Artificial neural network inference on FPGAs in PUNCH4NFDI

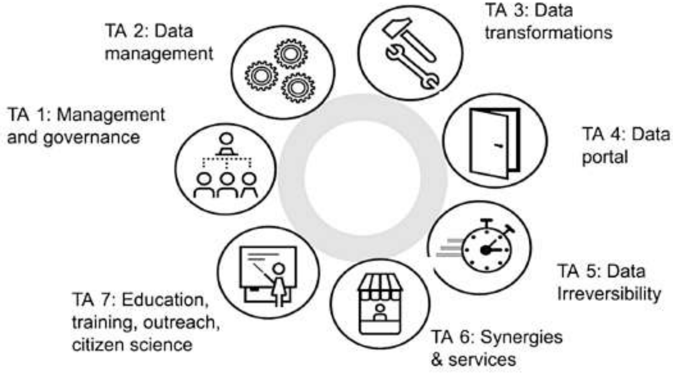
6th FAIRmat Users Meeting

Johann C. Voigt on behalf of PUNCH4NFDI TA5

3 July 2025

PUNCH 4NFDI TECHNISCHE UNIVERSITÄT DRESDEN INSTITUTE OF NUCLEAR AND PARTICLE PHYSICS

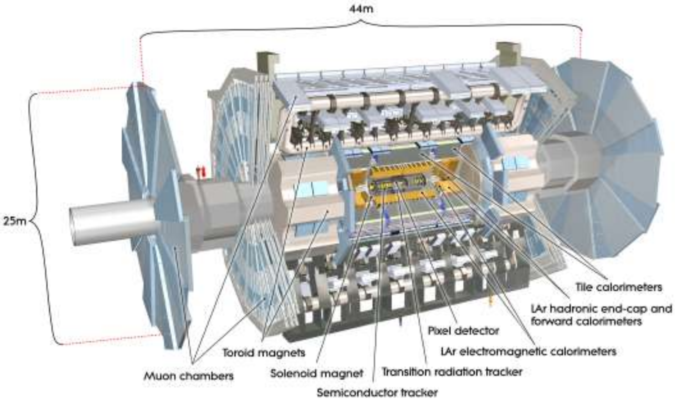
PUNCH4NFDI


TA 1: Management and governance
TA 2: Data management
TA 3: Data transformations
TA 4: Data portal
TA 5: Data Irreversibility
TA 6: Synergies & services
TA 7: Education, training, outreach, citizen science

https://www.punch4nfdi.de/about/task_areas/ [1]

- NFDI consortium for particle physics and astrophysics
- Main goal: Development of a FAIR science data platform to manage full lifecycle of research data
- TA5: Data irreversibility: Unavoidable loss of data if volume is too large
    - Intelligent filtering in real-time

Where our data comes from


44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

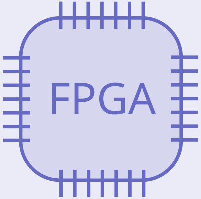
High energy particle physics: ATLAS experiment

Radioastronomy: (Future) Square Kilometre Array

Left: https://cds.cern.ch/images/CERN-GE-0803012-01 [2] Right: https://skao.canto.global/v/SKAOMediaKit/album/OMQJL?viewIndex=2&column=image&id=1chq7ohjc10ah3k4e0dm15mj7k [3]

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

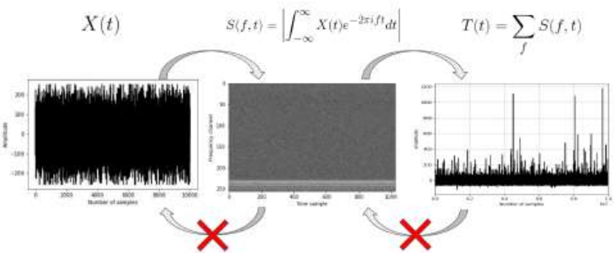
https://www.amd.com/en/products/accelerators/alveo/u55c/a-u55c-p00g-pq-g.html [4]

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
  - Specialized models with AI accelerators

