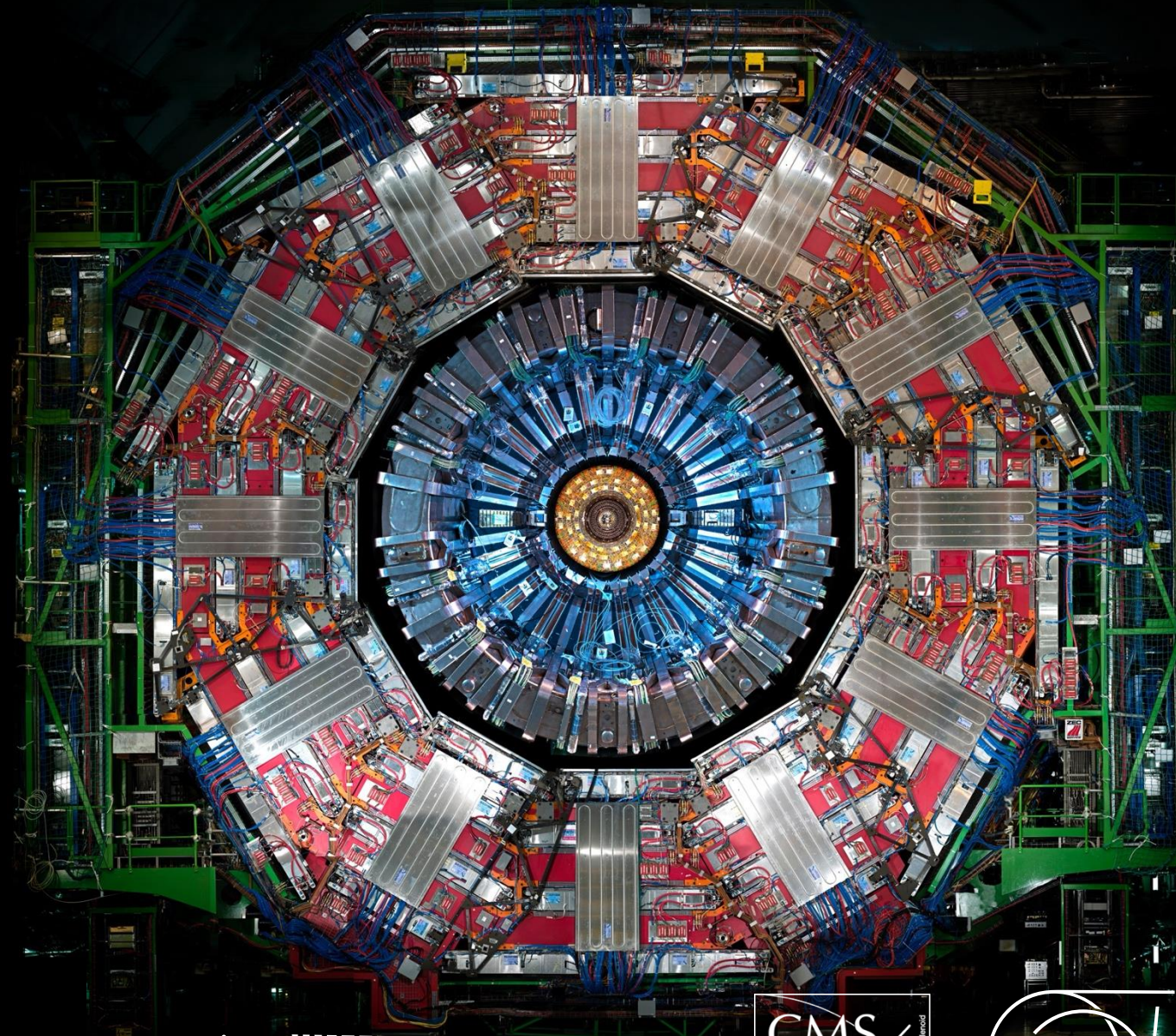


AI at the Extreme Edge

Jannicke Pearkes

ML4Jets

Caltech - Aug 21st, 2025



University of Colorado **Boulder**



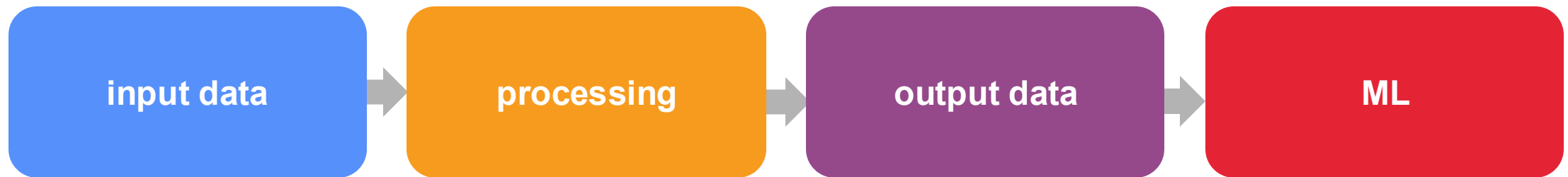
NextGen
Next Generation Triggers



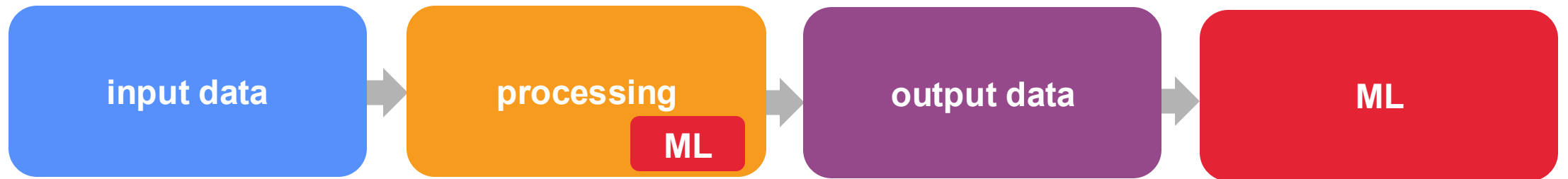
What is Edge AI/ML?

Computation as close to the input data source as possible

Traditional model:



Edge AI/ML model:



Useful for making the most out of low latency applications

What is Edge AI/ML?

Computation as close to the input sensor as possible

Example:

Taking a picture with your phone's autofocus feature



What is Edge AI/ML?

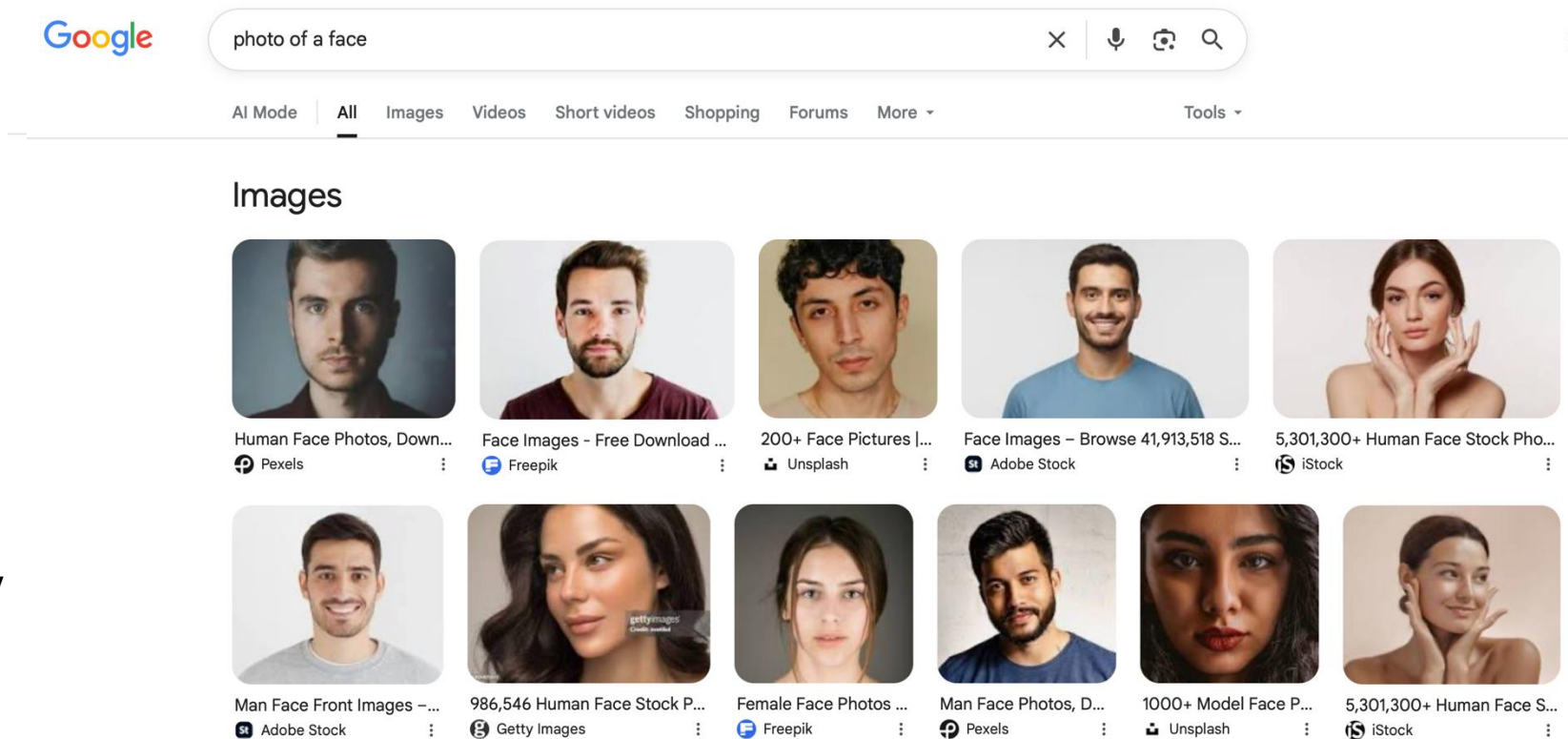
Computation as close to the input sensor as possible

Example:

Use your phone to see if it picks up the faces in this image.



(turn on airplane mode to check if the inference is happening directly on your phone)



What is Edge AI/ML?

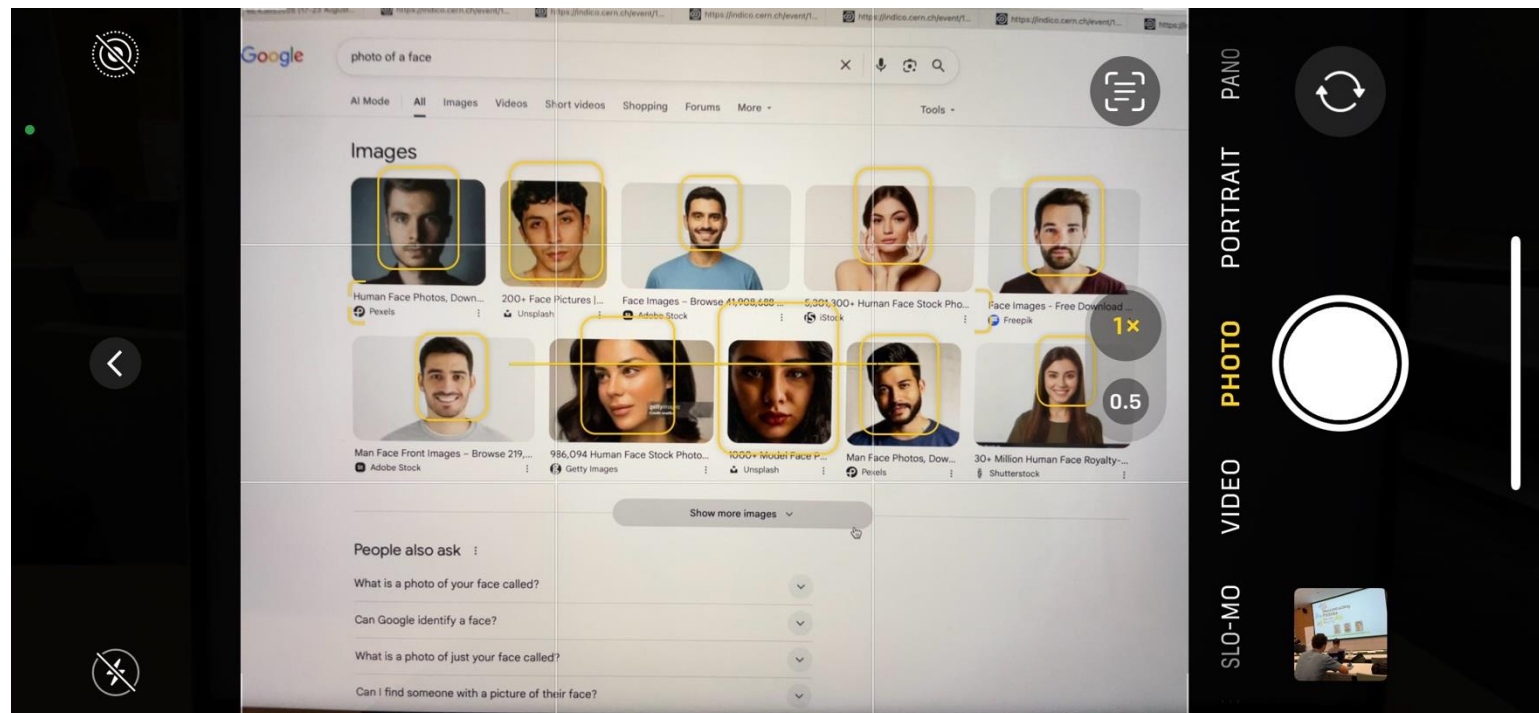
Computation as close to the input sensor as possible

Example:

Use your phone to see if it picks up the faces in this image.



(turn on airplane mode to check if the inference is happening directly on your phone)



Useful for real-time applications - like focusing on things that matter.

What is Edge AI/ML?

Computation as close to the input sensors as possible

Example:

Self-driving car – onboard computer making decisions

*Useful for applications where **bandwidth** & network responses are a constraint.*



What is Edge AI/ML?

Computation as close to the input sensors as possible

Example:

Wearable devices

*Applications where **power** is an important constraint – often more power efficient to do computations locally.*



ML in particle physics:

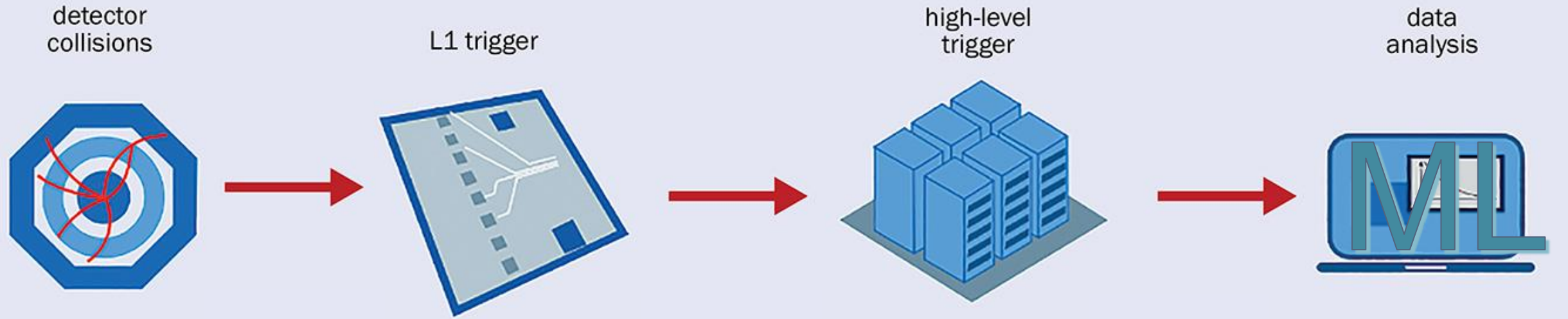
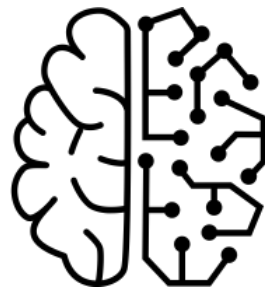


Image: CERN Courier



ML widely used in offline analysis. Trends towards bigger, more complex models with better performance.

