Tutorial
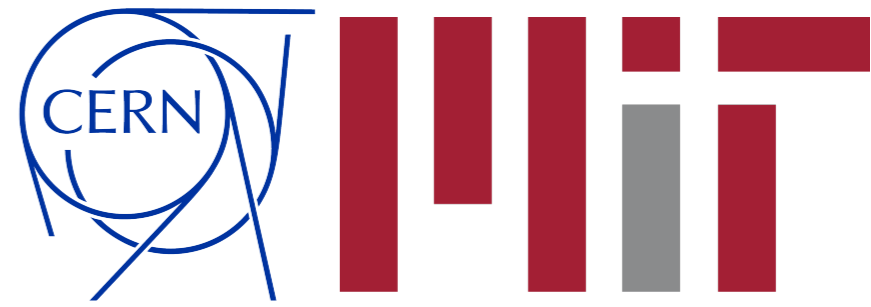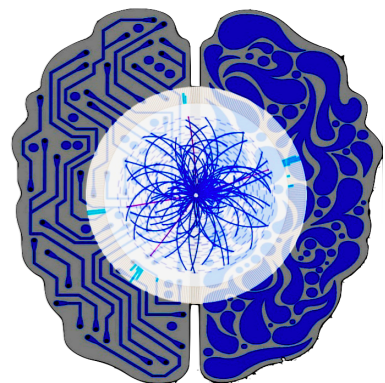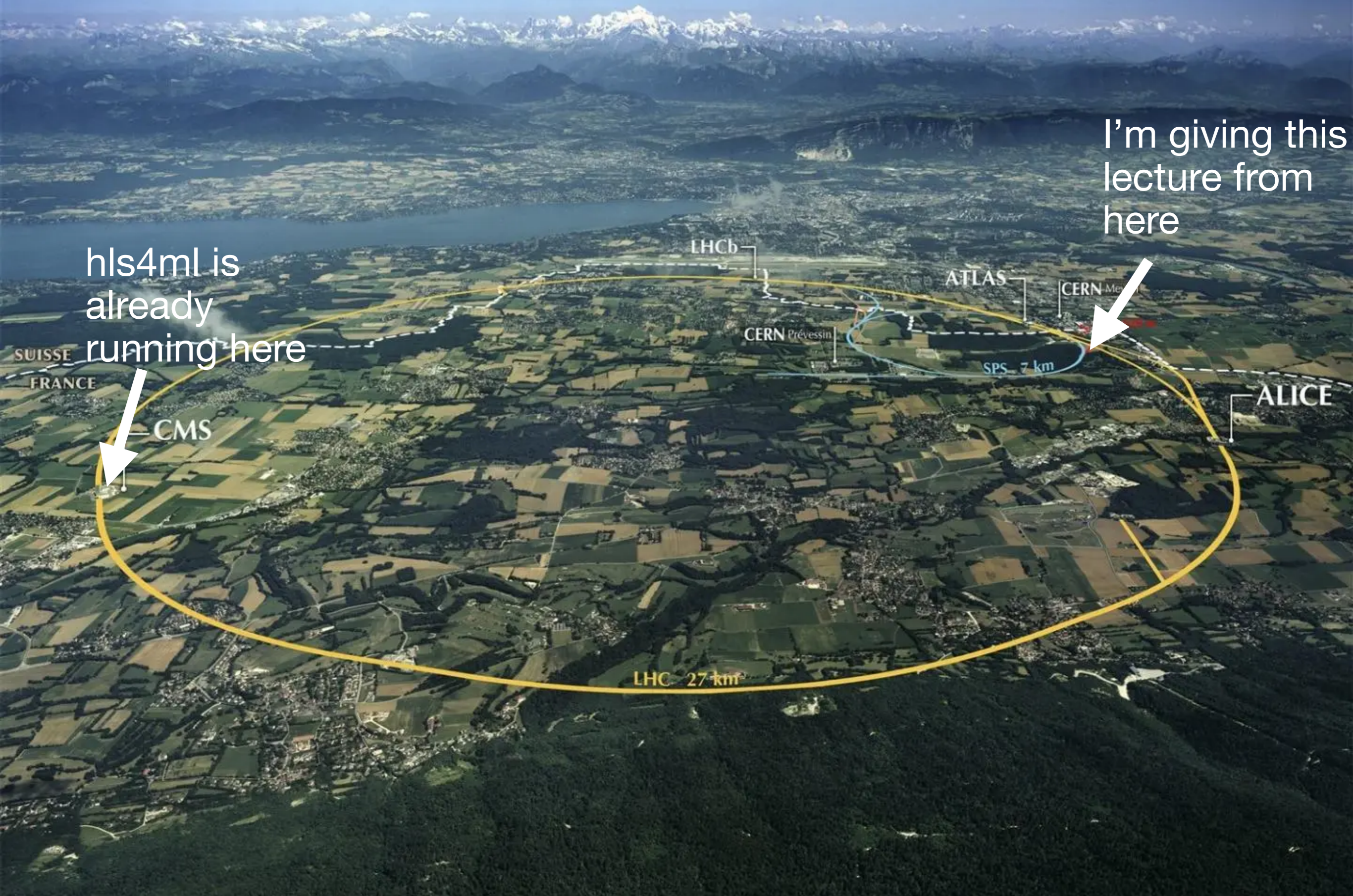
Duc Hoang (MIT) & Nhan Tran (Fermilab) for the FastML Community

