IMPERIAL

High-speed data processing in particle physics experiments

Alex Tapper

Altera, 15th December 2025

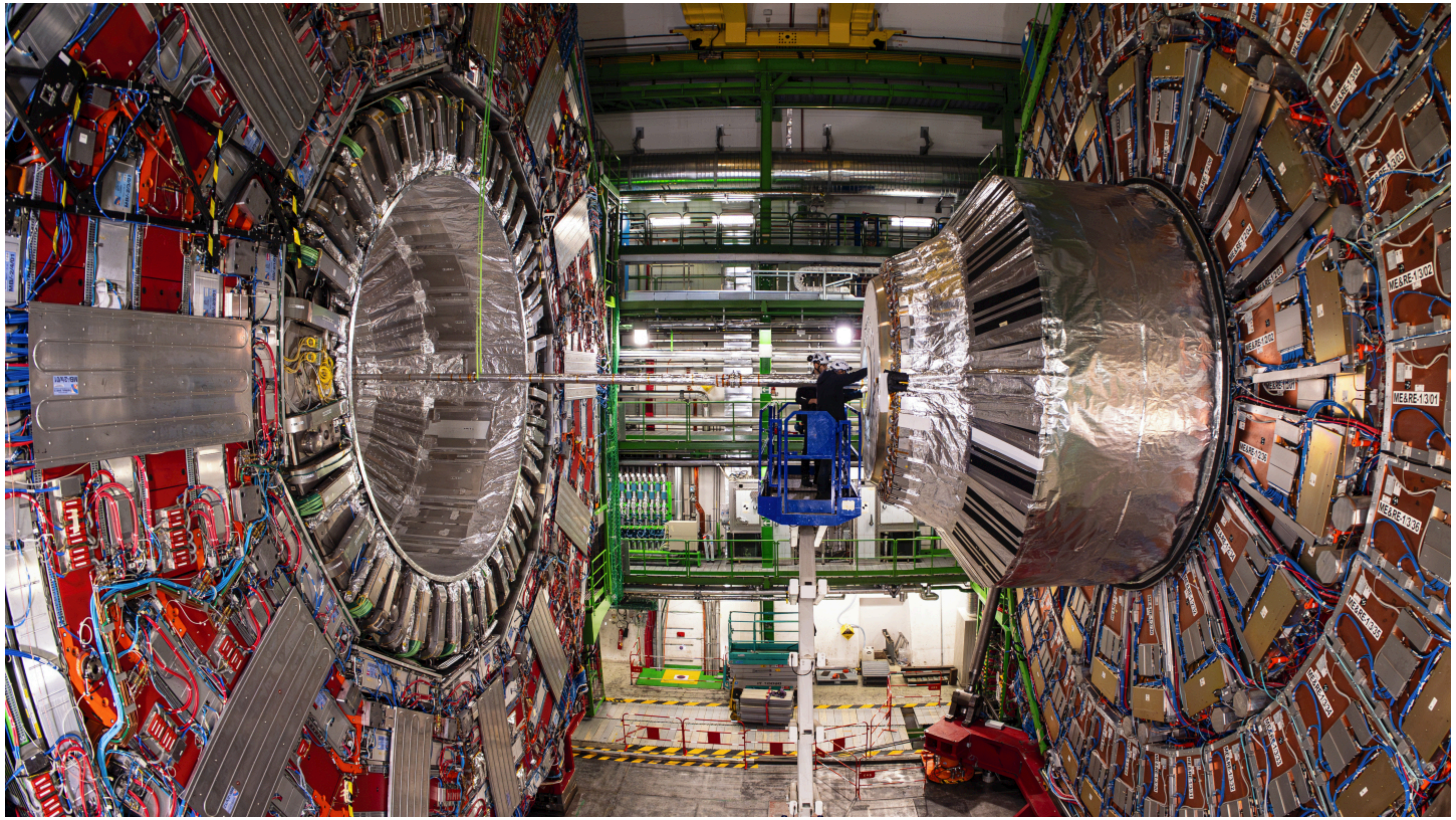
About me ..

- Particle physicist
  - Oxford → Imperial → Hamburg → Geneva ↔ Chicago
  - Build, operate and analyse data from large experiments in huge collaborations
  - Data acquisition and realtime data filtering/analysis

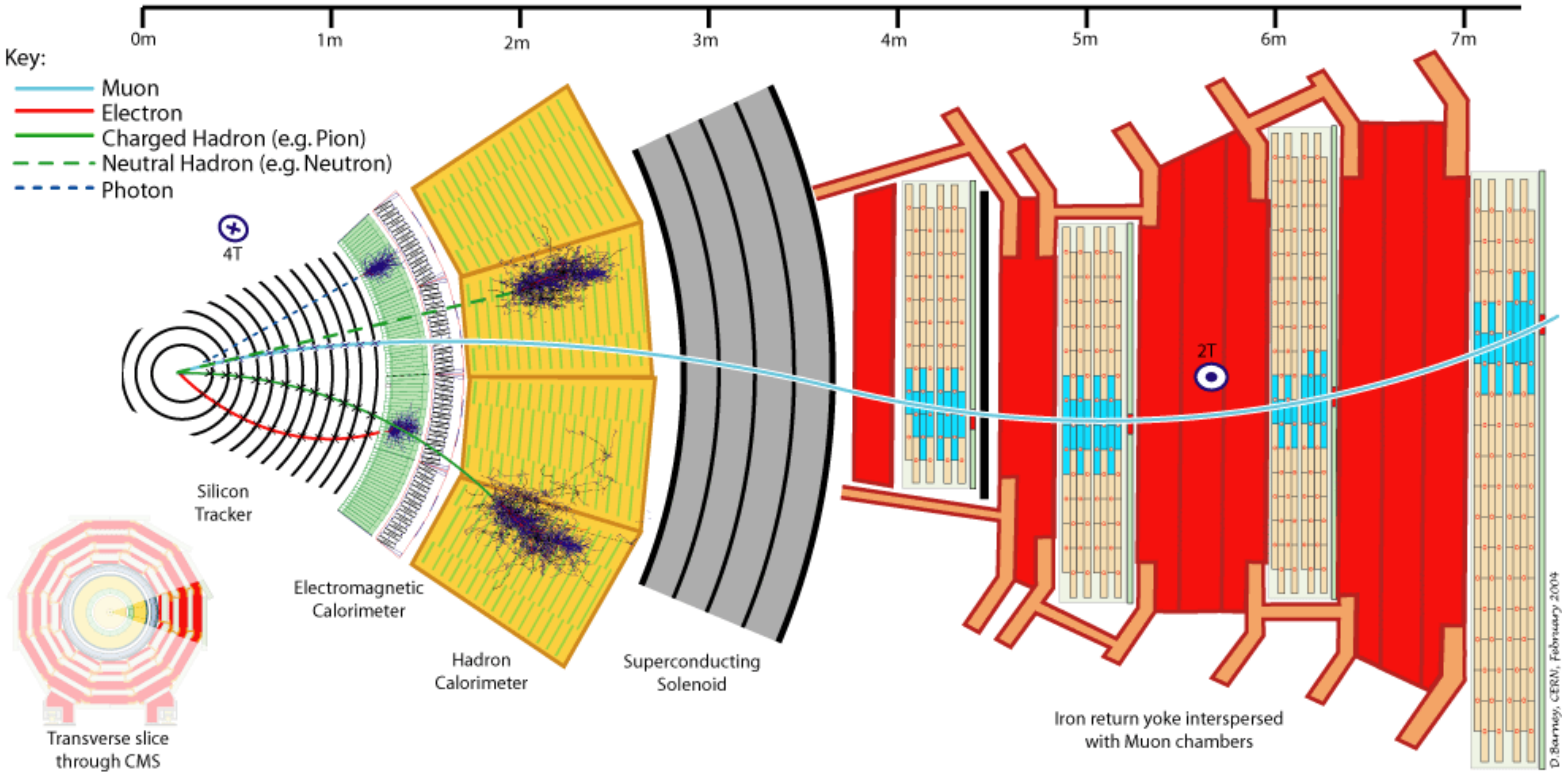
https://www.hep.ph.ic.ac.uk/~tapper/

CERN & Large Hadron Collider

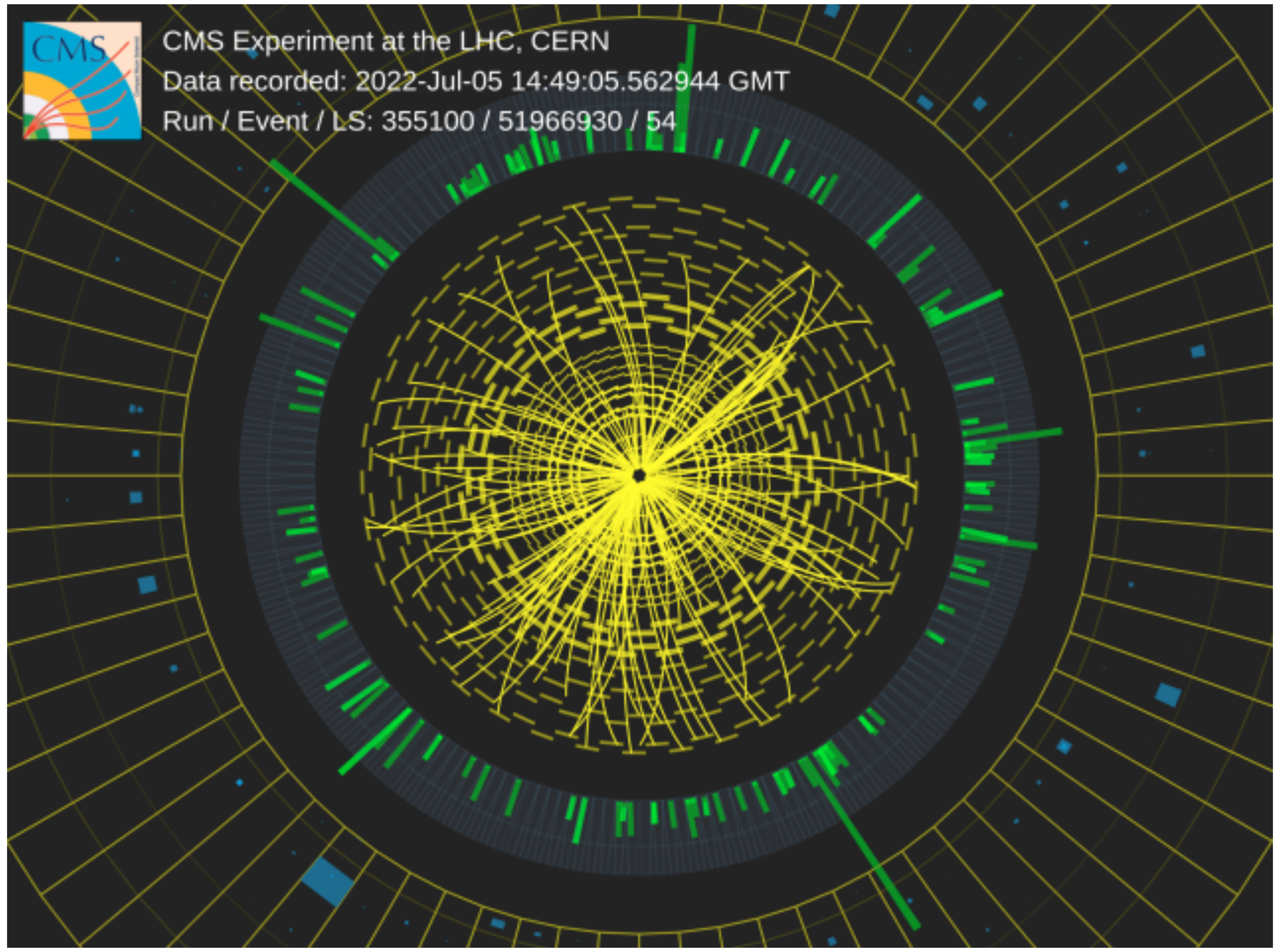
Detectors



Detectors

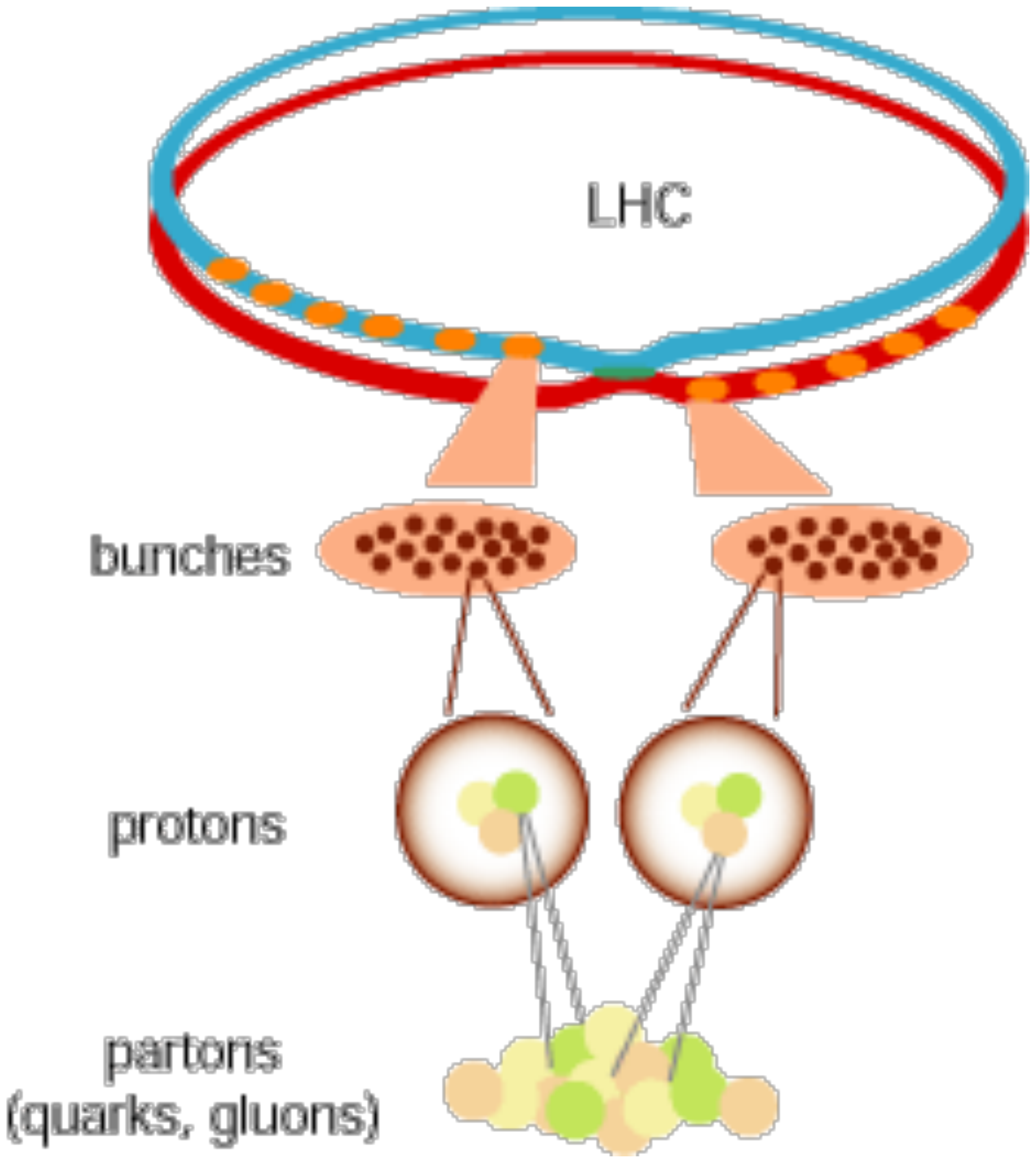


0m
1m
2m
3m
4m
5m
6m
7m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
Transverse slice through CMS
D.Barney, CERN, February 2004



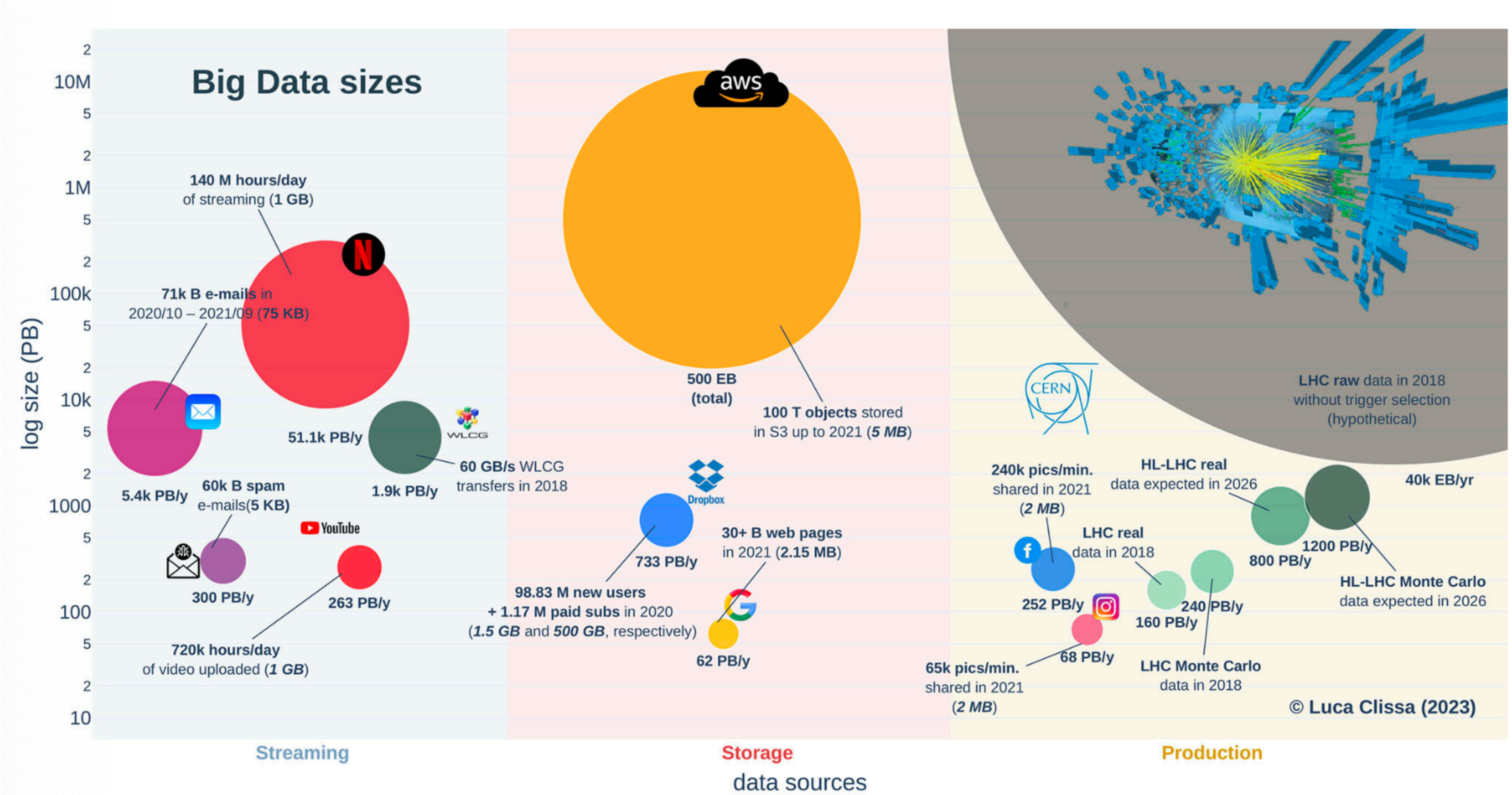
CMS Experiment at the LHC, CERN
Data recorded: 2022-Jul-05 14:49:05.562944 GMT
Run / Event / LS: 355100 / 51966930 / 54

