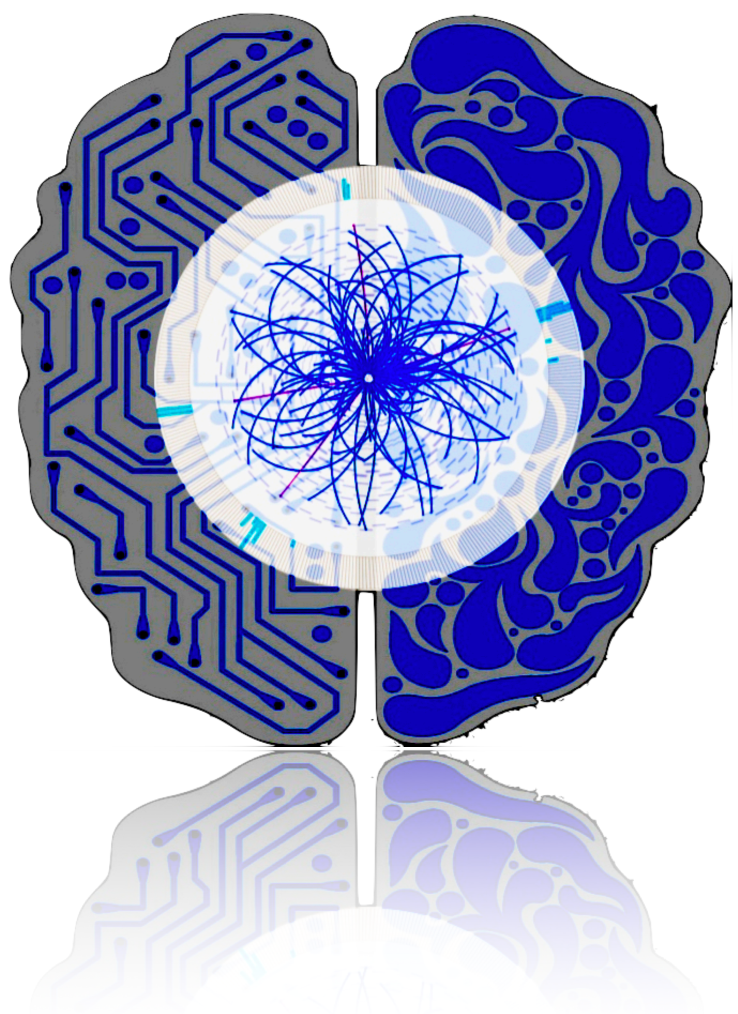
hls4ml enabling real-time deep learning in particle physics



Jennifer Ngadiuba (Fermilab)

on behalf of the hls4ml team

CPAD Instrumentation Frontier Workshop

Stony Brook University, March 18-22, 2021

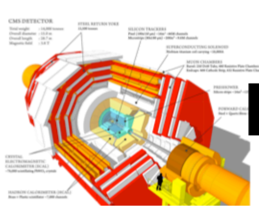
hls4ml @ the LHC

- **hls4ml** is a library for automatic translation of deep learning models to FPGA firmware for inference with ultra low latency

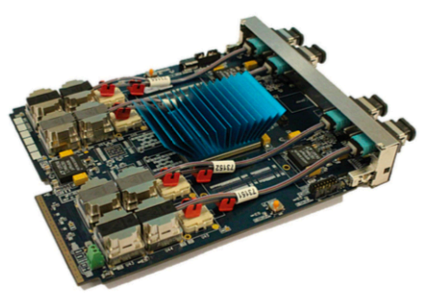
- **First target applications:**
  hardware trigger of LHC experiments and detector front-end electronics

J. Krupa's talk

Radiation Hard ASICs

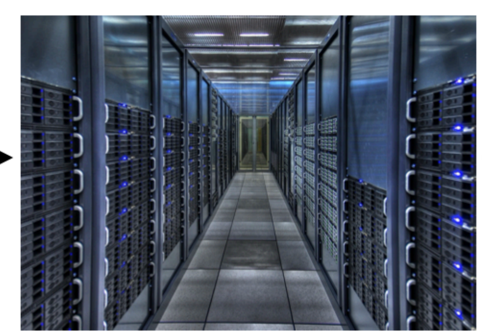


Level 1 Trigger



Fast
10 μs window
L1Trigger

High Level Trigger

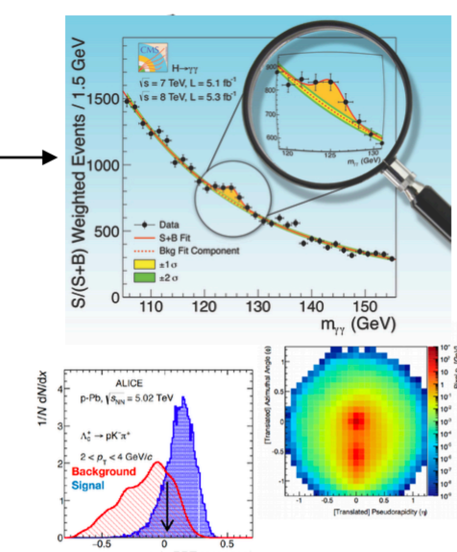


Intermediate
<500 ms window
High Level Trigger

Offline reconstruction

Slow
10 s window
Offline Cluster

Analysis

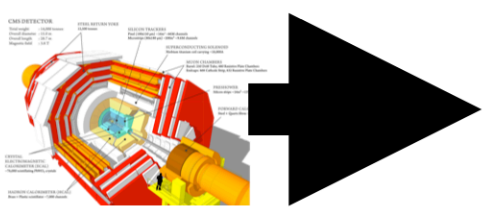


A 2-tier event filter reduces data rates by ~4 orders of magnitude

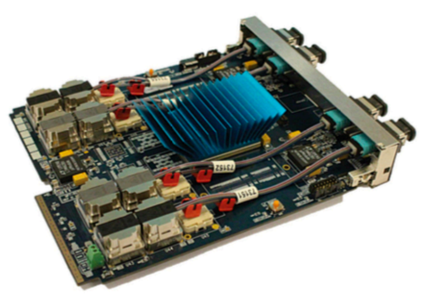
hls4ml @ the LHC

- **hls4ml** is a library for automatic translation of deep learning models to FPGA firmware for inference with ultra low latency
- **First target applications:**
  hardware trigger of LHC experiments and detector front-end electronics

Radiation Hard ASICs

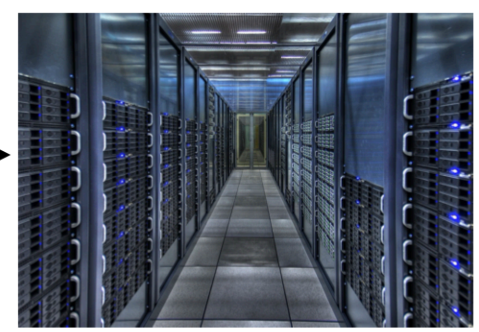


Level 1 Trigger



Fast
10 μs window
L1Trigger

High Level Trigger

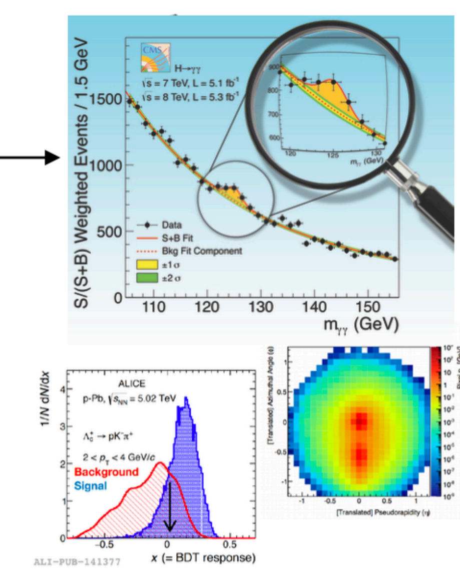


Intermediate
<500 ms window
High Level Trigger

Offline reconstruction

Slow
10 s window
Offline Cluster

Analysis

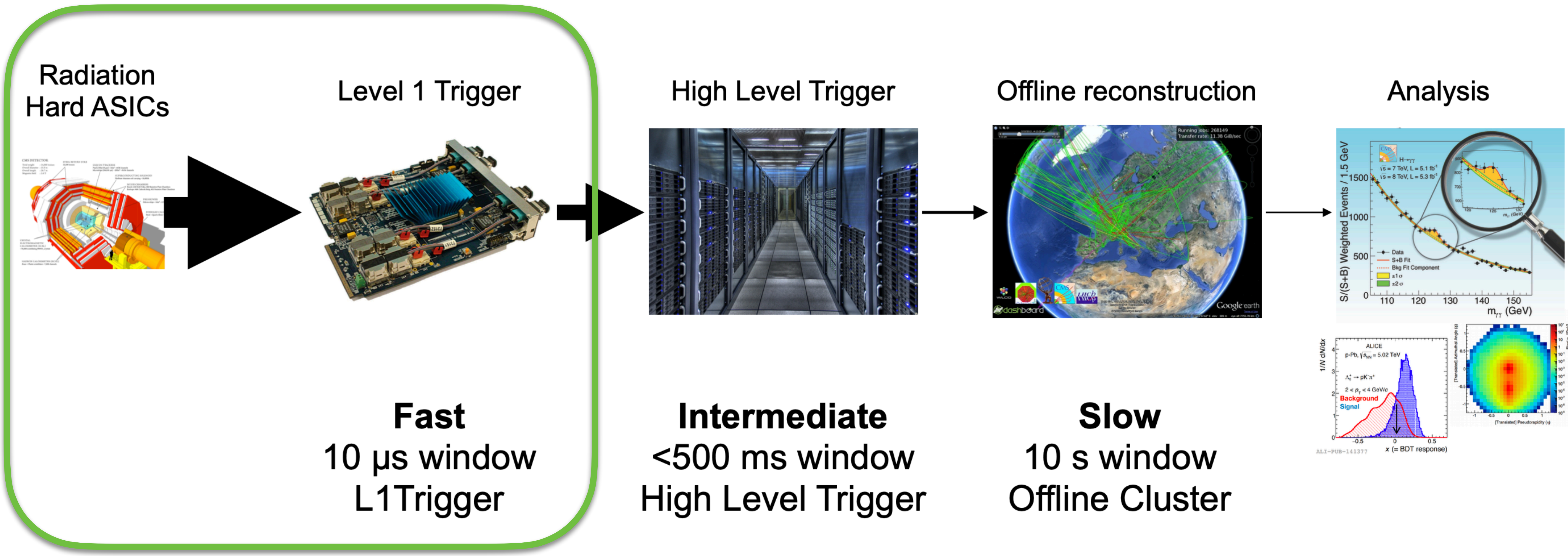


J. Krupa's talk: accelerate DL using co-processors (GPUs or FPGAs)

hls4ml @ the LHC

- **hls4ml** is a library for automatic translation of deep learning models to FPGA firmware for inference with ultra low latency

