ML on FPGA Developments in ATLAS

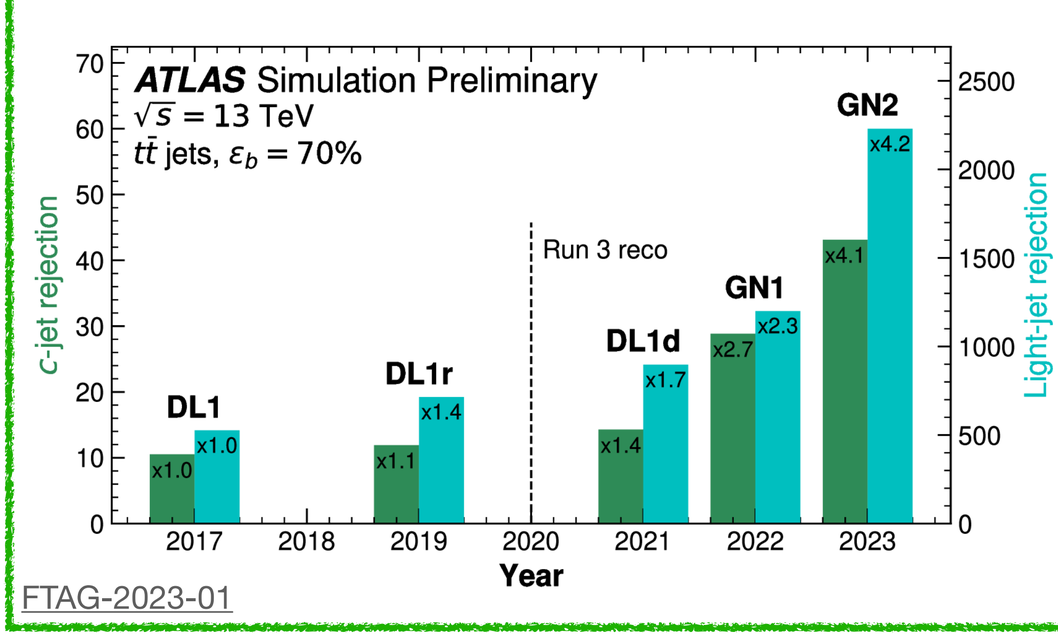
Dylan Rankin [UPenn]

KMI2025 : The 6th KMI International Symposium

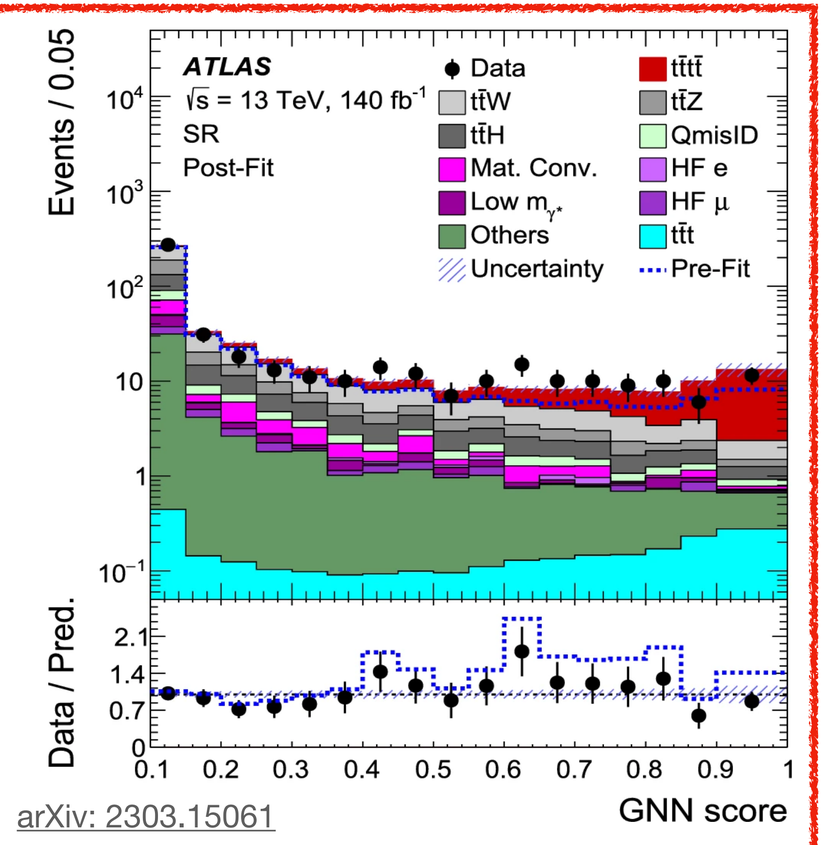
March 7th, 2025

Introduction

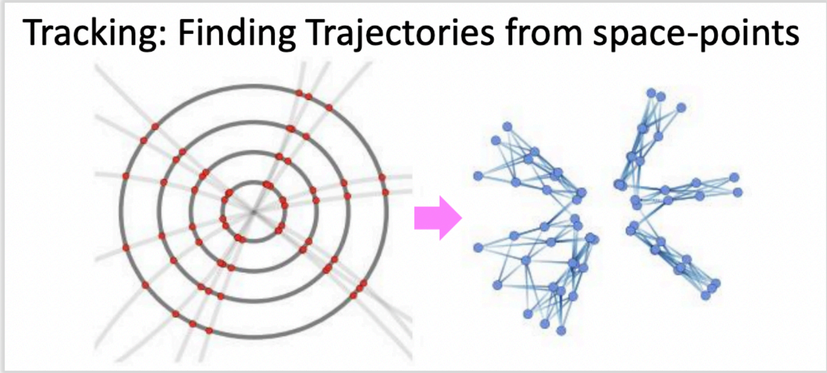
- Machine learning (ML) is becoming more and more popular, HEP/LHC/ATLAS no exception
- Better algorithms → improved sensitivity to new physics