LHC & CMS in numbers



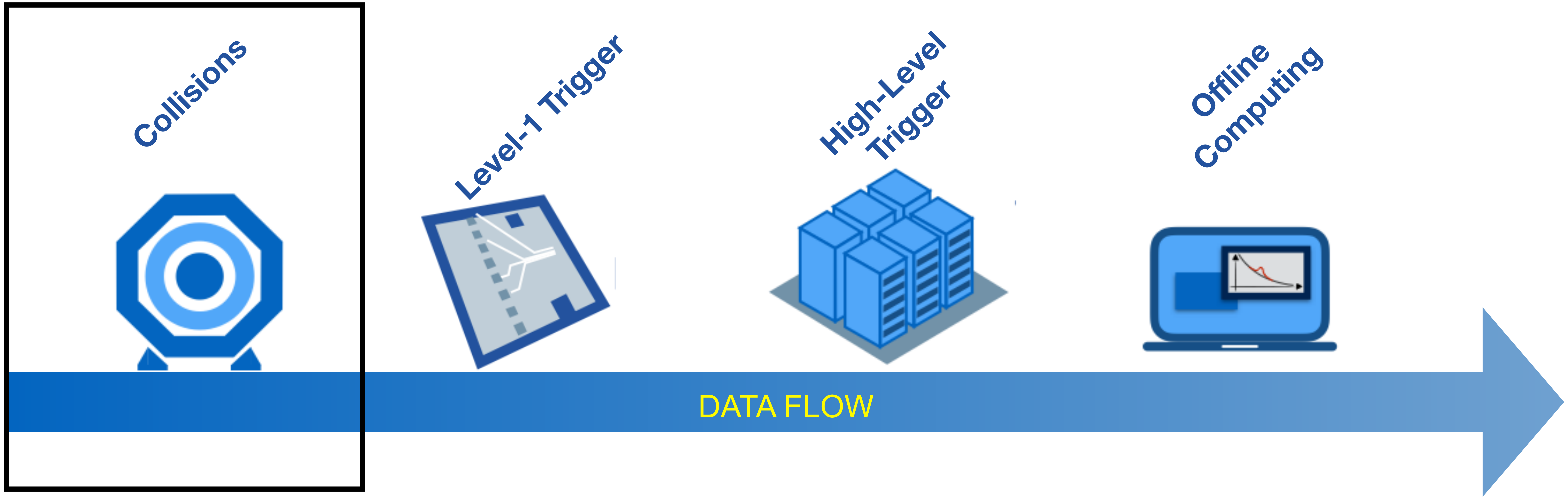
- 10 000 magnets to control beams
- Temperature -271.3 C
- Vacuum 10^{-13} atmospheres
- Beams made up of *bunches* of particles
- Around 3000 bunches, each 10^{11} particles
- Complete the 27km orbit 11000 times a second
- Collisions at 40 MHz

LHC & CMS in numbers



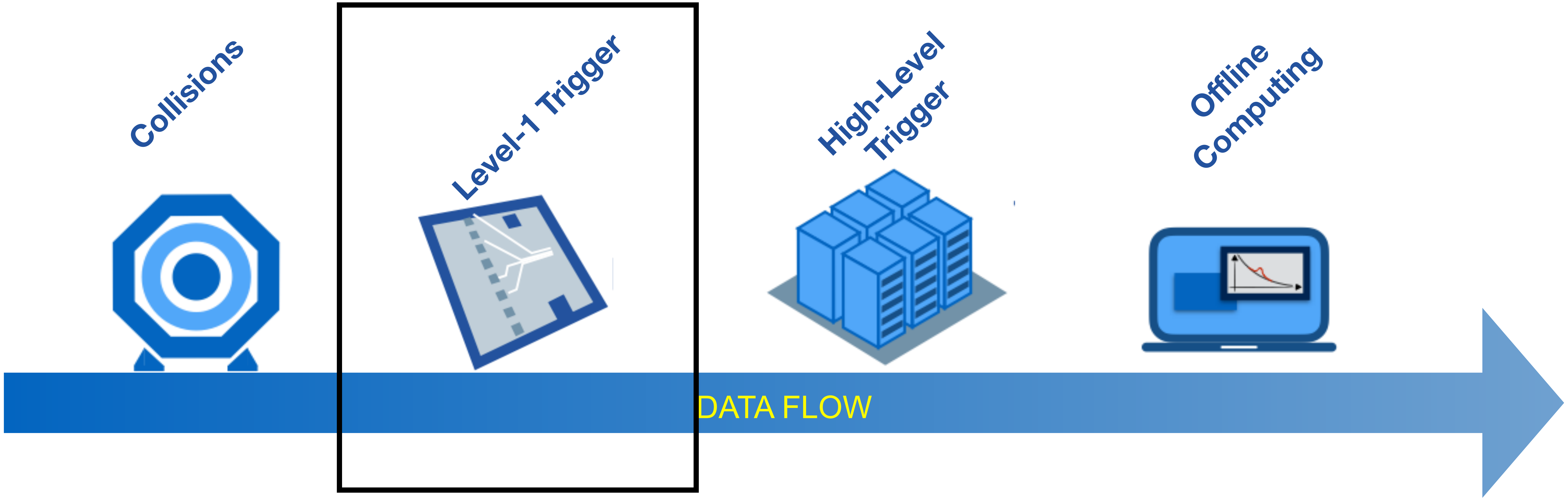
https://doi.org/10.3389/fdata.2023.1271639

Detector data flow & processing



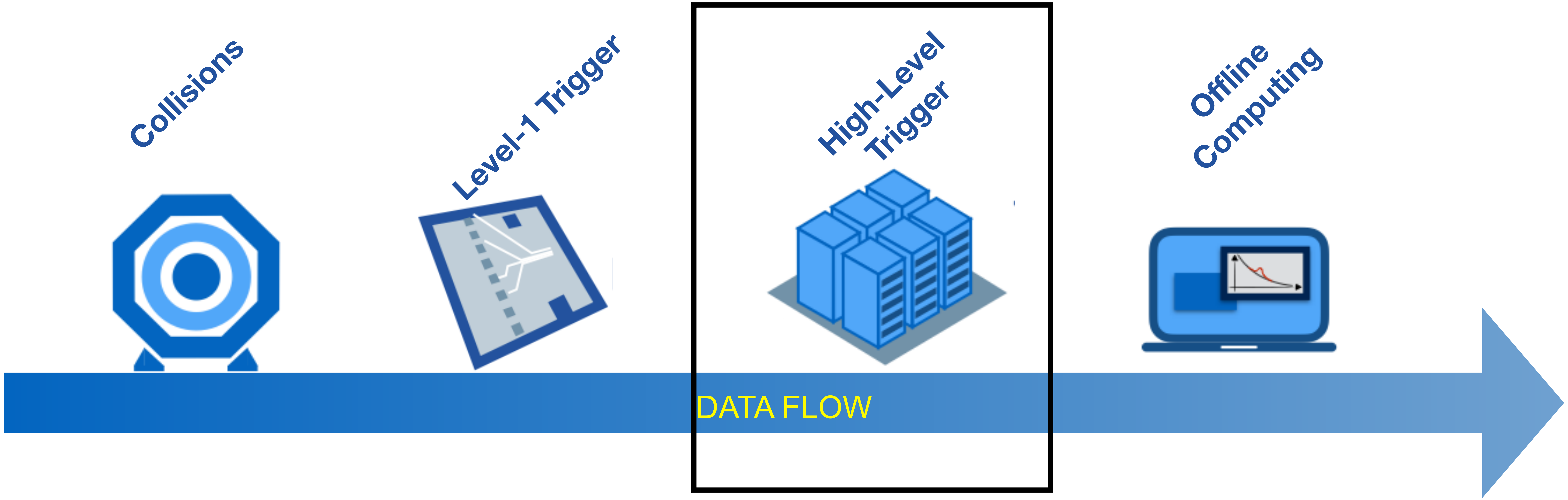
- 40 MHz collision rate x 1-8 MB per event = O (100 TB/s)
- On detector ASICs — noise reduction, pedestal subtraction, store in FIFO
- Impossible to store all data → required to filter data → buffer on detector O(10μs)

Detector data flow & processing



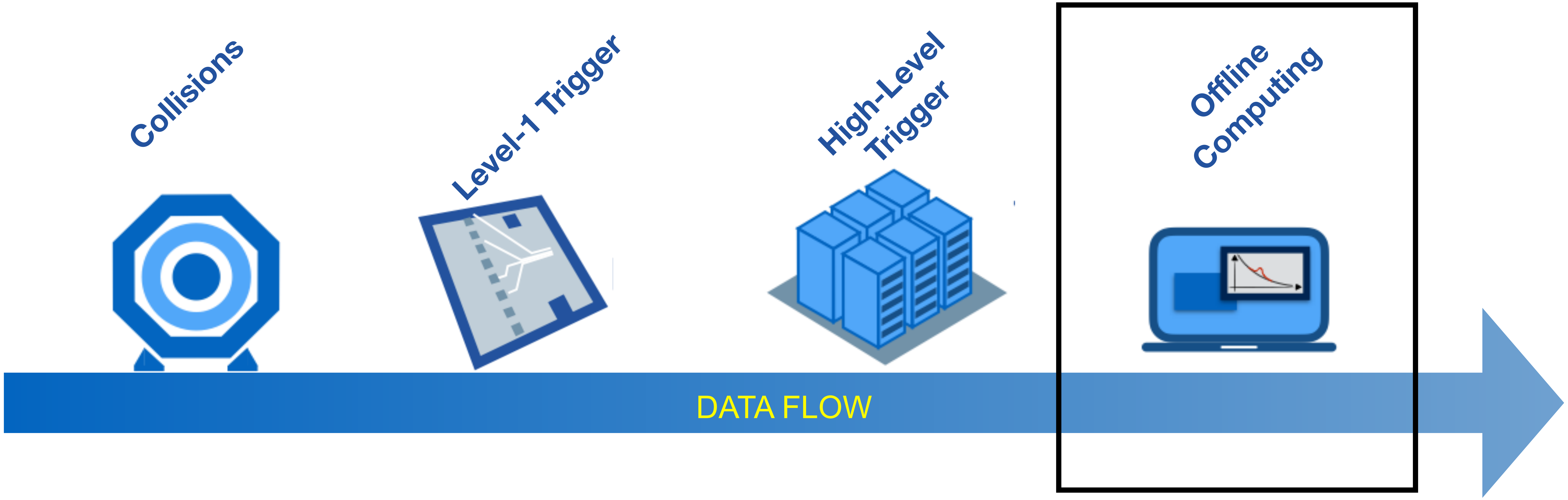
- 40 MHz input rate — 100 kHz output rate
- Process in O(10μs) **fixed** latency → algorithms O(1μs)
- Coarse/approximate detector information — currently classical algorithms

Detector data flow & processing



- 100 kHz input rate — O(10 KHz) output rate — 40 GB/s
- Software system — 30 000 x86 CPU cores + GPU accelerators (Alpaka lib)
- GPU offload 40% of processing — leads to 70% improvement in throughput

Detector data flow & processing



- Software system — ~500 000 x86 CPU cores distributed throughout world
- Also able to use ARM based resources, GPU and HPC resources where available
- Custom data and job management systems — Exa Byte dataset

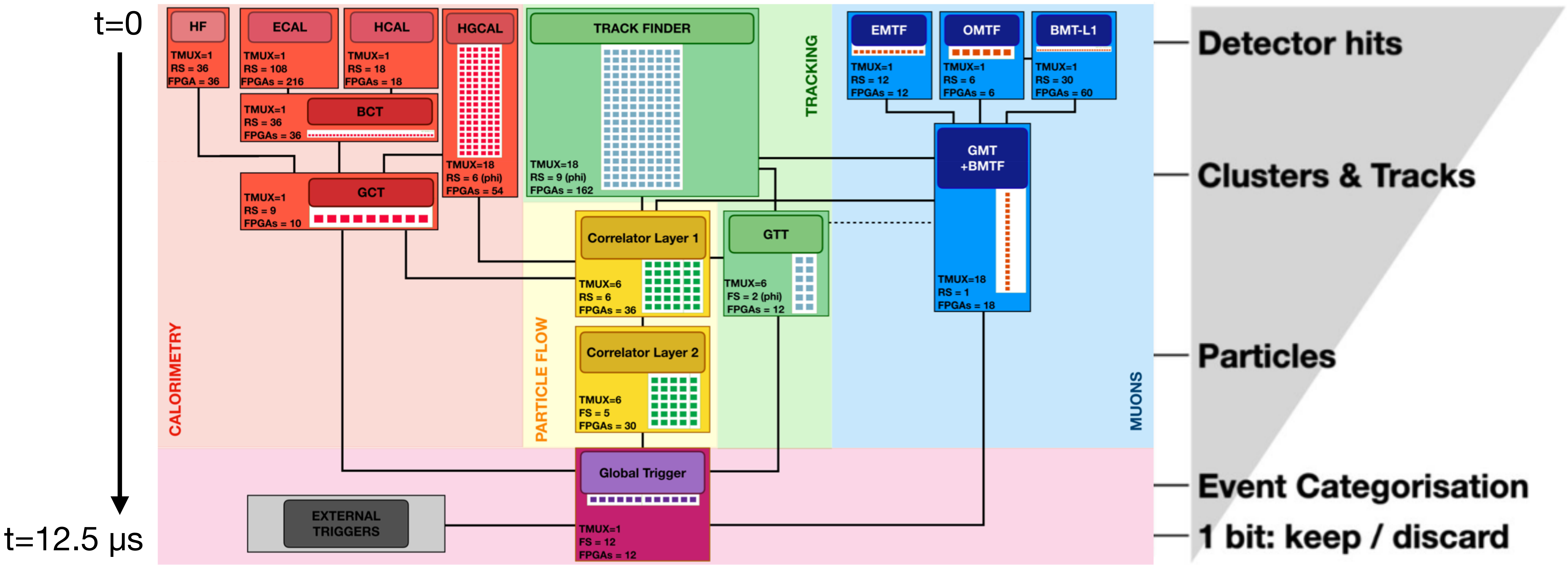
Detector data flow & processing

SERENITY

https://serenity.web.cern.ch/serenity/

- Generic data processing engine
- Advanced Telecommunications standard (ATCA)
- Samtec Firefly optical Rx/Tx — 124 optical links @ 28 Gb/s
- Single large FPGA — typically AMD Virtex UltraScale+ VU13P (4 SLRs) — also other variants
- Kria onboard control: Zync FPGA+ARM SoC
- Up to 7 Tb/s bandwidth

Detector data flow & processing

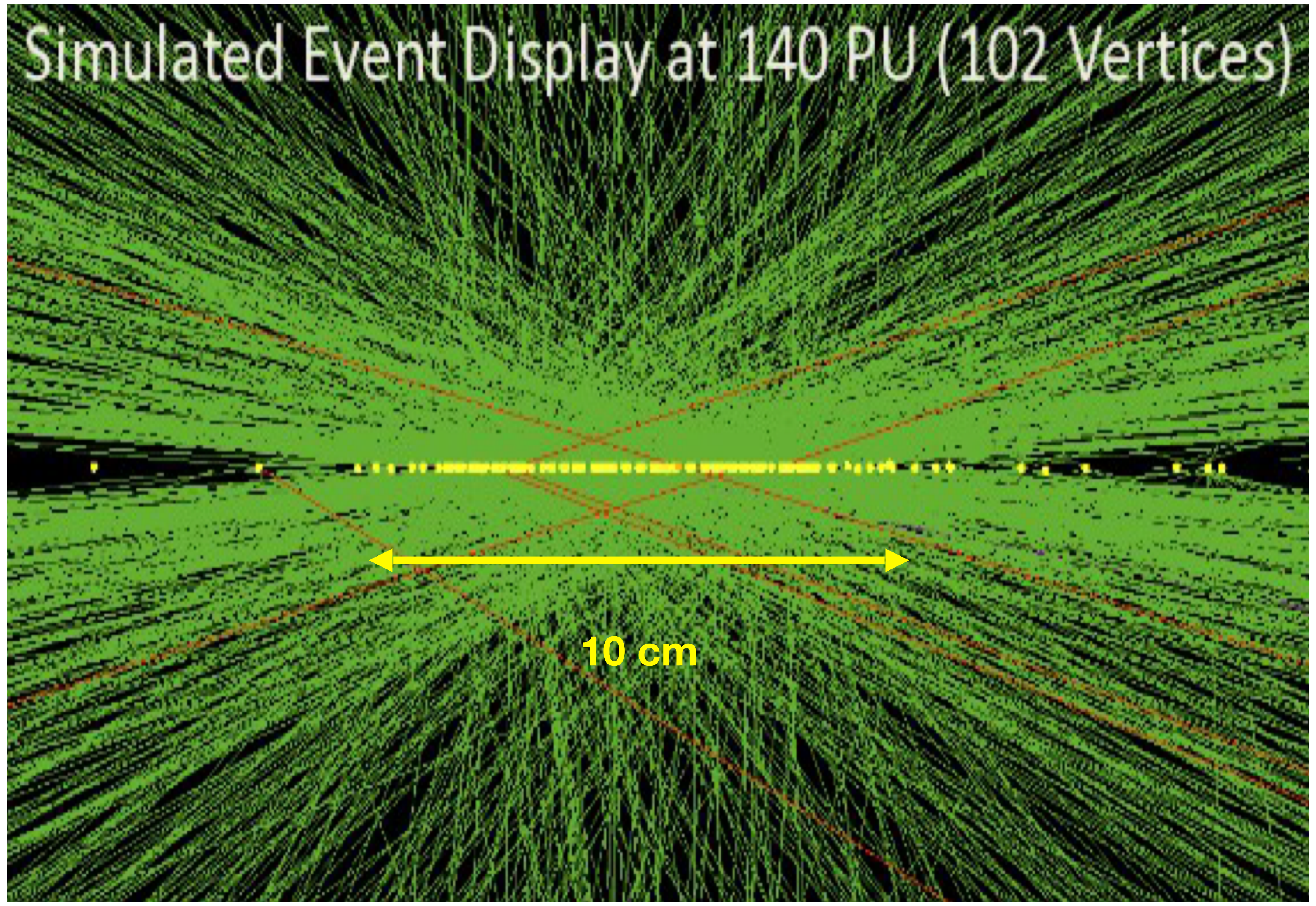


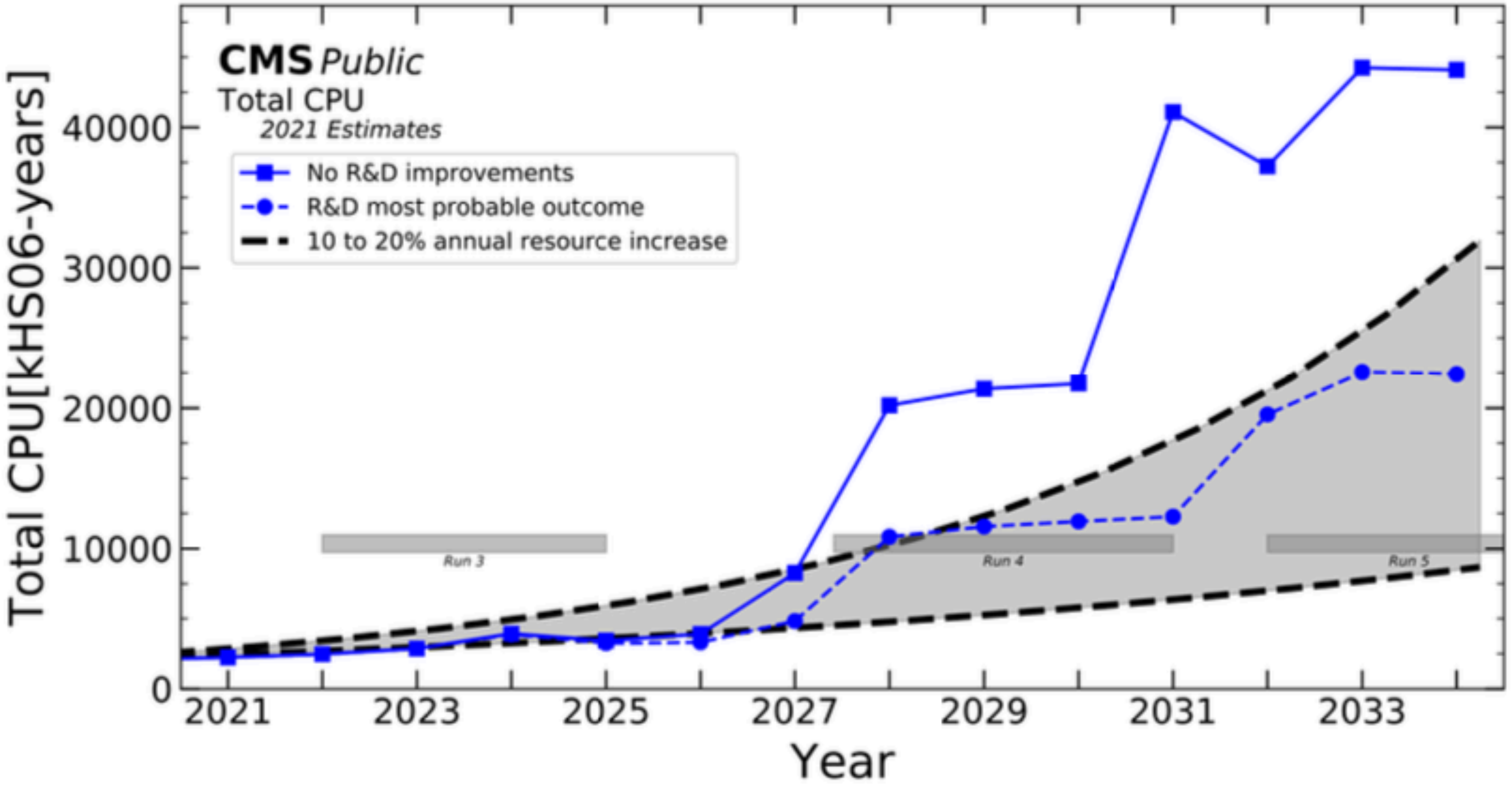
- Small boxes represent individual FPGAs
- Data processing partitioned in space and time
- Optical links with custom protocol (no forward error correction)
- Overall **~700** FPGAs in backend/ filtering system

