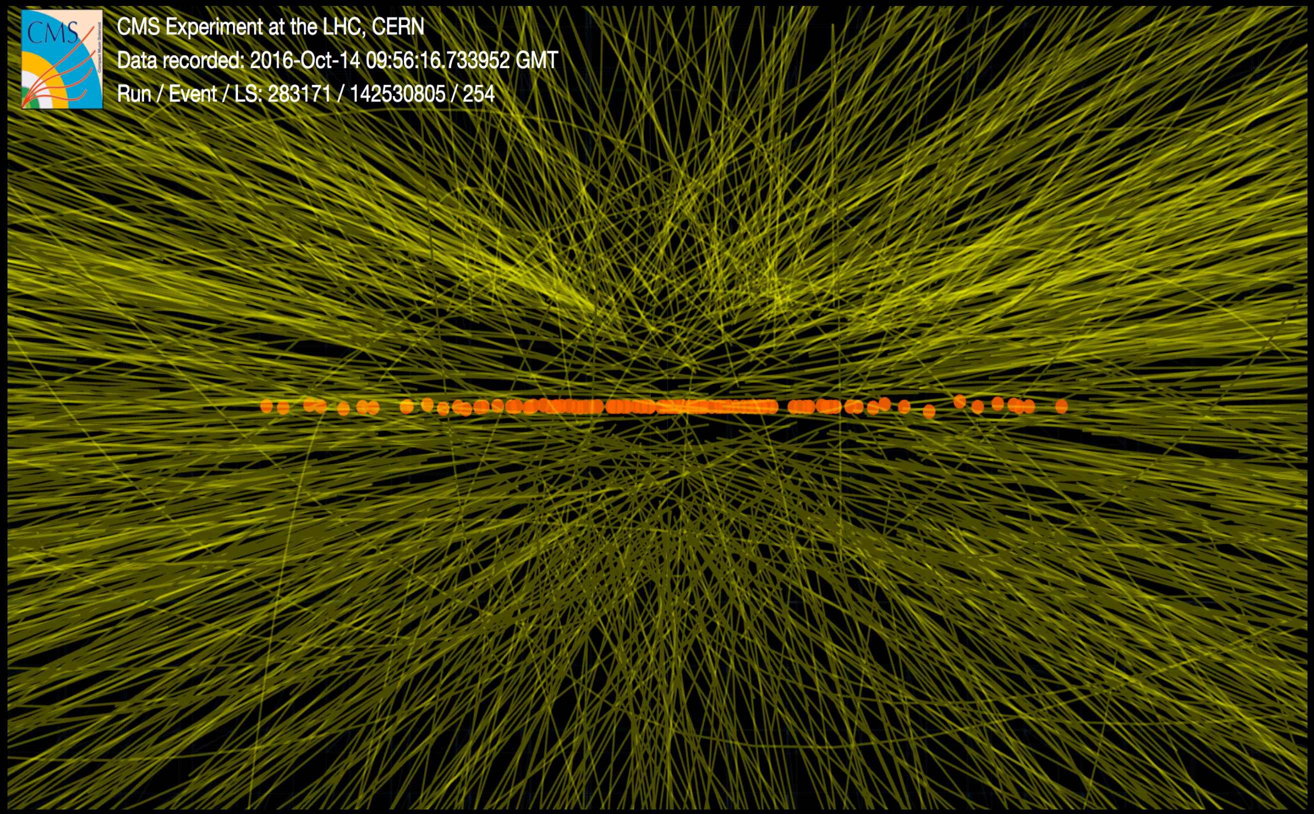
Deep learning and future challenges at the High-Luminosity LHC

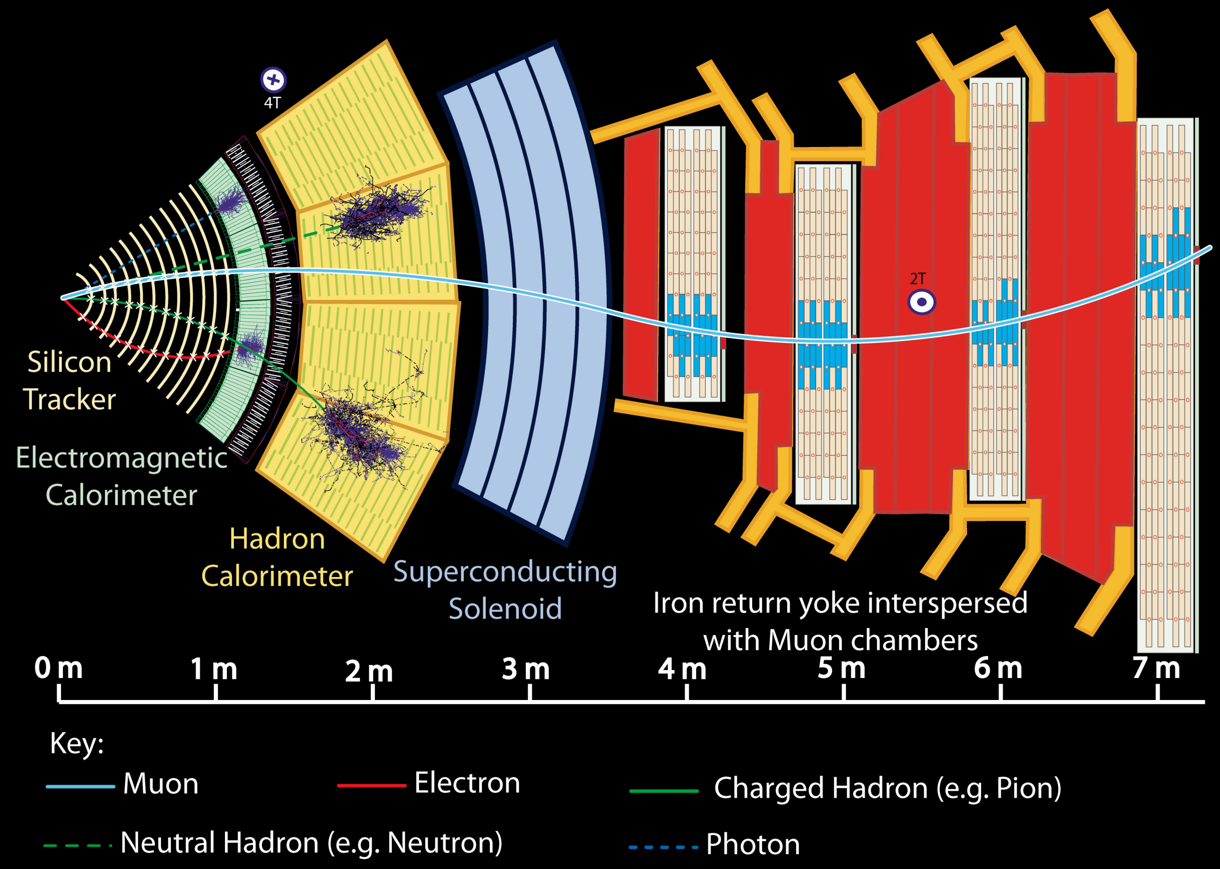
Jennifer Ngadiuba (CERN)

12th Terascale Detector Workshop
12-15 March, 2019, TU Dresden, Physics Department

The LHC big data problem

ex, Compact Muon Solenoid

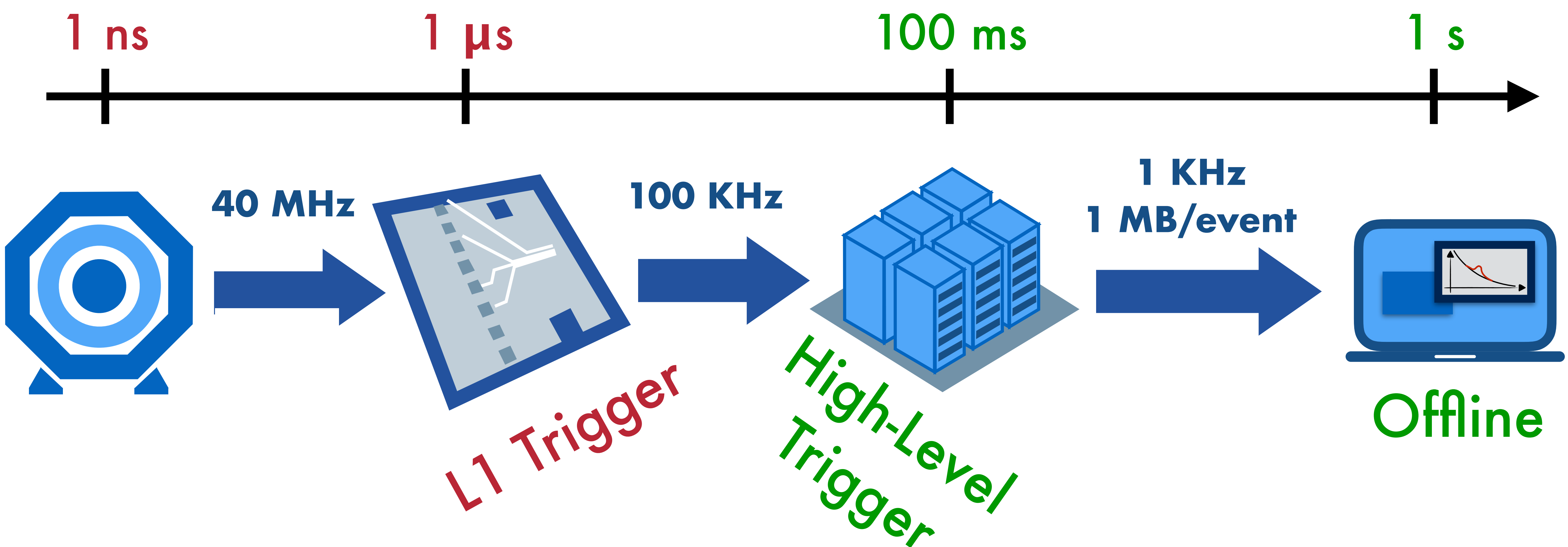




At the LHC the proton beams collide at a frequency of 40 MHz
Each collision produces $O(10^3)$ particles!
The detectors have $O(10^8)$ sensors used to detect these particles
Extreme data rates of O(100 TB/s)!

Event processing @ LHC

Reduce data rates to manageable levels for offline processing
by filtering events through multiple stages:



Absorbs 100s TB/s

Trigger decision to be made in O(µs)

Latencies require all-FPGA design

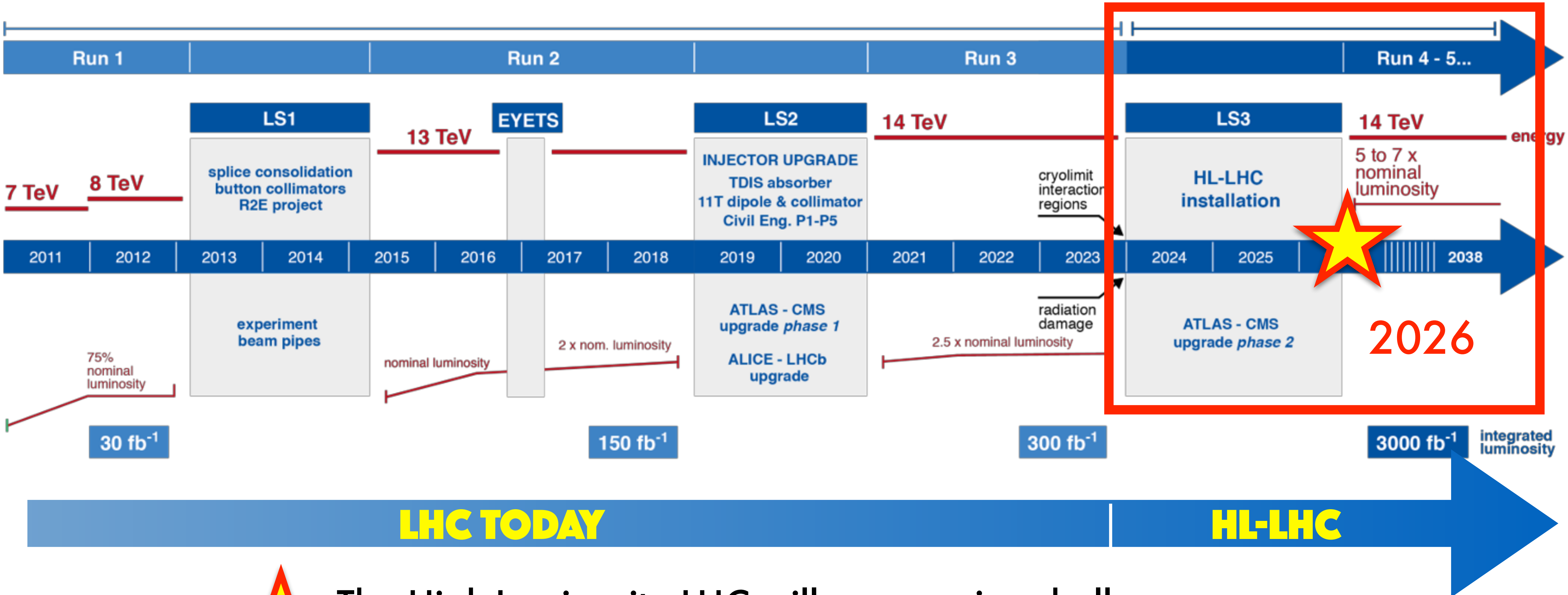
99.75% events rejected!

Analysis of the full event runs on commercial computers (30k CPU cores)

Latency O(100 ms)

99% events rejected!

The HL-LHC challenge



The High-Luminosity LHC will pose major challenges:

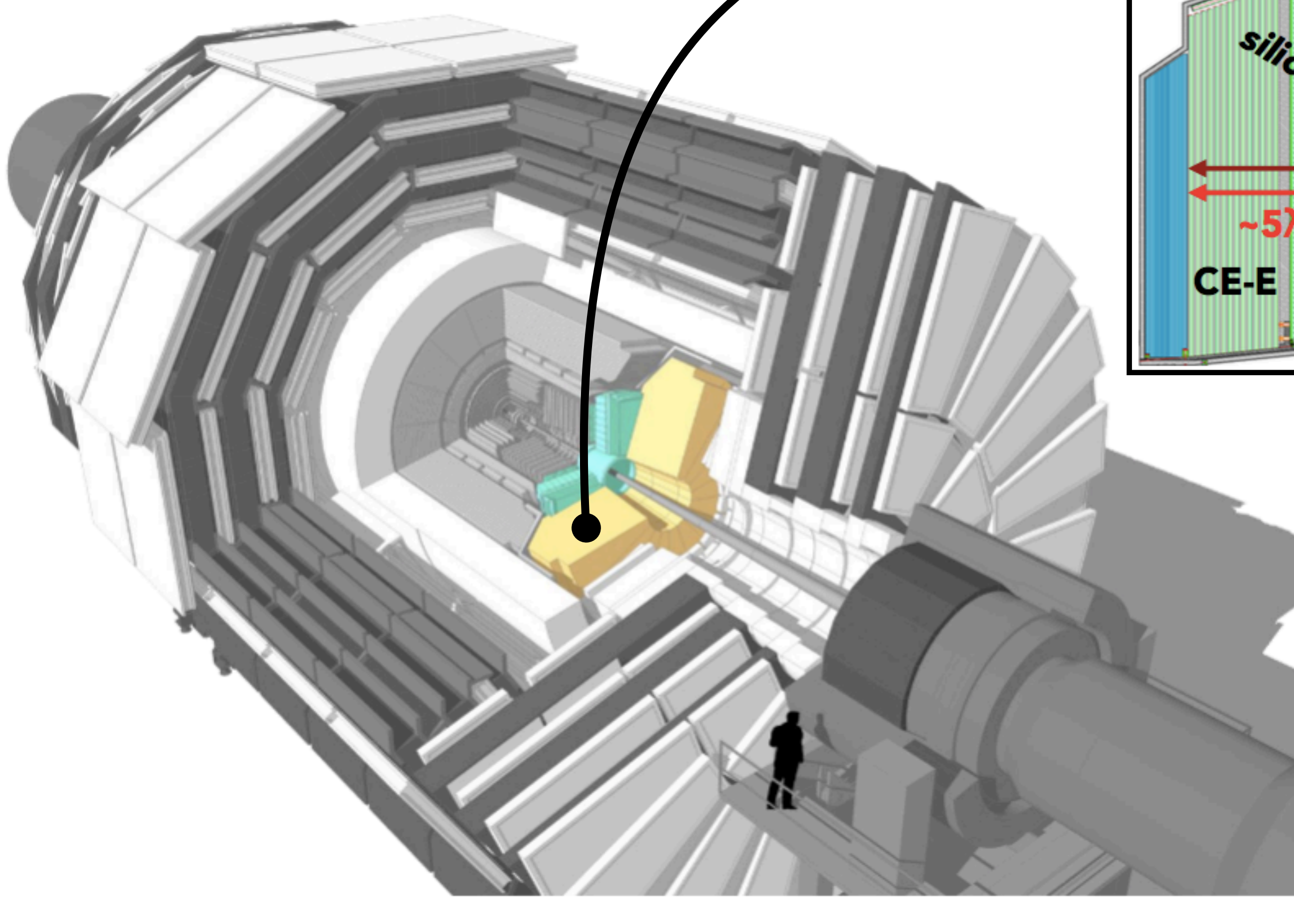
instantaneous luminosity x 5–7
particles per collision x 5
more data x 15
more granular detectors with x 10 readout channels

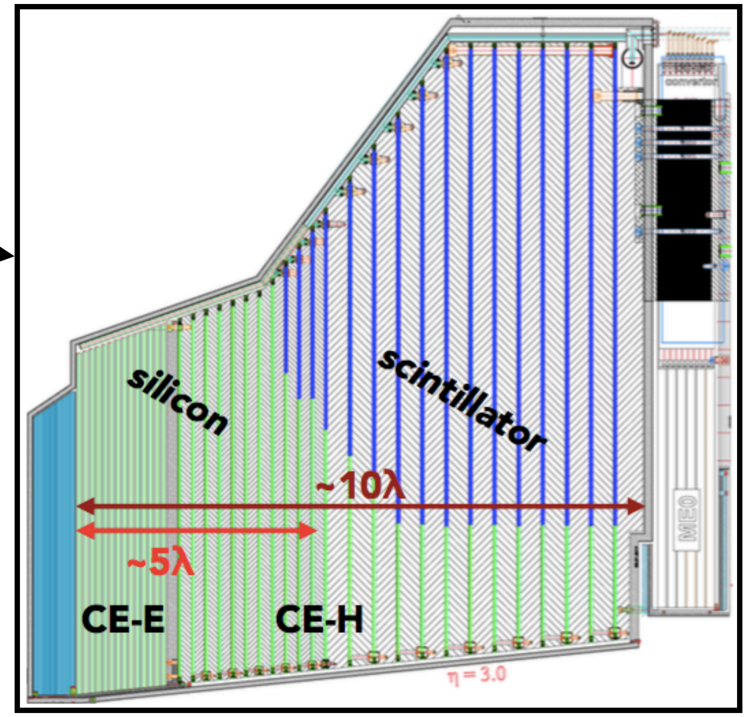
→ event rates & datasets will increase to unprecedented levels!

Detector upgrades for HL-LHC

ex: CMS High-granularity calorimeter

Novel technology for CMS endcap calorimeter:
52 layers with unprecedented
number of readout channels!





