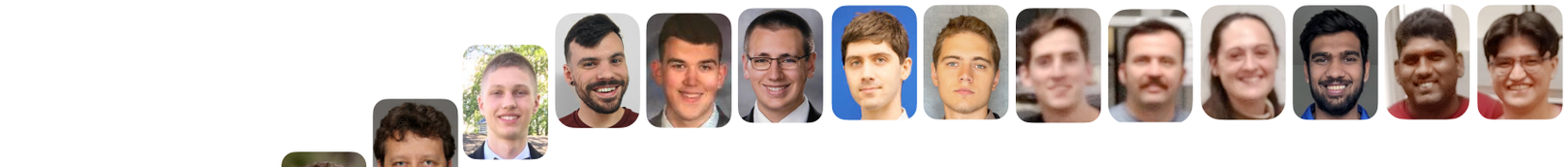
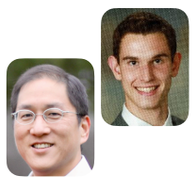
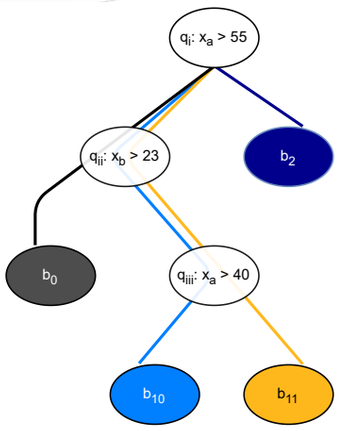


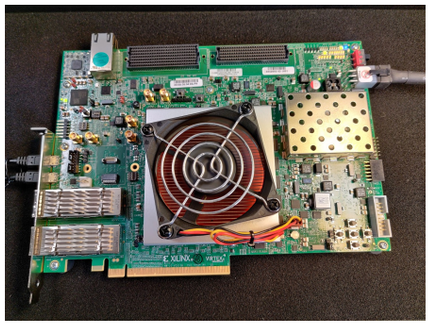
Tae Min Hong



University of Pittsburgh

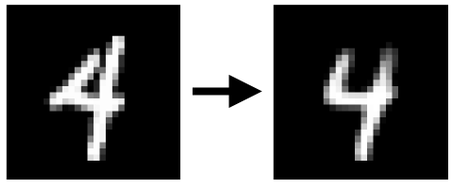


Decision tree



on FPGA to

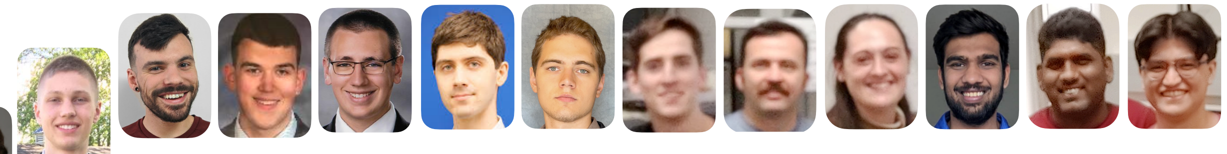
classify
estimate

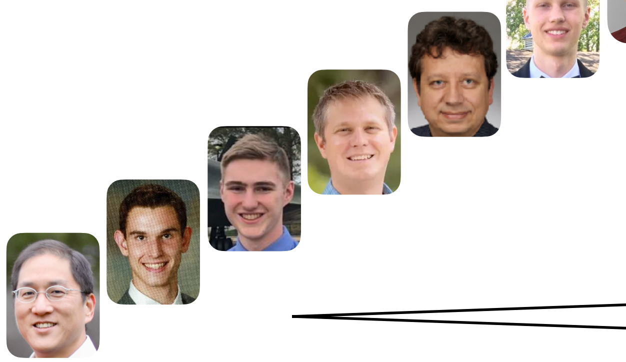


anomaly detect
(+ compress)

ML4FE Workshop, U. Hawaii (remote)
May 20, 2025
https://indico.phys.hawaii.edu/event/2506/contributions/8039/

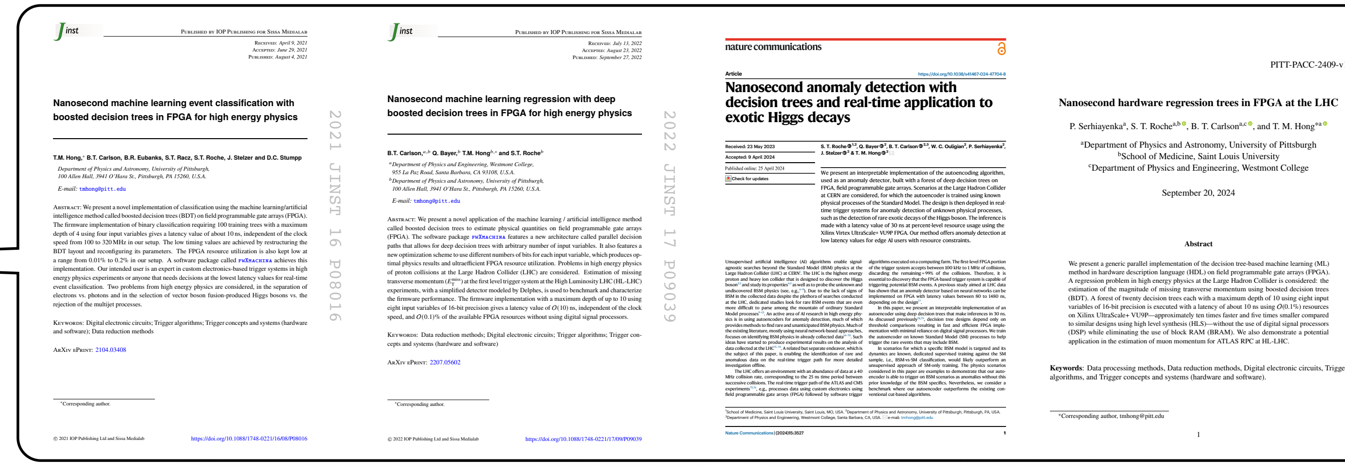
Outline





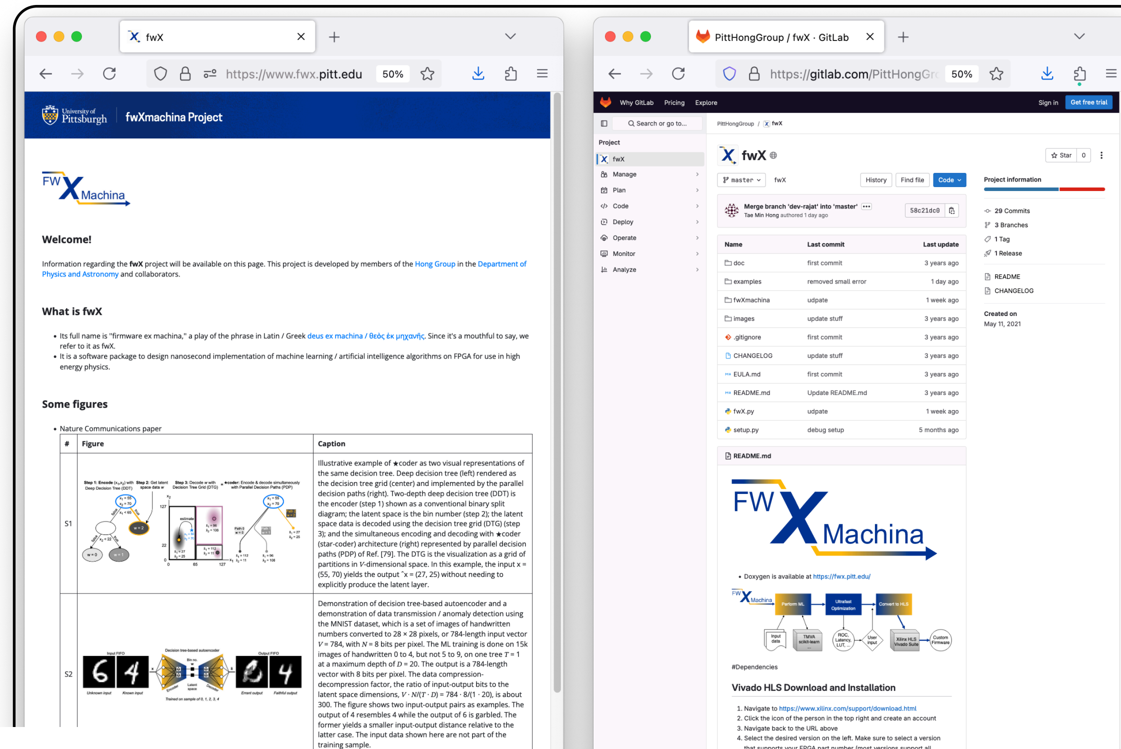
Introduction

- Who we are



FPGA design

- Parallelizing decision trees
- HLS trees → VHDL trees
- Autoencoder ⊕ model distillation



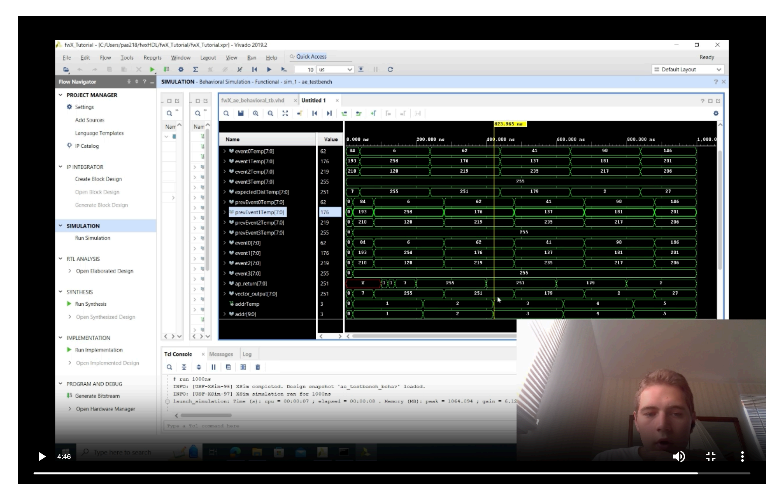
Thoughts on ML4FE

new
- Classification ⊕ Regression

idea
- Anomaly detection ⊕ Data compression

More info

- Code structure & git
- Slides & video tutorials



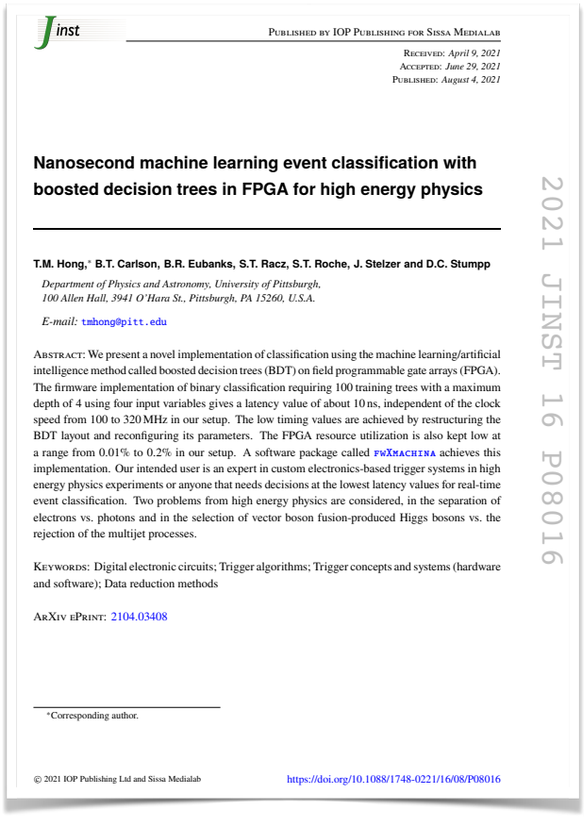
FW X Machina papers

Hong Pittsburgh

(Preliminary) new results

1.Classification

- parallel <u>cuts</u> using 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

2021 JINST 16 P08016

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

Department of Physics and Astronomy, University of Pittsburgh, 100 Allen Hall, 3941 O'Hara St., Pittsburgh, PA 15260, U.S.A.

E-mail: tmhong@pitt.edu

Abstract: We present a novel implementation of classification using the machine learning/artificial intelligence method called boosted decision trees (BDT) on field programmable gate arrays (FPGA). The firmware implementation of binary classification requiring 100 training trees with a maximum depth of 4 using four input variables gives a latency value of about 10 ns, independent of the clock speed from 100 to 320 MHz in our setup. The low timing values are achieved by restructuring the BDT layout and reconfiguring its parameters. The FPGA resource utilization is also kept low at a range from 0.01% to 0.2% in our setup. A software package called `fwXmachina` achieves this implementation. Our intended user is an expert in custom electronics-based trigger systems in high energy physics experiments or anyone that needs decisions at the lowest latency values for real-time event classification. Two problems from high energy physics are considered, in the separation of electrons vs. photons and in the selection of vector boson fusion-produced Higgs bosons vs. the rejection of the multijet processes.

