GOOD MEMORY AND NEURAL NETS

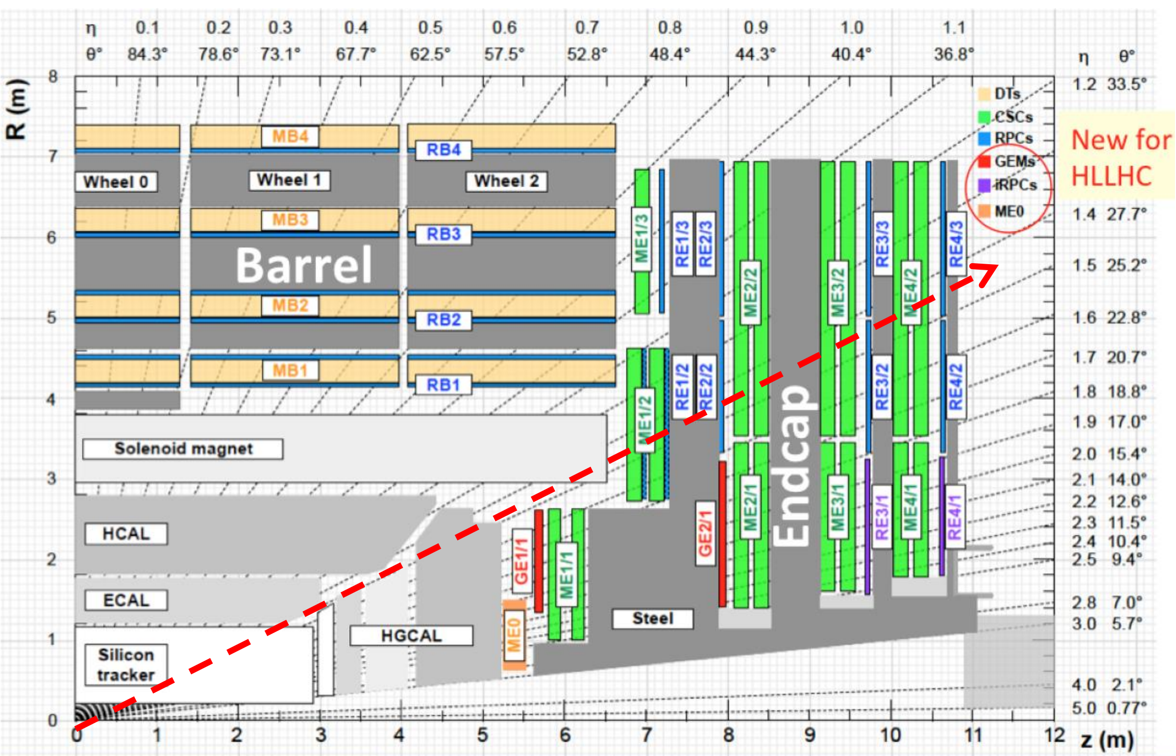
MACHINE LEARNING IN THE (L1) TRIGGER

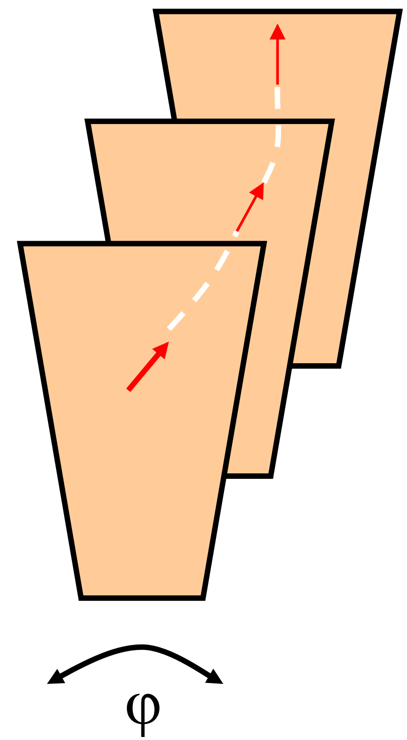
Darin Acosta, University of Florida

(on behalf of the UF and Rice Endcap Muon Trigger groups)