ML in particle physics:

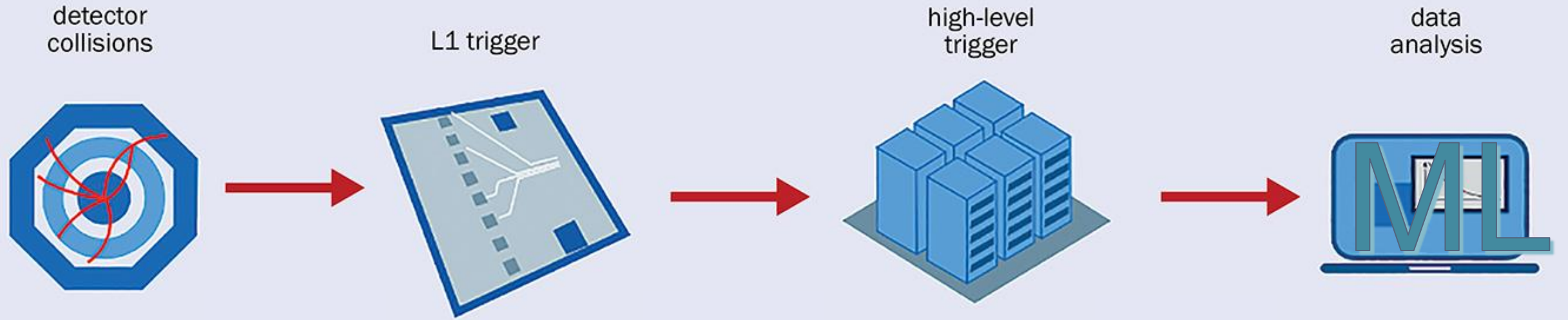
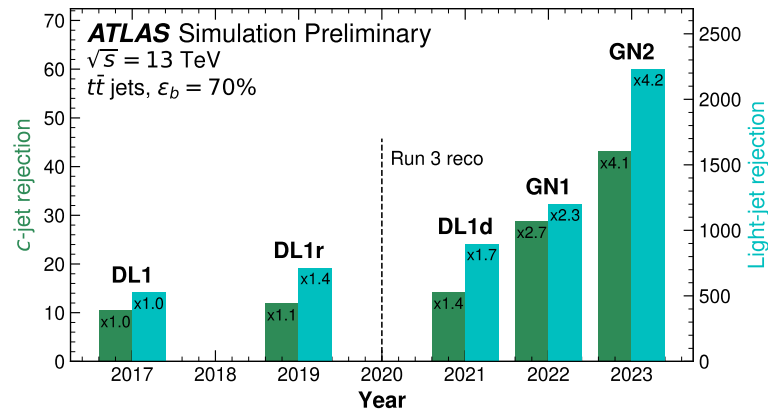


Image: CERN Courier



FTAG-2023-01

ML widely used in offline analysis. Trends towards bigger, more complex models with better performance.

ML in particle physics:

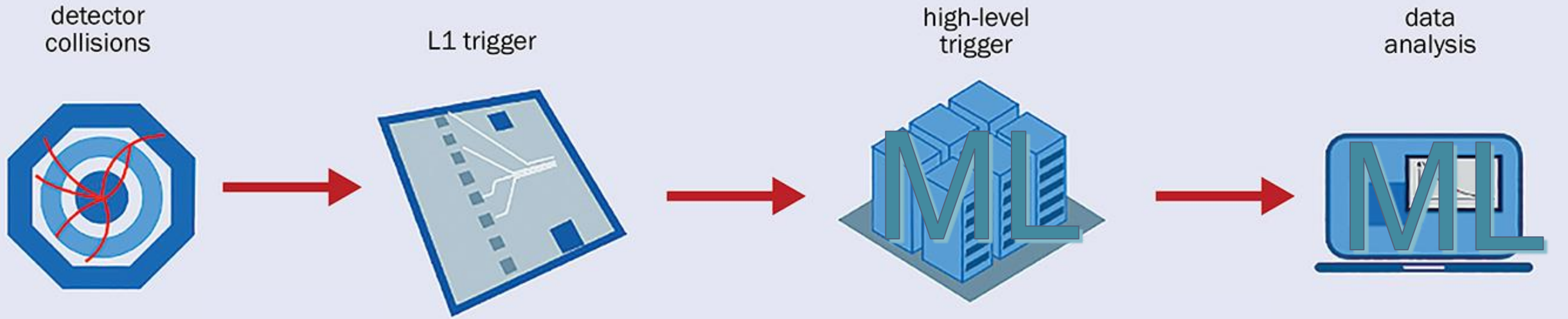


Image: CERN Courier

ML also deployed at HLT for reconstruction tasks such as b-jet tagging.

ML in particle physics:

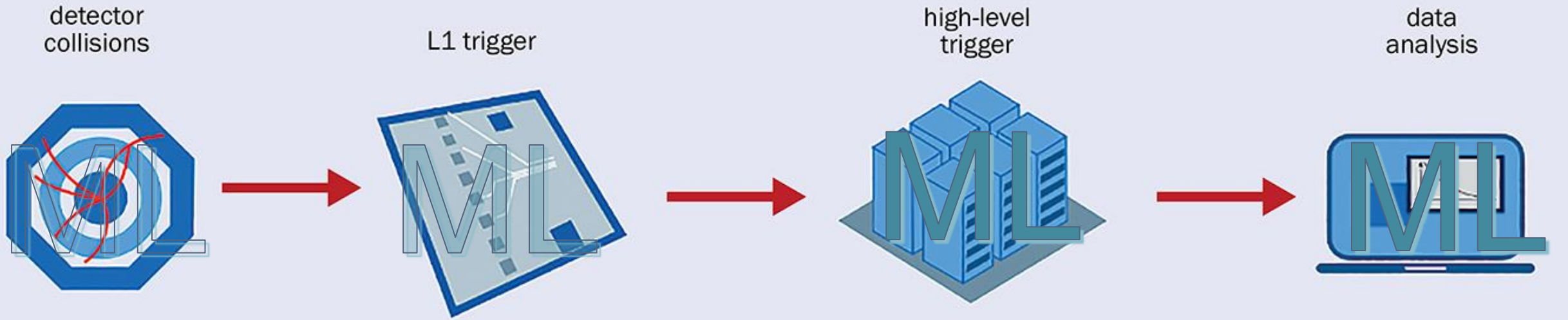
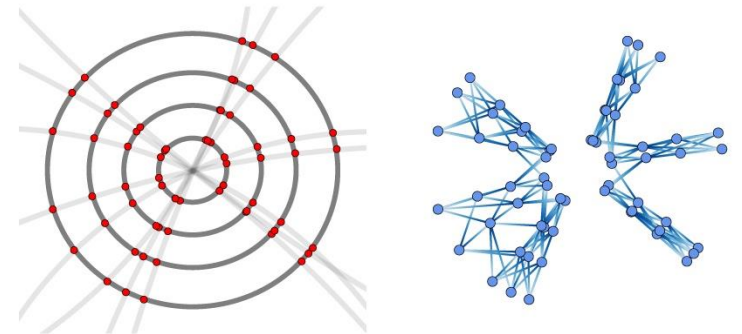


Image: CERN Courier

Growing ML applications here!

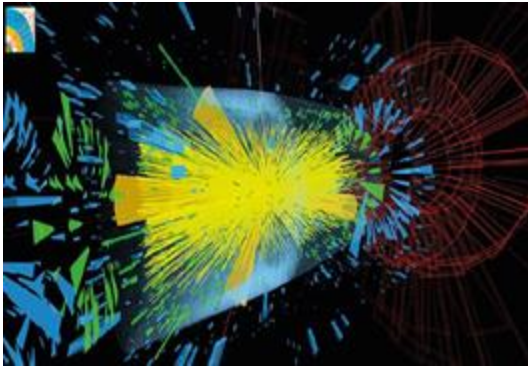
Intelligent:

- Compression
- Reconstruction
- Filtering



A league of our own:

~1-100 ns



~5 ms



~50 ms



~1-2s



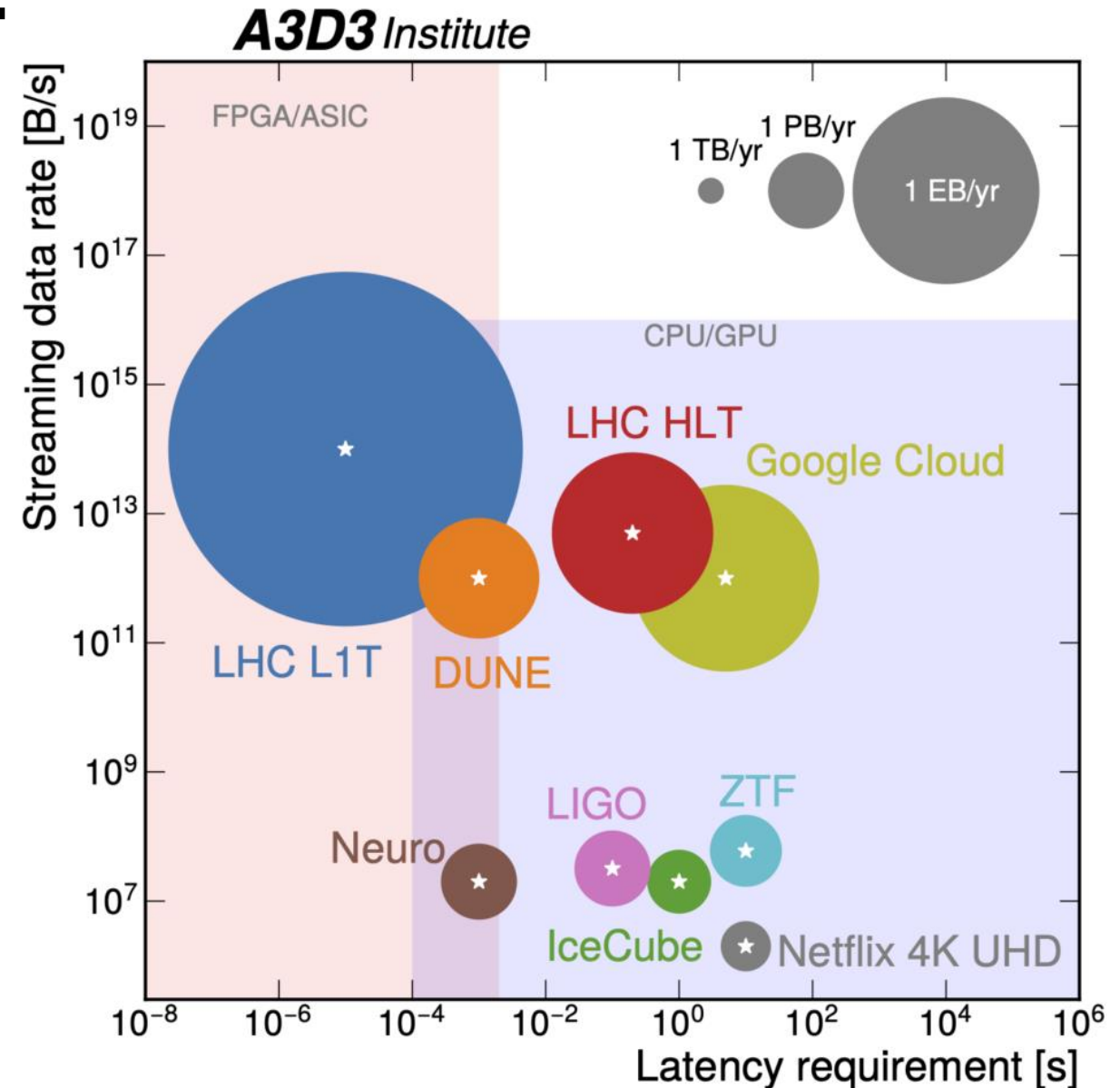
Many of the edge ML industry applications in industry run on timescales of achievable by microprocessors/GPU/CPU's.

A league of our own:

Extreme requirements on throughput, power and latency:

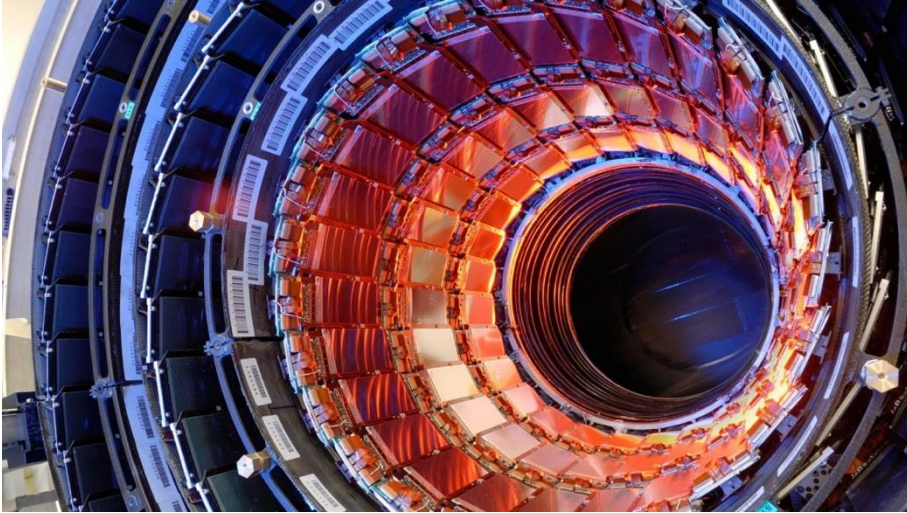
- Collisions every 25ns
- ~100 TB/s to be processed per experiment
- ~5% of global internet traffic¹

1) 1779 TB/s global internet bandwidth in 2024 ([ITU](#))



Why are our data rates so high?

