ML on the Edge at the LHC experiments

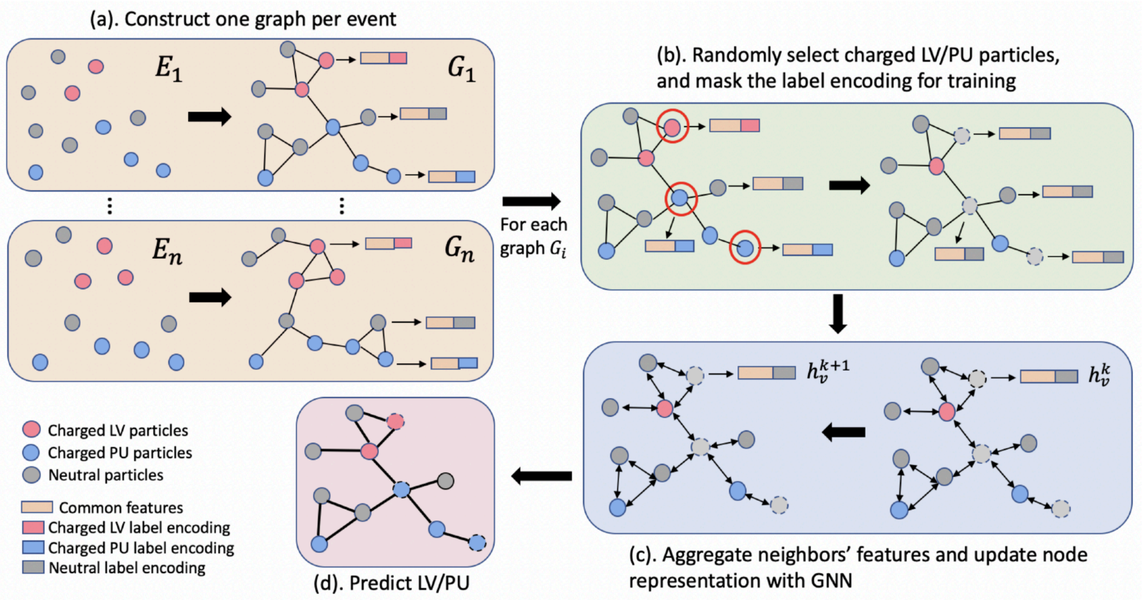
Dylan Rankin [UPenn]

SmartHEP Edge Machine Learning School

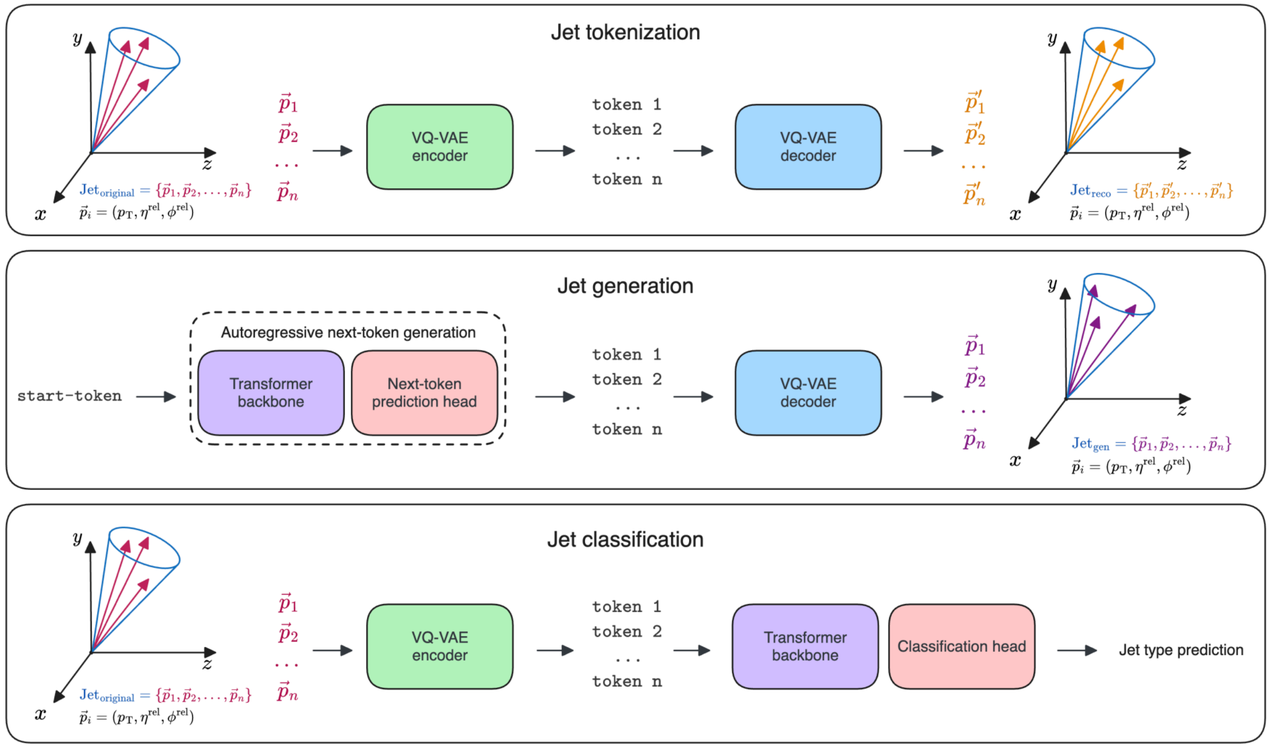
September 23rd, 2024

Introduction

- ML is becoming more and more popular, HEP and LHC is no exception
- HEP trends in ML towards bigger and more complicated models, more computing
- Availability of CPUs, GPUs, modern software has accelerated adoption



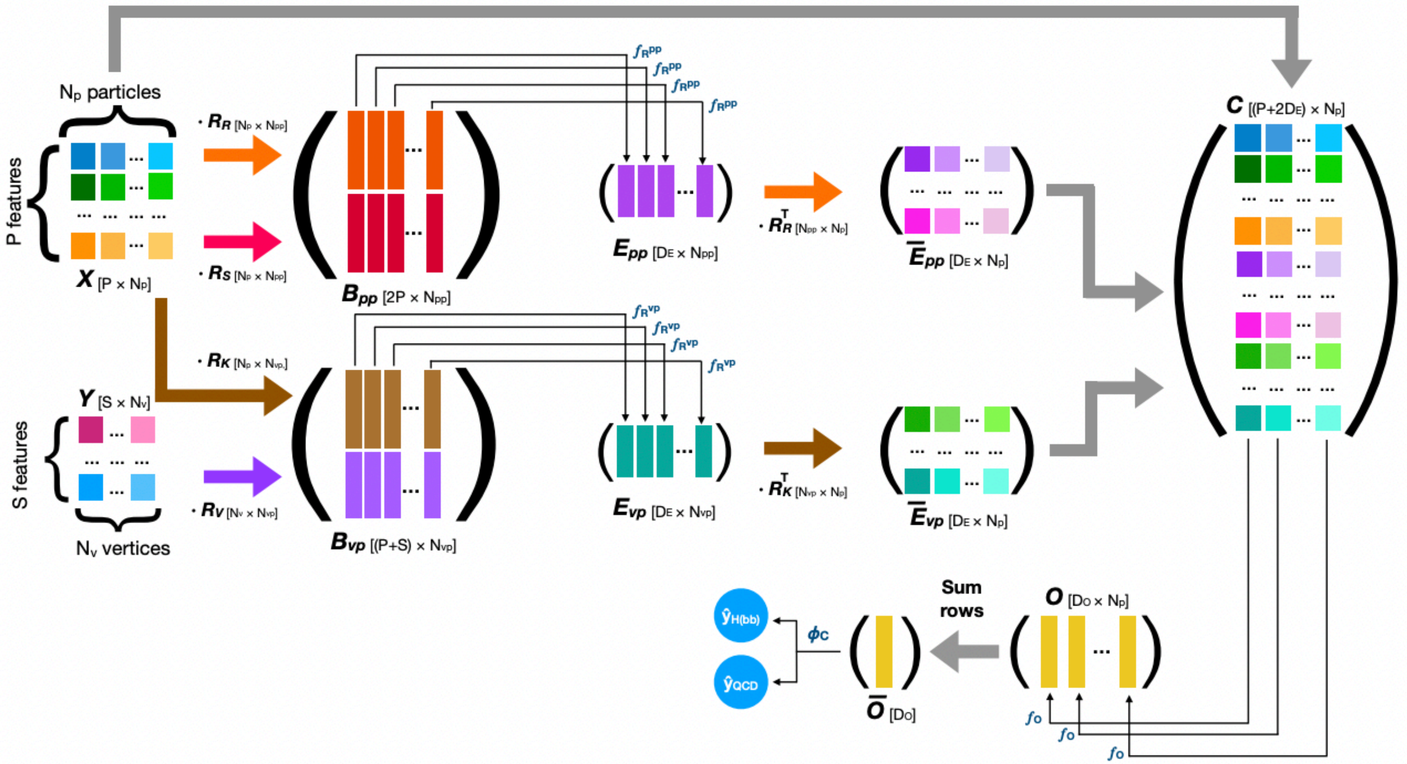
2203.15823



2403.05618

HEP ML

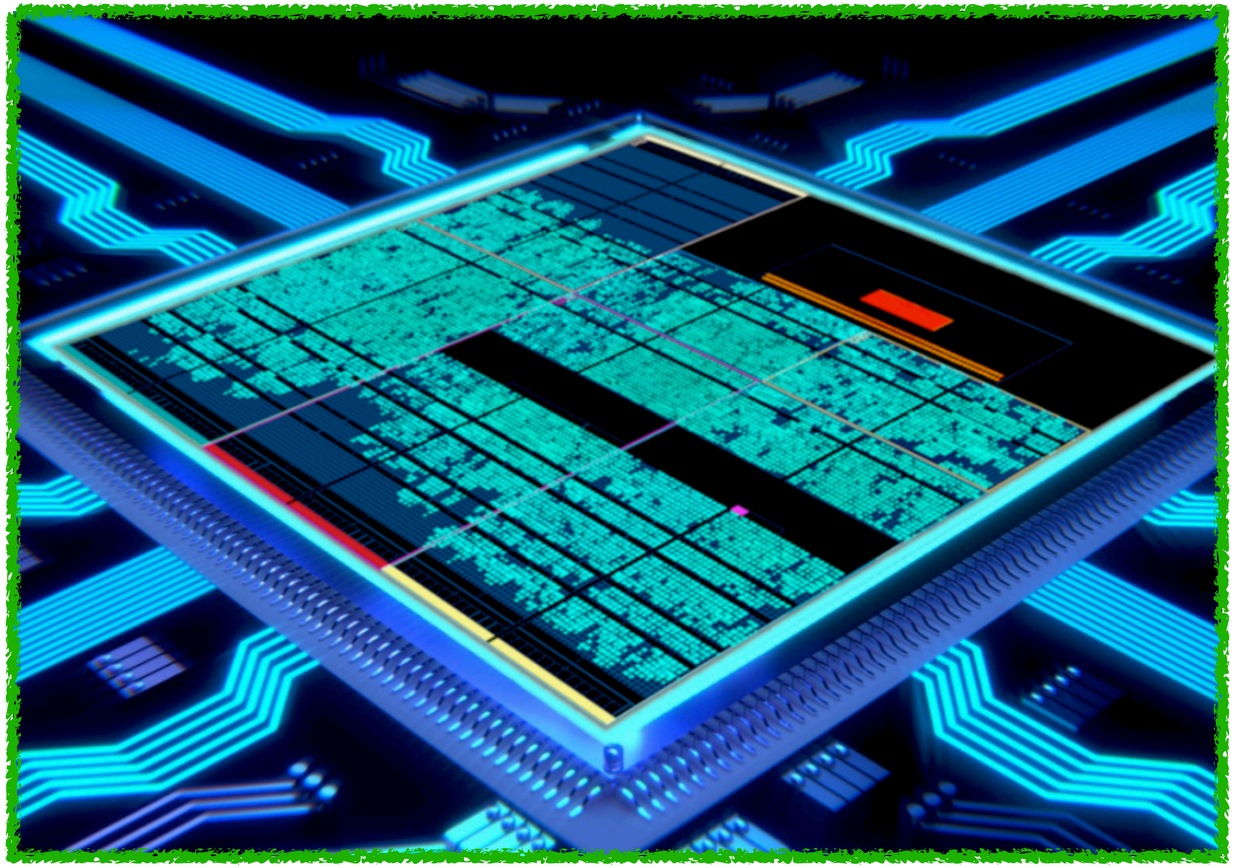
- **Majority of ML in physics is “off detector”**
  - System latency/resource limits are typically soft (if at all)
  - No radiation
  - Issues do not impact data collection
    - Can re-run algorithms/workflows



1909.12285

What if…

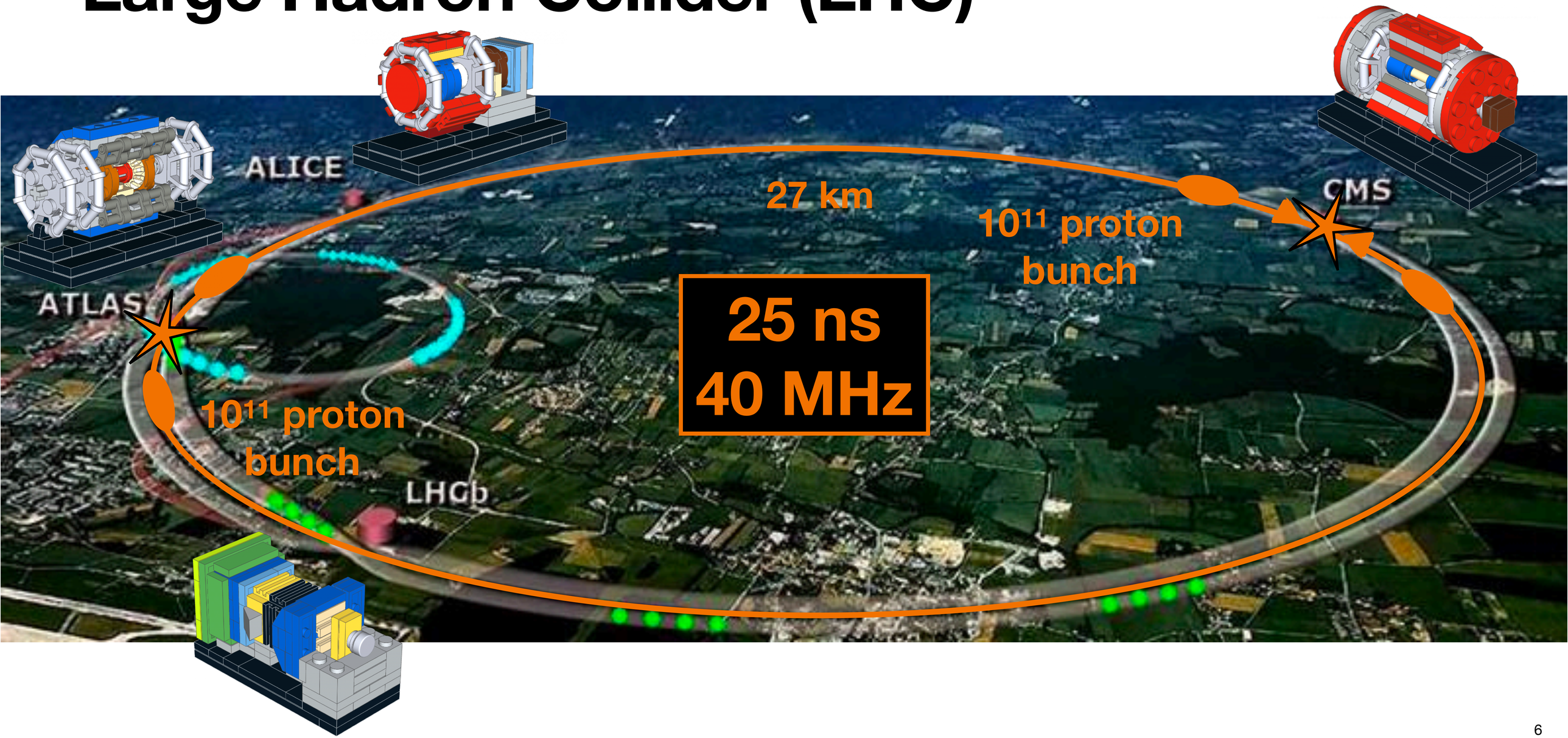
- What if:
  - System latency/resource limits are low?
  - High radiation?
  - No undo button?
- Requires dedicated hardware, strategies
- → **Edge ML**
  - “Edge ML is the process of running ML algorithms on computing devices at the periphery of a network to make decisions and predictions as close as possible to the originating source of data.”
  - Placing ML at sensors, running in real-time



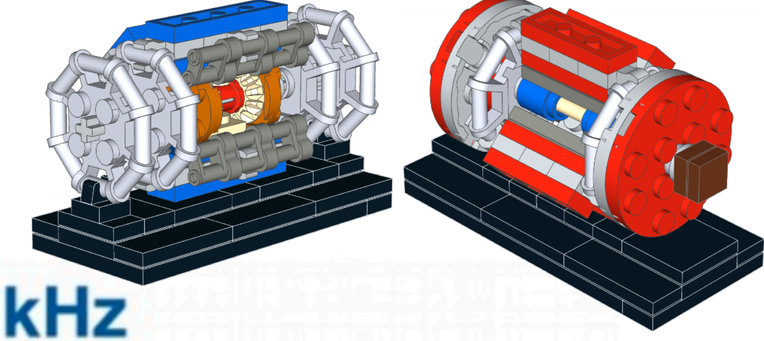
Edge ML

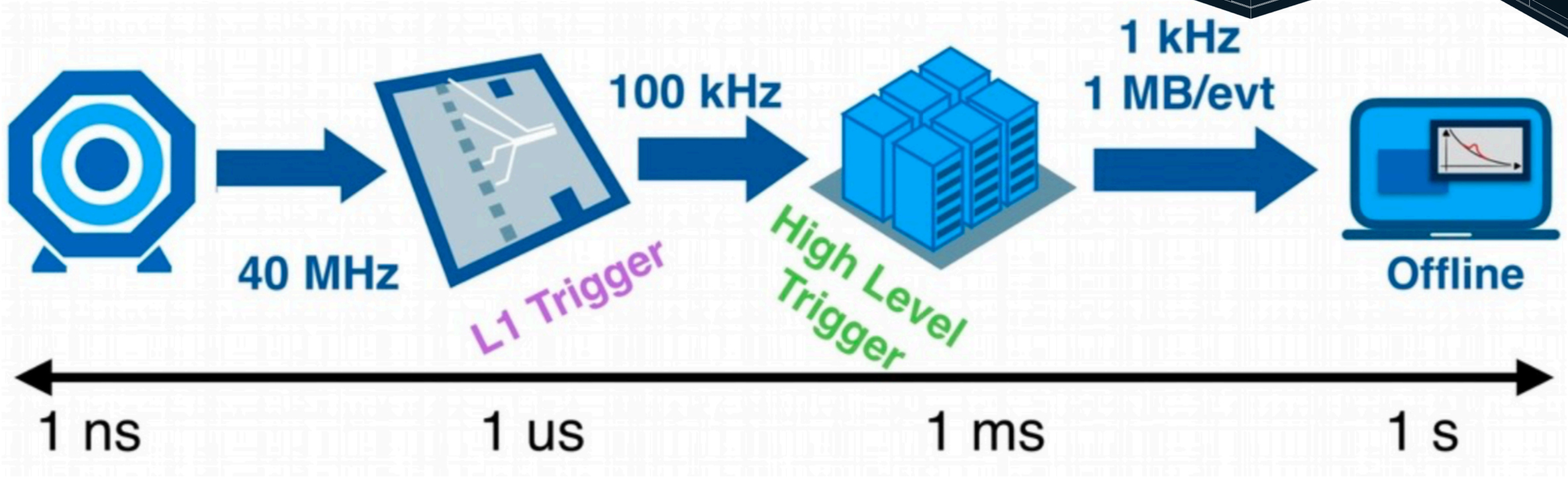
- Want to focus on two main components of edge ML:
- Specialized hardware → specialized tools
  - FPGAs, ASICs, GPUs (and more)
- Data source, format must be considered to be effective
  - How does data get to specialized device?
  - Does it arrive all at once?
  - Does it come with the features I want already?
  - Are there limitations from the environment/device/task?

Large Hadron Collider (LHC)



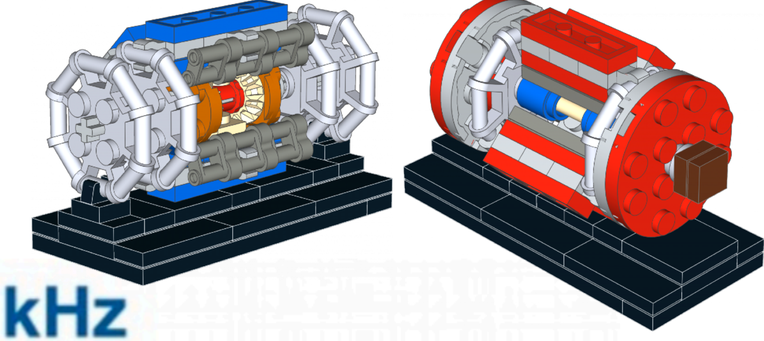
LHC Data Processing / Readout

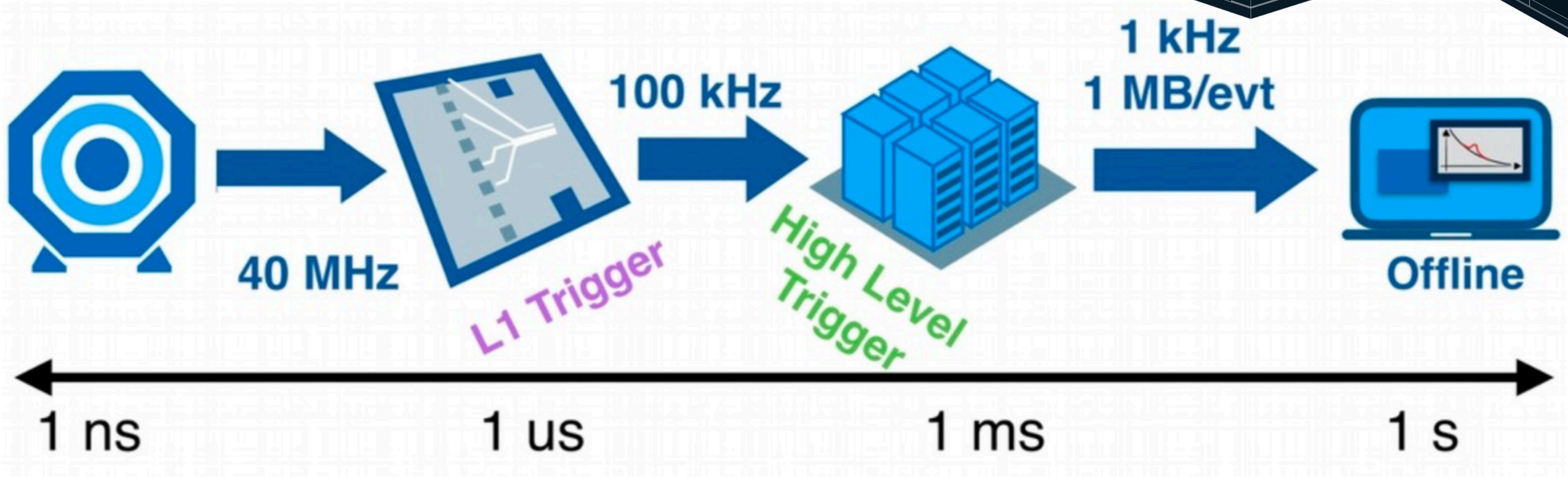




- **Level-1 Trigger** (FPGAs, ASICs) - O(μs) hard latency
- **High Level Trigger** (CPUs, GPUs, FPGAs?) - O(100 ms) soft latency
- **Offline** (CPUs, GPUs) → 1 s latencies

LHC Data Processing / Readout

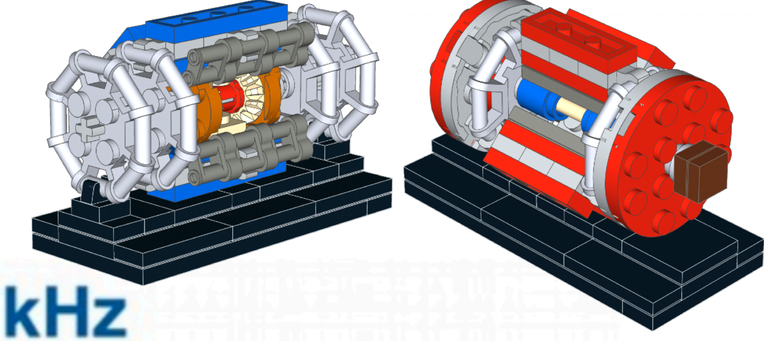


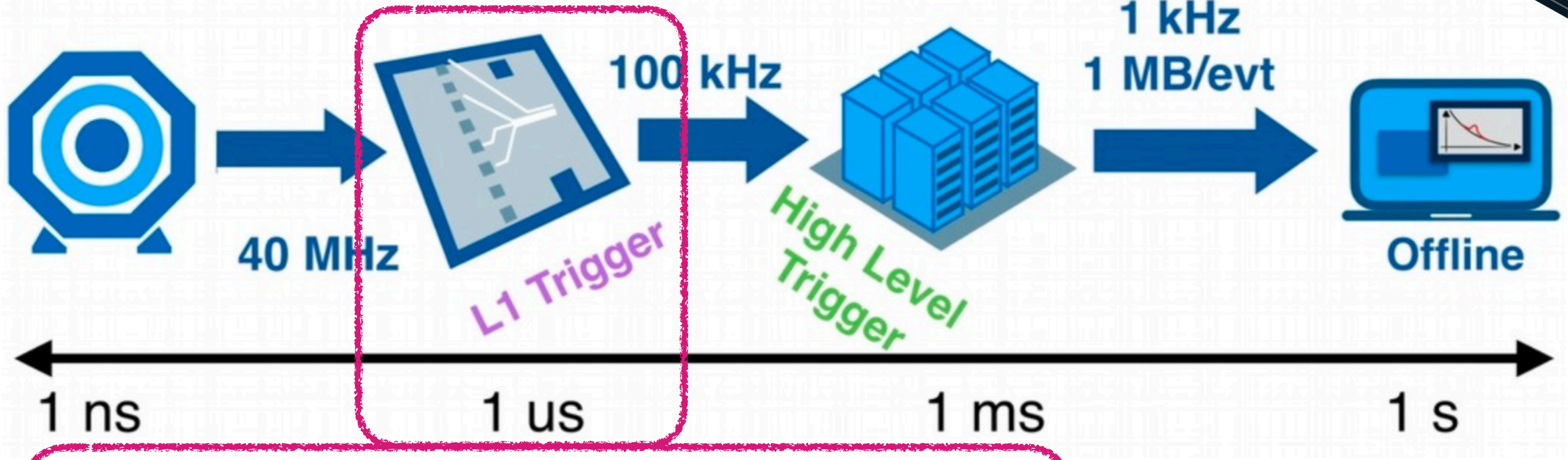


- **Level-1 Trigger** - O(μs) latency
- **High Level Trigger** - O(100 ms) latency
- **Offline** → 1 s latencies

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

LHC Data Processing / Readout





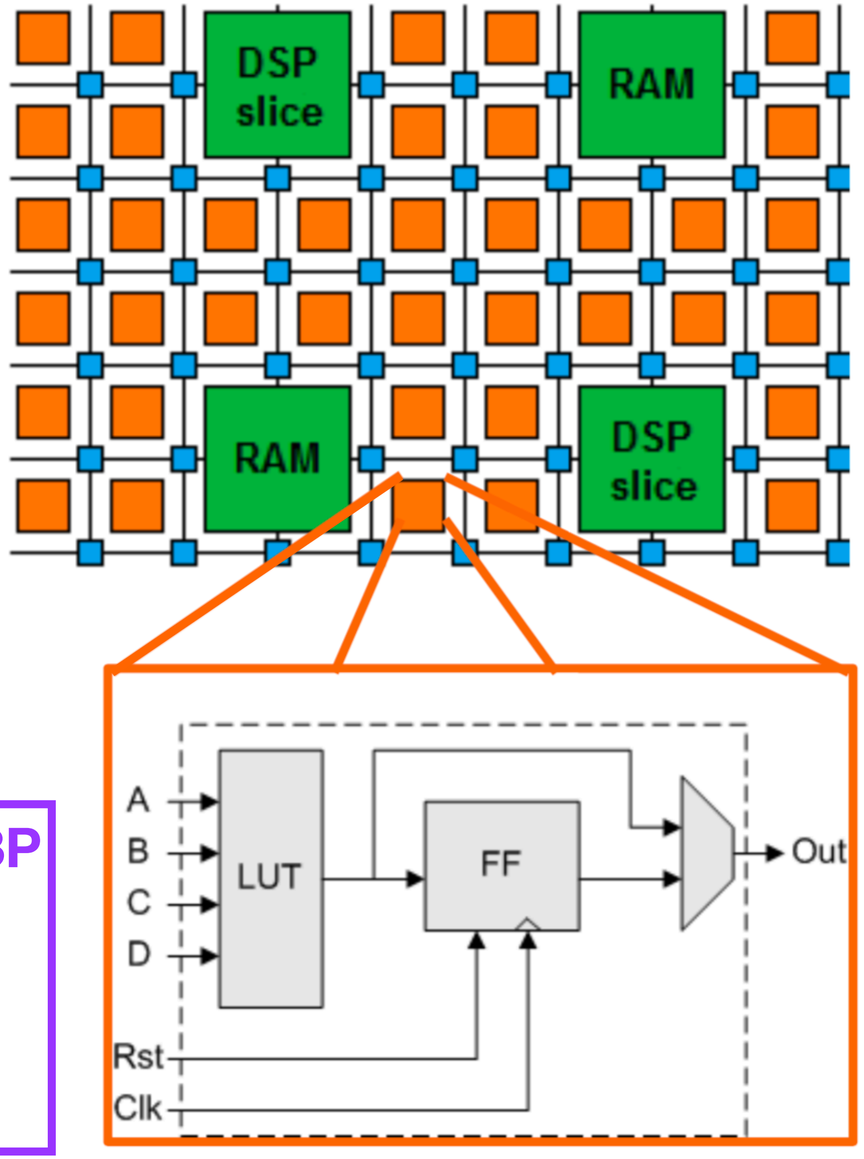
- **Level-1 Trigger** - O(μs) latency
- **High Level Trigger** - O(100 ms) latency
- **Offline** → 1 s latencies

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

What is an FPGA?

