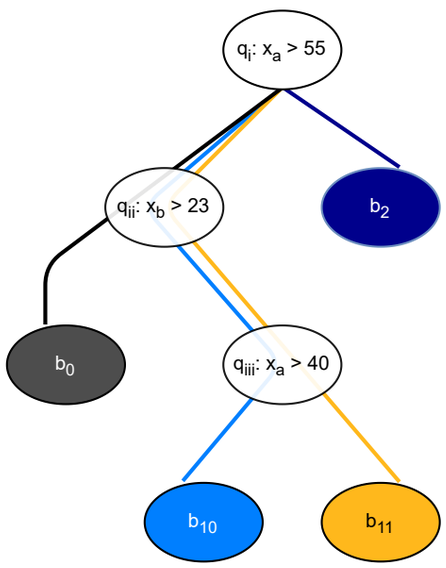
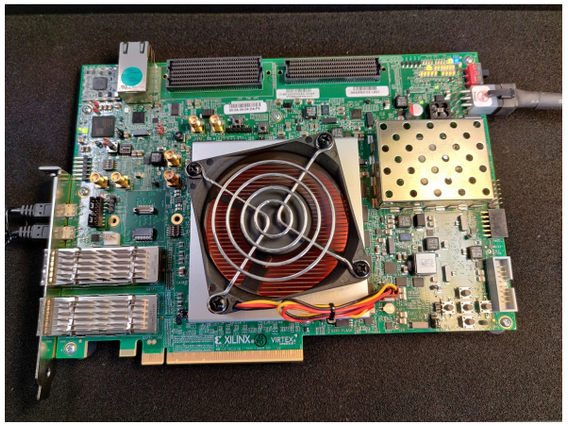
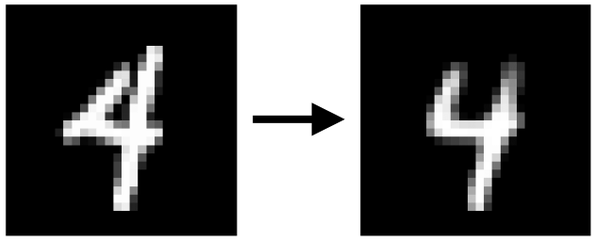
Tree-distilled

Autoencoder on FPGA to anomaly detect compress data

^







Tae Min Hong

University of Pittsburgh

AI4EIC, MIT

October 27, 2025

https://indico.bnl.gov/event/28082/contributions/115514/

Outline

Introduction

- **FPGA** Field programmable gate arrays
- **AE** Autoencoder
- **DT-AE** Decision Tree AE

Anomaly Detection

- Train DT-AE → Deploy Trees
- Regression

Trees on FPGA

- Decision Tree structure
- HLS vs. VHDL ← 1st time showing
- Train VAE → Deploy Trees

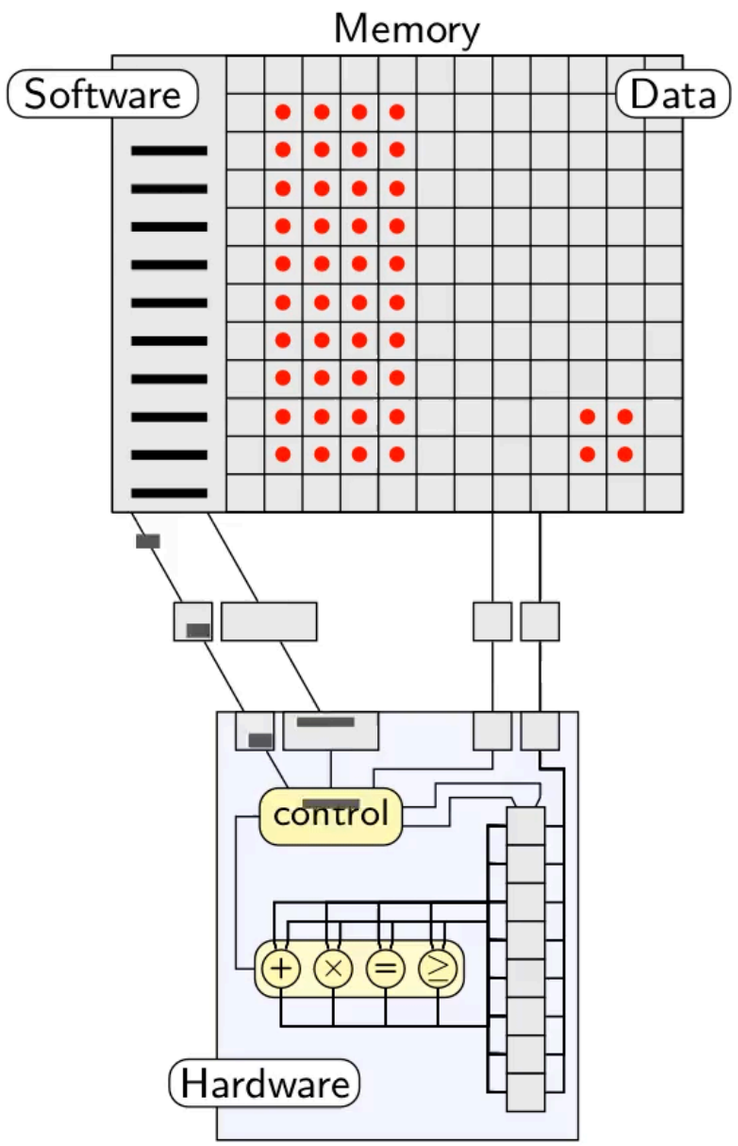
Data Compression

- Data compression ← 1st time showing
- Code & git & slides & videos

Why FPGA (vs. CPU)

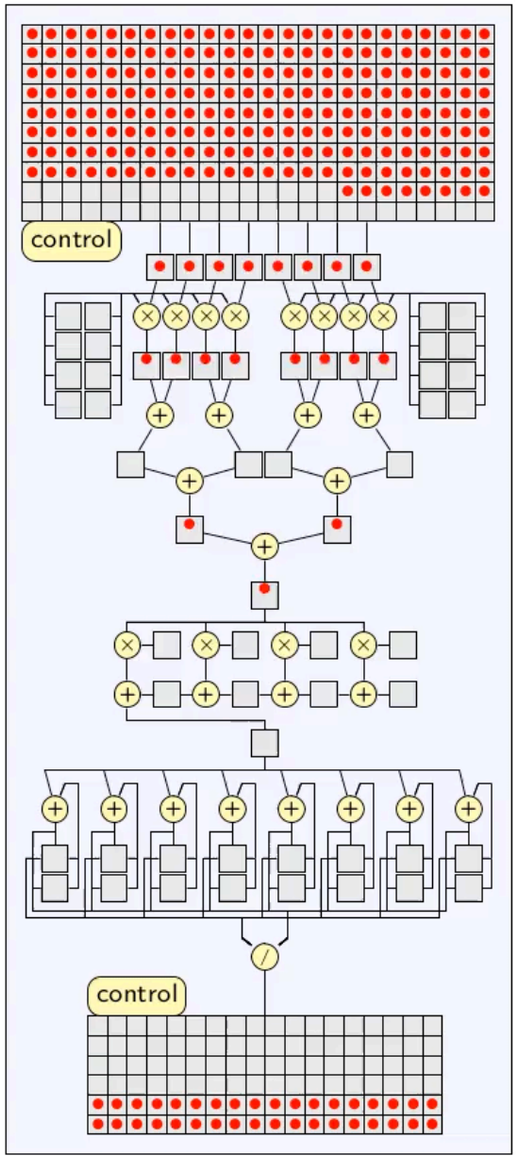
Hong
Pittsburgh

Animation from https://qbaylogic.com/wp-content/uploads/2024/05/Animation_CPU-vs-FPGA.mp4



Normal processor

Software controls computation process



FPGA

Computation integrated in hardware

One operation per clock tick

typical timing $10^{0\text{-}3}$ ms

Many operations per clock tick

typical latency ≪ µs, thruput ≫ 1 MHz

Autoencoder recap

Hong Pittsburgh

Example: handwritten digits

28 x 28 pixel images

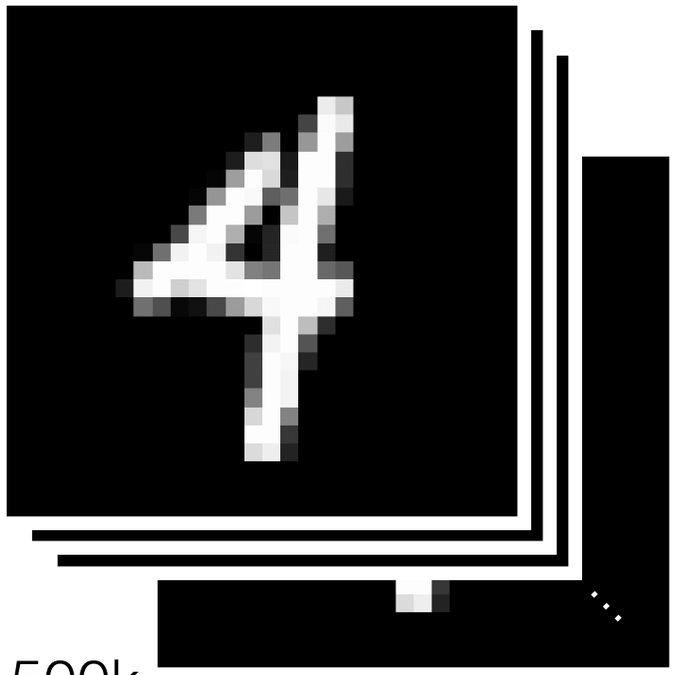


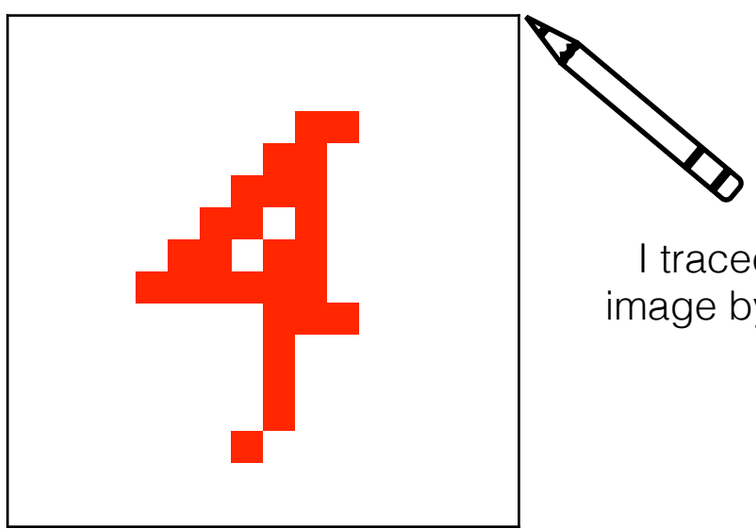
Image 1
2
3
500k

=

Corresponding MNIST data sample

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

16 x 16 pixel (1-bit)



I traced the image by hand

# pixels	28^2	=	784
# bits per pixel	8-bits	×	8
# bits per image			6272
			6272
# bits per image	$16^2 \bullet 1$ =	÷	256
Compression factor			24.5

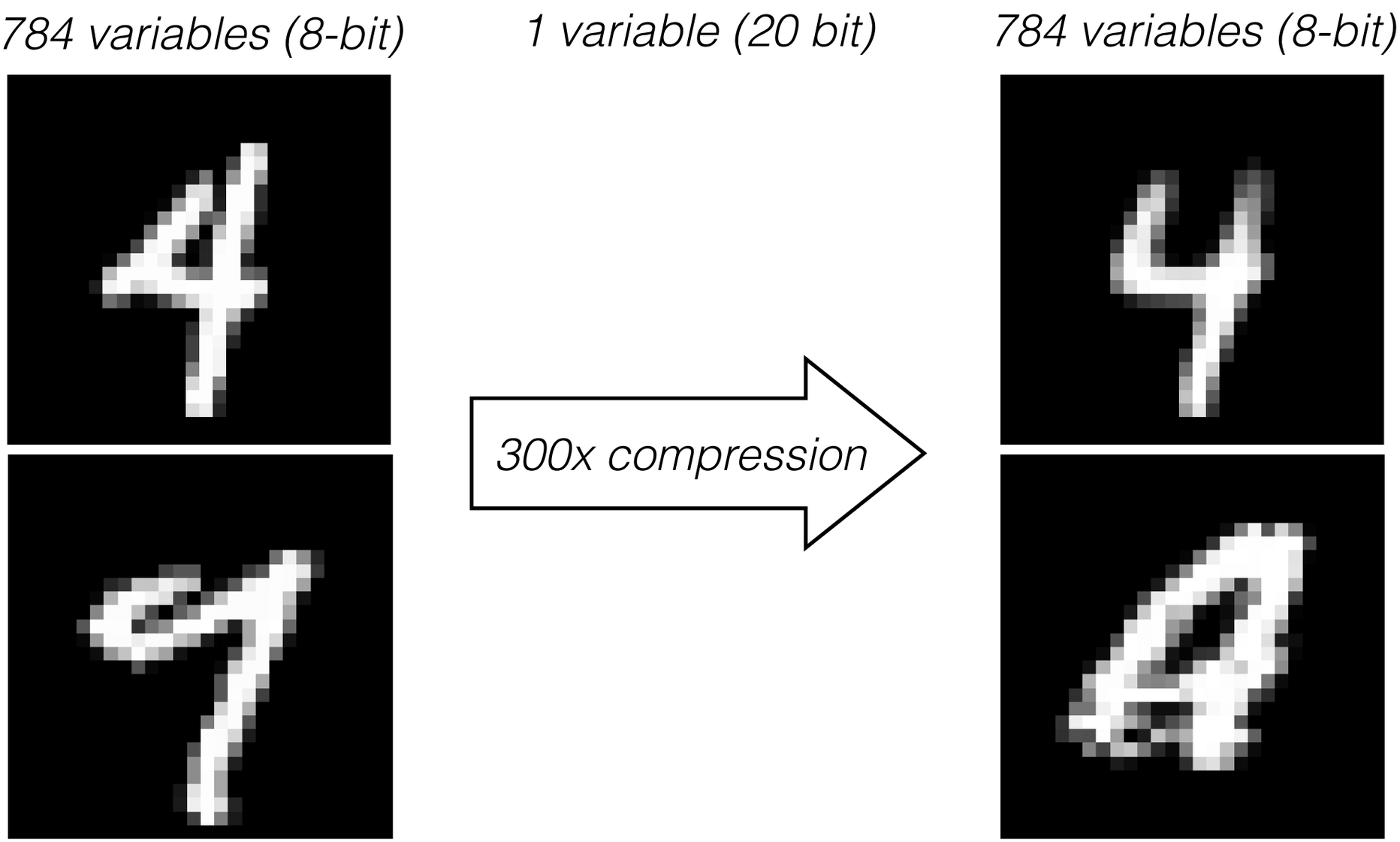
Compression isn't hard, but autoencoder can do better

Autoencoder recap

Hong Pittsburgh

Example: handwritten digits (8-bit pixels on 28x28 grid)

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



Meaning

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

Strategy

How should we call the autoencoder?

