FWXMachina tutorial

Tae Min Hong

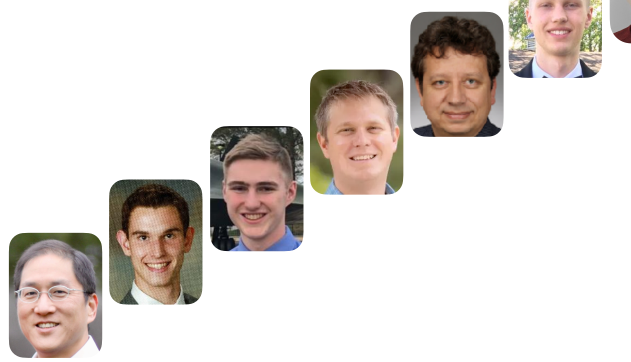
University of Pittsburgh

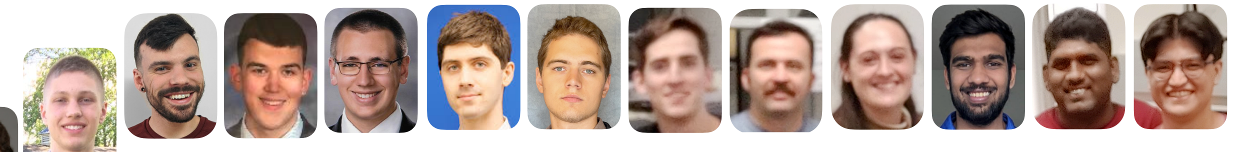
SMARTHEP Edge ML School

September 24, 2024

https://indico.cern.ch/event/1405026/contributions/6103378/

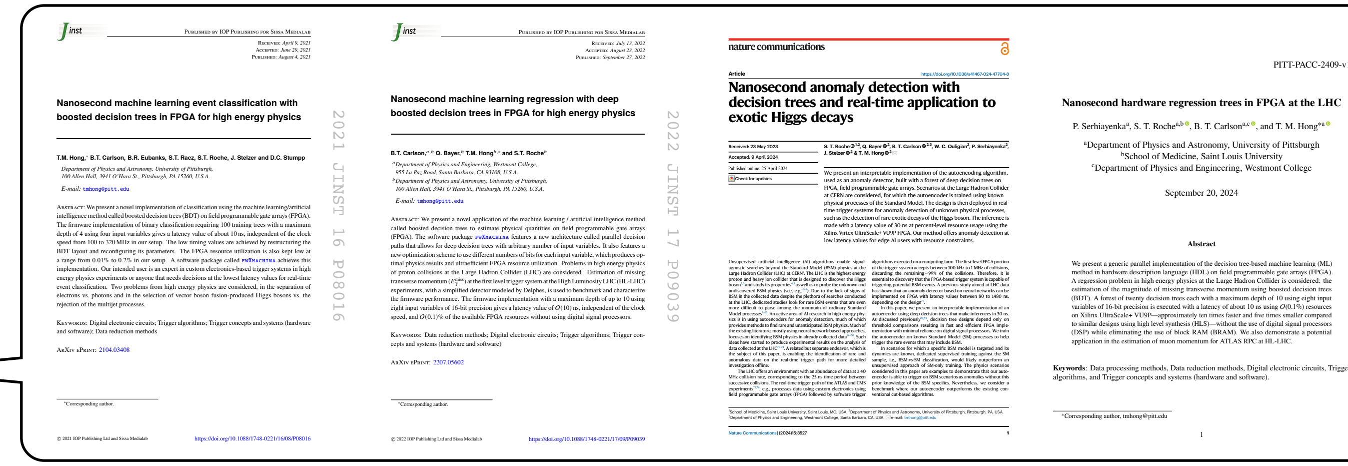
Outline

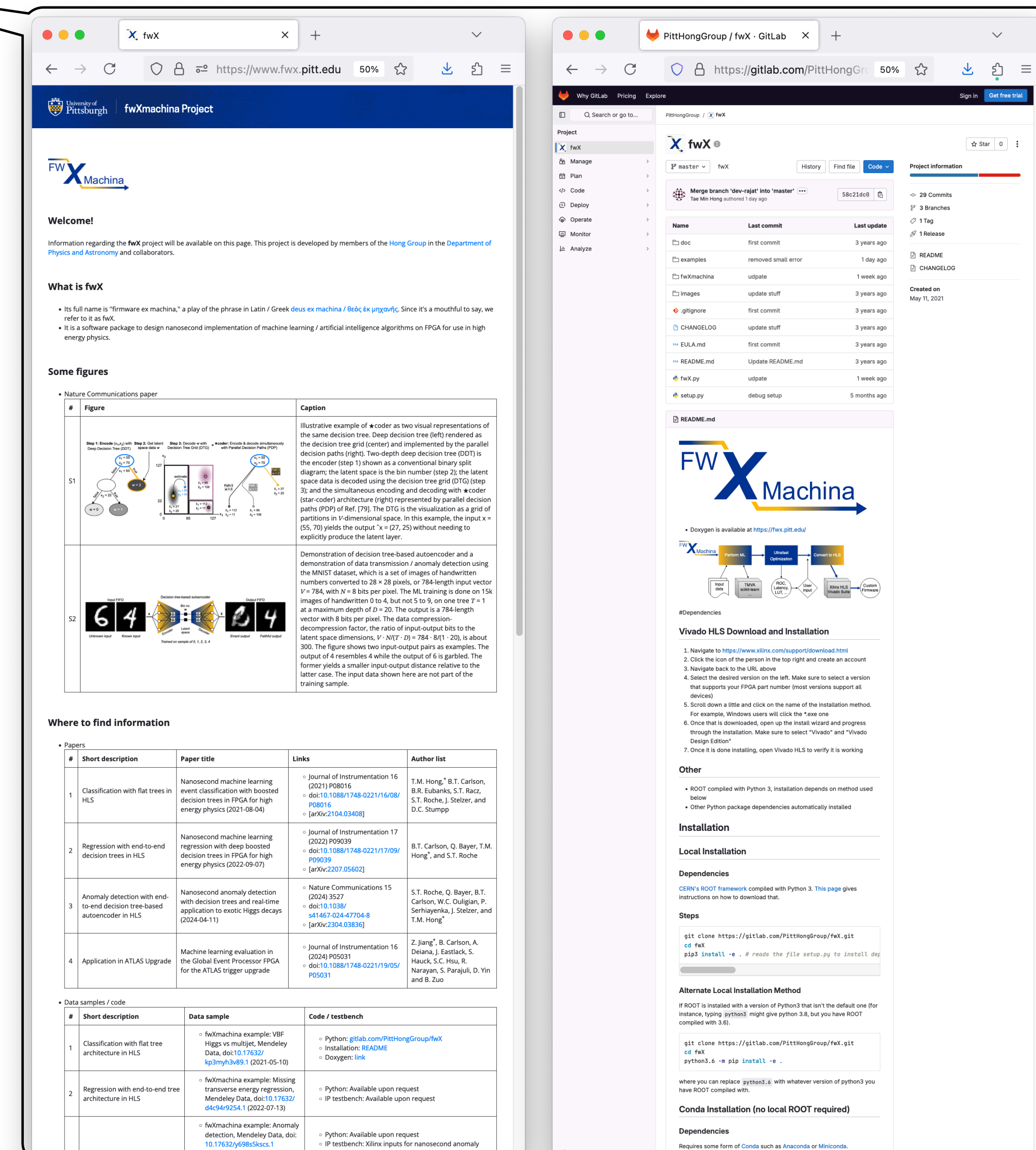




Introduction

- Who are we
- What products do we offer
- Summary if you have to leave





Theory

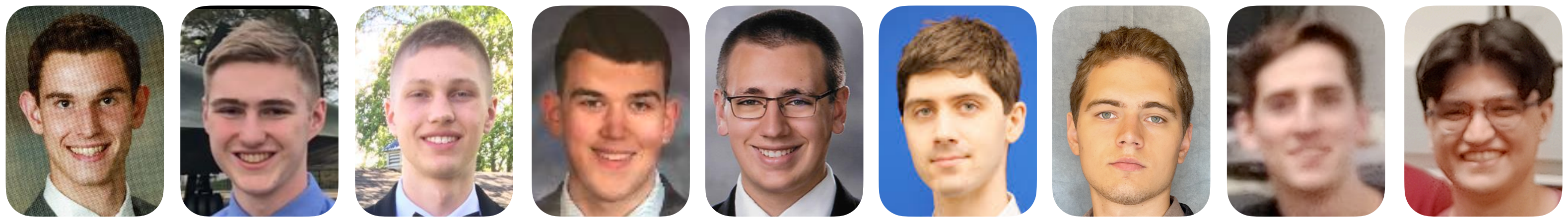
- Parallelizing decision trees
- Classification & regression
- Anomaly detection w/ autoencoder

Practice

- Hands-on tutorial videos with “teaching assistants” in person & in Zoom breakout rooms

Who we are

Undergraduate students

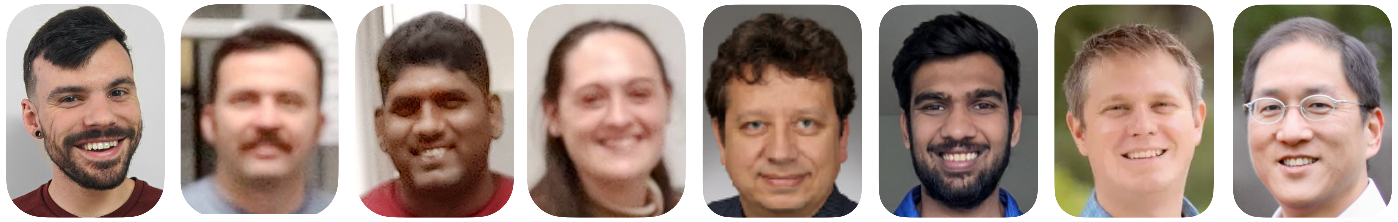


- Steve Roche — lead for all sw
- Dan Stumpp — lead for v1.0 fw
- Steve Racz — lead bit shifter
- Aleksa Rodic — studies vs. hls4ml
- Quincy Bayer — lead for v2.0 fw
- Will Ouligian — lead AE trainer
- Pavel Serhiayenka — lead for HDL fw
- Santiago Cane — lead ATLAS integration
- Kyle Mo — testing

Former

Those highlighted in blue helped prepare today's tutorial

Non students



- Brandon Eubanks — testbench support
- Emre Ercikti — testbench support
- Yuvaraj Elangovan — testbench support
- Isabelle Taylor — testbench support
- Joerg Stelzer — algorithm support
- Rajat Gupta — physics support
- Ben Carlson — physics lead
- Tae Min Hong — project lead

tmhong@pitt.edu

Former

Papers

Hong
Pittsburgh

1.Classification

parallel cuts using HLS

2.Regression

parallel paths using HLS

3.Autoencoder

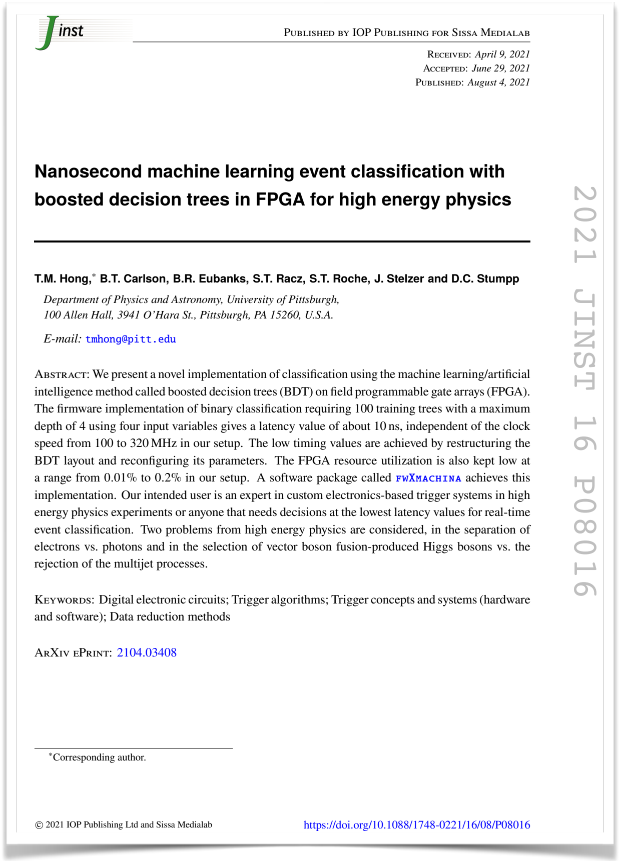
in-house training

bypassing latent space

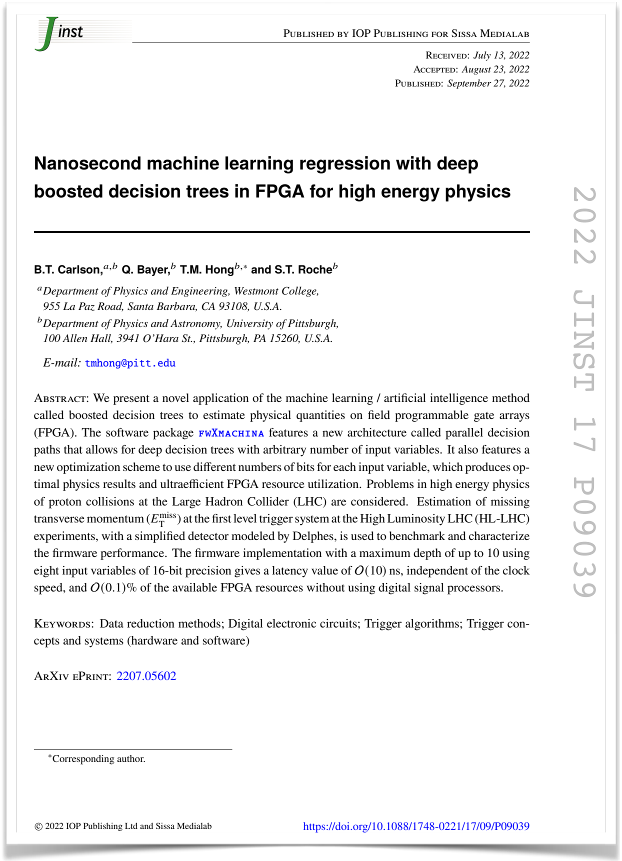
4.Hardware trees

faster & more efficient

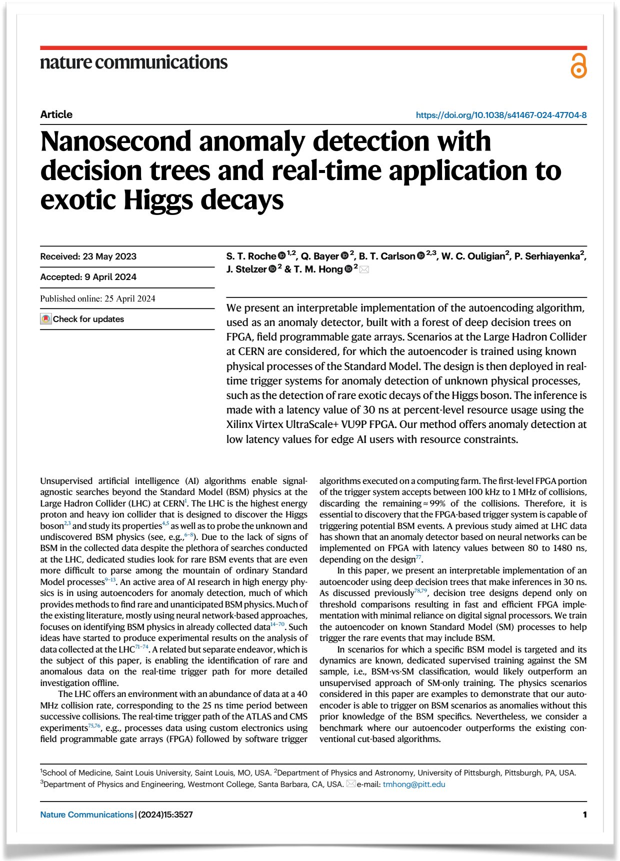
no more HLS


Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics

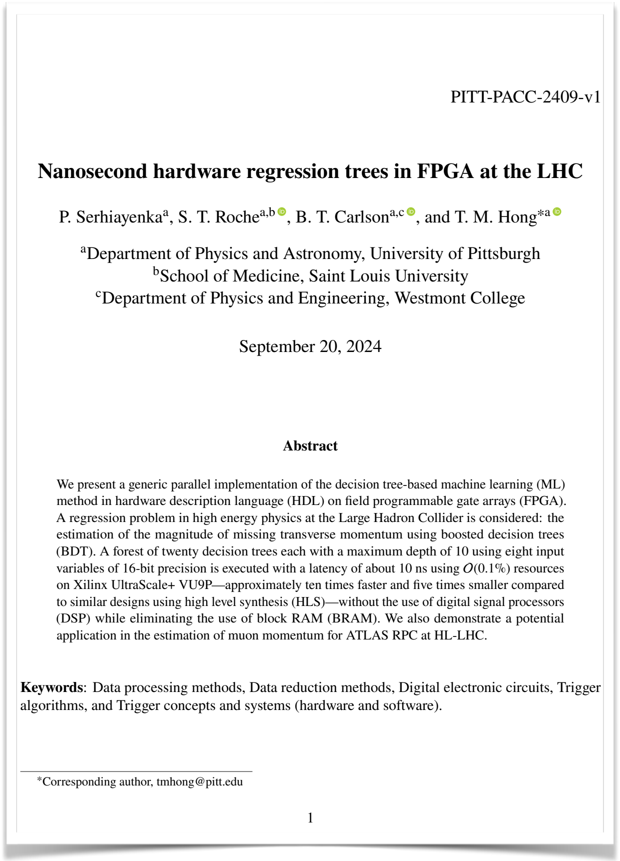
T.M. Hong, B.T. Carlson, B.R. Eubanks, S.T. Racz, S.T. Roche, J. Stelzer and D.C. Stumpp


Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics

B.T. Carlson, Q. Bayer, T.M. Hong and S.T. Roche


nature communications

Nanosecond anomaly detection with decision trees and real-time application to exotic Higgs decays


PITT-PACC-2409-v1

Nanosecond hardware regression trees in FPGA at the LHC

P. Serhiayenka, S. T. Roche, B. T. Carlson, and T. M. Hong

September 20, 2024

Hong et al.
JINST **16**, P08016 (2021)
http://doi.org/10.1088/1748-0221/16/08/P08016

Carlson et al.
JINST **17**, P09039 (2022)
http://doi.org/10.1088/1748-0221/17/09/P09039

Roche et al.
Nat. Comm. **15** (2024) 3527
https://arxiv.org/abs/2304.03836

Serhiayenka et al.
Preprint later this week
https://arxiv.org/abs/???

1st time showing

Python-based code

Hong
Pittsburgh

Availability

- gitlab.com/PittHongGroup/fwX

  parallel cuts (paper 1) - tutorial today

- Shared by email request

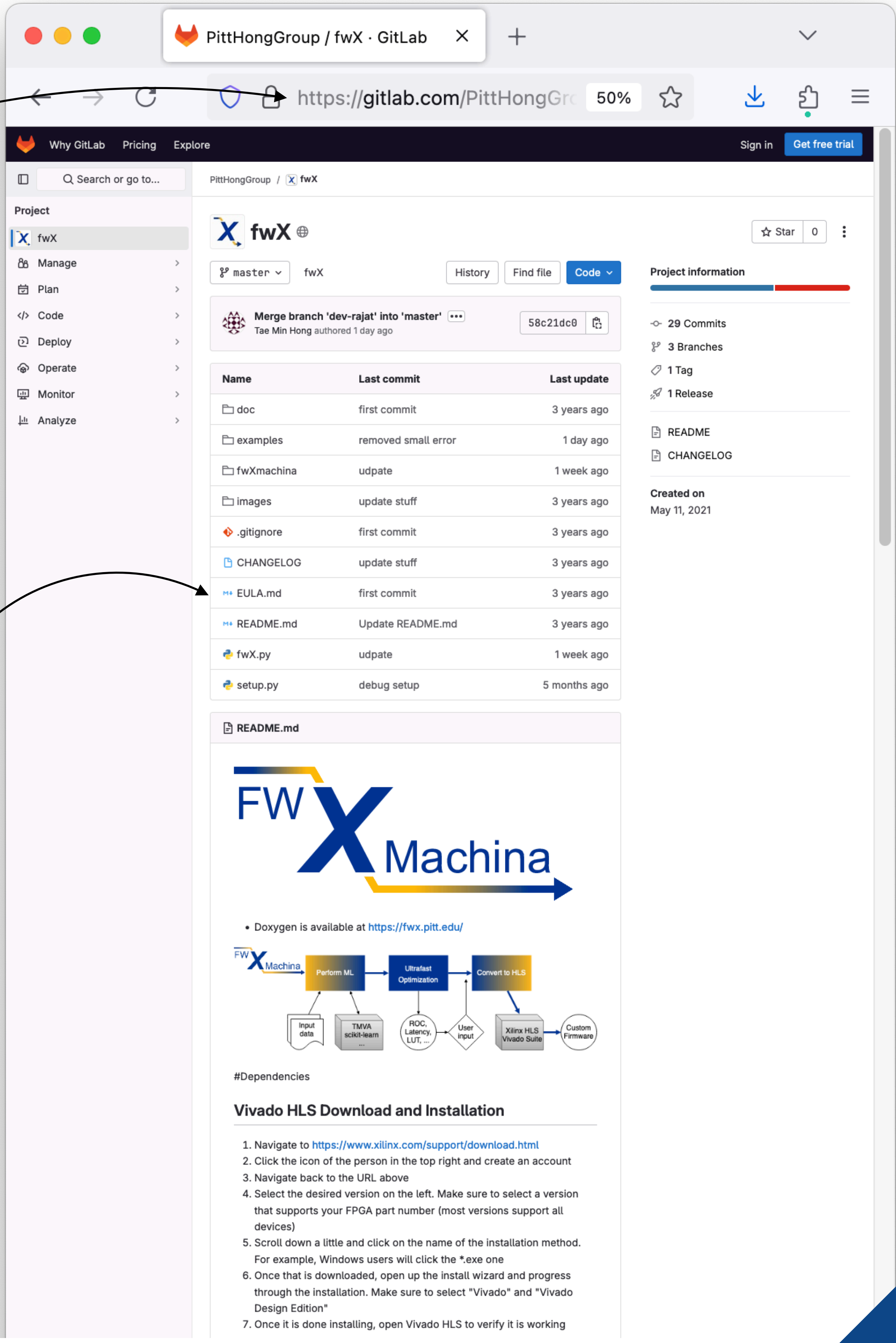
  parallel paths (paper 2)

  autoencoder (paper 3)

  hardware tree (paper 4)

Licensing

- Granted for “Non-Commercial, Educational and Research Purposes”
- Contact Univ. of Pittsburgh Innovation Institute for commerical use
- See EULA for details



Git structure

Hong Pittsburgh

Same structure for all methods

- gitlab.com/PittHongGroup/fwX

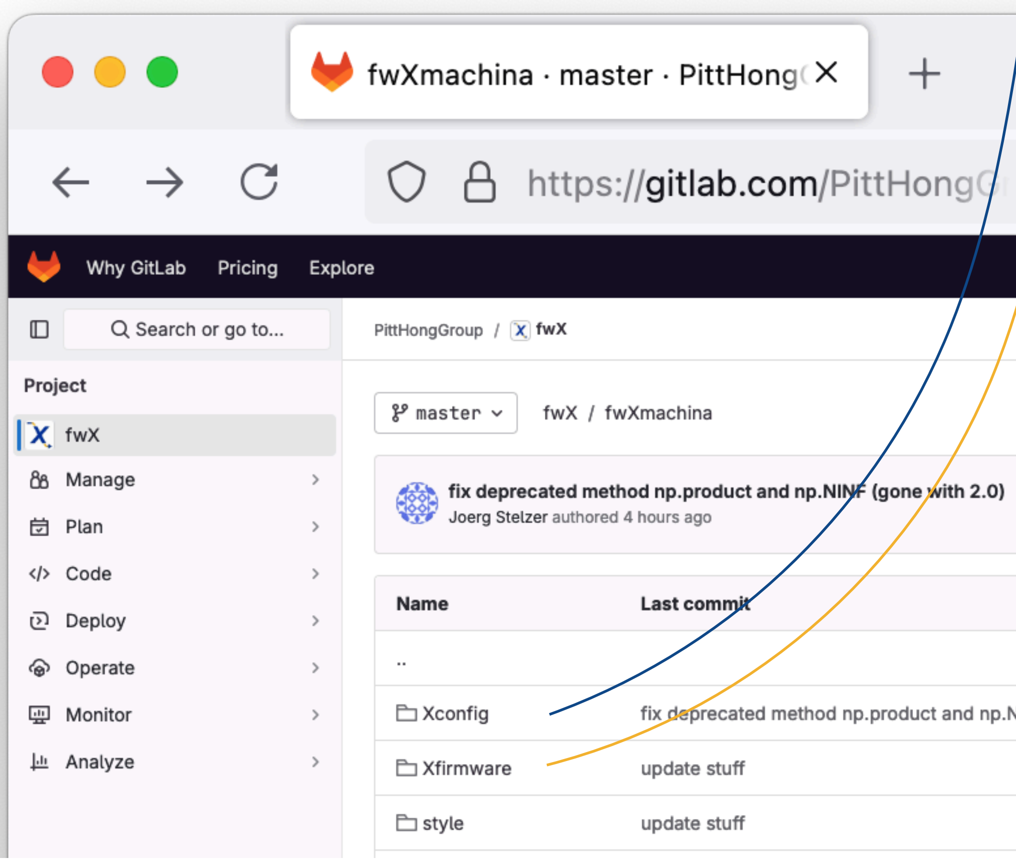
  parallel cuts (paper 1) - tutorial today

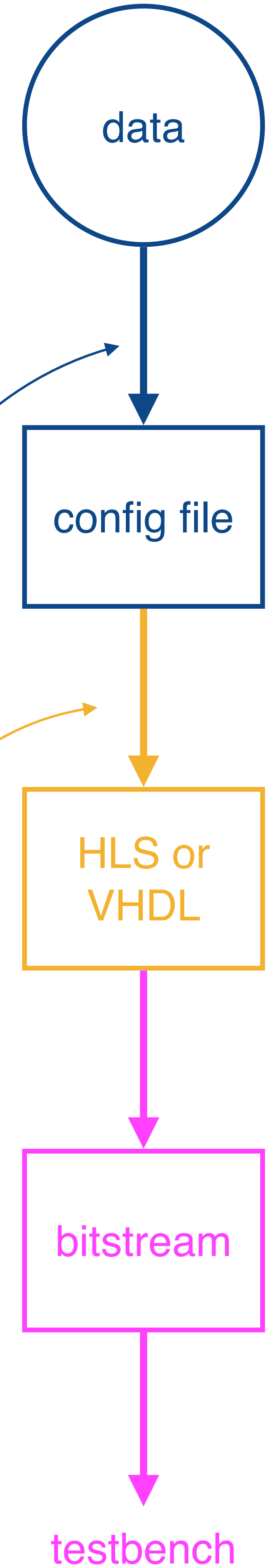
- Available by request

  parallel paths (paper 2)

  autoencoder (paper 3)

  hardware tree (paper 4)





- Xconfig

  creates model configuration

  tutorial - part 1 - Joerg

- Xfirmware

  writes HLS or VHDL

  tutorial - part 2 - Isabelle

- Vivado

  synthesize & testbench

  tutorial - part 3 - Pavel

Summary

Start page at

- fwx.pitt.edu

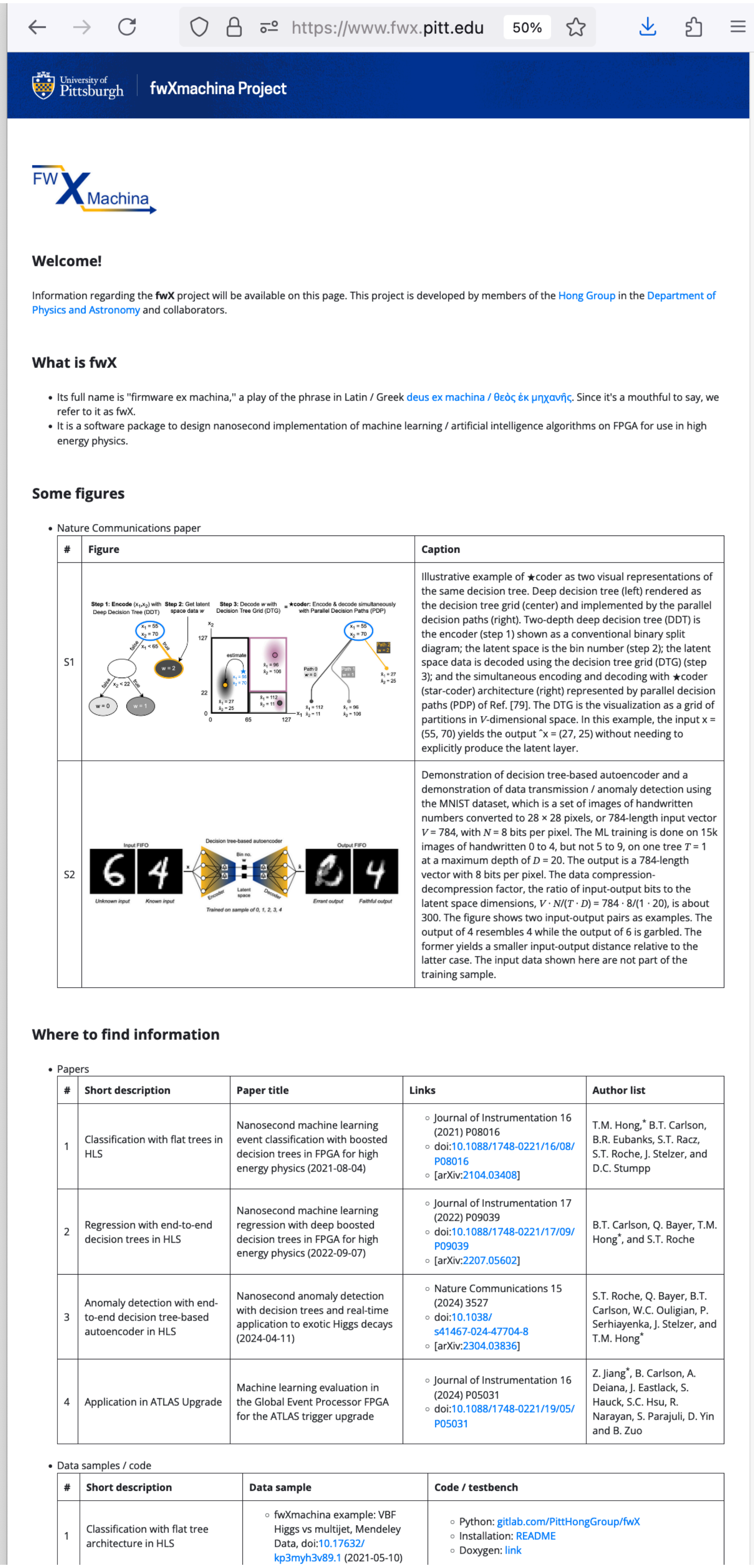
- Content

  Links to papers

  Links to talks

  Links to datasets

  Links to testbenches



https://www.fwx.pitt.edu 50%

University of Pittsburgh | fwXmachina Project

FWX Machina

Welcome!

Information regarding the **fwX** project will be available on this page. This project is developed by members of the Hong Group in the Department of Physics and Astronomy and collaborators.

What is fwX

- Its full name is "firmware ex machina," a play of the phrase in Latin / Greek deus ex machina / θεὸς ἐκ μηχανῆς. Since it's a mouthful to say, we refer to it as fwX.
- It is a software package to design nanosecond implementation of machine learning / artificial intelligence algorithms on FPGA for use in high energy physics.

Some figures

- Nature Communications paper

#	Figure	Caption
S1		Illustrative example of ★coder as two visual representations of the same decision tree. Deep decision tree (left) rendered as the decision tree grid (center) and implemented by the parallel decision paths (right). Two-depth deep decision tree (DDT) is the encoder (step 1) shown as a conventional binary split diagram; the latent space is the bin number (step 2); the latent space data is decoded using the decision tree grid (DTG) (step 3); and the simultaneous encoding and decoding with ★coder (star-coder) architecture (right) represented by parallel decision paths (PDP) of Ref. [79]. The DTG is the visualization as a grid of partitions in V-dimensional space. In this example, the input x = (55, 70) yields the output ˆx = (27, 25) without needing to explicitly produce the latent layer.
S2		Demonstration of decision tree-based autoencoder and a demonstration of data transmission / anomaly detection using the MNIST dataset, which is a set of images of handwritten numbers converted to 28 × 28 pixels, or 784-length input vector $V = 784$, with $N = 8$ bits per pixel. The ML training is done on 15k images of handwritten 0 to 4, but not 5 to 9, on one tree $T = 1$ at a maximum depth of $D = 20$. The output is a 784-length vector with 8 bits per pixel. The data compression-decompression factor, the ratio of input-output bits to the latent space dimensions, $V \cdot N/(T \cdot D) = 784 \cdot 8/(1 \cdot 20)$, is about 300. The figure shows two input-output pairs as examples. The output of 4 resembles 4 while the output of 6 is garbled. The former yields a smaller input-output distance relative to the latter case. The input data shown here are not part of the training sample.

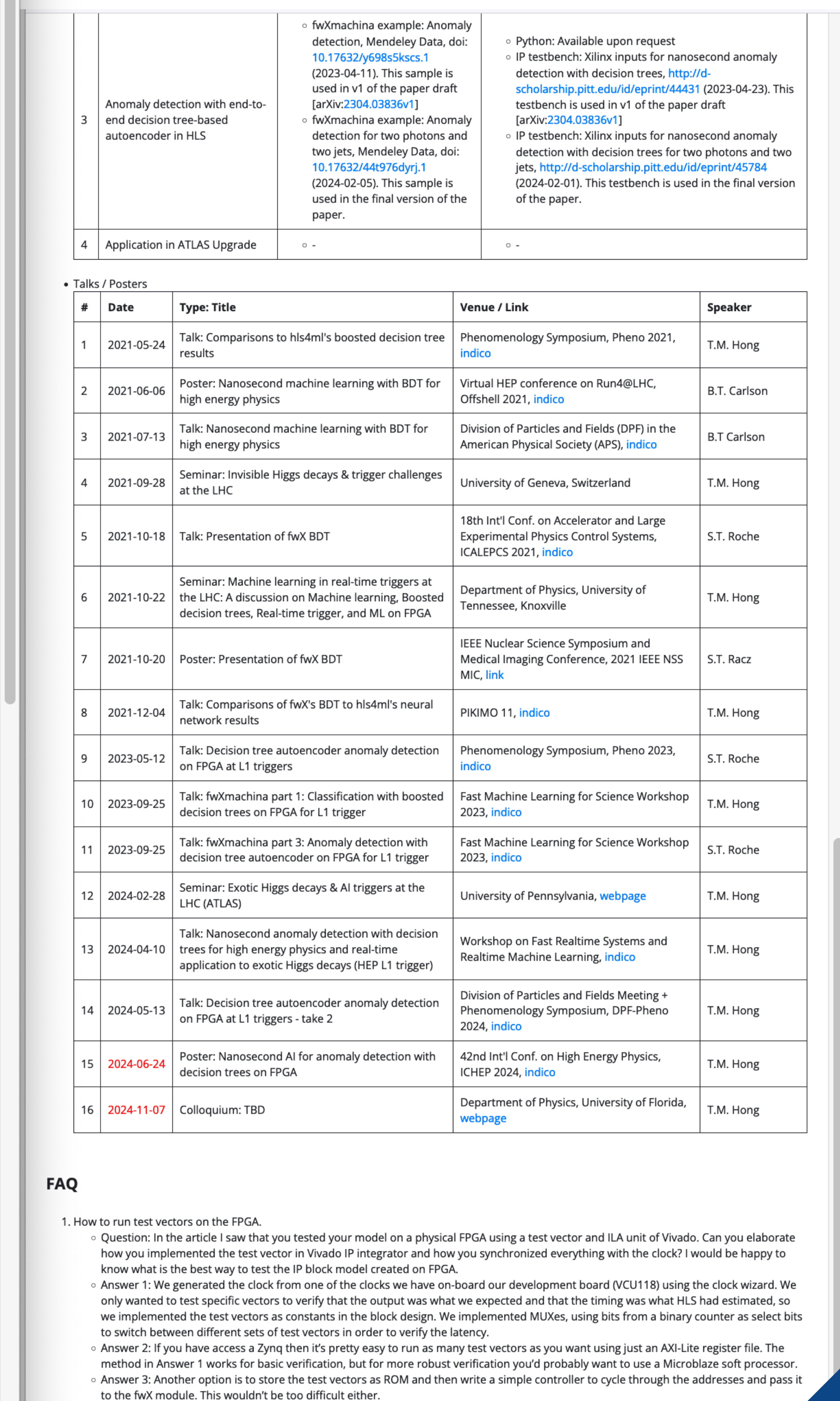
Where to find information

- Papers

#	Short description	Paper title	Links	Author list
1	Classification with flat trees in HLS	Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics (2021-08-04)	◦ Journal of Instrumentation 16 (2021) P08016 ◦ doi:10.1088/1748-0221/16/08/P08016 ◦ [arXiv:2104.03408]	T.M. Hong,* B.T. Carlson, B.R. Eubanks, S.T. Racz, S.T. Roche, J. Stelzer, and D.C. Stumpp
2	Regression with end-to-end decision trees in HLS	Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics (2022-09-07)	◦ Journal of Instrumentation 17 (2022) P09039 ◦ doi:10.1088/1748-0221/17/09/P09039 ◦ [arXiv:2207.05602]	B.T. Carlson, Q. Bayer, T.M. Hong*, and S.T. Roche
3	Anomaly detection with end-to-end decision tree-based autoencoder in HLS	Nanosecond anomaly detection with decision trees and real-time application to exotic Higgs decays (2024-04-11)	◦ Nature Communications 15 (2024) 3527 ◦ doi:10.1038/s41467-024-47704-8 ◦ [arXiv:2304.03836]	S.T. Roche, Q. Bayer, B.T. Carlson, W.C. Ouligian, P. Serhiayenka, J. Stelzer, and T.M. Hong*
4	Application in ATLAS Upgrade	Machine learning evaluation in the Global Event Processor FPGA for the ATLAS trigger upgrade	◦ Journal of Instrumentation 16 (2024) P05031 ◦ doi:10.1088/1748-0221/19/05/P05031	Z. Jiang*, B. Carlson, A. Deiana, J. Eastlack, S. Hauck, S.C. Hsu, R. Narayan, S. Parajuli, D. Yin and B. Zuo

- Data samples / code

#	Short description	Data sample	Code / testbench
1	Classification with flat tree architecture in HLS	◦ fwXmachina example: VBF Higgs vs multijet, Mendeley Data, doi:10.17632/kp3myh3v89.1 (2021-05-10)	◦ Python: gitlab.com/PittHongGroup/fwX ◦ Installation: README ◦ Doxygen: link



#	Short description	Data sample	Code / testbench
3	Anomaly detection with end-to-end decision tree-based autoencoder in HLS	◦ fwXmachina example: Anomaly detection, Mendeley Data, doi: 10.17632/y698s5kscs.1 (2023-04-11). This sample is used in v1 of the paper draft [arXiv:2304.03836v1] ◦ fwXmachina example: Anomaly detection for two photons and two jets, Mendeley Data, doi: 10.17632/44t976dyrj.1 (2024-02-05). This sample is used in the final version of the paper.	◦ Python: Available upon request ◦ IP testbench: Xilinx inputs for nanosecond anomaly detection with decision trees, http://d-scholarship.pitt.edu/id/eprint/44431 (2023-04-23). This testbench is used in v1 of the paper draft [arXiv:2304.03836v1] ◦ IP testbench: Xilinx inputs for nanosecond anomaly detection with decision trees for two photons and two jets, http://d-scholarship.pitt.edu/id/eprint/45784 (2024-02-01). This testbench is used in the final version of the paper.
4	Application in ATLAS Upgrade	◦ -	◦ -

- Talks / Posters

#	Date	Type: Title	Venue / Link	Speaker
1	2021-05-24	Talk: Comparisons to hls4ml's boosted decision tree results	Phenomenology Symposium, Pheno 2021, indico	T.M. Hong
2	2021-06-06	Poster: Nanosecond machine learning with BDT for high energy physics	Virtual HEP conference on Run4@LHC, Offshell 2021, indico	B.T. Carlson
3	2021-07-13	Talk: Nanosecond machine learning with BDT for high energy physics	Division of Particles and Fields (DPF) in the American Physical Society (APS), indico	B.T Carlson
4	2021-09-28	Seminar: Invisible Higgs decays & trigger challenges at the LHC	University of Geneva, Switzerland	T.M. Hong
5	2021-10-18	Talk: Presentation of fwX BDT	18th Int'l Conf. on Accelerator and Large Experimental Physics Control Systems, ICALEPCS 2021, indico	S.T. Roche
6	2021-10-22	Seminar: Machine learning in real-time triggers at the LHC: A discussion on Machine learning, Boosted decision trees, Real-time trigger, and ML on FPGA	Department of Physics, University of Tennessee, Knoxville	T.M. Hong
7	2021-10-20	Poster: Presentation of fwX BDT	IEEE Nuclear Science Symposium and Medical Imaging Conference, 2021 IEEE NSS MIC, link	S.T. Racz
8	2021-12-04	Talk: Comparisons of fwX's BDT to hls4ml's neural network results	PIKIMO 11, indico	T.M. Hong
9	2023-05-12	Talk: Decision tree autoencoder anomaly detection on FPGA at L1 triggers	Phenomenology Symposium, Pheno 2023, indico	S.T. Roche
10	2023-09-25	Talk: fwXmachina part 1: Classification with boosted decision trees on FPGA for L1 trigger	Fast Machine Learning for Science Workshop 2023, indico	T.M. Hong
11	2023-09-25	Talk: fwXmachina part 3: Anomaly detection with decision tree autoencoder on FPGA for L1 trigger	Fast Machine Learning for Science Workshop 2023, indico	S.T. Roche
12	2024-02-28	Seminar: Exotic Higgs decays & AI triggers at the LHC (ATLAS)	University of Pennsylvania, webpage	T.M. Hong
13	2024-04-10	Talk: Nanosecond anomaly detection with decision trees for high energy physics and real-time application to exotic Higgs decays (HEP L1 trigger)	Workshop on Fast Realtime Systems and Realtime Machine Learning, indico	T.M. Hong
14	2024-05-13	Talk: Decision tree autoencoder anomaly detection on FPGA at L1 triggers - take 2	Division of Particles and Fields Meeting + Phenomenology Symposium, DPF-Pheno 2024, indico	T.M. Hong
15	2024-06-24	Poster: Nanosecond AI for anomaly detection with decision trees on FPGA	42nd Int'l Conf. on High Energy Physics, ICHEP 2024, indico	T.M. Hong
16	2024-11-07	Colloquium: TBD	Department of Physics, University of Florida, webpage	T.M. Hong

FAQ

1. How to run test vectors on the FPGA.
   - Question: In the article I saw that you tested your model on a physical FPGA using a test vector and ILA unit of Vivado. Can you elaborate how you implemented the test vector in Vivado IP integrator and how you synchronized everything with the clock? I would be happy to know what is the best way to test the IP block model created on FPGA.
   - Answer 1: We generated the clock from one of the clocks we have on-board our development board (VCU118) using the clock wizard. We only wanted to test specific vectors to verify that the output was what we expected and that the timing was what HLS had estimated, so we implemented the test vectors as constants in the block design. We implemented MUXes, using bits from a binary counter as select bits to switch between different sets of test vectors in order to verify the latency.
   - Answer 2: If you have access a Zynq then it's pretty easy to run as many test vectors as you want using just an AXI-Lite register file. The method in Answer 1 works for basic verification, but for more robust verification you'd probably want to use a Microblaze soft processor.
   - Answer 3: Another option is to store the test vectors as ROM and then write a simple controller to cycle through the addresses and pass it to the fwX module. This wouldn't be too difficult either.

