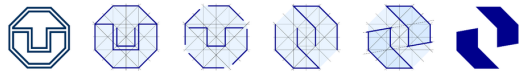
Real-time signal reconstruction using CNNs on FPGAs for the ATLAS LAr calorimeter readout

PUNCH4NFDI TA3 WP3 meeting

Johann C. Voigt - TU Dresden

22 September 2025

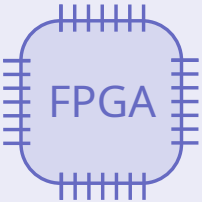


INSTITUTE OF
NUCLEAR AND
PARTICLE PHYSICS

With funding from the:
Federal Ministry
of Research, Technology
and Space

Why do we use FPGAs?

What is an FPGA?

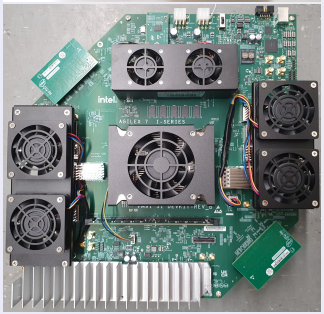


- Chip that is reconfigurable after manufacturing (*in the field*)
- Functional blocks: Logic cells, registers, RAM blocks, multipliers (digital signal processors/DSPs)
- Periphery: network transceivers, ...
- Flexible interconnect: Arbitrarily connect functional blocks

- High data throughput using pipelining, fixed-latency possible
- Good connectivity with high number of optical links
- Reconfigurability decreases project risk compared to ASICs
- Lower price compared to ASICs in low quantities
- Firmware can be simulated, but debugging more complex than for software

What do FPGAs look like in practice?

Development kit



Custom board

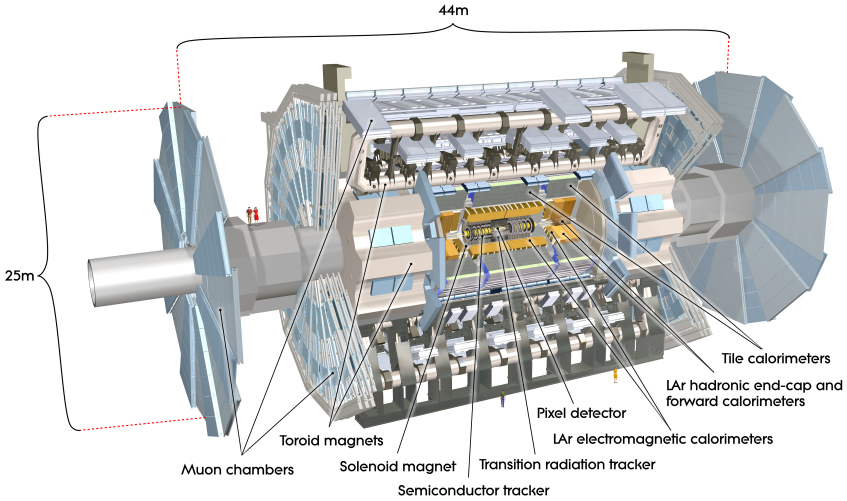
Data center card

https://www.amd.com/en/products/accelerators/alveo/u55c/a-u55c-p00g-pq-g.html [1]

Why ML on FPGA

- FPGAs increasingly used in fast/low latency readout systems
- ML algorithms promise performance increase
  - Better selection of interesting physics events decreases required storage or increases data that can be recorded
- Modern FPGAs offer enough resources for ML applications
  - DSP blocks ideal for multiply-accumulate operations

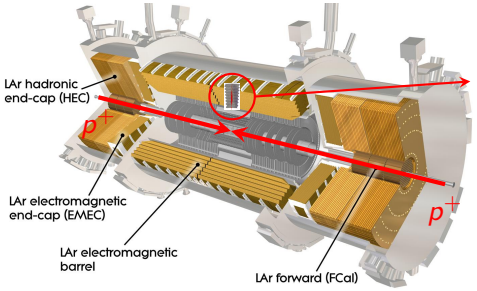
The ATLAS experiment

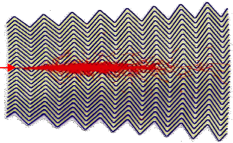


https://cds.cern.ch/images/CERN-GE-0803012-01 [2]

