Nov 29, 2018 • Joint P2 Muon Upgrade + P2 L1 Muon Algorithms Workshop

Phase-2 EMTF algorithm: endcap standalone muon trigger

Darin Acosta [1], Andrew Brinkerhoff [1], Luca Cadamuro [1], Javier Duarte [2], Sergo Jindariani [2], Mugeon Kim [1], Jacobo Konigsberg [1], Jia Fu Low [1], Alex Madorsky [1], Vladimir Rekovic [1][3]

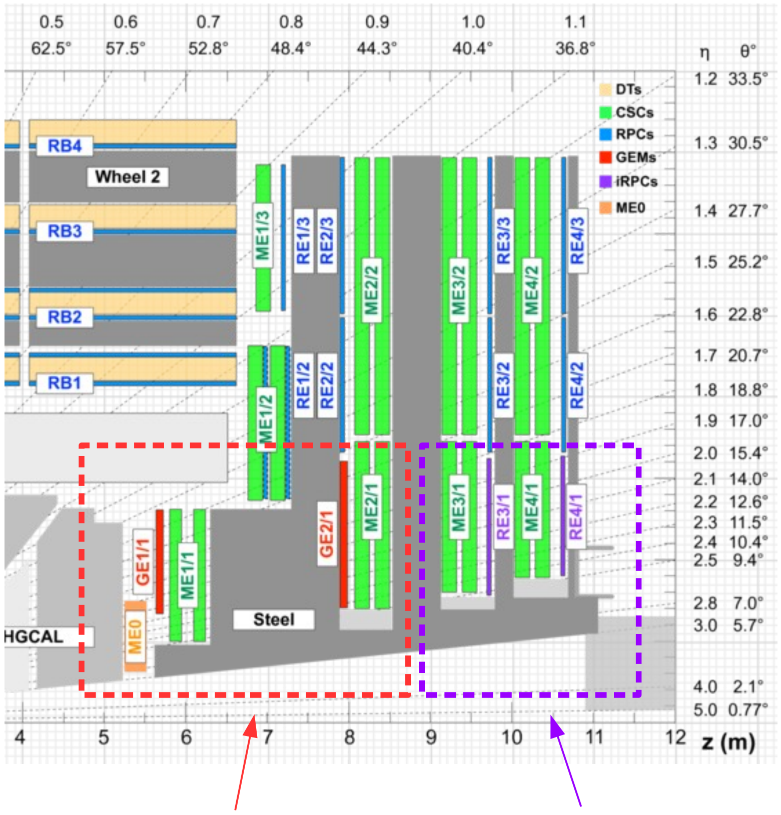
[1] University of Florida [2] Fermilab [3] Vinča Institute University of Belgrade

Overview

- EMTF++ algorithm basics
- Trigger primitives
- p_T assignment: Neural Network approach
- Performance plots
- NN firmware implementation using HLS4ML
- Run 3 NN prototype
- Summary

Phase 2 upgrade

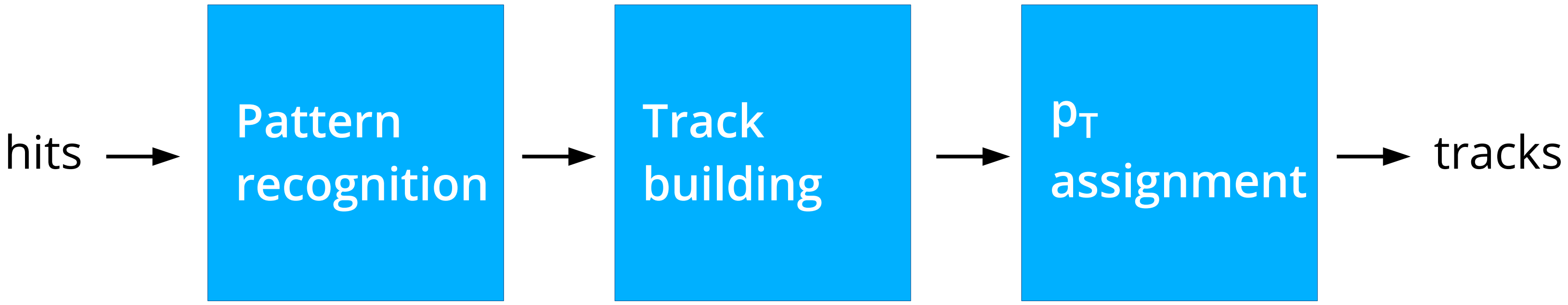
- EMTF performs standalone muon trigger in the endcap region ($1.24 < |\eta| < 2.4$)
- For Phase-2, EMTF has to evolve to
  - Incorporate the **new muon detectors**.
  - Improve **efficiency, redundancy, p_T resolution, timing**.
  - Maintain the same trigger threshold at <PU>=200 at a reasonable **rate**.
- Challenges❗
  - Highly **non-uniform magnetic field** with very little magnetic bending in the very forward region.
  - **Large background** from low p_T muons, punch-throughs, PU, etc that could lead to non-linear PU dependence.
  - Various detector types to accommodate.
- EMTF also has to interface with **the correlator** for matching muons with tracks from the track trigger
  - Refer to L. Cadamuro's and V. Rekovic's talks.
- In addition, EMTF has to **add sensitivity to new physics** scenario, e.g. displaced muons and HSCPs.
  - Significant work has been done by TAMU and the RPC group. From the EMTF side, we don't have any results yet.



GEM
- 🟥 GE1/1
- 🟥 GE2/1
- 🟧 ME0

iRPC
- 🟪 RE3/1
- 🟪 RE4/1

EMTF++ algorithm basics

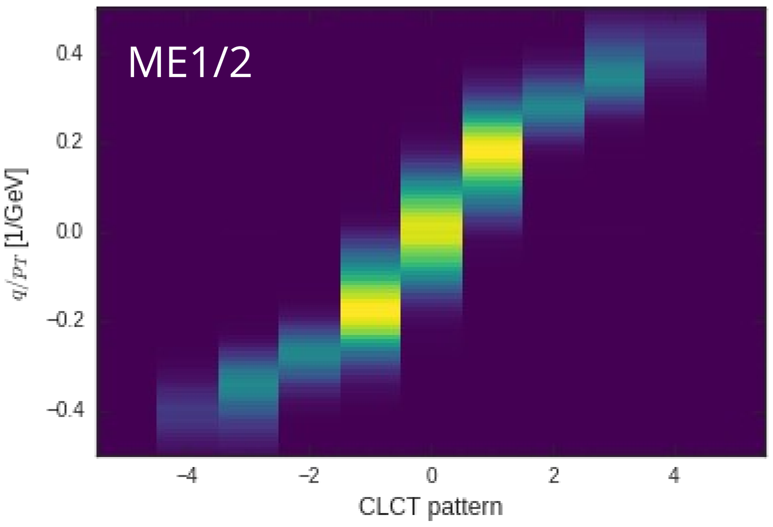


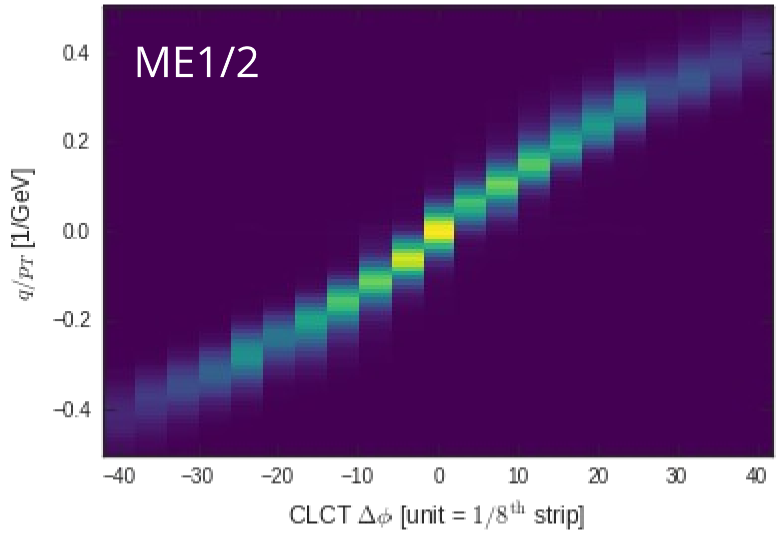
- EMTF++ is an extension of the current EMTF algorithm, and it follows the same design.
- **Pattern recognition**
  - Use patterns to find hits in the 4 muon stations that are consistent with muons of different p_T.
- **Track building**
  - Build tracks by picking a unique hit from each muon station.
- **p_T assignment**
  - Use machine learning (e.g. BDT, NN) to determine the track p_T using all the discriminating variables: $\Delta\phi$, $\Delta\theta$, bend, η, etc.
  - In the large LUT approach, the input variables are encoded using 30-bit address space (Phase 1). For Phase 2, the LUT will have 37-bit address space.
  - For Phase 2, also want to investigate running NN directly in the FPGA.

Trigger primitives

- EMTF++ has to handle various types of detectors
  - CSC, RPC, GEM, iRPC, ME0, (perhaps DT?)
  - At the moment, consider CSC and GEM as separate trigger primitives. We will also study the performance with GEM+CSC super-LCT in the future.
- 🟩 CSC
  - We already use all the CSC variables in the EMTF: ϕ, θ, CLCT pattern, F/R
  - Based on the studies shown by TAMU and UCLA (A. Peck's talk, W. Nash's talk), doing a least squares fit to the CLCT comparator digis can improve the hit ϕ position resolution.
  - We did our own study and found that, besides the position improvement, the bend from the fit is also very useful
    - Bend = Δϕ between the hit positions in the innermost and the outermost layers (a la Δϕ in ME0)
    - Comparator digis are in half-strip unit. We use bend in 1/8-strip unit by multiplying Δϕ by 4 and then converting it to an integer

Trigger primitives



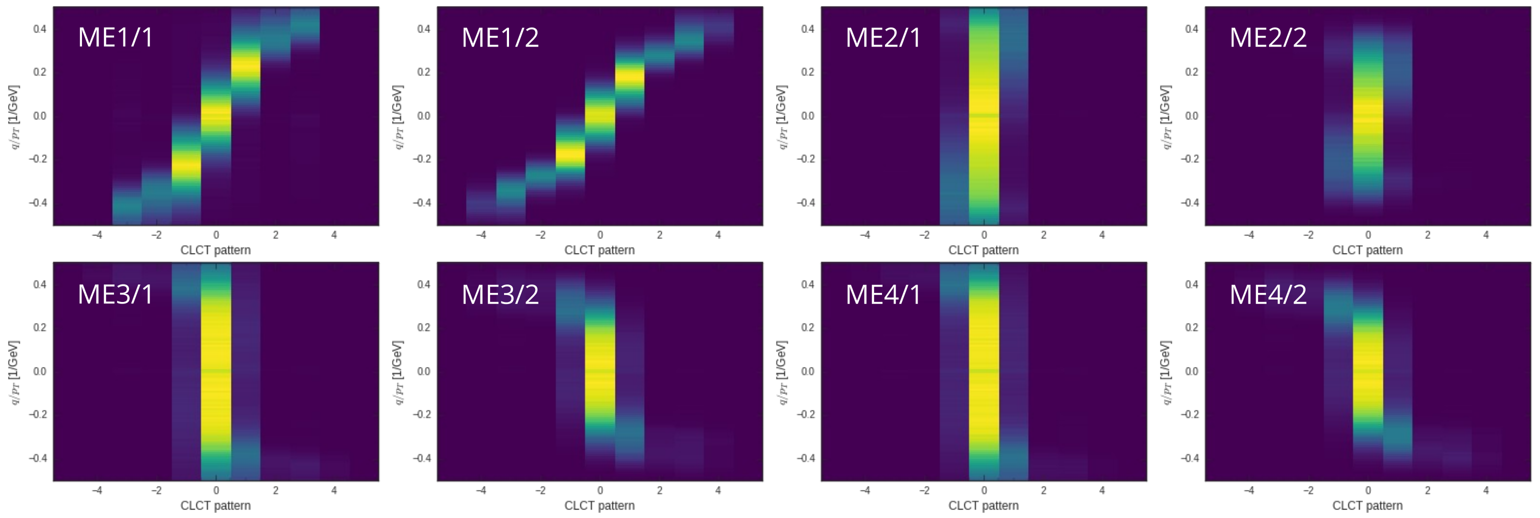


Current: we take the CLCT pattern which is a number [0-10], convert it to [-5,5] using a small LUT. Then we use it as input to the BDT. However, besides ME1/1 and ME1/2, most of the times we only get pattern numbers [8-10].

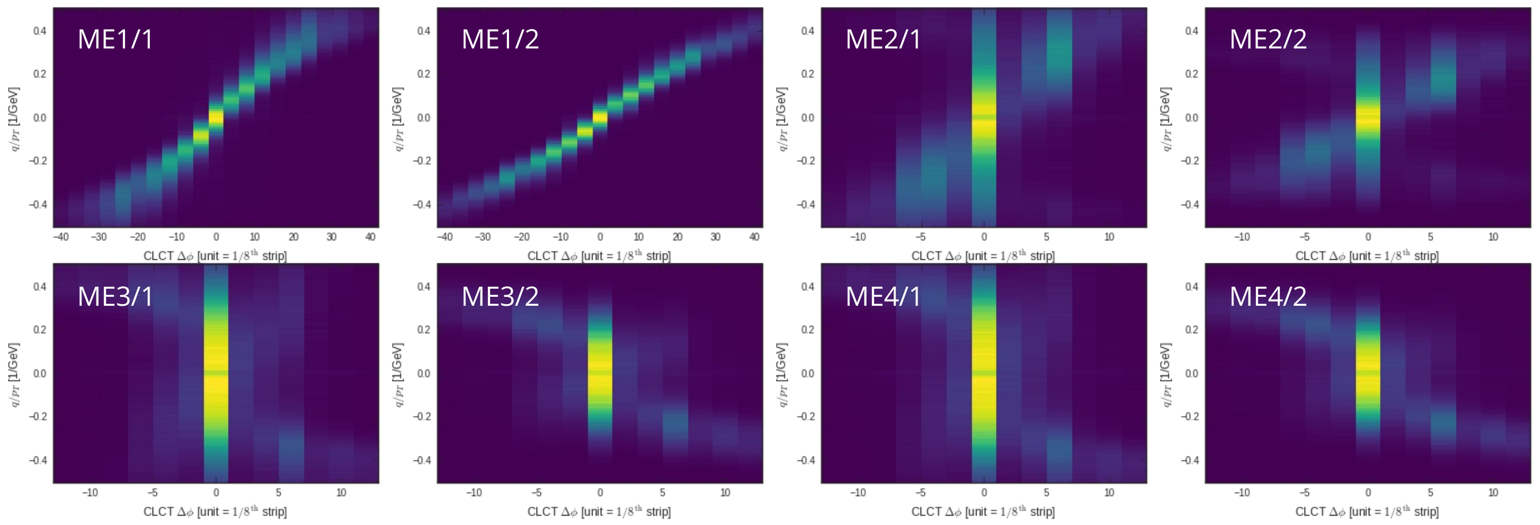
New: Δϕ from the least squares fit. It turned out that CSC **has always had** the ability to measure both the hit position and the local momentum of the muon as it crosses the detector. However, only works for ME1/1 and ME1/2.

