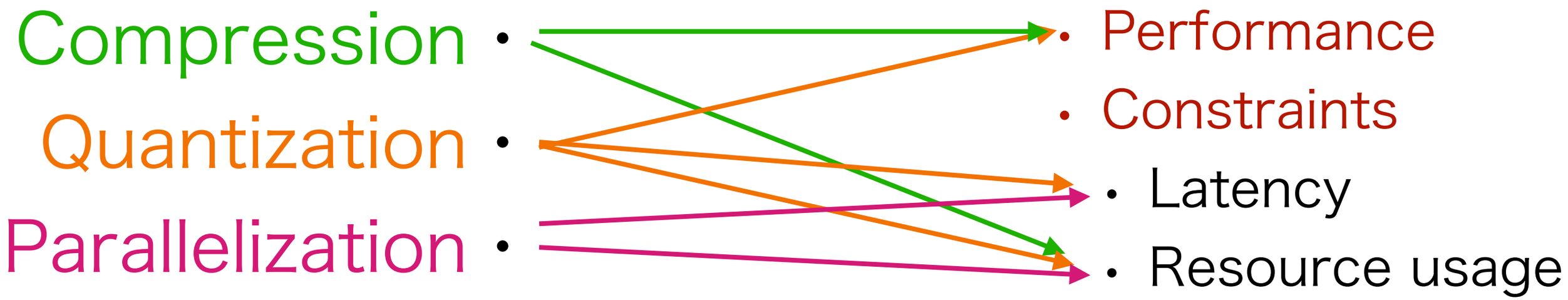
HLS4ML estimation

Latency
0.170us (34 cycles)

Name	BRAM_18K	DSP48E	FF	LUT
Estimation	1	2510	129542	51910
Available	1430	1440	445200	222600
Utilization(%)	~0	174 👎	29	23

Efficient network design

- Compression
  - Reduce the number of synapses or neurons
- Quantization
  - Reduces the precision of the calculations(weights, biases, etc)
- Parallelization
  - Reduce resource utilization by reusing DSPs(Tune how much to parallelize the multiplications required for a given layer computation)

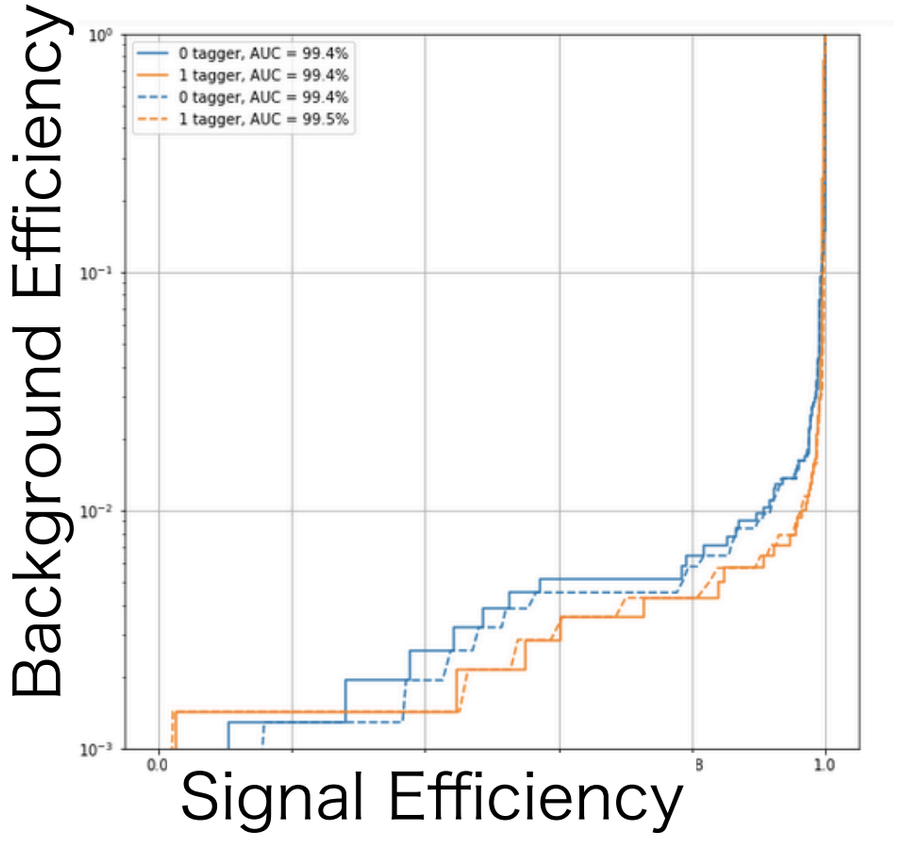


※Fast inference of deep neural networks in FPGAs for particle physics
arXiv:1804.06913v3 [physics.ins-det] 28 Jun 2018

Parallelization and Quantization

Parallelization

- Data precision<16,6>*
- Reuse factor ==2

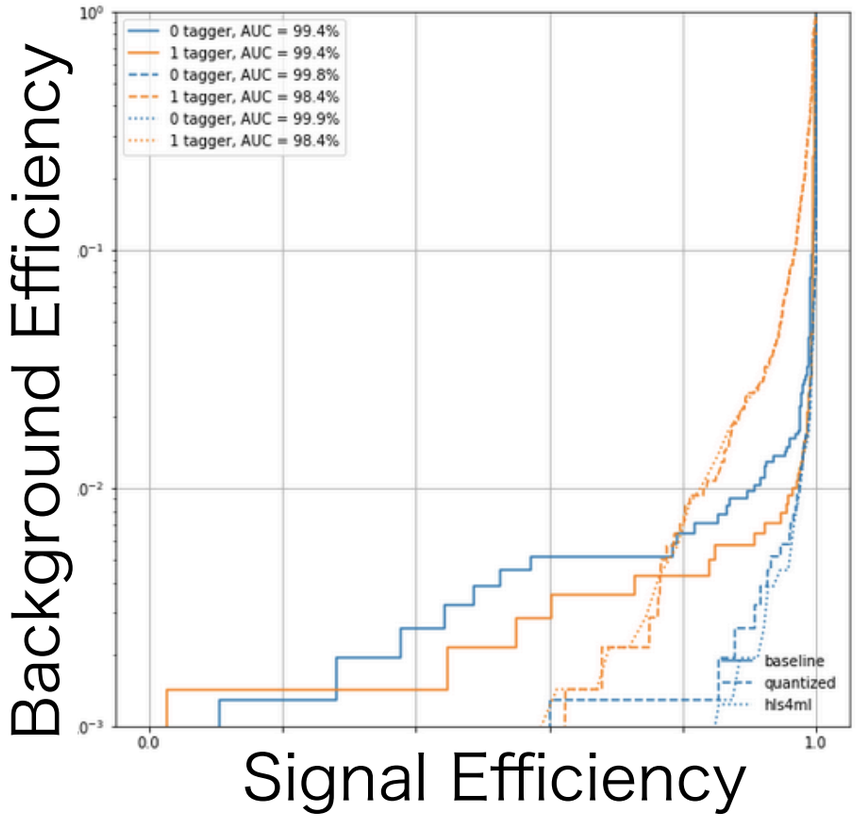


Latency
0.310us (62 cycles)

Name	BRAM_18K	DSP48E	FF	LUT
Estimation	1	1296	96412	87181
Available	1430	1440	445200	222600
Utilization(%)	~0	90	21	39

Quantization

- Inner layer data precision<7,1>*
- Reuse factor == 1



Latency
0.124us (25cycles)

Name	BRAM_18K	DSP48E	FF	LUT
Estimation	1	842	56666	65939
Available	1430	1440	445200	222600
Utilization(%)	~0	58	12	29

<X,Y>* X : total number of bits, Y : number of bits representing the signed number above the binary point

Comparison of the 3 cases

- Case1 Bench mark
- Case2 Reuse Factor ==2
- Case3 Inner Layer Data precision <7,1>

	latency	Reuse
Case1	0.170us (34 cycles)	1
Case2	0.310us (31 cycles X 2)	double 2
Case3	👍 0.124us (25 cycle)	1

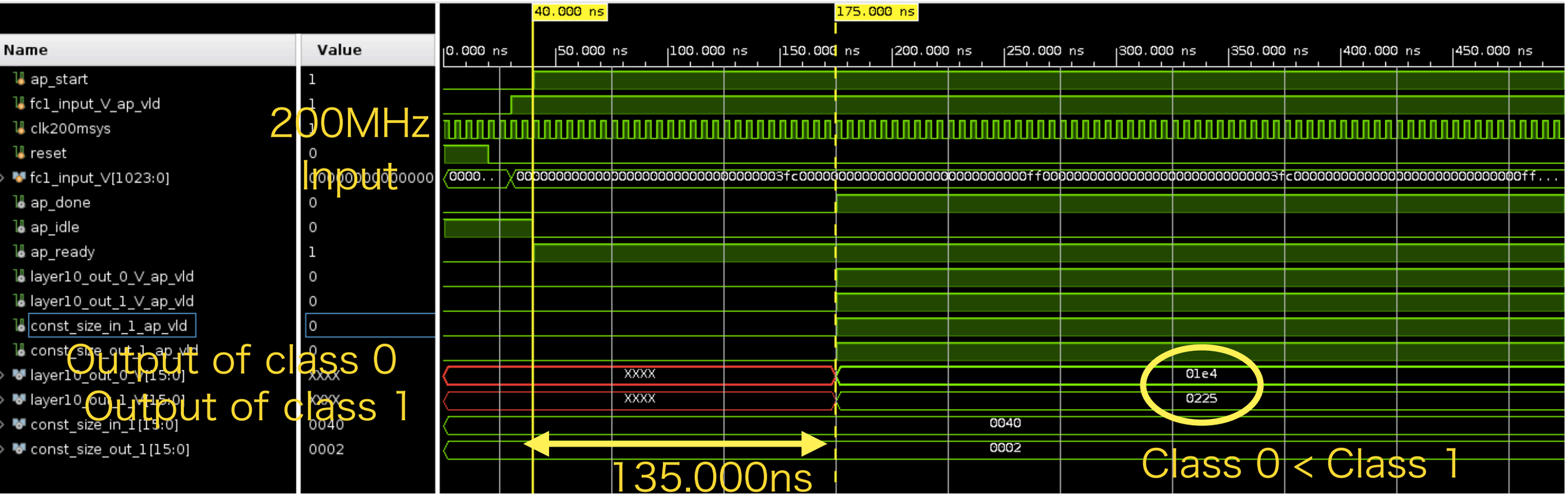
Name	BRAM_18K	DSP48E	FF	LUT
Available	1430	1440	445200	222600
Case1 Utilization(%)	~0	174	29	23
Case 2 Utilization(%)	~0	90 Half	21	39
Case 3 Utilization(%)	~0	👍 58	12	29

Manipulating the accuracy of the data seems to be effective both in reducing latency and in reducing DSP usage.

Simulation Result

The third pattern was High-Level-Synthesised in VIVADO HDL, and then simulated in VIVADO.

I input the signal for handwriting 1



The latency is two clocks longer, but the results are roughly as expected.