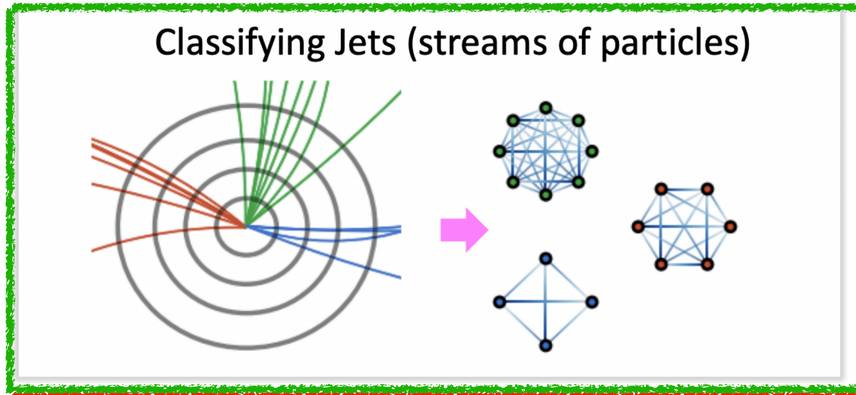


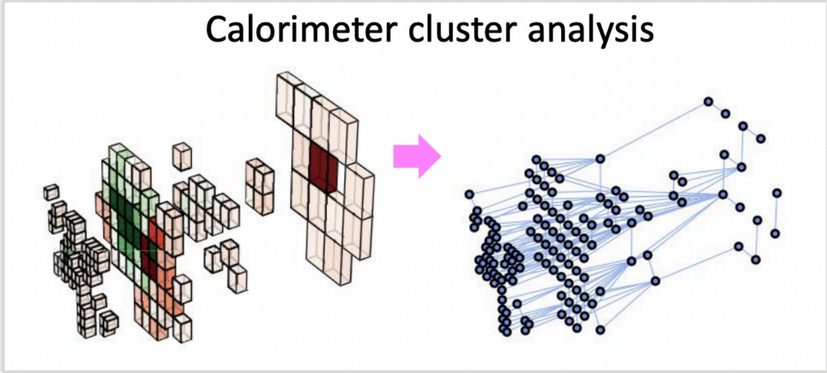
FTAG-2023-01

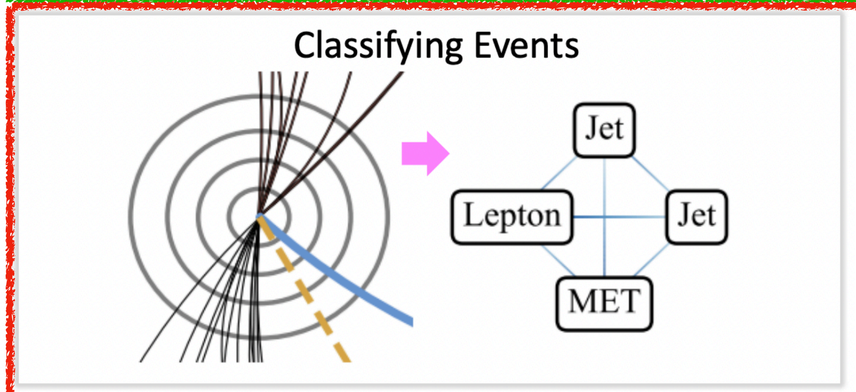


arXiv: 2303.15061



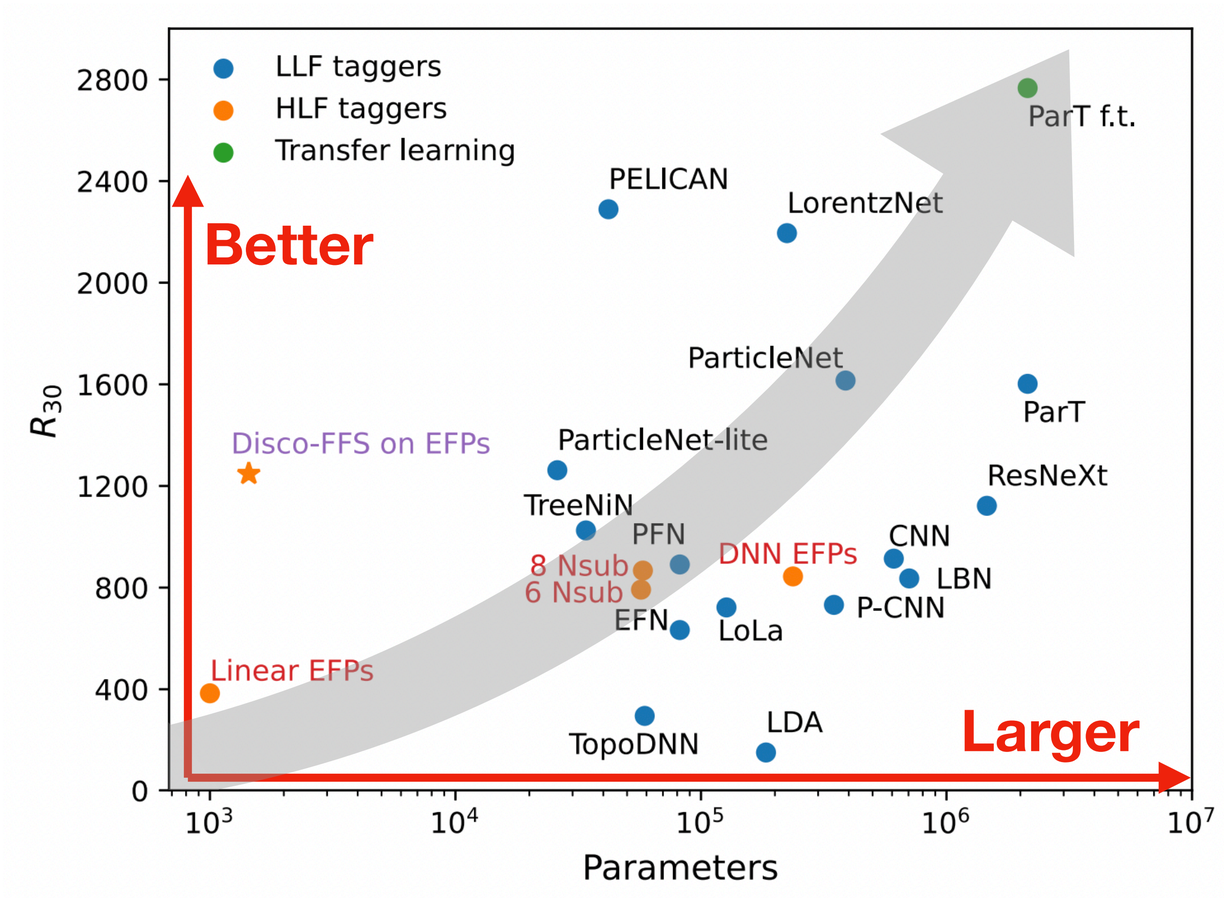






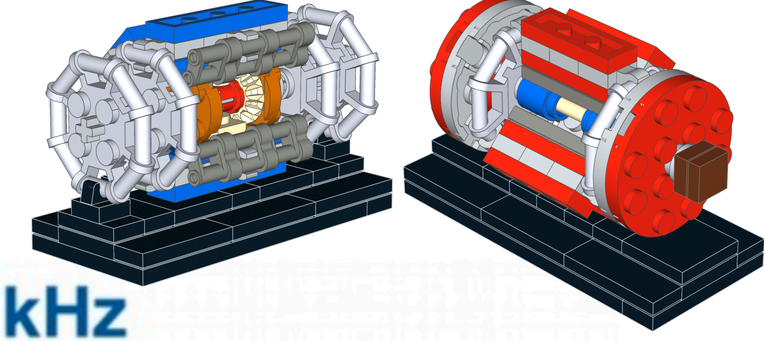
ML in HEP

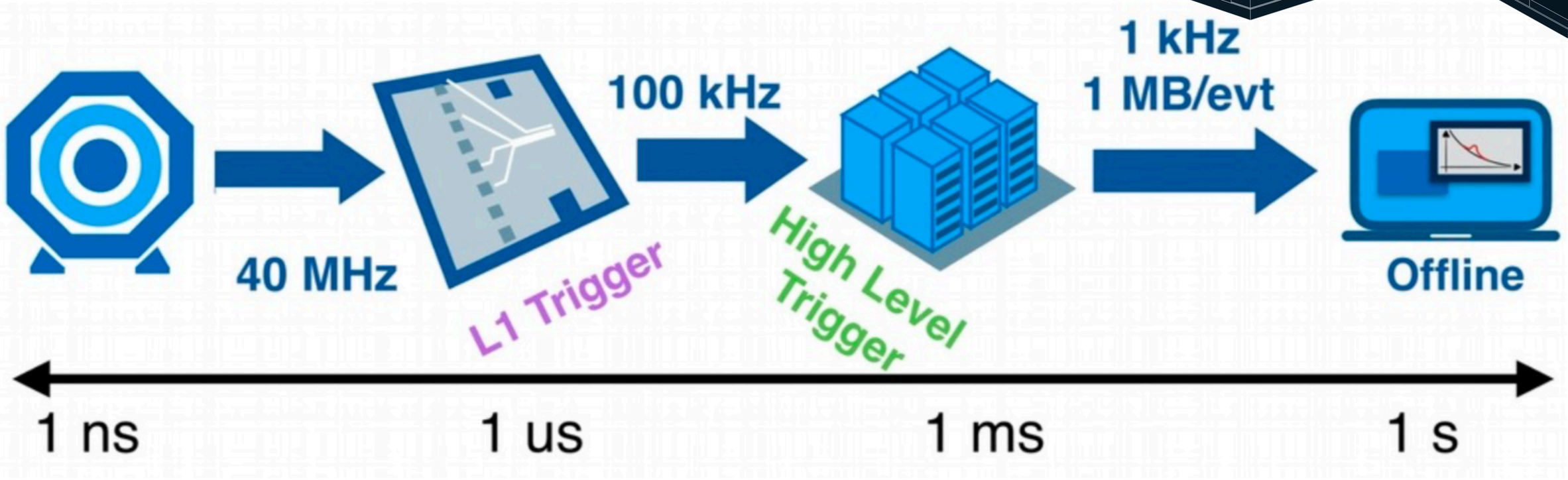
- HEP trends in ML towards **bigger and more complicated models**, more computing
- → **Majority of ML in physics is "off detector"**
  - System latency/resource limits are typically soft (if at all)
  - No radiation
  - Issues do not impact data collection
    - Can re-run algorithms/workflows



PhysRevD.109.054009

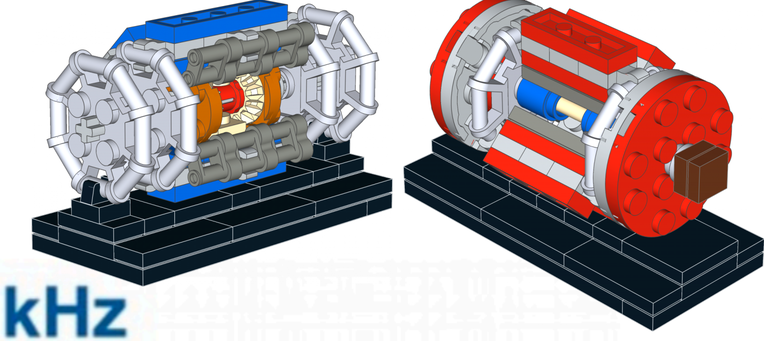
LHC Data Processing / Readout

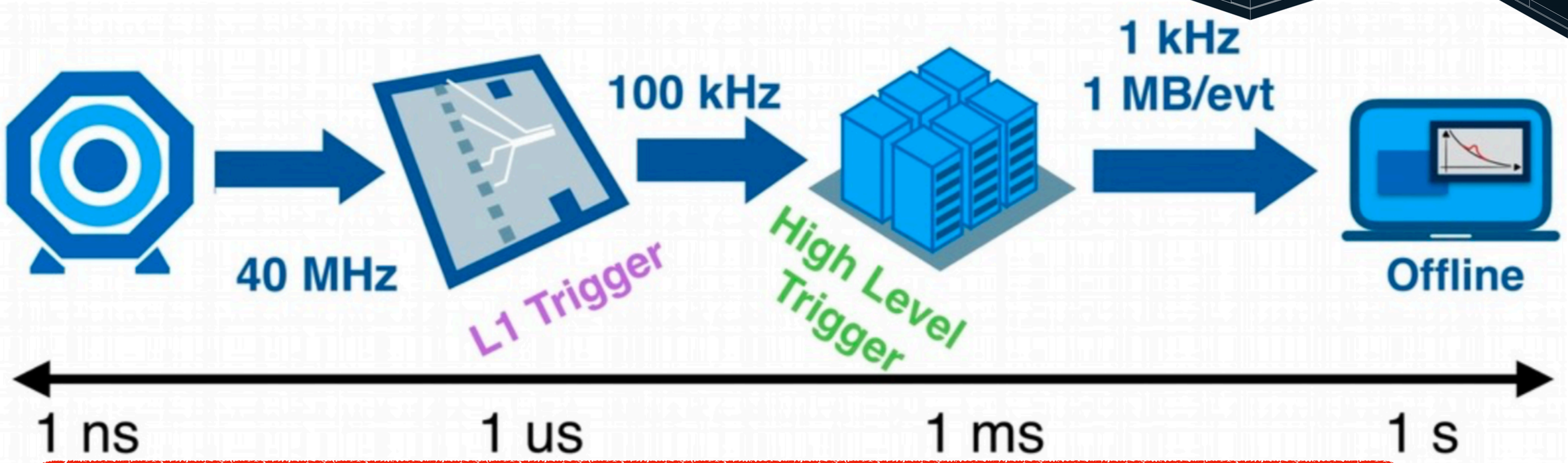




- **Level-1 Trigger** (FPGAs, ASICs) - O(μs) hard latency
- **High Level Trigger** (CPUs, GPUs, FPGAs?) - O(100 ms) soft latency
- **Offline** (CPUs, GPUs) → >1 s latencies ← **Most ML @ LHC lives here**

LHC Data Processing / Readout





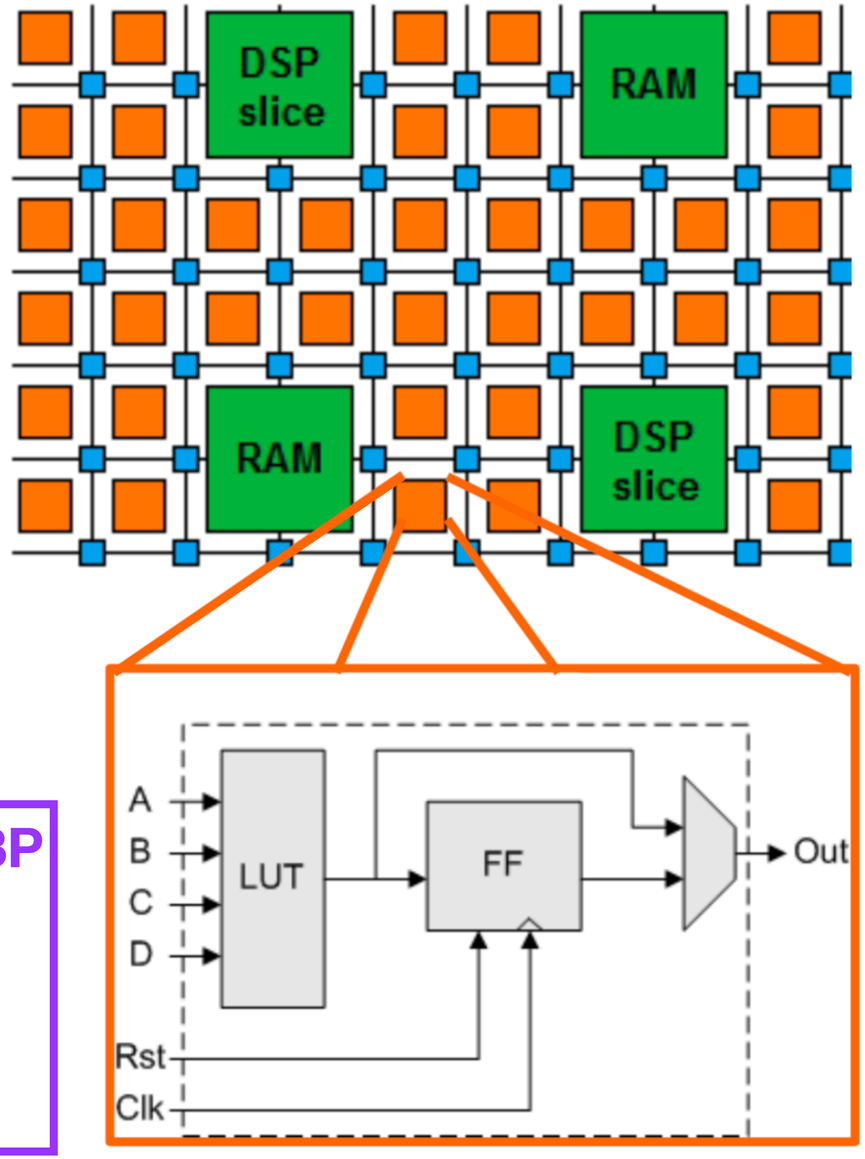
- **Level-1 Trigger** (FPGAs, ASICs) - O(μs) hard latency
- **High Level Trigger** (CPUs, GPUs, FPGAs?) - O(100 ms) soft latency
- **Offline** (CPUs, GPUs) → >1 s latencies

If we don't identify interesting events here we lose them forever!

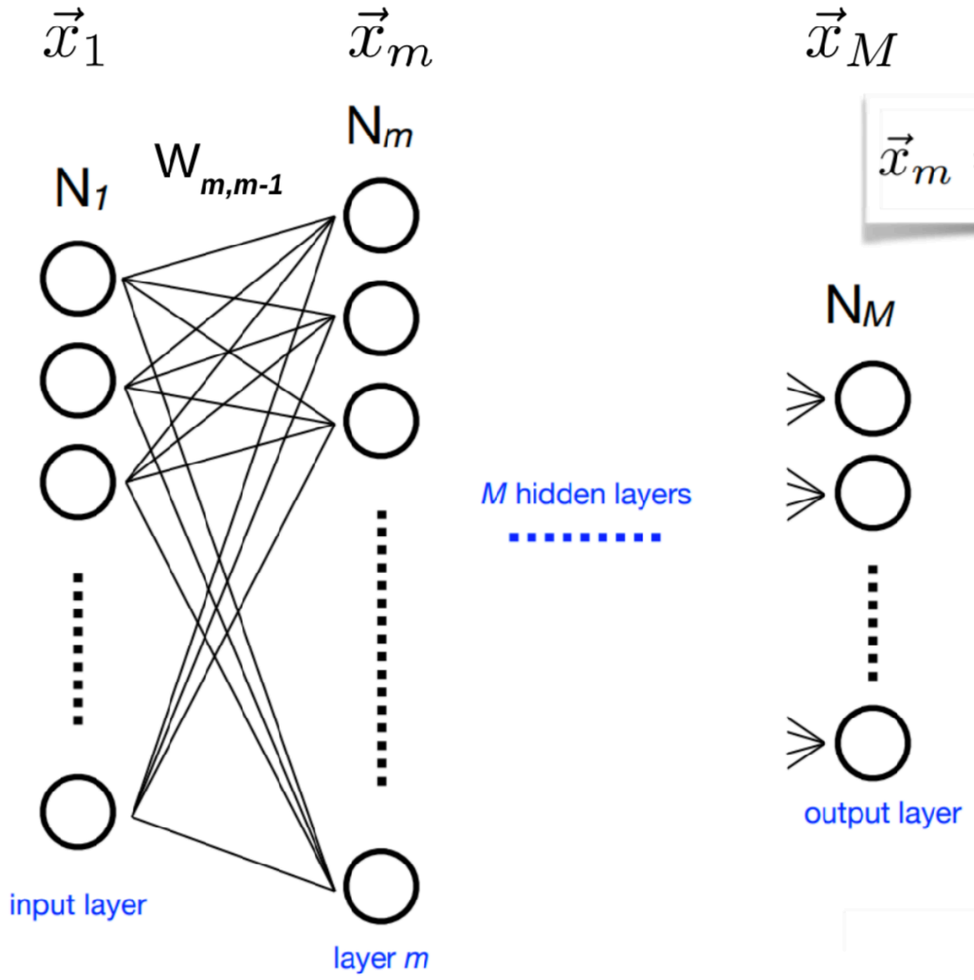
What is an FPGA?

- **F**ield-**P**rogrammable **G**ate **A**rray
- Building blocks:
  - **Multiplier units (DSPs) [arithmetic]**
  - **Look Up Tables (LUTs) [logic]**
  - **Flip-flops (FFs) [registers]**
  - **Block RAMs (BRAMs) [memory]**
- Algorithms are wired onto the chip
  - Can only use the resources on the chip
- Run at high frequency: hundreds of MHz, O(ns) runtime

Xilinx Virtex Ultrascale+ VU13P
12288 Multipliers
1.7M LUTs
3.4M FFs
95 Mb BRAM



What is a Neural Network?