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

ArXiv ePrint: 2104.03408

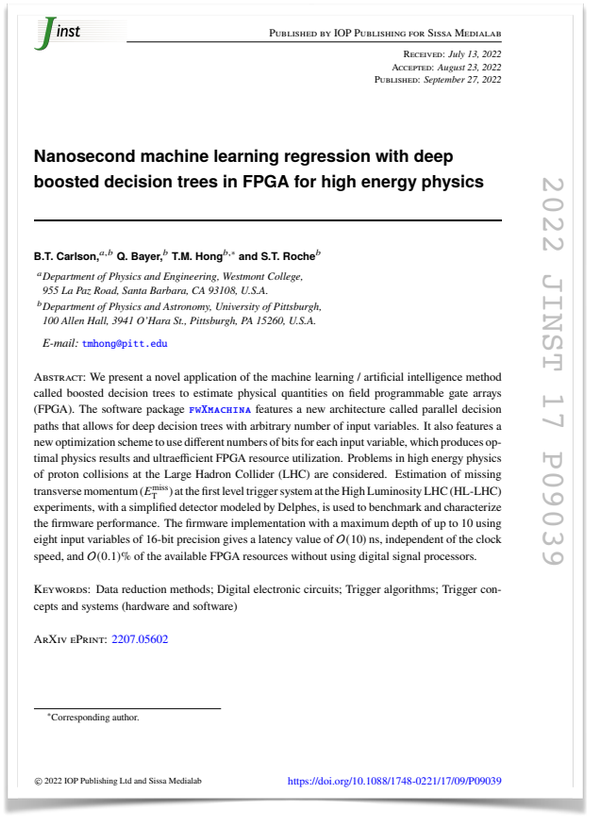
*Corresponding author.

https://doi.org/10.1088/1748-0221/16/08/P08016

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

2.Regression

- parallel <u>paths</u> using HLS



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

2022 JINST 17 P09039

B.T. Carlson,[a,b] Q. Bayer,[b] T.M. Hong[b,*] and S.T. Roche[b]

[a] *Department of Physics and Engineering, Westmont College, 955 La Paz Road, Santa Barbara, CA 93108, U.S.A.*
[b] *Department of Physics and Astronomy, University of Pittsburgh, 100 Allen Hall, 3941 O'Hara St., Pittsburgh, PA 15260, U.S.A.*

E-mail: tmhong@pitt.edu

Abstract: We present a novel application of the machine learning / artificial intelligence method called boosted decision trees to estimate physical quantities on field programmable gate arrays (FPGA). The software package `fwXmachina` features a new architecture called parallel decision paths that allows for deep decision trees with arbitrary number of input variables. It also features a new optimization scheme to use different numbers of bits for each input variable, which produces optimal physics results and ultraefficient FPGA resource utilization. Problems in high energy physics of proton collisions at the Large Hadron Collider (LHC) are considered. Estimation of missing transverse momentum ($E_{\mathrm{T}}^{\mathrm{miss}}$) at the first level trigger system at the High Luminosity LHC (HL-LHC) experiments, with a simplified detector modeled by Delphes, is used to benchmark and characterize the firmware performance. The firmware implementation with a maximum depth of up to 10 using eight input variables of 16-bit precision gives a latency value of $O(10)$ ns, independent of the clock speed, and $O(0.1)$% of the available FPGA resources without using digital signal processors.

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

ArXiv ePrint: 2207.05602

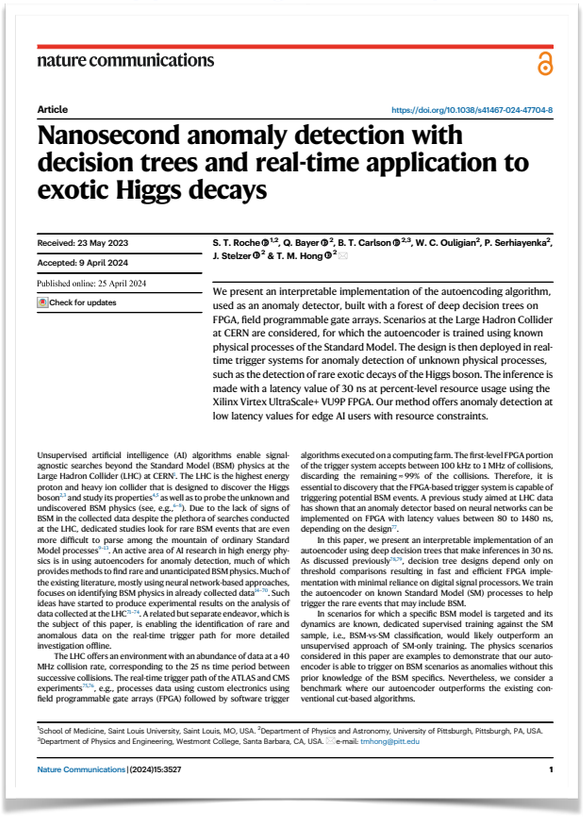
*Corresponding author.

https://doi.org/10.1088/1748-0221/17/09/P09039

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

3.Autoencoder

- in-house training
- bypassing latent space



nature communications

Article

https://doi.org/10.1038/s41467-024-47704-8

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

Received: 23 May 2023
Accepted: 9 April 2024
Published online: 25 April 2024
Check for updates

S. T. Roche[1,2], Q. Bayer[2], B. T. Carlson[2,3], W. C. Ouligian[2], P. Serhiayenka[2], J. Stelzer[2] & T. M. Hong[2]

We present an interpretable implementation of the autoencoding algorithm, used as an anomaly detector, built with a forest of deep decision trees on FPGA, field programmable gate arrays. Scenarios at the Large Hadron Collider at CERN are considered, for which the autoencoder is trained using known physical processes of the Standard Model. The design is then deployed in real-time trigger systems for anomaly detection of unknown physical processes, such as the detection of rare exotic decays of the Higgs boson. The inference is made with a latency value of 30 ns at percent-level resource usage using the Xilinx Virtex UltraScale+ VU9P FPGA. Our method offers anomaly detection at low latency values for edge AI users with resource constraints.

Unsupervised artificial intelligence (AI) algorithms enable signal-agnostic searches beyond the Standard Model (BSM) physics at the Large Hadron Collider (LHC) at CERN. The LHC is the highest energy proton and heavy ion collider that is designed to discover the Higgs boson[1,2] and study its properties[3,4] as well as to probe the unknown and undiscovered BSM physics (see, e.g.,[5–8]). Due to the lack of signs of BSM in the collected data despite the plethora of searches conducted at the LHC, dedicated studies look for rare BSM events that are even more difficult to parse among the mountain of ordinary Standard Model processes[9–11]. An active area of AI research in high energy physics is in using autoencoders for anomaly detection, much of which provides methods to find rare and unanticipated BSM physics. Much of the existing literature, mostly using neural network-based approaches, focuses on identifying BSM physics in already collected data[12–20]. Such ideas have started to produce experimental results on the analysis of data collected at the LHC[21–23]. A related but separate endeavor, which is the subject of this paper, is enabling the identification of rare and anomalous data on the real-time trigger path for more detailed investigation offline.

The LHC offers an environment with an abundance of data at a 40 MHz collision rate, corresponding to the 25 ns time period between successive collisions. The real-time trigger path of the ATLAS and CMS experiments[24,25], e.g., processes data using custom electronics using field programmable gate arrays (FPGA) followed by software trigger algorithms executed on a computing farm. The first-level FPGA portion of the trigger system accepts between 100 kHz to 1 MHz of collisions, discarding the remaining ~99% of the collisions. Therefore, it is essential to discovery that the FPGA-based trigger system is capable of triggering potential BSM events. A previous study aimed at LHC data has shown that an anomaly detector based on neural networks can be implemented on FPGA with latency values between 80 to 1480 ns, depending on the design[26].

In this paper, we present an interpretable implementation of an autoencoder using deep decision trees that make inferences in 30 ns. As discussed previously[26,27], decision tree designs depend only on threshold comparisons resulting in fast and efficient FPGA implementation with minimal reliance on digital signal processors. We train the autoencoder on known Standard Model (SM) processes to help trigger the rare events that may include BSM.

In scenarios for which a specific BSM model is targeted and its dynamics are known, dedicated supervised training against the SM sample, i.e., BSM-vs-SM classification, would likely outperform an unsupervised approach of SM-only training. The physics scenarios considered in this paper are examples to demonstrate that our autoencoder is able to trigger on BSM scenarios as anomalies without this prior knowledge of the BSM specifics. Nevertheless, we consider a benchmark where our autoencoder outperforms the existing conventional cut-based algorithms.

[1]School of Medicine, Saint Louis University, Saint Louis, MO, USA. [2]Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh, PA, USA. [3]Department of Physics and Engineering, Westmont College, Santa Barbara, CA, USA. e-mail: tmhong@pitt.edu

Roche et al.
Nat. Comm. **15** (2024) 3527
http://doi.org/10.1038/s41467-024-47704-8

4.Hardware trees

- faster & more efficient
- in VHDL, no more HLS

Nuclear Instruments and Methods in Physics Research A 1072 (2025) 170209

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

Full Length Article

Nanosecond hardware regression trees in FPGA at the LHC

Pavel Serhiayenka[a], Stephen T. Roche[a,b], Benjamin T. Carlson[a,c], Tae Min Hong[a,*]

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

ARTICLE INFO

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

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 (LHC) 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 ten using eight input variables of 16-bit precision is executed with a latency of less than 10 ns using $\mathcal{O}(0.1\%)$ 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 the High-Luminosity LHC (HL-LHC).

1. Introduction

Deep architectures for machine learning (ML) methods continue to empower high energy physics experiments, such as the ATLAS [1] and CMS [2] experiments at the Large Hadron Collider (LHC) [3]. Various simplified approaches for ML designs aimed for trigger systems using field programmable gate arrays (FPGA) exist in the literature. Many use high level synthesis (HLS), in which C-like syntax is converted to an register-transfer level (RTL) circuit using a translator provided by the vendor. The HLS version of neural networks by hls4 ml [4] gave rise to a body of work in the implementation of artificial intelligence (AI), such as autoencoders and convolutional neural networks [5,6], on firmware. Similarly, the HLS version of decision trees [7,8] by `fwXmachina` (developed by us) also gave rise to an autoencoder [9]. There are also non-HLS approaches using hardware description language (HDL) for neural networks [10] and decision trees [11], as well as more dedicated applications, e.g., [12,13].

In this paper, we present a more efficient implementation of decision trees with lower latency. We illustrate the physics potential using the problem in our previous paper [14], the estimation of missing transverse momentum ($E_{\mathrm{T}}^{\mathrm{miss}}$) at the LHC. $E_{\mathrm{T}}^{\mathrm{miss}}$ is an important signature of physics with minimal interaction with the detector materials, including neutrinos and beyond the Standard Model (BSM) particles, including dark matter [15,16] and supersymmetry (SUSY) [17–19]. The High-Luminosity LHC (HL-LHC) [20] upgrades of the trigger systems of ATLAS and CMS experiments [21,22] provide an opportunity to implement AI/ML with deeper designs for a variety of applications.

This paper is organized as follows. Section 2 describes the ML training and firmware implementation on FPGA. Section 3 presents the results, with comparisons to previous publications. Section 4 concludes. Appendix A provides the technical details of the subscore adder. Appendix B presents a study, following Ref. [23], of the estimation of muon momentum at the HL-LHC using hits in the resistive plate chamber (RPC) subdetector in the ATLAS level-0 trigger system.

2. Method

The ML training is described followed by the firmware design. `TMVA` [24] is used to train a boosted decision tree (BDT) for regression with the "truth" generator-level $E_{\mathrm{T}}^{\mathrm{miss}}$ as the target variable. The setup follows our previous publication [8] and uses the data samples described therein. These variables include four calculations of the event $E_{\mathrm{T}}^{\mathrm{miss}}$ using calorimeter deposits, hadronic jets, tracks from charged particles, and four estimates of the total transverse energy deposited in the detector.

