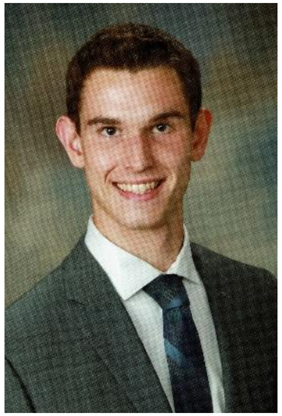
Nanosecond machine learning with BDT for high energy physics

University of Pittsburgh



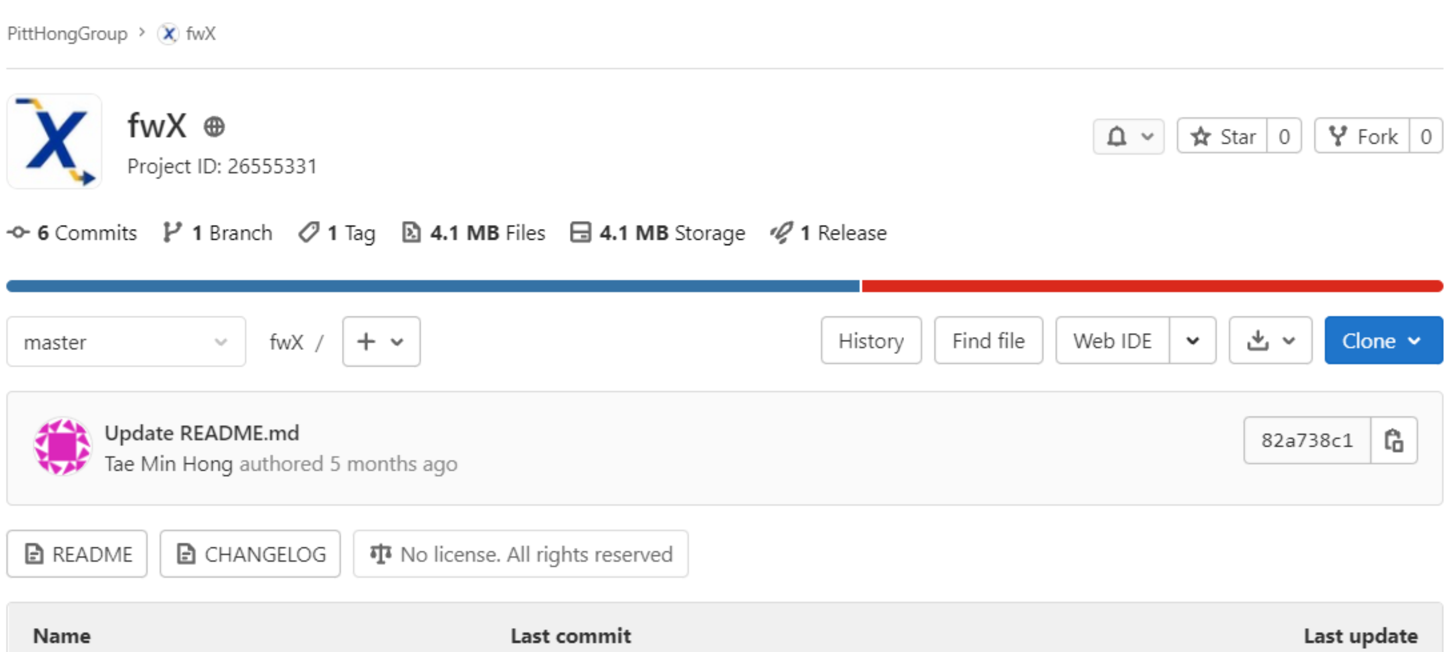
FW X Machina

Stephen Roche*
Tae Min Hong
On behalf of all authors of JINST 16 P08016 (2021)

ICALEPCS
October 21, 2021
https://whova.com/portal/webapp/icale_202110/Agenda/1923884

fwX is a tool that puts boosted decision trees on FPGAs for ultra-low latency real-time evaluation using minimal resources

- Published JINST 16 P08016 (2021)
- Preprint available on the arXiv: 2104.03408 [hep-ex]
- Tutorial, links, and FAQ at: fwX.pitt.edu
- Git located at: https://gitlab.com/PittHongGroup/fwX

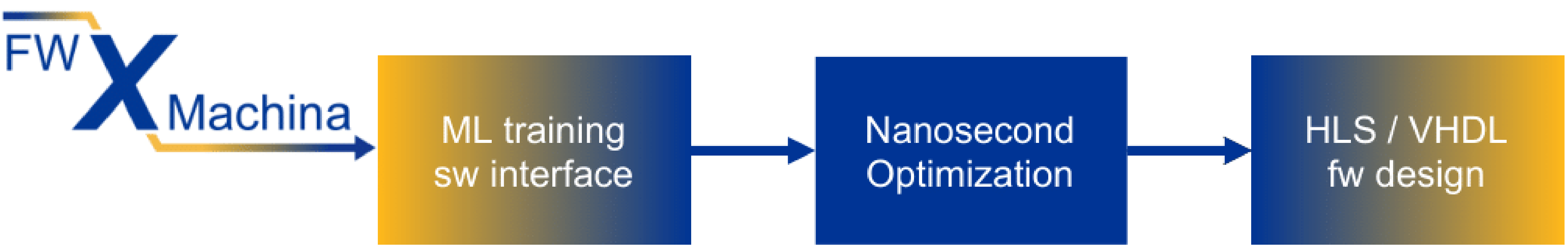


Overview

Steve Roche

Four Sections

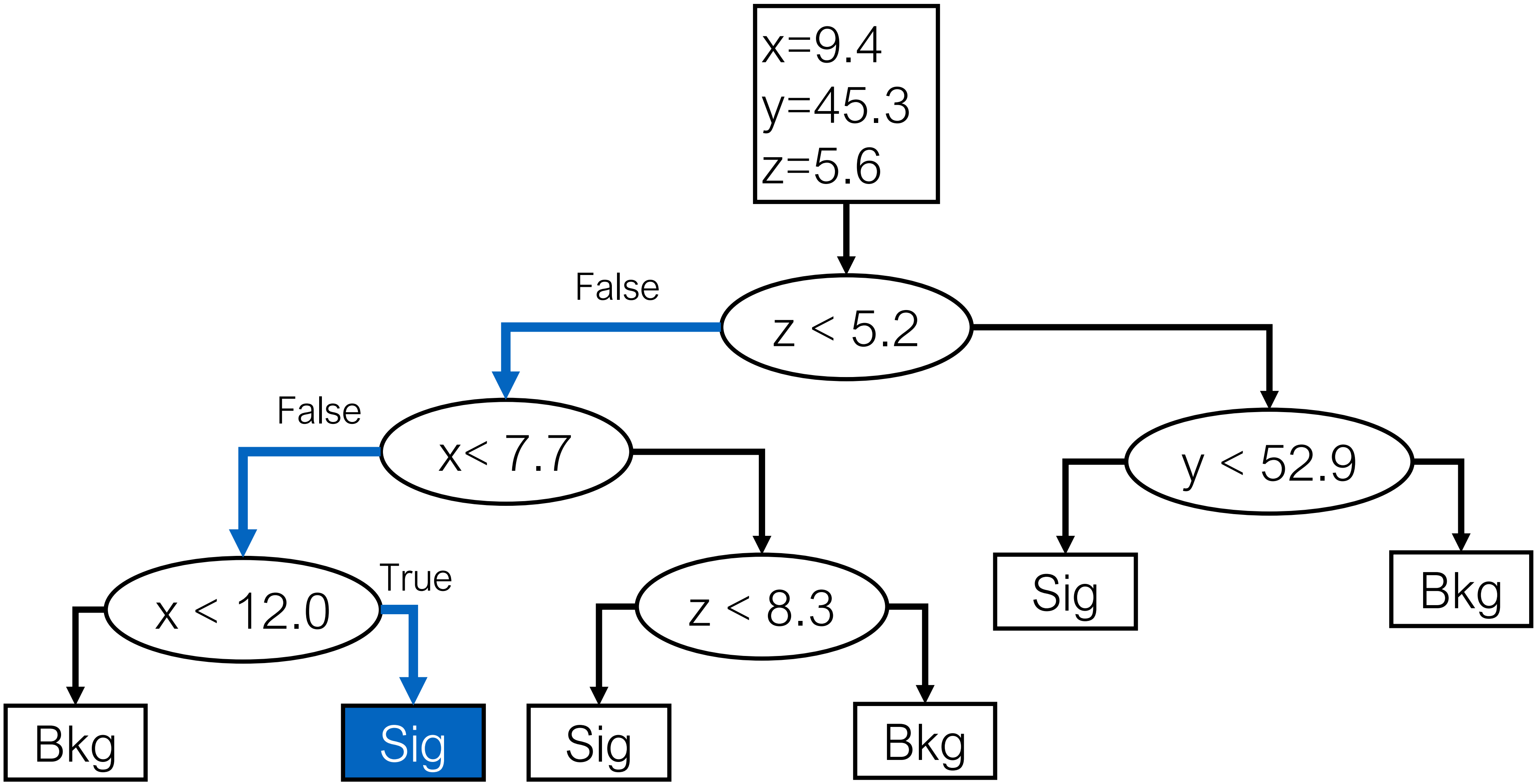
- Boosted decision tree introduction
- Motivation for putting BDTs on FPGAs
- Optimization
- Example problem