Hong Pittsburgh

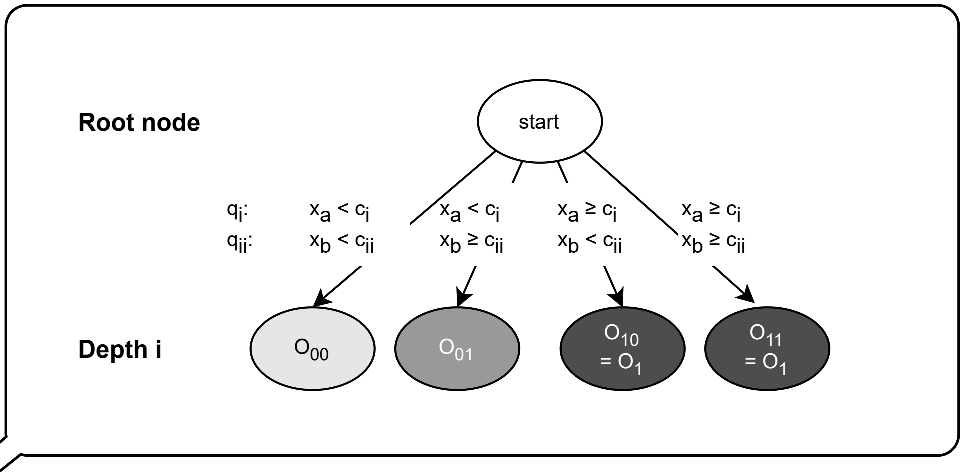
Outline

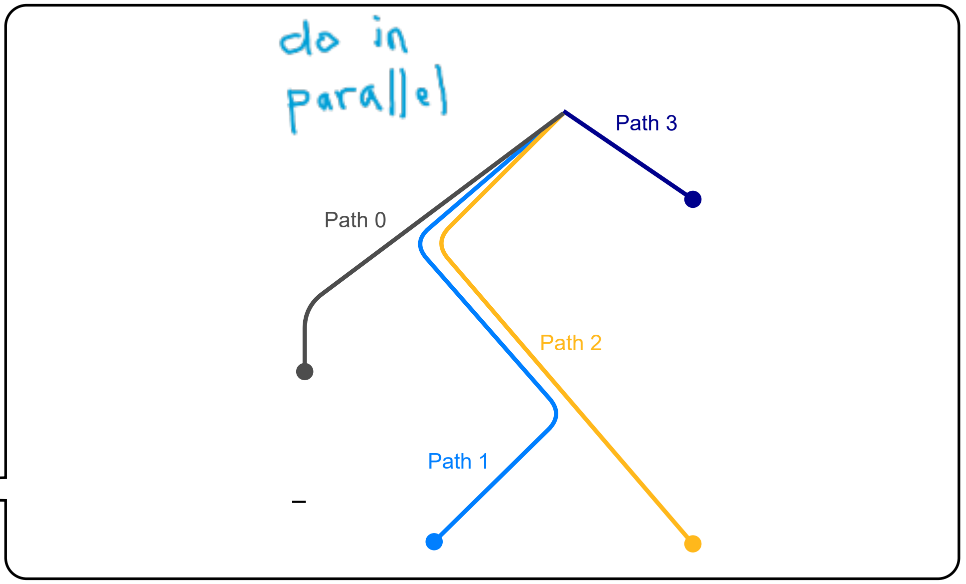
Introduction

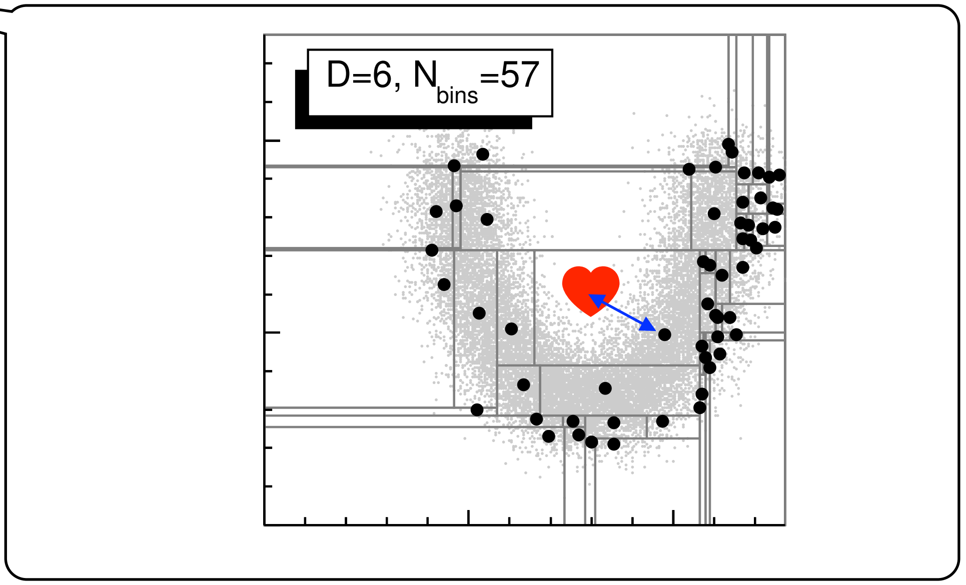
- Who are we
- What products do we offer
- Summary if you have to leave

Theory

- Parallelizing decision trees
- Classification & regression
- Anomaly detection w/ autoencoder







Practice

- Hands-on tutorial videos with “teaching assistants” in person & in Zoom breakout rooms

Papers

Hong
Pittsburgh

1. Classification

parallel cuts using HLS

2. Regression

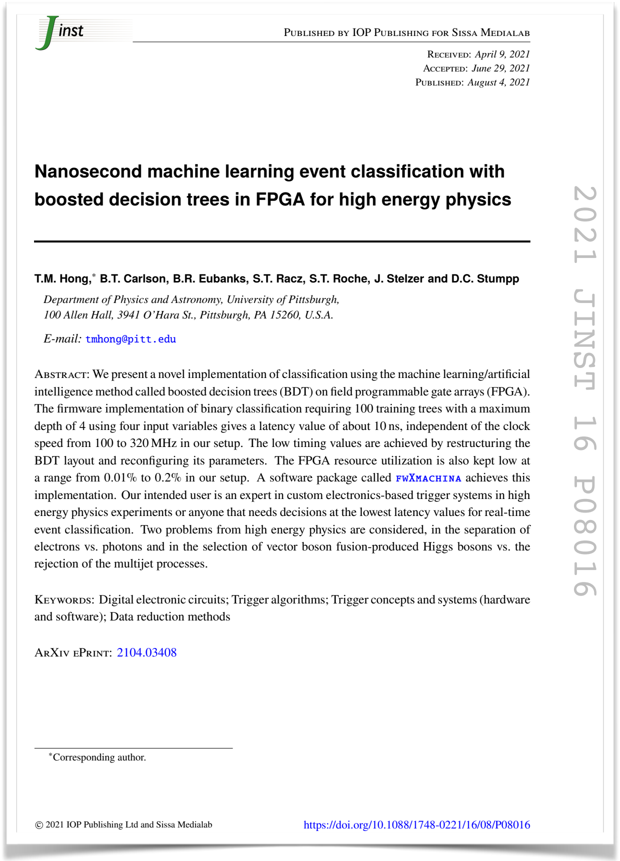
parallel paths using HLS

3. Autoencoder

in-house training
bypassing latent space

4. Hardware trees

faster & more efficient
no more HLS

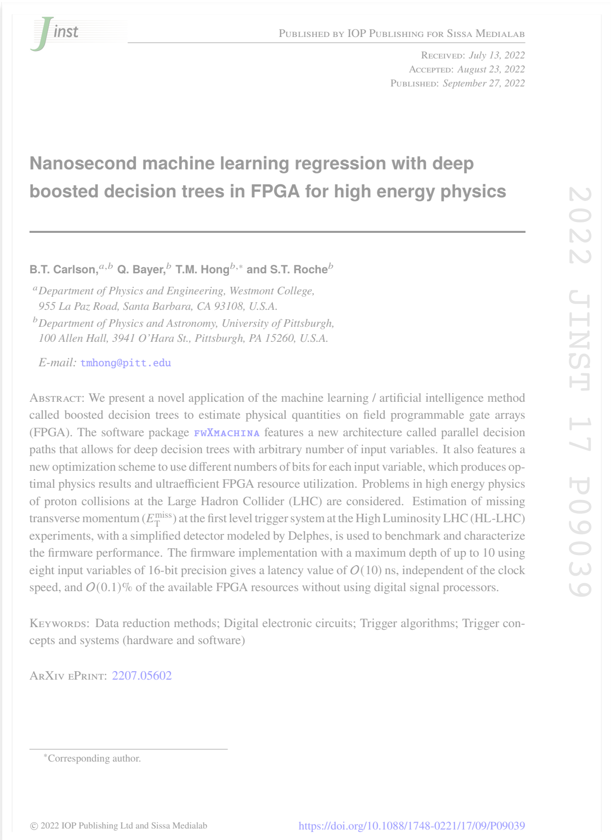


Published by IOP Publishing for Sissa Medialab
Received: April 9, 2021
Accepted: June 29, 2021
Published: August 4, 2021

Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics

T.M. Hong,* B.T. Carlson, B.R. Eubanks, S.T. Racz, S.T. Roche, J. Stelzer and D.C. Stumpp

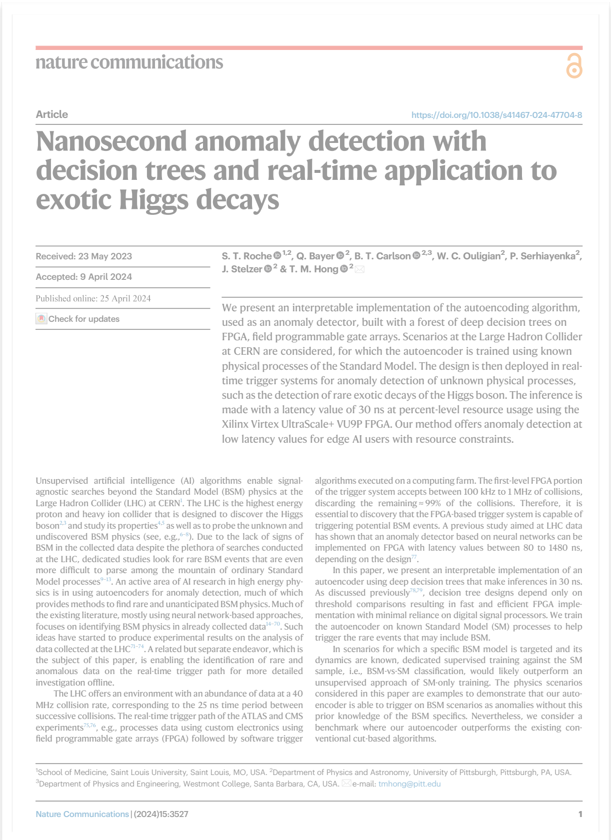
2021 JINST 16 P08016



Published by IOP Publishing for Sissa Medialab
Received: July 13, 2022
Accepted: August 23, 2022
Published: September 27, 2022

Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics

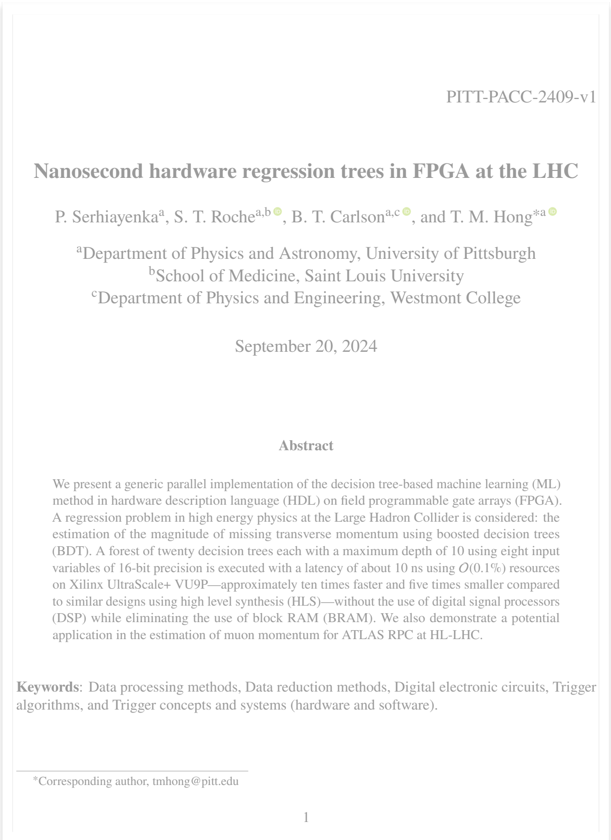
2022 JINST 17 P09039



nature communications

Article

Nanosecond anomaly detection with decision trees and real-time application to exotic Higgs decays



PITT-PACC-2409-v1

Nanosecond hardware regression trees in FPGA at the LHC

P. Serhiayenka, S. T. Roche, B. T. Carlson, and T. M. Hong

September 20, 2024

Abstract

Hong et al.
JINST 16, P08016 (2021)
http://doi.org/10.1088/1748-0221/16/08/P08016

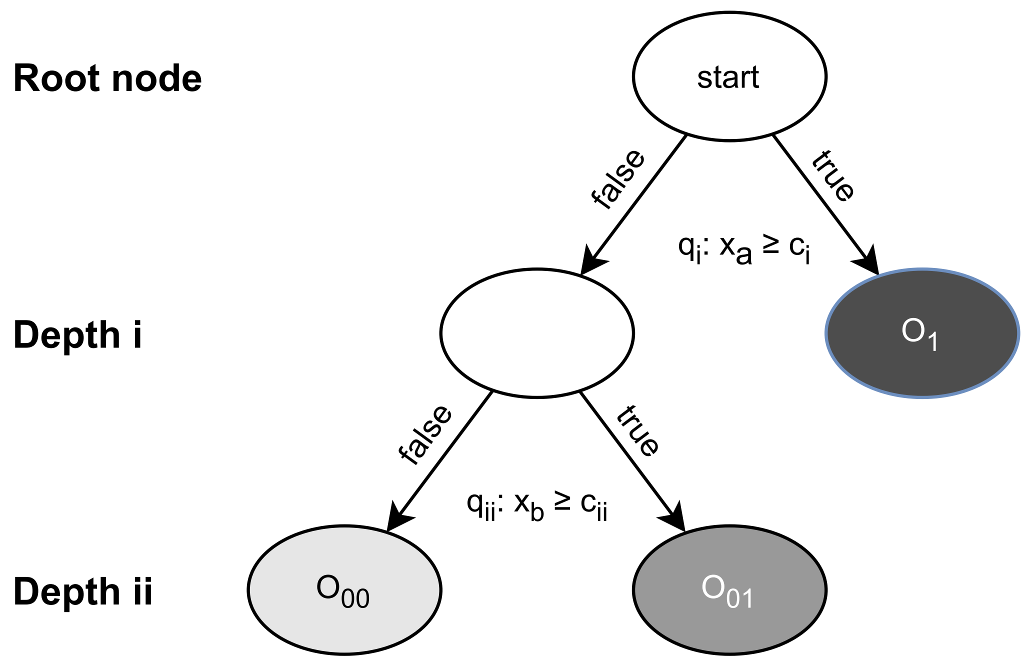
Carlson et al.
JINST **17**, P09039 (2022)
http://doi.org/10.1088/1748-0221/17/09/P09039

Roche et al.
Nat. Comm. **15** (2024) 3527
https://arxiv.org/abs/2304.03836

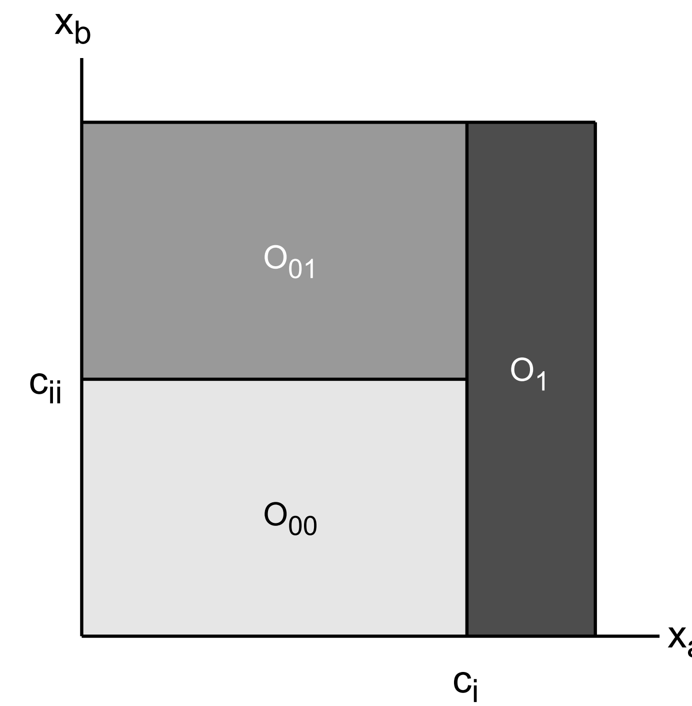
Serhiayenka et al.
Preprint later this week
https://arxiv.org/abs/???

1st time showing

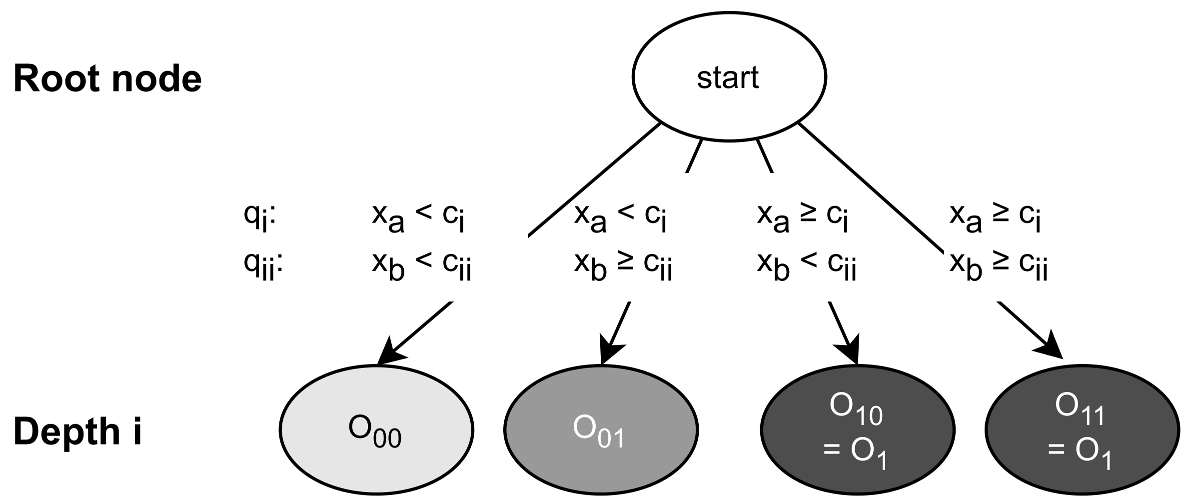
Decision tree

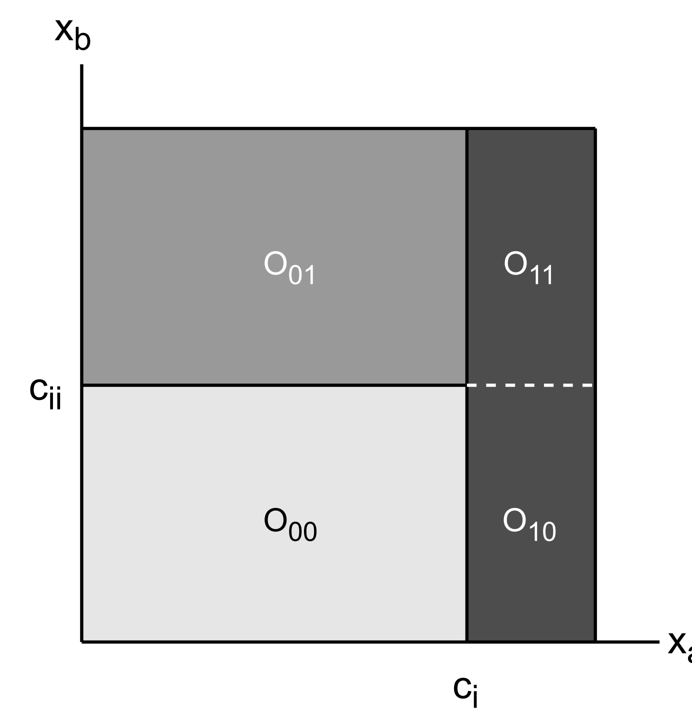


2d plane: x_a vs. x_b



Parallelize cuts

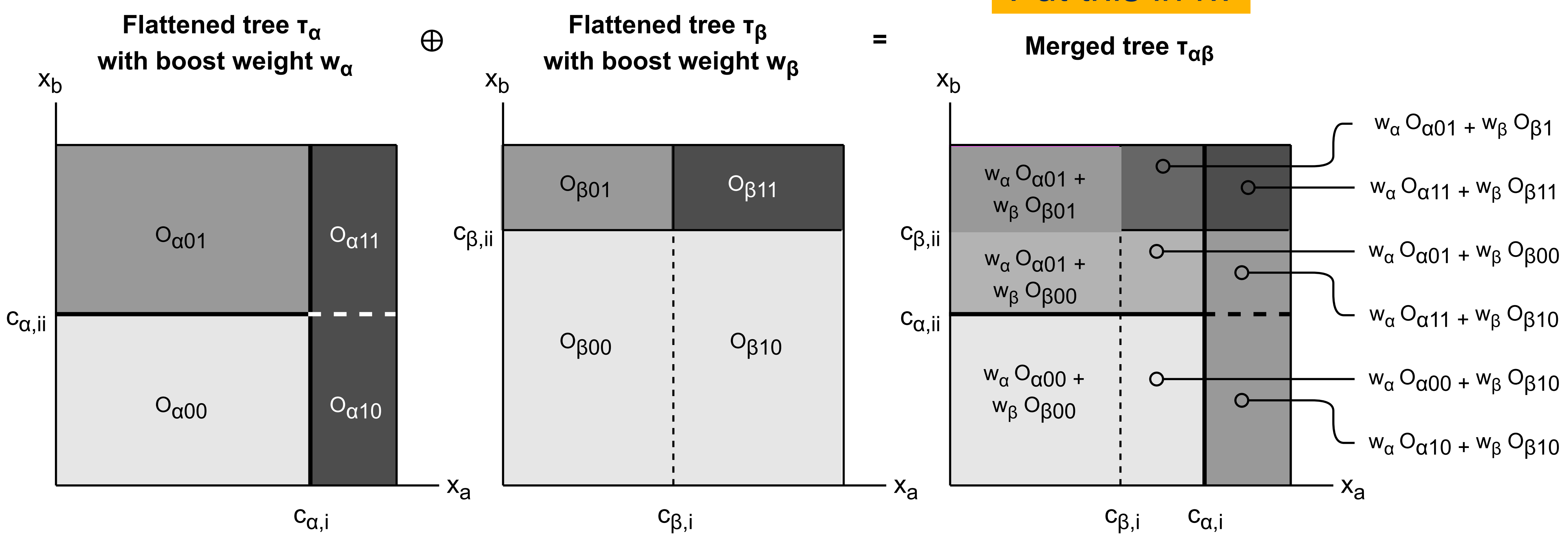




Paper 1: Pre-merge trees

TM Hong

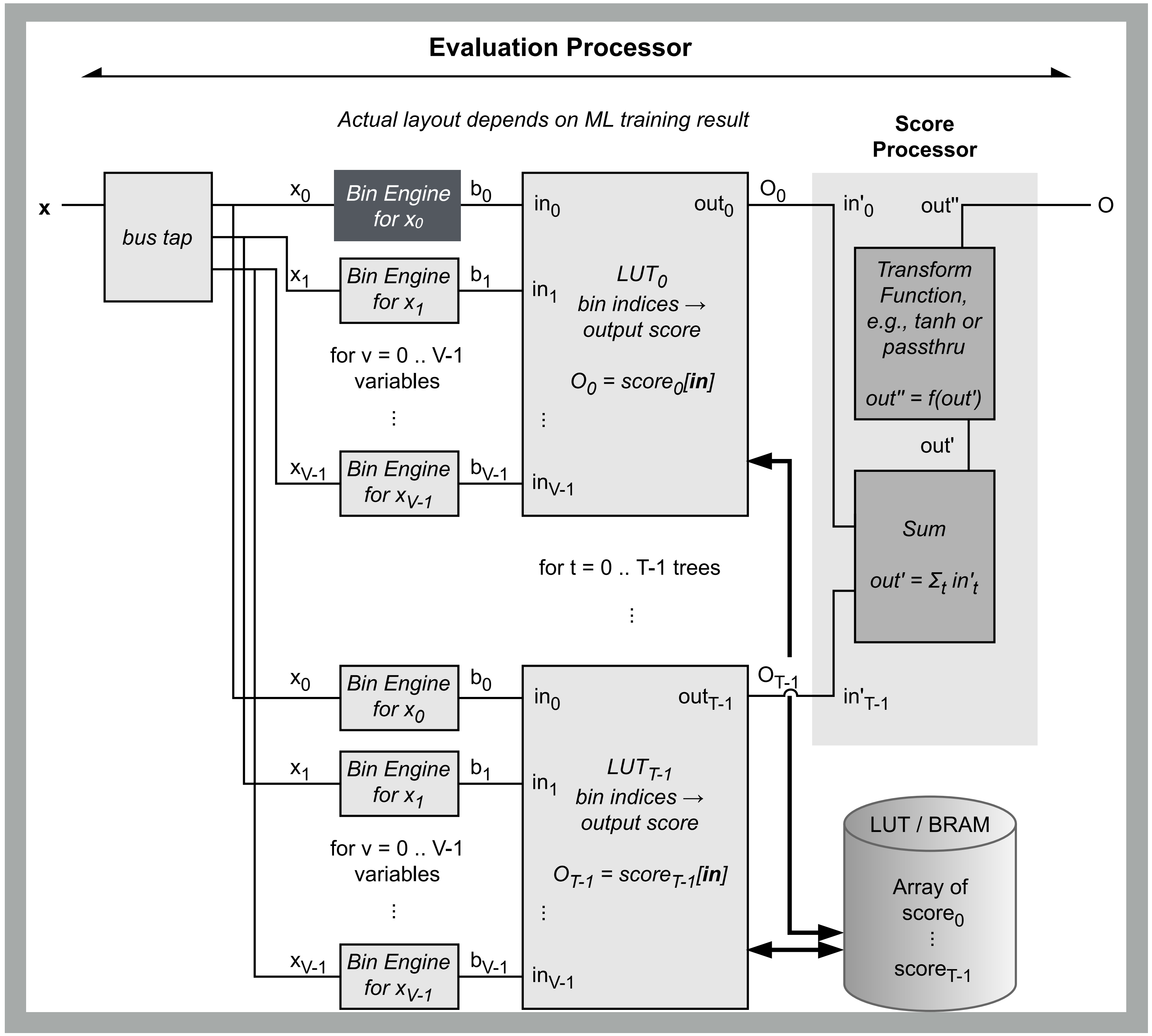
Put this in fw



Paper 1: Block diagram

TM Hong

Input variables



Output score

Parallel process per variable (next slide)

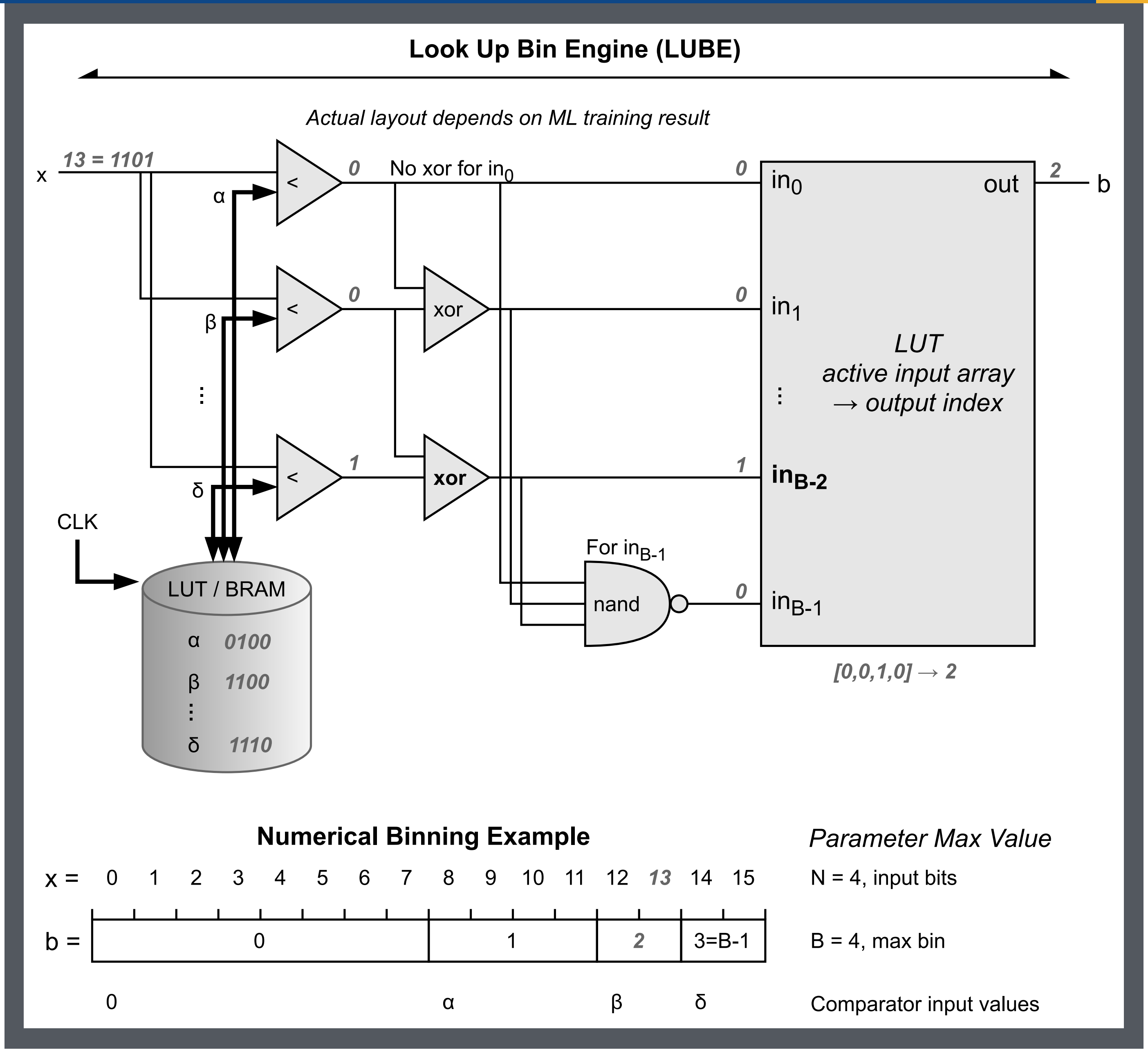
Look up the output score given per decision tree

Combine the scores of the trees in the forest

Paper 1: Look up bin engine

TM Hong

Input variable



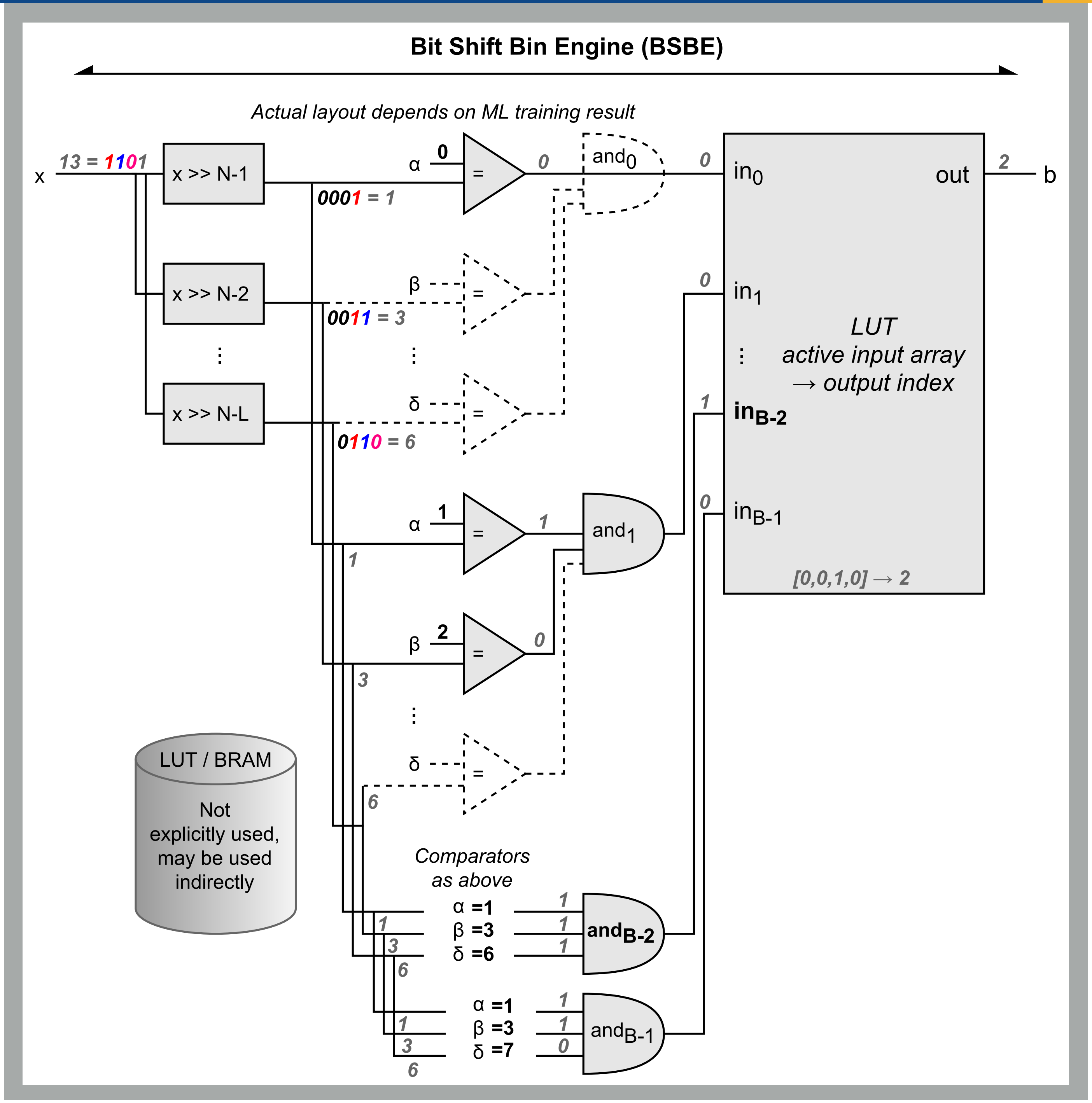
Bin index

- Search for the bin where the data point lives

Paper 1: Bit shifting

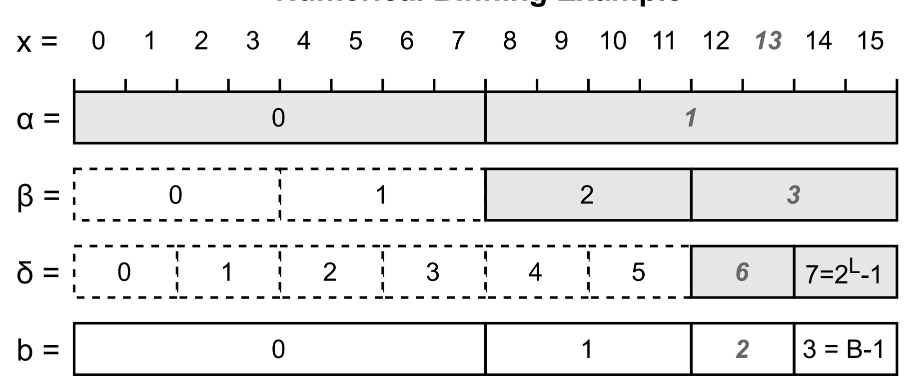
TM Hong

Input variable



Bin index

Numerical Binning Example



Parameter Max Value

N = 4, input bits

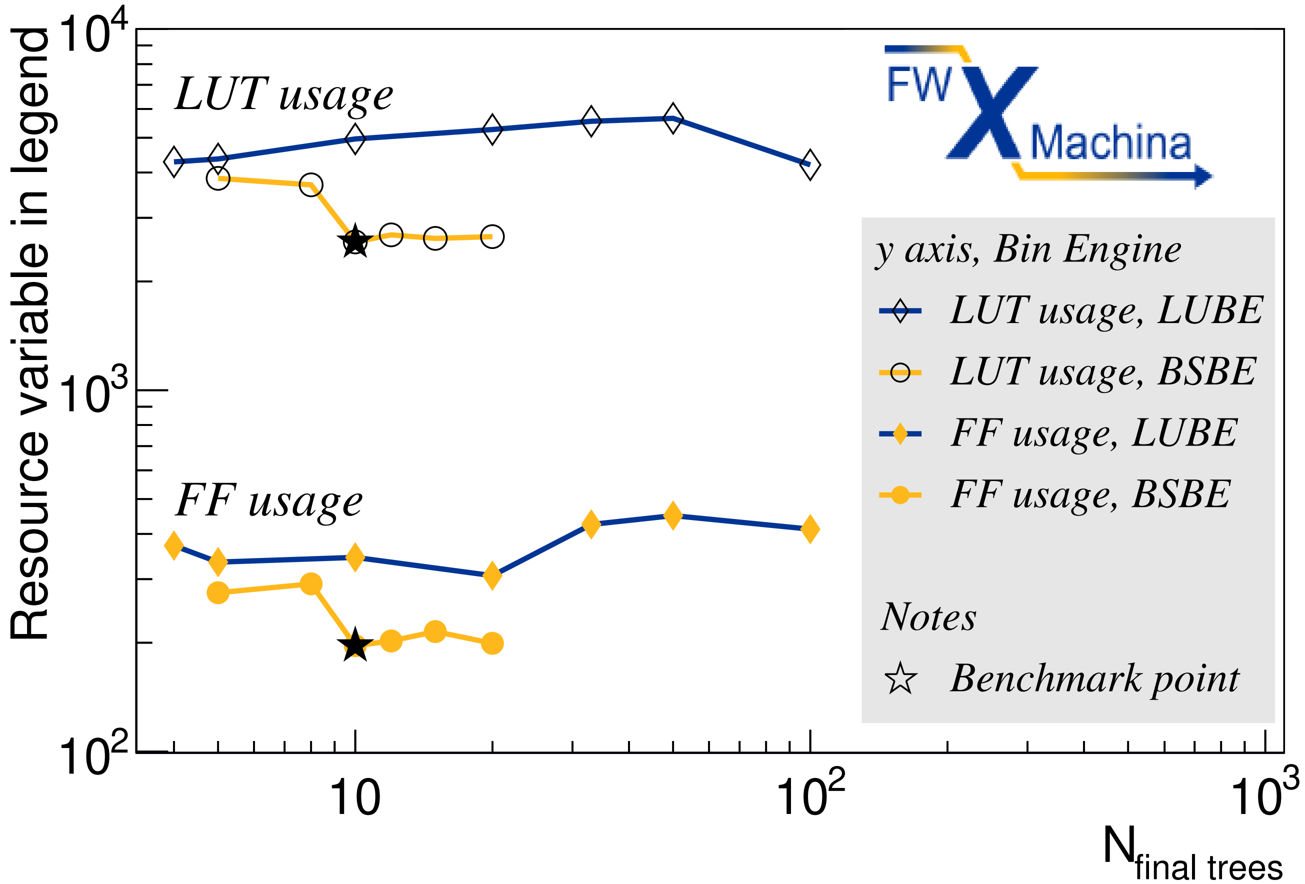
ℓ = 0, layer no.