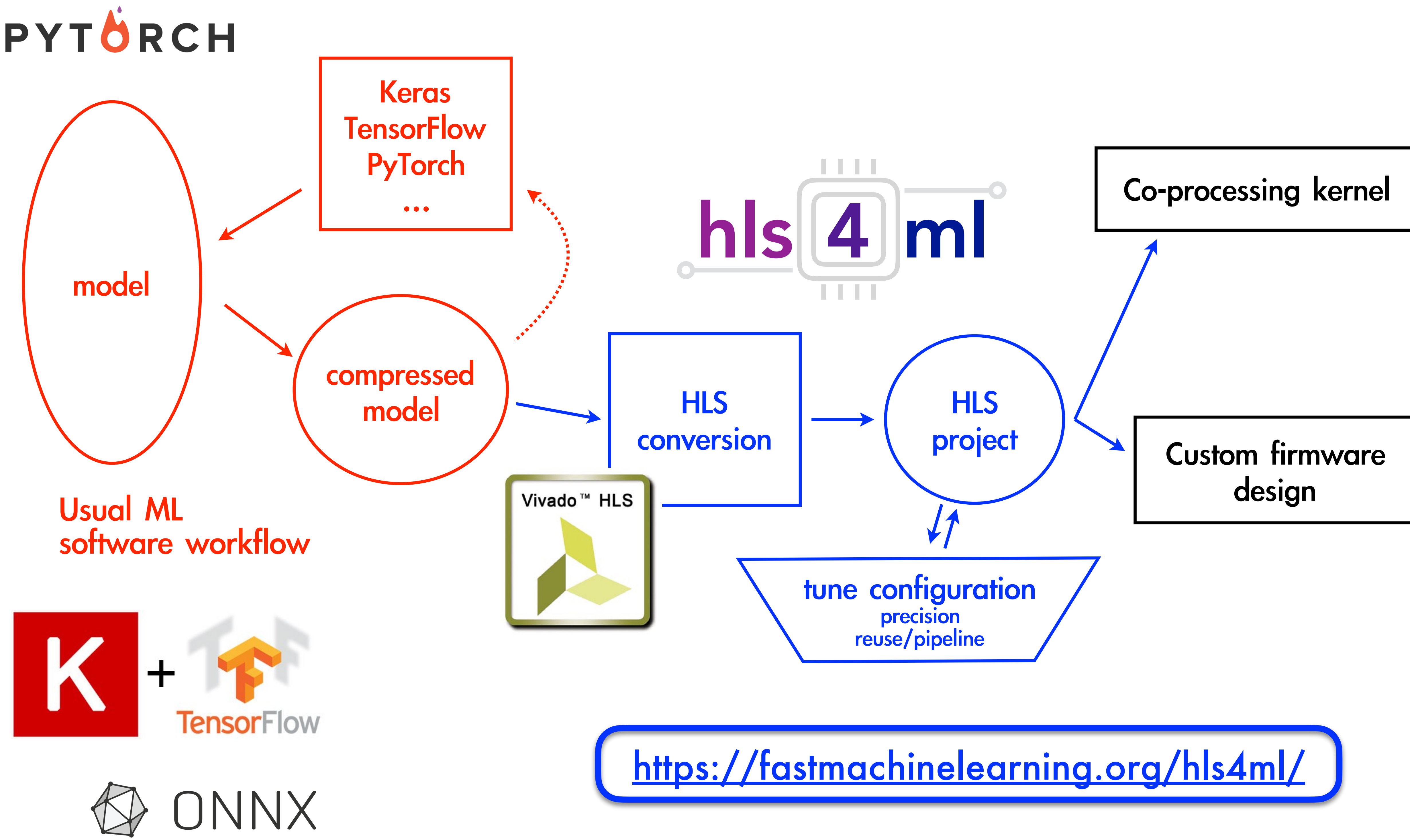
- **First target applications:**
  hardware trigger of LHC experiments and detector front-end electronics



THIS TALK! Limited resources and strict latency constraints

Bring DL to FPGA for L1 trigger with

high level synthesis for machine learning



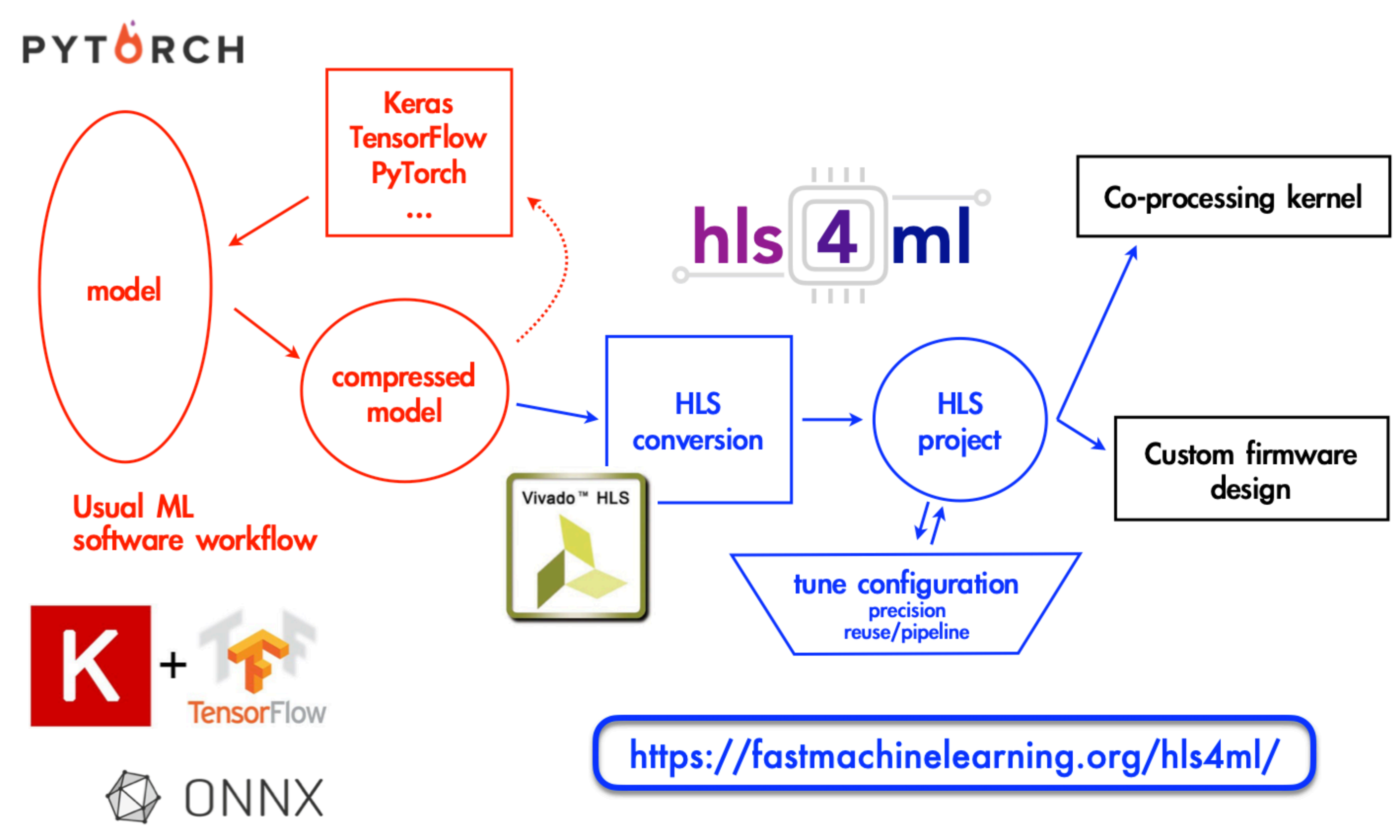
Bring DL to FPGA for L1 trigger with

high level synthesis for machine learning

- User-friendly automated tool
- Easy to tune the inference performance for your specific application: *precision, resource vs latency/throughput tradeoff*

- Can be used as API
- Includes several debugging utilities

- Most common DL layers and activation functions supported



hls4ml: recent developments

NEW!

- Since CPAD19 but the library has been significantly expanded!

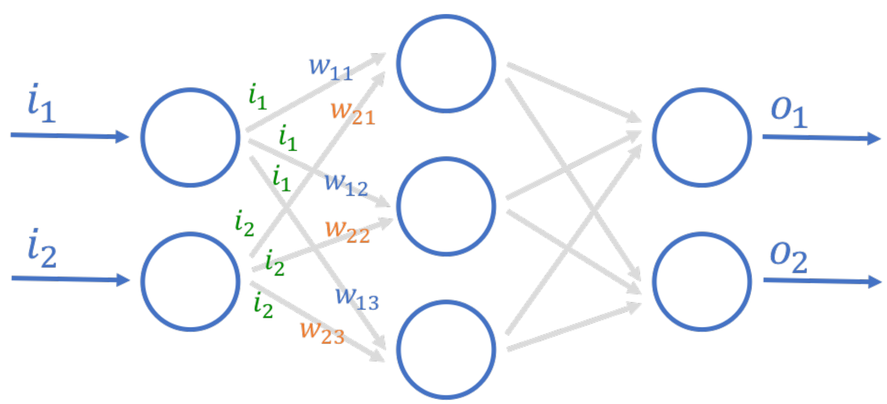
- **Quantization-aware training and pruning**
  - Google QKeras [arxiv.2006.10159] THIS TALK!
  - PyTorch Brevitas [arxiv.2102.11289] → coming soon
- **Convolutional neural networks** [arxiv.2101.05108]
- **Custom architectures as graph neural networks:**
  - GarNet/GravNet for calorimeter reconstruction [arXiv: 2008.03601]
  - Interaction networks for tracking [arxiv.2012.01563]

THIS TALK!