A real use case

I'm giving this lecture from here

hls4ml is already running here

$\sim 10^{11}$ **protons**

$\sim 10^{11}$ **protons**

At the **Large Hadron Collider (LHC)**, galaxies of protons collide **40 millions times per second**.
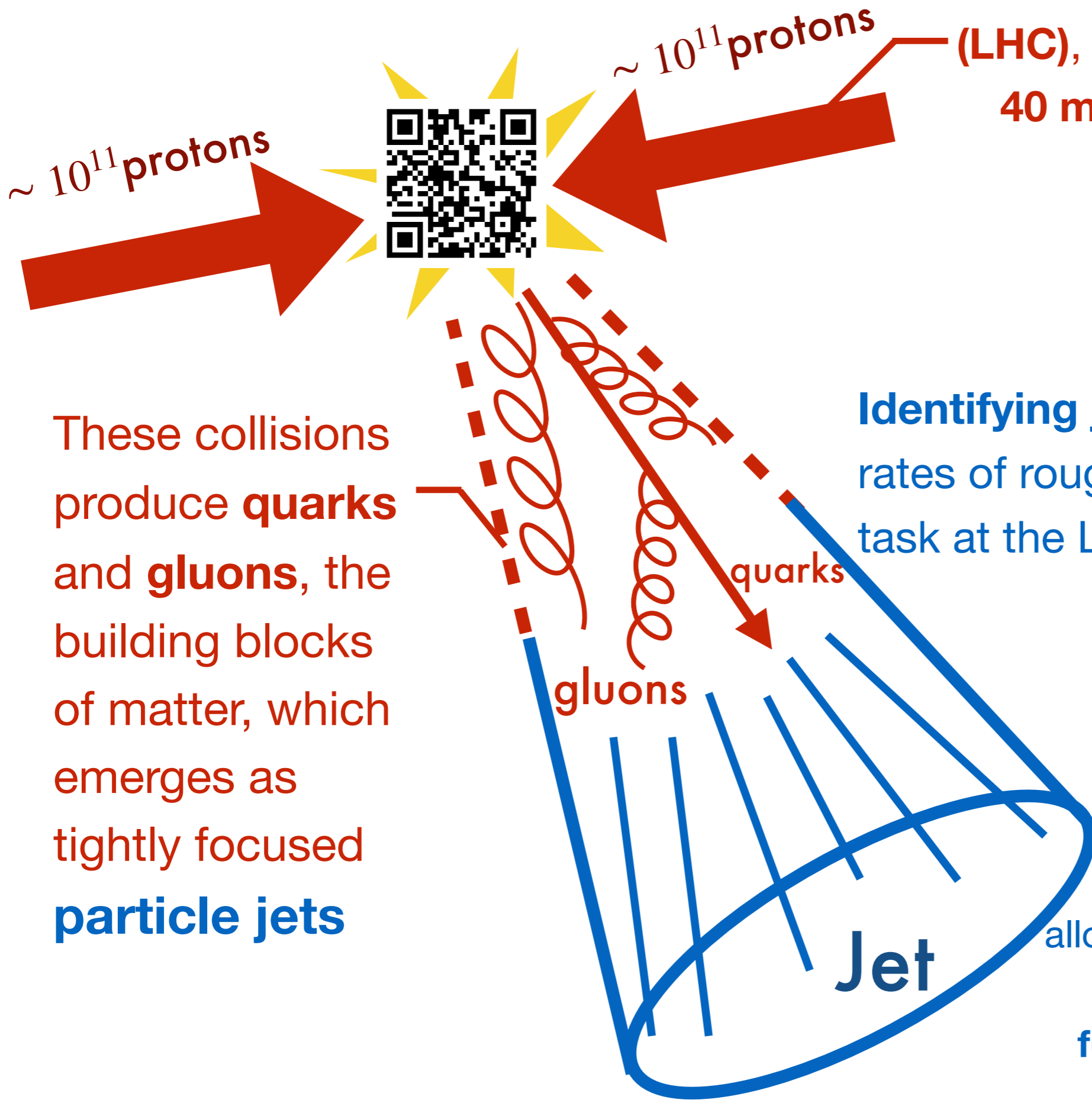
These collisions produce **quarks** and **gluons**, the building blocks of matter, which emerges as tightly focused **particle jets**

