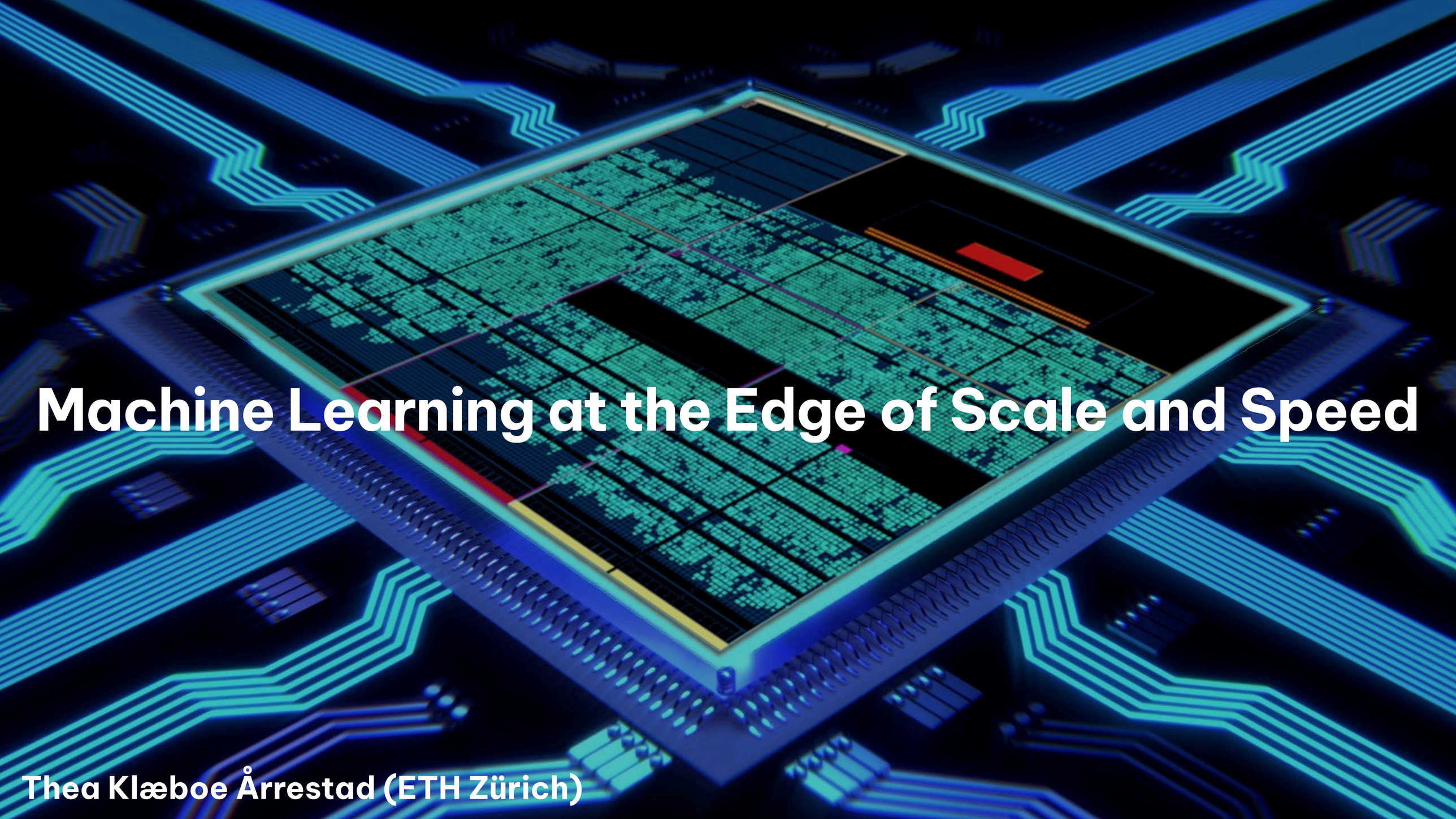


Machine Learning at the Edge of Scale and Speed

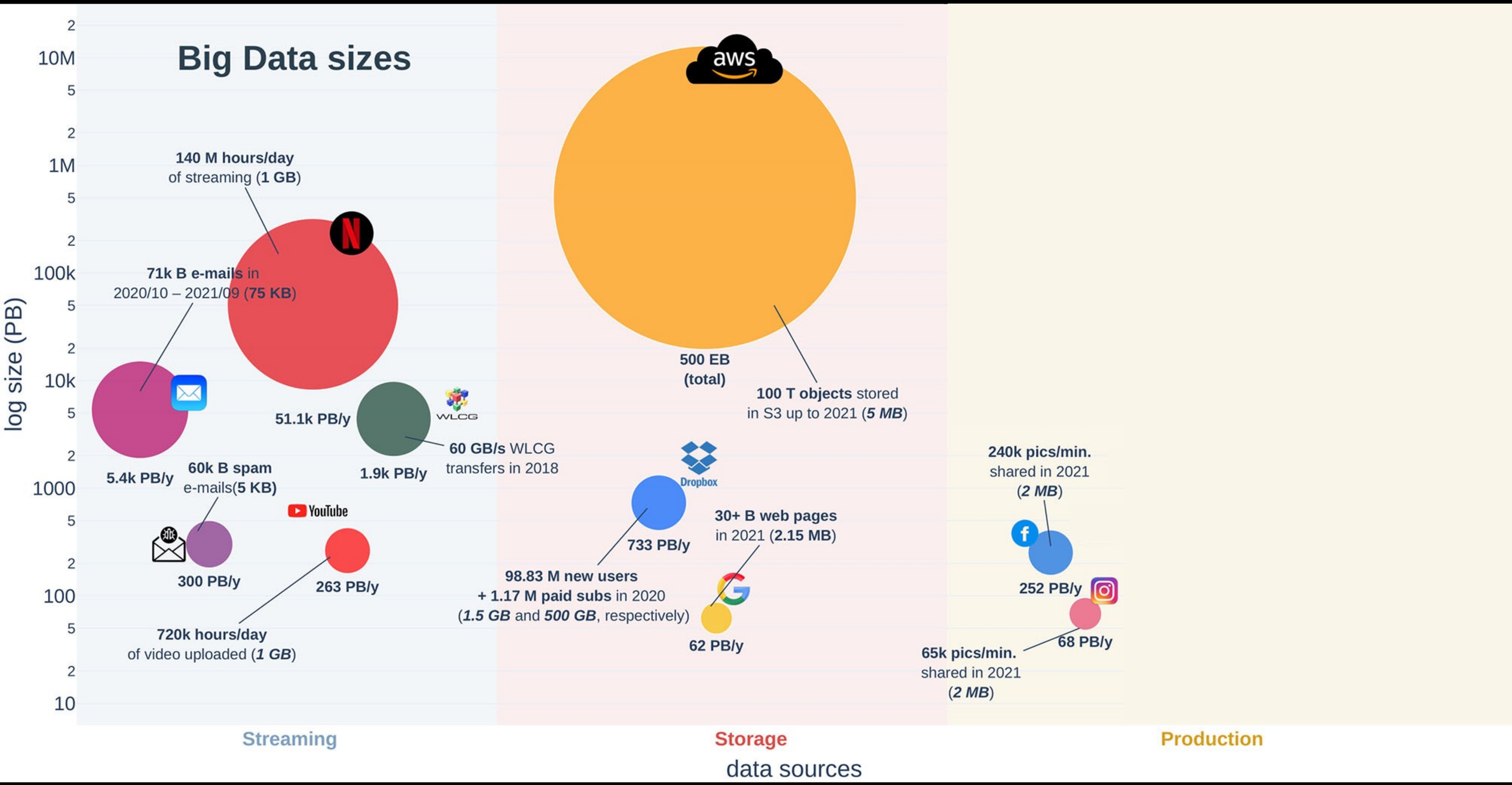
Thea Klæboe Årrestad (ETH Zürich)

The Zettabyte Era

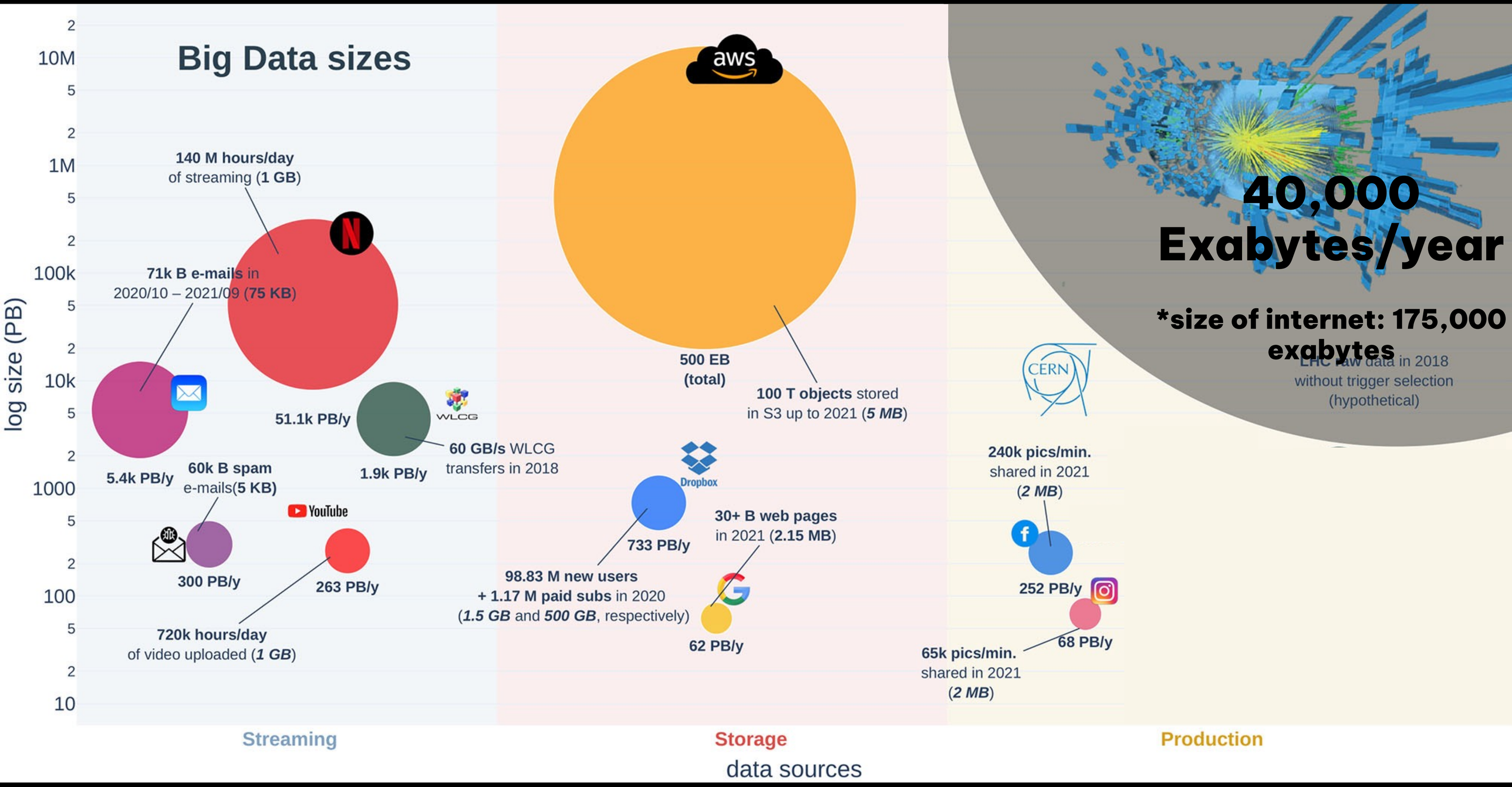
When will global Internet traffic reach an annual run rate of one Zettabyte? Well, that day has finally come. According to our estimate, the world's collective Internet use will reach the Zettabyte threshold for this calendar year on September 9, 2016. Finally... Consider the fact that the Internet essentially began to scale for global consumer

... (~size of internet ~175 Zettabytes)

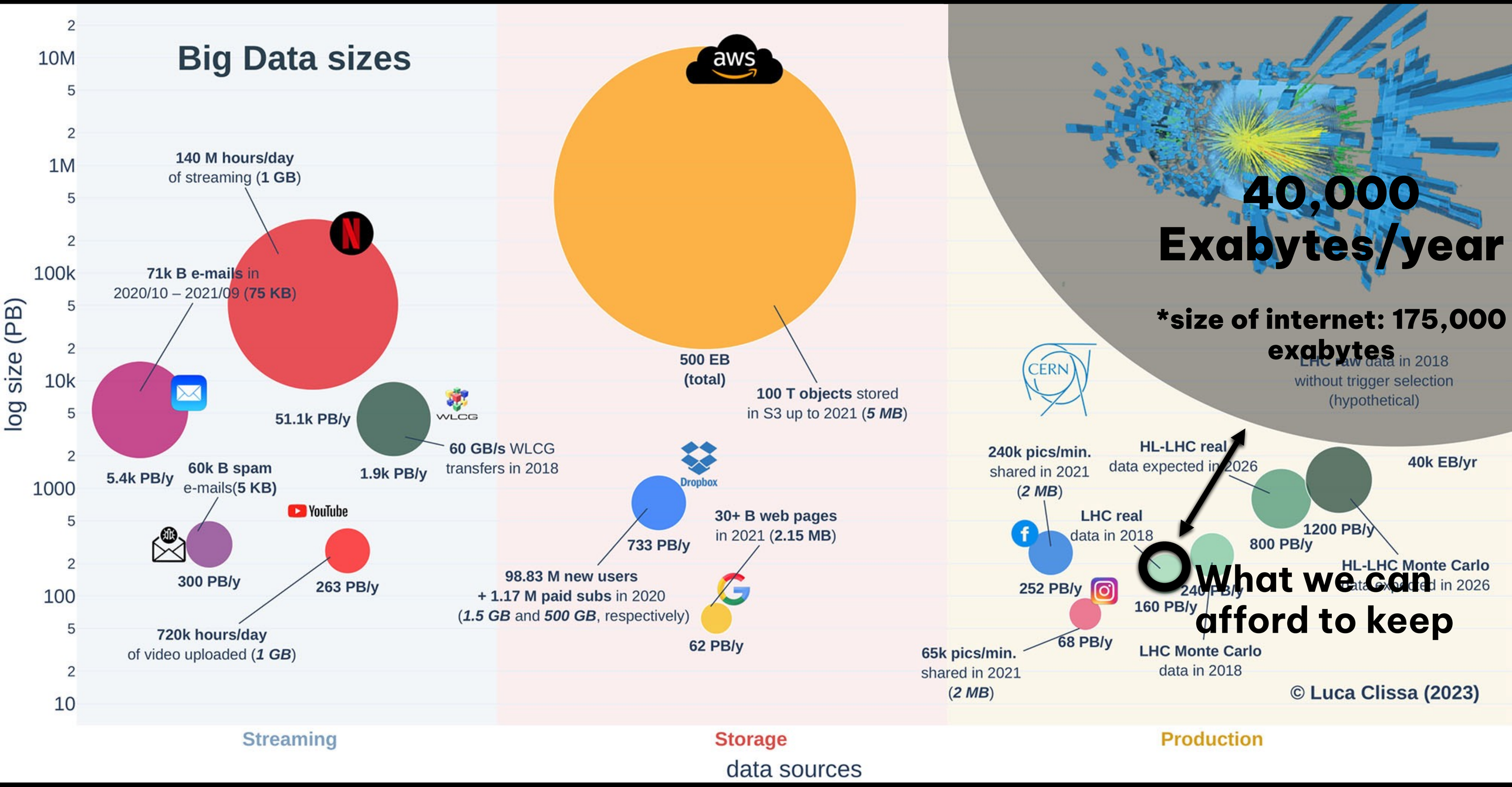
Cisco Annual Internet Report



Clissa et al. 10.3389/fdata.2023.1271639

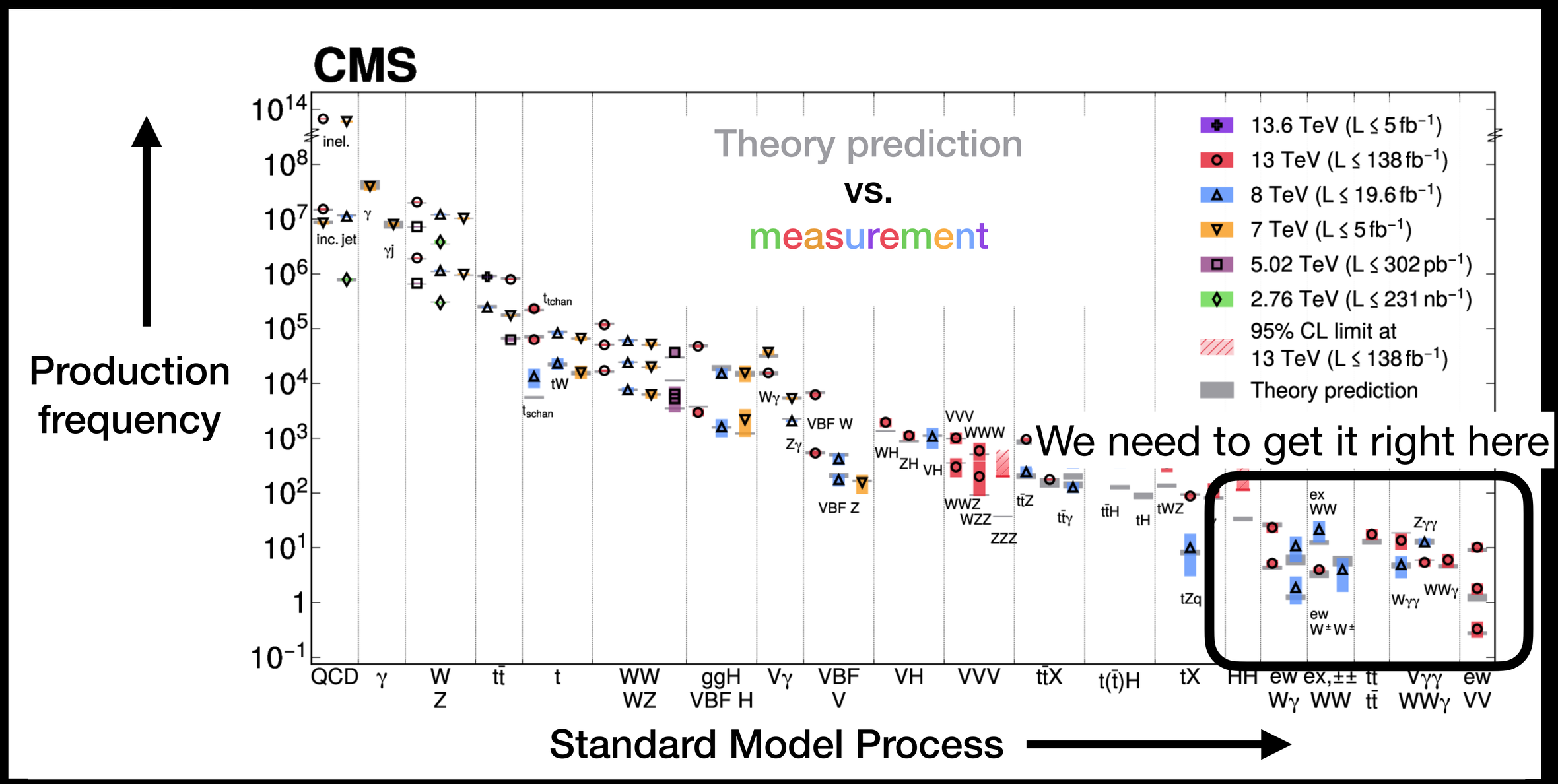


Clissa et al. 10.3389/fdata.2023.1271639

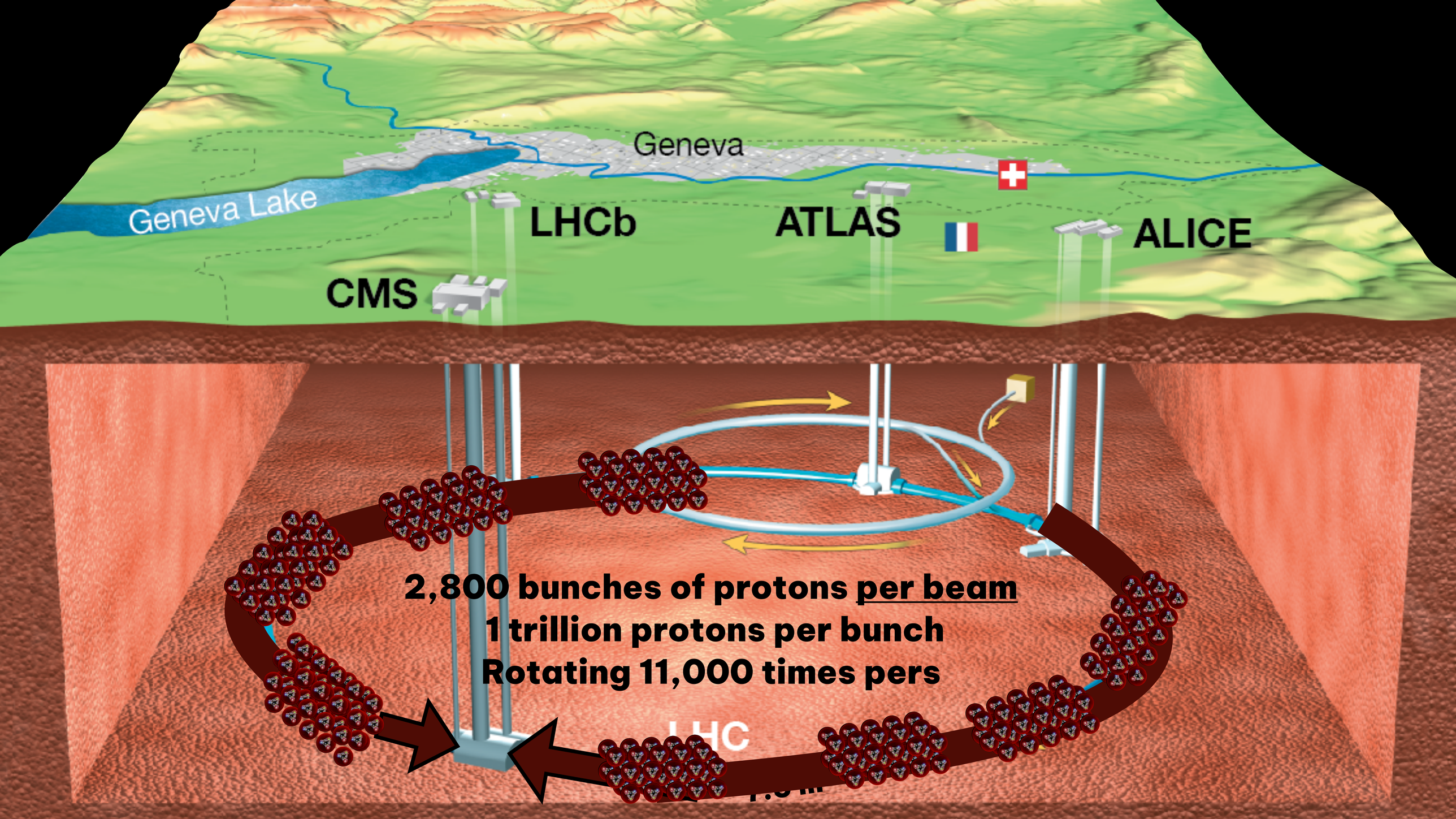


Clissa et al. 10.3389/fdata.2023.1271639

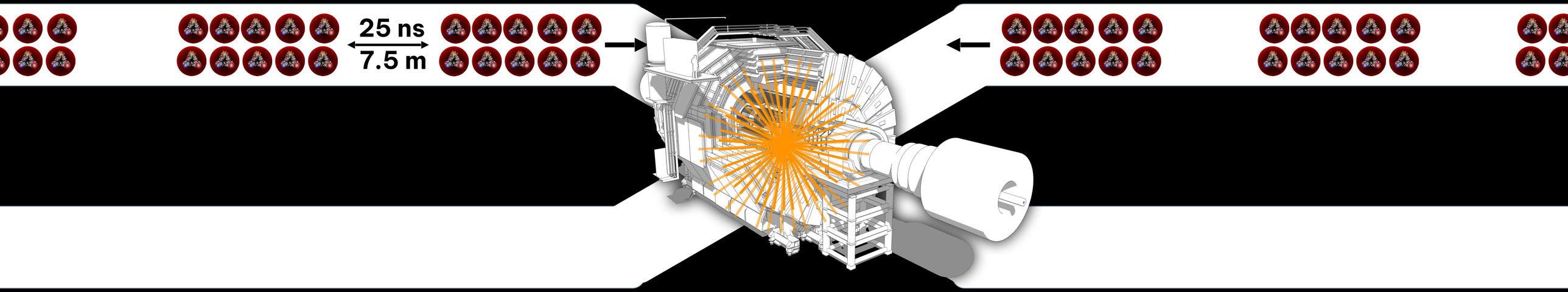
“While the humans guessed correctly 70 percent of the time, RankBrain had an 80 percent success rate [...]”


CMS
Theory prediction
vs.
measurement
13.6 TeV (L ≤ 5 fb⁻¹)
13 TeV (L ≤ 138 fb⁻¹)
8 TeV (L ≤ 19.6 fb⁻¹)
7 TeV (L ≤ 5 fb⁻¹)
5.02 TeV (L ≤ 302 pb⁻¹)
2.76 TeV (L ≤ 231 nb⁻¹)
95% CL limit at
13 TeV (L ≤ 138 fb⁻¹)
Theory prediction
Production frequency
Standard Model Process
We need to get it right here
QCD
γ
W Z
tt̄
t
WW WZ
ggH VBF H
Vγ
VBF V
VH
VVV
ttX
t(t̄)H
tX
HH
ew Wγ
ex,±± WW
tt tt̄
Vγγ WWγ
ew VV


Get it right one in a one hundred trillionth
Machine Learning and the needle in the needle stack

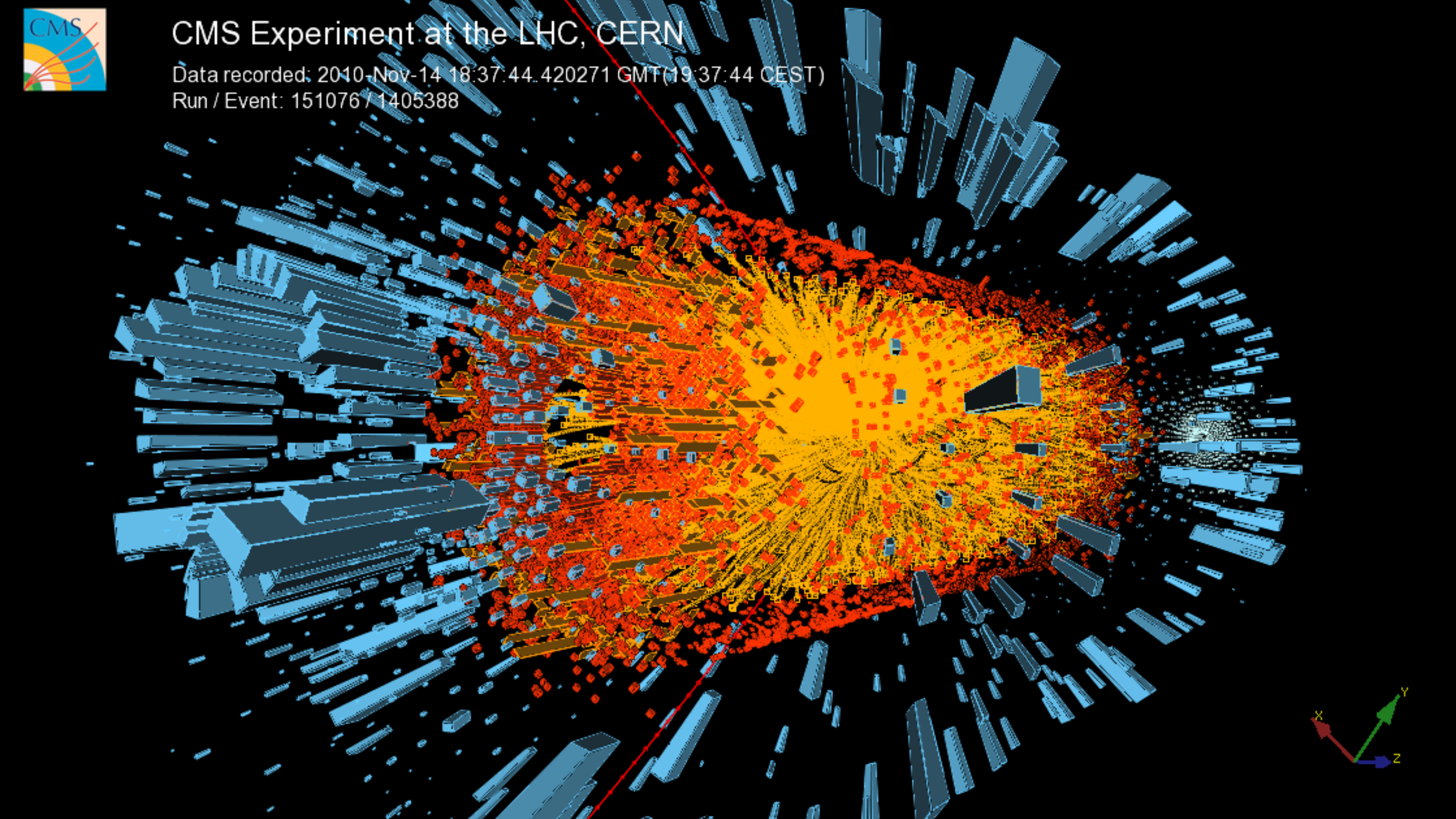

Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
2,800 bunches of protons per beam
1 trillion protons per bunch
Rotating 11,000 times pers
LHC

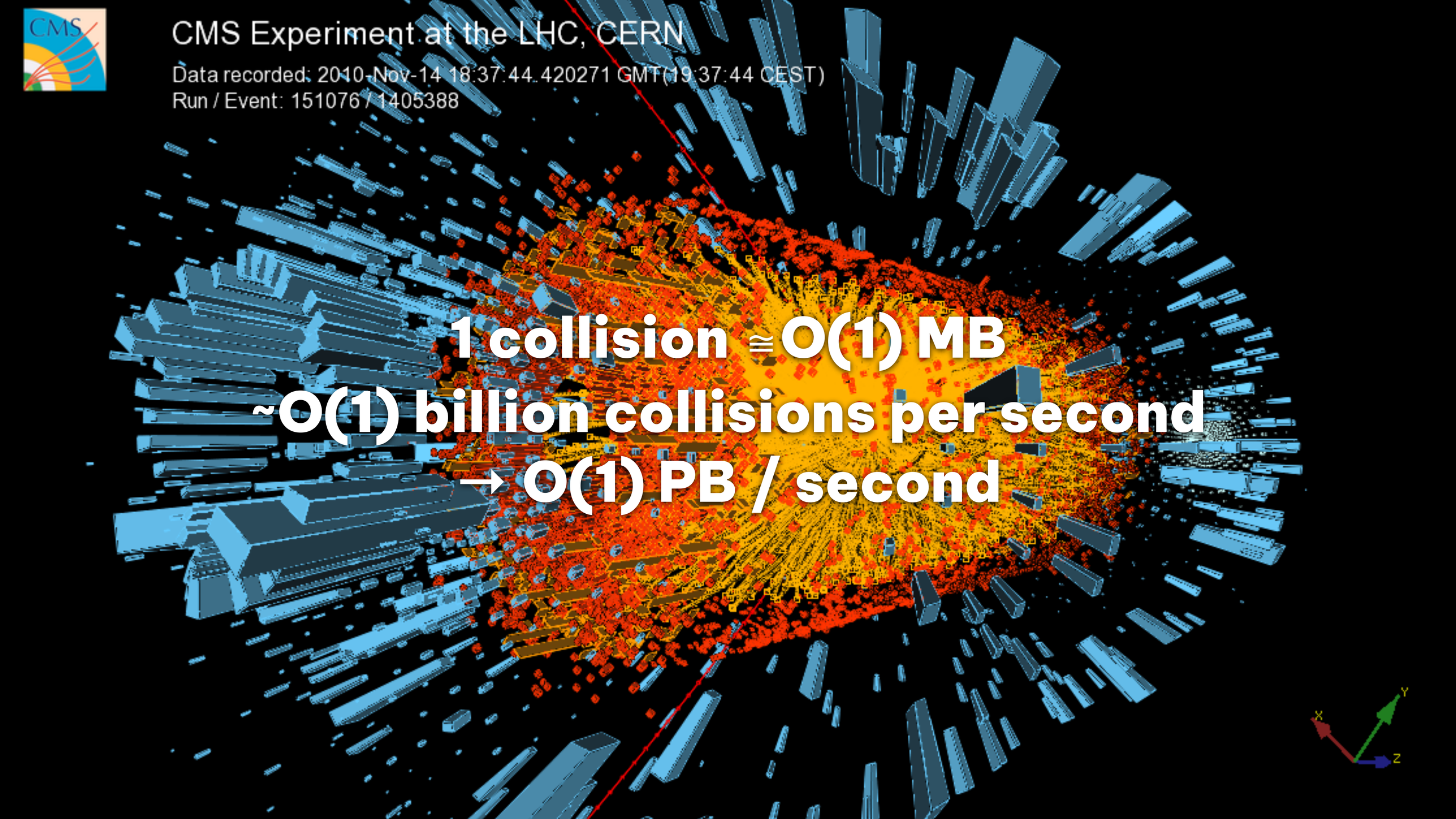
The bunches “collide” every 25 ns (40 MHz)

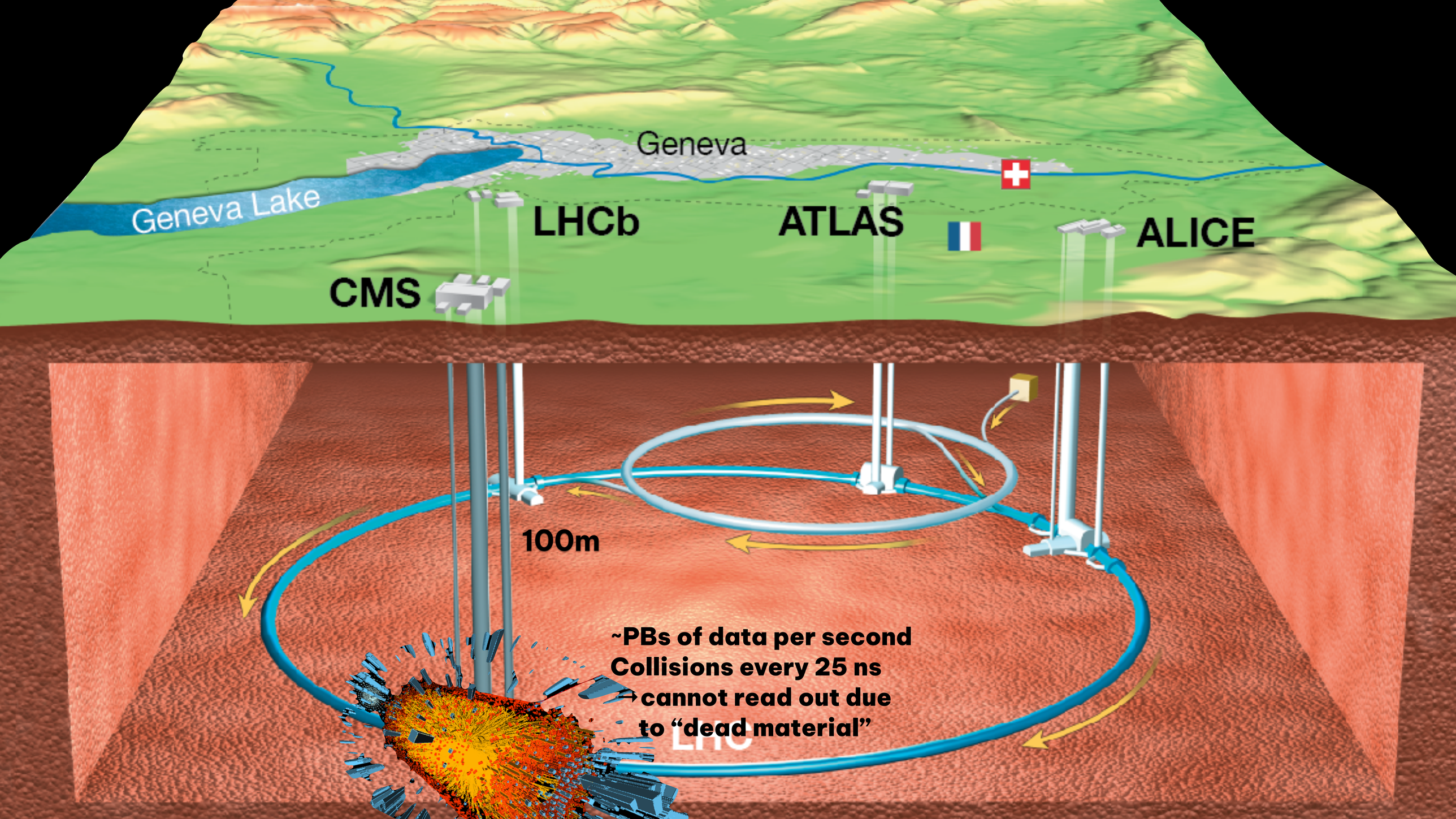


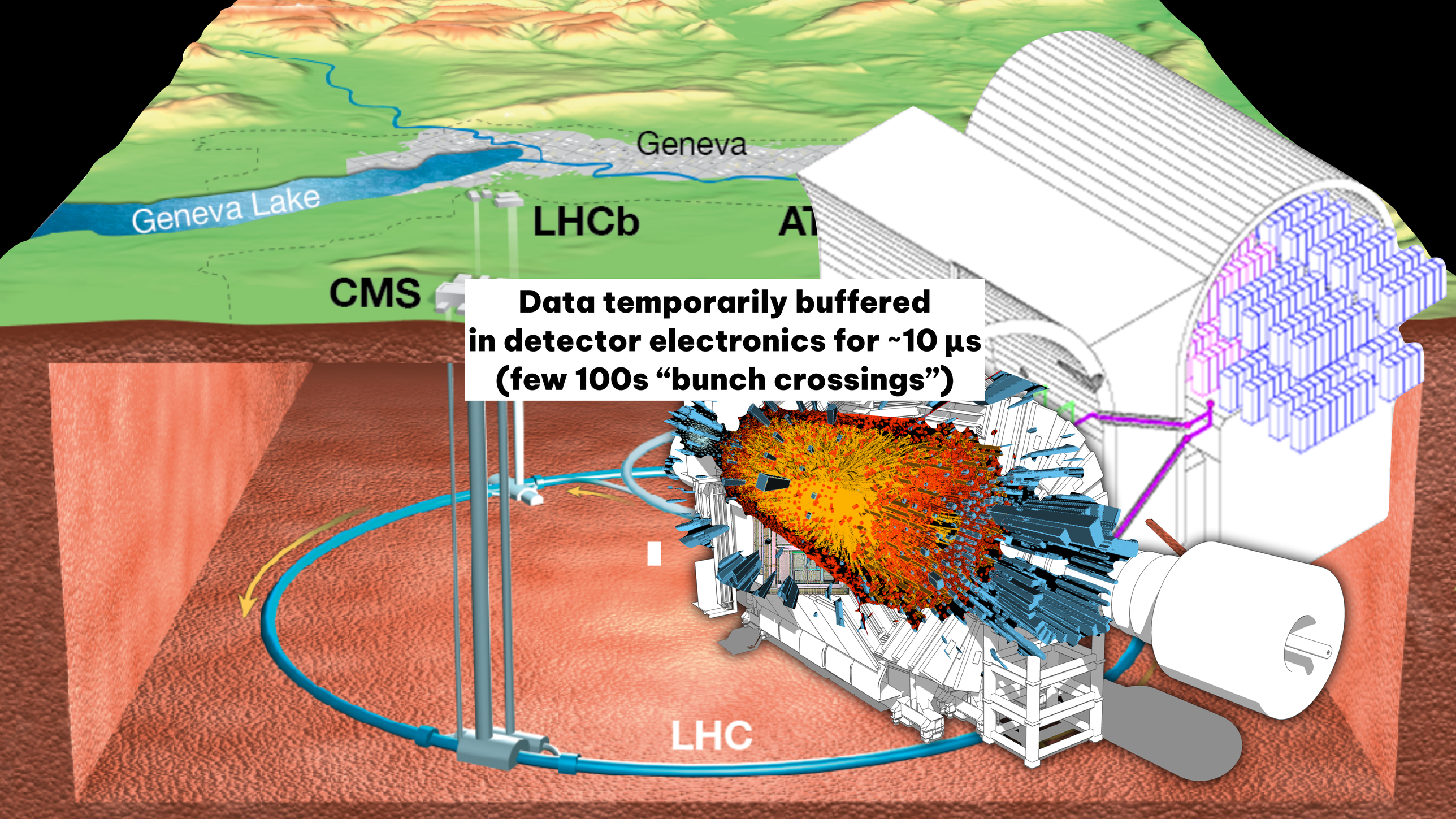
~Out of 10^{10} protons, only 60 collisions per crossing

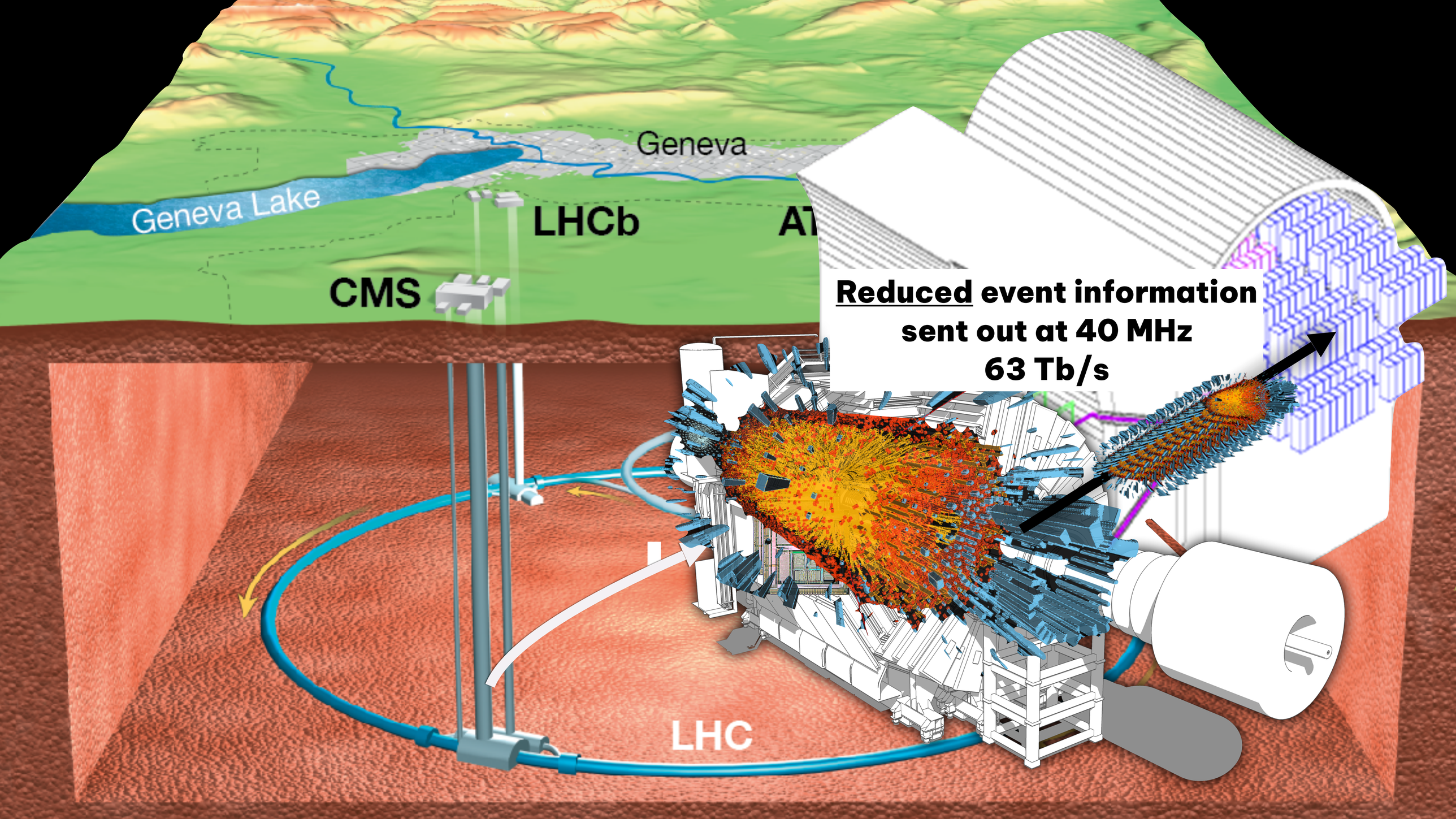
Interaction is extremely rare!

