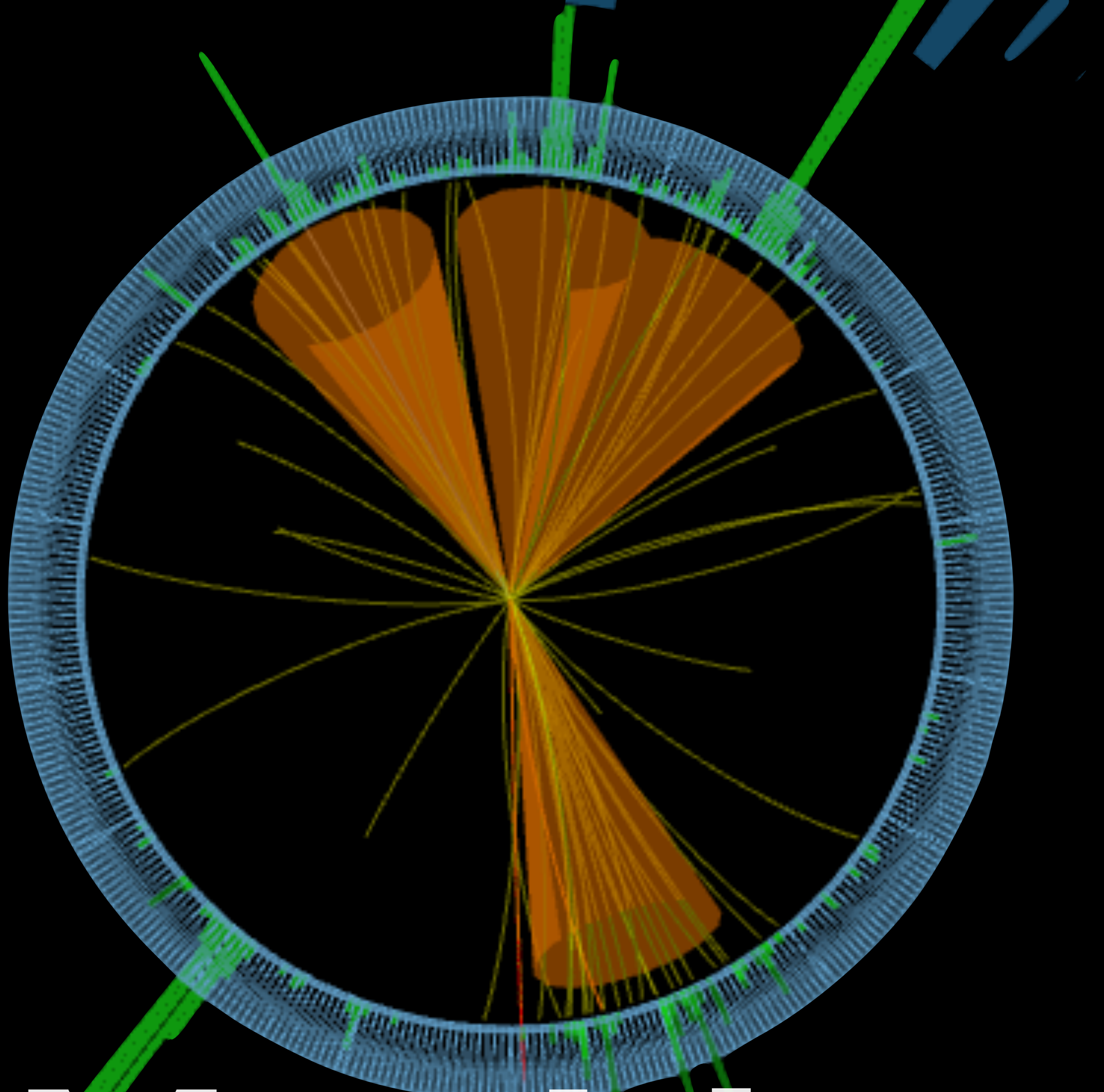
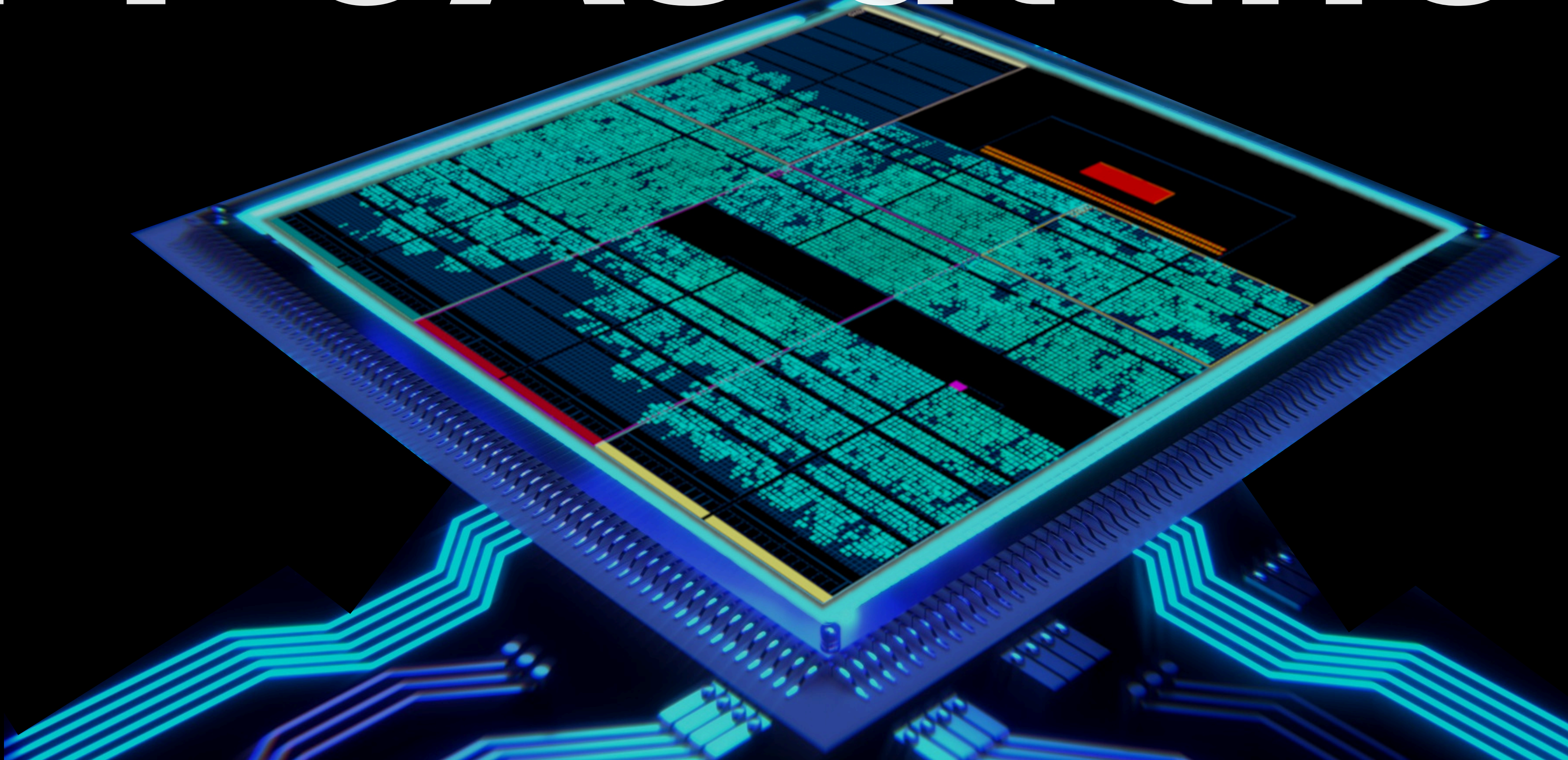
ETH zürich