Decision Trees

Machine learning method that separates classes (i.e., signal versus background) by cutting on variables

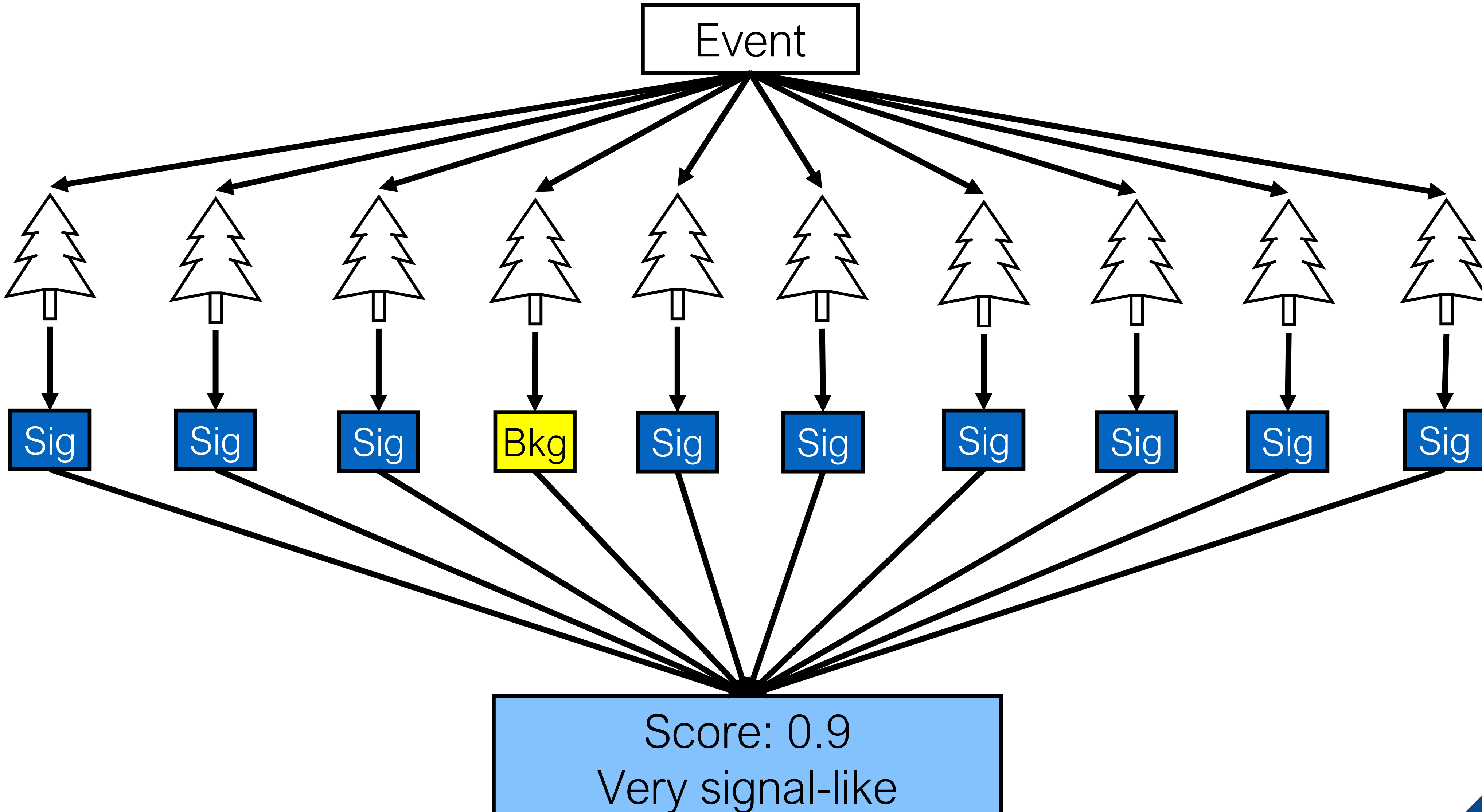
- Example: this 3-variable event is classified as signal



Decision Forests

Many trees are trained independently and given weights (boosting)

- Event classified as **weighted average score** of all these trees

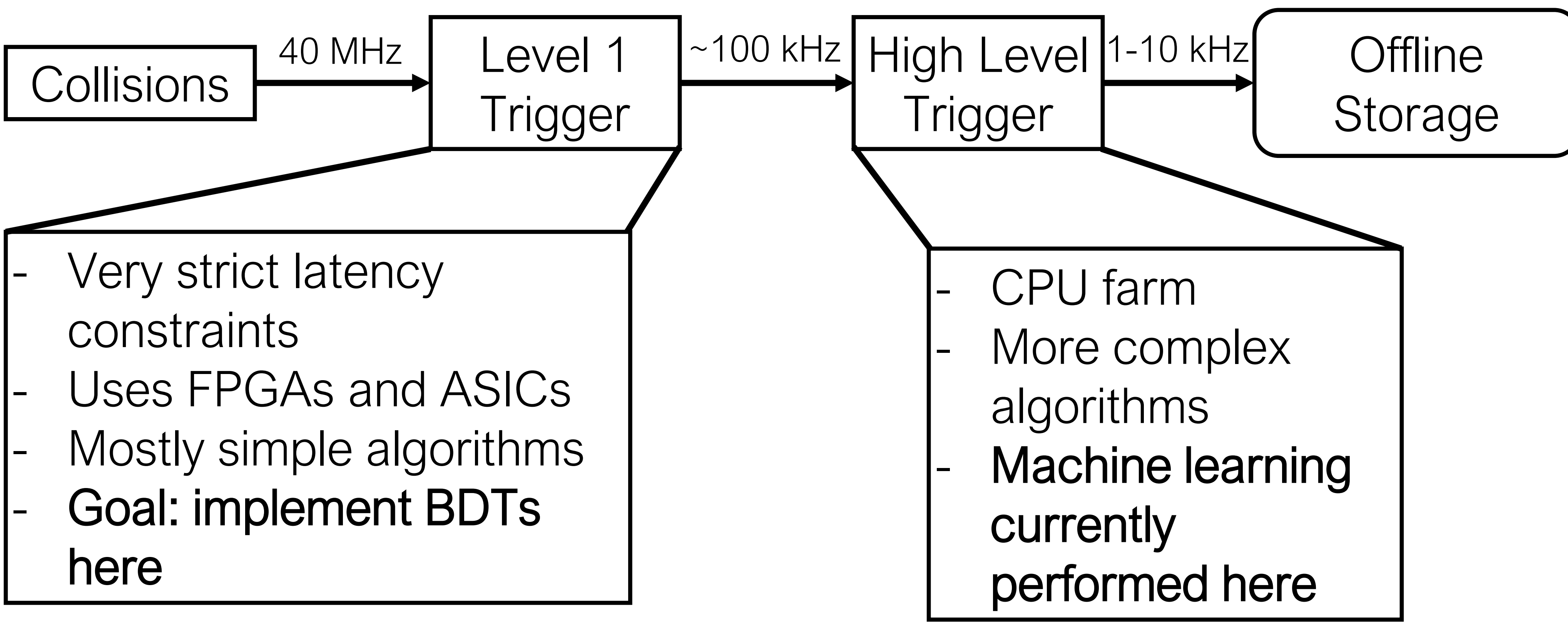


Motivation

Steve Roche

Triggering at hadron colliders

- Two-level trigger systems at detectors such as ATLAS and CMS
- Level 1 trigger (L1) cuts down 40 MHz bunch-crossing rate to ~100 kHz using custom electronics
- High level trigger (HLT) reduces rate to 1-10 kHz using CPU farm