- **Workflow for DL-dedicated ASICs** [arxiv.2103.05579] → J. Hirschauer talk
- **Support for other vendors: Intel and Mentor HLS** → coming soon

Neural network inference on FPGA

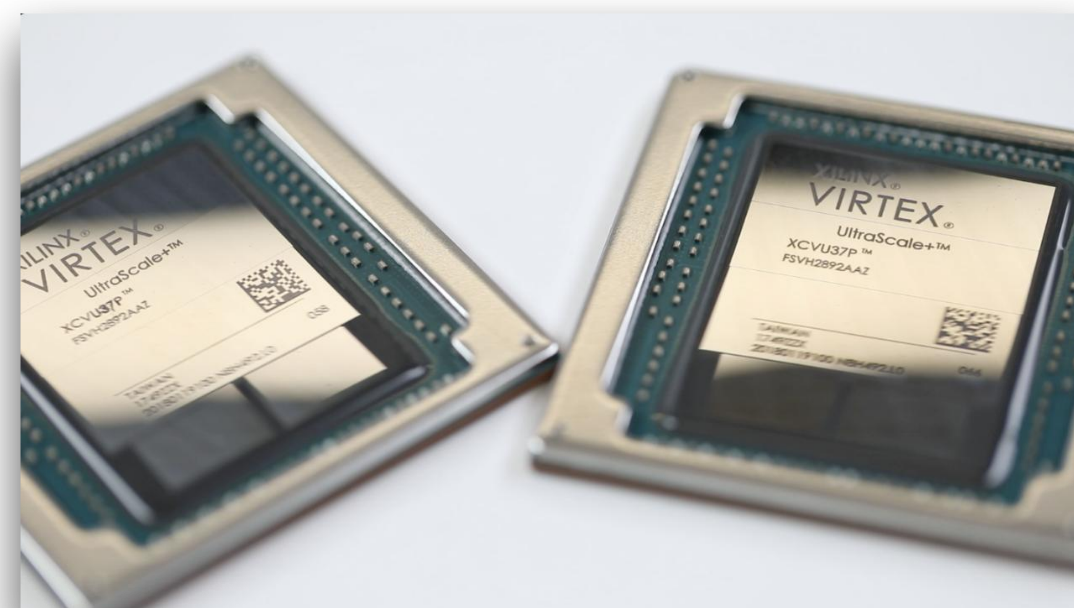
Neural network inference
=
matrix multiplication



$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11}\times i_1) + (w_{21}\times i_2) \\ (w_{12}\times i_1) + (w_{22}\times i_2) \\ (w_{13}\times i_1) + (w_{23}\times i_2) \end{bmatrix}$$

Efficient implementation on FPGA uses
DIGITAL SIGNAL PROCESSORS

There are about 5–10k DSPs in modern FPGAs!

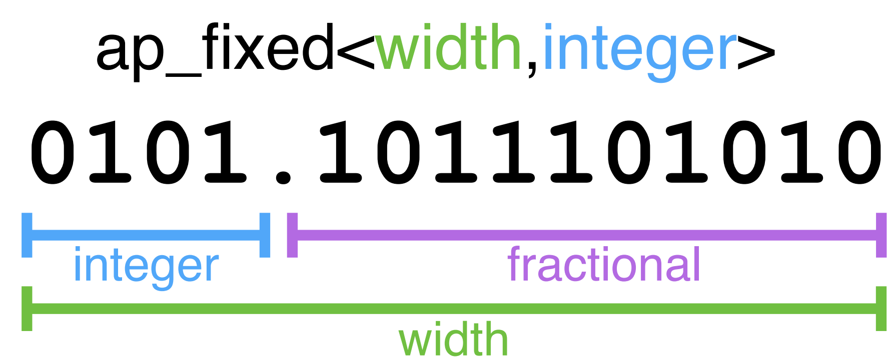


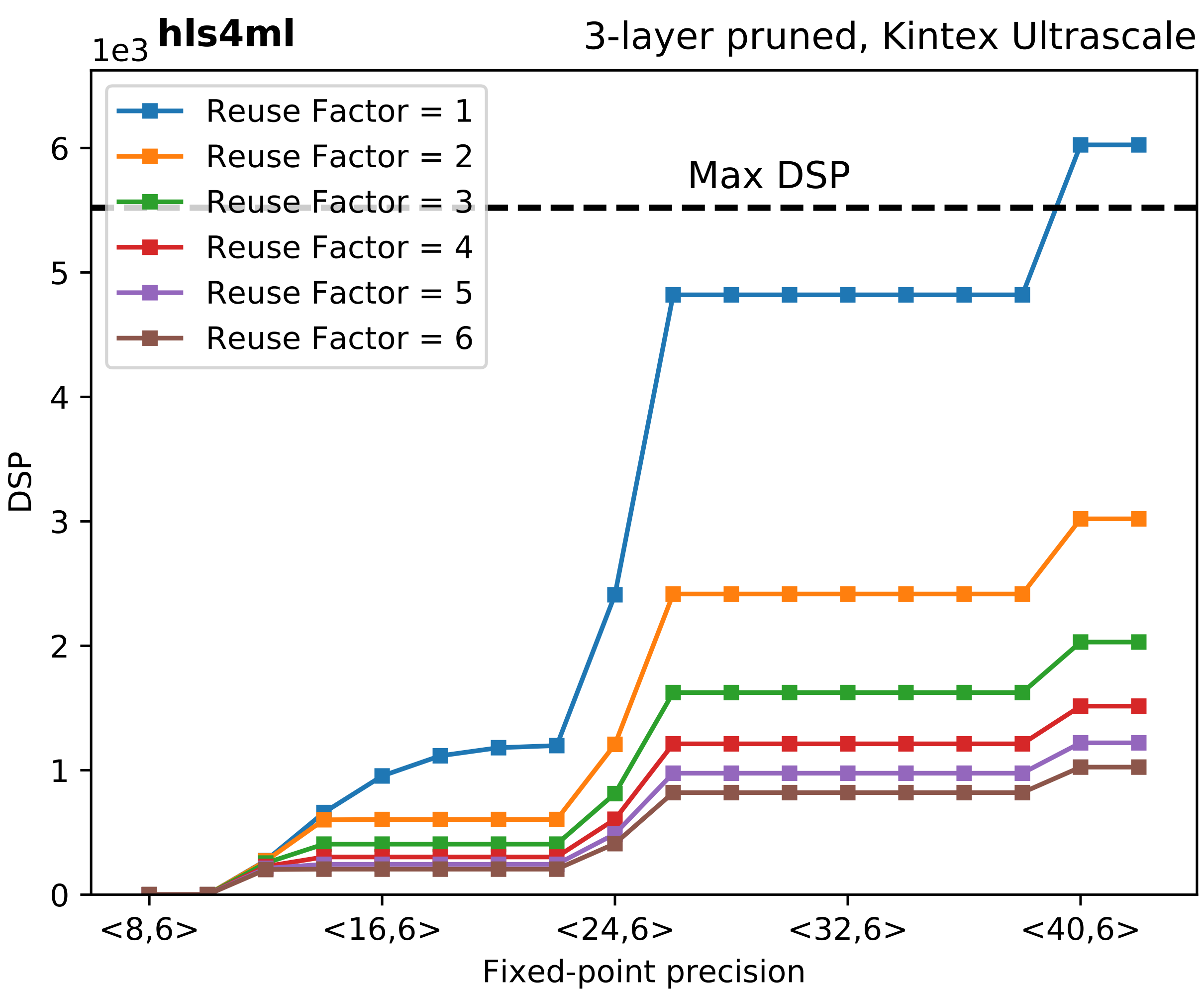
- DSPs are the most precious resource when mapping a NN into FPGA!
- Usage can be controlled in hls4ml by tuning how much to parallelize

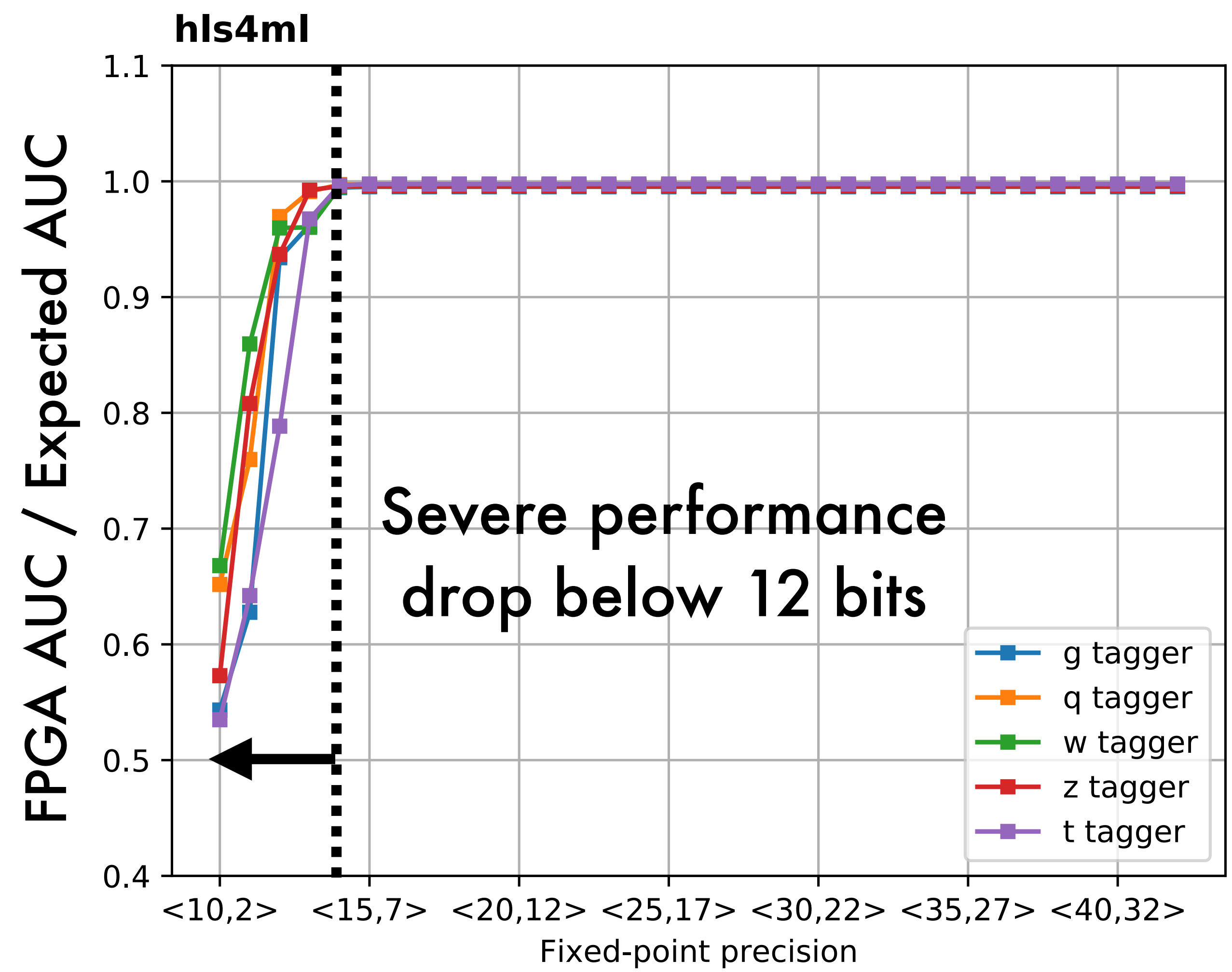
→ this affects the latency and it's a trade off that depends on the application