Identifying jets in real time at data rates of roughly **Petabit/s** is a crucial task at the LHC.
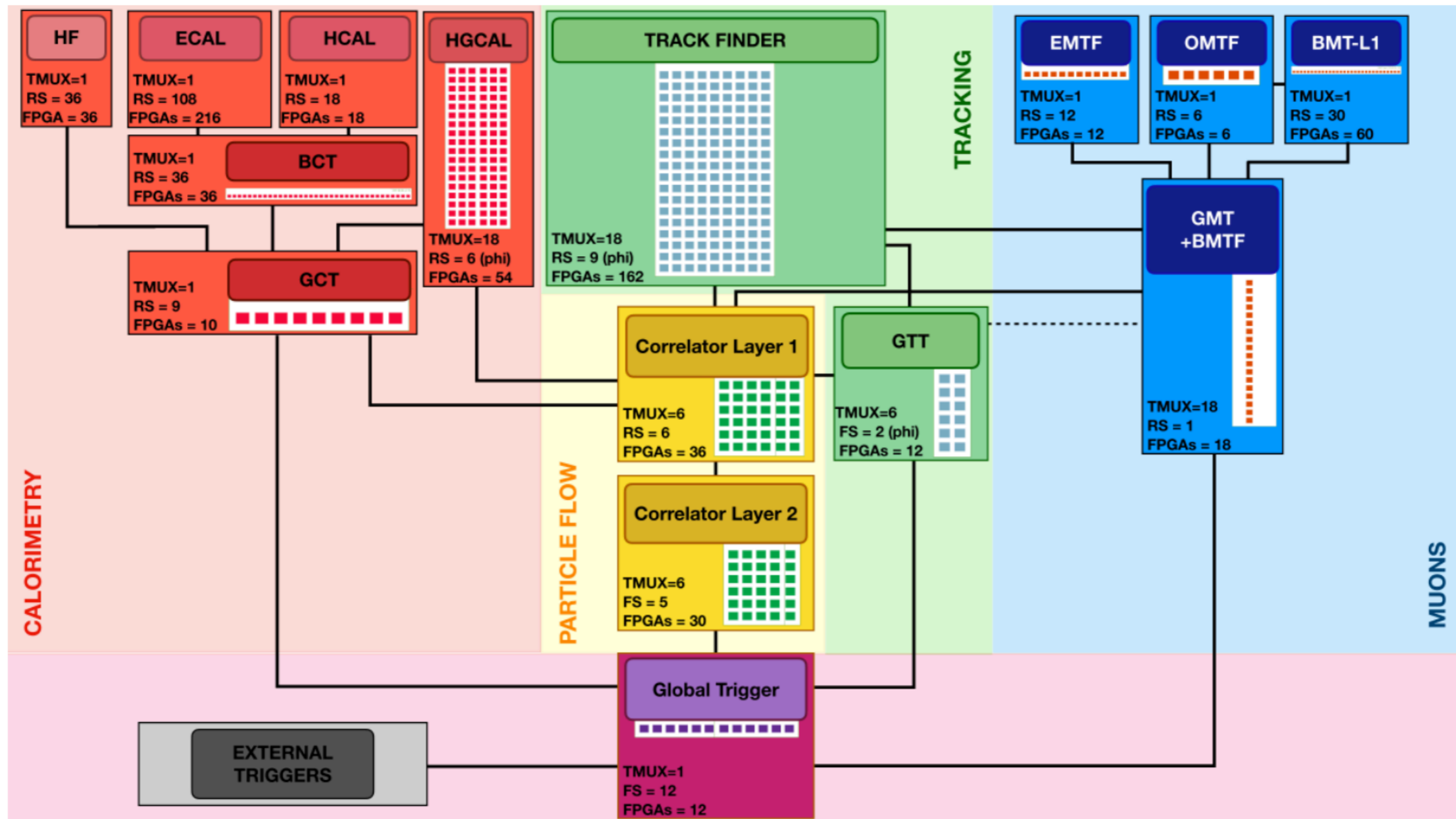
quarks

