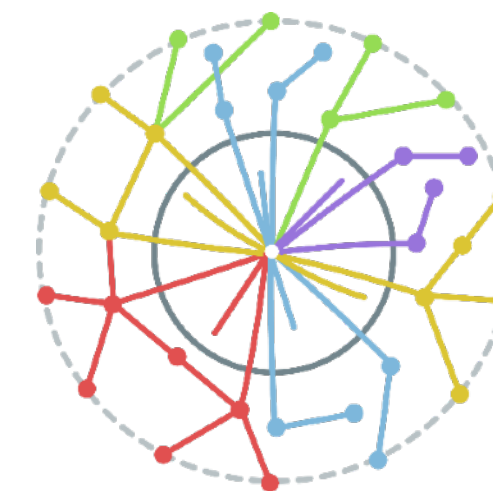
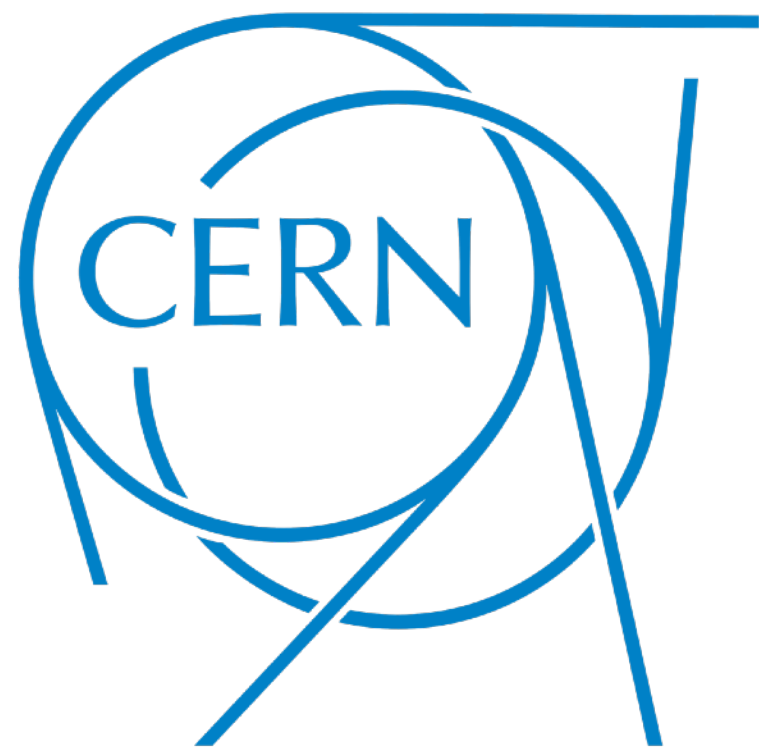


Next Generation Triggers for CMS — Level 1 Trigger and Scouting

Sioni Summers

INFN Rome

17th March 2025

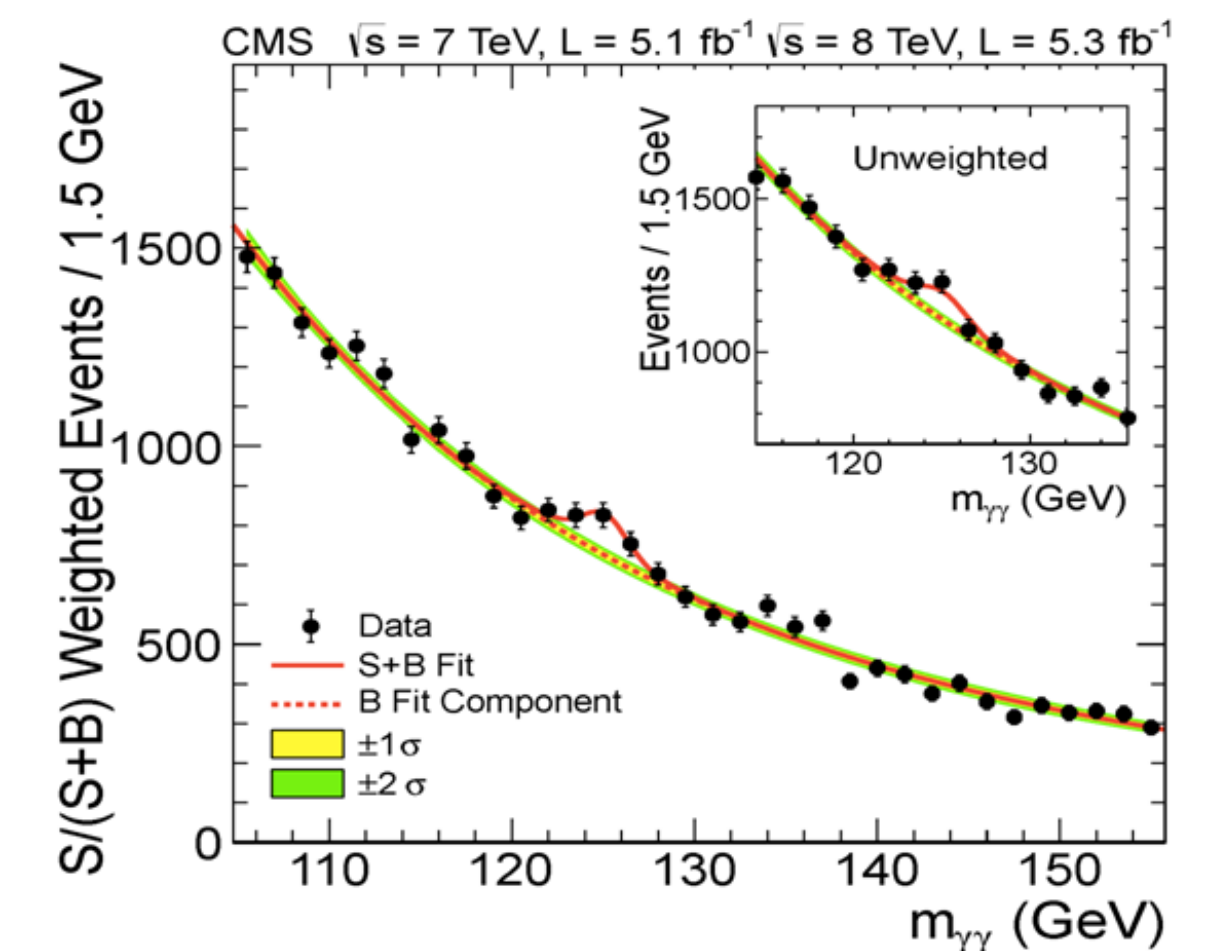
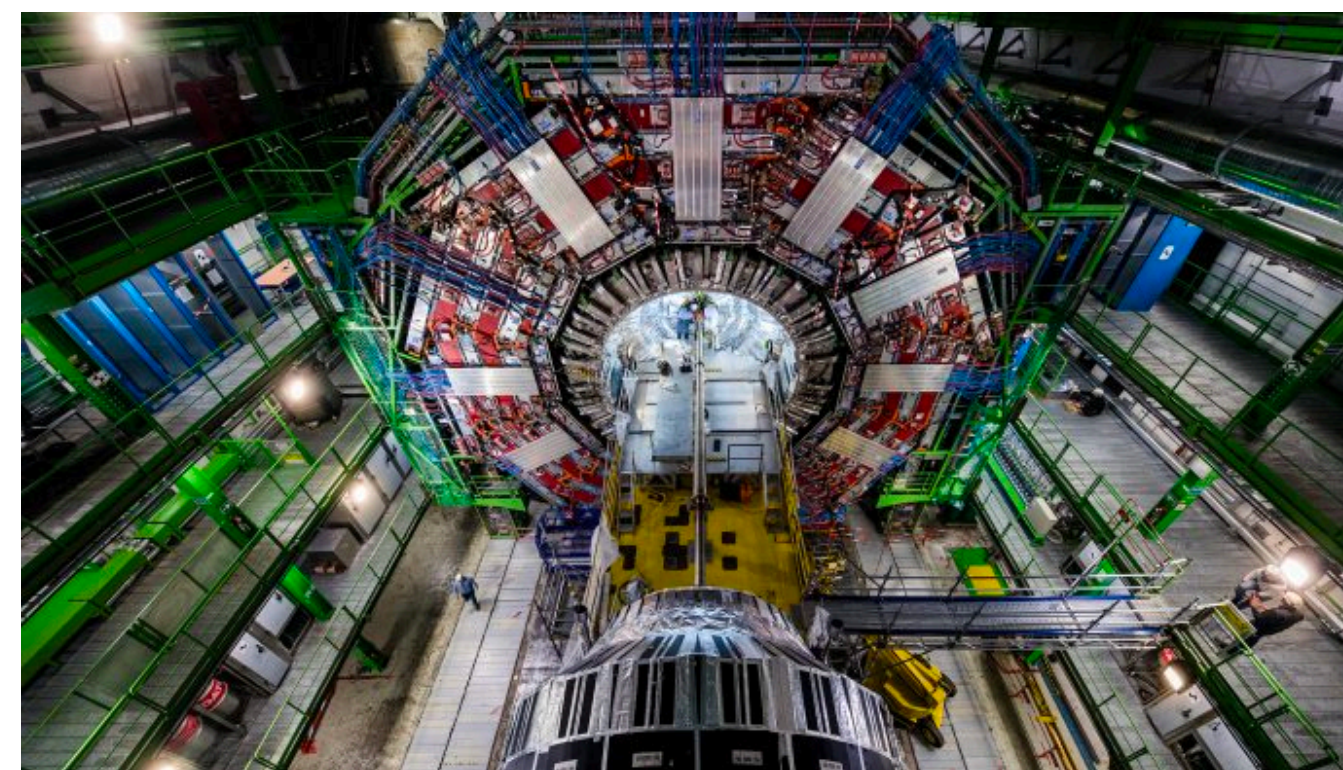
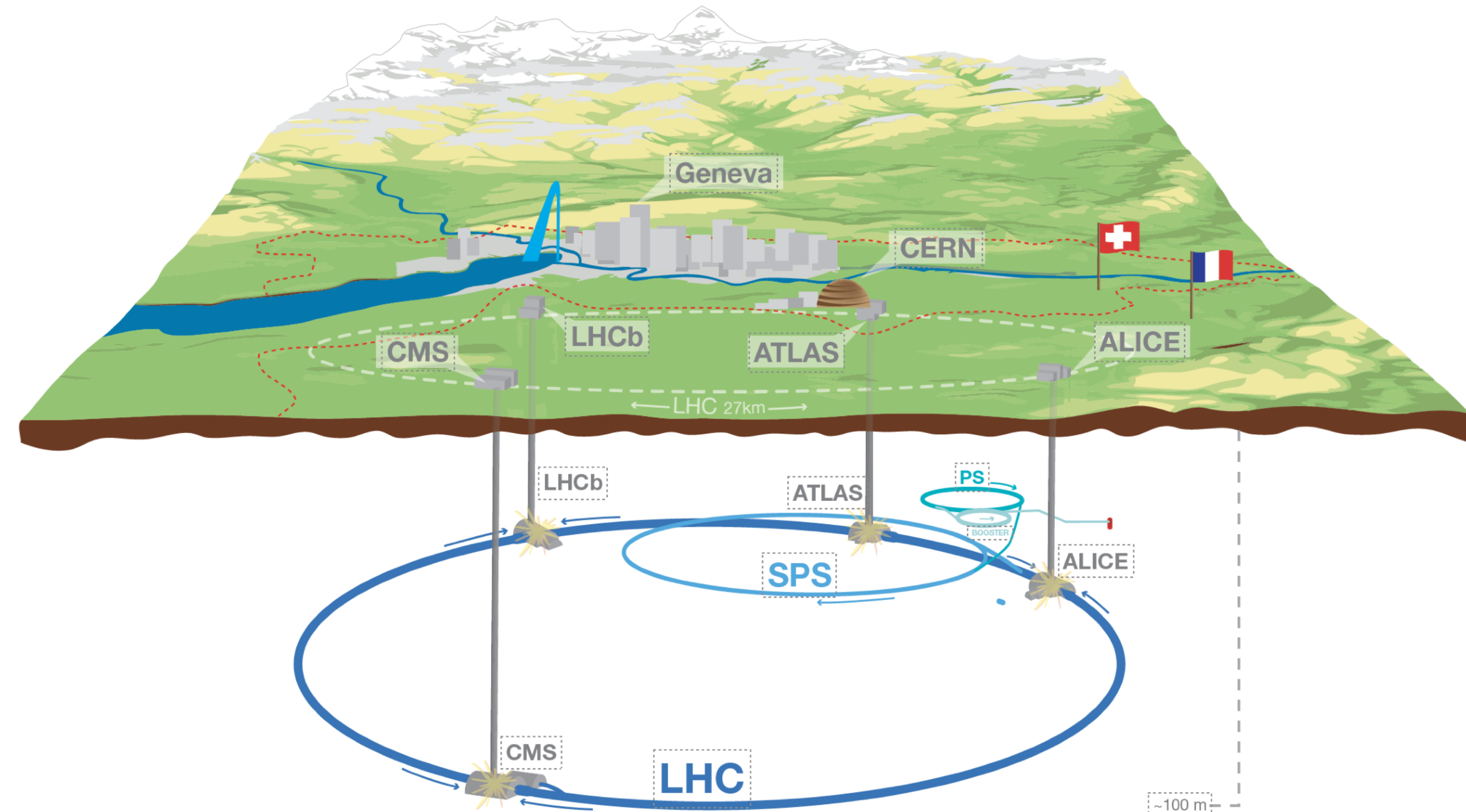


Introduction

- From 2030 the Large Hadron Collider will run as the High Luminosity LHC: delivering 10x more data between 2030-2040 than 2010-2026
 - Enabling searches for rare phenomena
- The CMS experiment will be upgraded to meet the challenge
 - Including completely new trigger system → deciding which data to keep in realtime
 - We're constructing new detectors today
 - We're designing new trigger approaches ready to run in 2030
- Next Generation Triggers is a CERN project to squeeze the most out of the increased data
- I'll talk about the CMS upgrades and triggering
 - Focussing on use of ultrafast Machine Learning for better triggering

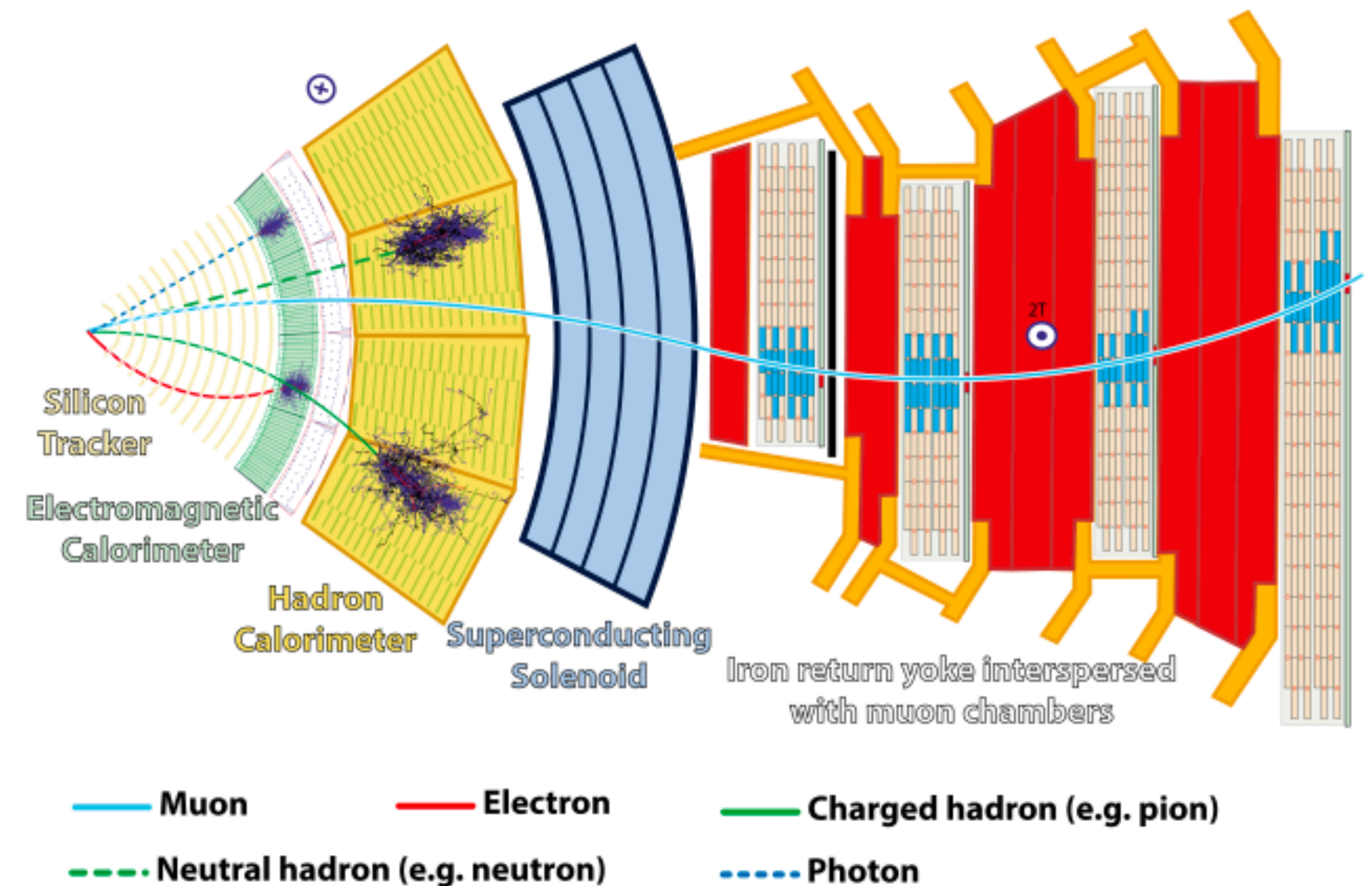
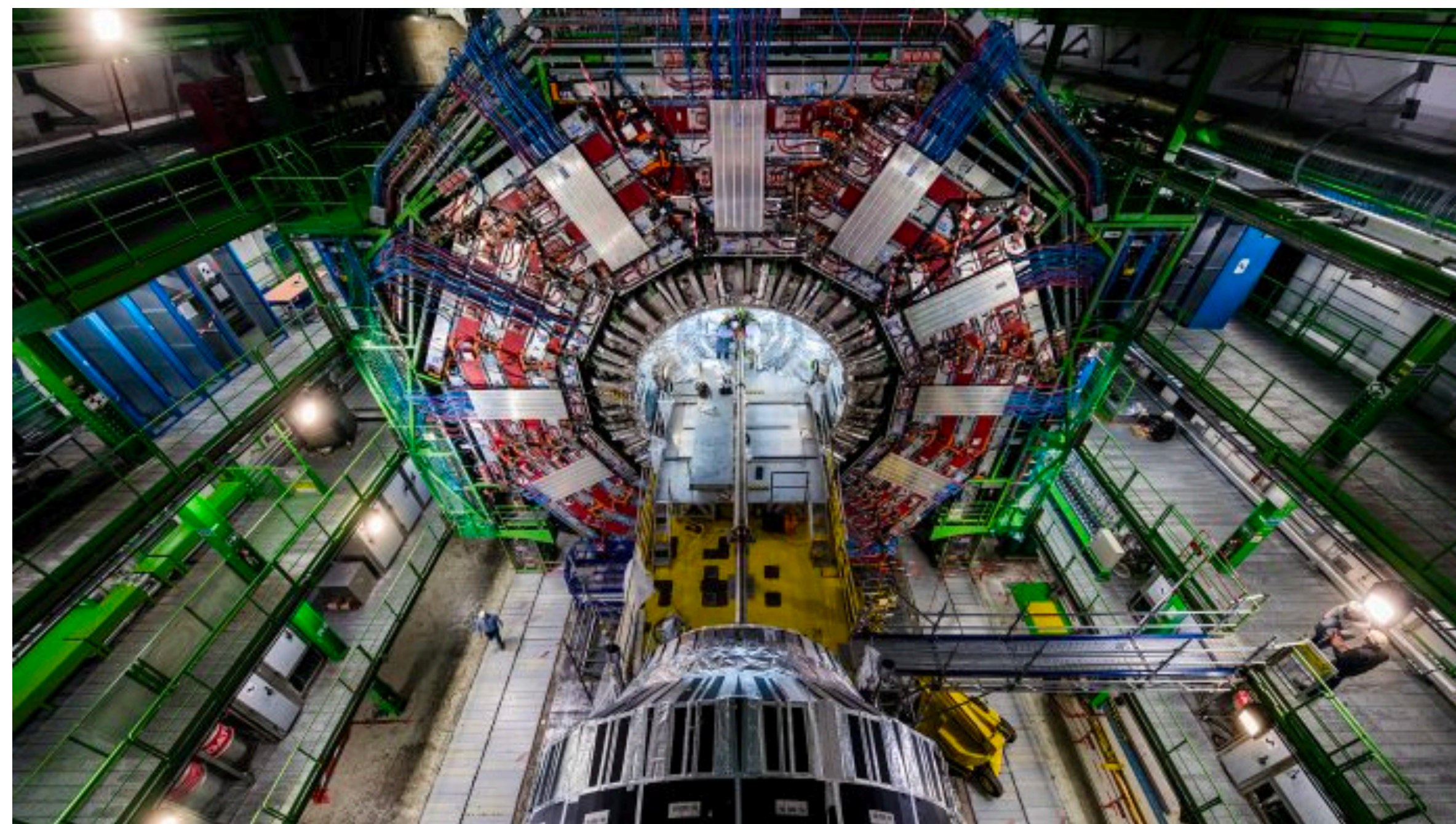
LHC

- The Large Hadron Collider is an accelerator at the Franco-Swiss border near Geneva
 - 27 km circumference ring, 100 m below ground
- Accelerating protons to 7 TeV and colliding them to study the fundamental building blocks of matter
- 4 large detectors situated around the ring detect particles produced in collisions
- We have a very precise theory of particles interactions - the Standard Model
 - Higgs boson discovered in 2012
 - But we know it's incomplete: what is the nature of dark matter? why is there more matter than antimatter?
 - We test this theory through precise measurements and searches for new physics



CMS Experiment

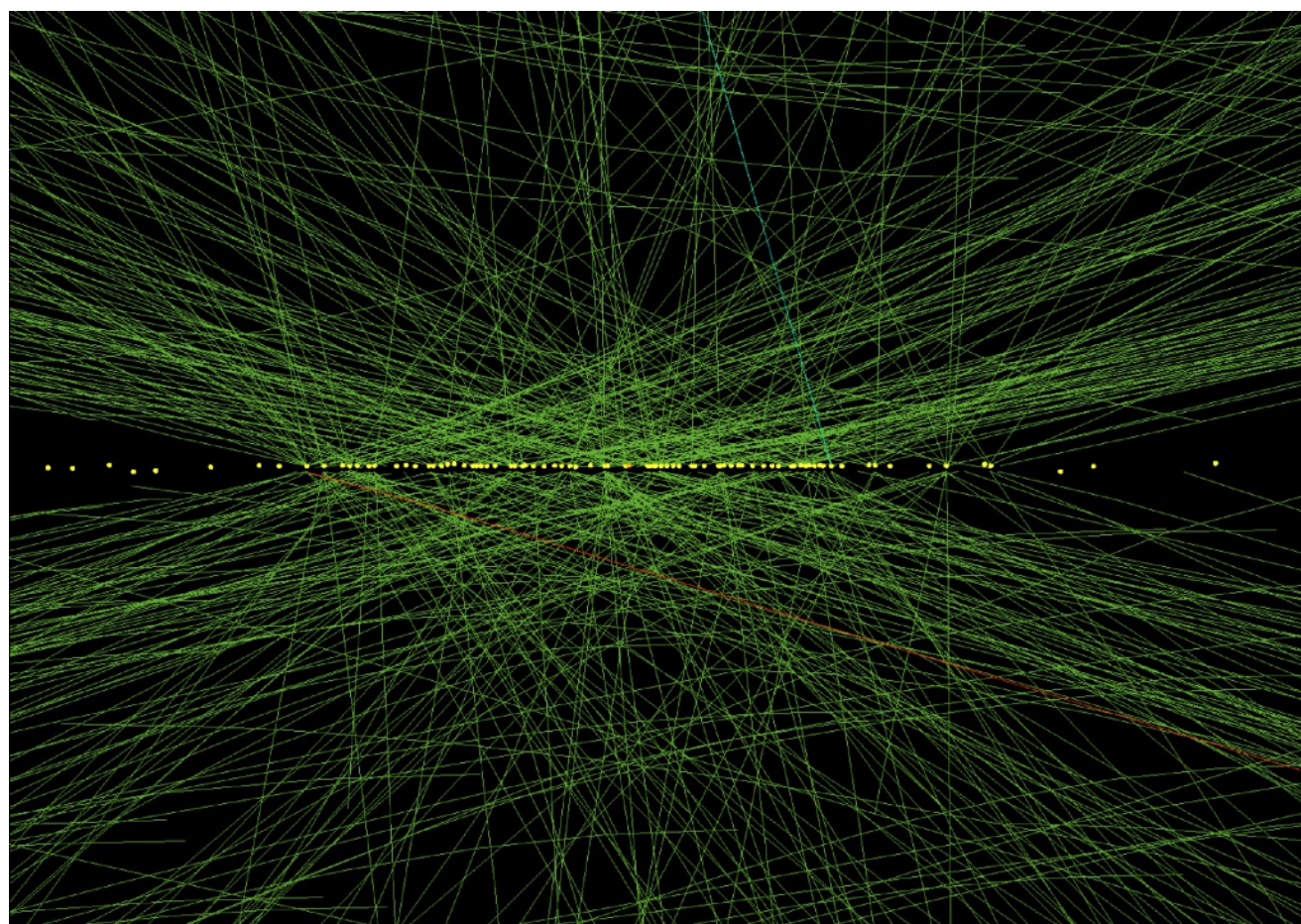
- One of the general purpose detectors at the Large Hadron Collider
- Physics program from measuring the Higgs boson (discovered 2012), to searching for new particles
- Multiple sub-detectors for detecting different types of particles (charged, neutral, muons), and their properties (momentum, energy)



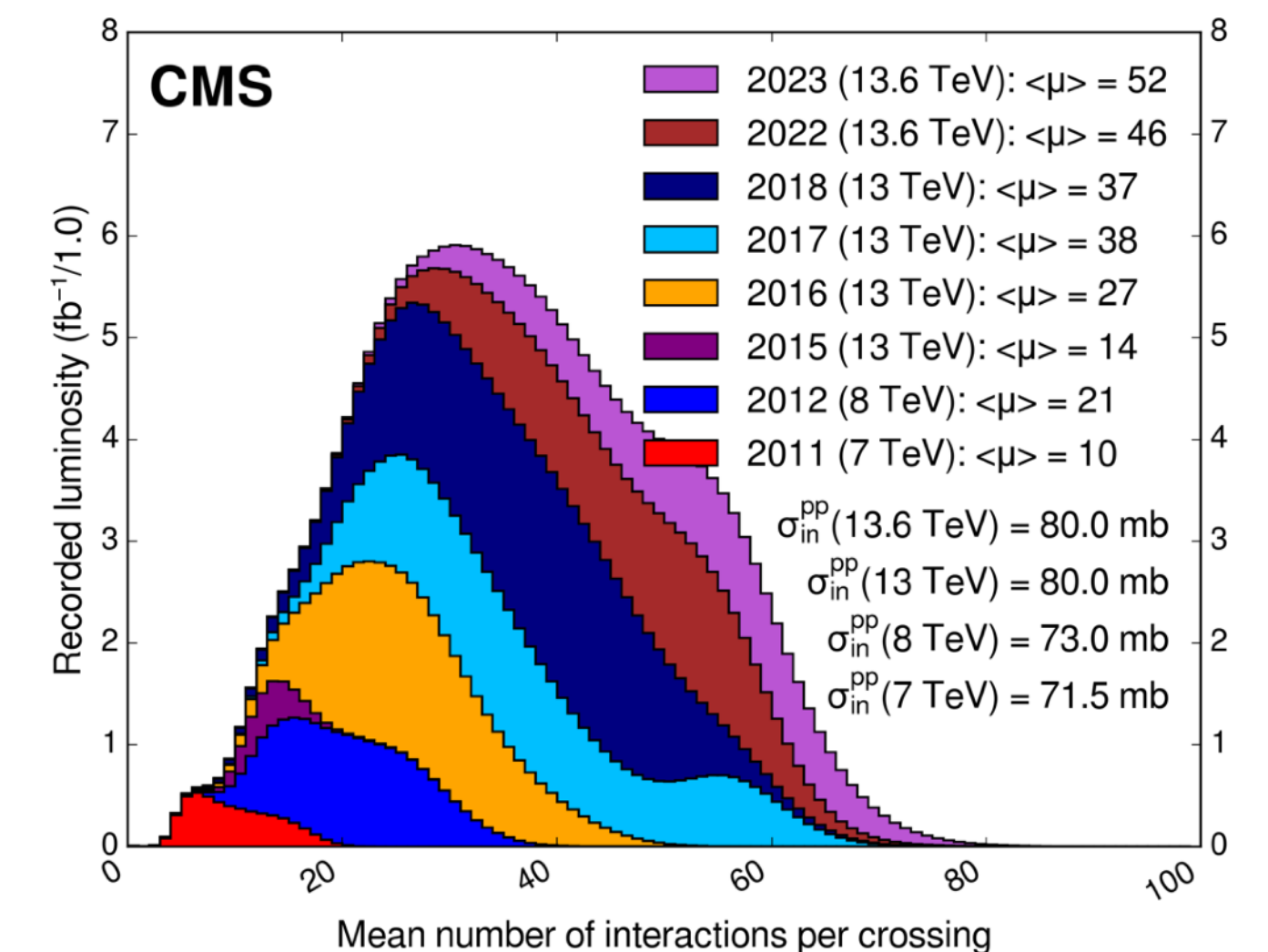
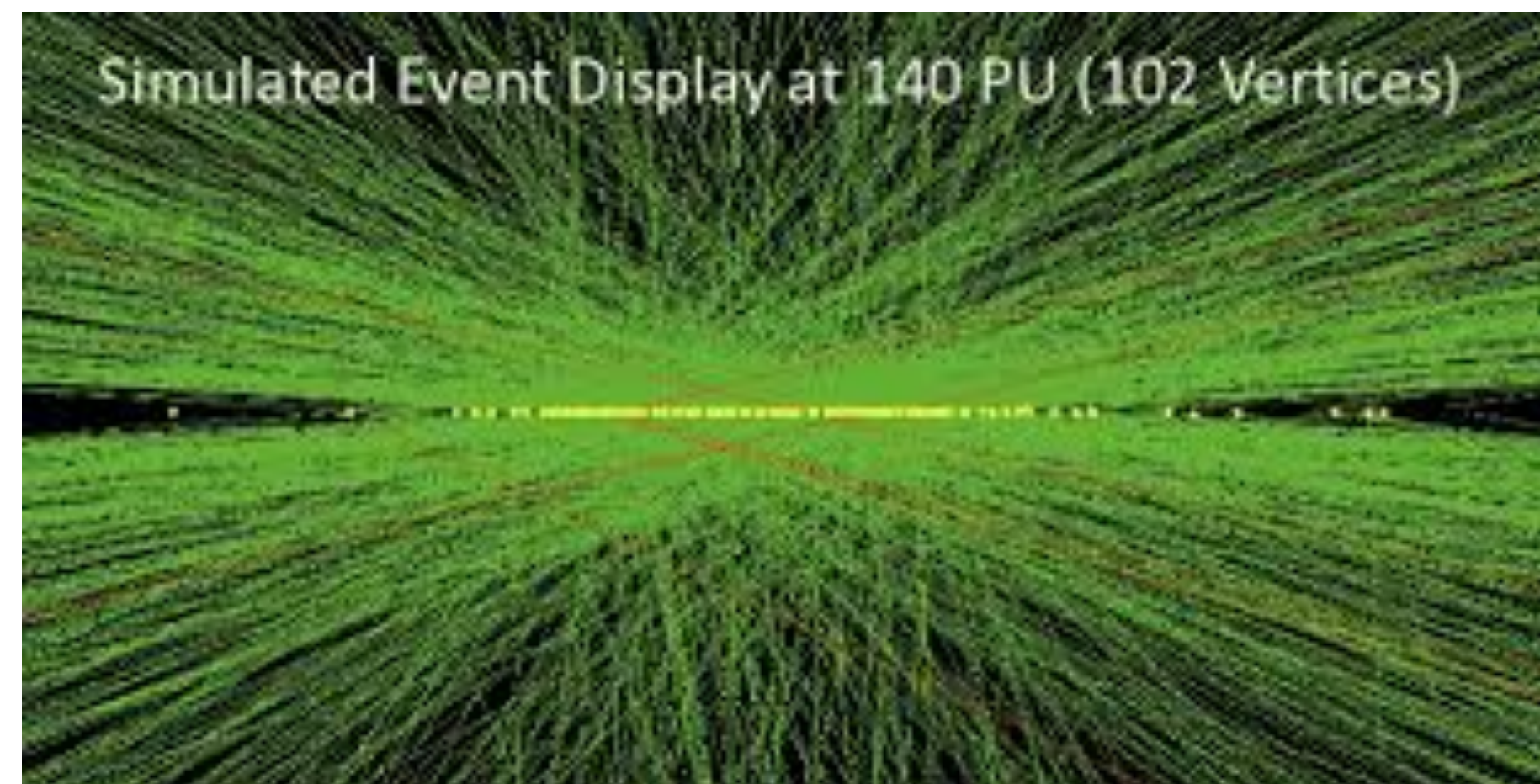
High Luminosity LHC

- LHC will upgrade to “High Luminosity LHC” by 2030
- Same collision energy of 13.6 TeV, same 40 MHz collisions, but 3-4x higher instantaneous luminosity
 - More data collected in the same running time: 10x more collision data for analysis → probe rare physics processes in detail
 - We won't be able to produce new particles with higher mass than we can today
- CMS will upgrade to “Phase 2” → replace some aged detectors, massively upgrade the trigger system
 - The trigger task is harder with more activity in each event

↓ 78 pileup vertices



↓ 140 pileup vertices



The challenge: triggering at CMS

At LHC protons collide at 40 MHz \rightarrow extreme data rates $O(100 \text{ Tb/s})$

Most collisions don't produce exciting new particles

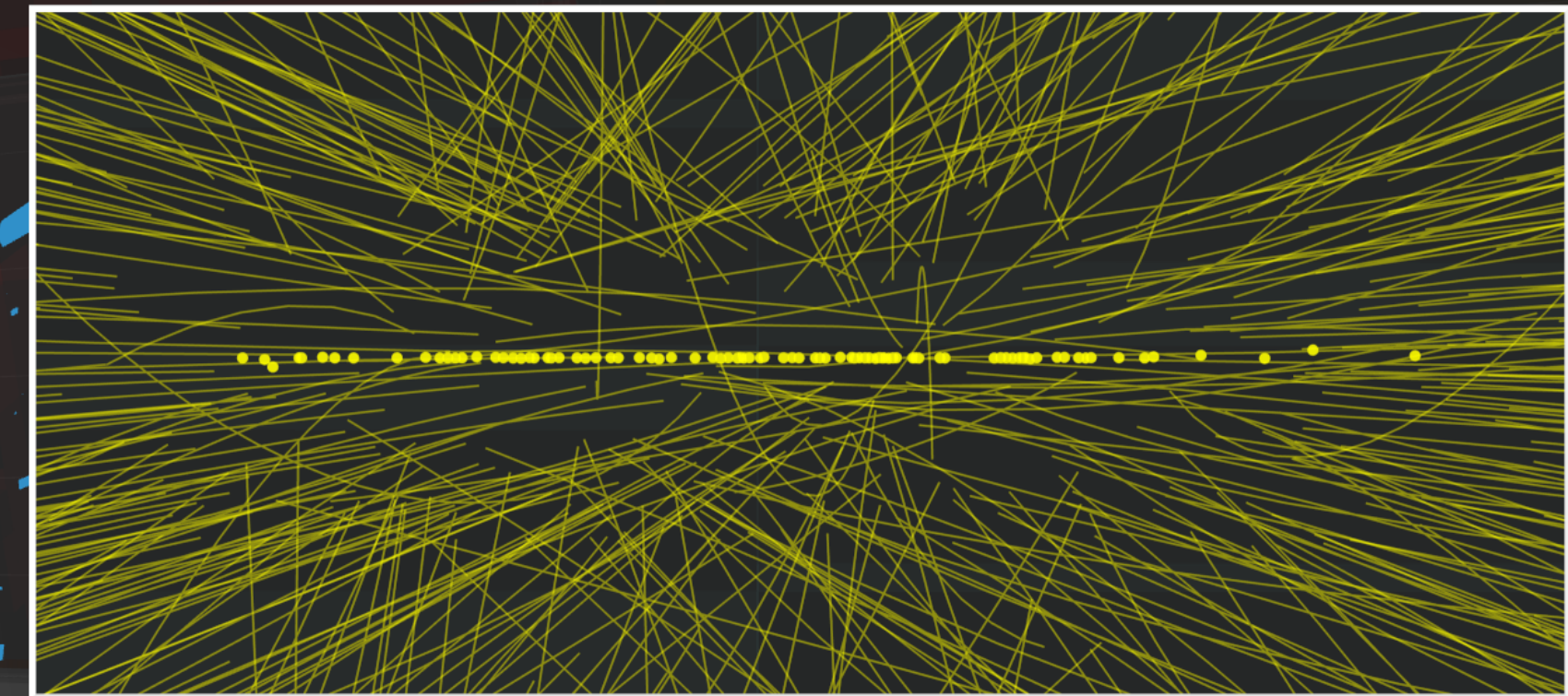
“Triggering” = filtering events to reduce data rates to manageable levels