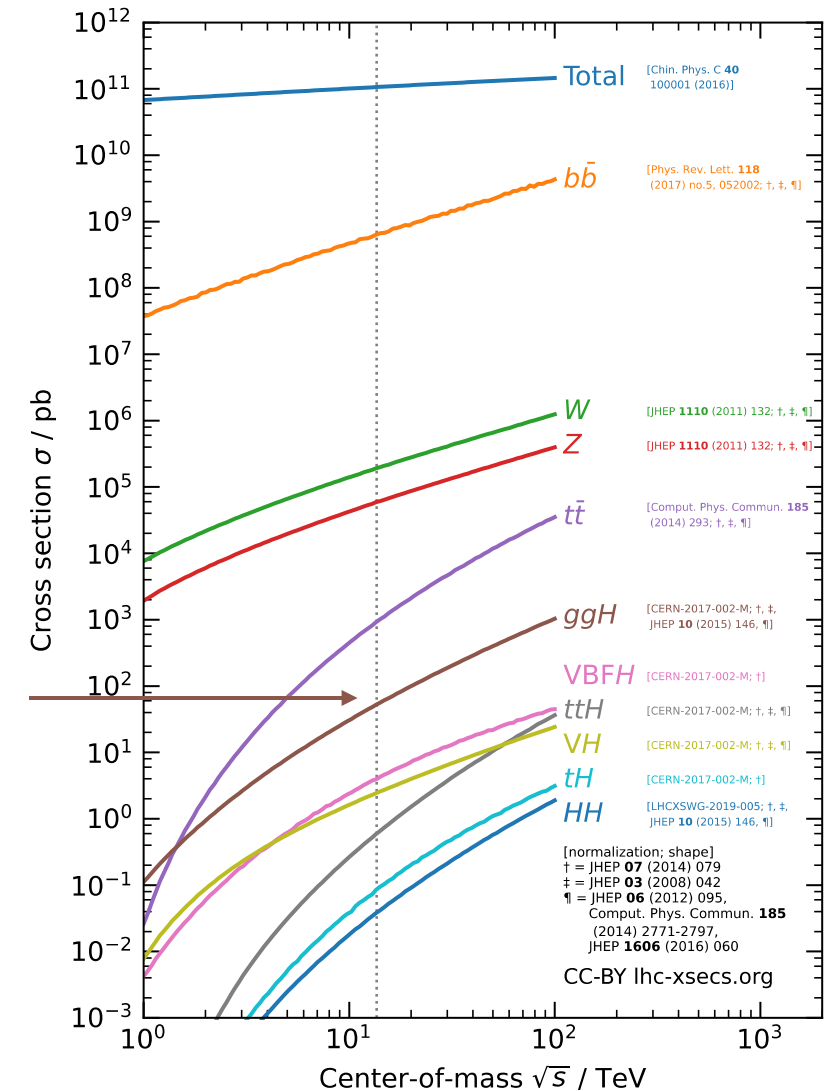
Our detectors are *very granular*



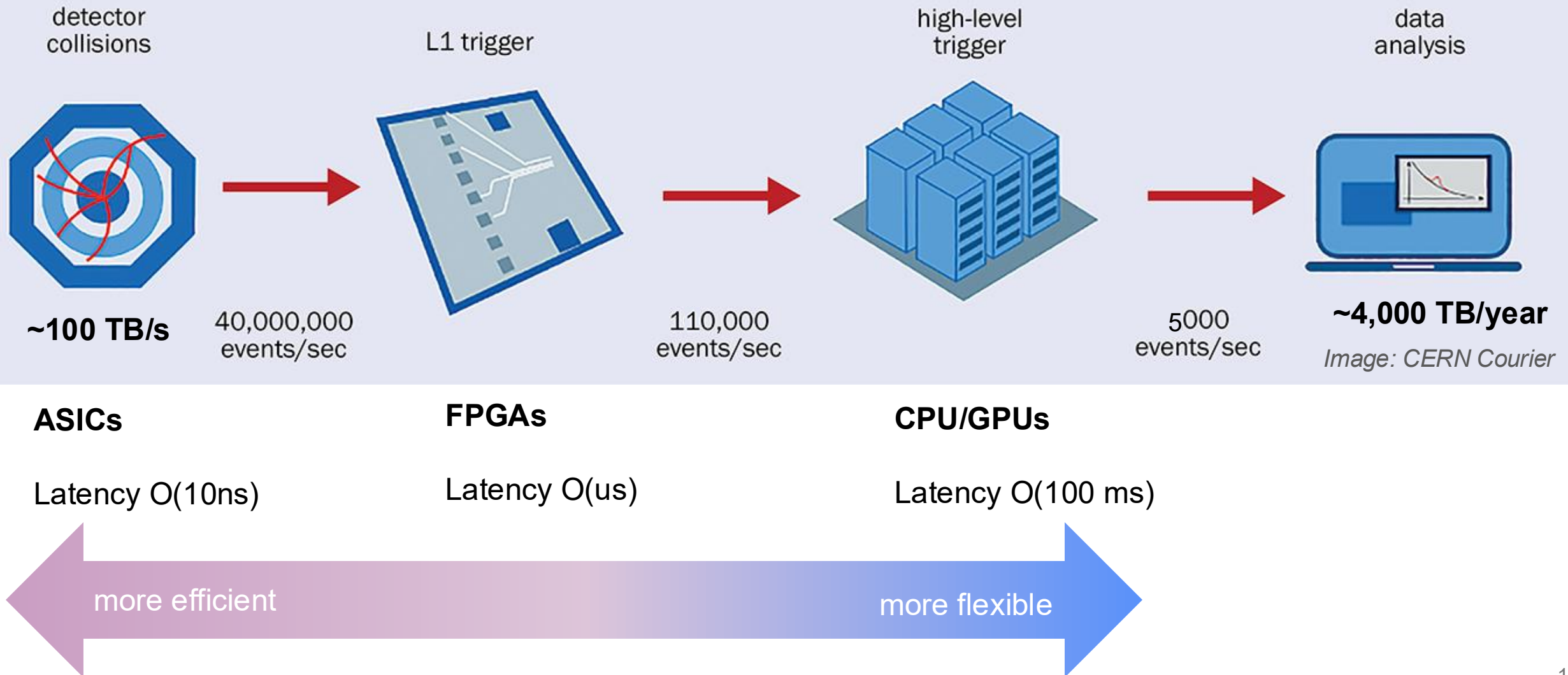
CMS tracker in Run 3 has ~135,000,000 readout channels.

~1 Higgs boson per
billion collisions

A lot of the physics we are interested in is
very rare.



Technologies in LHC detector readout stack:



What is an FPGA?

Field Programmable Logic Gate Array

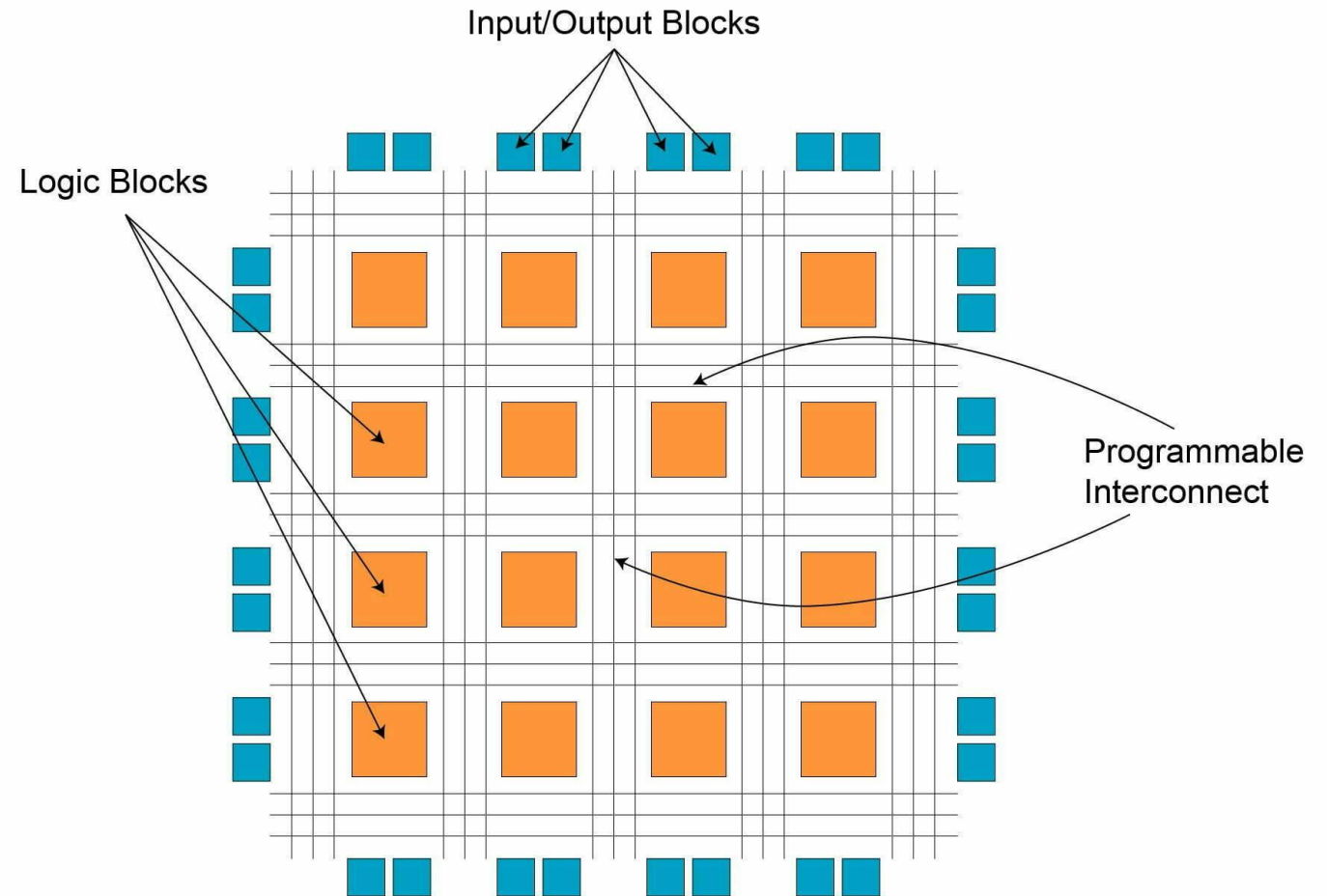
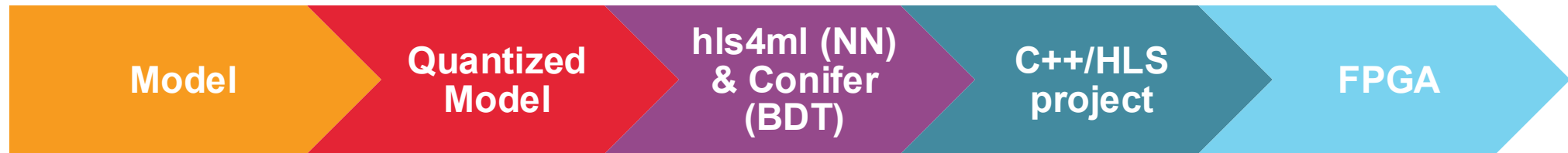
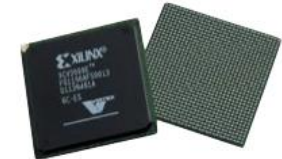
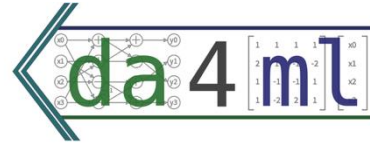


Image: Logic Fruit Technologies

ML Toolkit



Quantization and pruning:
QKeras, AutoQ (Keras)
Brevitas (Pytorch), HGQ



C++ Model

```
// defines.h - layer-precision
typedef ac_fixed<6,1,true> weight2_t;
typedef ac_fixed<16,6,true> layer2_t;
...
// w2.h - weights
weight2_t w2[1024] = {-0.1250,
-0.1875,...};
...
// myproject.cpp - network model
void myproject() {
  nnet::dense<layer2_t>{...};
  nnet::relu<layer4_t>{...};
  ...
}
```

FPGA RTL

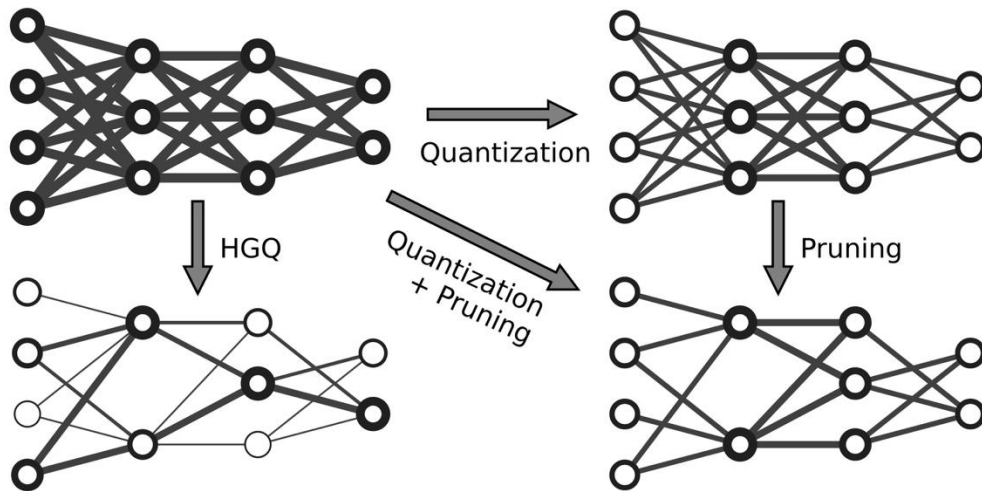
```
module myproject (
  clk, rst, fcl_input_rsc_dat,
  fcl_input_rsc_vid,
  fcl_input_rsc_rdy,
  layer13_out_rsc_dat,
  b2_rsc_dat, b2_triosy_1z, ...
)
endmodule
```


Designing ML for FPGAs

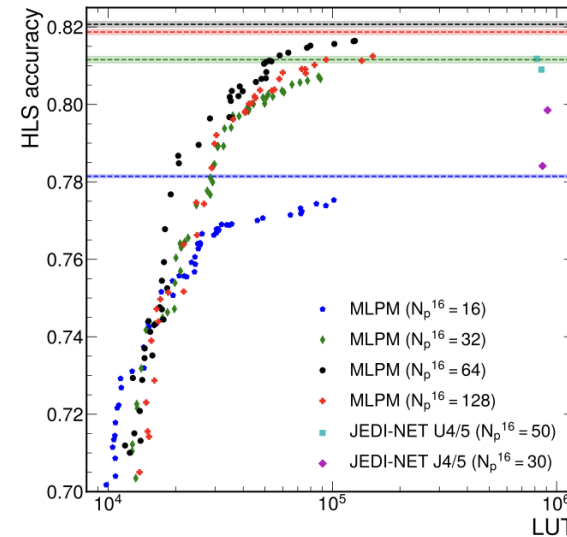
Design space exploration involves more than just optimizing the architecture for best AUC

Co-design with more constraints:

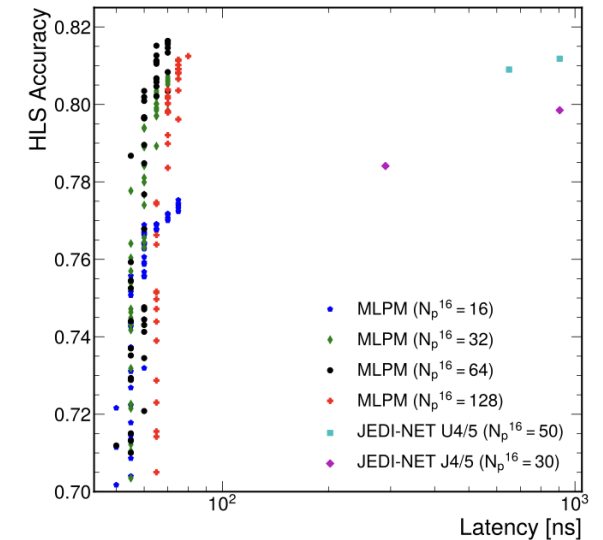
- Resource usage (DSP, LUT, FF, BRAM etc)
- Latency & initiation interval
- Power usage



Accuracy v.s. LUTs

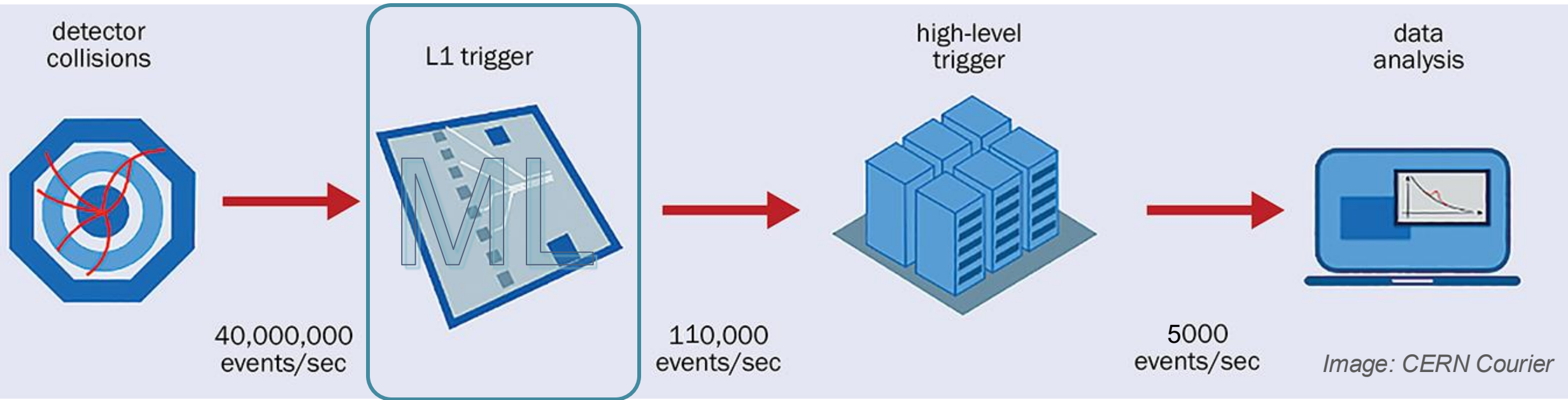


Accuracy v.s. Latency



[Chang's talk from yesterday](#)

Edge ML at the LHC in Run 3



L1 trigger rejects **99.75% of LHC events** through selection rules.

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

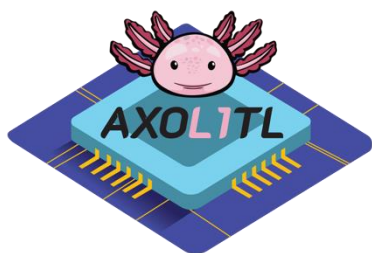


“What if we are missing new physics by using the wrong rules?”