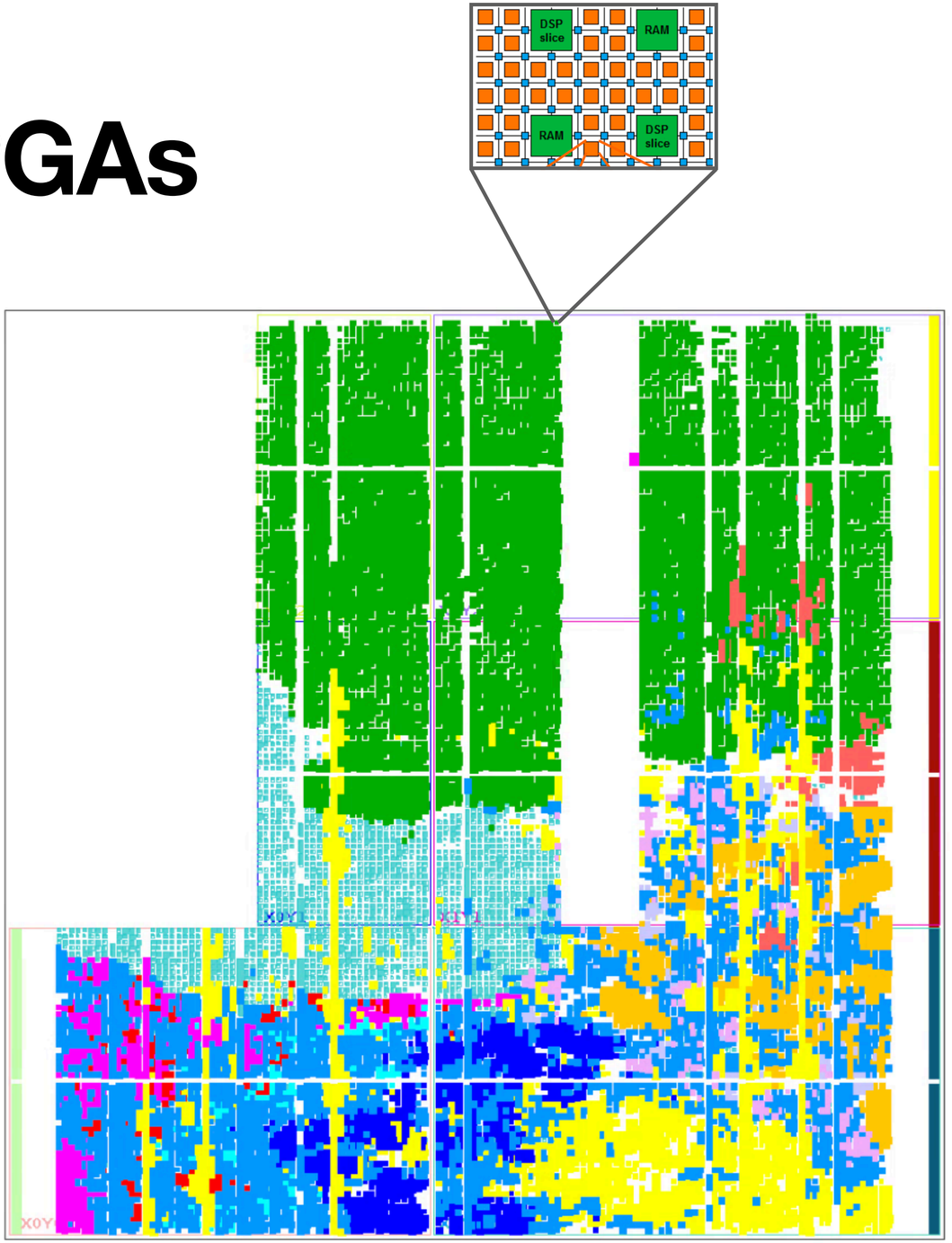
- **F**ield-**p**rogrammable **g**ate **a**rray
- Building blocks:
  - **Multiplier units (DSPs) [arithmetic]**
  - **Look Up Tables (LUTs) [logic]**
  - **Flip-flops (FFs) [registers]**
  - **Block RAMs (BRAMs) [memory]**
- Algorithms are wired onto the chip
  - Can only use the resources on the chip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

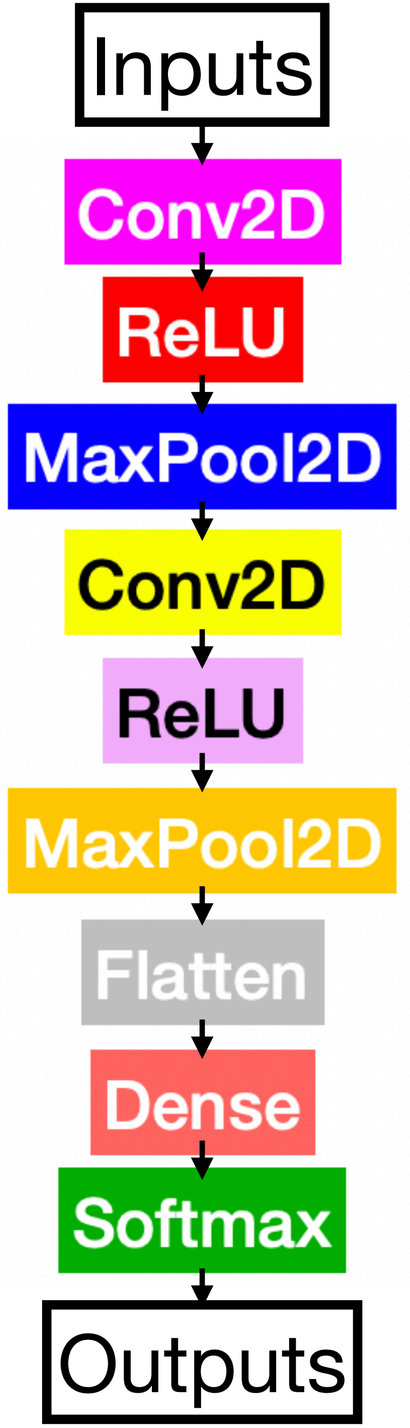
Xilinx Virtex Ultrascale+ VU13P
12288 Multipliers
1.7M LUTs
3.4M FFs
95 Mb BRAM



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!)





Many Tools (Tutorials this week)

- NNs:

arXiv: 1804.06913

arXiv: 2004.03021

- Boosted Decision Trees (BDTs):

arXiv: 2002.02534

arXiv: 2104.03408

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

ML Size / Complexity

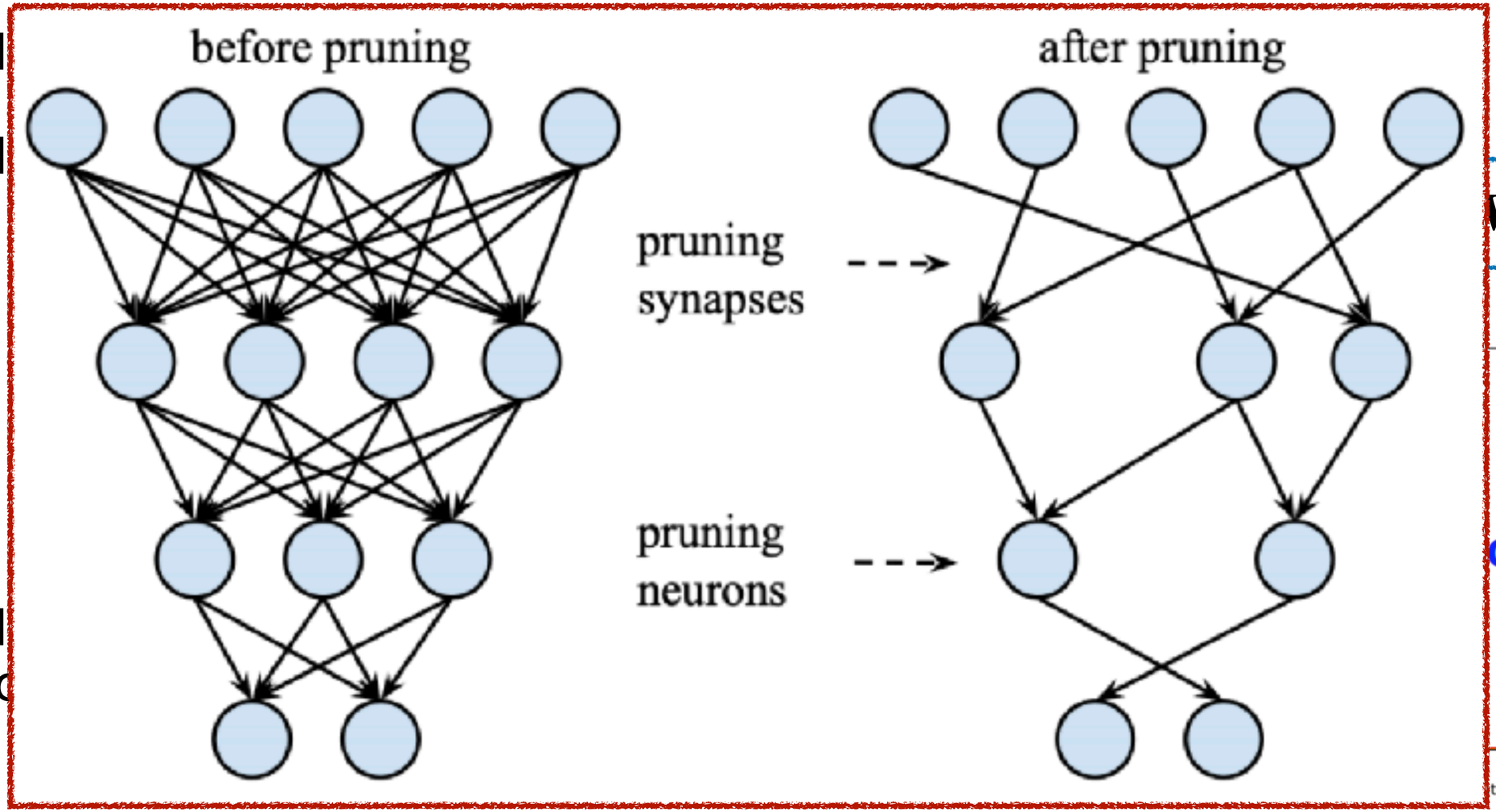
- Regardless of toolkit, limitation of doing edge ML is device size
  - Bigger device → more resources → more computation → larger ML models

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

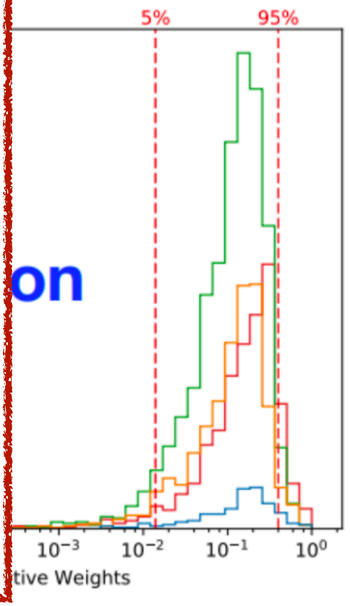
Pruning

- Are all the pieces a given network necessary?
- M
- M
  -
  -
  -
- M
  fr



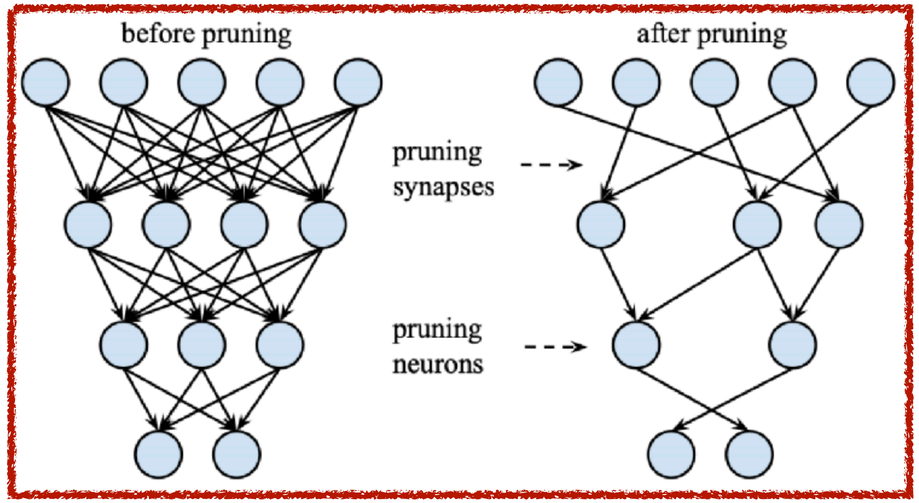
$$v) + \lambda \|\mathbf{w}\|$$

on

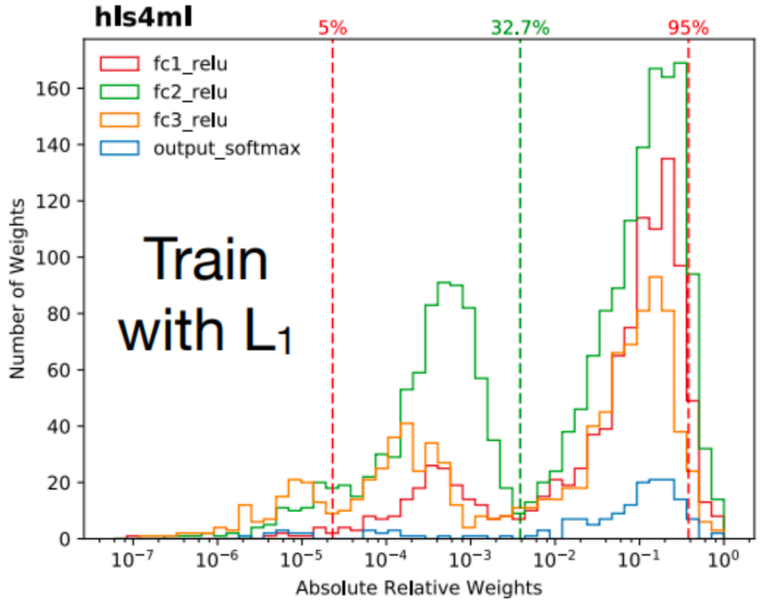


Pruning

- Are all the pieces a given network necessary?
- Many different types of pruning
- Magnitude-based:
  - Use regularization (penalty term in loss function for large weights)
  - Remove smallest weights
  - Repeat
- Multiplications by 0 can be completely removed from FPGA design

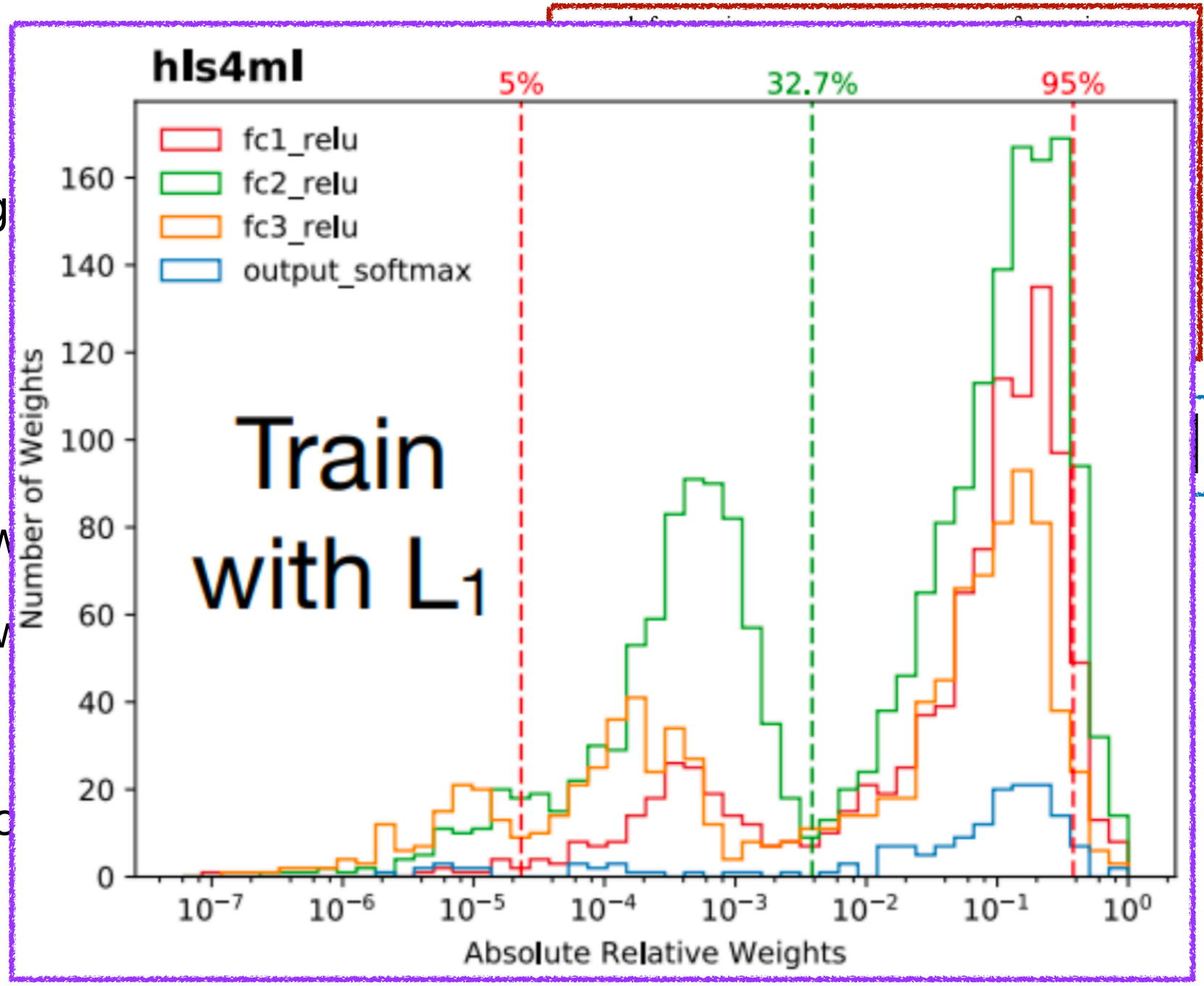


$$L_\lambda(\mathbf{w}) = L(\mathbf{w}) + \lambda \|\mathbf{w}\|$$



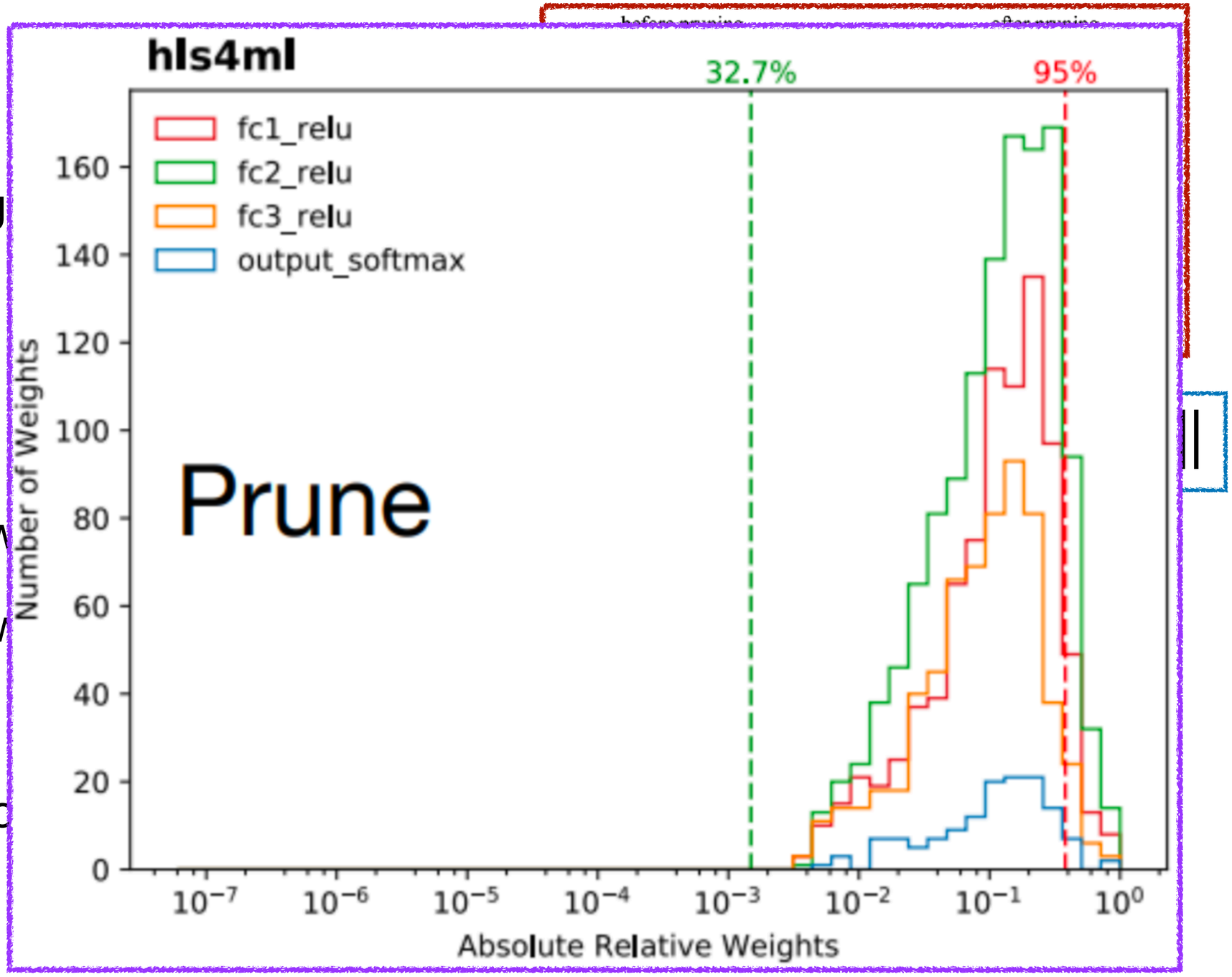
Pruning

- Are all the pieces a g
- Many different types
- Magnitude-based:
  - Use regularization function for large w
  - Remove smallest w
  - Repeat
- Multiplications by 0 c from FPGA design



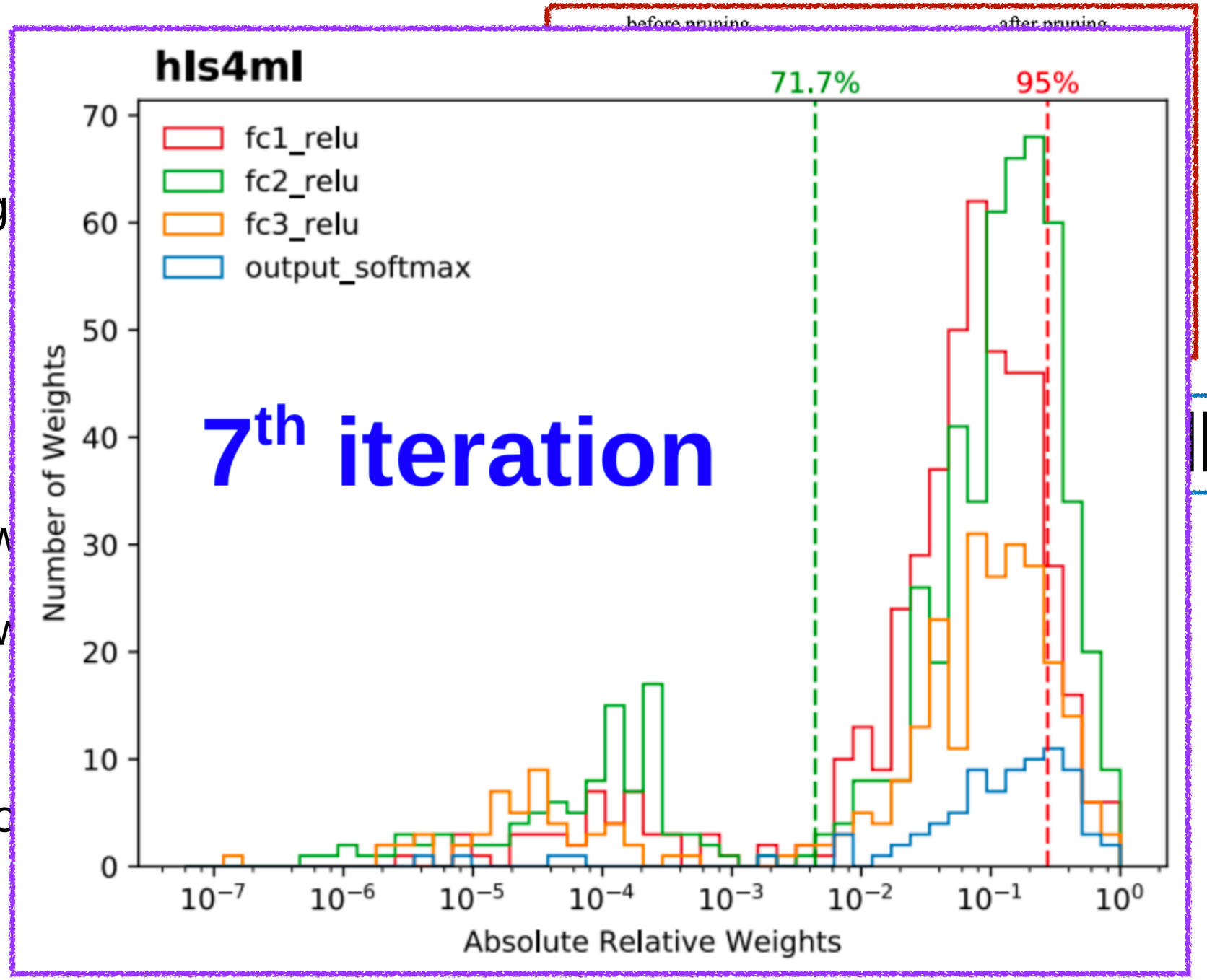
Pruning

- Are all the pieces a g
- Many different types
- Magnitude-based:
  - Use regularization function for large w
  - Remove smallest w
  - Repeat
- Multiplications by 0 c from FPGA design



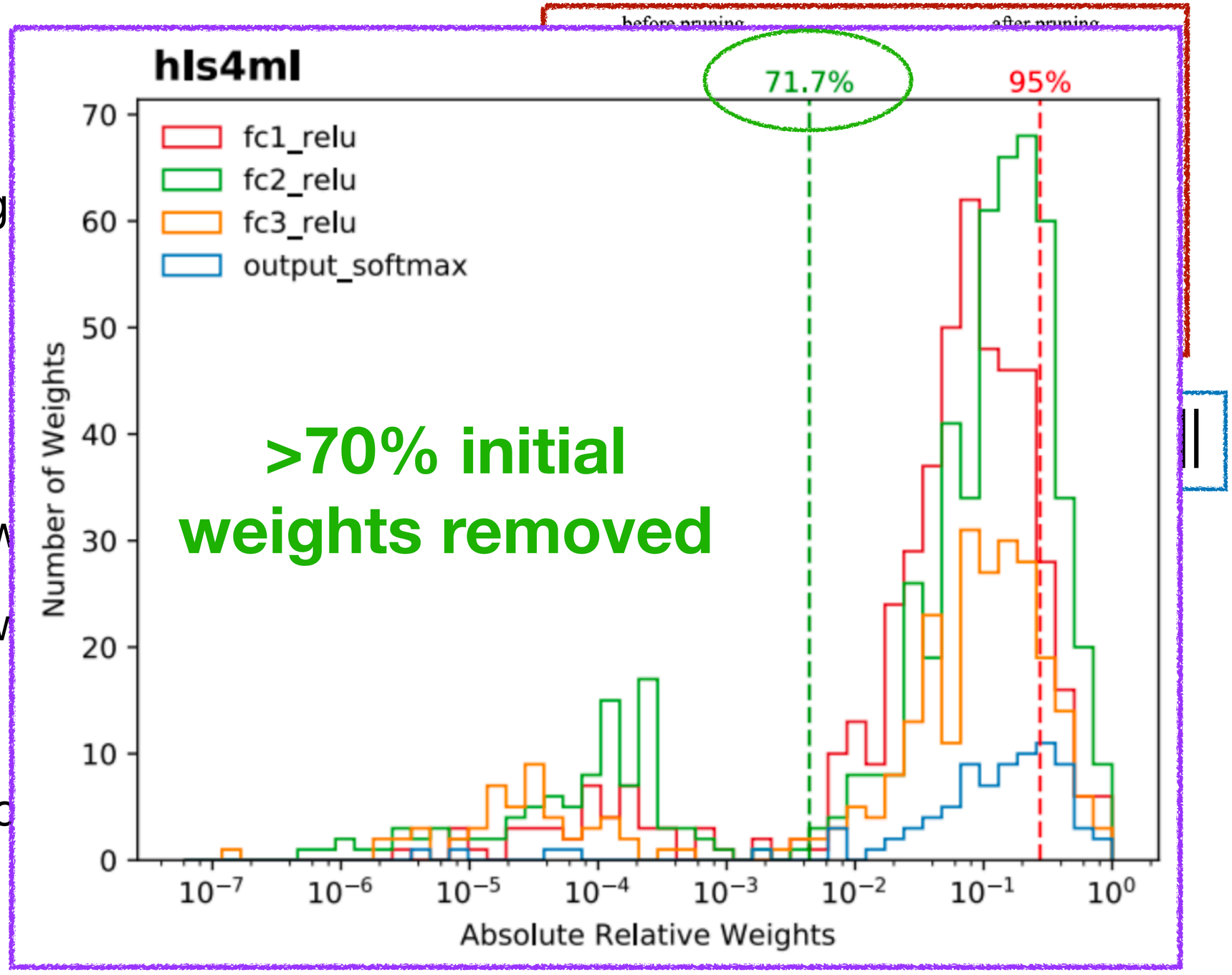
Pruning

- Are all the pieces a g
- Many different types
- Magnitude-based:
  - Use regularization function for large w
  - Remove smallest w
  - Repeat
- Multiplications by 0 c from FPGA design