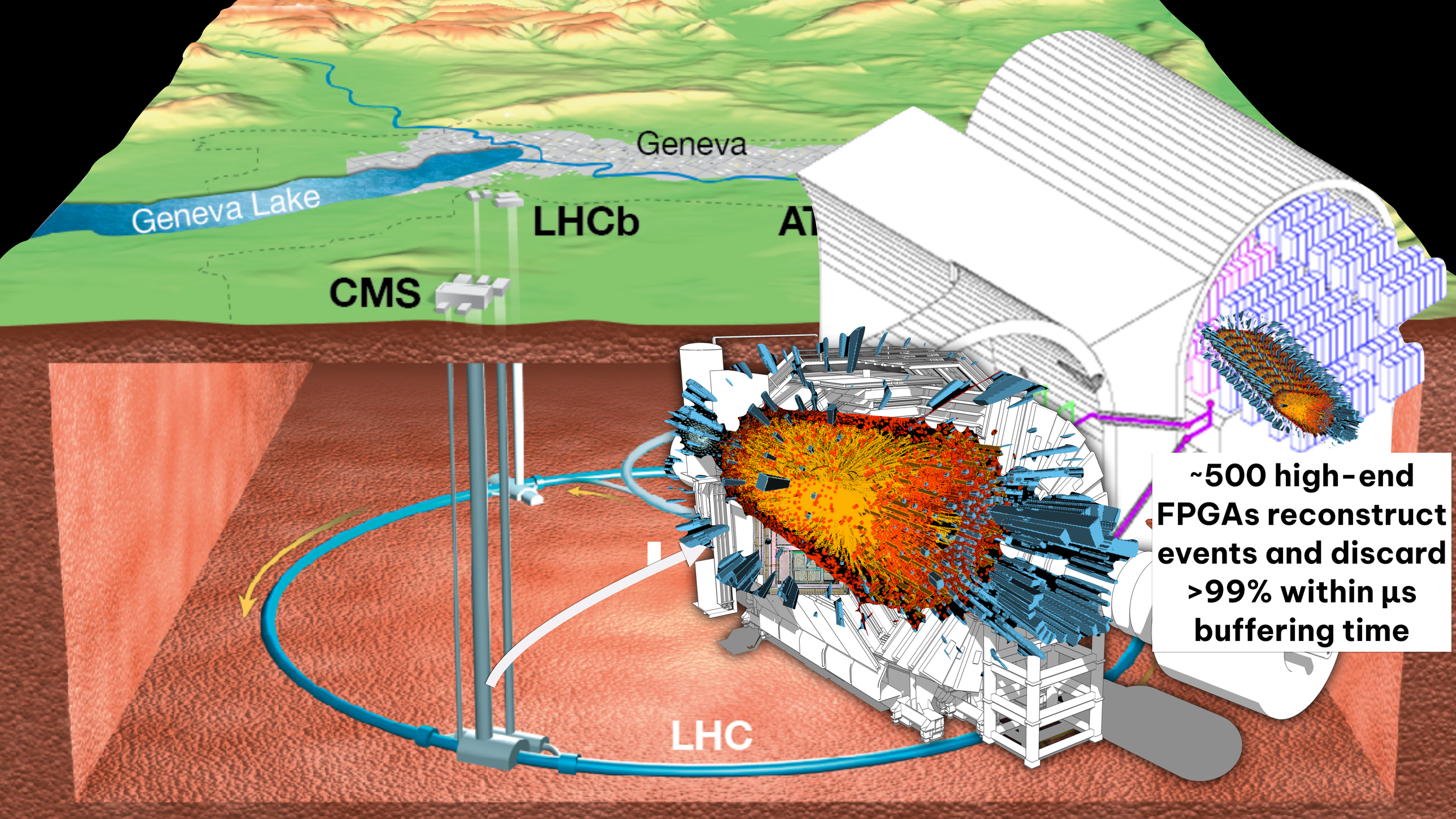

CMS Experiment at the LHC, CERN
Data recorded: 2010-Nov-14 18:37:44.420271 GMT(19:37:44 CEST)
Run / Event: 151076 / 1405388
Y
X
Z

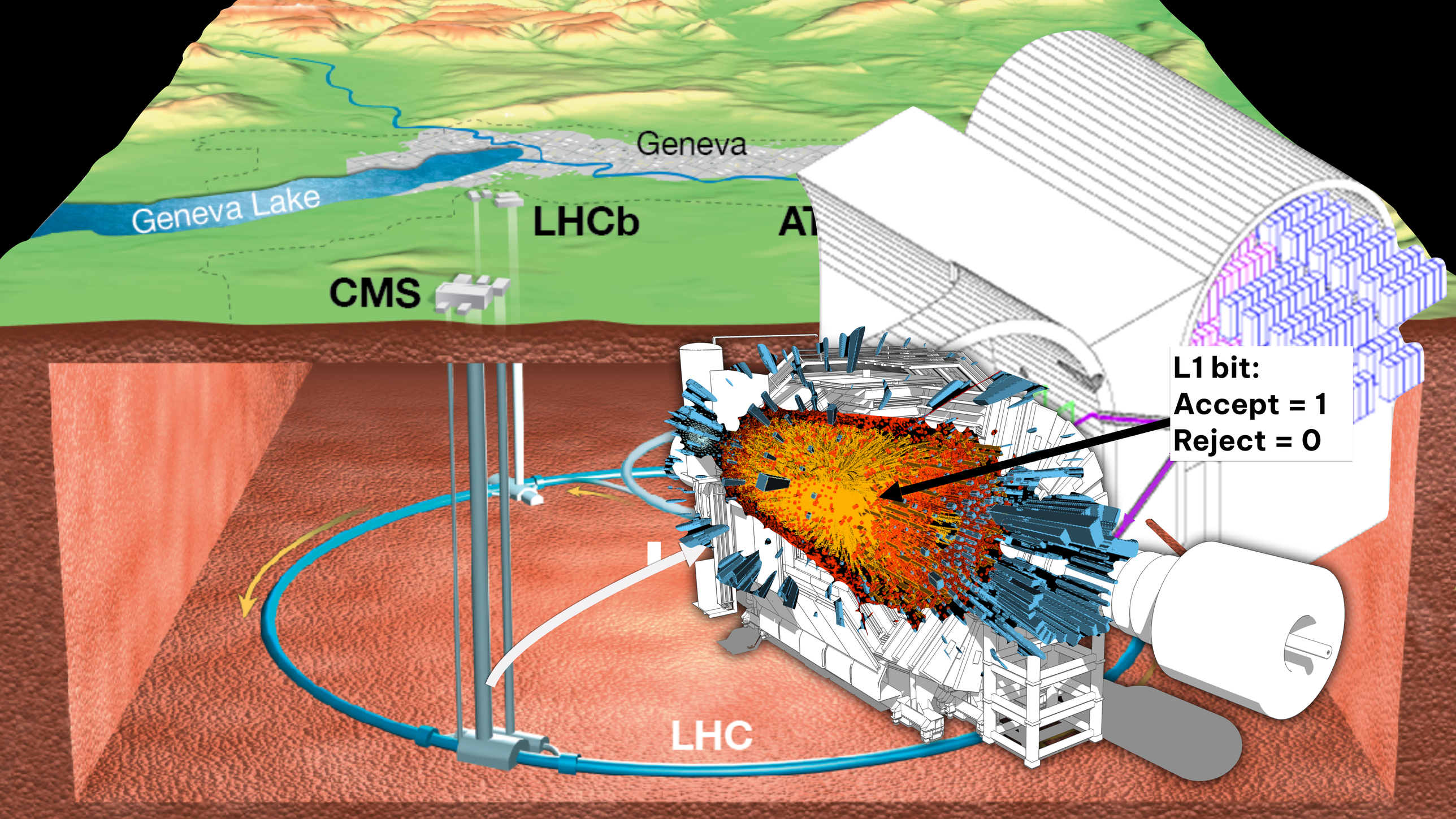

CMS Experiment at the LHC, CERN
Data recorded: 2010-Nov-14 18:37:44.420271 GMT(19:37:44 CEST)
Run / Event: 151076 / 1405388
1 collision ≅ O(1) MB
~O(1) billion collisions per second
→ O(1) PB / second
Y
X
Z

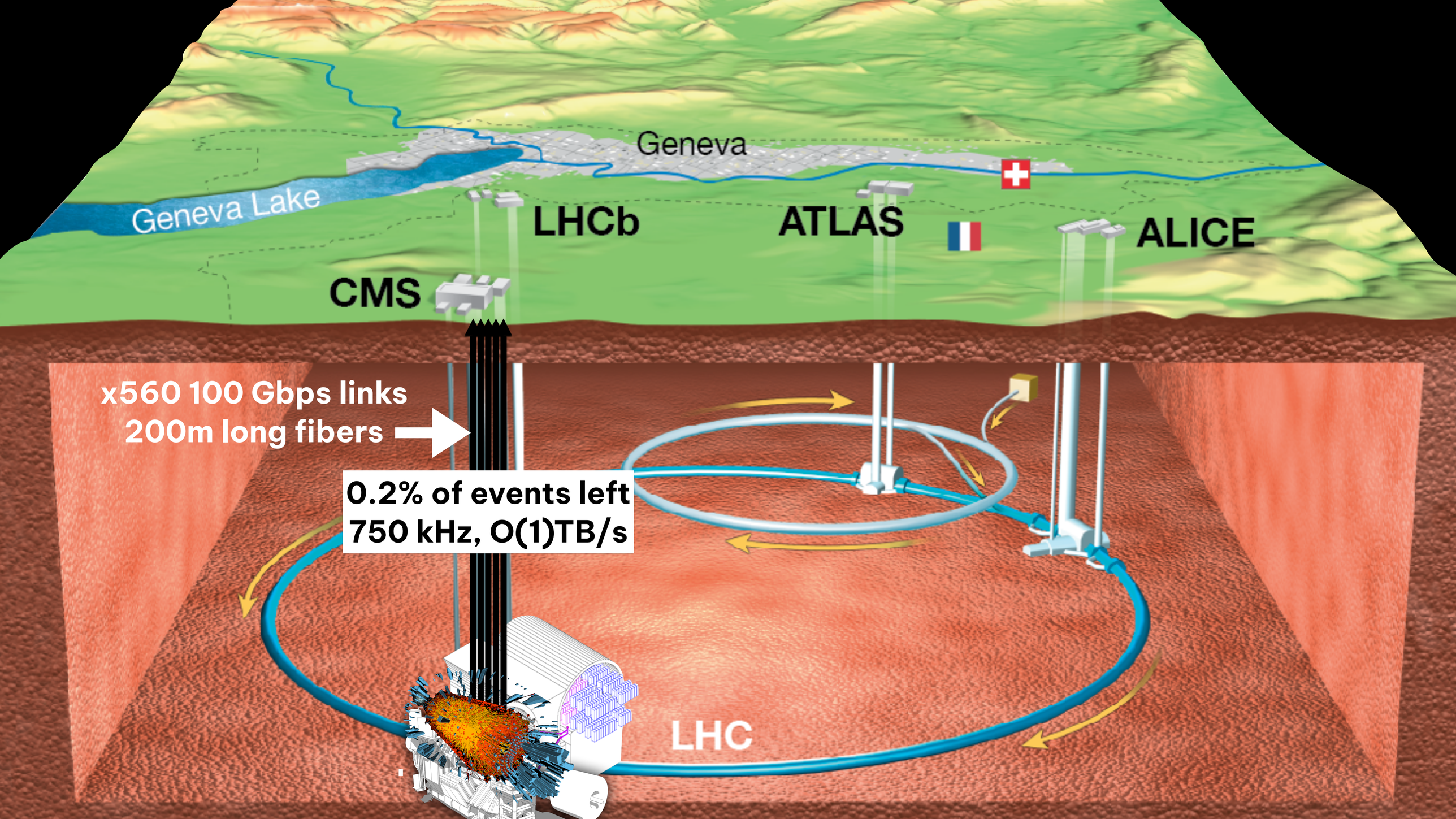

Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
100m
~PBs of data per second
Collisions every 25 ns
→ cannot read out due
to “dead material”
LHC

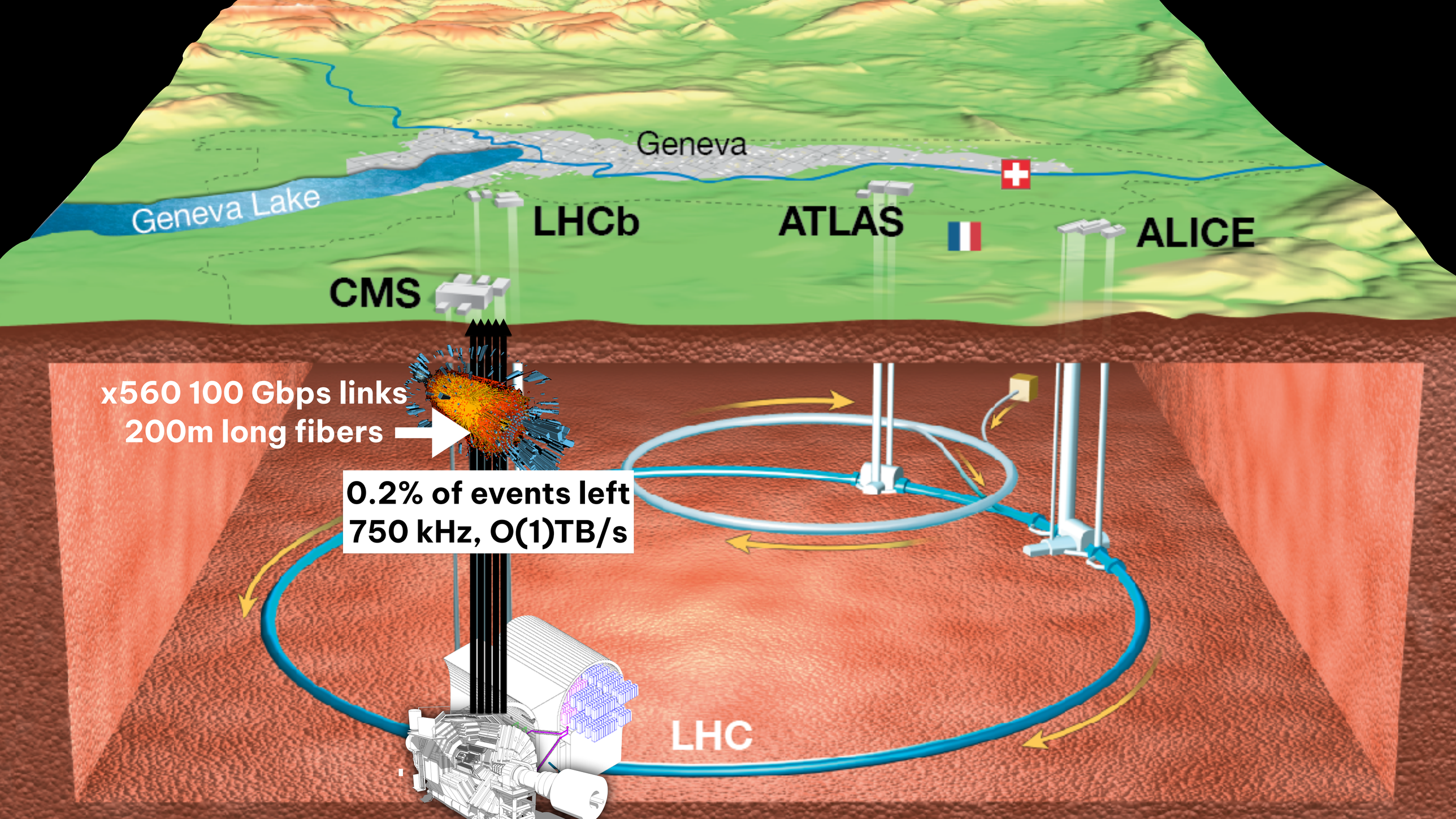

Geneva
Geneva Lake
LHCb
CMS
Data temporarily buffered
in detector electronics for ~10 µs
(few 100s “bunch crossings”)
LHC

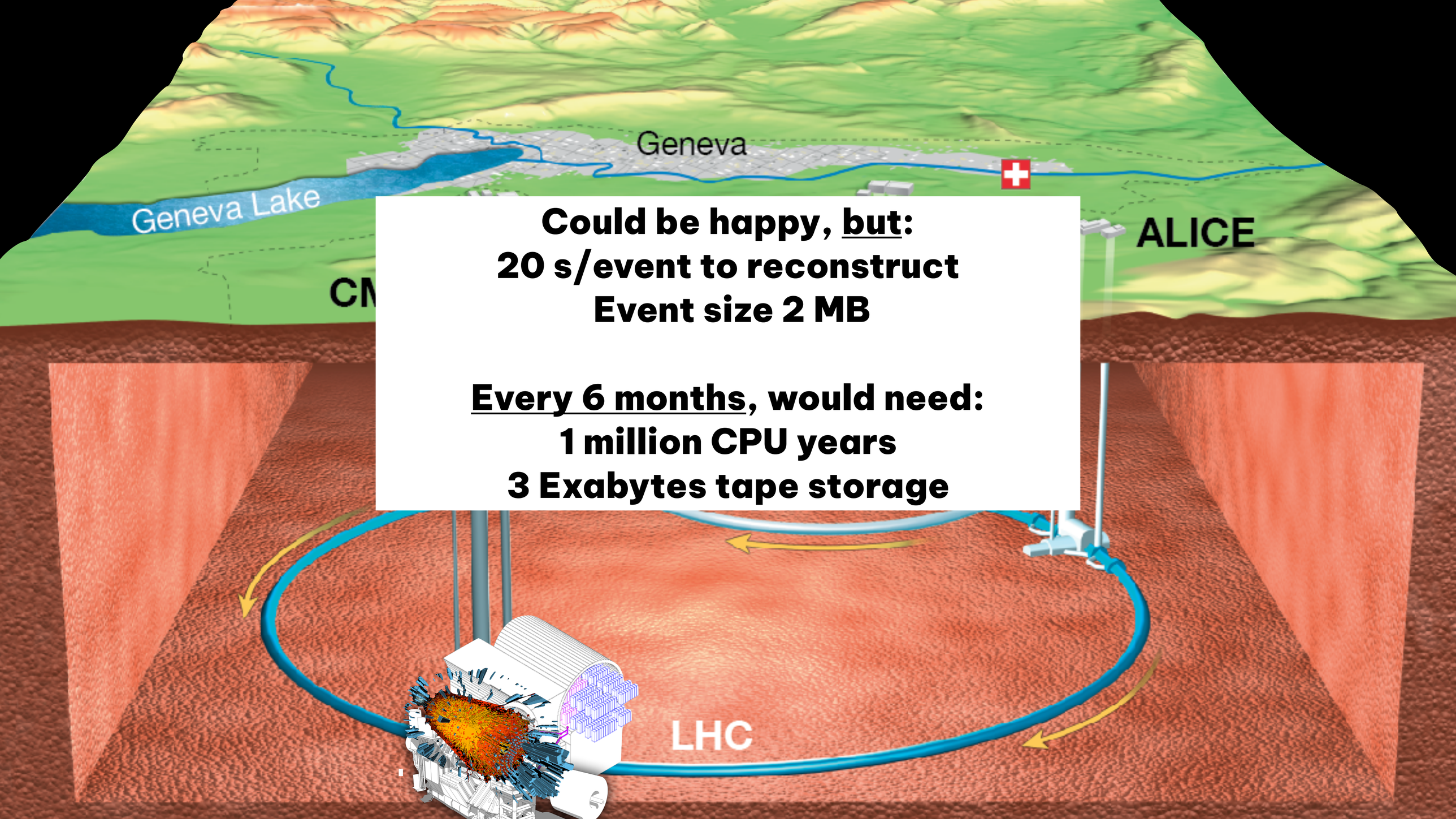

Geneva
Geneva Lake
LHCb
CMS
LHC
Reduced event information
sent out at 40 MHz
63 Tb/s


Geneva
Geneva Lake
LHCb
CMS
LHC
~500 high-end FPGAs reconstruct events and discard >99% within µs buffering time


Geneva
Geneva Lake
LHCb
CMS
L1 bit:
Accept = 1
Reject = 0
LHC

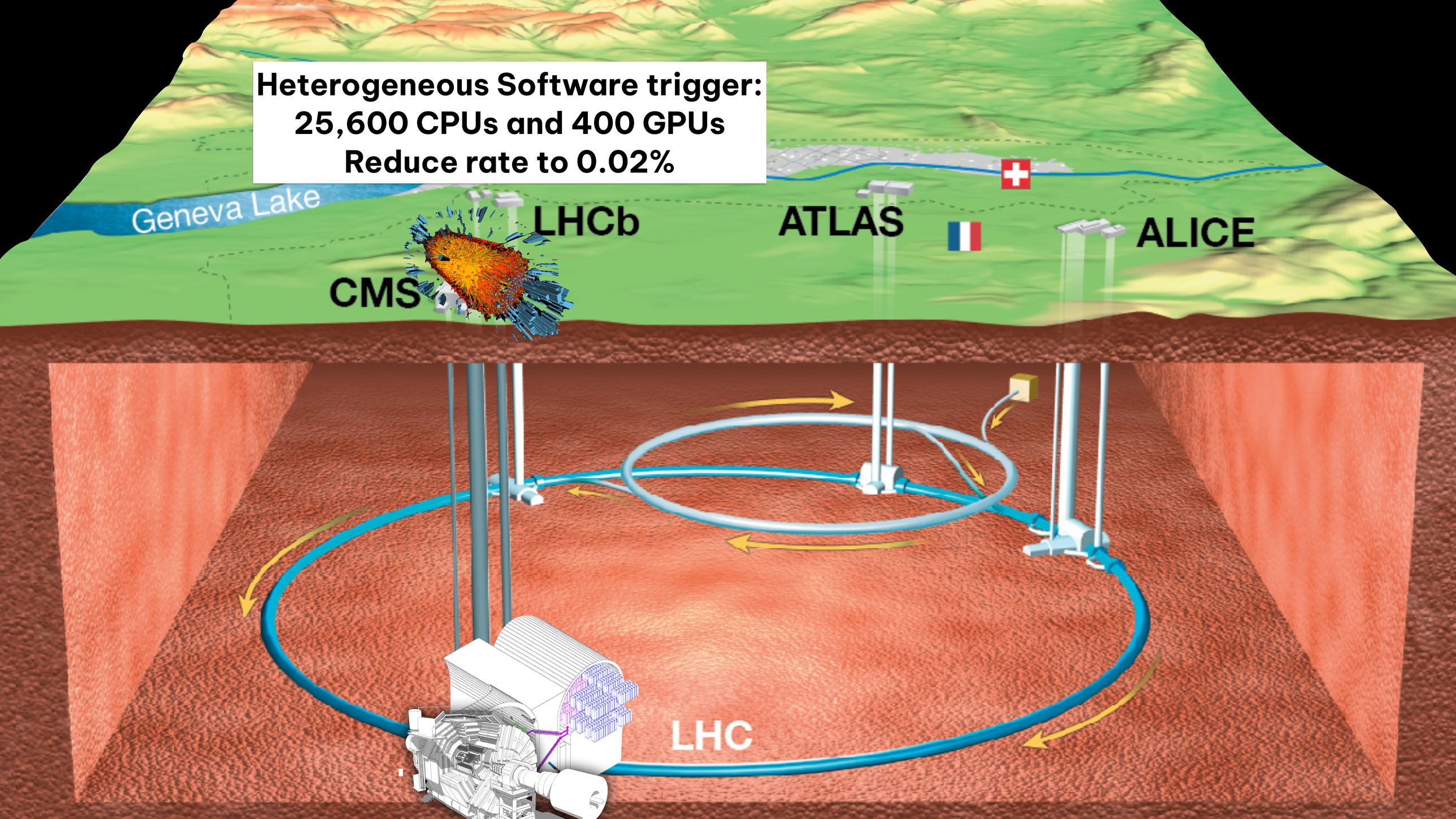

Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
x560 100 Gbps links
200m long fibers
0.2% of events left
750 kHz, O(1)TB/s
LHC

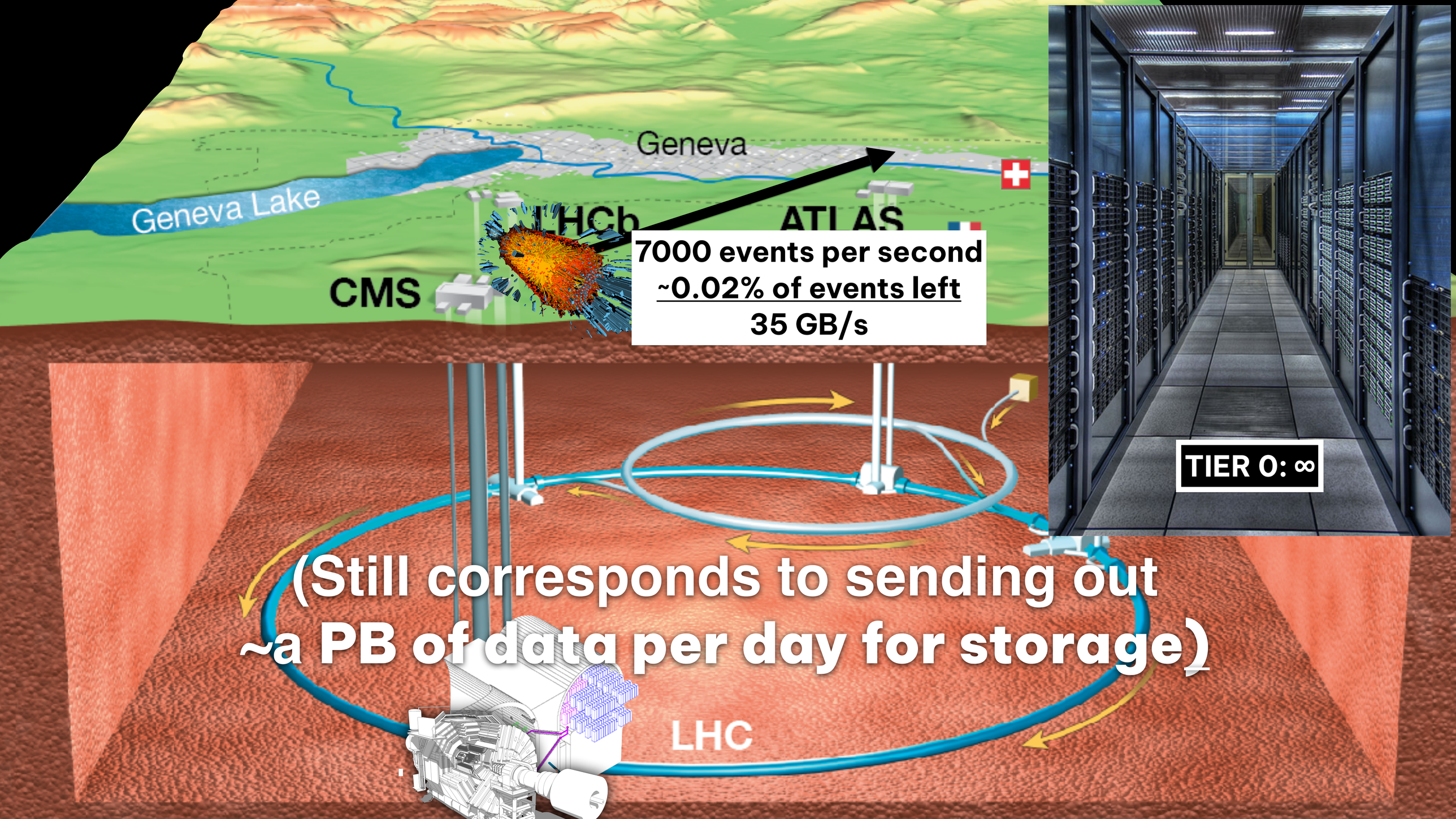

Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
x560 100 Gbps links
200m long fibers
0.2% of events left
750 kHz, O(1)TB/s
LHC

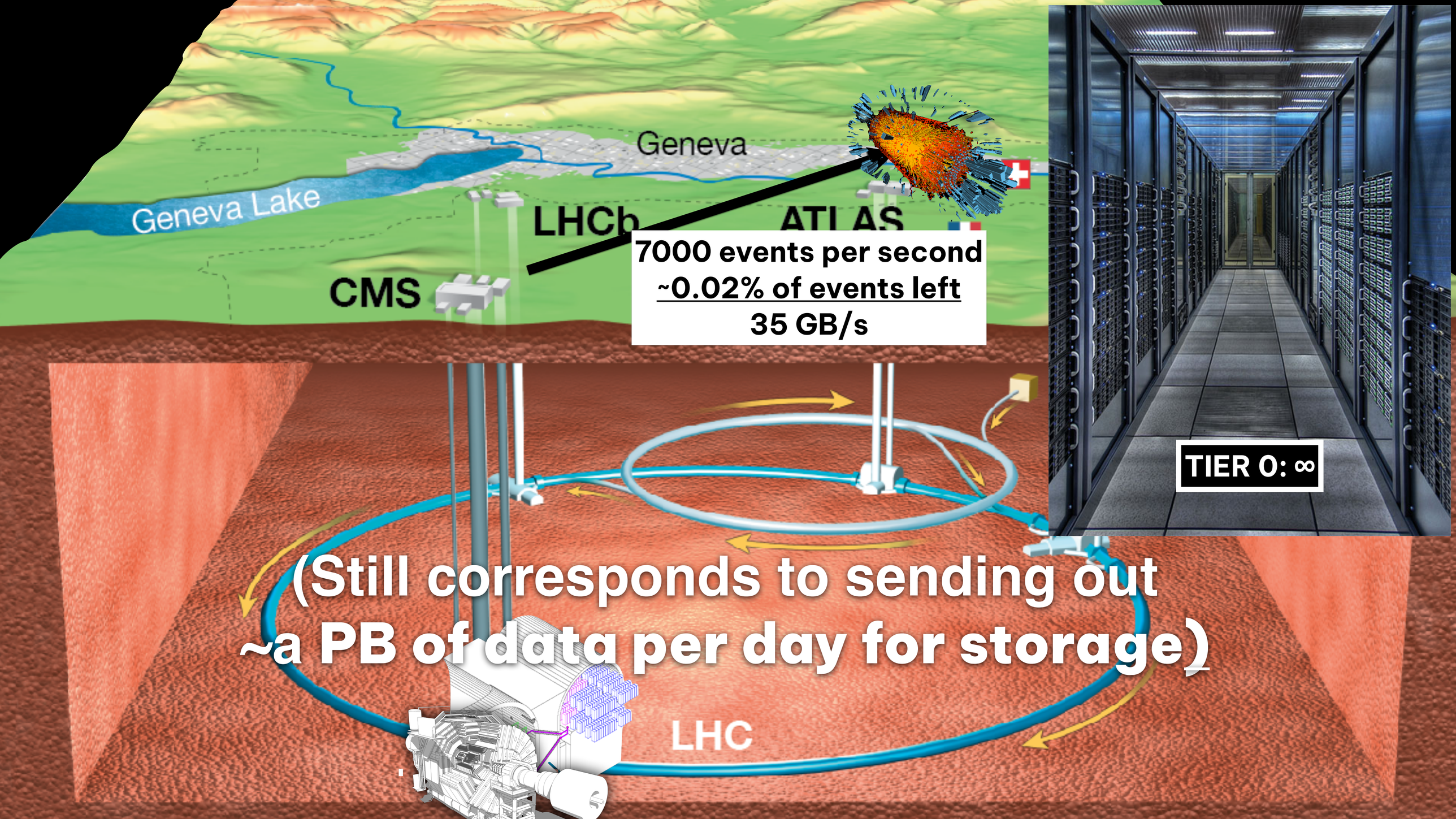


Could be happy, <u>but</u>:
20 s/event to reconstruct
Event size 2 MB

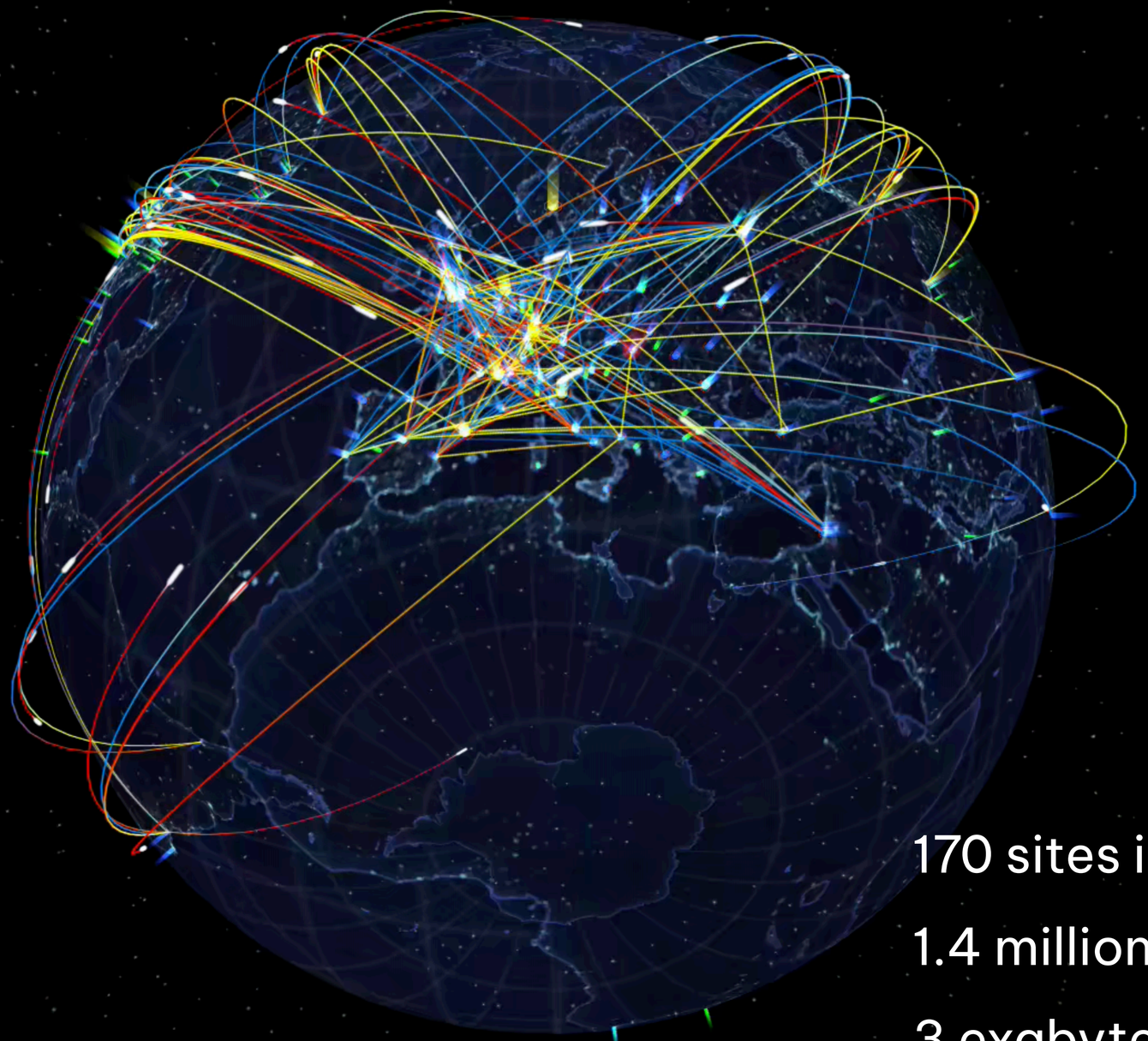
<u>Every 6 months</u>, would need:
1 million CPU years
3 Exabytes tape storage


Heterogeneous Software trigger:
25,600 CPUs and 400 GPUs
Reduce rate to 0.02%
Geneva Lake
LHCb
ATLAS
ALICE
CMS
LHC


Geneva
Geneva Lake
HCb
ATLAS
CMS
7000 events per second
~0.02% of events left
35 GB/s
TIER 0: ∞
(Still corresponds to sending out
~a PB of data per day for storage)
LHC


Geneva
Geneva Lake
LHCb
ATLAS
CMS
7000 events per second
~0.02% of events left
35 GB/s
TIER 0: ∞
(Still corresponds to sending out
~a PB of data per day for storage)
LHC