Real-time Machine Learning on FPGAs at the LHC



Thea K. Årrestad (ETH Zürich)
thea.aarrestad@cern.ch
@thea_kaa

LHCP 2023

Open AI (2023)

Example of GPT-4 visual input:

User What is funny about this image? Describe it panel by panel.

Source: `https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/`

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

Open AI (2023)

Example of GPT-4 visual input:

User What is funny about this image? Describe it panel by panel.

Source: `https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/`

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

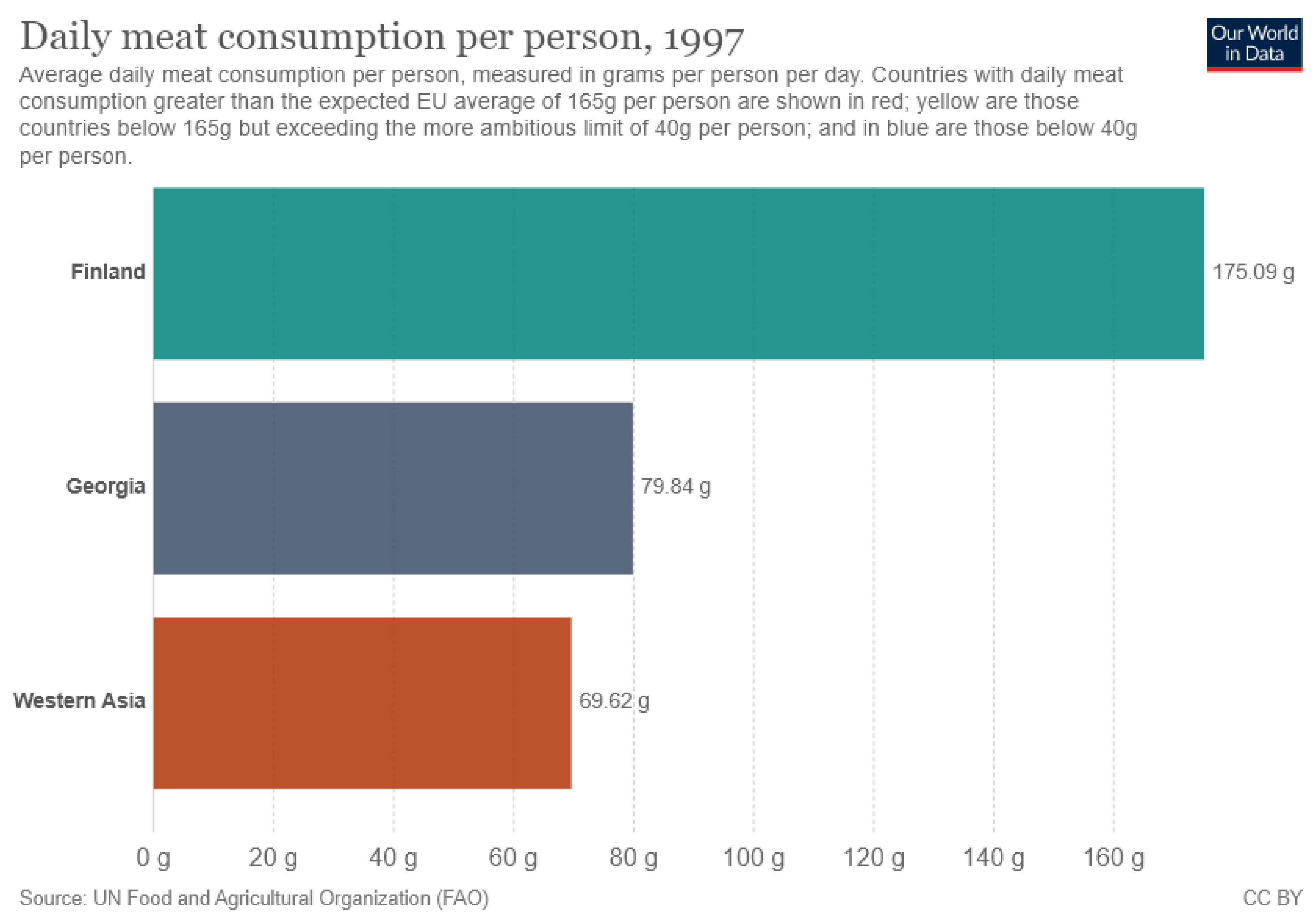
Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

GPT-4 visual input example, Chart Reasoning:

User What is the sum of average daily meat consumption for Georgia and Western Asia? Provide a step-by-step reasoning before providing your answer.