CMS Experiment at the LHC, CERN

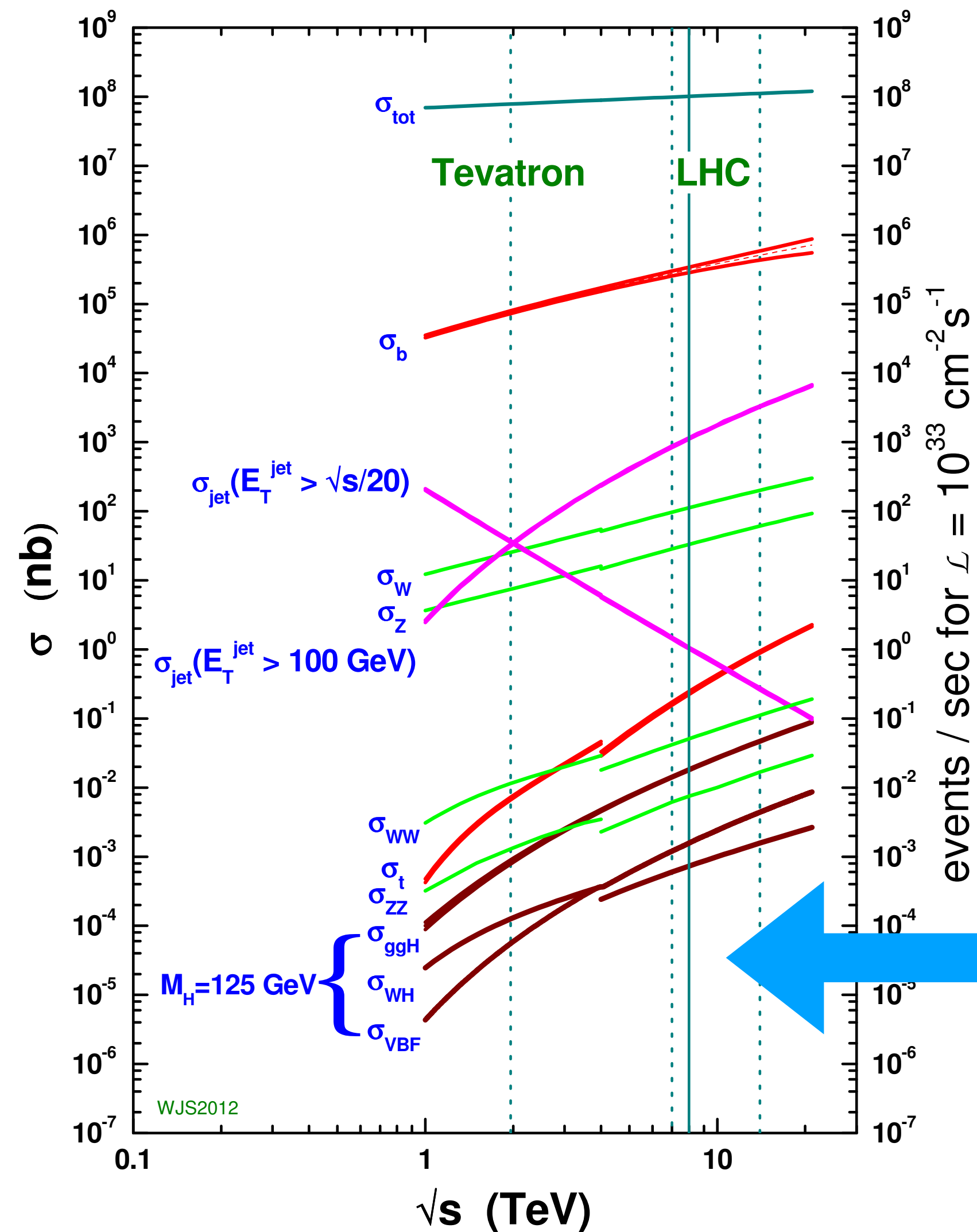
Data recorded: 2023-May-24 01:42:17.826112 GMT

Run / Event / LS: 367883 / 374187302 / 159

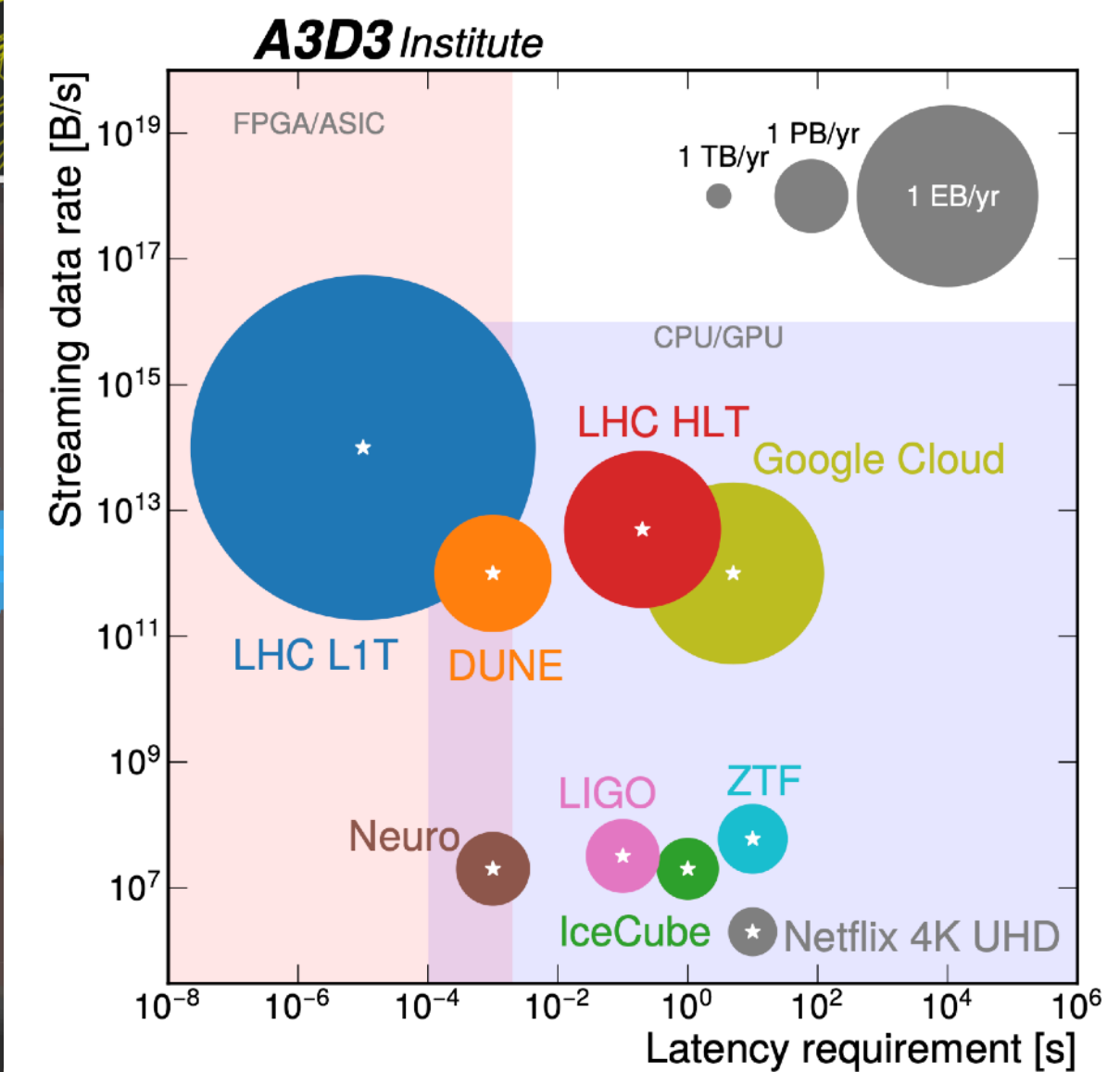
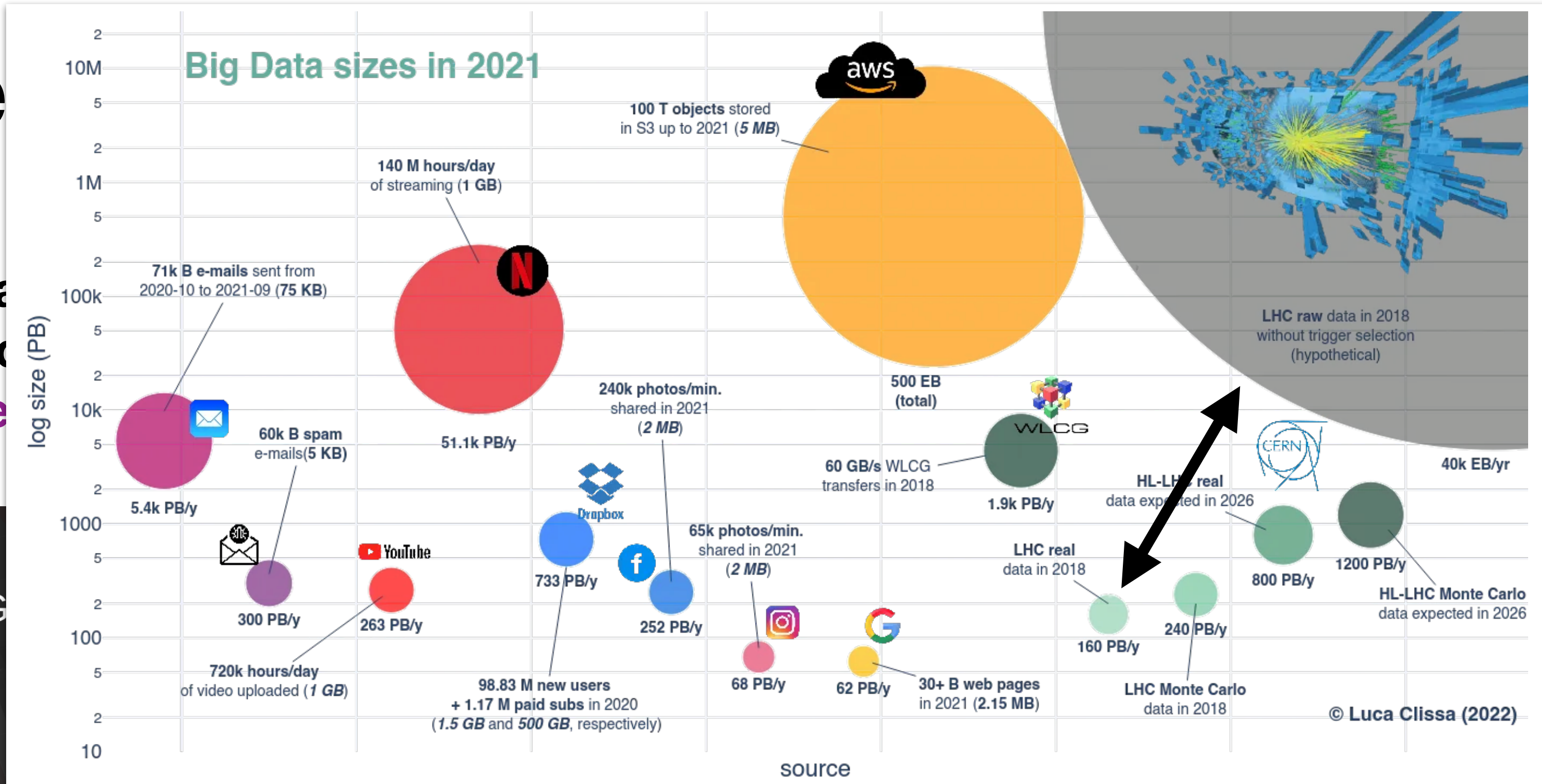


The challenge

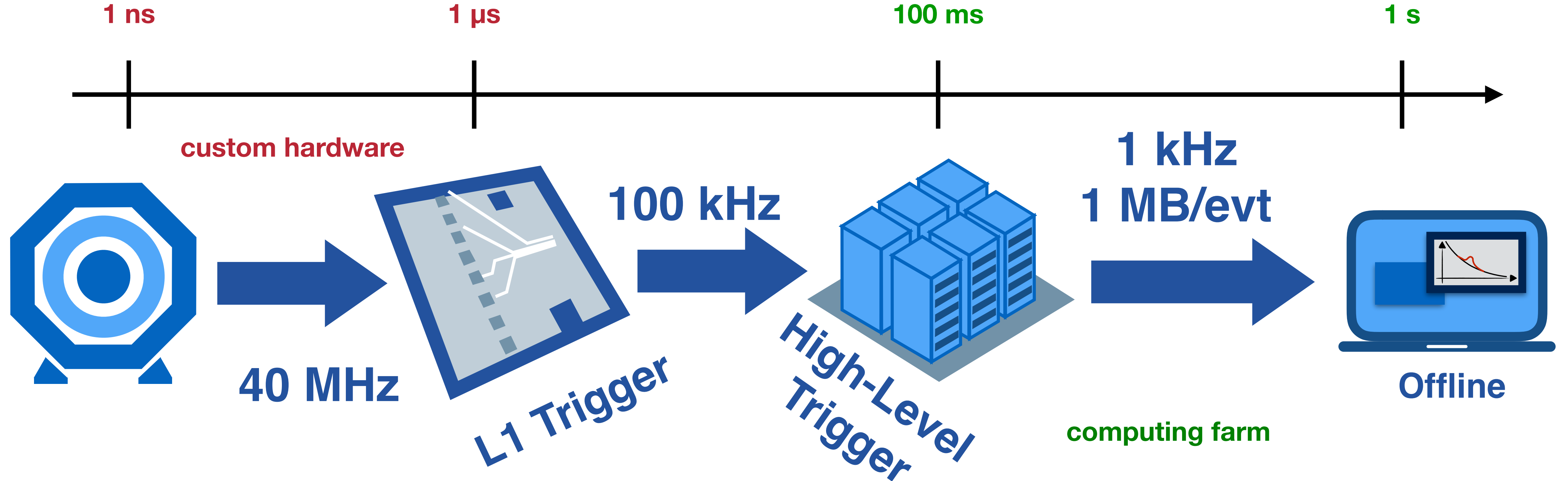
proton - (anti)proton cross sections



40 million collisions per second \rightarrow
roughly 1 Higgs boson
produced per second



Trigger systems at LHC



Triggering performed in multiple stages @ ATLAS and CMS

Reduce data rate in stages

Process 100s Tb/s

Trigger decision to be made in latency $O(\mu$ s)

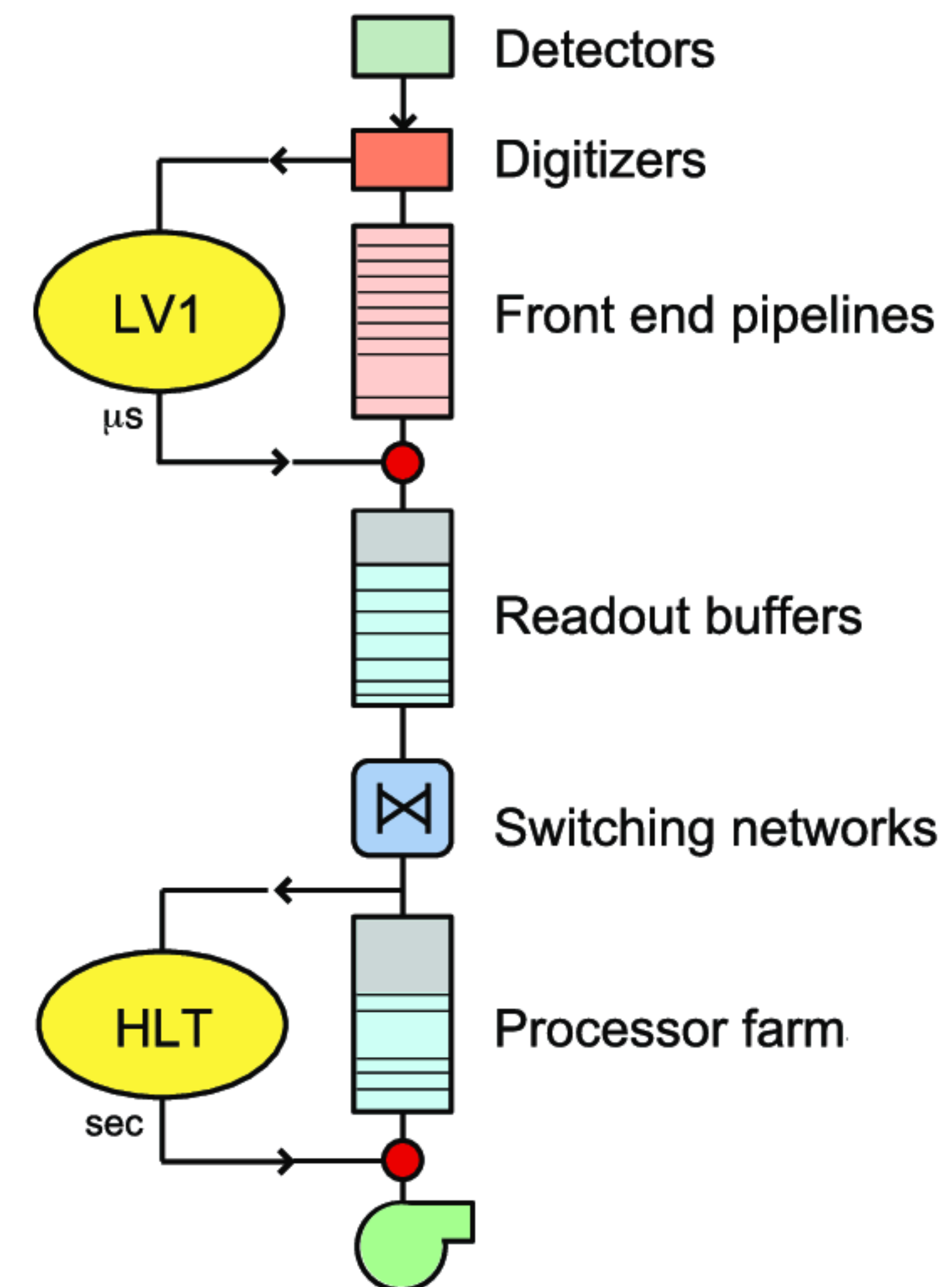
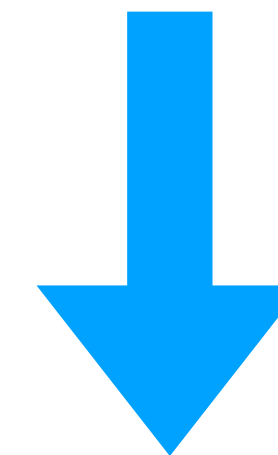
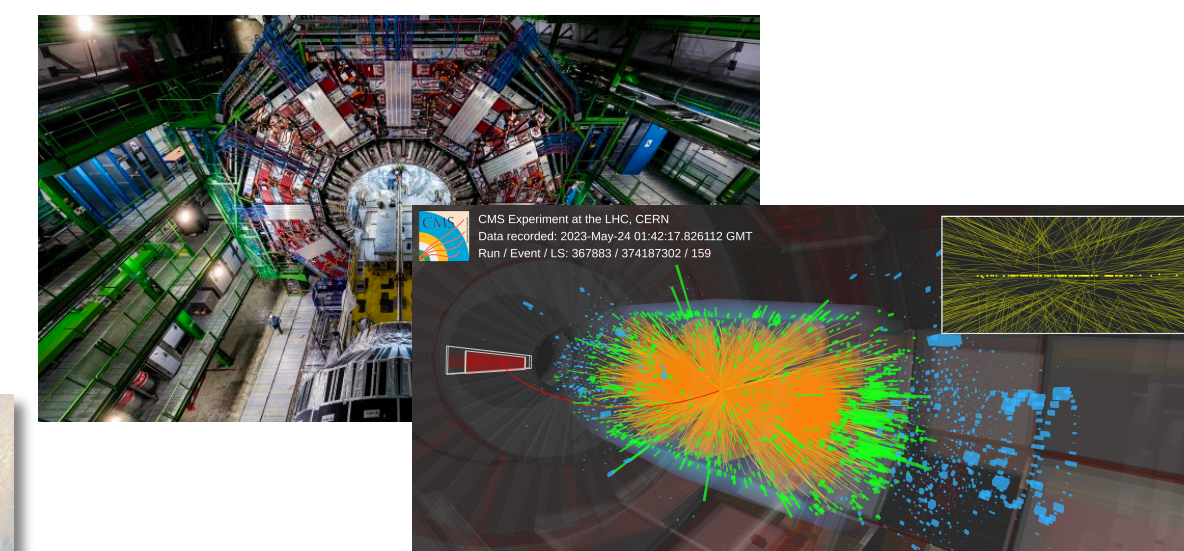
Frontends in radiation hard ASICs, processing in FPGAs

Computing farm for detailed analysis of the full event

Latency $O(100$ ms)

CMS Trigger

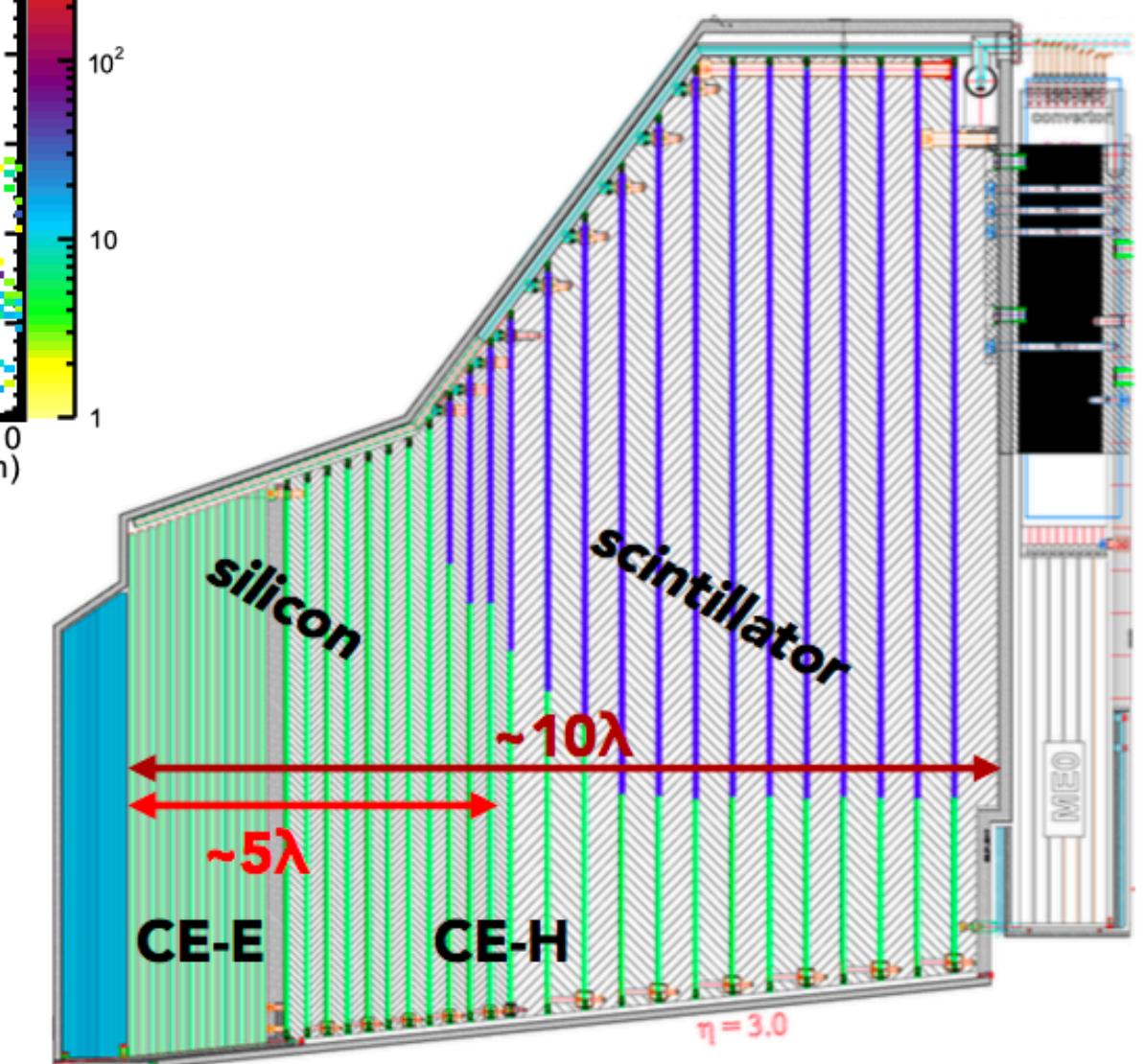
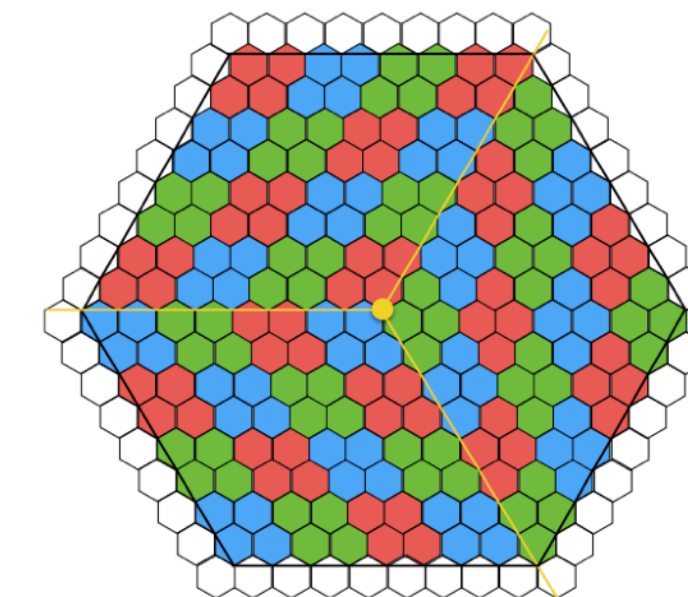
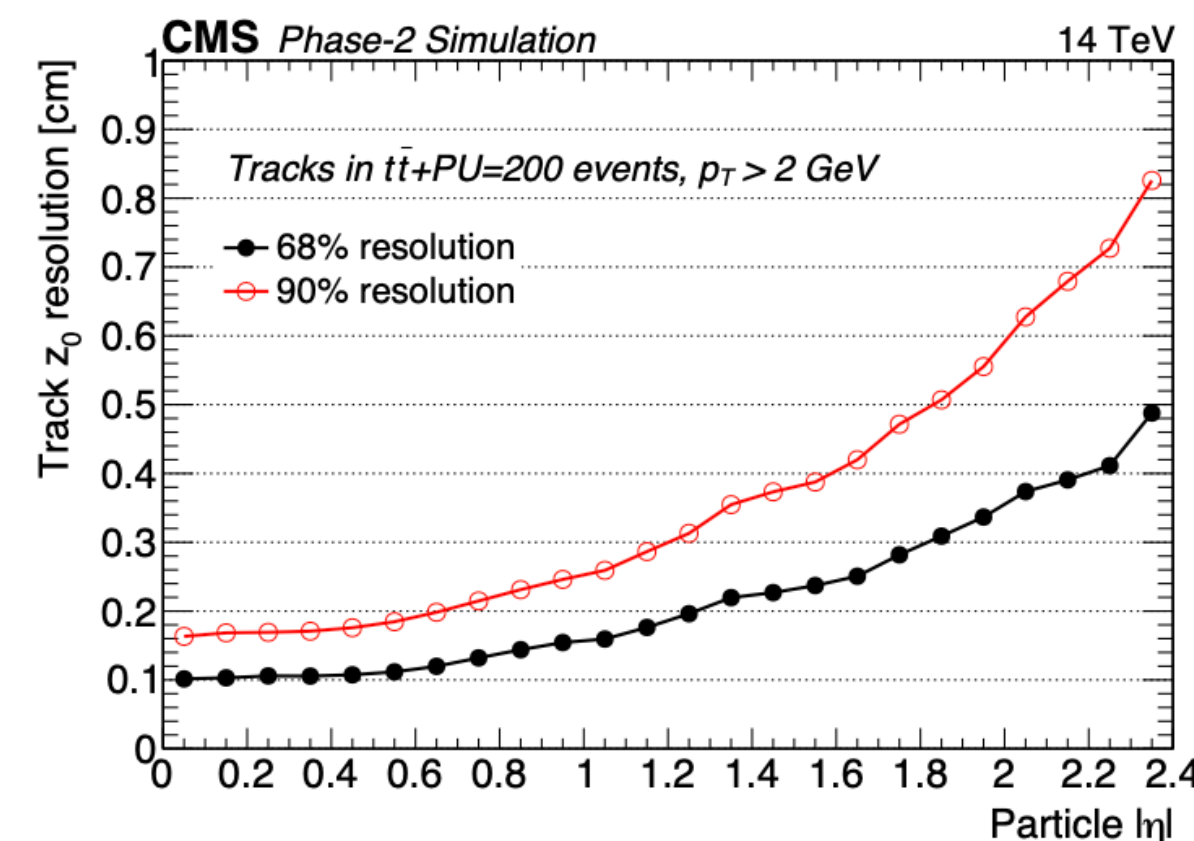
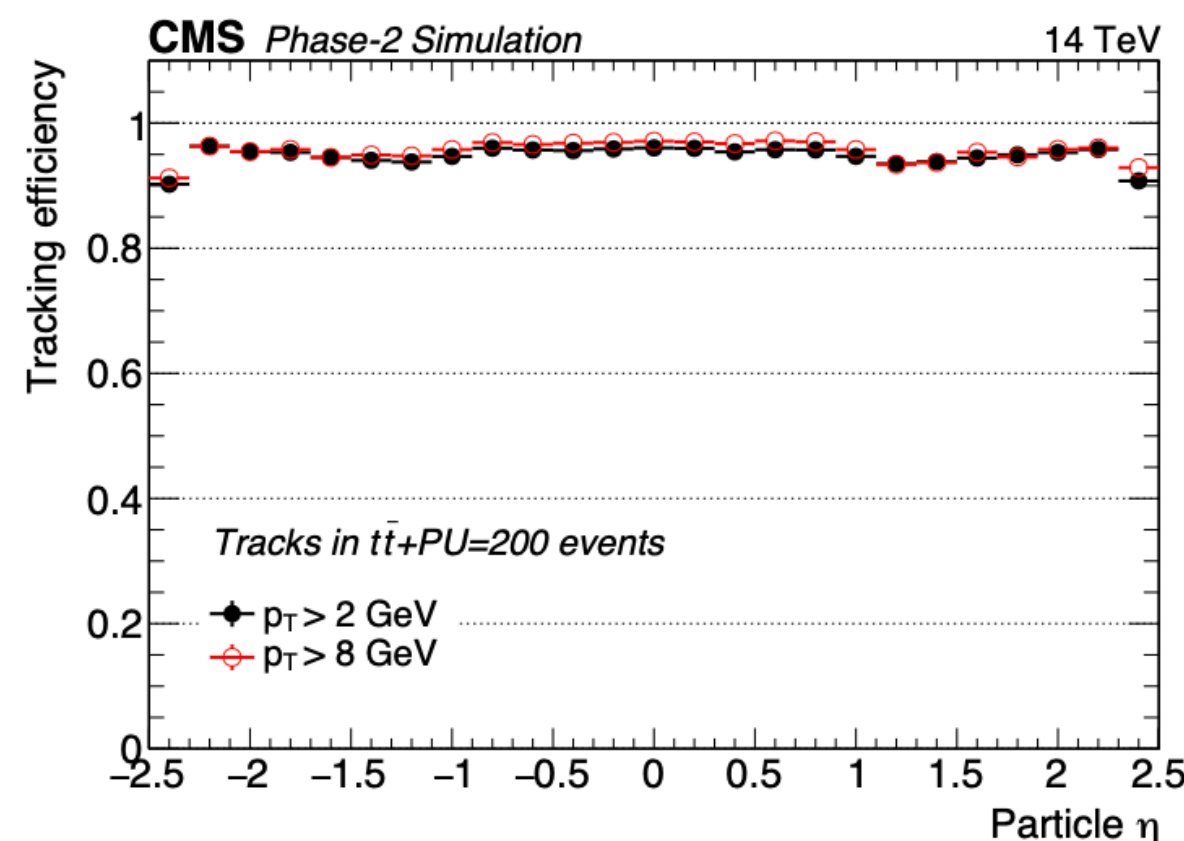
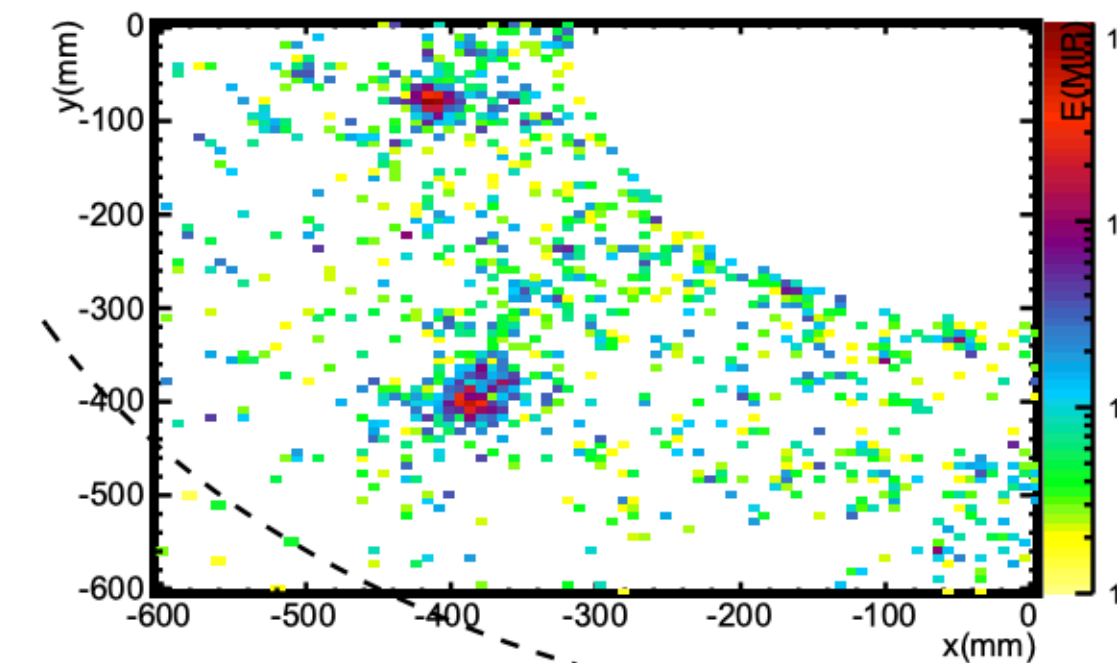
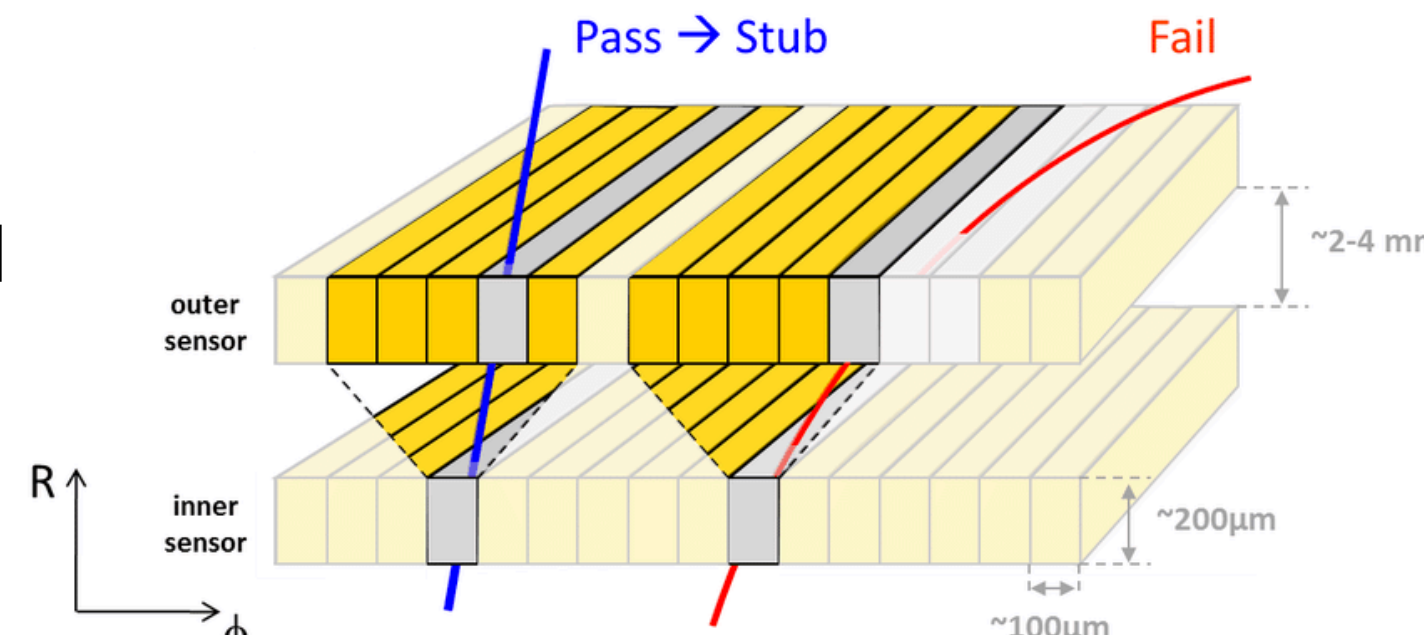
- Two stage system
- Level 1 Trigger
 - below ground
 - uses a subset of detector data
 - full data stays on the detector until triggered
 - all FPGAs on custom boards
 - 40 MHz event *throughput* in, 1 MHz out
 - 12 μs *latency* per event
- High Level Trigger
 - above ground, all commercial CPUs & GPUs
 - uses full detector data
 - 1 MHz events in, 1 kHz out
 - Roughly 100 ms per event



CMS Phase 2: New Detectors

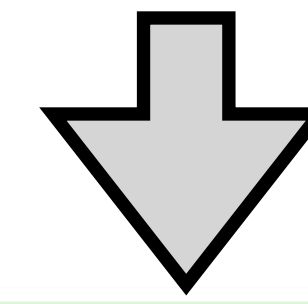
- **Track Reconstruction** at Level 1 Trigger for first time up to $|\eta| < 2.4$
- “Stubs” with $p_T > 2$ GeV will be sent to L1T from outer tracker
- Tracks in the Level 1 Trigger essential for 200 PU conditions
 - Primary vertex reconstruction, particle reconstruction
- L1T Track finding in around 200 FPGA processors
 - “Join the dots”

- **High granularity calorimeter:** silicon sampling calorimeter for the endcaps ($1.5 < |\eta| < 3$)
- 6.5 million channels (1 million to trigger) in 47 layers
 - Very fine transverse and longitudinal segmentation
- Around 200 FPGAs for 3D cluster reconstruction in L1T

