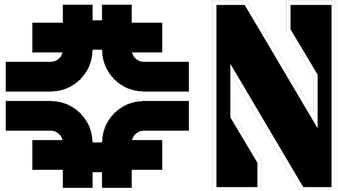
Real-time machine learning in embedded systems for particle physics

Nhan Tran
Fermilab/Northwestern ECE
October 12, 2020



Outline

Disclaimers & Motivations

Real-time opportunities

Efficient machine learning

Example applications

LHC Trigger, Accelerator Controls, Reconfigurable AI ASICs

Disclaimers & Motivation

Disclaimers

- **Disclaimer 1:**
  - This is a very exciting and rapidly moving field! I apologize for any biases I have based on my research focus. I am excited to hear about other work in this space
    (I am member of CMS experiment at LHC and based at Fermilab)

- **Disclaimer 2:**
  - Though I will talk about tools, methods, and example applications - it is still early days - much is still unexplored

- **Disclaimer 3:**
  - I have cut off the scope of this talk at embedded systems — but FPGAs and ASICs are also used for heterogeneous computing and have common techniques and lessons, see Mia Liu's talk on Tuesday

Motivation

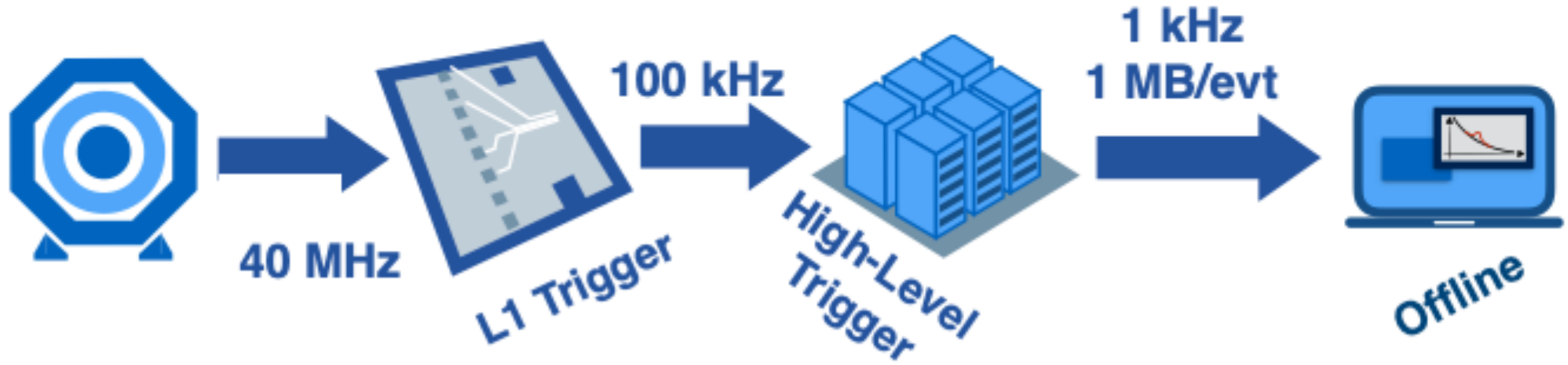
(a biased view that I hope you share!)

- Cutting edge science requires...
  Faster, more precise, bigger, more granular,...

- Science at the ~MHz scale and beyond creates massive amounts of data
  Online compression, featurization, filtering, feedback, etc. is vital - arguably more so than offline machine learning

- Real-time intelligence can greatly accelerate science allowing us to test hypotheses faster, enhance performance of detectors/accelerators, and save potentially lost data

CMS real-time processing



1 ns 1 us 1 ms 1 s

Custom electronics
Latency ~ 25ns - 10 µs

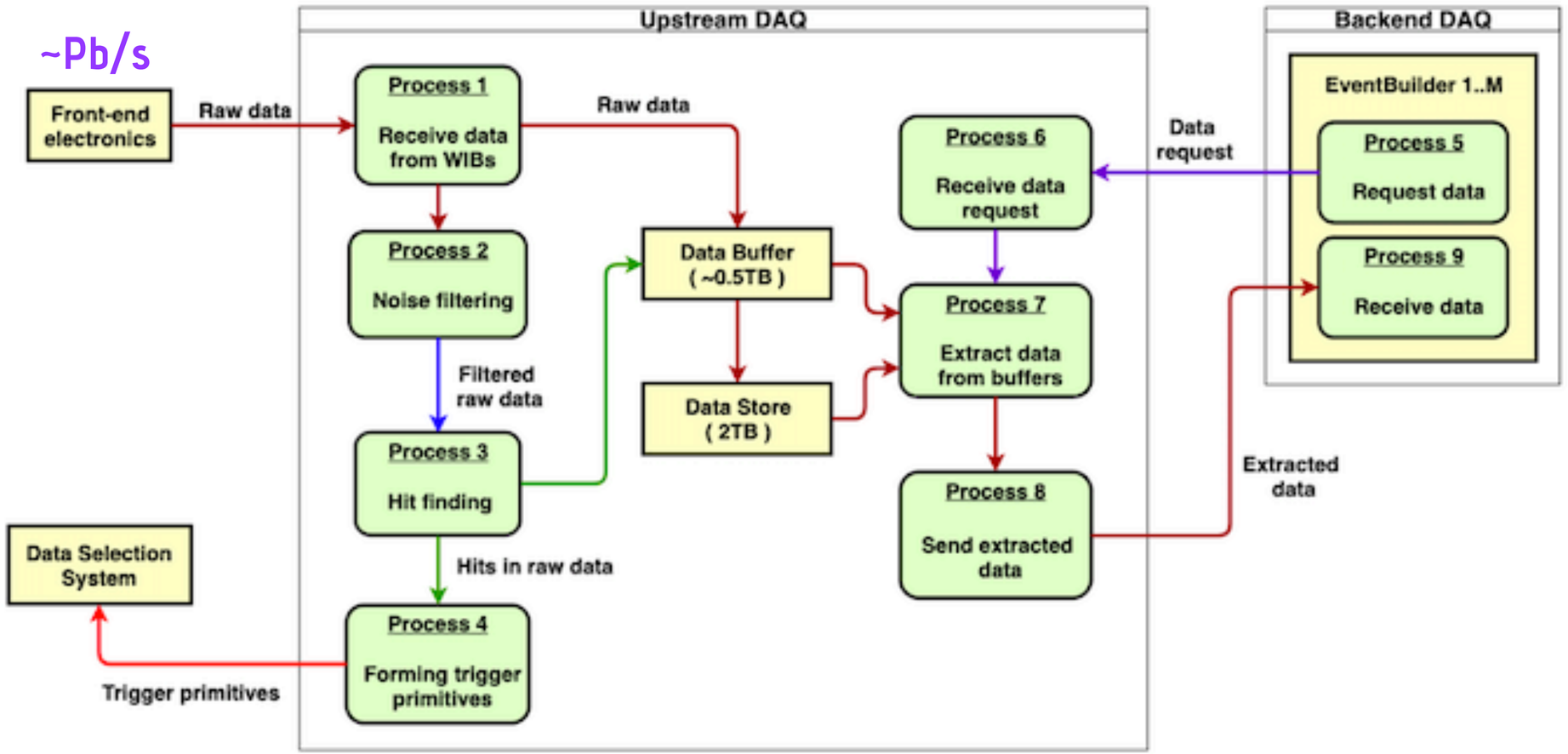
FPGAs/ASICs - high bandwidth low latency specialized compute hardware

Off-the-shelf computing
Latency ~ O(1+ ms)