The firmware design is a VHDL adaptation of the Deep Decision Tree Engine (DDTE) originally designed in HLS [8]. In Python, the `fwXmachina` program writes VHDL that reflects a given ML configuration from the training. In Fig. 1, each tree is represented by a HDL Tree Engine (HTE) and the forest is managed by the HDL Tree Manager. The summing of the subscores is either clocked or combinational; see Appendix A. We note that operations are triggered by the rising edge

* Corresponding author.
E-mail address: tmhong@pitt.edu (T.M. Hong).

https://doi.org/10.1016/j.nima.2025.170209
Received 1 October 2024; Received in revised form 25 November 2024; Accepted 7 January 2025
Available online 10 January 2025

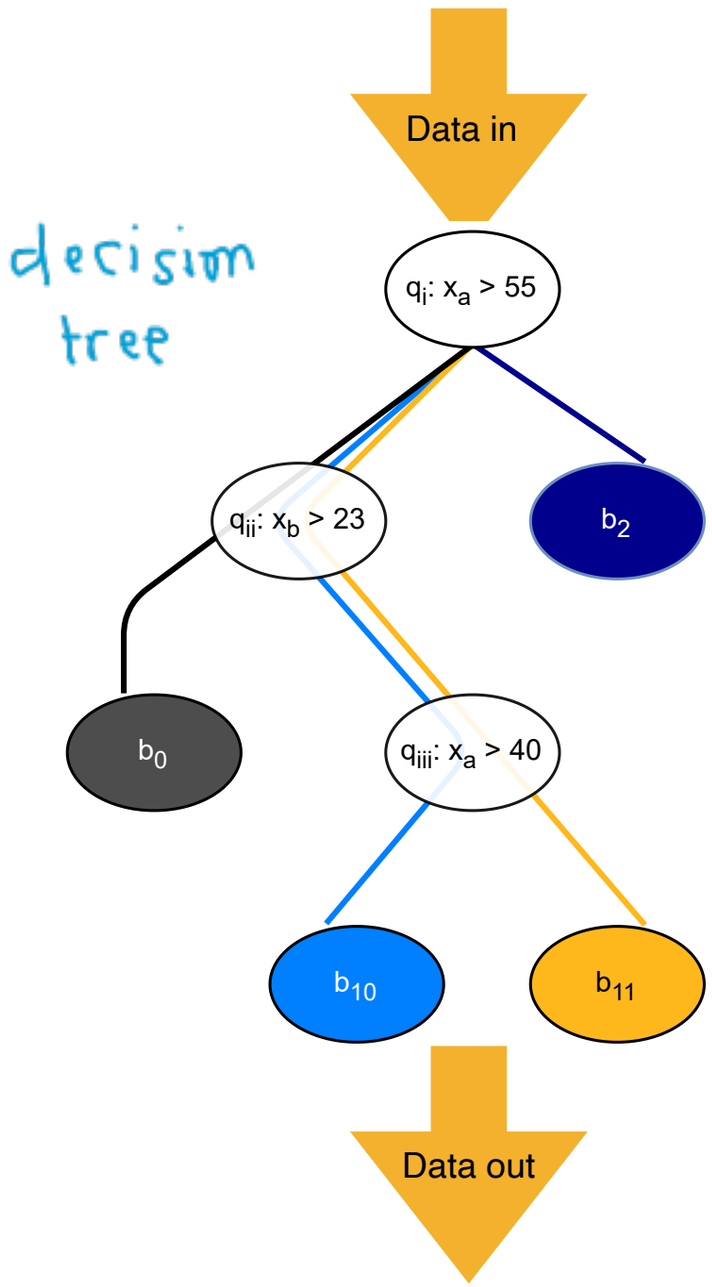
Serhiayenka et al.
NIM A **1072** (2025) 170209
http://doi.org/10.1016/j.nima.2025.170209

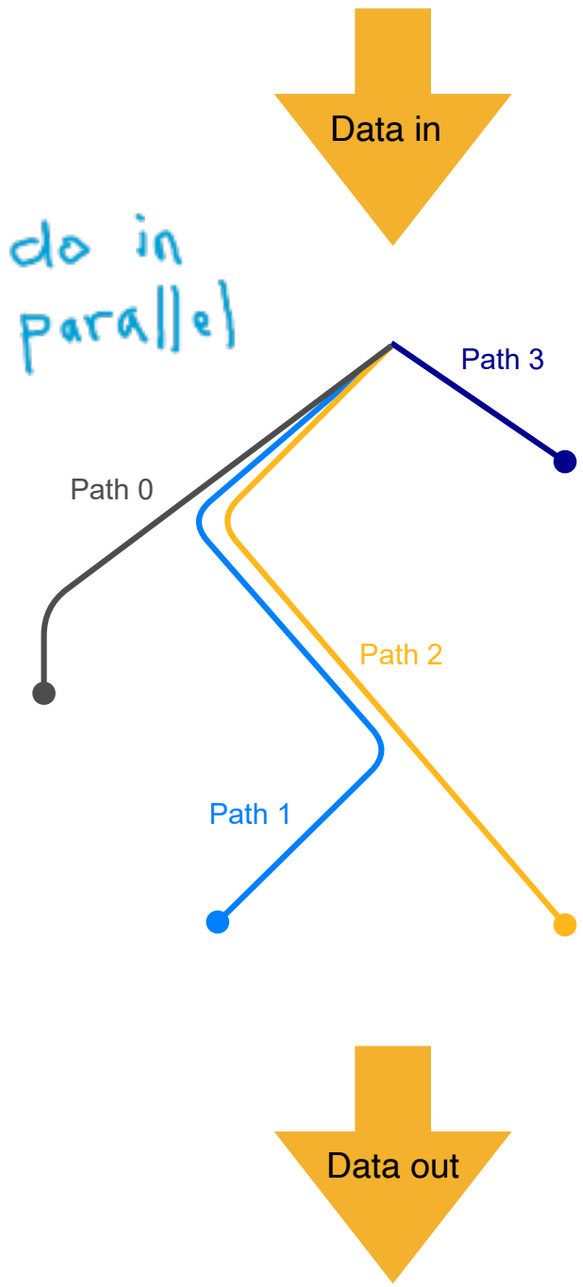
Parallel paths

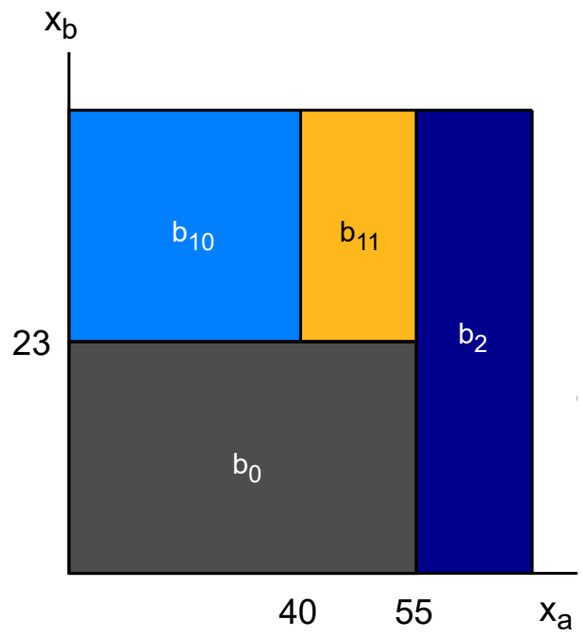
Hong Pittsburgh

Implementation of decision tree on FPGA

- B. Carlson et al., J. Instrumentation 17 (2022) P09039







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}

Classification ⊕ estimation

Hong
Pittsburgh

Example: Hard-scatter jet vs. Pileup jet ⊕ E_T regression

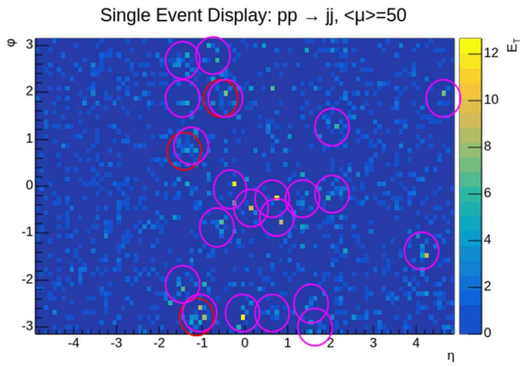
- S. Roche et al., Pheno May 19, 2025: https://indico.global/event/812/contributions/126530/ (paper in preparation)

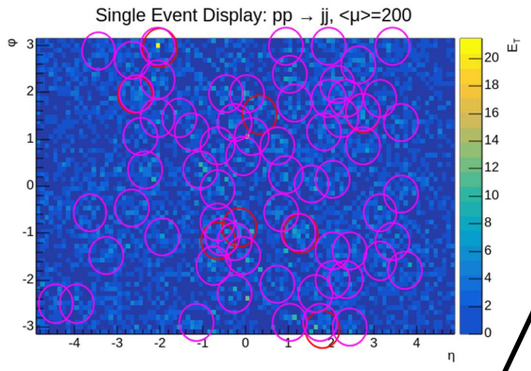
Effect of increased pileup

Stephen Roche

Markedly increases the number of jet candidates!

Makes L1 triggering on jets more difficult



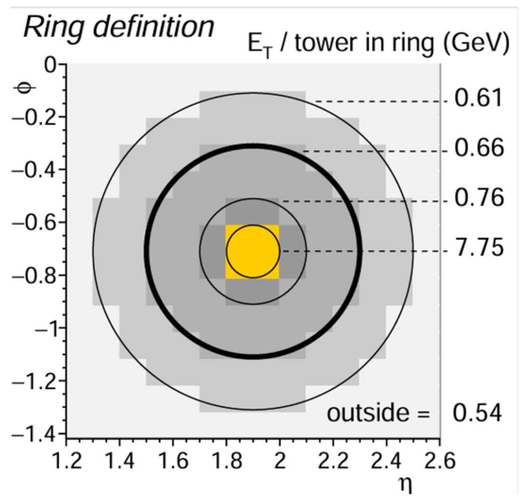


Goal 1:

Identify which sliding window jets are from pileup vs which are from hard scatter

Goal 2:

Accurately estimate the jet's true E_T in the high pileup environment



Machine learning inputs:

- E_T, E_T^{had}, E_T^{EM} for
- Ring center → 0.1
- Ring center → 0.4
- 0.1 → 0.2
- 0.2 → 0.4
- 0.4 → 0.6
- Ring Center Eta

Machine learning model:

30 trees, maximum depth 8

HL-LHC

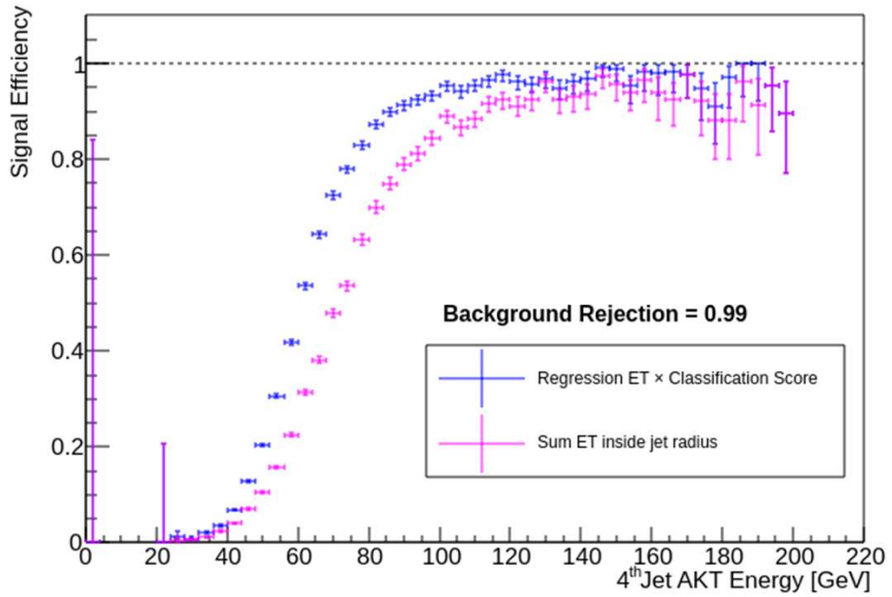
- Jets (red is anti-kt jet, magenta is sliding window)
- Study done using MadGraph+Delphes (not ATLAS)

Method

- Draw rings around L1 sliding window
- Feed two BDT (classify + estimate), multiply result

