Realtime Anomaly Detection with the CMS Level-1 Trigger

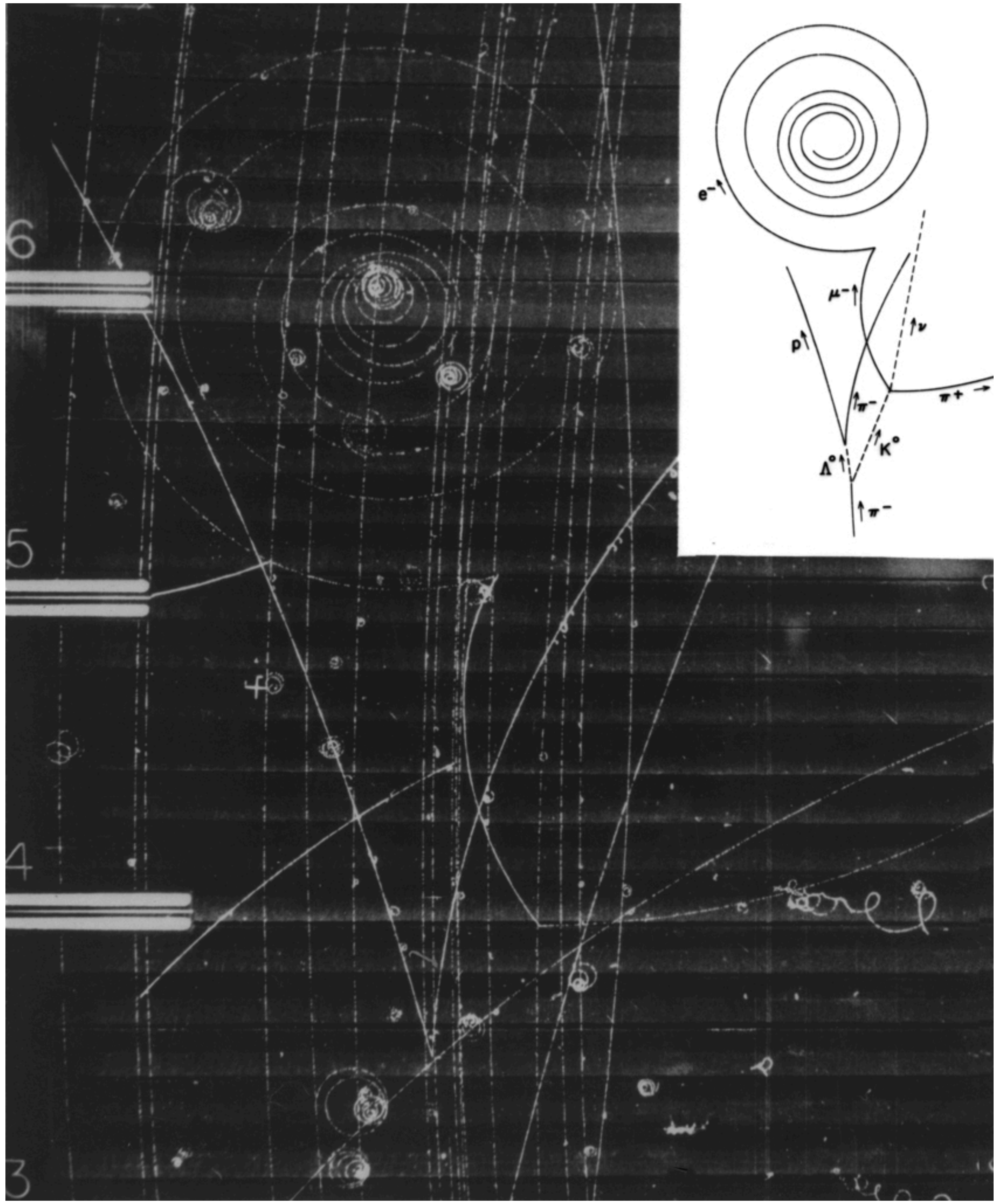
Artur Lobanov (Universität Hamburg)
on behalf of the CMS Collaboration

UHH
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Data-taking at the LHC

. . . PARTICLE PHYSICS 60 YEARS AGO

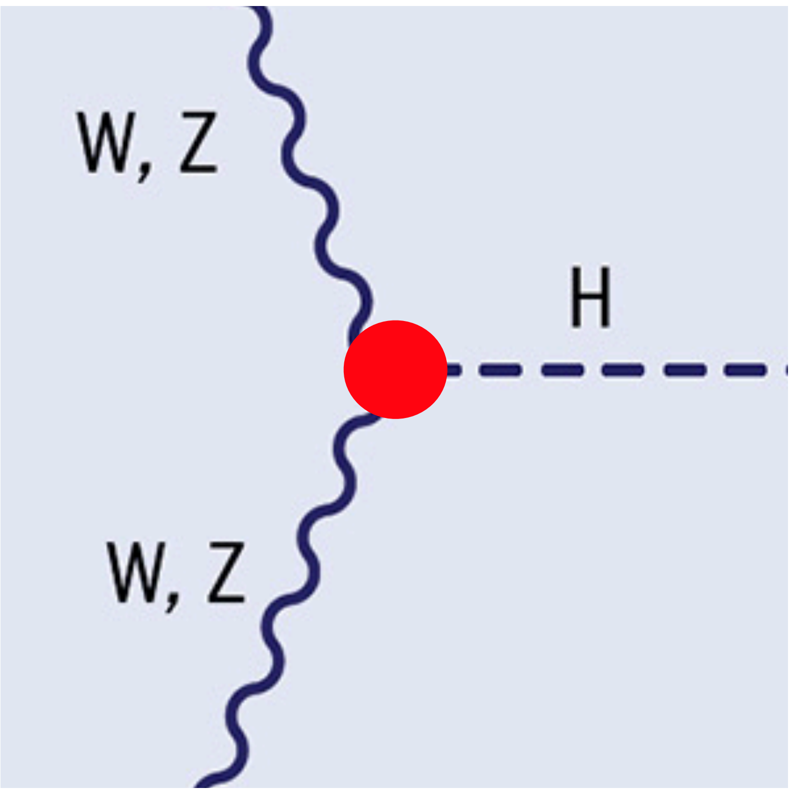
Actually taking pictures of particles



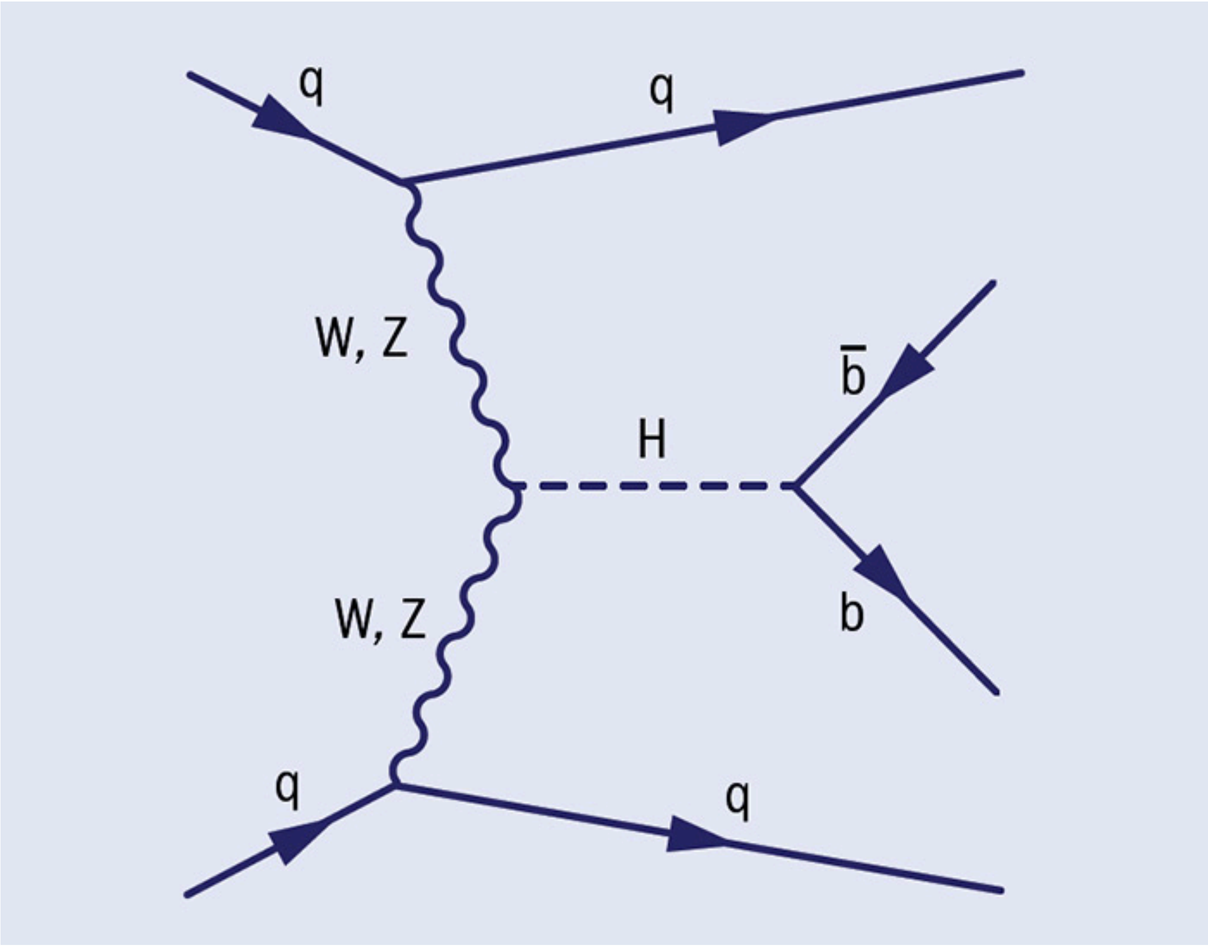
Bubble chamber event: Muonic decay of a neutral K meson

What we do today @ the Large Hadron Collider (LHC)

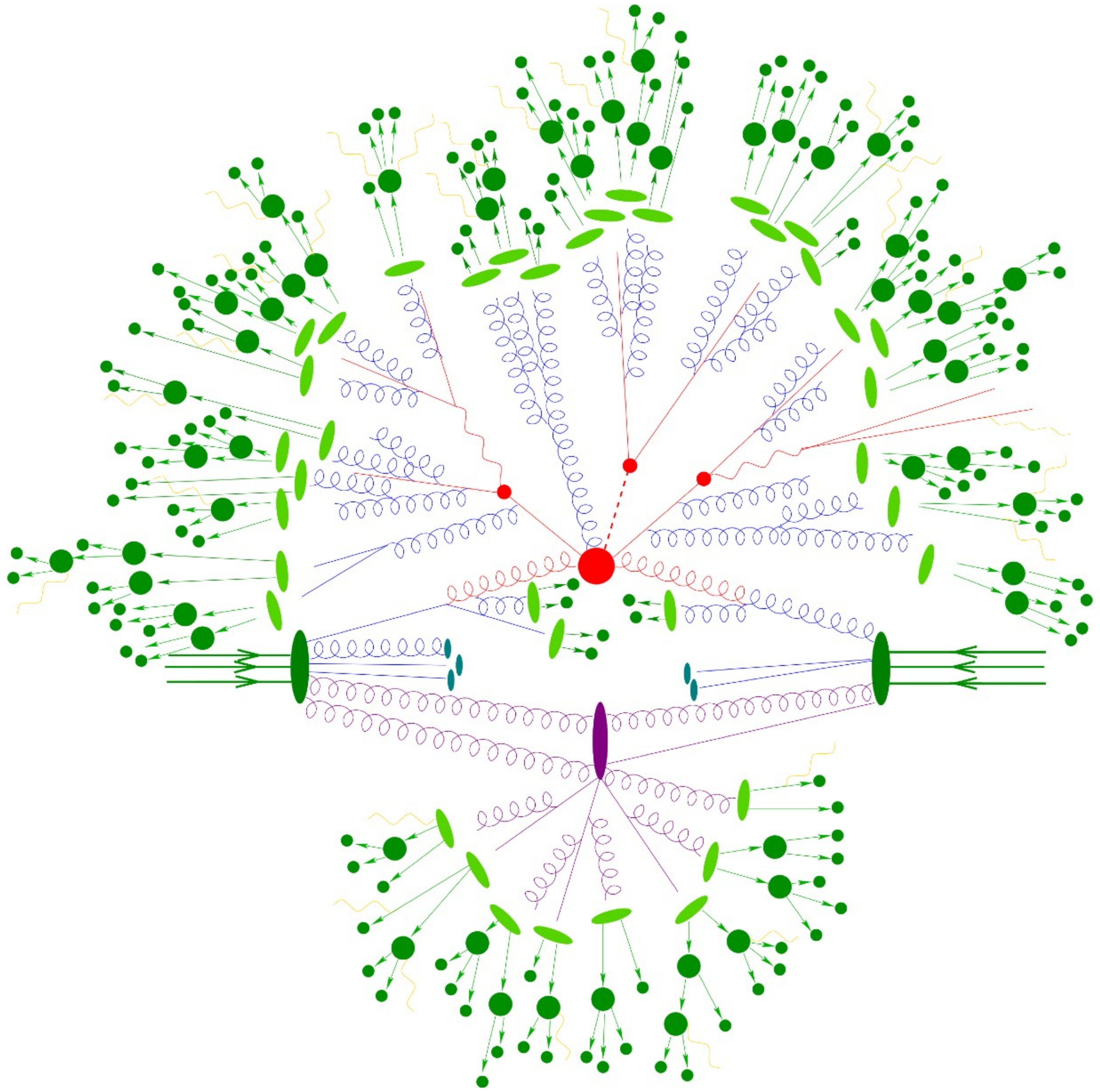
What we want to study



How collisions help us



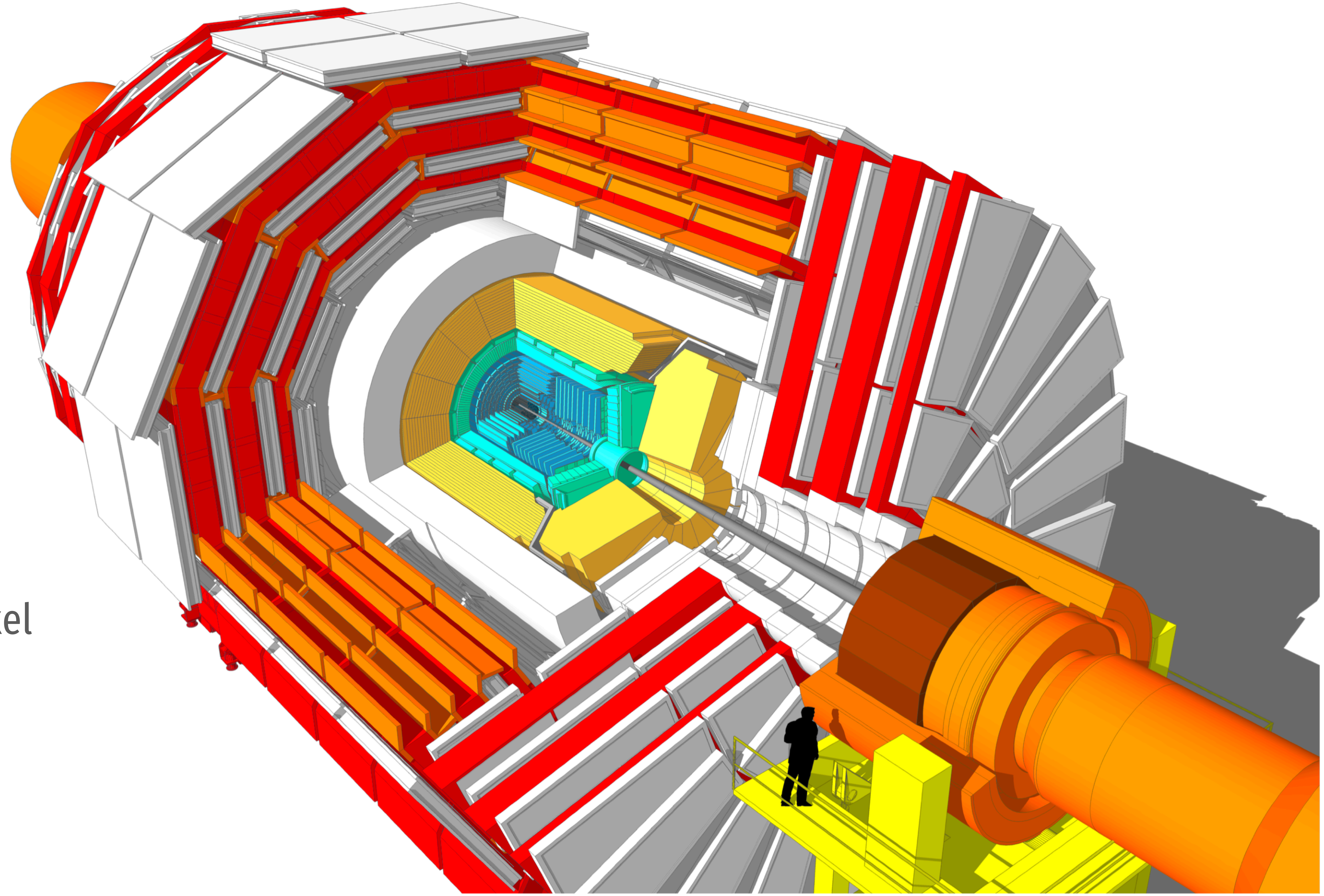
What actually happens



Production of a Higgs boson (H) through Vector Boson Fusion (W/Z)

Partons and hadronization

THE CMS EXPERIMENT AT THE LHC

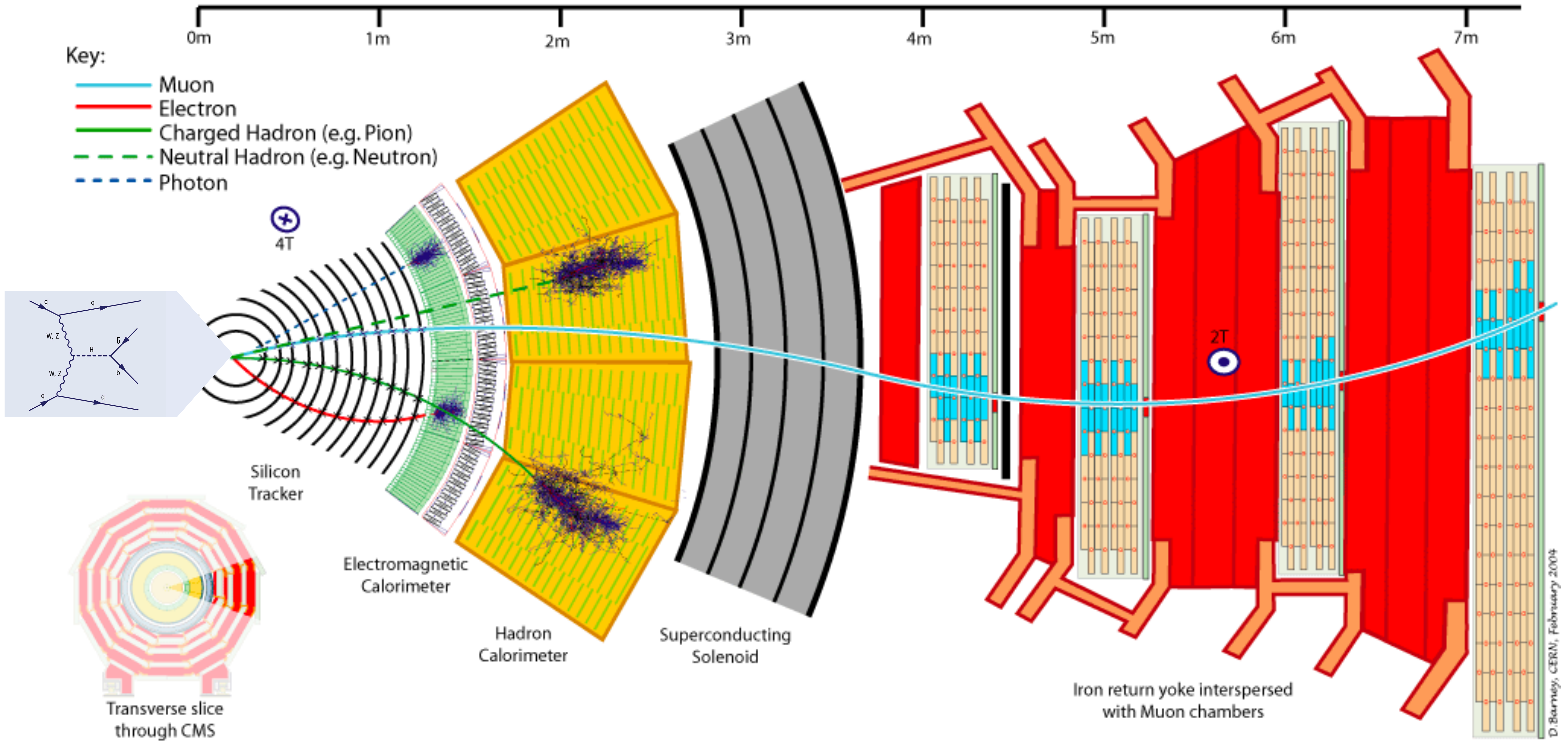


The CMS experiment:

