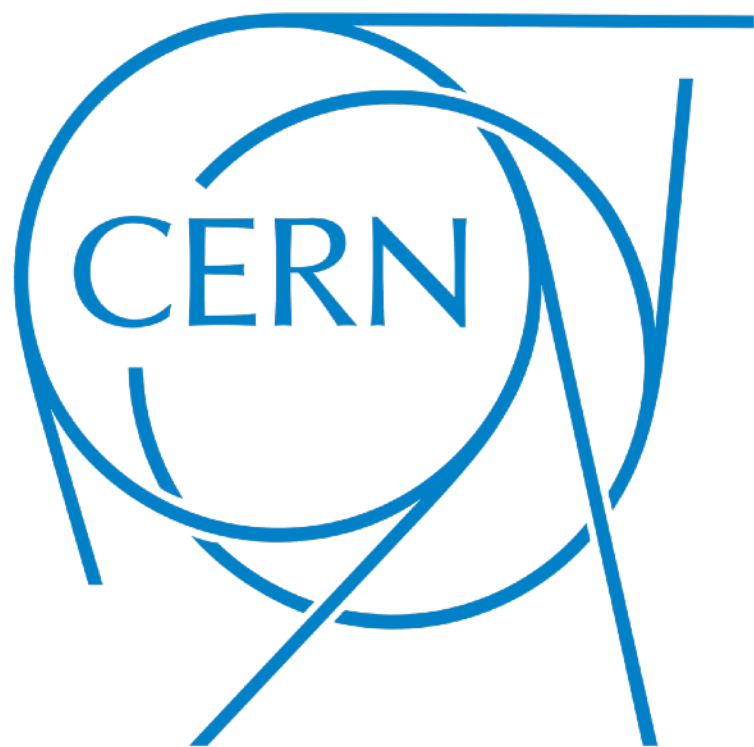


Edge ML in Particle Physics: Bringing Inference to the Detector

Sioni Summers

ML4Jets

17th September 2026



Introduction

- In the “big science” era of particle physics detectors, Machine Learning close to the sensor plays an important role in recording the relevant data while keeping backgrounds suppressed
- More powerful trigger chips, longer latency budget, improved software enabling us to bring more ML to the detector
 - For CMS Phase 2 I can account for 25 billion inferences / s from ML models
- I’ll present what we’re doing with Fast ML to improve triggering, how we do it, and where we’re going next
 - There’s a lot of innovation in methods and architectures suitable for hardware deployment
 - I’m focussing on methods and applications that are being used for data taking right now or at HL-LHC

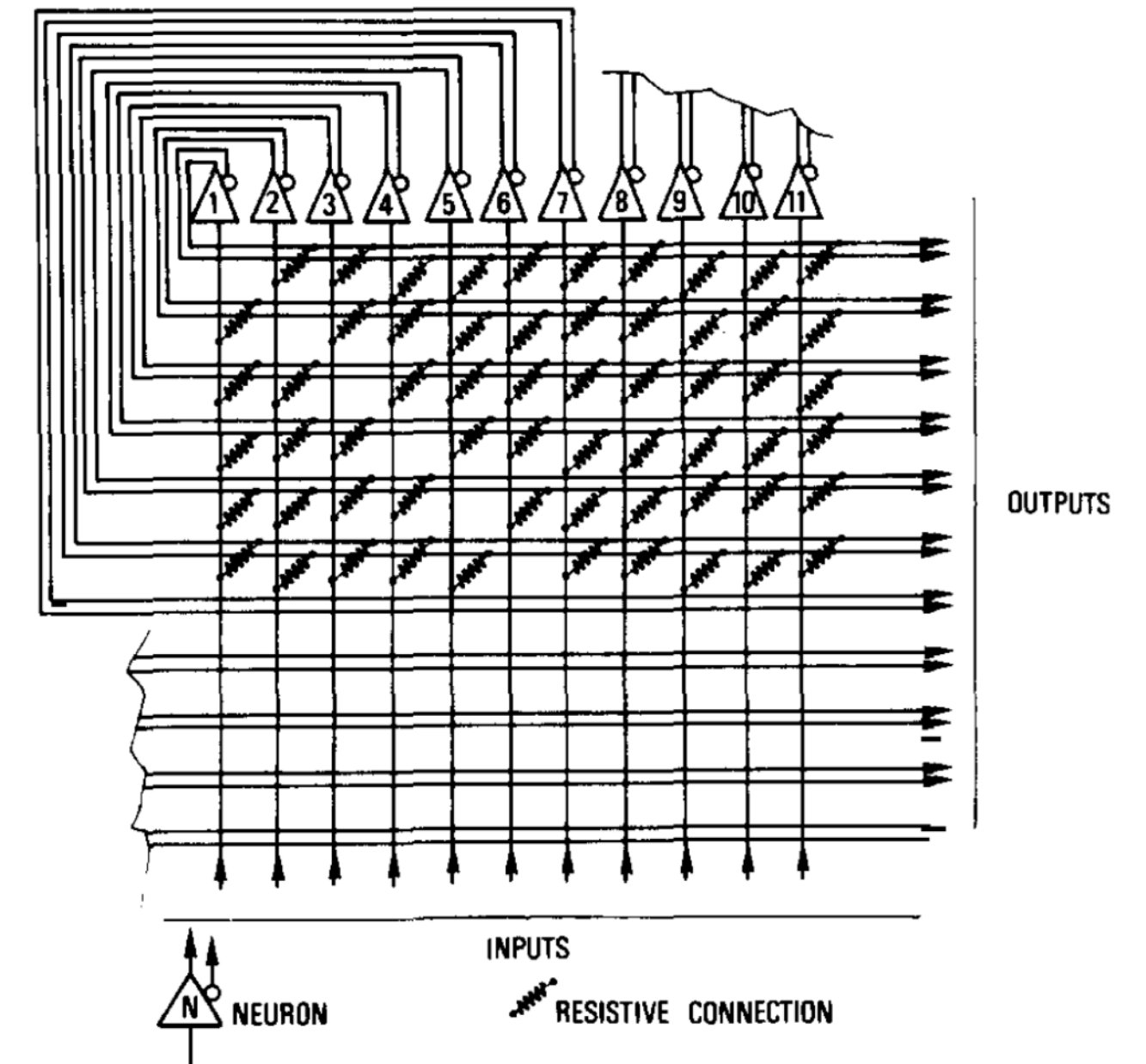
 sioni@cern.ch
 sioni.web.cern.ch
 @thesps
 @ssummers

History of ML in Particle Detectors

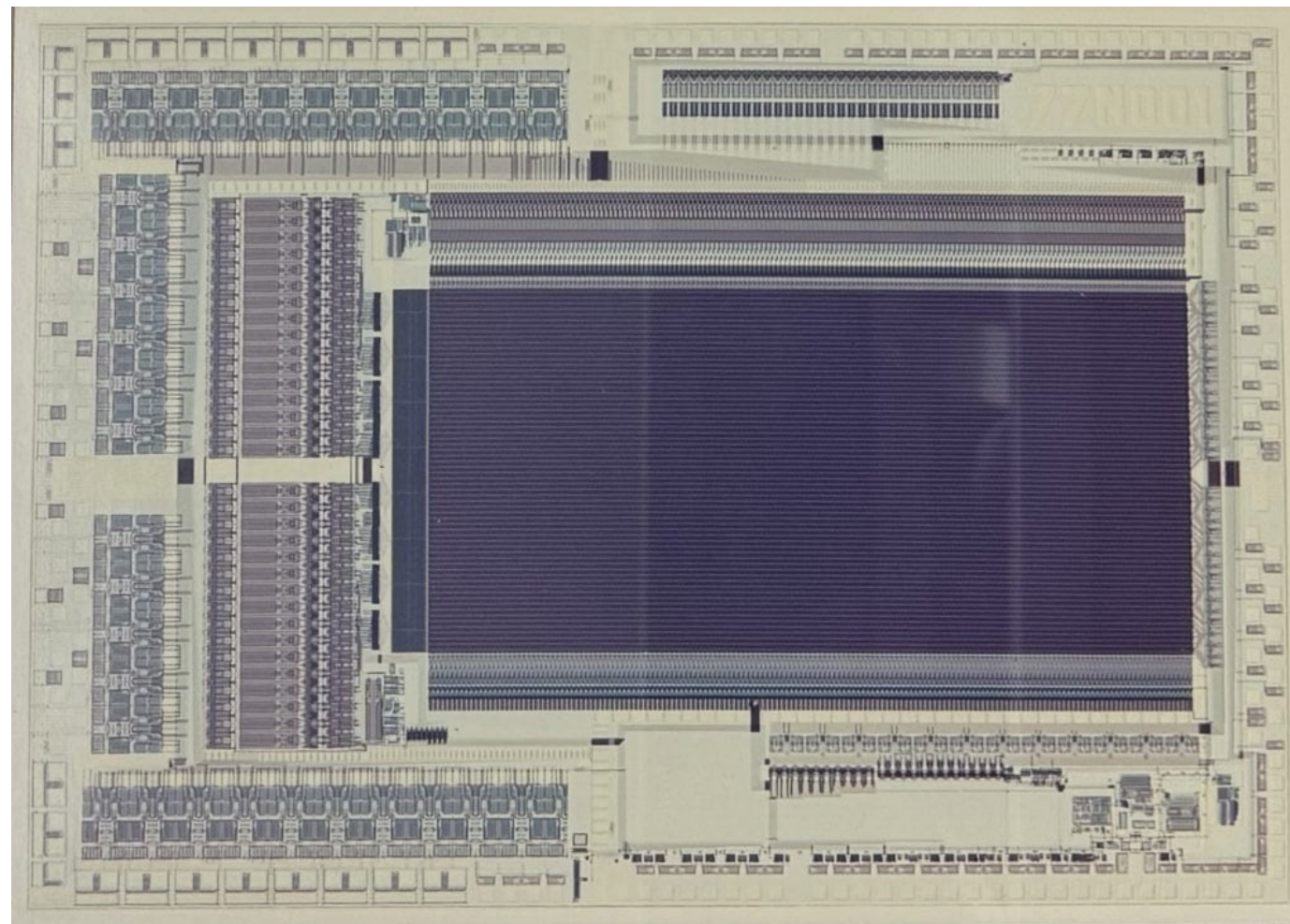
- Particle Physics community was already talking about Edge at the start of interest in Machine Learning
 - Then it was analog computing doing dot products with Ohm's law

It should also be possible, using existing technology, to build simplified [neural] networks that will allow detailed reconstructed event information to be used in **fast trigger decisions.**

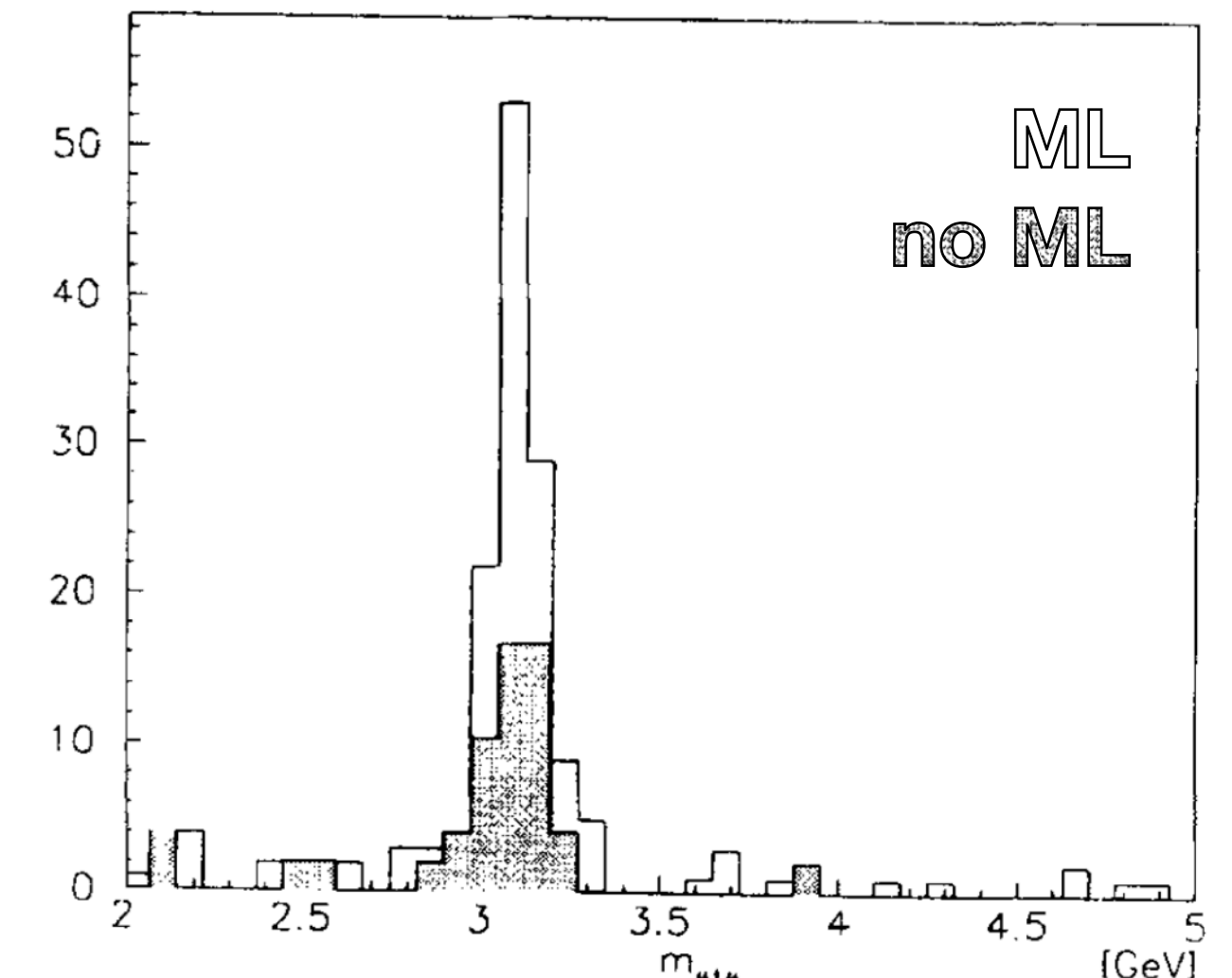
Bruce Denby, Neural networks and cellular automata in experimental high energy physics, 1988



computerhistory.org

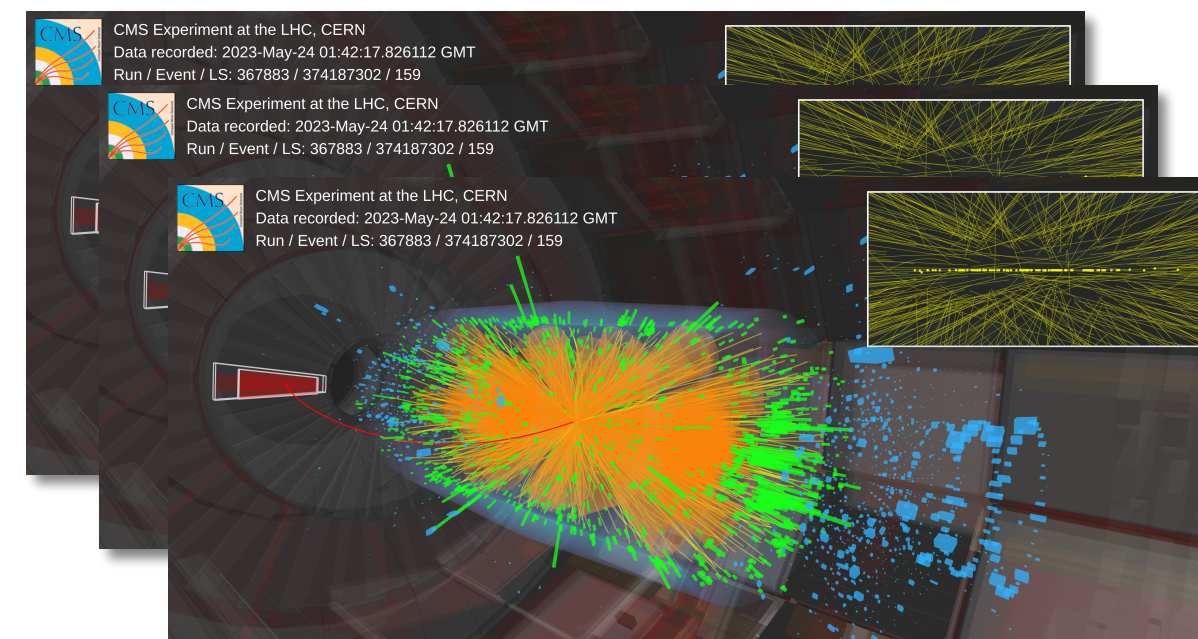
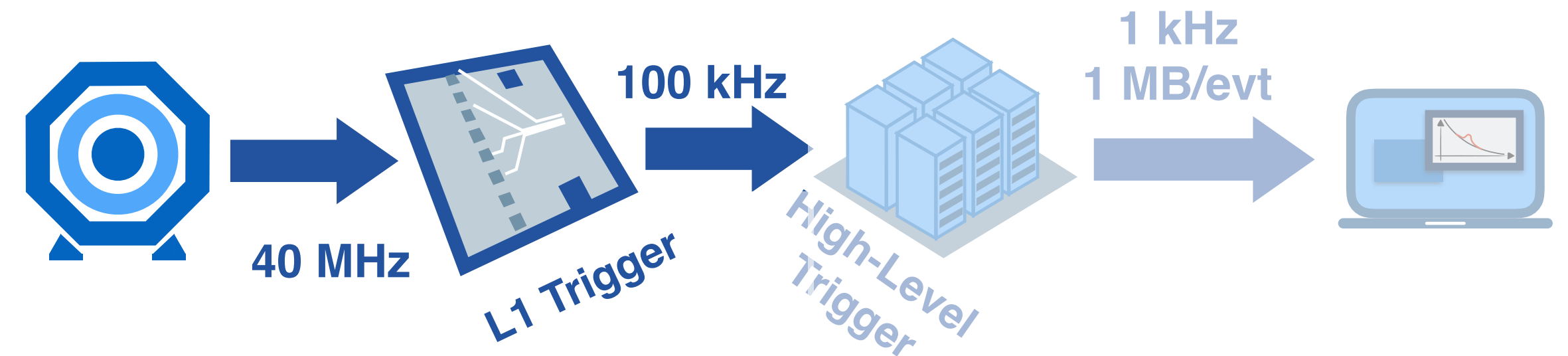


- 1992 Intel ETANN = Electronically Trainable Analog Neural Network
 - 5 μ s latency for Multi Layer Perceptron of 64-64-64
- 1996 H1 at HERA: Neural Networks in trigger taking data
 - Plot shows improvement in $m_{\mu\mu}$ from use of NN trigger



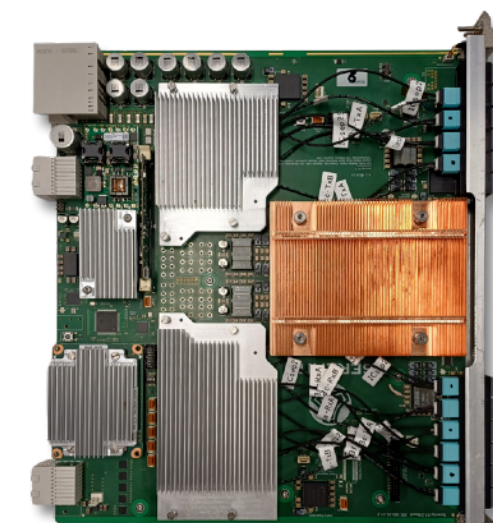
Trigger at LHC

- ATLAS and CMS use first stage hardware triggers
 - Underground, close to the experiment
- A trigger system is a big binary classifier system
 - Keep (1%) or reject (99%) collision events
 - CMS Phase 2 (2030) will have ~500 FPGA boards
 - Performs reconstruction, then selection against a “menu”
- We have started using Machine Learning to make better selections
 - At CMS we expect 25 billion inferences / s by ML models in our Phase 2 upgrade for HL-LHC
- Due to FPGAs and tight latency constraints, we’ve developed our own solutions to deploy ML

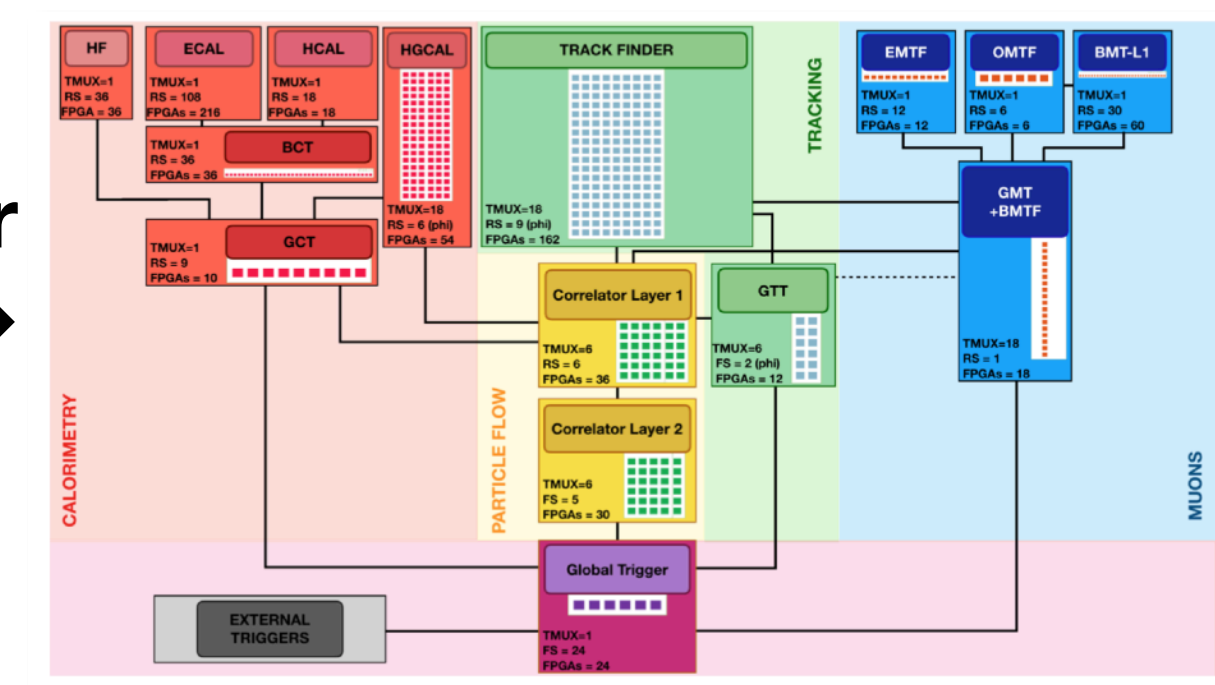


← 40 million collisions per second

12.5 μ s latency for each collision →



~500 FPGAs



99% rejection factor →



What are FPGAs?

