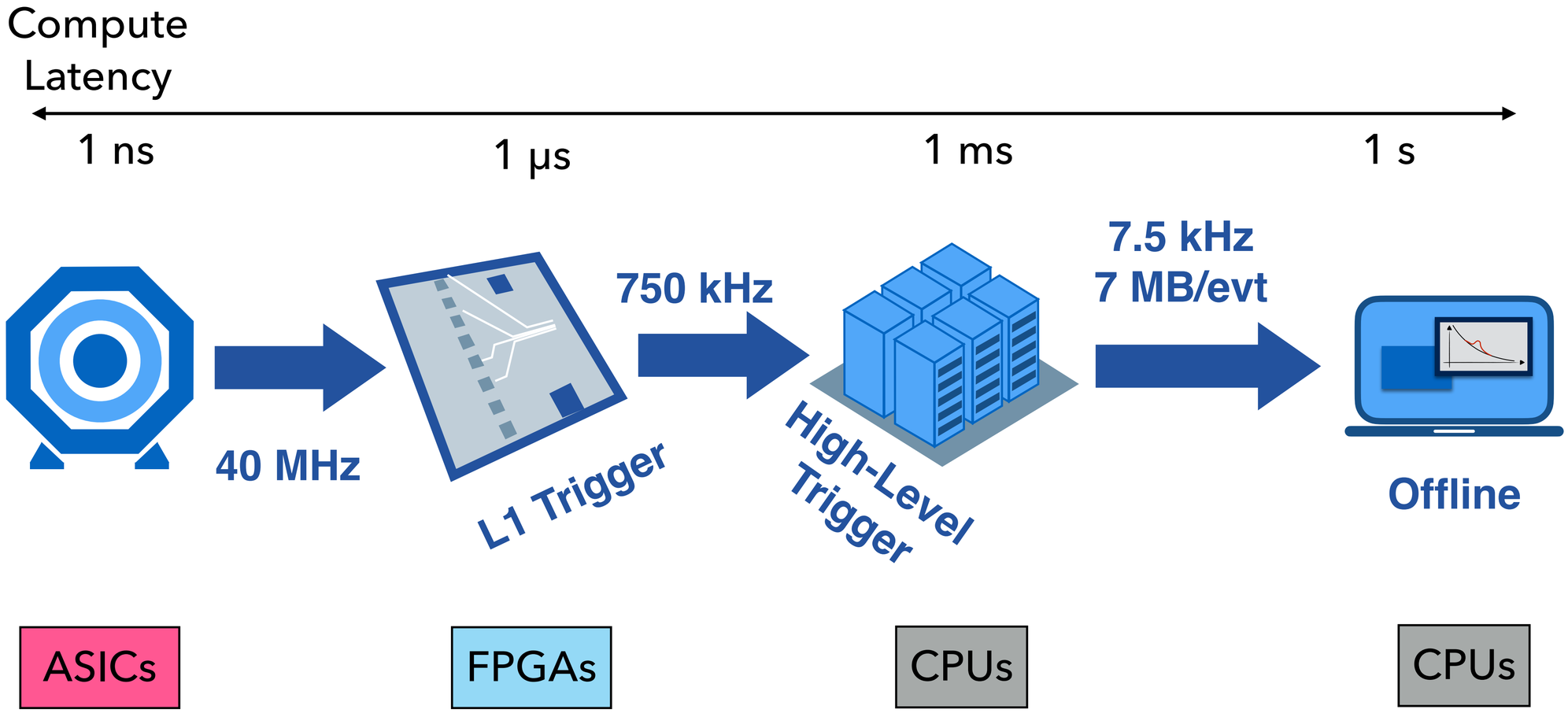
JAVIER DUARTE

ML4FE WORKSHOP

U HAWAII, MAY 19, 2025

hls4ml & FRIENDS: TOOLS FOR ML-BASED TRIGGER AND DAQ

HL-LHC EVENT PROCESSING



Challenges:

Each collision produces O(10^3) particles

The detectors have O(10^8) sensors

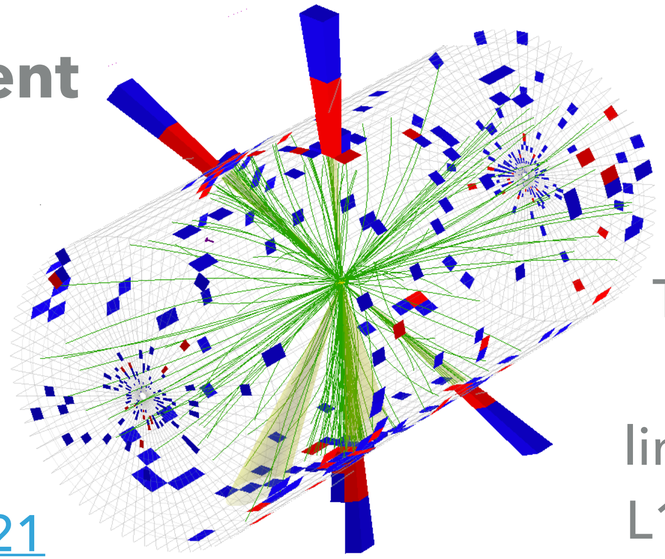
Extreme data rates of O(100 TB/s)

Exabyte-scale datasets

SIMPLIFIED HL-LHC L1 TRIGGER MENU

- Single/double/triple muons/electrons
- Photons
- Taus
- Hadronic
- Missing transverse energy
- "Cross" triggers (not shown)

4-jet event



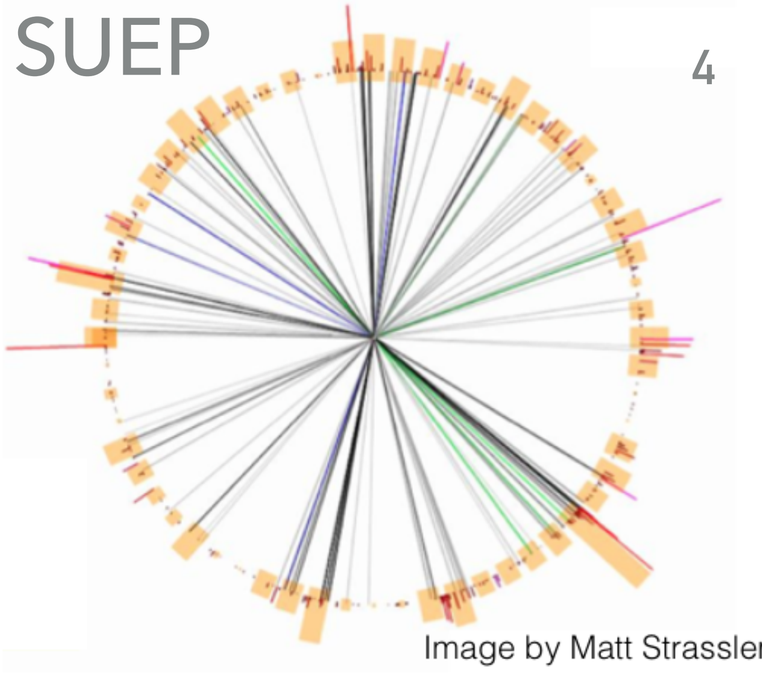
CMS-TDR-021

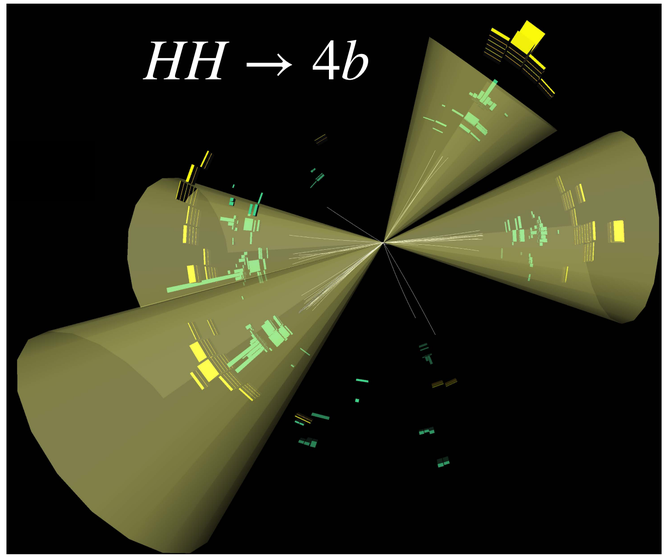
Thresholds set by backgrounds, limited resolution @ L1, and rate budget

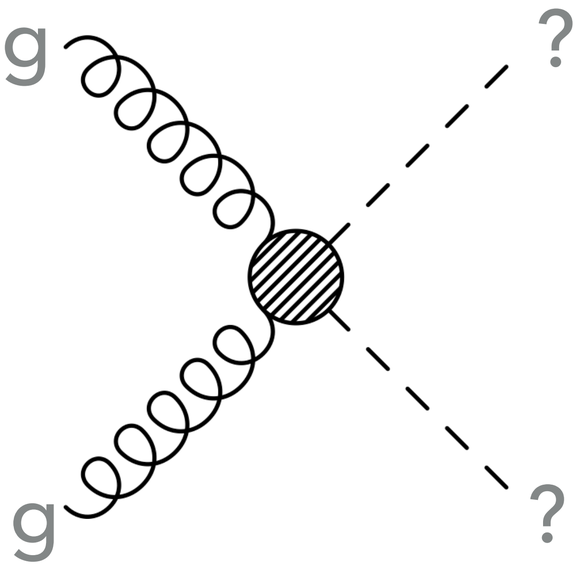
Trigger	Threshold [GeV]
1 μ	22
2 μ	15, 7
3 μ	5, 3, 3
1 e	36
2 e	25, 12
1 γ	36
2 γ	22, 12
1 τ	150
2 τ	90, 90
1 jet	180
2 jet	112, 112
H_T	450
4 jet + H_T	75, 55, 40, 40, 400
p_T^{miss}	200

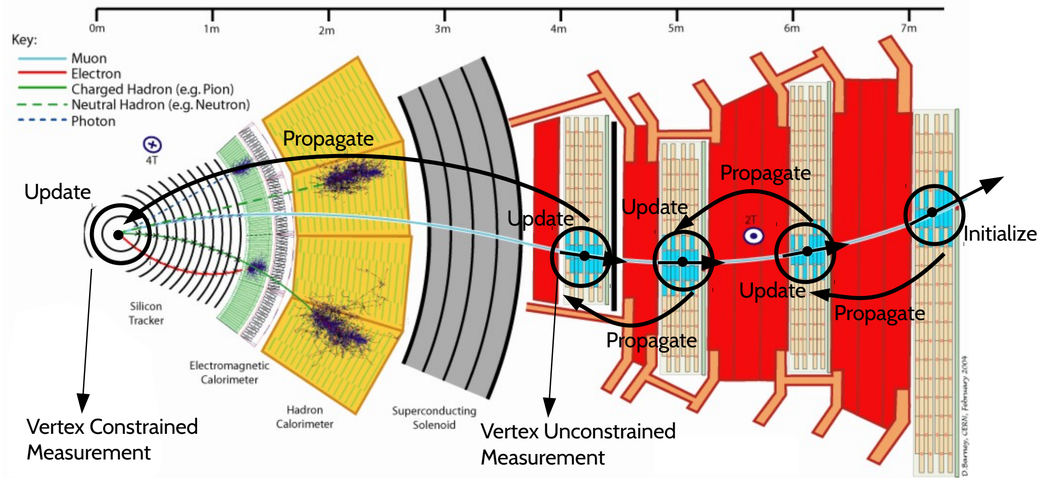
WHAT COULD WE BE MISSING?