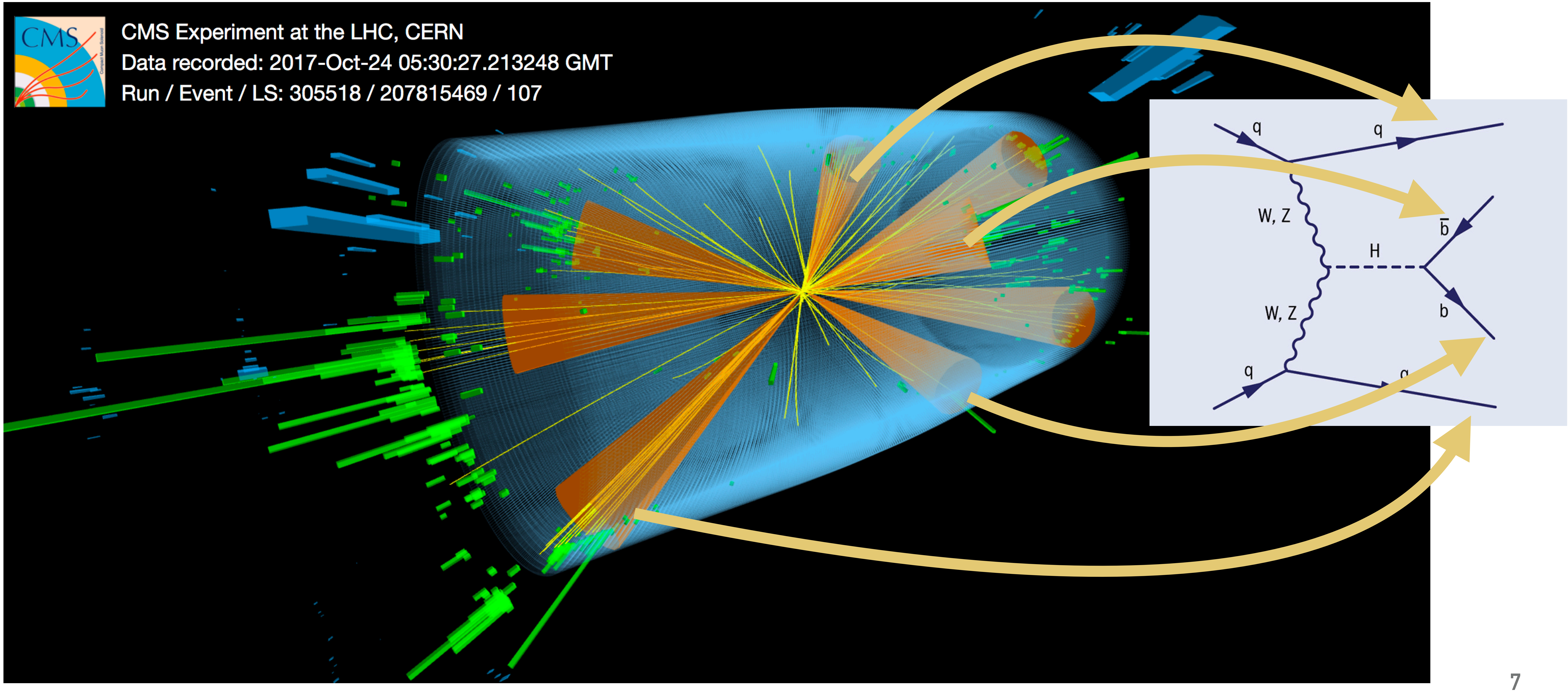
LHC camera with 100 Mpixel

How CMS sees particles

Different particle types can be measured with different detectors



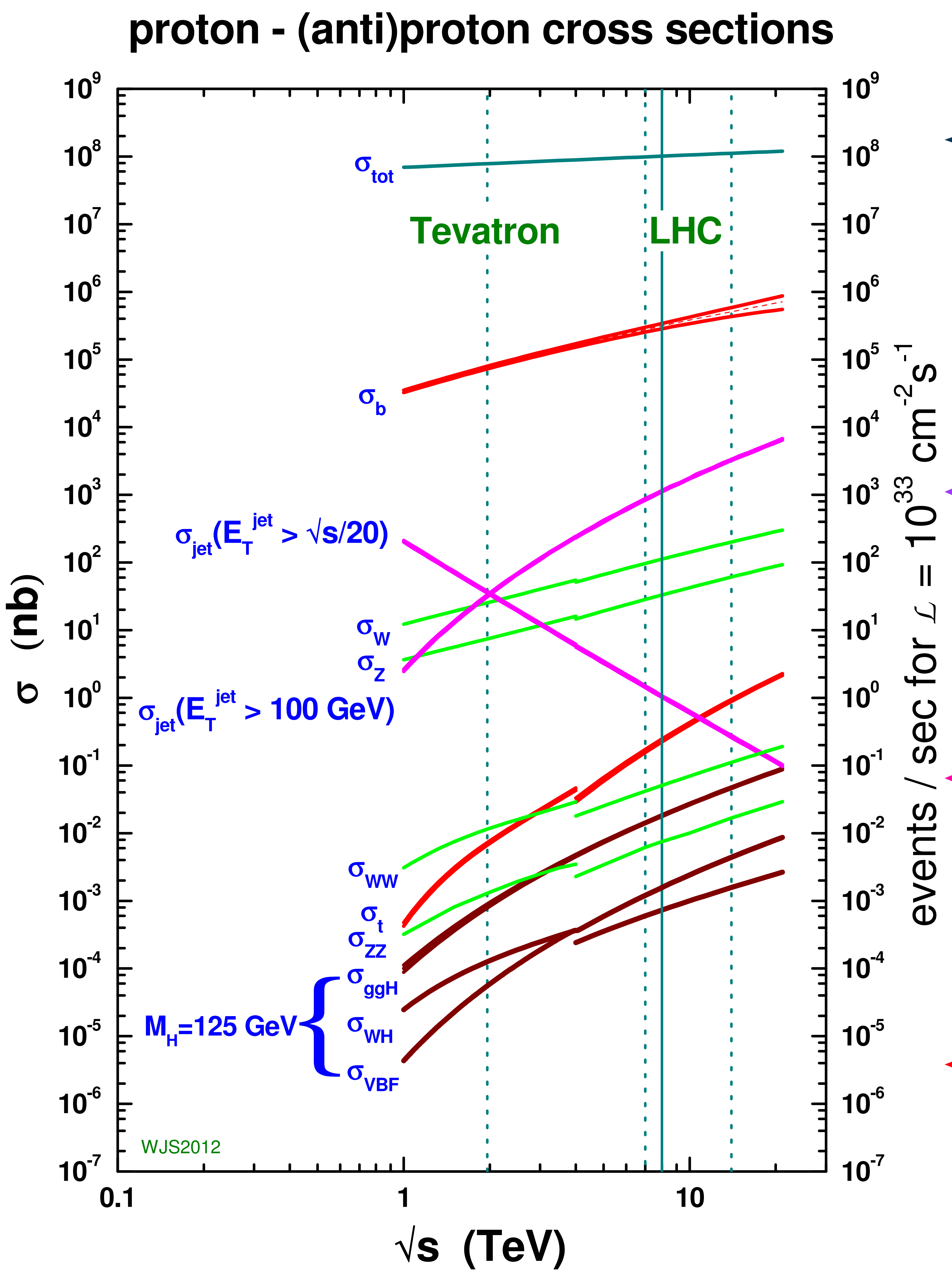
How LHC collisions LOOK like



Event selection: Trigger

Searching for the needle in the LHC Haystack

Probability decreasing ----->



LHC: 40 Million proton collisions per second

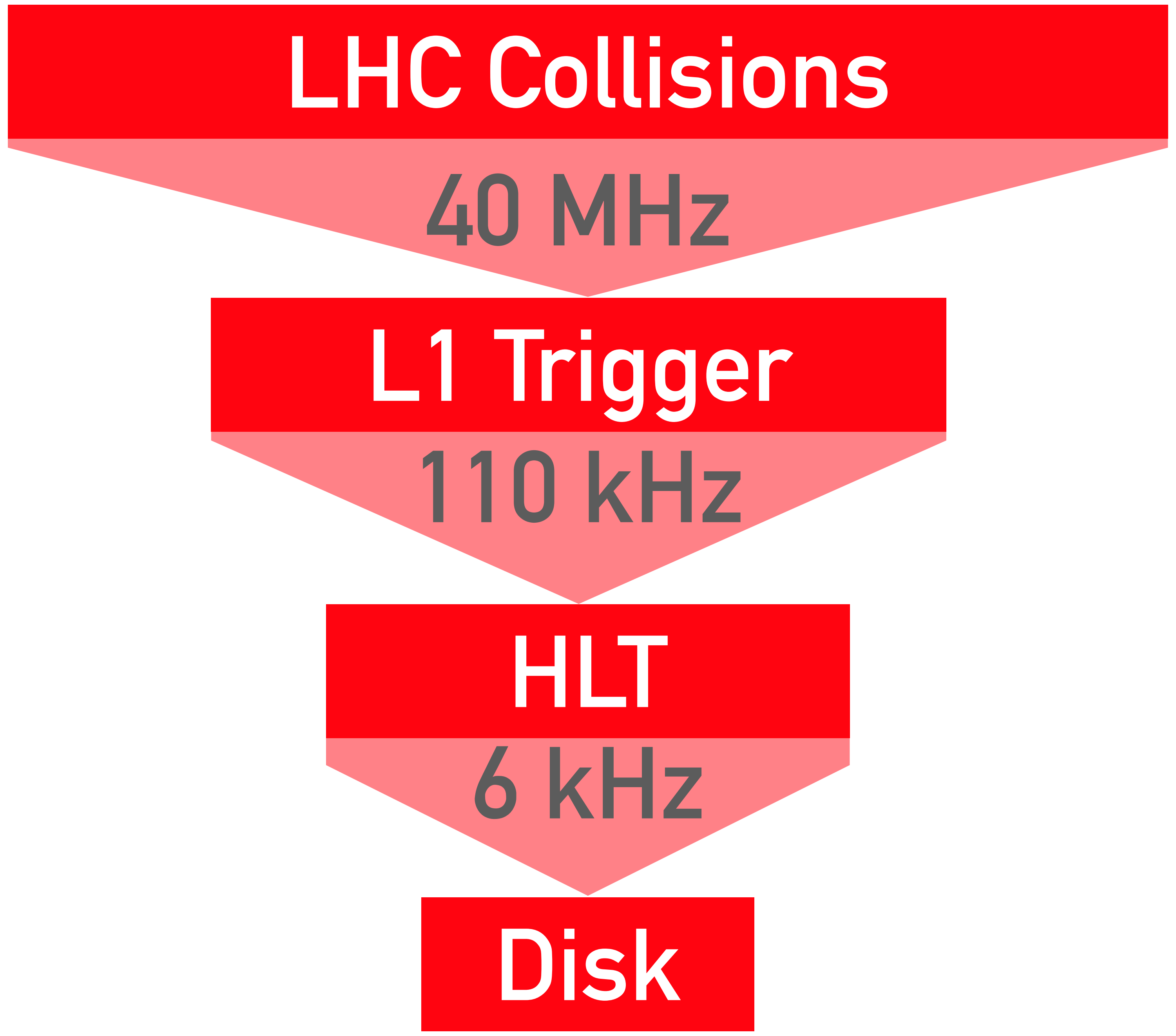
1000 W/Z bosons produced / second

1 Higgs boson is produced / second

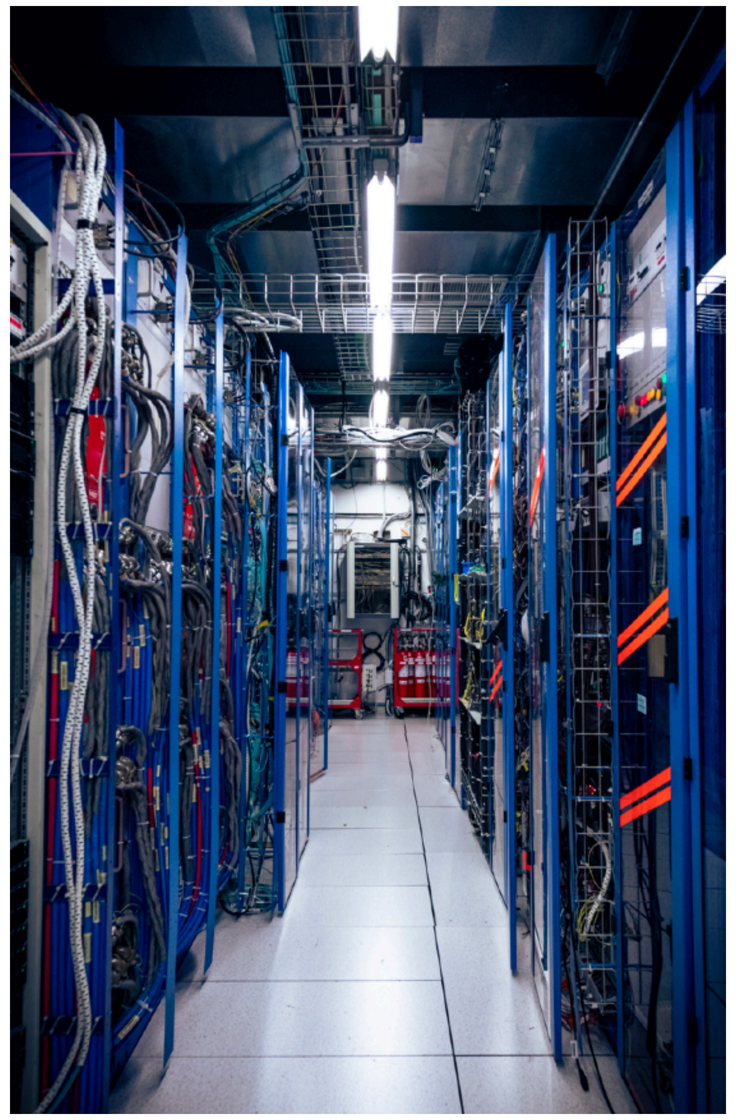
New physics (= Anomalies) hiding here?

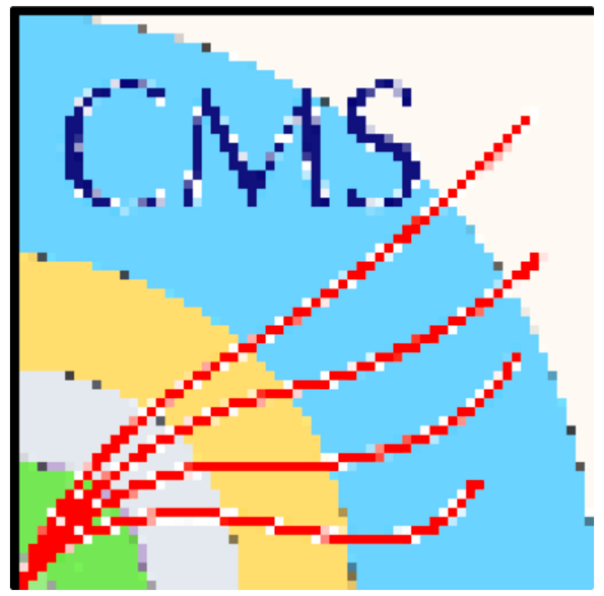
* LHC values from 2010 -> now higher luminosity

The CMS Trigger system



- Cannot record 40 MHz of collision data!
- **CMS** exploits **a two-level trigger (filter)**:
  1. **Level-1 Trigger** (L1T)
     - Implemented in **hardware** on **FPGAs***
     - Receives **coarse detector data**
     - **Decision within microseconds**
  2. **High-Level Trigger** (HLT)
     - Uses **CPU/GPUs in a computing farm**
     - Full resolution of detector data
     - **Decision within seconds**

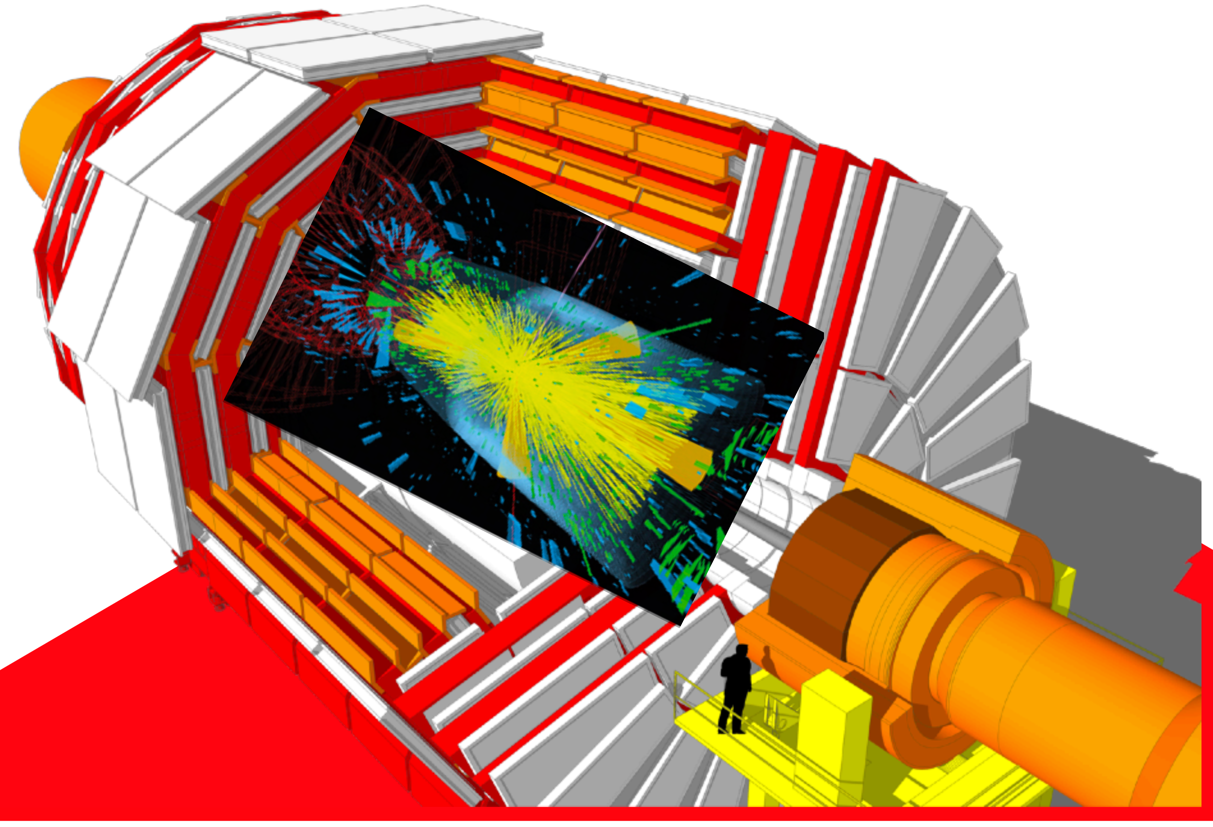




L1 vs HLT resolution

* details to come

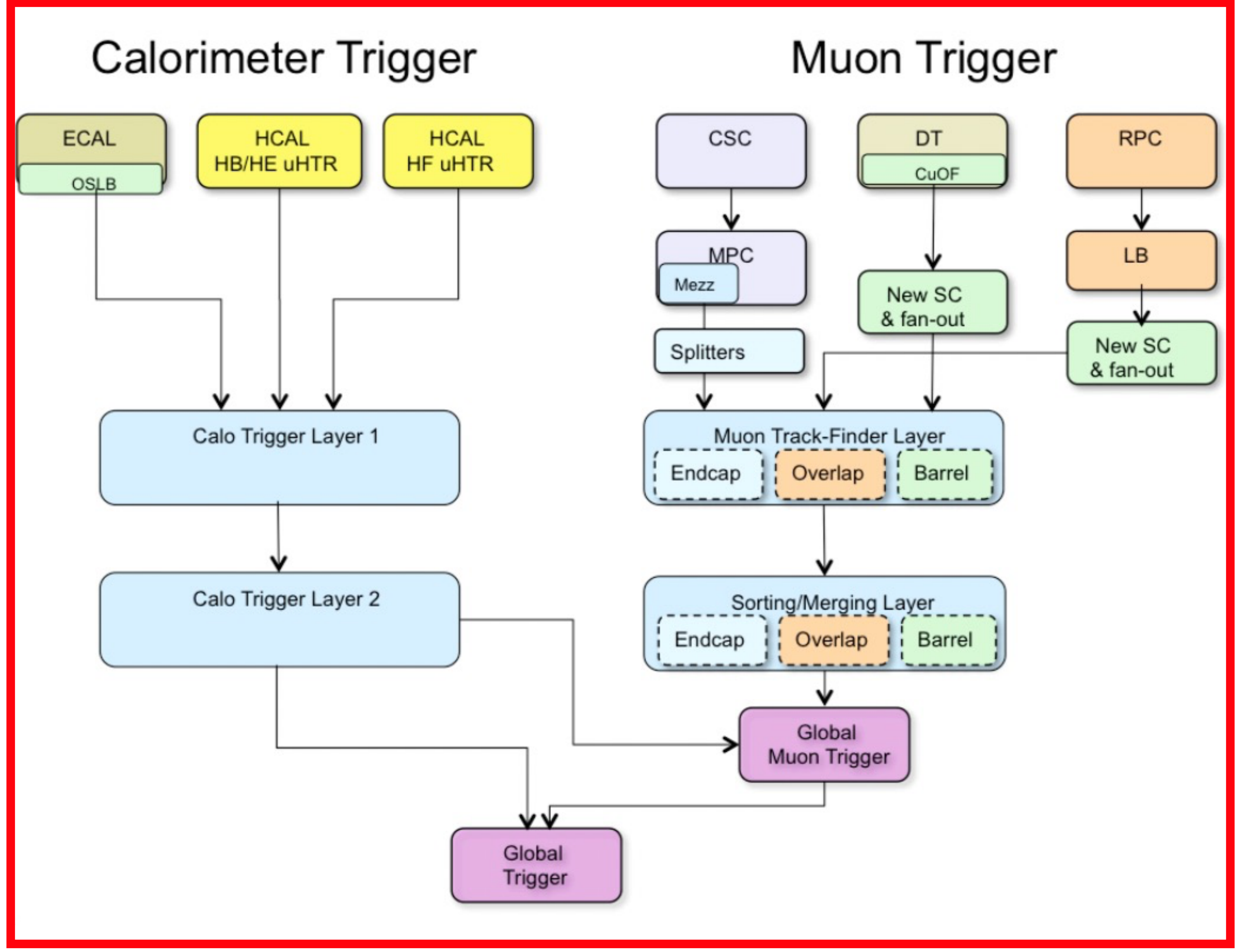
CMS Level-1 Trigger



Raw detector data "in"

Processing data and reconstructing physics objects

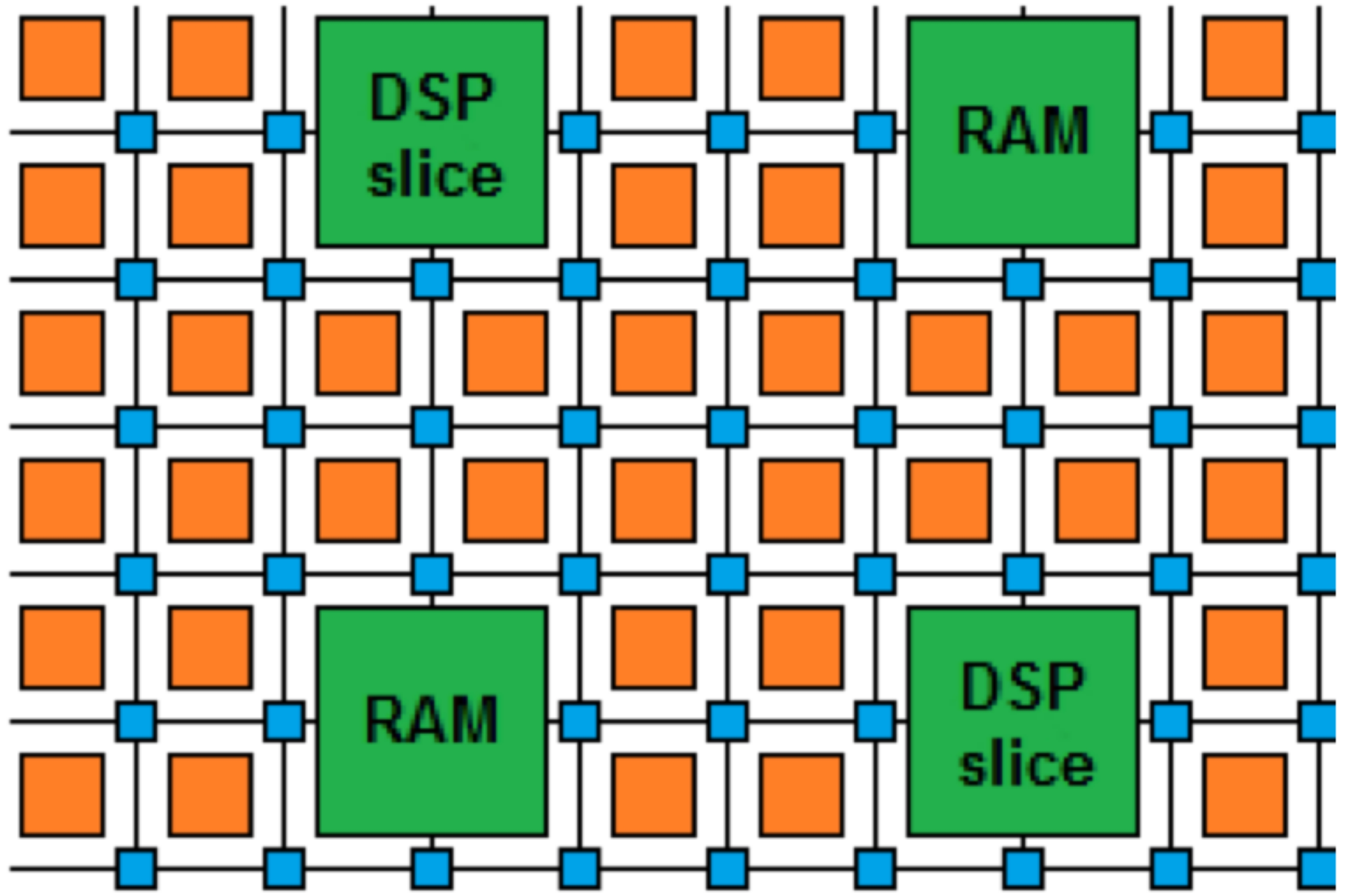
Taking decision

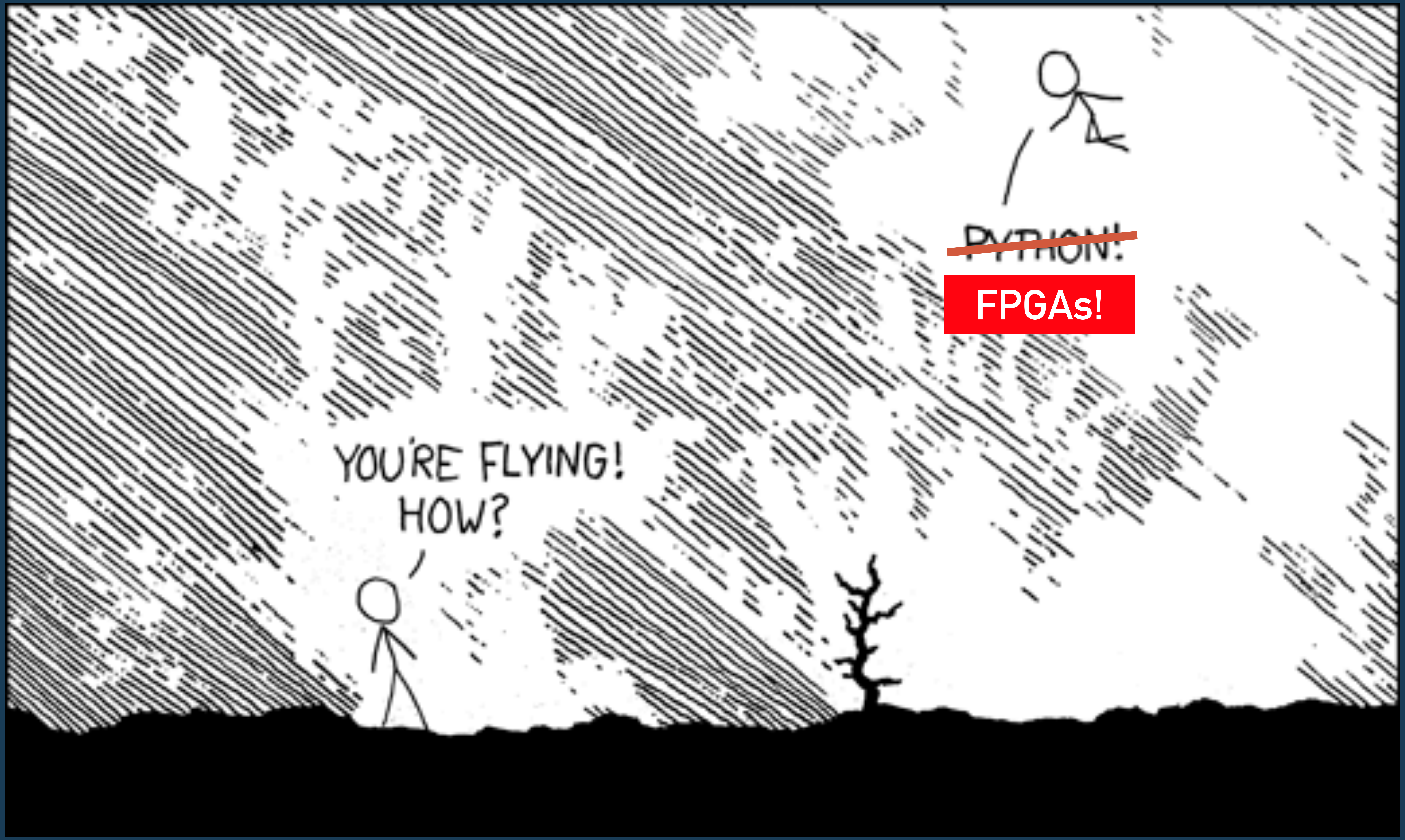


FPGA: Field Programmable Gate Arrays

The CMS L1 Trigger is based on 100s of FPGAs:

- **Integrated circuit** with **programmable logic**
  - Originally **introduced for prototyping** Application-specific Integrated Circuits (ASICs)
  - Contrary to ASIC: **(re)programmable in the "field"**
- **FPGAs consists of different parts of logic cells** for high throughput and I/O operations





[xkcd “Python”]

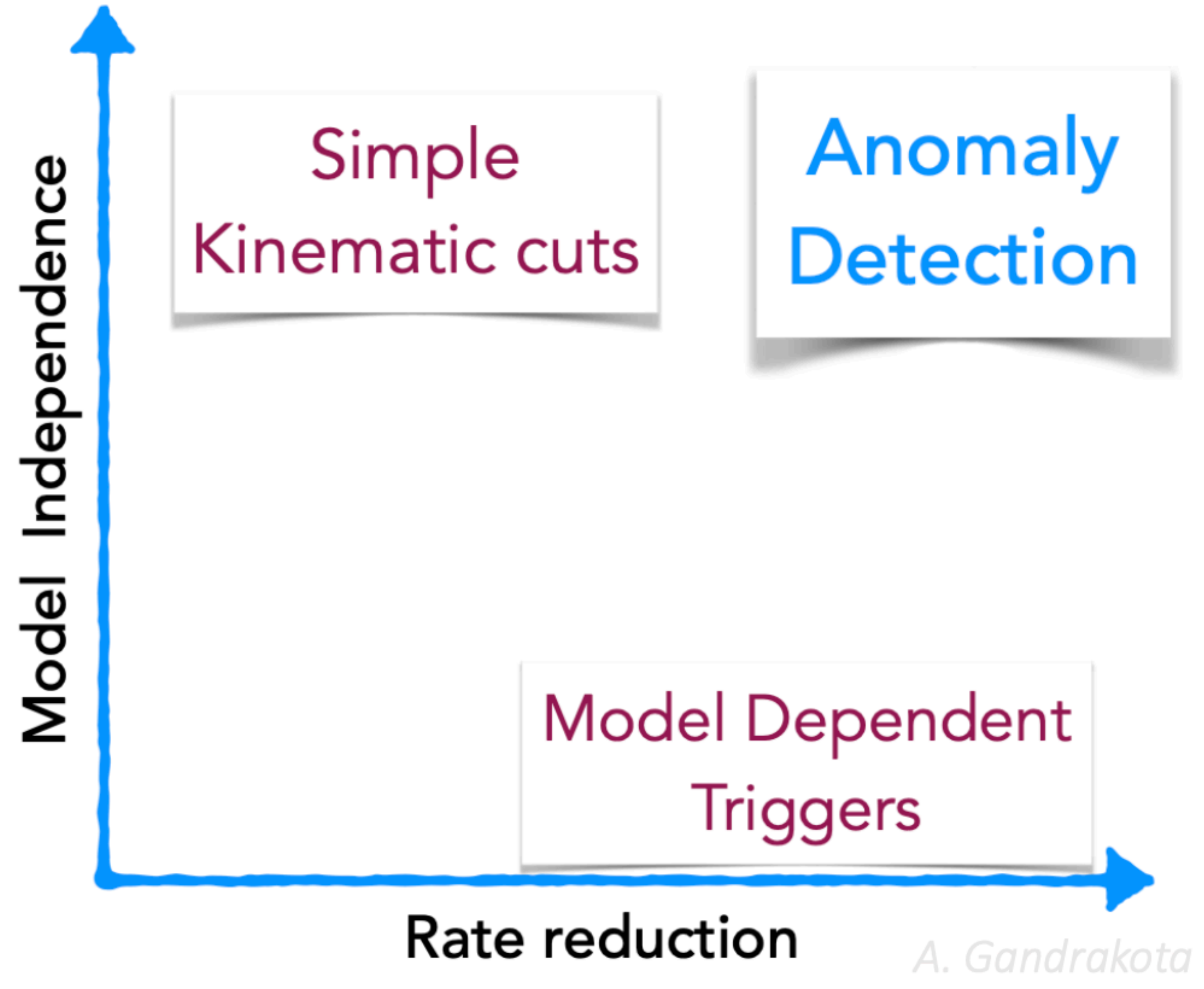
Anomaly detection @ CMS L1 Trigger

Anomaly detection in CMS

- Searching for new physics at the LHC – multiple fronts:
  - **Direct**: e.g. looking for exotic particles (peak or excess searches)
  - **Indirect**: precision measurements of particle parameters (e.g. H couplings)
  - **Anomaly detection** *using recorded data* (*examples at this conference*)

- All rely on existing selection (trigger) algorithms
  –> Model dependent or high energy thresholds

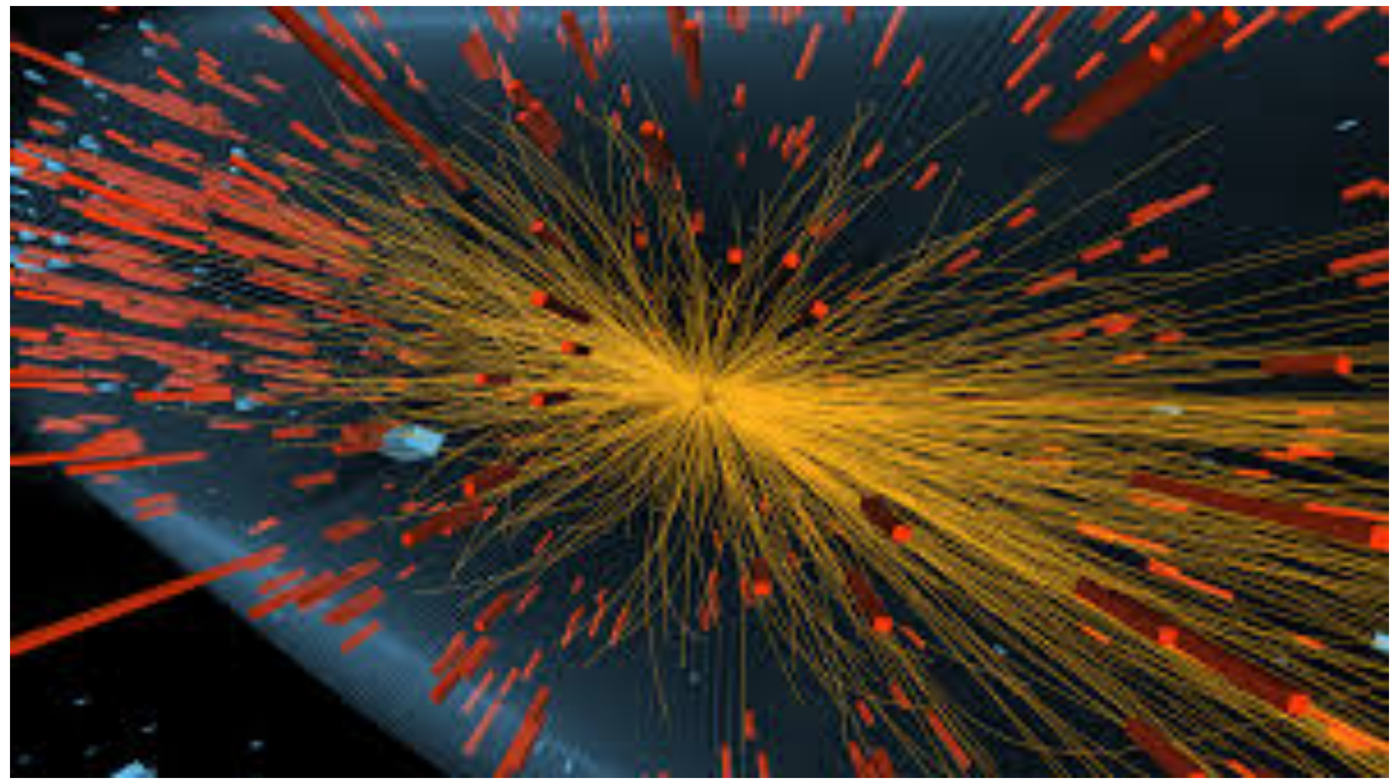
- **What if anomalous collisions are NOT RECORDED?**
  –> Anomaly detection at trigger level!

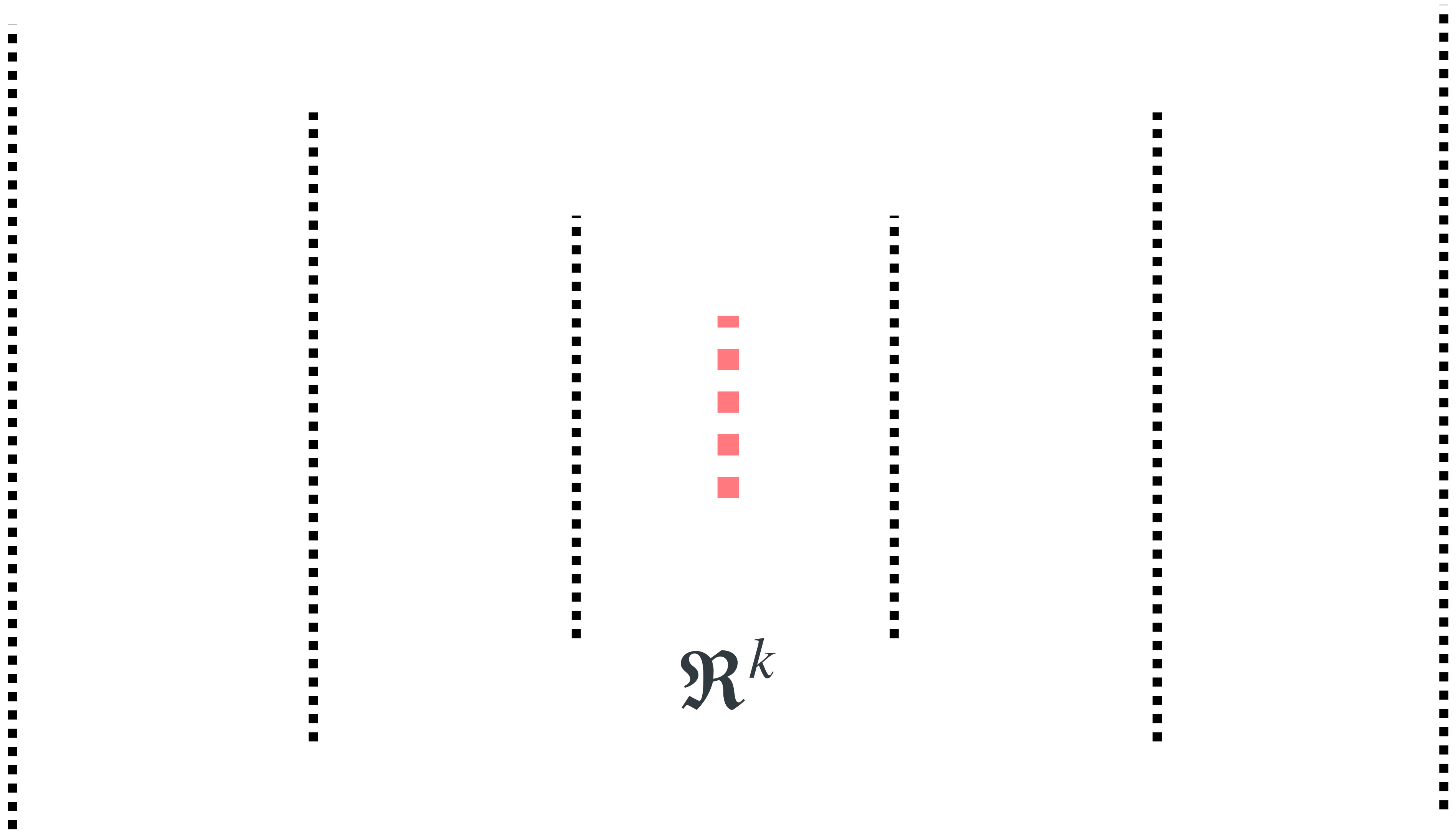


Anomaly detection with auto-encoders

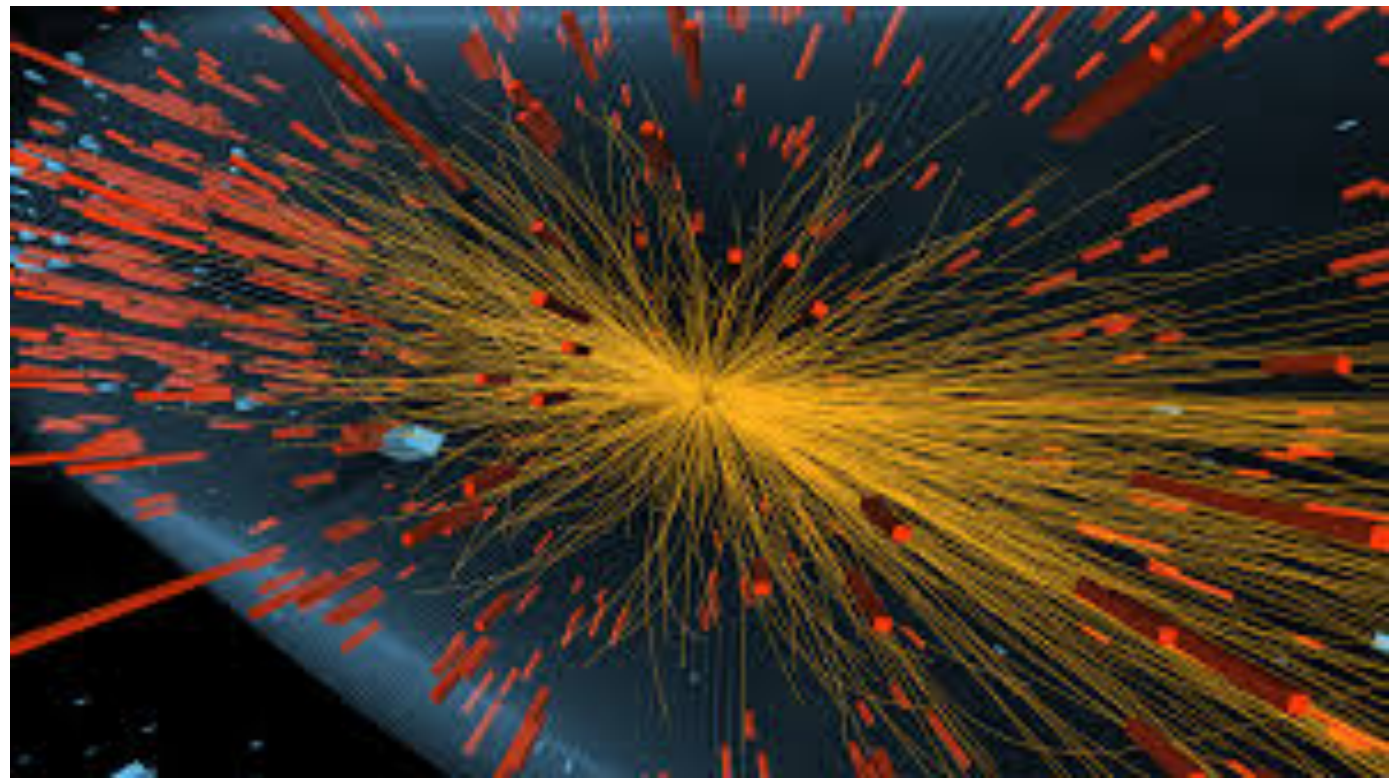
- **Autoencoders train unsupervised on data**
  - Learn to compress and to reconstruct the data
  - Difference $\hat{x} - x$ = "degree of abnormality”

Real data $\mathbf{x}$





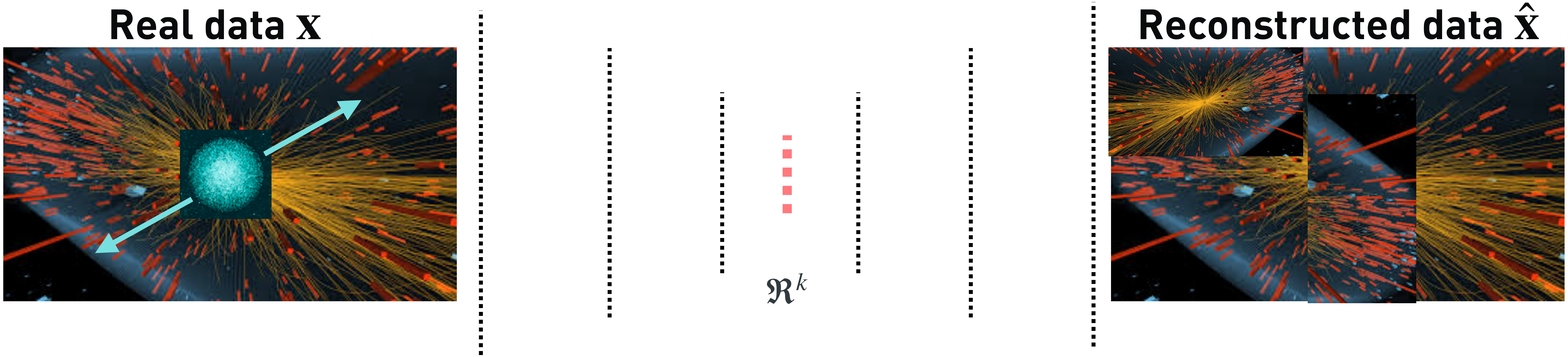
Reconstructed data $\hat{\mathbf{x}}$



ANOMALY DETECTION WITH AUTO-ENCODERS

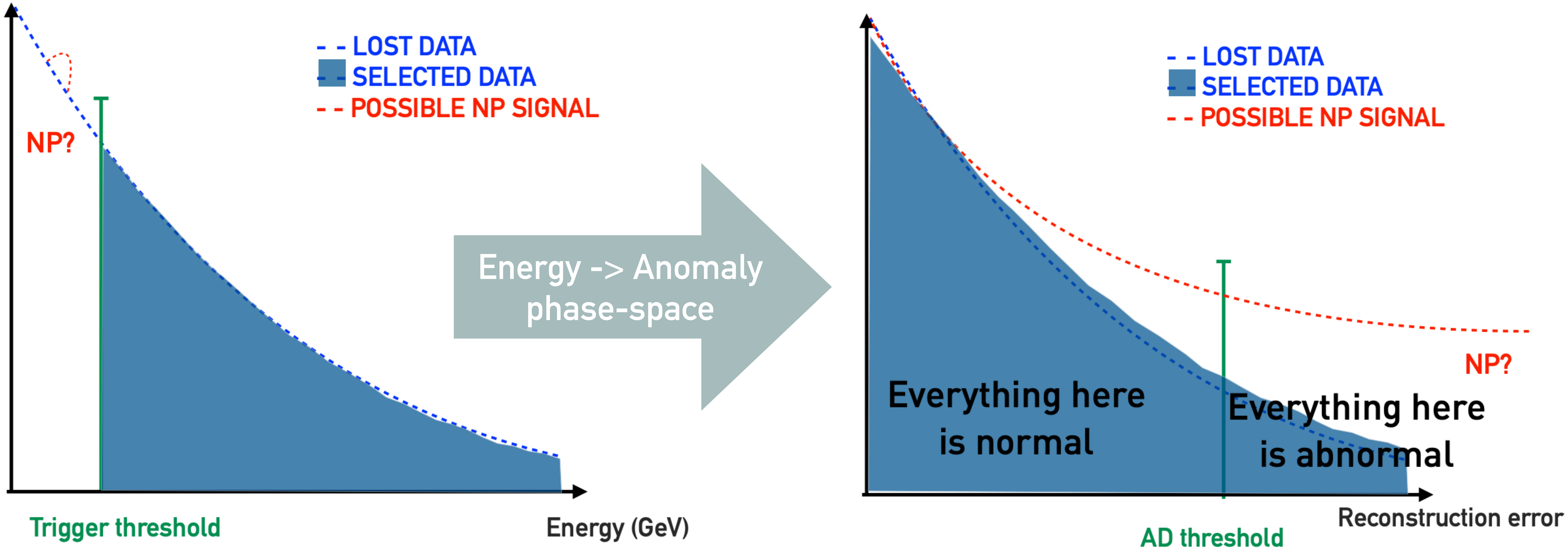
- **Autoencoders train unsupervised on data**
  - Learn to compress and to reconstruct the data
  - Difference $\hat{x} - x$ = "degree of abnormality"

➤ **If trained on "background" –> "signal" is anomalous!**



Anomaly detection for Triggering

- Traditional triggers: select dedicated (high-energy) phase space
- Anomaly detection (AD) trigger: trained on random LHC collisions (*ZeroBias*)
  - **New physics (NP) potentially results in a high reconstruction error**

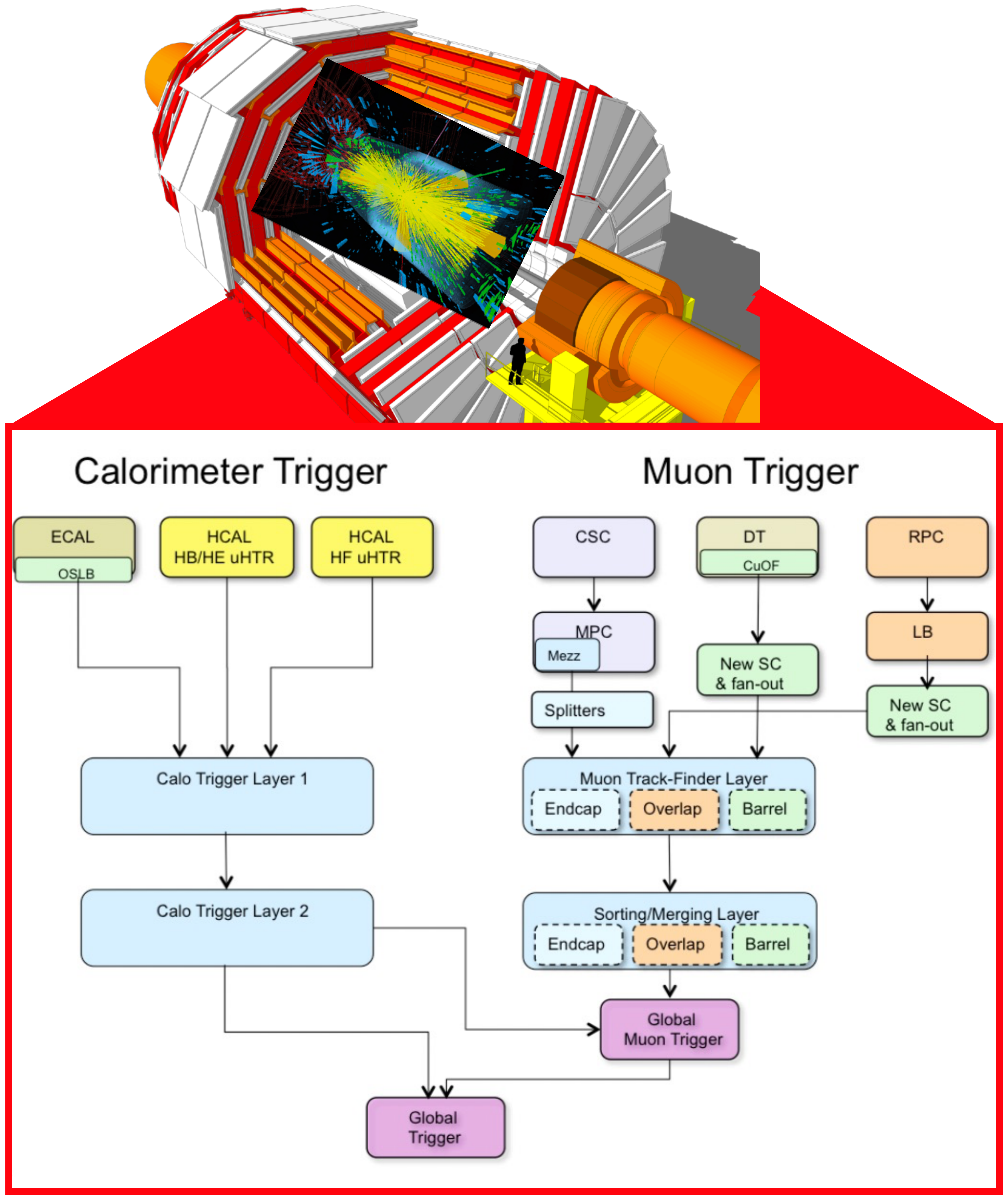


Anomaly detection @ CMS Level-1 Trigger

Raw detector data "in"