Trigger primitives

Current CSC bend



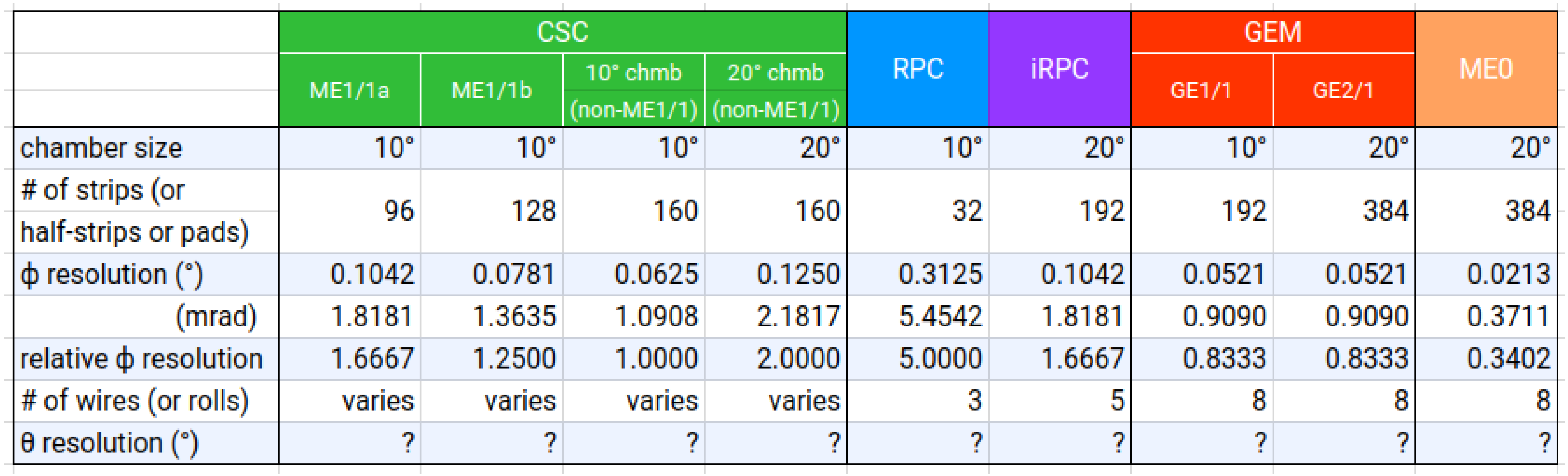
New CSC bend



Trigger primitives

- 🟦 RPC/🟪 iRPC
  - RPC has been included in EMTF since 2017.
  - For Phase 2, the new Link Board will provide ~1.5ns timing resolution. Not used in the study yet, but we will include it in the future.
  - iRPC is treated like RPC, but with 3x better ϕ resolution.
  - "2D readout" feature of iRPC can provide the position of the hit along the strip with a resolution of ~2 cm. Not used in the study yet, but we will include it in the future.
- 🟥 GEM
  - At the moment, GEM hits are included as separate trigger primitives.
  - GE1/1 chamber is 10° wide with 192 pads. GE2/1 is 20° with 384 pads. A pad = 2 strips ganged together.
  - GEM is a 2-layer detector. A coincidence is required in the two layers (with a window of ±3 pads for GE1/1, ±2 pads for GE2/1).
- 🟧 ME0
  - The current ME0 trigger simulation seems to be based on offline reconstruction. Not sure if the performance might be too good/unrealistic.
    - Using the digi collection "me0TriggerPseudoDigis" in CMSSW.
  - ME0 is a 6-layer detector. ME0 segments will be built by dedicated electronics. We expect to receive the segment ϕ, θ and bend (a.k.a. Δϕ).
    - Also assuming the ϕ resolution will be better than half-strip. Possibly close to 1/8-strip.
    - Not clear about the data format and the exact number of bits for the variables yet.

Trigger primitives



	CSC				RPC	iRPC	GEM		ME0
	ME1/1a	ME1/1b	10° chmb (non-ME1/1)	20° chmb (non-ME1/1)			GE1/1	GE2/1	
chamber size	10°	10°	10°	20°	10°	20°	10°	20°	20°
# of strips (or half-strips or pads)	96	128	160	160	32	192	192	384	384
ϕ resolution (°)	0.1042	0.0781	0.0625	0.1250	0.3125	0.1042	0.0521	0.0521	0.0213
(mrad)	1.8181	1.3635	1.0908	2.1817	5.4542	1.8181	0.9090	0.9090	0.3711
relative ϕ resolution	1.6667	1.2500	1.0000	2.0000	5.0000	1.6667	0.8333	0.8333	0.3402
# of wires (or rolls)	varies	varies	varies	varies	3	5	8	8	8
θ resolution (°)	?	?	?	?	?	?	?	?	?

- The relative ϕ resolution is relative to CSC 10° chambers (non-ME1/1). For ME0, the ϕ resolution in the table is 20°/384 x 1/sqrt(6).
- Internally, the ϕ coordinates of all the hits are converted into a common scale, where 1 unit = 1/60°. The θ coordinates are converted into a common scale where 1 unit = 0.285°.

Pattern recognition

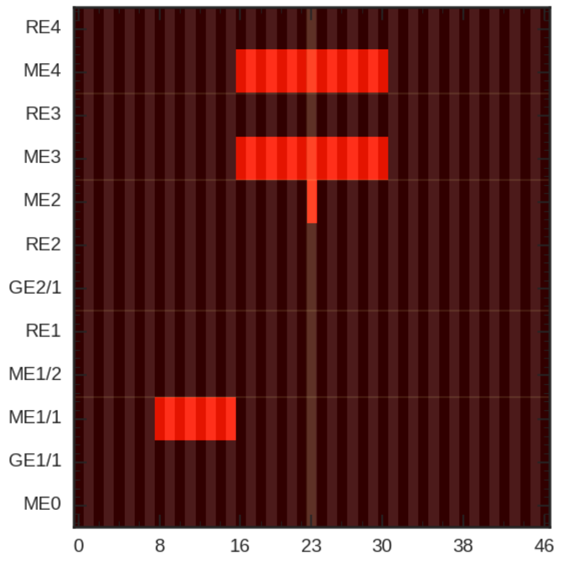
- The patterns have been reworked to accommodate the new detectors
  - Current EMTF patterns only have 4 windows for 4 muon stations.
  - Expanded to have windows for the following 12 "virtual" stations:

ME1/1	ME1/2	ME2	ME3	ME4	RE1	RE2	RE3	RE4	GE1/1	GE2/1	ME0

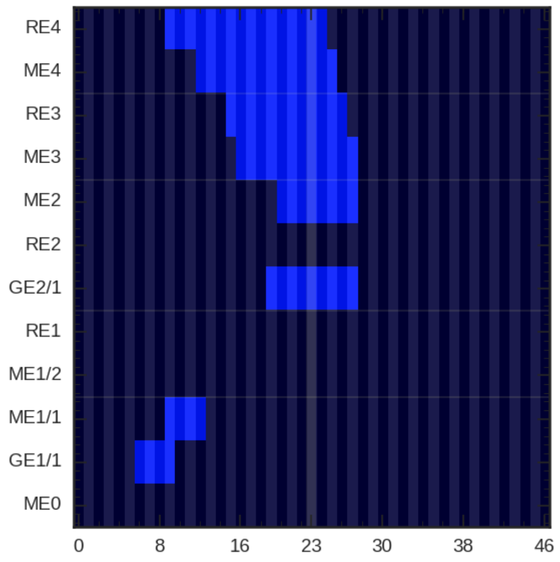
(ME1/1 and ME1/2 are split because they are actually very different due to different magnetic bending)

  - Patterns have different straightness (from 0 to 4) for muons of different p_T (low to high). The windows are tuned for each virtual station using a muon gun.
  - Also increased the number of zones (η partitions) from 4 to 6.

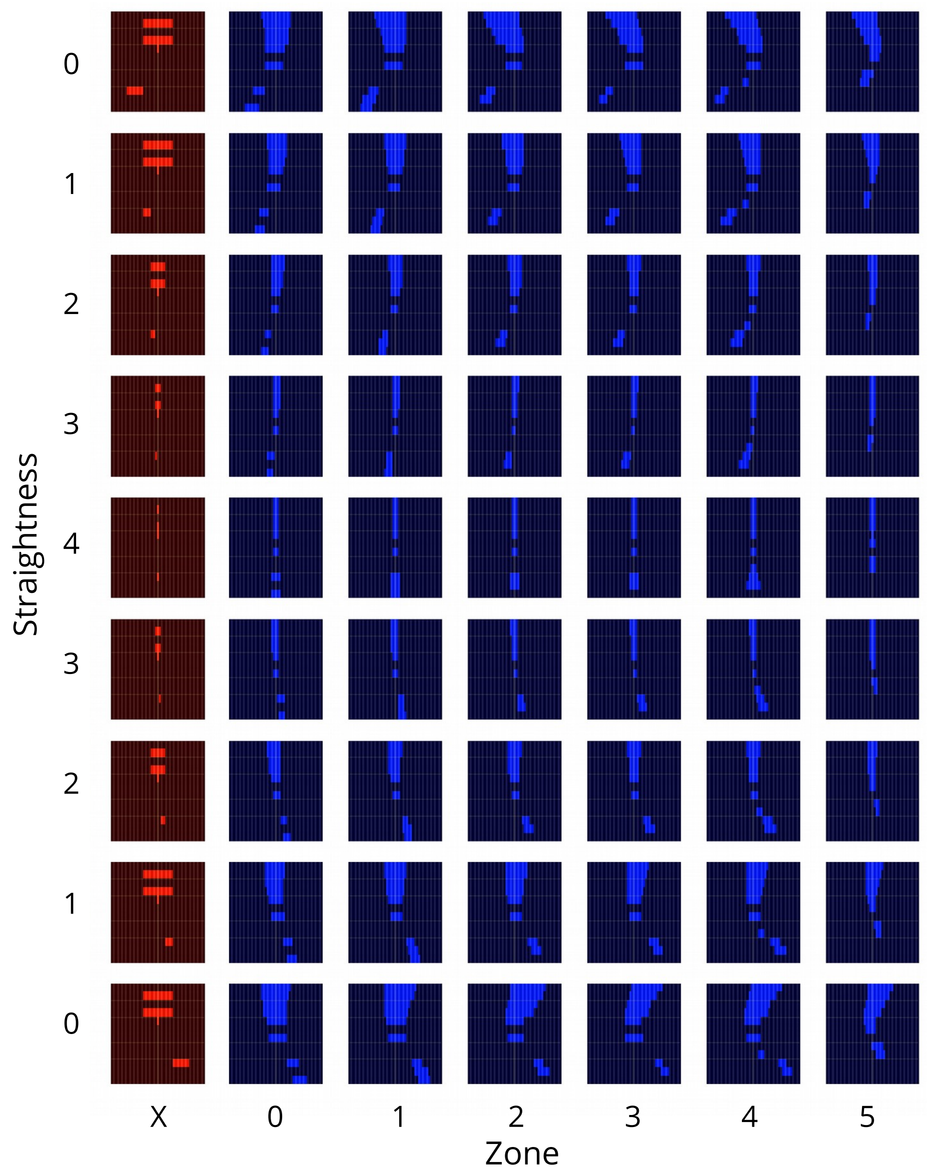
EMTF pattern
(straightness=0, charge=-1)



EMTF++ pattern (zone 1)
(straightness=0, charge=-1)



Pattern recognition

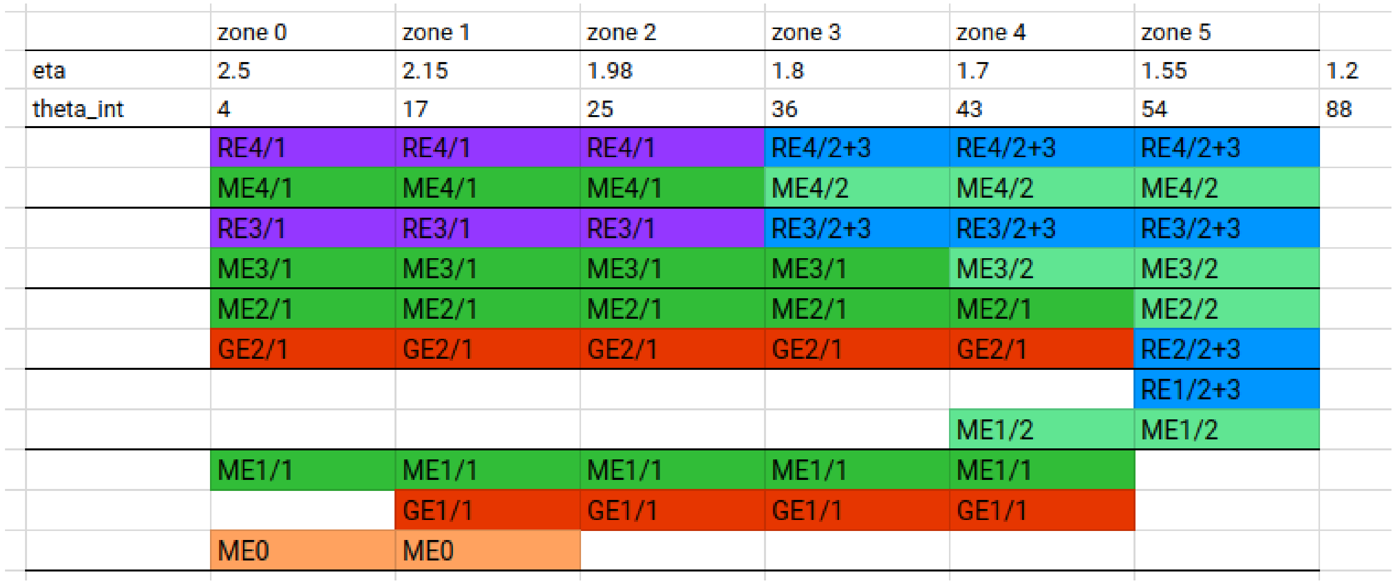


Larger version:

https://jiafulow.web.cern.ch/jiafulow/L1MuonTrigger2018/emtf_patterns.png
https://jiafulow.web.cern.ch/jiafulow/L1MuonTrigger2018/emtfpp_patterns.png

Pattern recognition

- The number of zones has increased from 4 to 6. The boundaries are placed close to where the rings terminate.
- As the result of this gerrymandering, there is less mixing of detector types or rings in the same zone.



	zone 0	zone 1	zone 2	zone 3	zone 4	zone 5	
eta	2.5	2.15	1.98	1.8	1.7	1.55	1.2
theta_int	4	17	25	36	43	54	88
	RE4/1	RE4/1	RE4/1	RE4/2+3	RE4/2+3	RE4/2+3	
	ME4/1	ME4/1	ME4/1	ME4/2	ME4/2	ME4/2	
	RE3/1	RE3/1	RE3/1	RE3/2+3	RE3/2+3	RE3/2+3	
	ME3/1	ME3/1	ME3/1	ME3/1	ME3/2	ME3/2	
	ME2/1	ME2/1	ME2/1	ME2/1	ME2/1	ME2/2	
	GE2/1	GE2/1	GE2/1	GE2/1	GE2/1	RE2/2+3	
						RE1/2+3	
					ME1/2	ME1/2	
	ME1/1	ME1/1	ME1/1	ME1/1	ME1/1		
		GE1/1	GE1/1	GE1/1	GE1/1		
	ME0	ME0					

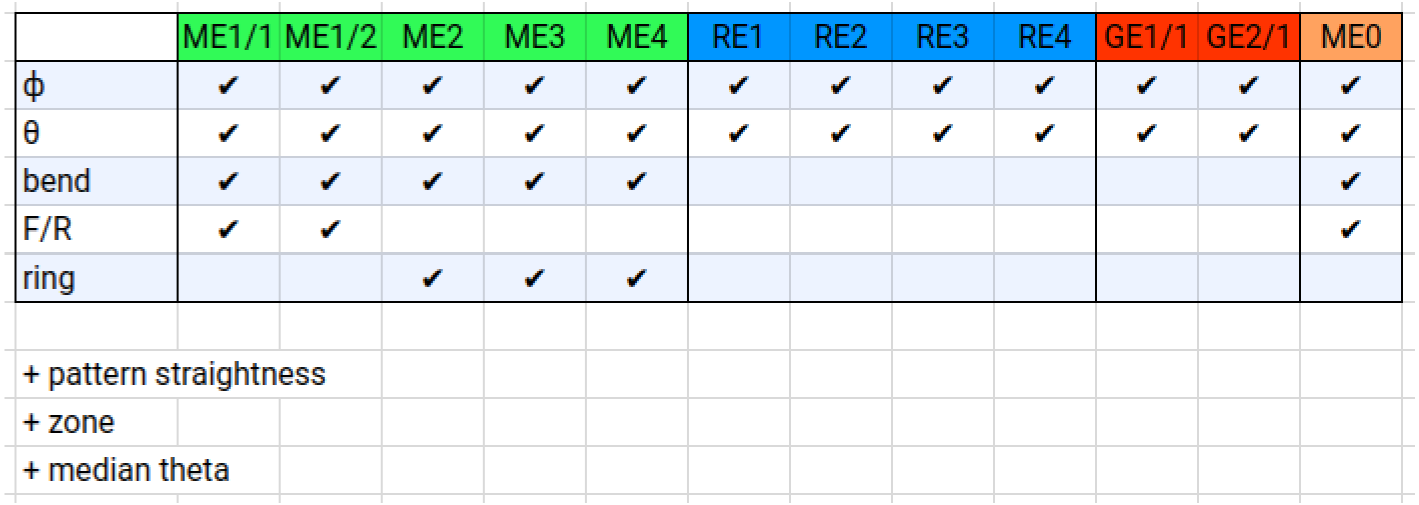
  - In any given zone, there are 8 or 9 “active” virtual stations.
  - Actually, RE4/1 and RE4/2 can coexist in zone 2, but we decided to use exclusively RE4/1 (which is iRPC) in that zone, ignoring RE4/2 (which is RPC)