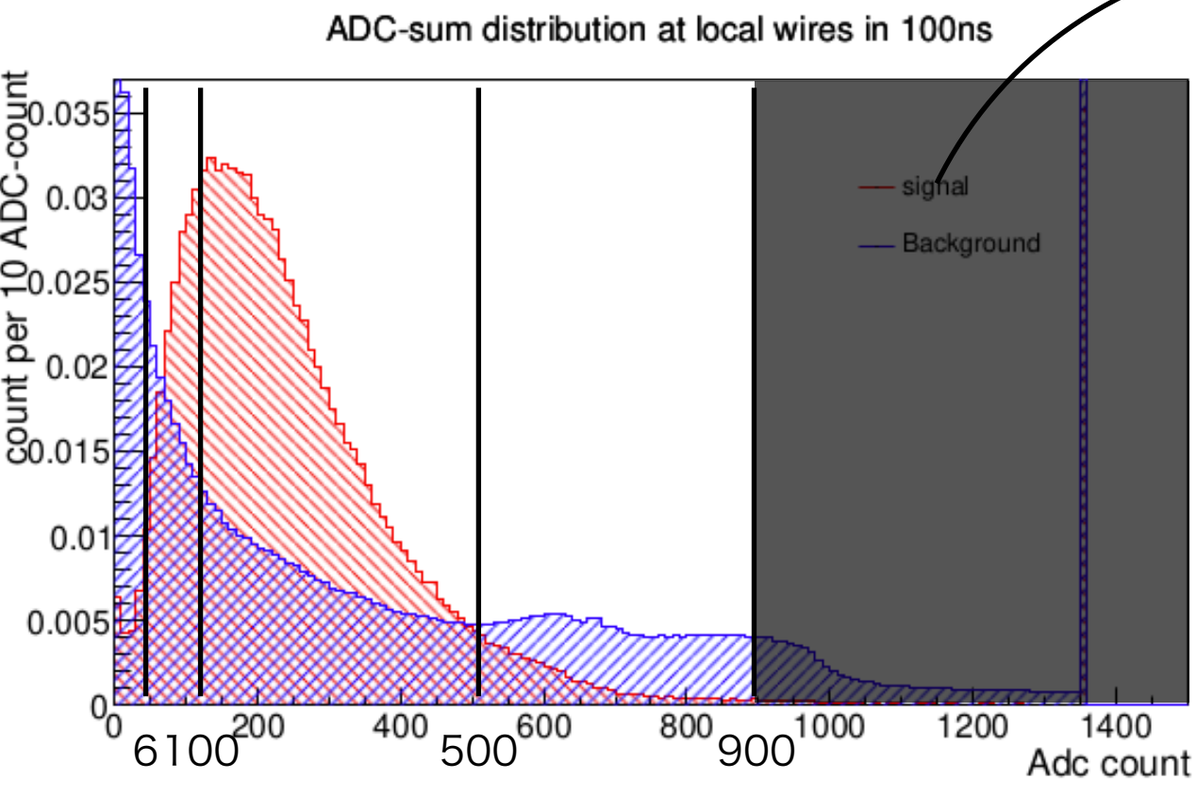
Summary

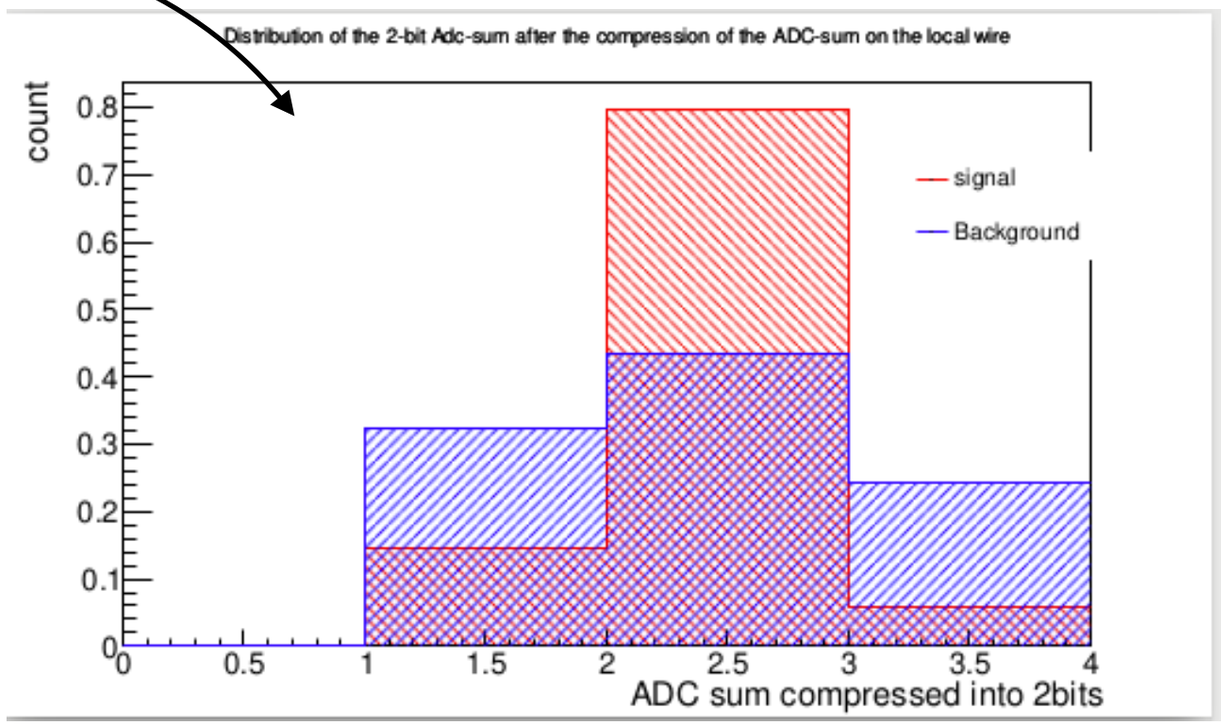
- The basis for the trigger system is already in place
- We are looking for more powerful event classification algorithms and neural networks are a good candidate, so I'm studying simple MLP first.
- Checking the feasibility of Neural Network implementations using MNIST handwritten character data sets

Back up

ADC-Sum compression

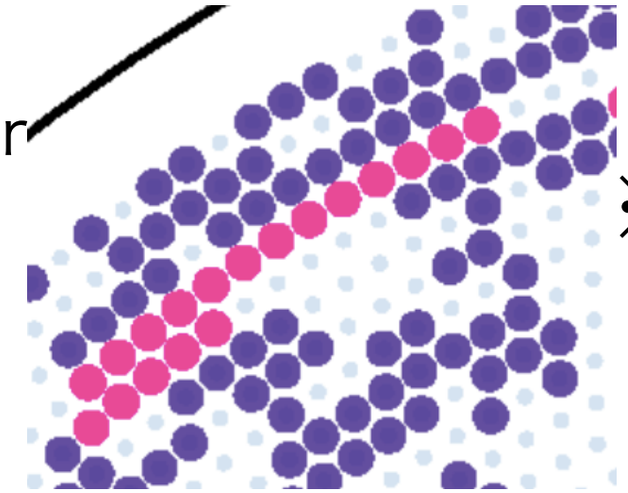
0 1 2 3





Hit characteristics

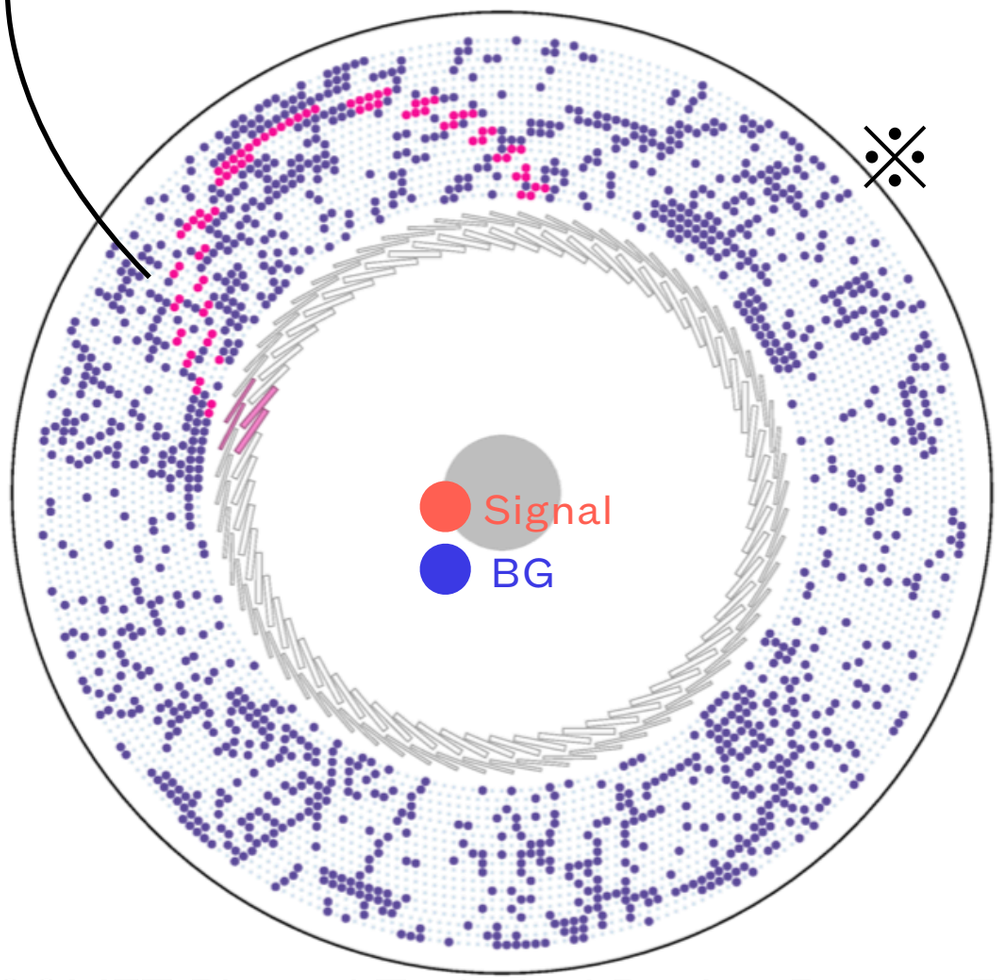
Continuous hits
on the same layer



※

Scope

Simulated event display



※COMET Phase-I Technical Design Report Fig43

- **Signal hit**
  - Continuous hits in the same layer
    - Spiral trajectory on CDC
  - Single hit in the same wire
  - Not reaching outer layers
  - MIP Level energy loss
- **Background hit**
  - Low energy electrons
    - Helical trajectory in the same cell
    - Multi hits in the same wire
  - Protons
    - High momentum
    - Large energy loss