fwX Overview

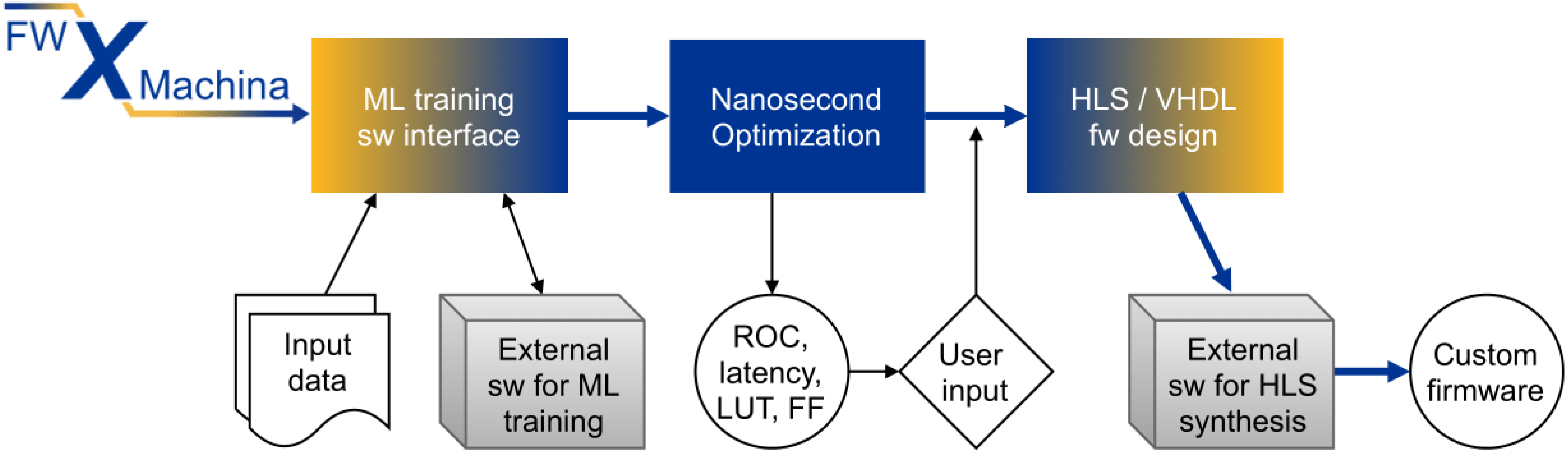
Steve Roche

Do once:

- Train BDT in software
- **Process trained forest with fwX, this optimizes for FPGA**
- Load synthesized firmware on FPGA

In real time:

- Use FPGA to evaluate/make inference of incoming events

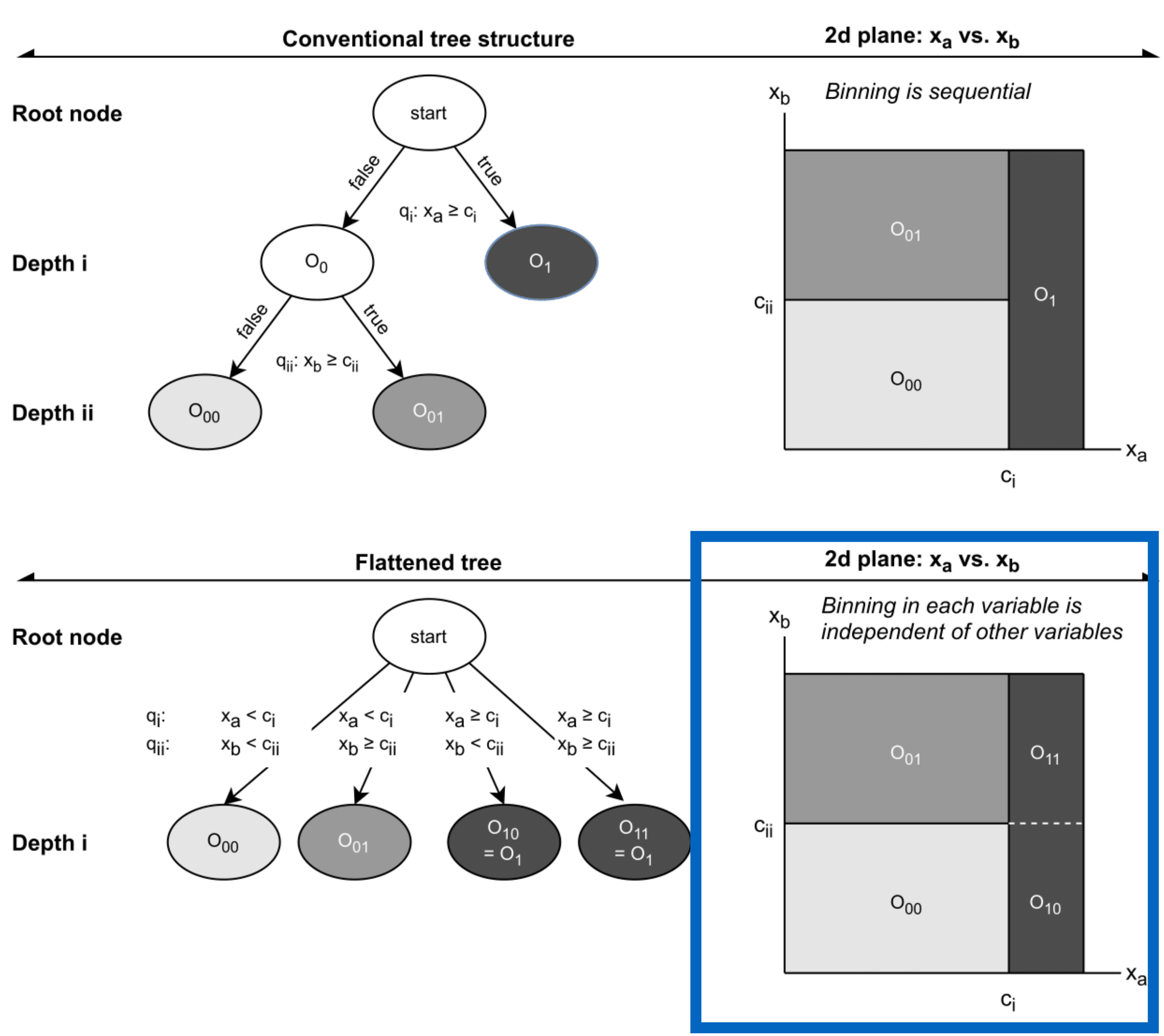


Tree Flattening

Steve Roche

Novel approach: "flatten" tree structure into grid

- Converts recursive problem into binning problem
- FPGA bins each variable independently in each direction
- Allows design to take advantage of parallelization
- Final bin location determines BDT score