Efficient NN design: quantization

- Post-training quantization on FPGA allows for large area reduction but severe model performance drop for too few bits



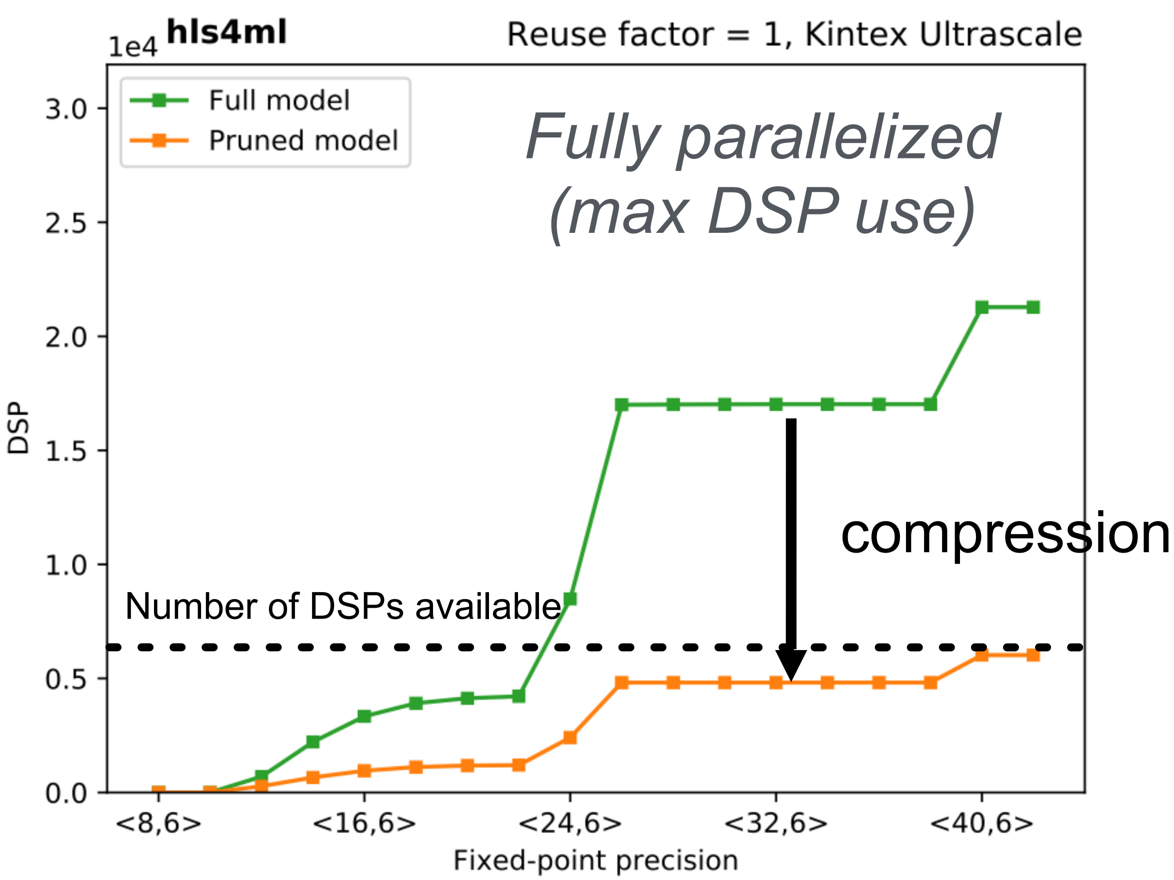


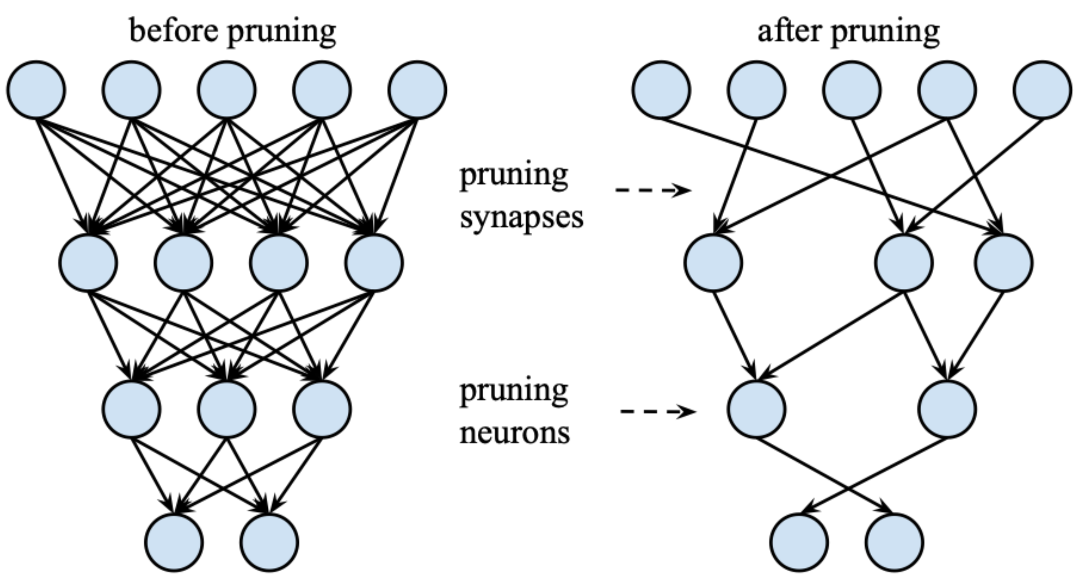


Efficient NN design: compression

- Neural Network compression is a widespread technique to reduce the size, energy consumption, and overtraining of deep neural networks

- Several approaches in literature [arxiv.1510.00149, arxiv.1712.01312, arxiv.1405.3866, arxiv.1602.07576, doi:10.1145/1150402.1150464]





eg, <u>tensorflow sparsity toolkit</u>
iteratively remove low magnitude weights, starting with 0 sparsity, smoothly increasing up to the set target as training proceeds

Efficient NN design with QKeras

NEW!

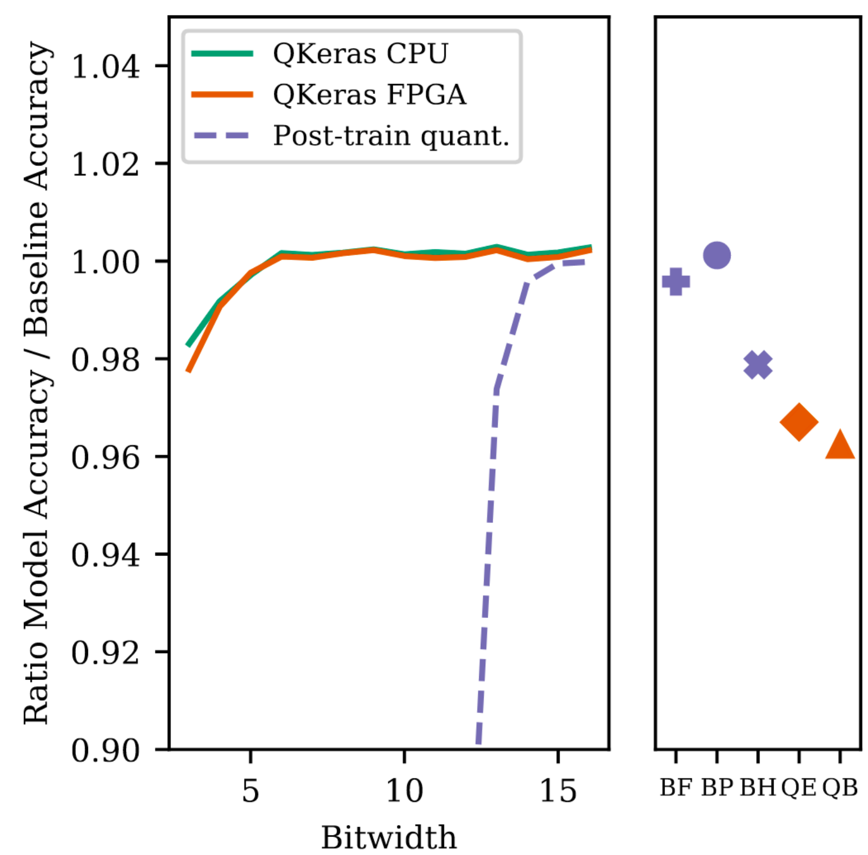
- QKeras is a library developed and maintained by Google to train models with quantization in the training

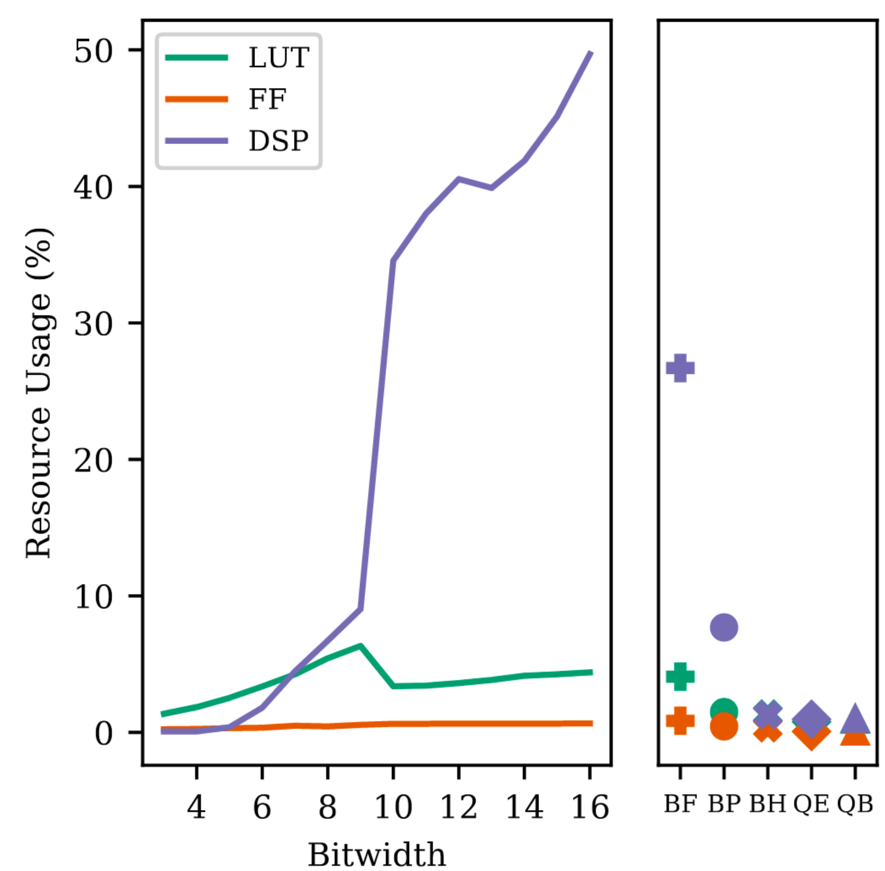
- Can achieve good performance with very few bits