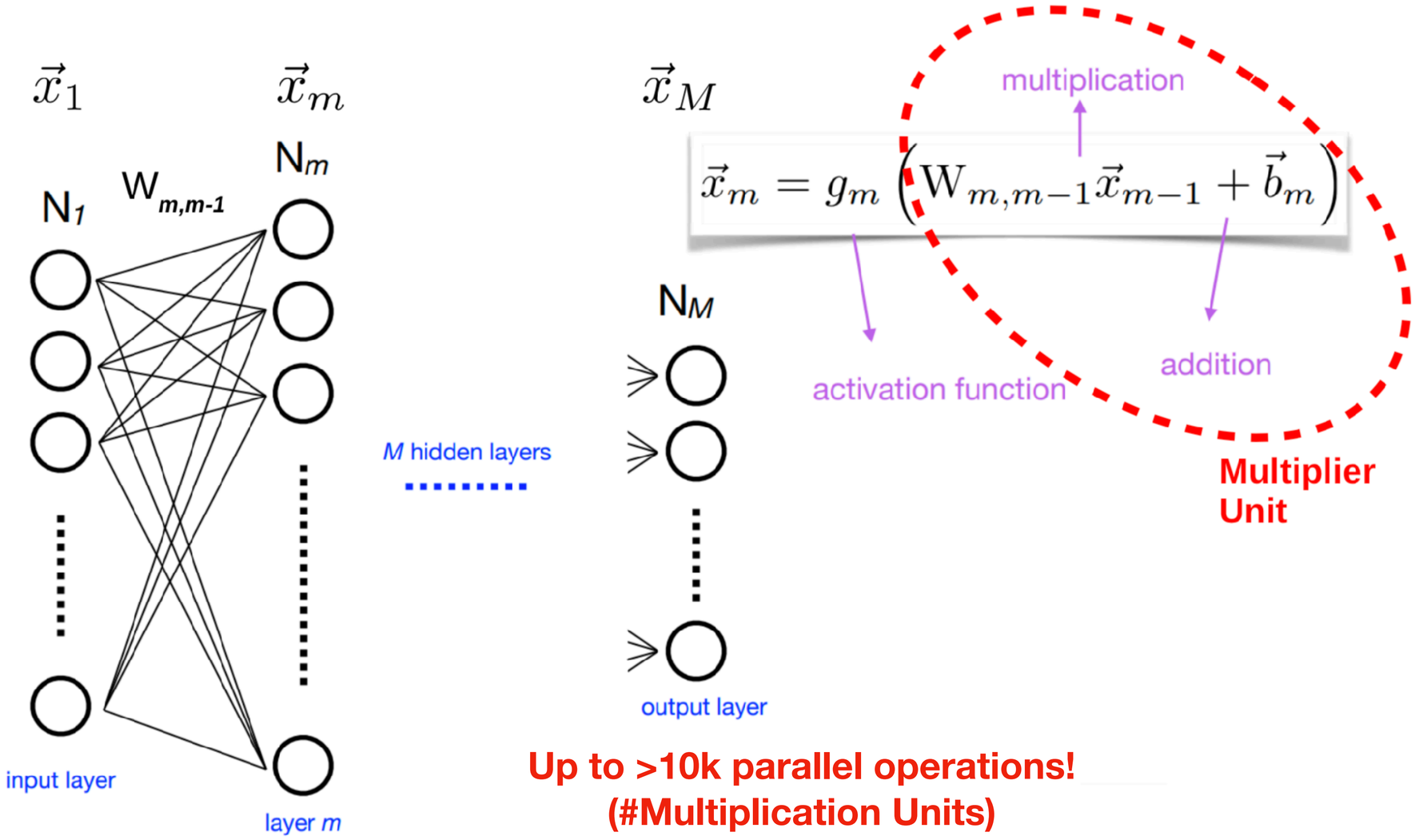
multiplication

$$\vec{x}_m = g_m\left(\mathrm{W}_{m,m-1}\vec{x}_{m-1} + \vec{b}_m\right)$$

activation function

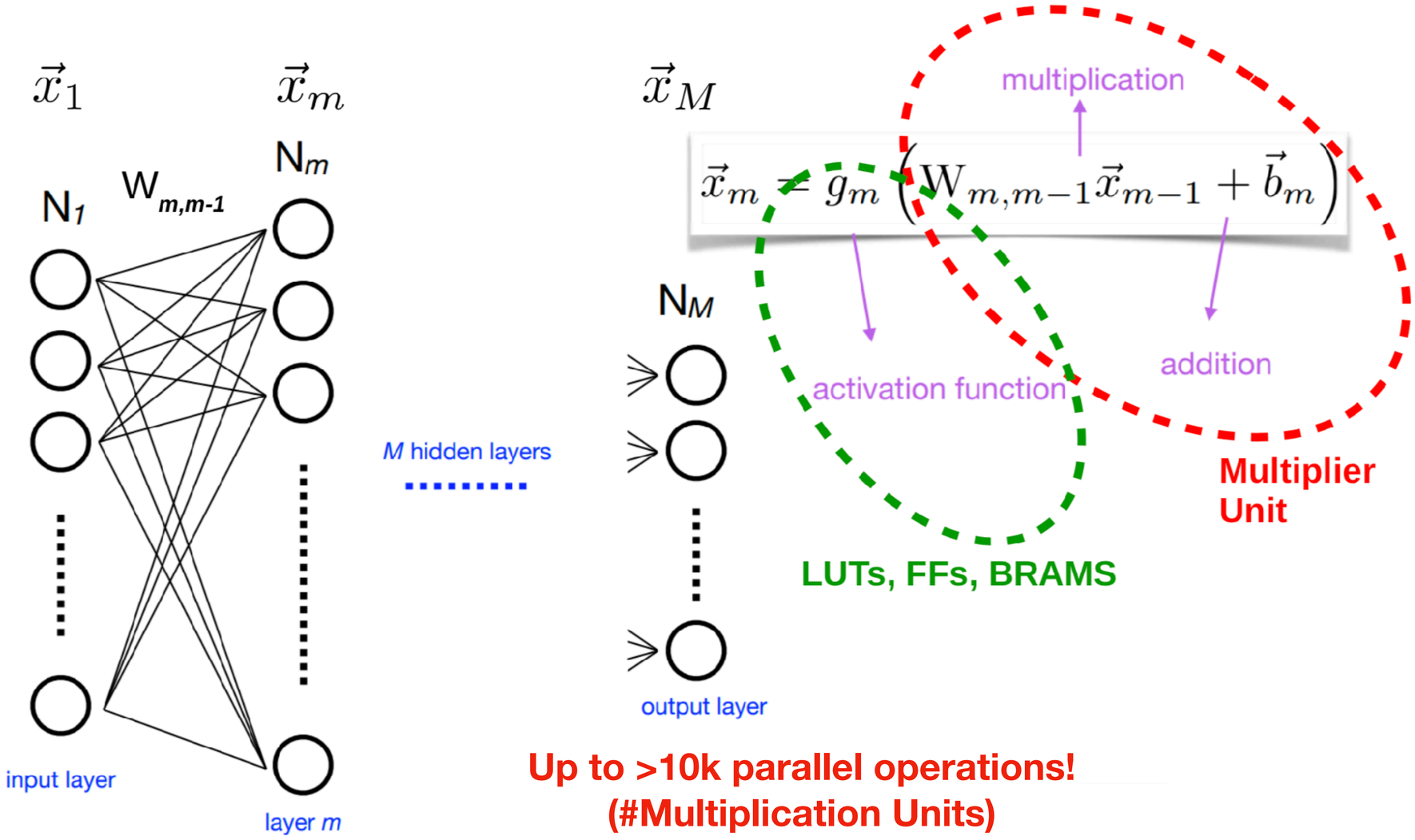
addition

Inference on FPGAs



Up to >10k parallel operations!
(#Multiplication Units)

Inference on FPGAs



Up to >10k parallel operations!
(#Multiplication Units)

- hls4ml is a software package for automatically creating implementations of neural networks for FPGAs and ASICs
  - https://fastmachinelearning.org/hls4ml/ [arXiv:1804.06913]
  - pip installable
- Supports common layer architectures and model software (keras, tensorflow, pytorch, ONNX)
  - Converts model to High-Level Synthesis (HLS) for use with FPGA vendor-specific tools (eg. Vitis HLS)
  - Active development of new architectures, related techniques

Many Other Tools

- NNs: FINN

arXiv: 2004.03021

SNL (?)

arXiv: 2305.19455

- Boosted Decision Trees (BDTs):

Conifer

arXiv: 2002.02534

FWXMachina

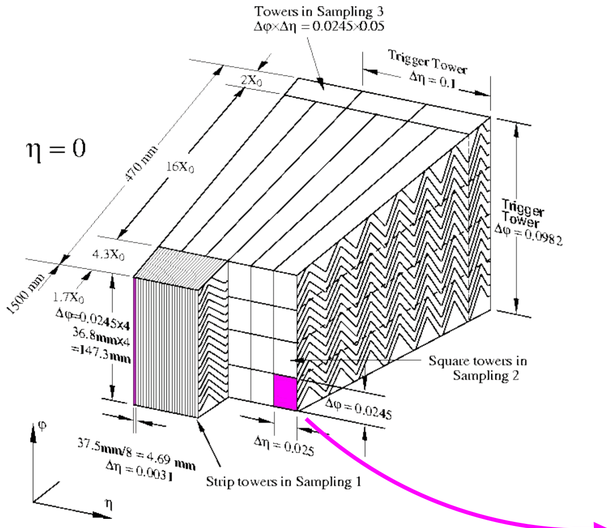
arXiv: 2104.03408

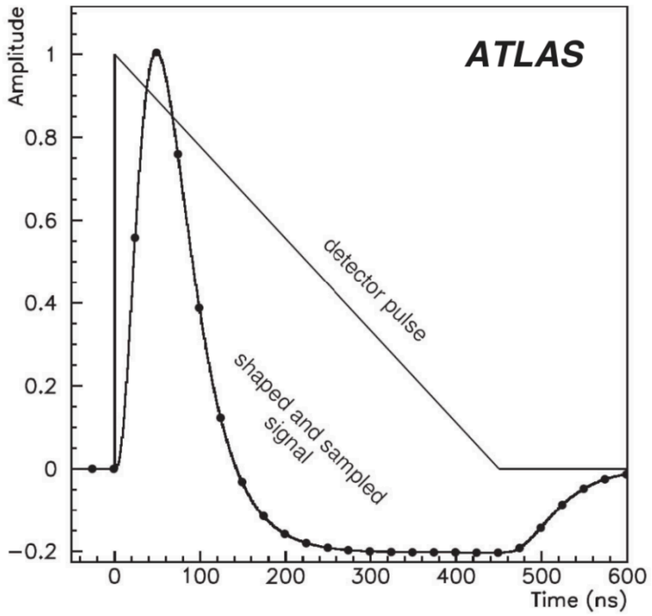
- Different tools have different methodology, target different designs/problems
- Entirely non-exhaustive list…

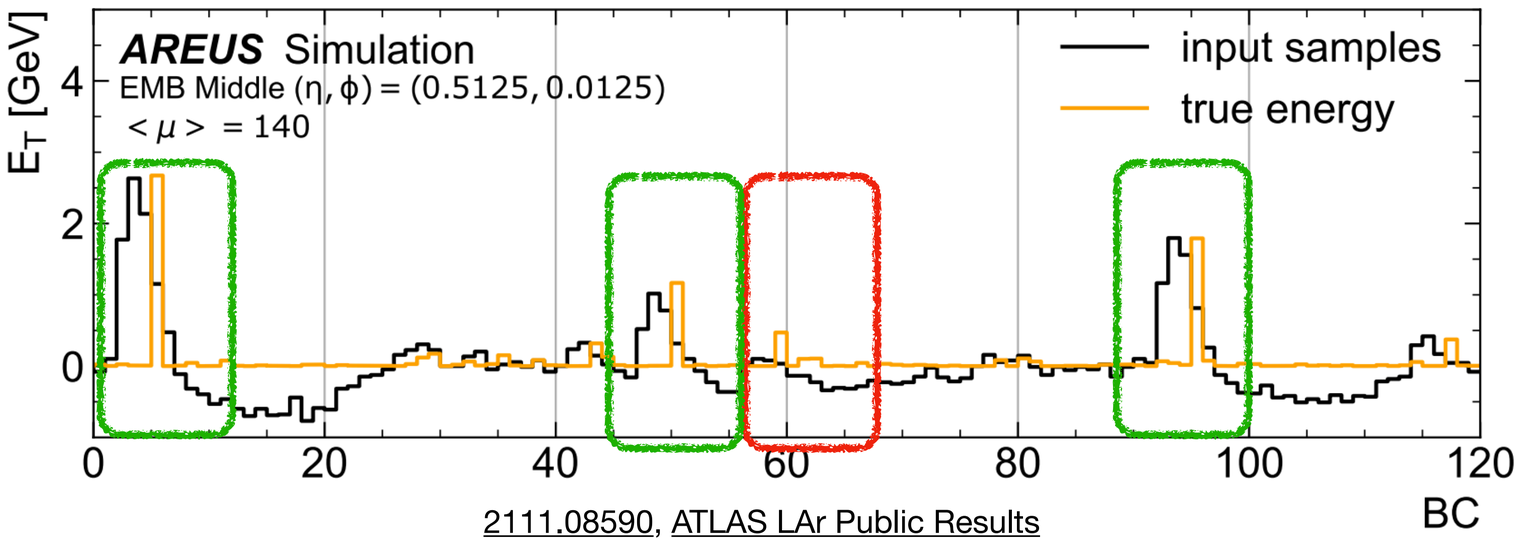
ATLAS Applications

LAr Peak Finding

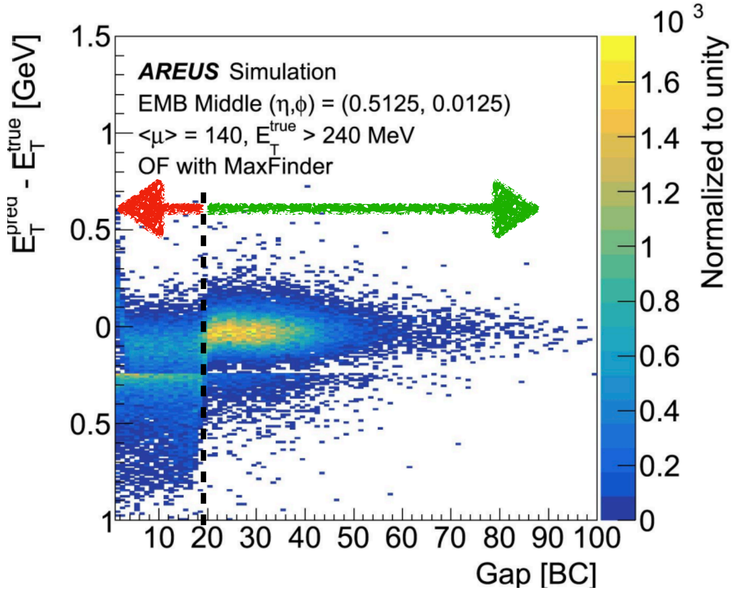
- ATLAS LAr calorimeter needs to measure time and energy of pulses
  - Overlapping pulses difficult for simple, fast algorithms to handle (150 ns = 6 BXs)
- CNN and LSTM architectures both able to significantly improve performance
  - Well-suited for data structure, able to account for non-linear correlations