Anomaly Detection in the L1 trigger

2024



May 2024

Start of data taking
with AXOL1TL



Oct 2024

CICADA starts taking
data

2025



May 2025

ATLAS starts taking data
with GELATO

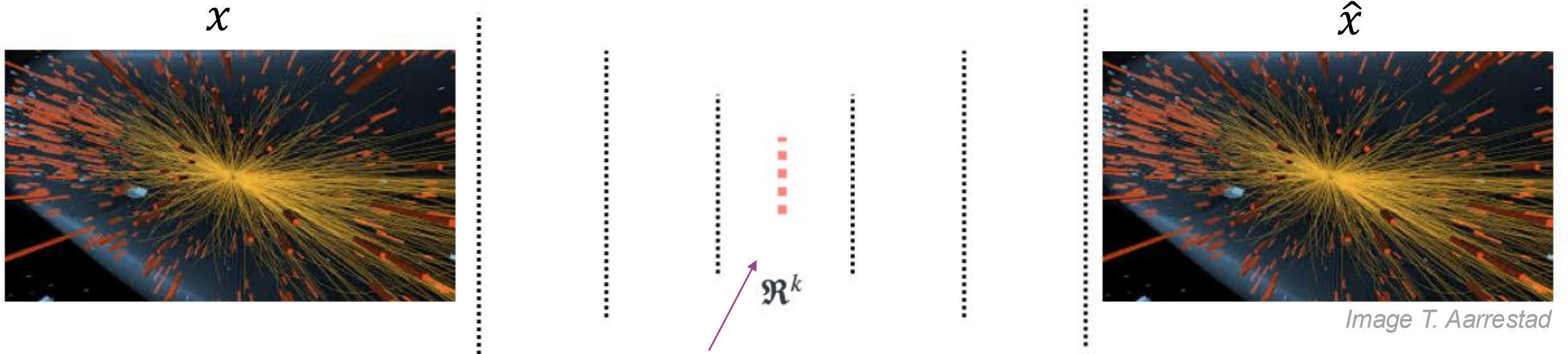
[Talk by Sagar today](#)

2025?

ATLAS starts taking data
with NOMAD?



Anomaly Detection with Autoencoders

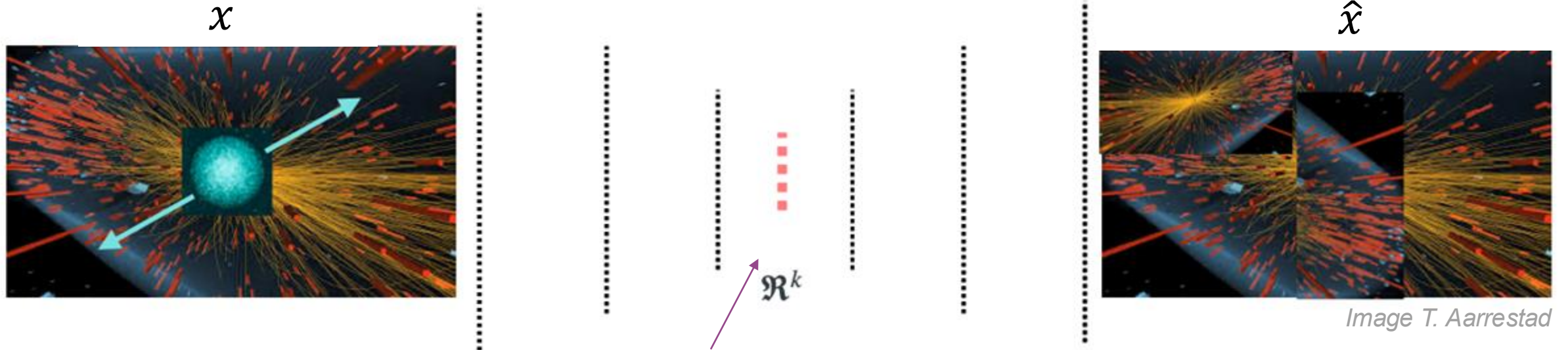


Train on randomly
sampled data

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

**Unsupervised
learning:**
 $x - \hat{x}$ represents
degree of abnormality

Anomaly Detection with Autoencoders



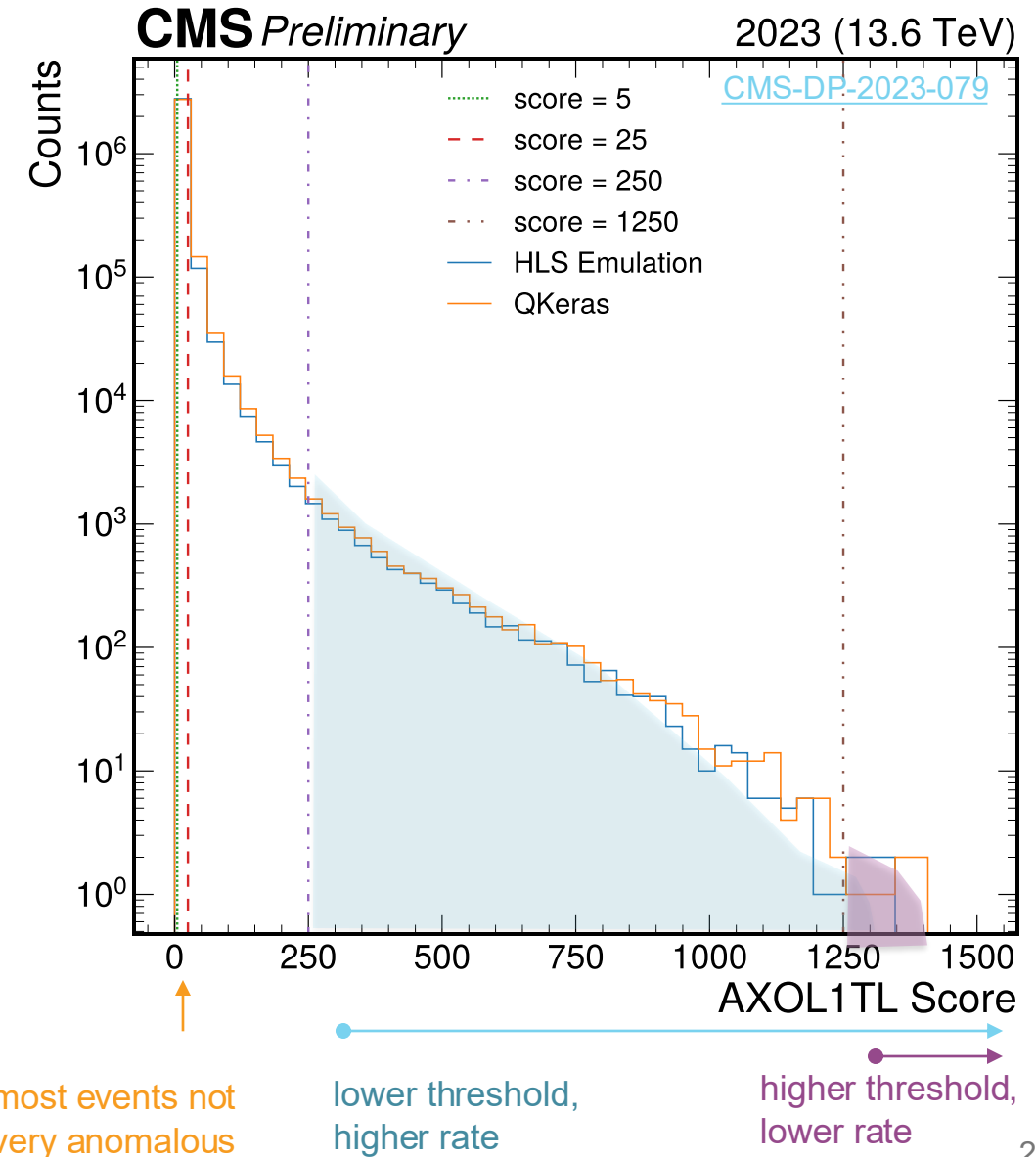
Train on randomly
sampled data

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

**Unsupervised
learning:**
 $x - \hat{x}$ represents
degree of abnormality

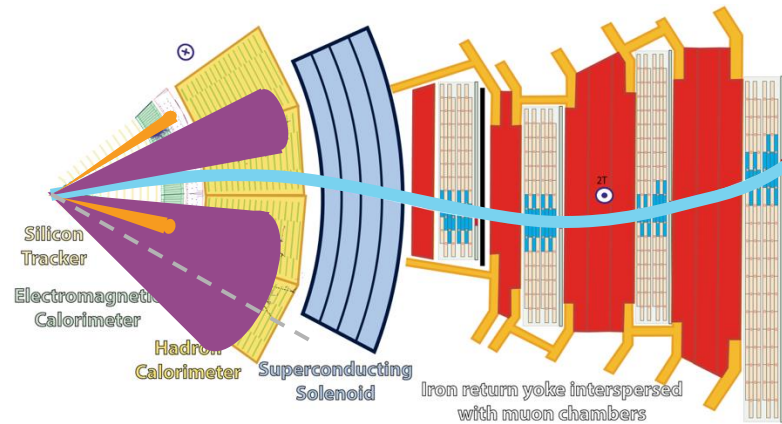
Anomaly Detection with Autoencoders

Different anomaly score thresholds are used to target different trigger rates



What are the inputs?

AXOL1TL



L1 trigger objects are inputs:

MET - (p_T , ϕ)

Up to 10 jets - (p_T , η , ϕ)

Up to 4 muons - (p_T , η , ϕ)

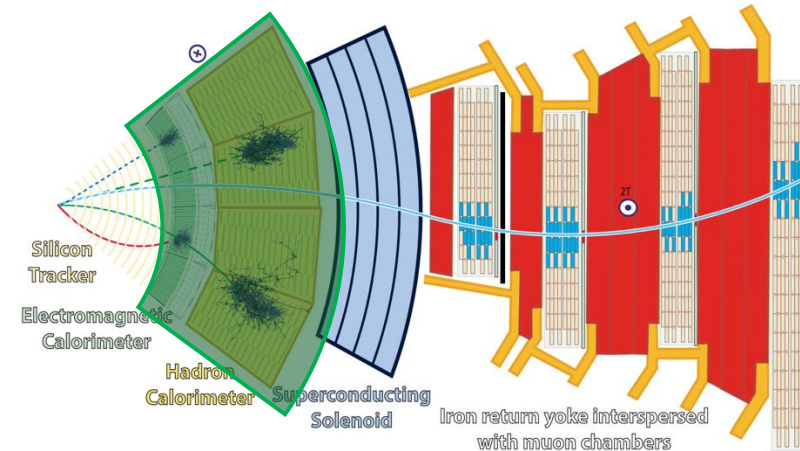
Up to 4 electrons / photons - (p_T , η , ϕ)

56 input variables total



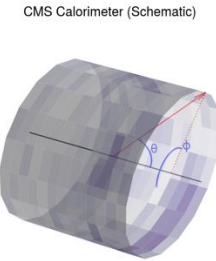
GELATO on ATLAS similar inputs

CICADA

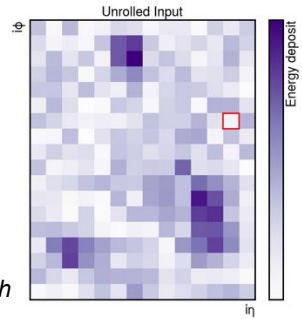


L1 calorimeter towers are inputs:

252 E_T deposits corresponding
to 14×18 towers in $\eta \times \phi$



CICADA image by L. Gerlach

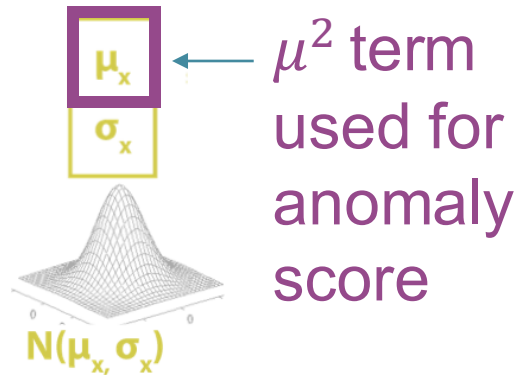
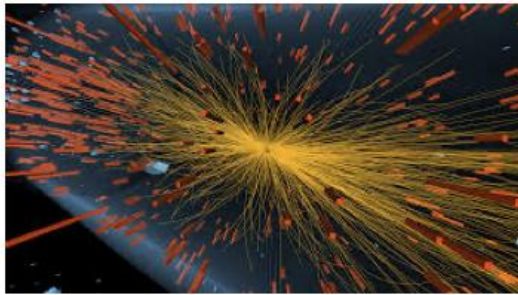


NOMAD on ATLAS uses only muon inputs

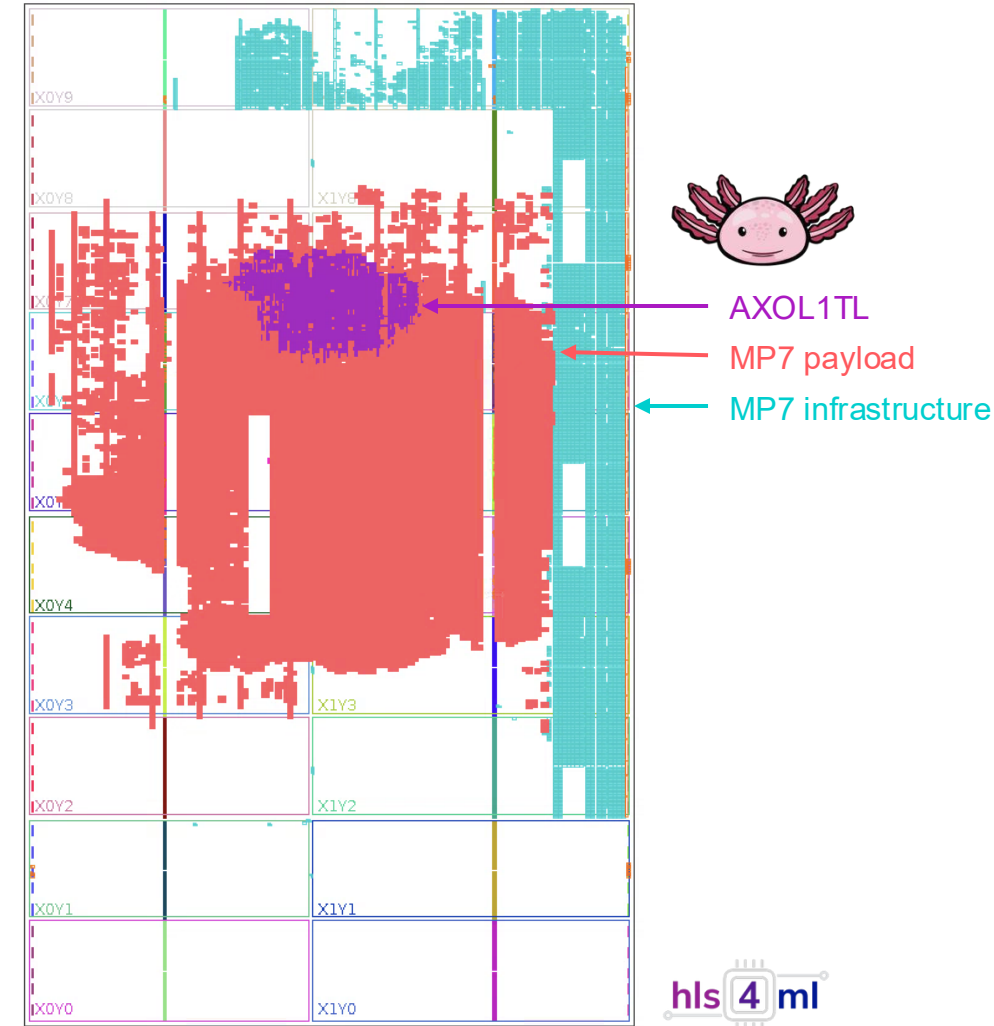


AXOL1TL Implementation

- Only deploy encoder half of the network, compute degree of abnormality directly from latent space
- Halves the network size and latency



Implemented on Xilinx Virtex-7 FPGA
50ns latency and resource requirements met



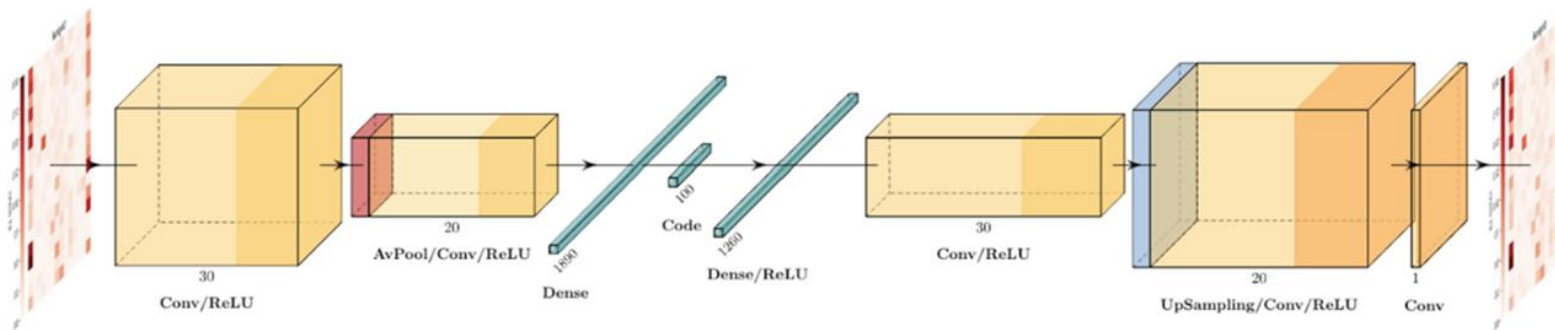
Knowledge Distillation



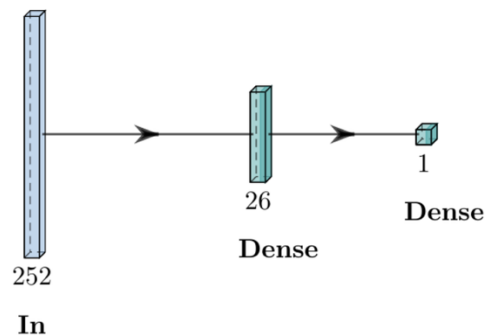
V. Sharma PPC 2024
CMS-DP-2024-121

The smaller student model learns from the larger teacher model.