Context: L1 Endcap Muon Track-Finder(s)

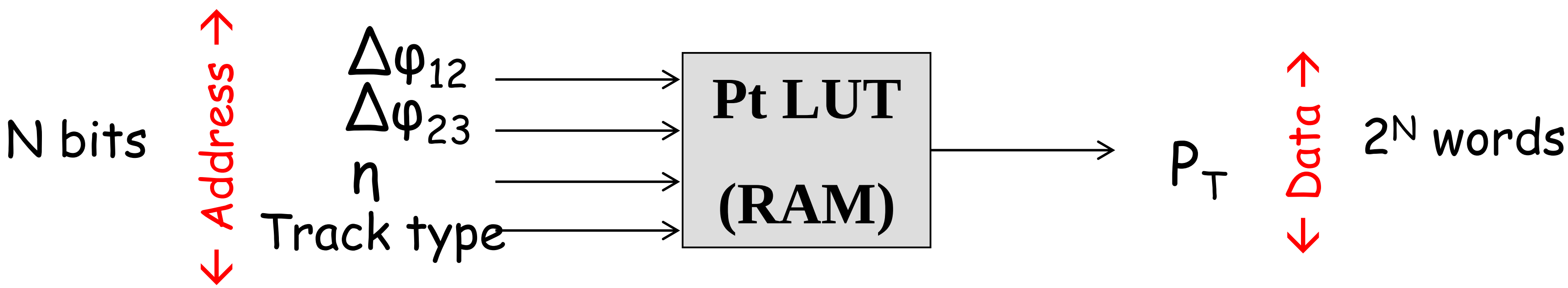
- A standalone muon tracking trigger (w/o inner tracker)
  - Link CSC, RPC, (+GEM) track segments into 3D tracks
  - Measure track p_T in the nonuniform fringe field of the endcap
    - Extracted from φ and η deflections from detector to detector when traversing the disks





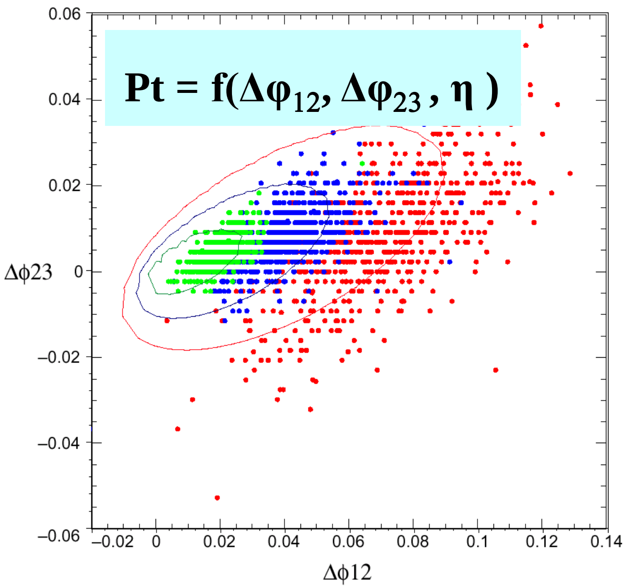
P_T Calculation by Memory

- The P_T is calculated from a memory look-up table
  - A "cheat" to do the calculation quickly (~50ns) in the L1 trigger.
    - Also must be fast for random addressing…
  - Don't really calculate it online at all (no CPU)
  - Instead, pre-calculate offline the muon momentum using whatever algorithm you want and with however much computing resources you have!
    - But you must do this for every possible input to the memory
- The challenge:
  - You must squeeze all the data for your track fit into the address for your memory
    - N bits of of data requires a memory of 2^N addresses