https://cds.cern.ch/record/2714892

Future challenges

https://cds.cern.ch/record/2815292

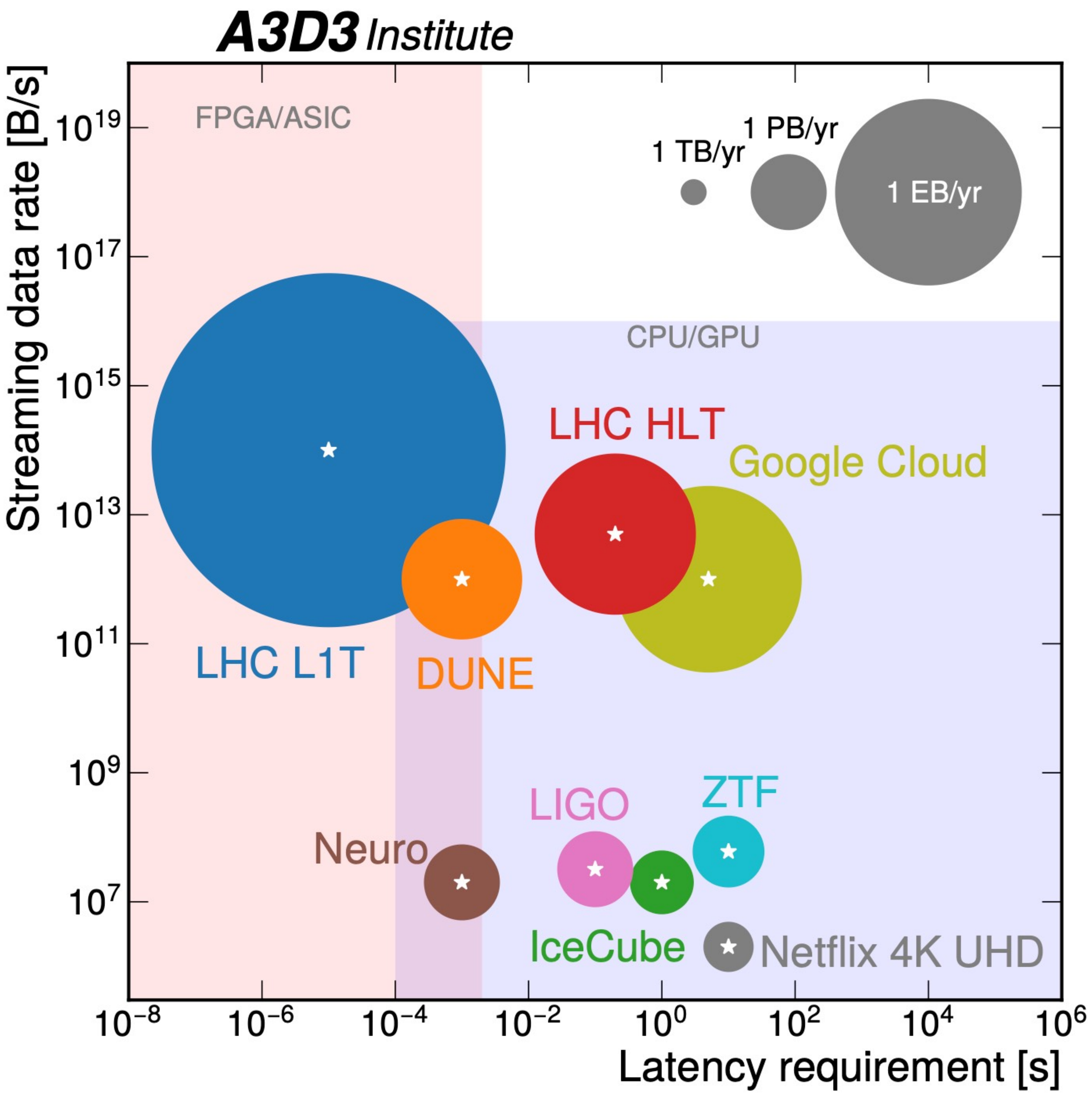




- Upgrade to be completed by 2030 — continue running until ~2040
- Higher intensity collisions → new detectors → larger data sizes

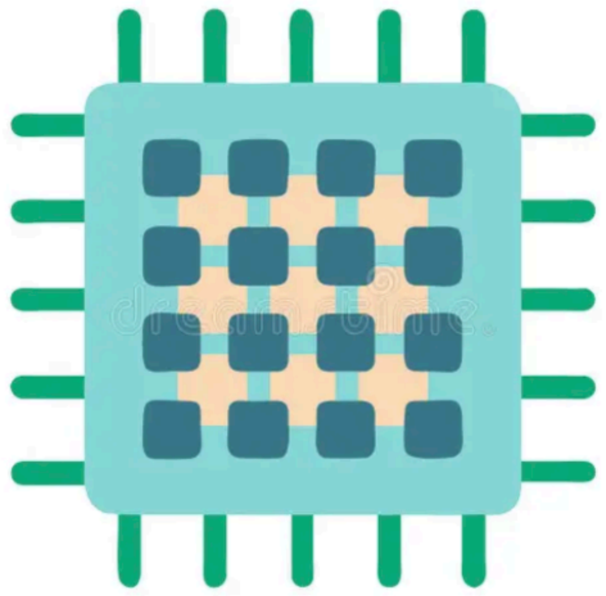
Solutions

- Hardware filtering — upgrade **FPGA-based** processors — ML algorithms
- Stream processing — **hybrid hardware/software** — new ideas
- Software filtering — **heterogeneous computing** — co-processor — IaaS
- Offline processing — **cloud, HPC, heterogeneous** computing …



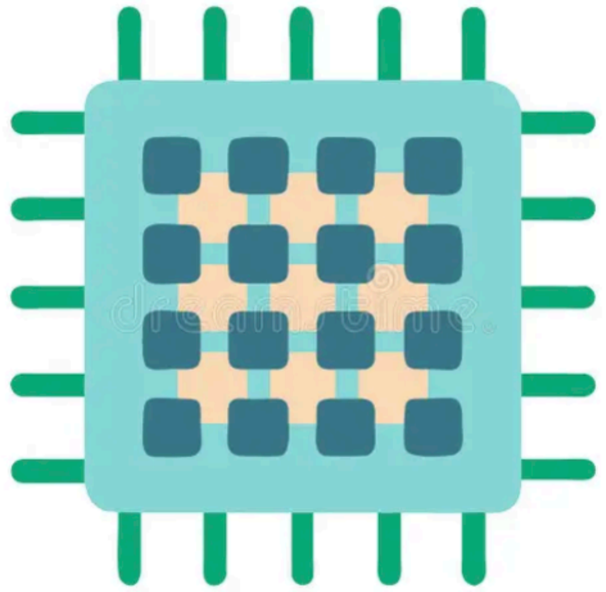
https://a3d3.ai

AI on FPGA: challenges & advantages



- **Challenges**
  - Need to operate with strict, fixed, latency constraints
  - Limited FPGA resources — model size critical
- **Advantages**
  - Fine-grained/resource parallelism — allows for low latency
  - Pipelined — allows for high throughout
  - Low power — compared to CPU/GPU
  - Custom data types — tailor at low-level to model

AI on FPGA: skillset & toolchains



- **Engineers**, typically also designers of custom electronics systems
  - VHDL framework firmware: data handling (links), clocking etc.
- **Physicists** (including students)
  - HLS (C++) for non-ML algorithms e.g. clustering, Kalman filter …
  - ML algorithms using high-level tools
  - VHDL *glue* together e.g. framework → hand crafted variables → ML model
- **Toolchain**
  - Currently main expertise AMD (Vivado/Vitis) — Stratix before ~2015

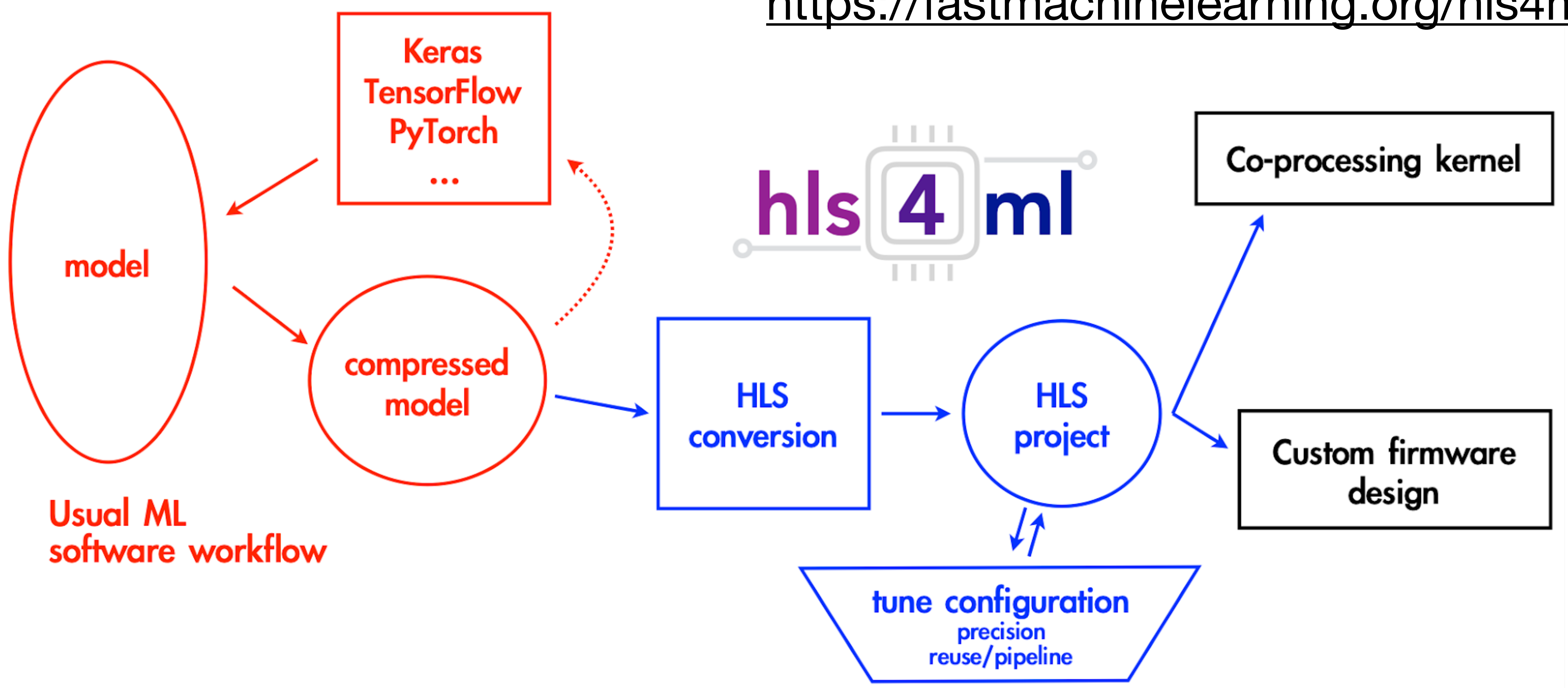
FastML tools: hls4ml

hls4ml: A Flexible, Open-Source Platform for Deep Learning Acceleration on Reconfigurable Hardware

JAN-FREDERIK SCHULTE*, Purdue University, USA
BENJAMIN RAMHORST*, ETH Zurich, Switzerland
CHANG SUN*, California Institute of Technology, USA
JOVAN MITREVSKI, Fermi National Accelerator Lab, USA
NICOLÒ GHIELMETTI, ENRICO LUPI, DIMITRIOS DANOPOULOS, and VLADIMIR LONČAR†, European Organization for Nuclear Research (CERN), Switzerland
JAVIER DUARTE, University of California San Diego, USA
DAVID BURNETTE, Catapult HLS - Siemens EDA, USA
LAURI LAATU, Imperial College London, United Kingdom
STYLIANOS TZELEPIS, National Technical University of Athens, Greece
KONSTANTINOS AXIOTIS and QUENTIN BERTHET‡, University of Geneva, Switzerland
HAOYAN WANG, PAUL WHITE, and SULEYMAN DEMIRSOY, Altera Corporation, USA
MARCO COLOMBO, Discovery Partners Institute, USA
THEA KLAEBOE AARRESTAD, ETH Zurich, Switzerland
SIONI SUMMERS and MAURIZIO PIERINI, European Organization for Nuclear Research (CERN), Switzerland
GIUSEPPE DI GUGLIELMO, JENNIFER NGADIUBA, JAVIER CAMPOS, BEN HAWKS, ABHIJITH GANDRAKOTA, FARAH FAHIM, and NHAN TRAN, Fermi National Accelerator Lab, USA
GEORGE A. CONSTANTINIDES, ZHIQIANG QUE, WAYNE LUK, and ALEXANDER TAPPER, Imperial College London, United Kingdom
DUC HOANG, NOAH PALADINO, and PHILIP HARRIS, Massachusetts Institute of Technology , USA
BO-CHENG LAI, National Yang Ming Chiao Tung University, Taiwan
MANUEL VALENTIN, RYAN FORELLI, and SEDA OGRENCI, Northwestern University, USA
LINO GERLACH, Princeton University, USA
RIAN BROOKS FLYNN and MIA LIU, Purdue University, USA
DANIEL DIAZ, ELHAM E KHODA, MELISSA QUINNAN, and RUSSELL MARROQUIN SOLARES, University of California San Diego, USA
SANTOSH PARAJULI and MARK S. NEUBAUER, University of Illinois Urbana-Champaign, USA
CHRISTIAN HERWIG, University of Michigan, USA

arXiv:2512.01463v1 [cs.AR] 1 Dec 2025

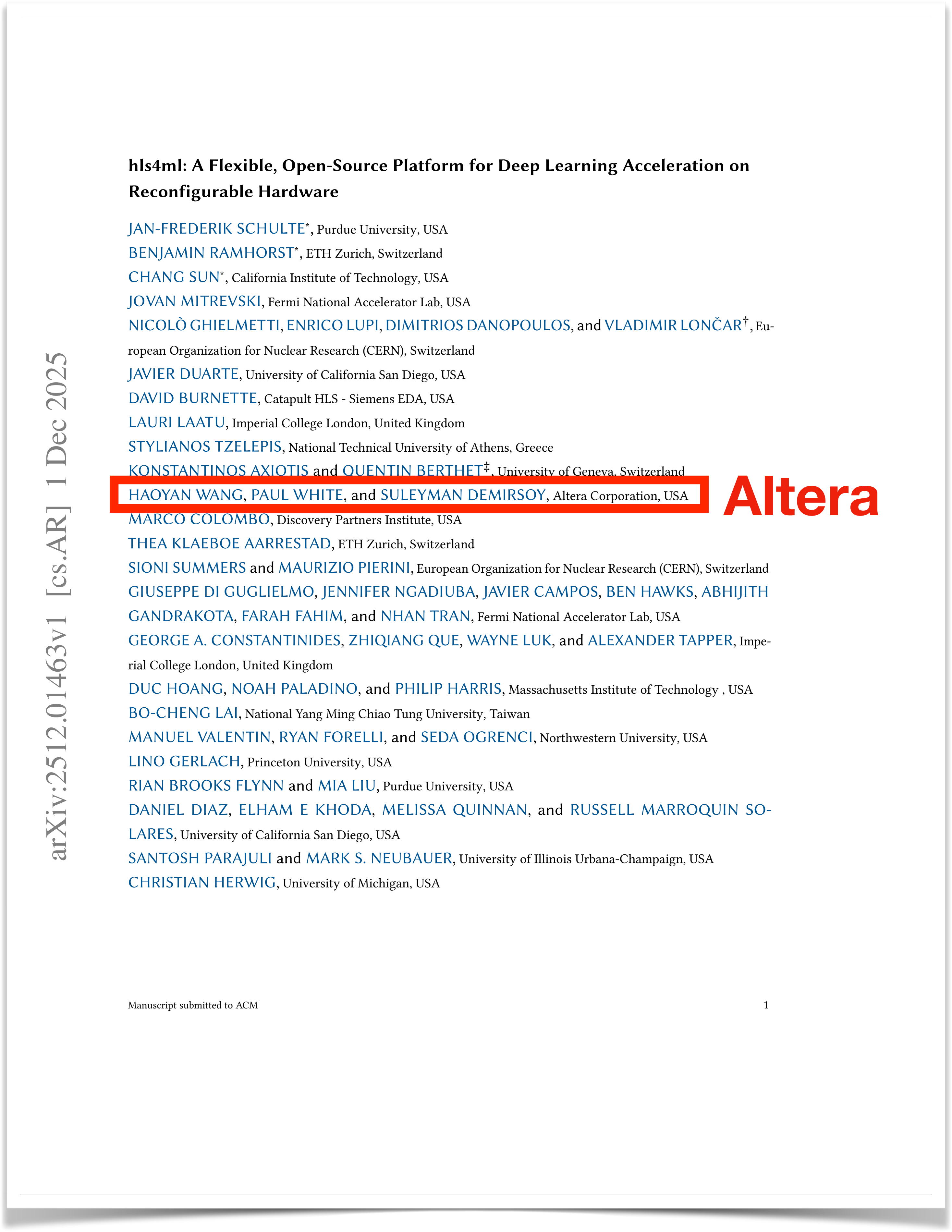
https://fastmachinelearning.org/hls4ml/



- Take ML models from your favourite framework
- Convert to HLS C++ → different targets (backends)
- Build IP core → integrate into board firmware

arXiv:2512.01463

FastML tools: hls4ml



arXiv:2512.01463v1 [cs.AR] 1 Dec 2025

hls4ml: A Flexible, Open-Source Platform for Deep Learning Acceleration on Reconfigurable Hardware