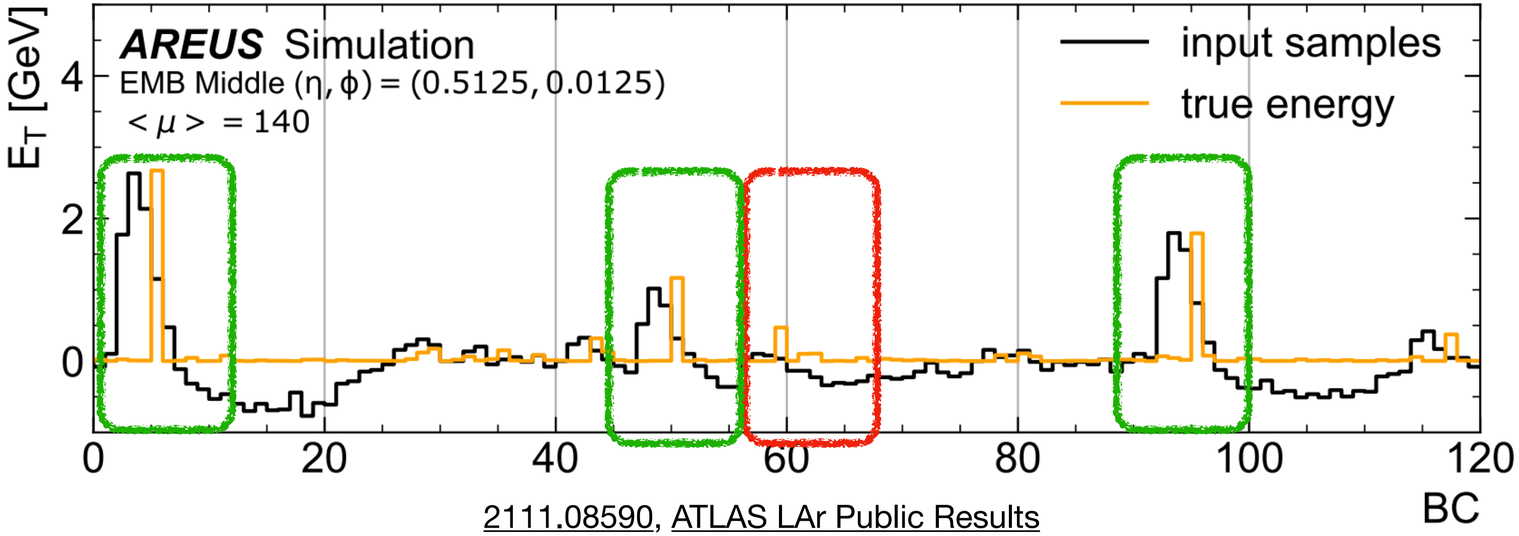


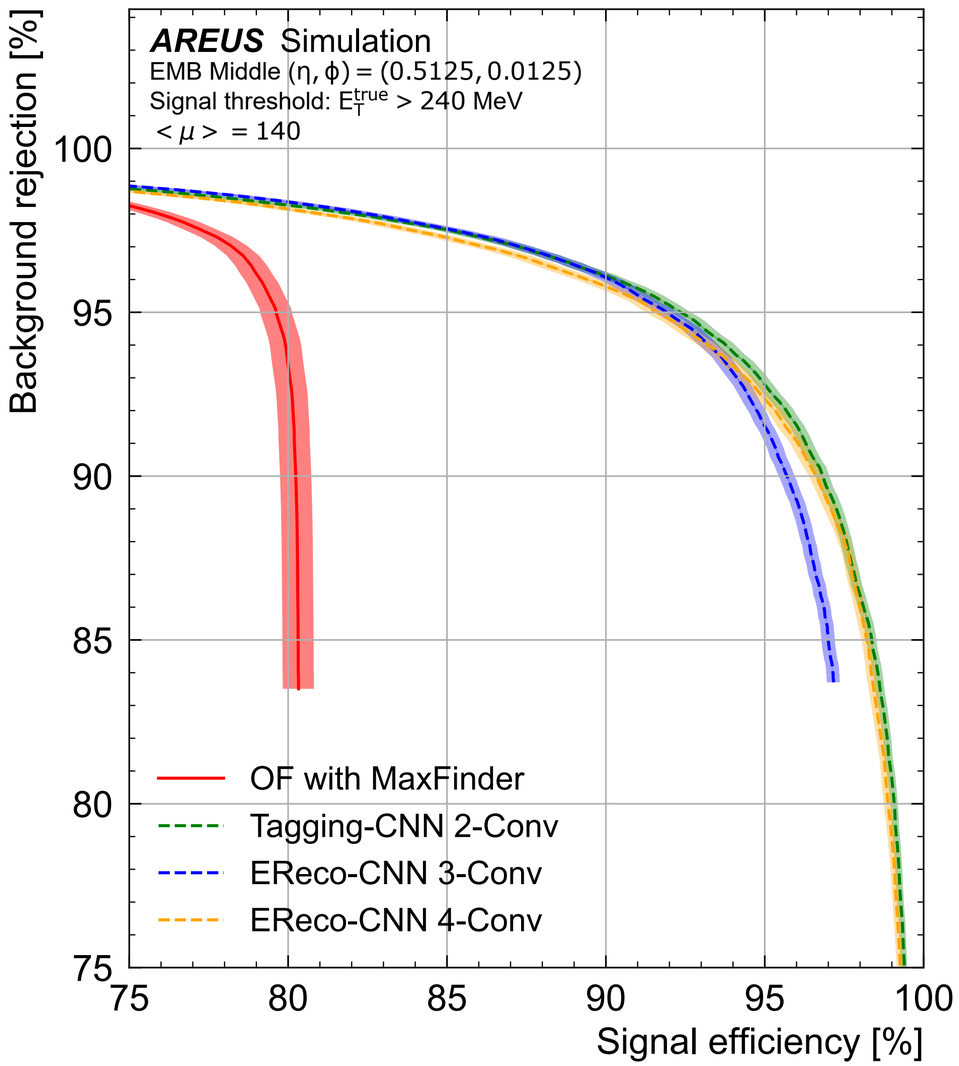
2111.08590, ATLAS LAr Public Results



LAr Peak Finding

- ATLAS LAr calorimeter needs to measure time and energy of pulses
  - Overlapping pulses difficult for simple, fast algorithms to handle (150 ns = 6 BXs)
- CNN and LSTM architectures both able to significantly improve performance
  - Well-suited for data structure, able to account for non-linear correlations

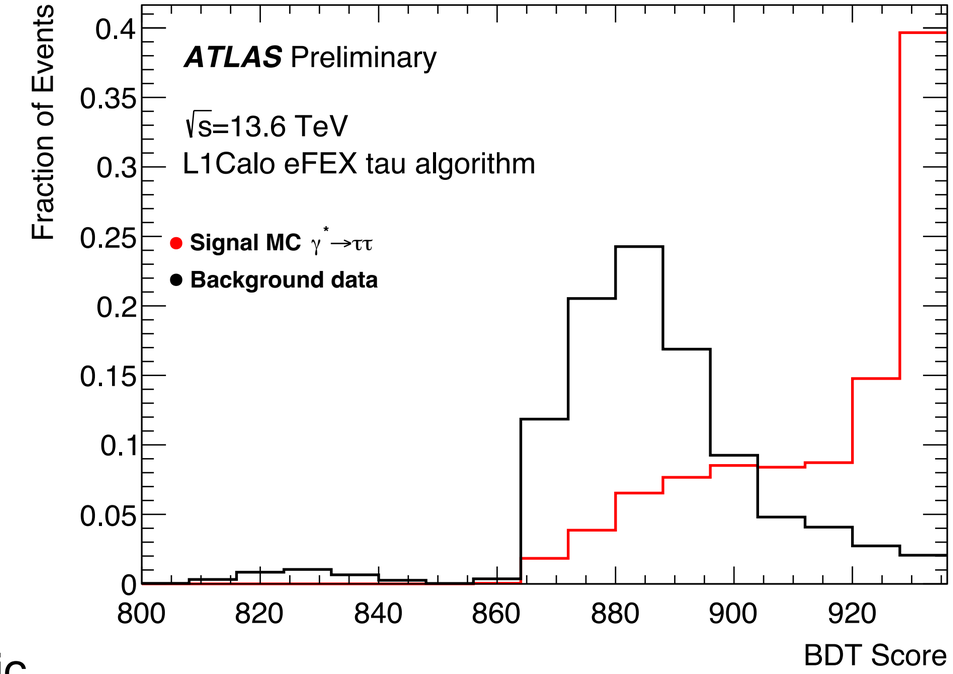




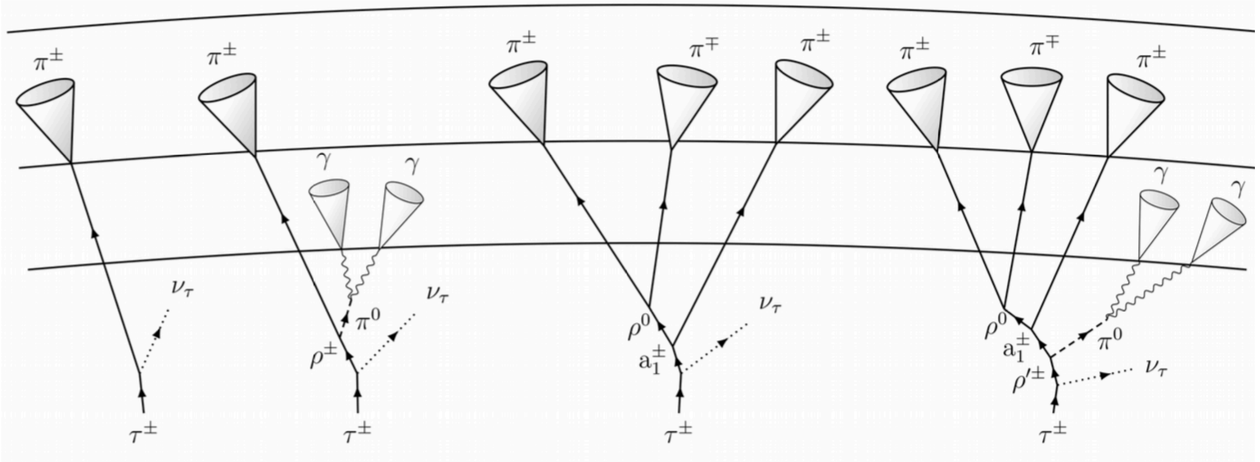
2111.08590, ATLAS LAr Public Results

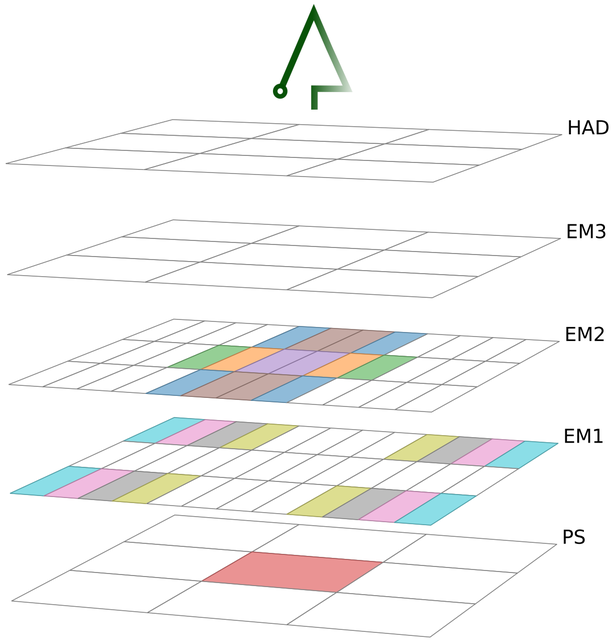
Hadronic τ Identification

- Tau leptons decay to hadrons ~65% of time (τ_h)
  - Difficult to distinguish from hadronic jets
  - Need to combine information from multiple different subdetectors
- Critical for many signals, eg. HH→bbττ
- BDT developed for identification of hadronic taus from energy in specific regions of calorimeters (+ total energy)
- Translated to firmware with conifer ( )



