Track quality machine learning models on FPGAs for the CMS Phase 2 Level 1 trigger

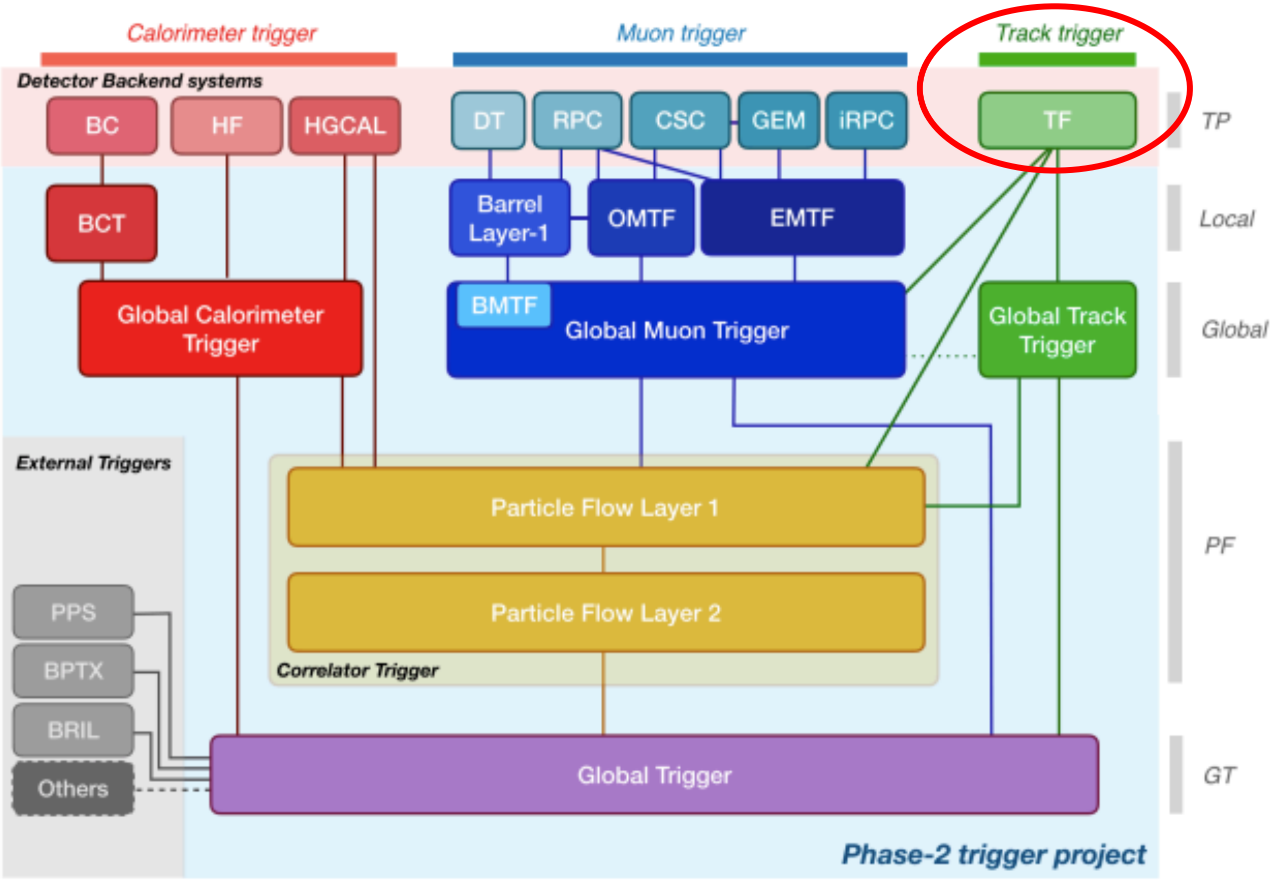
Claire Savard for CMS collaboration

Fast Machine Learning for Science Workshop

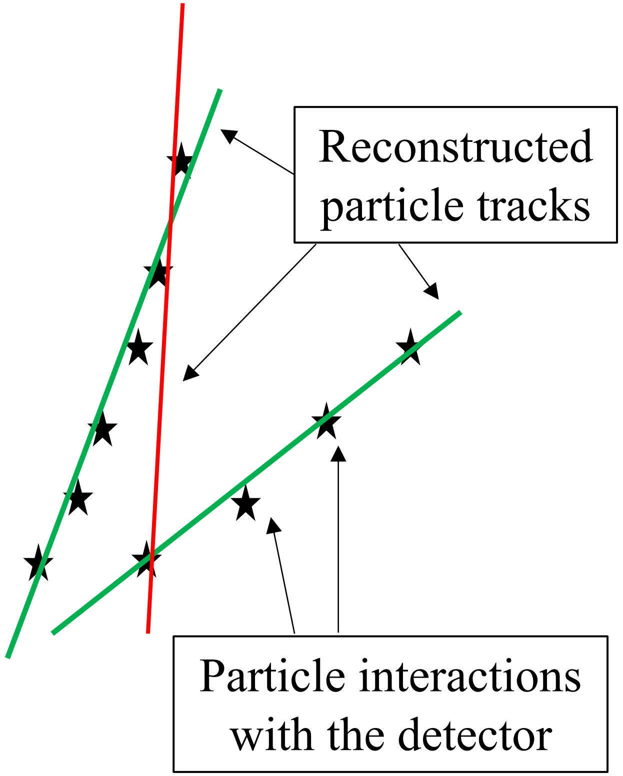
December 1st, 2020

CMS Phase 2 Level 1 trigger

- Increasing luminosity in LHC requires detector upgrades
- CMS Level 1 trigger = initial trigger system for event selection
- Phase 2 includes:
  - Track information available (TF)
  - More powerful FPGAs
  - ➢More sophisticated selection and reconstruction algorithms
- This work: track quality machine learning algorithm on TF