Hong Pittsburgh

Training
done offline on CPU

Deployment
for online on FPGA

Neural network-based training

Decision tree-based training

Anomaly detection
- Govorkova et al., Nat. Mach. Intell. 4 (2022) 154
- CMS Collaboration, Comput. Softw. Big Sci. 8 (2024) 11

Data compression
- Guglielmo et al., IEEE Trans. Nucl. Sci. 68 (2021) 2179

Not a comprehensive list

Anomaly detection
- Gupta (for ATLAS) Pheno Symposium 2025 https://indico.global/event/812/contributions/126571

Data compression
- **This talk**

Anomaly detection
- Roche, TMH et al., Nat. Comm. **15** (2024) 3527 http://doi.org/10.1038/s41467-024-47704-8
- Ercikti & TMH using VHDL **This talk**

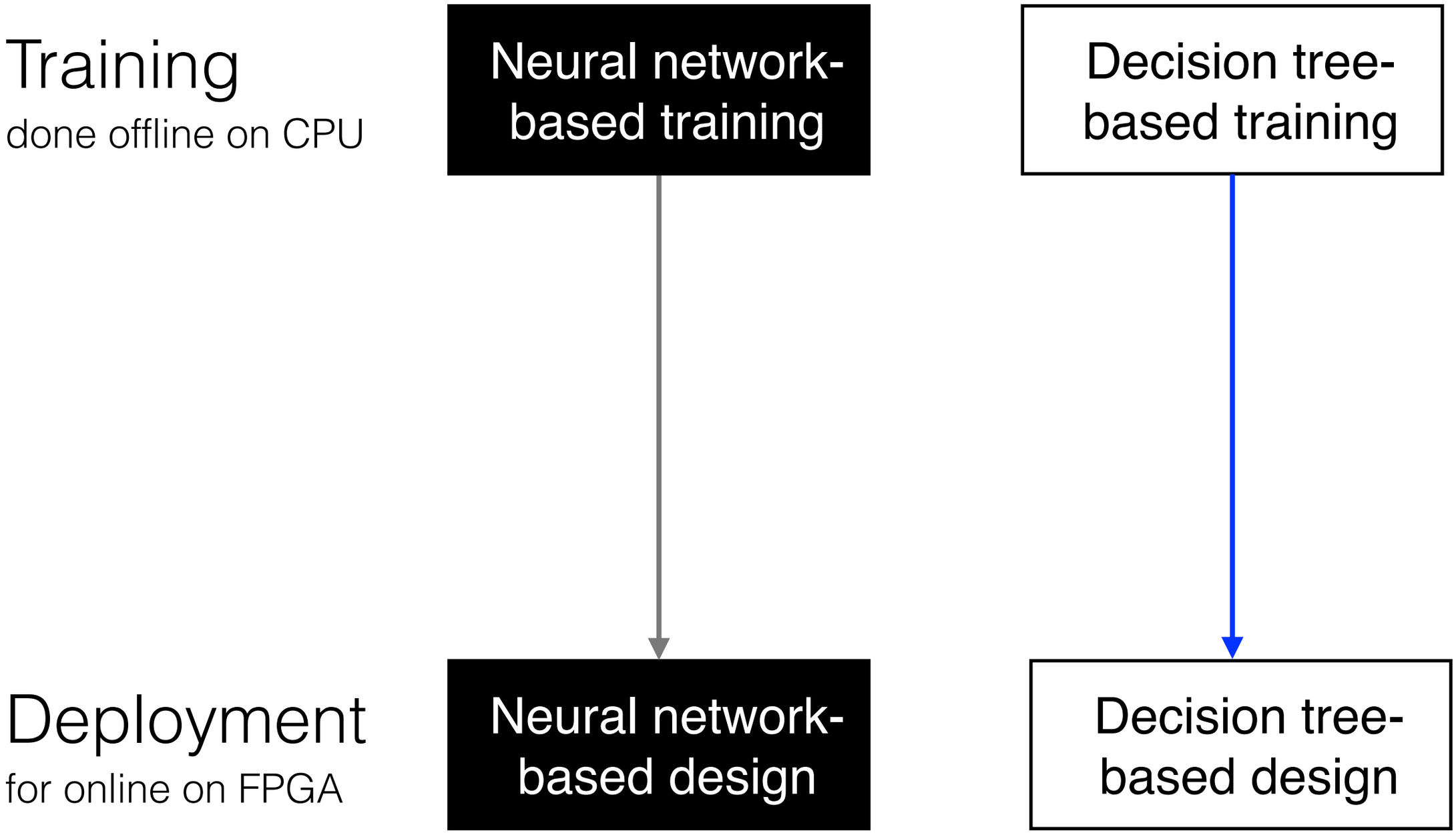
Neural network-based design

Decision tree-based design

Not sure if useful but certainly possible

I'll start with these

Hong
Pittsburgh

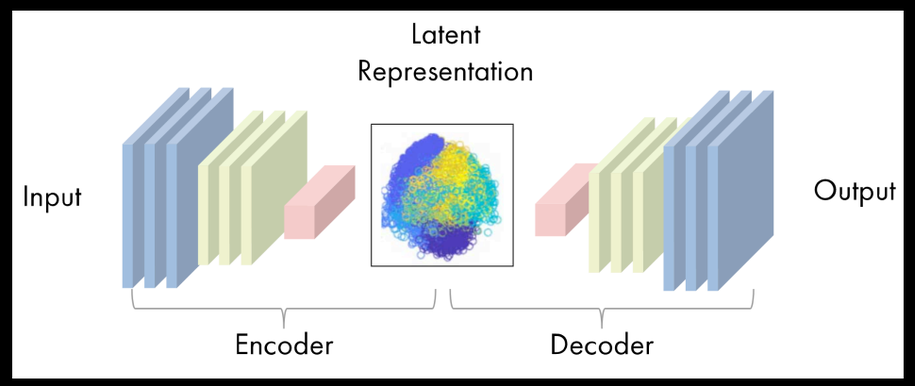


Autoencoders

Hong Pittsburgh

NN AE

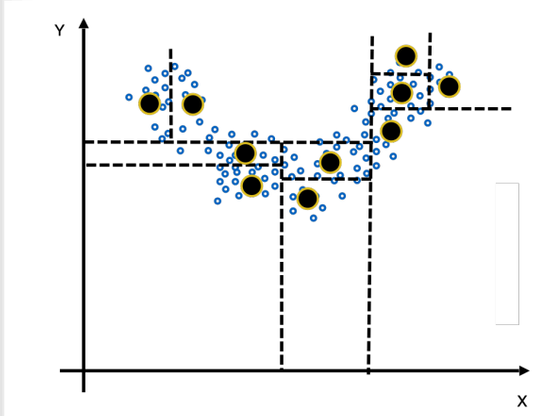
- Training is a black box, done offline
- Latent space is complex



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

"Starcoder" DT-AE

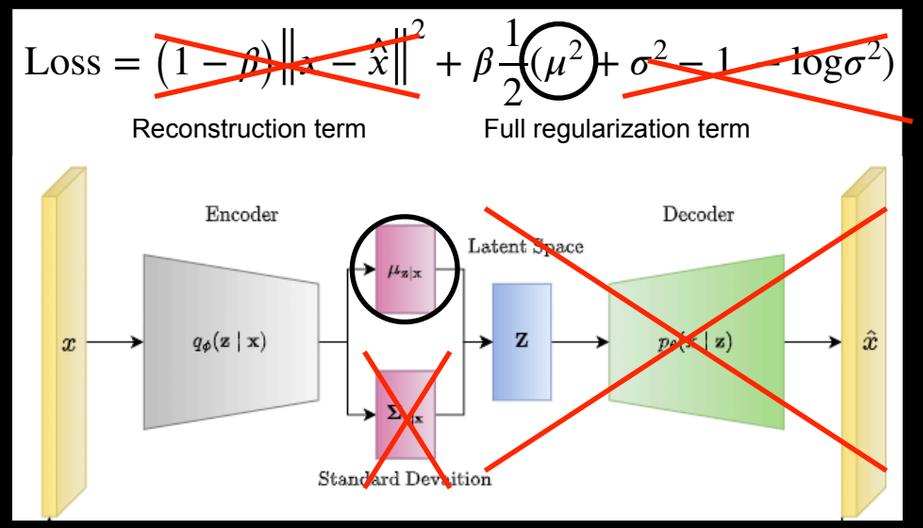
- Training is sampling of 1d pdfs
- Latent space is simple / interpretable



Latent data is the bin number

- FPGA version simplified for anomaly at CMS

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

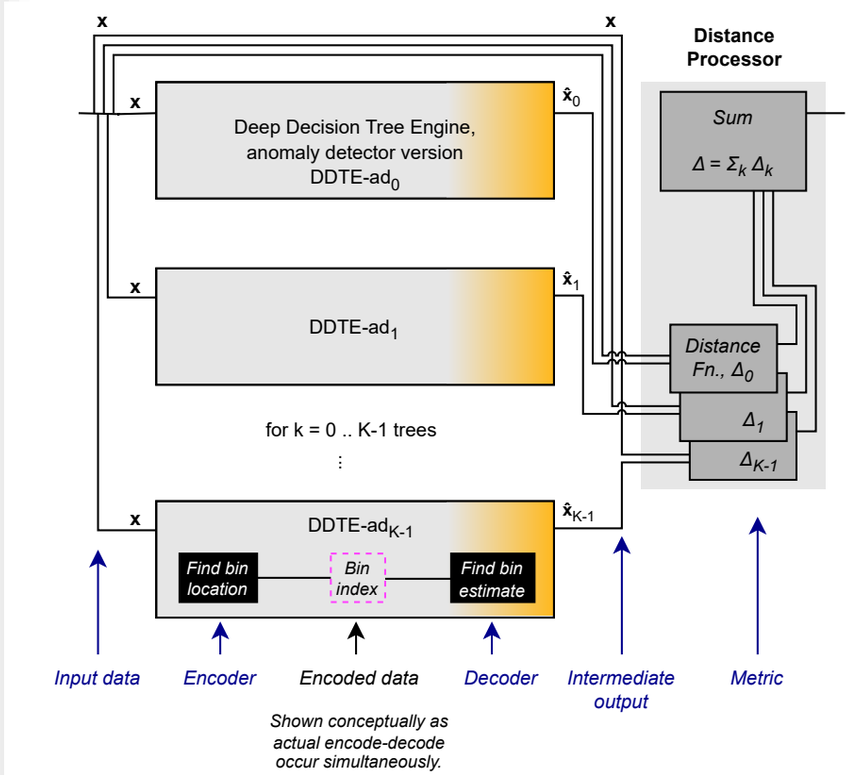


Cross-outs are mine

Cross-outs are mine

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

- FPGA version can optionally skip latent sp.



DT-AE training developed in-house

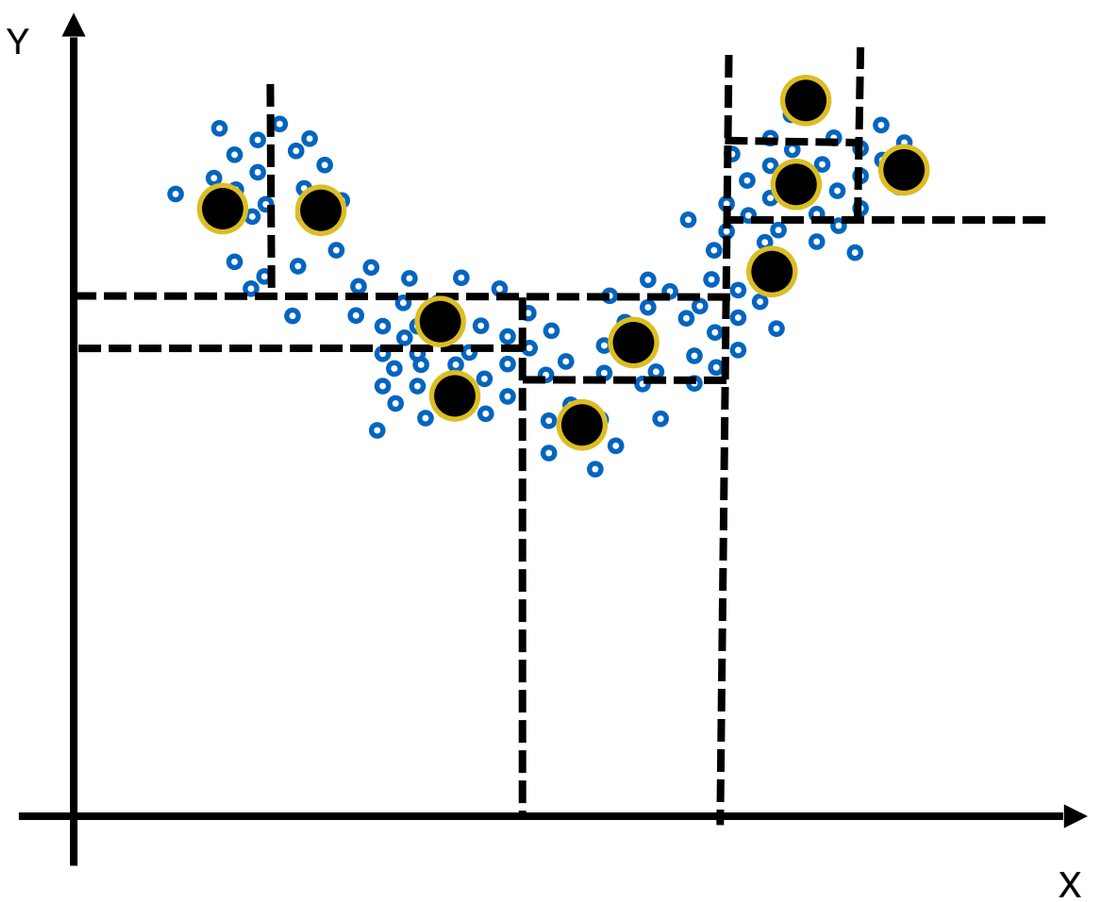
Hong Pittsburgh

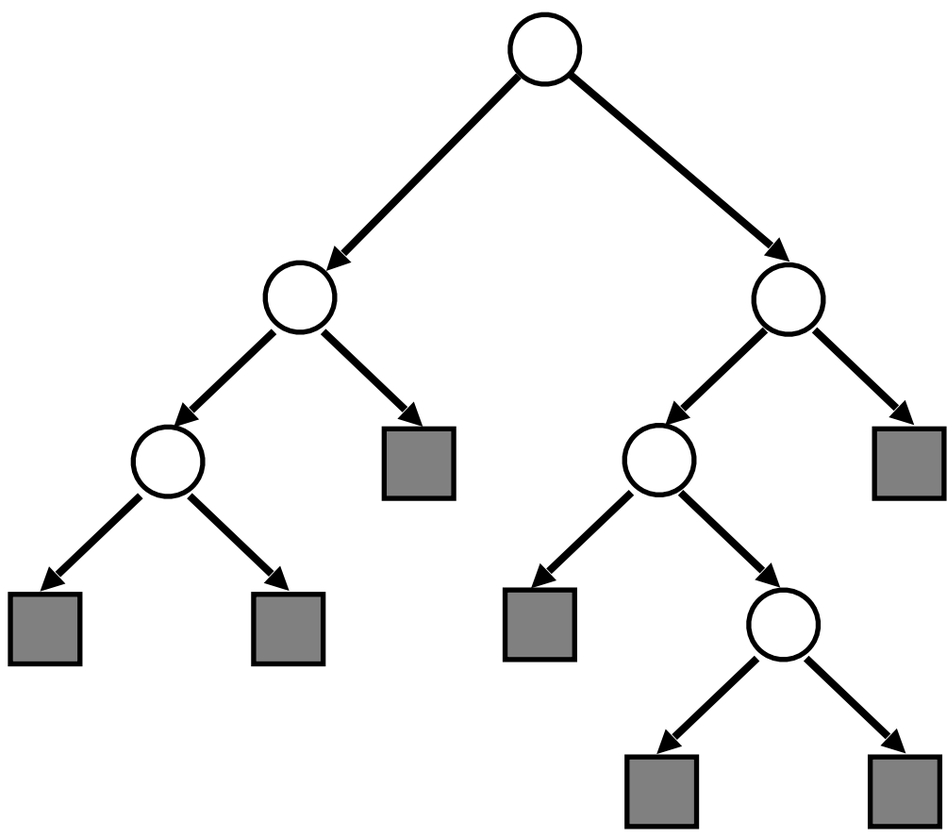
Training by sampling 1d projection of input variables