Track building

- If there are multiple hits in a station, the hit with the minimum (Δθ, Δϕ) is picked
  - Δθ = difference between hit θ and the track θ
    - where the track θ is the median value of all the hit θ's in the track.
  - Δϕ = difference between hit ϕ and the track ϕ + bias correction
    - where the track ϕ is the keystrip position of the pattern. A bias term is added based on the pattern straightness and zone.
  - Minimum (Δθ, Δϕ) means that Δθ is used to sort first, but if more than one hits have the same Δθ, then Δϕ is used as the tie-breaker for sorting.

p_T assignment

- For Phase 2, we want to explore p_T assignment using neural network
  - Alleviates bottleneck of # of address bits to LUT by using logic and DSPs in FPGA
  - Can use many variables without heavy compression
  - NN inference can be implemented in FPGA with low latency
    - See HLS4ML paper arXiv:1804.06913
- At the moment, consider 39 features
  - Can easily add more, provided enough FPGA resources (more on this later)

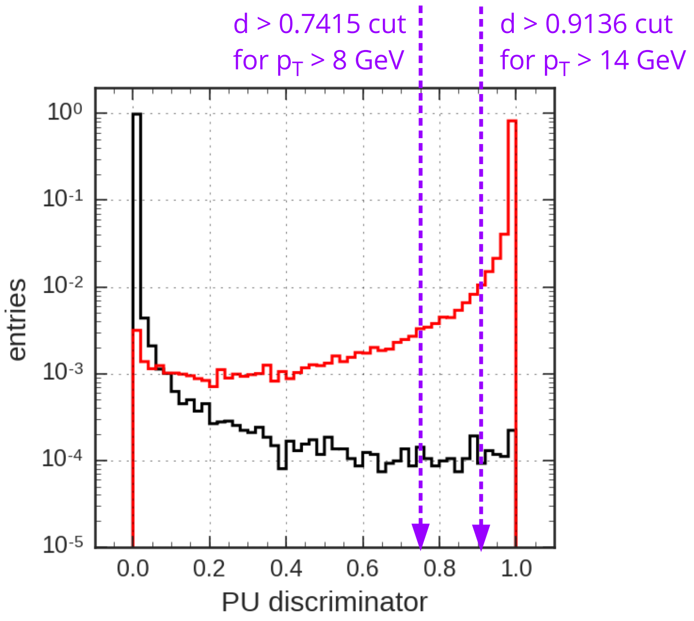


	ME1/1	ME1/2	ME2	ME3	ME4	RE1	RE2	RE3	RE4	GE1/1	GE2/1	ME0
ϕ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
θ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bend	✓	✓	✓	✓	✓							✓
F/R	✓	✓										✓
ring			✓	✓	✓							

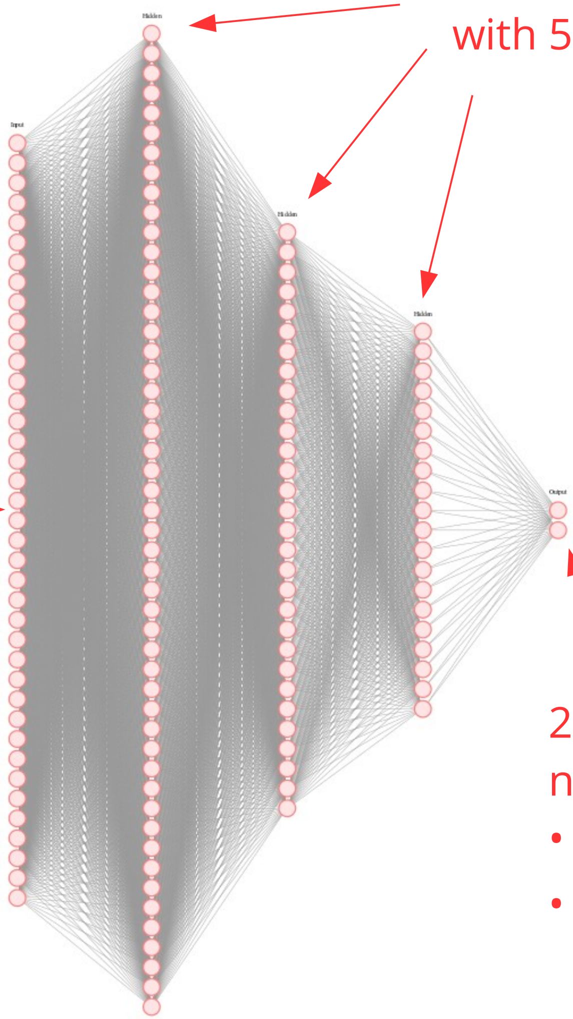
+ pattern straightness
+ zone
+ median theta

Neural network

- Using **Keras**, create a feedforward NN with **3 hidden layers** (50, 30, 20 nodes respectively). Batch normalization is applied at all the hidden layers.
- Two output nodes: **q/p_T** and **PU discriminator**, the latter is used to reject PU tracks and bad quality tracks.
  - q/p_T is a regression trained with muons to assign momentum
  - PU discriminator is a classification trained with:
    - S = muon $p_T > 8$ GeV
    - B = any track from 200 PU. Veto applied to any events with $p_T > 8$ GeV.
- PU discriminator is used as a cut:
  - If q/p_T output gives $p_T > 8$ GeV, a cut is applied to PU discriminator to accept or reject the muon. A tighter cut is applied to for $p_T > 14$ GeV muons.



Neural network

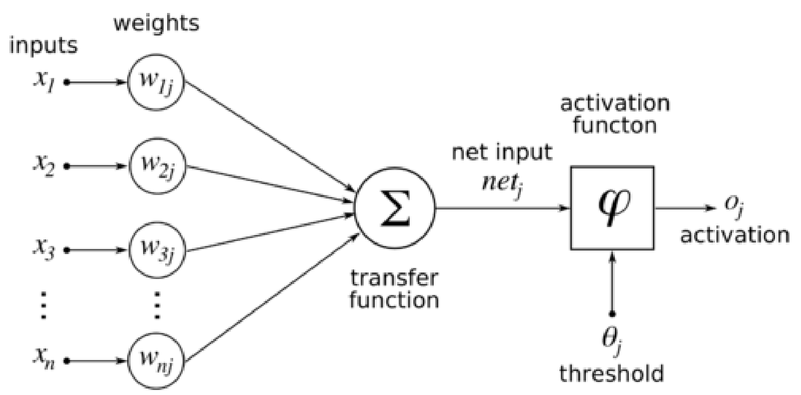


3 hidden layers
with 50/30/20 nodes

39 input
nodes

2 output
nodes:
- q/p_T
- PU discr

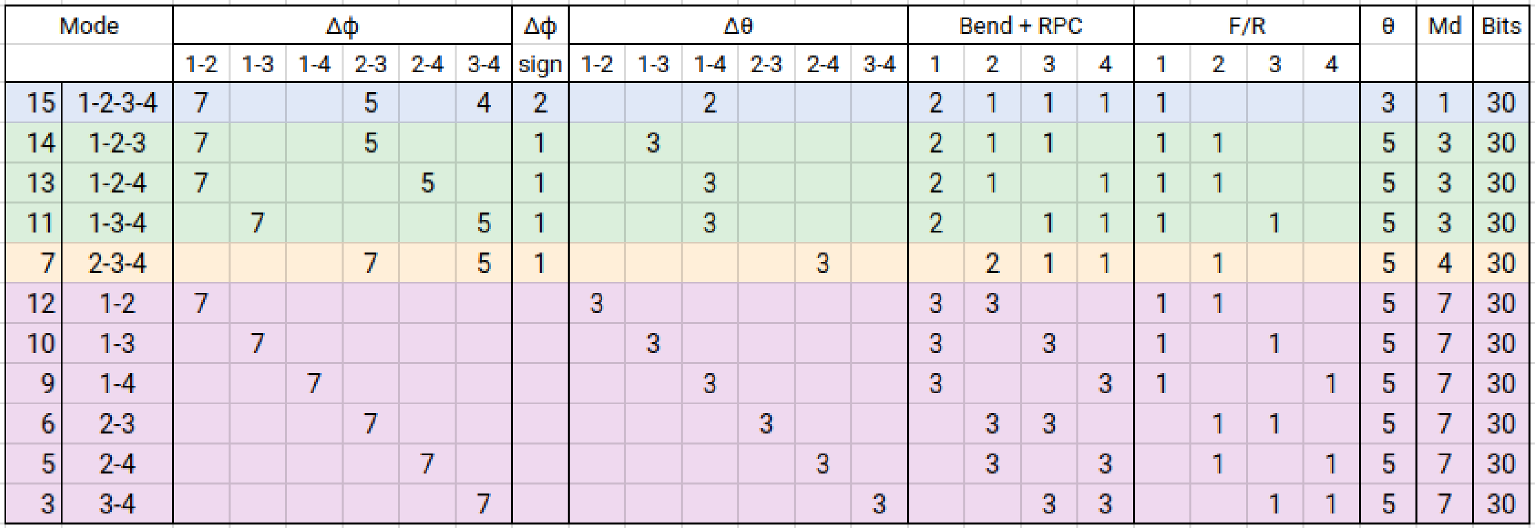
Each node computes $o_j = \varphi\left(\sum_{i=1}^{n} w_{ij} \cdot x_i + \theta_j\right)$



Pruning can be applied to remove unimportant synapses to reduce resource usage.

PTLUT approach

- We might also continue to use the large LUT approach as in the current EMTF.
  - BDT is trained offline and evaluated offline for all the possible combinations of the input variables. The BDT outputs are stored into the PTLUT. They are extracted from the PTLUT during running.
- The target Phase-2 board will have a larger address space, 30 bits → 37 bits (128 GB) for additional info such as GEM-CSC bend angles and RPC information to be encoded.
  - Using about 25 variables. FPGA does some preprocessing to compress data to address field.
  - To save address space, Δϕ's are converted into non-linear scale; different number of bits are used as the identifier of the track mode (1 bit for 4-station tracks, 3-4 bits for 3-station tracks, etc)

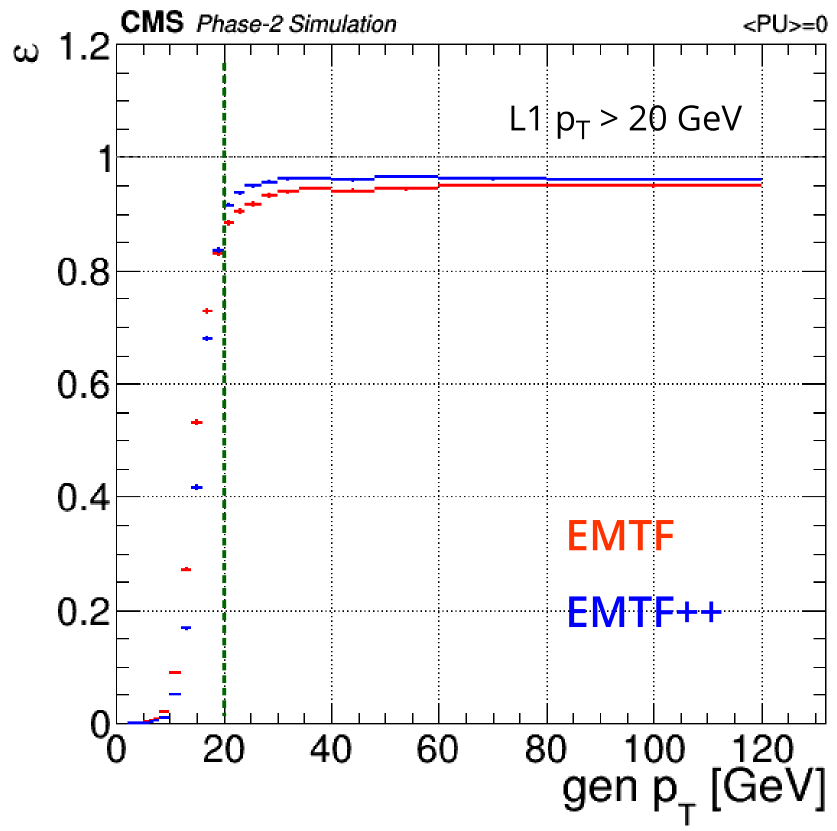


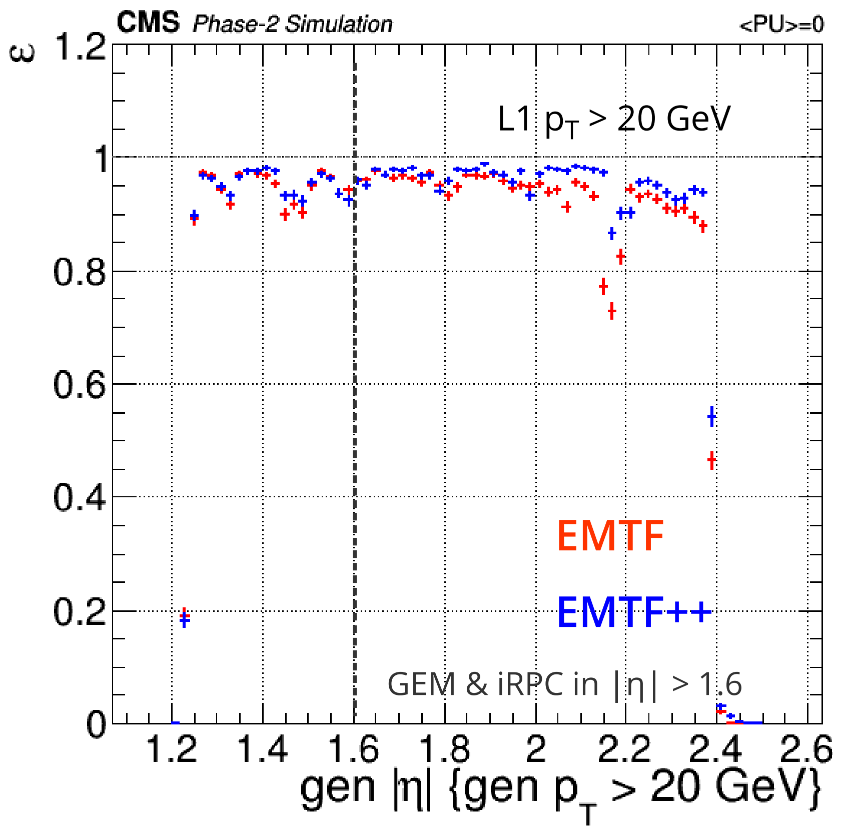
Mode		Δϕ						Δϕ	Δθ						Bend + RPC				F/R				θ	Md	Bits
		1-2	1-3	1-4	2-3	2-4	3-4	sign	1-2	1-3	1-4	2-3	2-4	3-4	1	2	3	4	1	2	3	4			
15	1-2-3-4	7			5		4	2			2				2	1	1	1	1				3	1	30
14	1-2-3	7			5			1		3					2	1	1		1	1			5	3	30
13	1-2-4	7				5		1			3				2	1		1	1	1			5	3	30
11	1-3-4		7				5	1			3				2		1	1	1		1		5	3	30
7	2-3-4				7		5	1					3			2	1	1		1			5	4	30
12	1-2	7							3						3	3			1	1			5	7	30
10	1-3		7							3					3		3		1		1		5	7	30
9	1-4			7							3				3			3	1			1	5	7	30
6	2-3				7							3				3	3			1	1		5	7	30
5	2-4					7							3			3		3		1		1	5	7	30
3	3-4						7							3			3	3			1	1	5	7	30

- Might think of ways to combine both the approaches
  - At the moment, problem is finding manpower and time to also train BDT.