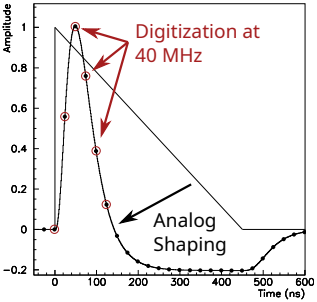
ATLAS LAr calorimeter

- Starting in 2030 upgraded Large Hadron Collider will provide ≈ 140 proton-proton collisions per bunch crossing (BC) $\widehat{=}$ every 25 ns $\widehat{=}$ 40 MHz
- Higher pileup and higher trigger rate require replacement of LAr calorimeter electronics
- 235 Tbit s^{-1} data stream from LAr calorimeters alone



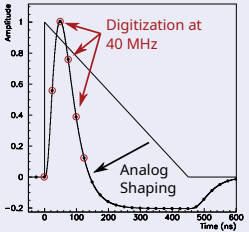


$\approx 182\,000$ detector cells



Left: https://cds.cern.ch/record/1095928 [3], Middle: https://www.particles.uni-freiburg.de/dateien/vorlesungsdateien/particledetectors/kap8 [4], Right: https://cds.cern.ch/record/331061 [5]

Digital energy reconstruction



https://cds.cern.ch/record/331061 [5]

182k cells
@ 40 MHz

Digital energy reconstruction (HL-LHC)

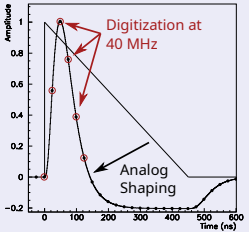
FPGA

- 556 Intel Agilex-7 FPGAs for real-time processing (off-detector)
- Optimal Filter (OF)

$$E_t = \sum_{i=1}^{5} c_i \cdot x_{t+i}$$

- Evaluating artificial neural networks (ANNs) for cell level energy reconstruction

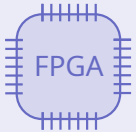
Digital energy reconstruction



https://cds.cern.ch/record/331061 [5]

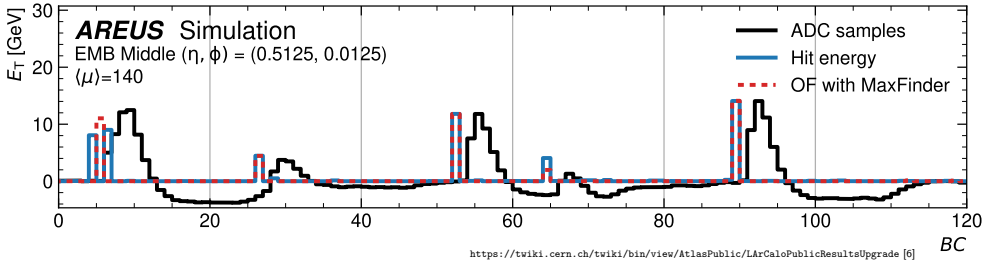
182k cells
@ 40 MHz

Digital energy reconstruction (HL-LHC)



- 556 Intel Agilex-7 FPGAs for real-time processing (off-detector)
- Optimal Filter (OF)

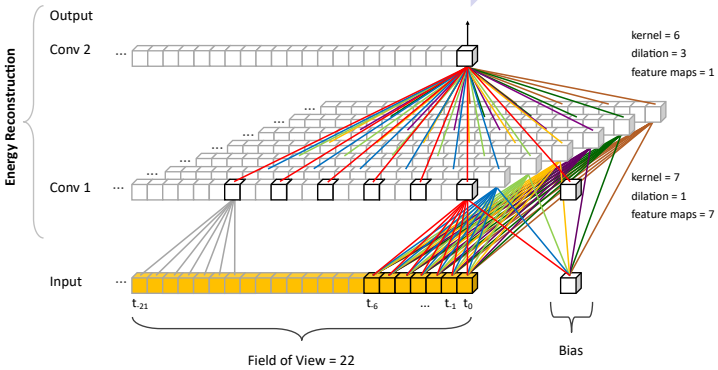
$$E_t = \sum_{i=1}^{5} c_i \cdot x_{t+i}$$



https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

Convolutional neural network architecture (CNN)