- It's doing density estimation
- Encoding: Input variables → which bin it's in

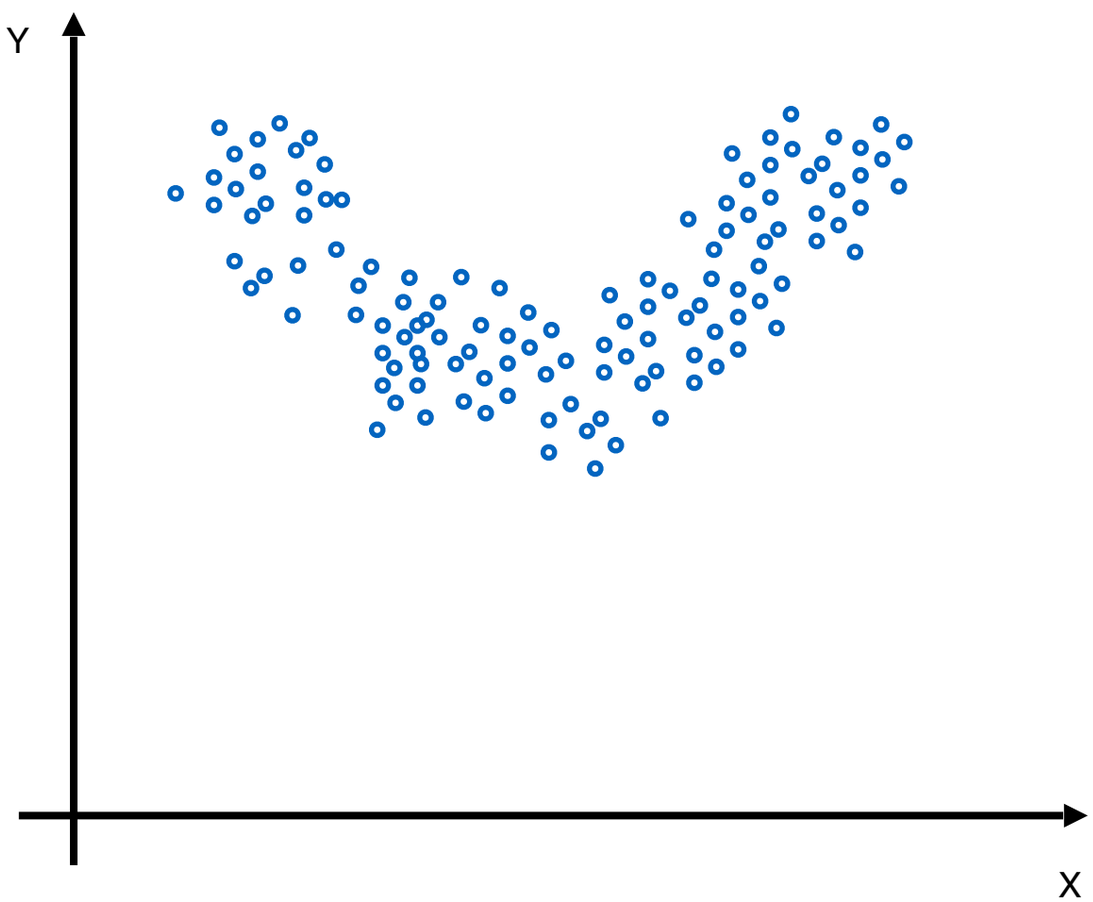
Decoding returns "reconstruction point"