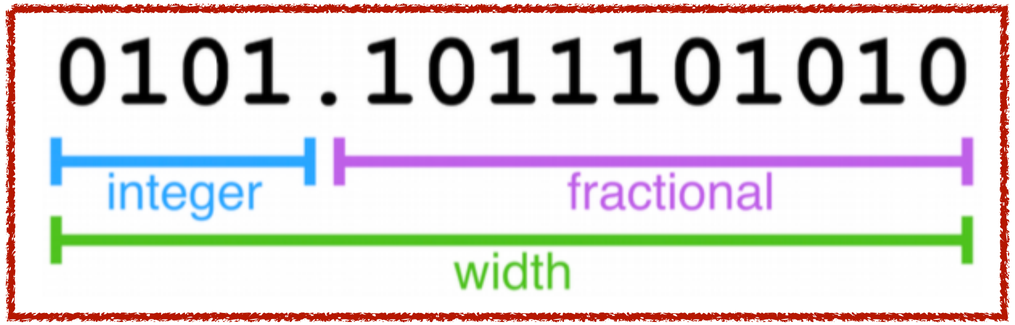
Pruning

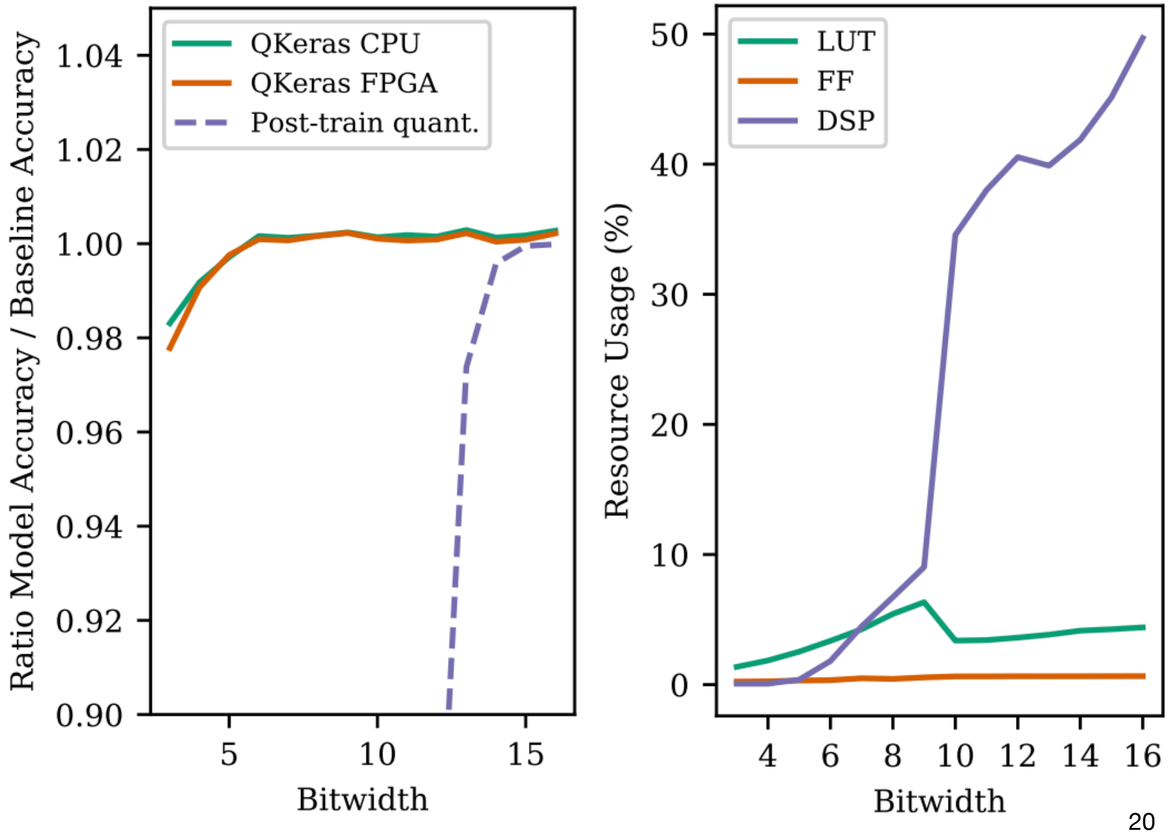
- Are all the pieces a g
- Many different types
- Magnitude-based:
  - Use regularization function for large w
  - Remove smallest w
  - Repeat
- Multiplications by 0 c from FPGA design



Quantization

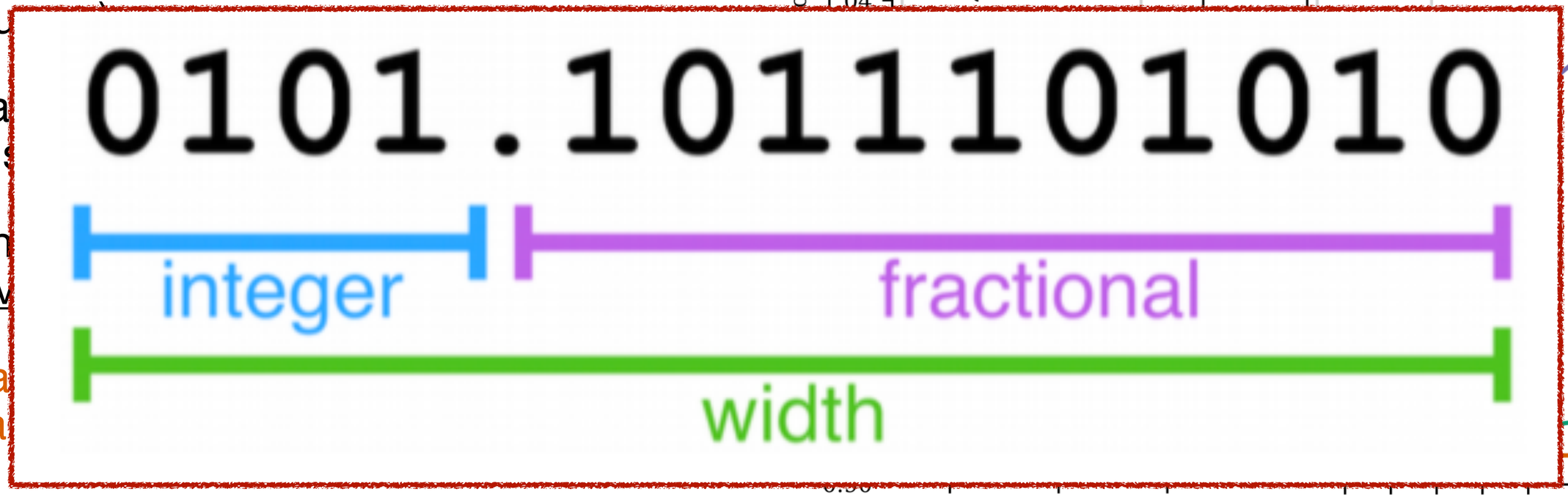
- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, resources)
  - Can be combined with other customizations
- Quantization-aware training [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization





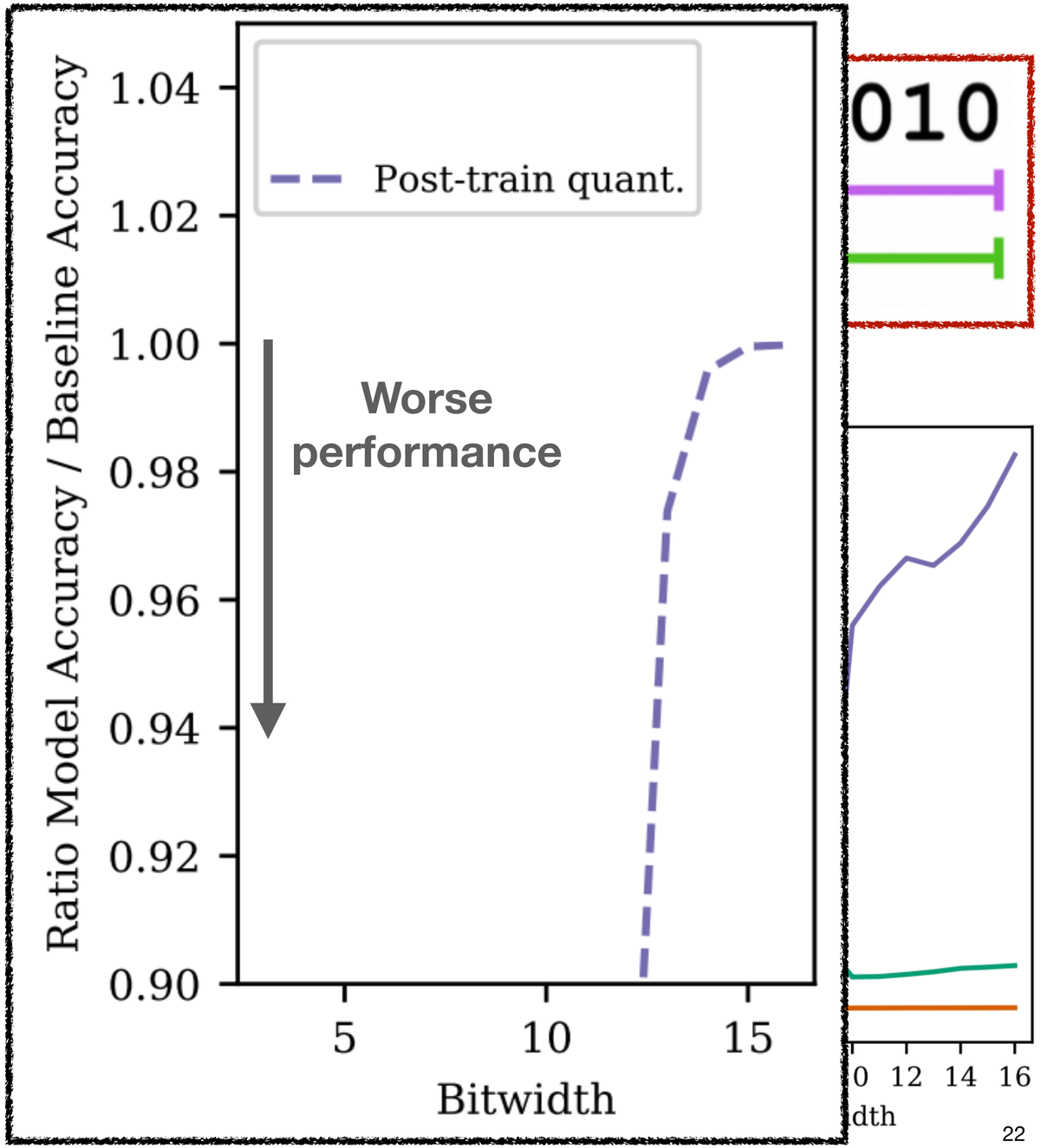
Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, resou
  - Ca
    cus
- Quan
  [arXiv
  - Ca
    tra



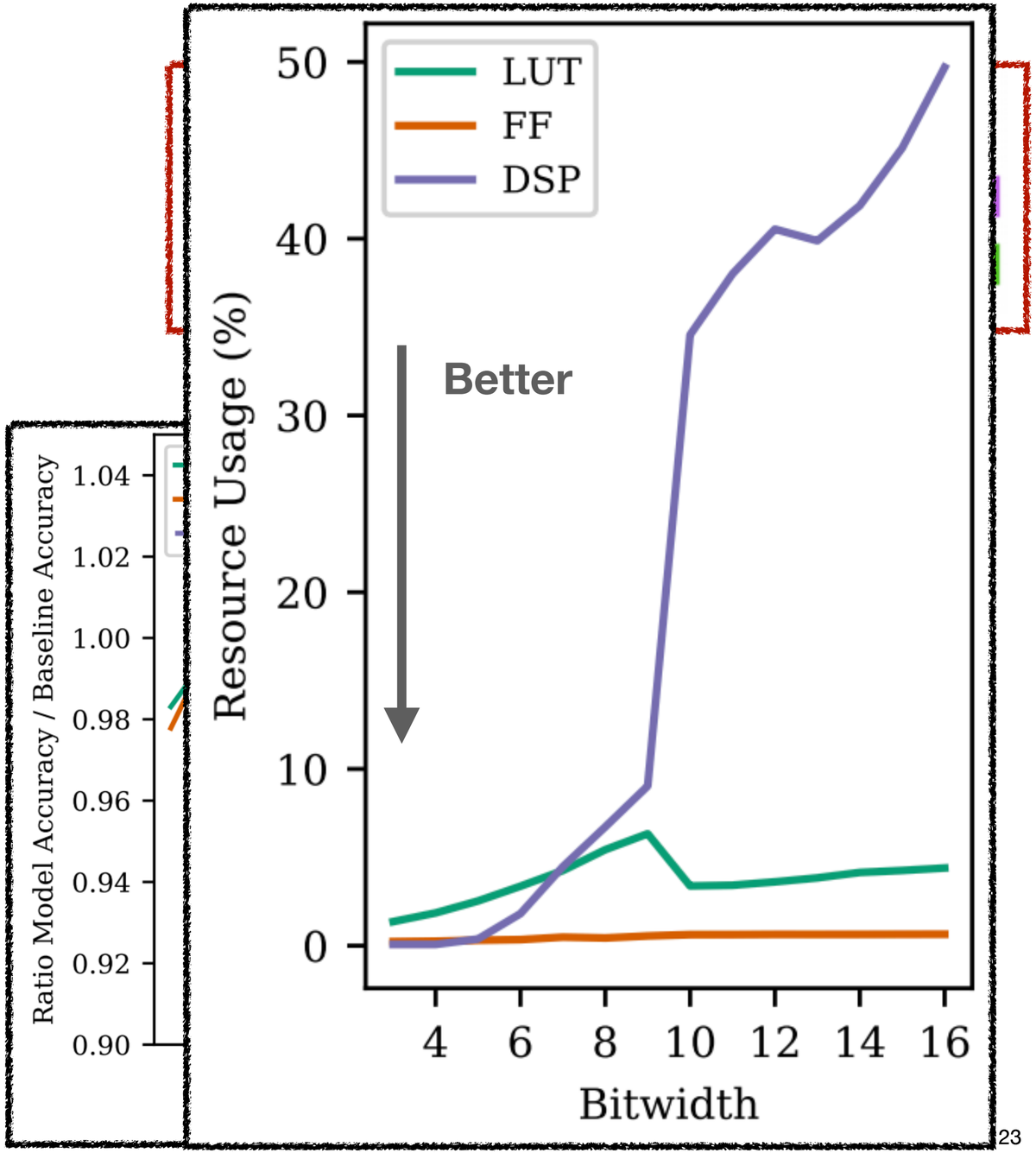
Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, **performance**, resources)
  - Can be combined with other customizations
- Quantization-aware training [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization



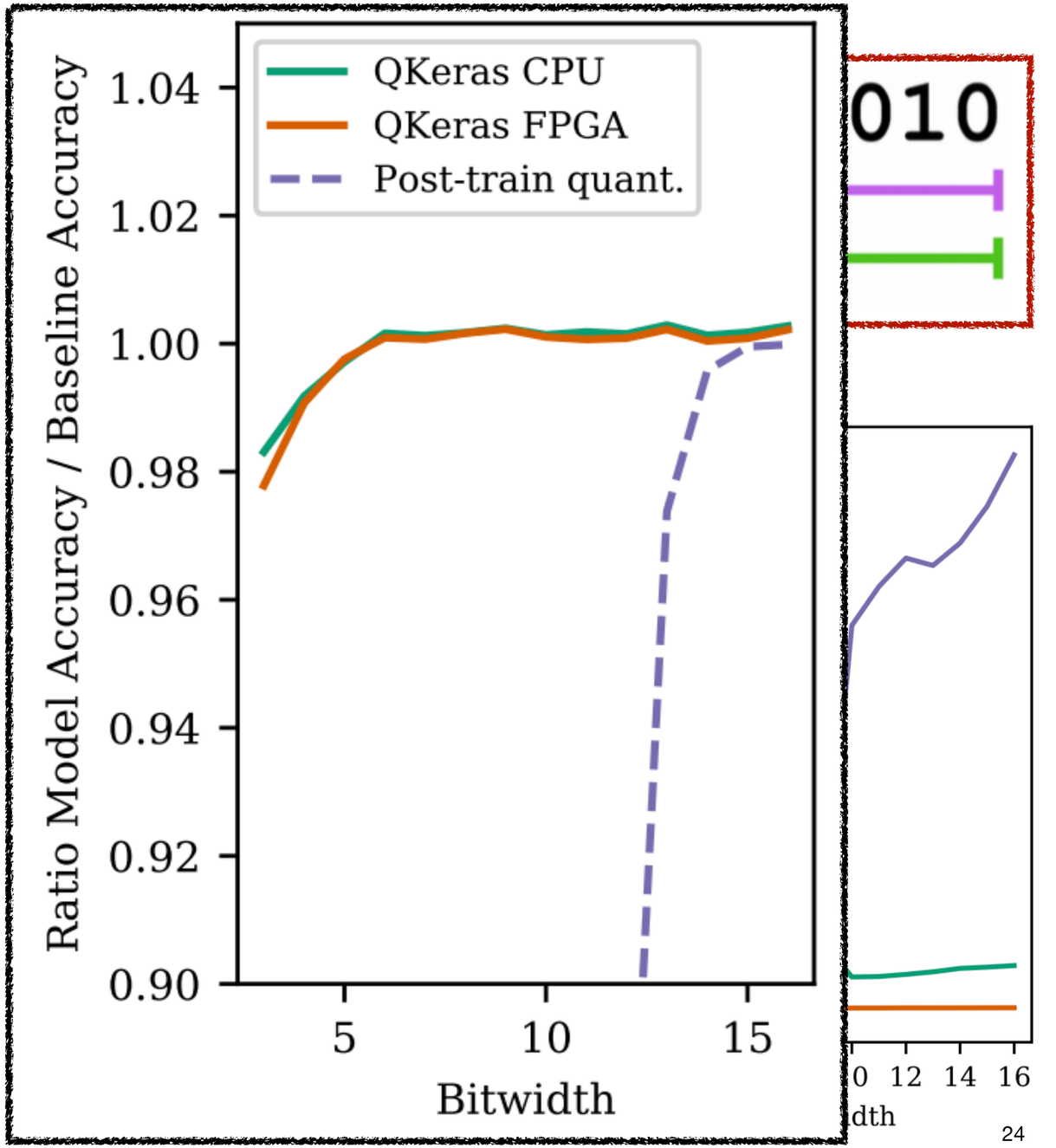
Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, **resources**)
  - Can be combined with other customizations
- Quantization-aware training [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization



Quantization

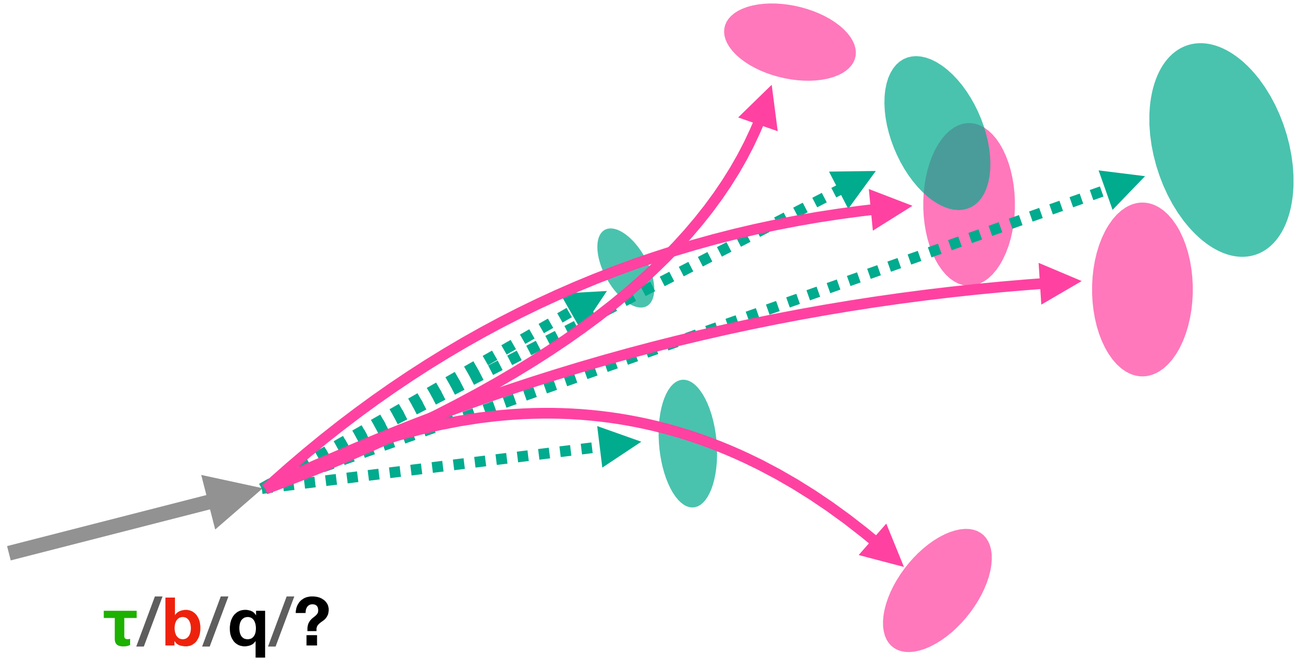
- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, resources)
  - Can be combined with other customizations
- **Quantization-aware training** [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization

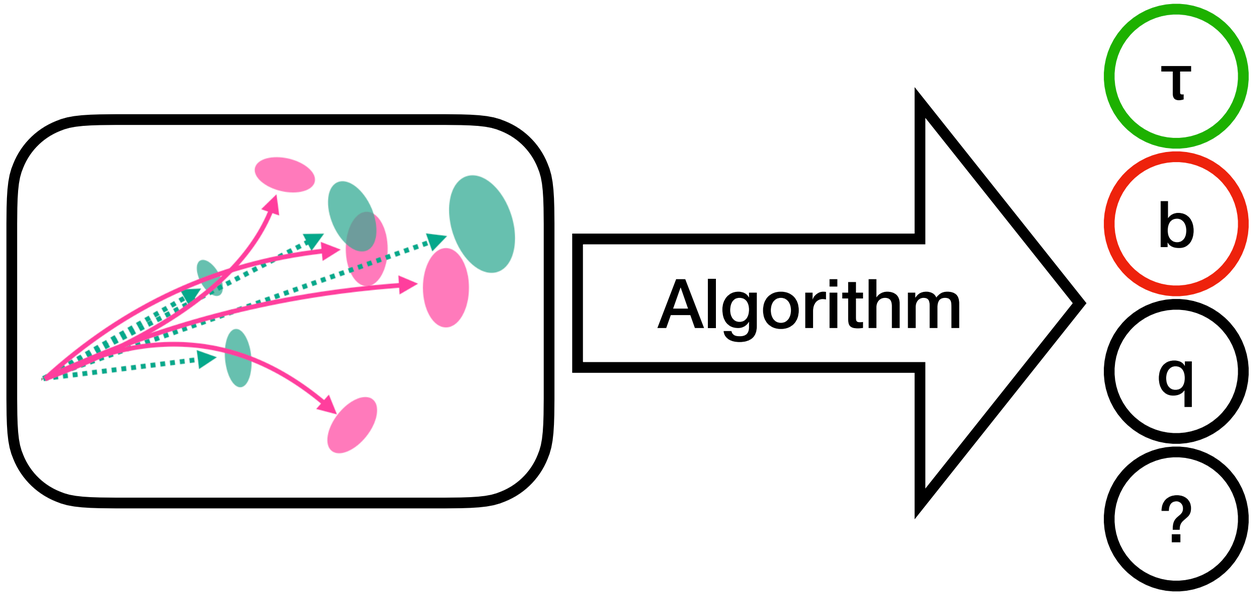


LHC Applications

Particle Identification

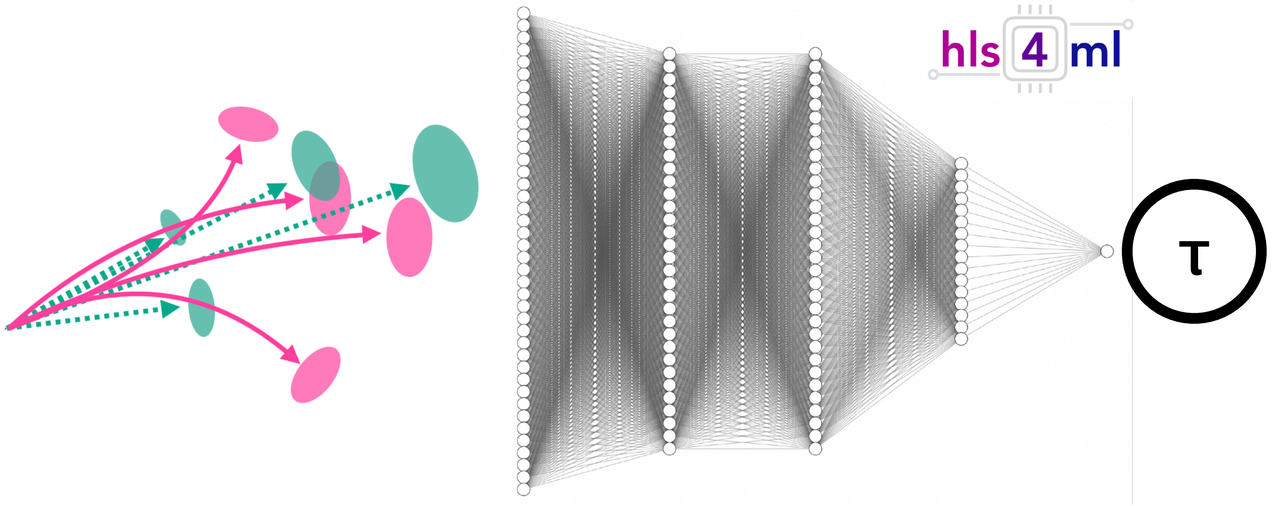
- LHC triggers must differentiate different collections of particles / detector signals from overwhelming backgrounds
  - τ lepton, bottom quark
  - Light quarks, gluons, noise, combinatorics
- Edge ML can enable this faster / better

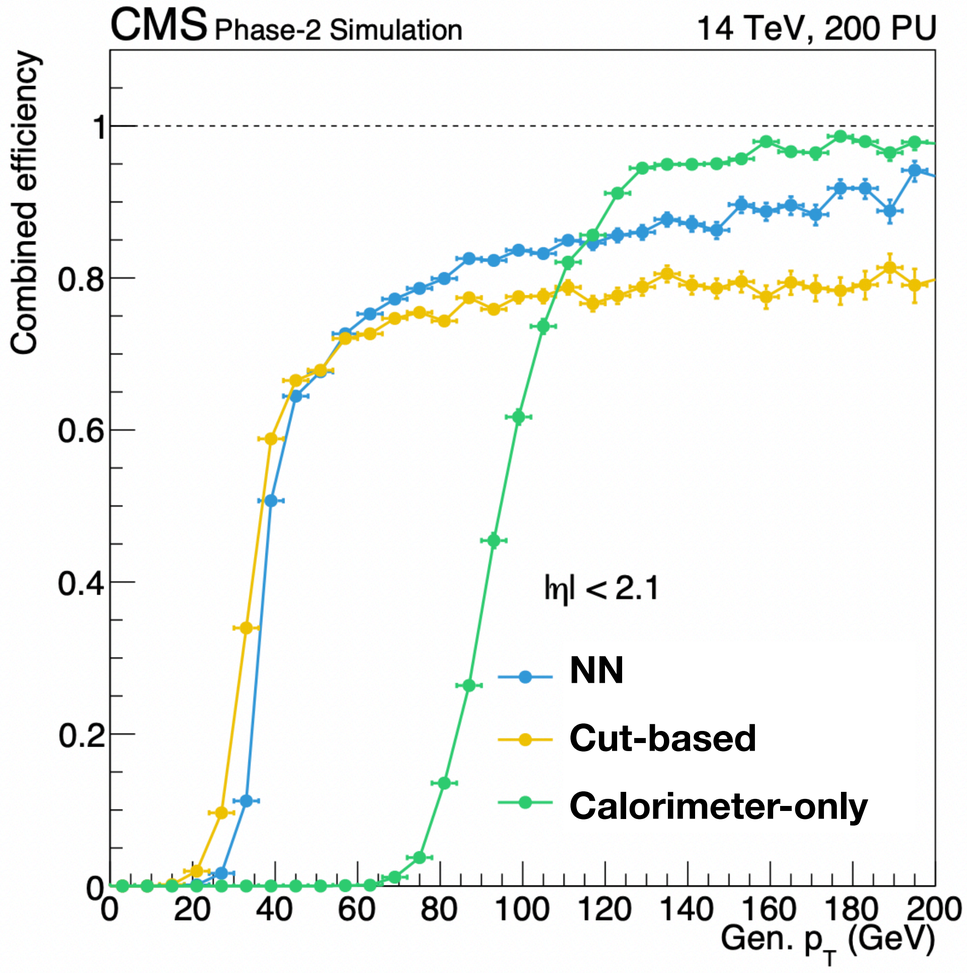




L1 τ Identification

- **NN algorithm** capable of accepting more τ leptons than traditional **cut-based method**
- Network is 3 layer dense model, uses information about particle p_T, η, ϕ, and type
- Outputs decision in 38 ns (9 clocks @ 240 MHz)