The Worldwide LHC Computing Grid



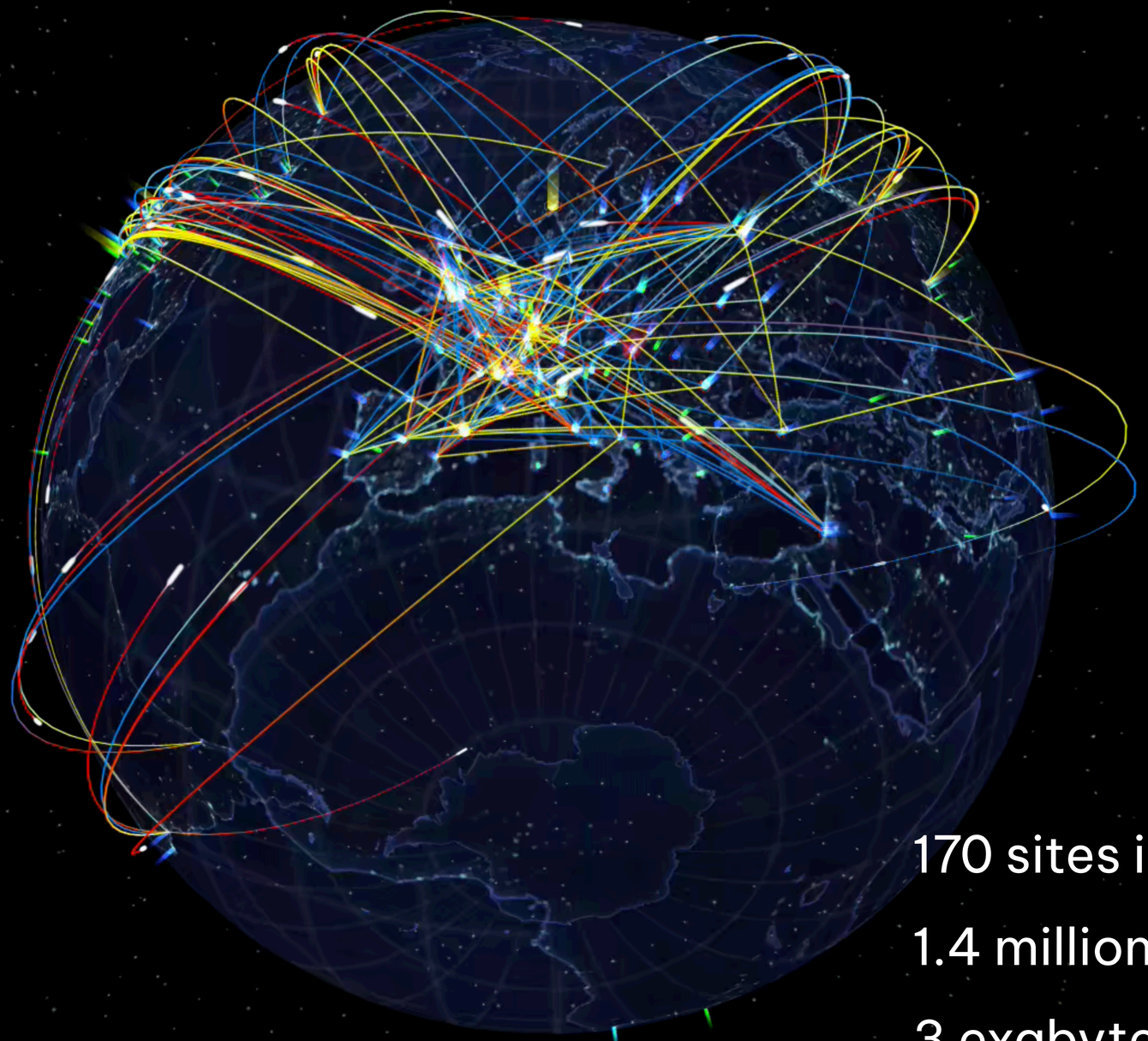
170 sites in 42 countries

1.4 million computer cores

3 exabytes of storage

Running jobs: 365644
Active CPU cores: 807139
Transfer rate: 21.54 GiB/sec

The Worldwide LHC Computing Grid



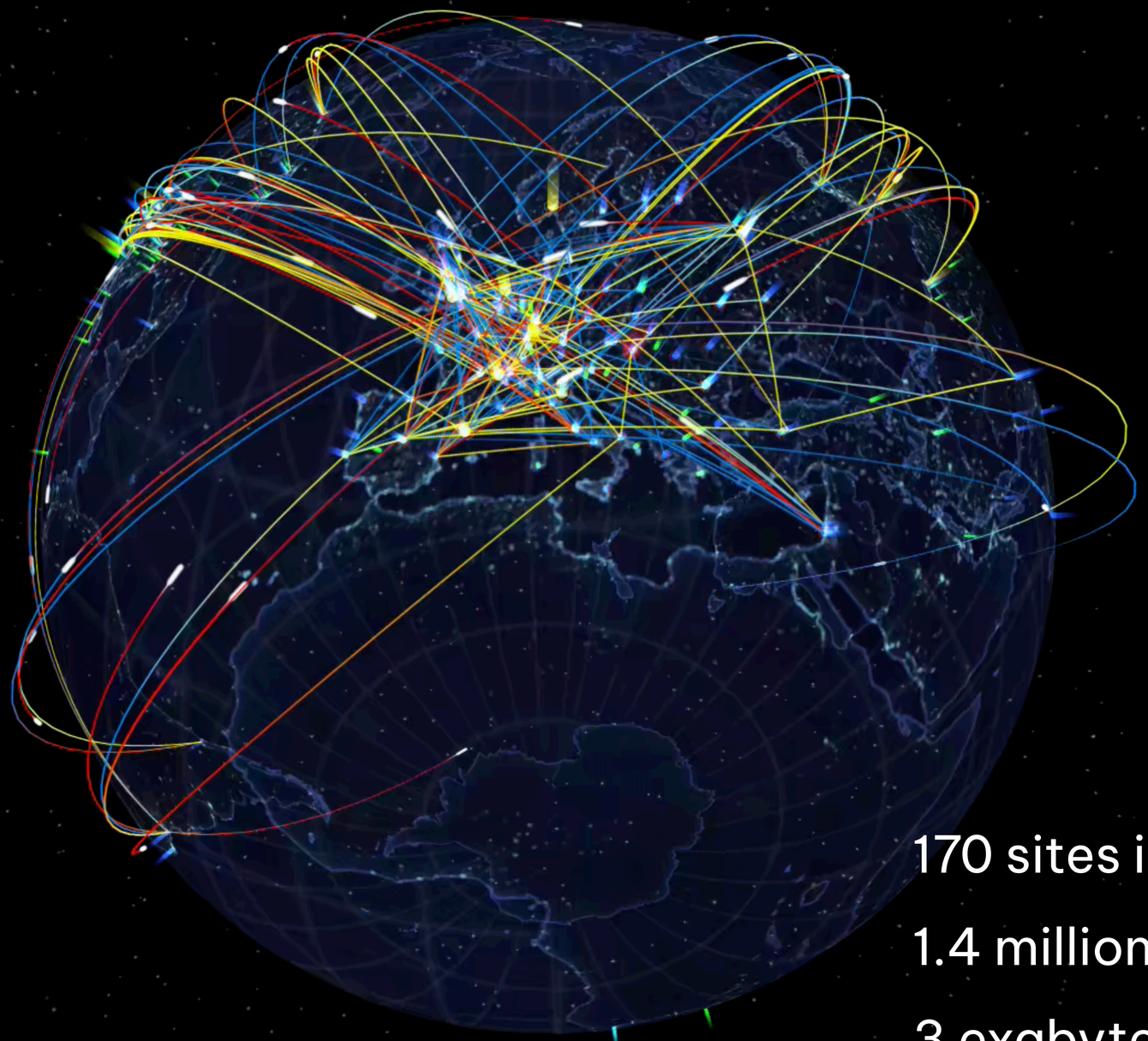
170 sites in 42 countries

1.4 million computer cores

3 exabytes of storage

Running jobs: 365644
Active CPU cores: 807139
Transfer rate: 21.54 GiB/sec

The Worldwide LHC Computing Grid



170 sites in 42 countries

1.4 million computer cores

3 exabytes of storage

Running jobs: 365644
Active CPU cores: 807139
Transfer rate: 21.54 GiB/sec

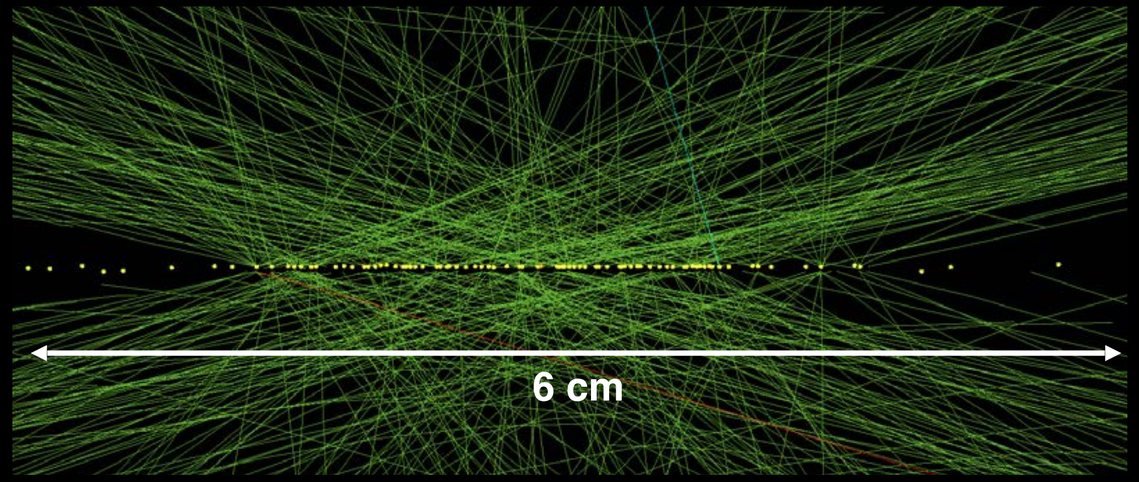
The Standard Model

$$-\tfrac{1}{2}\partial_\nu g^a_\mu \partial_\nu g^a_\mu - g_s f^{abc}\partial_\mu g^a_\nu g^b_\mu g^c_\nu - \tfrac{1}{4}g_s^2 f^{abc}f^{ade}g^b_\mu g^c_\nu g^d_\mu g^e_\nu + \tfrac{1}{2}ig_s^2(\bar{q}^\sigma_i\gamma^\mu q^\sigma_j)g^a_\mu + \bar{G}^a\partial^2 G^a + g_s f^{abc}\partial_\mu \bar{G}^a G^b g^c_\mu - \partial_\nu W^+_\mu \partial_\nu W^-_\mu - M^2 W^+_\mu W^-_\mu - \tfrac{1}{2}\partial_\nu Z^0_\mu \partial_\nu Z^0_\mu - \tfrac{1}{2c_w^2}M^2 Z^0_\mu Z^0_\mu - \tfrac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - \tfrac{1}{2}\partial_\mu H \partial_\mu H - \tfrac{1}{2}m_h^2 H^2 - \partial_\mu \phi^+ \partial_\mu \phi^- - M^2\phi^+\phi^- - \tfrac{1}{2}\partial_\mu \phi^0 \partial_\mu \phi^0 - \tfrac{1}{2c_w^2}M\phi^0\phi^0 - \beta_h[\tfrac{2M^2}{g^2} + \tfrac{2M}{g}H + \tfrac{1}{2}(H^2+\phi^0\phi^0+2\phi^+\phi^-)] + \tfrac{2M^4}{g^2}\alpha_h - igc_w[\partial_\nu Z^0_\mu(W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - Z^0_\nu(W^+_\mu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu) + Z^0_\mu(W^+_\nu \partial_\nu W^-_\mu - W^-_\nu \partial_\nu W^+_\mu)] - igs_w[\partial_\nu A_\mu(W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - A_\nu(W^+_\mu \partial_\nu W^-_\mu - W^-_\mu \partial_\nu W^+_\mu) + A_\mu(W^+_\nu \partial_\nu W^-_\mu - W^-_\nu \partial_\nu W^+_\mu)] - \tfrac{1}{2}g^2 W^+_\mu W^-_\mu W^+_\nu W^-_\nu + \tfrac{1}{2}g^2 W^+_\mu W^-_\nu W^+_\mu W^-_\nu + g^2 c_w^2(Z^0_\mu W^+_\mu Z^0_\nu W^-_\nu - Z^0_\mu Z^0_\mu W^+_\nu W^-_\nu) + g^2 s_w^2(A_\mu W^+_\mu A_\nu W^-_\nu - A_\mu A_\mu W^+_\nu W^-_\nu) + g^2 s_w c_w[A_\mu Z^0_\nu(W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) - 2A_\mu Z^0_\mu W^+_\nu W^-_\nu] - g\alpha[H^3 + H\phi^0\phi^0 + 2H\phi^+\phi^-] - \tfrac{1}{8}g^2\alpha_h[H^4+(\phi^0)^4+4(\phi^+\phi^-)^2+4(\phi^0)^2\phi^+\phi^-+4H^2\phi^+\phi^-+2(\phi^0)^2H^2] - gMW^+_\mu W^-_\mu H - \tfrac{1}{2}g\tfrac{M}{c_w^2}Z^0_\mu Z^0_\mu H - \tfrac{1}{2}ig[W^+_\mu(\phi^0\partial_\mu\phi^- - \phi^-\partial_\mu\phi^0) - W^-_\mu(\phi^0\partial_\mu\phi^+ - \phi^+\partial_\mu\phi^0)] + \tfrac{1}{2}g[W^+_\mu(H\partial_\mu\phi^- - \phi^-\partial_\mu H) - W^-_\mu(H\partial_\mu\phi^+ - \phi^+\partial_\mu H)] + \tfrac{1}{2}g\tfrac{1}{c_w}(Z^0_\mu(H\partial_\mu\phi^0 - \phi^0\partial_\mu H) - ig\tfrac{s_w^2}{c_w}MZ^0_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) + igs_wMA_\mu(W^+_\mu\phi^- - W^-_\mu\phi^+) - ig\tfrac{1-2c_w^2}{2c_w}Z^0_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) + igs_wA_\mu(\phi^+\partial_\mu\phi^- - \phi^-\partial_\mu\phi^+) - \tfrac{1}{4}g^2W^+_\mu W^-_\mu[H^2+(\phi^0)^2+2\phi^+\phi^-] - \tfrac{1}{4}g^2\tfrac{1}{c_w^2}Z^0_\mu Z^0_\mu[H^2+(\phi^0)^2+2(2s_w^2-1)^2\phi^+\phi^-] - \tfrac{1}{2}g^2\tfrac{s_w^2}{c_w}Z^0_\mu\phi^0(W^+_\mu\phi^- + W^-_\mu\phi^+) - \tfrac{1}{2}ig^2\tfrac{s_w^2}{c_w}Z^0_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) + \tfrac{1}{2}g^2s_wA_\mu\phi^0(W^+_\mu\phi^- + W^-_\mu\phi^+) + \tfrac{1}{2}ig^2s_wA_\mu H(W^+_\mu\phi^- - W^-_\mu\phi^+) - g^2\tfrac{s_w}{c_w}(2c_w^2-1)Z^0_\mu A_\mu\phi^+\phi^- - g^1s_w^2A_\mu A_\mu\phi^+\phi^- - \bar{e}^\lambda(\gamma\partial + m^\lambda_e)e^\lambda - \bar{\nu}^\lambda\gamma\partial\nu^\lambda - \bar{u}^\lambda_j(\gamma\partial + m^\lambda_u)u^\lambda_j - \bar{d}^\lambda_j(\gamma\partial + m^\lambda_d)d^\lambda_j + igs_wA_\mu[-(\bar{e}^\lambda\gamma^\mu e^\lambda) + \tfrac{2}{3}(\bar{u}^\lambda_j\gamma^\mu u^\lambda_j) - \tfrac{1}{3}(\bar{d}^\lambda_j\gamma^\mu d^\lambda_j)] + \tfrac{ig}{4c_w}Z^0_\mu[(\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{e}^\lambda\gamma^\mu(4s_w^2-1-\gamma^5)e^\lambda) + (\bar{u}^\lambda_j\gamma^\mu(\tfrac{4}{3}s_w^2 - 1-\gamma^5)u^\lambda_j) + (\bar{d}^\lambda_j\gamma^\mu(1-\tfrac{8}{3}s_w^2-\gamma^5)d^\lambda_j)] + \tfrac{ig}{2\sqrt{2}}W^+_\mu[(\bar{\nu}^\lambda\gamma^\mu(1+\gamma^5)e^\lambda) + (\bar{u}^\lambda_j\gamma^\mu(1+\gamma^5)C_{\lambda\kappa}d^\kappa_j)] + \tfrac{ig}{2\sqrt{2}}W^-_\mu[(\bar{e}^\lambda\gamma^\mu(1+\gamma^5)\nu^\lambda) + (\bar{d}^\kappa_jC^\dagger_{\lambda\kappa}\gamma^\mu(1+\gamma^5)u^\lambda_j)] + \tfrac{ig}{2\sqrt{2}}\tfrac{m^\lambda_e}{M}[-\phi^+(\bar{\nu}^\lambda(1-\gamma^5)e^\lambda) + \phi^-(\bar{e}^\lambda(1+\gamma^5)\nu^\lambda)] - \tfrac{g}{2}\tfrac{m^\lambda_e}{M}[H(\bar{e}^\lambda e^\lambda) + i\phi^0(\bar{e}^\lambda\gamma^5 e^\lambda)] + \tfrac{ig}{2M\sqrt{2}}\phi^+[-m^\kappa_d(\bar{u}^\lambda_jC_{\lambda\kappa}(1-\gamma^5)d^\kappa_j) + m^\lambda_u(\bar{u}^\lambda_jC_{\lambda\kappa}(1+\gamma^5)d^\kappa_j] + \tfrac{ig}{2M\sqrt{2}}\phi^-[m^\lambda_d(\bar{d}^\lambda_jC^\dagger_{\lambda\kappa}(1+\gamma^5)u^\kappa_j) - m^\kappa_u(\bar{d}^\lambda_jC^\dagger_{\lambda\kappa}(1-\gamma^5)u^\kappa_j] - \tfrac{g}{2}\tfrac{m^\lambda_u}{M}H(\bar{u}^\lambda_ju^\lambda_j) - \tfrac{g}{2}\tfrac{m^\lambda_d}{M}H(\bar{d}^\lambda_jd^\lambda_j) + \tfrac{ig}{2}\tfrac{m^\lambda_u}{M}\phi^0(\bar{u}^\lambda_j\gamma^5u^\lambda_j) - \tfrac{ig}{2}\tfrac{m^\lambda_d}{M}\phi^0(\bar{d}^\lambda_j\gamma^5d^\lambda_j) + \bar{X}^+(\partial^2-M^2)X^+ + \bar{X}^-(\partial^2-M^2)X^- + \bar{X}^0(\partial^2 - \tfrac{M^2}{c_w^2})X^0 + \bar{Y}\partial^2Y + igc_wW^+_\mu(\partial_\mu\bar{X}^0X^- - \partial_\mu\bar{X}^+X^0) + igs_wW^+_\mu(\partial_\mu\bar{Y}X^- - \partial_\mu\bar{X}^+Y) + igc_wW^-_\mu(\partial_\mu\bar{X}^-X^0 - \partial_\mu\bar{X}^0X^+) + igs_wW^-_\mu(\partial_\mu\bar{X}^-Y - \partial_\mu\bar{Y}X^+) + igc_wZ^0_\mu(\partial_\mu\bar{X}^+X^+ - \partial_\mu\bar{X}^-X^-) + igs_wA_\mu(\partial_\mu\bar{X}^+X^+ - \partial_\mu\bar{X}^-X^-) - \tfrac{1}{2}gM[\bar{X}^+X^+H + \bar{X}^-X^-H + \tfrac{1}{c_w^2}\bar{X}^0X^0H] + \tfrac{1-2c_w^2}{2c_w}igM[\bar{X}^+X^0\phi^+ - \bar{X}^-X^0\phi^-] + \tfrac{1}{2c_w}igM[\bar{X}^0X^-\phi^+ - \bar{X}^0X^+\phi^-] + igMs_w[\bar{X}^0X^-\phi^+ - \bar{X}^0X^+\phi^-] + \tfrac{1}{2}igM[\bar{X}^+X^+\phi^0 - \bar{X}^-X^-\phi^0]$$

…and measuring its 19 free parameters to 5σ (= 0.00003% chance of statistical fluctuation)…

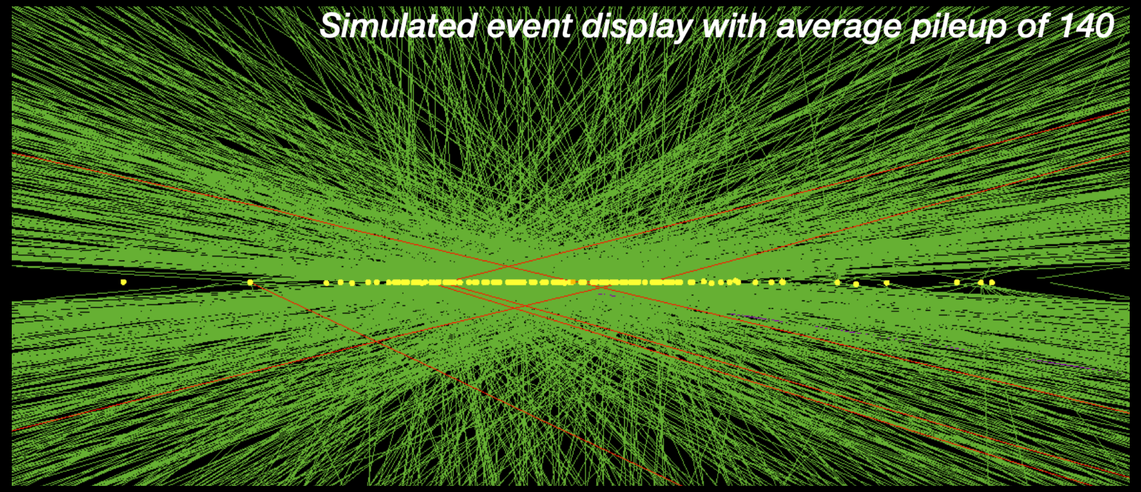
LHC (currently)

78 vertices
(average 60)

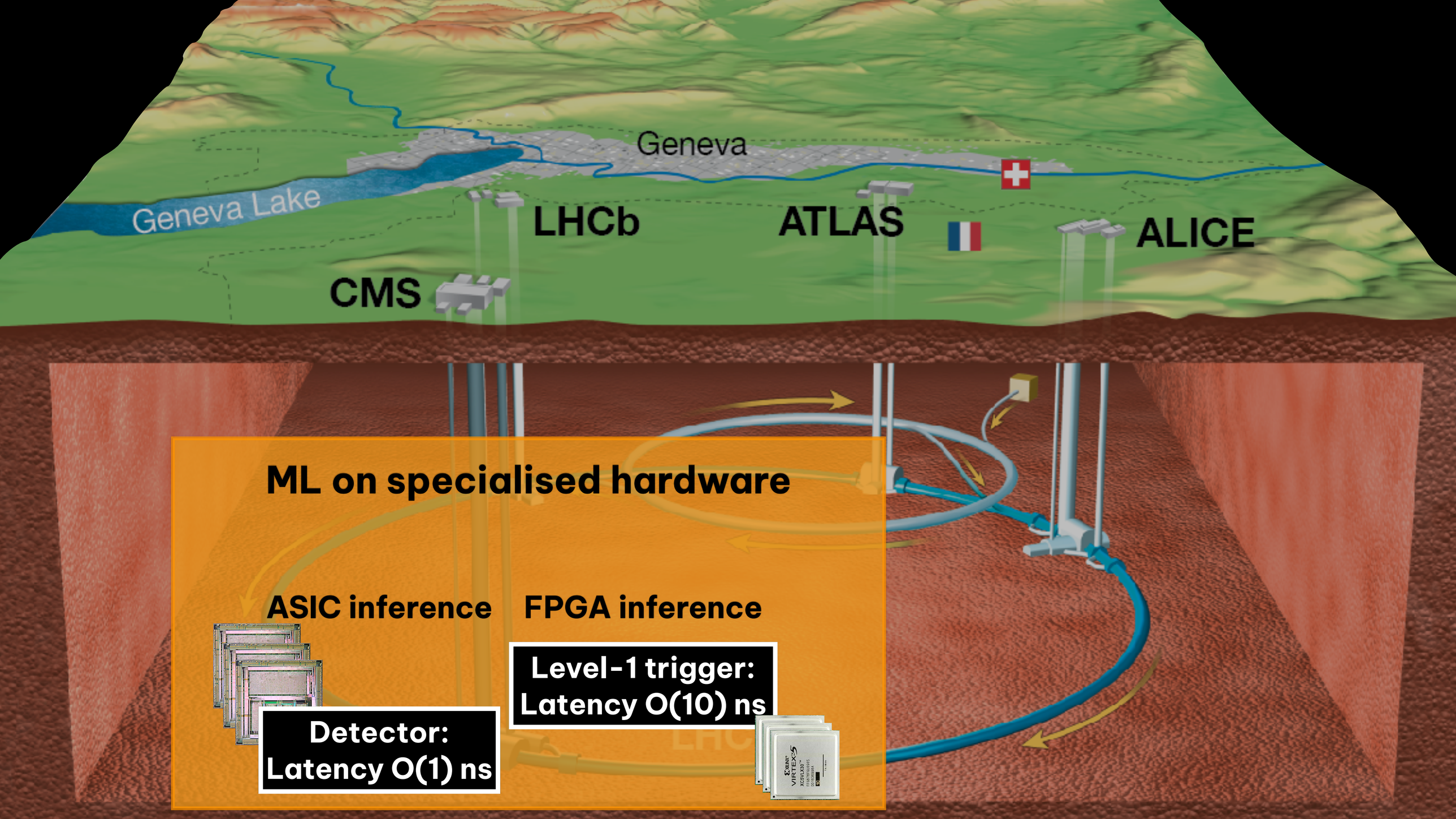


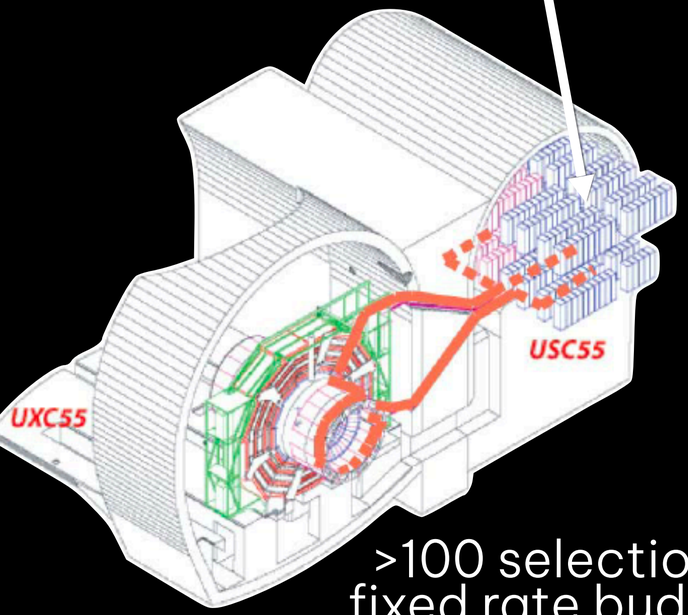
High Luminosity LHC (2030-2041)

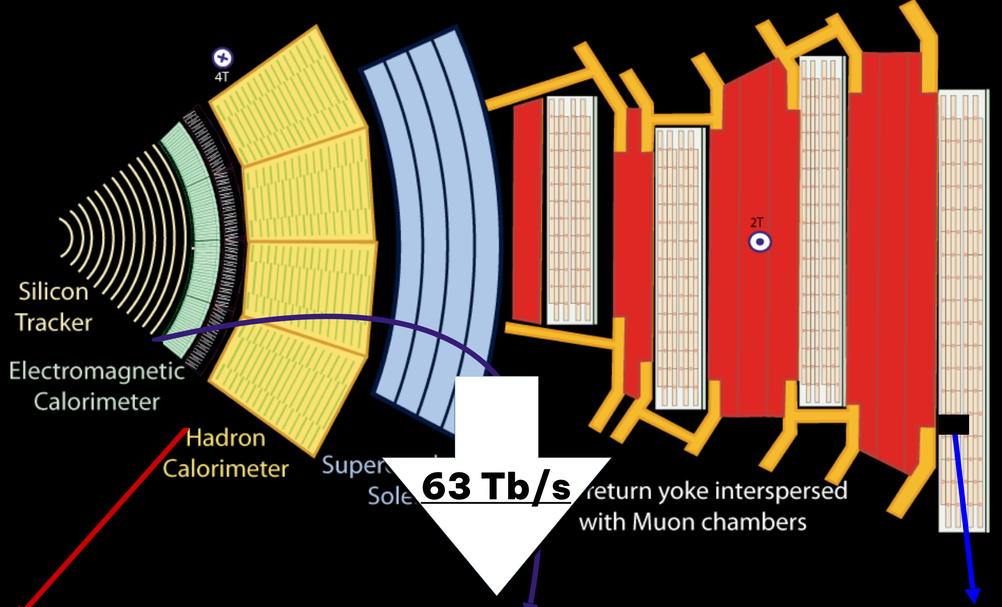
200 vertices
(average 140)



→event size: 2→8 MB,
→throughput: 4→63 Tb/s


Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
ML on specialised hardware
ASIC inference
FPGA inference
Level-1 trigger:
Latency O(10) ns
Detector:
Latency O(1) ns





CALORIMETER BACKENDS
360 FPGAs
VU13P-2

CALORIMETER TRIGGER
35 FPGAs
VU13P-2

TRACK FINDER
162 FPGAs
VU13P-2

TRACK TRIGGER
12 FPGAs
VU13P-2

MUON BACKENDS AND TRIGGER
112 FPGAs
18 Versals here

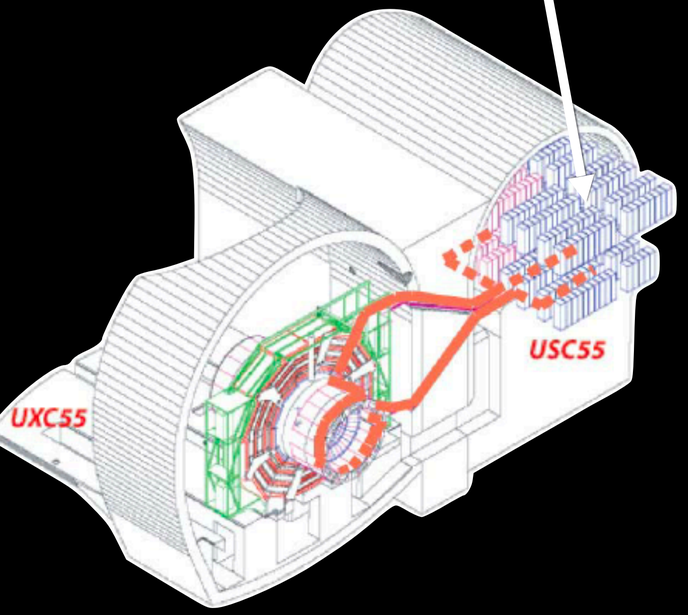
CORRELATOR TRIGGER:
66 FPGAs

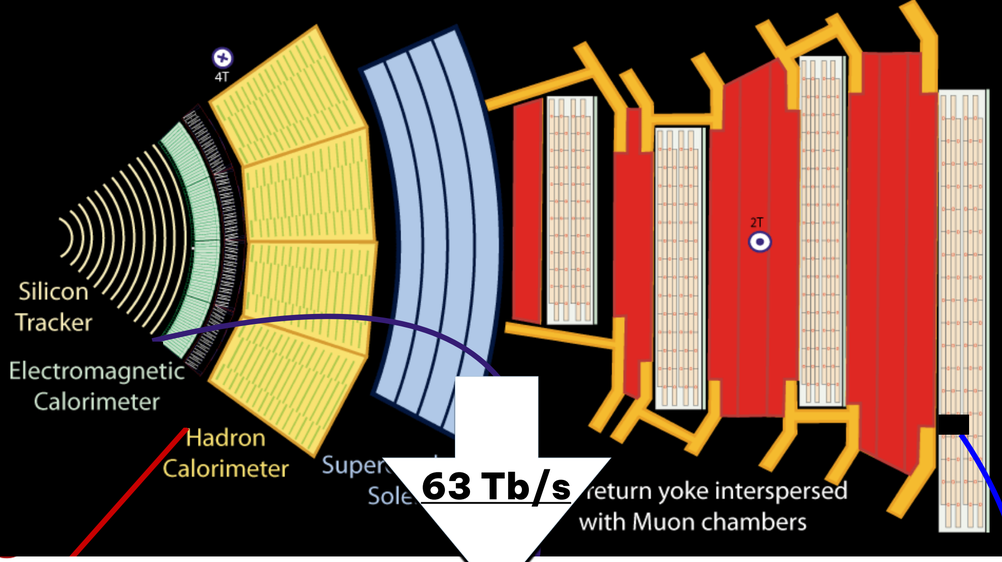
>100 selection algorithms with fixed rate budgets
(total <u>cannot</u> exceed 750 kHz)

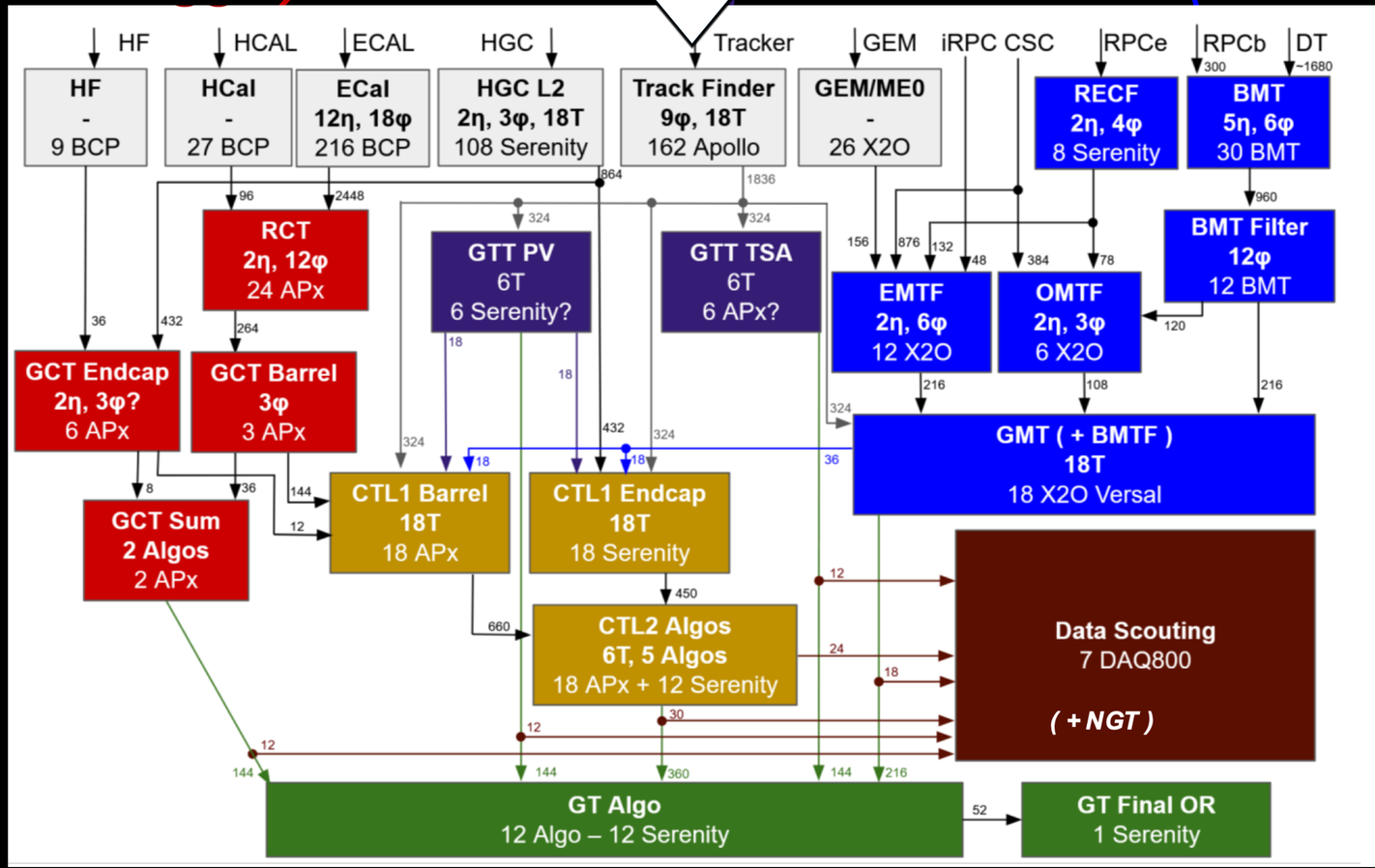
GLOBAL TRIGGER ALGOS
12 FPGAs

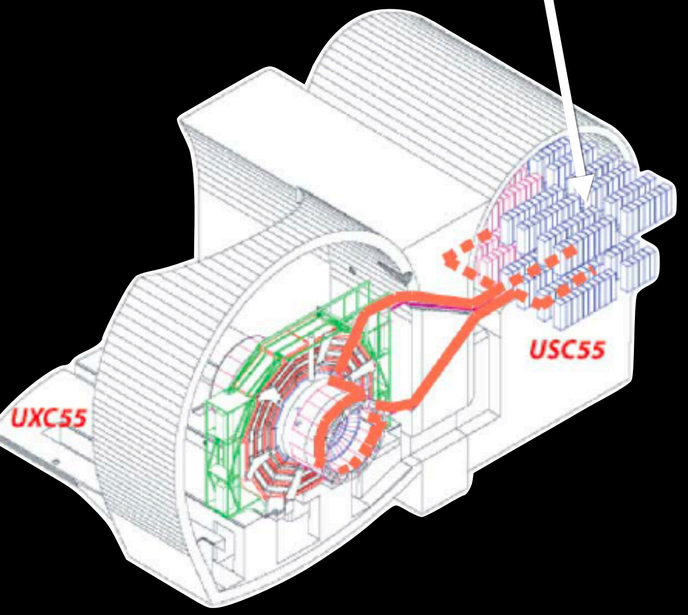
GLOBAL TRIGGER FINAL *OR*
1 FPGA

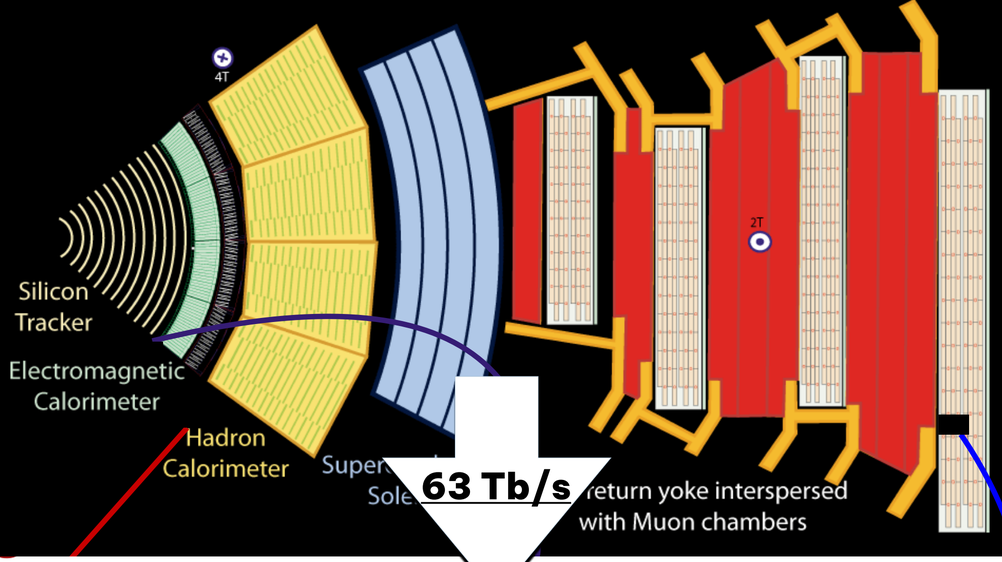
<u>12.5 µs</u>

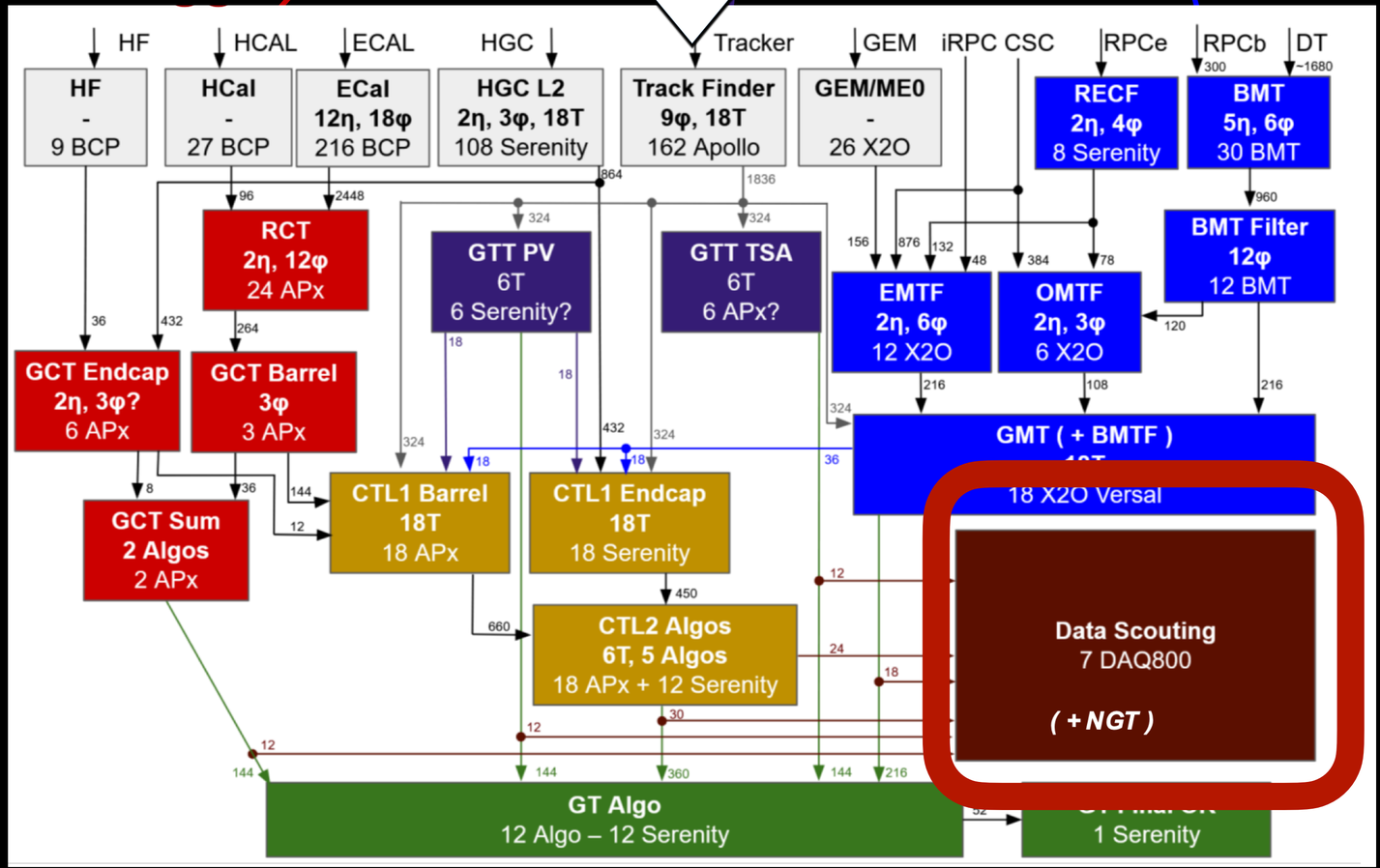

USC55
UXC55


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
63 Tb/s
return yoke interspersed with Muon chambers


HF
HCAL
ECAL
HGC
Tracker
GEM
iRPC CSC
RPCe
RPCb
DT
HF - 9 BCP
HCal - 27 BCP
ECal 12η, 18φ 216 BCP
HGC L2 2η, 3φ, 18T 108 Serenity
Track Finder 9φ, 18T 162 Apollo
GEM/ME0 - 26 X2O
RECF 2η, 4φ 8 Serenity
BMT 5η, 6φ 30 BMT
RCT 2η, 12φ 24 APx
GTT PV 6T 6 Serenity?
GTT TSA 6T 6 APx?
BMT Filter 12φ 12 BMT
EMTF 2η, 6φ 12 X2O
OMTF 2η, 3φ 6 X2O
GCT Endcap 2η, 3φ? 6 APx
GCT Barrel 3φ 3 APx
GCT Sum 2 Algos 2 APx
CTL1 Barrel 18T 18 APx
CTL1 Endcap 18T 18 Serenity
GMT (+ BMTF) 18T 18 X2O Versal
CTL2 Algos 6T, 5 Algos 18 APx + 12 Serenity
Data Scouting 7 DAQ800
(+ NGT)
GT Algo 12 Algo – 12 Serenity
GT Final OR 1 Serenity


USC55
UXC55

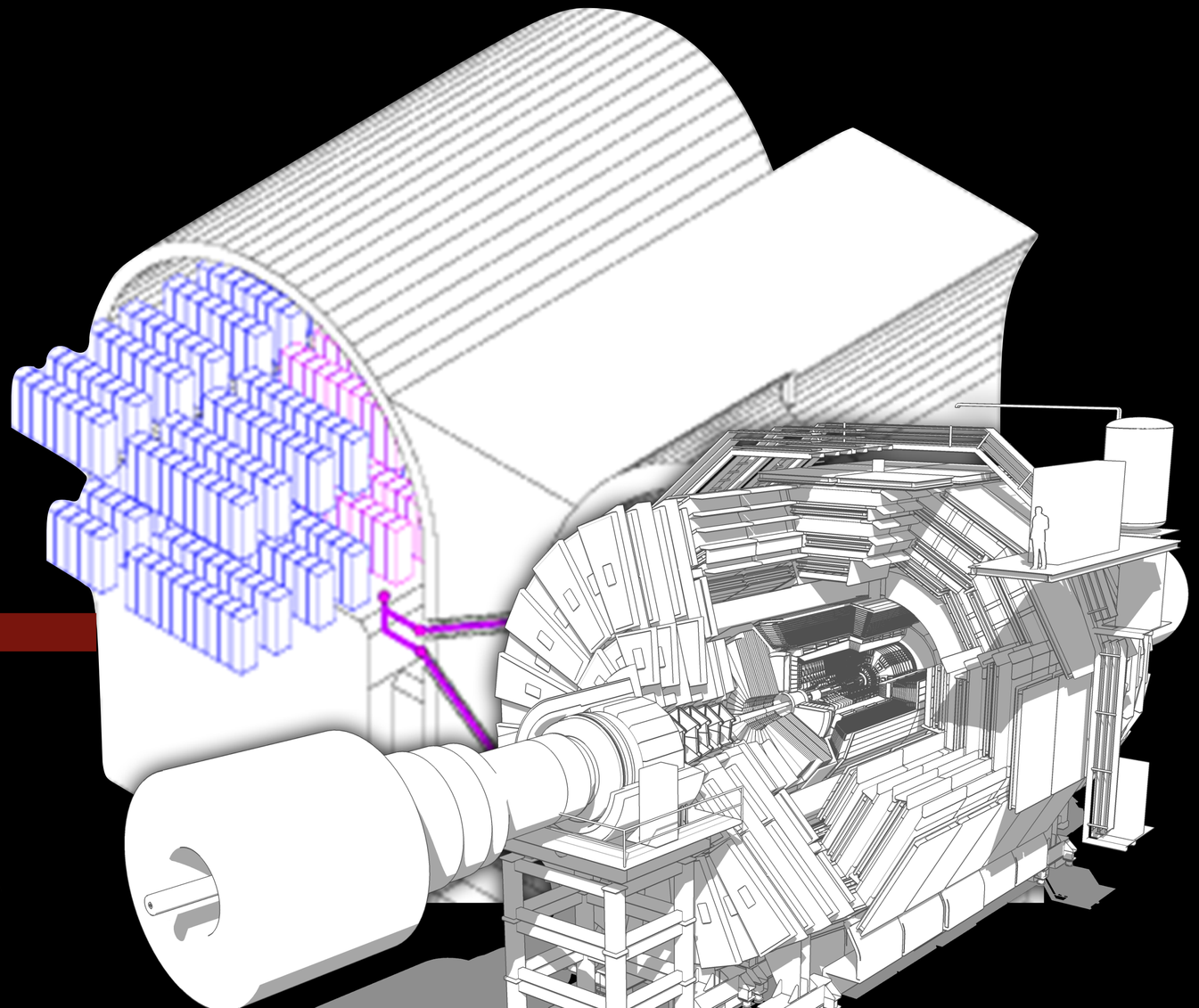

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
63 Tb/s
return yoke interspersed with Muon chambers

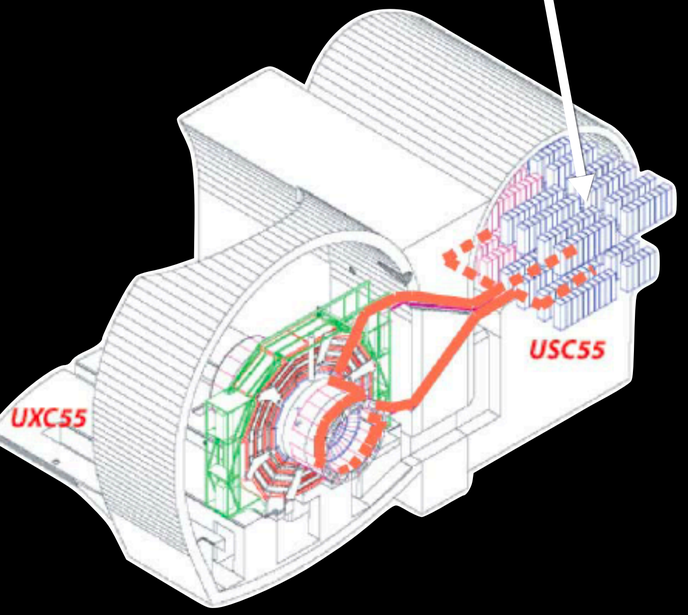

HF
HCAL
ECAL
HGC
Tracker
GEM
iRPC CSC
RPCe
RPCb
DT
HF - 9 BCP
HCal - 27 BCP
ECal 12η, 18φ 216 BCP
HGC L2 2η, 3φ, 18T 108 Serenity
Track Finder 9φ, 18T 162 Apollo
GEM/ME0 - 26 X2O
RECF 2η, 4φ 8 Serenity
BMT 5η, 6φ 30 BMT
RCT 2η, 12φ 24 APx
GTT PV 6T 6 Serenity?
GTT TSA 6T 6 APx?
BMT Filter 12φ 12 BMT
EMTF 2η, 6φ 12 X2O
OMTF 2η, 3φ 6 X2O
GCT Endcap 2η, 3φ? 6 APx
GCT Barrel 3φ 3 APx
GMT (+ BMTF)
18 X2O Versal
GCT Sum 2 Algos 2 APx
CTL1 Barrel 18T 18 APx
CTL1 Endcap 18T 18 Serenity
CTL2 Algos 6T, 5 Algos 18 APx + 12 Serenity
Data Scouting 7 DAQ800
(+ NGT)
GT Algo 12 Algo – 12 Serenity
1 Serenity

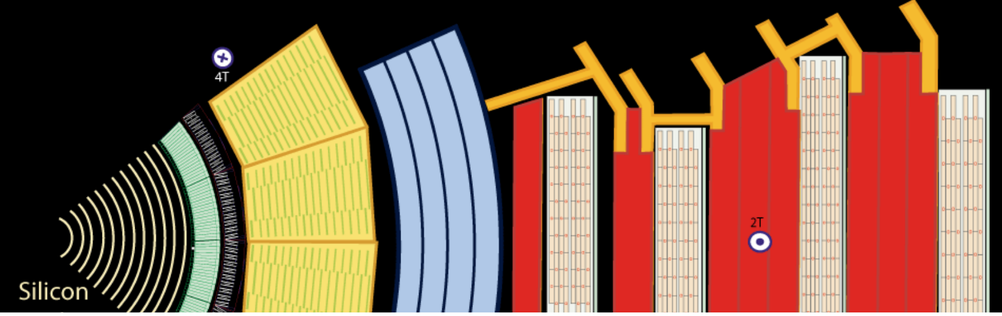
New system to read out <u>all hardware trigger data at 40 MHz</u>
Enables analysis of ***every*** collision event.