Physics result

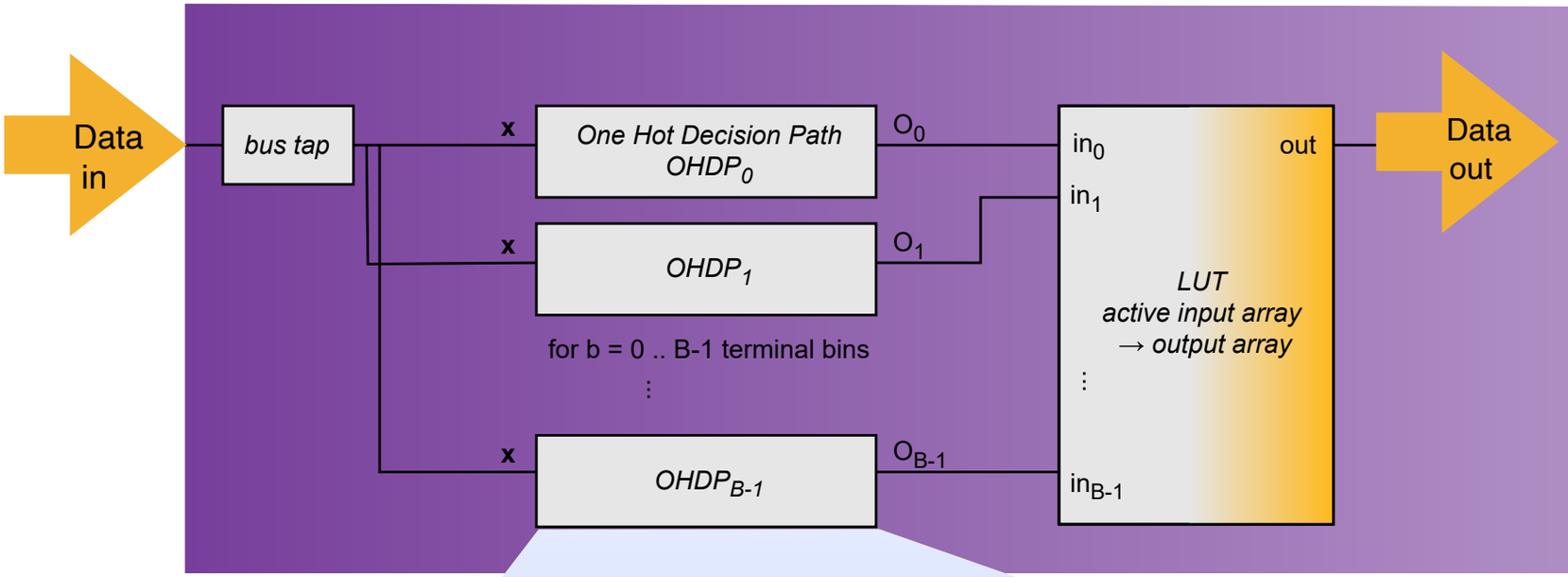
- 20 GeV shift in 4th jet turn-on curve (for HH → 4b)

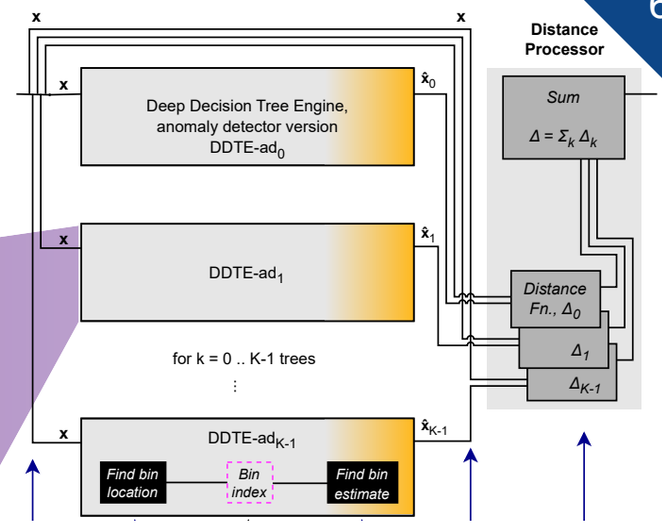


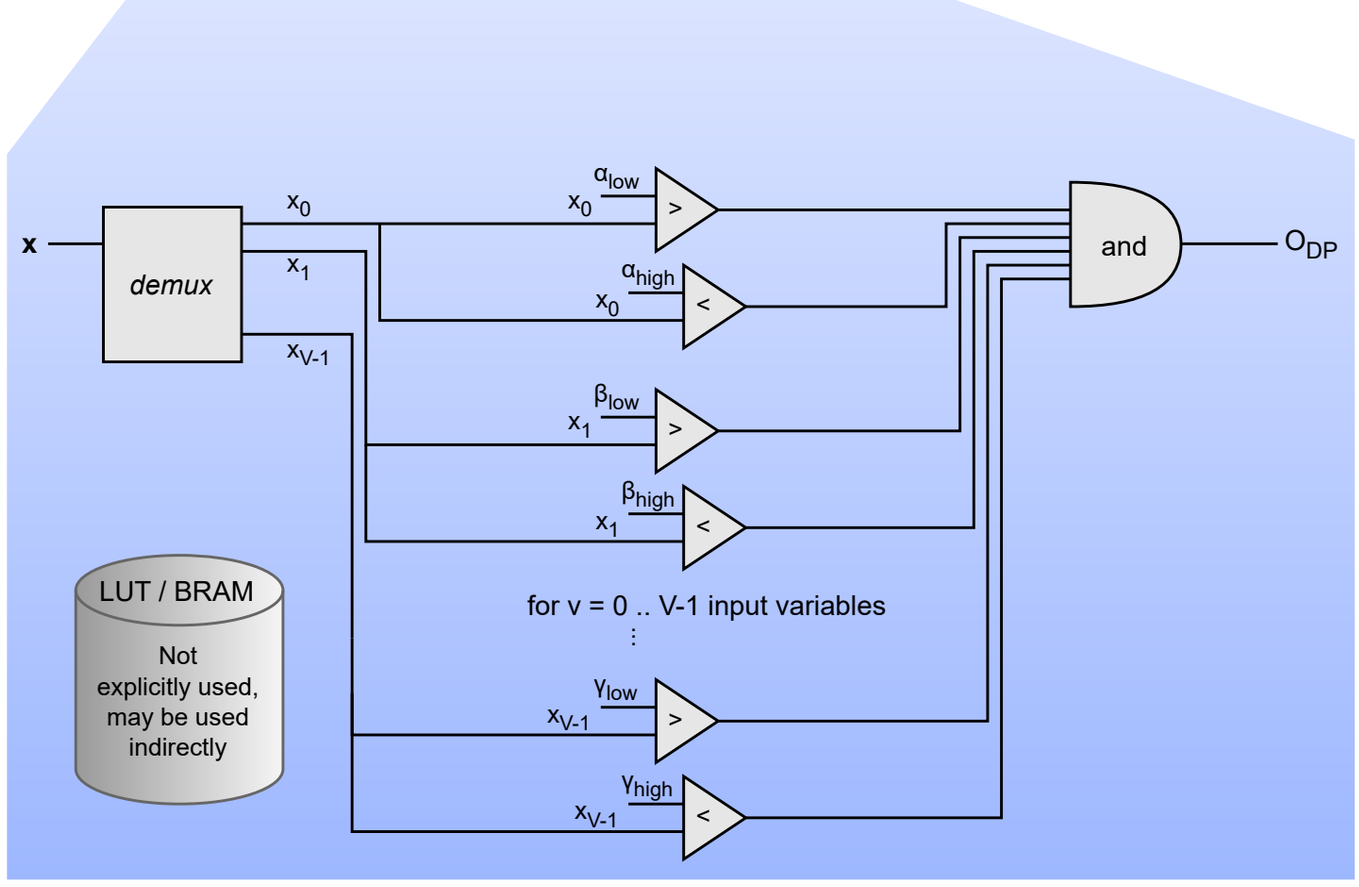
NB. This is a pheno paper, not an ATLAS result

FPGA results

- Parallel paths implementation in VHDL, Serhiayenka et al., NIM A **1072** (2025) 170209







Using Xilinx Ultrascale+ VU9P (vcu118) at 200 MHz

Feature	Value
Latency	2 clock ticks (50 ns)
Interval	1 clock tick (25 ns)
Flip-flops (FF)	10399 (0.44%)
Look-up tables (LUT)	13274 (1.1%)
Digital signal processors (DSPs)	0
Block-RAM (BRAM)	9 (0.36%)
Ultra-RAM (URAM)	0

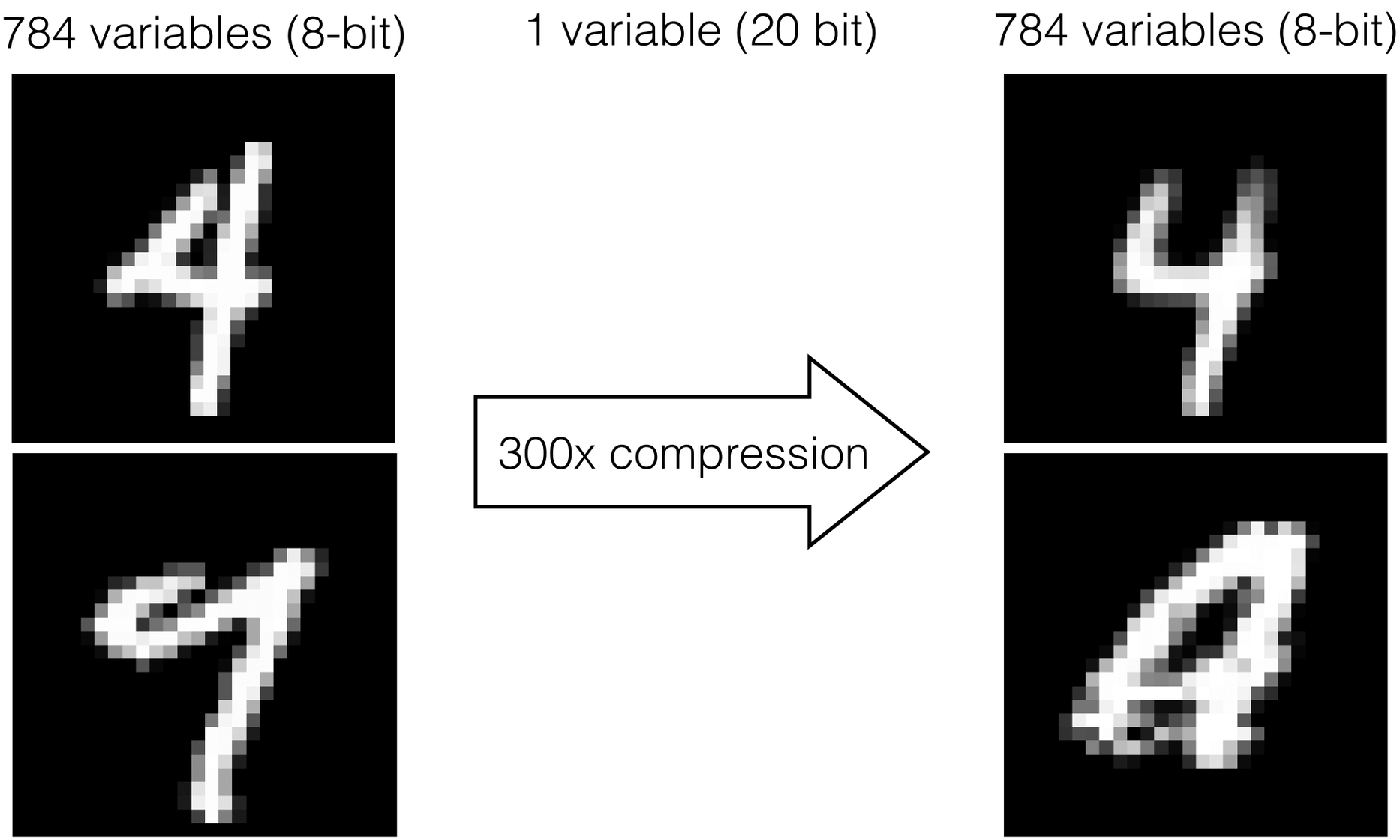
Hong Pittsburgh

Autoencoder intro

TM Hong

Example: handwritten numbers

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



Details

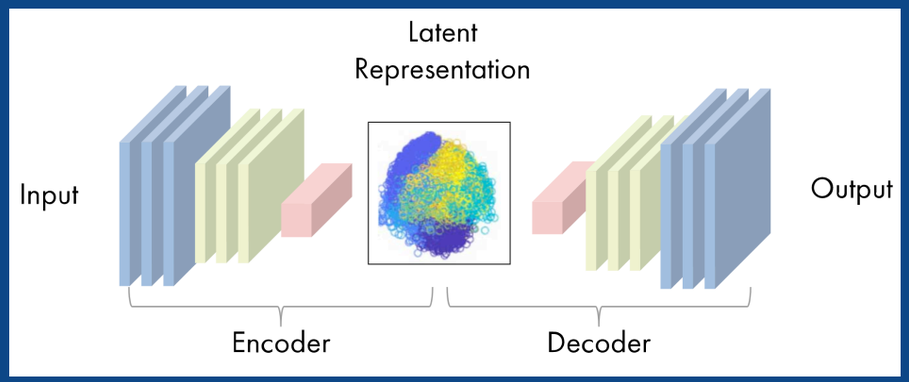
- Input-output distance is relatively small = good compression
- Input-output distance is relatively large = bad compression

Tree autoencoder, What?!

