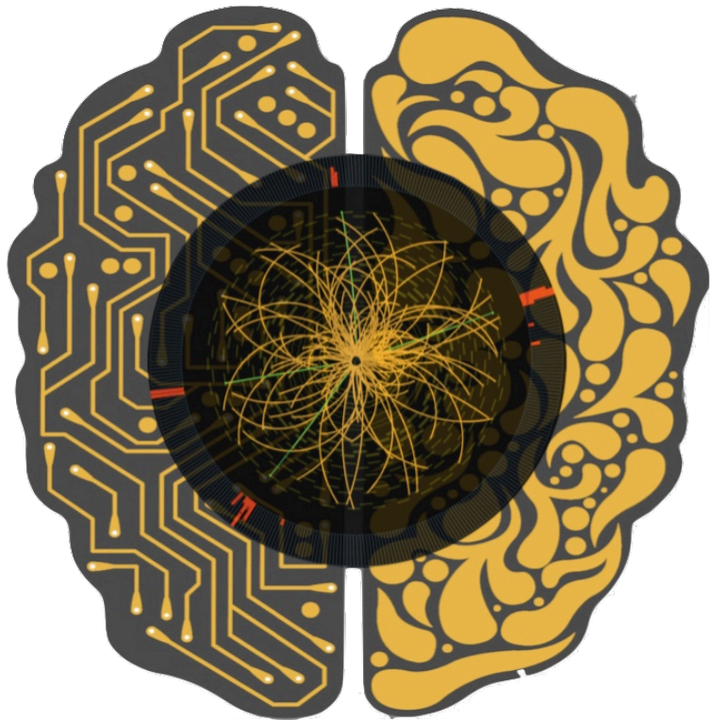


Fast deep neural network inference on FPGAs

CERN Openlab Technical Workshop
CERN - 22/1/2020

Javier Duarte, Christian Herwig, Sergo Jindariani, Ben Kreis, Ryan Rivera, Nhan Tran (Fermilab)
Thea Årrestad, Jennifer Ngadiuba, Maurizio Pierini, Vladimir Loncar, **Sioni Summers** (CERN)
Edward Kreinar (Hawkeye 360)
Phil Harris, Song Han, Dylan Rankin (MIT)
Zhenbin Wu (University of Illinois at Chicago)

Contents

- Introduction - LHC ‘big data’ problem
- Neural Network to FPGA translation with **hls4ml** and Quartus HLS