Analysis of pulsar data from radio telescope

Data formats


$X(t)$
$S(f,t) = \left| \int_{-\infty}^{\infty} X(t) e^{-2\pi i f t} dt \right|$
$T(t) = \sum_f S(f,t)$

Example data set (21 min):

Baseband data (1.6 TB)

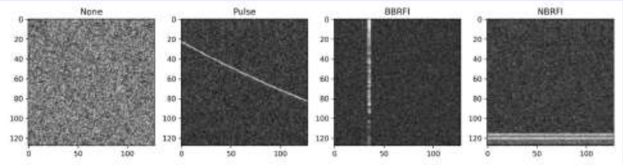
Spectrogram (3 GB)

Time series (49 MB)

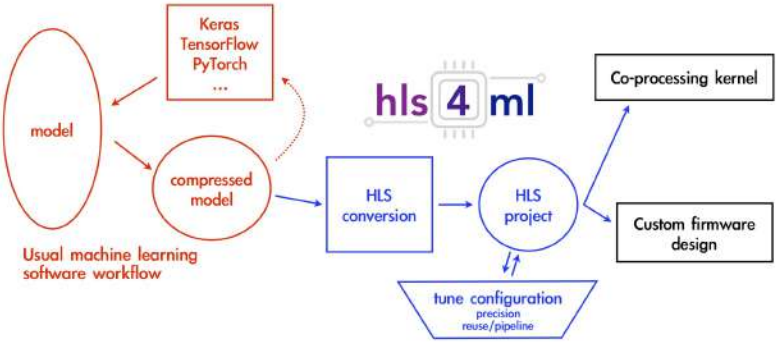
- Goal: Only store baseband data when signal is detected
- Use minimal pre-processing
  - Train ANN to classify spectrograms

https://indico.desy.de/event/45348/contributions/173482/ [5]

8 classes of signal and RFI



hls4ml



10.1088/1748-0221/13/07/p07027 [6]

- 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

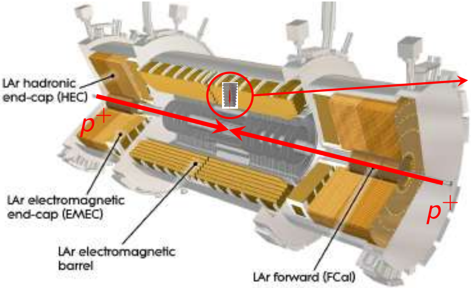
Workflow

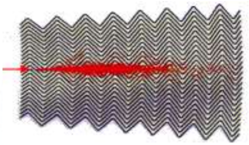
- Using FPGA as accelerator:
    - Host CPU manages the process
    - Shared High Bandwidth Memory (HBM) to transfer data to FPGA
    - AI kernel on FPGA (and data stream management)
    - Transfer results back via HBM
- AI kernel generated using hls4ml
- Tested with simplified example network: 2D CNN with 4572 parameters, results match expectation (can be tested out at `https://github.com/ypmen/punch_workshop` [7])
- Latency estimate $\approx 1\,\mathrm{ms}$ with 72 MHz maximum clock frequency
- Resource utilization estimates:

Resource utilization	BRAM 18k	DSPs	Registers	Logic (LUT)
Pre-synthesis	4 %	7 %	2 %	16 %
Post-synthesis	11 %	7 %	9 %	16 %

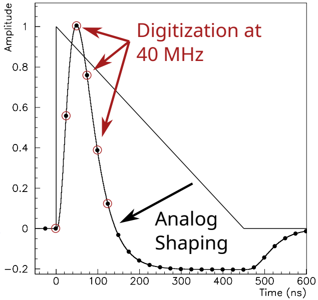
ATLAS LAr calorimeter

- Upgraded Large Hadron Collider will provide ≈ 140 proton-proton collisions per bunch crossing (BC) $\widehat{=}$ every 25 ns $\widehat{=}$ 40 MHz
- ATLAS experiment detects collision products using layered sub-detectors
- Higher pileup and higher trigger rate require replacement of LAr calorimeter electronics
- 235 Tbit s^{-1} data stream from LAr calorimeters alone