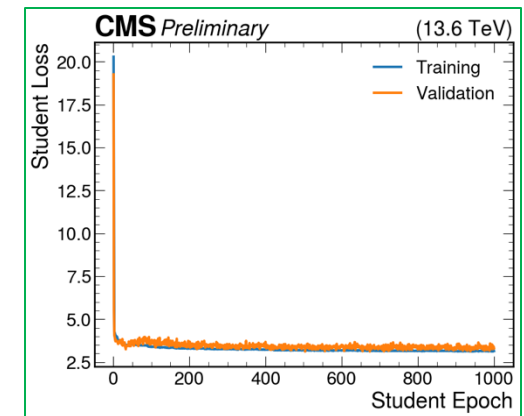
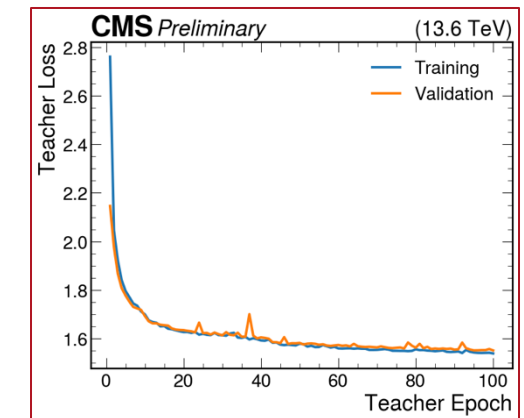
The **teacher** model



The **student** model



Inference latency ~ 100 ns
on Virtex-7 FPGA

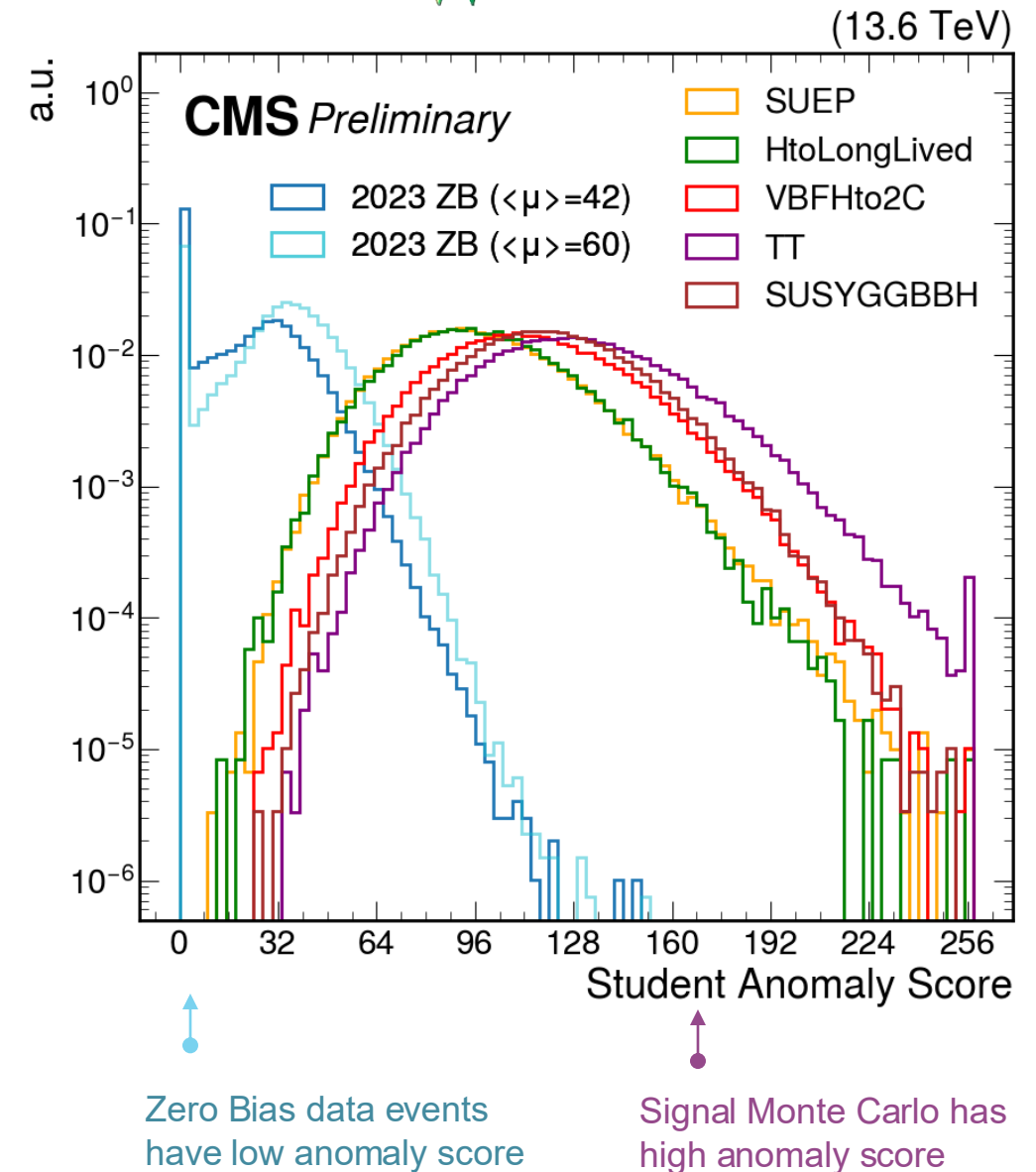


What might we be sensitive to? [CMS-DP-2024-121](#)

Nice separation between Zero Bias data and BSM signatures such as SUEPs and $H \rightarrow SS \rightarrow 4b$ ($c\tau = 900\text{mm}$)

Important caveat: Domain shift between data (training) and MC (evaluation)

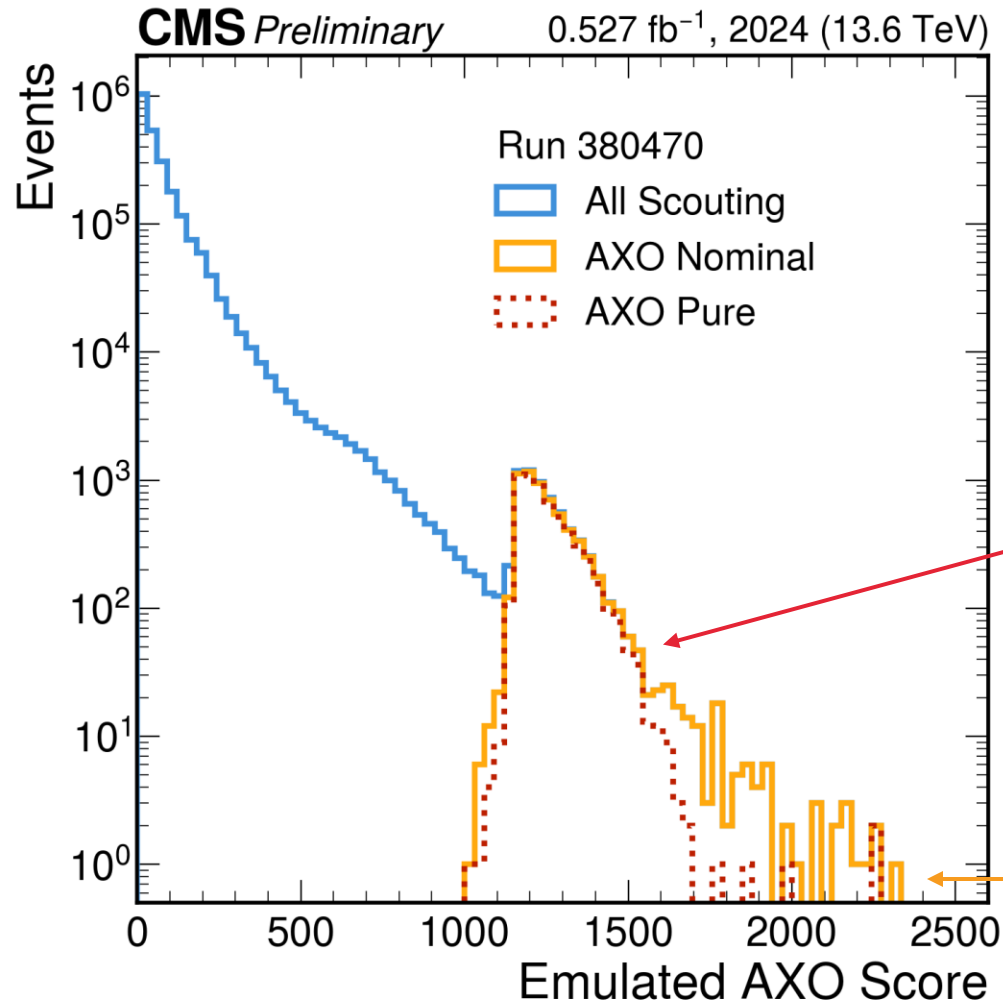
Developing techniques for studying trigger efficiencies in data and evaluating on standard candles.



Uniquely triggered events



[CMS-DP-2024-059](#)



Large fraction of unique events recorded that would otherwise be rejected

High anomaly score events, also triggered by existing L1 trigger

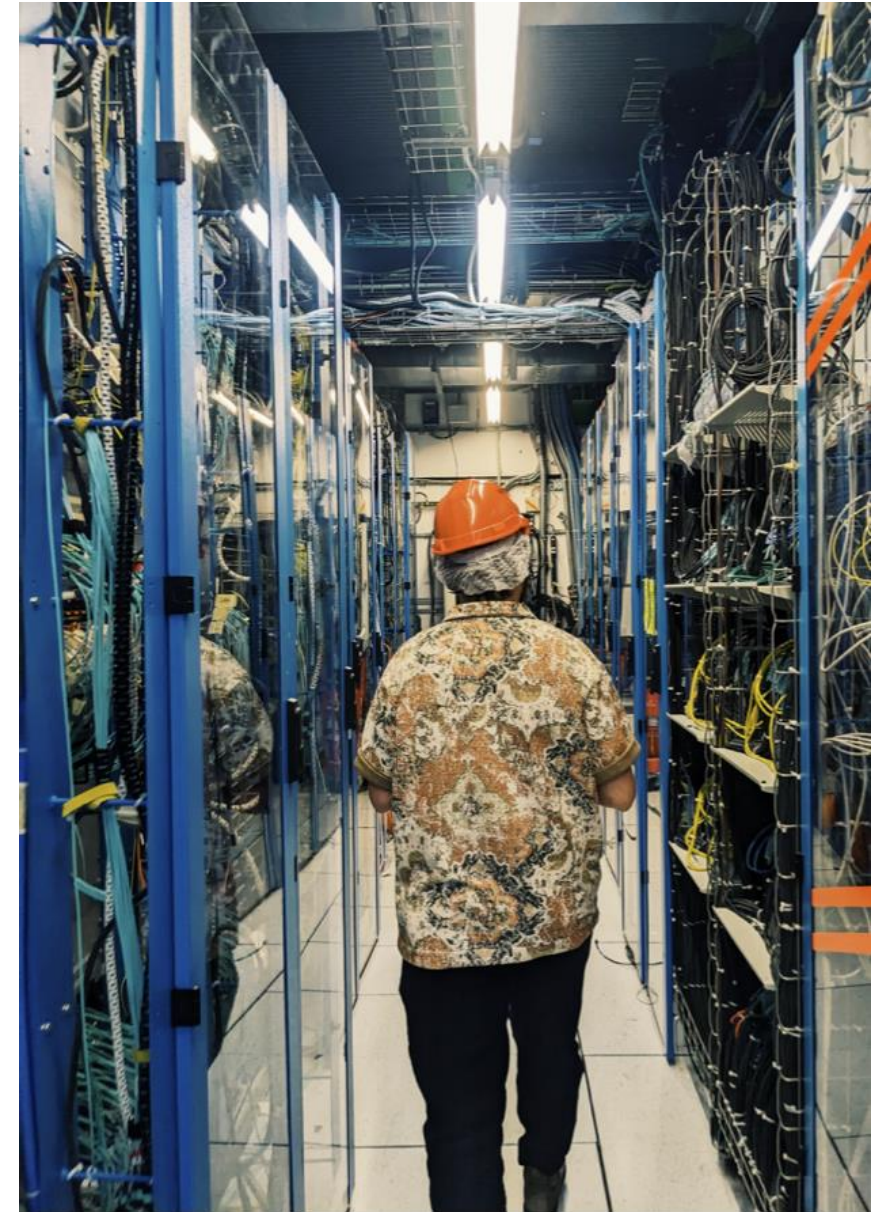
Short-term Outlook:

Anomaly detection triggers are currently taking data at L1 in both CMS and ATLAS! *First time unsupervised ML models are running in the trigger.*

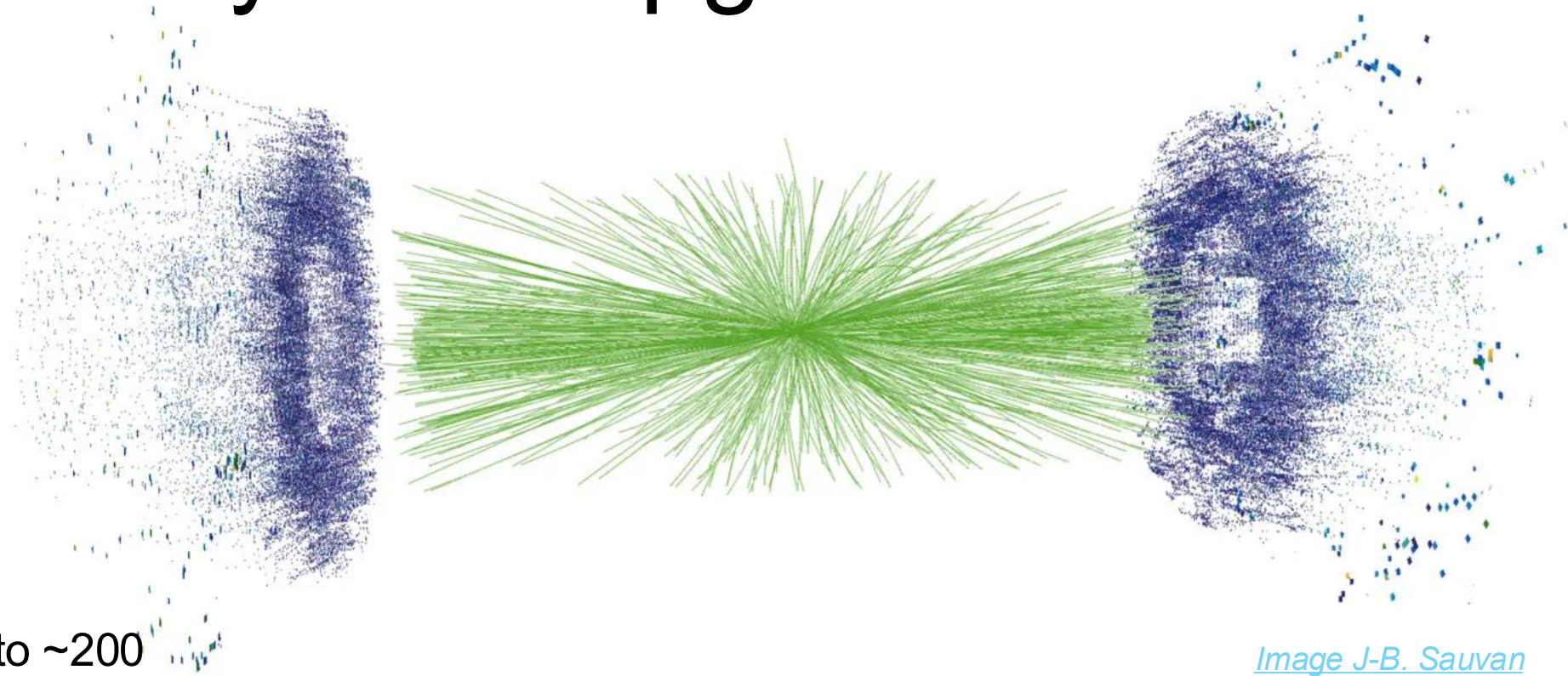
Possible developments for ML@L1T in Run 3:

- Wasserstein normalized autoencoders?
- Anomaly detection in other subsystems?
- (Semi)-Supervised L1 triggers?
- ML in L1 object reconstruction?