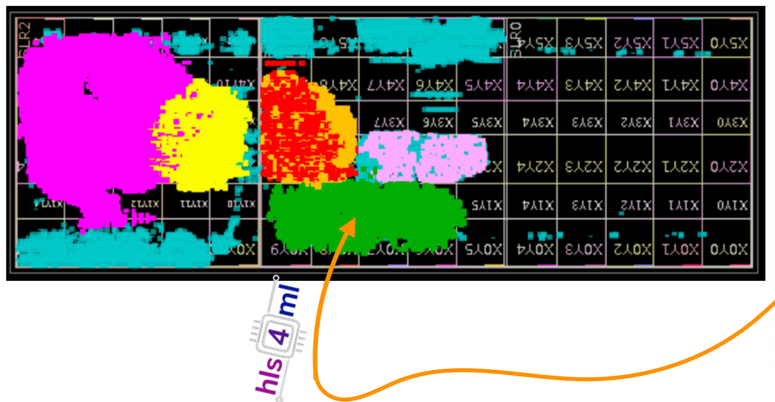


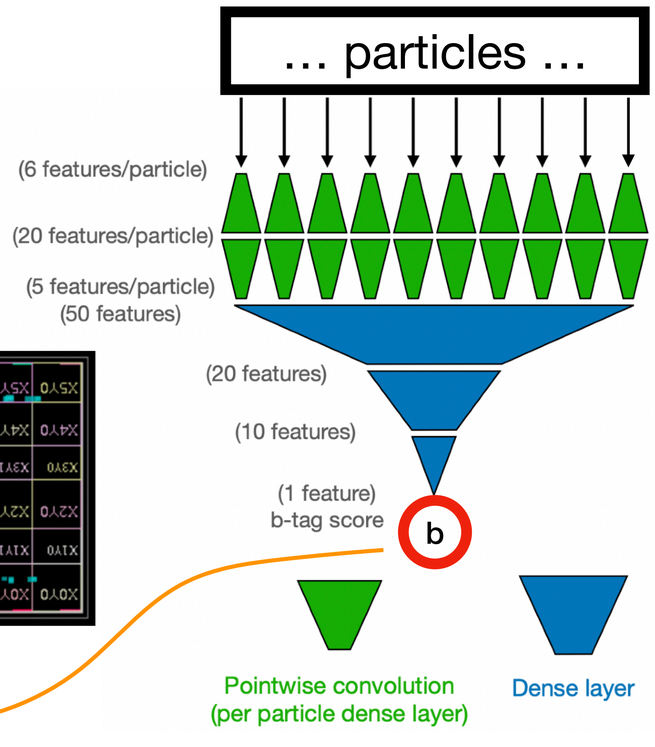


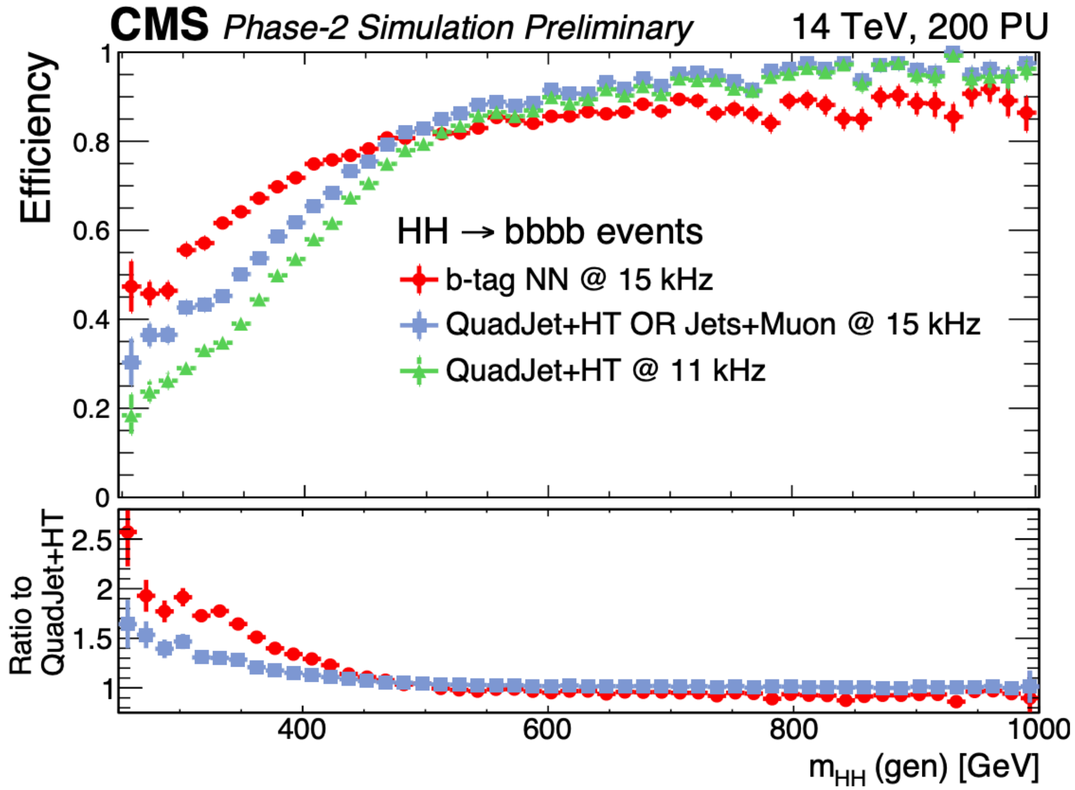
CMS TDR-021

L1 b-quark Identification

- NN trained to identify b-quarks using collection of particles
- Architecture includes featurizers that act on each particle individual
- **Significantly improved acceptance for HH→bbbb events with low m_{HH}** (compared to traditional cut-based methods)



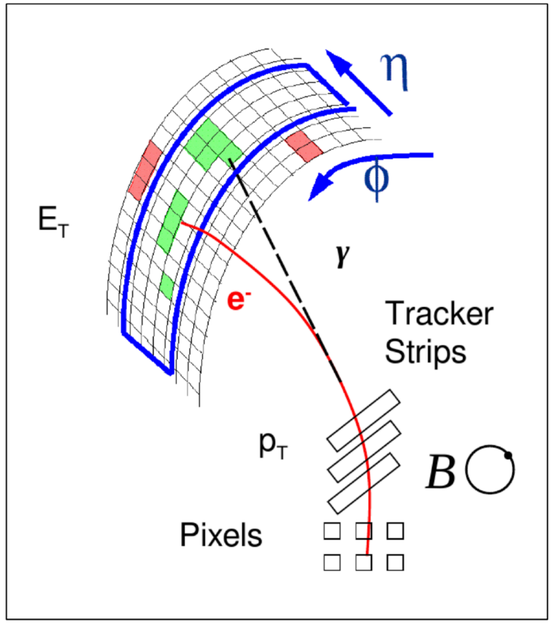


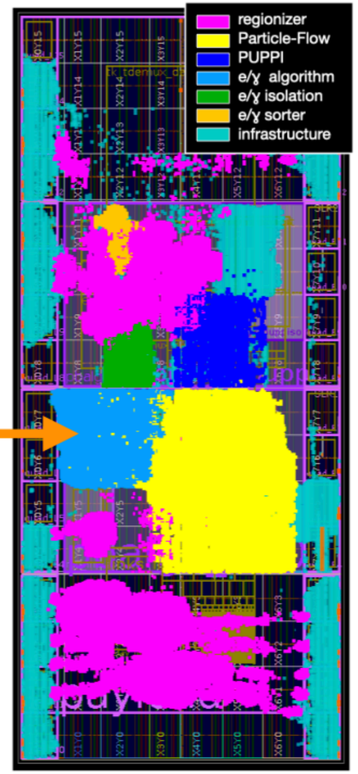


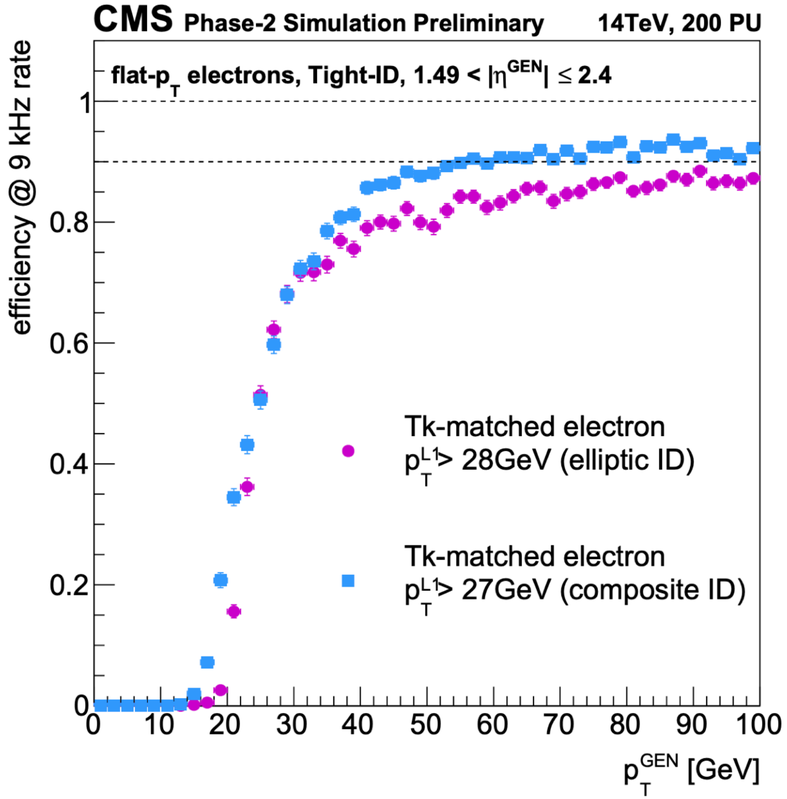
CMS DP-2022/021

L1 Electron Identification

- Electrons are complex signatures
  - Multiple sub detectors (tracker & calorimeter)
  - Undergo bremsstrahlung (e → e + γ)
- Edge ML well-suited to electron ID
  - Handles correlations between different inputs
  - 5-10% improvement in plateau efficiency
- Important for many different physics signatures

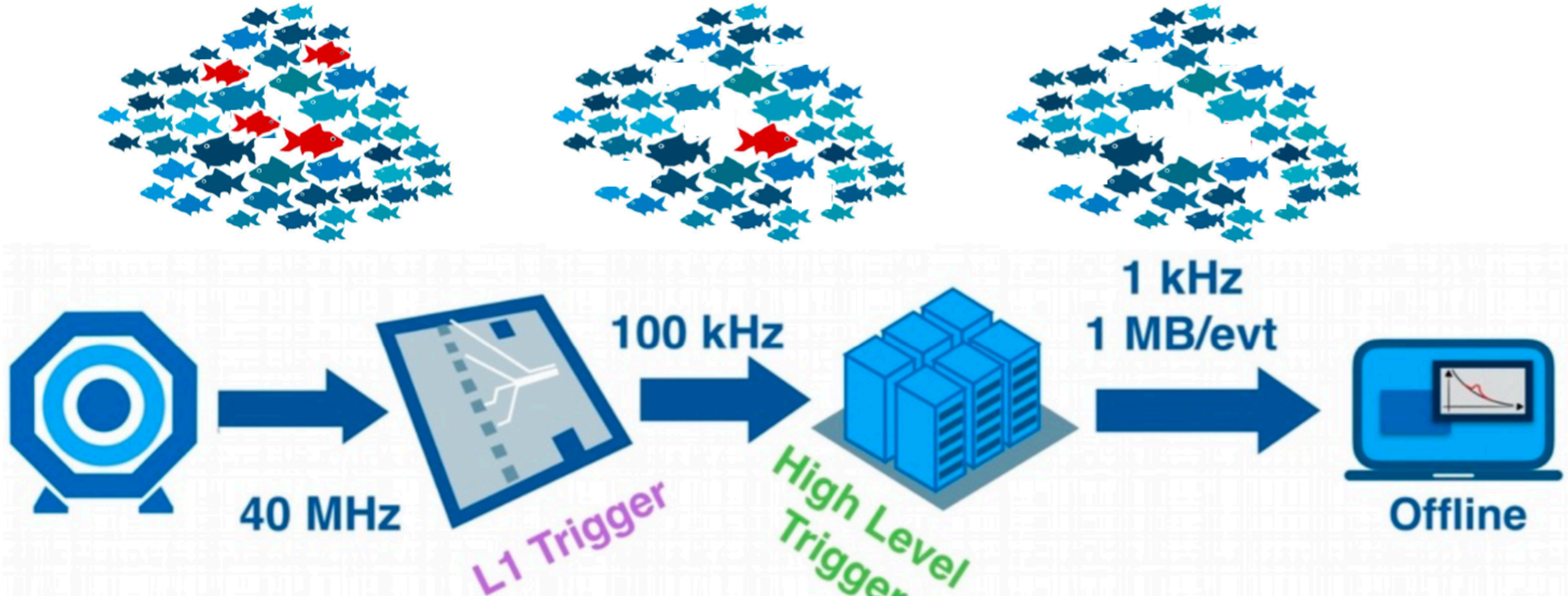






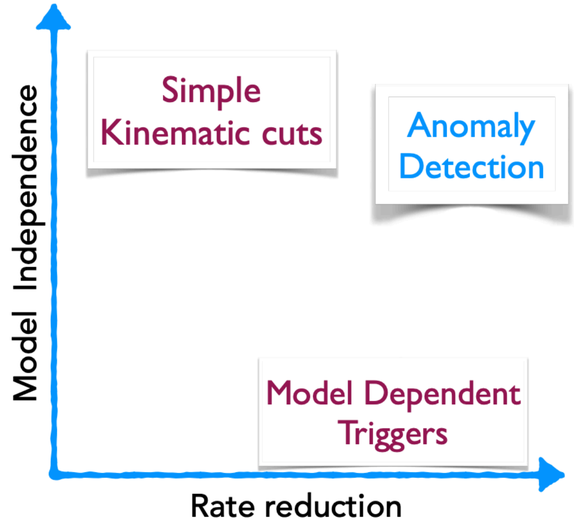
L1 Anomaly Detection Trigger

- What if we don’t know exactly what new physics looks like?
  - → anomaly detection (AD)
- Can reduce network size by removing decoder, using latent space directly



L1 Anomaly Detection Trigger

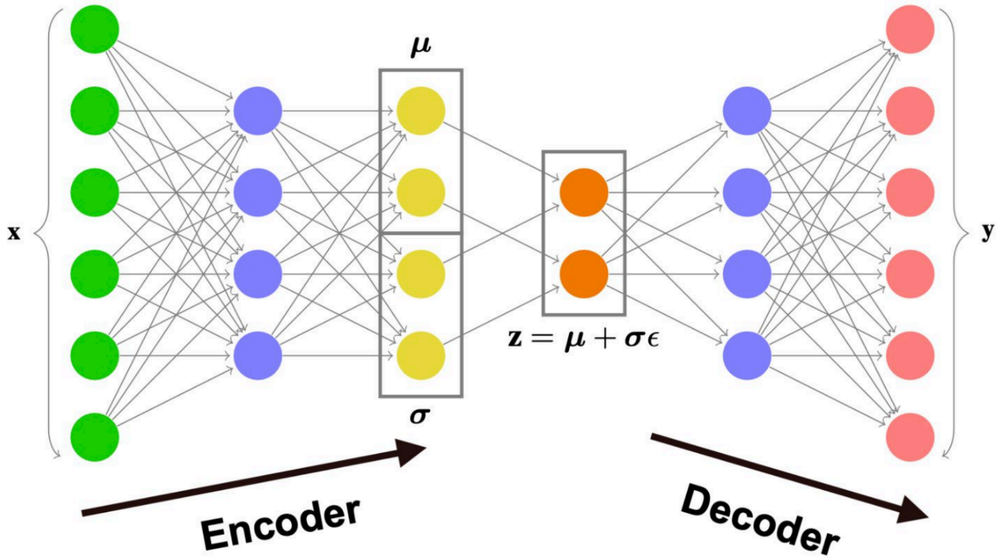
- What if we don't know exactly what new physics looks like?
  - → anomaly detection (AD)
- Can reduce network size by removing decoder, using latent space directly

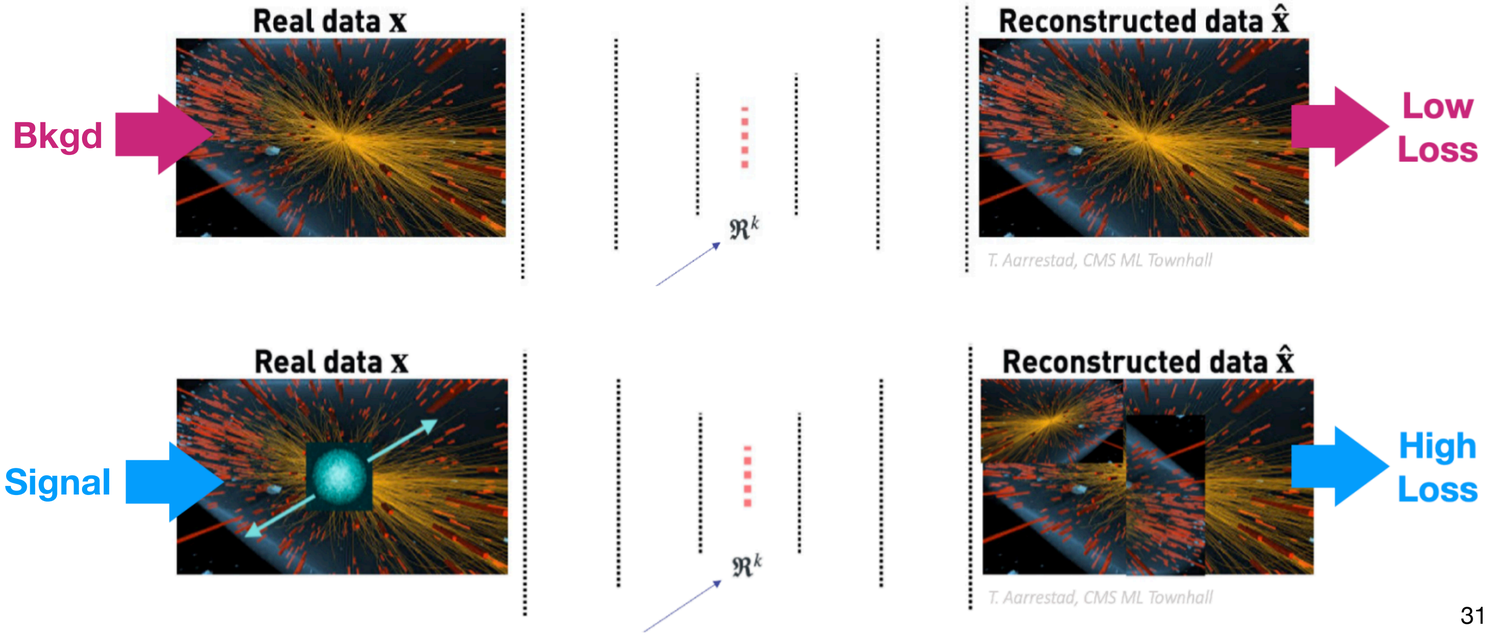


Train on ZeroBias LHC data

Bottleneck: autoencoder learns to compress high dimensional inputs into low dimensional latent space

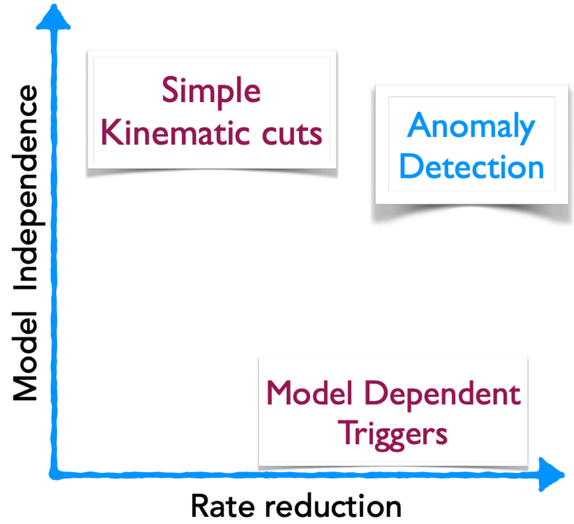
$x - \hat{x}$ represents degree of abnormality





L1 Anomaly Detection Trigger

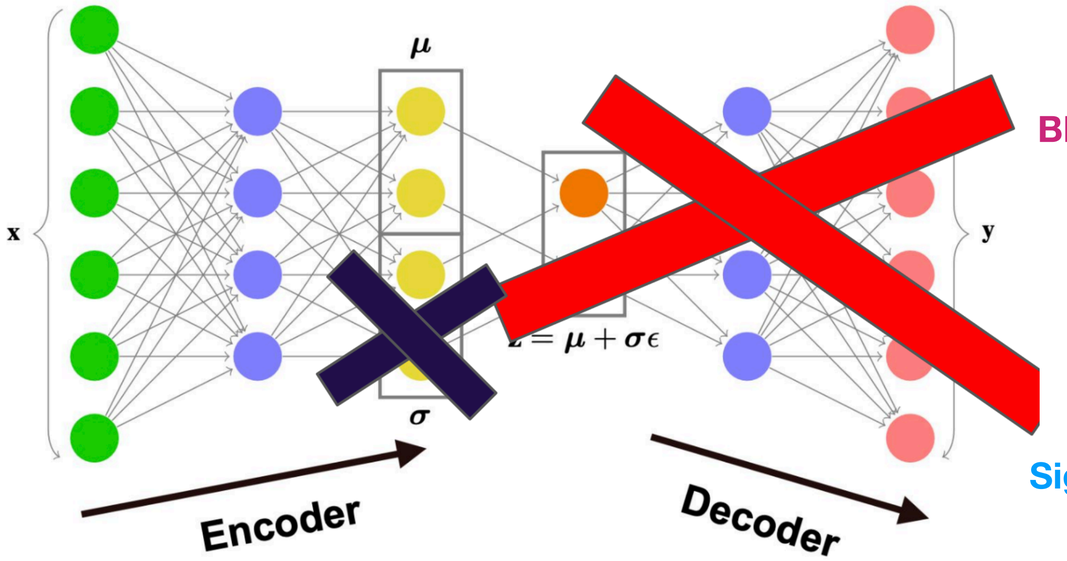
- What if we don't know exactly what new physics looks like?
  - → anomaly detection (AD)
- Can reduce network size by removing decoder, using latent space directly (allows to achieve <50 ns latency)

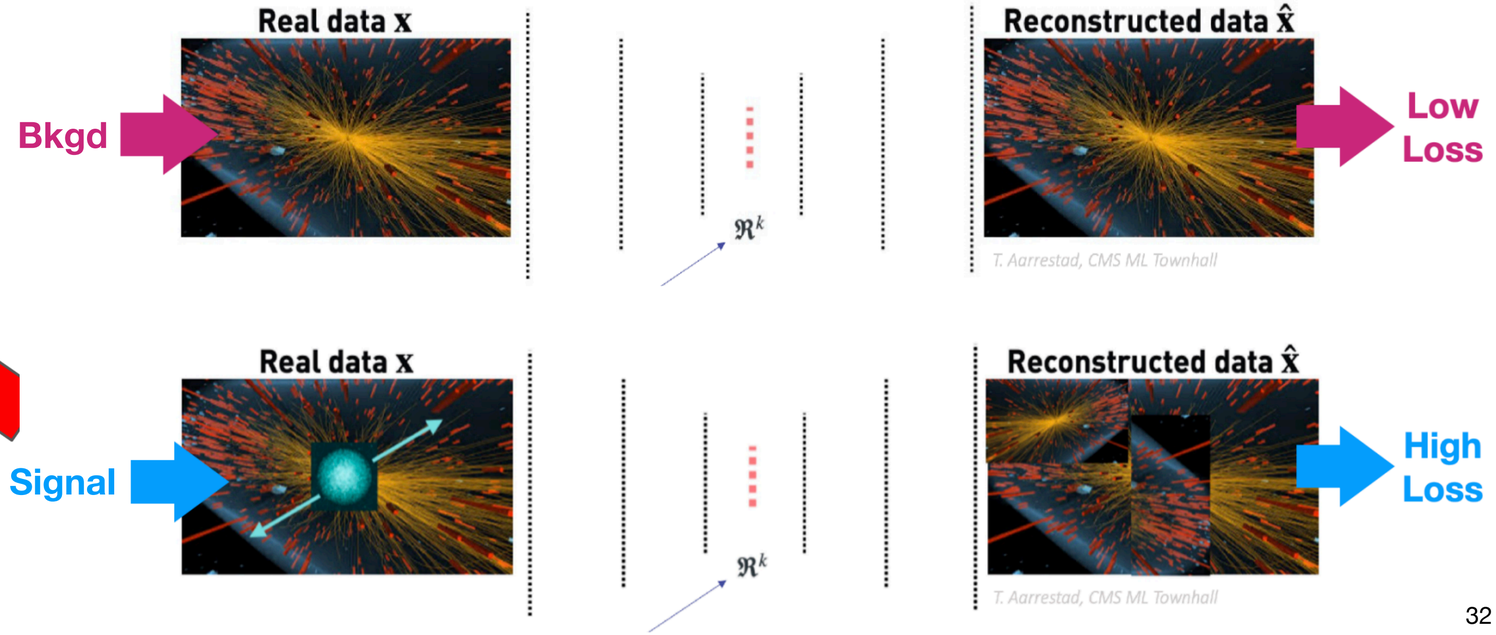


Train on ZeroBias LHC data

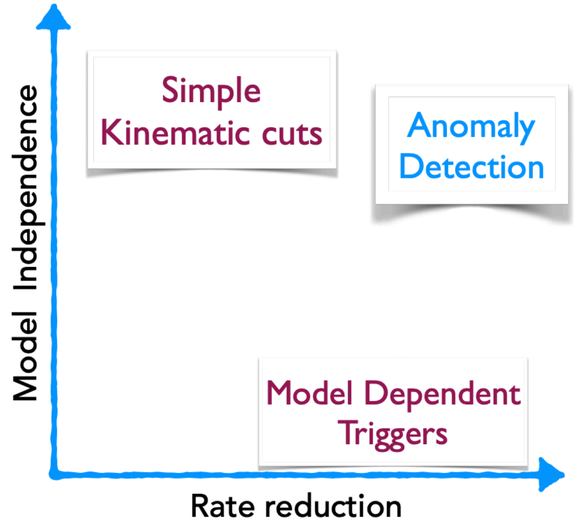
Bottleneck: autoencoder learns to compress high dimensional inputs into low dimensional latent space