- How can we trigger on more complex low-energy hadronic signatures? Long-lived/displaced particles?
- What if we don't know exactly what to look for?
- What if our signatures require complex multivariate algorithms (e.g. b tagging)?
- How can we improve on our traditional (often slow) reconstruction algorithms?



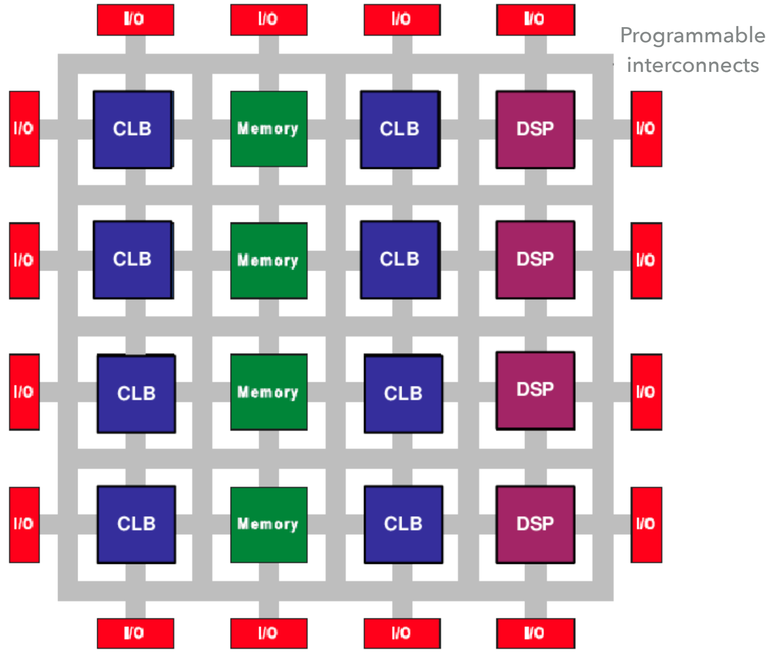


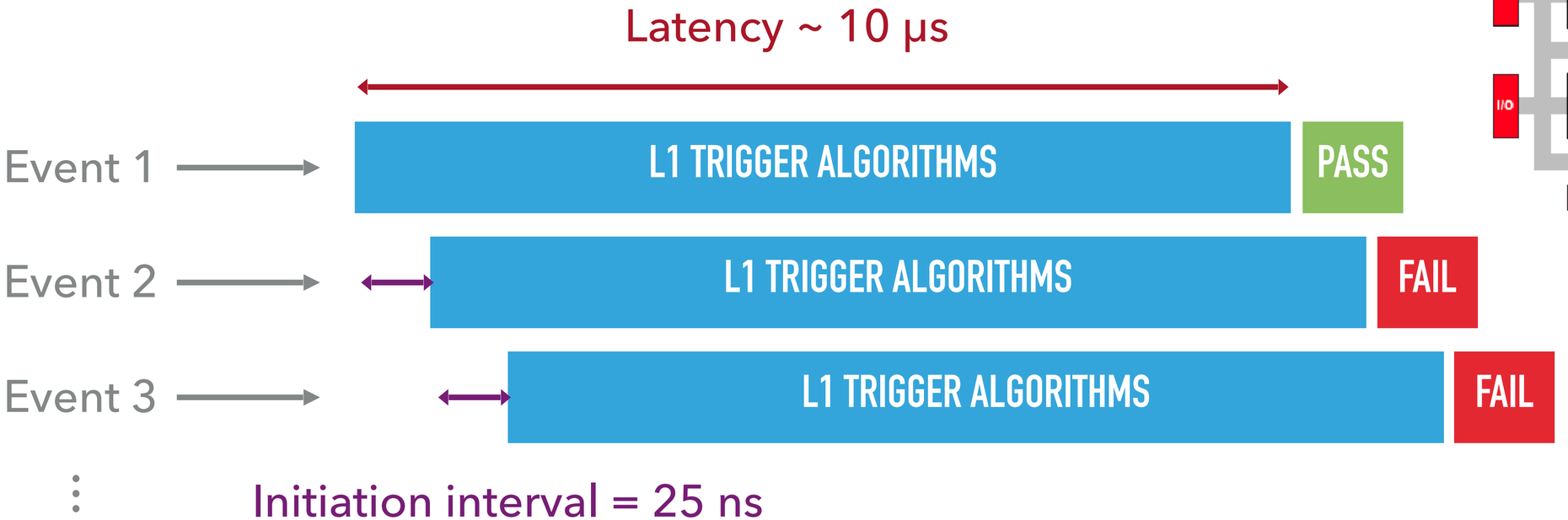




WHAT MAKES THIS HARD?

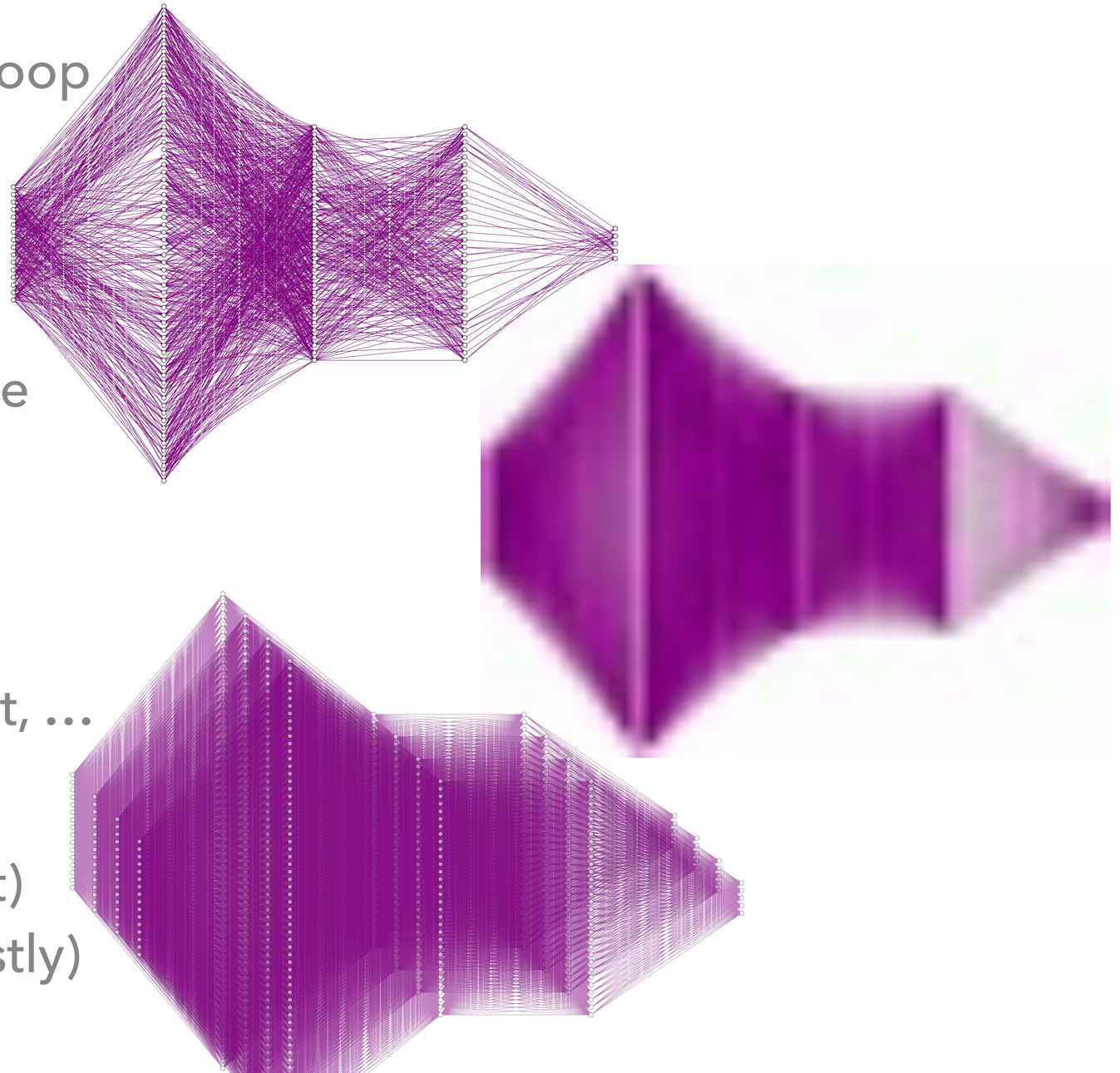
- Reconstruct all events and reject 98% of them in O(10) μs
  - Algorithms have to be <1 μs and process new events every 25 ns
- Latency necessitates all **FPGA** design
  - Algorithms have to fit on <1 FPGA
- How can we satisfy these constraints?