2026 is last year for gaining real experience with ML at L1T in preparation for HL-LHC – *we should try out as many new ideas as possible.*



High-Luminosity LHC Upgrades:



Pile-up will go from ~ 63 to ~ 200

CMS Tracker: 16x more channels

Run 3 $\sim 135,000,000$ channels

HL-LHC $> 2,200,000,000$ channels

CMS Calorimeter: 72x more channels

Run 3 $< 90,000$ readout channels

HL-LHC $> 6,500,000$ readout channels

Bunch spacing stays at 25ns (40 MHz)

HL-LHC Upgrades to the CMS L1 Trigger

- Machine Learning heavily incorporated into upgraded L1 trigger design
- Anticipate **25 billion inferences/s** from ML models at L1

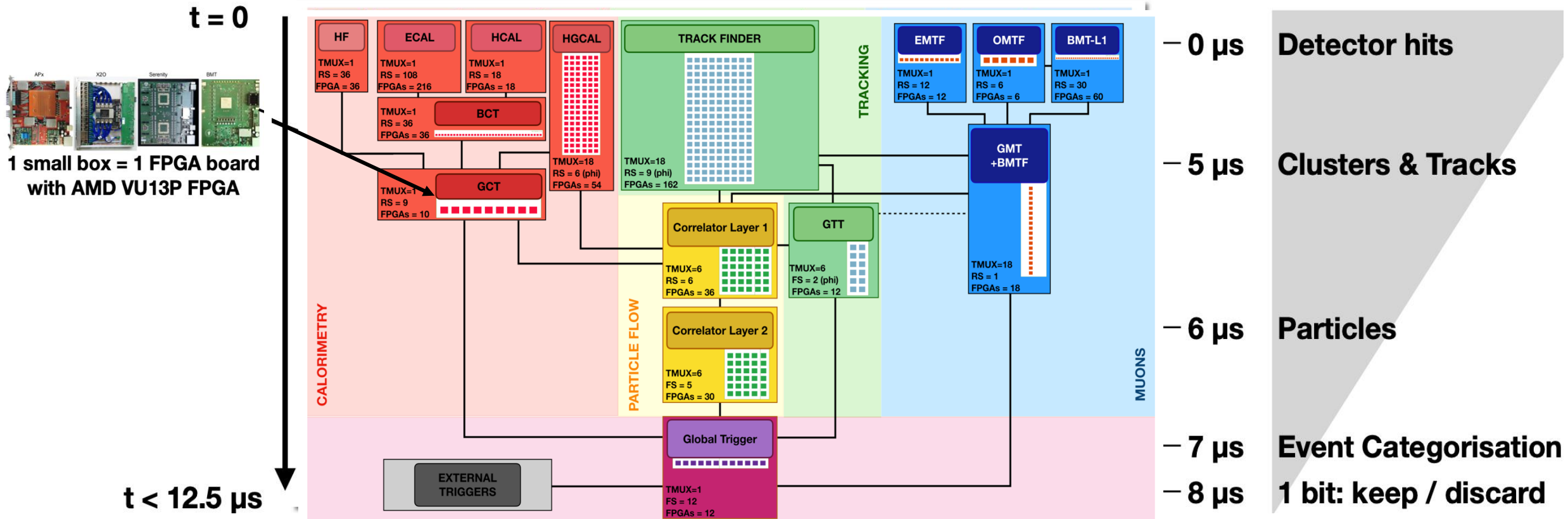
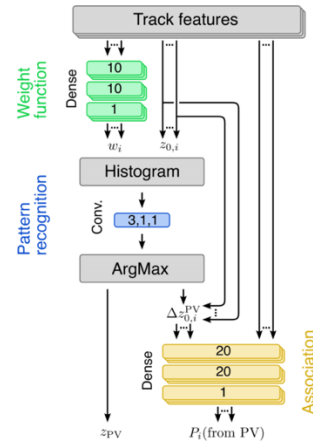


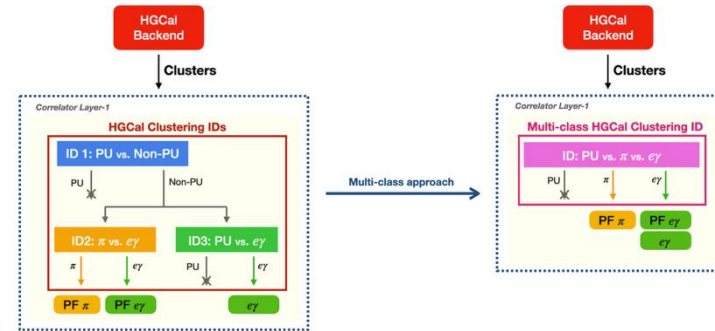
Image S. Summers

Examples of ML@L1T at CMS in HL-LHC

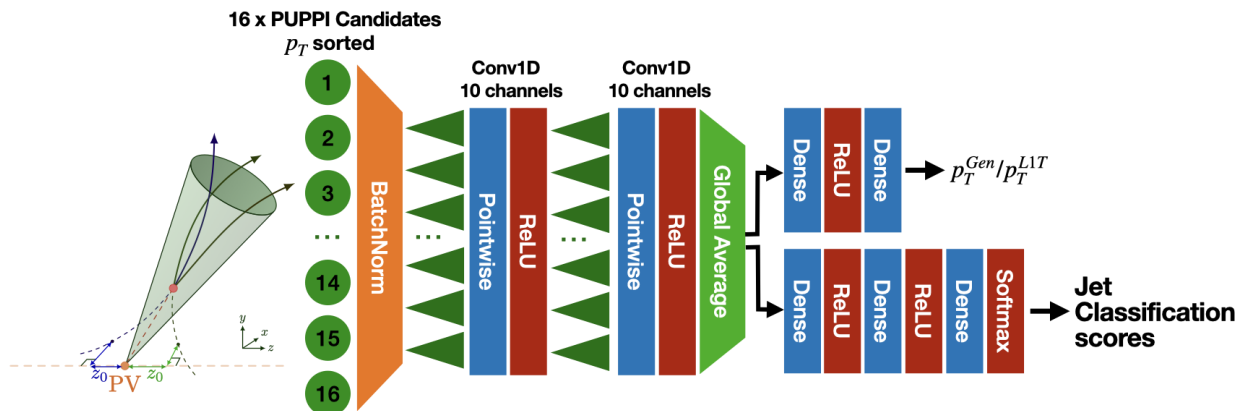
Primary Vertex Reconstruction



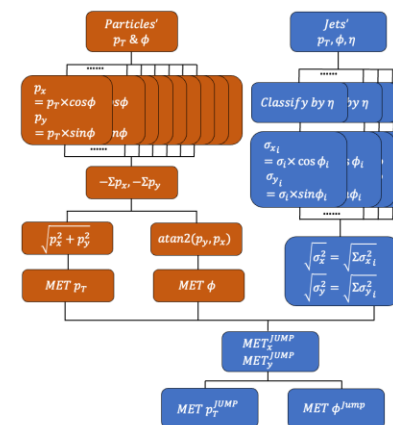
Electron ID



Jet Tagging



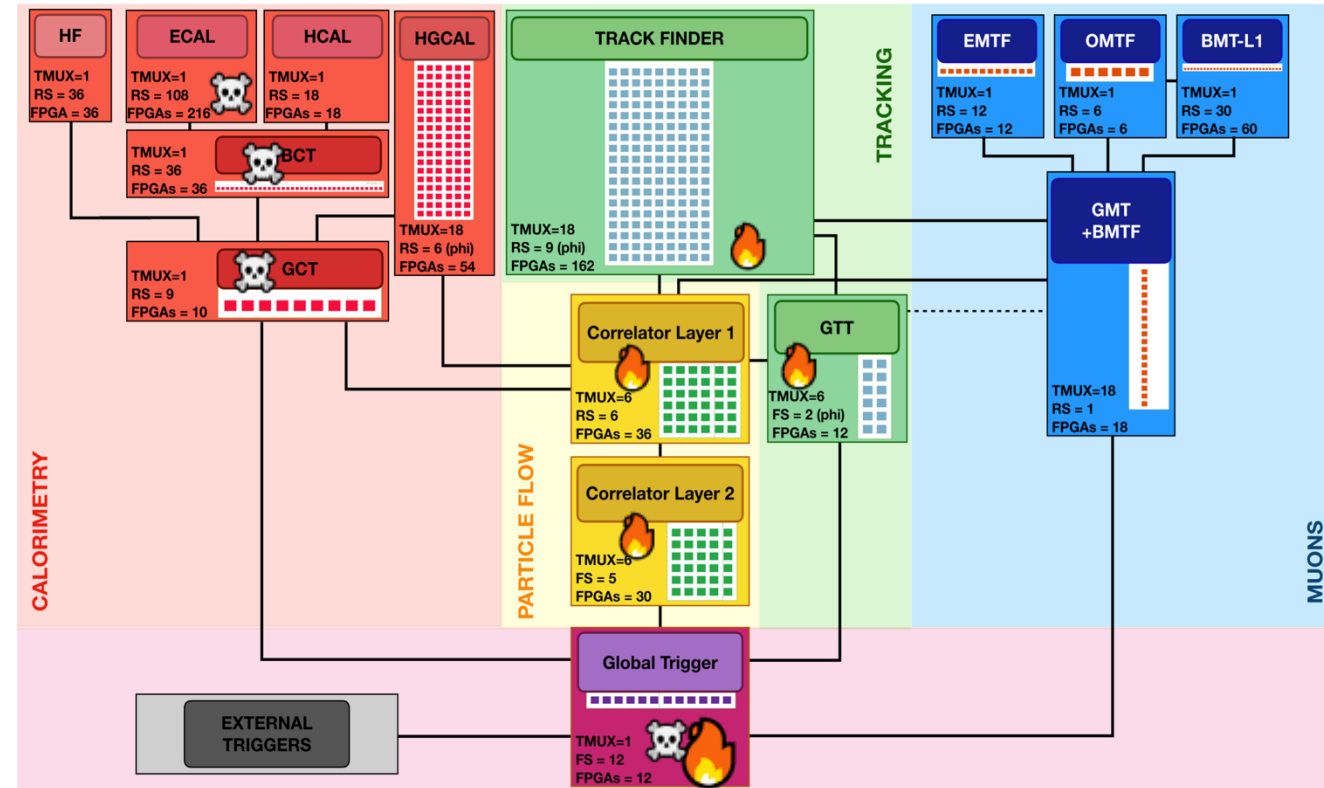
MET Reconstruction



And many, many more!

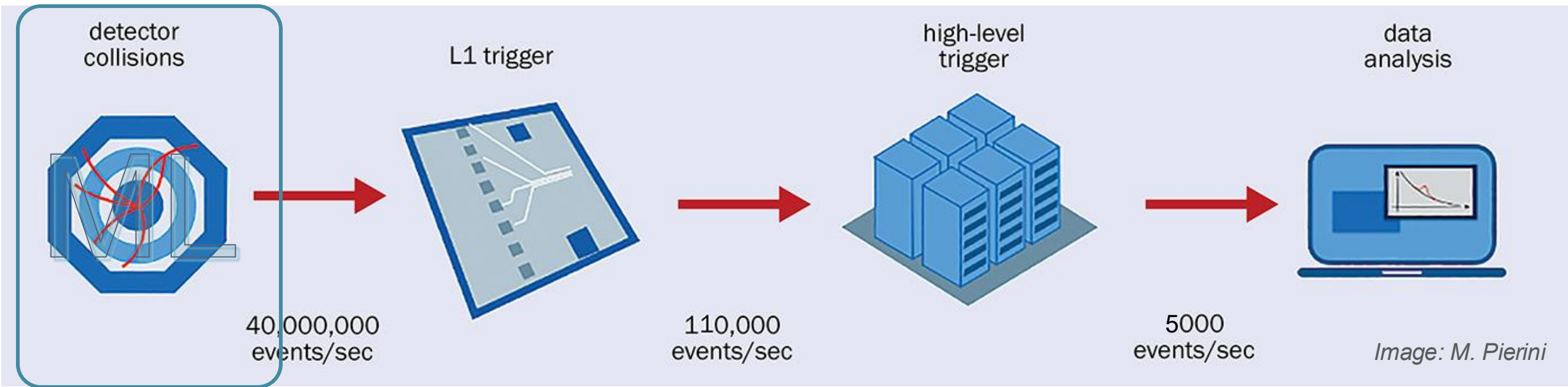
MLOps Challenges Ahead:

- Making models robust to inefficiencies or detector components not working
- Retraining models to adjust to *changing conditions* such as detector aging (continual learning)
- Cascaded ML models in the trigger
- Automating retraining pipelines
- Keeping track of deployed models

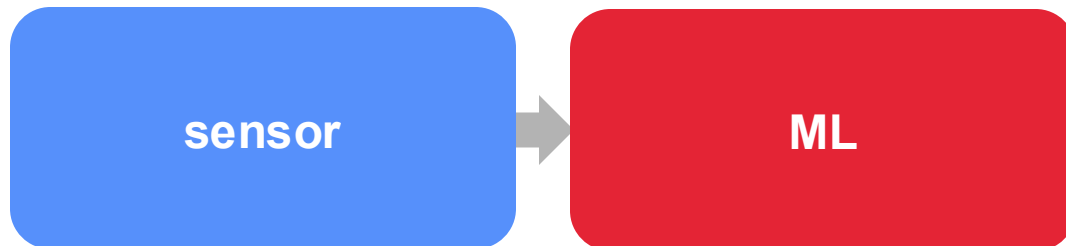


If 1% of data isn't triggered, 1% of physics is lost!

ML even closer to the detector



Can we use edge ML to build even more powerful readout systems?