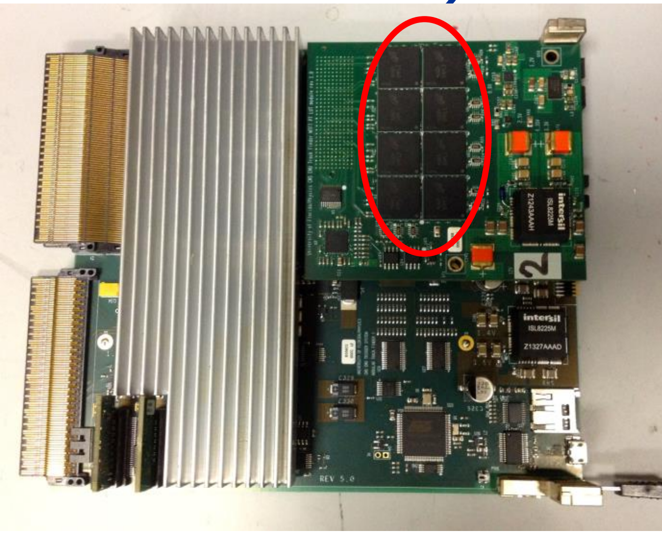


Version 1: CSC Track-Finder, 2005-2015

- 12 VME processors
  - Xilinx Virtex-5 FPGAs and memory
- P_T calculated from an SRAM memory look-up table
  - Largest available at time to do the job: 4MB → 22 bit address space
- Algorithm
  - Likelihood-based fit using Δφ bending between at most 3 detector stations to assign p_T
  - Multiple scattering in iron carries momentum information in addition to magnetic bending
- Data compression
  - Introduced nonlinear scales to "shoe-horn" in as much data as possible



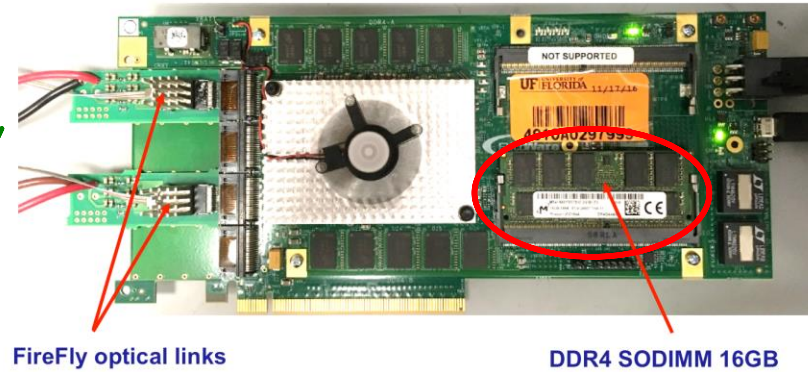
Version 2: Endcap Muon Track-Finder, (Phase-1 Upgrade of Previous, 2016+)



- 12 μTCA double-module processors
  - Xilinx Virtex-7 FPGA and memory
- P_T calculated from Reduced Latency DRAM
  - 1 GB → 30 bit address space
  - +8 bits (only) over previous CSCTF
- Algorithm
  - Machine Learning: Boosted Decision Trees (BDTs) used for regression to assign P_T
  - Can use $\Delta\varphi$ bending between 4 detector stations, and $\Delta\eta$, and bend angle in first station
  - But note, as before, algorithm is run offline and stored in memory

Version 3: Endcap Muon Track-Finder (Phase-2 Upgrade of Previous, 2026+)

- 12 ATCA processors
  - Xilinx Ultrascale+ FPGA and memory
- P_T calculated from DDR4
  - ~256 GB → 38 bit address space
  - +8 bits (only) over previous EMTF
- Algorithm
  - In development! But starting from current EMTF as conservative baseline
  - Expand P_T assignment with more angular measurements from new HLLHC muon detectors
  - Continuing ML as P_T assignment algorithm



Xilinx evaluation card

Appendix B - Schematic of 2017 PT LUT address bits

← 30 bits to encode data →

PT LUT address bits	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**4**	**4**	**3**	**2**	**1**	**0**
Two-station tracks	0	0	0	0	mode2			5b_theta					3b_clctB			3b_clctA			frB/A		3b_dThAB			7b_dPhAB						
Station 2-3-4 tracks	0	0	0	1	5b_theta					2b_rpc		clct2		fr	3b_dTh24			s	5b_dPh34					7b_dPh23						
Three-station tracks	0	mod3		5b_theta					2b_rpc		clctA		frB/A		3b_dThAC			s	5b_dPhBC					7b_dPhAB						
Four-station tracks	1	8b_theta_rpc_clct1								fr	dTh14		s34-23		4b_dPh34				5b_dPh23					7b_dPh12						

*** Some names truncated for space. **Two-station:** [frB/A] = [frB][frA]. **Station 2-3-4:** [fr] = [fr2], [s] = [sph34].
Three-station: [mod3] = [mode3], [frB/A] = [frB][frA], [s] = [sphBC]. **Four-station:** [fr] = [fr1], [s34-23] = [sph34][sph23], [dTh14] = [2b_dTh14].