"standard" CPU computing, coprocessors

DUNE upstream DAQ

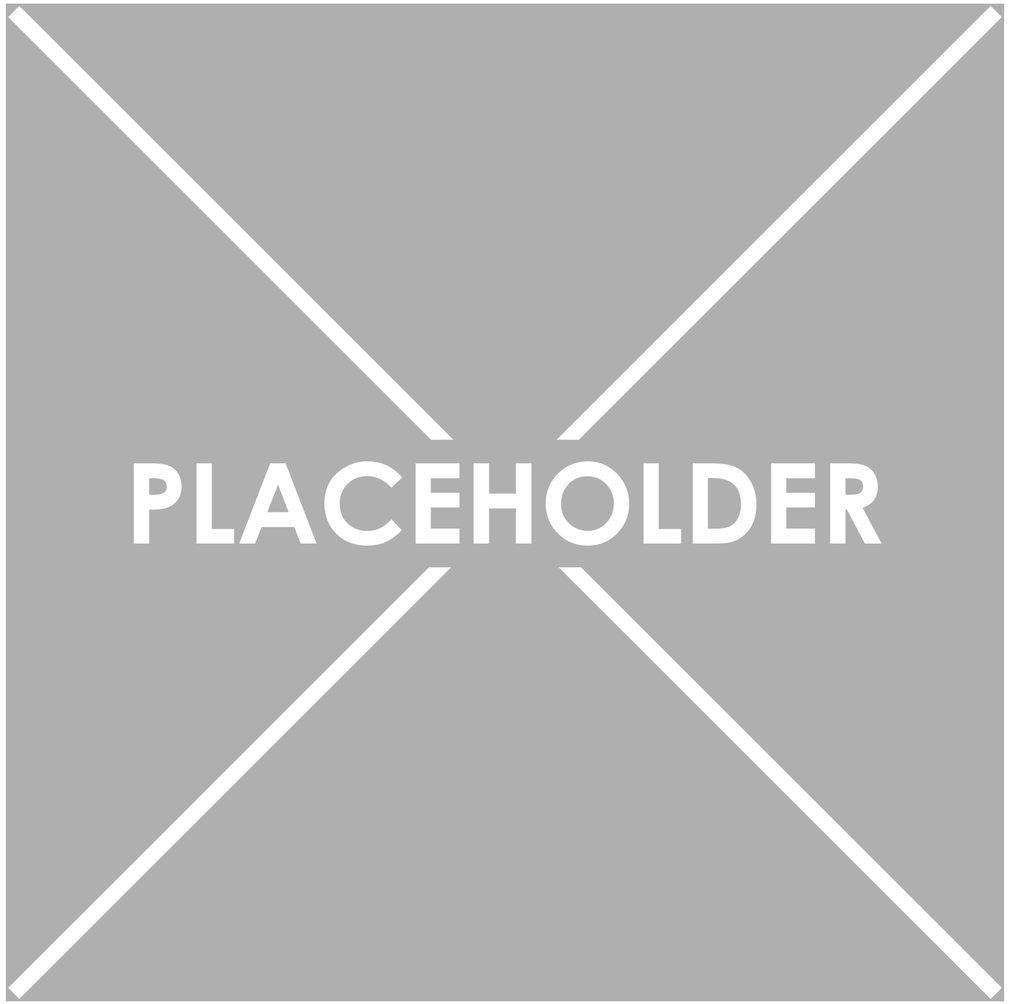
DUNE TDR



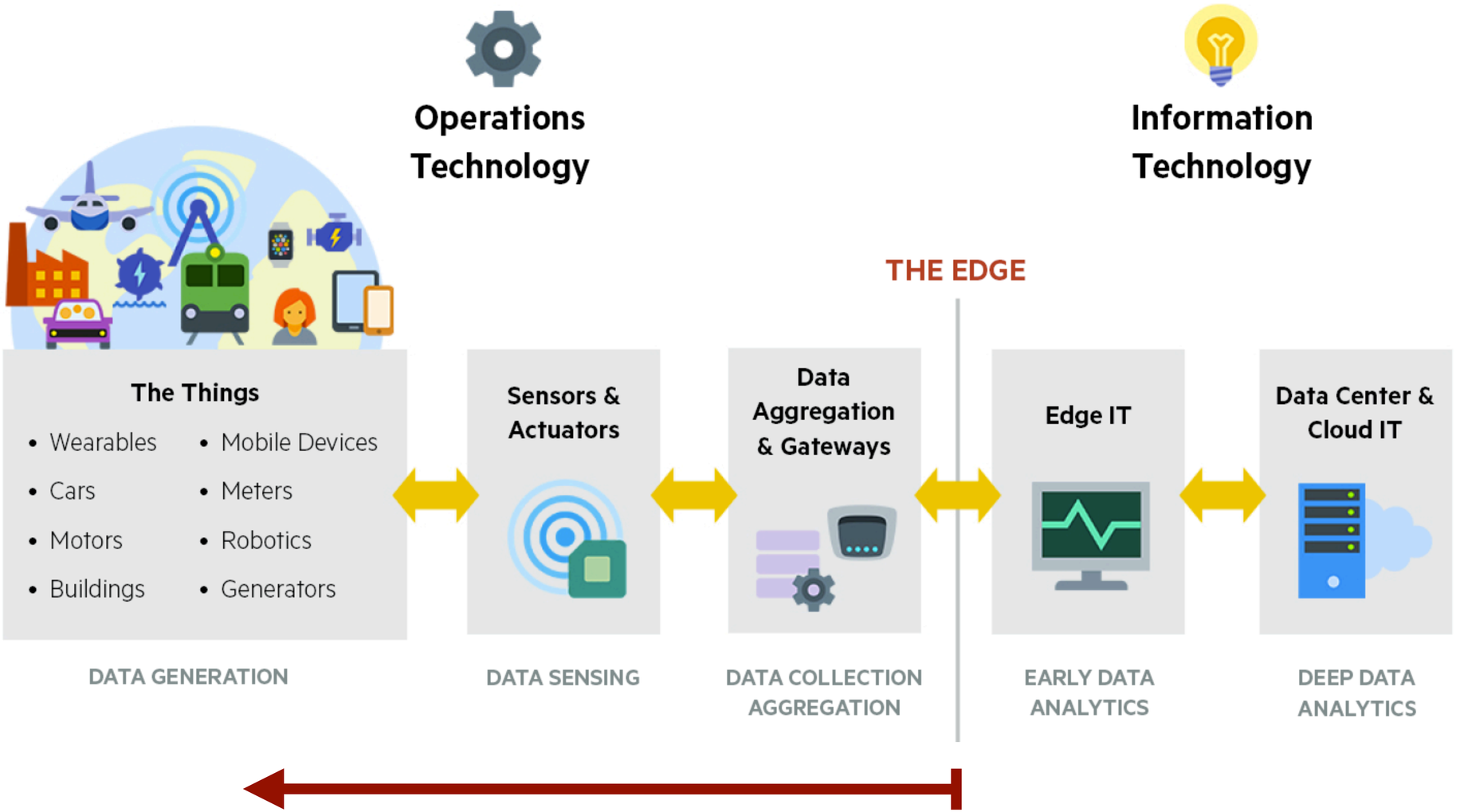
Accelerator controls

Fermilab accelerator complex

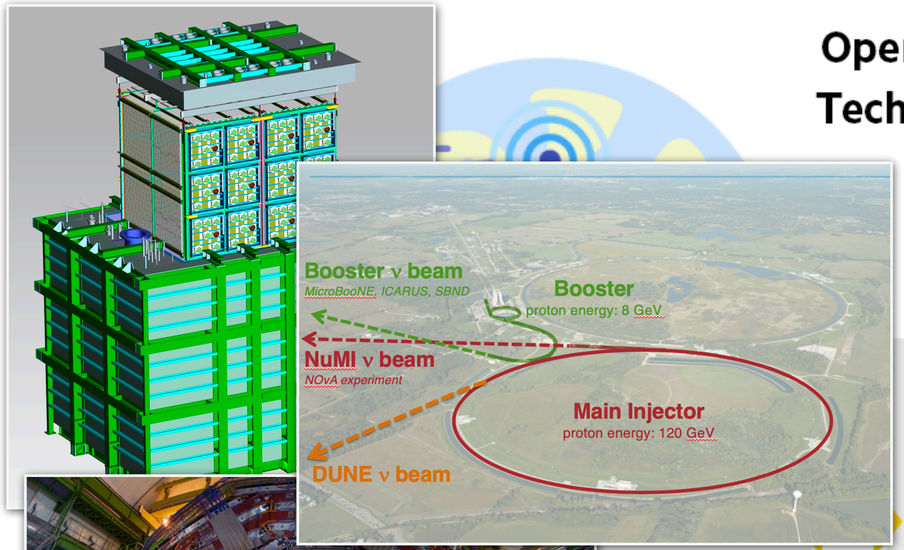
Insert your application here

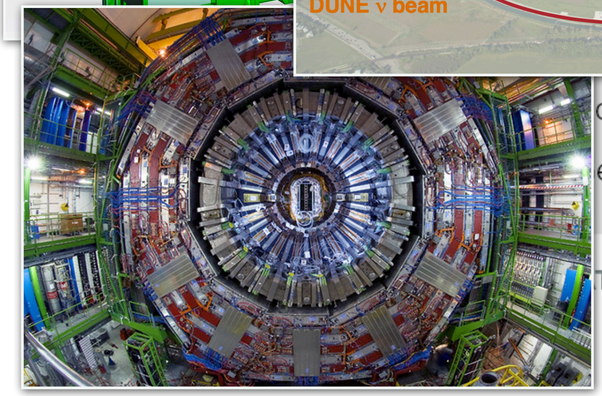


Internet of things...particle physics



Internet of things...particle physics





Operations Technology

Information Technology

THE EDGE

obotics

enerators

TION

Sensors & Actuators

Data Aggregation & Gateways

Edge IT

Data Center & Cloud IT

DATA SENSING

DATA COLLECTION AGGREGATION

EARLY DATA ANALYTICS

DEEP DATA ANALYTICS

Pushing intelligence to the edge

Efficient machine learning

{implementations, optimizations, tool flows for}

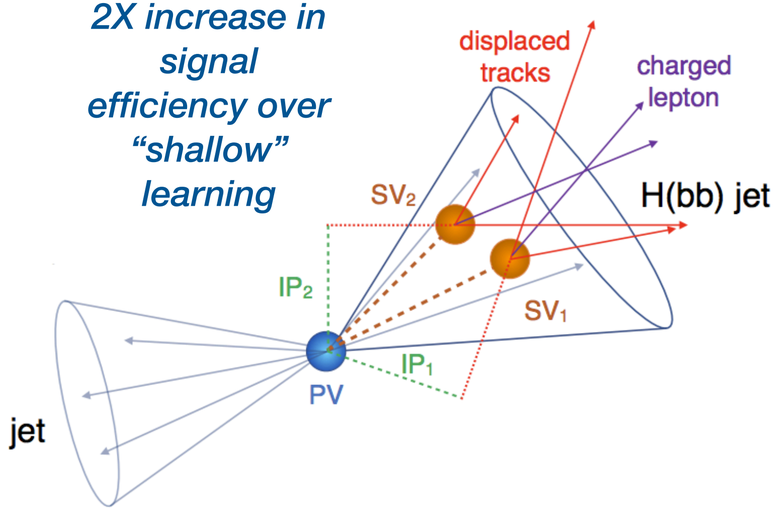
Efficient machine learning

Machine learning

Energy

Identification of boosted Higgs jet decay to two bottom quarks

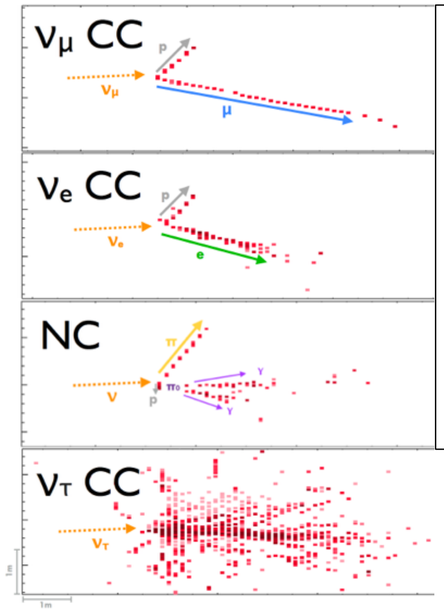
2X increase in signal efficiency over "shallow" learning



J. Duarte et al., CMS DP-2018/046

Intensity

NOvA event classification



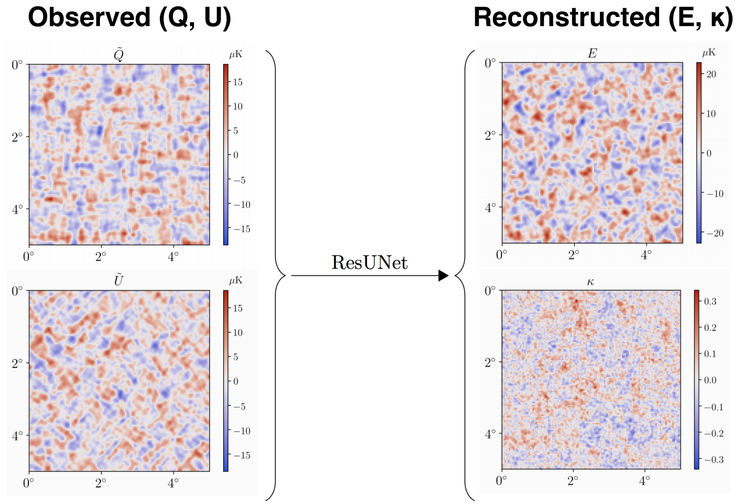
Deep learning improvement:

Effective 40% increase in NOvA active volume

A. Himmel, E. Niner, F. Pshihas et al.
https://arxiv.org/abs/1604.01444
1st deployed in oscillation analysis
https://arxiv.org/abs/1703.03328

Cosmic

Reconstruction of CMB polarization map from Stokes parameters

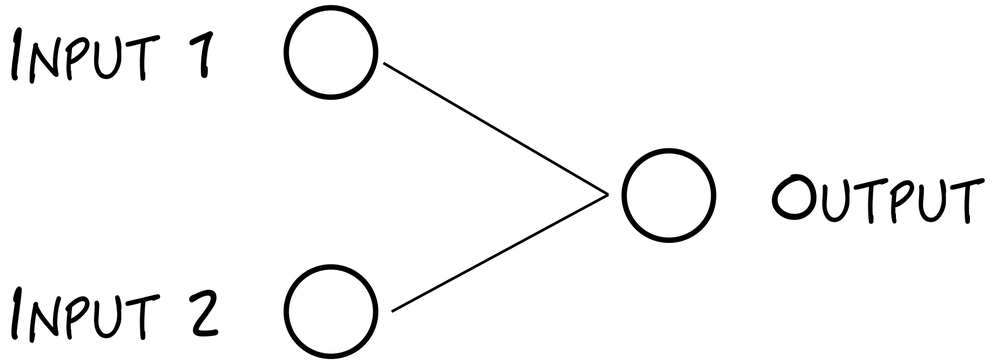


50% less noise vs. traditional methods across large range of scales

J. Caldeira, B. Nord, et al.,
https://arxiv.org/abs/1810.01483

ML applications is huge field by itself - see talk by Christof Buehler on Thursday
Many open opportunities for new applications as well!

NN inference in a nutshell

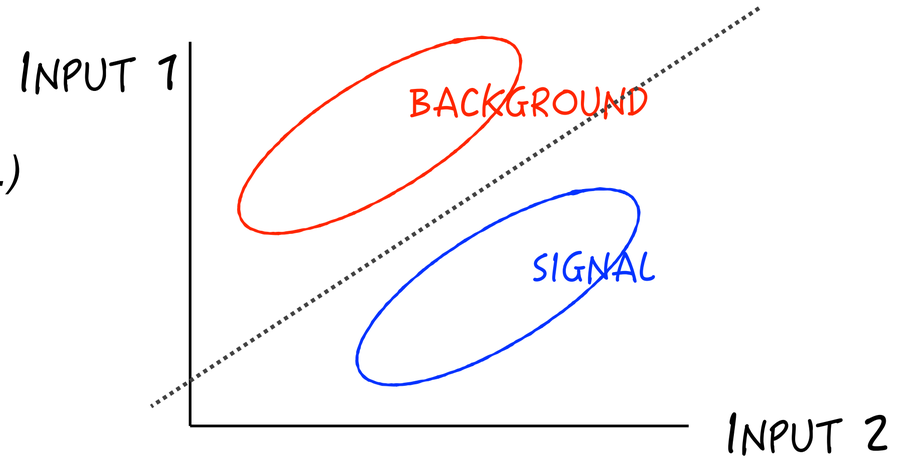


$$\vec{O_j} = \vec{I_i} \times \overleftrightarrow{W_{ij}} + \vec{b_j}$$

Simple 2 input example

(Fisher linear discriminant, linear support vector machine,…)

$$O_1 = I_1 \times W_{11} + I_2 \times W_{21} + b_1$$

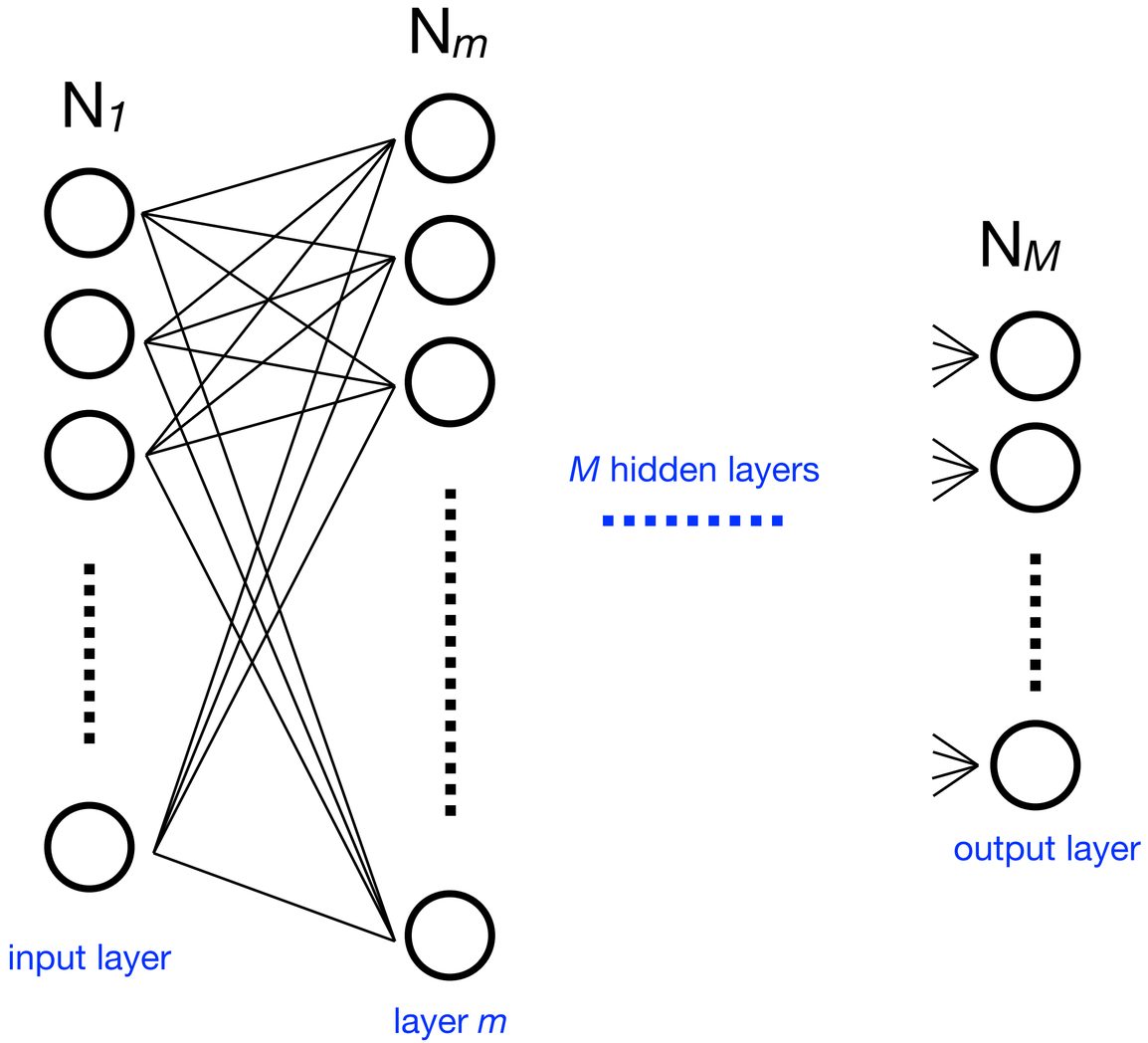


NN inference in a nutshell

$$\vec{O_j} = \Phi(\vec{I_i} \times \overleftrightarrow{W_{ij}} + \vec{b_j})$$

Φ = ACTIVATION FUNCTION (NON-LINEARITY)

