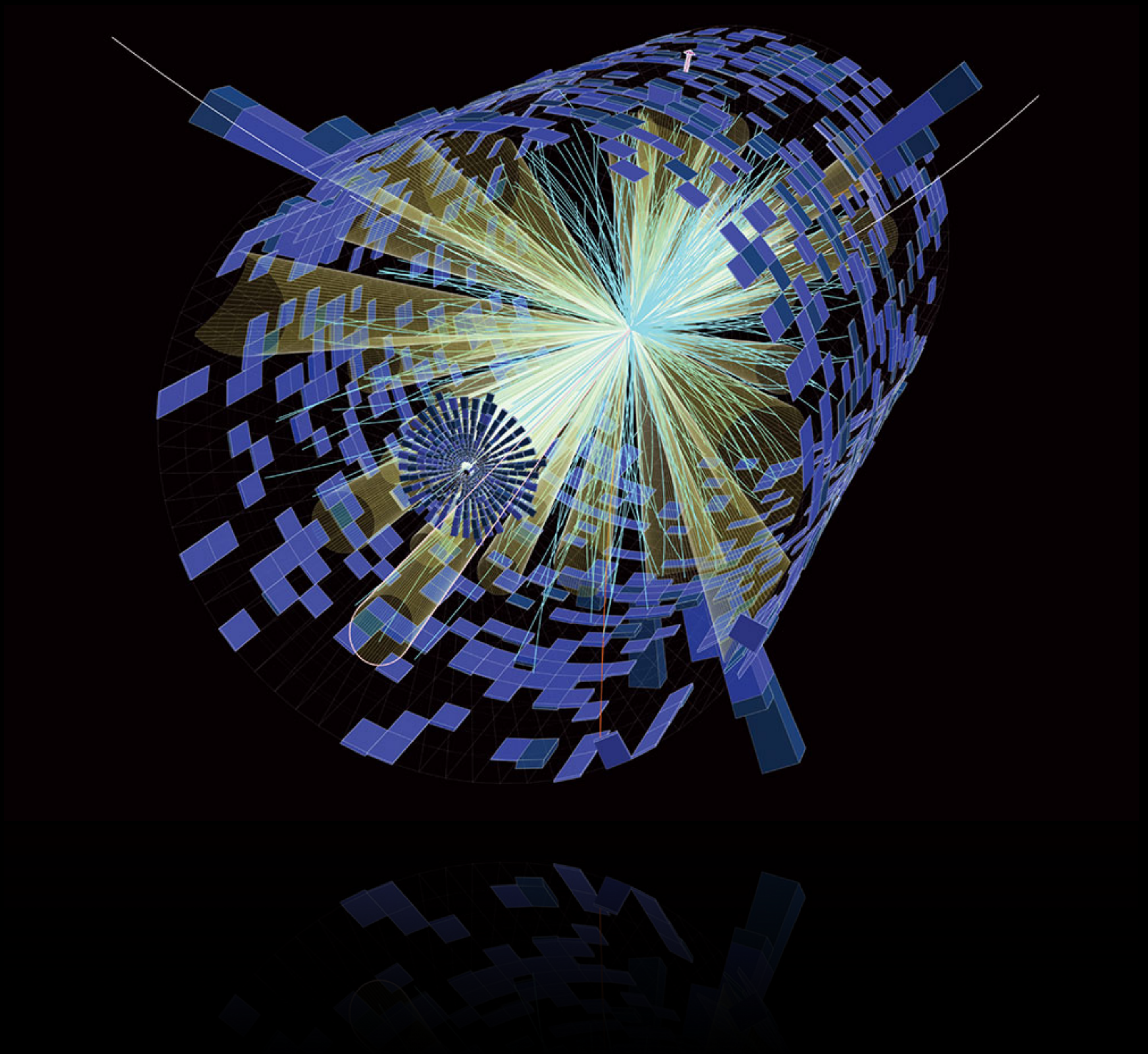
AI at Colliders: From Real-Time Decision Making to Data-Driven Discovery

Jennifer Ngadiuba (Fermilab)

EuCAIfCon 2025
Cagliari, Italy
June 16–20, 2025



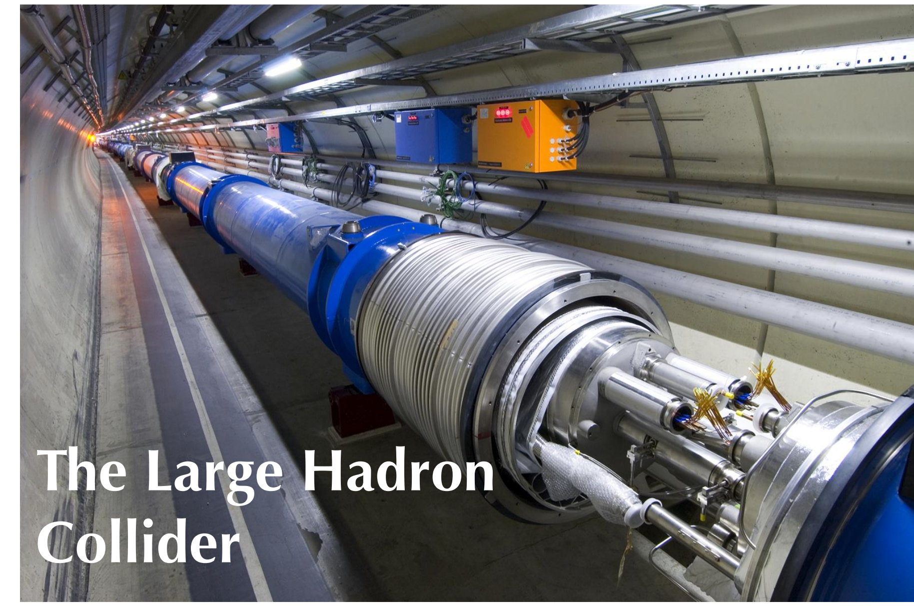
FastML Lab

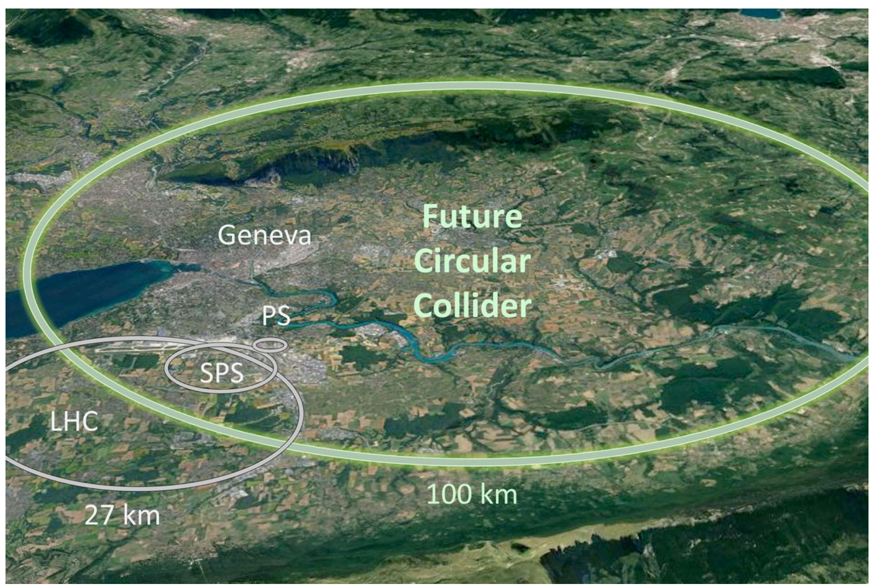
NexTGen
Next Generation Triggers

Fermilab

Big Science in 21st century

Probing the **fundamental structure of nature** requires complex experimental devices, large infrastructures and big collaborations.


The Large Hadron Collider

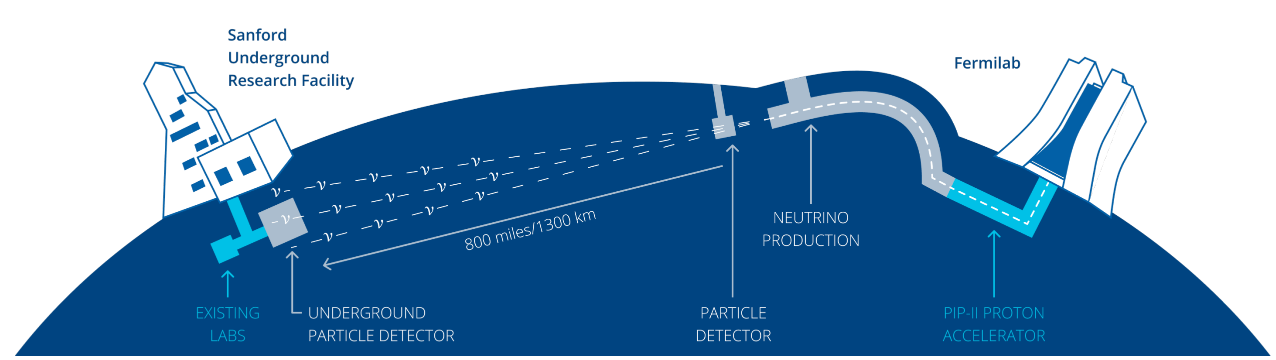


International
MUON Collider
Collaboration


LIGO/VIRGO interferometers


Vera C. Rubin Observatory

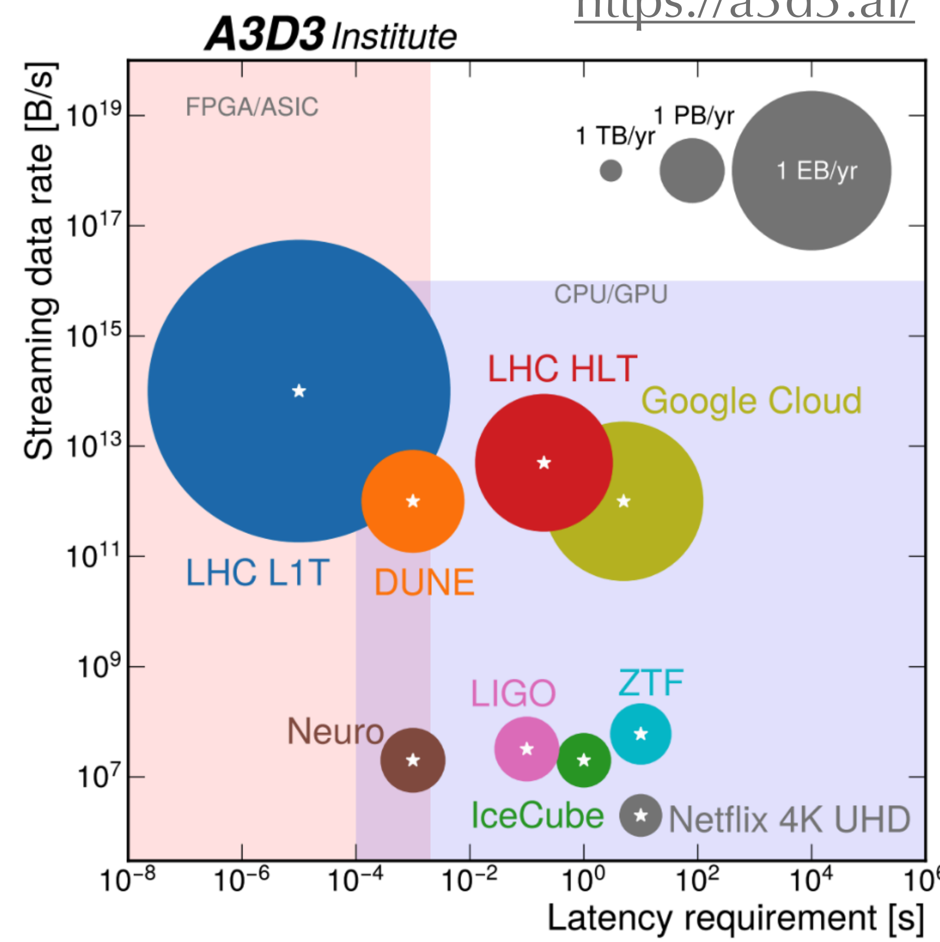
The DUNE neutrino experiment

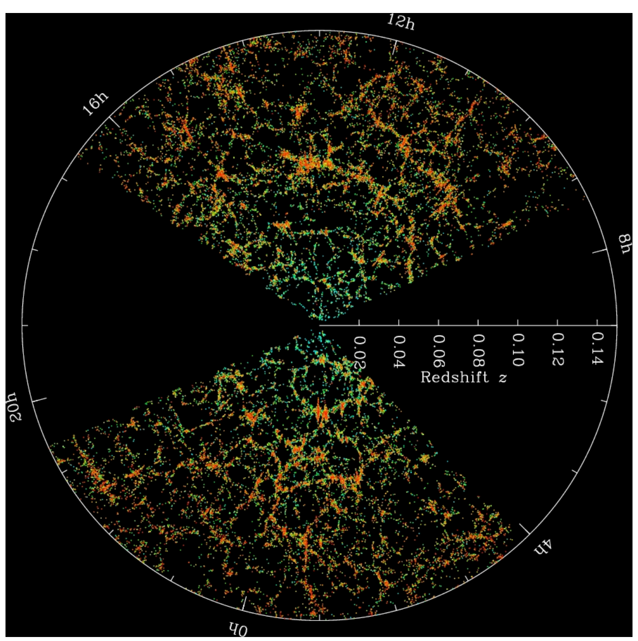


Big Science = Big Data

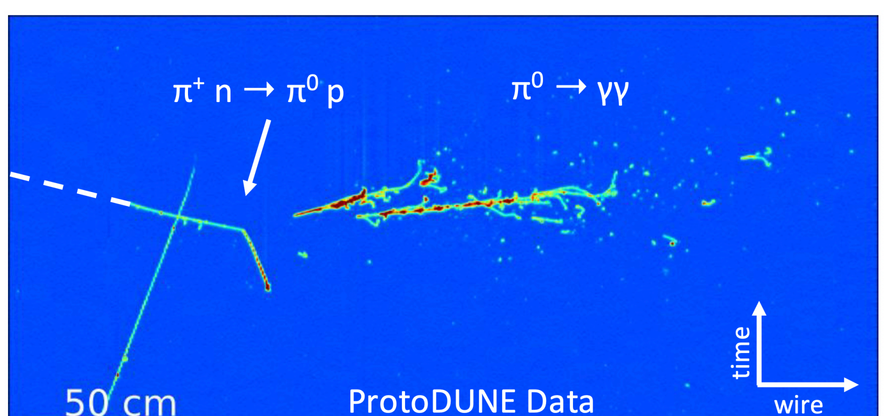
- Increasingly complex data both in **volume and dimensionality**
- Increasing need for **efficient and accurate data processing pipelines**
- Challenge in **simulating expectations** for what experiments may observe
- But also need for innovative **data & discovery driven** physics analyses approaches

https://a3d3.ai/

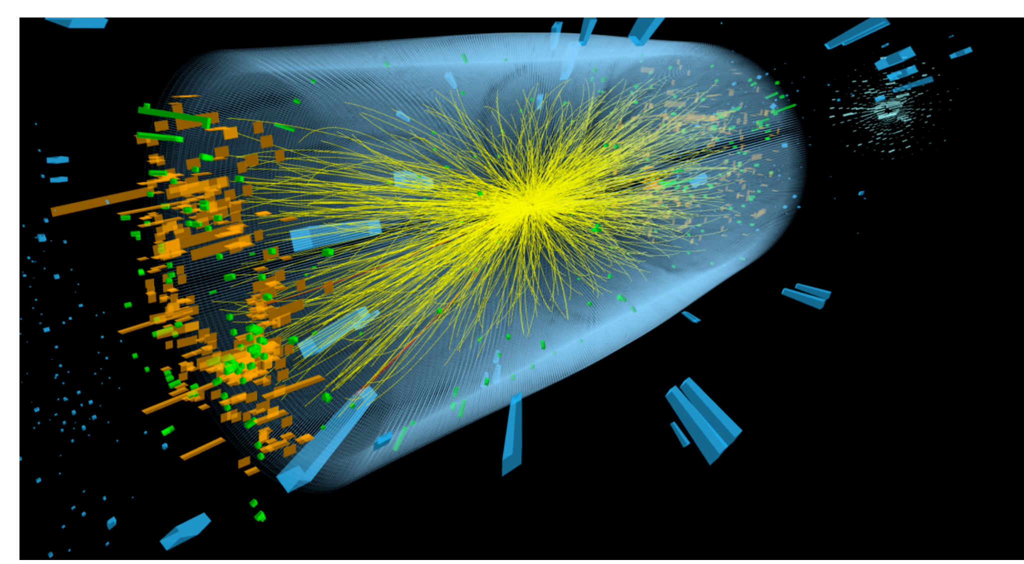




Sloan Digital Sky Survey



Interactions in LArTPC

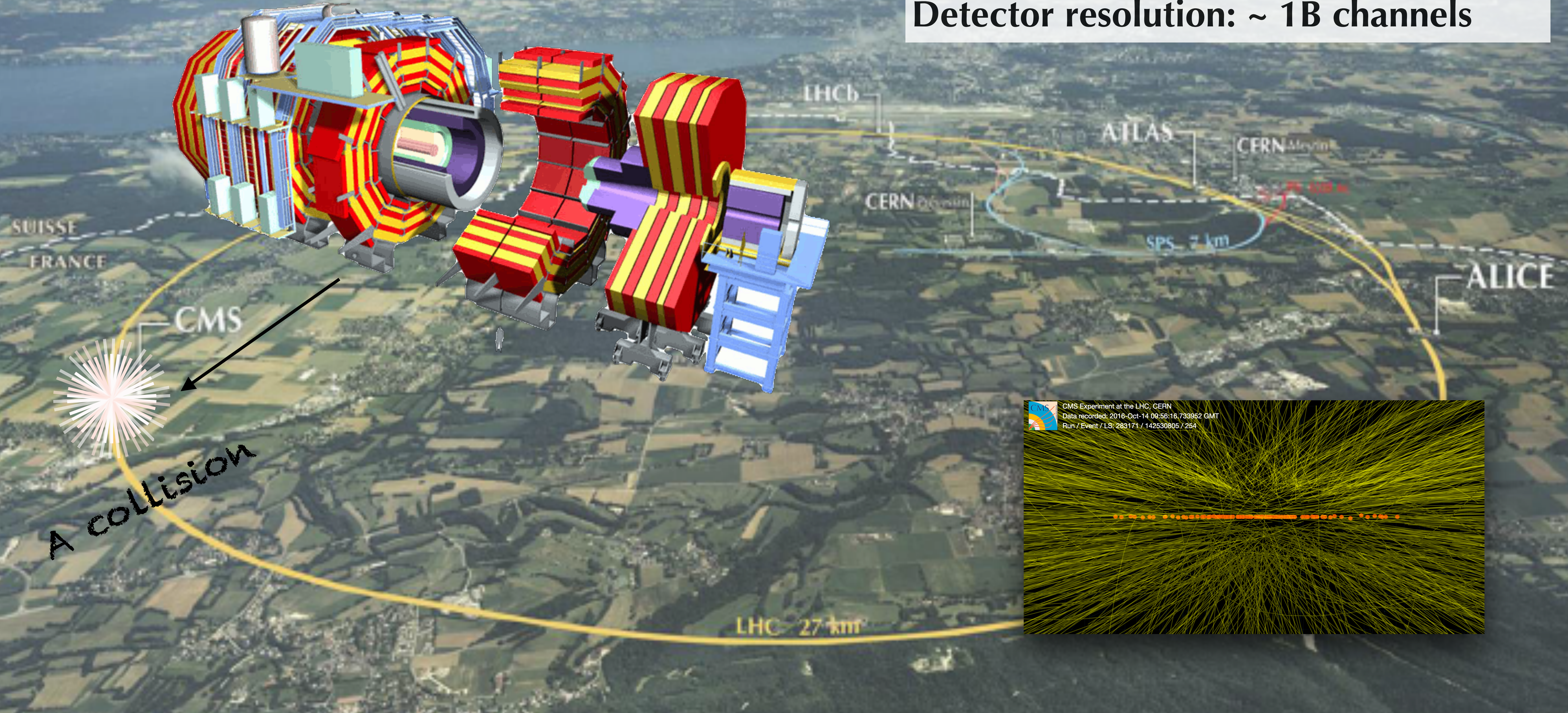


A LHC collision

Big Data @ the Energy Frontier

The Large Hadron Collider (LHC)

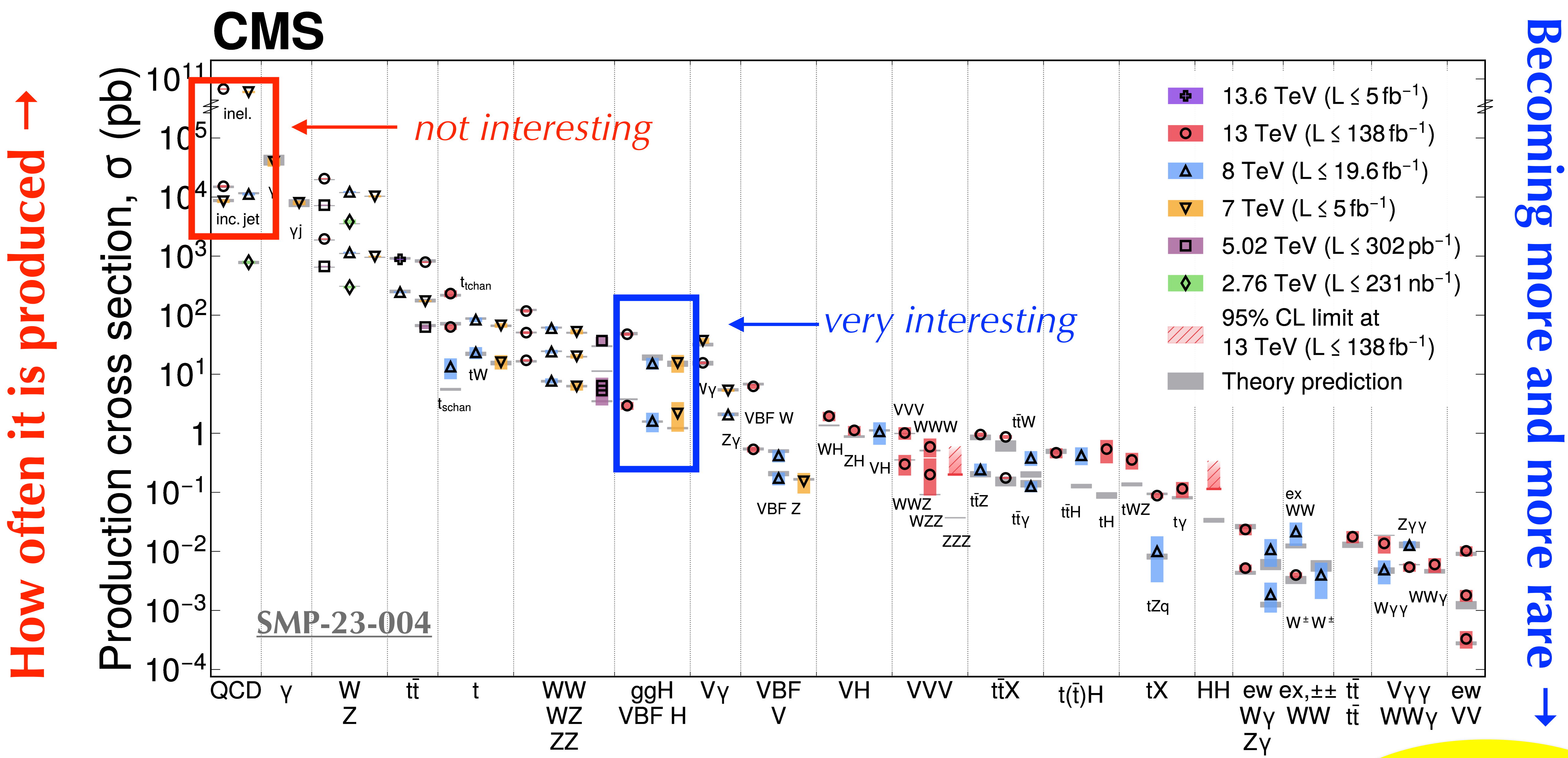
Collision frequency: 40 MHz
Particles per collision: O(10^3)
Detector resolution: ~ 1B channels



Extreme data rates of ~Pb/s!

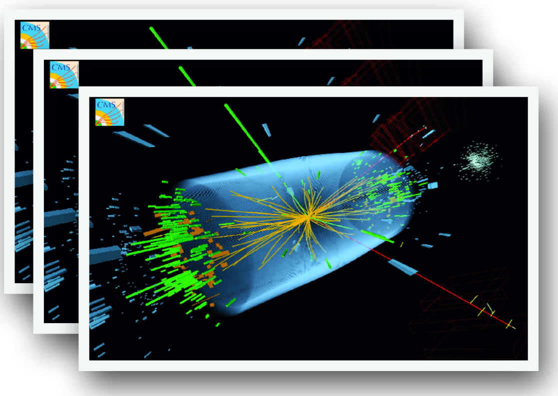
Collisions which produce interesting products (ex: Higgs boson) are typically very rare

The probability of producing a Higgs boson is 5-9 orders of magnitude smaller than producing only jets

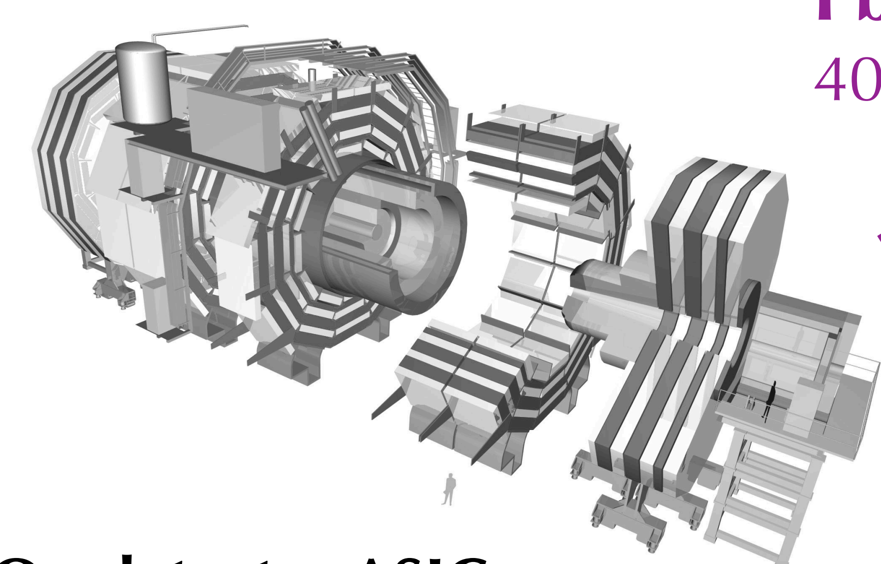


NEW PHYSICS!

Data reduction workflow @ LHC



CMS Experiment
40 MHz collision rate
~1B detector channels



On-detector ASIC compression
~100 ns latency

Pb/s
40 MHz

FPGA filter stack
~μs latency

Level-1 Trigger

10s Tb/s
100s kHz

On-prem CPU/GPU filter farm
~100 ms latency

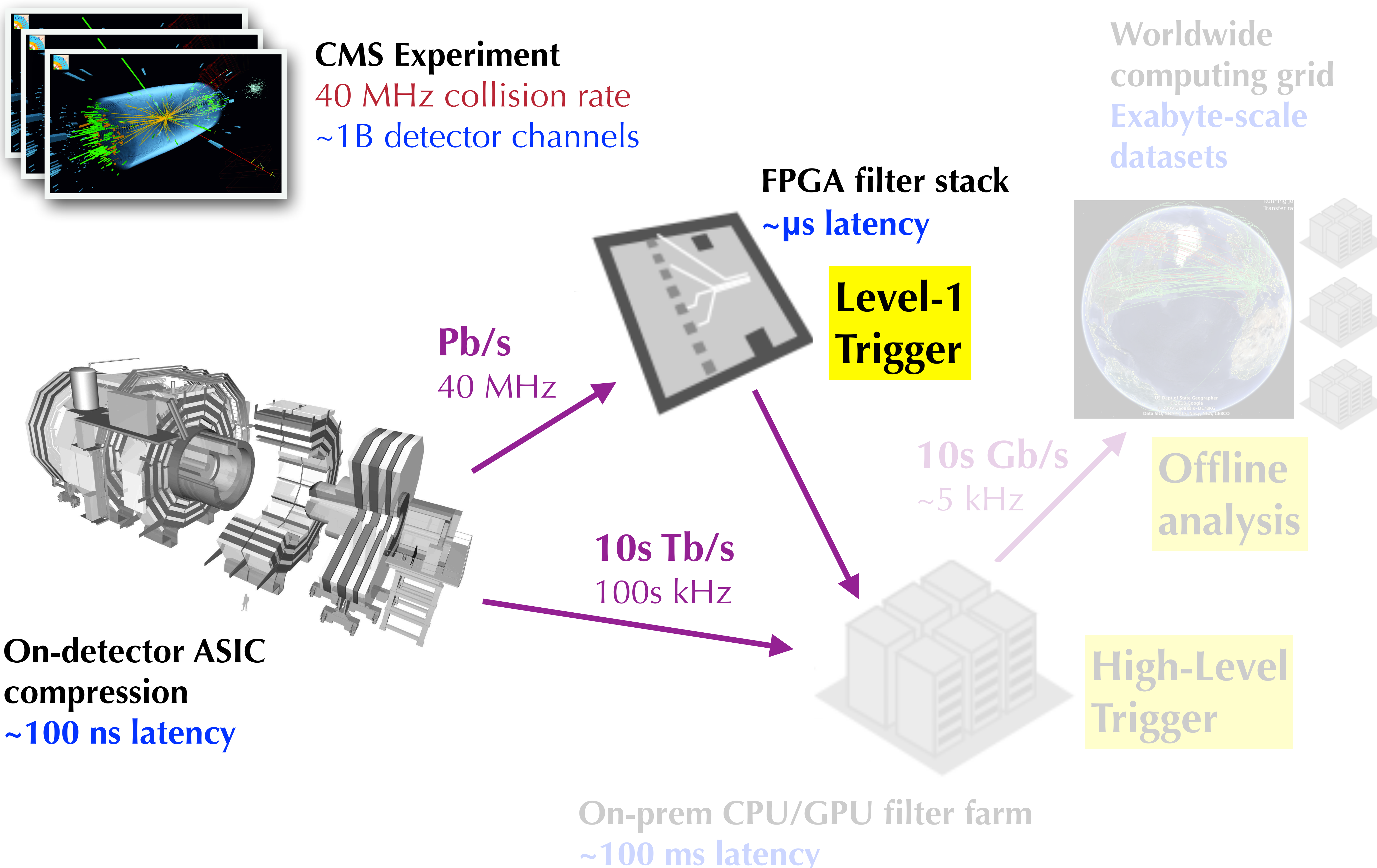
High-Level Trigger

10s Gb/s
~5 kHz

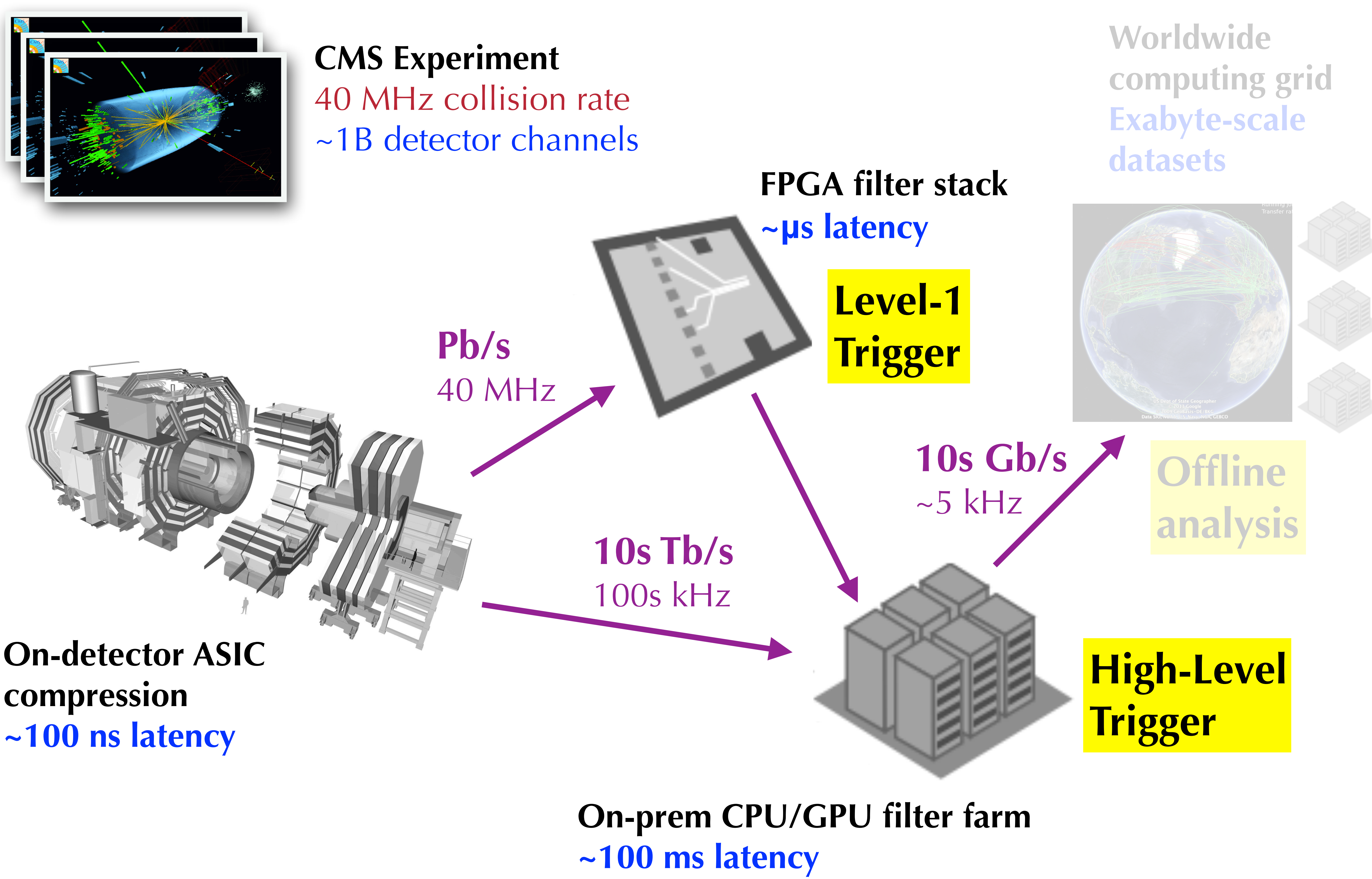
Worldwide computing grid
Exabyte-scale datasets

Offline analysis

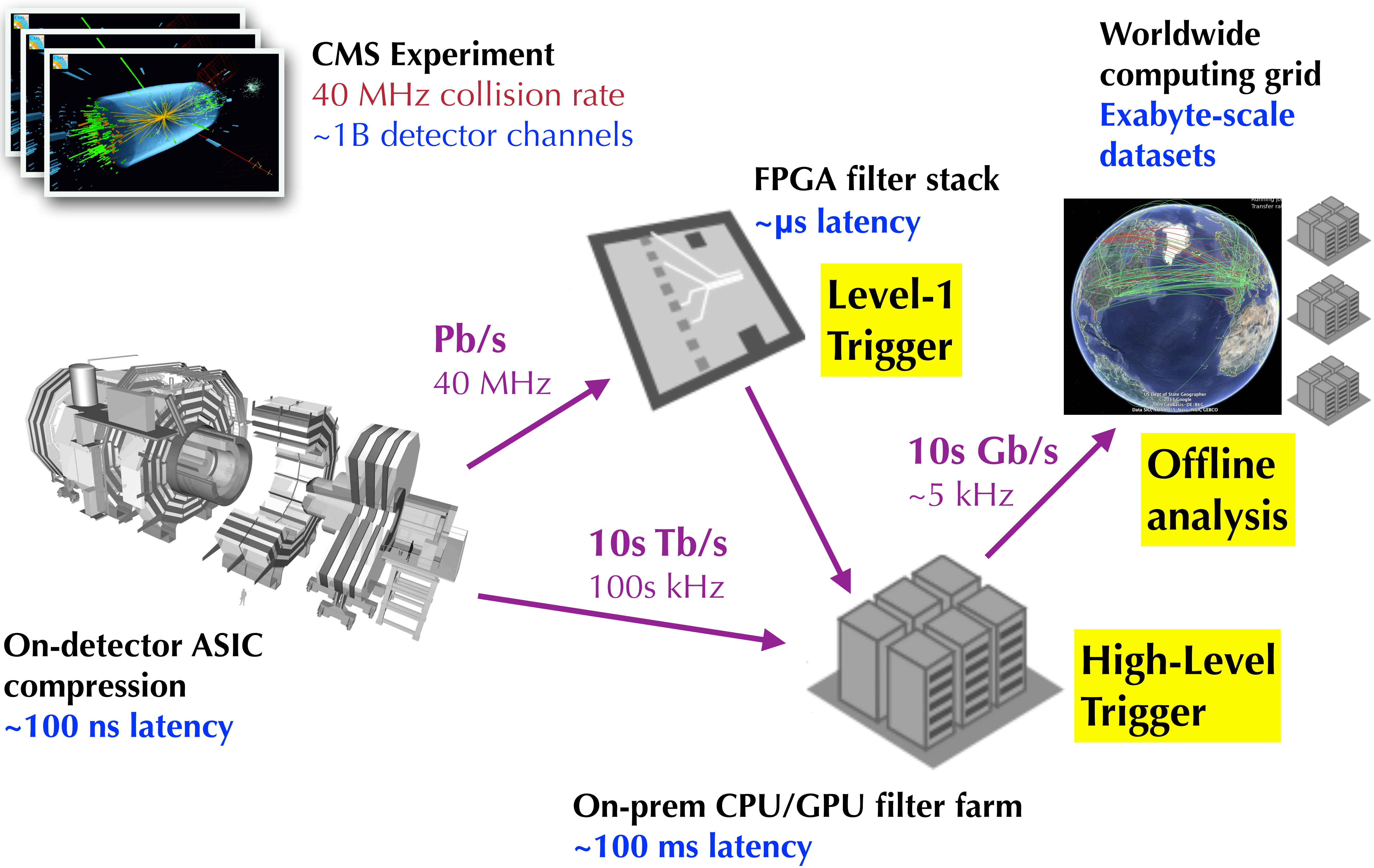
Data reduction workflow @ LHC



Data reduction workflow @ LHC

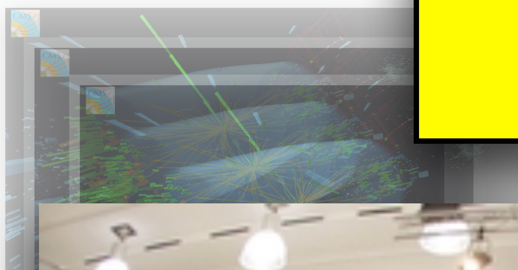


Data reduction workflow @ LHC



Make physics discoveries with 0,0025% of the events!

(the rest is lost…)



40 MHz collision rate

Exabyte-scale datasets

10s Gb/s

~5 kHz

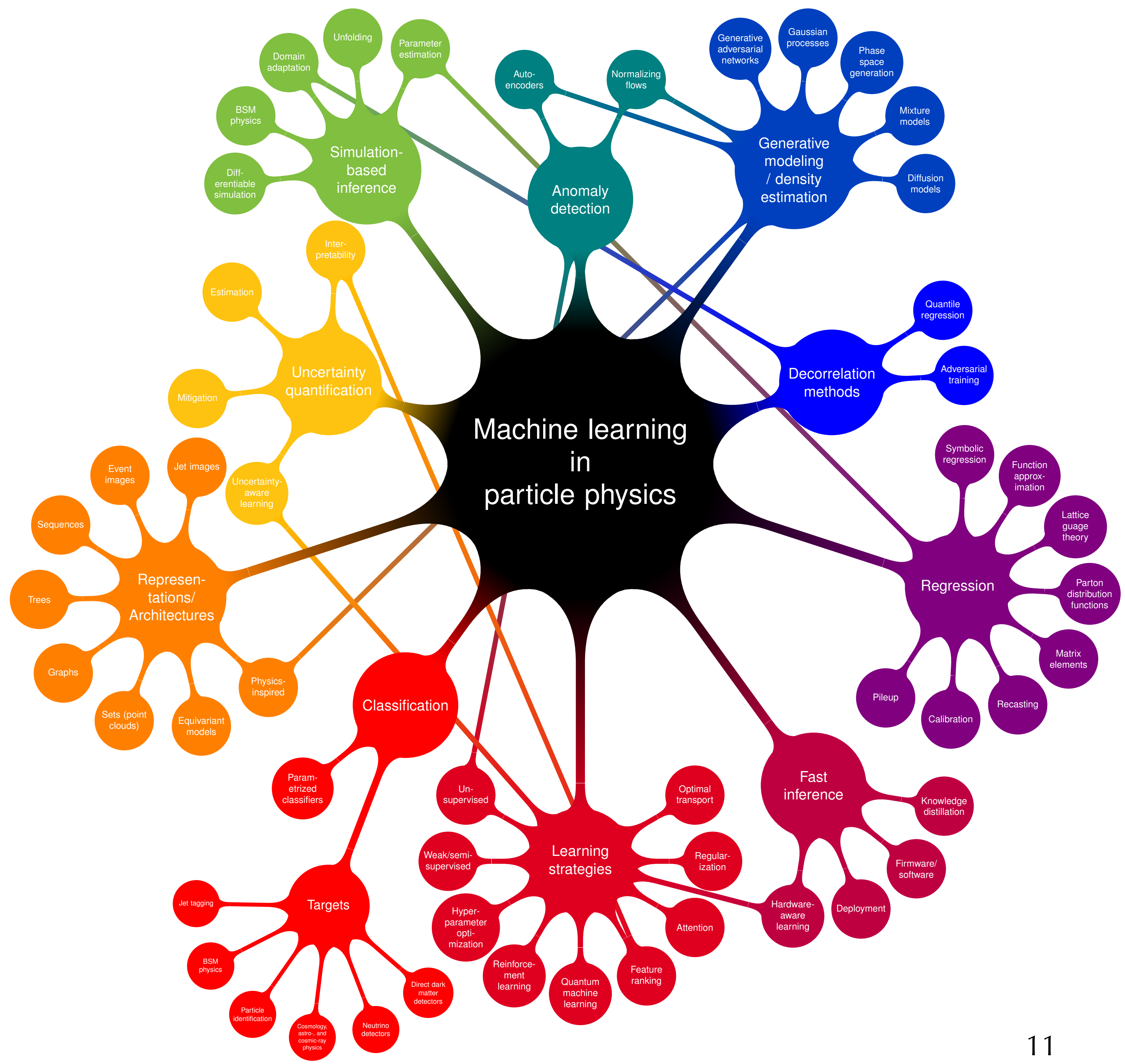
On-detector ASIC compression

~100 ns latency

The role of AI

- **Machine Learning is used in particle physics since the '80s**
  - it was shallow networks back then
- **Over the last decade a rapid progress guided by technological breakthrough led to a revolution in this area**
  - this the era of Deep Learning

https://iml-wg.github.io/HEPML-LivingReview/



Deep Learning @ LHC

DL for classification:

heavy jet tagging
heavy flavour jet tagging
exotic jets
tau leptons
event level

DL beyond classification:

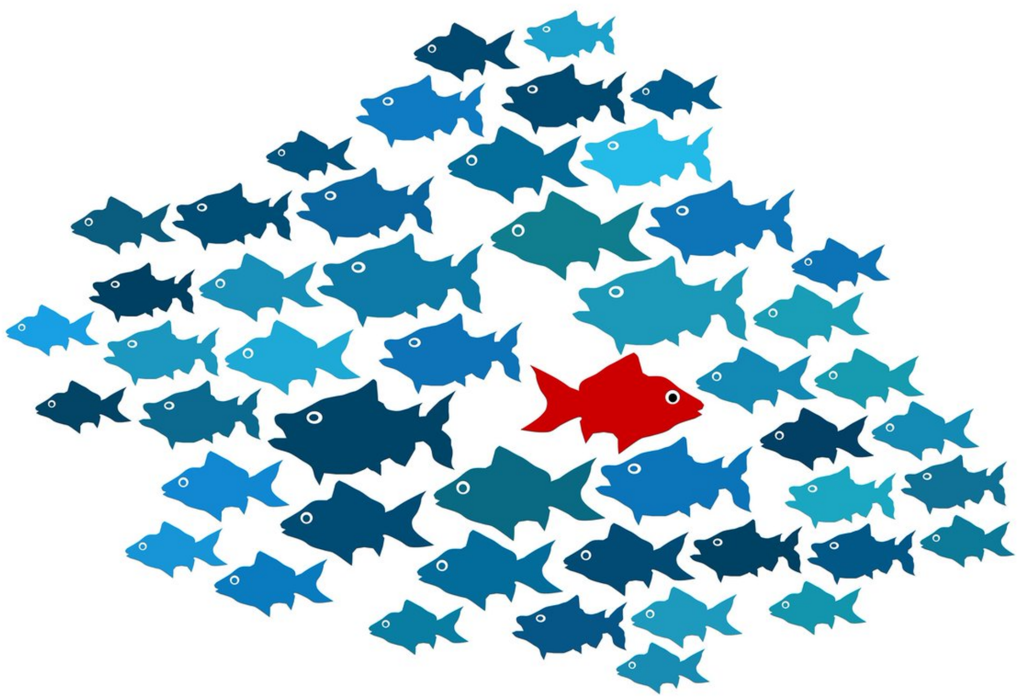
mass and energy regression
background estimation
simulation-based inference
inverse problems/unfolding
anomaly detection
uncertainty quantification
reconstruction & simulation
triggering

Computing software & hardware for DL:

optimized inference in central software for CPU/GPU
GPU hardware on-site for software trigger system & grid sites
more powerful chips in hardware trigger system & development of portable tools
ML-friendly central data format and scalable processing tools

In this talk, my personal choice of highlights (efforts I actively contribute to)
…there is a lot more ongoing!