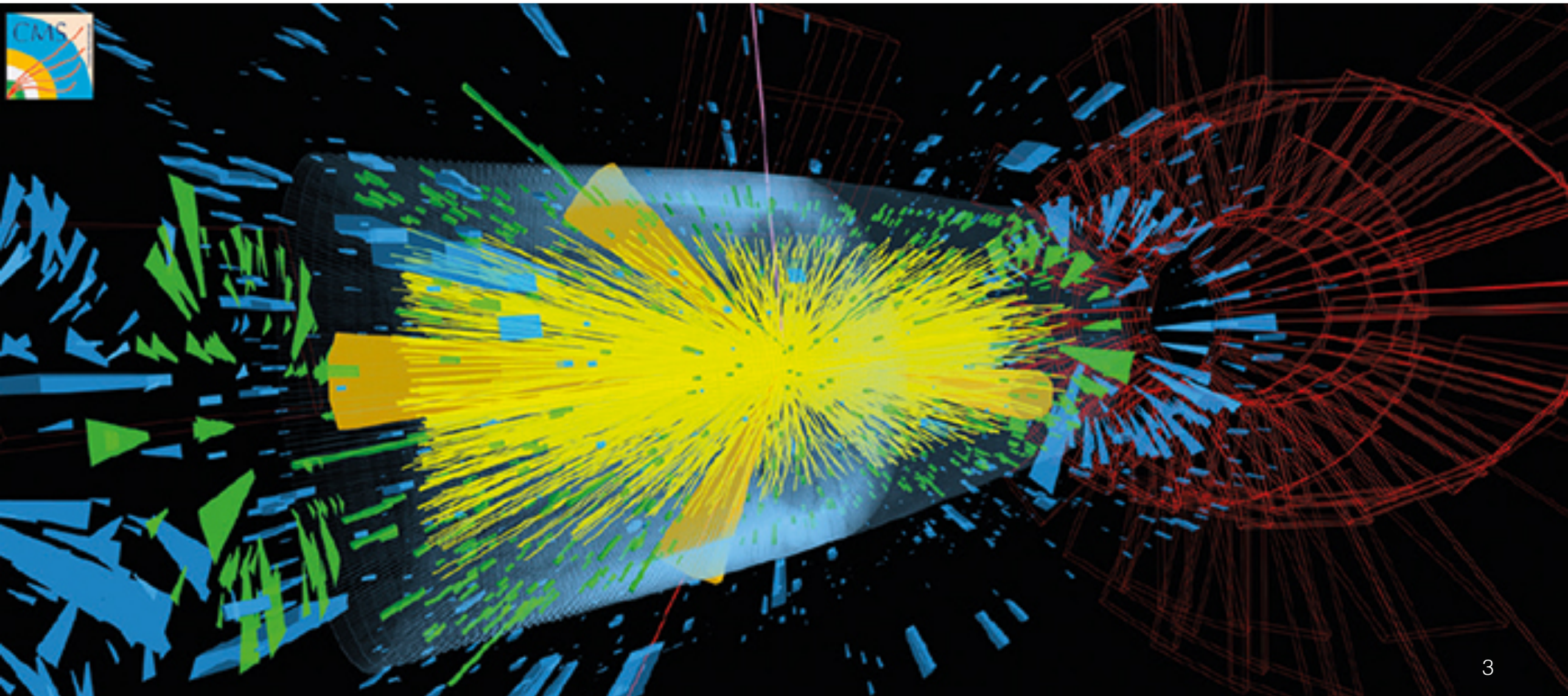
The challenge: triggering at (HL-)LHC

At HL-LHC, will expect 200 pileup collisions

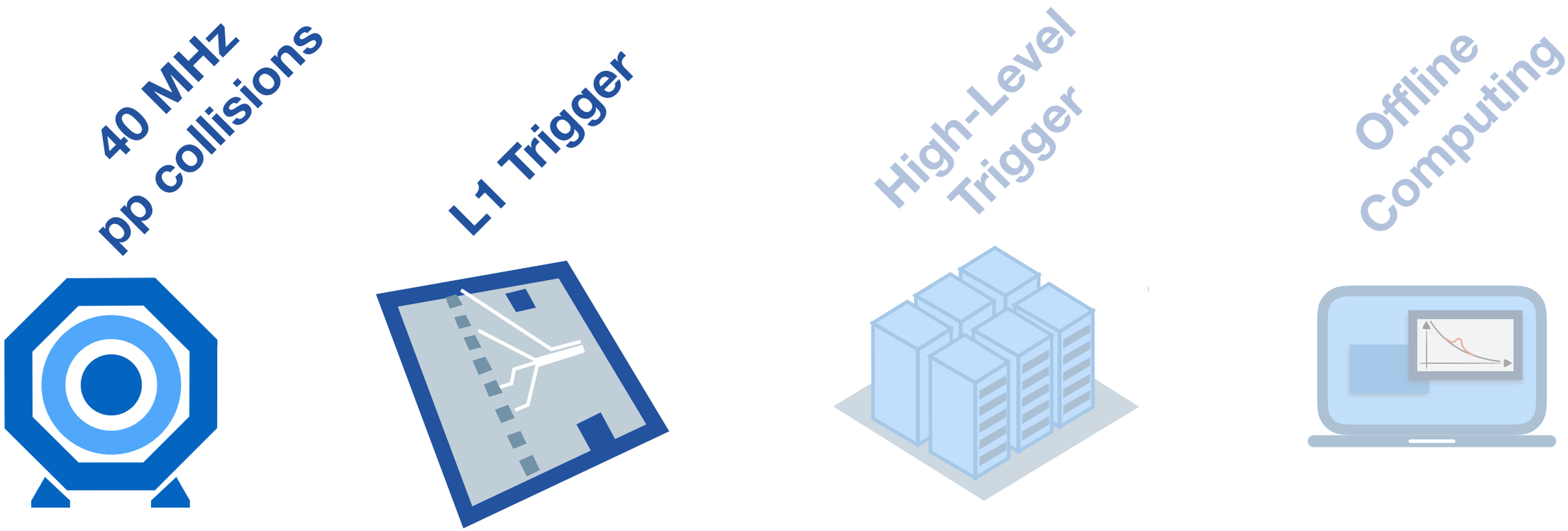
Extreme bunch crossing frequency of 40 MHz → extreme data rates O(100 TB/s)

“Triggering” = filter events to reduce data rates to manageable levels

The challenge: maintain trigger sensitivity in busier environment!