- Squeeze in all angular differences from all detectors without sacrificing precision
  - A data science project in itself!
  - Nonlinear binning, and address fields that are context driven
    - Provide the most data bits to the tracks that can be measured best
- Even larger address space for Phase-2 Upgrade allows additional information such as GEM-CSC bend angles

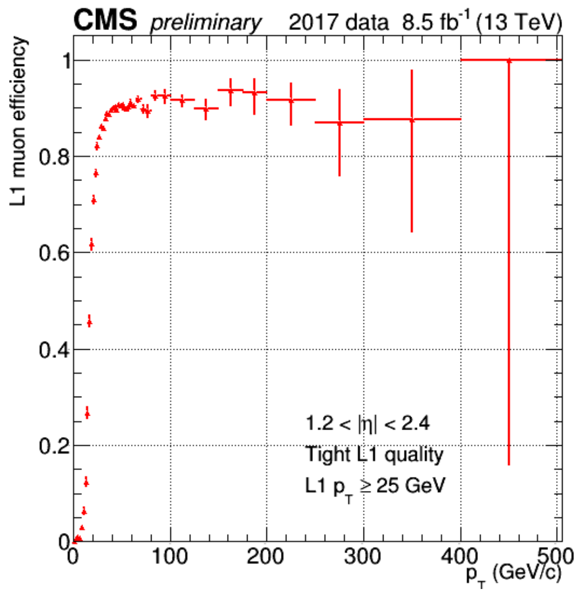
ML/BDTs for Regression

- Trigger application is somewhere between a classification problem and a regression
  - p_T above or below a threshold, but for multiple thresholds
- We use a transformation + loss function to focus on low p_T events (whose mismeasurement to high p_T drives the rate)
  - Target $1/p_T$ makes differences in low p_T count more in loss
  - Loss = $|1/p_{T,meas} - 1/p_{T,true}|^2$, but studied other loss functions
    - Focus on low p_T more → lower rate (good), lower effic. (bad)
    - Focus on low p_T less → higher rate (bad), higher effic. (good)
- With redundant measurements (4 detector stations), ML can identify outliers (e.g. TeV muon bremsstrahlung) and reject them to keep efficiency high
  - We used to have to introduce ad hoc algorithms to recover effic.
- See also ACAT2017 talk by A.Carnes on use of ML in L1Trigger (CMS CR -2017/357)

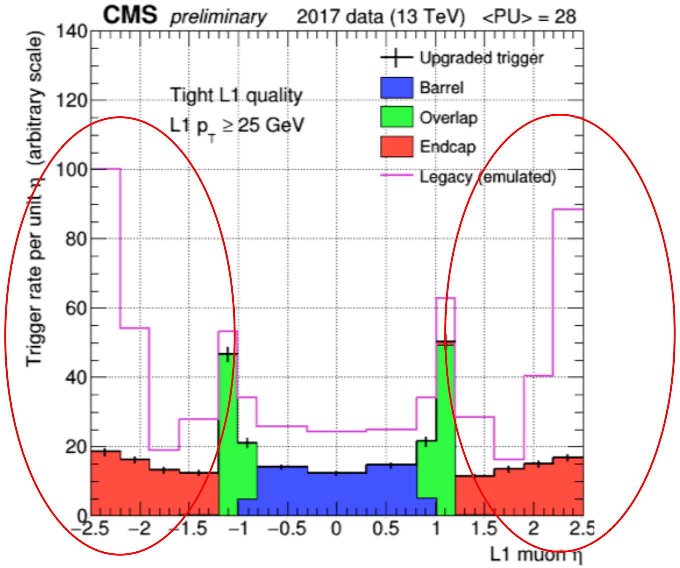
Current Endcap Muon L1 Trigger Results

(25 GeV threshold)