- We've recently added support for QKeras-trained models to hls4ml [arxiv.2006.10159]

  - the number of bits used in training is also used in inference

  - automatic heterogenous layer-by-layer quantization also possible



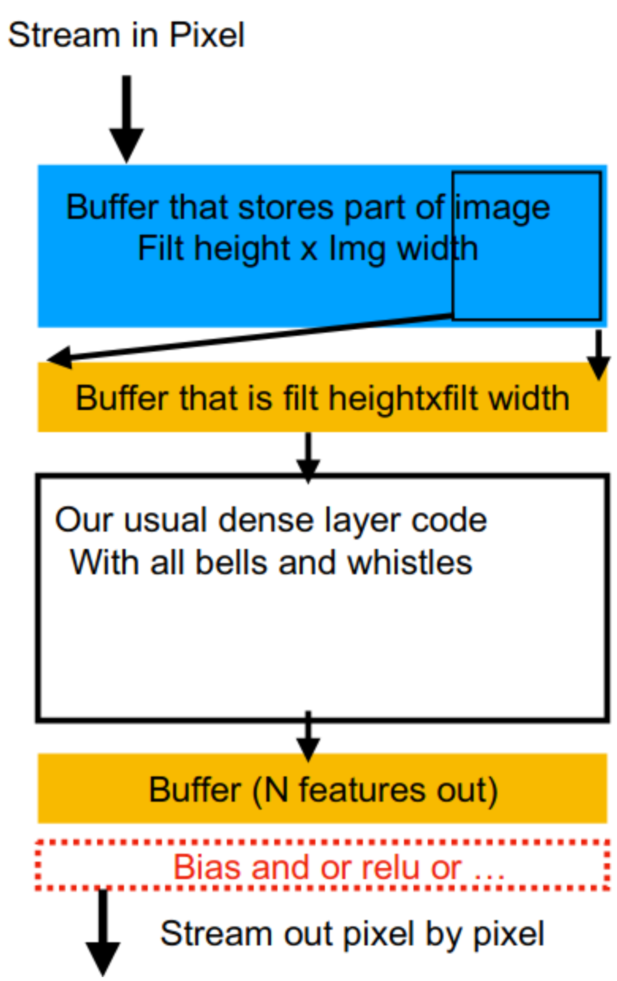


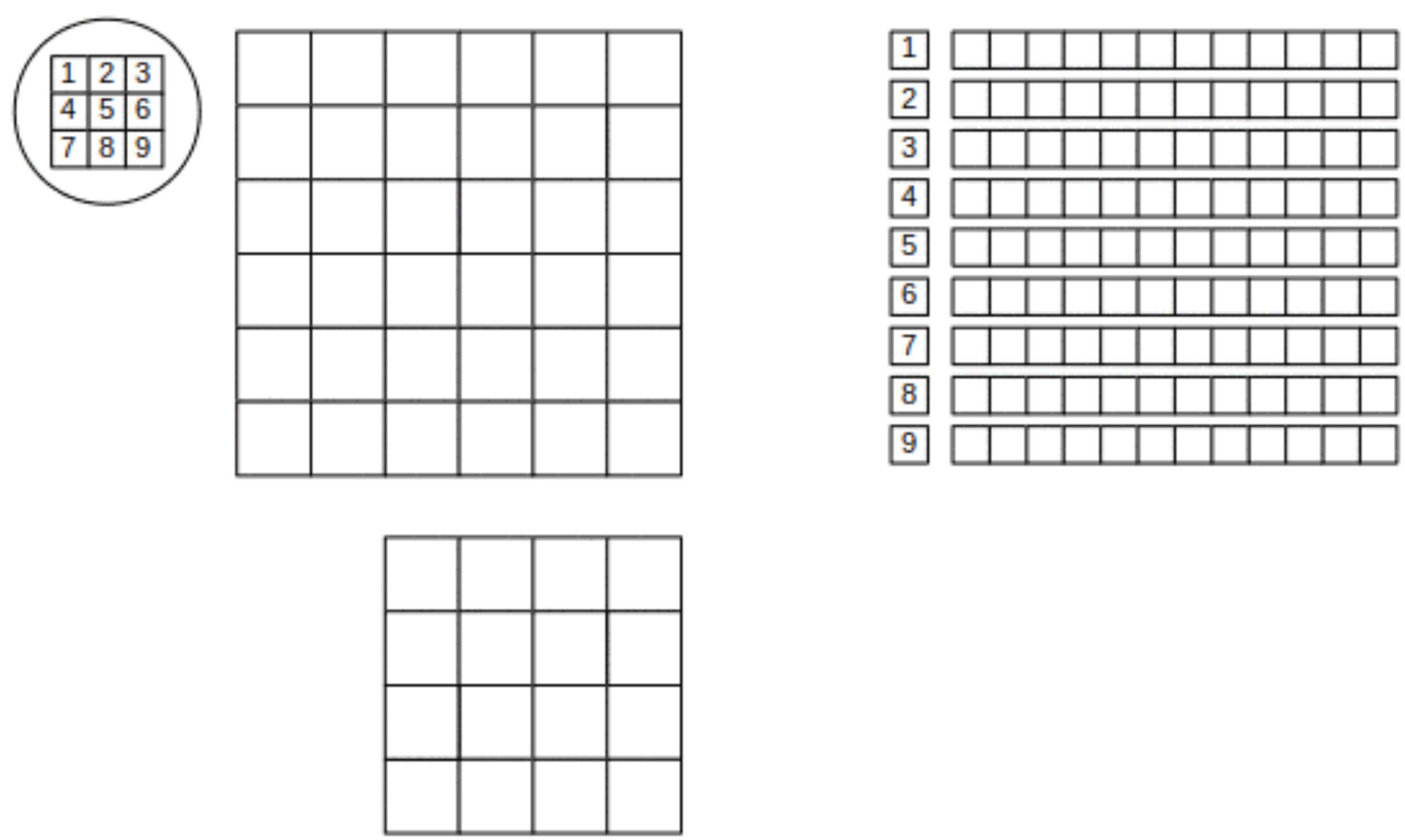
Fast convolutional neural networks

- Brand new implementation based on streaming `hls::stream<T>`

[arxiv.2101.05108](https://arxiv.org/abs/2101.05108)

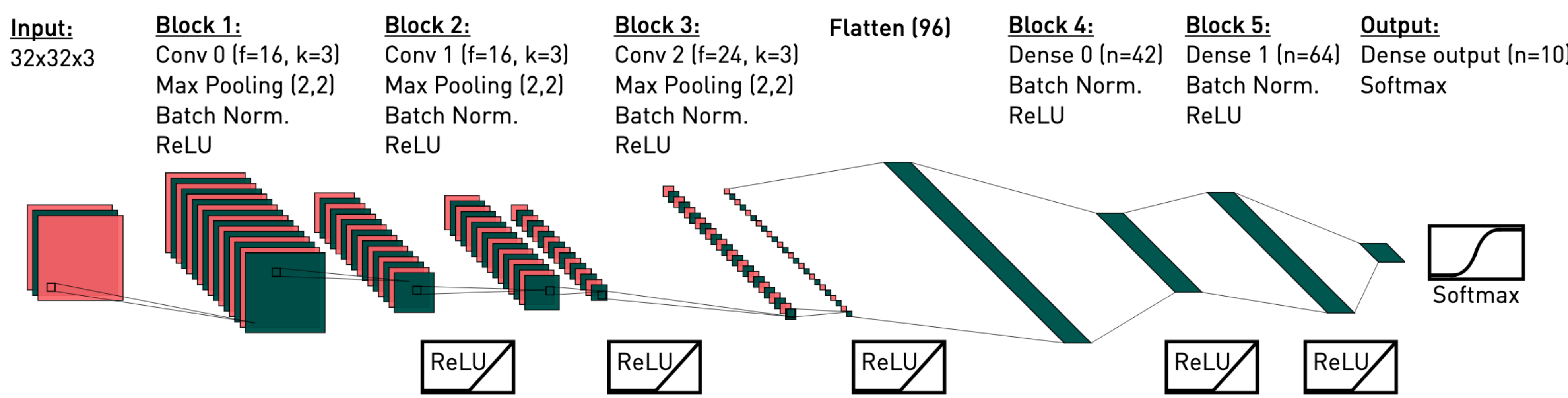
  - collect data from input pixels until we can compute one output (FIFOs)
  - compute the value of output pixel with a single call to matrix-vector multiplication
  - can reuse existing matrix-vector multiplication used for fully connected layers