On-GPU real-time analysis in O(1) µs, reduce rate to 5 GB/s!

On-FPGA processing in <25 ns
7.65 Tb/second

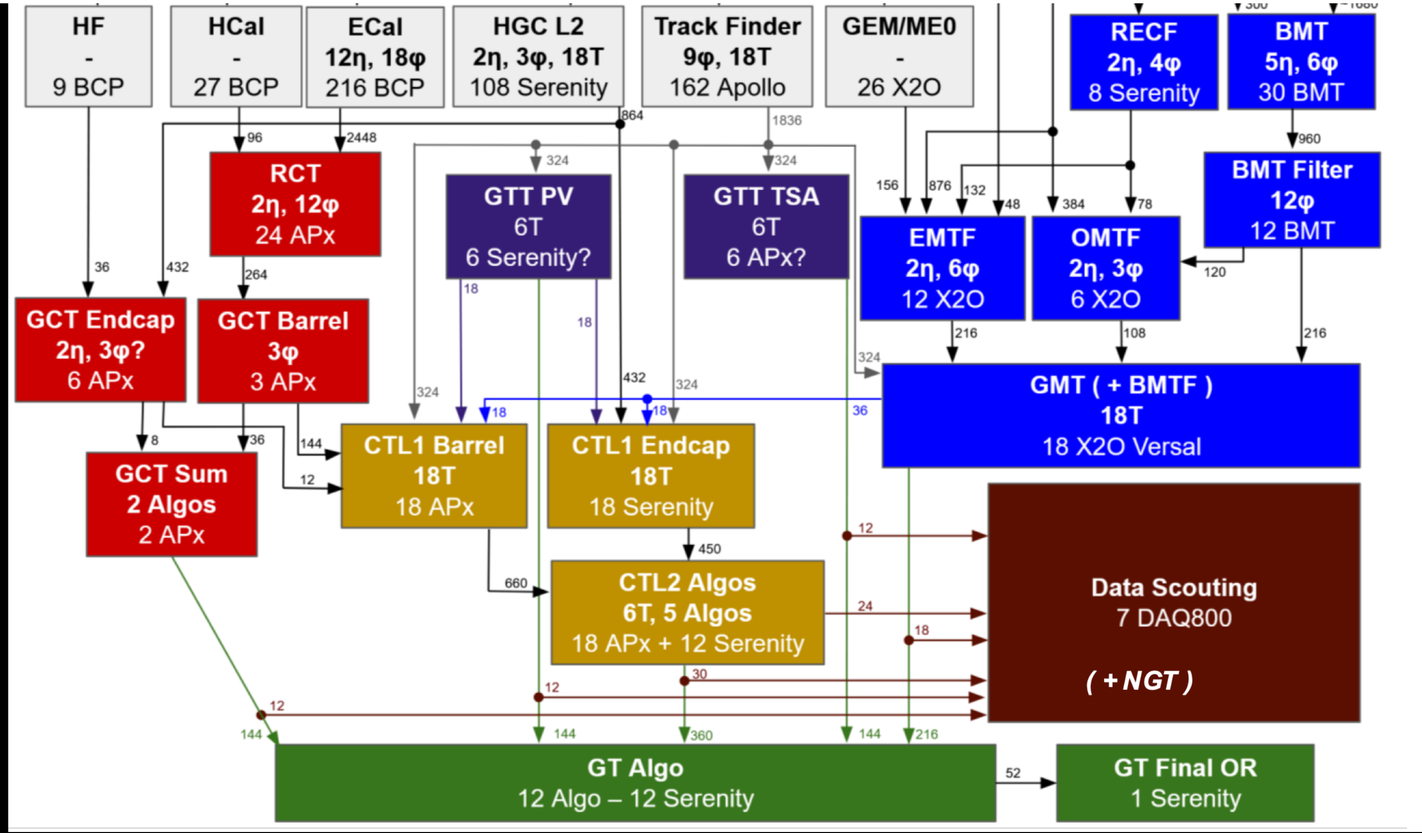




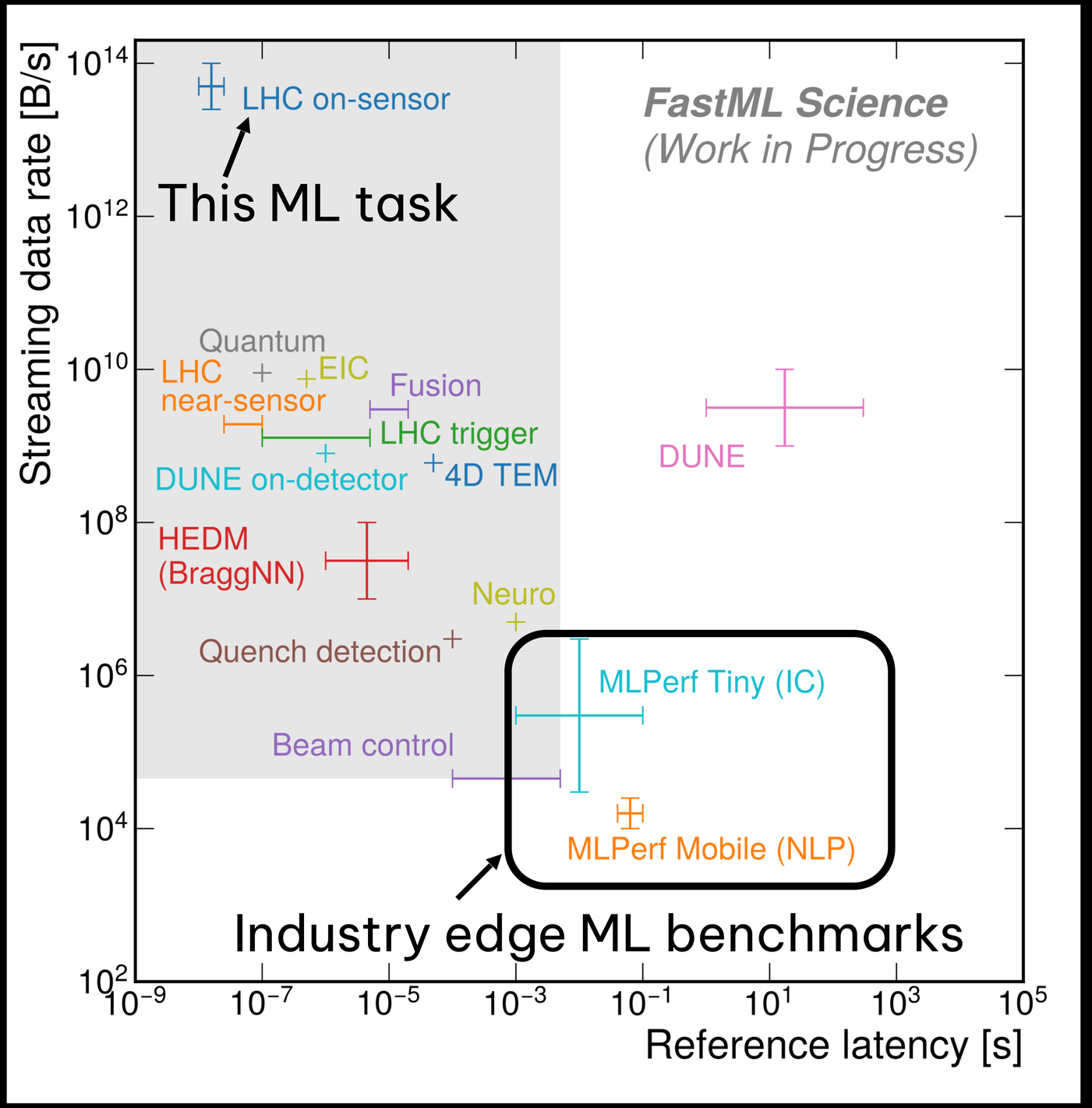


12 microseconds latency

Processing 5% of internet traffic

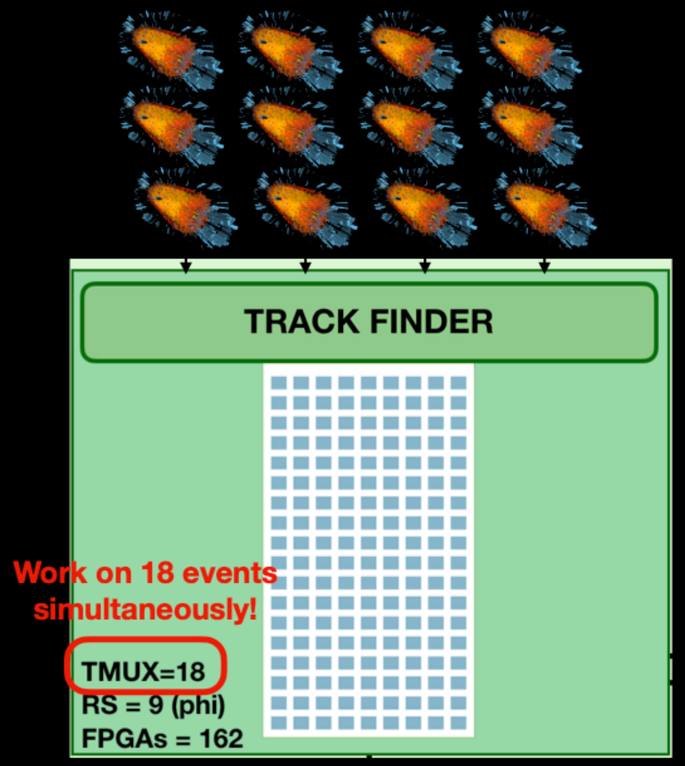


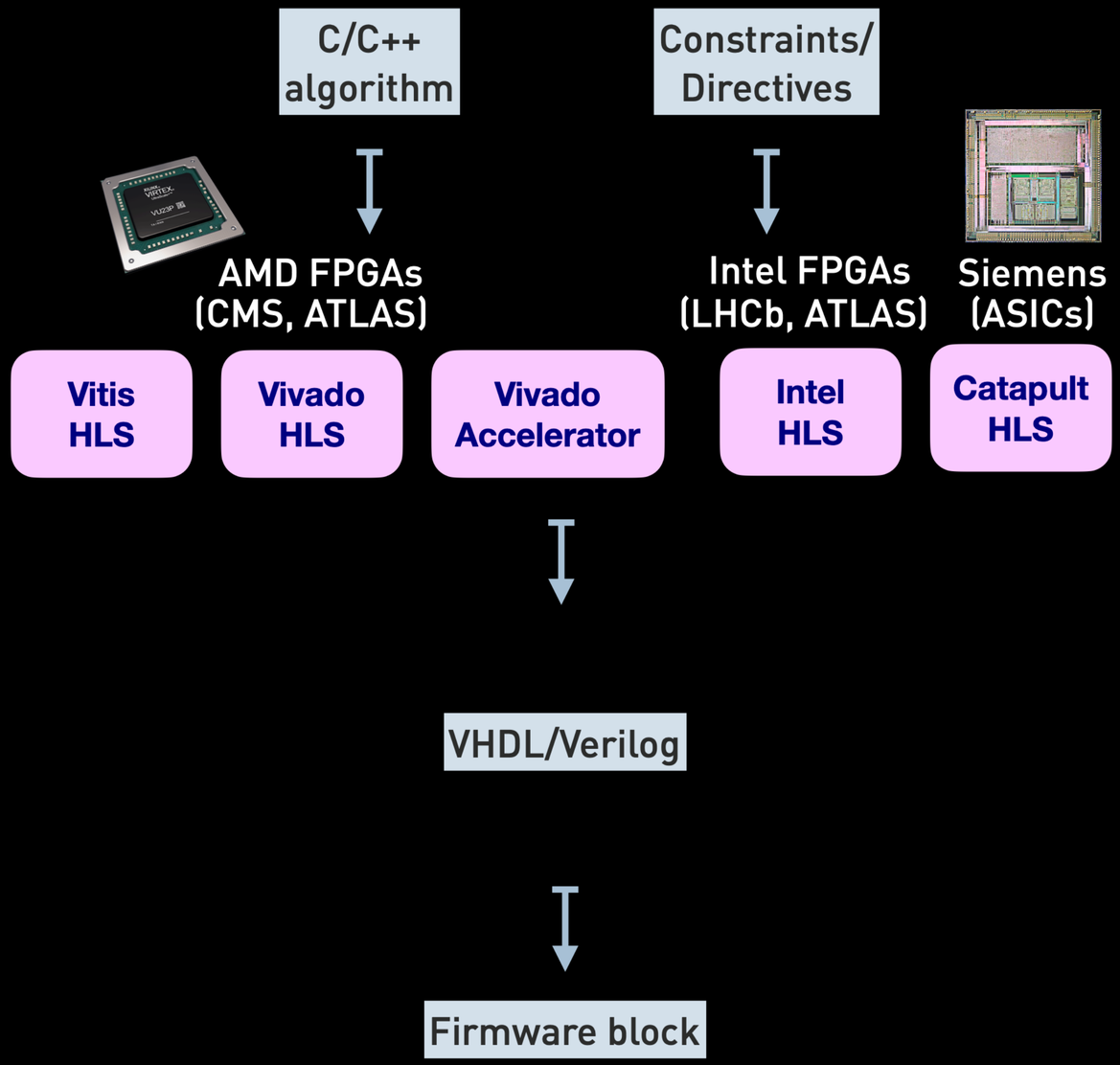
Had to design our own tools (to do inference in <100 ns)



Why FPGAs

- Resource parallelism ↔ latency
- Pipeline parallelism ↔ throughput
- Latency deterministic (40 MHz collisions)
- Specialized blocks for I/O
- High-speed transceivers with Tb/s total bandwidth (PCIe, 100Gbps ethernet, InfiniBand)
- Relatively low power per op



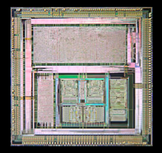

C/C++
algorithm
Constraints/
Directives
AMD FPGAs
(CMS, ATLAS)
Intel FPGAs
(LHCb, ATLAS)
Siemens
(ASICs)
Vitis
HLS
Vivado
HLS
Vivado
Accelerator
Intel
HLS
Catapult
HLS
VHDL/Verilog
Firmware block

C/C++ algorithm

Constraints/ Directives

AMD FPGAs (CMS, ATLAS)

Intel FPGAs (LHCb, ATLAS)



Siemens (ASICs)

Vitis HLS

Vivado HLS

Vivado Accelerator

Intel HLS

Catapult HLS

Hardware Description Languages:
programming languages which describe electronic circuits

VHDL/Verilog

Firmware block

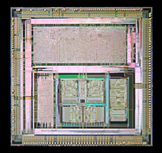
High Level Synthesis:
Compile from C/C++ to VHDL

Pre-processor directives and constraints used to optimize the design

Drastic decrease in firmware development time!

C/C++ algorithm

Constraints/ Directives



AMD FPGAs (CMS, ATLAS)

Intel FPGAs (LHCb, ATLAS)

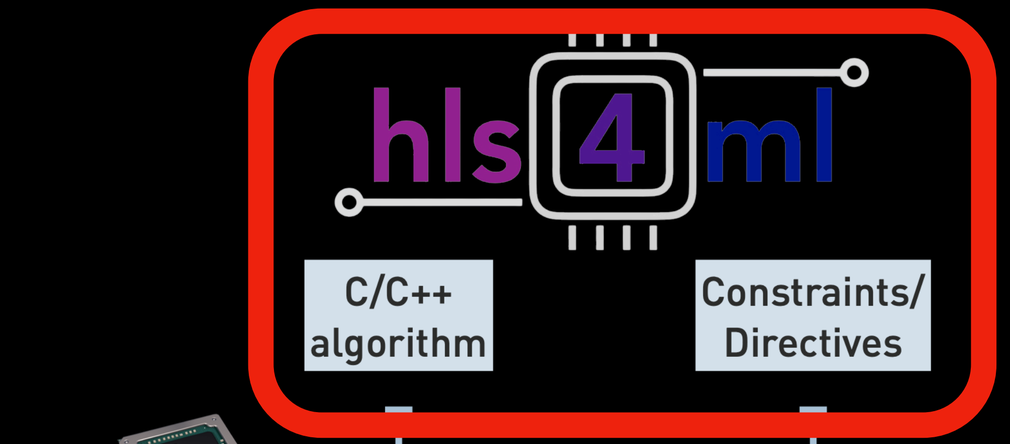
Siemens (ASICs)

Vitis HLS | Vivado HLS | Vivado Accelerator | Intel HLS | Catapult HLS

Hardware Description Languages:
programming languages which describe electronic circuits

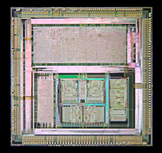
VHDL/Verilog

Firmware block



Let's tune this for nanosecond ML!

AMD FPGAs (CMS, ATLAS)



Intel FPGAs (LHCb, ATLAS)

Siemens (ASICs)

Vitis HLS | Vivado HLS | Vivado Accelerator | Intel HLS | Catapult HLS

VHDL/Verilog

Firmware block

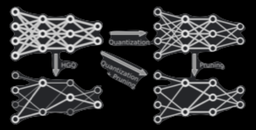
Floating point:

Model

Quantized:

Brevitas

QONNX



HQG

Floating point:

Keras + TensorFlow

ONNX

PYTORCH

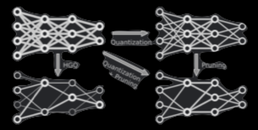
Model

Quantized:

QKeras

Brevitas

QONNX



HQG

hls4ml

HLS Model

Write HLS project
Emulate
Run synthesis

Quartus Prime

Vivado™ HLS

XILINX VITIS™

oneAPI

Catapult

Floating point:

K + TensorFlow

ONNX

PYTORCH

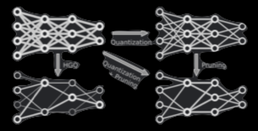
Model

Quantized:

QKeras

Brevitas

QONNX



HQG

hls4ml

HLS Model

Write HLS project
Emulate
Run synthesis

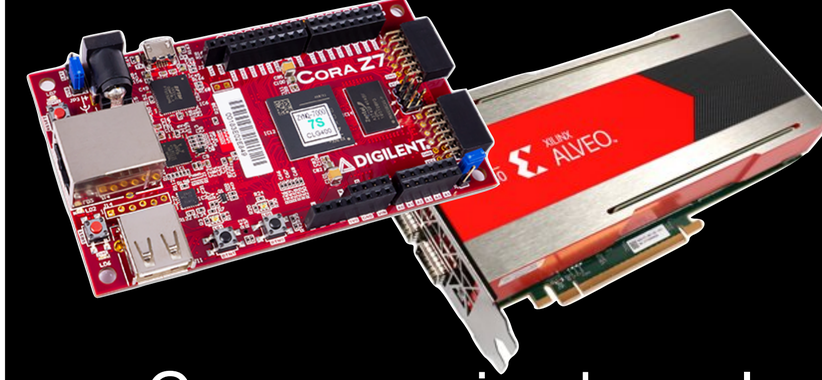
Quartus Prime

Vivado™ HLS

XILINX VITIS™

oneAPI

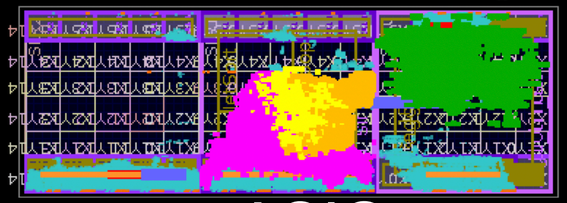
Catapult



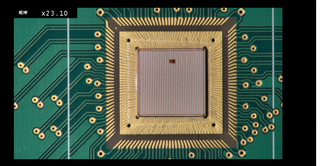
Co-processing kernel
(Accelerators/SoCs)

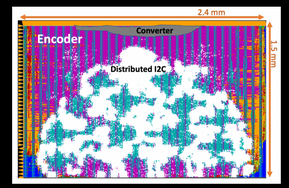
Hardware Model

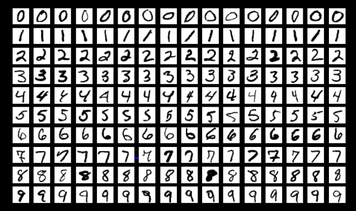
FPGA custom designs



ASICs







Conv2D

ReLU

MaxPool2D

Conv2D

ReLU

MaxPool2D

Flatten

Dense

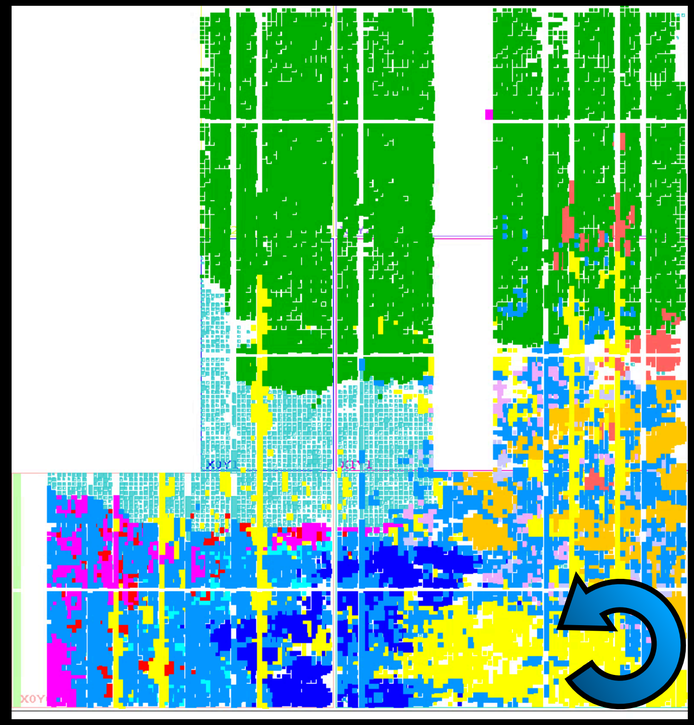
Softmax

Prediction

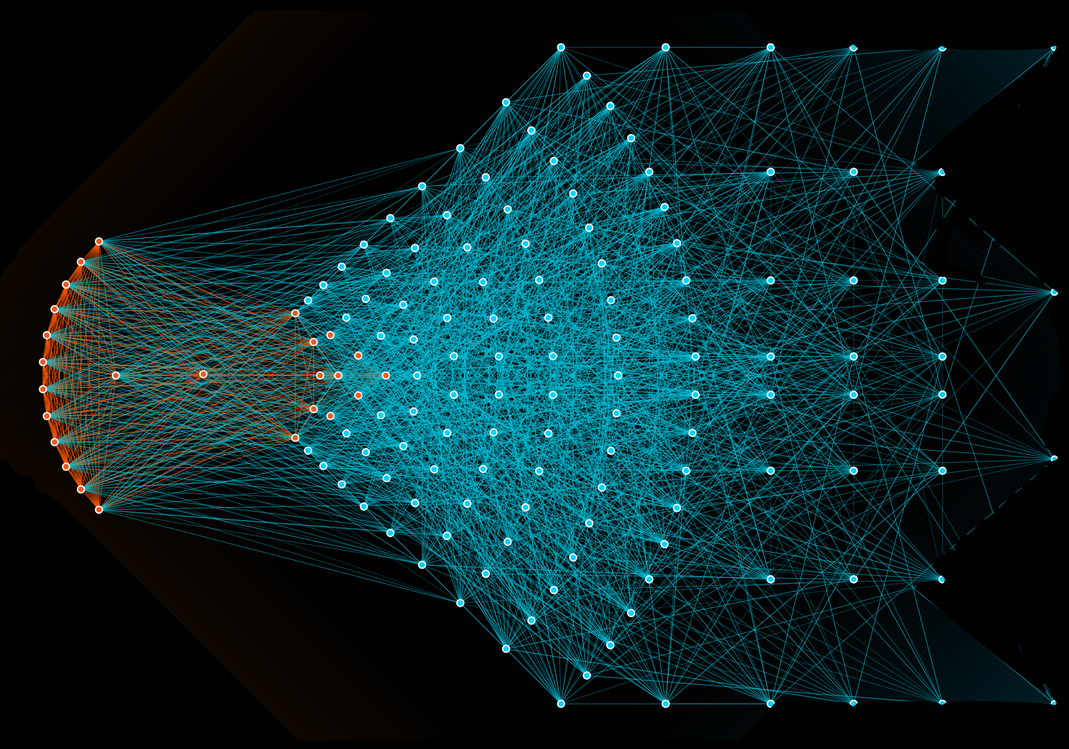
hls4ml

Data flow architecture

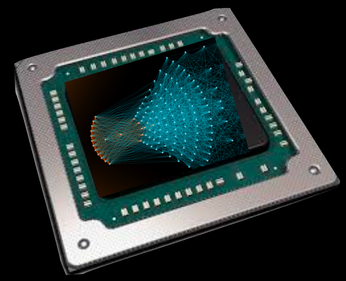
- Tailored hardware for a model (“Decisions are design time”)
- Each layer is separate compute unit
- **Stay on-chip**



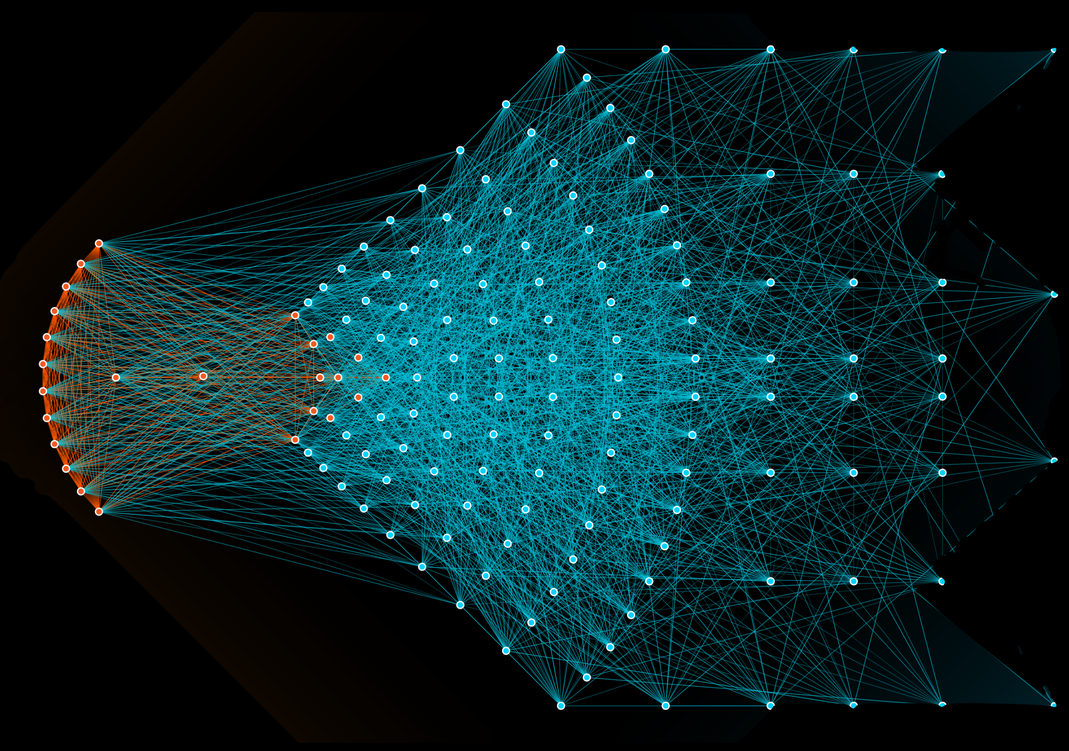
Dataflow



Ideally

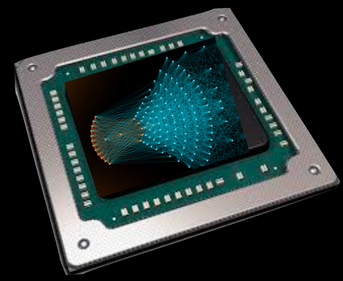


Reality

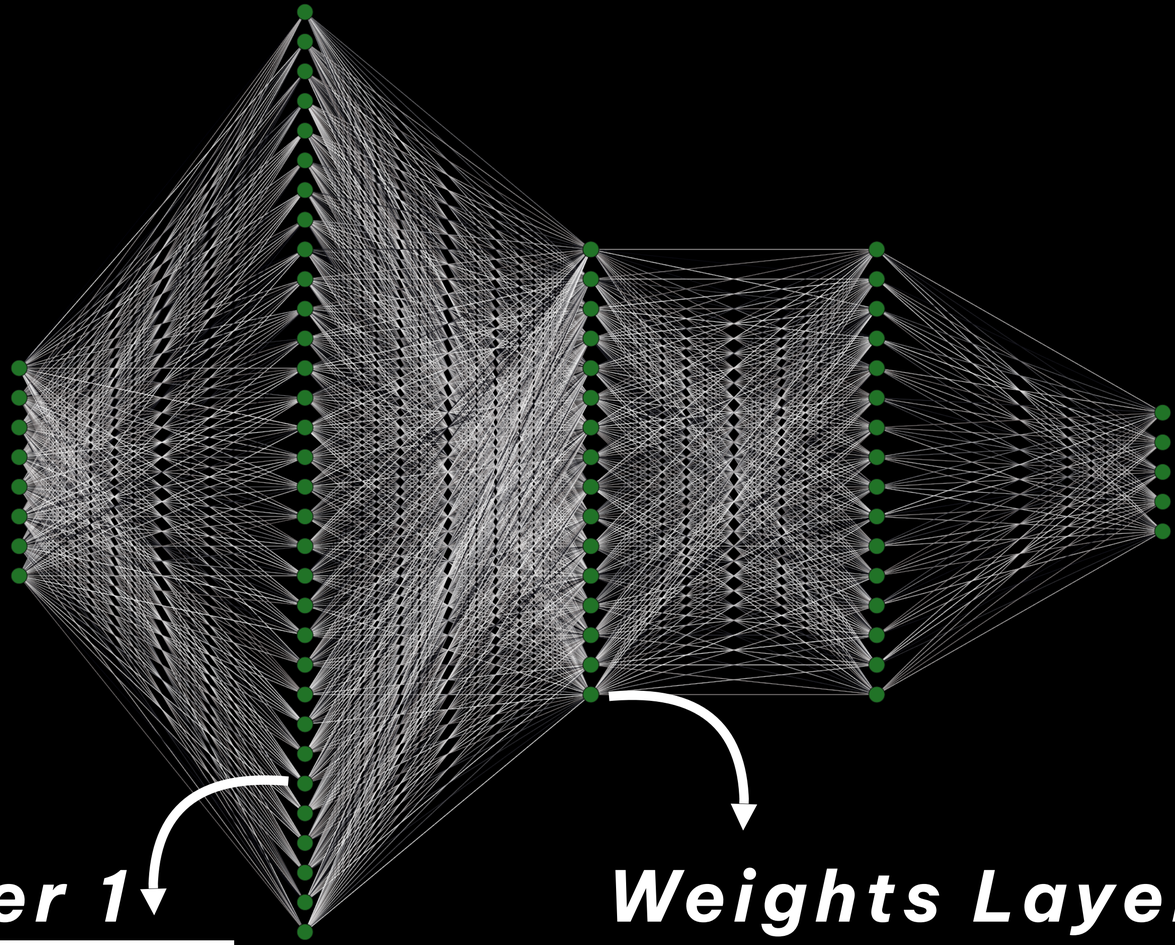


Ideally

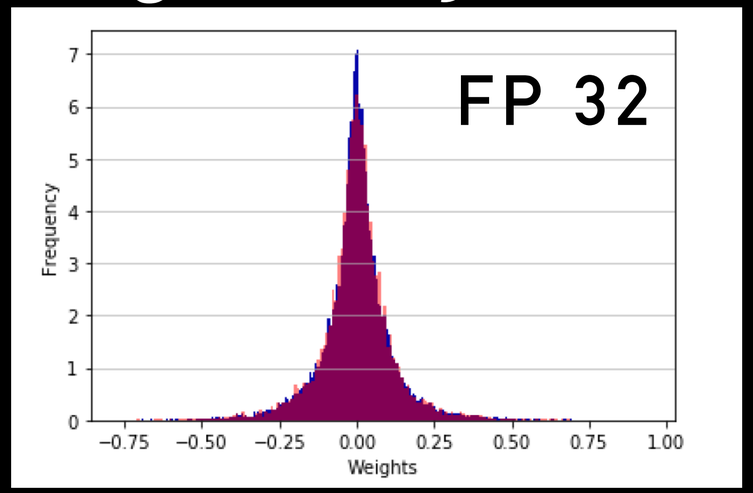
- Quantization
- Pruning
- Parallelisation
- Knowledge distillation



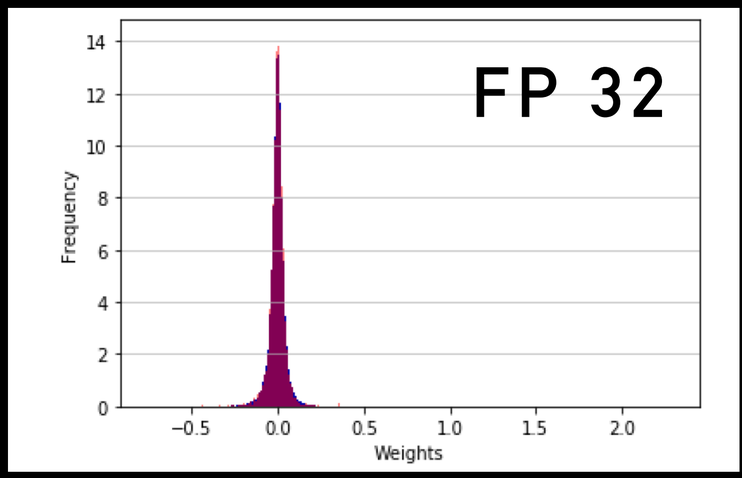
Reality

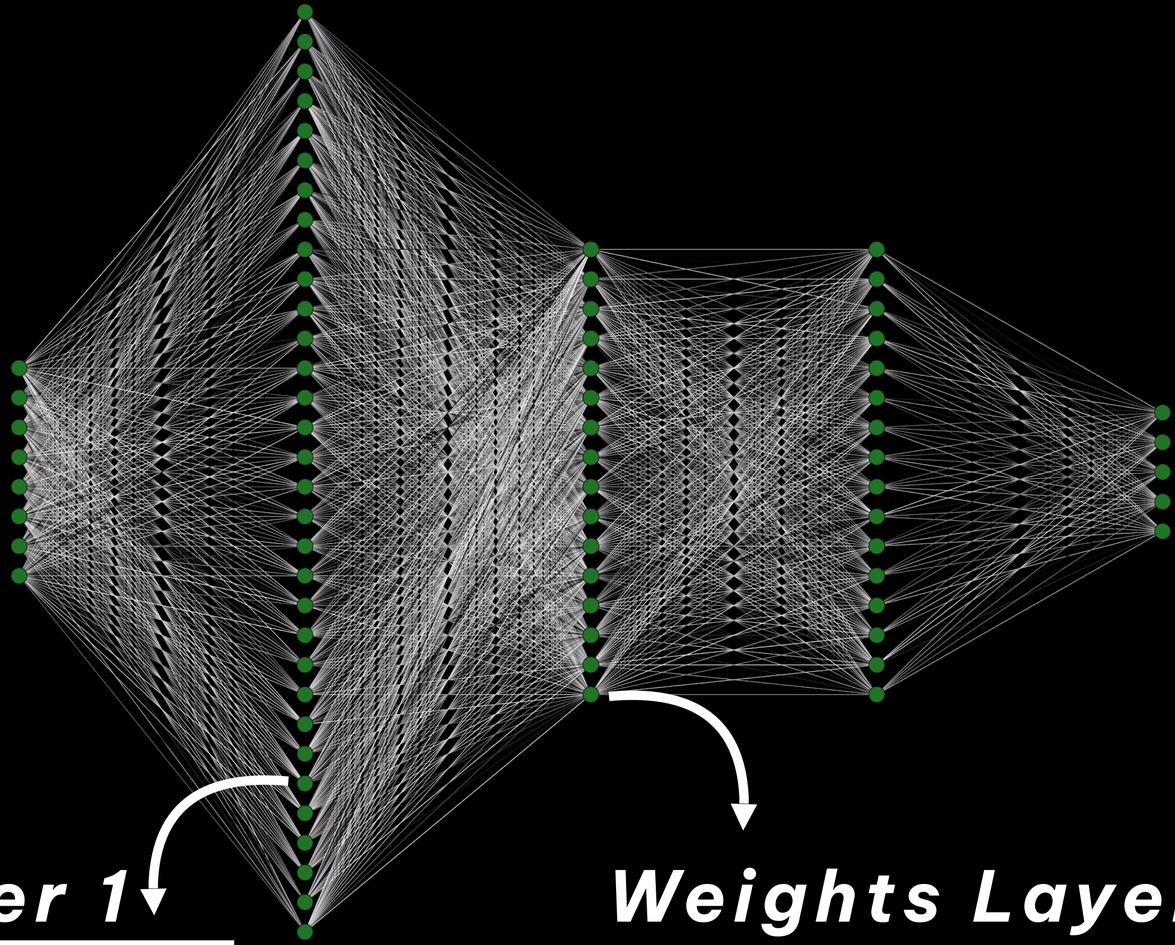


Weights Layer 1



Weights Layer 2





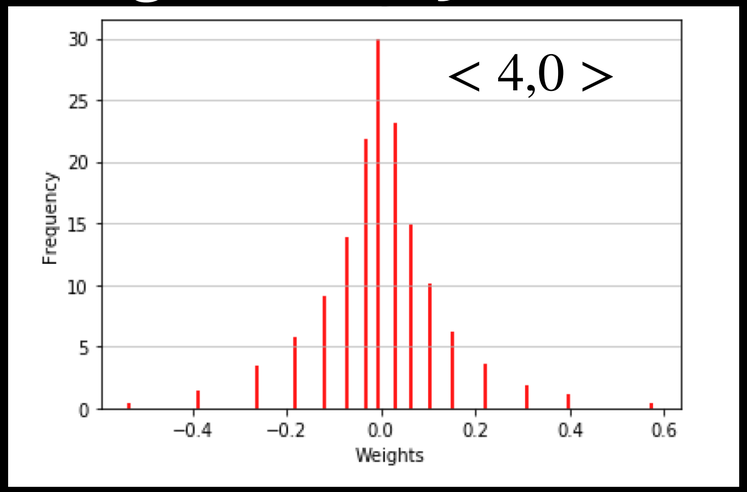
Fixed point

0101.1011101010

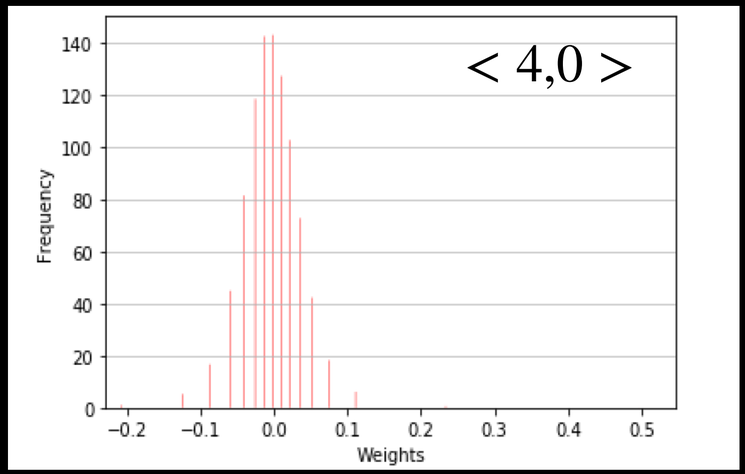
integer | fractional

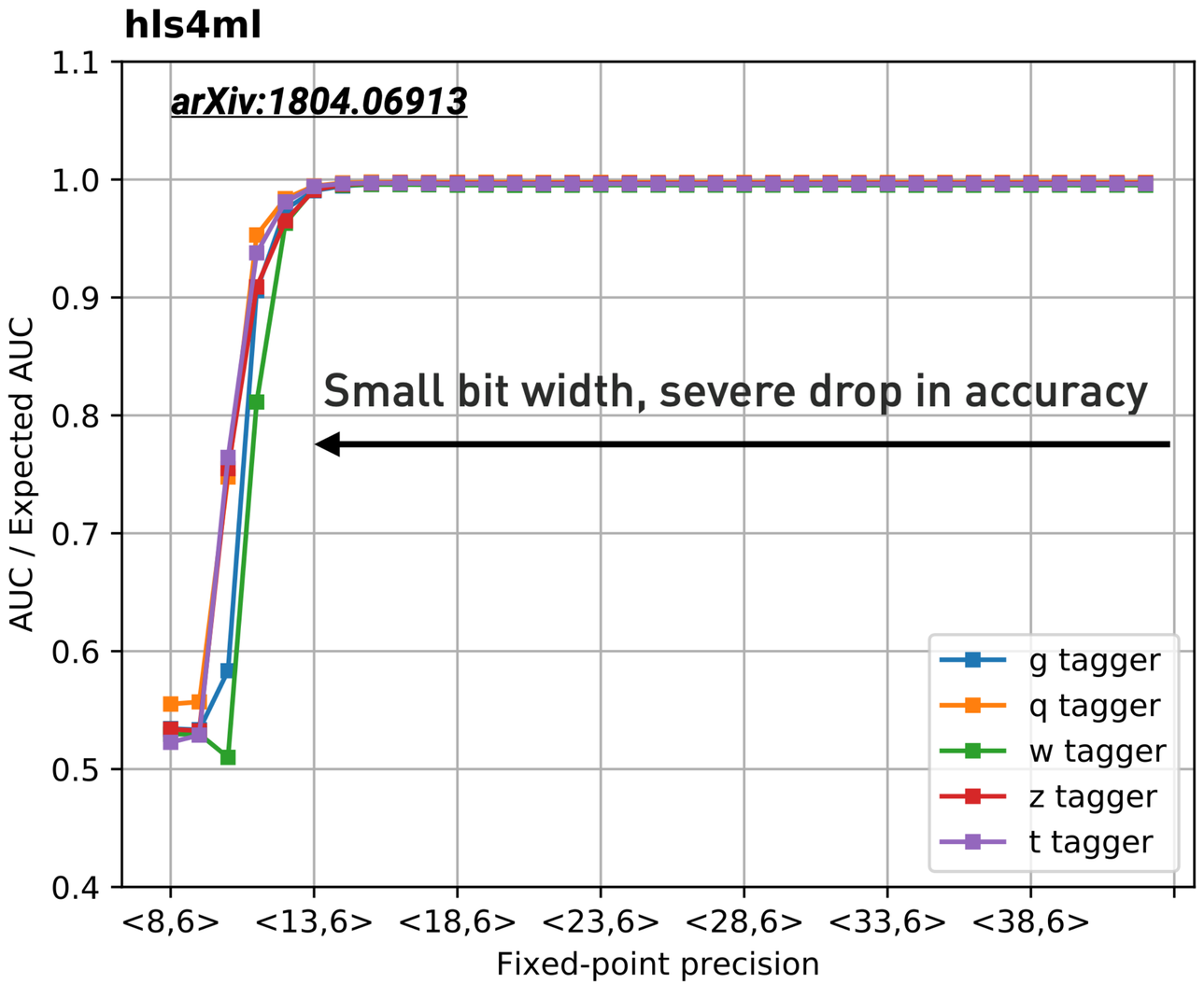
width

Weights Layer 1



Weights Layer 2

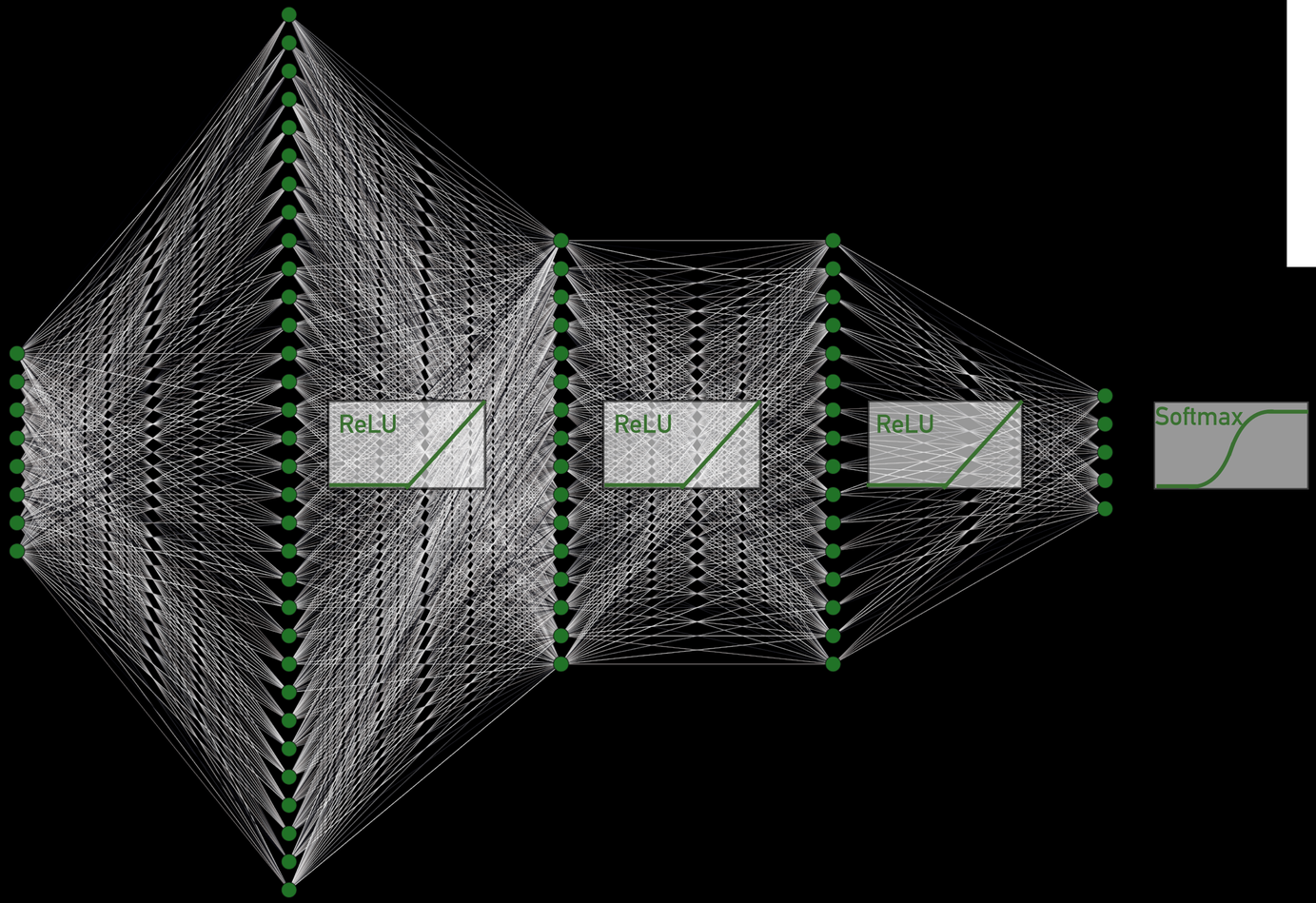


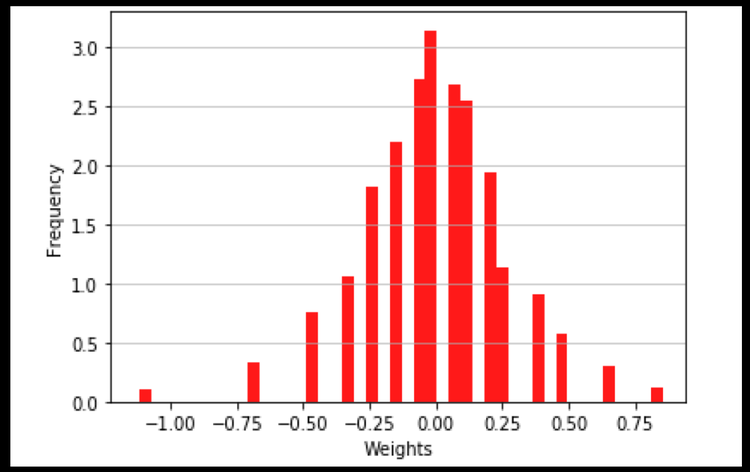

hls4ml
arXiv:1804.06913
Small bit width, severe drop in accuracy
AUC / Expected AUC
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
<8,6>
<13,6>
<18,6>
<23,6>
<28,6>
<33,6>
<38,6>
Fixed-point precision
g tagger
q tagger
w tagger
z tagger
t tagger