- Efficiency is high even to highest p_T (TeV-scale)



- Rate suppressed 3X in forward region relative to previous trigger, and comparable to barrel rate despite much less magnetic bending and high backgrounds



Future Training

- Currently studying improvements possible using Deep Neural Nets (DNNs) to improve performance beyond BDTs
  - RiceU taking lead on this
- Convolutional Neural Nets (CNNs) are already heavily used for image recognition, and tools readily available to process and train on images
- Interesting side project: translating a tracking problem to an image recognition problem!
  - UF and Rice groups are actively pursuing this possibility
  - Stay tuned!

Training on Data?

- The current P_T assignment training is based on MC simulation samples
  - Simple muon gun without any pileup…
  - Essentially concentrating on the "track fit" aspects of the problem, assuming perfect track building
- But with pileup and radiation-induced backgrounds in data, we can have wrong stub→track associations
  - See evidence for that in pileup dependence of trigger rate
- Also more complex algorithms, like Deep Neural Nets, require huge datasets, which becomes computationally expensive to generate
- Need to investigate training based on data
  - Real experimental conditions and real backgrounds!
  - But would need a rather large minimum bias data sample…
    - Need a pilot study

Training on Data in Situ?

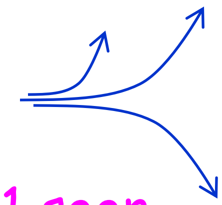
- Going beyond running on logged minbias data, how about running ML training on the HLT processor nodes?
  - HLT gets 10s of kHz of muons from L1
  - HLT has inner tracking information, with % resolution which is as good as perfect compared to standalone muon reco (20%)
    - CPU impact? How to collect and store training results?
- Phase-2: Self-train Muon Trigger entirely within L1?
  - For Phase-2, the L1 trigger also will have inner tracking info!
  - Access to MHz of muons!
  - Run L1 muon trigger in "training mode" first during a special run?
    - Or run training parasitically and asynchronously with more processors? Even a small fraction is still a high rate of muons
  - Does FPGA have enough resources for the training step?

Running Machine Learning on the FPGA?

- Avoid the address-space bottleneck of a LUT entirely and deploy the ML inference on the FPGA fabric
- This is big focus of computer engineering in industry and academics
  - Especially for the more computing intensive training step, which also is interesting for in situ training
- FPGAs are becoming coprocessors for computing, and available commercially
  - Amazon F1 instance, Microsoft catapult, Intel Xeon+FPGA, …
  - Can we leverage? (or even lead?)
    - Collaboration with UF ECE Dept, and ECE student (D.Ojika) to explore this option for us at UF.
    - Have an image classification example working on Altera FPGA and Amazon F1 (Xilinx)

Other Signatures

- Current ML application applies to reconstructing muons
- But there are other unique signatures:
  - Displaced muon-like particles
    - Identify tracks that do not project to IP, and measure momentum without beam constraint
      - Already in plans for HL LHC muon trigger
      - Also can come for free from Kalman filter approaches
  - $\tau \rightarrow 3\mu$
    - Muons are collimated (in η) and soft in p_T. May not penetrate full muon spectrometer
      - e.g. Planning to deploy a 2μ + stub trigger at L1 soon
    - Train to identify this signature within (HL)LHC environment
    - Access full luminosity with near zero p_T thresholds?
  - Muon (Lepton) jets, possibly displaced
    - Generalized collimated muons signature



Summary/Outlook

- Obviously can generalize beyond muon signatures
  - Calorimetric energy clustering, jet finding, etc.
- Started with a muon tracking trigger using a very large LUT for flexible calculations
- Machine learning algorithms are improving upon our "human learning" (likelihoods) methods
- Meanwhile electronics (FPGAs) and computing platforms are becoming blended, offering potentially novel and powerful architectures for implementation and training
- Perhaps start a Trigger ML forum if there is broad interest?