Track quality

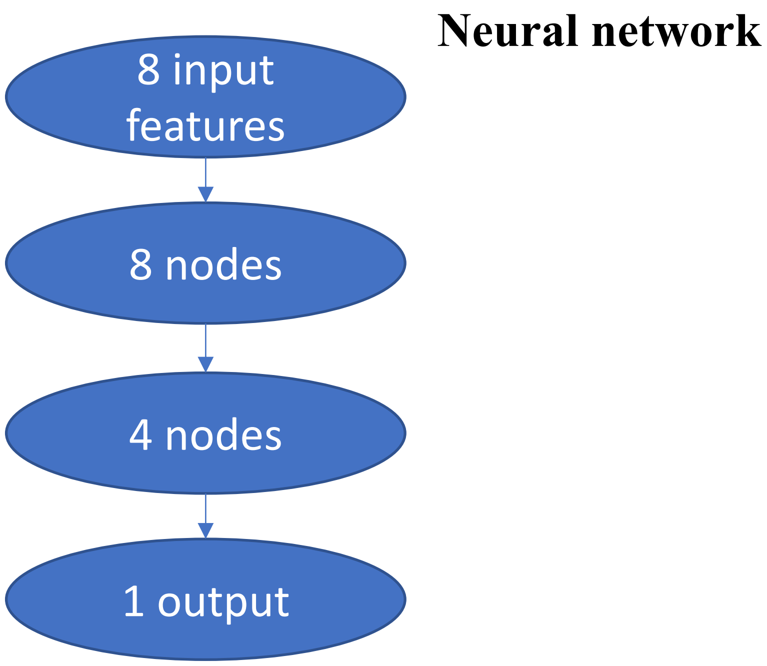


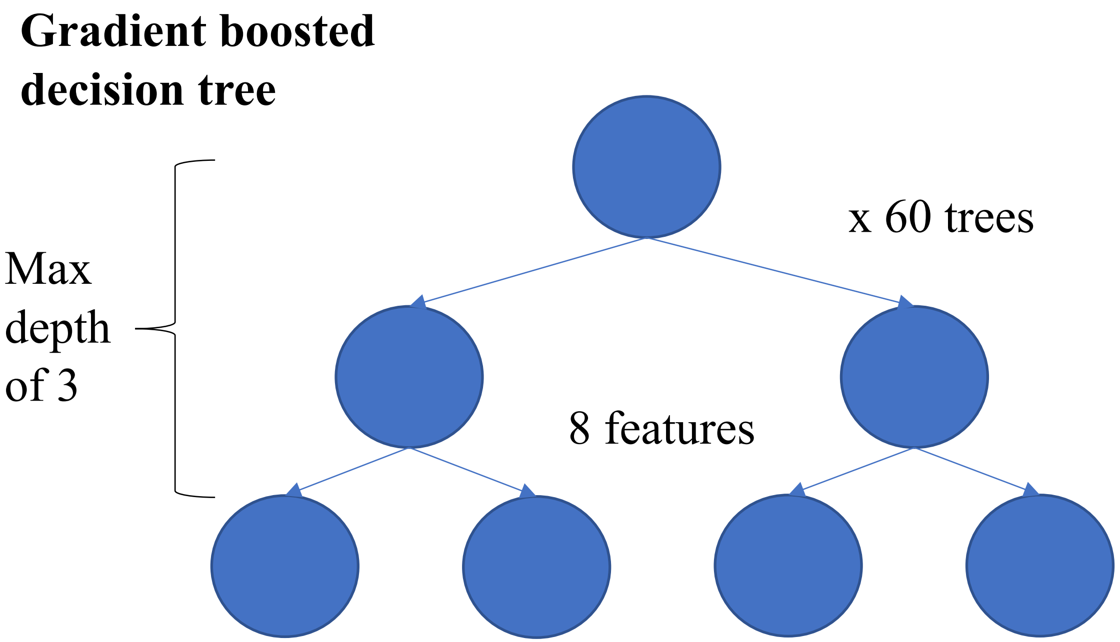
- Track quality = probability of how "real" a track is
  - "real" track is a reconstructed track that originates from an actual particle
  - "fake" track is a reconstructed track created through error in the reconstruction process
- Use track quality to suppress fake tracks
  - Mask real physics occurring

Building track quality classifiers

- Labeling real and fake tracks → classification problem
- Supervised machine learning application