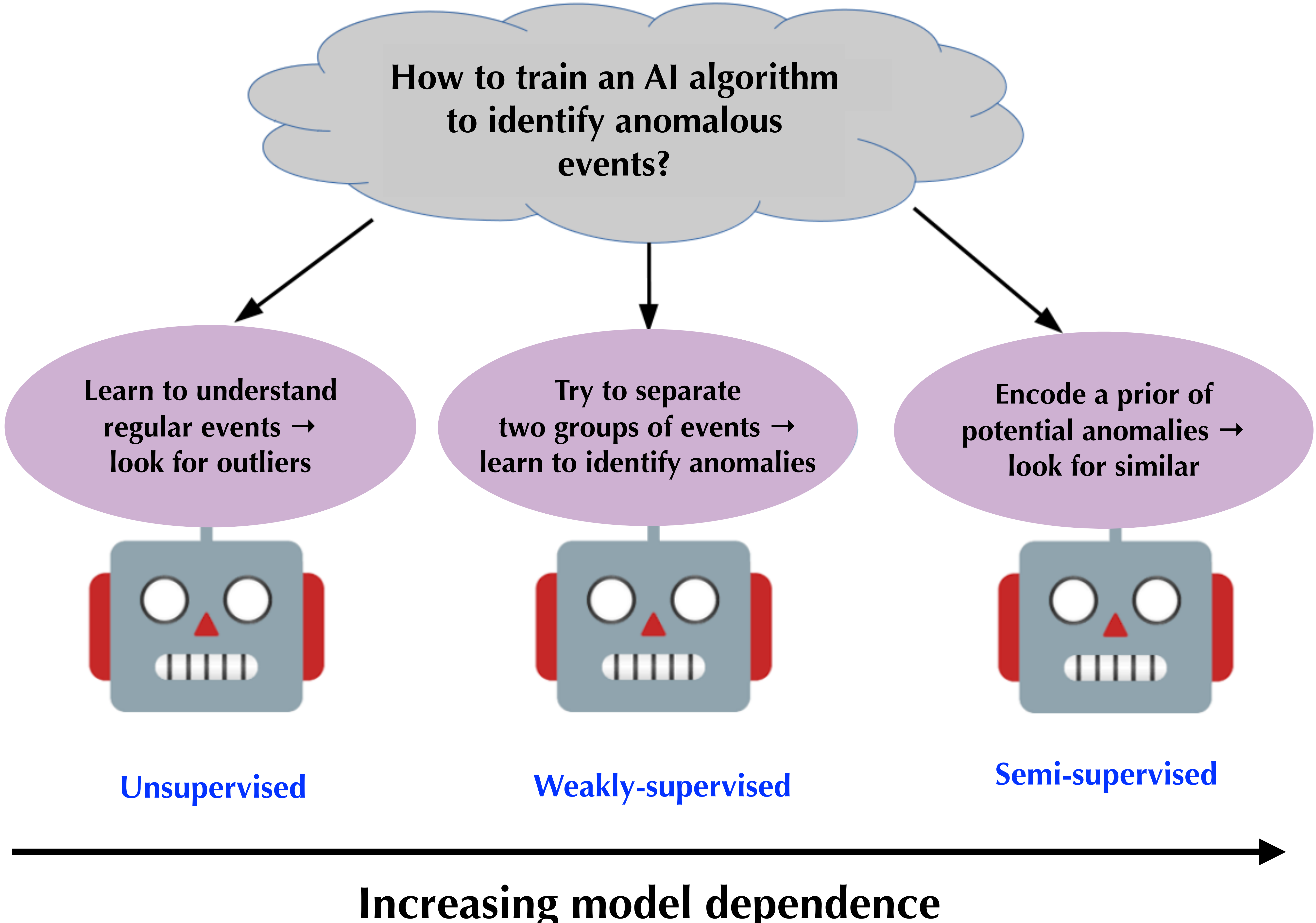
Anomaly detection in a nutshell



Machine learning based anomaly detection algorithms can be used to look at our data without model assumptions

Main idea: **learn directly from data how the standard model looks like**
⇒ **eliminate signal priors** and search for **anything**

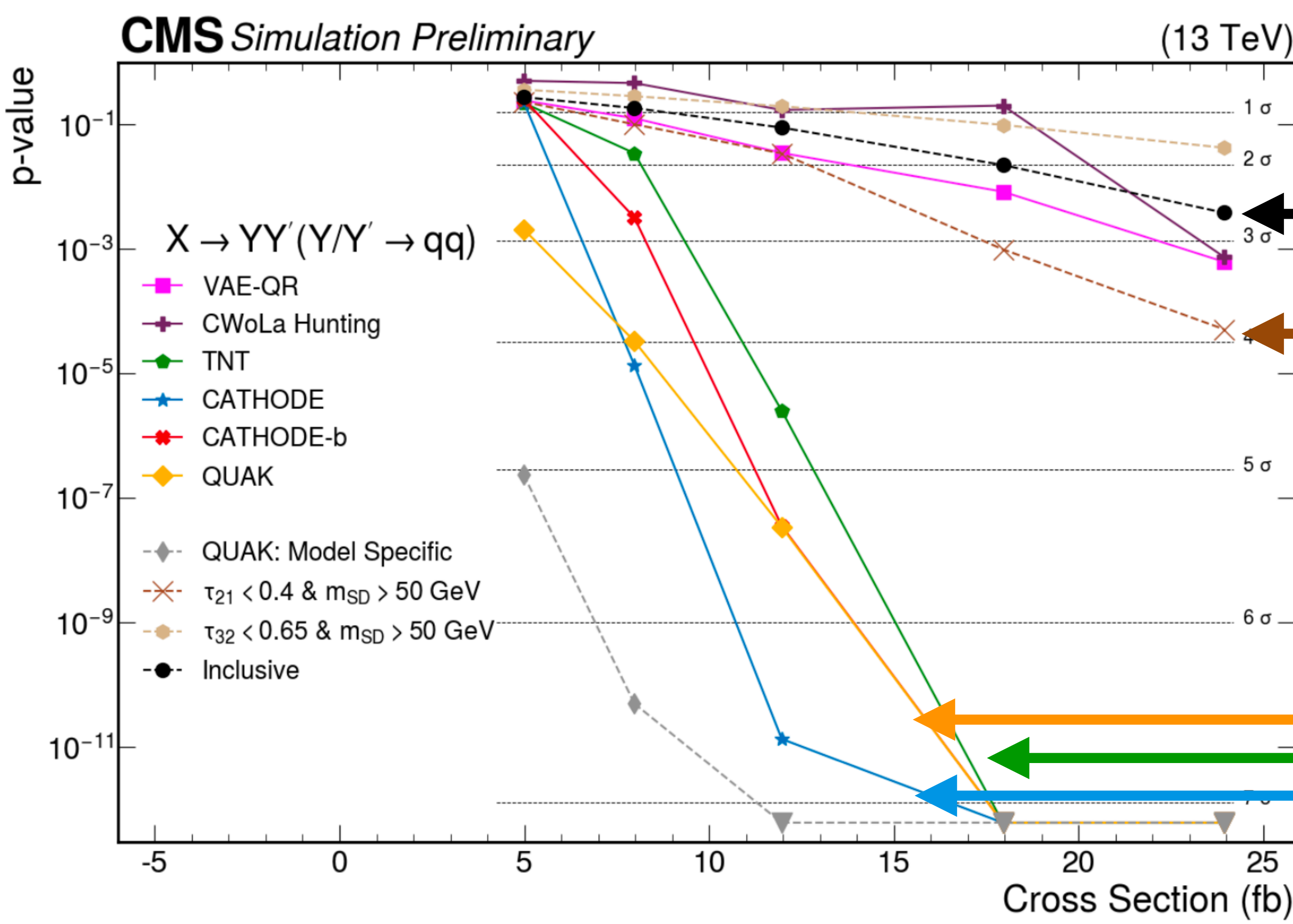
anomalous wrt standard model


How to train an AI algorithm to identify anomalous events?
Learn to understand regular events → look for outliers
Try to separate two groups of events → learn to identify anomalies
Encode a prior of potential anomalies → look for similar
Unsupervised
Weakly-supervised
Semi-supervised
Increasing model dependence

Anomaly detection in action!

- **Inject anomalies** of varying production rate (*cross section*) in background simulation and calculate **discovery sensitivity metric** (p-value)

- Obtain comparison of sensitivity of different methods against standard analysis methods



traditional
dijet search

dedicated search
for this signal

anomaly detection
methods

Higher discovery probability!

[CMS-EXO-22-026]

Apply to LHC data!

Anomaly detection in action
… but no discovery quite yet…

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Phys. Rev. Lett. 125 (2020) 131801
DOI: 10.1103/PhysRevLett.125.131801

CERN-EP-2020-062
12th January 2022

Dijet resonance search with weak supervision using $\sqrt{s}$ = 13 TeV pp collisions in the ATLAS detector

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Phys. Rev. Lett. 132 (2024) 081801
DOI: 10.1103/PhysRevLett.132.081801

CERN-EP-2023-112
February 22, 2024

Search for new phenomena in two-body invariant mass distributions using unsupervised machine learning for anomaly detection at $\sqrt{s}$ = 13 TeV with the ATLAS detector

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Phys. Rev. D 108 (2023) 052009
DOI: 10.1103/PhysRevD.108.052009

CERN-EP-2023-045
December 6, 2023

Anomaly detection search for new resonances decaying into a Higgs boson and a generic new particle X in hadronic final states using $\sqrt{s}$ = 13 TeV pp collisions with the ATLAS detector

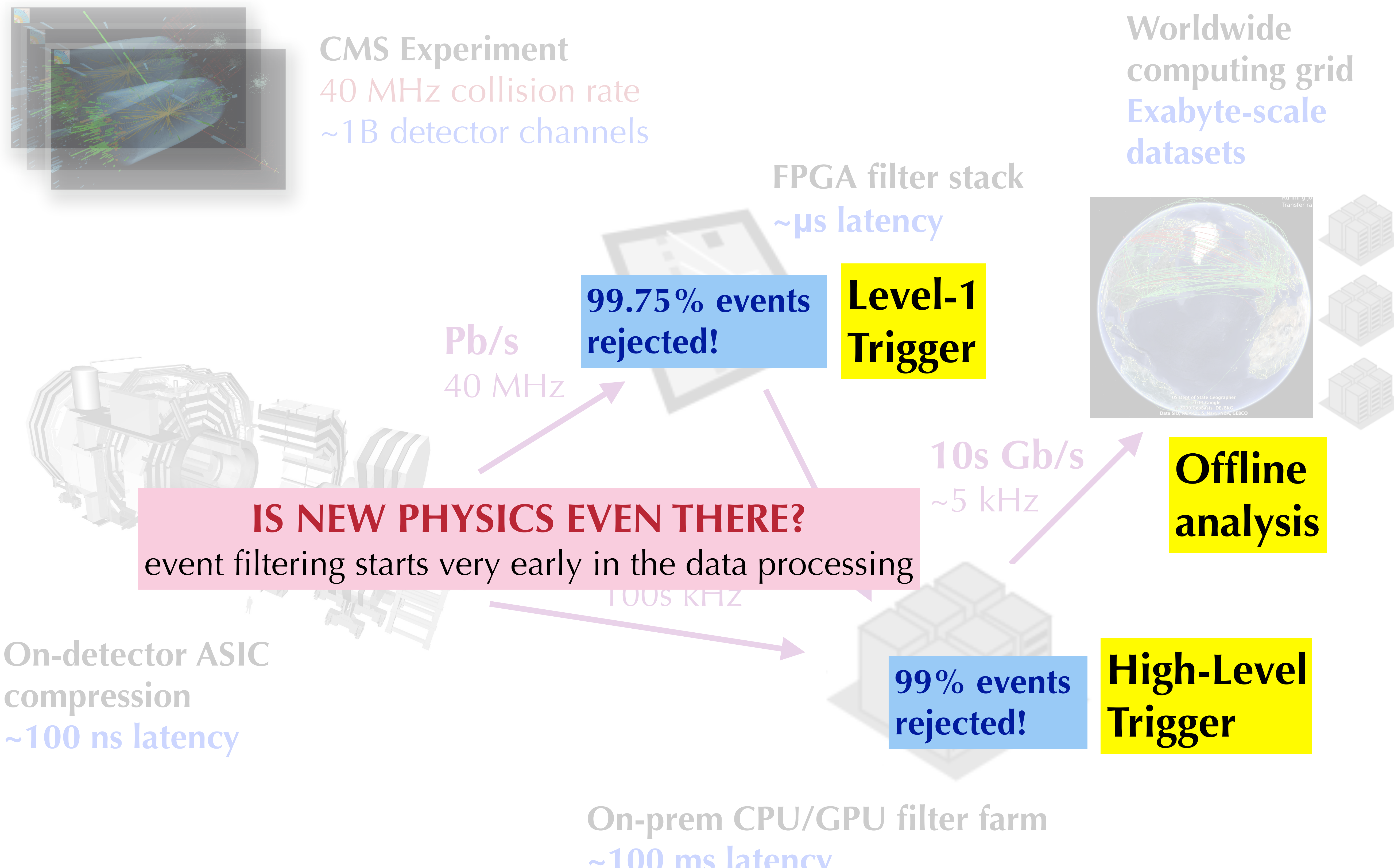
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

CERN-EP-2024-291
2024/12/06

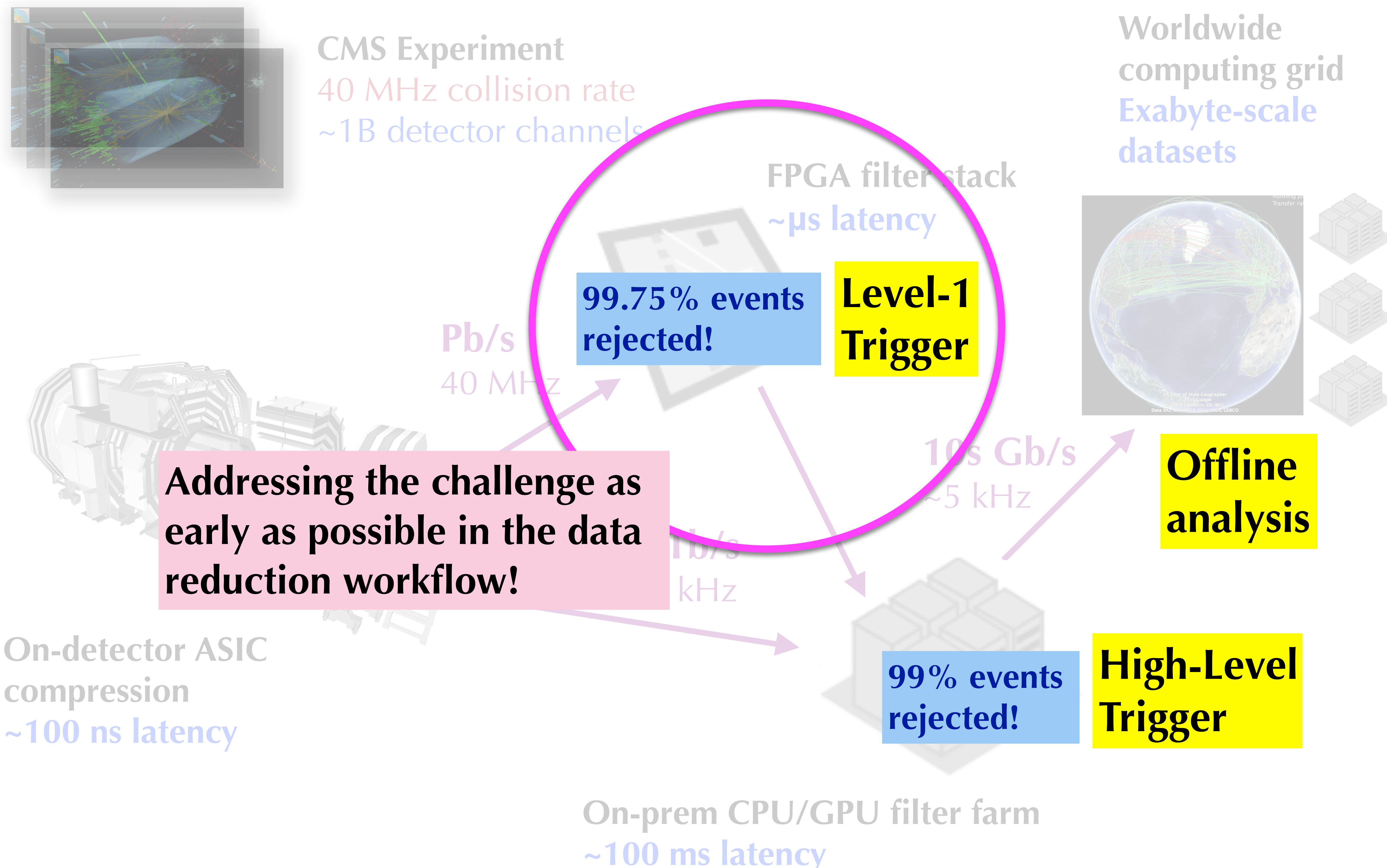
CMS-EXO-22-026

Model-agnostic search for dijet resonances with anomalous jet substructure in proton-proton collisions at $\sqrt{s} = 13$ TeV

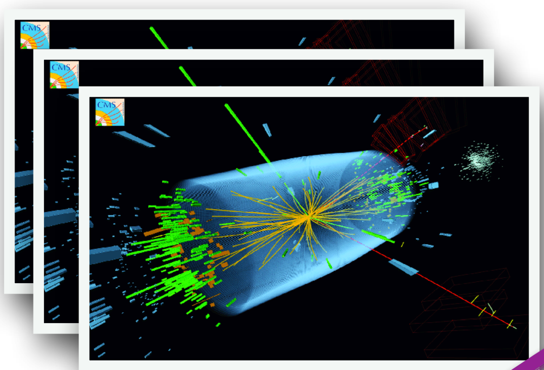
Data reduction workflow @ LHC



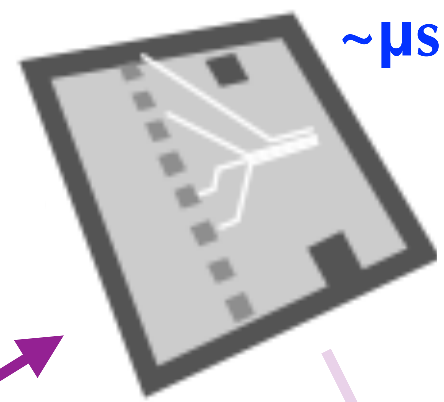
Data reduction workflow @ LHC



Data reduction workflow @ LHC

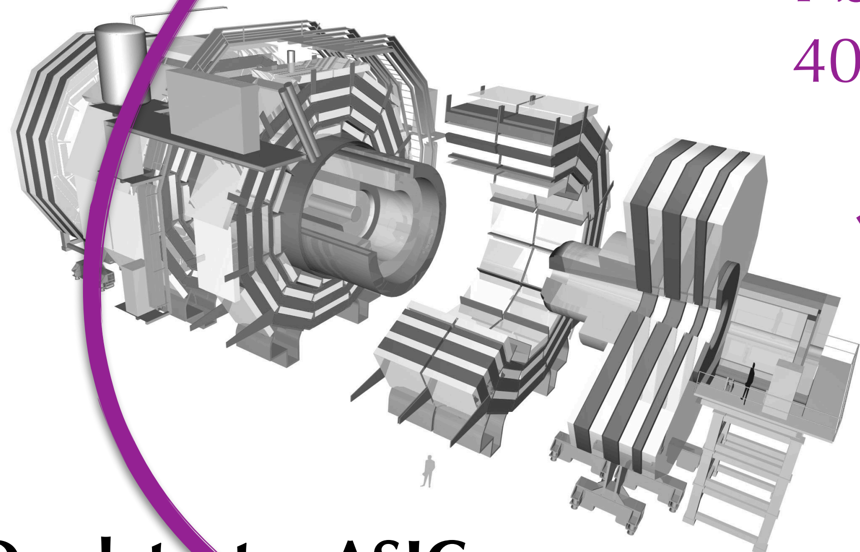


CMS Experiment
40 MHz collision rate
~1B detector channels



FPGA filter stack
~μs latency

Level-1 Trigger



Pb/s
40 MHz

Worldwide computing grid
Exabyte-scale datasets

10s Gb/s

On-detector ASIC compression
~100 ns latency

One of the challenges today is to bring complex deep learning models to the extreme environments closer to the data source!

Industry tools typically suboptimal…

On-prem CPU/GPU filter farm
~100 ms latency

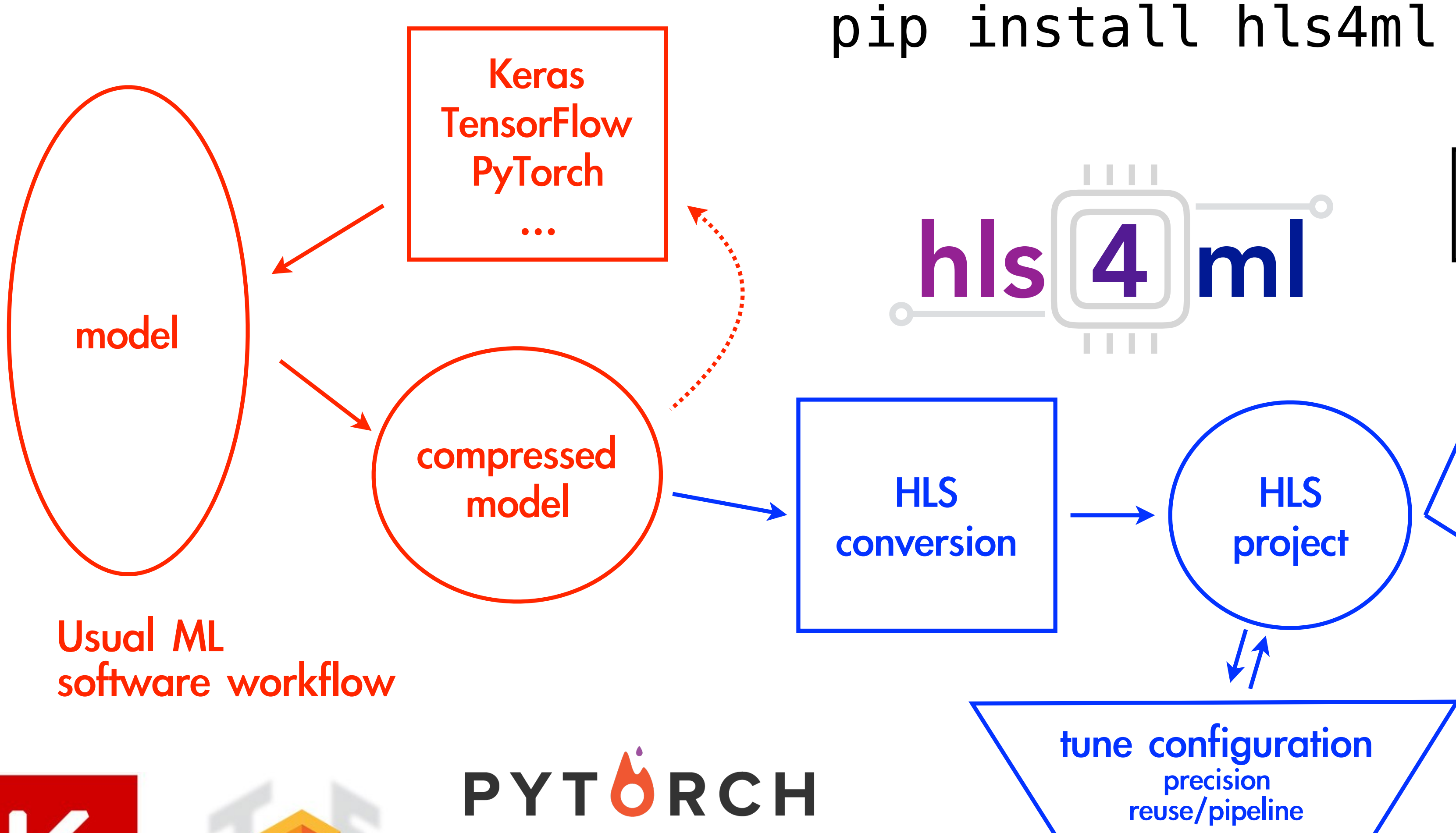
Bring ML models to hardware for real-time AI

high level synthesis for machine learning

A tool to efficiently program the FPGA hardware for Neural Networks with experimental constraints in mind!

Many use cases in HEP and beyond… and still growing!
(see Fast Machine Learning For Science Workshop Oct '24)

```
pip install hls4ml
```



Vivado™ HLS

HLS COMPILER

Catapult
Mentor
A Siemens Business

K + TensorFlow

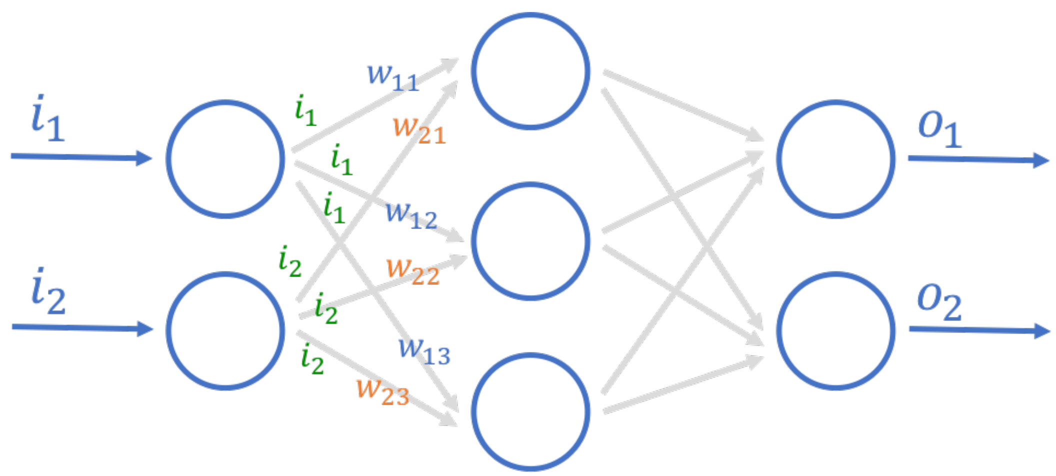
PYTORCH

ONNX

https://fastmachinelearning.org/hls4ml/

Neural Network inference on FPGA

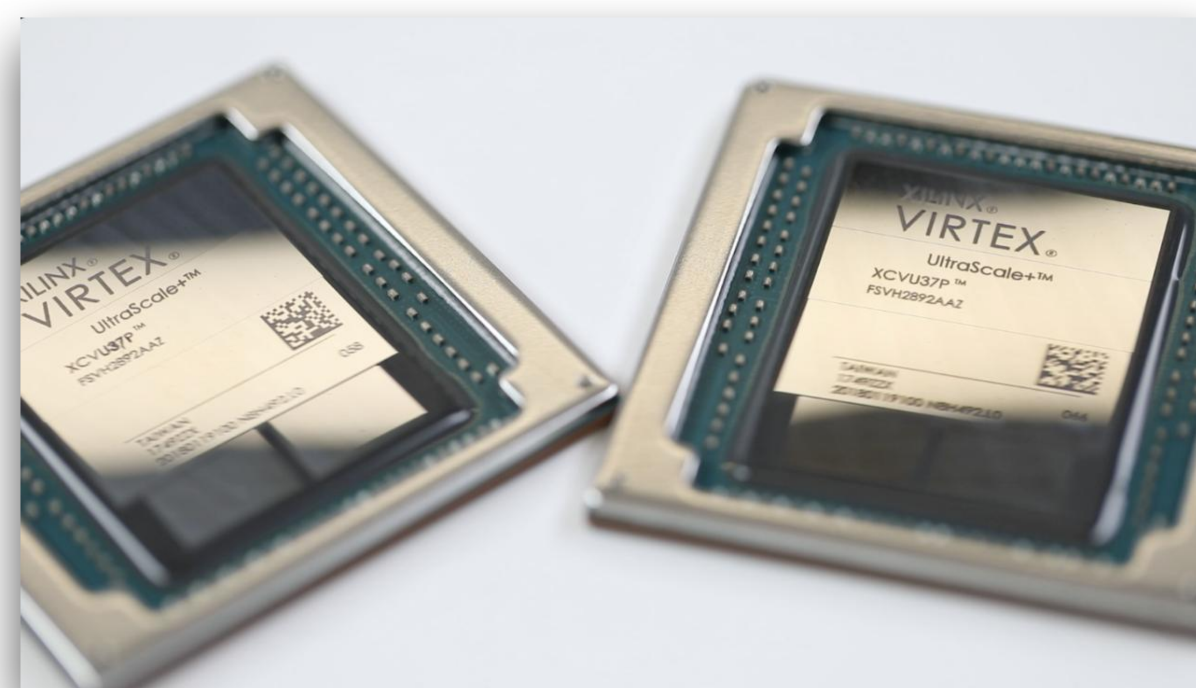
Neural network inference
=
matrix multiplication



$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

Efficient implementation on FPGA uses
DIGITAL SIGNAL PROCESSORS

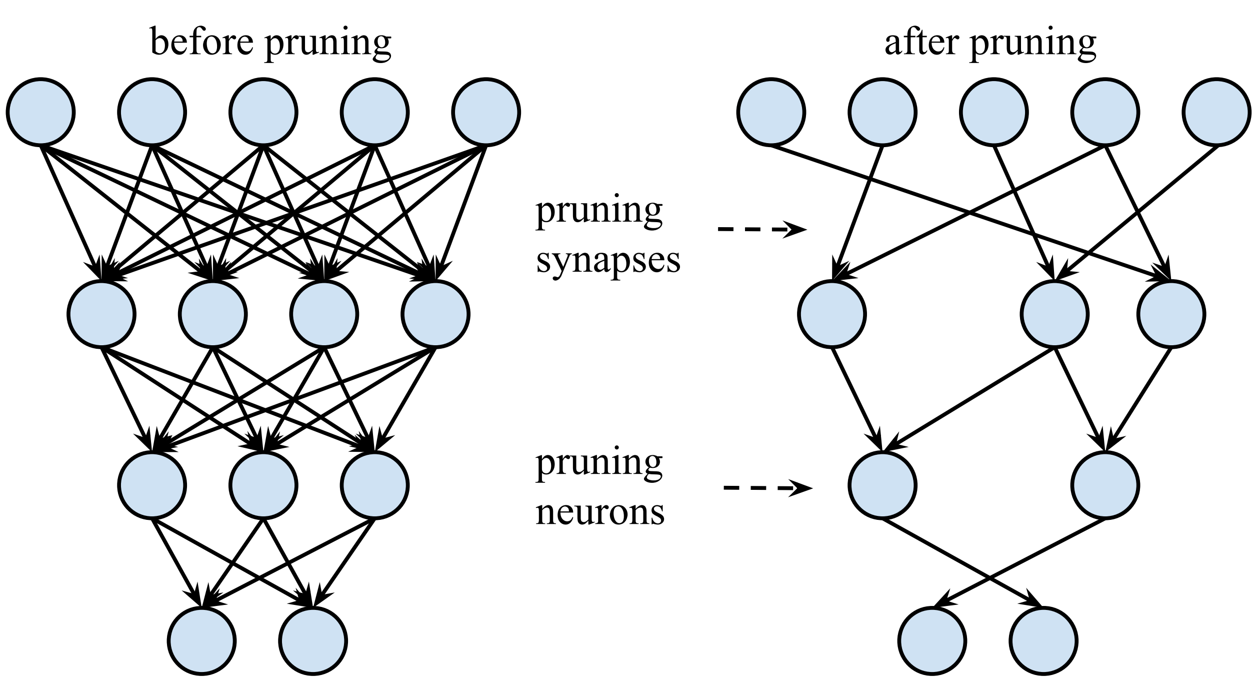
There are about 5–10k DSPs in modern FPGAs!

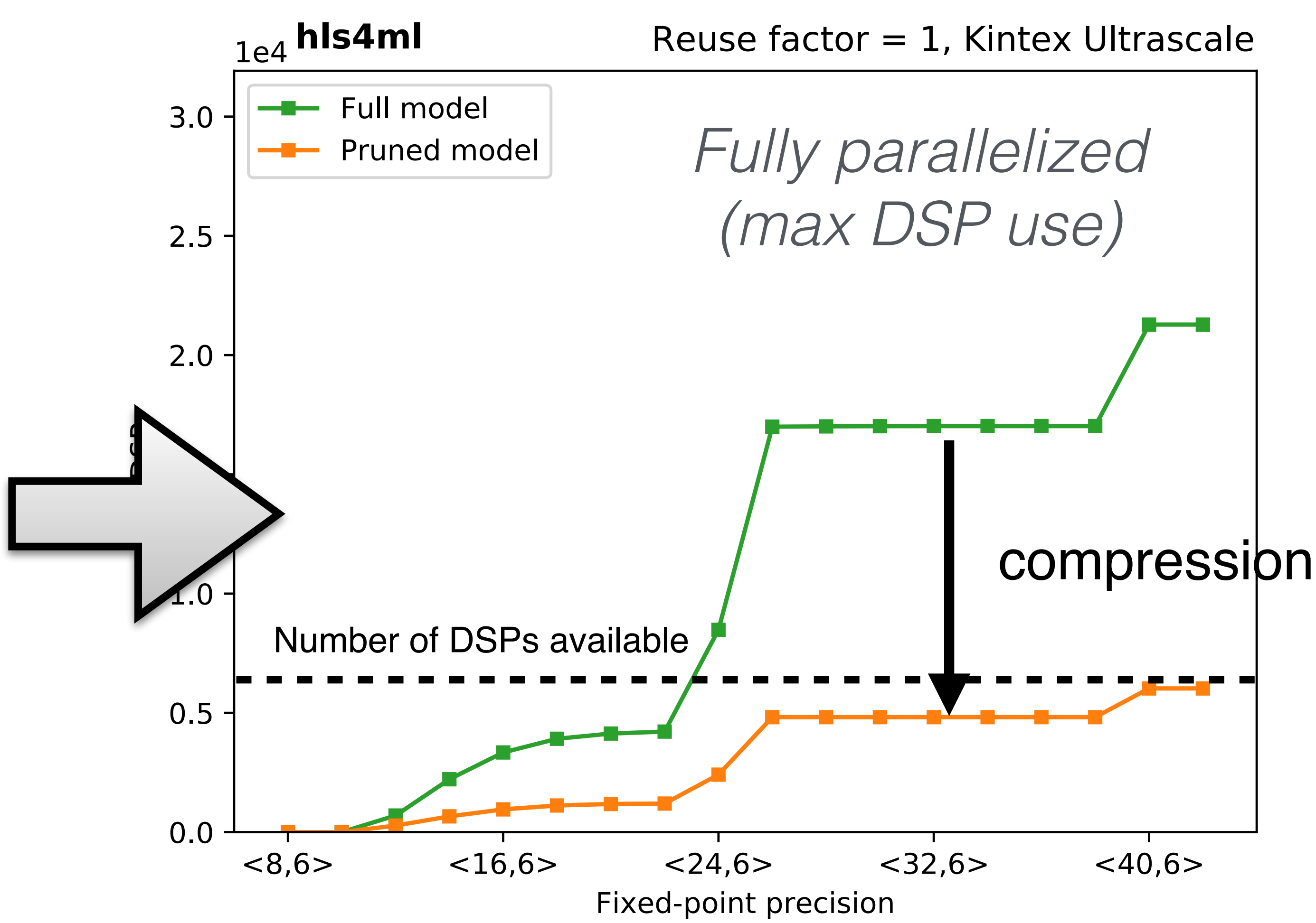


ex: Xilinx Virtex Ultrascale +

Make the model fit on one chip

- Some tricks are needed here:
    - **Compression/pruning:** remove the connections that play little role for final decision

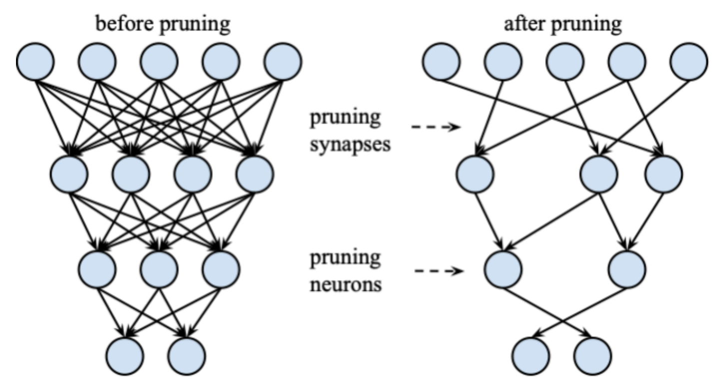


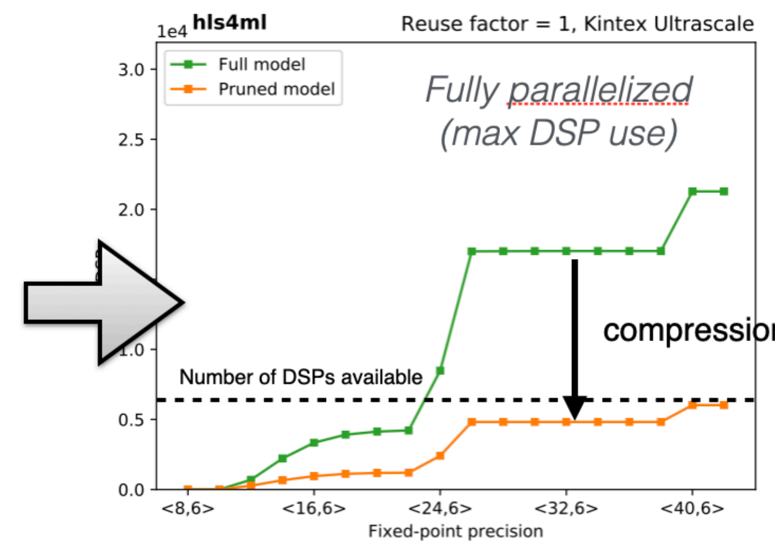


70% compression ~ 70% fewer DSPs

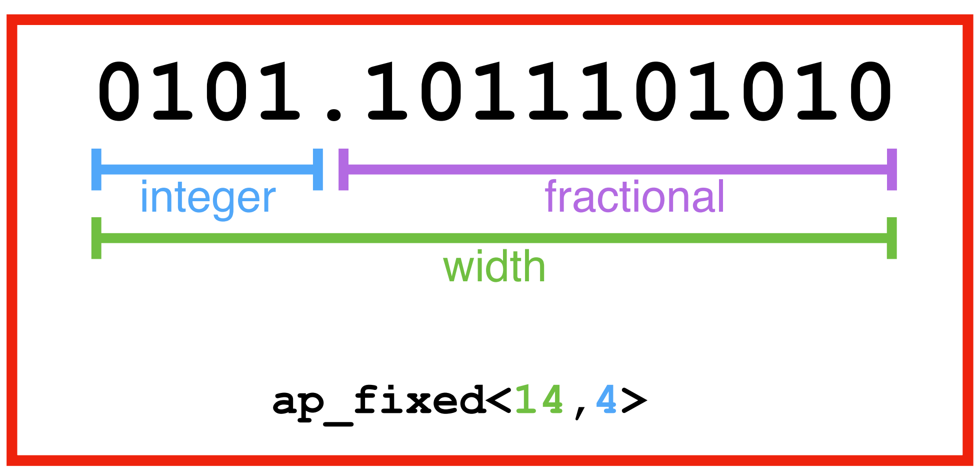
Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision



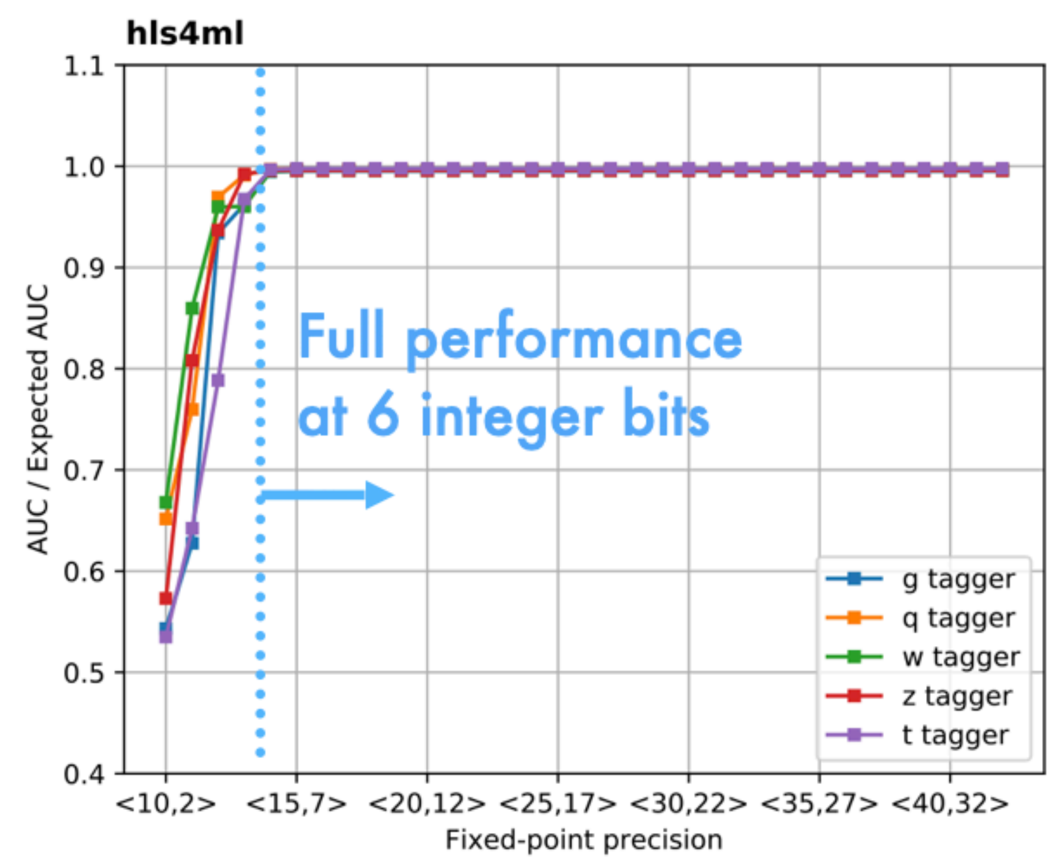


  - **Quantisation:** represents numbers with few bits reduce resources



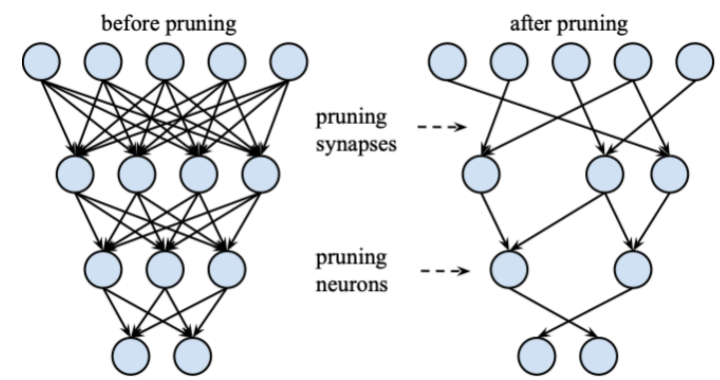
Scan integer bits

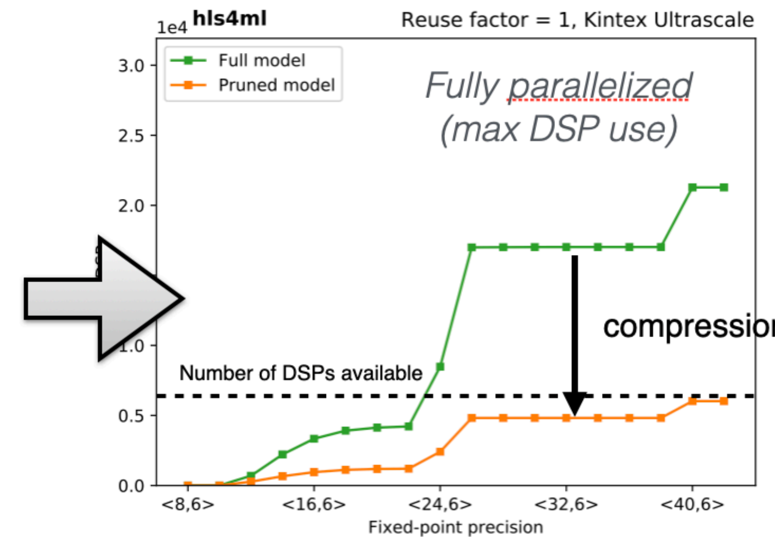
Fractional bits fixed to 8



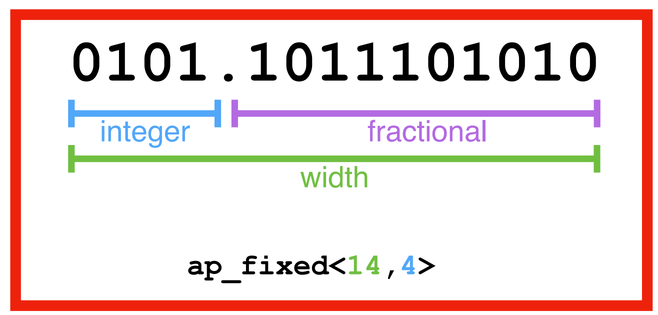
Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

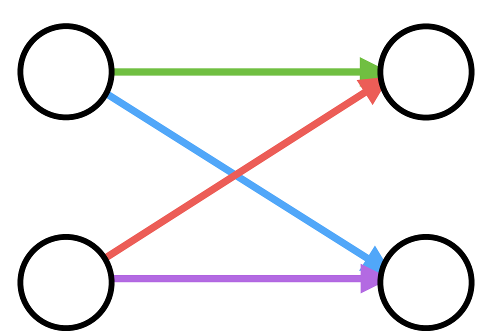


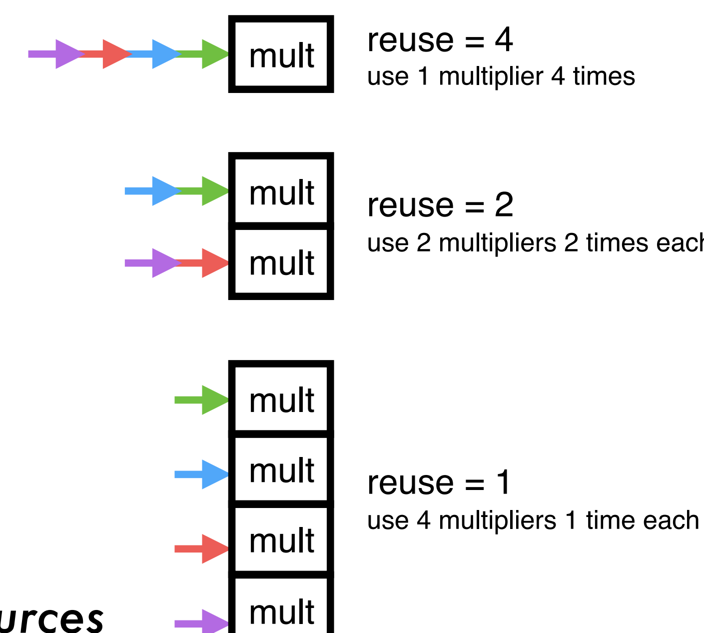


  - **Quantisation:** represents numbers with few bits reduce resources



  - **Reuse:** allocate resources for each operation (run all network in one clock) vs spread calculation across several clock cycles