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

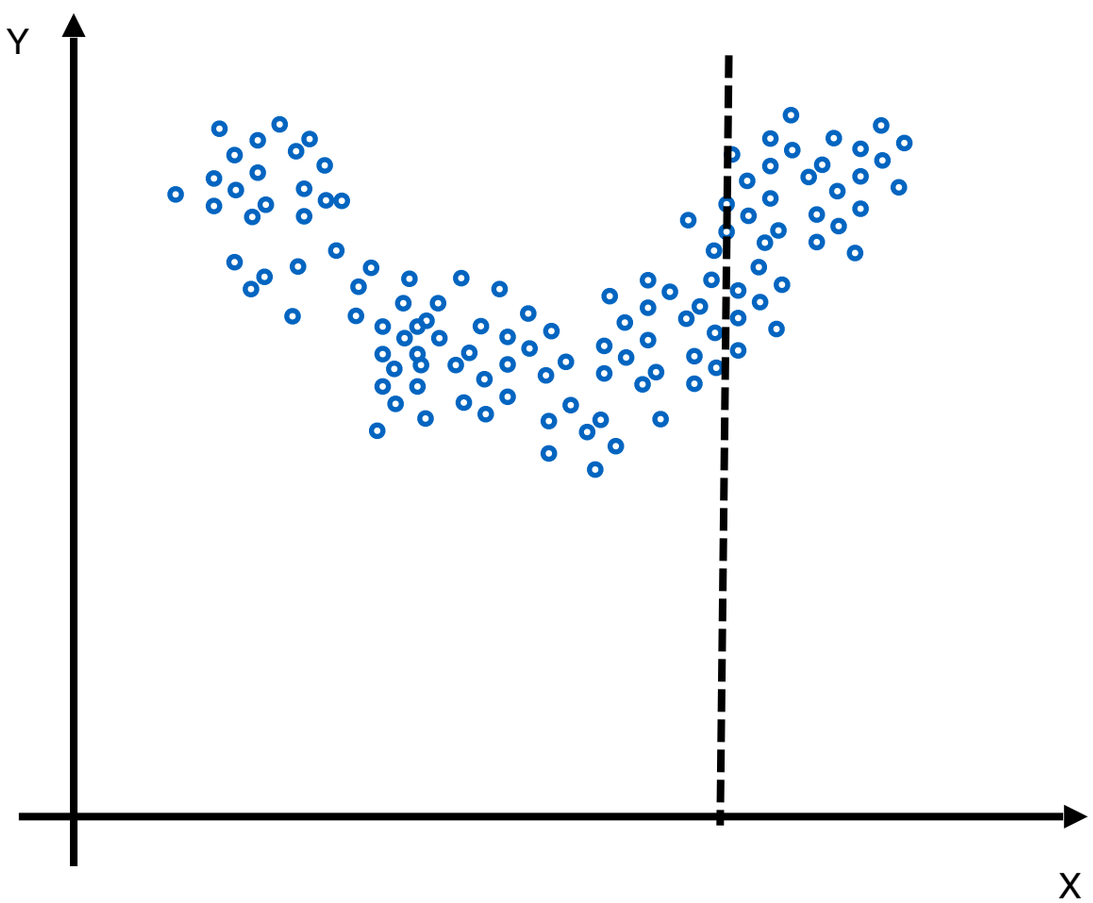


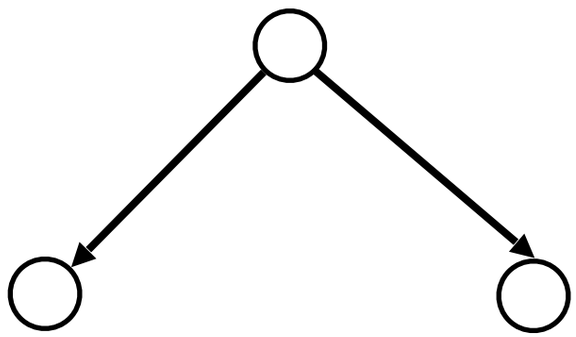


Start over

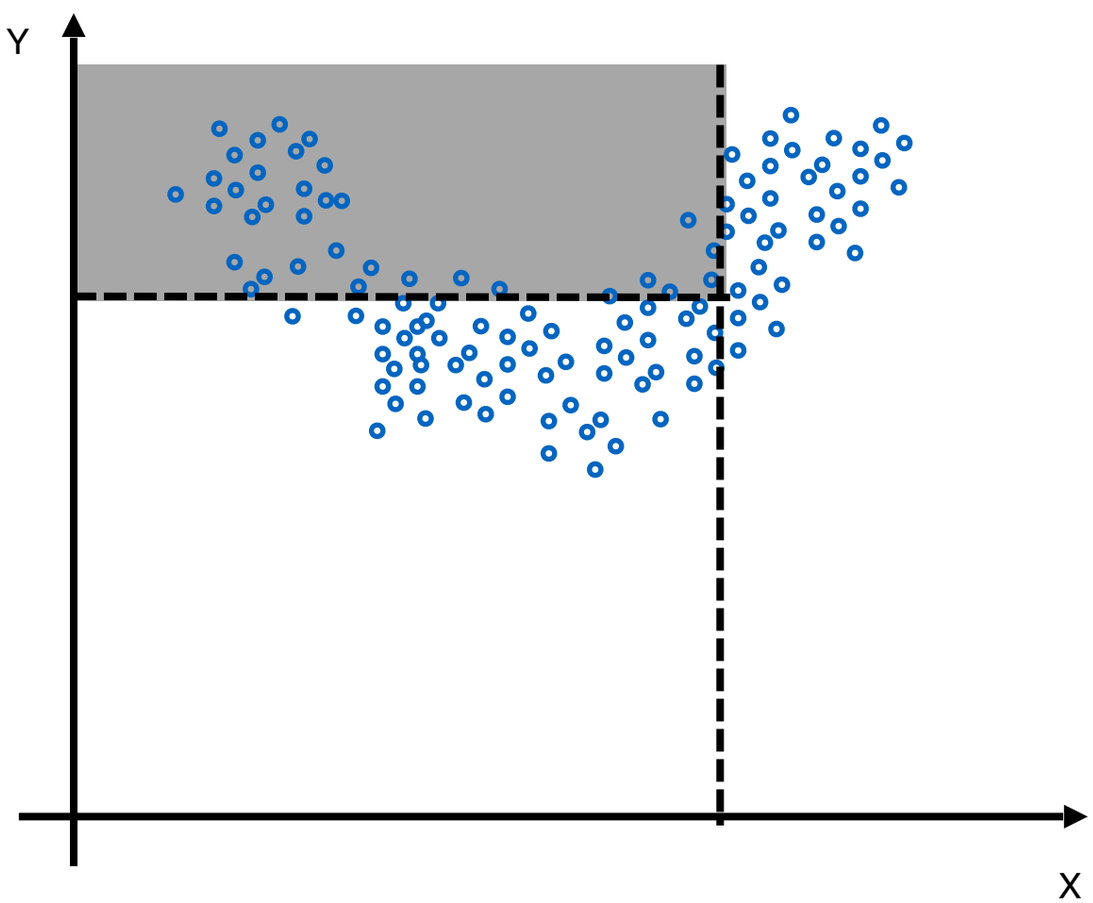


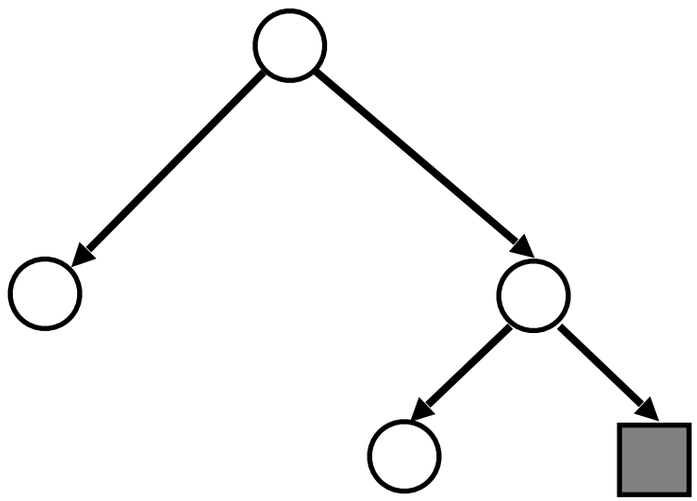
Choose 1 variable, sample a cut



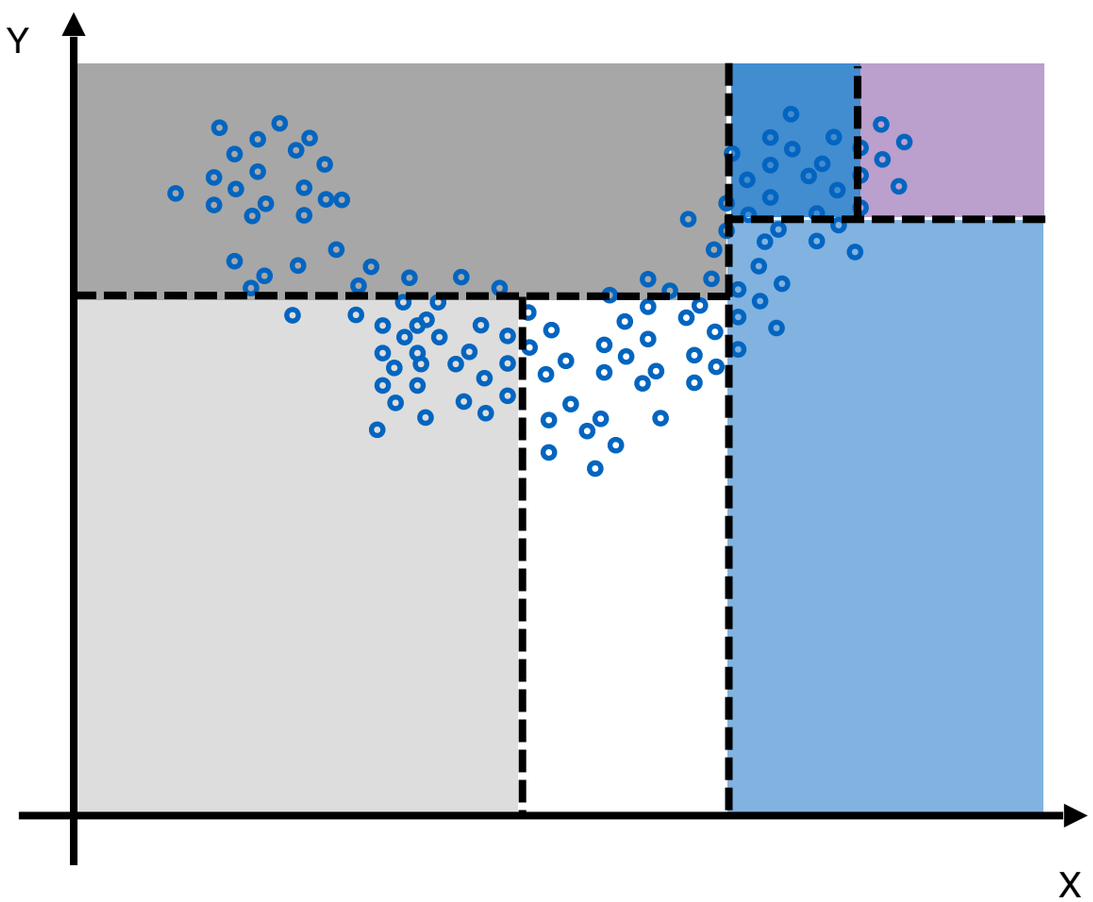


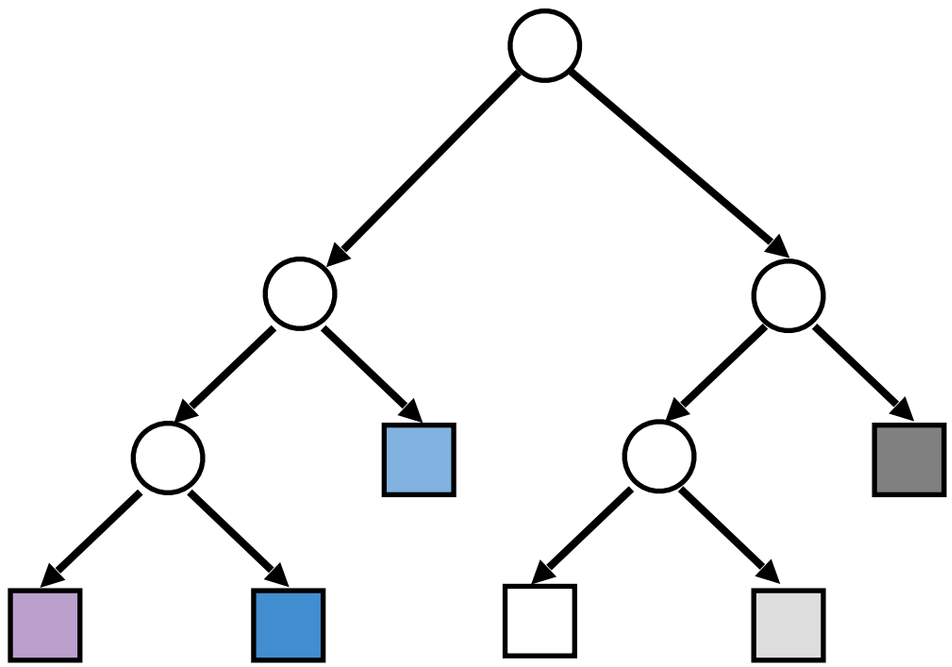
Repeat



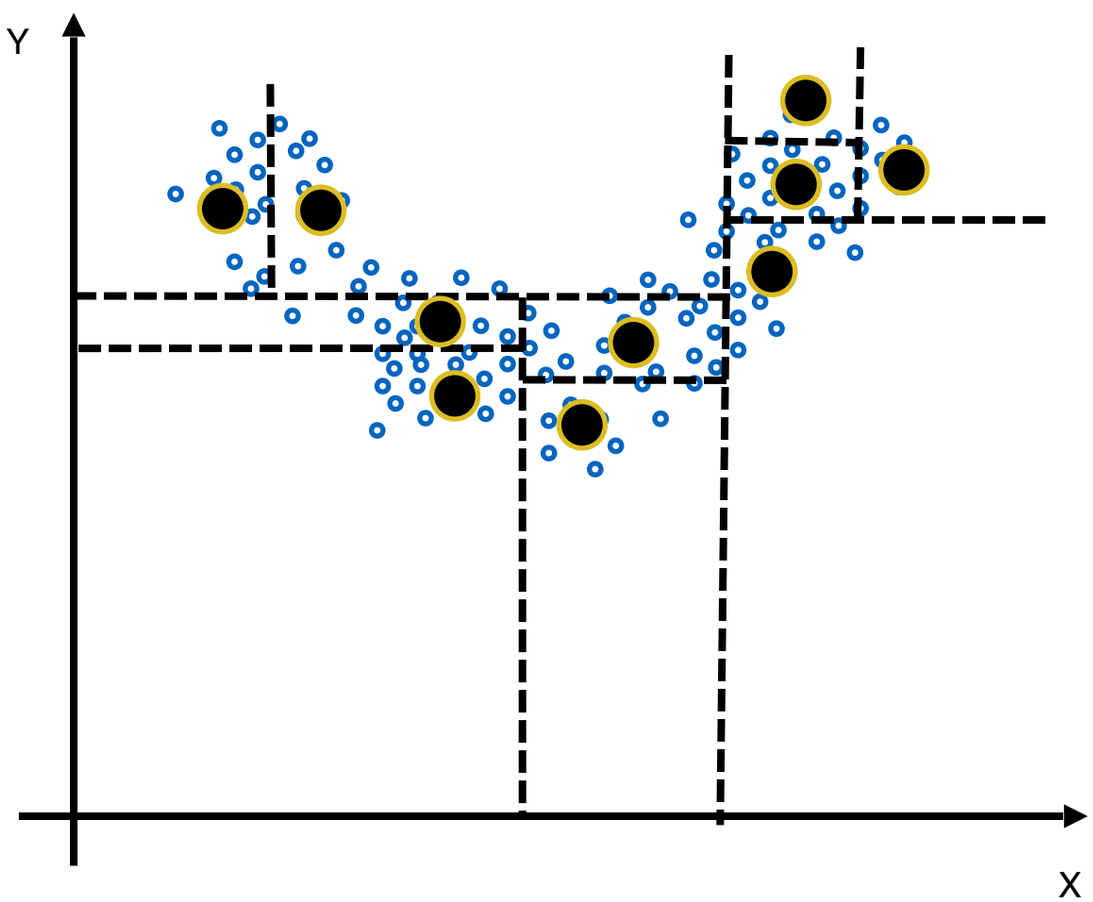


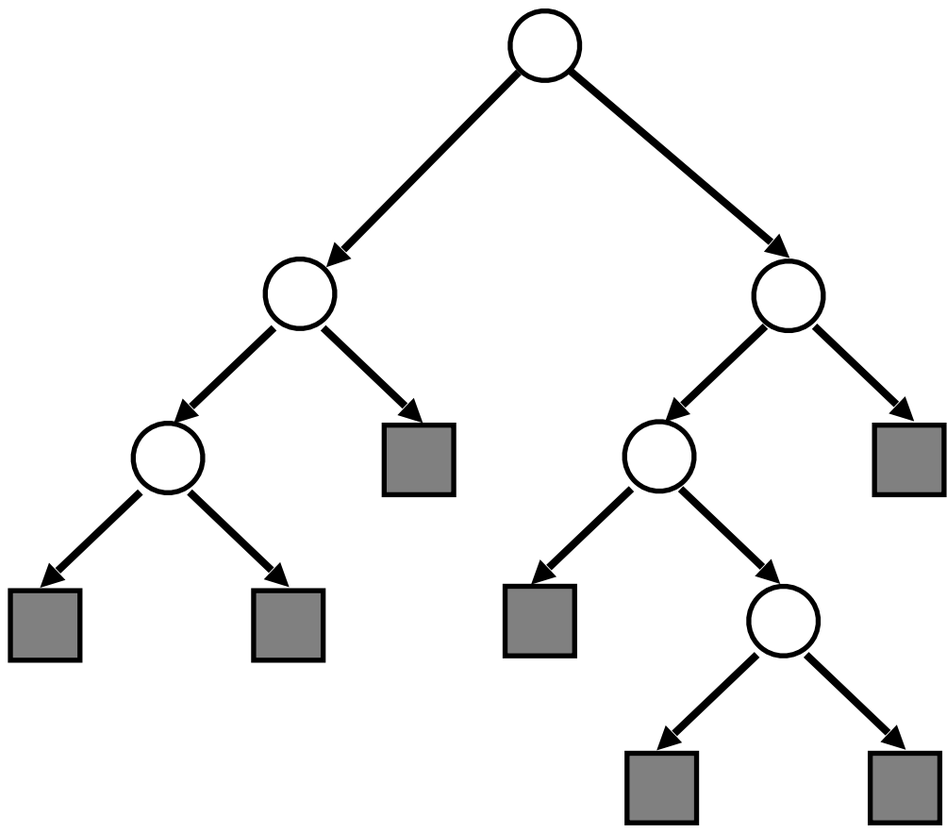
Bins





Choose median value





AE becomes anomaly detector

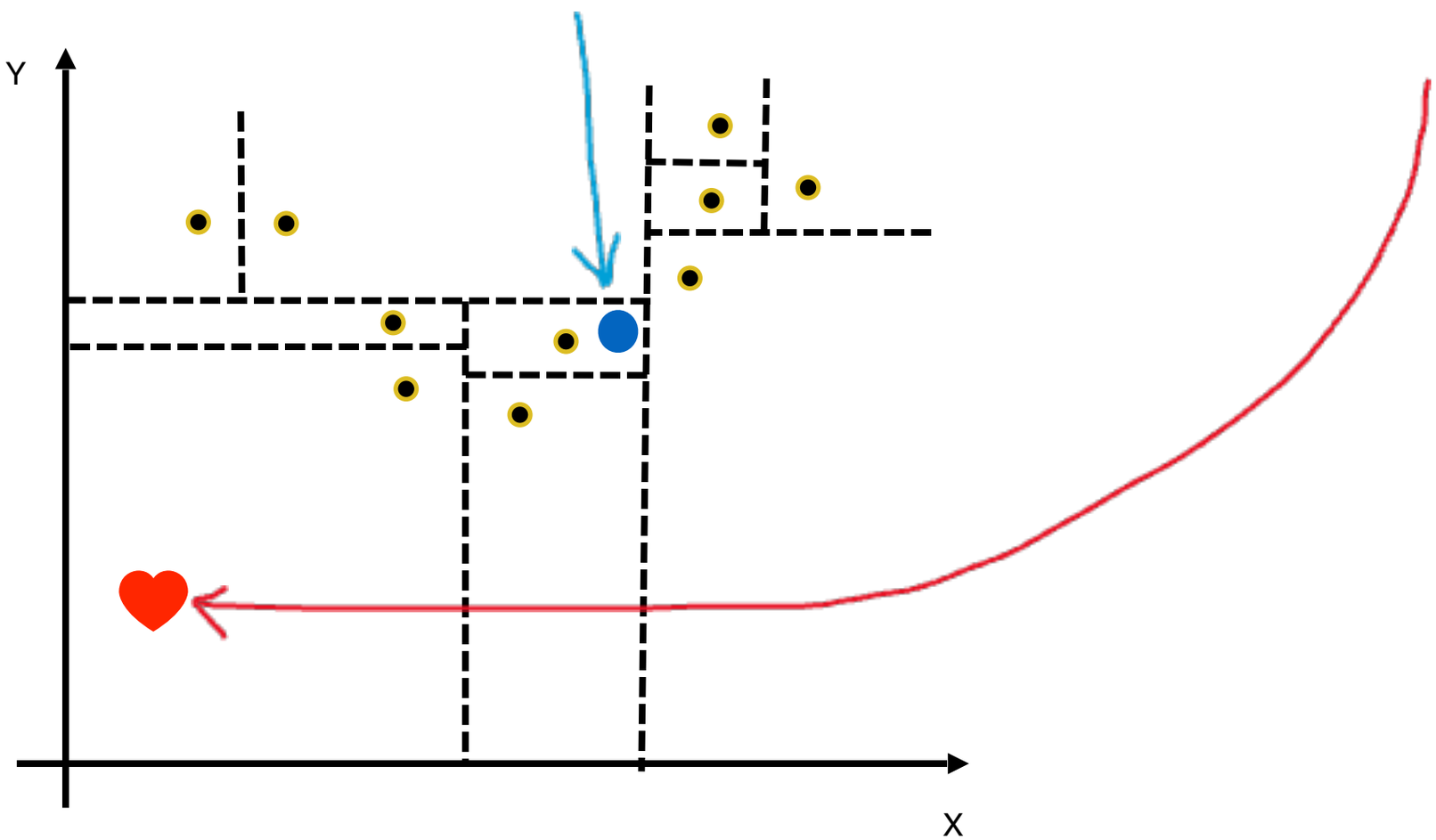
Hong Pittsburgh

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

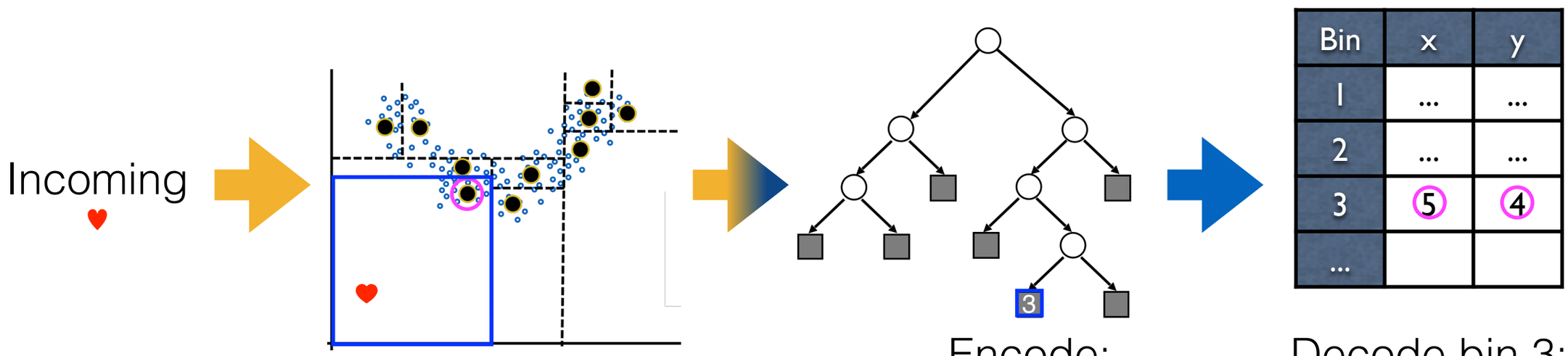


Realized we can skip latent space

Hong Pittsburgh

Decode?