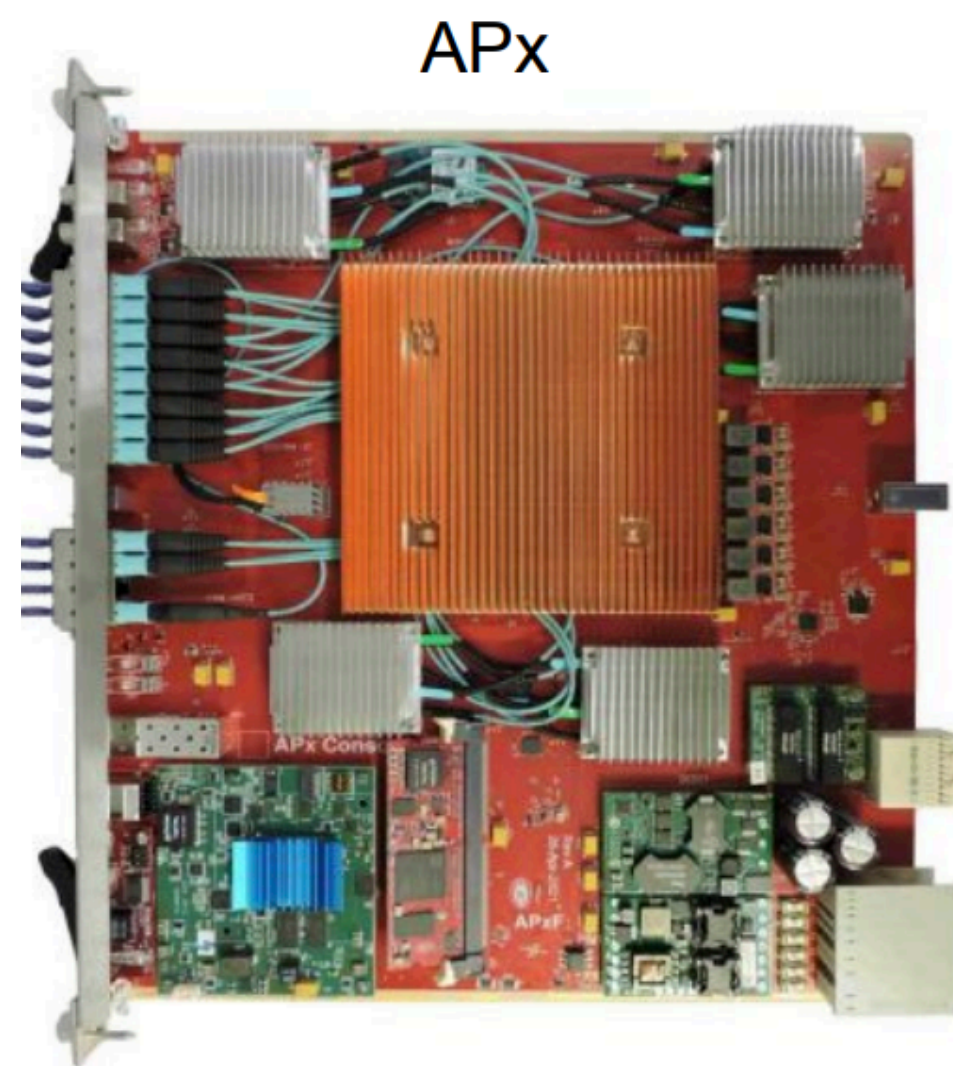
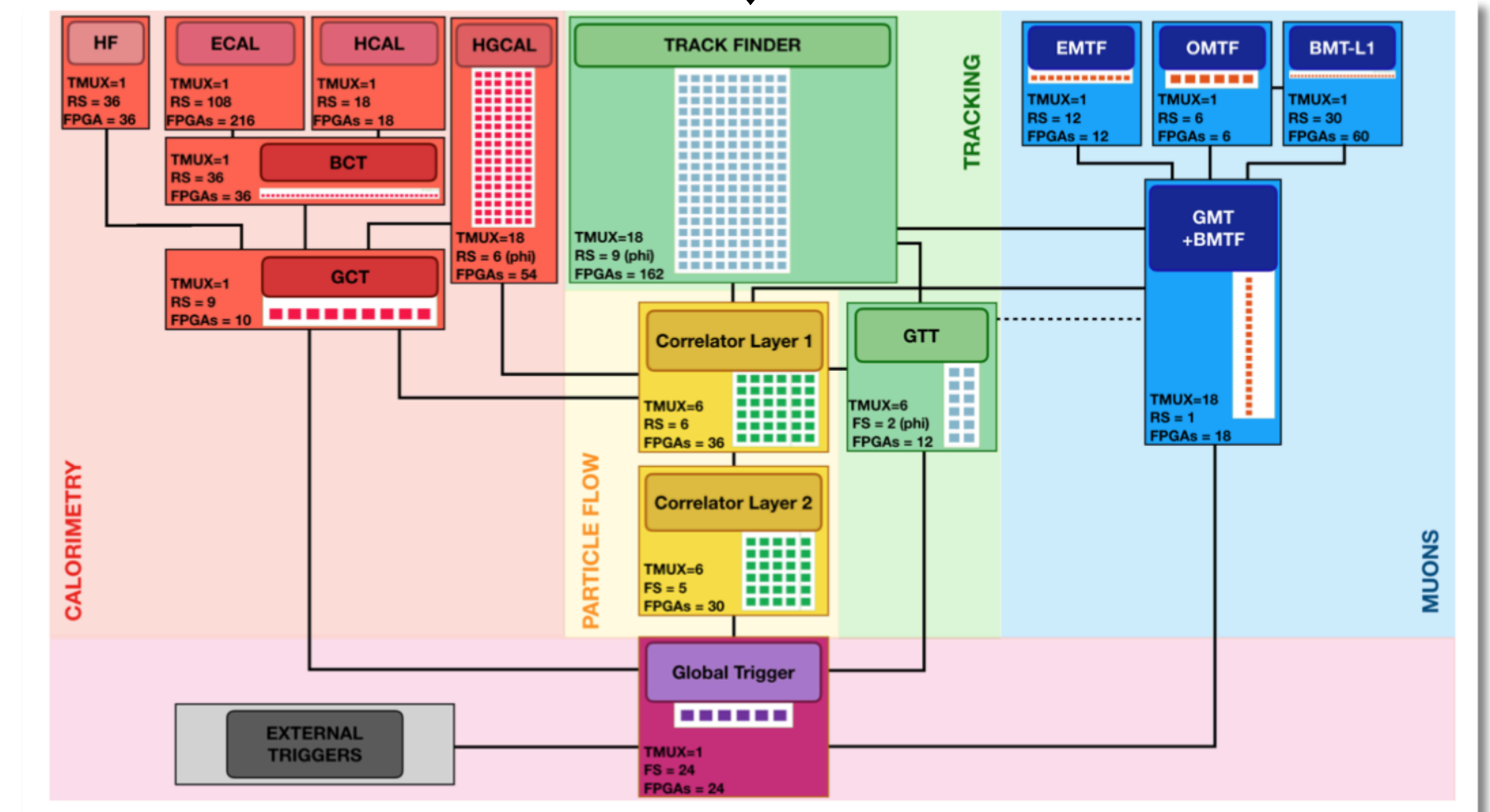


CMS-TDR-019
arXiv:1708.08234

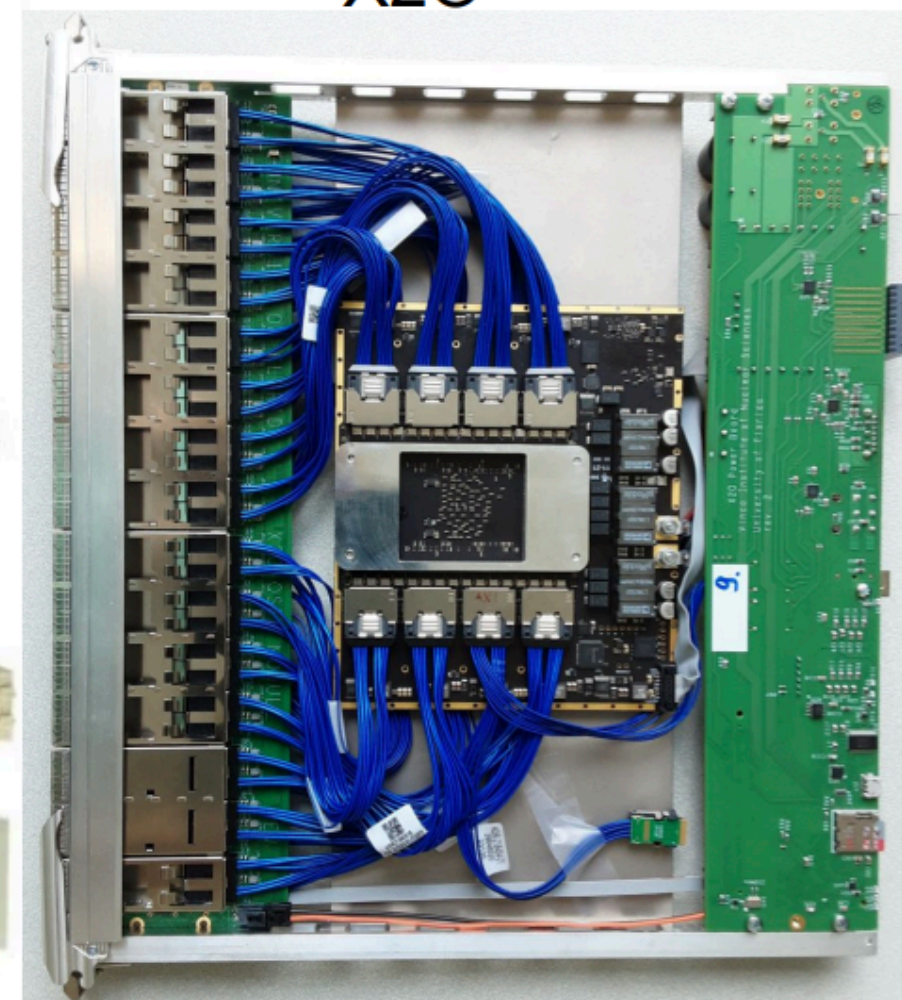
CMS Phase 2 Level 1 Trigger



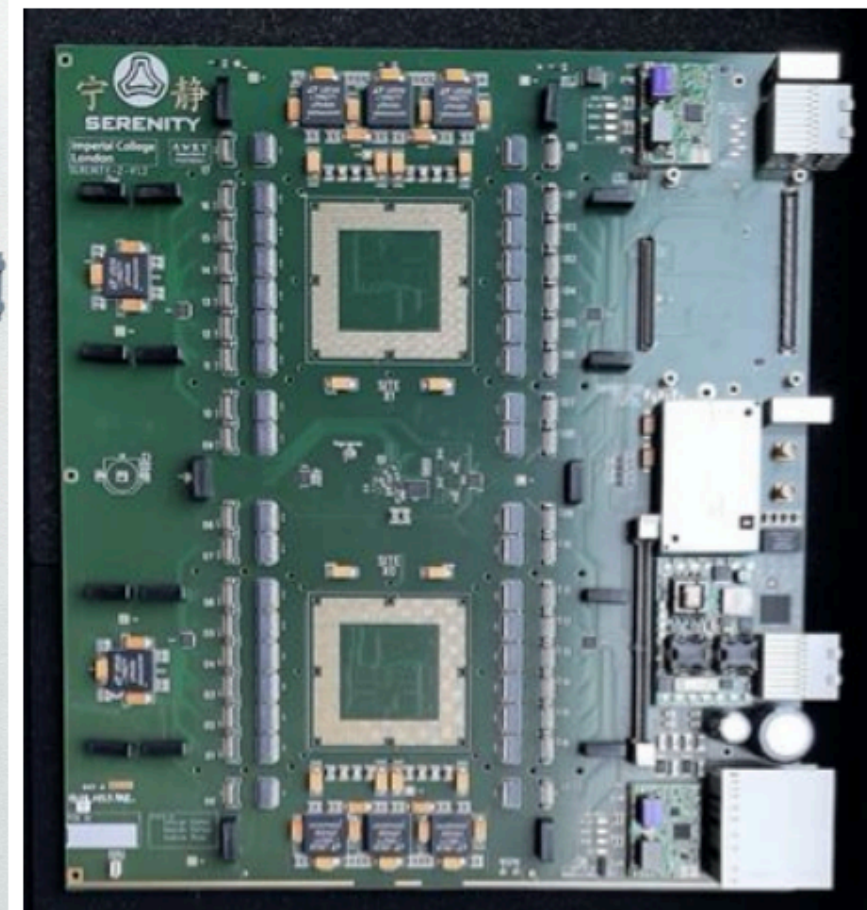
- Phase 2 Upgrade of CMS L1T will have hundreds of boards with FPGAs like those shown below - AMD/Xilinx Ultrascale+ FPGAs
- Data rate of multiple terabits per second into / out of each board on optical fibres
- System organised in layers with normally $\sim 1\text{-}2\ \mu\text{s}$ per step
 - Reducing raw detector data into physics objects (e.g. track finding: hits to tracks)
 - New event every 25 ns, latency for trigger decision for one event $12.5\ \mu\text{s}$
- Final output is one bit: keep or discard event



APx



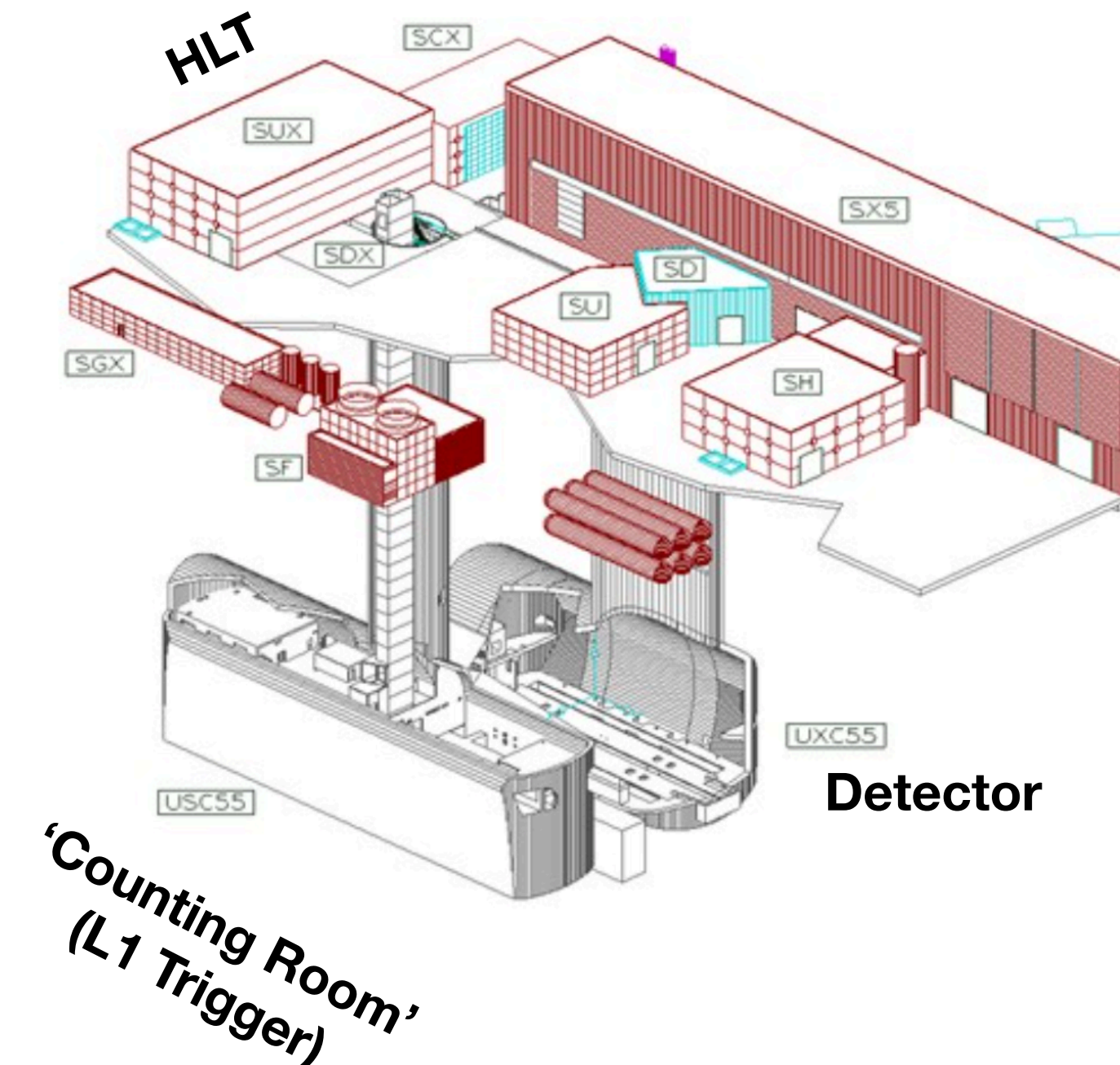
X20



Serenity

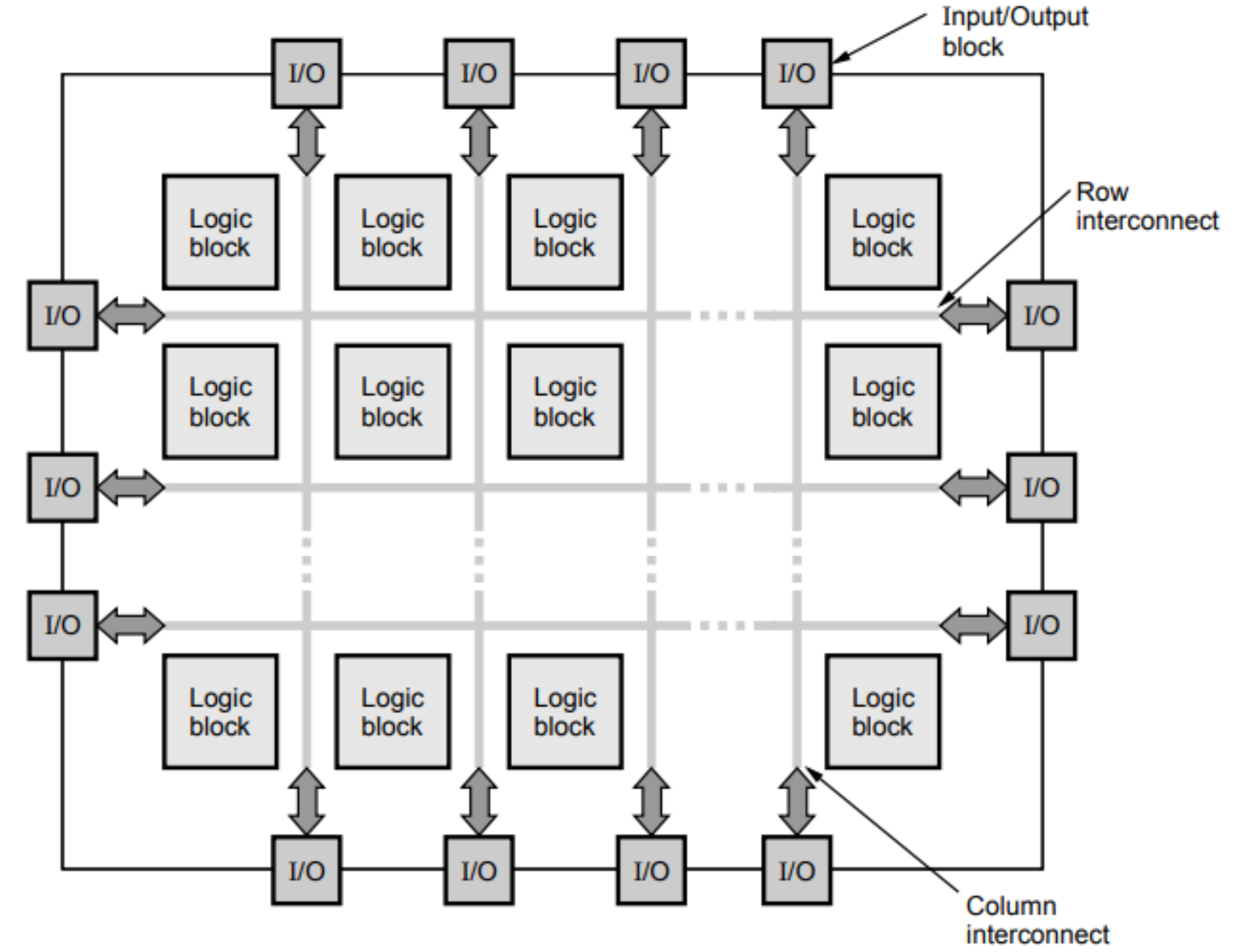


BMT



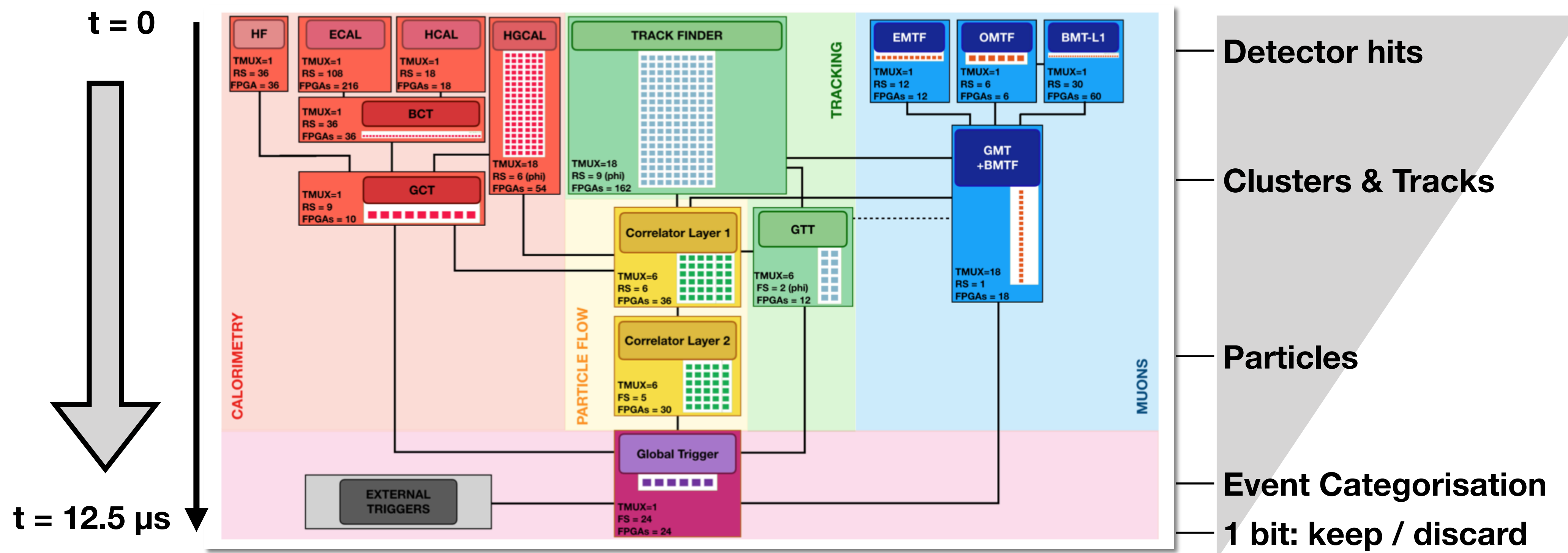
FPGAs

- What are FPGAs? = Field Programmable Gate Array
- Configurable electronic circuits
 - Can be used for wide variety of applications
 - Coding involves designing the *processor*, not a program
- We use them for:
 - Huge low-level compute parallelism
 - Huge input & output data rates
- Two types of parallelism: resource and pipeline
 - Resource parallelism enables us to do different tasks simultaneously to reach low latency
 - Pipeline parallelism enables us to do the same task on different data at high throughput
- In the automotive factory the many robots are resource parallelism and the conveyor belt is pipelining
- High performance requires use of both types

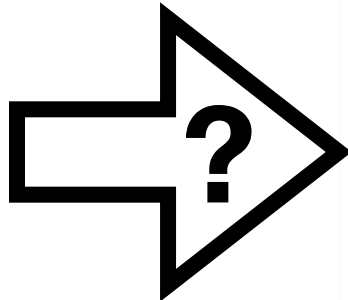
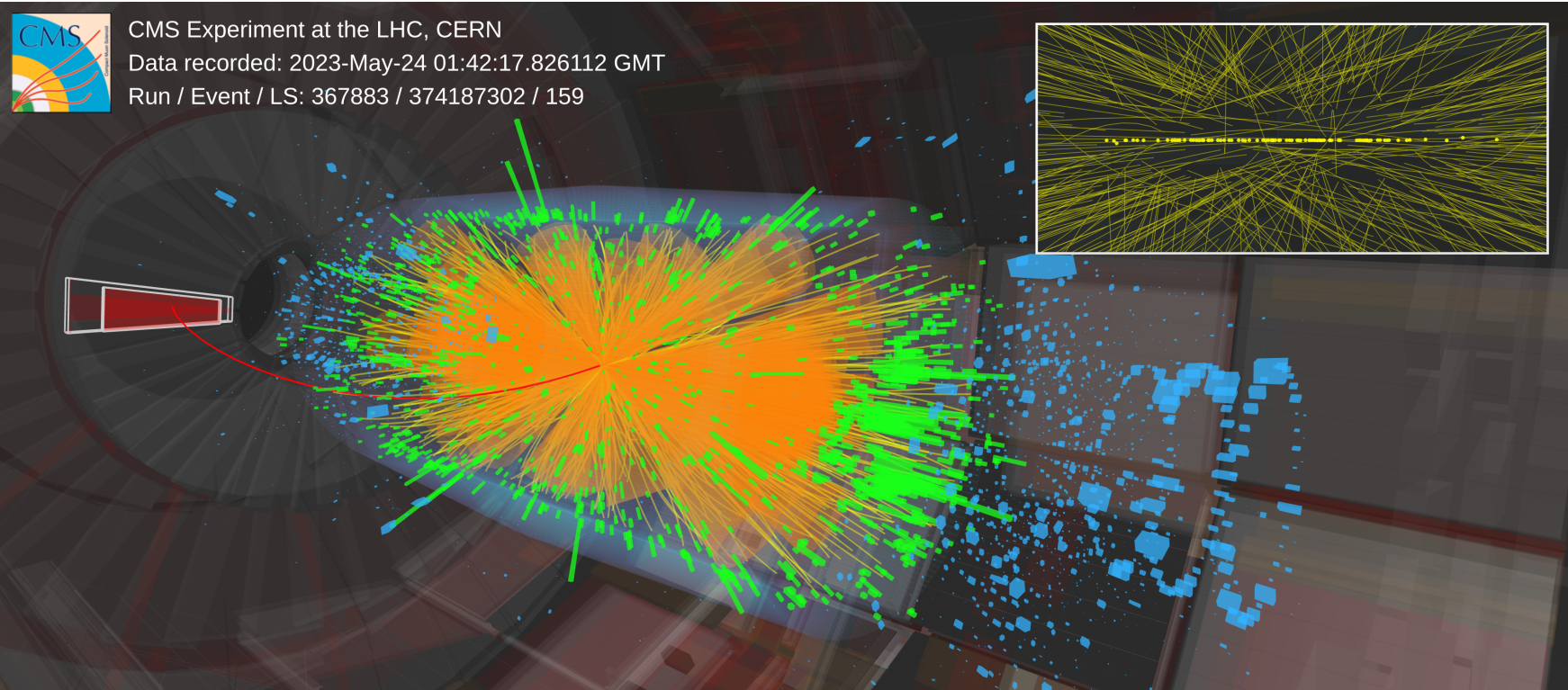


CMS Level 1 Trigger

- Level 1 Trigger decision requires reconstructing “high level” information from “low level” information
 - Low level: raw detector hits (digitised measurements from sensors)
 - High level: particle properties, event-level quantities like total energy, jets (sprays of particles)
- Final decision compares the high level quantities with a list of conditions to accept
- Processing mostly uses sophisticated physics algorithms for reconstruction



CMS Global Trigger — final decision



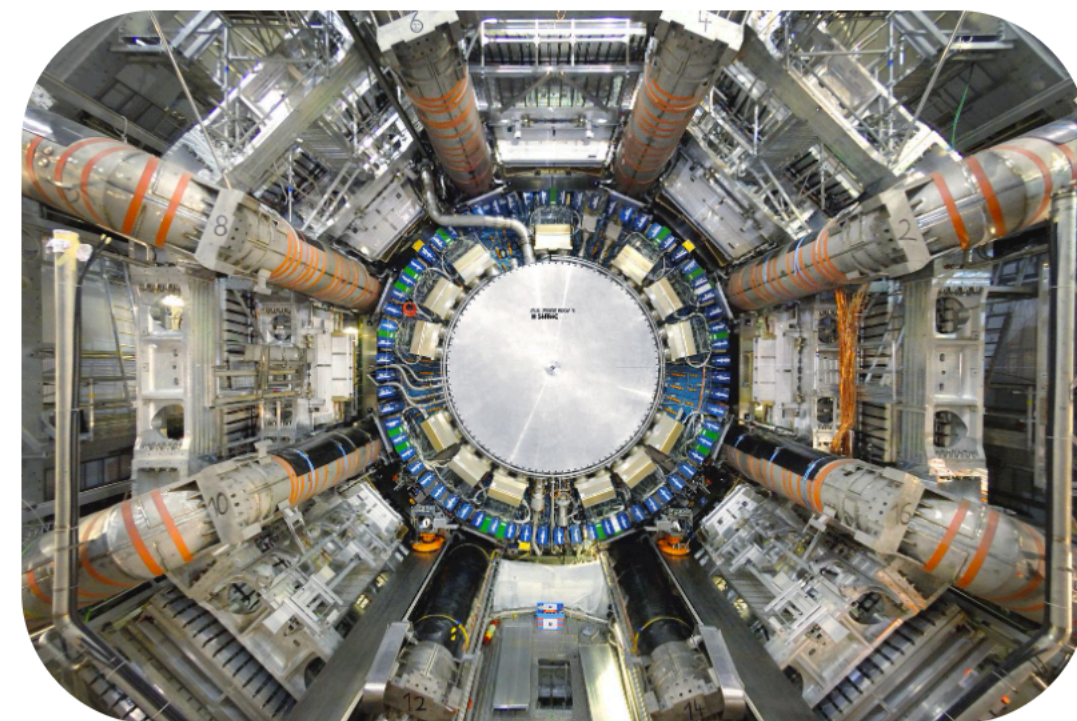
L1 Trigger seeds	Offline Threshold(s) at 90% or 95% (50%) [GeV]	Rate $\langle PU \rangle = 200$ [kHz]	Additional Requirement(s) [cm, GeV]	Objects plateau efficiency [%]
Single/Double/Triple Lepton (electron, muon) seeds				
Single TkMuon	22	12	$ \eta < 2.4$	95
Double TkMuon	15,7	1	$ \eta < 2.4, \Delta z < 1$	95
Triple TkMuon	5,3,3	16	$ \eta < 2.4, \Delta z < 1$	95
Single TkElectron	36	24	$ \eta < 2.4$	93

- Most of the trigger processing is dedicated to reconstruct particles from raw detector information
- The final decision is made by comparing the reconstructed particle information with a “menu” of 100s of conditions (“seeds”)
 - For example: is there a muon in the event with transverse momentum greater than 22 GeV? If there is: keep the event
 - Events passing at least one condition are read out and sent to the high level trigger
- Almost all trigger seeds have a minimum transverse momentum requirement: the main way to reject background

- Next Generation Triggers is an exciting five year project to get more information out of LHC collision data
 - Improving the processing that the trigger carries out with advanced computing and machine learning
- Participation including CERN Experimental Physics, IT and Theory departments, and CMS and ATLAS experiments
- Lots of hiring for Next Generation Triggers projects: <https://nextgentriggers.web.cern.ch/jobs/>
- Also plenty of events including training: <https://nextgentriggers.web.cern.ch/events/>



WP1: Infrastructure, Algorithms and Theory



WP2: Enhancing the ATLAS Trigger and Data Acquisition



WP3: Rethinking the CMS Real-Time Data Processing

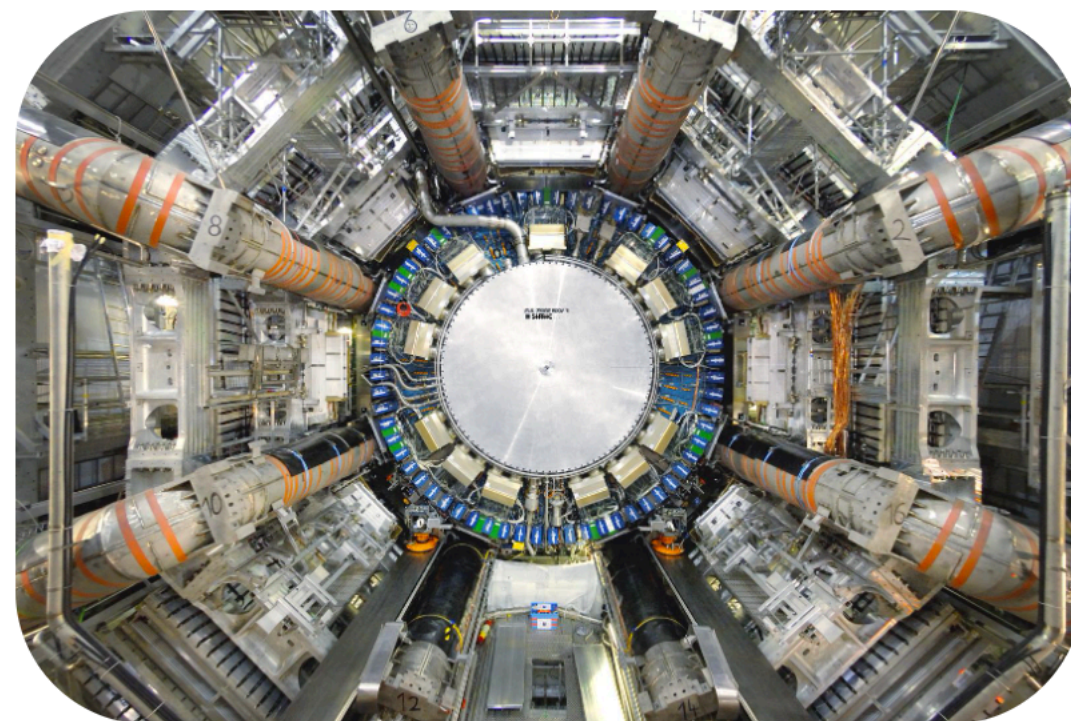


WP4: Education Programmes and Outreach

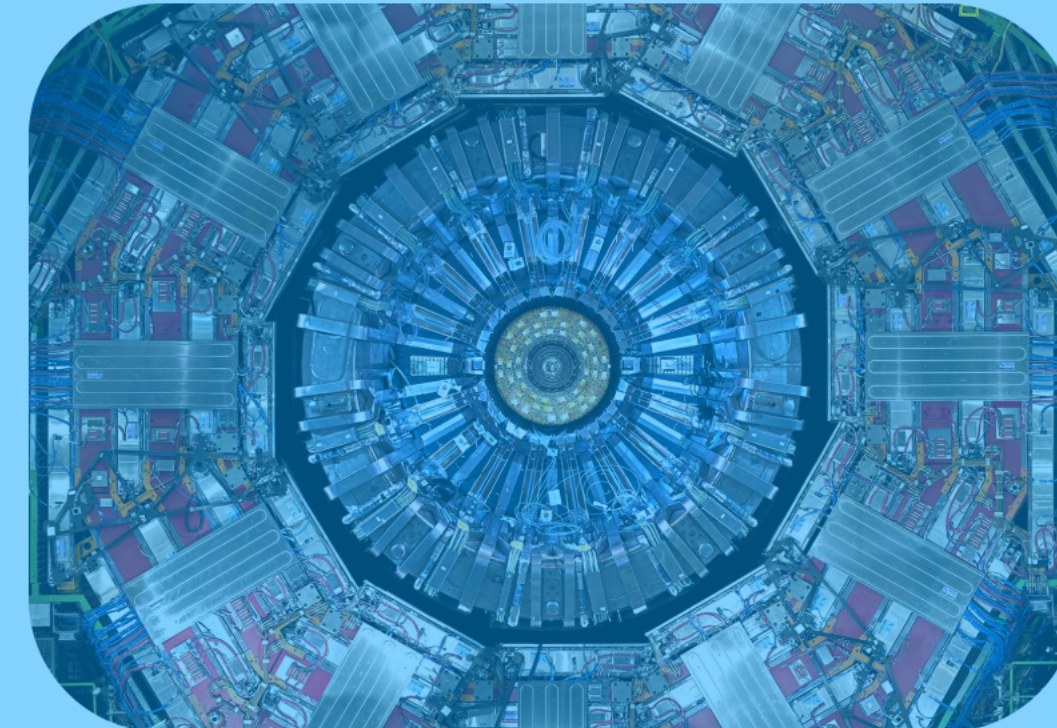
- Next Generation Triggers is an exciting five year project to get more information out of LHC collision data
 - Improving the processing that the trigger carries out with advanced computing and machine learning
- Participation including CERN Experimental Physics, IT and Theory departments, and CMS and ATLAS experiments
- Lots of hiring for Next Generation Triggers projects: <https://nextgentriggers.web.cern.ch/jobs/>
- Also plenty of events including training: <https://nextgentriggers.web.cern.ch/events/>



**WP1: Infrastructure, Algorithms
and Theory**



**WP2: Enhancing the ATLAS Trigger
and Data Acquisition**



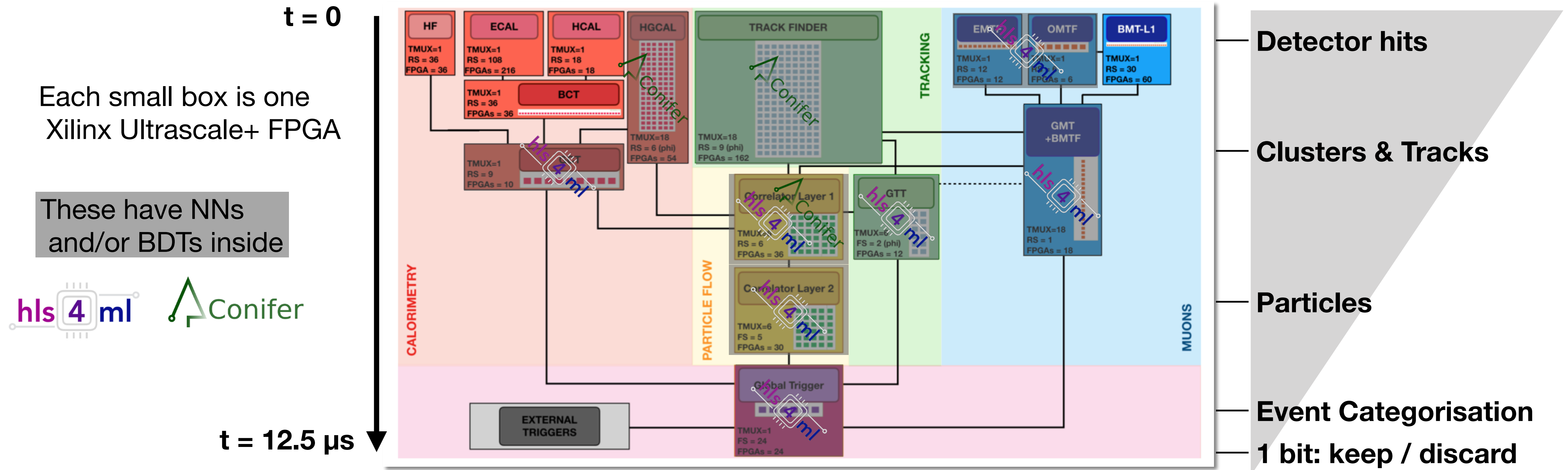
**WP3: Rethinking the CMS
Real-Time Data Processing**