hls4ml + Google

Quantization-aware training

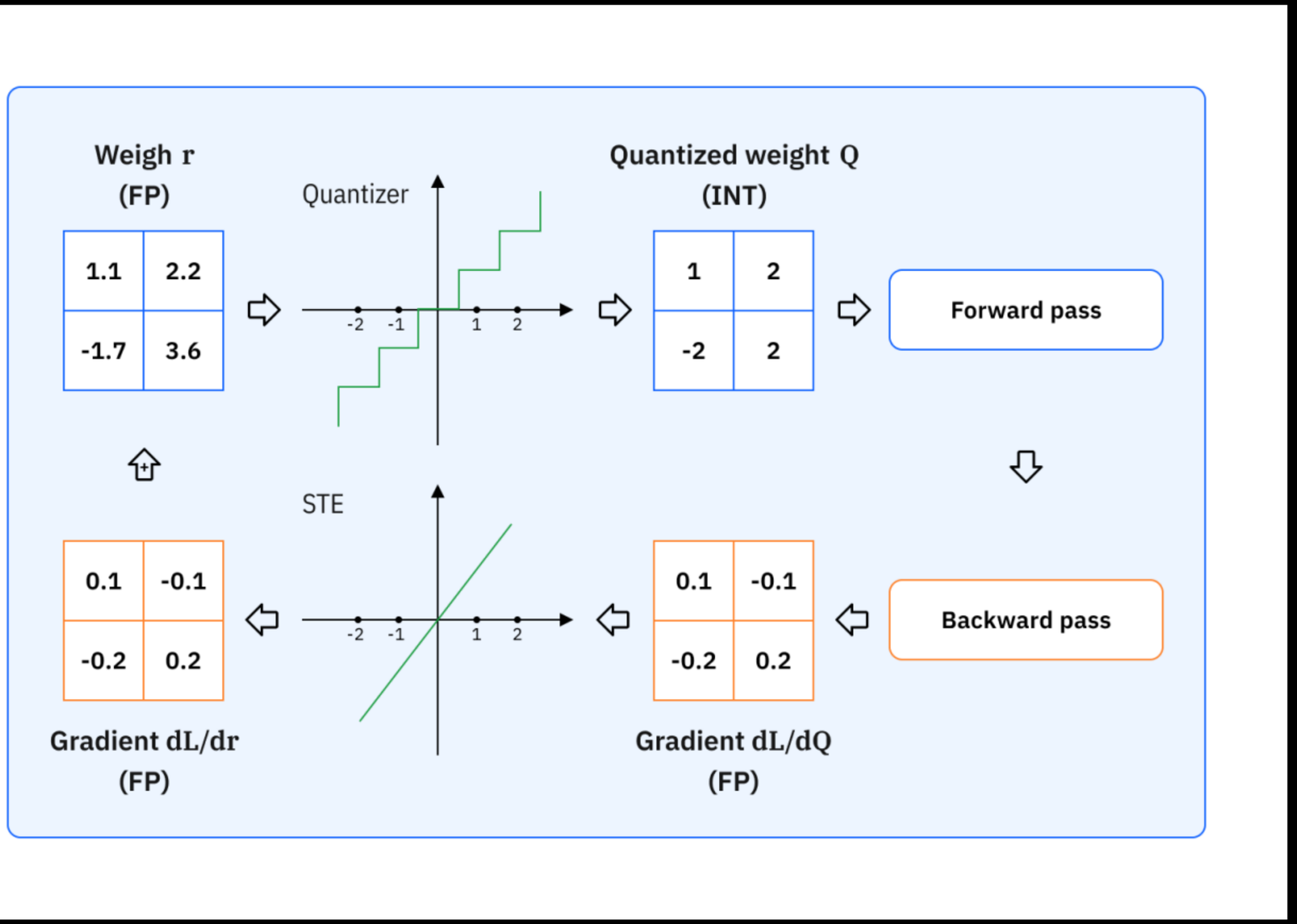
Forward pass →





[Nature Machine Intelligence 3 (2021)]

Straight-through estimator

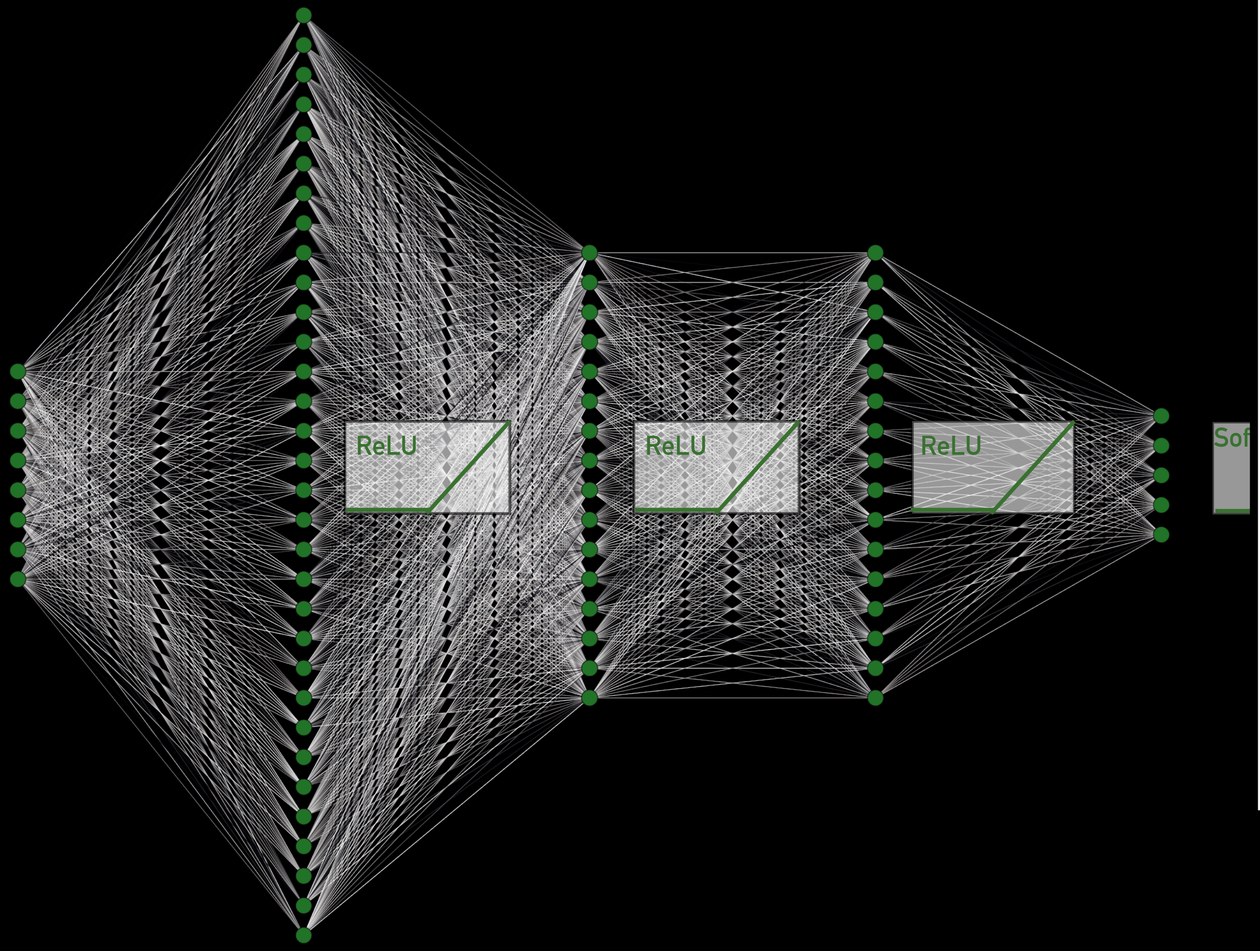


Sundays IEEE RT pre-conference program!

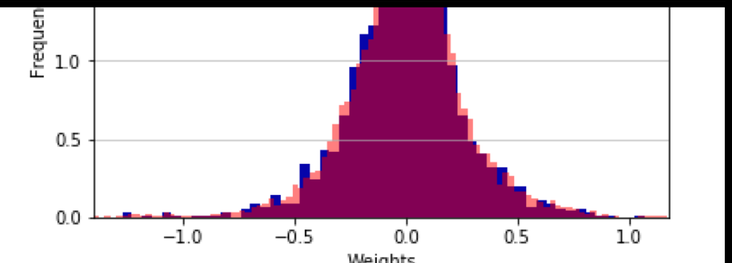
hls4ml + Google

Quantization-aware training

Forward pass →

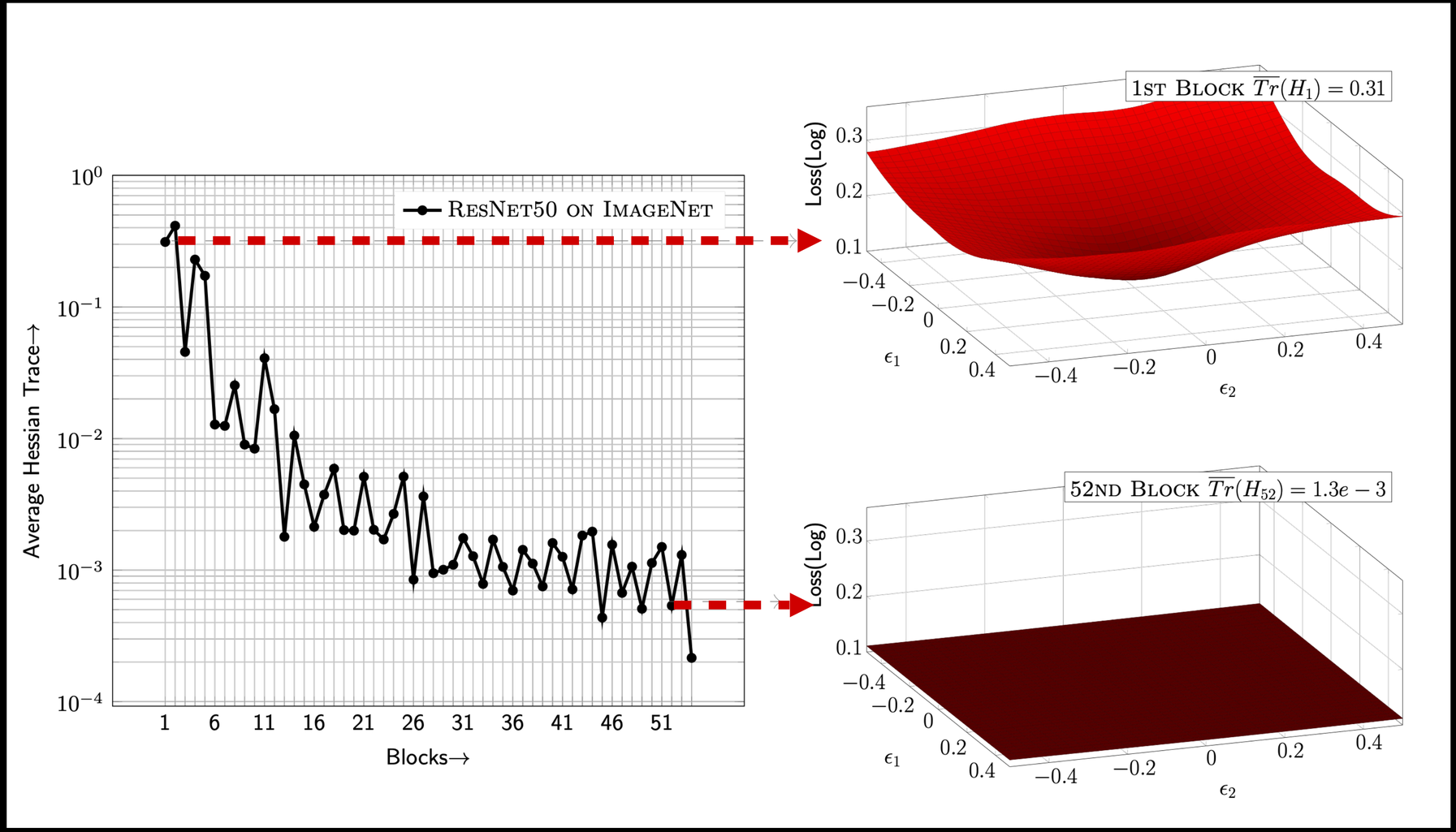



```
from tensorflow.keras.layers import Input, Activation
from qkeras import quantized_bits
from qkeras import QDense, QActivation
from qkeras import QBatchNormalization

x = Input((16))
x = QDense(64,
    kernel_quantizer = quantized_bits(6,0,alpha=1),
    bias_quantizer   = quantized_bits(6,0,alpha=1))(x)
x = QBatchNormalization()(x)
x = QActivation('quantized_relu(6,0)')(x)
x = QDense(32,
    kernel_quantizer = quantized_bits(6,0,alpha=1),
    bias_quantizer   = quantized_bits(6,0,alpha=1))(x)
x = QBatchNormalization()(x)
x = QActivation('quantized_relu(6,0)')(x)
x = QDense(32,
    kernel_quantizer = quantized_bits(6,0,alpha=1),
    bias_quantizer   = quantized_bits(6,0,alpha=1))(x)
x = QBatchNormalization()(x)
x = QActivation('quantized_relu(6,0)')(x)
x = QDense(5,
    kernel_quantizer = quantized_bits(6,0,alpha=1),
    bias_quantizer   = quantized_bits(6,0,alpha=1))(x)
x = Activation('softmax')(x)
```



[Nature Machine Intelligence 3 (2021)]

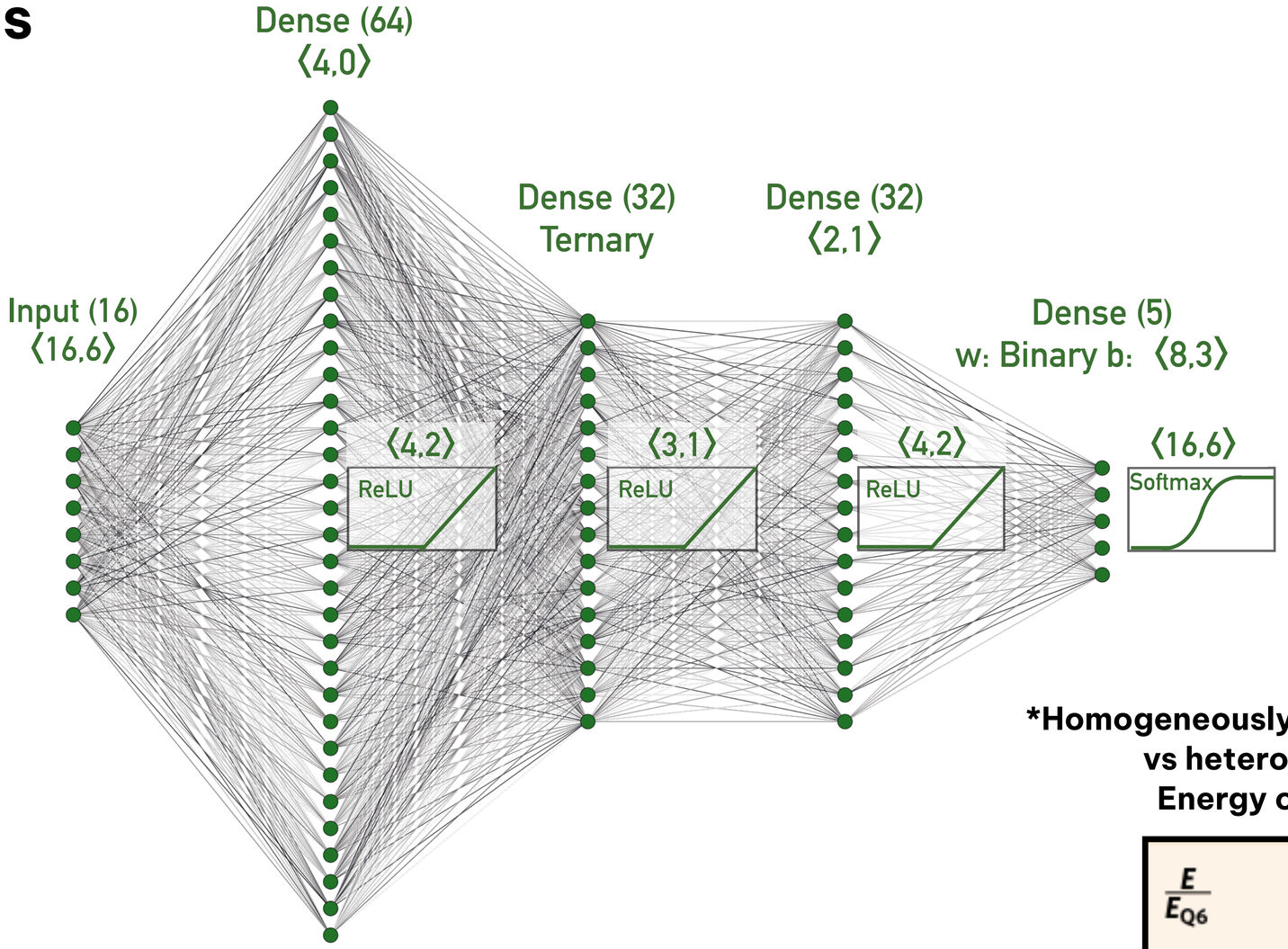
Some layer more accommodating to aggressive quantization than others!



HAWQ-V2, Z. Dong et al. 2020

[Nature Machine Intelligence 3 (2021)]

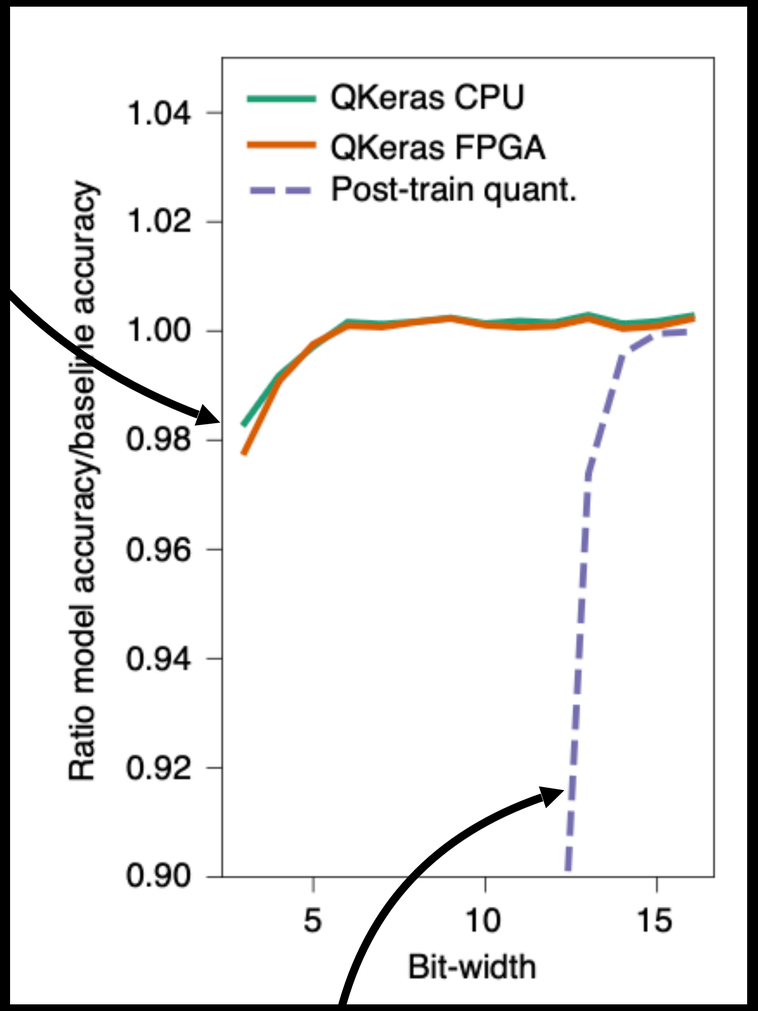
AutoQKeras



***Homogeneously quantized 6 bit model vs heterogeneous model Energy cost, chip area**

$\frac{E}{E_{Q6}}$	$\frac{\text{Bits}}{\text{Bits}_{Q6}}$
0.27	0.18

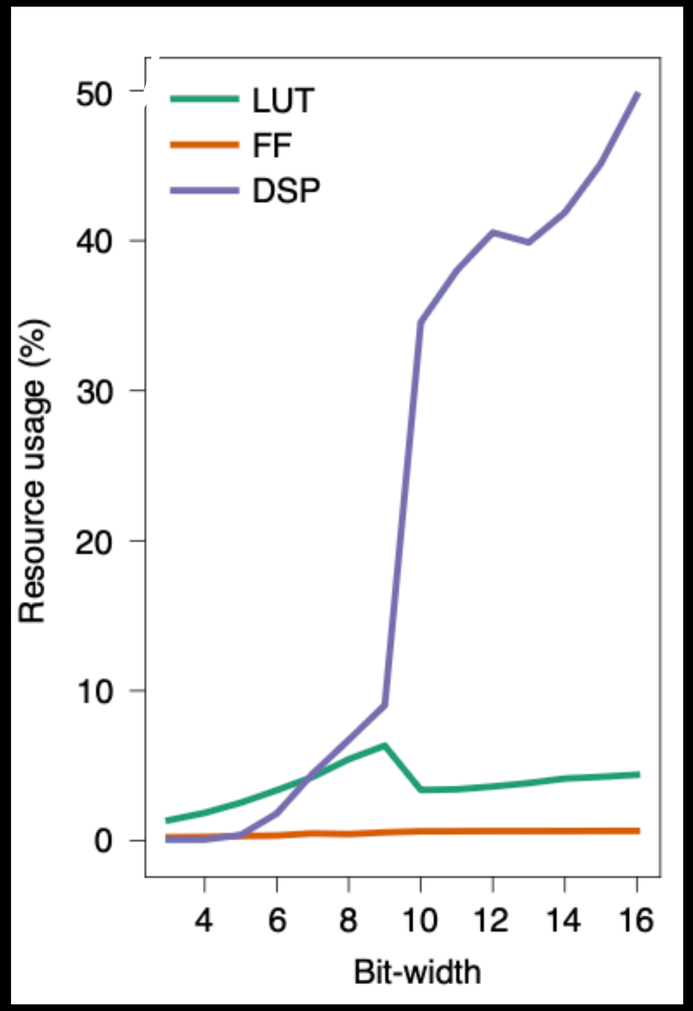
Accuracy



QAT

Post-training quantization

Resource cost



https://arxiv.org/abs/2405.00645
https://arxiv.org/abs/2510.24784

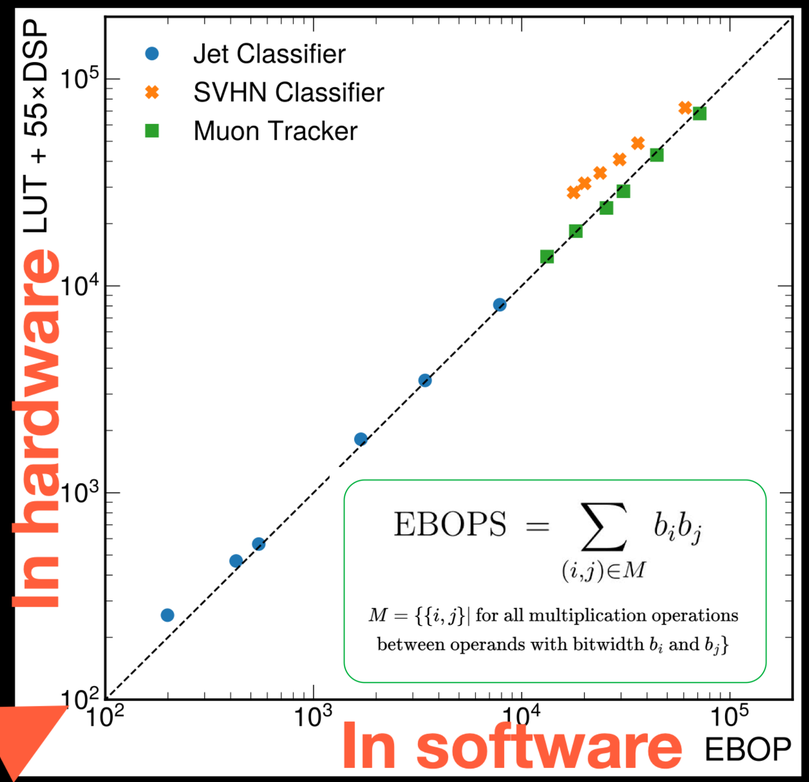
But why stop there?

```
from tensorflow.keras.layers import Input
from HGQ import HQuantize, HDense

inp = Input((16,))
out = HQuantize(name='inp_q', beta=beta)(out)
out = HDense(64, activation='relu', beta=beta)(out)
out = HDense(32, activation='relu', beta=beta)(out)
out = HDense(32, activation='relu', beta=beta)(out)
out = HDense(5, activation='linear', beta=beta)(out)

hgq_model = Model(inp, out)
```

HGQ: High Granularity Quantization

C.Sun et al. 2024

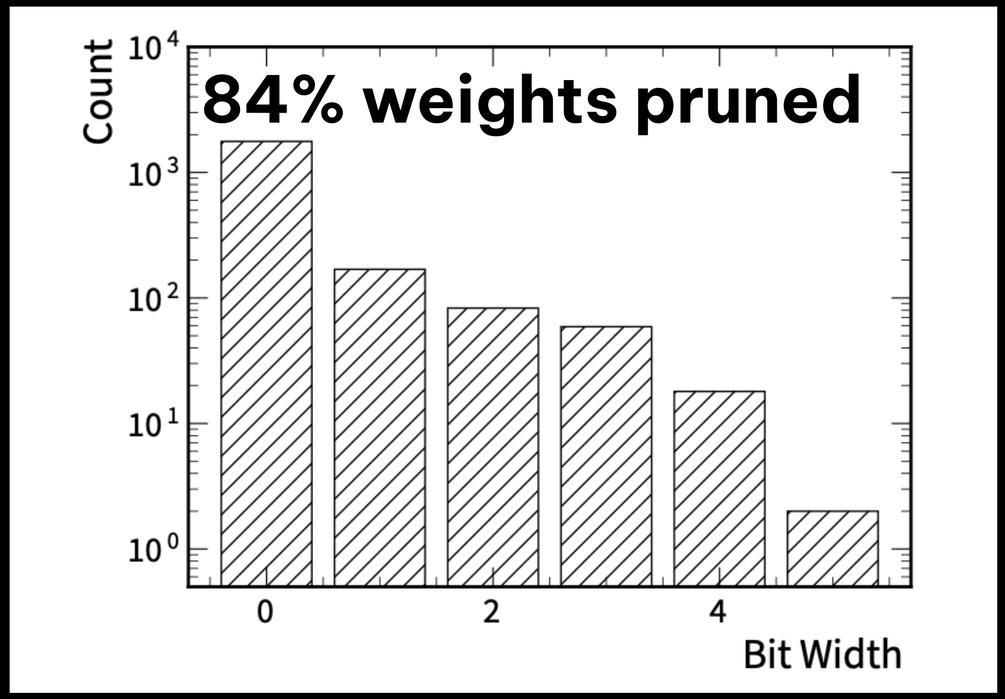
```
from tensorflow.keras.layers import Input
from HGQ import HQuantize, HDense

inp = Input((16,))
out = HQuantize(name='inp_q', beta=beta)(out)
out = HDense(64, activation='relu', beta=beta)(out)
out = HDense(32, activation='relu', beta=beta)(out)
out = HDense(32, activation='relu', beta=beta)(out)
out = HDense(5, activation='linear', beta=beta)(out)

hgq_model = Model(inp, out)
```

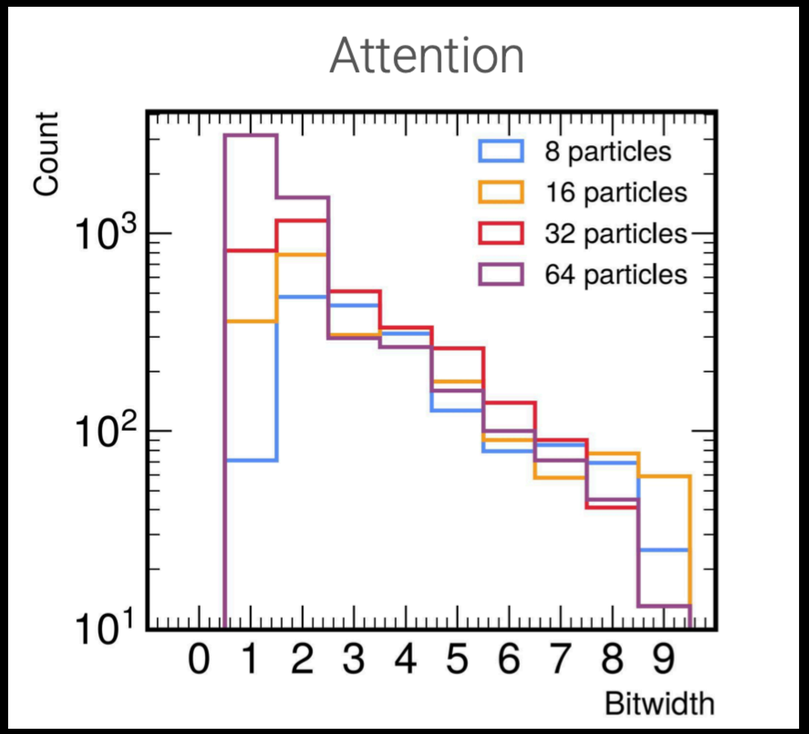


Proxy for resource cost

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \beta \cdot \overline{\text{EBOPs}} + \gamma \cdot \text{L1}_{\text{norm}}$$

Making bitwidths differentiable! Fully heterogeneous quantization!





It's not a FAD:
first results in using Flows for unsupervised Anomaly Detection at 40 MHz at the Large Hadron Collider

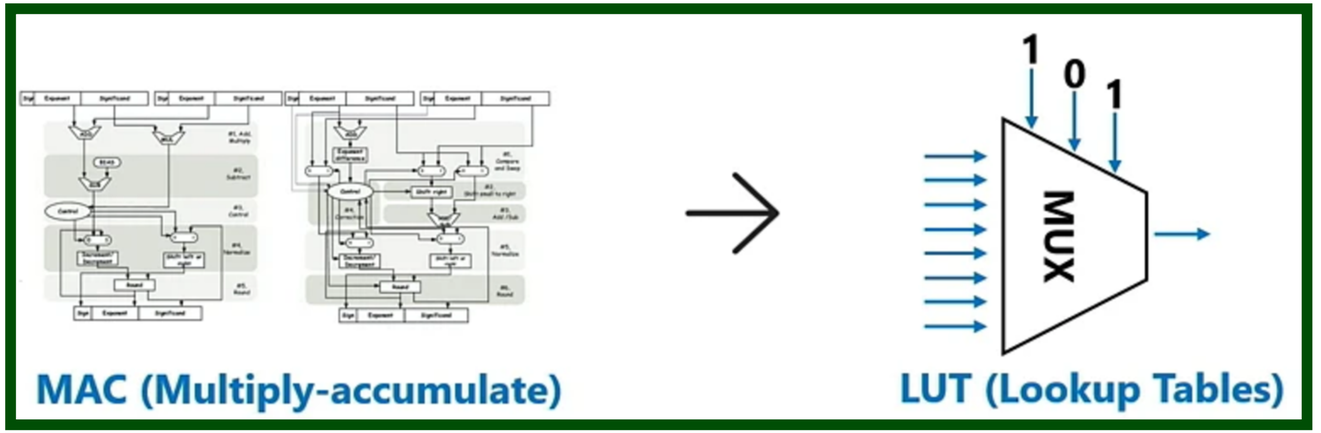
Normalizing Flows on chip with 35 ns latency

Heterogeneously quantized transformers with 100 ns latency!

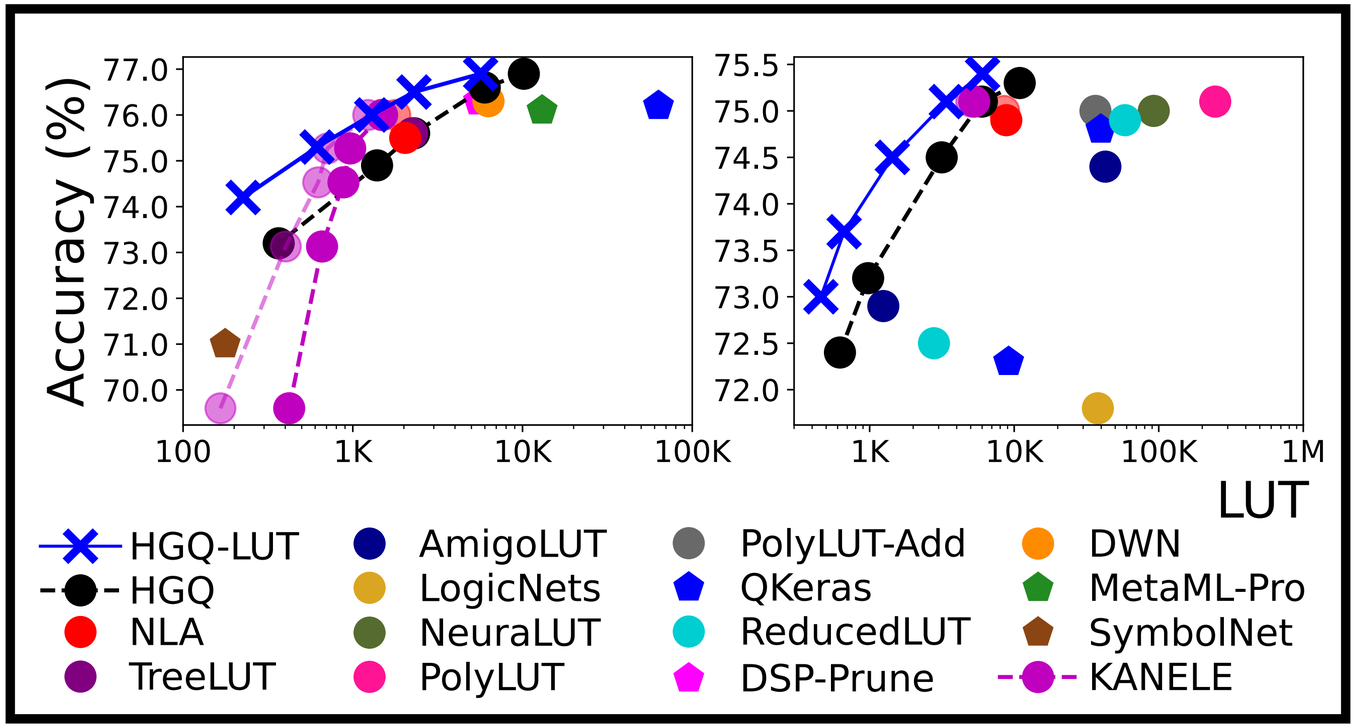
https://arxiv.org/html/2202.03772v3

HGQ-LUT

C. Sun et al (2026)



Blue is LUT dense layers
Black is without!



https://arxiv.org/abs/2604.22293

OpenReview.net

← Go to **NeurIPS 2022 Track Datasets and Benchmarks** h...

Why do tree-based models still outperform deep learning on typical tabular data?

Leo Grinsztajn, Edouard Oyallon, Gael Varoquaux

06 Jun 2022 (modified: 16 Jan 2023) NeurIPS 2022 Datasets and Benchmarks Readers: Everyone Show Bibtex Show Revisions

Abstract: While deep learning has enabled tremendous progress on text and image datasets, its superiority on tabular data is not clear. We contribute extensive benchmarks of standard and novel deep learning methods as well as tree-based models such as XGBoost and Random Forests, across a large number of datasets and hyperparameter combinations. We define a standard set of 45 datasets from varied domains with clear characteristics of tabular data and a benchmarking methodology accounting for both fitting models and finding good hyperparameters. Results show that tree-based models remain state-of-the-art on medium-sized data (~10K samples) even without accounting

Cornell University

arXiv > cs > arXiv:2210.05189

Search... Help | Advanc

Computer Science > Machine Learning

[Submitted on 11 Oct 2022 (v1), last revised 25 Oct 2022 (this version, v3)]

Neural Networks are Decision Trees

Caglar Aytekin

In this manuscript, we show that any neural network with any activation function can be represented as a decision tree. The representation is equivalence and not an approximation, thus keeping the accuracy of the neural network exactly as is. We believe that this work provides better understanding of neural networks and paves the way to tackle their black-box nature. We share equivalent trees of some neural networks and show that besides providing interpretability, tree representation can also achieve some computational advantages for small networks. The analysis holds both for fully connected and convolutional networks, which may or may not also include skip connections and/or normalizations.