TM Hong

NN AE

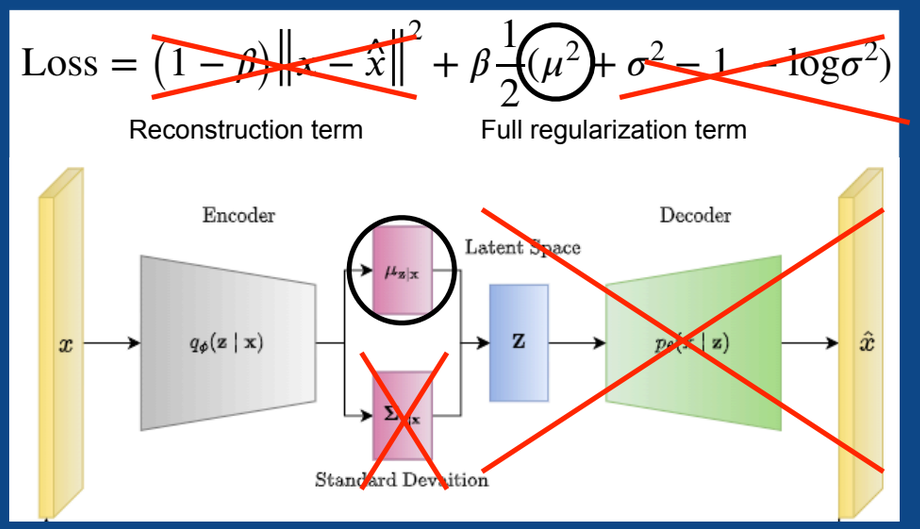
- Training is a black box, done offline
- Latent space is complex



From CMS Machine Learning Group
https://cms-ml.github.io/documentation/training/autoencoders.html

- FPGA version simplified for anomaly at CMS

From CMS Public Note, DP-2023/079
https://cds.cern.ch/record/2876546/files/DP2023_079.pdf



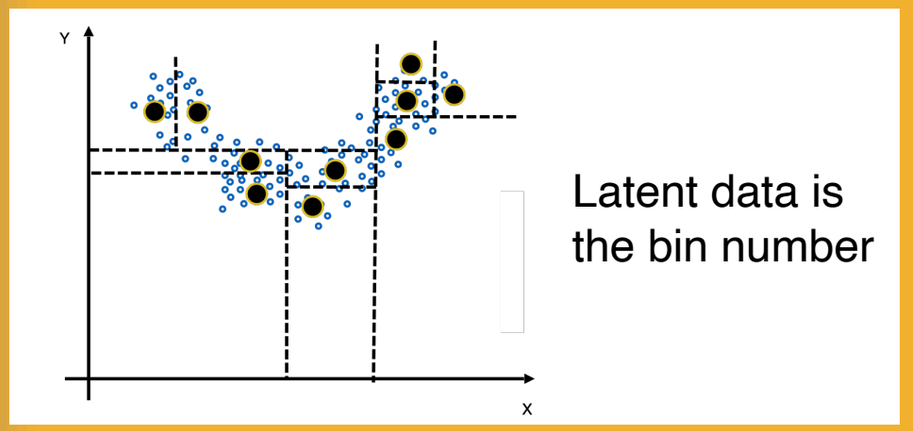
Cross-outs are mine

Cross-outs are mine

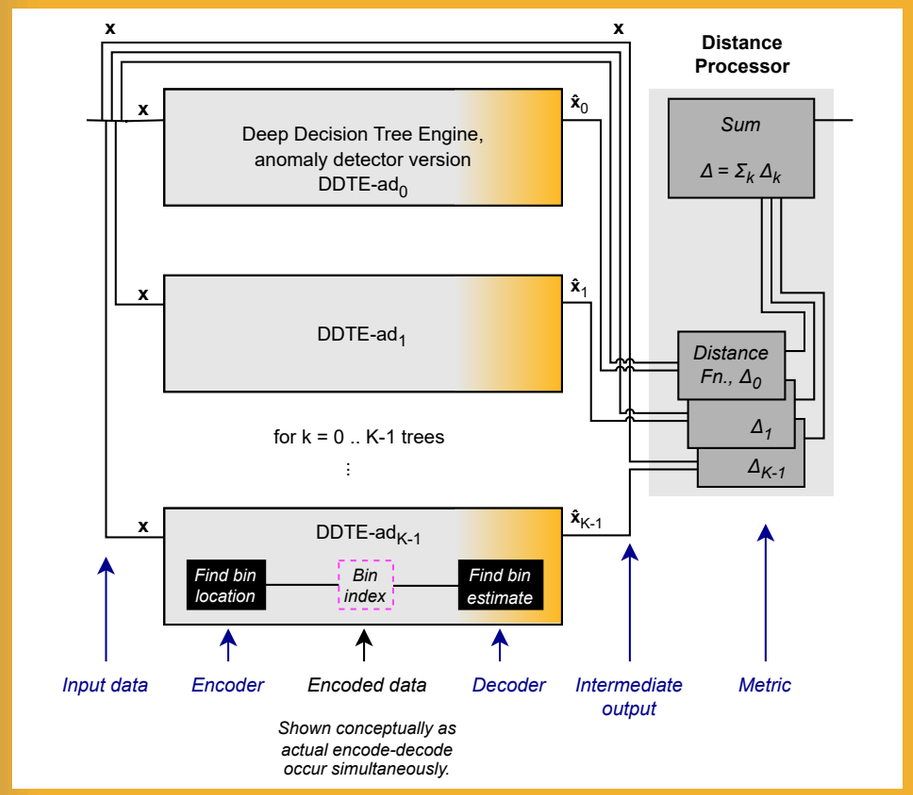
Image from
https://medium.com/@rushikesh.shende/autoencoders-variational-autoencoders-vae-and-β-vae-ceba9998773d

"Starcoder" tree AE

- Training is sampling of 1d pdfs
- Latent space is simple / interpretable



- FPGA version can optionally skip latent sp.



Training developed my us

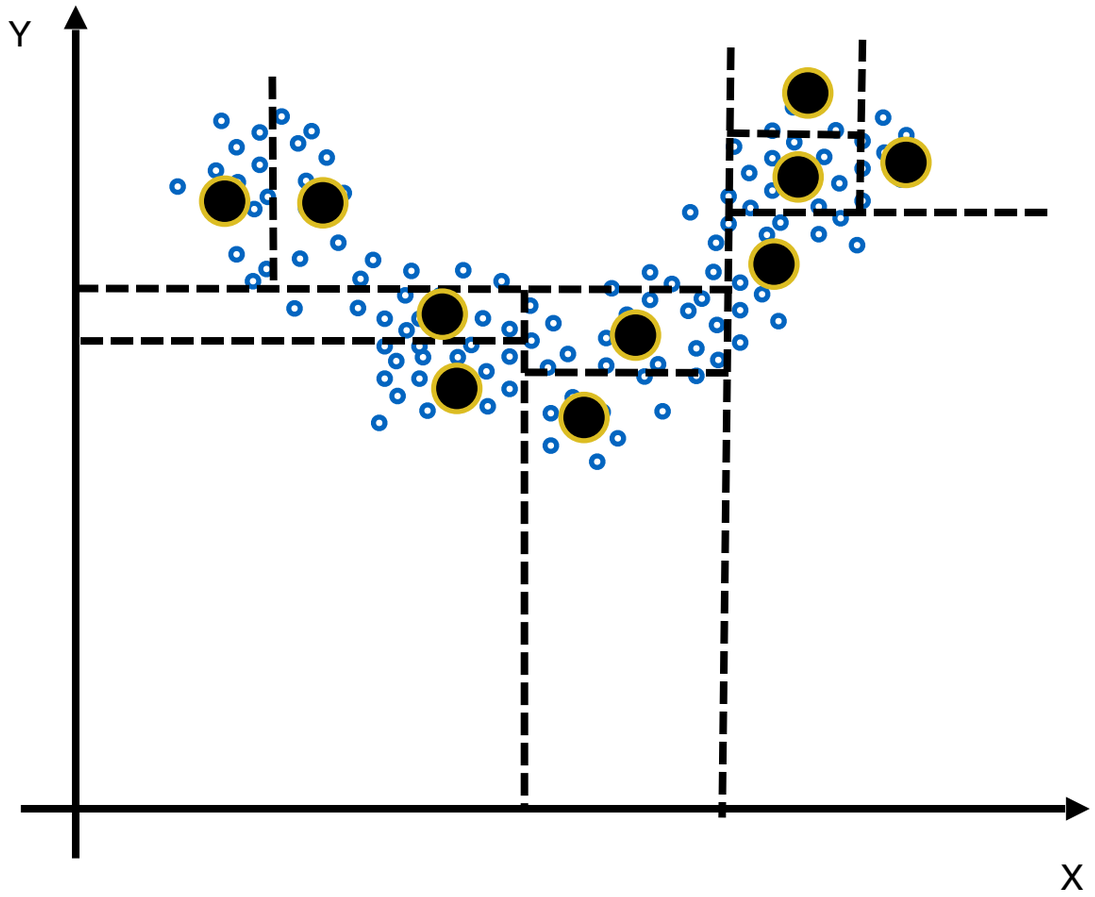
TM Hong

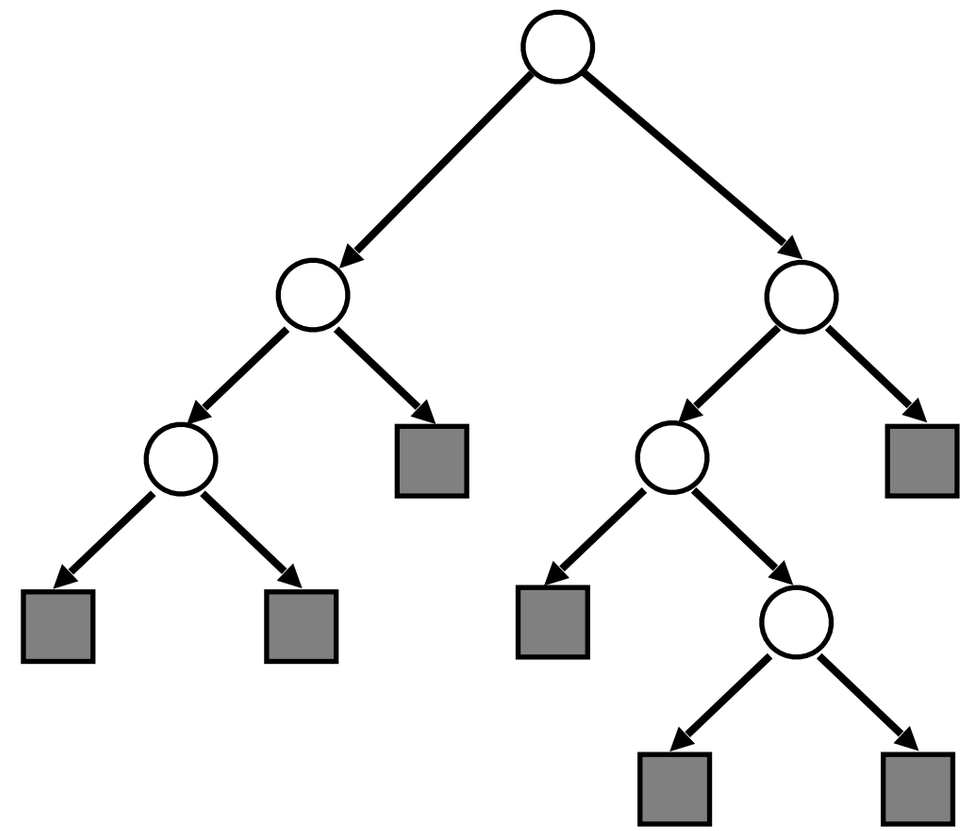
Train by sampling 1d projections

- Encoding: Event → which bin it's in

Decoding returns "reconstruction point"

