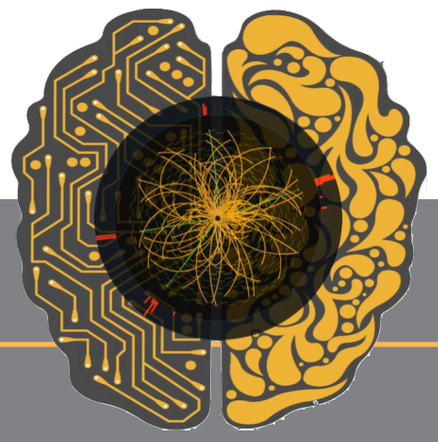

# Ultrafast deep learning inference on FPGAs

Jennifer Ngadiuba (CERN)

*on behalf of the hls4ml Collaboration*



# hls4ml: the origins

The project kicked off 3 years ago …
~ 10 people, mostly physicists (with little expertise in electronic engineering)

**Fast inference of deep neural networks in FPGAs for particle physics**

**J. Duarte,[a] S. Han,[b] P. Harris,[b] S. Jindariani,[a] E. Kreinar,[c] B. Kreis,[a] J. Ngadiuba,[d] M. Pierini,[d] R. Rivera,[a] N. Tran[a,1] and Z. Wu[e]**

[a] *Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.*
[b] *Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*
[c] *HawkEye360, Herndon, VA 20170, U.S.A.*
[d] *CERN, CH-1211 Geneva 23, Switzerland*
[e] *University of Illinois at Chicago, Chicago, IL 60607, U.S.A.*

# Current contributors

Many more contributors and users now!

https://fastmachinelearning.org/hls4ml/

Massachusetts Institute of Technology

Fermilab

erc European Research Council



UIC UNIVERSITY OF ILLINOIS AT CHICAGO

CERN

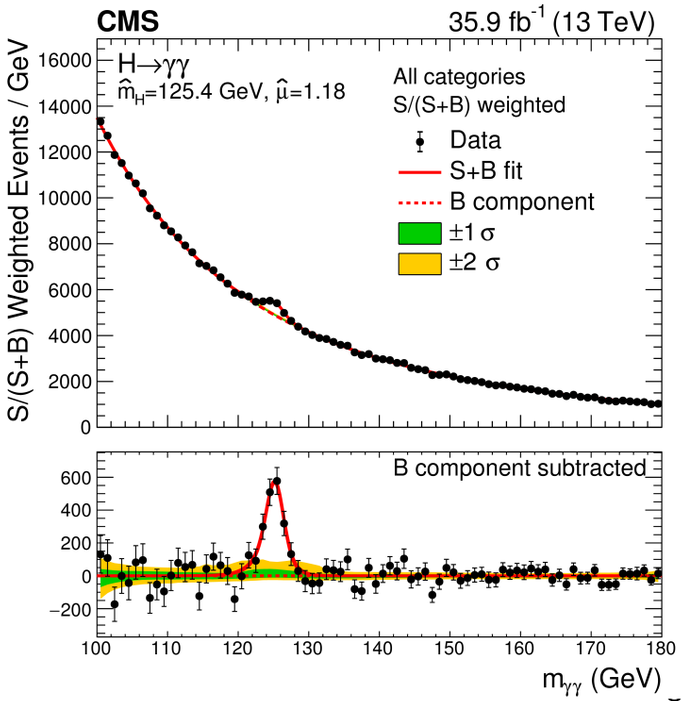
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UC San Diego

HawkEye 360
An Allied Minds Company

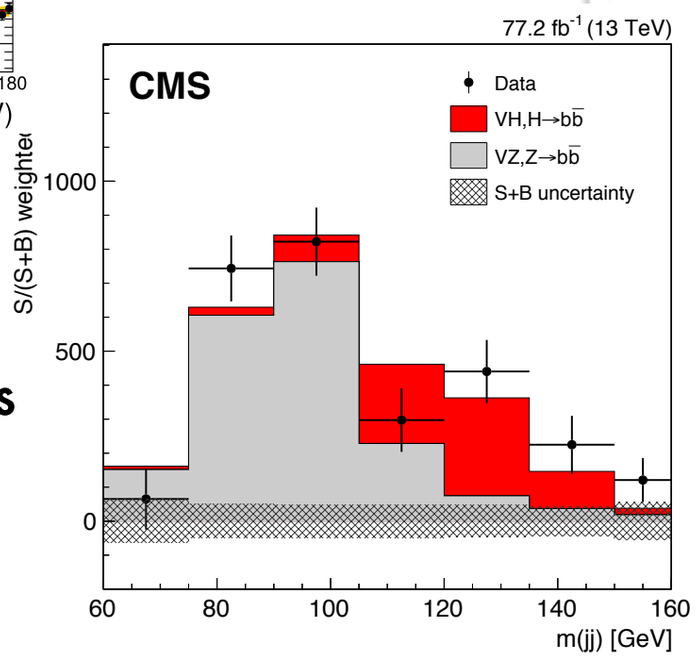
# Deep learning in HEP

ML is used in particle physics since the ‘80s.
Shallow networks back then, mostly BDTs since ~2004 (ex, Higgs boson discovery)
But application of modern DL technologies only gaining ground NOW!




## Higgs → photons

JHEP 11 (2018) 185



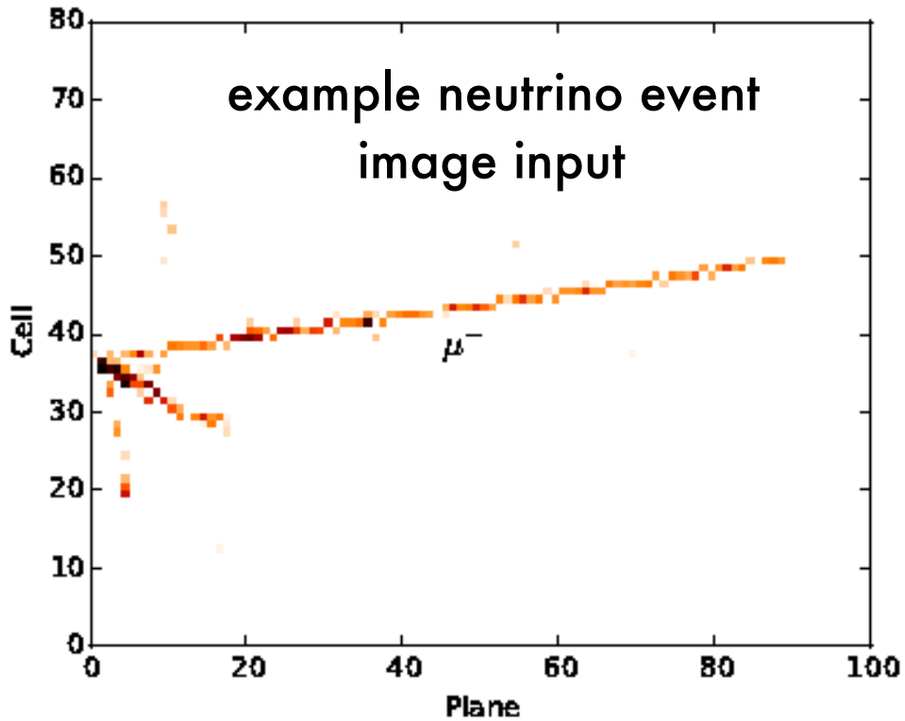

## Higgs → bottom quarks

Phys. Rev. Lett. 121, 121801 (2018)