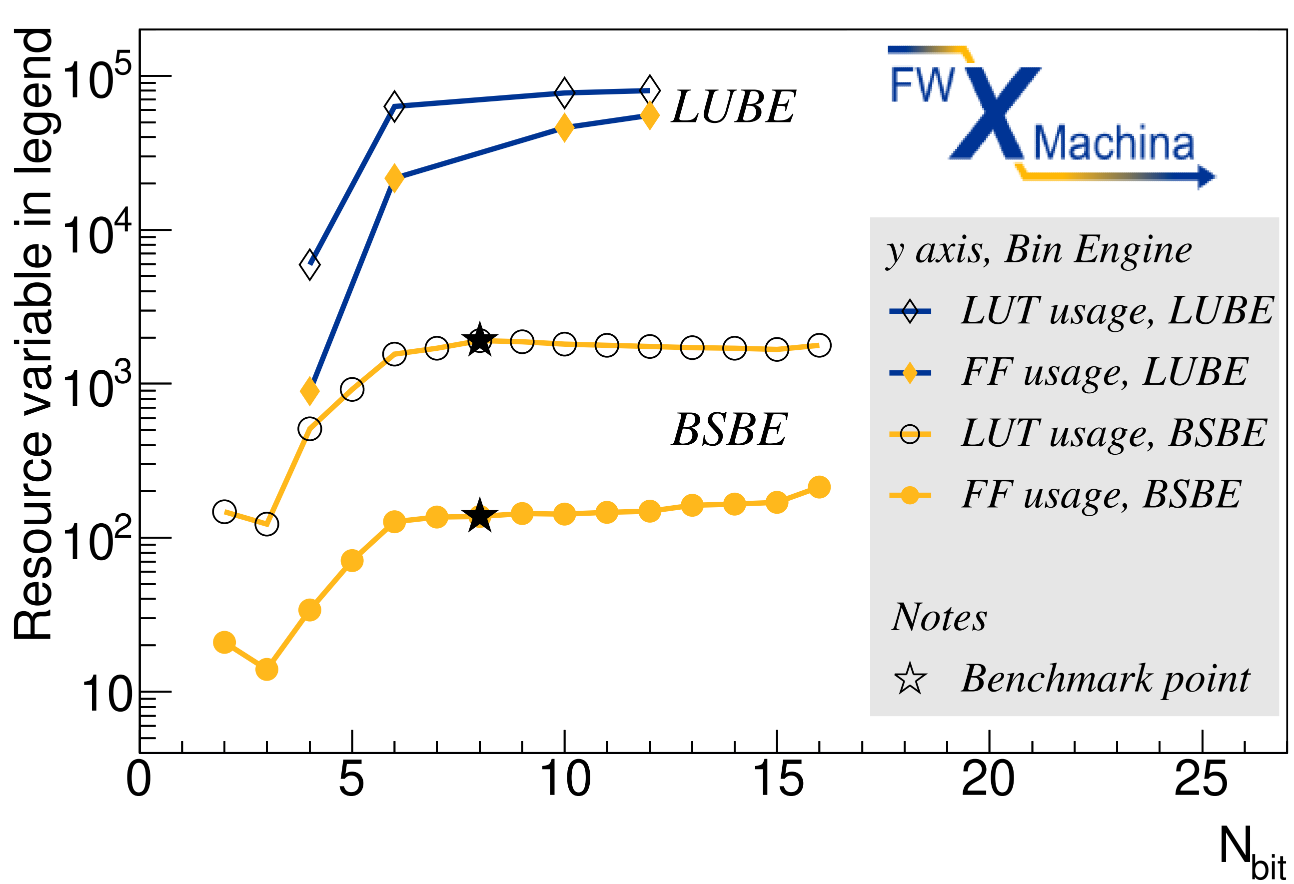
ℓ = 2, layer no.

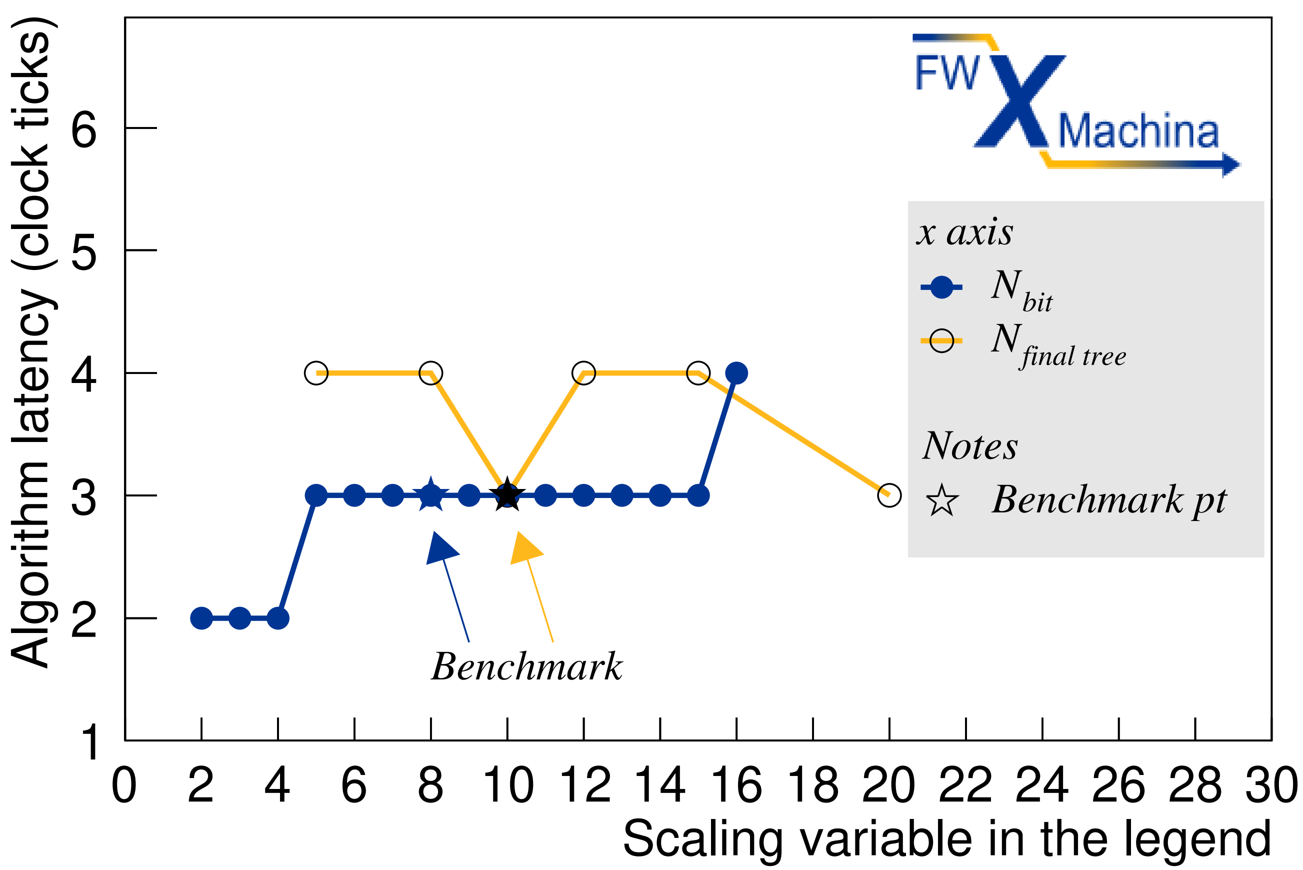
ℓ = 3 = L - 1, max layer

B = 4, max bin

Paper 1: Scaling







- Does <u>not</u> scale with trees or bits

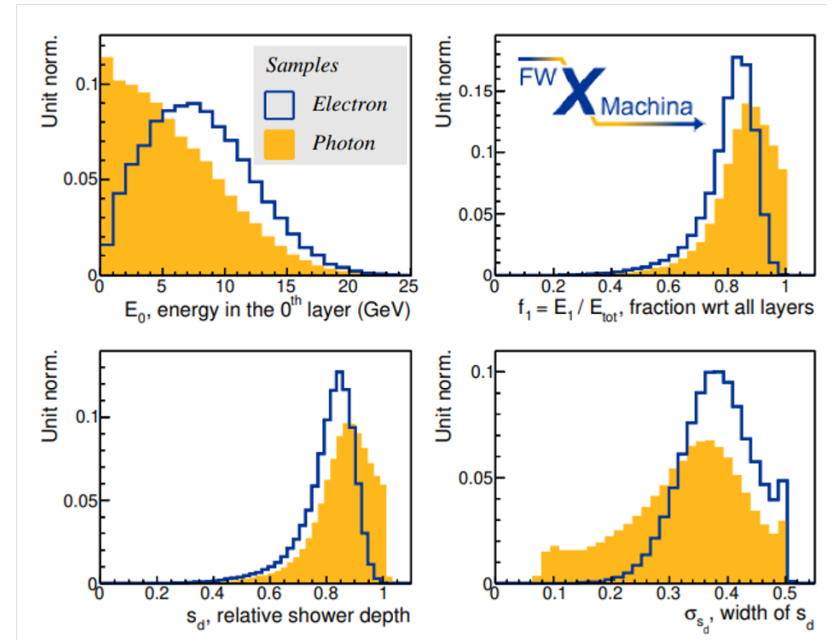
Paper 1: vs. hls4ml family

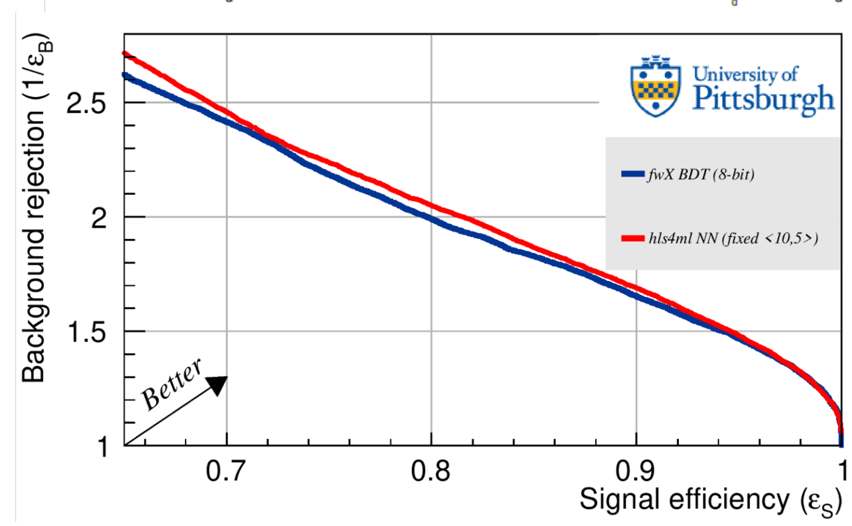
- Setup

Physics	4 variables for e vs. γ[1]
FWX paper 1	100 trees, 4 deep
hls4ml BDT	” identical config for BDT
hls4ml NN	Out-of-the-box config

- fwX paper 1

vs. hls4ml NN	Comparable[2]
vs. hls4ml BDT	Same (since identical config)
Resource	< 1% for all methods
Latency	FWX’s parallel + no clocked operations





	fwX paper 1	hls4ml NN	hls4ml conifer-BDT
# bits	〈 8 〉	〈 10, 5 〉	〈 10, 5 〉
LUT	0.06%	0.1%	0.3%
Flip Flops	0.01%	0.01%	0.1%
BRAM	0.1%	0.2%	0
DSP	0.03%	0.02%	0
Latency	10 ns	25 ns	47 ns
Interval	1 clock tick	1 clock tick	1 clock tick

[1] Nachman et al., https://data.mendeley.com/datasets/kp3myh3v89/1
[2] Hong, PIKIMO 11, https://indico.cern.ch/event/1091676/contributions/4639362/

Papers

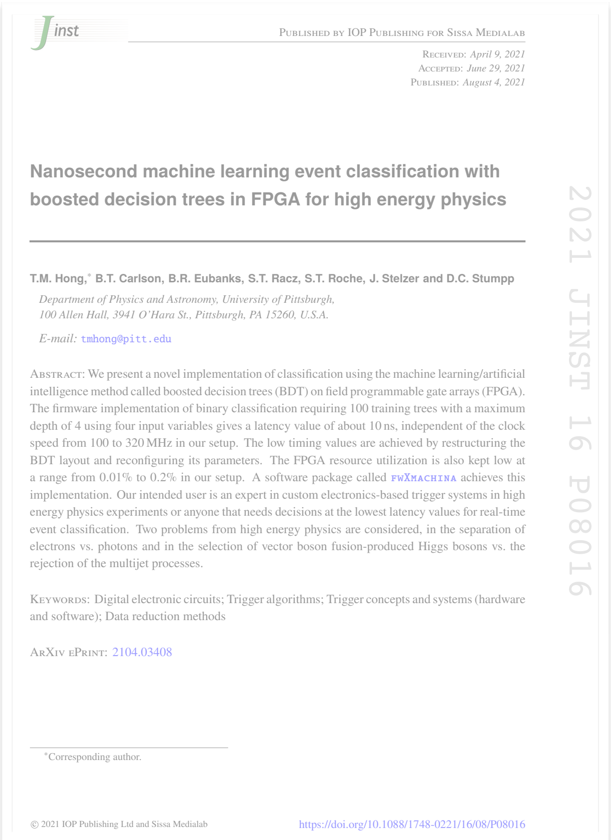
Hong
Pittsburgh

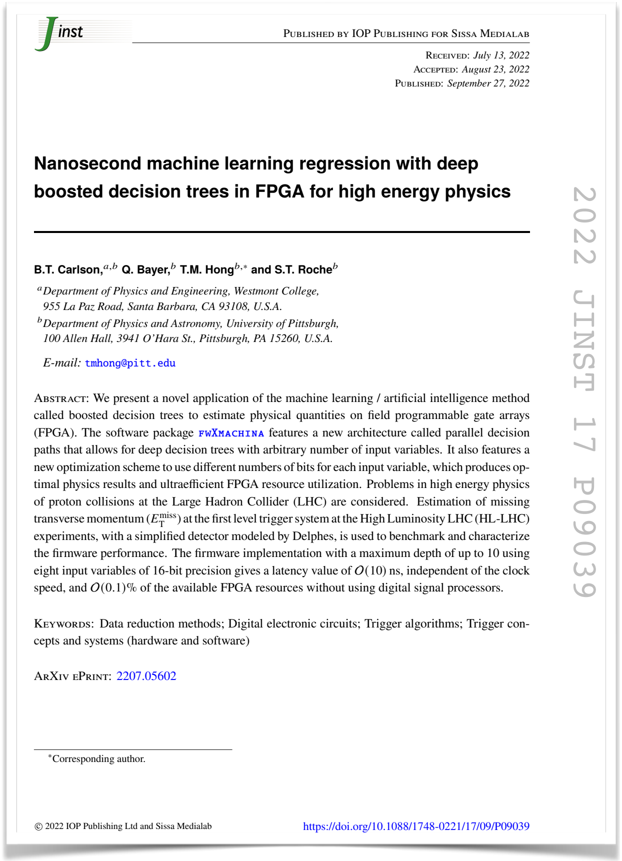
1.Classification
parallel cuts using HLS

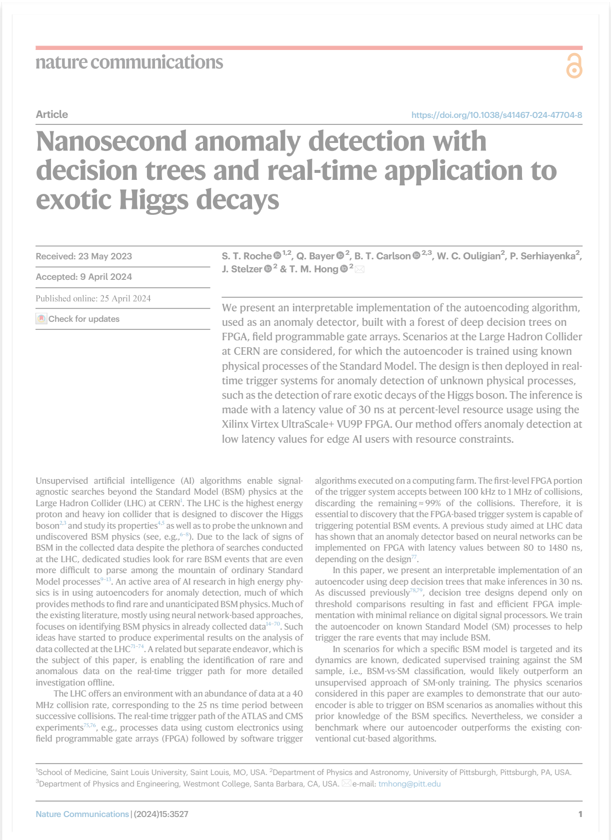
2.Regression
parallel paths using HLS

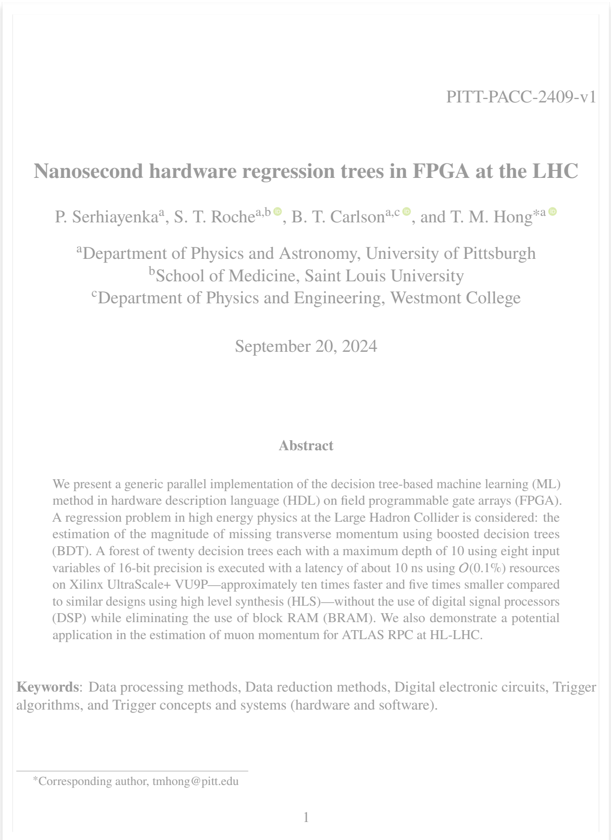
3.Autoencoder
in-house training
bypassing latent space

4.Hardware trees
faster & more efficient
no more HLS









Hong et al.
JINST 16, P08016 (2021)
http://doi.org/10.1088/1748-0221/16/08/P08016

Carlson et al.
JINST **17**, P09039 (2022)
http://doi.org/10.1088/1748-0221/17/09/P09039

Roche et al.
Nat. Comm. **15** (2024) 3527
https://arxiv.org/abs/2304.03836

Serhiayenka et al.
Preprint later this week
https://arxiv.org/abs/???

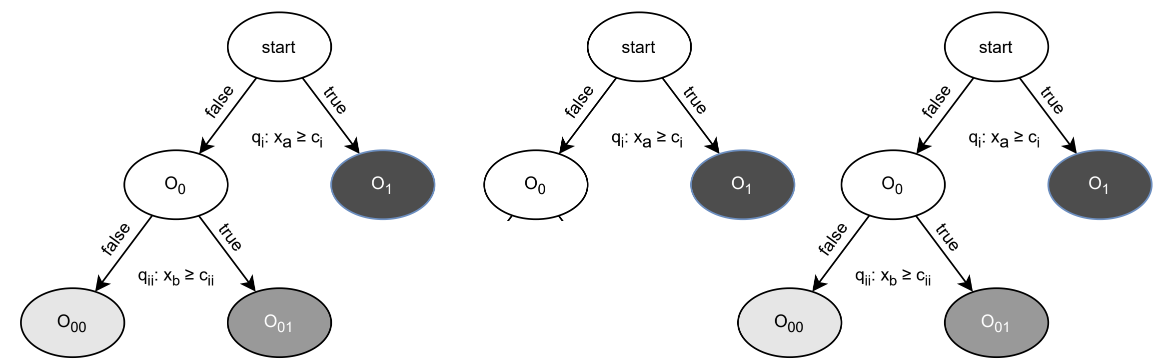
1st time showing

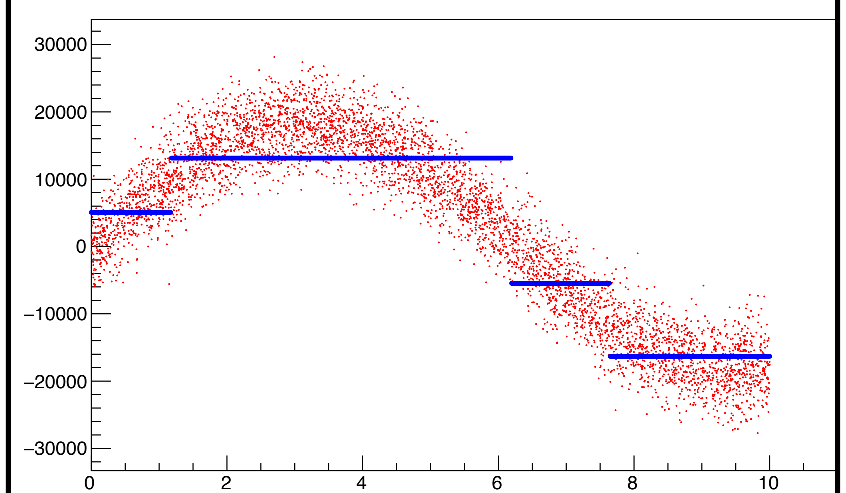
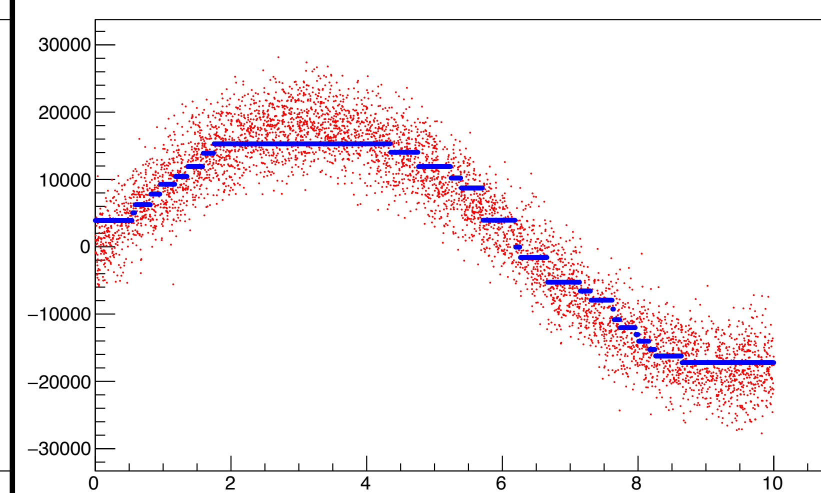
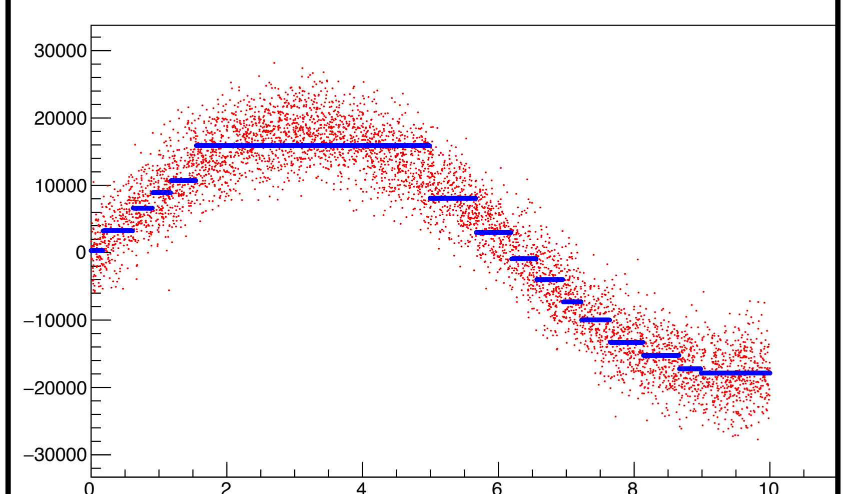
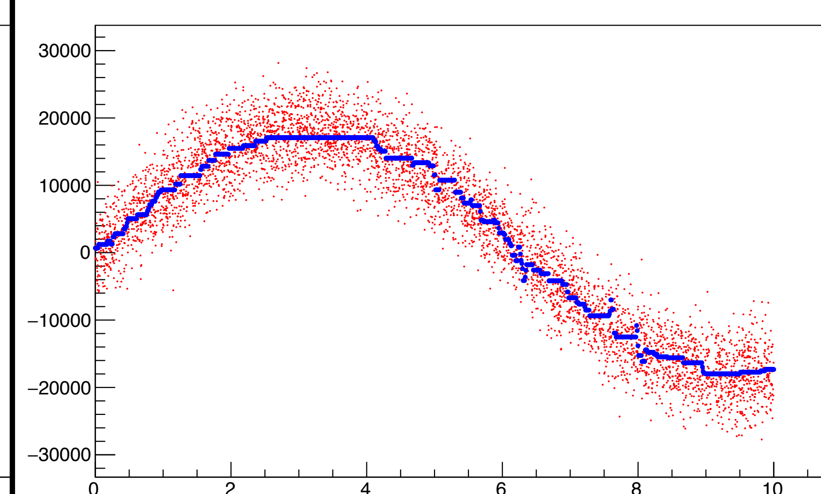
Paper 2: Estimation

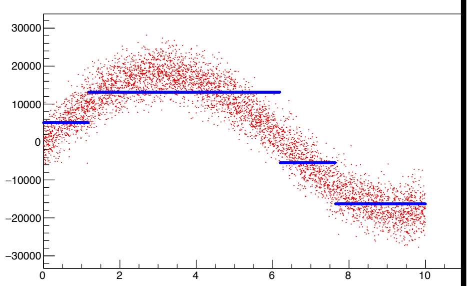
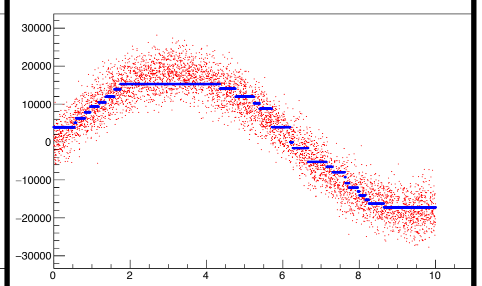
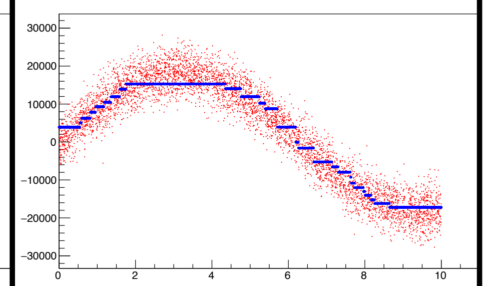
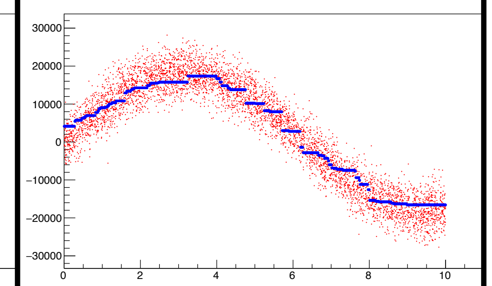
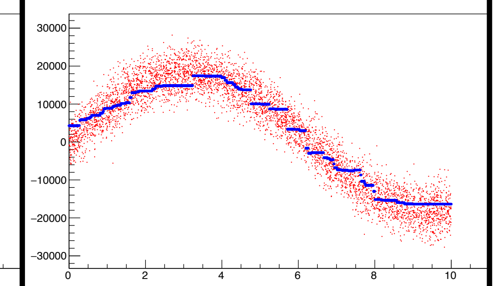
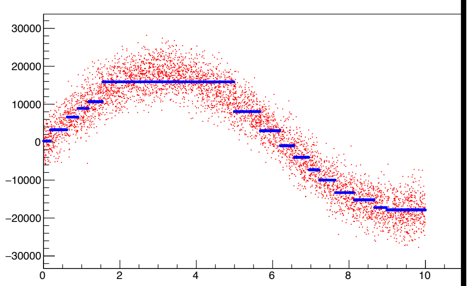
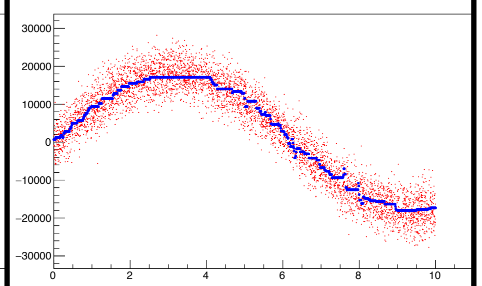
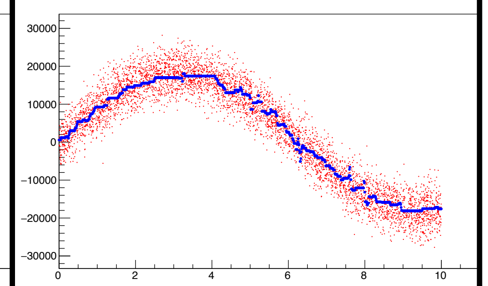
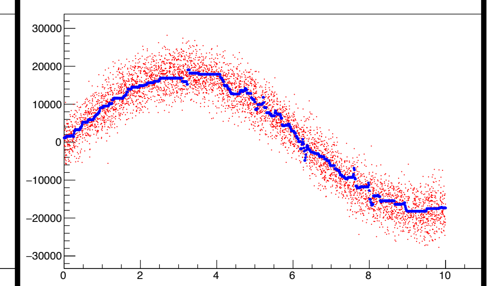
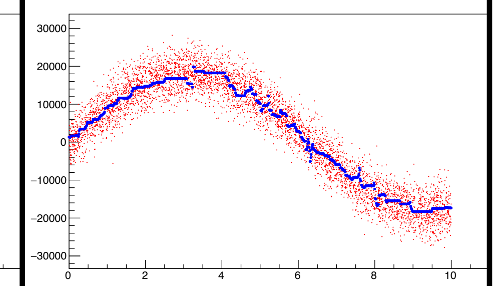
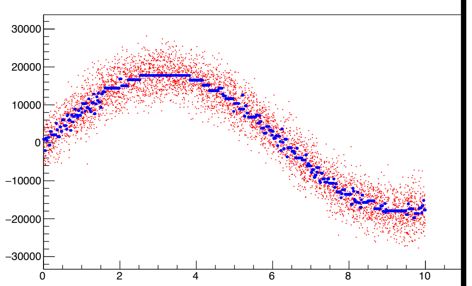
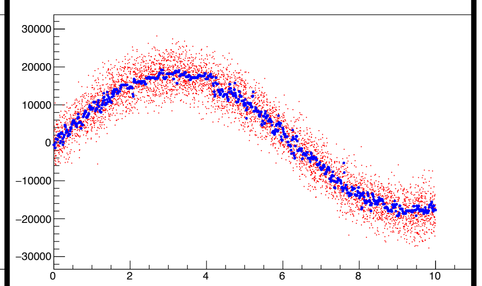
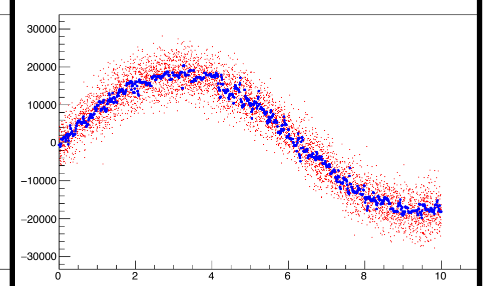
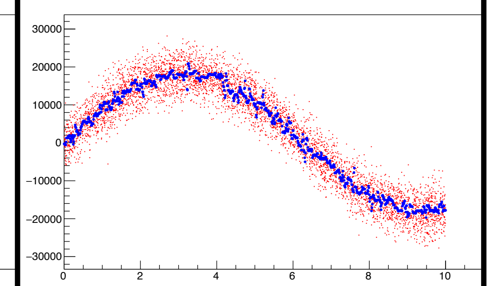
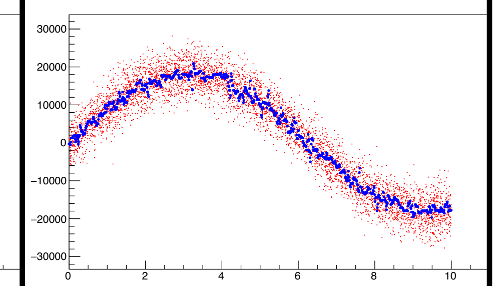
TM Hong

Regression (using BDT)

- Toy problem in 1-d
- Train / test on f(x) = sin(x) + Gaussian(x)
- For sample of x: y = f(x) in 16 bits



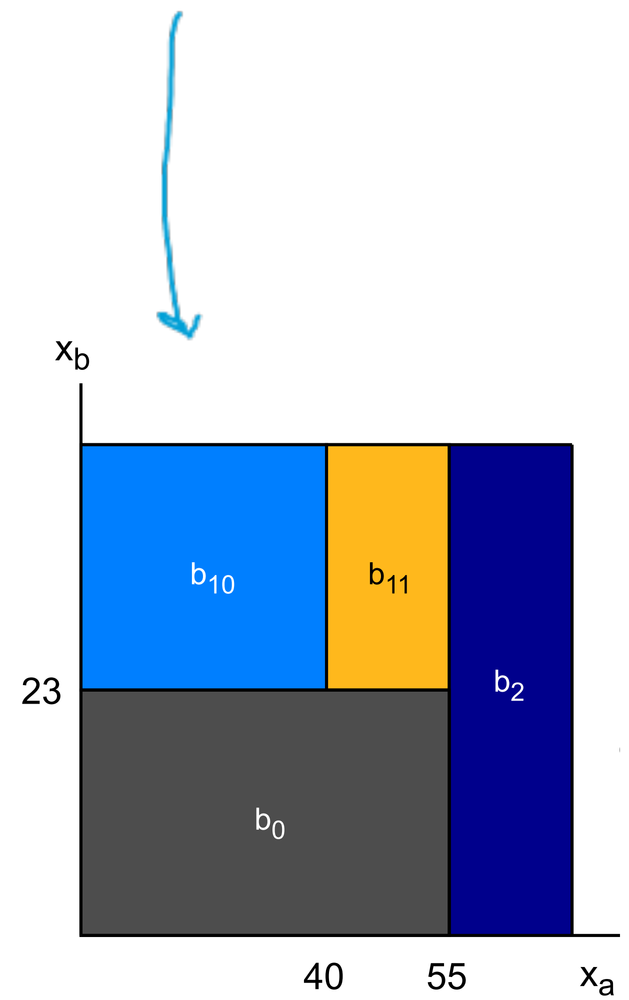
	$N_{tree} = 1$	$N_{tree} = 10$
$N_{depth} = 2$		
$N_{depth} = 4$		

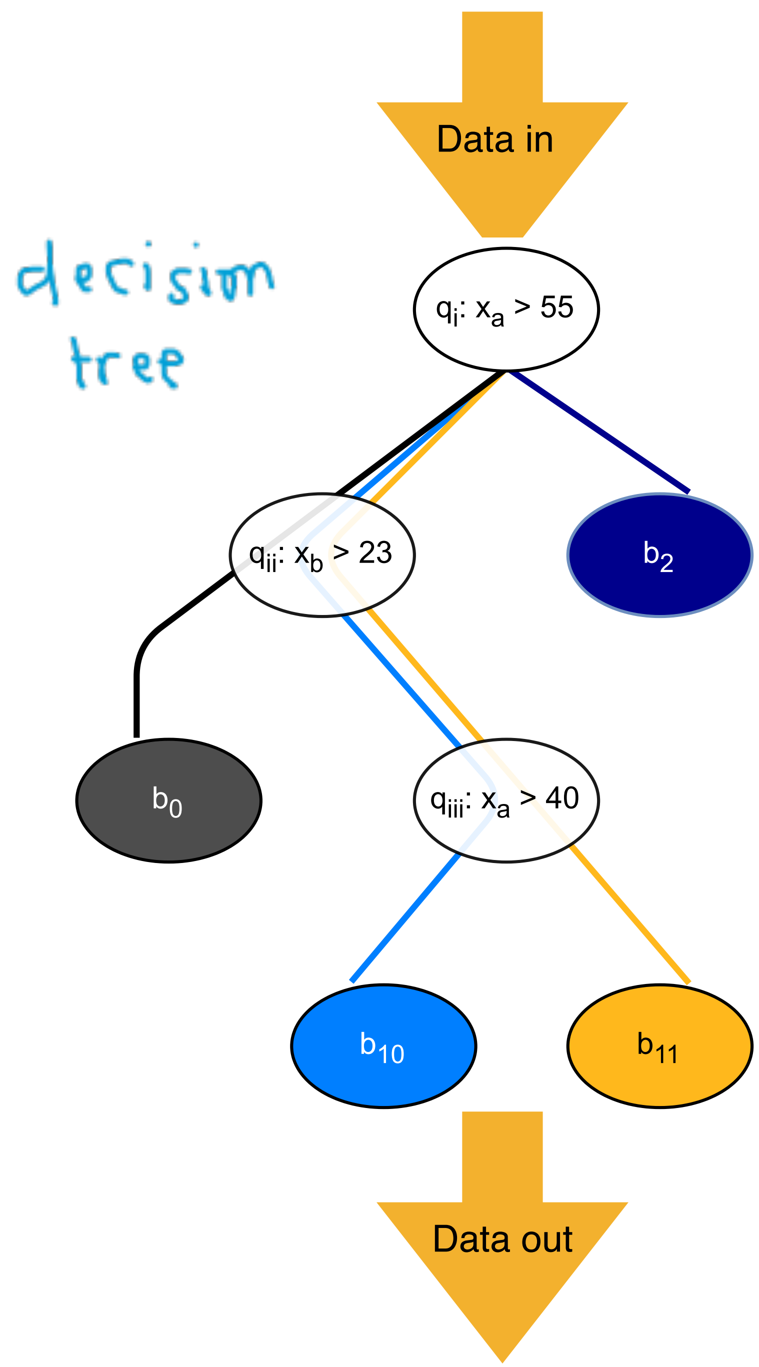
(x,y) = (N_{tree}, N_{depth})	Ntree = 1	10	25	100	400
2					
4					
8					

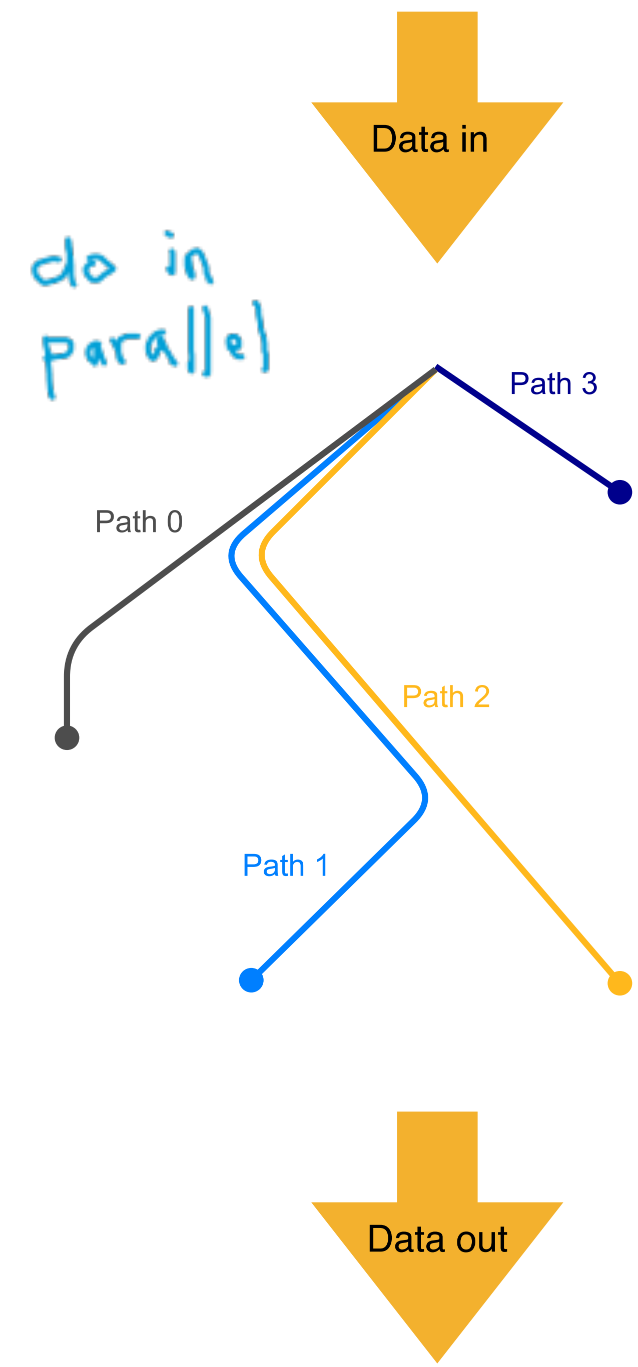
Paper 2: Parallel paths

TM Hong

- Example
  - 2d toy dataset, say x = p_T and y = eta for some SM sample







Destination bin	Decision path
b_0	not(q_i) and not(q_{ii})
b_2	q_i
b_{10}	not(q_i) and q_{ii} and not(q_{iii})
b_{11}	not(q_i) and q_{ii} and q_{iii}

Paper 2: Results

Table 3: Benchmark configuration and the FPGA cost. Three groups of information are given. The top-most group defines the FPGA setup. The second group defines the ML training used for the MET problem and the Nanosecond Optimization. The third group gives the actual results measured on the FPGA for four tree-depth combinations of 40-5, 40-6, 20-7, and 10-8.