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



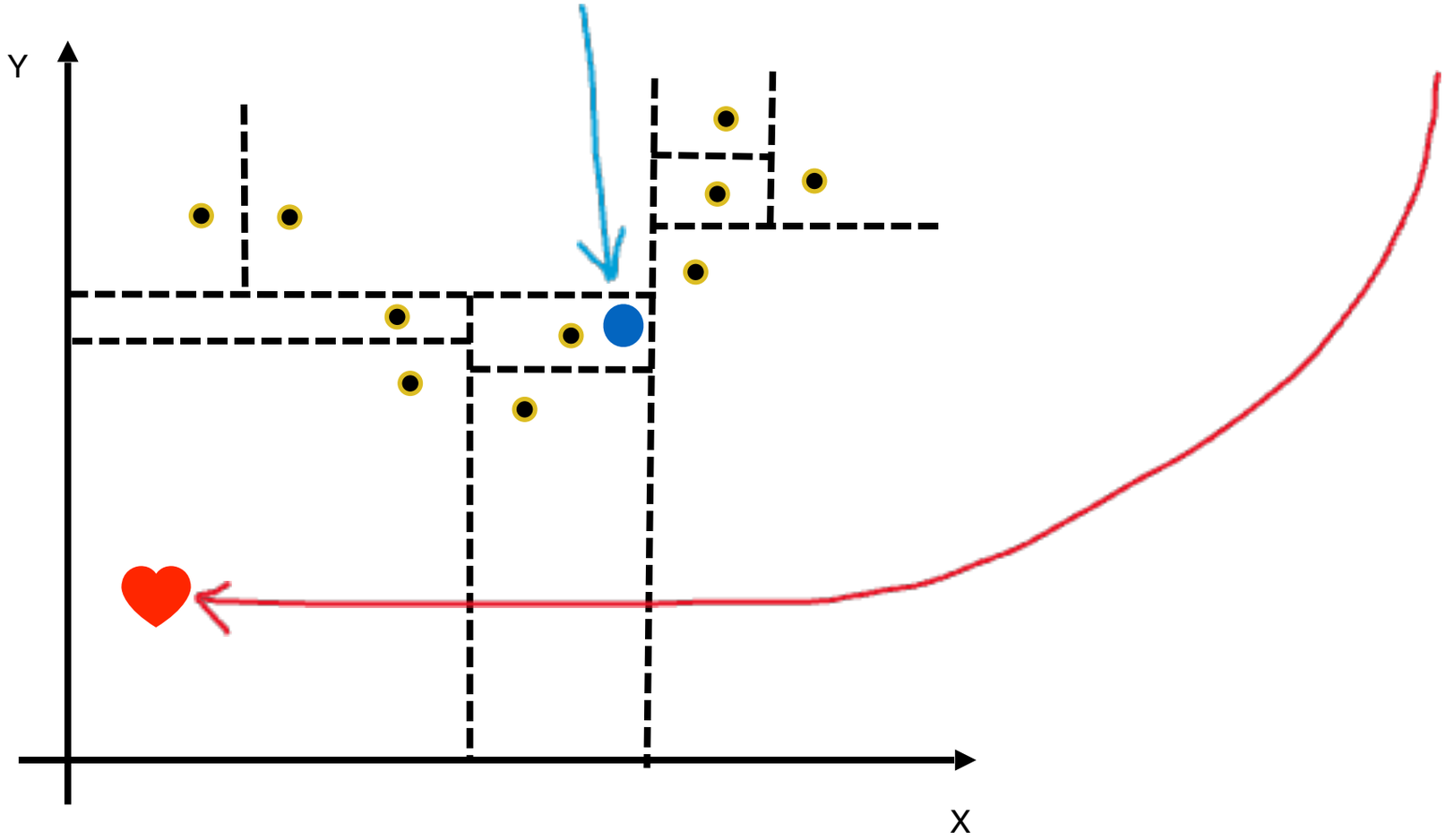


AE to anomaly detector

TM Hong

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

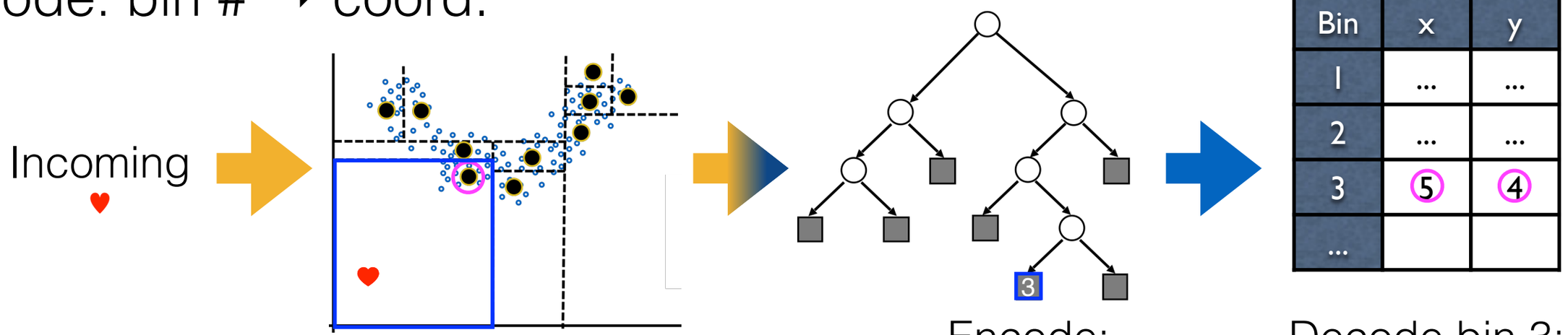


Realized we can skip latent space

TM Hong

Decode?

- Encode: input var → bin #
- Decode: bin # → coord.

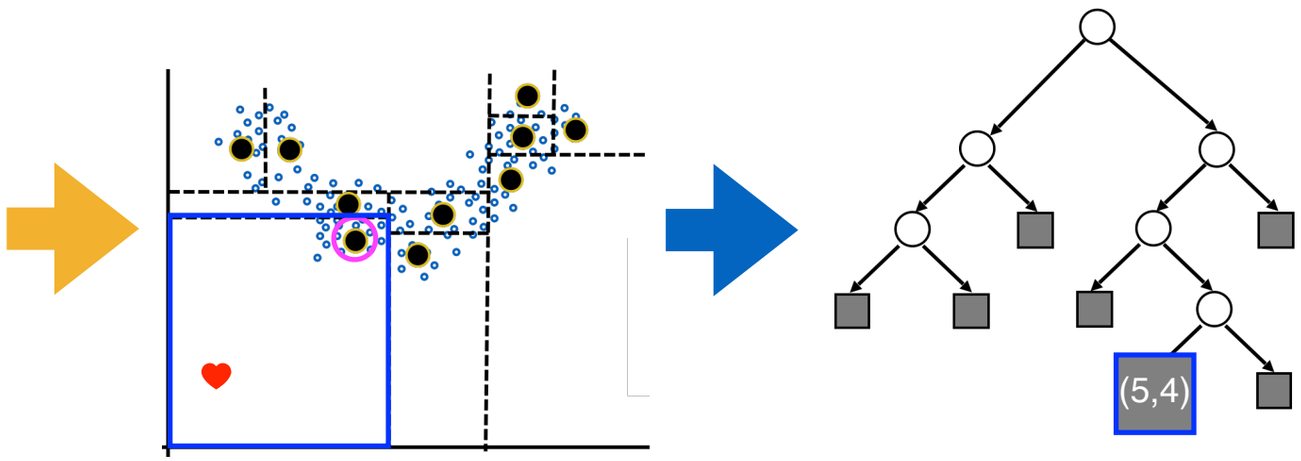


Encode: return bin 3

Decode bin 3: return (5,4)

No need to encode

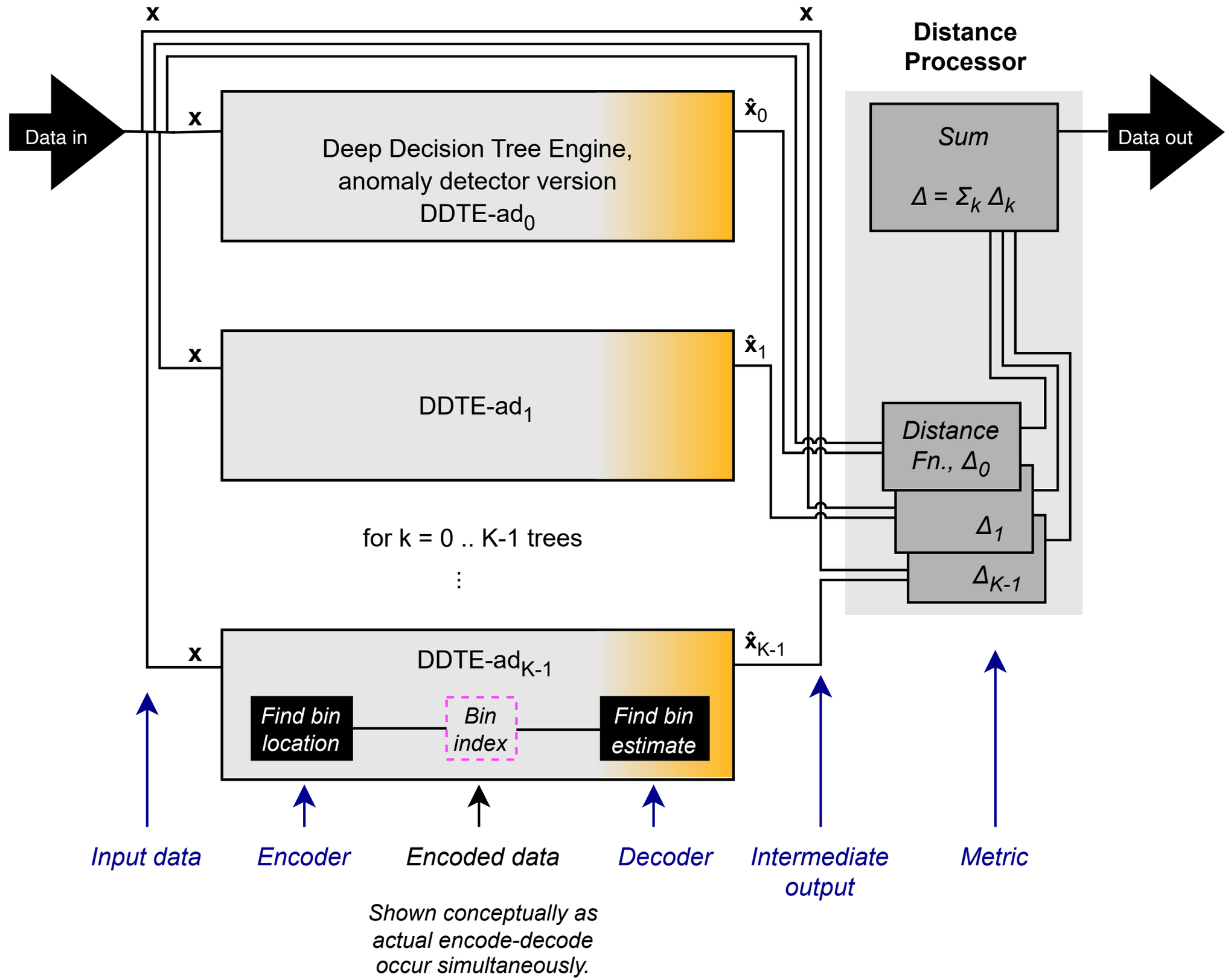
- Starcode: input var → coord.



Encode is Decode: return (5,4)

Block diagram

TM Hong



Starcoder vs. hls4ml

TM Hong

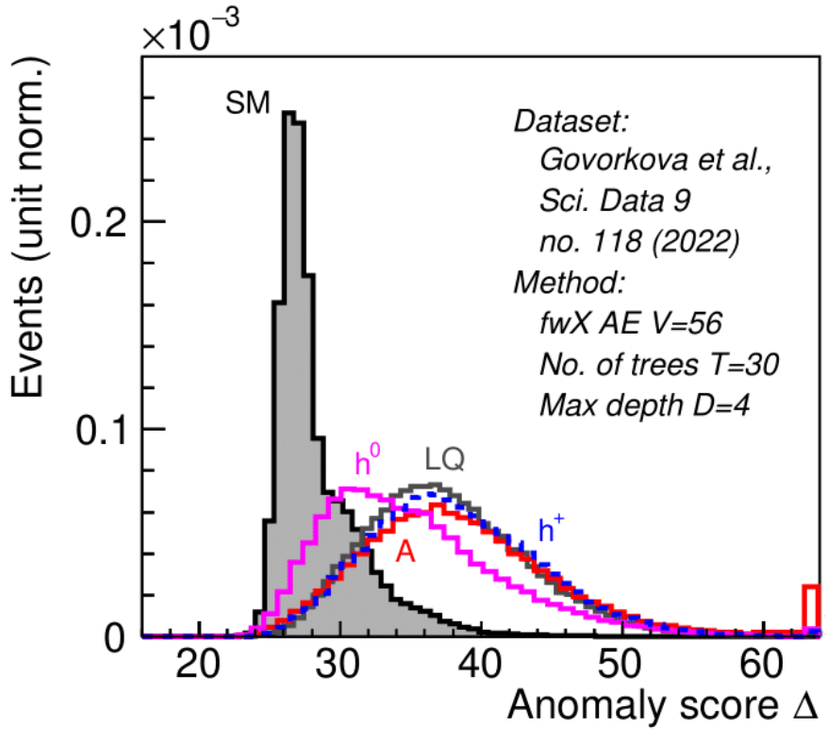
Works well

- Physics (plots)
- FPGA (table)

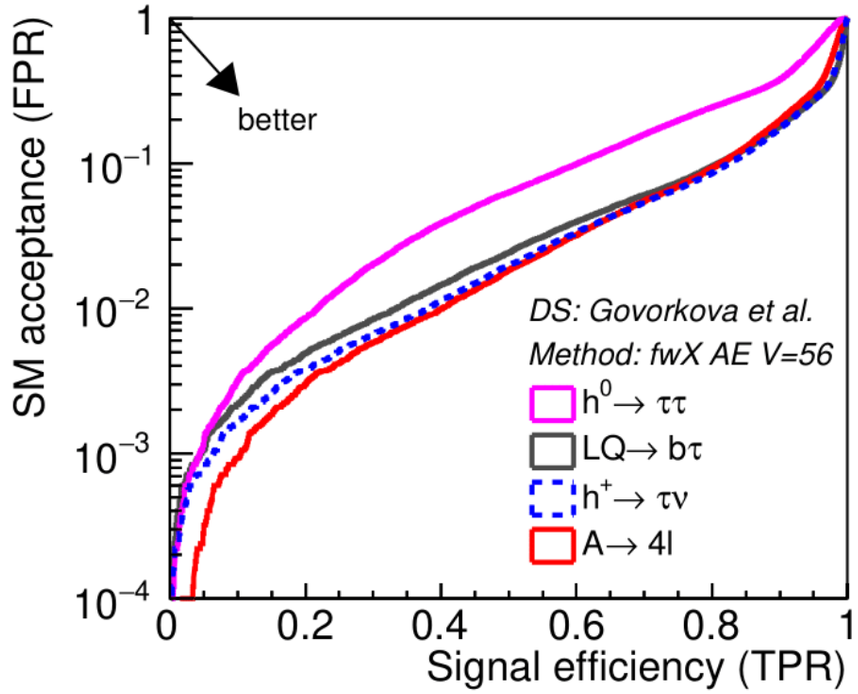
Comparison

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

Distribution



ROC curve



	hls4ml	starcoder
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

Key take-away:

This result uses HLS trees. Using VHDL trees projected to be smaller by 2-5x (preliminary).