# Deep learning in HEP

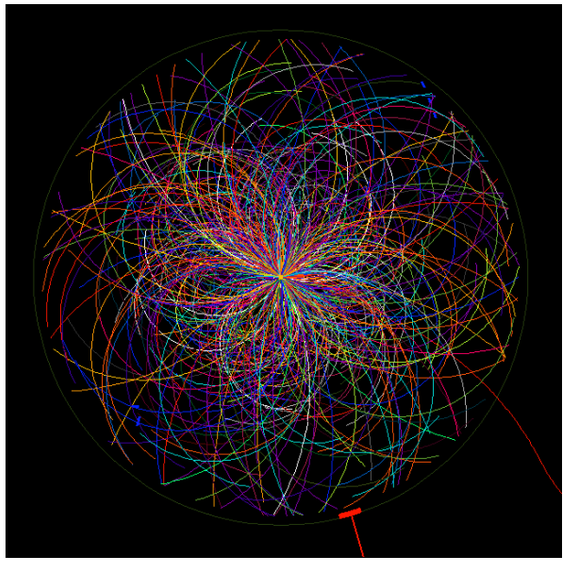
**Computer vision for neutrino experiments**

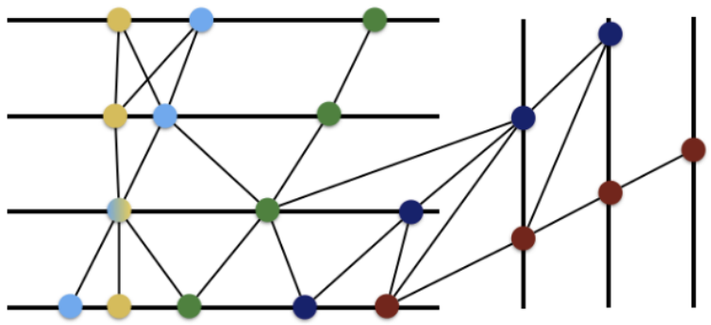
ex: JINST 11 P09001, GoogleNet inspired architecture for neutrino events classification for NOvA




**Graph NN for reconstruction @ LHC**

ex: charged particle trajectories (HEP.TrkX project)





**Recurrent NN for jet classification @ LHC**

ex: CMS-DP-2017-005, ATL-PHYS- PUB-2017-003, …

exploit natural jet sequential clustering history

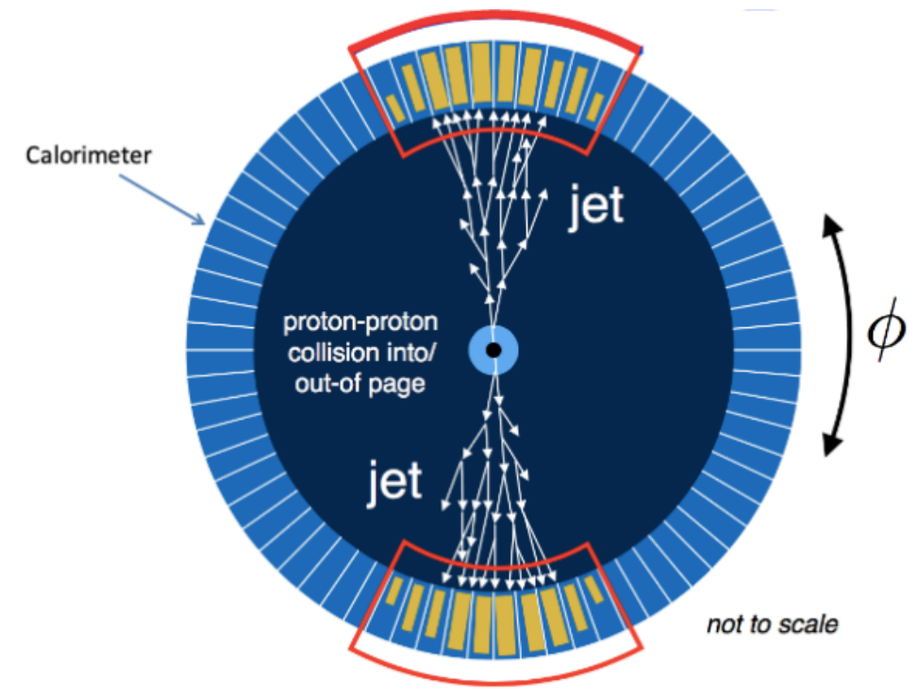
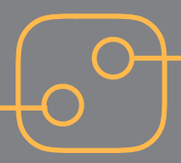



Image from B. Nachman

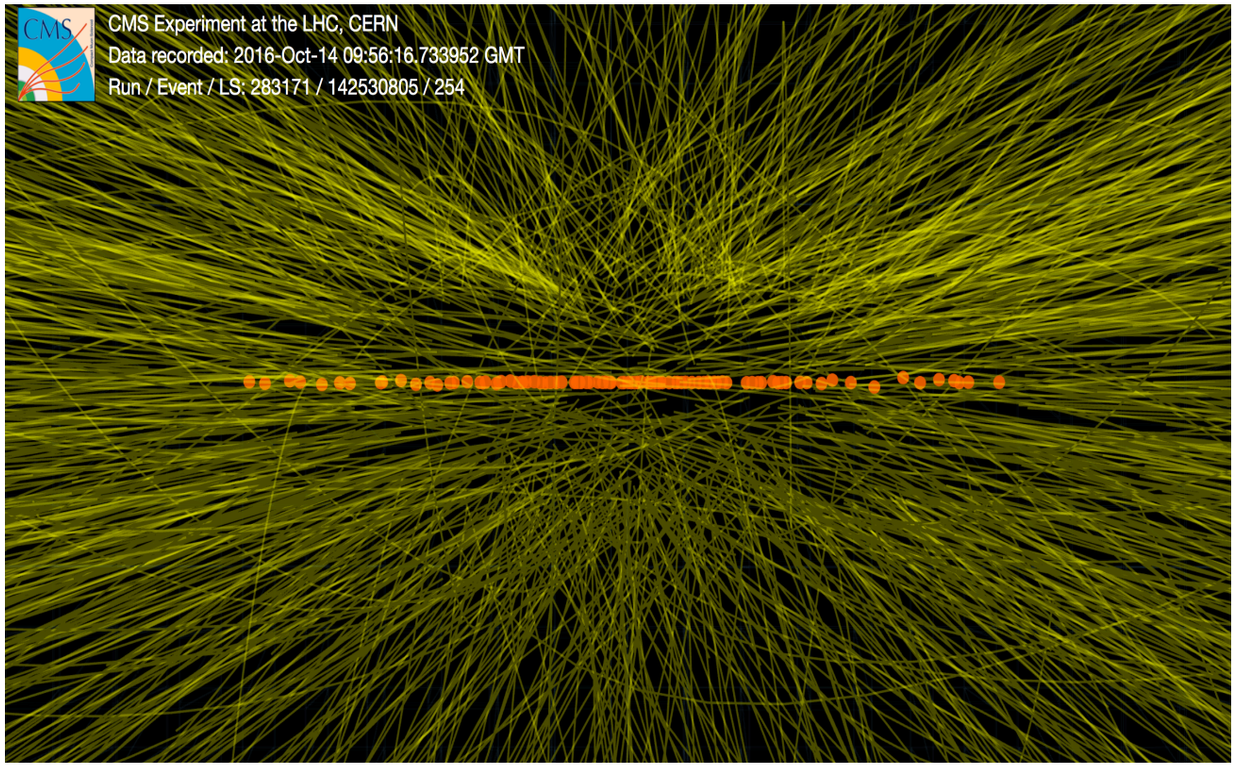
See Jean-Roch, Thomas, Georgia, Lindsey talks