GPT-4 To find the sum of average daily meat consumption for Georgia and Western Asia, we will follow these steps:

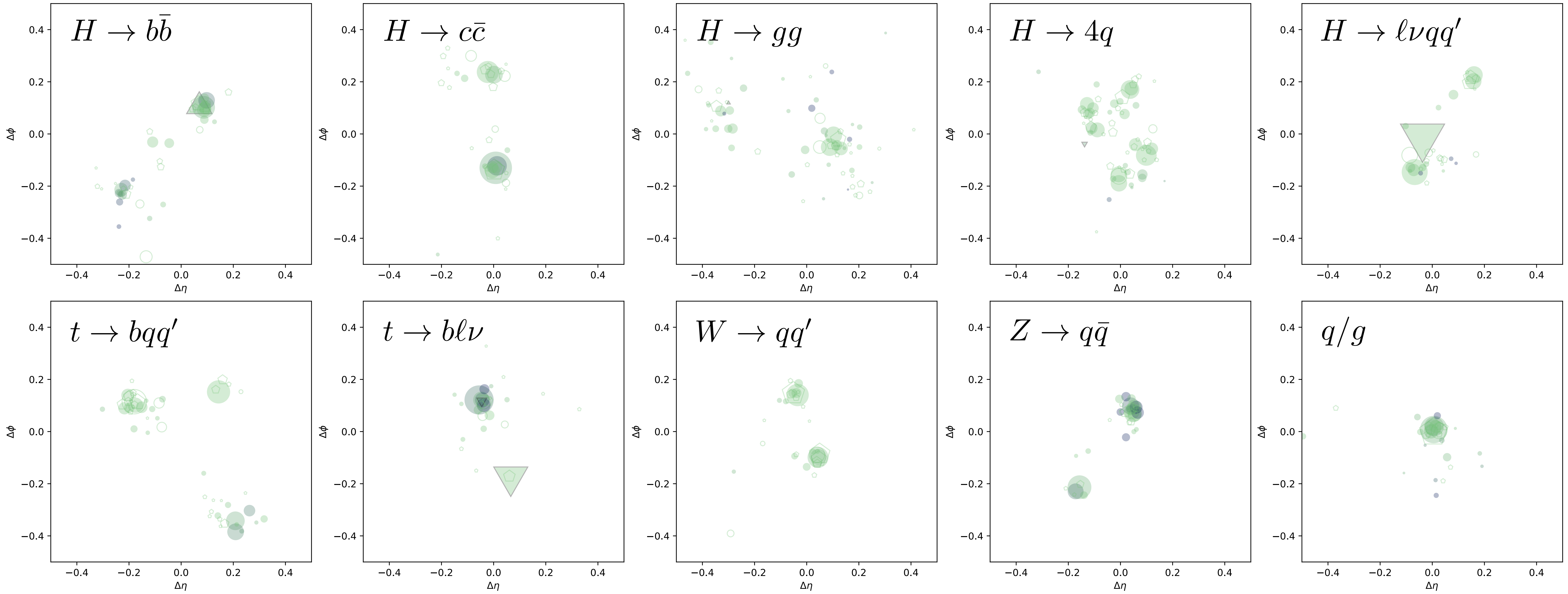
1. Identify the average daily meat consumption for Georgia.
2. Identify the average daily meat consumption for Western Asia.
3. Add the values from steps 1 and 2.

Step 1: From the provided chart, we see that the average daily meat consumption for Georgia is 79.84 grams per person per day.
Step 2: From the chart, we see that the average daily meat consumption for Western Asia is 69.62 grams per person per day.
Step 3: Add the values from steps 1 and 2.
79.84 g (Georgia) + 69.62 g (Western Asia) = 149.46 g