Total Silicon:
- 600 m^2

Total scintillator
- 500 m^2

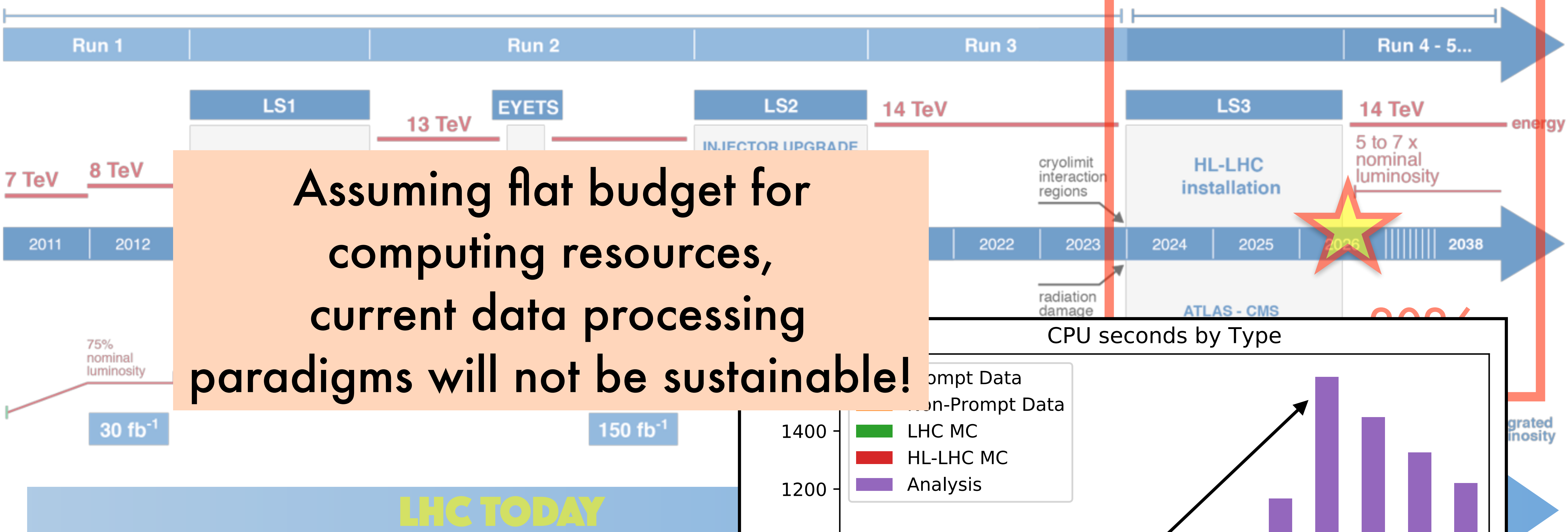
6 M Channels

CMS HGCAL TDR

	CMS	ATLAS	CMS HGCal
Diameter (m)	15	25	
Length (m)	28.7	46	
B-Field (T)	3.8	2/4	
EM Cal channels	~80,000	~110,000	4.3M
Had Cal channels	~7,000	~10,000	1.8M

P.Merkel

The HL-LHC challenge



Assuming flat budget for computing resources, current data processing paradigms will not be sustainable!

x 5

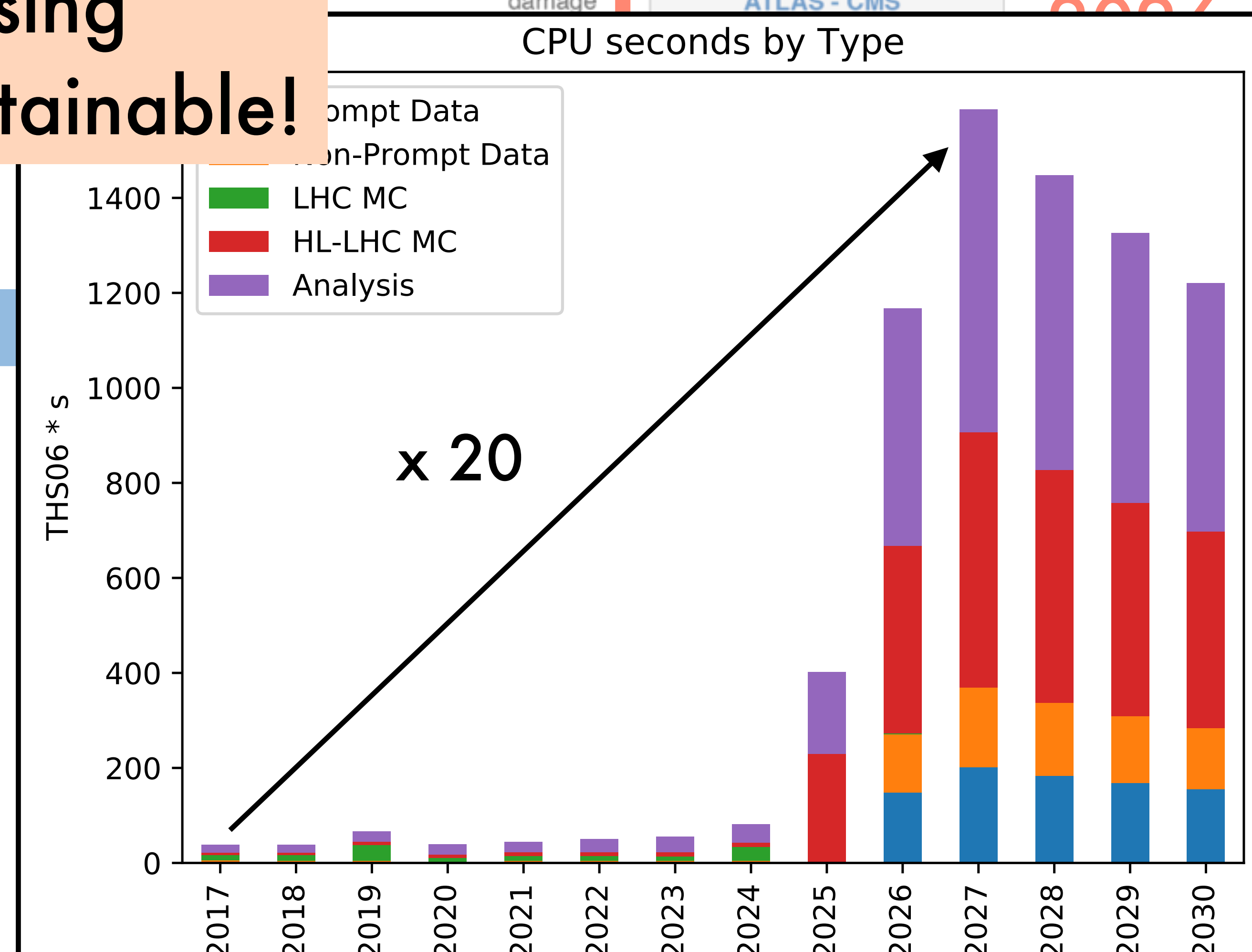
Event complexity

x 50

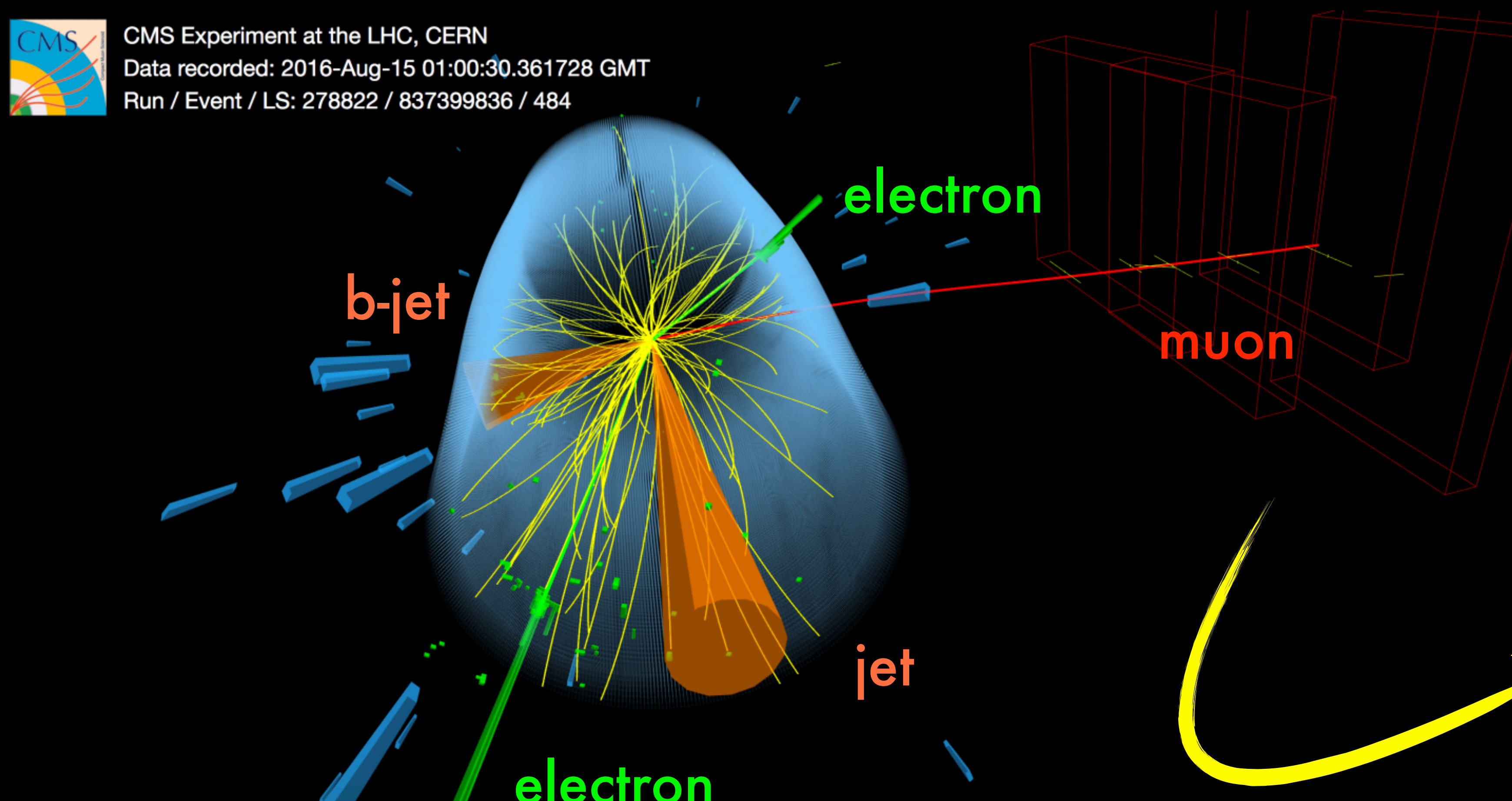
Processing time

x 20

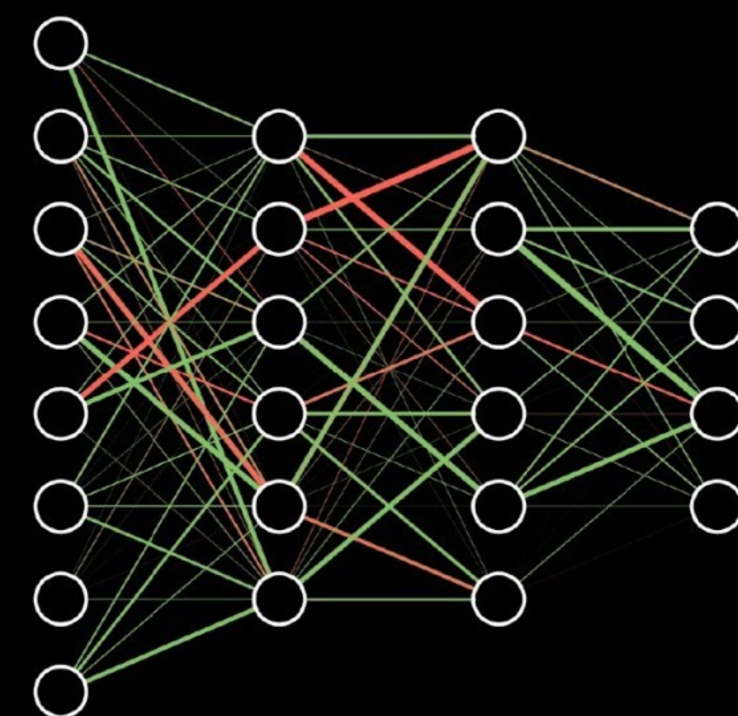
Computing resources/ Disk storage



Modern deep learning algorithms might be the way out!



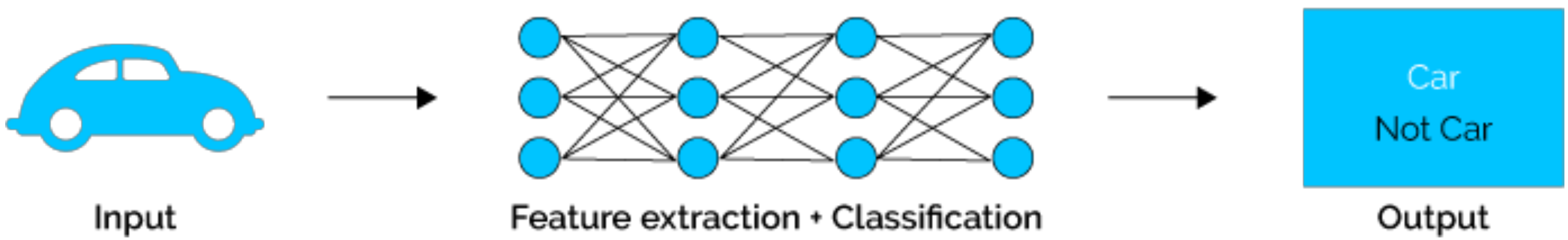
Event reconstruction in CMS



A neural network

Recast particle physics problem
into a machine learning problem!

What is machine learning?



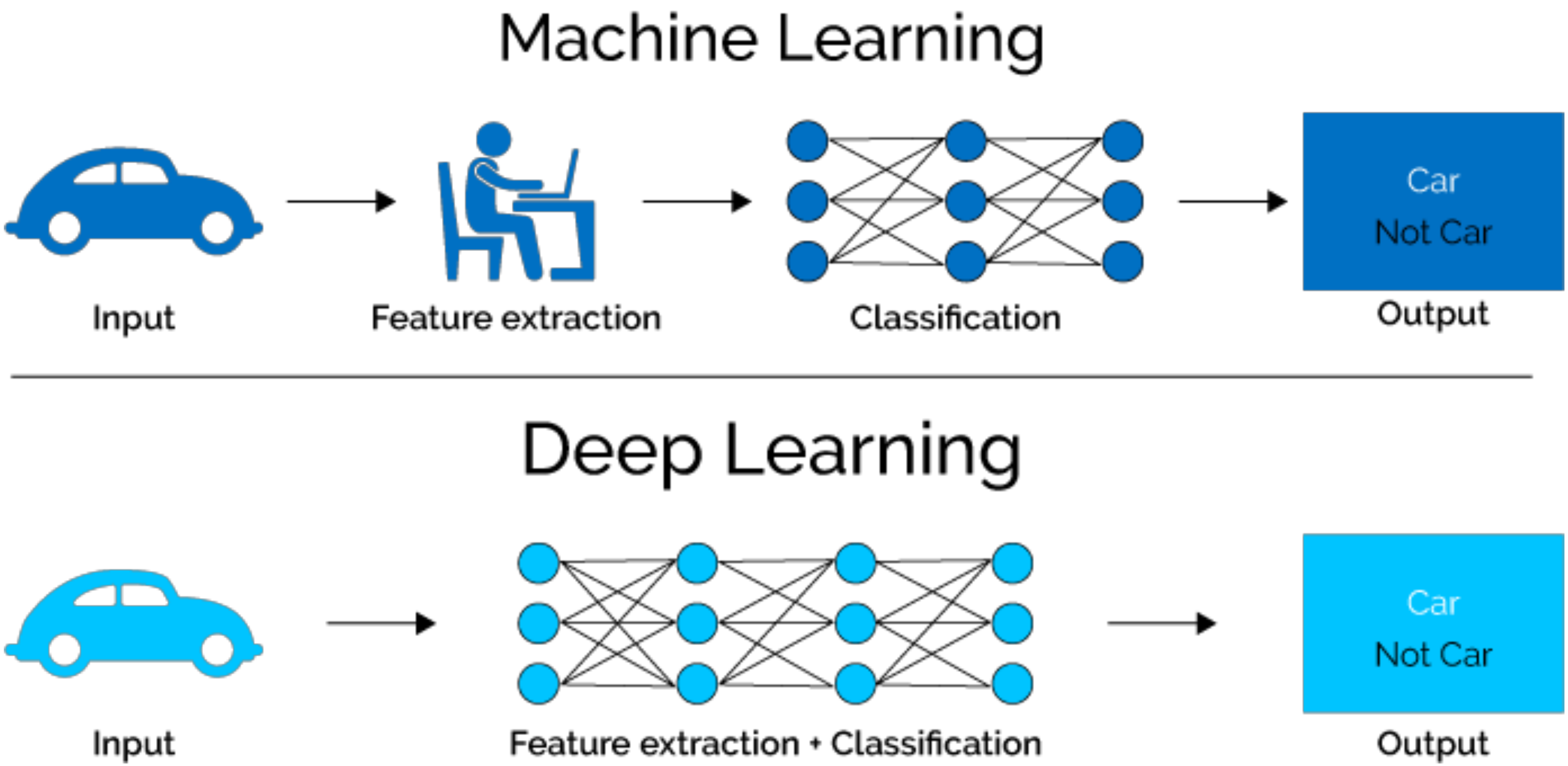
Learning mathematical model from input data that characterize patterns, regularities, and relationships among variables.

Three key components:

- **model** - chosen mathematical model (depends on the task and type of data)
- **learning** - estimate model from data
- **prediction** - use learnt model to make predictions on new data points (also called "inference")

<u>While training a ML algo can take a long time, the inference is usually very fast!</u>

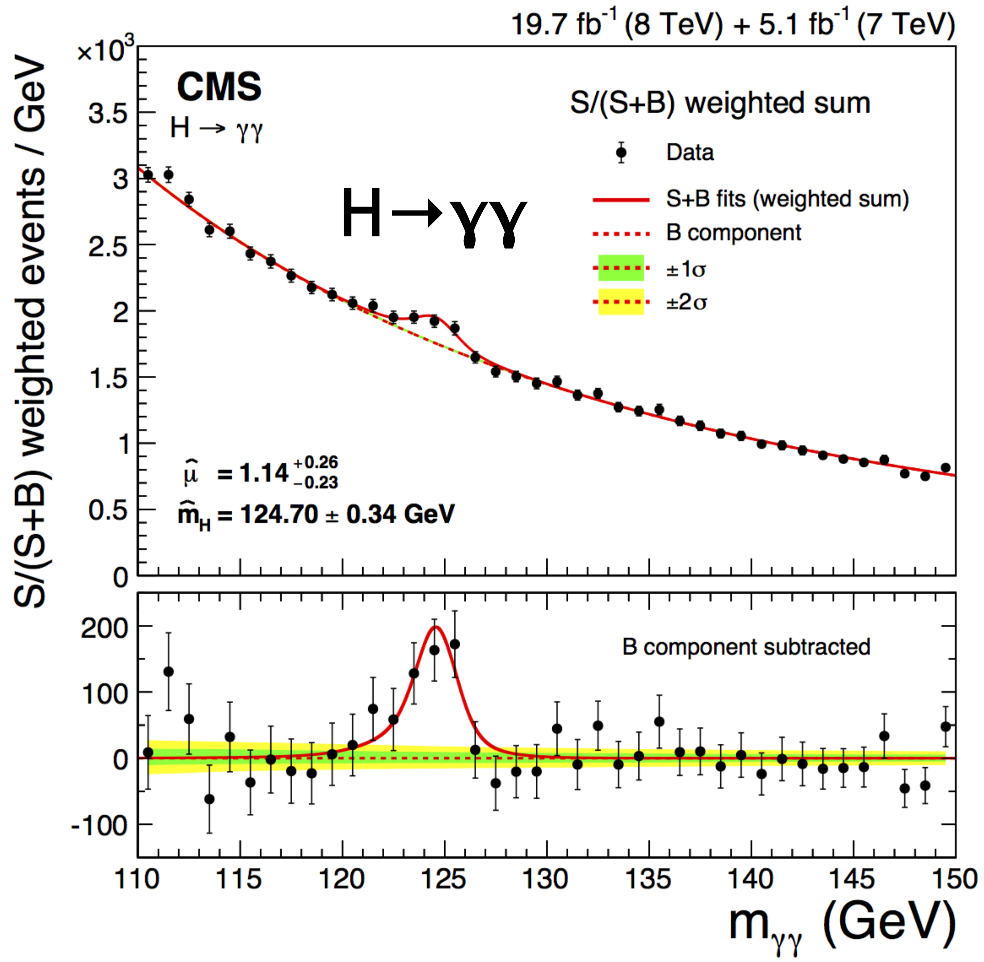
And deep learning?