Performance plots

Results: efficiency



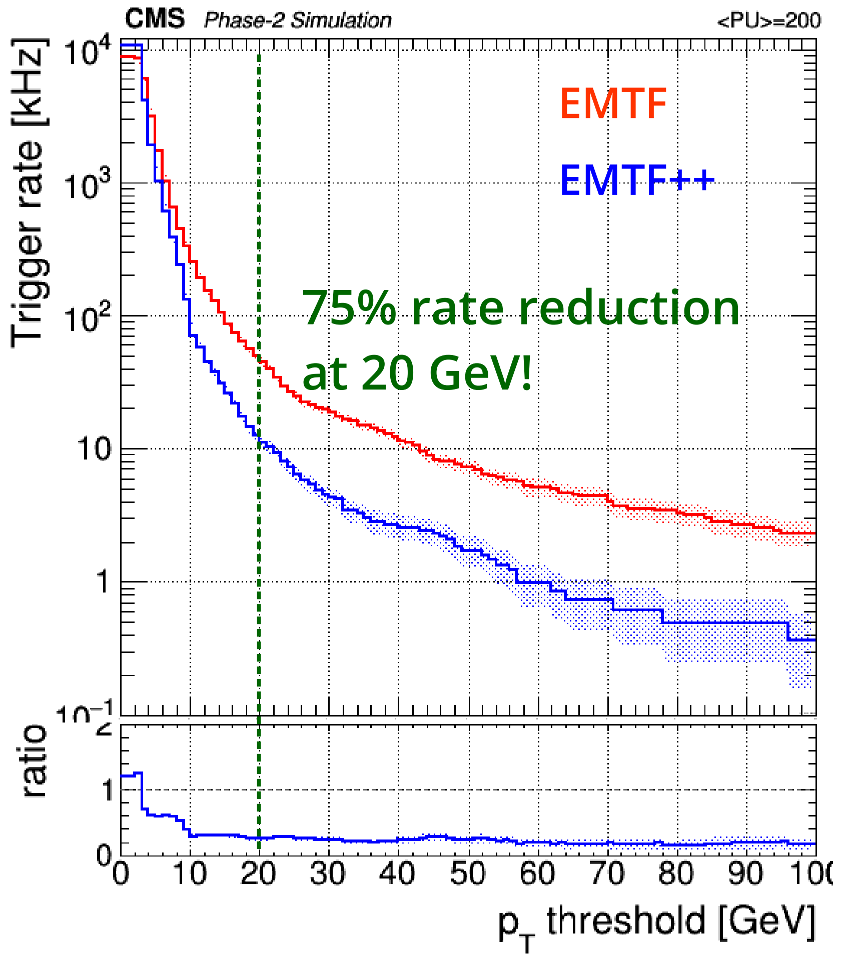


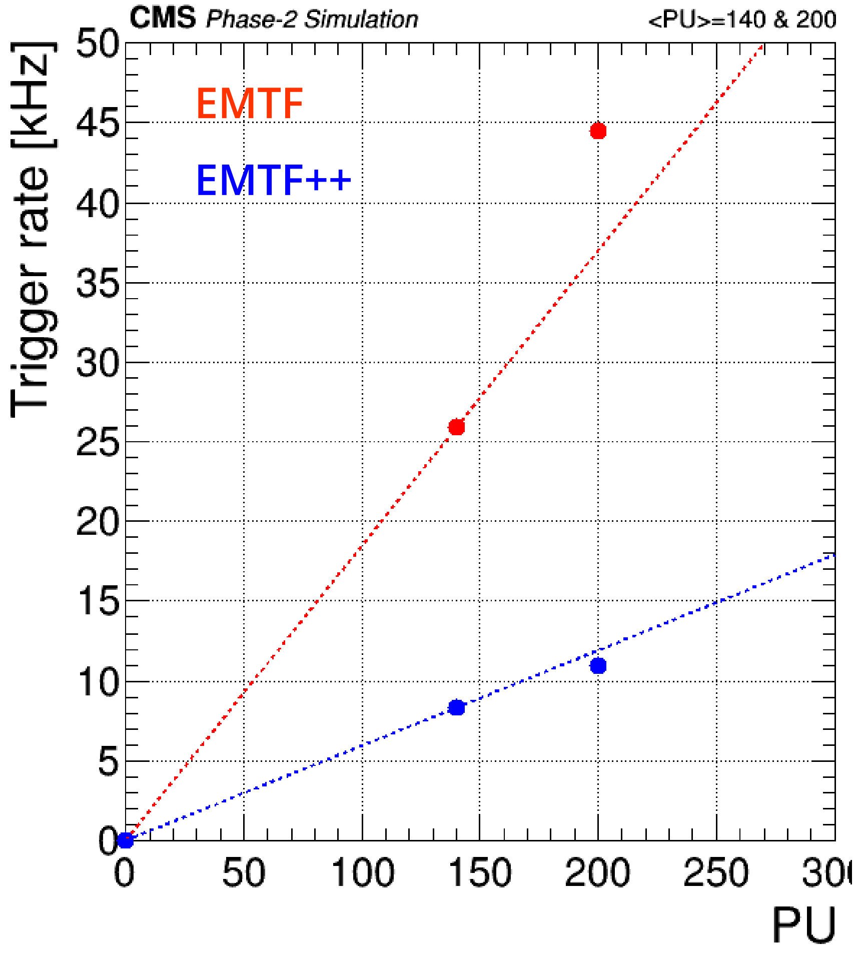
Sharper turn on + higher plateau efficiency

Efficiency more flat in η

Inefficiency at η~2.15 due to low efficiency at ME3, ME4 and RE3 at the same time, and it's also the transition from ME1/1a to ME1/1b.

Results: rates @ 200 PU

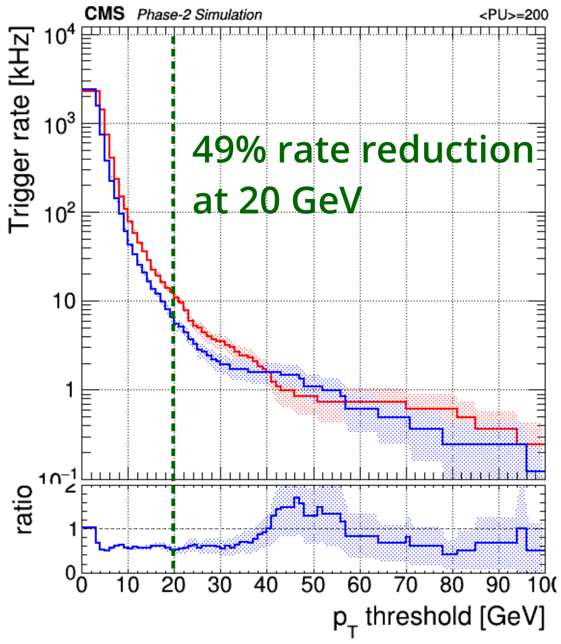




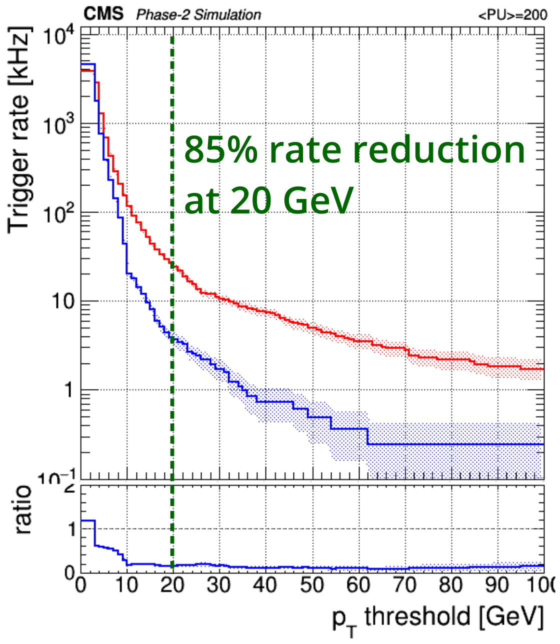
Rate @ 20 GeV threshold is 11 kHz. Linear PU dependence comparing 140 vs 200 PU. O(10%) stat uncertainties in these rates.

Results: rates @ 200 PU

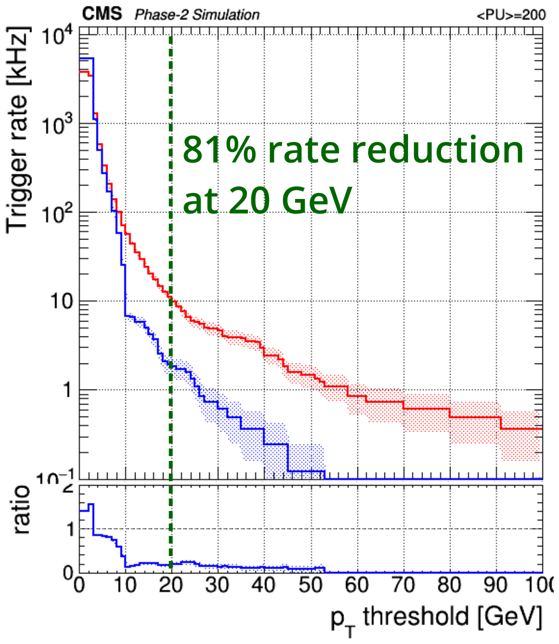
$1.24 < |\eta| < 1.65$
(no new Phase-2 detectors)



$1.65 < |\eta| < 2.15$
(GE1/1, GE2/1, ME0, iRPC)



$2.15 < |\eta| < 2.4$
(GE2/1, ME0, iRPC)



Significant rate reduction in regions with GEM and ME0 detectors.

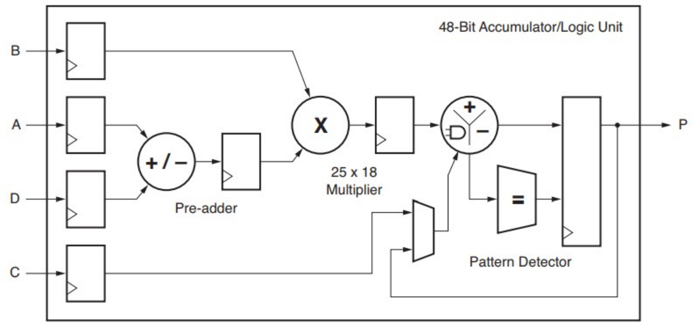
Results from Sergo

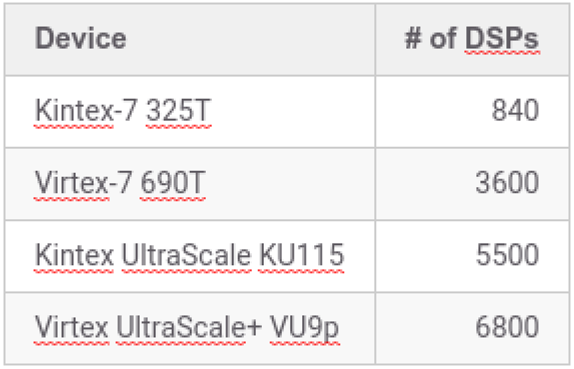
NN firmware impl. using HLS4ML

NN firmware implementation

- hls4ml toolkit
  - A toolkit to implement fast neural network inferences in FPGAs using Vivado High Level Synthesis (HLS).
  - Can convert NN models from popular ML libraries (Keras, Tensorflow, etc) into VHDL codes, which can be used to generate the firmware.
    - Optimize use of DSPs in the modern FPGAs for multiply-accumulate operations
  - See paper arxiv:1804.06913

Basic DSP48E1 Slice functionality

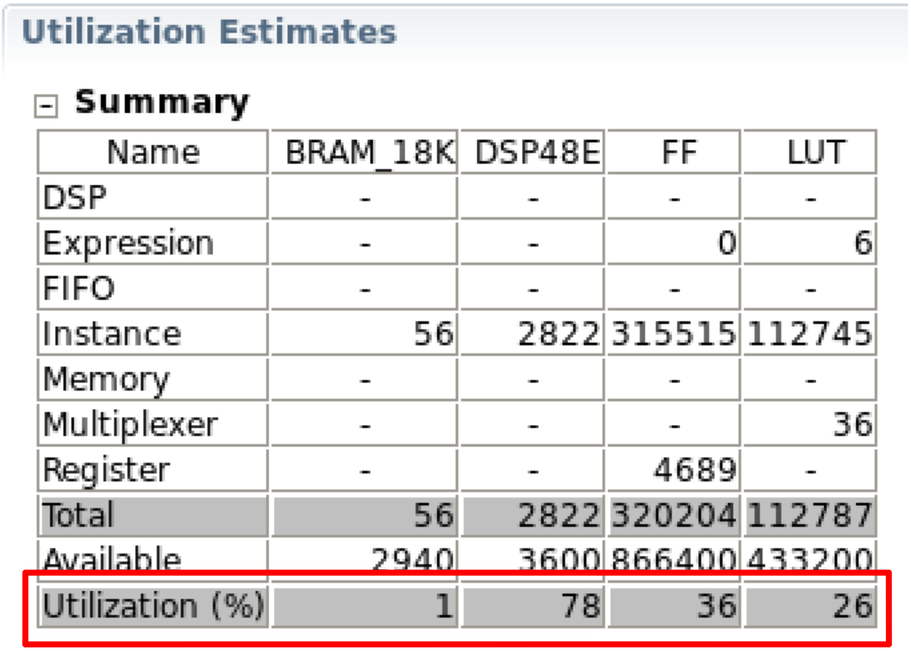




Device	# of DSPs
Kintex-7 325T	840
Virtex-7 690T	3600
Kintex UltraScale KU115	5500
Virtex UltraScale+ VU9p	6800

NN firmware resource usage

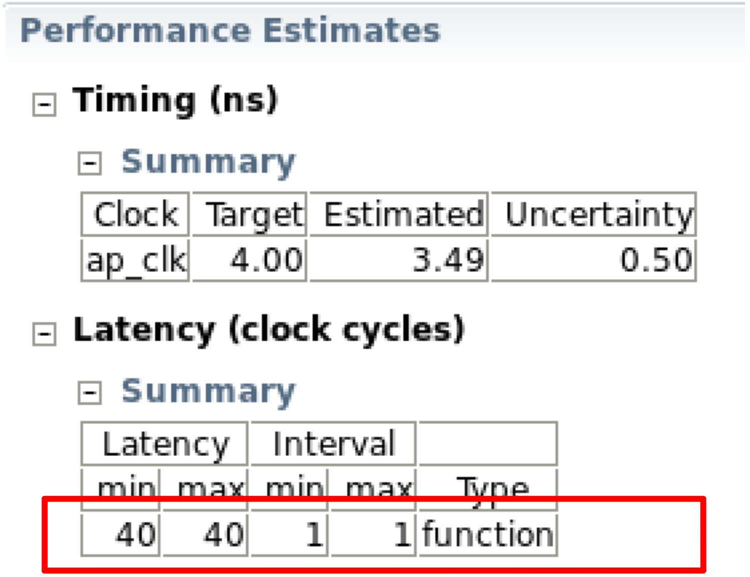
- Converted an earlier version of NN using HLS4ML
  - 80 input nodes, 3 hidden layers with 64/32/16 nodes, and 1 output node
  - Pruning is applied to remove 50% of the unimportant synapses
  - Use <18,8>-precision for inputs & output. 8 integer bits, 18 total bits
  - Note: current version is much smaller (fewer inputs and fewer nodes)!
- Xilinx Virtex-7 target FPGA (as in the MTF7 board)
  - Clock of 250 MHz



Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	6
FIFO	-	-	-	-
Instance	56	2822	315515	112745
Memory	-	-	-	-
Multiplexer	-	-	-	36
Register	-	-	4689	-
Total	56	2822	320204	112787
Available	2940	3600	866400	433200
Utilization (%)	1	78	36	26



Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	4.00	3.49	0.50

Latency (clock cycles)

Summary

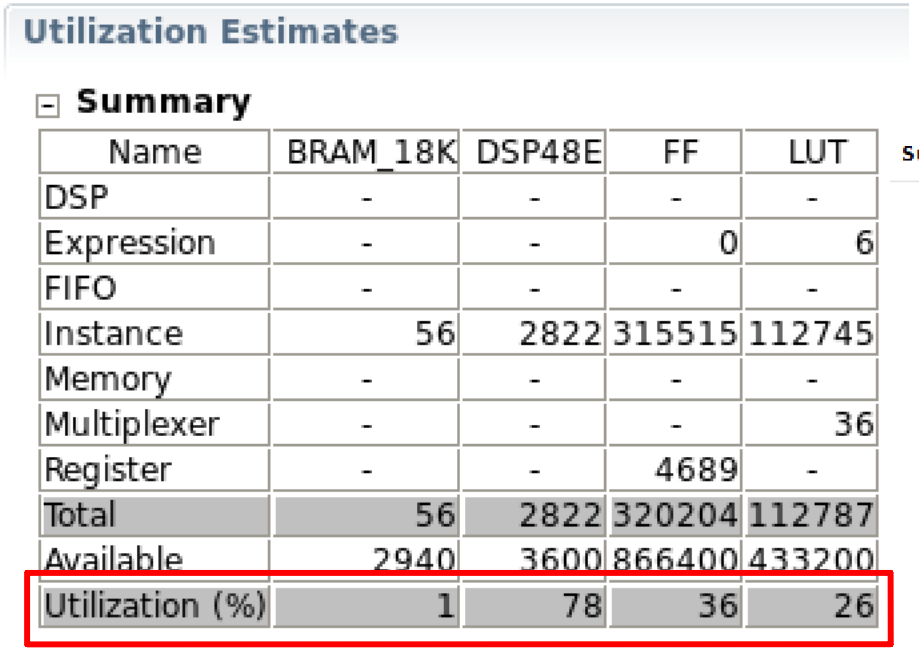
Latency		Interval		
min	max	min	max	Type
40	40	1	1	function

~160ns fixed latency

NN firmware resource usage

- HLS estimates and actual FW implementation resources agree reasonably well
- DSP usage is spot on. LUT and FF are over-estimated in HLS (seen before). Conclusions agree well with earlier observations using generic NNs.

HLS Estimates

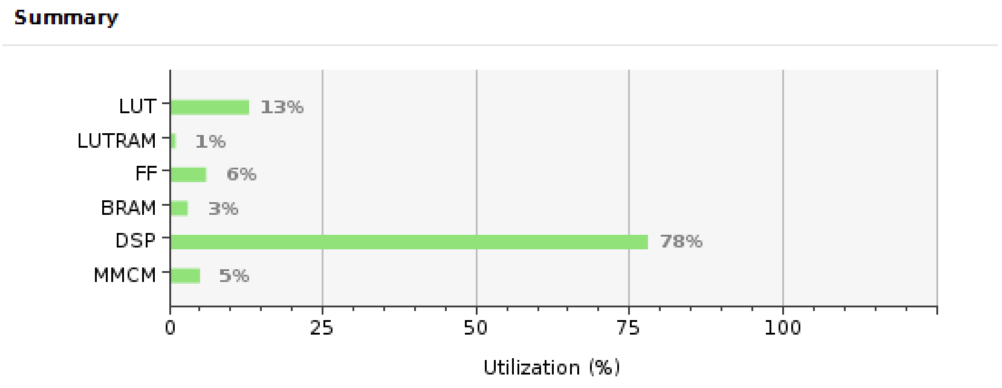


Utilization Estimates

Summary

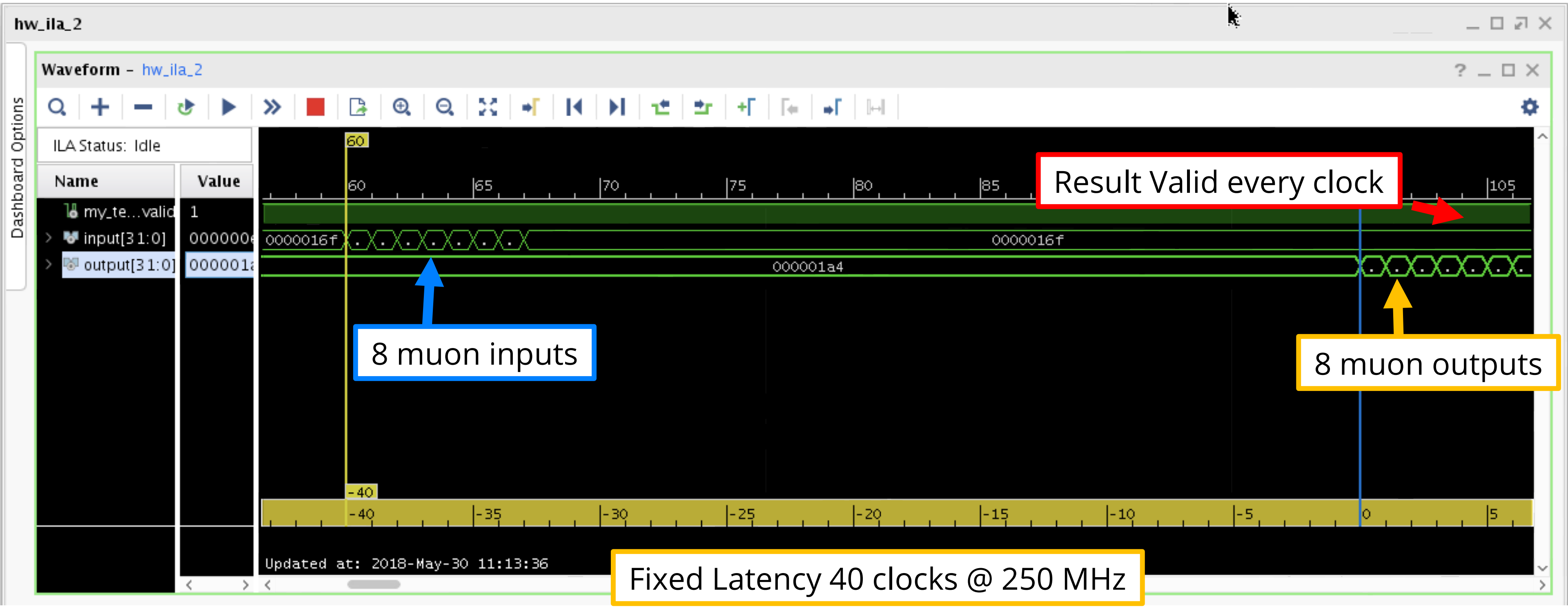
Name	BRAM_18K	DSP48E	FF	LUT
DSP	-	-	-	-
Expression	-	-	0	6
FIFO	-	-	-	-
Instance	56	2822	315515	112745
Memory	-	-	-	-
Multiplexer	-	-	-	36
Register	-	-	4689	-
Total	56	2822	320204	112787
Available	2940	3600	866400	433200
Utilization (%)	1	78	36	26

Design synthesis in Vivado 2018.1



NN firmware in MTF7 hardware

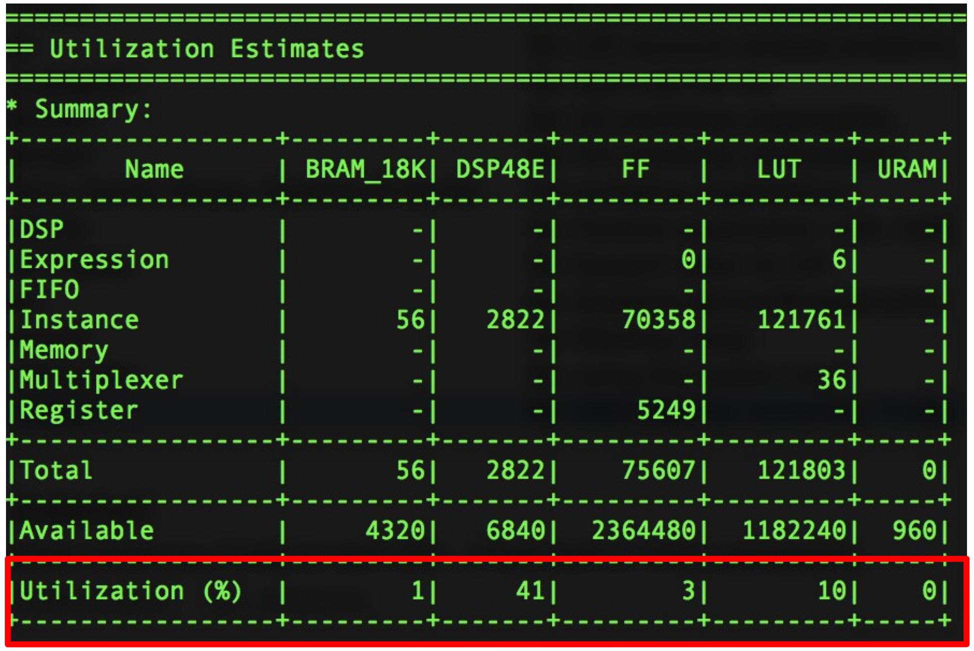
- Reading inputs corresponding to 8 simulated muons and calculating their p_T.
- HLS IP and design target frequency 250 MHz.



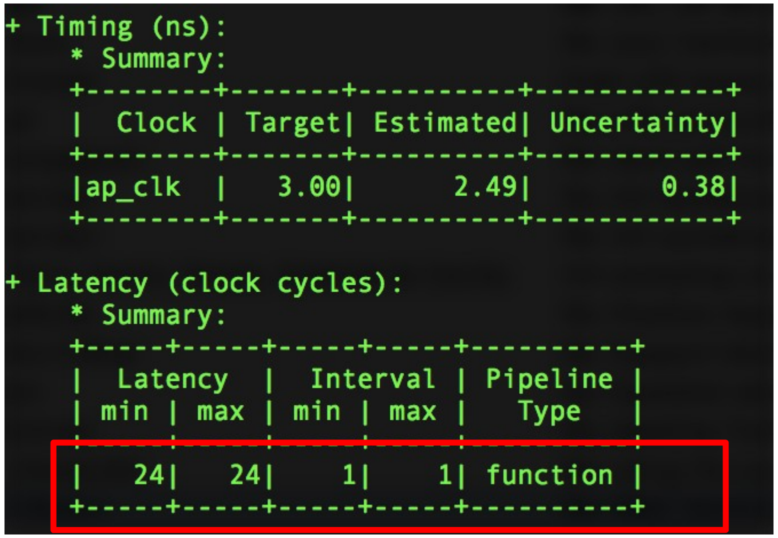
Expect latency of ~40 + N_{muons} clocks
99% percentile of tracks to fit in both endcaps is 5. Per processor 5/12 = 0.4

Extrapolate to Virtex Ultrascale+

- A look at the future devices:
  - Tried VU9P US+ FPGA, Device ID: xcvu9p-flga2104-2-i
  - Targeted by the USCMS APd board
  - Clock of 333 MHz



```
================================================================
== Utilization Estimates
================================================================
* Summary:
+---------------+---------+-------+--------+--------+-----+
|     Name      | BRAM_18K| DSP48E|   FF   |   LUT  | URAM|
+---------------+---------+-------+--------+--------+-----+
|DSP            |        -|      -|       -|       -|    -|
|Expression     |        -|      -|       0|       6|    -|
|FIFO           |        -|      -|       -|       -|    -|
|Instance       |       56|   2822|   70358|  121761|    -|
|Memory         |        -|      -|       -|       -|    -|
|Multiplexer    |        -|      -|       -|      36|    -|
|Register       |        -|      -|    5249|       -|    -|
+---------------+---------+-------+--------+--------+-----+
|Total          |       56|   2822|   75607|  121803|    0|
+---------------+---------+-------+--------+--------+-----+
|Available      |     4320|   6840| 2364480| 1182240|  960|
+---------------+---------+-------+--------+--------+-----+
|Utilization (%)|        1|     41|       3|      10|    0|
+---------------+---------+-------+--------+--------+-----+
```



```
+ Timing (ns):
    * Summary:
    +--------+-------+----------+-----------+
    | Clock  | Target| Estimated| Uncertainty|
    +--------+-------+----------+-----------+
    |ap_clk  |   3.00|      2.49|       0.38|
    +--------+-------+----------+-----------+

+ Latency (clock cycles):
    * Summary:
    +-----+-----+-----+-----+----------+
    |  Latency  |  Interval | Pipeline |
    | min | max | min | max |   Type   |
    +-----+-----+-----+-----+----------+
    |   24|   24|    1|    1| function |
    +-----+-----+-----+-----+----------+
```

Fixed Latency is 24 cycles
@333MHz this is 72ns

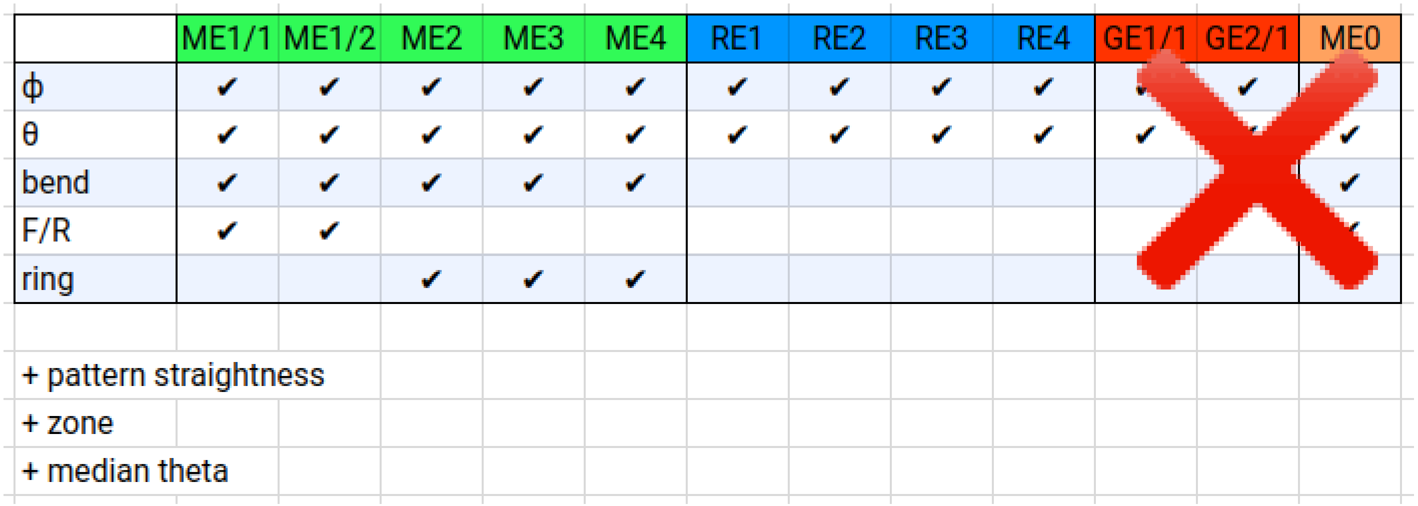
The algorithm should fit comfortably in FPGAs considered for Phase-2 boards

Results from Sergo

Run-3 NN prototype

NN configuration for Run 3

- Plan to develop and run NN in parallel in Phase-1 EMTF for Run 3
  - Run in parasitic mode a la K-BMTF during Run 2 (if able to fit in the MTF7 board)
- Develop a Run 3 NN prototype
  - Without GE1/1 for now. It will be added in the future.
  - Using only CSC and RPC inputs. Retrain the NN with 31 features.
    - Note: the new CSC bend is used. Forgot to change it back.
  - Using NN with reduced size: 24/16/12 nodes in the 3 hidden layers.