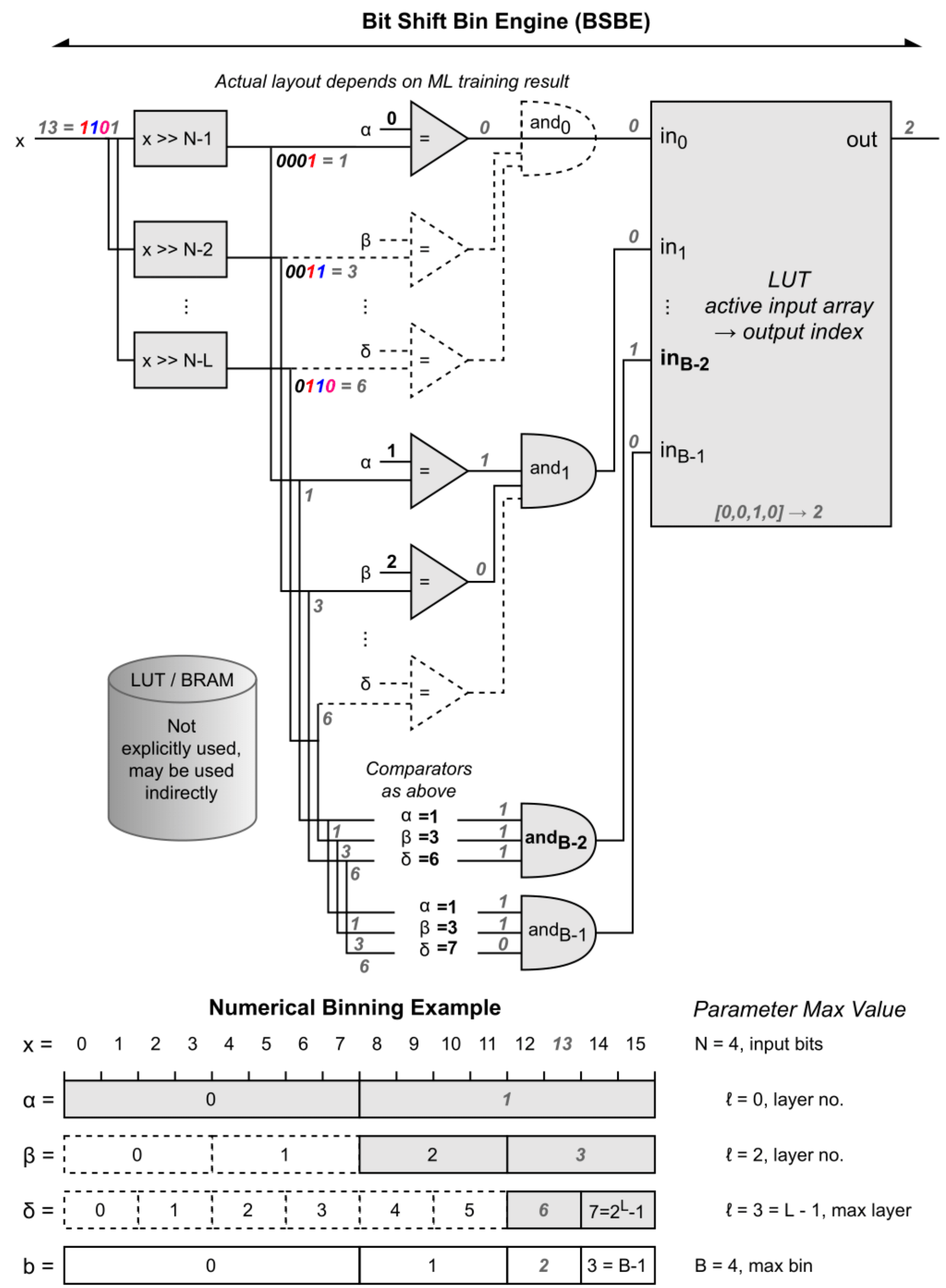
What is implemented on FPGA

Bit shift bin engine

Steve Roche

Novel approach: localize data by bit-shifting

- Layers of bit-shifting to approximate bin location
- All combinatoric logic
- Final bin location determines BDT score

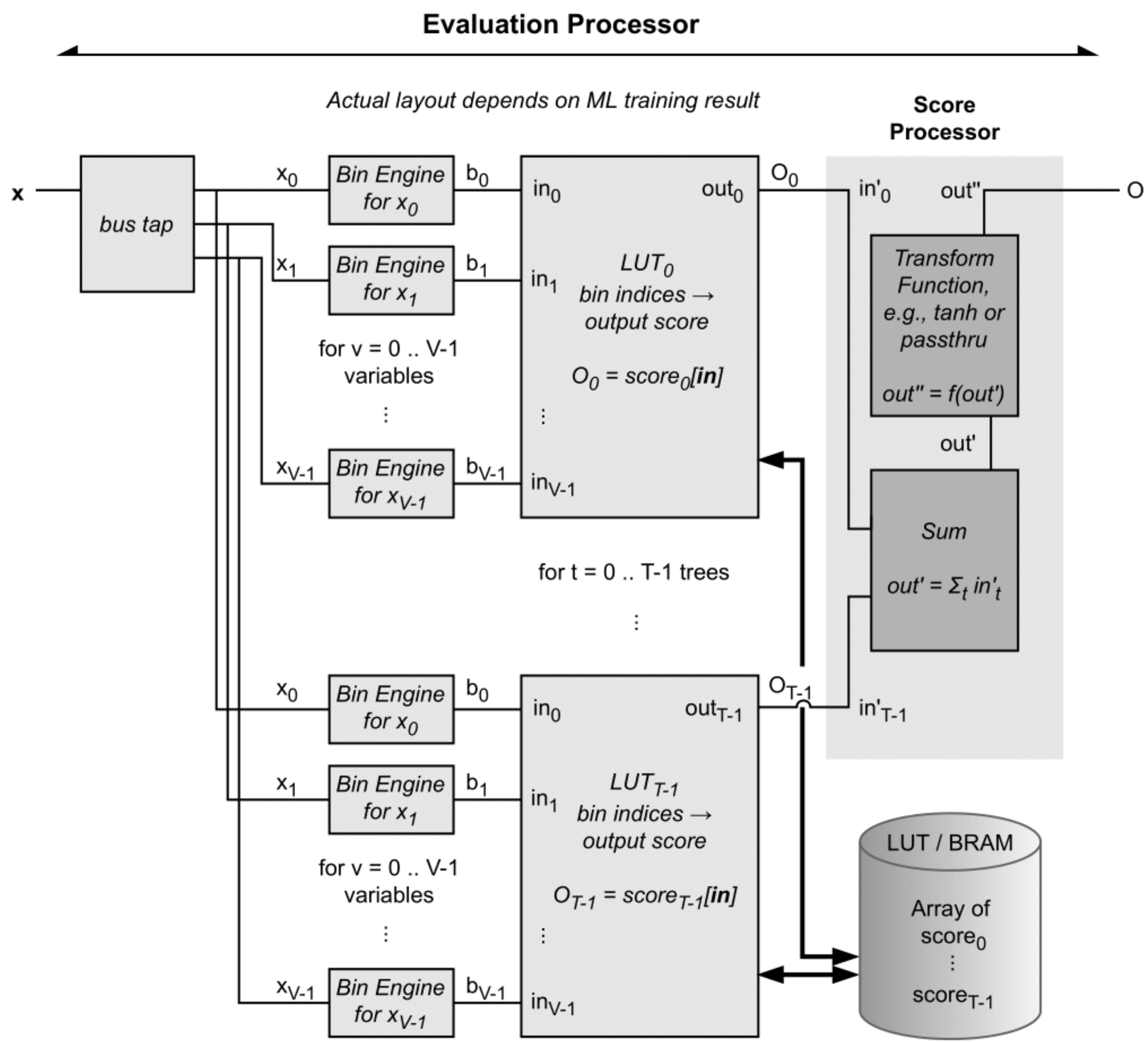


Firmware design

Steve Roche

Novel approach: parallelize input variables and trees

- Bin engines used to bin each variable in parallel
- Look-up table used to find score for resulting bin
- Trees processed in parallel
- Scores for each tree combined



Other optimizations

Steve Roche

Tree merging

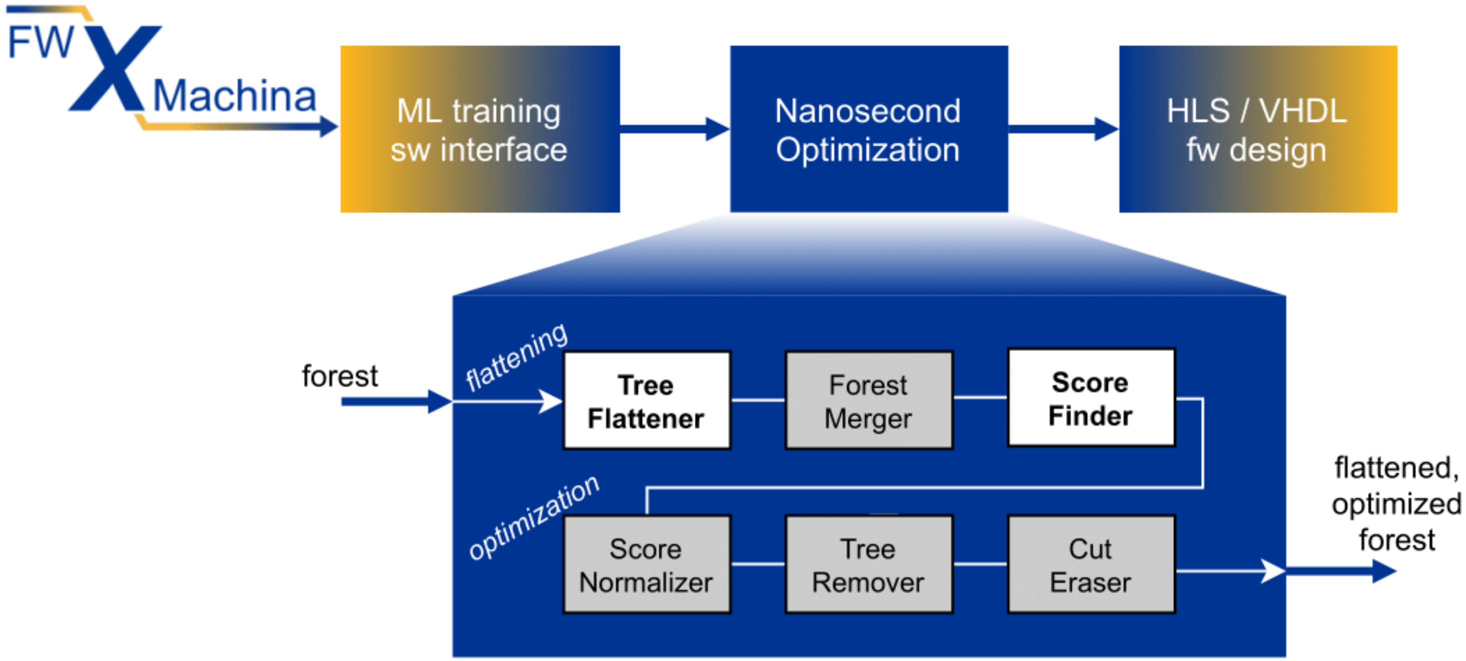
- Can merge trees by summing grids

Integer precision

- Can use floating point values or convert to integer precision to speed up evaluation