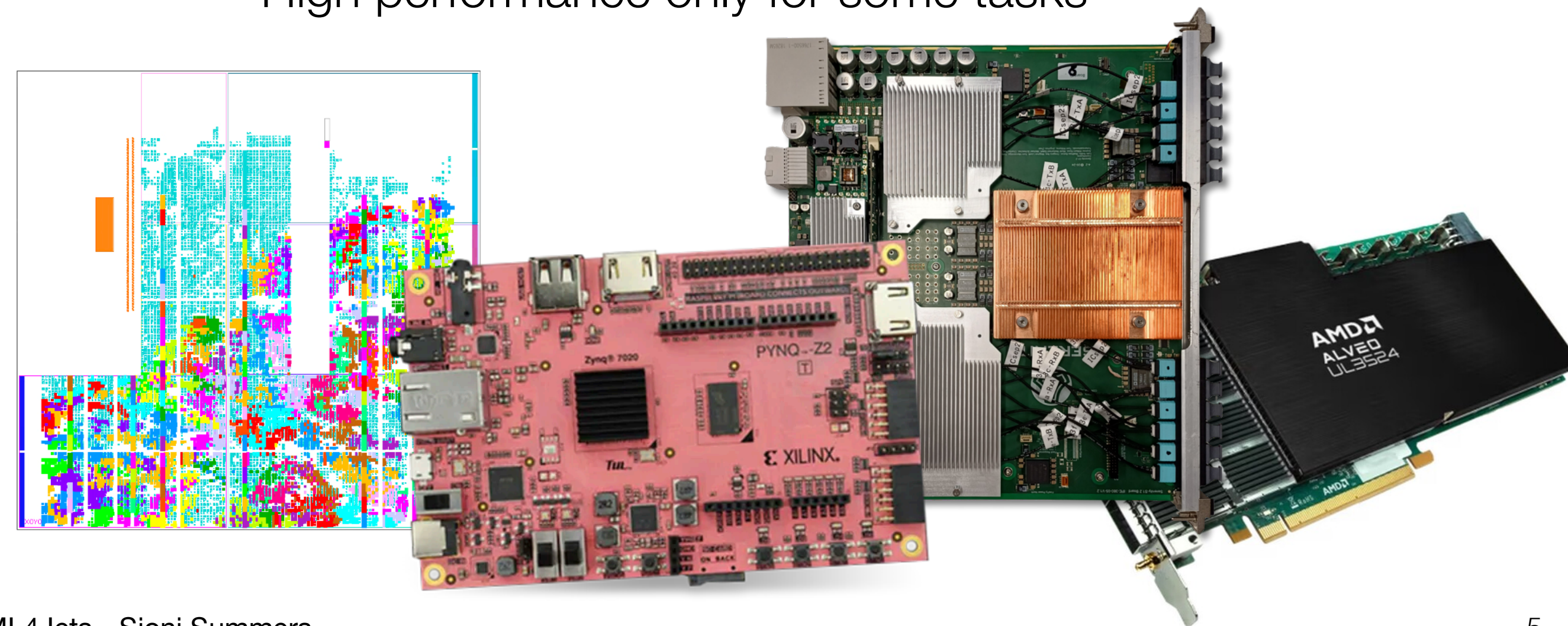
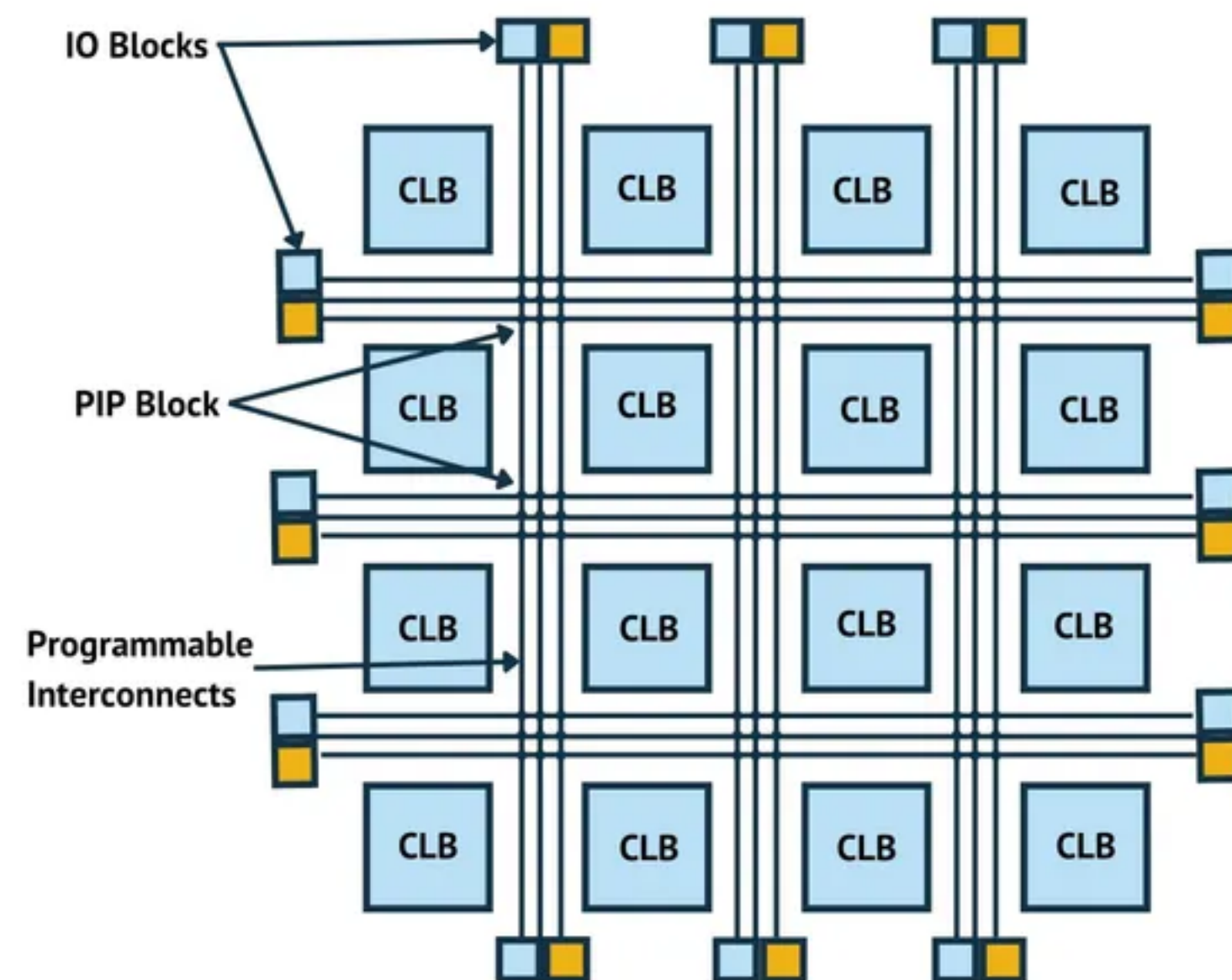
- FPGA stands for “Field Programmable Gate Array”
 - Reconfigurable circuits → like LEGO computers with building blocks
 - Lots of “resources” that are configured and connected for your specific task → you describe the *hardware* with code
- Originally used to prototype ASICs before printing, now used for computing too

- Advantages:

- Massively parallel computation thanks to many logic blocks: low latency; high throughput
- Hugely customisable; massive connectivity and in/out bandwidth; reconfigurable
- Available in radiation tolerant variants

- Disadvantages:

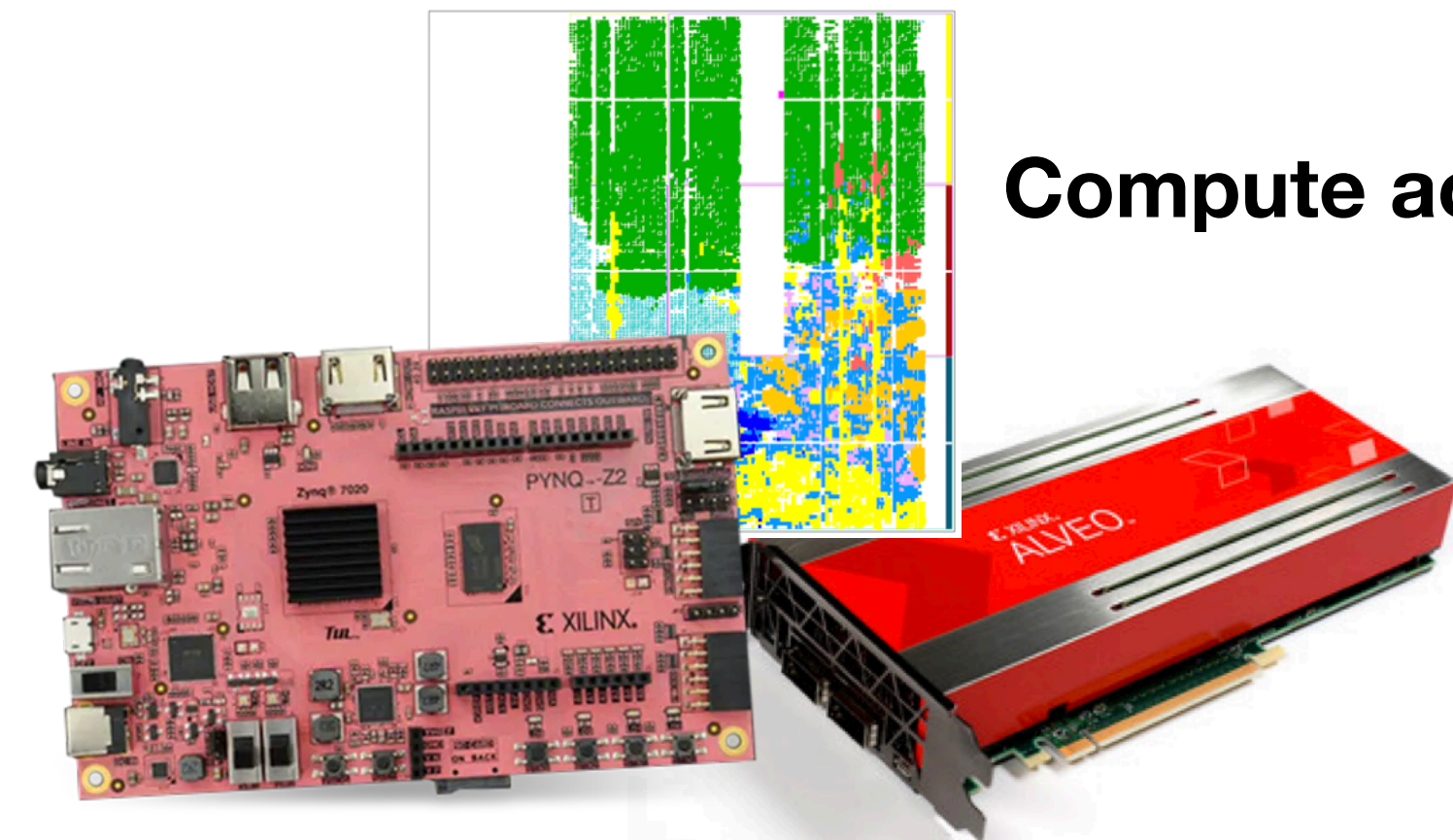
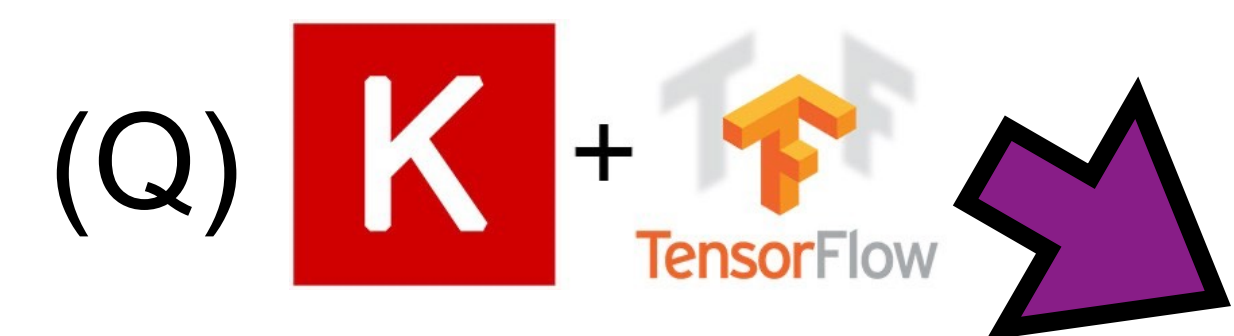
- High cost - much more than microcontrollers
- Complexity - high development effort
- High performance only for some tasks



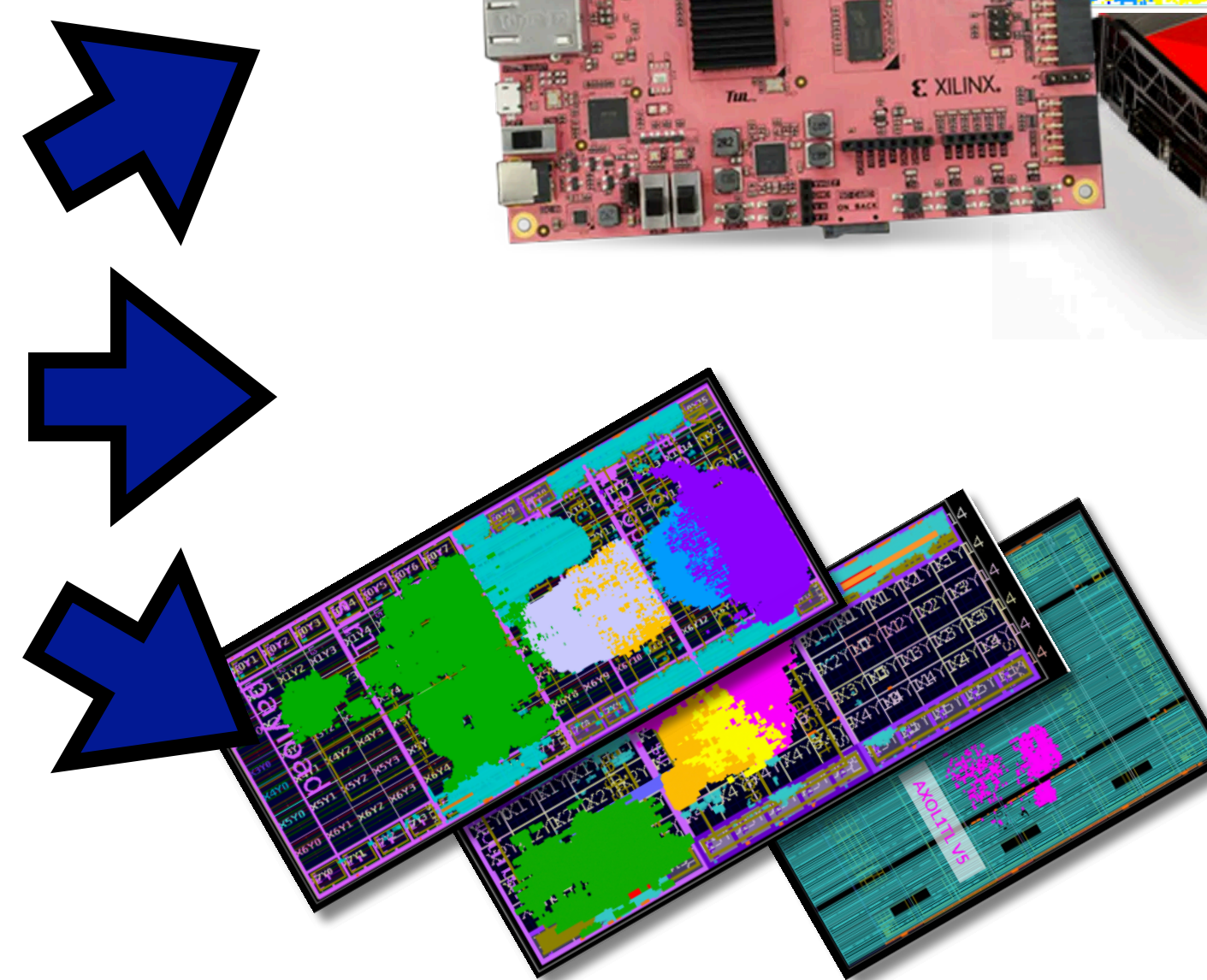


high level synthesis for machine learning

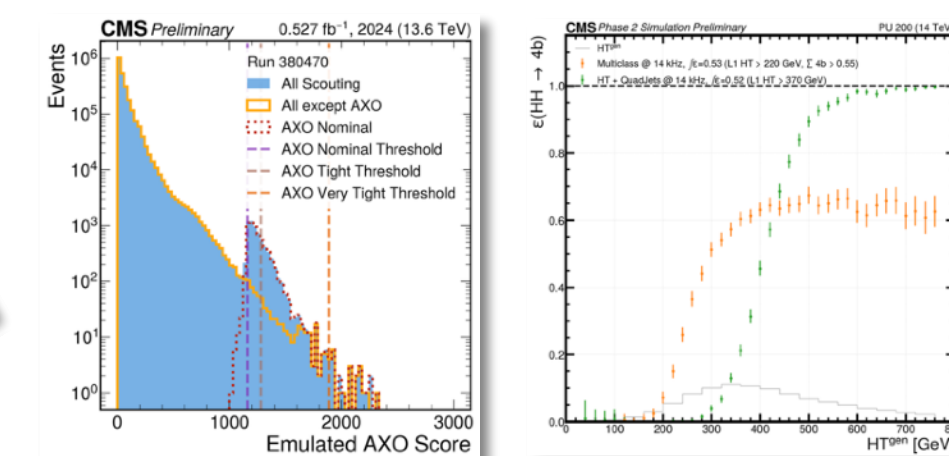
fastmachinelearning.org/hls4ml



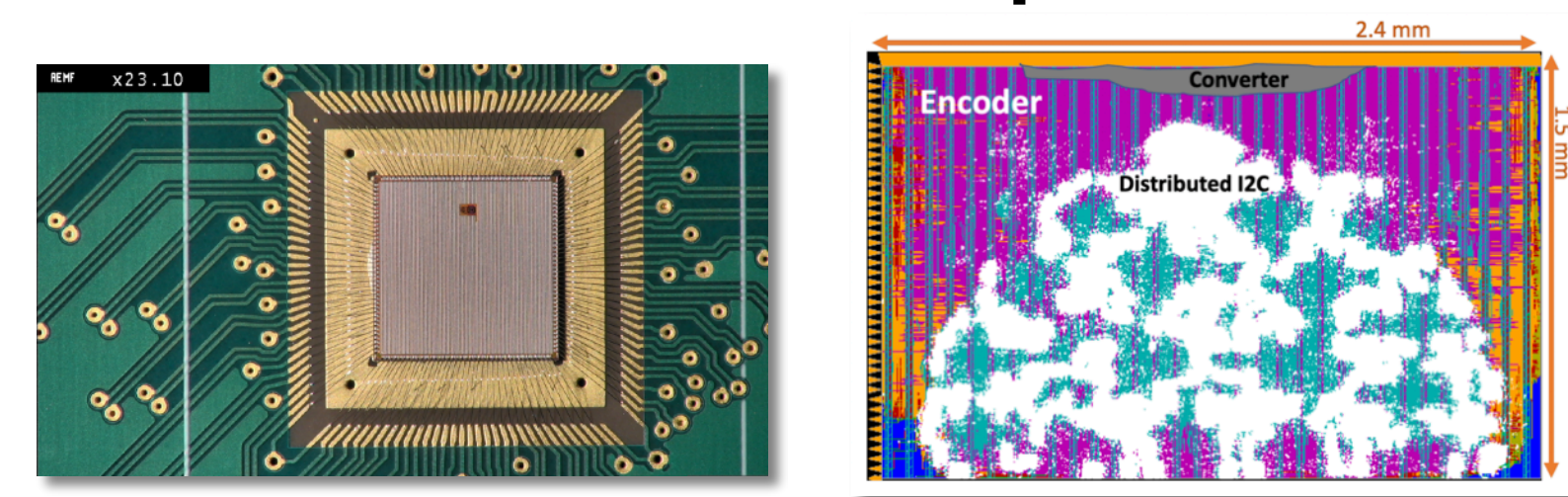
Compute acceleration



Triggering for physics

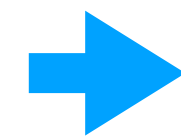


AI chips for frontends



hls4ml - Dataflow Architecture

- Dataflow architecture: each Neural Network layer is an independent compute unit
 - Independently tuneable parallelism, quantization and pruning
- Fully on-chip: great for high performance and power efficiency, but NN size is limited by available resources
- Example: small CNN trained on handwritten digits (MNIST)