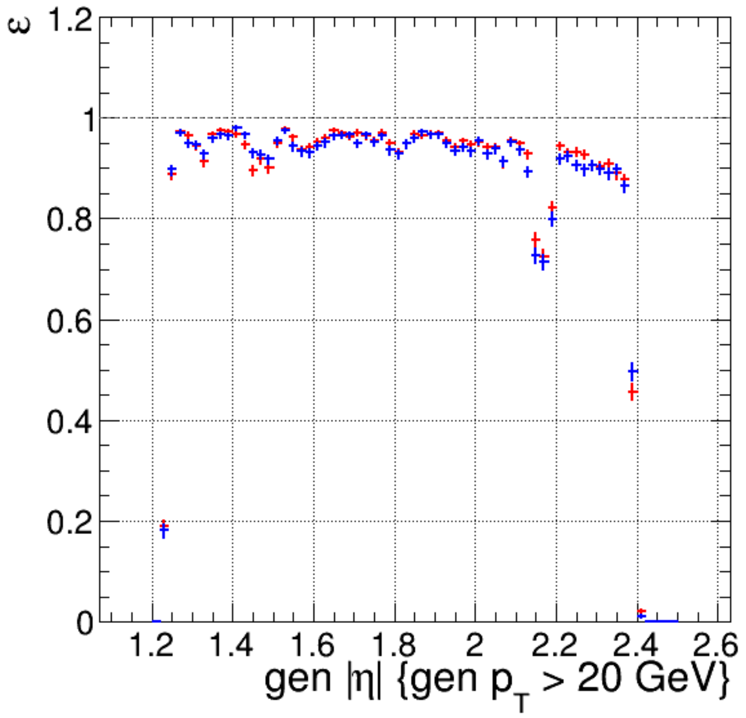


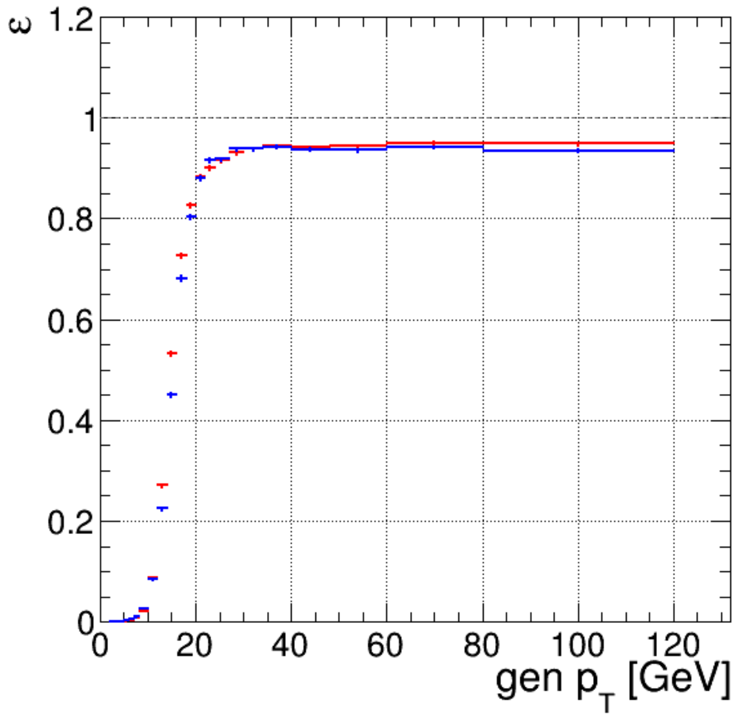
	ME1/1	ME1/2	ME2	ME3	ME4	RE1	RE2	RE3	RE4	GE1/1	GE2/1	ME0
ϕ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	[illegible]
θ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bend	✓	✓	✓	✓	✓							✓
F/R	✓	✓										✓
ring			✓	✓	✓							

+ pattern straightness
+ zone
+ median theta

Run 3 NN: efficiency

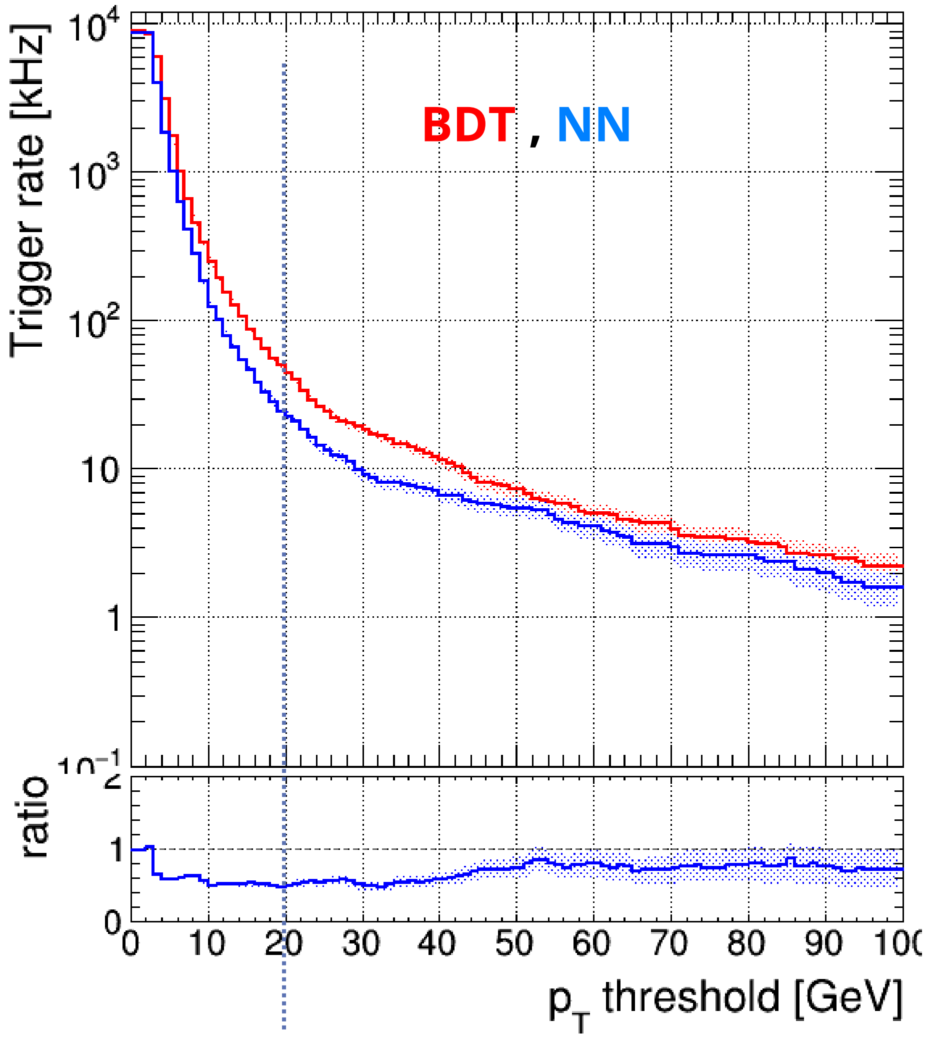
BDT , NN





- Plateau efficiencies of the two algorithms are similar
- NN underperforms at high p_T (fluctuation?)– something to look into more carefully
- Note: the inputs to BDT and NN are not identical. The EMTF emulator has not been adapted to work with NN.

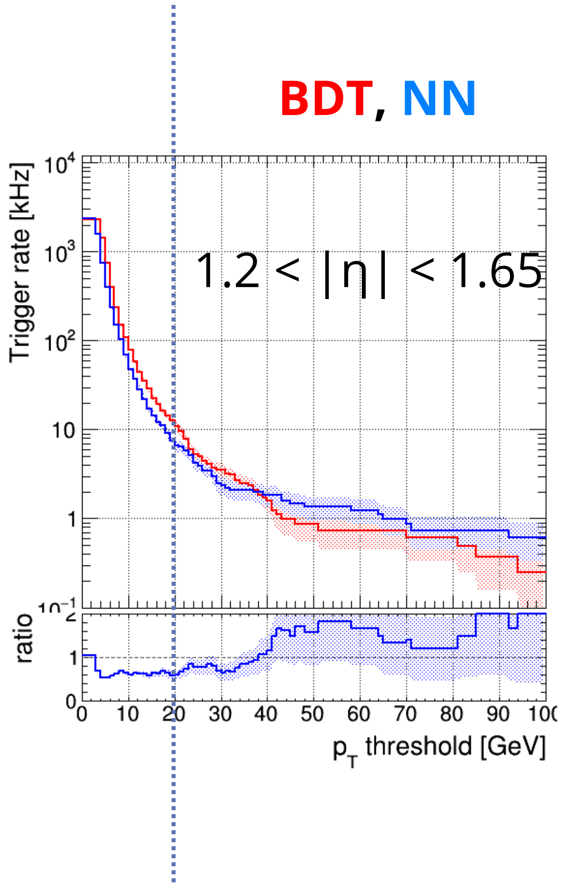
Run 3 NN: rates @ 200 PU

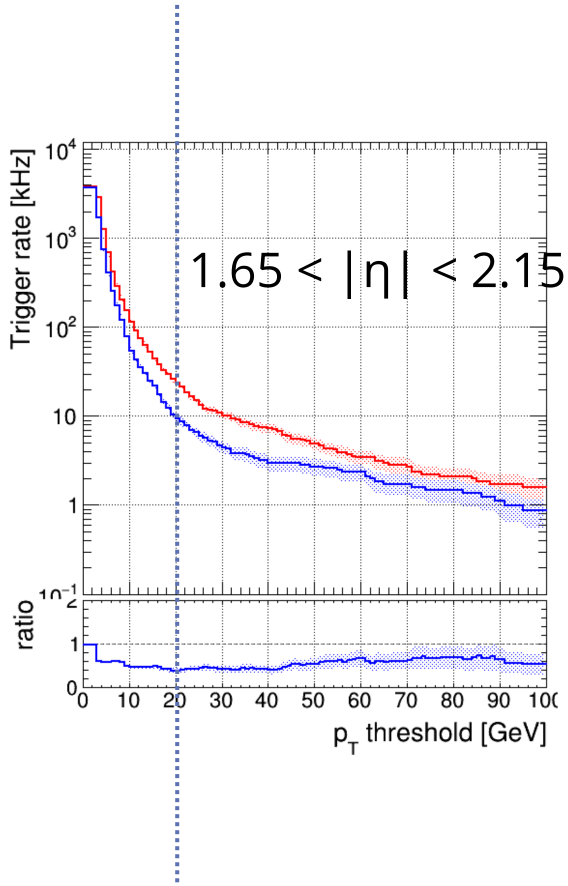


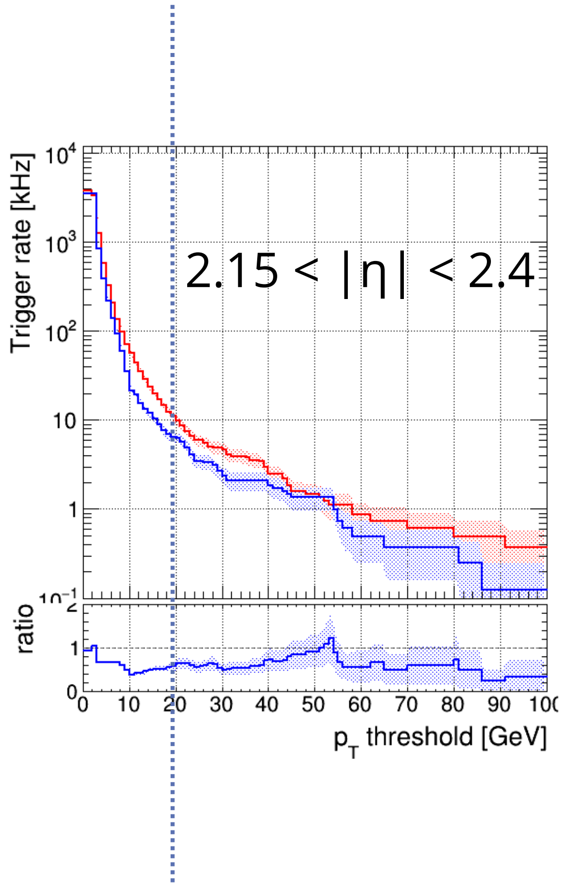
About 40% reduction in the inclusive rates

Run 3 NN: rates @ 200 PU

BDT, **NN**







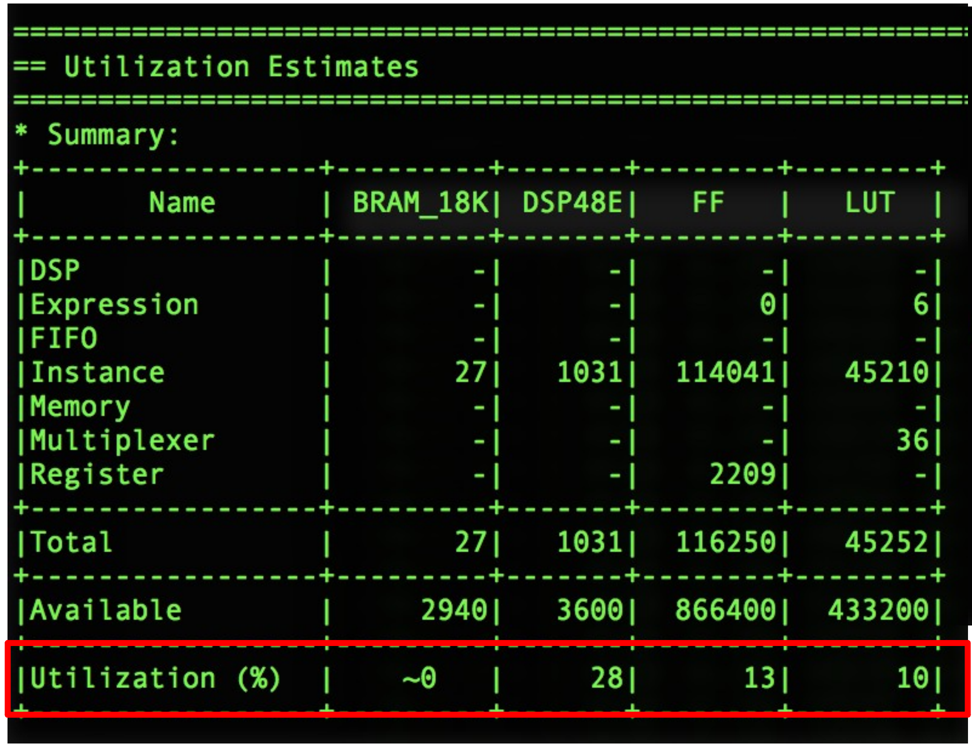
39% rate reduction @ 20 GeV

61% rate reduction @ 20 GeV

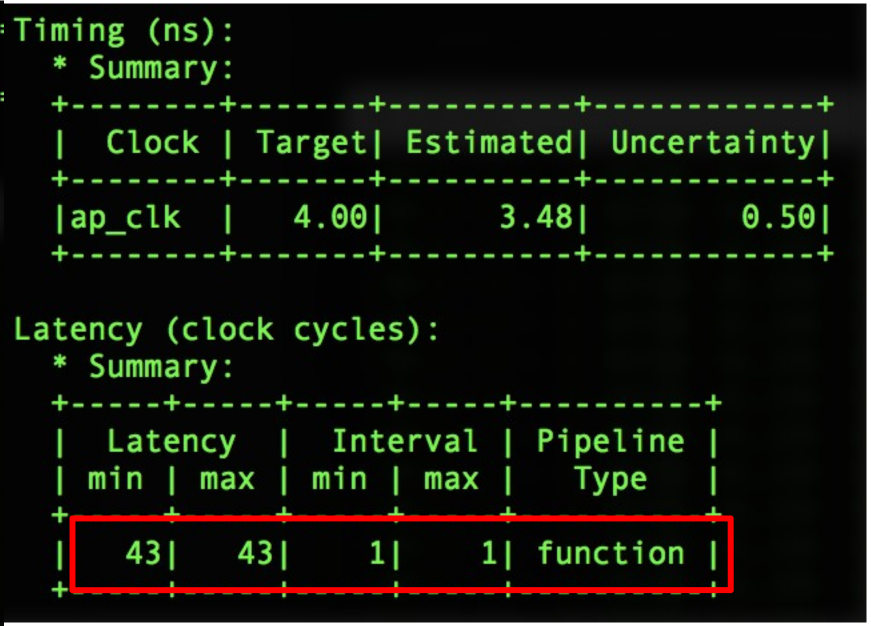
36% rate reduction @ 20 GeV

Run 3 NN: resource usage

- Converted NN into VHDL codes using HLS4ML, target Virtex-7 FPGA
  - Use <18,8>-precision for inputs & output. 8 integer bits, 18 total bits
  - Clock of 250 MHz
  - Only 28% DSP utilization



```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+--------+--------+
|       Name      | BRAM_18K| DSP48E|   FF   |   LUT  |
+-----------------+---------+-------+--------+--------+
|DSP              |        -|      -|       -|       -|
|Expression       |        -|      -|       0|       6|
|FIFO             |        -|      -|       -|       -|
|Instance         |       27|   1031|  114041|   45210|
|Memory           |        -|      -|       -|       -|
|Multiplexer      |        -|      -|       -|      36|
|Register         |        -|      -|    2209|       -|
+-----------------+---------+-------+--------+--------+
|Total            |       27|   1031|  116250|   45252|
+-----------------+---------+-------+--------+--------+
|Available        |     2940|   3600|  866400|  433200|
+-----------------+---------+-------+--------+--------+
|Utilization (%)  |    ~0   |     28|      13|      10|
+-----------------+---------+-------+--------+--------+
```