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



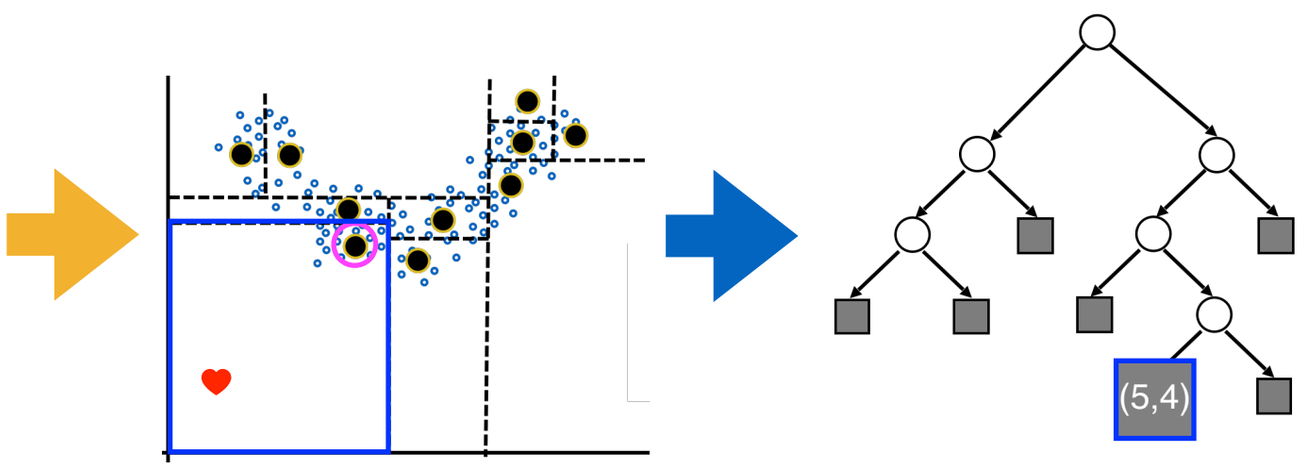
Bin	x	y
1	...	...
2	...	...
3	5	4
...		

Encode: return bin 3

Decode bin 3: return (5,4)

No need to encode

- Starcode: input var → coord.



Encode is Decode: return (5,4)

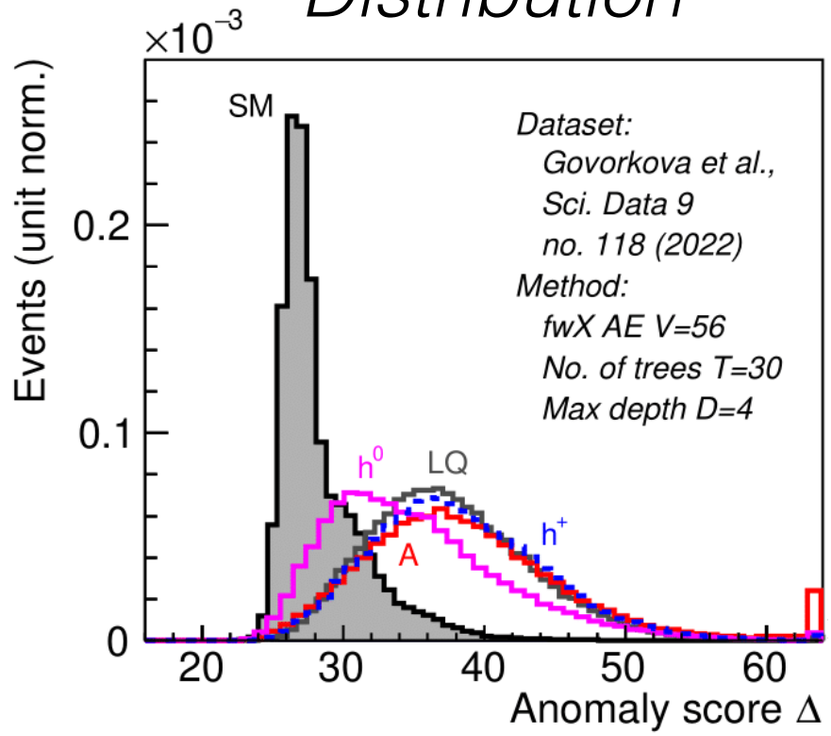
Starcoder vs. hls4ml

Hong Pittsburgh

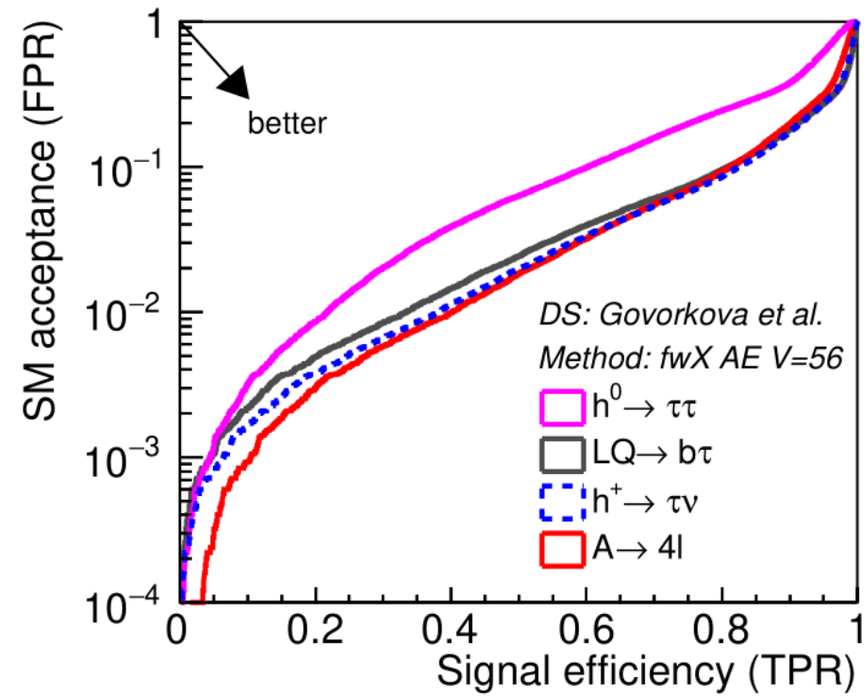
Works well

- Physics (plots)
- FPGA (table)

Distribution



ROC curve



Comparison

- hls4ml NN-AE
  Nature Mach. Intell. 4 (2022) 154
- Physics: comparable AUC
- FPGA results

	hls4ml	starcoder-hls
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

Update

- Starcoder result here uses HLS trees. Recently upgraded to VHDL trees (next slide)

HLS vs. VHDL trees

Hong Pittsburgh

Starcoder got an update

- HLS High-Level Synthesis ~~**C code**~~ → RTL / VHDL / Verilog
- VHDL VHSIC Hardware Description Language write VHDL directly

  Very High-Speed Integrated Circuit
- Tree design based on

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

Results

- VHDL is 3-5x more efficient
- Finishing up final testbench
- Will write-up in proceedings

	starcoder-hls	starcoder-vhdl
Clock speed	200 MHz	320 MHz
Latency	30 ns	25 ns
FF	0.6%	3.2x smaller
LUT	9%	5.3x smaller
DSP	0.8%	0
BRAM	0	0

Now the diagonal - 1

Hong
Pittsburgh

Training
done offline on CPU

Neural network-based training

Decision tree-based training

Anomaly detection
- Gupta (for ATLAS)
  Pheno Symposium 2025
  https://indico.global/event/812/contributions/126571

Deployment
for online on FPGA

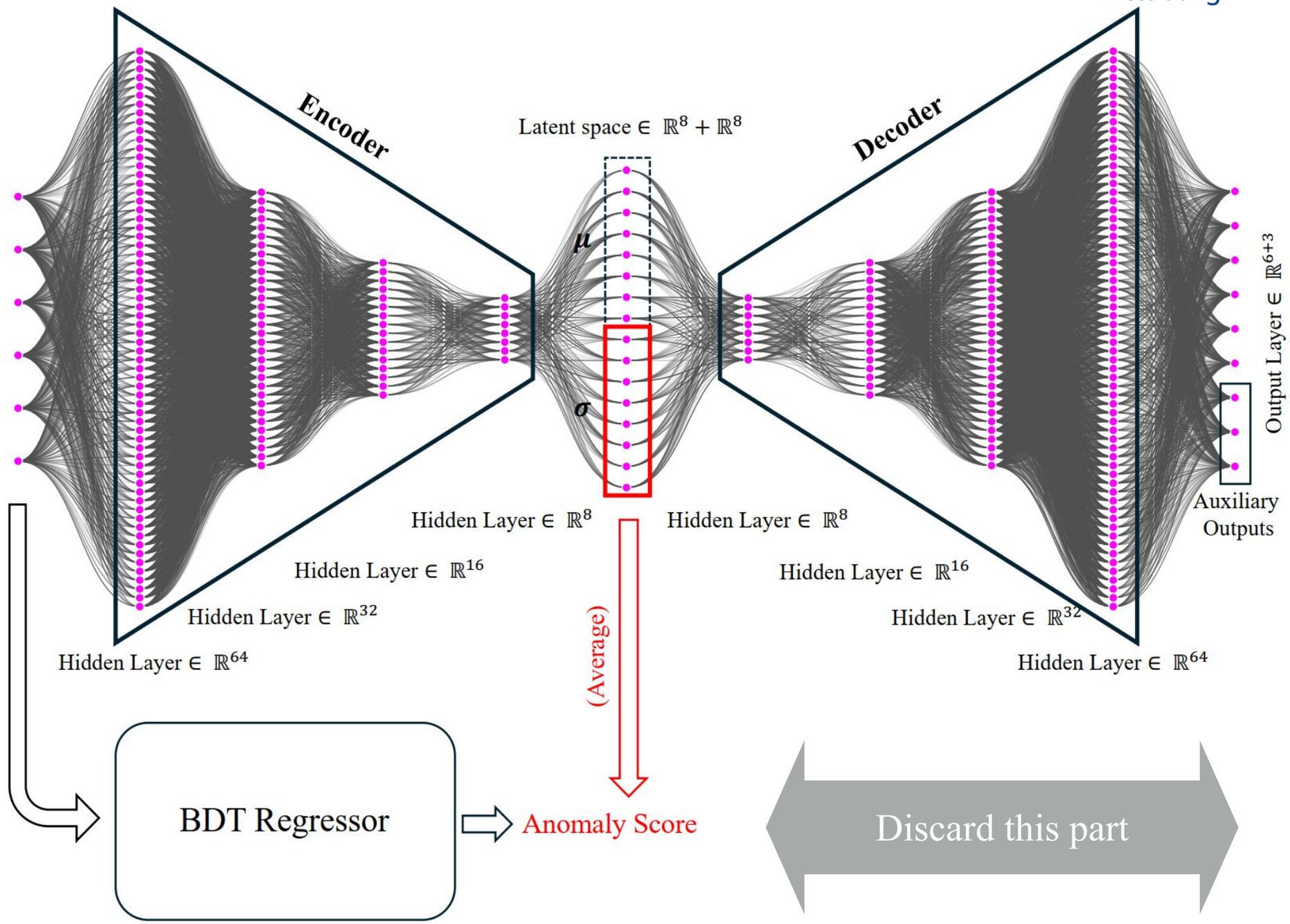
Neural network-based design

Decision tree-based design

VAE variational autoencoder

Hong
Pittsburgh

Train VAE



Regress

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

Deploy Trees on FPGA

How it looks like in an experiment

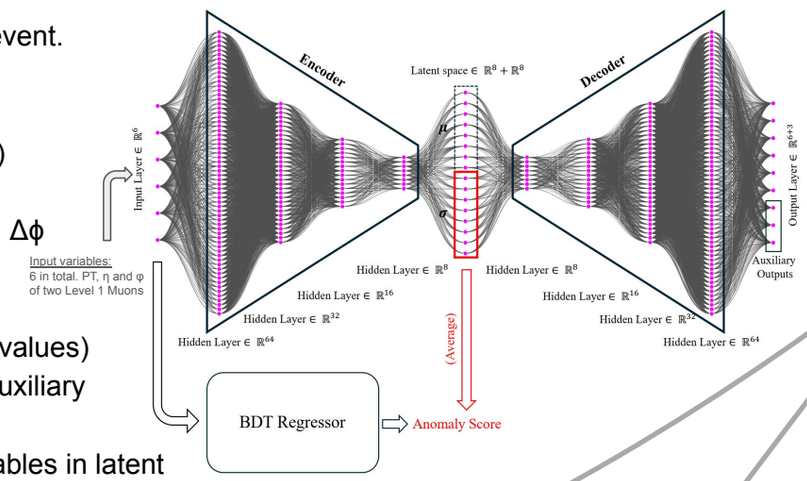
Hong Pittsburgh

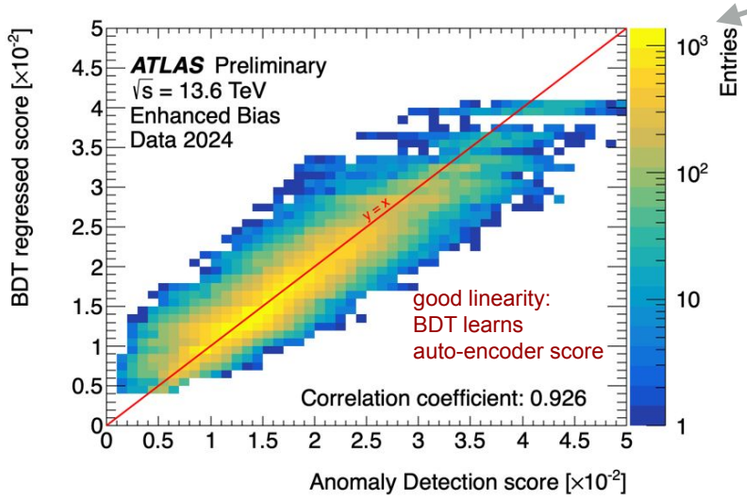
I'm not representing ATLAS today

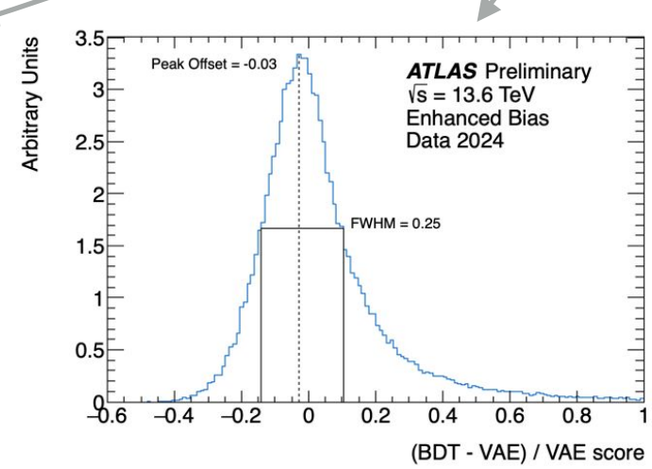
Training: Variational Autoencoder

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.