Conv2D

ReLU

MaxPool2D

Conv2D

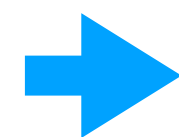
ReLU

MaxPool2D

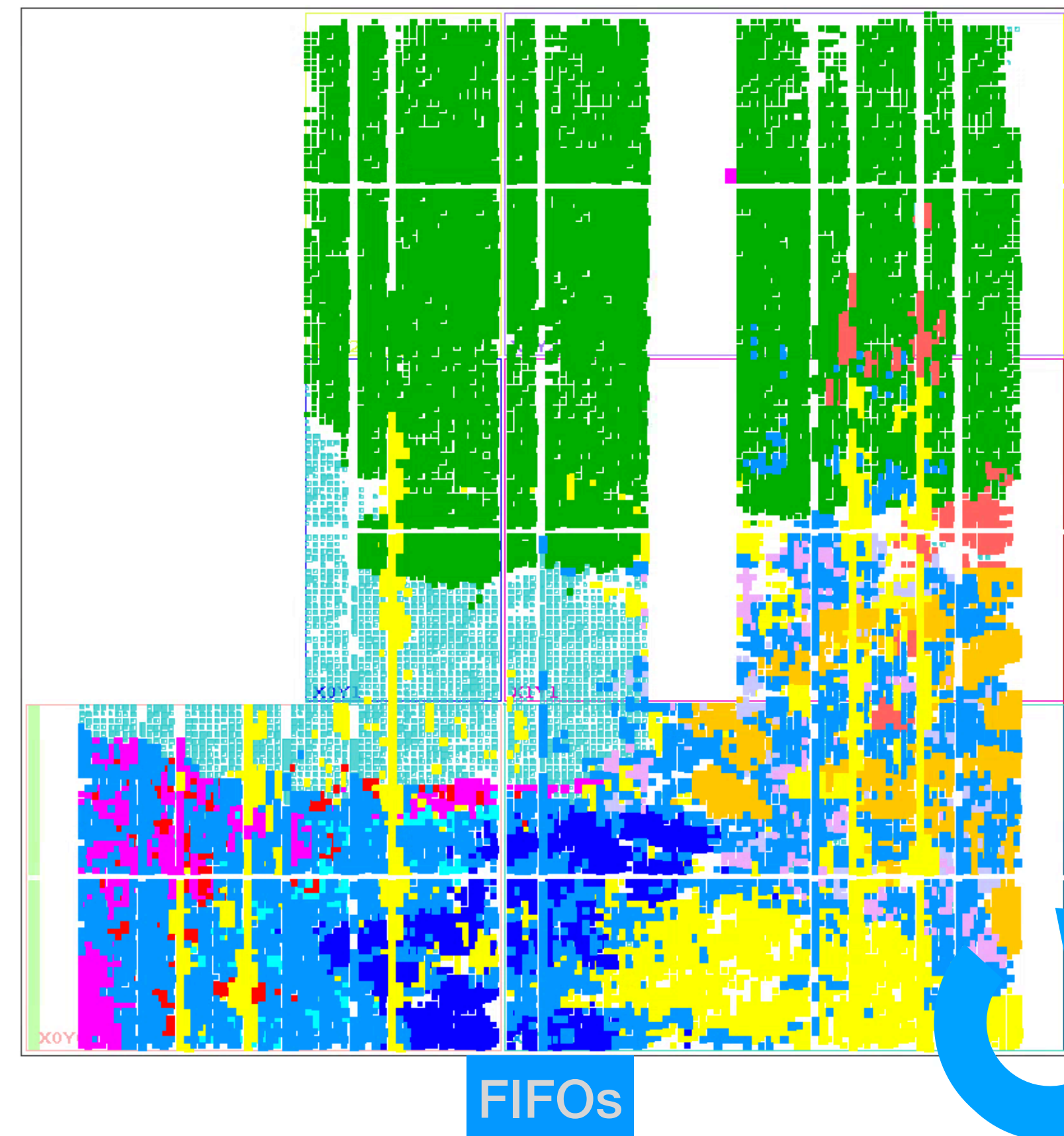
Flatten

Dense

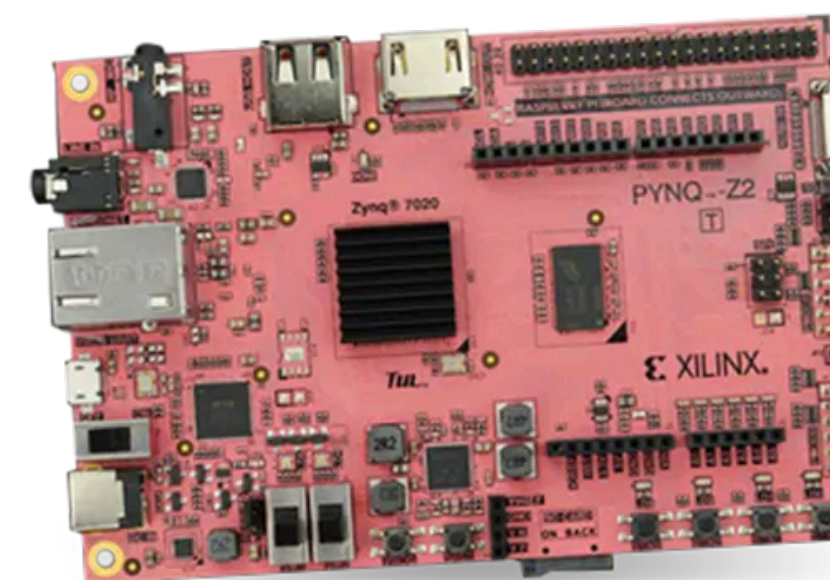
Softmax



Prediction

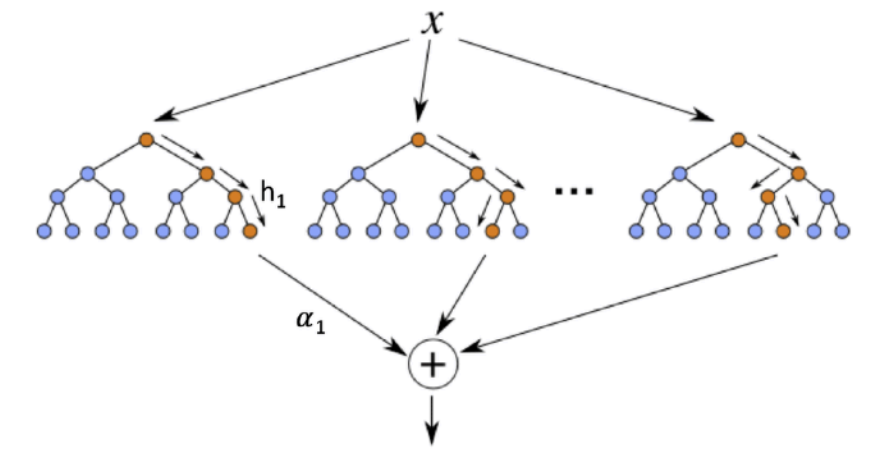


FIFOs



conifer for Decision Forests

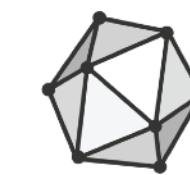
- The old ways still work well for edge-constrained ML: **conifer** is for Decision Forests to FPGA (i.e. BDT, Random Forest)
- Frontend support for popular BDT training libraries
- Backends: HLS, (hand-written) VHDL, Forest Processing Unit (FPU) for reconfigurable hardware
- Capable of inference at 10 ns latency, O(100) MHz throughput
- HLS and VHDL backends map one DF to one hardware implementation
 - Supporting classification, regression, anomaly detection (isolation forest)
- Surrogate models for rapid silicon-cost estimation
- Used in ATLAS data taking during 2025-2026
- [GitHub](#), [website](#), [paper](#), `pip install conifer`



Yandex
CatBoost



LightGBM



ONNX

dmlc

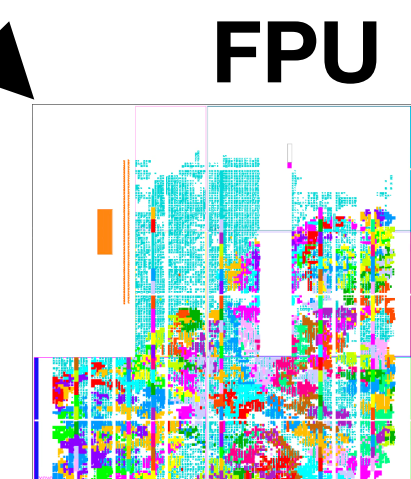
XGBoost



scikit
learn



YDF
Yggdrasil Decision Forests



Models for Edge ML

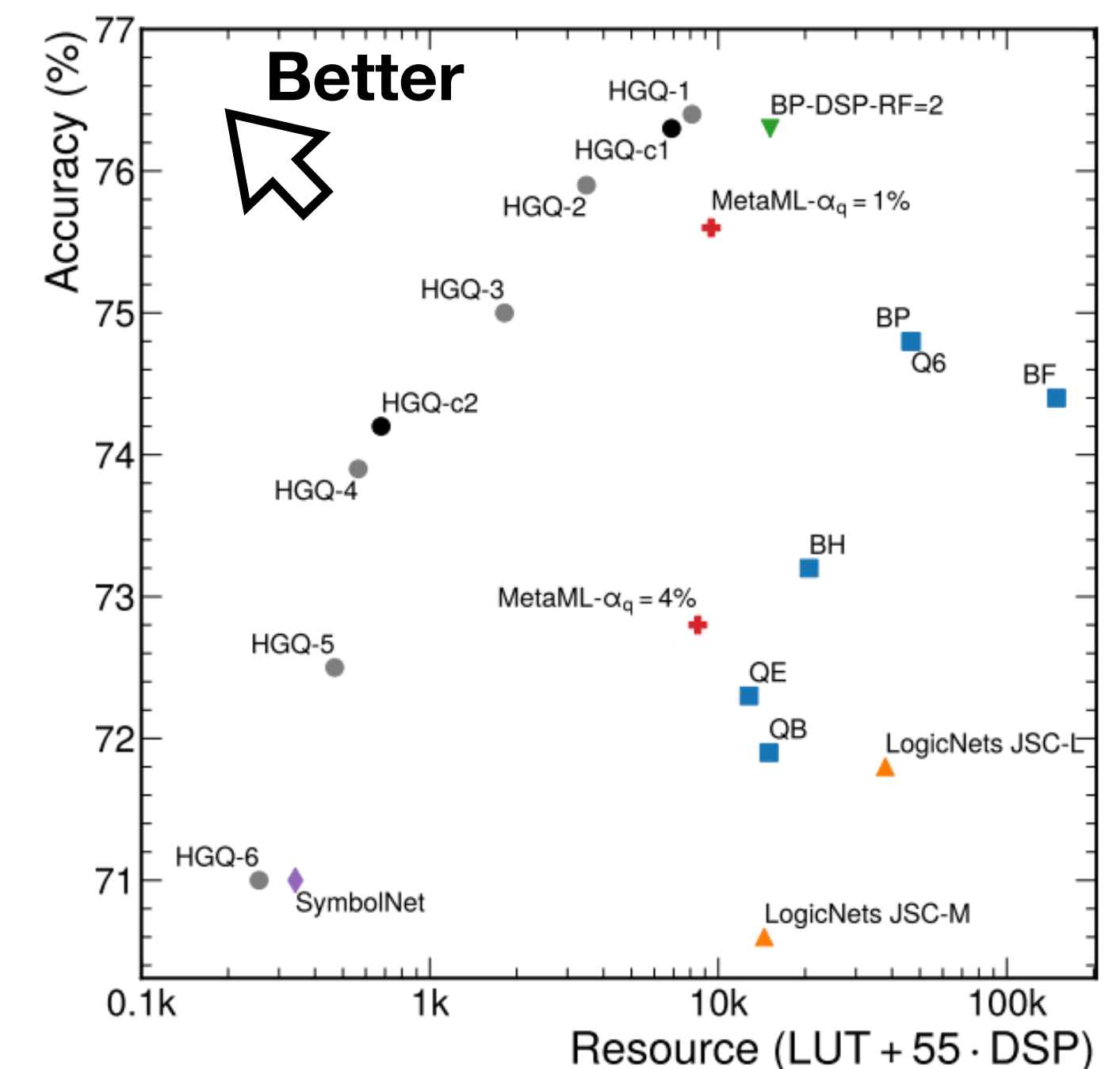
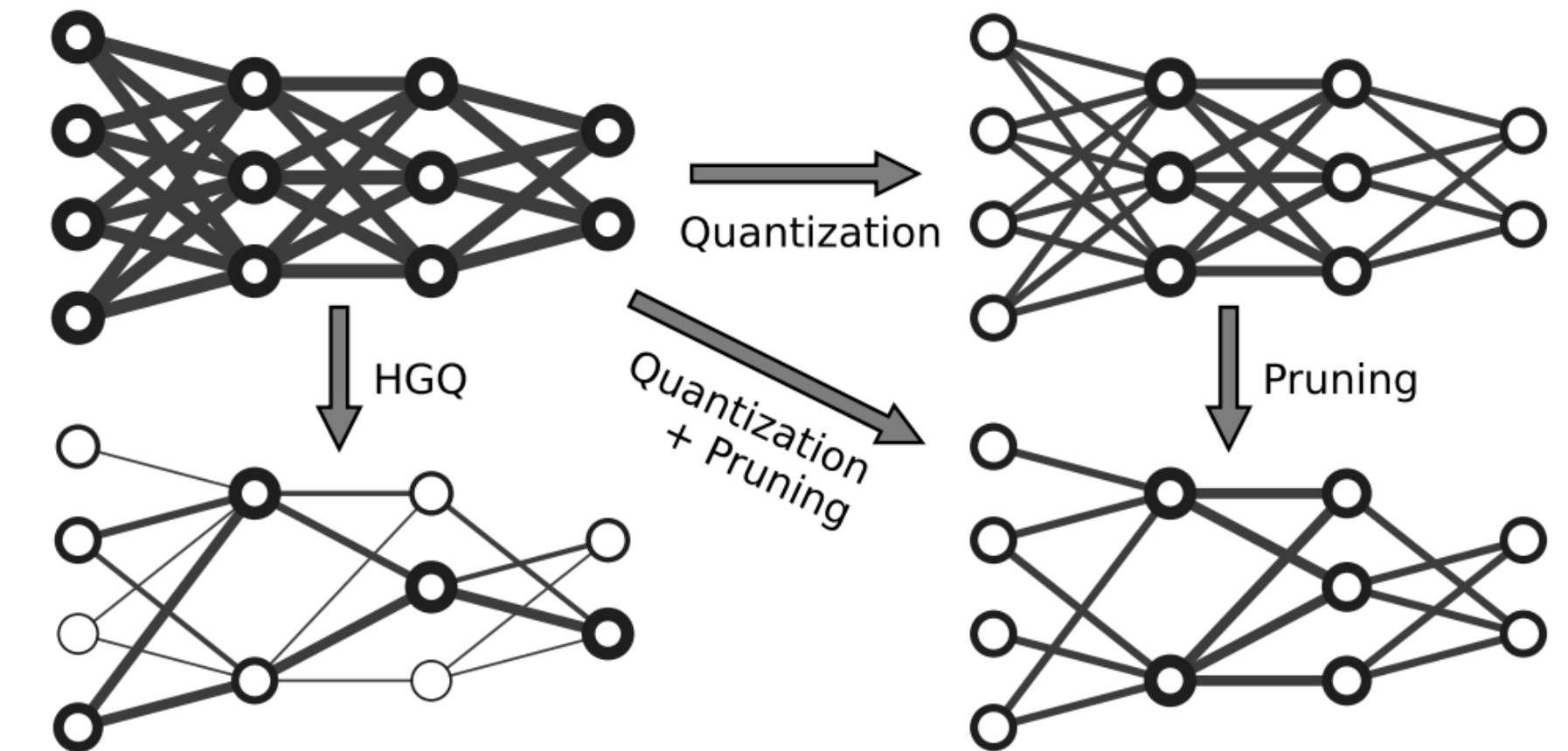
- For Edge ML we need small, simple models
 - Massively parallel unrolled hardware for ultra low latency
 - → uses a lot of silicon area
- Models with few parameters (1-10k)
 - Proxy for compute cost but number of operations is what really matters
- *Quantization aware training* is the main technique for shrinking models
 - Each multiply-accumulate operation is mapped on different hardware → every weight, bias and activation can be independently quantized
 - Heterogeneous & high granularity quantization
- See D. Sloot today, A. Petitjean tomorrow