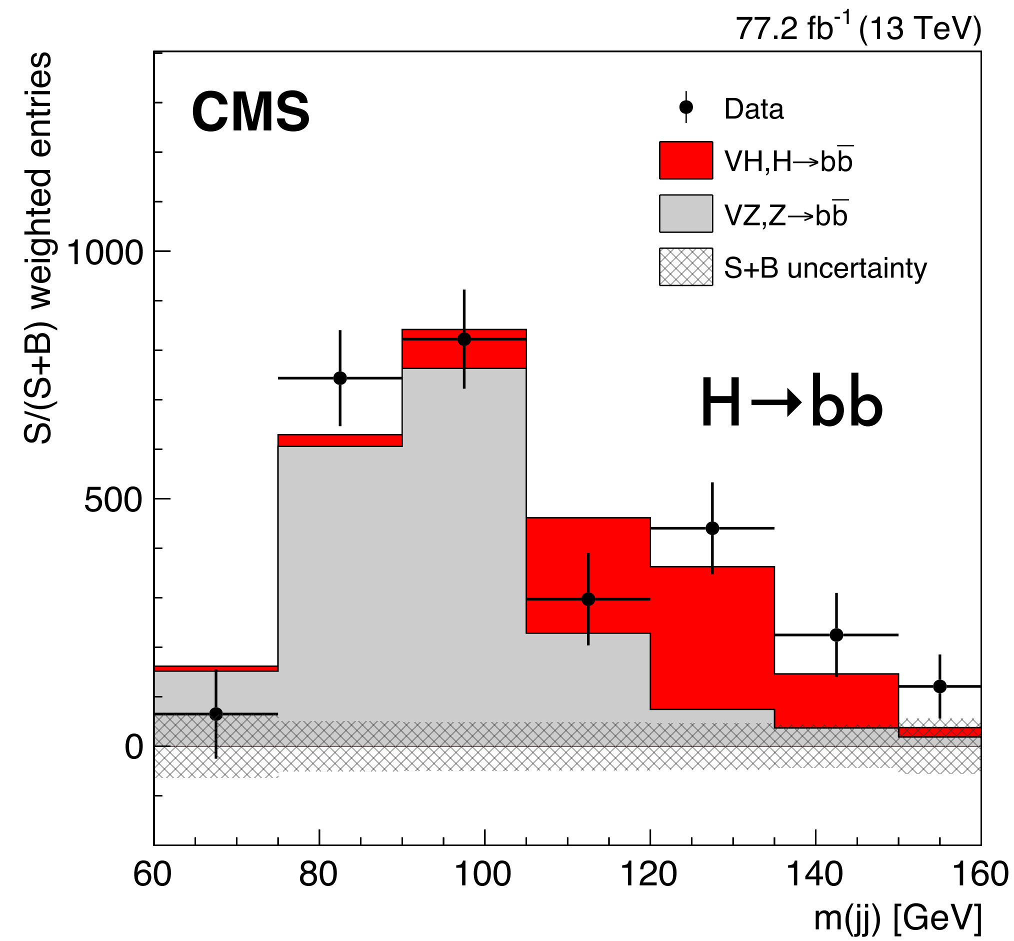


The success of ML in HEP

ML methods widely used in HEP
showing excellent physics performance in offline analysis

ex, Higgs boson discovery



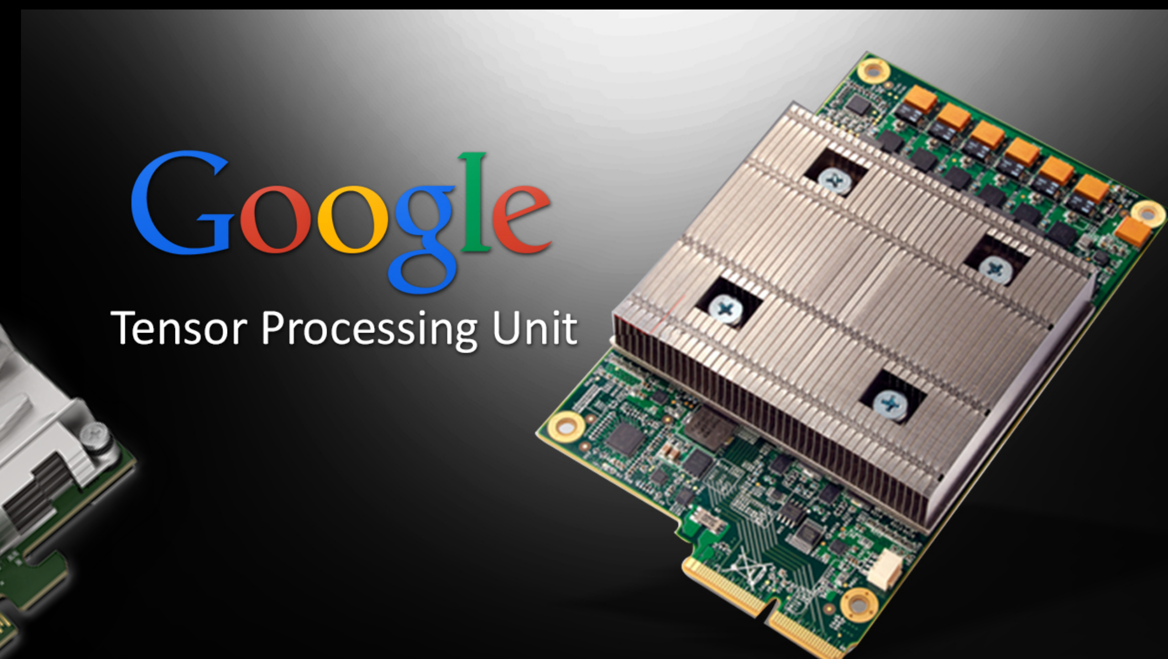


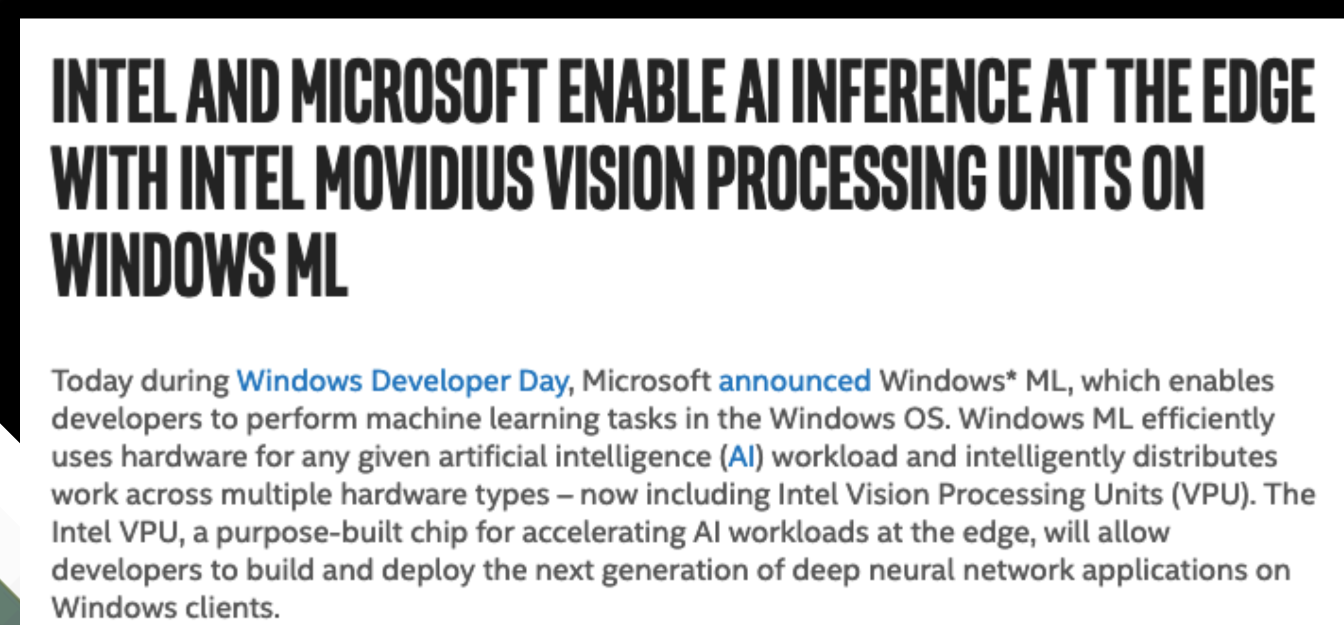
ML algorithms used offline for

- improving Higgs mass resolution with particle energy regression
- enhancing signal/background discrimination

HEP learning from industry

Take advantage of industry trends in developing new devices optimized for ML and speed up the inference

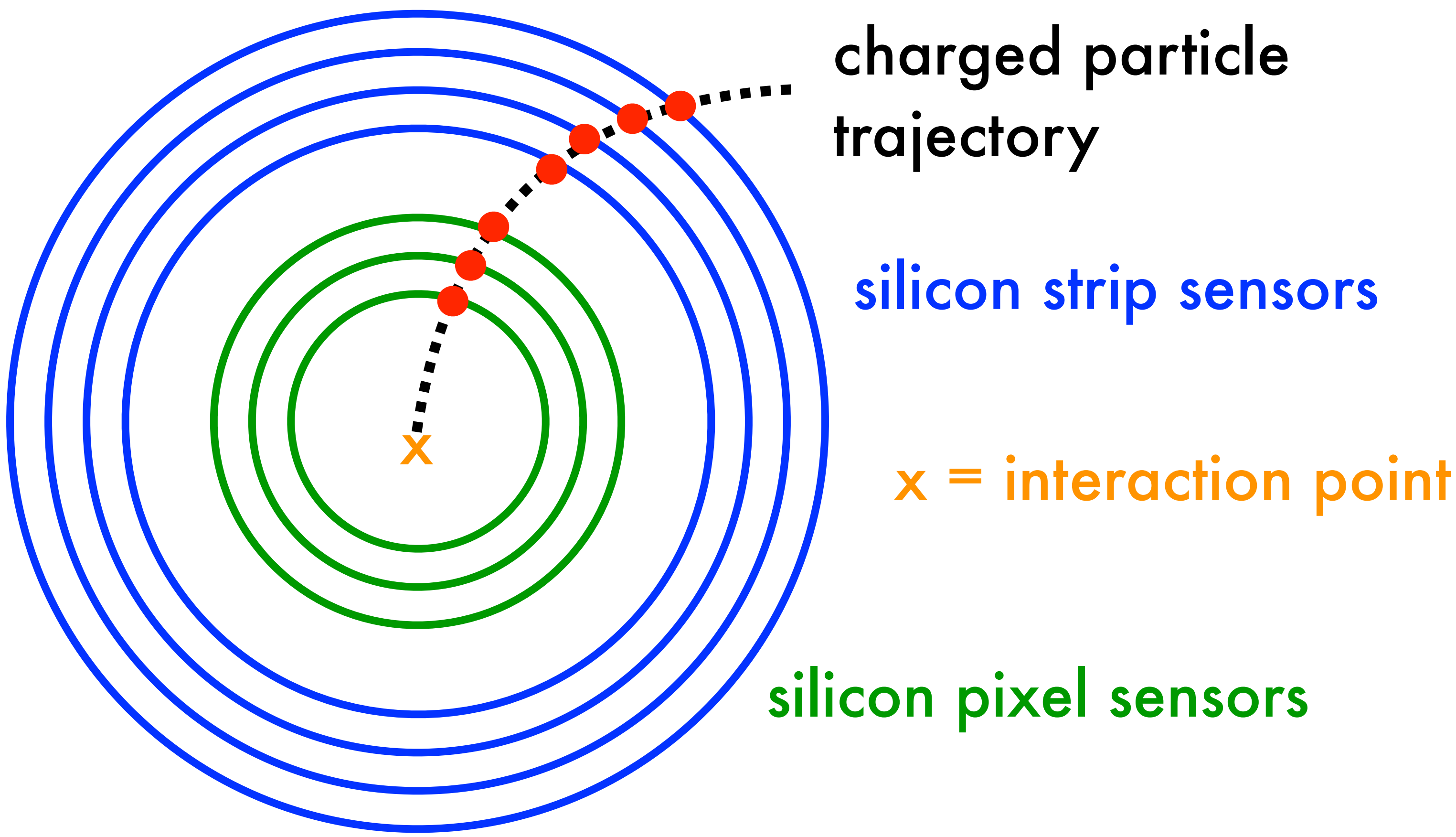


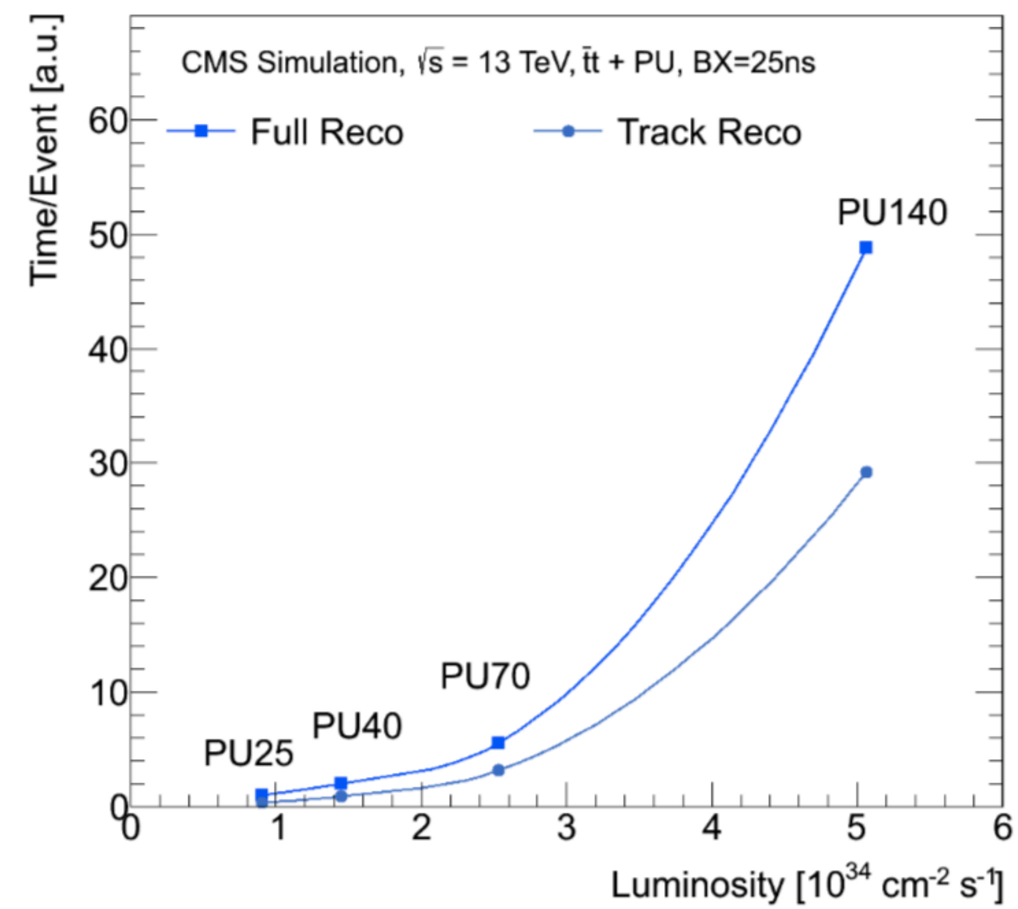


INTEL AND MICROSOFT ENABLE AI INFERENCE AT THE EDGE WITH INTEL MOVIDIUS VISION PROCESSING UNITS ON WINDOWS ML

Today during Windows Developer Day, Microsoft announced Windows* ML, which enables developers to perform machine learning tasks in the Windows OS. Windows ML efficiently uses hardware for any given artificial intelligence (AI) workload and intelligently distributes work across multiple hardware types – now including Intel Vision Processing Units (VPU). The Intel VPU, a purpose-built chip for accelerating AI workloads at the edge, will allow developers to build and deploy the next generation of deep neural network applications on Windows clients.