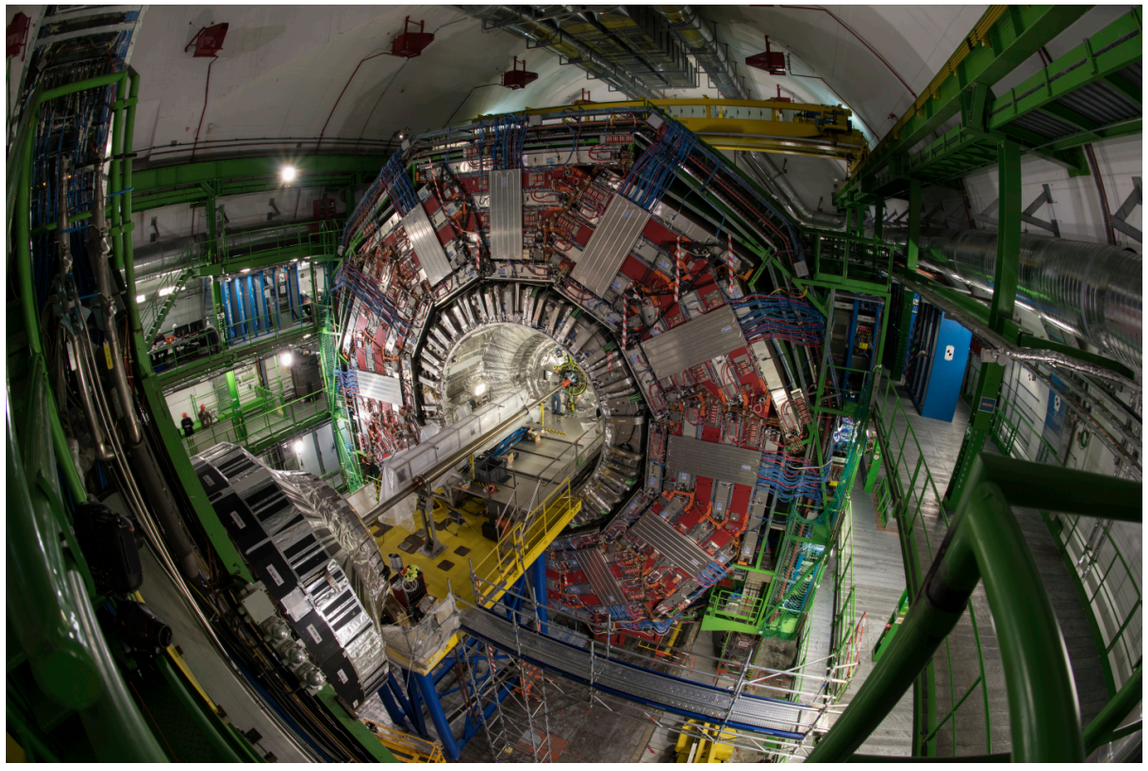

# Why ultrafast



# The LHC big data problem

## ex, Compact Muon Solenoid

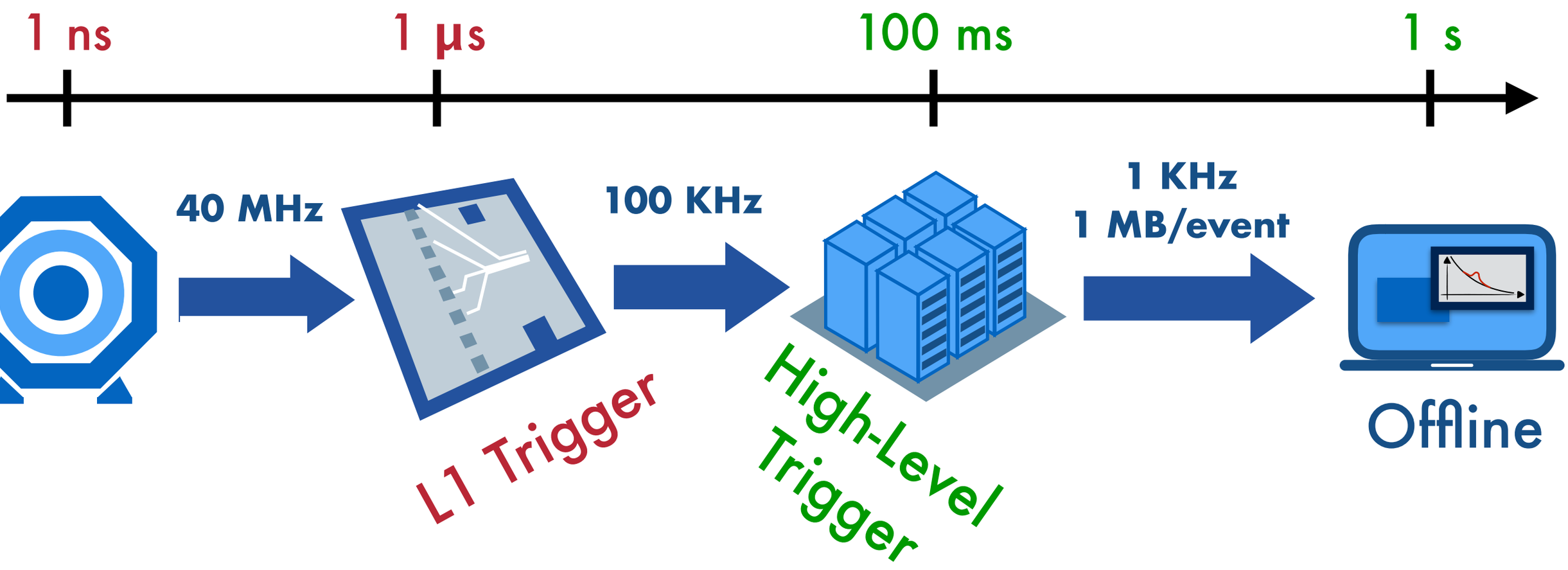


At the LHC proton beams collide at a frequency of 40 MHz
Each collision produces O($10^3$) particles!
The detectors have O($10^8$) sensors used to detect these particles
**Extreme data rates of O(100 TB/s)!**