Based on these characteristics, input features were chosen

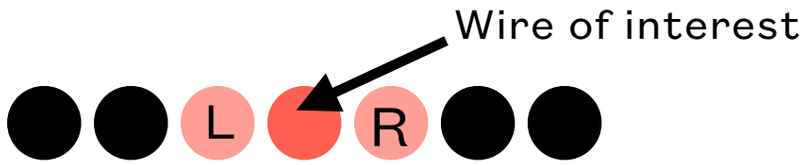
GBDT input feature

Hit classification by GBDT@COTTRI FE

※Cut multi hits on the same wire before hit classification

★2bit Edep data

- Energy loss information
- 0 if the interest wire has no hits



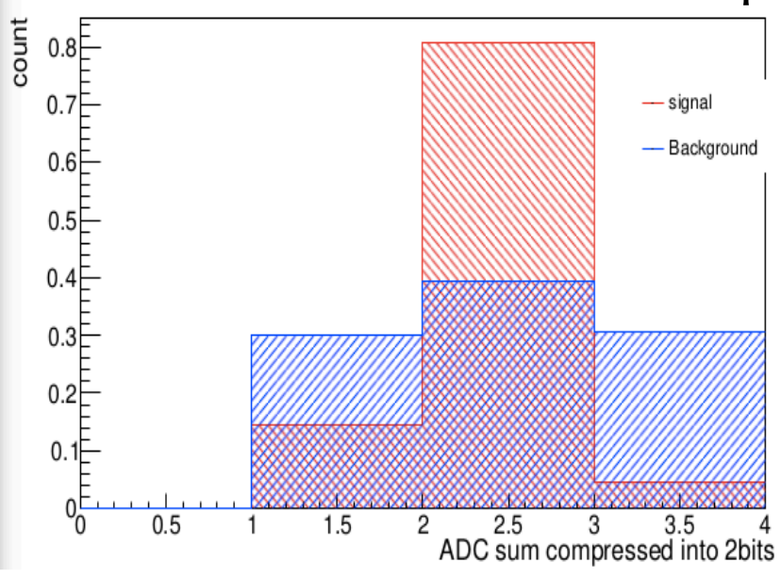
★LayerID: radial distance from the CDC center

- The distribution of Signal has a peak, but B.G. does not.
- Cut hits in innermost & 3 outermost layers
  - Signal electrons are hard to reach the outer layer

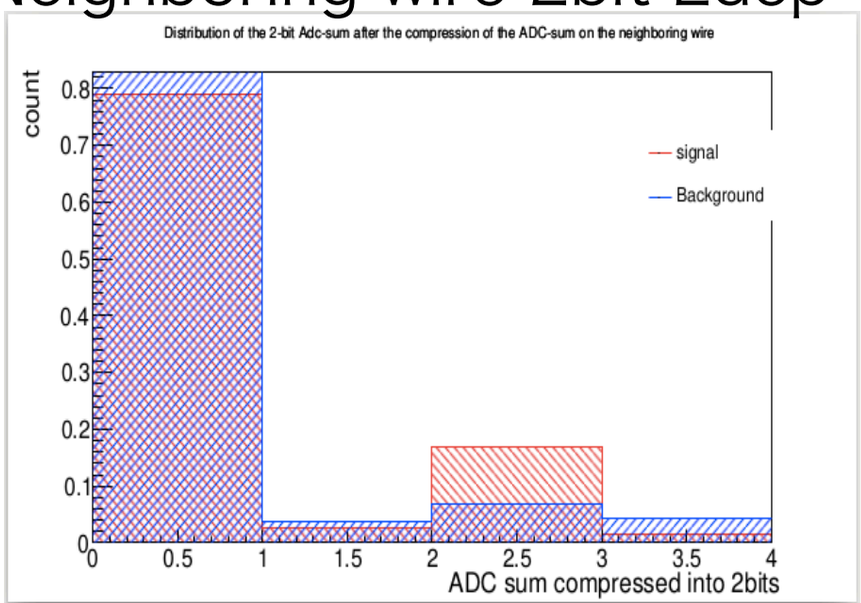
- Right wire 2bit Edep
- Interest wire 2bit Edep
- Left wire 2bit Edep
- LayerID

GBDT

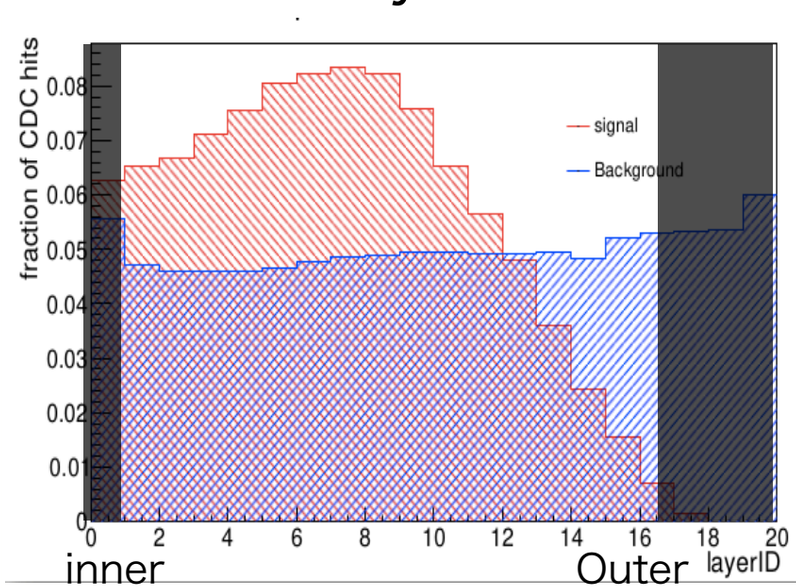
Interest wire 2bit Edep



Neighboring wire 2bit Edep

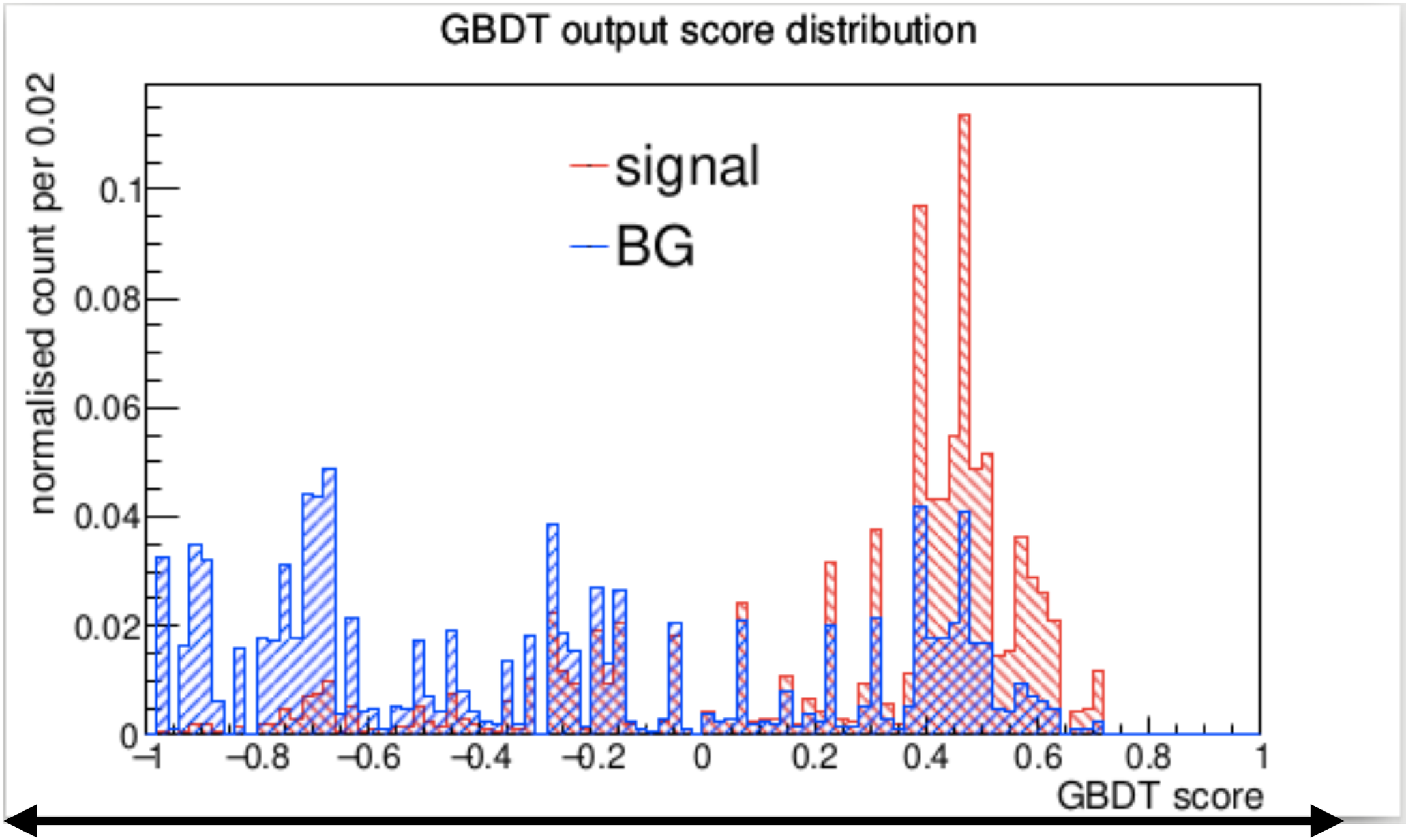


LayerID



GBDT output distribution

Hit classification by GBDT@COTTRI FE



BG like

Signal like

This hit classifier gives hit scores to each wire hit

CDC
4986ch

RECBE

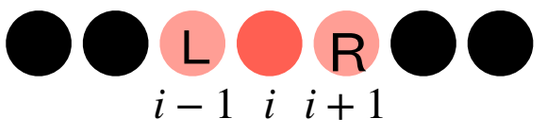
10 bit data
@30MHz sampling

Sum up 3samples

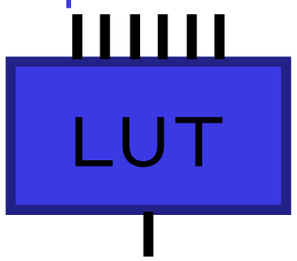
Compress to 2bit

32bit
×
5frames
×
10MHz
=
1.6Gbps

COTTRI FE



2bit/wire×3wires input
6input-LUTs



6bit output

Occupancy ~50%
(Current version)

2bit/channel
×
48channel/RECBE
×
10RECBE/FE
×
10MHZ
=
9.6Gbps/FE

6bit Score@10MHz

To COTTRI MB

CDC
4986ch

RECBE

10 bit data
@30MHz sampling

Sum up 3samples

Compress to 3bit

Transceivers (GTXs)on the FPGA
Max rate 6.25Gbps
(3.125Gbps/lane × 2 lane)

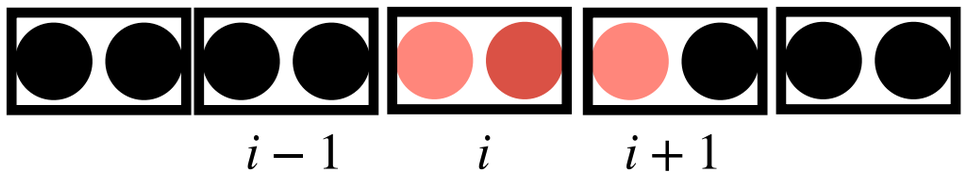
32bit
×
7frames
×
10MHz
=
2.24Gbps

Cluster化をするのはどこ?

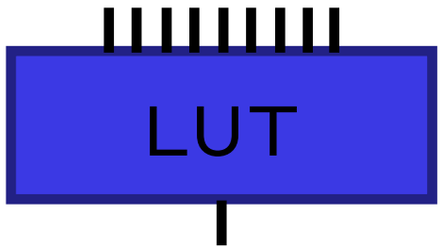
@RECBE?@FE?