Raw detector images: CICADA

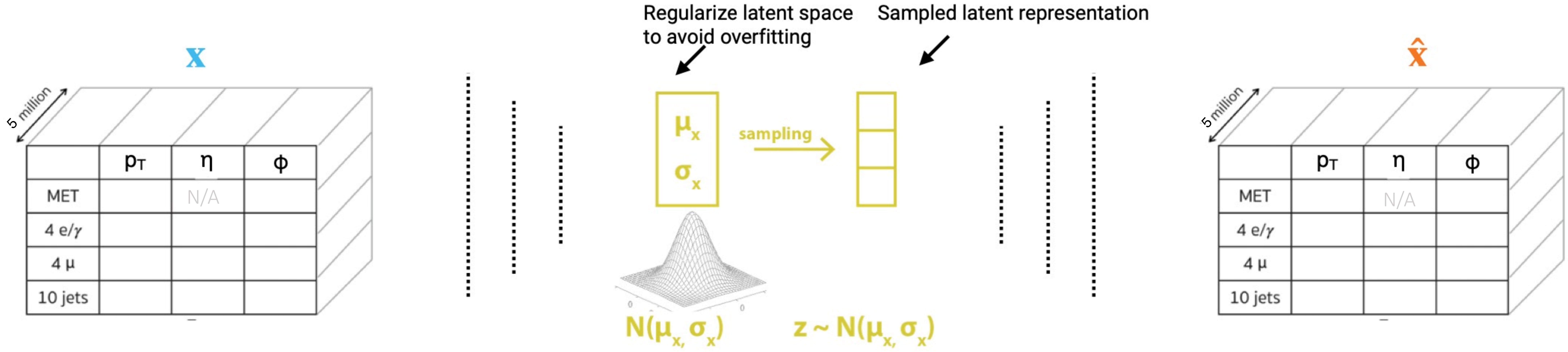
Reconstructed objects: AXOL1TL



High-level inputs: AXOL1TL

AXOL1TL: ANOMALY DETECTION WITH OBJECT TOPOLOGY

- AXOL1TL (**A**nomaly e**X**traction **O**nline **L**evel-**1** **T**rigger a**L**gorithm) **is a variational auto-encoder:**
  - Encodes input as a distribution over the latent space
  - Add regularisation term in loss: KL divergence, how different is distribution from Gaussian
- **Inputs: L1 trigger objects 4-vectors** (pT, η, ϕ)
  - Most energetic 4 electron/photons, 4 muons, 10 jets and missing transverse energy (MET)



$$\text{loss} = ||x - \hat{x}||^2 + \mathrm{KL}[\,N(\mu_x, \sigma_x), N(0, I)\,]$$

CMS-DP-2023-079

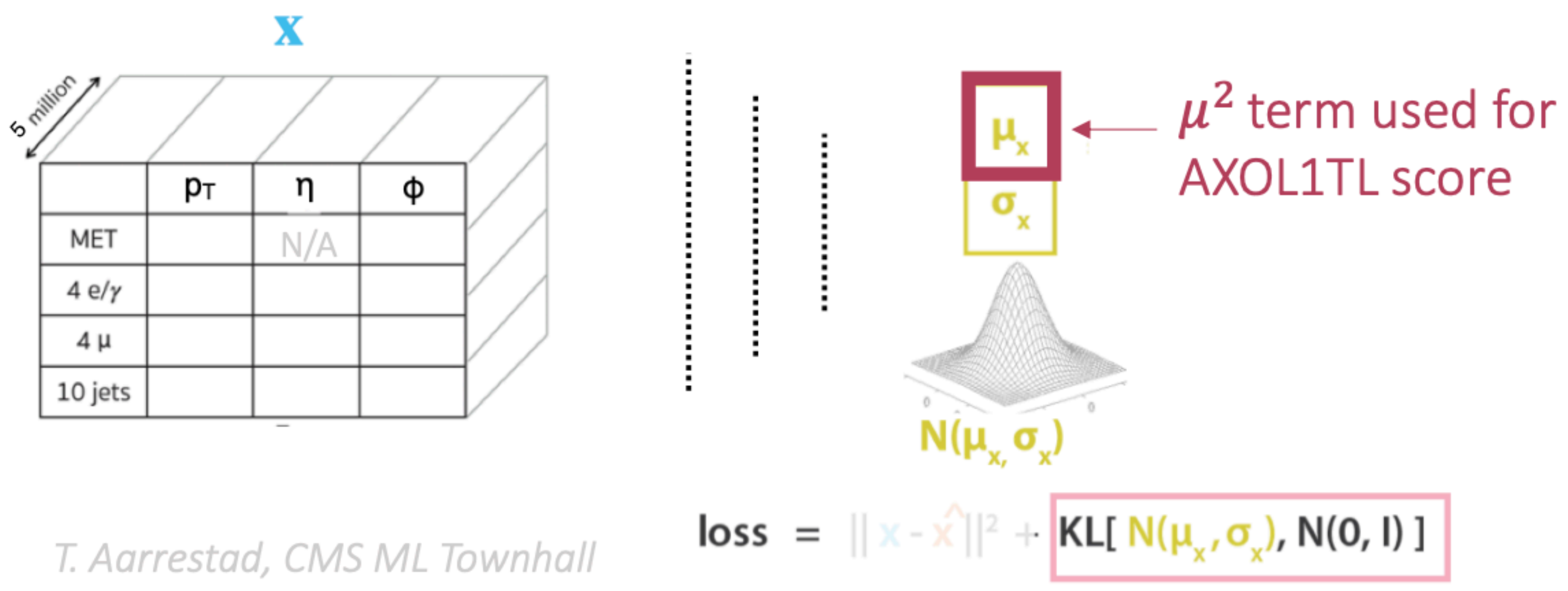
T. Aarrestad

AXOL1TL: ARCHITECTURE OPTIMISATION

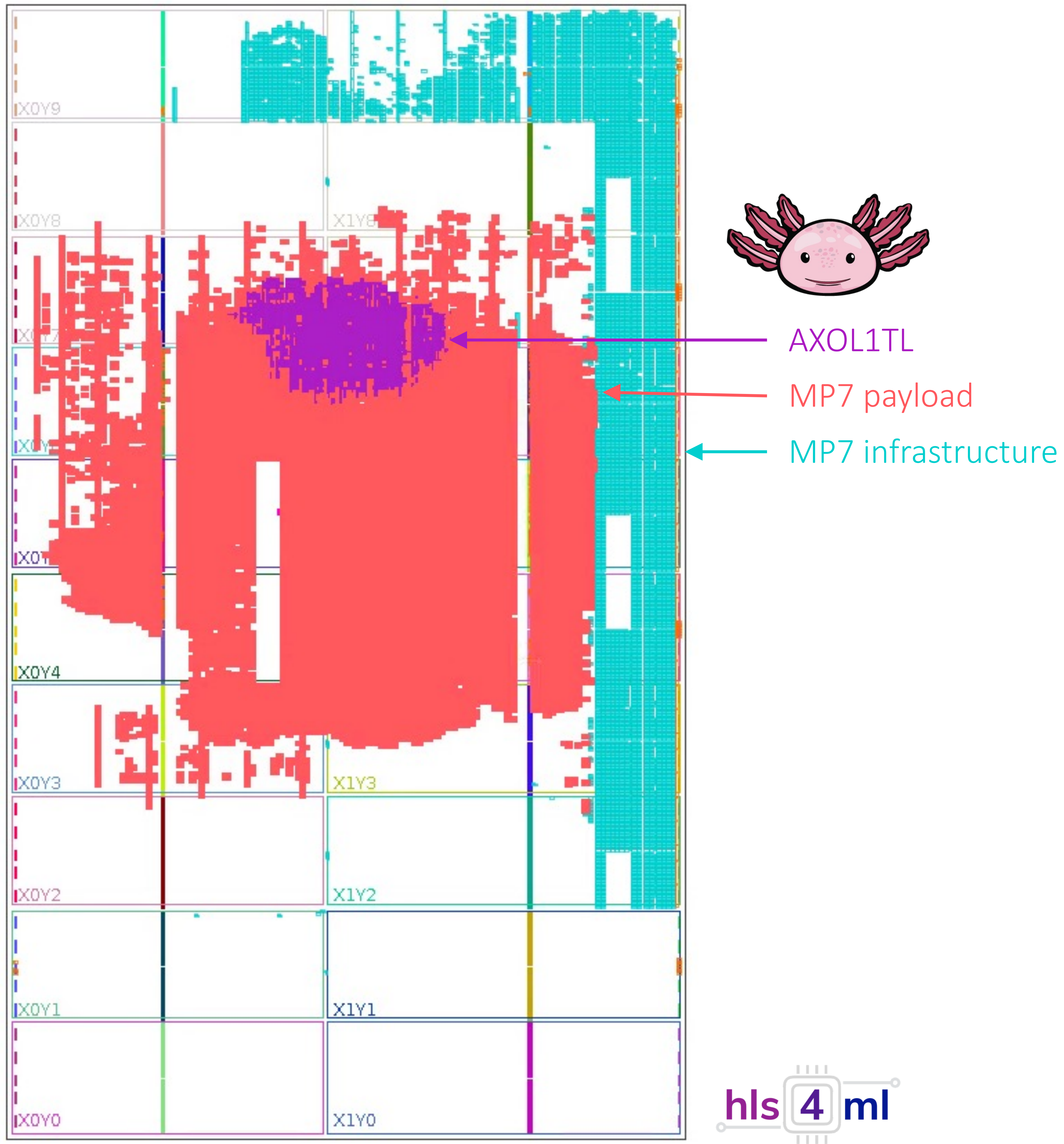
- Full NN architecture does not fit the L1/FPGA constraints

CMS-DP-2023-079

  - **–> only use encoder half of the network**
    - Compute degree of abnormality from latent space directly
    - No need to use inputs for anomaly score computation
    - **Half network size and latency!**



AXOL1TL: FPGA Implementation



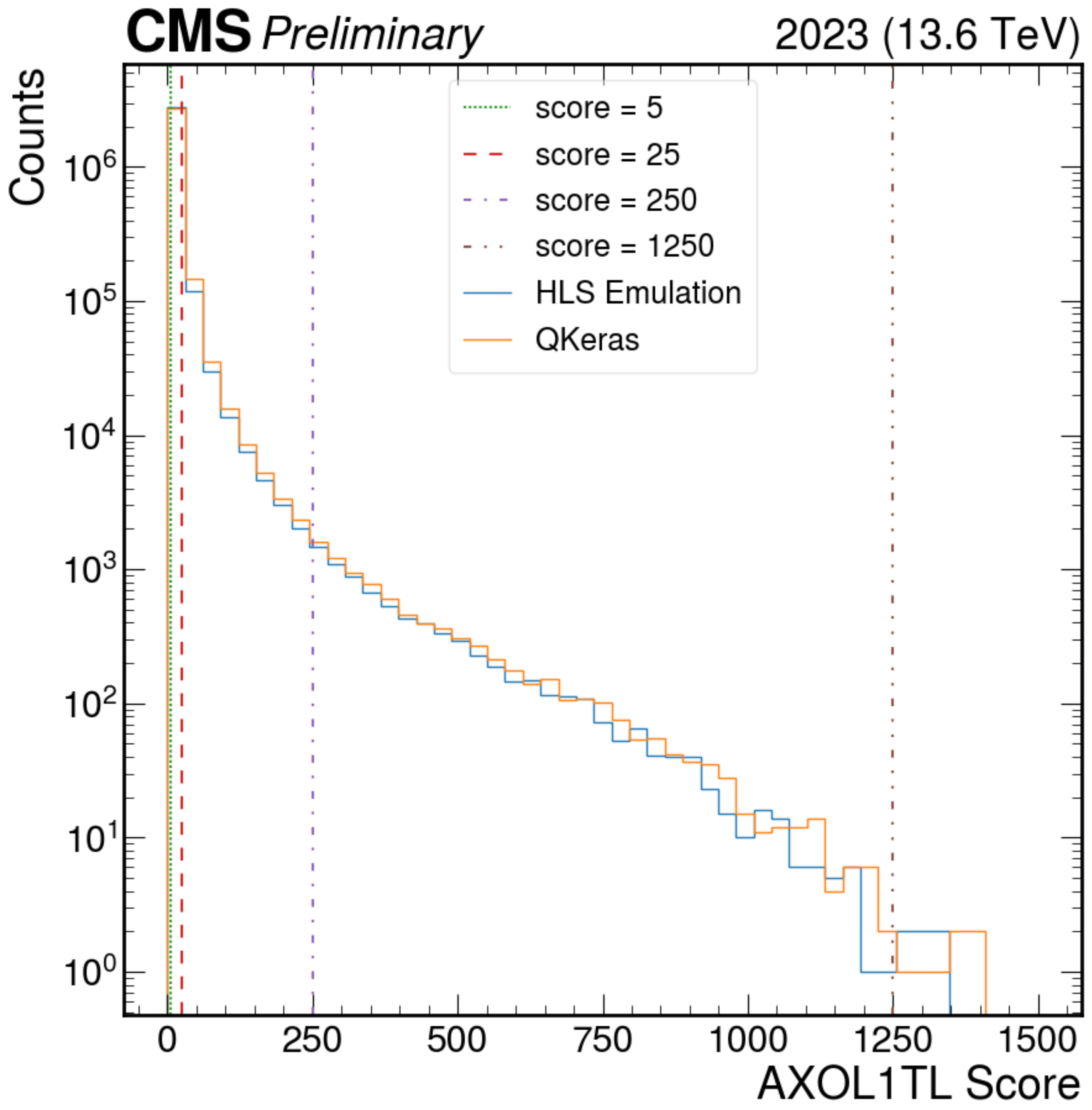
- Implemented on Xilinx Virtex-7 XCVU9P FPGA
- **Met requirements on latency and resources**

Resource utilization of Virtex-7 FPGA chip on Imperial College MP7 µGT board

	Latency	LUTs	FFs	DSPs	BRAMs
AXOL1TL	2 ticks 50 ns	2.1%	~0	0	0

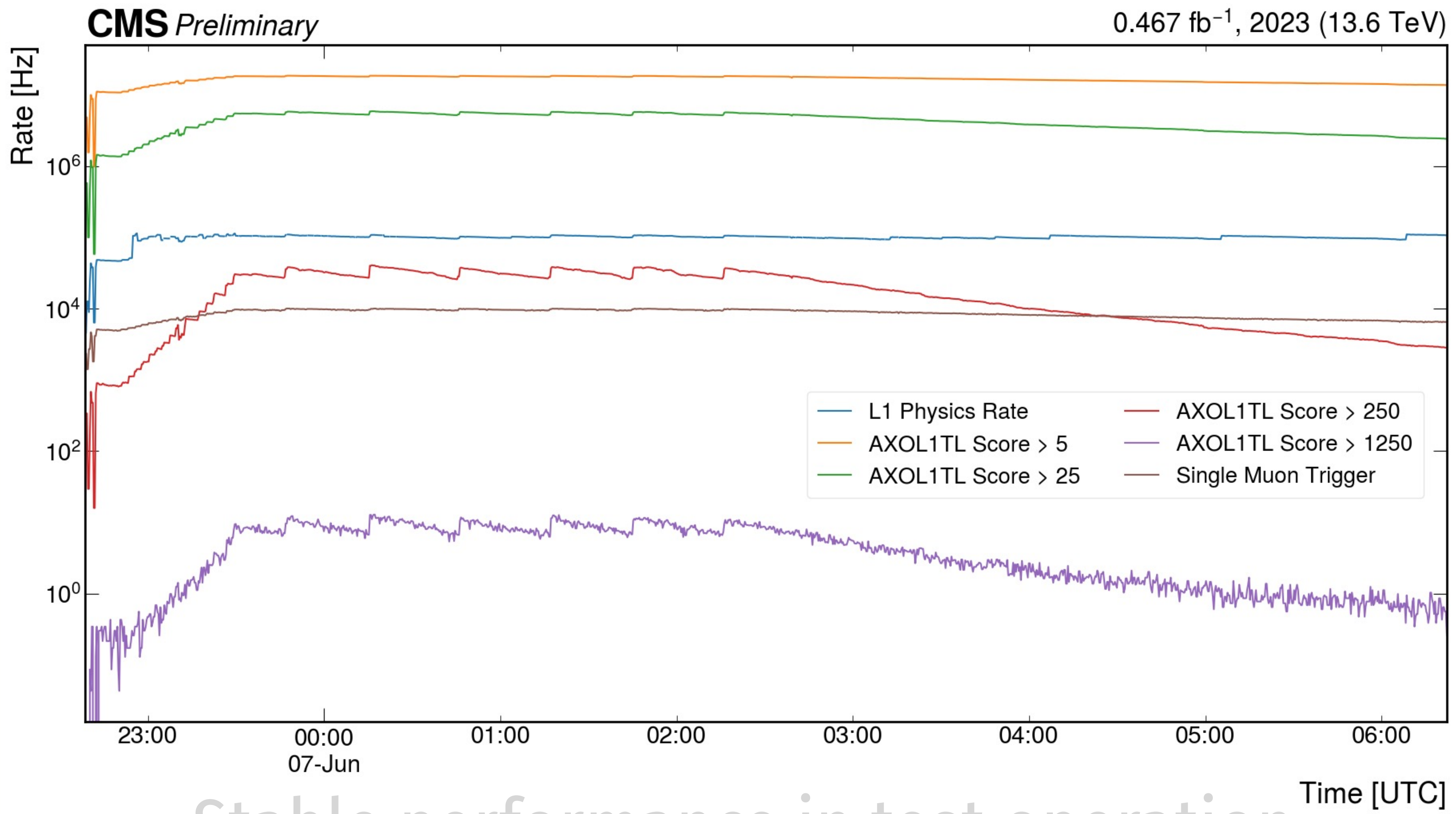
CMS-DP-2023-079

AXOL1TL: Commissioning



Anomaly score distribution for unbiased (random) LHC collision data

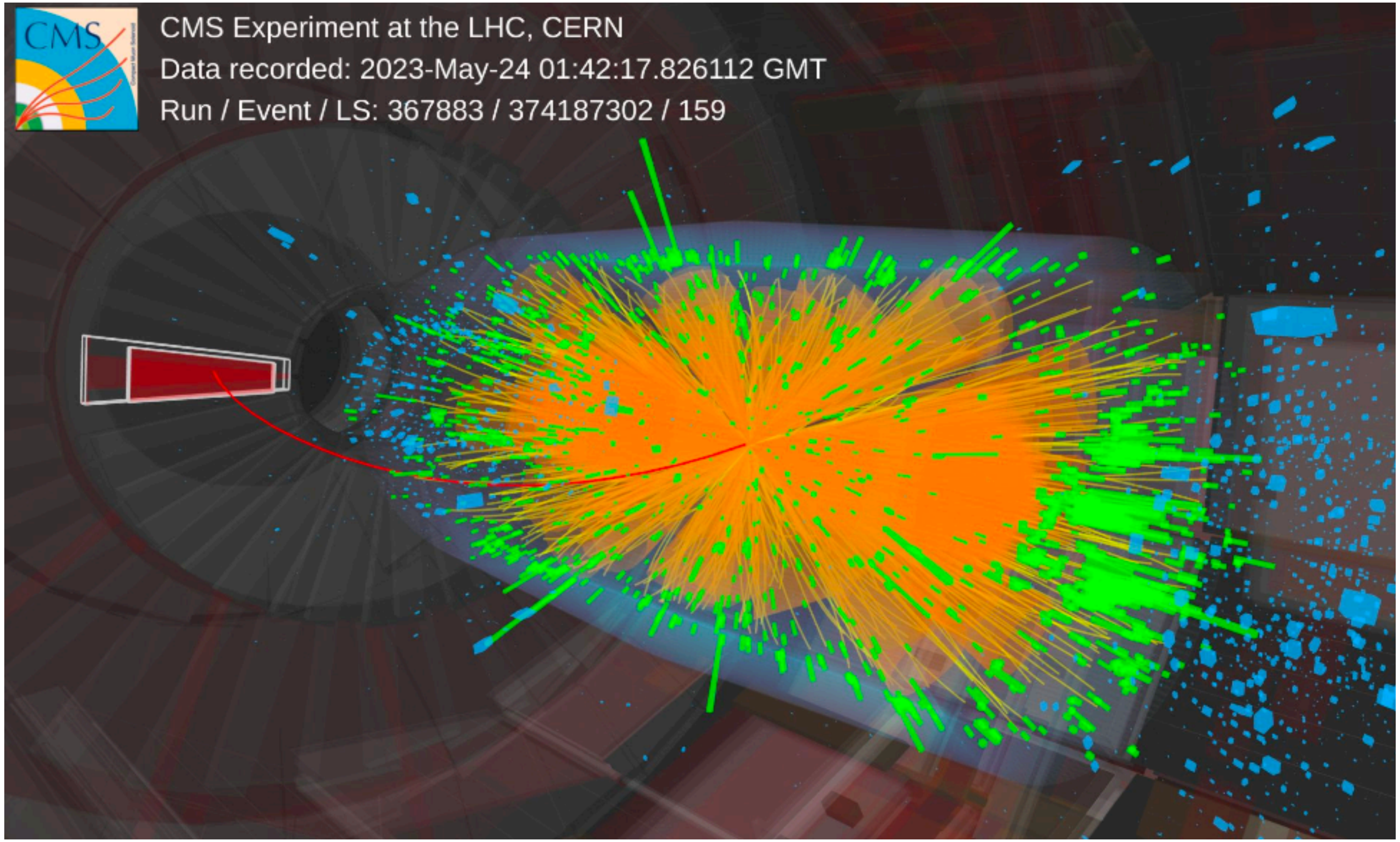
- AXOL1TL is trained with unbiased data collected by CMS during 2023 with √s=13.6 TeV
  - 10.5 million events (50/50% for training/testing)
  - Selected 5 test scores in firmware
- **Commissioned in Global Trigger Test Crate during proton collisions** in 2023 –> stable as standard triggers



Stable performance in test operation

CMS-DP-2023-079

AXOL1TL: EVENT DISPLAY



- Example of an anomalous event during 2023 pp collisions (from random trigger dataset)
  - **Highest anomaly score event not triggered by L1**
- L1 objects:
  - 11 jets with pT > 20 Gev
- Offline objects:
  - 7 jets with pT > 15 GeV from the same vertex
  - 75 identified vertices

CMS-DP-2023-079

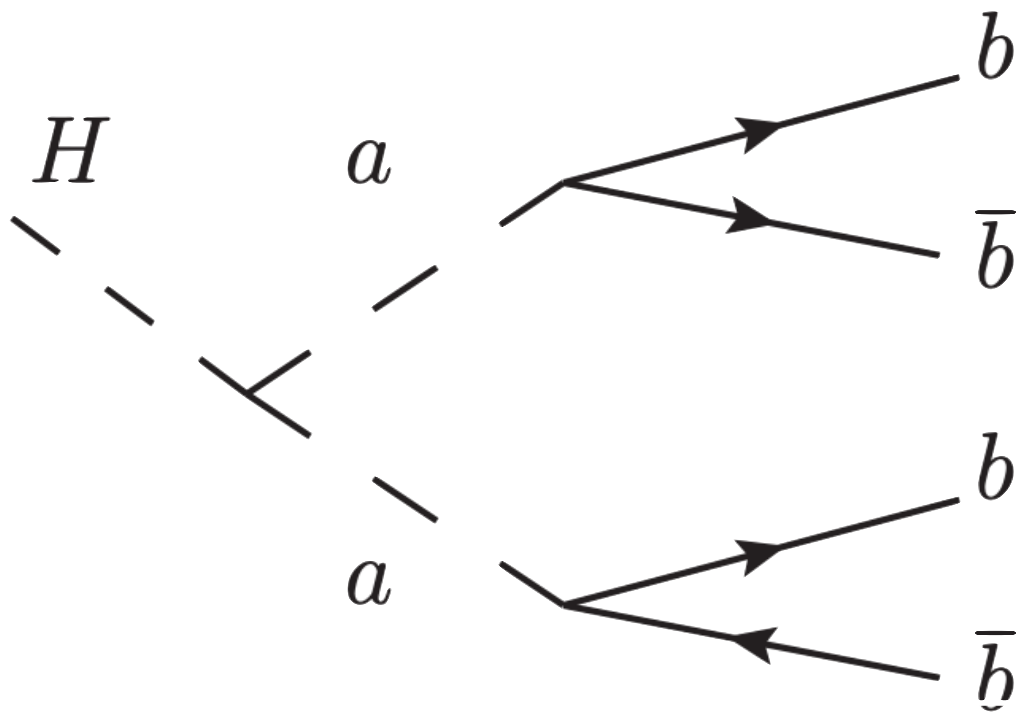
AXOL1TL: Physics performance

- Use **simulated hypothetical exotic signal as a anomaly candidate**
- Significant **performance improvement on various SM and beyond the SM signals** by adding AXOL1TL to the 2023 trigger menu

$$\text{Improvement} = \frac{\text{L1 Efficiency w/ AXOL1TL@freq}}{\text{L1 Efficiency w/o AXOL1TL}} - 1$$