**WP4: Education Programmes
and Outreach**

Machine Learning at the CMS Level 1 Trigger

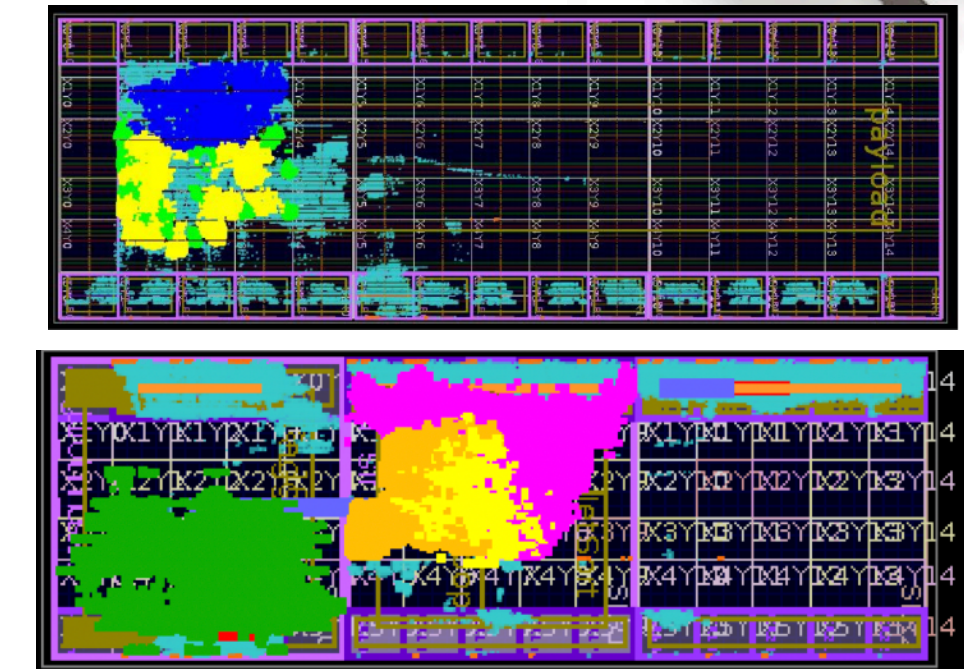
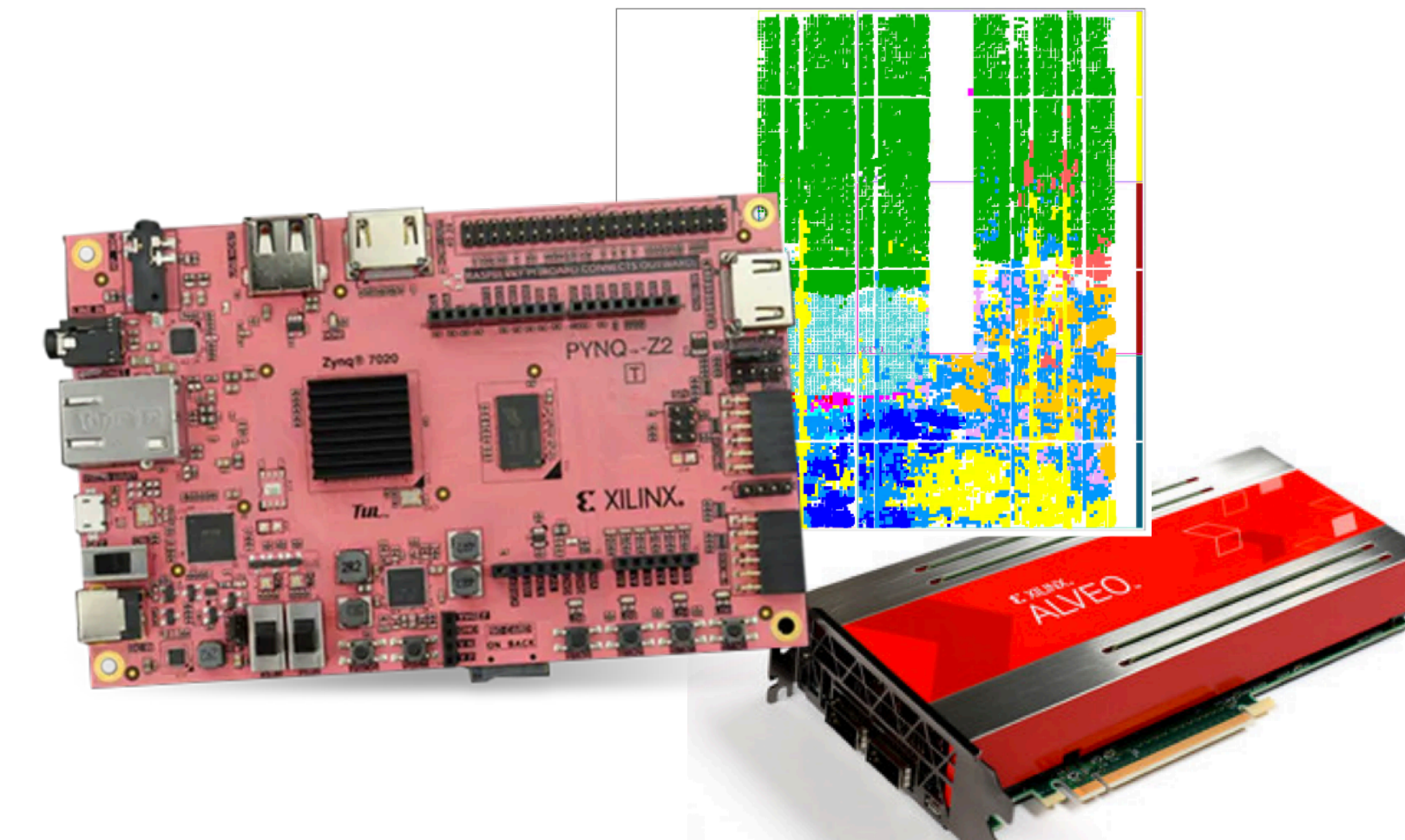
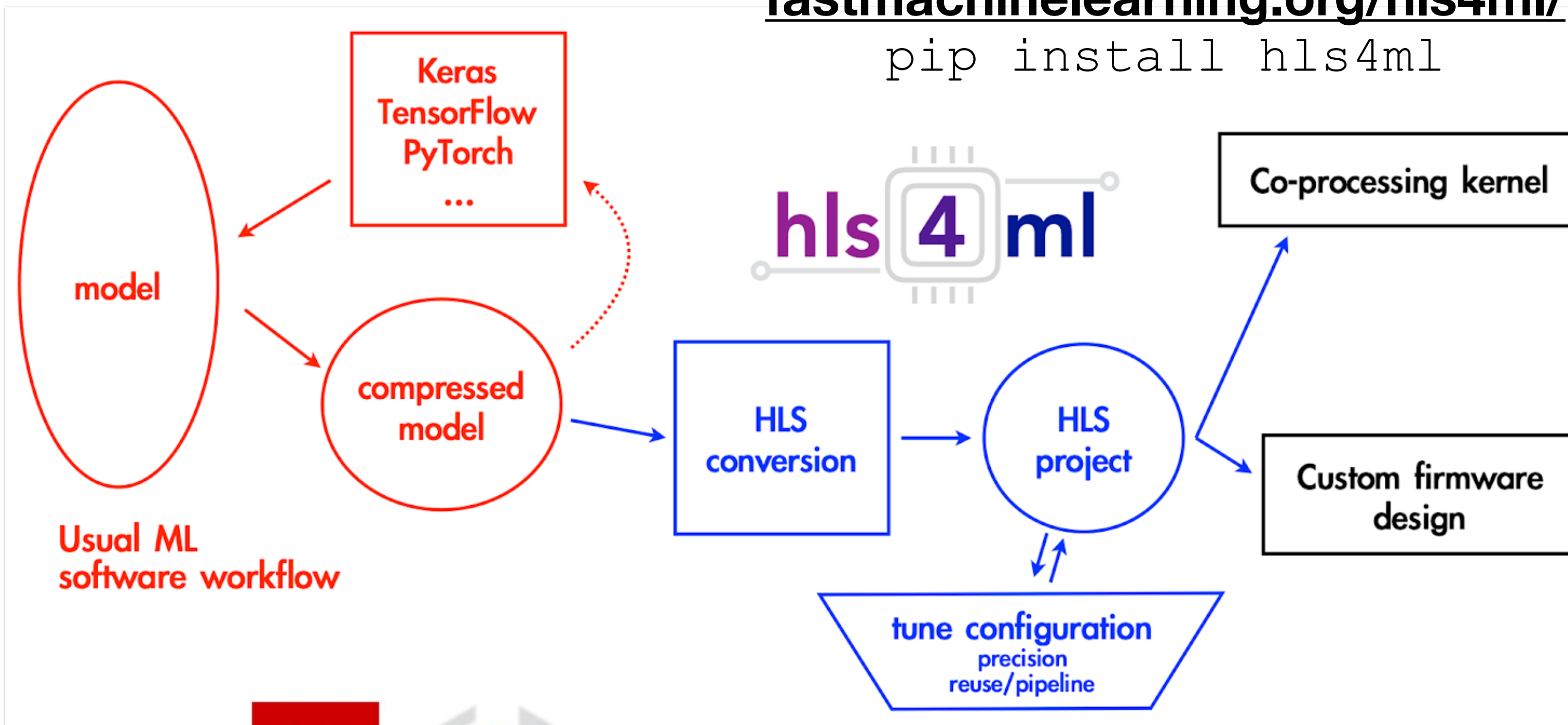
- The trigger is a binary classifier: keep / reject - can we use Machine Learning to do this task better?
- We use ML to trigger anomalies today in Run 3 (more later!)
- Machine Learning will be used throughout the Phase 2 System
 - Conservatively estimate **25 billion ML inferences per second**
- Next I will describe how we can put Machine Learning into our custom FPGA boards



hls4ml high level synthesis for machine learning

fastmachinelearning.org/hls4ml/

```
pip install hls4ml
```



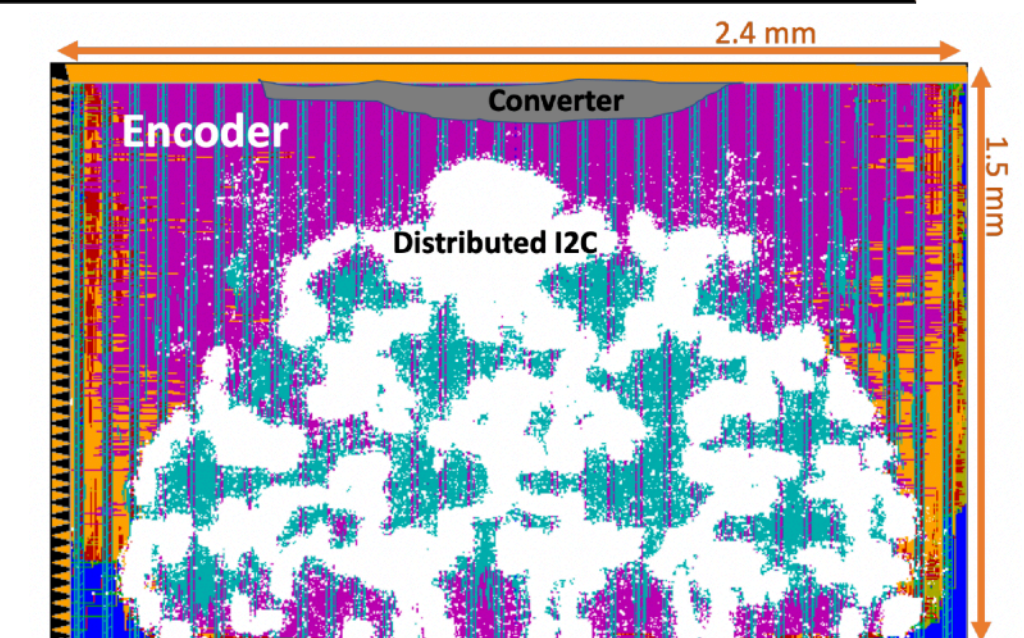
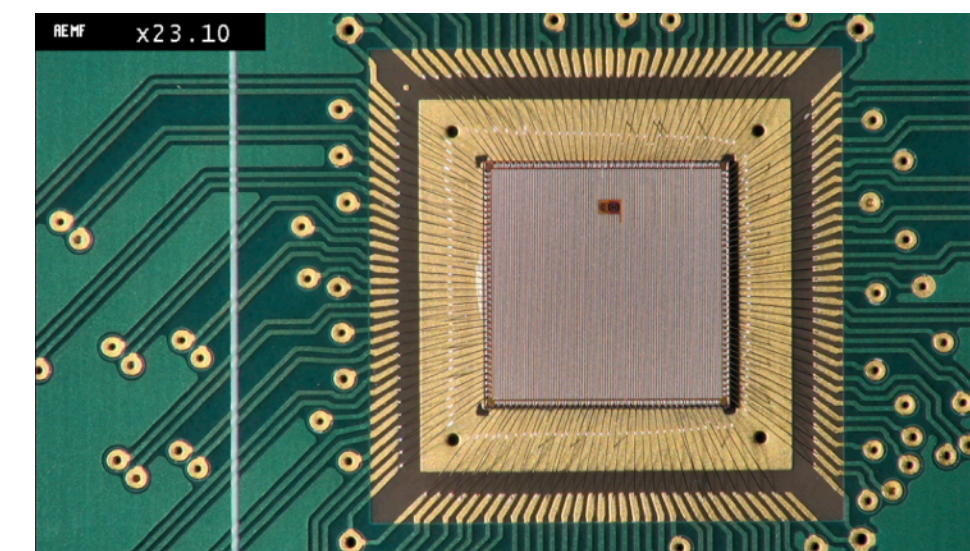
(Q)  + 

(Q)  ONNX
PYTORCH



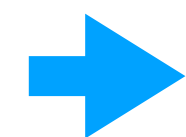
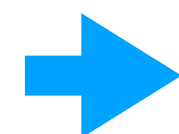
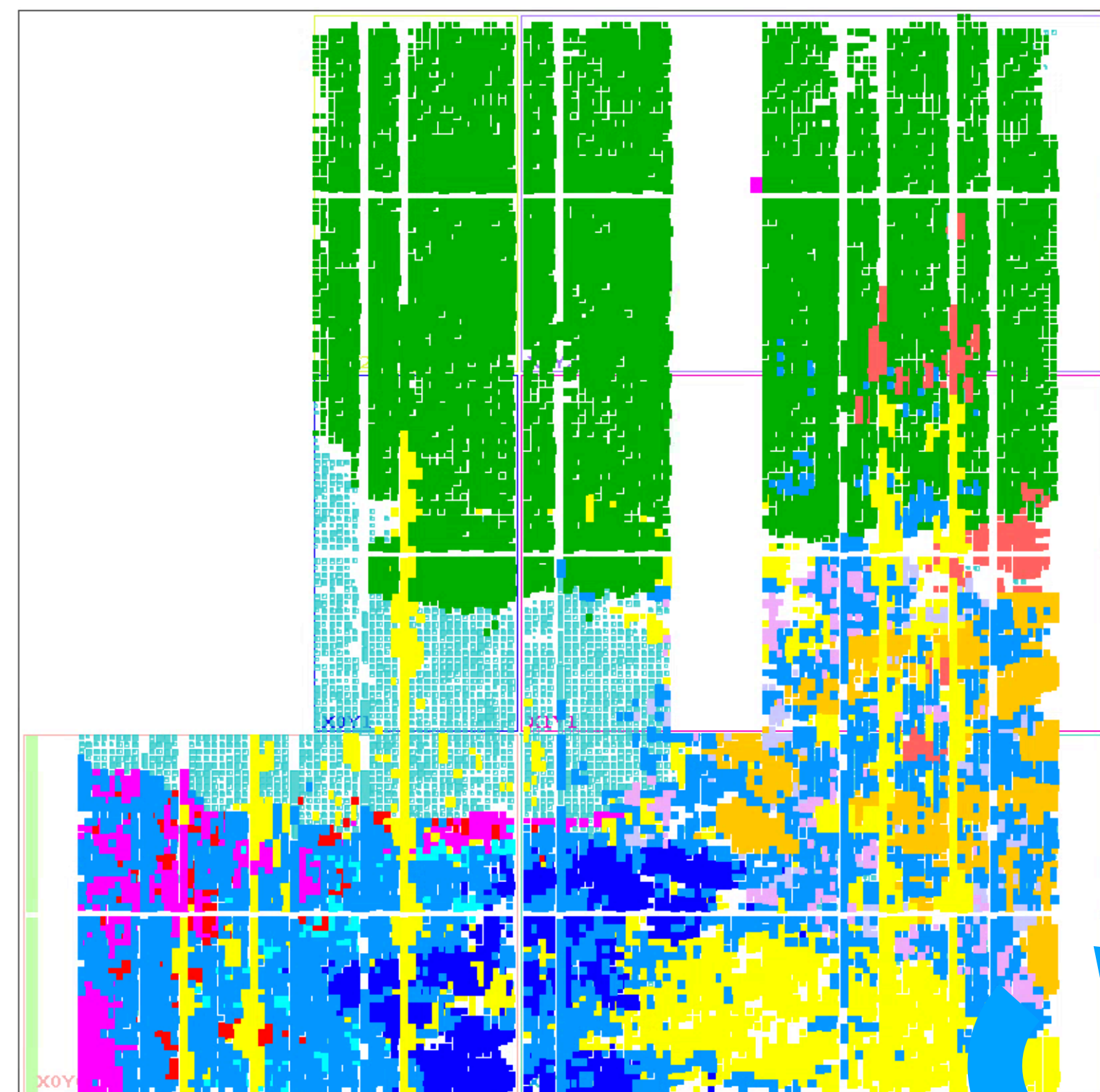
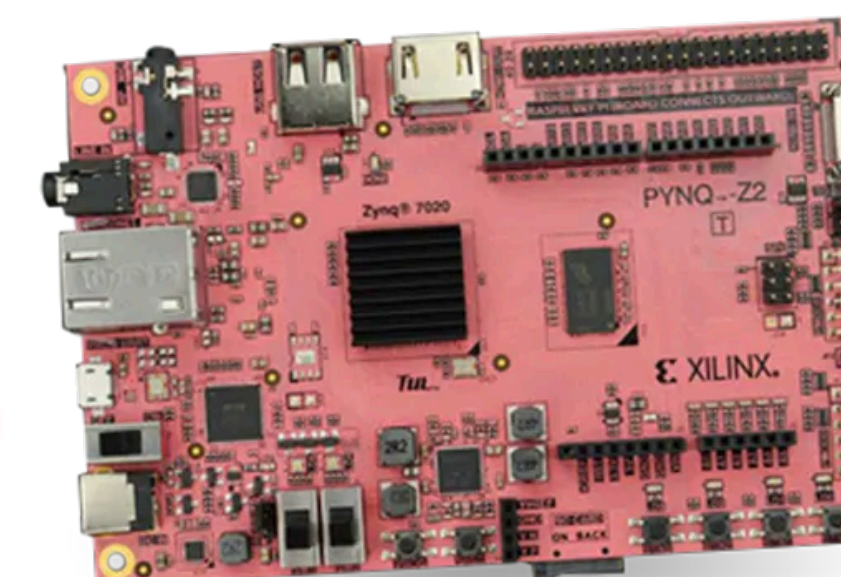
intel
1
oneAPI

Catapult AI NN
SIEMENS



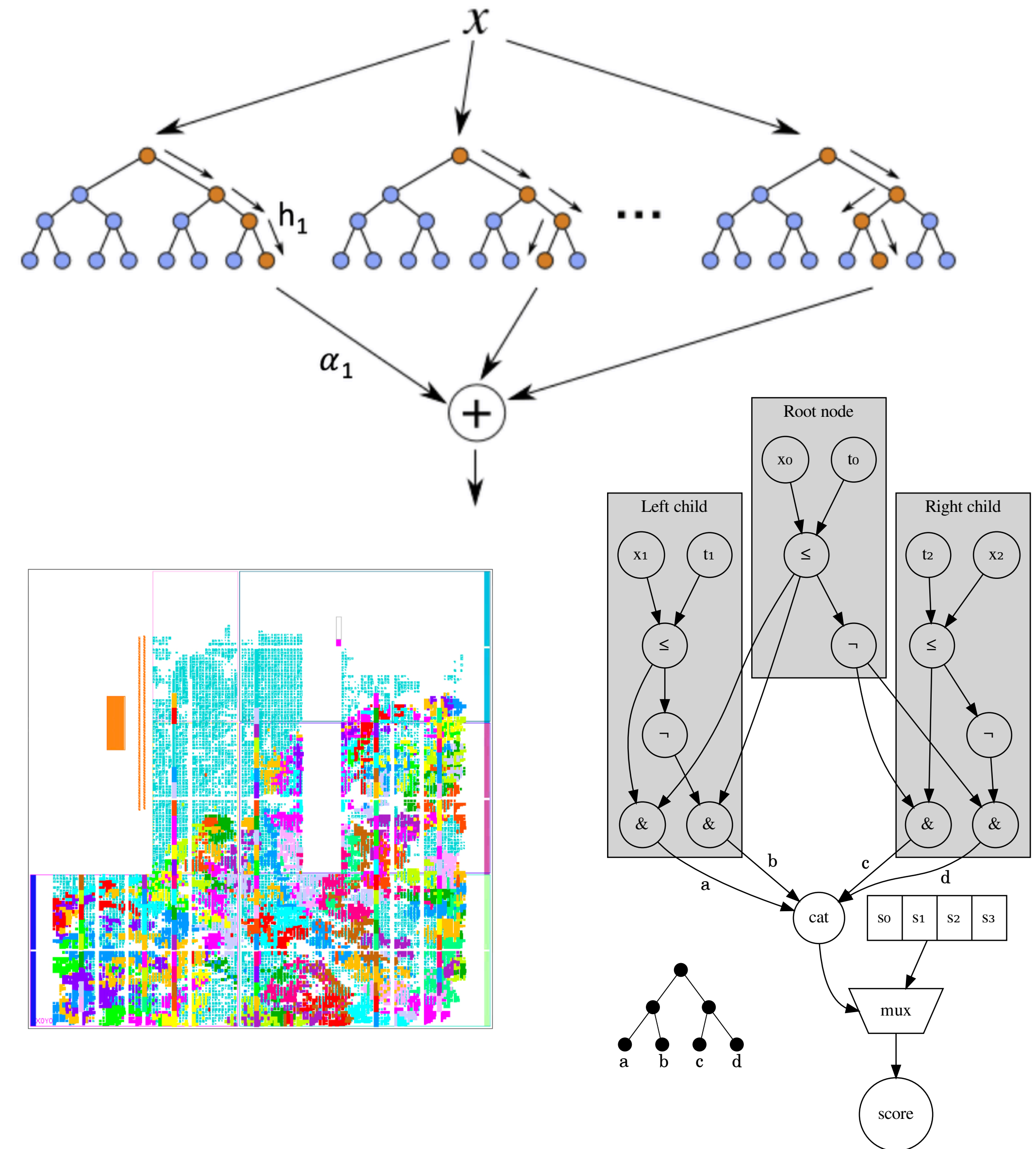
hls4ml - Dataflow Architecture

- Dataflow architecture: each layer is an independent compute unit
 - With tunable parallelism and quantization
- Fully on-chip: NN must fit within available FPGA resources (pynq-z2 floorplan shown)
 - Example: small CNN trained on MNIST

**Conv2D****ReLU****MaxPool2D****Conv2D****ReLU****MaxPool2D****Flatten****Dense****Softmax****Prediction****FIFOs**

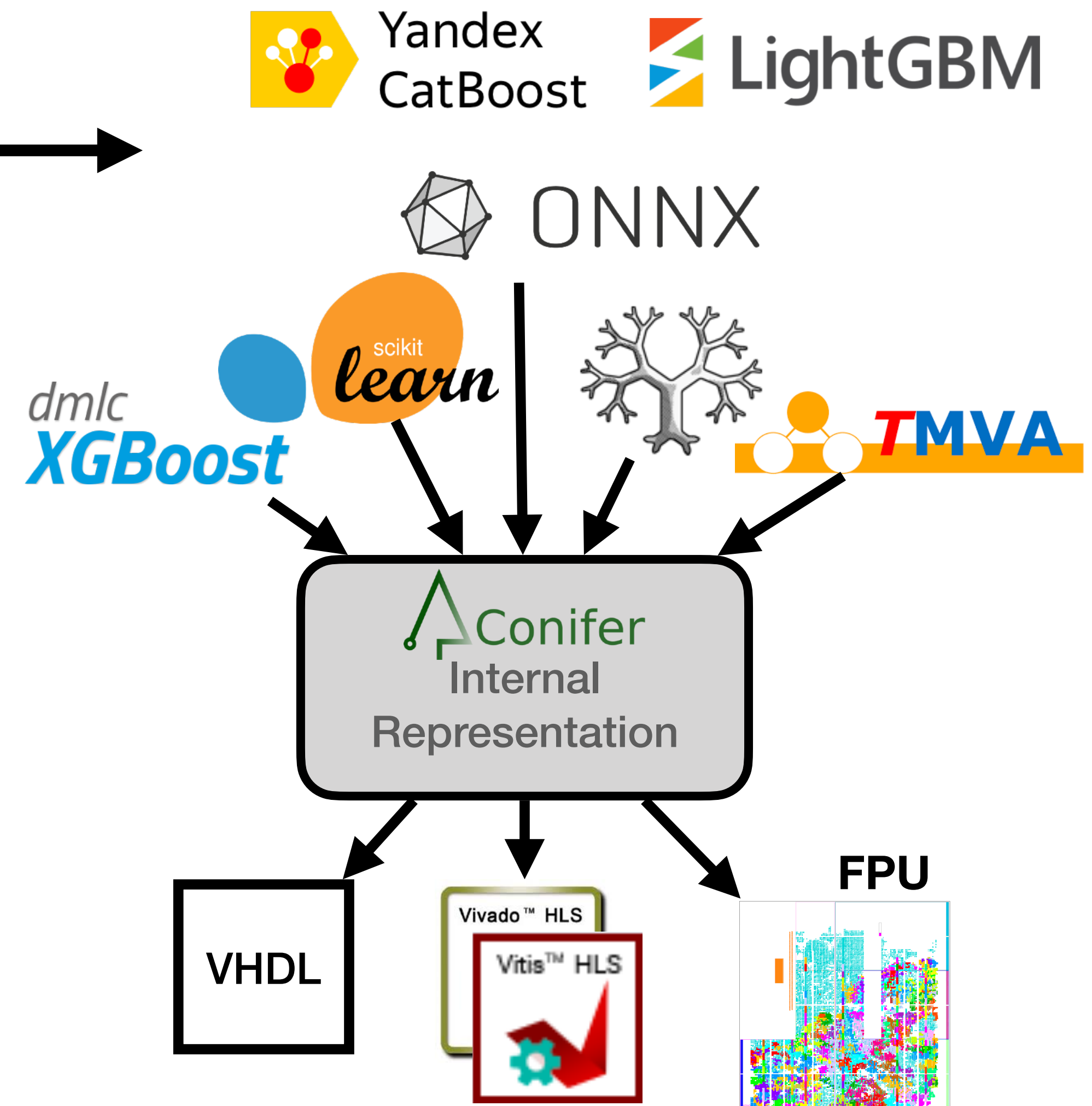
conifer for Decision Forests

- Neural Networks like Transformers for Large Language Models dominate the ML discourse
- But the old ways are still relevant: Decision Forests (“MVAs”)
 - Fast, lightweight, robust ([arXiv:2207.08815](https://arxiv.org/abs/2207.08815), [IML keynote](#))
- **conifer** is to DFs as hls4ml is to NNs
- A Decision Tree *splits* on data variables until reaching a *leaf*
 - Leaves associate a score corresponding to prediction probability
- A Decision Forest is an ensemble of Decision Trees
 - Randomisation of each DT as a form of regularisation
 - Ensemble score is an aggregation over trees e.g. sum
- **conifer** maps DFs onto FPGA logic
 - Implemented with high parallelism for low latency and high throughput



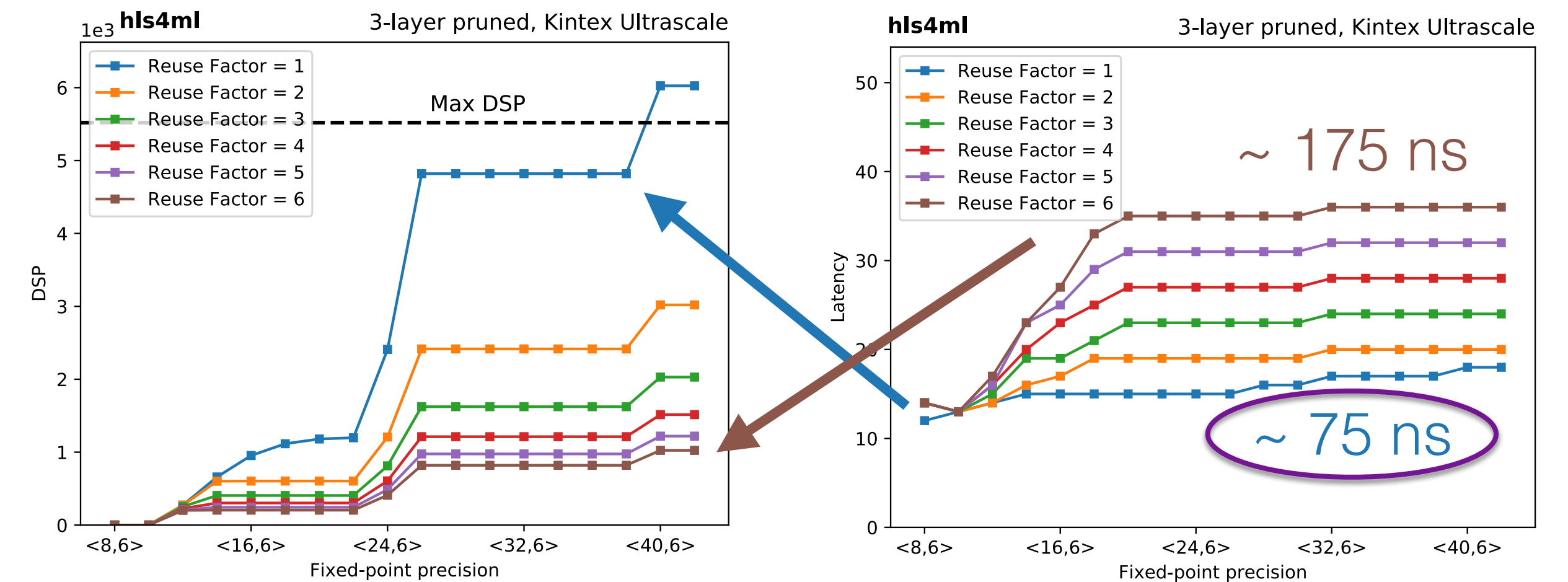
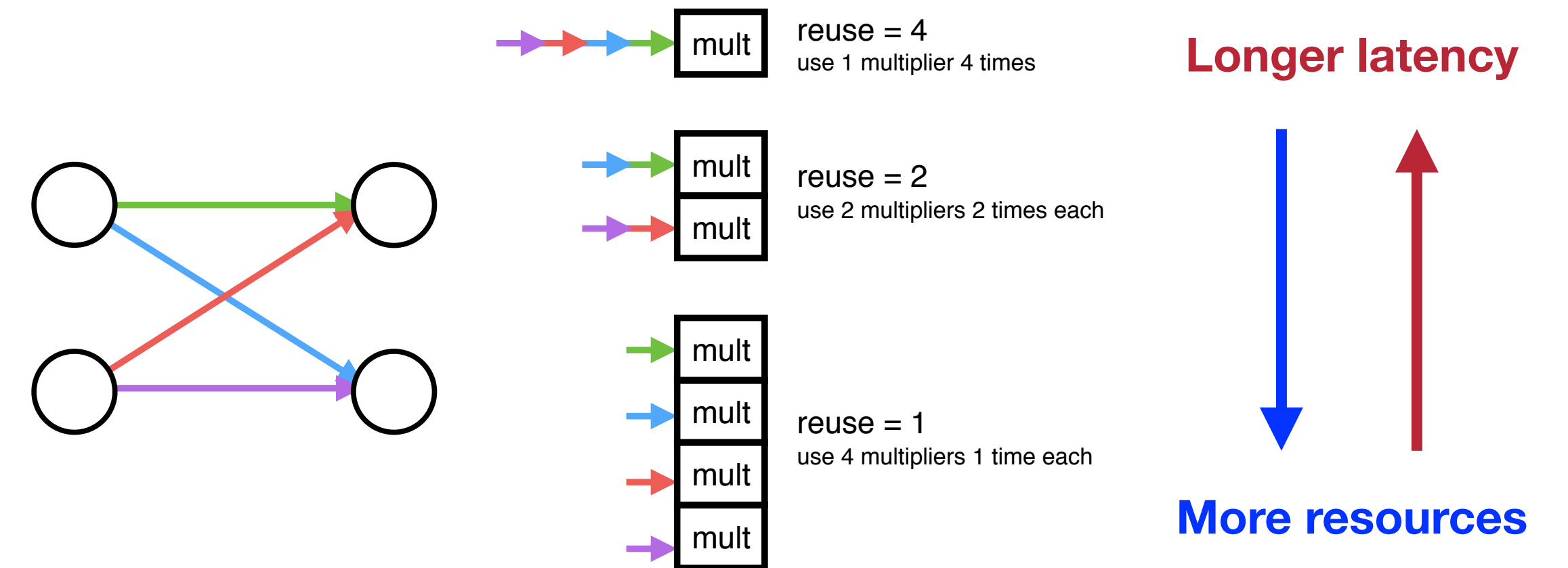
conifer implementations

- Very much like **hls4ml**, **conifer** has frontends, an Internal Representation, and backends
- Frontend support for popular BDT training libraries →
- Backends: HLS, (hand-written) VHDL, Forest Processing Unit (FPU)
- HLS and VHDL backends map one DF to one hardware implementation
 - Capable of inference at O(10) ns latency, O(100) MHz throughput
- FPU is a reconfigurable module that new models can be loaded onto
 - Binaries for some AMD devices [for download](#)
 - Implemented with HLS
- [GitHub](#), [website](#), [paper](#), `pip install conifer`



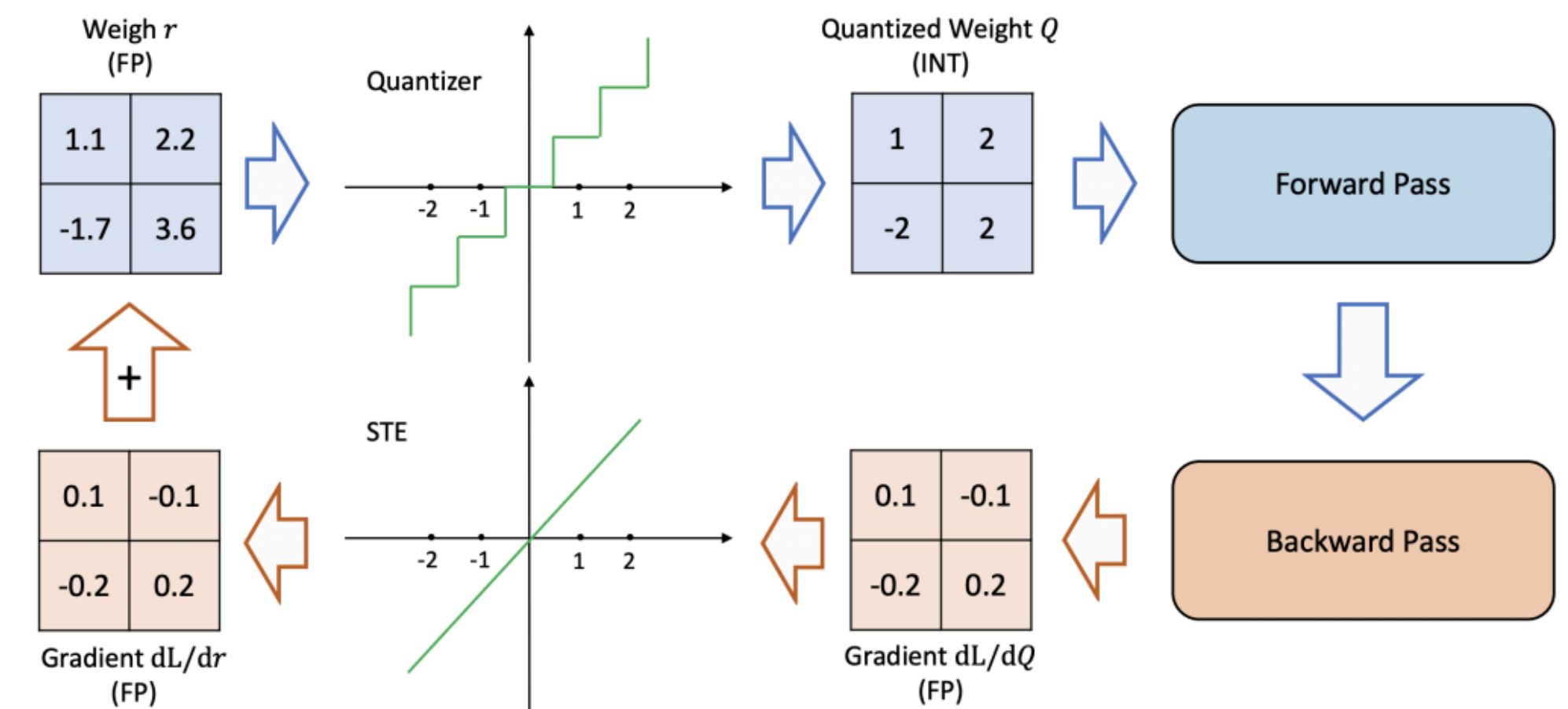
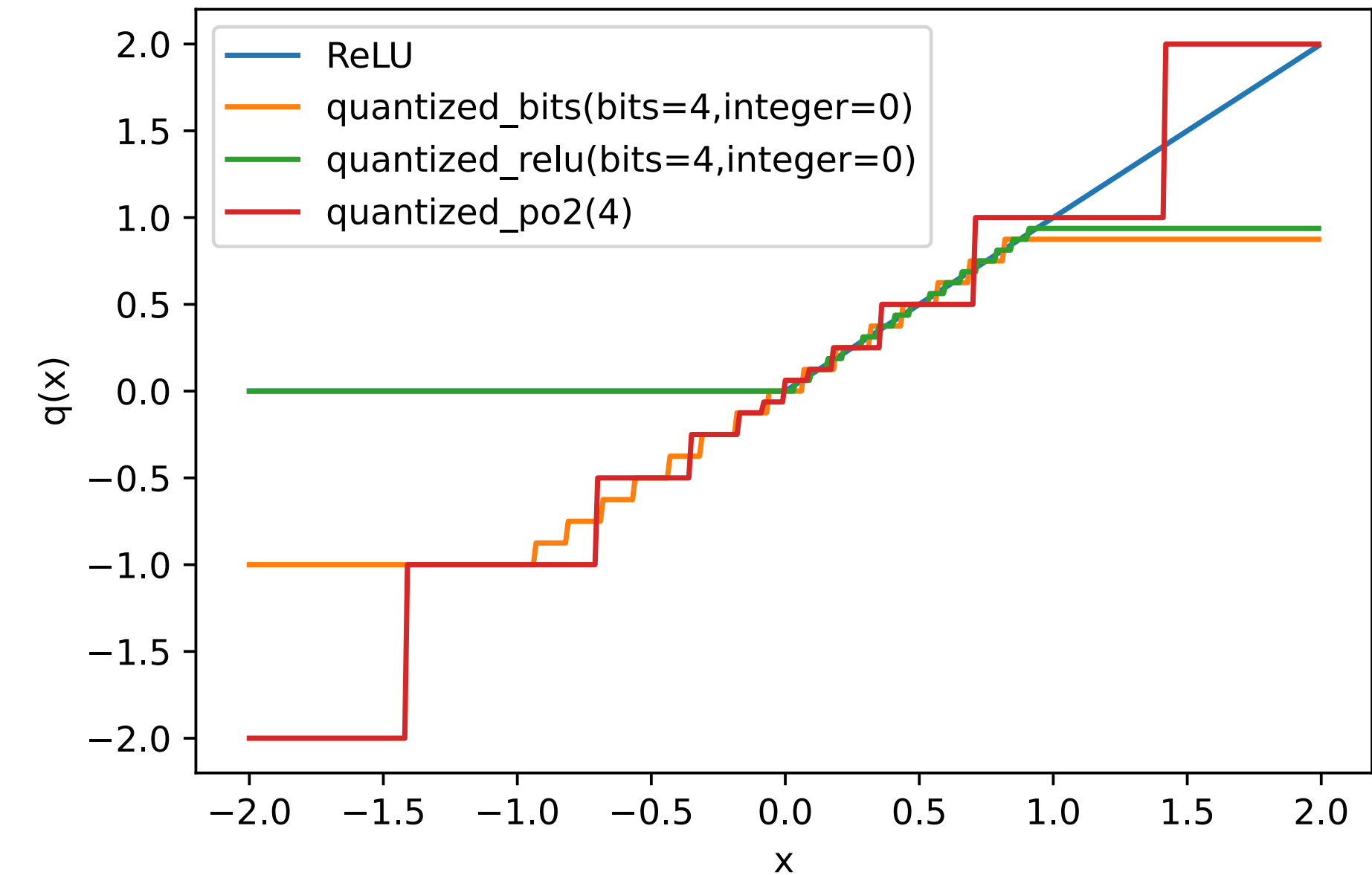
hls4ml and conifer key features