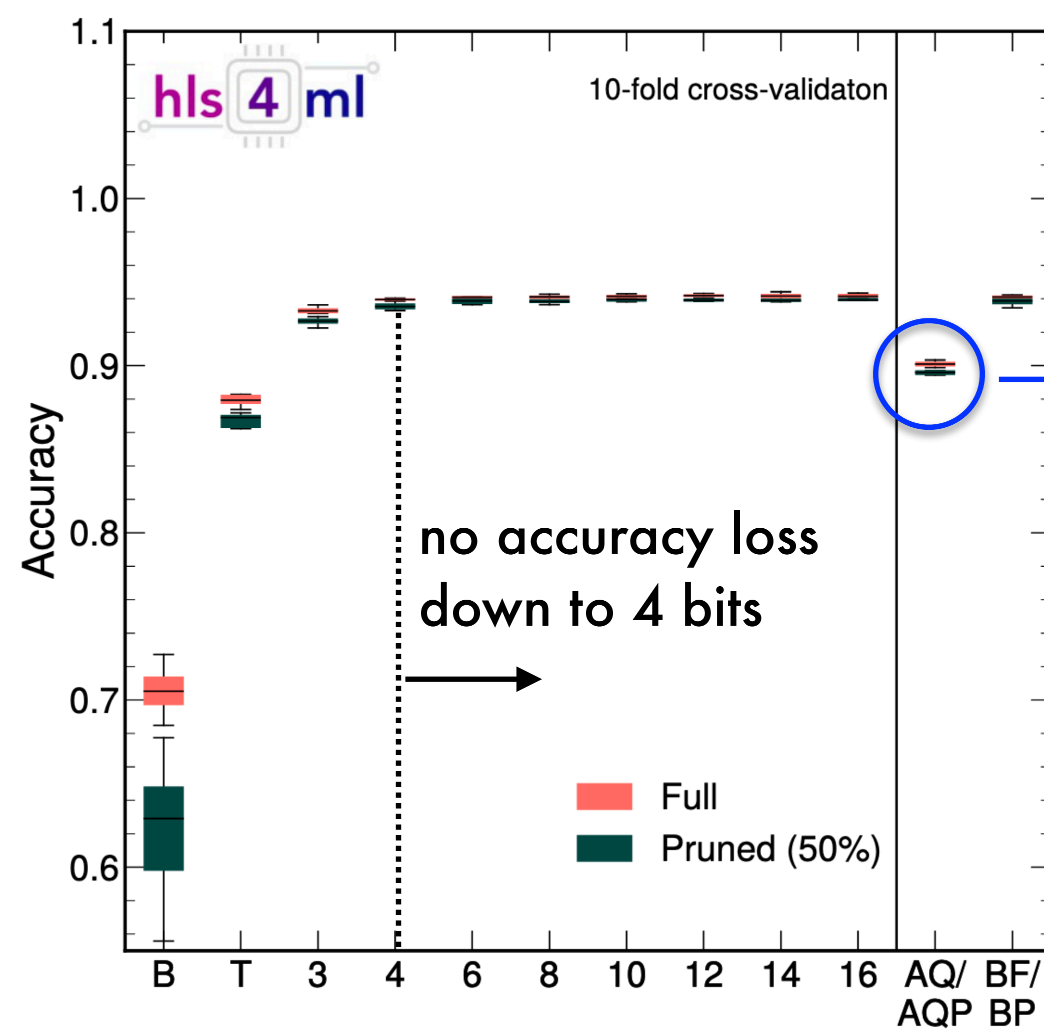


Fast convolutional neural networks

NEW!



Evaluate performance on street-view house numbers dataset (32x32x3)



heterogeneously quantized model through bayesian optimization

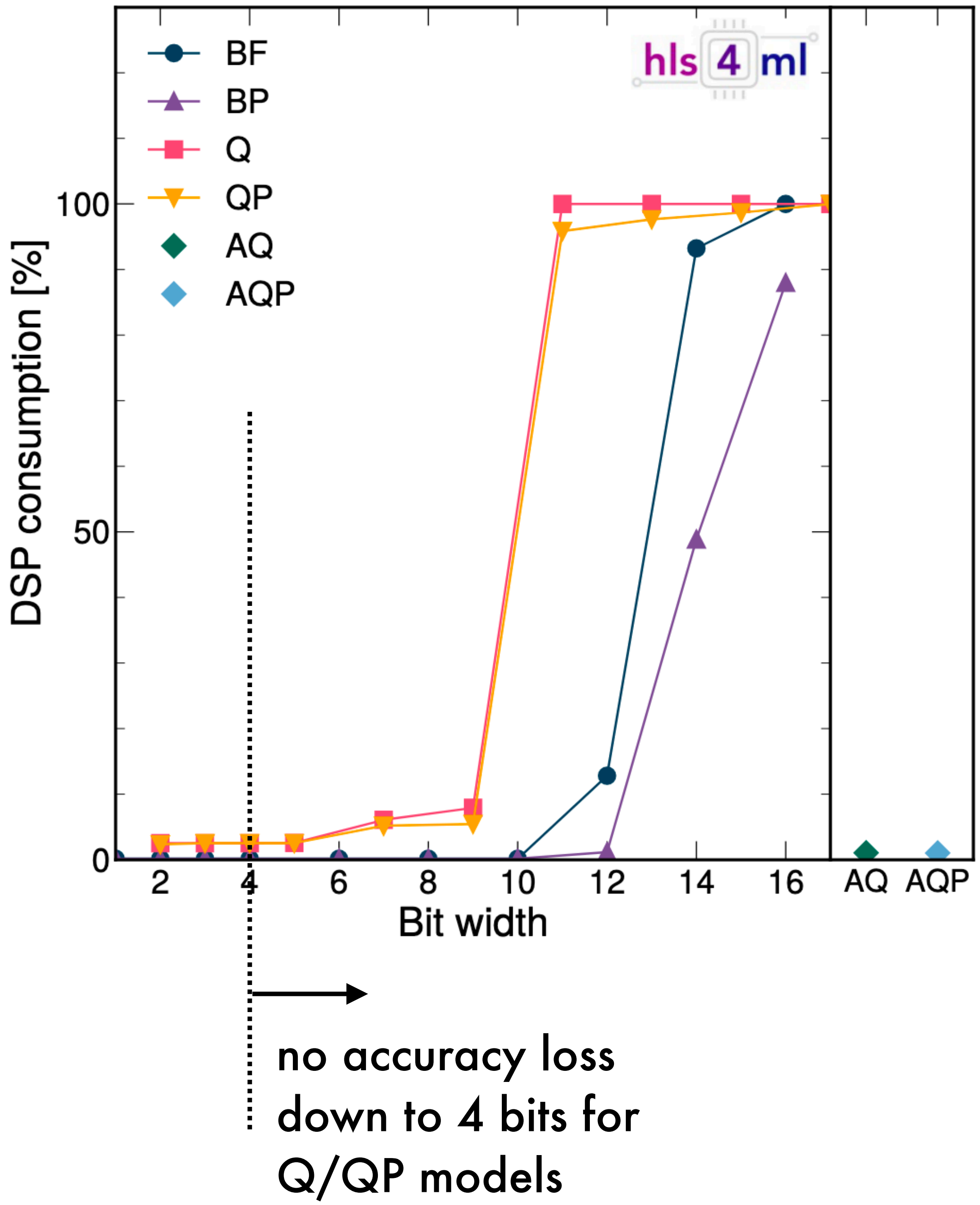
[arxiv.2101.05108](https://arxiv.org/abs/2101.05108)

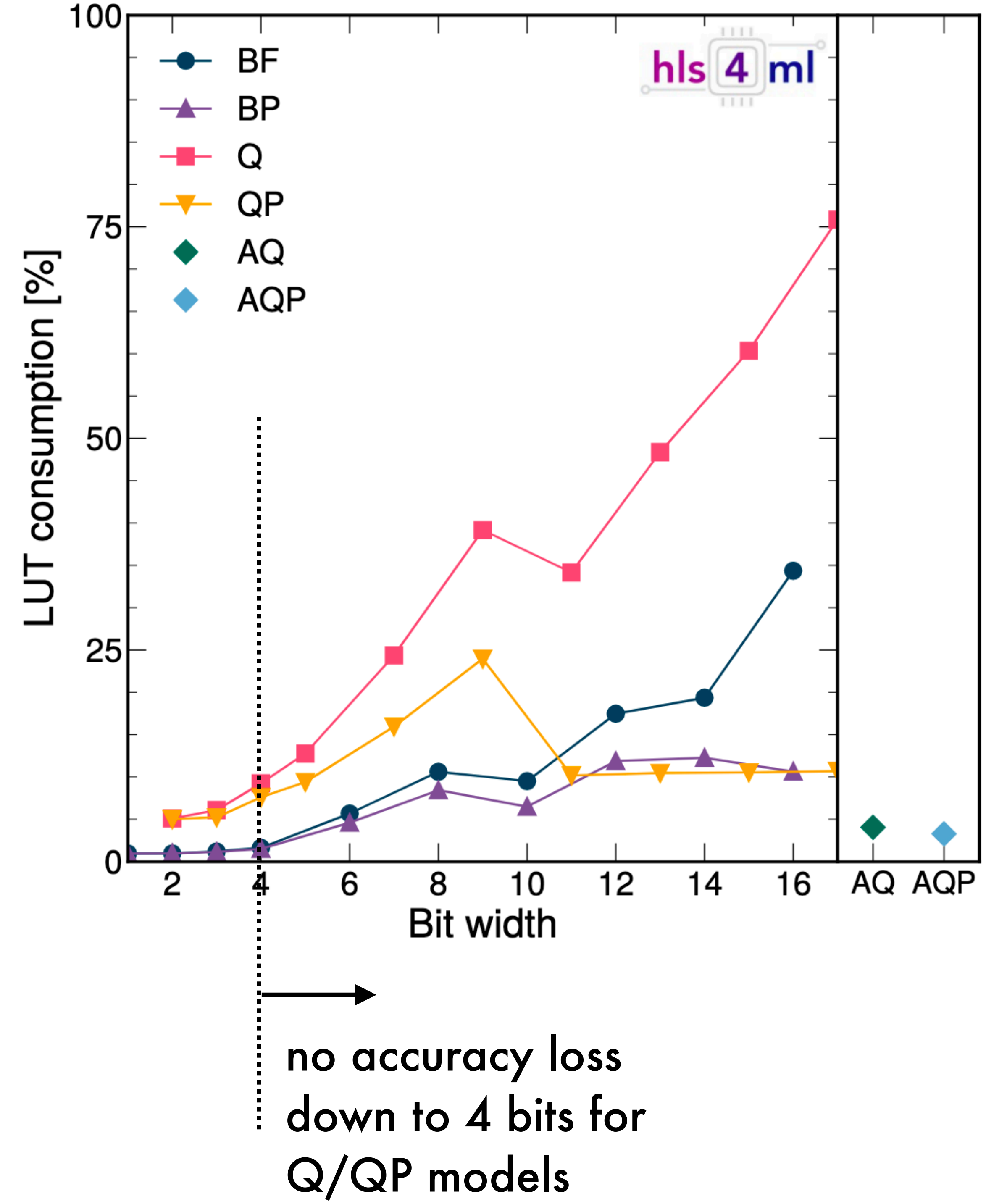
Fast convolutional neural networks

NEW!