- Example performance improvement for H->aa[15 GeV]->4b signal:

AXOL1TL Rate	1 kHz	5 kHz	10 kHz
Signal Efficiency Gain	46%	100%	133%

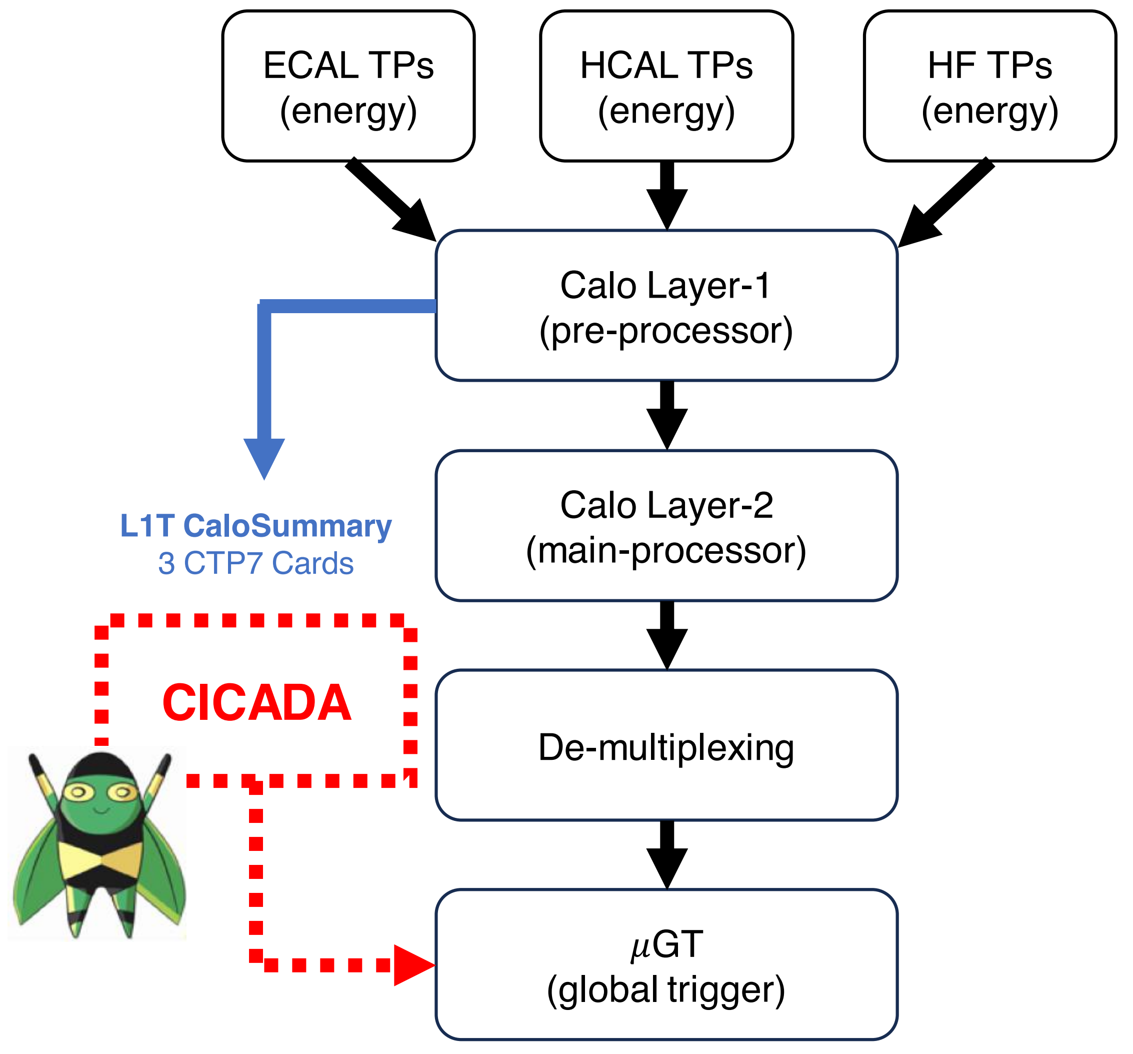


- **Planning to start data-taking with ~*O*(100) Hz L1 rate in 2024 pp collisions!**

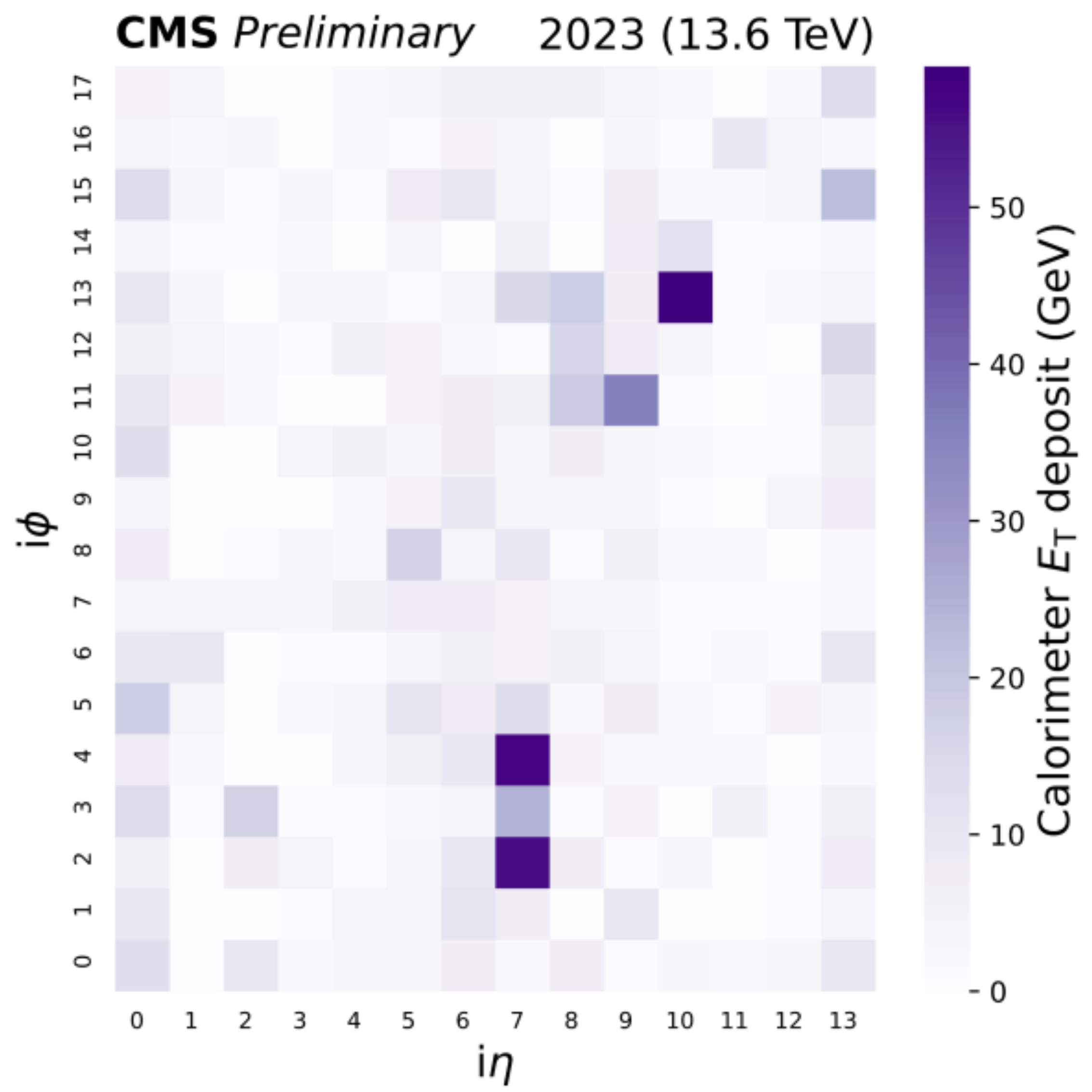
CMS-DP-2023-079

RAW FEATURES: CICADA

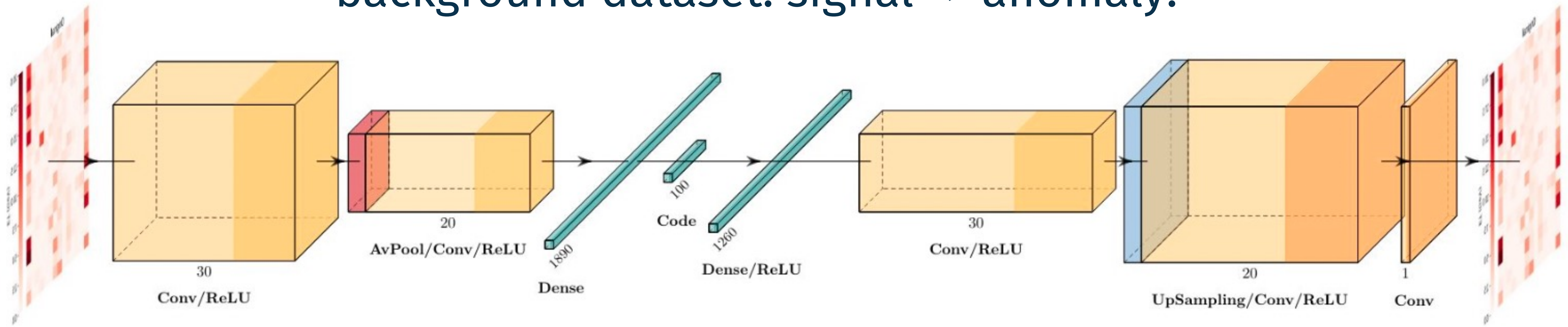
CICADA: Anomaly Trigger on Raw inputs



- CICADA (CMS DP-2023/086):
  Calorimeter **I**mage **C**onvolutional
  Anomaly **D**etection **A**lgorithm
- **Using raw inputs of calorimeter:**
  - Image of 18 x 14 energy deposits
  - **Independent of domain knowledge** (standard trigger algorithms)



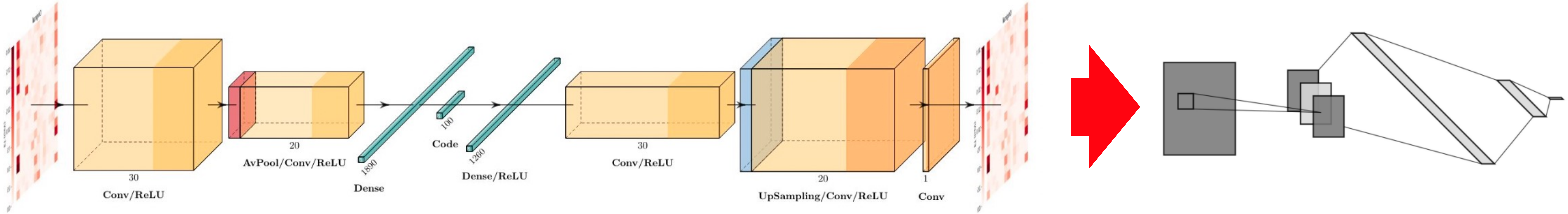
- Convolutional auto-encoder trained on background dataset: signal -> anomaly!



CMS DP-2023/086

CICADA: Knowledge Distillation

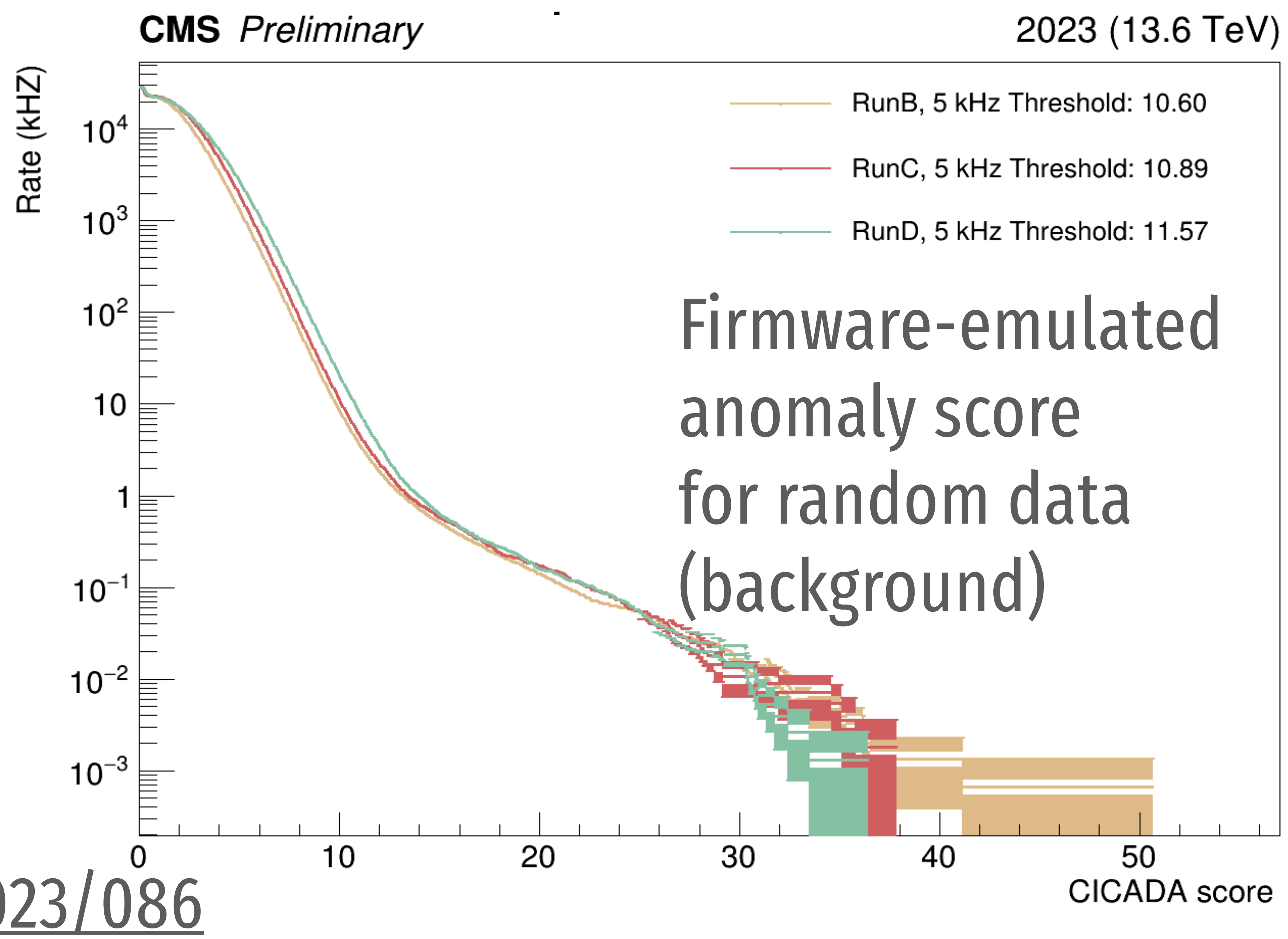
- Full CICADA model is too complex for FPGA resources / L1 Trigger requirements
  –> **use Student-Teacher Knowledge Distillation**
  - **Teacher model**: complete encoding and decoding of the original input data
    - **Anomaly score (reconstruction error)**: average of the squared error (predicted – input) in reconstruction for each of the 252 individual energy deposits (Mean Squared Error)
  - **Student model: regresses the anomaly score of the teacher model**
    - Smaller convolutional layer with only 4 filters + few dense layers
      -> 10x faster & less resources -> fits FPGA/L1T requirements

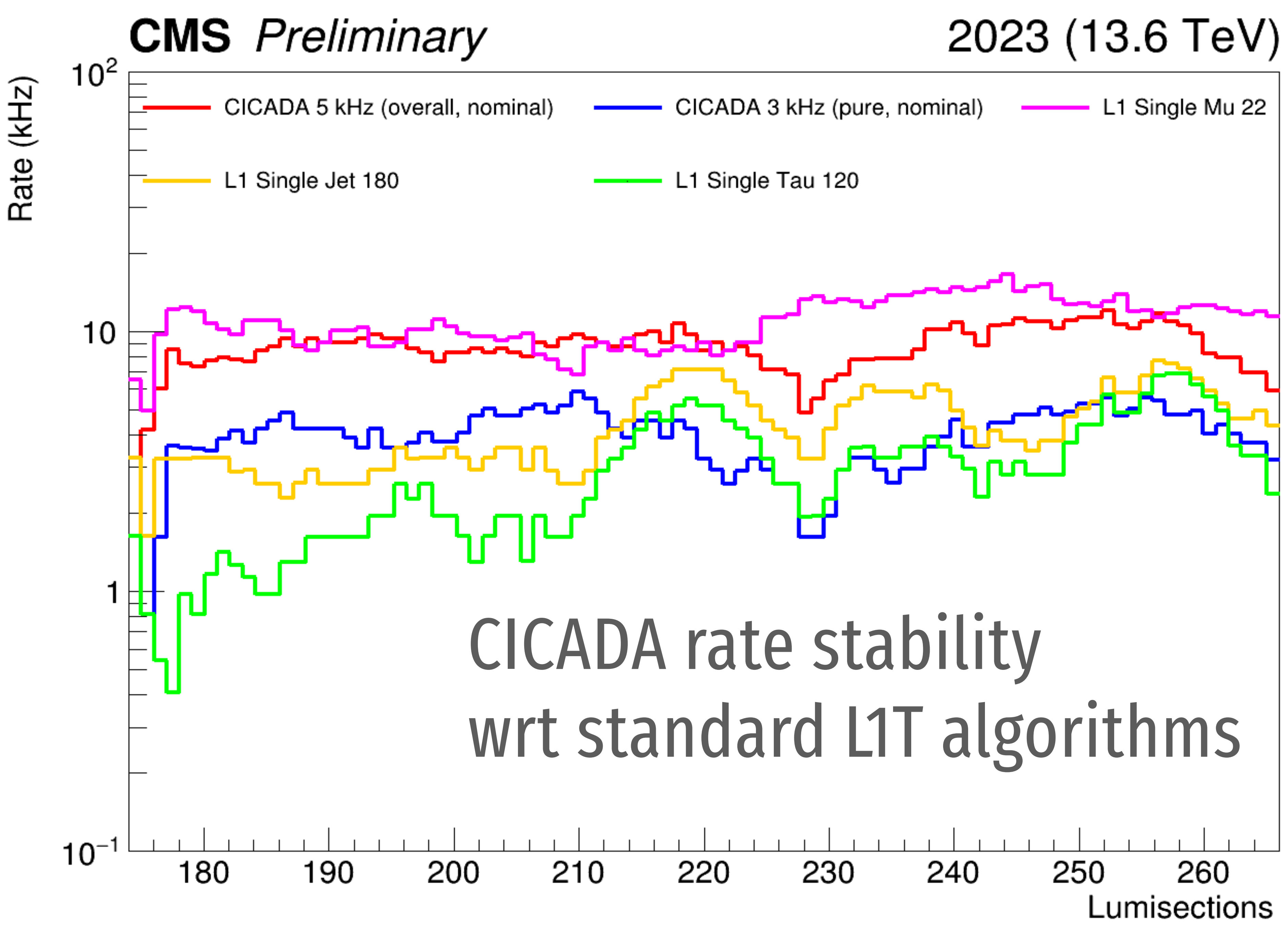


CMS DP-2023/086

CICADA: Commissioning

- **CICADA currently being commissioned in the L1 Trigger test system**
  - Software-based emulation based on Firmware (HLS4ML) and validated
  - Preliminary performance estimates promising + operational stability tested
- **This is the first anomaly detection on low-level inputs in a LHC trigger system!**