COTTRI FE

Make cluster by connecting 2 adjacent wires



3bit/cluster×3clusters input
9input-LUTs



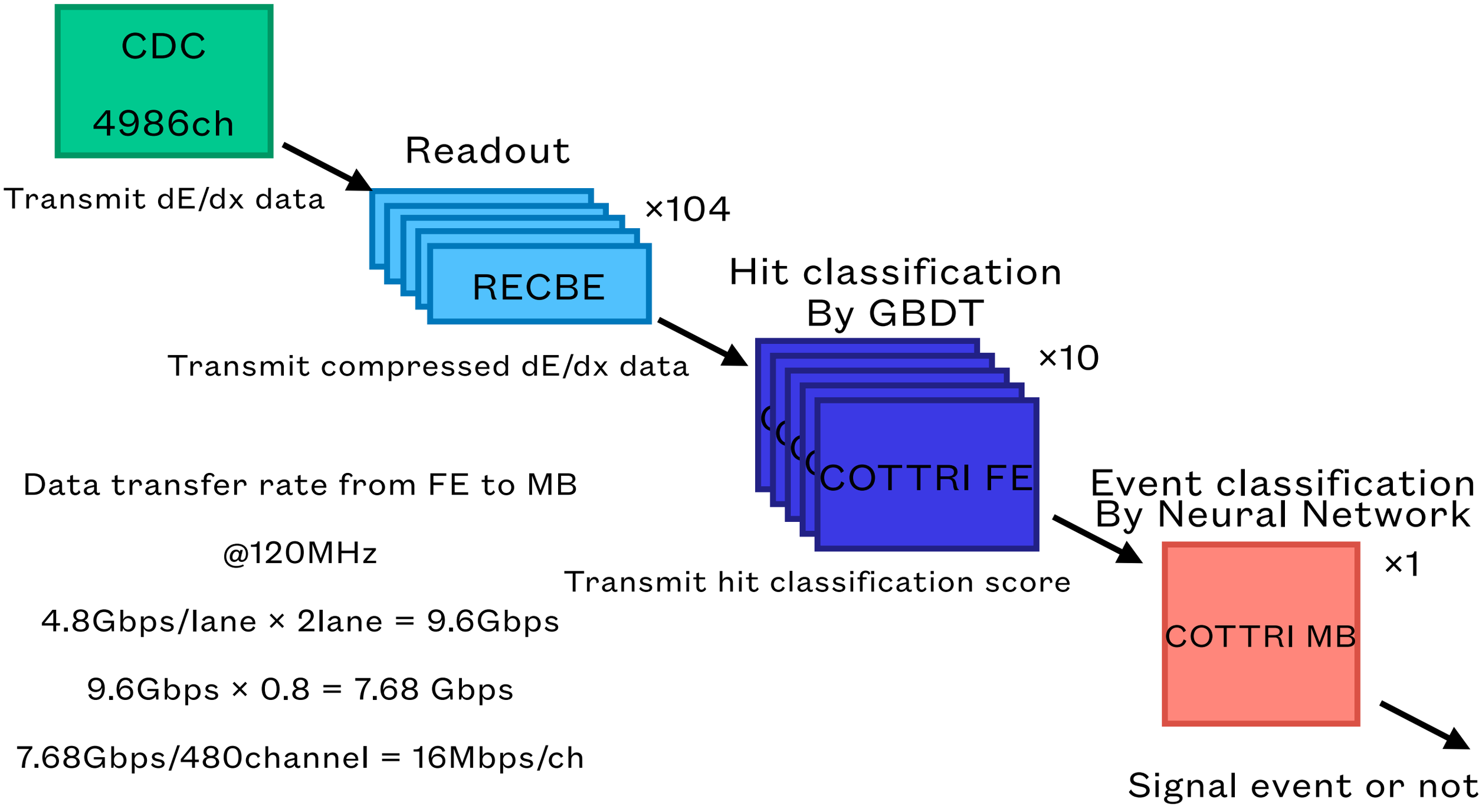
3bit output
Occupancy ~100%?