gluons

Jet

Successful jet identification allows scientists to **reconstruct collisions** and probe the **fundamental physics behind them**.
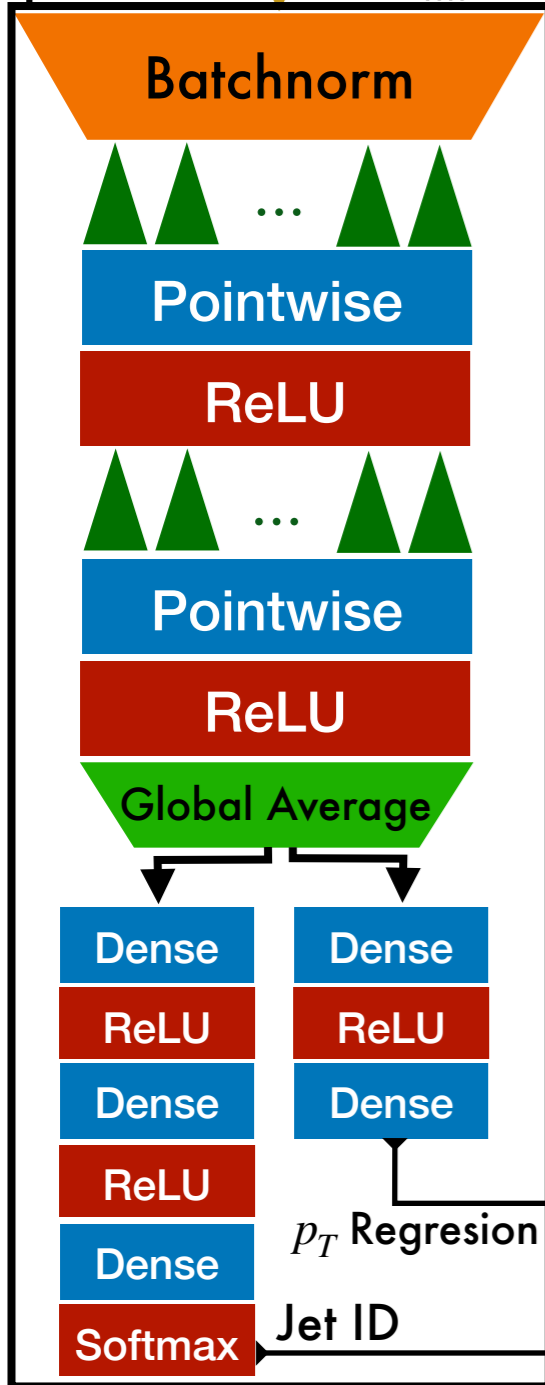
CMS Detector O(Pb/s) data rate

Level 1 trigger

A farm of **500 FPGAs** will receive data from the detector

The system have a $12.5\ \mu s$ time window to process an event.

Real-time Machine Learning

Reconstruct particles constituents

Sort particles by their momentum (p_T)

Standard jet reconstruction chain

Jet objects

Synchronizer Final jet objects

16 particles

hls 4 ml

Batchnorm

...

Pointwise

ReLU

Pointwise

ReLU

Global Average

Dense | Dense

ReLU | ReLU

Dense | Dense

ReLU

Dense

Softmax

p_T Regresion

Jet ID

Each event is assigned a **unique token** for tracking during parallel processing.

NN results and jet objects are saved in **64 bit words**

003C000000000000
0072000000000000

0018DBFA023A
0018F3F9425B

① == ② ?