VAE model distillation

Hong
Pittsburgh

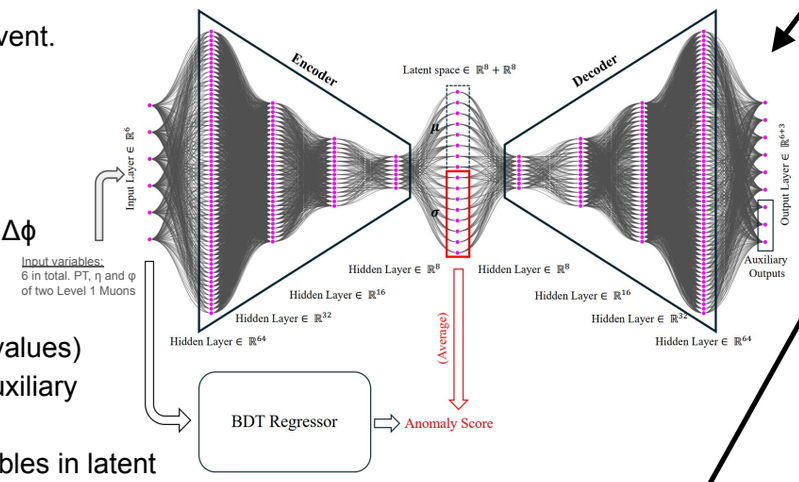
Convert NN model with DT

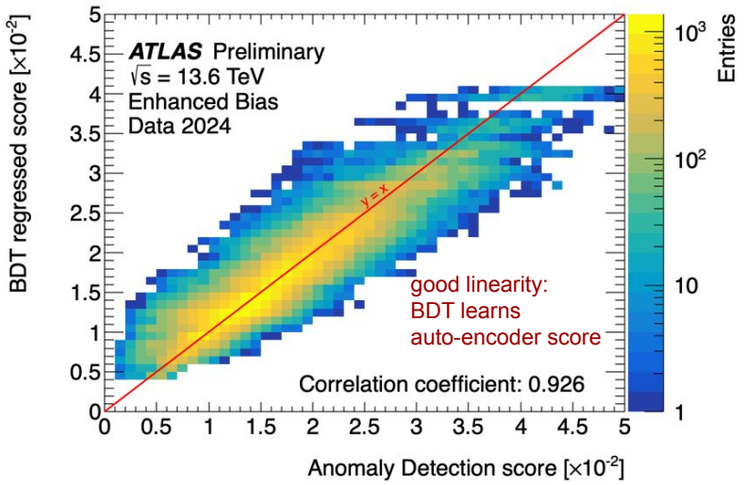
- ATLAS work presented by R. Gupta today at Pheno 2025, https://indico.global/event/812/contributions/126571

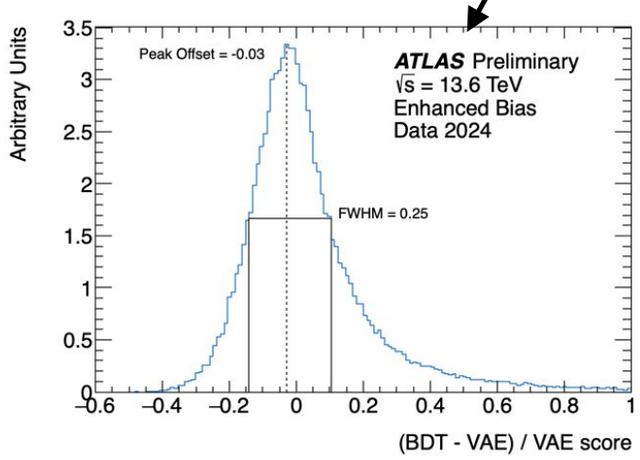
Training: Variational Autoencoder

R. Gupta 8

- Data: enhanced bias 2024 data (ATL-DAQ-PUB-2016-002)
- Preselection: 2MU3VF
- At most 3 leading muons (leading in p_T) per event.
- Train on unlabeled dimuon events (p_T, η, ϕ of muon pairs) (details in backup).
- Consider all combinations (e.g. 3 muon pairs) for one event.
- Regressed variables in the model: mass, ΔR, Δϕ
- VAE architecture:
  - Encoder: 4 layers
  - Latent space: 8D Gaussian (μ + σ → 16 values)
  - Decoder reconstructs inputs + predicts auxiliary vars ($m_{\mu\mu}$, ΔR, |Δϕ|)
- Anomaly score = average value over 8σ variables in latent space.
- train a BDT regression to learn the NN score
- Same 6 variables as inputs for the BDT (2 muons' P_T, η, φ)
- BDTs trained with TMVA: 200 trees, Depth of 20.







Source: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/MuonTriggerPublicResultsNoMAD_Nanosecond_Anomaly_detecti

Now at ATLAS

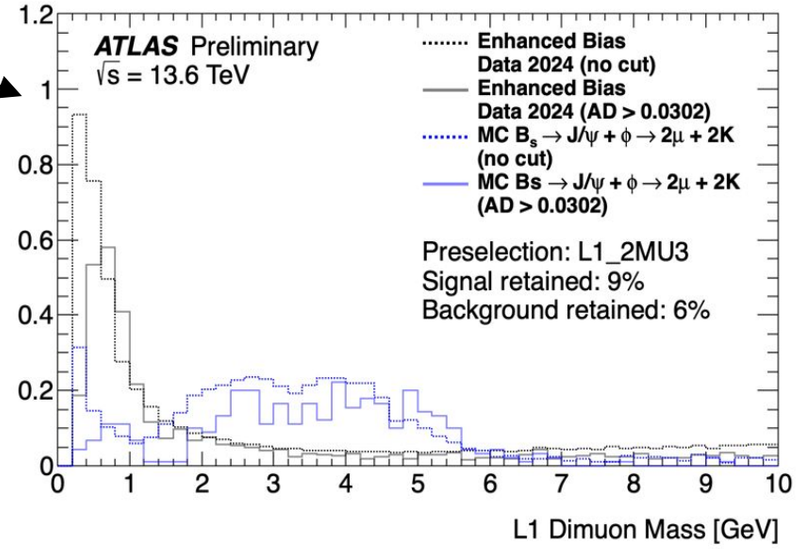
- Train VAE for 3 muons

Method

- Chop-off the decoder
- Regress the latent space variables

Physics result

- Unique B physics signal at L1



NB. This is ATLAS result I'm quoting

Compression

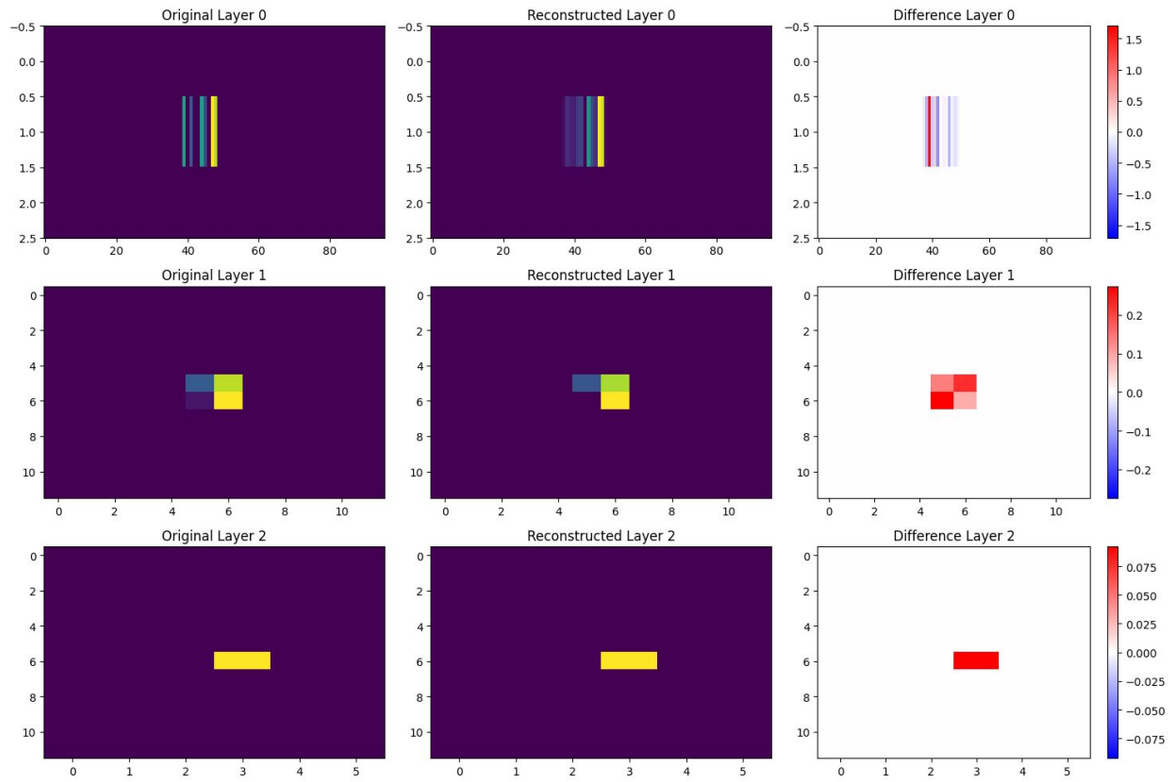
Hong
Pittsburgh

Jet images in multiple calorimeter layers

- Study by R. Gupta, paper in preparation

Comparison (original vs reconstructed)

R. Gupta 12



Latent dim = 64
Compression ~ 8x

Python-based code

Hong Pittsburgh

Availability

- gitlab.com/PittHongGroup/fwX

  parallel cuts (paper 1)

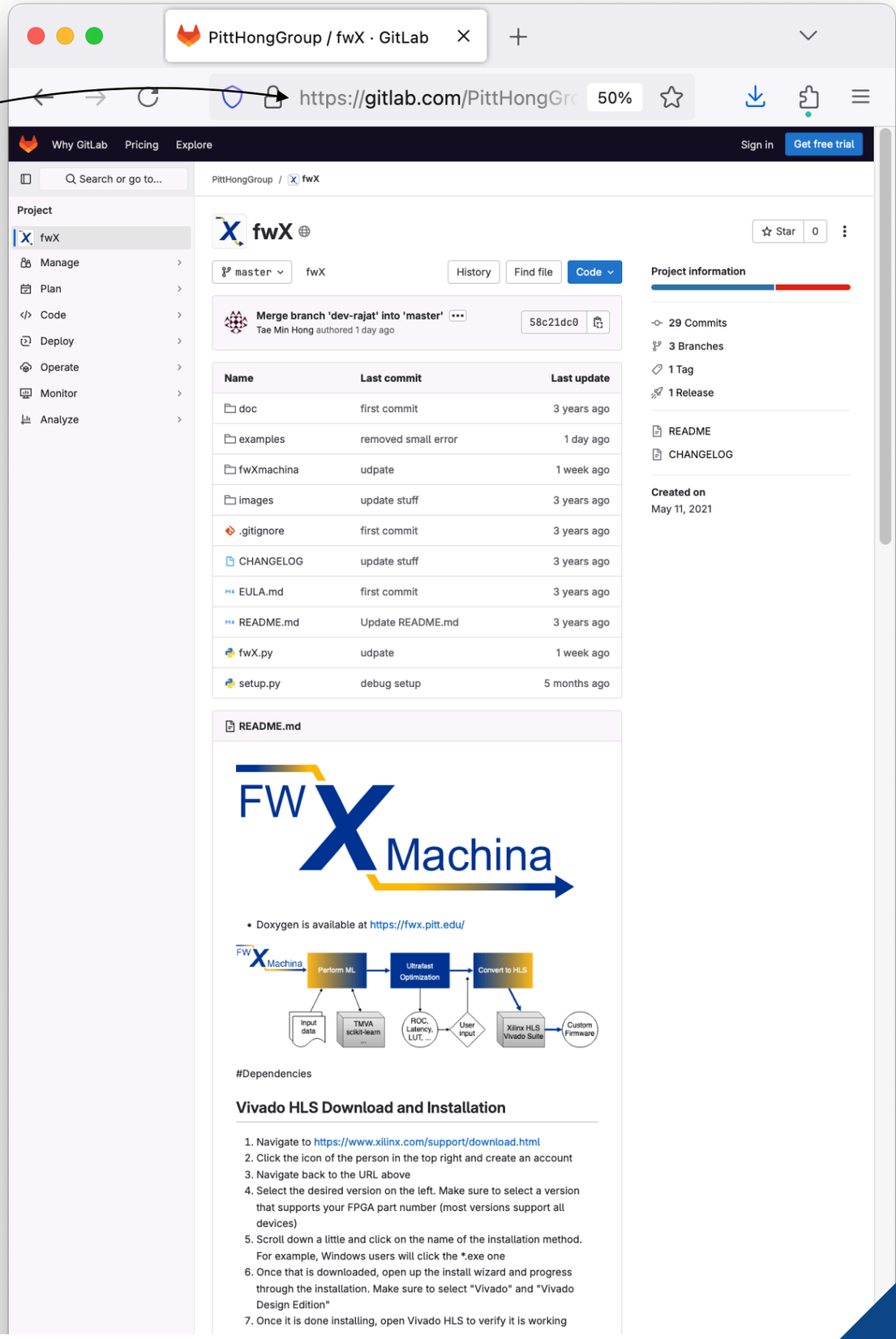
- Shared by email request

  parallel paths (paper 2)

  autoencoder (paper 3)