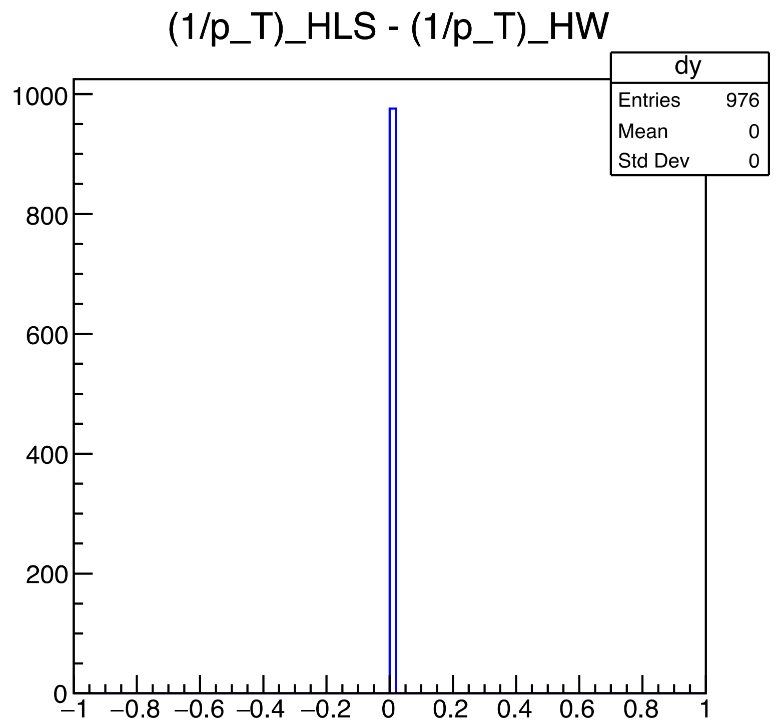


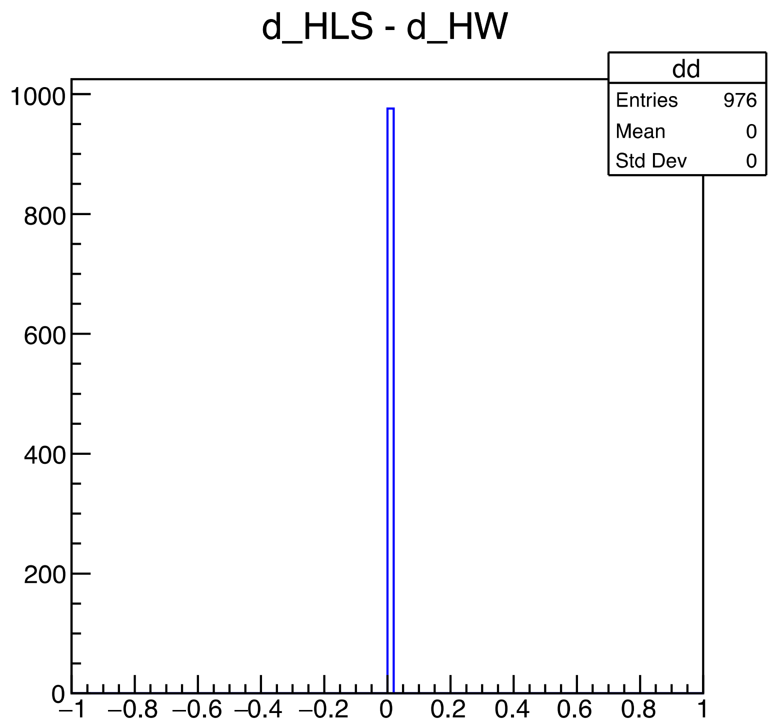
```
Timing (ns):
  * Summary:
  +--------+-------+----------+------------+
  |  Clock | Target| Estimated| Uncertainty|
  +--------+-------+----------+------------+
  |ap_clk  |   4.00|      3.48|        0.50|
  +--------+-------+----------+------------+

Latency (clock cycles):
  * Summary:
  +-----+-----+-----+-----+----------+
  |  Latency  |  Interval | Pipeline |
  | min | max | min | max |   Type   |
  +-----+-----+-----+-----+----------+
  |   43|   43|    1|    1| function |
  +-----+-----+-----+-----+----------+
```

Large scale HW tests

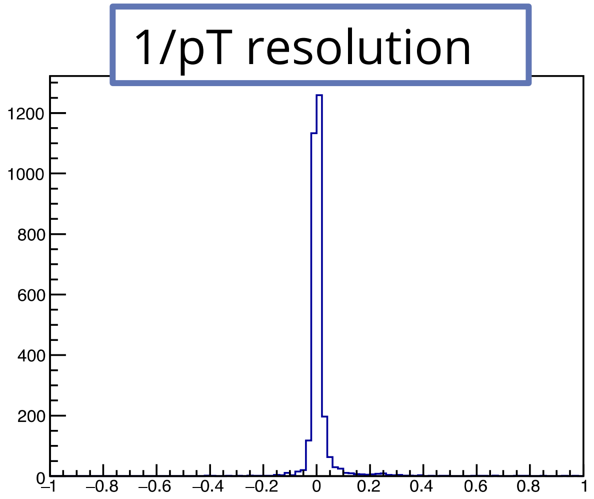
- Inject ~1000 muons in a pipelined burst
- Compare HW outputs (q/p_T and PU discriminator) with Vivado HLS simulation
  - Perfect agreement observed!

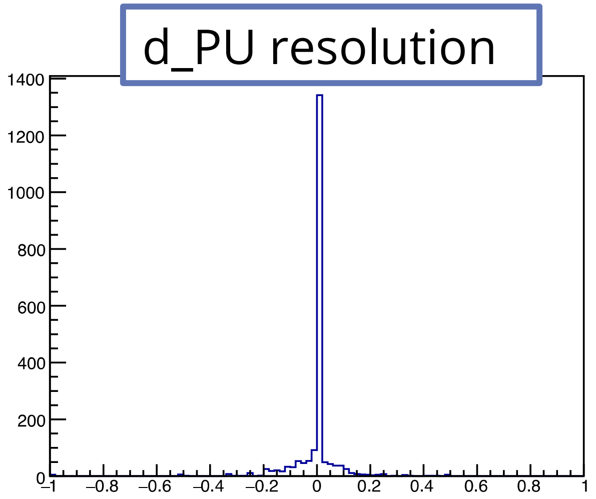


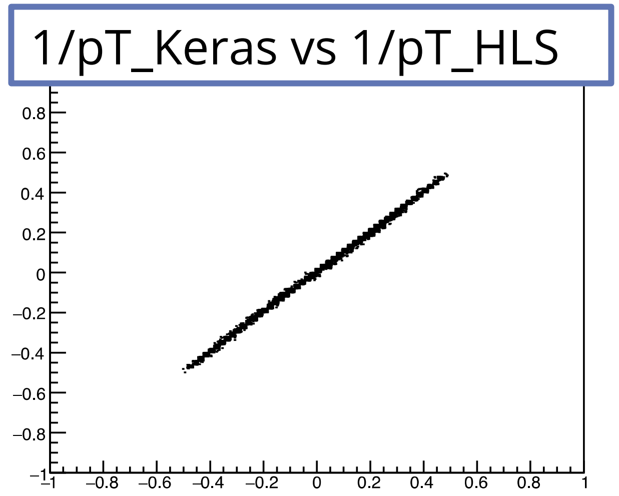


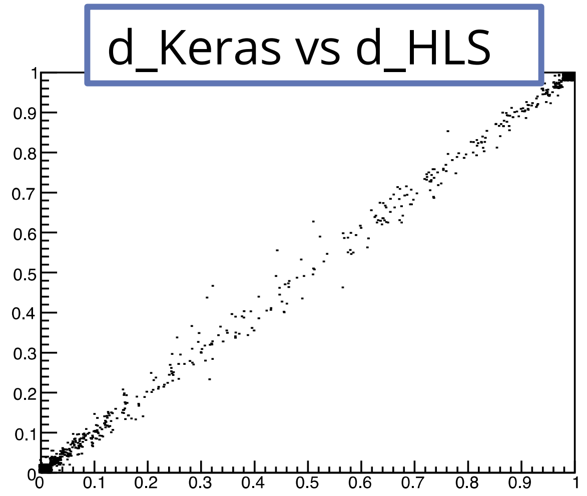
Large scale HW tests

- Compare Keras vs HLS simulation outputs. Keras uses floating-point precision, HLS simulation uses <18,8> precision. Resolution is typically at 10^{-3} level.









Summary

- A “v1” version of EMTF++ has been developed.
  - 11.0 kHz @ 20 GeV trigger threshold for 200 PU, which is **4x rate reduction**. Larger rate reduction in the region with new muon detectors.
  - Overall **efficiency has been improved**.
  - Addition of the new muon detectors, particularly ME0, helps significantly.
- Plans:
  - Need to investigate if EMTF++ can be implemented in the firmware, and rework it if necessary.
  - Need to work on putting EMTF++ into CMSSW
- A lot of progress made in the **NN implementation in FW**.
  - HLS4ML toolkit allows to convert NN model into VHDL codes, which can be used to generate the firmware. HLS estimates of resource usage are excellent.
  - NN prototype firmware has been implemented on the real hardware (MTF7). HW outputs have perfect agreement with HLS simulation using fixed-point precision.
- FW development plans:
  - Plan to **develop and run NN** in parallel in Phase-1 EMTF **for Run 3**.
  - NN retrained for Run 3 configuration (no GE1/1 yet). Very good results with about 2x rate reduction @ 20 GeV for 200 PU.

```
if (goingToSegFault) {
    dont();
}
```

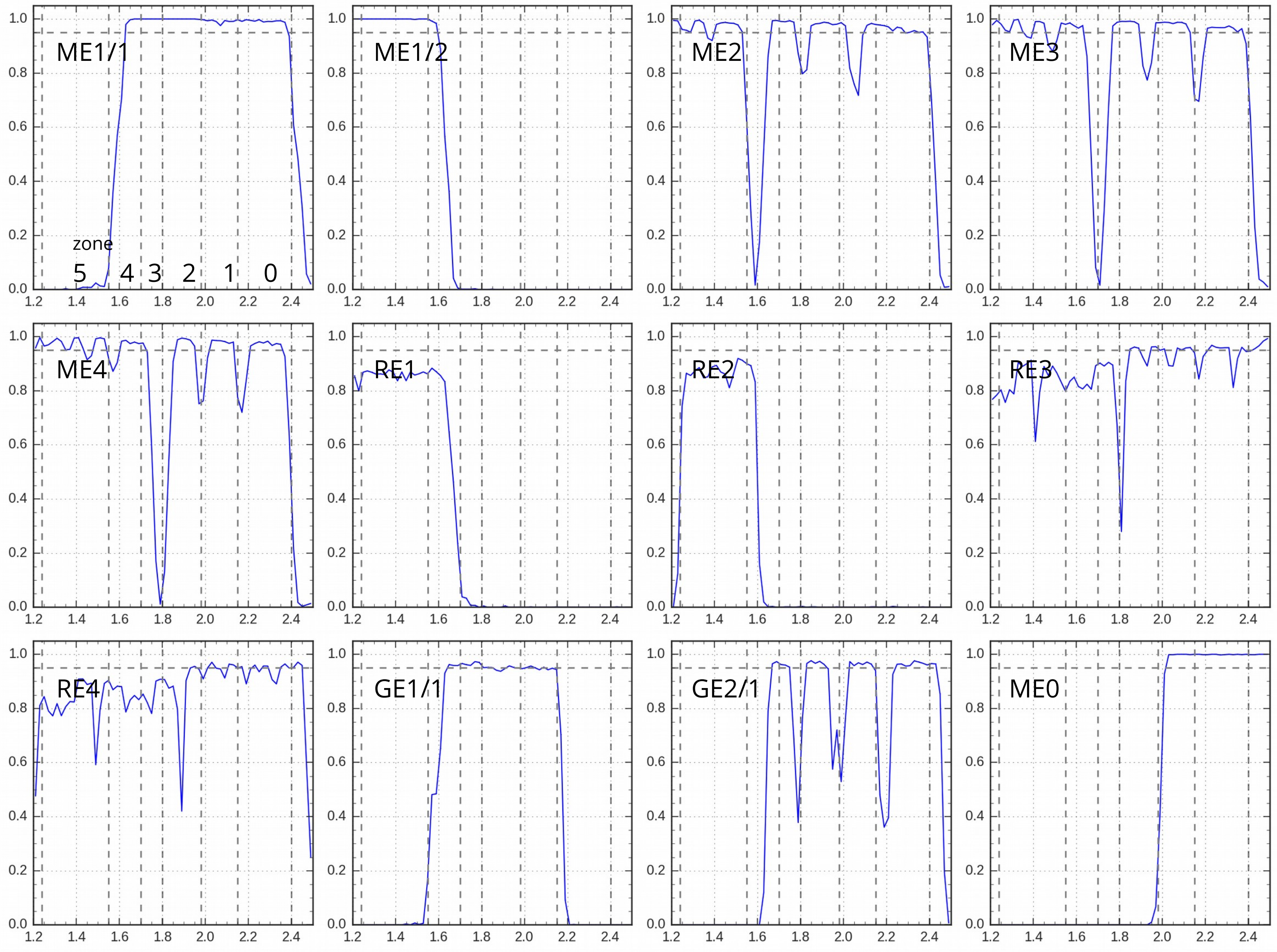
Backup

Assumptions

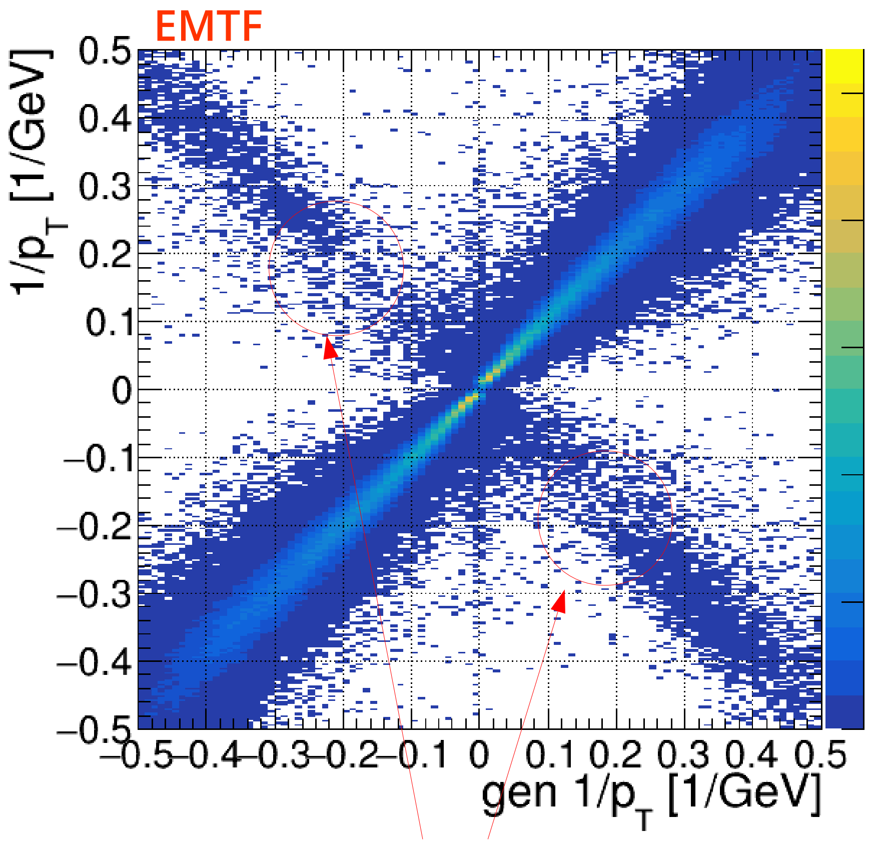
- The following efficiency and rate plots are obtained with the assumptions:
  - Performance of the ME0 trigger primitive in the current simulation is realistic
  - CSC CLCT comparator digi fits are possible
  - The new patterns can fit inside the FPGA
  - For rate plots, I'm using # of colliding bunches = 2808