Other optimizations

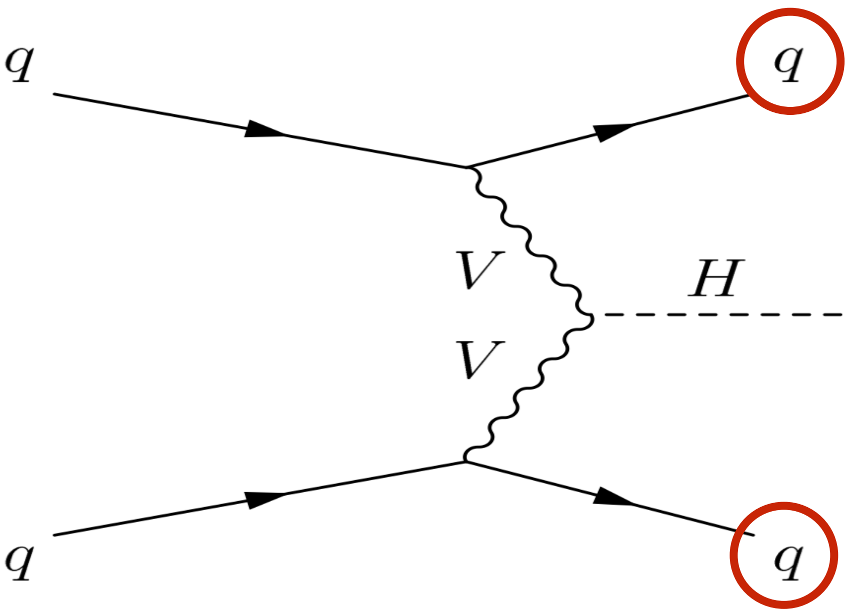
- Remove trees that don’t have an impact due to low boost-weight
- Remove cuts that are redundant due to merging



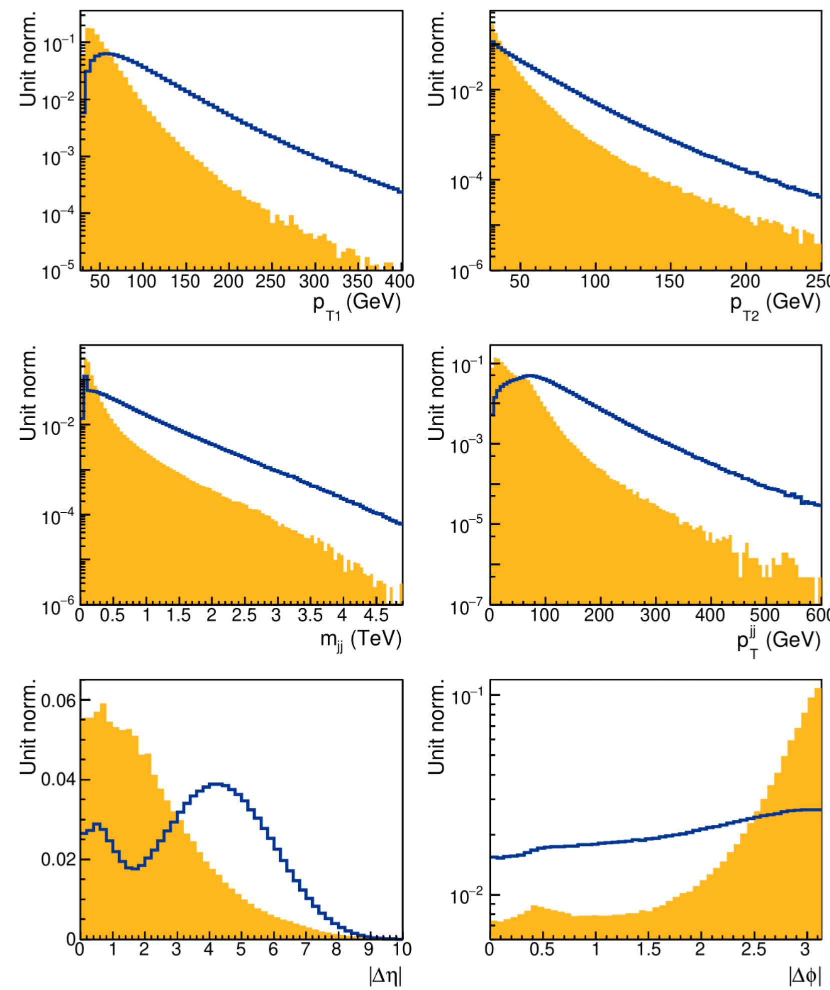
Example use case: Higgs identification

Goal: identify Higgs production in vector boson fusion (VBF)

- Strategy: identify events with a VBF jet pair
- Reject background



Attributes of VBF jet pair look different than QCD jets



Samples

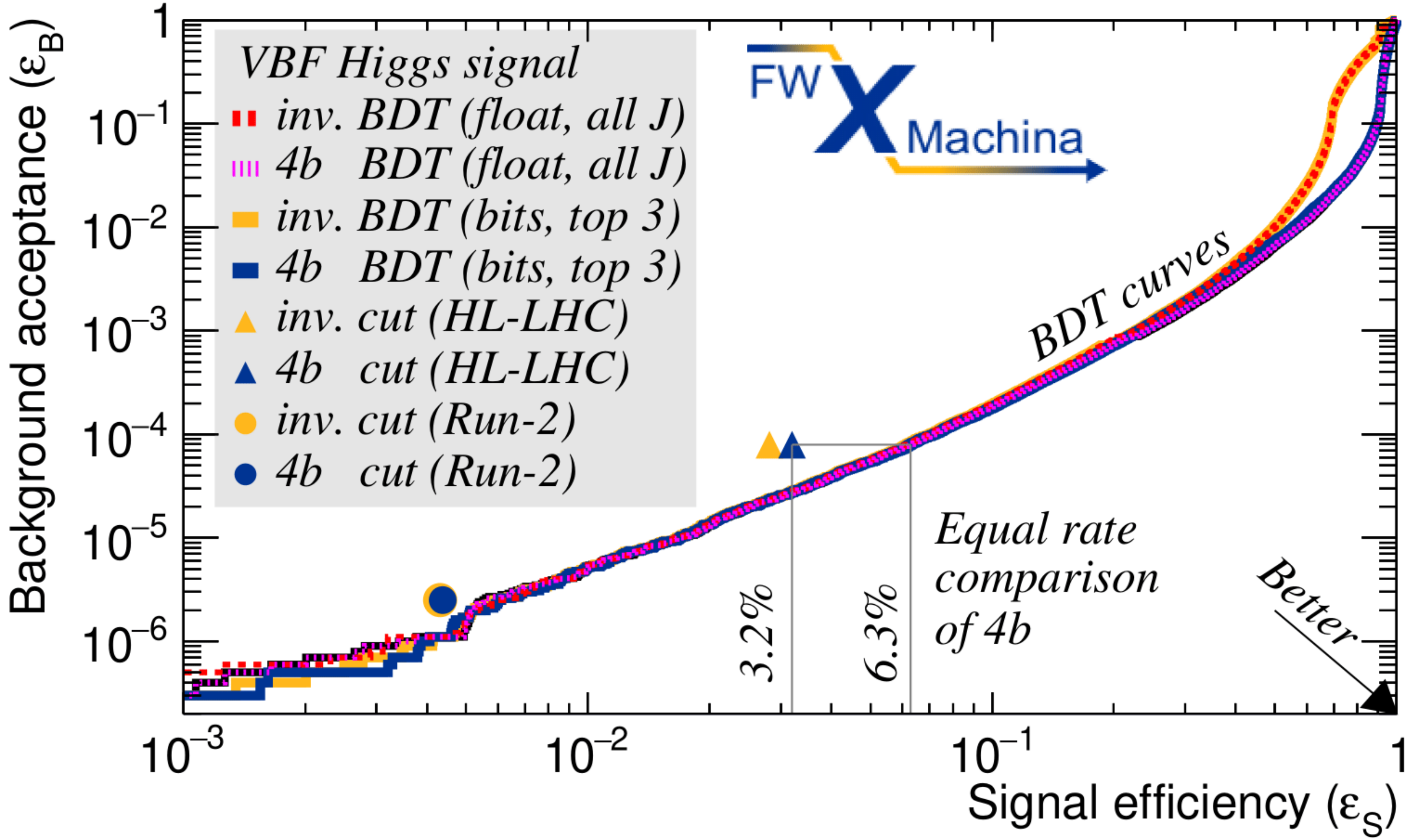
- *VBF Higgs*
- *Multijet*

Notes

- *Unit norm.*
- *Inputs for BDT*
- *Generated with MadGraph5 + Delphes smearing*