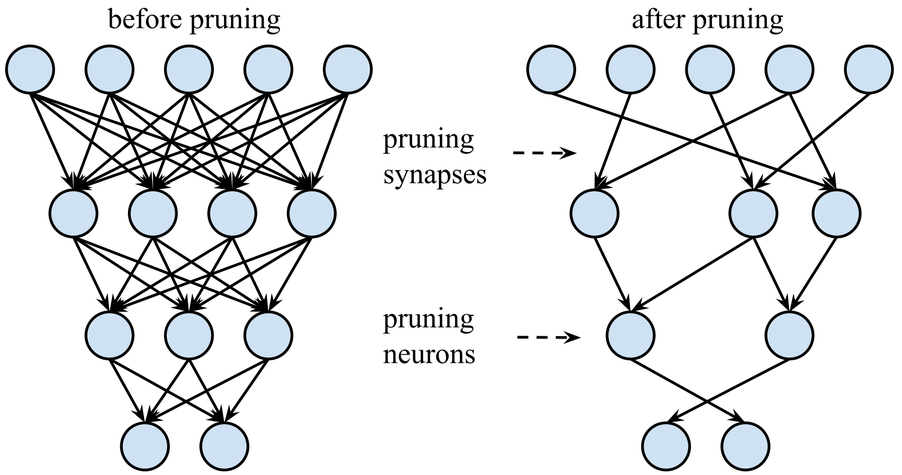
NN inference =
a bunch of multiplications /additions and LUTs (look up tables) for activation functions



FULLY CONNECTED HIDDEN LAYER

(Energy) Efficient Neural Networks

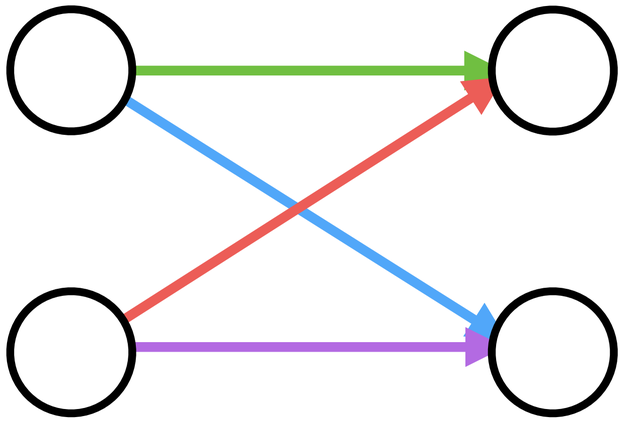
- Important engineering field, efficient implementation of NN architecture
- **Parallelization**: performing operations simultaneously
- **Compression/Pruning**:
  - maintain the same performance while removing low weight synapses and neurons (many schemes)
- **Quantization/Approximate math**:
  - 32-bit floating point math is overkill
  - 20-bit, 18-bit, ...? fixed point, integers? binarized NNs?

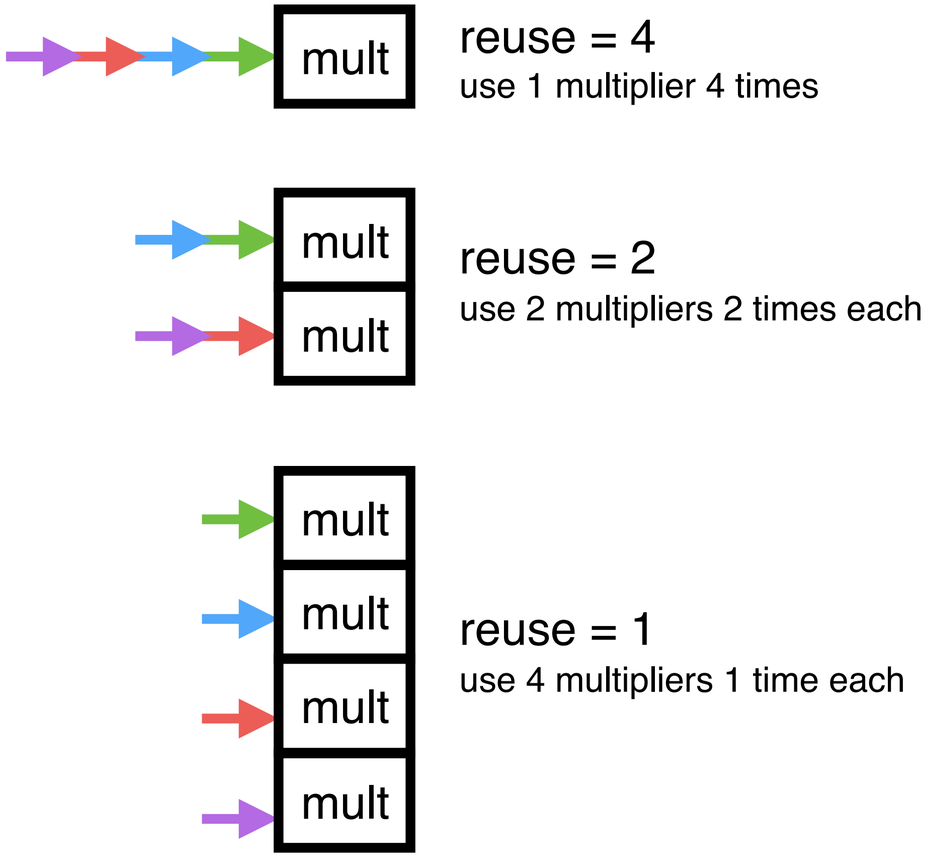


For further reading, start here: https://arxiv.org/pdf/1510.00149v5.pdf

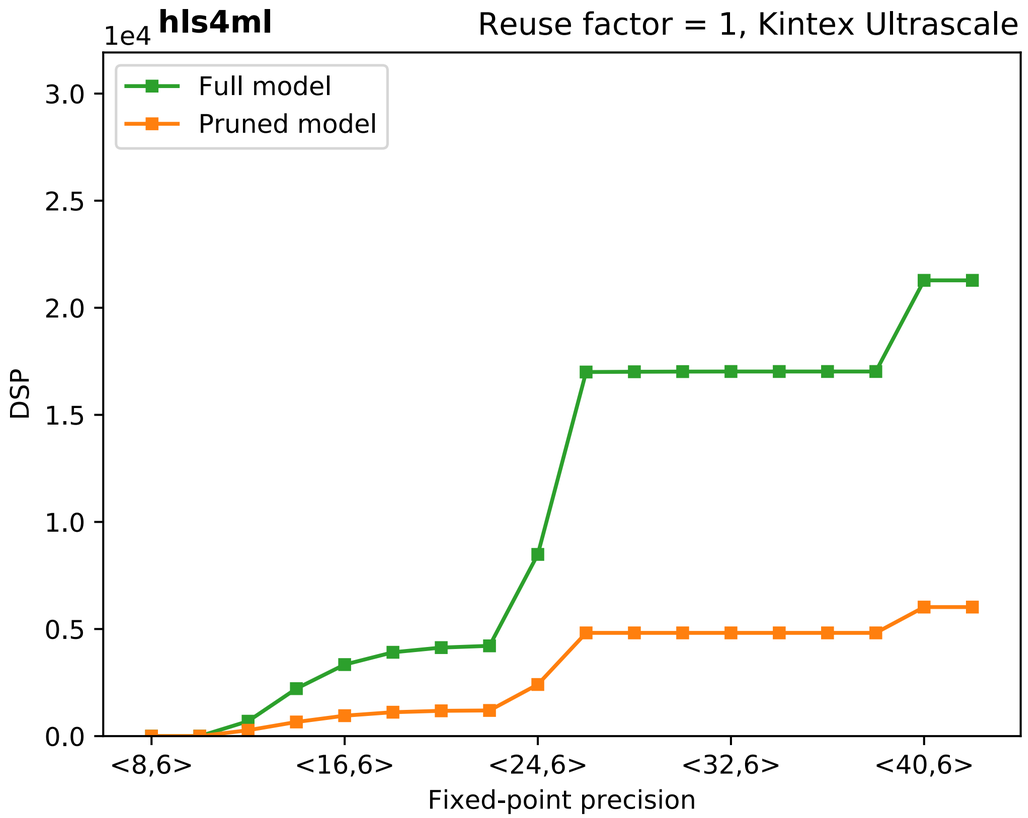
Example: Parallelization

ReuseFactor: how much to parallelize operations a hidden layer



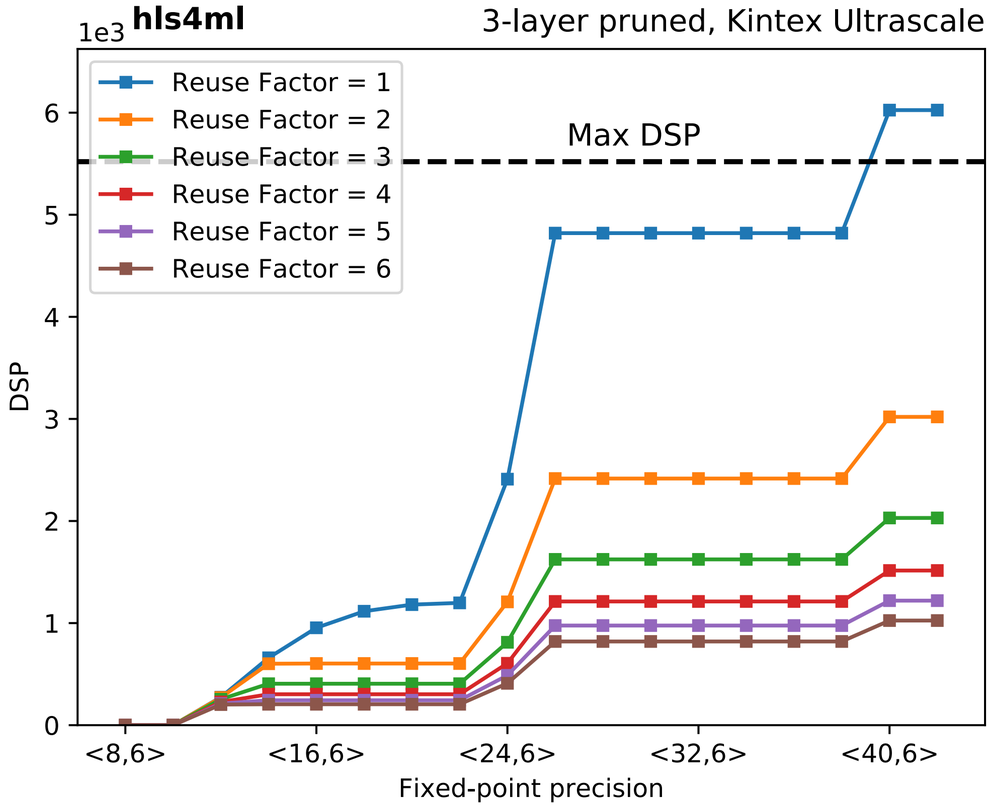


Parallelization, pruning, quantization

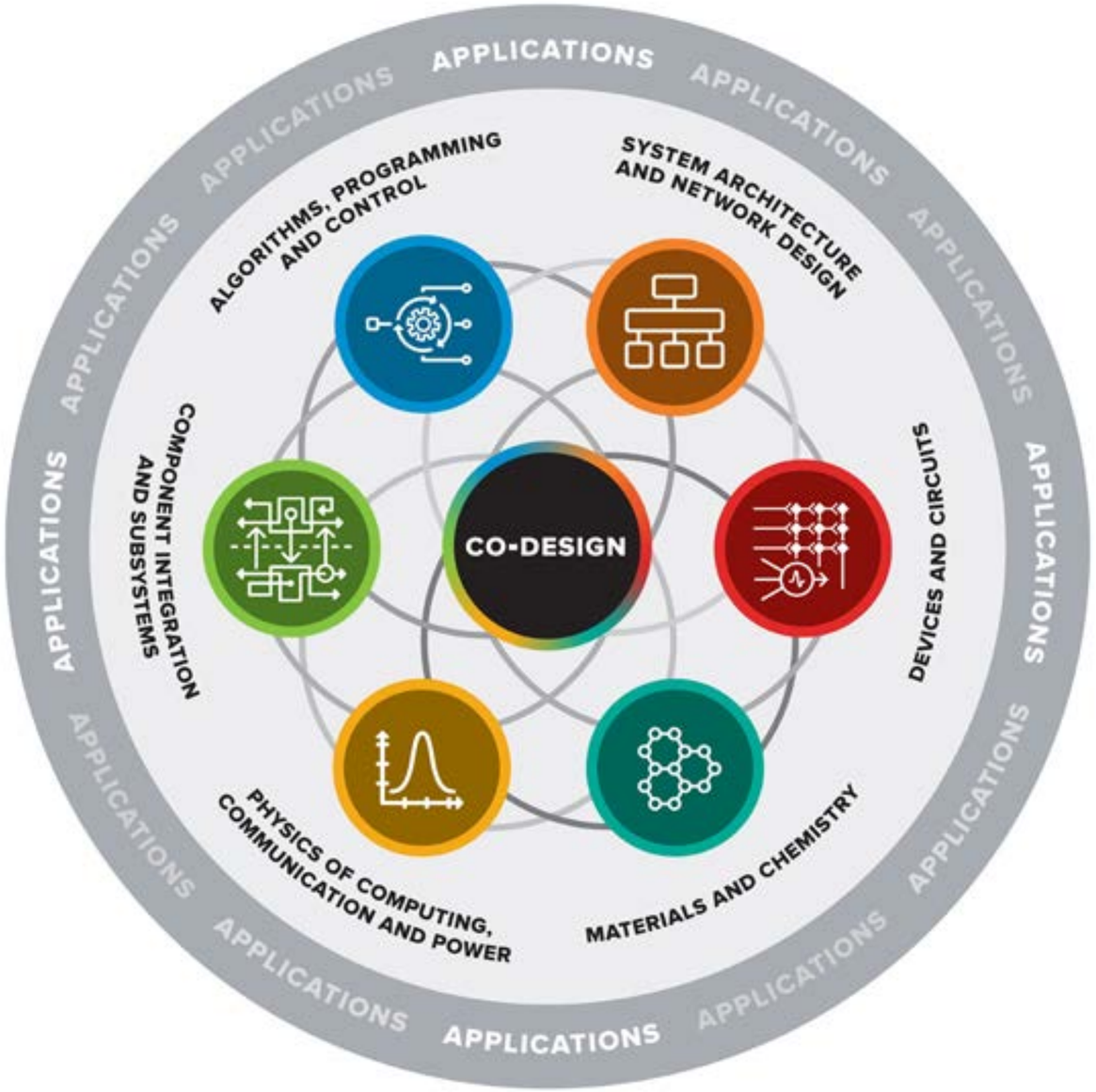


Unstructured iterative pruning with L1 regularization

Increased unroll factor to reduce DSP usage (at latency cost)