3bit/cluster
×
24clusters/RECBE
×
10RECBE/FE
×
10MHz
=
7.2Gbps/FE

3bit Score@10MHz

To COTTRI MB

Event Classification flow



Data transfer rate from FE to MB

@120MHz

4.8Gbps/lane × 2lane = 9.6Gbps

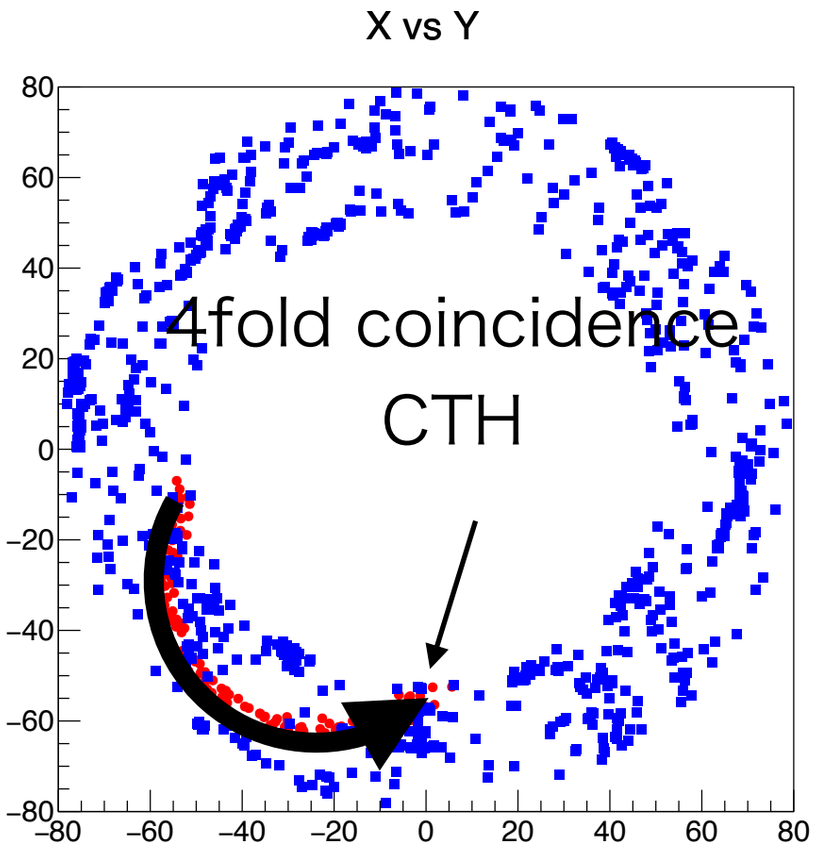
9.6Gbps × 0.8 = 7.68 Gbps

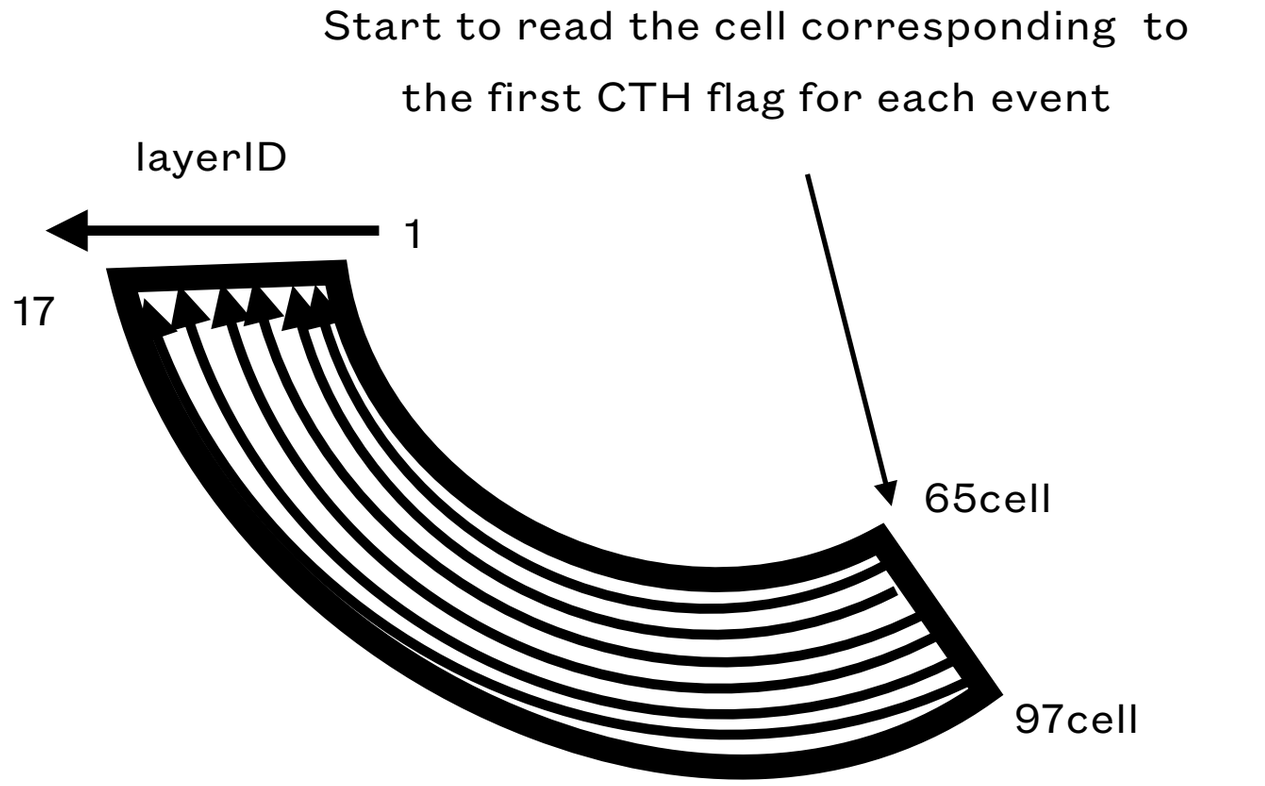
7.68Gbps/480channel = 16Mbps/ch

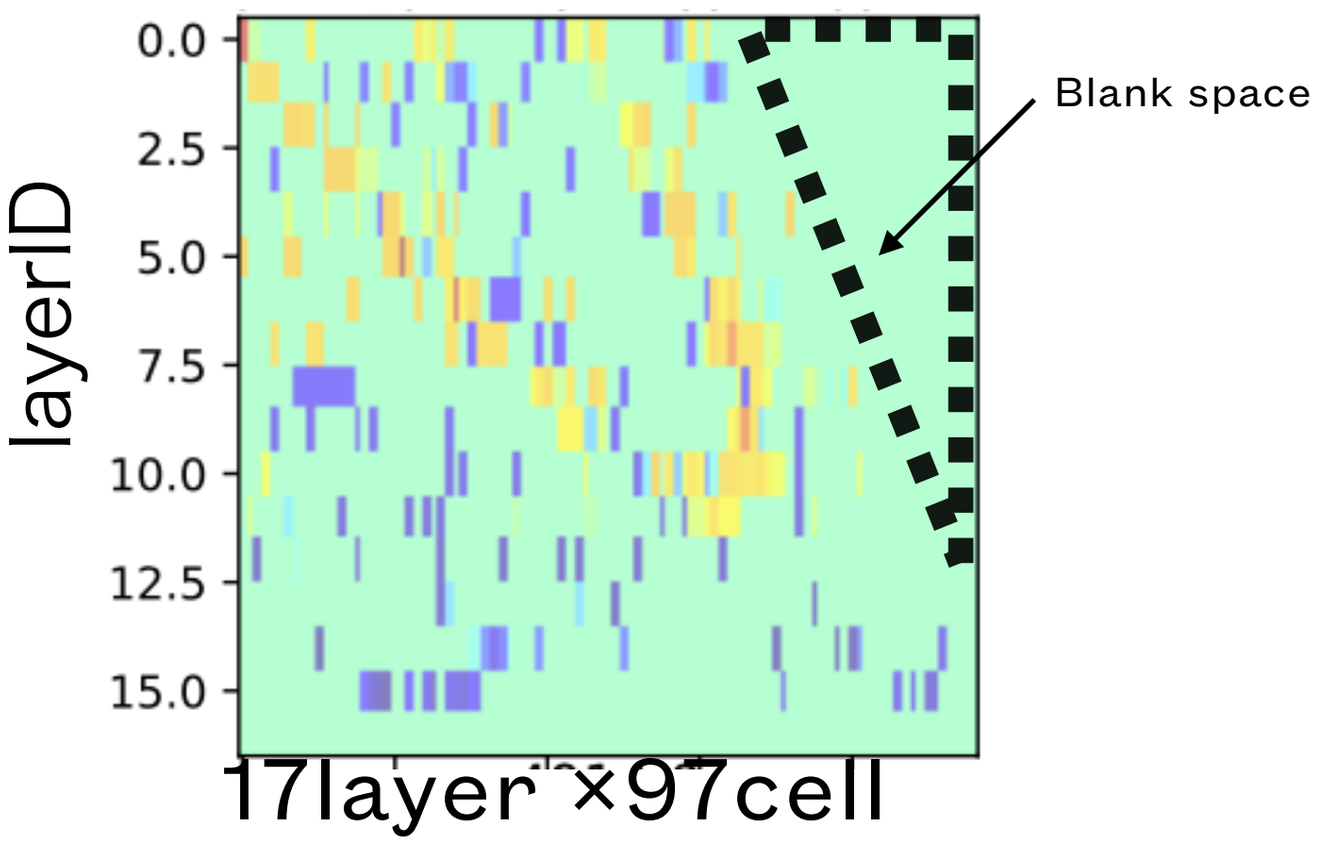
@10Mhz 16Mbps/ch->1.6bit/ch

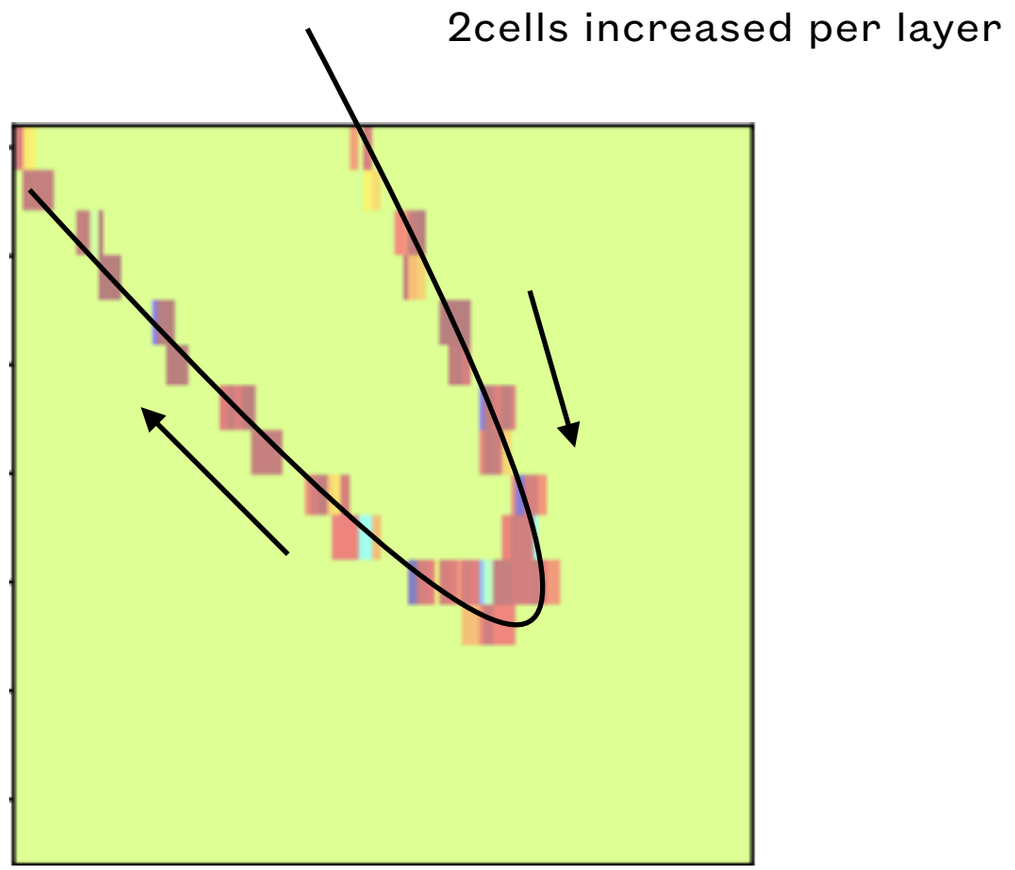
1Cluster/2ch ~3bit/ch

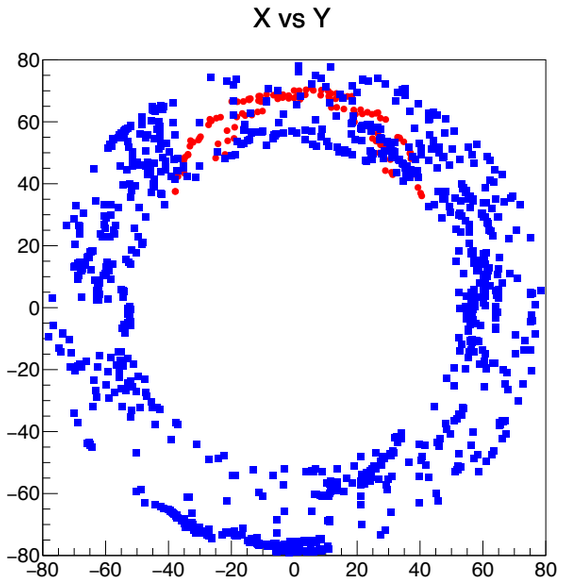
How to make event-classification input data