  - Simulated data where labels (real/fake) are known
- Boosted decision trees ([scikit-learn]())
  - Intuitive, continuous + categorical data friendly, easy tuning
  - Similar to applying cuts to variables
- Neural networks ([keras]())
  - Great for complex data, matrix multiplication (FPGA friendly)
  - Increasingly popular in particle physics

Machine learning model stuctures





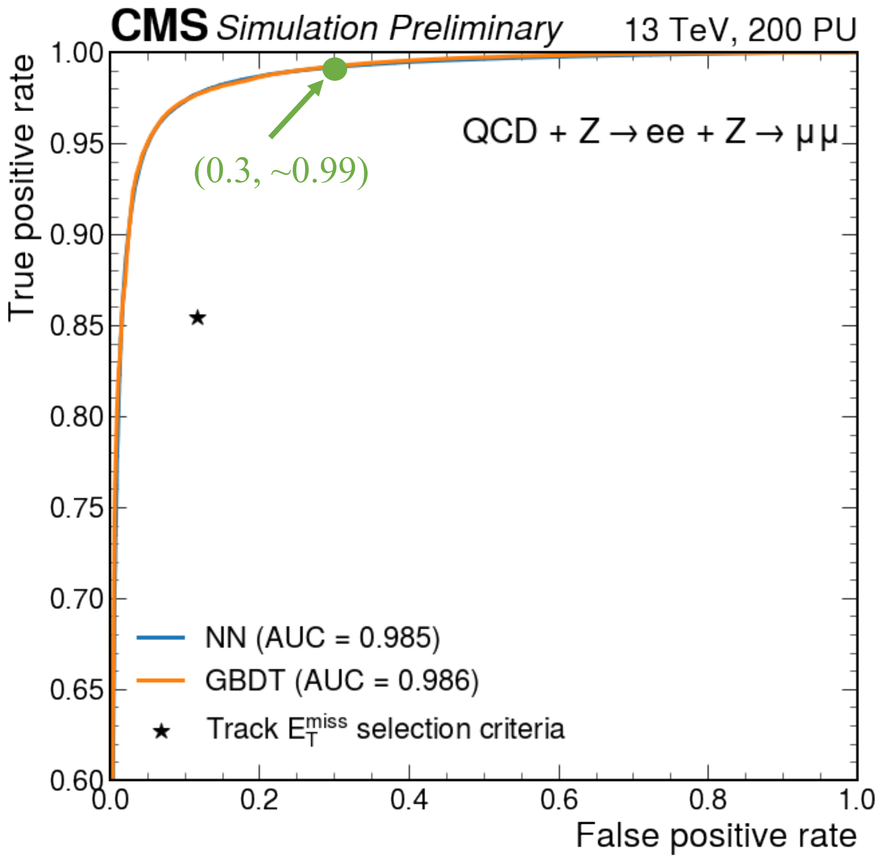
- Use small models to minimize FPGA resource usage
- After pruning and feature selection