Codesign



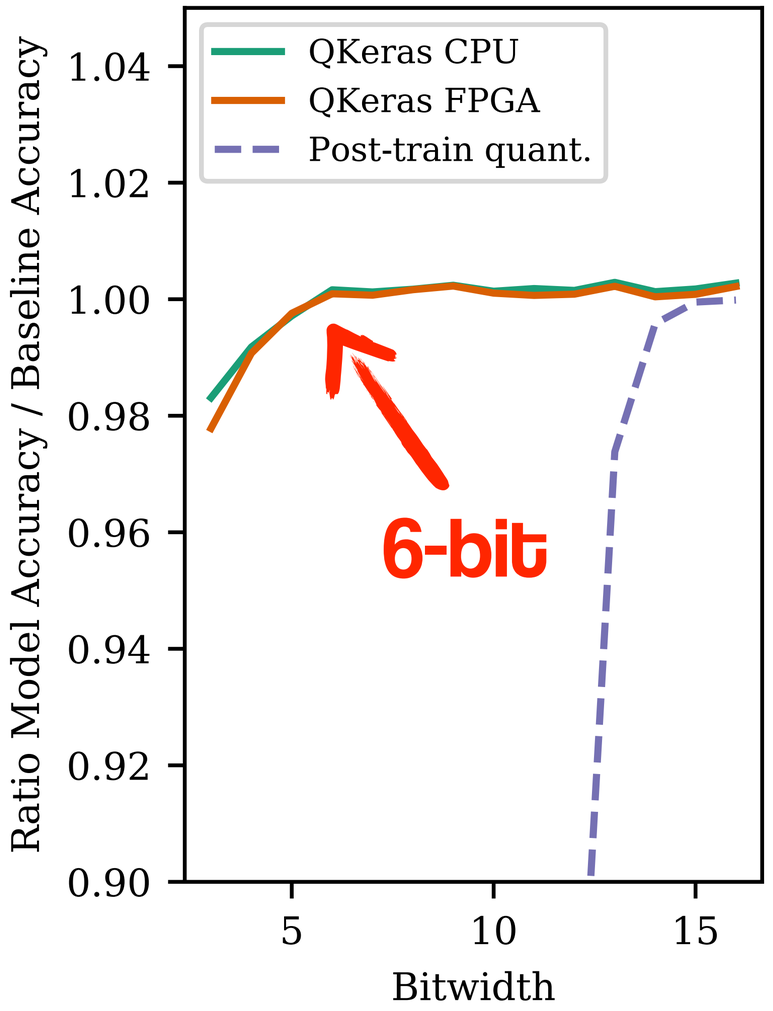
Intrinsic development loop between algorithm design, training, and implementation

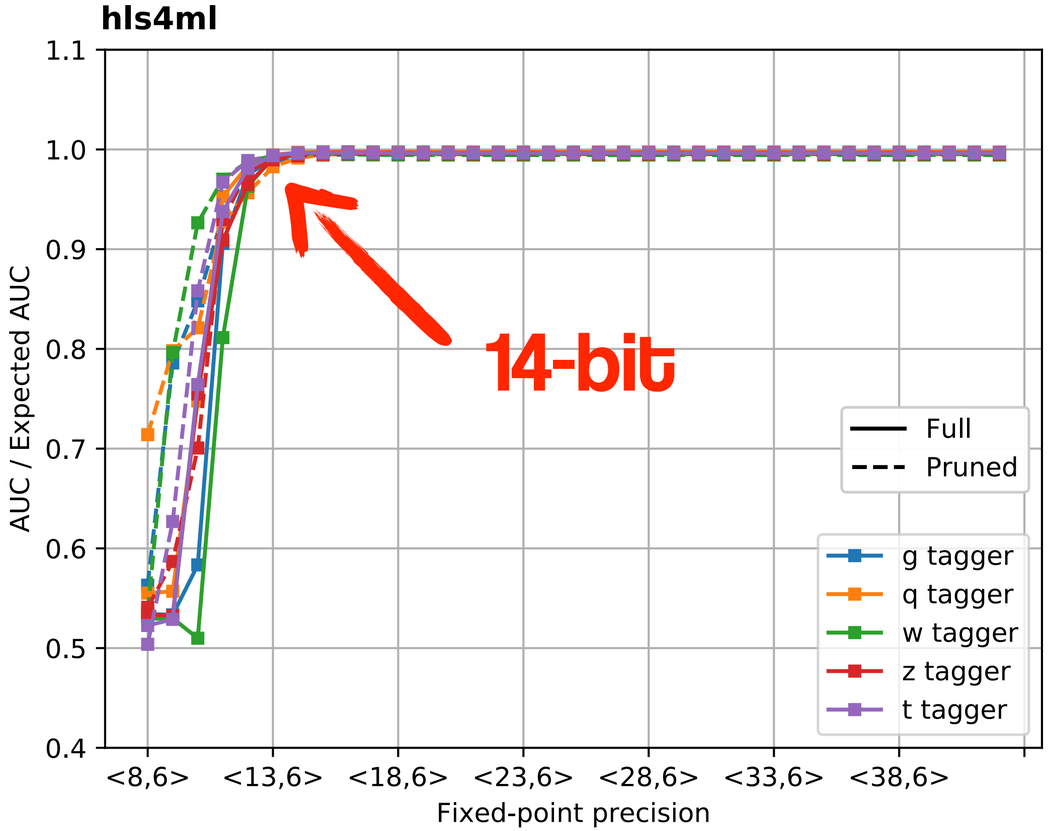
Acceleration of design space exploration is important

Q.A.T.

qkeras paper
arxiv:2006.10159

Quantization-aware training



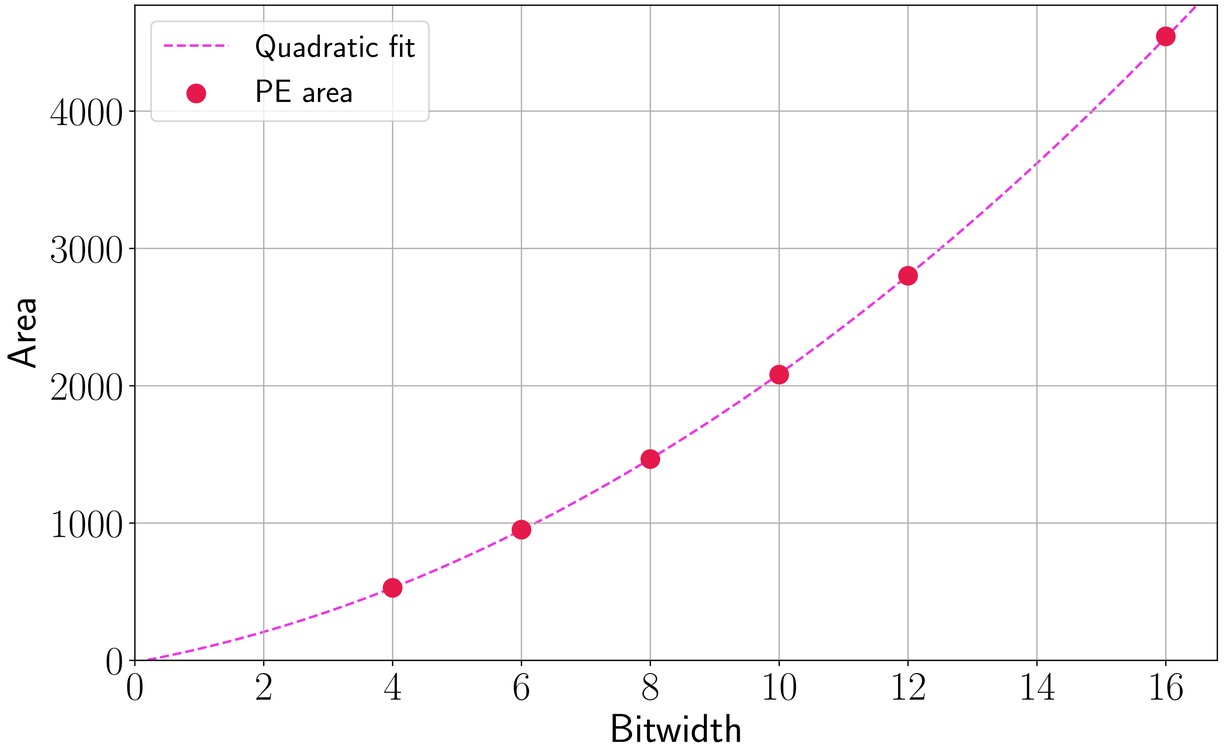


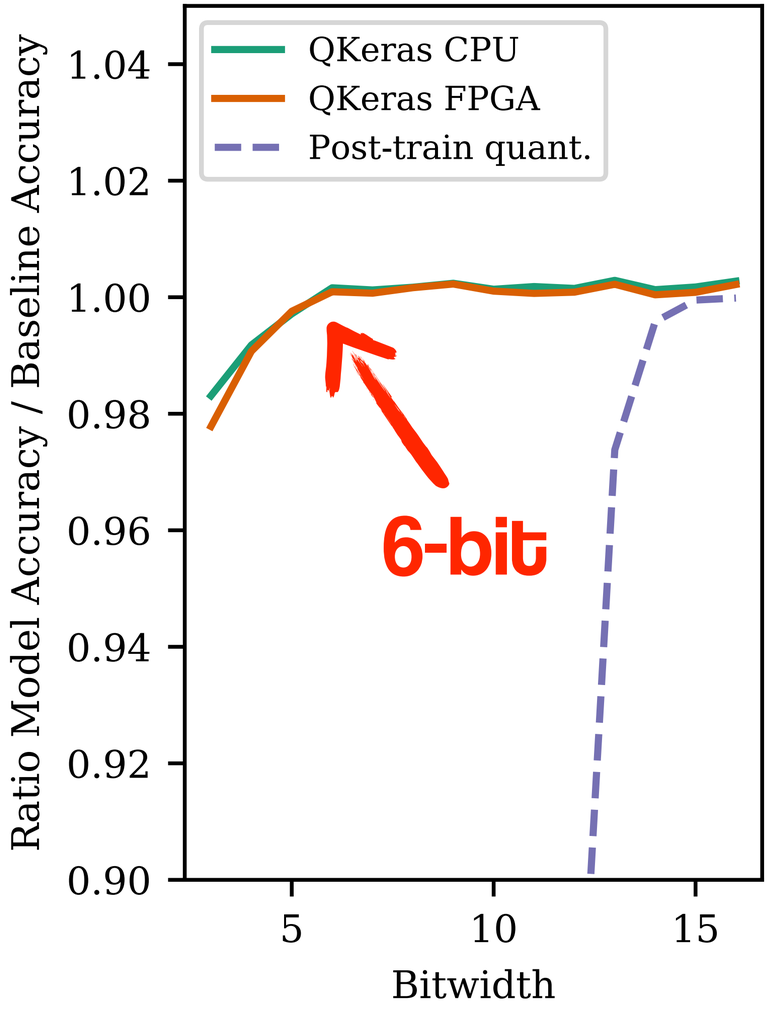
Post-training quantization

Q.A.T.