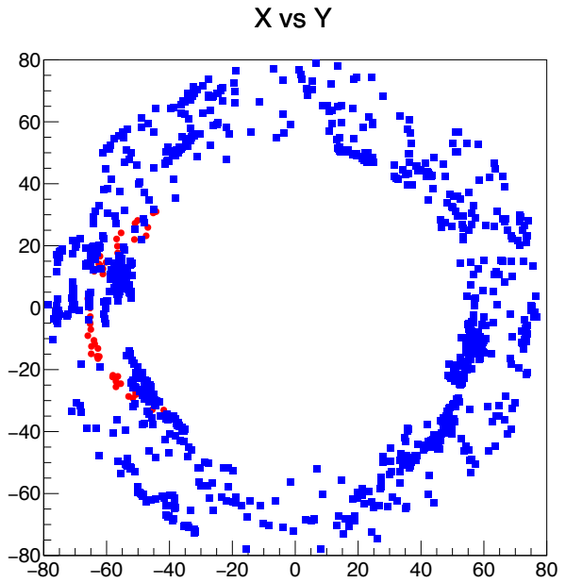


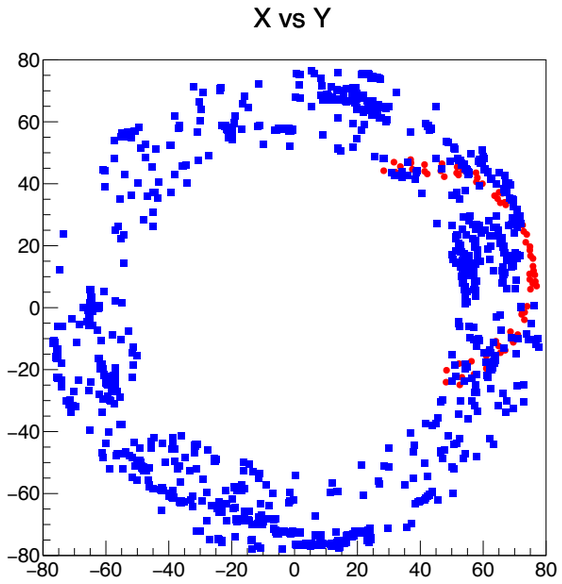


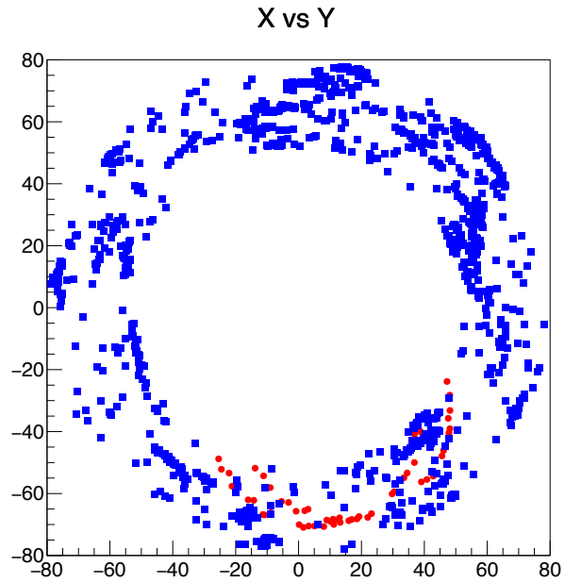


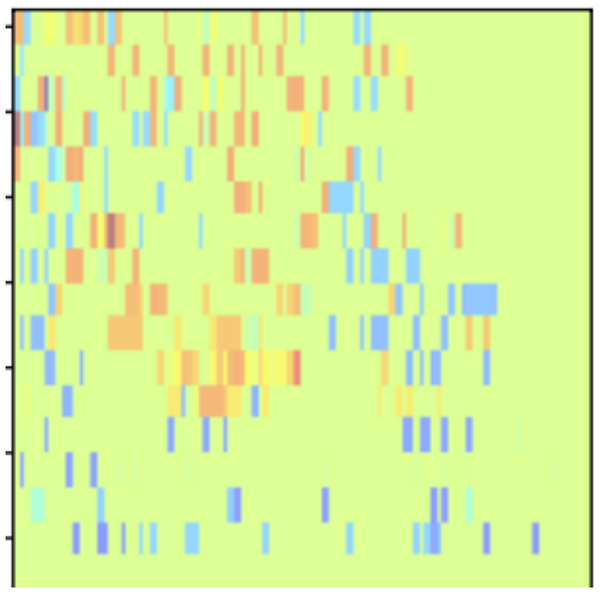


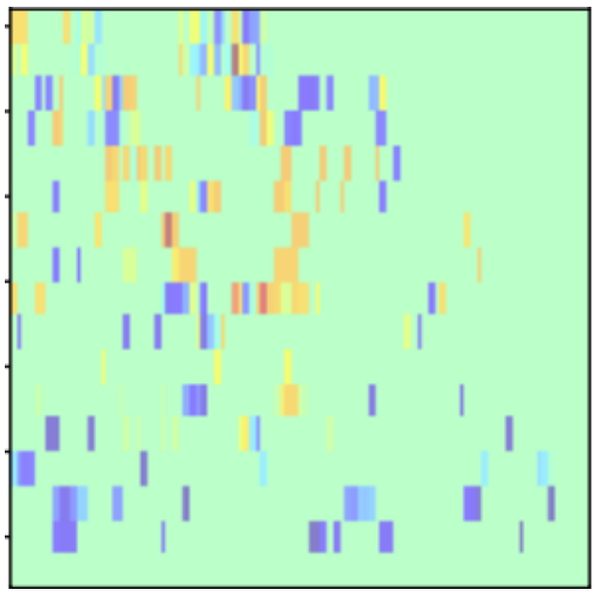

X vs Y

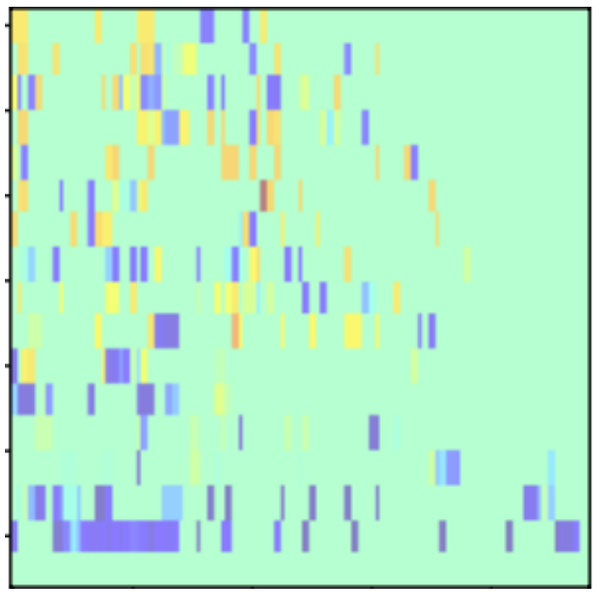

X vs Y

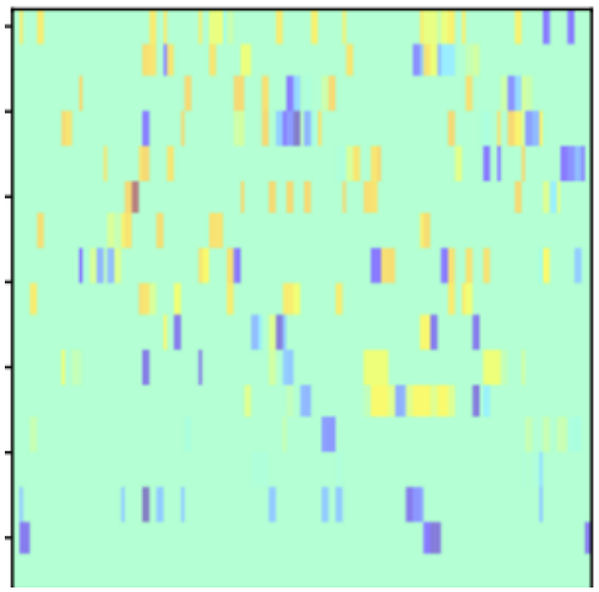

X vs Y

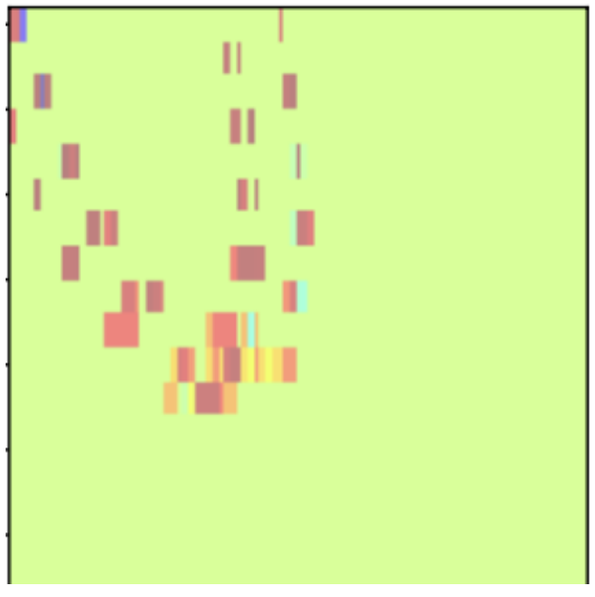

X vs Y

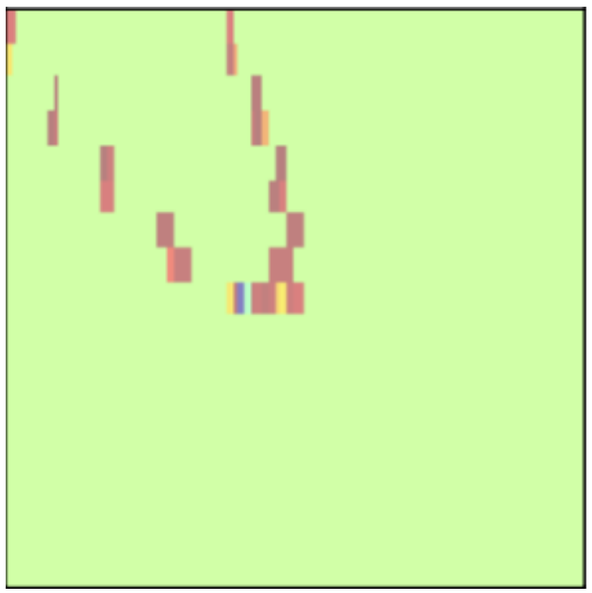


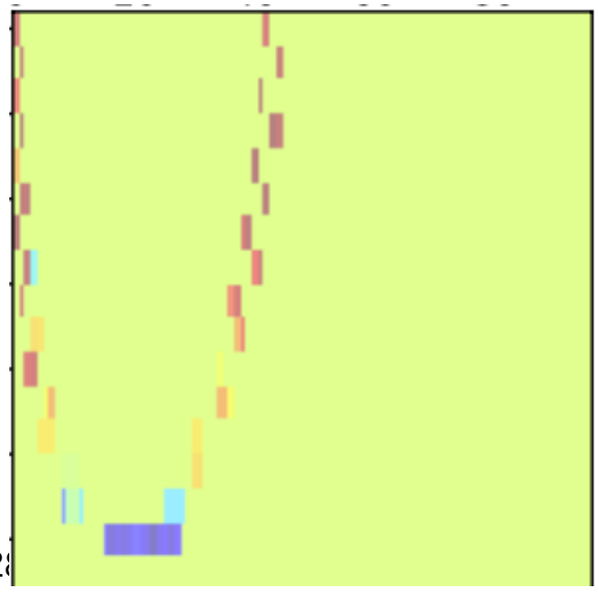


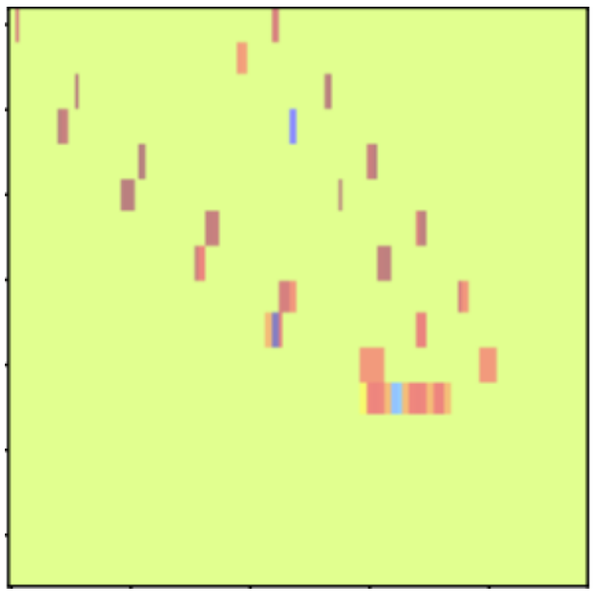


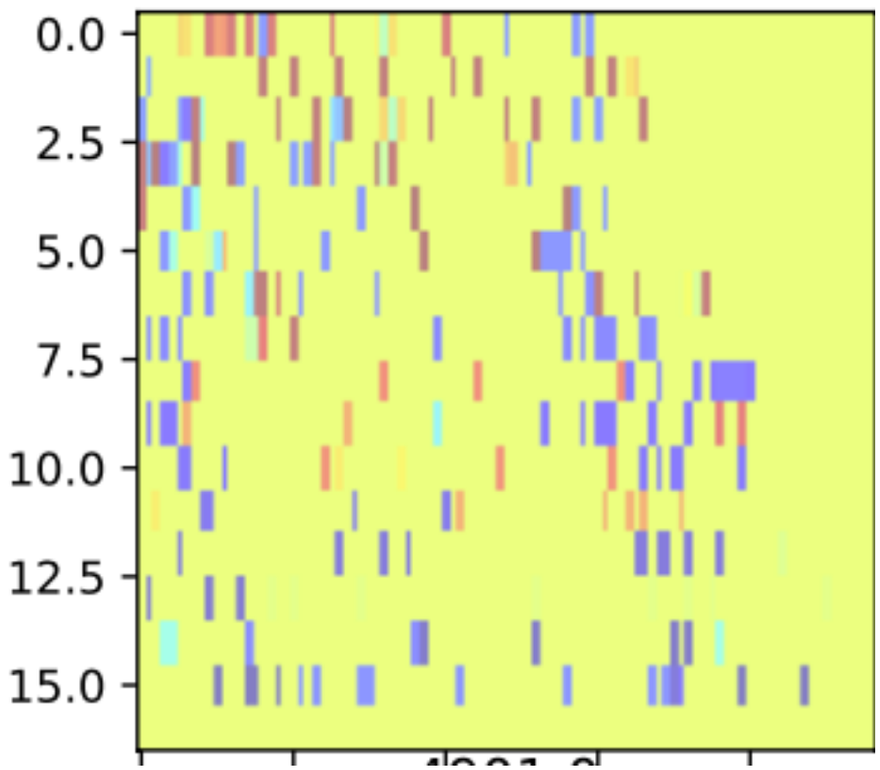


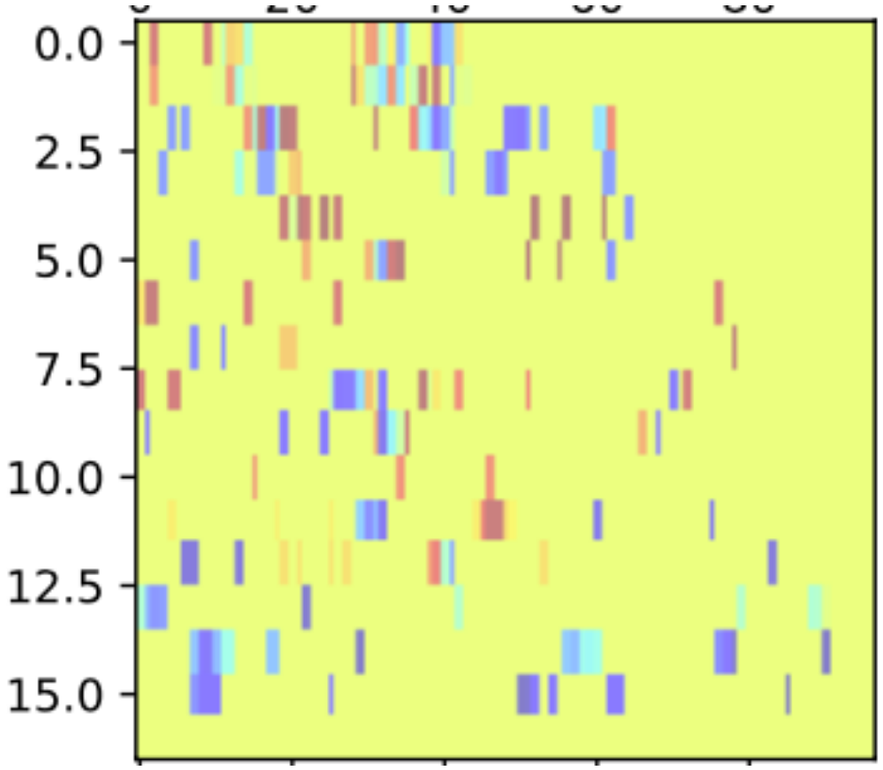


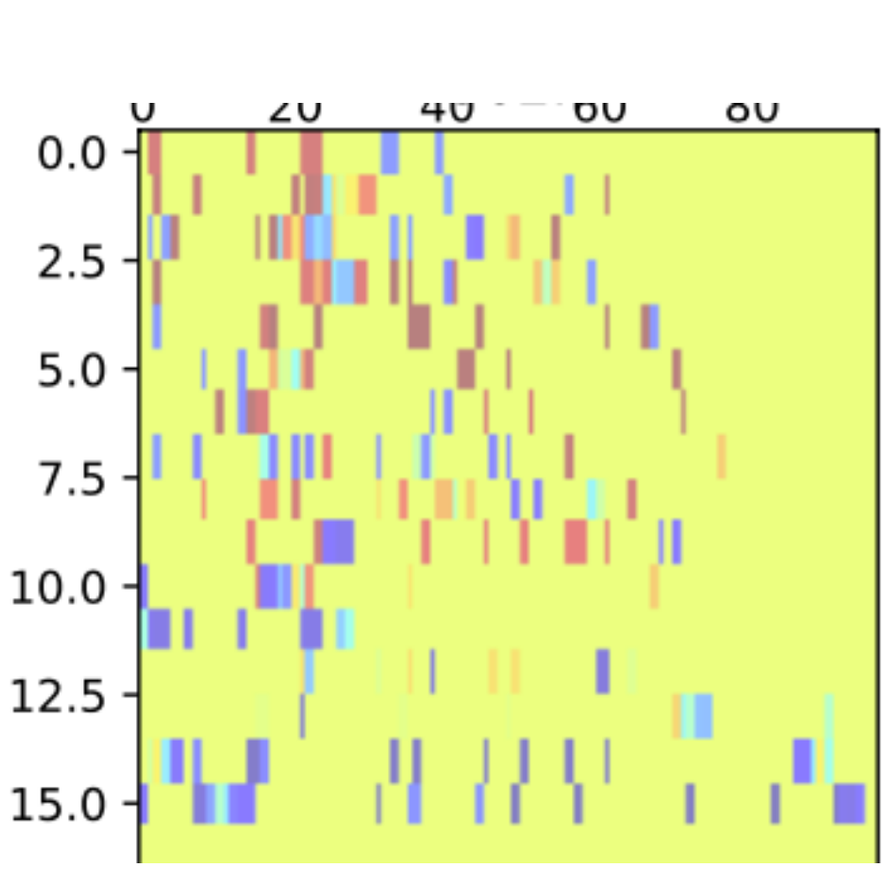


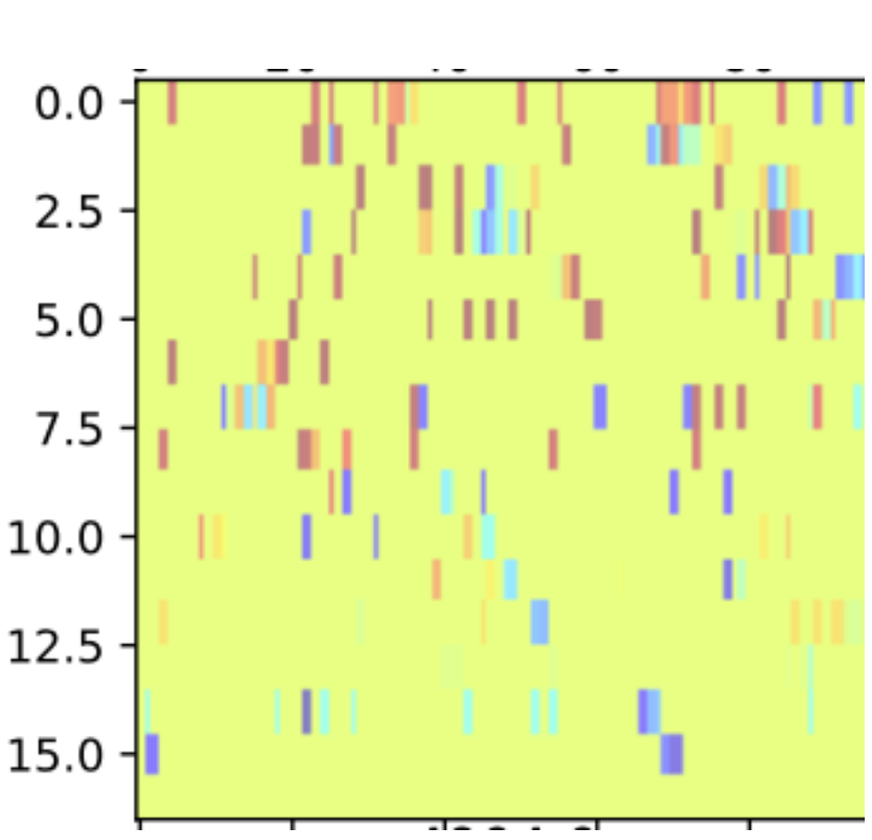




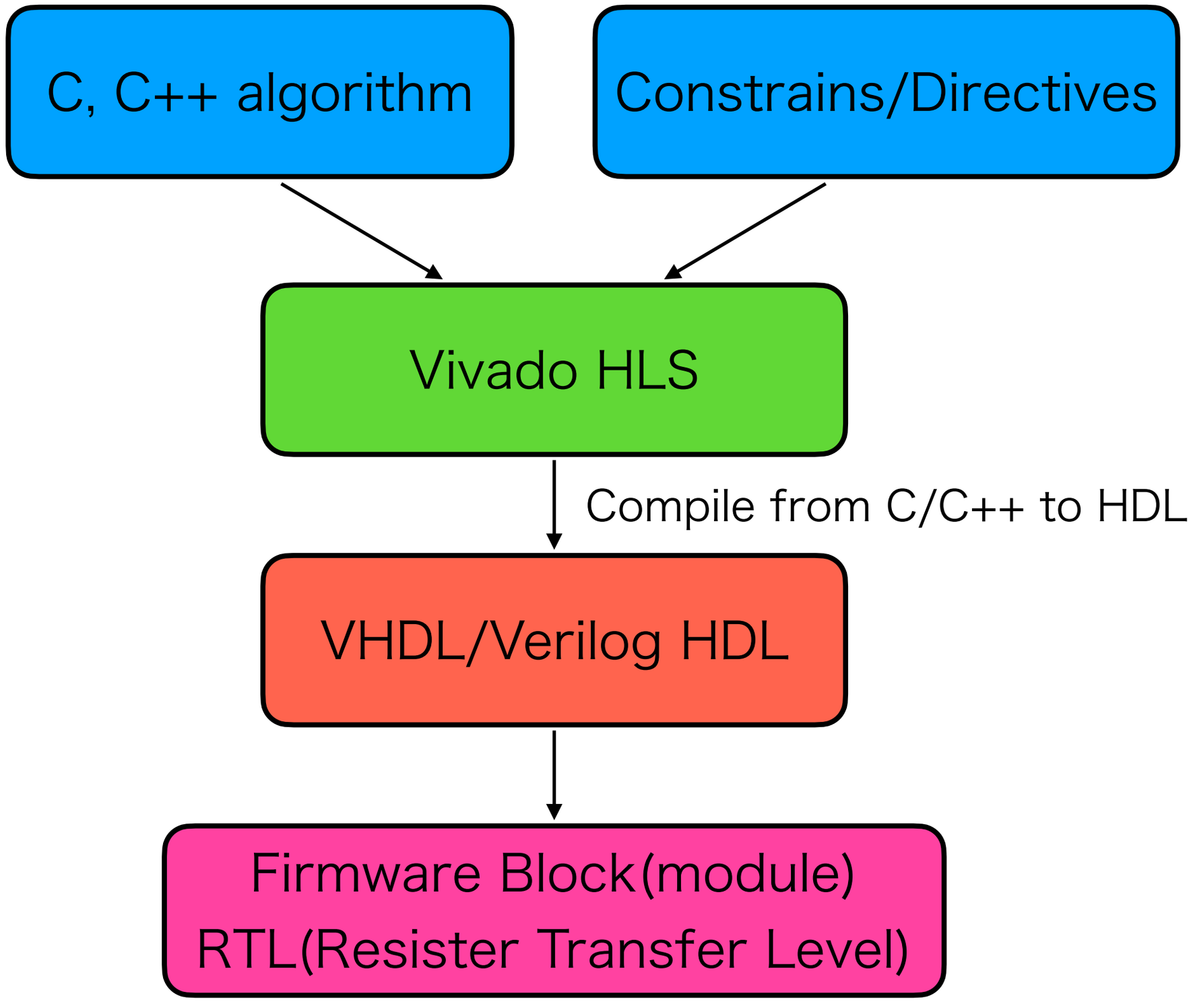

0.0
2.5
5.0
7.5