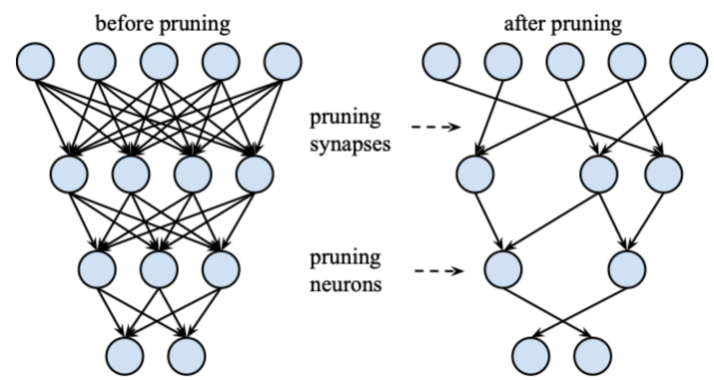


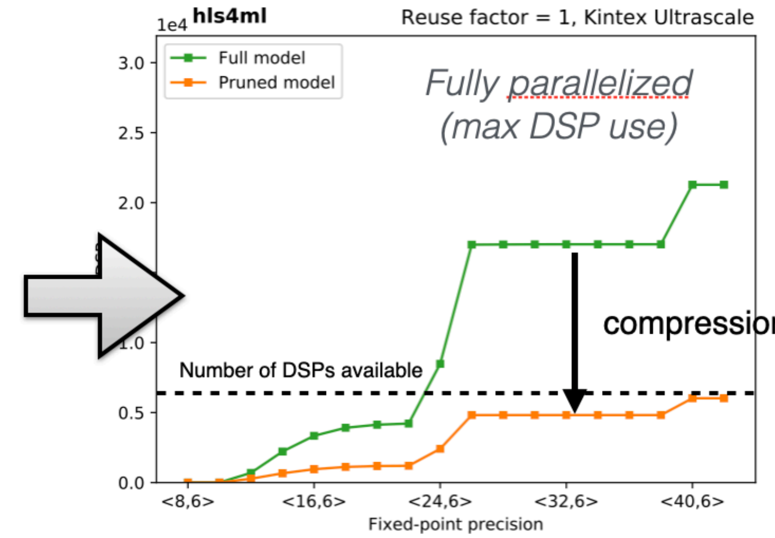


more parallelization → more resources

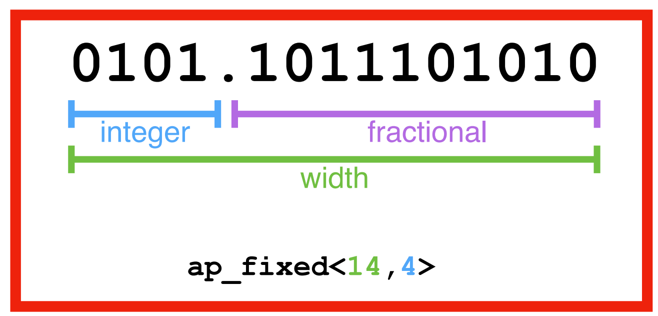
Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

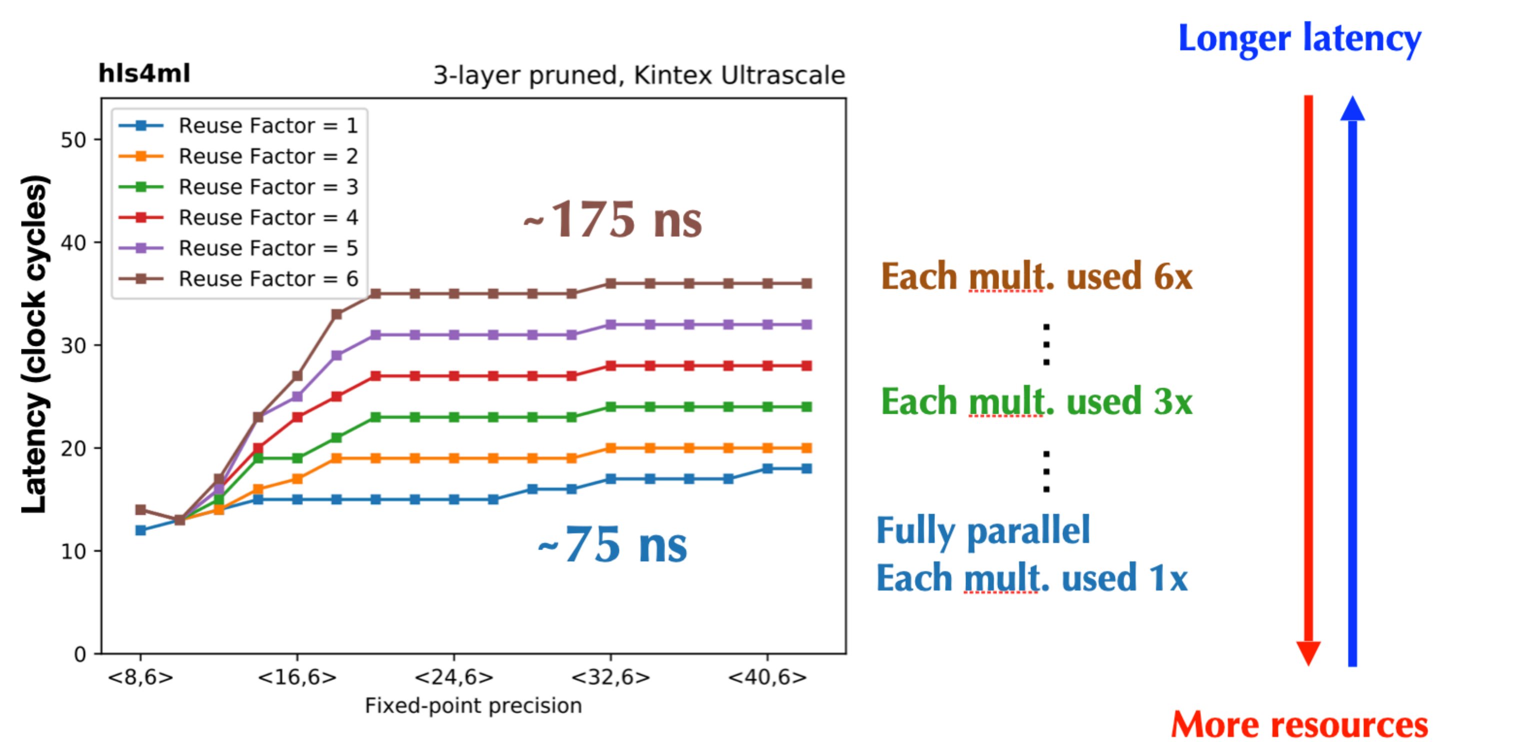




  - **Quantisation:** represents numbers with few bits reduce resources

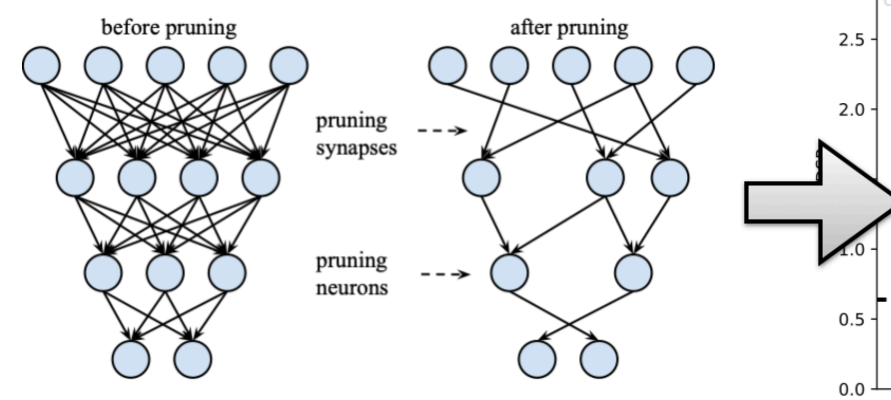


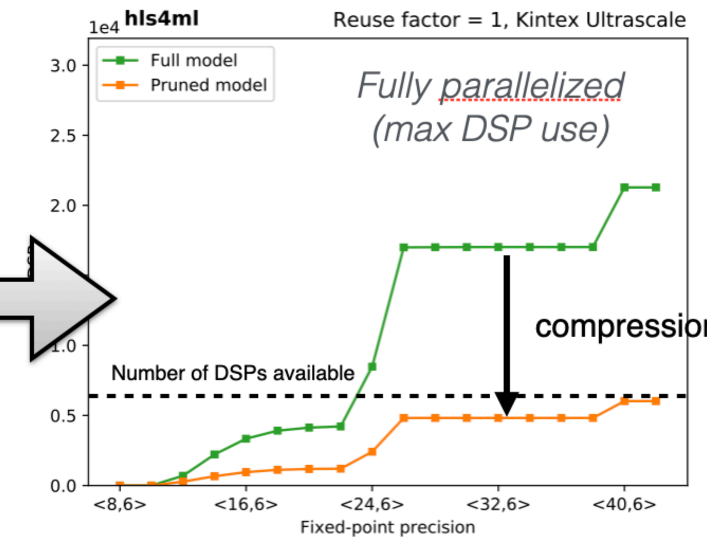
  - **Reuse:** allocate resources for each operation (run all network in one clock) vs spread calculation across several clock cycles



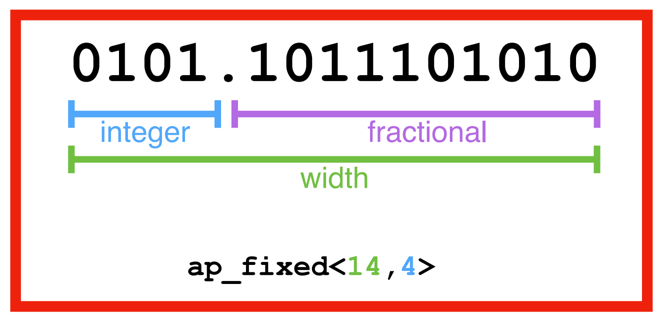
Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

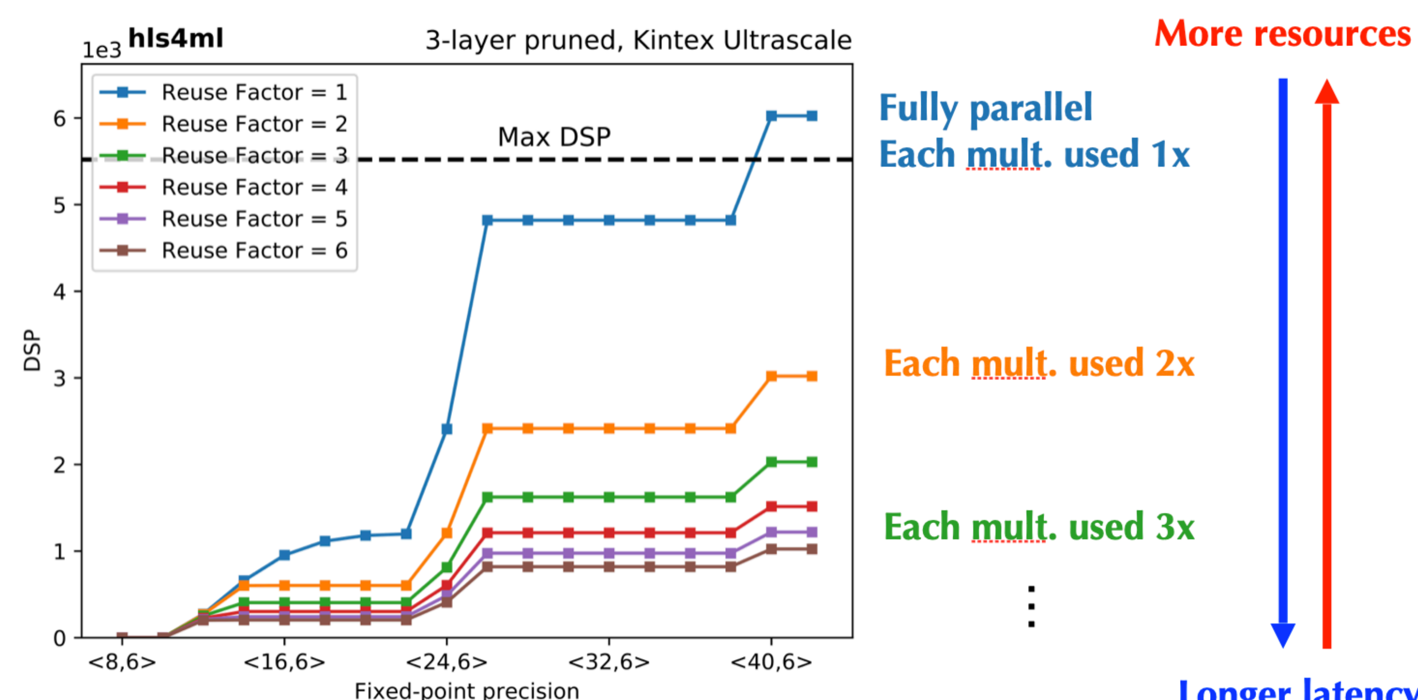




  - **Quantisation:** represents numbers with few bits reduce resources

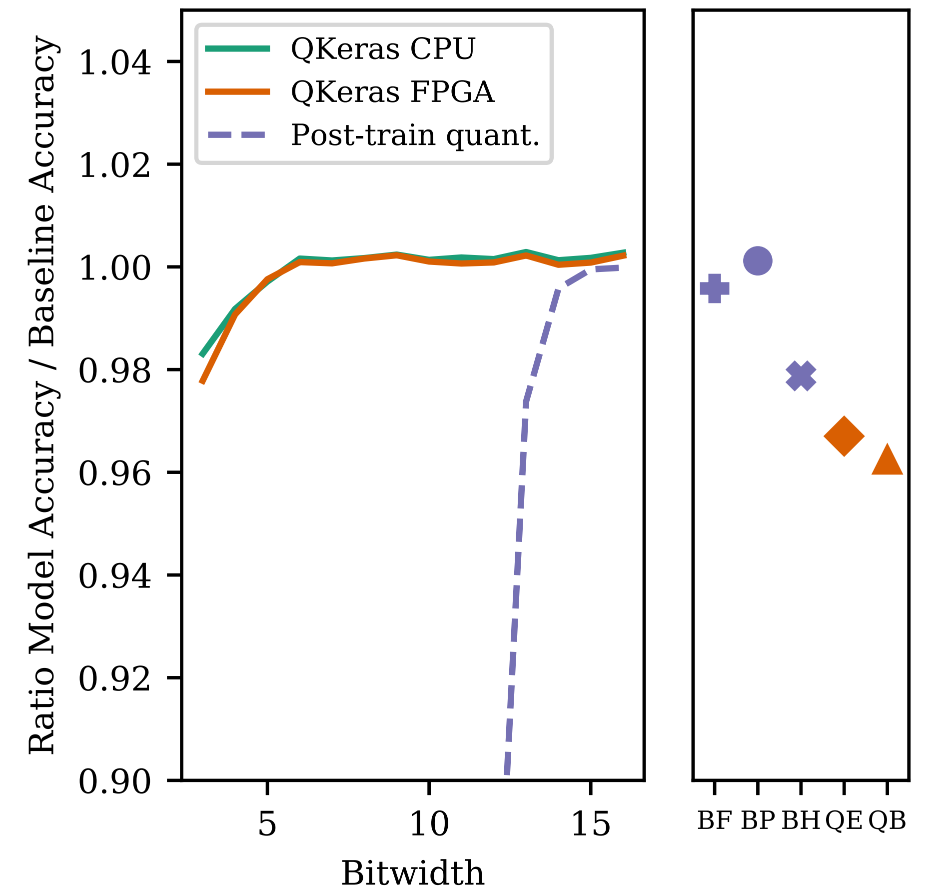


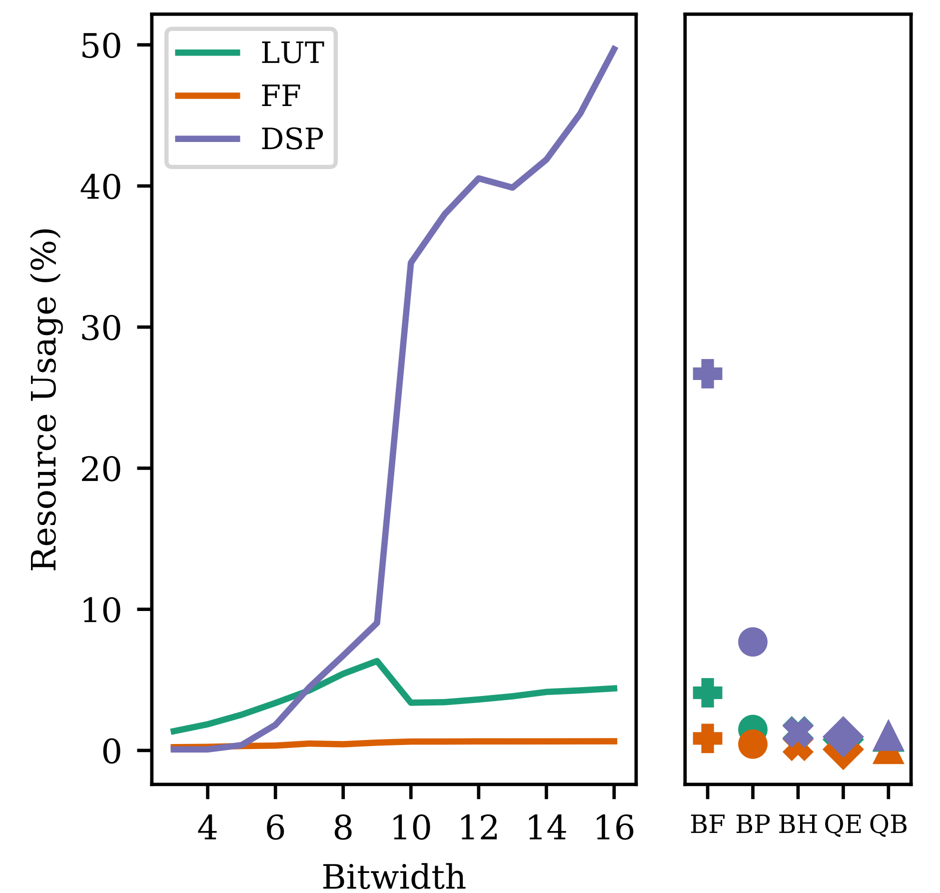
  - **Reuse:** allocate resources for each operation (run all network in one clock) vs spread calculation across several clock cycles



Quantization-aware training

- Post-training quantization can affect accuracy
  - for a given bit allocation, the loss minimum at floating-point precision might not be the minimum anymore
- One could specify quantization while look for the minimum
  - maximize accuracy for minimal FPGA resources
- Workflow: quantization-aware training with [Google QKeras]() and firmware design with [hls4ml]() for best NN inference on FPGA performance





C. N. Coelho et al.: Nature Machine Intelligence, Volume 3 (2021)

Ultra-fast anomaly detection @ CMS

Learn typicality: by training on unbiased dataset

CMS establishing a new trigger paradigm with sub-μs autoencoders for anomaly detection!

AXOL1TL

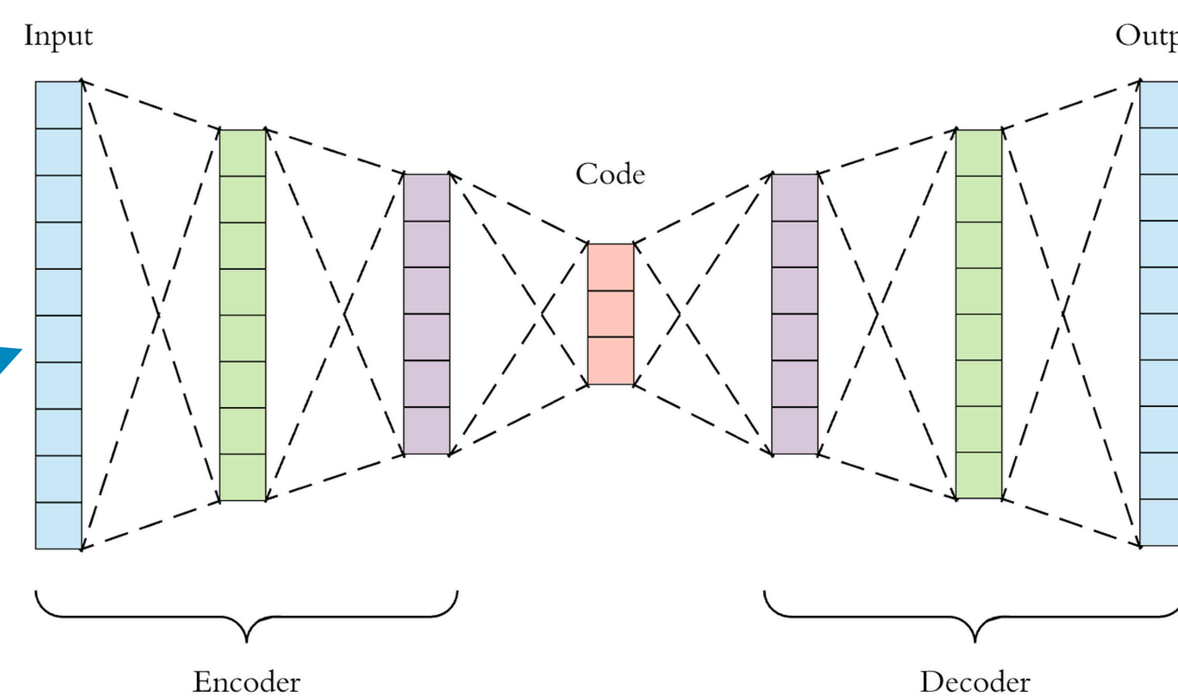
	p_T	η	φ
MET		N/A	
4 e/γ			
4 μ			
10 jets			

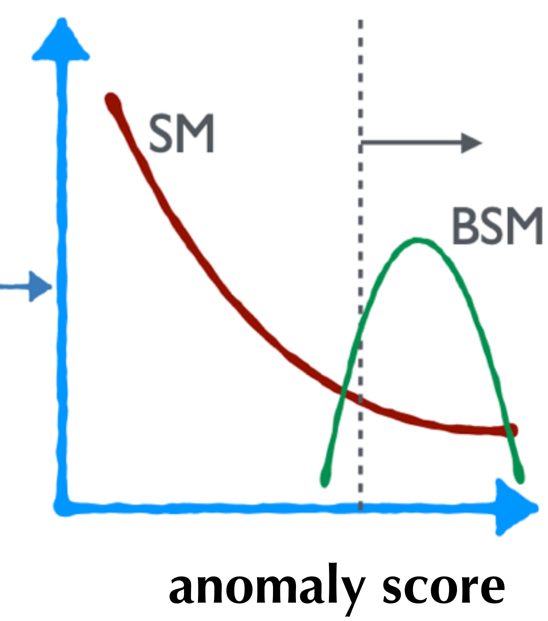
From calorimeter and muon trigger system:

Objects: 10 jets, 4 muons, e/**γ**, MET

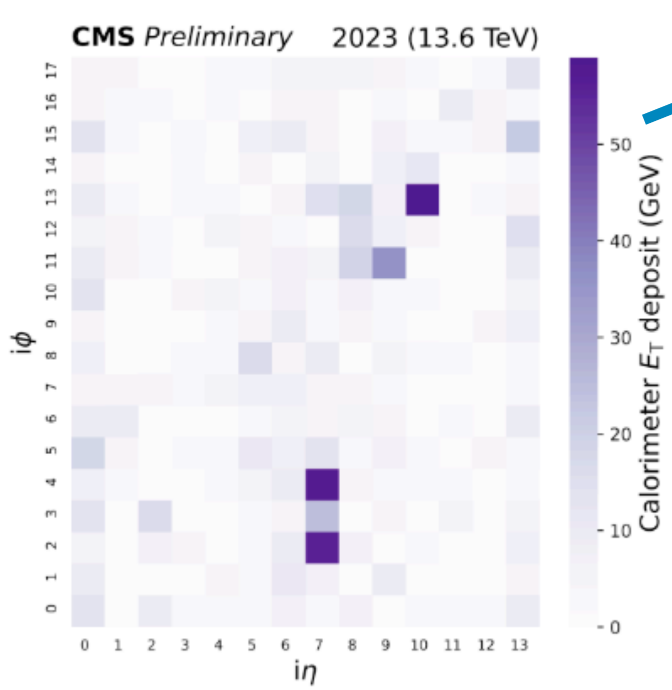
Features: p_T, η, ϕ (in raw integer values)

Architecture: MLP





CICADA



Low-level inputs: aggregated calorimeter towers

Architecture: 2D CNN w/ knowledge distillation

[CMS-DP-2023-086]

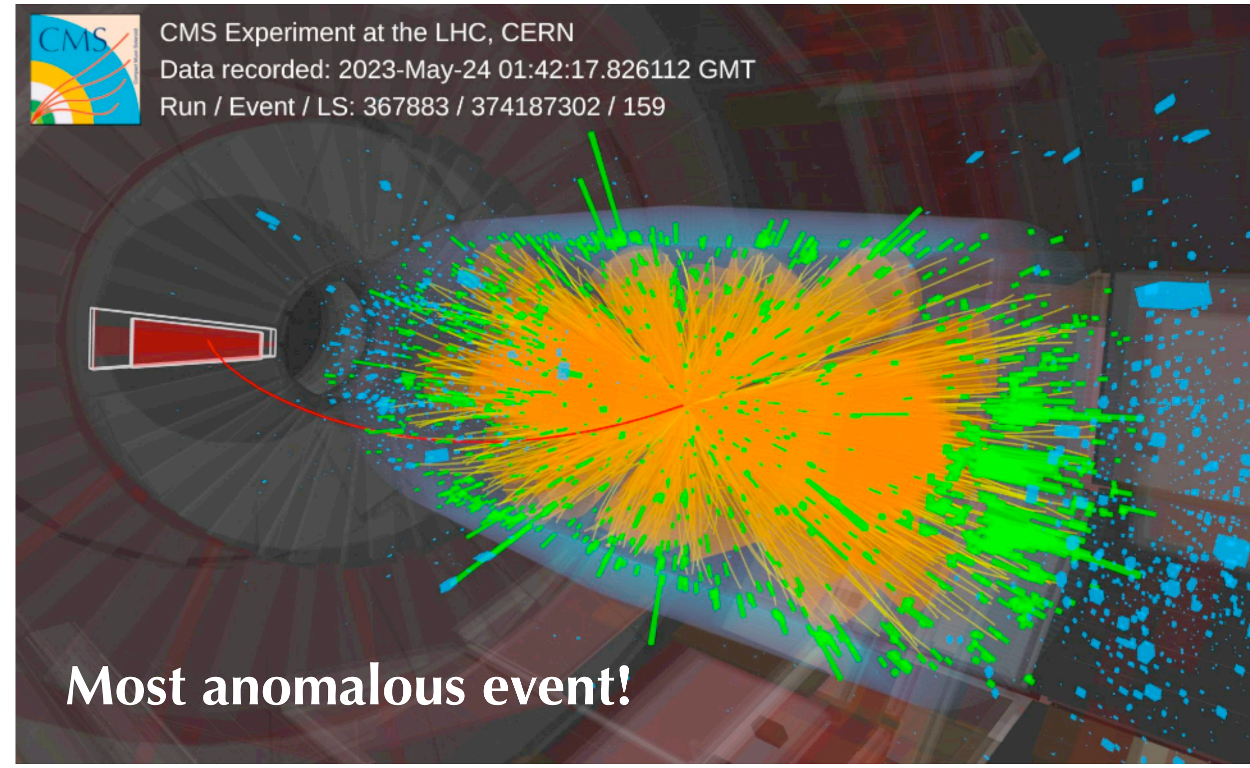
Ultra-fast anomaly detection @ CMS

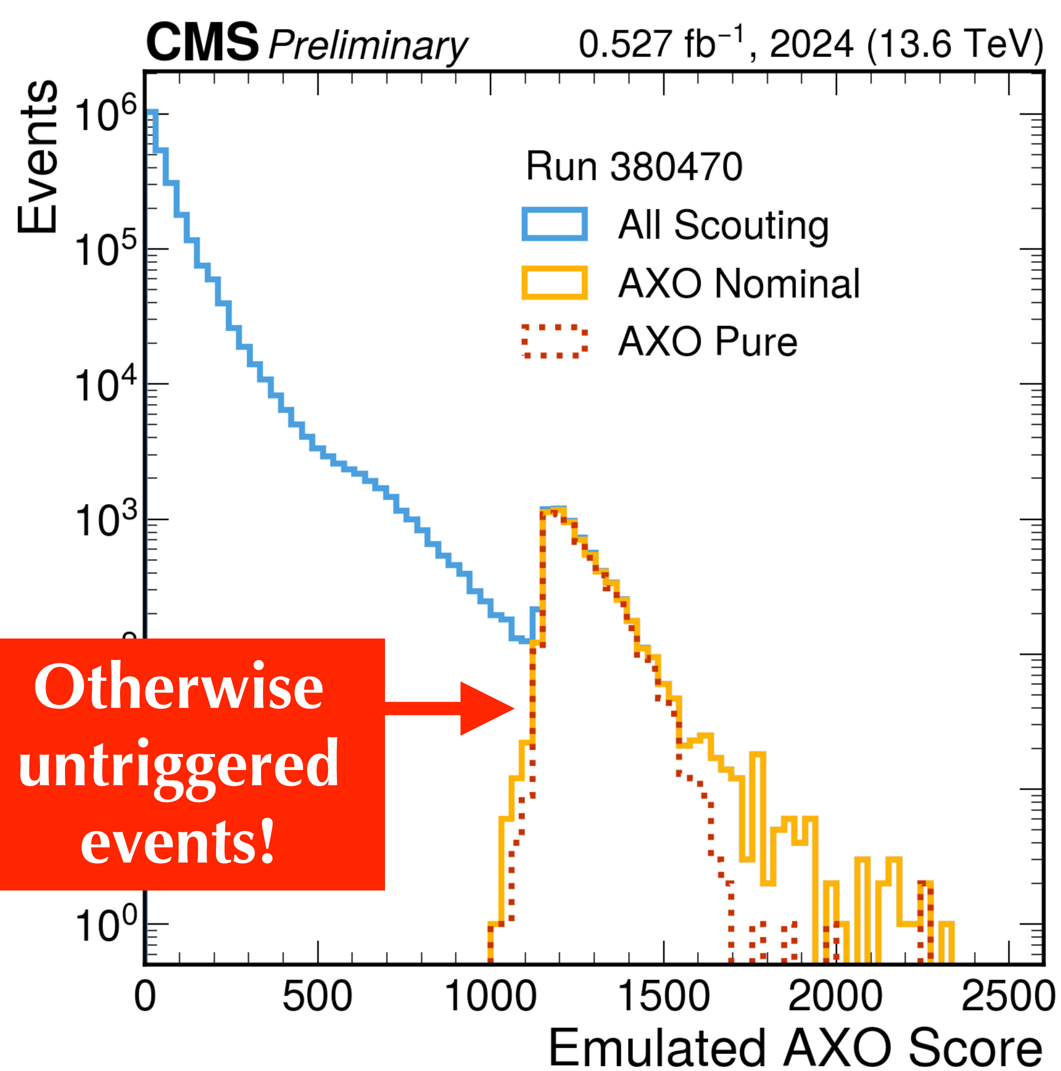
Anomaly eXtraction Online Level-1 Trigger aLgorithm

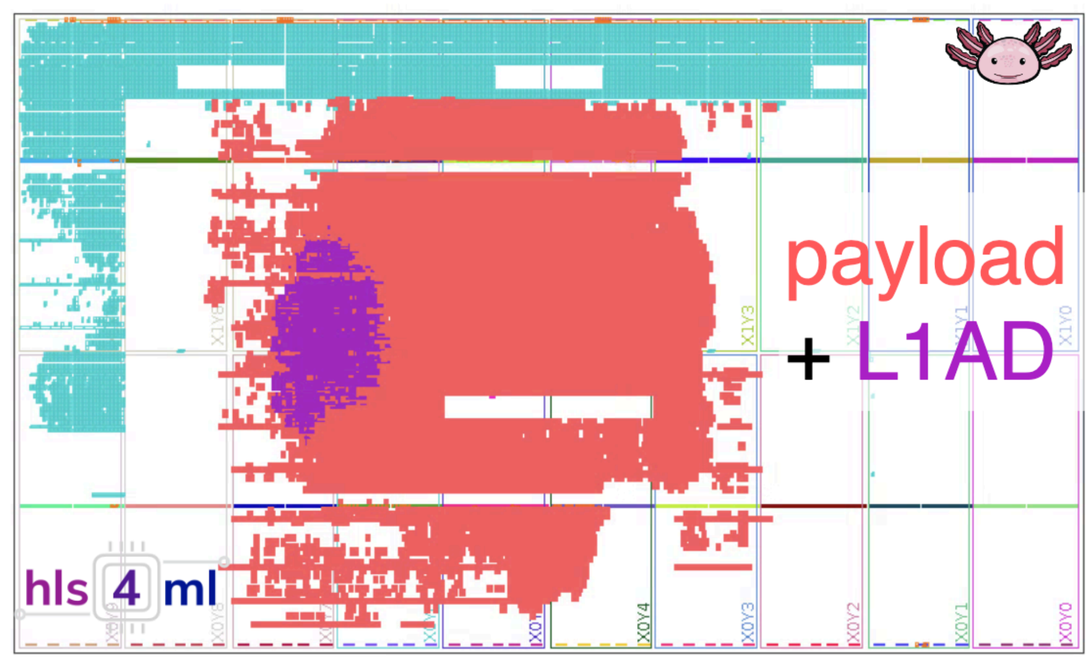
[CMS-DP-2023-079](#)
[CMS-DP-2024-059](#)

Online since Spring '24!

Full analysis and interpretation of dataset ongoing… stay tuned!







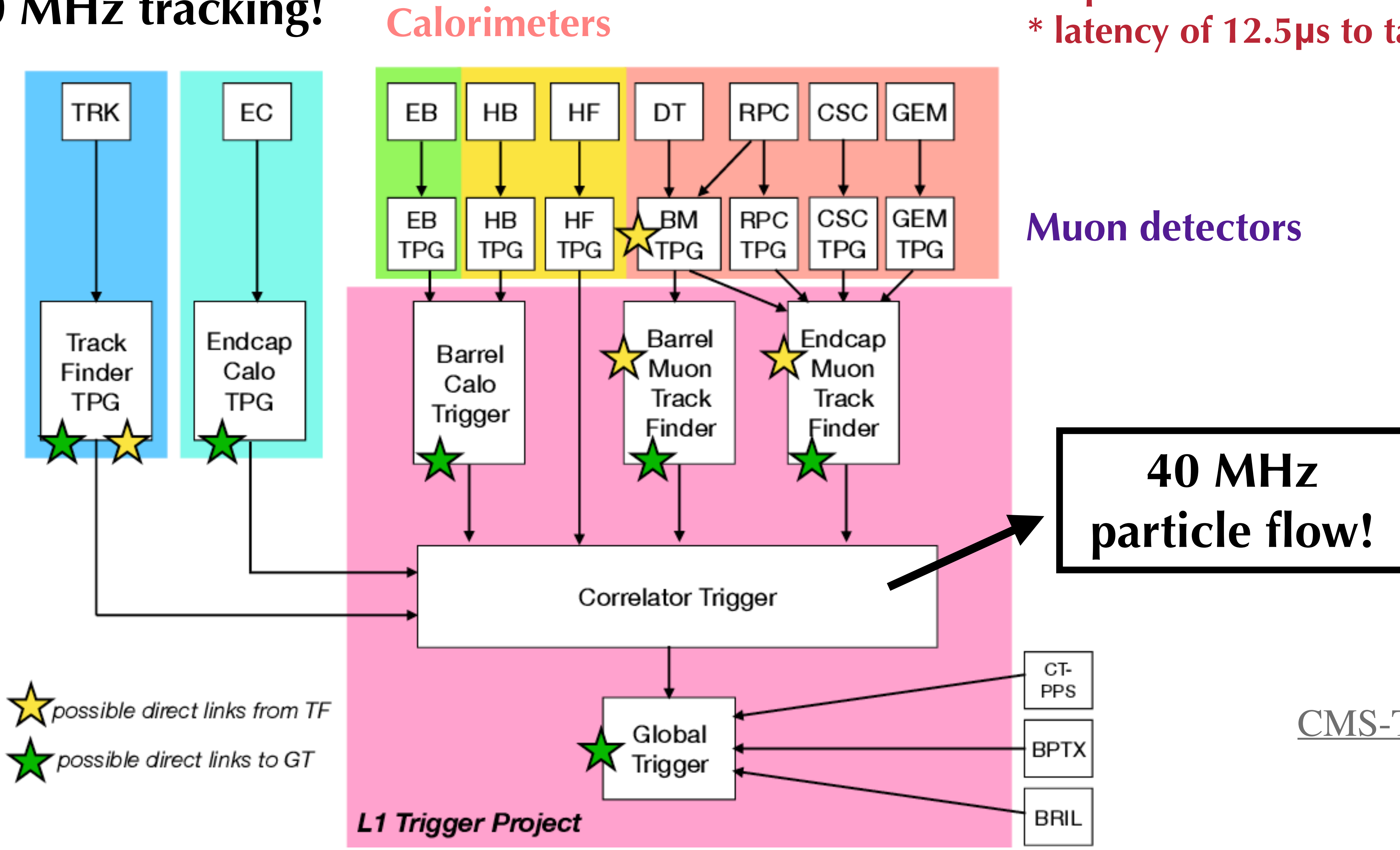
	Latency	LUTs	FFs	DSPs	BRAMs
AXOL1TL	2 ticks 50 ns	2.1%	~0	0	0

The HL-LHC challenge: CMS Phase 2

At HL-LHC, up to 200 pile-up interactions: *CMS is upgrading the L1T and HLT to enable the same physics program we are doing now (at @60 PU)*

40 MHz tracking!

* **input data from 2 Tb/s to 63 Tb/s**
* **latency of 12.5μs to take decision**



CMS-TDR-021

The HL-LHC challenge: CMS Phase 2

At HL-LHC, up to 200 pile-up interactions: *CMS is upgrading the L1T and HLT to enable the same physics program we are doing now (at @60 PU)*

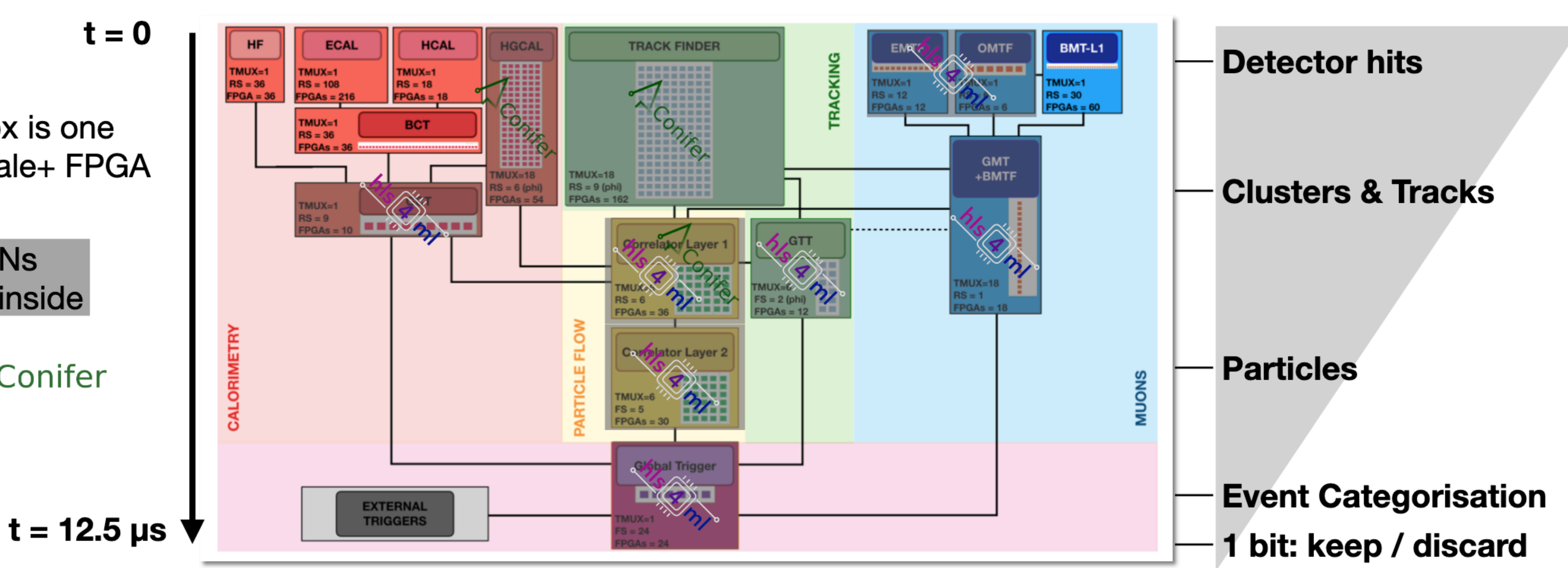
With significantly more powerful compute we expect ML to be well embedded into L1T to exploit higher information granularity:

Around 20 projects (NNs, BDTs) in development accounting for 25 billion ML inferences per second

Each small box is one Xilinx Ultrascale+ FPGA

These have NNs and/or BDTs inside

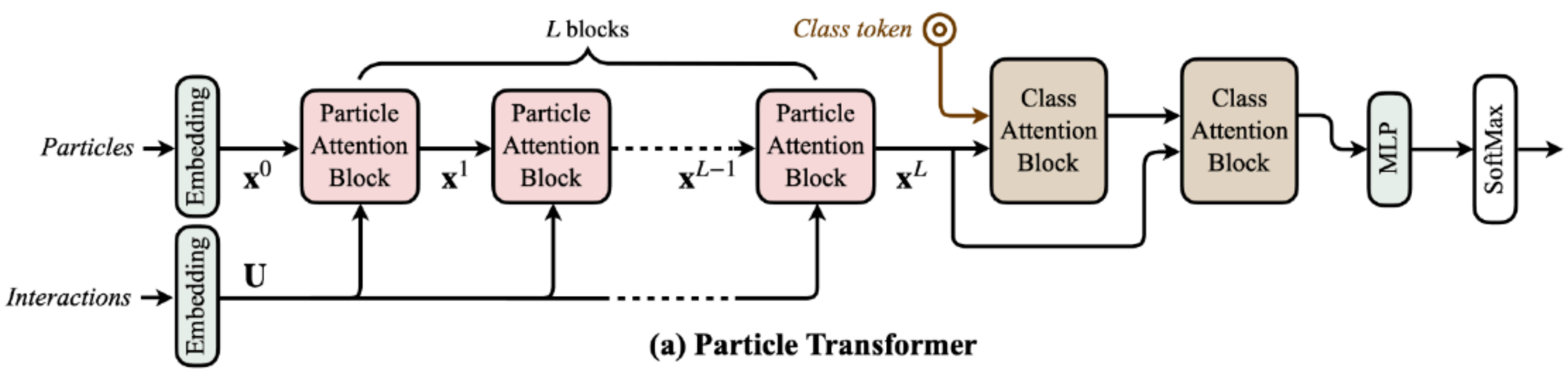
hls4ml Conifer



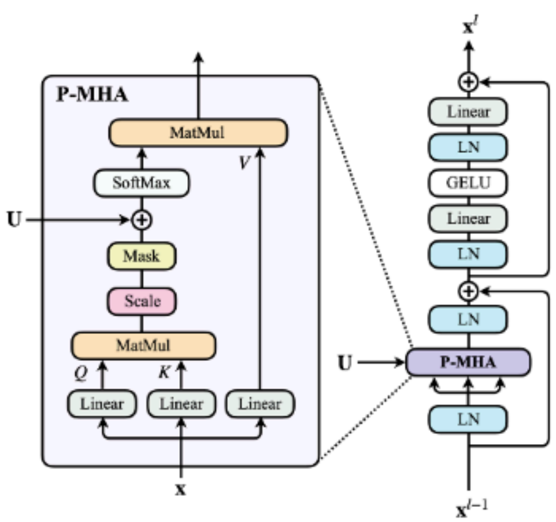
Credits: Sioni Summers (CERN)

Finding the best NN architecture

- **At offline level: chose the architecture with highest accuracy even if not efficient…**
- Current SOA is particle-based transformers — ***learn which neighbour particles are relevant through attention mechanism***
  - input embedding of both single particle and pair-wise features information
  - the pair-wise features encode physics principles → modifiers of standard dot-product attention weights in **Particle Attention Block**



(a) Particle Transformer



(b) Particle Attention Block

$$\text{P-MHA}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d_k} + \mathbf{U})V$$

d_k: dimension of K

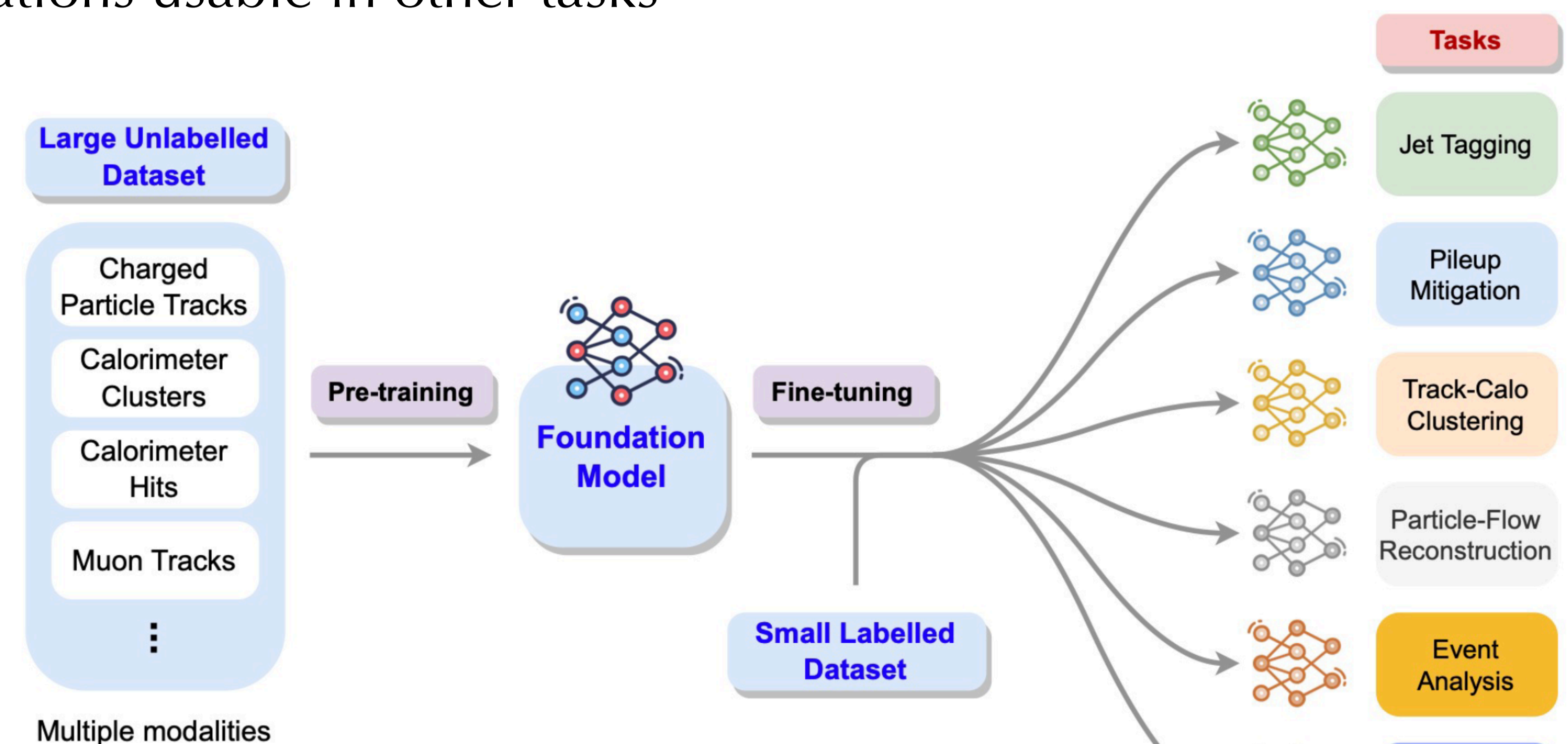
Choice of the pair-wise features: from LundNet