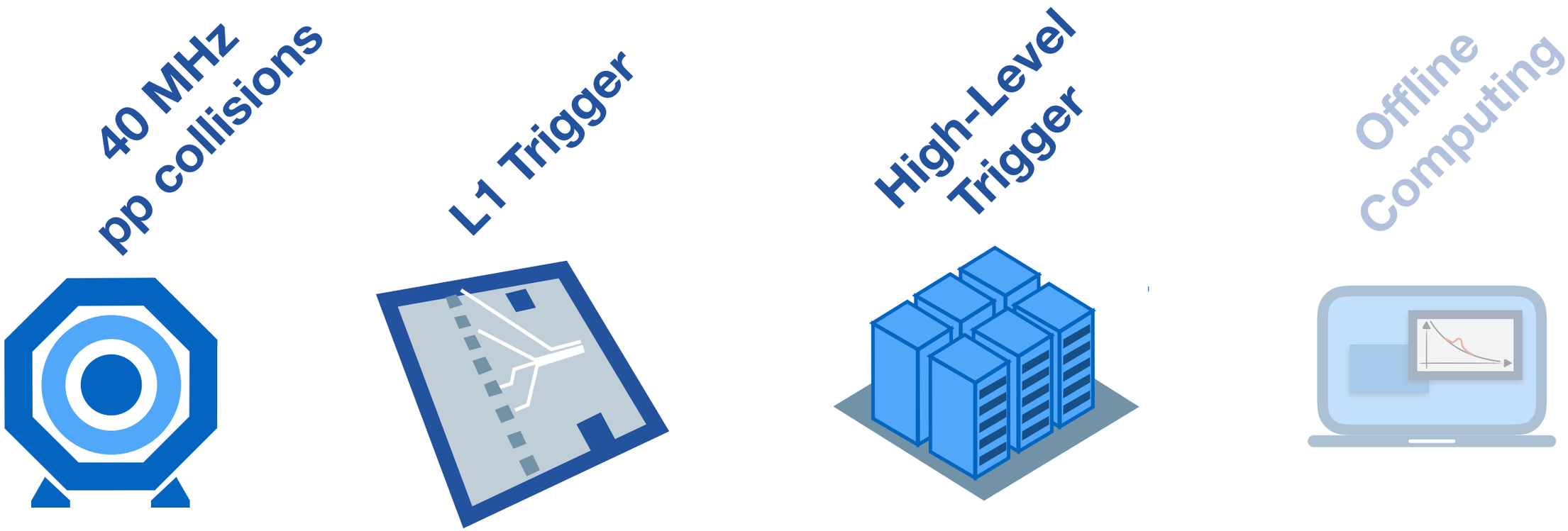


The LHC big data problem



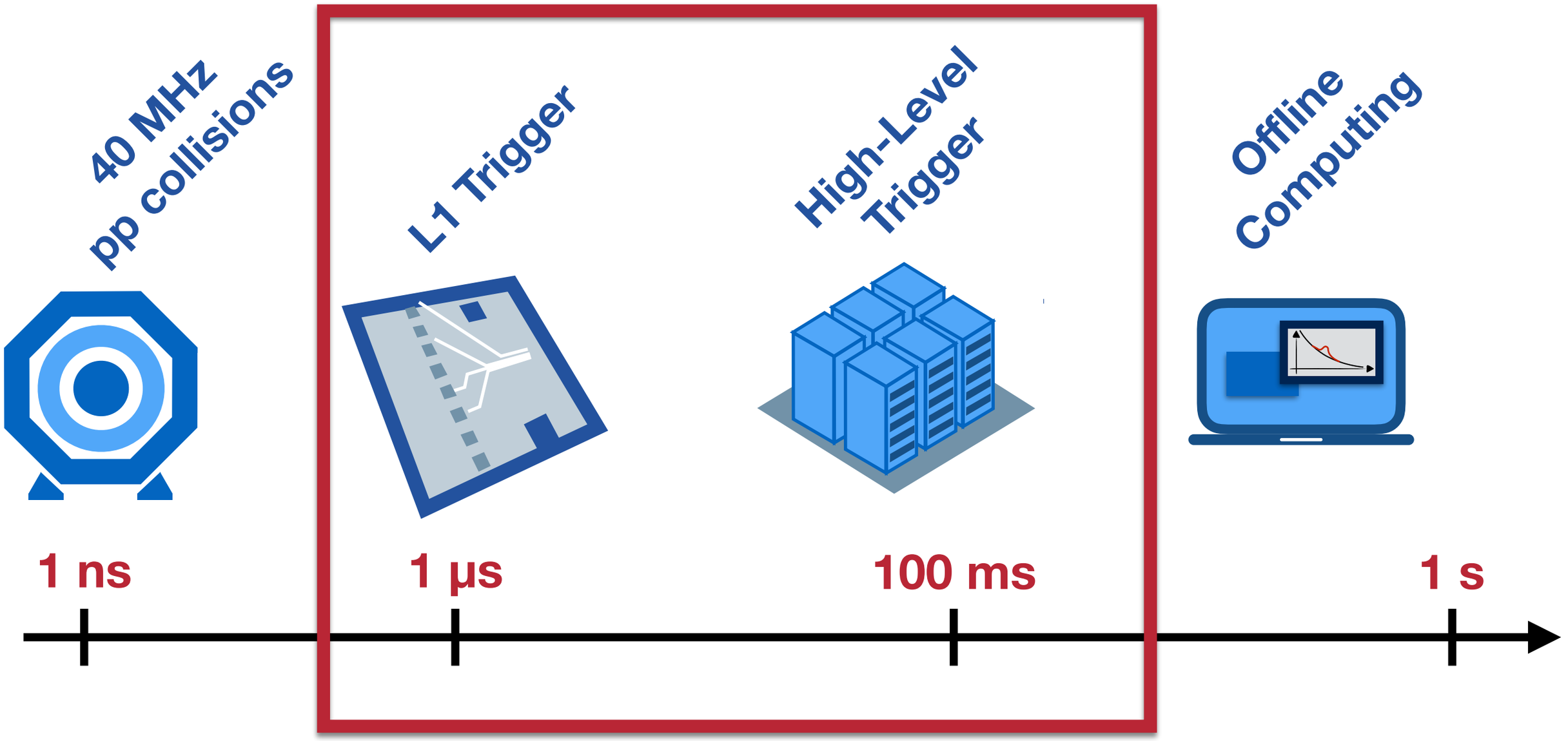
- 40 MHz in / 100 KHz out
- Absorbs 100s TB/s
- Trigger decision to be made in **~ 10 µs**
- Coarse local reconstruction
- FPGAs / Hardware implemented

The LHC big data problem



- 100 KHz in / 1 KHz out
- Output: ~ 500 KB/event
- Processing time **~ 300 ms**
- Simplified global reconstruction
- Software implemented on CPUs