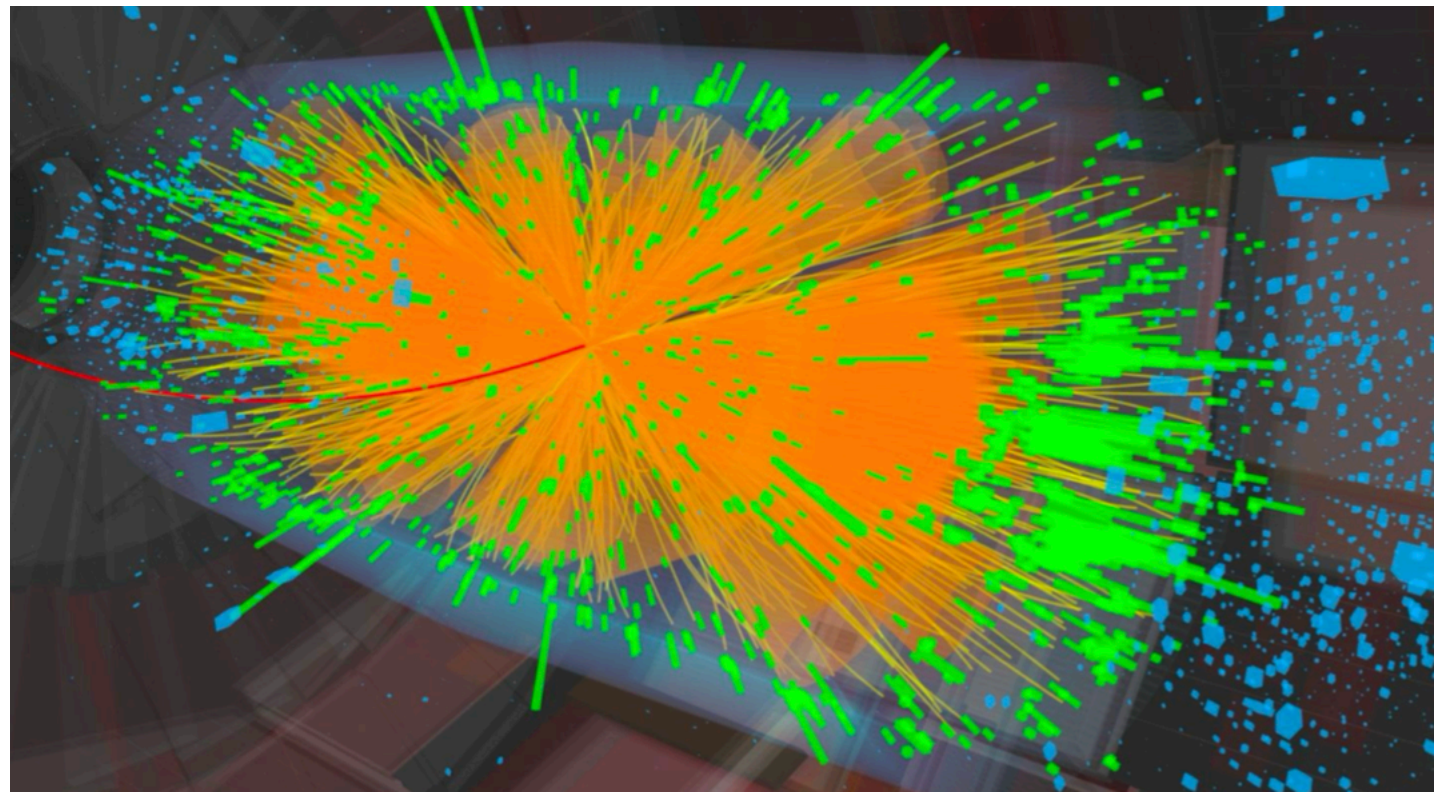


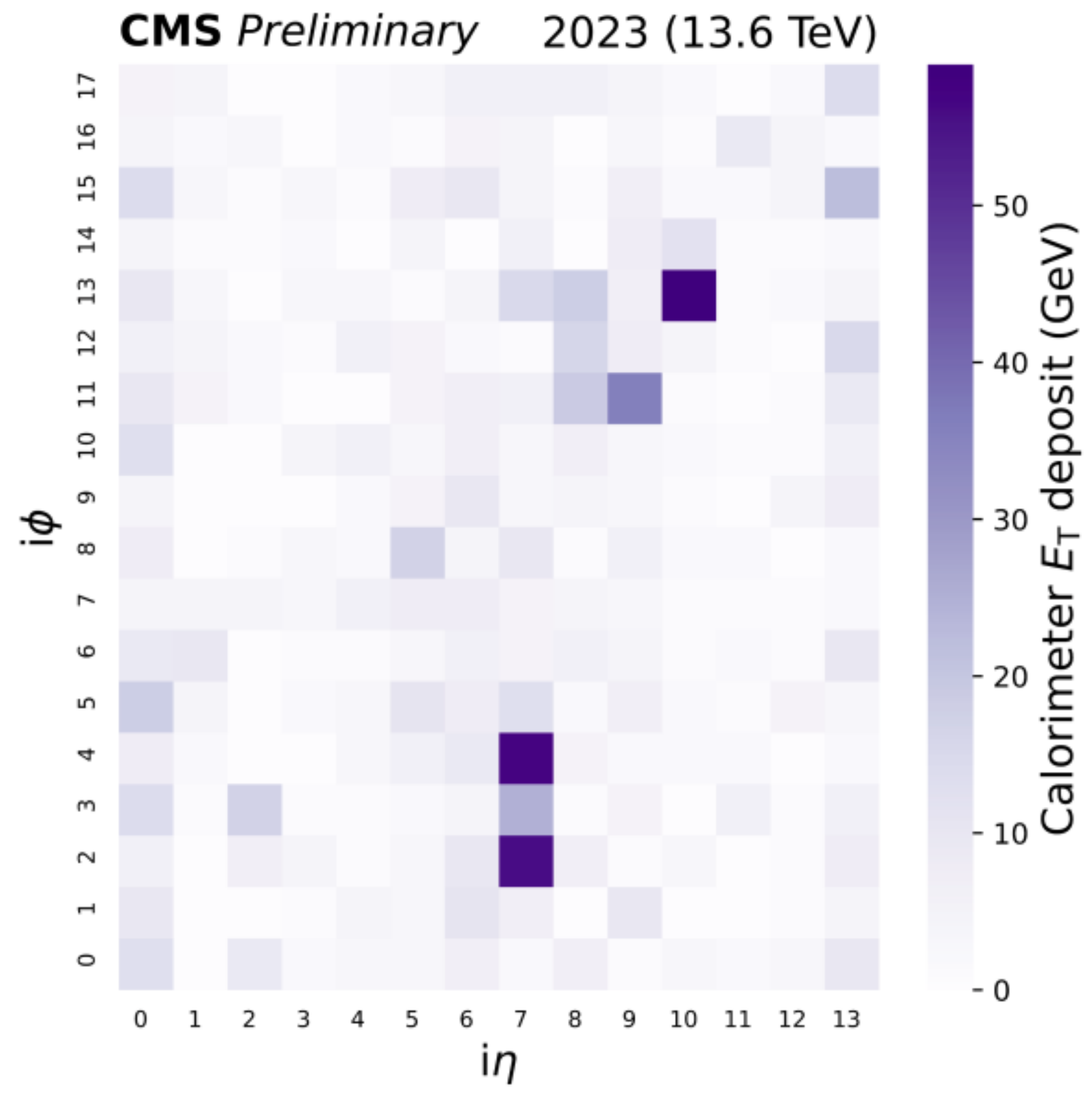
CMS DP-2023/086

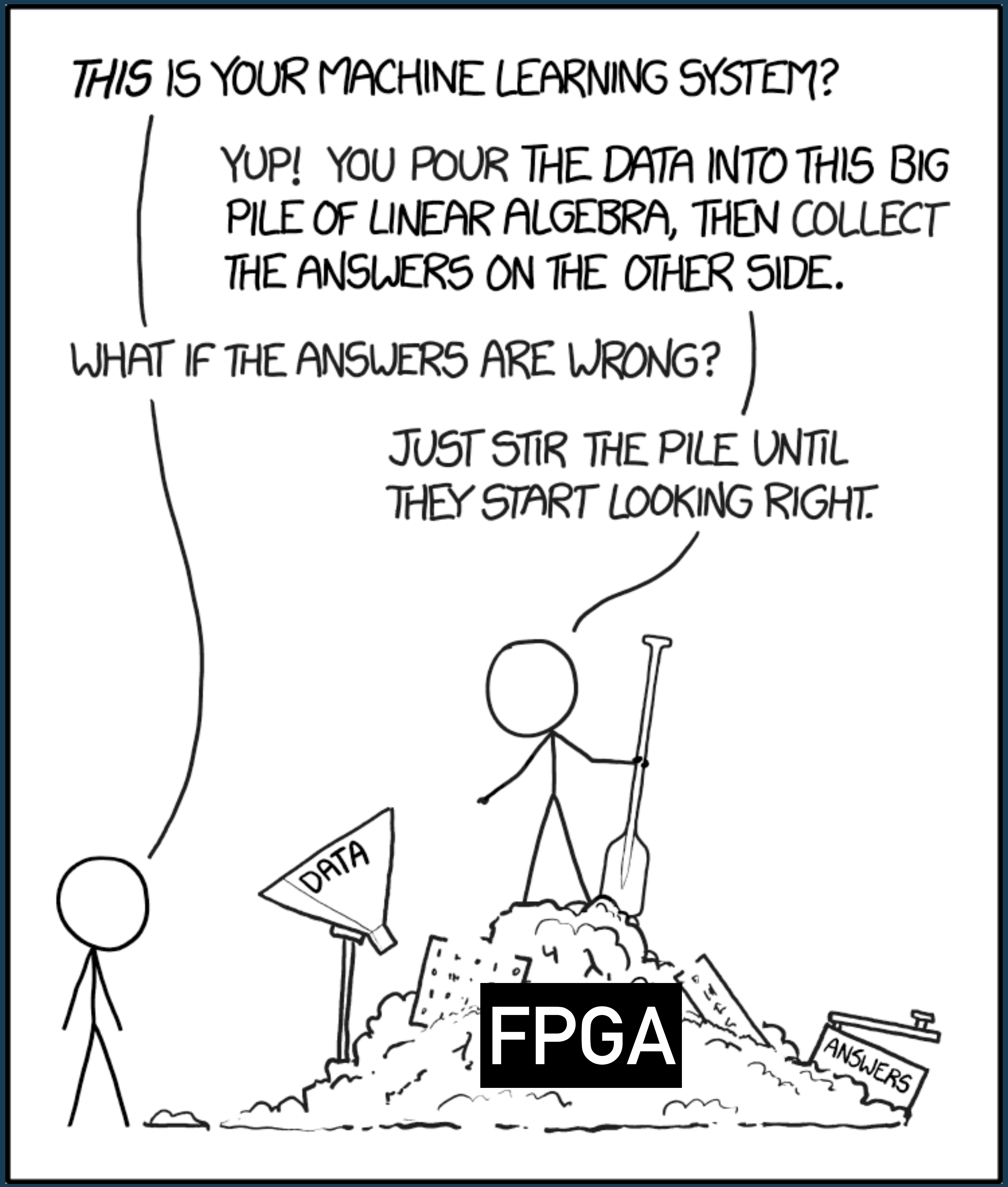
Summary

Anomaly Detection with the CMS Level-1 Trigger

- Various **anomaly searches for new physics performed at the LHC**
- Opening **a new direction:**
  anomaly detection in the CMS Level-1 Trigger
  - **Challenging environment for L1T**:
    - Hardware/FPGAs: restricted resources and latency (ns!)
    - Physics: <60> simultaneous collisions,
      only calorimeter and muon detector data
- **Two auto-encoder approaches being commissioned** in CMS:
  - **AXOL1TL**: using high-level physics objects [CMS-DP-2023-079]
  - **CICADA**: using raw detector data [CMS DP-2023/086]
- **Promising prospects for anomaly triggering in CMS!** [HL-LHC L1T]







[xkcd “Machine Learning”]()

References

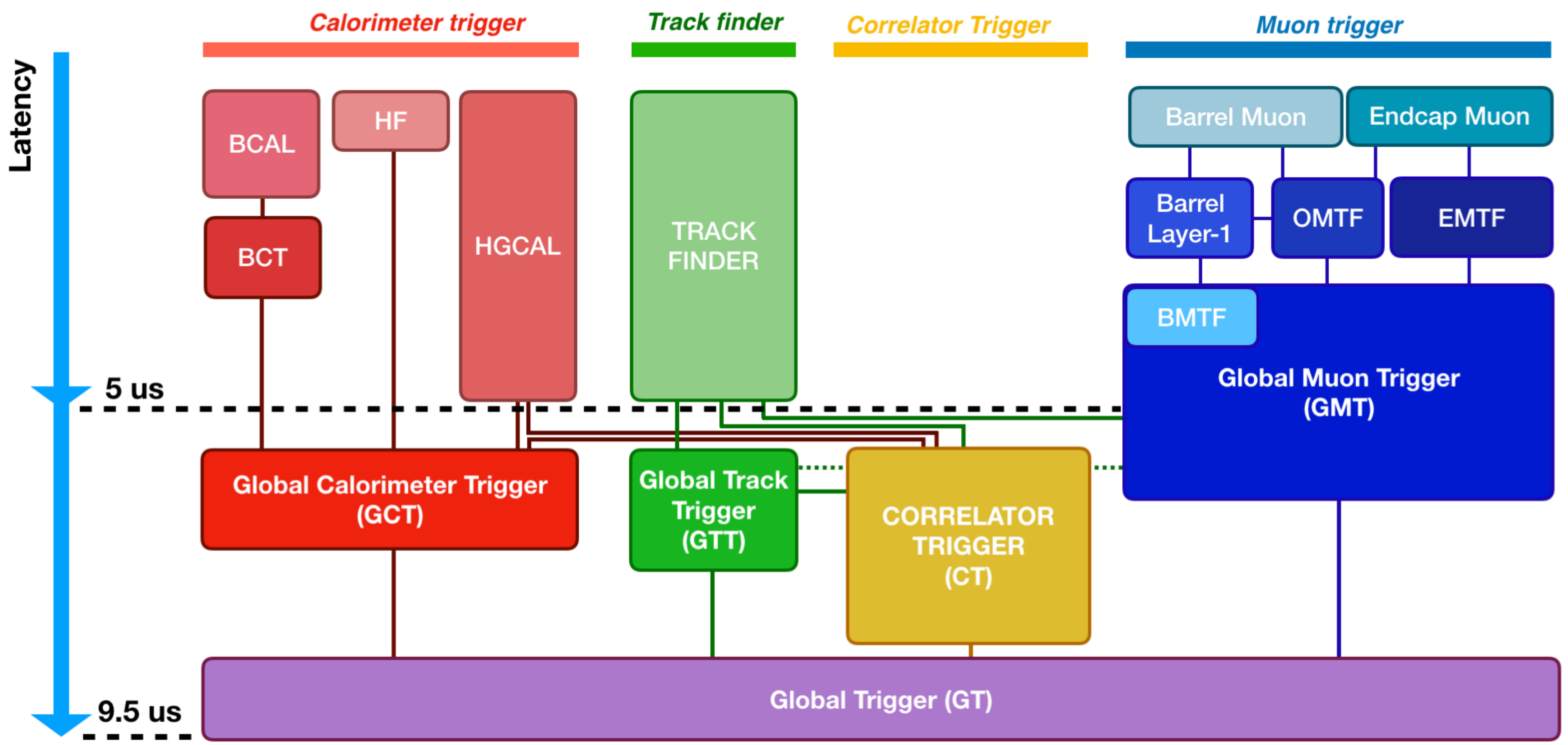
- CMS Collaboration. ”Anomaly Detection in the CMS Global Trigger Test Crate for Run 3”. CMS-DP-2023-079 , CERN-CMS-DP-2023-079 (2023): https://cds.cern.ch/record/2876546
- CMS Collaboration. "Level-1 Trigger Calorimeter Image Convolutional Anomaly Detection Algorithm”, CMS-DP-2023-086 ; CERN-CMS-DP-2023-086 https://cds.cern.ch/record/2879816
- J. Pearkes, “Realtime Anomaly Detection in the CMS Experiment Global Trigger Test Crate”, ML4Jets 2023, https://indico.cern.ch/event/1253794/timetable/?view=standard#57-realtime-anomaly-detection
- N. Zipper, “Testing a Neural Network for Anomaly Detection in the CMS Global Trigger test crate during Run 3”, TWEPP 2023 https://indico.cern.ch/event/1255624/contributions/5444028/
- C. Sun, “Realtime Anomaly Detection in the CMS Experiment Global Trigger Test Crate”, FastML 2023, https://indico.cern.ch/event/1283970/contributions/5554350/
- CMS Collaboration. “CMS Technical Design Report for the Level-1 Trigger Upgrade”, CERN-LHCC-2013-011 ; CMS-TDR-12 https://cds.cern.ch/record/1556311
- E. Govorkova, et al. "Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider". Nat. Mach Intell. 4, 154 (2022). https://doi.org/10.1038/s42256-022-00441-3
- FastML Team. hls4ml (Version v0.7.1) [Computer software]. https://doi.org/10.5281/zenodo.1201549
- J. Duarte, et al. "Fast inference of deep neural networks in FPGAs for particle physics". JINST 13, P07027 (2018). https://doi.org/10.1088/1748-0221/13/07/P07027

Backup

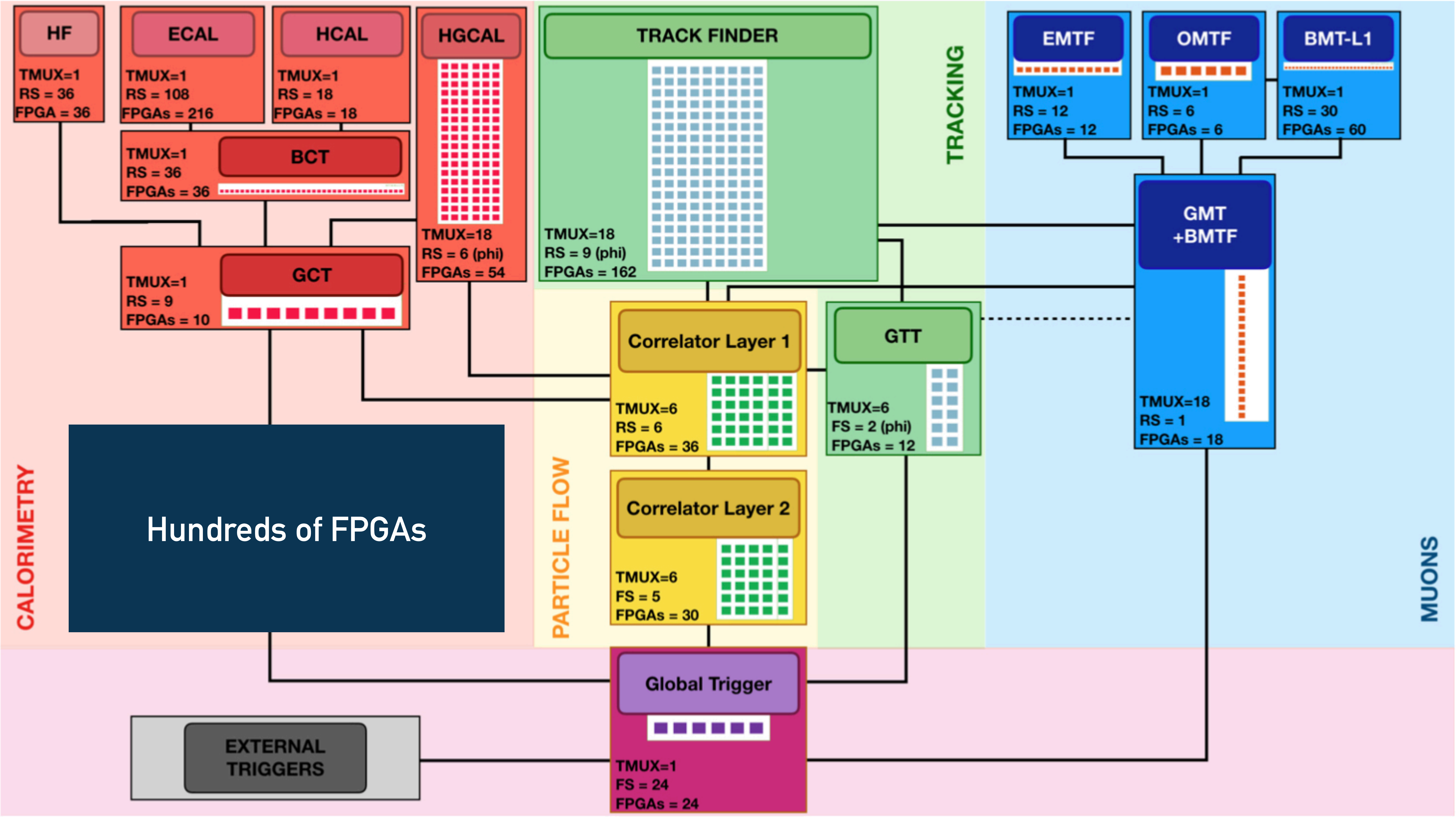
Towards the High-Luminosity LHC

CMS L1 Trigger for the High-Luminosity LHC

- **High-Luminosity phase of the LHC (HL-LHC) will start in 2029:** 3x higher instantaneous luminosity and pileup wrt current conditions
  - CMS will upgrade most of its detectors, including the (trigger) electronics
- **L1 Trigger for the HL-LHC:**
  - Bandwidth: 2 –> 63 TB/s
  - Output 100 –> 750 kHz
  - Latency: 4 –> 12 us
- **Tracking @ L1T + new processing systems will enable "offline-like" reconstruction**



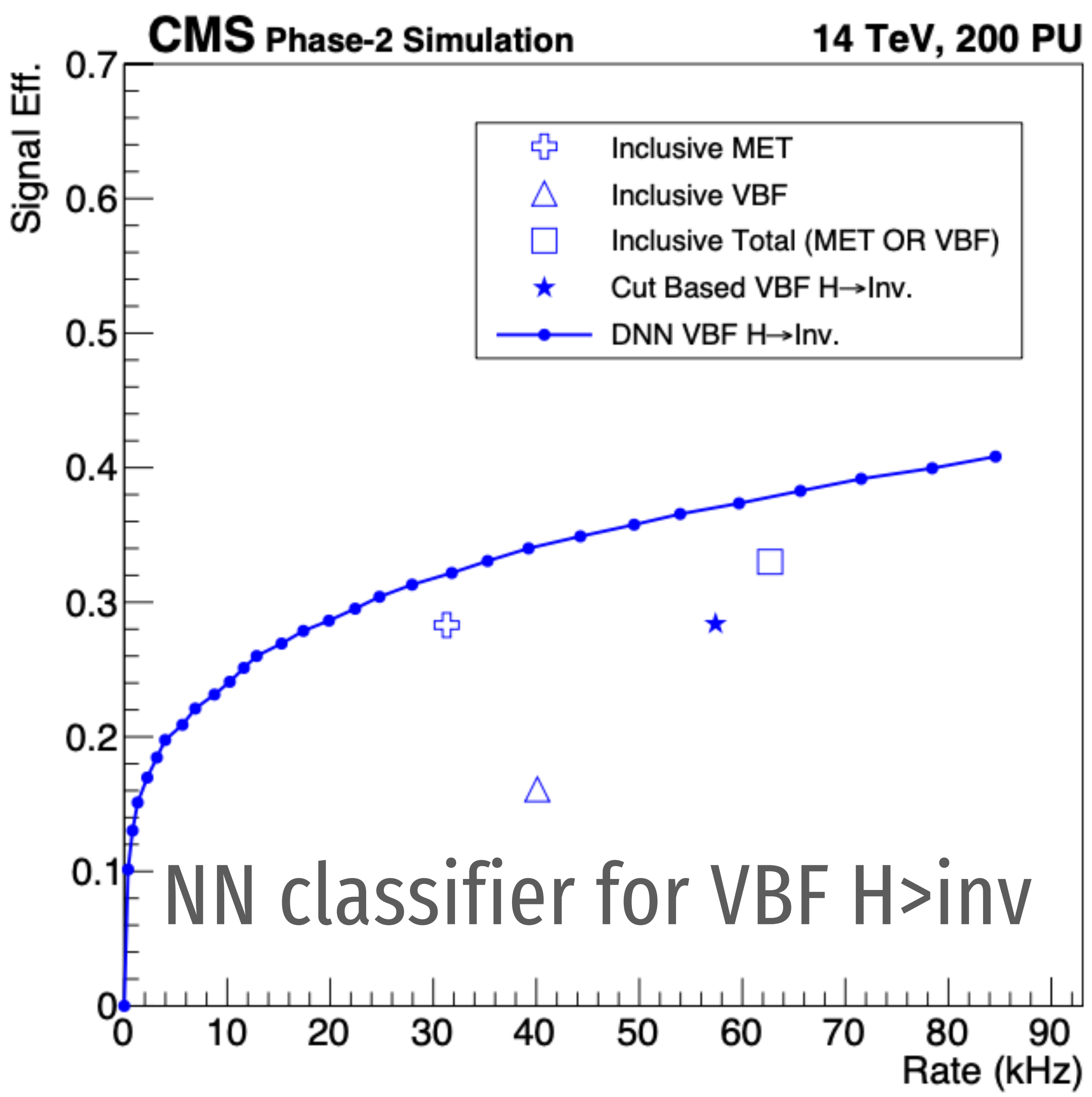
FPGAs: workhorse of the CMS Level-1 Trigger

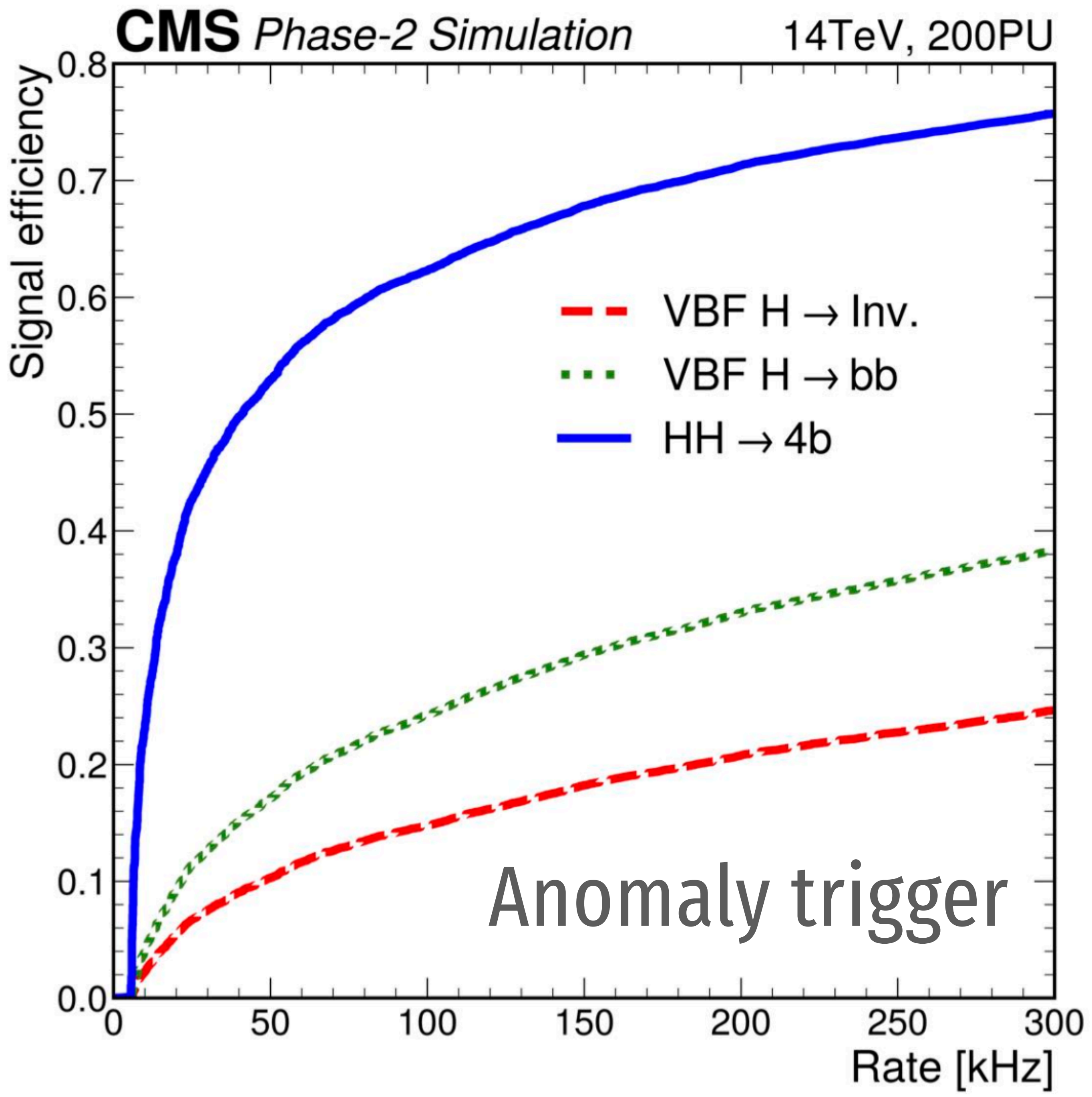


L1 Anomaly Triggering @ HL-LHC

- ML-based triggers proposed in the L1T "TDR" for the High-Luminosity LHC
- **Classifier approach**: binary classifier for known signals trained on simulation (DNN)
- **Anomaly detection**: auto-encoder based on L1 trigger objects (as AXOL1TL)
  - Sensitivity at the ~same order as of the classifier approach (e.g. VBF H>inv)