Example: particle tracking



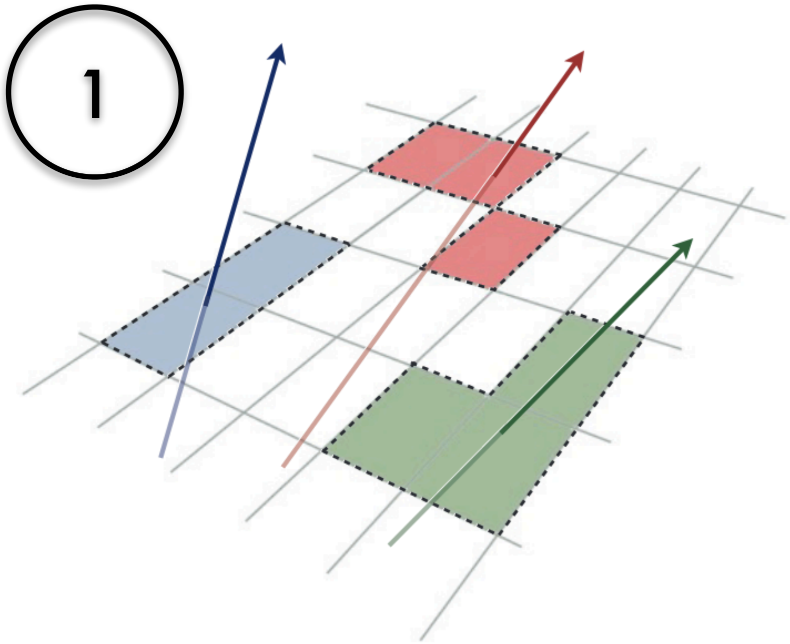


Thousands of particles leaving charge deposition (hits) on O(10) layers of sensors
Curved trajectory due to magnetic field

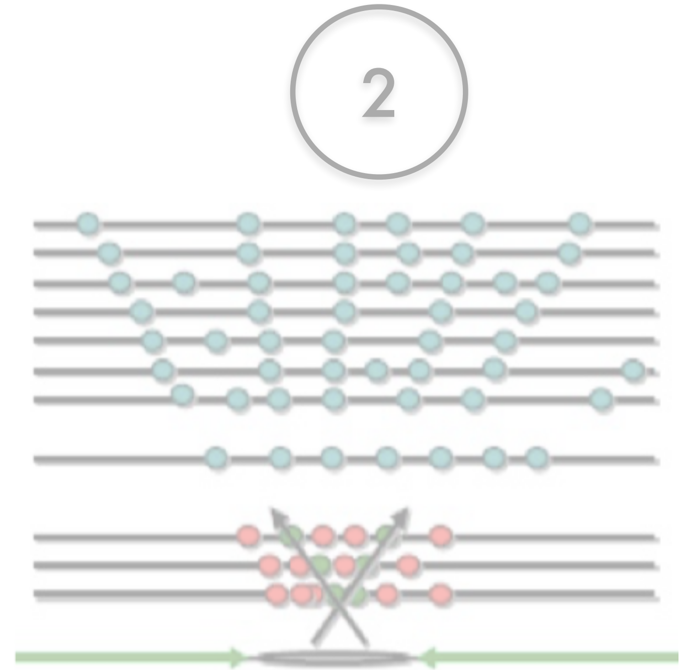
Reconstructing the particle trajectory is the most computing expensive part of physics event reconstruction → scale quadratically or worse with detector occupancy

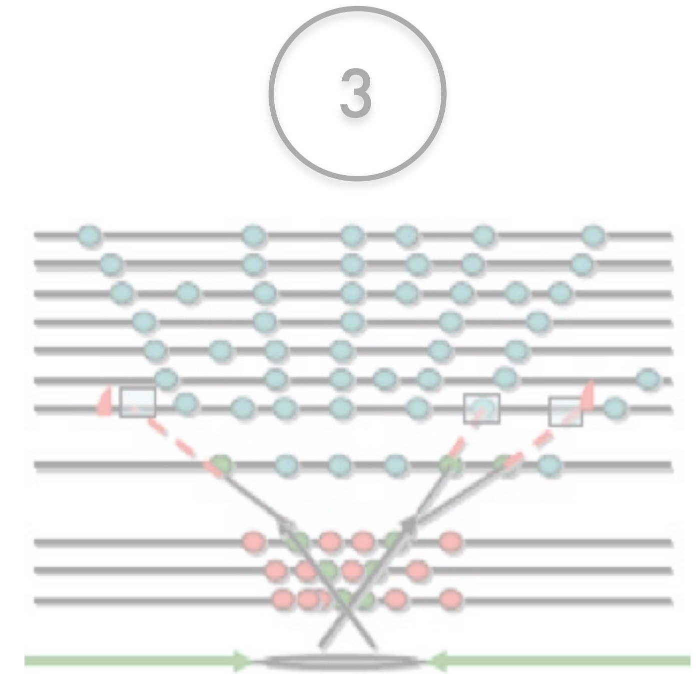
Optimizations (to fit in computational budgets) mostly saturated!

Traditional tracking algorithm

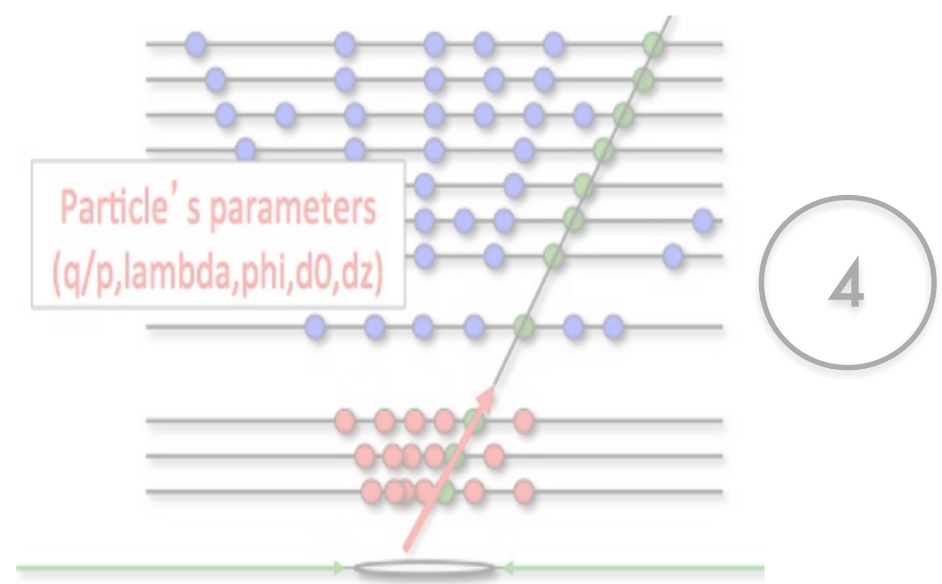


Hit preparation

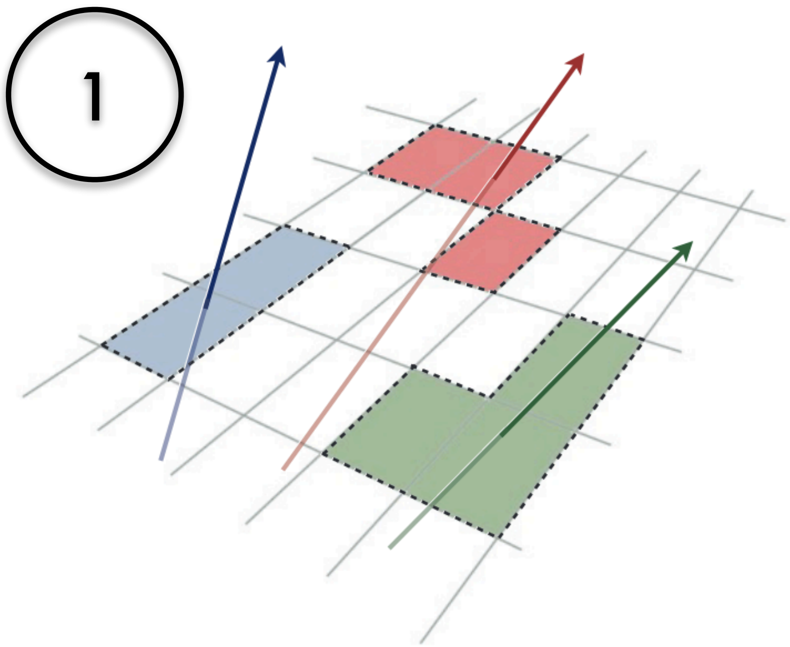




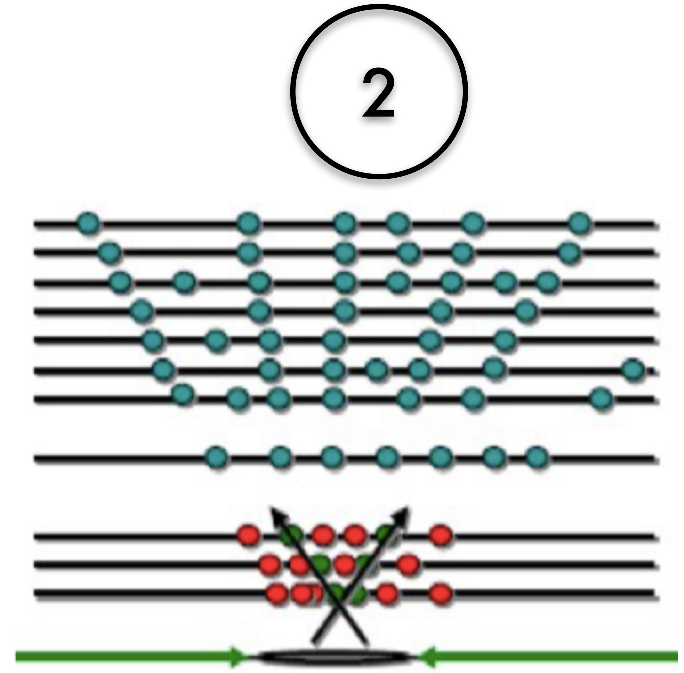
1. Clustering of single energy deposits into a particle "hit"



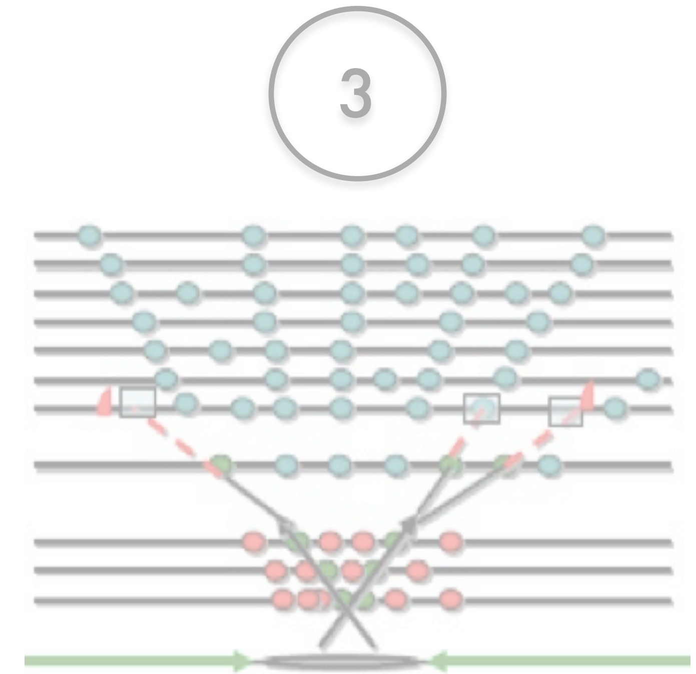
Traditional tracking algorithm



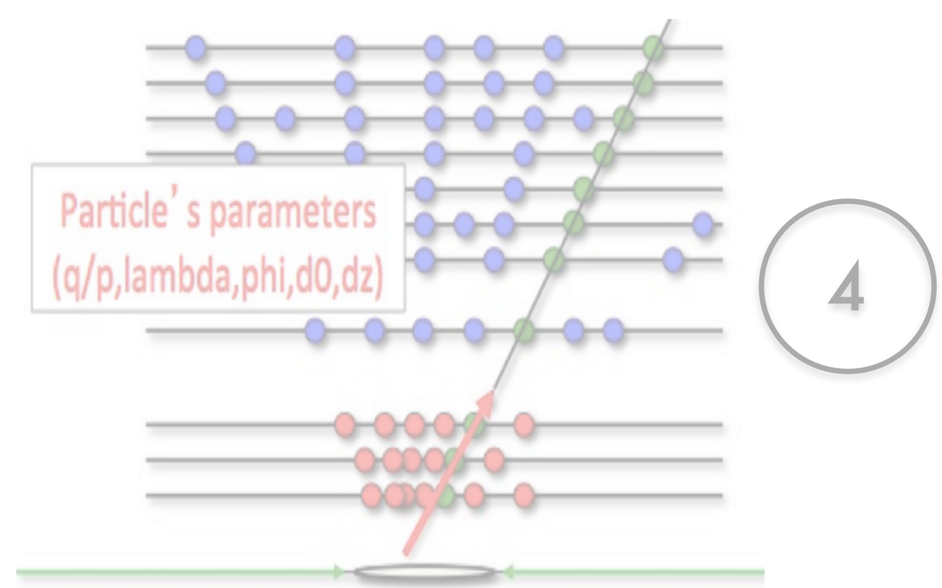
Hit preparation



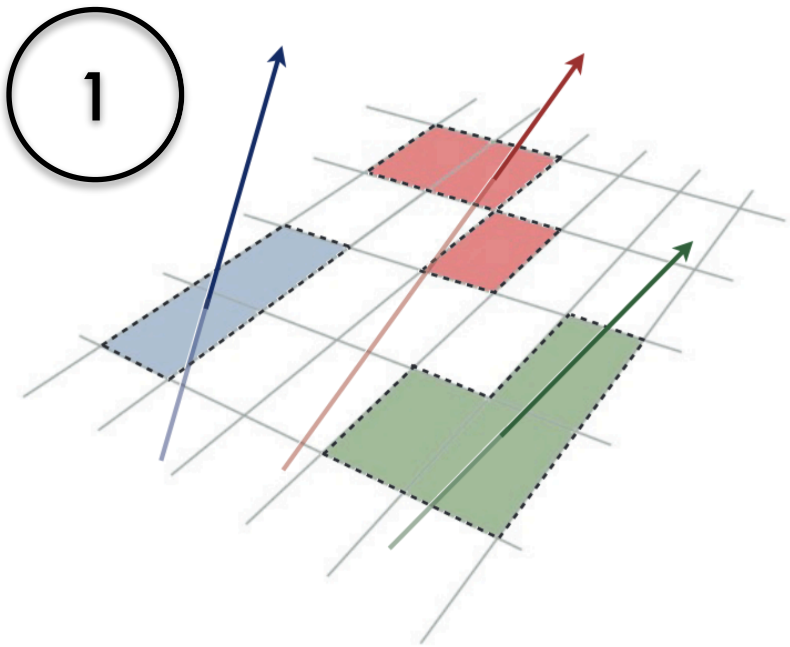
Track seeding



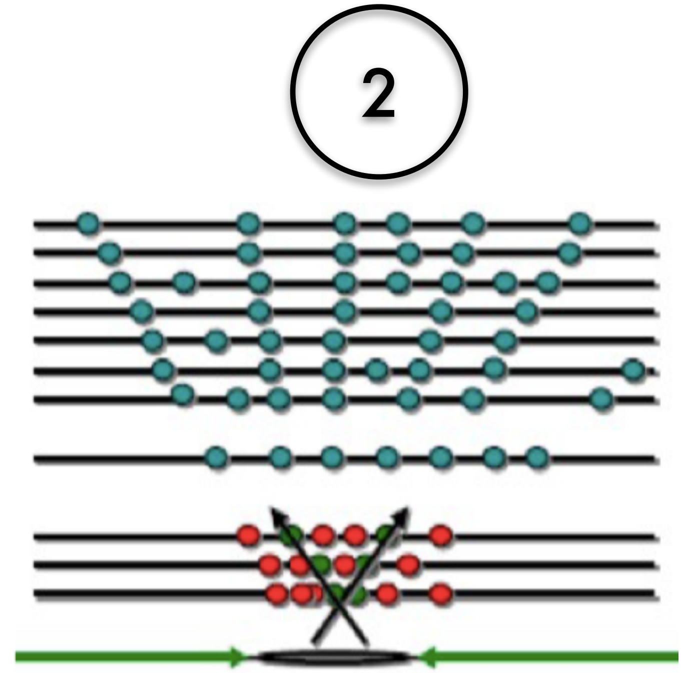
2. Seeding: compatible sets of three hits in the inner pixel layers used as track seed
    - fixes the combinatorics
      → fixes CPU usage