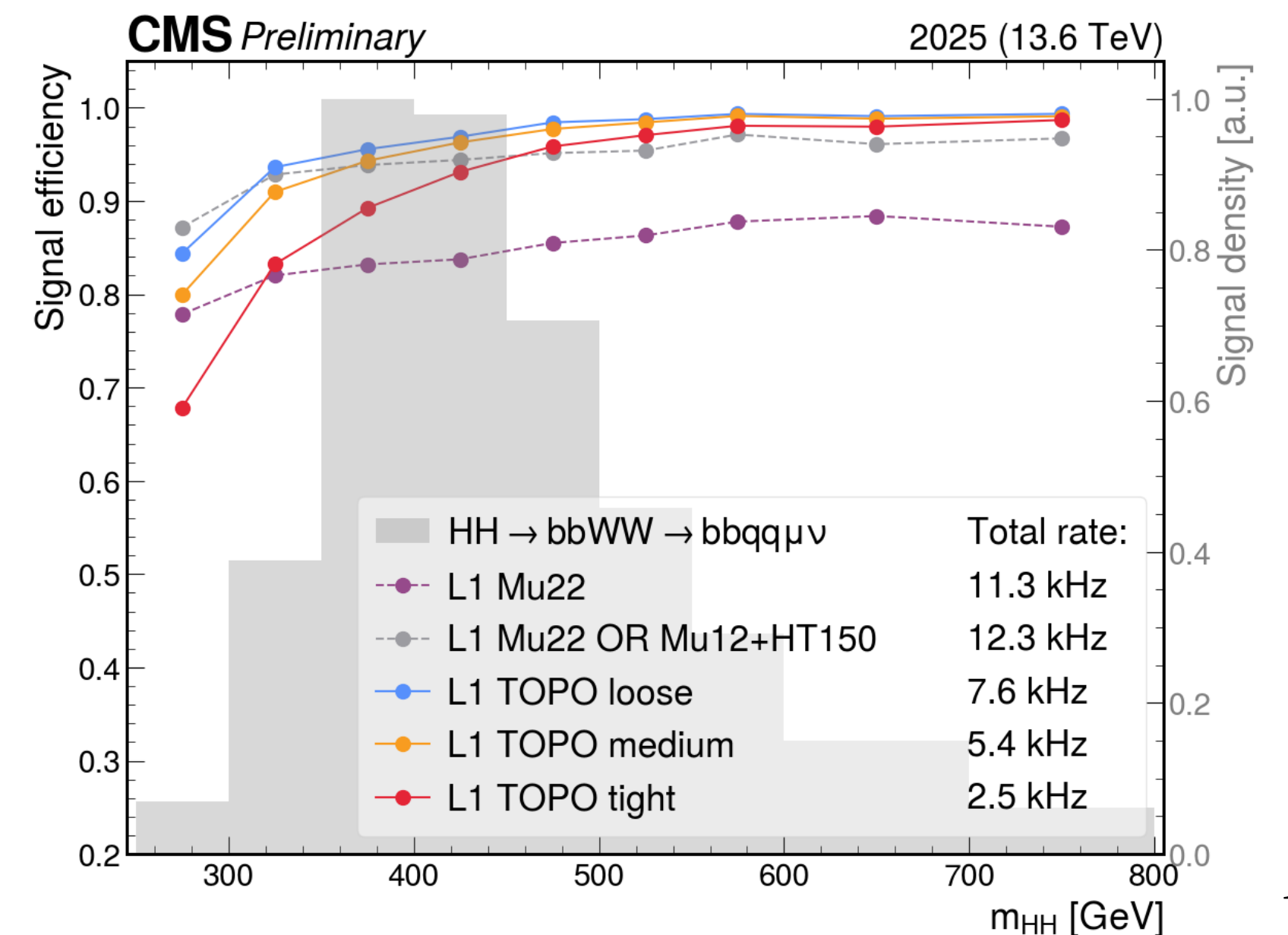
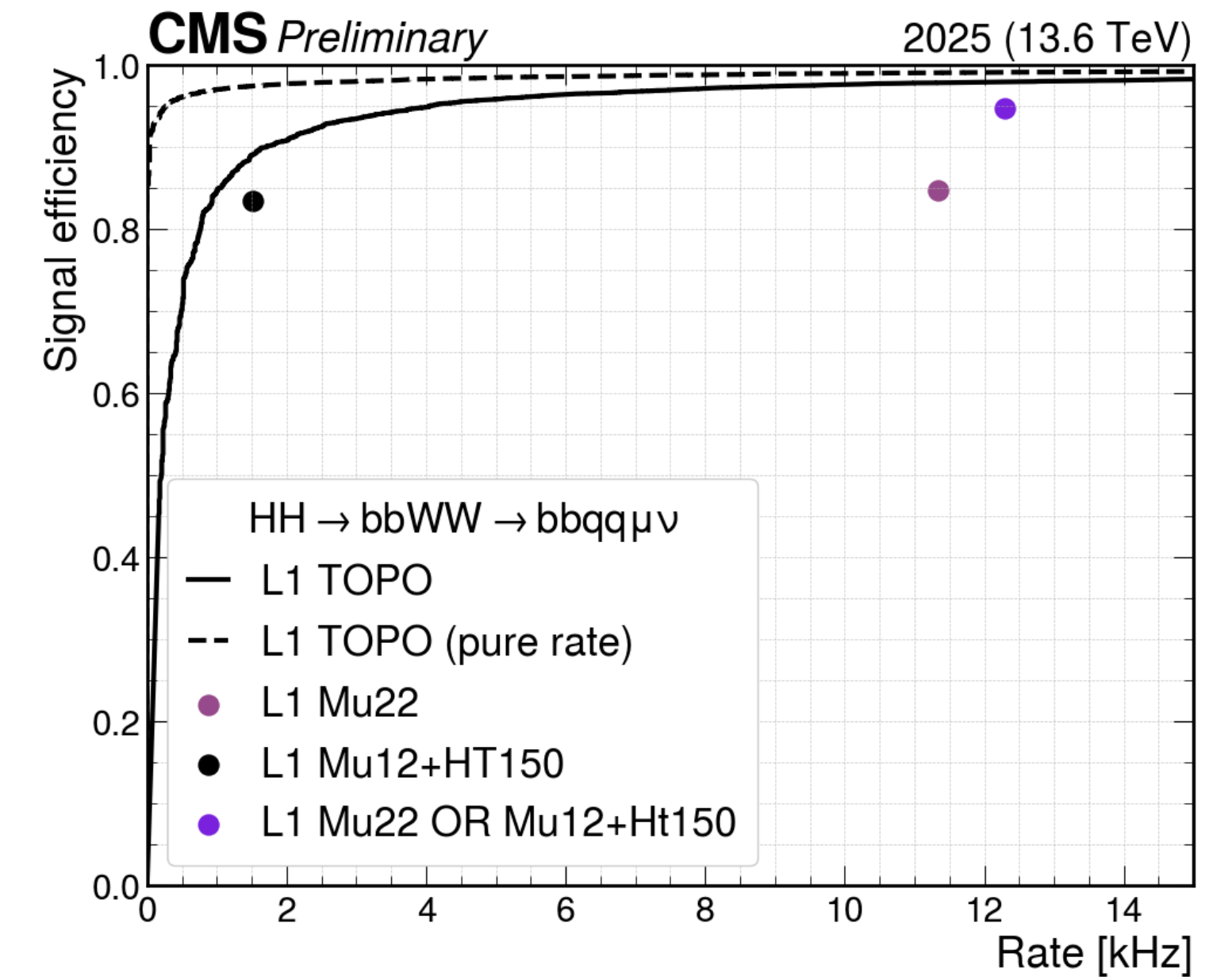
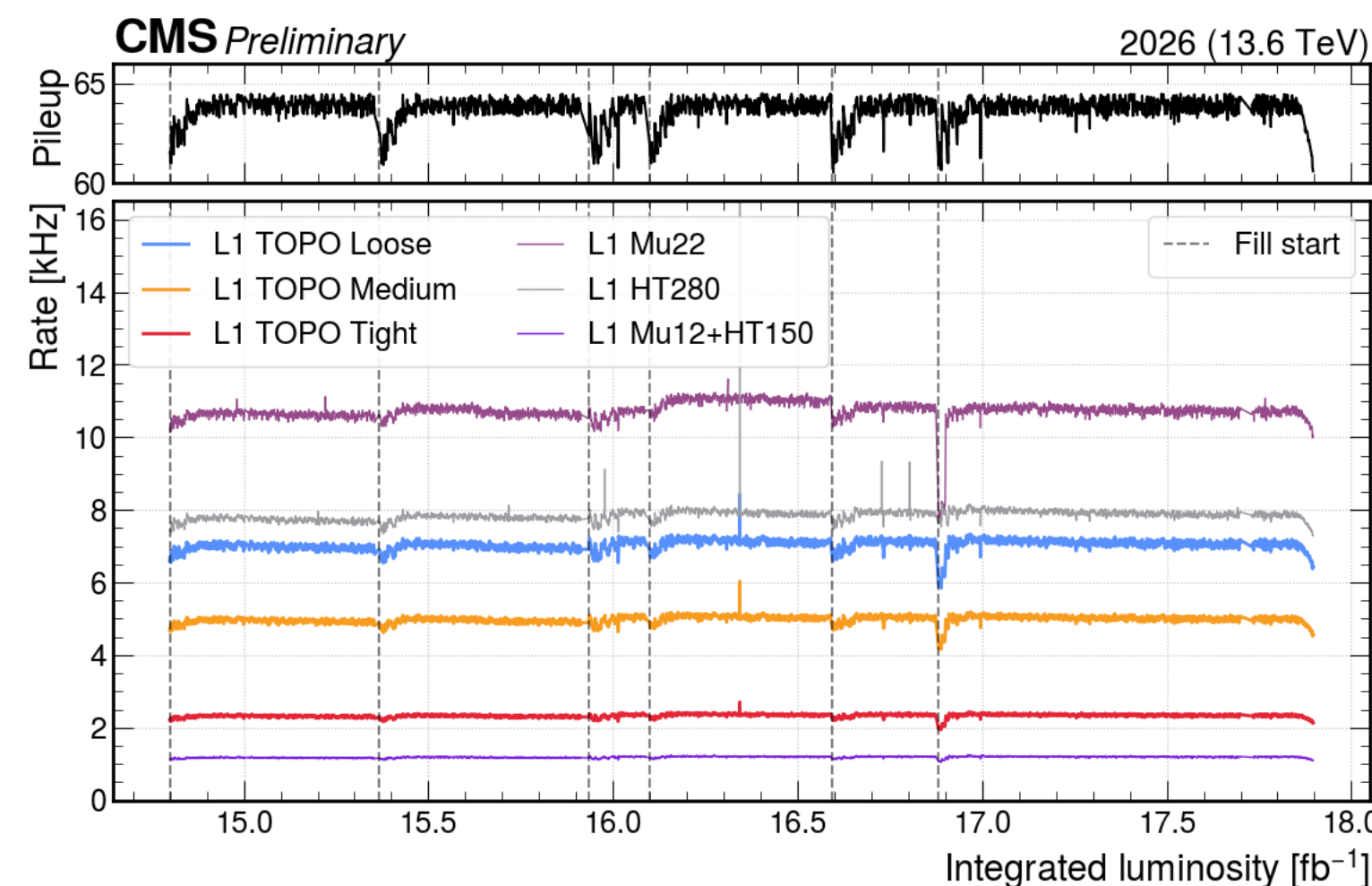
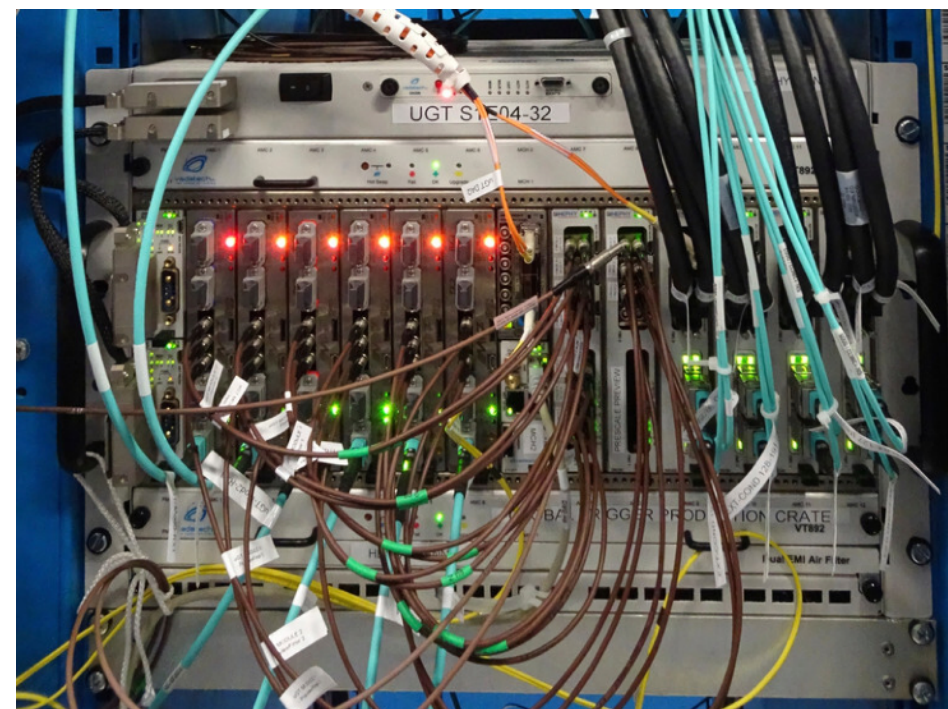
High Granularity Quantization

- QKeras introduced heterogeneous quantization: any tensor can have a different bitwidth, introducing additional hyperparameters
- High Granularity Quantization makes the bitwidth a learnable parameter via gradient descent
 - Every individual weight quantization can be learned independently (maximally heterogeneous)
- Pruning is implicit via learning 0-width weights
- Resource-aware training via Effective Bit Operations (EBOPs) metric
- Brings additional reduction in resources and latency compared to QKeras
 - **HGQ** vs **Q{E,B,6}** in plot
 - QKeras was already a 10x resource saving over Post Training Quantization
- HGQ2 package recently released and supported in **hls4ml**



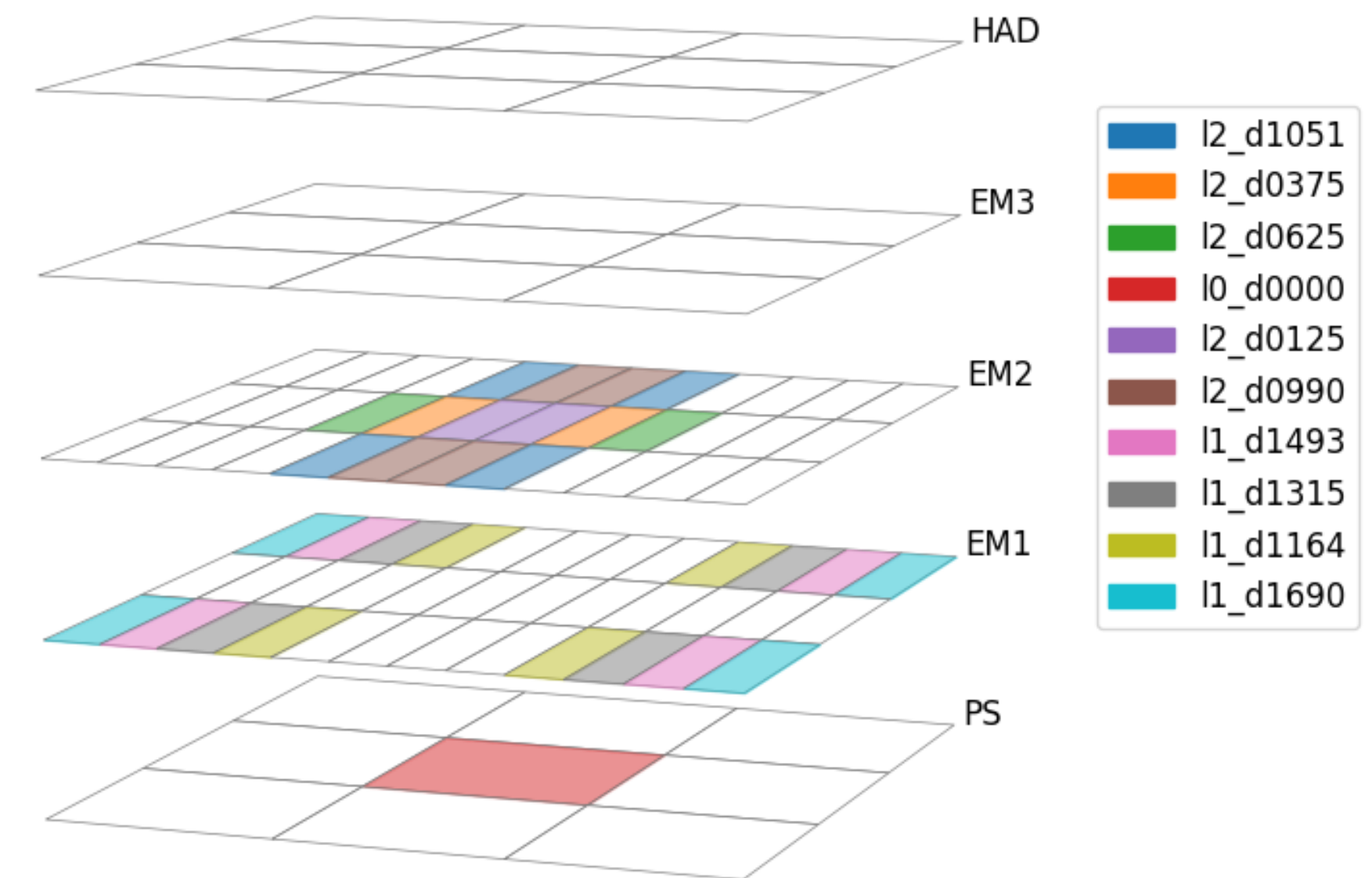
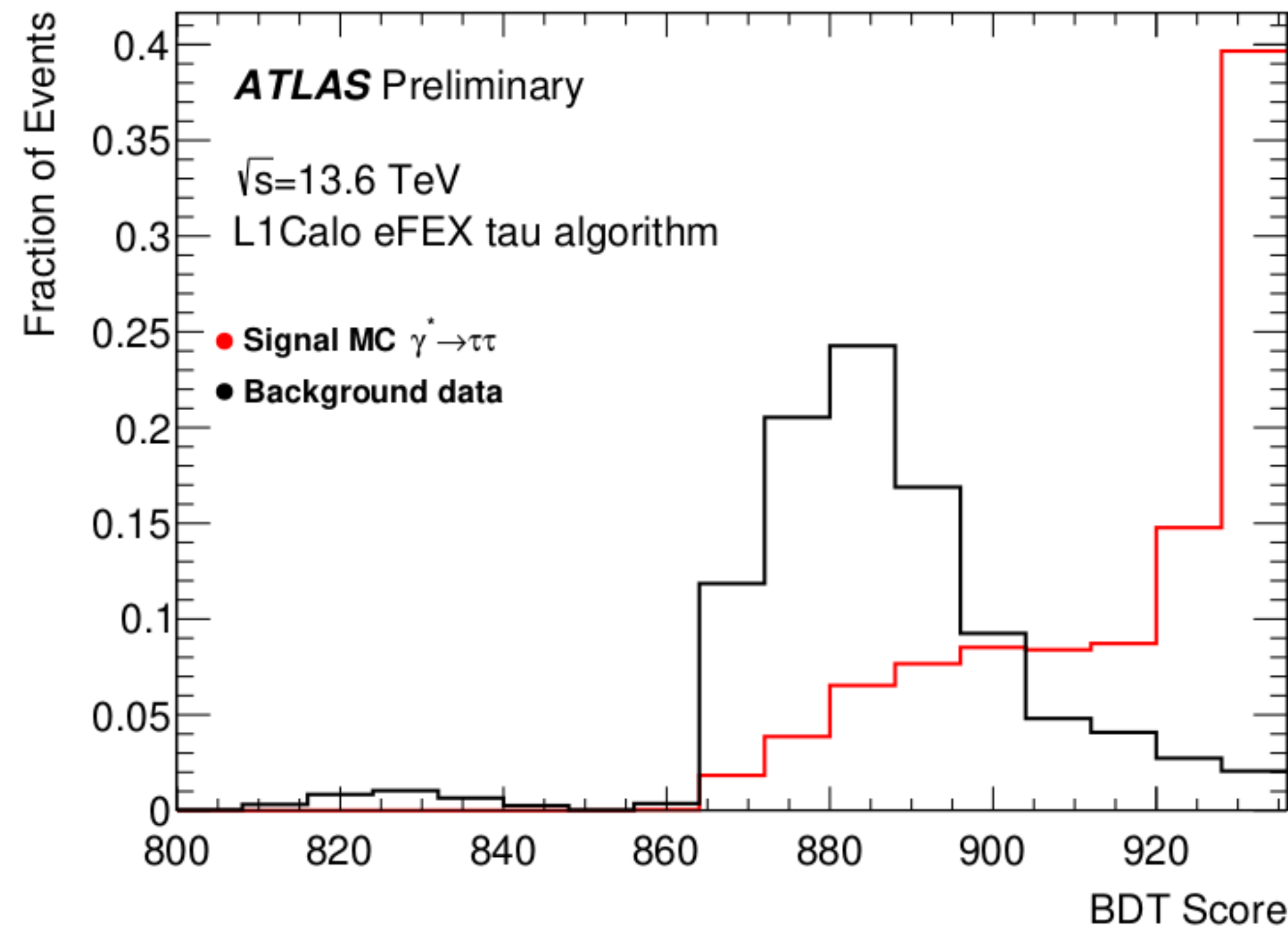
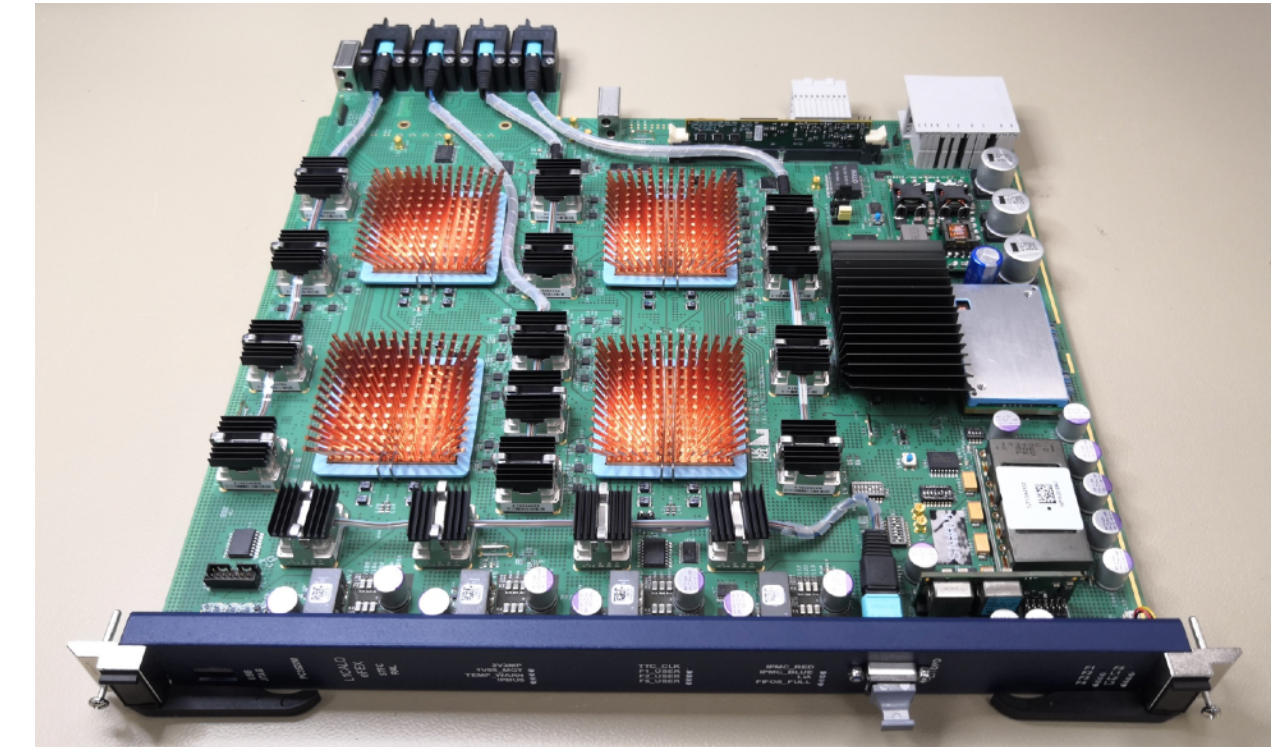
CMS Run 3 Topological Trigger

- A small classifier model trained to target $HH \rightarrow bbWW \rightarrow bbqq\mu\nu$ events
 - Event features: kinematic properties of jets, muons, and HT
 - Classification task: $HH \rightarrow bbWWqq\mu\nu$ vs background (Zero Bias)
- Yields higher trigger efficiency compared to leading kinematic-only selections
 - Single muon $p_T > 22$ GeV; muon $p_T > 12$ GeV & $HT > 150$ GeV
- Great example of how ML can help trigger decisions: ML classifiers do better than cuts
- Deployed in data taking during 2026