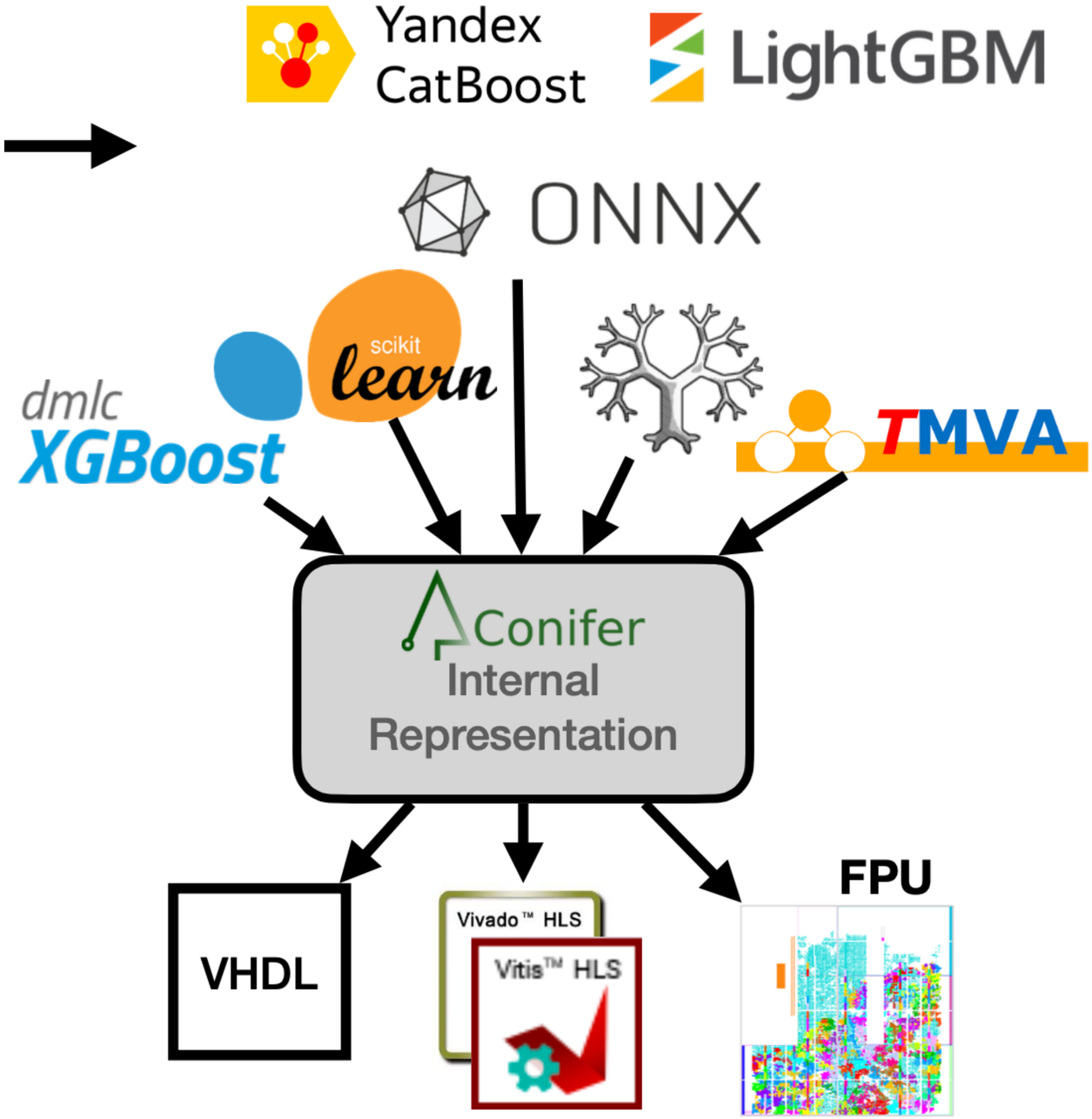
Subjects: **Machine Learning (cs.LG)**
Cite as: arXiv:2210.05189 **[cs.LG]**
(or arXiv:2210.05189v3 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2210.05189

Submission history

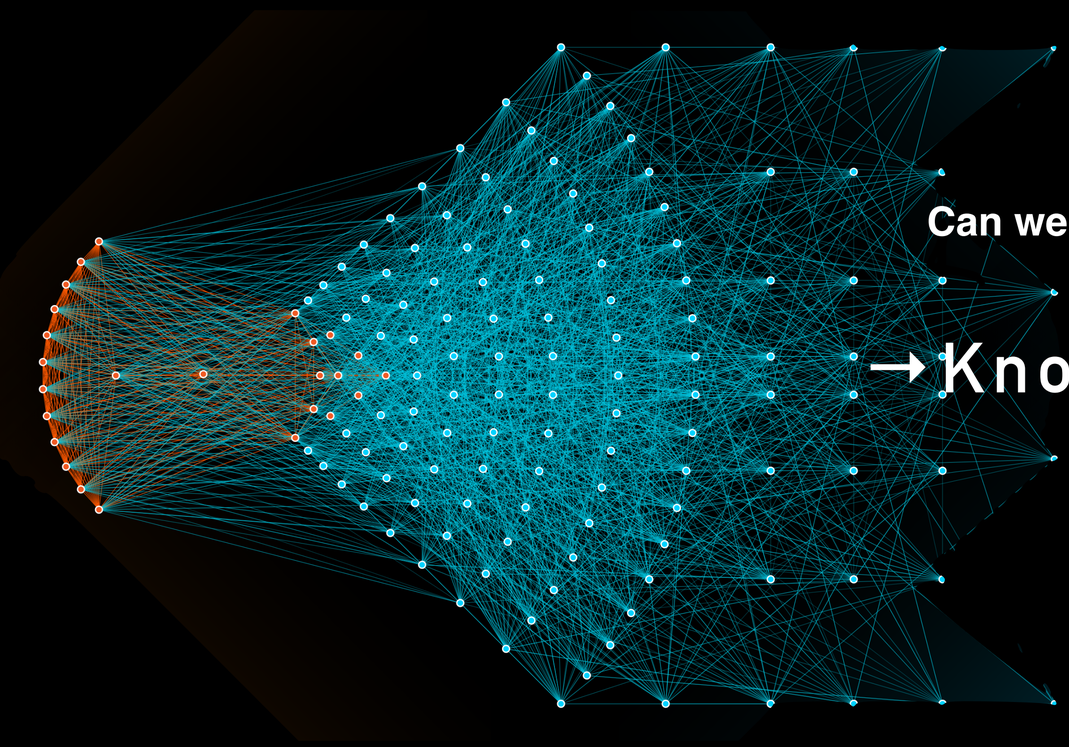
From: Çağlar Aytekin [view email]
[v1] Tue, 11 Oct 2022 06:49:51 UTC (216 KB)
[v2] Mon, 17 Oct 2022 15:18:14 UTC (224 KB)
[v3] Tue, 25 Oct 2022 17:32:33 UTC (240 KB)

https://arxiv.org/abs/2210.05189

%VU9P	Accuracy	Latency	DSP	LUT
qDNN	75.6%	40 ns	22 (~0%)	1%
BDT	74.9%	5 ns	-	0.5%

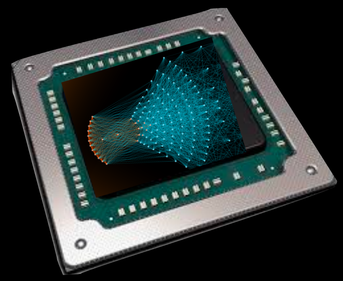


[Yggdrasil+Conifer]



Can we have the best of both worlds?

→Knowledge Distillation

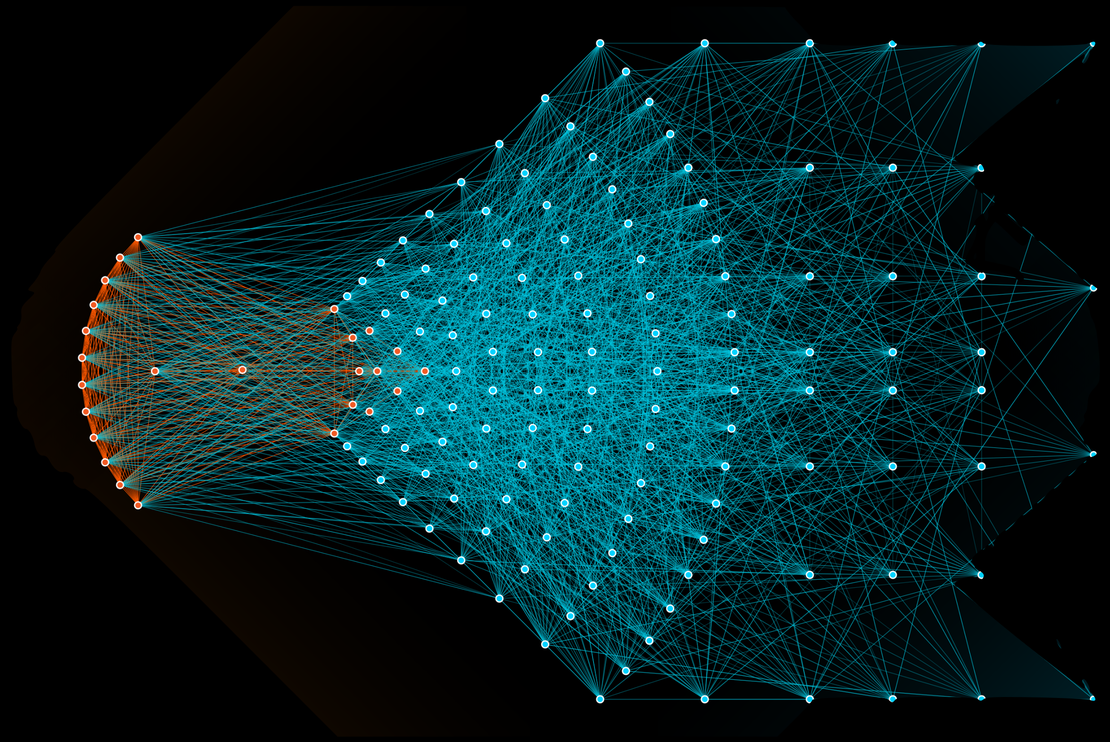


Train

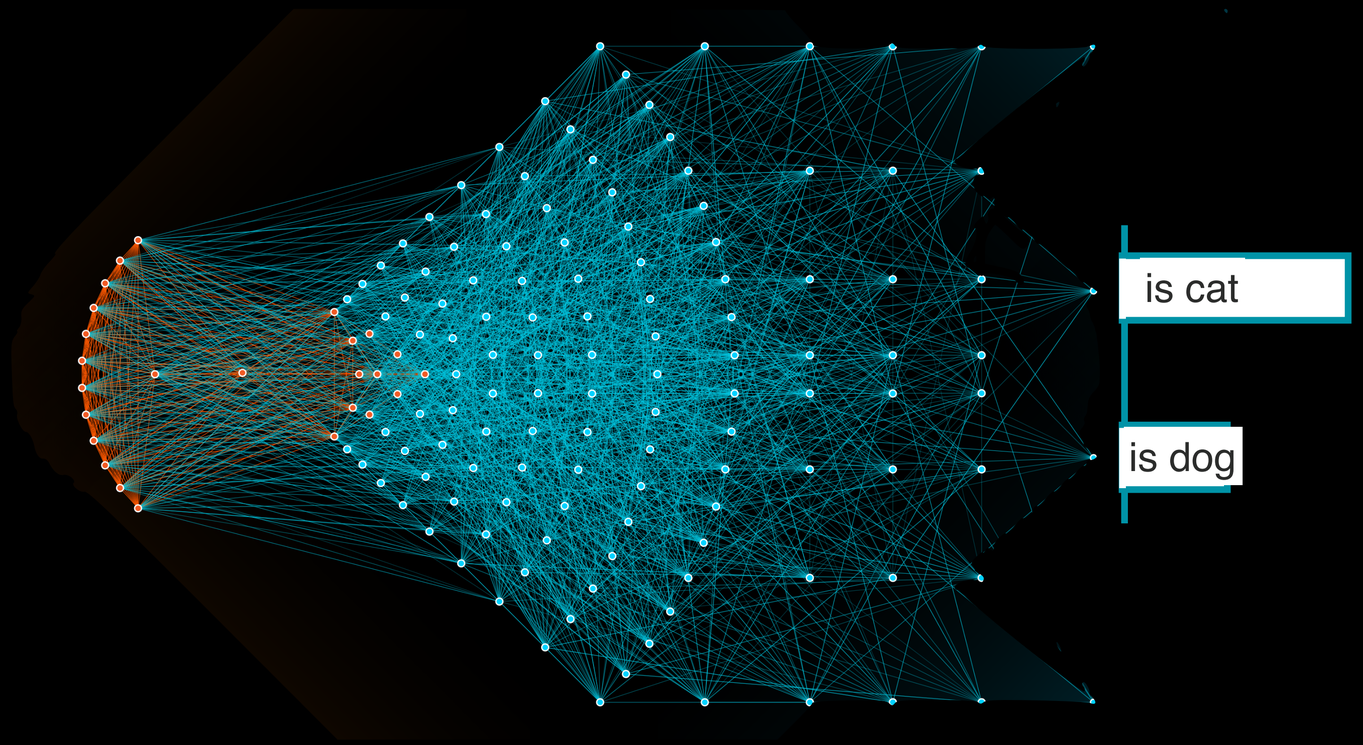
Inference

Cat

Dog



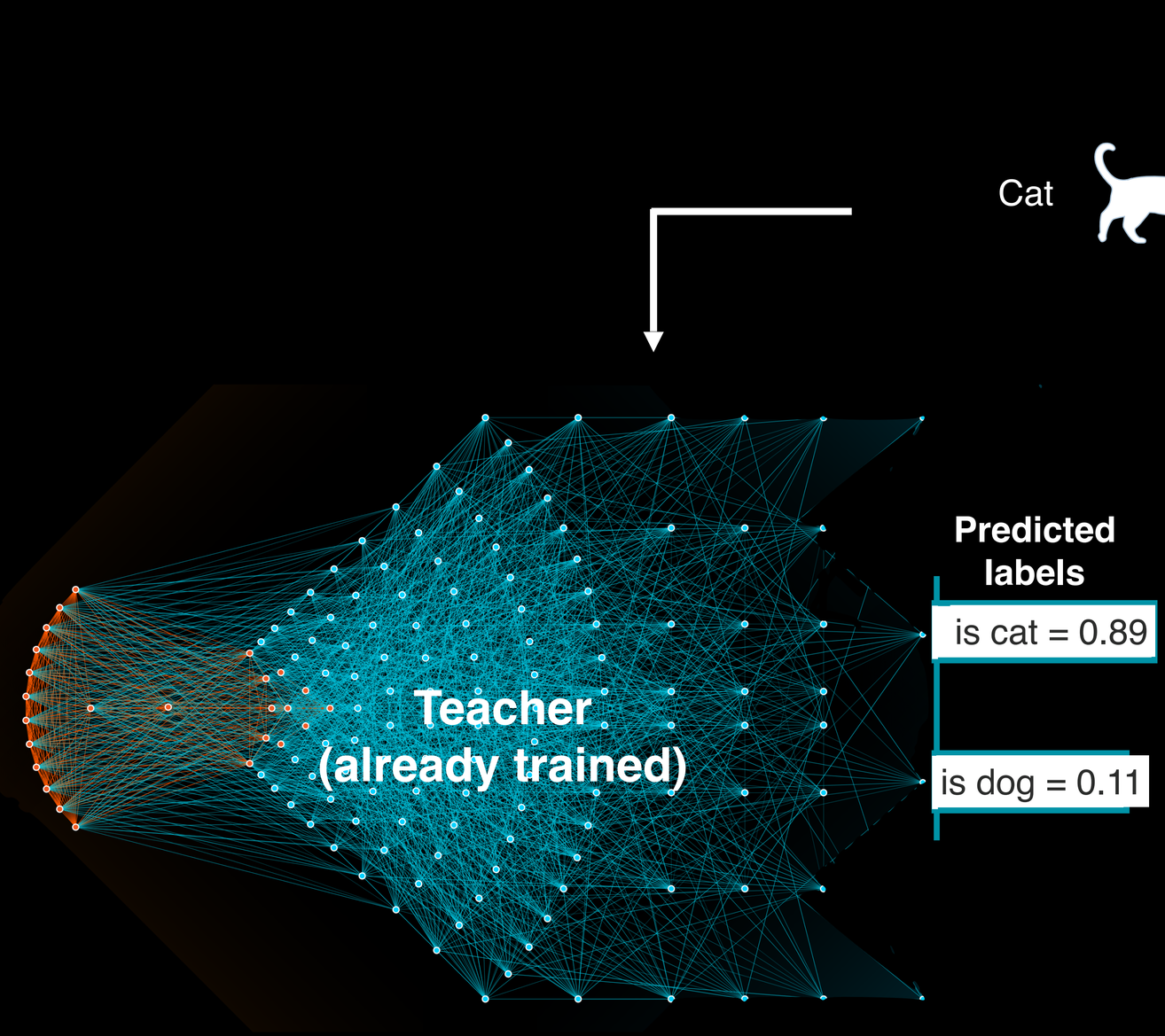
Cat



is cat

is dog

Cat

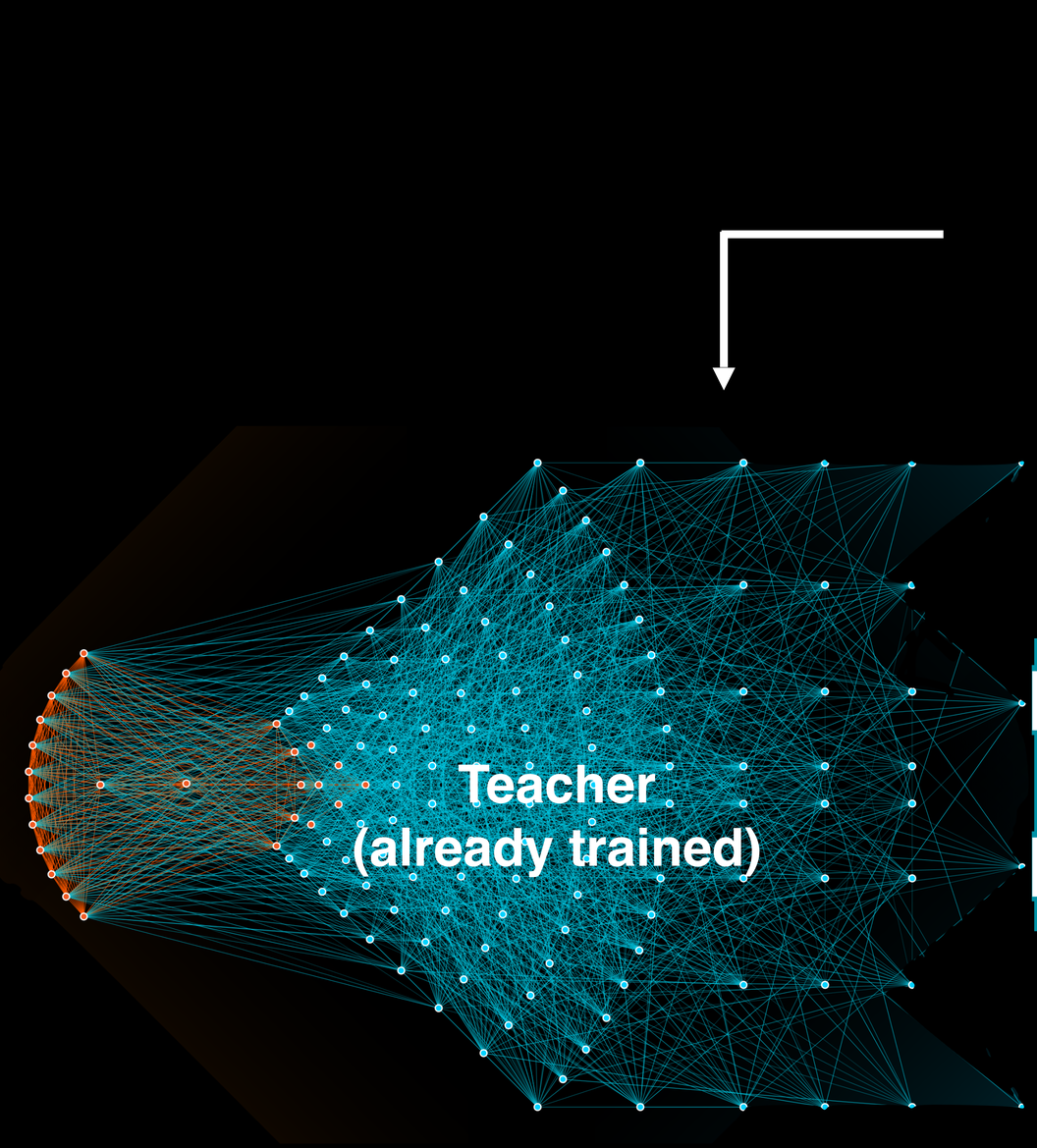


Predicted labels

is cat = 0.89

is dog = 0.11

Cat



Predicted labels

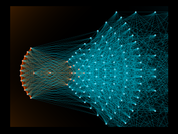
is cat = 0.89

is dog = 0.11

True labels

is cat = 1

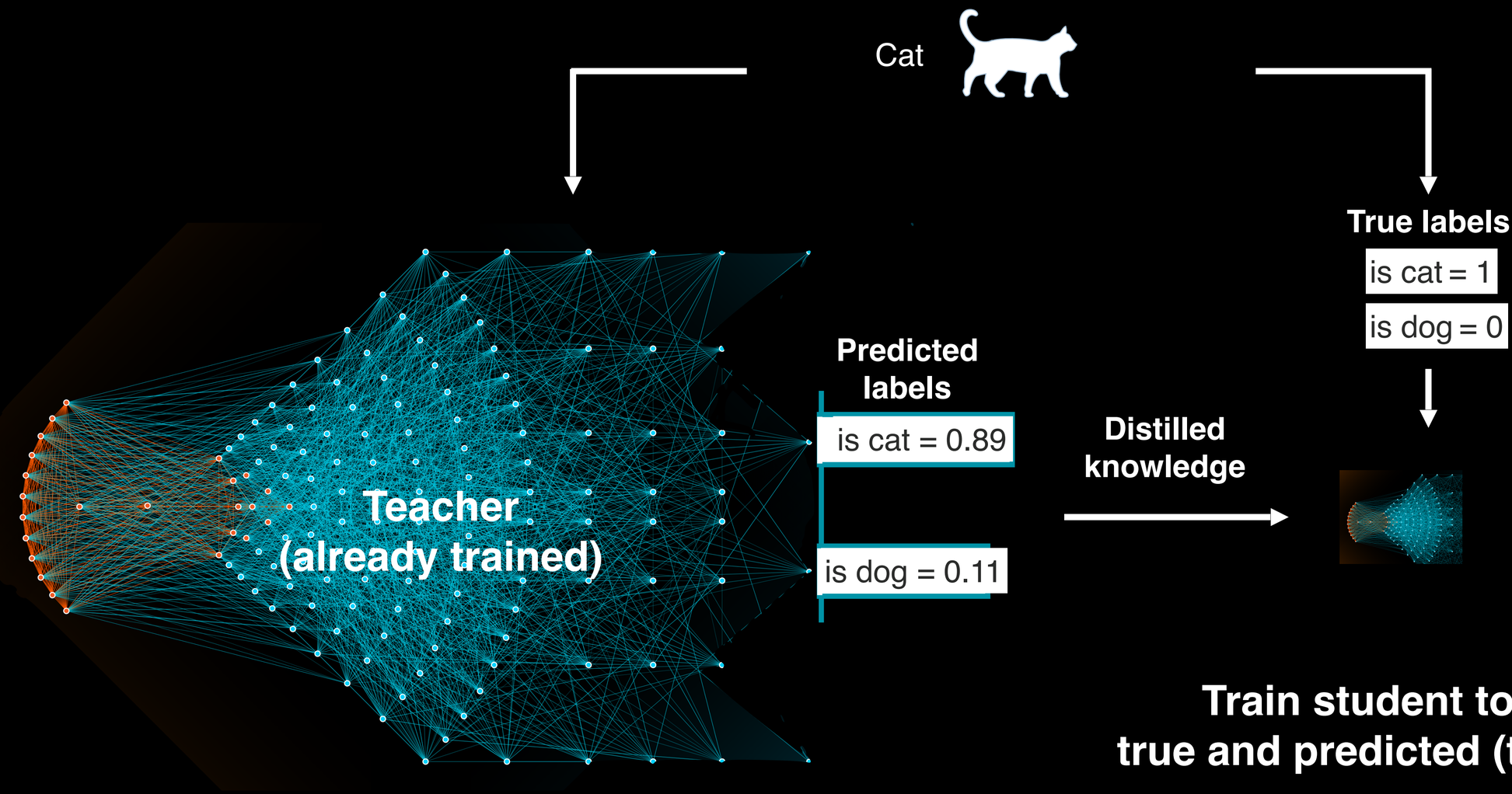
is dog = 0



Predicted labels	True labels
is cat = 0.46	is cat = 0
is dog = 0.54	is dog = 1

Soft labels contain information!!

Predicted labels	True labels
is cat = 0.03	is cat = 0
is dog = 0.97	is dog = 1

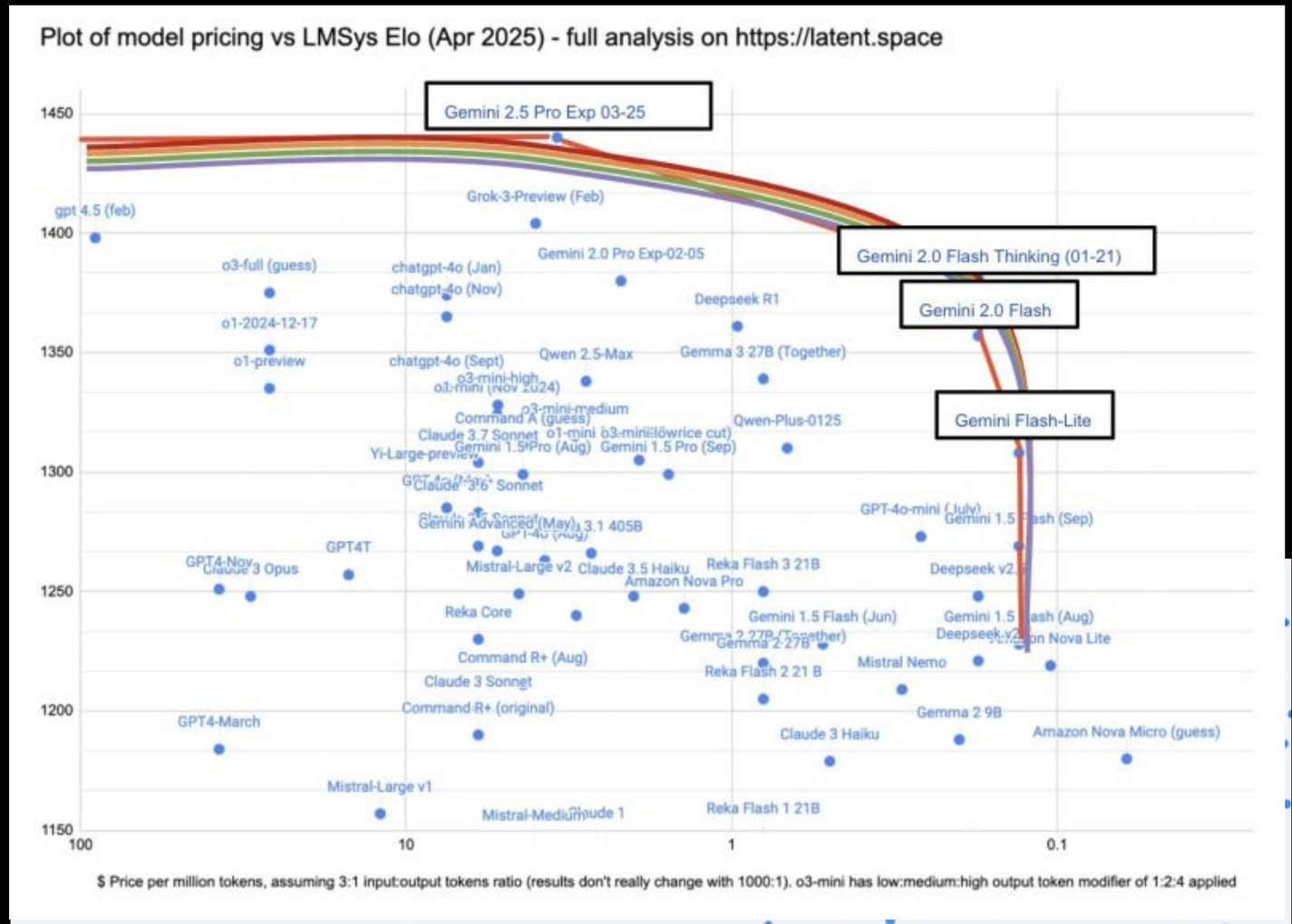


Train student to learn both true and predicted (teacher) labels!

$$L_{total} = \beta \times L_{Distillation} + \alpha \times L_{student}$$

Student learns subtle learned features from teacher!

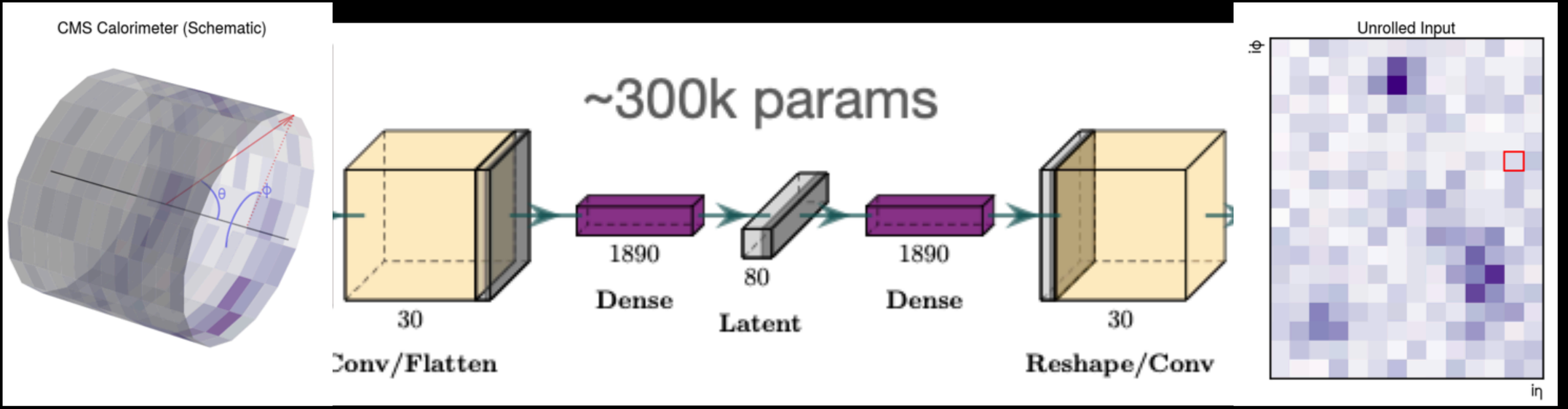
Used for Gemini Flash



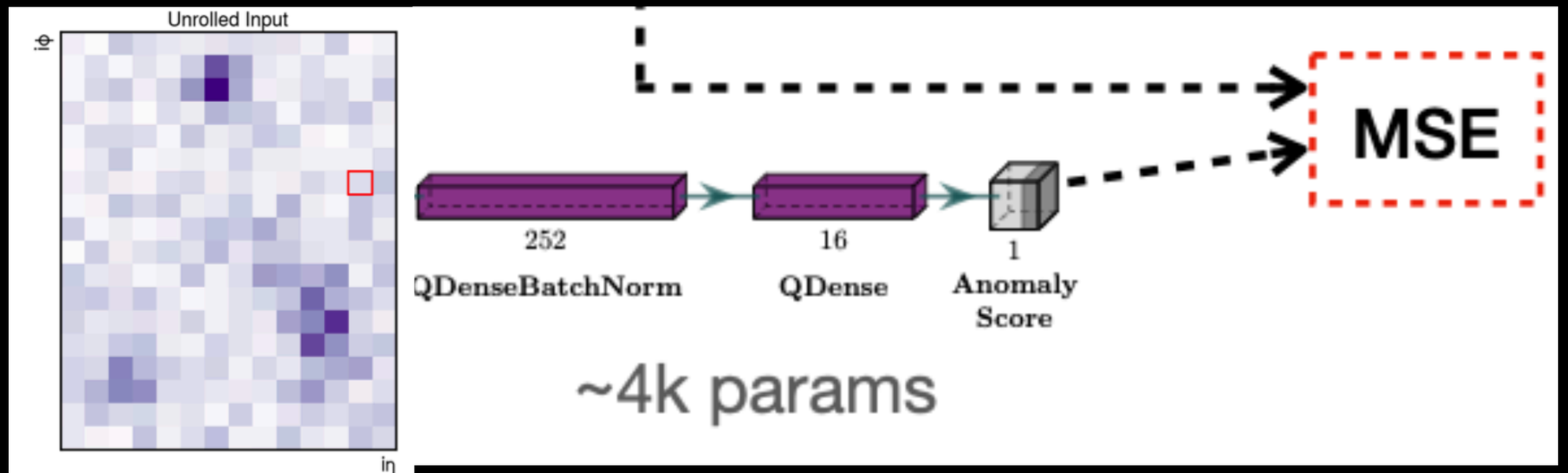
Gemini 3 Flash

www.latent.space/,
Jeff Dean HLF

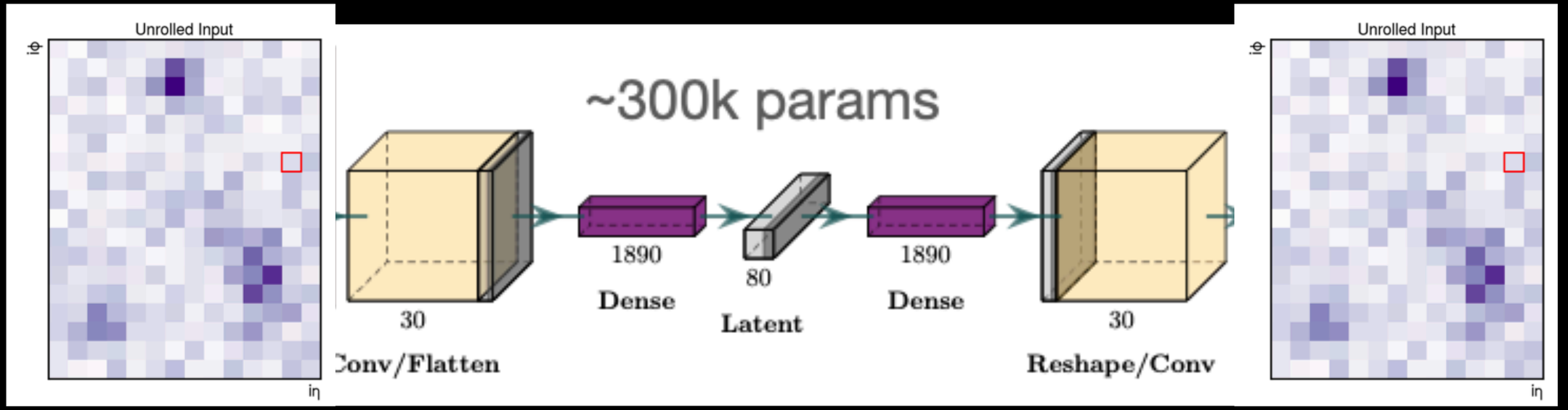
Used in CMS trigger!



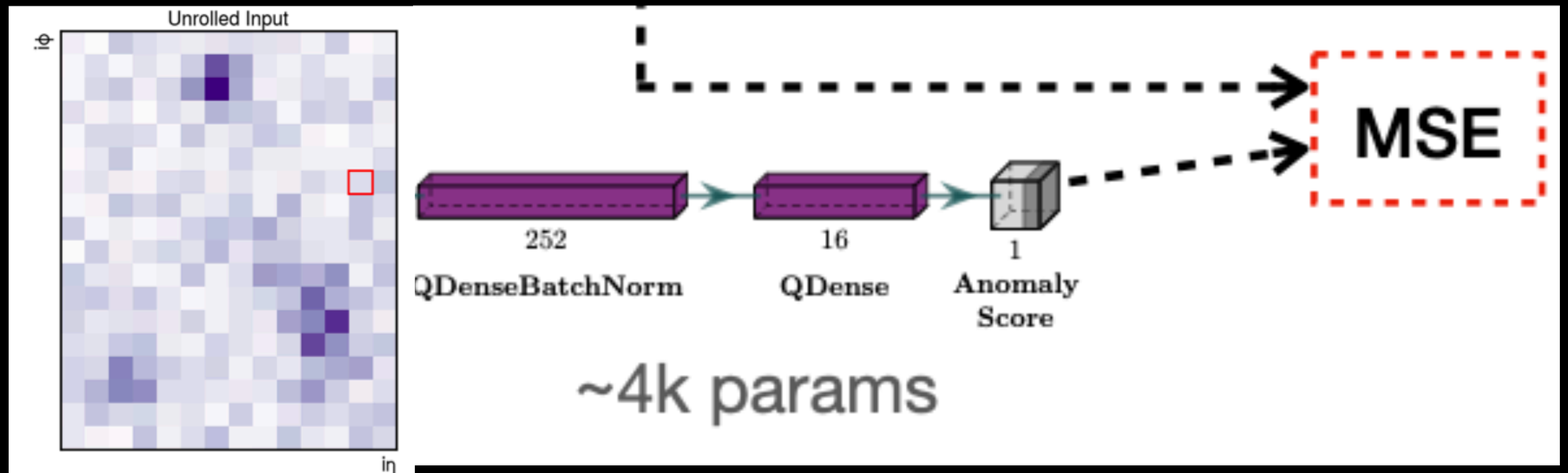
Knowledge distillation: input to anomaly score

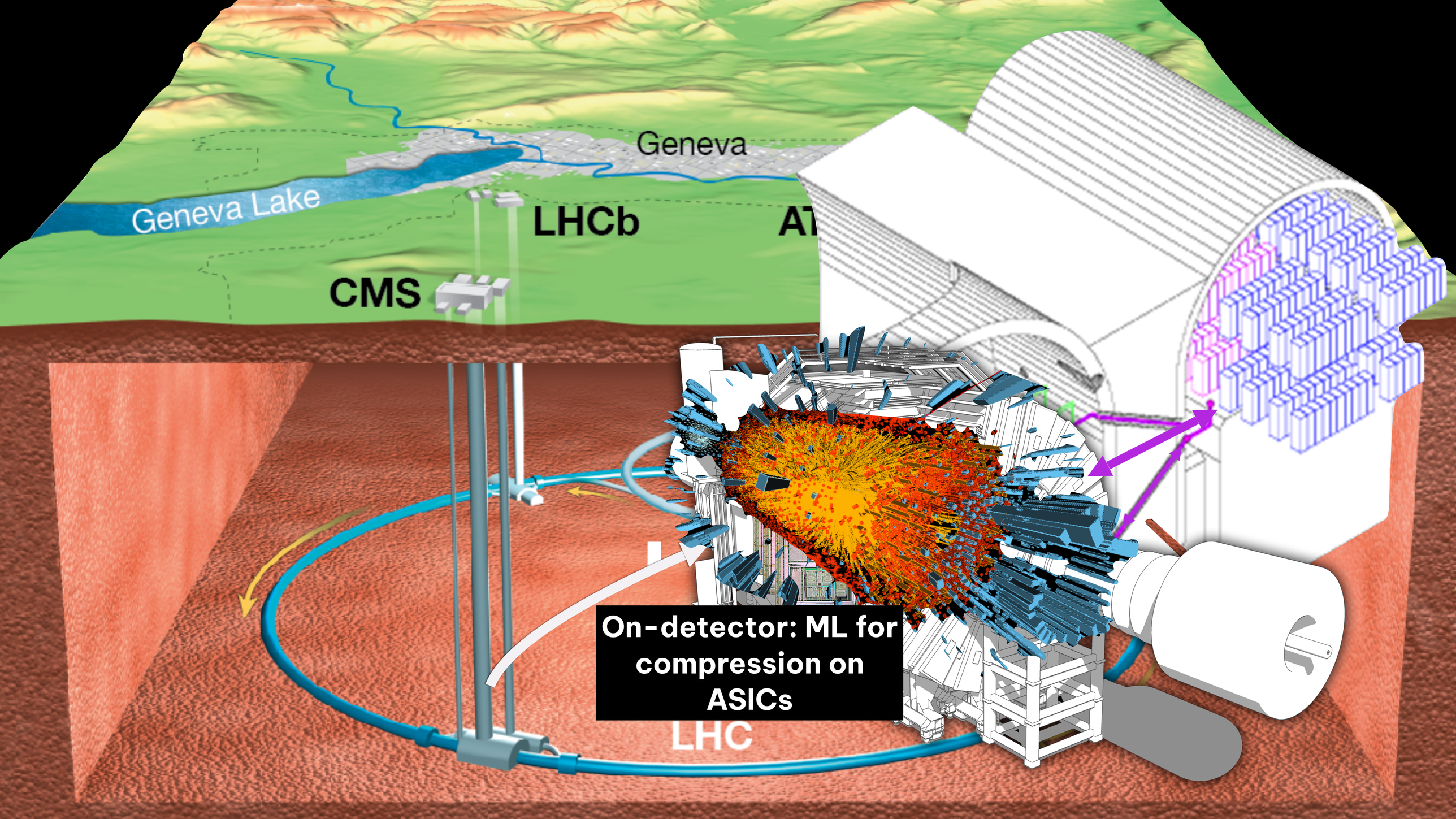


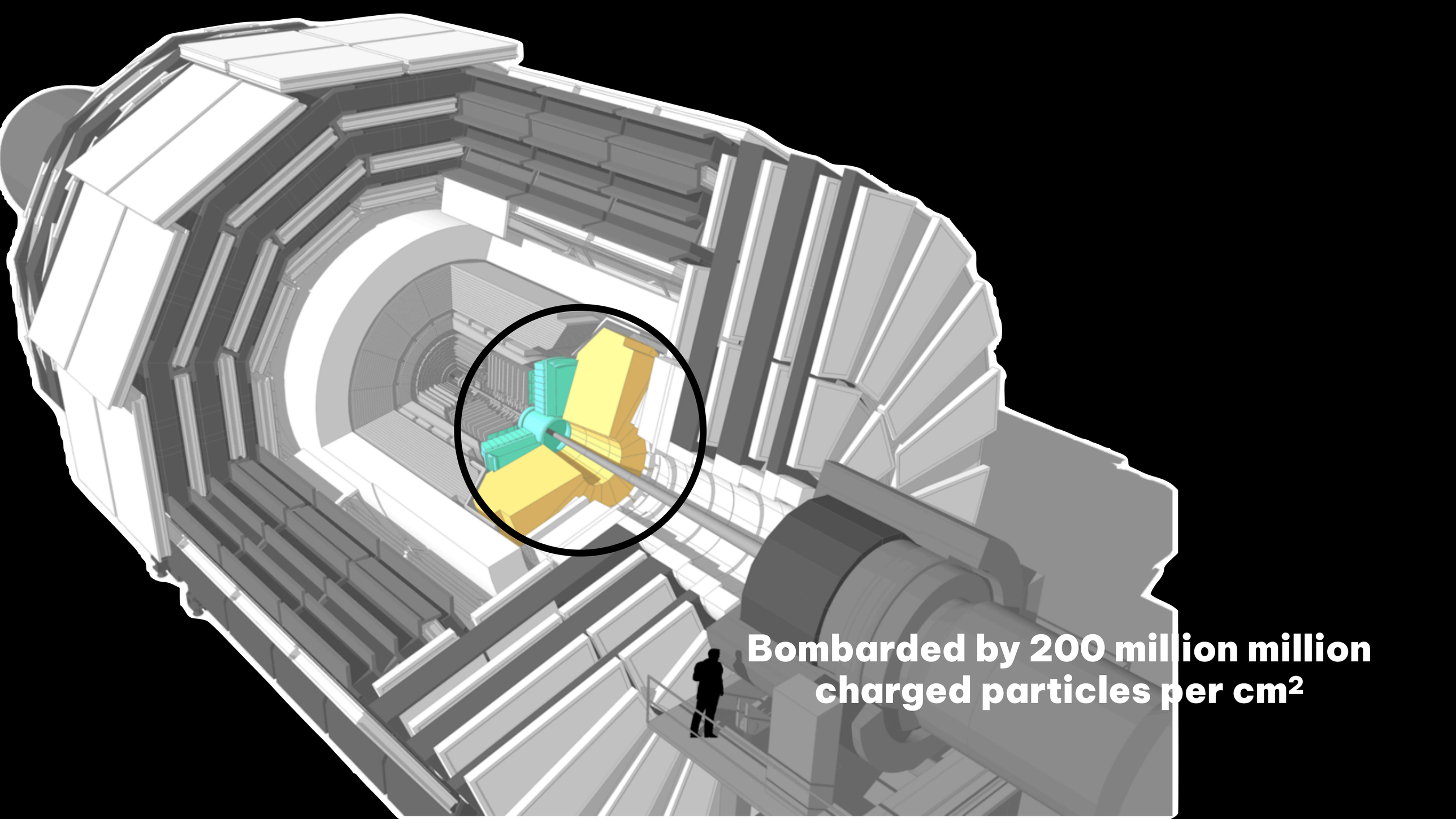
Used in CMS trigger!