$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2}$$
$$k_\mathrm{T} = \min(p_{\mathrm{T},a}, p_{\mathrm{T},b}) \cdot \Delta$$
$$z = \min(p_{\mathrm{T},a}, p_{\mathrm{T},b}) / (p_{\mathrm{T},a} + p_{\mathrm{T},b})$$
$$m^2 = (E_a + E_b)^2 - \|\mathbf{p}_a + \mathbf{p}_b\|^2$$

H. Qu et al.: arXiv.2202.03772

Towards foundation models in HEP

- A foundation model is a large ML model trained on a vast quantity of data such that it can be adapted to a wide range of downstream tasks (e.g., BERT, GPT, …)
    - **self-supervised learning:** use the data itself to create training objective
    - **outputs:** powerful representations usable in other tasks



from M. Kagan at H&N 2023

reusable — one backbone used for several tasks

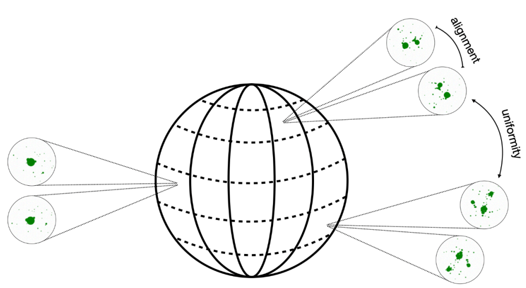
train on huge real data – leverage experimental data

leverage multi-modal methods – combine data from different detectors to address more complex tasks

uncertainty reduction – reduce dependence on simulation-based training

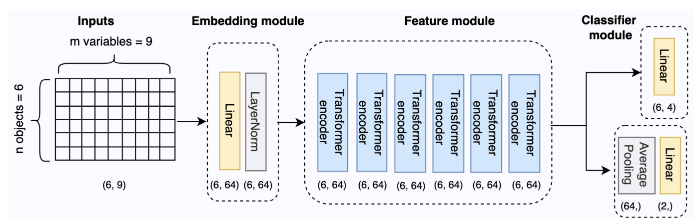
Towards foundation models in HEP

Contrastive Learning: Symmetry Augmentation



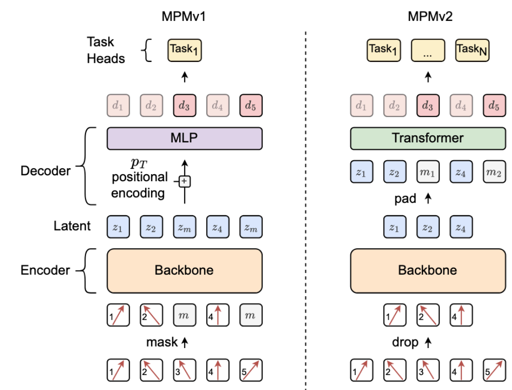
Dillon, Kasieczka, Olischlager
Plehn, Sorrenson, Vogel, [2108.04253](https://arxiv.org/abs/2108.04253)

Masked Particle Type Prediction



Kishimoto, Morinaga, Saito
Tanaka, [2312.06909](https://arxiv.org/abs/2312.06909)

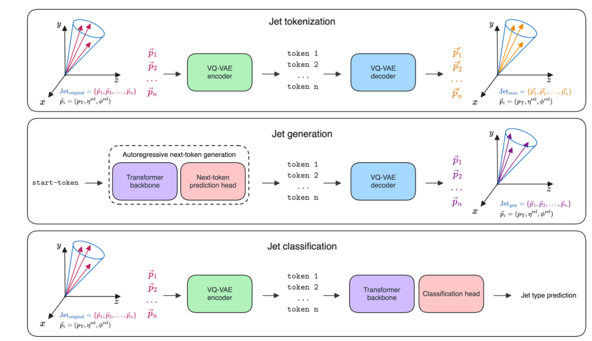
Masked Particle Modeling



Golling, Heinrich, Kagan, Klein,
Leigh, Osadchy, Raine,
[2401.13537](https://arxiv.org/abs/2401.13537)

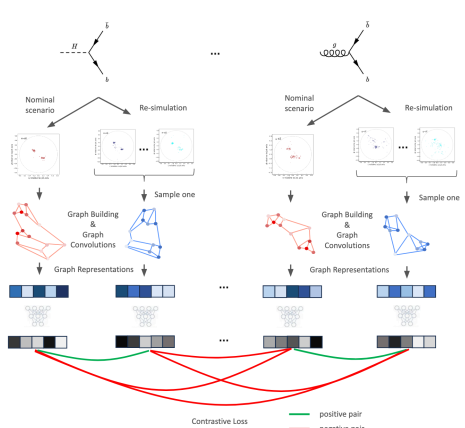
Leigh, Klein, Charton, Golling,
Heinrich, Kagan, Ochoa, Osadchy,
[2409.12589](https://arxiv.org/abs/2409.12589)

Next Token Prediction



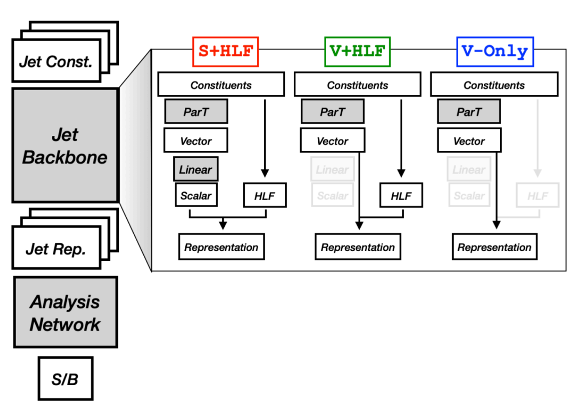
Birk, Hallin, Kasieczka, [2403.05618](https://arxiv.org/abs/2403.05618)

Contrastive Learning: Re-Simulation



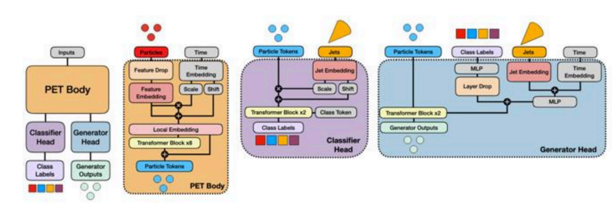
Harris, MK, Krupa, Maier, Woodward,
[2403.07066](https://arxiv.org/abs/2403.07066)

Supervised Pre-training and Joint Optimization



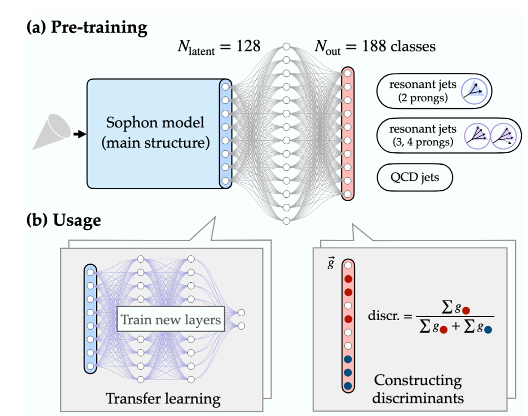
Vigl, Hartman, Heinrich, [2401.13536](https://arxiv.org/abs/2401.13536)

Supervised Classification and Generation



Mikuni, Nachman [2404.16091](https://arxiv.org/abs/2404.16091)

Large-Scale Fine-Grained Classification

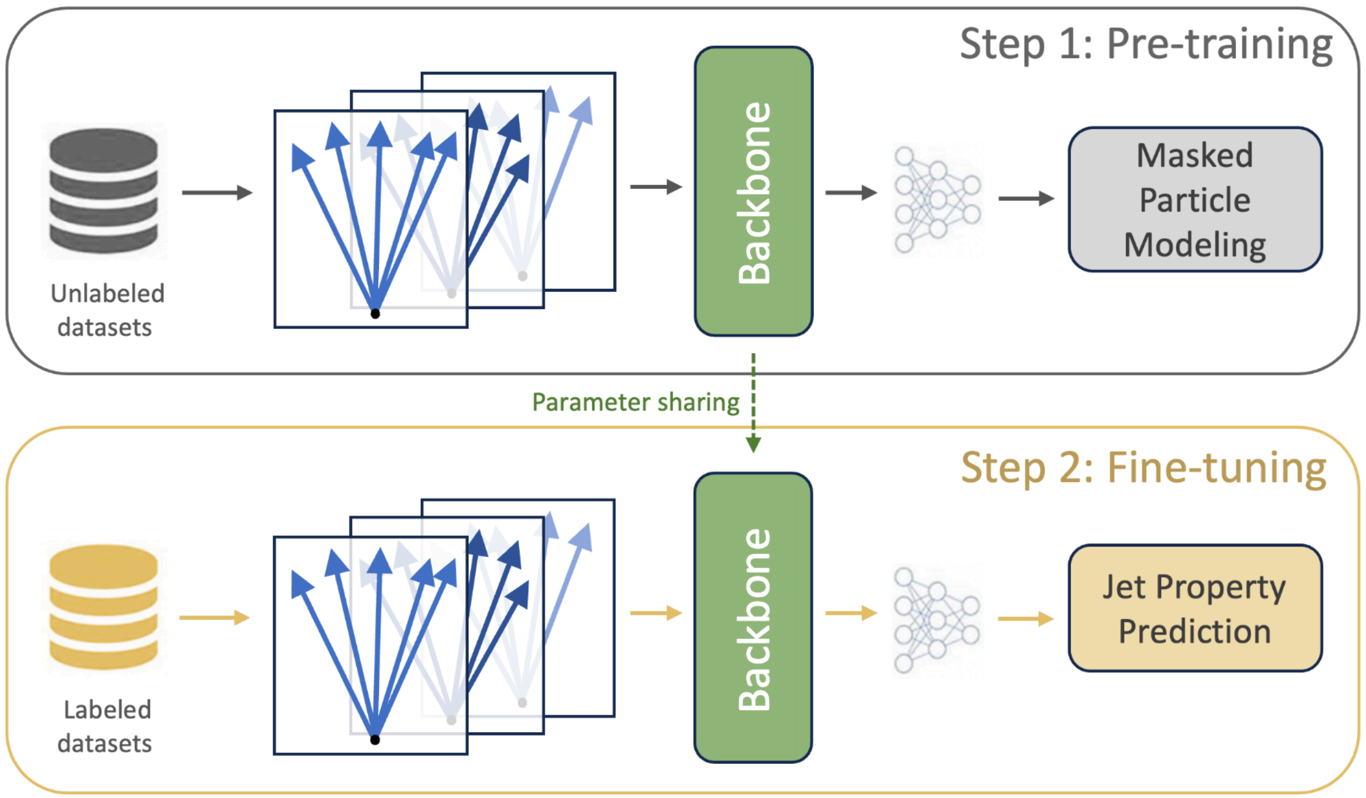


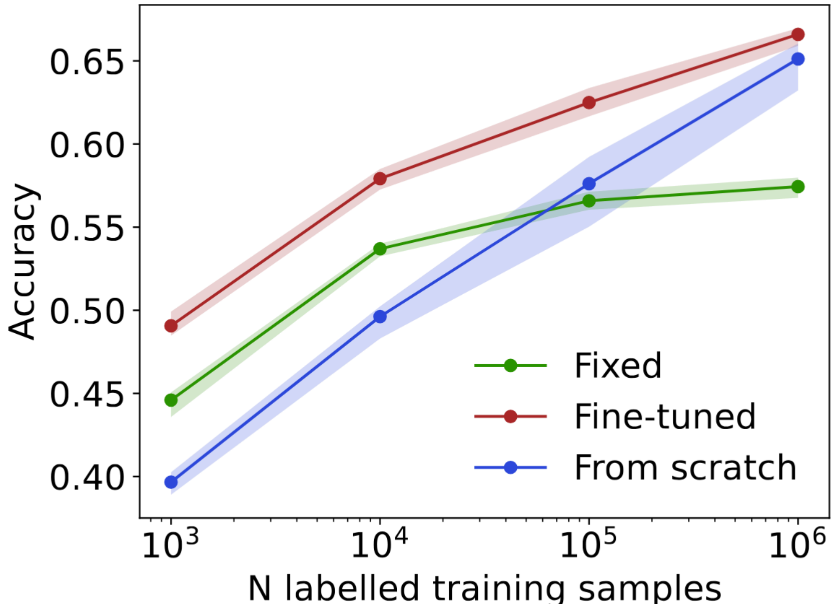
Li, Li, et al. [2405.12972](https://arxiv.org/abs/2405.12972)

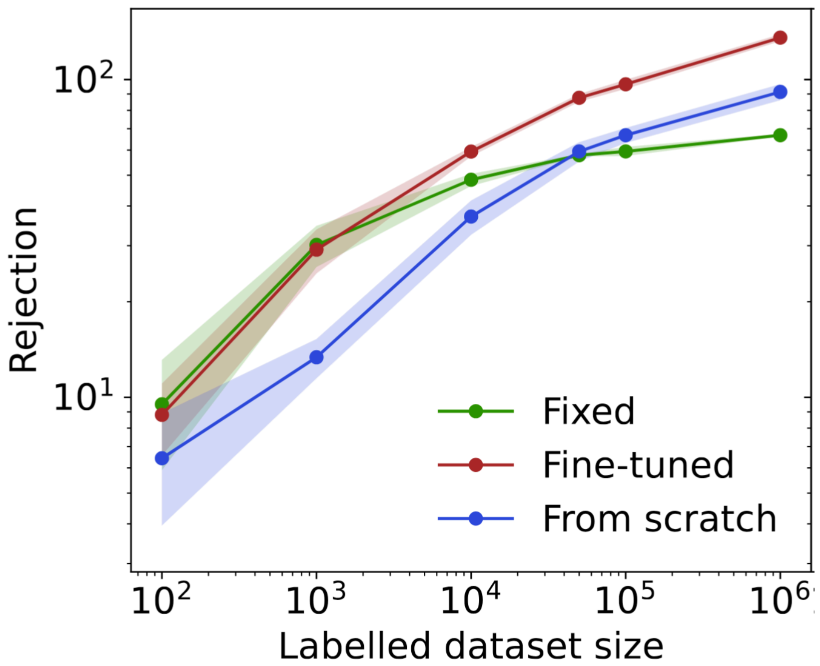
Towards foundation models in HEP

Golling, Heinrich, Kagan, Klein, Leigh, Osadchy, Raine, 2401.13537

- Two steps training:
  - Pre-training gives better performance and can use less data on downstream Tasks
  - Also shown better domain adaptation properties → robustness

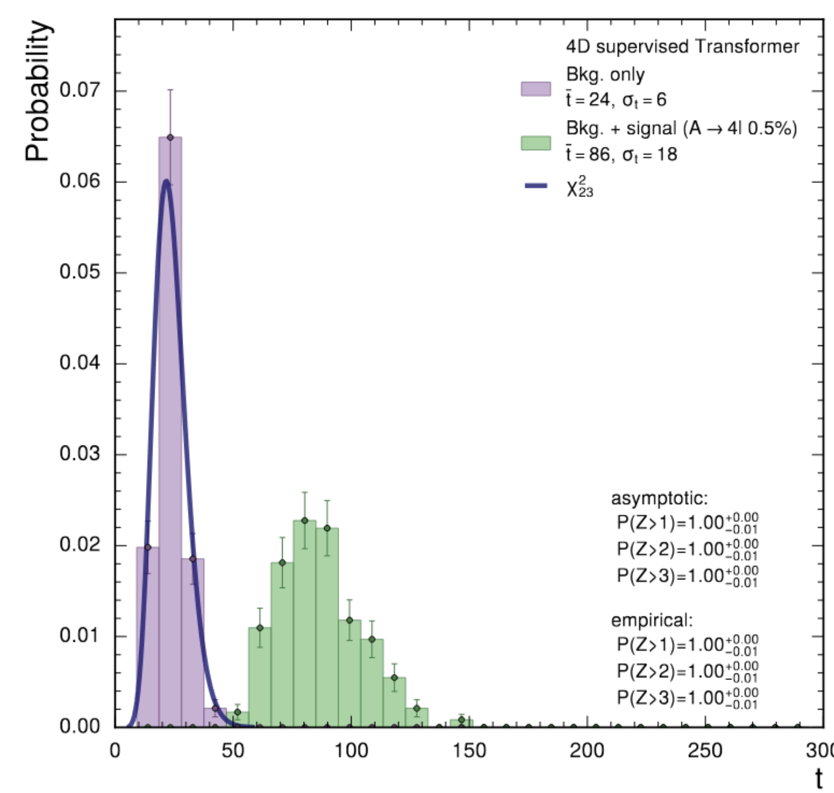




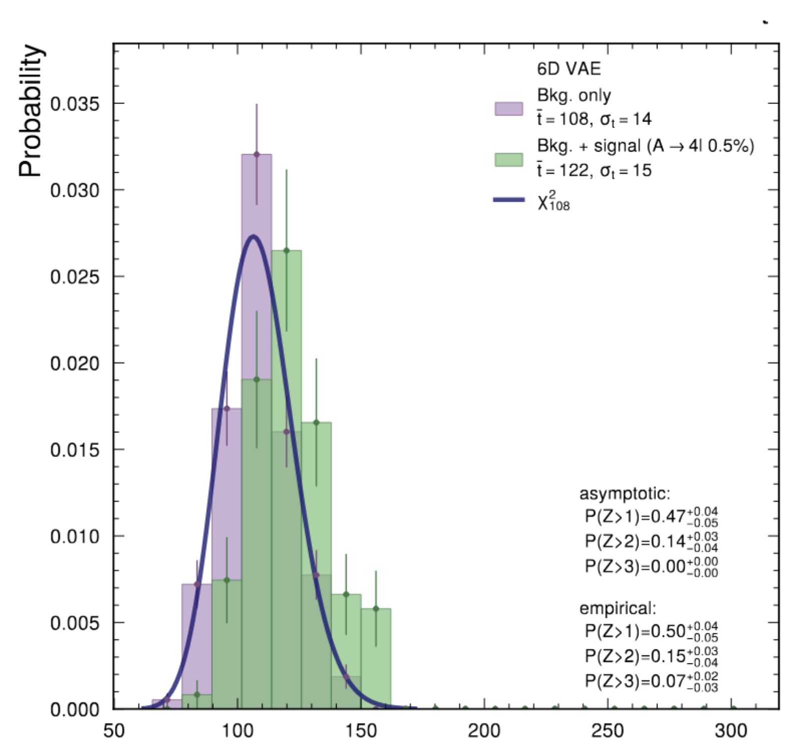


Pre-training models for anomaly detection

- **Contrastive learning** is a self-supervised learning (SSL) technique that aims to learn representations by comparing similar and dissimilar samples (called "augmentations")
  - can learn powerful latent representation for anomaly detection
- Downstream: usual unsupervised clustering techniques (e.g., autoencoders)
- Two approaches:
  - self supervised: augmentation by masking
  - supervised: augmentation by label



Sample name	Number of samples	Type
SM processes[23]	4,000,000	B
$LQ \to b\tau$[24]	340,544	S
$A \to 4\ell$[25]	55,969	S
$h^0 \to \tau\tau$[26]	691,283	S
$h^\pm \to \tau\nu$[27]	760,272	S
blackbox[28]	4,210,492	S+B



[2502.15926](https://arxiv.org/abs/2502.15926)

Finding the best NN architecture

- Many offline applications moving to SOA **transformer architectures** → not trivial mapping of MHA to FPGA circuit
  - attention map requires N^2 computations
  - softmax in those computation is slow and expensive
  - large weights matrices easily saturate memory
- First vanilla solutions for HEP being explored recently
- Expect more R&D in this direction in the near future
  - e.g., alternative architectures, aggressive quantization and pruning of weights and/or attention scores, state space models, …

Wayne Luk, et al.

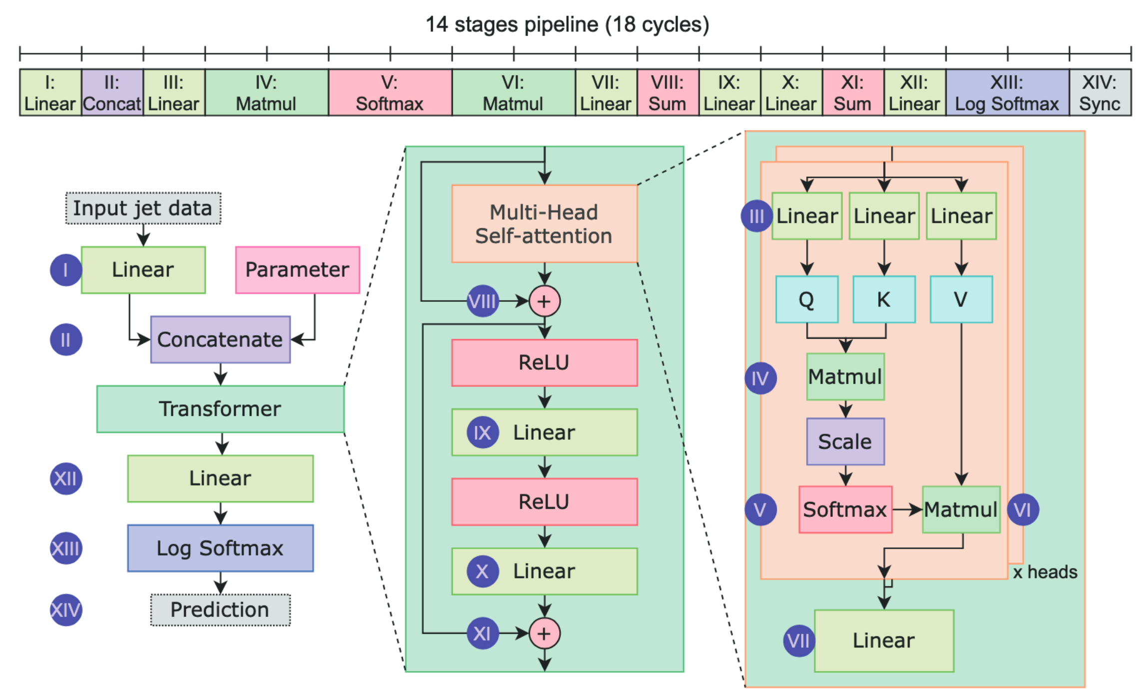


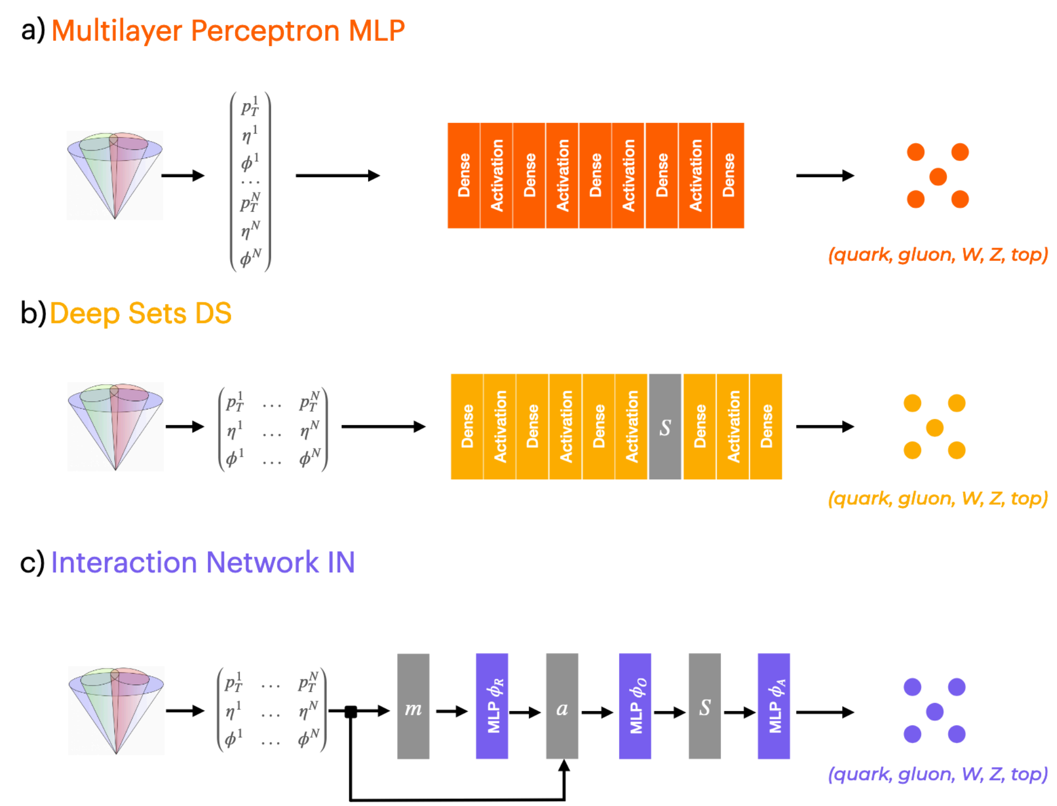
TABLE III
FPGA RESOURCES UTILIZATION

	BRAM 18K	DSP48E	FF	LUT
Total used	12	4,351	58,942	298,881
Available	5,376	12,288	3,456,000	1,728,000
Utilization	0.22%	35.41%	1.71%	17.30%

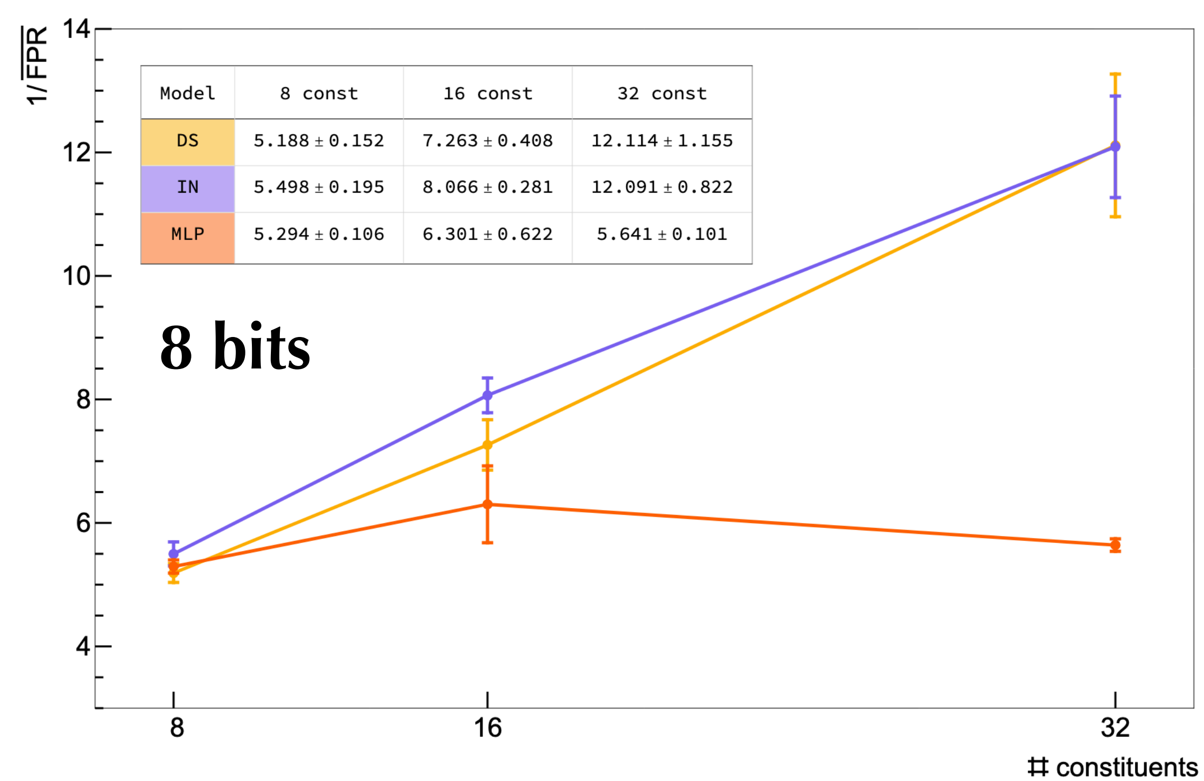
90 ns inference time
73% accuracy on jet tagging
16 jet level features

Finding the best NN architecture

- **At offline level: chose the architecture with highest accuracy even if not efficient…**
- For edge applications this is not an option: crucial to **co-design the architecture with the application and its constraints**



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



Graph NNs at O(100) ns latency!

FPGA: Xilinx Virtex UltraScale+ VU13P

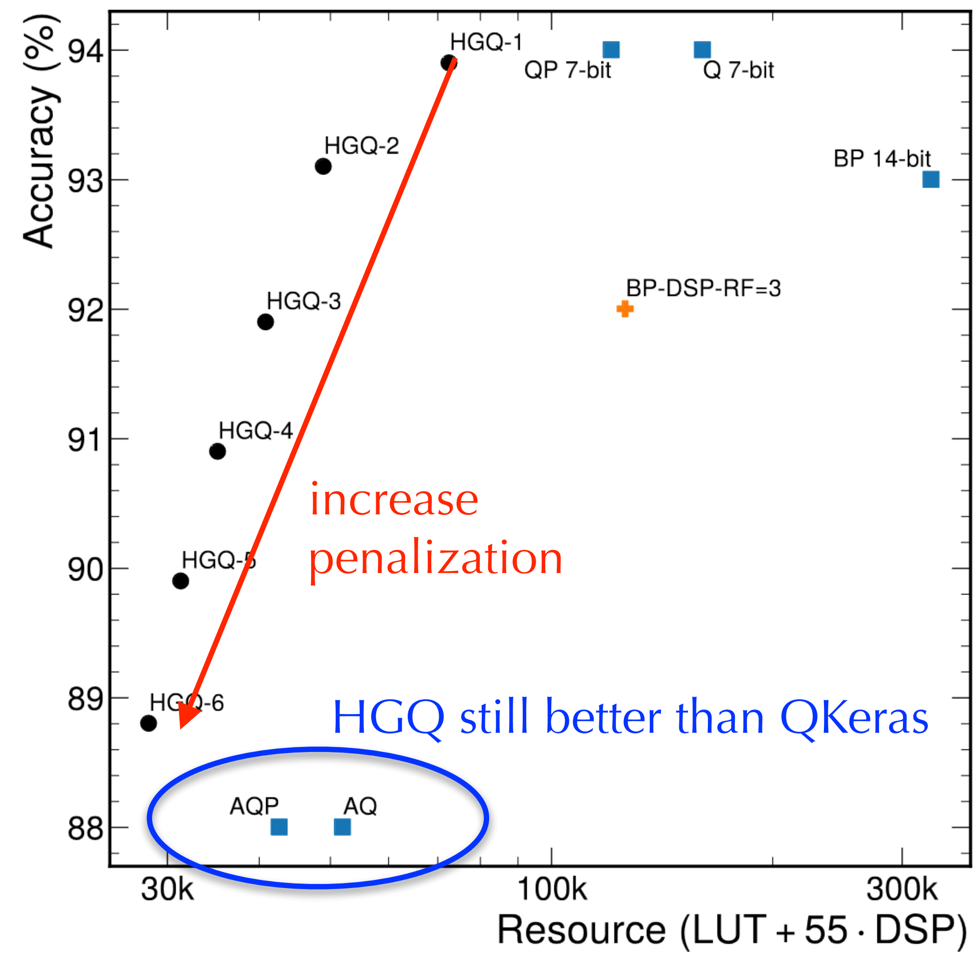
Architecture	Constituents	RF	Latency [ns] (cc)	II [ns] (cc)	DSP	LUT	FF	BRAM18
MLP	8	1	105 (21)	5 (1)	262 (2.1%)	155,080 (9.0%)	25,714 (0.7%)	4 (0.1%)
	16	1	100 (20)	5 (1)	226 (1.8%)	146,515 (8.5%)	31,426 (0.9%)	4 (0.1%)
	32[a]	1	105 (21)	5 (1)	262 (2.1%)	155,080 (7.2%)	25,714 (0.7%)	4 (0.1%)
DS	8	2	95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

hls4ml

More aggressive quantization

- **Solution: optimize the individual bitwidths alongside the NN accuracy using gradient descent**

- **How:**
  - treat the bitwidths as continuous variables
  - introduce surrogate gradients for discrete variables such as bitwidths
  - introduce a novel on-chip resource consumption metric that when incorporated into the loss function penalizes larger bitwidths efficiently
  - pruning integrated naturally in the optimization step (gradient descent reduces certain bitwidths to zero)



Gradient-based Automatic Mixed Precision Quantization for Neural Networks On-Chip

Chang Sun,[1,2,*] Thea K. Årrestad,[1] Vladimir Loncar,[3,4] Jennifer Ngadiuba,[5] and Maria Spiropulu[2]

[1] *ETH Zurich (Zurich, Switzerland)*
[2] *California Institute of Technology (CA, USA)*
[3] *Massachusetts Institute of Technology (MA, USA)*
[4] *Institute of Physics Belgrade (Belgrade, Serbia)*
[5] *Fermi National Accelerator Laboratory (IL, USA)*

Fully supported in hls4ml!

Fermilab

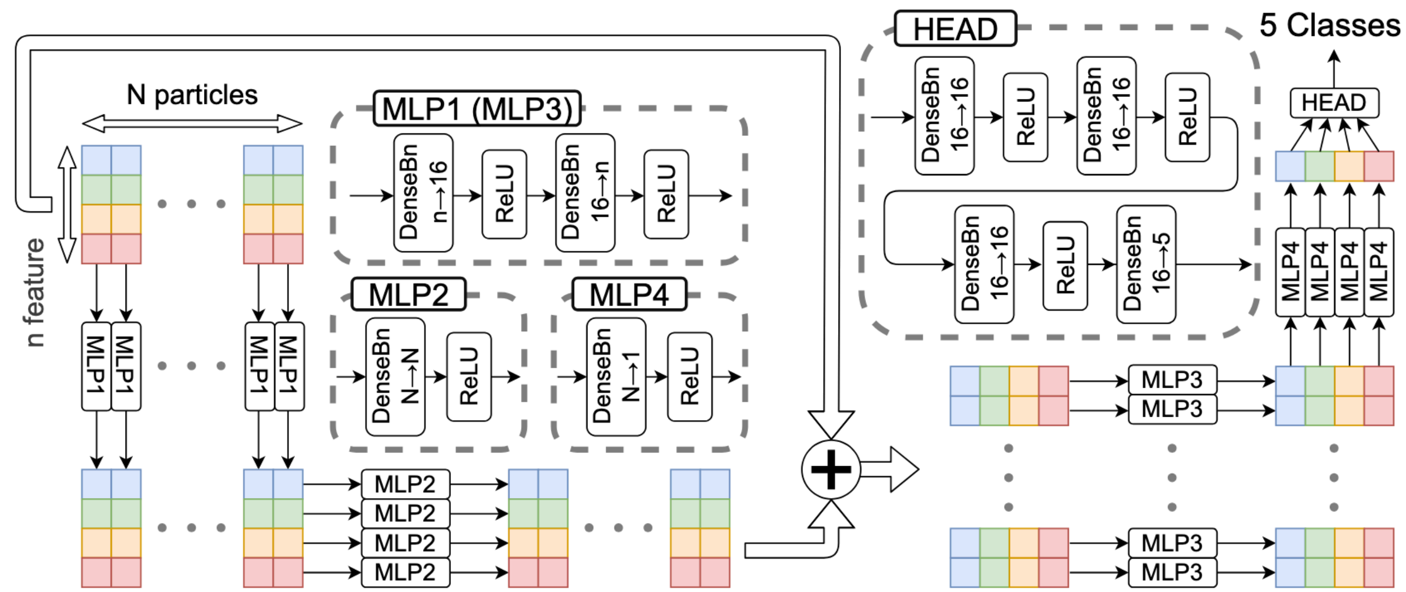
ETH zürich

Paper on arxiv ready for submission!

Finding the best NN architecture

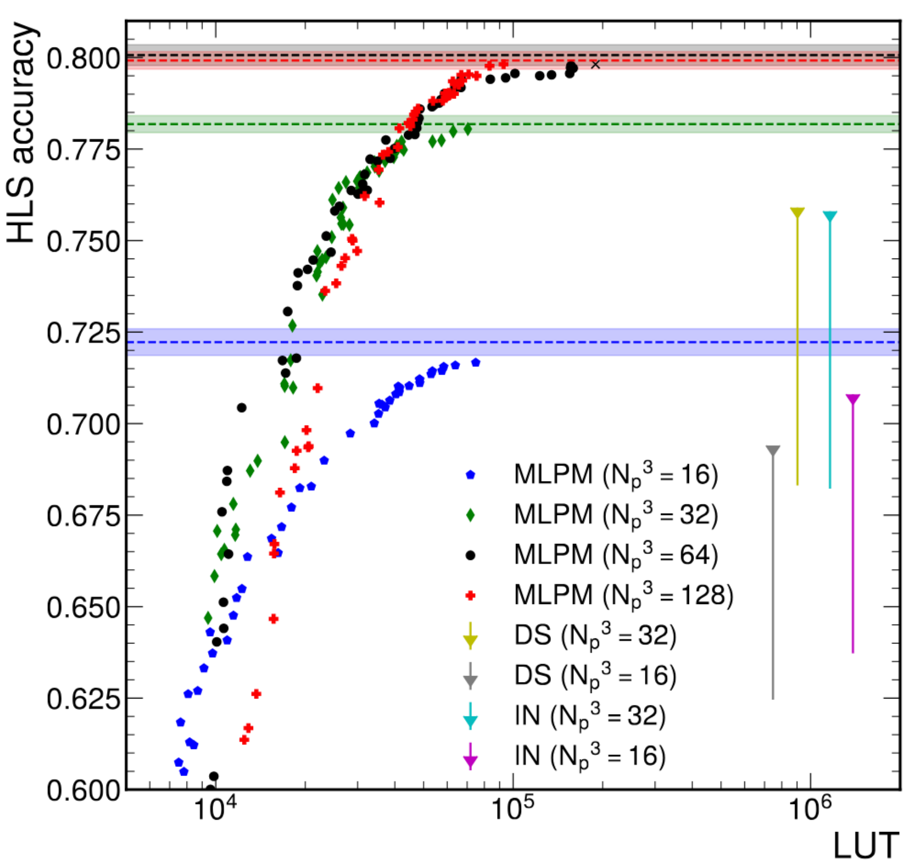
2503.03103

- Also found more recently that **MLP-mixer architecture** can further enhance performance



- Processes sets (like particle clouds or jet constituents) using only MLPs and alternates between two key components:
  - Token-mixing MLP
    – mixes information across particles (rows),
  - Channel-mixing MLP
    – mixes features within each particle (columns)

- **Not naturally permutation-invariant**
  → not necessarily needed for ordered sets (as in the trigger)
  → enables it to learn which features to retain and which to discard, facilitating HGQ



Finding the best NN architecture

2503.03103

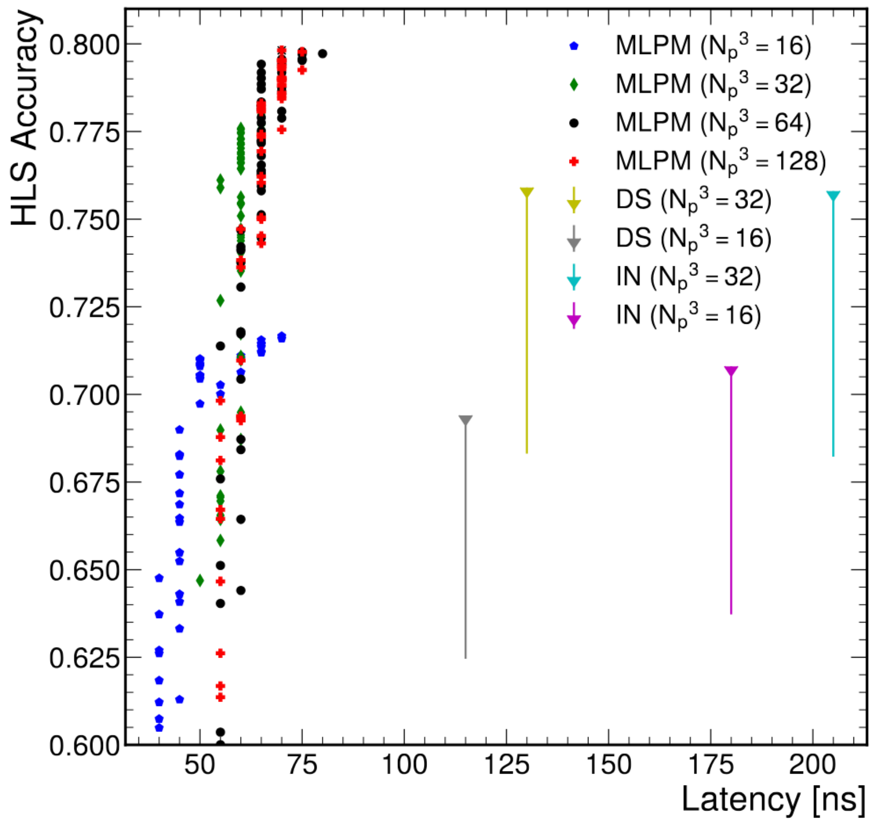
- Also found more recently that **MLP-mixer architecture** can further enhance performance

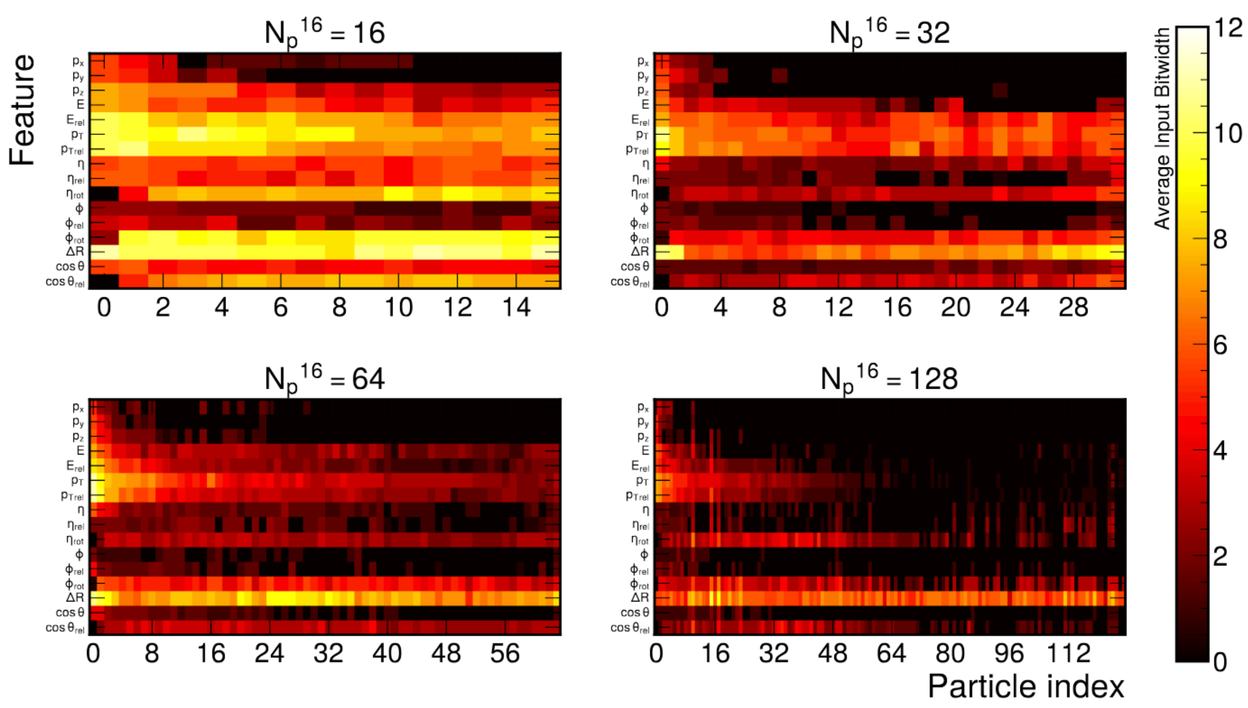
- Processes sets (like particle clouds or jet constituents) using only MLPs and alternates between two key components:

  - Token-mixing MLP
    – mixes information across particles (rows),

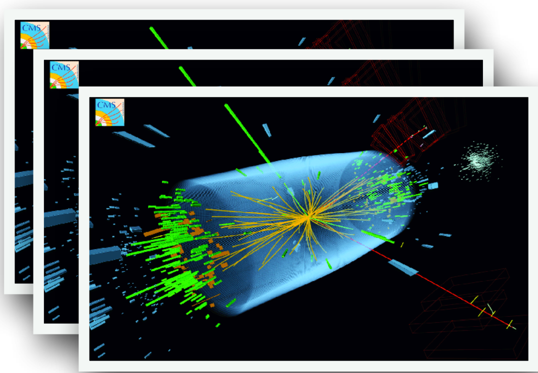
  - Channel-mixing MLP
    – mixes features within each particle (columns)

- **Not naturally permutation-invariant**
  → not necessarily needed for ordered sets (as in the trigger)
  → enables it to learn which features to retain and which to discard, facilitating HGQ





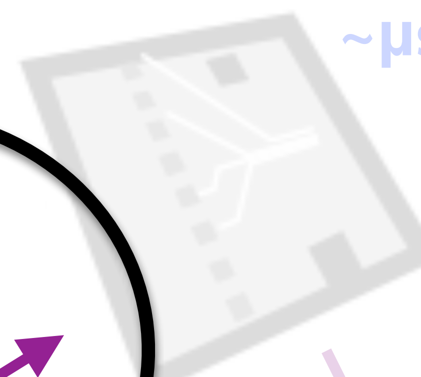
AI @ Extreme Edge



CMS Experiment
40 MHz collision rate
~1B detector channels

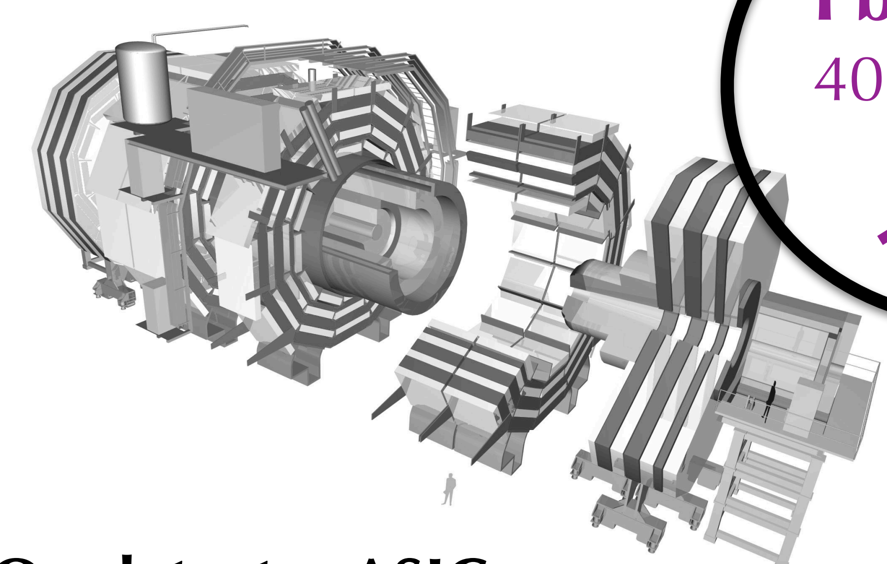
Worldwide
computing grid
Exabyte-scale
datasets

FPGA filter stack
~μs latency



Level-1
Trigger

Pb/s
40 MHz



ASICs typically used at the front end for sensors read out: directly embed ML in here to allow intelligent data compression at the very edge

100s kHz

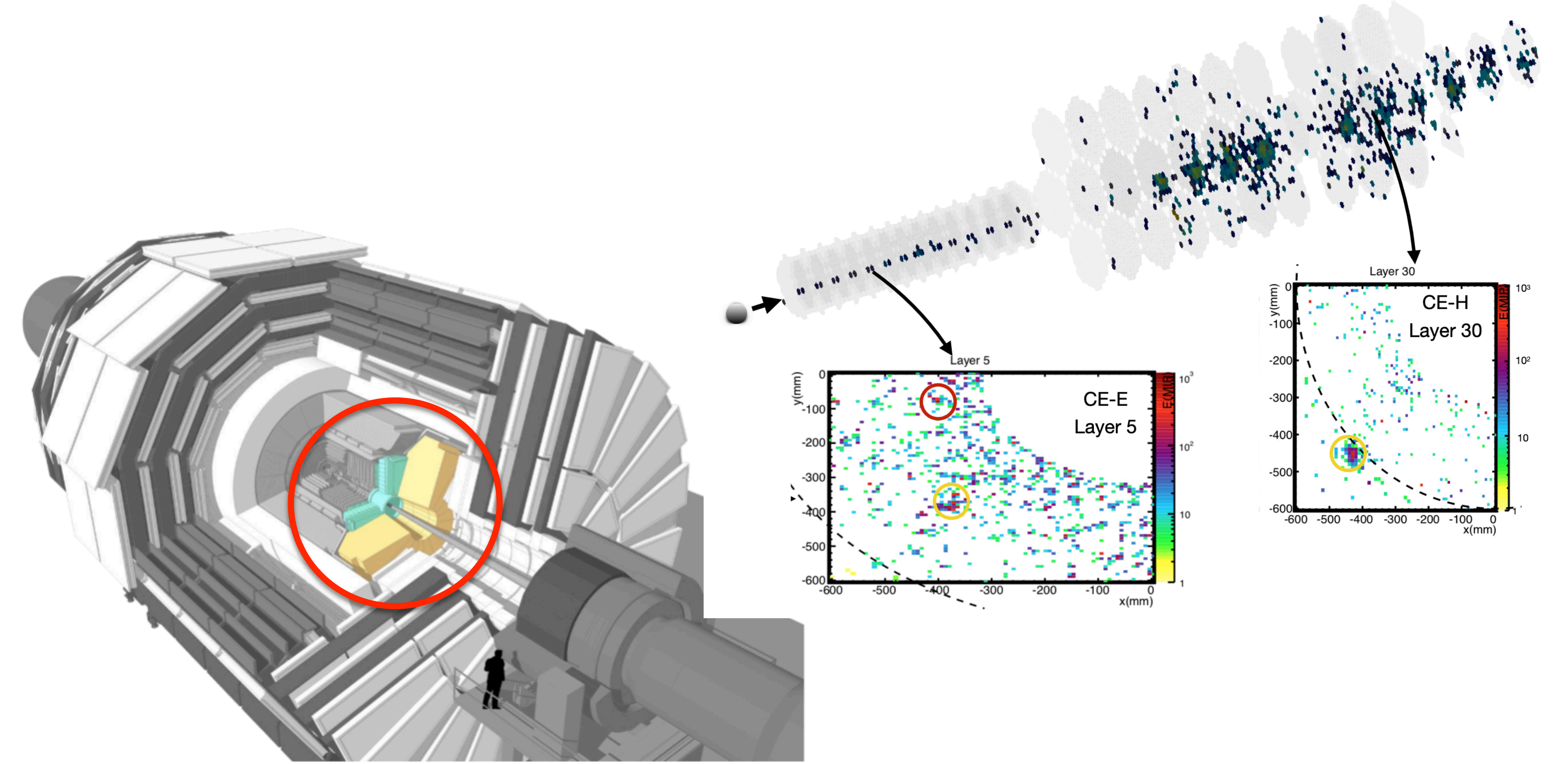
High-Level
Trigger

On-detector ASIC compression
~100 ns latency

On-prem CPU/GPU filter farm
~100 ms latency

Example:
High-granularity calorimeter @ HL-LHC

Novel technology for future CMS endcap calorimeter:
50 layers with unprecedented number of readout channels (6M)!