1-prong (+ π⁰) *3-prong (+ π⁰)*

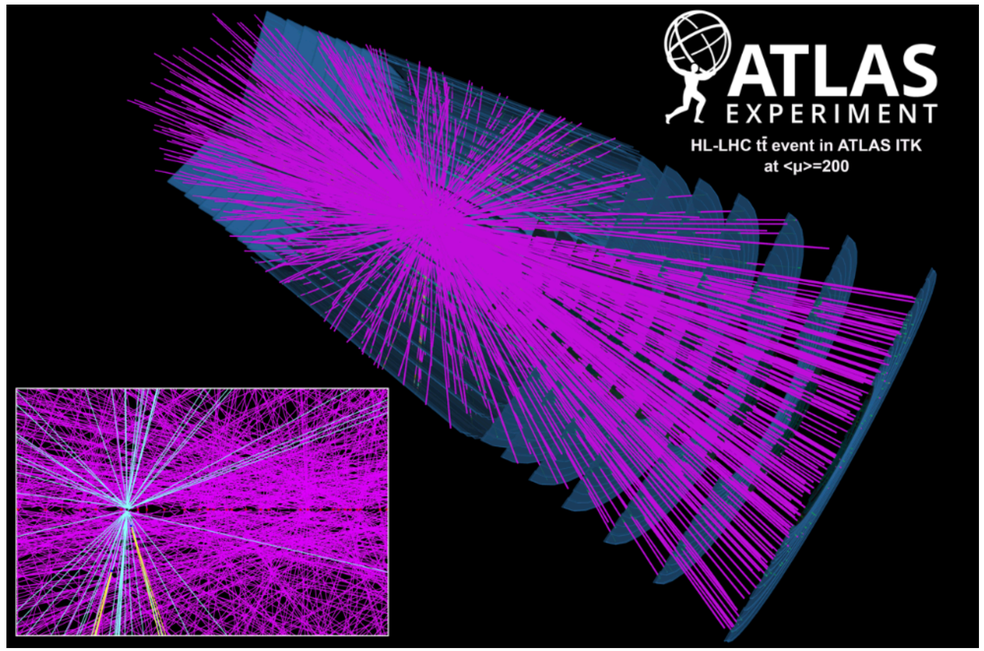


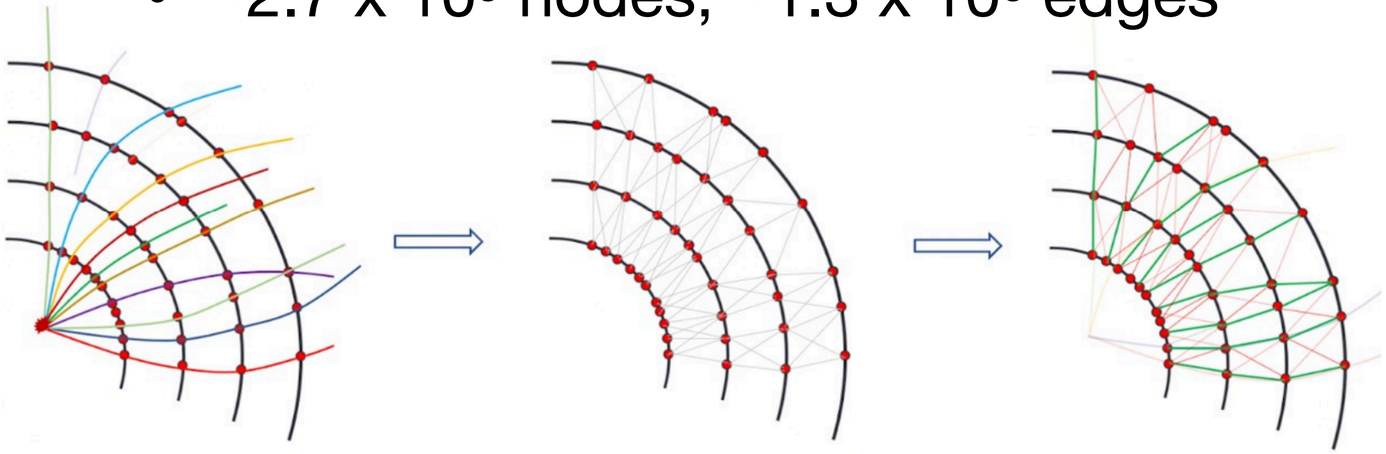


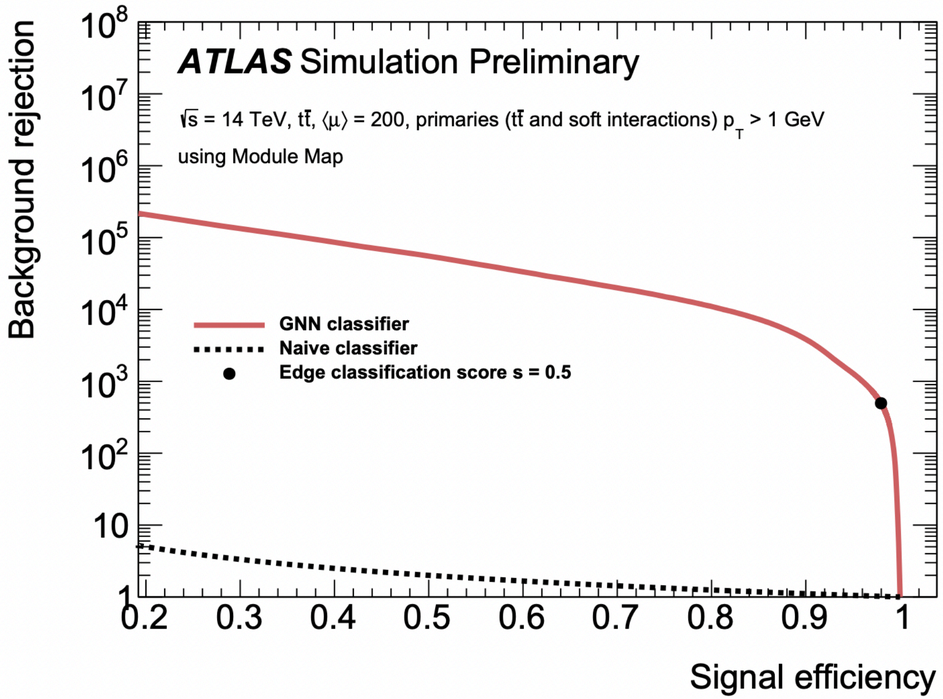
ATLAS L1Calo Trigger Public Results

Particle Tracking

- Tracking is an incredibly hard problem, tracking in HLT even harder
  - Huge combinatorics, only going to get worse
- Graph neural networks (GNNs) show promise for HL-LHC
  - ~2.7 x 10^5 nodes, ~1.3 x 10^6 edges



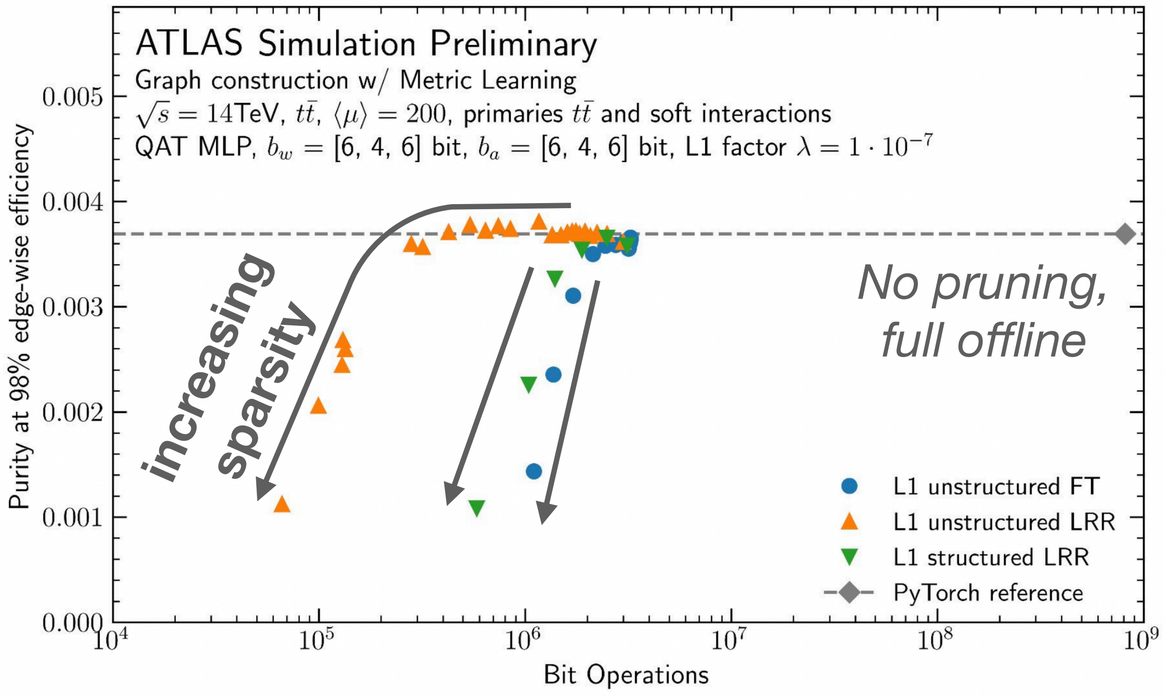




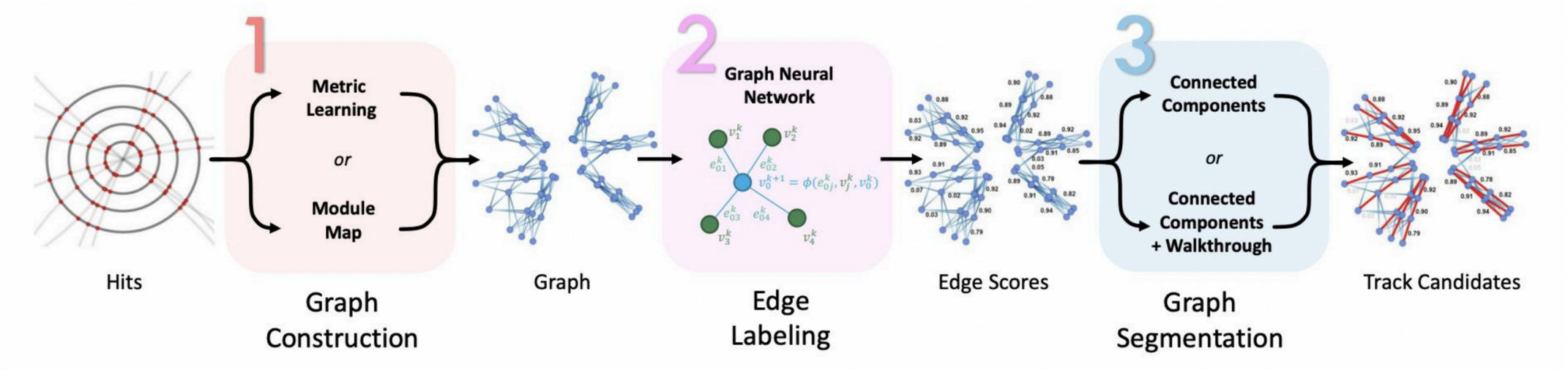
ATL-ITK-PROC-2022-006

GNN Tracking

- Pipeline from raw hits to track candidates involves multiple steps
- Complicated workflow, large networks
- Pruning (removing nodes of network) one potential option for reducing size
  - Especially effective for inference on FPGAs