- Easy to use
 - Reduce the barrier to entry for hardware non-experts
 - Python packages with nice interfaces to EDA tools
 - `pip install hls4ml conifer`
 - Configuration interface for fine-grained control
 - [Tutorial](#) and documentation for getting started
- High Level Synthesis implementations (C++)
 - More accessible, and powerful Design Space Exploration
- Open source software, open communities
 - fastmachinelearning.org
- Massively parallel for low latency and high throughput
 - ‘Unrolled’ implementations
- Common interfaces



Efficient Training: Quantization

- Possibly the main technique for making NNs cheaper in FPGAs!
- Using regular TensorFlow Keras or PyTorch, you typically train with floating point
 - We like to avoid *floating point* in edge hardware - much more costly in resources & latency than *fixed point*
 - You can do *Post-Training Quantisation* (PTQ): represent the float values with some fixed point
- With *Quantization Aware Training* (QAT), you constrain weights/biases/activations to fewer values during training
 - Superior to PTQ for lower bitwidths - can go all the way down to 1 bit (representing ± 1)
 - Use quantizations with efficient hardware operators: integer, fixed point, power of 2
 - Use 'Straight Through Estimator' for back propagation step
 - With a fully unrolled hardware implementation we can learn different bitwidths for different parts of the network: HGQ

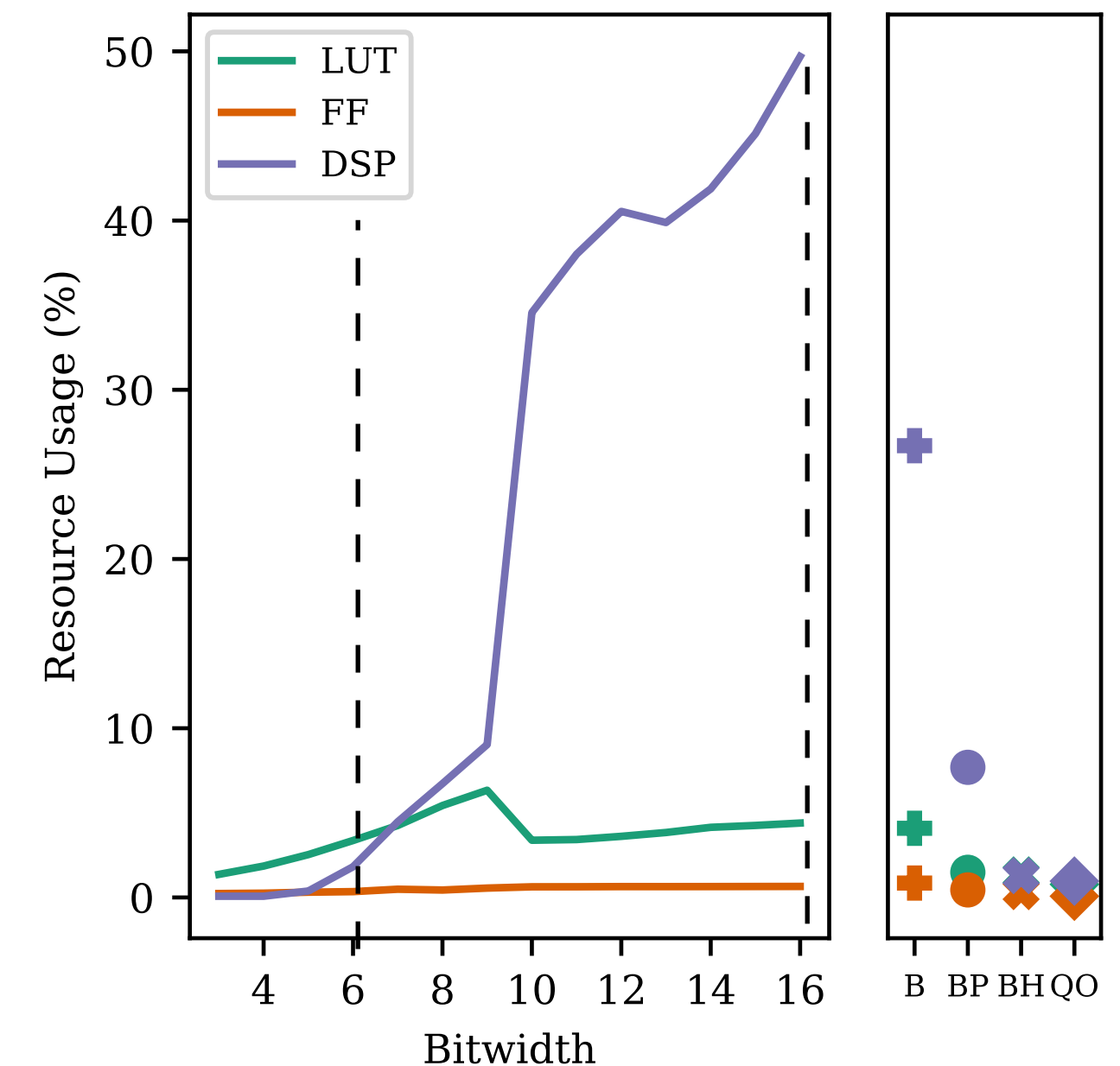
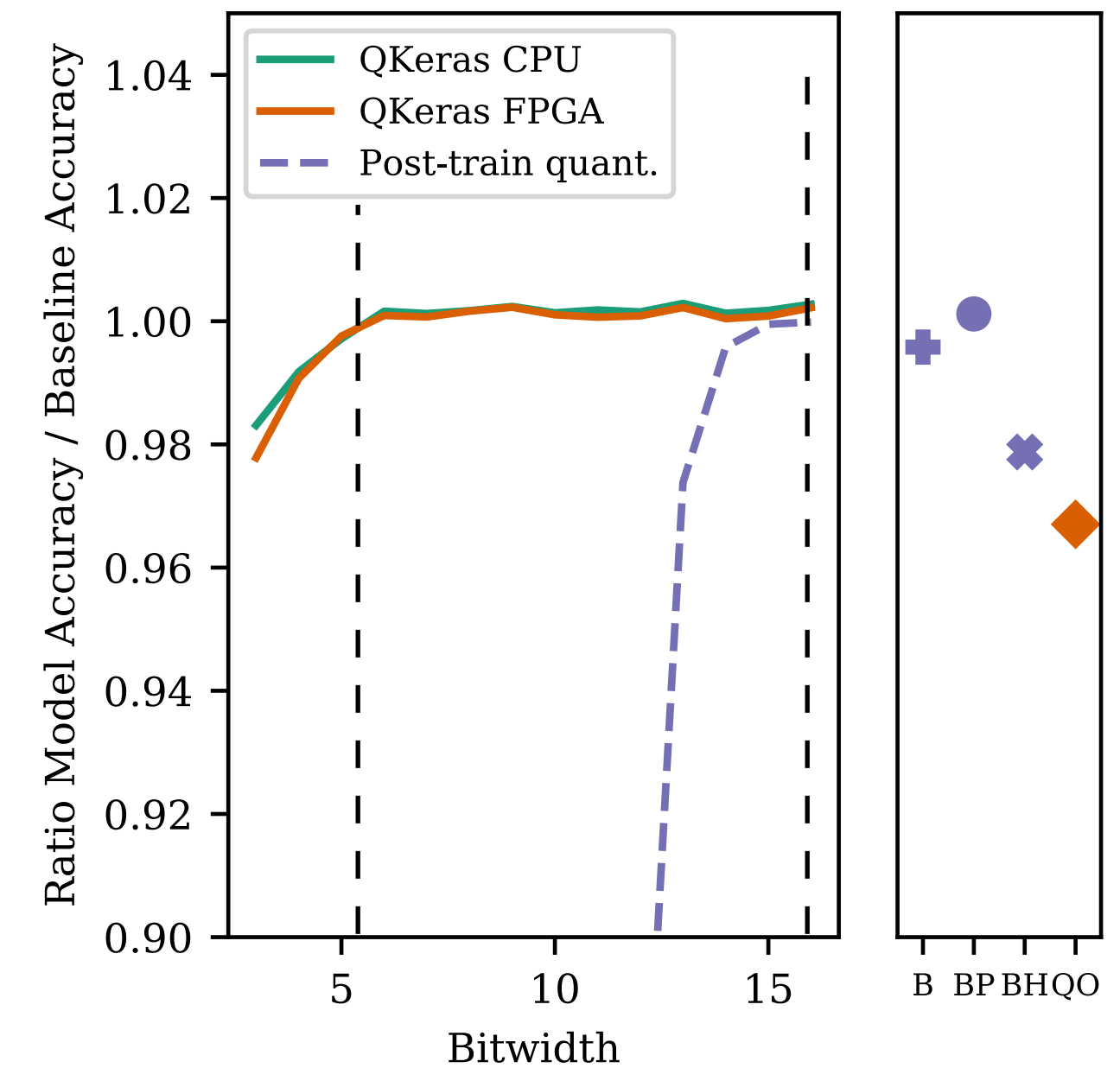


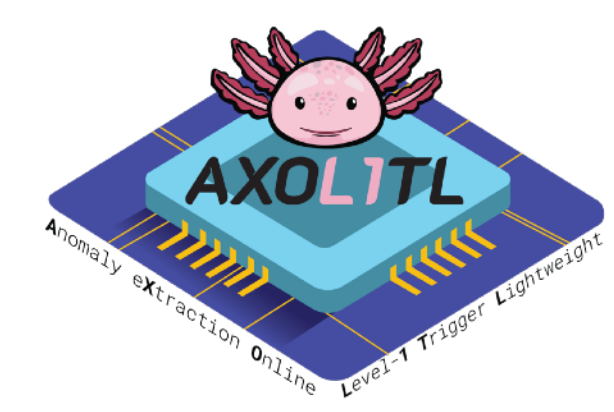
[arXiv:2103.13630](https://arxiv.org/abs/2103.13630)

QKeras

doi: 10.1038/s42256-021-00356-5

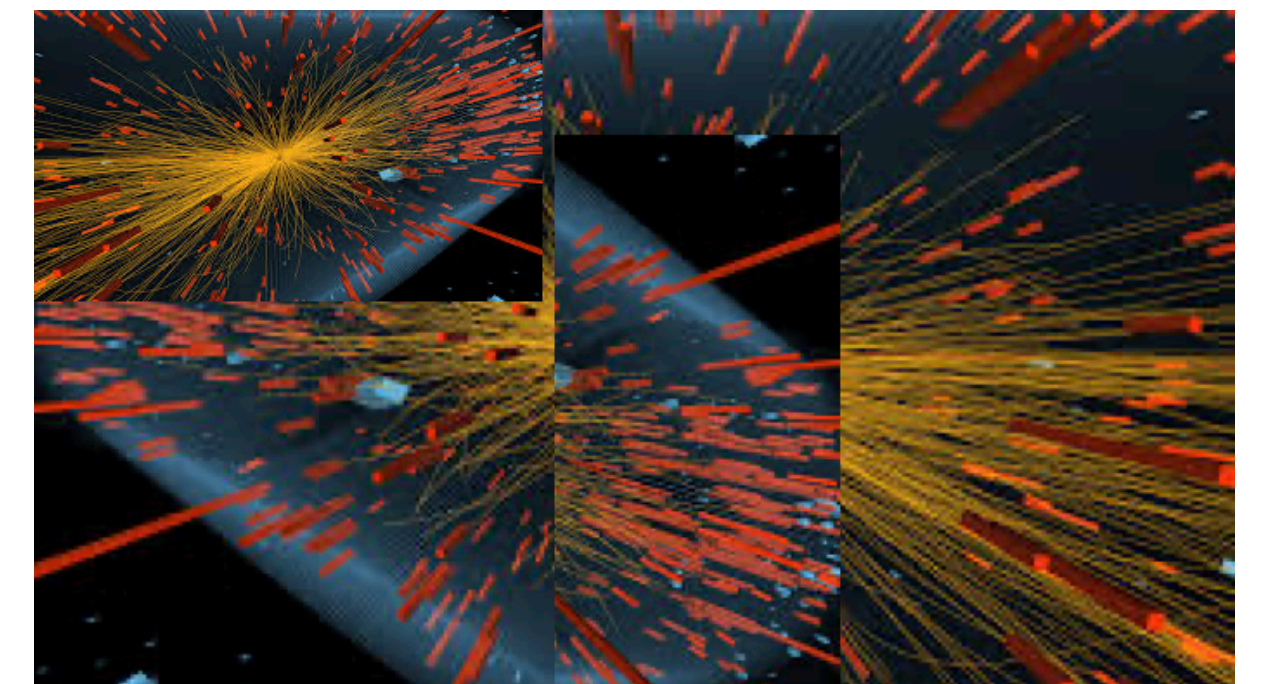
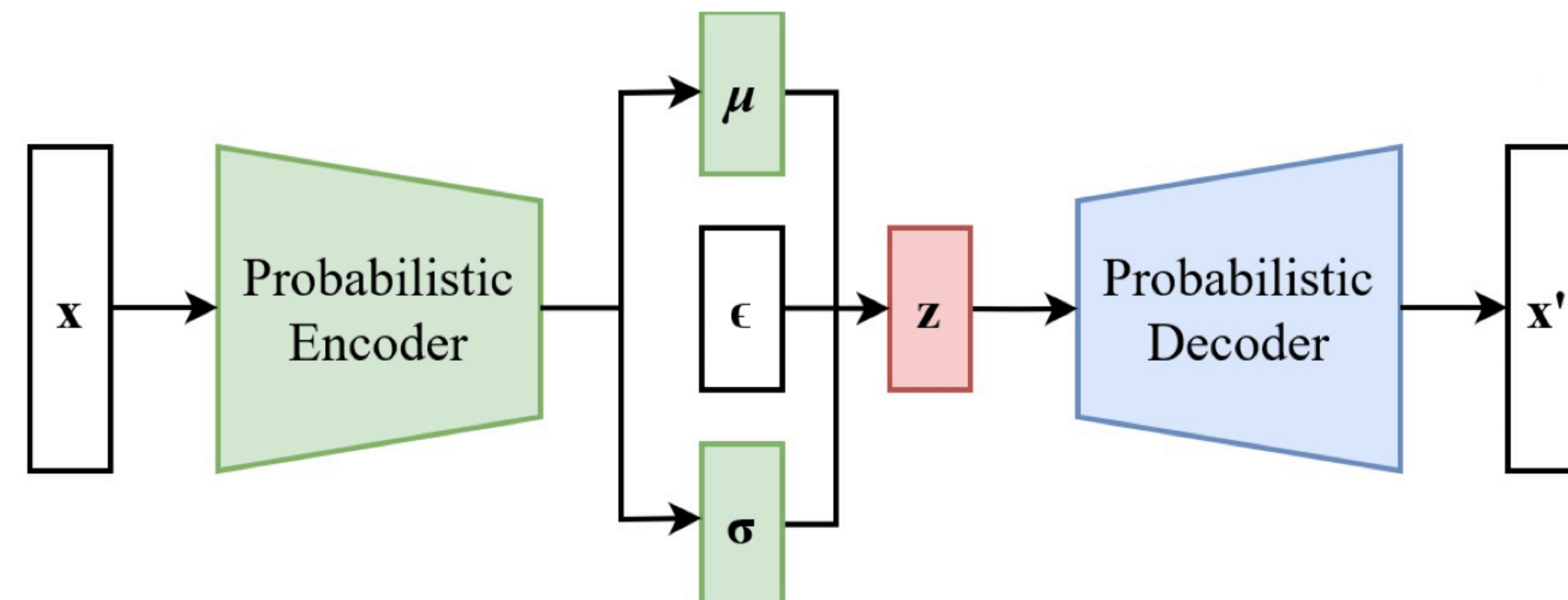
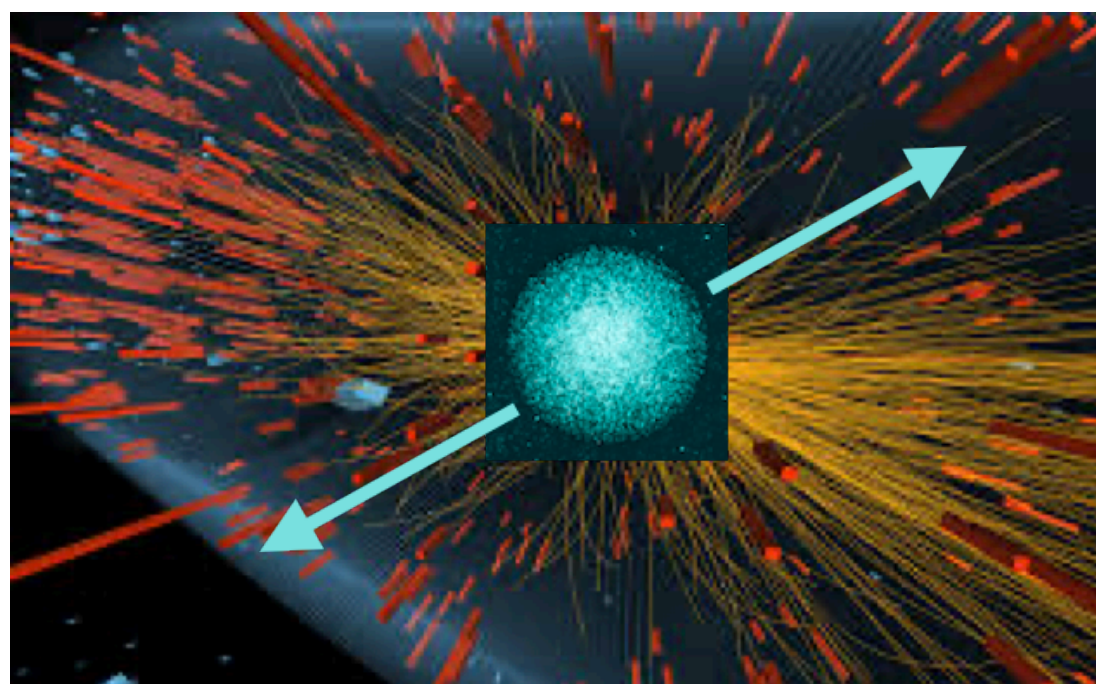
- QKeras is the Quantization Aware Training extension of Keras
- We trained ‘normal’ floating point NNs with Keras and low-precision NNs with QKeras on a benchmark jet tagging problem
- (Top plot) accuracy with QKeras training down to 6-bits is lossless wrt floating-point Keras
 - Big improvement over ‘post-training quantization’
 - Dashed line → solid lines
- As we reduce bitwidth, resource usage goes down
 - At small bitwidths LUTs are preferred over DSPs
 - The ‘critical resource’ usage decreases from **56%** (DSPs) for the Baseline (B) to **3.4%** (LUTs) for the 6-bit QKeras model (no performance loss)
 - QO model is tiny (1% DSPs/critical), 2% lost accuracy
- Right panel ‘QO’ shows some automatic optimisation of the bitwidth trading accuracy vs resource cost (AutoQ)

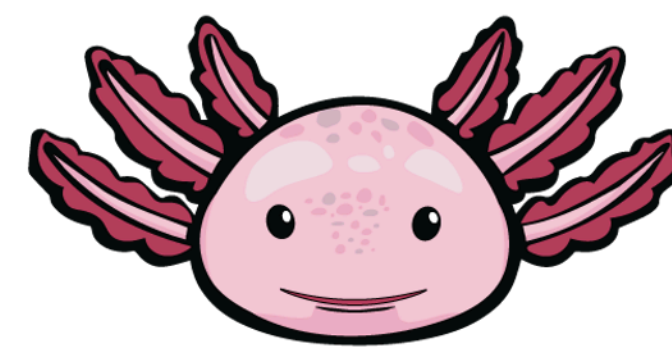
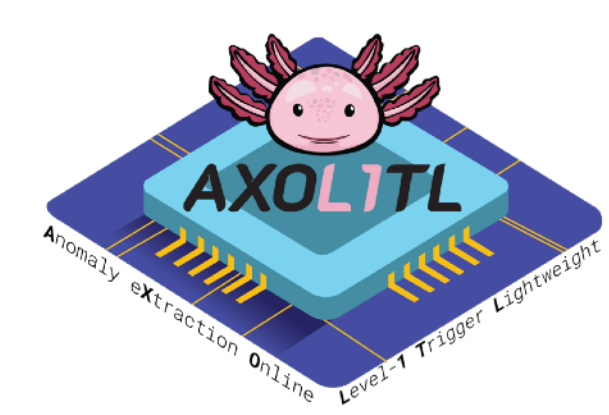




New Trigger strategies: anomaly detection

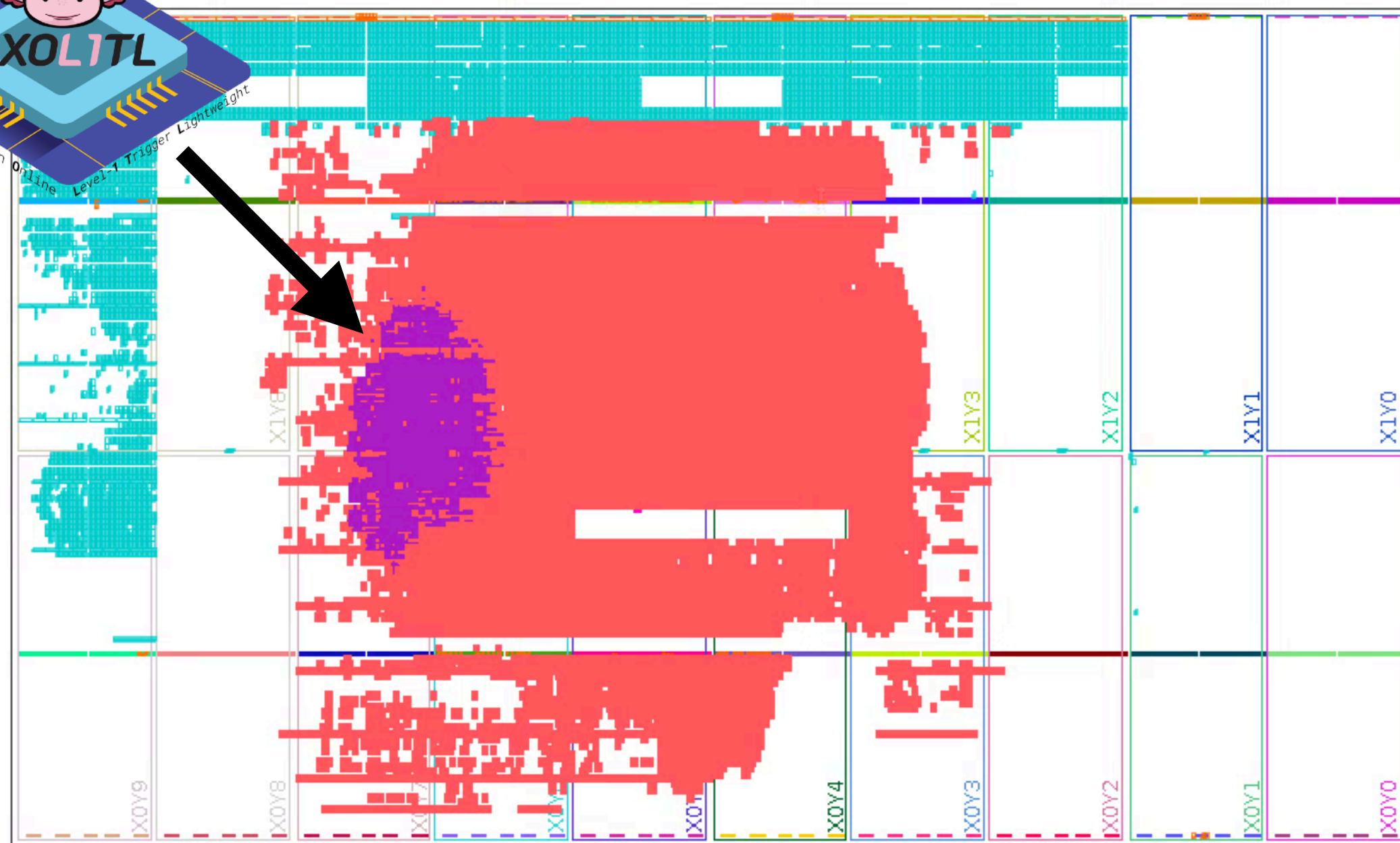
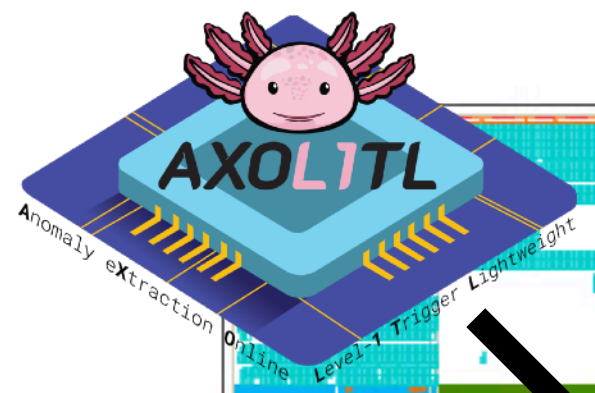
- Trigger selections introduce a bias into downstream analysis
- What if the selections we make in the trigger are wrong? We could be missing the New Physics
- Anomaly Detection method proposed to search for New Physics in a model agnostic / unbiased way
 - Train a Variational AutoEncoder on unbiased data (background + ϵ new physics)
- Variational AutoEncoder is a Neural Network trained to learn $\hat{x} = x$ with a low-dimensional latent representation
 - x is the vector of reconstructed quantities in the event: Missing Transverse Energy, Jets, electrons/photons, muons
 - Network learns \hat{x} very well for common examples, but badly for uncommon examples \rightarrow trigger on $|| \hat{x} - x ||^2$
 - For latency saving in inference we trigger on μ^2



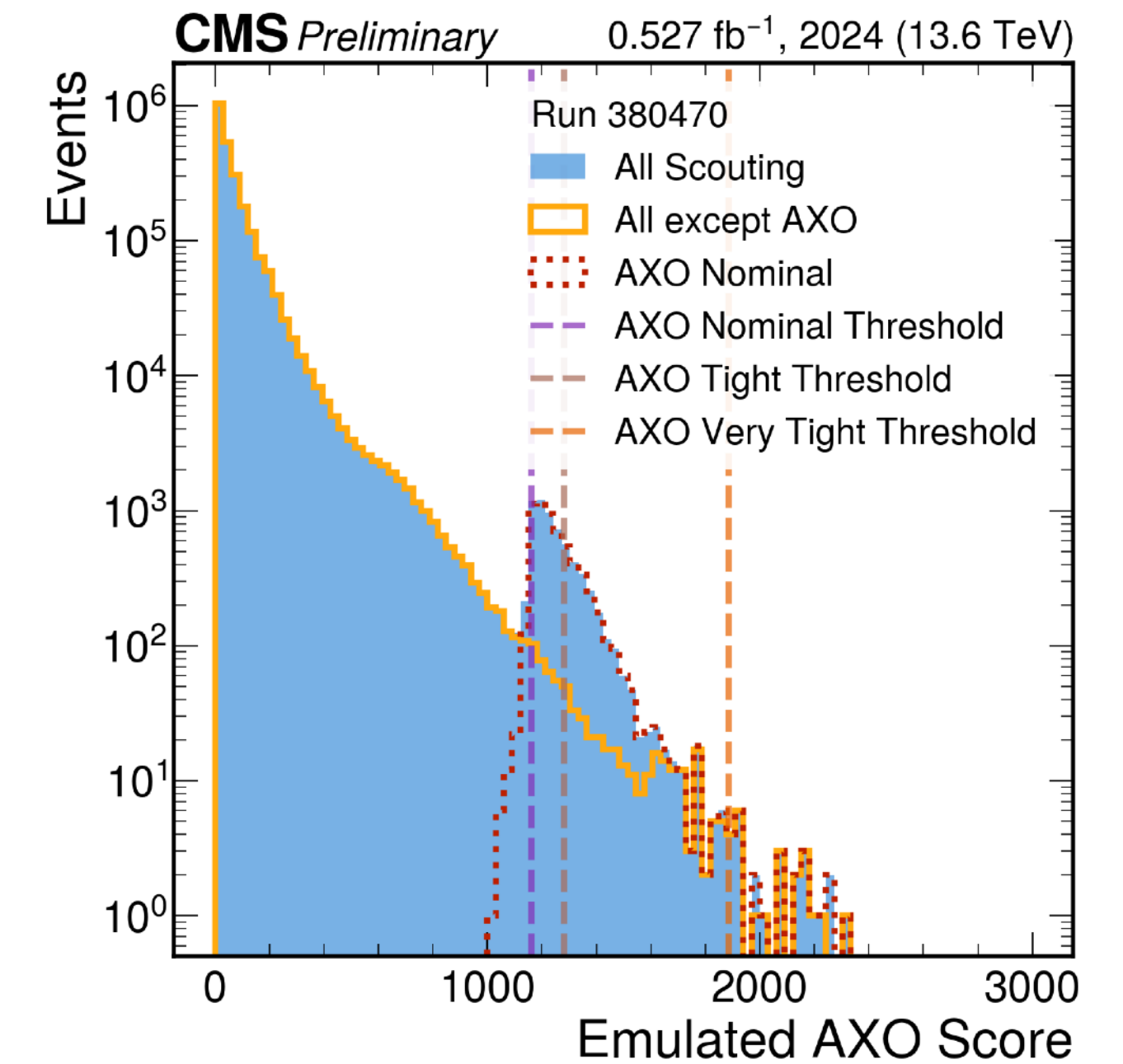
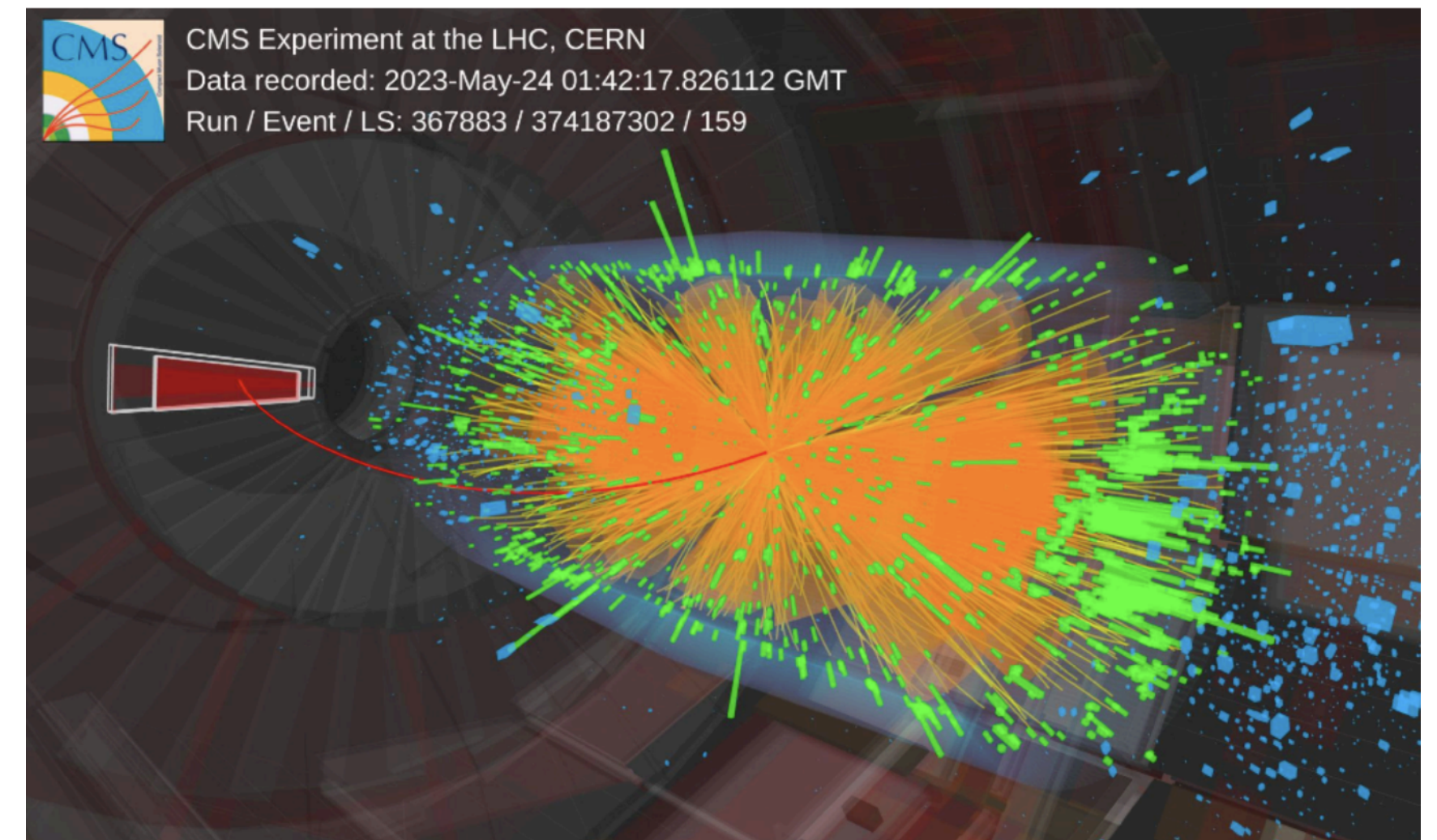


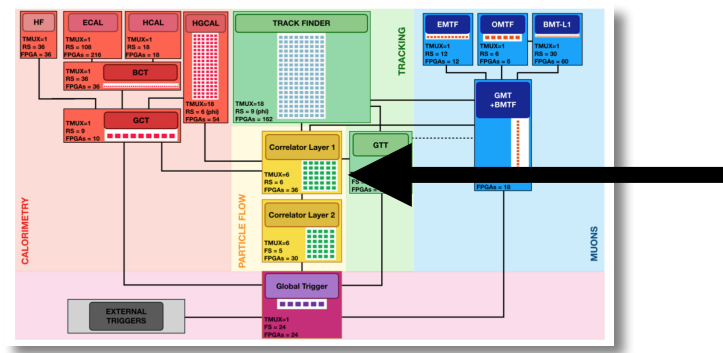
AXOL1TL

- Deployed in CMS Run 3 Global Trigger with 40 MHz event rate, 50 ns prediction latency
 - Trained with Quantization Aware Training, firmware produced with **hls4ml**
- Switched on in CMS during 2024 at 300 Hz, again for 2025
 - Analysis work is ongoing!
 - Model has been improved for 2025 thanks to Next Generation Triggers