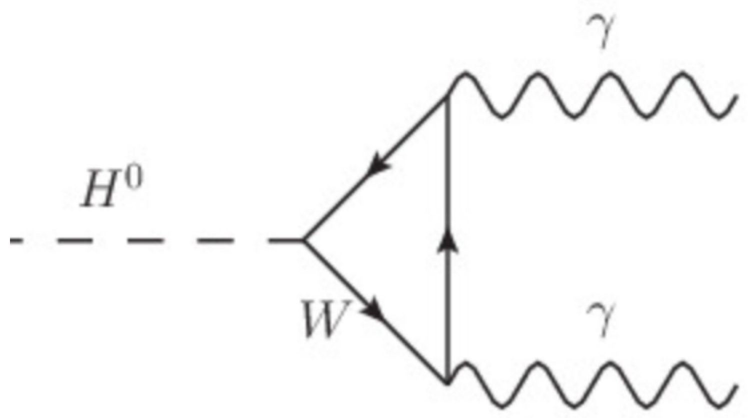
- **Tests of AXOL1TL and CICADA pave the way for anomaly triggering at the HL-LHC in CMS!**

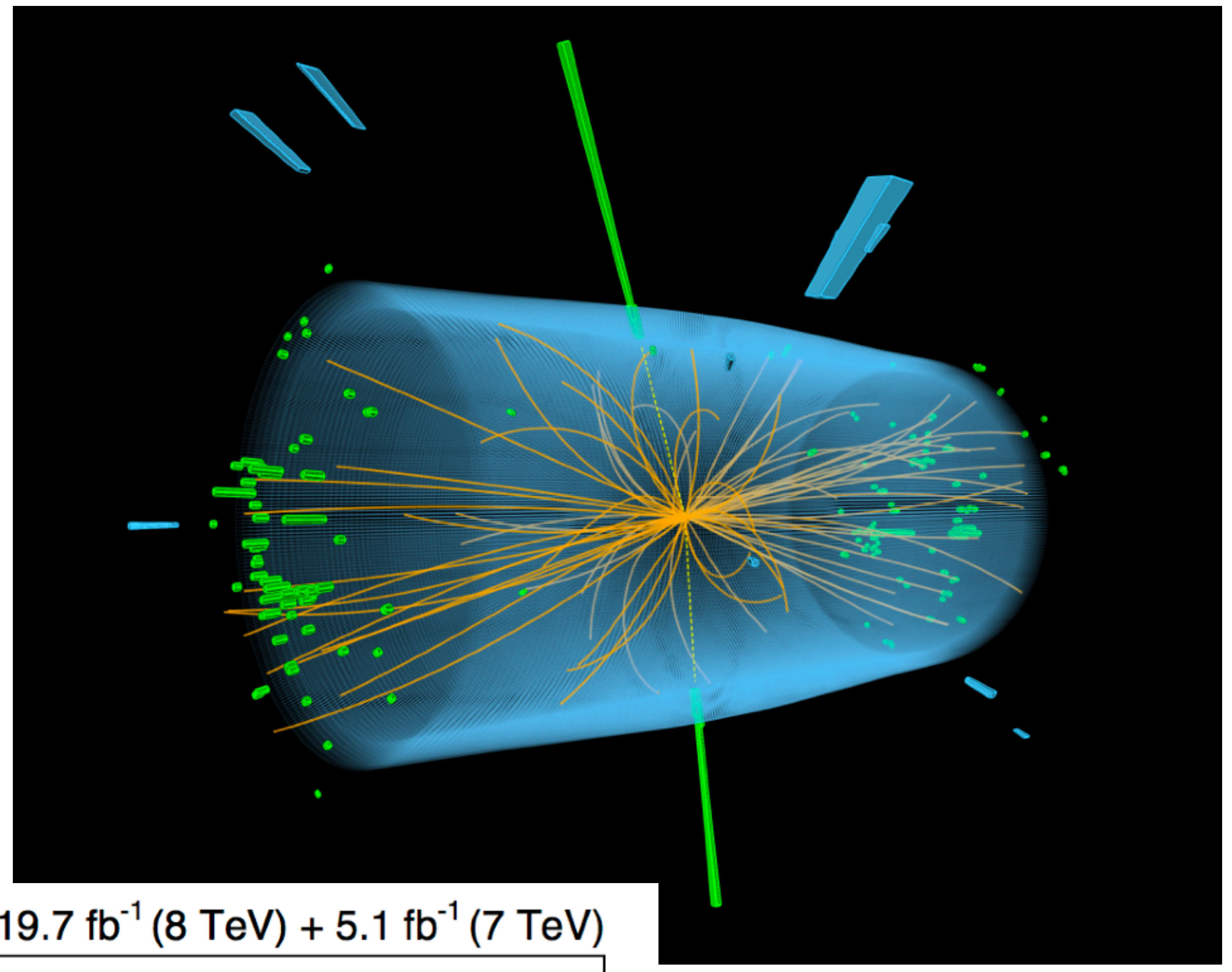




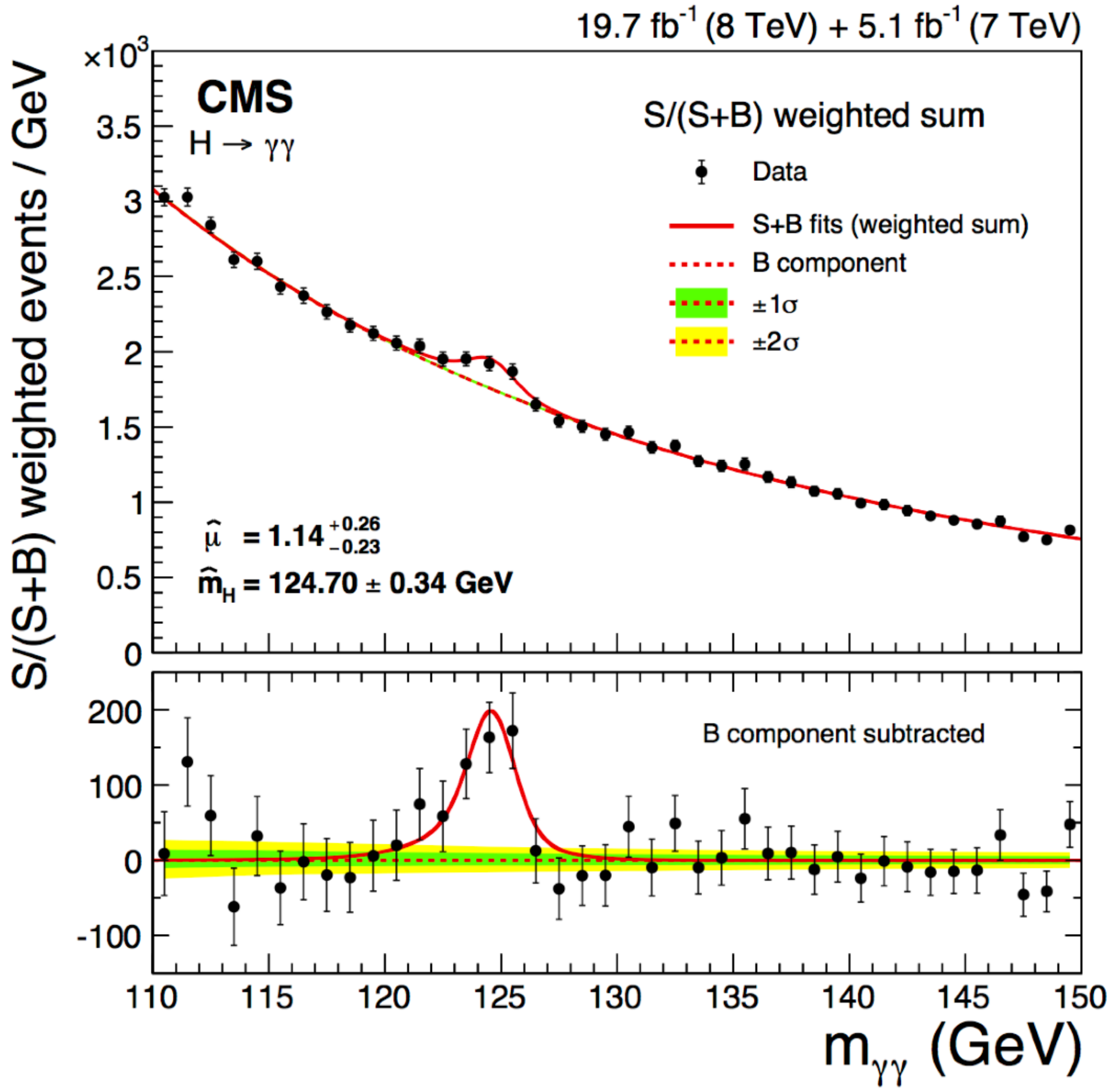
EXTRACTING ANOMALIES FROM LHC DATA

Example signal: Higgs decay to two photons





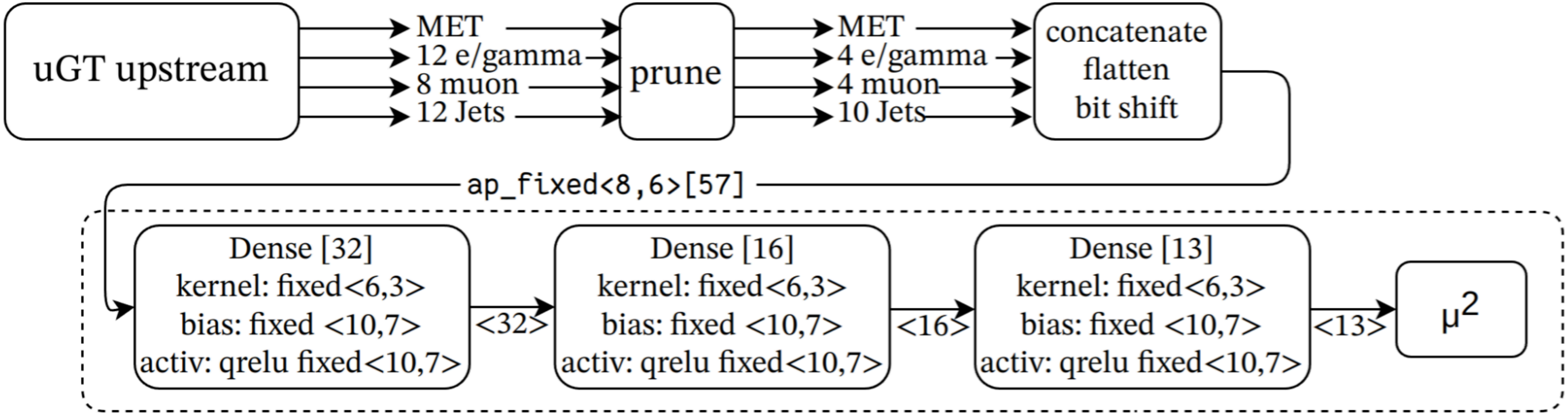
1. Select events: 2 high-energy photons
2. Reconstruct H candidates: invariant mass of two photons
   - Higgs is a resonance –> peak in m_yy spectrum
   - Backgrounds –> falling spectrum
3. Hypothesis testing p(theory|data):
   - Null hypothesis: background-only
   - Signal hypothesis: signal+background



- **New physics can affect/appear in/ all stages**

AXOL1TL: COMPRESSION

- **Quantization-aware training with QKeras and FPGA adaptation with HLS4ML**
  - Narrow, shallow model, aggressively quantised
- Output is one vector [13,1], corresponding to μ part of [μ,σ] KL loss (dropping σ as it is small -> reduces processing time)
- **Anomaly score: sum squared of the μ vector**



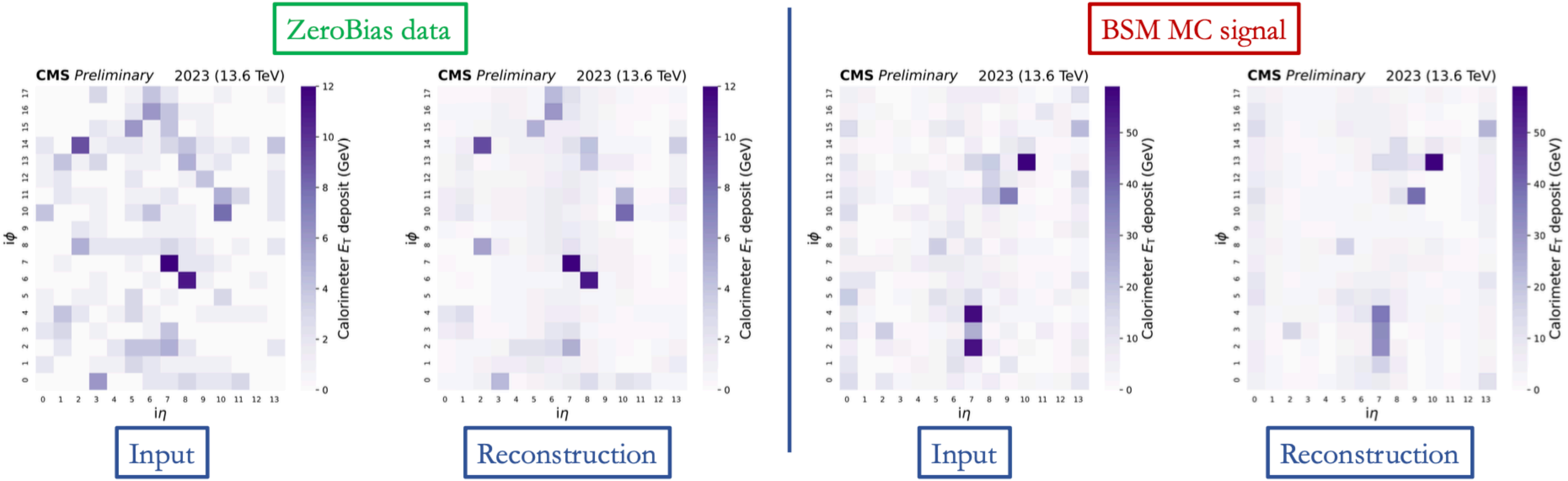
AXOL1TL: LOSS TERM

The AXOL1TL anomaly detection uses a Variational Autoencoder (VAE). A dense feed-forward neural network reads in (p_T, η, ϕ) hardware inputs of 19 L1 objects. The encoder network computes a latent space vector of Gaussian probability distributions, $N(\mu_8, \sigma_8)$. The decoder network reconstructs the original input from the latent space.

$$\text{Loss} = \underbrace{(1-\beta)\left\|x - \hat{x}\right\|^2}_{\text{Reconstruction term}} + \underbrace{\beta\frac{1}{2}(\mu^2 + \sigma^2 - 1 \ - \log\sigma^2)}_{\text{Full regularization term}}$$

Equation: VAE loss function. The reconstruction term is computed from the difference between the input (x) and output ($\hat{x}$) of the VAE. The second, full regularization term, is the Kullback–Leibler divergence (KL-divergence) between the latent space distribution and a standard normal distribution with mean μ and standard deviation σ. The parameter β can be tuned to balance the reconstruction performance with more efficient latent space encoding. At inference time, the loss is approximated by the mean-squared term $\Sigma\mu_i^2$ of the KL-divergence for latency considerations. This approximation has no impact on performance.

CICADA: Anomaly detection on Raw inputs

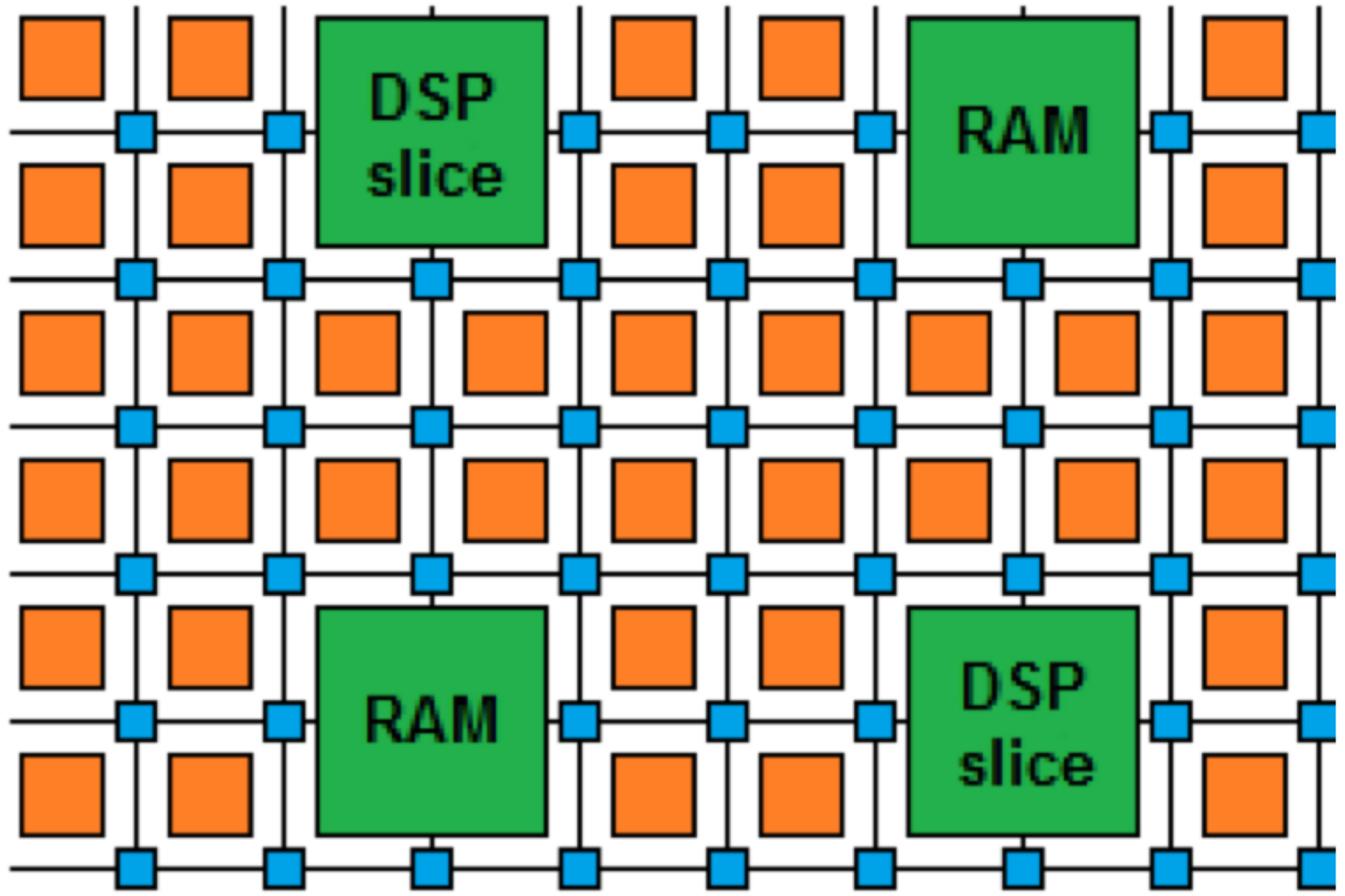


Shown here is a comparison of the teacher model ability to reconstruct a Zero Bias (ZB) beam event (original: far left, reconstructed: center left) versus a signal sample, Soft Unclustered Energy Patterns (SUEP) on the right (original: center right, reconstructed: far right). In general, the teacher model is better able to reconstruct the Zero Bias beam event as evidenced by a far lower loss (0.81) compared to the SUEP loss (14.21). This example shows how the CICADA anomaly detection mechanism works to find anomalies. From [CMS DP-2023/086]

ML@FPGA

FPGA: Field Programmable Gate Arrays

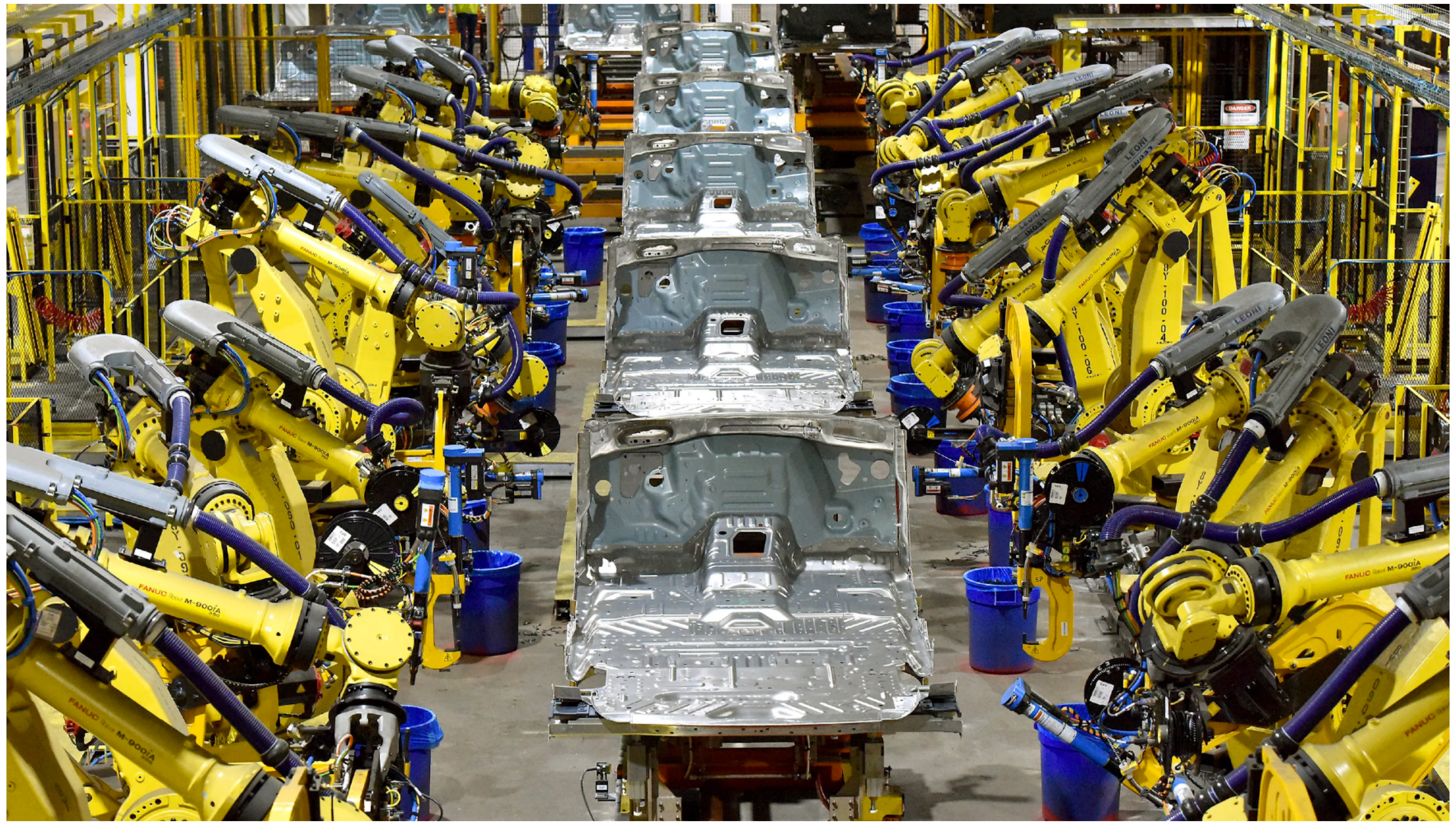
- **Integrated circuit** with **programmable logic**
  - Originally **introduced for prototyping** Application-specific Integrated Circuits (ASICs)
- Contrary to ASIC: **(re)programmable in the "field"**
- FPGAs consists of **different parts of logic cells:**
  - Look-up Tables (LUT), Flip-Flops (FF), Digital Signal Processors (DSP)
  - Also contain RAMs, fast I/O etc,



Wiki

Why are FPGAs FAST?