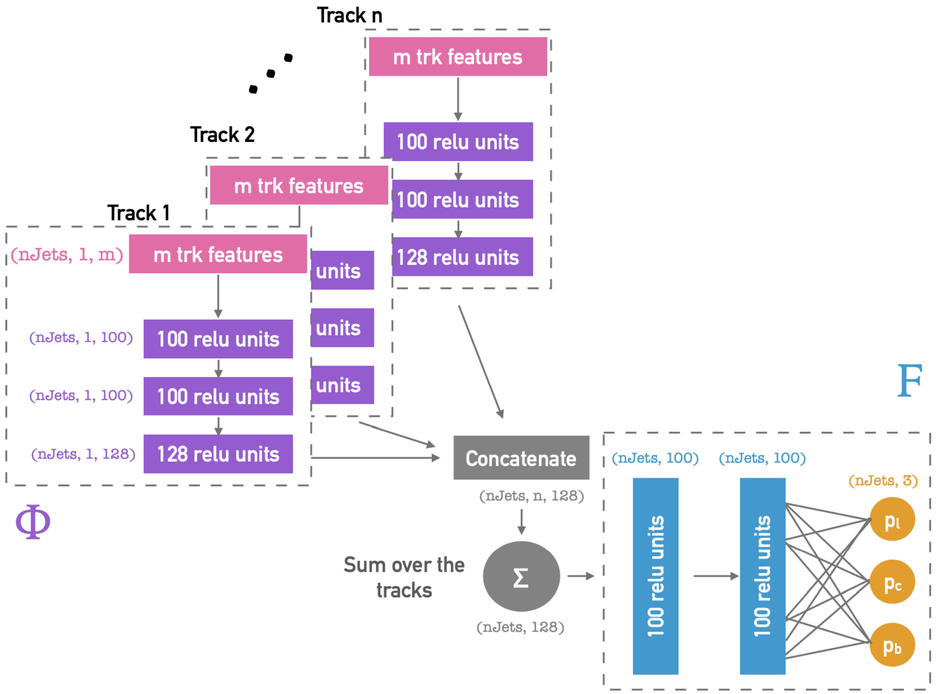


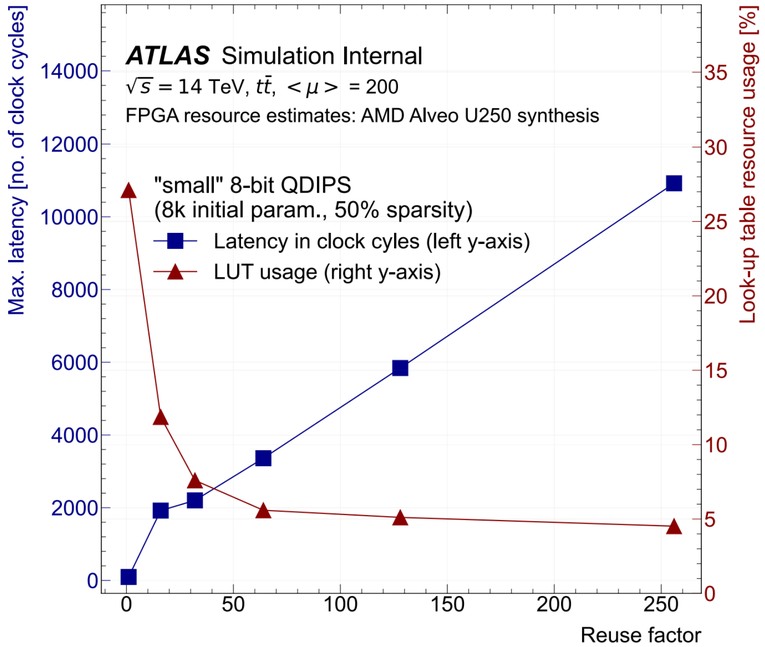
ATL-COM-DAQ-2024-004

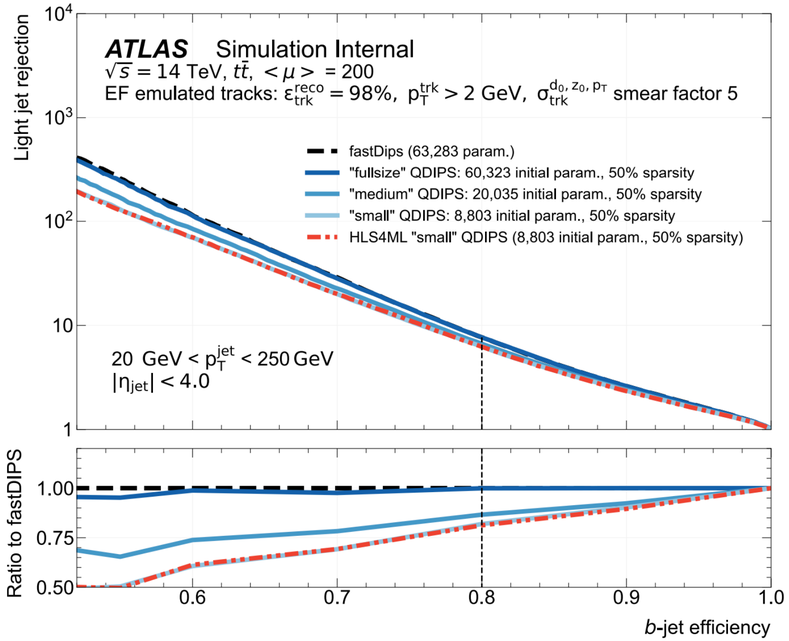


Fast b-tagging

- Many complex downstream tasks rely on tracks
  - b-tagging most notable, very complex offline algorithms
- Fast Deep Impact Parameter Sets (fastDIPS) developed as possible fast preselection algorithm
- QDIPS is a small quantized version of DIPS for use on FPGAs
  - Able to maintain near full performance w.r.t fastDIPS
  - Options to trade off **FPGA resources** and **algorithm latency**
- DIPS architecture (DeepSets) also applicable to many other tasks



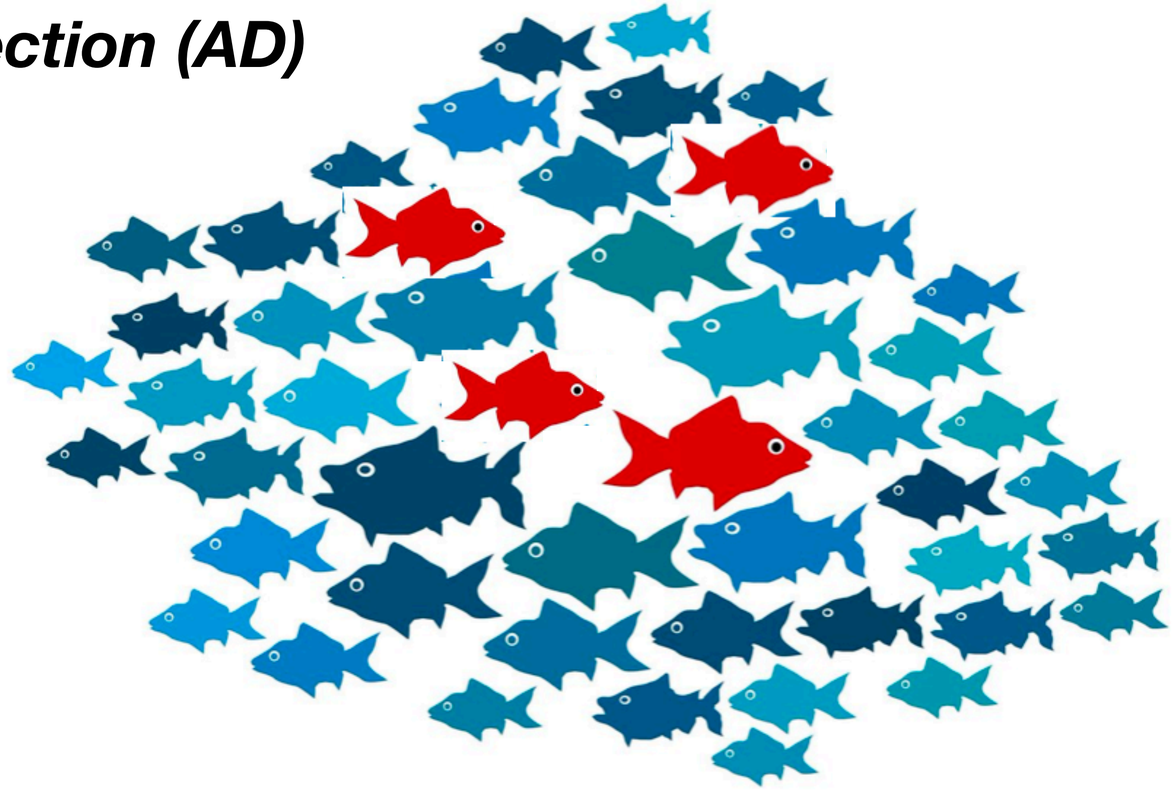




ATL-COM-DAQ-2024-102

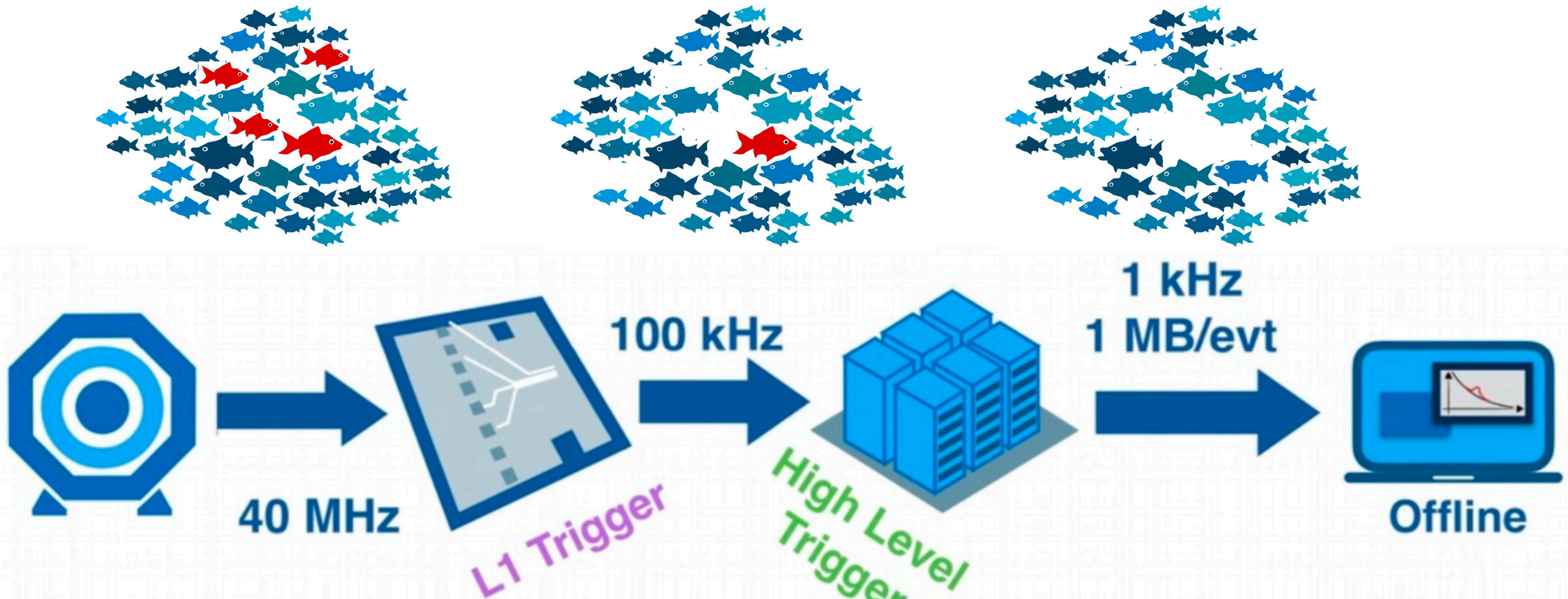
Anomaly Detection

- What if we don’t know exactly what we are looking for?
- ML offers unique solution to this challenge (no traditional alternative)
  - Broad field of ***anomaly detection (AD)***



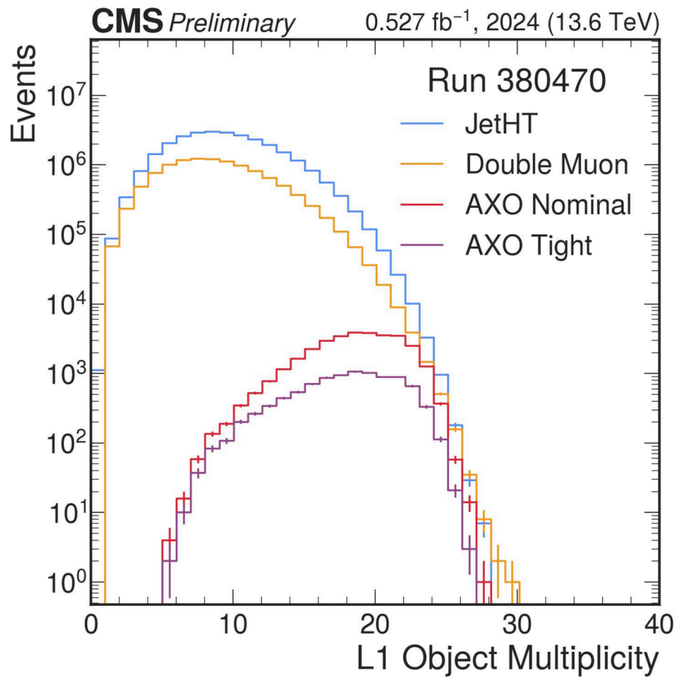
L1 Trigger AD

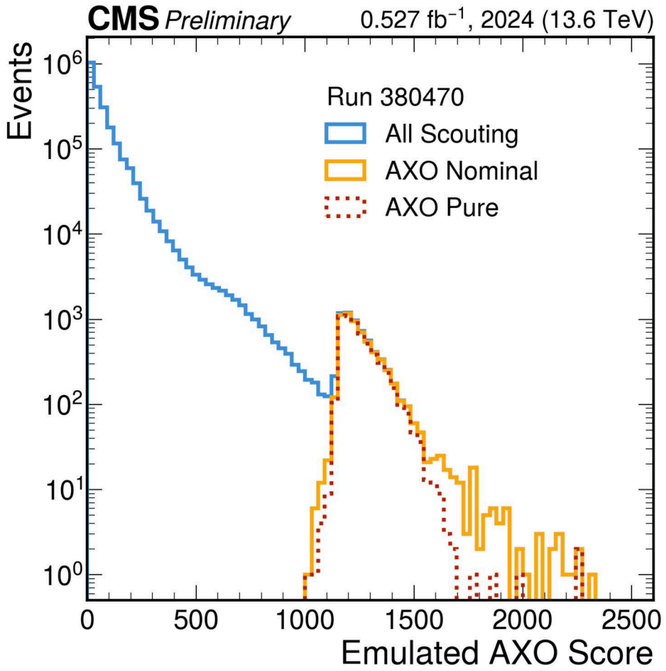
- Depending on anomaly, we could have none left in recorded data
- Low-latency ML on FPGAs is the only option! (eg. autoencoders)