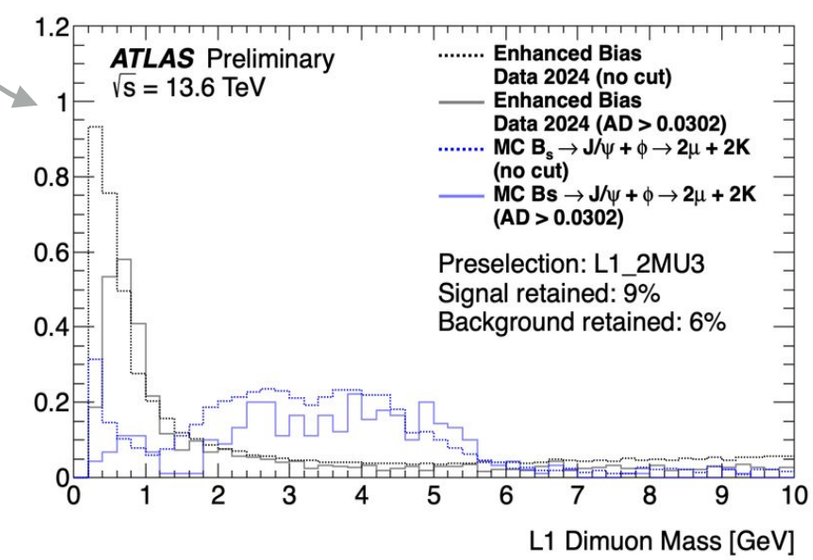


Source: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/MuonTriggerPublicResults#NoMAD_Nanosecond_Anomaly_detecti

Being commissioned at ATLAS
- Train VAE for 3 muons
- Executes in 1 clock tick (25 ns)

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

Physics result
- Unique B physics signal at L1



- Gupta
  Pheno Symposium 2025 in Pittsburgh
  https://indico.global/event/812/contributions/126571

Now the diagonal - 2

Hong
Pittsburgh

Training
done offline on CPU

Neural network-based training

Decision tree-based training

Data compression
- **This talk**

Deployment
for online on FPGA

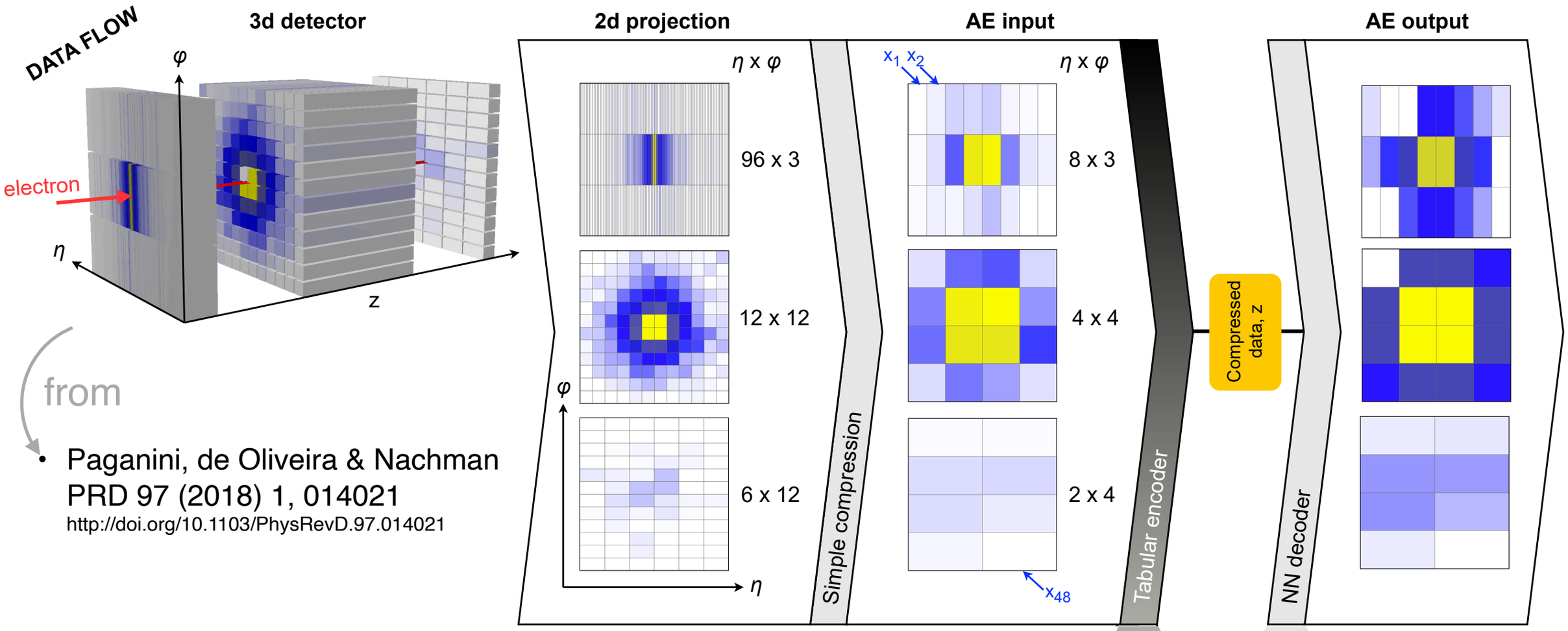
Neural network-based design

Decision tree-based design

Data compression using VAE

Hong Pittsburgh

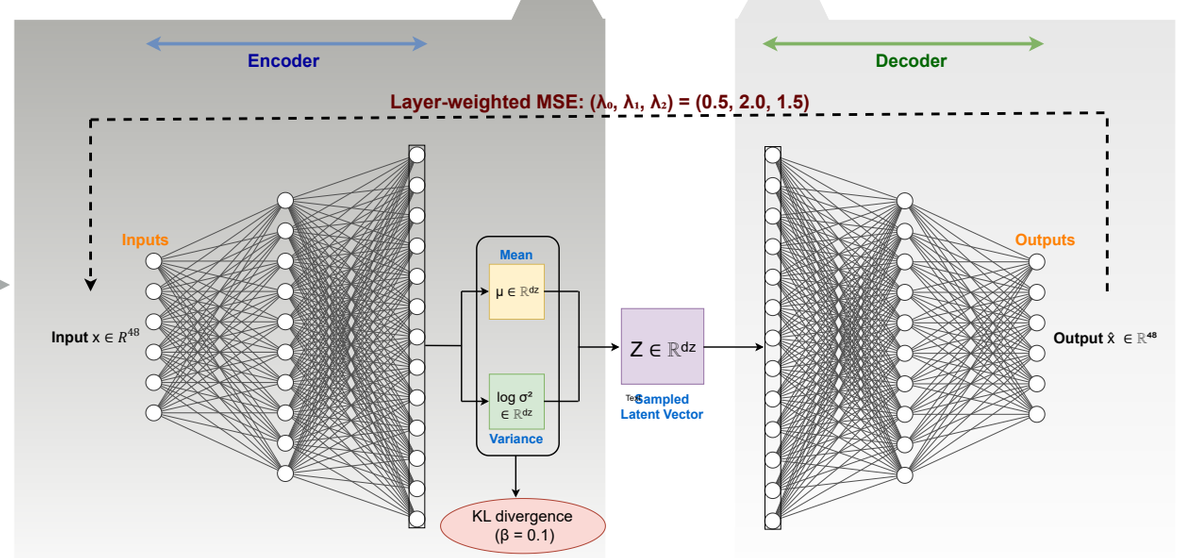
Use toy calorimeter dataset



from

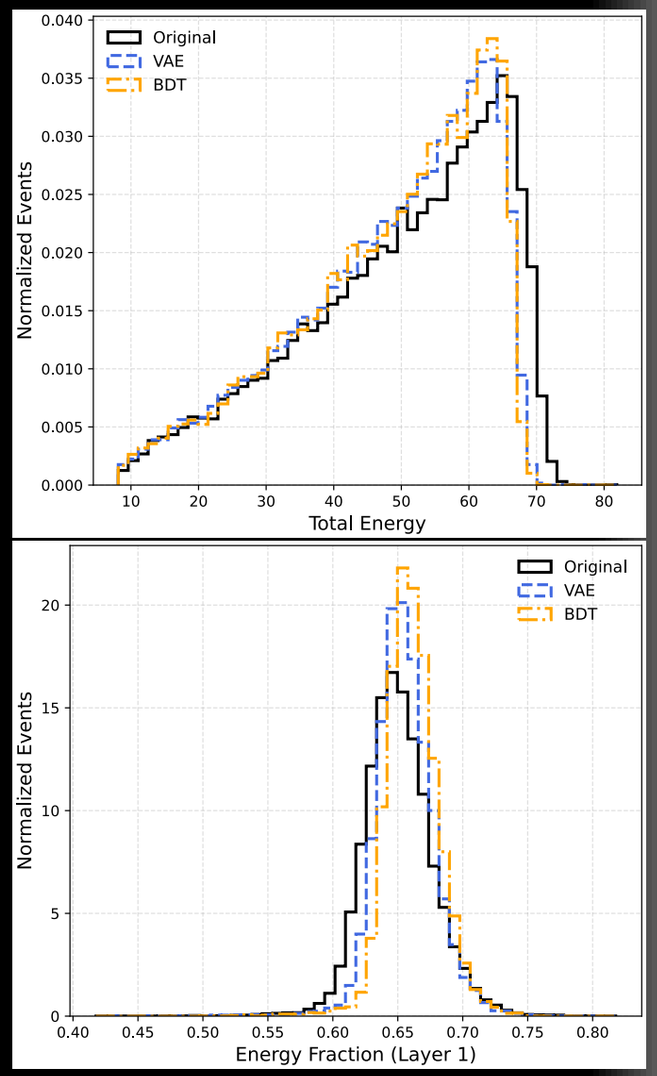
- Paganini, de Oliveira & Nachman PRD 97 (2018) 1, 014021
  http://doi.org/10.1103/PhysRevD.97.014021