BOPS definition
arxiv:1804.10969

Quantization-aware training





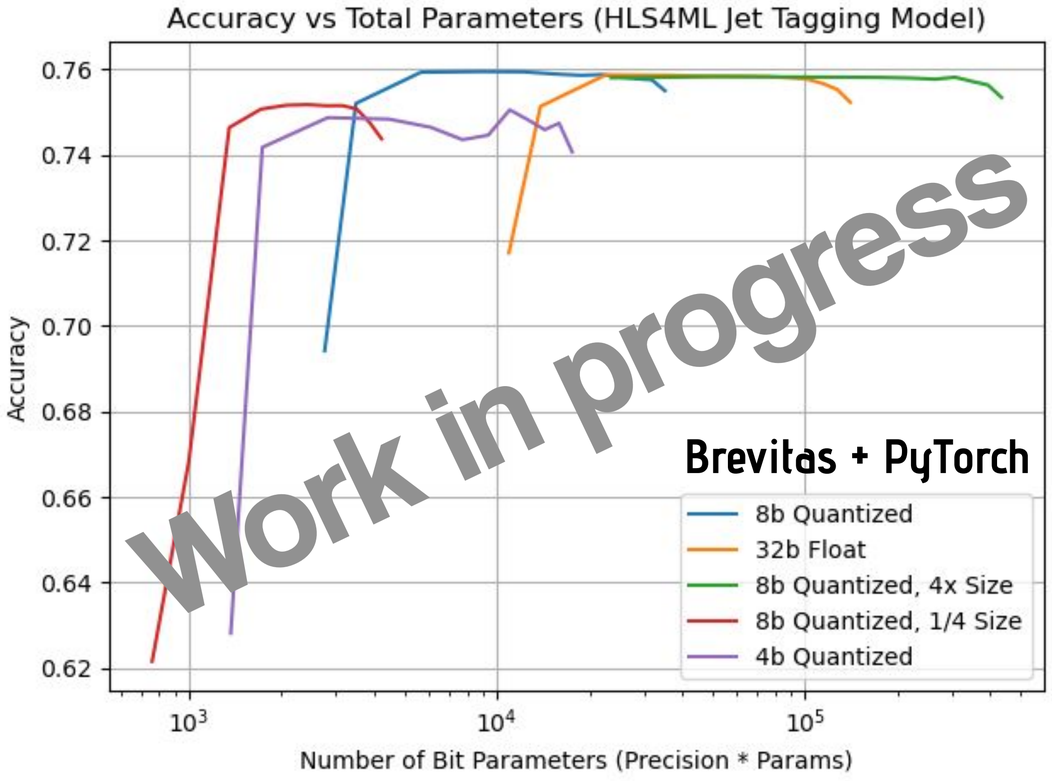
Area & power go quadratically with bit width

E.g. deeper ML questions

(a personal view)

Work in progress with Xliinx Research
Ben Hawks, J. Duarte, N.T. et al

- What about combinations of techniques? Unstructured (synaptic) pruning + QAT
  - How does that compare to hyper-parameter optimization and Bayesian Optimization?
- What did the network learn? How comparable are they? Active research on "neural efficiency" of networks and the "distance" between neural networks
- How does the learning (training) process affect efficiency? 2nd order optimization vs. stochastic gradient descent?



Domain-inspired ML

Energy Flow Networks
arXiv:1810.05165
Lorentz Group Network
arXiv:2006.04780

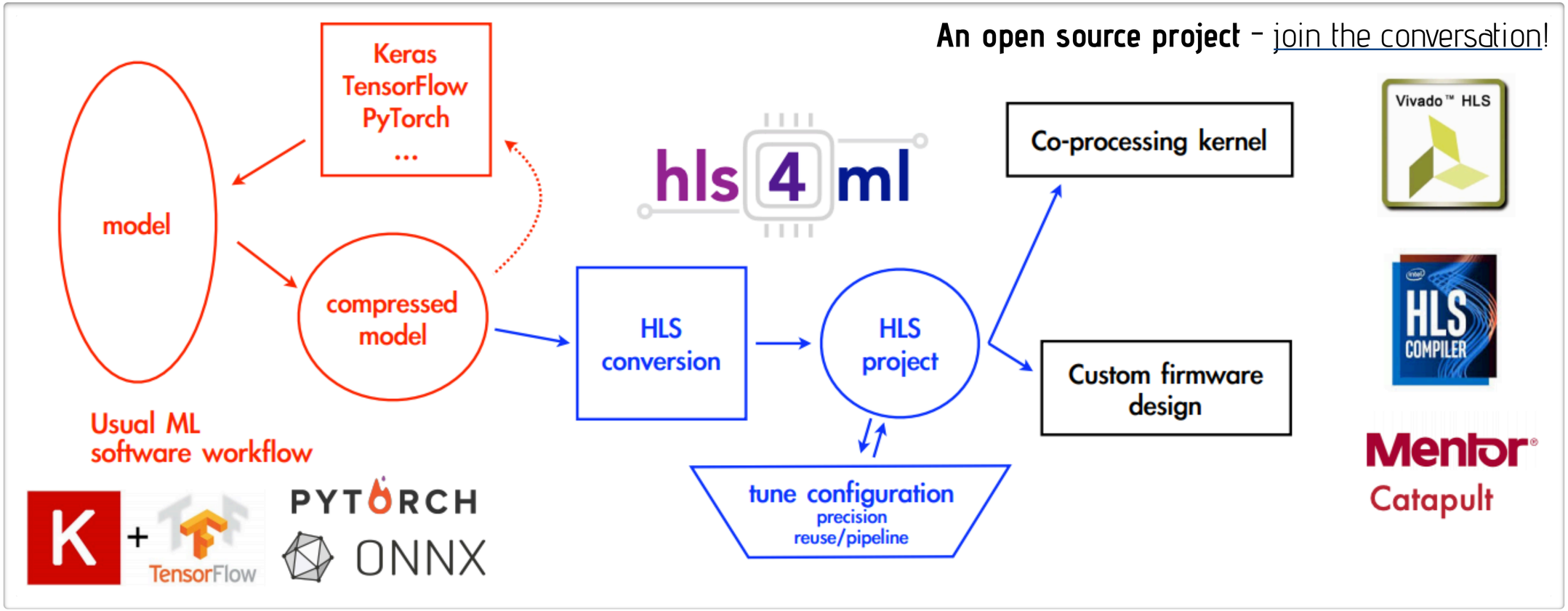
Architecture	Accuracy	AUC	$1/\epsilon_B$	#Param
ParticleNet	0.938	0.985	1298 ± 46	498k
P-CNN	0.930	0.980	732 ± 24	348k
ResNeXt	0.936	0.984	1122 ± 47	1.46M
EFP	0.932	0.980	384	1k
EFN	0.927	0.979	633 ± 31	82k
PFN	0.932	0.982	891 ± 18	82k
TopoDNN	0.916	0.972	295 ± 5	59k
LGN	0.929 $\pm$ .001	0.964 $\pm$ 0.018	435 ± 95	4.5k

Potentially HUGE gains in efficiency from more intelligent network design incorporating domain knowledge

Energy flow polynomial neural network

Lorentz group equivariant neural network

hls4ml: accelerating ML on hardware



Originally designed for LHC triggers applications but broad and growing user base
Example: ML training (qkeras/brevitas), backend devices (Xilinx, Intel, Microsemi, ASIC), network architectures (Conv, RNN/LSTM, Graph), and applications (...)

hls4ml: accelerating ML on hardware

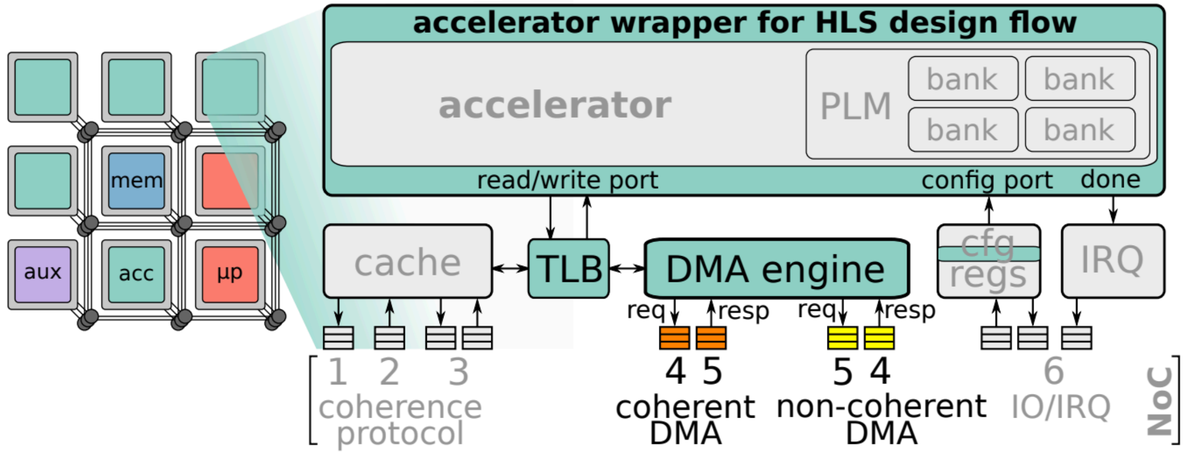
fastmachinelearning.org/hls4ml

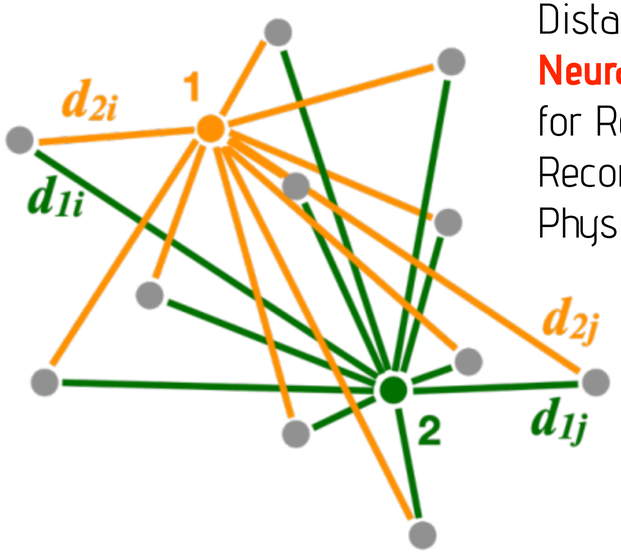
Join us for the tutorial on Thursday!

Originally designed for LHC triggers applications but broad and growing user base
Example: ML training (qkeras/brevitas), backend devices (Xilinx, Intel, Microsemi, ASIC), network architectures (Conv, RNN/LSTM, Graph), and applications (...)

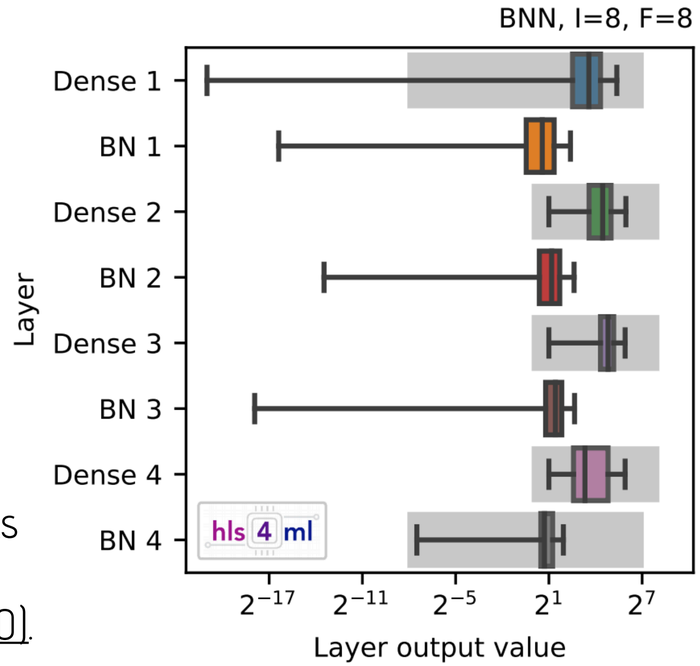
More hls4ml

ESP4ML: Platform-Based Design of Systems-on-Chip for Embedded Machine Learning, DATE Conference 2020 .

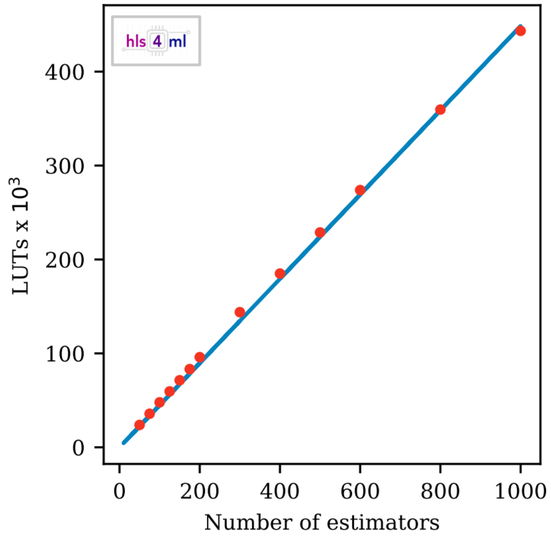




Distance-Weighted **Graph Neural Networks** on FPGAs for Real-Time Particle Reconstruction in High Energy Physics, arXiv:2008.03601



Compressing deep neural networks on FPGAs to **binary and ternary** precision with hls4ml, MLST (2020).