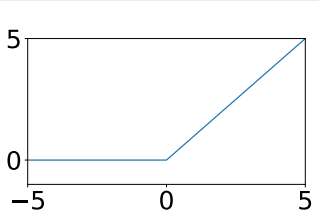
Energy prediction for every BC



https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

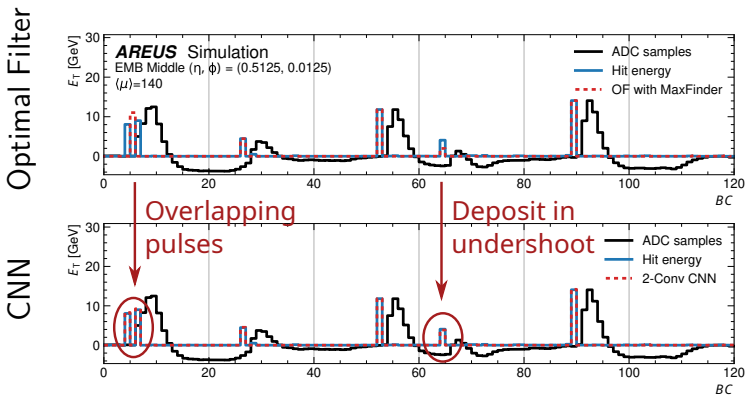
Continuous input from one detector cell

ReLU activation



- Two convolutional layers
- 100 to 400 parameters
- 22 BC field of view

Example sequence

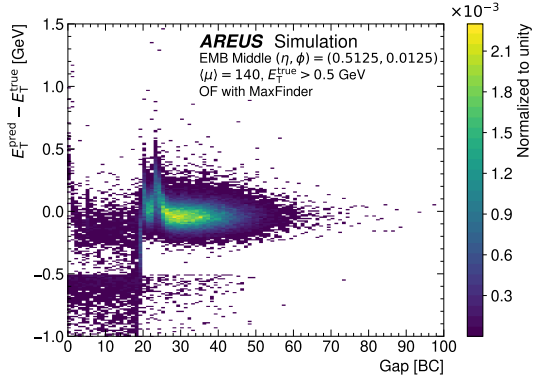


https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

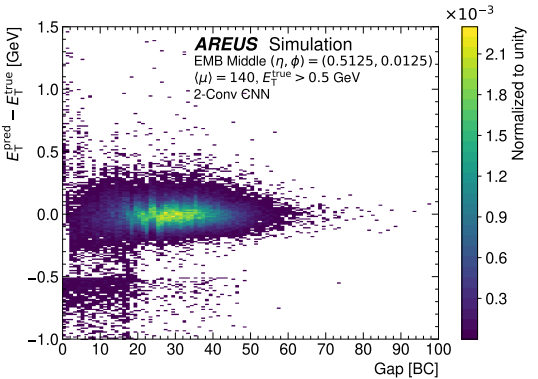
- Input: Simulated detector sequences
  - Pileup from detector simulation
  - Uniform energy distribution of injected signal
- True energy available as training target
- CNNs trained in Python using Tensorflow/Keras framework
- Optimal Filter/CNN output compared to true energy