# The LHC big data problem

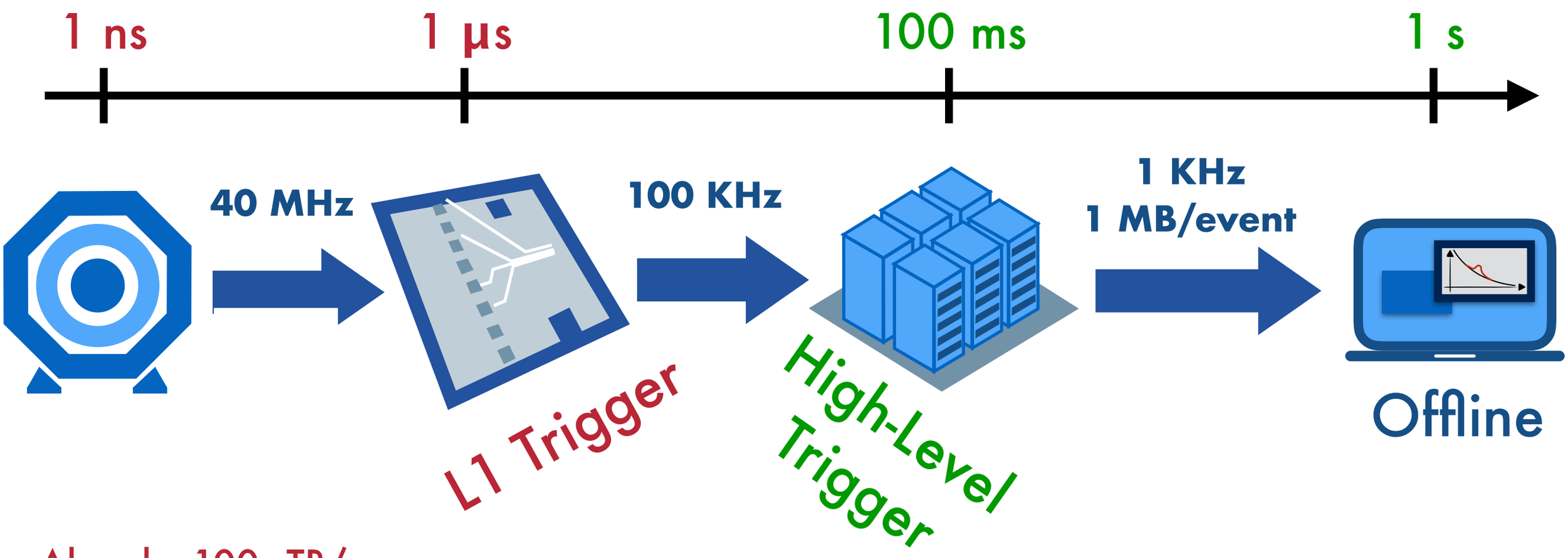
Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:




See Isobel talk

# The LHC big data problem

Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:




Absorbs 100s TB/s

Trigger decision to be made in O(µs)

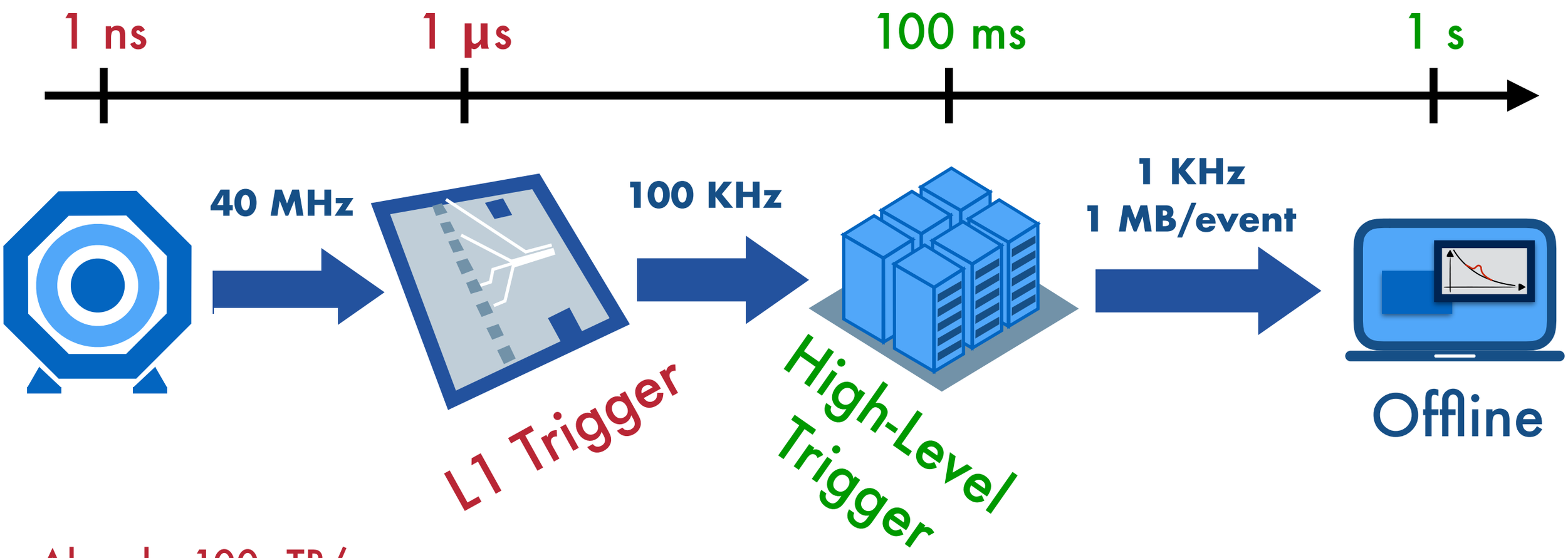
Latencies require all-FPGA design

99.75% events rejected!

See Isobel talk

# The LHC big data problem

Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:




Absorbs 100s TB/s

Trigger decision to be made in O(µs)

Latencies require all-FPGA design

99.75% events rejected!

See Isobel talk

Analysis of the full event runs on commercial computers (30k CPU cores)

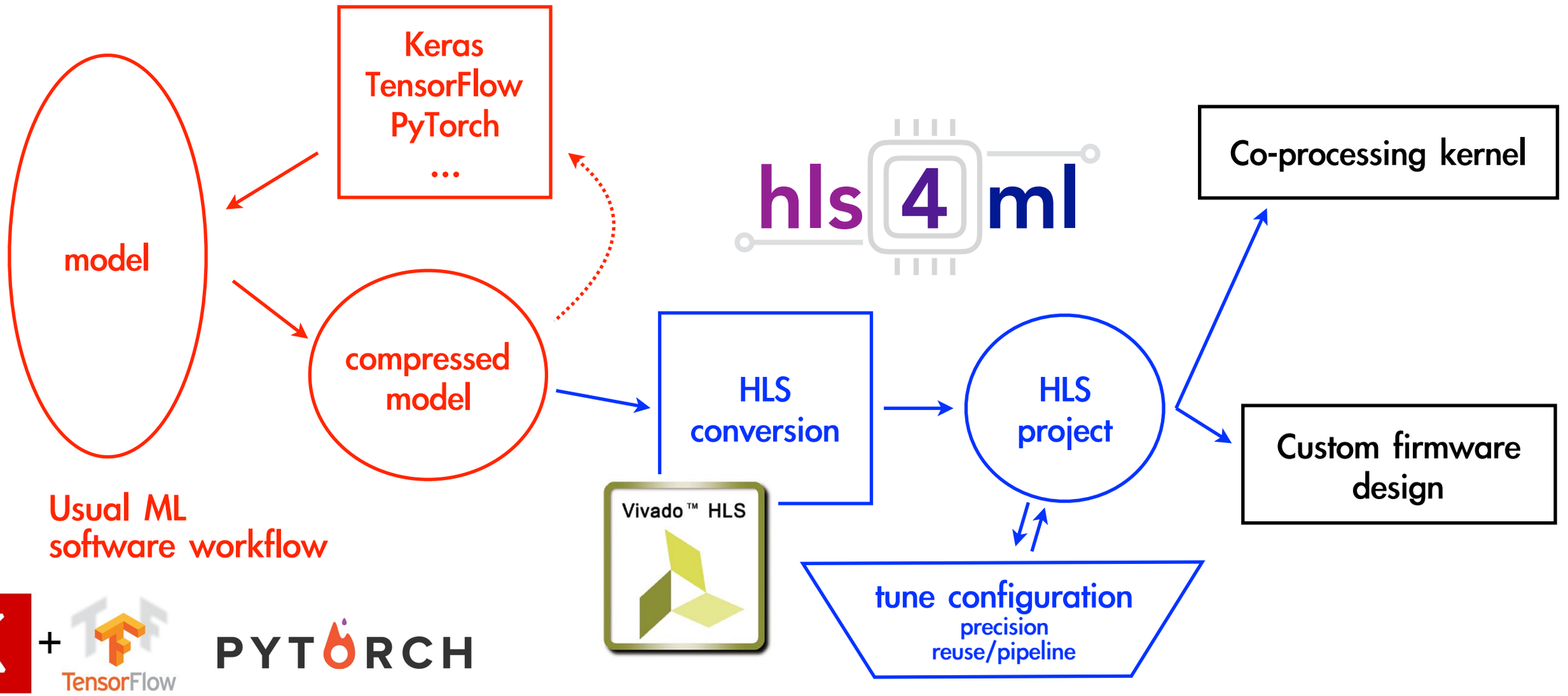
Latency O(100 ms)