$x - \hat{x}$ represents degree of abnormality

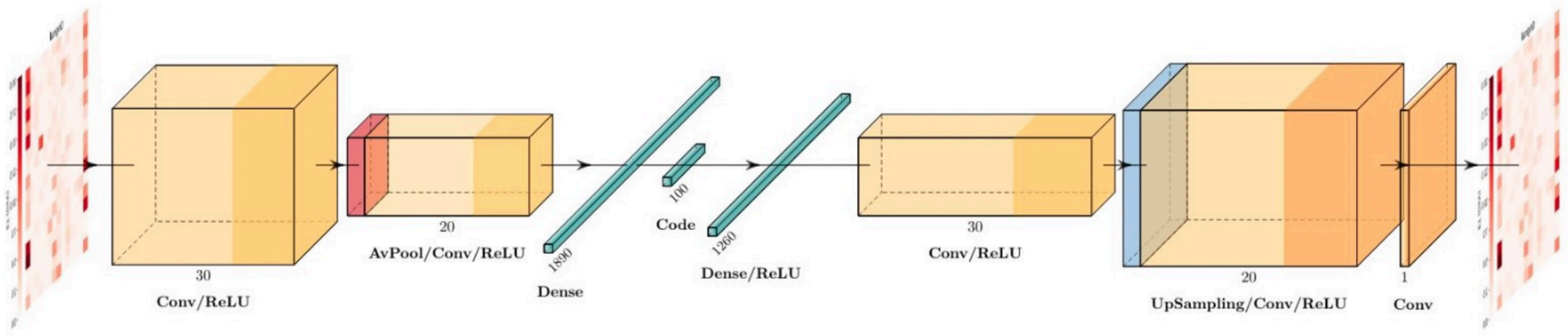




L1 Anomaly Detection Trigger



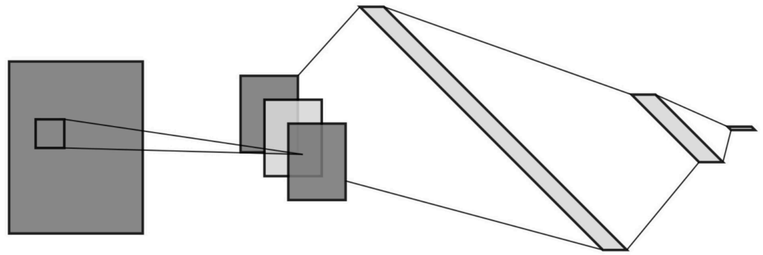
- What if we don't know exactly what new physics looks like?
  - → anomaly detection (AD)
- Can reduce network size by training student network to predict teacher network MSE



$$\mathscr{L} = \|x - x^{\text{teacher}}_{\text{pred}}\|^2$$

Teacher network

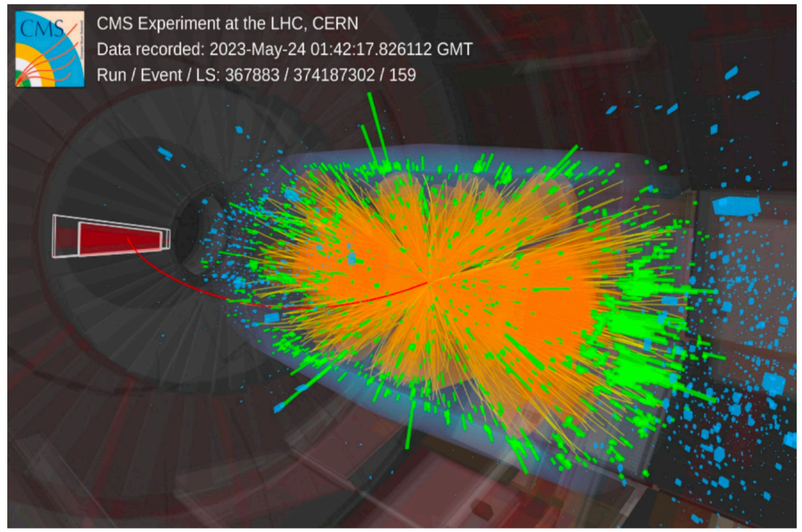
Student network

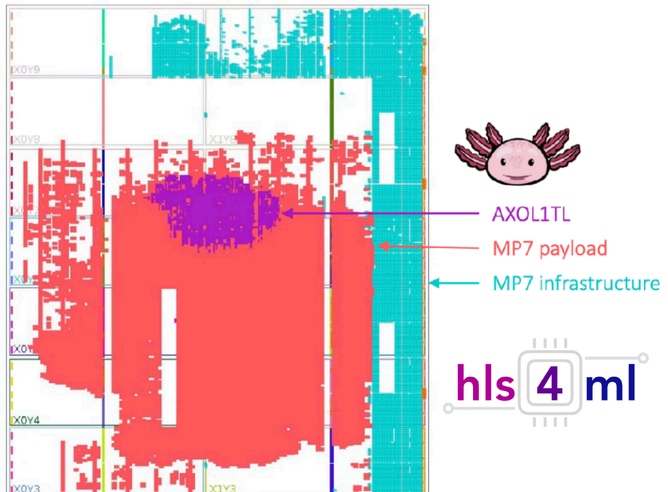


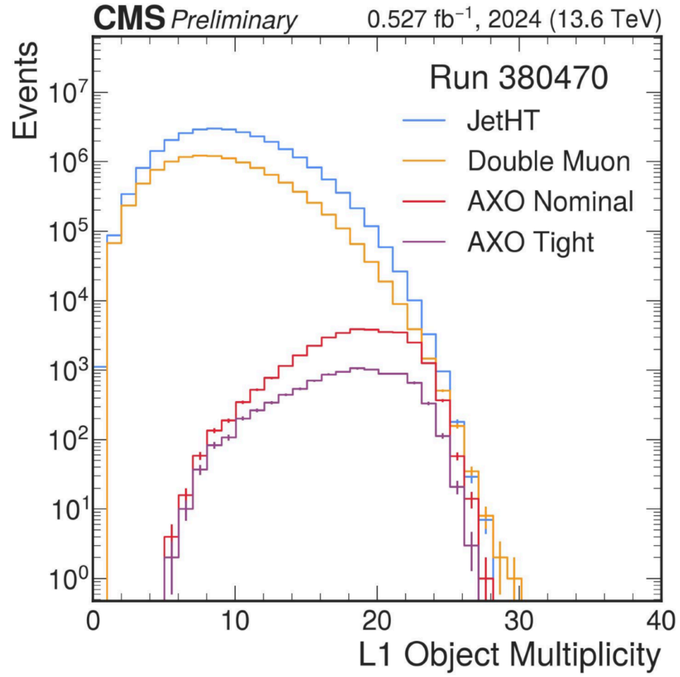
$$\mathscr{L} = \|(\|x - x^{\text{teacher}}_{\text{pred}}\|^2) - x^{\text{student}}_{\text{pred}}\|^2$$

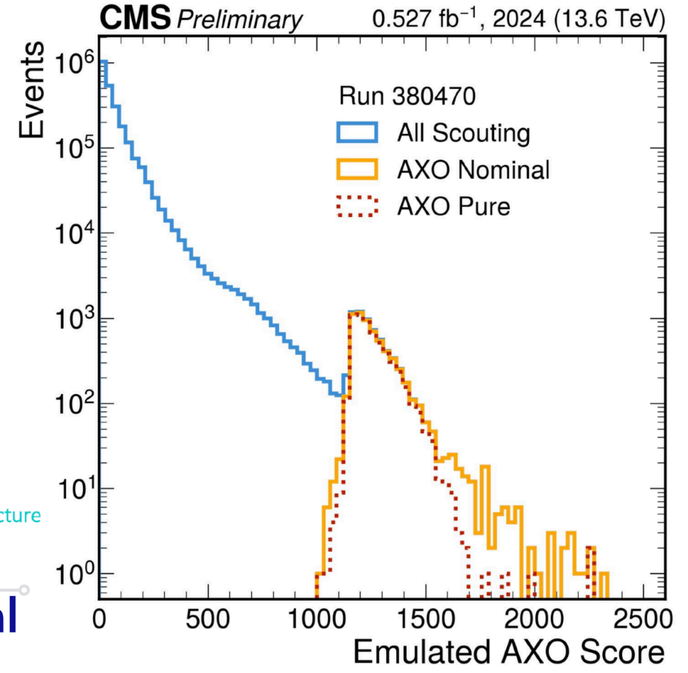
L1 AD Trigger

- 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
  - Network preferentially identifies large multiplicity events
  - Potentially large gains in new physics acceptance



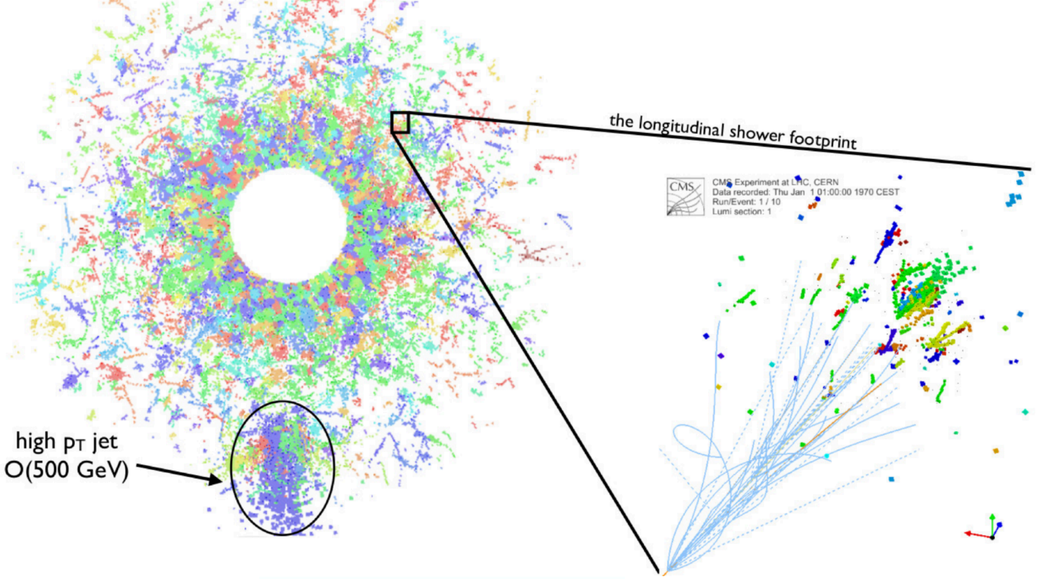


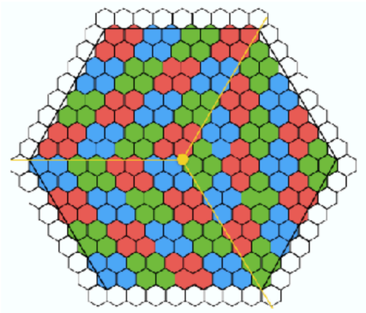


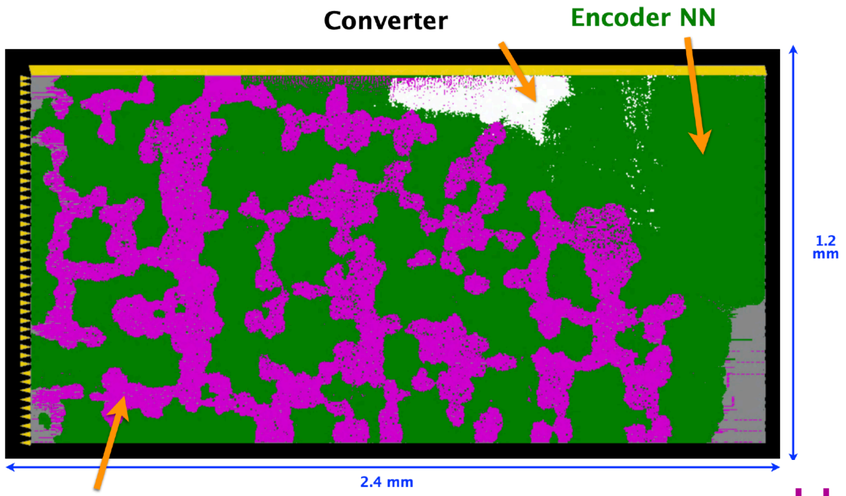


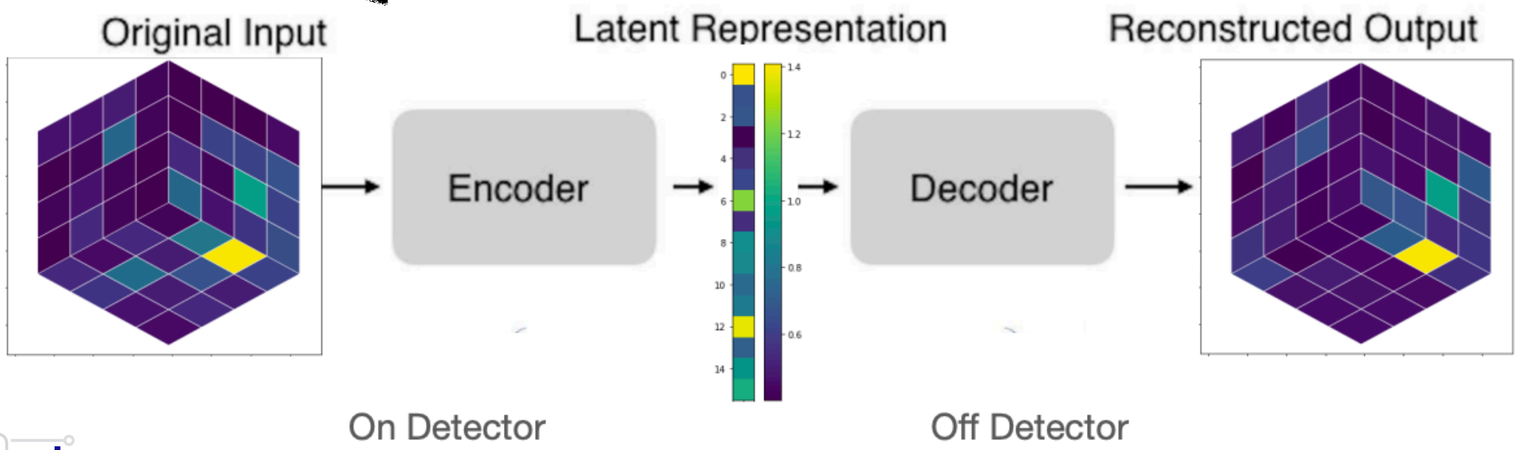
Data Compression

- What if there's simply too much data to get off the detector in the first place?!
  - CMS High Granularity Calorimeter will have 6.5 million readout channels, 50 layers → need some compression
- AEs well-suited (only transmit latent space)
- Model must be run in high radiation environment (ECON-T ASIC, logic triplicated) [2105.01683]





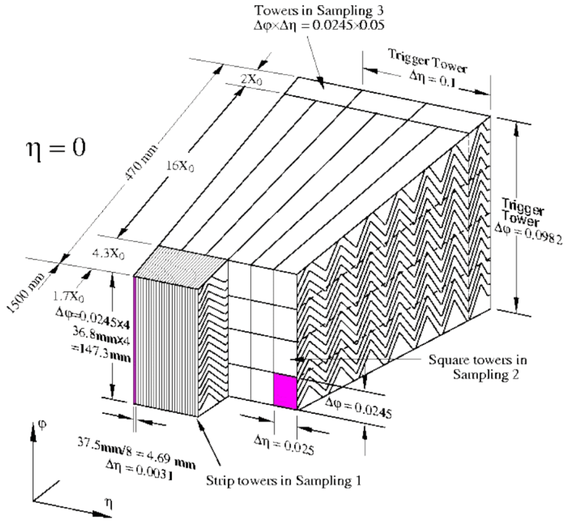


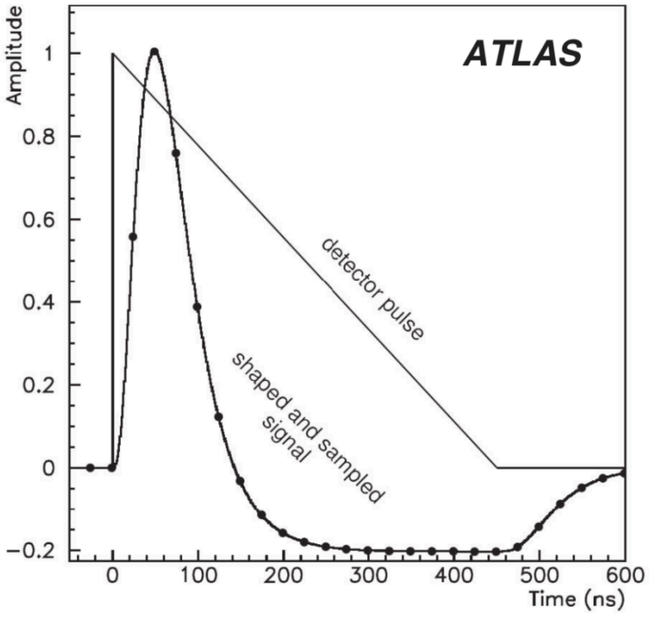


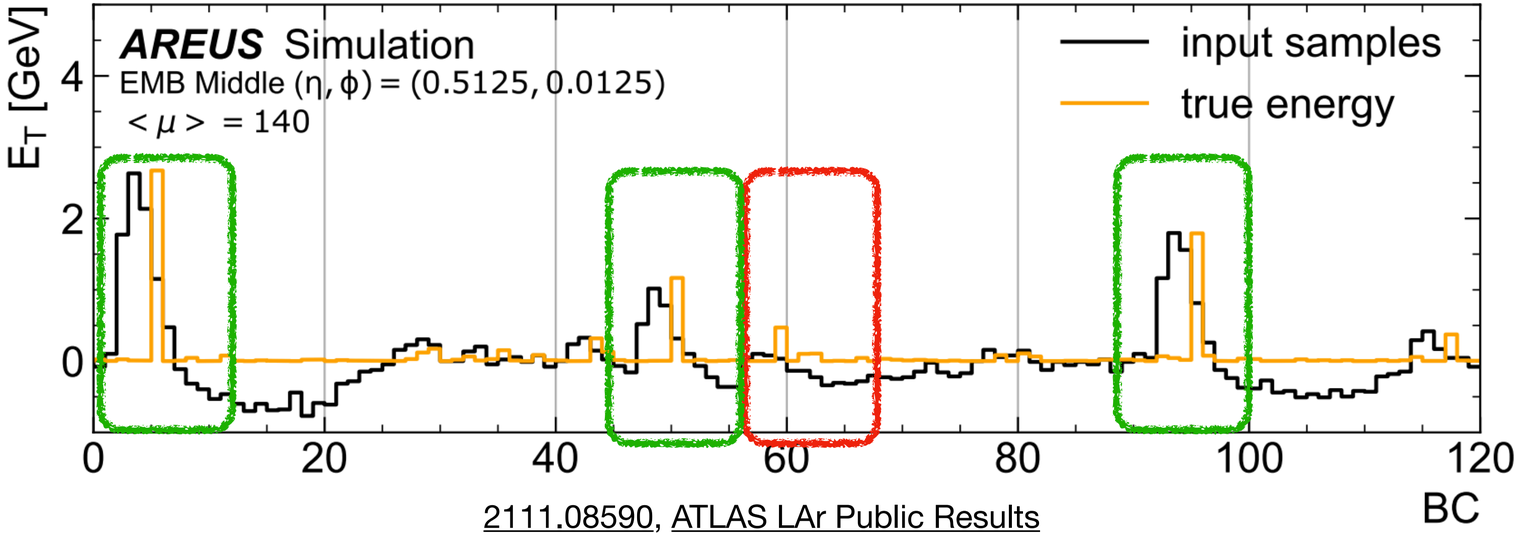
hls 4 ml

LAr Peak Finding

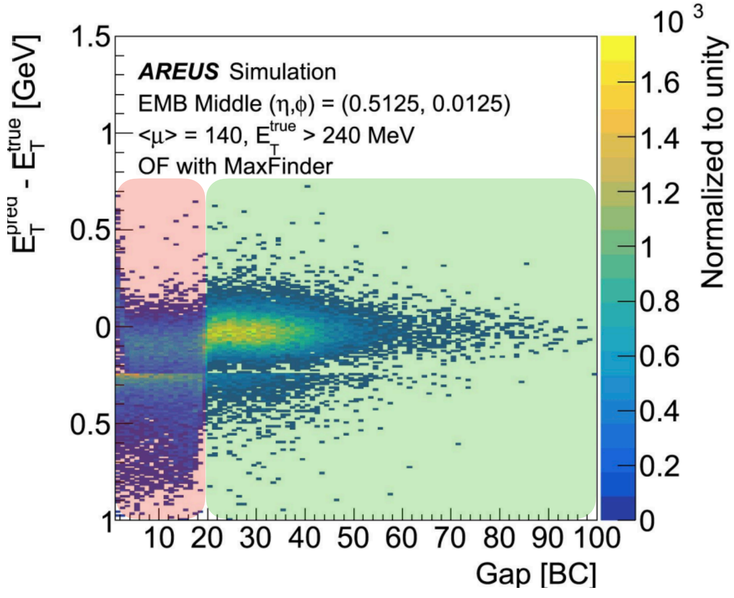
- ATLAS LAr calorimeter needs to measure time and energy of pulses
  - Overlapping pulses difficult for simple, fast algorithms to handle (150 ns = 6 BXs)
- CNN and LSTM architectures both able to significantly improve performance
  - Well-suited for data structure, able to account for non-linear correlations





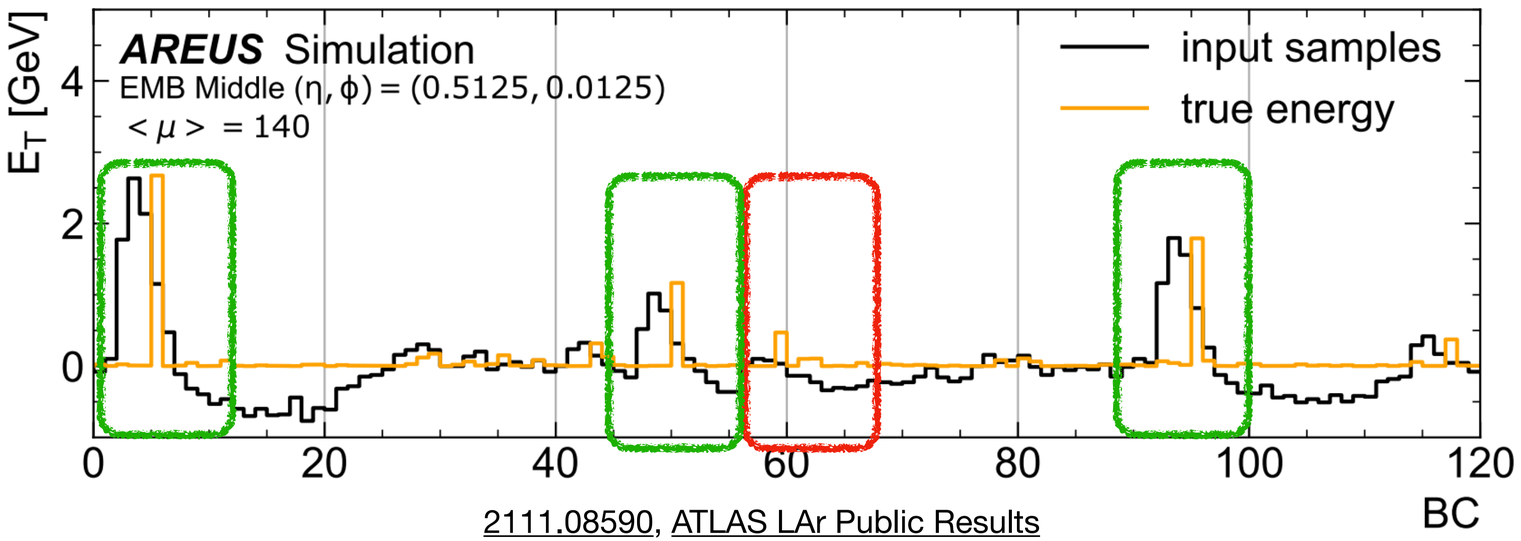


2111.08590, ATLAS LAr Public Results

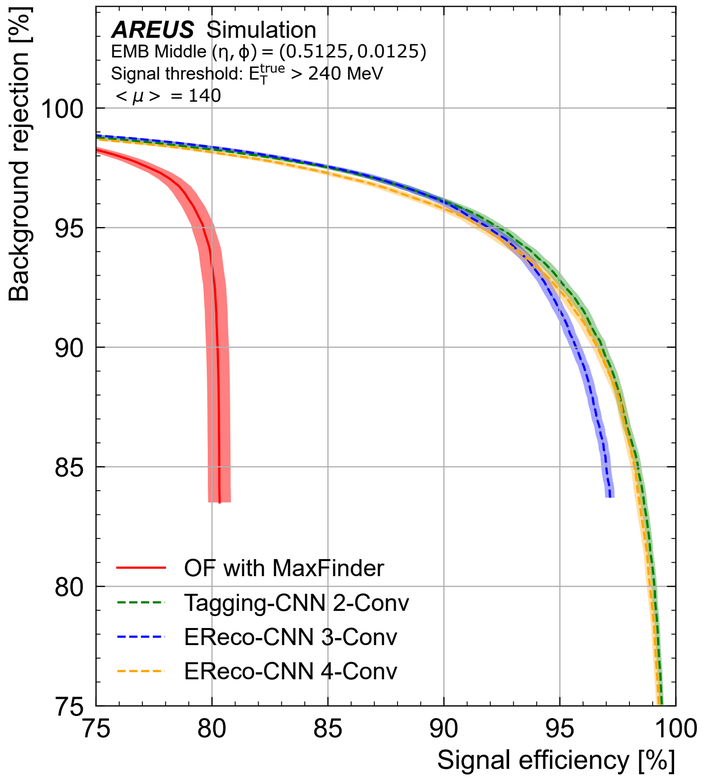


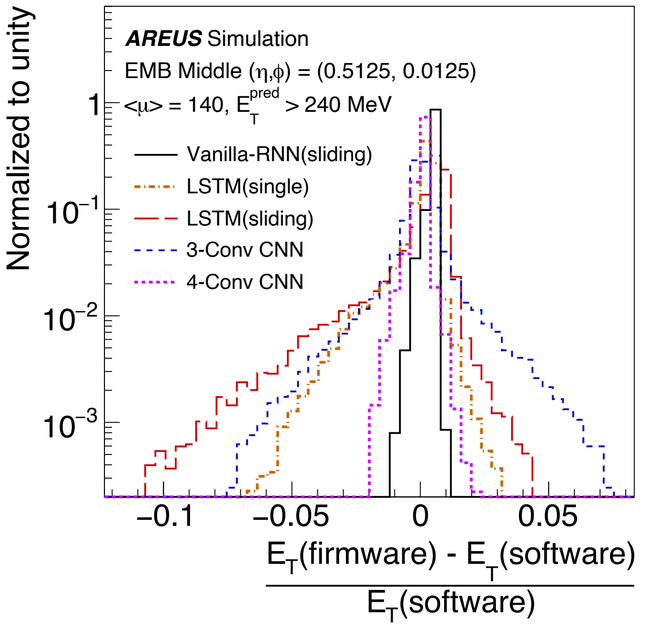
LAr Peak Finding

- 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

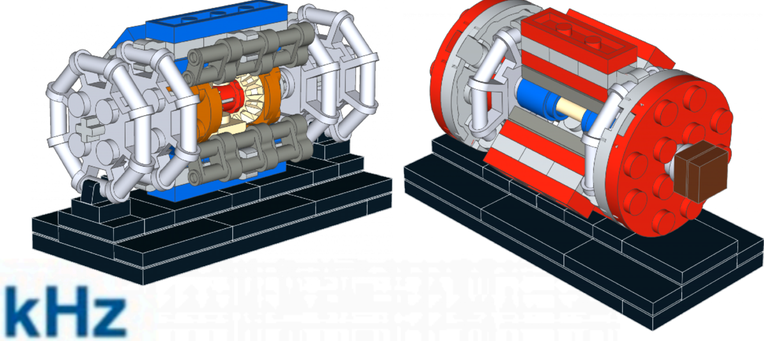


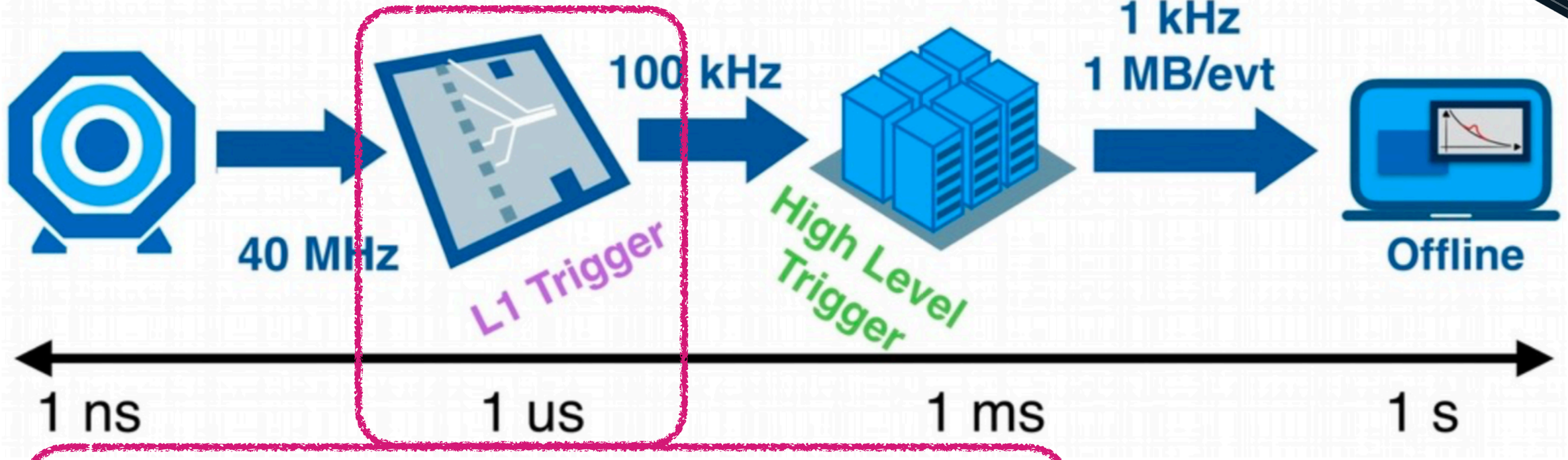
2111.08590, ATLAS LAr Public Results





LHC Data Processing / Readout

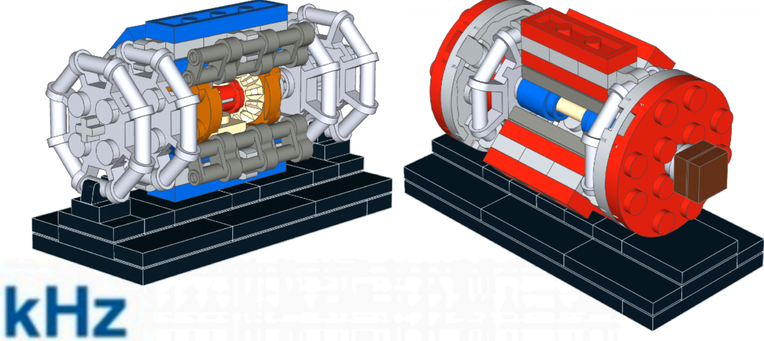


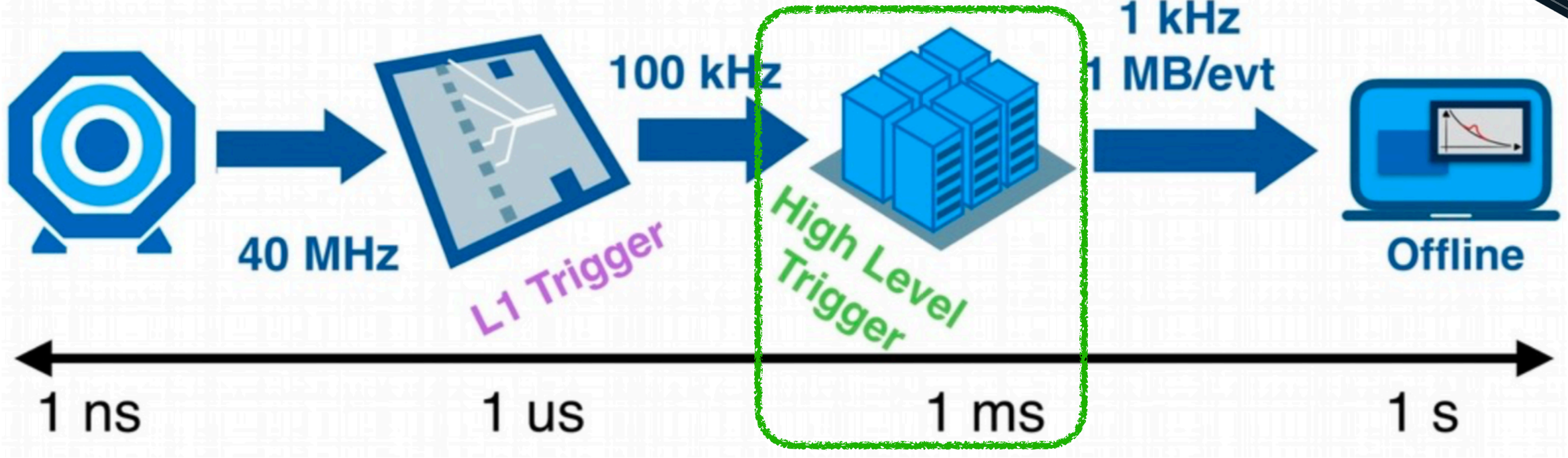


- **Level-1 Trigger** - O(μs) latency
- **High Level Trigger** - O(100 ms) latency
- **Offline** → 1 s latencies