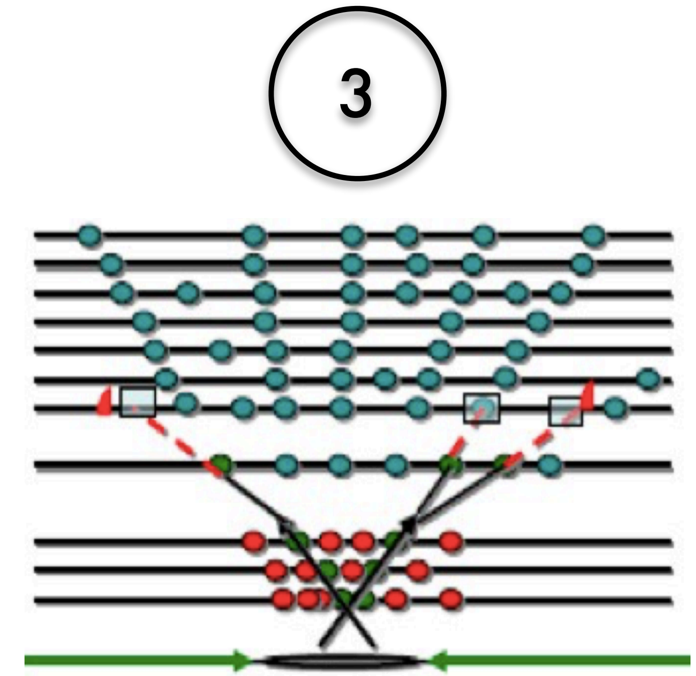
Traditional tracking algorithm



Hit preparation

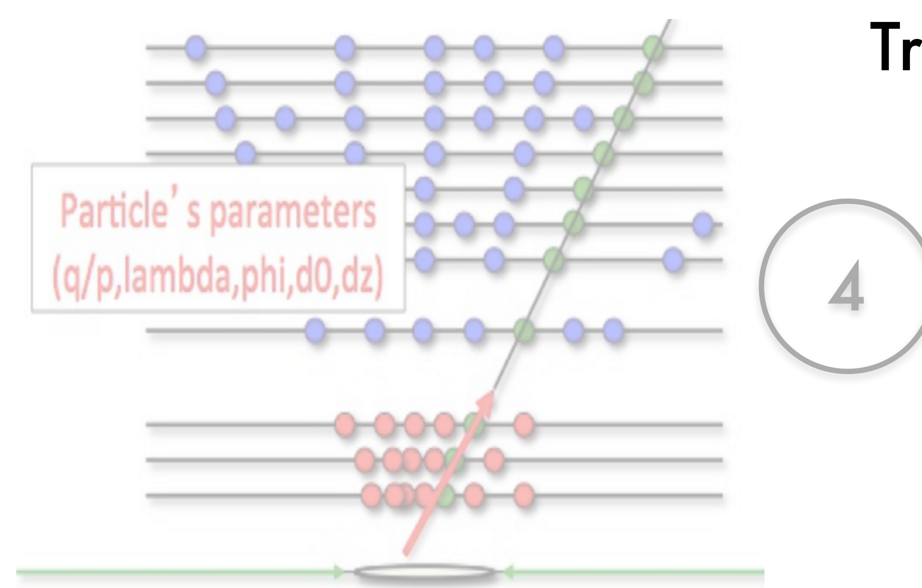


Track seeding



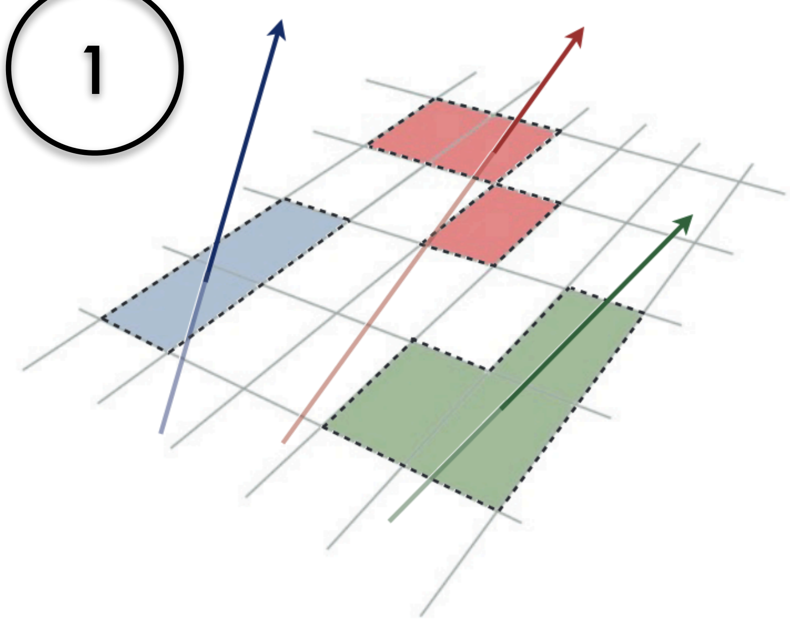
Track building

3. Build the track from the chosen seeds using Kalman Filter to predict hit position in the next layer



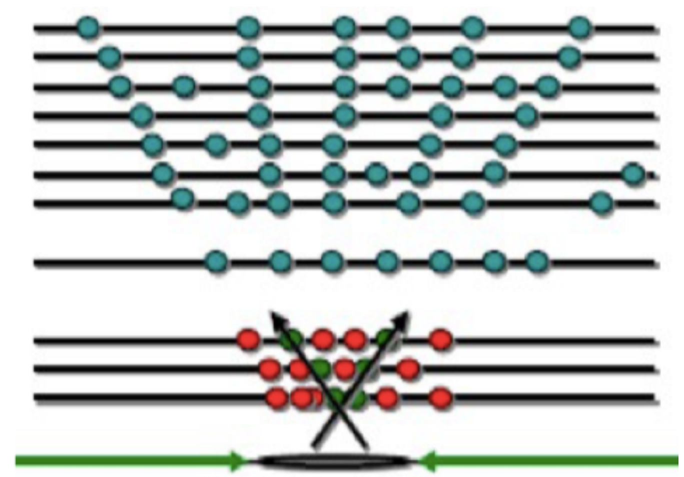
Traditional tracking algorithm

1



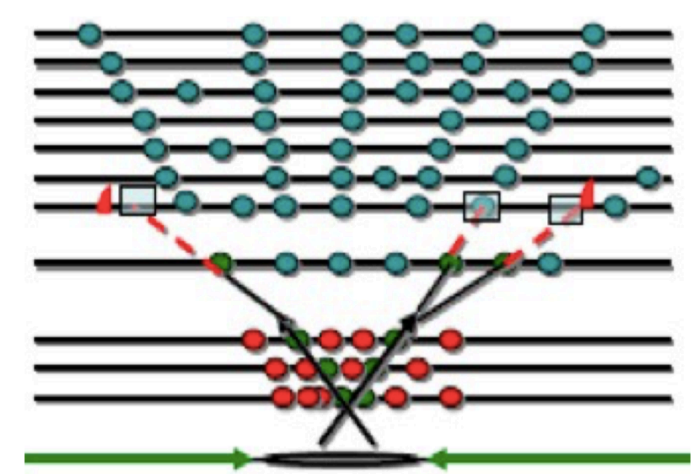
Hit preparation

2



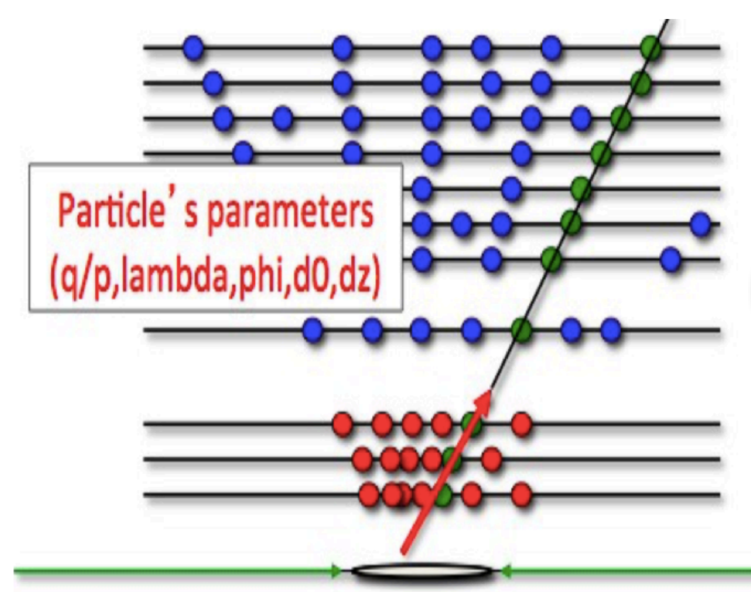
Track seeding

3



Track building

4. Final full resolution fit of the candidate trajectories to extract particle properties



4

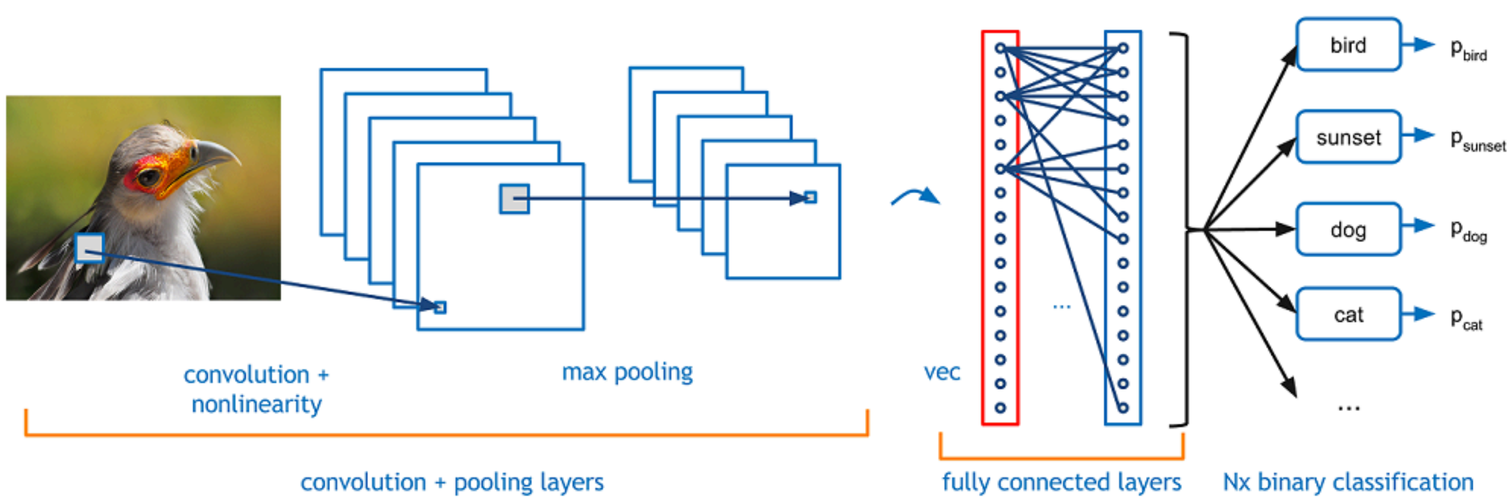
Track fitting

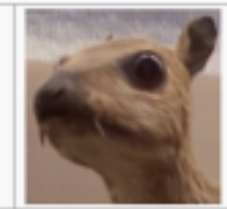
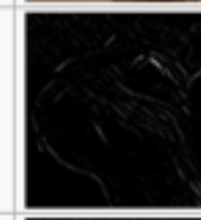
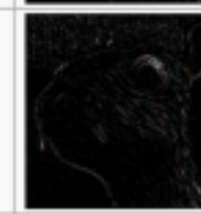
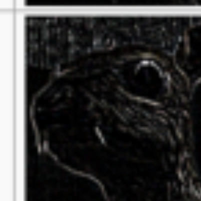
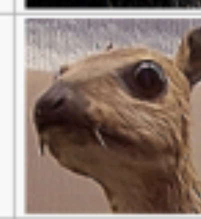
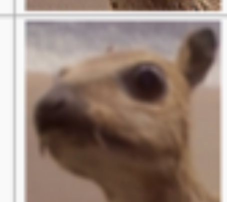
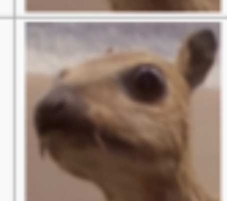
Speed up tracking with computer vision

- Computer vision methods automatically extract, analyse and understand useful information from very low level inputs such as pixels in an image

- Make use of Convolutional Neural Networks

  identify low level features (edges,curves,..) through filters and then build them up to abstract concepts

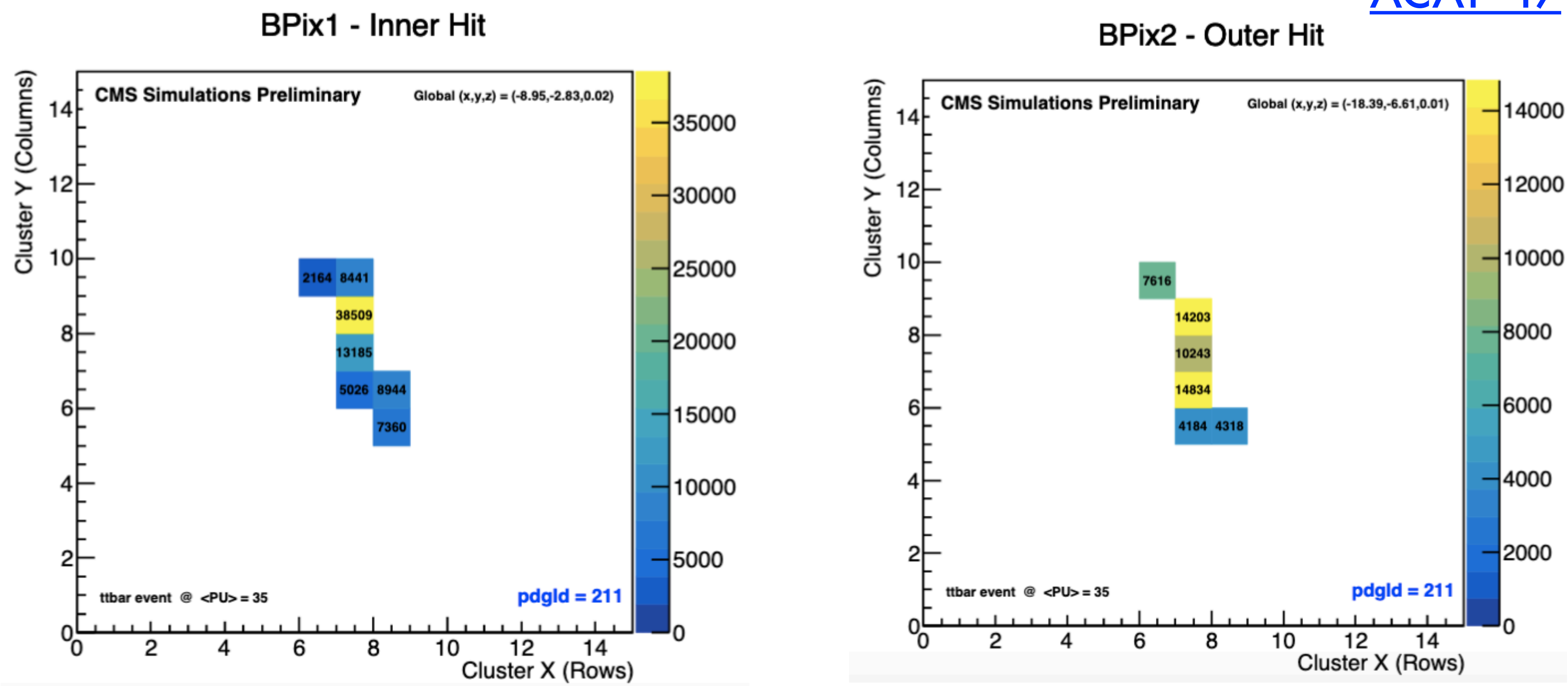


Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Speed up tracking with computer vision

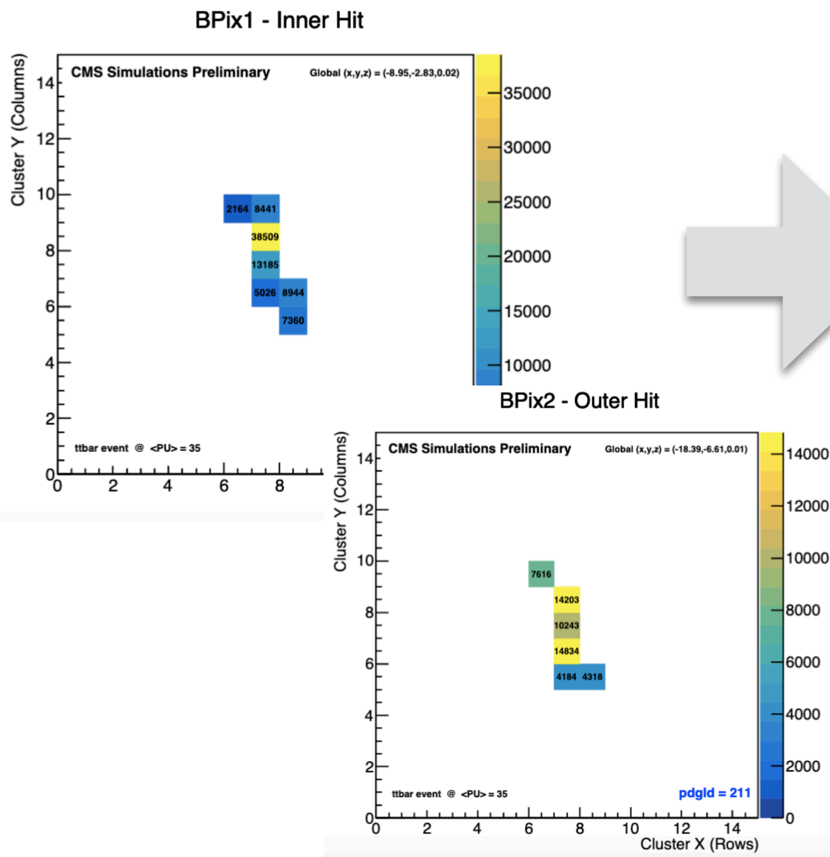
- Track building demands a lot of computational resources, so one should choose carefully which seeds to use → remove fake seeds due to combinatorial background
- Reduce this effect by taking into account the shape of the hit pixel cluster to check the compatibility between two hits

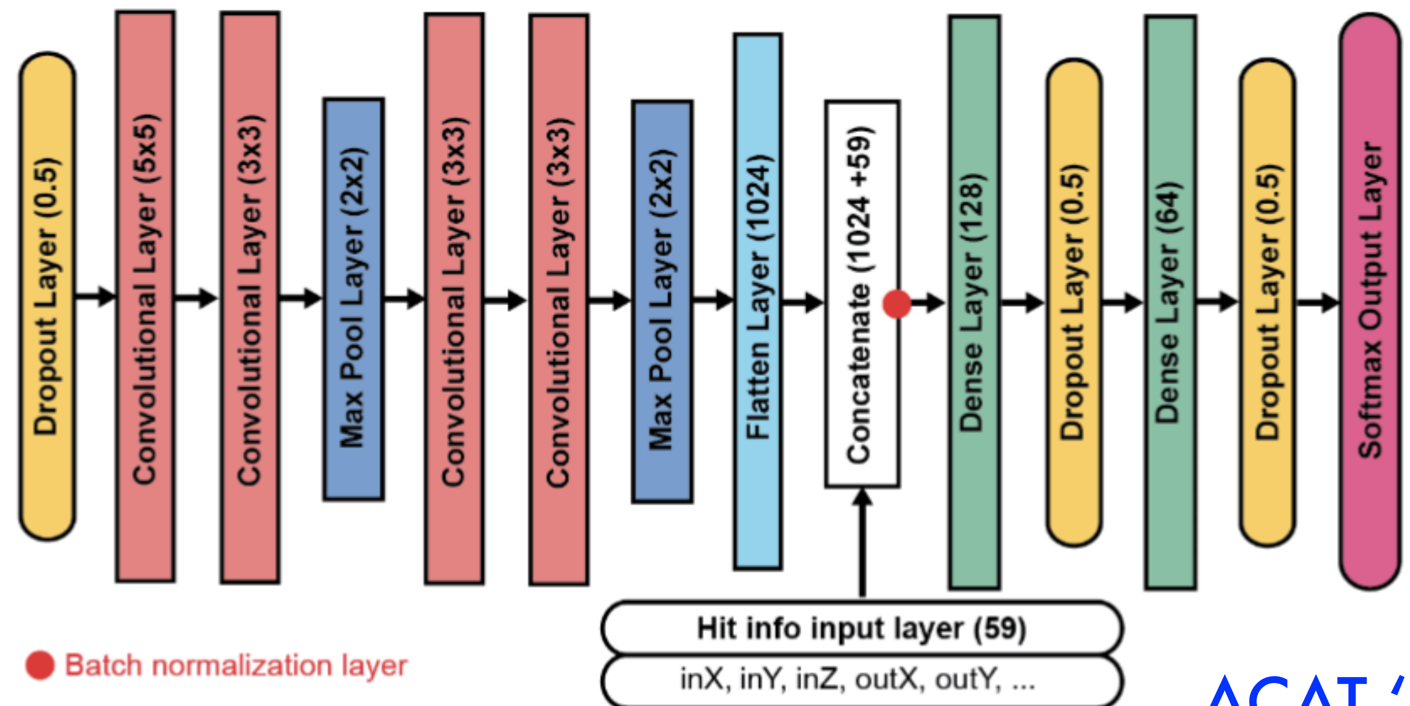
ACAT '17



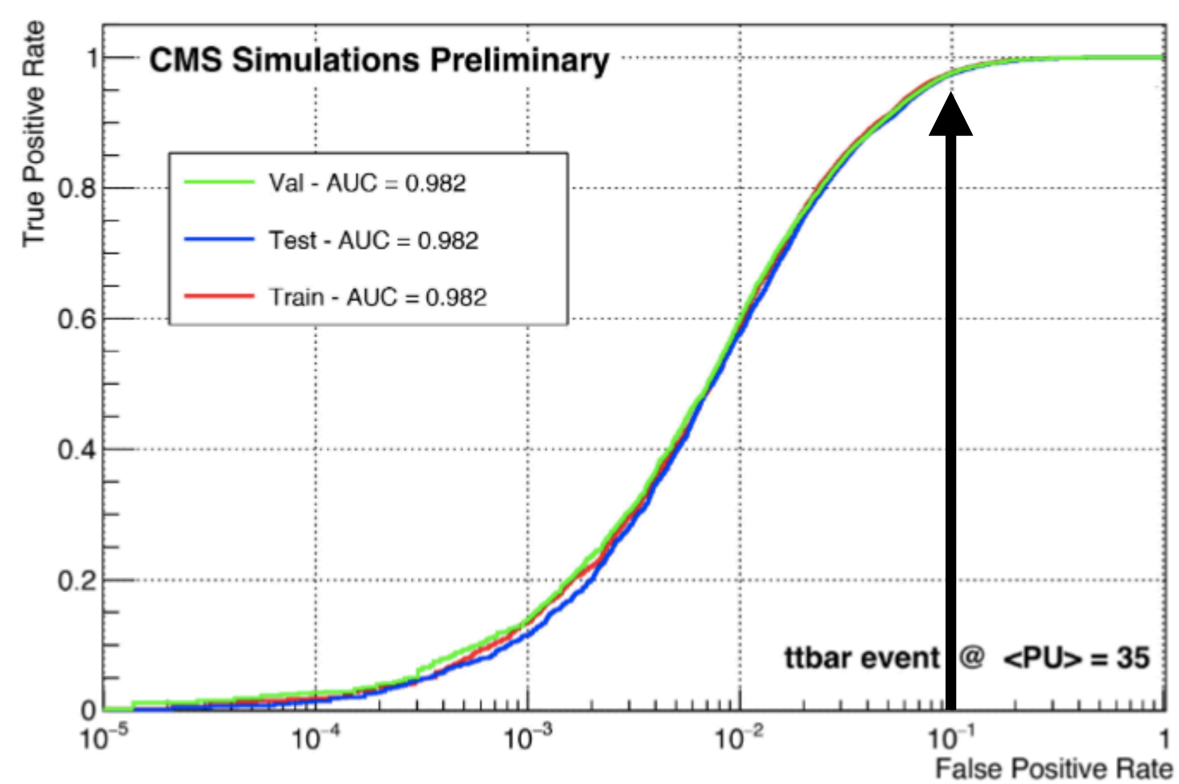
color code = how much energy the particle leaves in each pixel

Speed up tracking with computer vision





[ACAT '17]

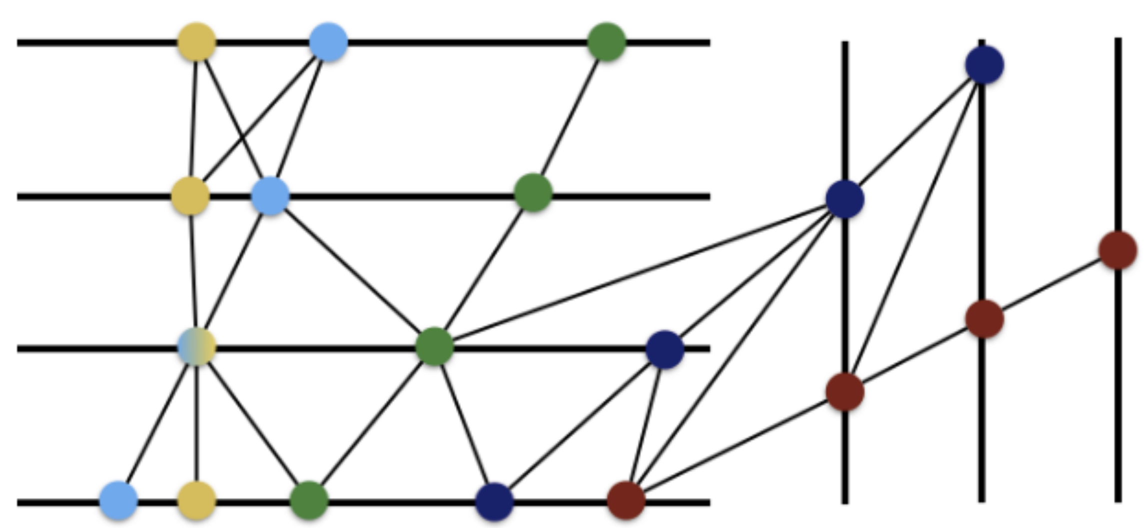


Reduce fake rate by one order of magnitude with only few % loss in efficiency!