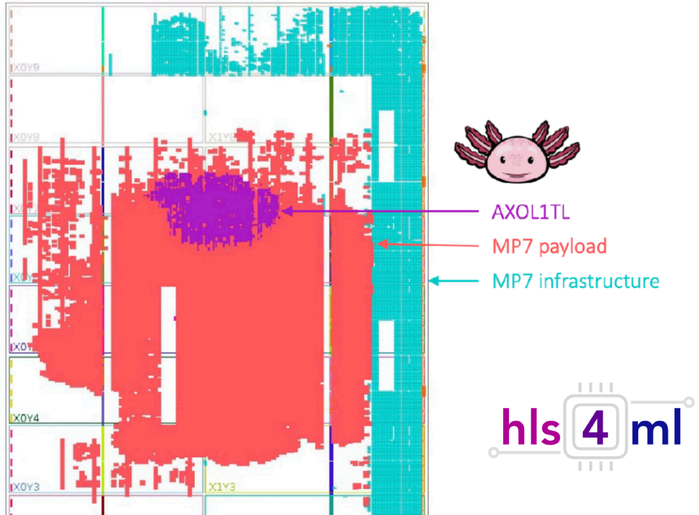


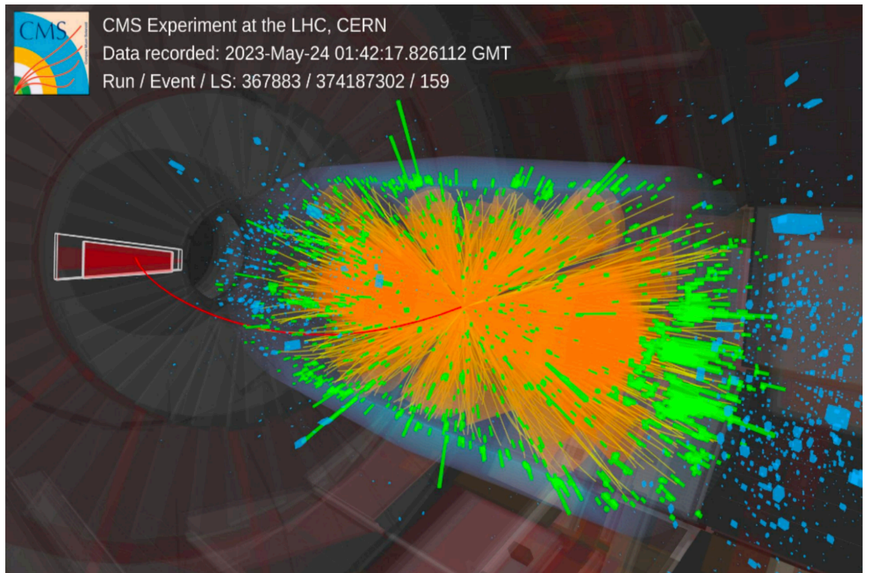
L1 Trigger AD

- CMS has already deployed multiple AD algorithms in trigger
  - AXOL1TL [CMS DP-2023/079, CMS DP-2024/059] & CICADA [CMS DP-2023/086]
- Currently collecting interesting events that would have been missed
- Development ongoing in ATLAS as well, expect to deploy this year!



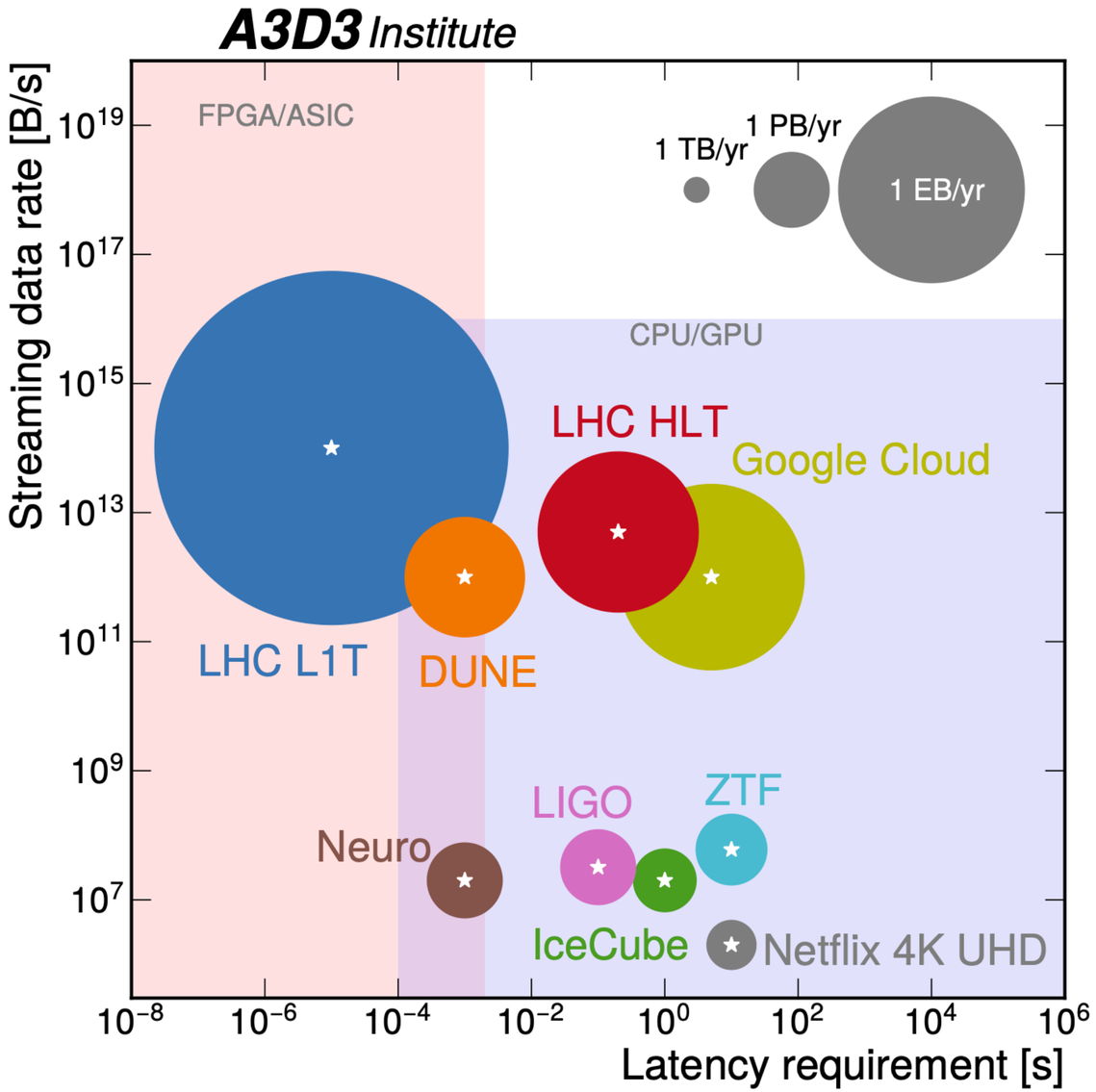






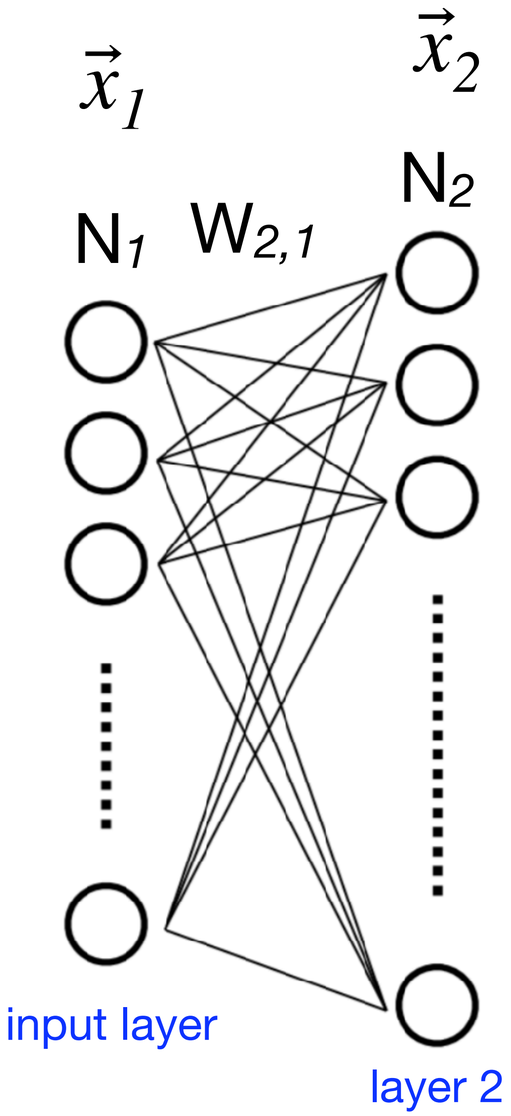
Conclusions

- Increasingly possible and necessary to perform real time ML in LHC experiments
- Many more developments than I could show, especially for future (HL-LHC)!
  - eg. Next Generation Triggers (NGT) [see here]
- ML offers improved performance over traditional algorithms
  - Advancing ML off-detector brings better alignment of offline and online algorithms
- Has the potential to enable discovery of new physics!
- Applications in many other fields, areas too



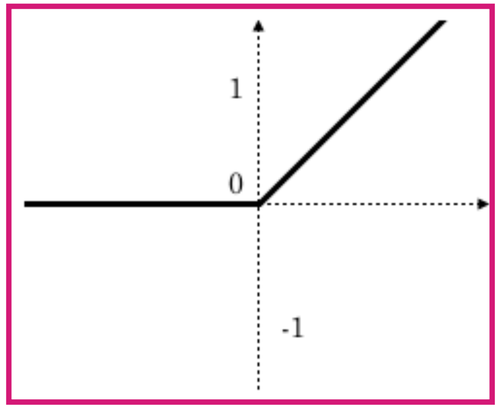
BACKUP

What is a Neural Network?



$$\vec{x}_1 = \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ \vdots \\ x_{1,N_1} \end{pmatrix} \qquad W_{2,1} = \begin{pmatrix} w_{1,1}^{(2,1)} & w_{2,1}^{(2,1)} & w_{3,1}^{(2,1)} & & & w_{N_1,1}^{(2,1)} \\ w_{1,2}^{(2,1)} & w_{2,2}^{(2,1)} & w_{3,2}^{(2,1)} & \cdot & \cdot & \cdot \; w_{N_1,2}^{(2,1)} \\ w_{1,3}^{(2,1)} & w_{2,3}^{(2,1)} & w_{3,3}^{(2,1)} & & & w_{N_1,3}^{(2,1)} \\ & \vdots & & \cdot & \cdot & \cdot \\ w_{1,N_2}^{(2,1)} & w_{2,N_2}^{(2,1)} & w_{3,N_2}^{(2,1)} & & & w_{N_1,N_2}^{(2,1)} \end{pmatrix} \qquad \vec{b}_2 = \begin{pmatrix} b_{2,1} \\ b_{2,2} \\ b_{2,3} \\ \vdots \\ b_{2,N_2} \end{pmatrix}$$

$g_2(\cdot) =$



$$\vec{x}_2 = g_2(W_{2,1}\vec{x}_1 + \vec{b}_2)$$

ML Size / Complexity

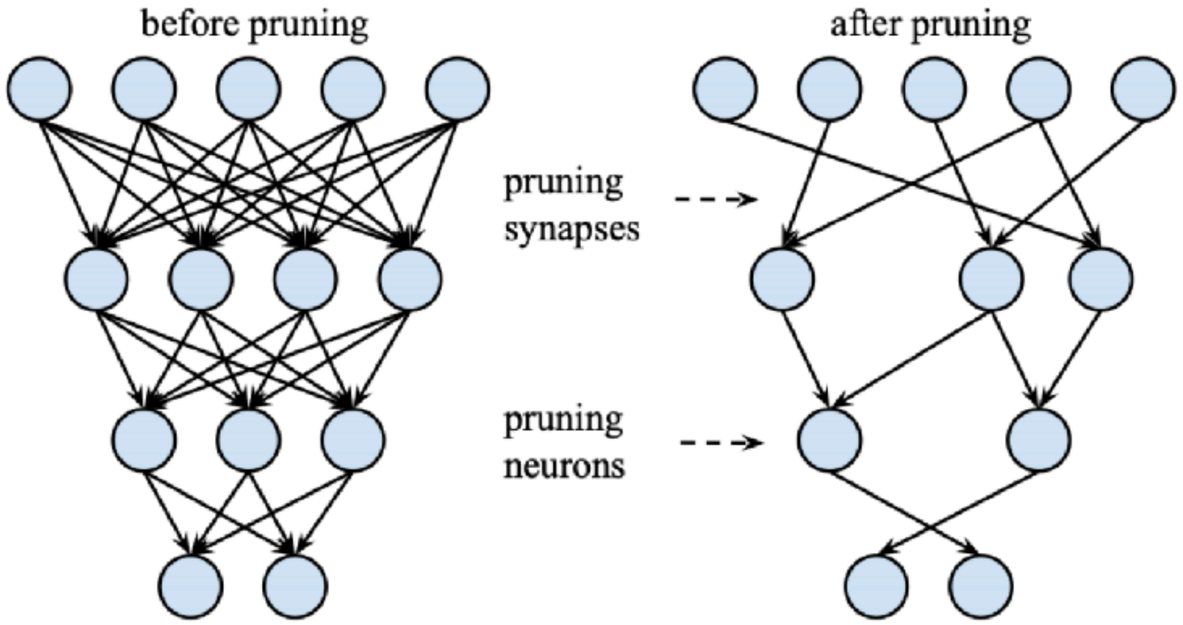
- Regardless of toolkit, big limitation of doing ML fast is device size
  - Bigger device → more resources → more computation → larger ML models