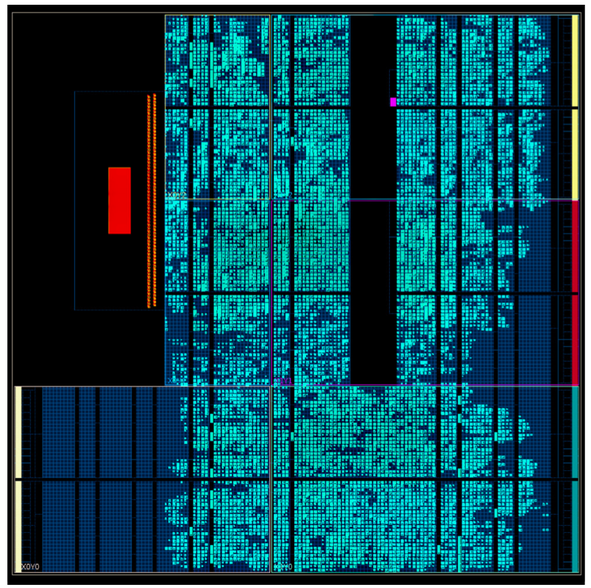
CODESIGN

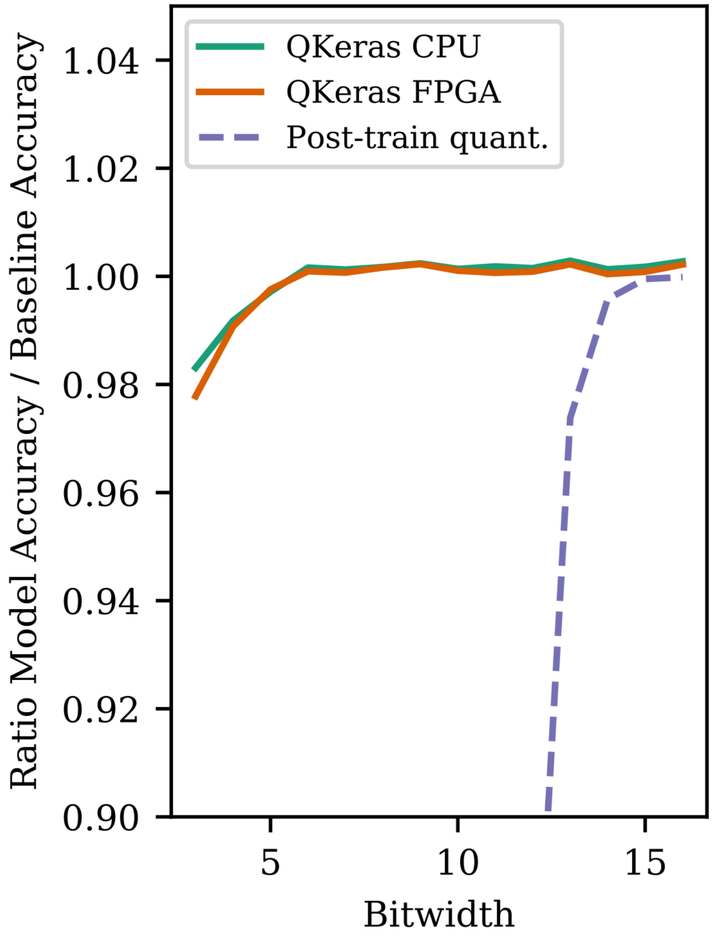
- **Codesign**: intrinsic development loop between ML design, training, and implementation
- Pruning
  - Maintain high performance while removing redundant operations
- Quantization
  - Reduce precision from 32-bit floating point to 8-bit fixed point, …
- Parallelization
  - Balance parallelization (how fast) with resources needed (how costly)



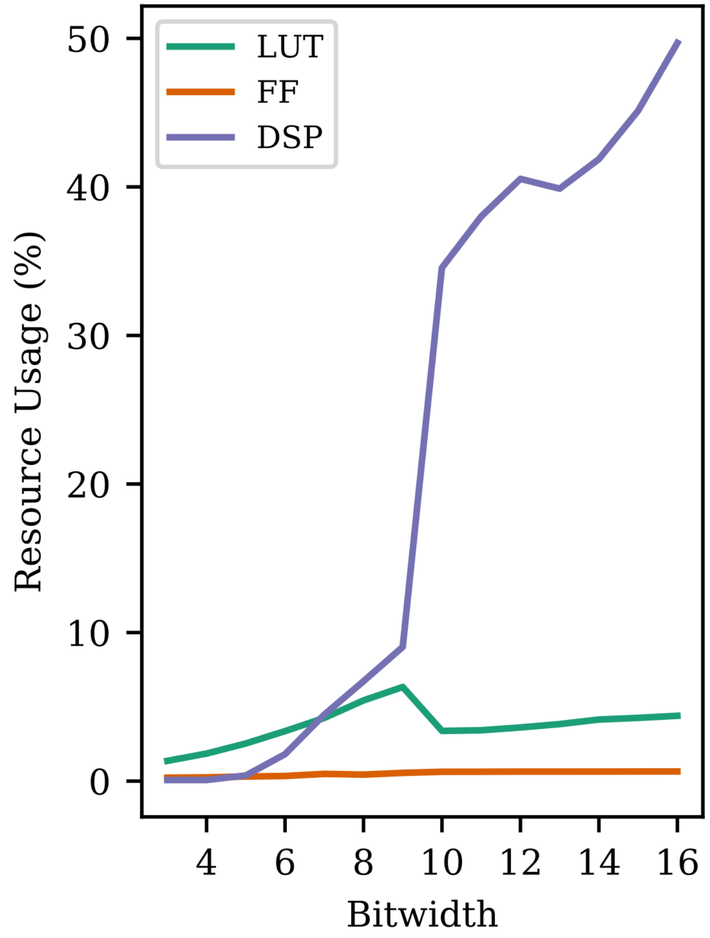
QUANTIZATION-AWARE TRAINING

- Small NN benchmark correctly identifies 5 classes of jets 70-80% of the time
- Full performance with 6 bits instead of 14 bits
- Much smaller fraction of resources



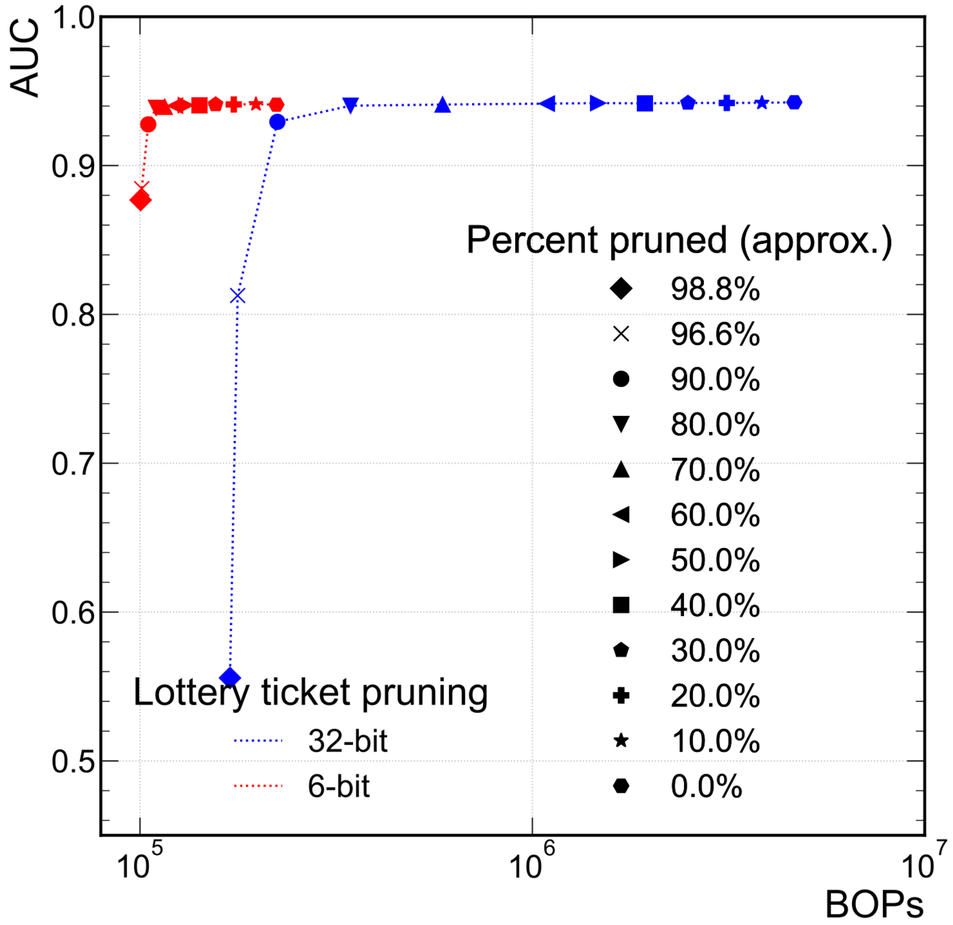


Xilinx VU9P



PRUNING + QUANTIZATION-AWARE TRAINING

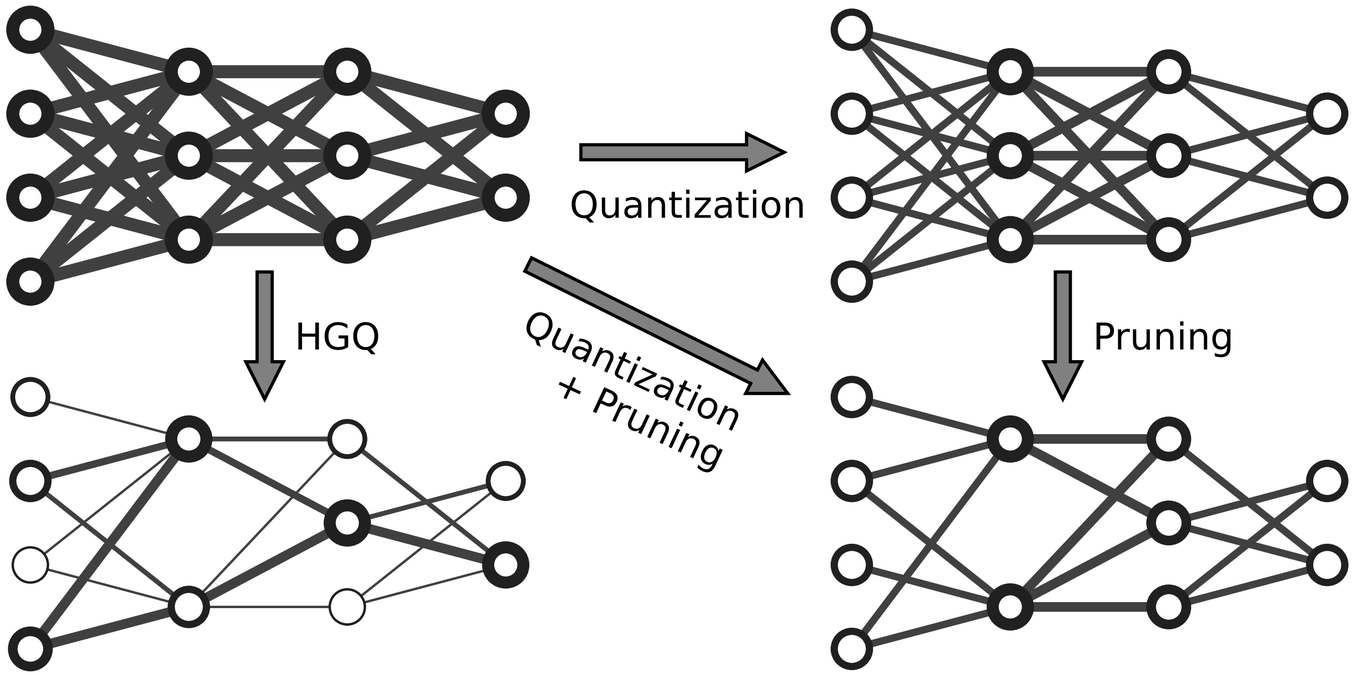
- Iterative pruning further reduces hardware complexity of a quantized model
  - Can remove 90% of parameters with almost no loss in performance!
- After pruning and quantization-aware training, can achieve $50\times$ reduction in bit operations compared to original 32-bit model