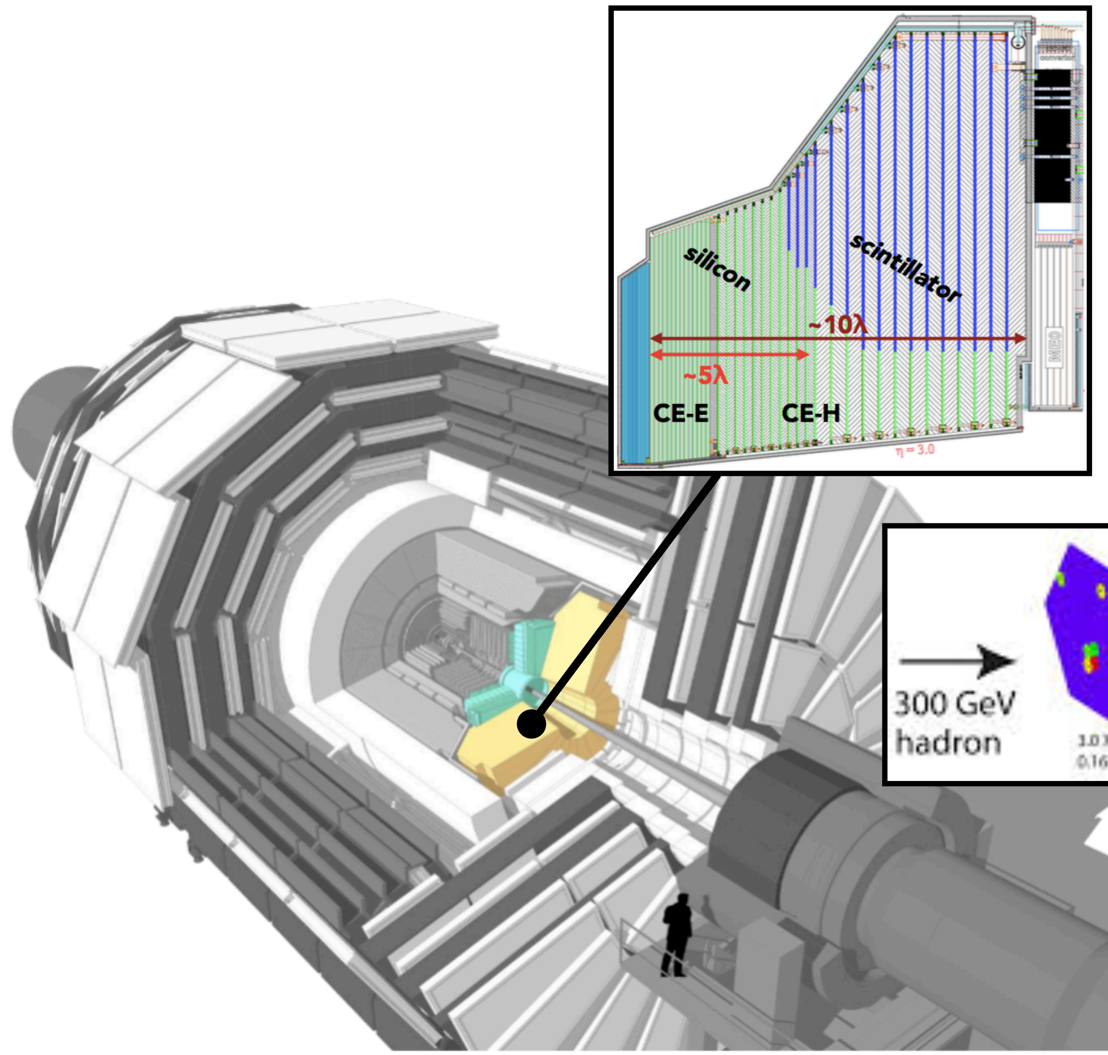
Deep learning for tracking

- Other work on particle tracking with deep learning: https://heptrkx.github.io/
  - computer vision approach
  - recurrent neural networks
  - graph neural networks

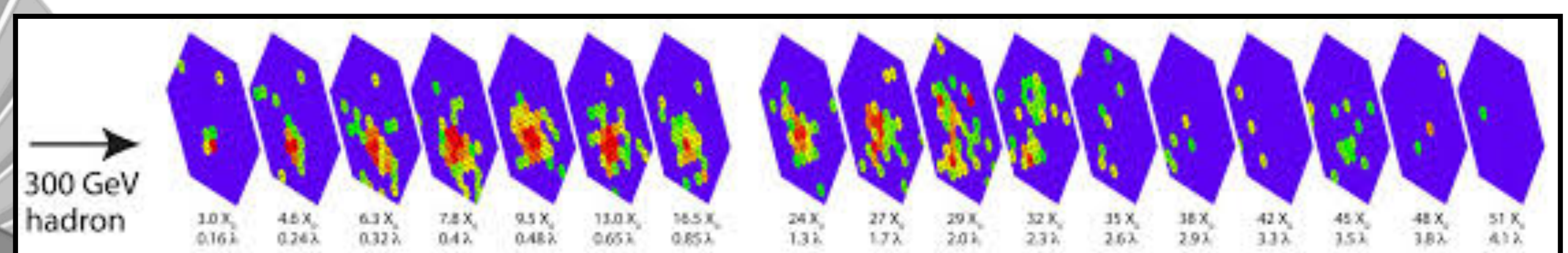
operate on the spacepoint representation of track measurements ("hits")

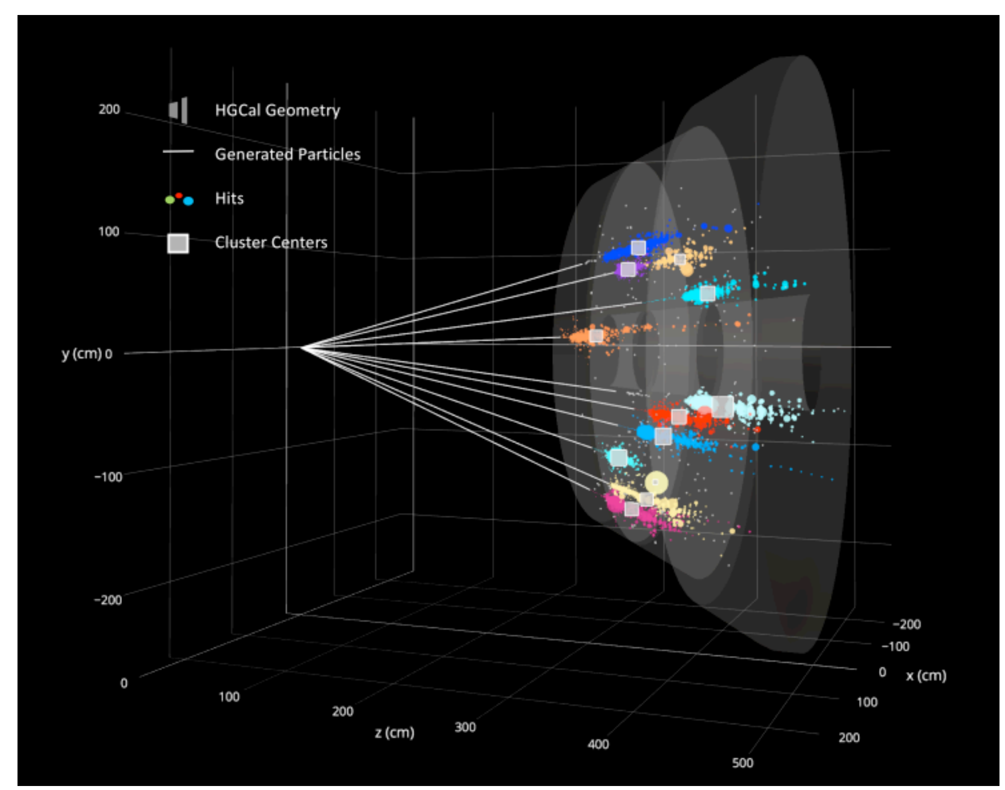


Calorimetry with computer vision



With highly granular 3D arrays of hexagonal pixels will improve ability to identify and characterize particles.

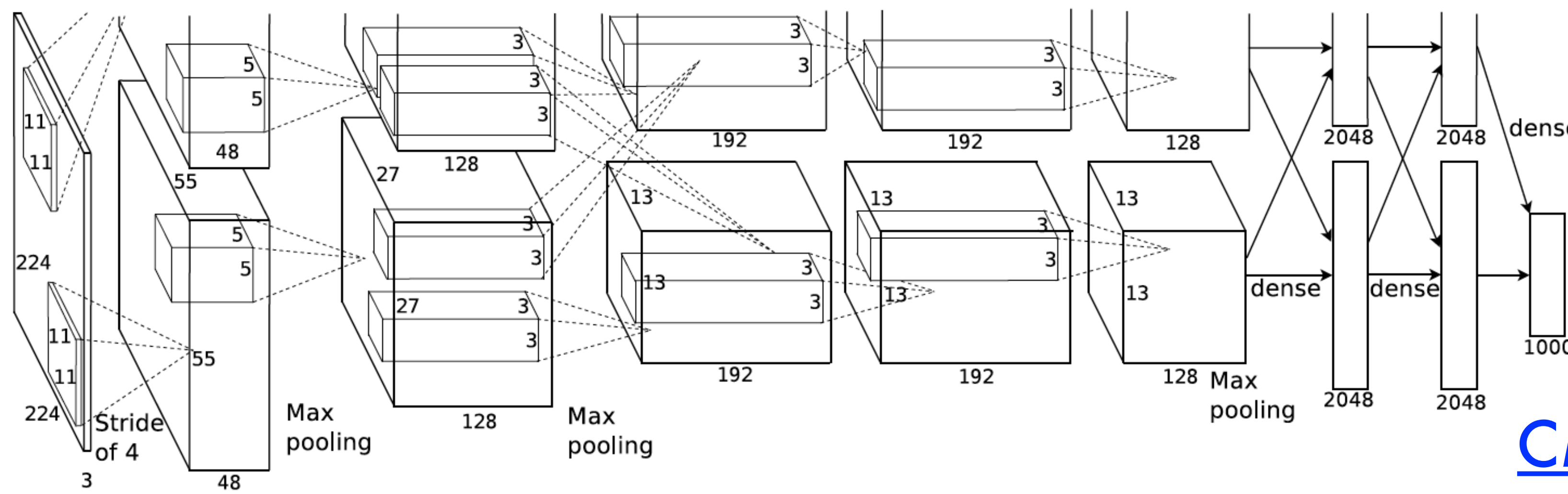




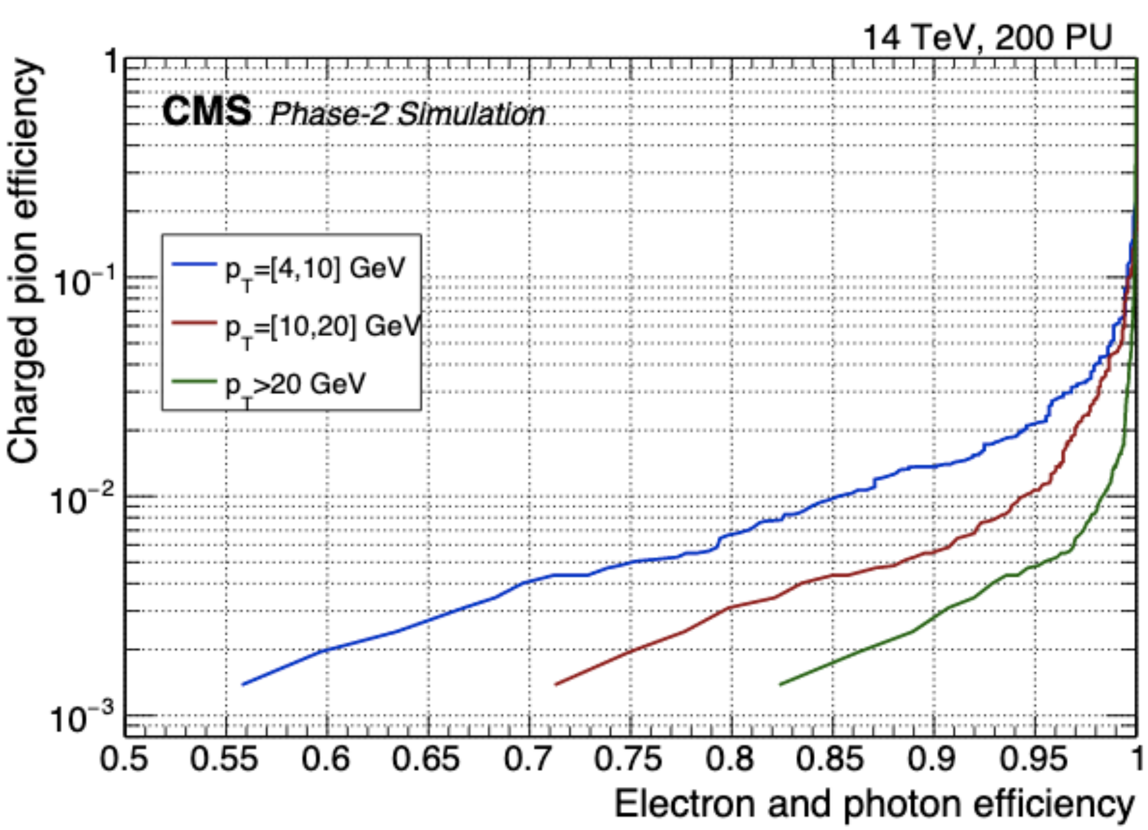
Ideal geometry and resolution to apply computer vision with 3D convolutional NN to speed up calorimetry and improve performances.

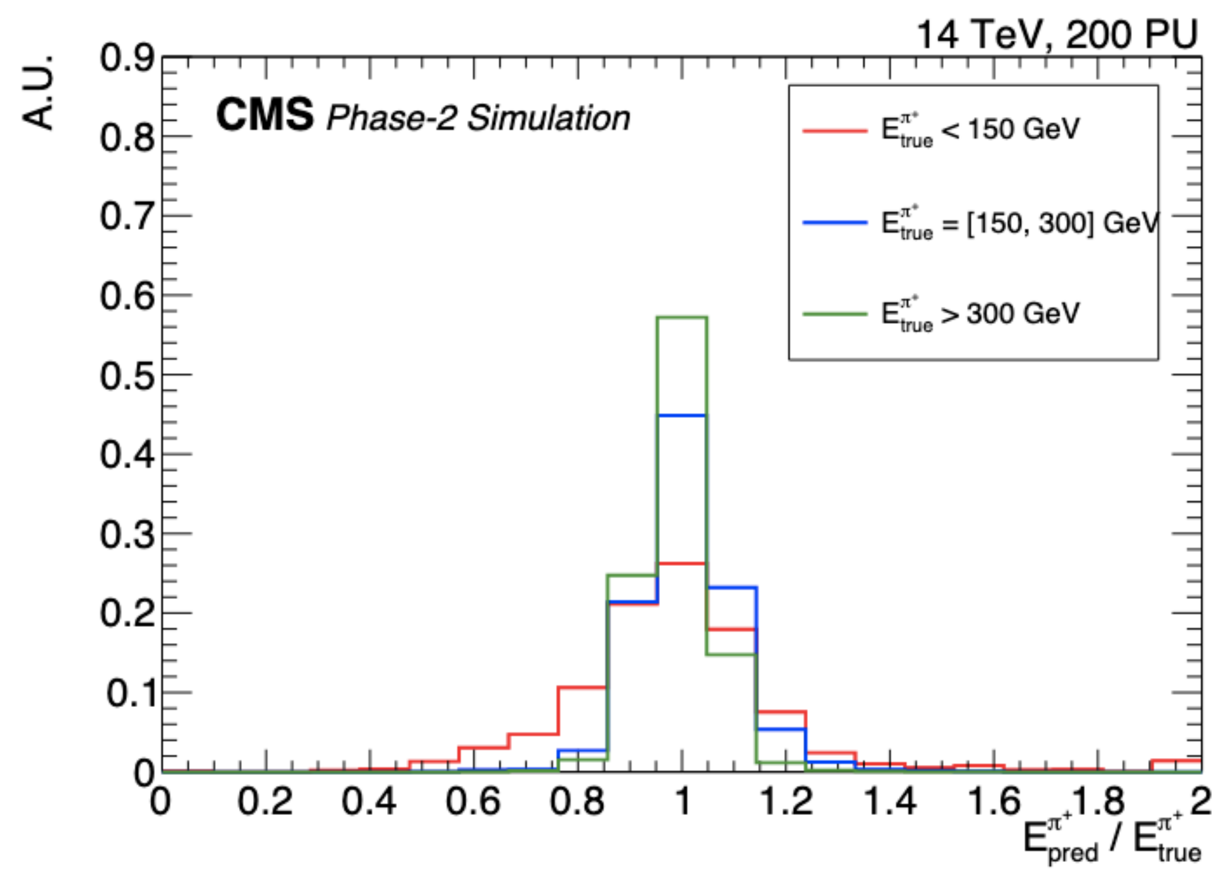
Calorimetry with computer vision

Feed raw 3D pixelated image of the calorimeter to Conv 3D NN architecture to achieve state-of-the-art performance in terms of particle identification and energy measurement



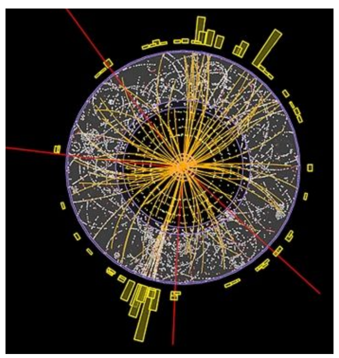
[CMS HGCAL TDR]



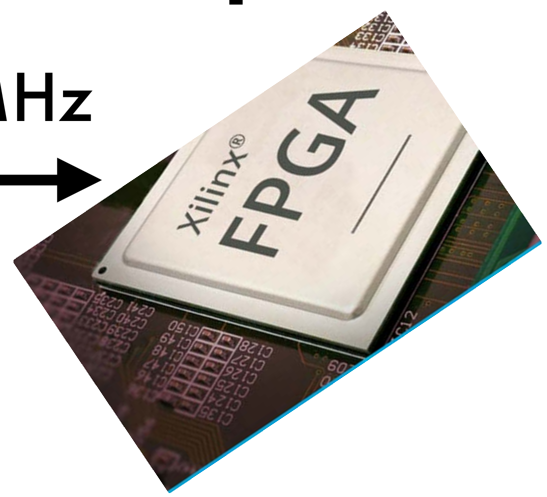


How fast can we do a NN inference?