Dataset

- Reconstructed tracks from combined QCD + Z→ee + Z→μμ simulated samples + real/fake track label
- Training set
  - 20,000 tracks
  - Contains equal amounts of muons, hadrons, electrons, and fake tracks to emphasize equal importance on each particle type
- Will compare neural network and boosted decision tree to current cut-based method optimized for track E_T^{miss}

Track quality classifier performance

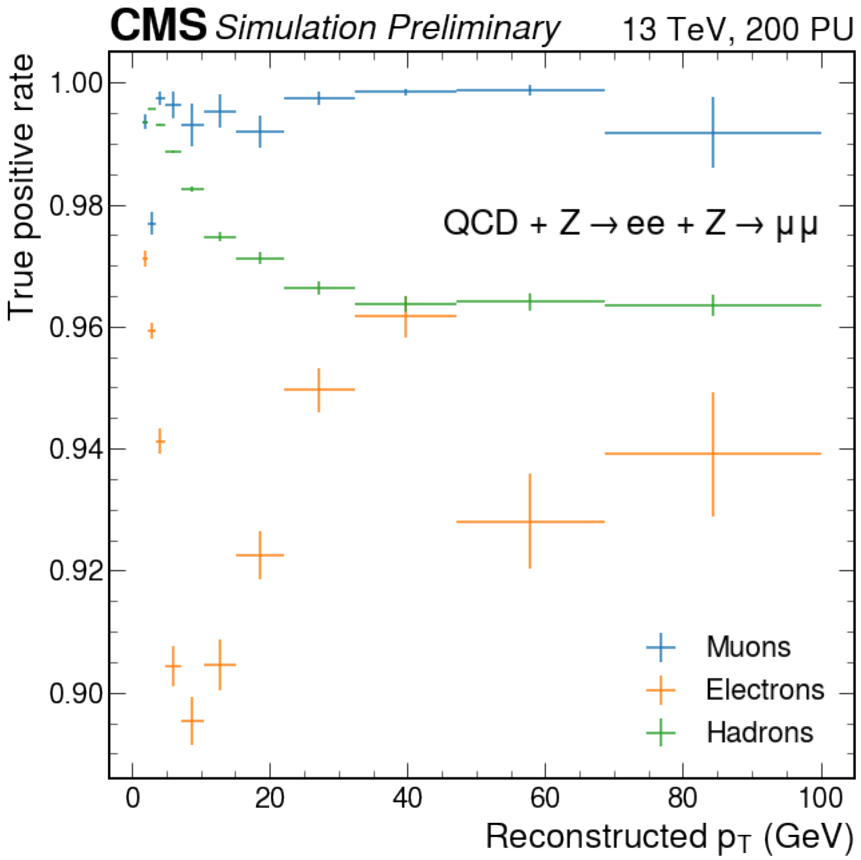
- Machine learning classifiers outperform track E_T^{miss} selection criteria
  - NN = neural network
  - GBDT = gradient boosted decision tree
- NN and GBDT comparable
- Look further at a specific point on curve where false positive rate = 0.3



True positive rate = % correctly identified real tracks
False positive rate = % incorrectly identified fake tracks
AUC = area under curve (accuracy)

Track quality classifier performance

- GBDT results
- Average false positive rate of 0.3
- Efficiency highest for muons, followed by hadrons and electrons
  - Muons tend to be isolated
  - Hadrons undergo nuclear interactions
  - Electrons undergo Bremsstrahlung



True positive rate = % correctly identified real tracks
p_T = transverse momentum

FPGA timing and resource usage

Xilinx VU9P FPGA, 240 MHz clock cycle, initial interval = 1, 10 bit precision (5 bits for int part)

Model	Python AUC	HLS AUC	Latency (clk)	LUT %	FF %	DSP %
NN	0.985	0.982	8	0.104	0.029	0.292
GBDT	0.986	0.981	3	0.140	0.027	0.0

- Project usage from simulated FPGA using HLS
  - Used HLS4ML for synthesizing NN
  - Used Conifer for synthesizing GBDT
- Both models perform the same in accuracy (AUC) and use minimal resources

Updates and future work

- GBDT track quality model included in [Level 1 tracker CMSSW] development branch, working on integration into central CMSSW
- Further testing on physical FPGAs
- Working on electron-specific track quality classifier
- Working on track quality classifier for displaced tracks
  - Displaced = does not originate from proton-proton collision site

Questions?