Max parallelization, i.e. reuse factor = 1

arxiv.2101.05108



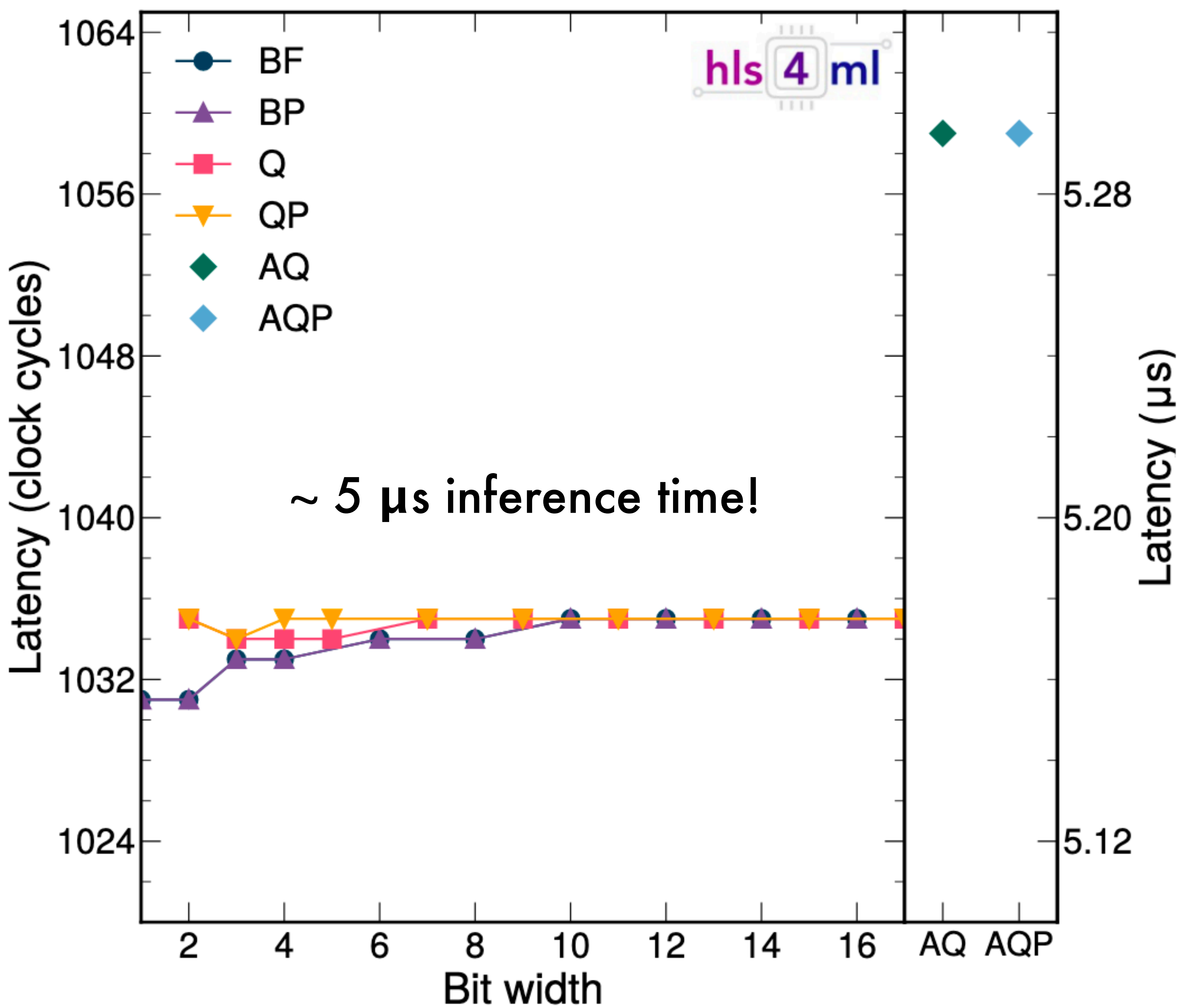


Fast convolutional neural networks

NEW!

arxiv.2101.05108

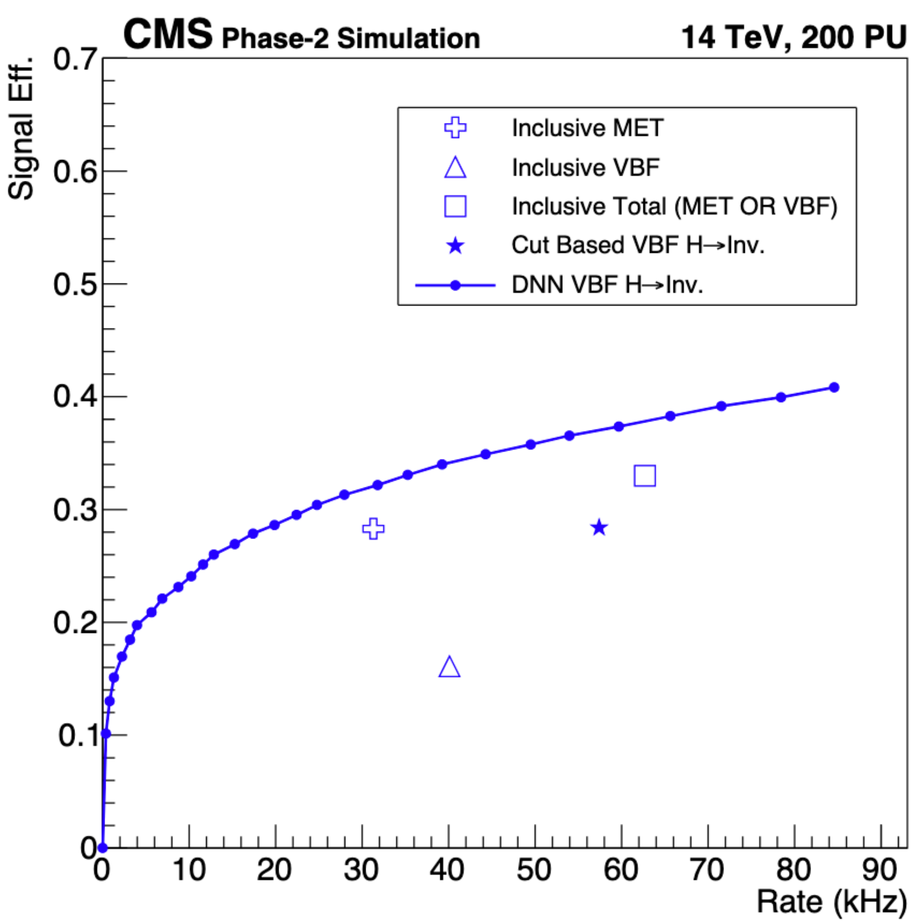
Max parallelization

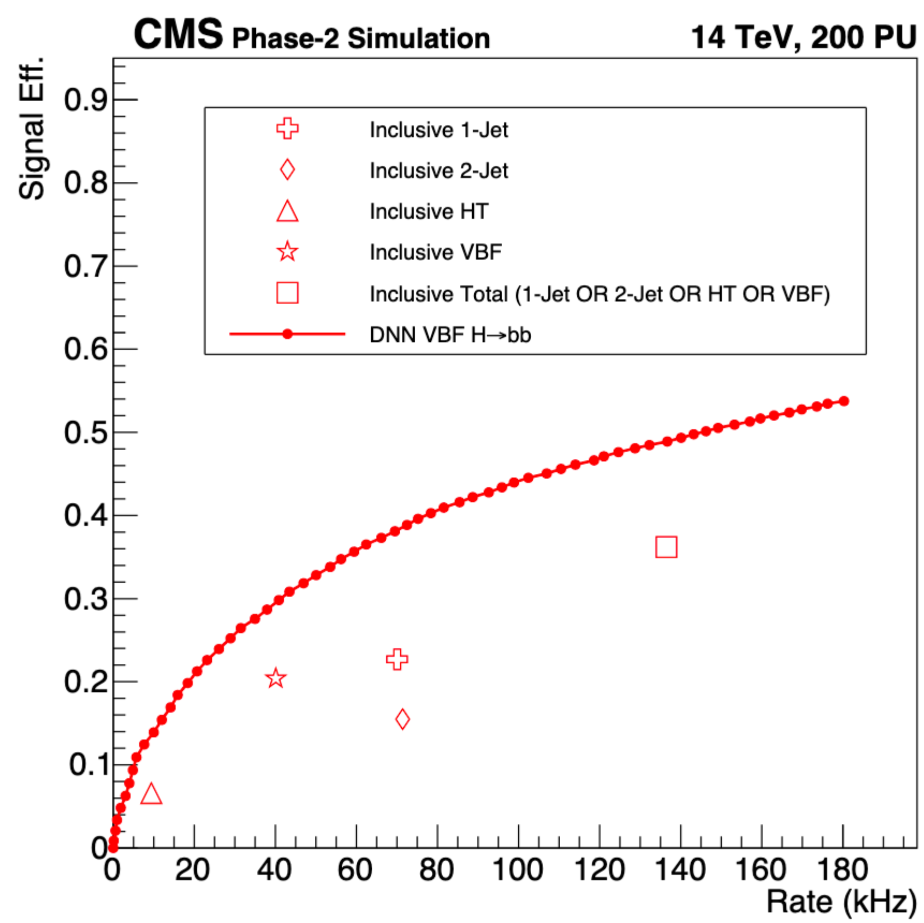


hls4ml for triggering @ 40 MHz

- hls4ml enabled developments of new trigger algorithms with large gain for physics!
  - replace standard cut-based algorithms