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

LHC Data Processing / Readout



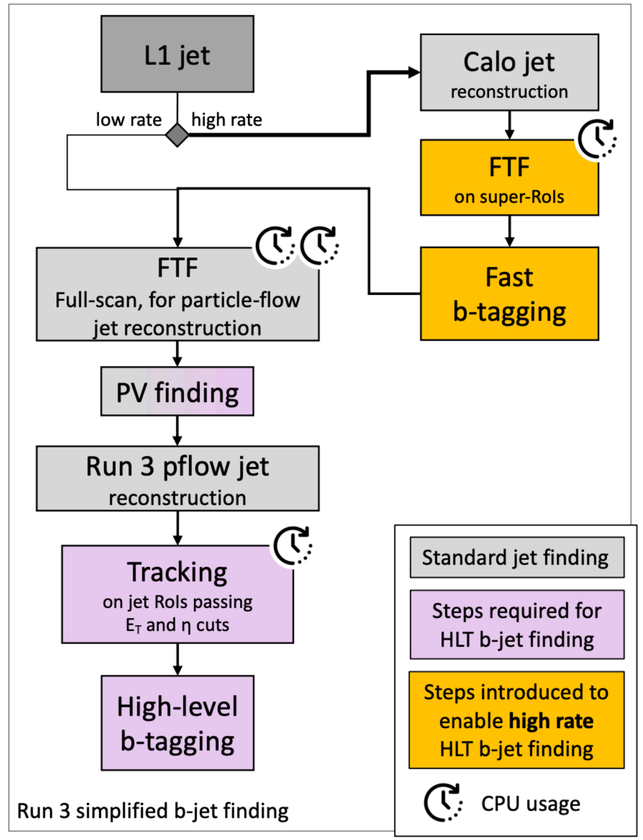


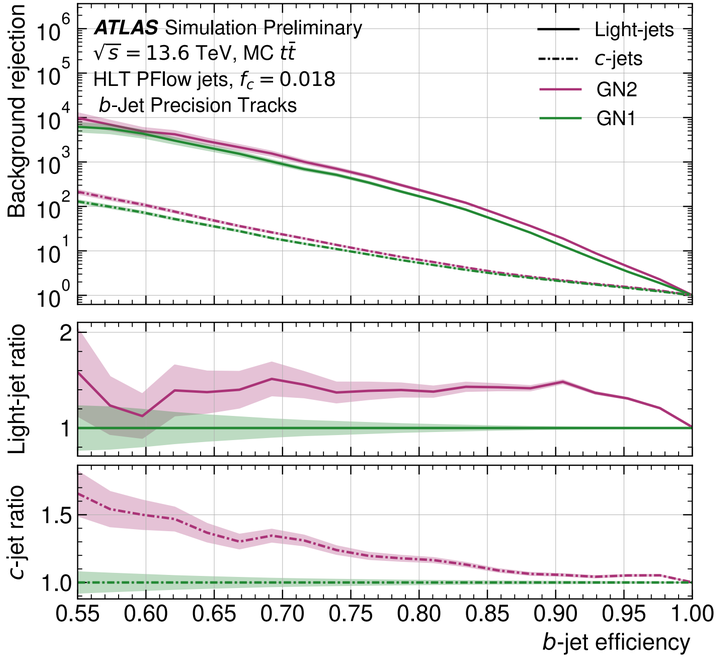
- **Level-1 Trigger**
- **High Level Trigger**
- **Offline**

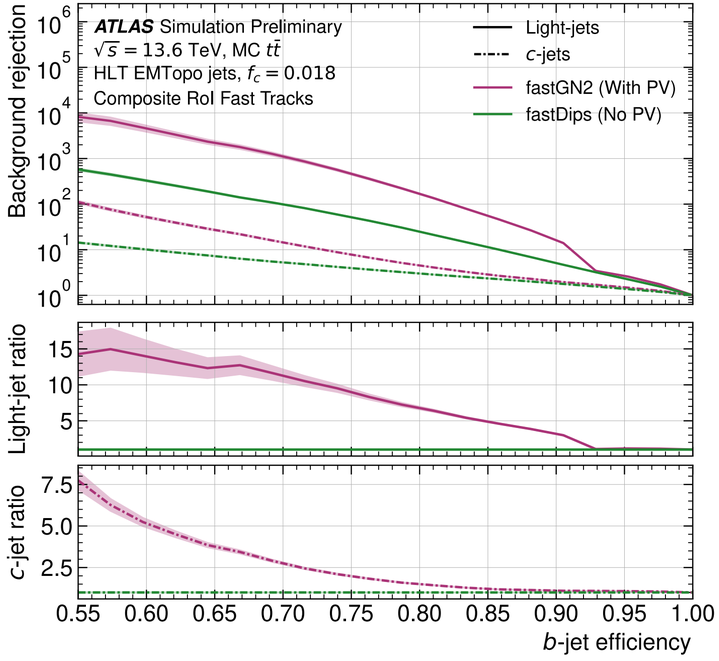
If we don't interesting identify events in trigger we lose them forever!

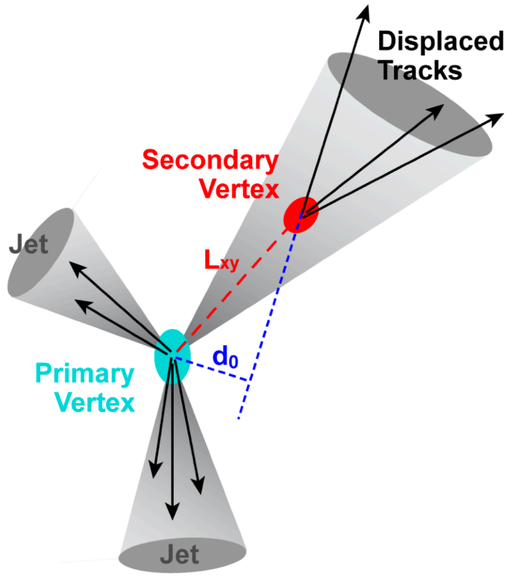
HLT b-tagging

- Early usage of ML at LHC for b-tagging
  - High complexity, physics motivation → significant ML gains
- Algorithms have to evolve quickly to keep up with modern ML
  - BDTs → MLPs & DeepSets → GNNs (+ attention)
- Tiered reconstruction/filtering allows running computationally intensive algorithms in trigger



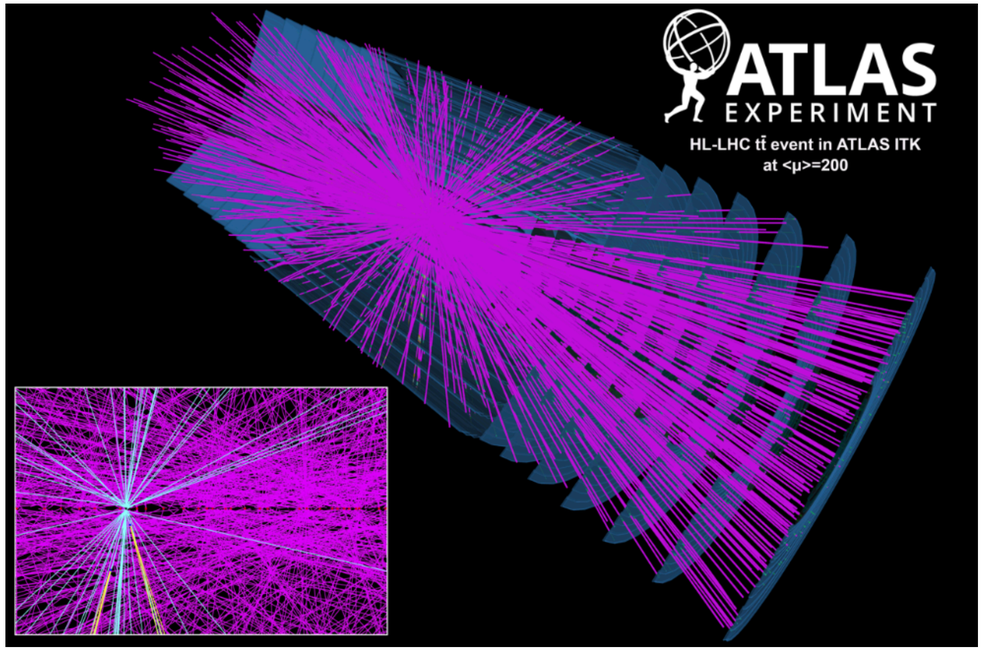


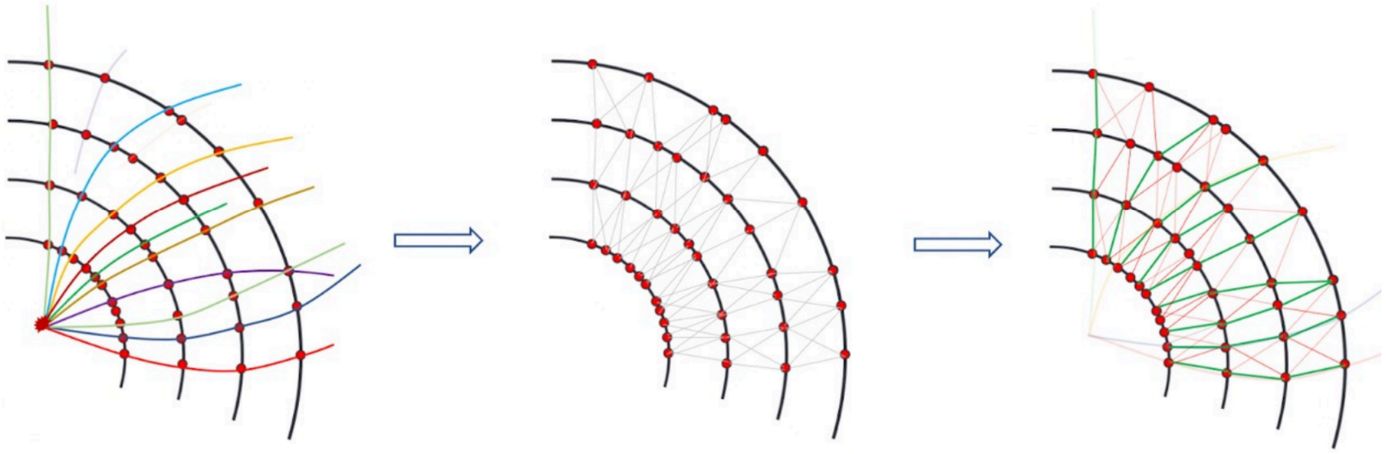




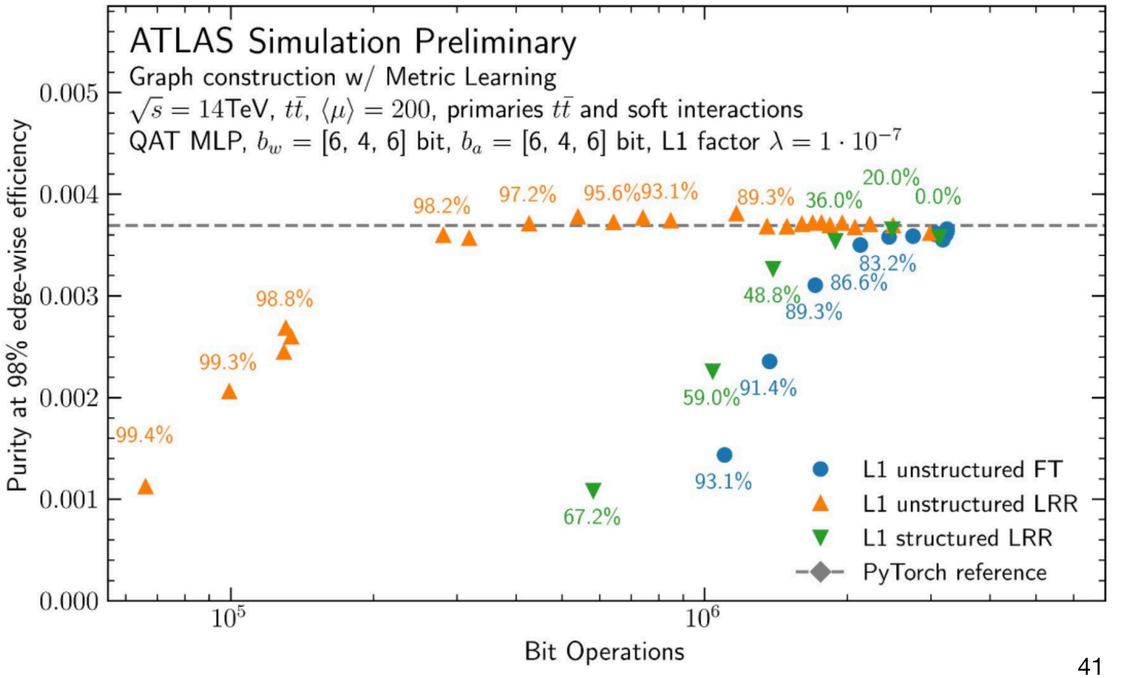
GNN Tracking

- Tracking is an incredibly hard problem, tracking in HLT even harder
  - Huge combinatorics, only going to get worse
- GNNs show promise for HL-LHC
- FLOPs/power important considerations
  - Pruning one potential option for reduction

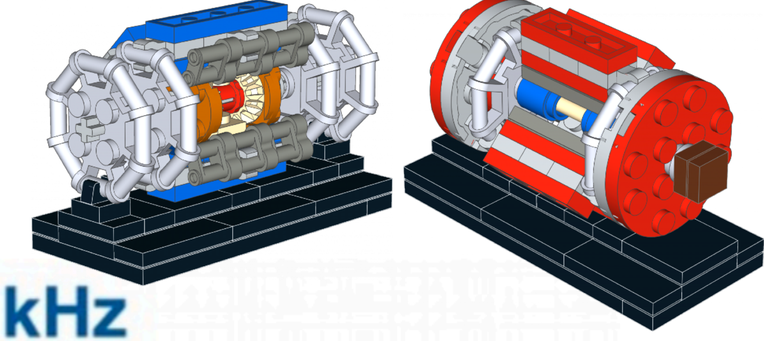


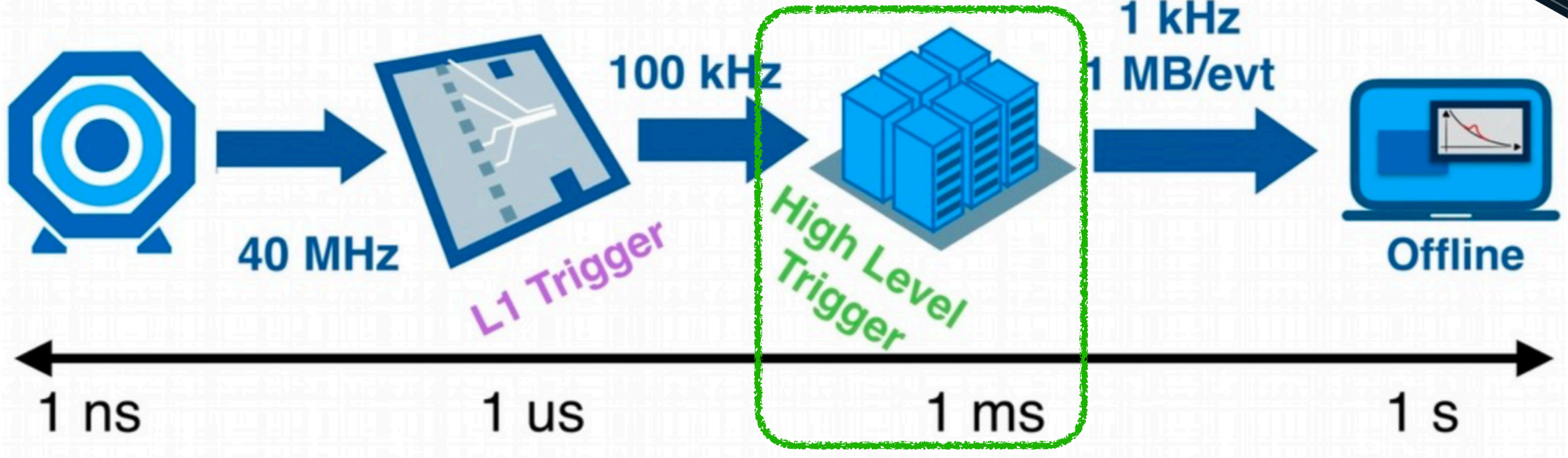


ATL-COM-DAQ-2024-004



LHC Data Processing / Readout

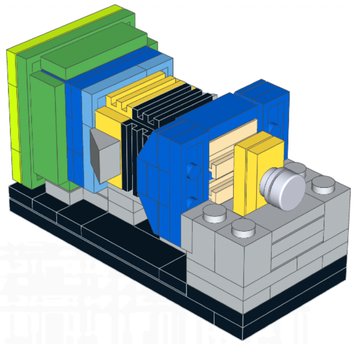


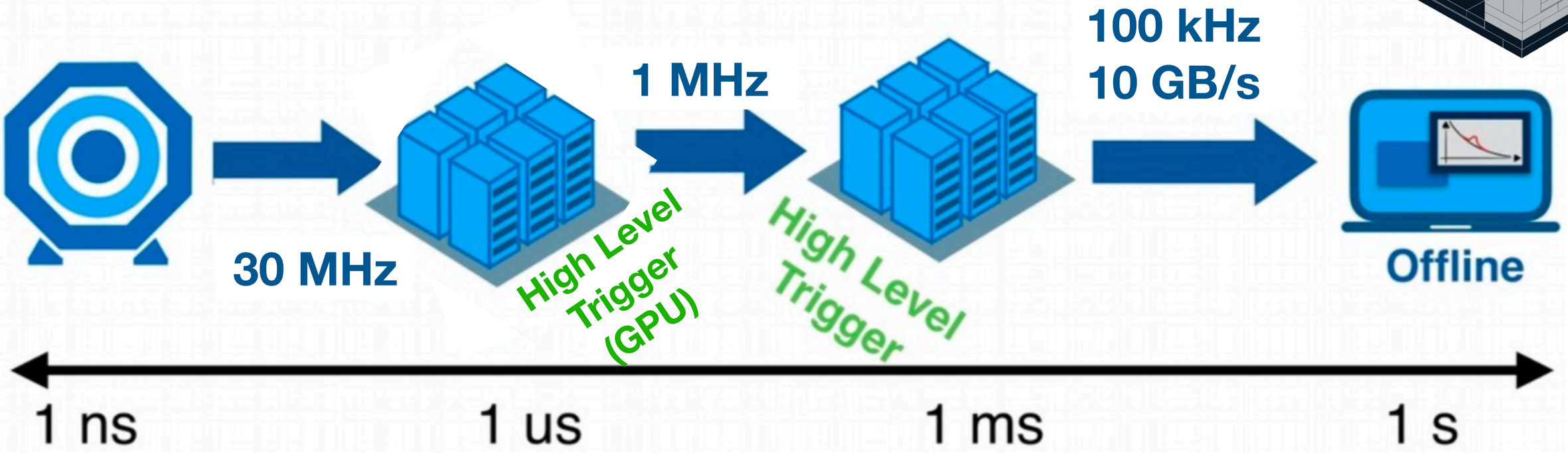


- Level-1 Trigger
- High Level Trigger
- Offline

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

LHC Data Processing / Readout



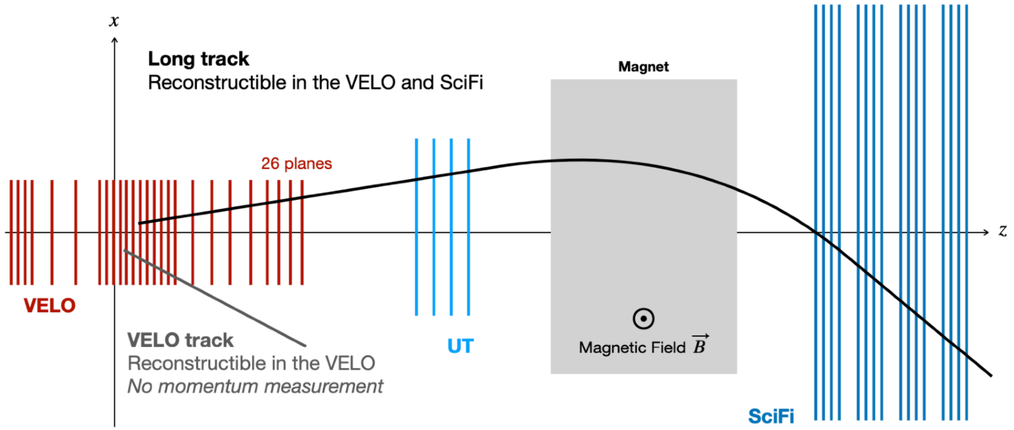


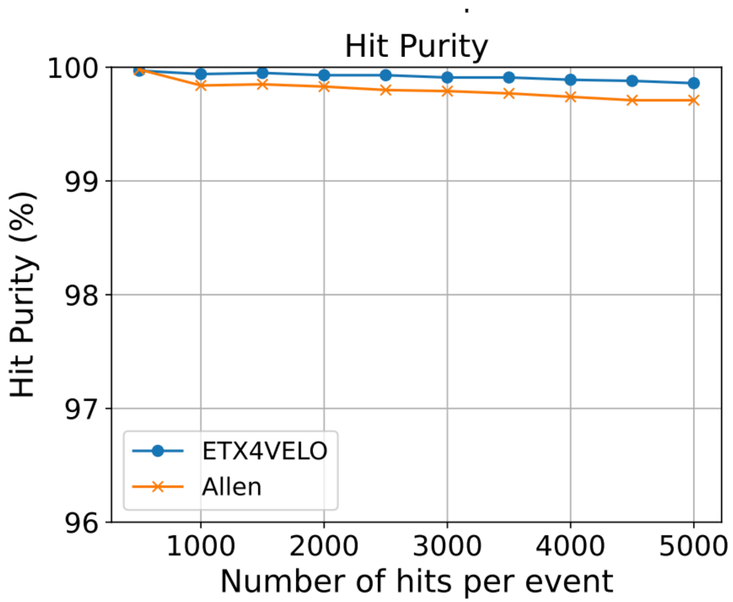
- High Level Trigger (GPU)
- High Level Trigger (CPU)
- Offline

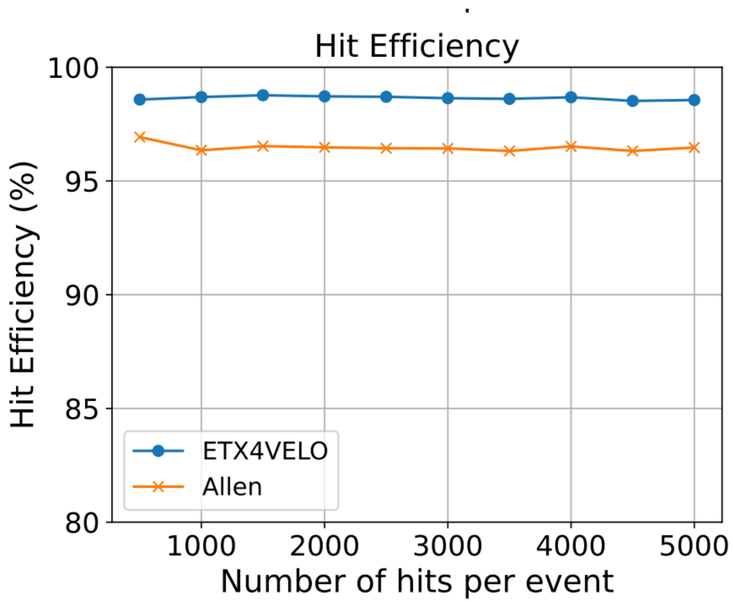
If we don't interesting identify events in trigger we lose them forever!

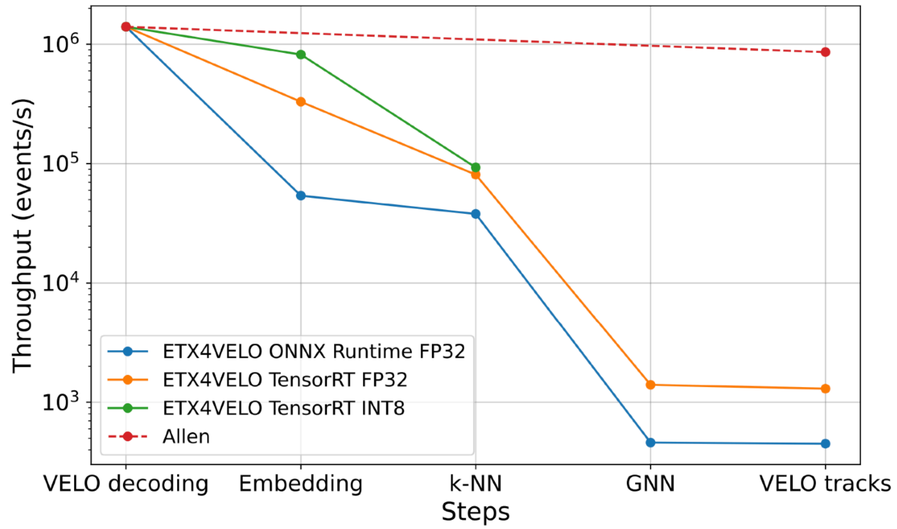
GNN Tracking (LHCb)

- LHCb must do tracking at 30 MHz
  - Exploring use of **GNNs** [2407.12119]
- Demonstrated good performance possible
- Achieving necessary throughput is a challenging problem



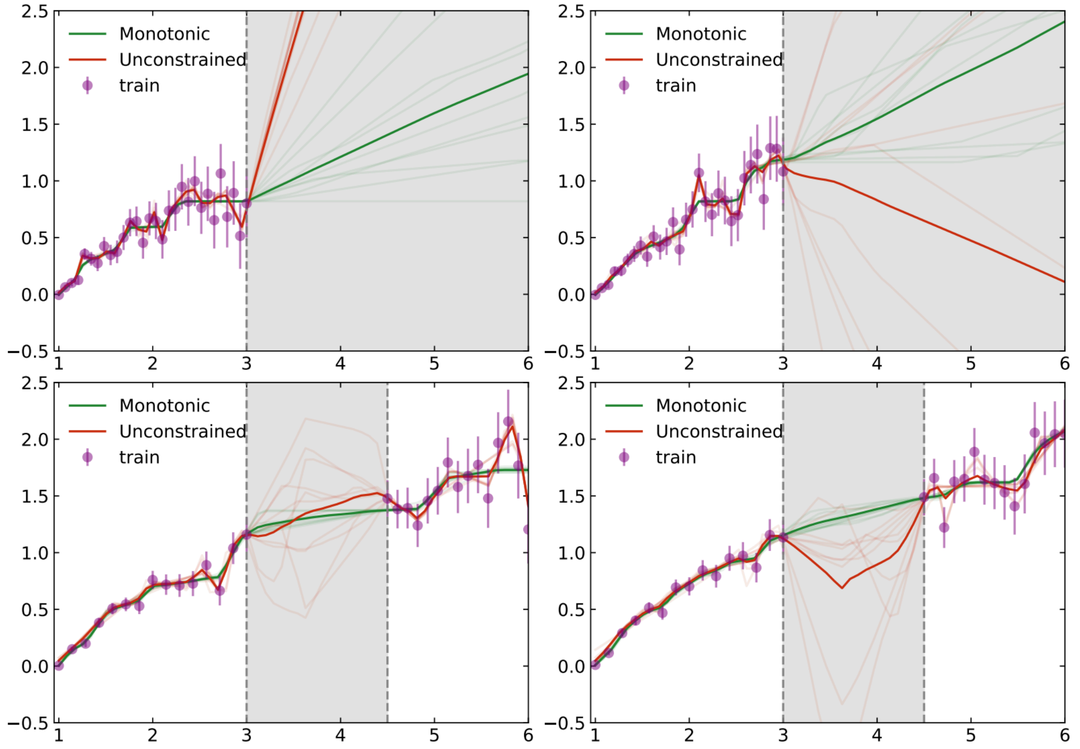


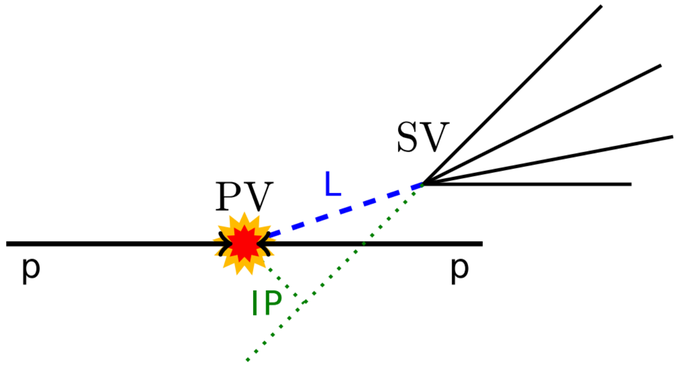


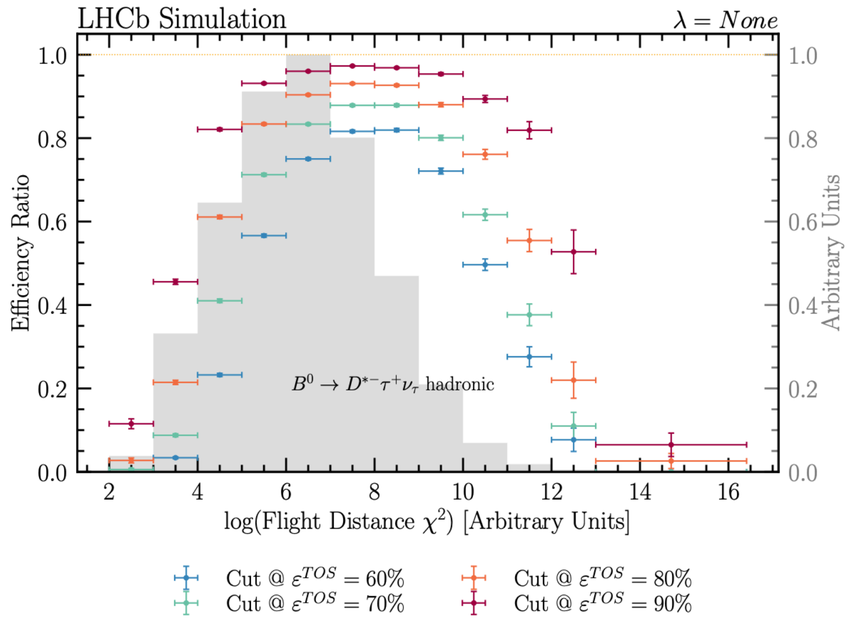


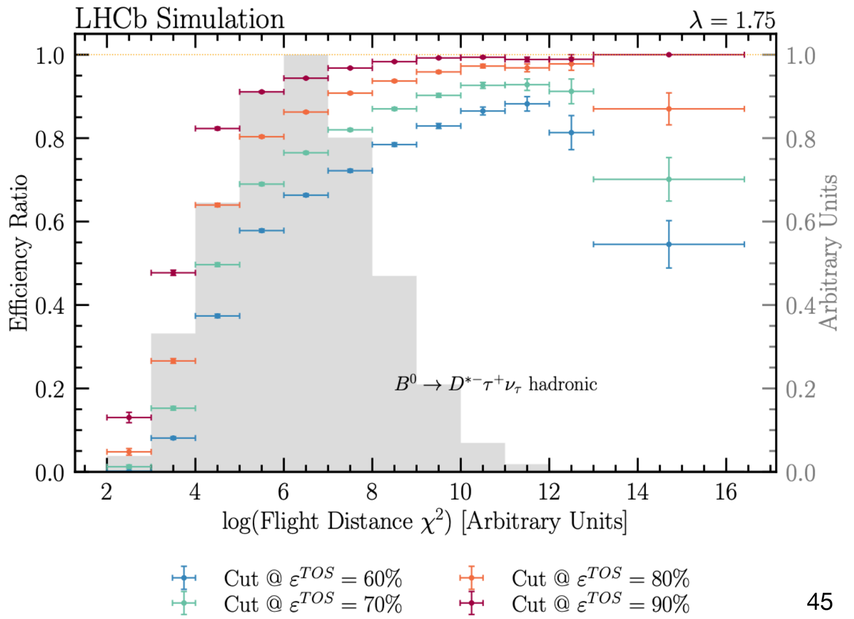
Lipschitz Monotonic NN

- On-detector ML is not just about speed
  - Robustness and understandability are also very important
- Networks can be made provably monotonic [2112.00038]
- LHCb has used this technique to design NNs for use in HLT
  - Eg. smooth dependence on flight distance for heavy flavor decays
- Improved stability