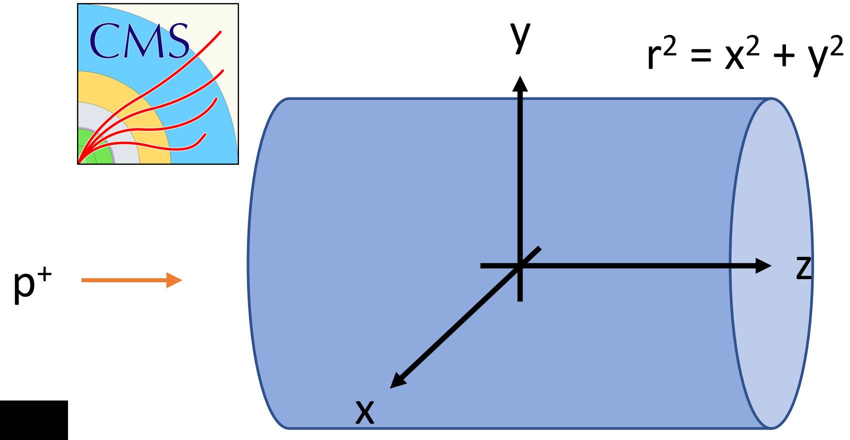
Backup

Track E_T^{miss} selection criteria

- Optimized to lower fake rate (% of fake leftover in sample after selection)

Track variable	Criteria
N_{stubs}	≥ 4
p_T	≥ 2 GeV
χ^2/dof	< 10
χ^2_{bend}	< 2.2

Features



feature	description	range
ϕ	angle in xy-plane	$[-\pi, \pi]$
η	pseudorapidity, describes angle relative to z axis	[-2.4,2.4]
z_0 (cm)	Impact parameter along z	[-30,30]
$nstubs$	number of stubs associated with the track	[4,7]
$nlaymiss_{interior}$	number of layers missed within the sequence of those where stubs were found	[0,5]
χ^2_{bend}	measure of bend consistency of track	[0,inf)
χ^2_{rz}	measure of goodness-of-fit in rz-plane	[0,inf)
$\chi^2_{r\phi}$	measure of goodness-of-fit in rϕ-plane	[0,inf)

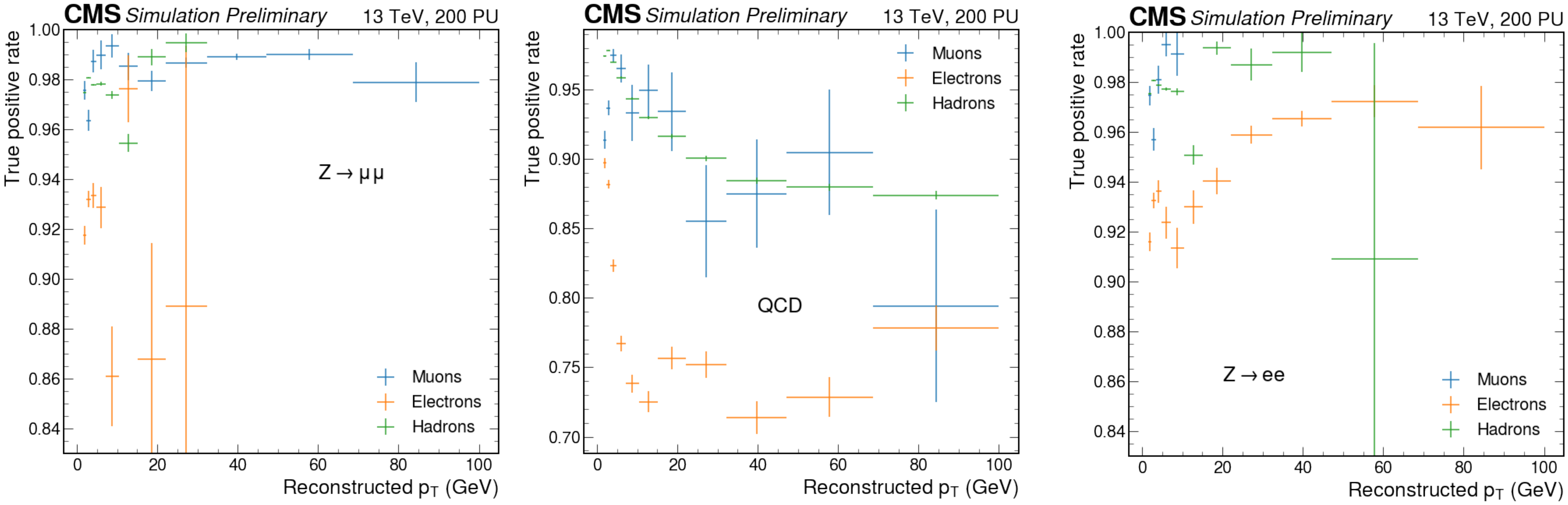
Not good for FPGAs!