The LHC big data problem

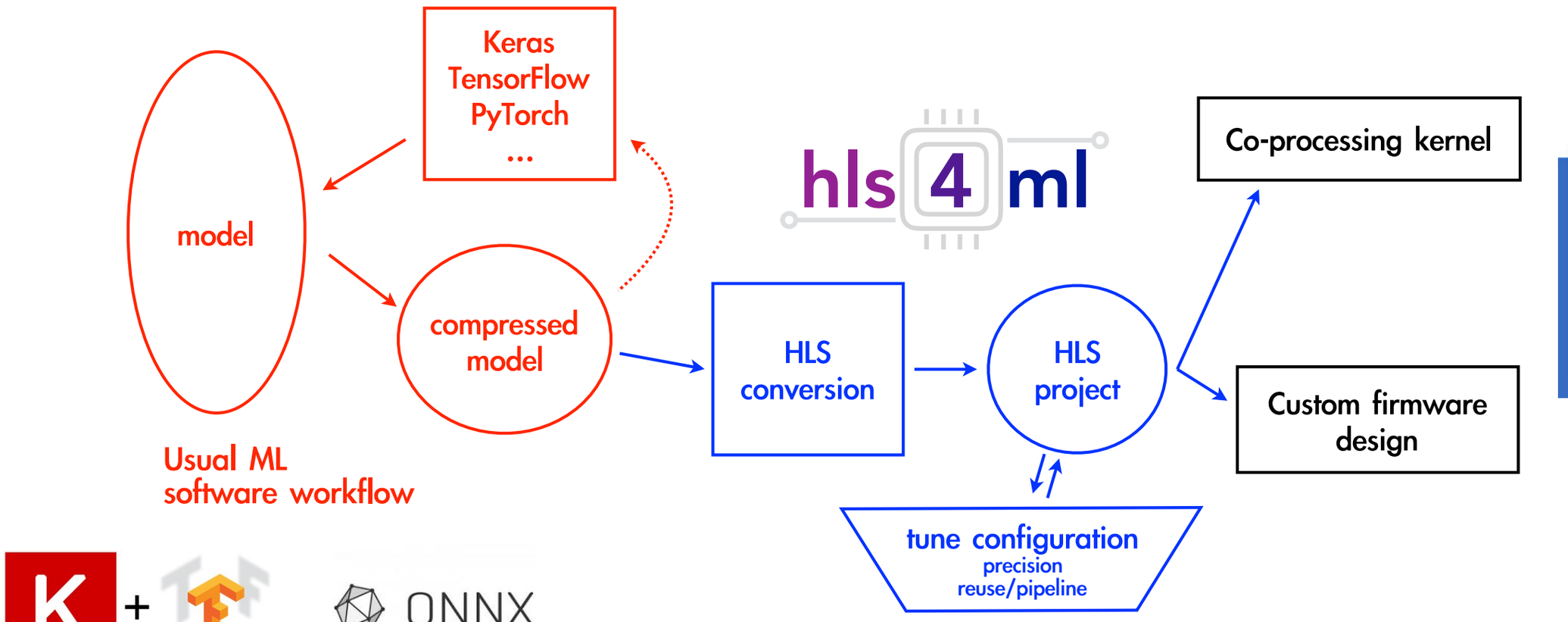


Deploy ML algorithms very early in the game
Challenge: strict latency constraints!

high level synthesis for machine learning

Implemented a user-friendly, open source tool to develop and optimize FPGA firmware design for DL inference:

- Reads as input models trained with standard DL libraries
- Uses HLS software (accessible to non-FPGA-expert, e.g. HEP physicist)
- Comes with implementation of common ingredients (layers, activation functions)
- And optimizations