LAr hadronic end-cap (HEC)
p^+
LAr electromagnetic end-cap (EMEC)
LAr electromagnetic barrel
LAr forward (FCal)
p^+



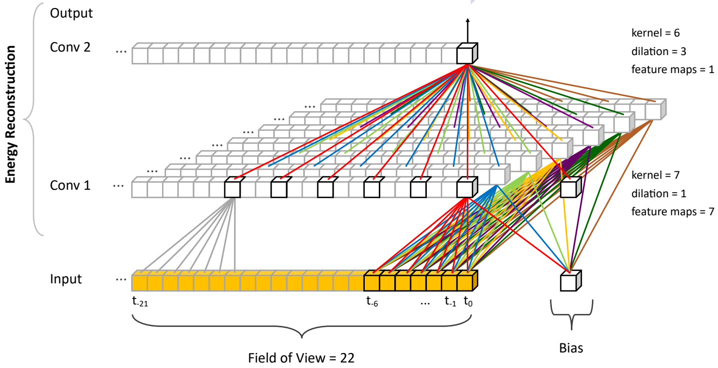
≈ 182 000 detector cells



Left: https://cds.cern.ch/record/1095928 [8], Middle: https://www.particles.uni-freiburg.de/dateien/vorlesungsdateien/particledetectors/kap8 [9], Right: https://cds.cern.ch/record/331061 [10]

Convolutional neural network architecture (CNN)

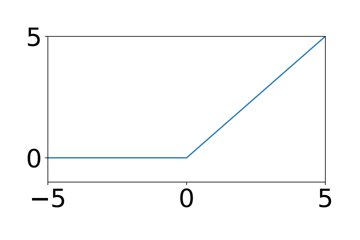
Energy prediction for every BC



https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [11]

Continuous input from one detector cell

ReLU activation

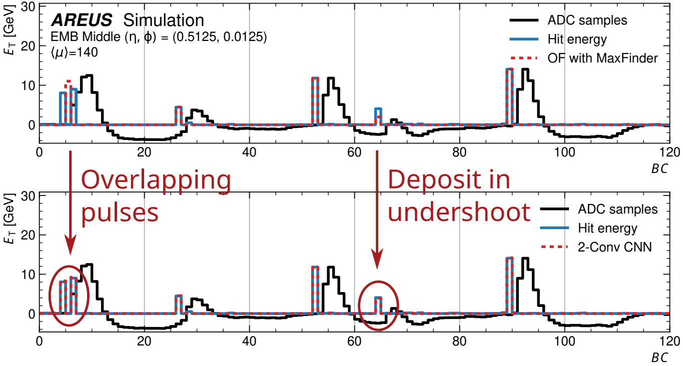


- Two convolutional layers
- 100 to 400 parameters
- 22 BC field of view

Example sequence

Optimal Filter

CNN



Overlapping pulses

Deposit in undershoot

https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LArCaloPublicResultsUpgrade [11]

- Input: Signal enriched simulated detector sequences
- True energy available as training target
- CNNs trained in Python using Tensorflow/Keras framework
- Optimal Filter/CNN output compared to true energy

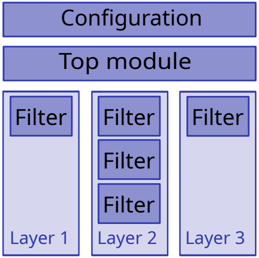
▶ CNNs show improved performance for overlapping pulses

Requirements for the CNN architecture and firmware

Constraint	**Architecture/Training**	**Firmware**
Low latency ($\approx 150\,\mathrm{ns}$)	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 output files
- Support multiplexing
  - Design runs at 12× ADC frequency
  - Cyclically processes 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