Physics performance

- At same background rejection, accepts **twice as many** VBF events as our approximation of benchmark cut-based trigger



Example use case: Higgs identification

Firmware performance

- Very low latency and minimal resource usage

Quantity	Value
Latency	5 clock cycles (15.6 ns)
Interval	1 clock cycle (3.125 ns)
LUT	1.0%
FF	< 0.1%
BRAM	2.3%
URAM	0
DSP	0

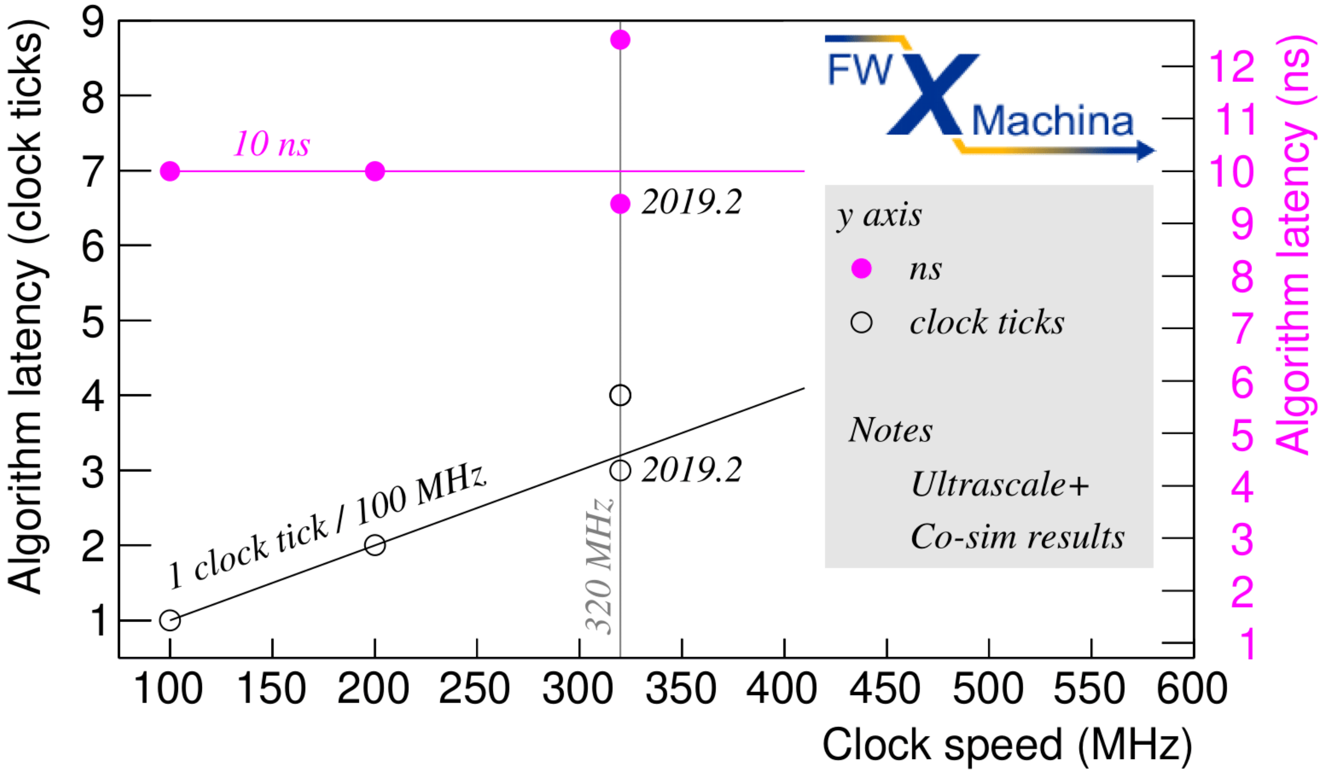
Latency and interval dependence

Latency depends on problem

- Scales with number of trees, depth of trees, and number of input variables
- For “reasonable” problems we can get **very low latencies** (< 10 ns)

Latency is independent of clock speed

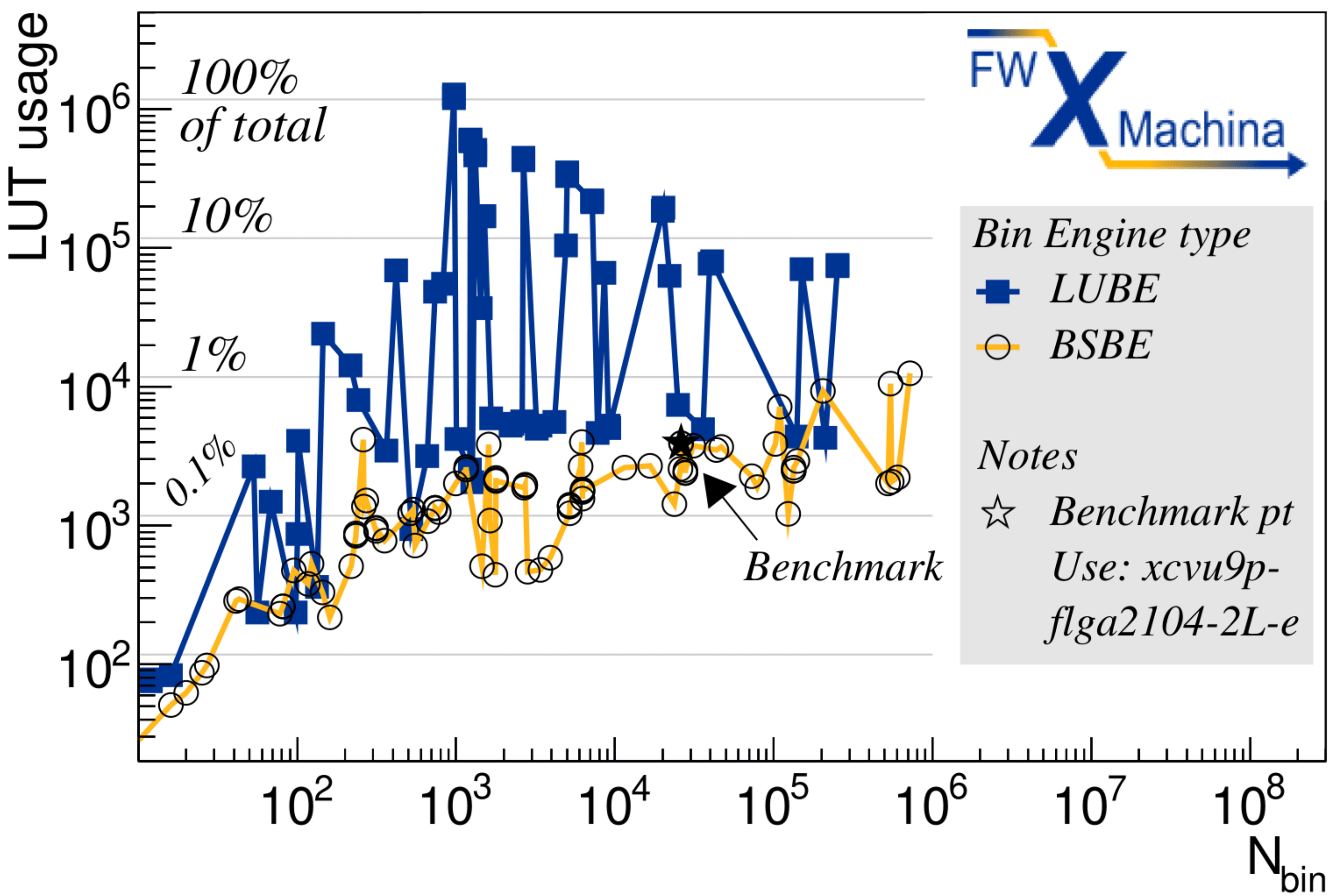
- None of our operations are clocked



Resource usage dependence

Also depends on problem

- Depends on same things as before, along with integer precision used
- For "reasonable" problems we can get **very low resource usage** (< 1%)