003C 0018DBFA023A
0072 0018F3F9425B

NN	%VU9P
LUT	35
FF	16
DSPs	15
BRAM	0
Latency	148ns (II=2.8ns)

Floorplan (*Xilinx FPGA VU9P-2*)

Deregionizer | NN
JetLoop | Rest of Algo
JetComputer
JetSort

Correlator Layer 1

TMUX=6
RS = 6
FPGAs = 36

Correlator Layer 2

TMUX=6
FS = 5
FPGAs = 30
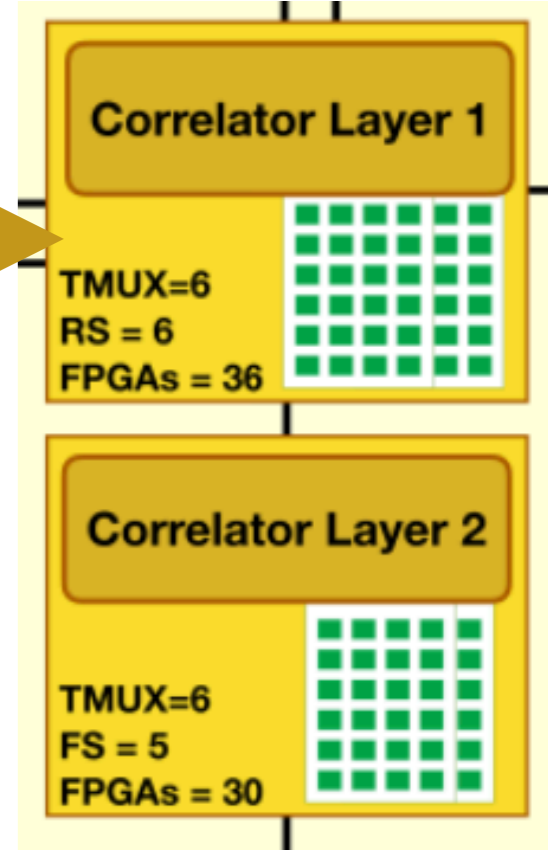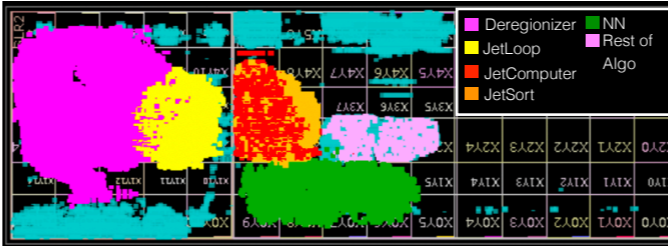
The **L1 trigger** decides if detector data is sent to next tier.
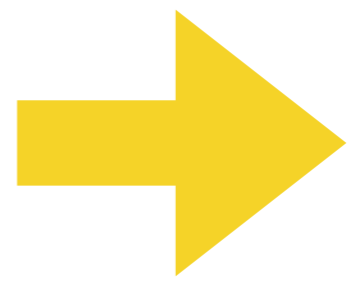
Yes/No?

Increased Physics Performance

CMS *Phase-2 Simulation Preliminary* 14 TeV, 200 PU

Efficiency

HH → bbbb events
b-tag NN @ 15 kHz
QuadJet+HT OR Jets+Muon @ 15 kHz
QuadJet+HT @ 11 kHz

Ratio to QuadJet+HT

m_{HH} (gen) [GeV]
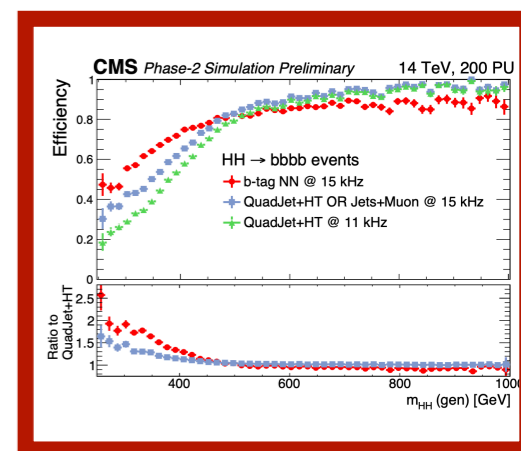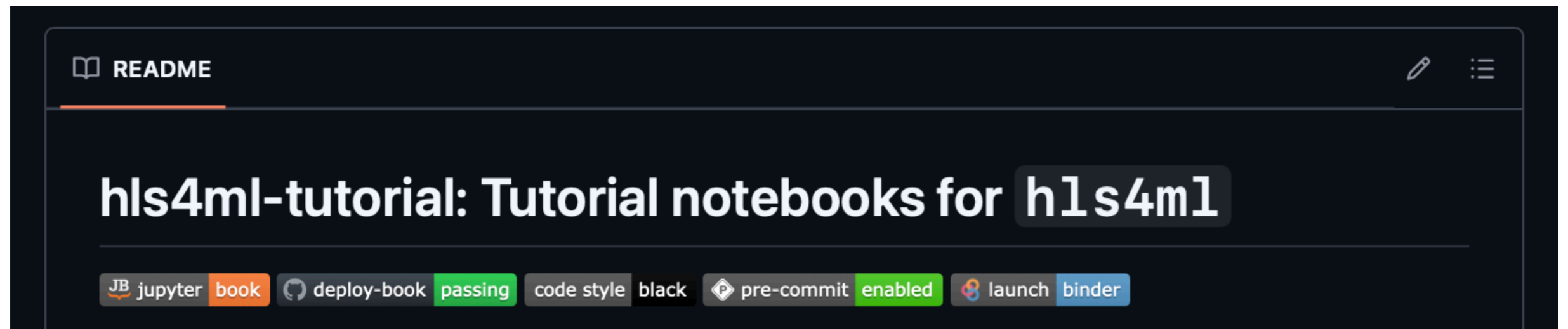
Increased signal efficiency in the most background-dominant regions.