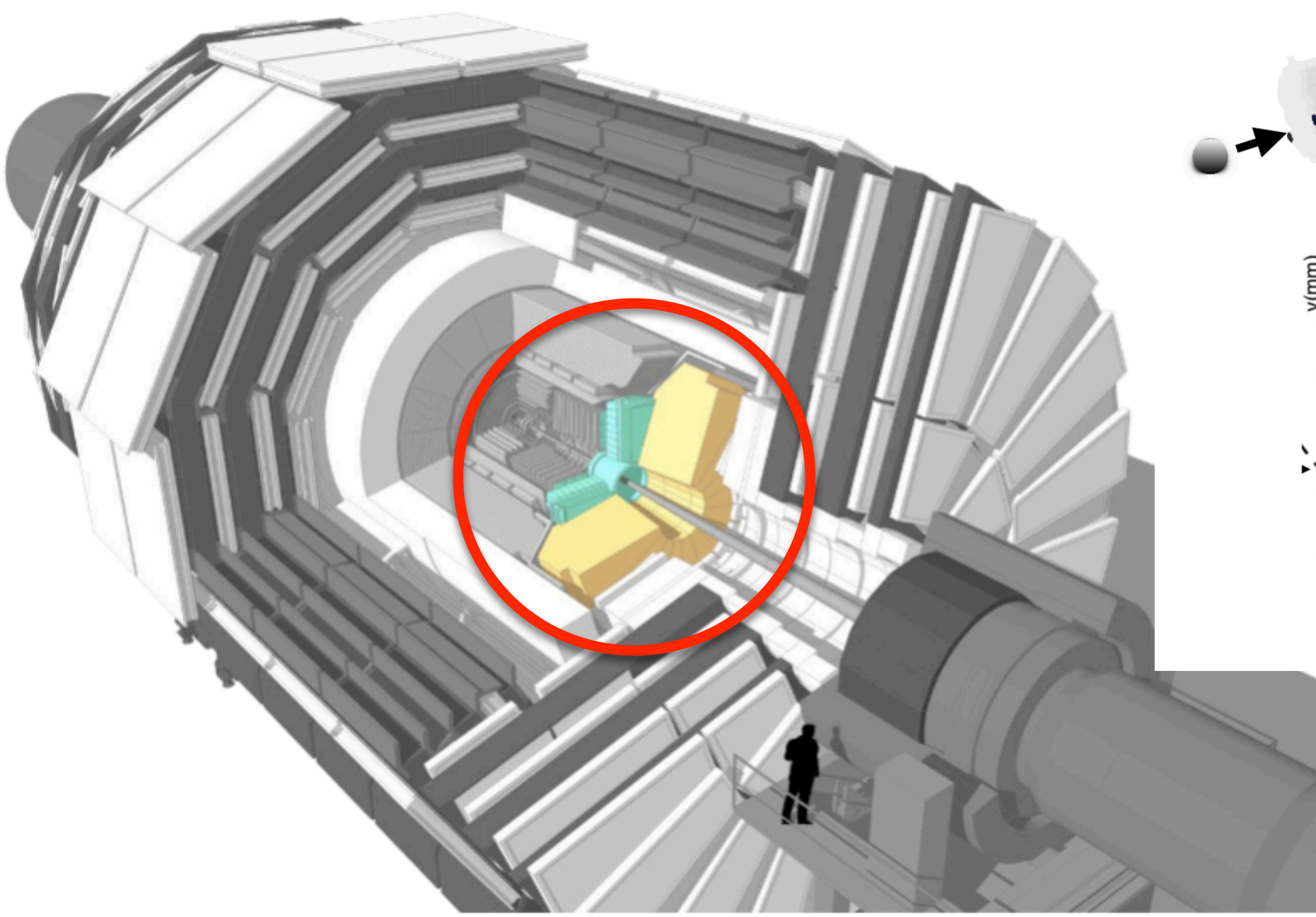


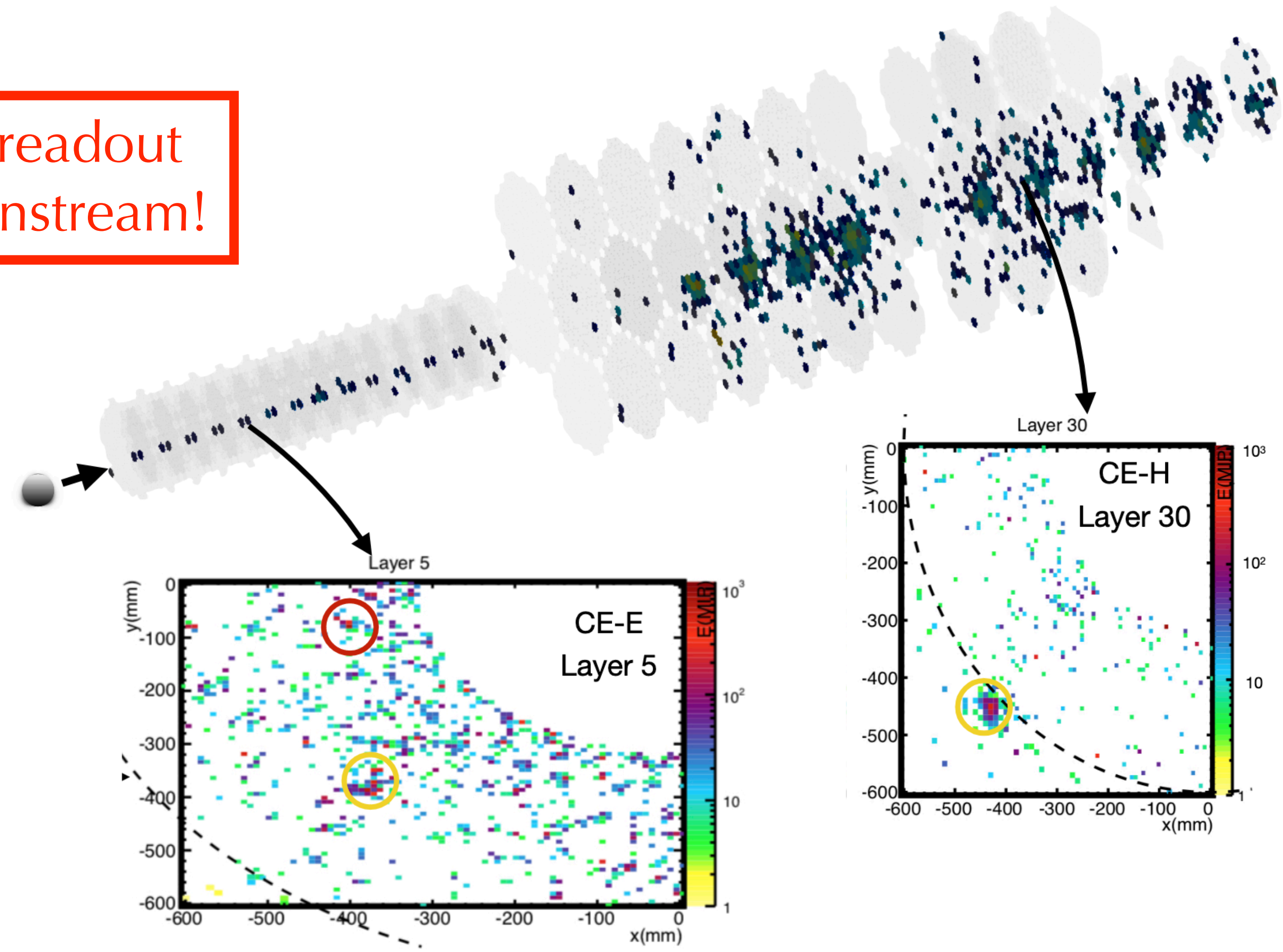
CMS HGCAL TDR

Example:
High-granularity calorimeter @ HL-LHC

Novel technology for future CMS endcap calorimeter:
50 layers with unprecedented number of readout channels (6M)!

Not enough bandwidth and latency to readout and put together all these channels downstream!

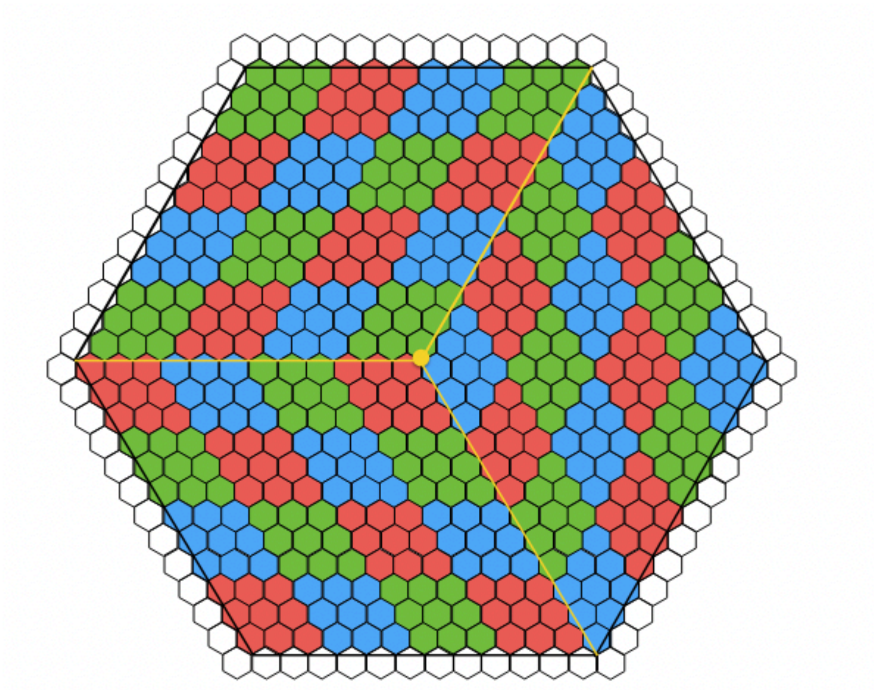




CMS HGCAL TDR

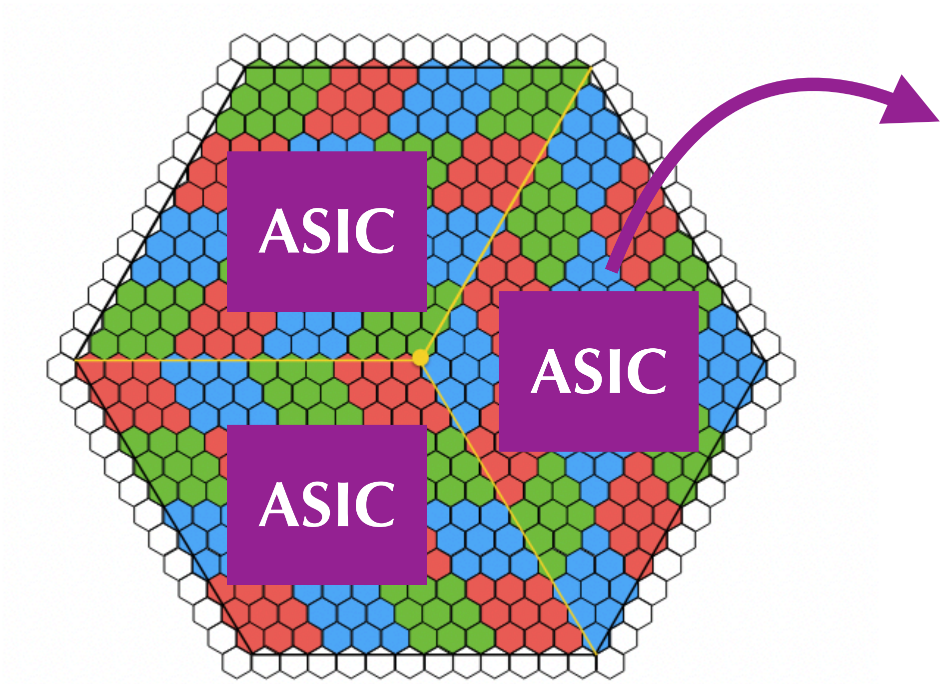
Example: CMS HG calorimeter

One module = 432 sensors

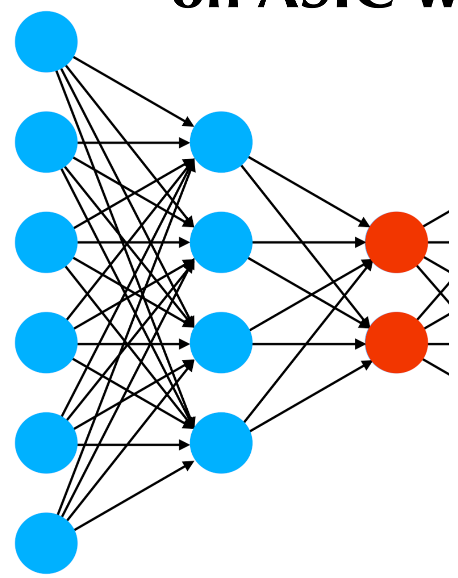


Example: CMS HG calorimeter

One module = 432 sensors



Encode to N bits on ASIC with NN



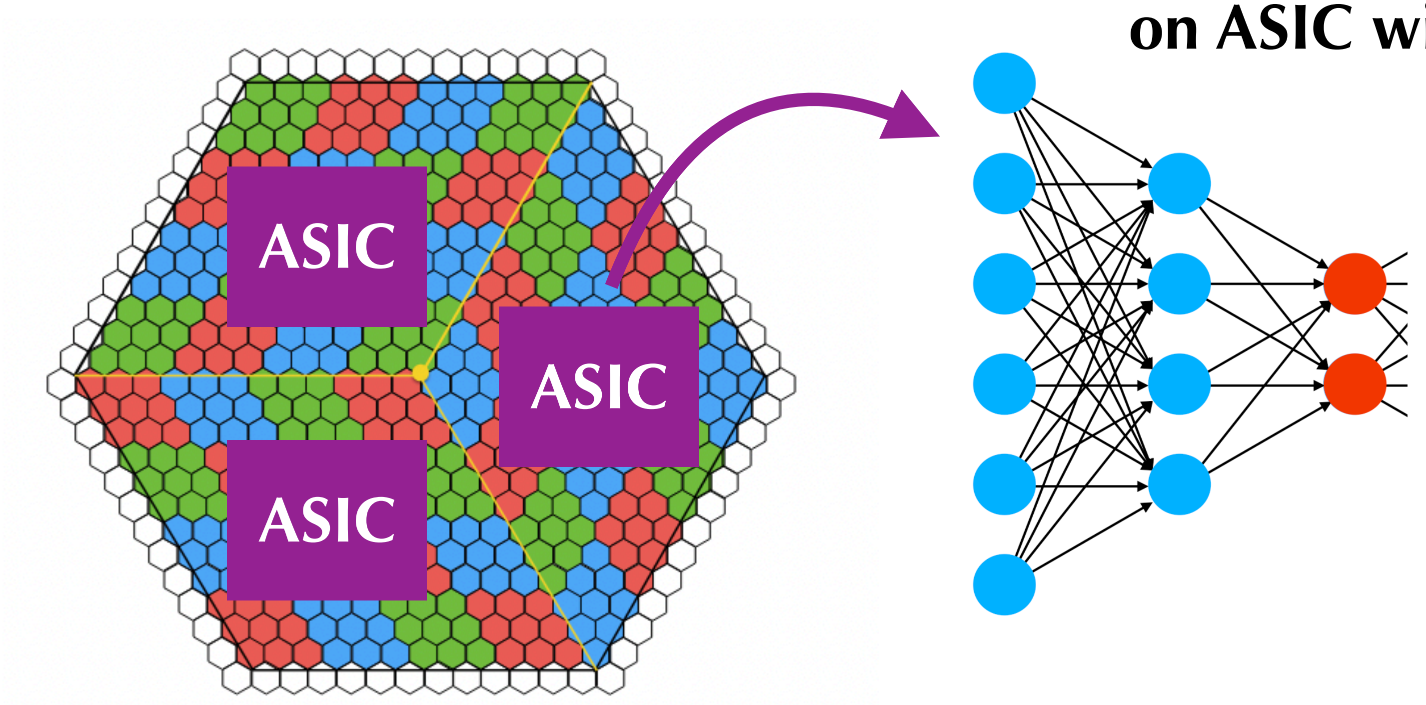
Compress data on sensor in ASIC:

High radiation
Cooled to -30 C → low power
O(100) ns latency

Example: CMS HG calorimeter

One module = 432 sensors

Encode to N bits on ASIC with NN

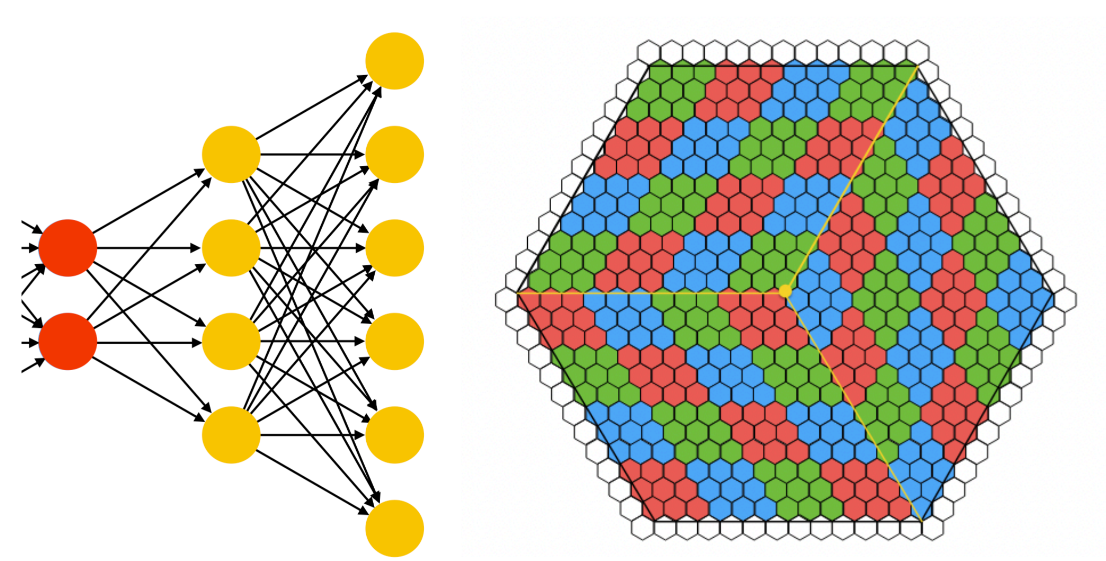


Transmit encoded data

Reconstruct or do latent space analysis on downstream processors (FPGAs)

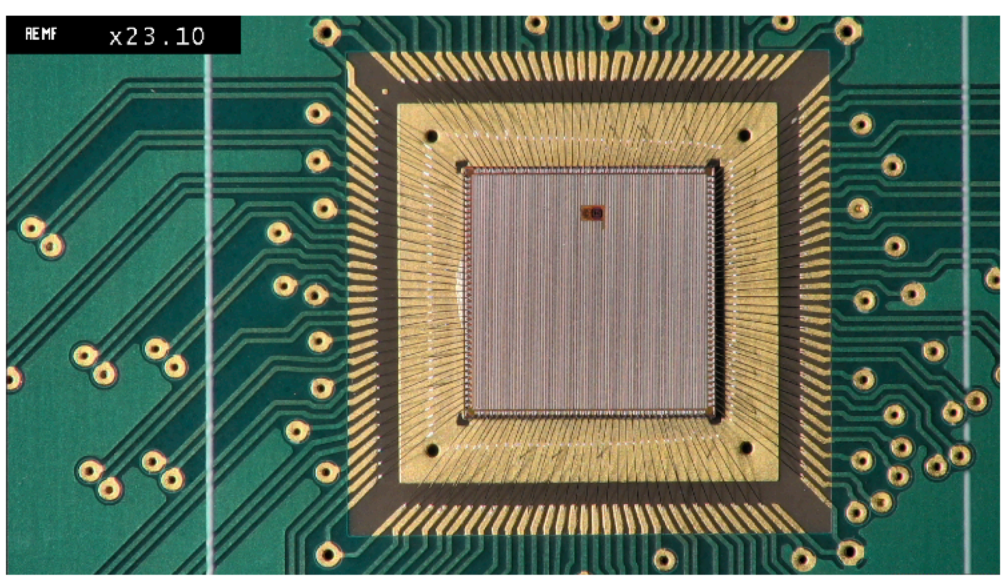
Compress data on sensor in ASIC:

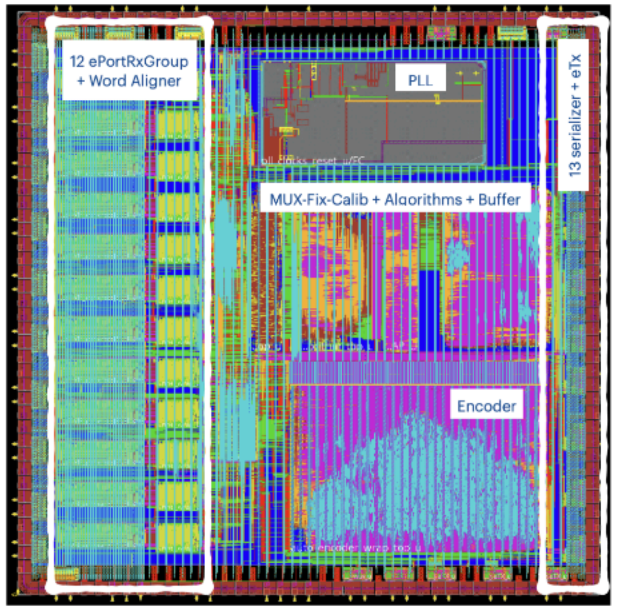
High radiation
Cooled to -30 C → low power
O(100) ns latency

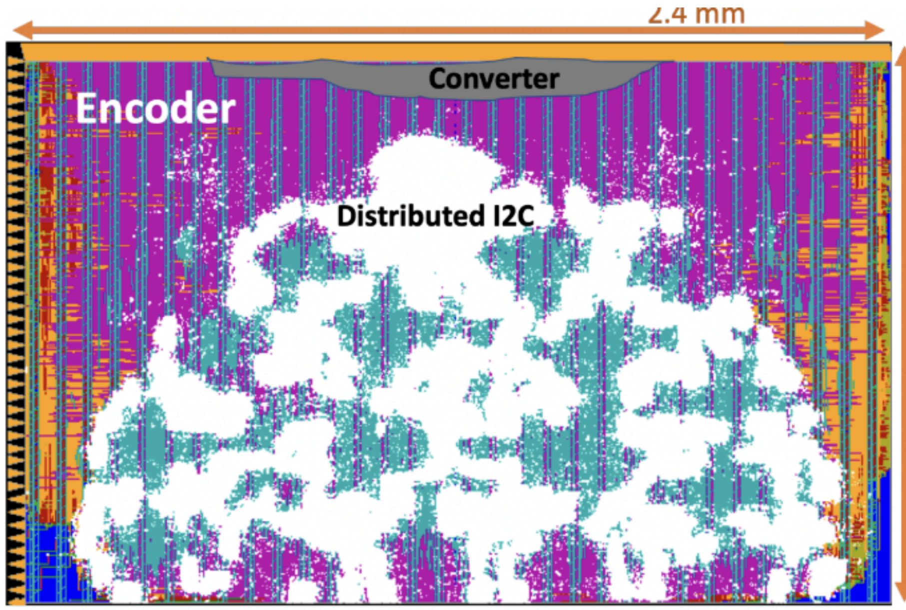


AI @ Extreme Edge: HGCAL

- **Tiny and heavily quantized NN —> low latency (50 ns) & low power (2.4 nJ/inf)**
- NN IP block created for the ASIC with **Catapult HLS (Mentor/Siemens) and hls4ml**
  - NN architecture is fixed, weights can be reprogrammed over I2C
  - NN parameters (weights and biases) triplicated for radiation tolerance → 200% overhead
- Developed in parallel a tool — FKeras — that performs **bit-level sensitivity study of each weight in the NN**
  - allows to prioritize which bits need protection and which may be safely disregarded, reducing resource overhead





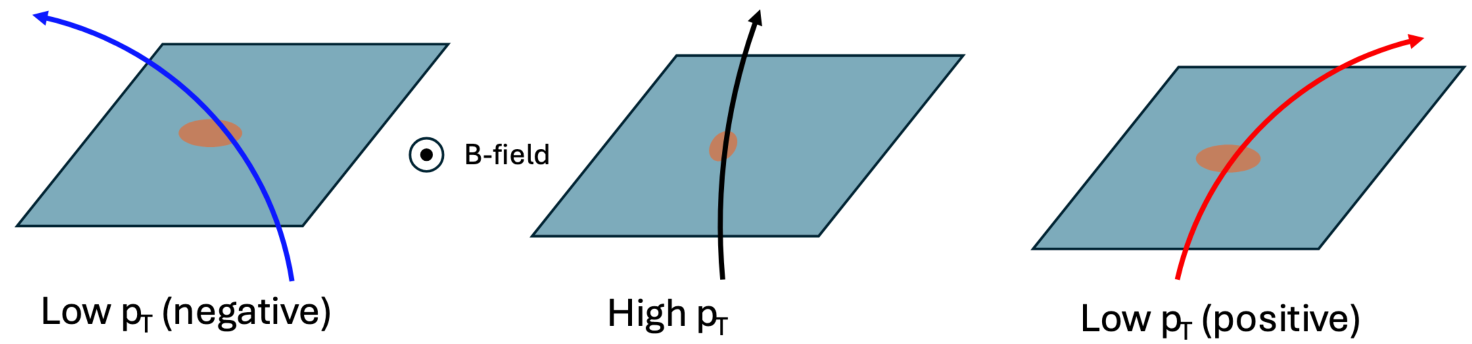


AI @ Extreme Edge: Smart Pixels

- **Challenge:** unable to read pixel detector for real time analysis because of **too high granularity**

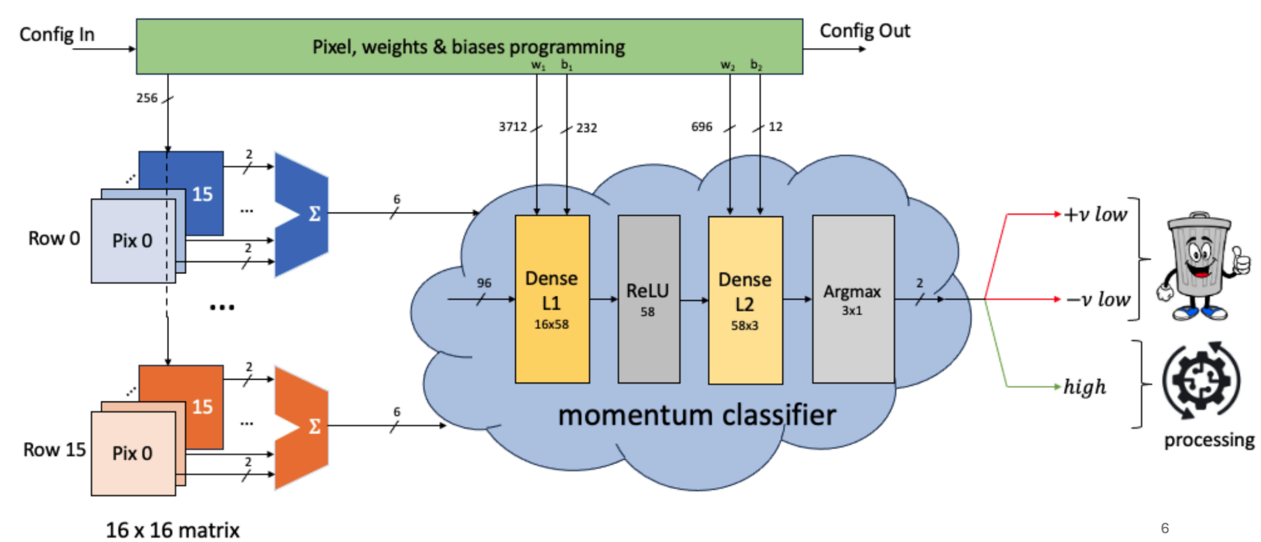
- **Solution:** use cluster shape to extract incident angle of particle traversing pixel sensors
→ distinguish low p_T from high p_T charged particles and select only high pT ones

Yoo et al 2024 Mach. Learn.: Sci. Technol. 5 035047



- Can be done with a **NN implemented on sensor with the hls4ml+Catapult bundle**!

- Found data reduction by **54.4% - 75.4%** with **low latency (3.9 ns)** and **low power (300 µW/cm²)**

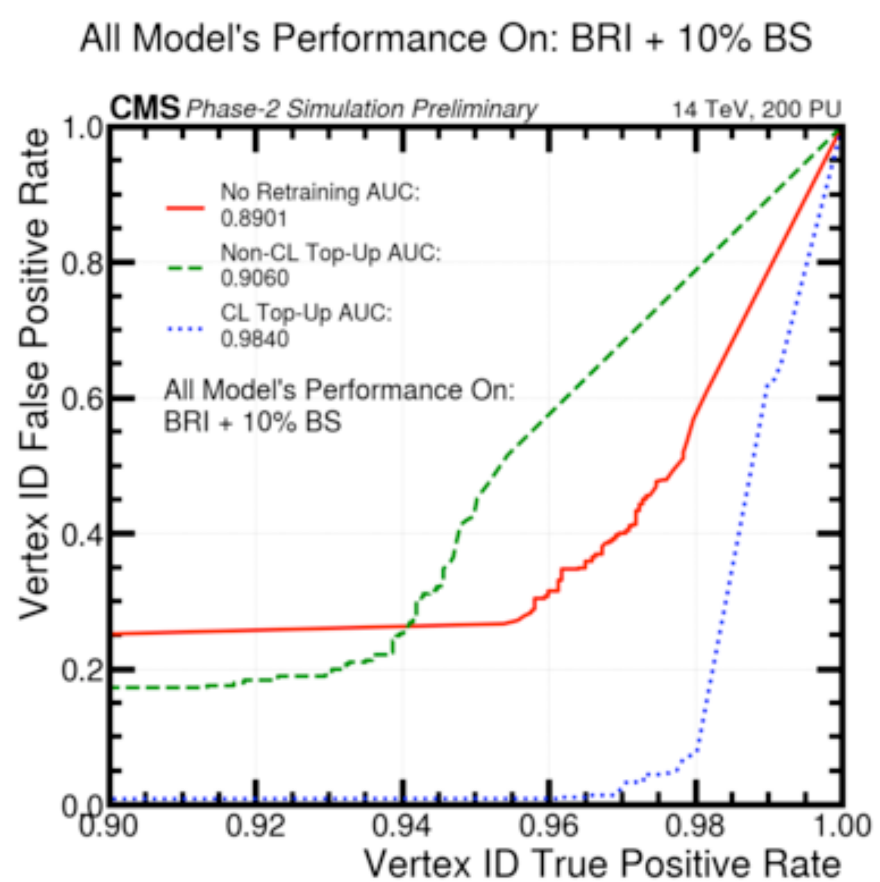


- **Prototype 1.5mm2 ASIC with momentum filtering NN in 28nm CMOS has been fabricated and tests are in progress**

arXiv.2406.14860

Adaptability and robustness

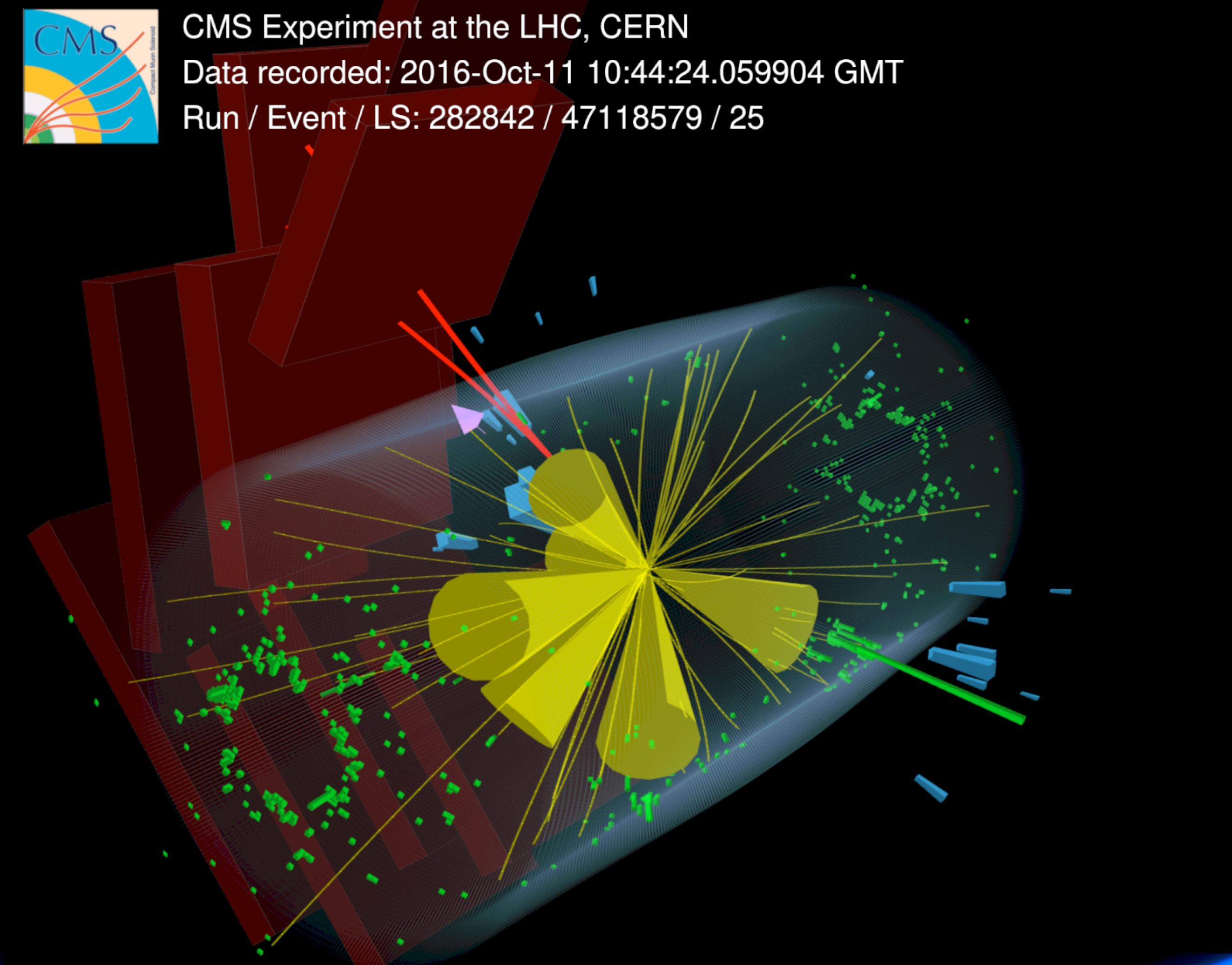
- One of the challenges in both real-time and offline workflows is **adaptability and robustness to external conditions** → improve domain adaptation & thus uncertainty quantification
  - **offline:** mostly driven by data/simulation disagreement → rely on already robust offline reconstruction (updated ~ once per year)
  - **online:** less robust reconstruction and changes on the scale of seconds to days / few weeks
- **Possible approaches** (not mutually exclusive) under active R&D:
  - robust **pre-training strategies**
  - **continual learning** (e.g., top up trainings)
  - **agent-based for autonomous adjustments** (e.g., through reinforcement/active learning)
  - establishing **MLOps pipelines**: train, deploy, maintain

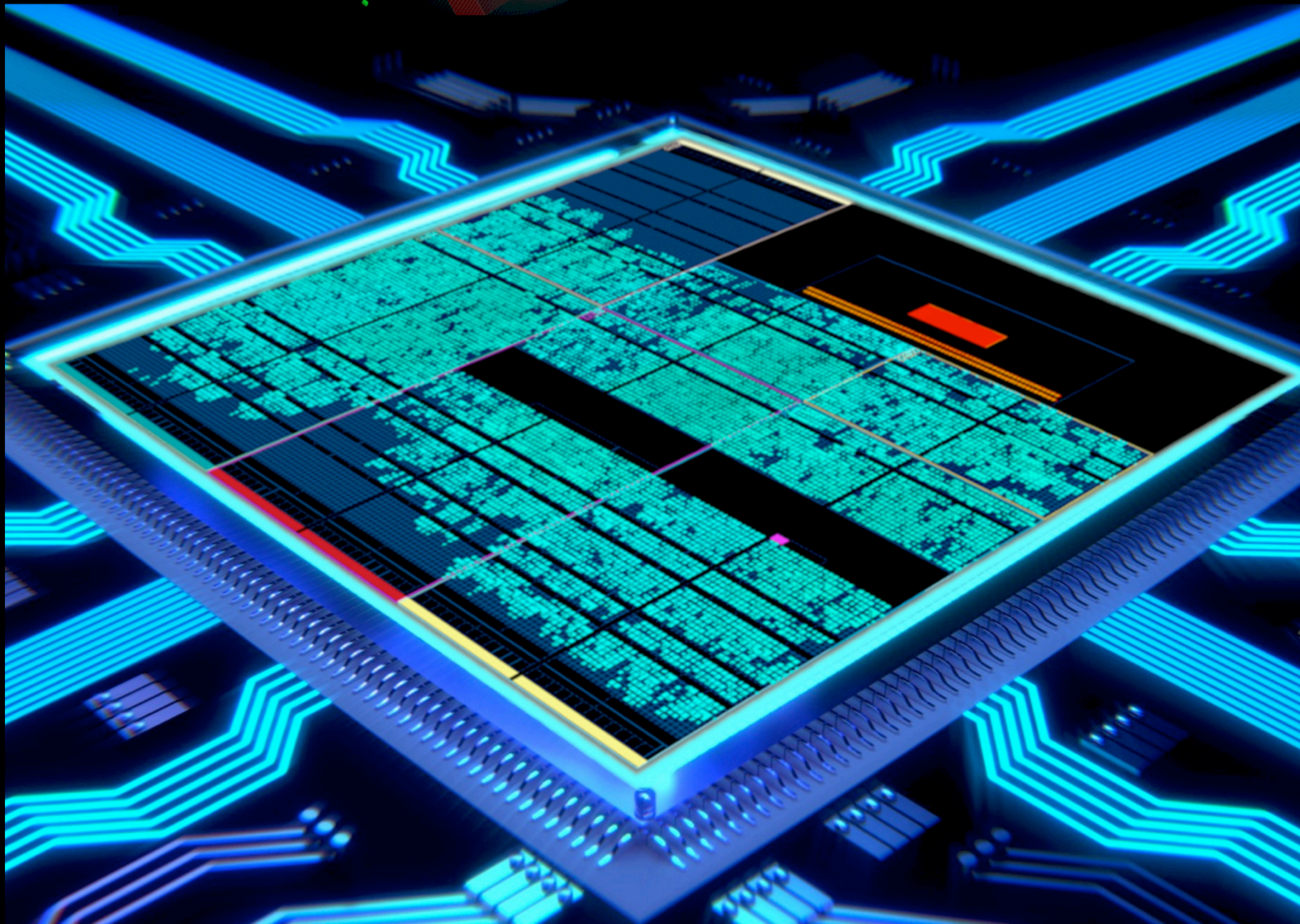


Example: continual learning for CMS Phase 2 tracker degradation
[CMS-DP-2023-022]

Summary

- **We hope to understand the fundamental structure of nature**
  - we expect new phenomena to answer those questions
  - but these are rare so we build large scale experimental setups
- **The challenge ahead is big**
  - more data, more complex data, not enough resources
- **This is why we need to push ML to all pipelines down to the edge**
  - to do more with less (faster & better)
- **And hopefully discover new phenomena!**





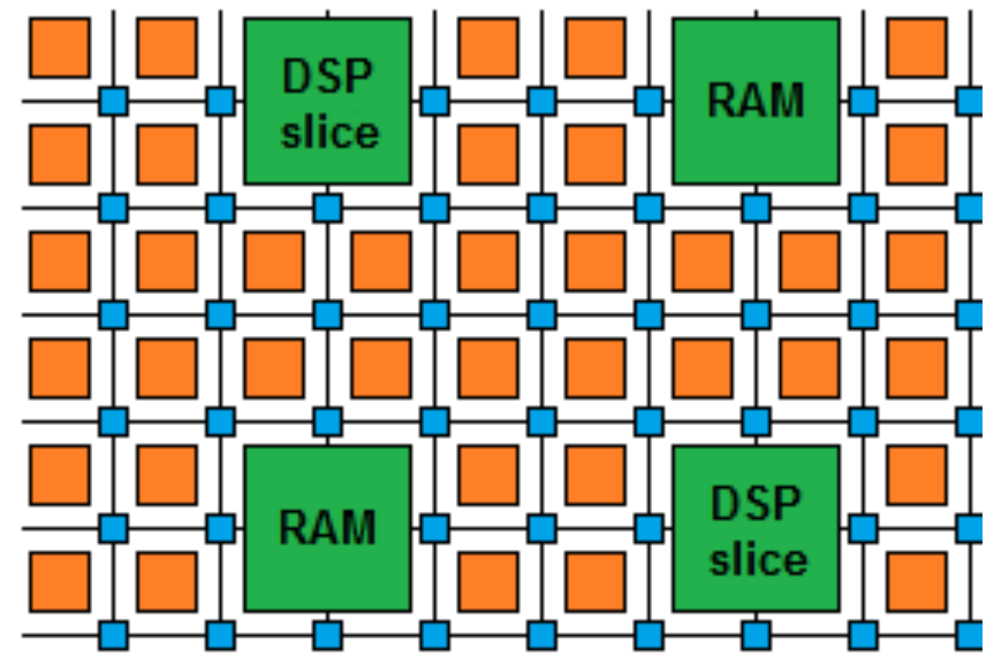
Backup

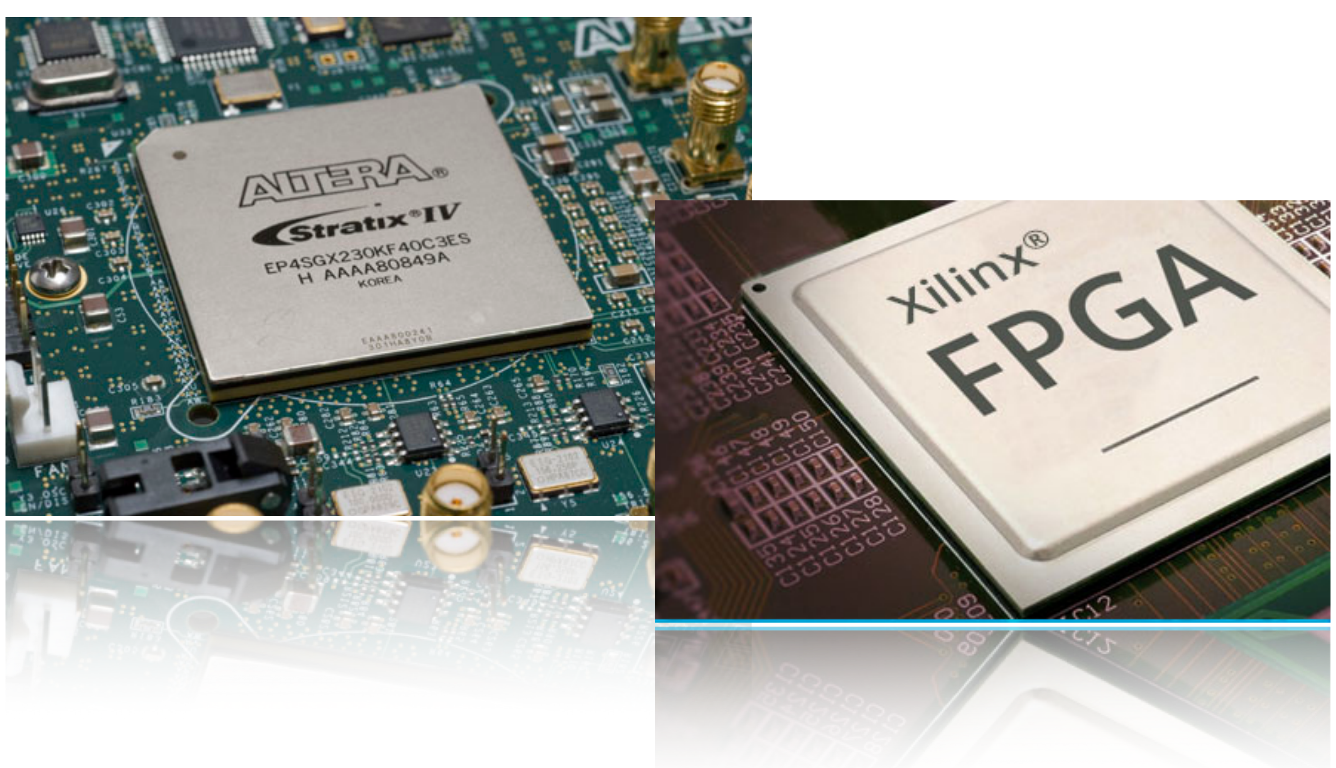
What are FPGAs?