99% events rejected!

# hls4ml in a nutshell

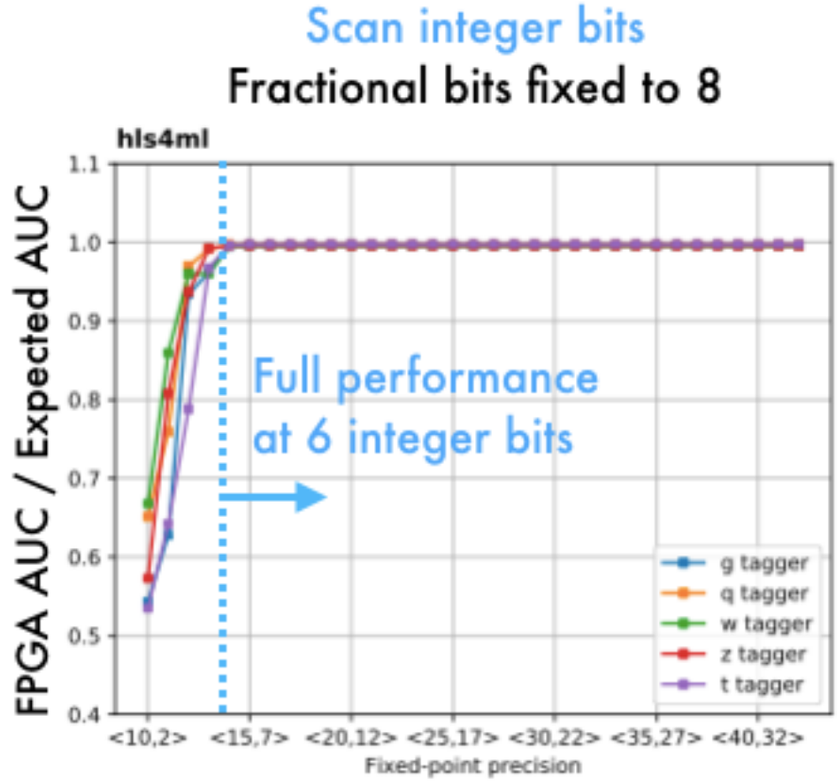
**User-friendly tool to automatically build and optimize DL models for FPGAs:**

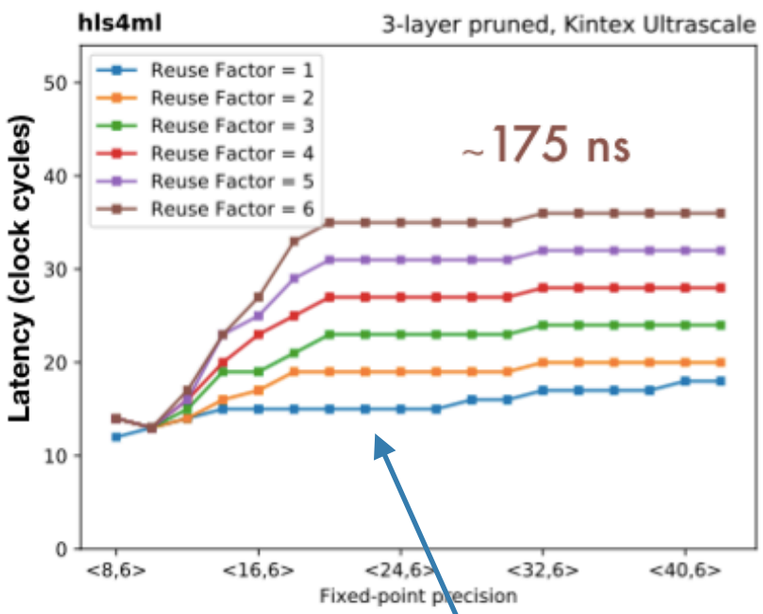
- reads as input models trained with standard DL libraries
- uses Xilinx HLS software (accessible to non-expert, engineers resource not common in HEP)
- comes with implementation of common ingredients (layers, activation functions, binary NN ...)