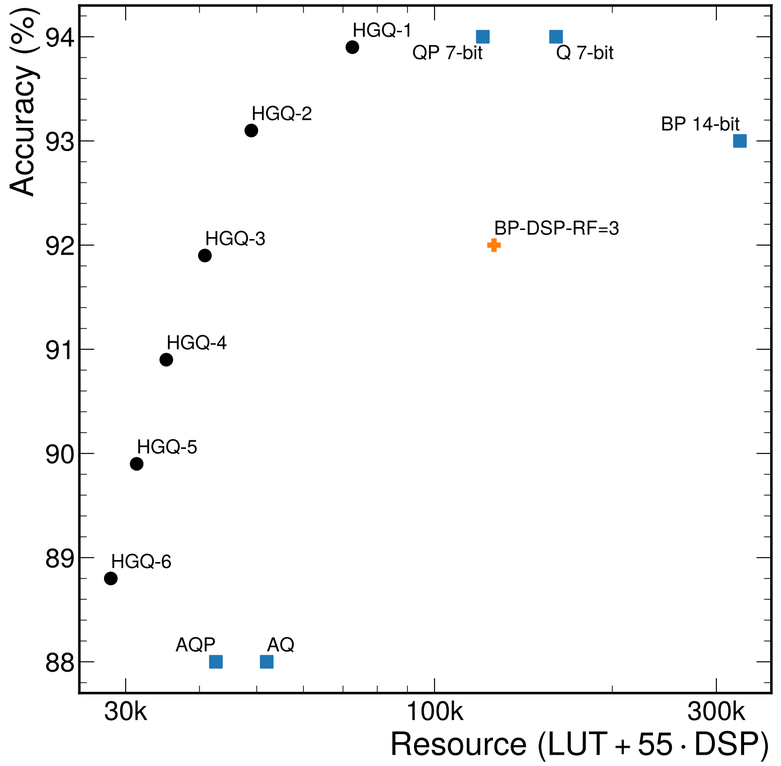


Bit operations (BOPs) definition: arXiv:1804.10969

HIGH-GRANULARITY QUANTIZATION

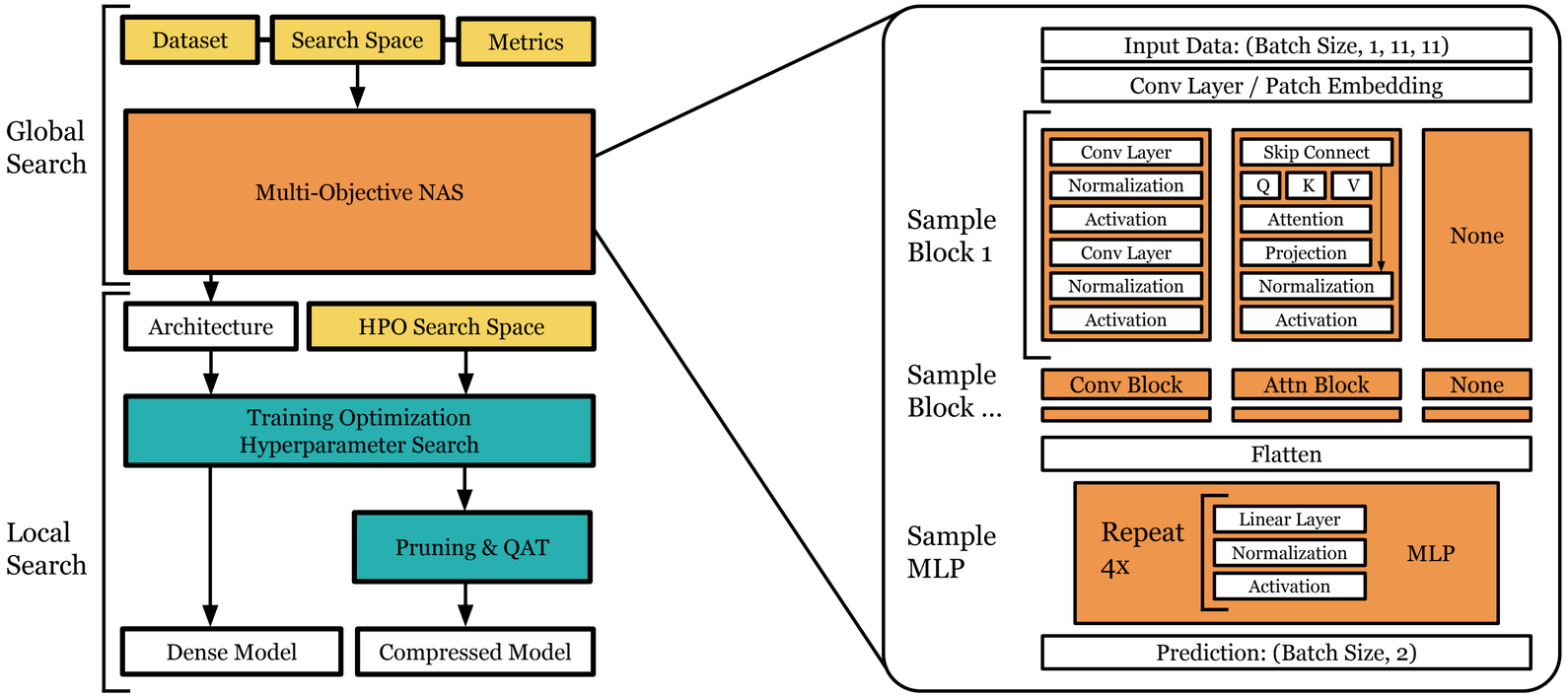
- High-granularity quantization (HGQ) further extends this paradigm by implementing gradient-based automatic bit width optimization at (up to) per-weight and per-activation granularity
- Demonstration on a CNN image classification (SVHN) benchmark

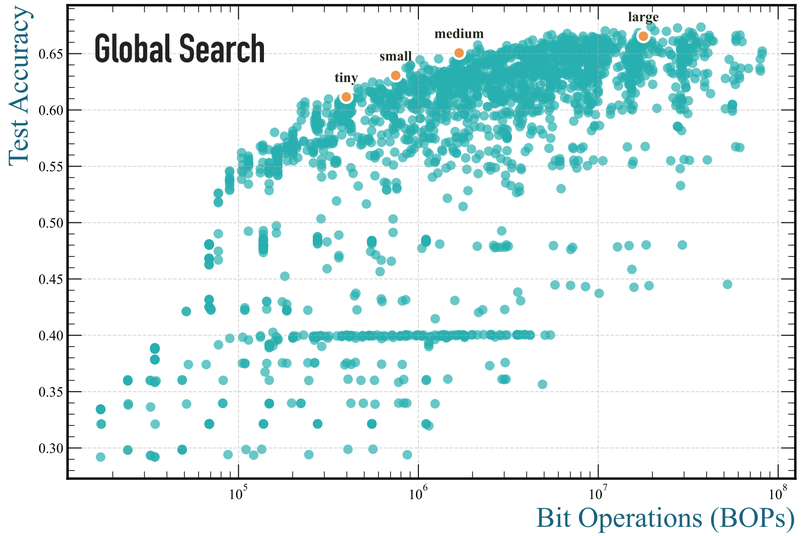


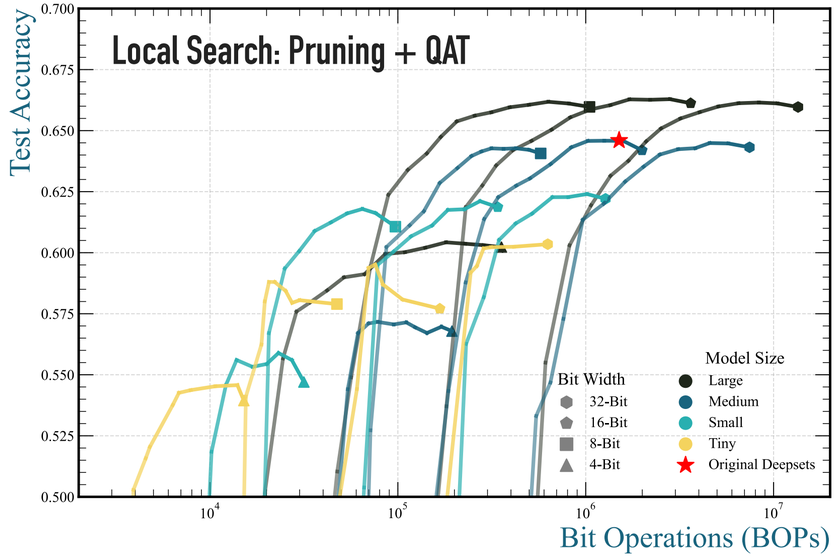


NEURAL ARCHITECTURE CODESIGN

- Combine neural architecture search with hardware-aware codesign to find ***Pareto-optimal front***
- Example: set-based jet tagging model in FPGAs [arXiv:2402.01876]