1 ns — 1 μs — 100 ms — 1 s

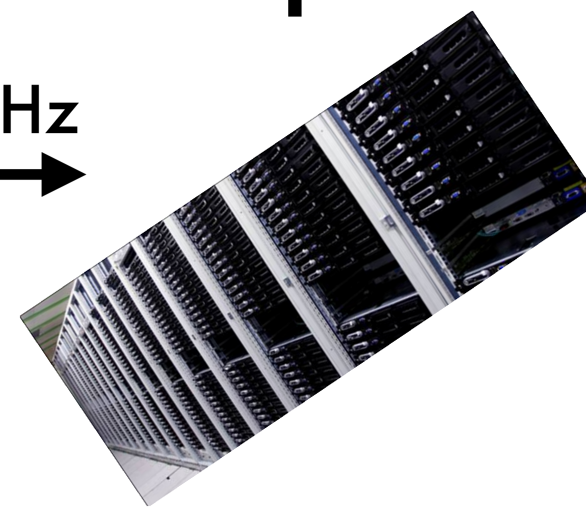


40 MHz →



Level 1 trigger

100 KHz →



High-Level Trigger

1 KHz →

1 MB/event

Offline reconstruction @ Tier 0

Absorbs 100s TB/s

Trigger decision to be made in O(μs)

Latencies require all-FPGA design

Computing farm for detailed analysis of the full event

Latency O(100 ms)

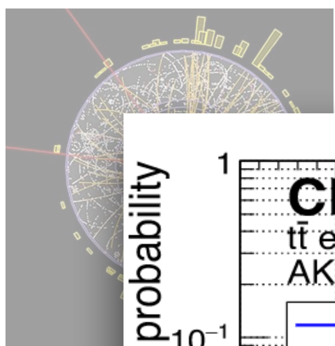
How fast can we do a NN inference?

1 ns

1 μs

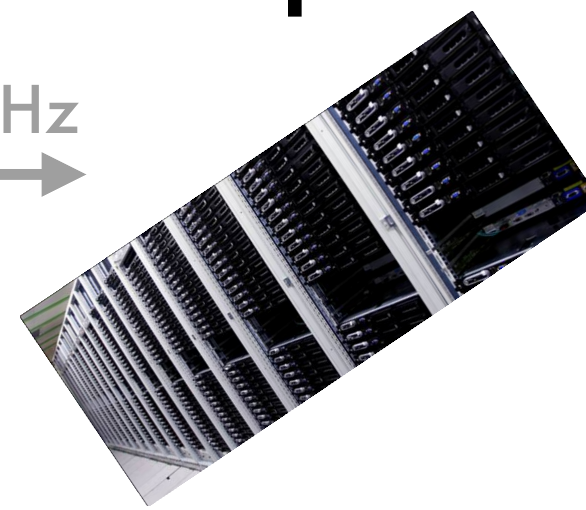
100 ms

1 s



40 MHz

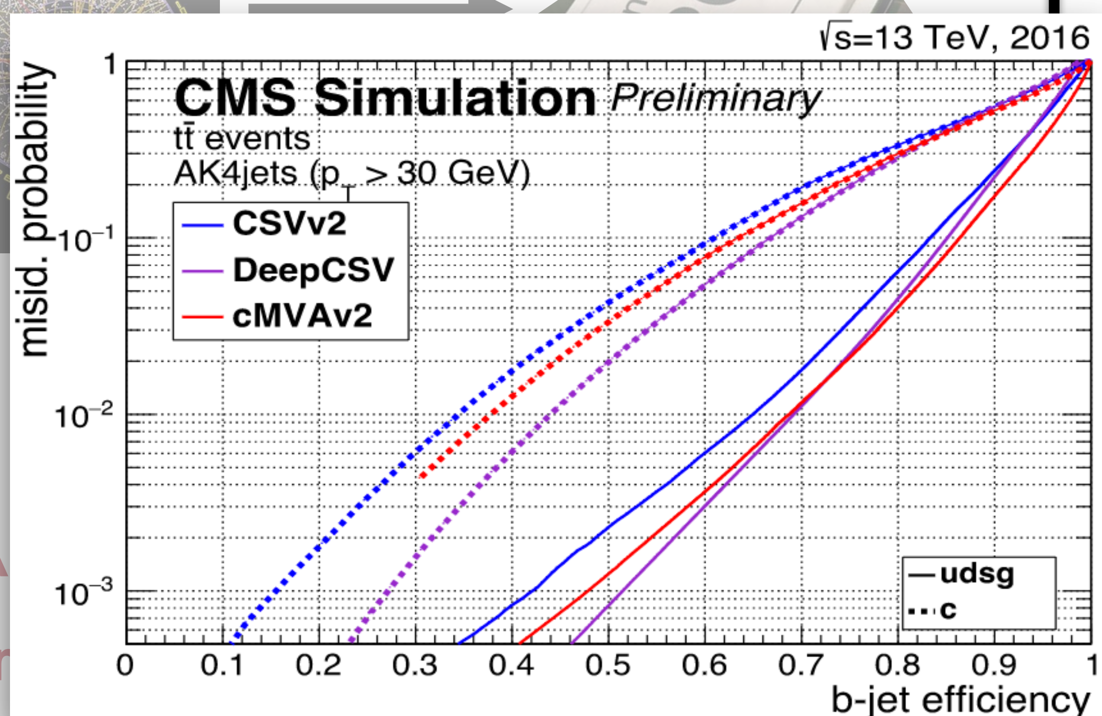
100 KHz



1 KHz

1 MB/event

Offline reconstruction @ Tier 0



High-Level Trigger

A

Tr

Latencies require all-FPGA design

Computing farm for detailed analysis of the full event

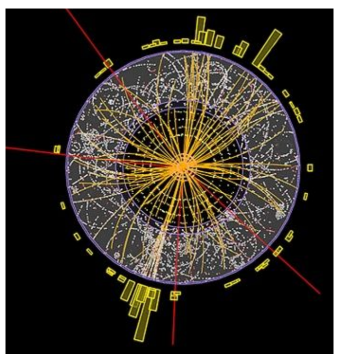
Latency O(100 ms)

Deep neural network based on high-level features for b-quark jets identification (offline & HLT)

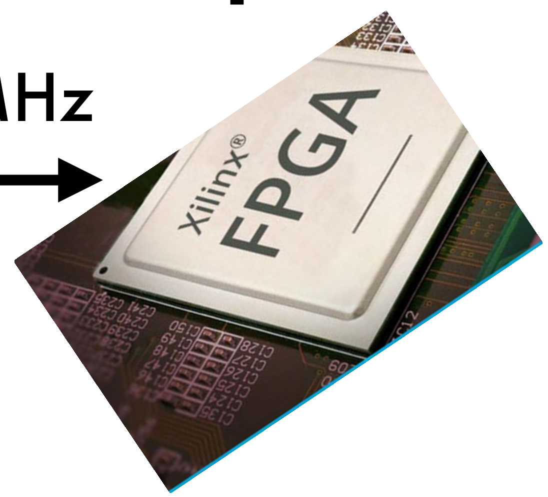
We are already applying Deep Learning here!

Ultra low-latency DL for L1 trigger

1 ns — 1 μs — 100 ms — 1 s

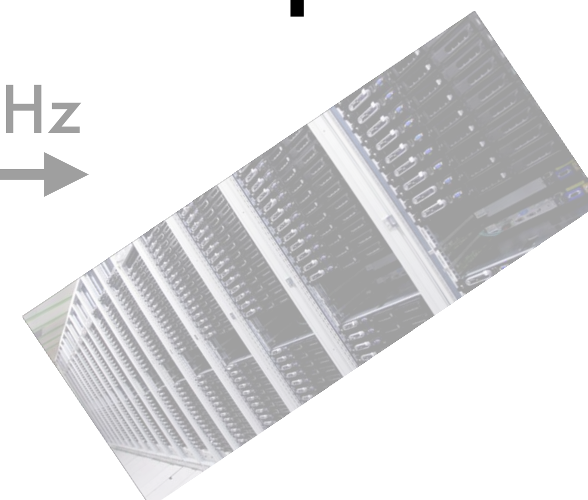


40 MHz



Level 1 trigger

100 KHz



1 KHz

1 MB/event

Offline reconstruction @ Tier 0

High-Level Trigger

Absorbs 100s TB/s

Trigger decision to be made in O(μs)

Latencies require all-FPGA design

Computing farm for detailed analysis of the full event

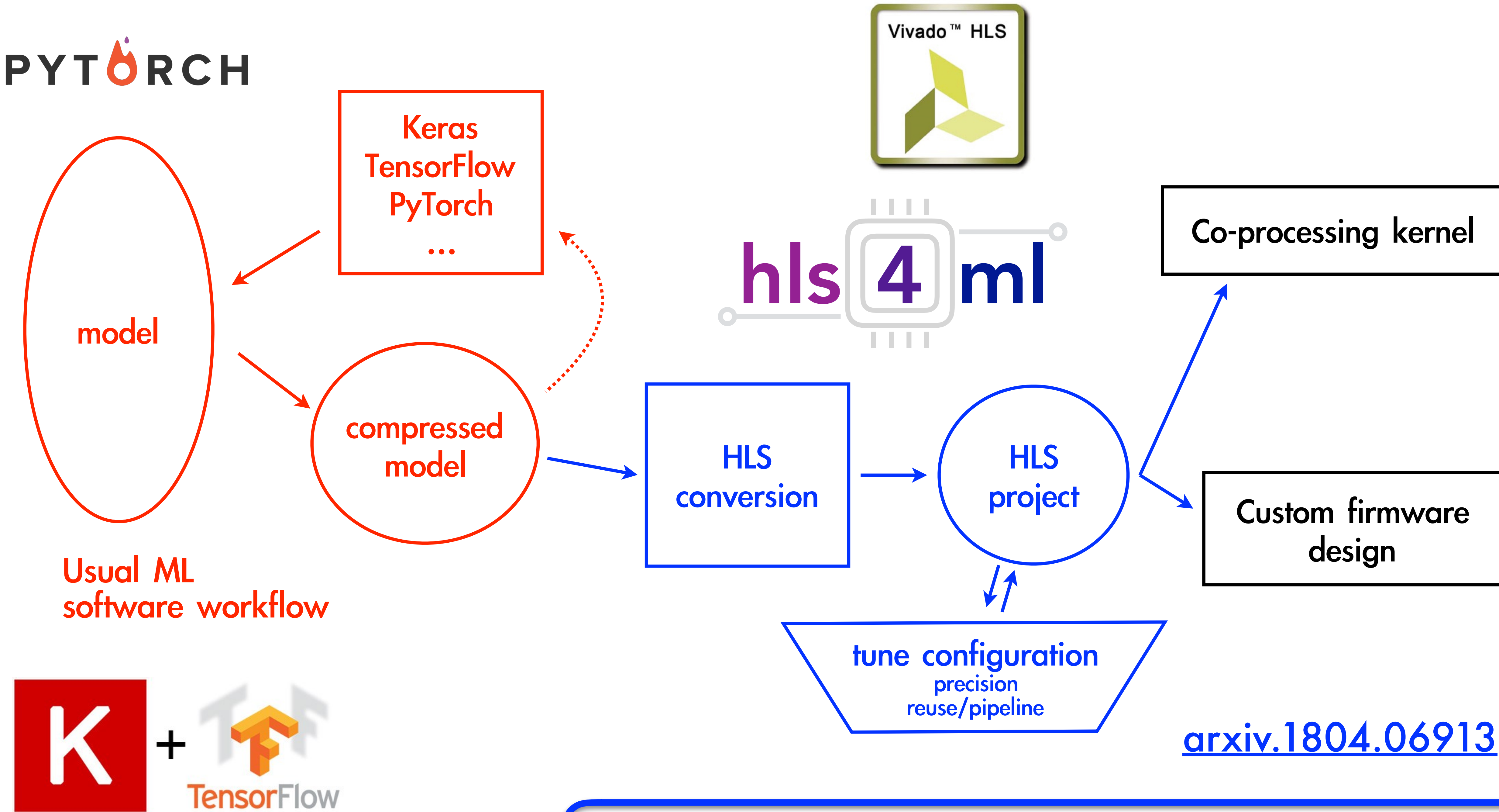
Latency O(100 ms)

Can we do real-time AI in O(μs) on one FPGA?

We are already applying Deep Learning here!

Bring DL to FPGA for L1 trigger with

high level synthesis for machine learning

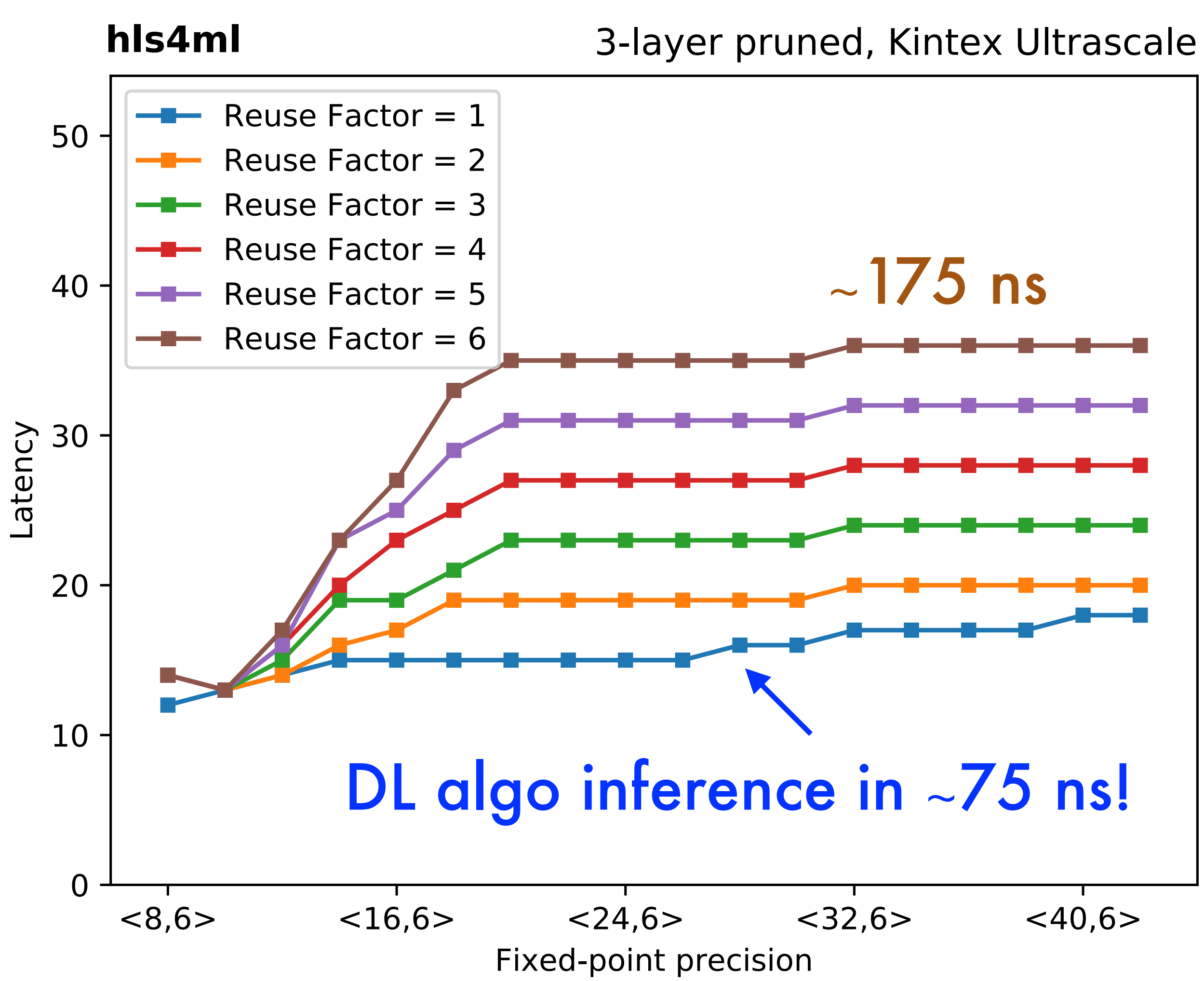


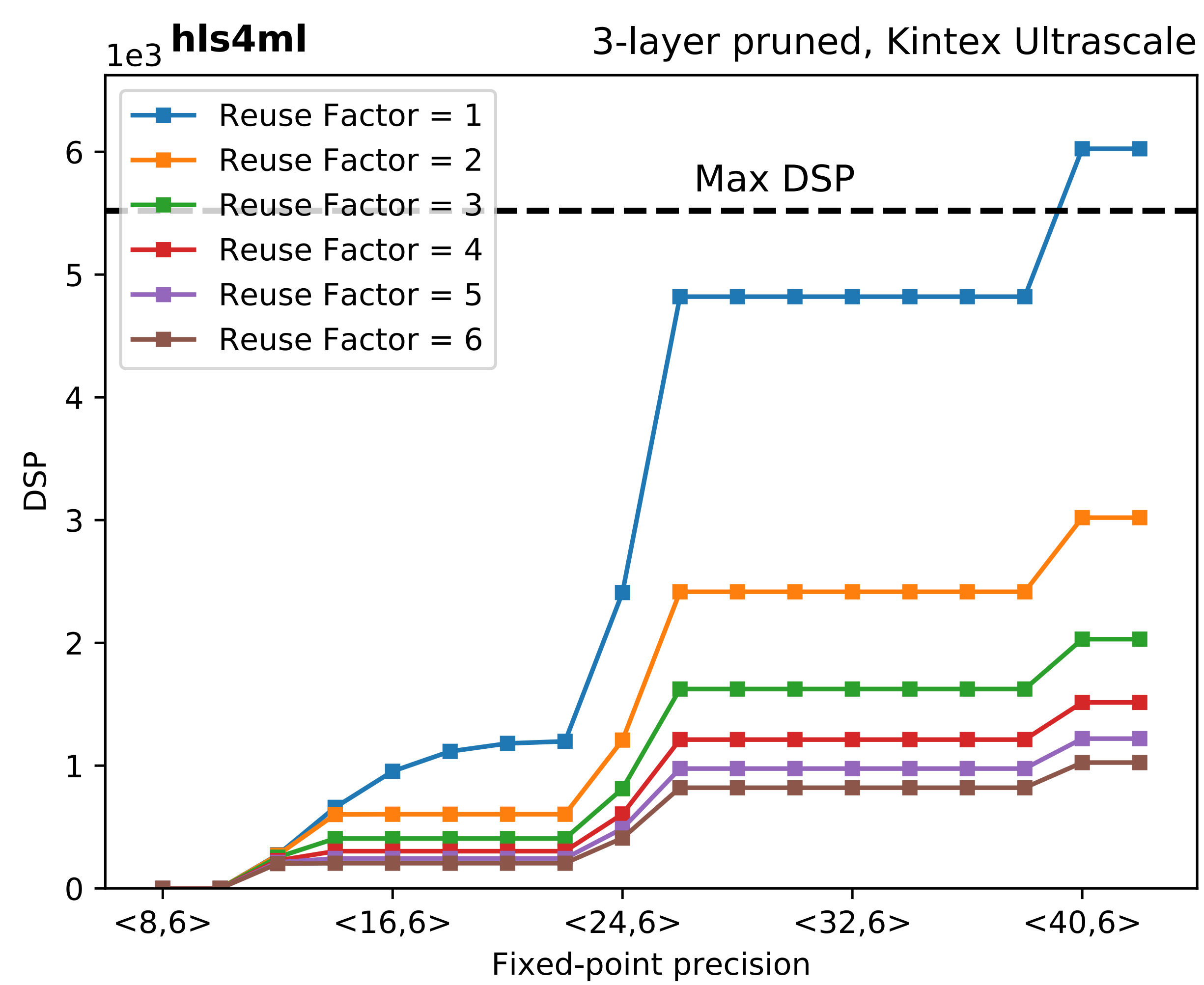
https://hls-fpga-machine-learning.github.io/hls4ml/