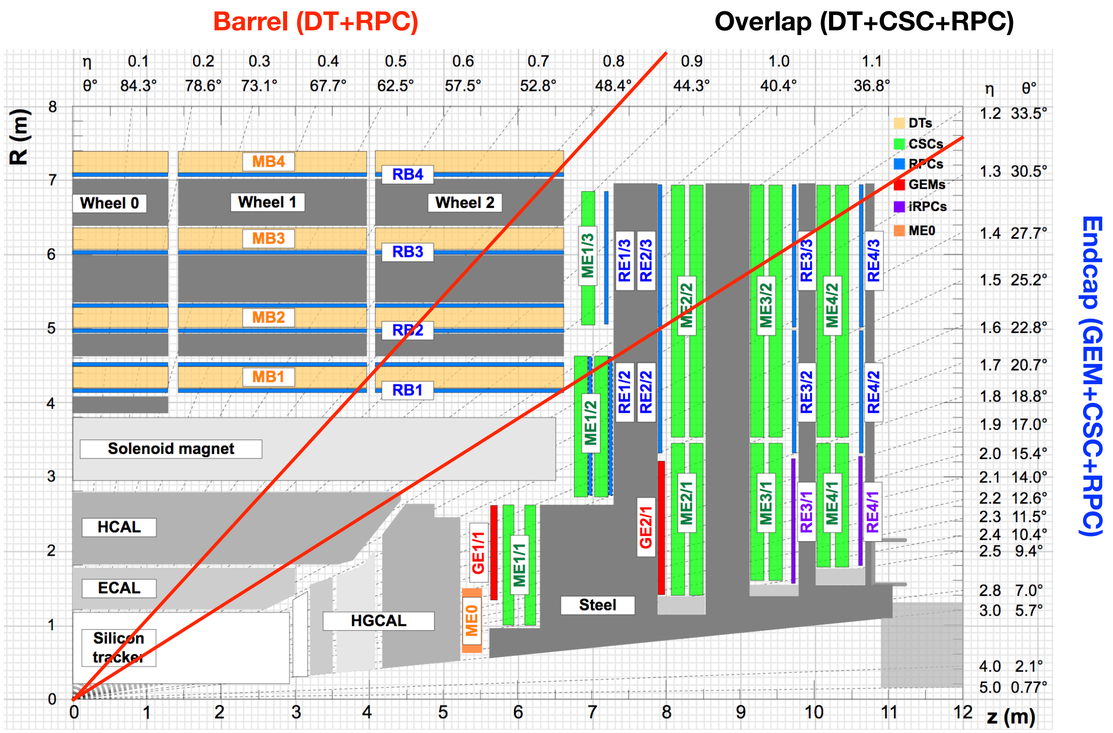


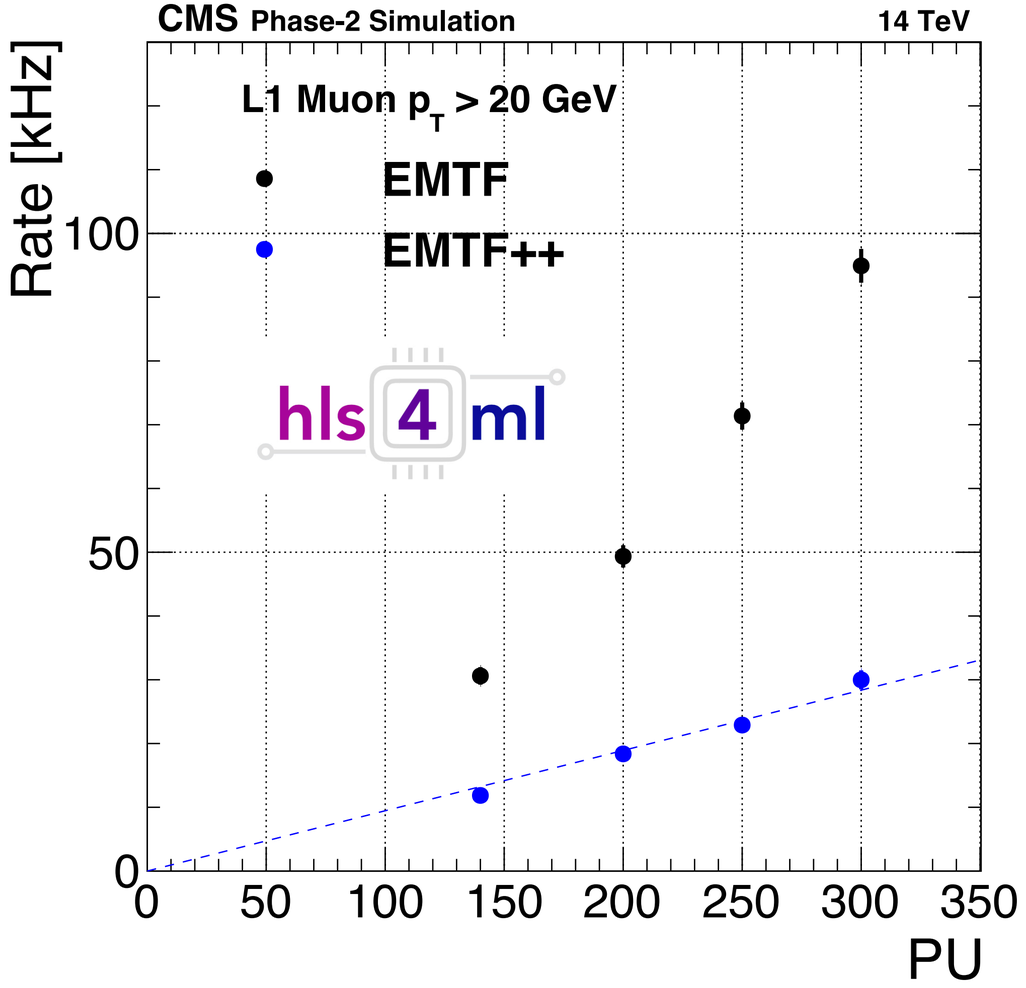
Fast inference of **Boosted Decision Trees** in FPGAs for particle physics, JINST 15, P05026 (2020).

Example applications

CMS Trigger TDR

Case study: muon trigger upgrade





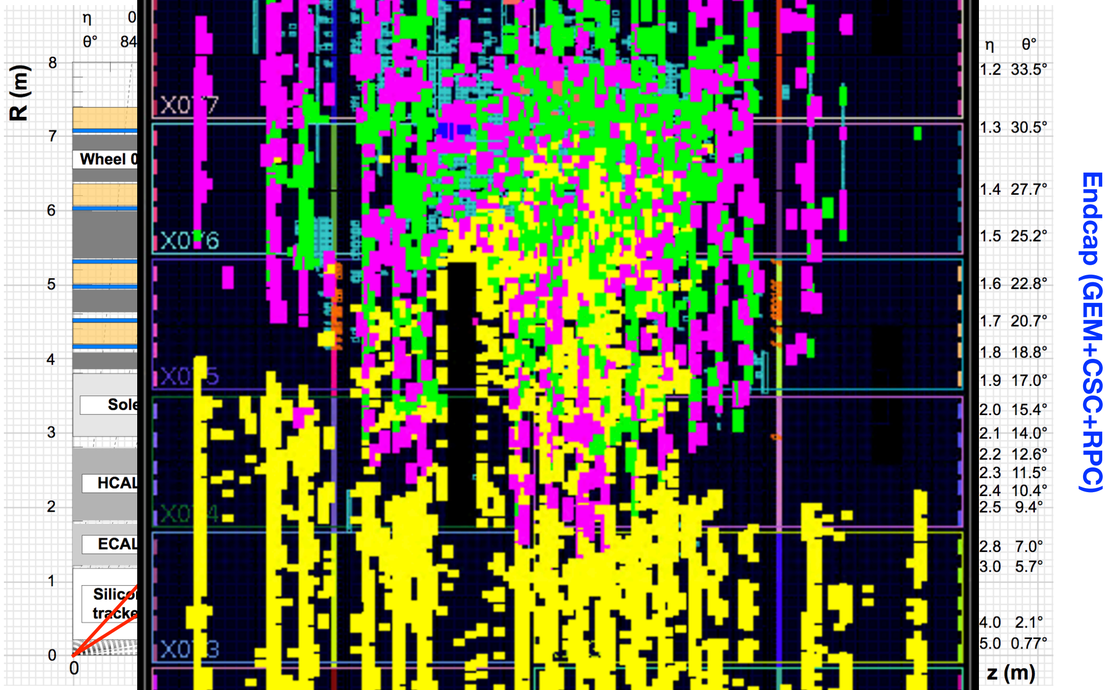
EMTF = BDT (external memory)

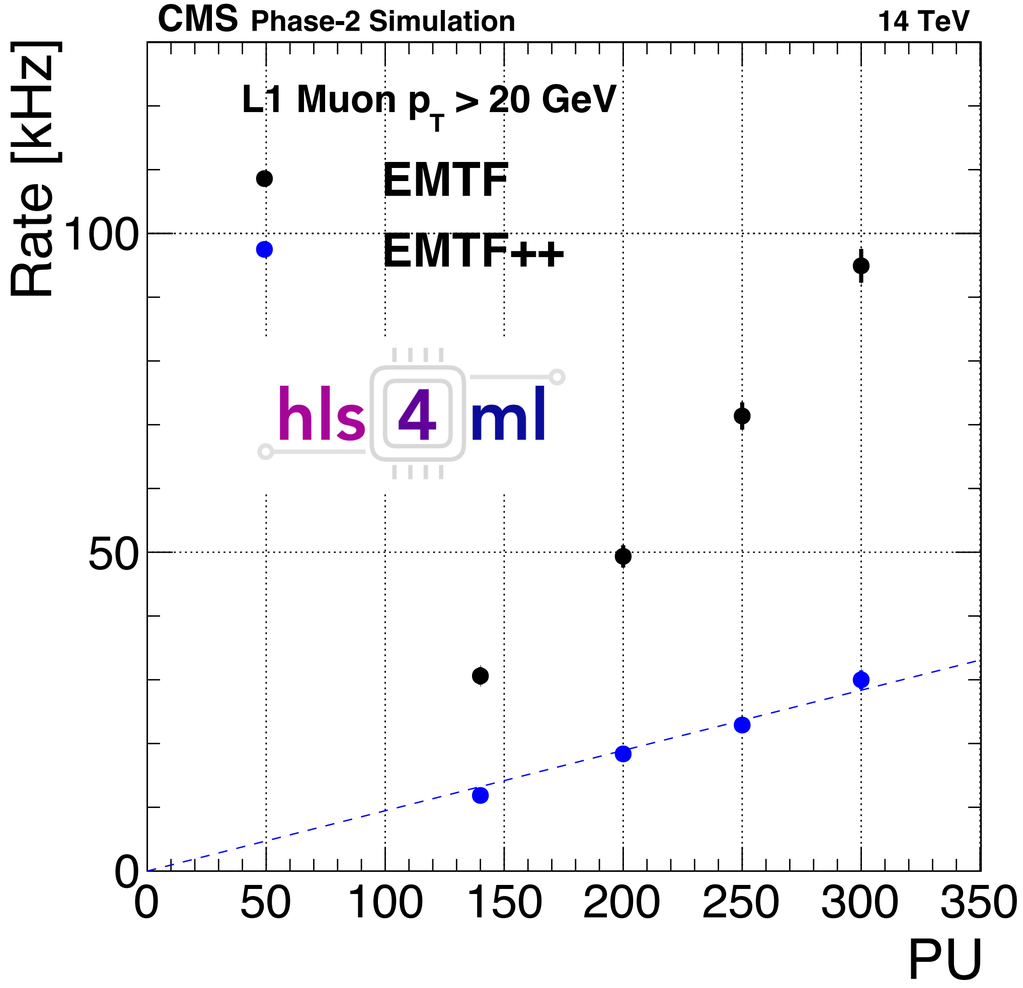
EMTF++ = NN

~3x reduction in the trigger rate for neural network!