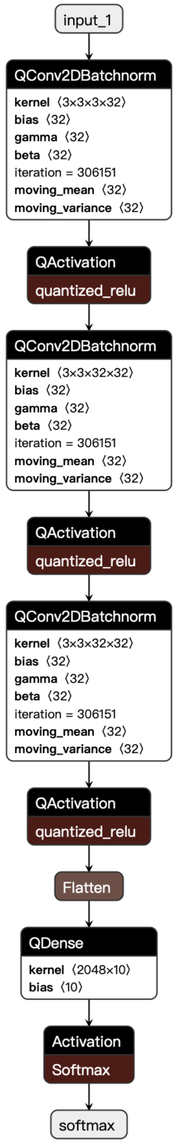




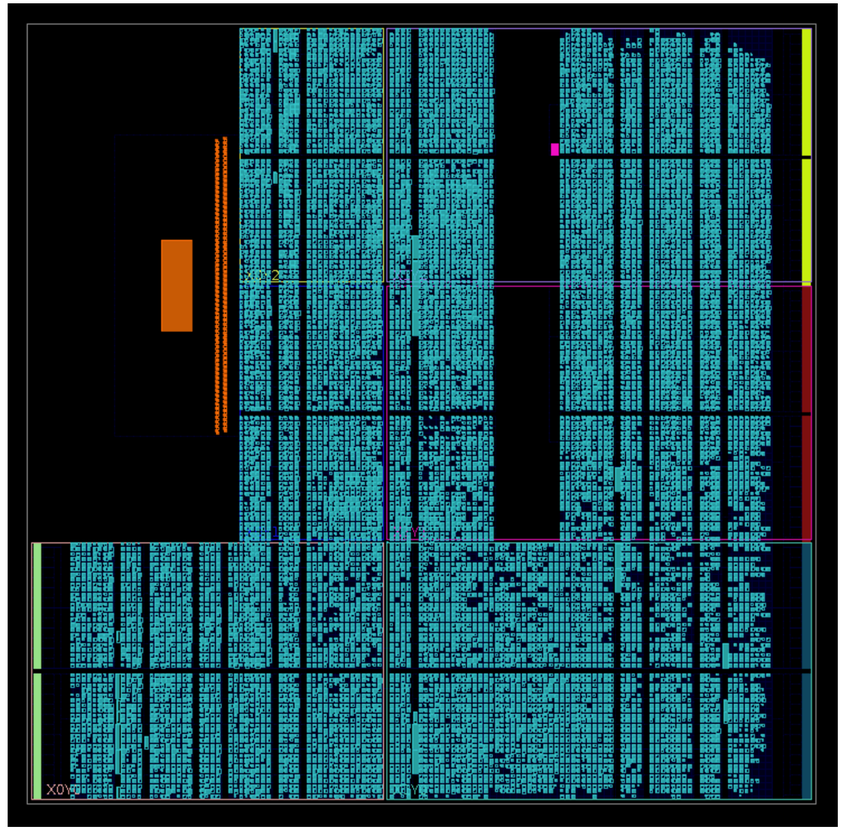


PROGRAMMING HARDWARE (FPGAS)

- How do we implement an optimized AI model (e.g., in QONNX) for an FPGA?

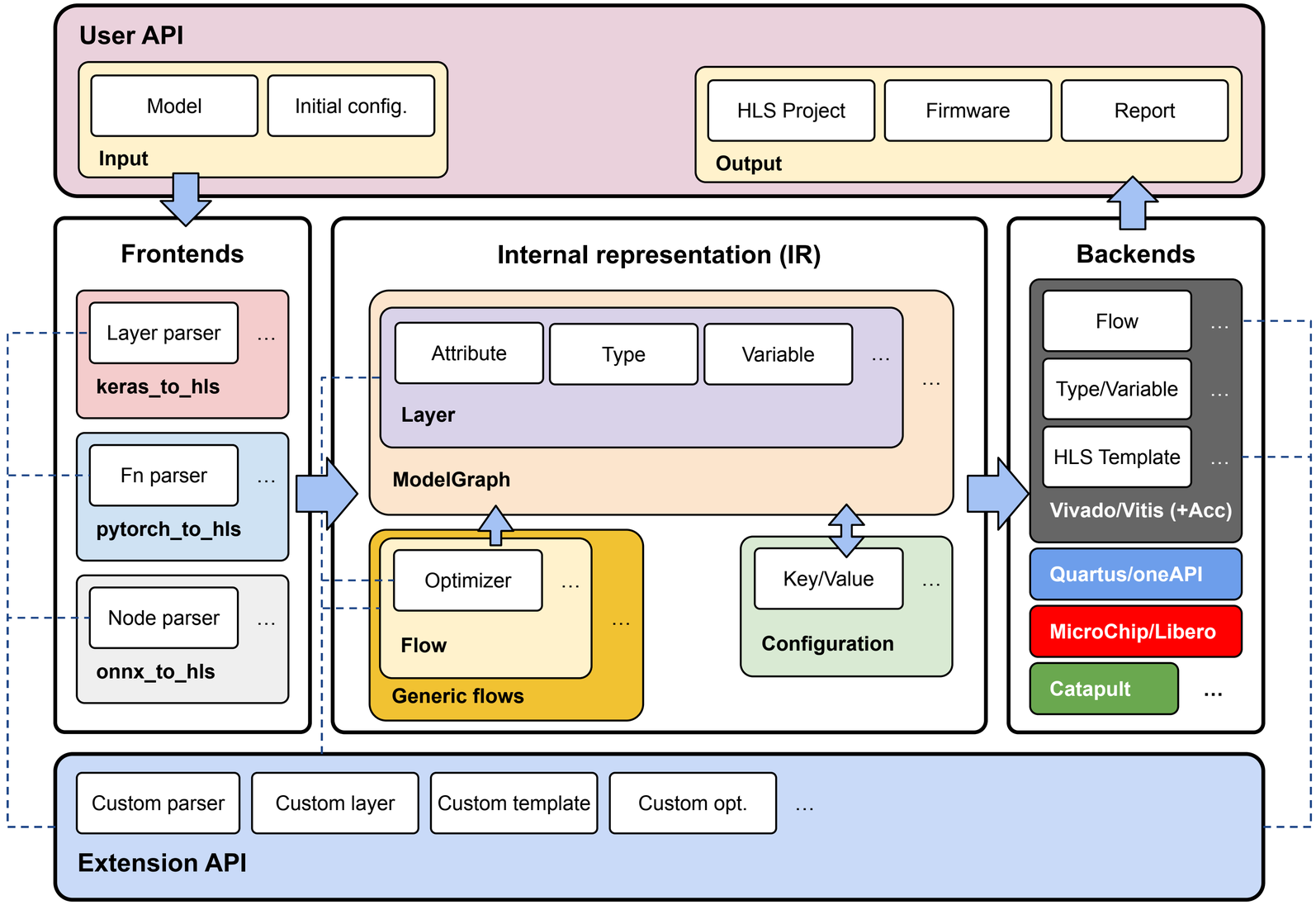


High-Level Synthesis



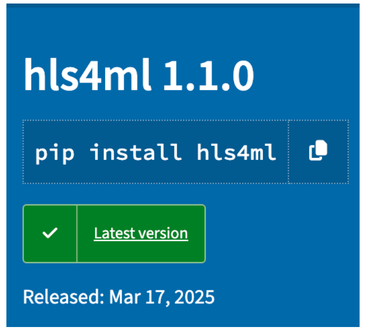
HLS4ML

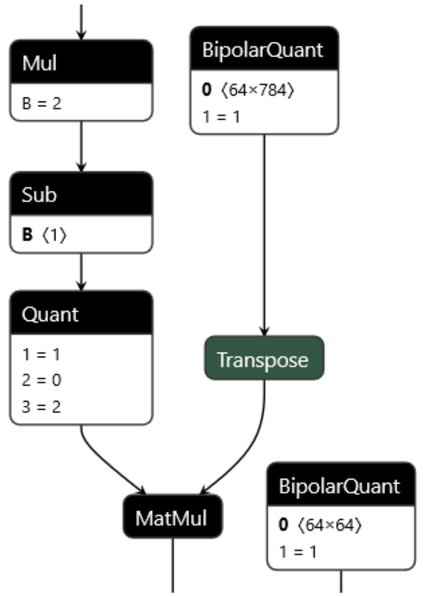
- hls4ml for scientists or ML experts to translate ML algorithms into RTL firmware
- NN model is parsed through the front ends into the IR, modified by the optimizer, and passed to the back ends, which create the output

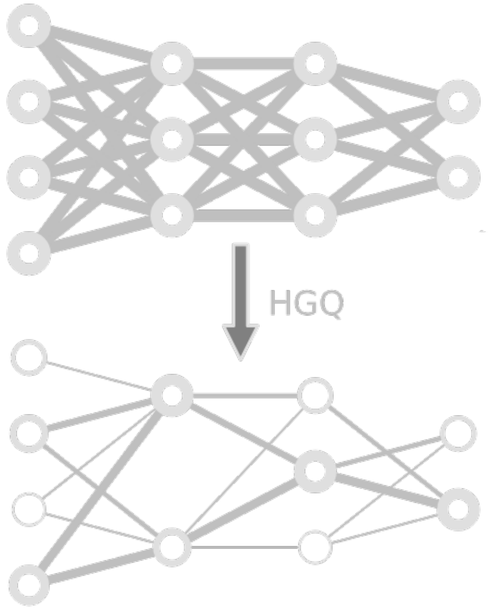


LATEST FEATURES IN HLS4ML 1.1.0

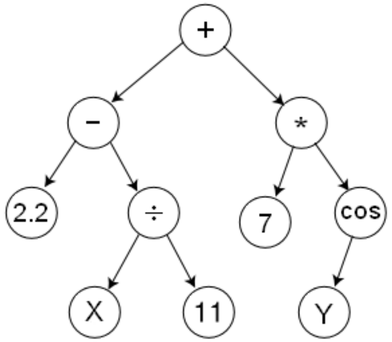
- QONNX front end
- Automatic precision inference
- Support for HGQ models
- Symbolic regression, hardware-aware optimization API, and more…








```
config = hls4ml.utils.config_from_keras_model(model, default_precision='ap_fixed<16,6>', granularity='model')
config['LayerName'] = {
    # Infer all types of these layers
    'first_layer': {
        'Precision': 'auto',
    },
}
hls_model = hls4ml.converters.convert_from_keras_model(
    model, hls_config=config, io_type=io_type, output_dir=odir, backend=backend
)
```