Conclusions and review

Steve Roche

fwXmachina can implement BDT classifiers on FPGAs

- Novel approach to decision tree evaluation optimized for firmware
- Can provide low latency and minimal resource usage for some realistic physics problems
- Potential to allow for machine learning in FPGA-based level 1 trigger systems

Vector boson fusion Higgs production demonstrates example test case demonstrating reasonable problem

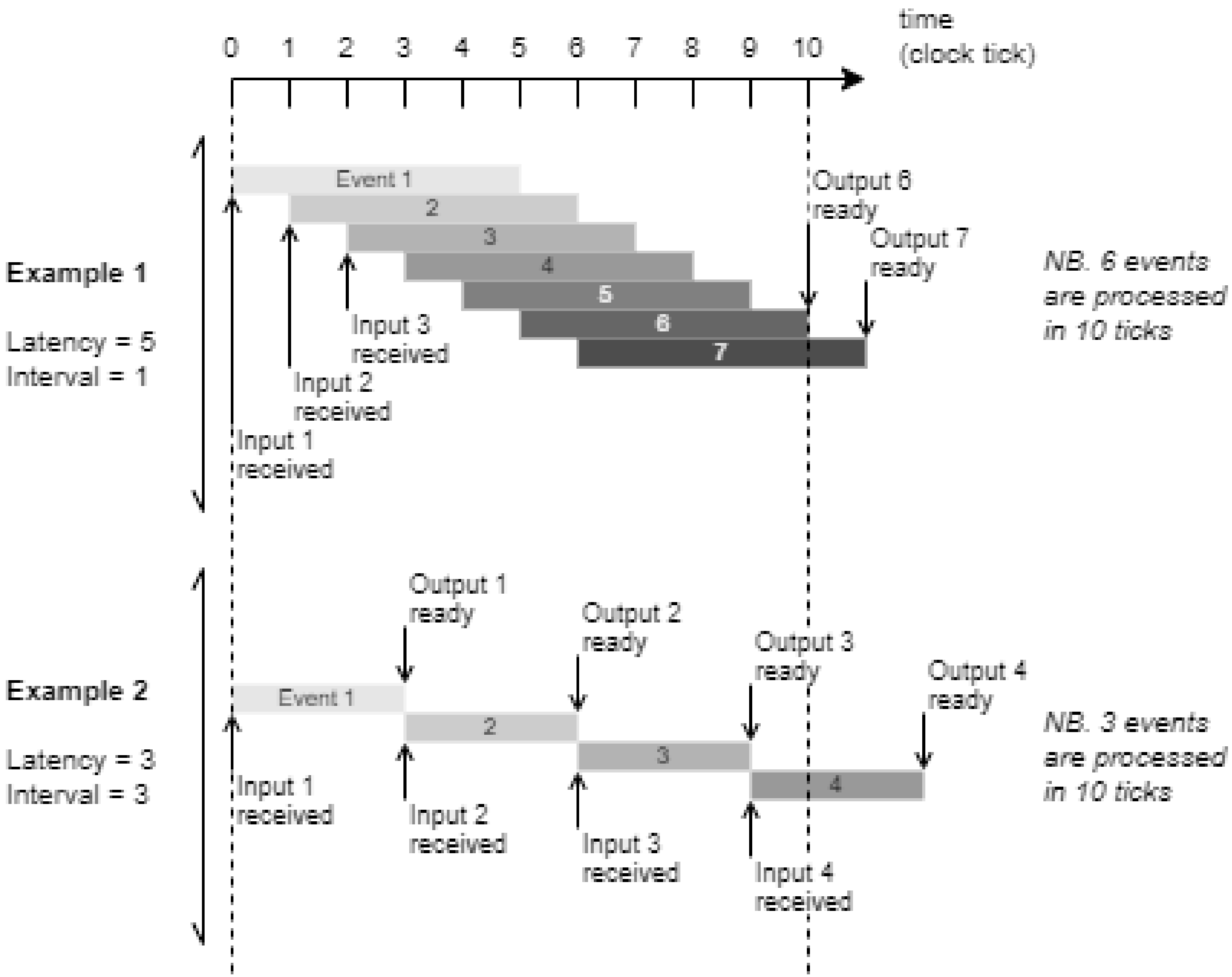
- Representative of problems that L1 trigger systems solve
- Shows that BDTs may be better than cut-based in some cases
- fwX can provide firmware evaluation within the latency and resource constraints of L1 triggers

Backup: latency vs interval

Steve Roche

Latency- how long it takes to unpack the first box you put on the conveyor belt

Interval- how long it takes to put the next box on the conveyor belt



Backup: Currently supported

Software package

- Python version 3.x
- Tested on Linux machines, should work on other OS systems as long as ROOT is supported

Machine learning

- Binary classification with TMVA
- BDT
- Cut-based (not discussed in this talk)

Firmware

- Xilinx Vivado HLS

Backup: fwX vs hls4ml/Conifer

Steve Roche

Another group does something similar

- We compare our results to theirs using same BDT forest
- Problem: electron vs photon in simulated calorimeter (see paper)

Parameter	fwXmachina	hls4ml-Conifer	Comments
ML training setup			
Training software	*TMVA*	*TMVA*	same
Physics problem	electron vs. photon	electron vs. photon	same
Training samples	from ref. [56]	from ref. [56]	same
No. of event classes	2	2	same
No. of training trees	100	100	same
Max. depth	4	4	same
No. of input variables	4	4	See figure 18
Other *TMVA* parameters	*TMVA* defaults	*TMVA* defaults	same
Nanosec. Optimization	Flattened & merged to 10 final trees, without Tree Remover or Cut Eraser	N/A	Unique to fwX
FPGA and firmware setup			
Chip family	Xilinx Virtex Ultrascale+	Xilinx Virtex Ultrascale+	same
Chip model	xcvu9p-flga2104-2L-e	xcvu9p-flga2104-2L-e	same
Vivado HLS version	2019.2	2019.2	same
Clock speed, period	320 MHz, 3.125 ns	320 MHz, 3.125 ns	same
Precision	ap_int⟨8⟩	ap_ufixed⟨10, 5⟩	See text
Bin Engine	BSBE	N/A	Unique to fwX
FPGA cost			
Latency	3 clock ticks, 9.375 ns	15 clock ticks, 46.875 ns	-
Interval	1 clock tick, 3.125 ns	1 clock tick, 3.125 ns	same
LUT	1903, $<0.2\%$ of total	2.3 M, 192% of total	See caption
FF	138, $<0.01\%$ of total	1.1 M, 44% of total	-
BRAM 18k	8, $<0.2\%$ of total	0	-
URAM	0	0	same
DSP	0	0	same

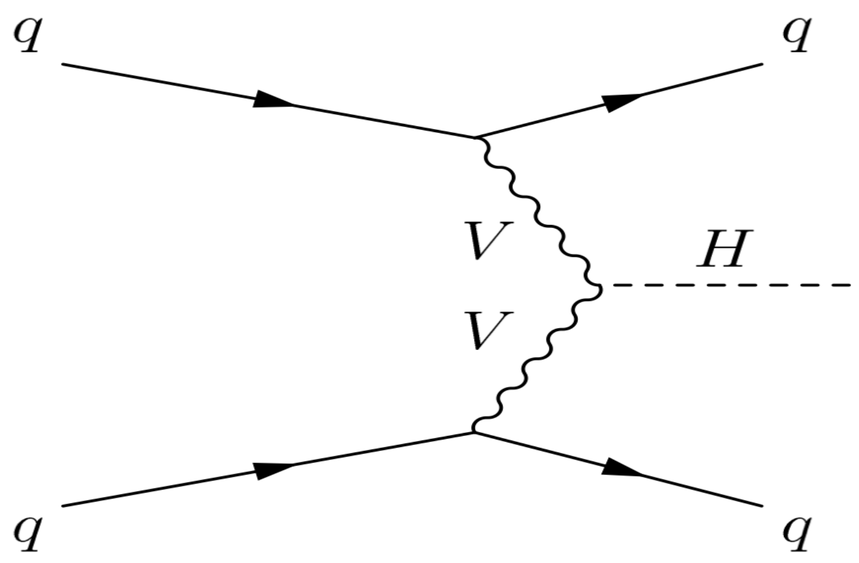
Backup: VBF identification problem

Vector boson fusion (VBF) Higgs production

- VBF is a known Higgs production mechanism
- Currently triggers exist that identify possible VBF events at Level 1

Two questions:

- Is it possible to improve existing L1 VBF triggers by using boosted decision trees?
- If so, can fwX be used to evaluate them on FPGAs under the strict timing and resource constraints necessary at L1?