- CNNs show improved performance for overlapping pulses

Energy reconstruction performance as a function of gap between 2 pulses



Optimal Filter

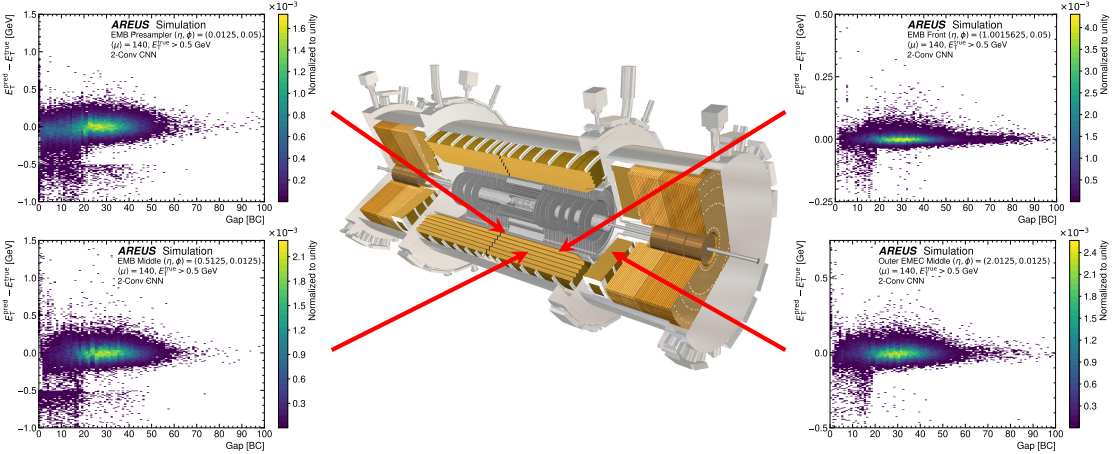


2-Conv CNN

- Improvements in reconstruction of overlapping pulses (gap < 20 BC)

https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

Performance for different detector regions



- Same architecture trained for different detector regions
- Identified clusters of similar cells based on pulse shapes for future trainings

https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6], https://cds.cern.ch/record/1095928 [3]

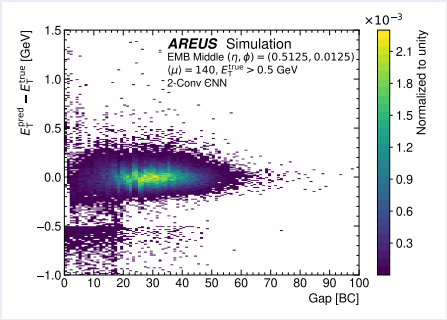
Pipeline for network training

- Cannot train one network for each detector cell individually
  - Clustering of similar cells based on pulse shape (23 barrel region clusters for now)
- Training not perfectly reproducible
  - Using Hyperband algorithm to initialize large number of networks and select best one
- Can submit training to Slurm HPC system for all clusters in parallel
- Problems with low GPU utilization
  - Shape of input data suboptimal for speed (sequences of length 10000 samples)
  - Increasing batch size degrades energy resolution and less trainings converge
  - Using 32 parallel steps improves performance without degrading resolution
  - No success with declaring logical GPUs to run multiple trainings on same GPU in parallel
  - Can use $\frac{1}{7}$ fractional GPU using Nvidia MIG mechanism provided by HPC cluster

Different concepts of networks

Continuous reconstruction

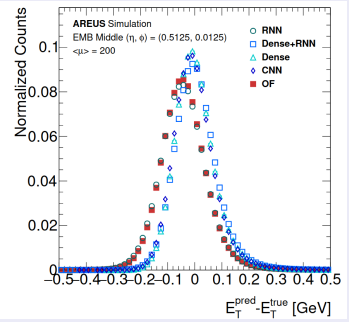
- Valid energy prediction at every BC
- Trained to deal with overlapping signals
- Energy resolution for isolated hits similar to OF



https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

Triggered networks

- Valid energy prediction only when externally triggered
- Better energy resolution for isolated hits
- ANNs reach better resolution and smaller bias than OF

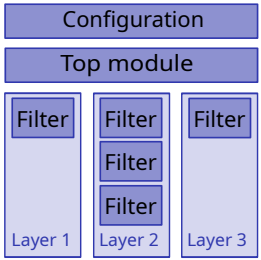


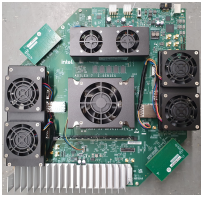
Requirements for the CNN architecture and firmware

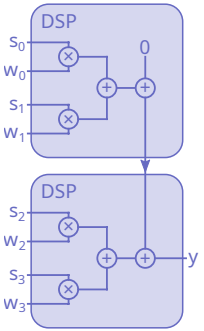
Constraint	**Architecture/Training**	**Firmware**
Low latency ($\approx 150\,\text{ns}$)	Limited number of layers	Latency optimized pipeline
384 cells per FPGA @ 40 MHz	Only 400 parameters quantization aware training	Resource optimization, quantization
Single compiled firmware for all FPGAs	Fixed architecture, no pruning	Weights configurable via network interface
Intel FPGA	Quantization to 18 bit	Limited choice of available ML firmware frameworks