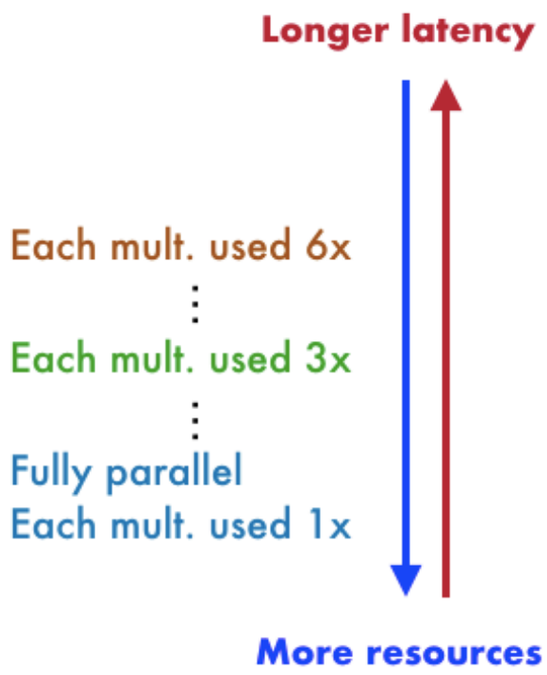


# First results with hls4ml

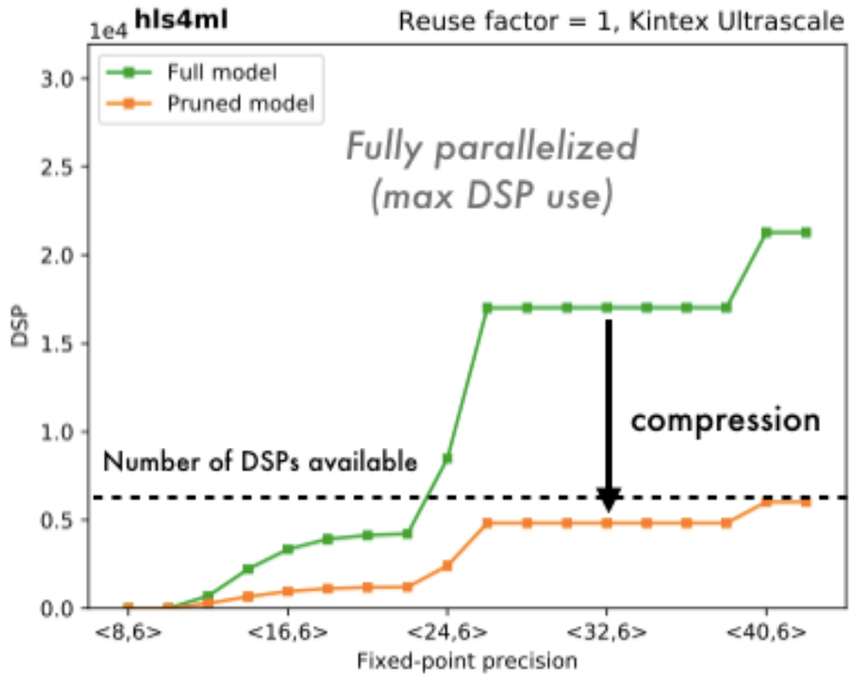
- Demonstrated that by exploiting the high FPGA hardware flexibility we can fit DL solutions in L1 trigger latency:
  - **reduce precision** of calculations to small number of bits
  - **compress** the network by setting to zero unecessary weights
  - exploit maximum **parallelization**




DL algo inference in ~75 ns!




*70% compression ~ 70% fewer DSPs*

# hls4ml: how to

Credits: Vladimir Loncar

- Easy to **install** via `pip`: `git clone … && cd hls4ml && pip install .`
- Easy to **configure** through yaml config file

  `Inputs:` your trained model
  `Precision:` inputs, weights, biases, …
  `ReuseFactor:` how much to parallelize
  `Strategy:`
  `Resource` for large NN
  `Latency` for pipelined-based code for small NN

```
KerasJson: keras/KERAS_3layer.json
KerasH5:   keras/KERAS_3layer_weights.h5
OutputDir: my-hls-test
ProjectName: myproject
XilinxPart: xcku115-flvb2104-2-i
ClockPeriod: 5

HLSConfig:
  Model:
    Precision: ap_fixed<16,6>
    ReuseFactor: 1
    Strategy: Latency #Resource
  LayerName:
    dense1:
      ReuseFactor: 2
      Strategy: Latency #Resource
      Compression: True
```

keras-config.yml

- Easy to **run**:

  Conversion: `hls4ml convert -c keras-config.yml`
  Build: `hls4ml build -p my-hls-test -c -s -r`
  Help: `hls4ml -h / hls4ml command -h`

- Easy to accelerate with CPU+FPGA co-processor systems (ex, Galapagos/SDAccel)

Do not miss the tutorial by Zhenbin, Sioni, Dylan and Javier tomorrow!

# hls4ml: current status

- Since the first results lot of work went into expanding the tools capabilitites
- Supported architectures
  - **MLP**

**NEW:** improved scaling with model size

  - **Binary and Ternary MLP**
    - ultra-low precision: 1- or 2-bits weights with limited loss in performance [*]
    - layer calculation implemented in LUTs
  - **Conv1D/Conv2D (only small)**
    - scale up + Binary/TernaryConv2D coming soon

- Other features: NEW
  - general optimization framework for hls4ml specific model but pluggable (i.e., easy to add your own)
  - Tools for C and RTL simulation results comparison

[*] arxiv.1602.02830, arxiv.1605.04711, ...

# A few applications

- Development of realistic ML models for L1 trigger with hls4ml ongoing
  - replace standard cut-based algorithms with significant reduction of background rates while preserving interesting physics

| **NN Tau Algo being run on 6 candidates every 12 clocks (TMUX 2)** | **NN Tau Algo being run on 6 candidates every 36 clocks (TMUX 6)** |
|---|---|
| Latency: 56 | Latency: 201 |
| II : 12 | II : 36 |
| BRAM :1 (0%) | BRAM :196 (0%) |
| DSP : 1387 (25%) | DSP : 483 (8%) |
| FF : 263k (19%) | FF : 412k (31%) |
| LUT : 408k (61%) | LUT : 336k (50%) |

- Some examples with relatively small NN and/or compressed:
  - triggering on Higgs boson events in topologies overwhelmed by background (VBF Higgs boson→invisible/H→bottom quarks)
  - calorimeter clusters classification (electrons,pions,photons versus background) or energy calibrations
  - identification of tau leptons and muon reconstruction

NN VBF H→bb

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

- Beyond physics: see talk today on hls4ml for developments of new and specialized real time AI systems [*]

[*] Credits: Giuseppe Di Guglielmo


hls4ml

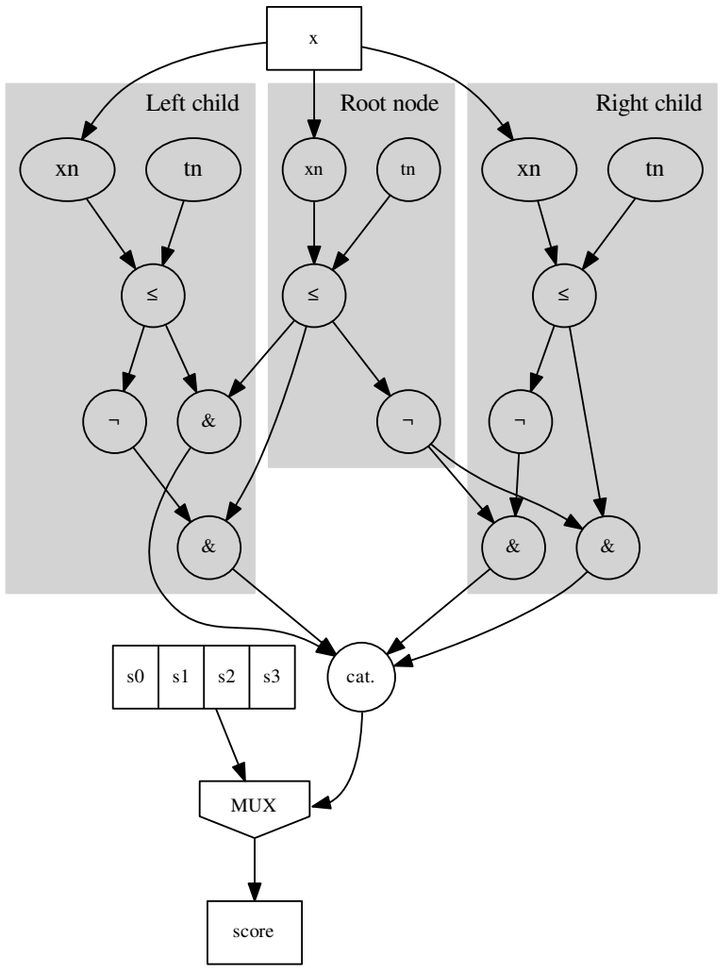
# Ongoing developments

(coming soon)

# BDT on FPGAs

Credits: Sioni Summers

- BDTs have been popular for a long time in HEP reconstruction and analysis
- Suitable for highly parallel implementation in FPGAs
- Implementation in hls4ml optimised for low latency
- No 'if/else' statement in FPGAs → evaluate all options and select the right outcome
  - compare all features against thresholds, chain together outcomes to make the 'tree'
- Text for model with 16 inputs, 5 classes, 100 trees, depth 3 on VU9P FPGA:
  - 4% LUTs, 1% FFs (0 DSPs, 0 BRAMs)
  - 25 ns latency with II=1