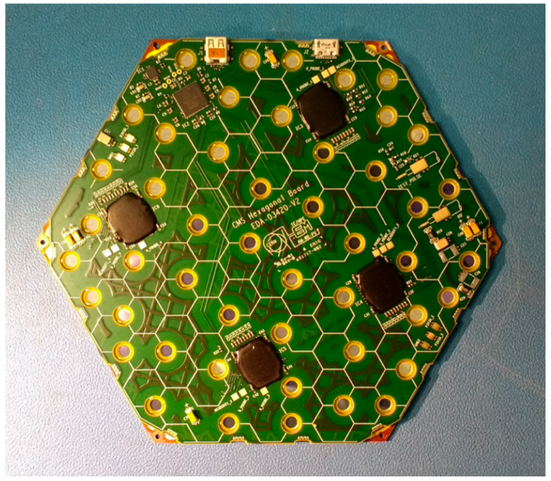


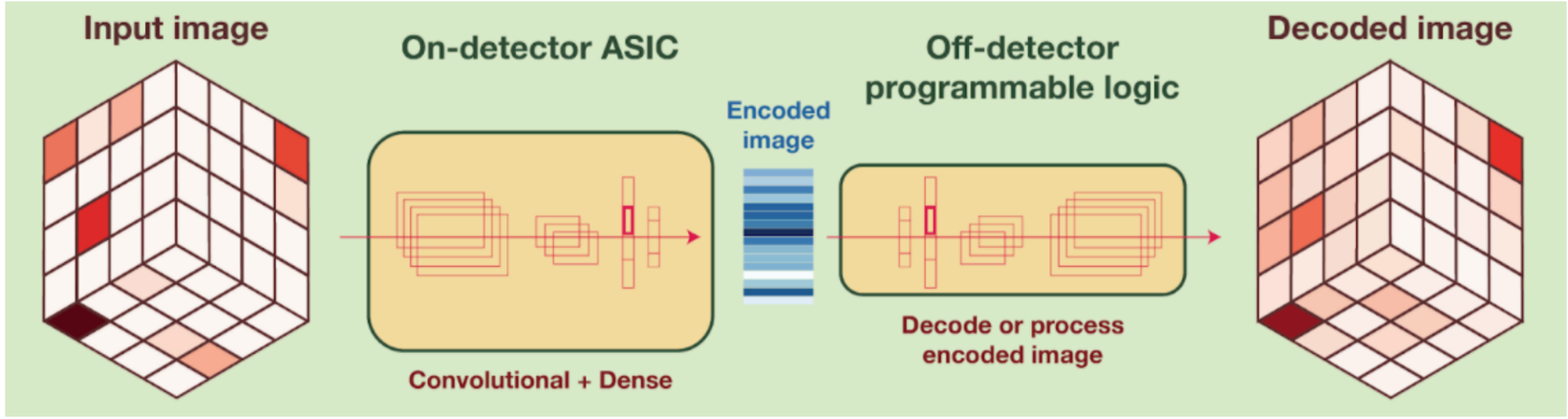
APPLICATION: SENSOR DATA COMPRESSION ON ASIC

- **Autoencoder** on the detector front-end for **data compression:** ~100 ns latency and area/power/radiation tolerance requirements

- **Reconfigurable ASIC** to address: **evolving LHC conditions** (beam-related), **detector performance** (noise, dead channels), and **updated performance metrics** (resolution, new physics signatures)

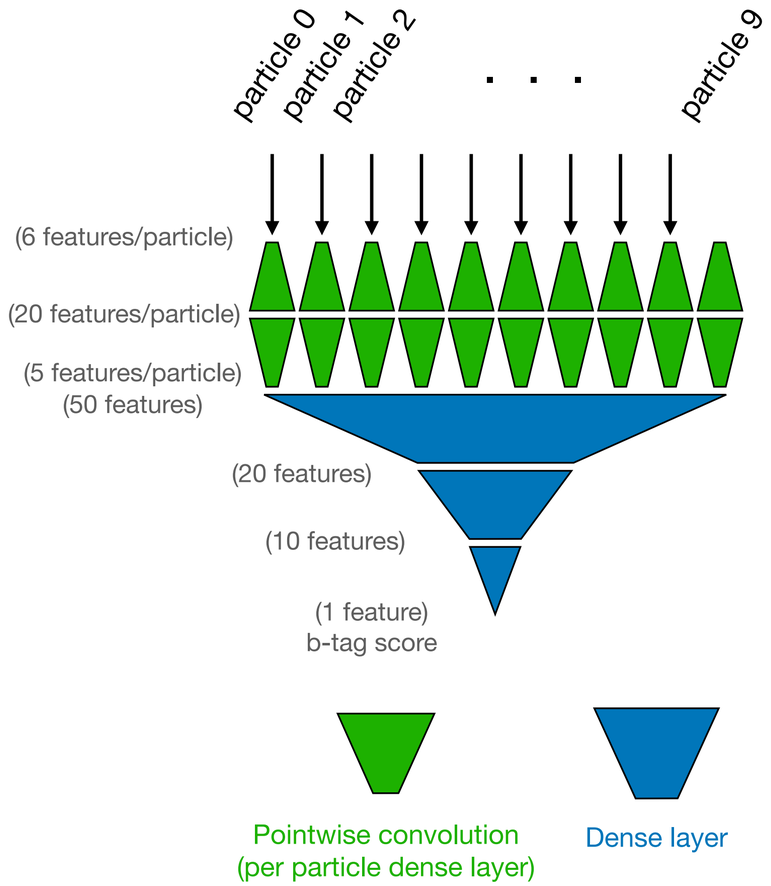
8″ hexagonal silicon module
(1 out of ~27,000)

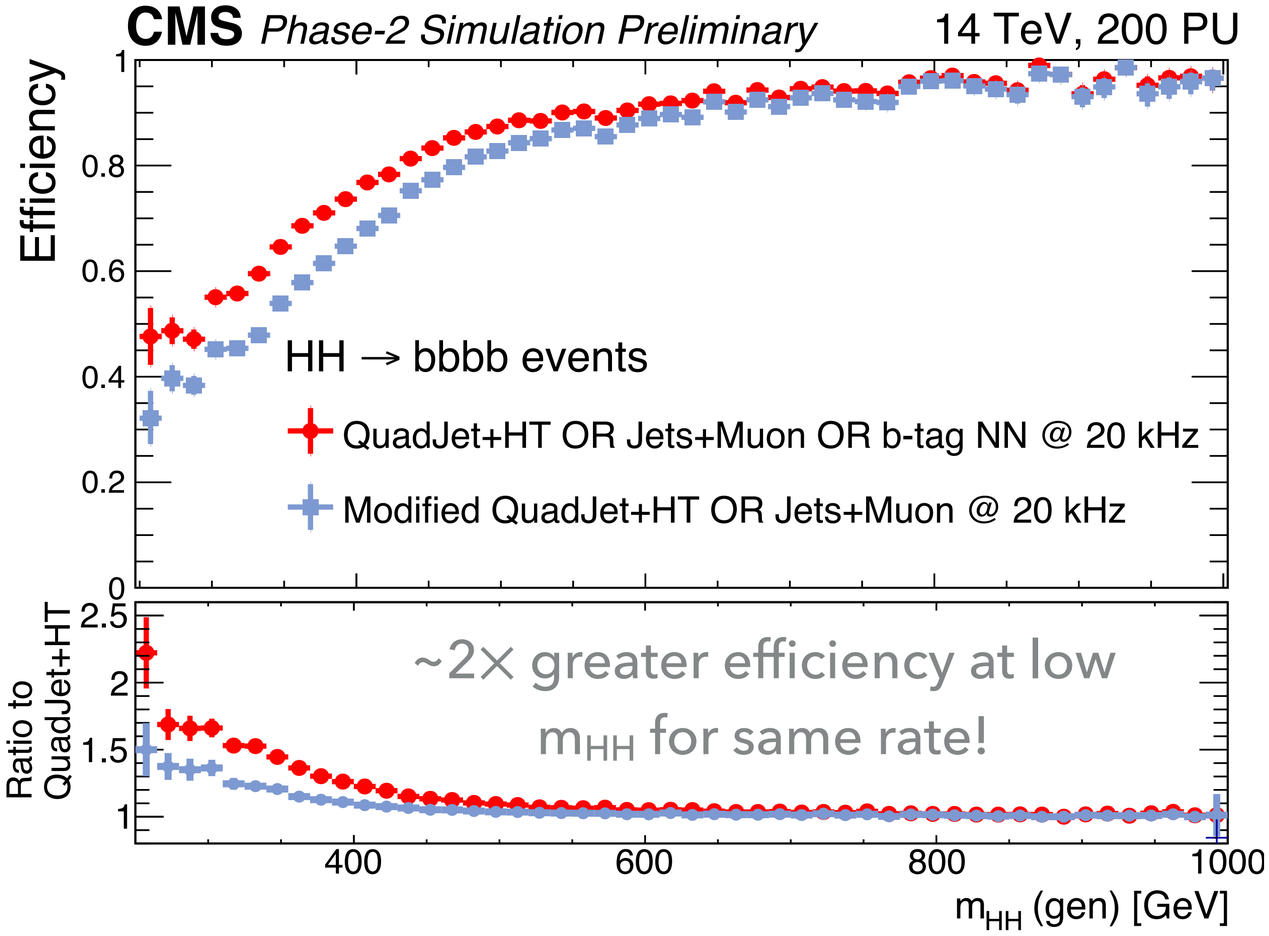




APPLICATION: JET TAGGING @ L1

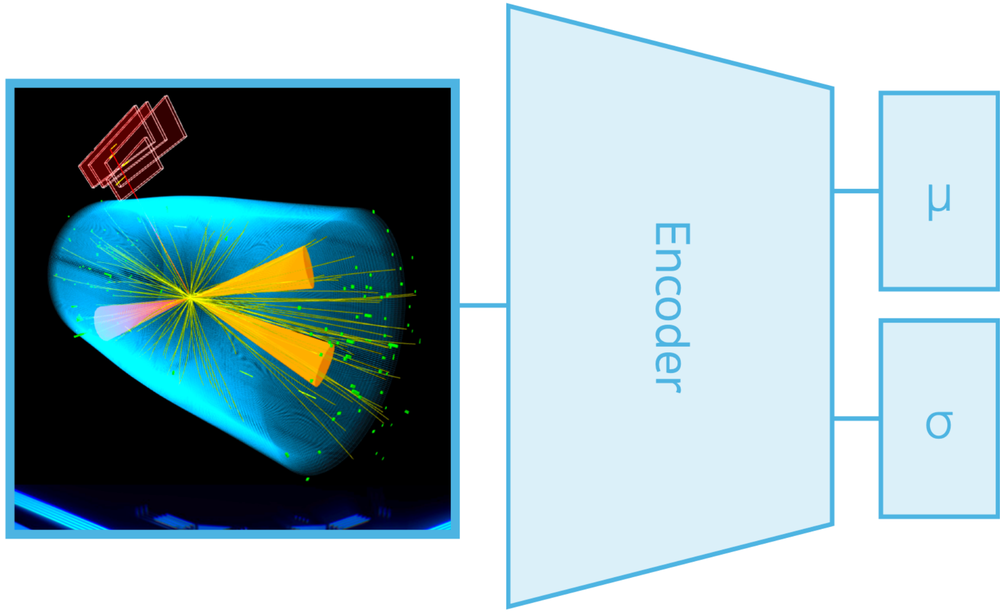
- Upgraded HL-LHC level-1 track trigger information enables b-tagging with a ***neural network*** to improve the $HH \rightarrow 4b$ search
  - Input features for 10 particles within each jet: particle type, momentum, and vertex information