CNN firmware implementation

- CNN inference implemented in VHDL
- Model architecture configurable and automatically extracted from Keras/QKeras output files
- Support multiplexing
  - Design runs at 12× ADC frequency
  - Cyclically processes serialized stream from 12 detector cells
- Development for Intel Agilex-7 FPGA
  - Calculation in 18 bit fixed point numbers
  - DSP can be chained for multiply-accumulate operations
  - Structure of FPGA directly considered in firmware design

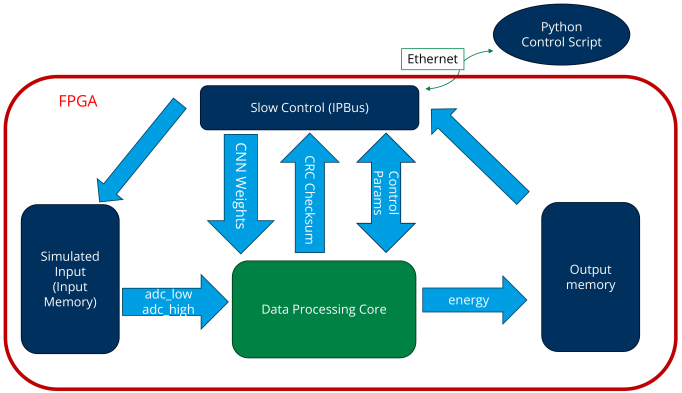


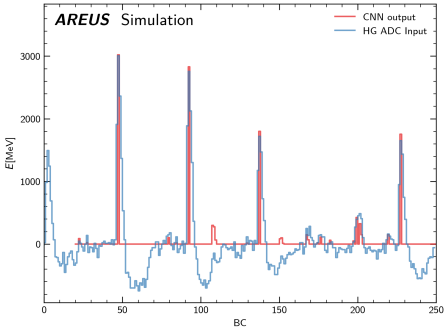




Testing firmware on development kit

- Wrap firmware containing CNNs with input and output memory
- Configure input data, CNN weights, general configuration and load back results to PC using IPbus





FPGA resource estimation

- Latency requirement by ATLAS trigger of $\approx 150\,\text{ns}$ met
- Can process required number of 384 detector cells
  - 12-fold multiplexing with 33 parallel instances
- Resource estimates based on Intel Quartus reports

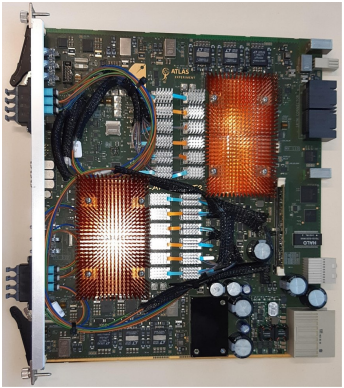
Network	Multiplex.	Detector cells	$f_{\max}$	Logic (ALMs)	DSPs
CNN (100 param.)	12	396	539 MHz	4 %	13 %
CNN (400 param.)	12	396	510 MHz	19 %	50 %

General recommendations

- Finishing summary document with recommendations about ML on FPGA
- hls4ml offers good solution for most situations and is good starting point
  - Other options: Vitis AI, FINN, Conifer, ...
- Constraints from project may require more custom solutions
  - High level synthesis solutions for easier implementation and better maintainability
  - VHDL/Verilog for direct control over resource allocation
- Implementation needs to be embedded into firmware project
  - Software for simulation and synthesis (and potentially build system like hdlmake)
  - Core library, e.g. Colibri
  - Verification, e.g. UVM, UVVM, OSVVM, Cocotb

Summary and outlook

- CNNs outperform Optimal Filter in real-time energy reconstruction
  - Tradeoff between reconstruction of overlapping and isolated signals
  - Trained network for clusters of cells covering barrel region of calorimeter
  - First studies on effects of new cell energy reconstruction on photon, electron and jet measurements ongoing
- VHDL implementation of CNNs with low latency available
- CNNs fit target FPGA and run at required clock frequency
- Tested on FPGA devkit, but not fully integrated with other firmware components yet



Sources I

[1] AMD. *AMD Alveo U55C High Performance Compute Card*. 2025. URL: https://www.amd.com/en/products/accelerators/alveo/u55c/a-u55c-p00g-pq-g.html (visited on 06/27/2025).