Applications

Data rate reduction through:

- Early **compression** with autoencoders
 - Calorimeter data concentrator [2105.01683](#)
 - Sparsepixels [2411.01118](#)
- Early **filtering** with neural networks
 - Smartpixels [2310.02474](#), [2312.11676](#), [2406.14860](#)

Challenges

- Lower latencies
- Radiation hardness
- Power consumption
- Higher costs & specialized skills for ASIC design and fabrication
- Accurate simulation, inference and retraining campaigns

Technologies:



C++ Model

```
// defines.h - layer-precision
typedef ac_fixed<6,1,true> weight2_t;
typedef ac_fixed<16,6,true> layer2_t;
...
// w1.h - weights
weight2_t w2[1024] = {-0.1250,
-0.1875,...};
...
// myproject.cpp - network model
void myproject() {
  nnet::dense<layer2_t>{...};
  nnet::relu<layer2_t>{...};
  ...
}
```



hls4ml + Catapult AI

- Collaboration between hls4ml community and Siemens Catapult AI industry partner
- Customized hls4ml back-end
- ASICs can target even faster applications $O(10\text{ns})$ in high radiation environments

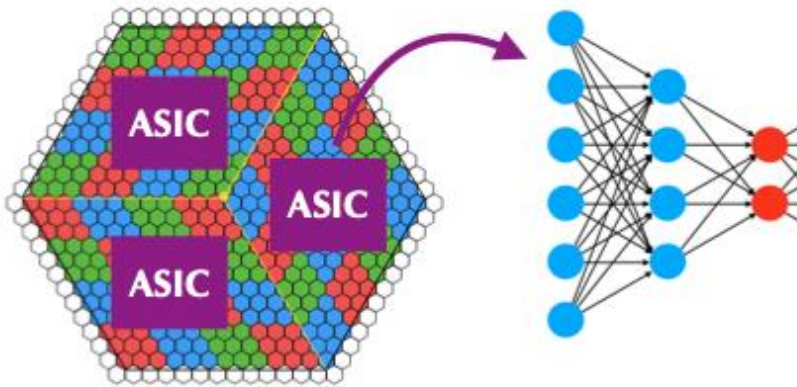


ECON-T Calorimeter Data Concentrator

[2105.01683](#)

Compress the data coming from the 6 million high-granularity calorimeter channels with auto-encoder

Encode on detector



**Transmit latent space
vector to L1 trigger**



Decode off detector

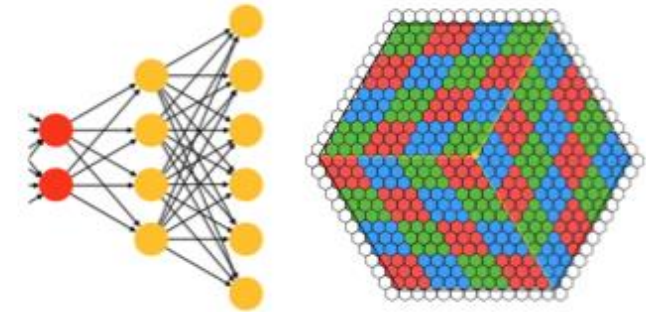


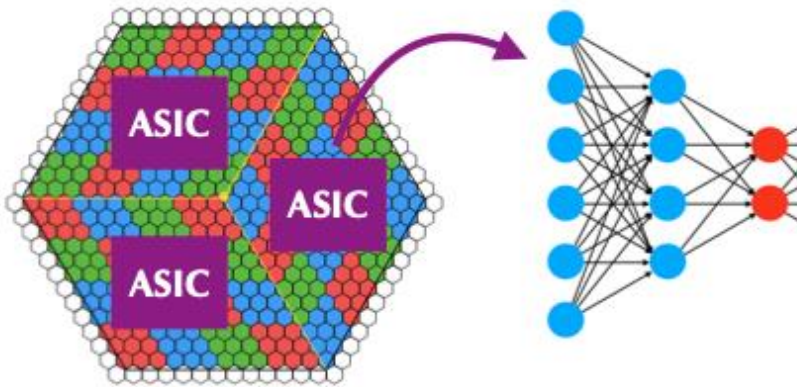
Image adapted from: J. Ngadiuba

ECON-T Calorimeter Data Concentrator

[2105.01683](#)

Compress the data coming from the 6 million high-granularity calorimeter channels with auto-encoder

Encode on detector



Transmit latent space vector to L1 trigger



Decode off detector

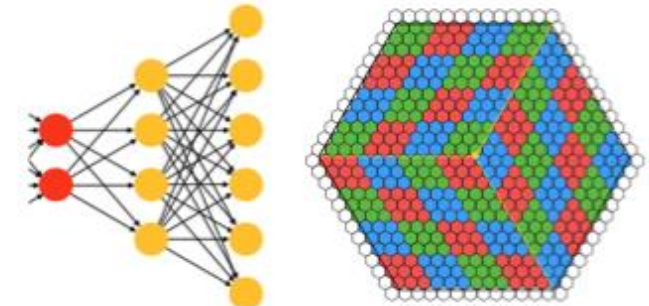
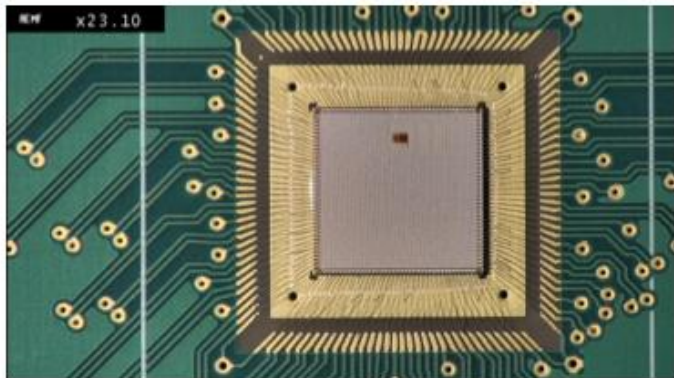


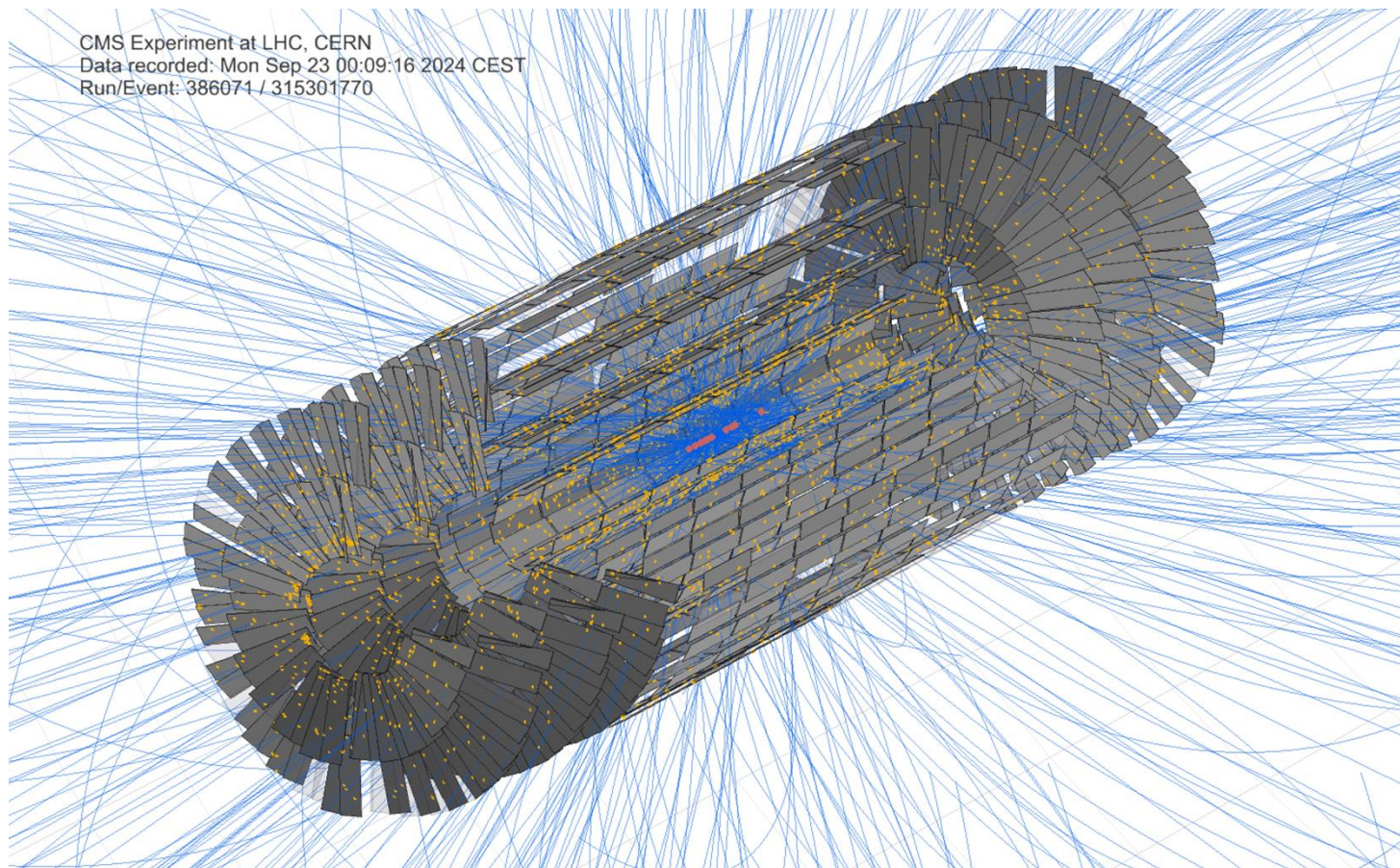
Image adapted from: J. Ngadiuba

- Implemented in ECON-T 65nm CMOS ASIC
- Fixed NN architecture with re-programmable weights and biases through I²C
- Weights and biases triplicated for radiation tolerance
- Low power (must stay at -30°C), 75-100mW, 1.5us latency

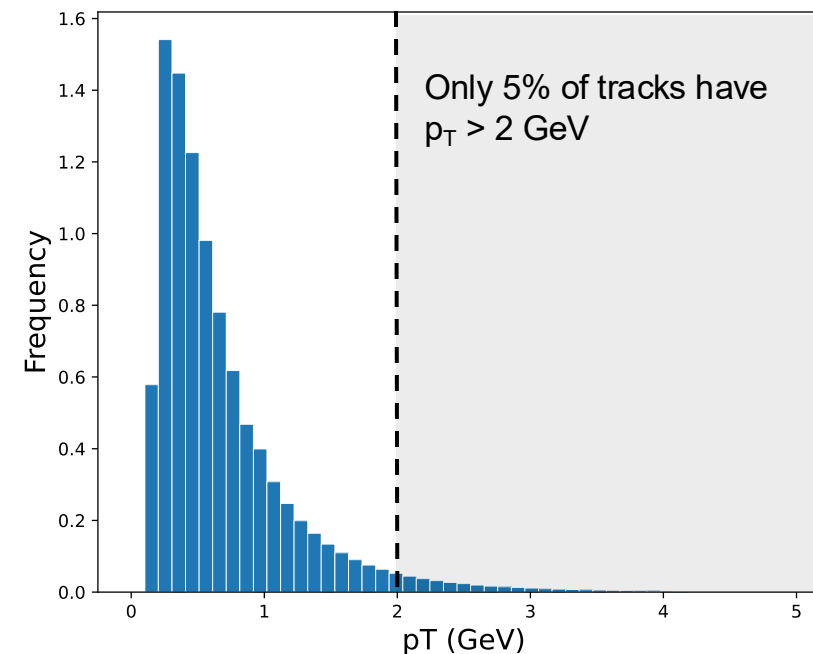
See [Zachary's talk](#) today!



Track Reconstruction Challenges



High pile-up
→ large amount of combinatorics

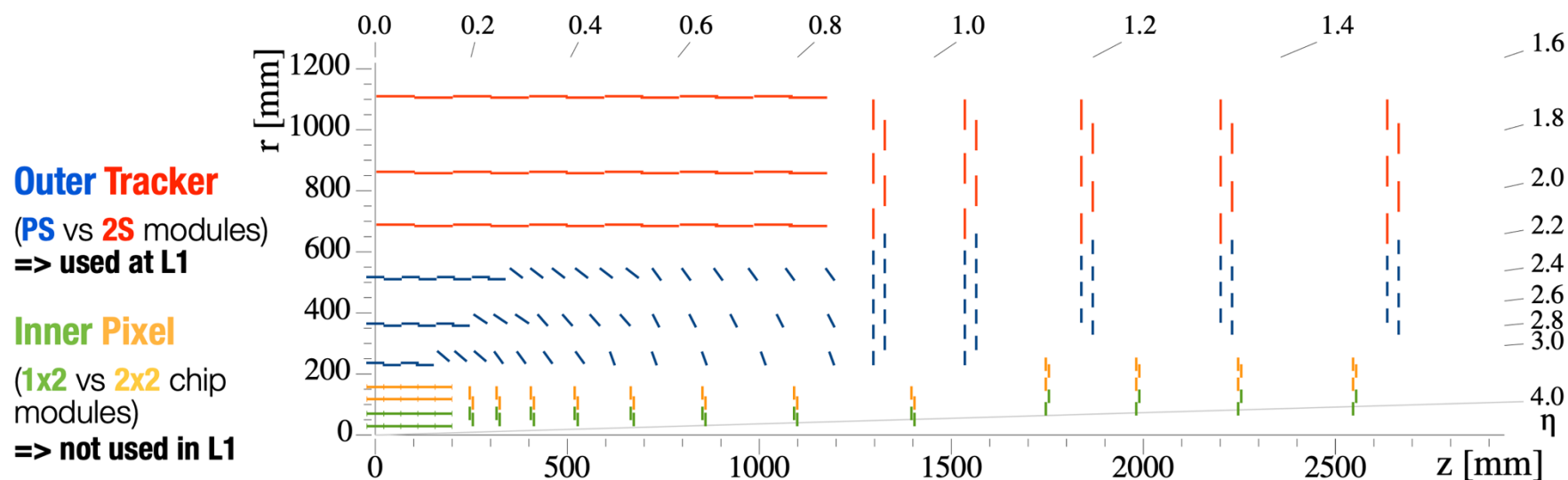
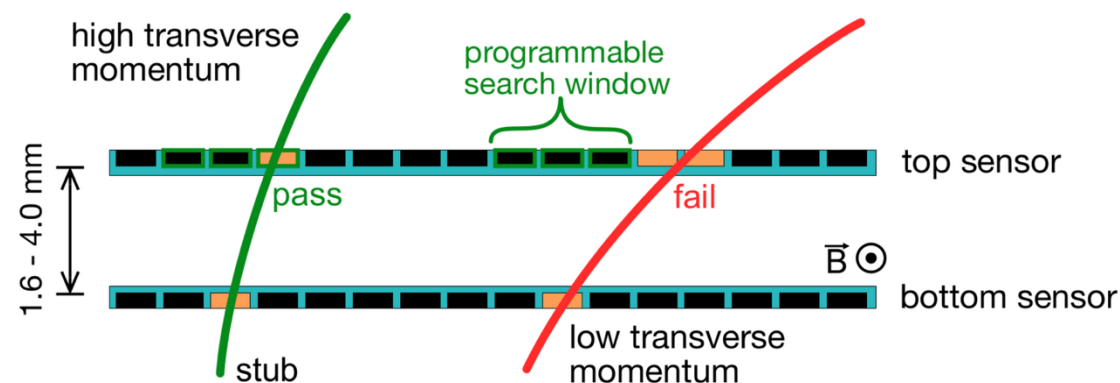


Momentum distribution of tracks reconstructed by the CMS detector during Run 2 data taking [2310.02474](#)

Momentum filtering in outer tracker

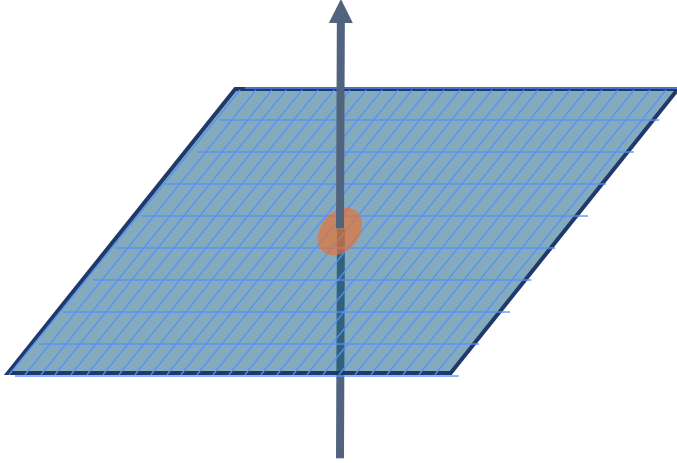
CMS double layer design choice in outer tracker allows for early filtering of low momentum tracks.

Key to CMS L1 track trigger!

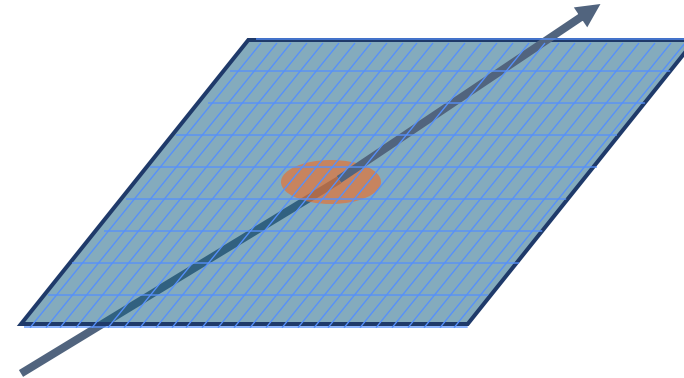


Can we do a similar thing in a single layer of silicon?