The tutorial will walk you through how you can create your own NN on FPGA system!

- https://tutorials.fastmachinelearning.org/
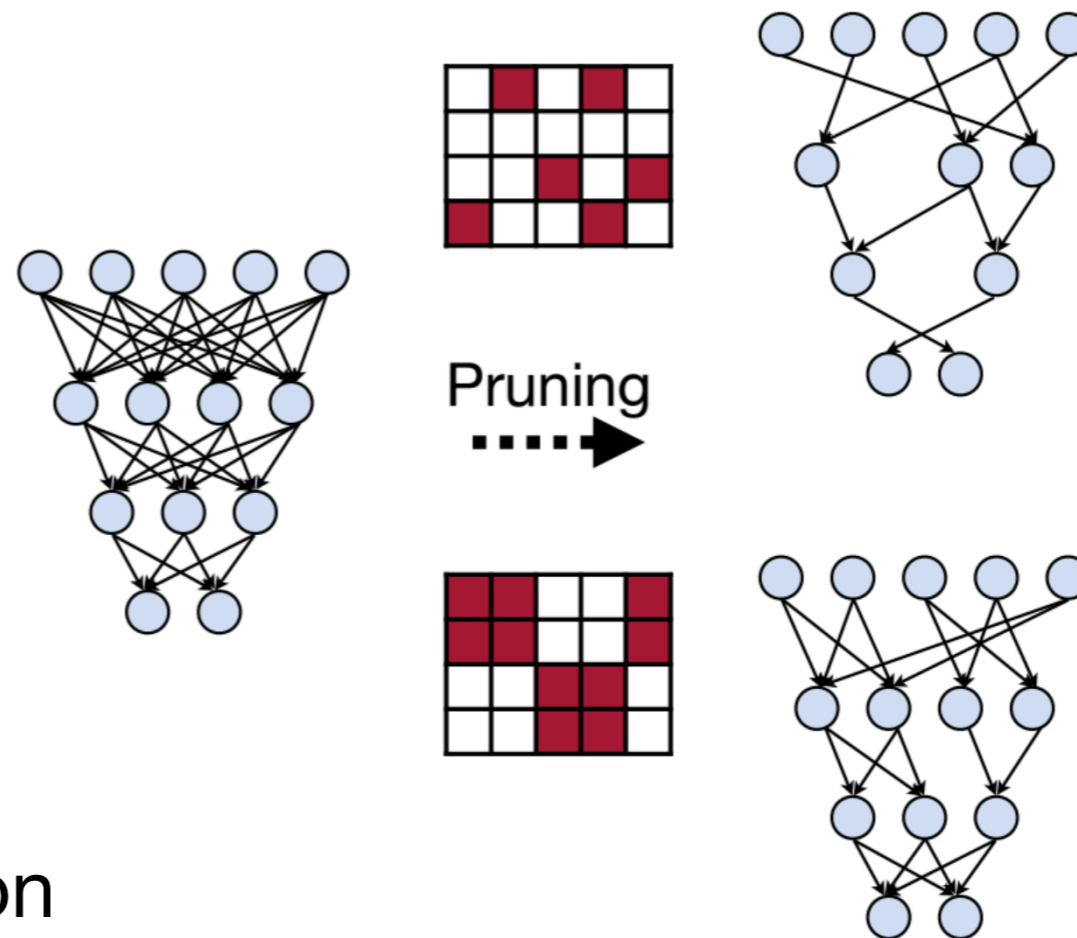
- Done in Jupyter Notebook!

Focusing on 3 topics today

1. Train a NN (for jet tagging)

2. Train a NN with pruning

3. Train a NN with quantization