Field Programmable Gate Arrays are reprogrammable integrated circuits

Contain many different building blocks (‘resources’) which are connected together as you desire

FPGA diagram



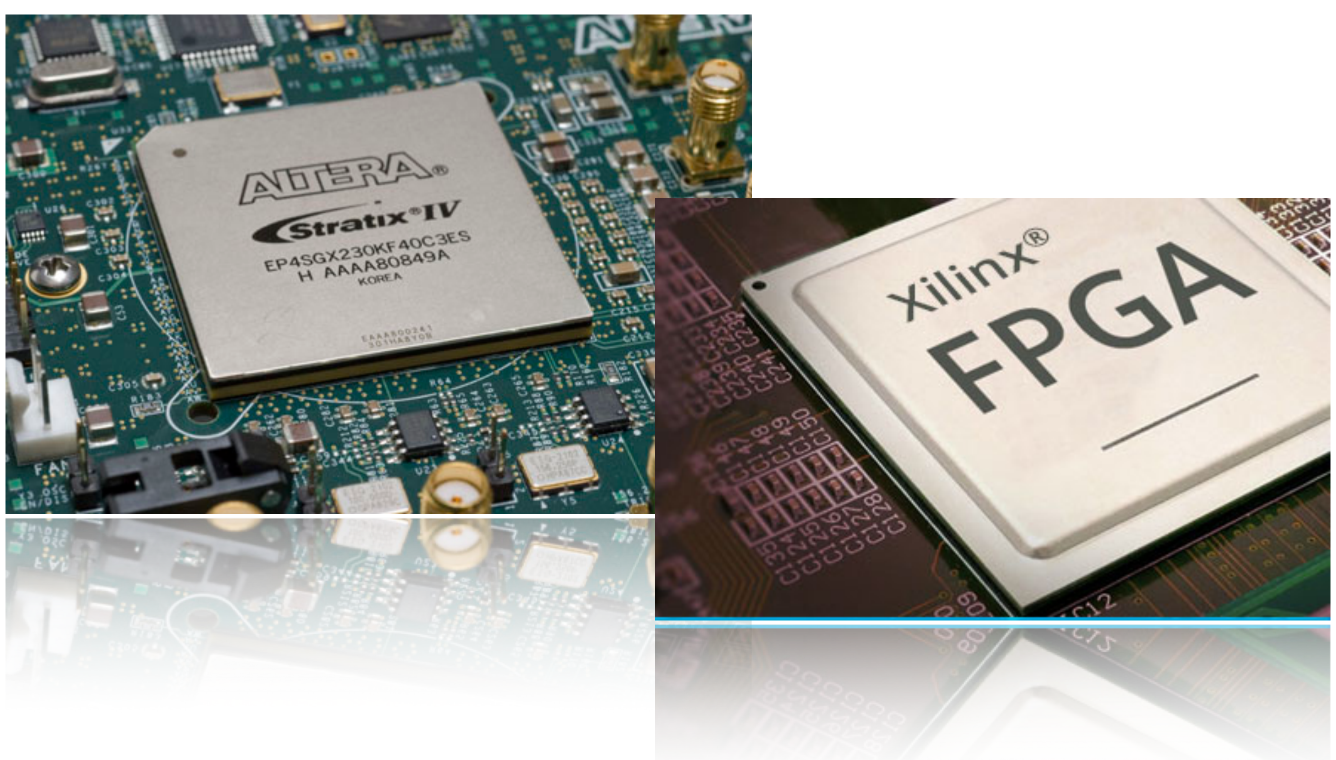


What are FPGAs?

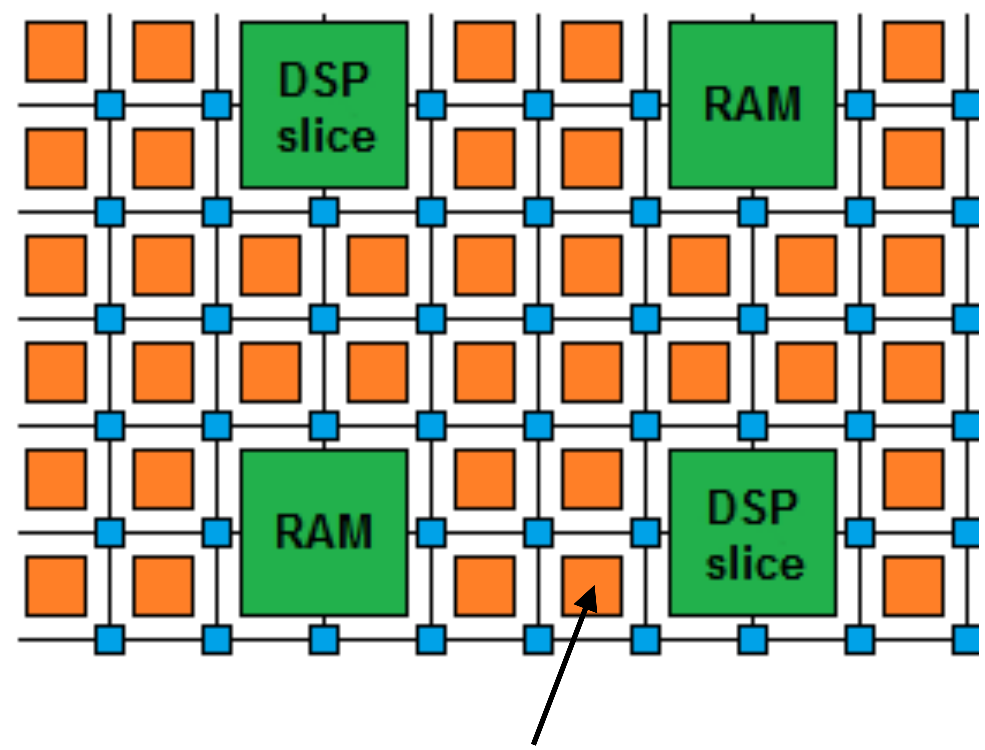
Field Programmable Gate Arrays
are reprogrammable integrated circuits

Look Up Tables (LUTs) perform arbitrary functions on small bitwidth inputs (2-6 bits)
→ used for boolean operations, arithmetics, memory

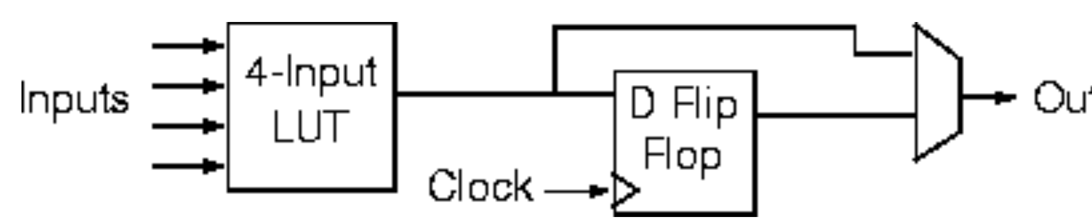
Flip-flops register data in time with the clock pulse



FPGA diagram



Logic cell



Look-up table (logic)

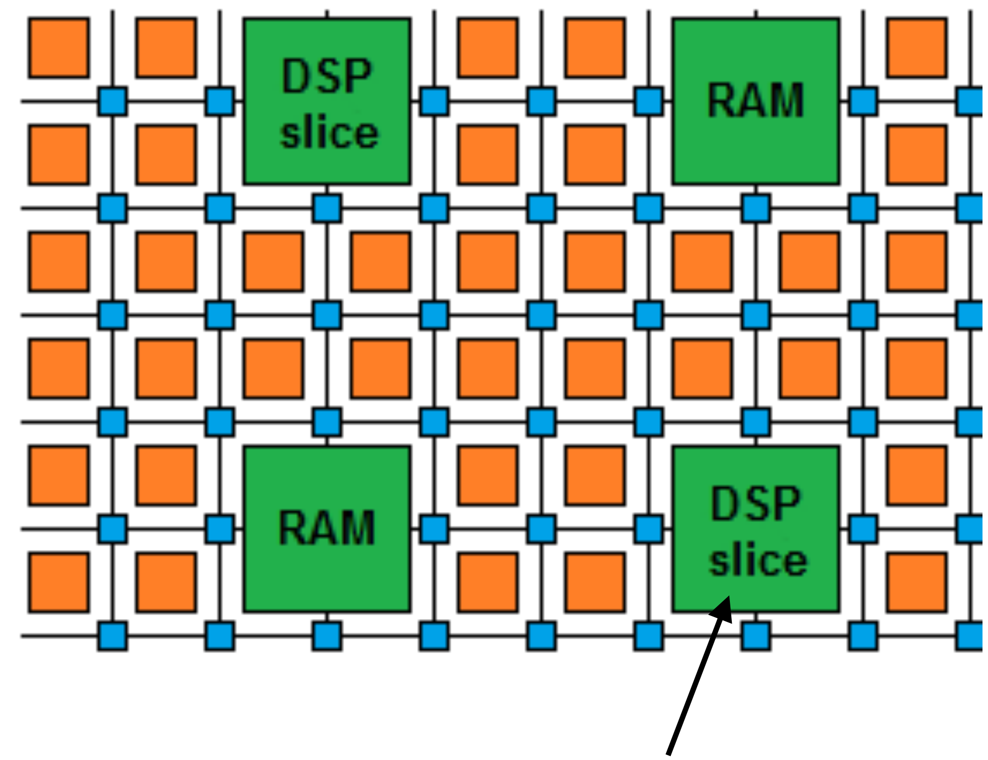
Flip-flop (registers)

What are FPGAs?

Field Programmable Gate Arrays
are reprogrammable integrated circuits

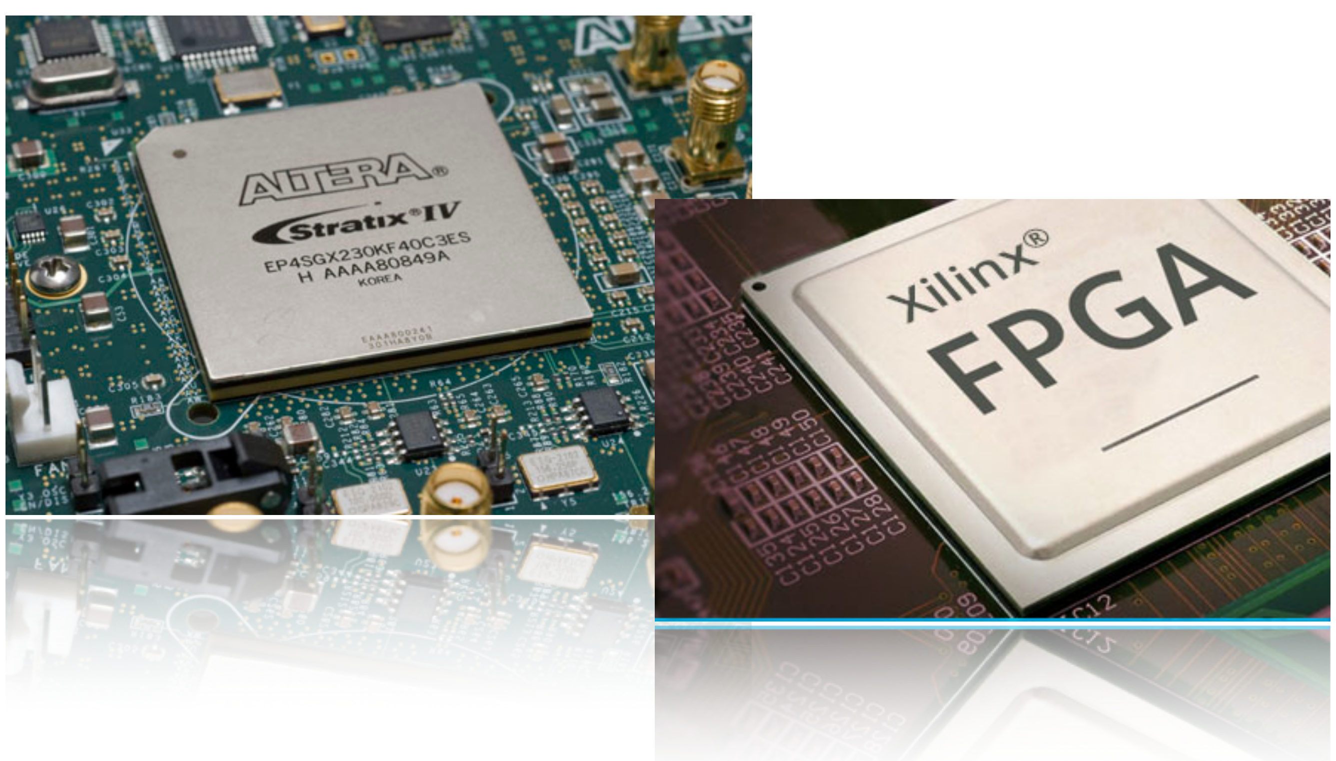
DSPs are specialized units for multiplication and arithmetic
→ faster and more efficient than LUTs for these type of operations
→ for deep learning, they are often the most precious resource

FPGA diagram



Also contain embedded components:

Digital Signal Processors (DSPs): logic units used for multiplications



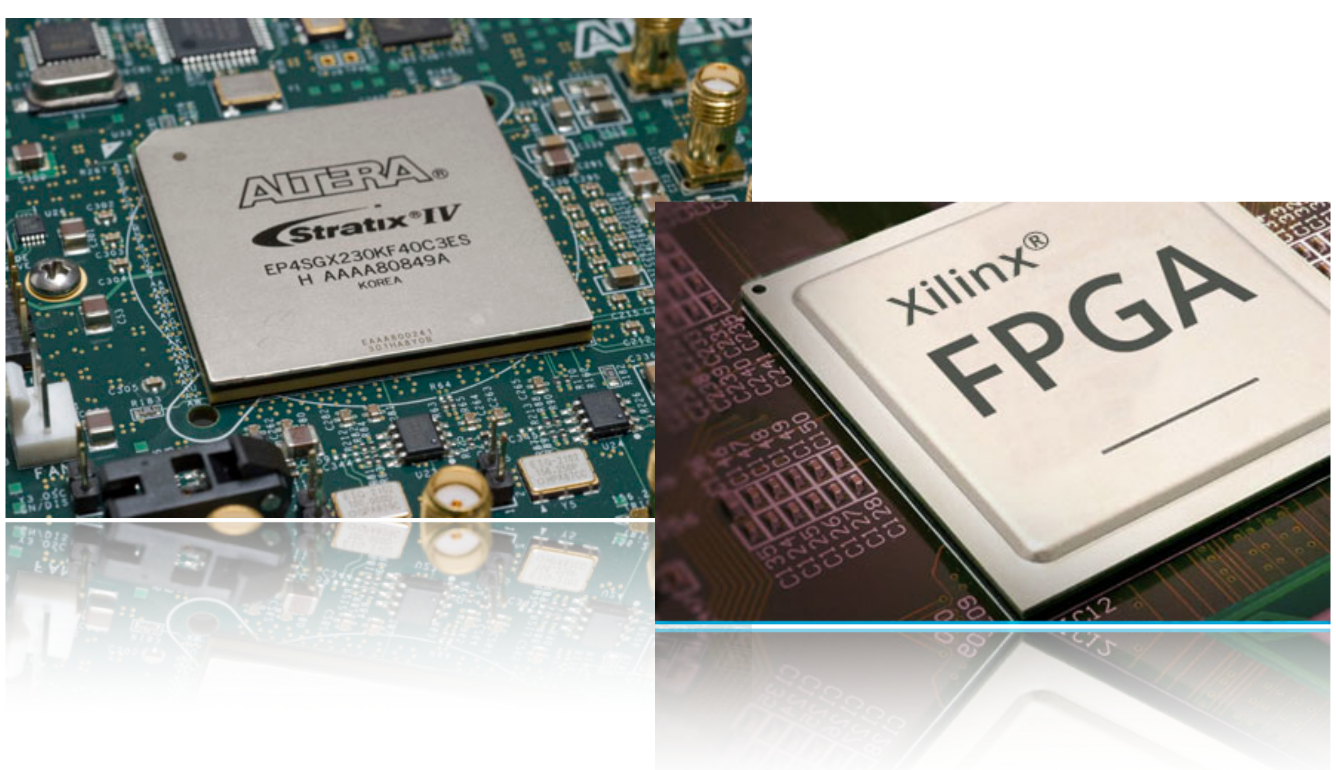
What are FPGAs?

Field Programmable Gate Arrays are reprogrammable integrated circuits

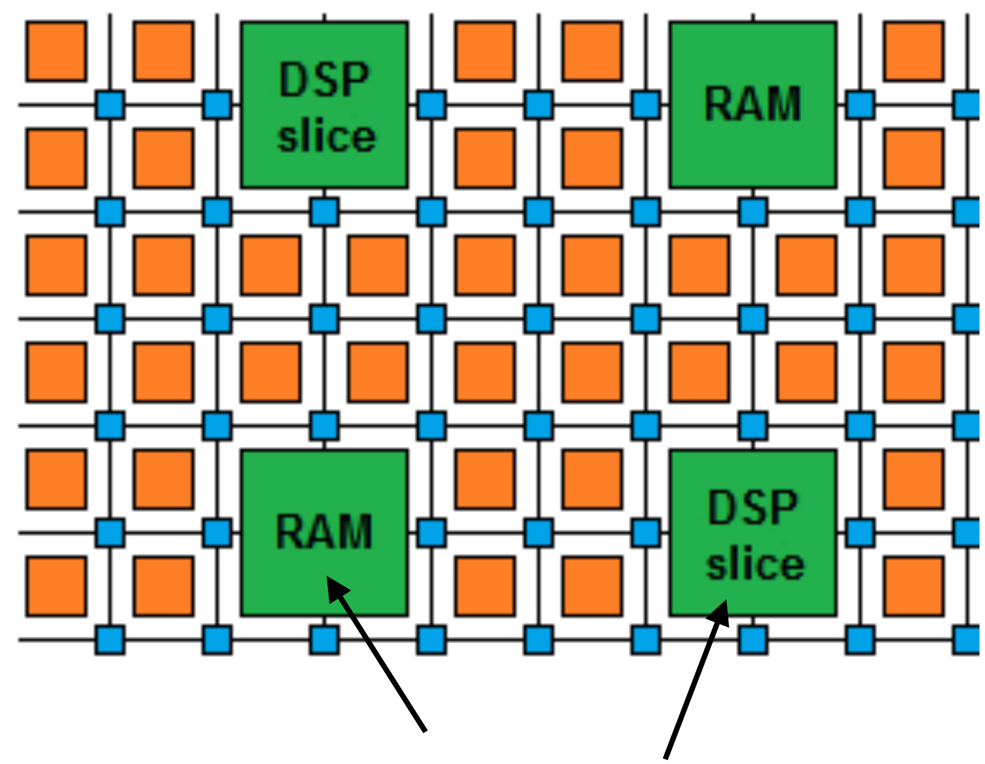
BRAMs are small, fast memories (ex, 18 Kb each)

→ more efficient than LUTs when large memory is required

Modern FPGAs have ~100 Mb of BRAMs, chained together as needed



FPGA diagram



Also contain embedded components:

Digital Signal Processors (DSPs): logic units used for multiplications

Random-access memories (RAMs): embedded memory elements

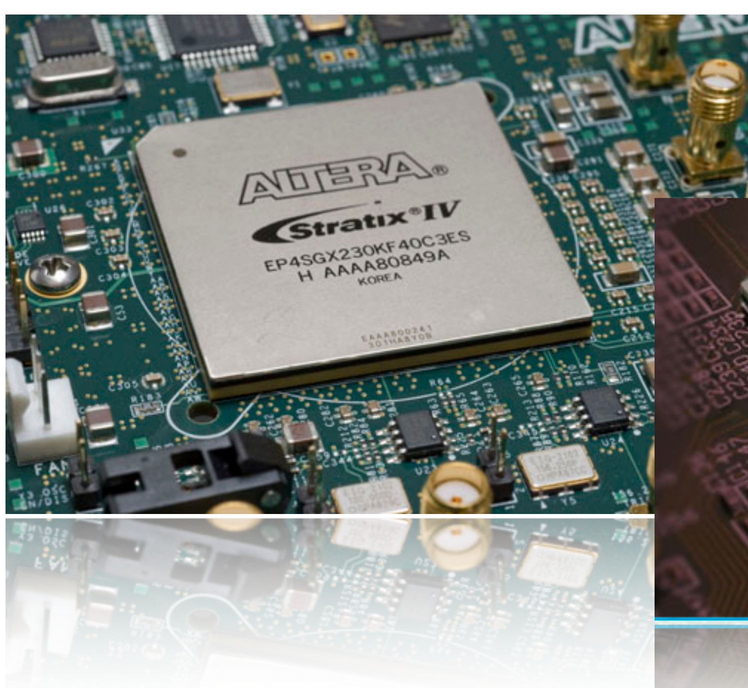
What are FPGAs?

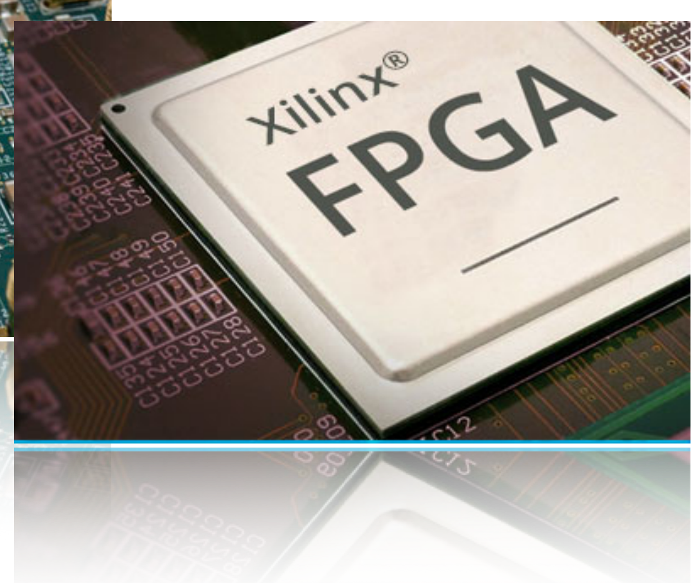
Field Programmable Gate Arrays
are reprogrammable integrated circuits

Contain array of **logic cells** embedded with **DSPs**, **BRAMs**, etc.

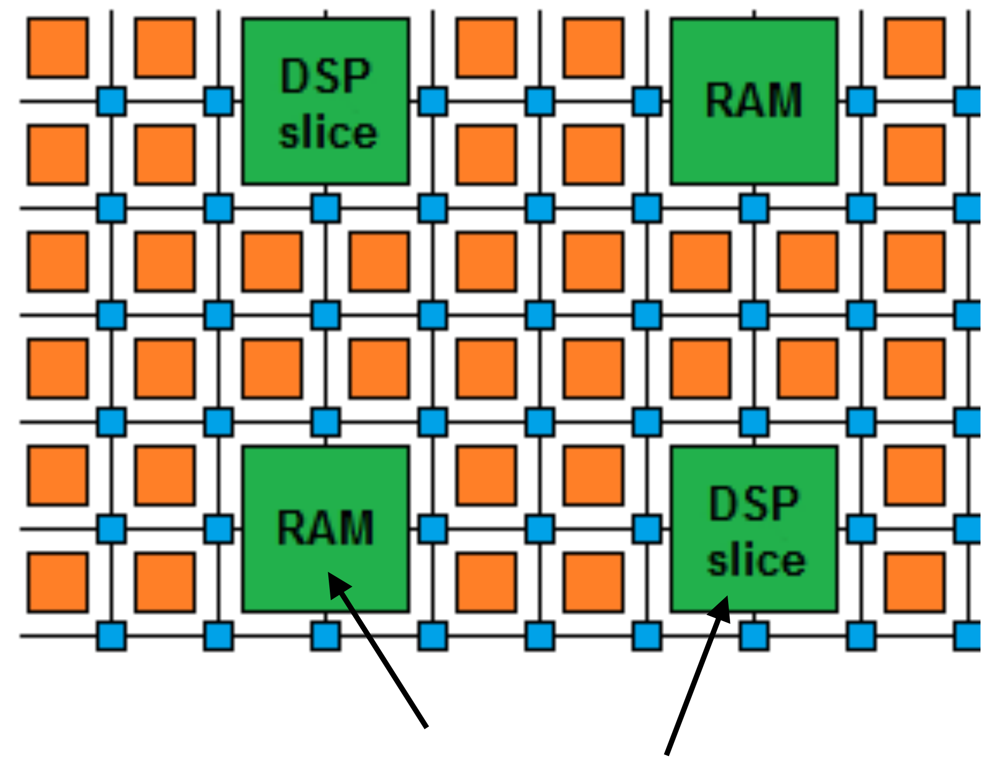
Support **highly parallel** algorithm implementation

Low power per Op (relative to CPU/GPU)





FPGA diagram



Also contain embedded components:

Digital Signal Processors (DSPs): logic units used for multiplications

Random-access memories (RAMs): embedded memory elements

Why are FPGAs fast?

- Fine-grained / resource parallelism
  - use the many resources to work on different parts of the problem simultaneously
  - allows us to achieve **low latency**
- Most problems have at least some sequential aspect, limiting how low latency we can go
  - but we can still take advantage of it with…
- **Pipeline parallelism**
  - instruct the FPGA to work on different data simultaneously
  - allows us to achieve **high throughput**

Like a production line for data…

How are FPGAs programmed?

Hardware Description Languages

HDLs are programming languages which describe electronic circuits

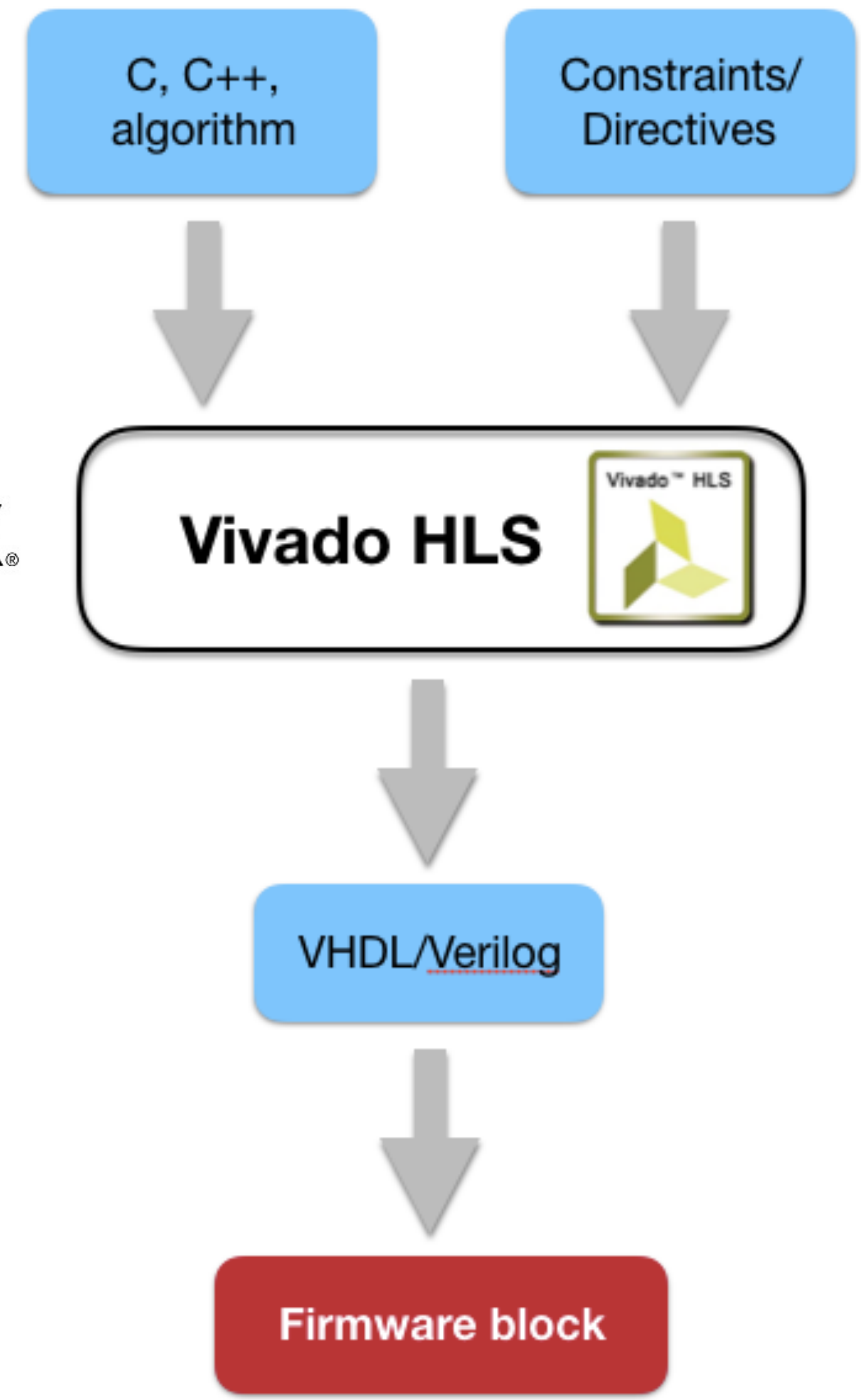
High Level Synthesis

generate HDL from more common C/C++ code

pre-processor directives and constraints used to optimize the timing

drastic decrease in firmware development time!

See Xilinx Vivado HLS, Intel HLS, Catapult HLS

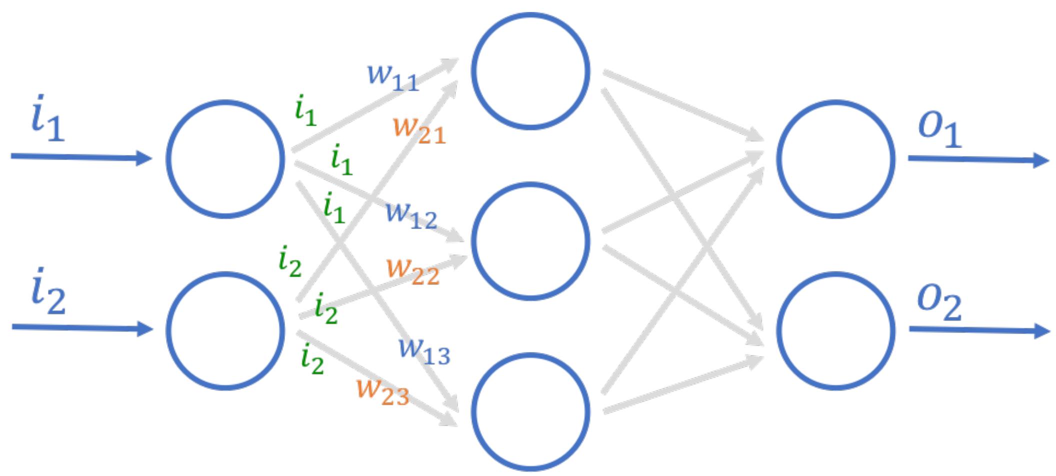


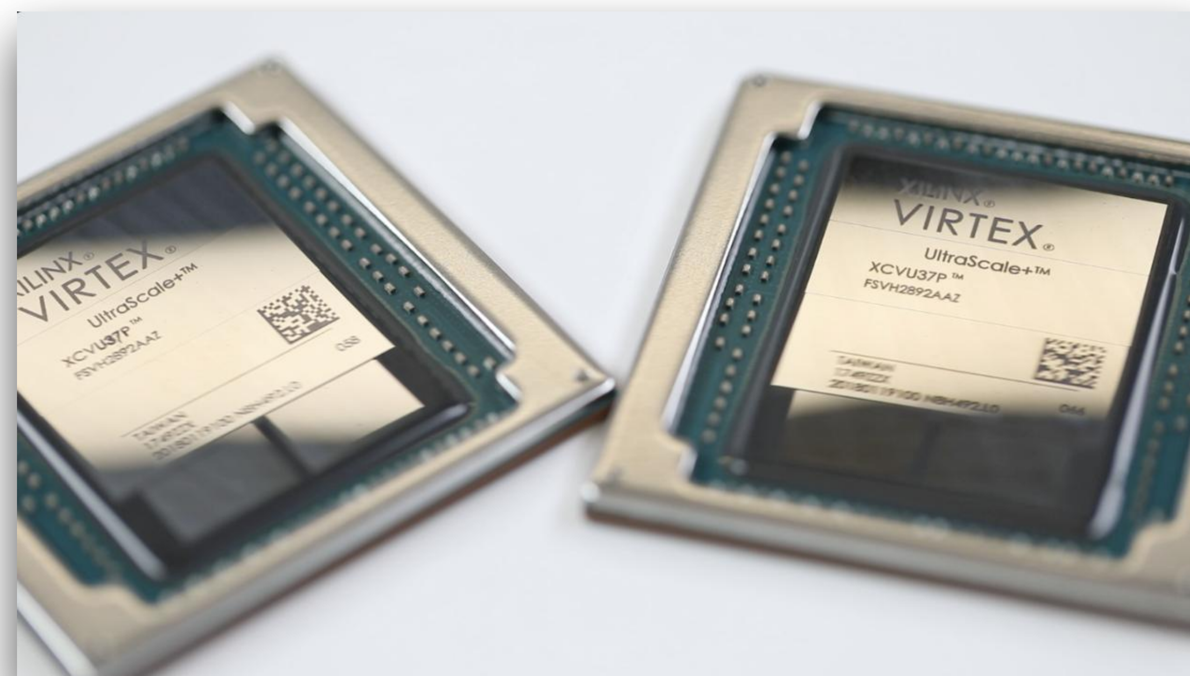
Neural Network inference on FPGA

Neural network inference
=
matrix multiplication

Efficient implementation on FPGA uses
DIGITAL SIGNAL PROCESSORS

There are about 5–10k DSPs in modern FPGAs!



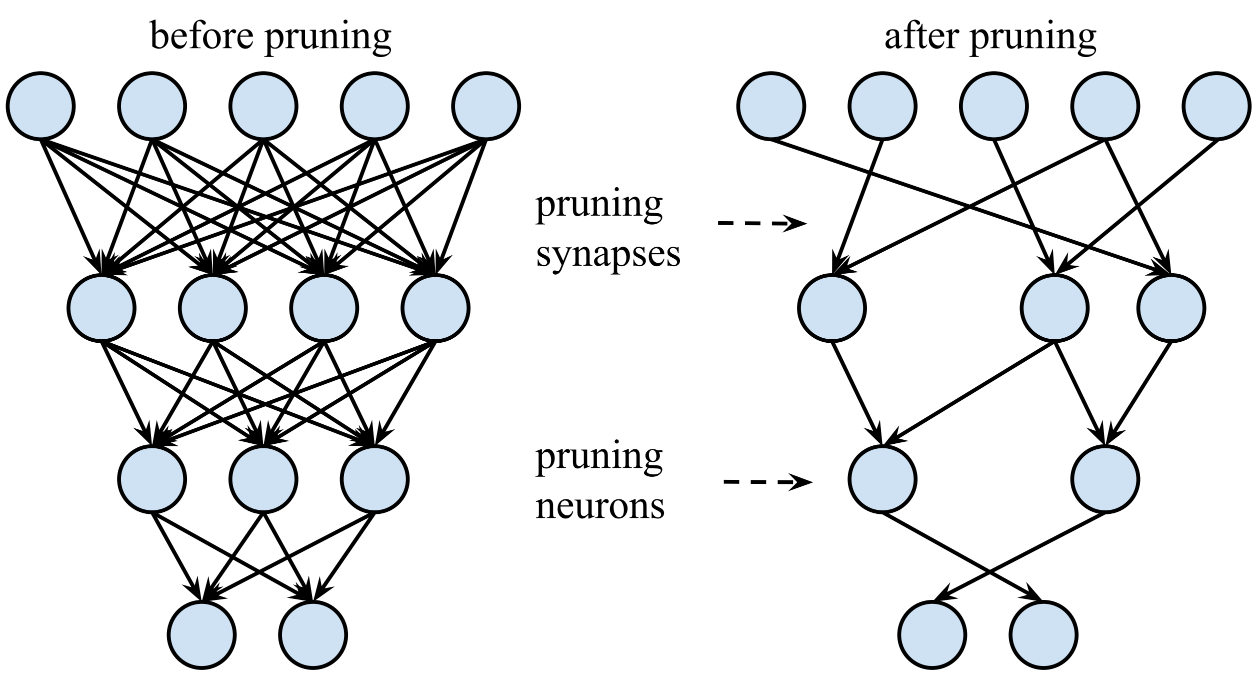


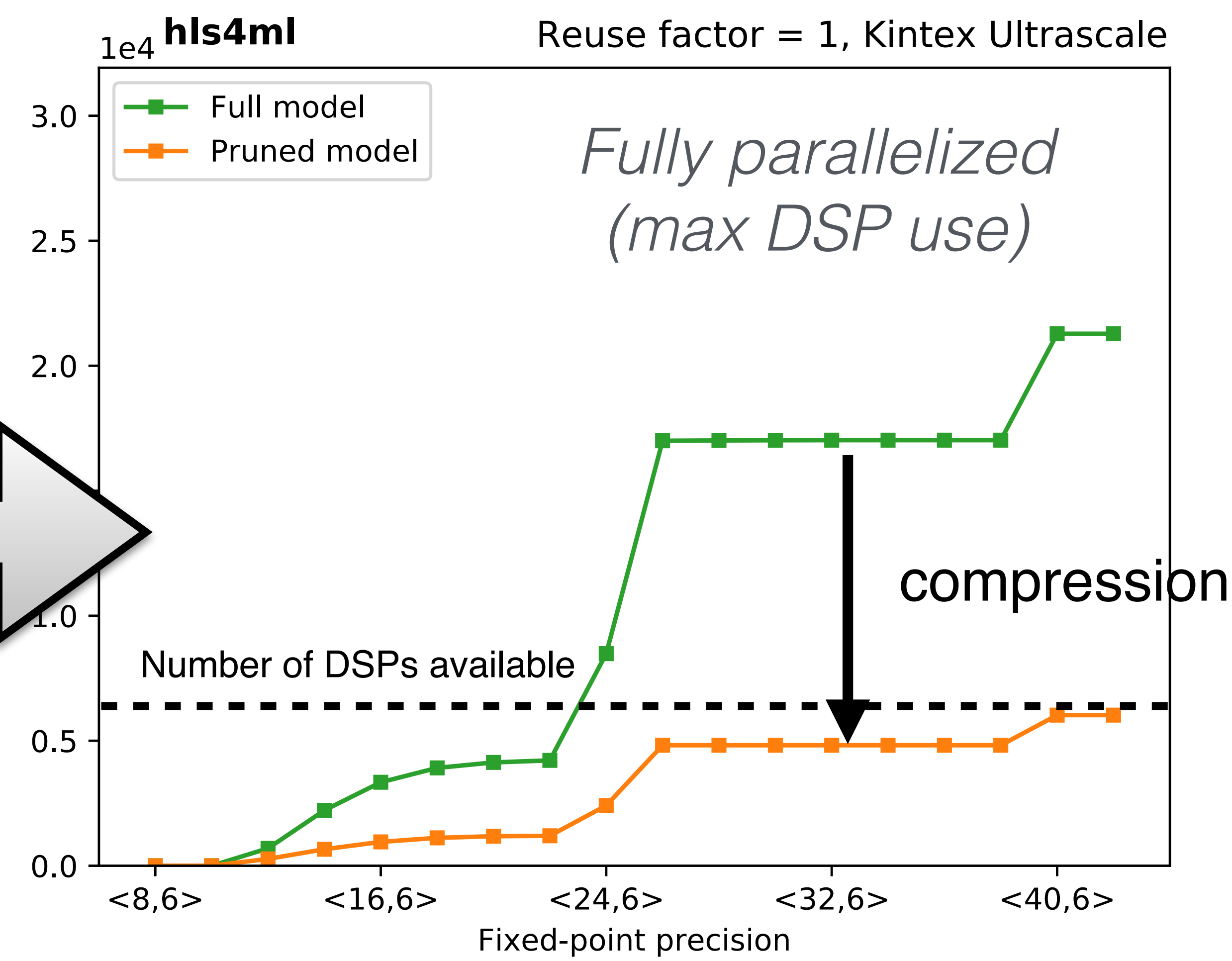
ex: Xilinx Virtex Ultrascale +

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

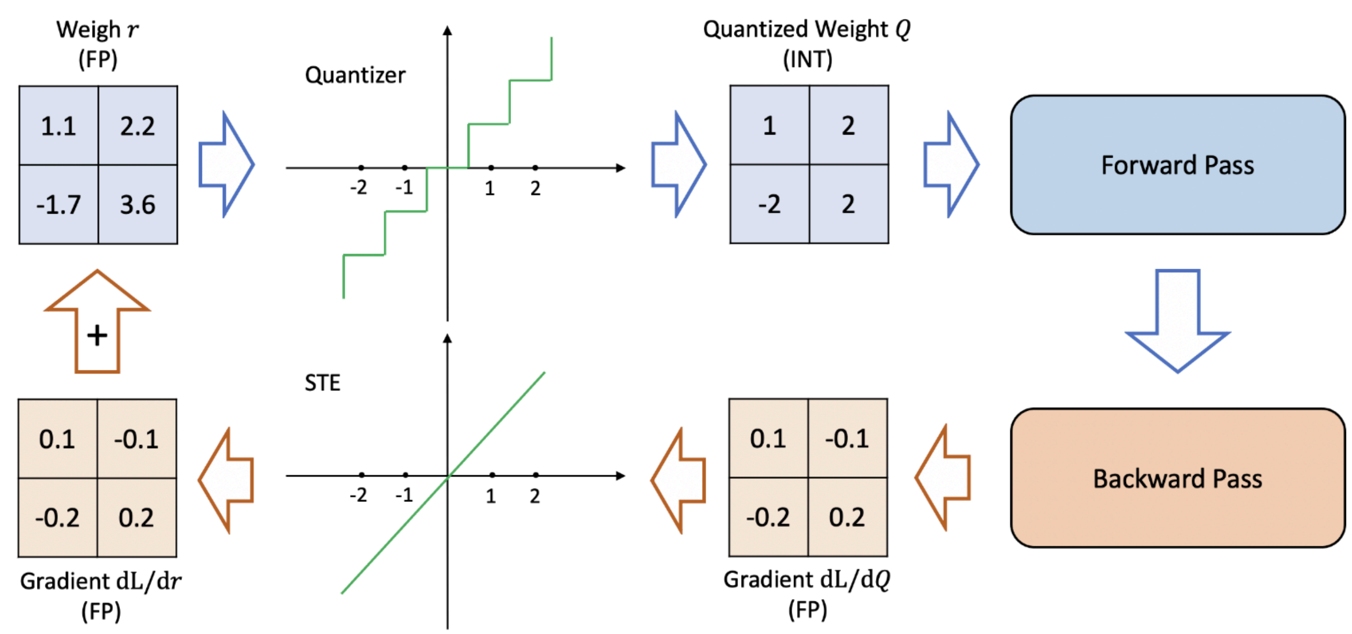




70% compression ~ 70% fewer DSPs

Quantization-aware training

- **Post-training quantization can affect accuracy**
    - for a given bit allocation, the loss minimum at floating-point precision might not be the minimum anymore
- One could **specify quantization while look for the minimum during training**
    - quantization functions applied to weights and activations only in the forward pass
    - use Straight Through Estimator for back propagation step
- **Our workflow:** quantization-aware training with `Google QKeras` and firmware design with `hls4ml` for most efficient NN inference on chip!