Concept behind smartpixels



Charged particle path perpendicular to sensor:
Regular charge cluster shape

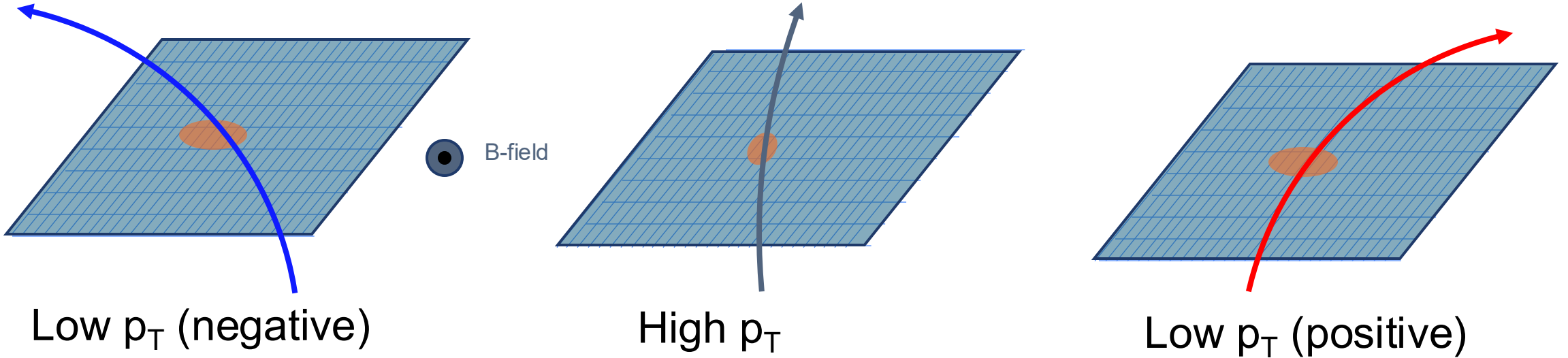


Charged particle path at angle to sensor:
Smeared charge cluster shape

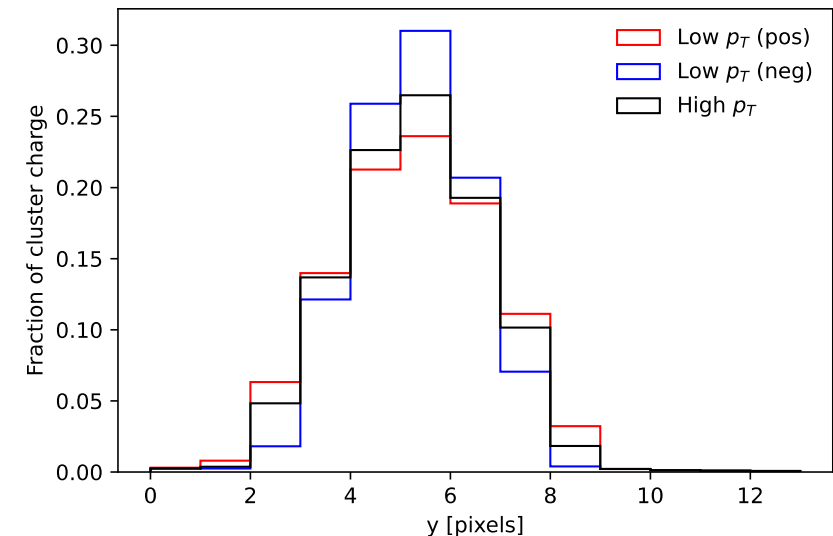
- Use cluster shape to extract incident angle of particle traversing pixel sensors
- **By filtering on track momentum, reduce the data volume at the source,** lowering both rates and power consumption

Concept behind smartpixels

[Yoo et al 2024 Mach. Learn.: Sci. Technol. 5 035047](#)



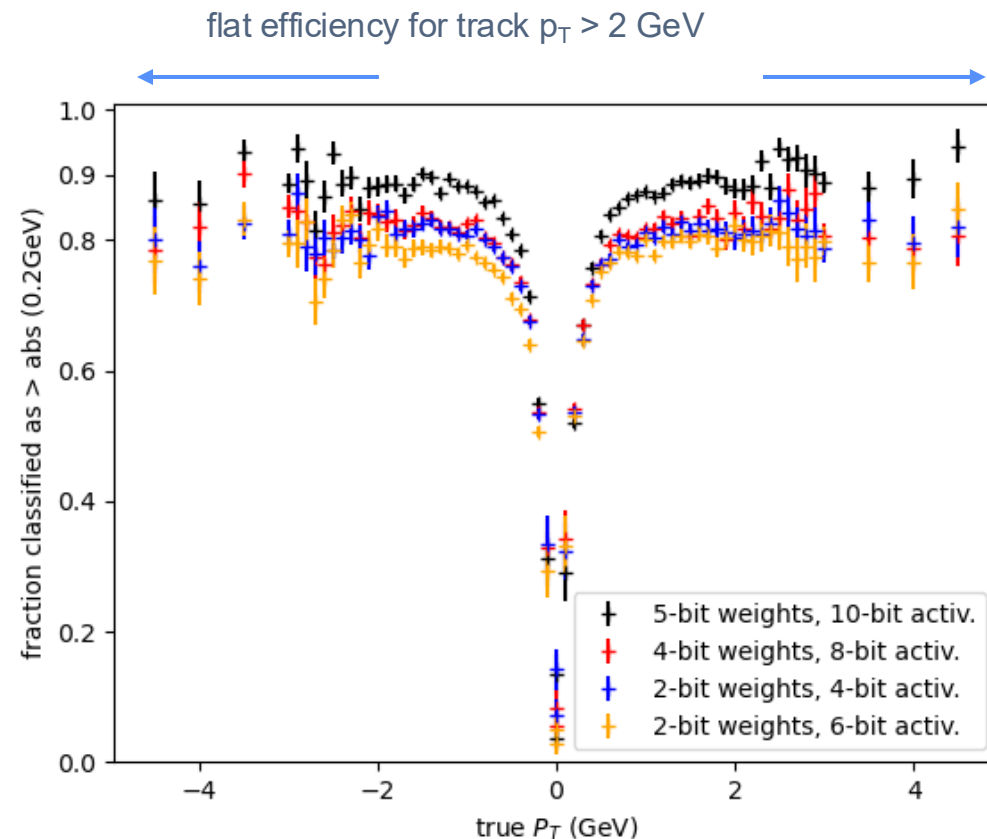
- Can use a neural network in sensor readout to distinguish low p_T from high p_T charged particles
- Lorentz drift shifts cluster charge distribution



How well does it work?

[Yoo et al 2024 Mach. Learn.: Sci. Technol. 5 035047](#)

- Simple neural network quite effective (1 layer - 128 neurons, 2,307 parameters)
- **Can reduce data rate by 54.4% - 75.4%**
- Expected power consumption 300 $\mu\text{W}/\text{cm}^2$
- Expected latency 3.9 ns
- Used hls4ml + Catapult AI to place momentum filtering NN on ASIC with programmable weights
- Developing more sophisticated architectures for regression too!
- *An example of ML not just in the trigger, but embedded directly on sensor readout*



Chip tape-out and testing:

Prototype 1.5mm² ASIC with momentum filtering NN in 28nm CMOS has been fabricated*

Tests of the bare chip are currently in progress

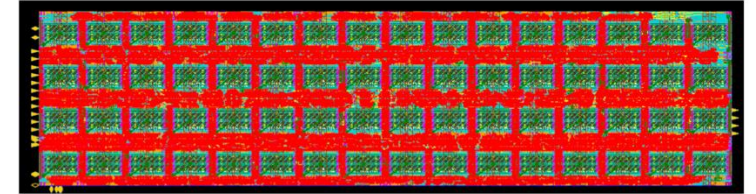
Next steps after testing the prototype:

Build a bigger chip to bump bond to sensor & test in a testbeam

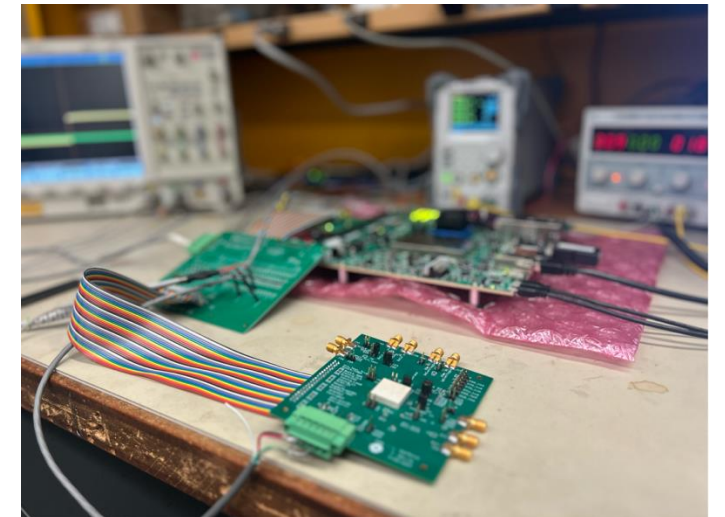
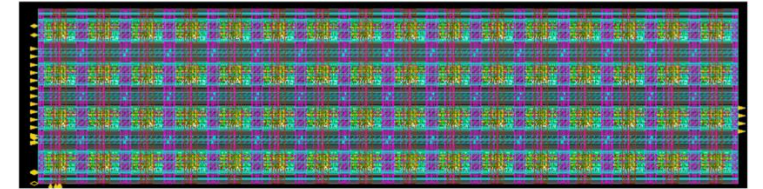
*for characterizations of radiation hardness of 28nm, see [G. Borghello, TWEPP 2023](#)

[2406.14860](#)

Red = classifier algorithm



Floorplan with analog pixels with power and bias grid



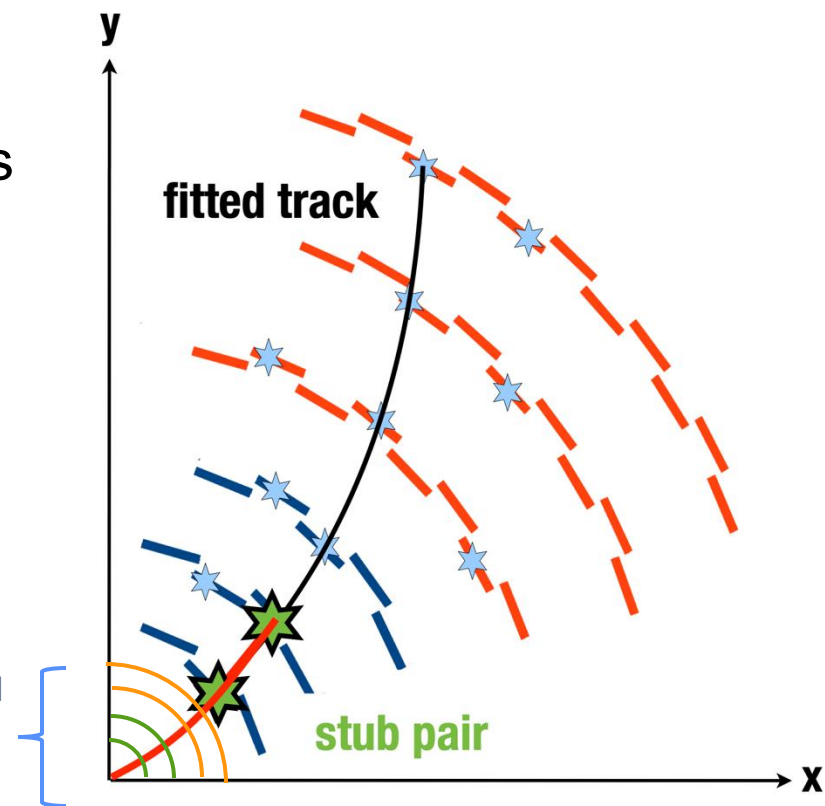
Next steps <https://fastmachinelearning.org/smart-pixels>

Continued model development for momentum regression

Studies examining technical feasibility and physics outcomes of integrating within CMS L1 track trigger for Phase III (~2035)

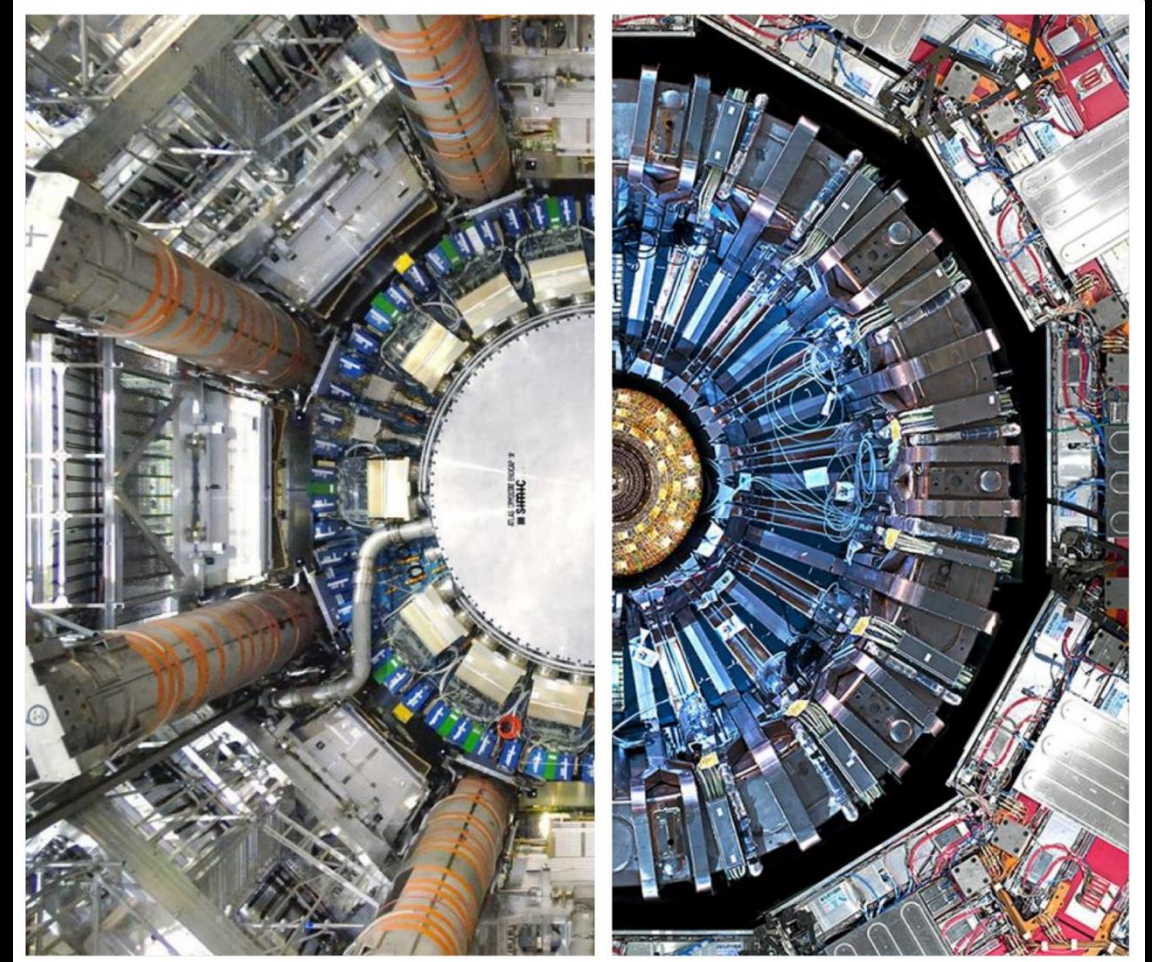
Applications in future colliders, e.g. rejecting beam-induced-backgrounds at muon colliders.

Pixel layers currently unused in CMS L1 track trigger upgrade



Conclusions

- LHC experiments have high data rates and strict latency requirements
- ML applications on FPGAs & ASICs are rapidly growing at the LHC
- ML is *currently* being deployed within the CMS & ATLAS L1 trigger systems
- On-sensor ML has applications for the HL-LHC and future colliders
- These new technologies are expanding our potential for future discoveries

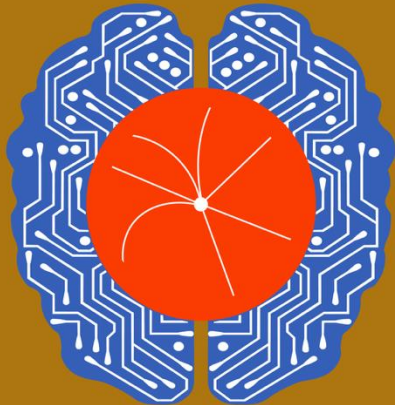


Thank you!



1-5 September 2025
**fast machine learning
for science**

Real-time
and
accelerated
ML
for
fundamental
sciences



indi.to/fastml25

ETH zürich