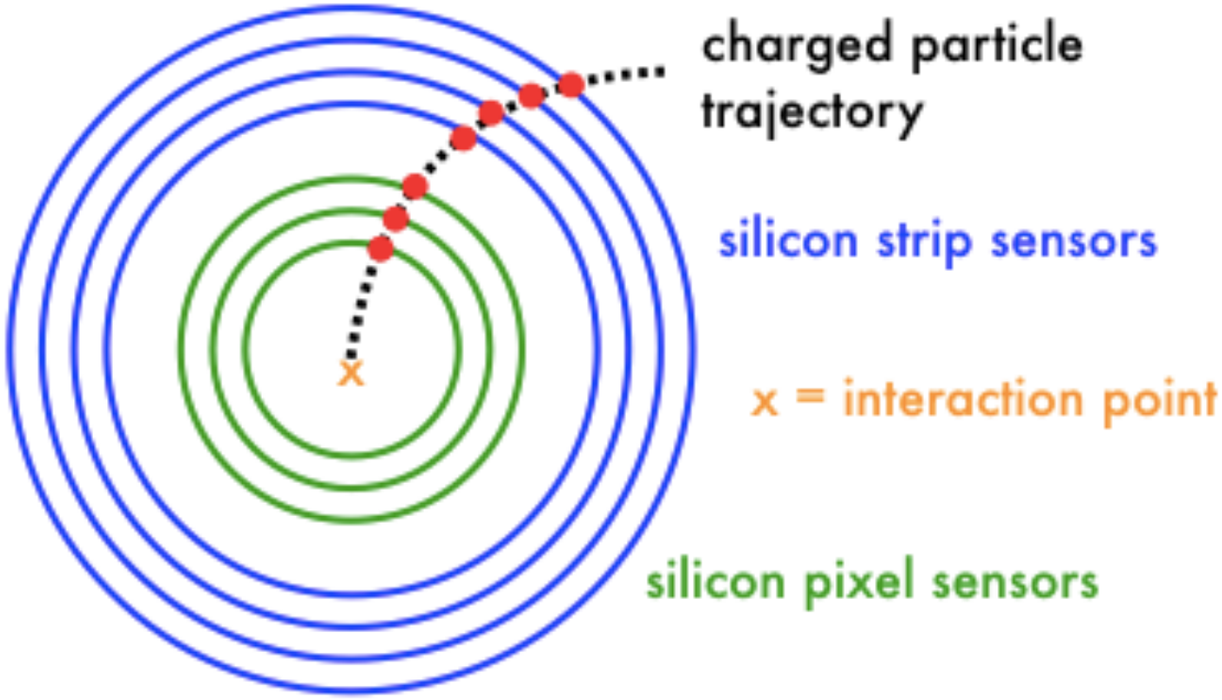


# Graph NN on FPGA

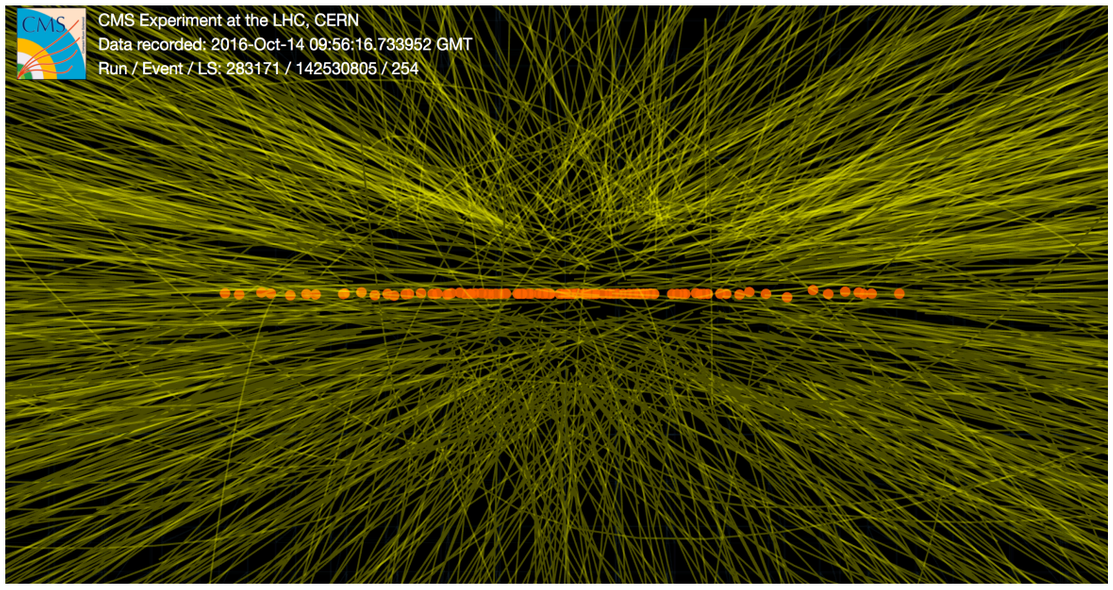
Reconstructing the trajectories of charged particles passing the detector is really like "connecting the dots"

Graph NNs seem a natural solution for this task

This sketch: 1 track

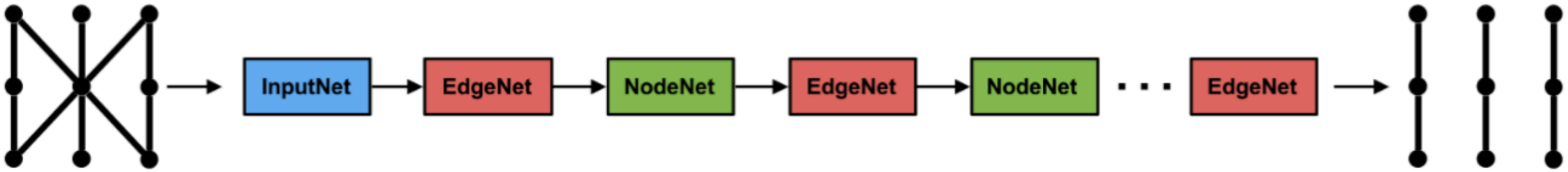


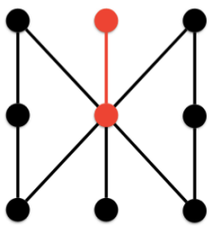

This real collision: thousands of tracks!



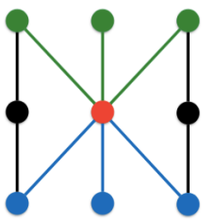
# Graph NN on FPGA

**Model:** binary classification on the edges of the graph todistinguish true hit pairs
based on HEP.TrkX GNN v1 architecture [arXiv:1810.06111]



**Edge network** uses the node features to compute edge weights



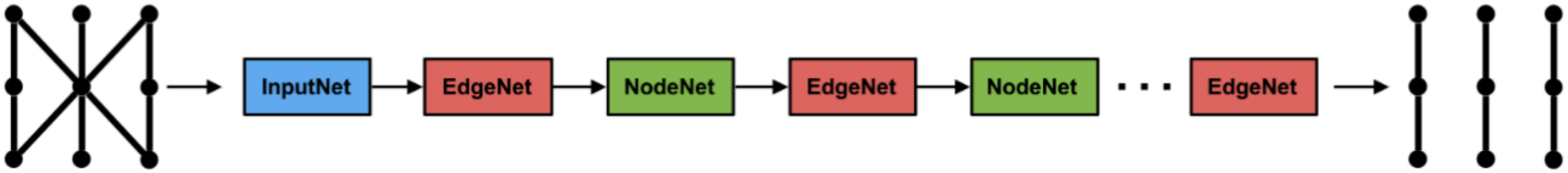
**Node network** aggregates forward and backward node features with the edge weights and updates node features

With each iteration, the model propagates information through the graph, strengthens important connections, and weakens useless ones.