A. Gholami et al, arxiv.2103.13630

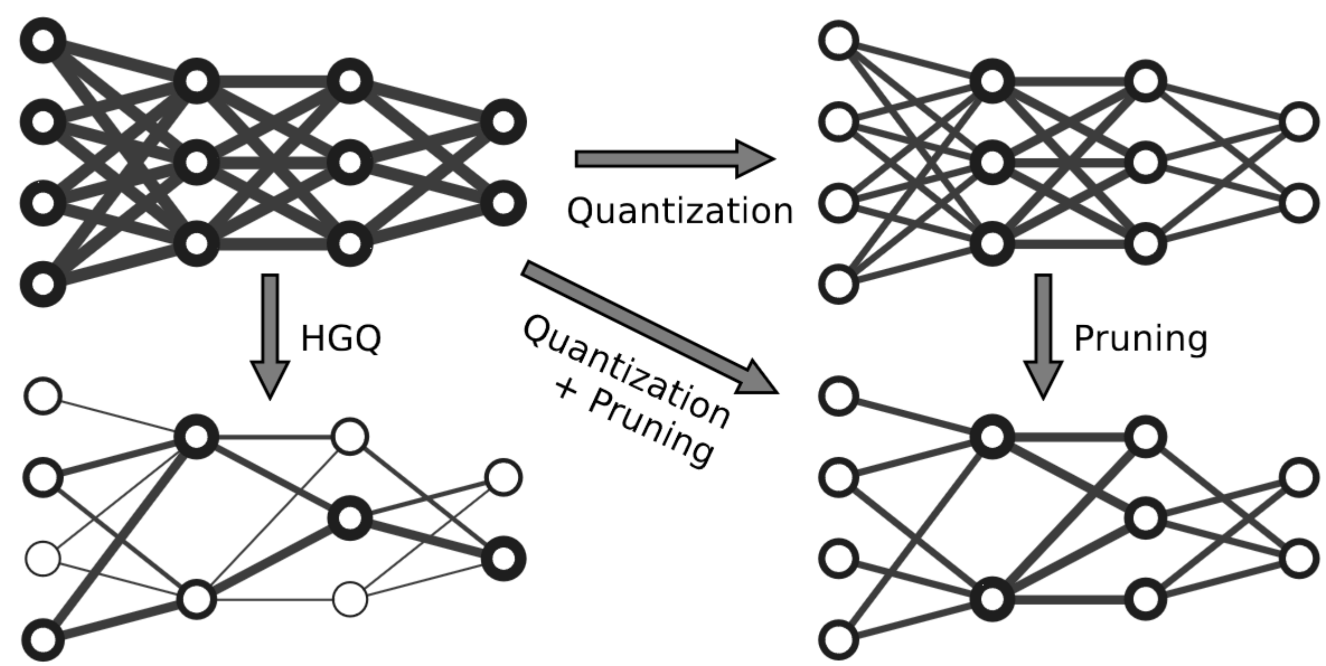
High-Granularity Quantization

- **The wish:** squeeze even more NN inference performance when each parameter in the network may have its unique bitwidth

- **Limitations of QKeras:**

  - bitwidths for NN parameters are optimized in predefined, structured block (e.g., per layer)

  - bitdiwdth is not part of optimization → need to run your own hyperparameter scan

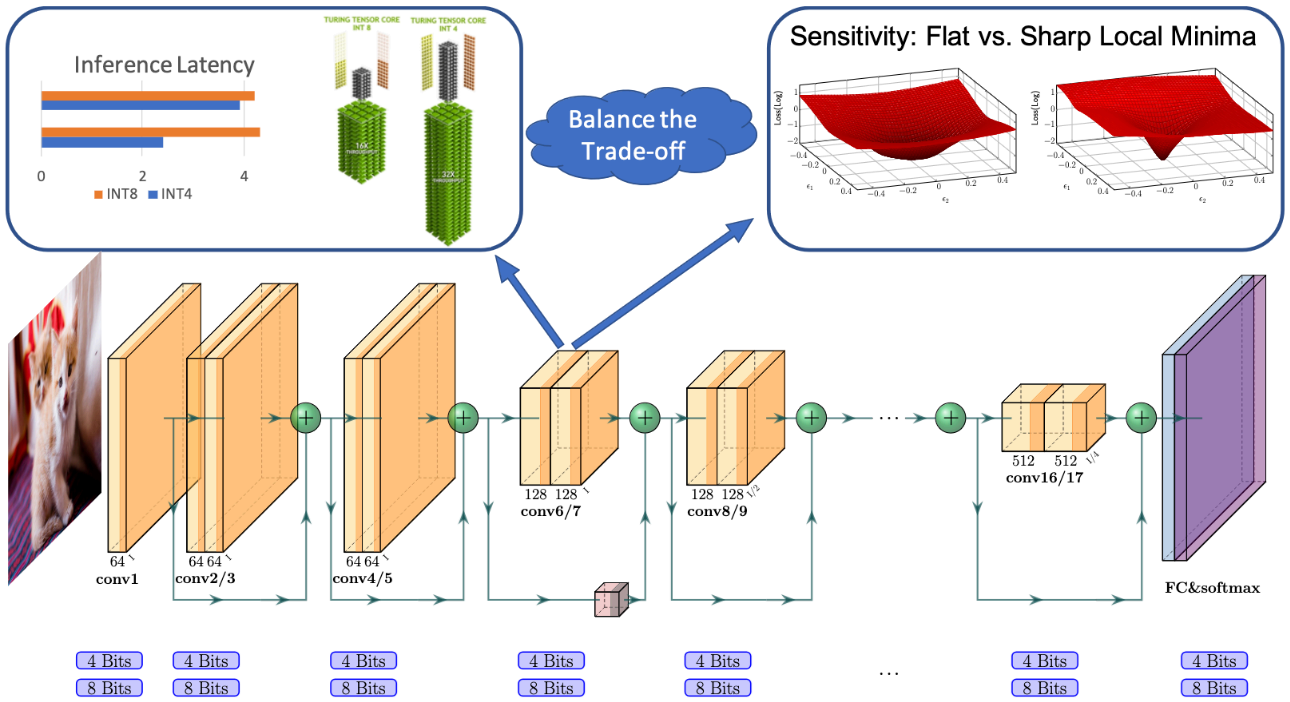


- **Solution: optimize the individual bitwidths alongside the NN accuracy using gradient descent**

Other quantization methods

HAWQ — Hessian AWare Quantization

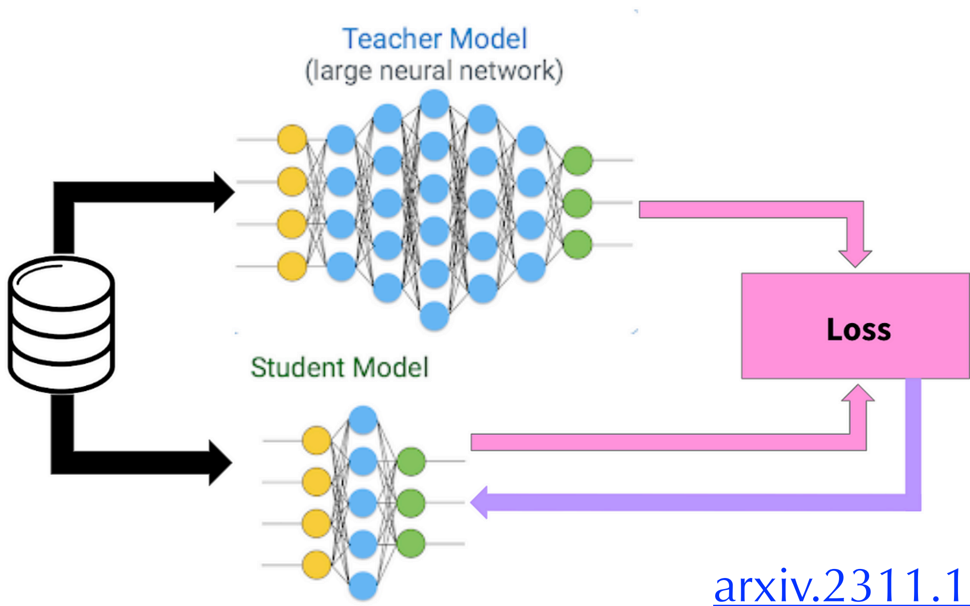
- mixed-precision quantization tool written for PyTorch
- main idea: ***sensitive*** layers are kept at higher precision than less ***sensitive*** layers
- problem: search space is **exponential** to the number of layers in models



- solution: use ILP to find the optimal trade-off between model perturbation (through Hessian trace) and application-specific constraints (latency, BOPs, size limit,...)
- Scales linearly w.r.t to the number of layers and bitwidth options

arxiv.2011.10680

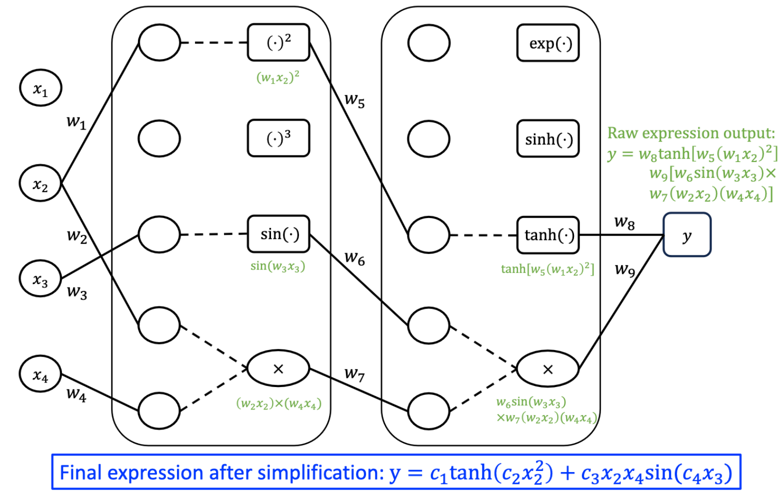
Efficiency beyond quantization



arxiv.2311.14160

Knowledge distillation

- Allows to deploy smaller student NN at similar accuracy of more complex teacher NN
- And to transfer powerful inductive bias to the student NN

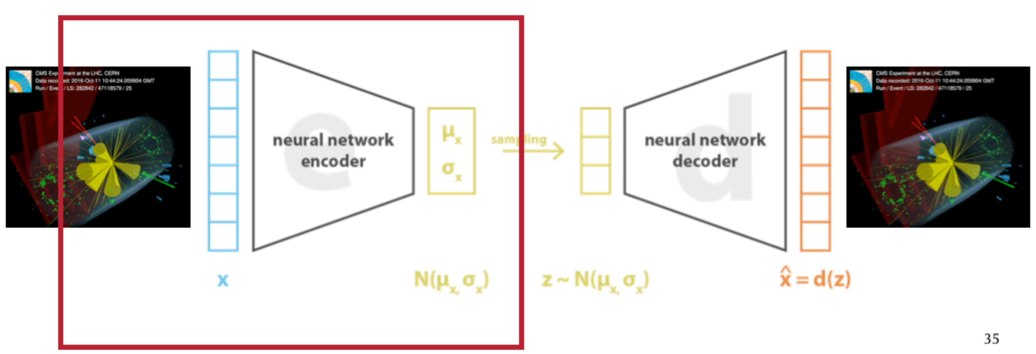


arxiv.2401.09949
arxiv.2406.16752

Symbolic regression

- Trained with gradient-based approach can achieve high sparsity and compact representation
- Mathematical operations can be implemented efficiently in HLS with LUTs

Fast autoencoders for anomaly detection



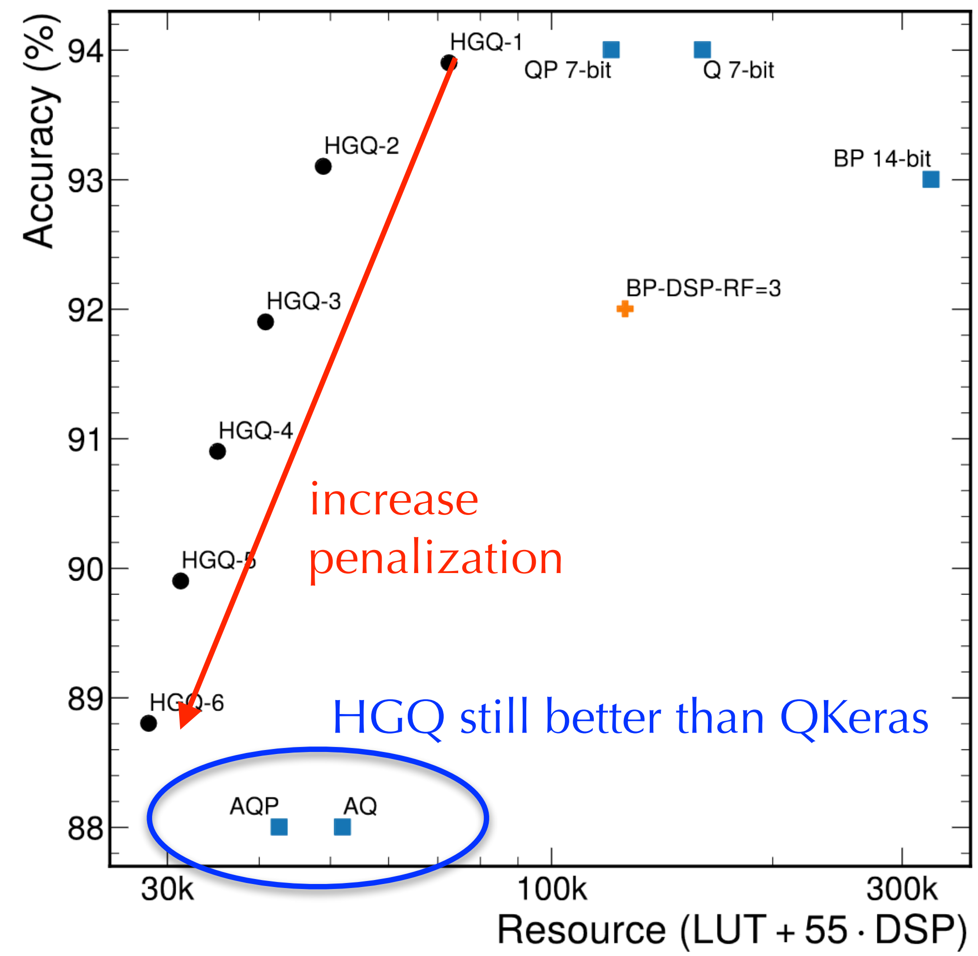
arxiv.2311.17162, 2401.08777, 2108.03986

- If variational, define anomaly metric in latent space → deploy only encoder in inference → half latency and model size
- Informed latent representations can lead to more efficient model → SSL and compact foundation models/ transformers

High-Granularity Quantization

- **Solution: optimize the individual bitwidths alongside the NN accuracy using gradient descent**

- **How:**
  - treat the bitwidths as continuous variables
  - introduce surrogate gradients for discrete variables such as bitwidths
  - introduce a novel on-chip resource consumption metric that when incorporated into the loss function penalizes larger bitwidths efficiently
  - pruning integrated naturally in the optimization step (gradient descent reduces certain bitwidths to zero)



Gradient-based Automatic Mixed Precision Quantization for Neural Networks On-Chip

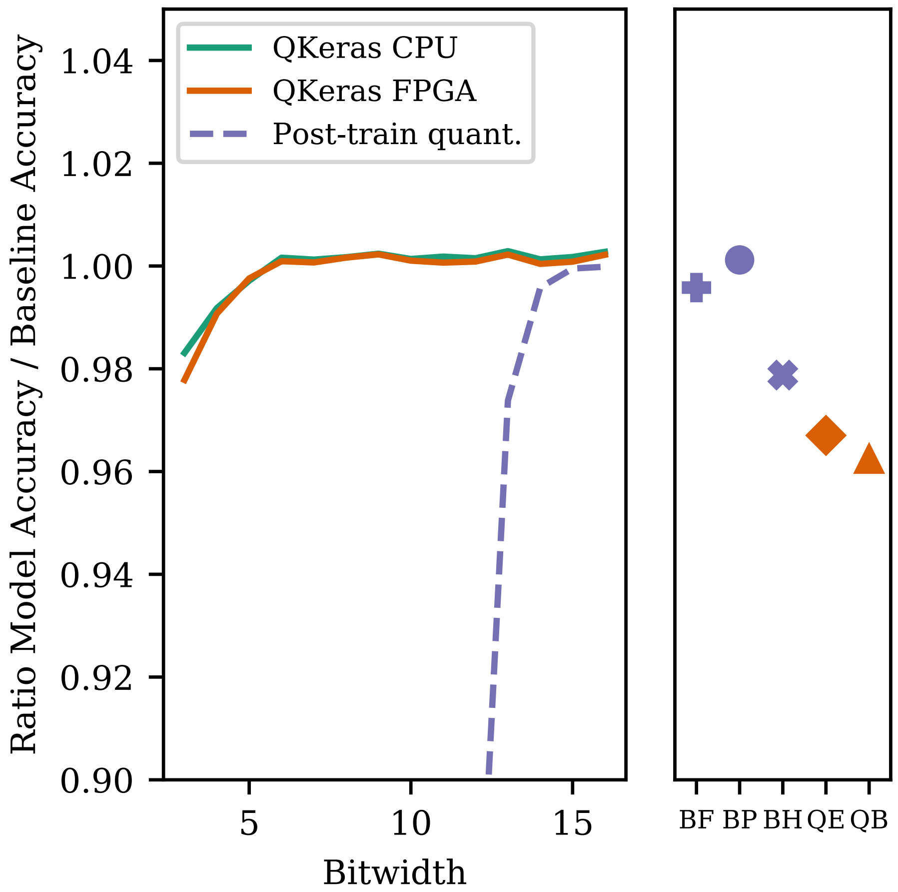
Chang Sun,[1,2,*] Thea K. Årrestad,[1] Vladimir Loncar,[3,4] Jennifer Ngadiuba,[5] and Maria Spiropulu[2]

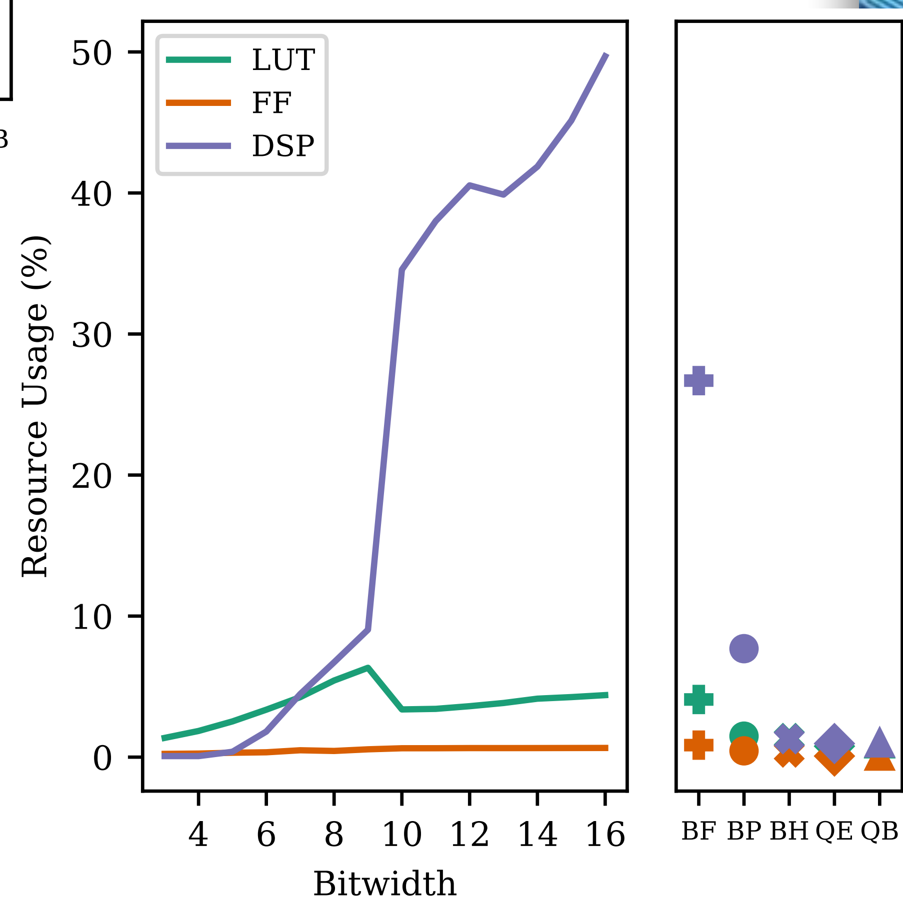
[1] *ETH Zurich (Zurich, Switzerland)*
[2] *California Institute of Technology (CA, USA)*
[3] *Massachusetts Institute of Technology (MA, USA)*
[4] *Institute of Physics Belgrade (Belgrade, Serbia)*
[5] *Fermi National Accelerator Laboratory (IL, USA)*

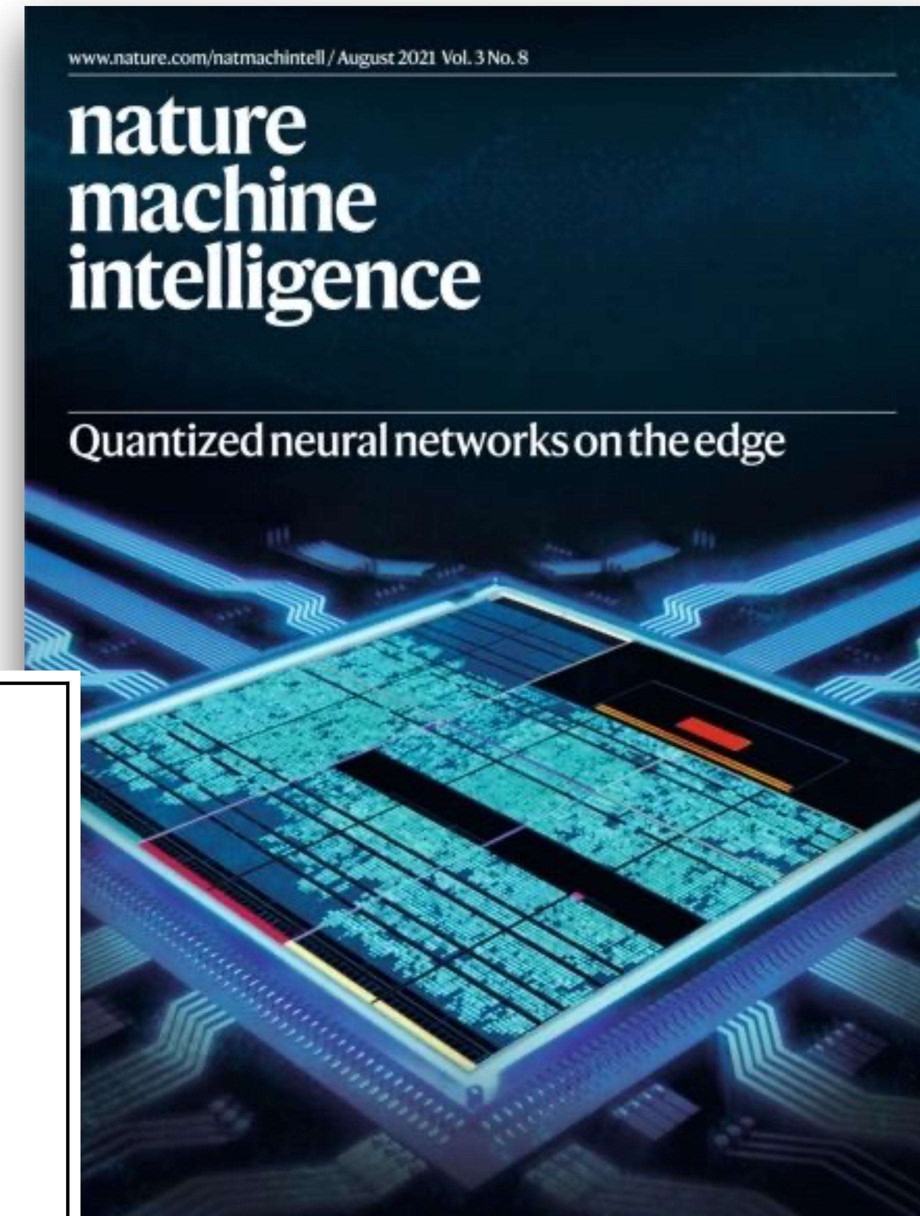
Fully supported in hls4ml!

Paper on arxiv ready for submission!

QKeras & hls4ml







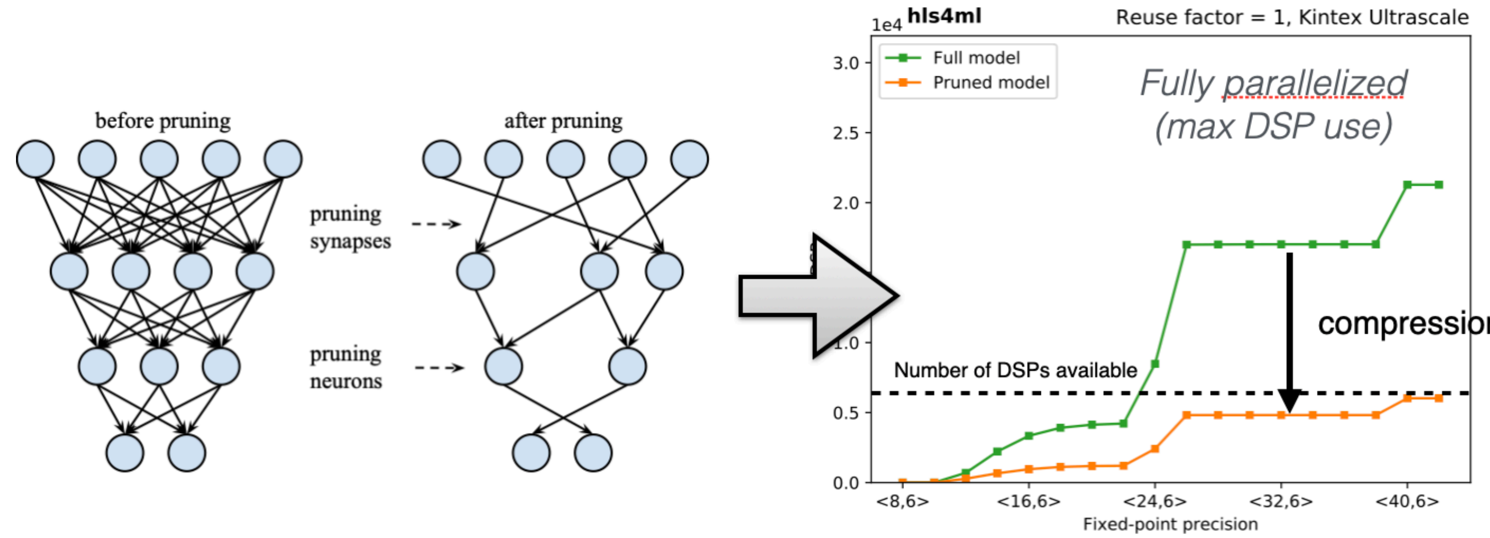
Matches ap_fixed exactly!

- same granularity as hls4ml
- same precision at training and inference

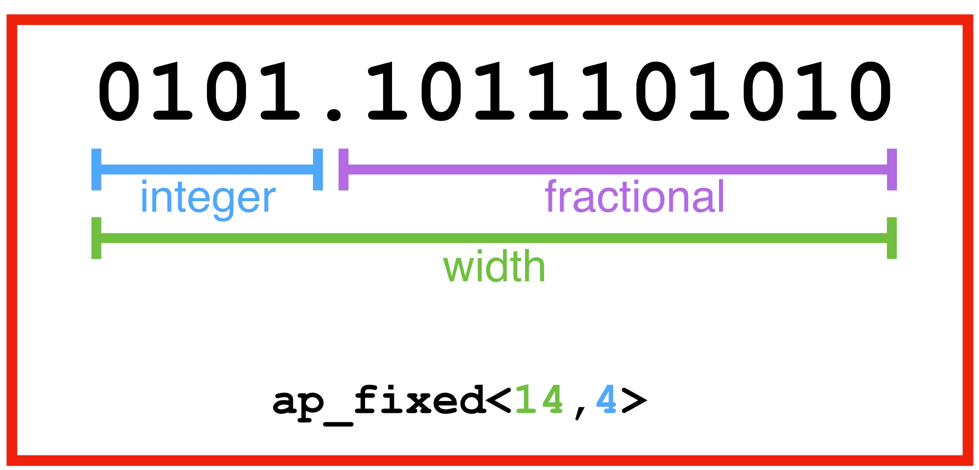
C. N. Coelho et al.: Nature Machine Intelligence, Volume 3 (2021)

Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

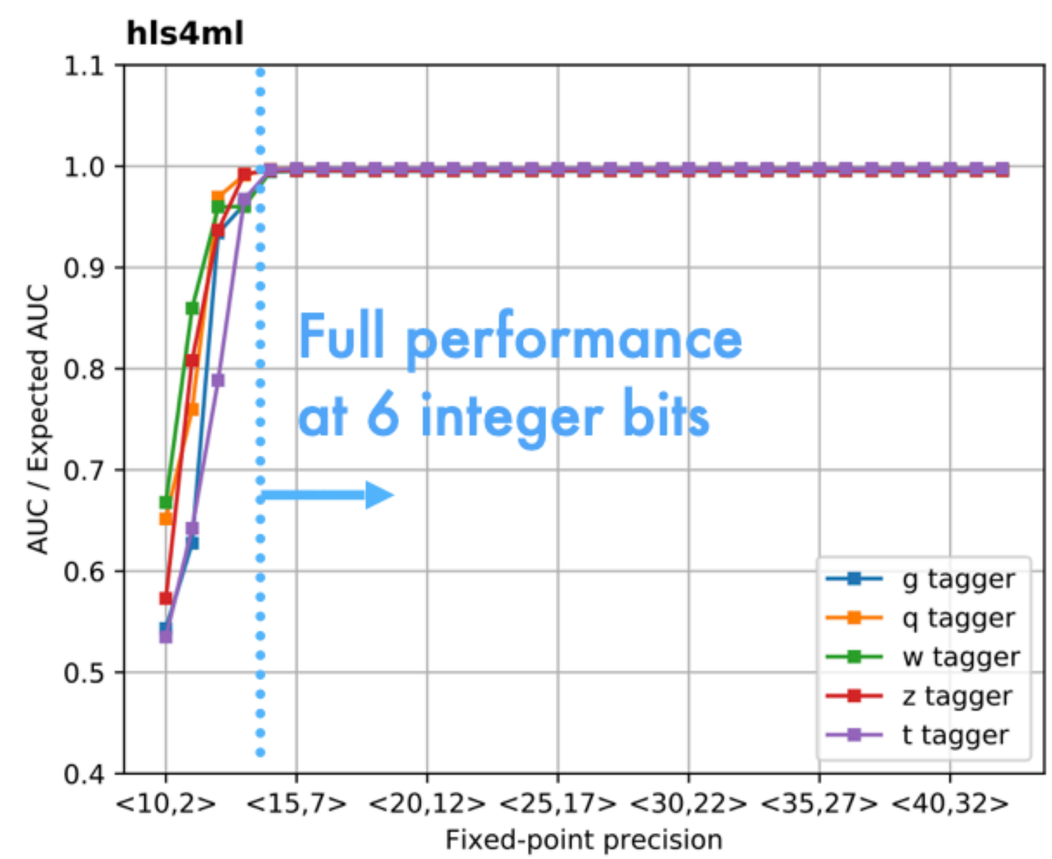


  - **Quantisation:** represents numbers with few bits reduce resources



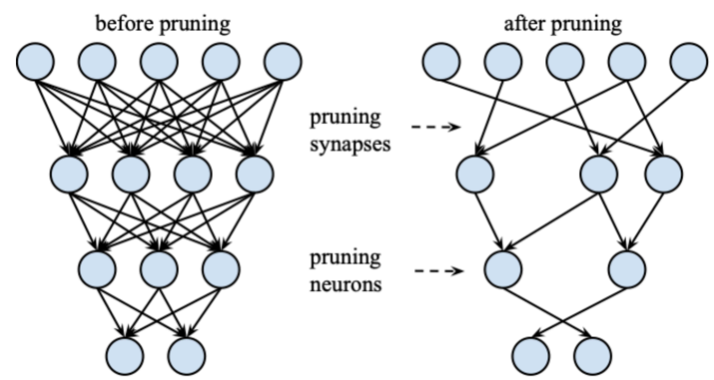
Scan integer bits

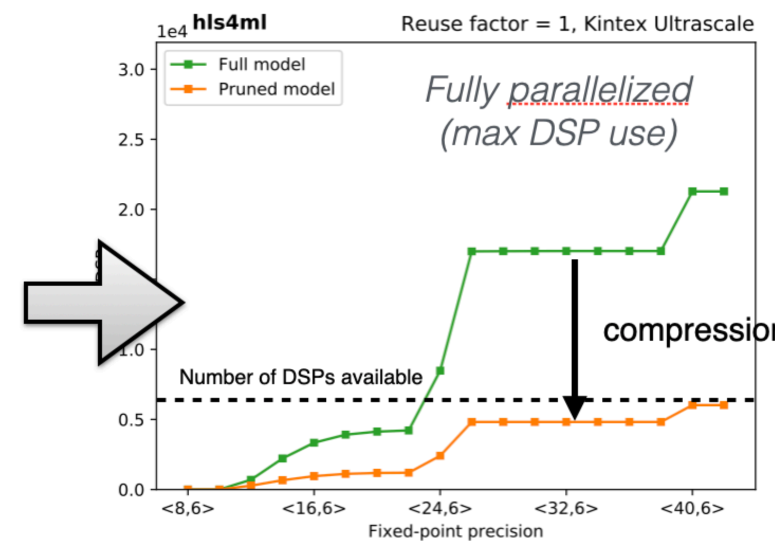
Fractional bits fixed to 8



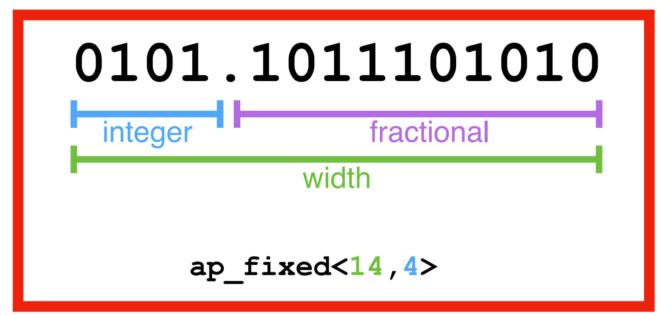
Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

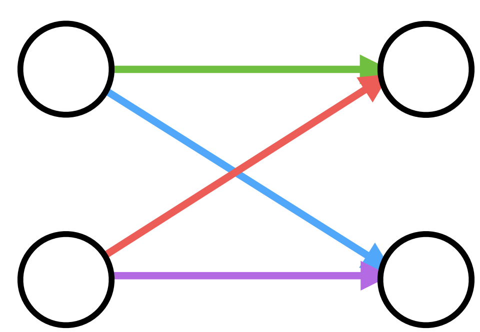


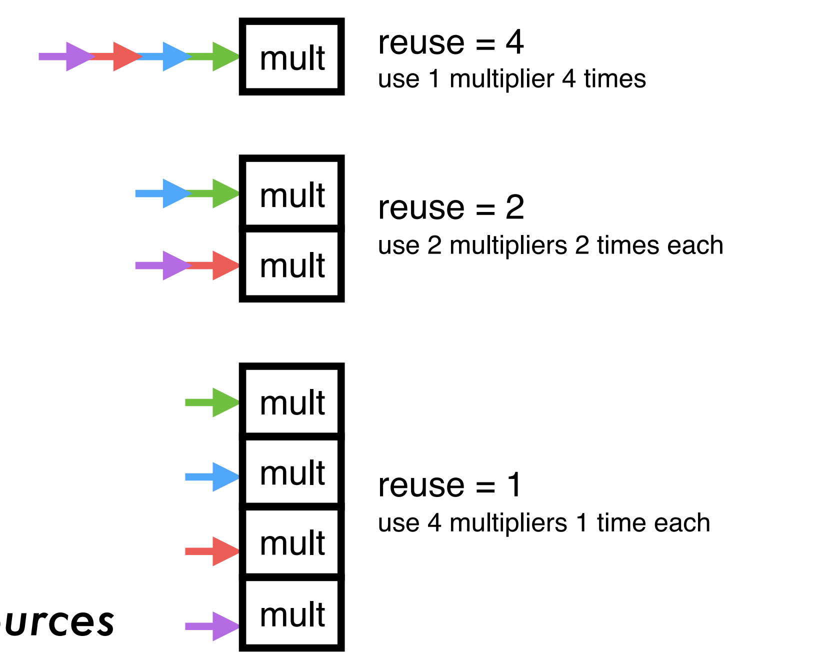


  - **Quantisation:** represents numbers with few bits reduce resources



  - **Parallelization:** allocate resources for each operation (run all network in one clock) vs spread calculation across several clock cycles

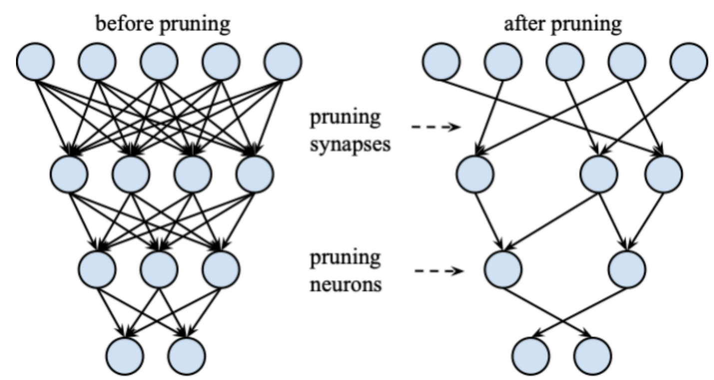


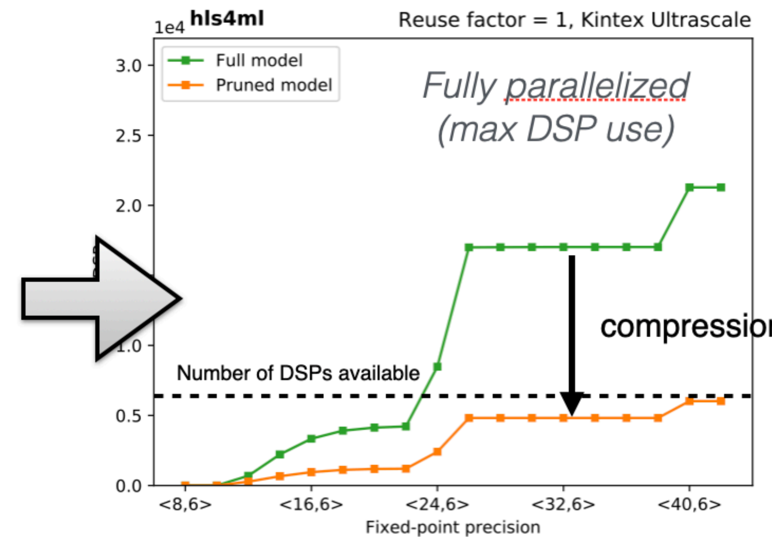


more parallelization → more resources

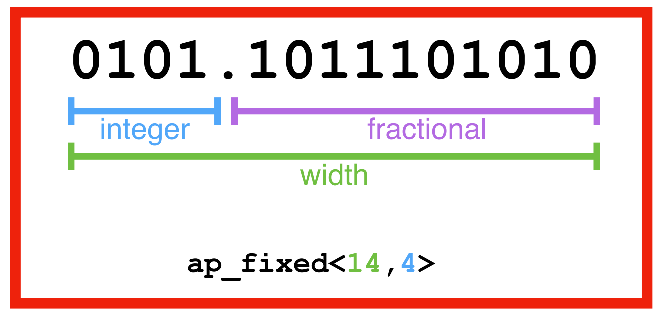
Make the model fit on one chip

- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision

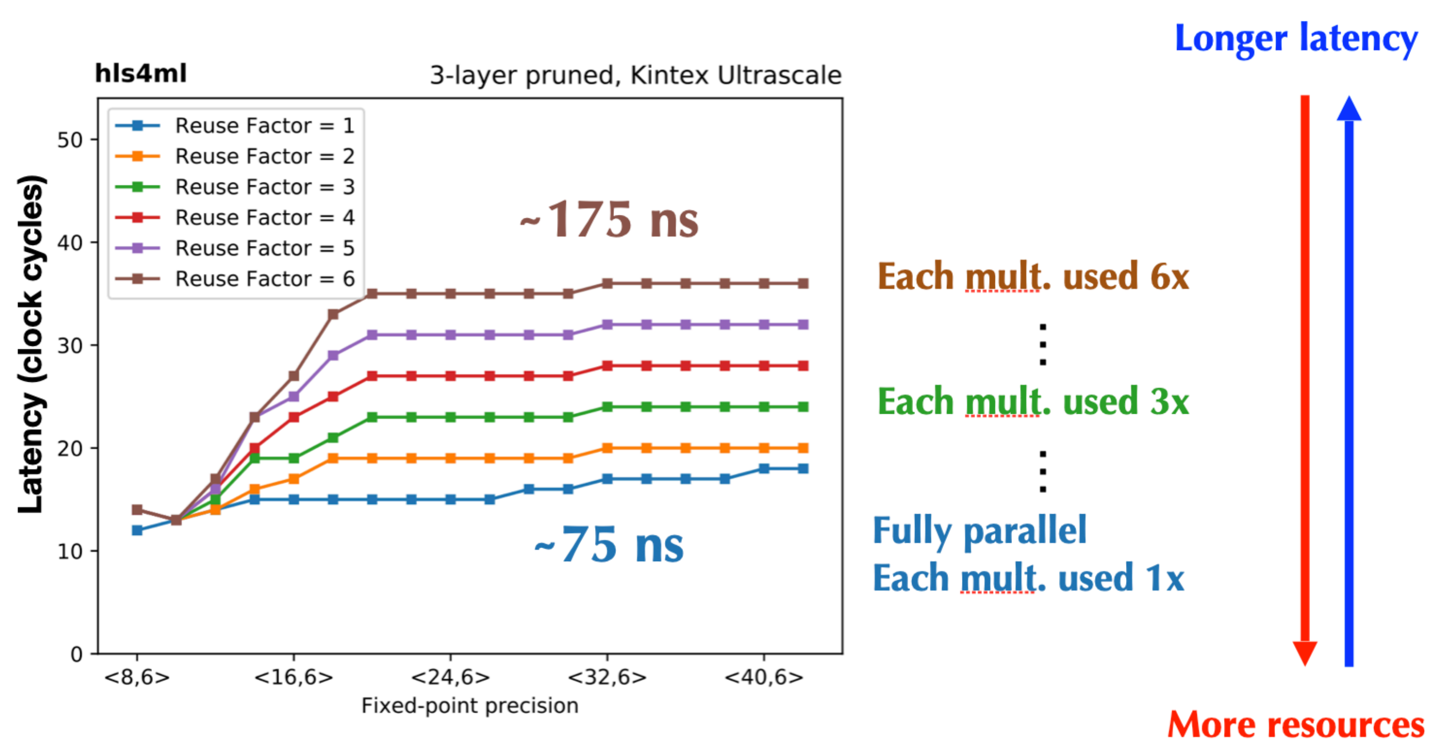




  - **Quantisation:** represents numbers with few bits reduce resources

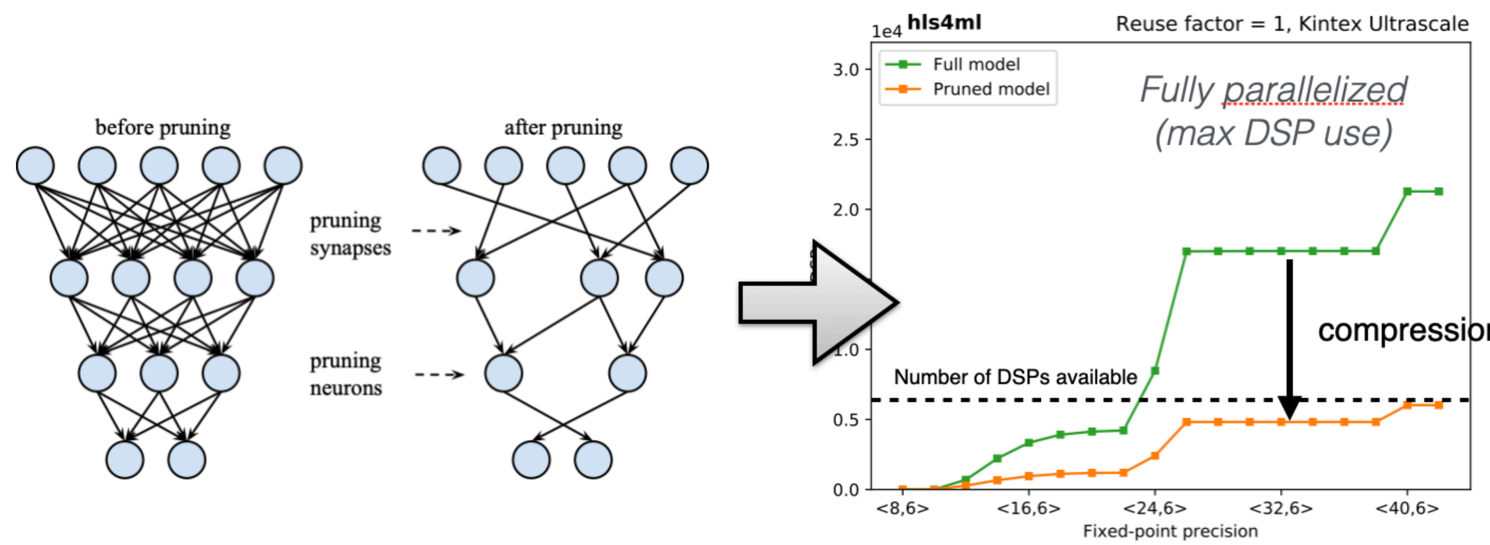


  - **Parallelization:** allocate resources for each operation (run all network in one clock) vs spread calculation across several clock cycles