CMS Phase-2 L1 trigger upgrade TDR





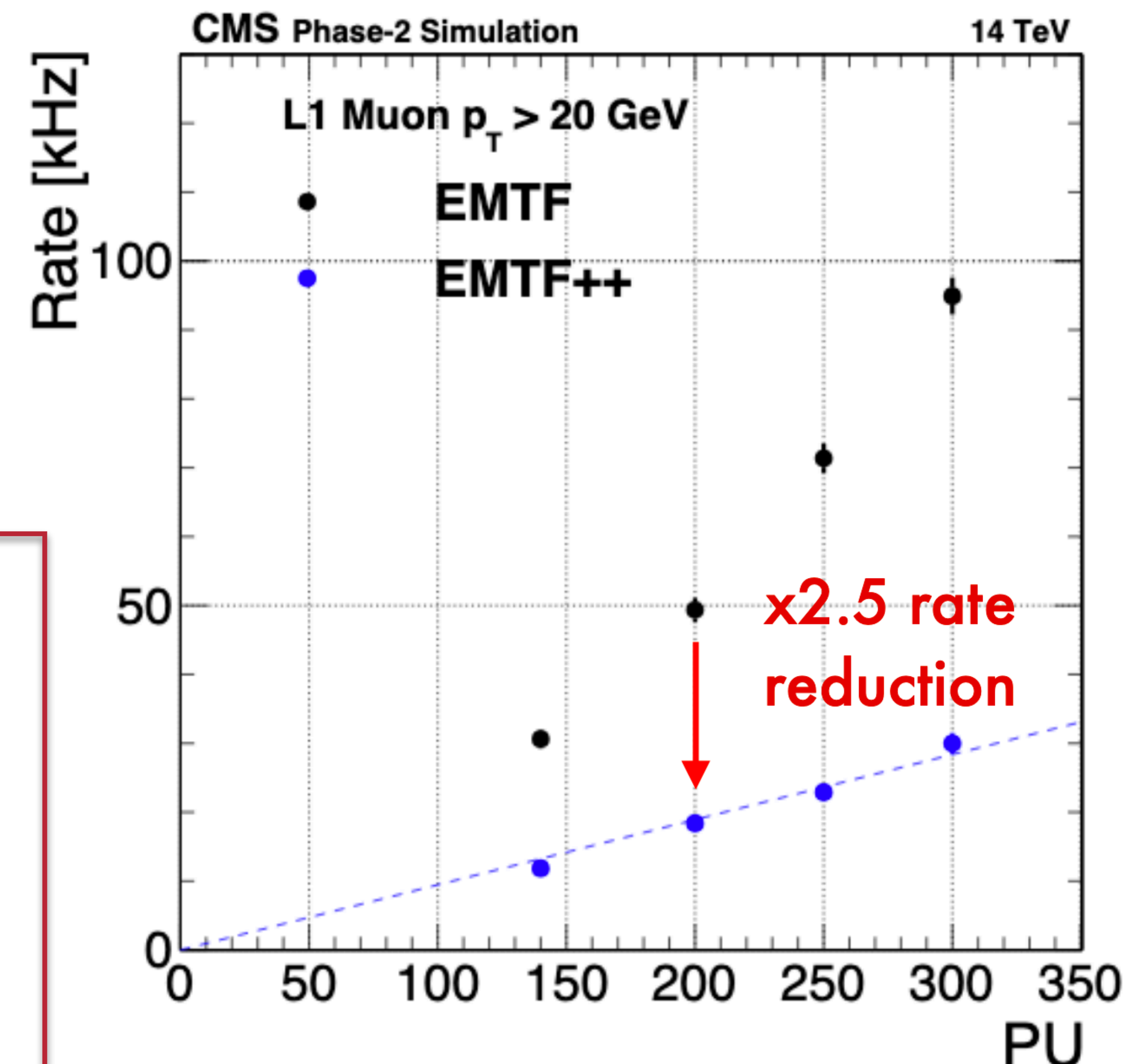
NN VBF H→bb

	Usage	Percentage
Latency	24 clk @ 200MHz	
II	5	
DSP48E	484	8%
FF	32634	2%
LUT	62358	9%

hls4ml for triggering @ 40 MHz

- hls4ml enabled developments of new trigger algorithms with large gain for physics!
  - replace standard cut-based algorithms
  - improve physics objects reconstruction (muons, taus, jets)

36 INPUT FEATURES:
φ,θ of track segments in muon stations
track segment quality
track segment curvature
⬇
3 HIDDEN LAYERS (30x25x20)
⬇
1 OUTPUT: muon p_T



CMS Phase-2 L1 trigger upgrade TDR

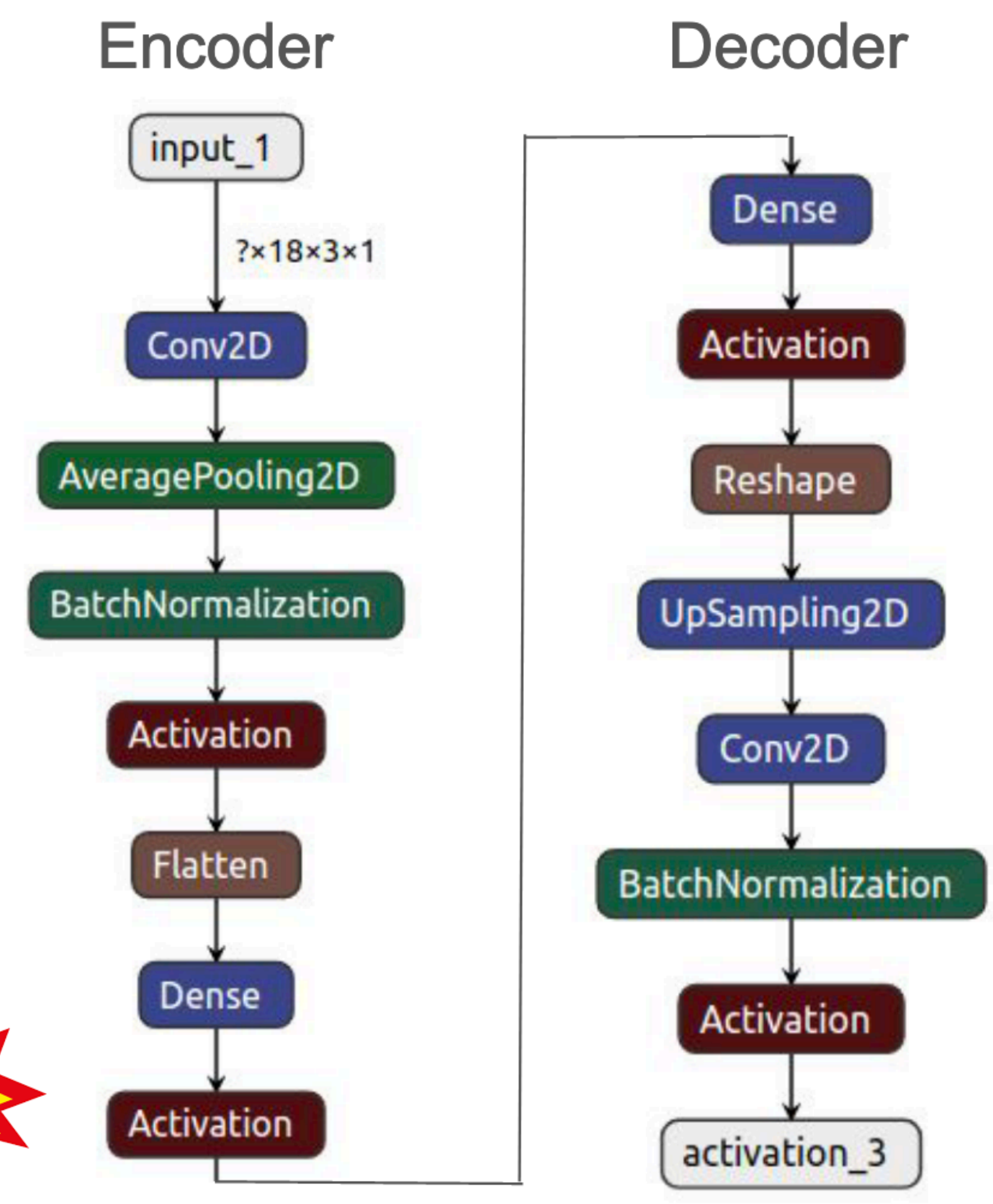
hls4ml for triggering @ 40 MHz

- hls4ml enabled developments of new trigger algorithms with large gain for physics!
  - replace standard cut-based algorithms
  - improve physics objects reconstruction (muons, taus, jets)
  - develop new strategies like anomaly detection with autoencoders for signal-agnostic triggering

21 inputs: $p_T/\eta/\Phi$ of 4 e/γ, 4 μ, 10 jets, and MET
→ input 19x3 input image

300 ns latency
30% DSPs
10% FFs
30% LUTs

with no pruning/quantization!



K. Govorkova @ Fast Machine Learning workshop 20

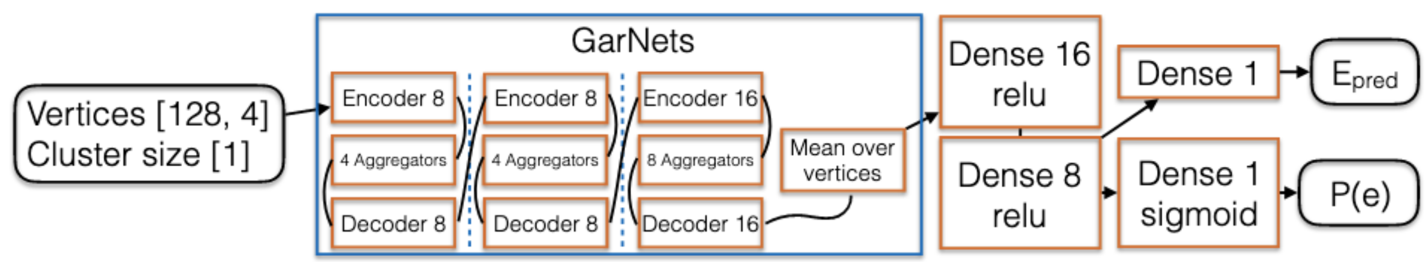
hls4ml for triggering @ 40 MHz

- hls4ml enabled developments of new trigger algorithms with large gain for physics!
  - replace standard cut-based algorithms
  - improve physics objects reconstruction (muons, taus, jets)
  - develop new strategies like anomaly detection with autoencoders for signal-agnostic triggering

Frontiers in Big Data 3 (2021) 44

- Allows also for integration of custom architectures like graph NNs to achieve ultra-low inference latency
  - calorimeter clusters classification [CTD 2020]
  - charged particles track reconstruction [NeurIPS 2020]

NEW!



	Continuous	Quantized
Latency	155 clk (0.83 μs)	148 clk (0.80 μs)
Initiation interval	55 clk (0.28 μs)	50 clk (0.25 μs)
LUT	57k (8.6%)	70k (11%)
FF	39k (3.0%)	41k (3.1%)
DSP	3.1k (57%)	1.6k (28%)
BRAM	1.8 Mb (2.3%)	1.9 Mb (2.4%)

Summary

- hls4ml enables automatic translation of modern deep learning architectures to synthesizable FPGA firmware and more
  - today presented most recent developments
- Presented applications for the hardware trigger at LHC experiments but many others ongoing beyond LHC
  - eg, accelerator controls → see C. Herwig talk
  - other cases being identified with common challenges (eg., large scale LArTPC experiments or gravitational waves detection)
- The library is also expanding beyond FPGAs
  - see J. Hirschauer talk on the application of hls4ml to achieve DL-dedicated ASICs design for CMS high-granularity calorimeter (and our recent paper arxiv.2103.05579)
- Very active developers team… stay tuned for new features and applications!

high level synthesis for machine learning

For more info:

https://fastmachinelearning.org/hls4ml/

Fast inference of deep neural networks in FPGAs for particle physics [JINST 13 P07027 (2018)]
Fast inference of Boosted Decision Trees in FPGAs for particle physics [JINST 15 P05026 (2020)]
Compressing deep neural networks on FPGAs to binary and ternary precision with HLS4ML [2020 Mach. Learn.: Sci. Technol]
Automatic deep heterogeneous quantization of Deep Neural Networks for ultra low-area, low-latency inference on the edge at particle colliders [arxiv.2006.10159]
Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics [arxiv.2008.03601]
Fast convolutional neural networks on FPGAs with hls4ml [arxiv.2101.05108]
Accelerated Charged Particle Tracking with Graph Neural Networks on FPGAs [arxiv.2012.01563]
hls4ml: An Open-Source Codesign Workflow to Empower Scientific Low-Power Machine Learning Devices [arxiv.2103.05579]

Thank you!