Slide from Steve Farrell

# Graph NN on FPGA

**Model:** binary classification on the edges of the graph to find true hit pairs
based on HEP.TrkX GNN v1 architecture [arXiv:1810.06111]




**Preliminary implementation in hls4ml under test**:

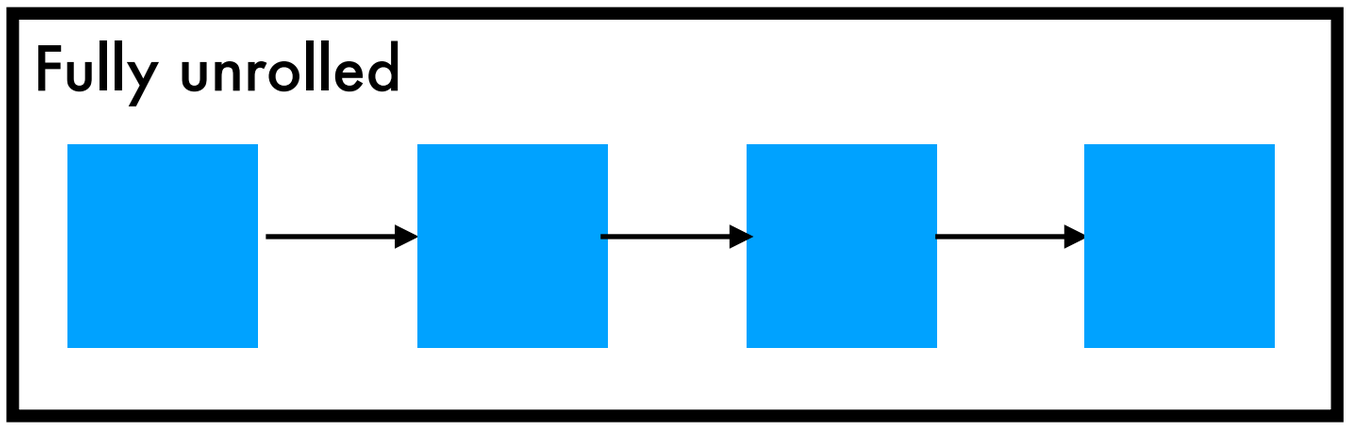
- use default MLP and activation function implementations but applied to each row of the input matrix
- develop new functions to do concatenations and special (binary/sparse) matrix multiplications for edge-node associaciation matrices

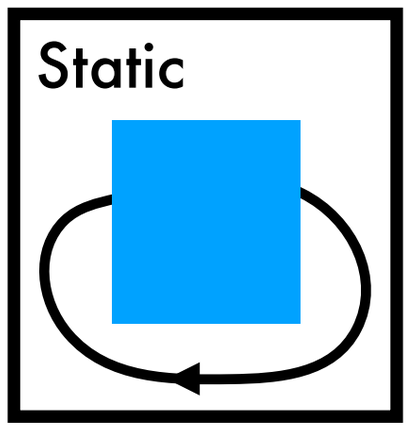
Succesfully tested a small example with 4 tracks, 4 layers, no iteration
→ major effort now to scale this up
To be automatize in hls4ml keras-to-hls conversion tool with custom model

Credits: Javier Duarte, Kazi Ahmed Asif Fuad

# RNN on FPGA

Credits: Phil Harris




- Two RNN implementations in hls4ml under test:
  - **Fully unrolled:** latency optimized with II=1 possible but large resource usage
  - **Static:** same resources used for weights and multiplications
    - Latency is slower and II limited to clock time for each layer (small network its 10 clk)
    - However N (N=latency of layer) copies can go through at the same time
- Supported network architectures: simple RNN, LSTM, GRU
- Works for small network cases → now scale it up!

# Conclusions

- Other future developments in hls4ml:
  - Autoencoders
  - Other graph NN architectures (GarNet, …)
  - Alternate HLS implementations (Intel/Altera, Mentor…)
  - Integration of co-processor acceleration projects in hls4ml
  - hls4ml for multi-FPGA (inference & training)
  - Inference engine for CPUs based on hls4ml (for CMS software)

- For more info about hls4ml:
  - Tutorials by Zhenbin, Sioni, Dylan and Javier tomorrow!
  - https://fastmachinelearning.org/hls4ml/
  - https://github.com/hls-fpga-machine-learning
  - https://arxiv.org/abs/1804.06913

# Thank you!

Resources Blog Journalists

CISION PR Newswire

News Products Contact

News in Focus Business & Money Science & Tech Lifestyle & Health Policy & Public Interest People & Culture

## Zenuity and CERN Team up on Fast Machine Learning for Autonomous Driving

ZENUITY

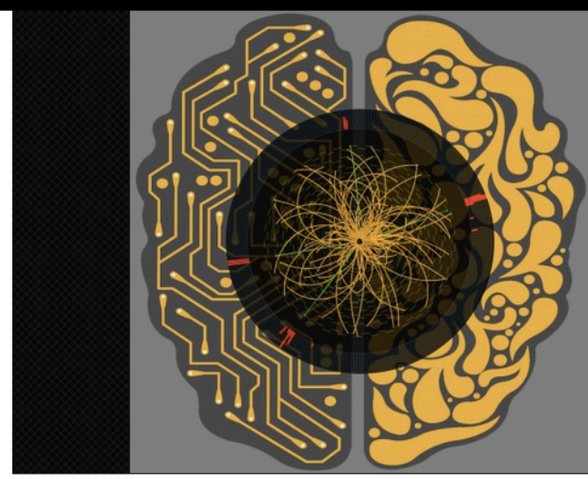
SHARE THIS ARTICLE

NEWS PROVIDED BY
Zenuity →
29 Aug, 2019, 07:00 BST

## MIT News

ON CAMPUS AND AROUND THE WORLD

Browse or Search



FULL SCREEN

Artificial intelligence interfaced with the Large Hadron Collider can lead to higher precision in data analysis, which can improve measurements of fundamental physics properties and potentially lead to new discoveries.

Image: FermiLab

Boosting computing power for the future of particle physics

## Artificial Intelligence Accelerates Dark Matter Search

Integrating Inference Acceleration with Sensor Pre-processing in Xilinx FPGAs Delivers Performance Unachievable by GPUs and CPUs

**AT A GLANCE:**

**Customer:** High energy physics researchers from an association of leading international institutions (CMS Institute) conducting experiments at the European particle physics laboratory, CERN.

**Industry:** Scientific Research

**Employees:** CMS Institute has more than 4,000 global scientific collaborators representing 200 ...tes and universities from more than 40

Uber Nobelius

Salva

Commenta

ALGORITMI

## Auto a guida autonoma, Zenuity collaborano sull'apprendimento veloce

La partnership tra l'organizzazione per la ricerca nucleare e la società svedese punta ad accelerare l'apprendimento veloce e quindi i tempi di reazione sulla sicurezza

di A.Mac.

## D'enregistrer les collisions à les éviter

**Comment des techniques d'apprentissage automatique développées au CERN pourraient améliorer la technologie des véhicules autonomes**

29 AOÛT, 2019 | Par Kate Kahle