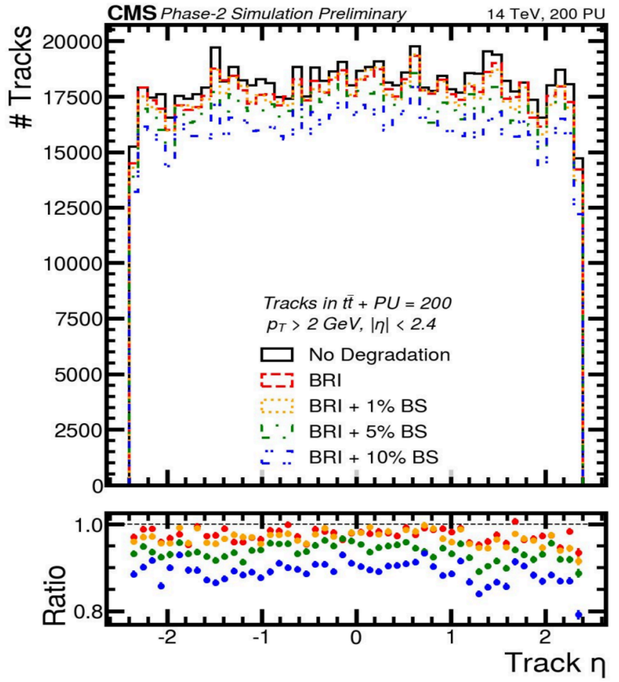




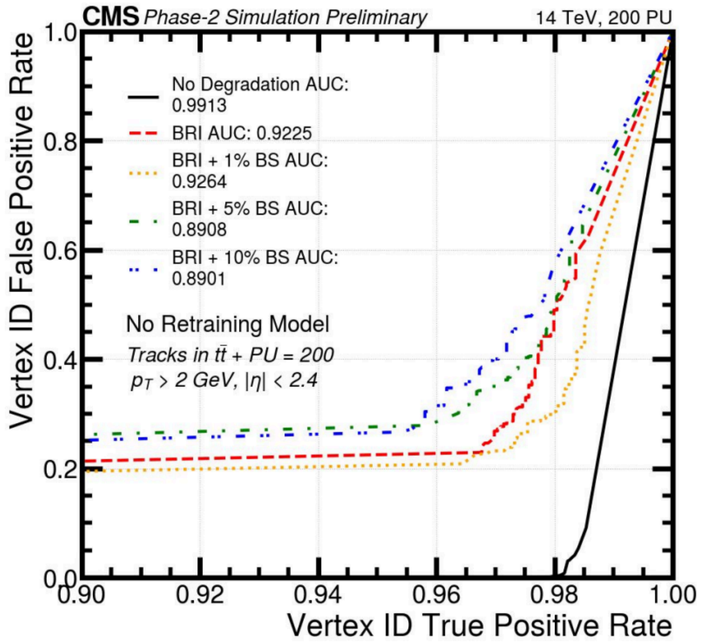
2306.09873 , 2312.14265

Continual Learning

- On-detector ML has no re-do button
  - Cannot just reprocess with new network if conditions change
- Continual learning method uses mix of original and new data to retrain model
  - Better performance than simple retraining (or no retraining)
- Important consideration especially when conditions can change significantly
- Example from CMS considers degradations in L1 tracking



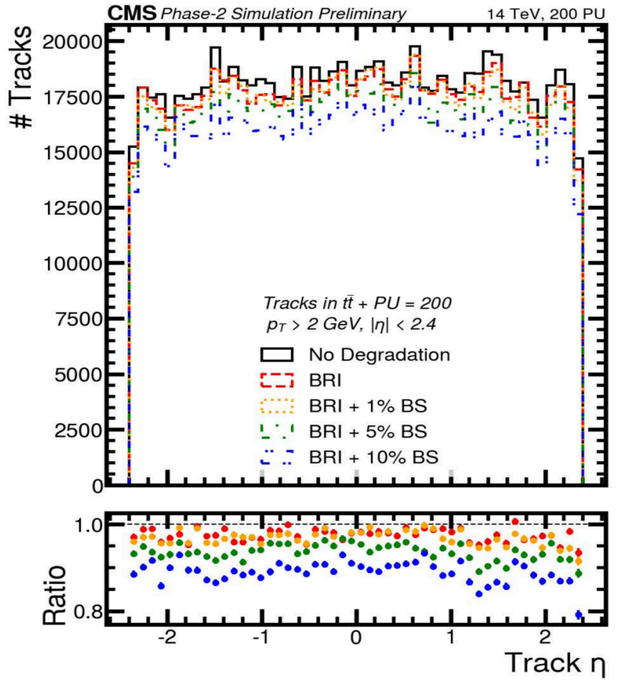
No Retraining Model



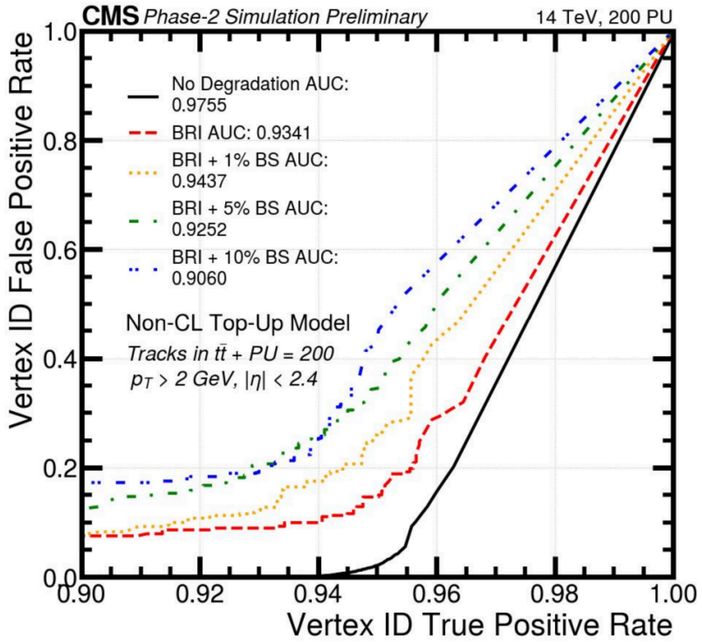
CMS DP-2023/022

Continual Learning

- On-detector ML has no re-do button
  - Cannot just reprocess with new network if conditions change
- Continual learning method uses mix of original and new data to retrain model
  - Better performance than simple retraining (or no retraining)
- Important consideration especially when conditions can change significantly
- Example from CMS considers degradations in L1 tracking

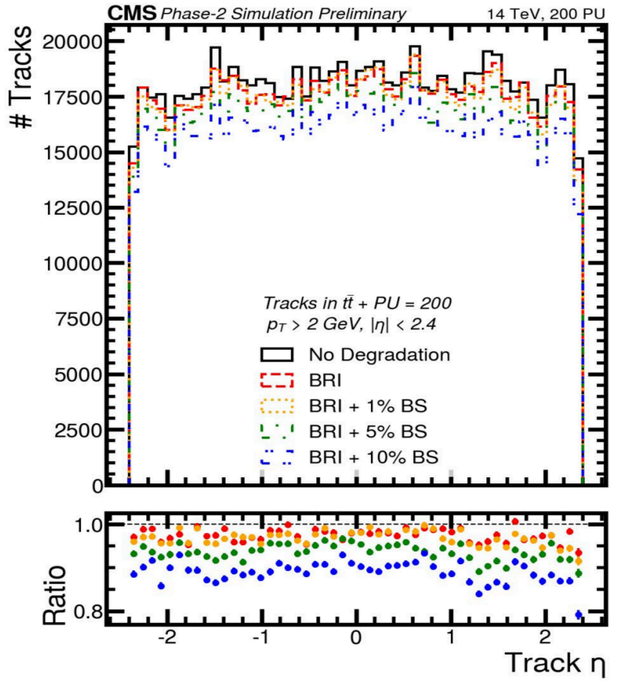


Non-CL Top-Up Model

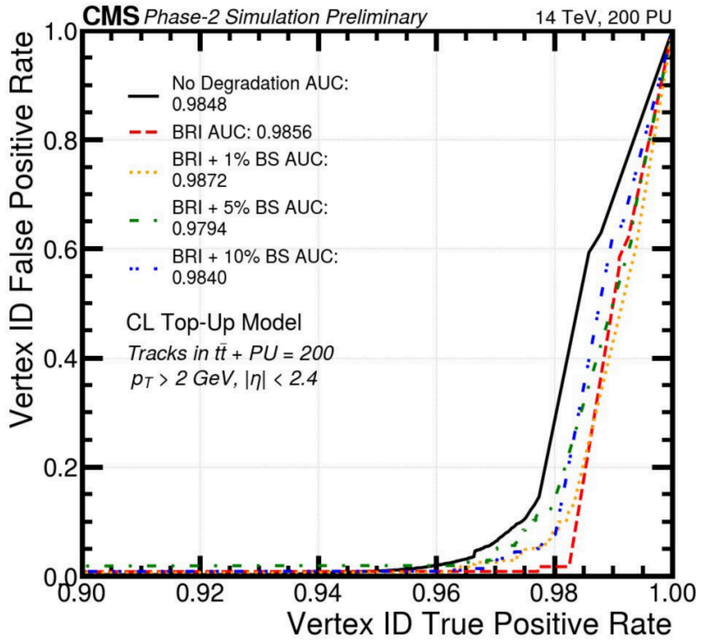


Continual Learning

- On-detector ML has no re-do button
  - Cannot just reprocess with new network if conditions change
- Continual learning method uses mix of original and new data to retrain model
  - Better performance than simple retraining (or no retraining)
- Important consideration especially when conditions can change significantly
- Example from CMS considers degradations in L1 tracking

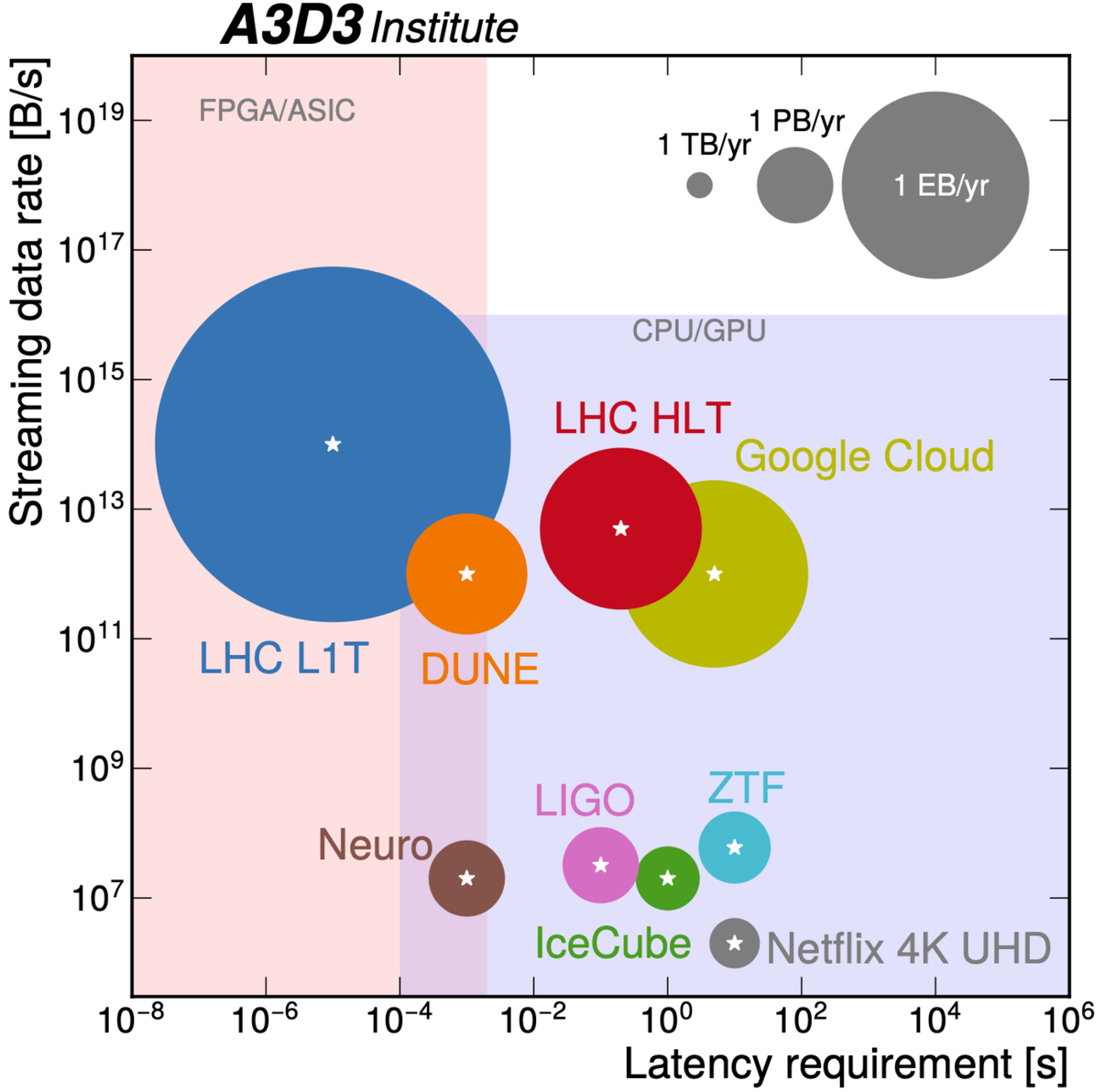


CL Top-Up Model



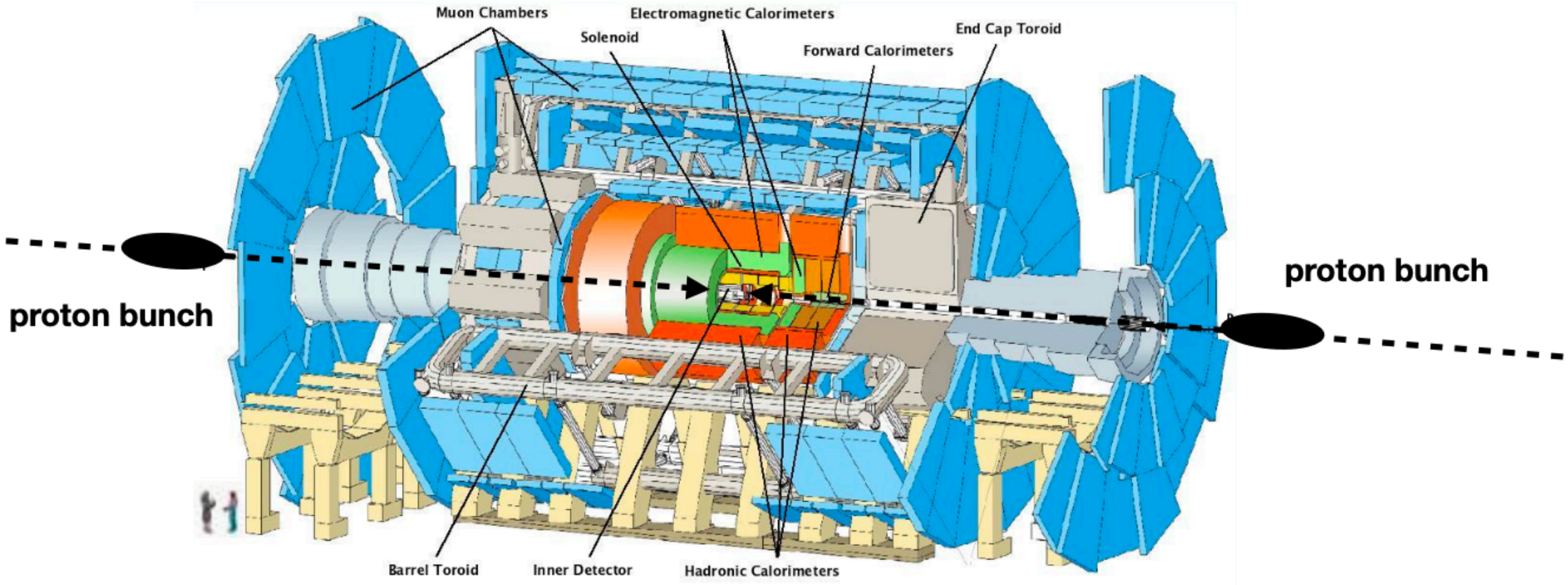
Conclusions

- Increasingly possible and necessary to perform real time edge ML in LHC experiments
  - FPGA and GPUs are main hardware tools but not only ones!
- ML offers improved performance over traditional algorithms
  - With advancing ML off-detector brings better alignment of offline and online algorithms
- Applications in many other fields, areas too!

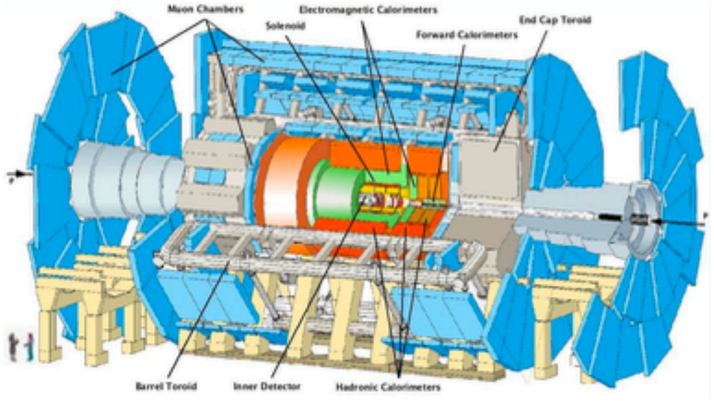


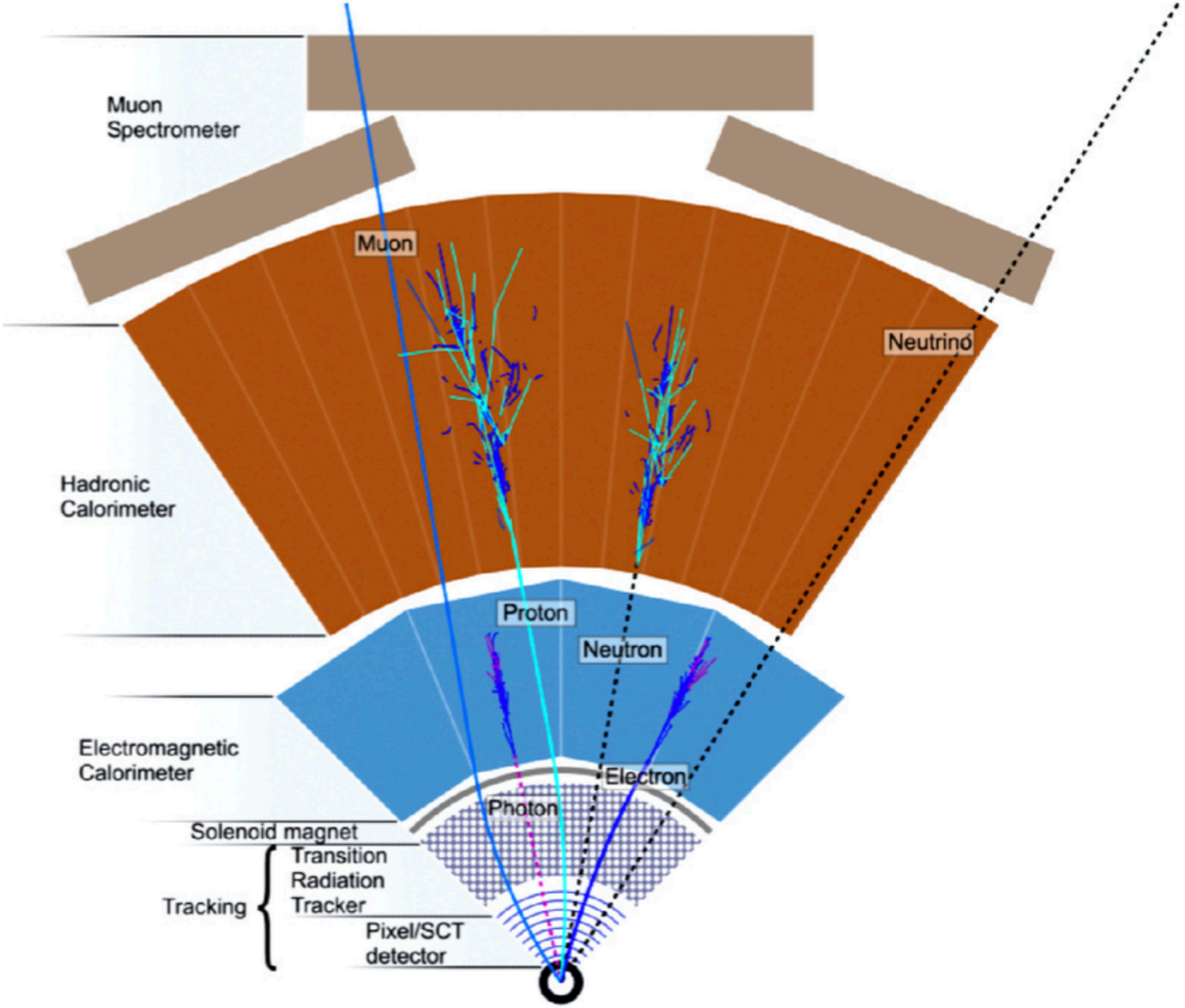
BACKUP

A Toroidal LHC ApparatuS (ATLAS)

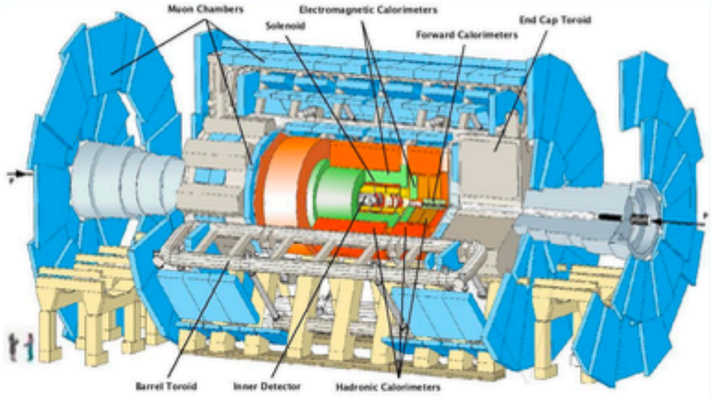


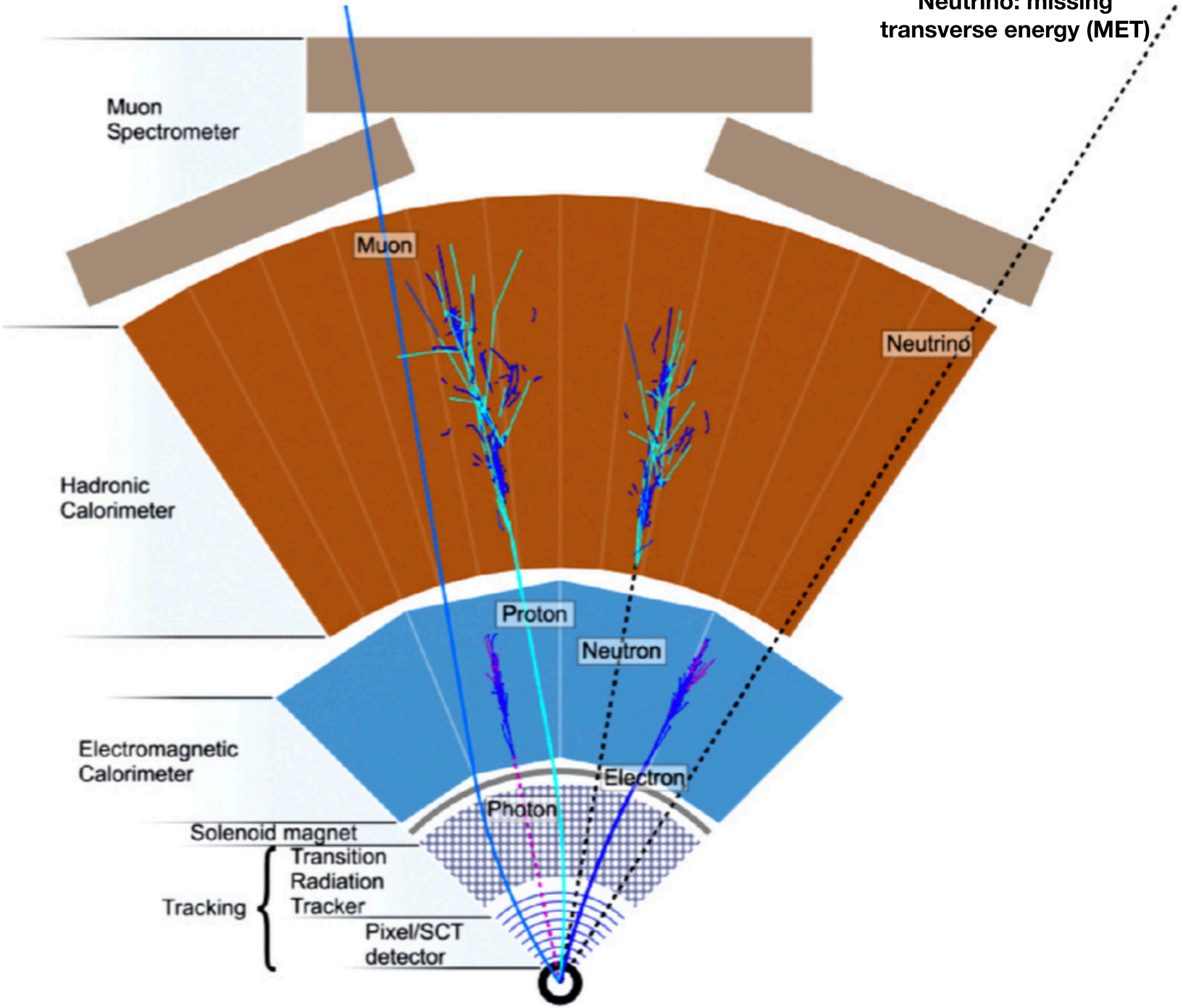
ATLAS Slice


Muon Chambers
Electromagnetic Calorimeters
Solenoid
Forward Calorimeters
End Cap Toroid
Barrel Toroid
Inner Detector
Hadronic Calorimeters