[2] Joao Pequenao. *Computer generated image of the whole ATLAS detector*. CERN, ATLAS. Feb. 27, 2015. URL: https://cds.cern.ch/images/CERN-GE-0803012-01 (visited on 08/06/2020).

[3] Joao Pequenao. *Computer generated image of the ATLAS Liquid Argon*. CERN. Mar. 27, 2008. URL: https://cds.cern.ch/record/1095928 (visited on 03/29/2021).

[4] Karl Jakobs. *Particle Detectors 2015. Chapter 8*. 2015. URL: https://www.particles.uni-freiburg.de/dateien/vorlesungsdateien/particledetectors/kap8 (visited on 03/21/2024).

Sources II

[5] LHC Experiments Committee, LHCC. *ATLAS liquid-argon calorimeter: Technical Design Report*. Technical design report. ATLAS. Geneva: CERN, 1996. URL: https://cds.cern.ch/record/331061 (visited on 04/06/2021).

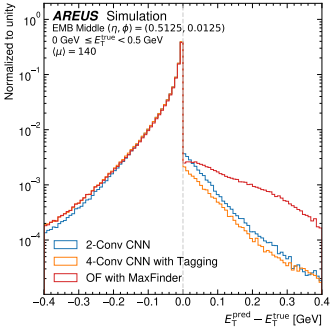
[6] ATLAS LAr Calorimeter Group. *Public Liquid Argon Calorimeter Plots on Upgrade*. URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade.

[7] ATLAS Collaboration. *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.2LBB.4IAL. URL: https://cds.cern.ch/record/2285584 (visited on 06/30/2025).

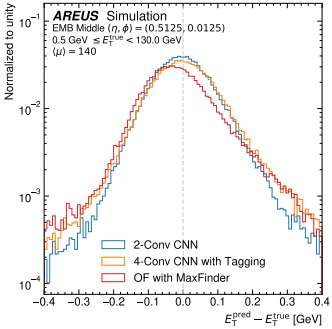
[8] J. Duarte et al. "Fast inference of deep neural networks in FPGAs for particle physics". In: *Journal of Instrumentation* 13.07 (July 2018), P07027–P07027. DOI: 10.1088/1748-0221/13/07/p07027. (Visited on 03/29/2021).

Distribution of deviation from true energy

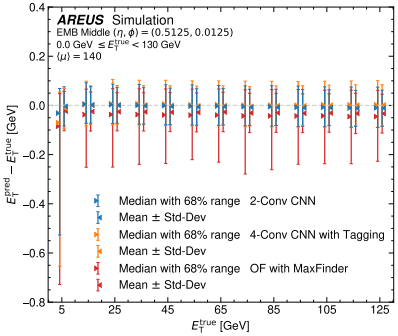
$E_{\mathrm{T}}^{\mathrm{true}} < 0.5\,\mathrm{GeV}$