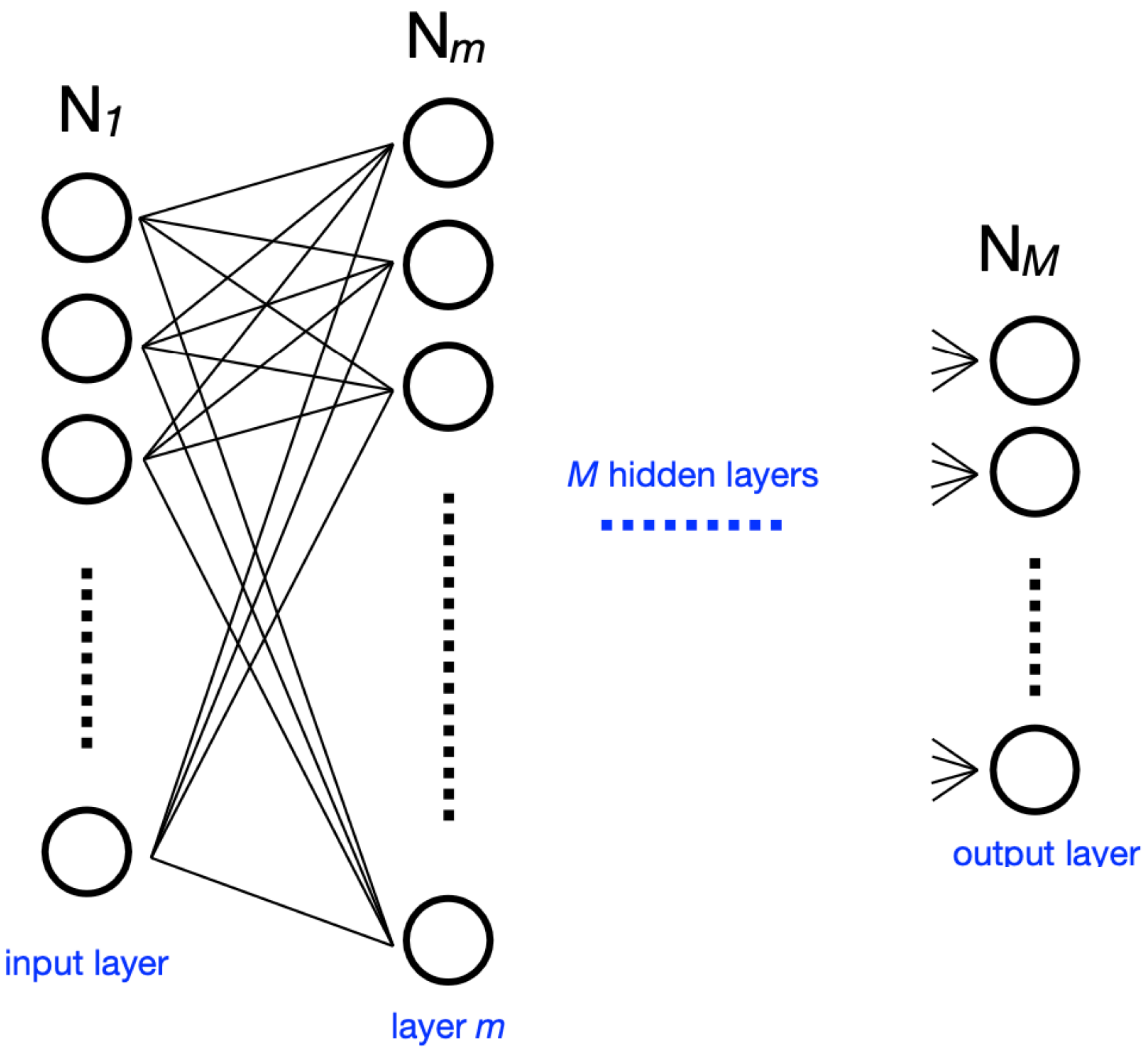
- **Resource parallelism**
  - Use the many resources to work on different parts of the problem simultaneously
  - Achieve **low latency**
- **Pipeline parallelism**
  - Use the register pipeline to work on different data simultaneously
  - Achieve **high throughput**



FPGAs as a data conveyor belt

Why ML@FPGA?

- Example: fully connected Neural Network



$$x_m = g_m \left(W_{m,m-1} x_{m-1} + b_m \right)$$

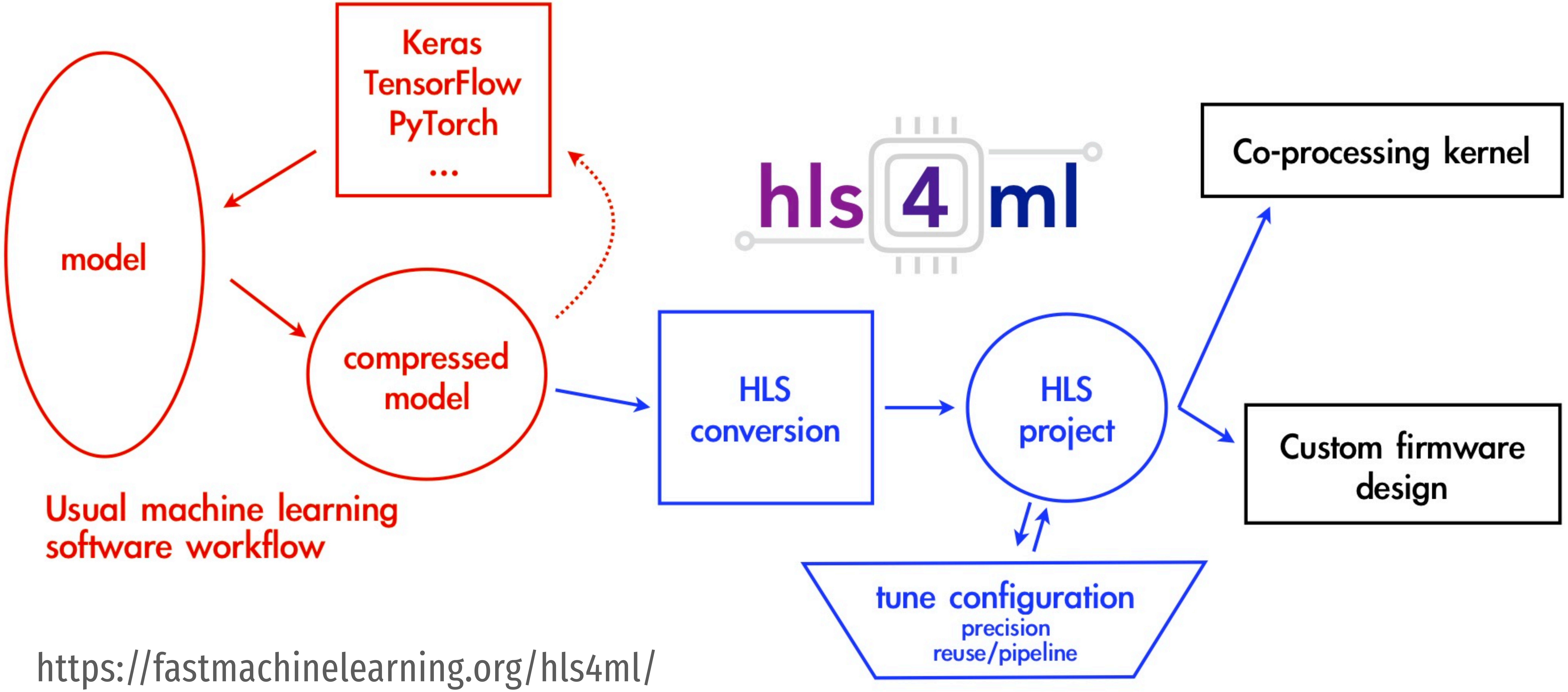
activation function: precomputed and stored in BRAMs
multiplication: DSPs
addition: logic cells

Parallelise-able and robust against reduced precision

Perfect for **ML Inference**

FROM PC TO FPGA

- hls4ml: package for translating NN to FPGA firmware



https://fastmachinelearning.org/hls4ml/

Efficient NN design: quantization

ap_fixed<width bits, integer bits>

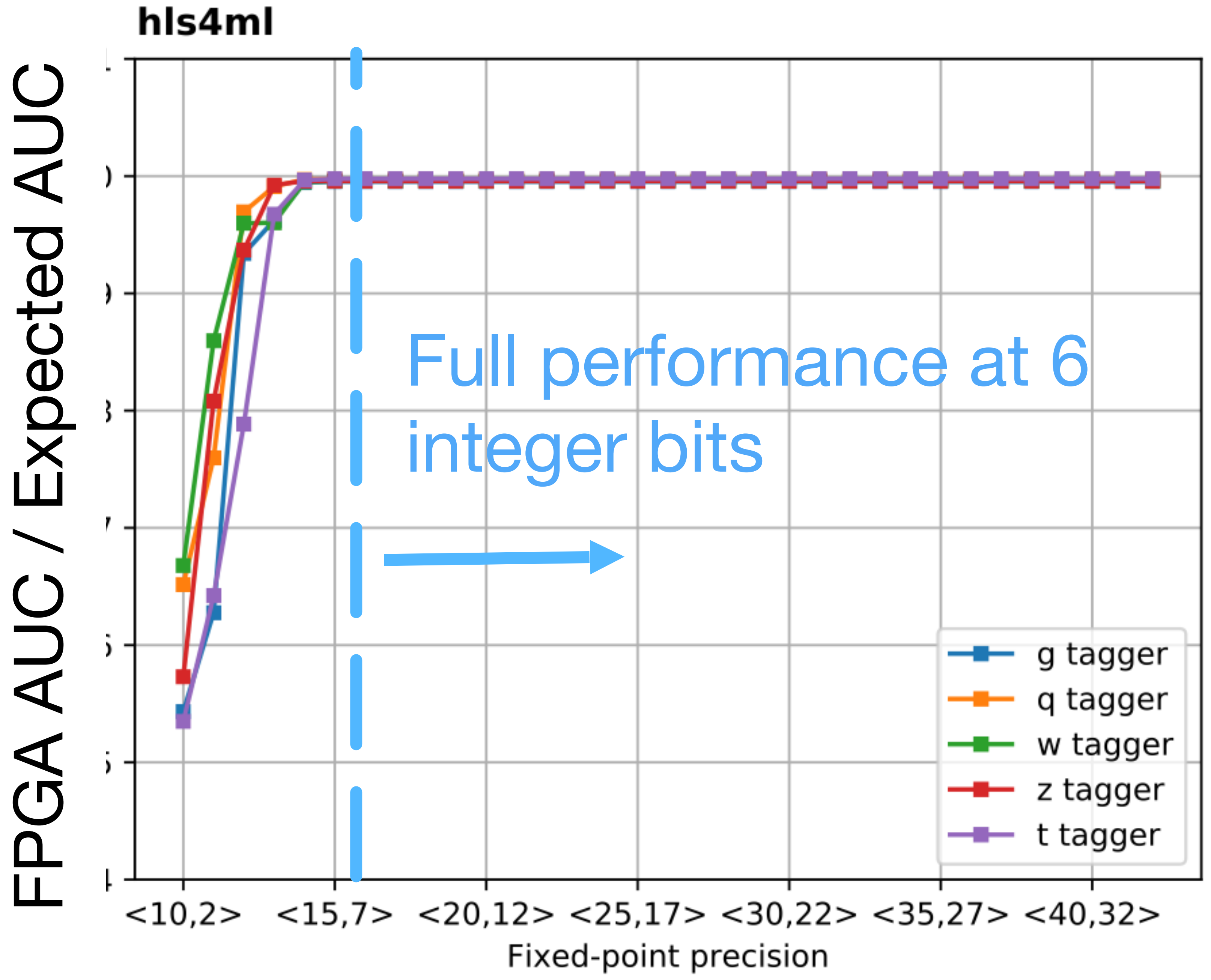
0101.1011101010

integer
fractional
width

- **In the FPGA fixed point representation is used!**
- Operations are integer ops, but one can represent fractional values
- But we have to make sure we've used the correct data types!

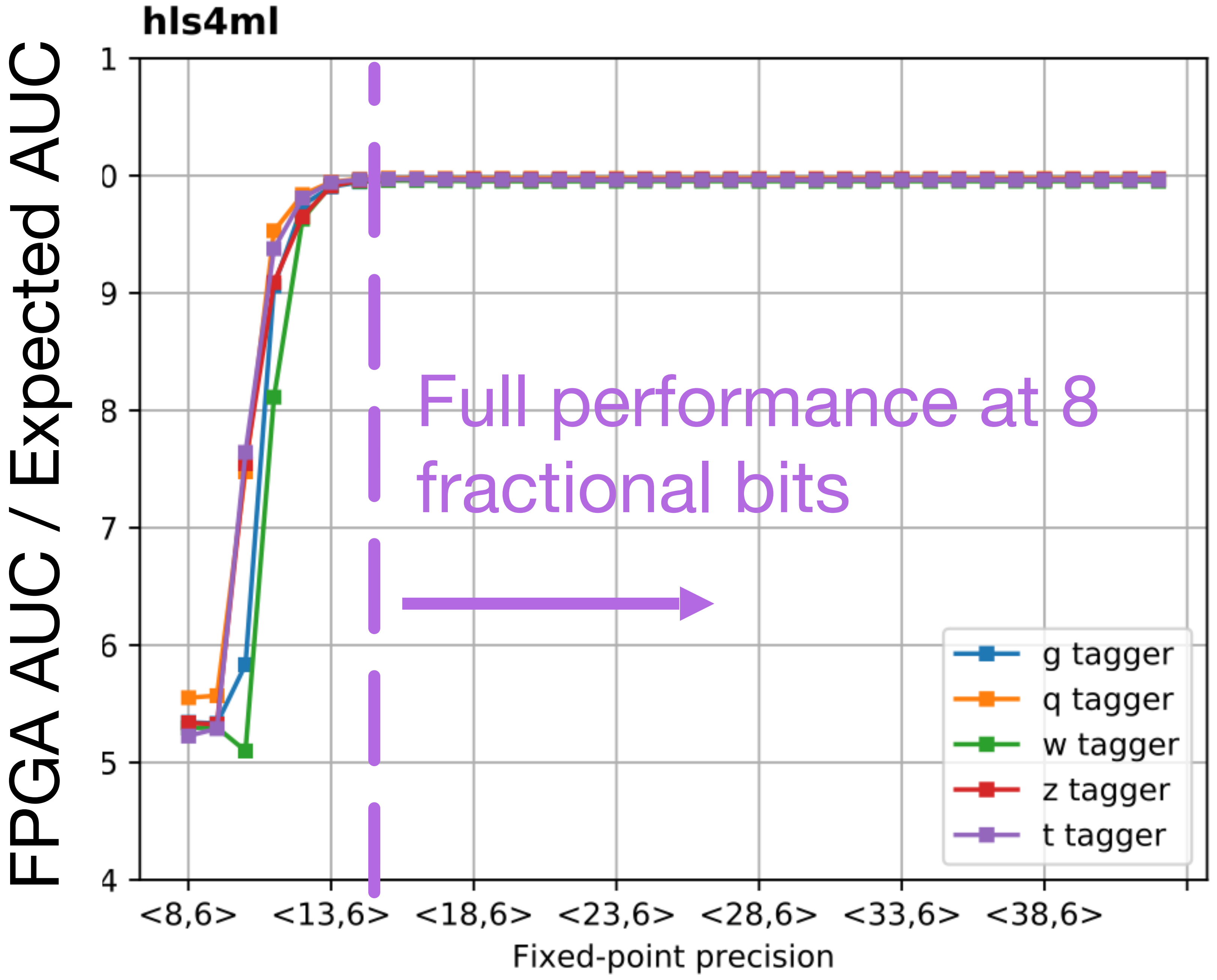
Scan integer bits

Fractional bits fixed to 8



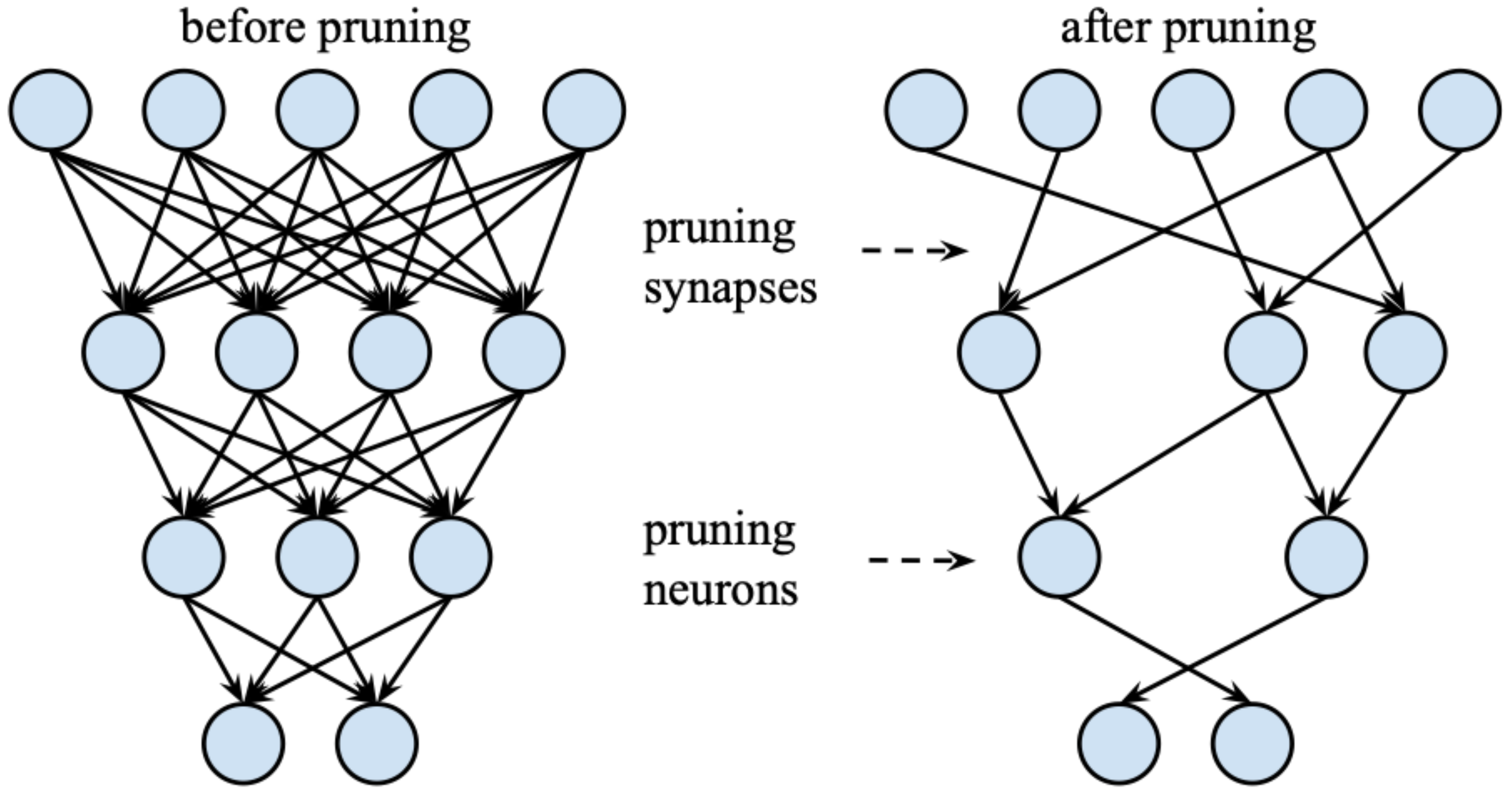
Scan fractional bits

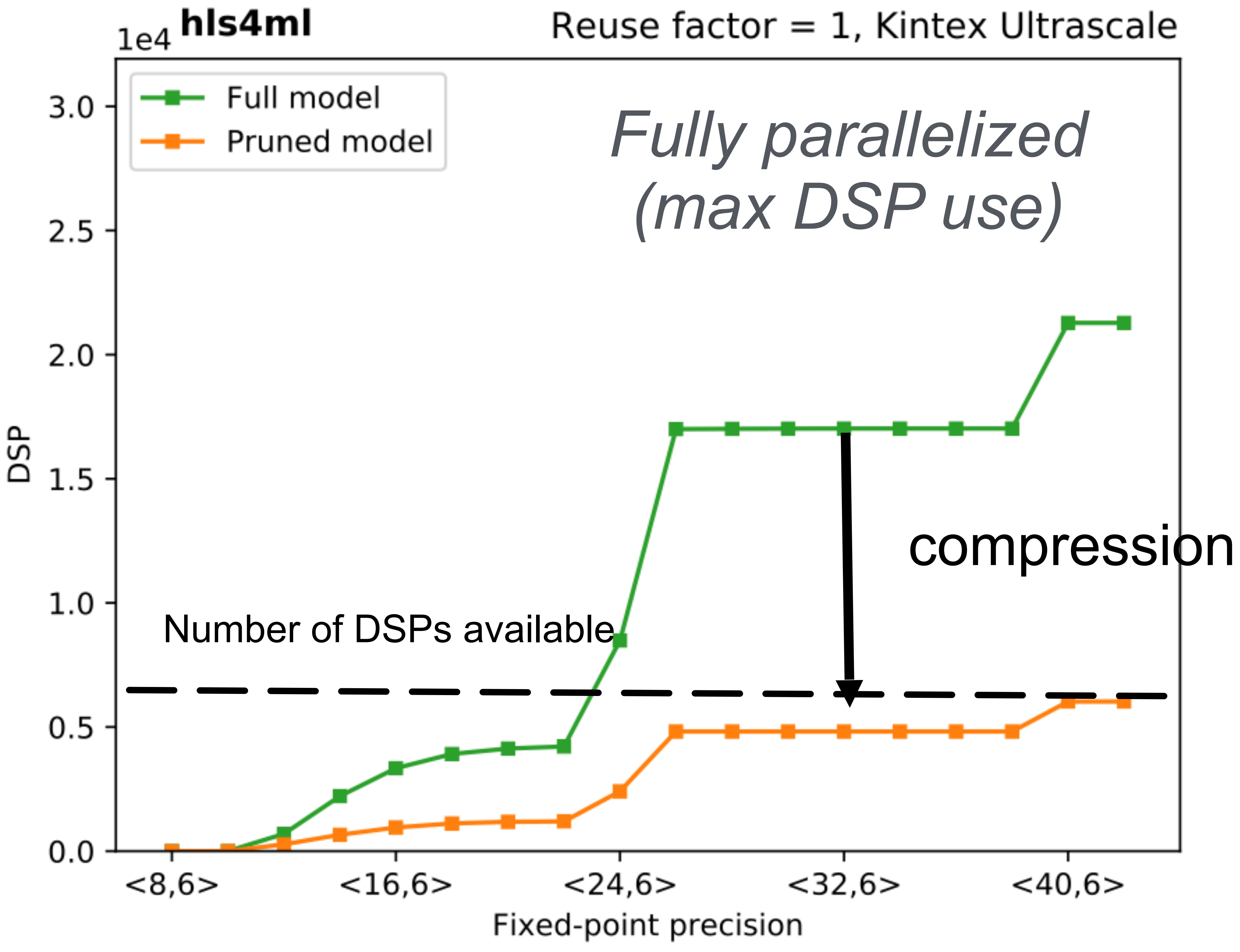
Integer bits fixed to 6



EFFICIENT NN DESIGN: COMPRESSION

- **Network compression:**
  widespread technique to **reduce the size, energy consumption, and overtraining** of deep neural networks
- Remove redundancy in model: crucial for FPGAs!

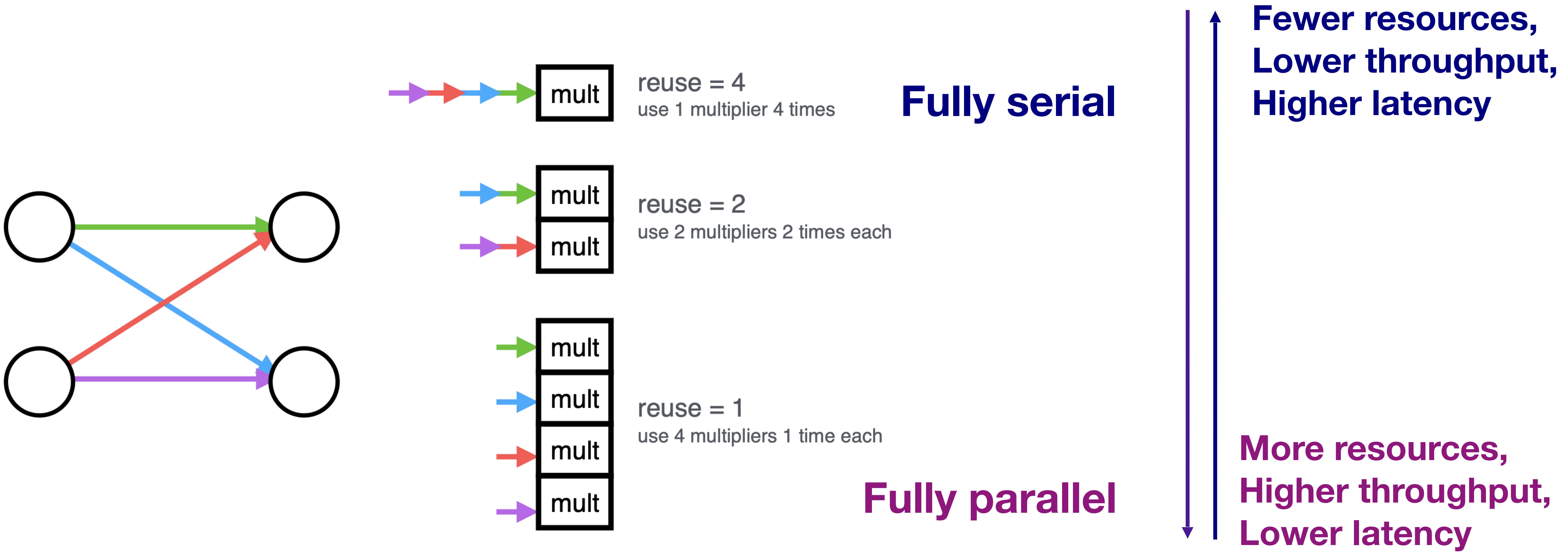




70% compression ~ 70% fewer DSPs

EFFICIENT NN DESIGN: PARALLELIZATION

- **Trade-off between latency and FPGA resource** usage determined by the parallelization of the calculations in each layer
- Configure the "reuse factor" = number of times a multiplier is used to do a computation



Reuse factor: how much to parallelize operations in a hidden layer