Binning χ^2 features

- FPGAs require fixed precision
- Bin the 3 χ^2 variables to only use 3-4 bits
  - Ex: $\chi^2_{bend} = 2.2 \rightarrow$ bin 3

Track variable	Digitization	Num. of bits
χ^2_{bend}	0, 0.5, 1.25, 2, 3, 5, 10, 50, inf	3
χ^2_{rz}	0, 0.25, 0.5, 1, 2, 3, 5, 7, 10, 20, 40, 100, 200, 500, 1000, 3000, inf	4
$\chi^2_{r\phi}$	0, 0.25, 0.5, 1, 2, 3, 5, 7, 10, 20, 40, 100, 200, 500, 1000, 3000, inf	4

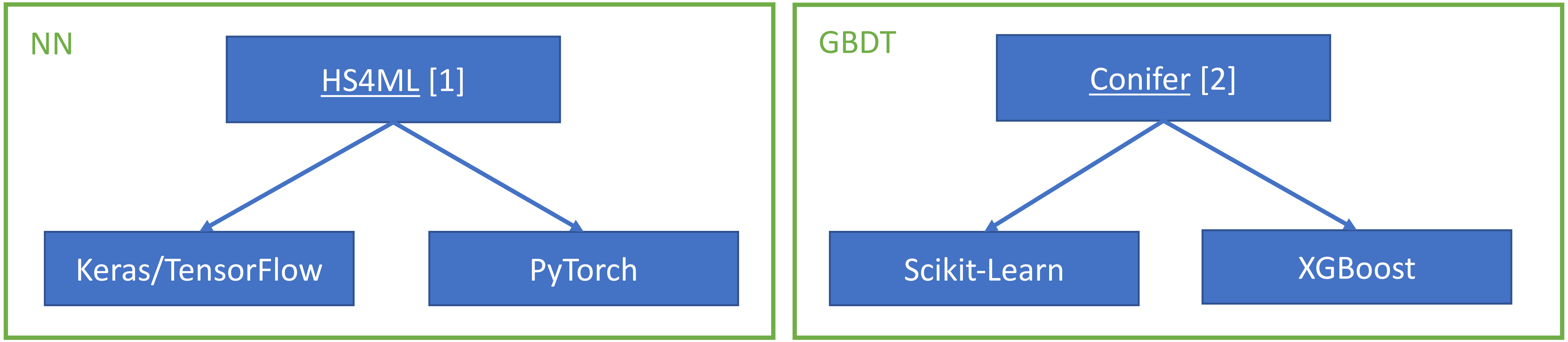
Track quality classifier performance



- GBDT results
- Average false positive rate = 0.3

Track quality on FPGAs

- On FPGA in Phase 2 Level 1 trigger
- Used python packages for machine learning development
- Need to convert python models to FPGA-readable languages



[1] J. Duarte *et al.*, "Fast inference of deep neural networks in FPGAs for particle physics", JINST 13 P07027 (2018)
[2] S. Summers *et al.*, "Fast inference of boosted decision trees in FPGAs for particle physics", JINST 15 P05026 (2020)

Conifer overview (addition to HLS4ML)

- Facilitates conversion of Scikit-Learn/XGBoost decision tree models to FPGA firmware
  - Boosted decision tree
  - Random forest
- Model developed using Xilinx Vivado HLS or at RTL using VHDL
- Lightweight models, uses mostly LUTs, no DSPs
  - HLS4ML limited by DSP usage