Parameter	Value	Comments
FPGA setup		
Chip family	Xilinx Virtex Ultrascale+	
Chip model	xcvu9p-flga2104-2L-e	
Vivado version	2019.2	
Synthesis type	C synthesis	
HLS or RTL	HLS	HLS interface pragma: None
Clock speed	320 MHz	Clock period is 3.125 ns
ML training configuration & Nanosecond Optimization configuration		
ML training method	Boosted decision tree	Regression, Adaptive boosting
No. of input variables	8	
Bin Engine type	Deep Decision Tree Engine (DDTE)	
No. of bits for all variables	16 bits for each	binary integers
FPGA cost for 40 trees, 5 depth		
Latency	6 clock ticks	18.75 ns
Look up tables	1675 out of 1 182 240	0.1% of available
Flip flops	1460 out of 2 364 480	$<$ 0.1% of available
FPGA cost for 40 trees, 6 depth		
Latency	9 clock ticks	28.125 ns
Look up tables	4566 out of 1 182 240	0.4% of available
Flip flops	2516 out of 2 364 480	0.1% of available
FPGA cost for 20 trees, 7 depth		
Latency	15 clock ticks	46.875 ns
Look up tables	4568 out of 1 182 240	0.4% of available
Flip flops	2697 out of 2 364 480	0.1% of available
Block RAM	4.5 out of 4320	0.1% of available
FPGA cost for 10 trees, 8 depth		
Latency	21 clock ticks	65.625 ns
Look up tables	2556 out of 1 182 240	0.2% of available
Flip flops	2299 out of 2 364 480	0.1% of available
Block RAM	5 out of 4320	0.1% of available
Common values for the above configurations		
Interval	1 clock tick	3.125 ns
Block RAM	0 out of 4320	If not listed above
Ultra RAM	0 out of 960	Same for all trees and all depth
Digital signal processors	0 out of 6840	Same for all trees and all depth

Papers

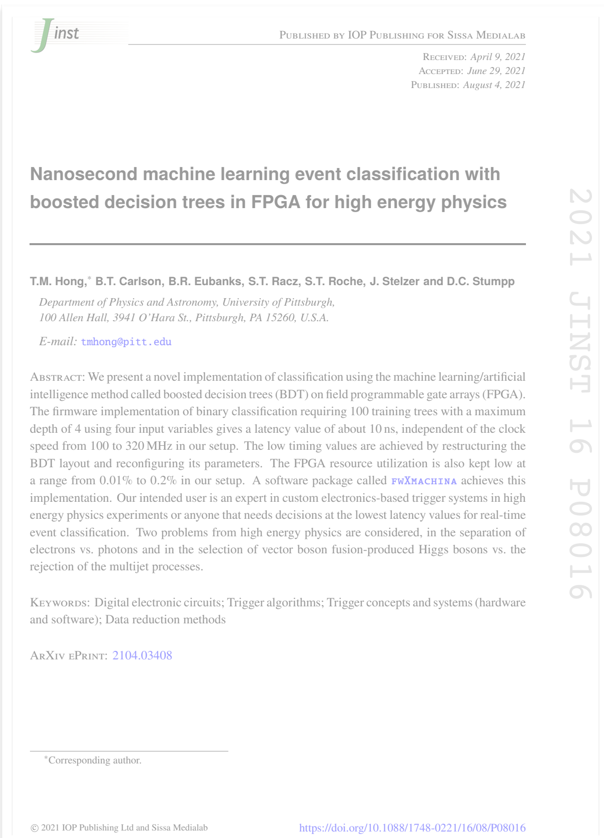
Hong
Pittsburgh

1. Classification
parallel cuts using HLS

2. Regression
parallel paths using HLS

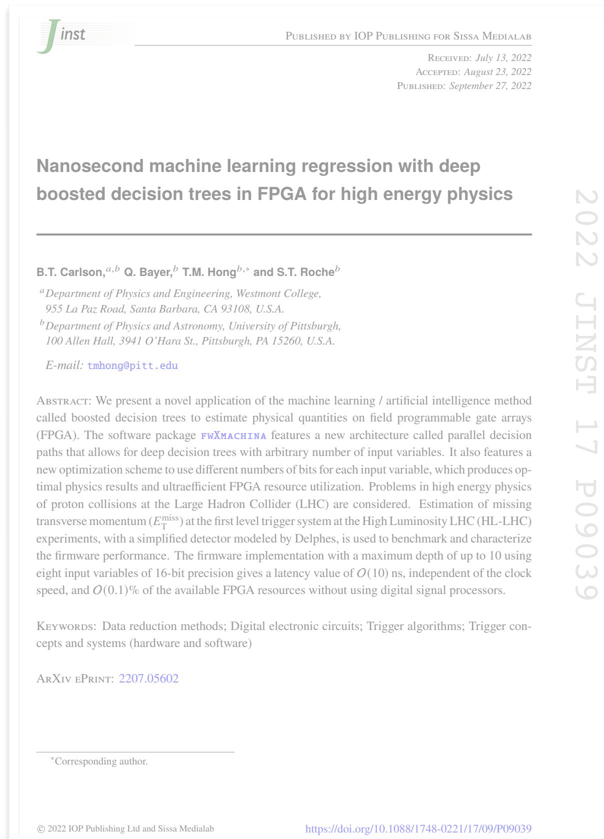
3. Autoencoder
in-house training
bypassing latent space

4. Hardware trees
faster & more efficient
no more HLS



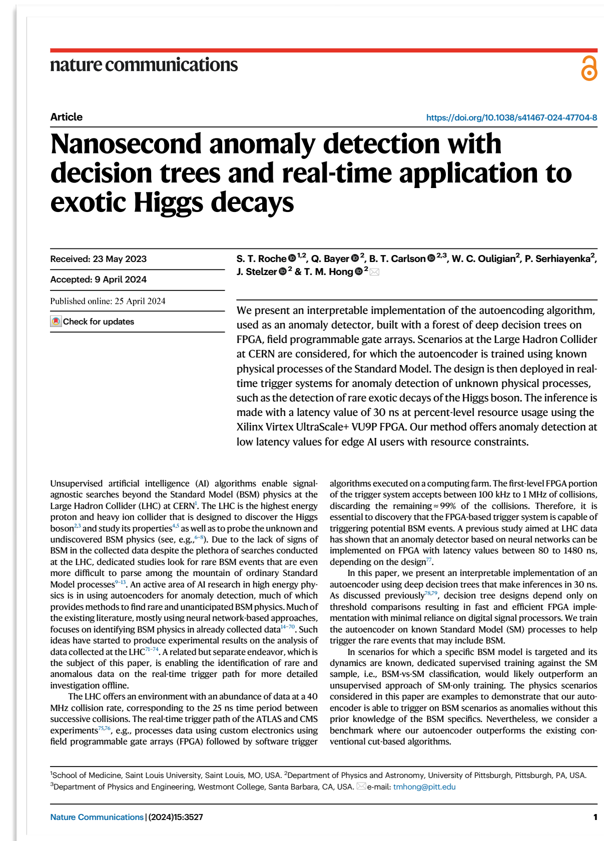
Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics

T.M. Hong, B.T. Carlson, B.R. Eubanks, S.T. Racz, S.T. Roche, J. Stelzer and D.C. Stumpp



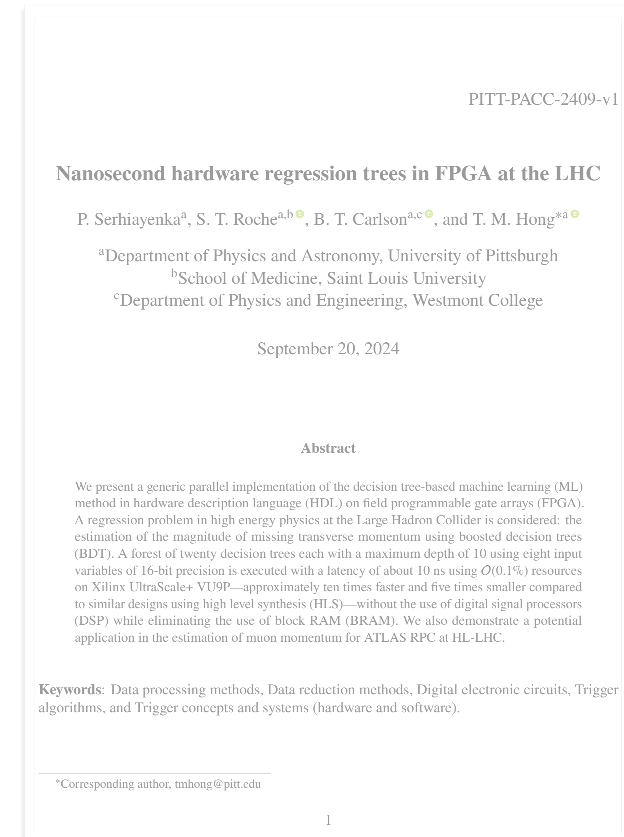
Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics

B.T. Carlson, Q. Bayer, T.M. Hong and S.T. Roche



nature communications

Nanosecond anomaly detection with decision trees and real-time application to exotic Higgs decays



PITT-PACC-2409-v1

Nanosecond hardware regression trees in FPGA at the LHC

P. Serhiayenka, S. T. Roche, B. T. Carlson, and T. M. Hong

Hong et al.
JINST 16, P08016 (2021)
http://doi.org/10.1088/1748-0221/16/08/P08016

Carlson et al.
JINST **17**, P09039 (2022)
http://doi.org/10.1088/1748-0221/17/09/P09039

Roche et al.
Nat. Comm. **15** (2024) 3527
https://arxiv.org/abs/2304.03836

Serhiayenka et al.
Preprint later this week
https://arxiv.org/abs/???

1st time showing

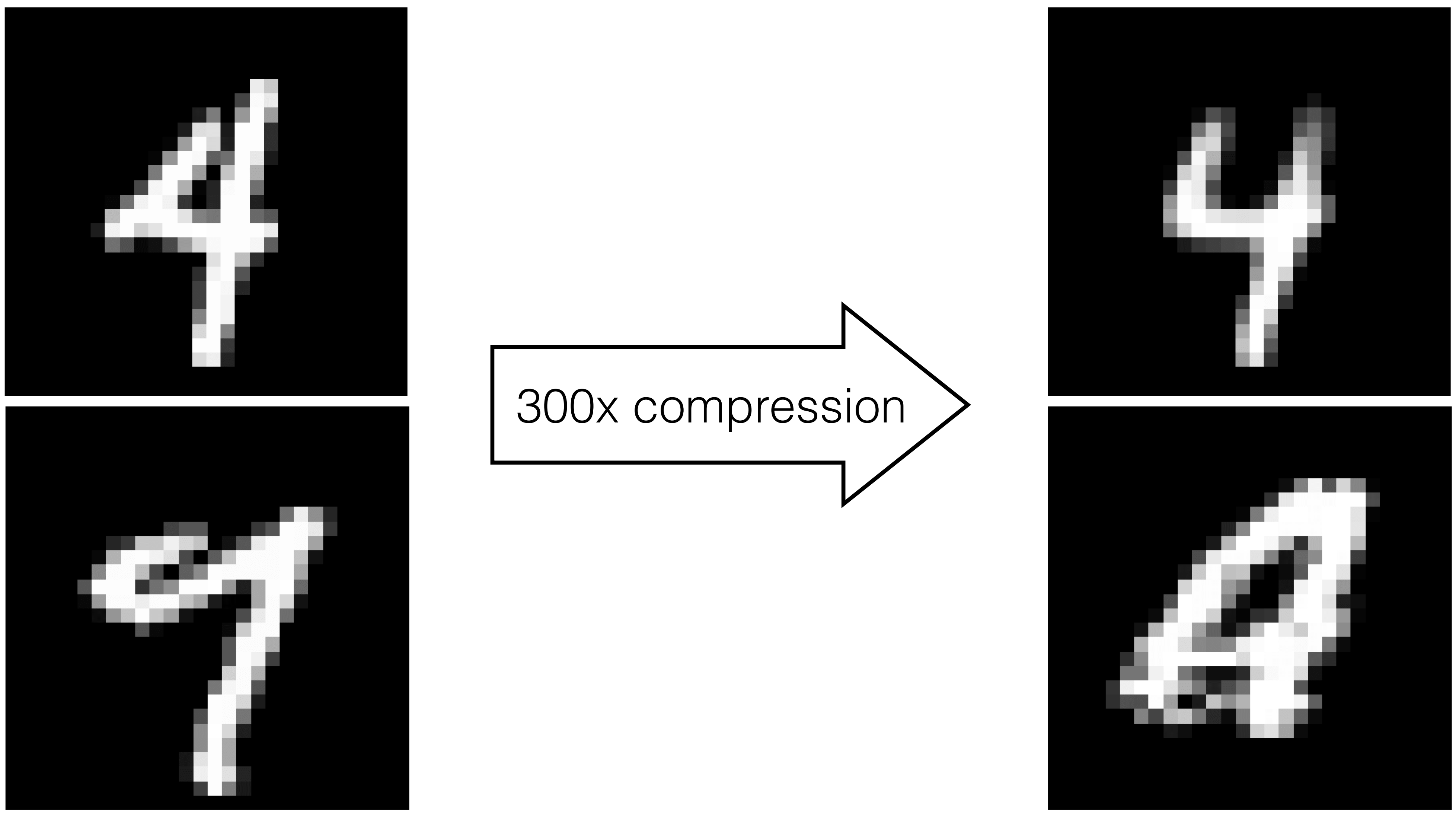
Paper 3: Autoencoder intro

TM Hong

Example: handwritten numbers

- Teach it 0, 1, 2, 3, 4 with a sample (doesn't know about 9!)

784 variables (8-bit) 1 variable (20 bit) 784 variables (8-bit)



Details

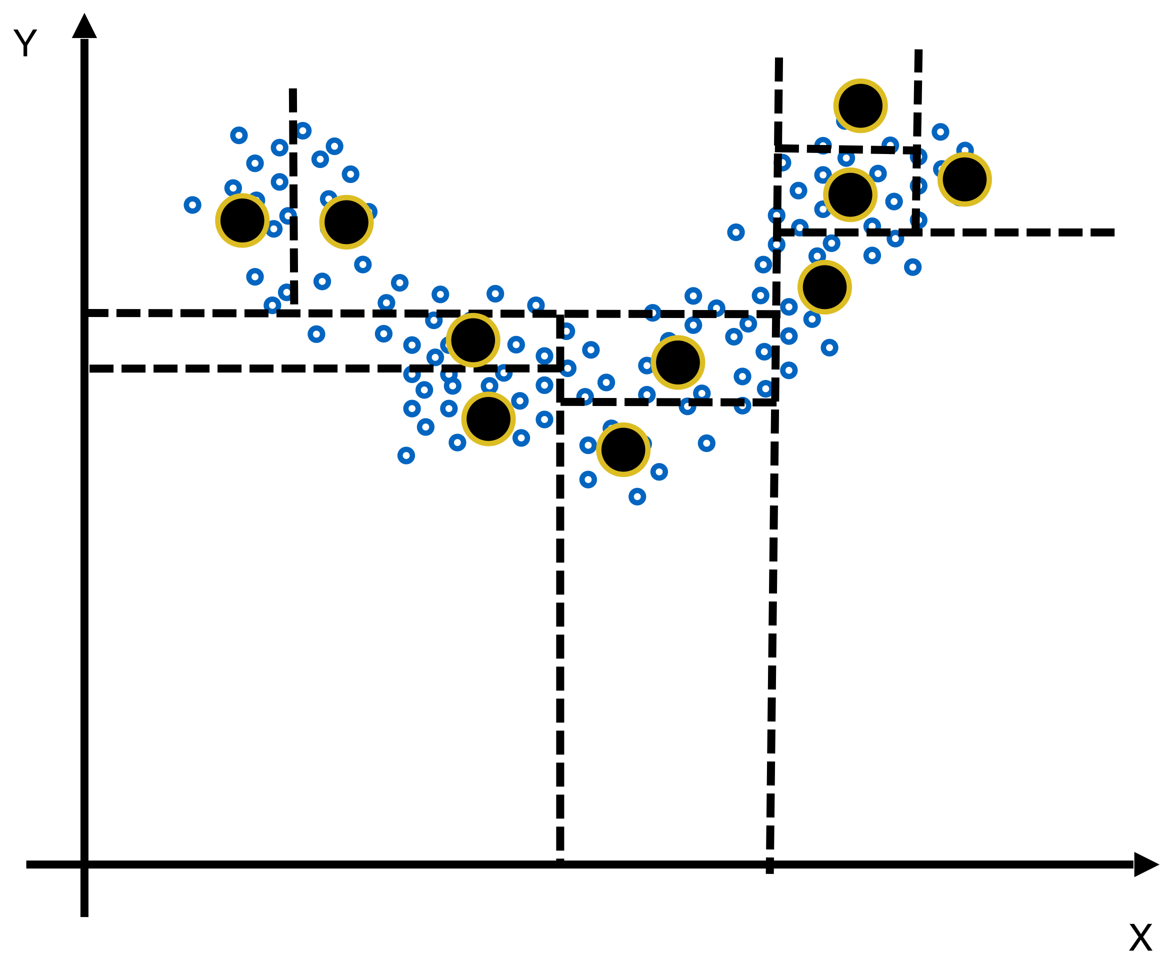
- Input-output distance is relatively small = good compression
- Input-output distance is relatively large = bad compression

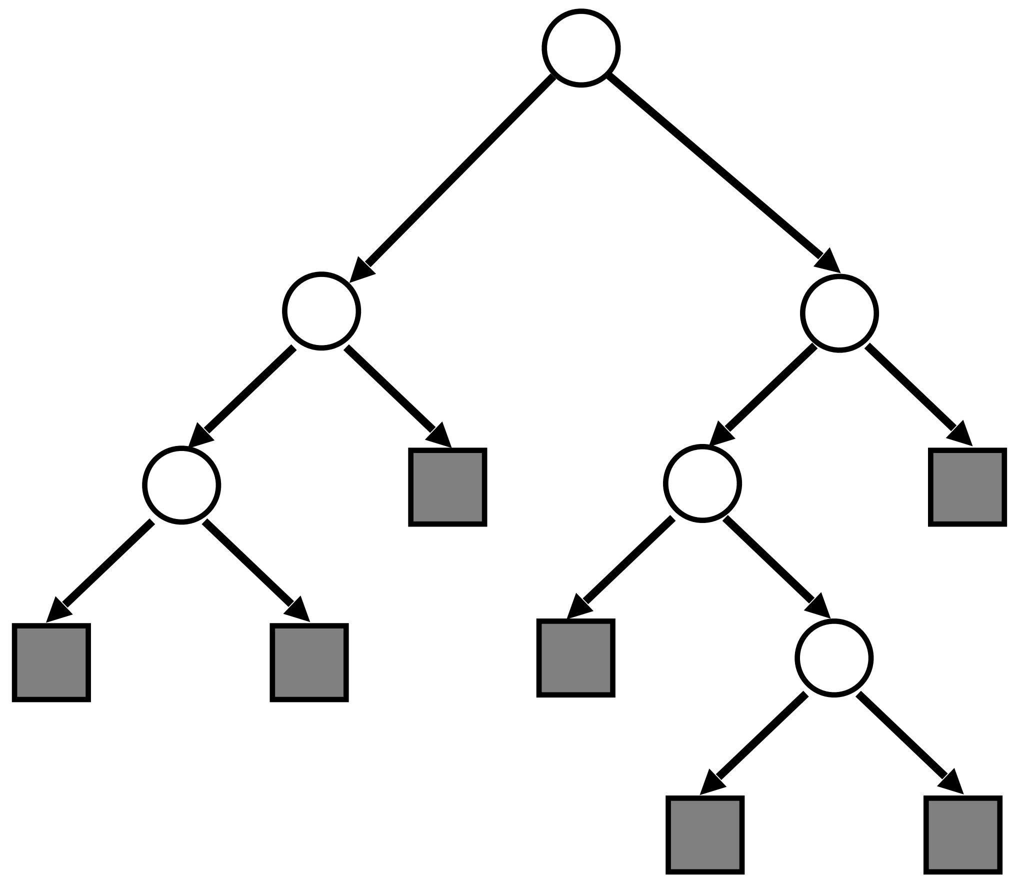
Train by sampling 1d projections

- Encoding: Event → which bin it's in

Decode by returning a "reconstruction point"

- Decoding: Bin → median of the training data in bin



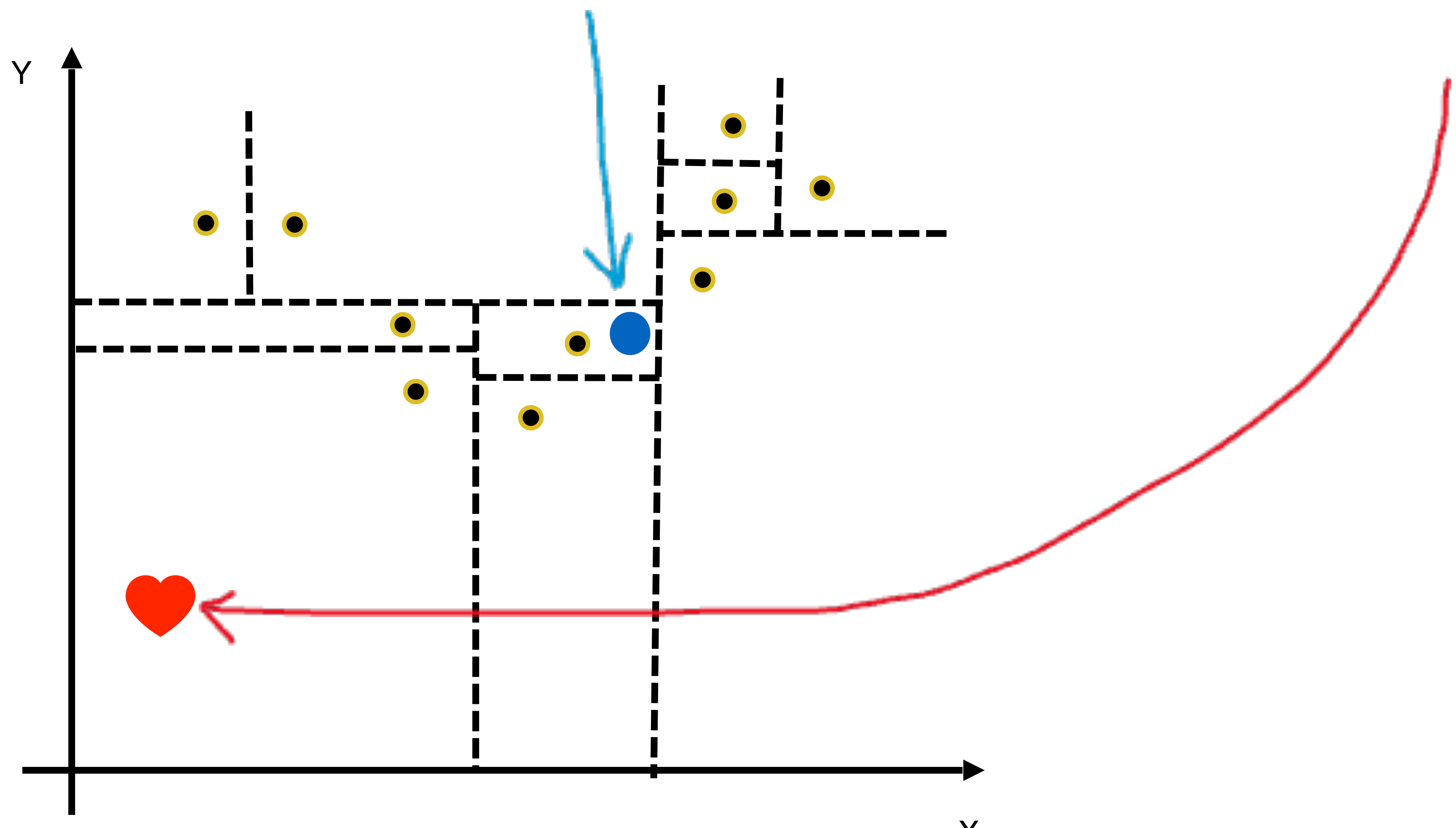


How does this detect anomalies?

- Define: Distance between input – output = anomaly score

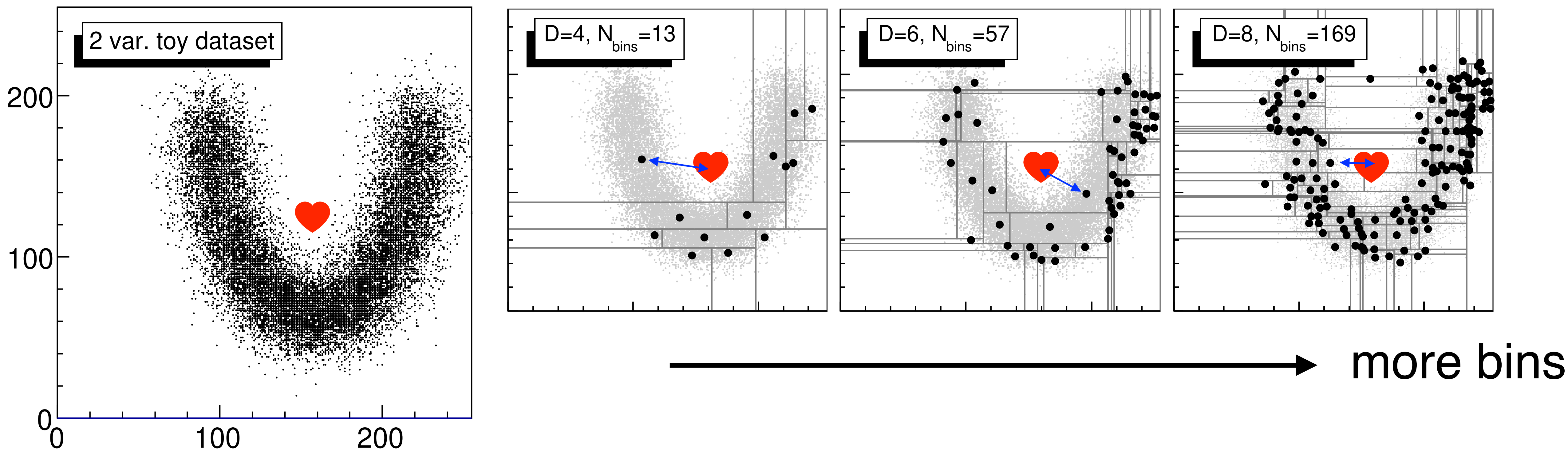
- Non-anomaly
  - Input is similar to training data
  - Will likely land in a small bin → close to the reconstruction point

- Anomaly
  - Input is not similar to training data
  - Will likely land in a large bin → far from the reconstruction point



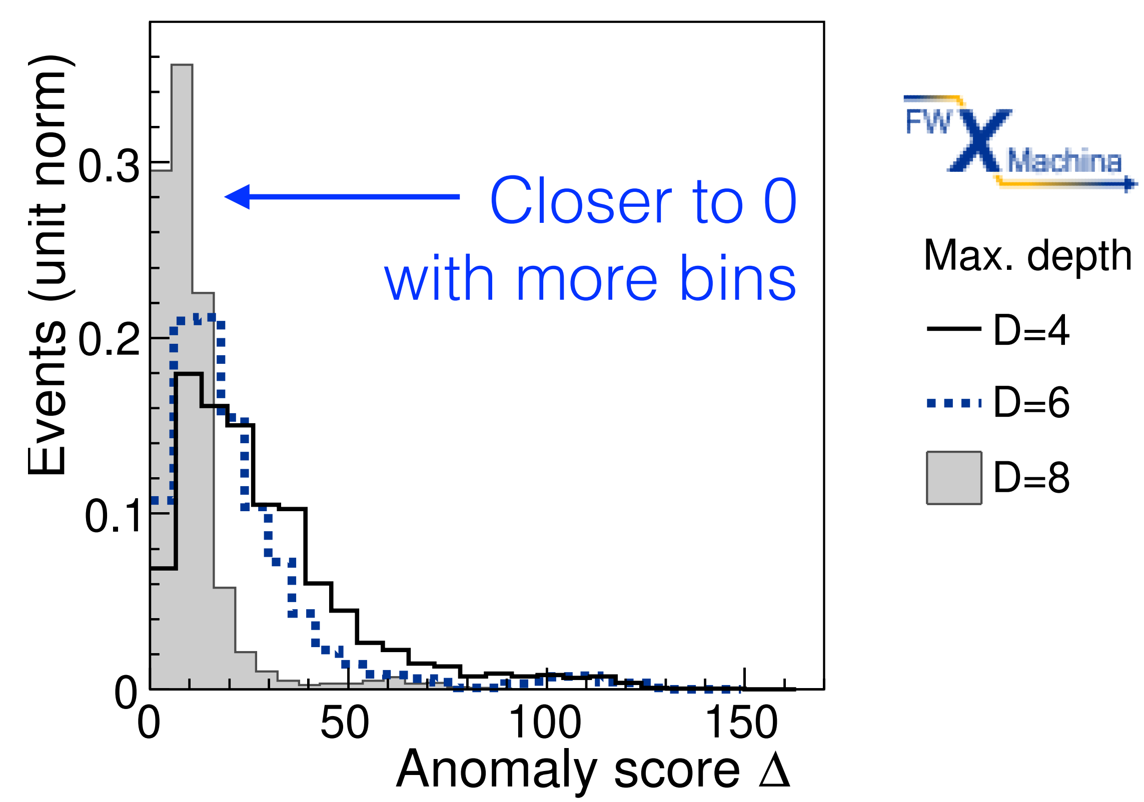
Paper 3: Toy dataset (2 variables)

TM Hong



Anomaly score

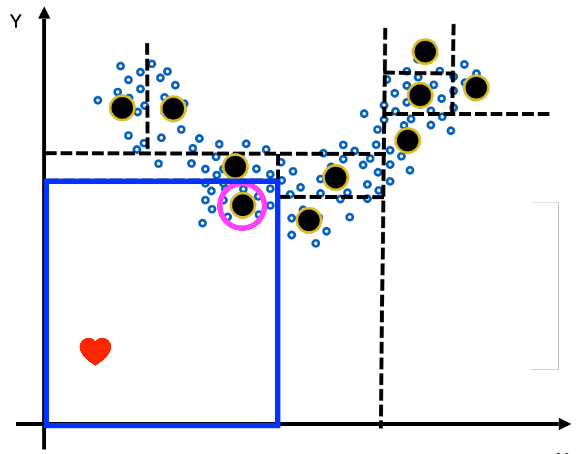
- Feed back in the training sample
  - Should be near 0, like E_T^{miss} resolution

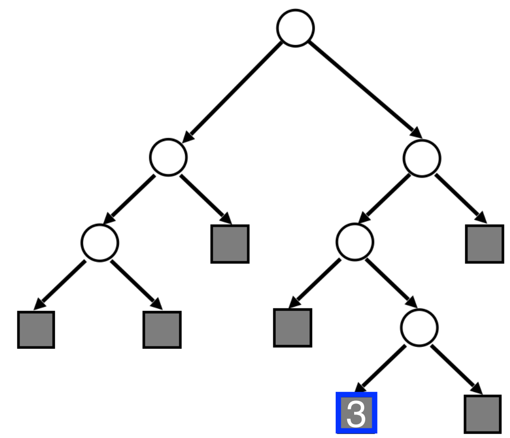


Paper 3: Skip latent space

Don’t need latent space in firmware

- Closer look at what it means to encode





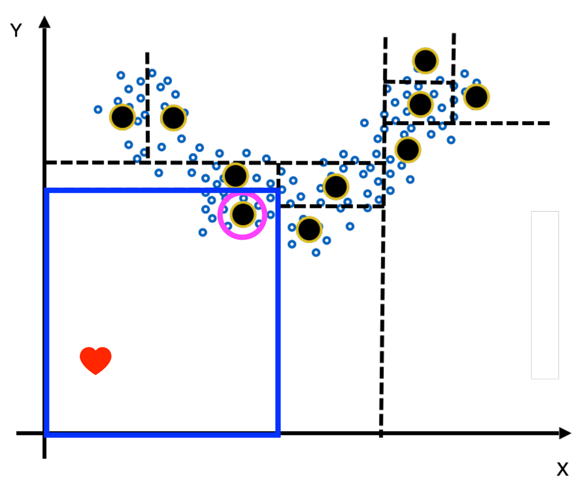
Bin	x	y
1	...	...
2	...	...
3	5	4
...		