Xilinx Virtex Ultrascale+ VU13P
12288 Multipliers
1.7M LUTs
3.4M FFs
95 Mb BRAM

- Alternatively, is it possible to reduce network size without hurting performance?
  - ***Pruning*** and ***quantization*** are two potential ways

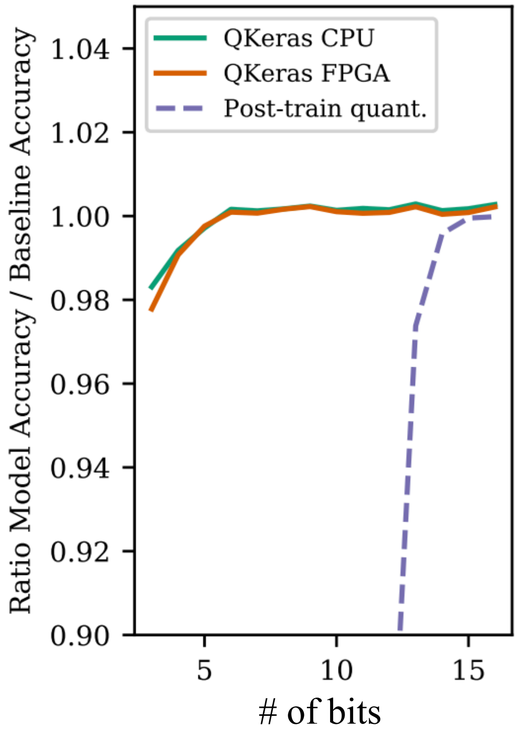
Pruning

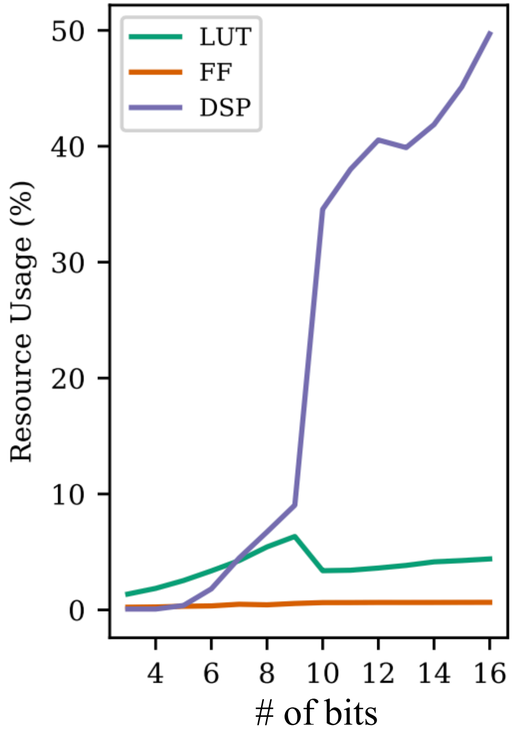
- Are all the pieces a given network necessary?
- Many different types of pruning
  - Structured vs. unstructured
- Multiplications by 0 can be completely removed from FPGA design



Quantization

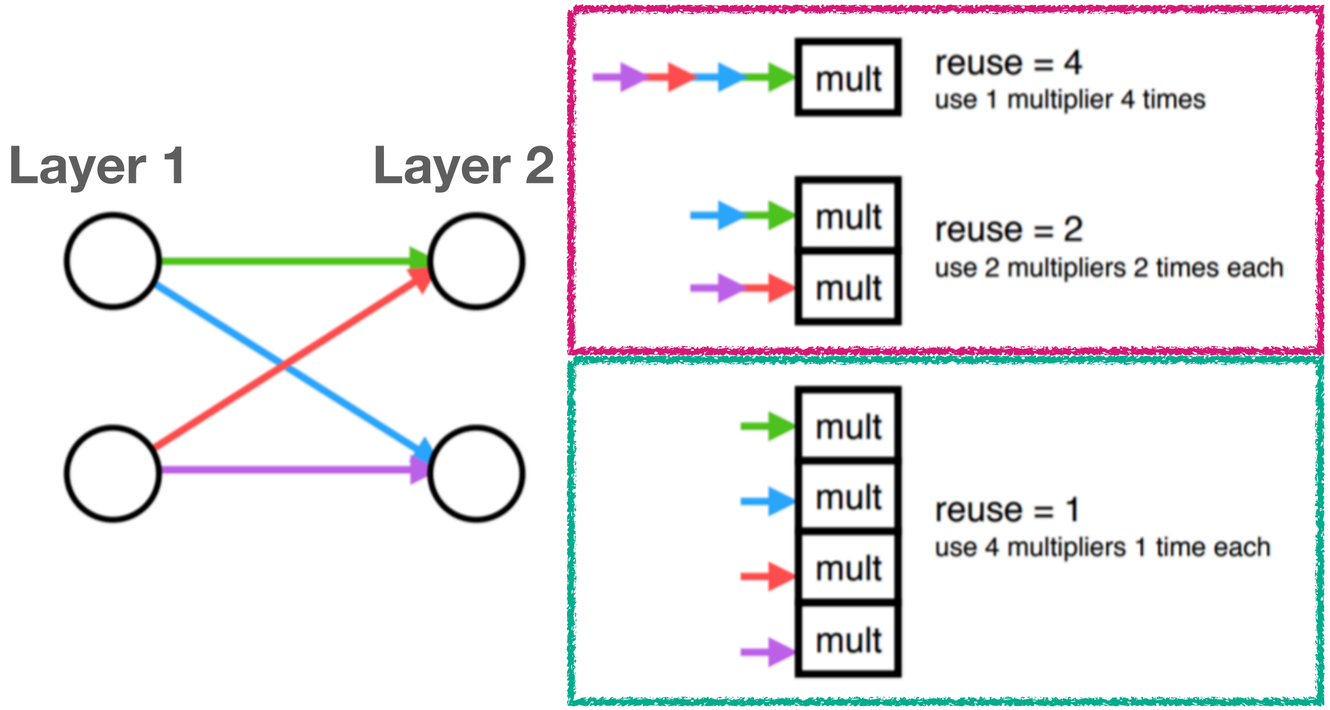
- FPGAs are well suited to fixed-point numbers, not floating point
- Number of bits can be adjusted as needed (impacts accuracy, performance, resources)
- Can greatly reduce number of bits needed by training with knowledge of quantization

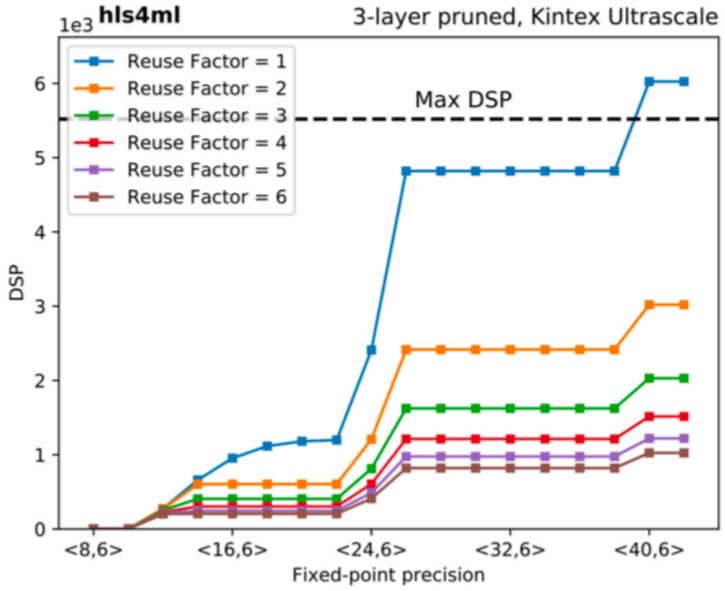


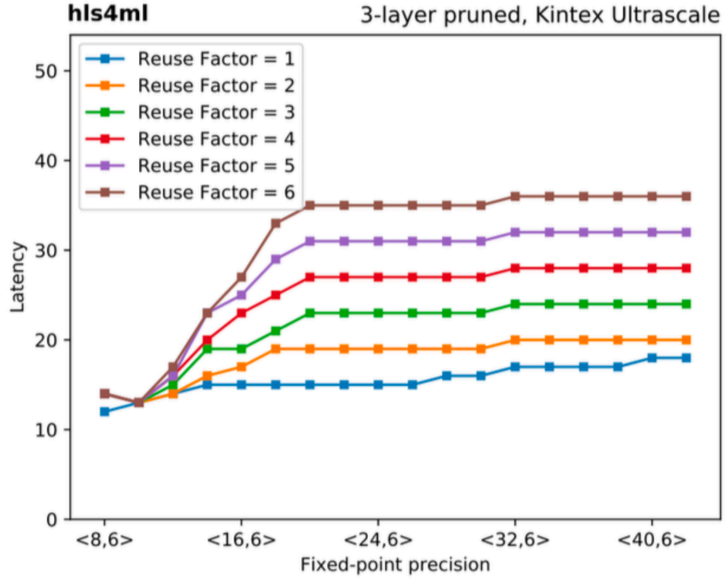


Reuse

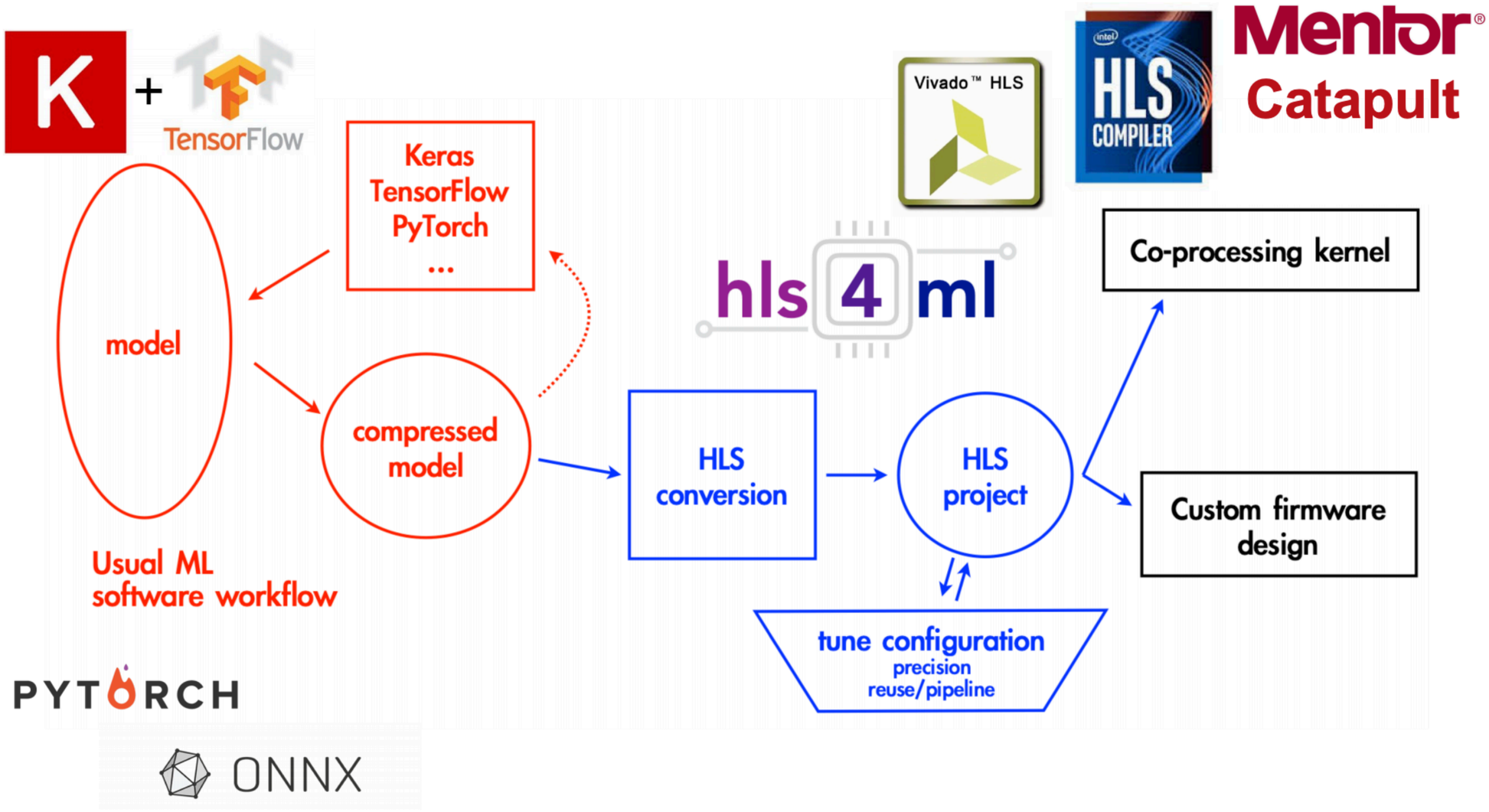
- For lowest latency, compute all multiplications at once
  - Reuse = 1 (fully parallel) → latency = # layers)
- Larger reuse implies more serialization
- Allows trading higher latency for lower resource usage







hls4ml Workflow



Inference on FPGAs

- Each part of network must be placed on the FPGA, connected together
- Cannot implement an algorithm if there are no resources left
  - Cannot just run things slower (25 ns!)