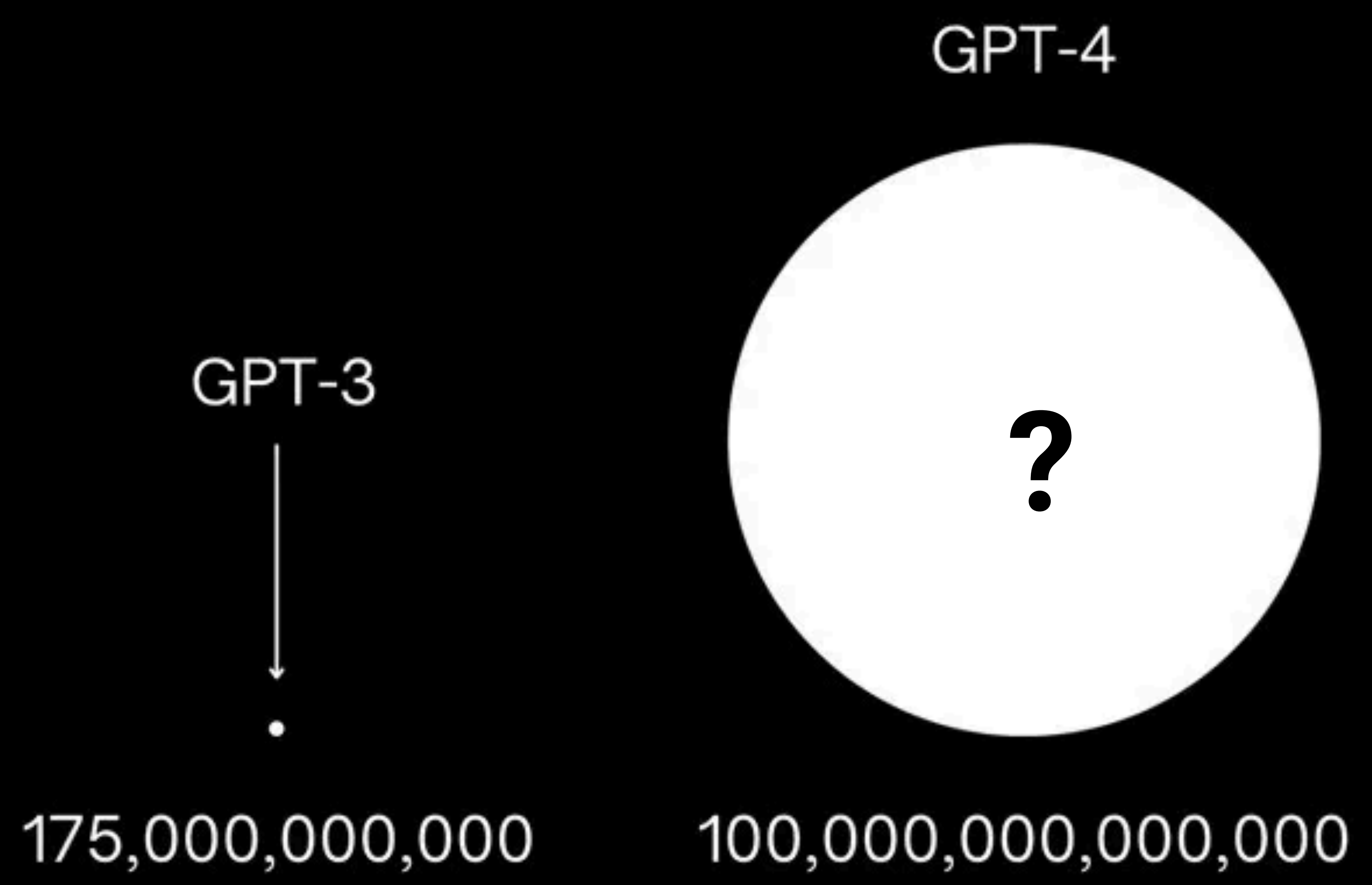
So, the sum of average daily meat consumption for Georgia and Western Asia is 149.46 grams per person per day.

100 million jets for training

	Accuracy	# params
PFN	0.772	86.1 k
P-CNN	0.809	354 k
ParticleNet	0.844	370 k
ParT	**0.861**	2.14 M
ParT (plain)	0.849	2.13 M



"Particle Transformer For Jet Tagging" H. Qu, C. Li, S. Qian



(0.16% of neurons in your brain) (All the neurons in your brain?)

NVIDIA Press Release 2020

Train (GPT-3):

- **285,000 CPU cores**
- **10,000 GPUs**
- **400 Gb/s network**
- **Several weeks**
- **Trained on ~25 km high book of text**

NVIDIA Press Release 2020

Train (GPT-3):

- 285,000 CPU cores
- 10,000 GPUs
- 400 Gb/s network
- Several weeks
- Trained on ~25 km high book of text

What should I include in my talk on fast inference at LHCP?

When preparing a talk on fast inference at the LHCP (Large Hadron Collider Physics) conference, it's essential to focus on key aspects related to efficient and rapid processing of data in high-energy physics experiments. Here are some suggested topics and points to include in your talk:

1. Introduction:
   - Briefly explain the significance of fast inference in high-energy physics experiments.
   - Discuss the challenges posed by the large volume of data generated by particle detectors and the need for real-time or near-real-time analysis.
2. Data Reduction and Selection:
   - Highlight the importance of efficient data reduction techniques to handle the vast amounts of raw data.
   - Discuss the strategies employed to select and prioritize interesting physics events for further analysis.
   - Emphasize the role of fast inference algorithms in data selection, such as triggering systems.
3. Machine Learning for Fast Inference:
   - Introduce machine learning methods and their applications in fast inference.
   - Discuss the use of deep learning techniques for real-time analysis, including neural network architectures suitable for rapid decision-making.
   - Highlight specific machine learning algorithms or frameworks that have been

Inference (GPT-3):

- ?