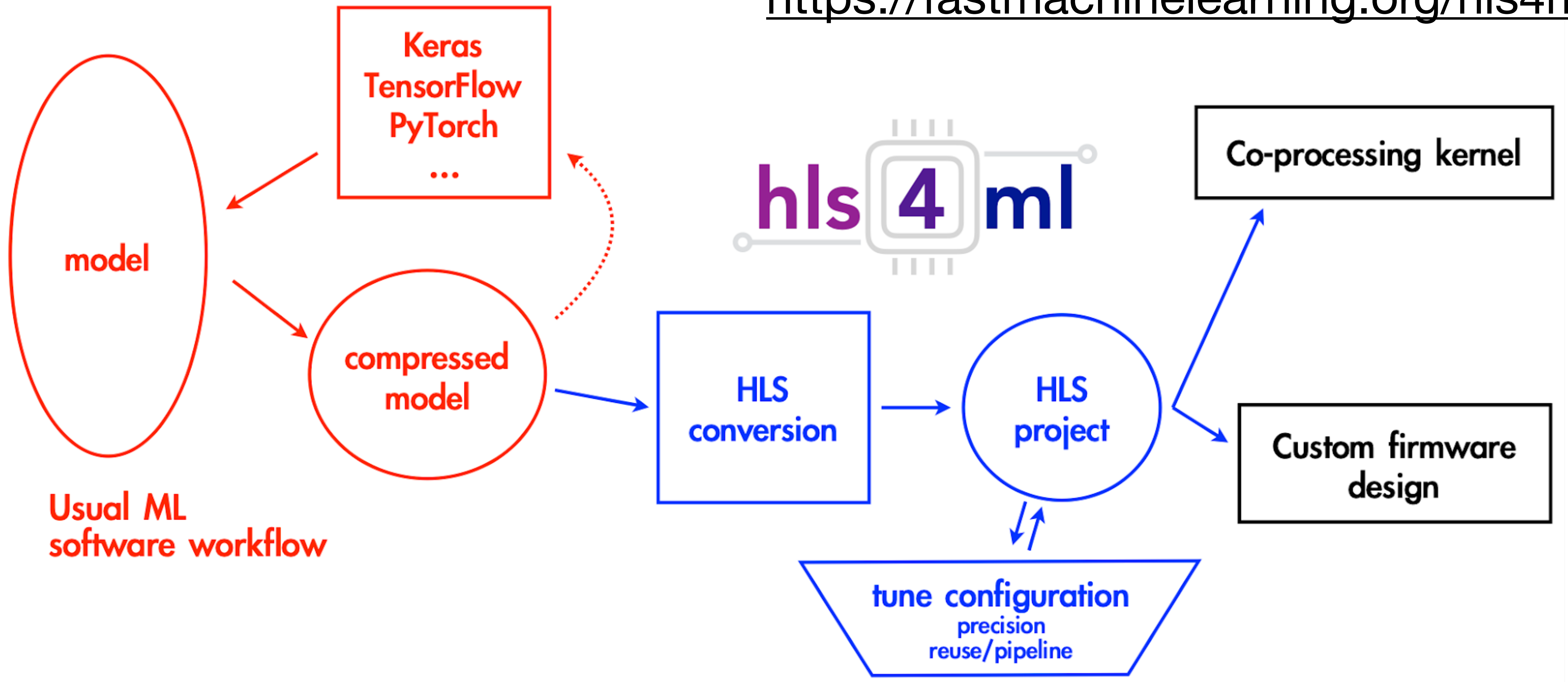
JAN-FREDERIK SCHULTE*, Purdue University, USA
BENJAMIN RAMHORST*, ETH Zurich, Switzerland
CHANG SUN*, California Institute of Technology, USA
JOVAN MITREVSKI, Fermi National Accelerator Lab, USA
NICOLÒ GHIELMETTI, ENRICO LUPI, DIMITRIOS DANOPOULOS, and VLADIMIR LONČAR†, European Organization for Nuclear Research (CERN), Switzerland
JAVIER DUARTE, University of California San Diego, USA
DAVID BURNETTE, Catapult HLS - Siemens EDA, USA
LAURI LAATU, Imperial College London, United Kingdom
STYLIANOS TZELEPIS, National Technical University of Athens, Greece
KONSTANTINOS AXIOTIS and QUENTIN BERTHET‡, University of Geneva, Switzerland
HAOYAN WANG, PAUL WHITE, and SULEYMAN DEMIRSOY, Altera Corporation, USA
MARCO COLOMBO, Discovery Partners Institute, USA
THEA KLAEBOE AARRESTAD, ETH Zurich, Switzerland
SIONI SUMMERS and MAURIZIO PIERINI, European Organization for Nuclear Research (CERN), Switzerland
GIUSEPPE DI GUGLIELMO, JENNIFER NGADIUBA, JAVIER CAMPOS, BEN HAWKS, ABHIJITH GANDRAKOTA, FARAH FAHIM, and NHAN TRAN, Fermi National Accelerator Lab, USA
GEORGE A. CONSTANTINIDES, ZHIQIANG QUE, WAYNE LUK, and ALEXANDER TAPPER, Imperial College London, United Kingdom
DUC HOANG, NOAH PALADINO, and PHILIP HARRIS, Massachusetts Institute of Technology , USA
BO-CHENG LAI, National Yang Ming Chiao Tung University, Taiwan
MANUEL VALENTIN, RYAN FORELLI, and SEDA OGRENCI, Northwestern University, USA
LINO GERLACH, Princeton University, USA
RIAN BROOKS FLYNN and MIA LIU, Purdue University, USA
DANIEL DIAZ, ELHAM E KHODA, MELISSA QUINNAN, and RUSSELL MARROQUIN SOLARES, University of California San Diego, USA
SANTOSH PARAJULI and MARK S. NEUBAUER, University of Illinois Urbana-Champaign, USA
CHRISTIAN HERWIG, University of Michigan, USA

Manuscript submitted to ACM 1

Altera

arXiv:2512.01463

https://fastmachinelearning.org/hls4ml/

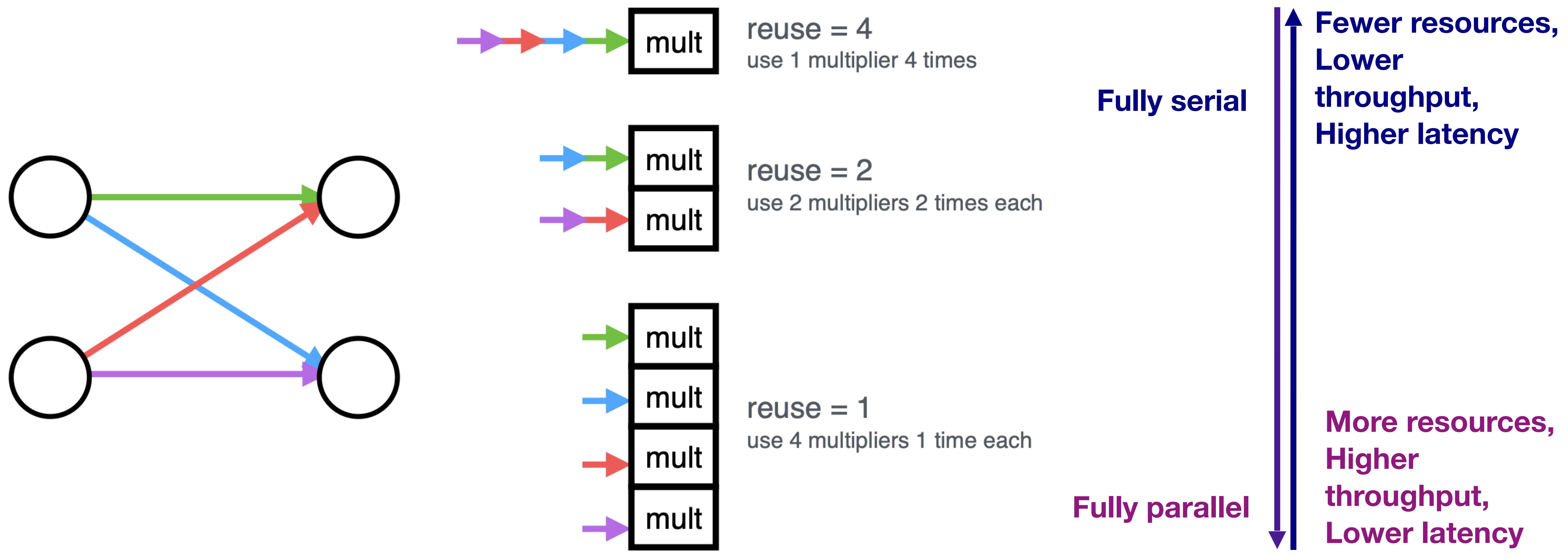


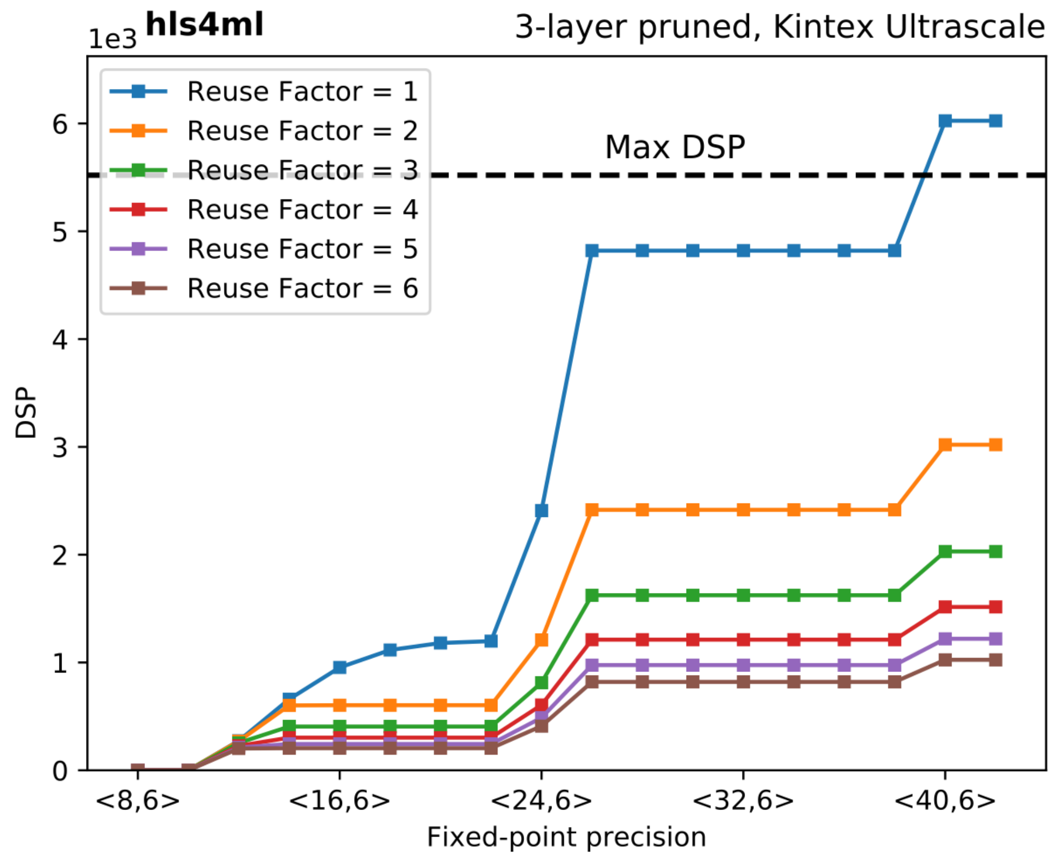
- New features — depth-wise convolutions, high granularity quantisation support, working on transformer support
- Code optimisation based on feedback from Altera of the **OneAPI** backend — efficient placement or variables
- Feedback to Altera on single bit fixed point support