ATLAS Run 3 Tau ID

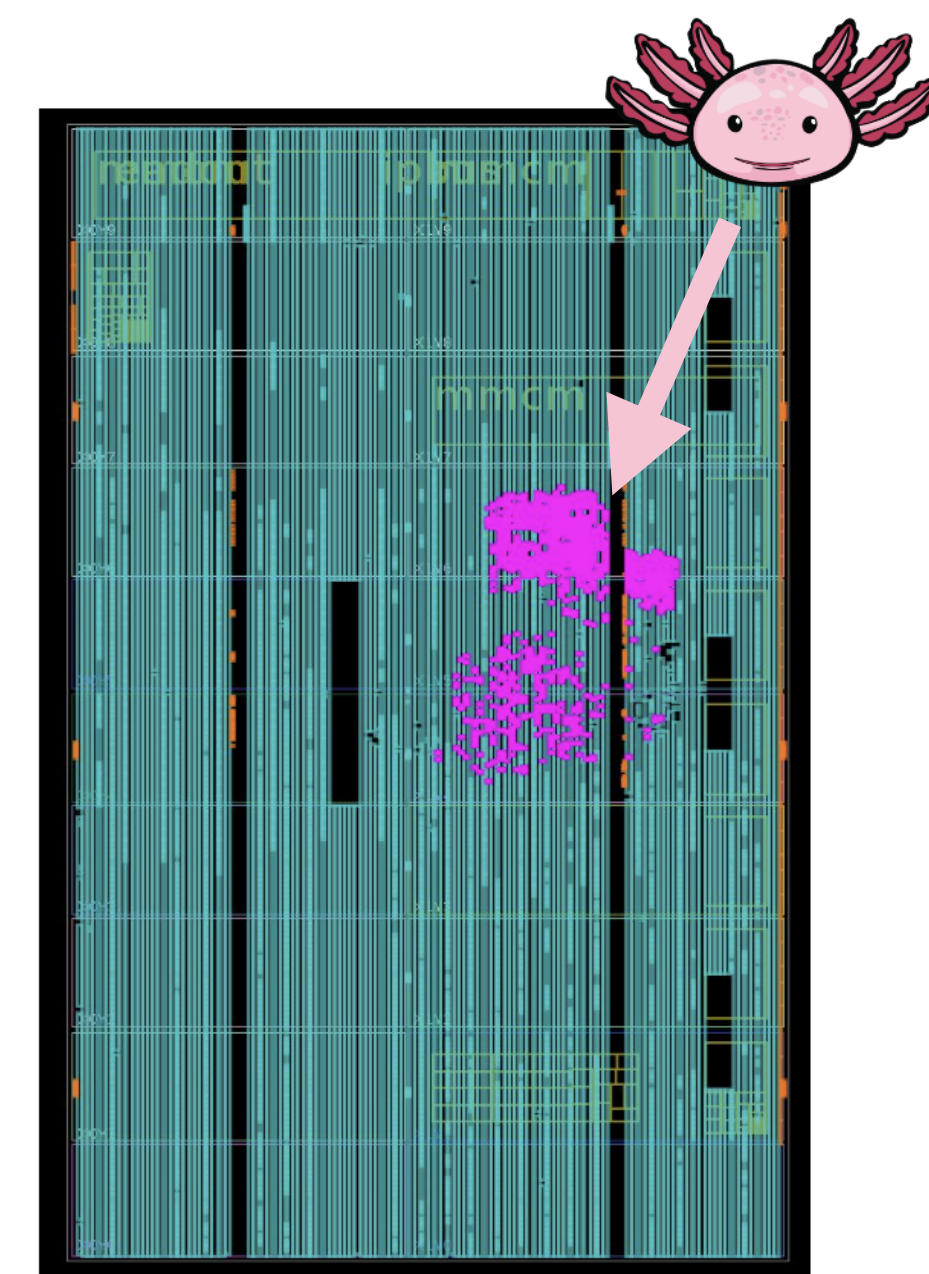
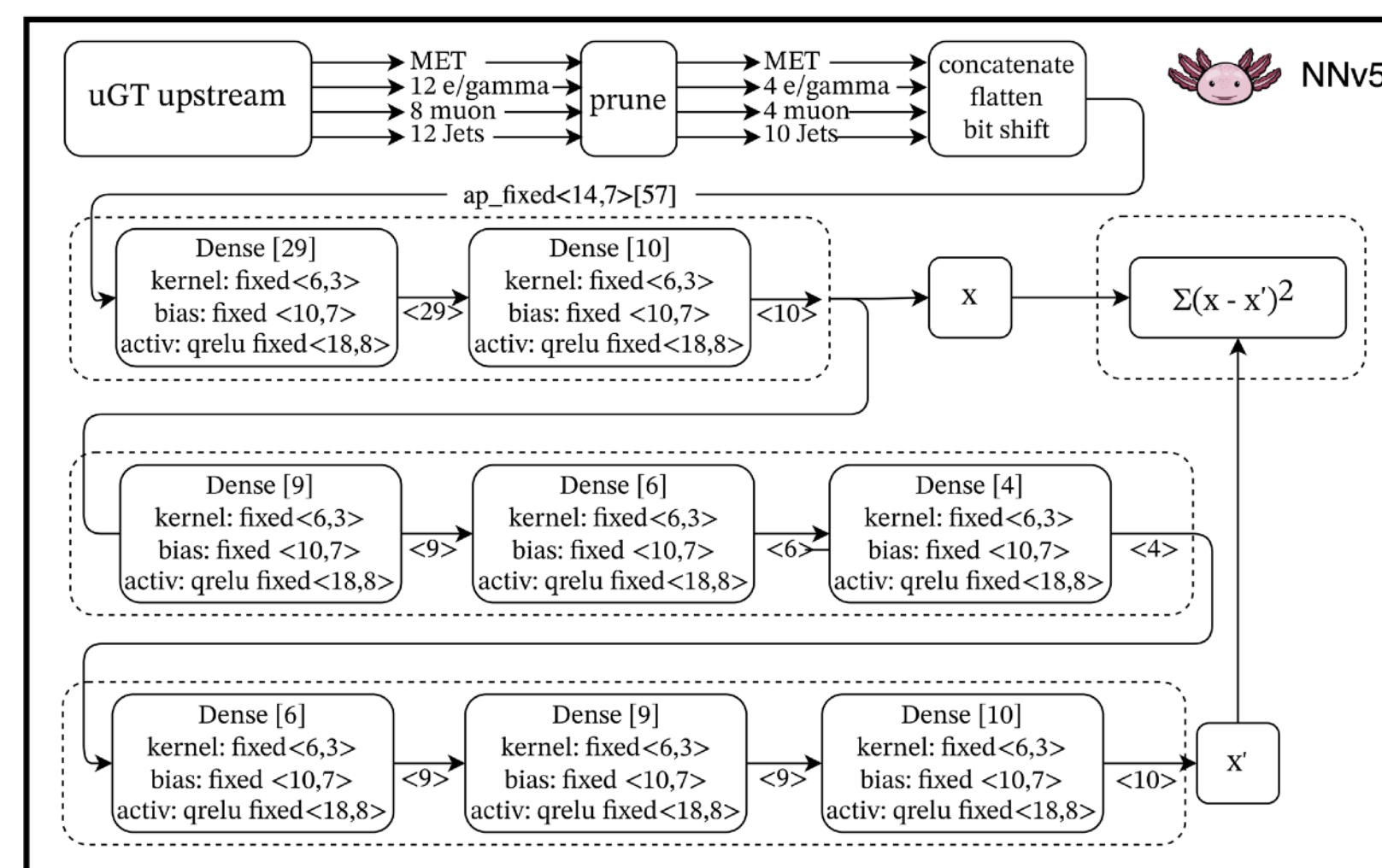
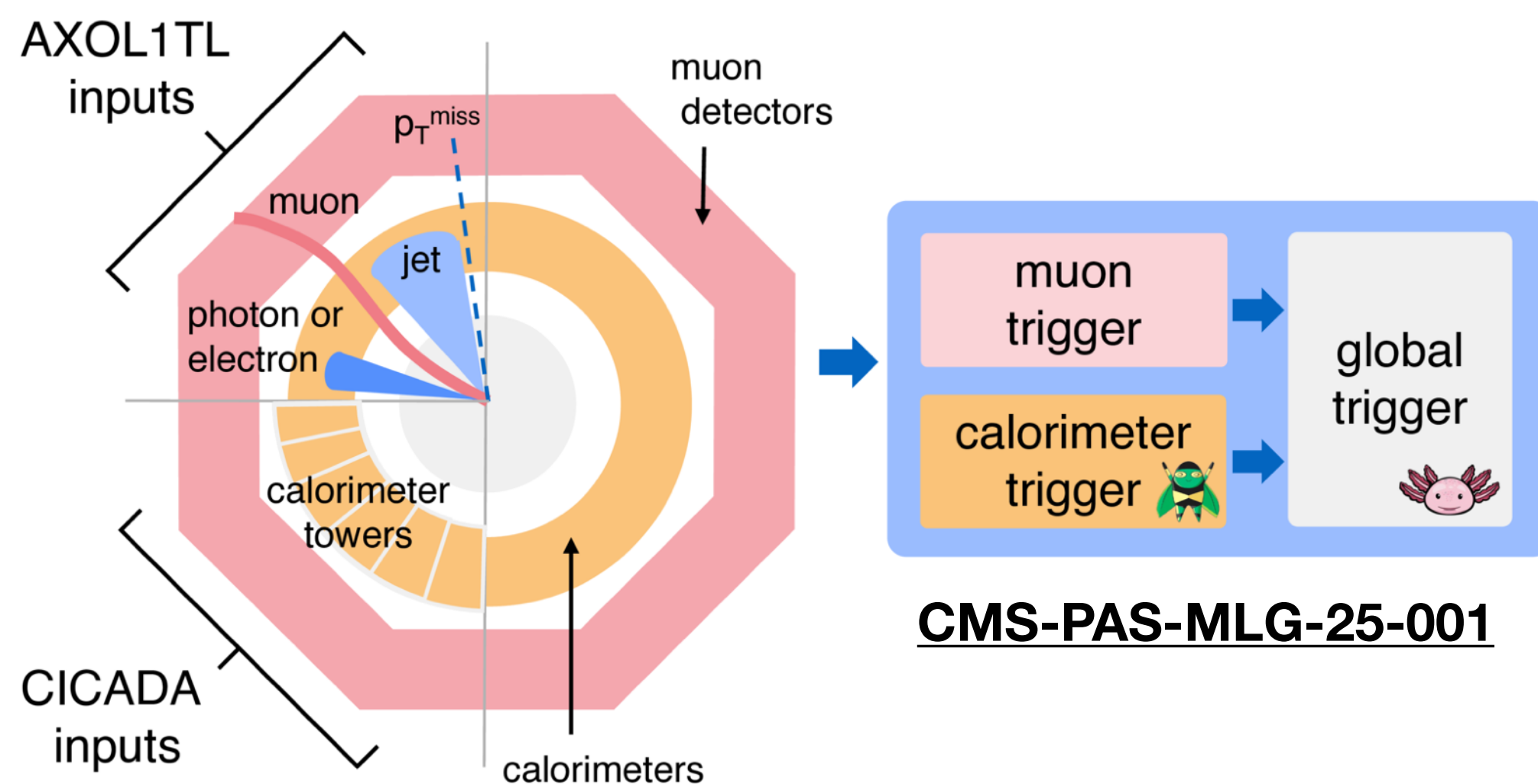
- ATLAS deployed a τ ID BDT in the calorimeter trigger in 2024
 - Using calorimeter cell energy deposit information
 - **conifer** for BDT FPGA deployment
 - Lighter, faster, and more performant than the classical algorithm it replaced





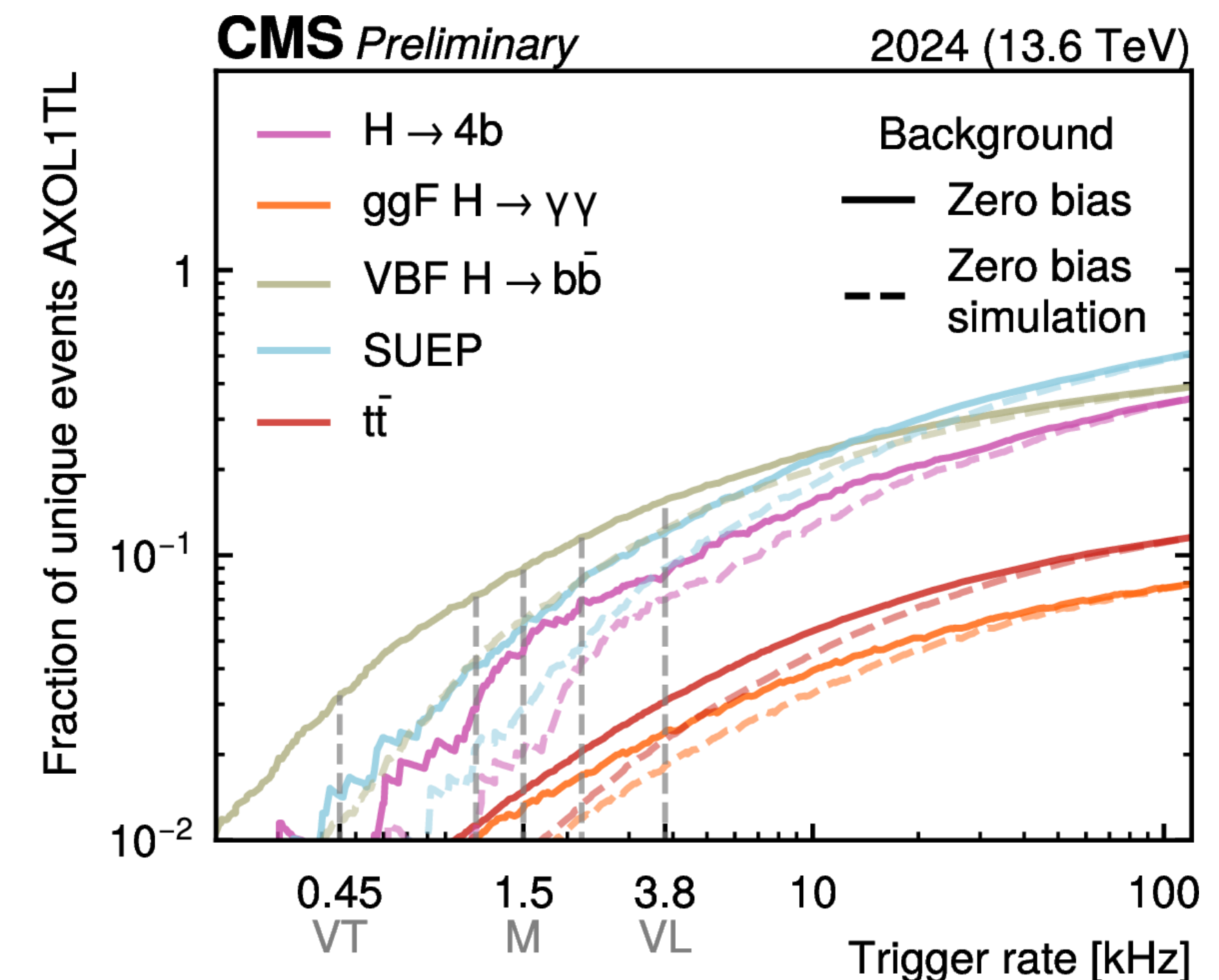
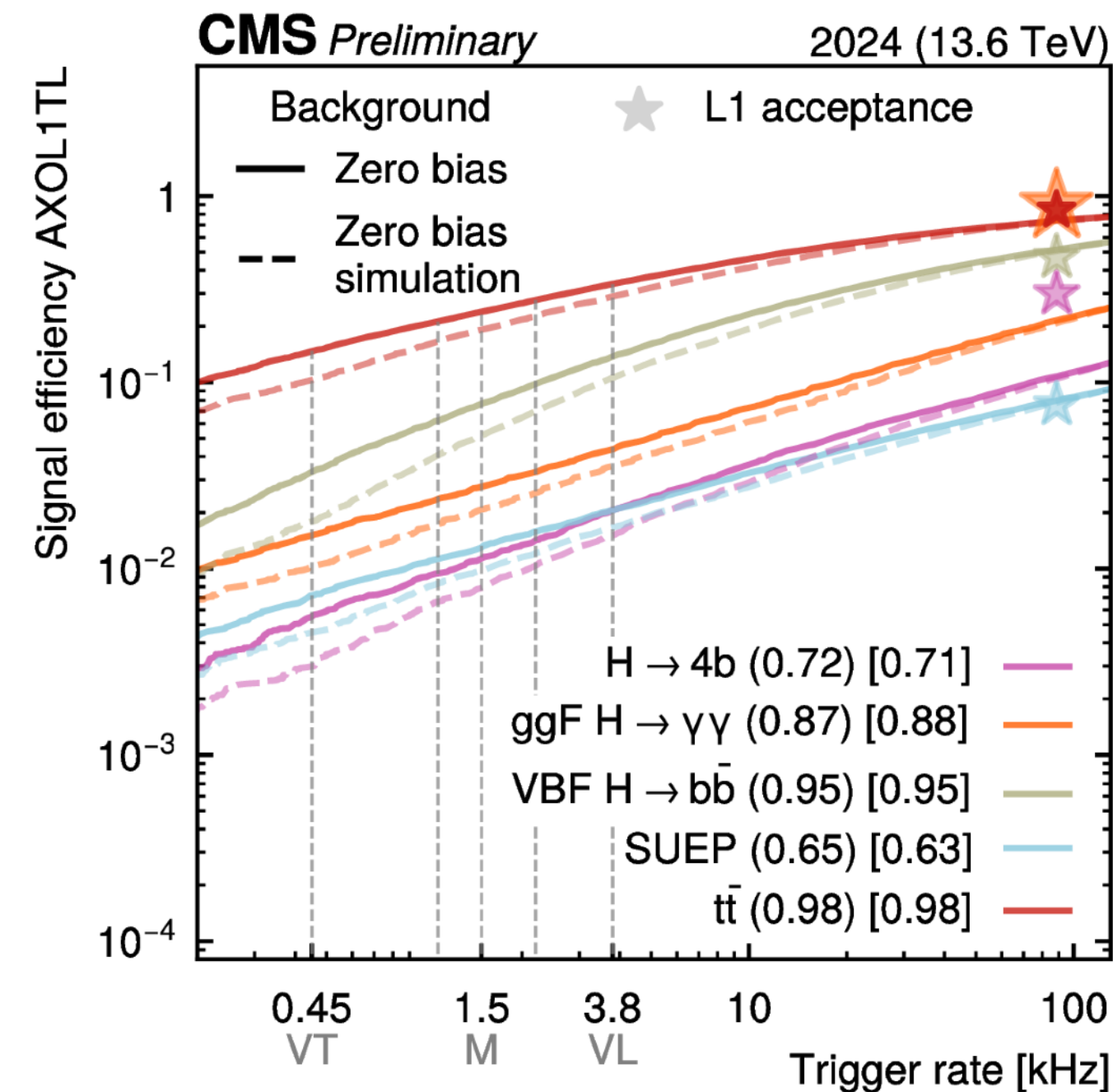
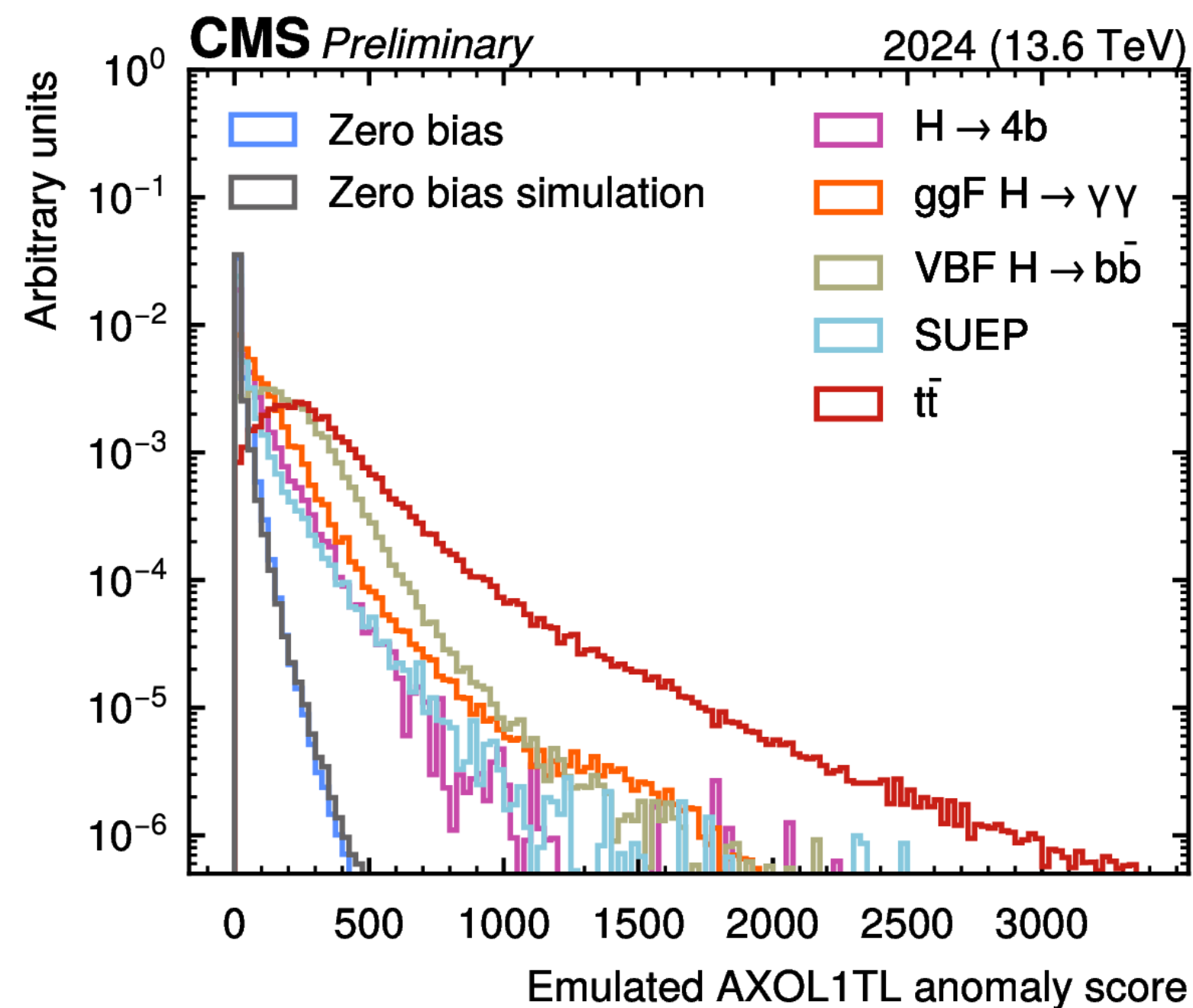
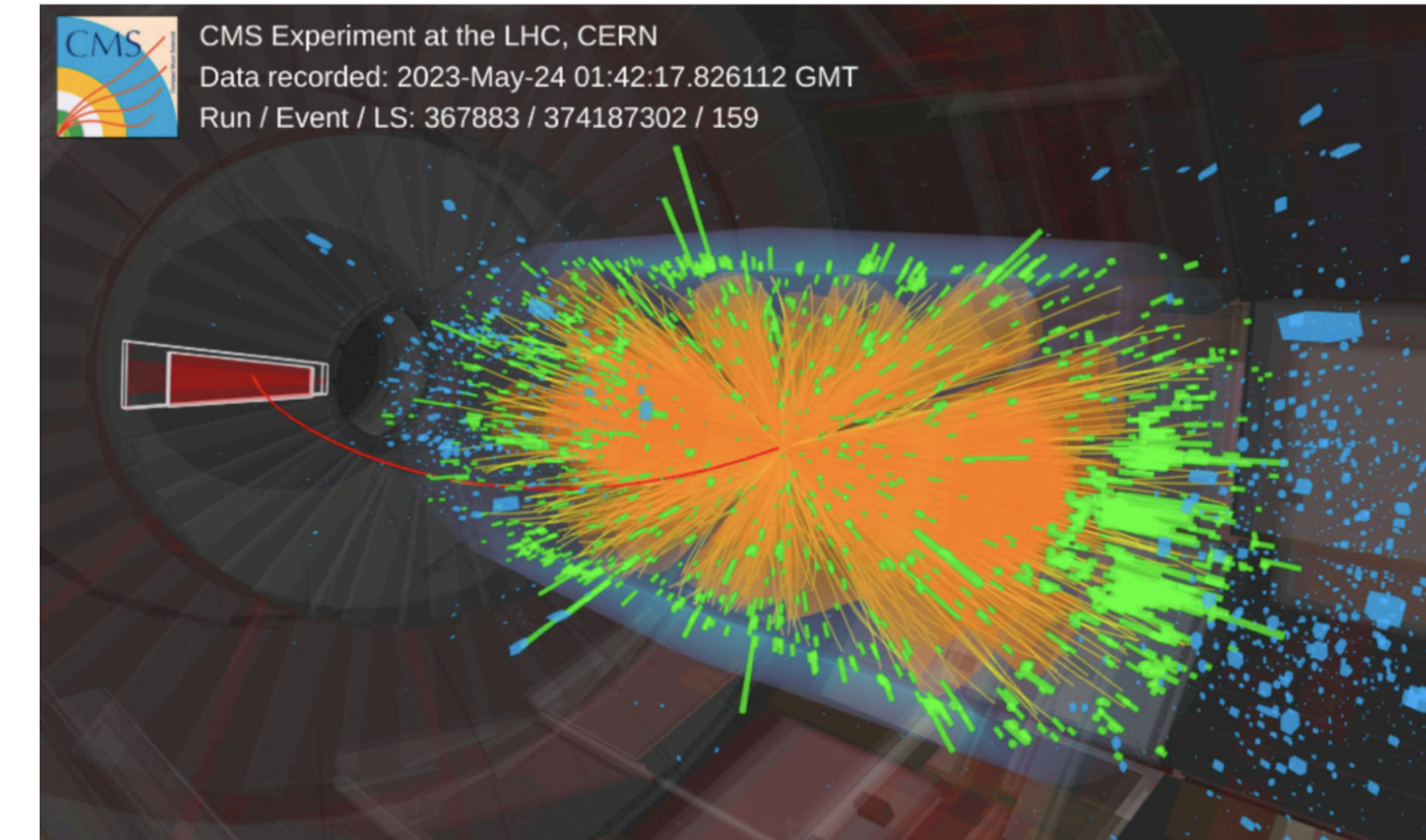
Anomaly Detection