APPLICATION: ANOMALY DETECTION AT 40 MHZ

- Challenge: if new physics has an unexpected signature that doesn't align with existing triggers, precious signal events will be discarded
- Can we use unsupervised algorithms to detect non-SM-like anomalies?
  - Autoencoders (AEs): compress input to a smaller dimensional latent space then decompress and calculate difference
  - Variational autoencoders (VAEs): model the latent space as a probability distribution; possible to detect anomalies purely with latent space variables

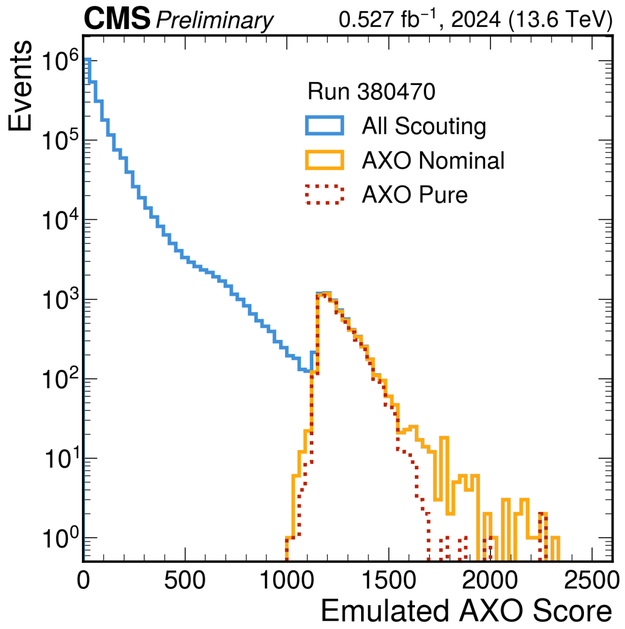


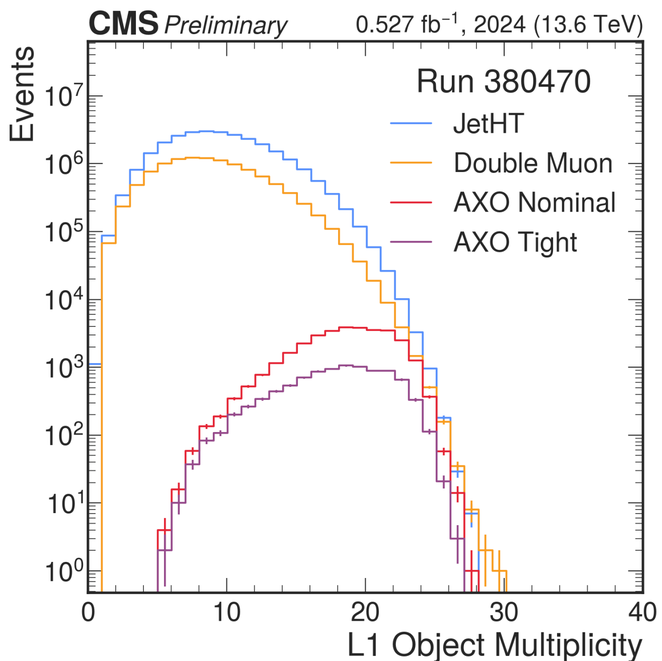
Key observation: Can build an anomaly score from the latent space of VAE directly! No need to run decoder!

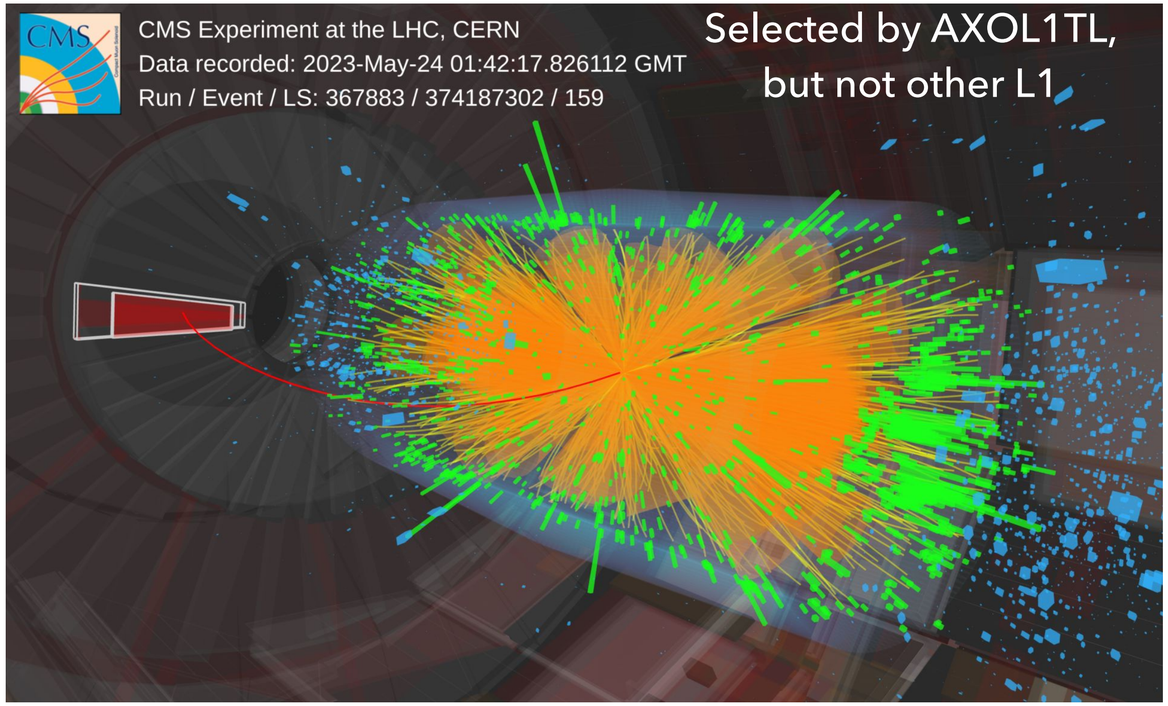
$$R_z = \sum_i \frac{\mu_i^2}{\sigma_i^2}$$

APPLICATION: ANOMALY DETECTION AT 40 MHZ

- AXOL1TL anomaly detection algorithm for the trigger based on a variational autoencoder
- Selects unique events relative to existing triggers
- Preference for high multiplicity events