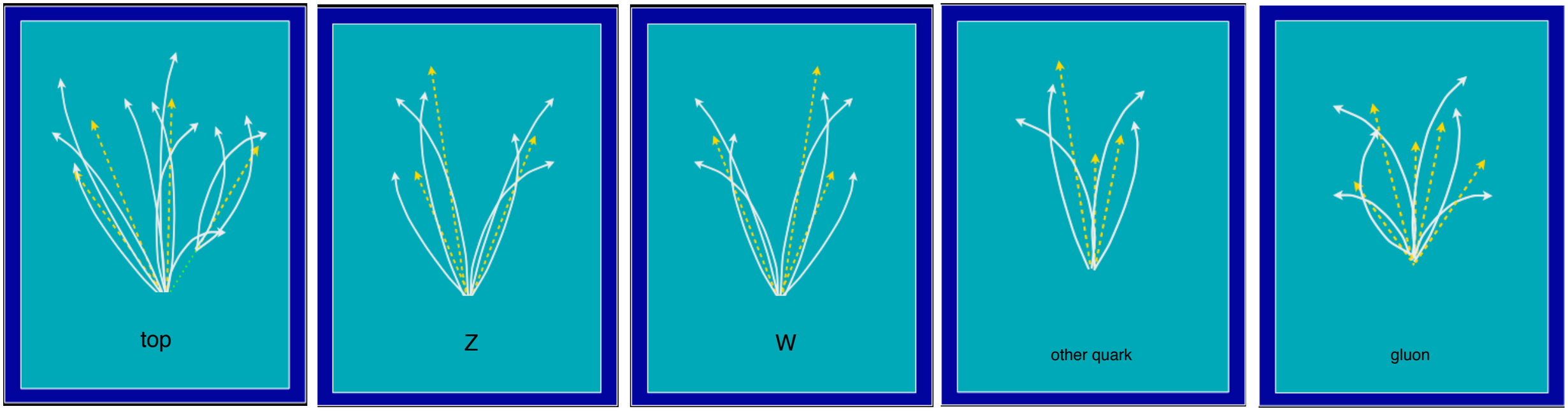


https://hls-fpga-machine-learning.github.io/hls4ml/

https://arxiv.org/abs/1804.06913

Physics case: jet tagging

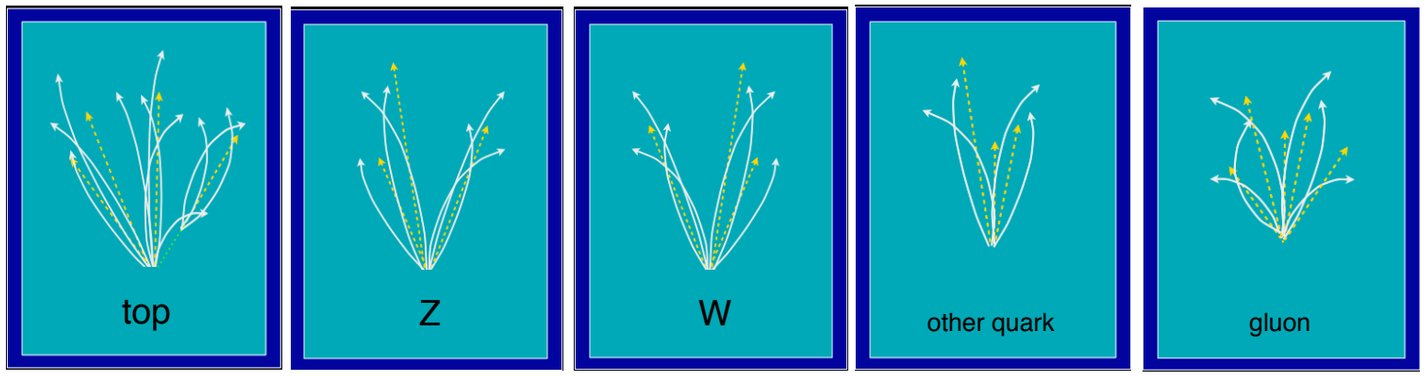
Study a **multi-classification task to be implemented on FPGA:** discrimination between highly energetic (boosted) ***q, g, W, Z, t*** initiated jets

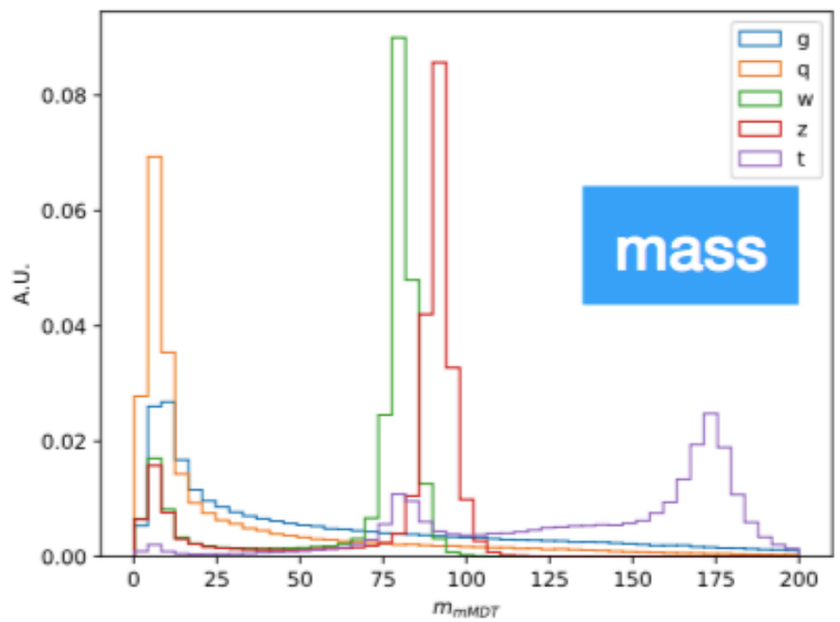


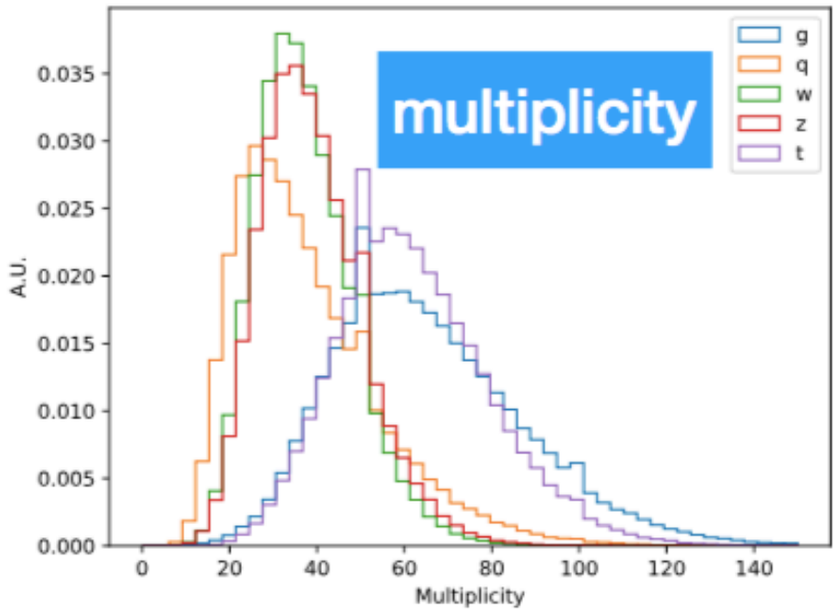
t→bW→bqq	**Z→qq**	**W→qq**	**q/g background**
3-prong jet	2-prong jet	2-prong jet	no substructure and/or mass ~ 0