Cas trigger upgrade

CMS Trigger TDR





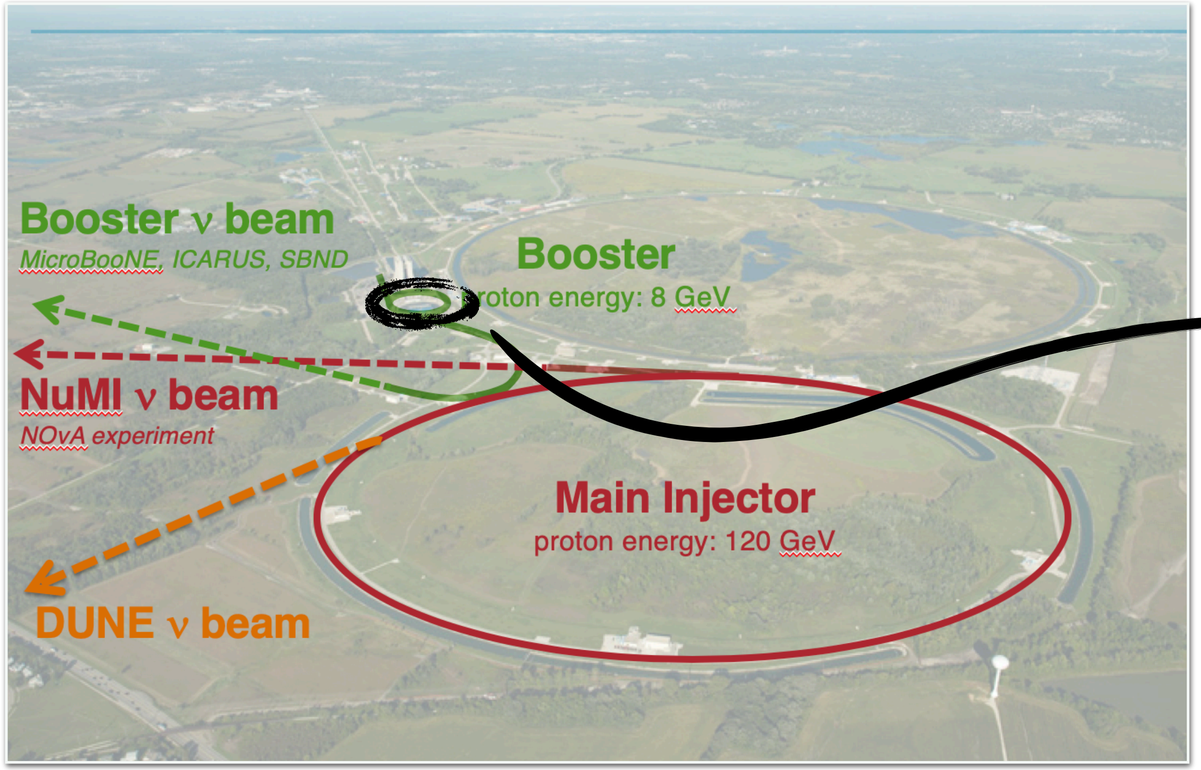
EMTF

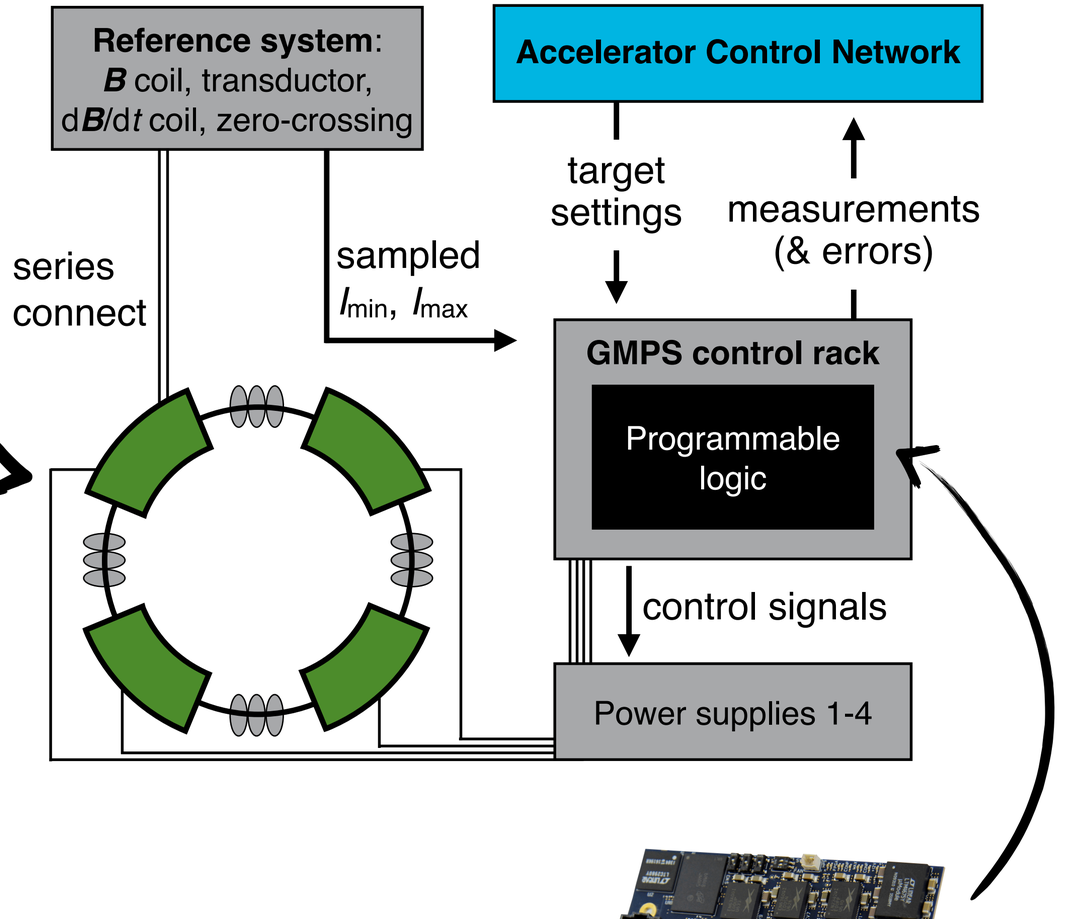
EMTF

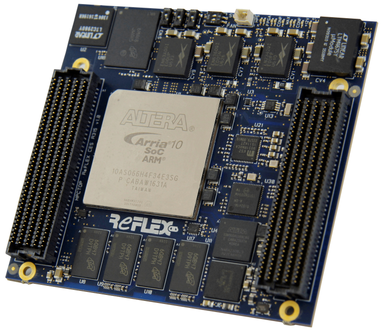
~3x r neural network!

Accelerator controls

FNAL, PNNL, UCSD collaboration
arXiv: 2010:YYYYY







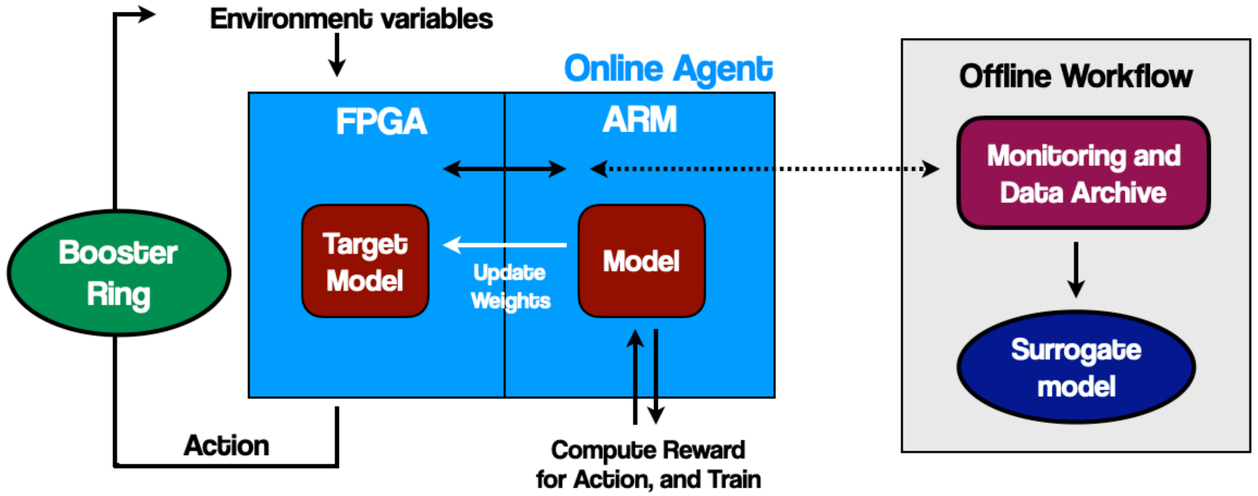
Arria10 SoM

The Task:

Control Gradient Magnet Power Supplies for Booster Ring at 15 Hz to reduce beam losses using reinforcement learning setup

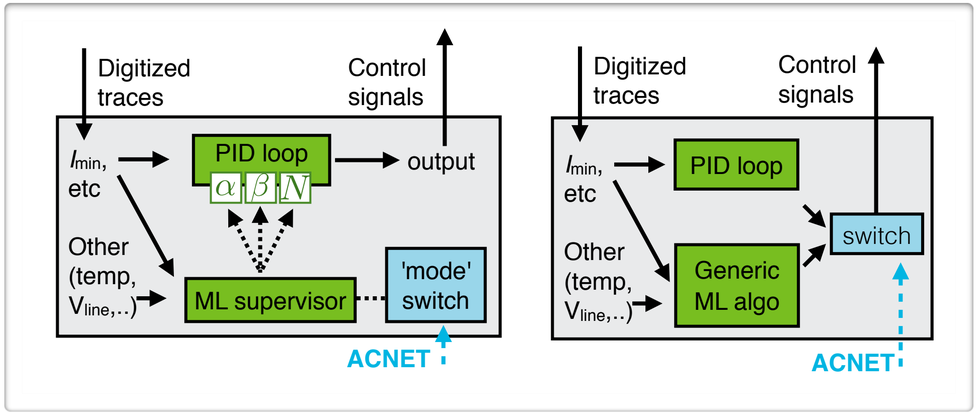
RL setup

FNAL, PNNL, UCSD collaboration
arXiv: 2010:YYYYY

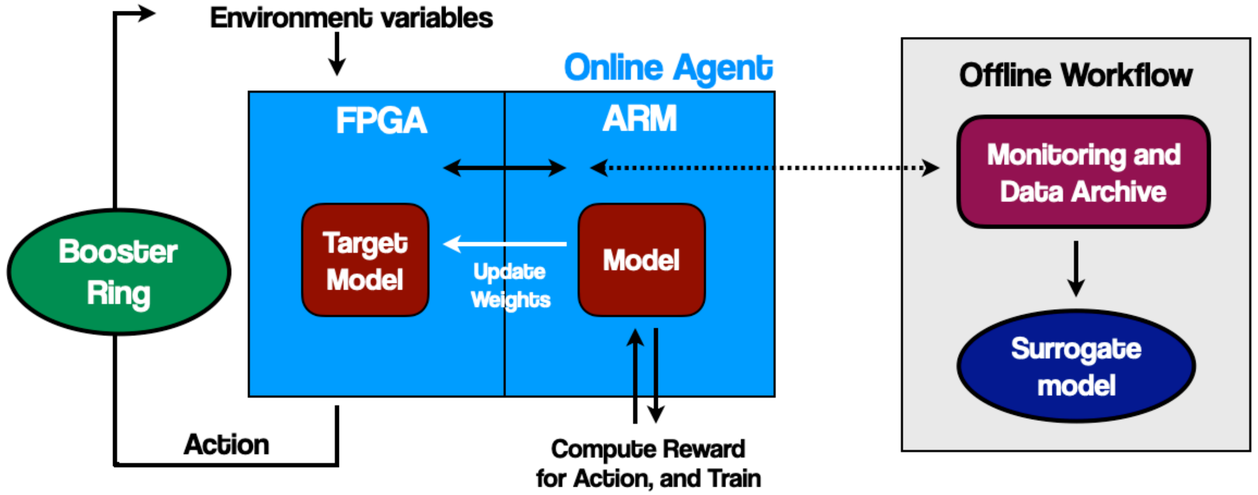


RL setup

FNAL, PNNL, UCSD collaboration
arXiv: 2010:YYYYY

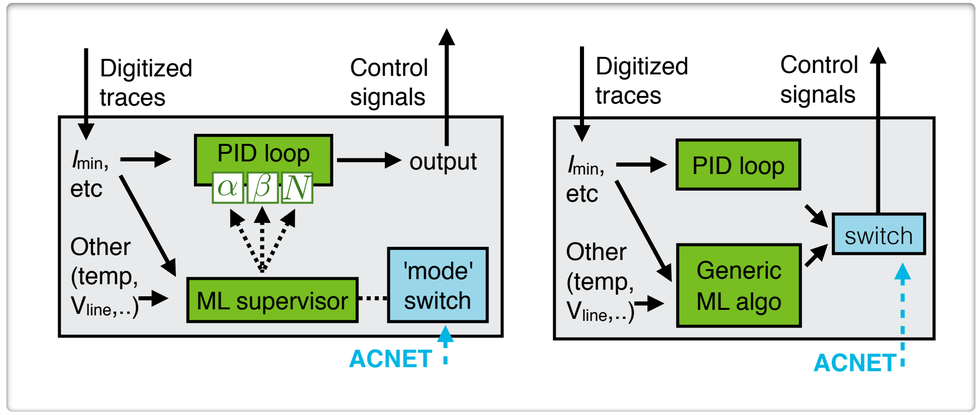


For safe-guarding online algorithm, start with model-based RL; plan to compare against model-free online agent

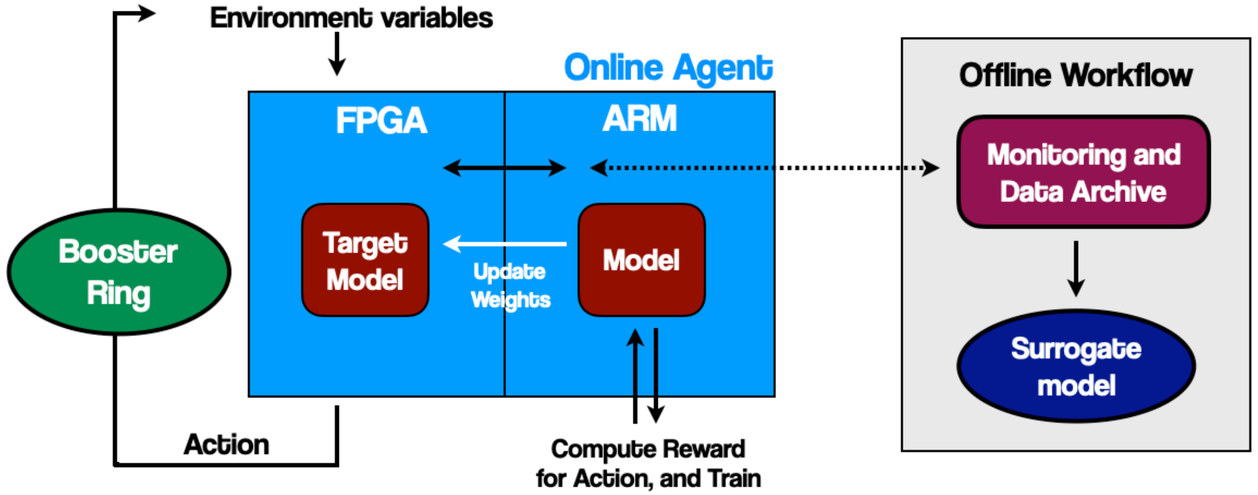


RL setup

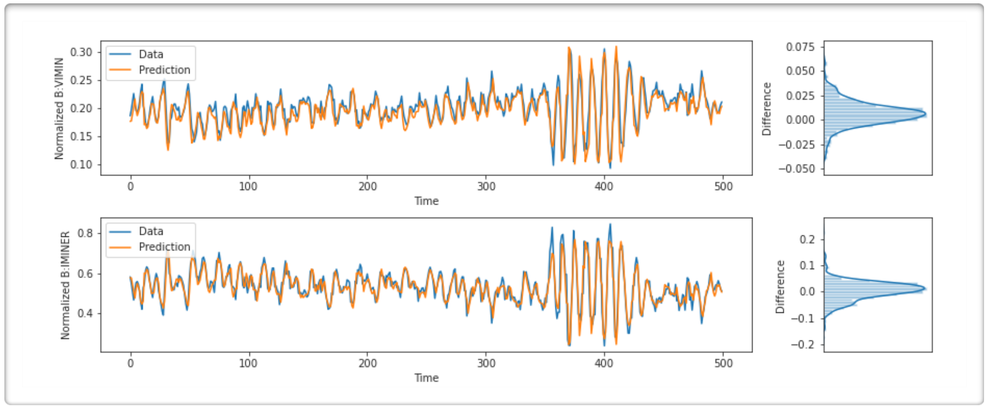
FNAL, PNNL, UCSD collaboration
arXiv: 2010:YYYYY