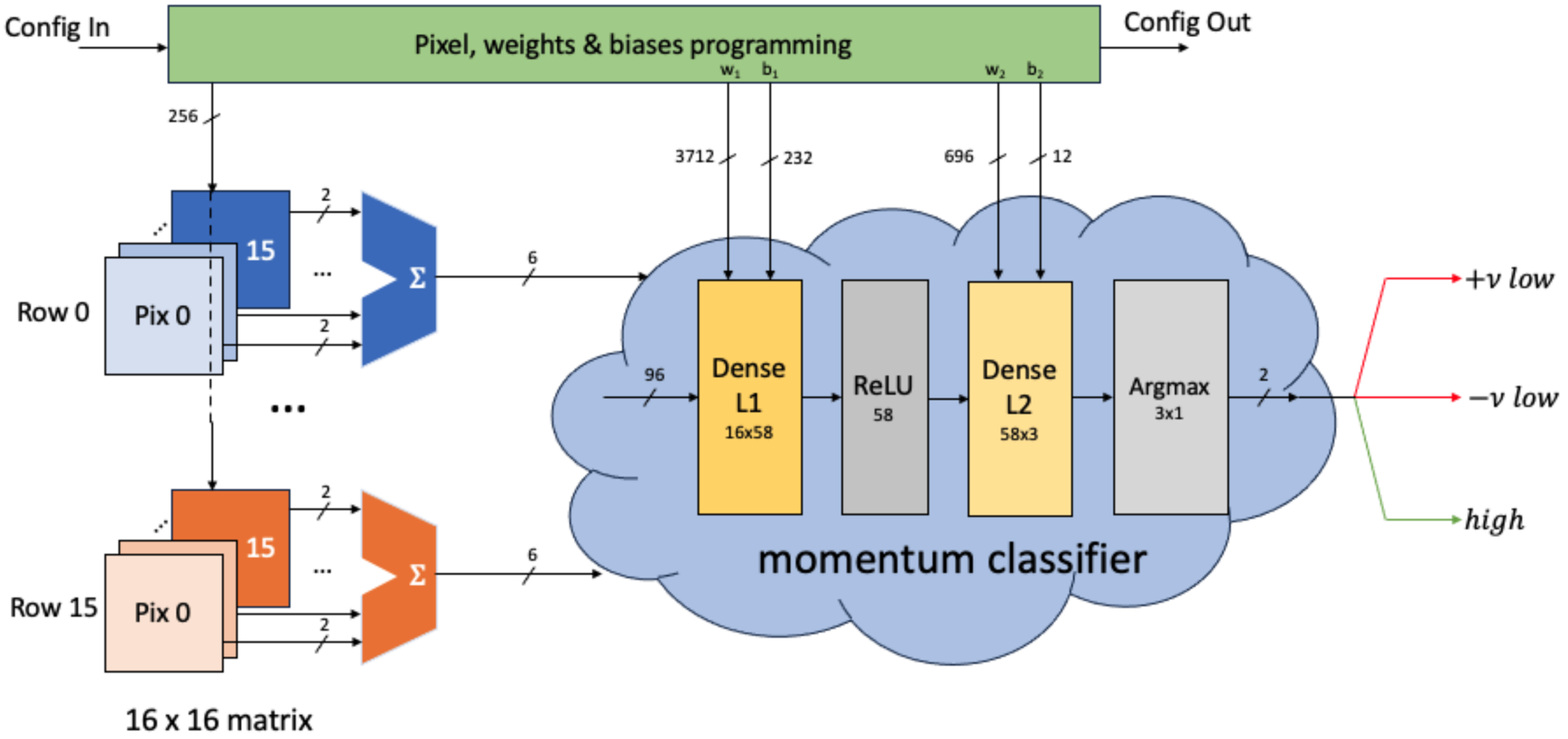


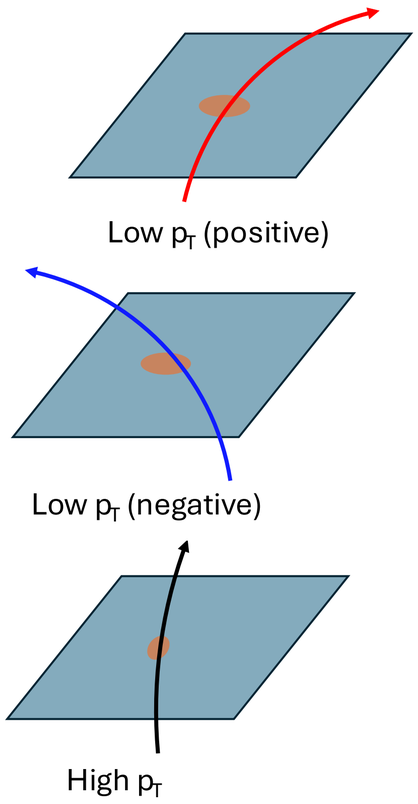


APPLICATION: SMART PIXELS

Mach. Learn. Sci. Tech. 5 (2024) 3, 035047
arXiv:2406.14860

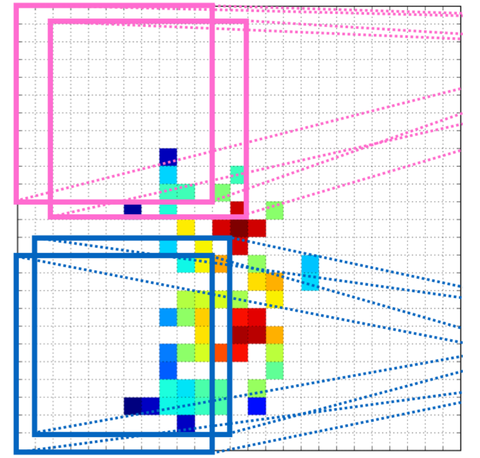
- Locally customized neural network for p_T filtering of charged particles in sensor readout can enable data reduction by 55-75% with low power consumption of ~300 μW and latency of 3.9 ns
- hls4ml + Catapult AI to ASIC workflow

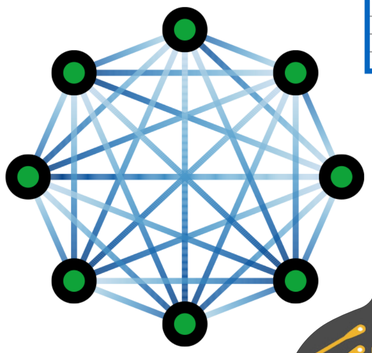


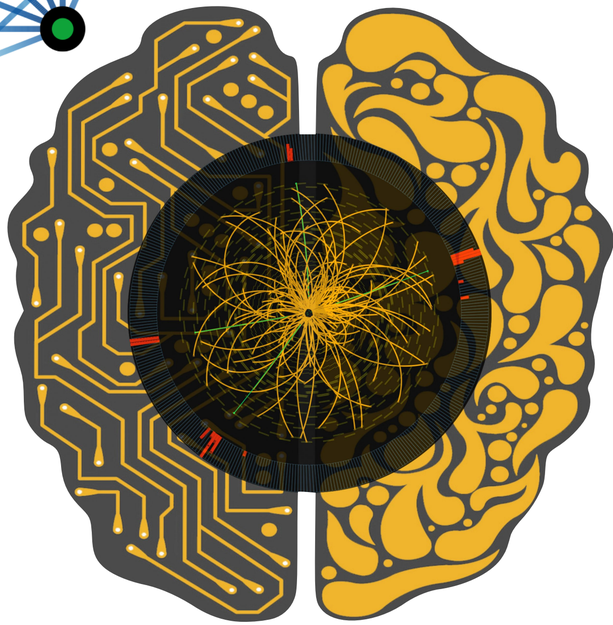


SUMMARY AND OUTLOOK

- ML allows us to better reconstruct our data and save potentially overlooked data
- ***Codesign*** principles can enable ML on hardware with stringent constraints
- Community (fastmachinelearning.org, e-group hls-fml@cern.ch), Institute (a3d3.ai) and CERN Project (NextGen Triggers) developing open-source tools and techniques to enable this
  - hls4ml: expanding open-source toolkit for translating ML into hardware aimed at trigger applications and more…
- Applications range from jet tagging to tracking, and more!
  - Enhance **future particle physics program**







JAVIER DUARTE

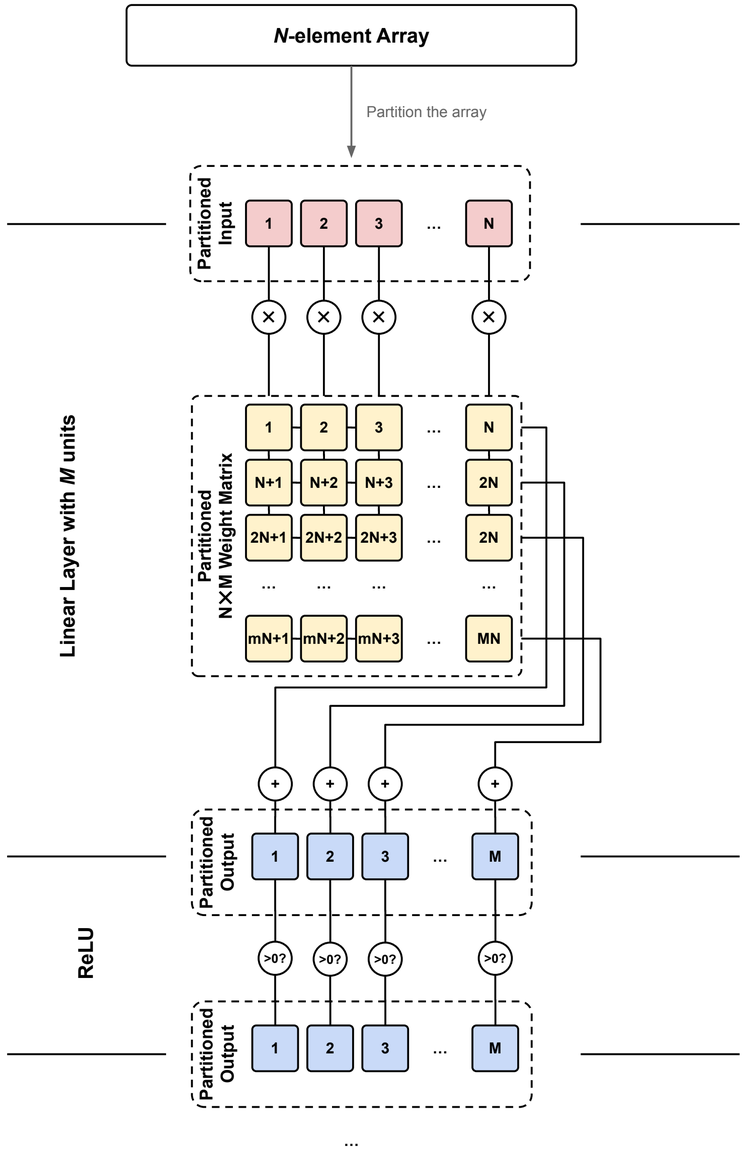
ML4FE WORKSHOP

U HAWAII, MAY 19, 2025

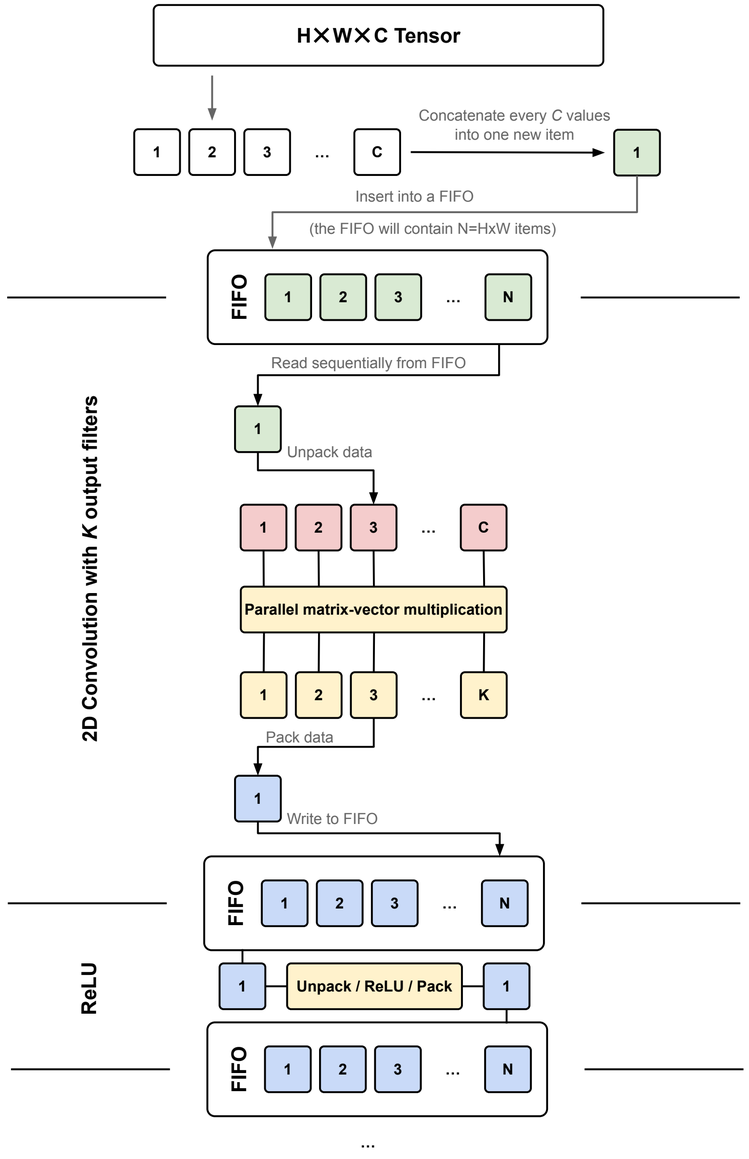
BACKUP

PARALLEL OR STREAMING INPUTS

- hls4ml supports two modes for parallel or streaming inputs



(a) io_parallel



(b) io_stream