Zones

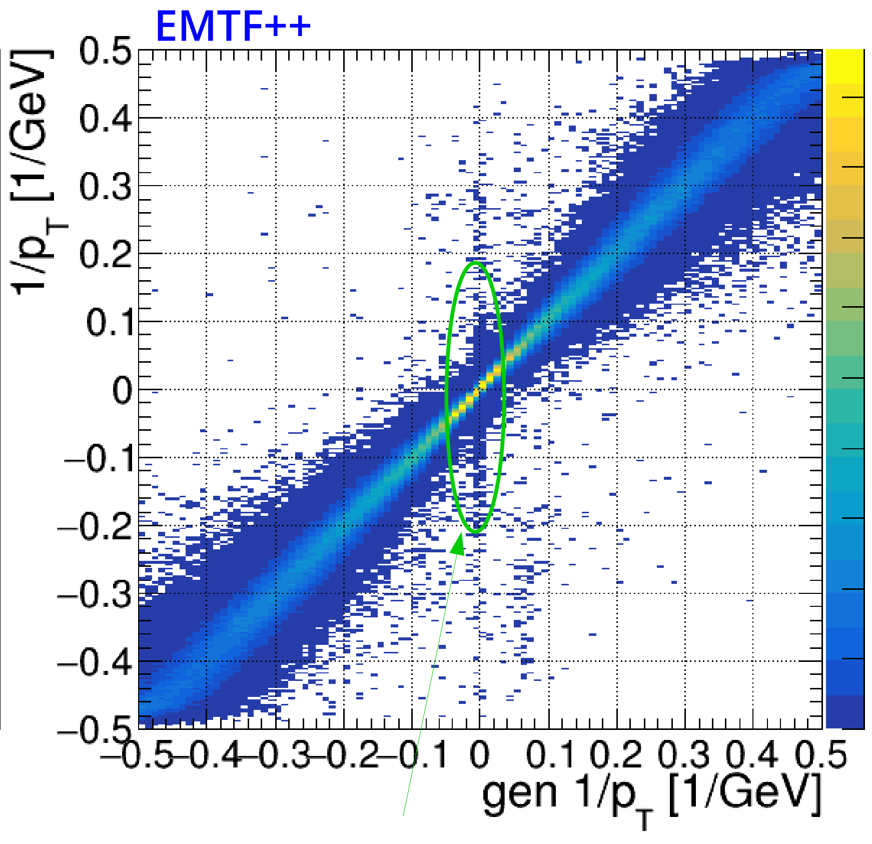
Efficiency for each detector type



Results: p_T resolution



Incorrect charge assignment as the charge was not included in training



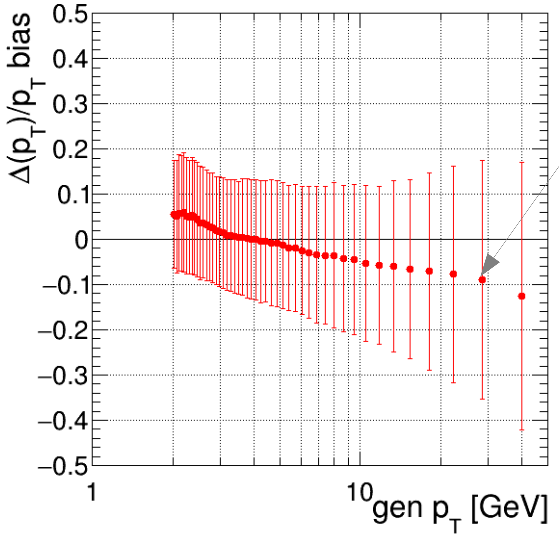
Less spread at high p_T

< 20% resolution at 20 GeV!

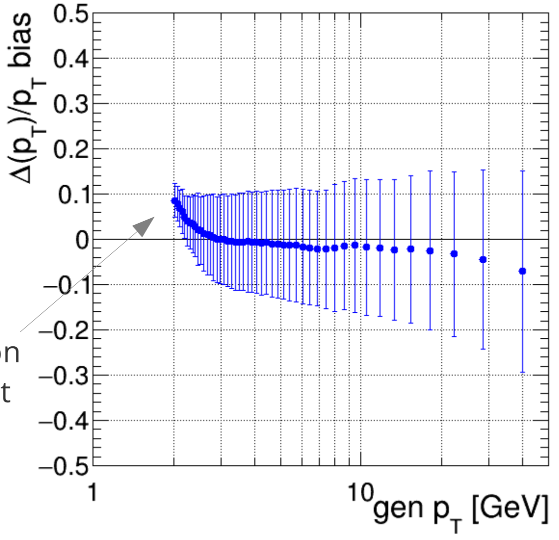
Results: p_T resolution

EMTF

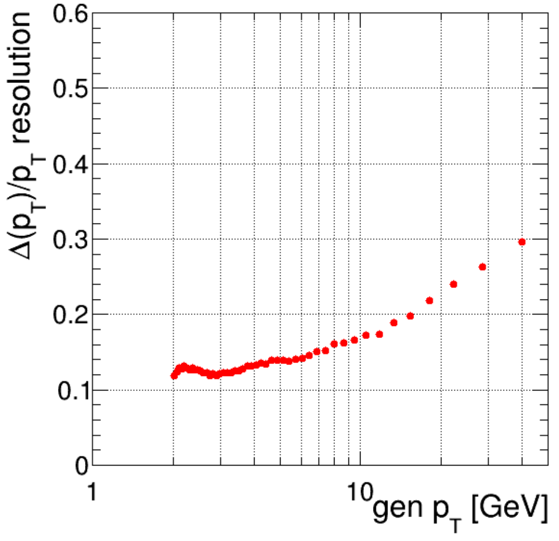
EMTF++

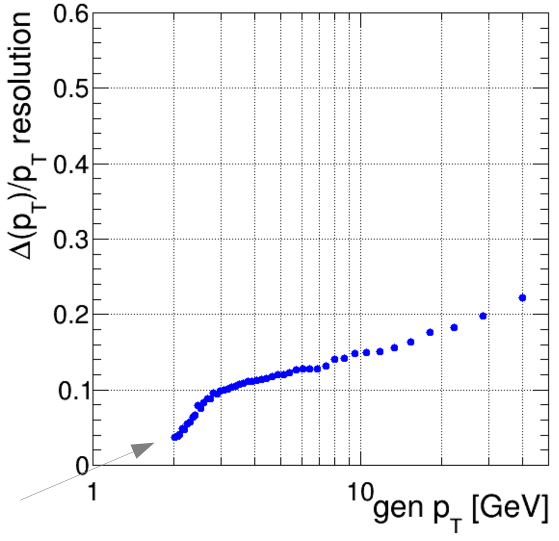


Underestimation at high p_T and has a mild p_T dependence



Overestimation at 2-3 GeV, but less than 10%



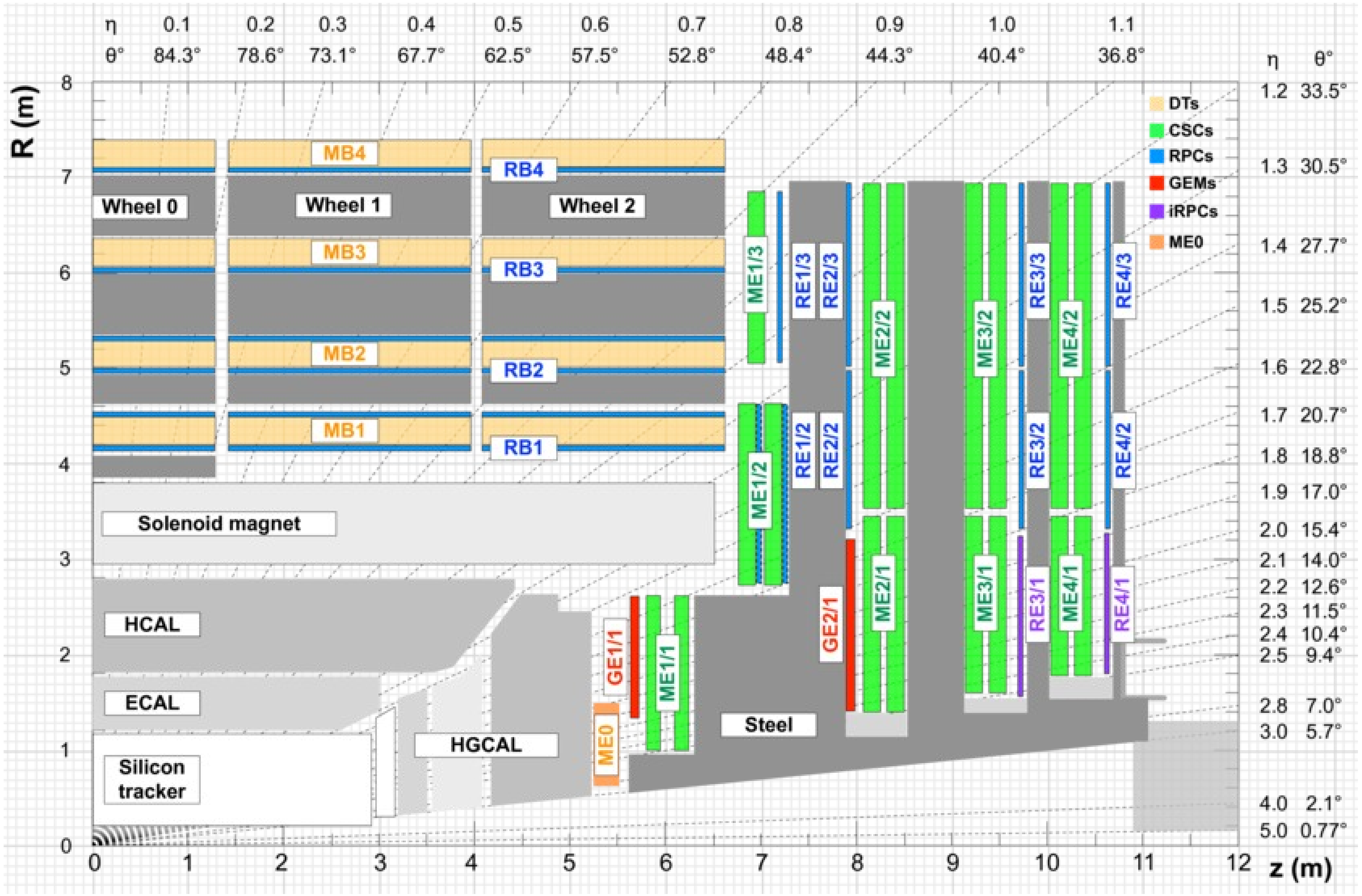


Due to bias at 2-3 GeV

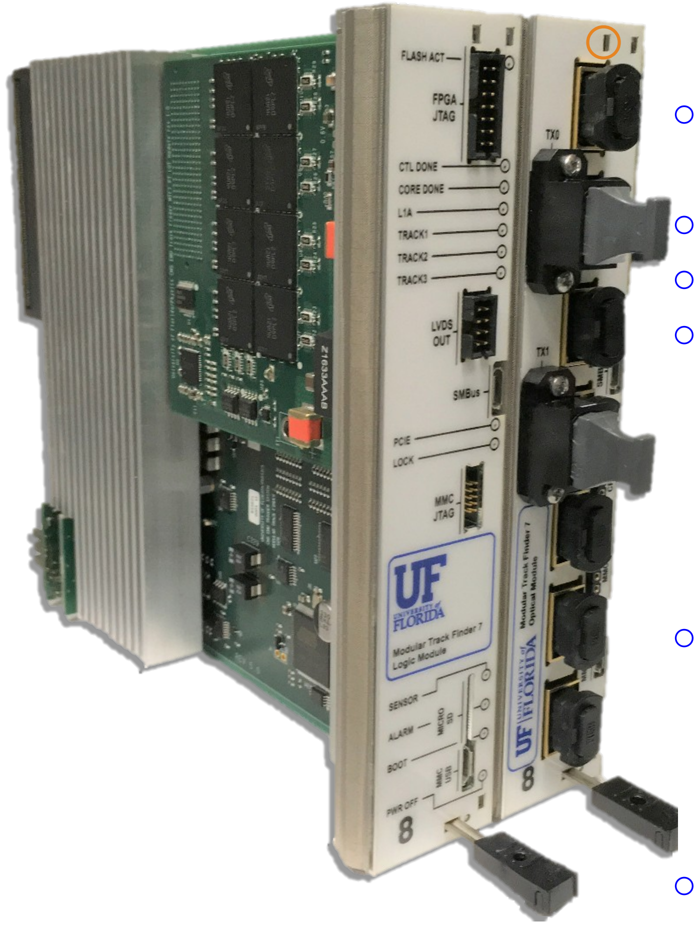
Algorithm development plans and timeline

- Initial EMTF algorithm developed
  - Still to migrate to CMSSW_10 release
- Continue refining the track-finding algo, and PT assignment with ML
  - Add ME0
  - Make sure all potentially useable data are there
    (need a dedicated GEM-CSC bend angle, and how best to measure?)
  - Extend to overlap region
  - Trim to minimum number of variables, bits, layers, connections, etc.
  - Aim to minimize impact of pileup segments
- Adaptation of Kalman filter approach to the endcap
  - Or otherwise develop dedicated displaced muon patterns, PT assignment
  - Could use a volunteer!
- Incorporation of RPC timing information
  - For pileup mitigation and HSCP identification
  - Could use a volunteer!
- Aim for a refined set of algorithms for the EMTF by the end of 2018 (although surely not the “final” algorithm)

Phase-2 CMS detector quadrant



Running in Hardware



Modular Track Finder 7

- µTCA standard
  - Adopted by CMS for LS1 upgrades
- Based on Virtex-7 FPGA
- Modular design
- Logic module (left):
  - Main FPGA – Virtex-7
  - Control FPGA – Kintex-7
  - Control interfaces:
    - PCI Express, Gen2, 2 lanes
    - IPbus
- Optical module (right):
  - 7 receivers 12 channels each, up to 10 Gbps
    - 84 RX channels total (currently used: 56)
  - 2 transmitters 12 channels each, up to 10 Gbps
    - 24 TX channels total (currently used: 1)
- Logic and Optical modules connect via custom 10Gbps backplane section
- PT LUT module (mezzanine)
  - Based on Reduced Latency DRAM (RLDRAM 3)
  - 1 G x 9 bits of space

Samples

```
dataset=/SingleNeutrino/PhaseIIFall17D-L1TPU200_93X_upgrade2023_realistic_v5-v1/GEN-SIM-DIGI-RAW
dataset=/SingleNeutrino/PhaseIIFall17D-L1TPU140_93X_upgrade2023_realistic_v5-v1/GEN-SIM-DIGI-RAW
```

Software

```
CMSSW_10_1_5
+ changes from cms-l1t-offline:l1t-phase2-v2.16.6
+ modified EMTF emulator to include GEM, iRPC and ME0
```