We convert this left part into decision trees

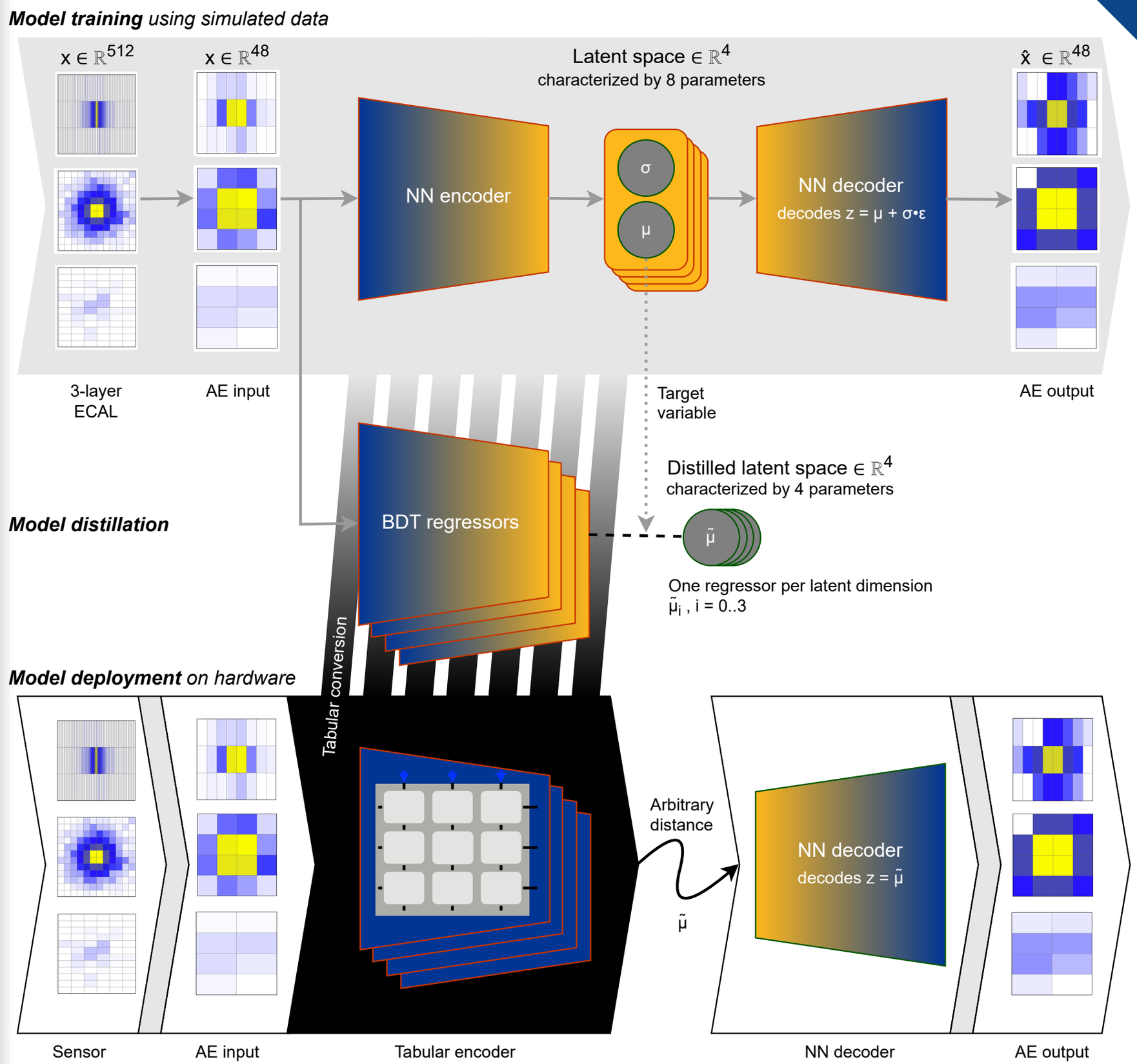


Tabular encoder

- Physics is preserved



- Convert VAE into tabular format
- Tabular? Next slide



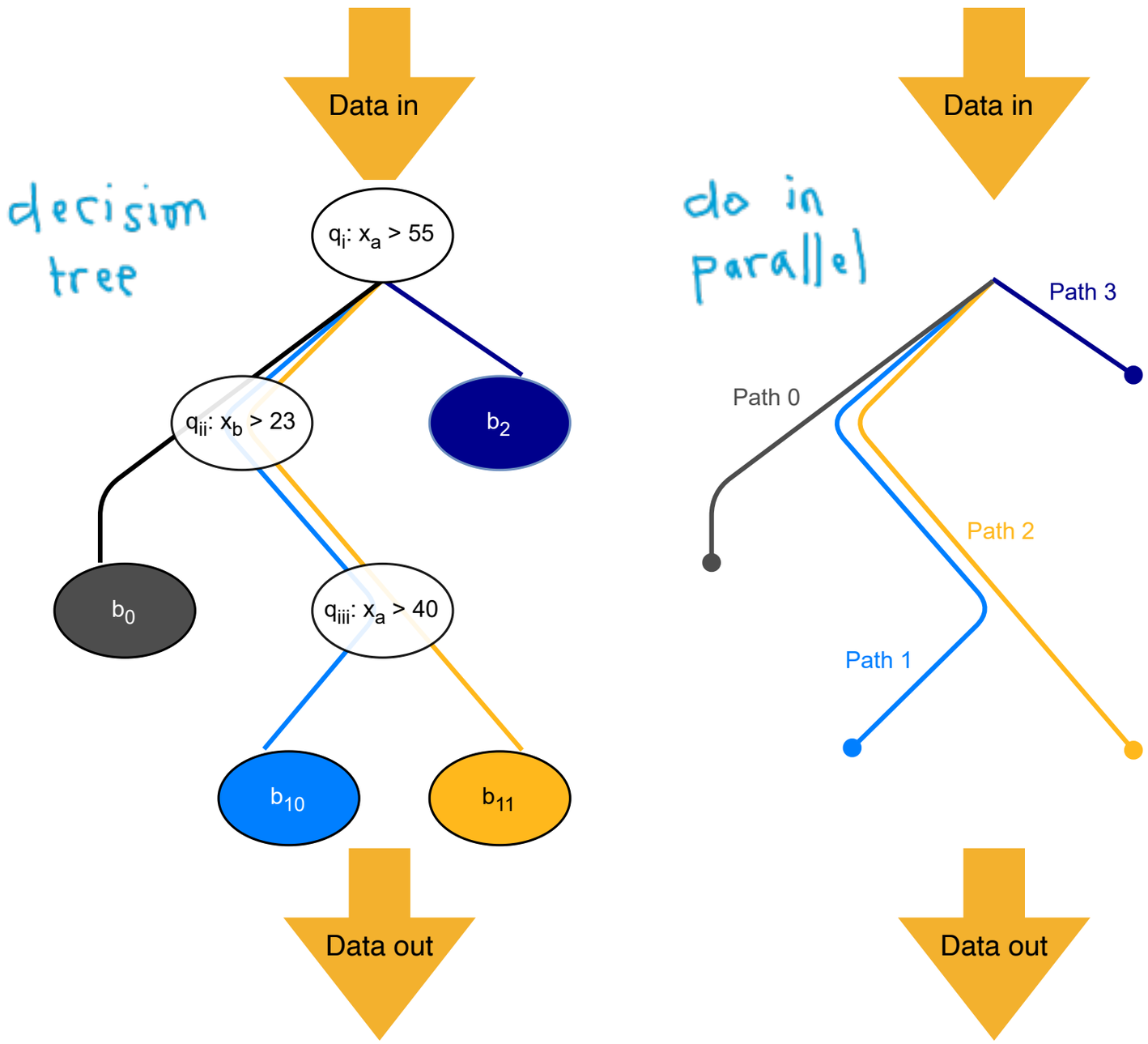
- Gupta, TMH et al., Paper in preparation

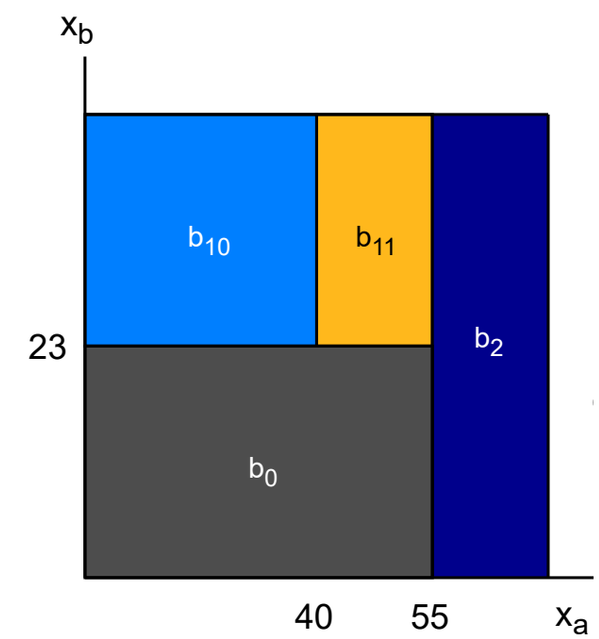
Hong Pittsburgh

Tabular?

Parallel implementation of decision tree on FPGA

- B. Carlson, TMH et al.
  J. Instrum. 17 (2022) P09039
  http://doi.org/10.1088/1748-0221/17/09/P09039

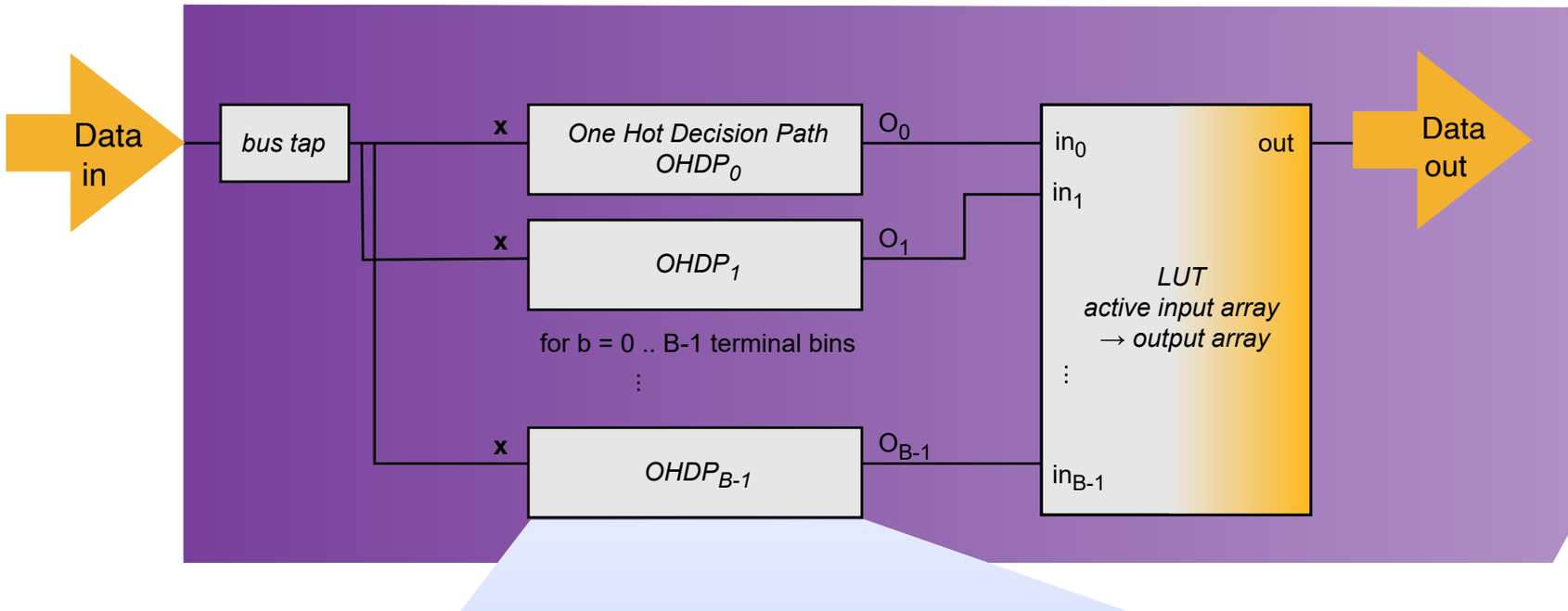


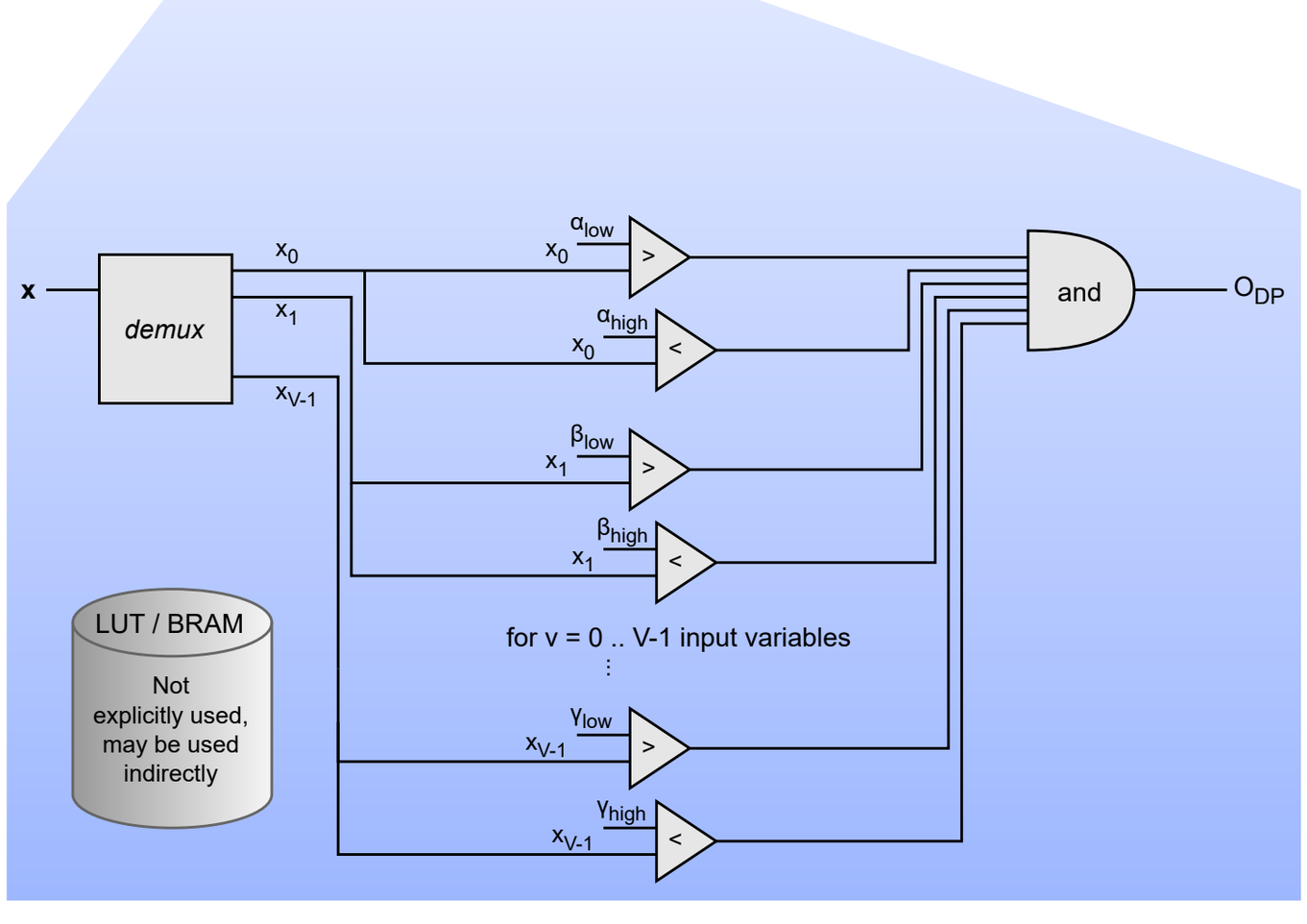


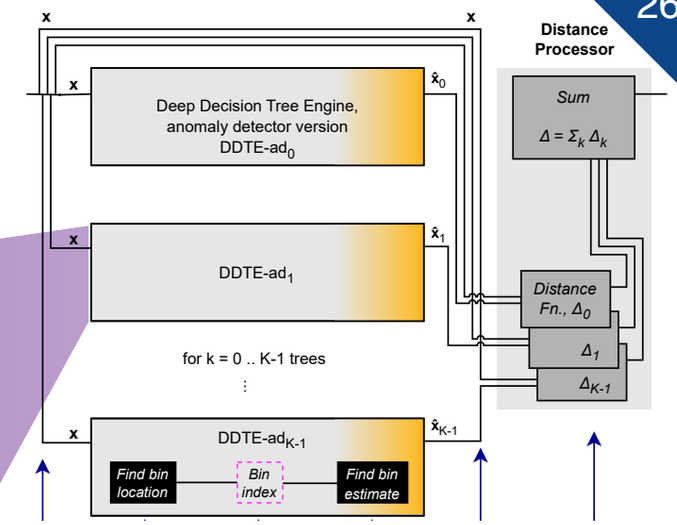
Destination bin	Decision path
b_0	not(q_i) and not(q_{ii})
b_2	q_i
b_{10}	not(q_i) and q_{ii} and not(q_{iii})
b_{11}	not(q_i) and q_{ii} and q_{iii}

Tabular design

Parallel implementation







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

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

Hong Pittsburgh

Fun slide - question

Can you guess what these are? Hint: Belgian traffic signs

https://btsd.ethz.ch/shareddata

AE output

- Timofte et al.,
  IEEE Workshop on Appl. of Comp. Vision, WACV 2009
  https://btsd.ethz.ch/shareddata/publications/Timofte-WACV-2009.pdf

Fun slide - answer

Hong Pittsburgh

Were you right?

bicycle | bicycle | road ends | don’t pass | wrong way

AE input

Tabular VAE

AE output

- Timofte et al.,
  IEEE Workshop on Appl. of Comp. Vision, WACV 2009
  https://btsd.ethz.ch/shareddata/publications/Timofte-WACV-2009.pdf

Python-based code

Hong Pittsburgh

Availability

- gitlab.com/PittHongGroup/fwX

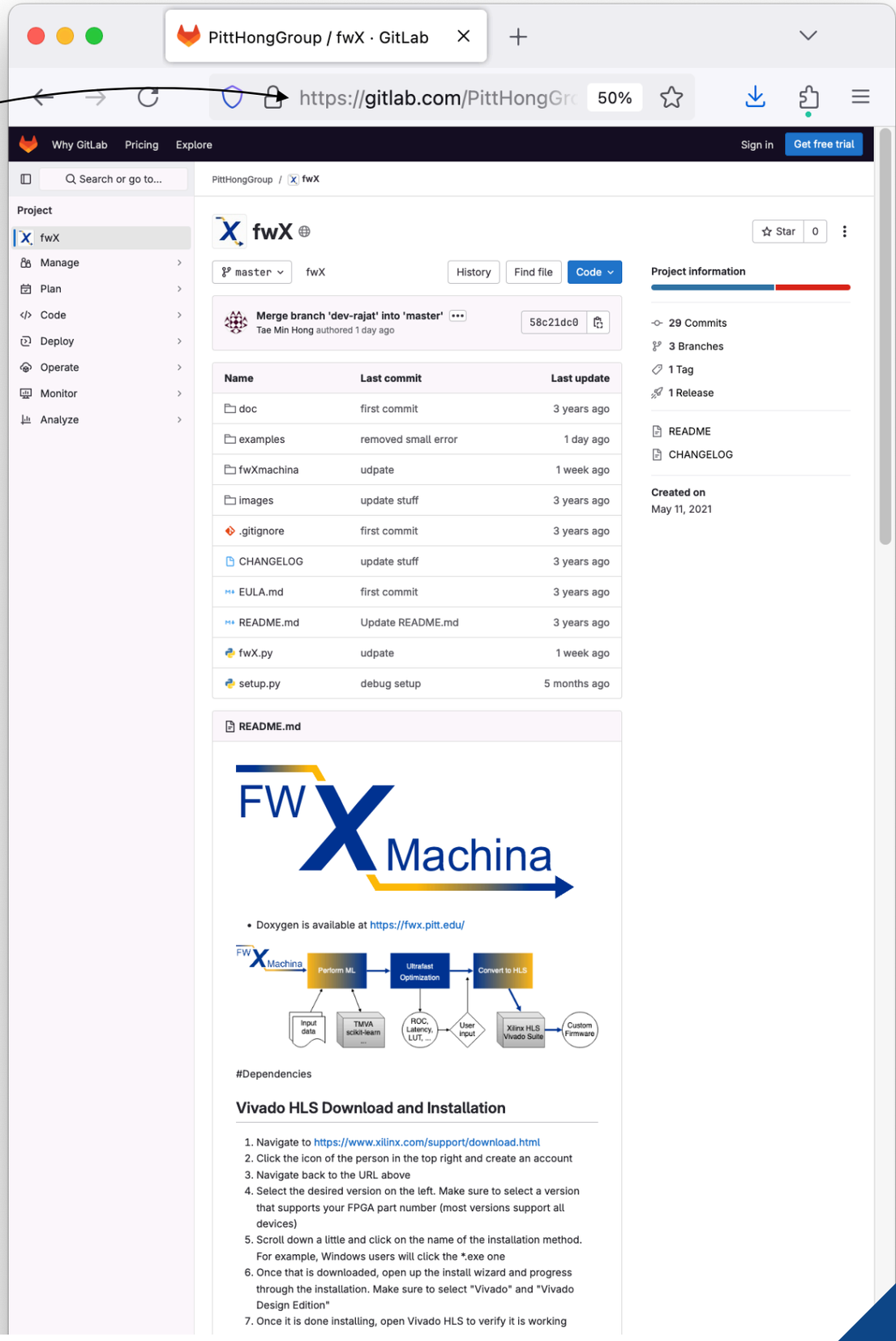
  parallel cuts (paper 1)