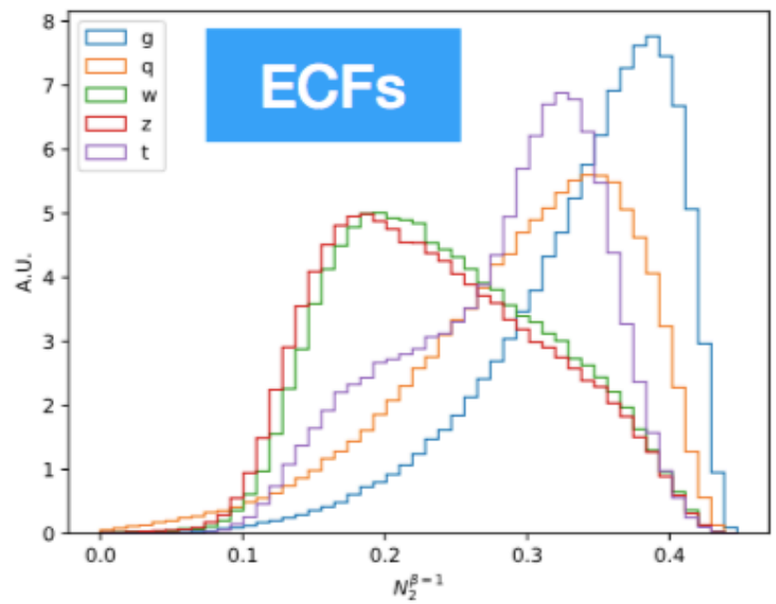
Reconstructed as one massive jet with substructure

Physics case: jet tagging









Input variables: several observables known to have high discrimination power from offline data analyses and published studies [*]

[*] D. Guest at al. PhysRevD.94.112002, G. Kasieczka et al. JHEP05(2017)006, J. M. Butterworth et al. PhysRevLett.100.242001, etc..

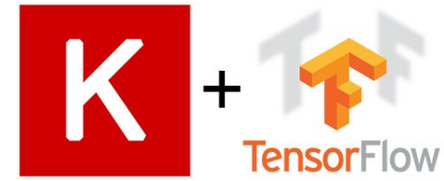
Physics case: jet tagging

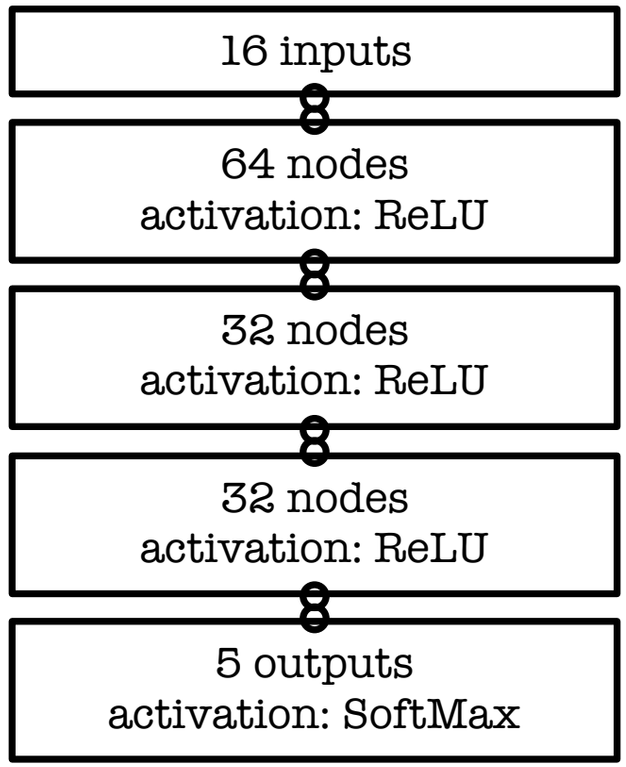
- We train (on GPU) the **five output multi-classifier** on a sample of ~ 1M events with two boosted WW/ZZ/tt/qq/gg anti-k_T jets