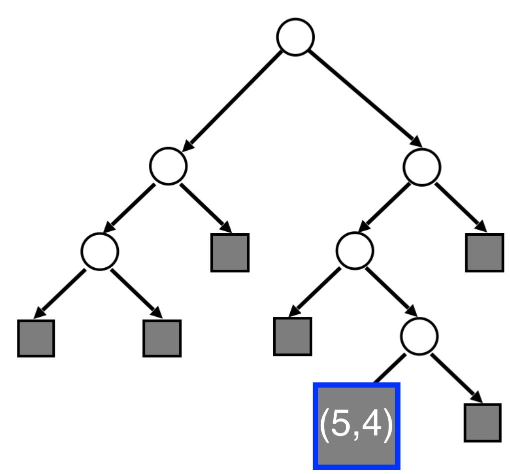
Incoming heart

Encode: return bin 3

Decode bin 3: return (5,4)

- Skip the encoding & decoding



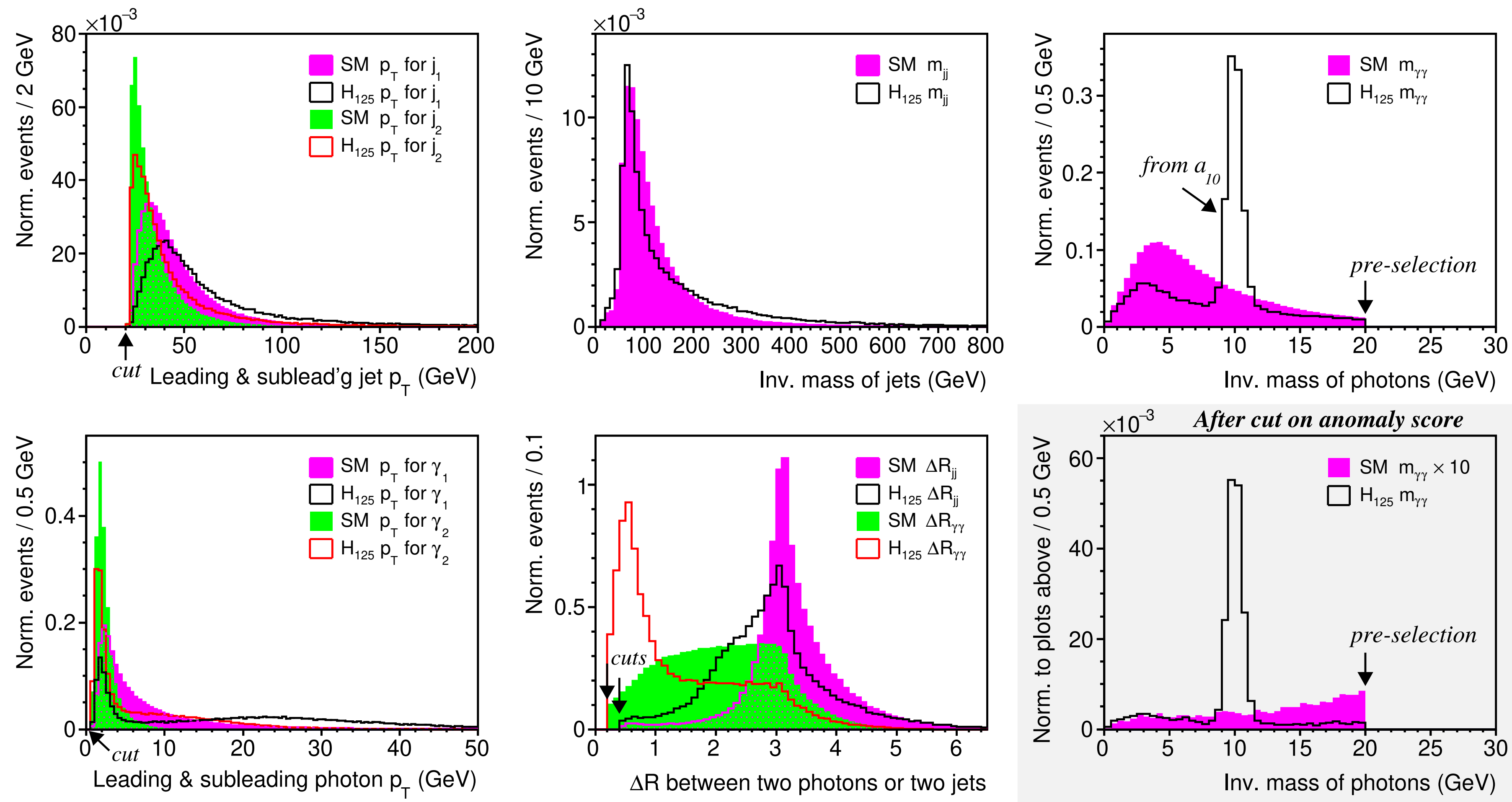


Incoming heart

Encode is Decode: return (5,4)

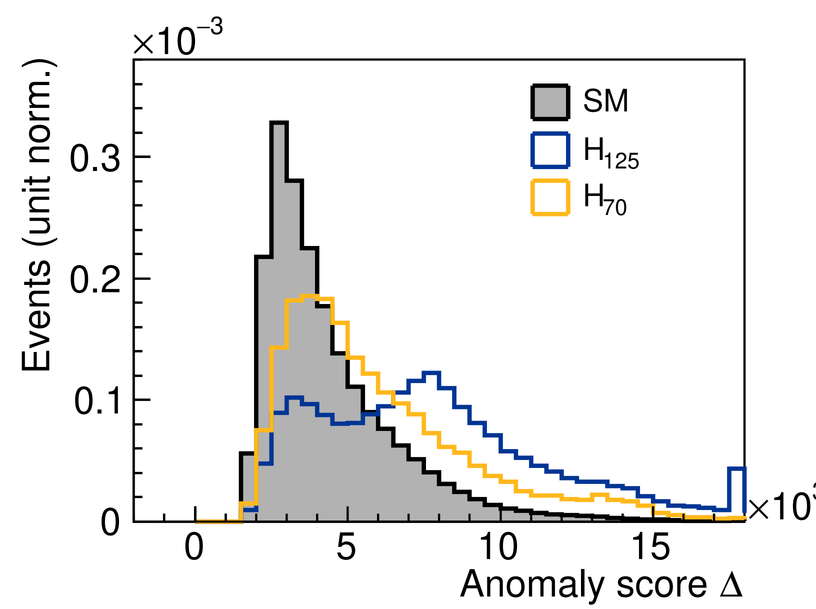
Inputs

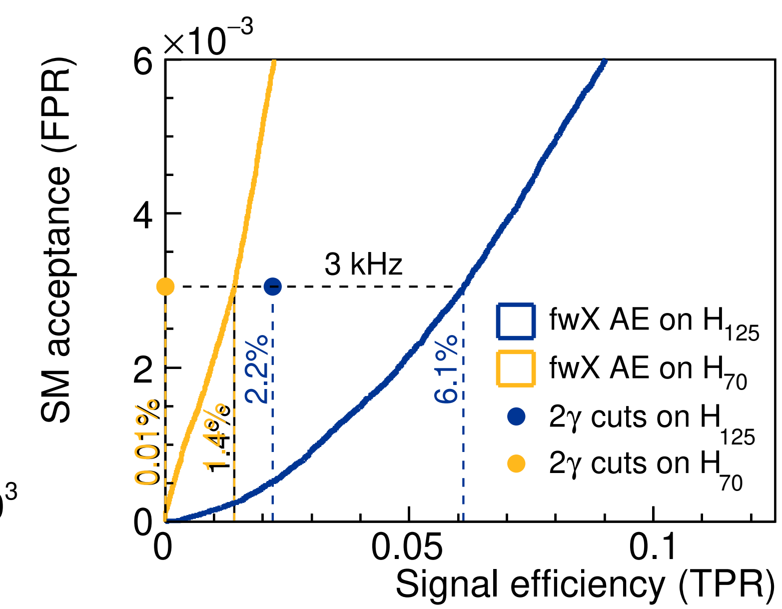
- Sample
  - MadGraph5_aMC 2.9.5
  - Hadron'n+Shower: Pythia8
  - Detector: Delphes 3.5.0, CMS
- Variables
  - 8 inputs: jets, photons, ΔR



Results

- Compare
  - vs. 3 kHz Run-2 ATLAS rate
- Better
  - 3x gain in signal



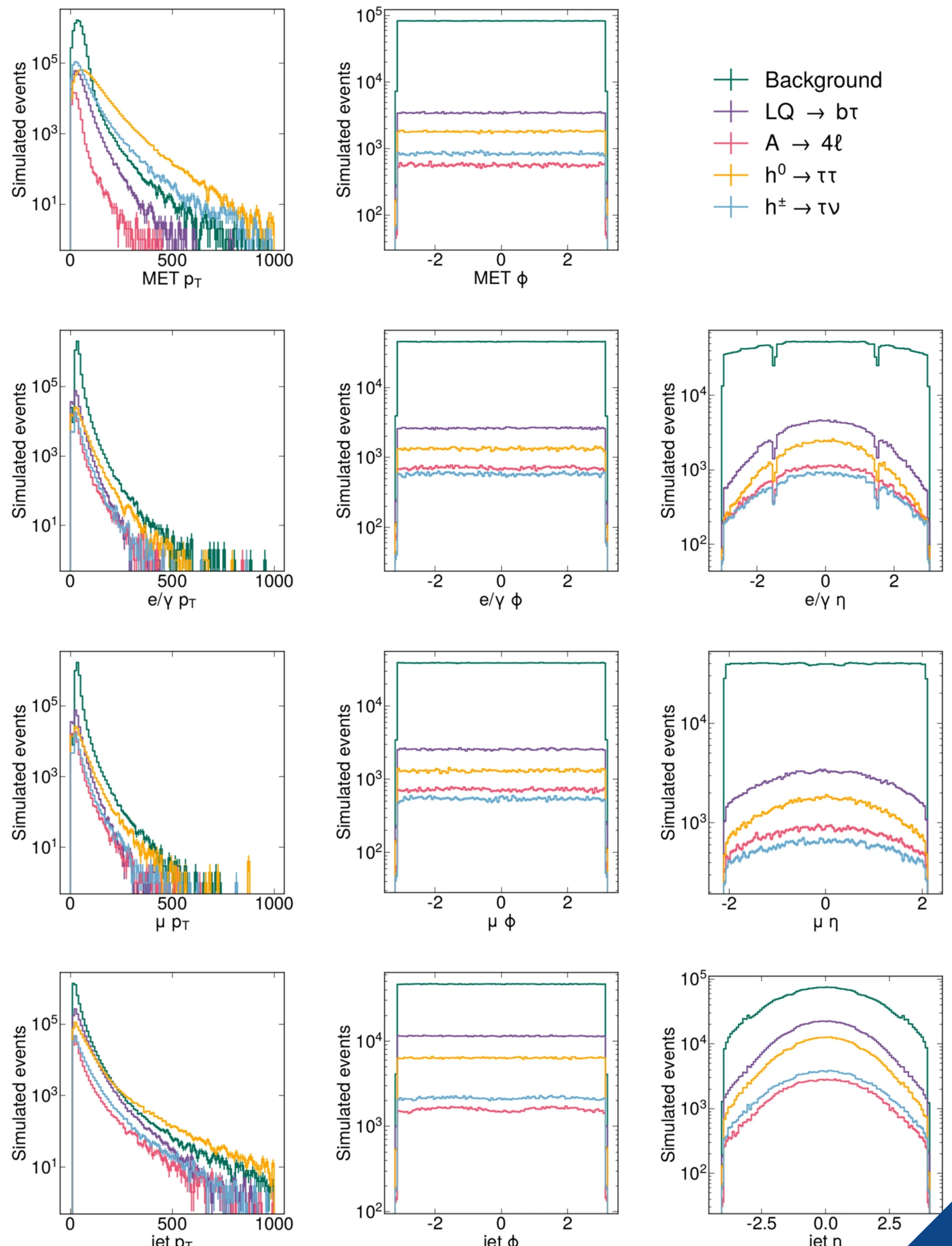


Paper 3: Compare with hls4ml

LHC anomaly detection ds

[Sci Data 9, 118]

- Background
  - W → lν, Z → ll, multijet, ttbar
- Signal
  - 4 BSM scenarios
- Input variables
  - 54 variables
  - p_T, η, ϕ of the 4 leading μ, 4 leading e, 10 leading jets, MET
  - See distributions on the right
- Sample selection
  - Require ≥1 lepton w/ p_T > 23 GeV
  - (L1 will already save these...)



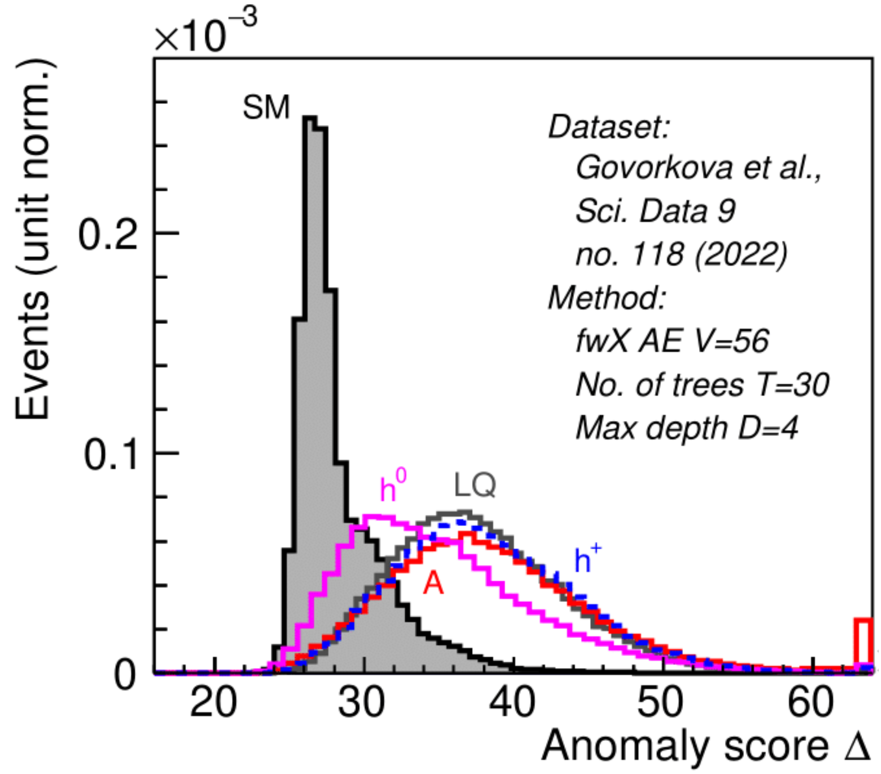
Works well

- Physics (plots)
- FPGA (table)

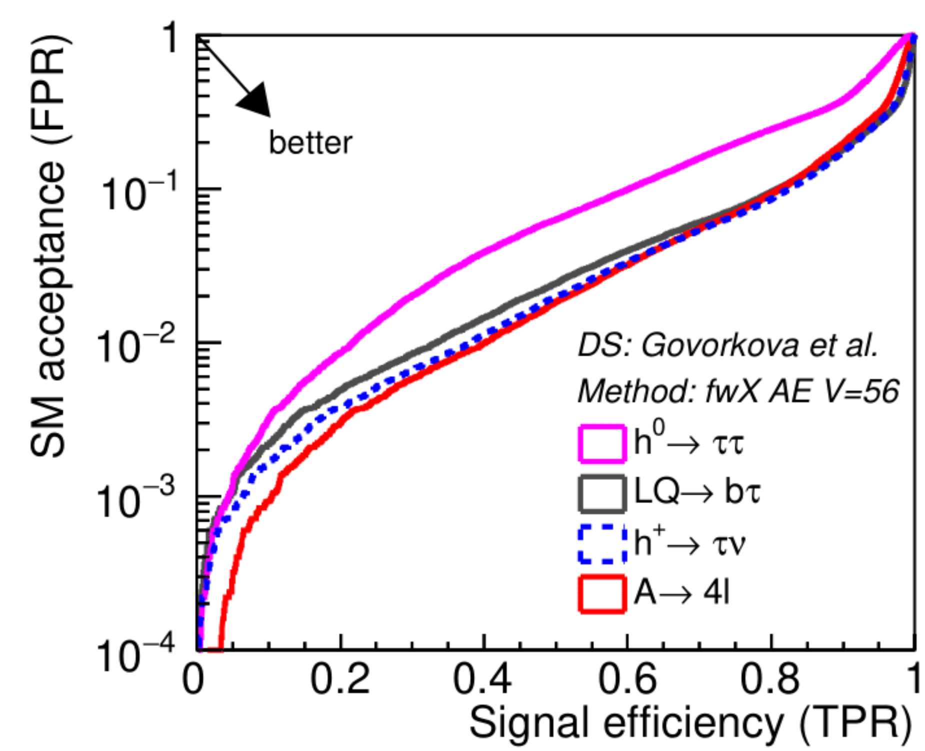
Comparison

- Hls4ml NN-AE
  [Nature Mach. Intell. 4 (2022) 154–161]
- Physics: comparable AUC
- FPGA results

Distribution



ROC curve

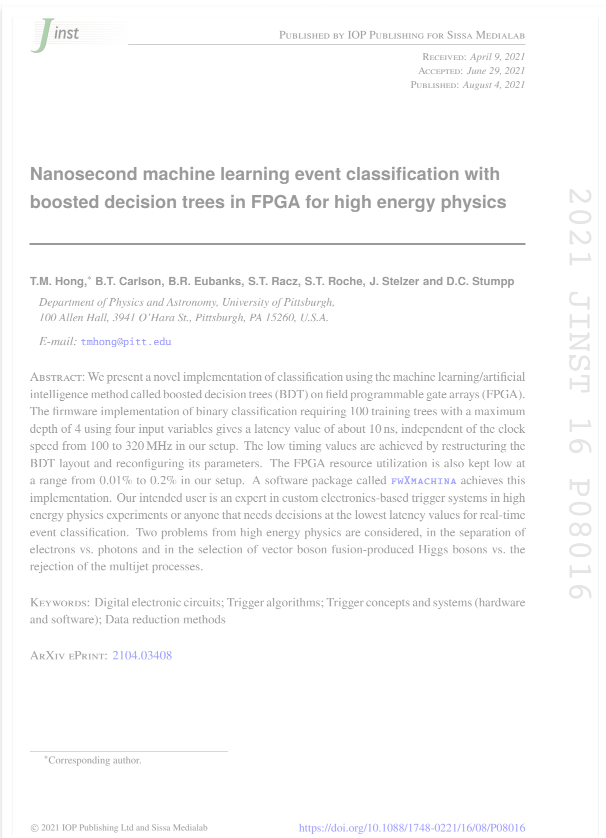


	hls4ml	fwX (this)
Clock speed	200 MHz	200 MHz
Latency	80 ns	30 ns
Interval	5 ns	5 ns
FF	0.5%	0.6%
LUT	3%	9%
DSP	1%	0.8%
BRAM	0.3%	0

Papers

Hong
Pittsburgh

1.Classification	2.Regression	3.Autoencoder	4.Hardware trees
parallel cuts using HLS	parallel paths using HLS	in-house training bypassing latent space	faster & more efficient no more HLS



JINST

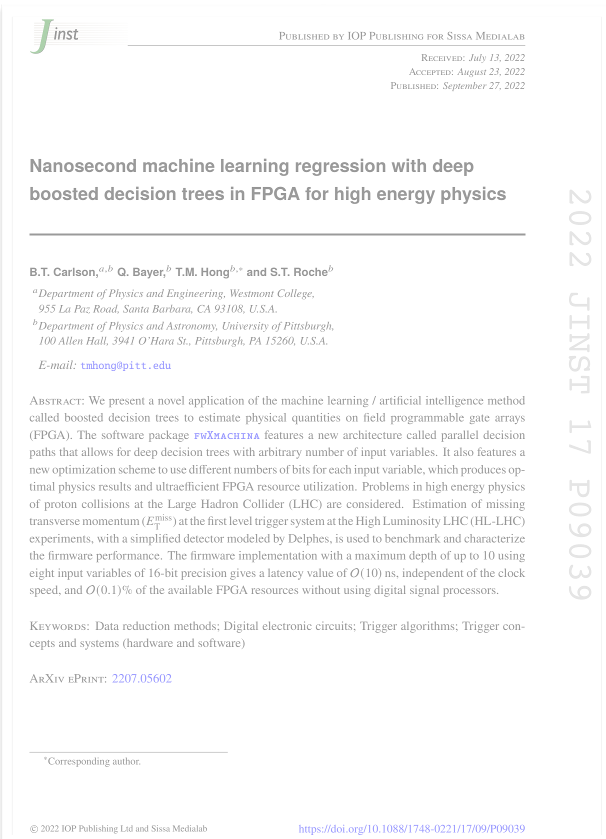
Published by IOP Publishing for Sissa Medialab

Received: April 9, 2021
Accepted: June 29, 2021
Published: August 4, 2021

Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics

T.M. Hong,* B.T. Carlson, B.R. Eubanks, S.T. Racz, S.T. Roche, J. Stelzer and D.C. Stumpp

2021 JINST 16 P08016



JINST

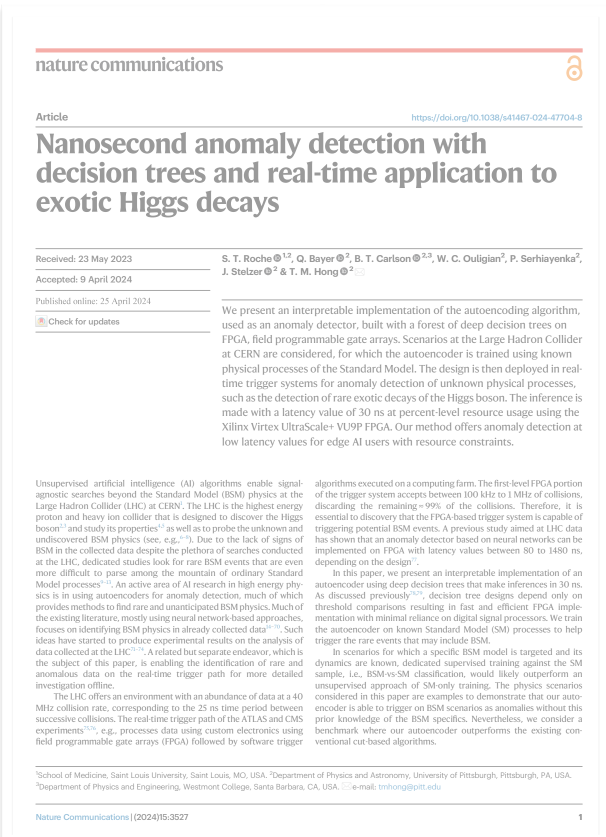
Published by IOP Publishing for Sissa Medialab

Received: July 13, 2022
Accepted: August 23, 2022
Published: September 27, 2022

Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics

B.T. Carlson, Q. Bayer, T.M. Hong and S.T. Roche

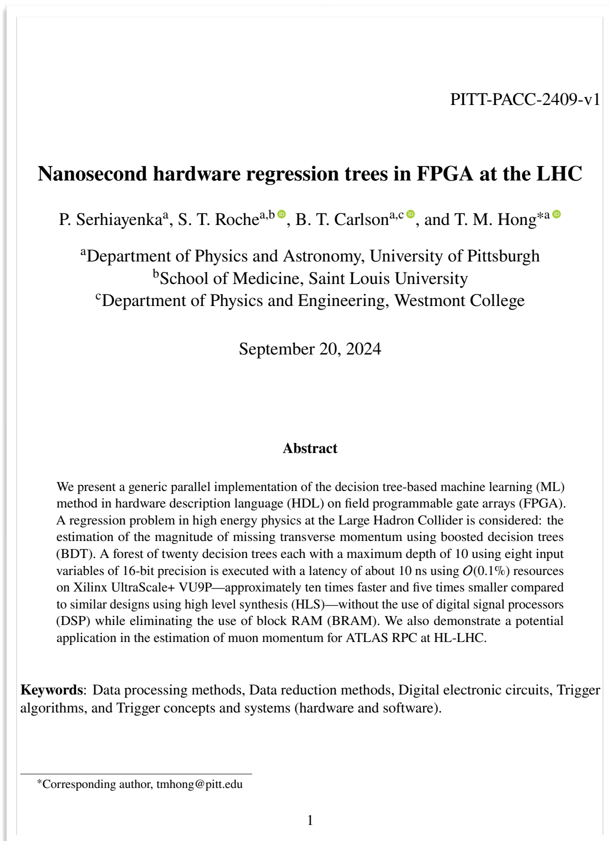
2022 JINST 17 P09039



nature communications

Article

Nanosecond anomaly detection with decision trees and real-time application to exotic Higgs decays



PITT-PACC-2409-v1

Nanosecond hardware regression trees in FPGA at the LHC

P. Serhiayenka[a], S. T. Roche[a,b], B. T. Carlson[a,c], and T. M. Hong*[a]

[a]Department of Physics and Astronomy, University of Pittsburgh
[b]School of Medicine, Saint Louis University
[c]Department of Physics and Engineering, Westmont College

September 20, 2024

Abstract

We present a generic parallel implementation of the decision tree-based machine learning (ML) method in hardware description language (HDL) on field programmable gate arrays (FPGA). A regression problem in high energy physics at the Large Hadron Collider is considered: the estimation of the magnitude of missing transverse momentum using boosted decision trees (BDT). A forest of twenty decision trees each with a maximum depth of 10 using eight input variables of 16-bit precision is executed with a latency of about 10 ns using $O(0.1\%)$ resources on Xilinx UltraScale+ VU9P—approximately ten times faster and five times smaller compared to similar designs using high level synthesis (HLS)—without the use of digital signal processors (DSP) while eliminating the use of block RAM (BRAM). We also demonstrate a potential application in the estimation of muon momentum for ATLAS RPC at HL-LHC.

Keywords: Data processing methods, Data reduction methods, Digital electronic circuits, Trigger algorithms, and Trigger concepts and systems (hardware and software).

*Corresponding author, tmhong@pitt.edu

1

Hong et al. JINST 16, P08016 (2021) http://doi.org/10.1088/1748-0221/16/08/P08016	Carlson et al. JINST **17**, P09039 (2022) http://doi.org/10.1088/1748-0221/17/09/P09039	Roche et al. Nat. Comm. **15** (2024) 3527 https://arxiv.org/abs/2304.03836	Serhiayenka et al. Preprint later this week https://arxiv.org/abs/???

1st time showing

Summary

- Python to write VHDL

Table 1: FPGA results and comparison with Refs. [7, 8, 11]. All results in the table uses the same FPGA model Xilinx Ultrascale+ VU9P (vu9p-flgb2104-2L-e) with the following available resources 1.2 M LUT, 2.4 M FF, 6.8 k DSP, and 4.3 k BRAM. Effective depth d is defined as so that $2^d = N_{\text{bin}}/N_{\text{tree}}$.

Goal	5 classif'n	2 classif'n	E_T^{miss} regression		E_T^{miss} regression			
Reference	[11]	[7]	[8]		This paper			
Setup								
Design	VHDL	HLS	HLS	HLS	VHDL	VHDL	VIDL	VHDL
Sum strategy	-	-	-	-	pipeline	combin.	combin.	pipeline
Parallelize	-	cutwise	pathwise	pathwise	pathwise	pathwise	pathwise	pathwise
Clock (MHz)	250	320	320	320	320	320	200	320
Bit precision	fixed_{18}	int_8	int_{16}	int_{16}	int_{16}	int_{16}	int_{16}	int_{16}
N_{var}	16	4	8	8	8	8	8	8
N_{tree}	100	100	40	10	40	10	20	100
Max. depth D	4	4	6	8	6	8	10	12
N_{bin}	-	-	1.7 k	1.4 k	1.7 k	1.4 k	2.9 k	15.7 k
Effective depth d	-	-	5.4	7.2	5.4	7.2	7.2	7.3
Notable				identical	identical		slower clock	larger forest
Results								
LUT	96 k	1 k	6.4 k	75 k	5.1 k	10 k	15.5 k	38 k
FF	43 k	0.1 k	35 k	24 k	1.6 k	4.7 k	6.6 k	19.4 k
DSP	0	2	0	0	0	0	0	0
BRAM	0	5.5	0	10	0	0	0	0
URAM	-	0	0	0	0	0	0	0
Latency (ns)	52 ns	9.375 ns	38 ns	119 ns	25 ns	19 ns	10 ns	28 ns
″ (tick)	13	3	12	38	8	6	2	9
Interval (tick)	1	1	1	1	1	1	1	1
Notable				benchmark			in abstract	

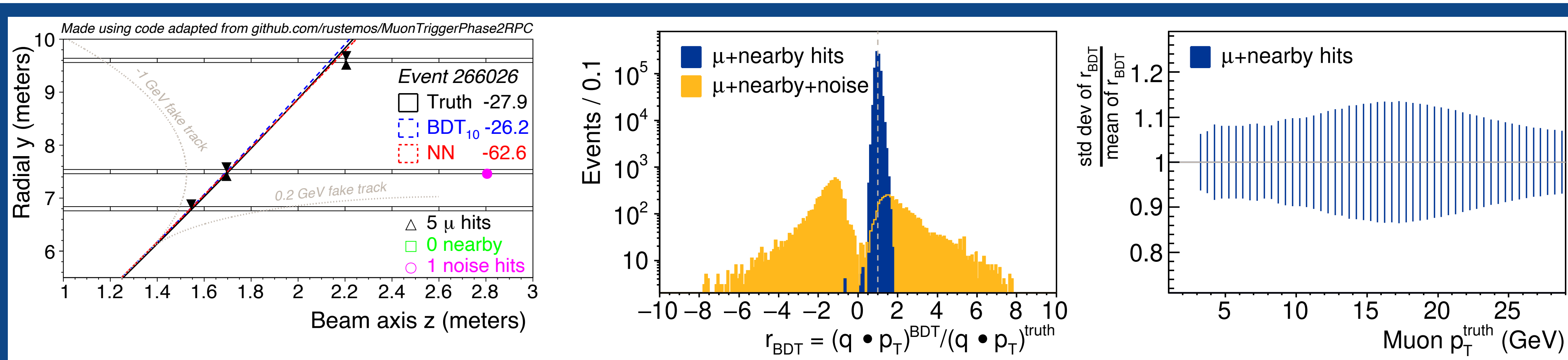
Results

- 5x smaller
- 10x faster

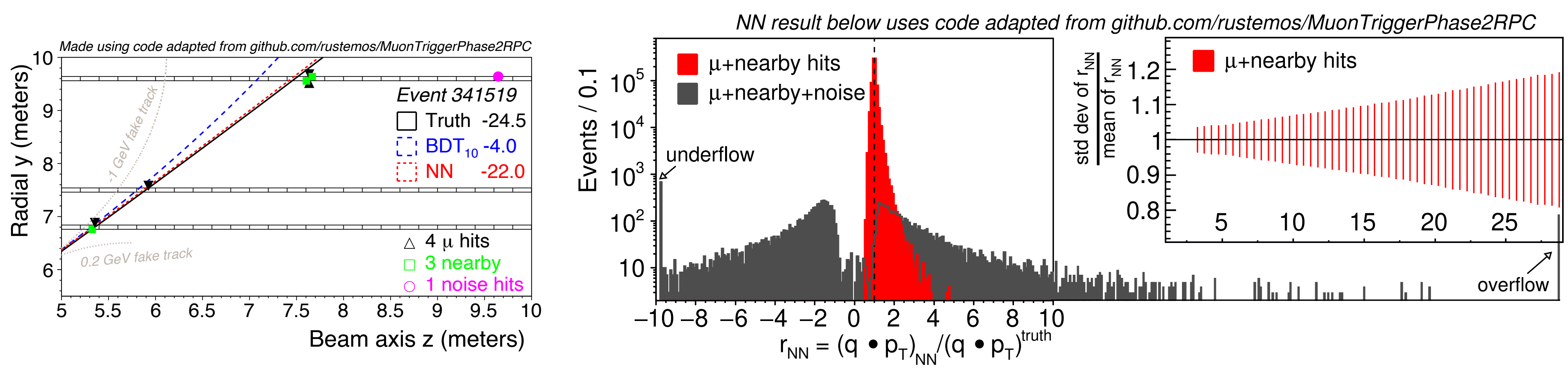
Test case

- Mock-up ATLAS RPC for Phase-2

R. Ospanov, C. Feng, W. Dong, W. Feng, and S. Yang, *Development of FPGA-based neural network regression models for the ATLAS Phase-II barrel muon trigger upgrade*, Eur. Phys. J. Web of Conf. **251**, 04031 (2021).



BDT



NN

Results

- NN core is sharper
- BDT tail is shorter

Outline

Introduction

- Who are we
- What products do we offer
- Summary if you have to leave

Theory

- Parallelizing decision trees
- Classification & regression
- Anomaly detection w/ autoencoder

Practice

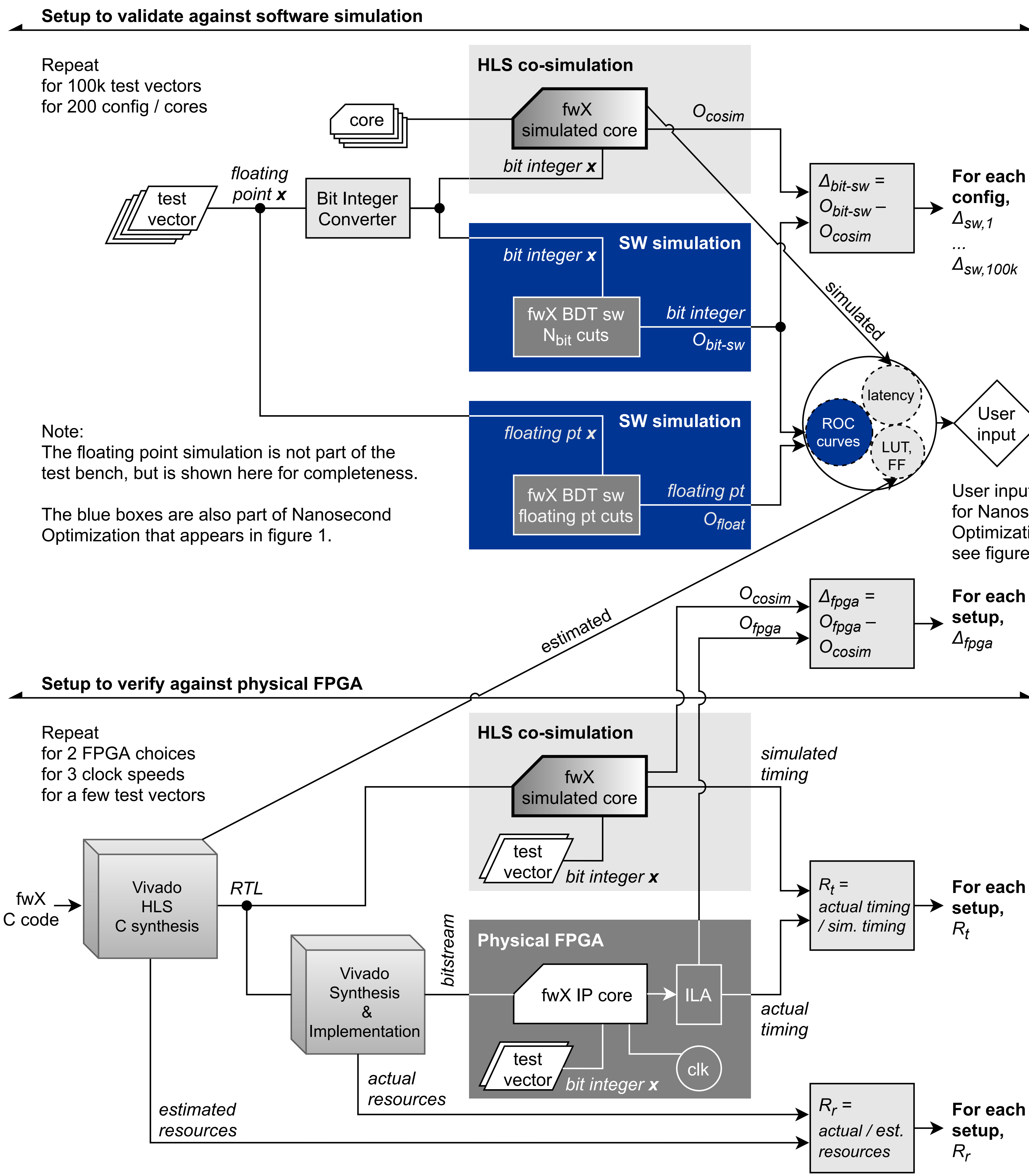
- Hands-on tutorial videos with "teaching assistants" in person & in Zoom breakout rooms

Test bench setup

TM Hong

Philosophy

- Every training ships with test vectors
- Every design creates its own testbench
- Performance values from implementation, not estimate



Compared

- Estimated usage / latency vs. actual usage / latency

Table 12: FPGA cost verification against physical FPGA. Comparison of the FPGA cost using the bitstream on the FPGA (actual), simulated timing using co-simulation and estimated resources using Vivado HLS (estimated). The actual-to-estimated ratios are given as *R*. Two FPGA choices and three clock speeds are considered; the 320 MHz group of columns represent the benchmark clock. For all other configurable parameters, see table 1. The timing values are reported in units of clock ticks. The Xilinx Vivado version used for the actual and estimated columns are noted. For the ratios, “1” signifies no difference.