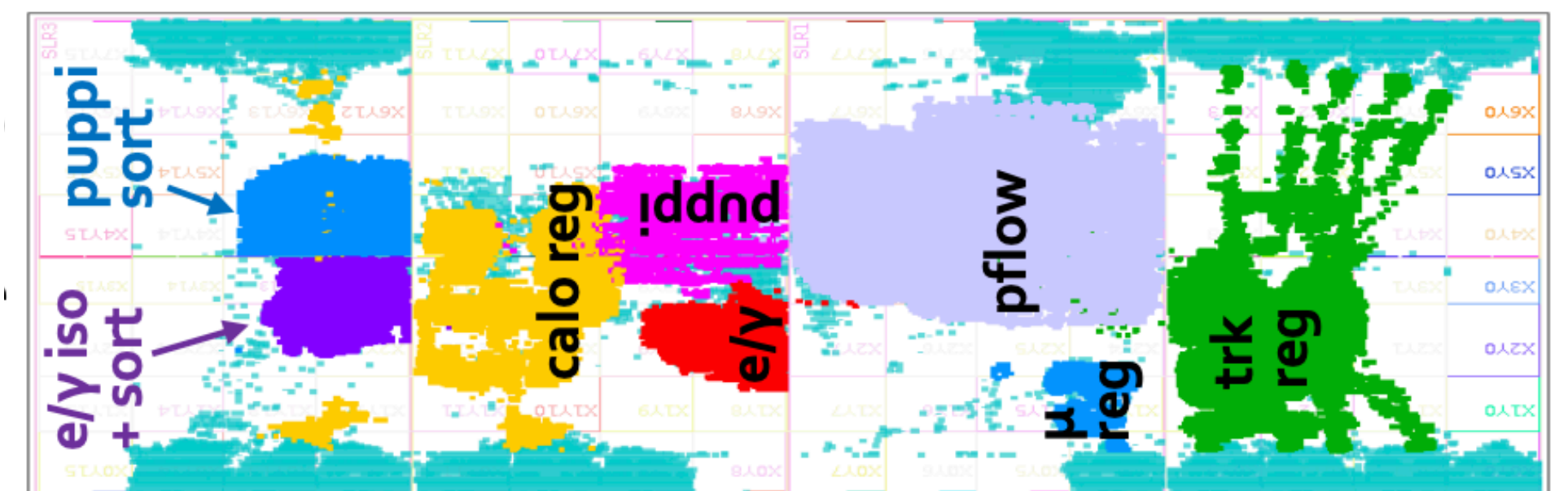
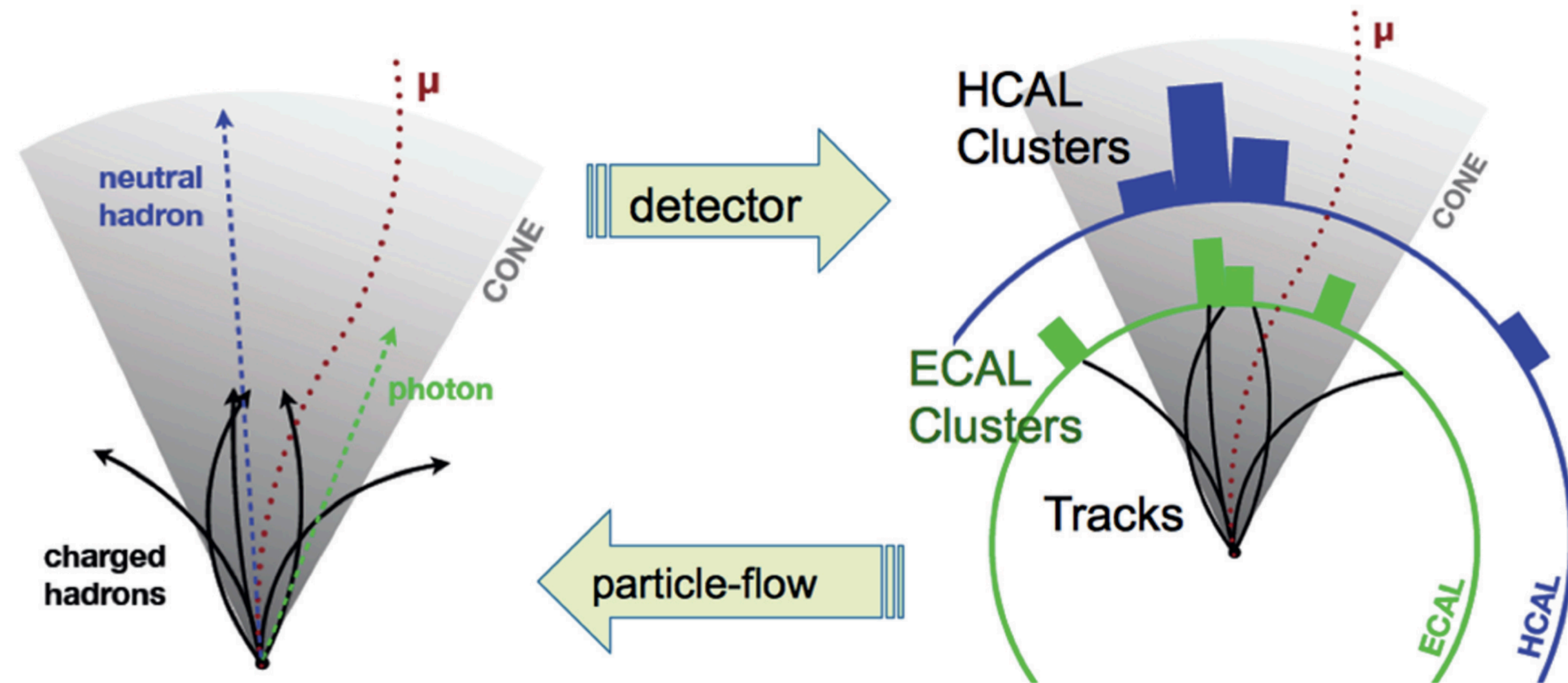
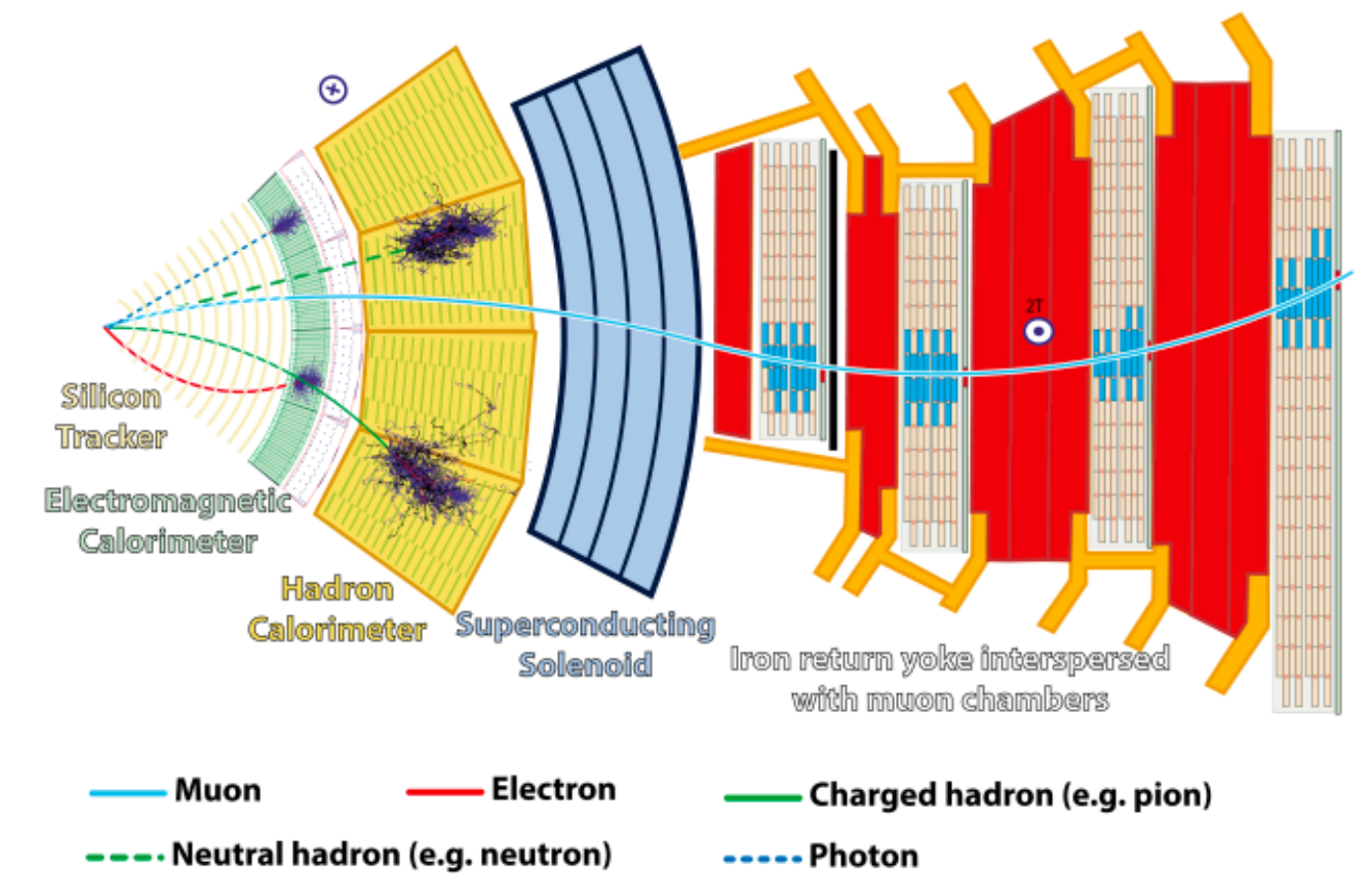
Most anomalous event only
chosen by AXOL1TL

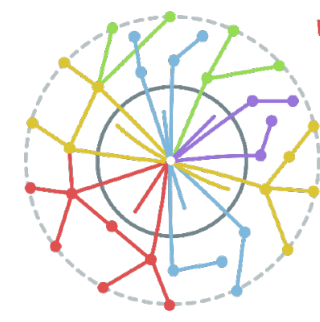
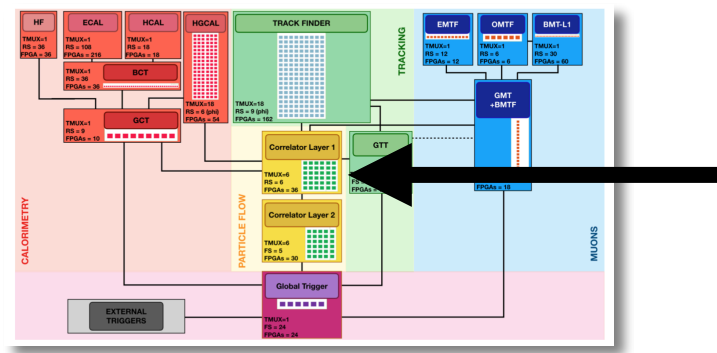




Particle Flow

- We first reconstruct different elements in individual sub-detectors
 - Like the tracks on the previous slides
- *Particle Flow* links elements from different sub-detectors to reconstruct final state *particles*
 - Need to search for tracks that link to clusters and muons that link to tracks
- We split the detector into small regional chunks
- Use FPGA *pipeline parallelism* to process them faster
- About 1 μ s to link all the clusters & tracks to particles
- 36 FPGAs to keep up with 40 MHz collisions

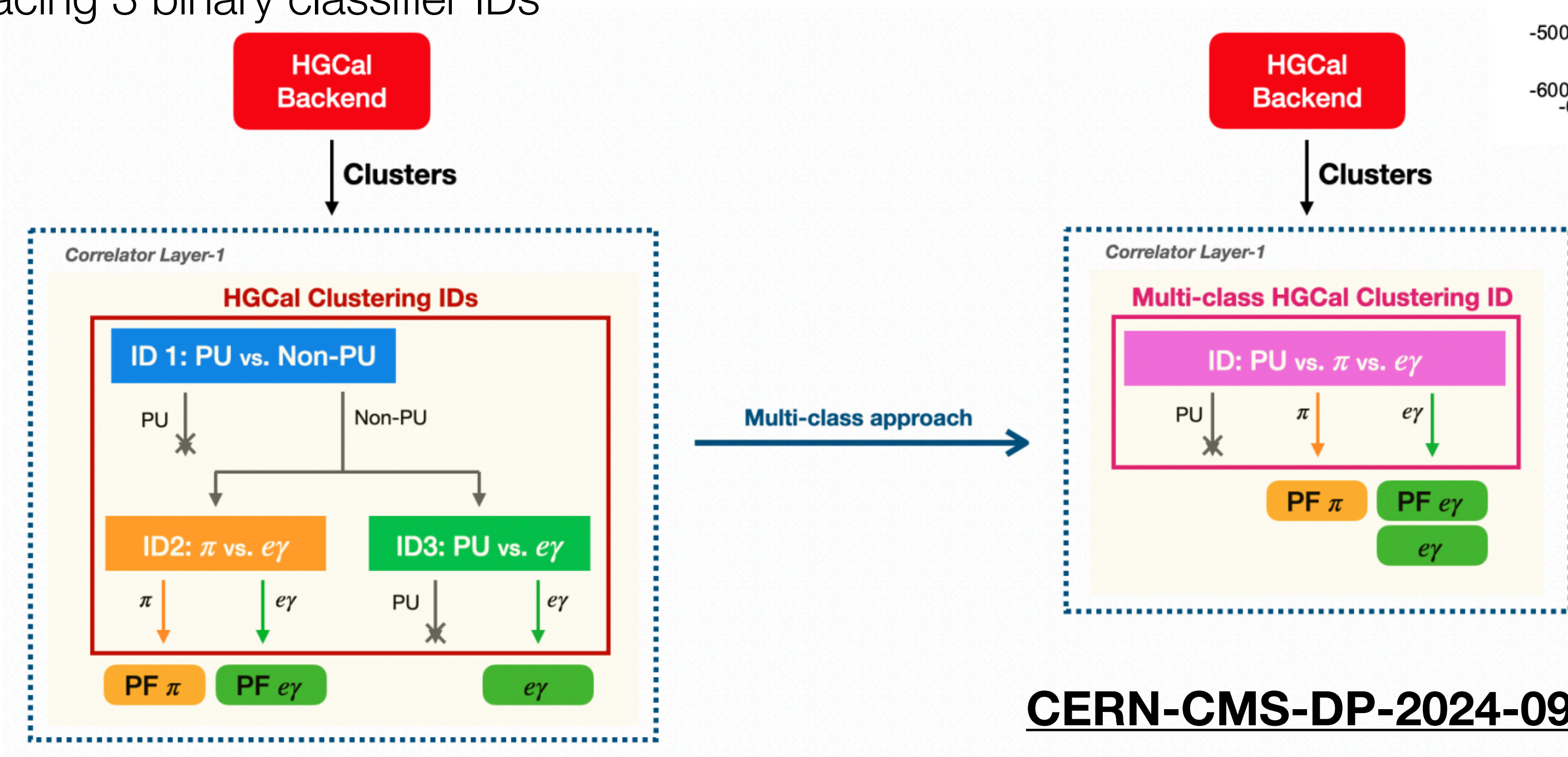
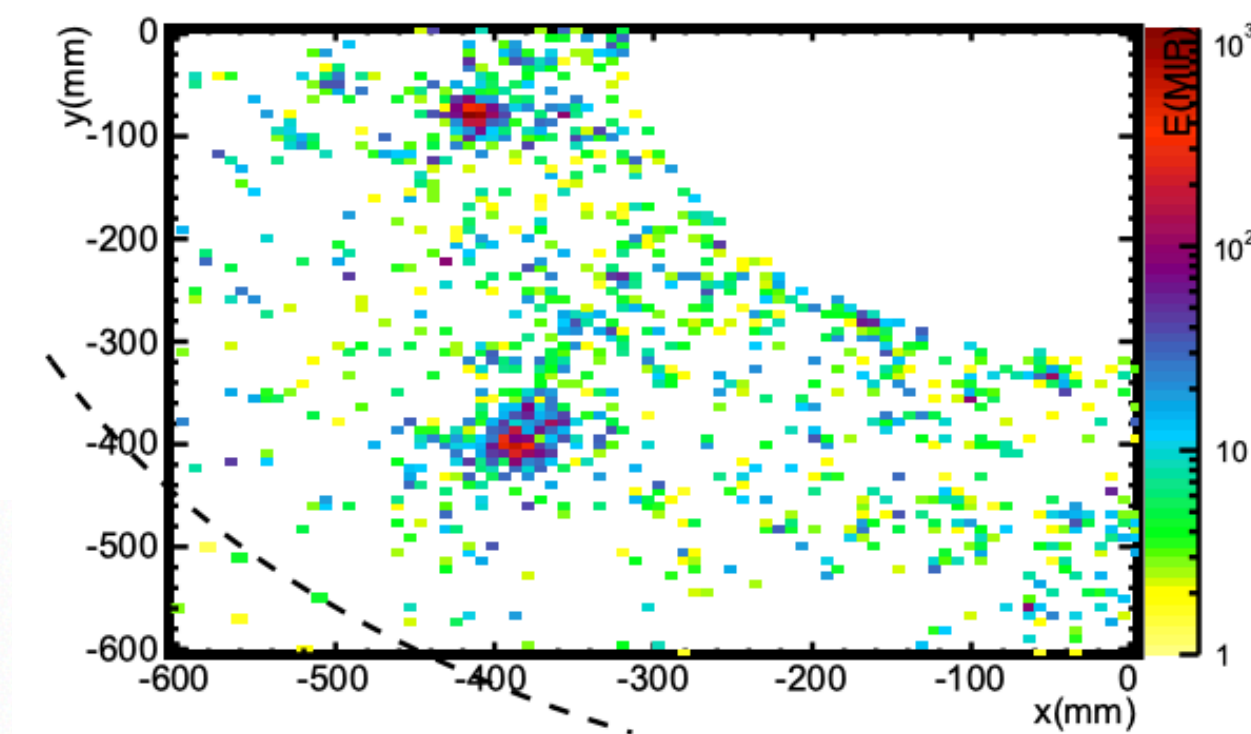
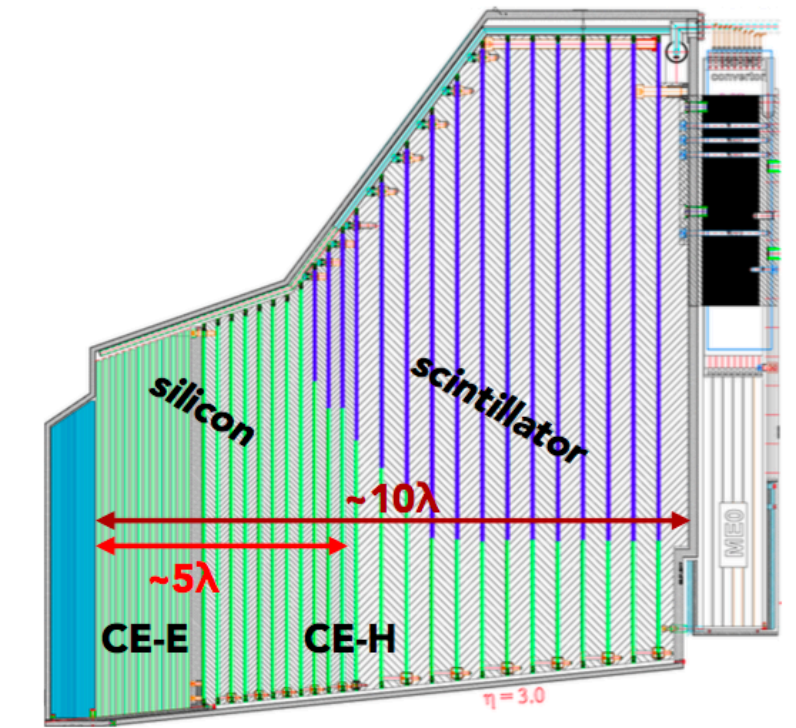


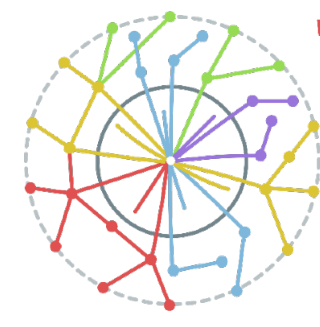
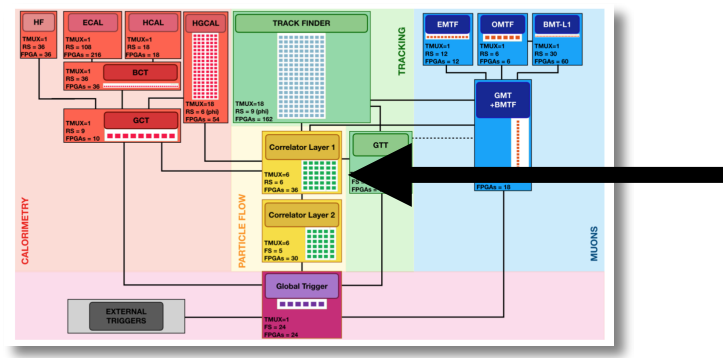


NextGen cluster ID

Next Generation Triggers

- In the CMS High Granularity Endcap, clusters are filtered before the track-cluster linking step
 - Many clusters are from pileup
 - We can distinguish electromagnetic from hadronic clusters with cluster properties
- Cluster filtering improves background rejection, and reduces combinatorics for linking logic
- Using BDTs for lightweight fast cluster ID
- In 2024 we improved and harmonised the cluster filtering step
 - One 3-class ID replacing 3 binary classifier IDs

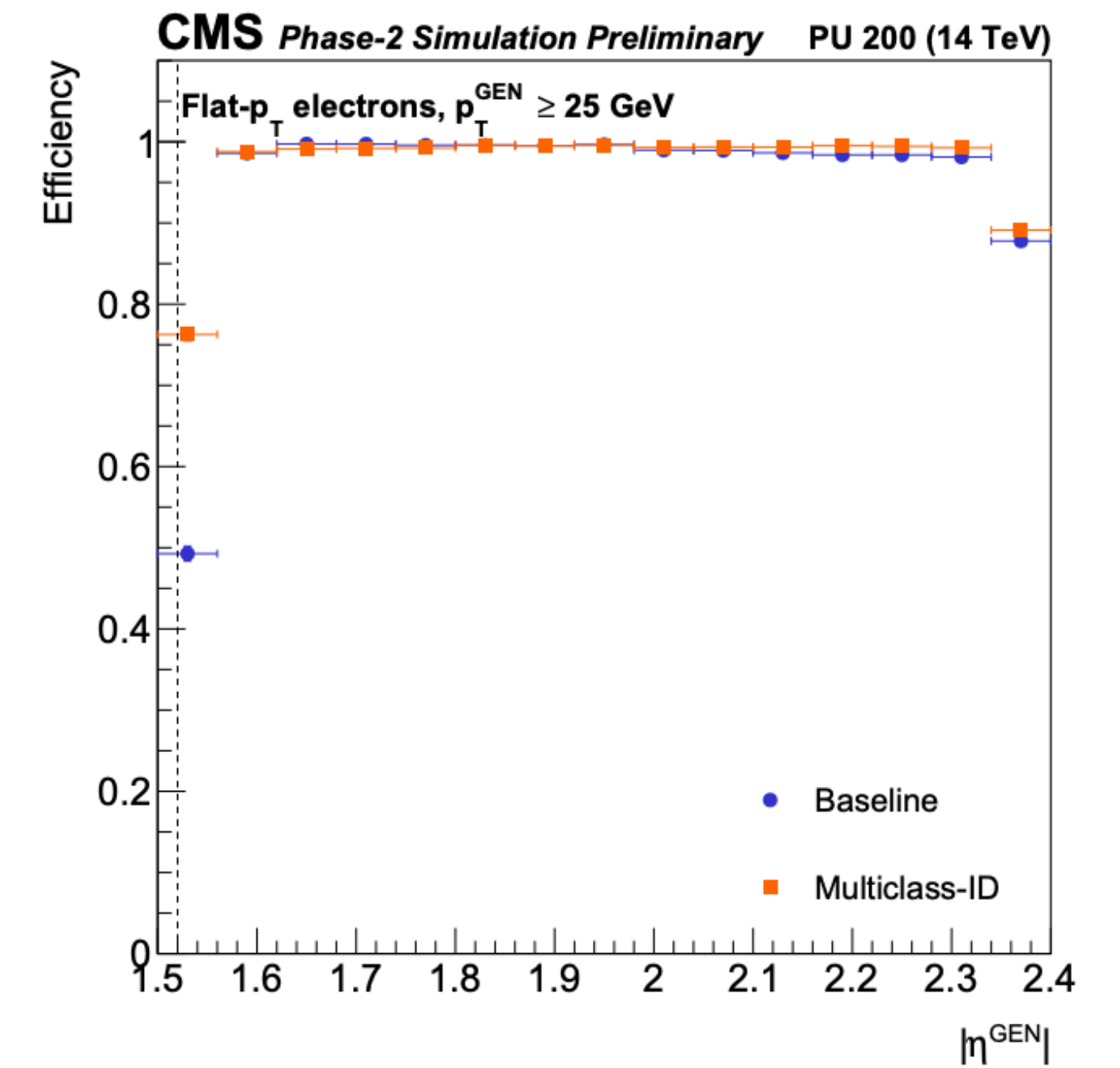
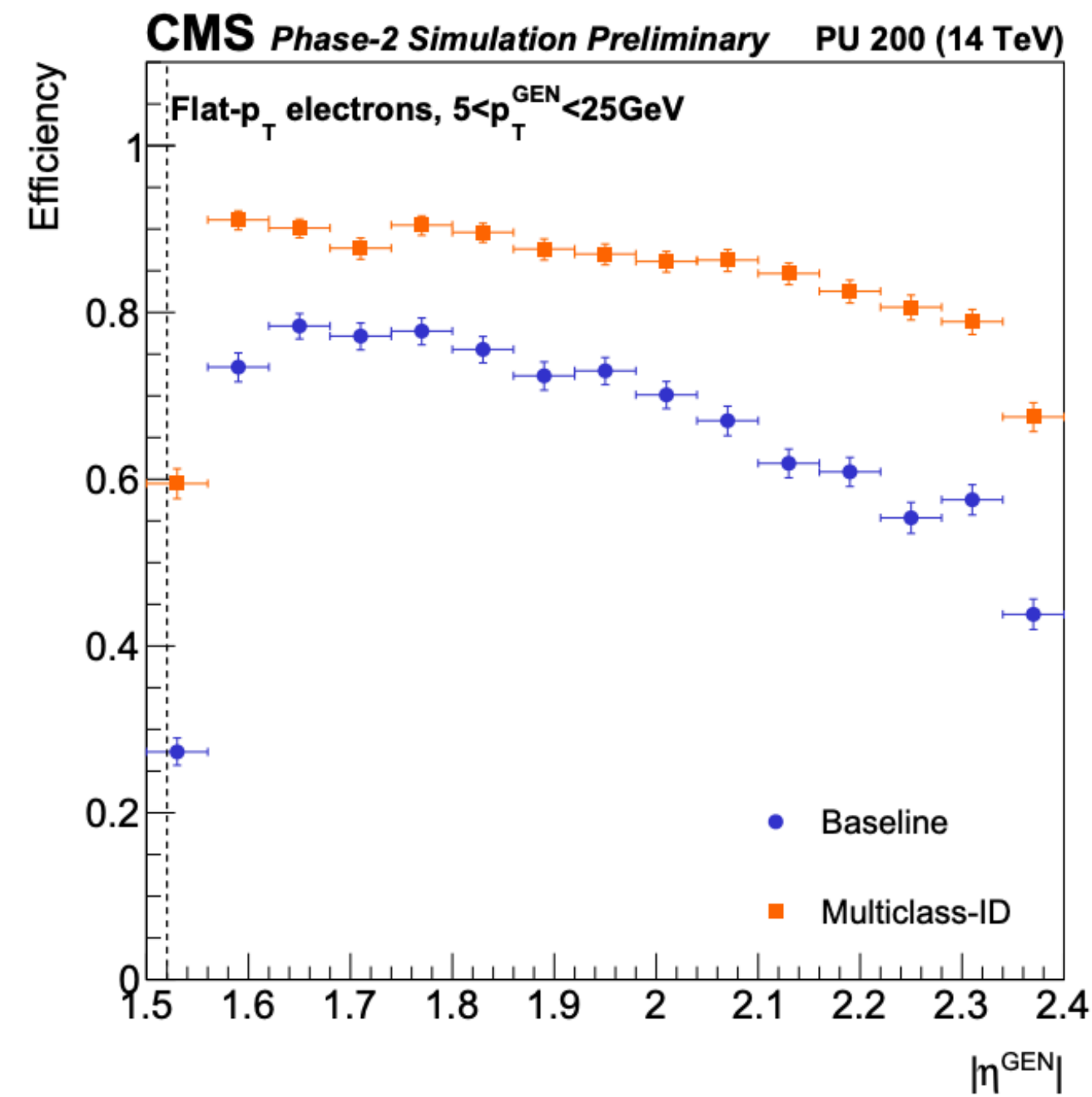
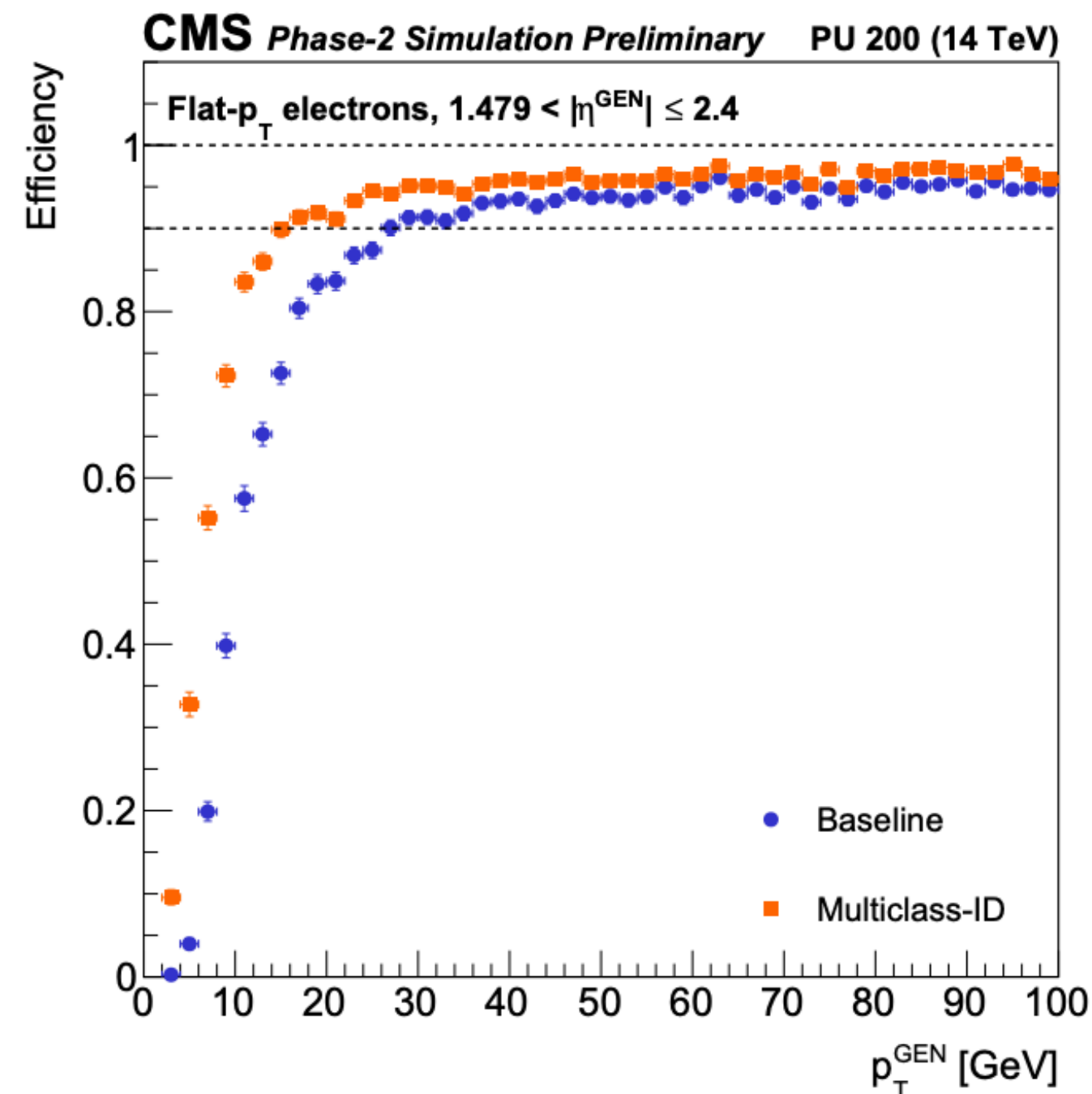


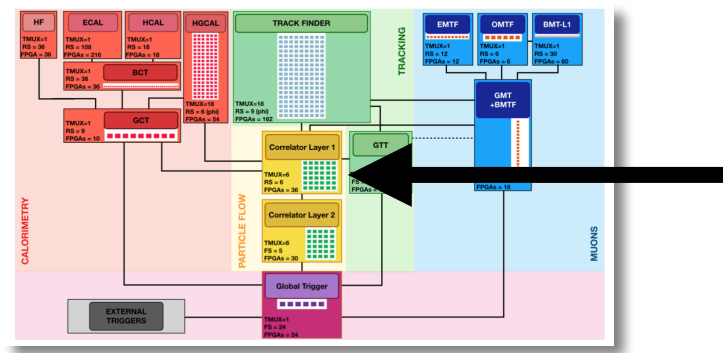


NextGen cluster ID

Next Generation Triggers

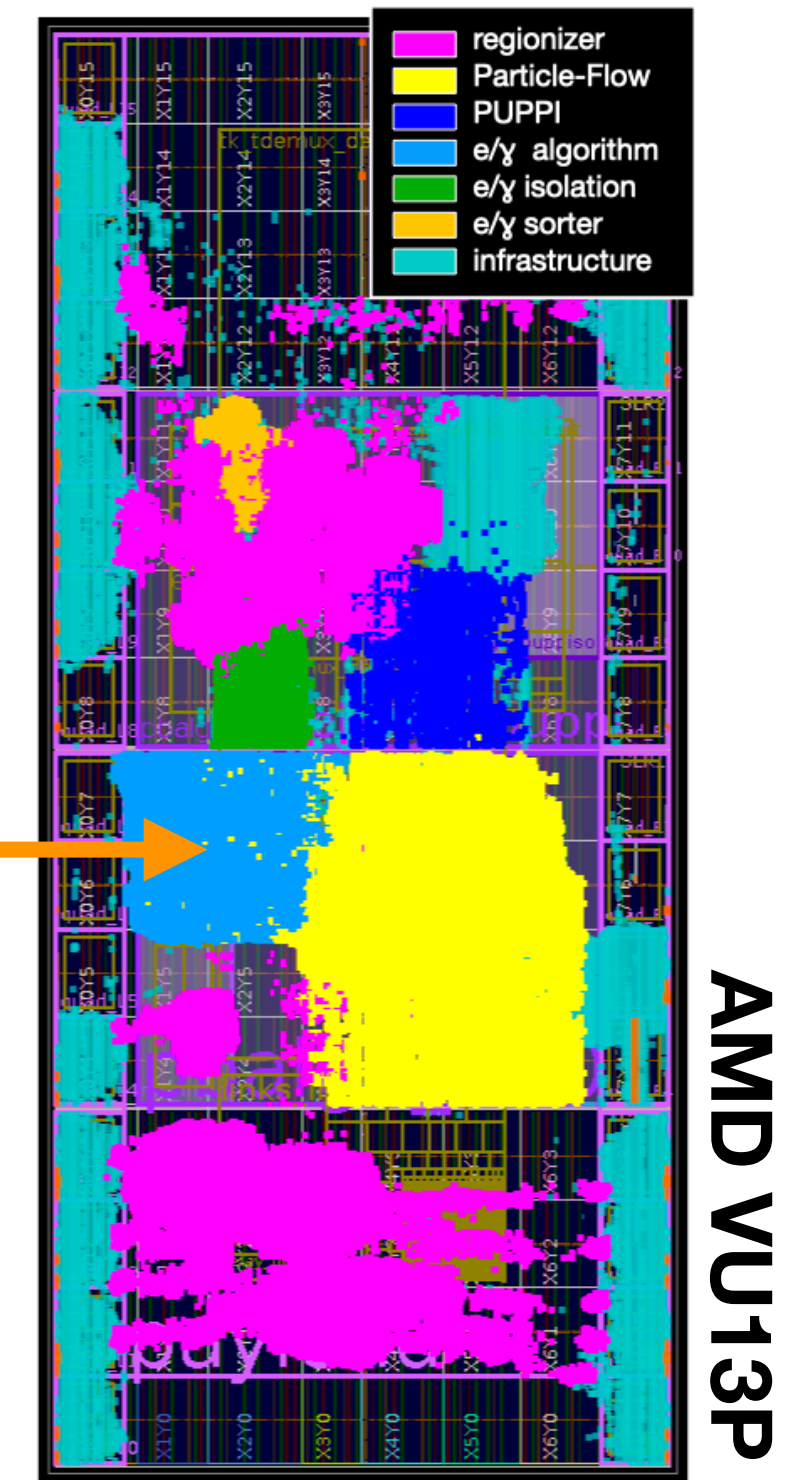
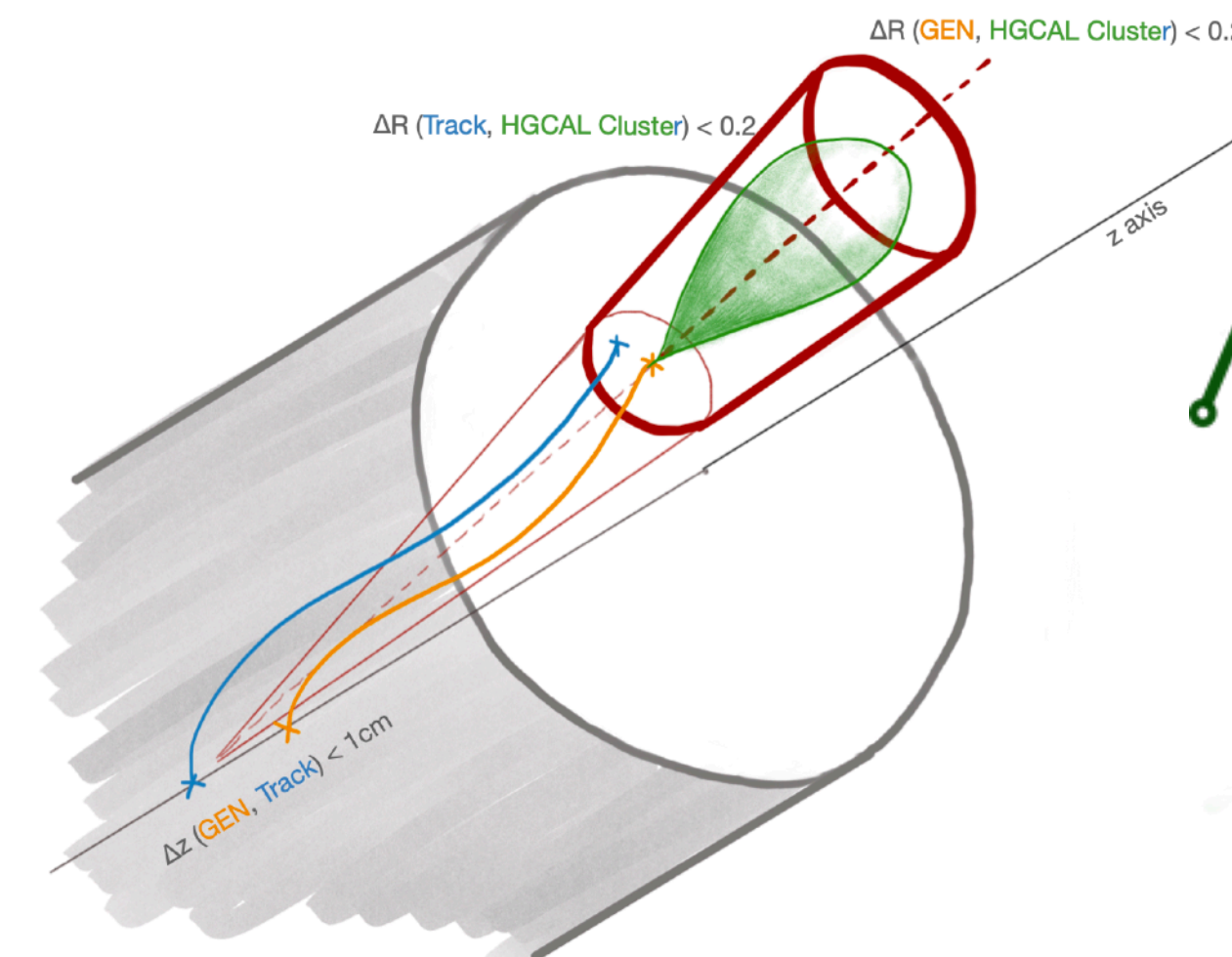
- New multiclass ID uses cluster variables:
 - Core shower length, shower length, electromagnetic energy fraction, shower shapes ($\sigma_{\phi\phi}$, $\sigma_{\eta\eta}$, σ_{zz}), $|\eta|$, mean z
- Only clusters with high probability of electromagnetic class are used for electron track-cluster linking
- High probability pileup clusters are rejected from electron linking and Particle Flow
- New ID improves efficiency of triggering on electrons especially for low p_T - useful for low p_T analyses in L1 Scouting!