- Trigger selections introduce a (well motivated) bias into downstream analysis
- What if the selections we make in the trigger are wrong? We could be missing the *New Physics*
- Trigger Anomaly Detection computes some “anomaly score” for an event and triggers them with a threshold
 - CMS deployed 2 of them during Run 3 (AXOL1TL and CICADA), and ATLAS triggered with GELATO
 - CMS models were trained on Zero Bias data with a reconstruction loss → signals reconstructed with high anomaly score
- Final AXOL1TL models (v5 & v6) were tiny VICReg embedding followed by a AE with reconstruction loss
 - Relied on da4ml for hardware deployment: 22k LUTs and 50 ns latency → ultra tiny



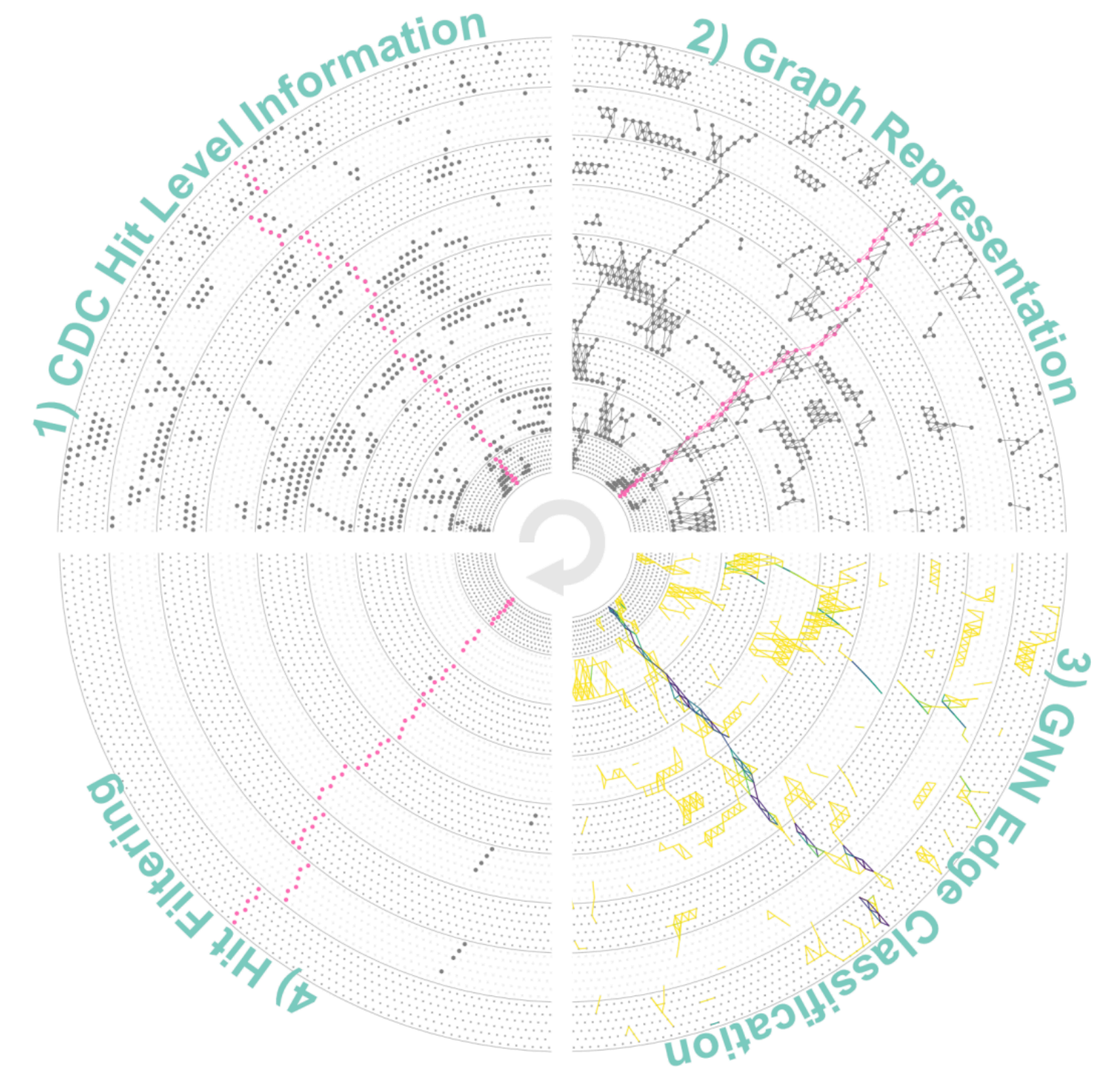
Anomaly Detection

- Analysis of the triggered data is underway
- Shown here: score distributions on different signals (MC); rate vs signal efficiency; rate vs unique event fraction
- The anomaly detection captures events with soft objects, and high multiplicity, that the trigger otherwise misses



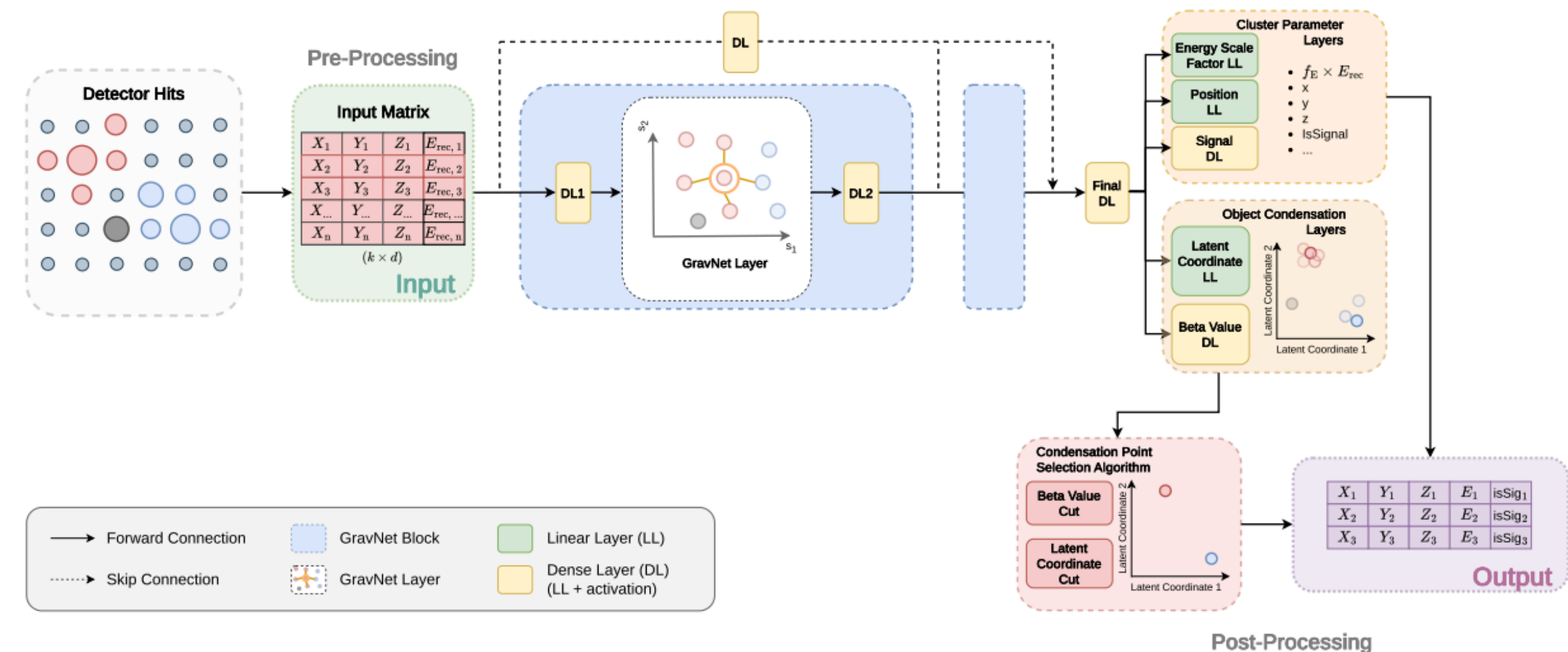
Edge ML at Belle II

- **Belle II** have been actively deploying ML in their trigger, especially *Graph Neural Networks*
 - Using a slightly different set of tooling than QKeras/HGQ → hls4ml
- Calorimeter cluster reconstruction using GravNet architecture
 - Small, controlled dimensionality
 - Simplified edge weights (distance based)
 - ~10 MHz throughput and 3 μ s latency
 - Operating alongside the conventional trigger system
- Hit filtering for the Central Drift Chamber - G. Heine et al
 - Hit (x, y), ADC count vertex features, Δr , $\Delta\phi$, Δt edge features
 - 91% background rejection at 95% hit efficiency



↓ I. Heide *et al*, 2602.15118

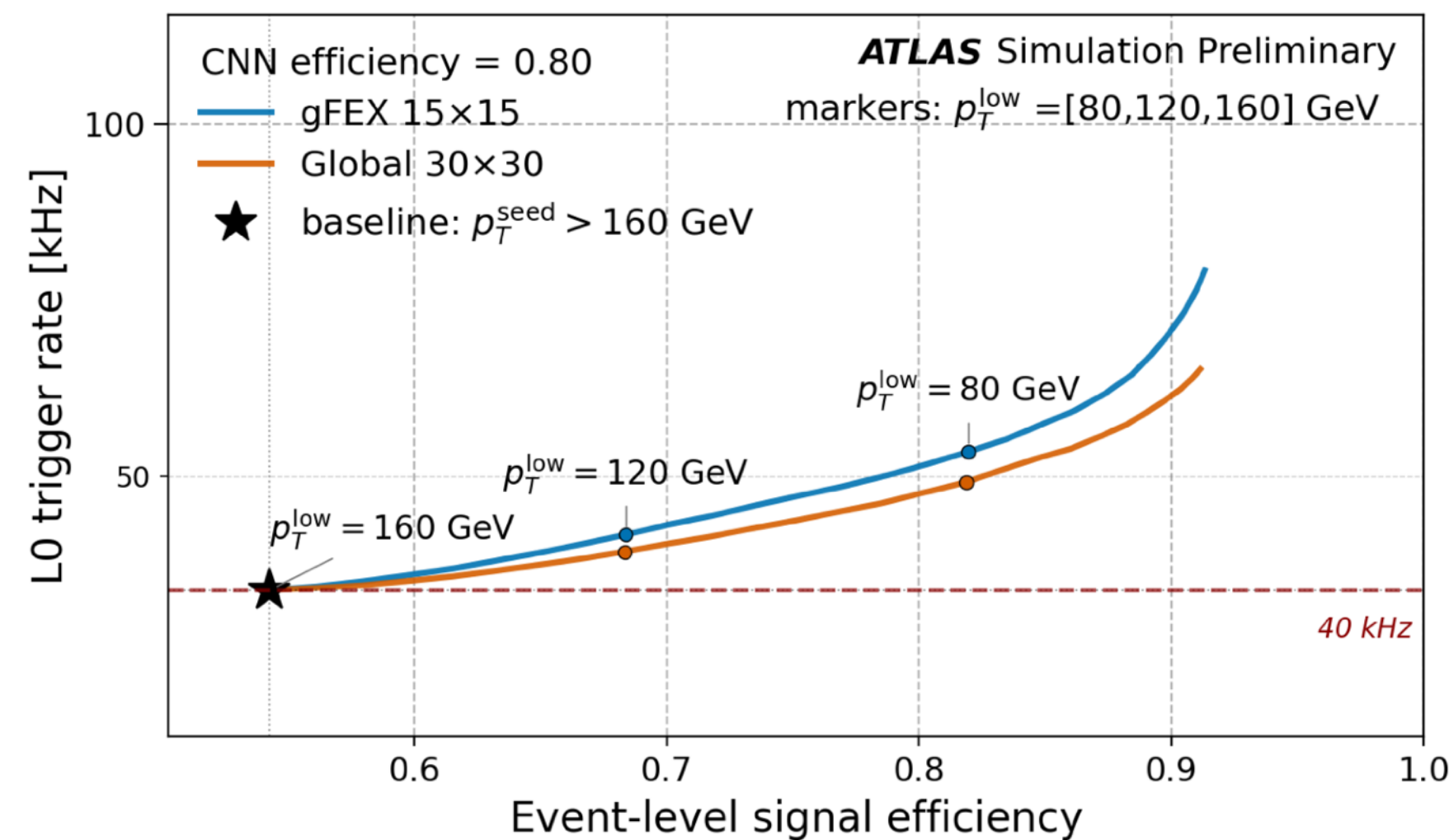
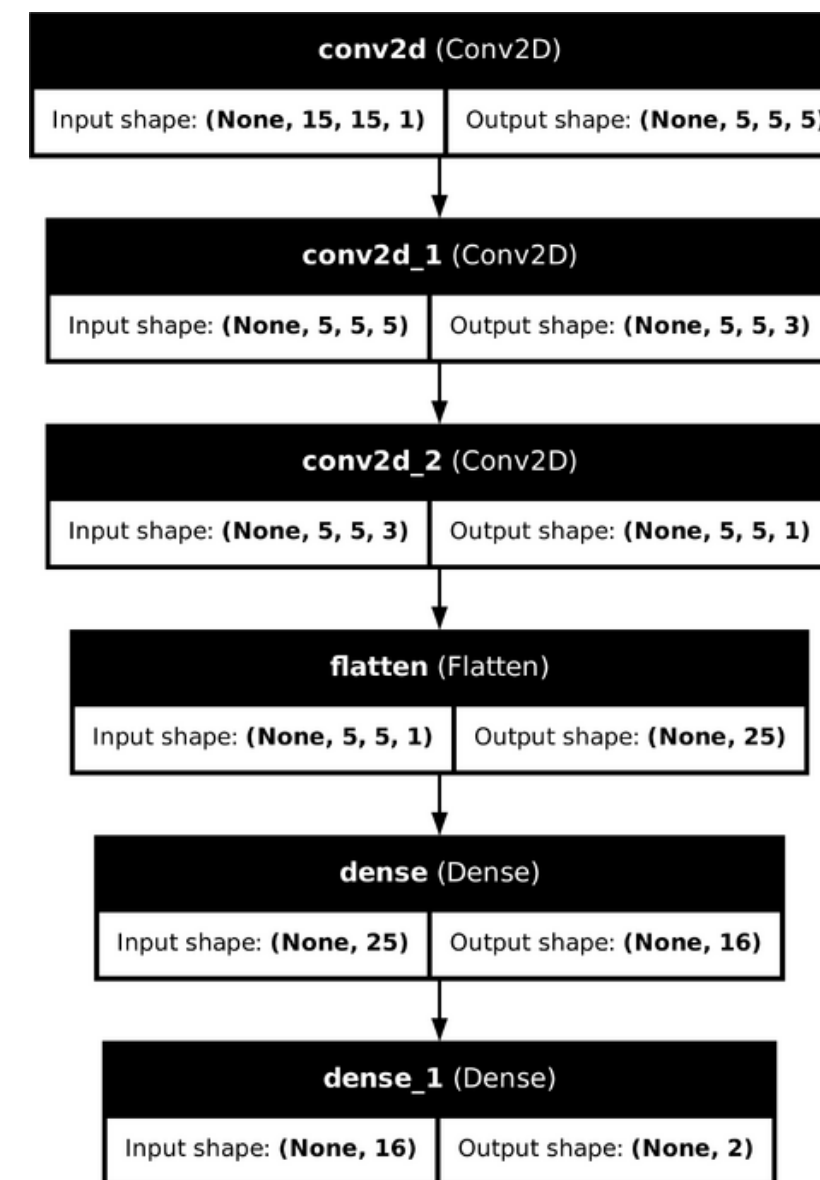
↑ G. Heine, ACAT 2025