Configuration

Top module

Filter

Layer 1

Filter

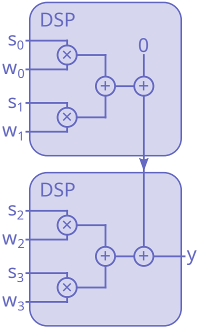
Filter

Filter

Layer 2

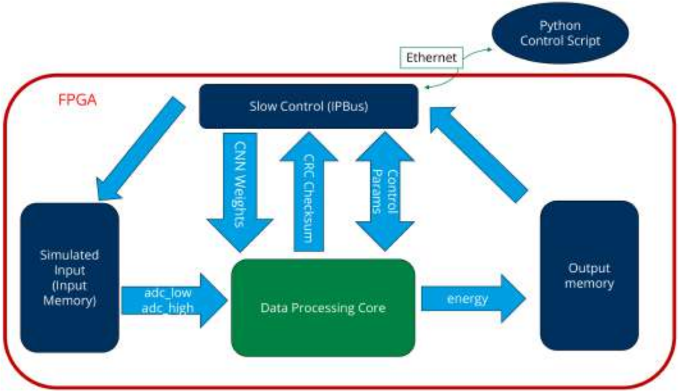
Filter

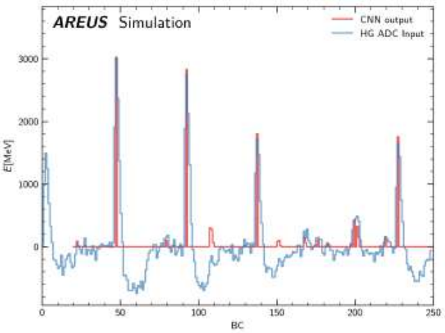
Layer 3



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



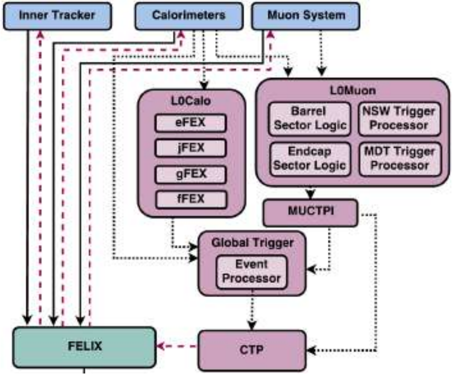


FPGA resource estimation

- Latency requirement by ATLAS trigger of ≈ 150 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 %

Data flow to the ATLAS trigger



https://cds.cern.ch/record/2285584 [12]

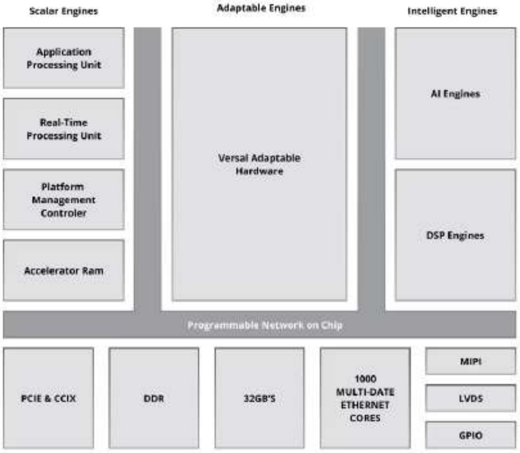
- First trigger level implemented in hardware (e.g. FPGAs)
- Strict latency budget due to limited buffering capabilities of readout
- Specialized trigger modules for different subdetectors
- ML algorithms improve trigger efficiency

Evaluation of AMD/Xilinx AI engines for the ATLAS trigger

- New AMD/Xilinx Versal family offers devices combining regular FPGA fabric with specialized AI engines
  - Processing units with instruction set, not just multipliers
  - Each engine can do e.g. 32 16 bit multiplications in one clock cycle