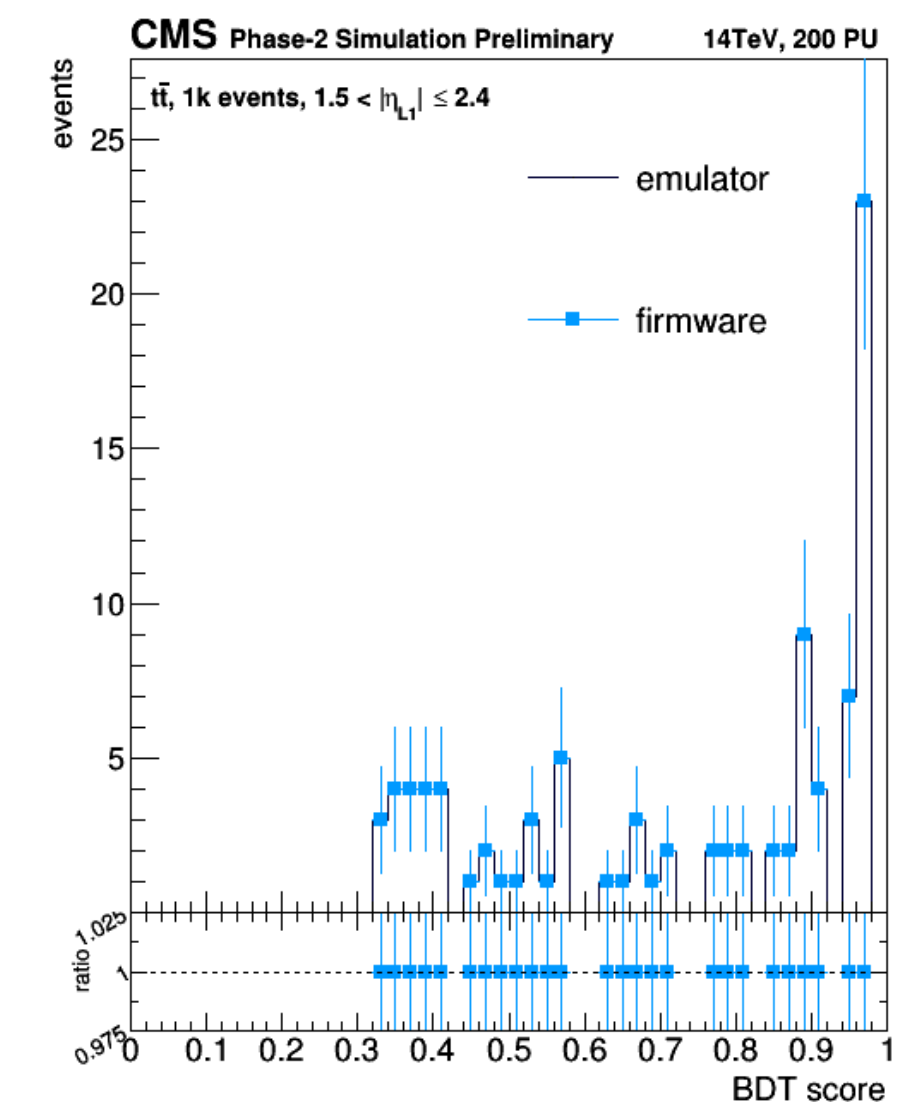
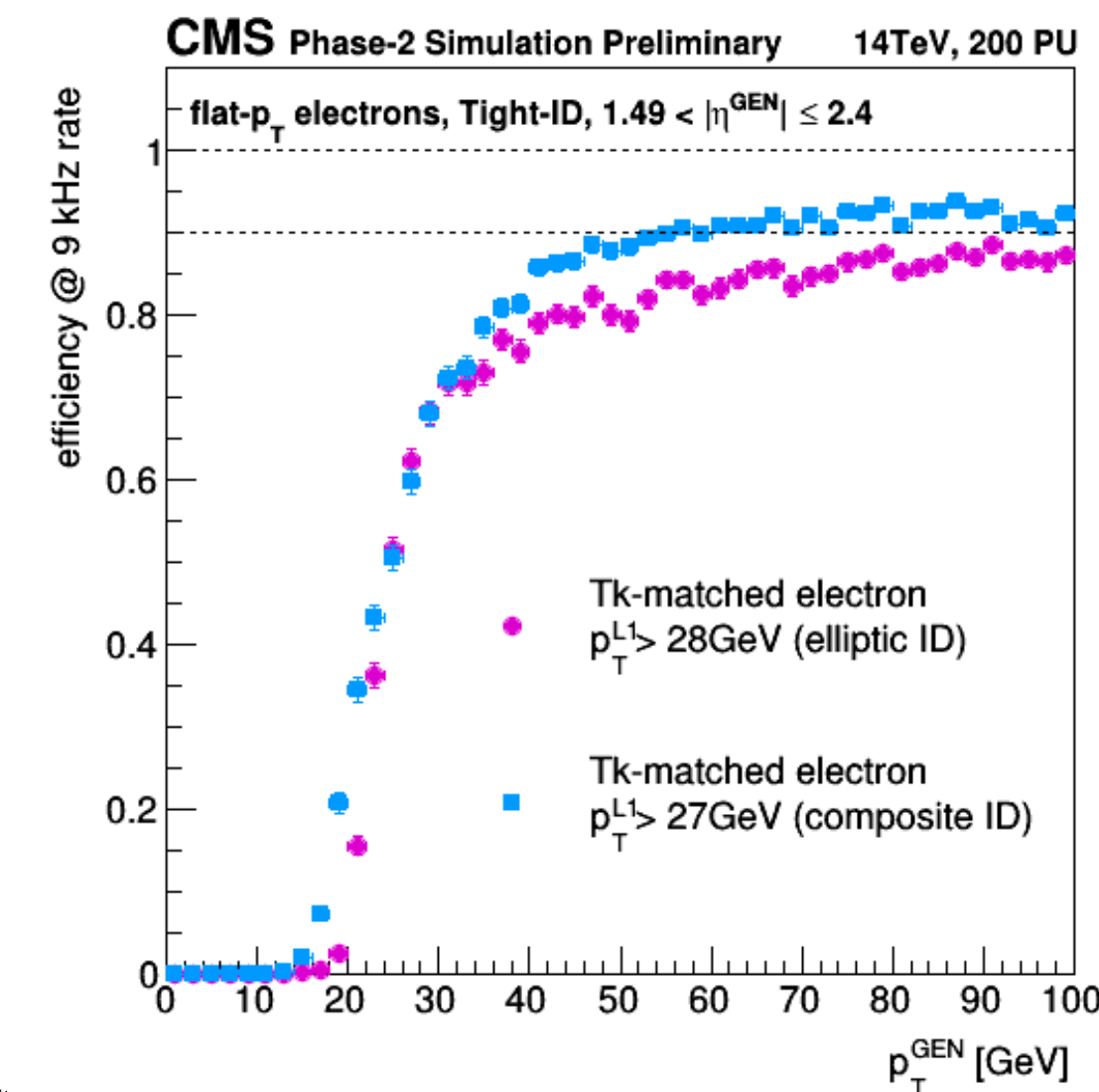
CMS Phase 2 L1T electron ID

- Electrons will be reconstructed by linking a track with a calorimeter cluster
- Neither reconstruction is perfect, and electrons radiate photons
- **Baseline kinematic approach** used distance and p_T compatibility to make a link
- **New BDT approach** first makes a loose kinematic selection, then uses ML to predict probability that the track & cluster both originated from an electron
- Improved electron reconstruction efficiency with new method (bottom left)
- xgboost for model training, **conifer** for inference
 - Tiny model with 10 trees & maximum depth 4
 - 10 parallel model copies to keep up with electron rate
 - Well within system resource and latency budget

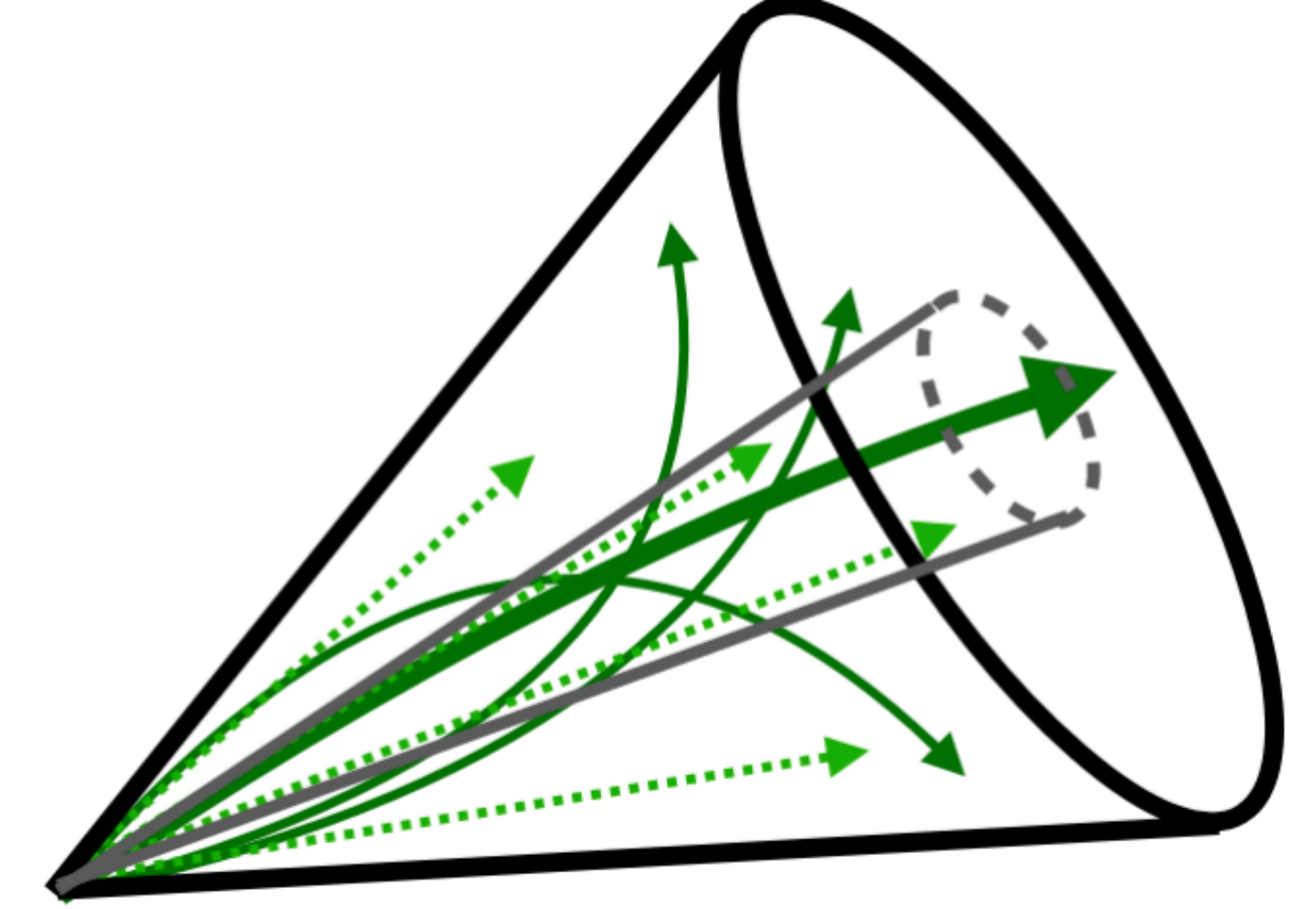
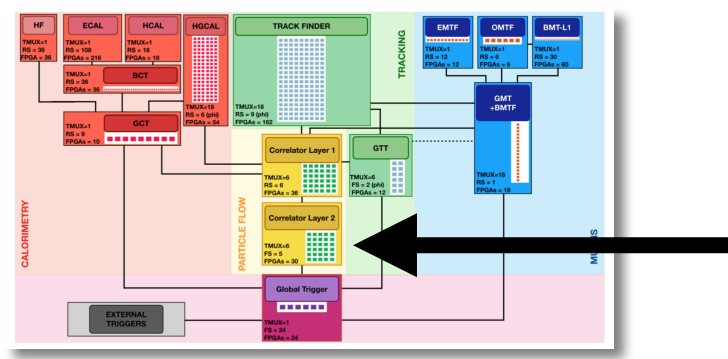


AMD VU13P

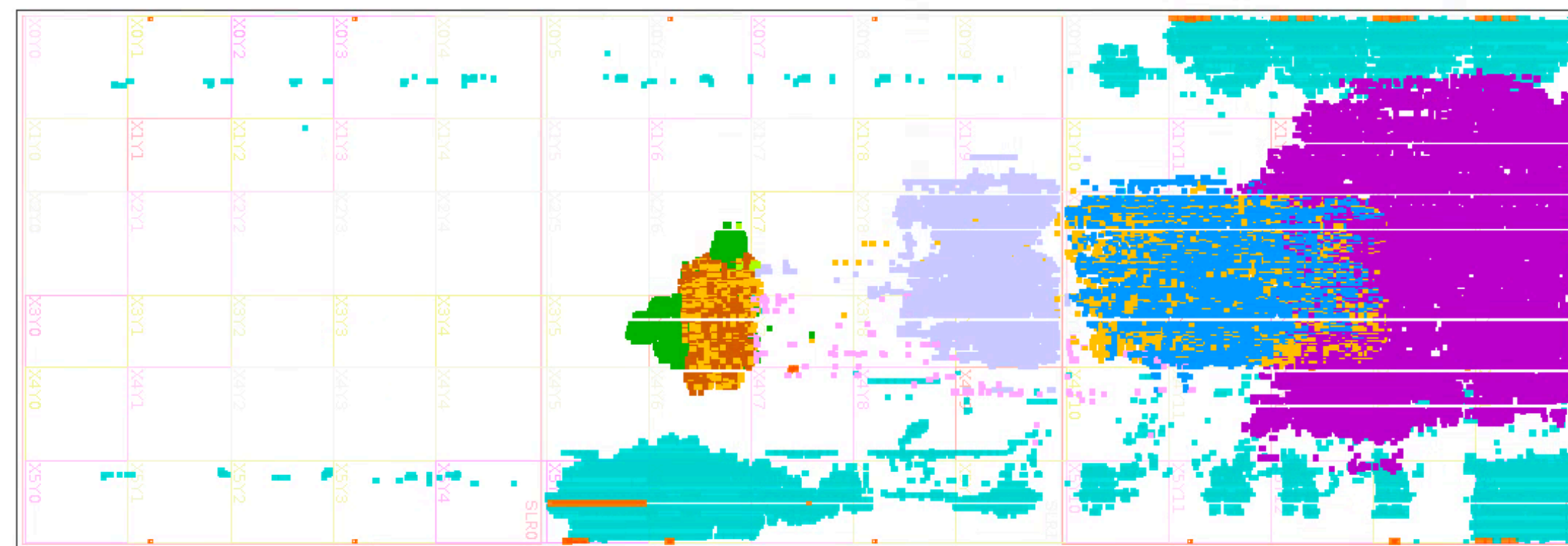
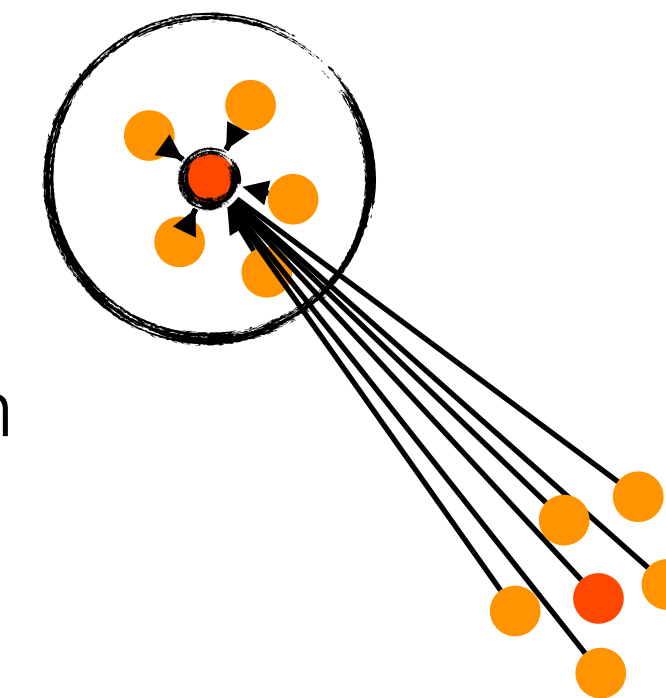
CMS-DP-2023-047



Jet Tagging

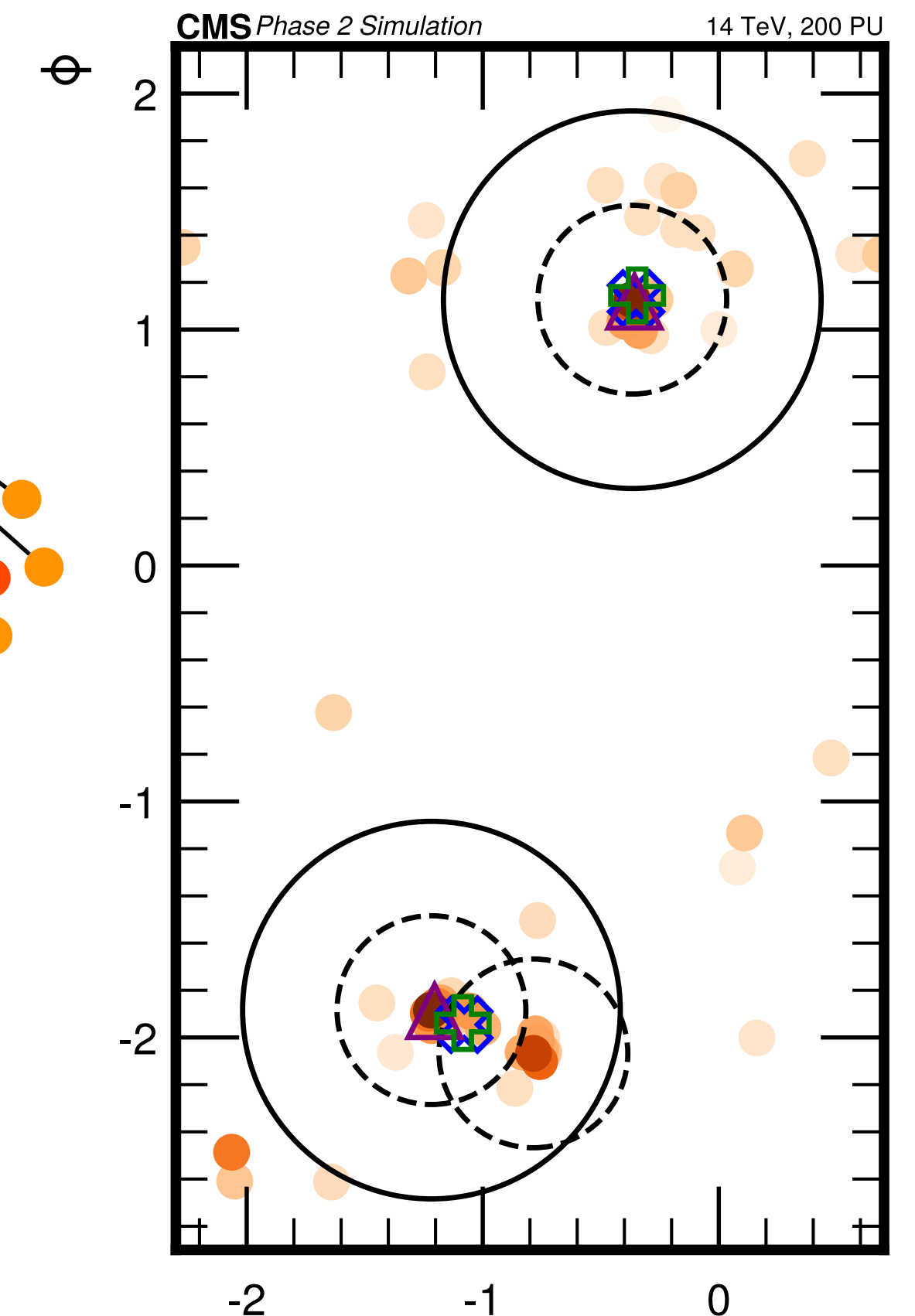


- Jets are collimated sprays of particles from the hadronization of quarks and gluons
- We measure the final state particles in the silicon tracker, calorimeters, muon systems
- We are interested in the properties of the initiating particle: its momentum, direction, and particle type (flavour)
- We implement a simple cone algorithm to find jets and their constituents
 - Buffer the constituents in the FPGA → give to a Neural Network for identification



AMD VU9P

- 750 ns from first particle in to last jet out
- 100 M jets / second
- ~ 10% FPGA logic used



doi.org/10.1051/epjconf/202429502024

Jet Tagging Architectures for L1T

- Now we have the clustered particles in one place in the FPGA, we can send them to a Neural Network for tagging
- Which Neural Network model architectures performs well for jet tagging, and can we deploy it in an FPGA with O(100) ns latency and O(100) MHz throughput?

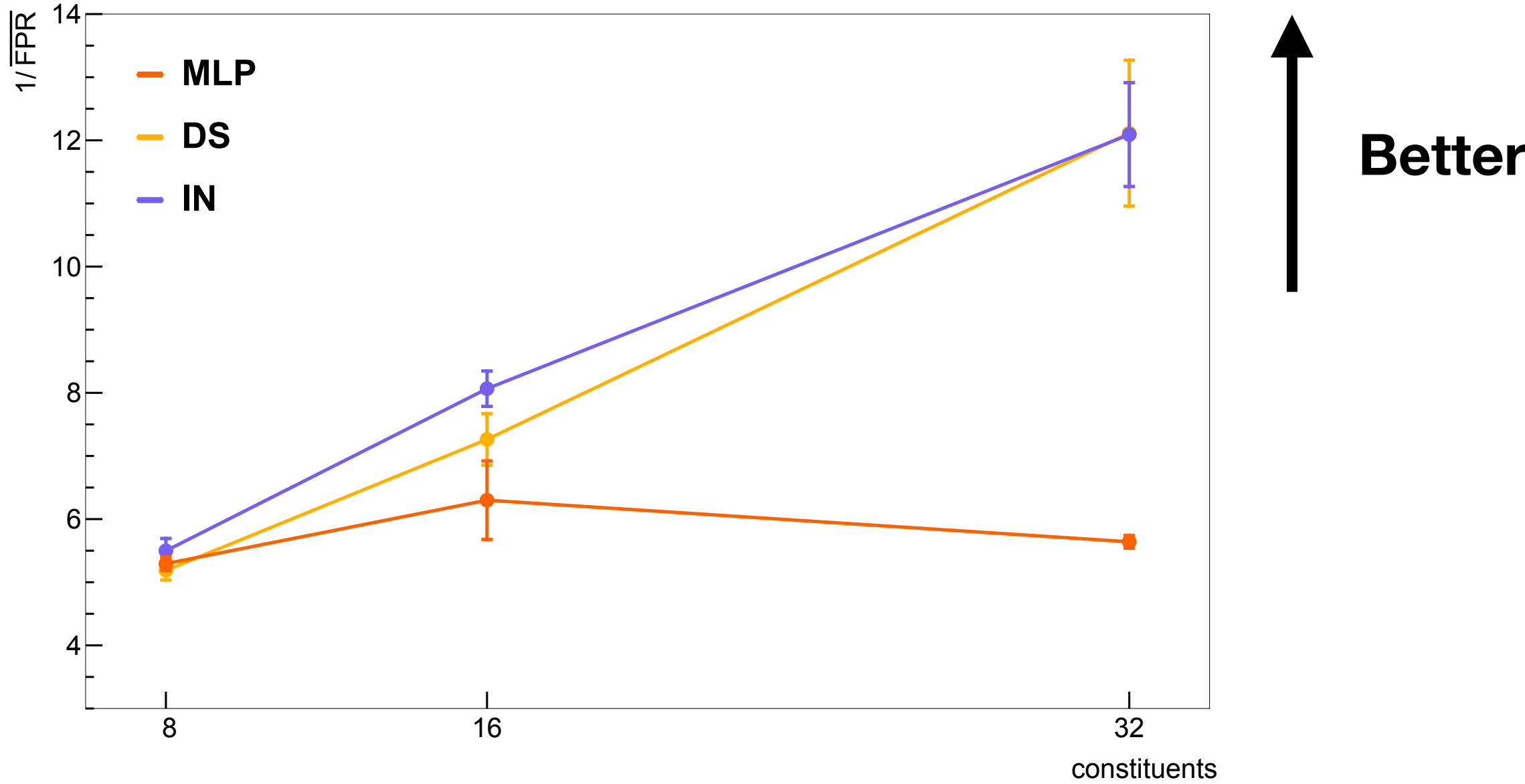
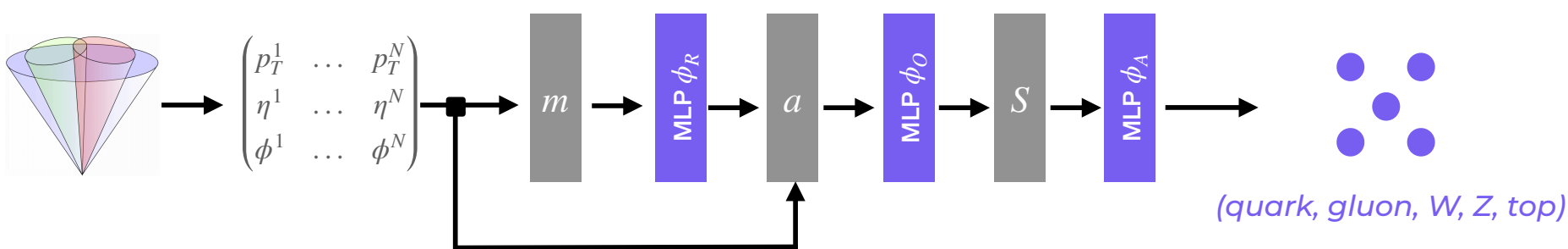
a) Multilayer Perceptron MLP






b) Deep Sets DS

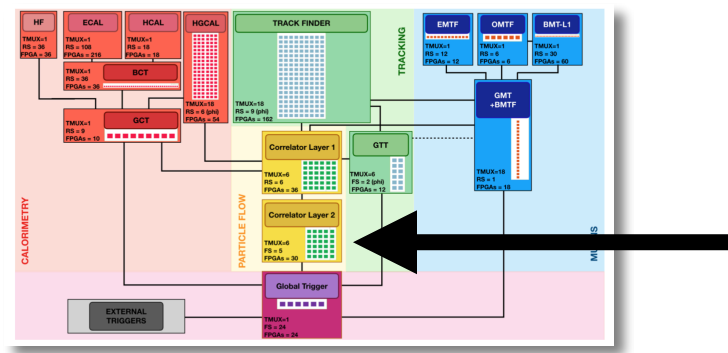


c) Interaction Network IN



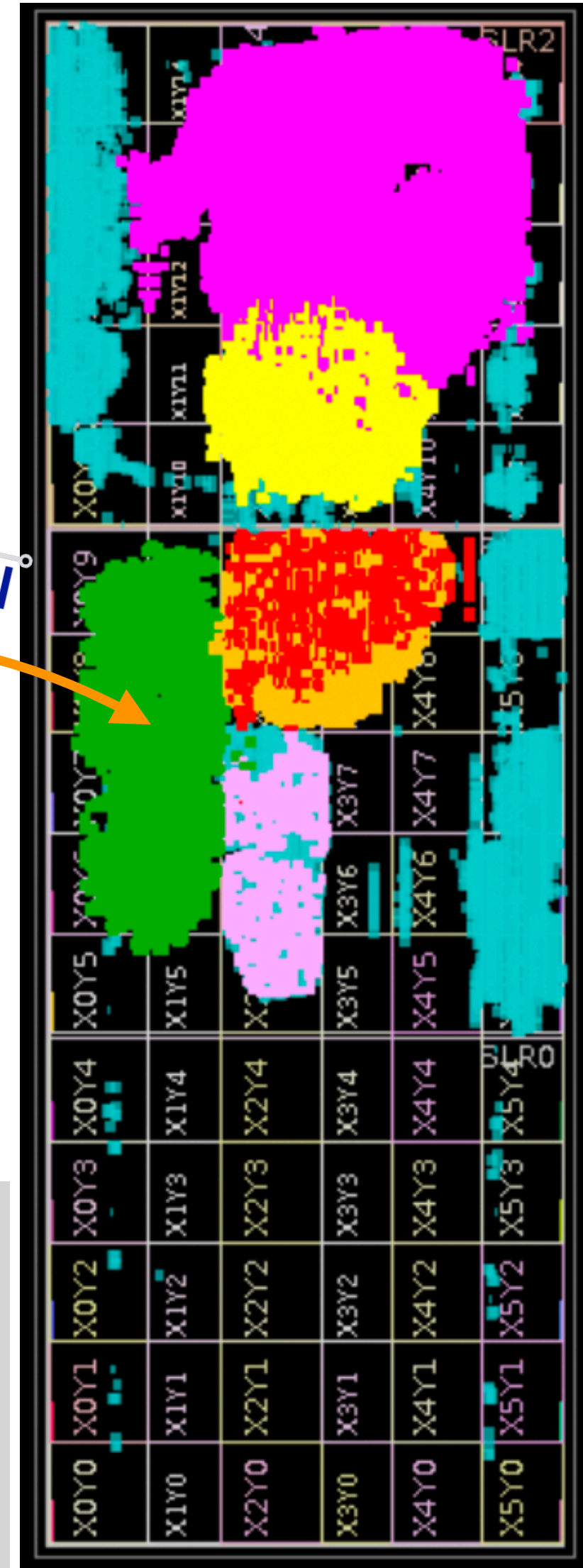
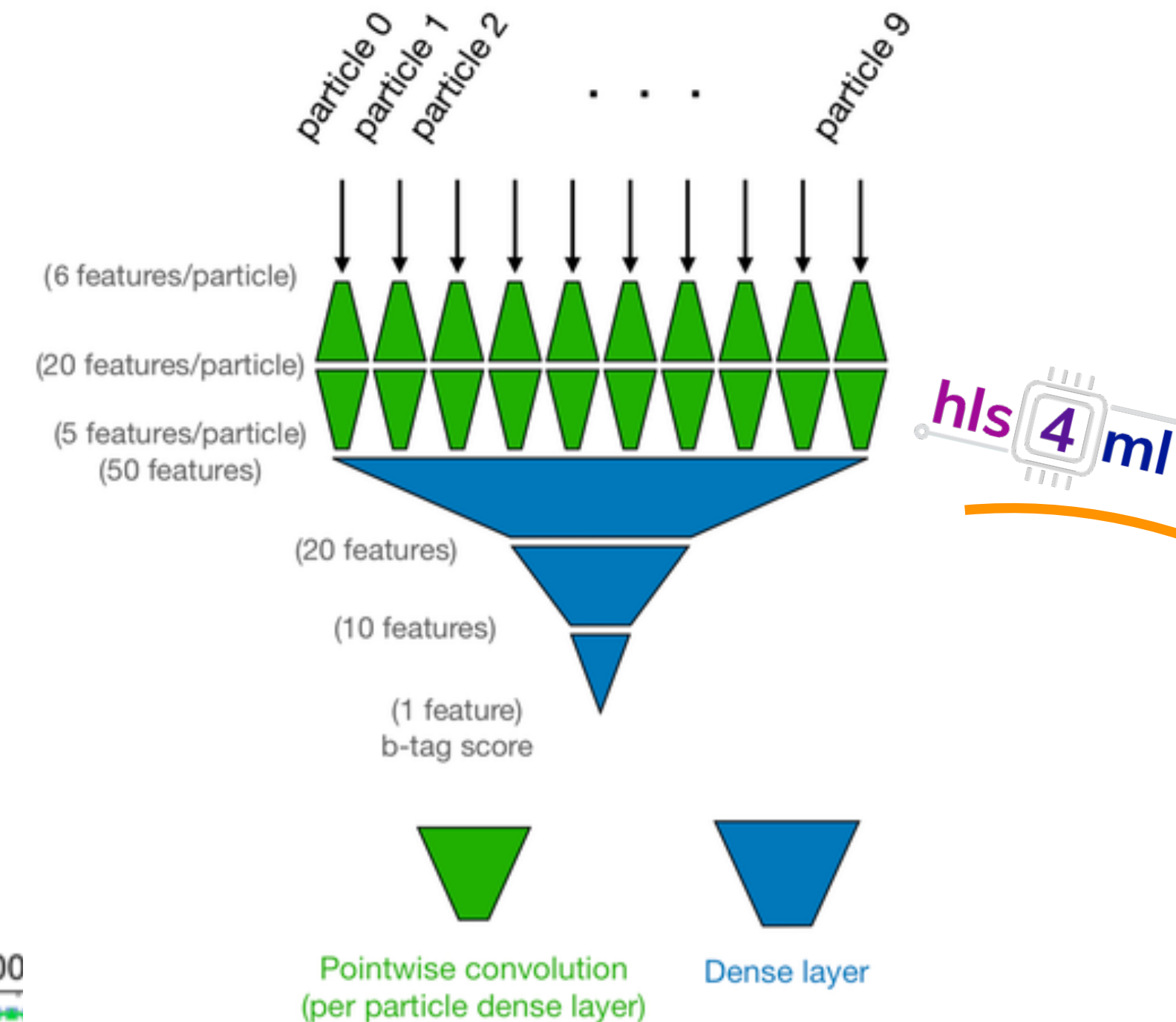
[arXiv:2402.01876](https://arxiv.org/abs/2402.01876)

	Architecture	Constituents	RF	Latency [ns] (cc)	II [ns] (cc)	DSP	LUT	FF	BRAM18
	MLP	8	1	105 (21)	5 (1)	262 (2.1%)	155,080 (9.0%)	25,714 (0.7%)	4 (0.1%)
		16	1	100 (20)	5 (1)	226 (1.8%)	146,515 (8.5%)	31,426 (0.9%)	4 (0.1%)
		32 ^a	1	105 (21)	5 (1)	262 (2.1%)	155,080 (7.2%)	25,714 (0.7%)	4 (0.1%)
	DS	8	2	95 (19)	15 (3)	626 (5.1%)	386,294 (22.3%)	121,424 (3.5%)	4 (0.1%)
		16	4	115 (23)	15 (3)	555 (4.5%)	747,374 (43.2%)	238,798 (6.9%)	4 (0.1%)
		32 ^a	8	130 (26)	10 (2)	434 (3.5%)	903,284 (52.3%)	358,754 (10.4%)	4 (0.1%)
	IN	8	2	160 (32)	15 (3)	2,191 (17.8%)	472,140 (27.3%)	191,802 (5.5%)	12 (0.2%)
		16	4	180 (36)	15 (3)	5,362 (43.6%)	1,387,923 (80.3%)	594,039 (17.2%)	52 (1.9%)
		32 ^a	8	205 (41)	15 (3)	2,120 (17.3%)	1,162,104 (67.3%)	761,061 (22.0%)	132 (2.5%)