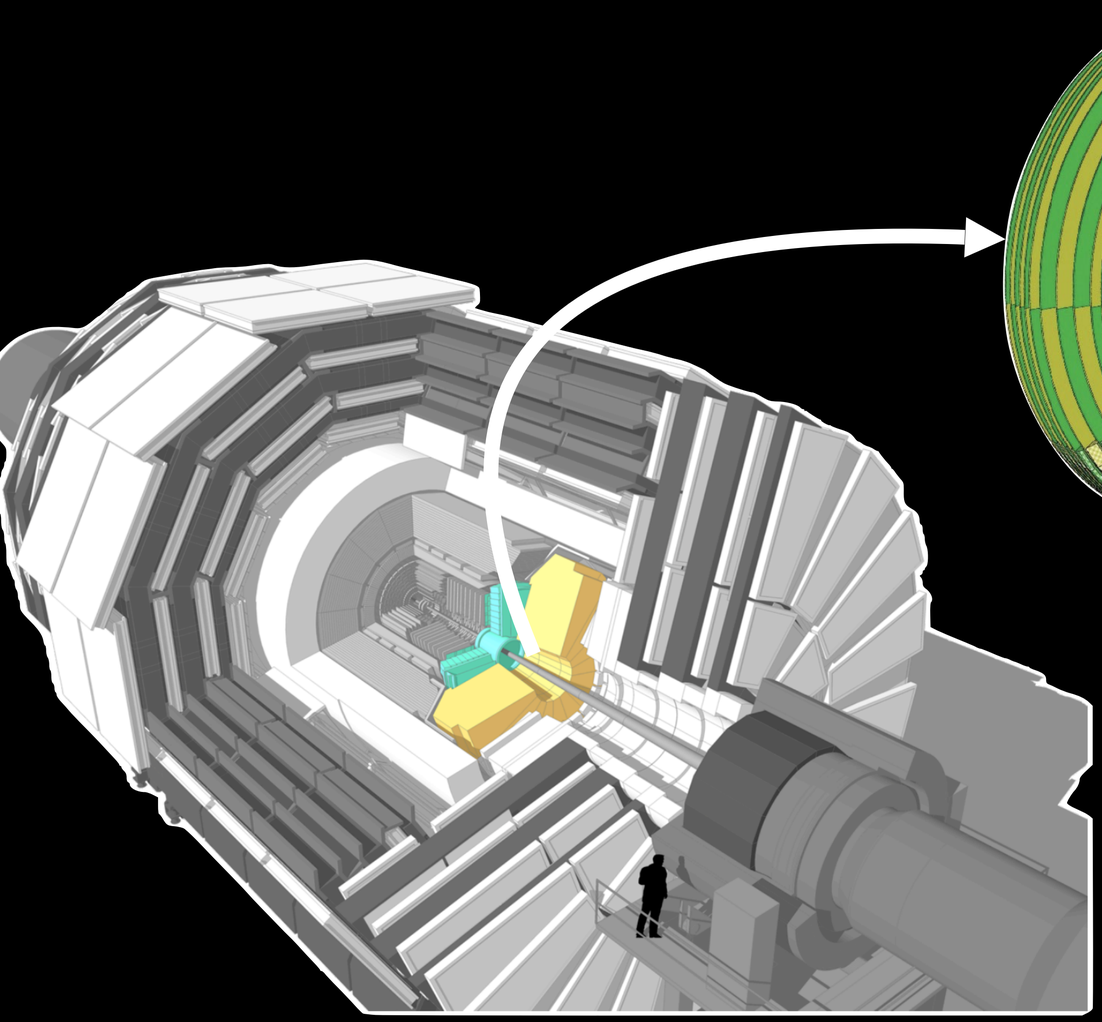


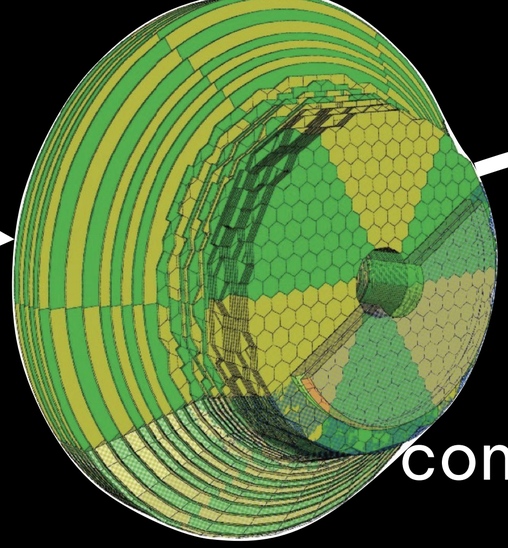
Knowledge distillation: input to anomaly score



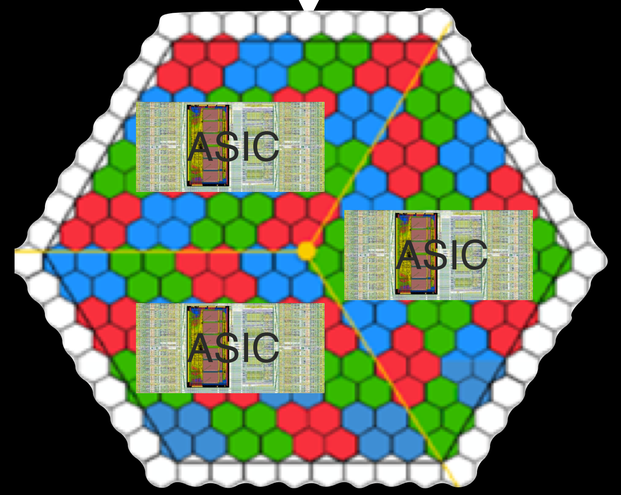

Geneva
Geneva Lake
LHCb
CMS
On-detector: ML for compression on ASICs
LHC


Bombarded by 200 million million
charged particles per cm²

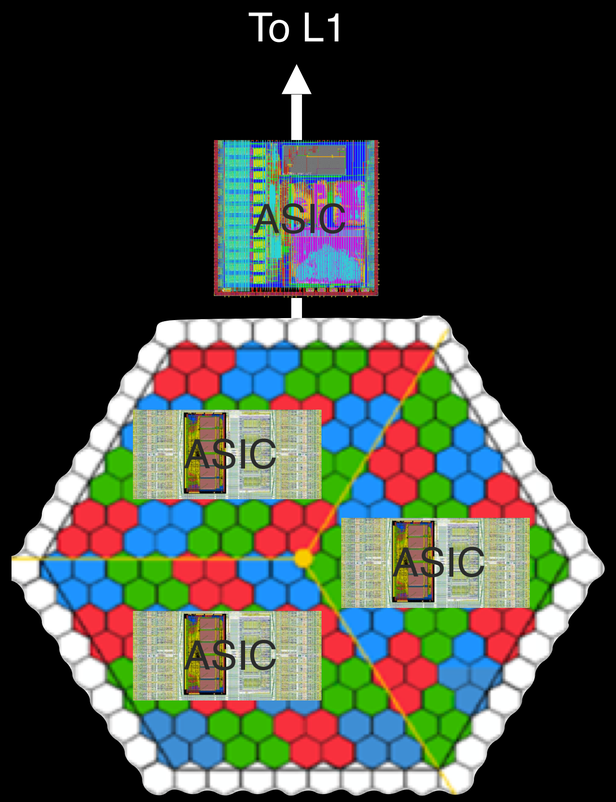




10,000
compression chips
per endcap

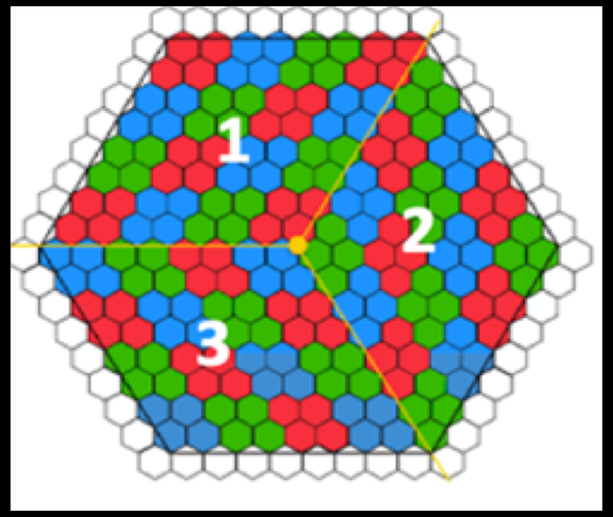


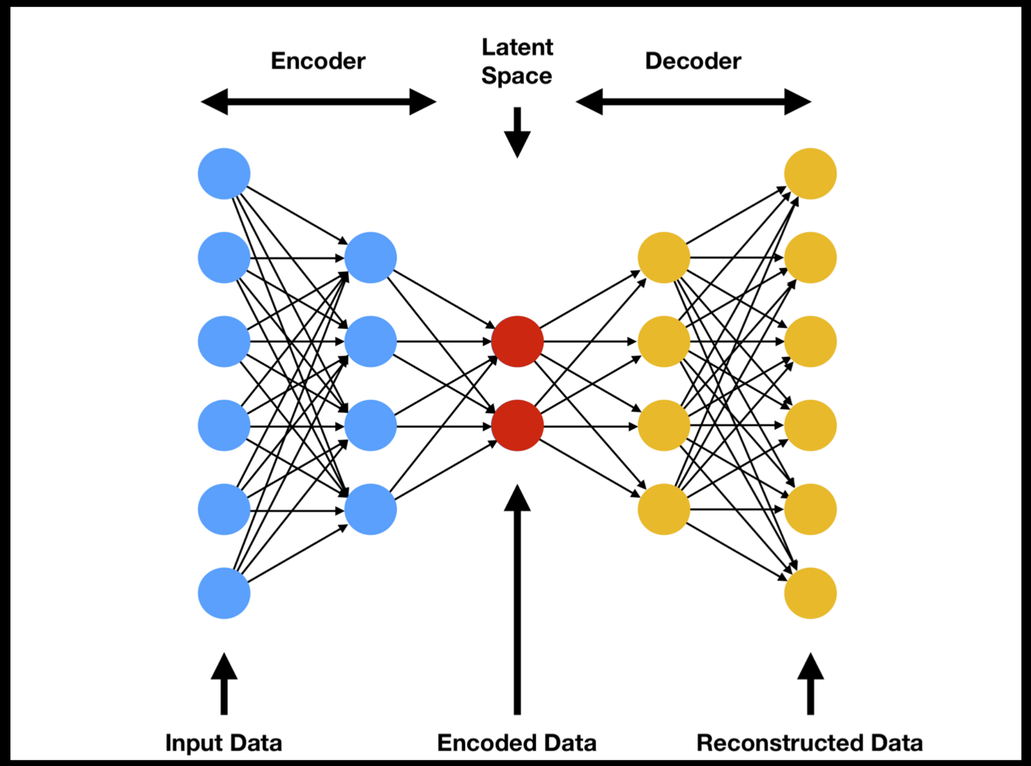
Total 6.5 million readout-channels.
Can not read out all at 40 MHz, need to compress

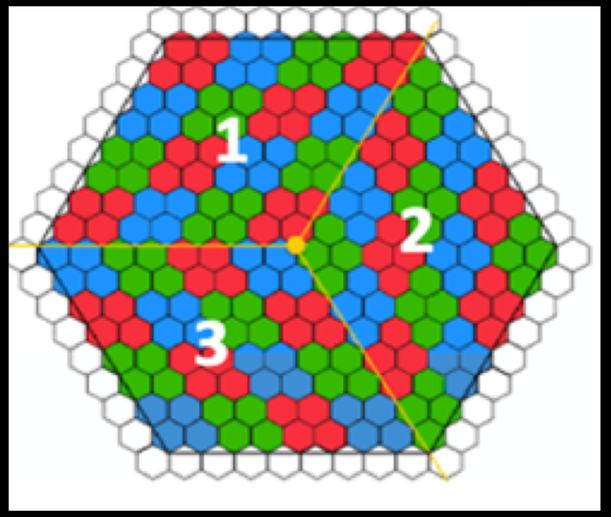


Must compress ON DETECTOR

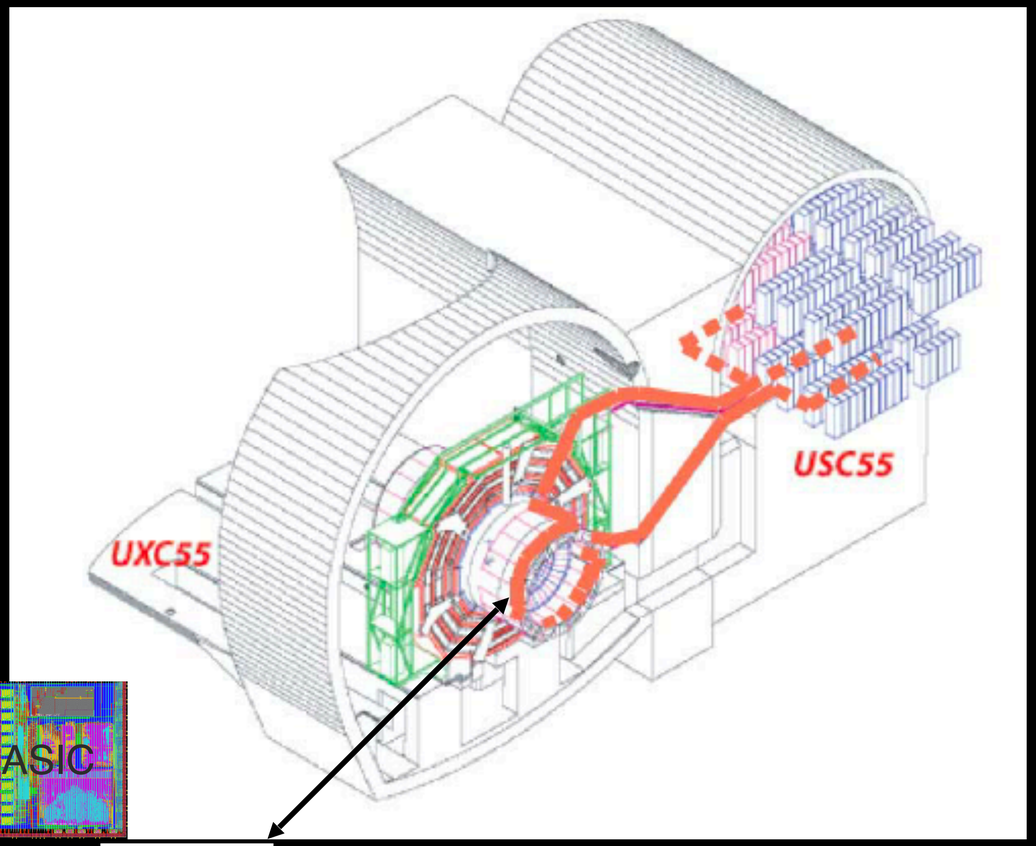
- High radiation
- Cooled to -30 degrees
- 400 ns latency budget

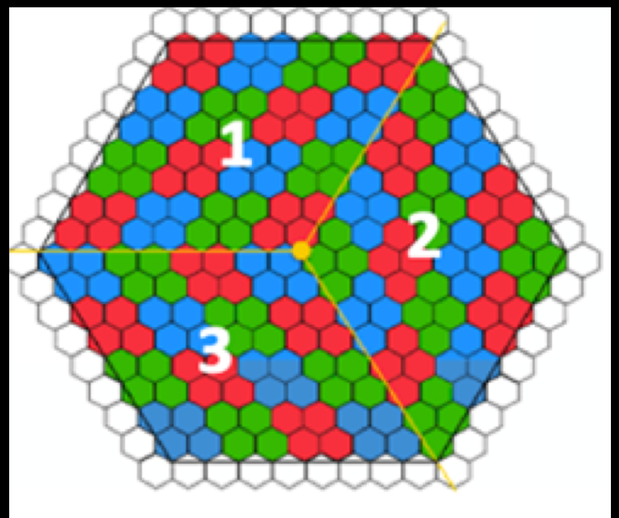




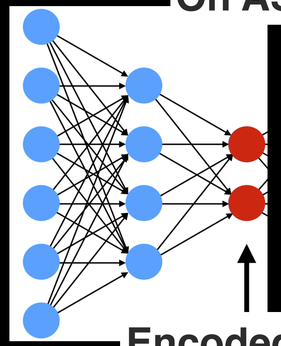


Variational Autoencoder





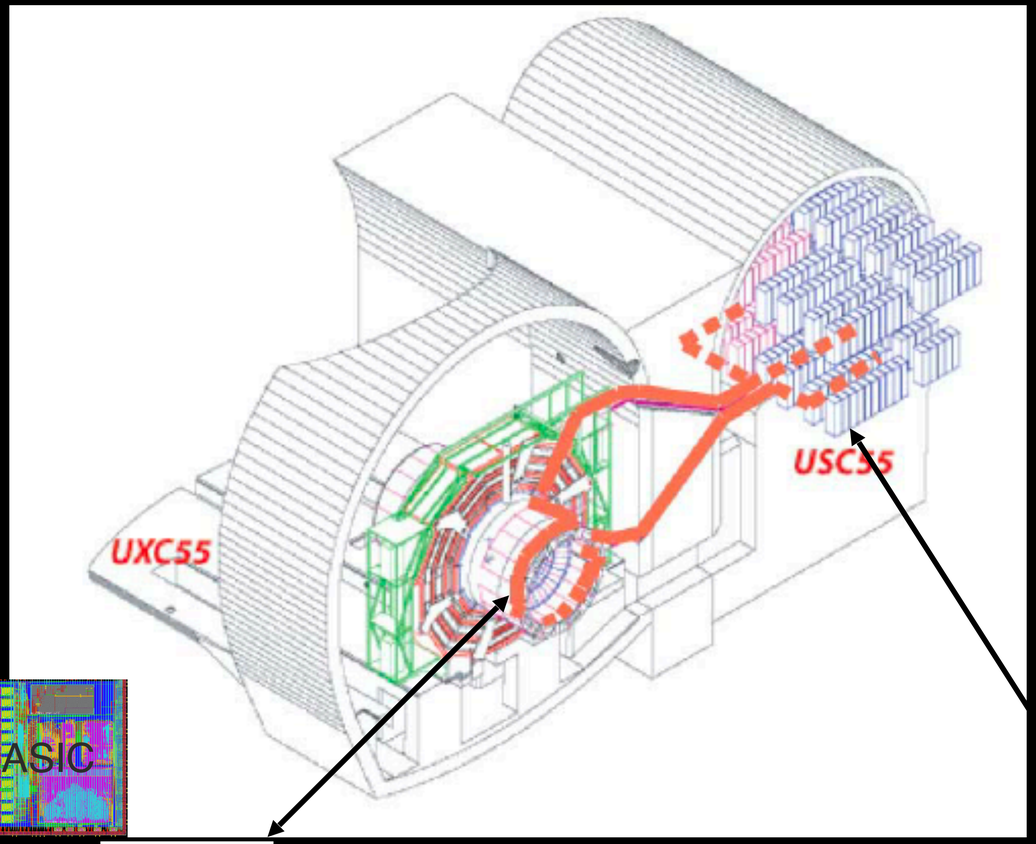
On ASIC



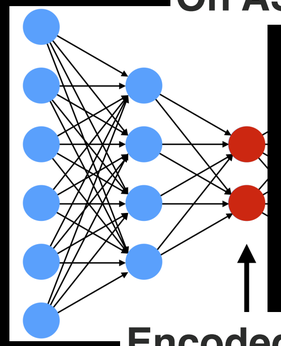
Transmit encoded data!

Encoded data

ECON-T
ECON CAE

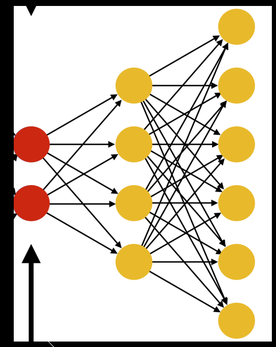


On ASIC

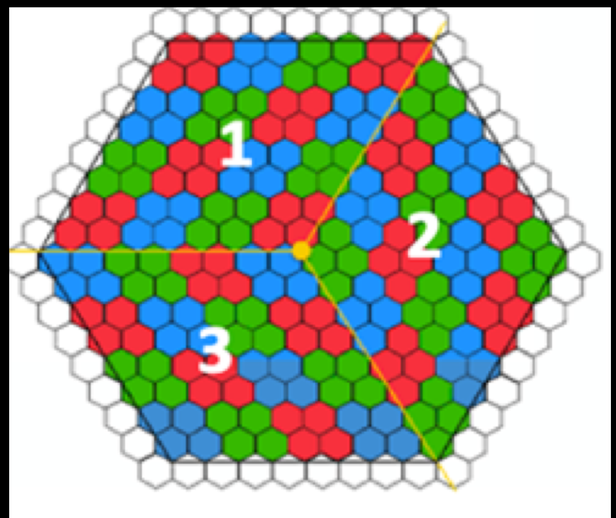


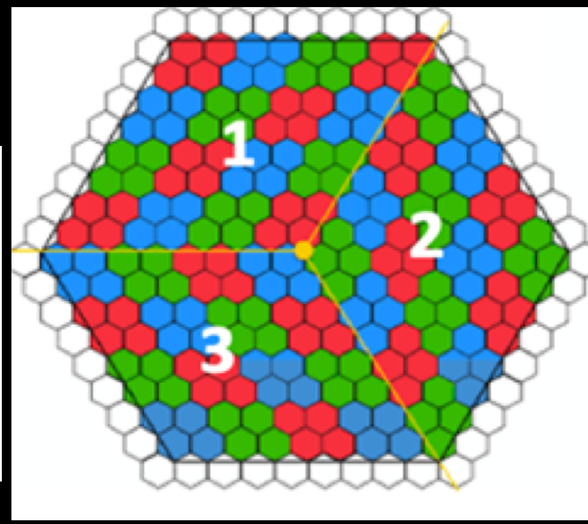
Encoded data

Transmit encoded data!

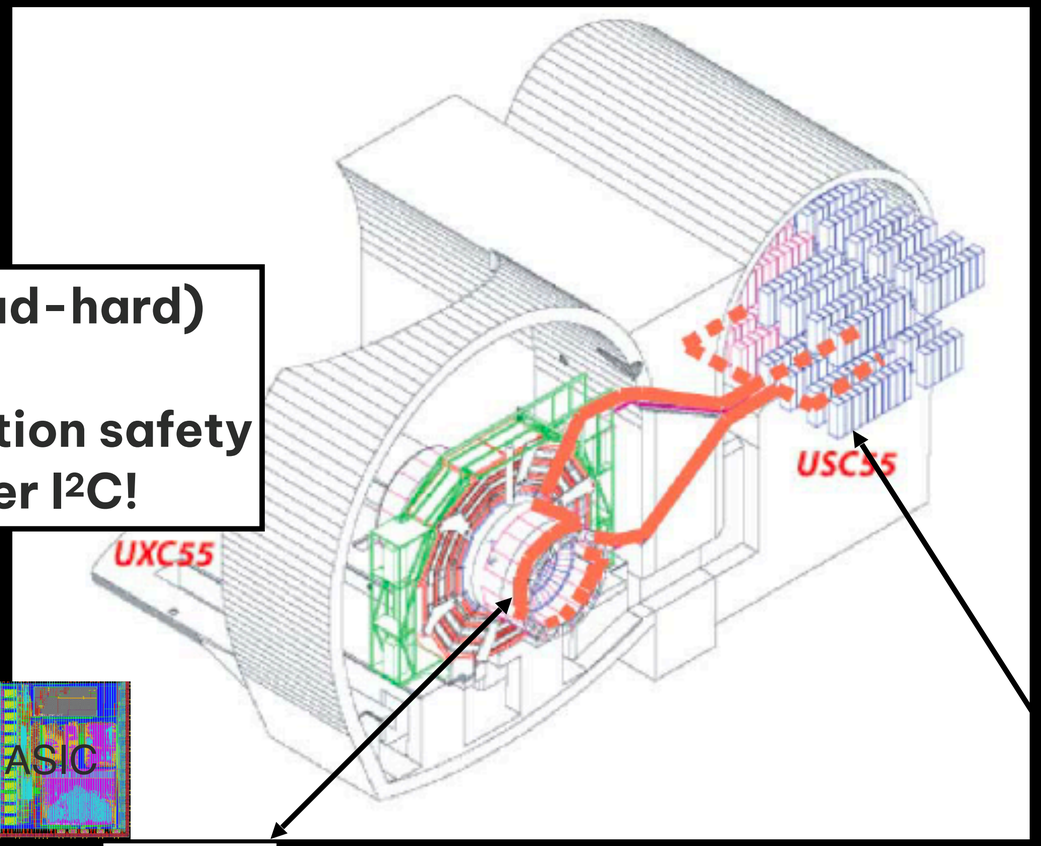


Encoded data

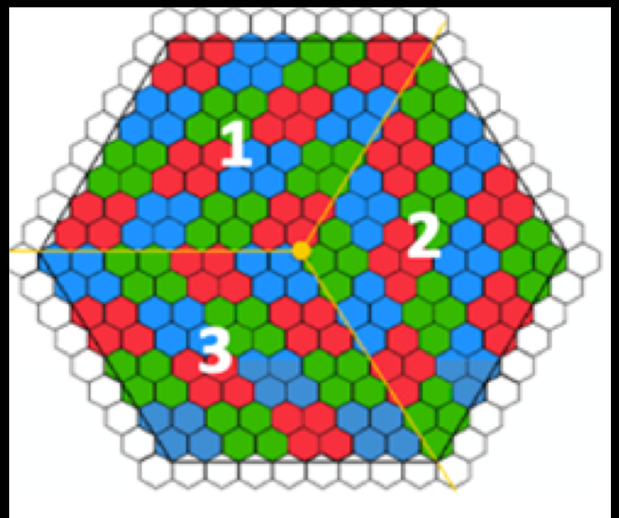


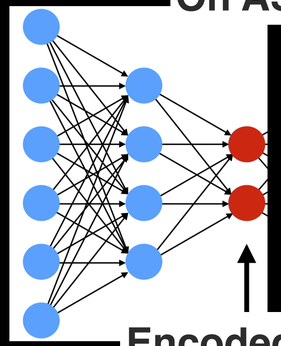


- ASIC inference (65 nm, rad-hard)
- < 4.5 mW/channel
- Triplicated w/b for radiation safety
  Reprogrammable w/b over I²C!



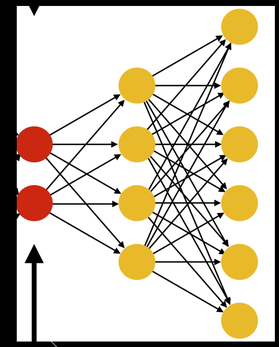
On ASIC



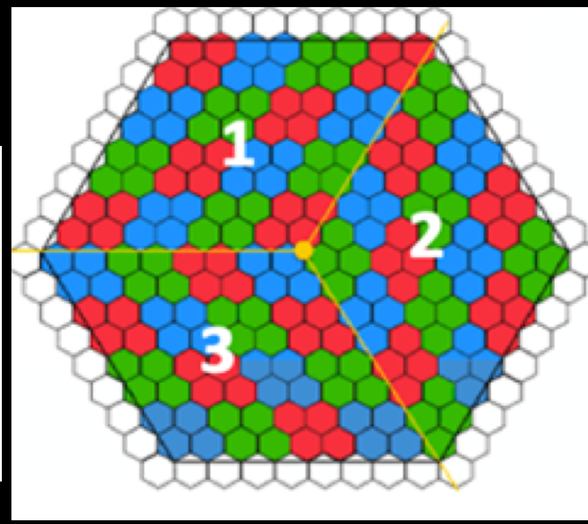


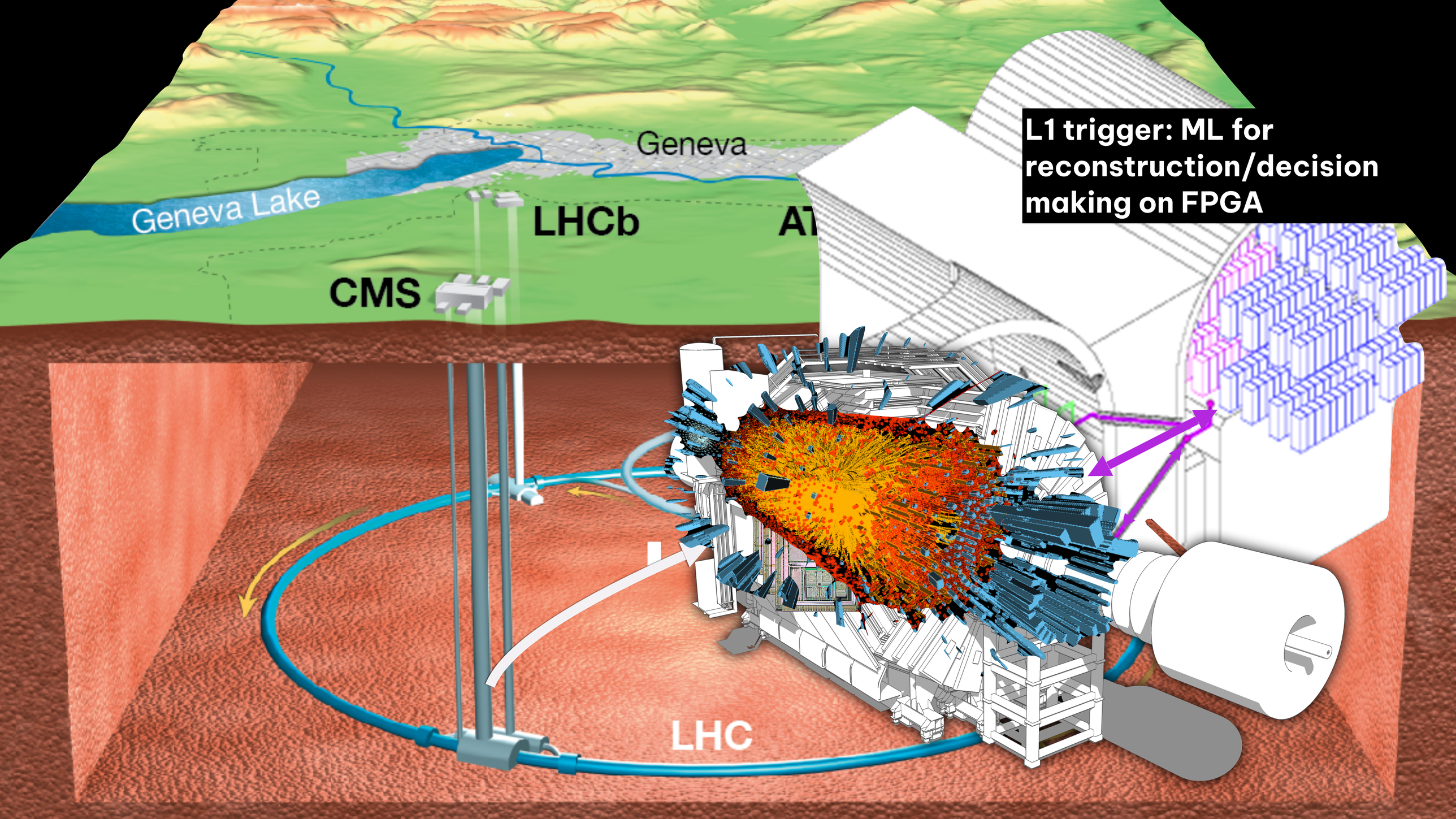
Encoded data

Transmit encoded data!



Encoded data



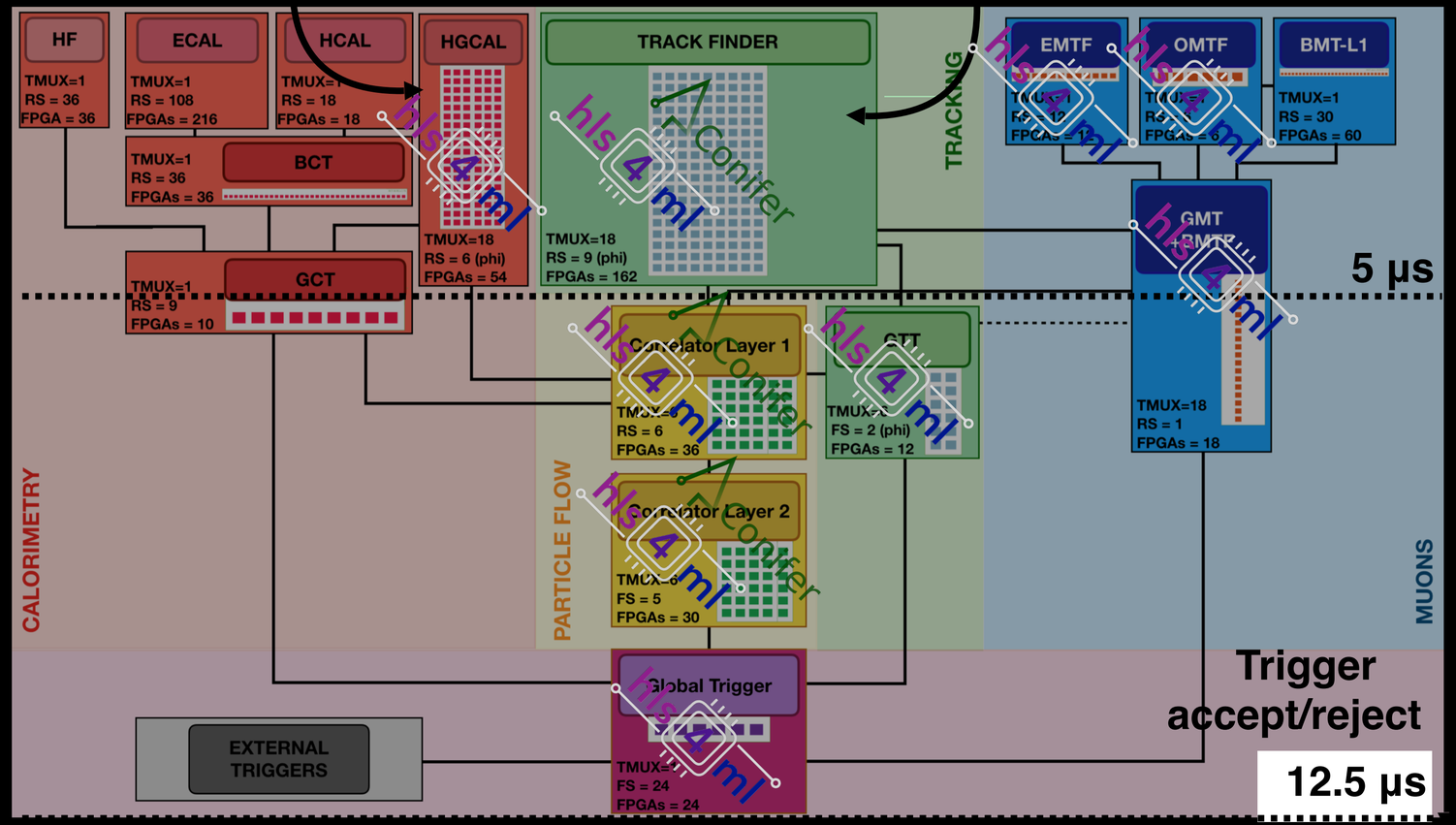

Geneva
Geneva Lake
LHCb
CMS
LHC
L1 trigger: ML for reconstruction/decision making on FPGA

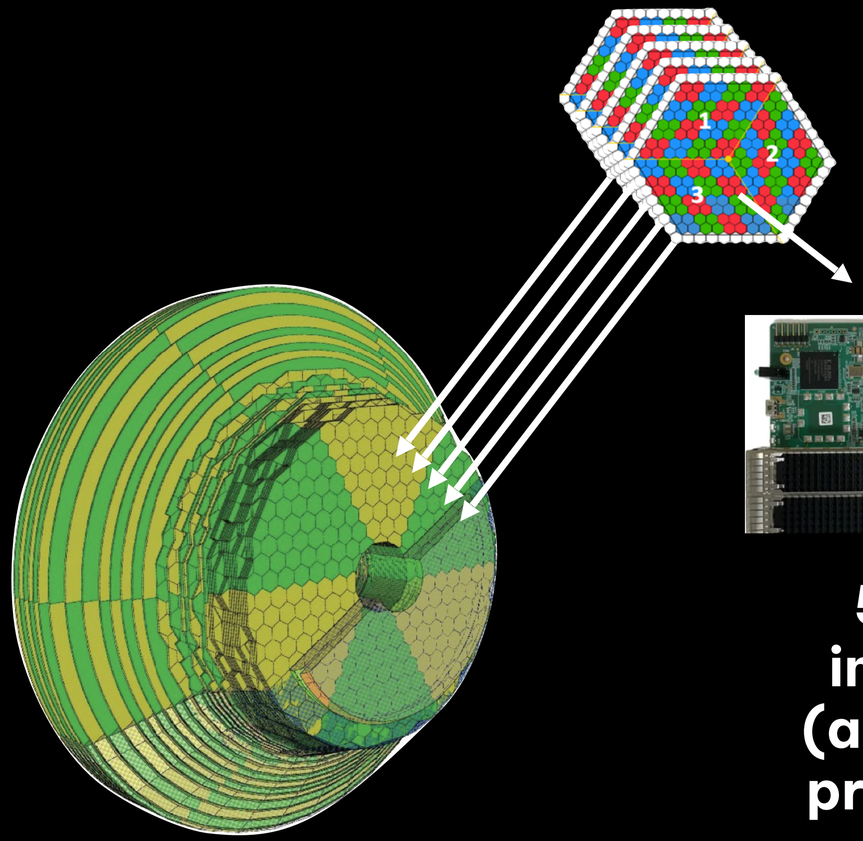
Nanosecond ML inference on FPGAs!

~40 billion inferences/s during HL-LHC

Conifer hls4ml

HEP developed libraries for fast ML on FPGAs

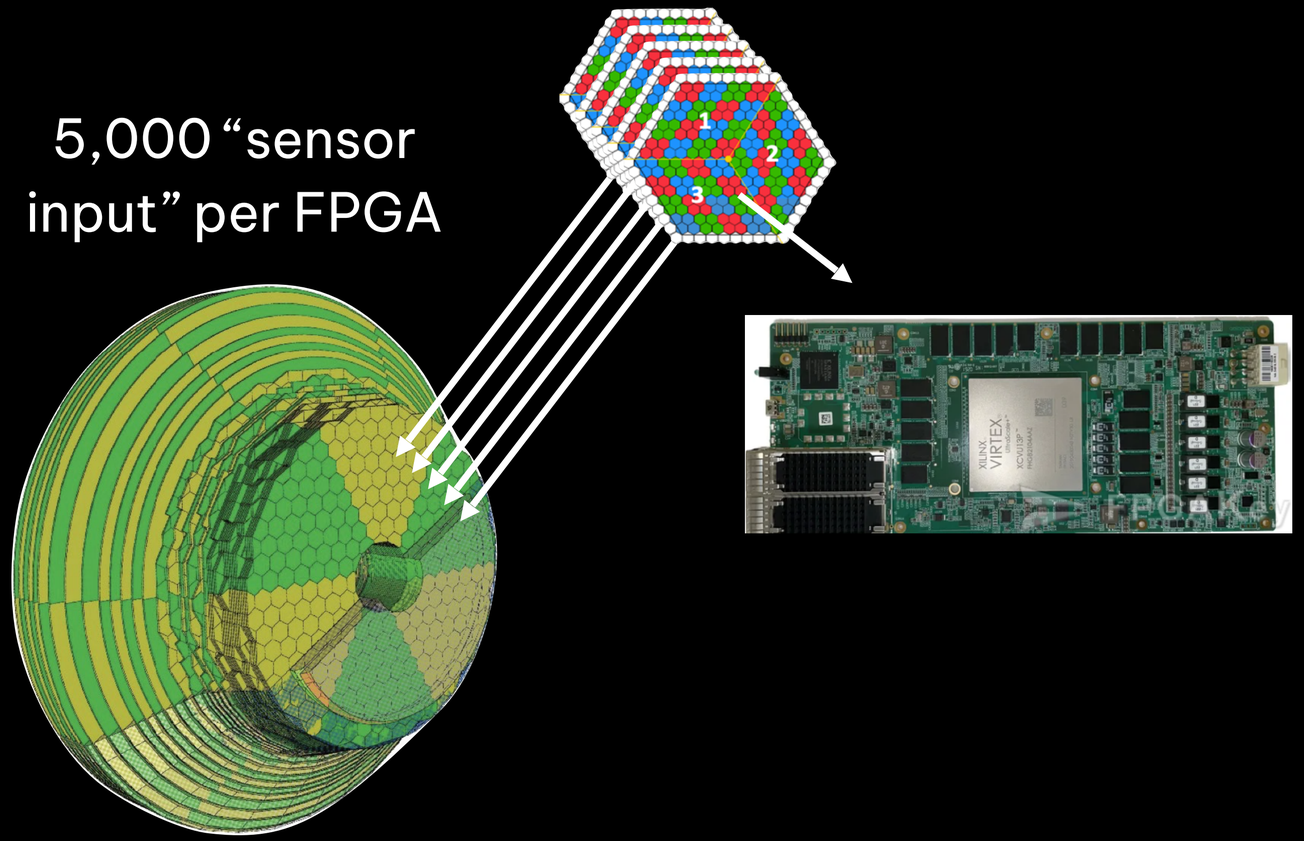




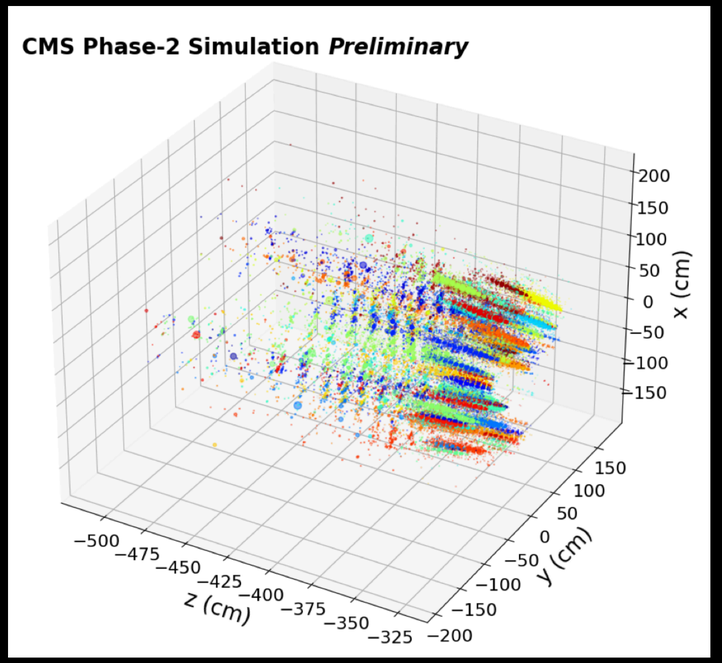
5,000 “sensor input” per FPGA (and 18 of them to process 18 events in parallel)

DOI:10.3389/fdata.2020.598927

5,000 “sensor input” per FPGA

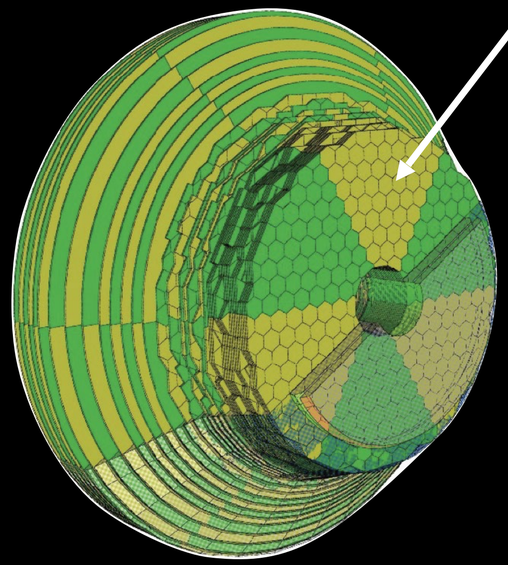


2 µs to cluster into distinct particle energy clusters!