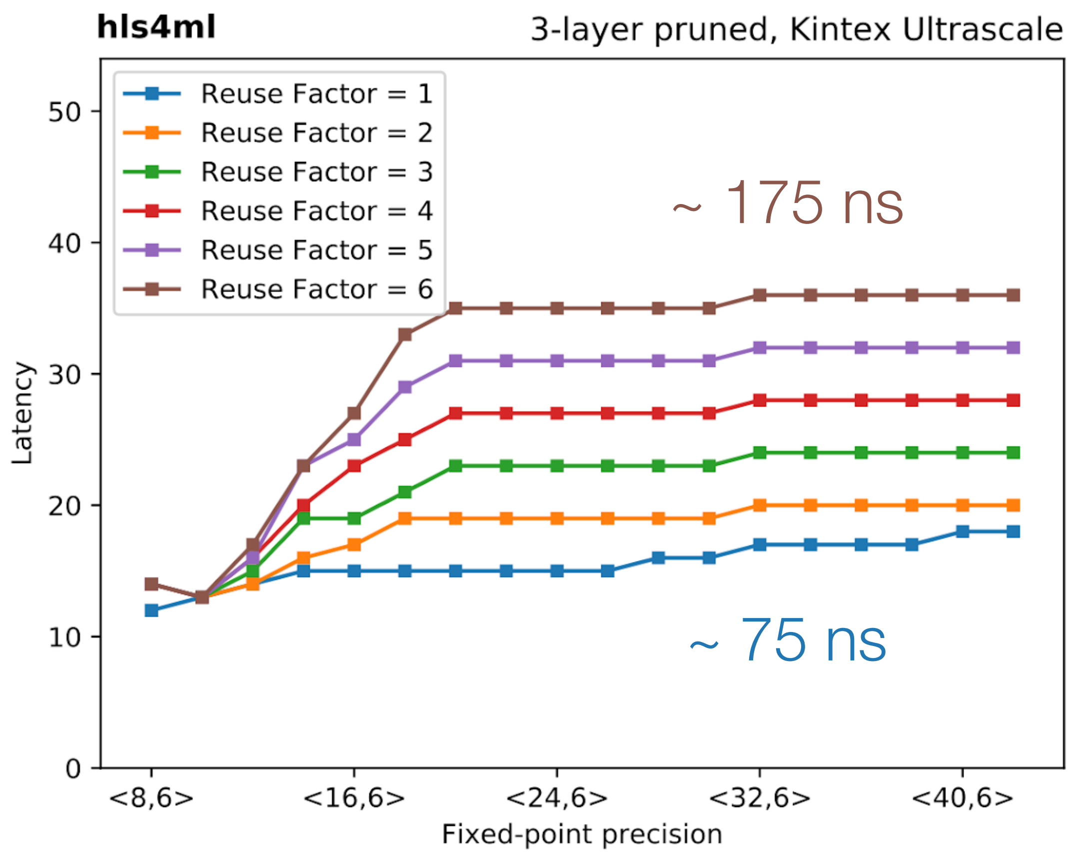
FastML tools: parallelism

Control over resources → necessary to optimise firmware core

Reuse factor — unrolling controls parallelism







https://github.com/fastmachinelearning/hls4ml-tutorial

FastML tools: precision

Control over resources → necessary to optimise firmware core

<width bits, integer bits>

0101.1011101010

integer fractional

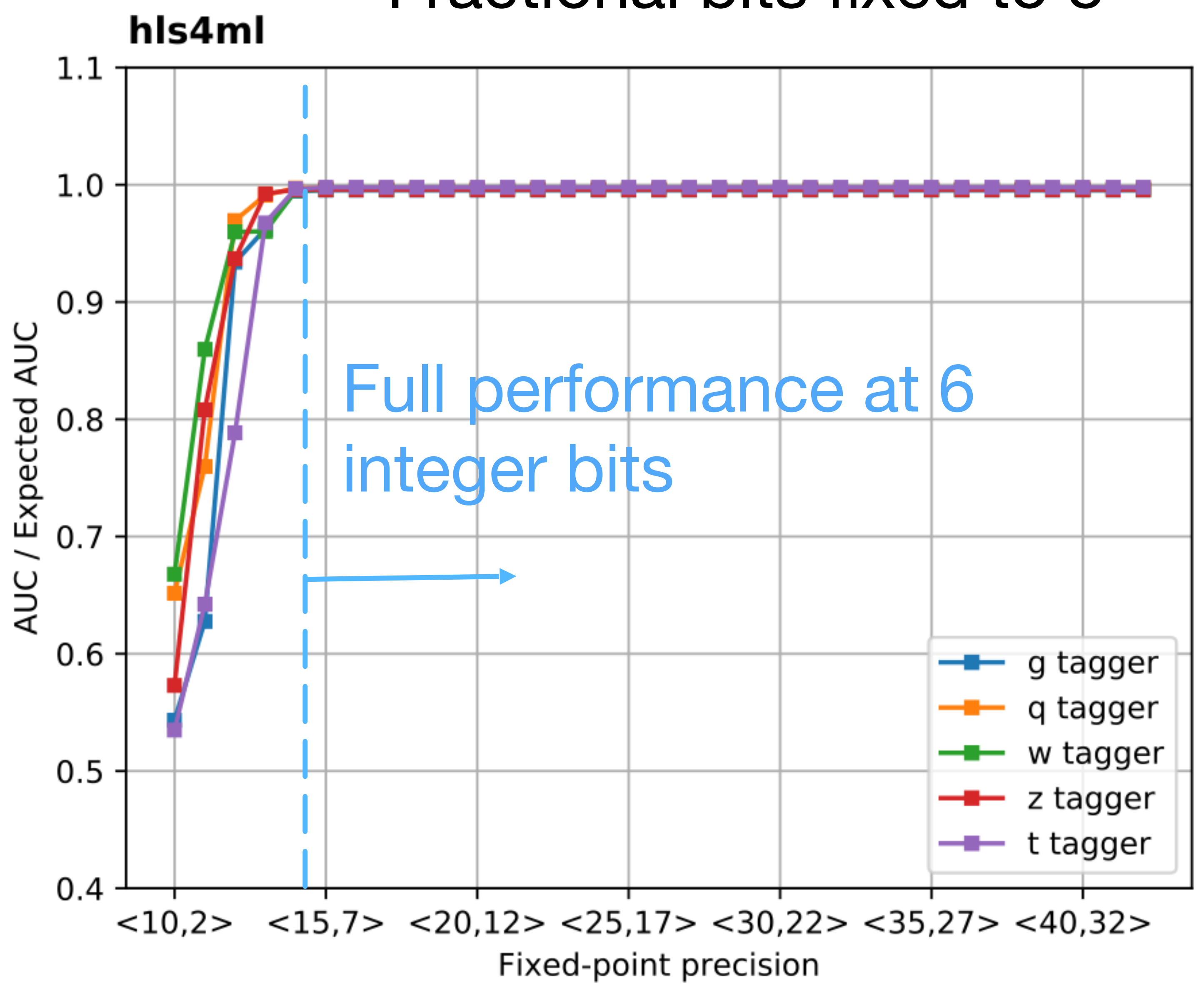
width

Precision — tune to give optimum data type for model

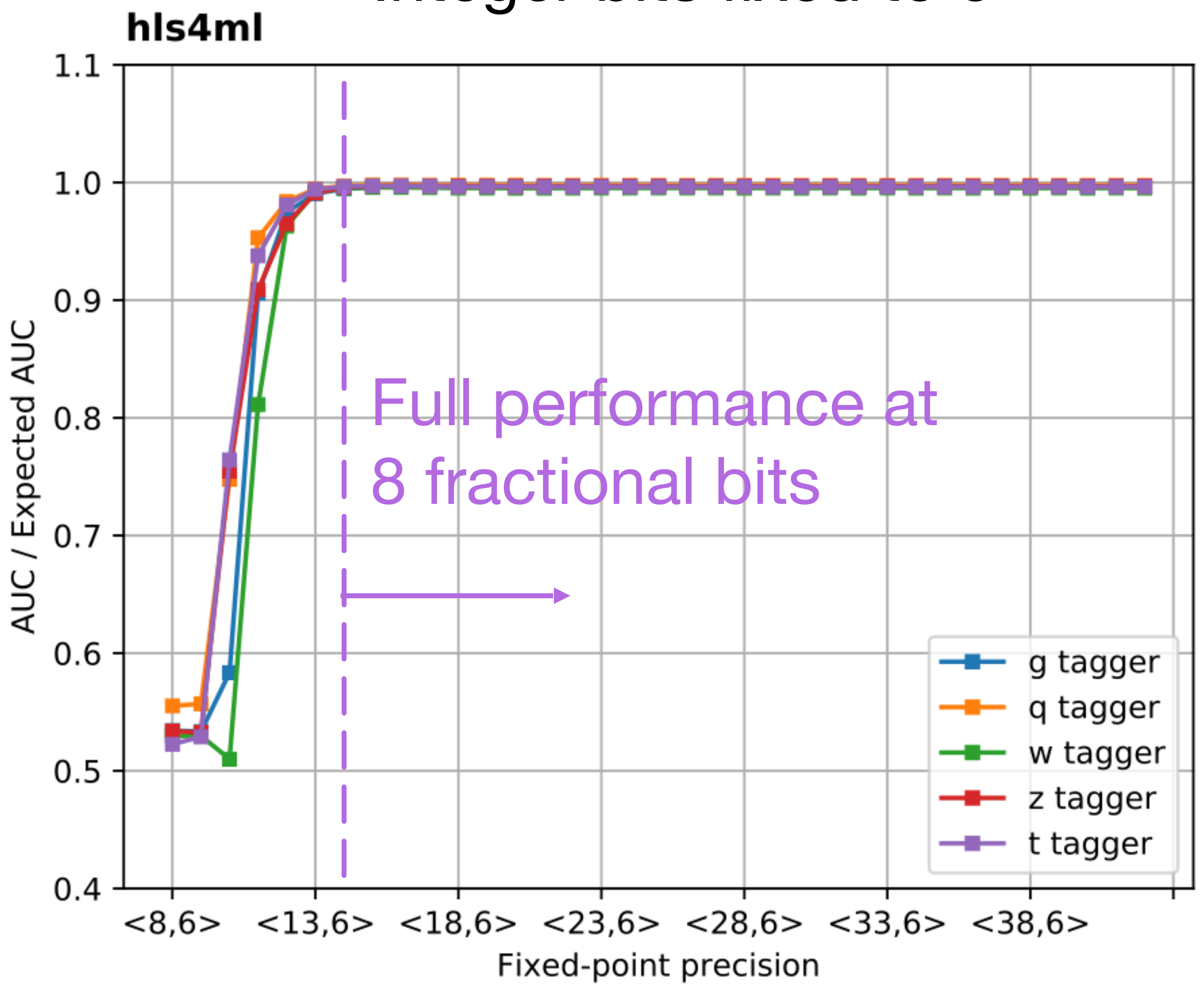
Weights and activations

https://github.com/fastmachinelearning/hls4ml-tutorial

Scan integer bits
Fractional bits fixed to 8

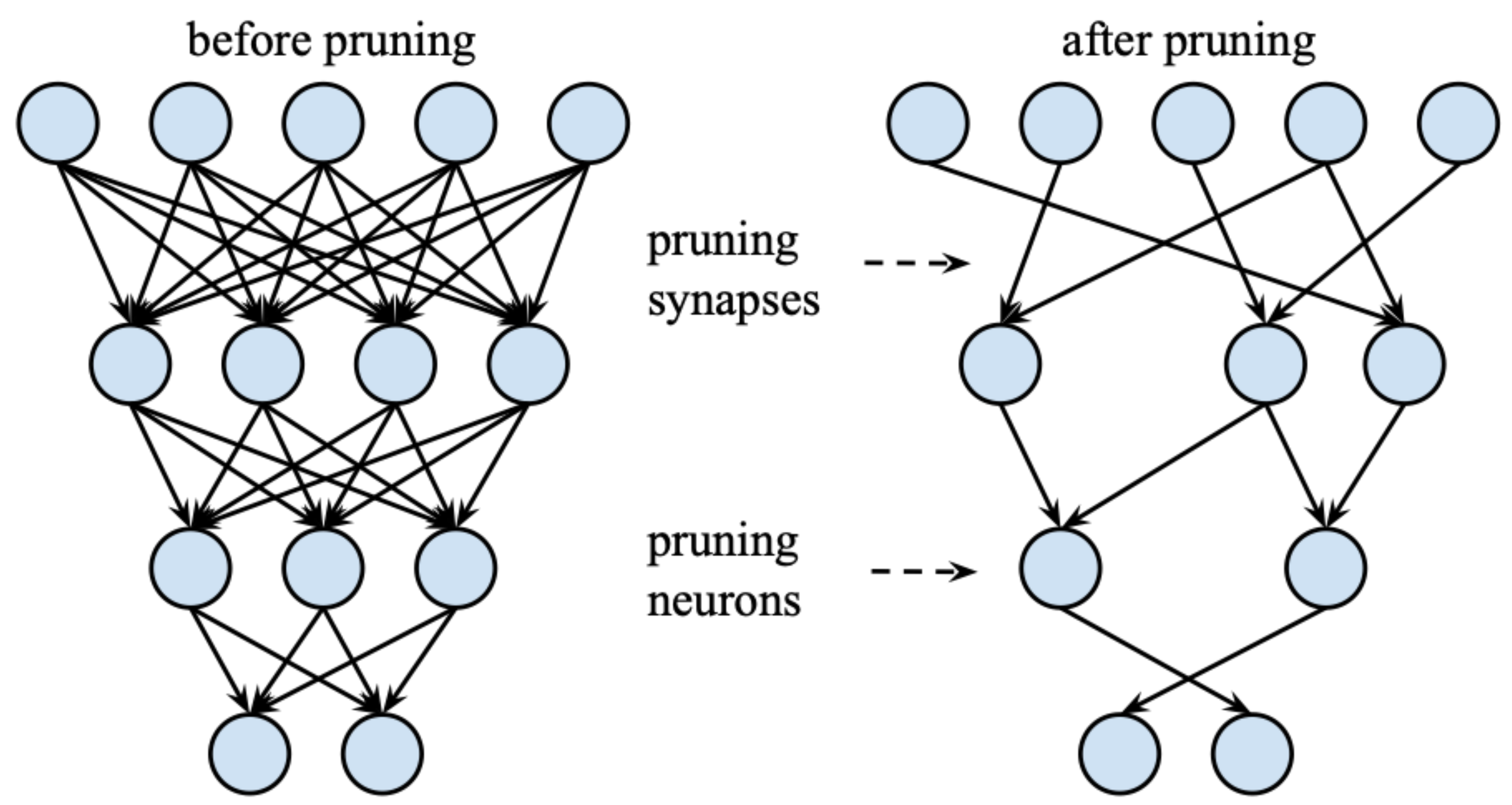


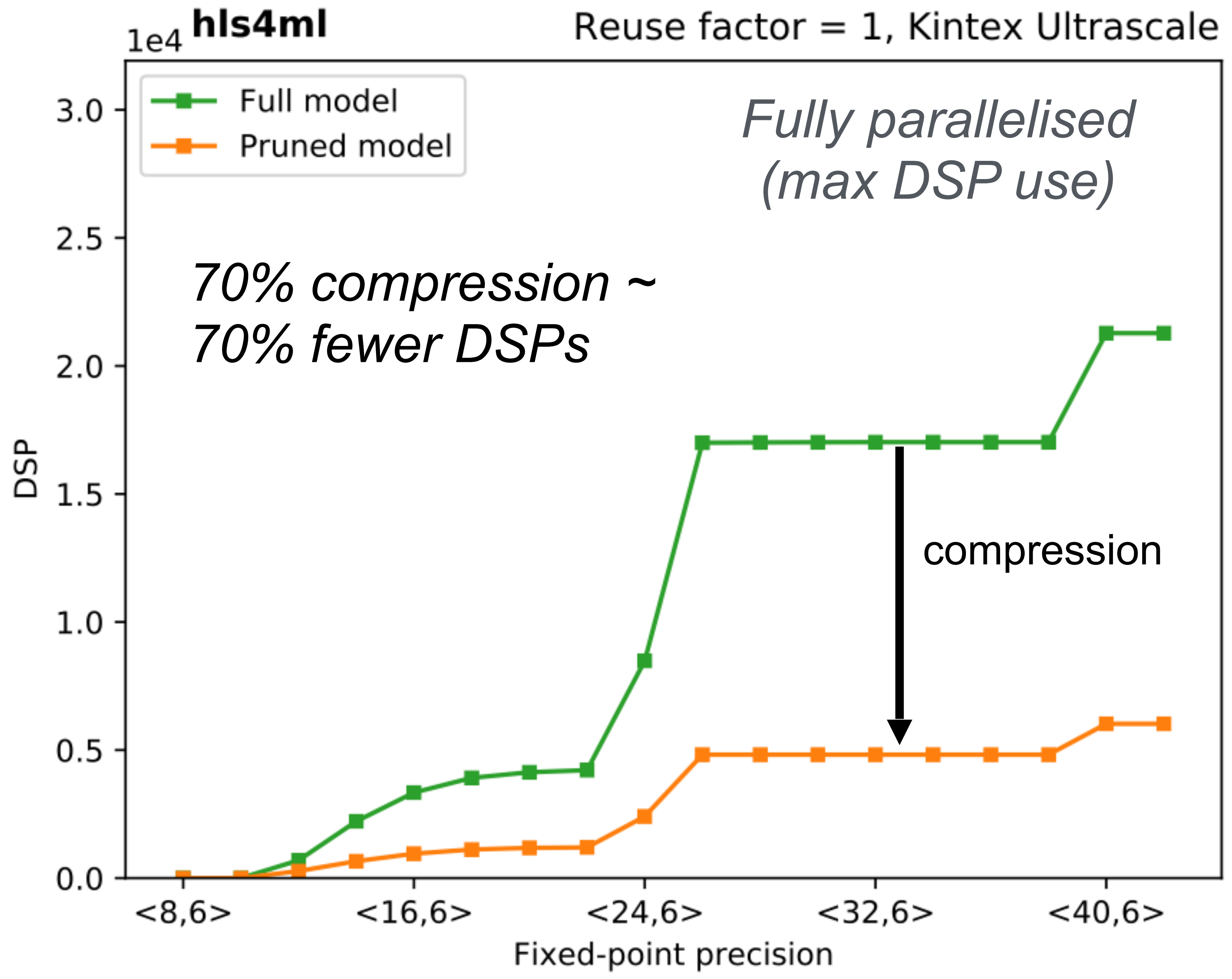
Scan fractional bits
Integer bits fixed to 6



FastML tools: pruning

- Standard technique in ML ops
  - Remove weights, nodes, layers with low weights
  - Check accuracy
  - Repeat while accuracy loss is acceptable
- Large reduction in resource usage possible while retaining high accuracy

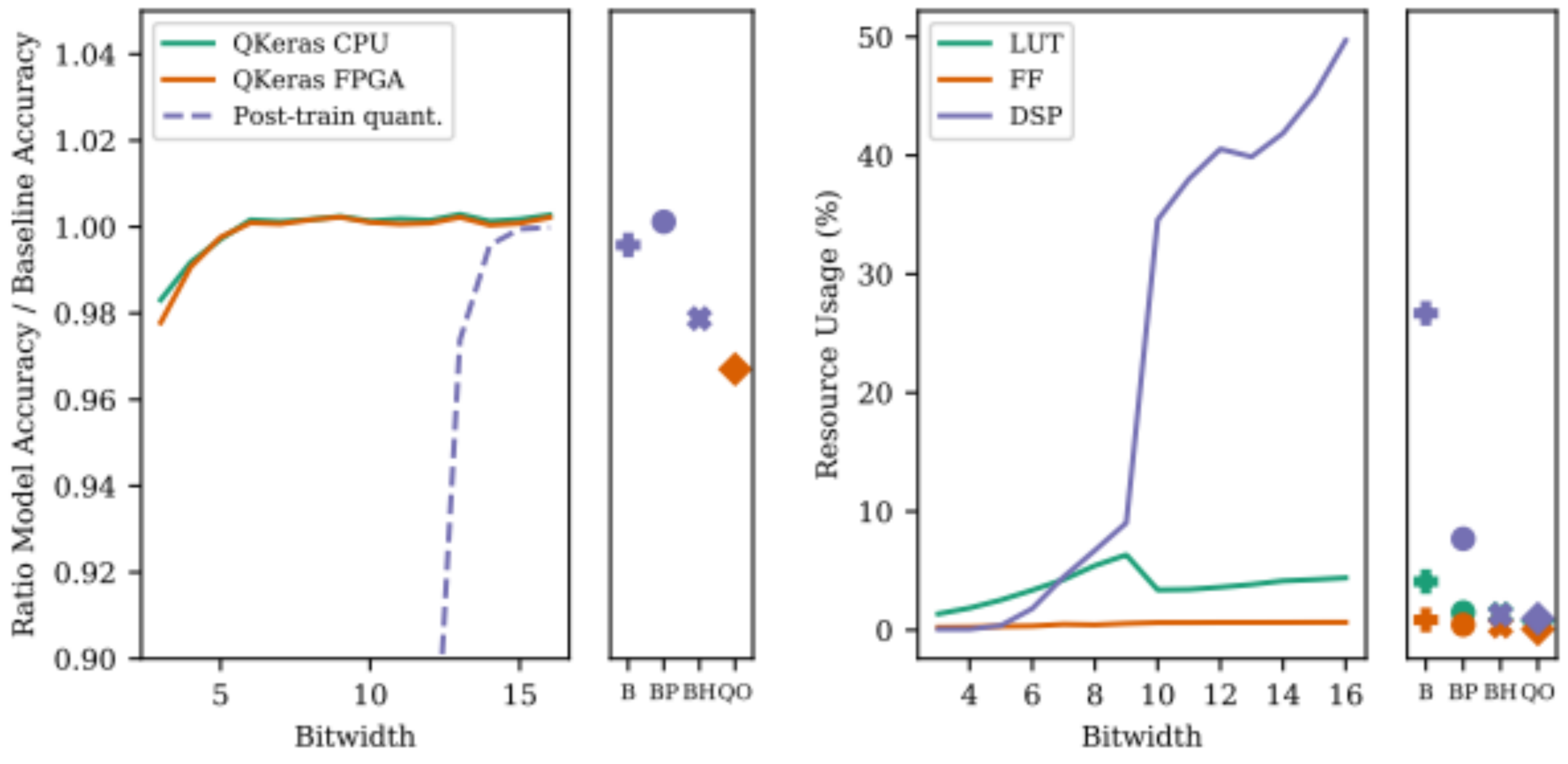


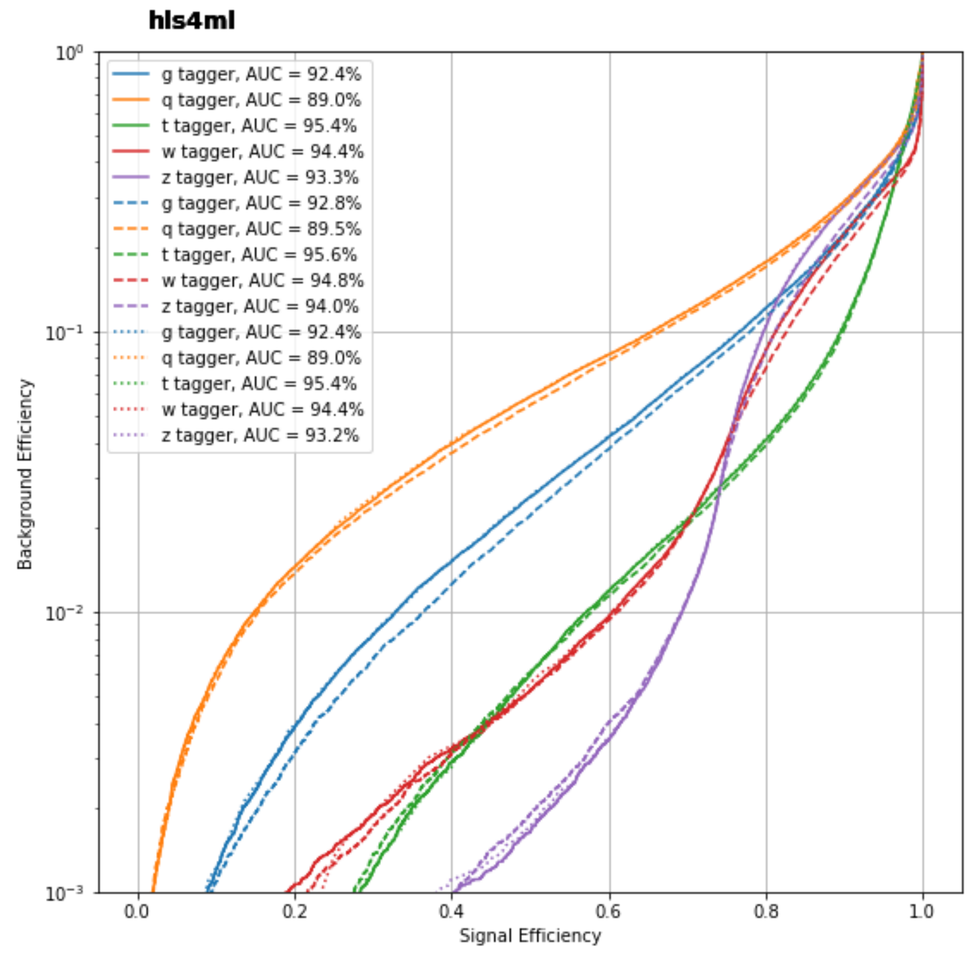


https://github.com/fastmachinelearning/hls4ml-tutorial

FastML tools: QAT

- **Quantisation Aware Training**
  - More sophisticated optimisation of precision
  - Model quantisation is included in training forward pass → training can adapt robustly to quantised data types
  - Provides better optimisation than post-training quantisation
  - Example using QKeras (Google)