Jet Tagging at CMS L1T

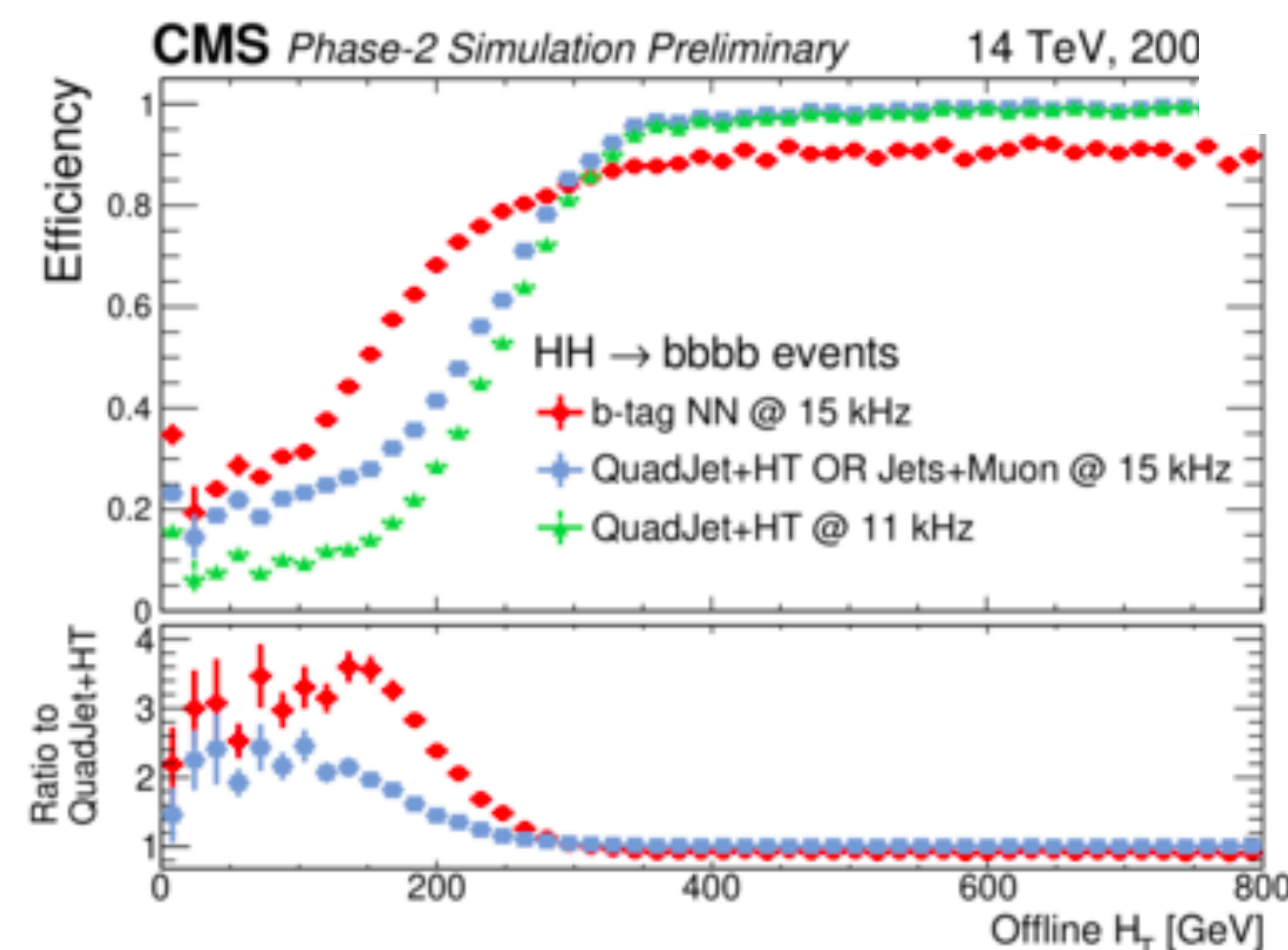
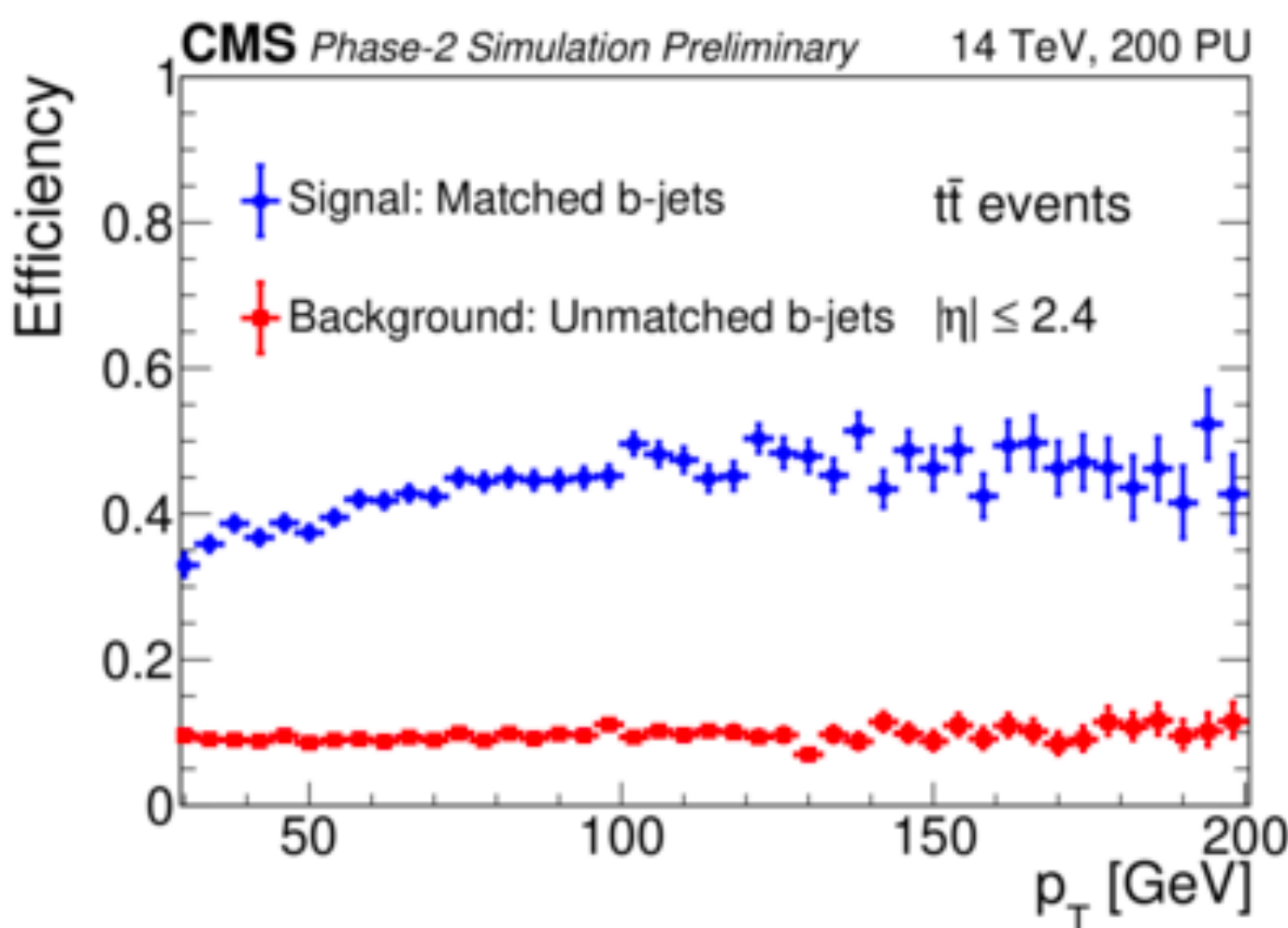
- Now we give the jet constituents to a Neural Network
- Use variables of all particles to identify b jets
 - Relies on track displacement measurement from L1 track finder
- Tiny model improves trigger reach to important final states (HH → bbbb shown)
- Fits in FPGA (right) and latency of 200 ns (total reconstruction + tagging less than 1 μs)
- We've been enhancing this algorithm with Next Generation Triggers

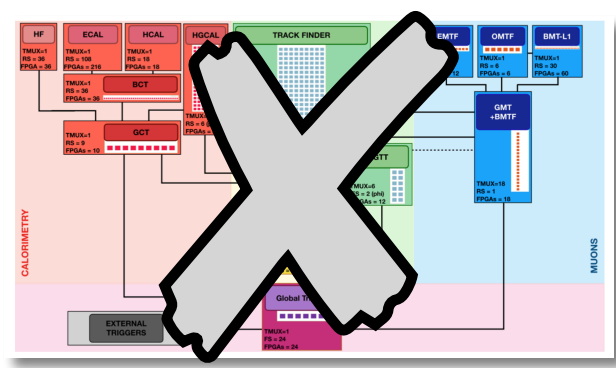


Particle Receiving
Jet Constituent Finding
Jet Axis Computation
Sorting, Buffering
B tagging Neural Network

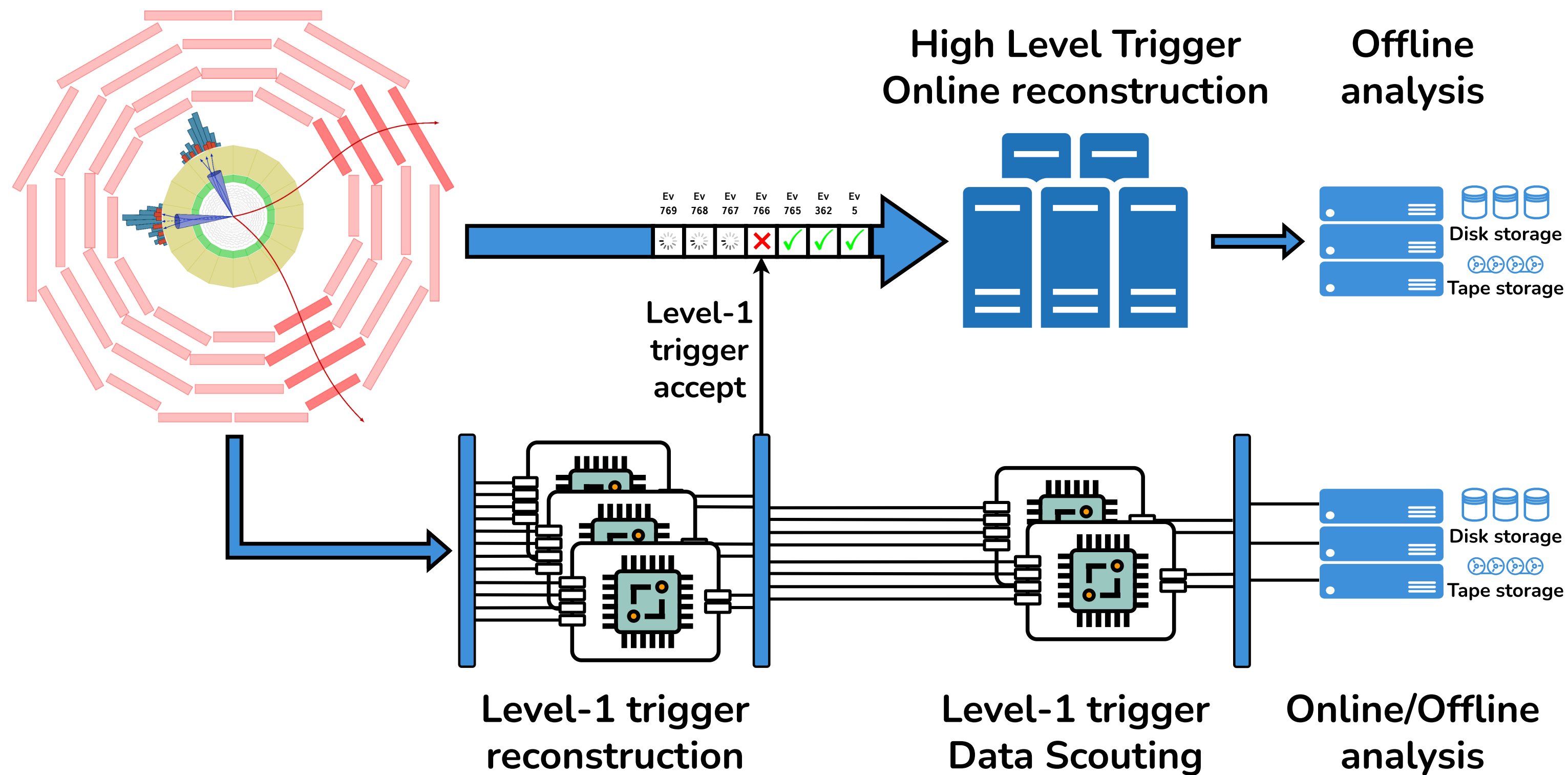
CMS-DP-2022-021

AMD VU9P





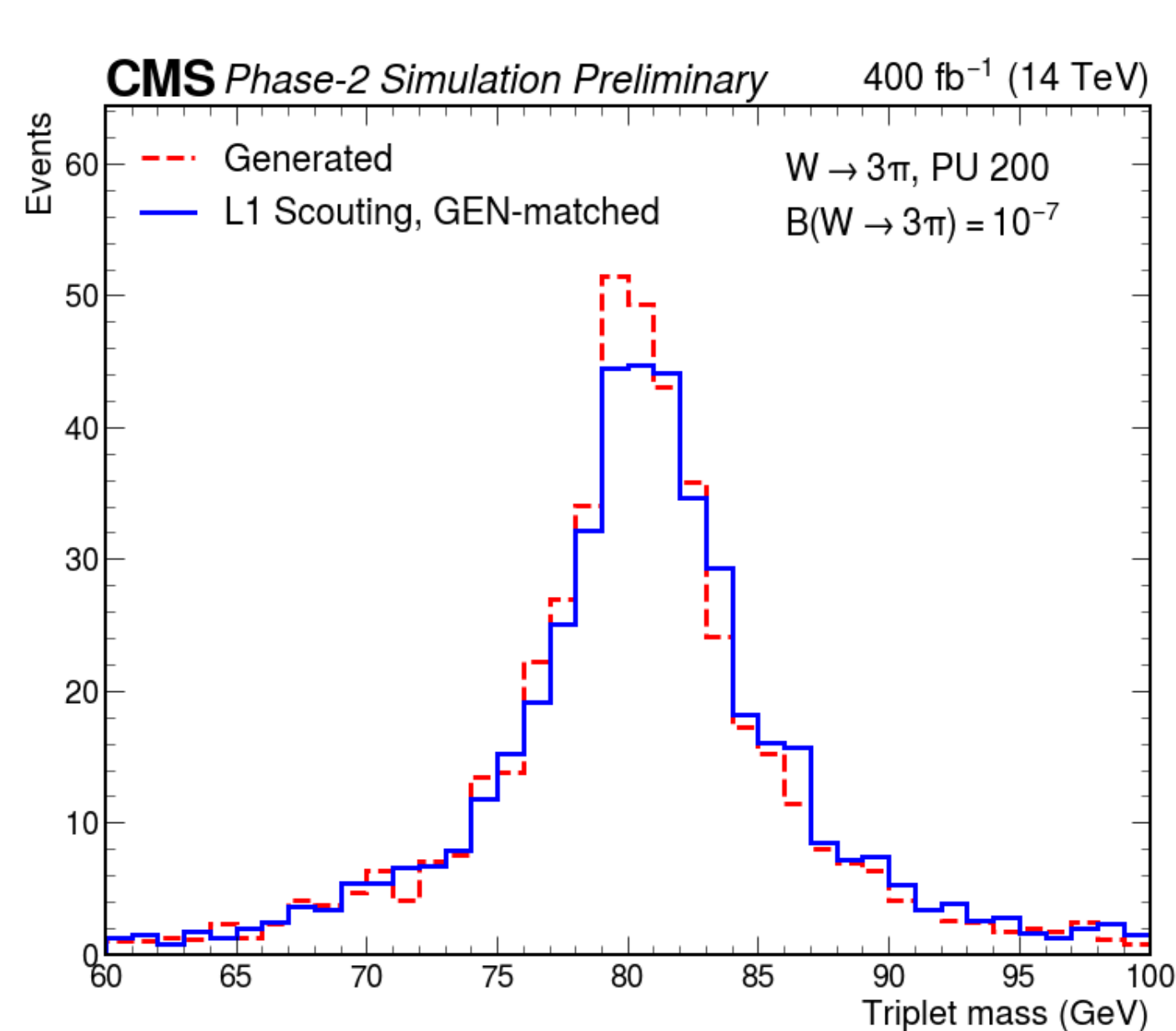
40 MHz Scouting: bypass the trigger completely



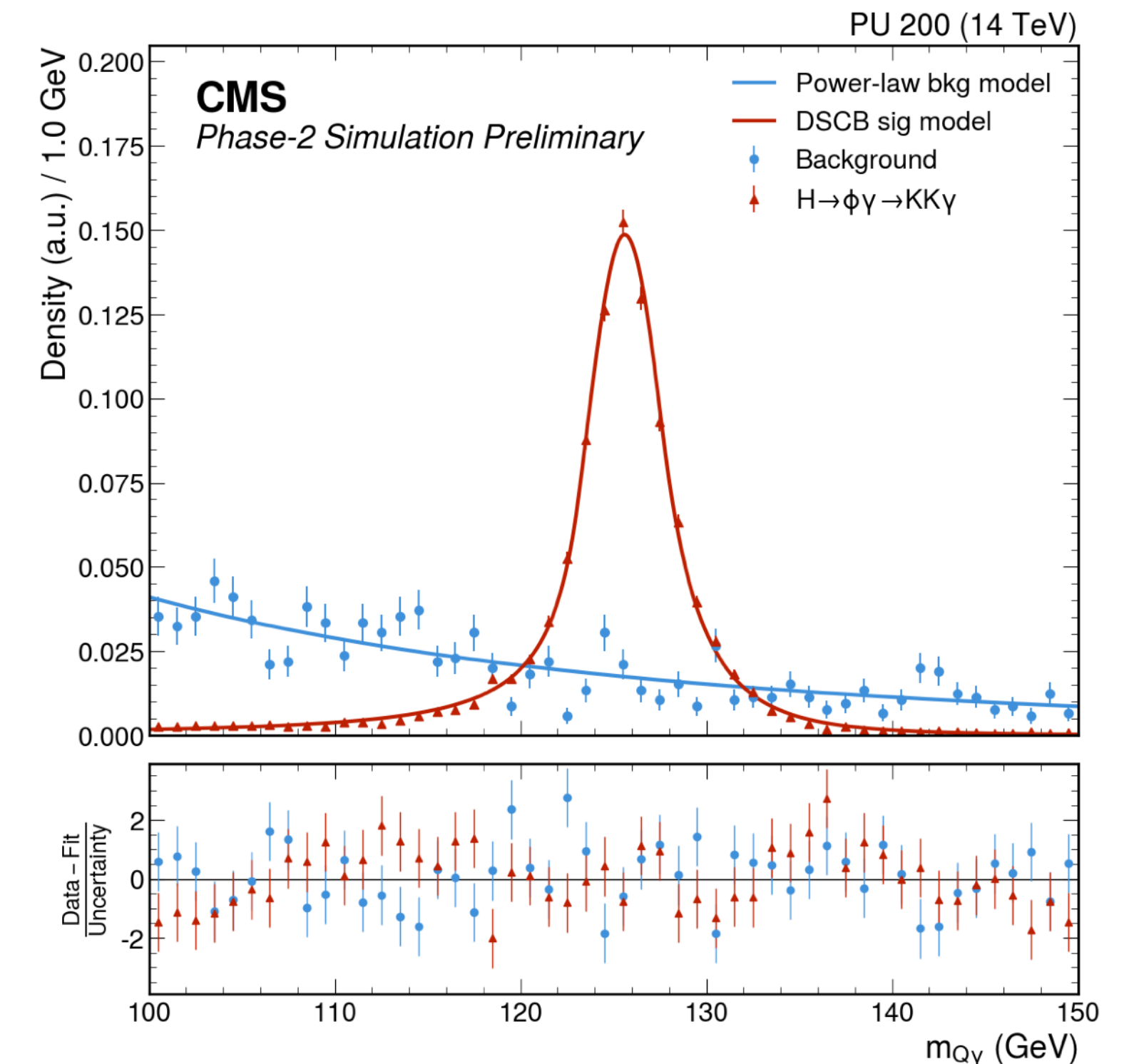
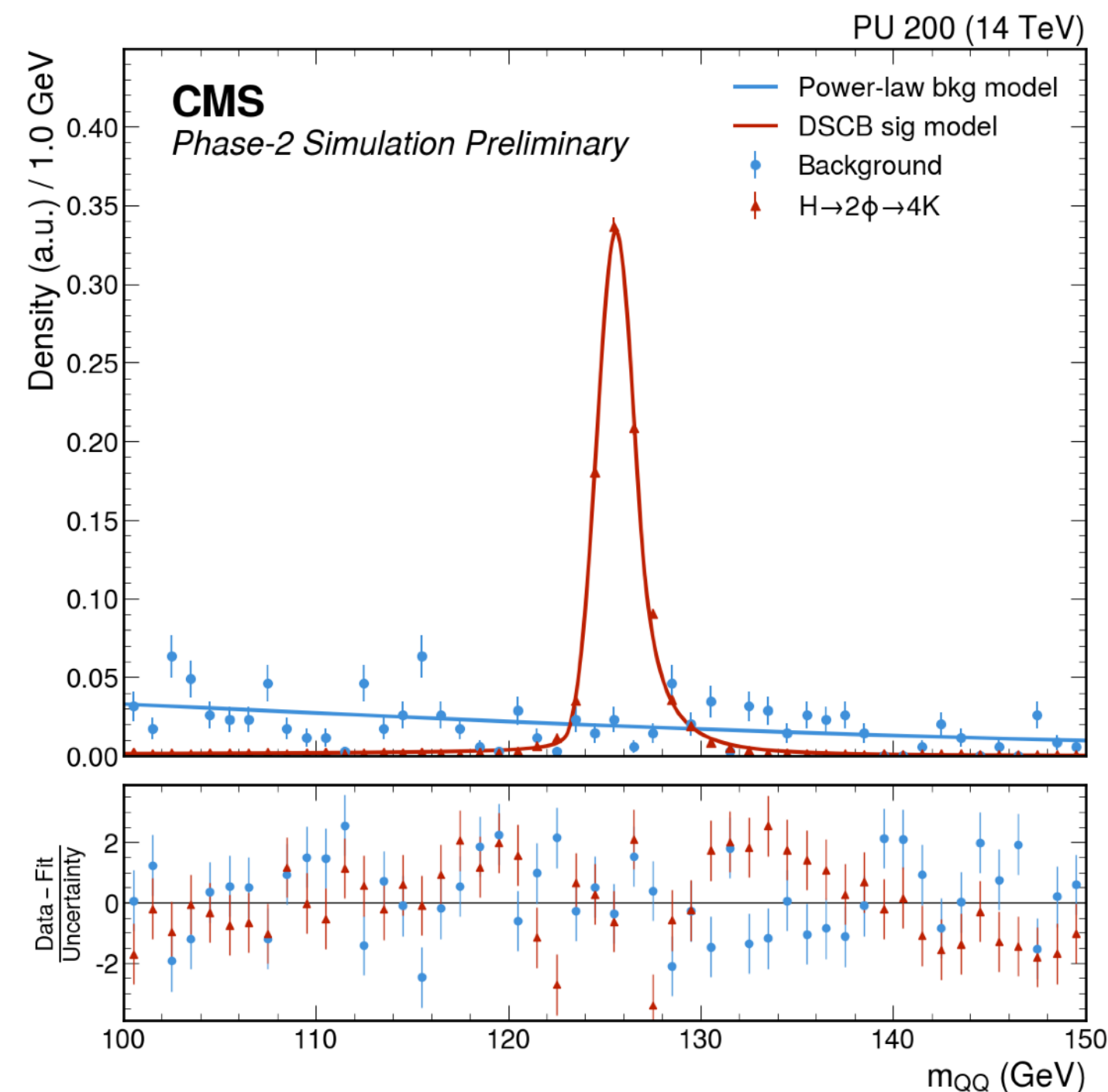
- The trigger selects & rejects events: lost events are never available for offline analysis
- The Level 1 Trigger processing computes lots information for every event to make its decision: particle properties
- 40 MHz scouting system will record and use the intermediate information for analysis
 - Lower quality event data than full offline analysis, but many more events

40 MHz Scouting use cases

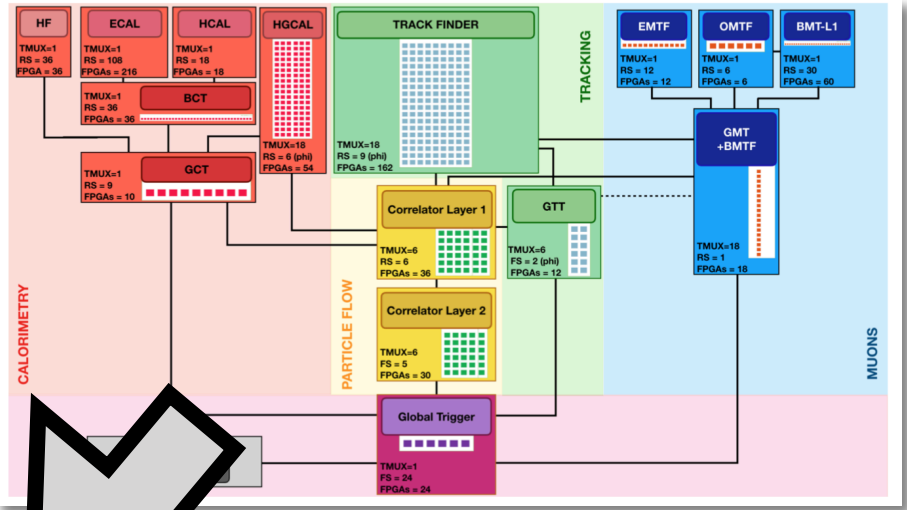
- Most trigger selections include a minimum transverse momentum requirement
- Some benchmarks have been studied: rare decays of W and Higgs bosons — $W \rightarrow 3\pi$, $H \rightarrow 2\phi \rightarrow 4K$, $H \rightarrow \phi\gamma \rightarrow KK\gamma$
 - Measure properties of known particles to probe the Standard Model
- All share: low momentum particles in the final state; rare; visible as a narrow peak on a smoothly falling background
- Next Generation Triggers is growing the set of physics processes to study with 40 MHz scouting



CMS-DP-2024-096

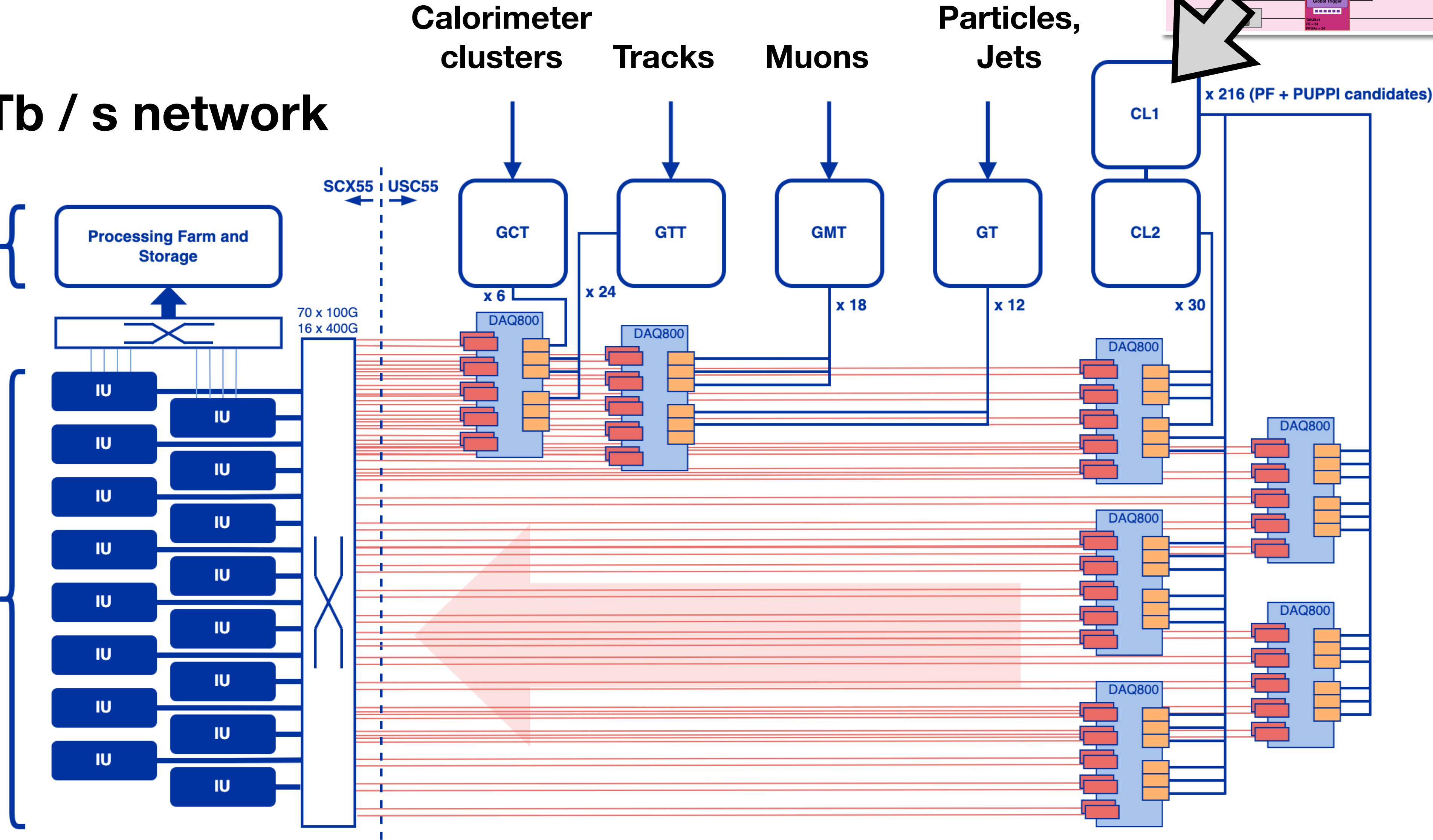


40 MHz Scouting System

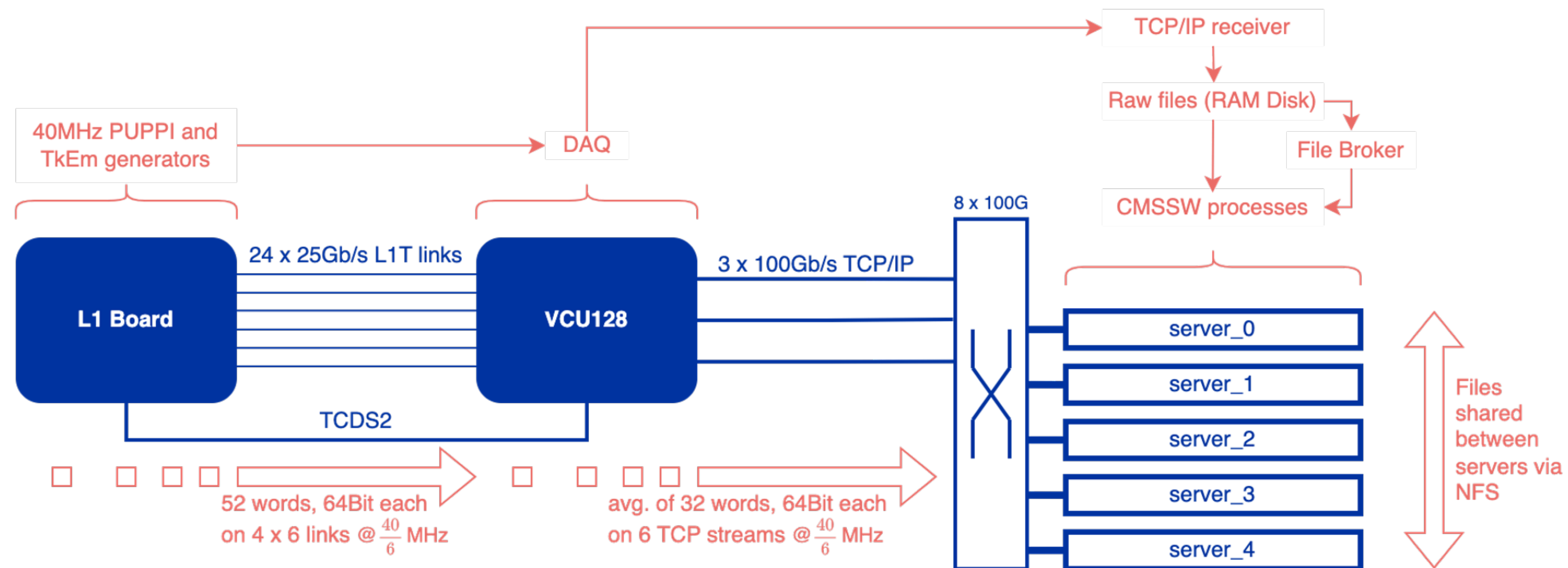


6-7 Tb / s network

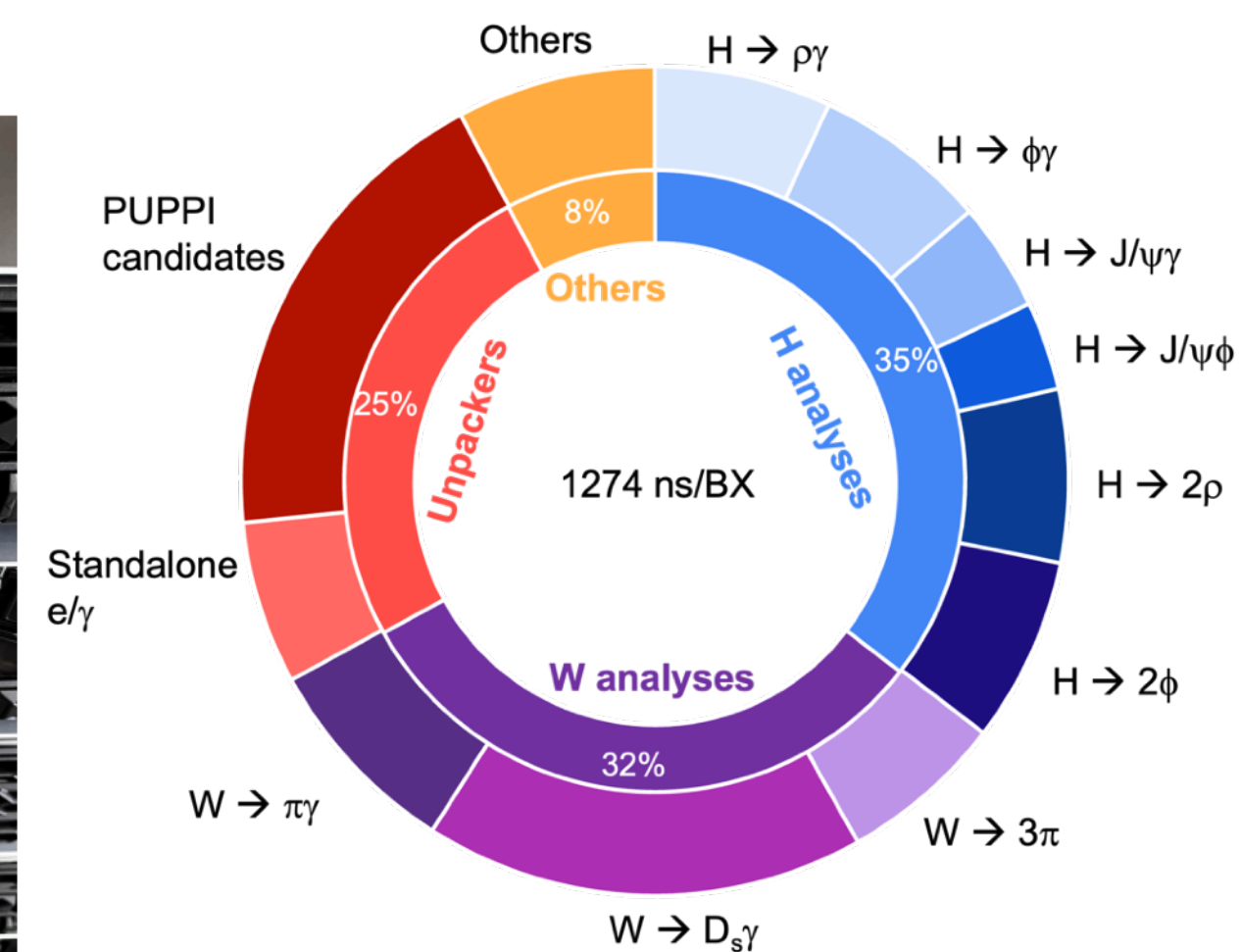
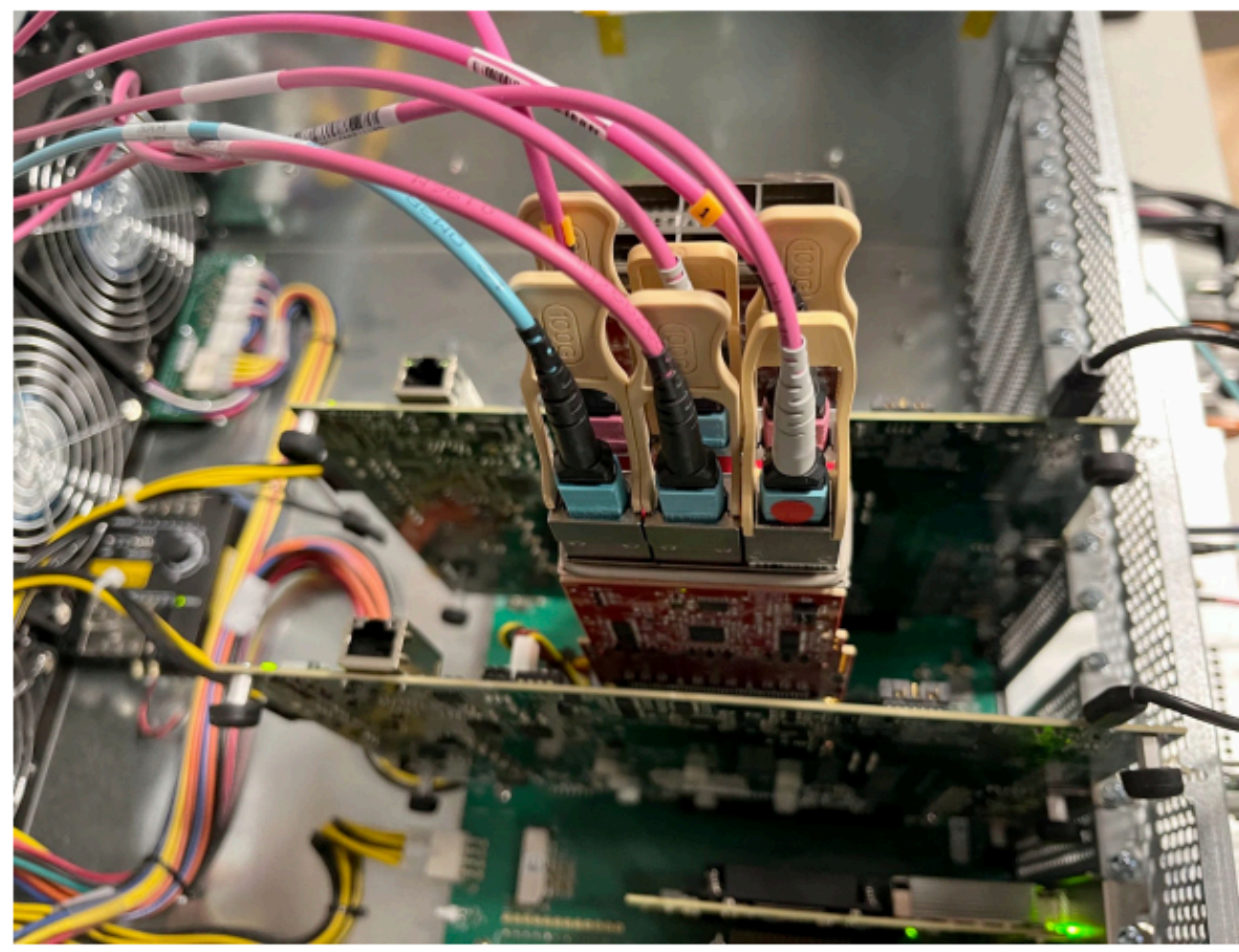
- 3. Distributed **online processing** creates reduced data stream for analysis
- 2. Ingestion units (IUs) **aggregate data** into orbits (orbit builder) and buffer for processing



40 MHz Scouting prototyping



- Small scale system in a lab for prototyping
- L1 Board mimics data coming from CMS
- Data Acquisition system prototype receives data and runs the analyses



Summary

- The CMS experiment will undergo a major upgrade for the High Luminosity LHC
- The trigger system will be completely replaced to keep up with the increasing data
- Next Generation Triggers project aims to extract the most physics out of the extra data
 - Using Machine Learning in Level 1 Trigger to enhance event selections and processing
 - Expanding the 40 MHz scouting system to do analysis without the trigger altogether!
- Using ML in the trigger requires sophisticated techniques
 - Strict constraints (low latency, high throughput, low area, low power) and often highly custom compute platforms
 - Projects like **hls4ml** and **conifer** aim to lower the barrier to entry for deployment of ML