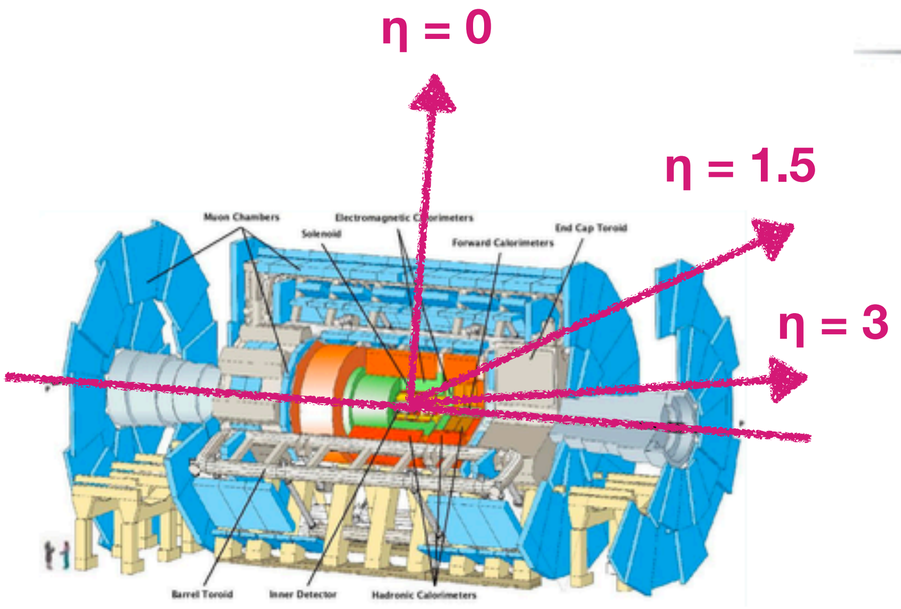

Muon Spectrometer
Muon
Neutrino
Hadronic Calorimeter
Proton
Neutron
Electromagnetic Calorimeter
Electron
Photon
Solenoid magnet
Tracking
Transition Radiation Tracker
Pixel/SCT detector

ATLAS Slice

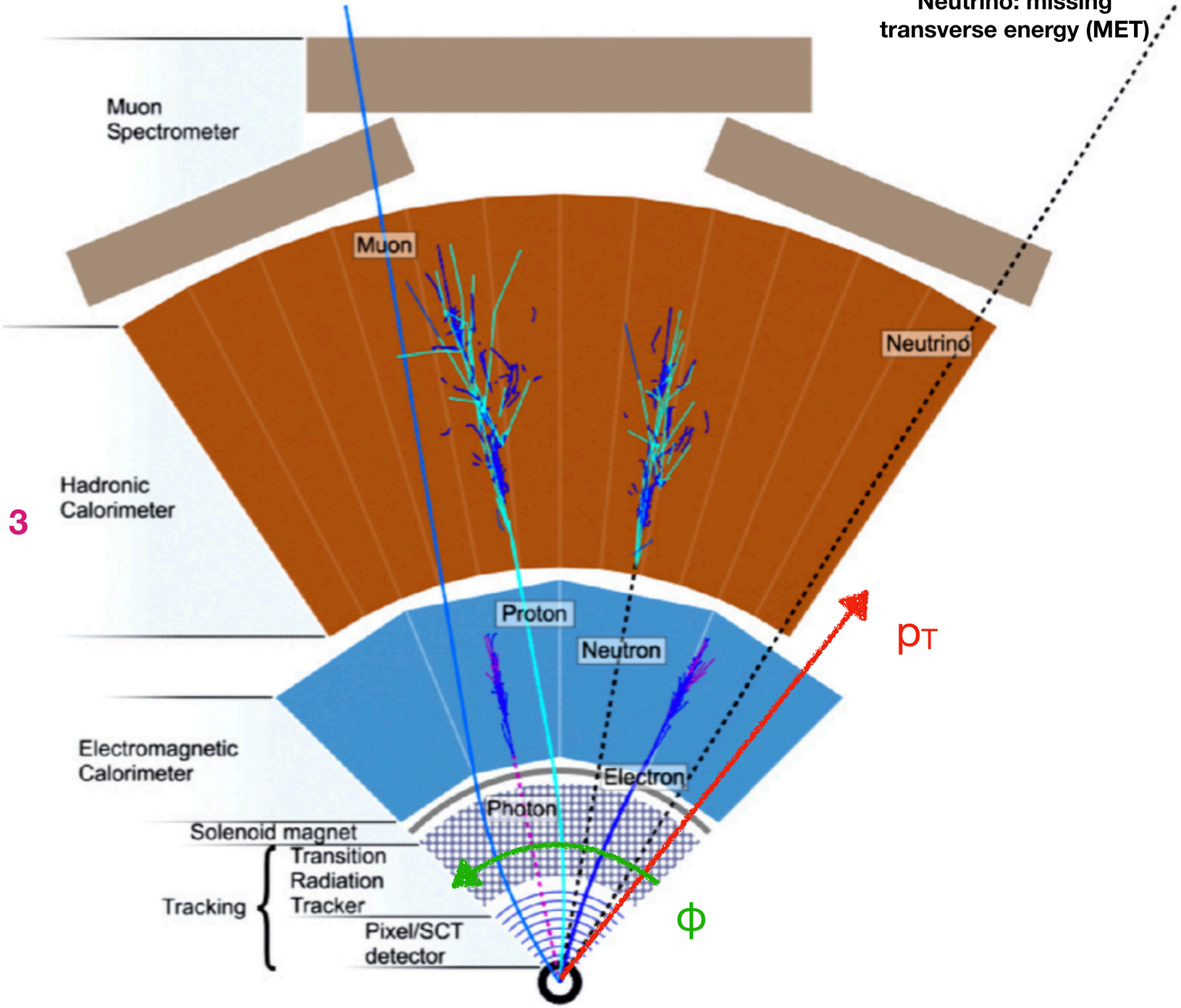

Muon Chambers
Electromagnetic Calorimeters
Solenoid
Forward Calorimeters
End Cap Toroid
Barrel Toroid
Inner Detector
Hadronic Calorimeters


Neutrino: missing transverse energy (MET)
Muon Spectrometer
Muon
Neutrino
Hadronic Calorimeter
Proton
Neutron
Electromagnetic Calorimeter
Electron
Photon
Solenoid magnet
Tracking
Transition Radiation Tracker
Pixel/SCT detector

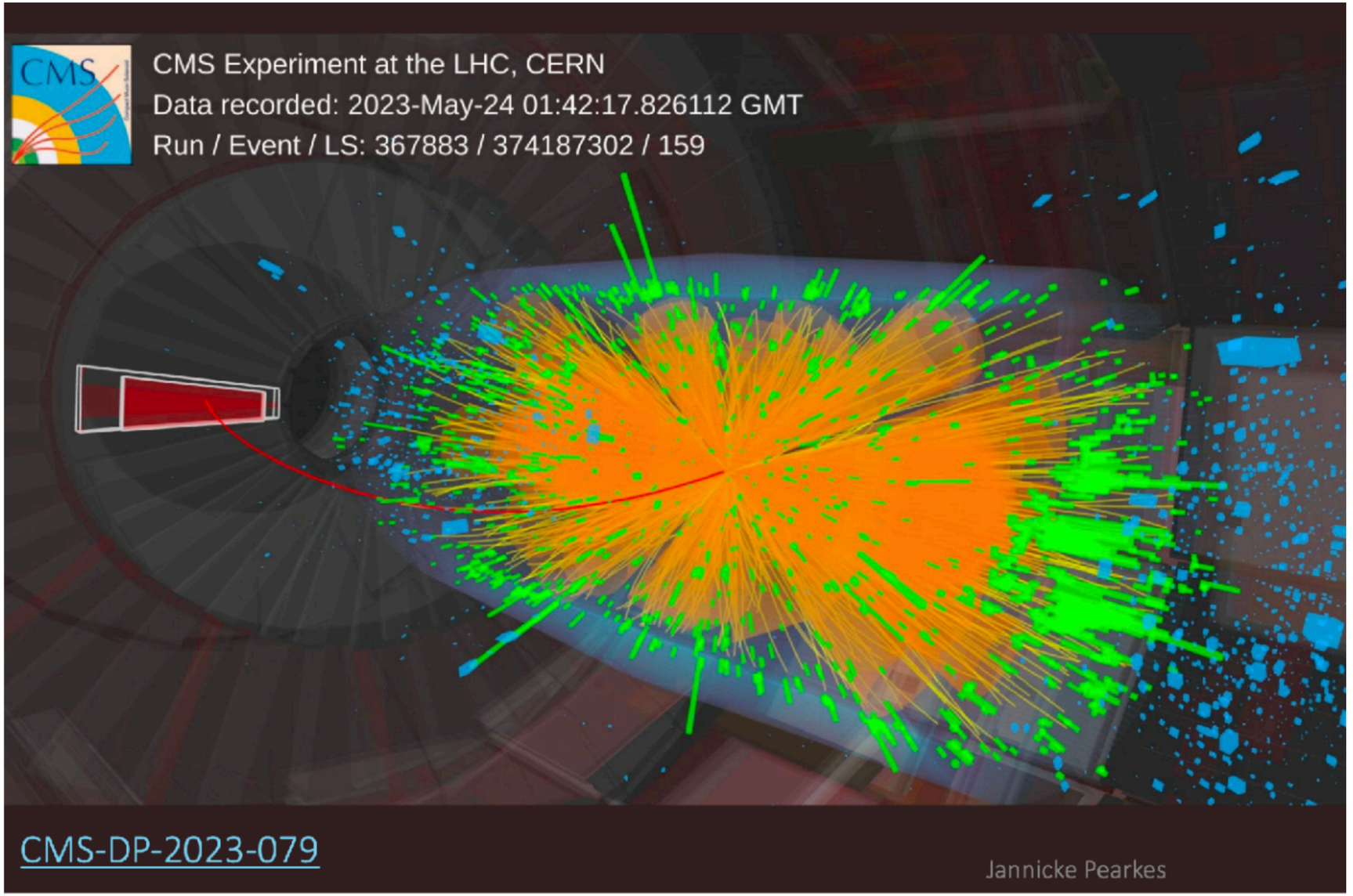
ATLAS Slice



$\mathbf{p_4} = (p_T, \eta, \phi, E)$

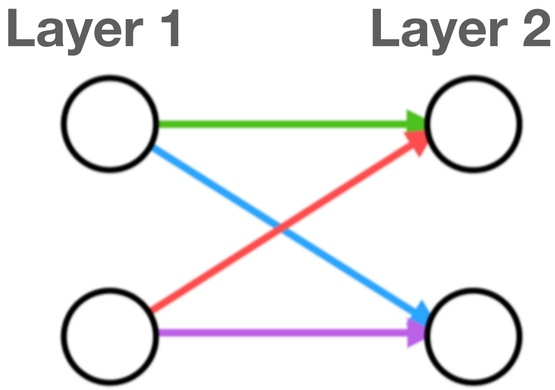


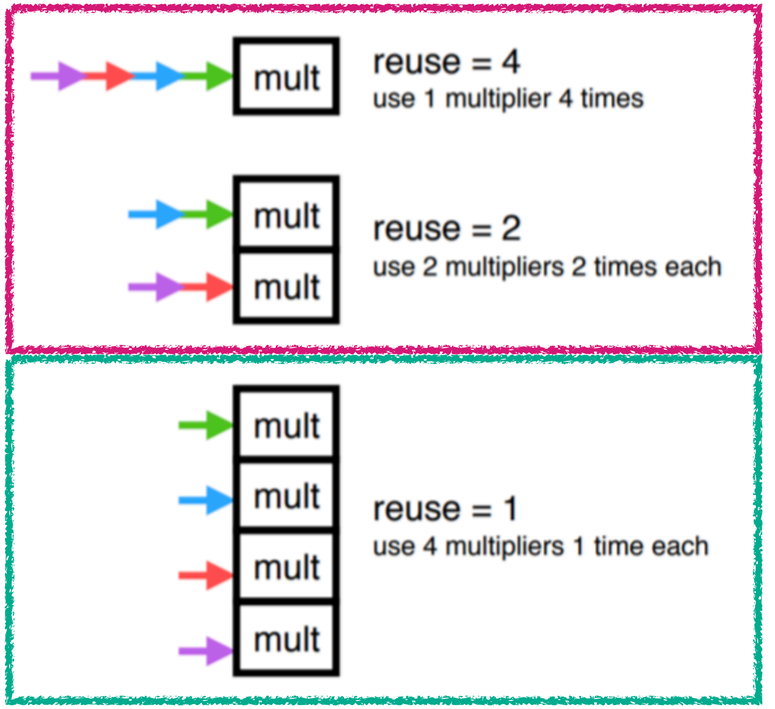
L1 AD

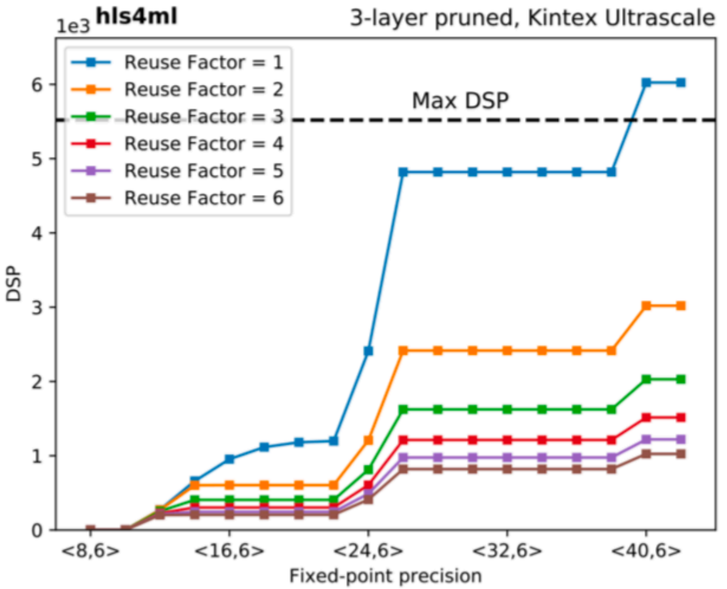

CMS
CMS Experiment at the LHC, CERN
Data recorded: 2023-May-24 01:42:17.826112 GMT
Run / Event / LS: 367883 / 374187302 / 159
CMS-DP-2023-079
Jannicke Pearkes

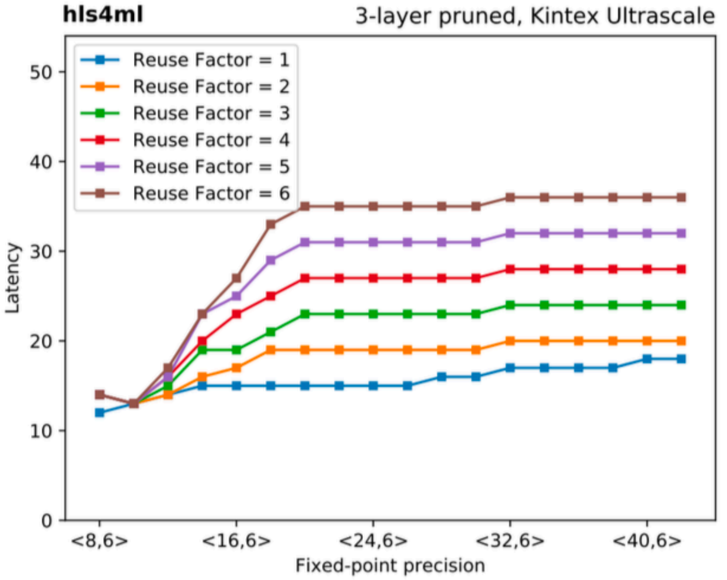
Reuse

- 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

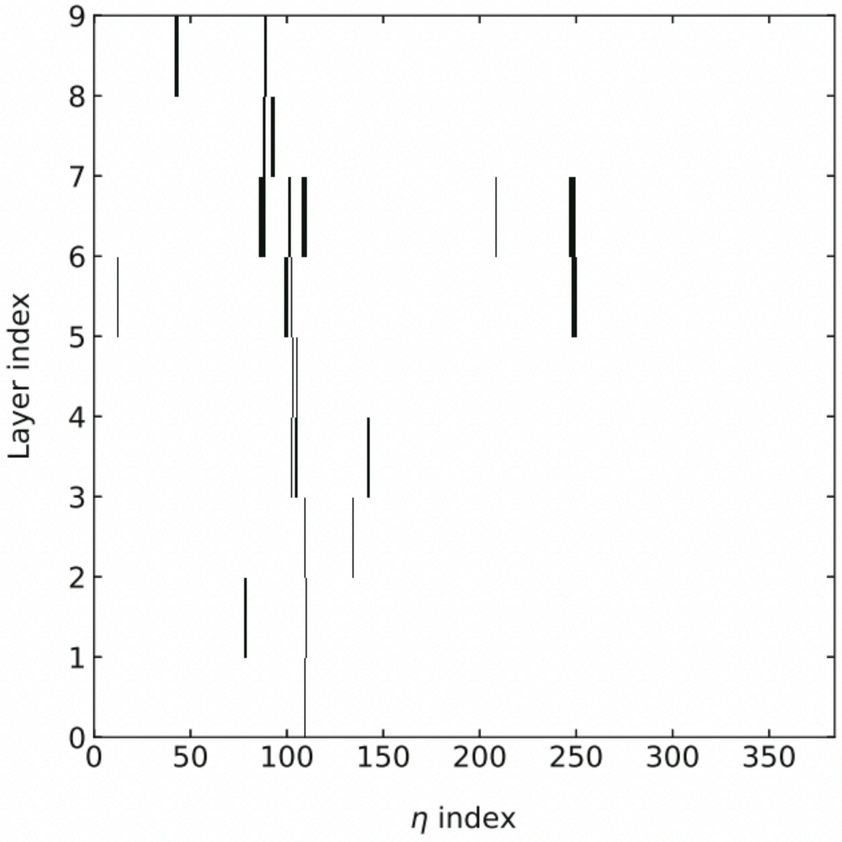




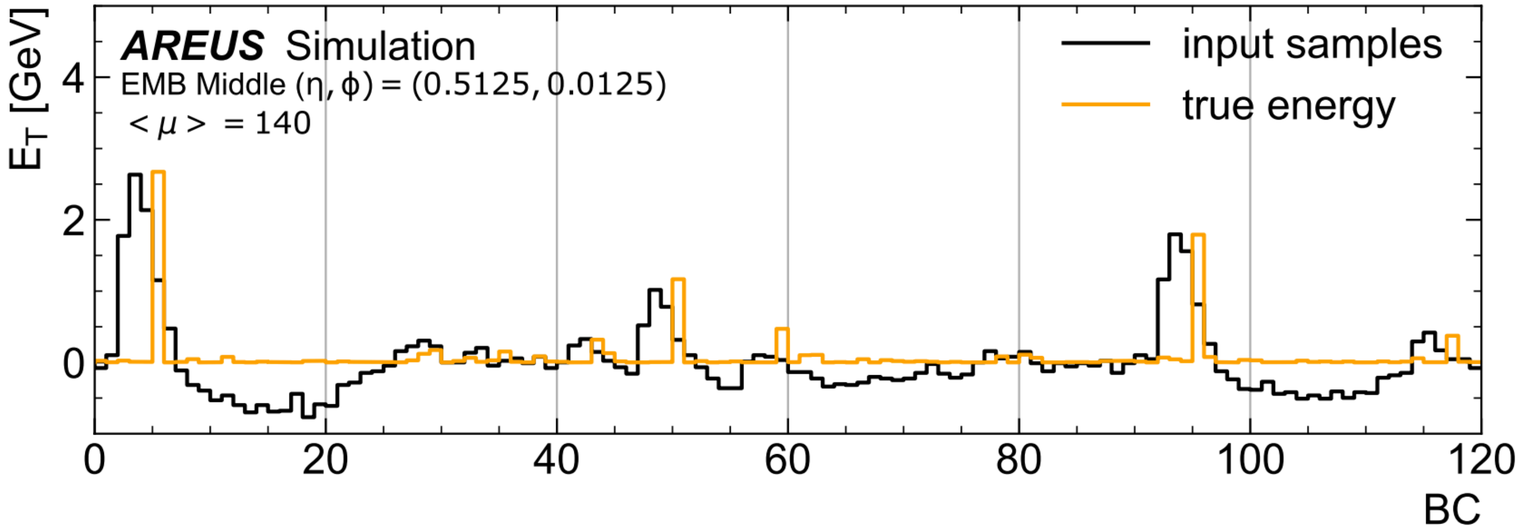




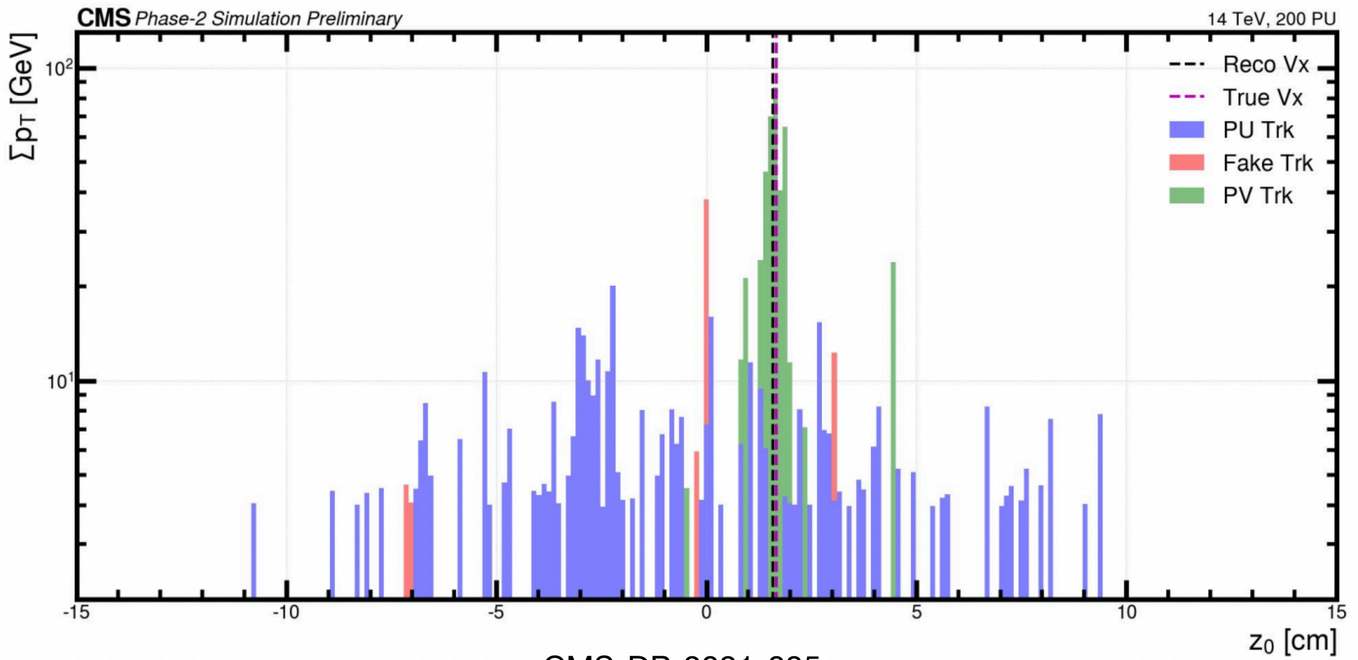
Applications



Eur. Phys. J. C (2021) 81 :969



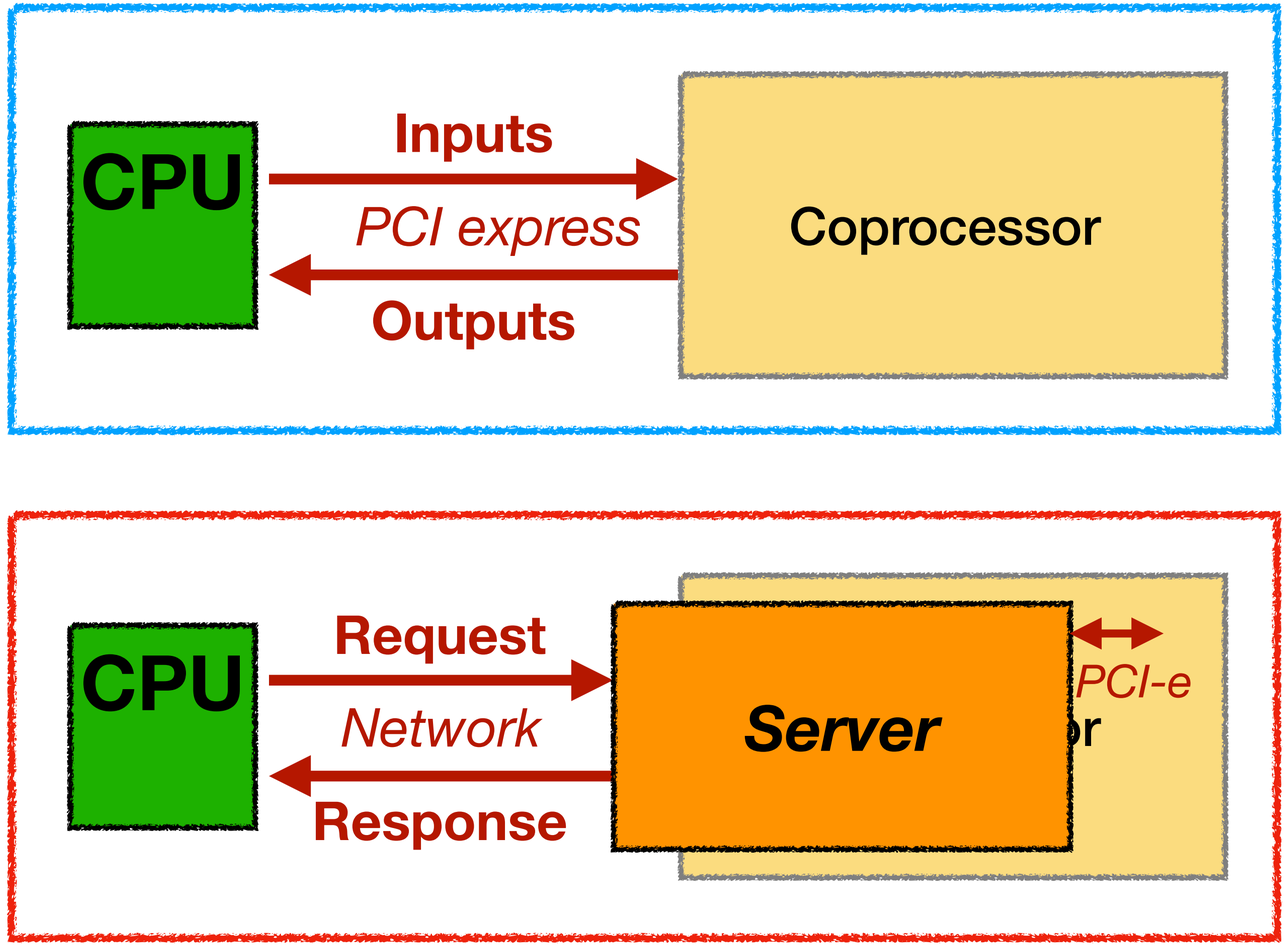
arXiv: 2111.08590



CMS-DP-2021-035

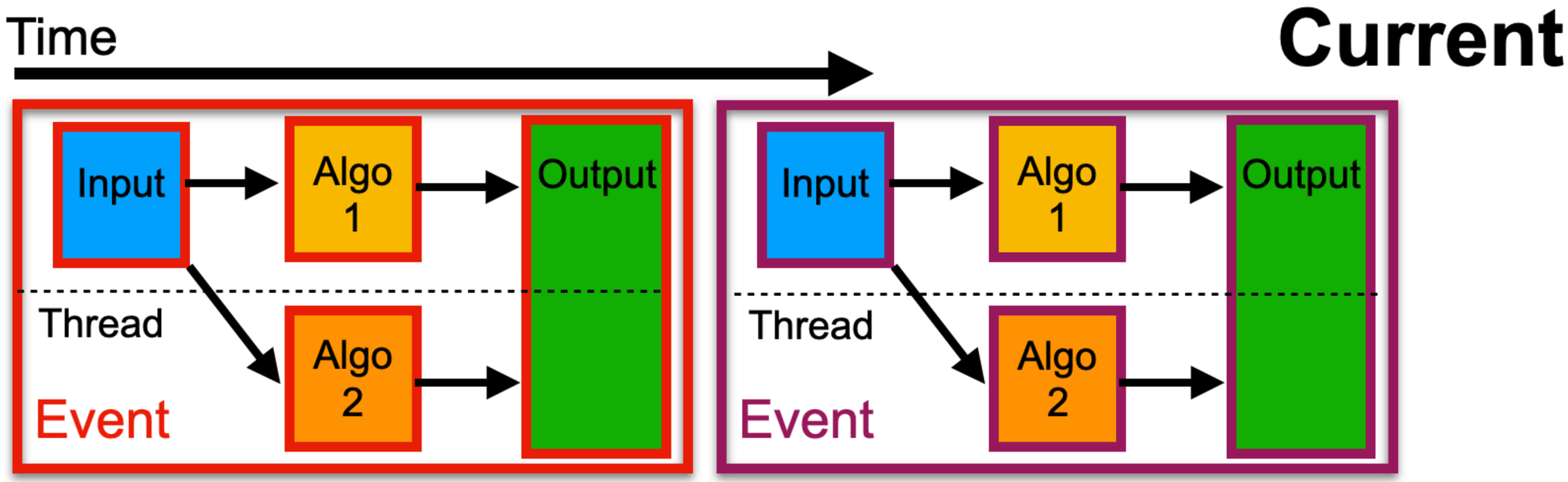
Heterogeneous Computing

- Direct connect
  - Simple connections
  - Reduced network load
- **As-a-service (aaS)**
  - Simple support for mixed hardware
  - Scaleable
  - Throughput optimizations for multiple-client
  - Simple client-side



As-a-service computing

- Biggest gains come from algorithms that are faster to run on accelerator, workflows that can be parallelized



As-a-service computing

- Biggest gains come from algorithms that are faster to run on accelerator, workflows that can be parallelized