Bring DL to FPGA for L1 trigger

Exploiting high FPGA hardware flexibility we can fit DL solutions @ L1:

- *highly-parallel algorithm implementation*
- *large bandwidth*
- *reduced calculation precision without loss in performance* [arxiv.1804.06913](https://arxiv.org/abs/1804.06913)



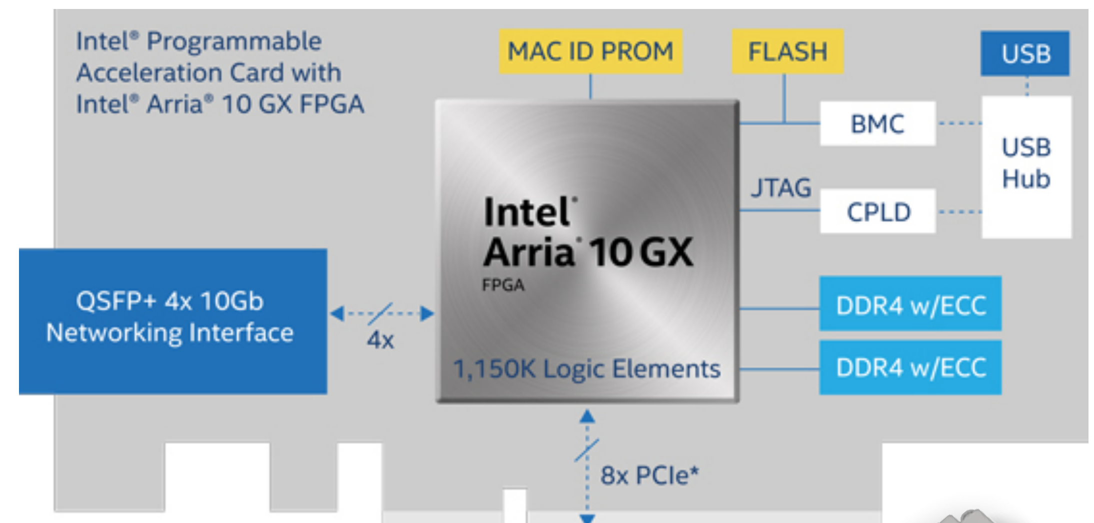


Heterogeneous computing

Offload a CPU from the computational heavy parts to a FPGA "accelerator"

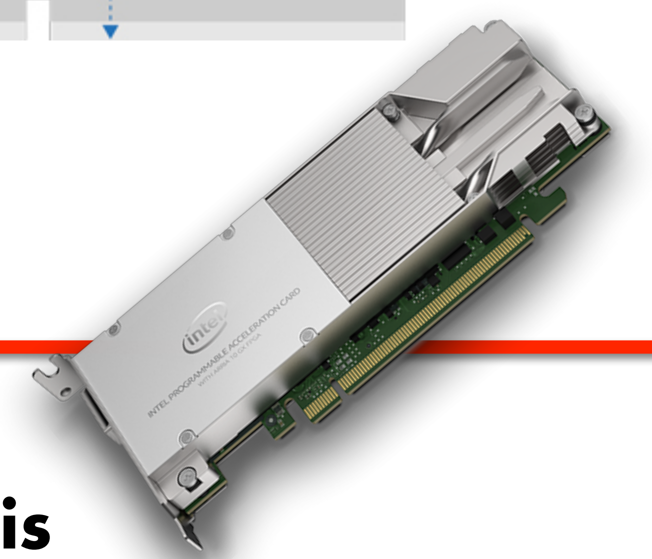
Increased computational speed of 10x-100x
Reduced system size of 10x
Reduced power consumption of 10x-100x

Intel® Programmable Acceleration Card with Intel Arria® 10 GX FPGA



Increasing popularity of co-processor systems

CPU+FPGA / CPU+GPU / CPU+TPU / …
Common setup for FPGA connects to CPU through PCI-express

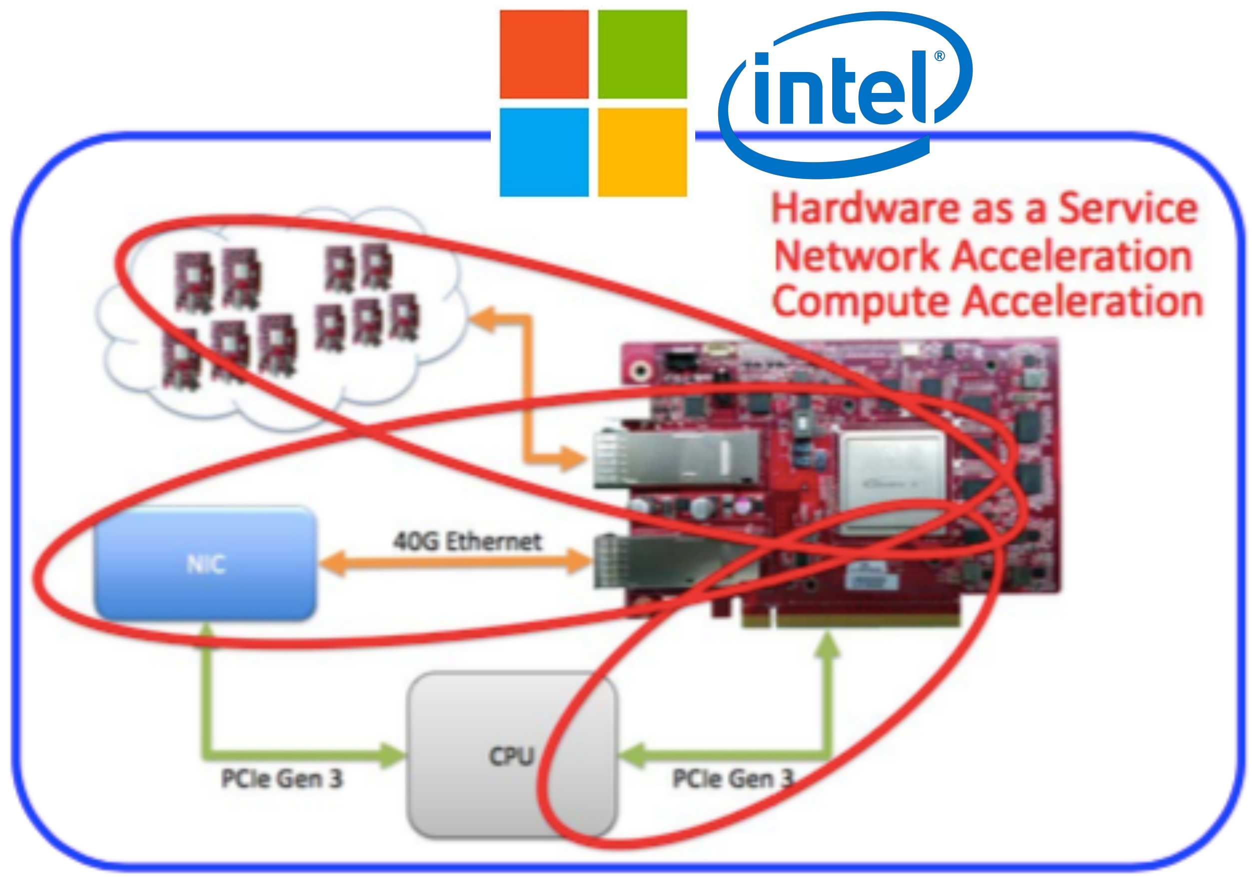


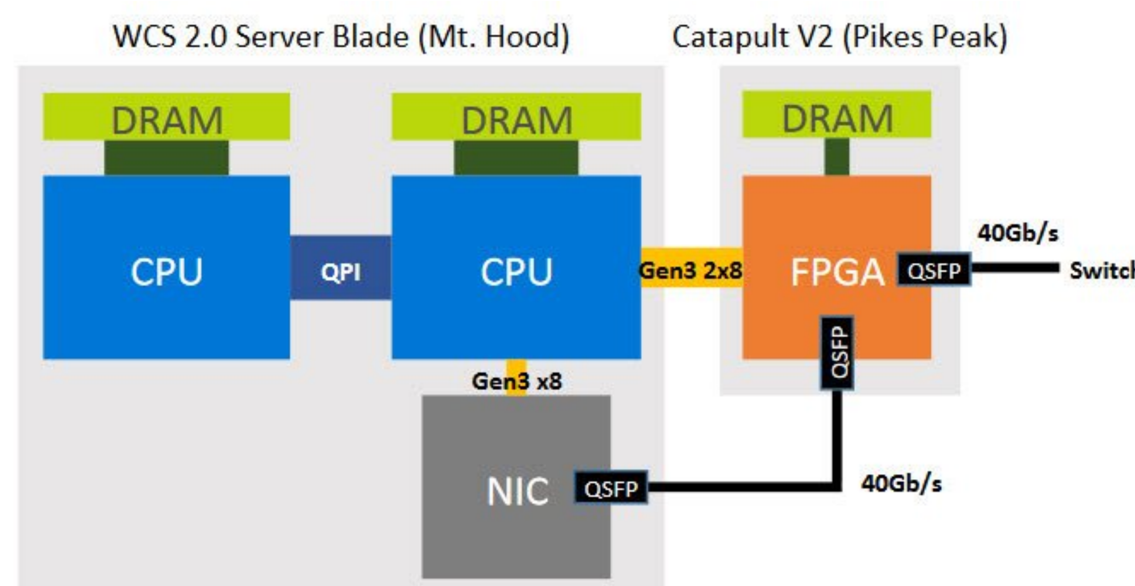
Use case @ LHC to accelerate slow algorithms (ex: tracking) and ML inference for HLT and offline analysis

Ongoing R&D on heterogeneous computing on-site (@CERN) and on commercial clouds (Microsoft Brainwave, Amazon Web Services, Google TPU cloud)

Co-processors as a service with Brainwave

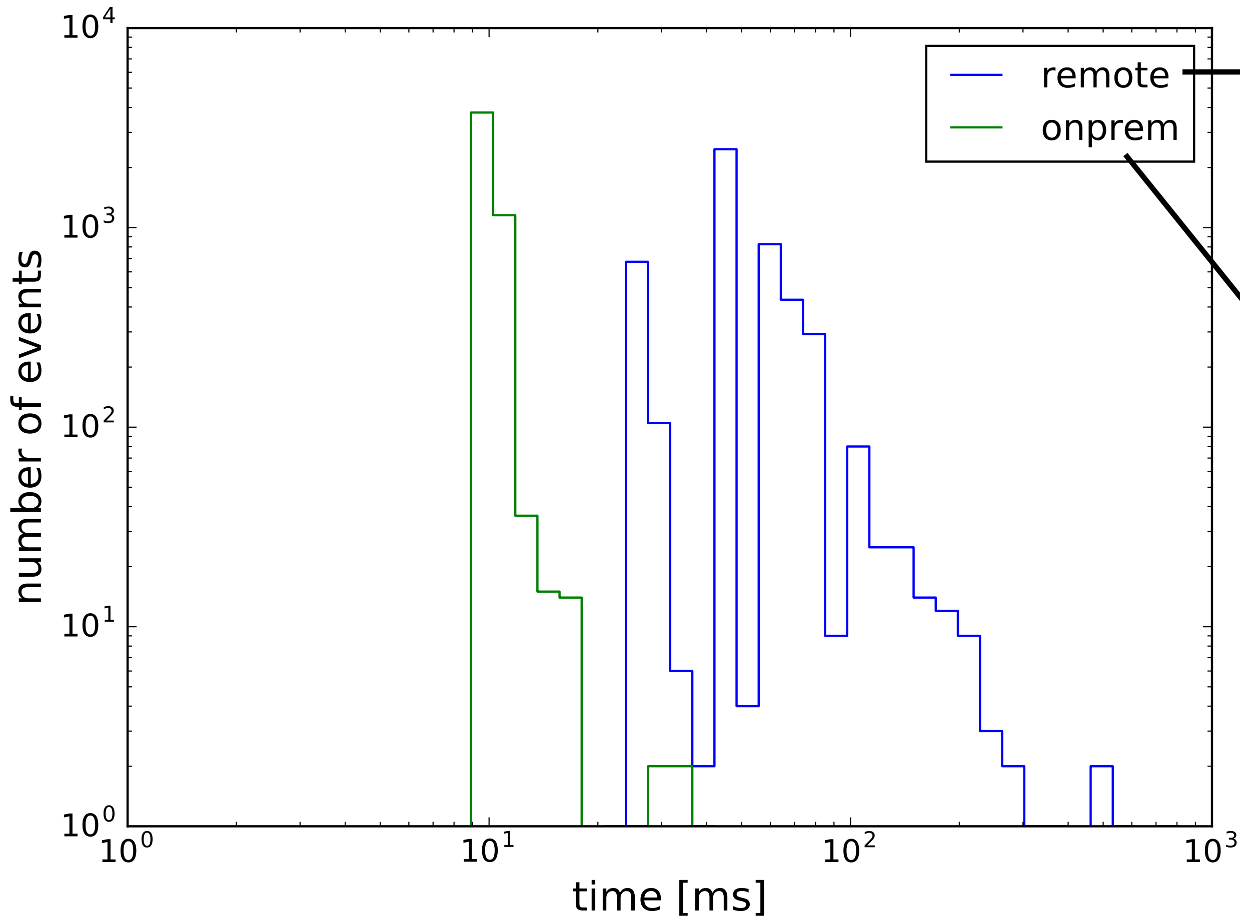
- On-site co-processors interesting solution for HLT computing farm where latency is the bottleneck
- For offline, better solution is using co-processors as a service on the cloud
  - *not feasible to buy specialized hardware for each T1, T2, T3 computing center*
- **Project Brainwave provides a full scalable real-time AI service** on Azure cloud (more than just a single co-processor)
  - *Multi-FPGA+CPU fabric accelerating both computing and network*
  - *Caveat: currently supports only selected computing vision off-the-shelf networks*

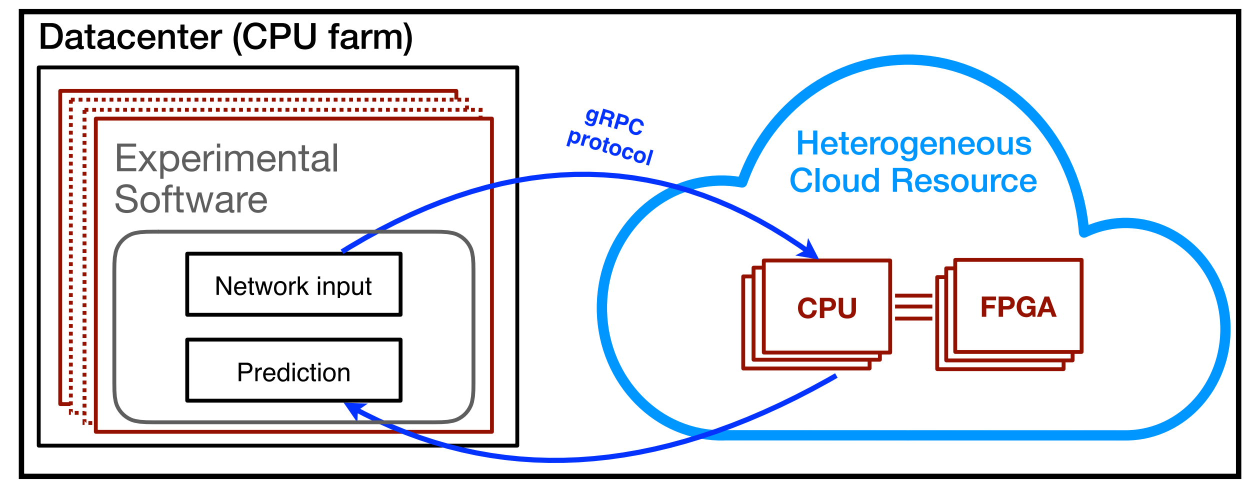


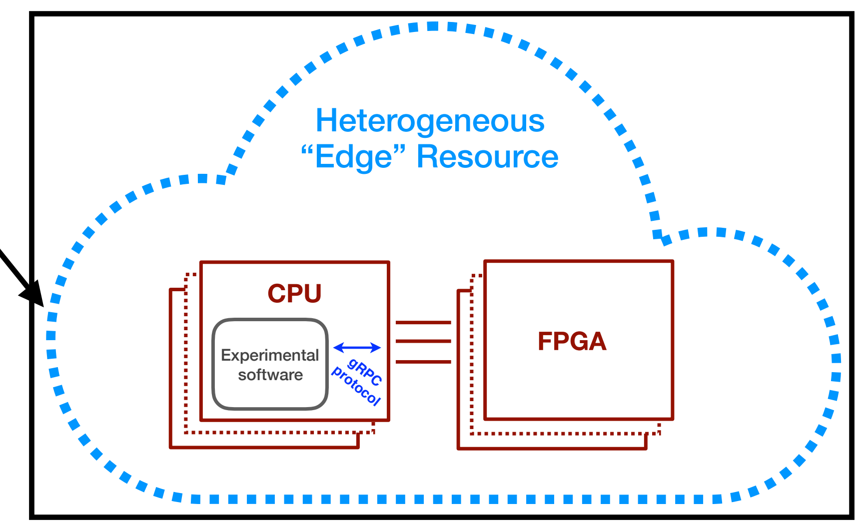


Co-processors as a service with Brainwave

See talk at ACAT19







remote test:
FROM CPU @ Fermilab, Illinois
TO Azure @ Virginia
→ <time> = 60 ms
(limited by distance and speed-of-light)

on-prem test:
run CMS software on Azure VM
→ <time> = 10 ms
(~ 2ms on FPGA, rest is classifier and I/O)

The HL-LHC is expected to start operations in 2026.
With data rates and pileup levels much higher than previously achieved
it will pose major challenges at all levels of data collection and processing.

Deep Learning and new computing technologies offer
the possibility to help facing these challenges.

Join this effort to keep making new discoveries at CERN possible!

Thank you!