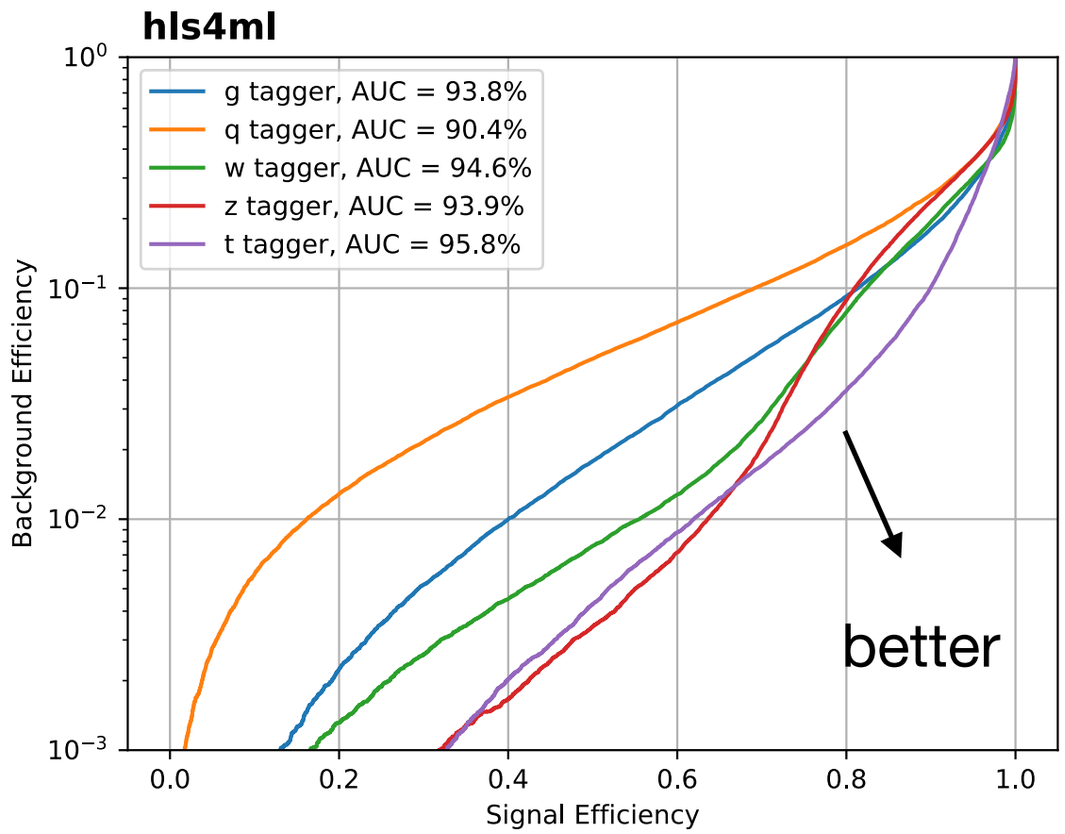
- Fully connected neural network with **16 expert-level inputs**:

  - Relu activation function for intermediate layers

  - Softmax activation function for output layer







AUC = area under ROC curve (100% is perfect, 20% is random)

Results: Quartus HLS

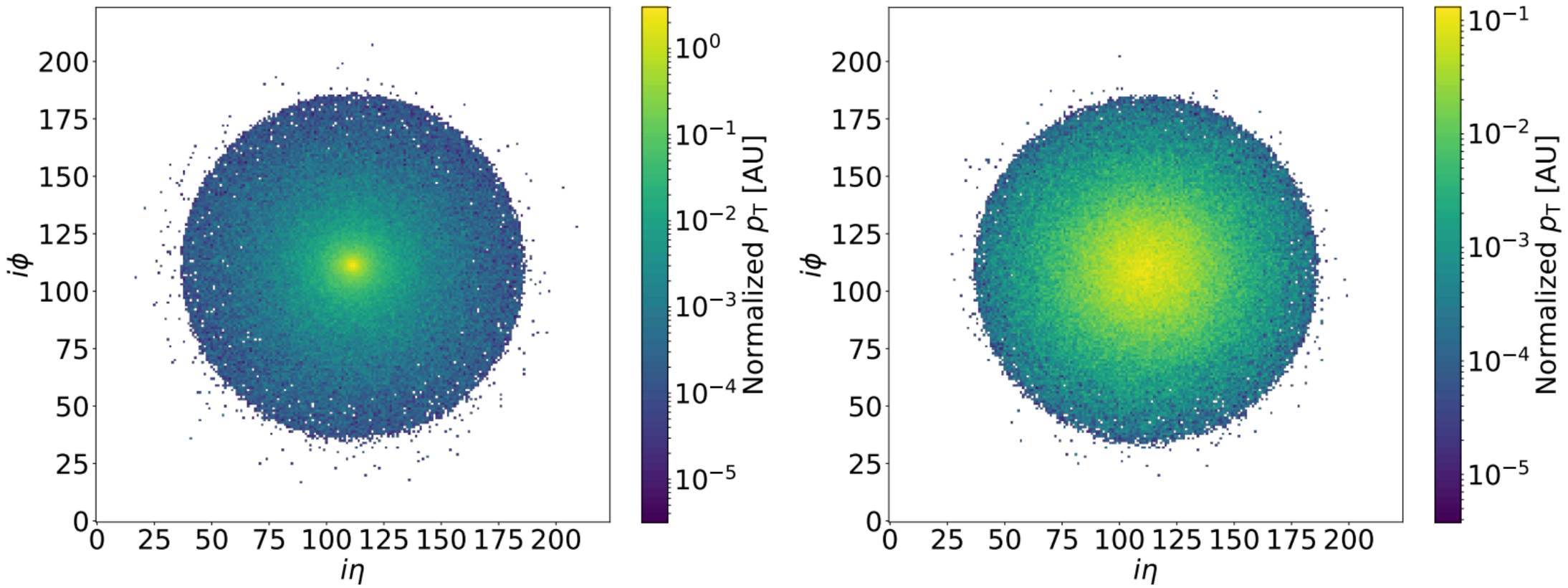
- Promising resources and latency achieved
- Timing: 54 clock cycles at 4 ns clock period = 220 ns latency

Resource	DSP	LUT	FF
Usage (% Arria 10)	500 (16%)	500×10^3 (43%)	140×10^3

- Room for further optimisation
  - Particularly to encourage compiler to use DSPs rather than LUTs
- Need to implement ‘reuse’ : user controlled tradeoff between resources and latency