ATLAS substructure tagging

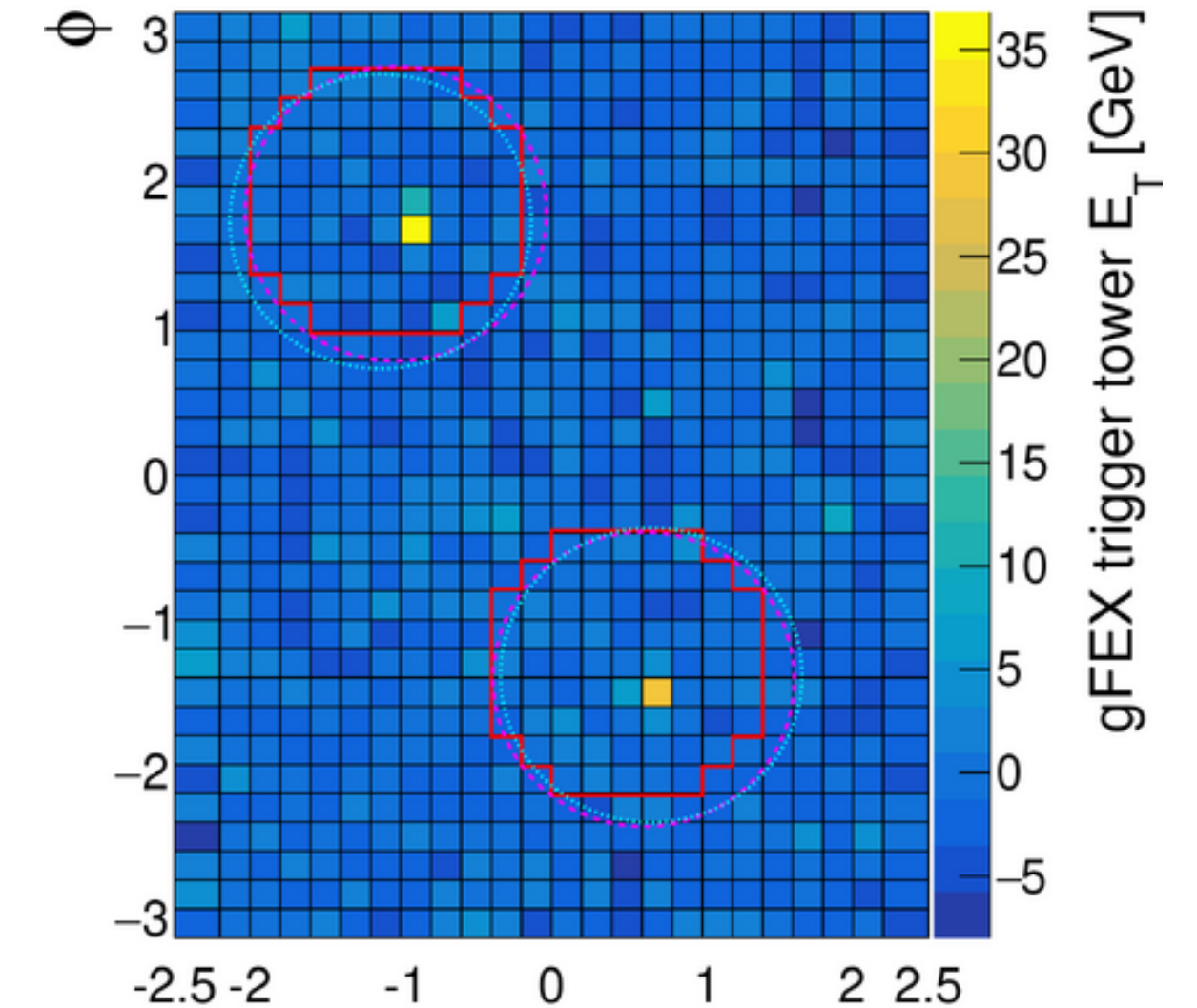
ATL-DAQ-PUB-2026-003

- ATLAS developed a CNN for Phase 2 targeting $HH \rightarrow bbbb$
- Uses image / pixel representation of the event
 - Two views: coarse- and fine-grained calorimeter towers
- Network trained to classify $HH \rightarrow bbbb$ vs QCD
 - QKeras and hls4ml
 - 2-11% resource usage and 0.3-3 μ s depending on granularity
- Applied to augment a kinematic trigger selection

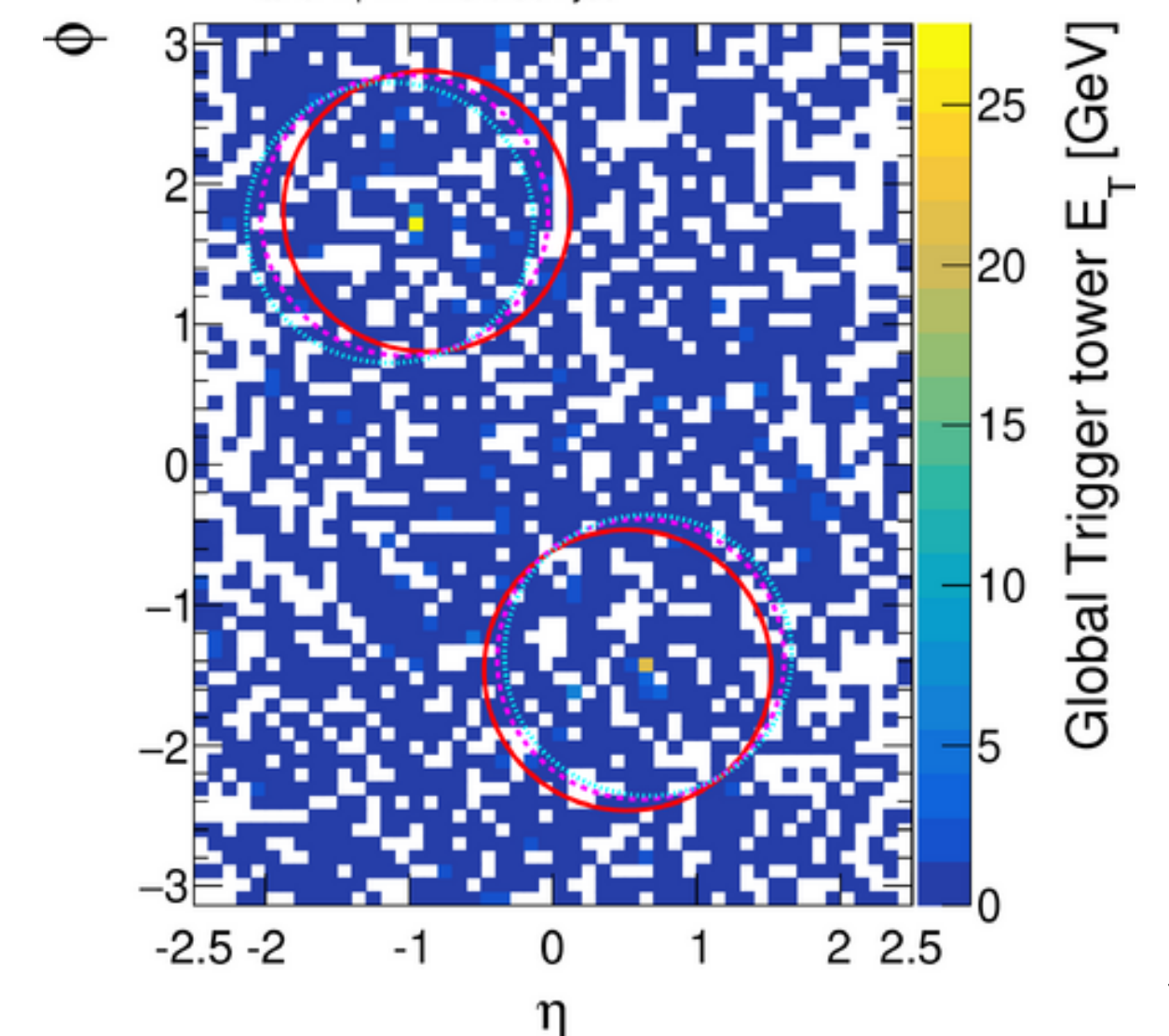


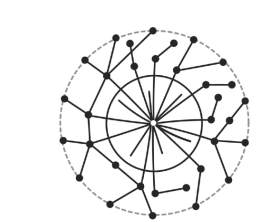
Edge ML - ML4Jets - Sioni Summers

ATLAS Simulation Preliminary
 $HH \rightarrow b\bar{b}b\bar{b}$, pileup = 200
gFEX large radius jet $E_T > 20$ GeV, $|\eta| < 2.5$



ATLAS Simulation Preliminary
 $HH \rightarrow b\bar{b}b\bar{b}$, pileup = 200
Global Trigger large-radius jet $E_T > 20$ GeV, $|\eta| < 2.5$





NexTGen

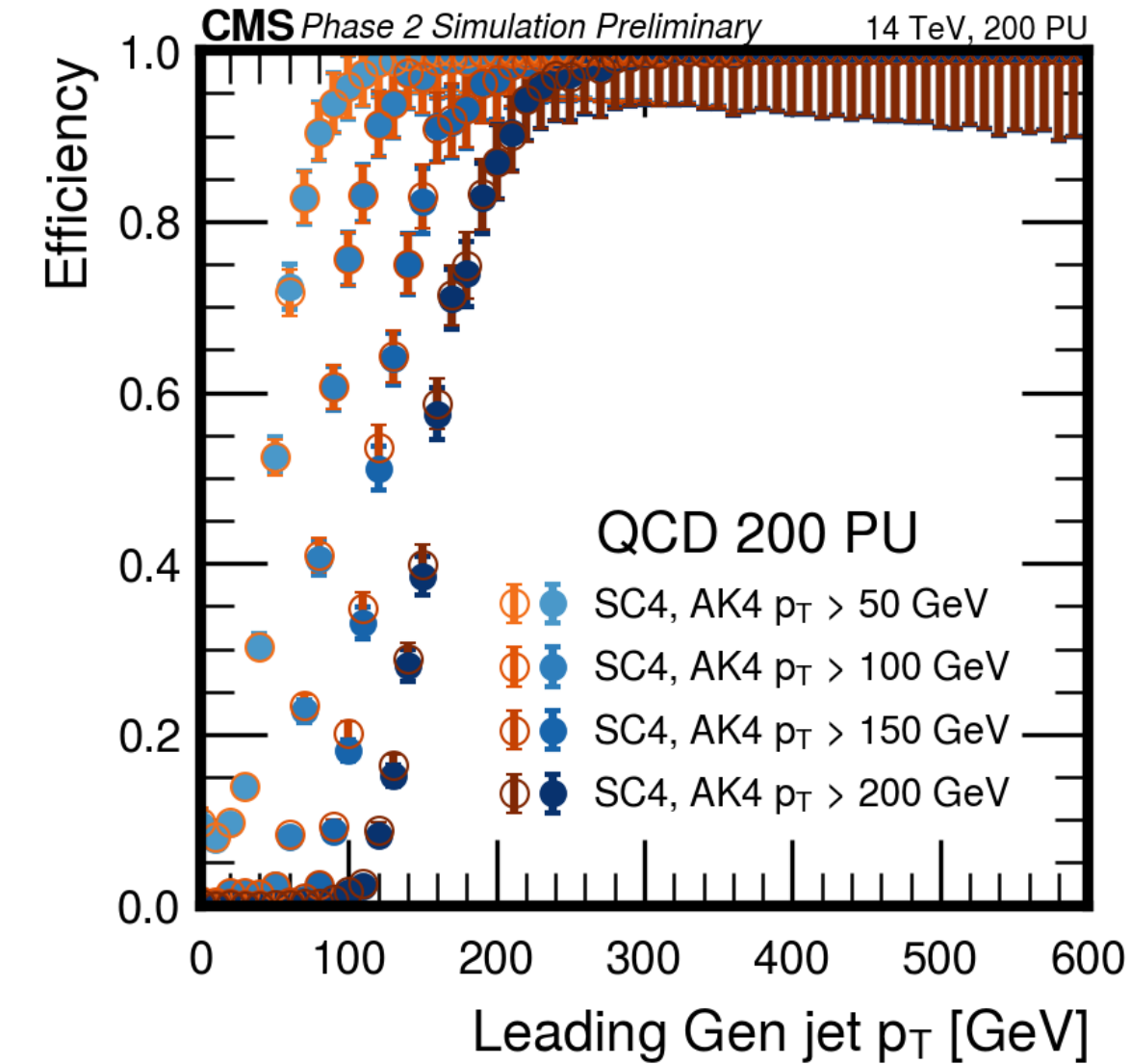
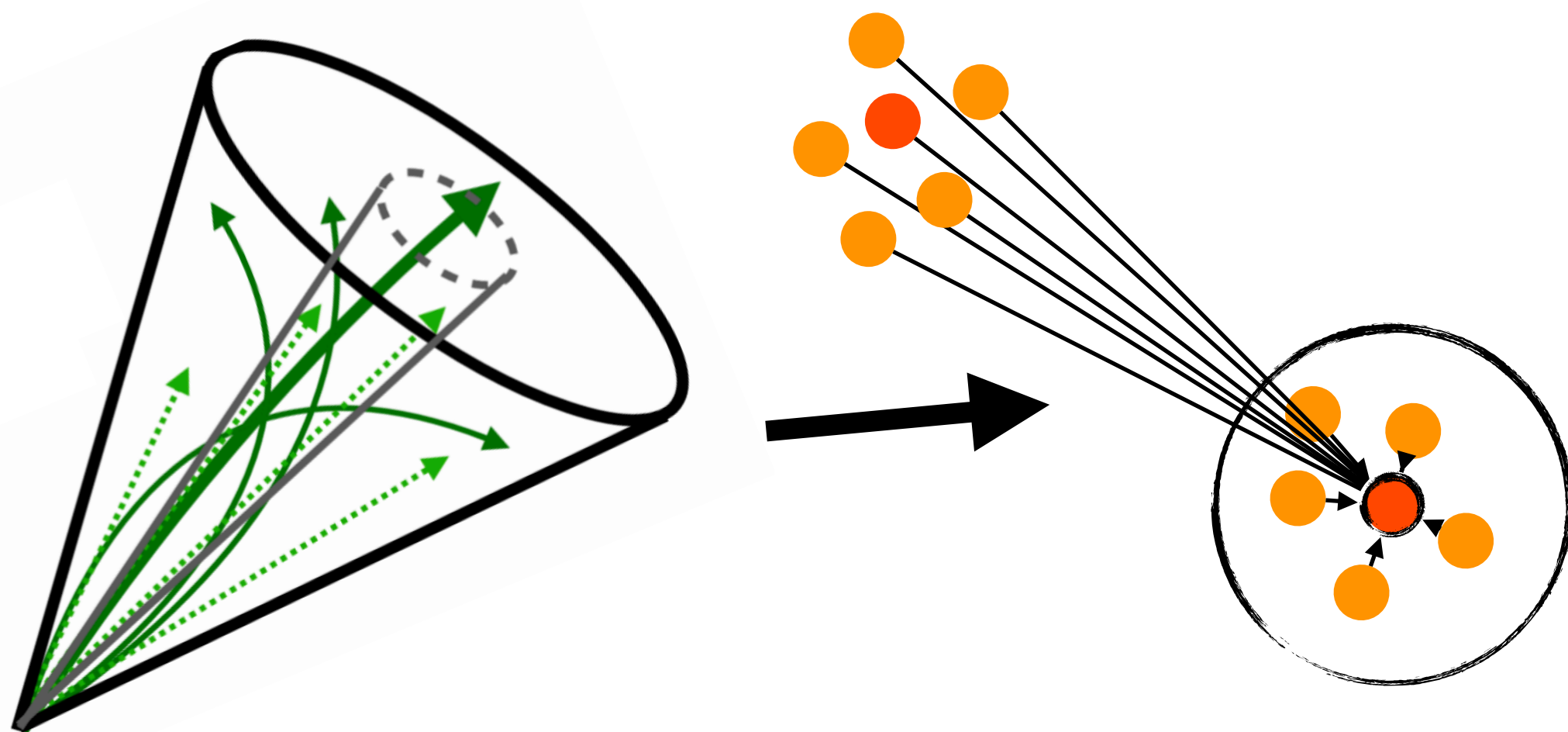
arXiv:2310.08062

arXiv:2509.24371

S. Schaefer today

ML4Jets... at L1T

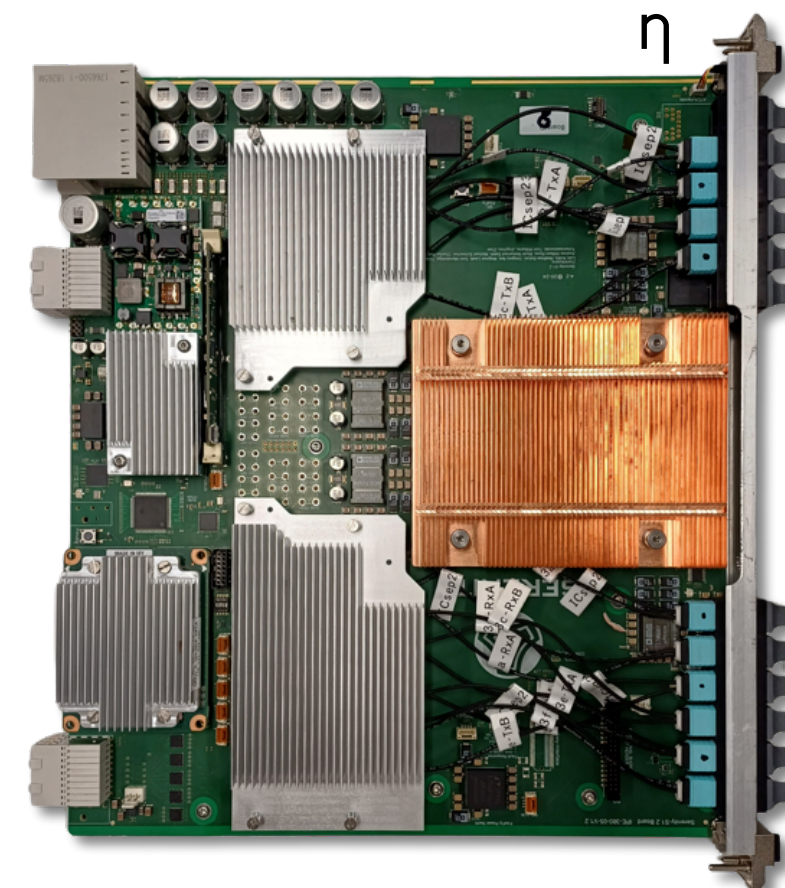
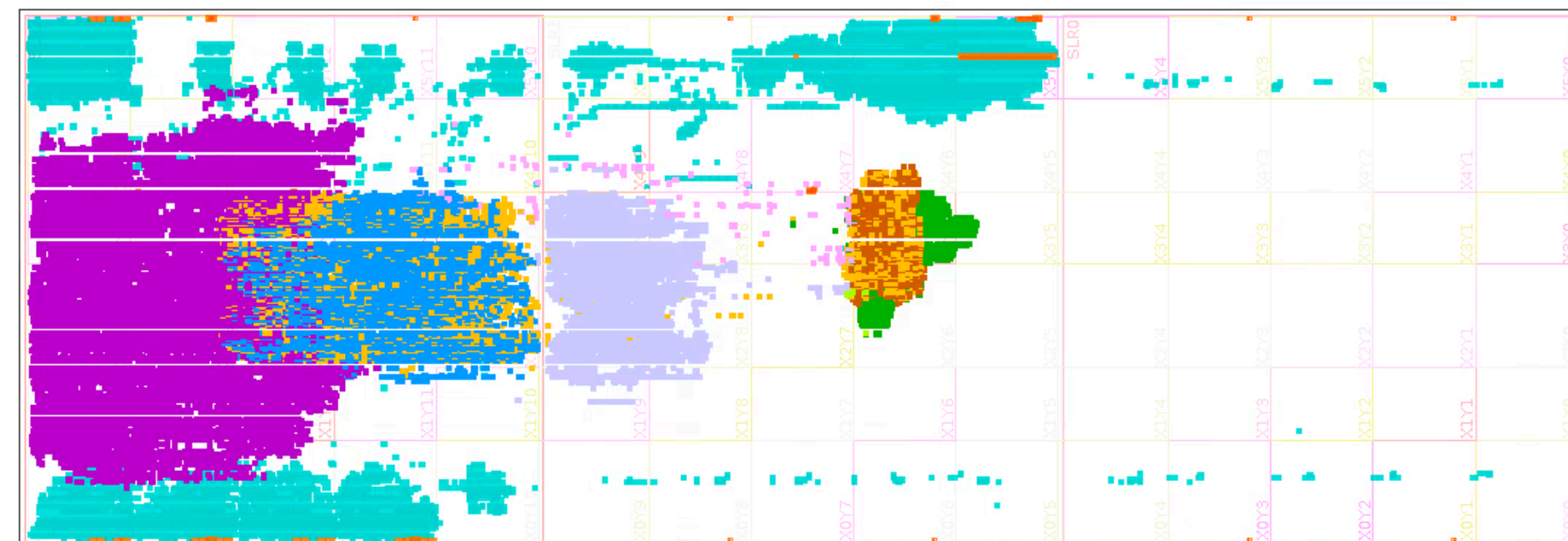
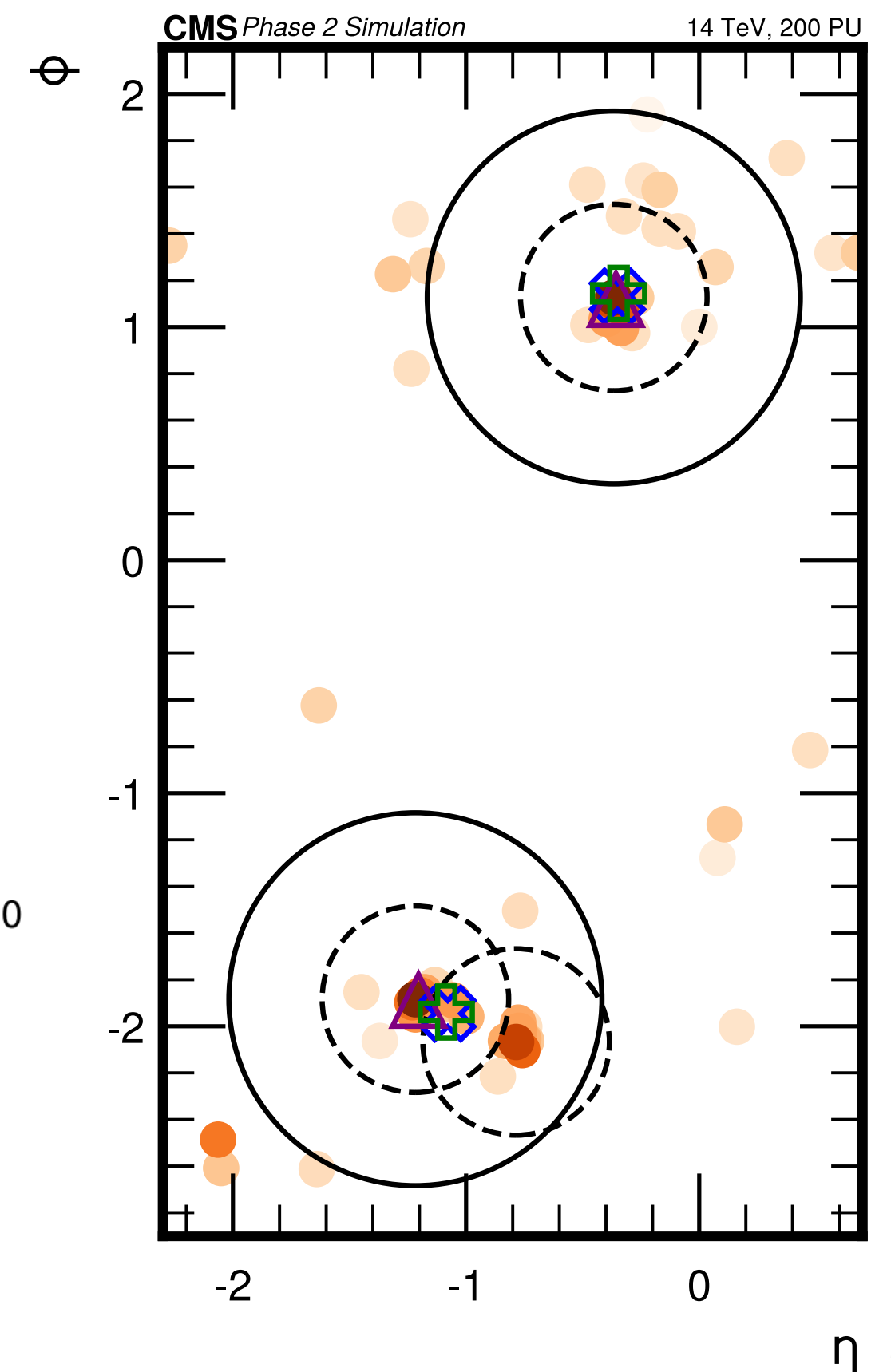
- For **Phase 2 CMS** will have tracking and Particle Flow at Level 1
 - With caveats: $p_T > 2$ GeV tracking, outer tracker only
 - Can cluster jets and keep hold of the set of candidates for tagging
- We developed a simple, fast jet clustering algorithm
 - A high p_T seed particle $\rightarrow$ cone
- We reconstruct two sets of jets: $R=0.4$ and $R=0.8$
- Performance is quite close to anti-kT (on same L1 particles)