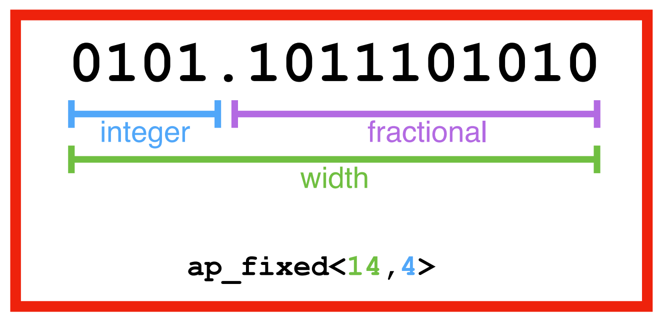


Make the model fit on one chip

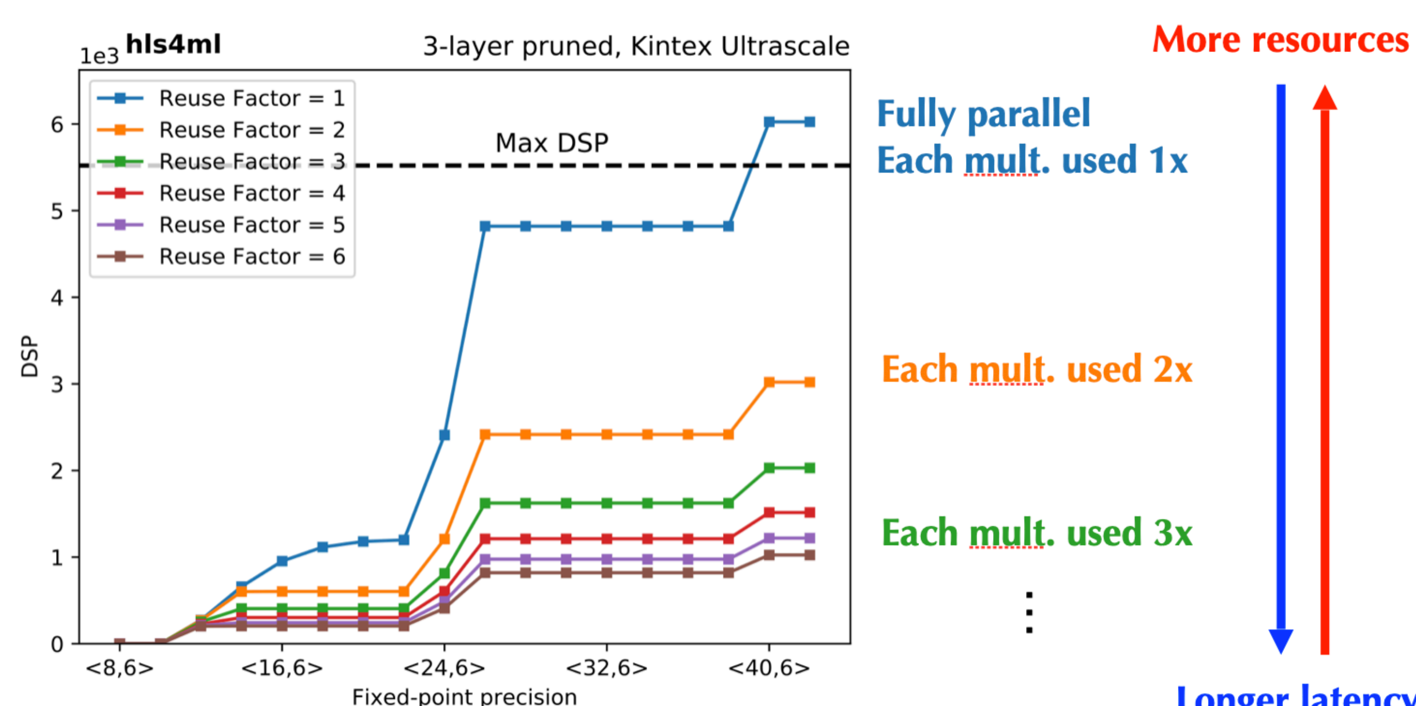
- Some tricks are needed here:
  - **Compression/pruning:** remove the connections that play little role for final decision



  - **Quantisation:** represents numbers with few bits reduce resources

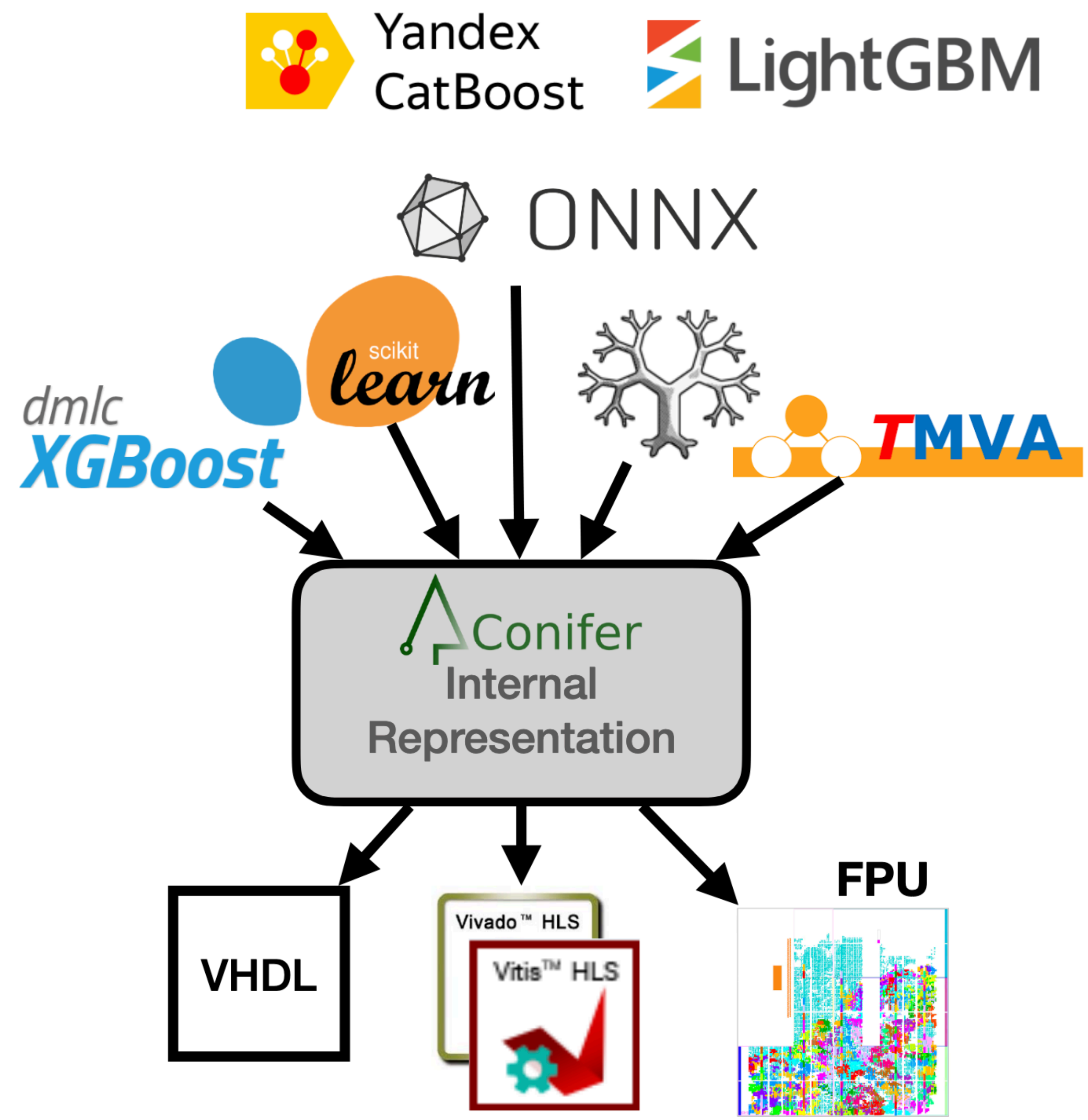


  - **Parallelization:** allocate resources for each operation (run all network in one clock) vs spread calculation across several clock cycles



The Conifer tool for BDTs

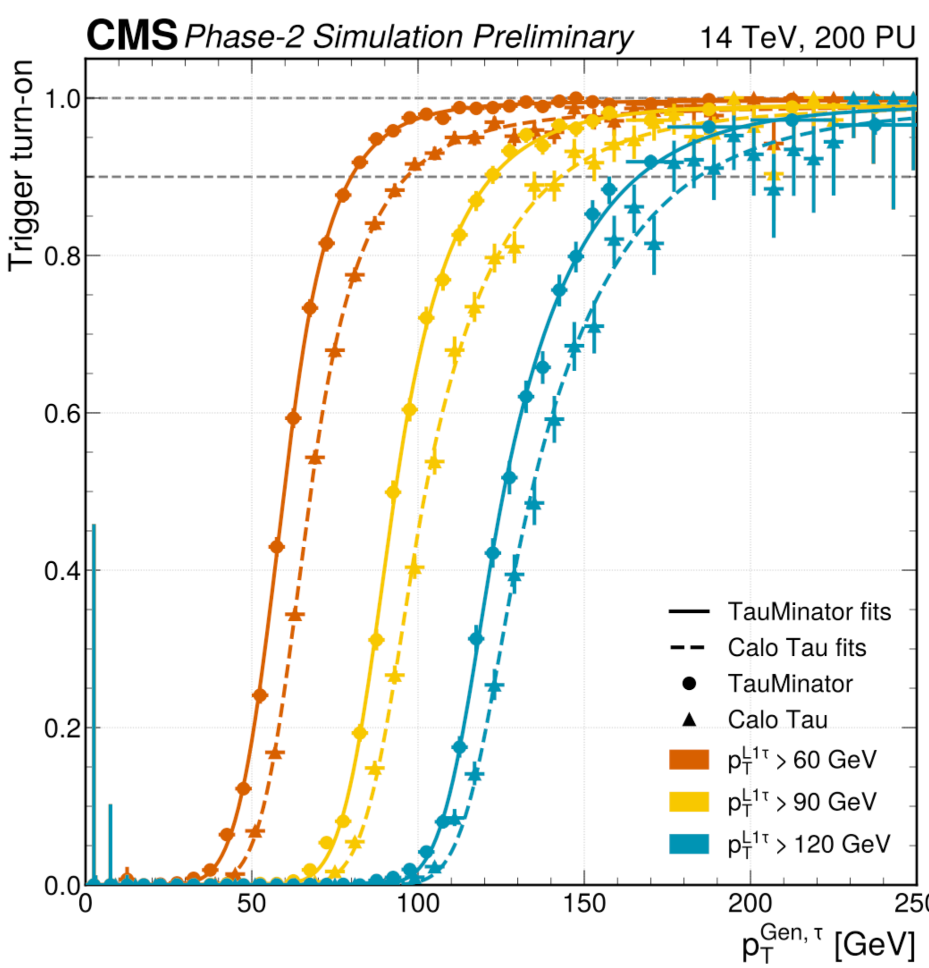
- Conifer is to DFs as hls4ml is to NNs
- Very much like hls4ml, conifer has frontends, an Internal Representation, and backends
- Frontend support for popular BDT training libraries
- Backends: HLS, (hand-written) VHDL, Forest Processing Unit (FPU)
- Conifer maps DFs onto FPGA logic: Implemented with high parallelism for low latency and high throughput

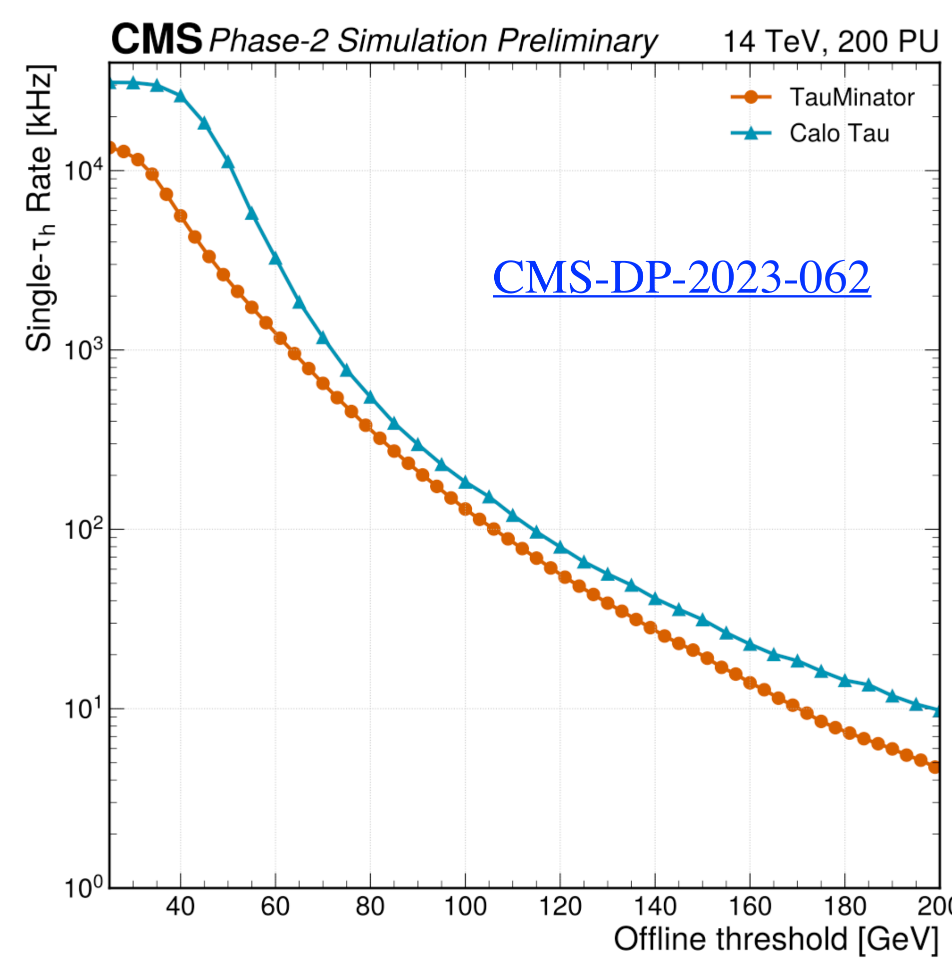


A few applications at the LHC

Hadronic τ reconstruction

- **2-layers 2D CNN for ID and calibration** with 2D images of seeded calorimeter clusters
    - for the HGCAL endcap additional inputs of 3D cluster shape included
- Quantization and pruning applied to achieve **55.6 ns latency @ 360 MHz and < 1% DSPs** on VU13P AMD chip for a single instance of the NN





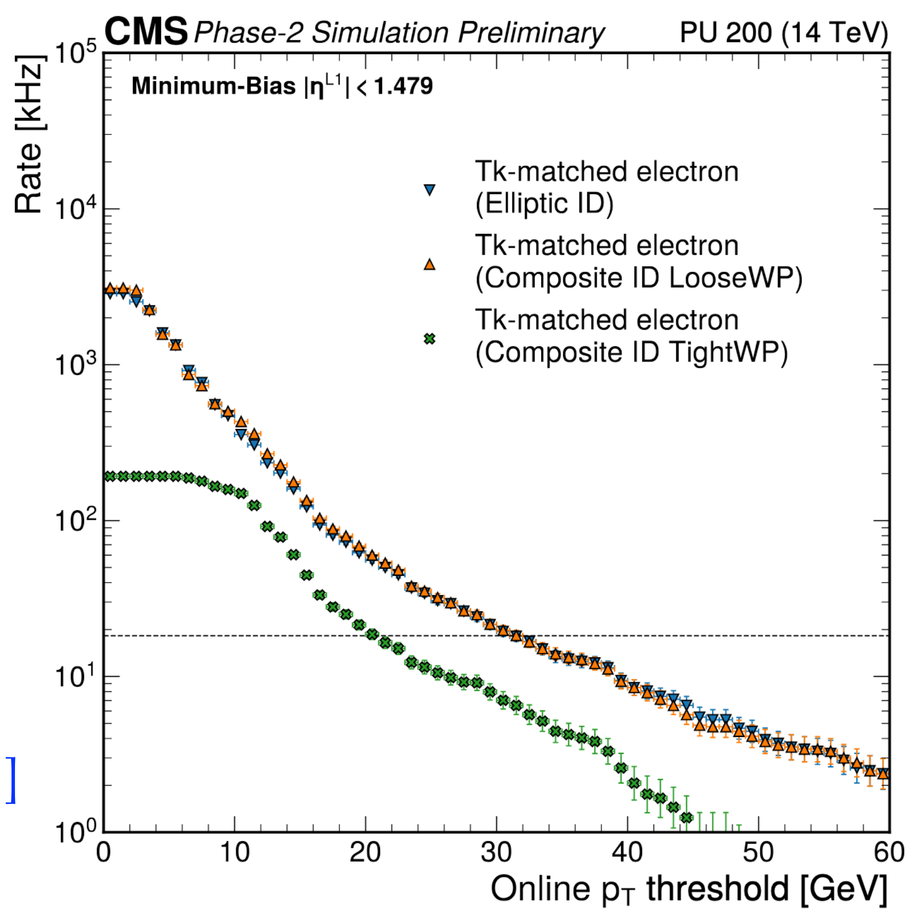
[CMS-DP-2023-062](CMS-DP-2023-062)

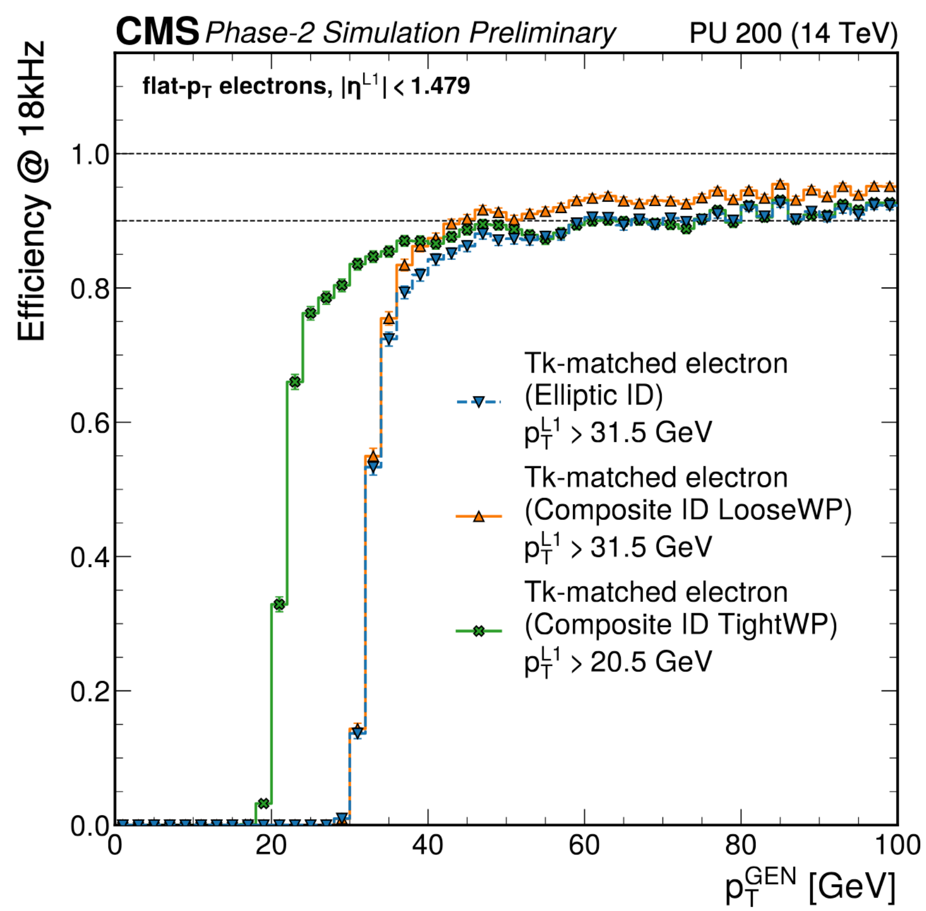
A few applications at the LHC

Electron identification

- PF electrons will be reconstructed by linking a track with a calorimeter cluster
- Baseline kinematic approach used distance and p_T compatibility to make a link
- **New BDT approach** combines calorimeter cluster shape variables, track qualities, and track-matching features
- Improved electron reconstruction efficiency at **27.8 ns latency @ 180 MHz and < 1% DSPs** on VU13P AMD chip

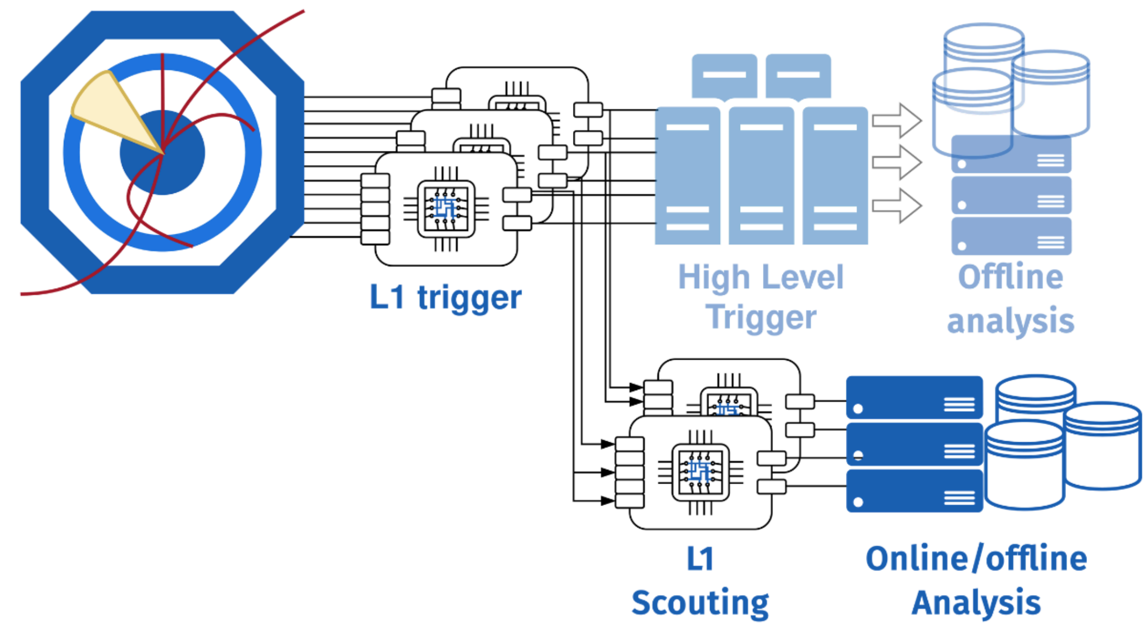
[DP Note soon!]





The CMS L1 Scouting system

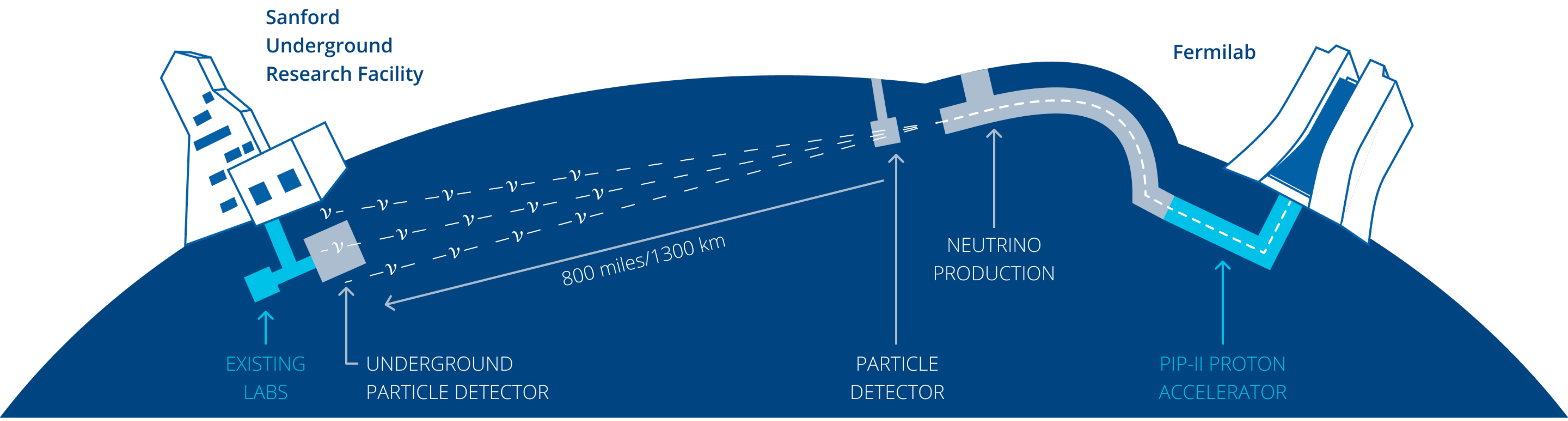
- **L1T Data Scouting:** acquire and analyse the L1 Trigger information for all events
- Look for physics signatures identifiable with **just coarse L1 information** but that would evade the L1T → HLT → Offline chain, e.g.:
  - too large "irreducible" backgrounds, e.g. narrow resonances of low mass
  - complex signatures exceeding the computing capabilities of the L1 system
  - signal identification requires time-correlation across several BXs, e.g. slow or long-lived BSM



- FPGA-equipped boards that receive L1 data via optical links and transfer it to PCs and the software world via TCP/IP or PCI express
- At HL-LHC: can profit from much improved L1T object reconstruction quality
- **However, prohibitive downstream bandwidth and storage** → to store all L1 info at 40 MHz a factor O(10) compression/reduction needed
  - opportunity to explore AI methods for data reduction or compression, e.g. through SSL

Big data @ the Intensity Frontier

The Deep Underground Neutrino Experiment (DUNE)



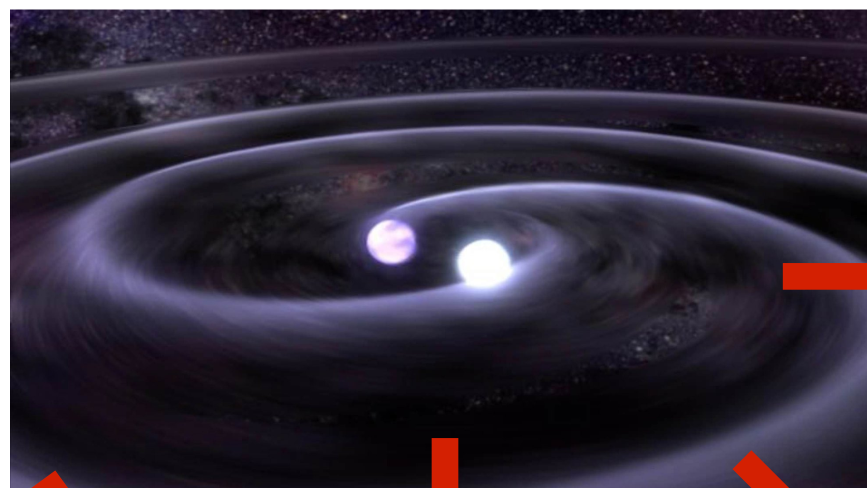
- Next generation neutrinos oscillation experiment now under construction and R&D to start operations in late 2020s
- Massive far detector 1 mile underground comprising **70k tons of Liquid Argon** and advanced technology to record neutrino interactions with extraordinary precision
- Uncompressed continuous readout of modules will yield **O(100) Tb/s** → unprecedented for this type of experiment!

Multi-messenger astronomy

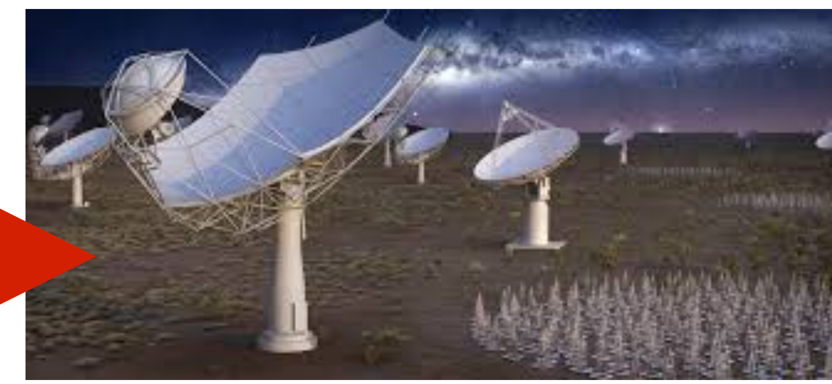
Multi-messenger astronomy **probes the Universe using different cosmic messengers**

X-rays/Gamma-rays

Cosmic event
(ex, binary neutron star merger)



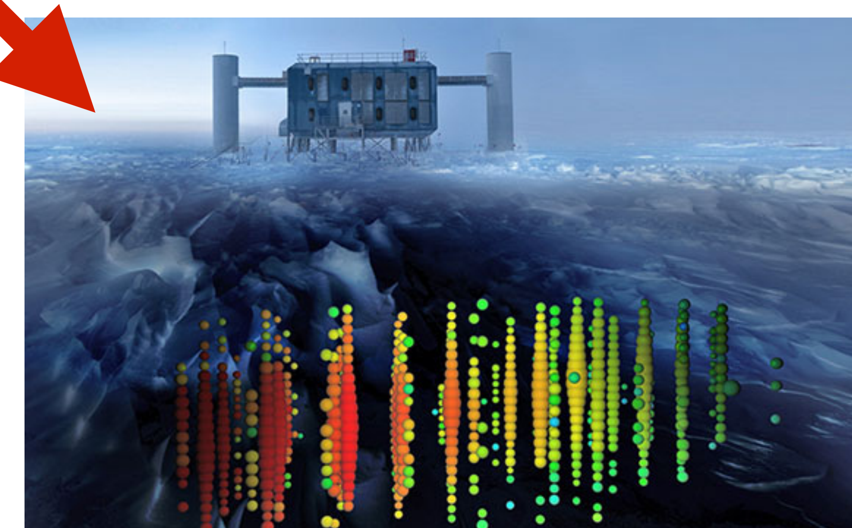
Radio telescope



Visible/Infrared light

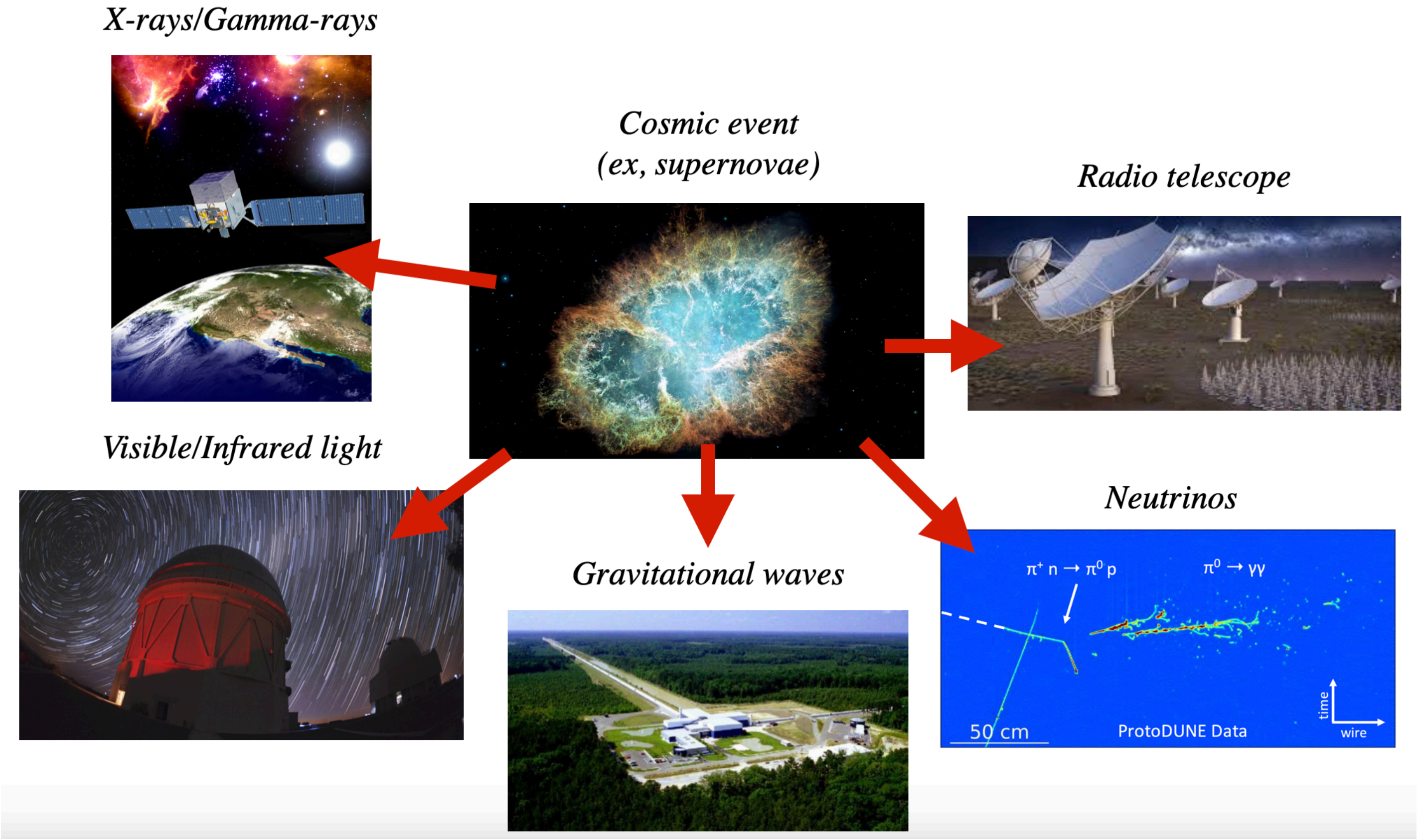
Gravitational waves

Neutrinos



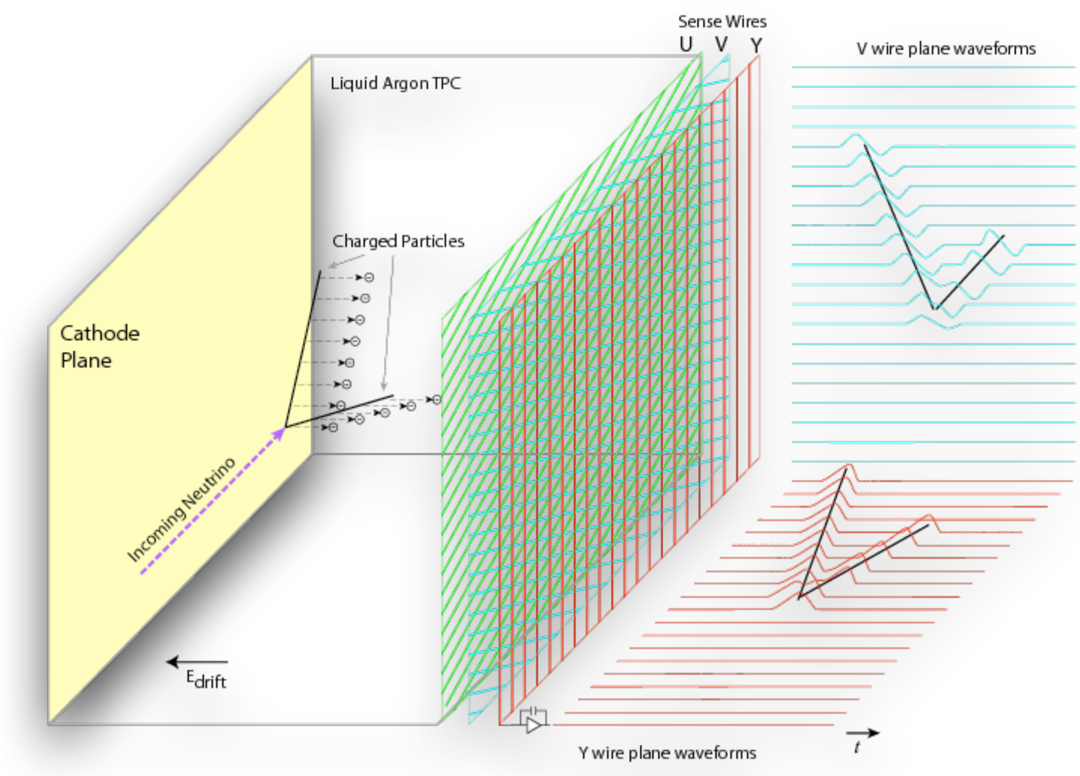
Multi-messenger astronomy w/ Neutrinos

- **Core-collapse supernovae are a huge source of neutrinos of all flavours**
  - 99% of energy released is carried away by neutrinos
- Rich information embedded in neutrino signal plus associated gravitational and electromagnetic signals
  - **supernova physics:** core-collapse mechanism, black hole formation, nucleosynthesis, …
  - **particle physics:** flavor transformation in SN core, mass ordering, BSM…
- **Detection and pointing in real-time in large scale neutrino experiments is an active field of research!**

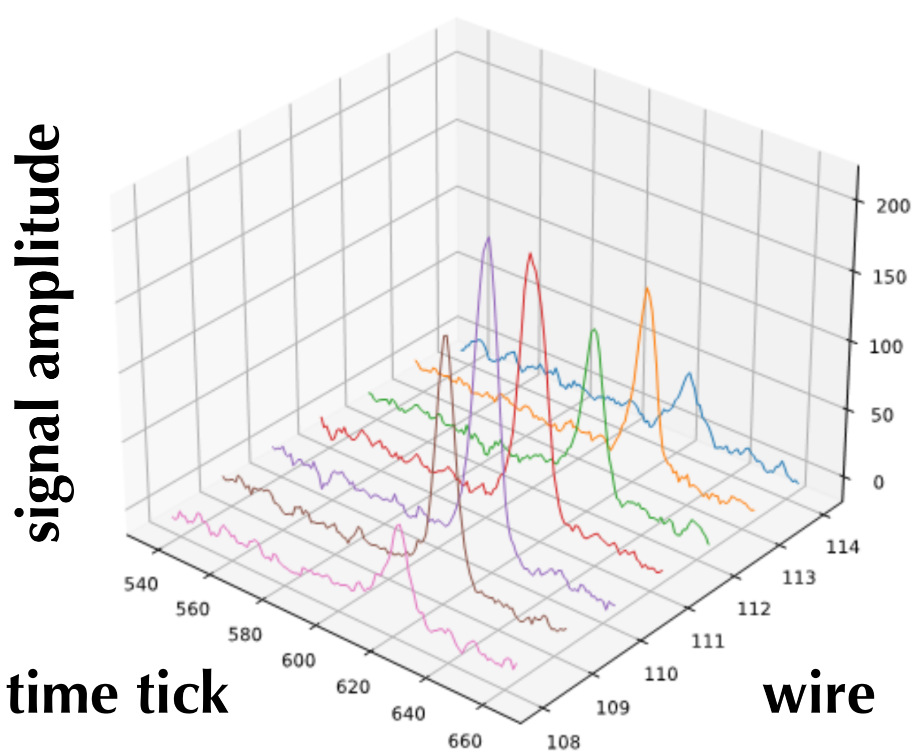


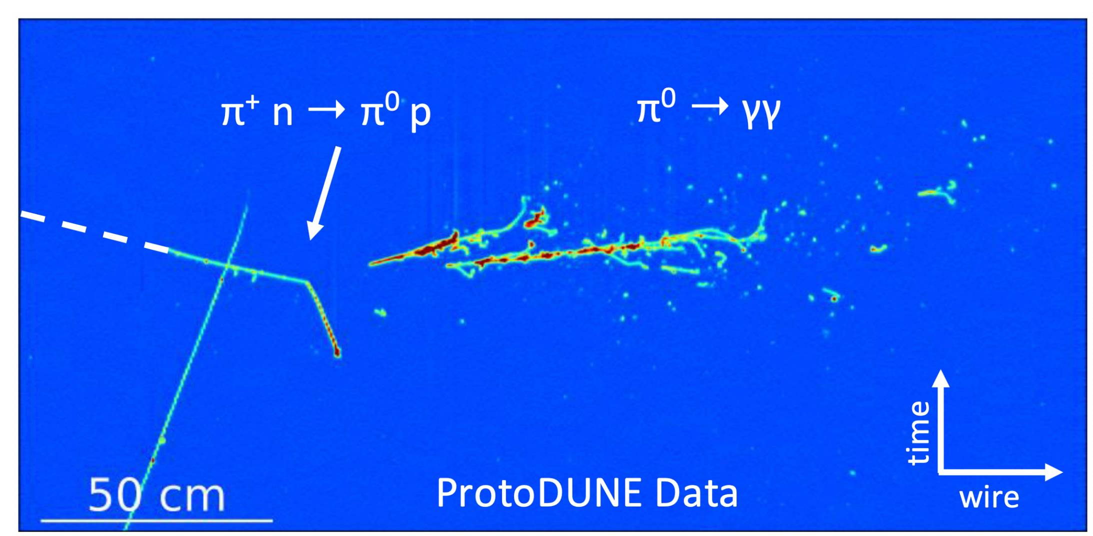
Big data @ the Intensity Frontier

Operating principle of a LArTPC



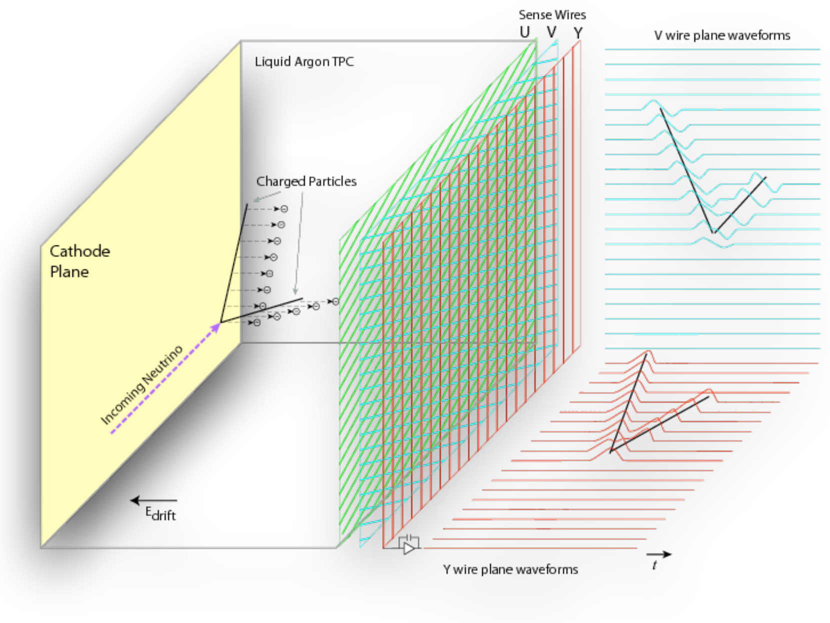
- Neutrinos interacting with the LAr produce **charged particle**s, which in turn produce **electrons**
- Electrons are **collected by anode wires**
- The **signal** from each wire channel is a **wave form**
- There are **3 planes of wires** for a full 3D reconstruction of the interaction
- The result is a continuous stream of 3D images of detector volume yielding a **high-resolution "video"**





Big data @ the Intensity Frontier

Detector
South Dakota

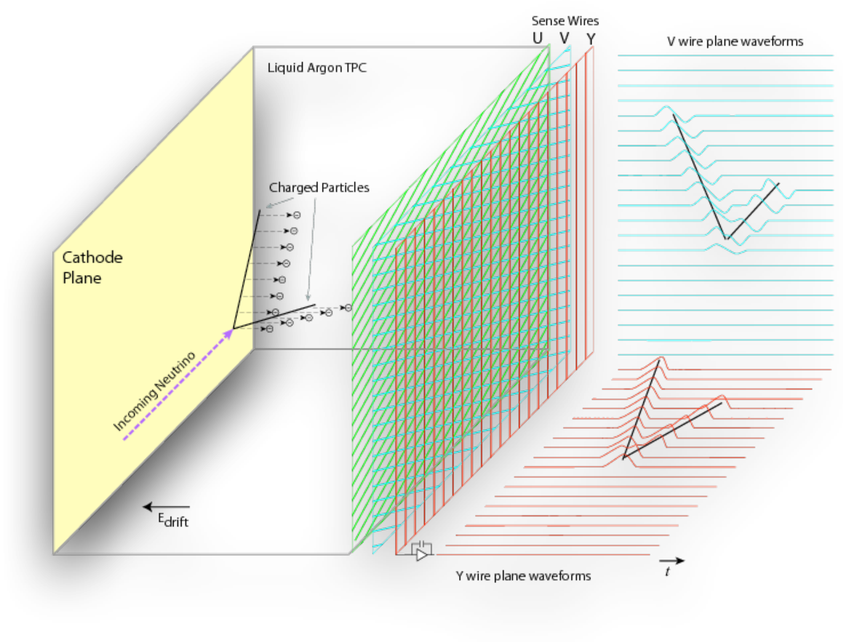


4.8 TB/s
100 seconds: 480 TB
100 Gbps

HPC
Fermilab, Illinois

Big data @ the Intensity Frontier

Detector
South Dakota



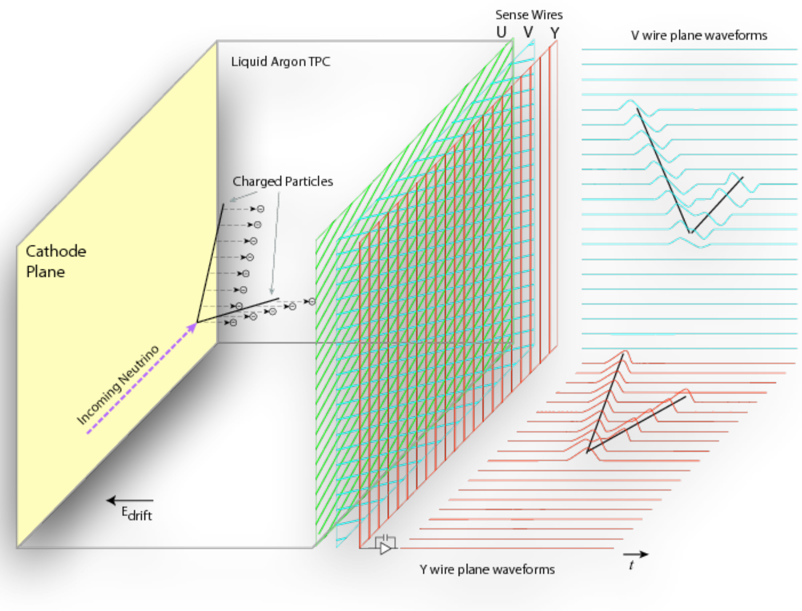
4.8 TB/s
100 seconds: 480 TB
100 Gbps

At least 12 hours before we can detect a supernova and reconstruct point of origin!

HPC
Fermilab, Illinois

Big data @ the Intensity Frontier

Detector
South Dakota



4.8 TB/s
100 seconds: 480 TB
100 Gbps

At least 12 hours before we can detect a supernova and reconstruct point of origin!

HPC
Fermilab, Illinois

- Aggressive data reduction must happen underground **close to the data source**
- Must be smart as neutrinos from supernova are challenging → **Machine Learning**
- Very limited power underground requires dedicated hardware → **FPGAs**