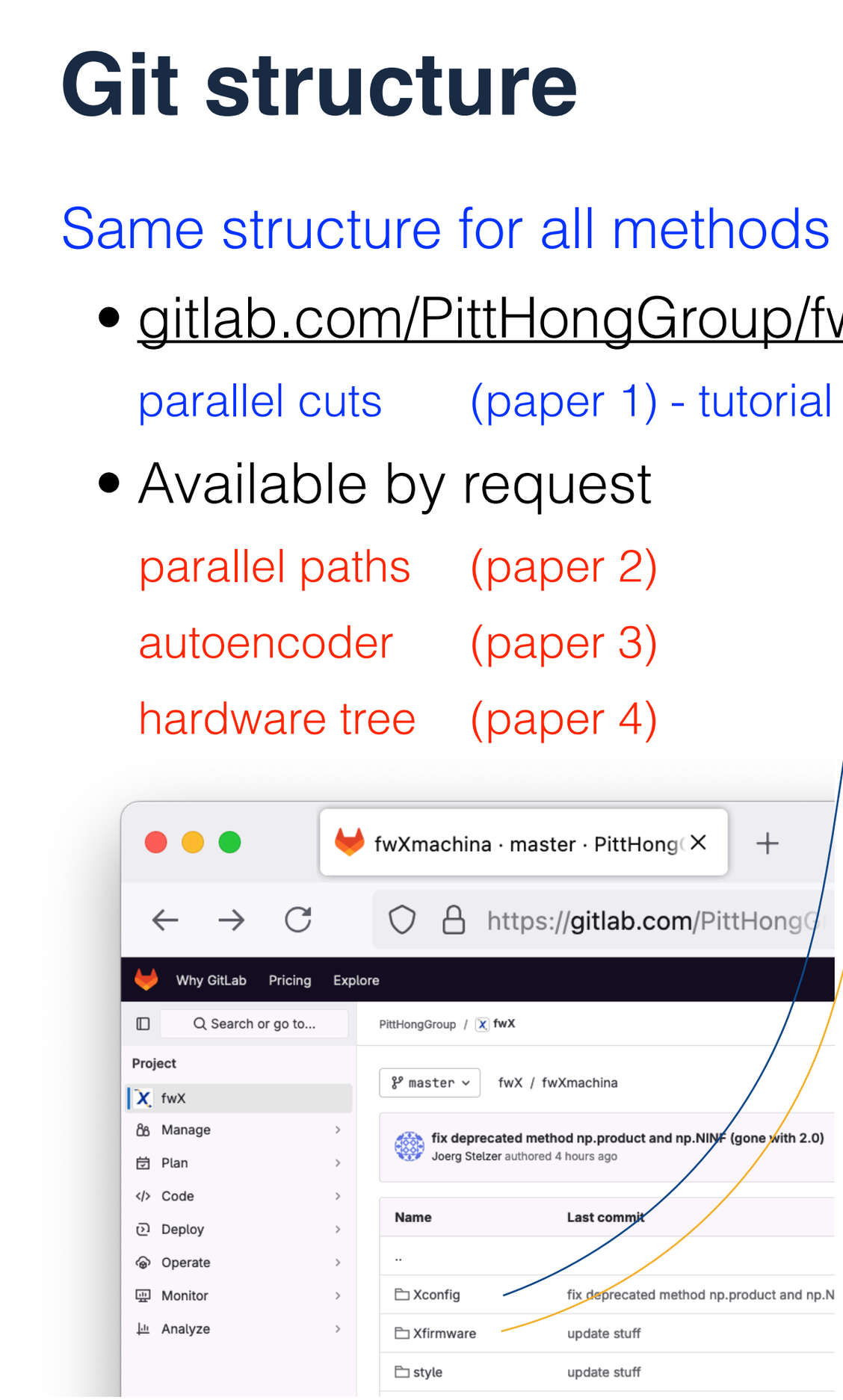
  hardware tree (paper 4)

- **Collaborators welcome**



Git structure

Hong
Pittsburgh



Same structure for all methods

- gitlab.com/PittHongGroup/fwX

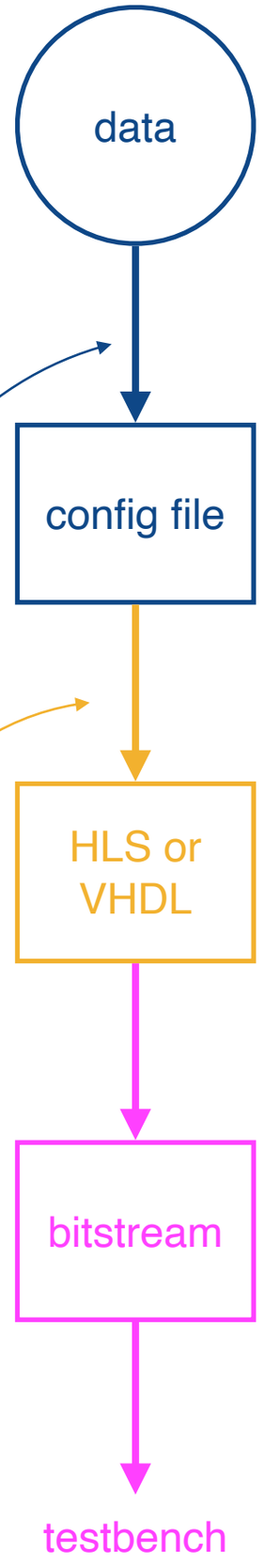
  parallel cuts (paper 1) - tutorial today

- Available by request

  parallel paths (paper 2)

  autoencoder (paper 3)

  hardware tree (paper 4)



- Xconfig

  creates model configuration

  tutorial - part 1

- Xfirmware

  writes HLS or VHDL

  tutorial - part 2

- Vivado

  synthesize & testbench

  tutorial - part 3

FW testbench w/ IP available

Hong Pittsburgh

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

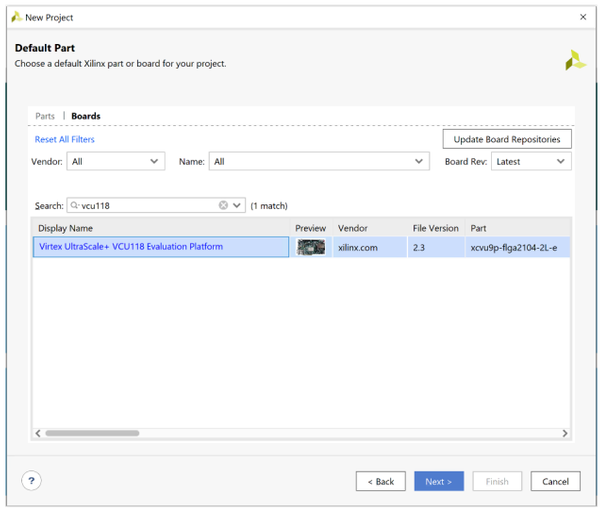
Before Beginning

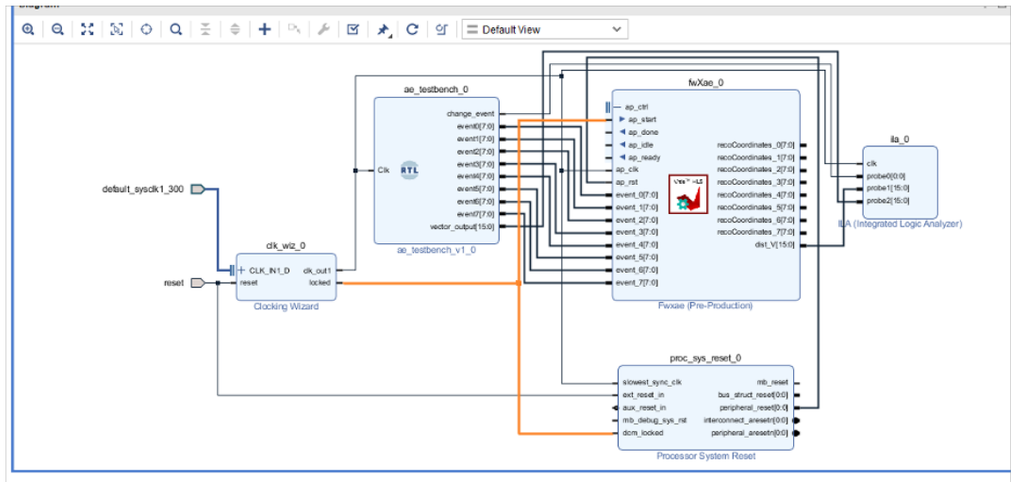
Before beginning, please make sure that you have (and know the location of) the autoencoder IP folder, and the VHDL testbench files:

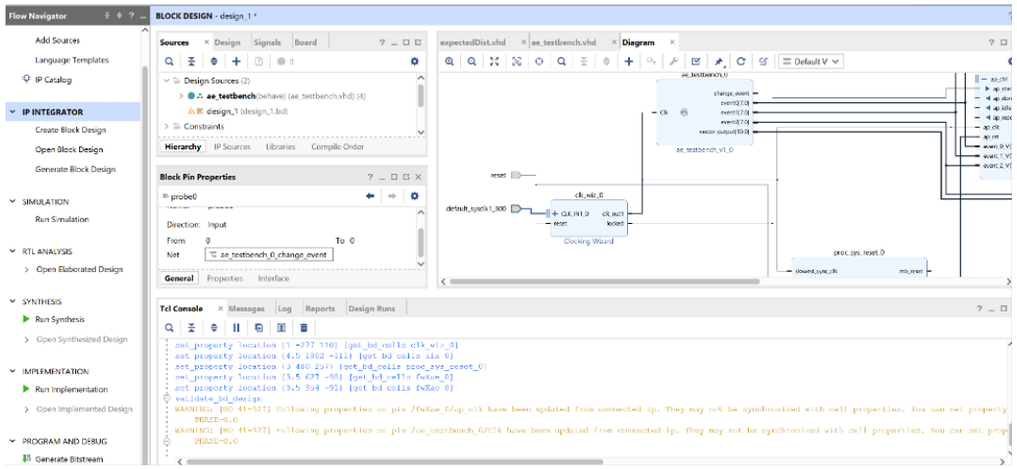
Name	Date modified	Type	Size
autoencoder8var_ip	2/7/2024 1:30 PM	File folder	
tb_vhd_files	2/8/2024 11:50 AM	File folder	

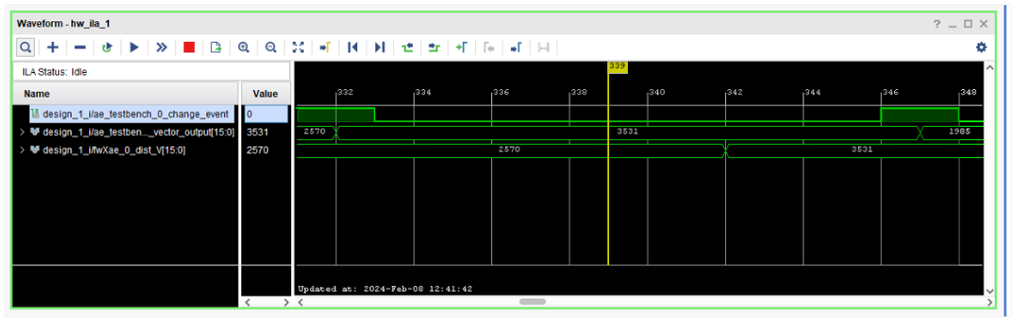
Creating New Project in Vivado

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.









More info

Start page

- 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

FW X 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.

Hong Pittsburgh

#			
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

Tutorial

- SMARTHEP Edge ML School 9/24/24

  Slides

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

  Videos on synthesizing & test bench

  indico.cern.ch/event/1405026/contributions/6103386/