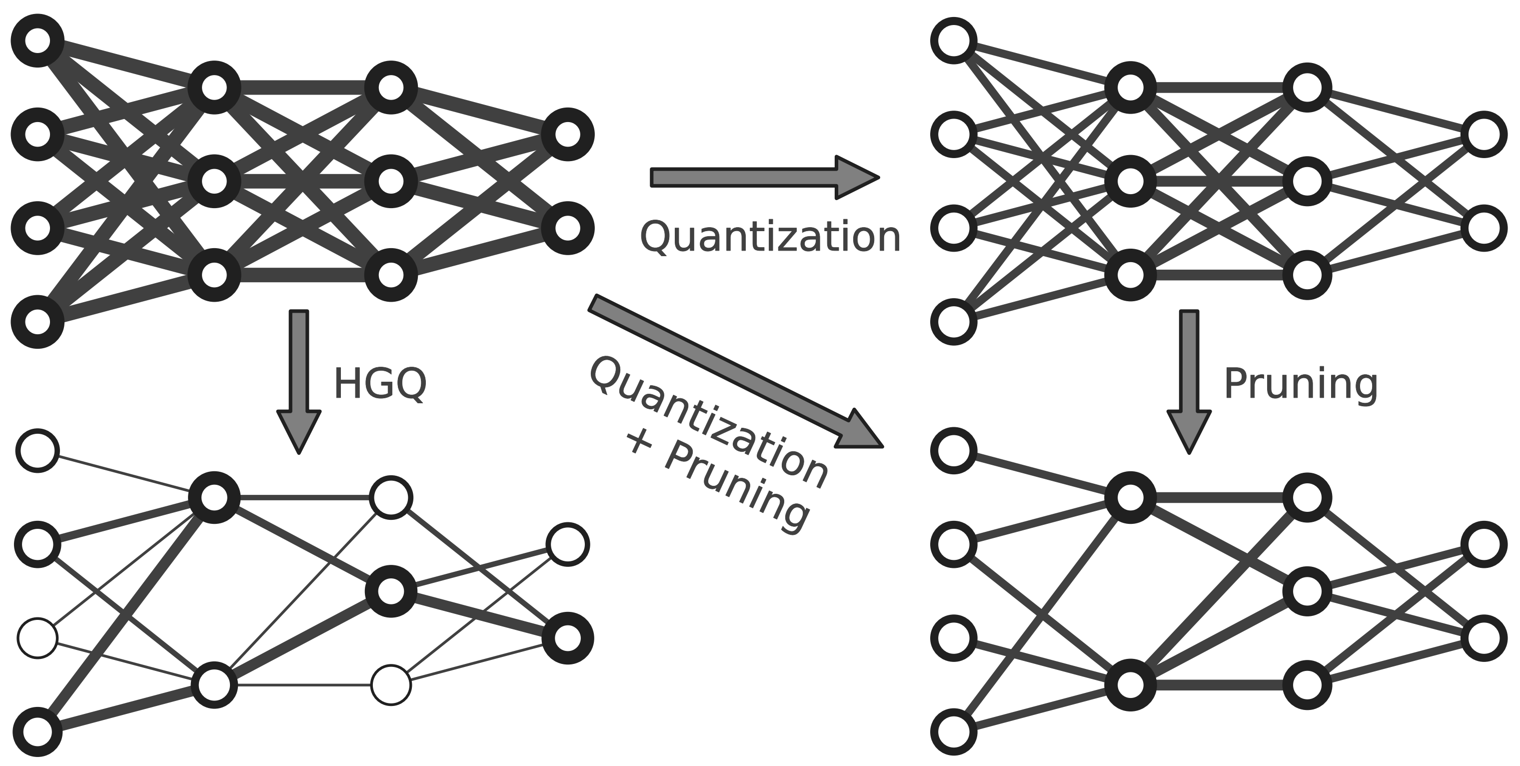


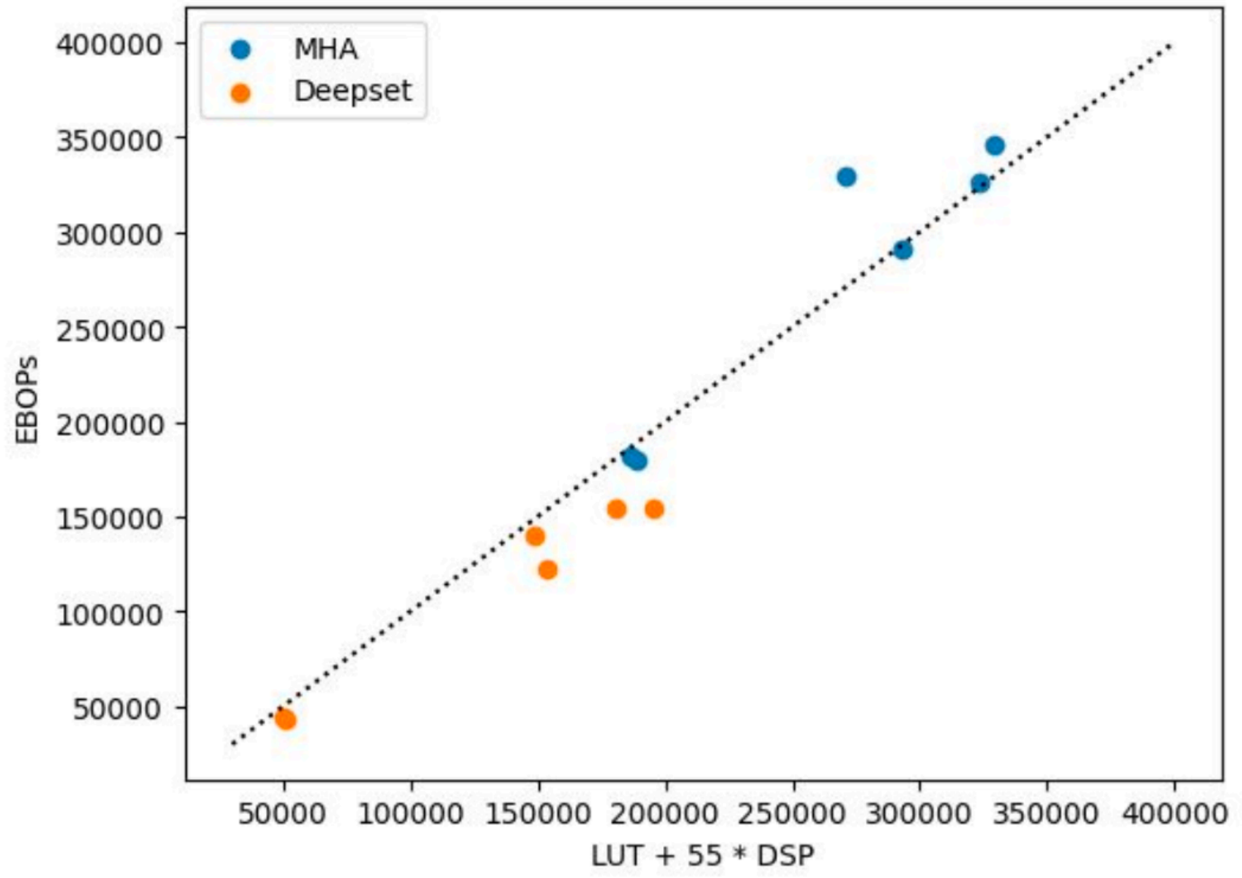


Mach. Learn.: Sci. Technol. 2, 015001 (2020)

FastML tools: HGQ

- **High Granularity Quantisation**
    - Combination of pruning and quantisation
    - Quantisation per weight, bias and activation — high granularity
    - Bit widths optimised during training — pruning occurs automatically when width = 0
    - Incorporates estimate of resource usage in training (EBOPS)





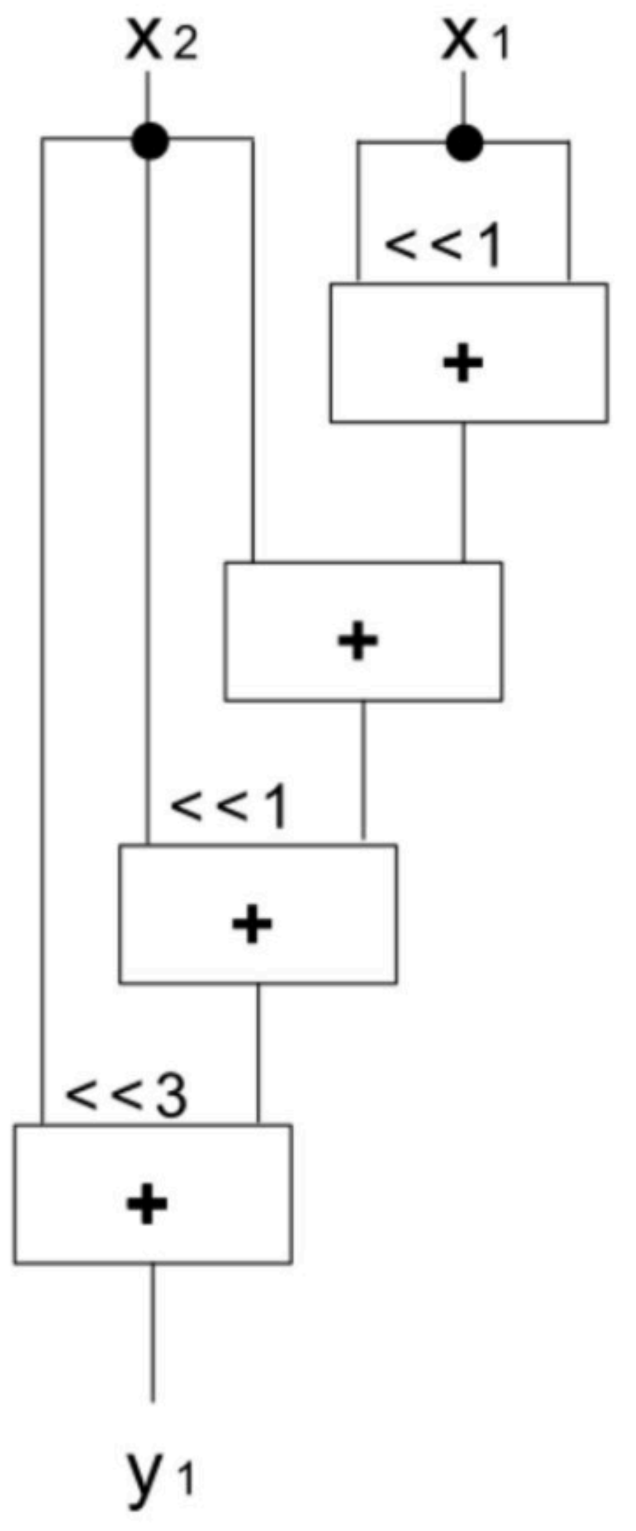
$$\mathcal{L} = \mathcal{L}_{\mathrm{base}} + \beta \cdot \overline{\mathrm{EBOPs}} + \gamma \cdot \mathrm{L1}_{\mathrm{norm}}$$

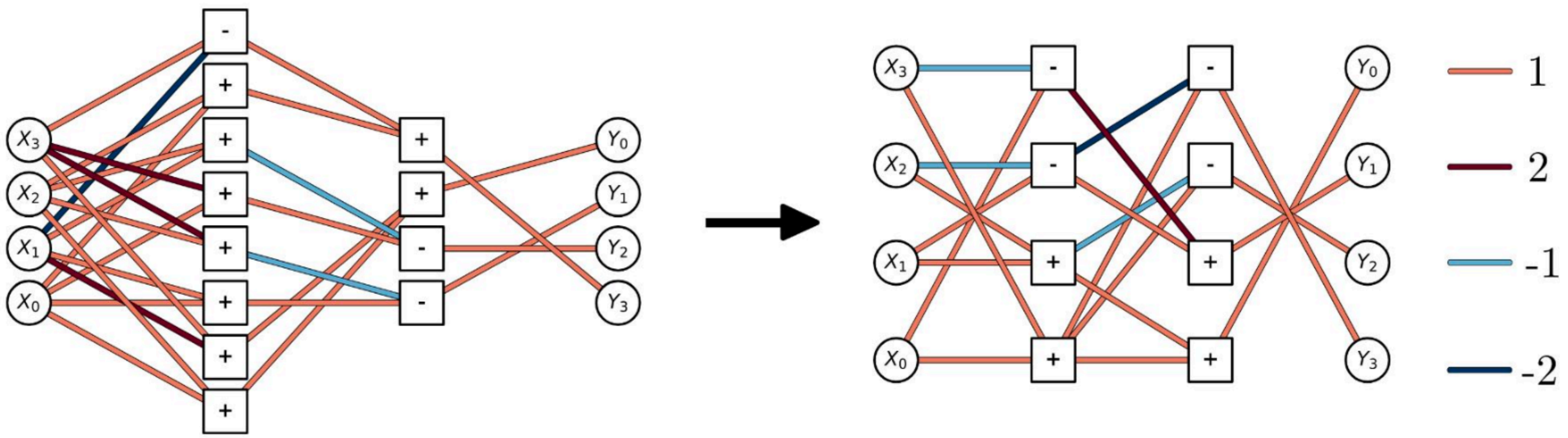
arXiv:2405.00645

FastML tools: DA

- Constant Matrix Vector Multiplication (CMVM) operations optimised using **Distributed Arithmetic**
- Splits multiplications into additions and bit-shifts to further reduce the FPGA resource usage after quantisation

$$y_1 = 3x_1 + 11x_2$$

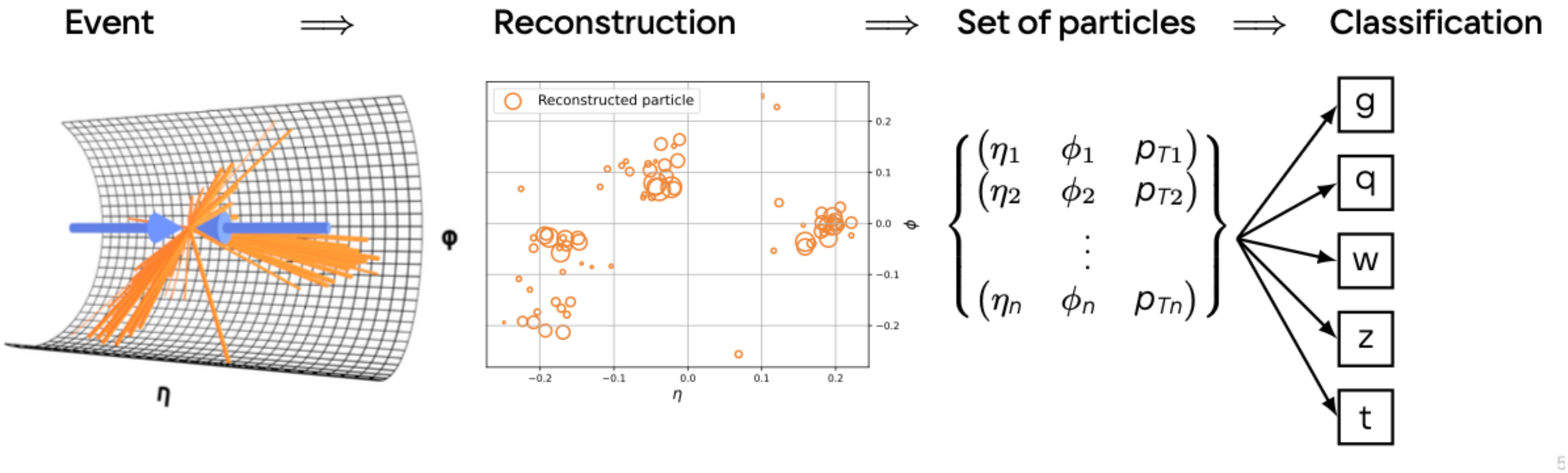


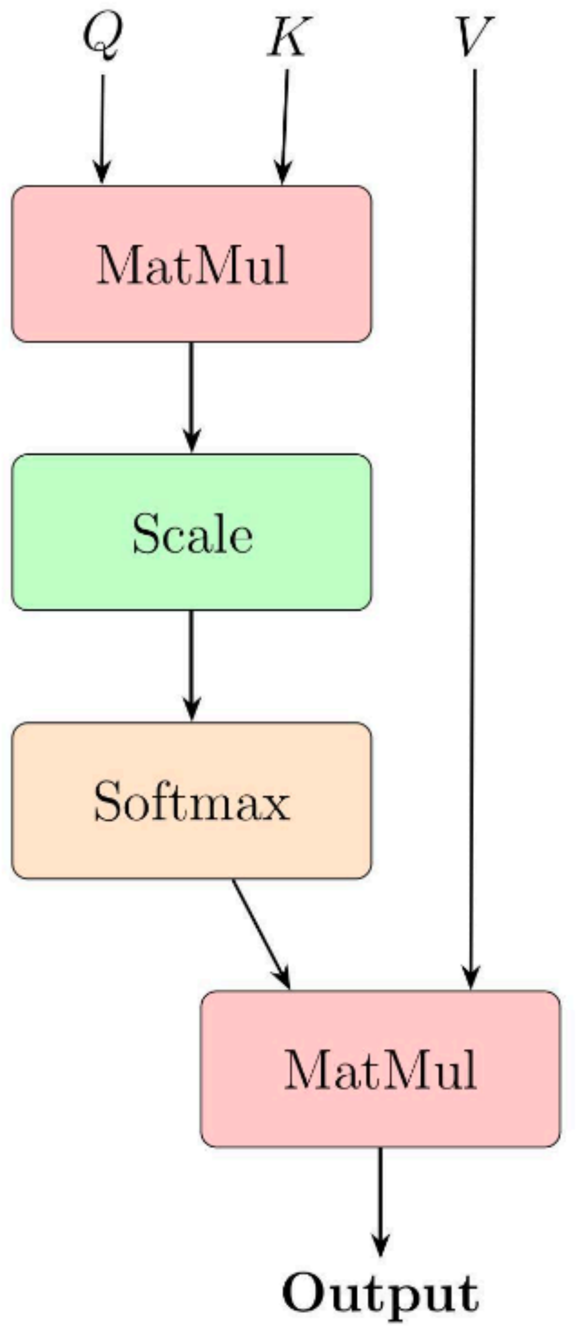


arXiv:2507.04535

Future algorithms: transformer

- Example
    - Jet tagging → examine attention between vectors of particles to classify combination





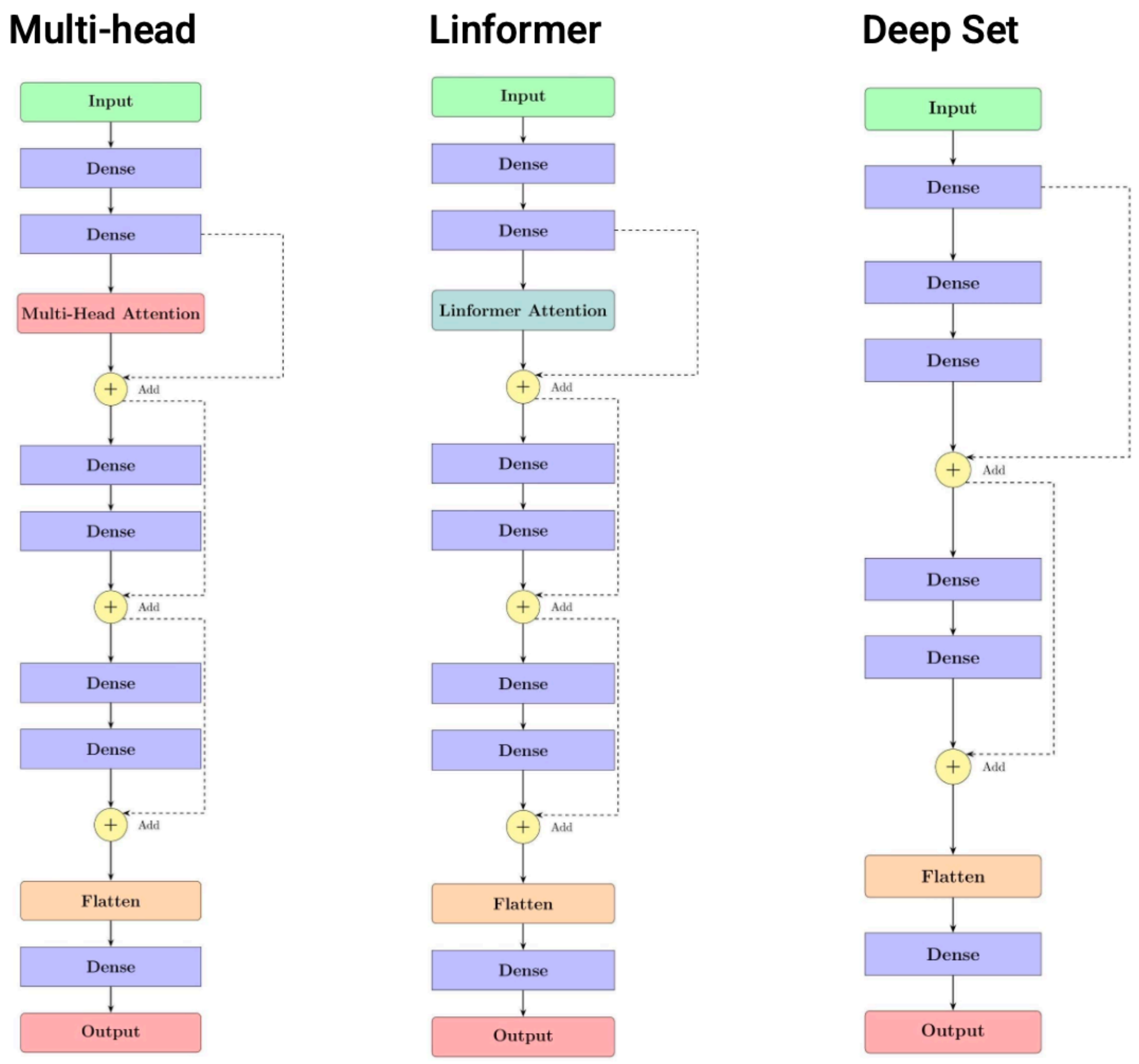
- Large numbers of particles — complexity grows $O(n^2)$