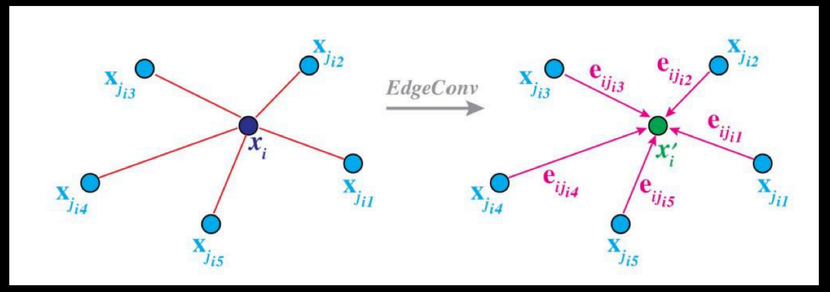


DOI:10.3389/fdata.2020.598927

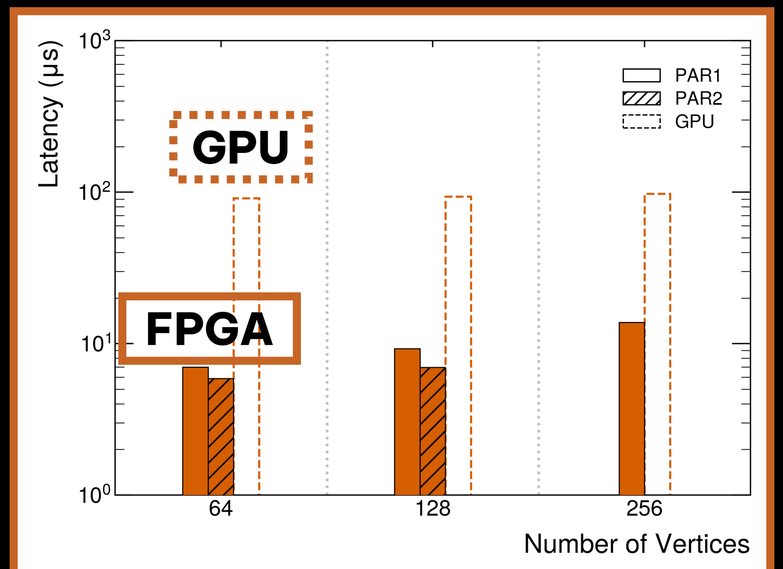
5,000 “vertices” per FPGA



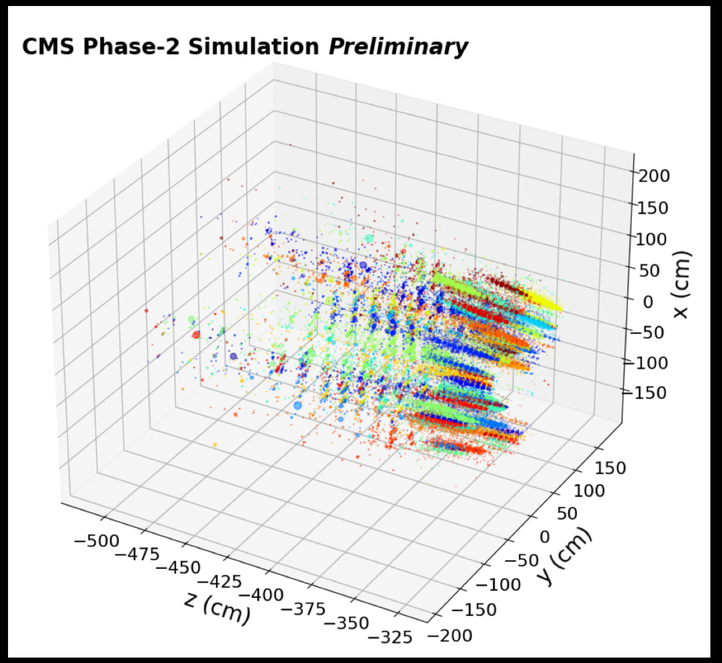
GNN with dynamic graph building is **clustering surrogate**:
Hits belonging to same object clustered



~2 µs latency constraints!

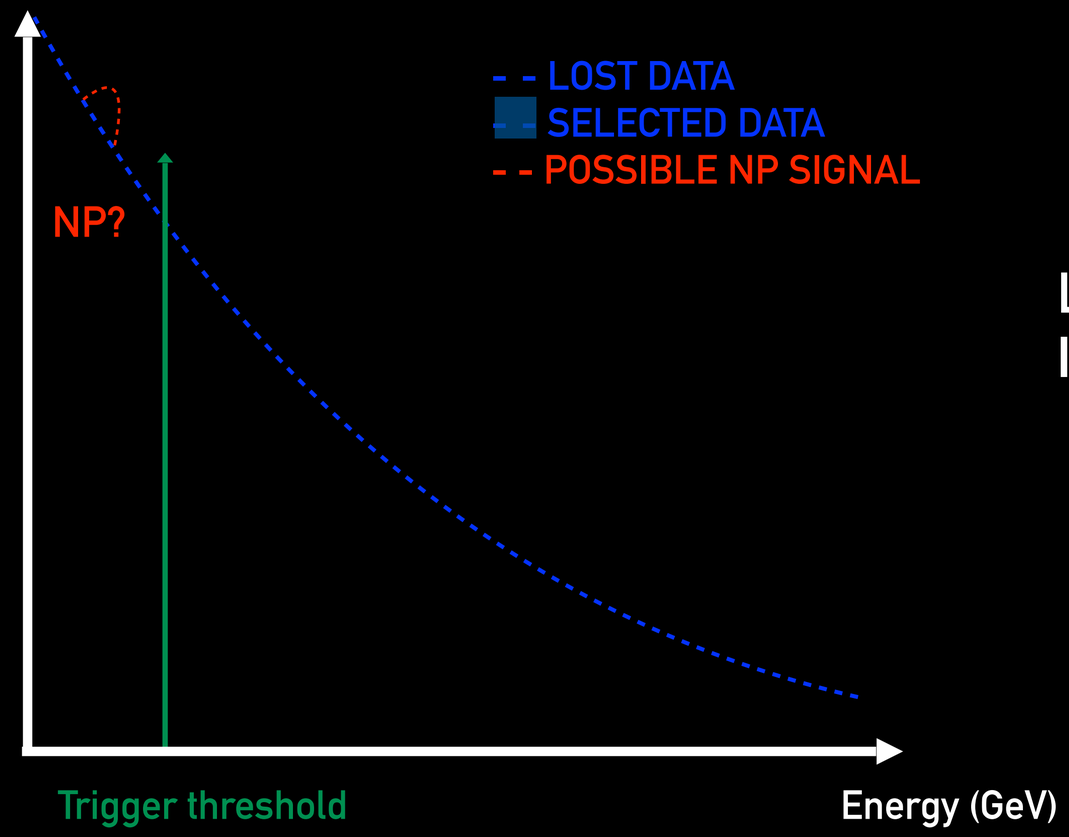


Cluster into distinct particle energy clusters



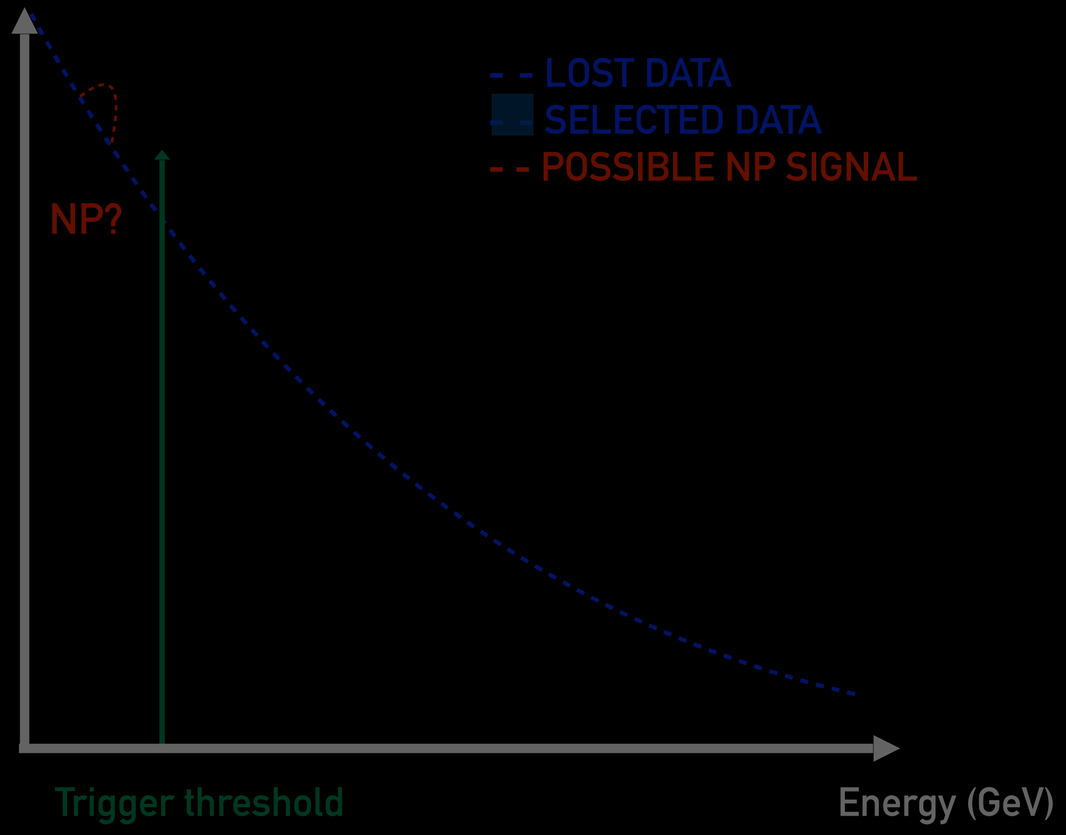
DOI:10.3389/fdata.2020.598927

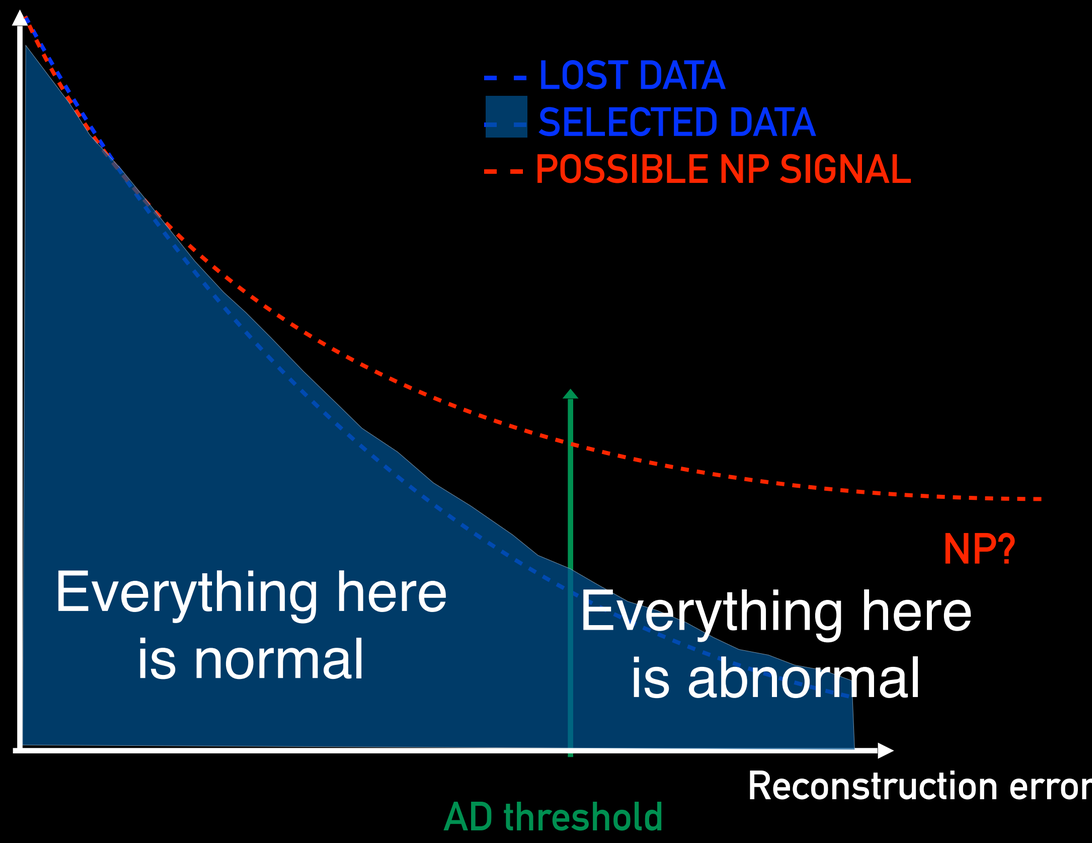
Anomaly Detection on FPGA



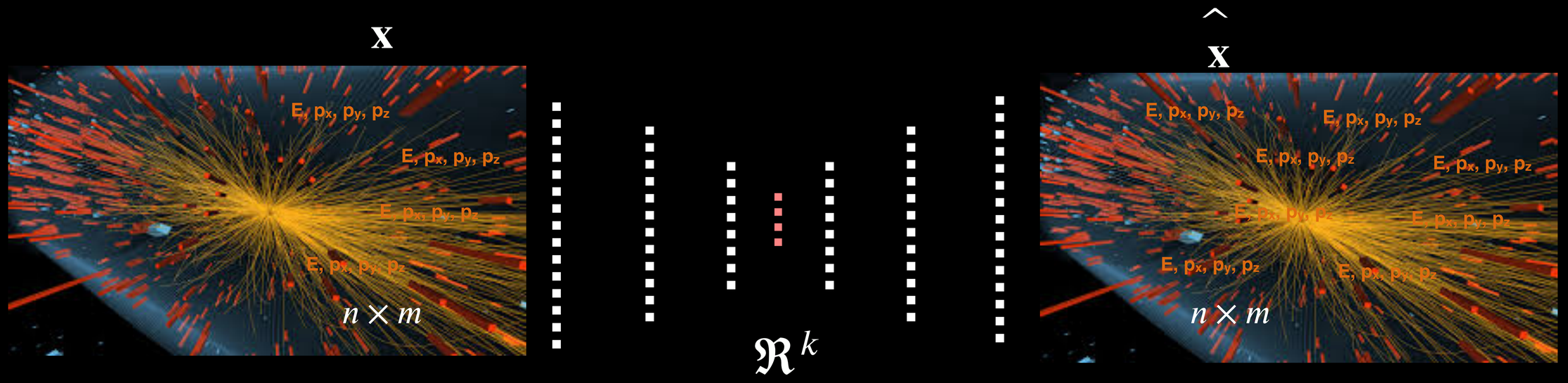
Level-1 rejects >99% of events!
Is there a smarter way to select?

Anomaly Detection triggers



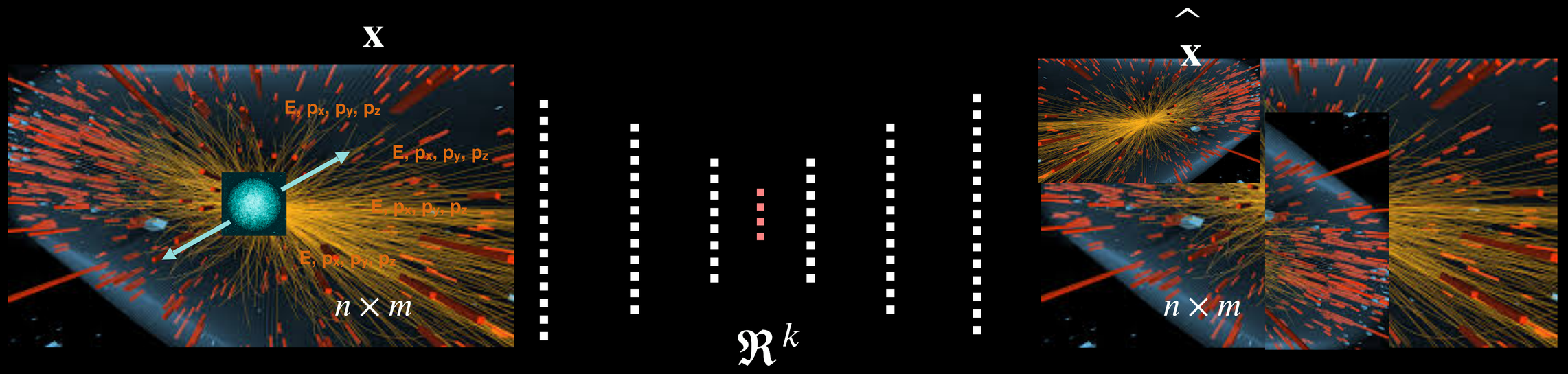


Outlier detection

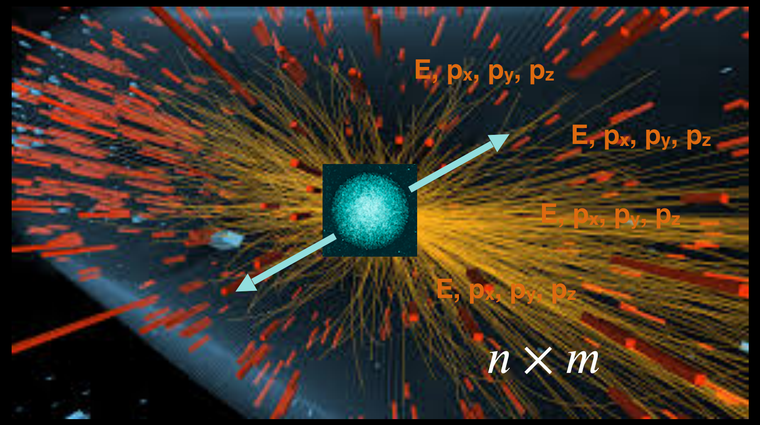


Compressed representation of x.
Latent space $\mathfrak{R}^k$, k < m×n
prevents memorisation of input, must learn

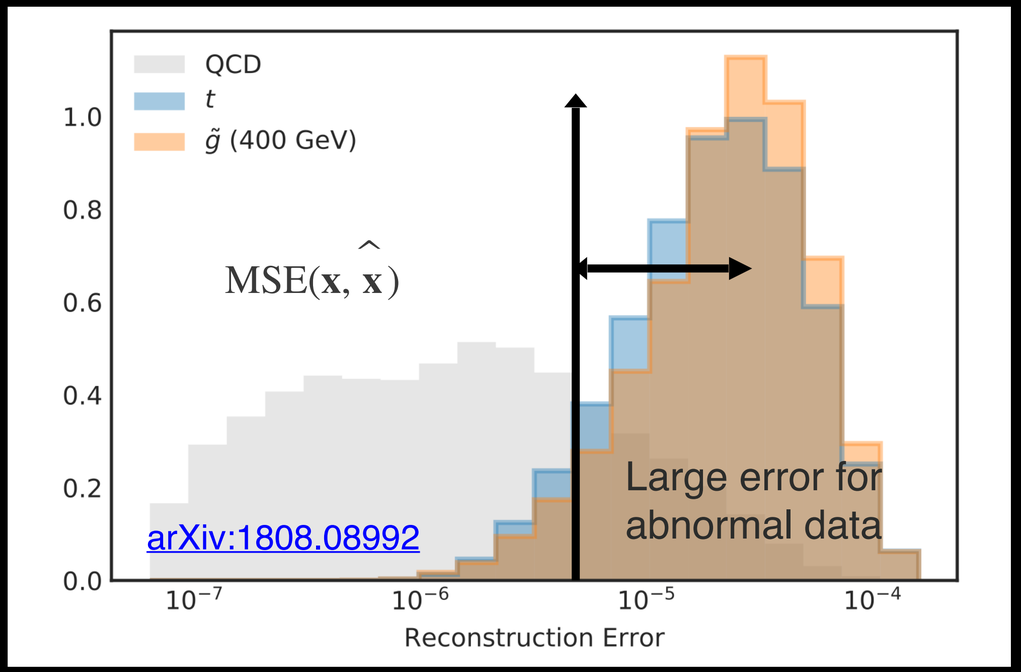
Outlier detection



$\mathscr{L}(\mathbf{x}, \hat{\mathbf{x}})$ is Mean Squared Error$(\mathbf{x}, \hat{\mathbf{x}})$, “high error events” proxy for “degree of abnormality”


E, px, py, pz
E, px, py, pz
E, px, py, pz
E, px, py, pz
n × m

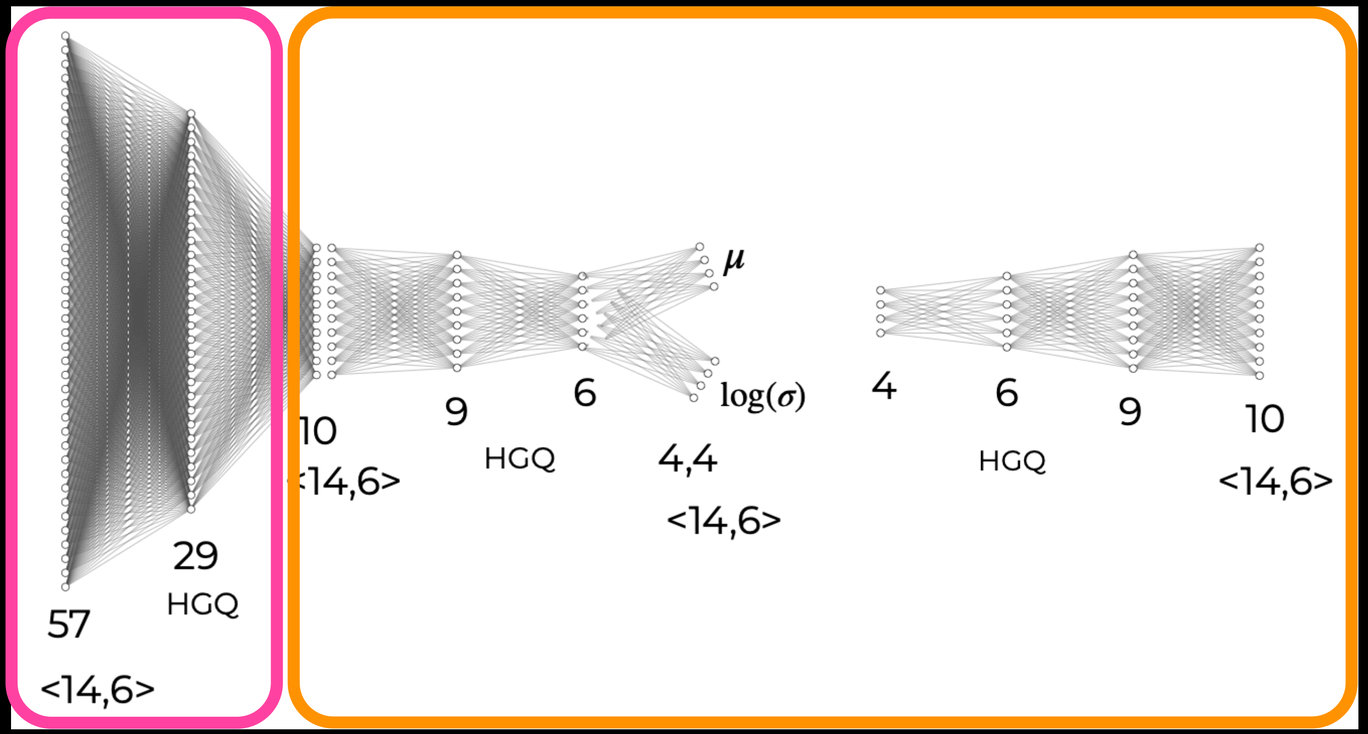
$\mathfrak{R}^{k}$


QCD
t
g̃ (400 GeV)
MSE(x, x̂)
1.0
0.8
0.6
0.4
0.2
0.0
10^−7
10^−6
10^−5
10^−4
Reconstruction Error
arXiv:1808.08992
Large error for
abnormal data

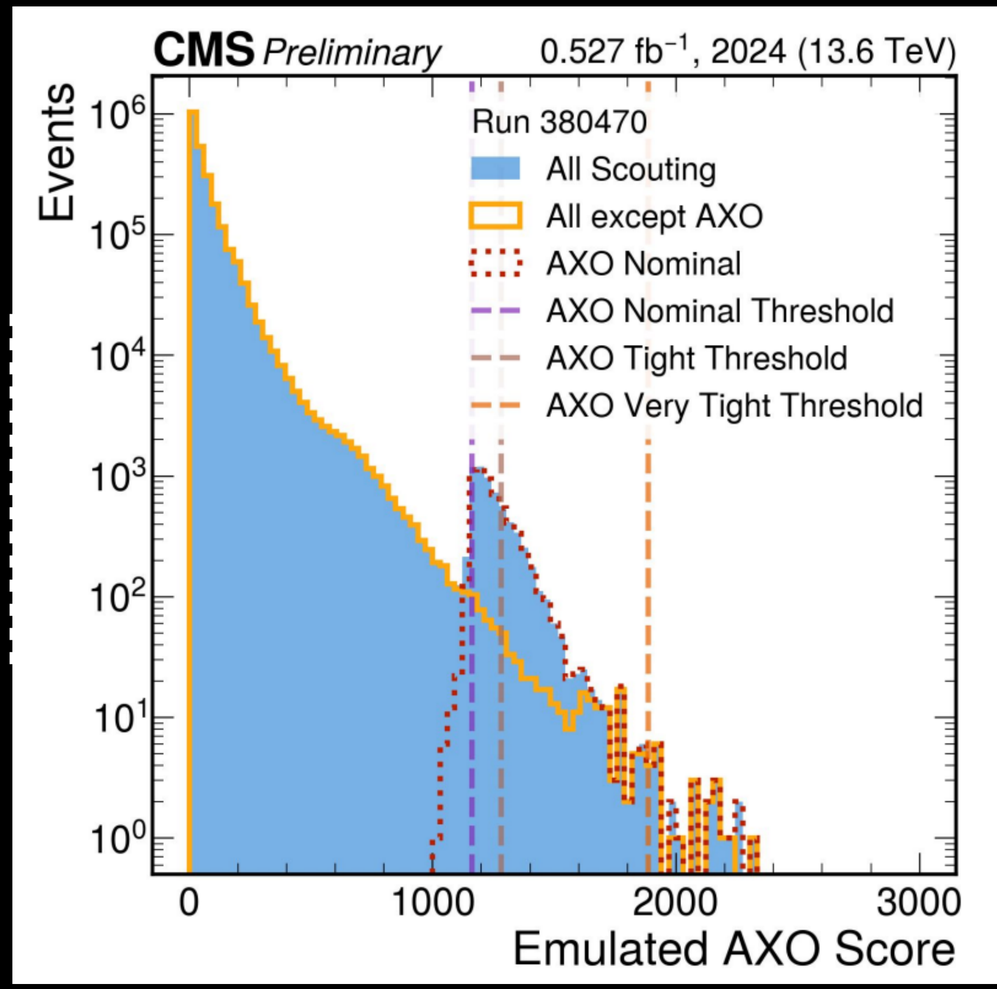
2411.19506

AXOL1TL

Neural embedding with VicReg contrastive learning

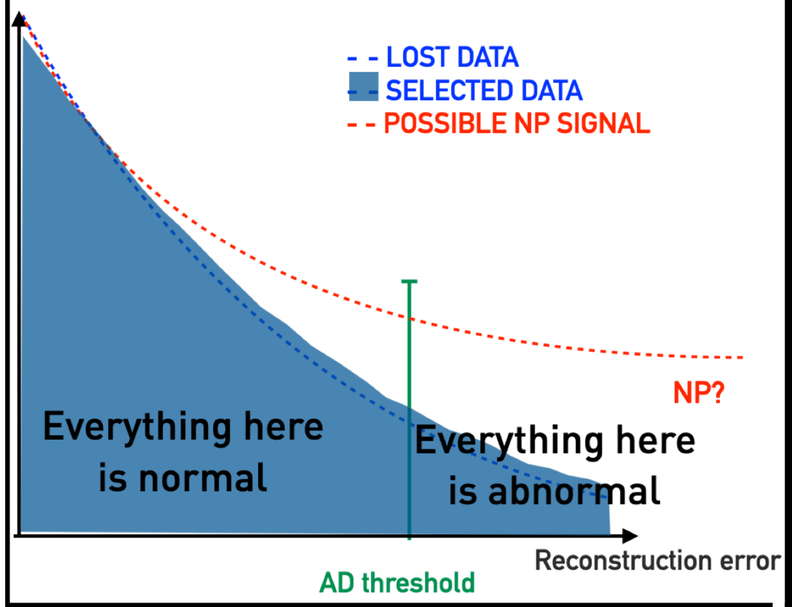


Variational Autoencoder



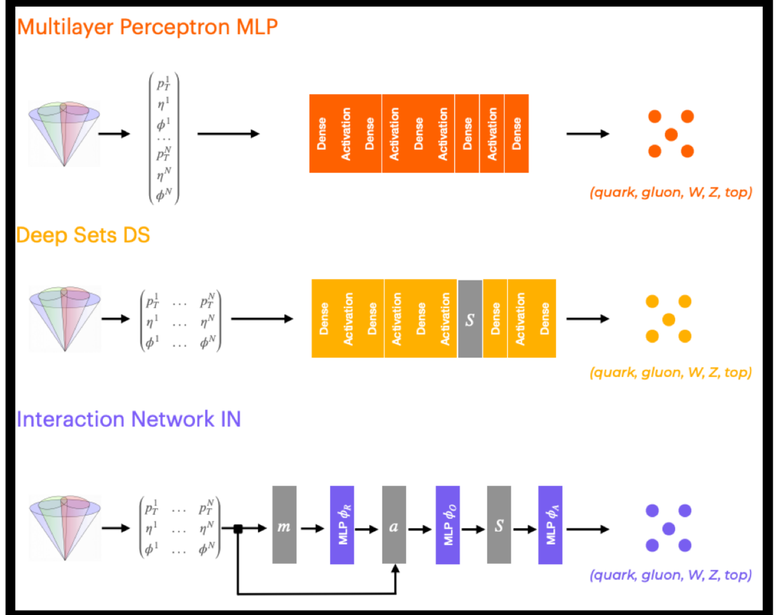
8 layers in 50 ns on FPGA,
~1% of total rate selected by AD

Anomaly detection with VAEs in 50 ns



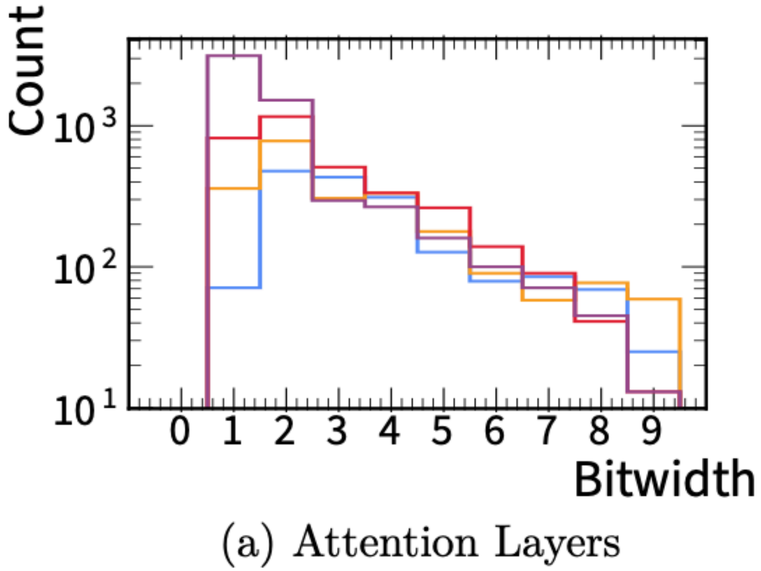
CMS DP2023_079
E. Govorkova et al (2022)

Quantised Interaction Networks and Deep Sets in <160 ns



P. Odagiu et al. 2024

Fully heterogeneously quantized transformers in <100 ns

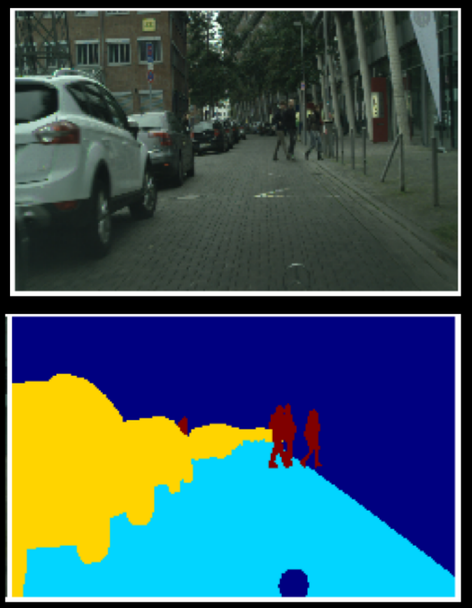


(a) Attention Layers

arxiv:2510.24784

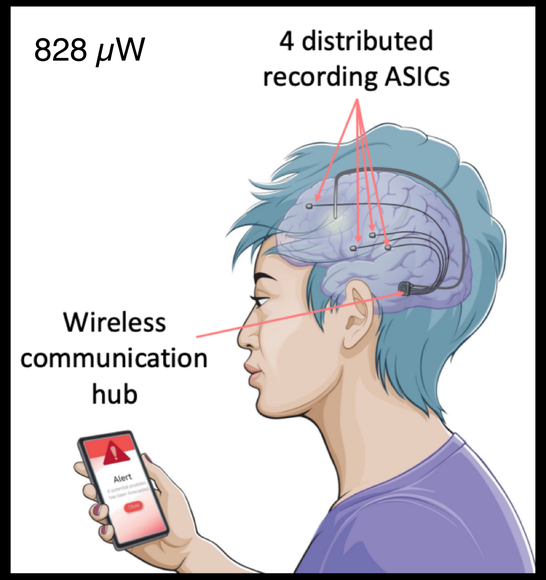
...and outside

Semantic segmentation for autonomous vehicles



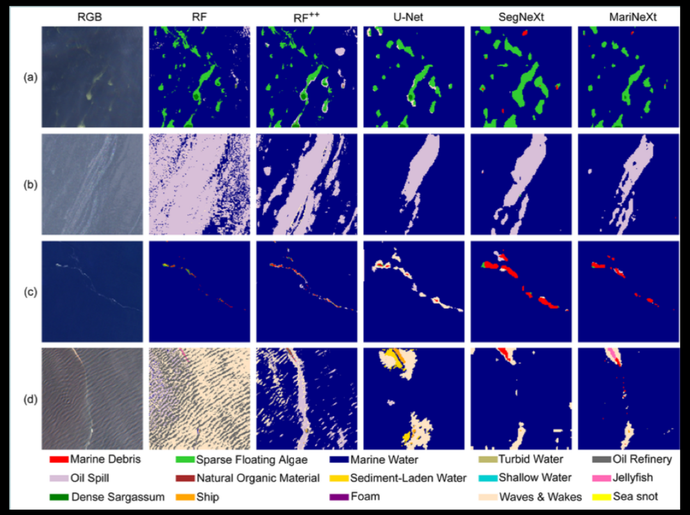
N. Ghielmetti et al. 2022

Seizure Predicting Brain Implant



W. Lemaire et al. 2022

Earth monitoring in satellites



Edge SpAIce, S. Summers

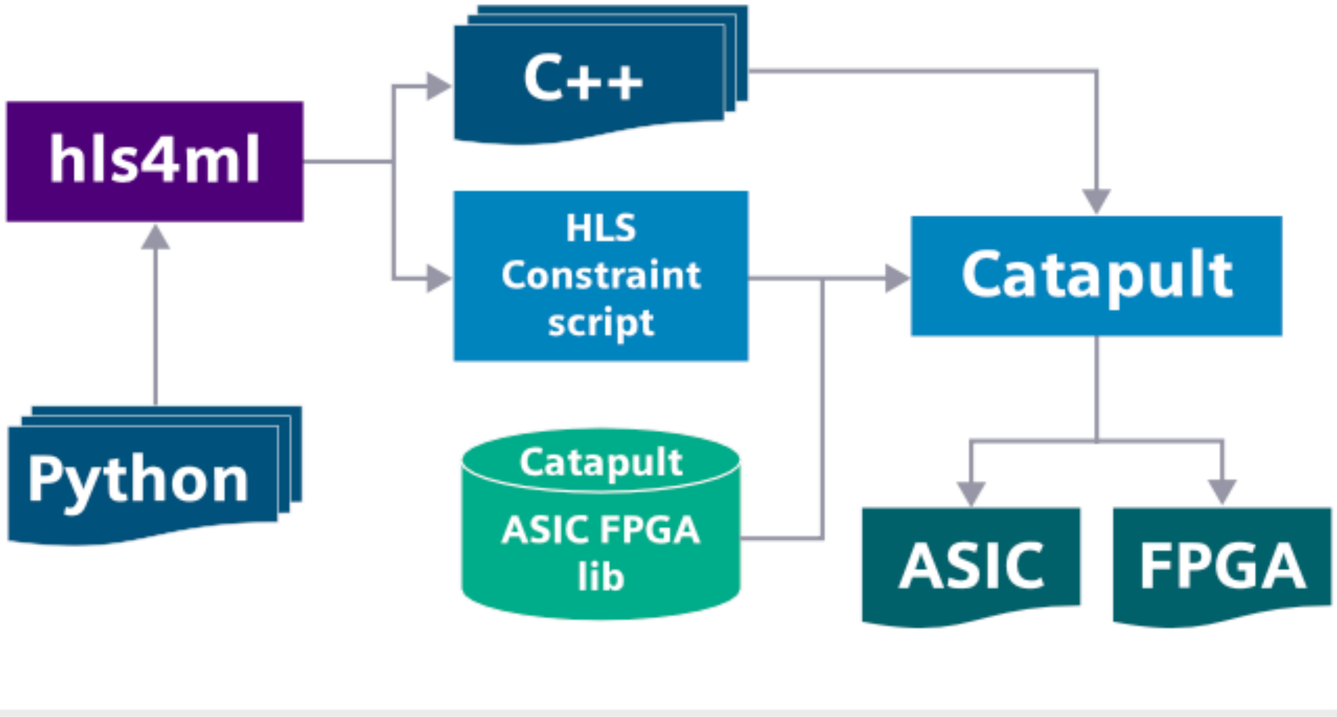
- MLPerf tinyML benchmarking
- For fusion science phase/mode monitoring
- Crystal structure detection
- Triggering in DUNE
- Accelerator control
- Magnet Quench Detection
- Food contamination detection
- Quantum control etc....

https://news.siemens.com/en-us/siemens-catapult-ai-nn/

PRESS RELEASE

Siemens simplifies development of AI accelerators for advanced system-on-chip designs with Catapult AI NN

May 21, 2024
Plano, Texas

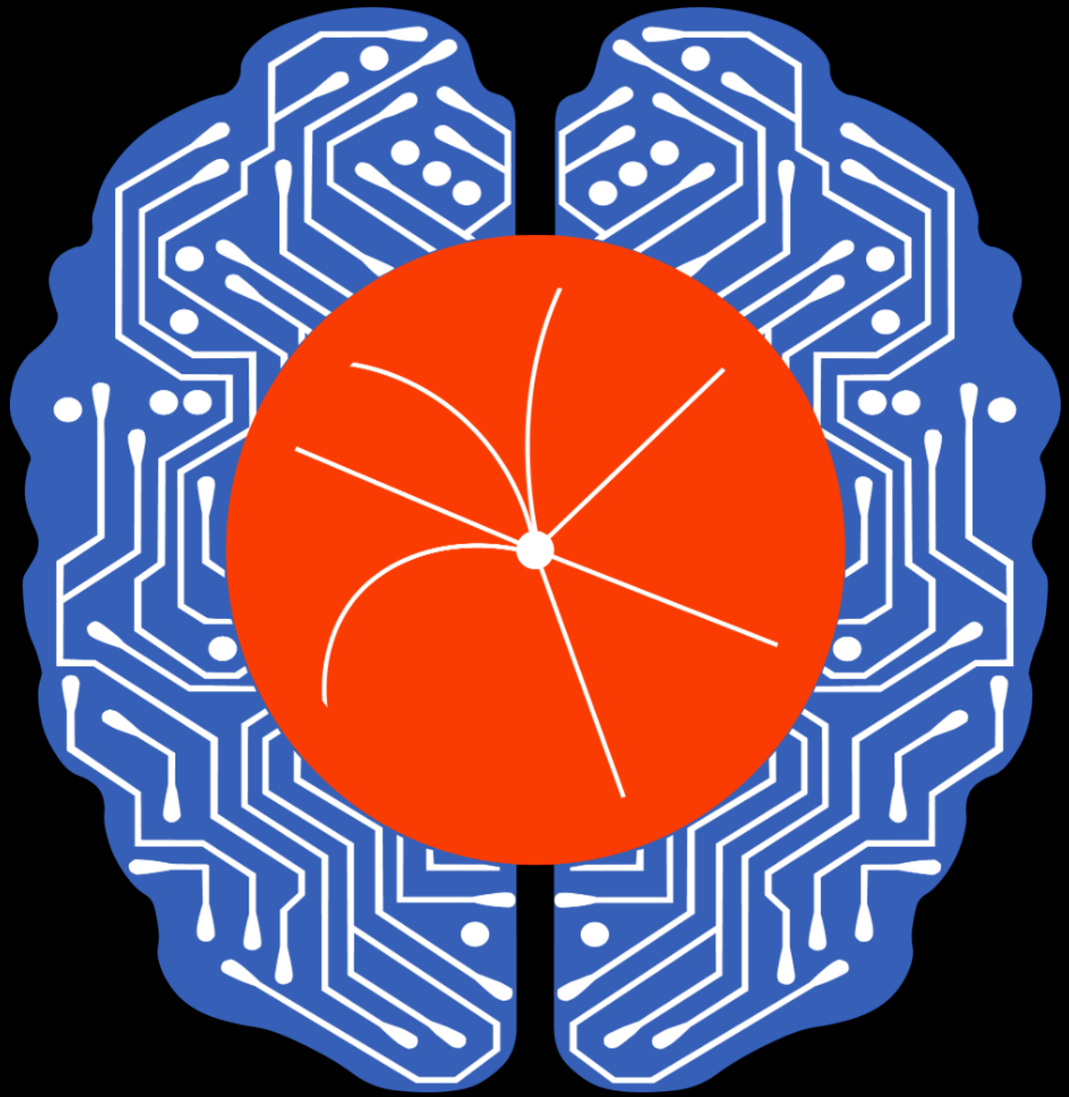


“...brings together hls4ml and Siemens' Catapult™ HLS software...

... addresses the unique requirements of
ML accelerator design for power, performance, and area **on custom silicon"**

fast machine learning foundation

fastmachinelearning.org



Collaborating Institutes

- Argonne National Laboratory
- CERN
- California Institute of Technology
- Carnegie Mellon University
- ETH Zurich
- Fermi National Accelerator Laboratory
- Georgia Institute of Technology
- HEPIA - Haute école du paysage, d'ingénierie et d'architecture
- Imperial College London
- Jefferson Lab
- Jožef Stefan Institute
- KIT
- LPNHE, CNRS/IN2P3, France
- Los Alamos National Laboratory
- Massachusetts Institute of Technology
- National Yang Ming Chiao Tung University
- Northwestern University
- Norwegian University of Science and Technology
- Pacific Northwest National Laboratory
- Princeton University
- Purdue University
- SLAC
- Southern Methodist University
- Technical University of Denmark
- Tohoku University
- UC San Diego
- UCL
- University of Bristol
- University of Chicago
- University of Colorado Boulder
- University of Geneva
- University of Illinois at Urbana-Champaign
- University of Michigan
- University of Pennsylvania
- University of Washington
- University of Wisconsin-Madison
- Université de Sherbrooke

Fast Machine Learning for Science

August 31 – September 4, 2026

indi.to/fastml26

UC San Diego

The Fast ML community

https://fastmachinelearning.org/

Join our Slack to get updates!