Parameter	Benchmark FPGA									Smaller FPGA		
FPGA setup												
Family	Xilinx Virtex Ultrascale+									Xilinx Artix-7		
Model	xcvu9p-flga2104-2L-e									xc7z020-clg400-1		
Speed	320 MHz			200 MHz			100 MHz			100 MHz		
Period	3.125 ns			5 ns			10 ns			10 ns		
Vivado	2019.2	2019.2		2018.2	2018.2		2018.2	2018.2		2019.1	2019.2	
FPGA cost	actual	/ estim.	= *R*	actual	/ estim.	= *R*	actual	/ estim.	= *R*	actual	/ estim.	= *R*
Latency	3	/ 3	= 1	2	/ 2	= 1	1	/ 1	= 1	4	/ 4	= 1
Interval	1	/ 1	= 1	1	/ 1	= 1	1	/ 1	= 1	1	/ 1	= 1
LUT	717	/ 1903	= 0.4	717	/ 4015	= 0.2	717	/ 4007	= 0.2	482	/ 3572	= 0.1
FF	147	/ 138	= 1.1	147	/ 113	= 1.3	147	/ 2	= 73.	245	/ 362	= 0.7
BRAM	5.5	/ 8	= 0.7	5.5	/ 15	= 0.4	5.5	/ 15	= 0.4	7.5	/ 15	= 0.5
URAM	0	/ 0	= 1	0	/ 0	= 1	0	/ 0	= 1	NA	/ NA	= NA
DSP	2	/ 0	= NA	2	/ 2	= 1	2	/ 2	= 1	2	/ 2	= 1

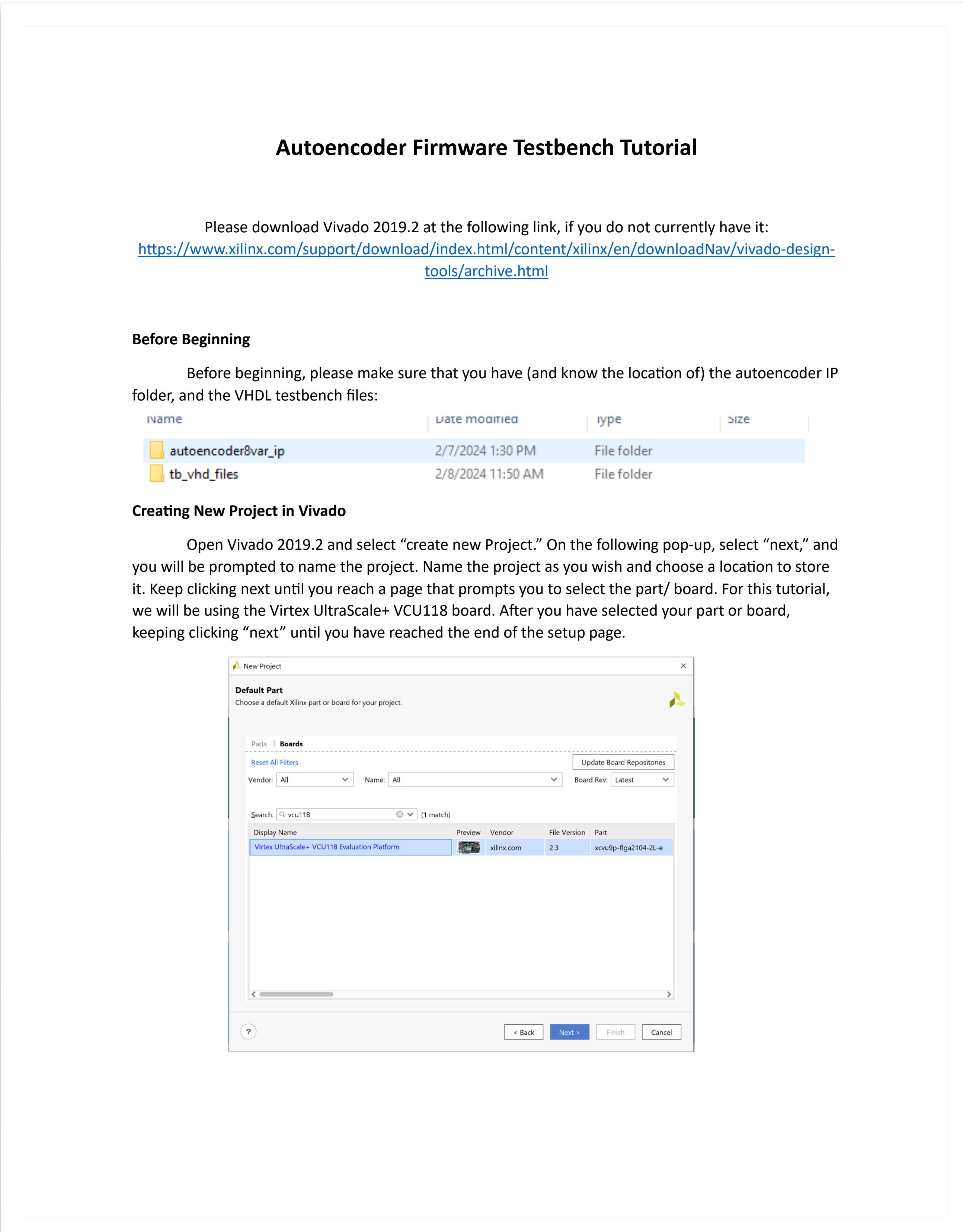
- Not always 1

FW testbench w/ IP available

TM Hong

http://d-scholarship.pitt.edu/45784/

Screenshots in the document



Autoencoder Firmware Testbench Tutorial

Please download Vivado 2019.2 at the following link, if you do not currently have it:
https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html

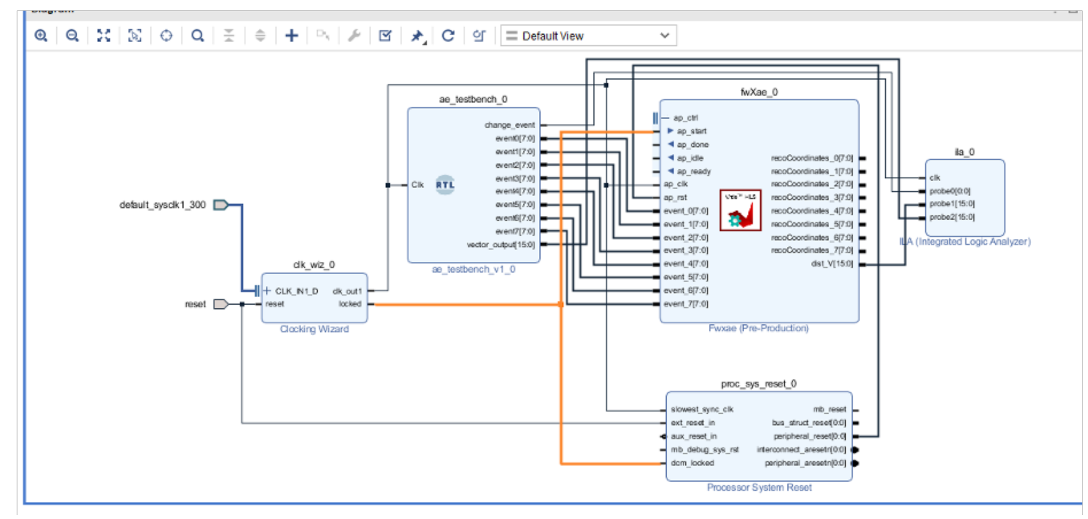
Before Beginning

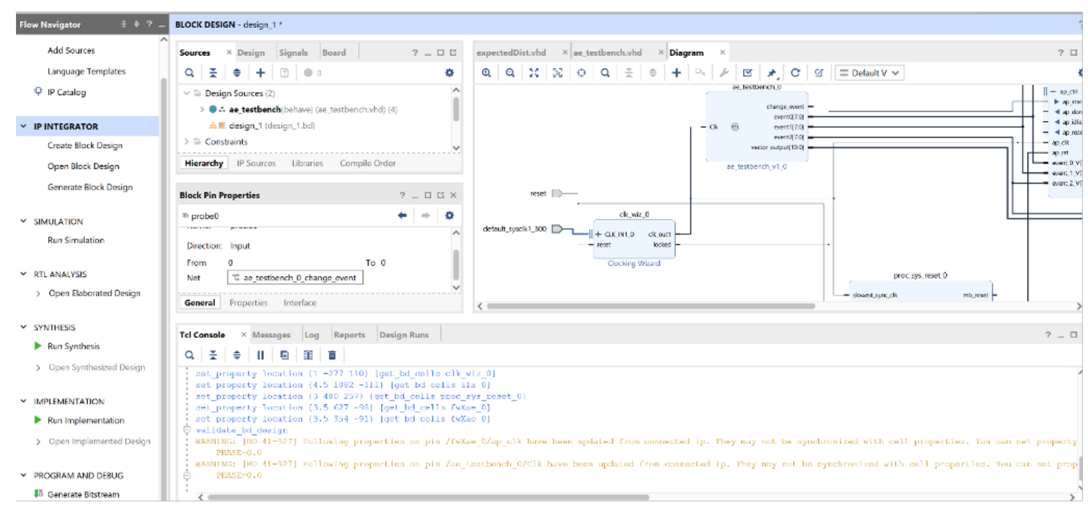
Before beginning, please make sure that you have (and know the location of) the autoencoder IP folder, and the VHDL testbench files:

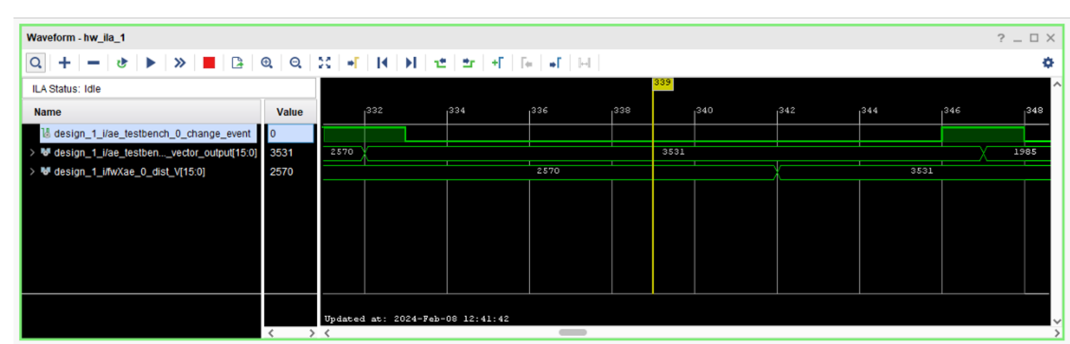
Name	Date modified	Type	Size
autoencoder8var_ip	2/7/2024 1:30 PM	File folder	
tb_vhd_files	2/8/2024 11:50 AM	File folder	

Creating New Project in Vivado

Open Vivado 2019.2 and select "create new Project." On the following pop-up, select "next," and you will be prompted to name the project. Name the project as you wish and choose a location to store it. Keep clicking next until you reach a page that prompts you to select the part/ board. For this tutorial, we will be using the Virtex UltraScale+ VCU118 board. After you have selected your part or board, keeping clicking "next" until you have reached the end of the setup page.







Backup

Example: handwritten numbers

- Teach it about the number 4

784 variables (8-bit)

=

Corresponding data set

Image	Pixel 1	Pixel 2	...	Pixel 300	...	Pixel 783	Pixel 784
1	0	0	...	240	...	0	0
2	0	1	...	255	...	0	0
...	...	...	...	...	...	...	...
500k	0	0	...	231	...	0	0

Details

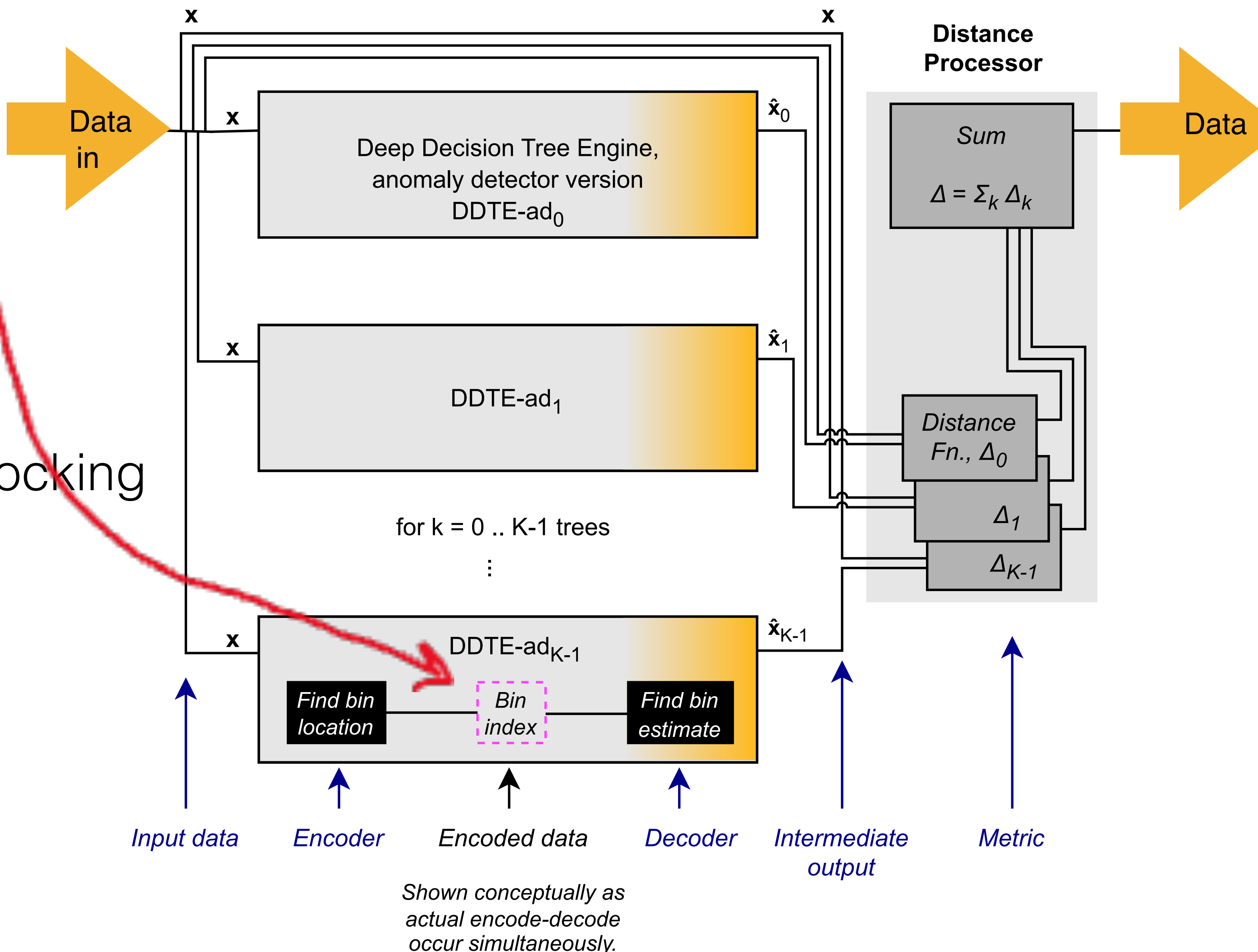
- Each pixel in the data set are unrelated to each other

Logic flow

- Left-to-right data flow (see right)
- Realized that we can bypass the latent space!
  - Encoding = Decoding

Details

- Parallel computing
  - TREE ENGINES eval. in parallel
  - All combinatoric logic, so no clocking between steps = fast
  - Mostly comparisons = fast
  - No multiplication = fast
- Technical info in backup & [2304.03836]



Design v2: Parallelize terminal bins

Go deeper from 4 → 8

- Improve FWXv1

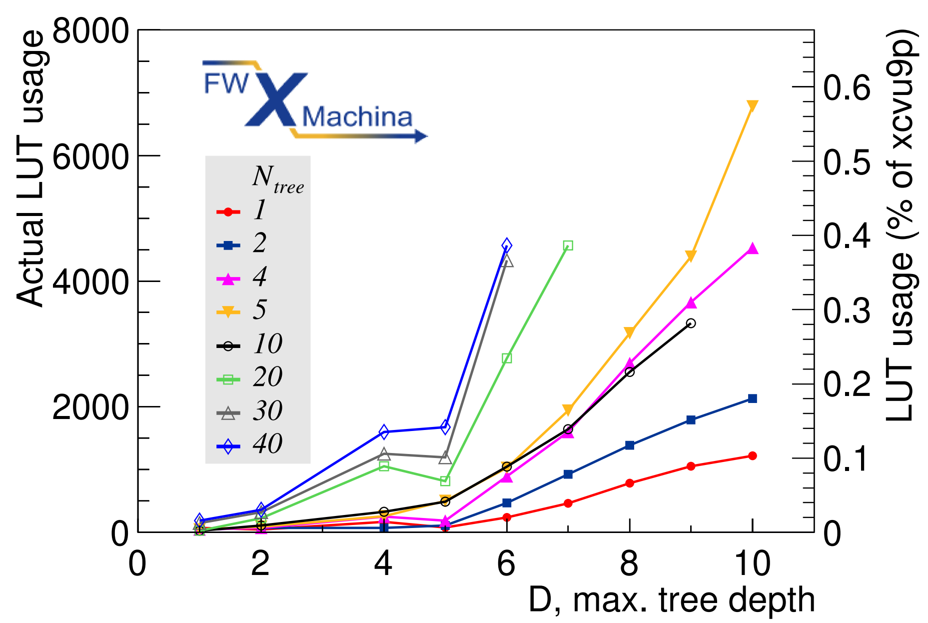
Challenge — Does not scale well w/ tree depth & # variables

Cut redundancy 2^D

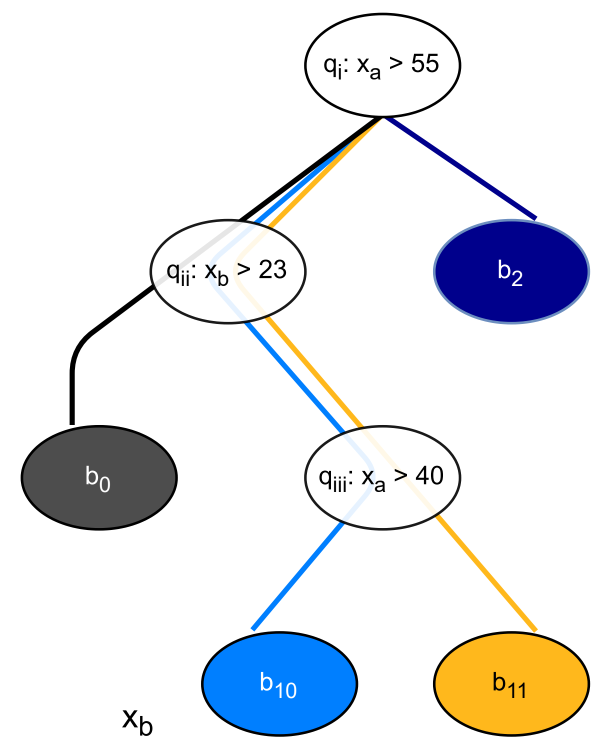
- FWXv2

Key design — Evaluate decision paths

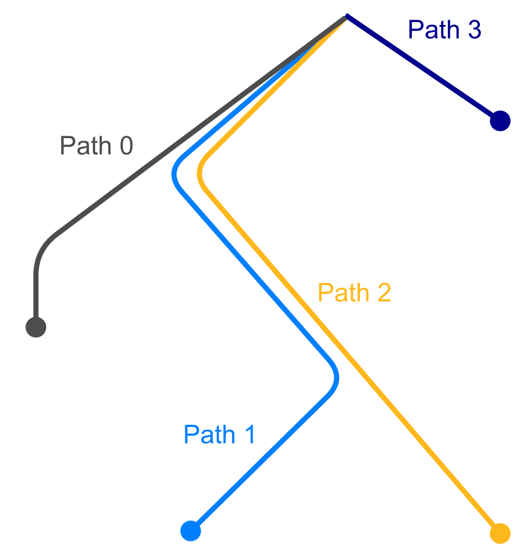
Benefit — Softer scaling vs 2^D

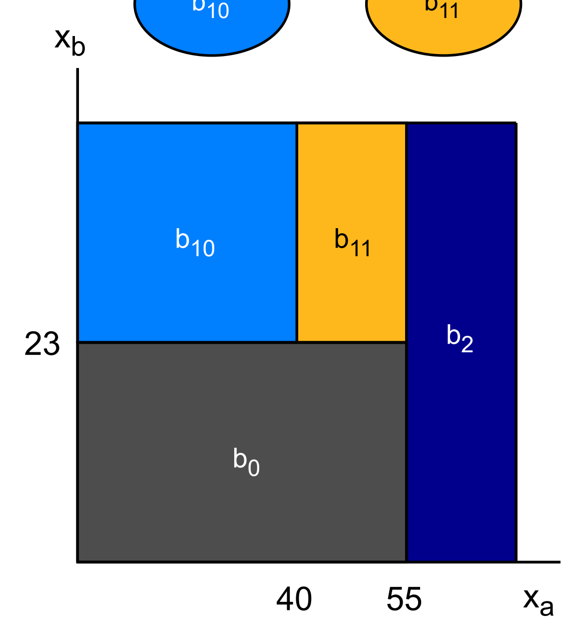


Standard decision tree



FWX design v2





Destination bin	Decision path
b_0	not(q_i) and not(q_{ii})
b_2	q_i
b_{10}	not(q_i) and q_{ii} and not(q_{iii})
b_{11}	not(q_i) and q_{ii} and q_{iii}

Carlson et al., JINST 17, P09039 (2022)

Machine learning

Focus on the most popular use cases in HEP

Supervised classification

- Neural networks & Boosted decision trees
- Others (SVM, kNN, Matrix element, etc.)

Not covered

Structural similarities: NN & BDT

Focus of this section

- Step function boundary
- Fuzzy boundary

Use cases

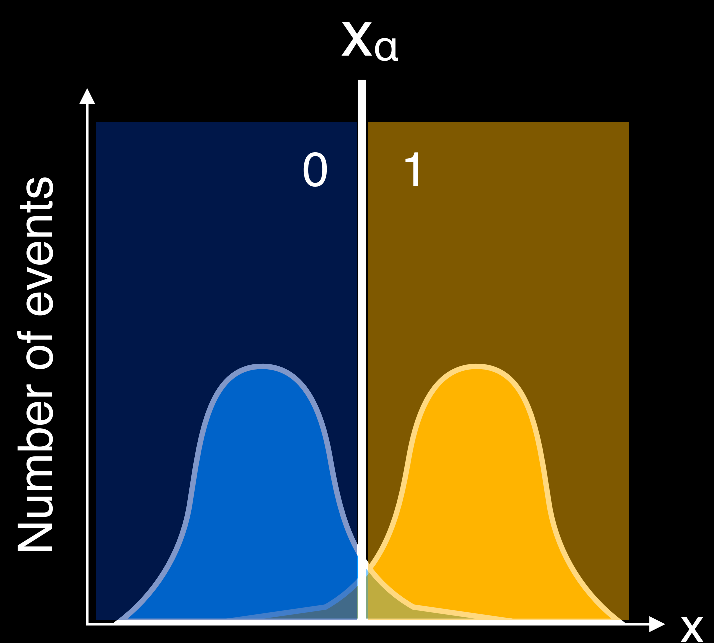
- Regression — Previous slides
- Classification S vs. B — If time
- Anomaly detection B vs. not-B — Later slides

Will discuss other approaches (estimation, unsupervised) after intro

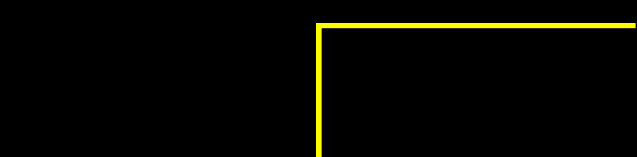
Neural networks basics

From Bruce Denby, *Tutorial on Neural Network Applications in High Energy Physics: A 1992 Perspective*, FERMILAB-CONF-92 / 121-E

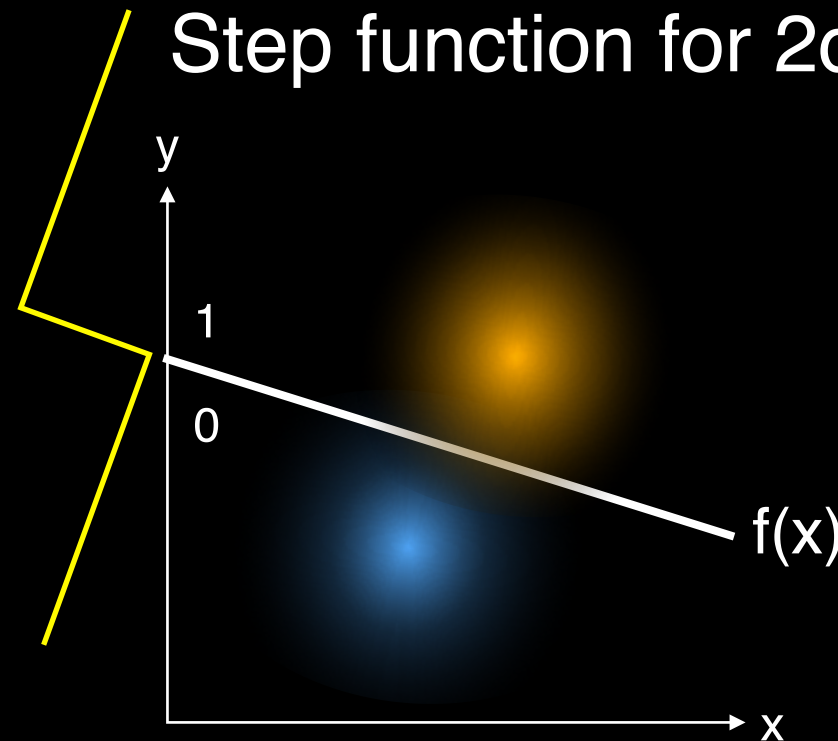
Step function for 1d



$\Theta(x_\alpha - x)$

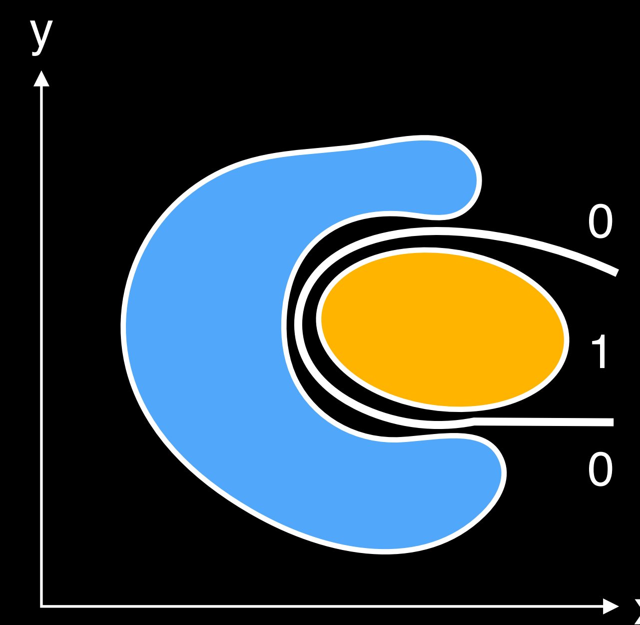


Step function for 2d



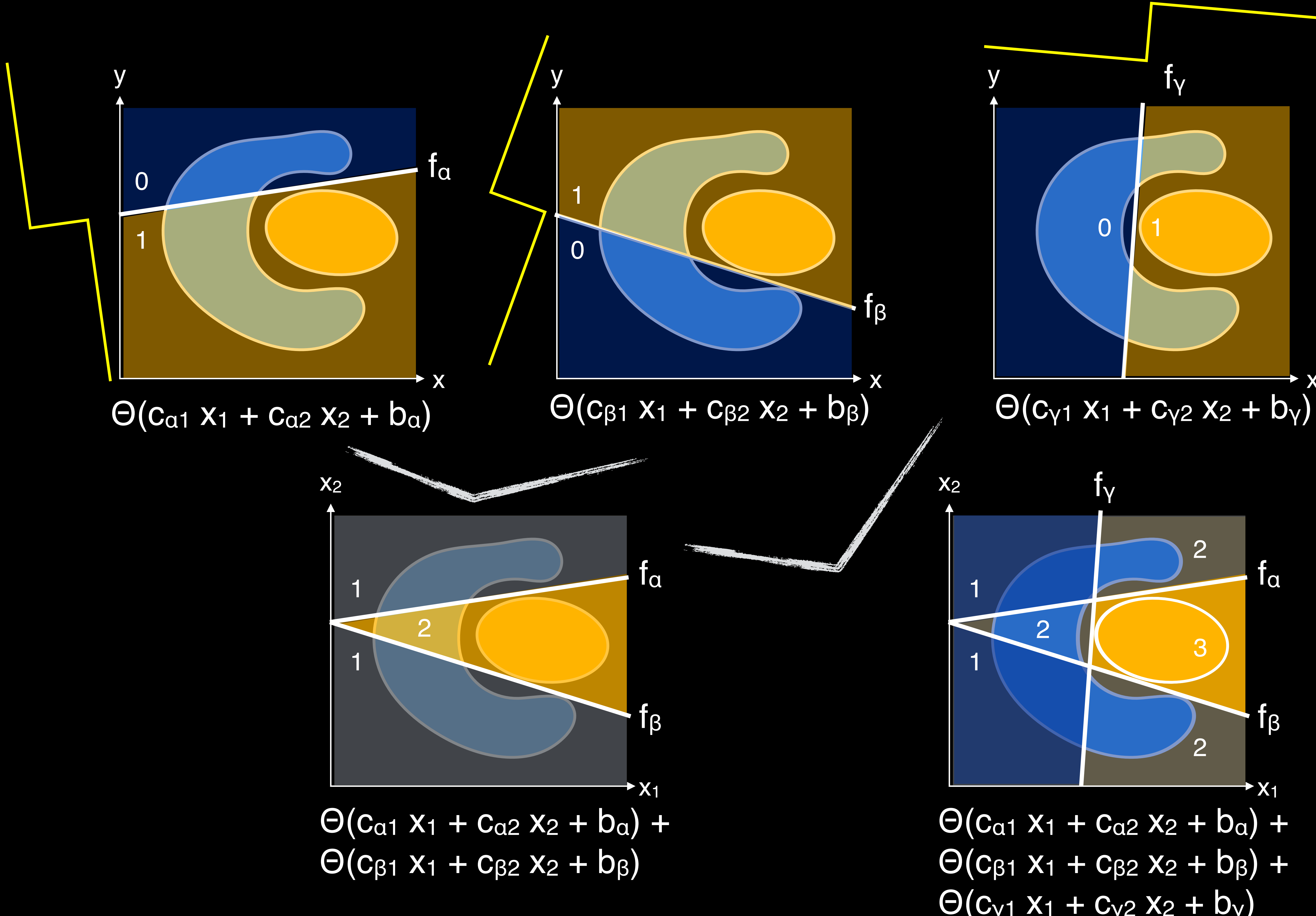
$\Theta(y - f(x))$

Curved step fn? for 2d

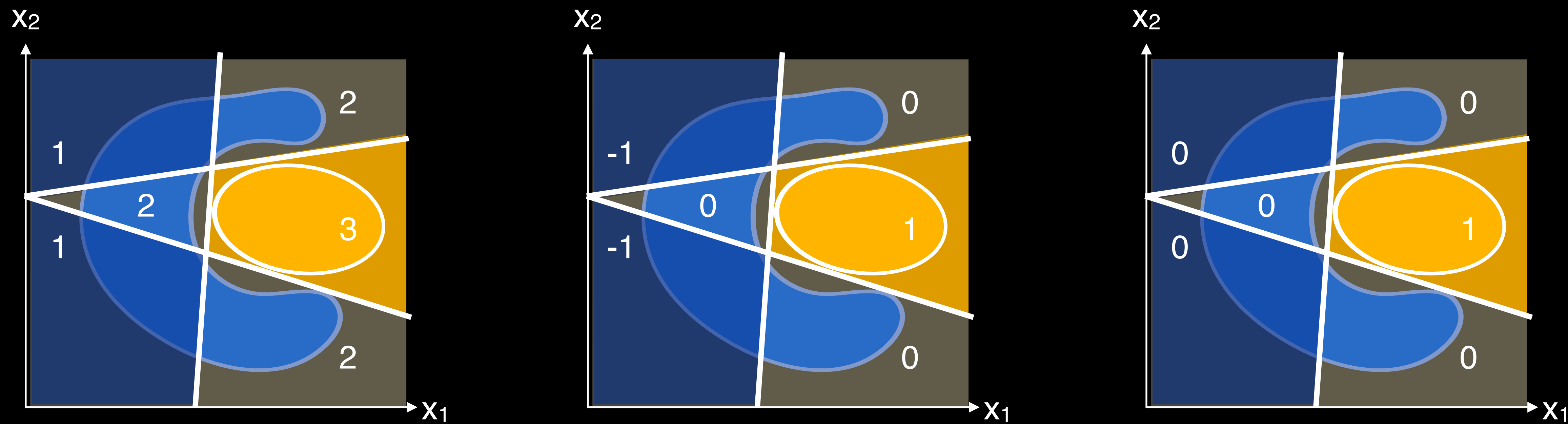


$= \Theta(y - (mx + b))$	substitute
$= \Theta(c_1 y + c_2 x + b)$	multiply by c_1 & define c_2
$= \Theta(c_1 x_1 + c_2 x_2 + b)$	generalized notation
$= \Theta(\mathbf{c} \cdot \mathbf{x} + b)$	vector notation

Step functions divide samples given a desired true / false positive rates



Sum of step functions can approximate the desired contour



$$\Theta(\mathbf{c}_\alpha \cdot \mathbf{x} + b_\alpha) + \Theta(\mathbf{c}_\beta \cdot \mathbf{x} + b_\beta) + \Theta(\mathbf{c}_\gamma \cdot \mathbf{x} + b_\gamma)$$

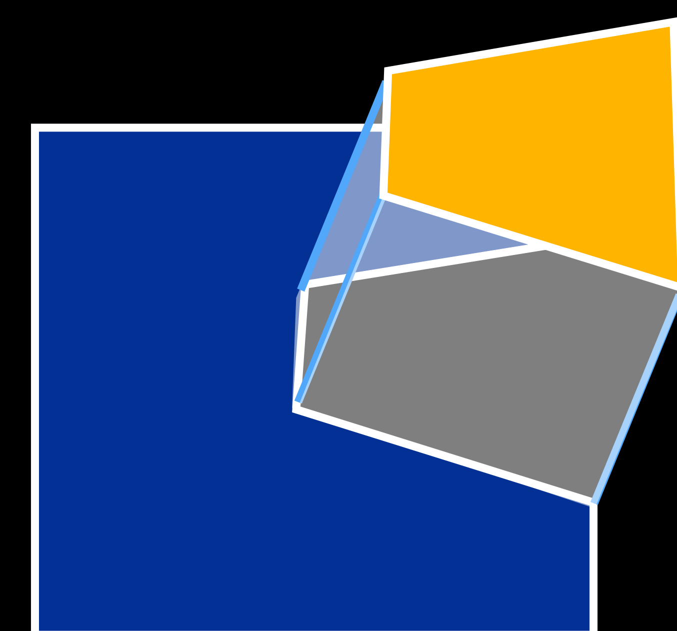
subtract 2

$$\Theta(\mathbf{c}_\alpha \cdot \mathbf{x} + b_\alpha) + \Theta(\mathbf{c}_\beta \cdot \mathbf{x} + b_\beta) + \Theta(\mathbf{c}_\gamma \cdot \mathbf{x} + b_\gamma) - 2$$

threshold

$$\Theta\big(\Theta(\mathbf{c}_\alpha \cdot \mathbf{x} + b_\alpha) + \Theta(\mathbf{c}_\beta \cdot \mathbf{x} + b_\beta) + \Theta(\mathbf{c}_\gamma \cdot \mathbf{x} + b_\gamma) - 2\big)$$

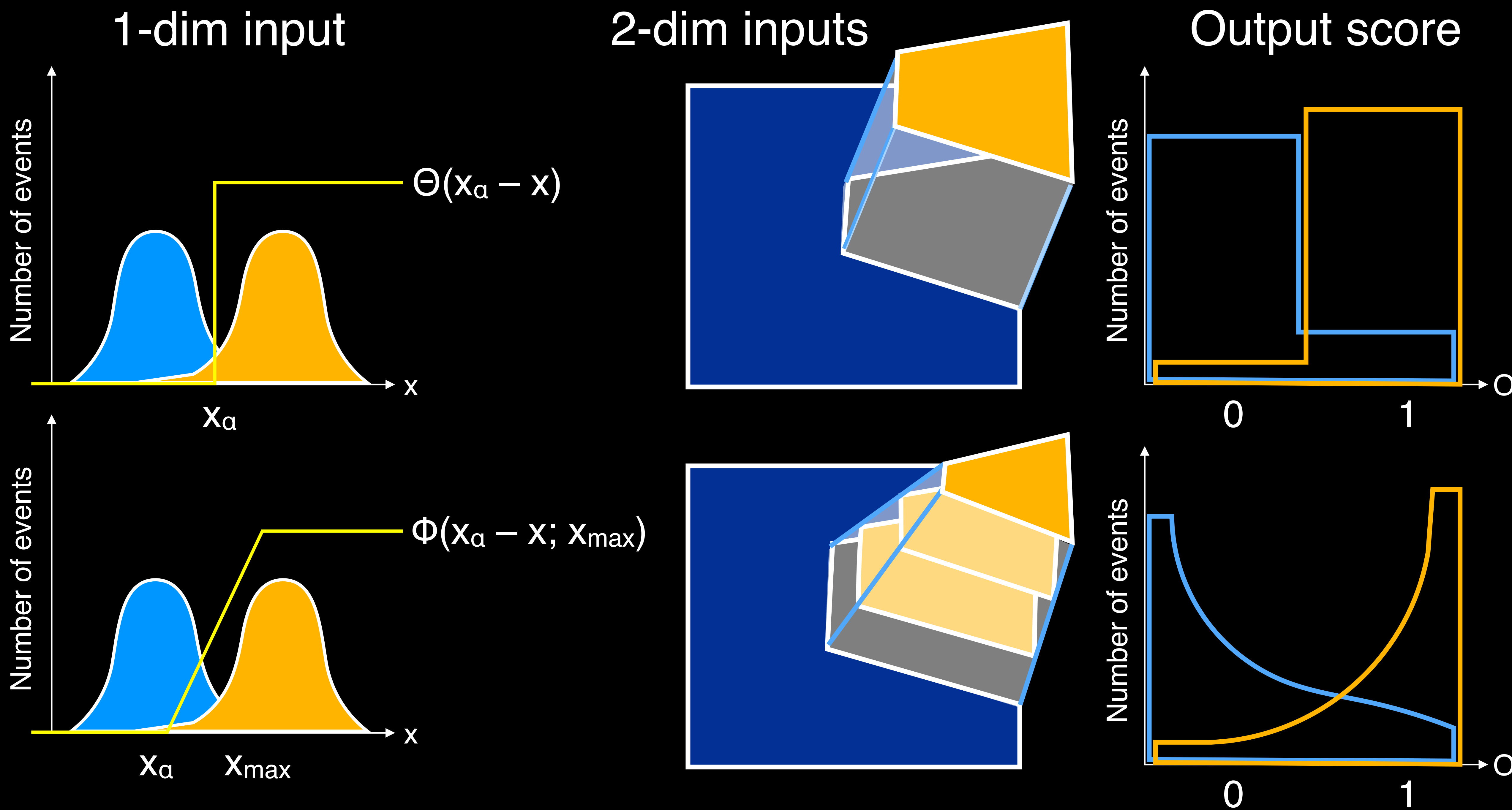
Step function for 2-dim inputs



The contour is converted to the final step function

Activation function

Fuzzy boundary using a function

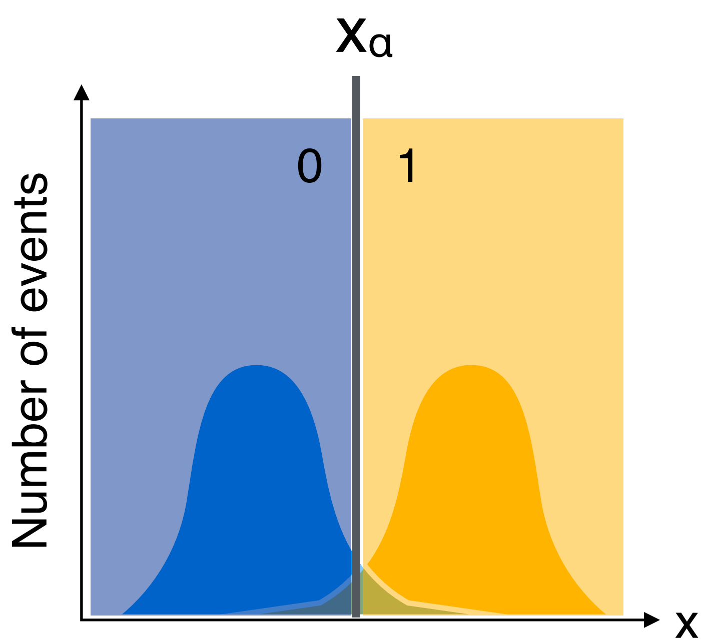


Activation fn gives users a handle to control true / false positive rates

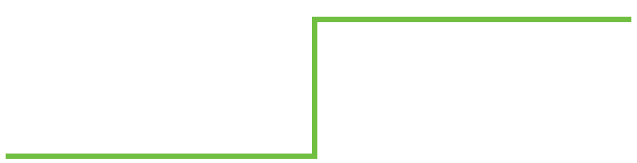
Decision tree basics

And how it achieves the same result as NN

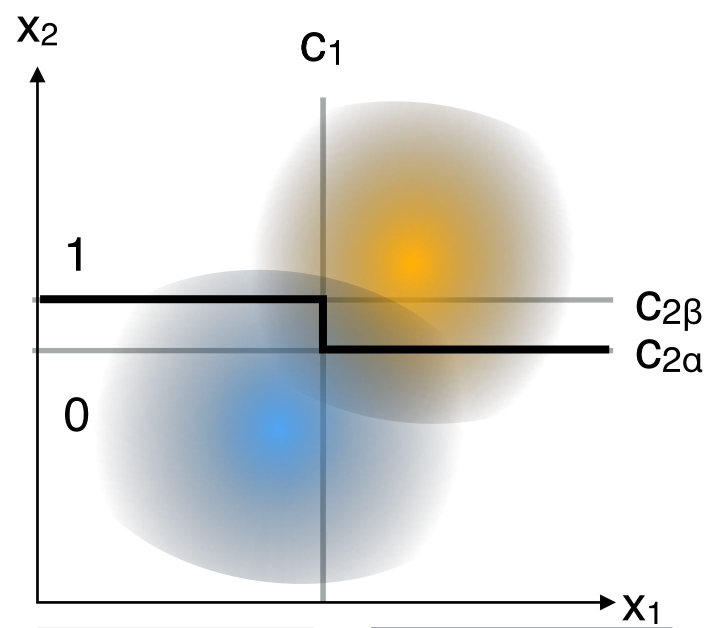
Step function for 1d



$\Theta(x_\alpha - x)$

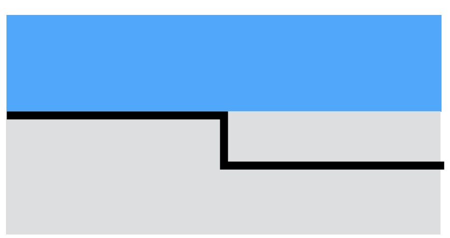


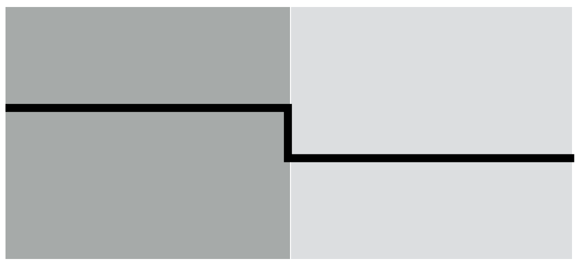
Step function for 2d



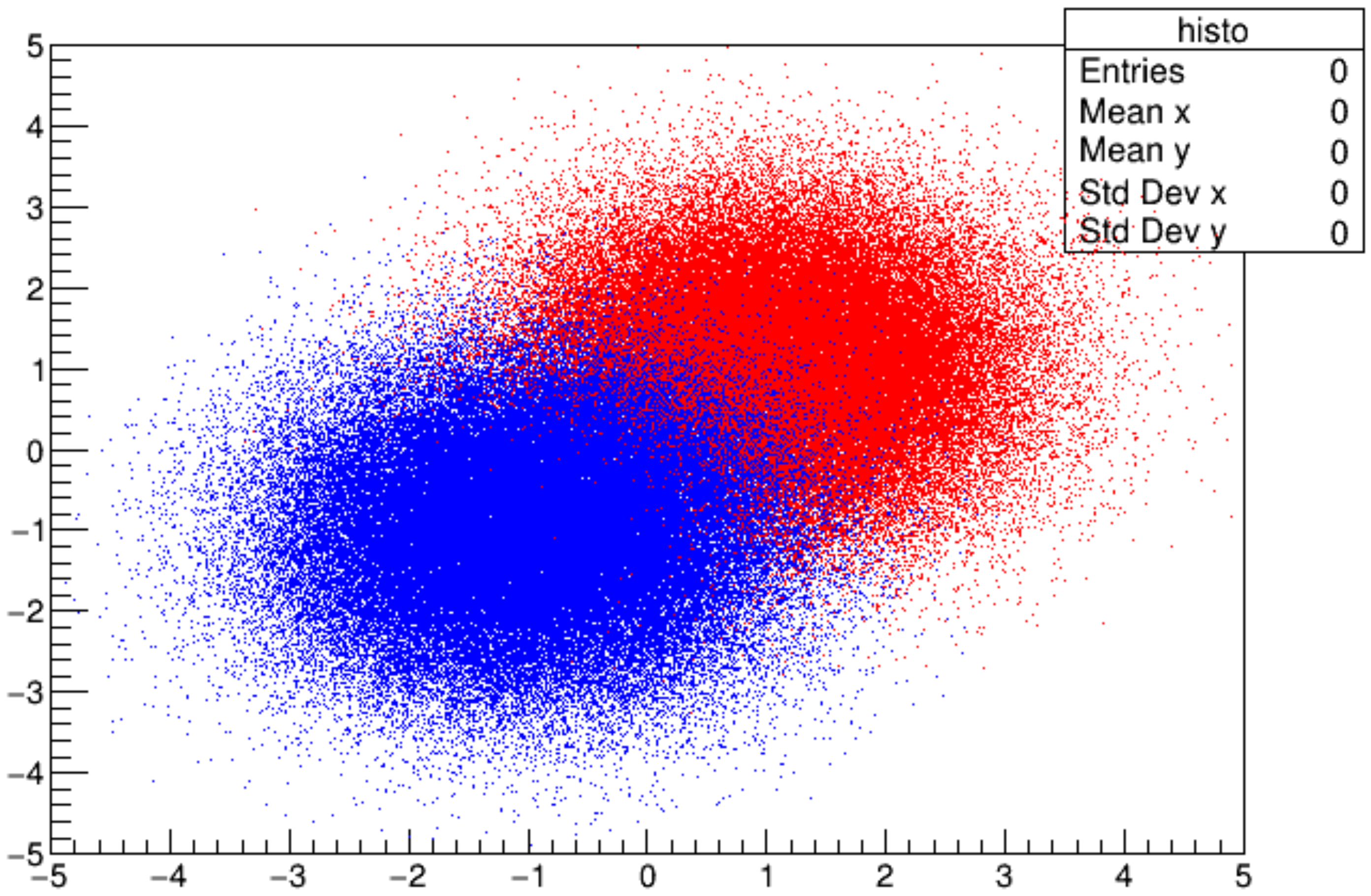
$\Theta(x_1 - c_1) \cdot \Theta(x_2 - c_{2\alpha}) +$