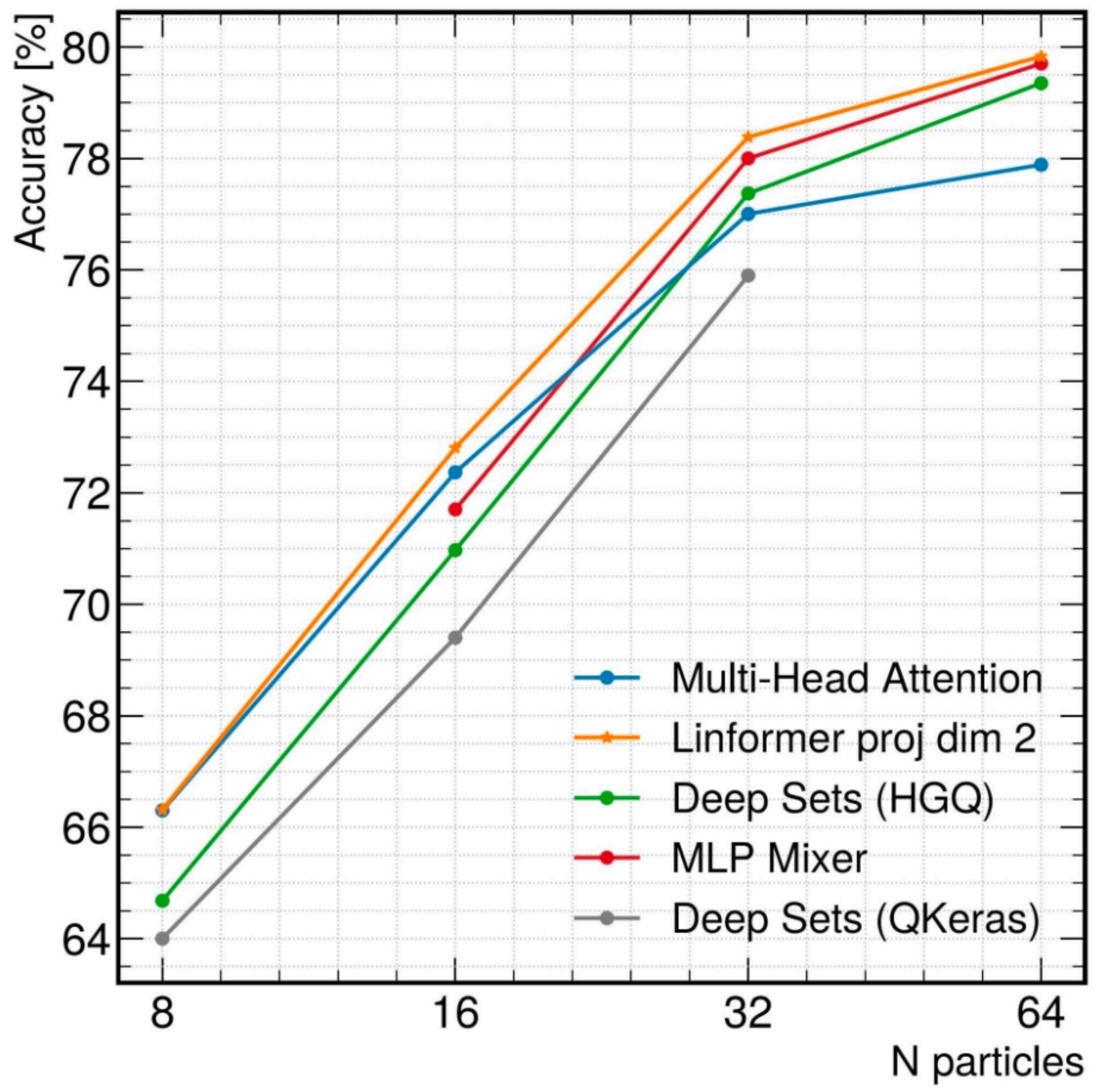
arXiv:2510.24784

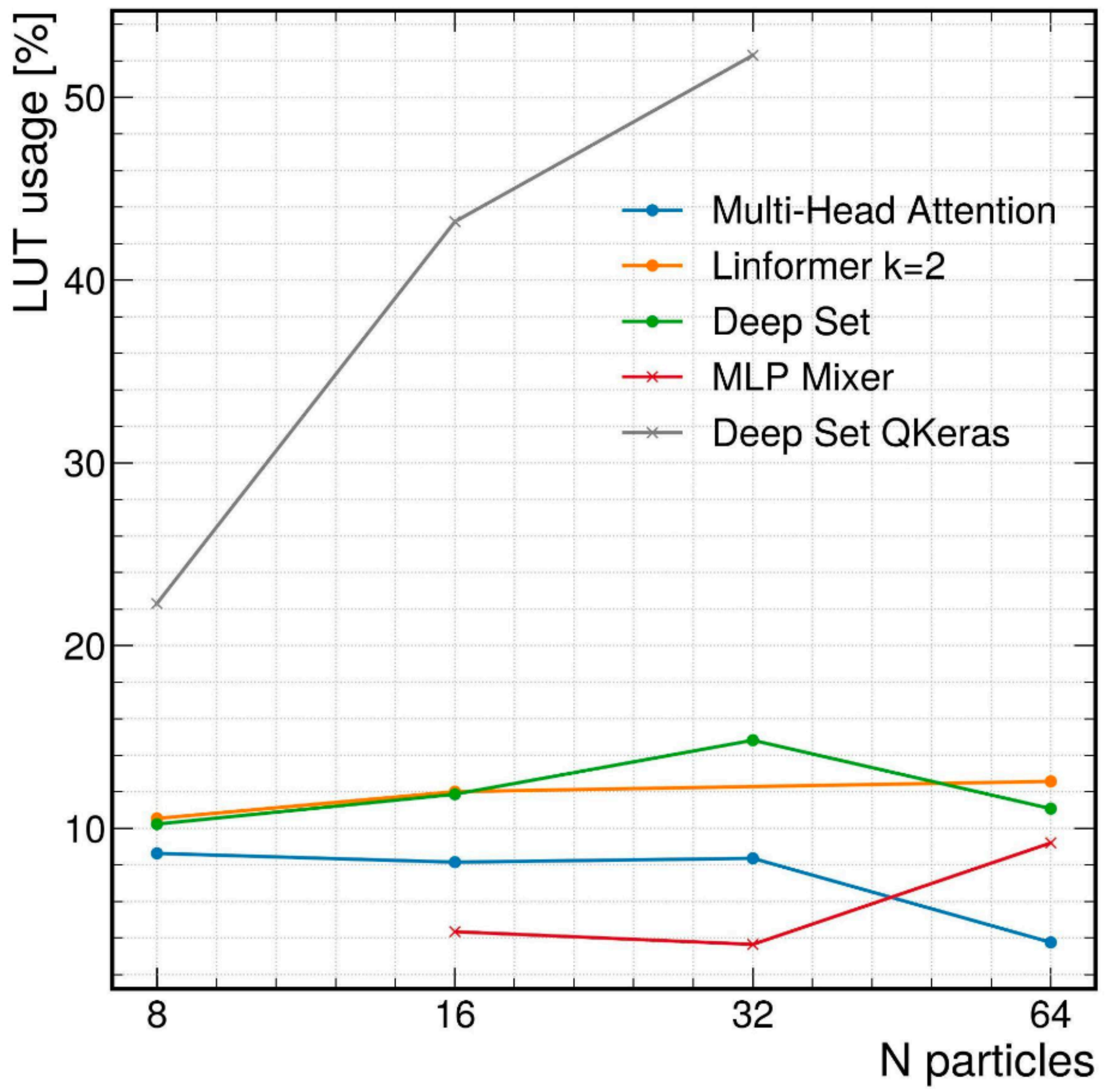
Future algorithms: transformer

- Using the tools discussed:
  - HGQ & DA — fully parallel — ~100 ns latency — 350K EBOPs
- Three low-level input features — five classes output