Jet images with ResNet-50

- Create pixelised jet images from jet particle constituents: sum pT in (η, ϕ) grid
- Use ResNet-50 featurizer (Convolutional Neural Network) with custom Fully Connected classifier at the end: 90% accuracy
- Image: Average QCD (left) and top quark (right) initiated jet images over 5,000 jets

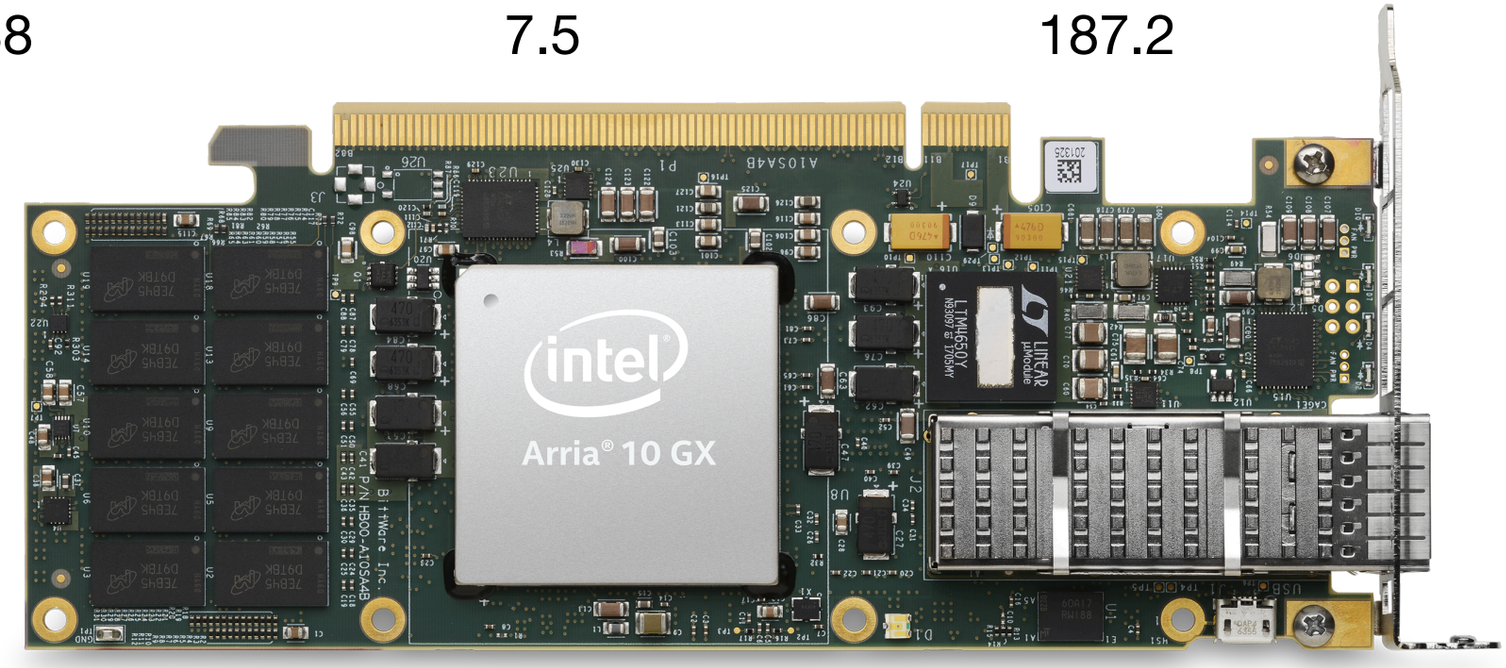


arXiv:1904.08986

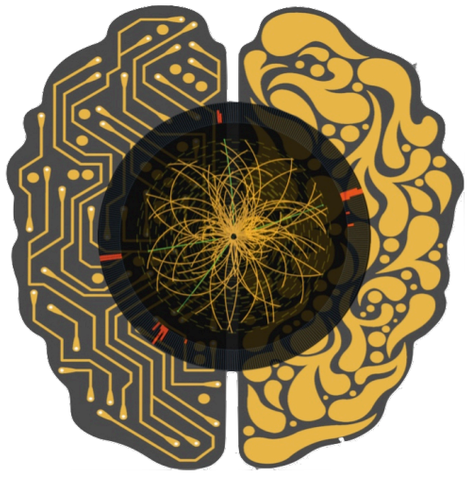
Classification with OpenVINO™

- OpenVINO: rapid, optimised deployment of Computer Vision models (e.g. ResNet-50) on Intel CPU, Integrated Graphics, FPGA, Neural Compute Stick
- Deploy jet-image retrained ResNet-50 on Intel Arria 10 PAC using OpenVINO

Hardware	Accuracy	Latency (ms)	Max Throughput (img/s)
2 x Xeon 8 core 2.1GHz	0.91	88	11.4
Arria 10 PAC FP16	0.91	14	84.4
Arria 10 PAC FP11	0.88	7.5	187.2



Conclusion



hls4ml

- hls4ml software package translates trained neural networks into synthesizable FPGA firmware
- Targeting Level 1 Trigger - big FPGAs, O(1 μs) latency
- Increasing support for Quartus HLS and gaining more optimal results
- Benchmarking Intel PAC with jet-tagging ResNet-50
- Website: https://fastmachinelearning.org/
- Paper: https://iopscience.iop.org/article/10.1088/1748-0221/13/07/P07027
- Code: https://github.com/hls-fpga-machine-learning/HLS4ML