$E_{\mathrm{T}}^{\mathrm{true}} \geq 0.5\,\mathrm{GeV}$

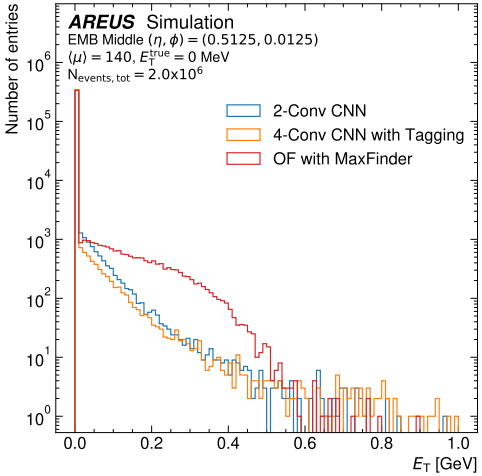


Median/Mean over $E_{\mathrm{T}}^{\mathrm{true}}$ range



https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

Prediction in BCs without energy deposit

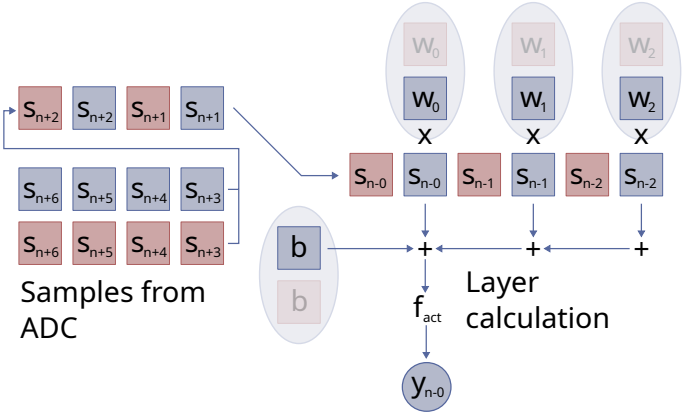


https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [6]

CNN multiplexing concept

- One FPGA needs to fit 33 CNN instances
- Each instance uses 12× multiplexing
  → Design needs to run at 12× the ADC frequency: 480 MHz

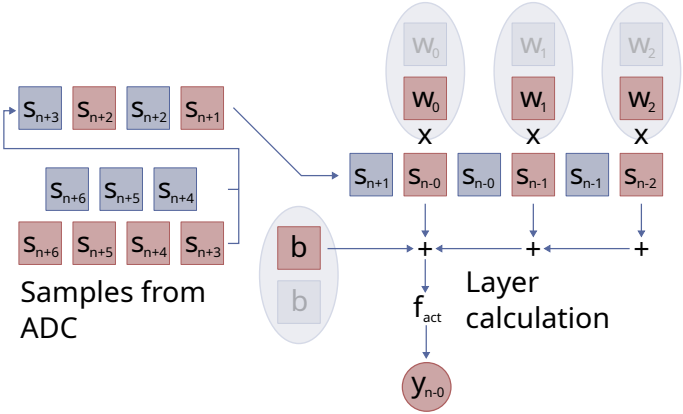
Example for two ADCs:



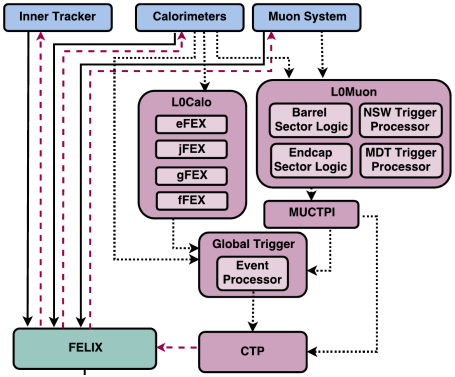
CNN multiplexing concept

- One FPGA needs to fit 33 CNN instances
- Each instance uses 12× multiplexing
  → Design needs to run at 12× the ADC frequency: 480 MHz

Example for two ADCs:



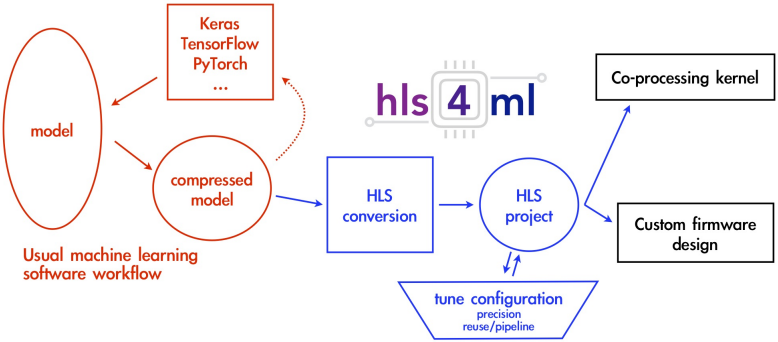
Data flow to the ATLAS trigger



https://cds.cern.ch/record/2285584 [7]

- First trigger level implemented in hardware (e.g. FPGAs)
- Strict latency budget due to limited buffering capabilities of readout
- Specialized trigger modules for different subdetectors
- ML algorithms improve trigger efficiency

hls4ml



10.1088/1748-0221/13/07/p07027 [8]

- Automatic conversion from trained models to HLS code
- Final translation into firmware depends on vendor tools
- Support for typical optimization techniques like pruning and quantization

- Supports growing number of ANN architectures, ML frameworks in frontend and HLS languages in backend
- Other tools exist, but hls4ml offers broadest compatibility and feature set