For safe-guarding online algorithm, start with model-based RL; plan to compare against model-free online agent

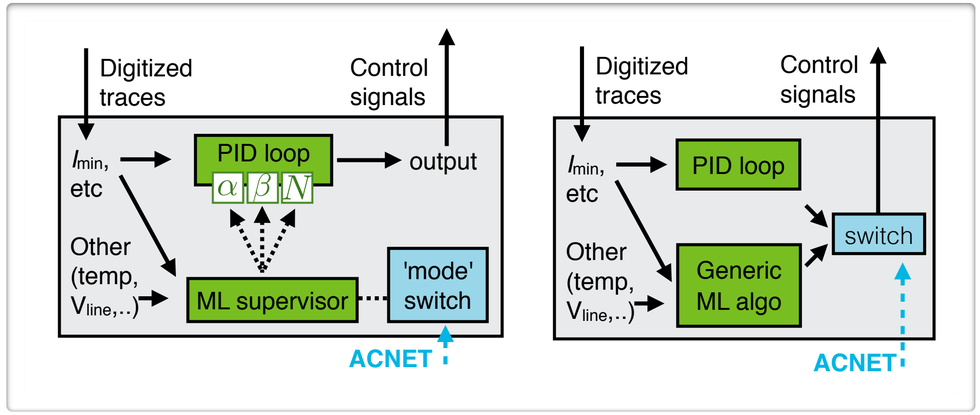


LSTM-based network using Deep Q-learning framework for surrogate model to mimic behavior of Booster

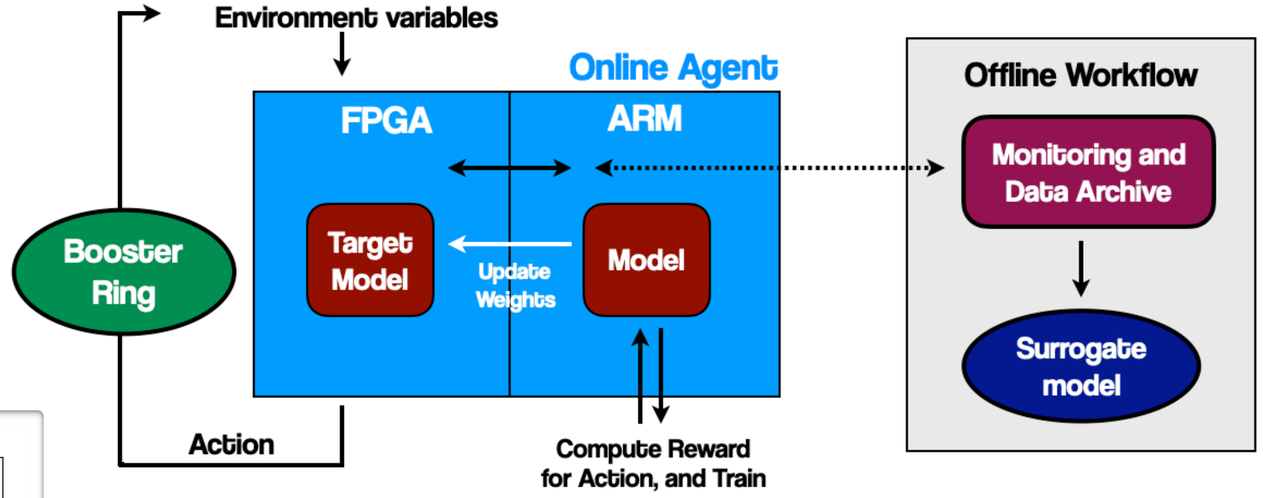


RL setup

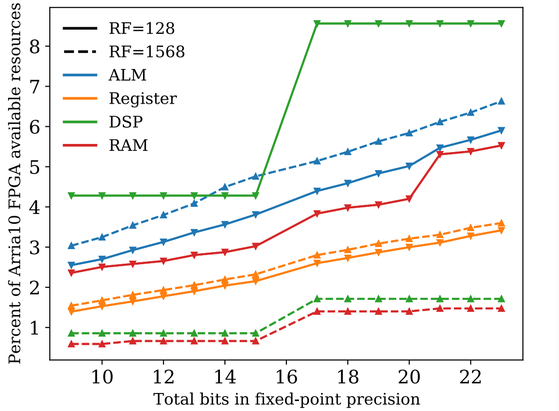
FNAL, PNNL, UCSD collaboration
arXiv: 2010:YYYYY



For safe-guarding online algorithm, start with model-based RL; plan to compare against model-free online agent

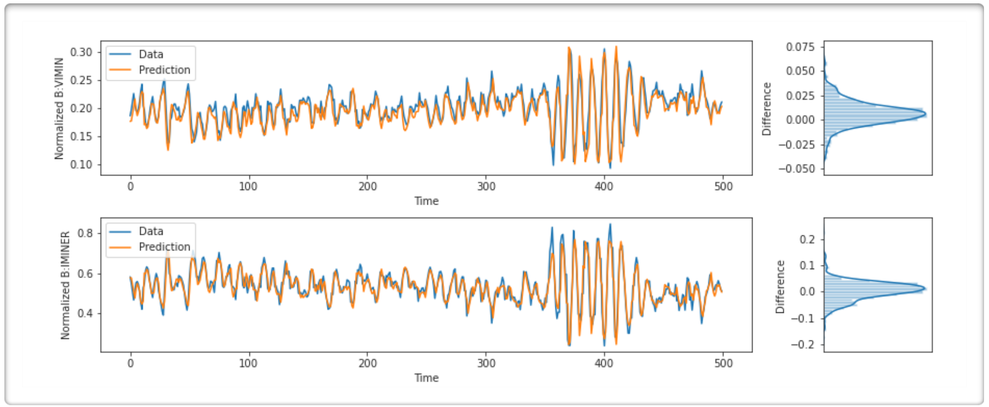


DNN-based ensemble model for RL online agent



Requires development of PL-PS interface for weight updating

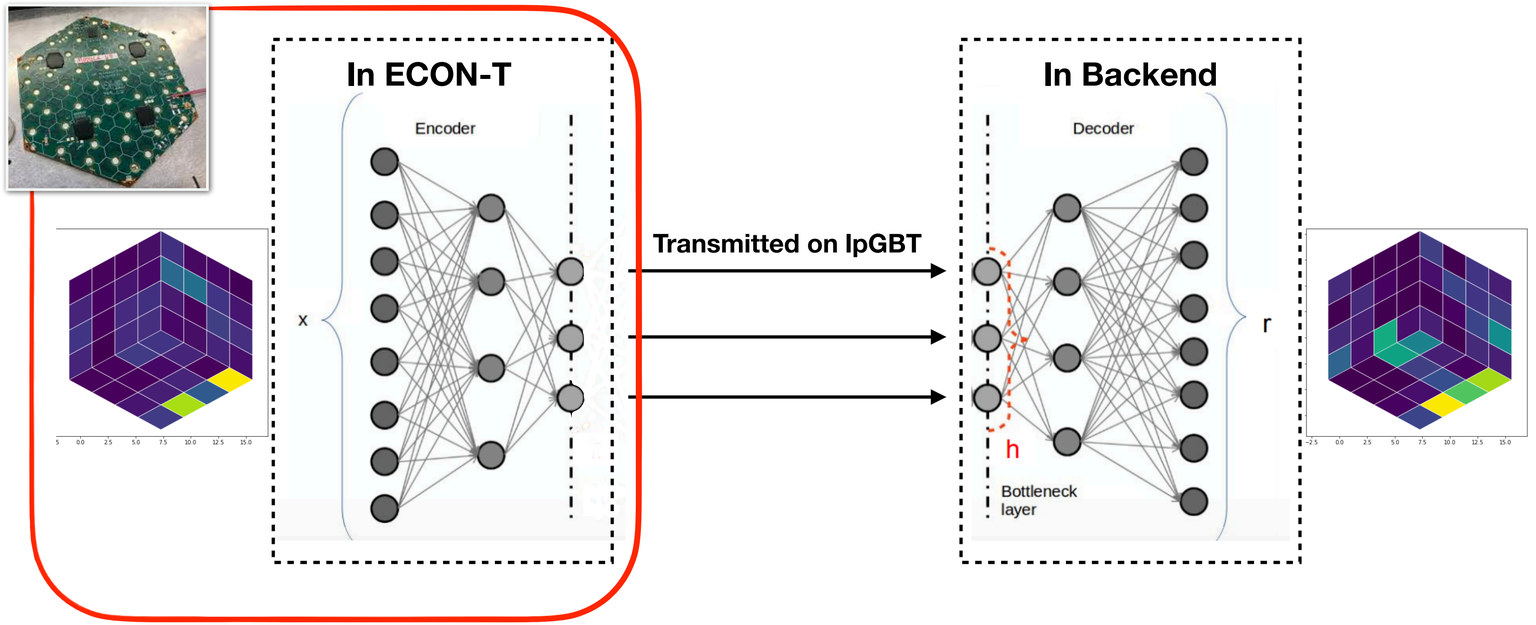
LSTM-based network using Deep Q-learning framework for surrogate model to mimic behavior of Booster



What about ASICs?

See poster here by Manuel Blanco Valentin
Talks at IEEE NSS (Christian Herwig, Farah Fahim)

- Putting a neural network on the detector front-ends for data compression
  - Application: **CMS HGCal ECON (concentrator) ASIC**
  - ASIC required due to radiation tolerance and power requirements
- **Fully reconfigurable ASIC** to address:
  future unknown unknowns' including **evolving LHC conditions** (pileup, beam bkgs), **detector performance** (noise, dead channels), **performance metrics** (resolution, substructure, new physics signatures)



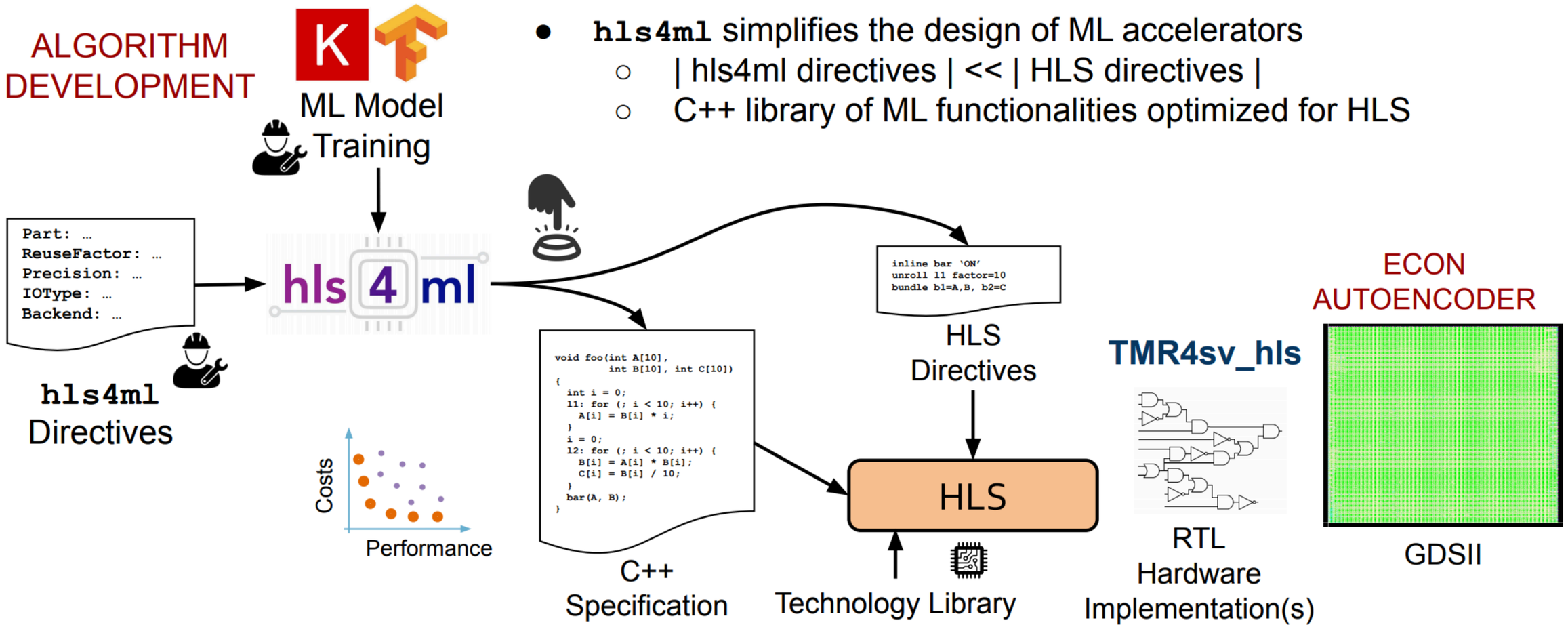
Auto-encoder design:
Conv2D + Dense

Performance metrics defined based on Energy Mover's Distance and physics resolutions/rates

Quantization aware training very important!

AS1C workflow

See poster here by Manuel Blanco Valentin
Talks at IEEE NSS (Christian Herwig, Farah Fahim)



First implementation of AI in an ASIC for particle physics

TSMC 65nm process: ~95 mW, 3.6mm^2, 25ns latency, 40M img/s, ~4000 ops, 200 MRad spec

Summary

Summary and outlook

Fast ML for Science Workshop
https://indico.cern.ch/e/fml2020
Join us! Nov 30-Dec 3

- Real-time intelligence can greatly accelerate science allowing us to test hypotheses faster, enhance performance of detectors/accelerators, and save potentially lost data

- I look forward to seeing exciting tools, techniques, and applications of real-time machine learning at this conference

- Early days towards smart sensors, autonomous detectors, and accelerated experimental loops