10.0
12.5
15.0


0.0
2.5
5.0
7.5
10.0
12.5
15.0


0
20
40
60
80
0.0
2.5
5.0
7.5
10.0
12.5
15.0


0.0
2.5
5.0
7.5
10.0
12.5
15.0

FPGA programing Flow



Key metrics for an FPGA implementation

1. Latency

- The total time required for a single iteration of the algorithm to complete

2. Initiation interval

- The number of clock cycles required before the algorithm may accept a new input

3. Resource usage

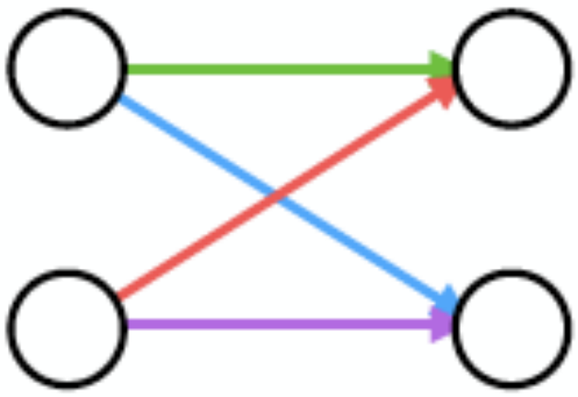
- BRAM:Block RAM
  - Hardened RAM resource
- DSPs : Digital Signal Processor
  - Performs multiplication and other arithmetic in the FPGA
- FF : Flip Flops
  - Register data in time with the clock pulse
- LUTs : Look Up Table(Logic)
  - Generic functions on small bit width inputs.