$\Theta(c_1 - x_1) \cdot \Theta(x_2 - c_{2\beta})$



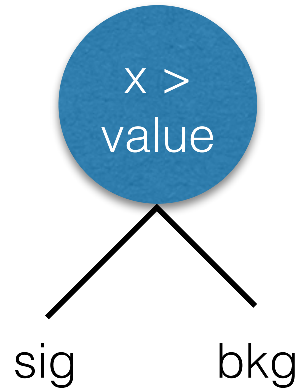


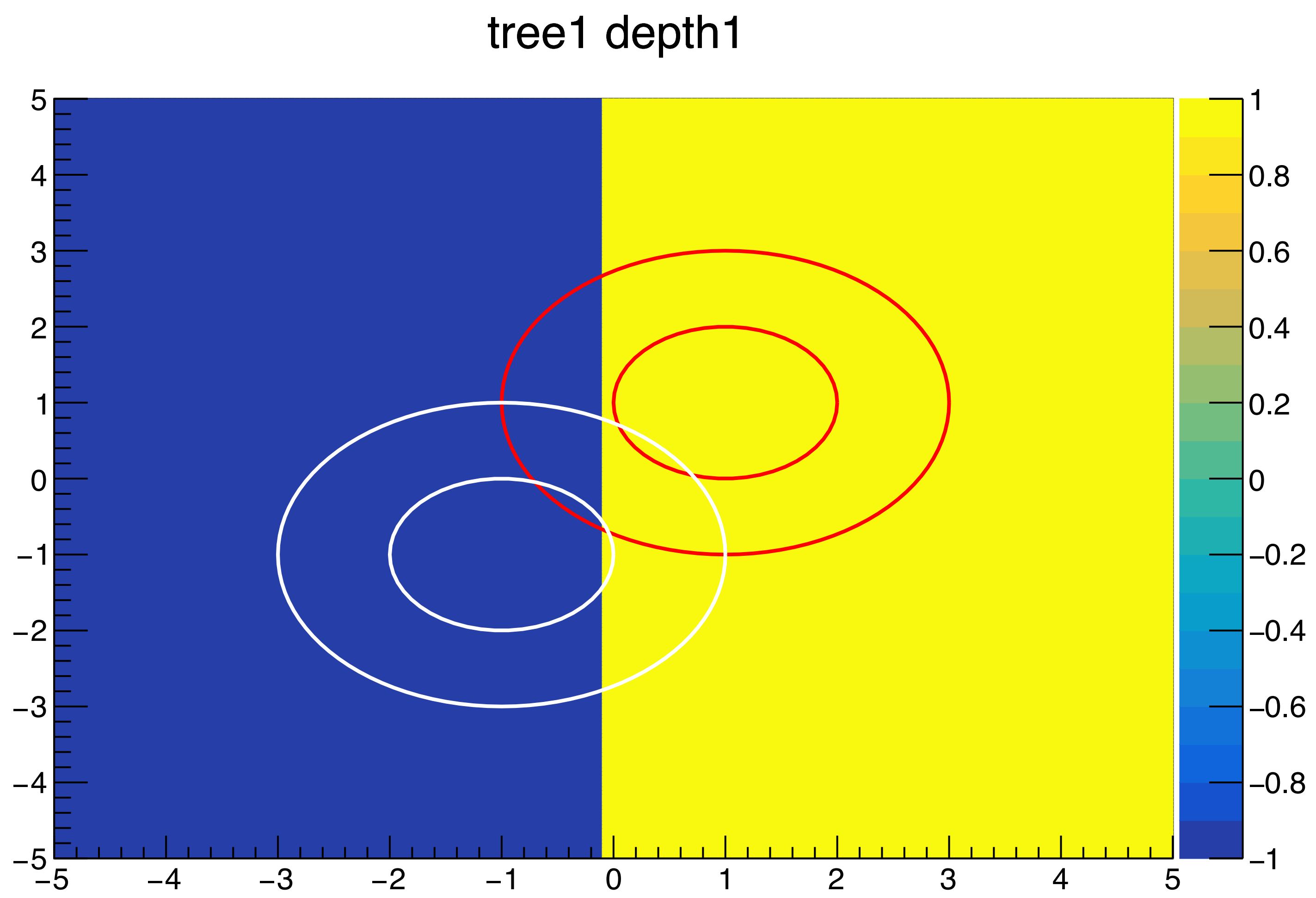
Flip book

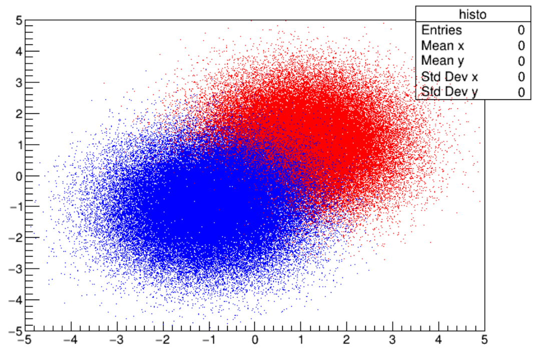


Unit gaussians of two variables

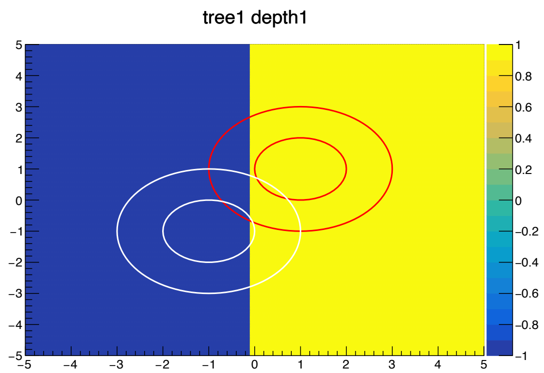
One decision tree

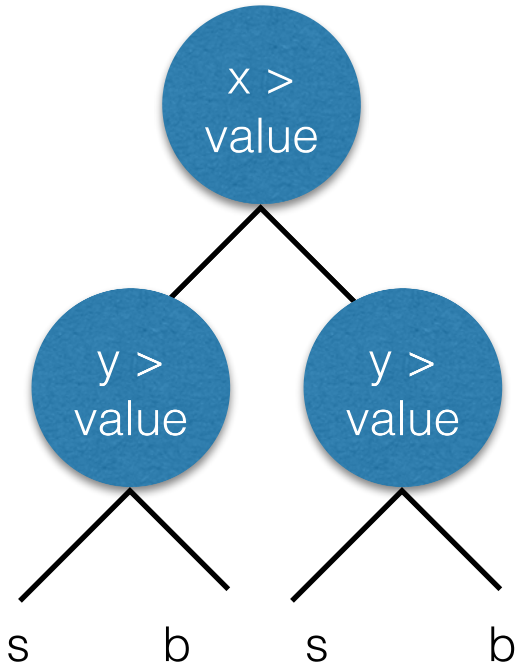


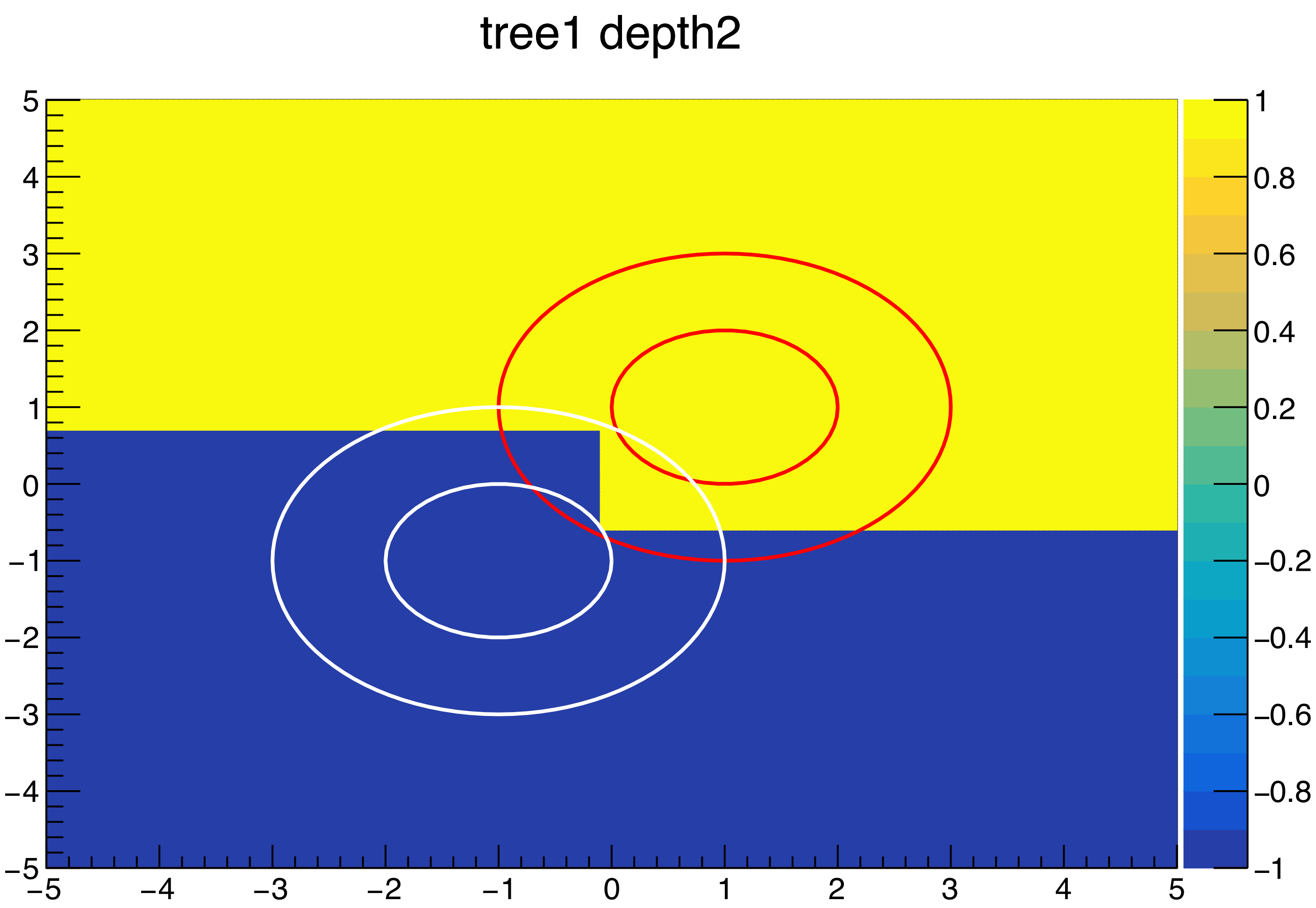




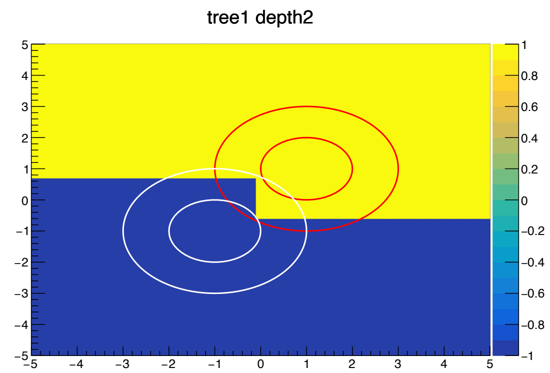
One decision tree

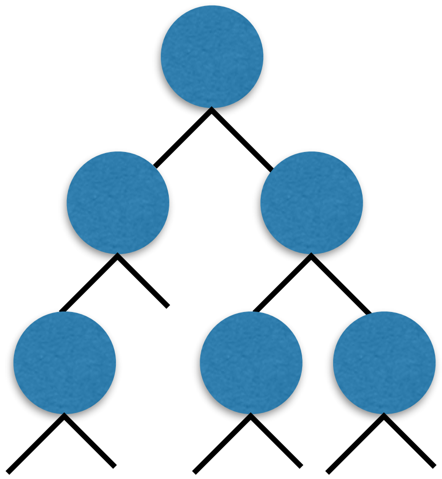


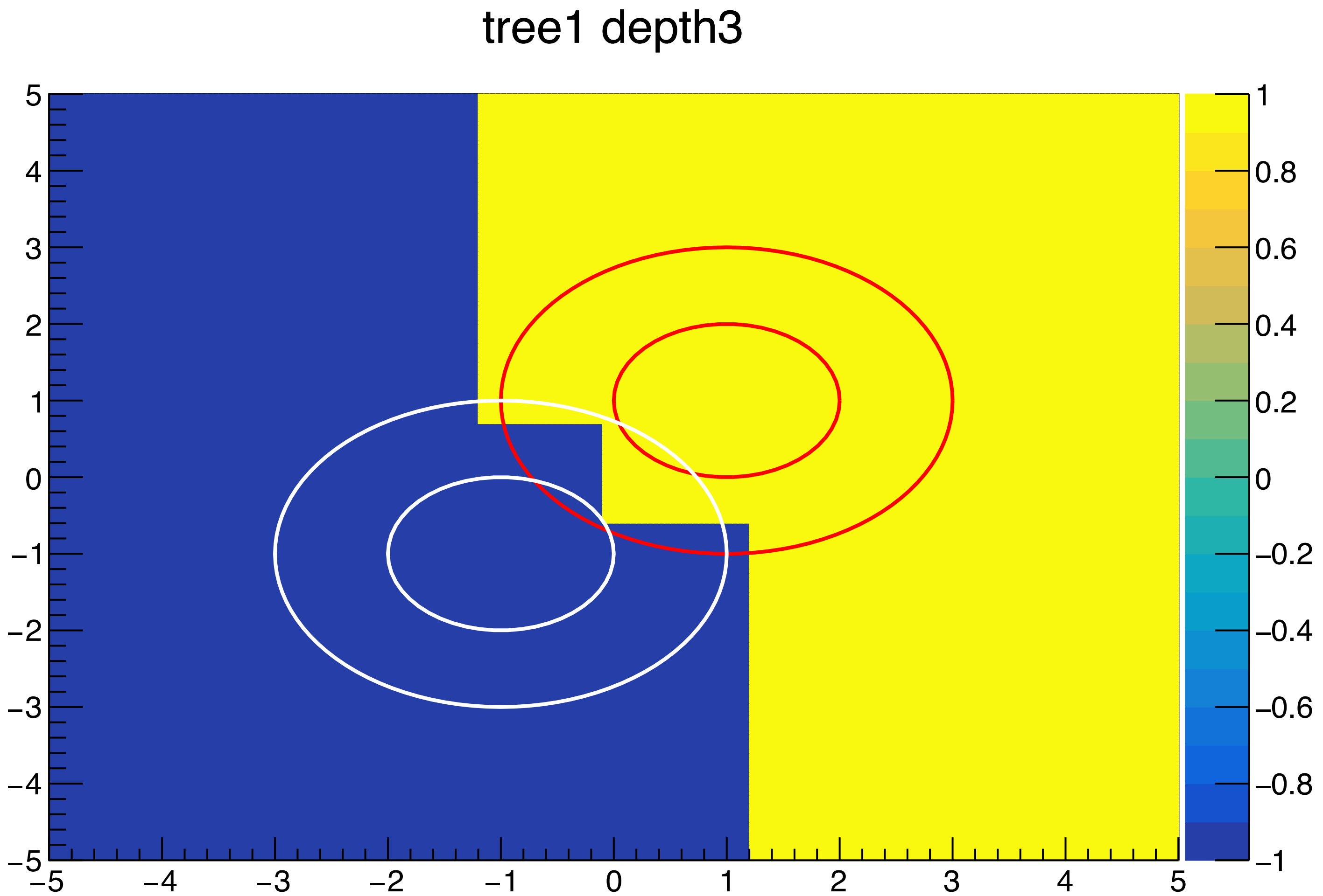




One decision tree

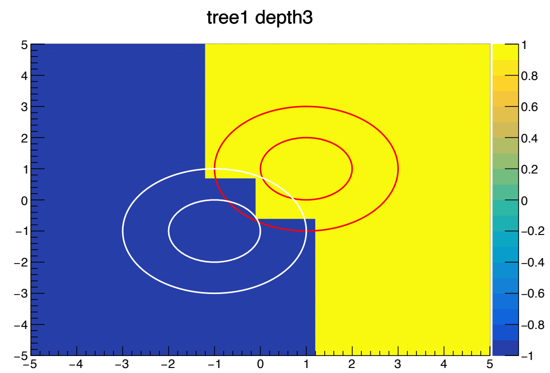


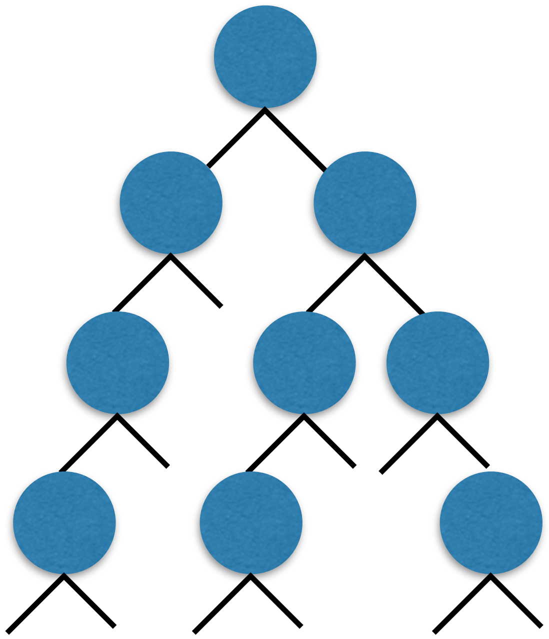


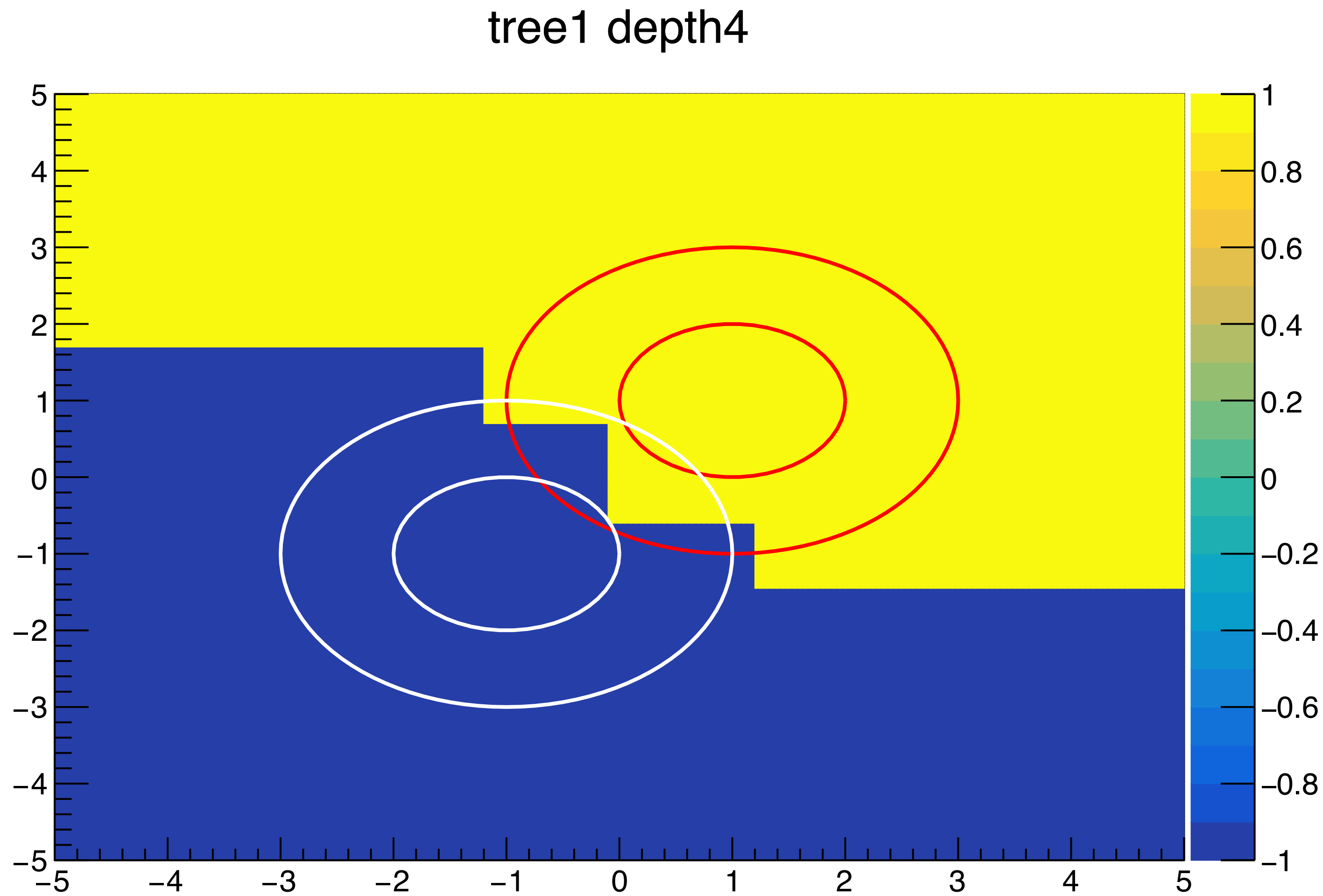


Binary classification

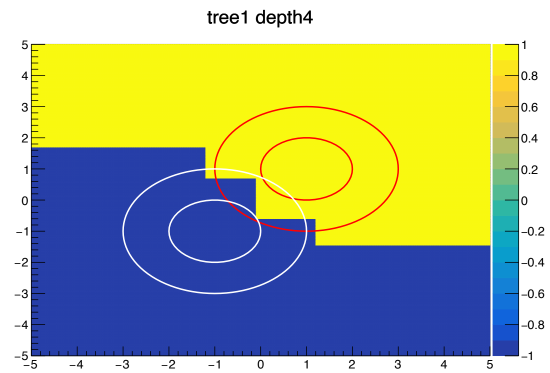
One decision tree

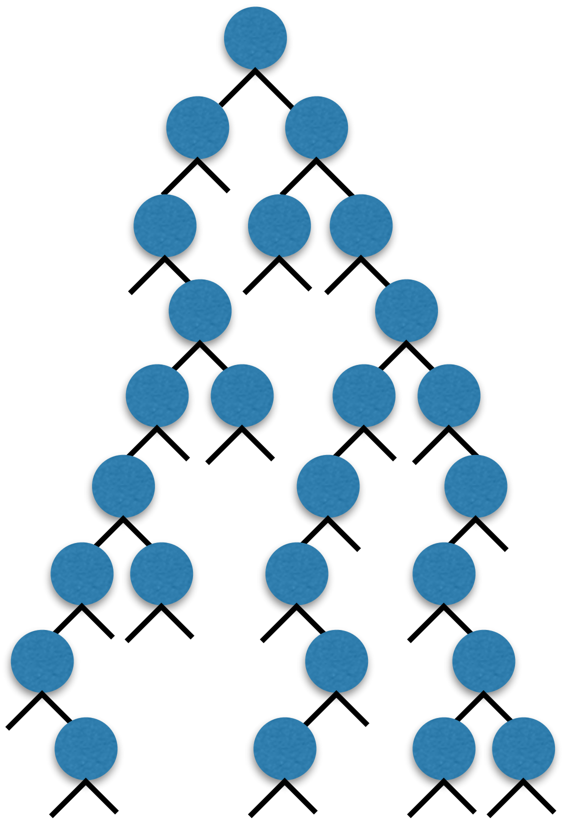


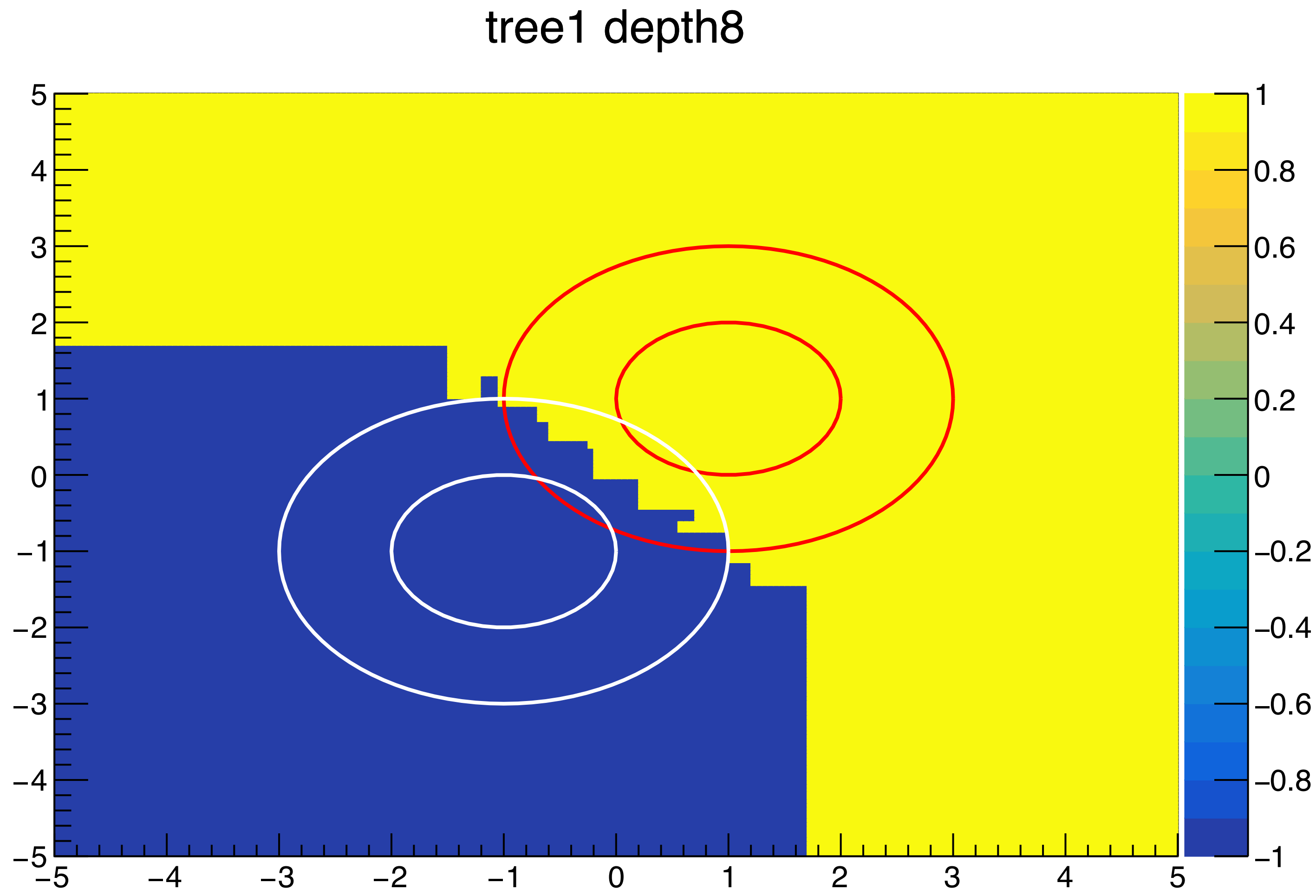




One decision tree





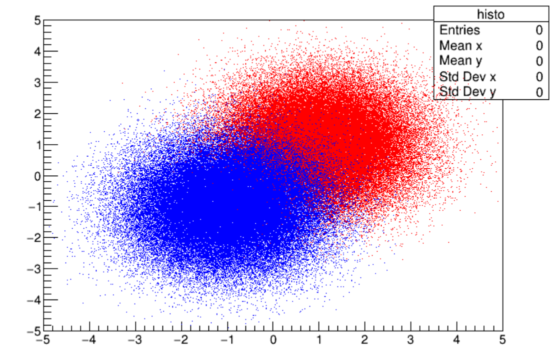


Draws diagonal

Depth 2

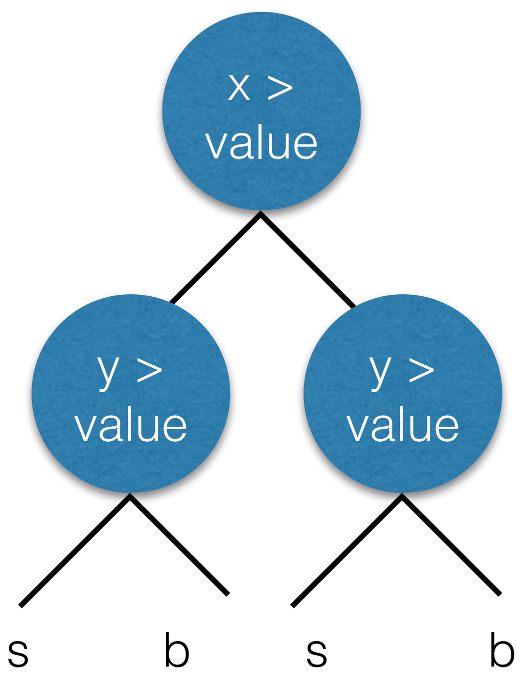
vary trees

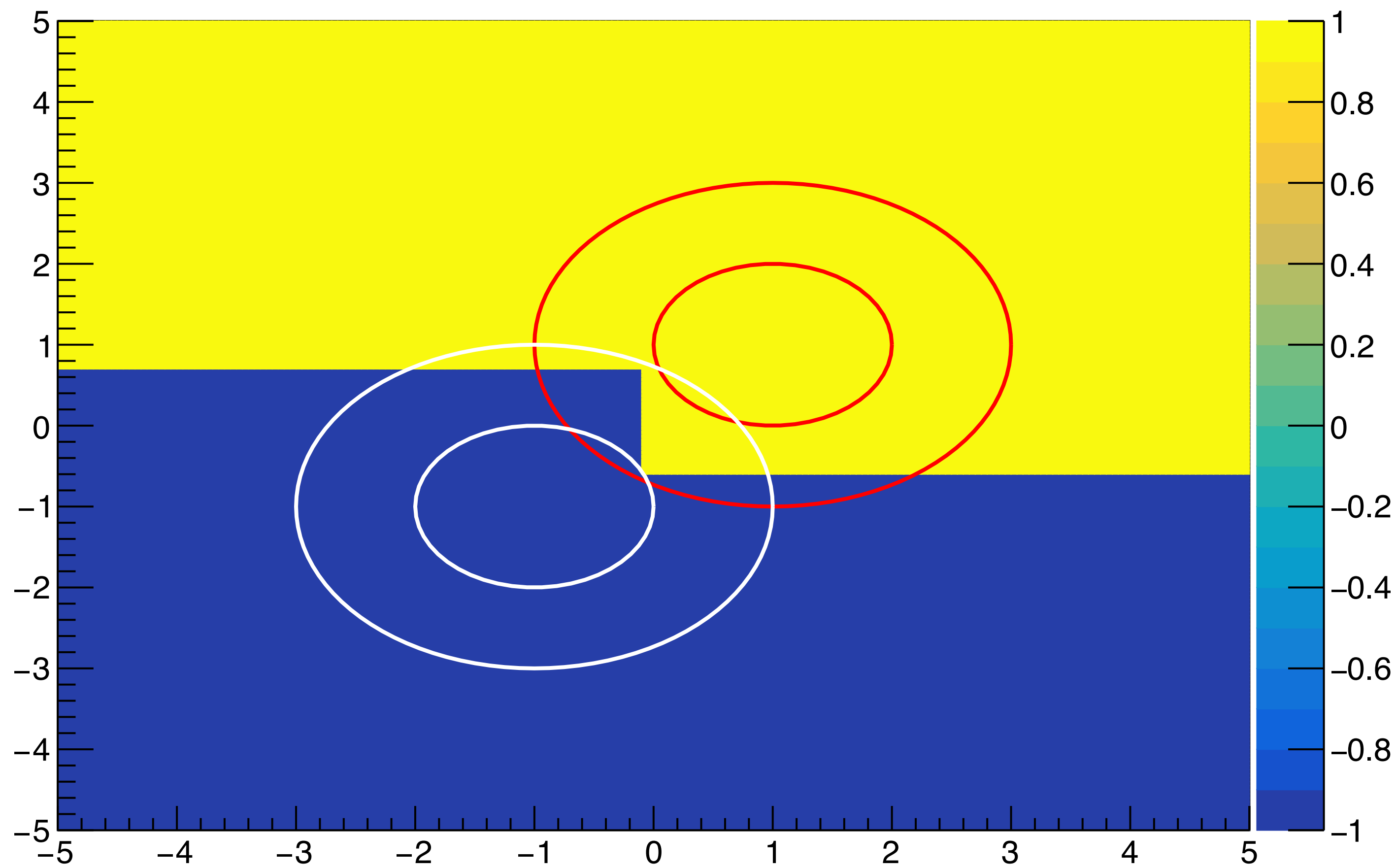
Depth 2



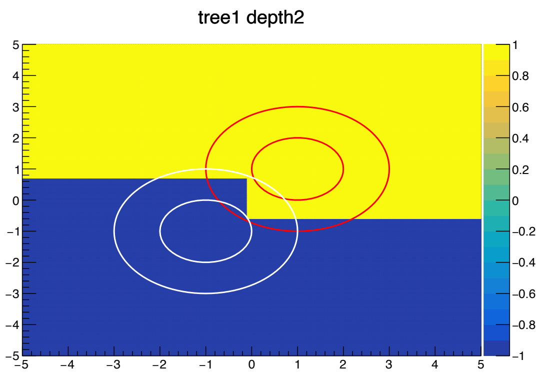
vary

tree1 depth2

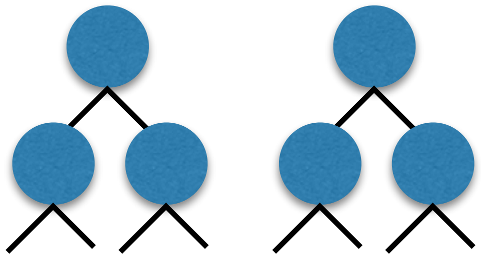




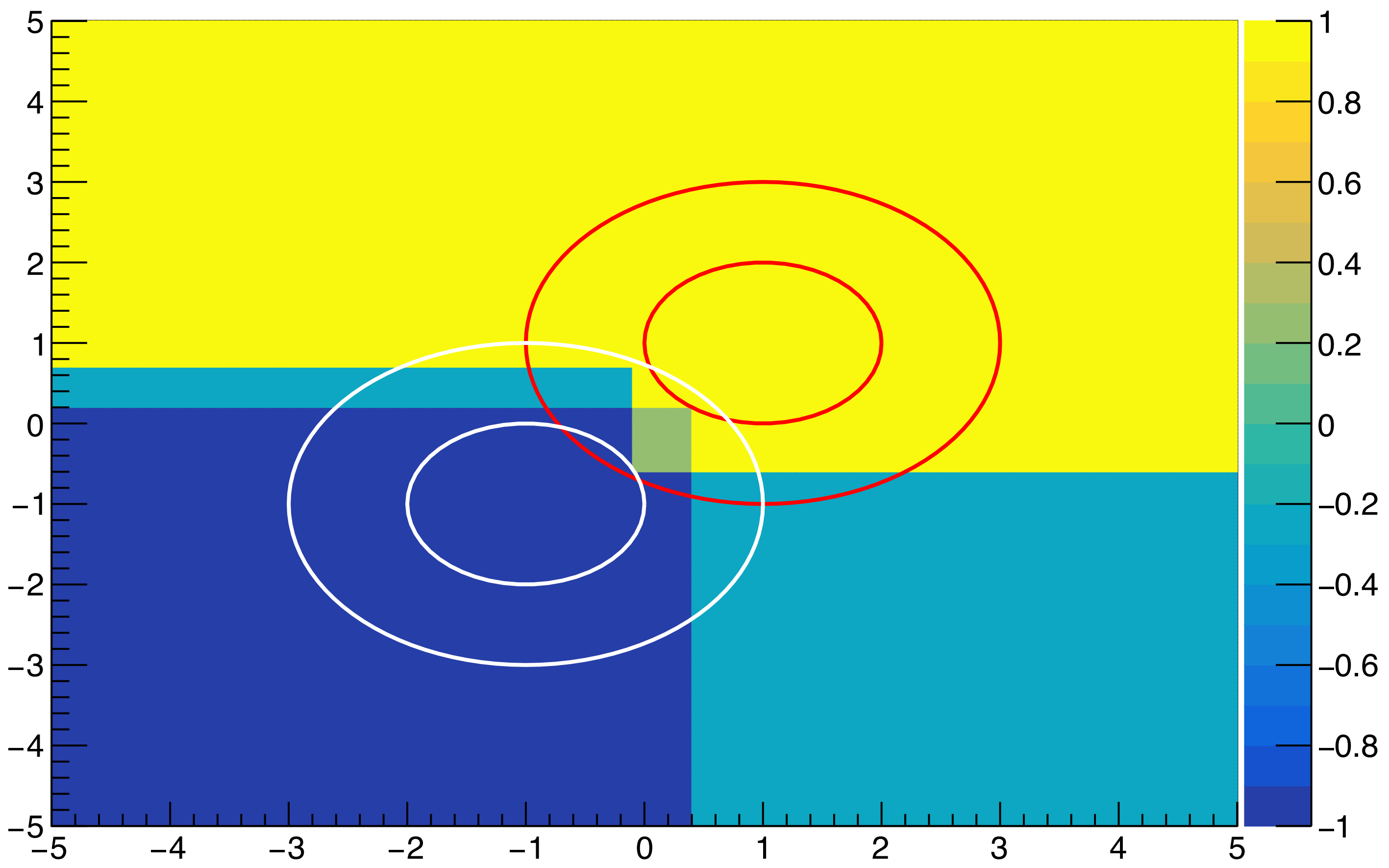
Depth 2



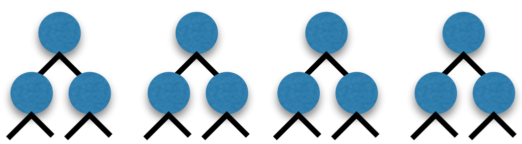
vary



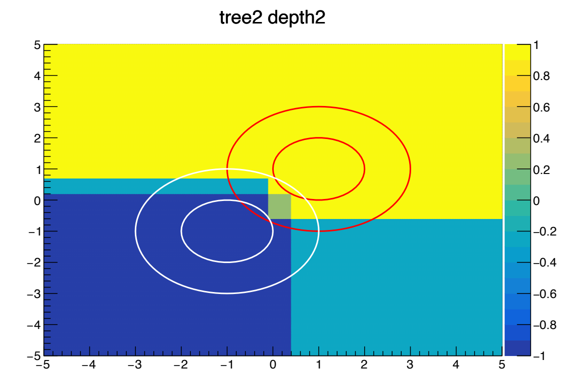
tree2 depth2



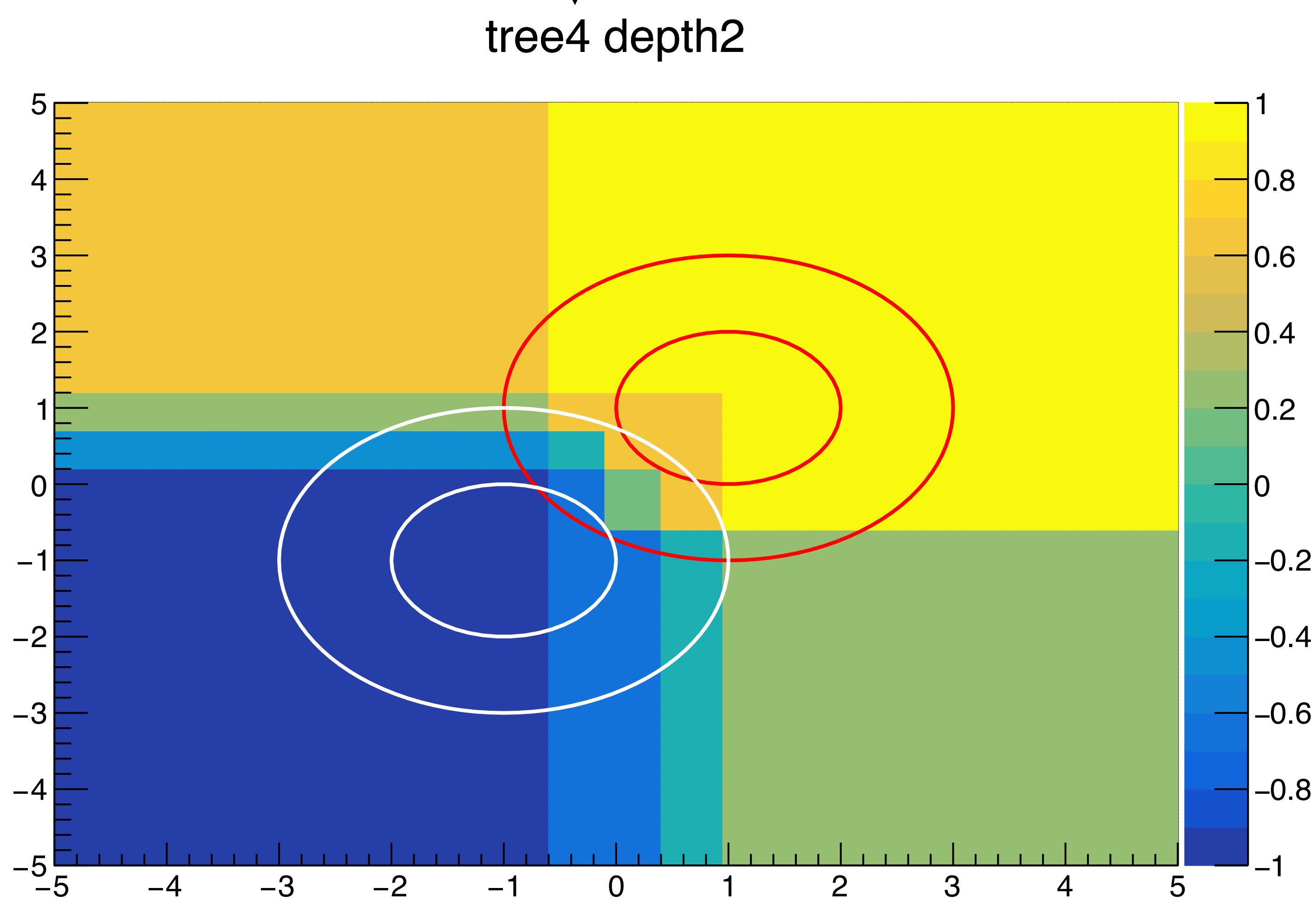
Depth 2



vary

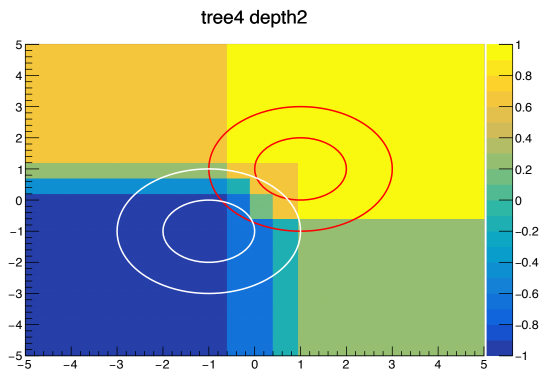


tree2 depth2

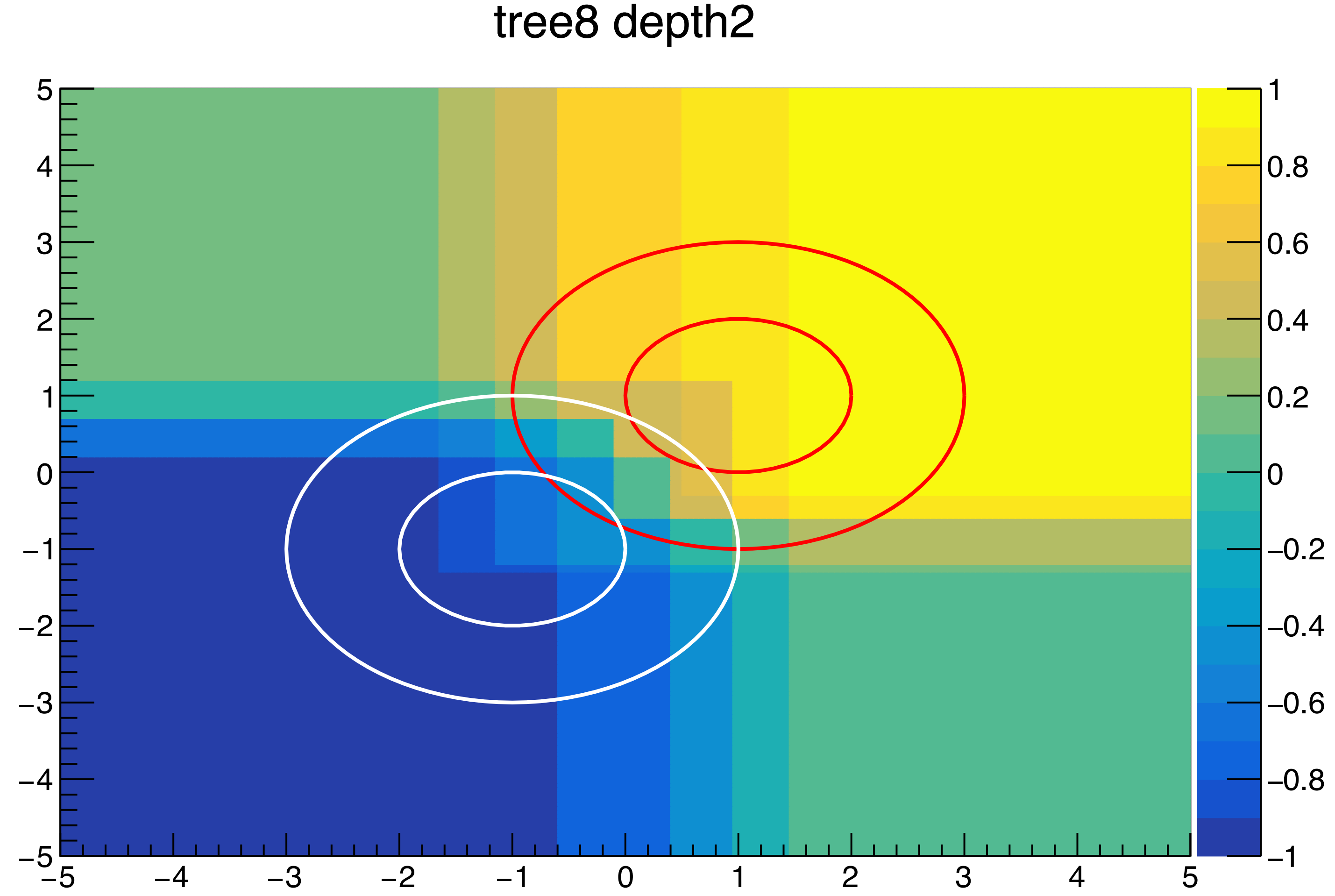


tree4 depth2
5
4
3
2
1
0
−1
−2
−3
−4
−5
−5 −4 −3 −2 −1 0 1 2 3 4 5
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Depth 2

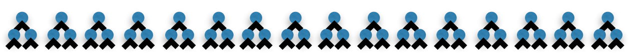

tree4 depth2

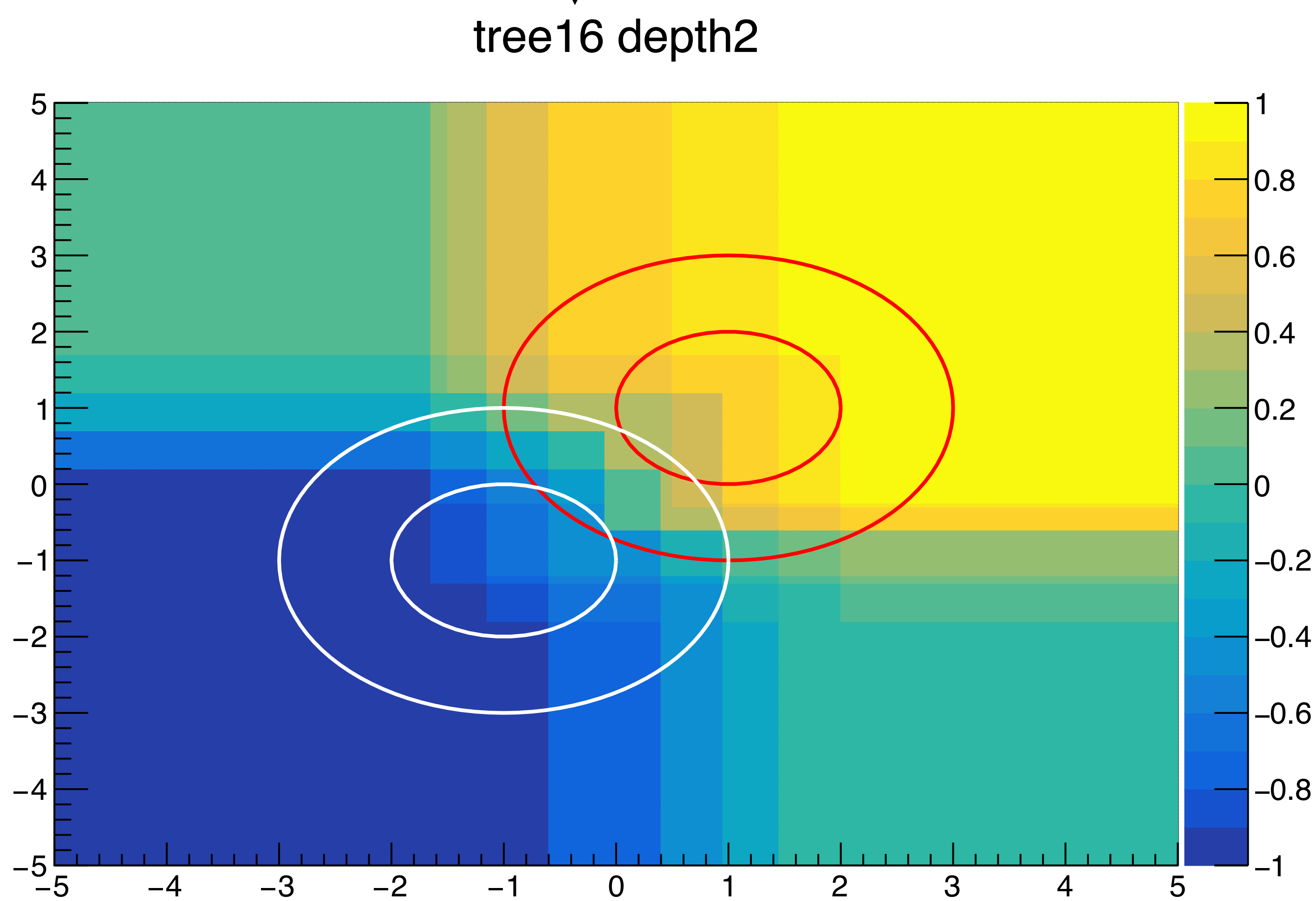