AI & Memory Wall

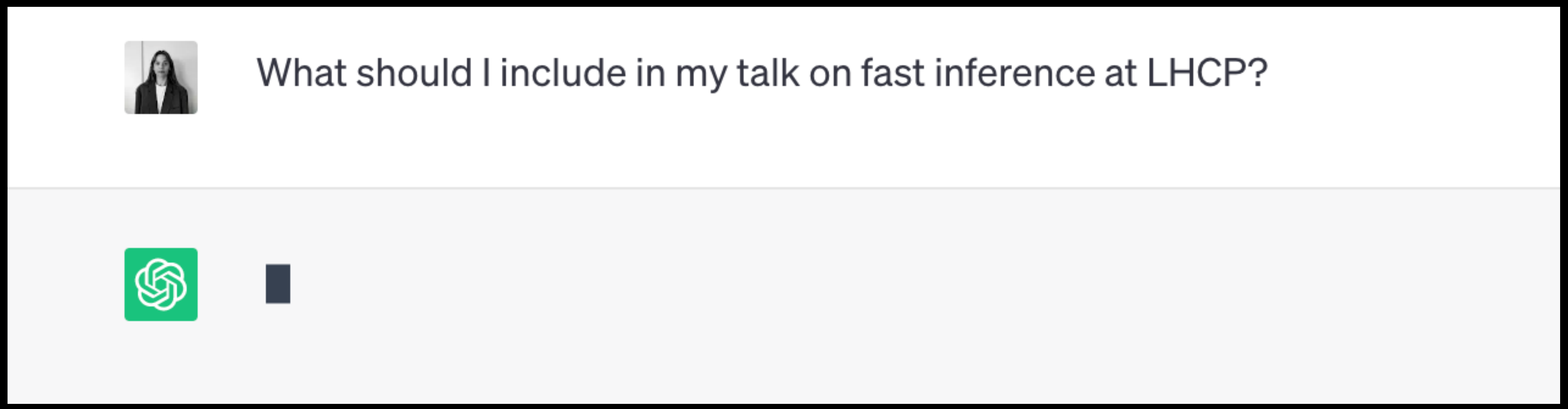
Resources: ***11 interconnected GPUs***

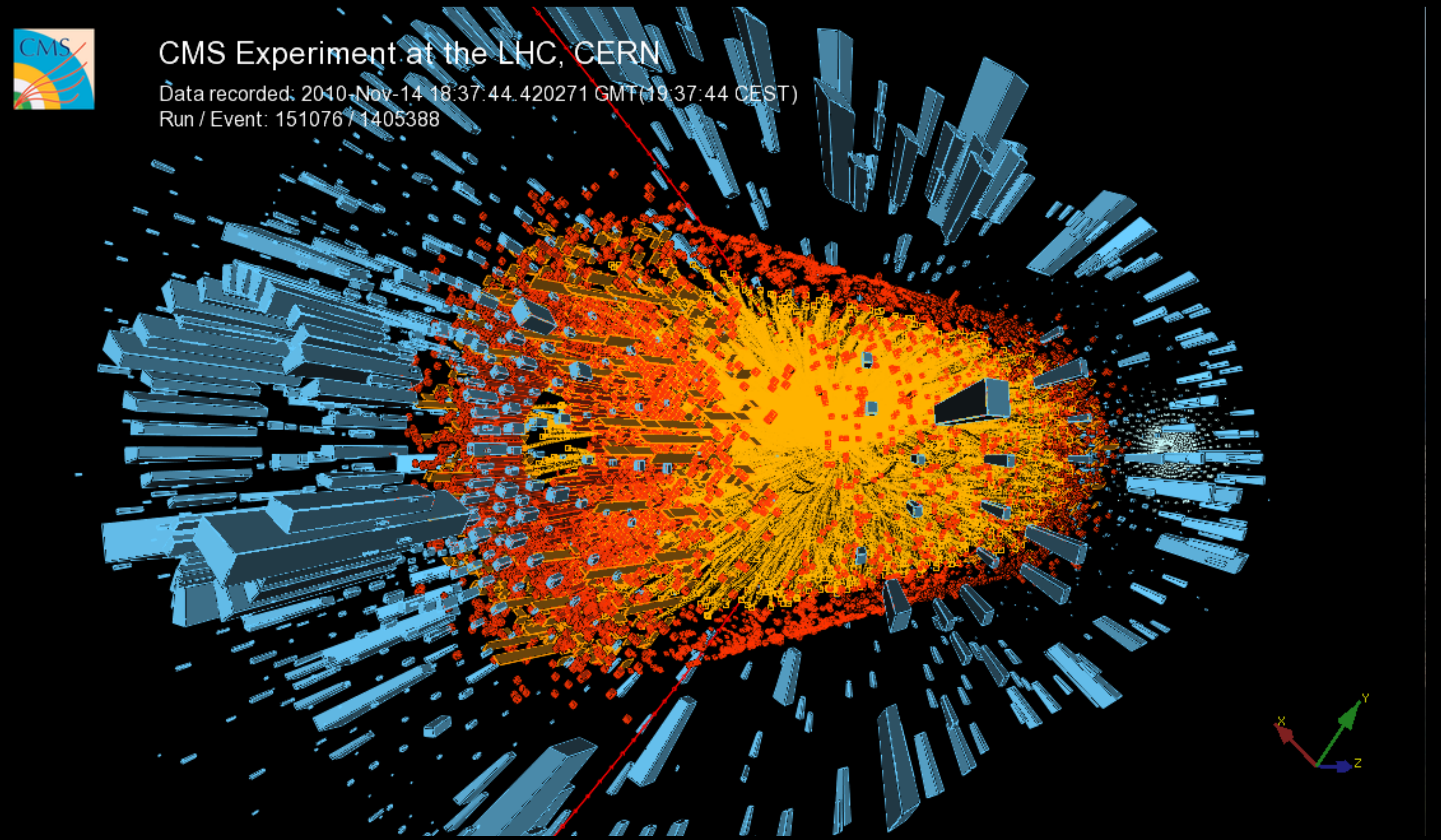
Latency : ***10^1 seconds***

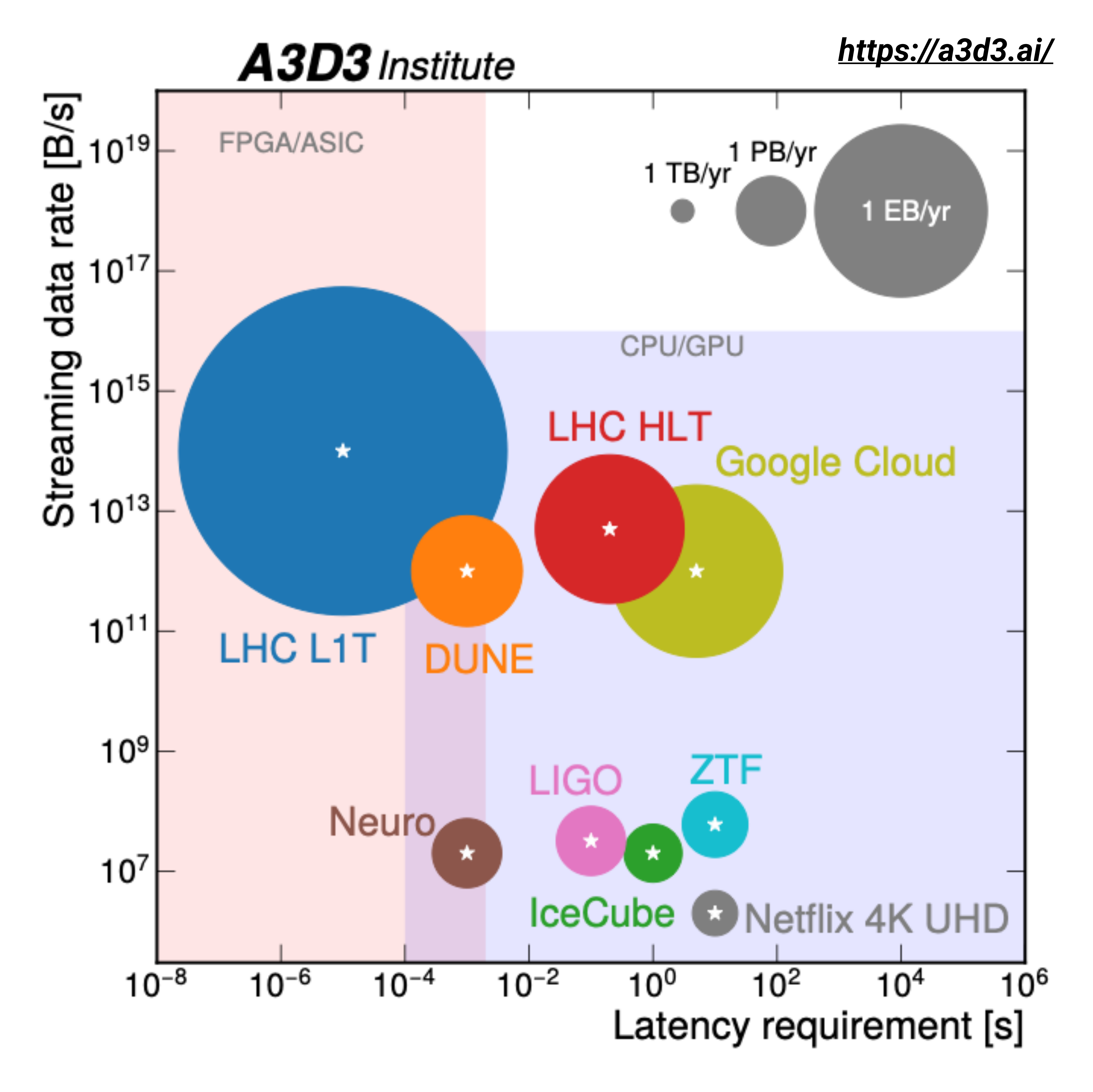
What should I include in my talk on fast inference at LHCP?

Resources: 11 interconnected GPUs
Latency : 10^{1} seconds

Resources: One single chip
Latency: 10^{-9} seconds







A3D3 Institute
https://a3d3.ai/
Streaming data rate [B/s]
Latency requirement [s]
FPGA/ASIC
CPU/GPU
1 TB/yr
1 PB/yr
1 EB/yr
LHC L1T
DUNE
LHC HLT
Google Cloud
LIGO
ZTF
Neuro
IceCube
Netflix 4K UHD


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 billion collisions per second
~1 PB of data per second
X
Y
Z


Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
~1 billion collisions per second
~1 PB of data per second

To read out, would need:

- huge computer farms (and big money!)

but also:

– must get all data out from detector

– must supply detector with much power

– resolution degradation due to amount of material in detector

See more in M. Fontana's talk

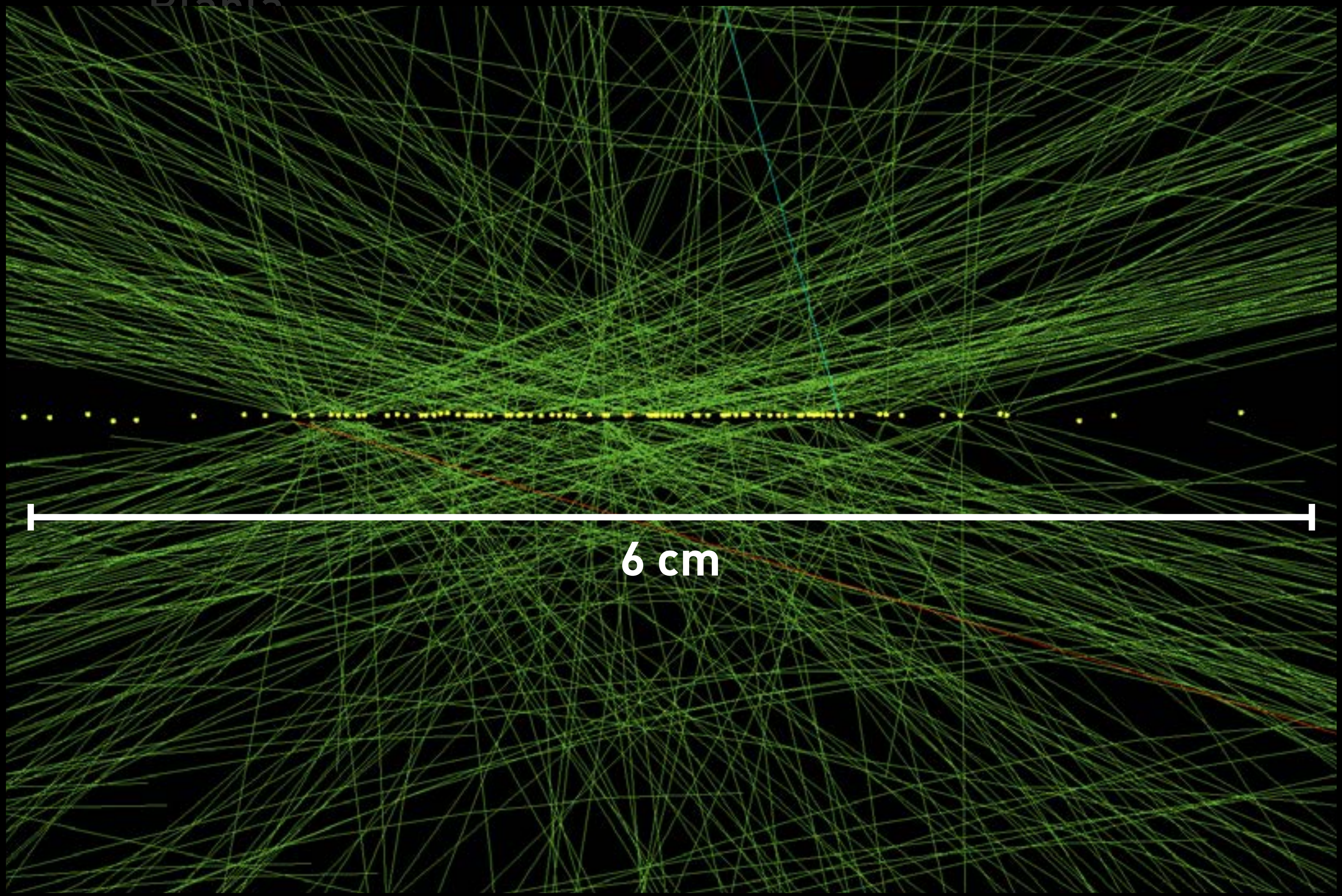

Geneva
Geneva Lake
LHCb
CMS
LHC
2 Tb/s to L1
Data temporarily stored
in detector electronics for ~4 μs
L1 trigger:
Decide which
event to keep
within ~4 μs
XC5VLX30

TIER 0: ∞
Geneva
~0.0025% of collision events remaining
High Level Trigger:
Latency O(100) ms
Geneva
LHCb
DATA
1 kHz
~Gb/s
ALICE
CMS
DATA
100(110) kHz
~Tb/s
LHC

intel
Xeon® 7500
TESLA
GPU/CPU
SOFTWARE
Geneva
Geneva Lak
TIER 0: ∞
LHCb
ATLAS
ALICE
CMS
ASIC
HARDWARE
FIRMWARE
FPGA
LHC
VIRTEX-5

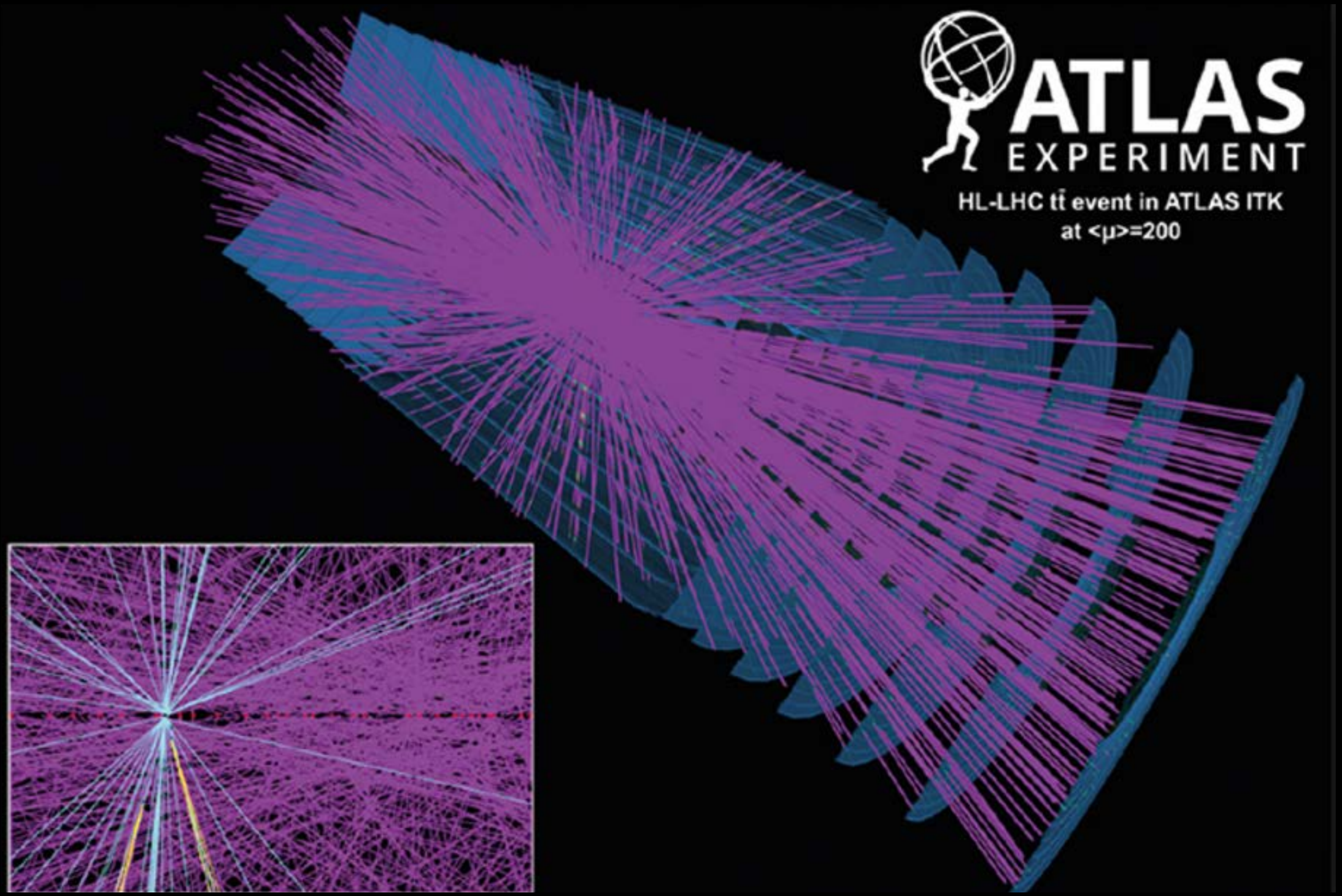
LHC

78 vertices
(average 60)



High Luminosity LHC

200 vertices
(average 140)