Backup: VBF identification strategy

Steve Roche

Background

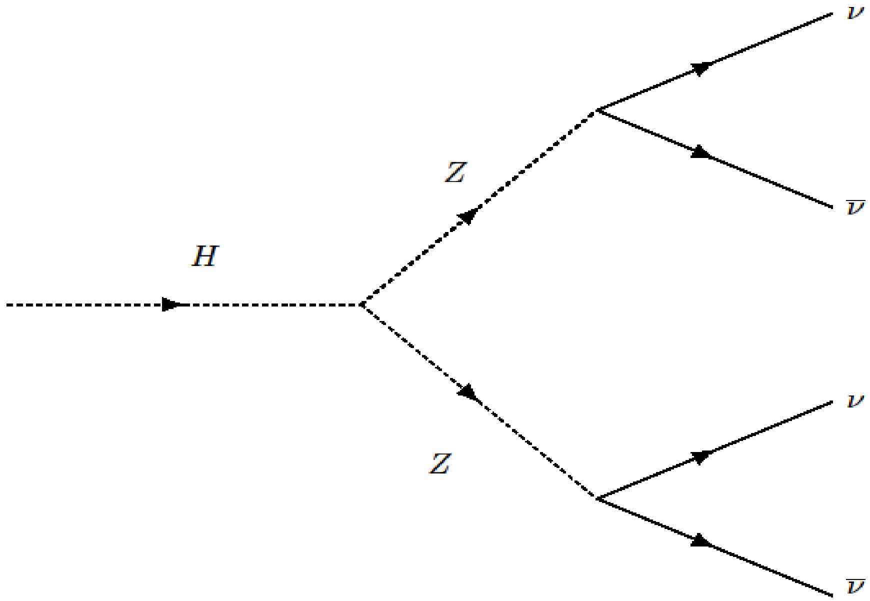
- QCD multijet

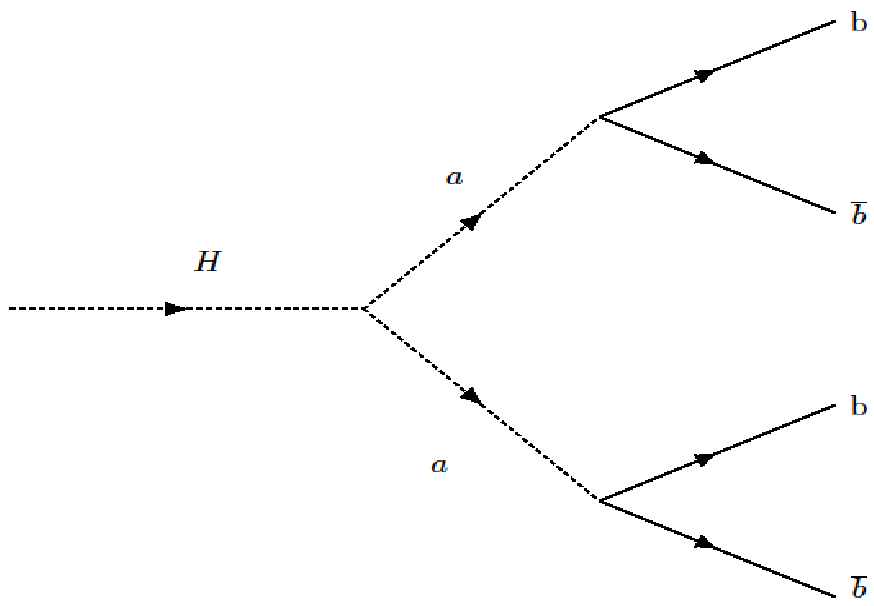
Training signal

- Higgs decay to neutrinos
- Neutrinos invisible to detector
- Only jets seen come from VBF pair

Testing signal

- Higgs decay to 4 bottom quarks
- Tests that classifier identifies VBF pair without getting confused by other jets

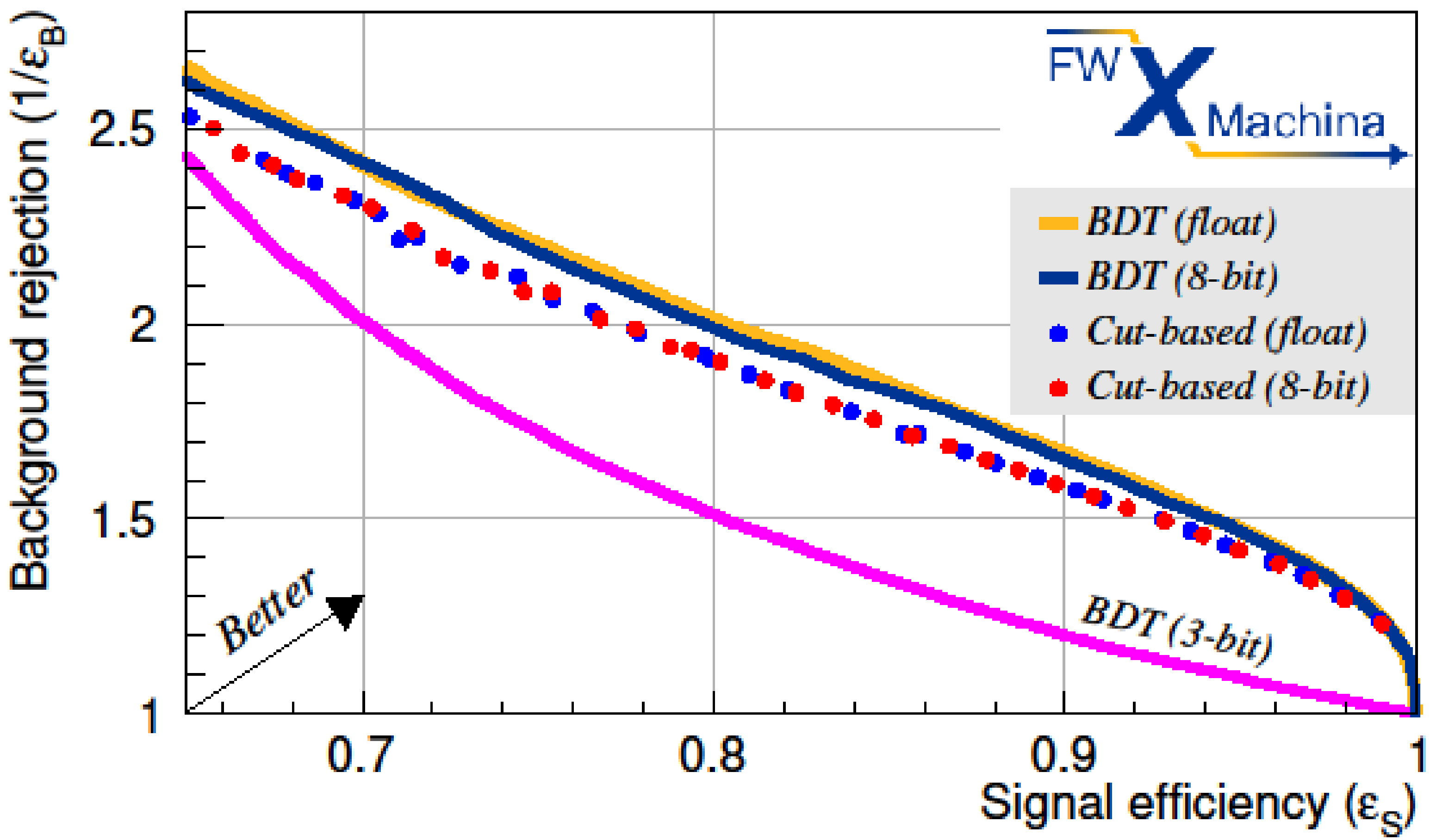




Backup: Classifier performance

For high enough integer precision, little to no loss in classifier performance

- Definition of "high enough" depends on problem
- In problem shown below, 8 bits is enough



Backup: Motivation and other literature

Steve Roche

Goal: improve performance of Level 1 trigger systems

- Evaluate machine learning algorithms on customizable electronics
- Flexible packages to implement neural networks on FPGA exist (1804.06913, 2003.06308, 2101.05108, 2103.05579)
- Flexible package to implement BDTs on FPGAs exist (2002.02534)
- **Our contribution: novel optimizations to implement boosted decision trees on FPGAs**