- 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

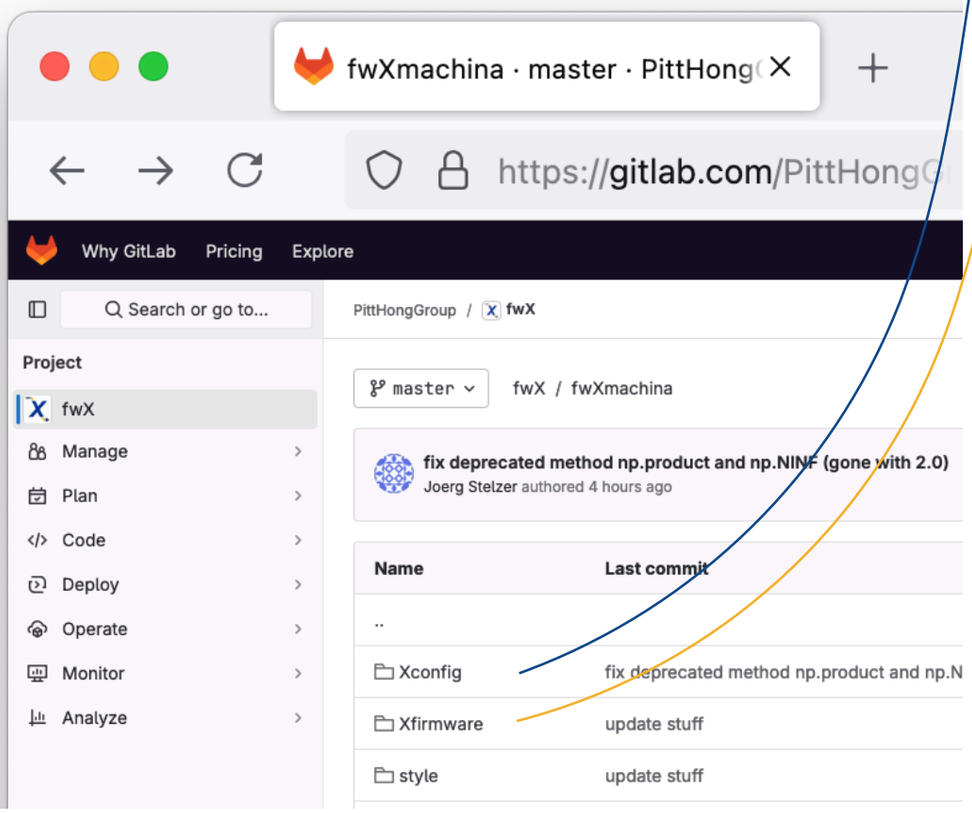
  parallel cuts (paper 1) - tutorial today

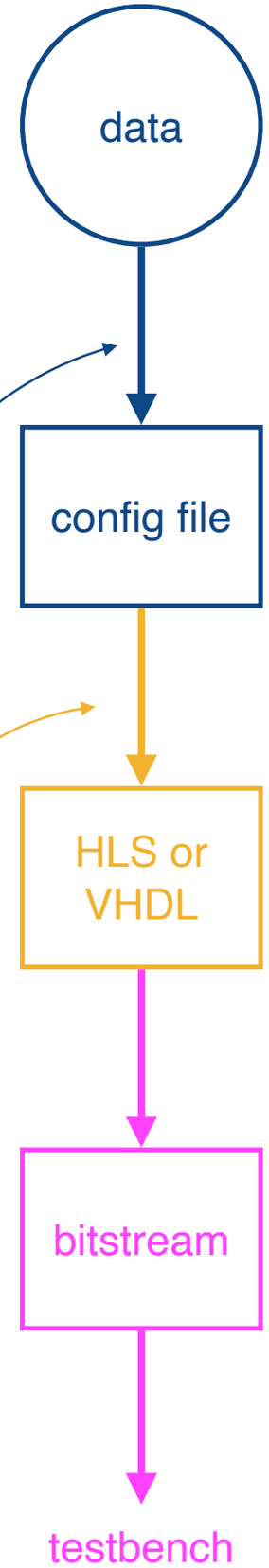
- Available by request

  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

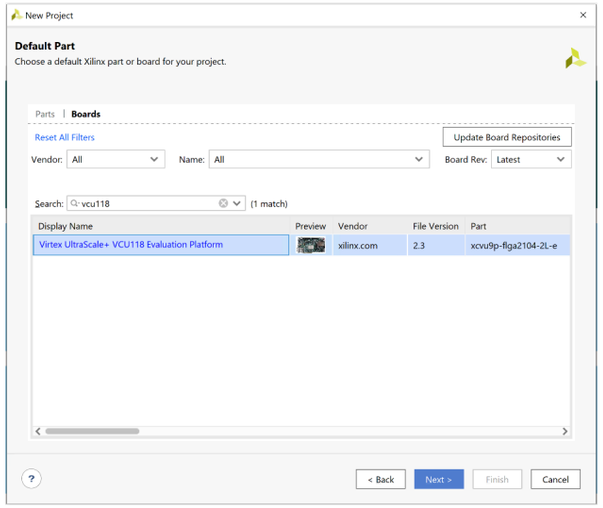
Before Beginning

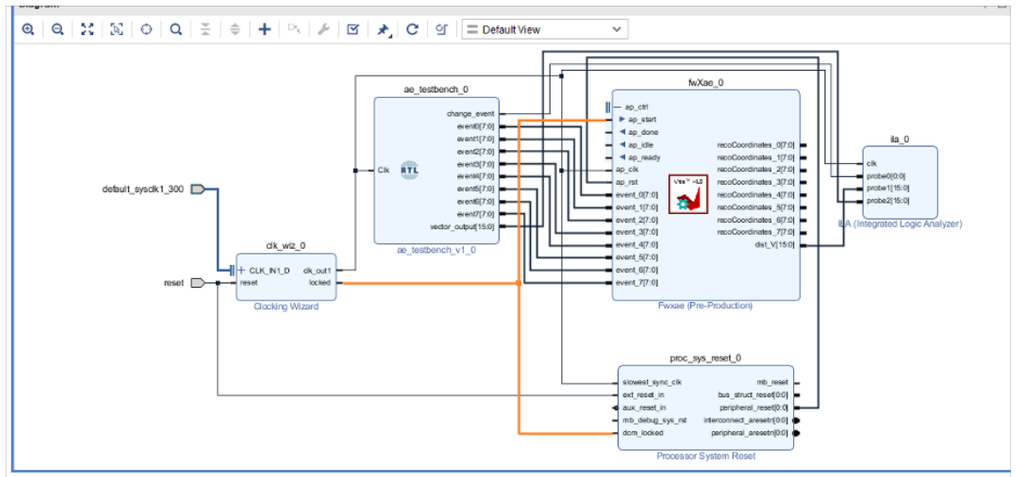
Before beginning, please make sure that you have (and know the location of) the autoencoder IP folder, and the VHDL testbench files:

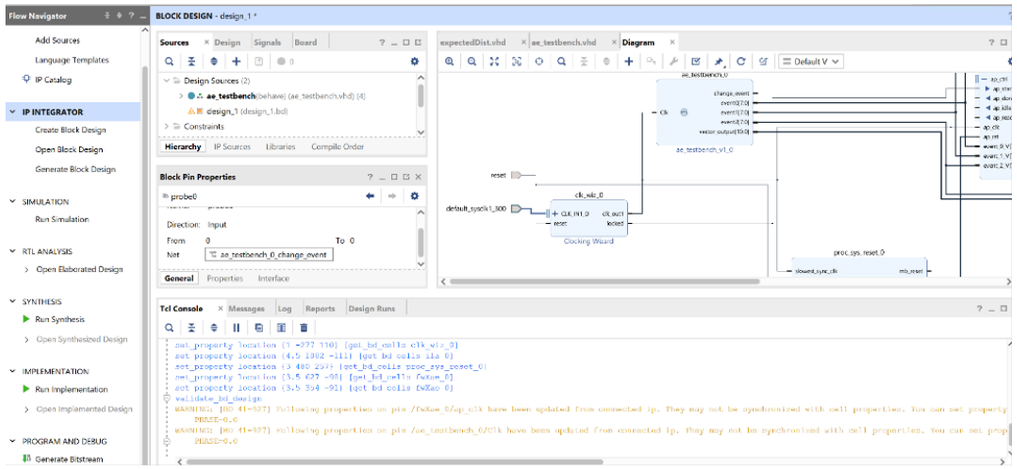
Name	Date modified	Type	Size
autoencoder8var_ip	2/7/2024 1:30 PM	File folder	
tb_vhd_files	2/8/2024 11:50 AM	File folder	

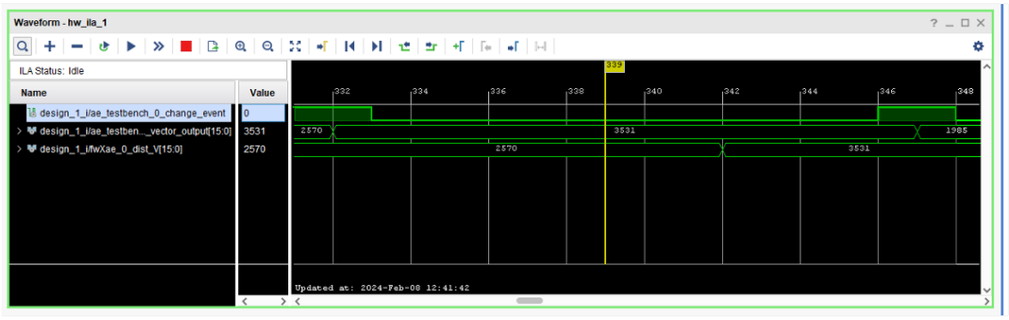
Creating New Project in Vivado

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.







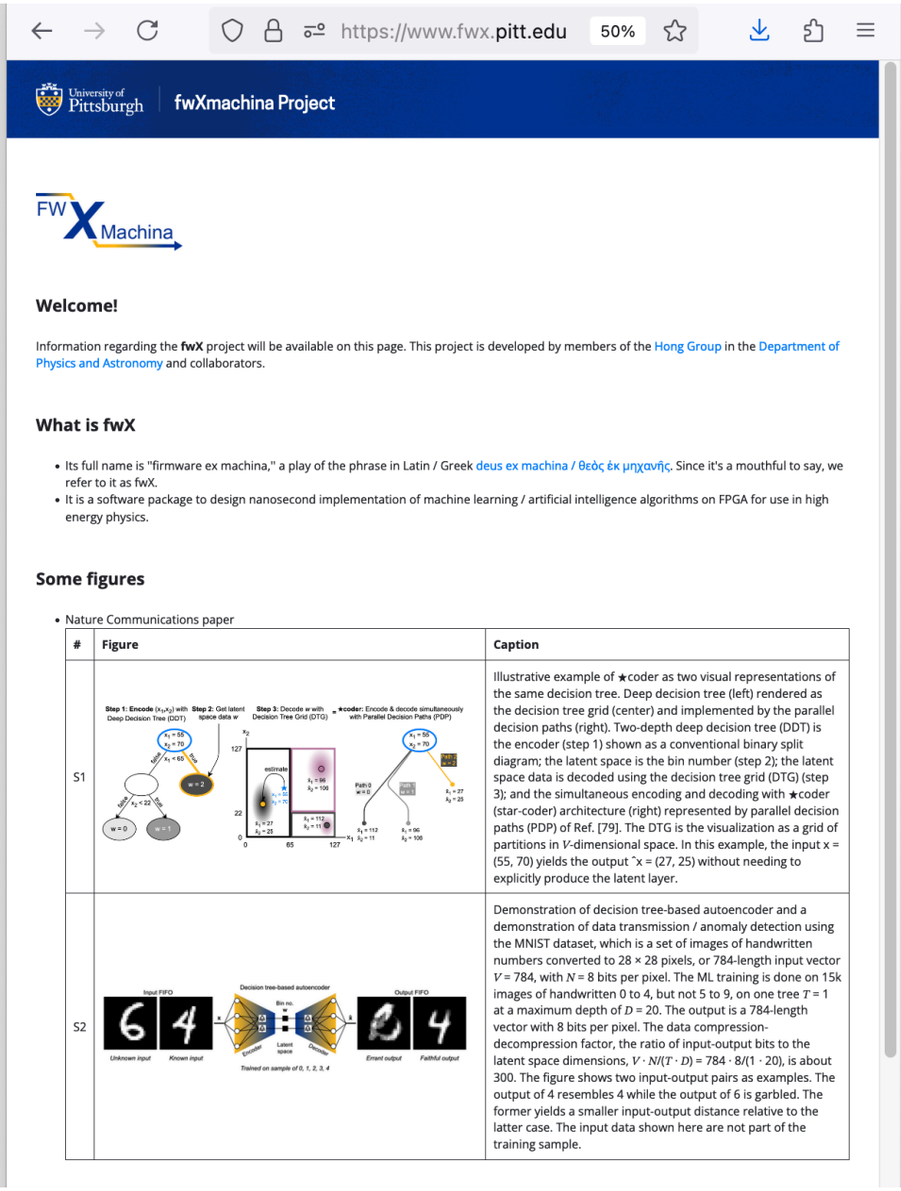


More info

Hong
Pittsburgh

Start page

- fwx.pitt.edu



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/