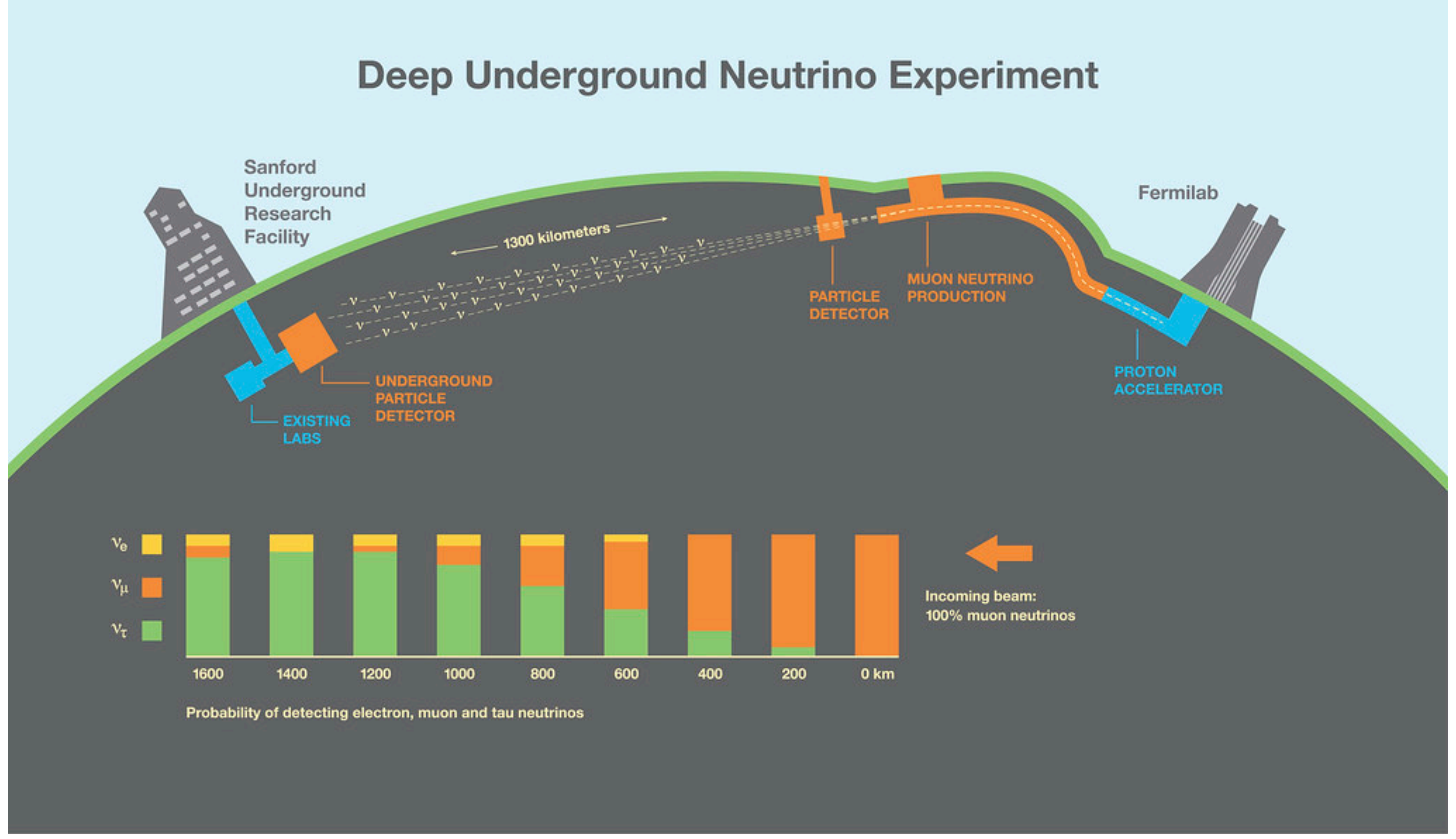
arXiv:2510.24784





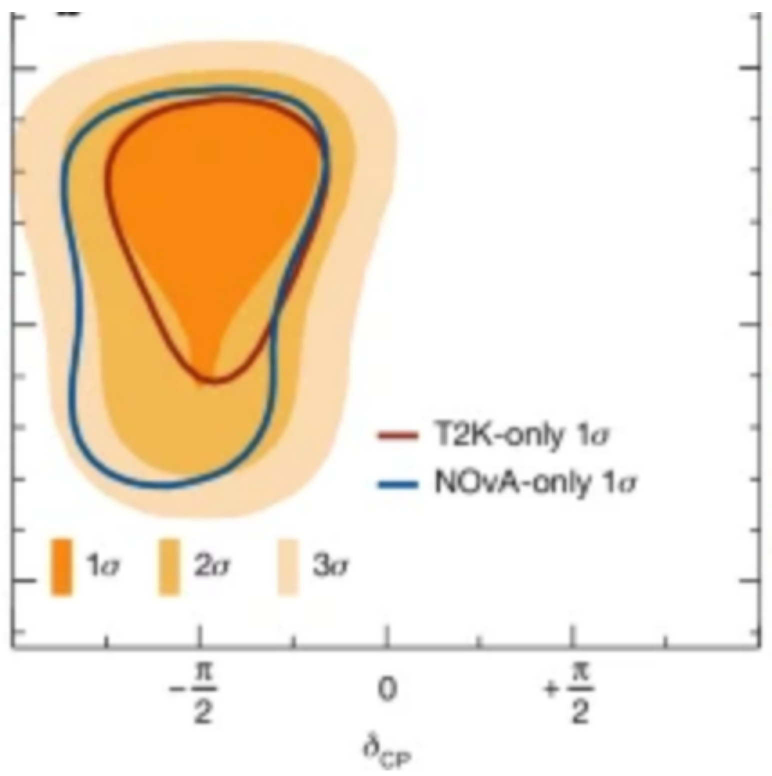


Future experiments

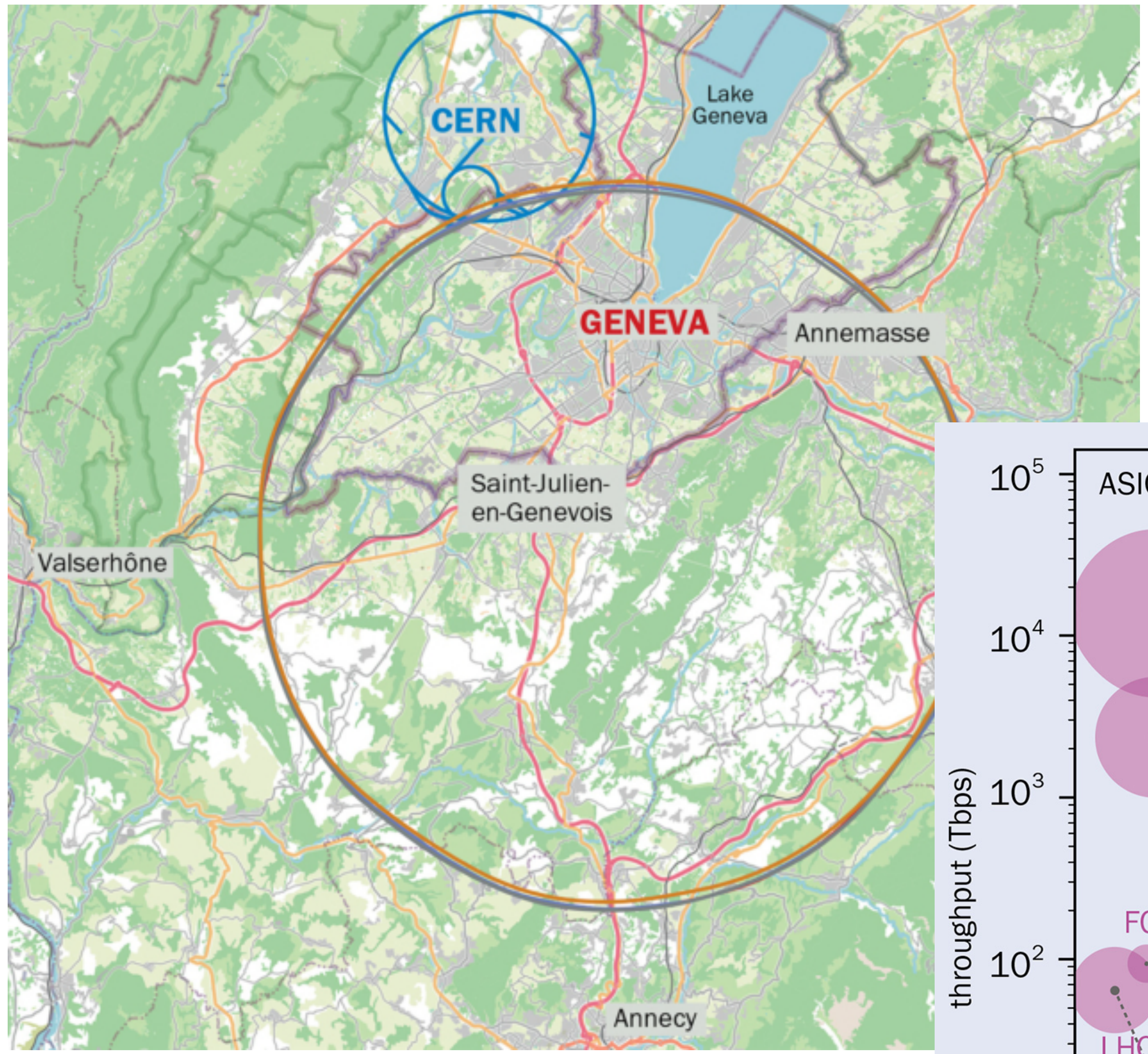


Altera

Flagship US neutrino experiment — 2030+



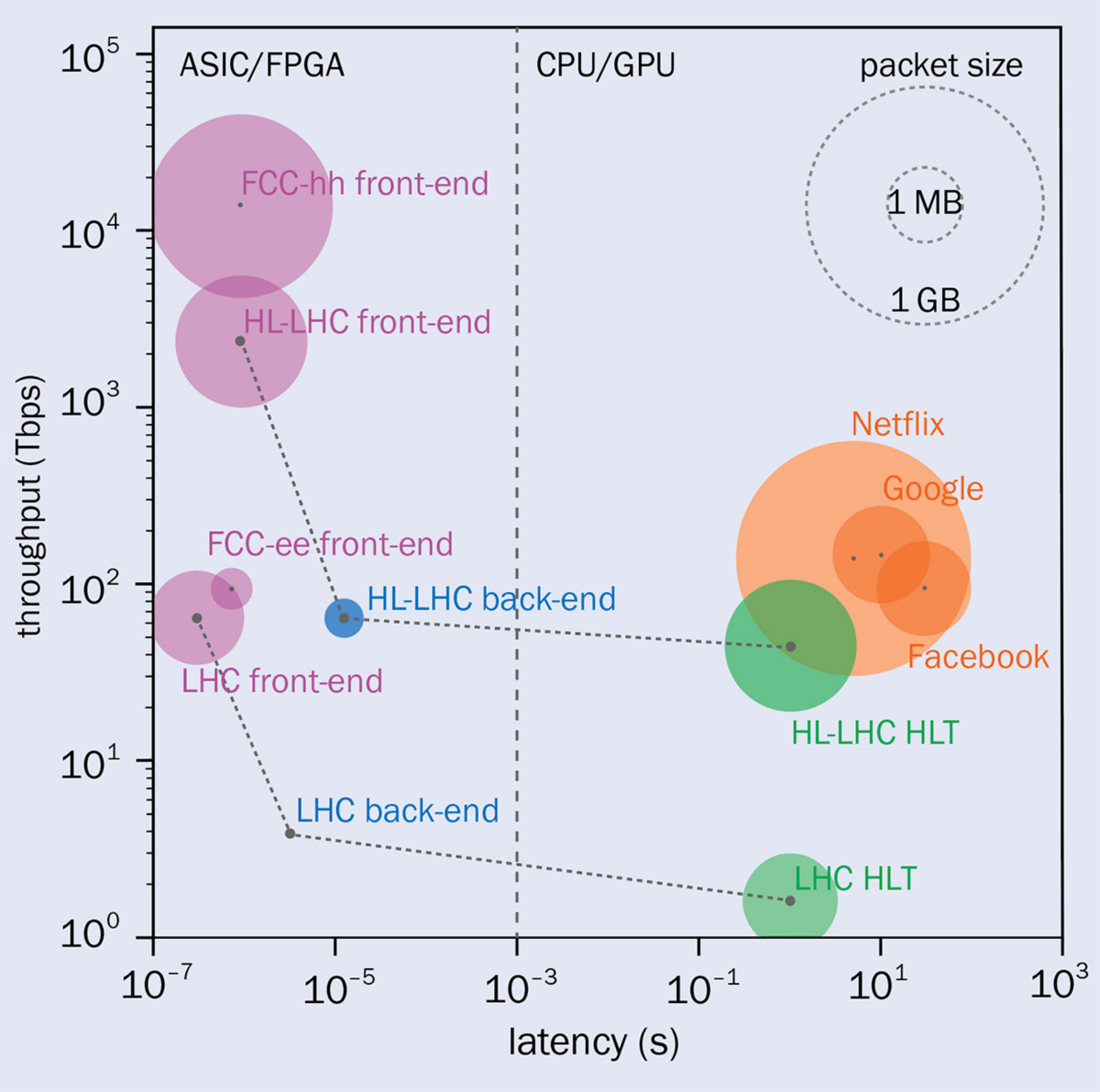
https://lbnf-dune.fnal.gov/



Flagship European proposal — 2040+

https://fcc.web.cern.ch/

Altera

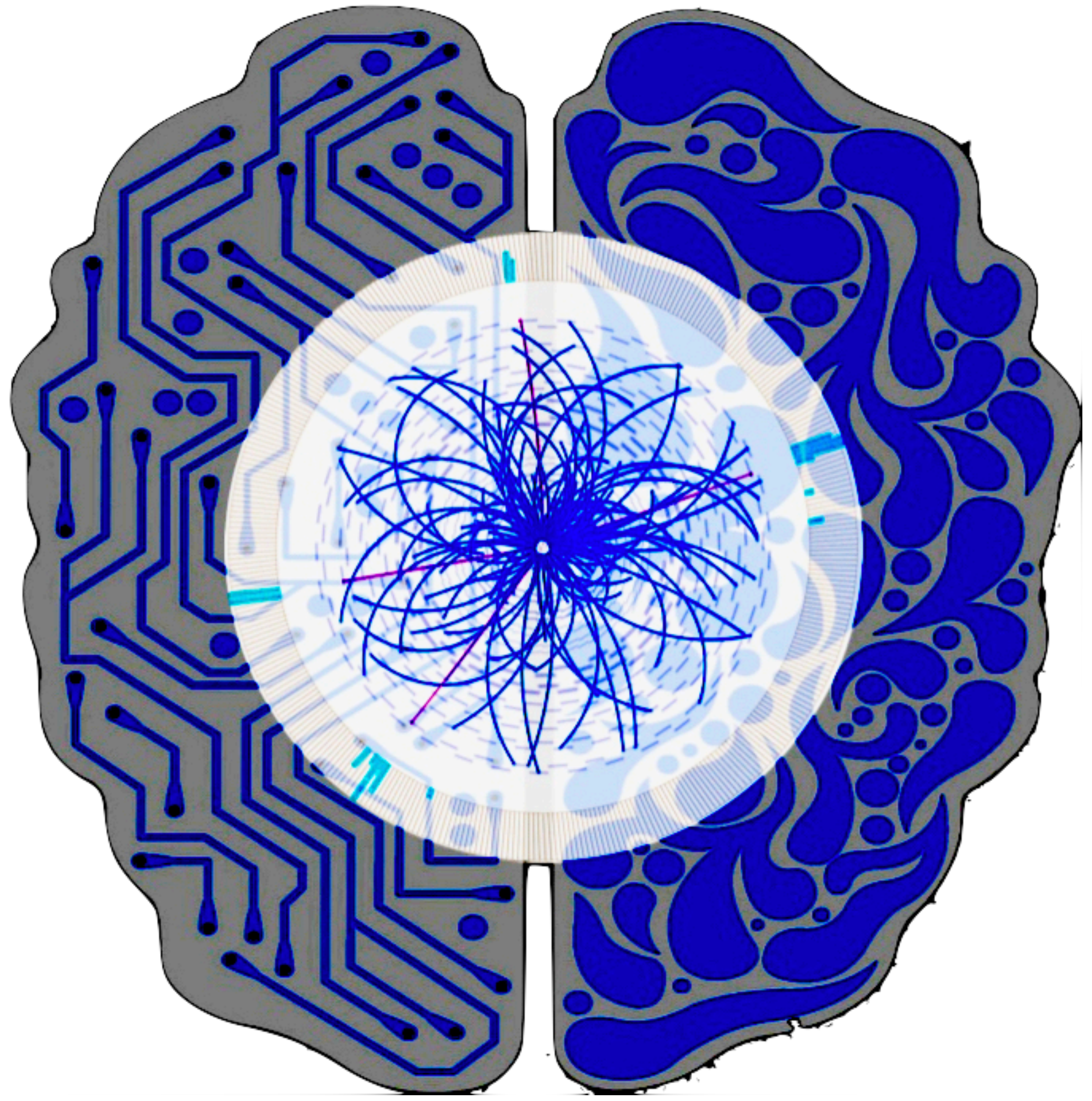


FastML community

- Growing community interested in using fast machine learning for science
- Annual workshop established with strong participation
- Particle physics, neuroscience, biology, fusion, astronomy… Altera, AMD, Nvidia, Groq, Graphcore …

FastML foundation

- Working towards a non-profit foundation to support FastML tools, training and outreach
- Advisory Board with representatives from all academic and industry members
- Currently 30+ members
- Membership through attendance at annual workshop

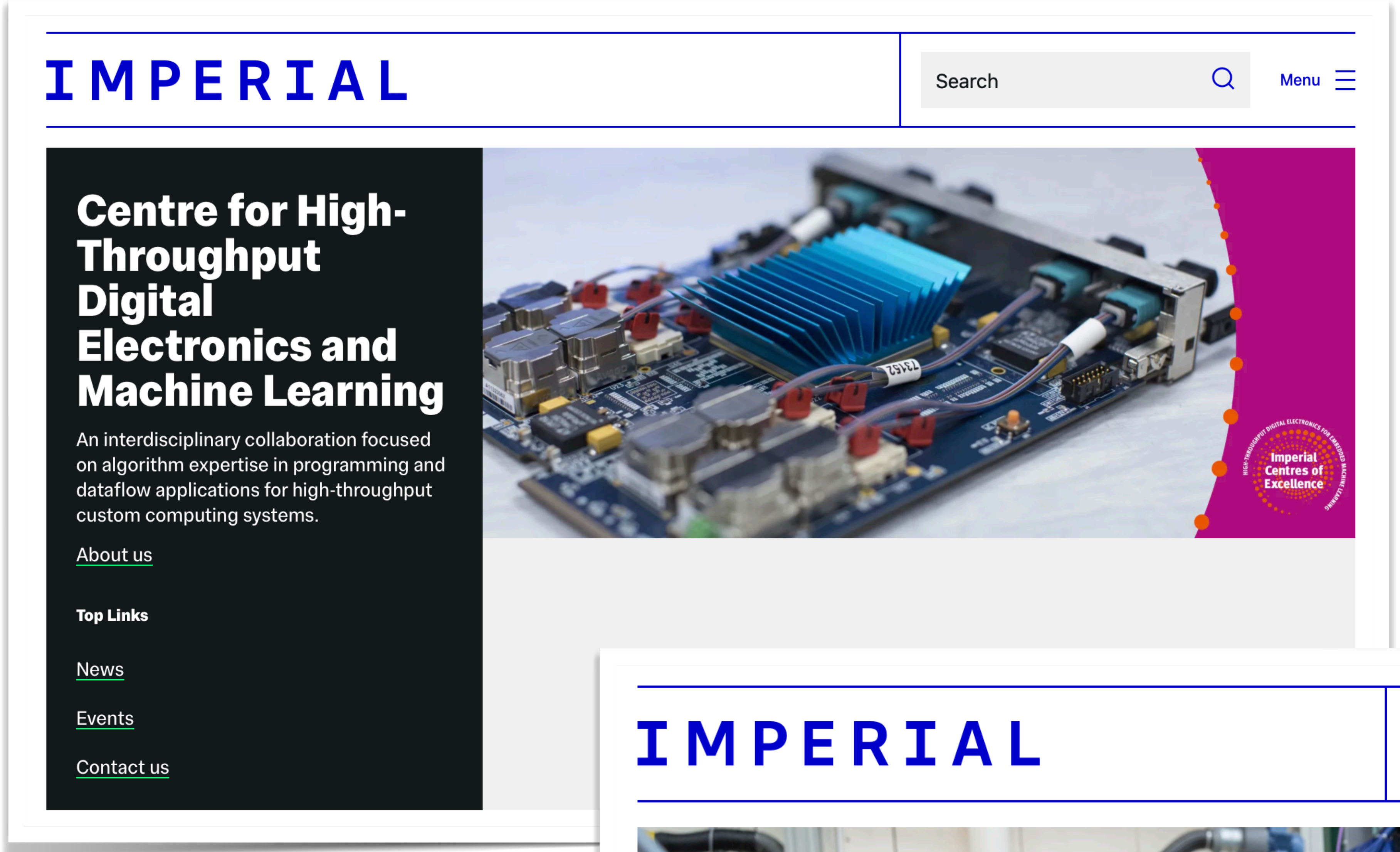


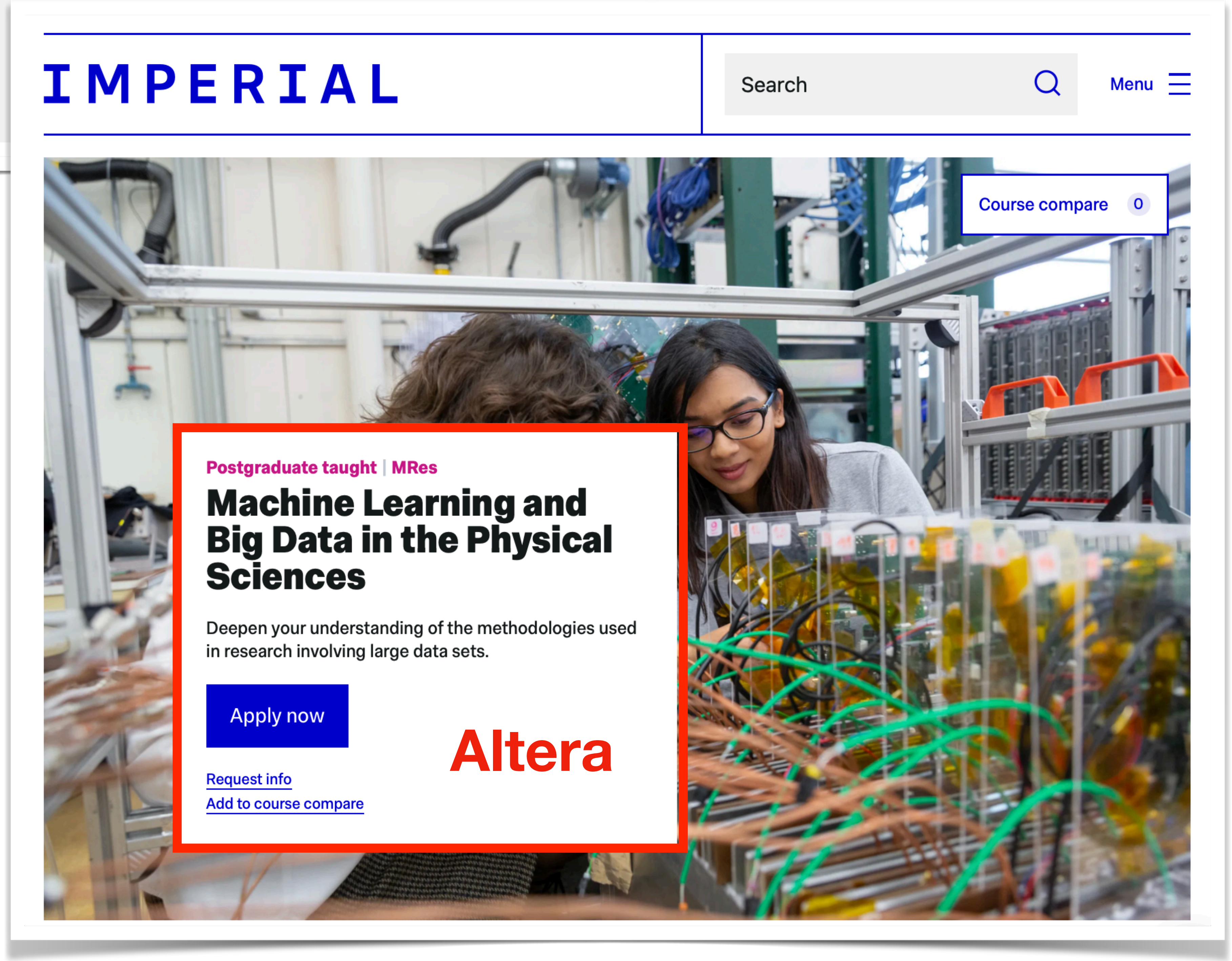
https://fastmachinelearning.org/

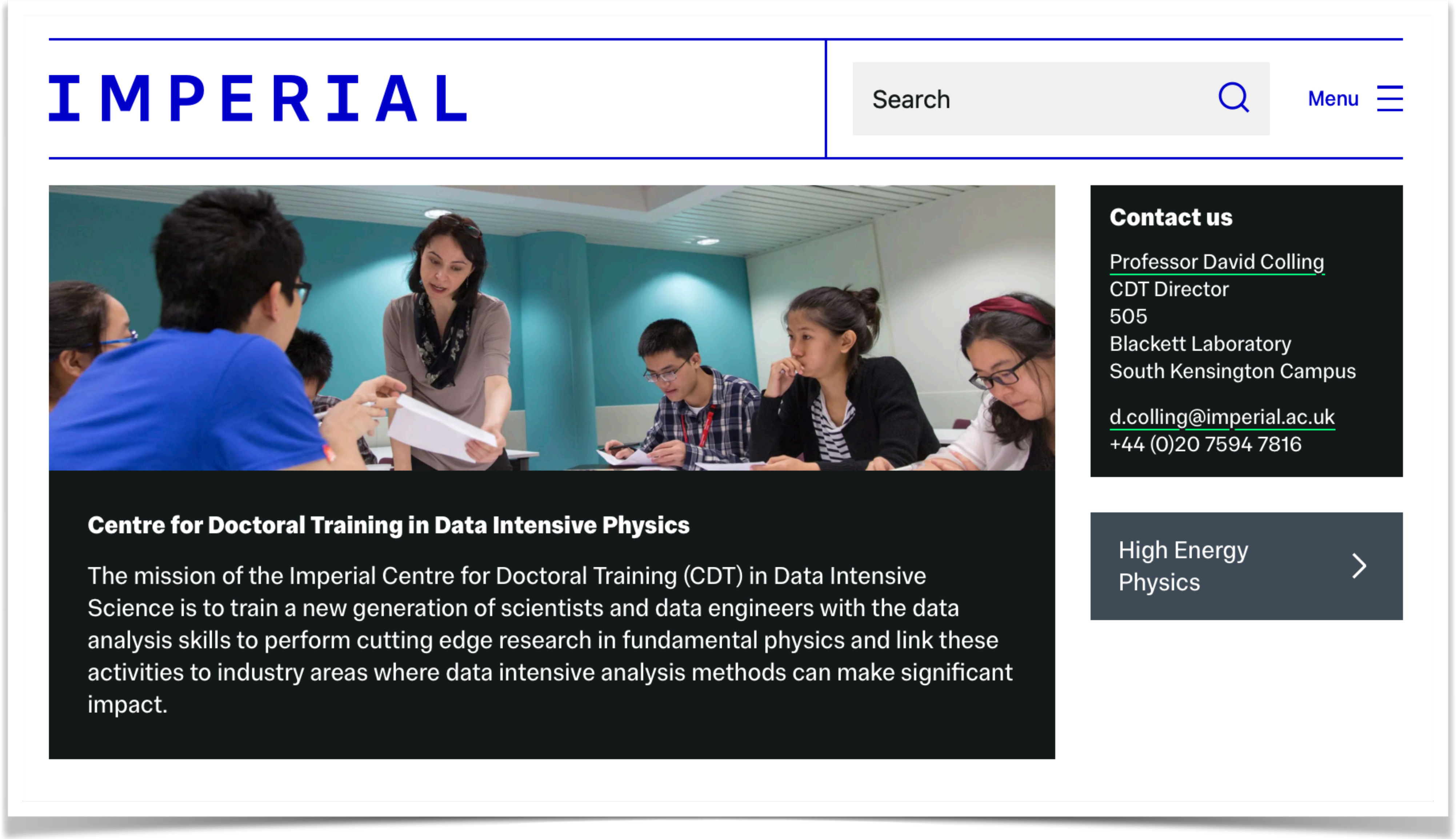
(to be updated soon …)

Opportunities

- Masters & PhD student projects
- Input into courses and materials
- Collaborate on 6 or 9 month projects with partners







Summary/conclusions

- Broad approach to future challenges: FPGA, GPU, HPC, AI, IaaS …
    - Machine learning set to be key — low latency niche is a focus
- Looking further ahead starting to plan for future experiments > 10 years away
    - Intelligent detectors — processing on detector
    - High-speed machine learning — in network processing …
    - Quantum algorithm development
- Wide range of opportunities to collaborate with industry partners