vary

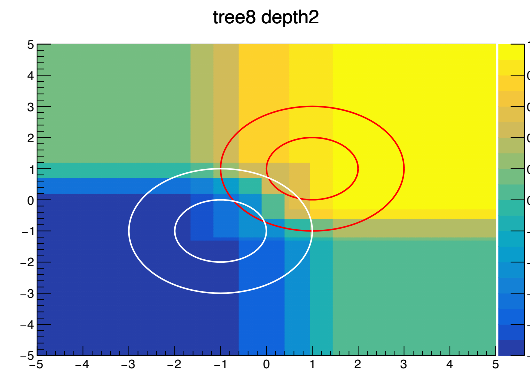

tree8 depth2
5
4
3
2
1
0
−1
−2
−3
−4
−5
−4
−3
−2
−1
0
1
2
3
4
5
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

Depth 2

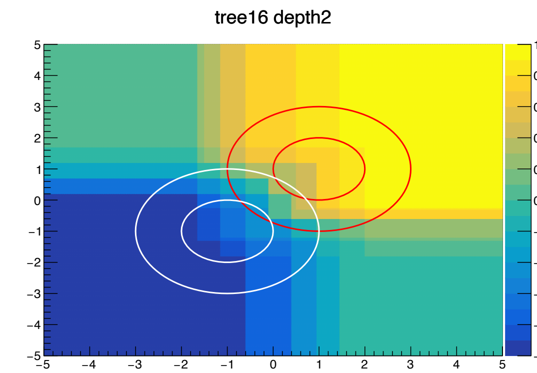
vary




tree16 depth2
5
4
3
2
1
0
−1
−2
−3
−4
−5
−4
−3
−2
−1
0
1
2
3
4
5
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

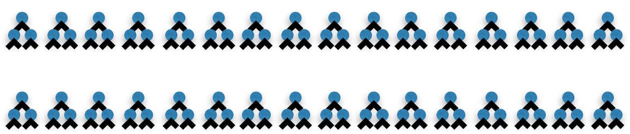

tree8 depth2

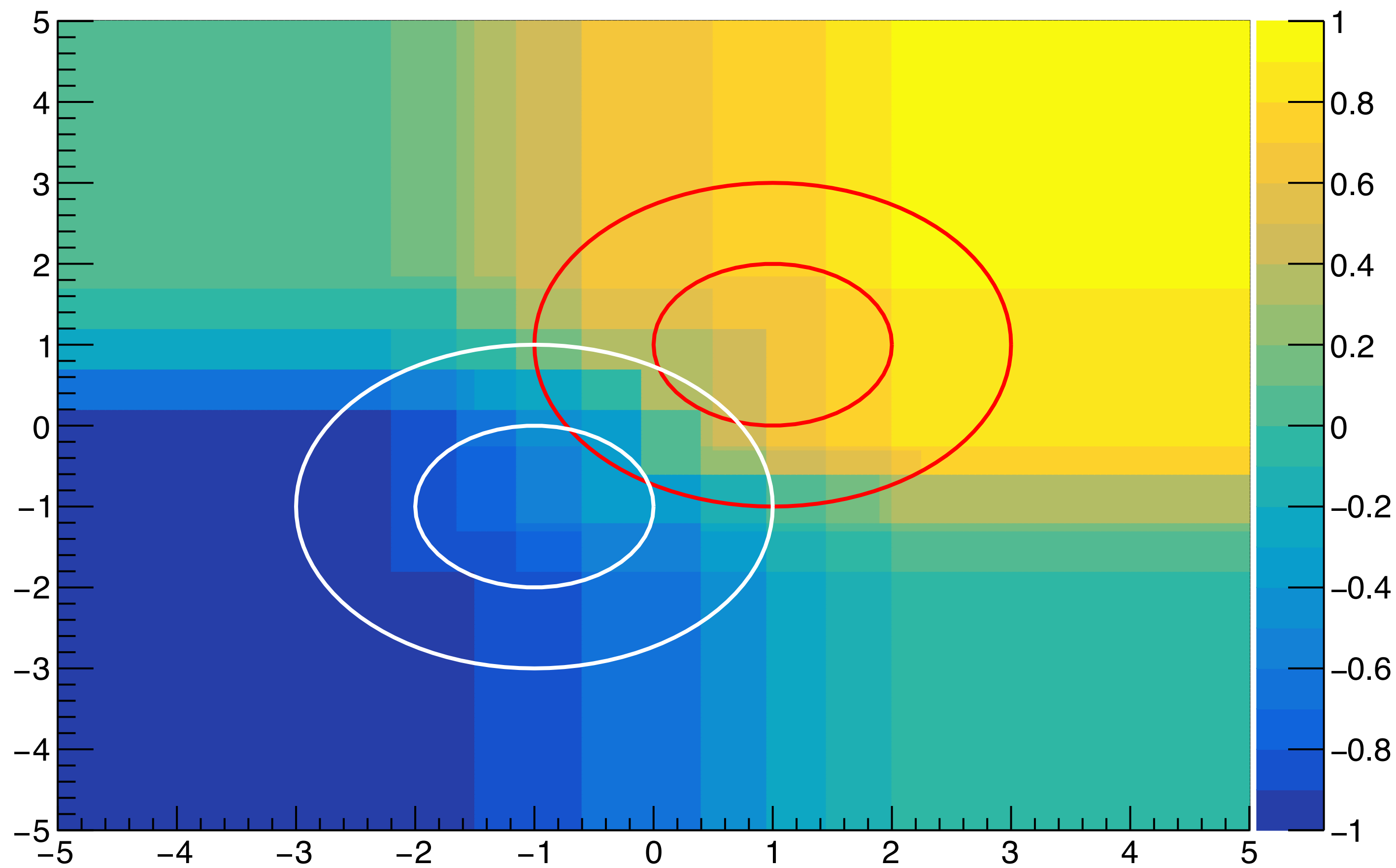
Depth 2



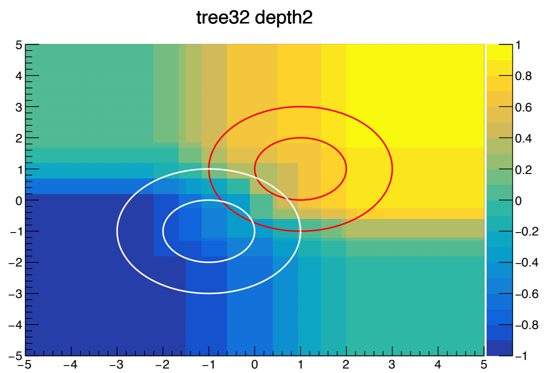
vary

tree32 depth2

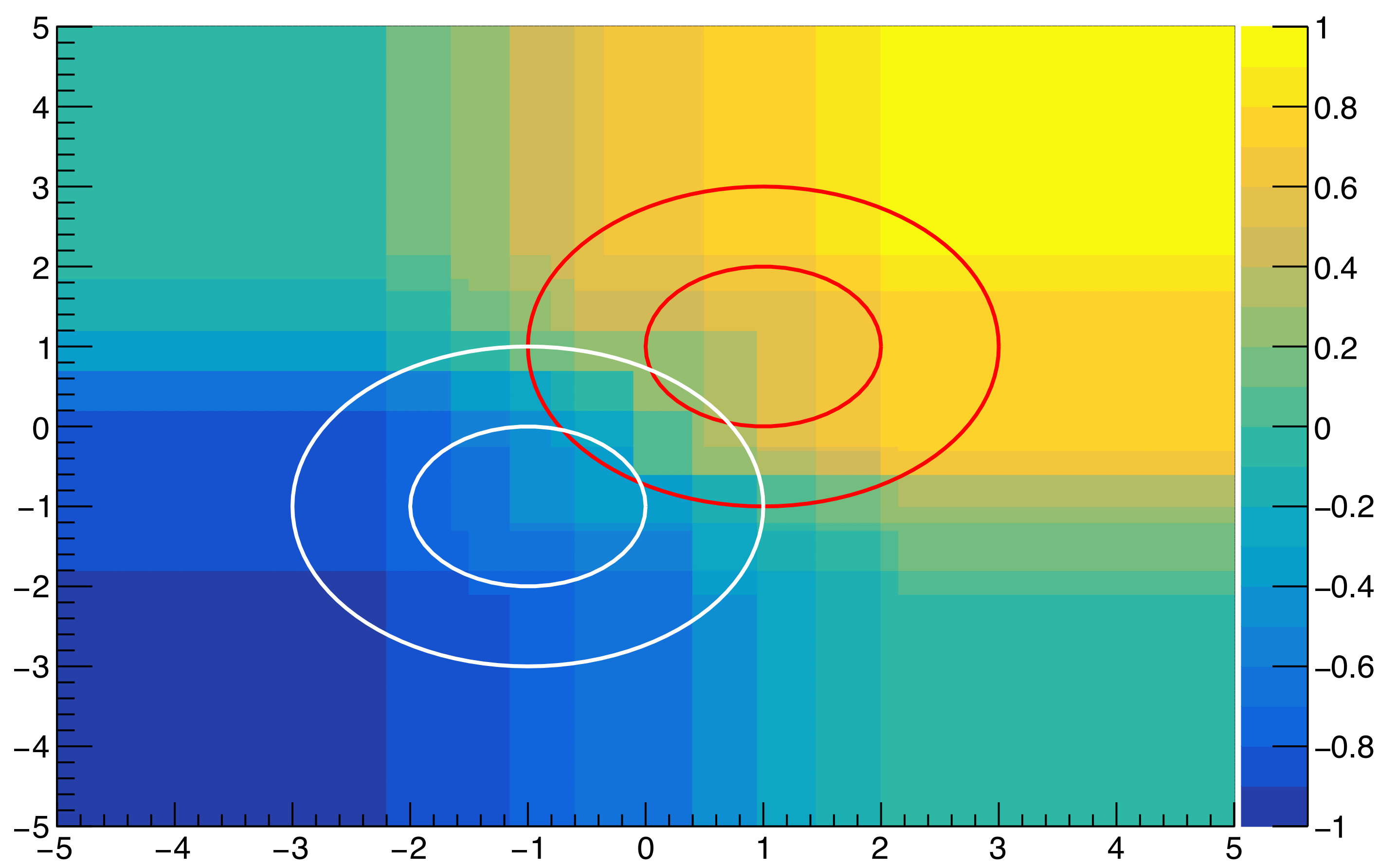




Depth 2

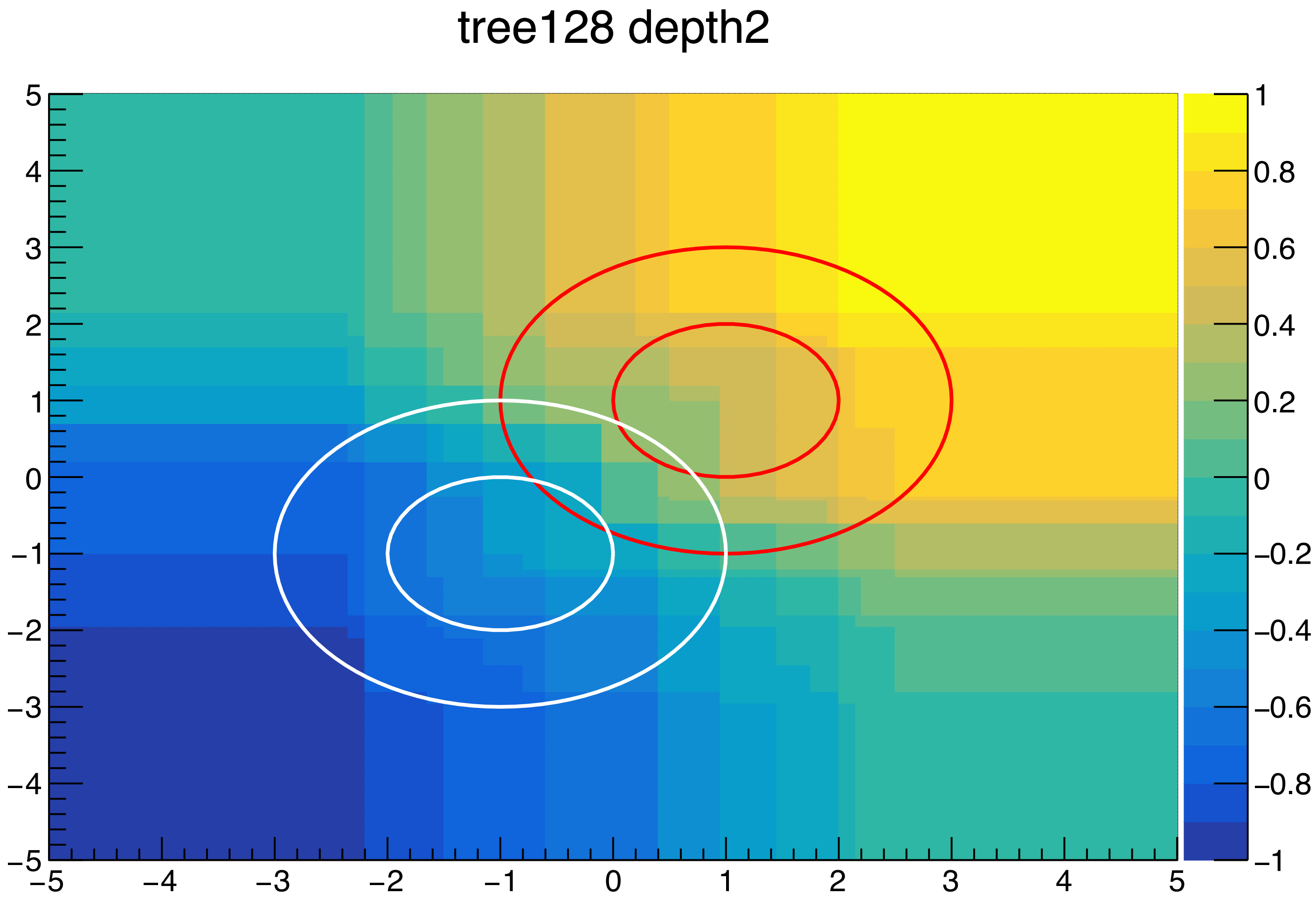


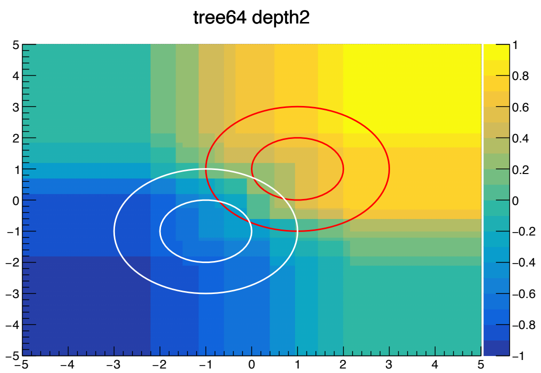
vary



Depth 2

vary



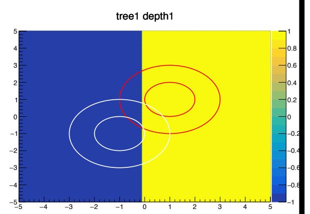
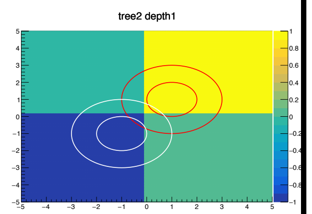
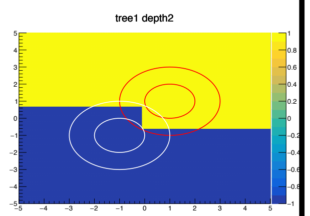
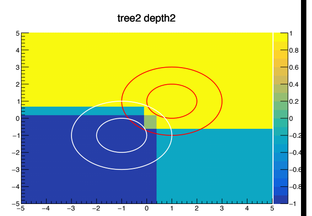
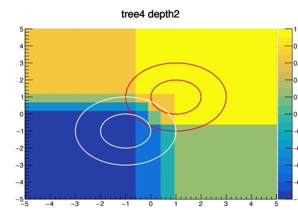
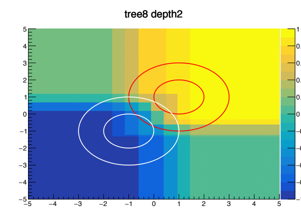
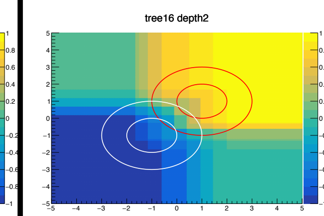
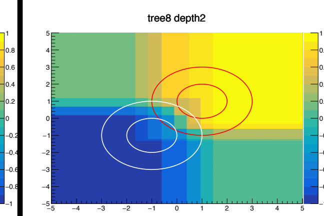
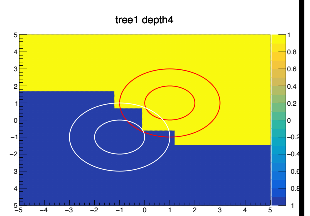
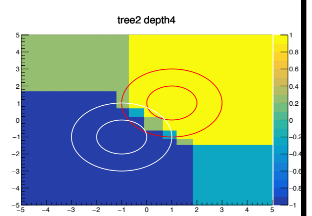
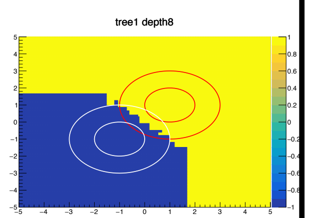
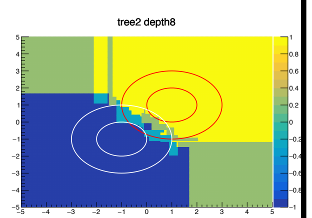
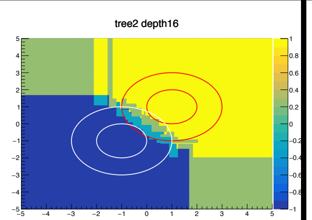


becomes very blurry

Put it together on one slide

Tree →

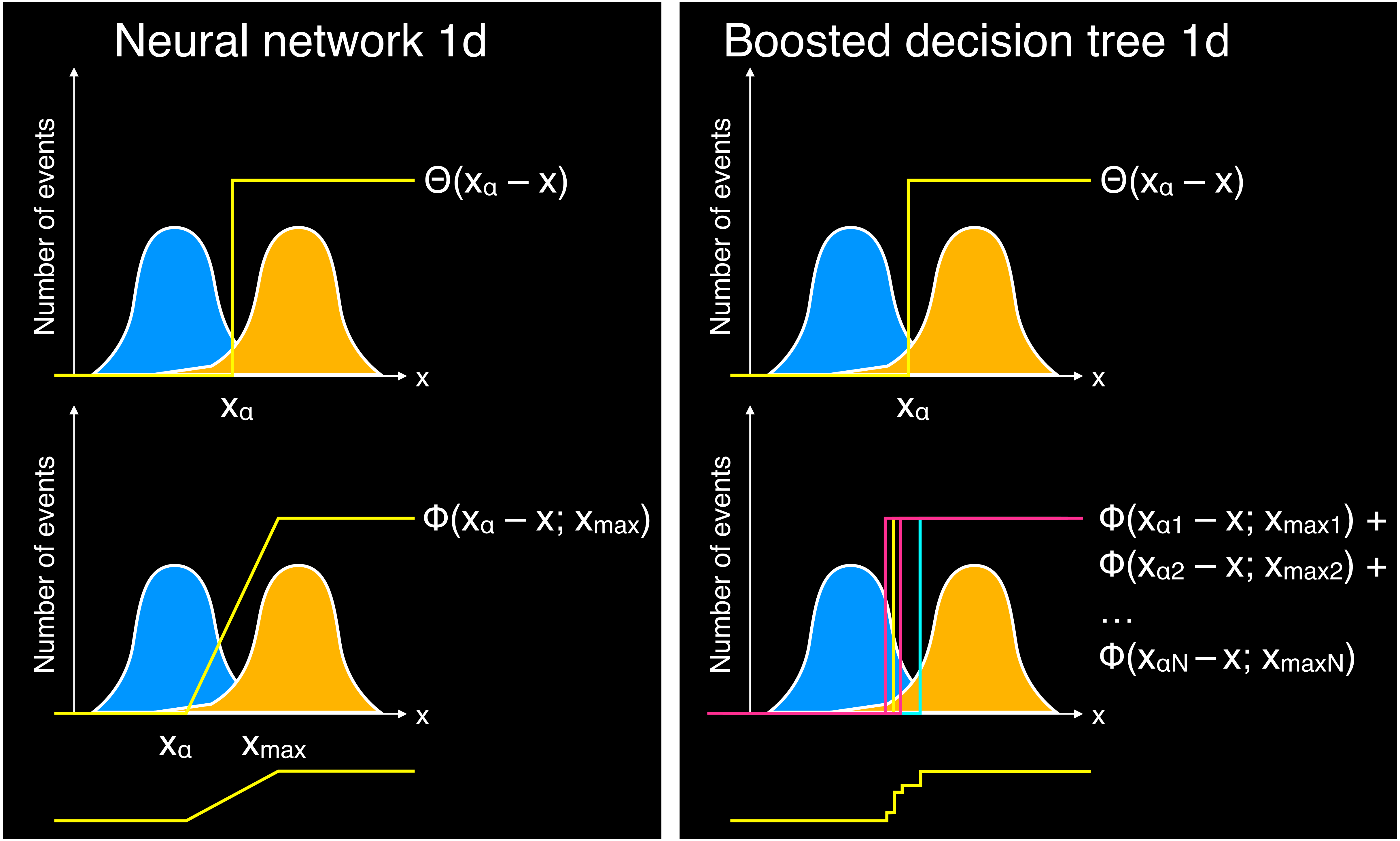
Depth ↓

	1	2	4	8	16	32
1						
2						
4						
8						
16						

Sweet spot depends on the physics problem

Forest of decision trees

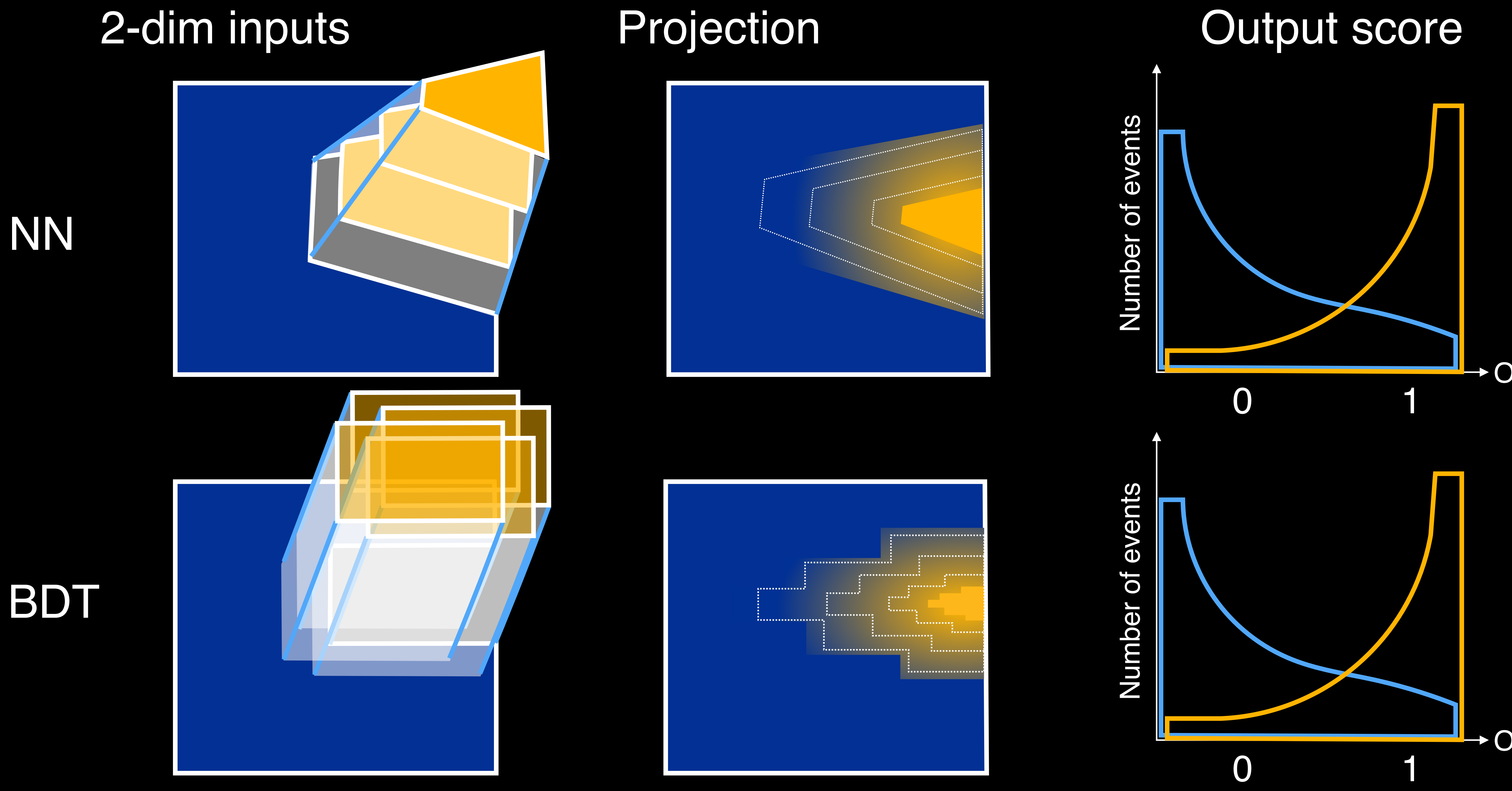
Fuzzy boundary by averaging step functions



Forest of decision trees provides the gradient

Activation function

Fuzzy boundary using a function



Different approach, but same result