↑ Not far from anti-kT

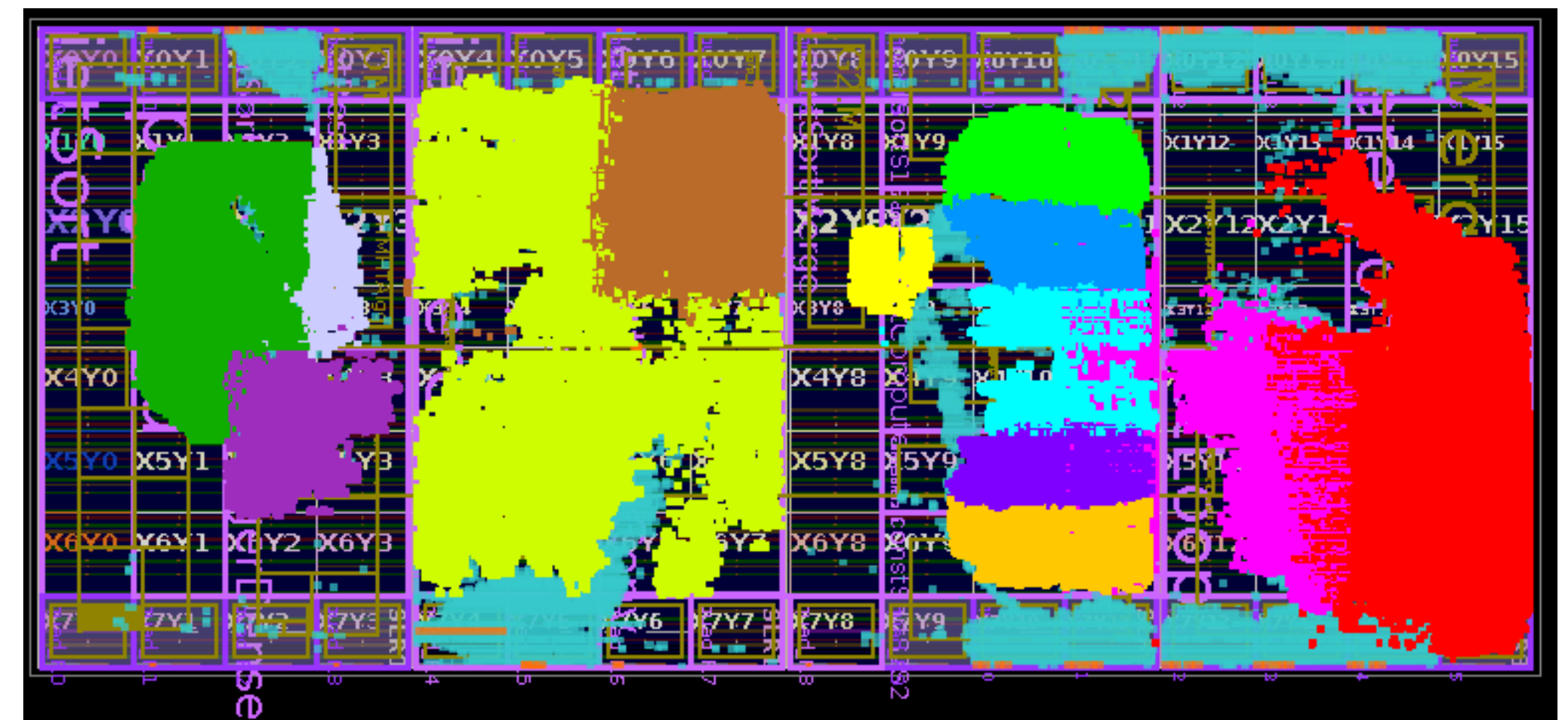
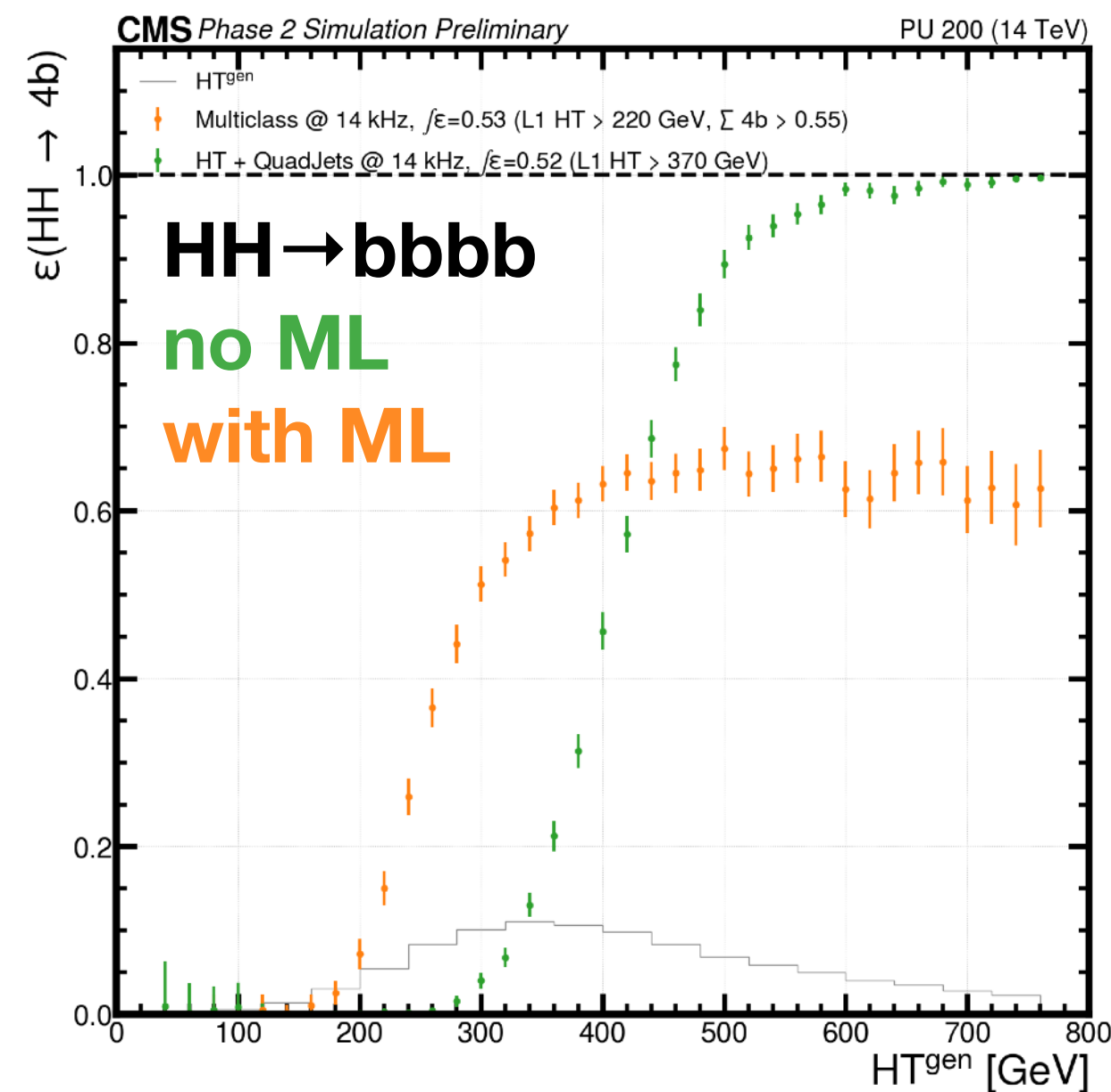
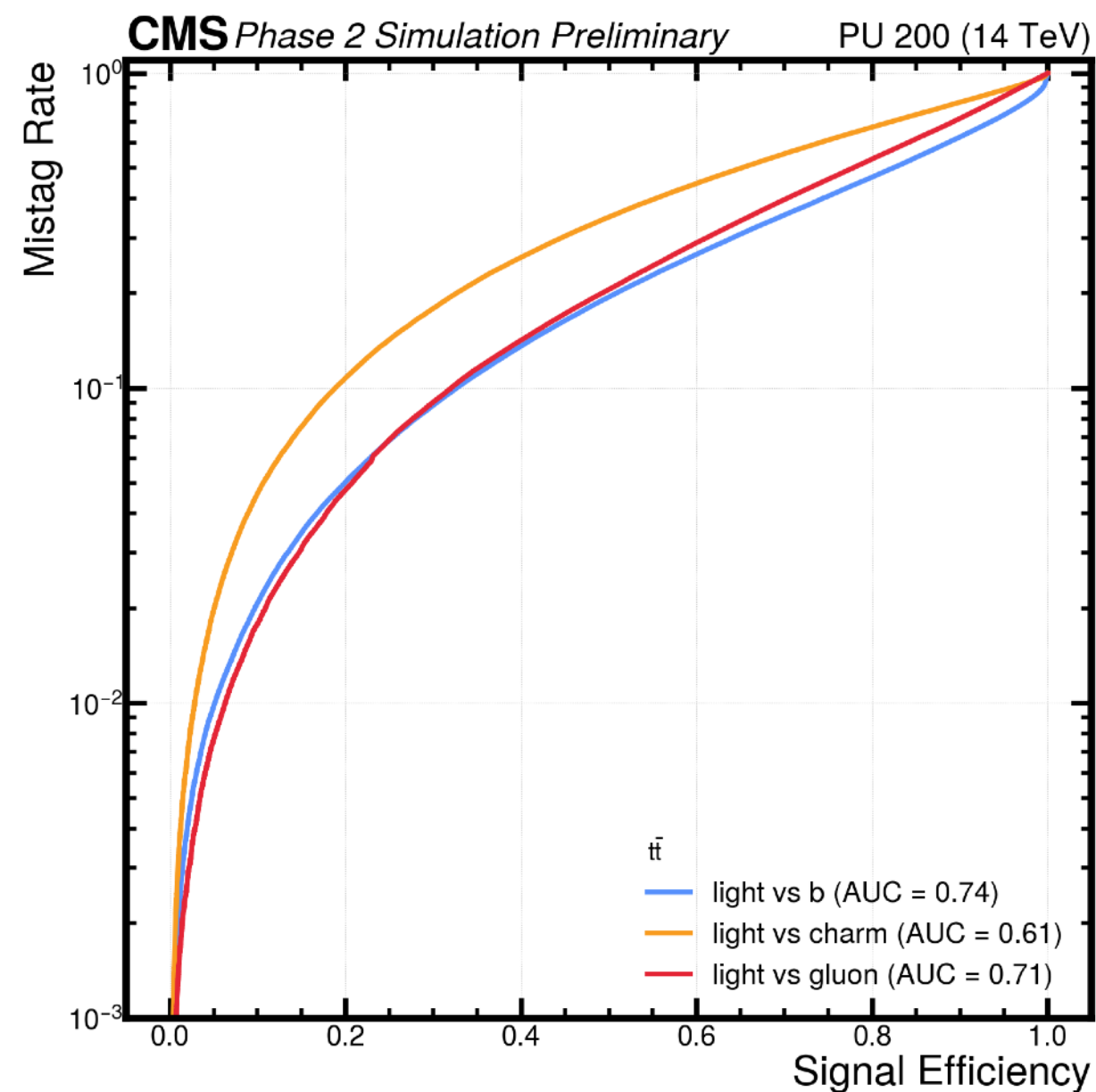
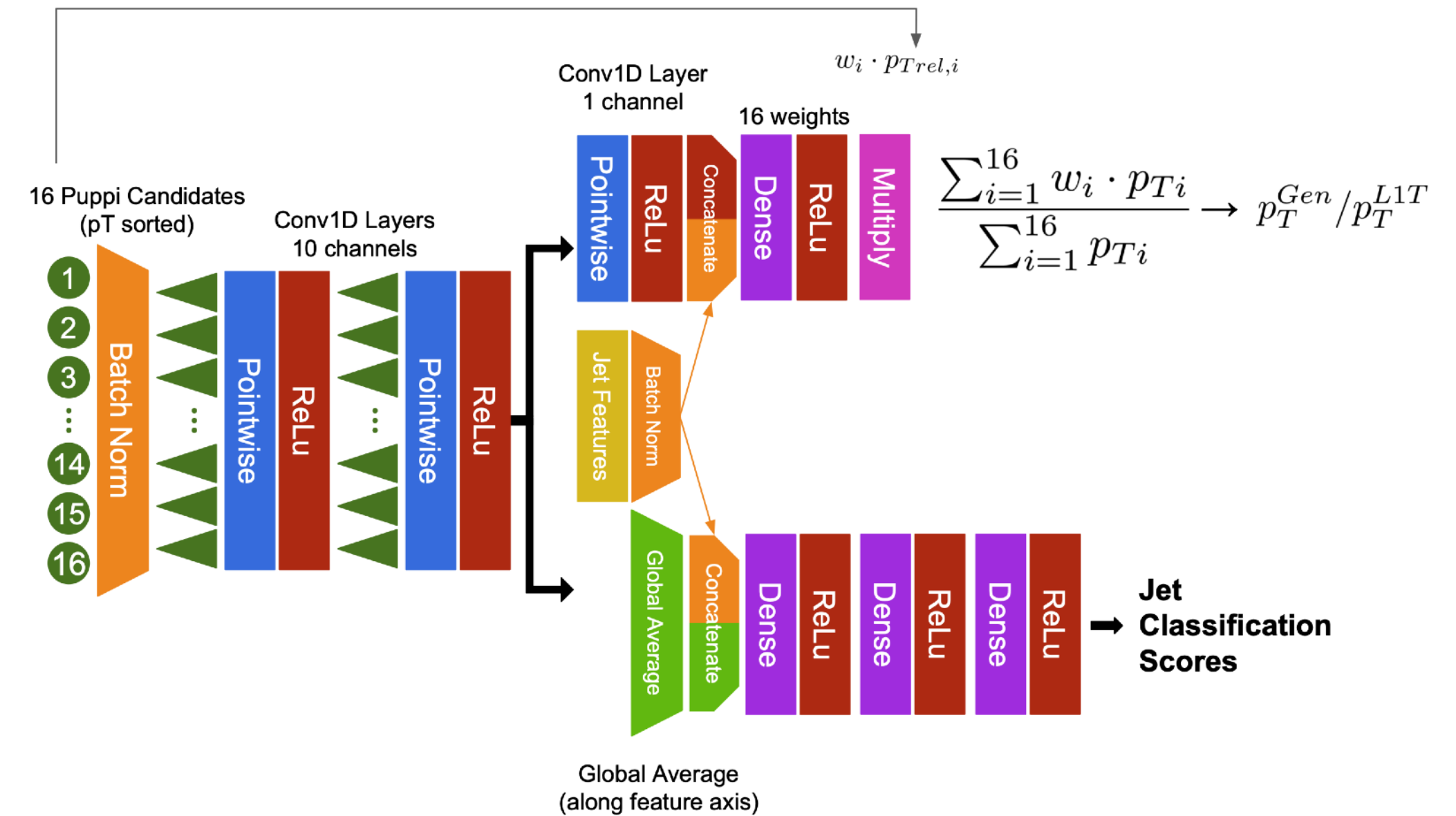
$R=0.4$ & $R=0.8$ jets $\rightarrow$

↓ Works in the chip



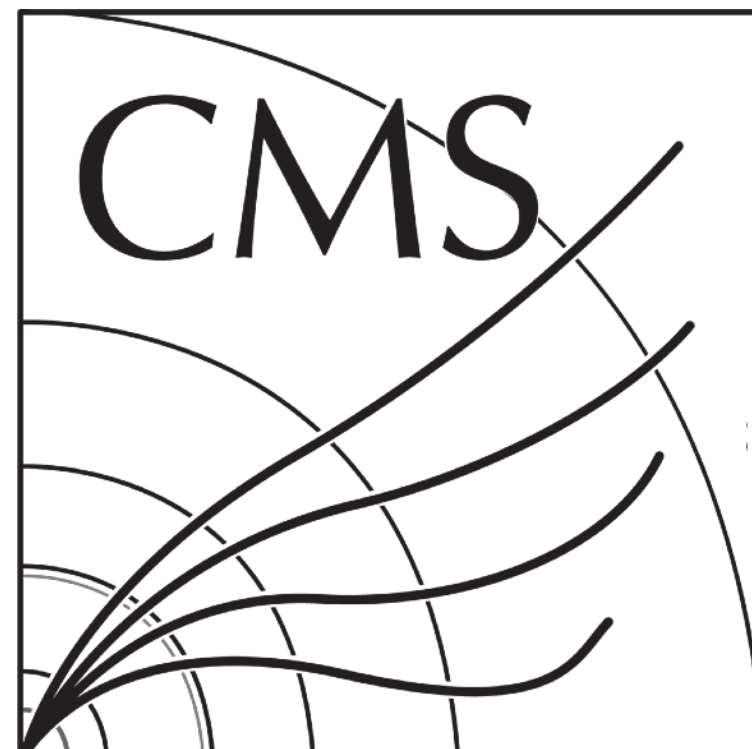
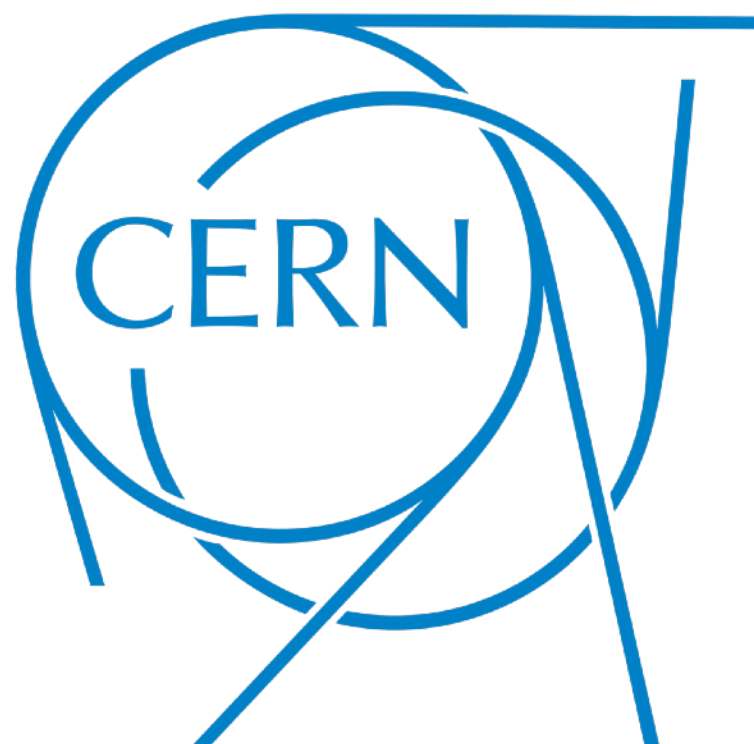
ML4Jets... at L1T

- Trained a DeepSets permutation invariant model on a mixture of MC signals
 - Using 21 features of 16 top- p_T constituents, and 2 jets features
- Multiclass object tagging (g, uds, b, c, e, μ , τ_h^+ , τ_h^-) & p_T regression
- We also developed a dedicated tagger for R=0.8 jets targeting low mass resonances
- Event latency ~ **1 μ s**, throughput **1B tagged jets / second**
- More from S. Schaefer today



Conclusion

- Fast Machine Learning is changing the way we process detector data
 - Applications ran during LHC Run 3 and elsewhere, can account for 25 billion inferences / s for Phase 2 CMS alone
- Embedding ML closer to the detector requires very different ML models to offline
 - Strict constraints (low latency, high throughput, low area, low power) and often highly custom compute platforms (FPGA)
- Projects like hls4ml and conifer ease deployment of ML models optimised for the needs of HEP
- I presented the tools, and techniques like Quantization Aware Training, pruning, and hardware aware training
- We reviewed applications, from Run 3 to Phase 2
- Fast ML software ecosystem and ML triggers at ATLAS and CMS supported by NextGeneration Triggers



✉ sioni@cern.ch
🌐 sioni.web.cern.ch
🐙 @thesps
🔥 @ssummers