These limitations of the metrics become constraints.

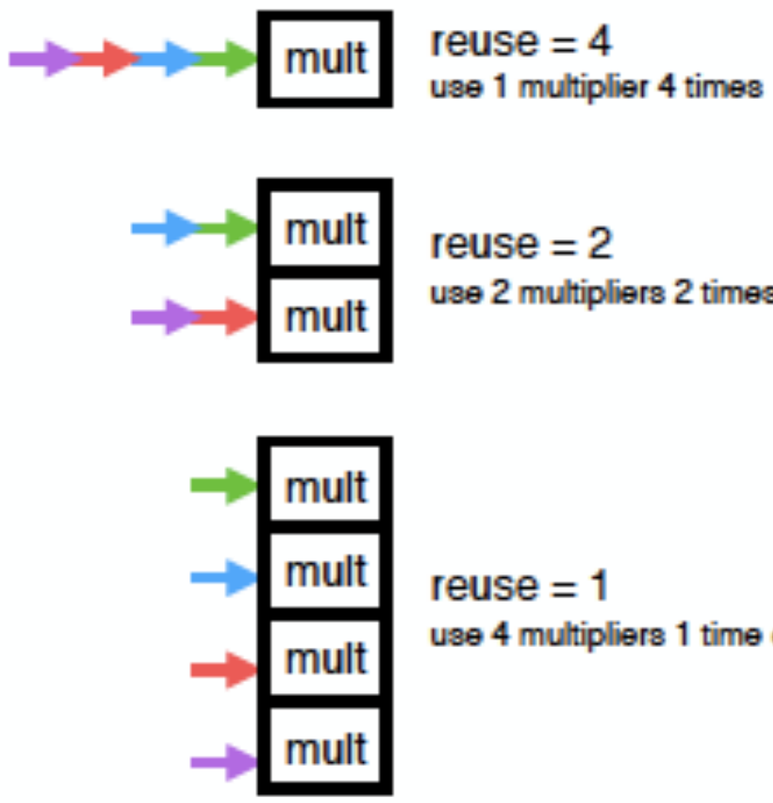
Parallelization

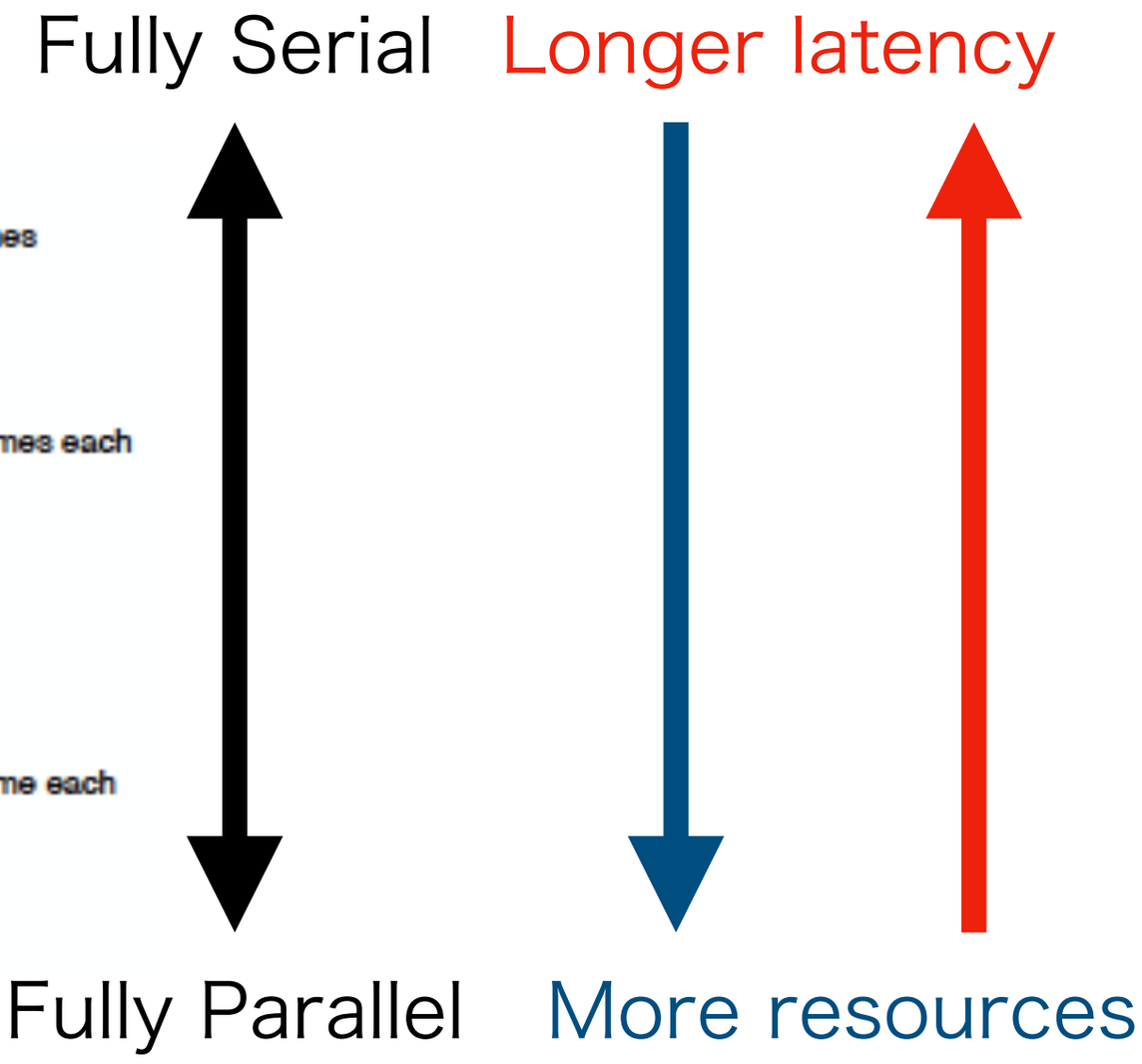
$$L_m = L_{mult} + (R - 1) \times \mathbf{II}_{mult} + L_{activ}$$

- L_m: The latency of layer m computation
- L_{mult}: The latency of multiplier
- R: Reuse factor
- $\mathbf{II}_{mult}$: Initiation interval of the multiplier
- L_{activ}: The latency of the activation Function Computaion



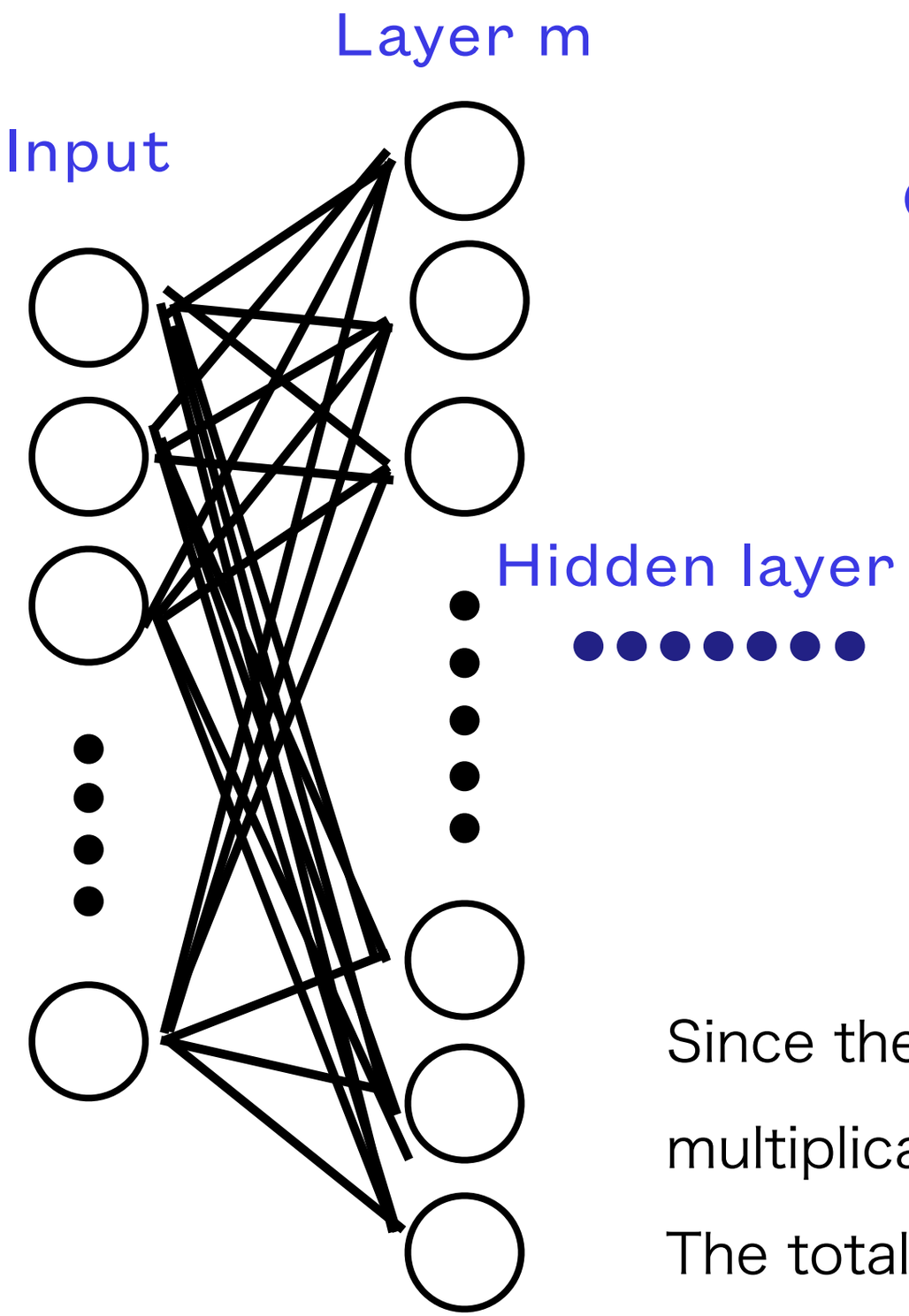
※





※Fast inference of deep neural networks in FPGAs for particle physics
arXiv:1804.06913v3 [physics.ins-det] 28 Jun 2018

Neural Network



$$\mathbf{x}_m = g_m\left(\mathbf{W}_{m,m-1}\mathbf{x}_{m-1} + \mathbf{b}_m\right)$$

$\mathbf{g}_m$: activation function for layer m

- Precomputed and stored in BRMs

$\mathbf{W}_{m.m-1}$: matrix of weights between layers m-1 and m

$\mathbf{b}_m$: bias

- Addition performed by Logic cell

Since the size of $\mathbf{W}_{m.m-1}$ is $N_m \times N_{m-1}$, the number of multiplication between layer m-1 and m is also $N_m \times N_{m-1}$.

The total number of multiplication is

$$N_{multiplication} = \sum_{m=1}^{M} N_{m-1} \times N_m \quad \propto \text{DSPs}$$