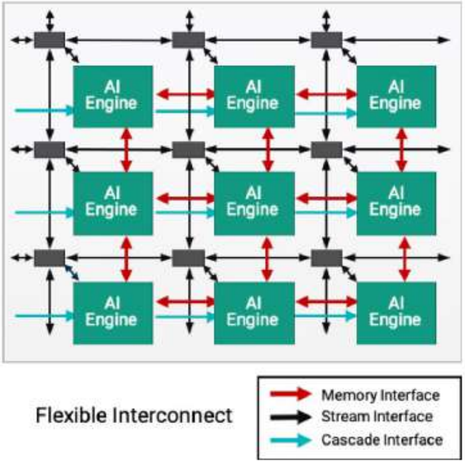
Objective

- Evaluate suitability of AI engines for low latency applications


Scalar Engines
Adaptable Engines
Intelligent Engines
Application Processing Unit
Real-Time Processing Unit
Platform Management Controler
Accelerator Ram
Versal Adaptable Hardware
AI Engines
DSP Engines
Programmable Network on Chip
PCIE & CCIX
DDR
32GB'S
1000 MULTI-DATE ETHERNET CORES
MIPI
LVDS
GPIO

https://www.amd.com/en/products/adaptive-socs-and-fpgas/technologies/ai-engine.html [13]

Program and connect the AI engines

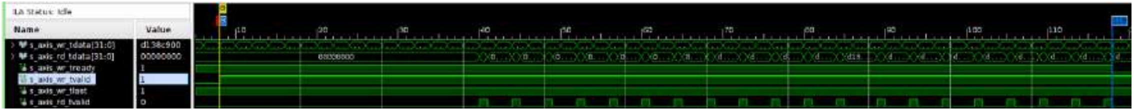


https://www.amd.com/en/products/adaptive-socs-and-fpgas/technologies/ai-engine.html [13]

- AMD recommends usage of Vitis AI tools
  - Implements DPU IP-core
  - Seems to cause latency of $> 30\,\mu s$ even for simple networks
- Alternative manual approach
  - Develop firmware for programmable logic with AXI interface to AI engines
  - Export project to Vitis
  - Write kernels for AI engines (C++) and connect to AXI interfaces
  - Build full project in Vitis and Vivado

Latency measurement

- Interface can run at up to half of AI engine frequency
- First tests suggest round-trip latency between programmable logic (FPGA) and AI engines 51.2 ns (40 clock cycles at 625 MHz)
- Constant writing on interface leads to backpressure due to FIFO filling up



ILA Status: Idle

Name	Value
s_axis_wr_tdata[31:0]	d138c900
s_axis_rd_tdata[31:0]	00000000
s_axis_wr_tready	1
s_axis_wr_tvalid	1
s_axis_wr_tlast	1
s_axis_rd_tvalid	0

https://indico.desy.de/event/45348/contributions/173483/ [14]

Summary and recommendations

- Finishing summary document with recommendations about ML on FPGA
- FPGAs are essential component in particle physics and astrophysics experiment readouts due to low latency and high data throughput
  - ANNs on FPGA improve physics performance, especially for trigger systems
  - Hybrid architectures with AI accelerators promising for future applications
- 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

Sources I

[1] PUNCH4NFDI. *Task areas*. URL: https://www.punch4nfdi.de/about/task_areas/ (visited on 06/30/2025).

[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] SKAO. *SKA-Mid - close up artist impression*. June 24, 2021. URL: https://skao.canto.global/v/SKAOMediaKit/album/OMQJL?viewIndex=2&column=image&id=1chq7ohjc10ah3k4e0dm15mj7k (visited on 06/26/2025).

[4] 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).

Sources II

[5] Andrei Kazantsev. *Deep Learning for Real-time Classification of Astronomical Radio Signals: Current Status*. Sept. 18, 2024. URL: https://indico.desy.de/event/45348/contributions/173482/ (visited on 07/01/2025).

[6] 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).

[7] Yunpeng Men. *PUNCH4NFDI TA5 Workshop (Dresden, 2024)hls4ml*. 2024. URL: https://github.com/ypmen/punch_workshop (visited on 07/02/2025).

[8] 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).

Sources III

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

[10] 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).

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

[12] 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).

Sources IV

[13] AMD. *AMD AI Engine Technology*. 2025. URL: https://www.amd.com/en/products/adaptive-socs-and-fpgas/technologies/ai-engine.html (visited on 06/27/2025).

[14] Christian Kahra. *Experience with Xilinx Versal AI*. Sept. 18, 2024. URL: https://indico.desy.de/event/45348/contributions/173483/ (visited on 07/01/2025).