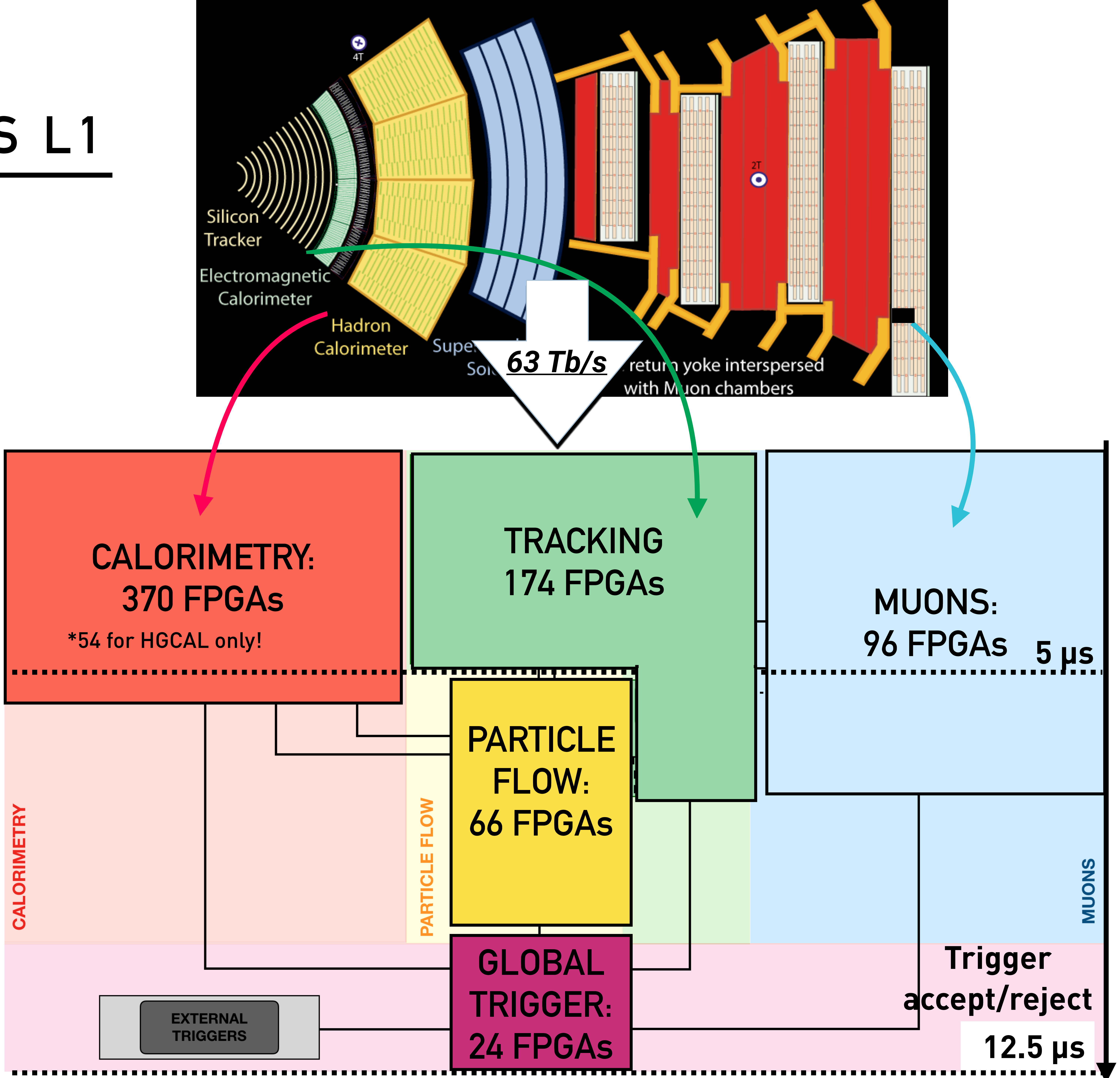


2022	2023	2024	2025	2026	2027	2028	2029	...	2037	2038
LHC				MAJOR UPGRADE			HL-LHC			

Run 3

Run 4+5

HL-LHC: CMS L1



To cope with increased data complexity, new at L1:

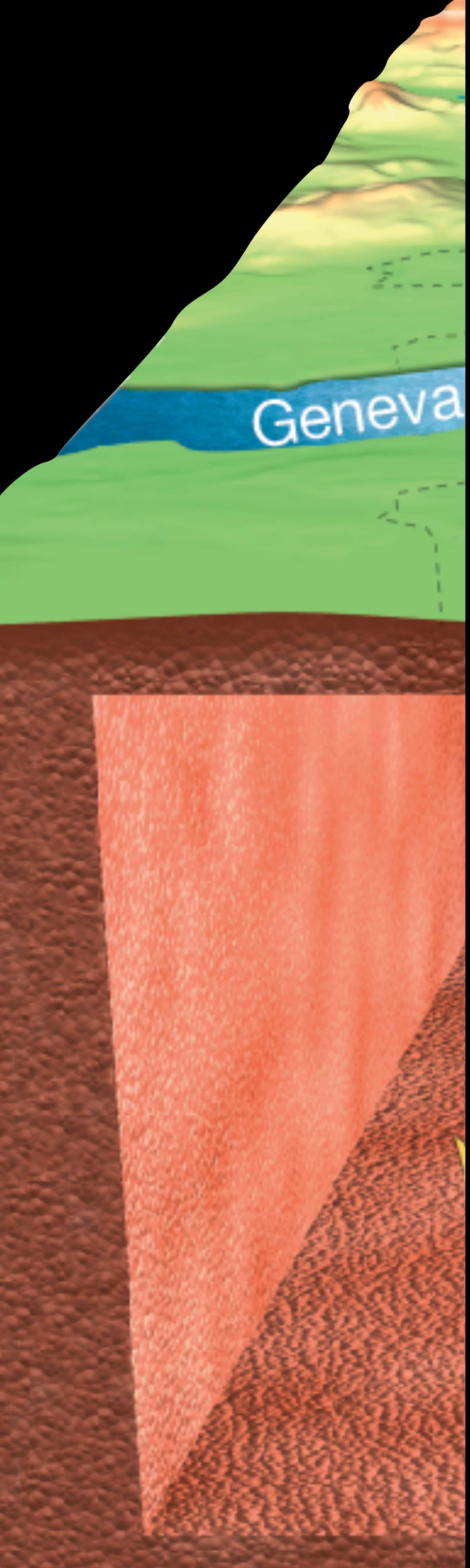
- Tracking
- Particle Flow
- O(1)M channel HGCal

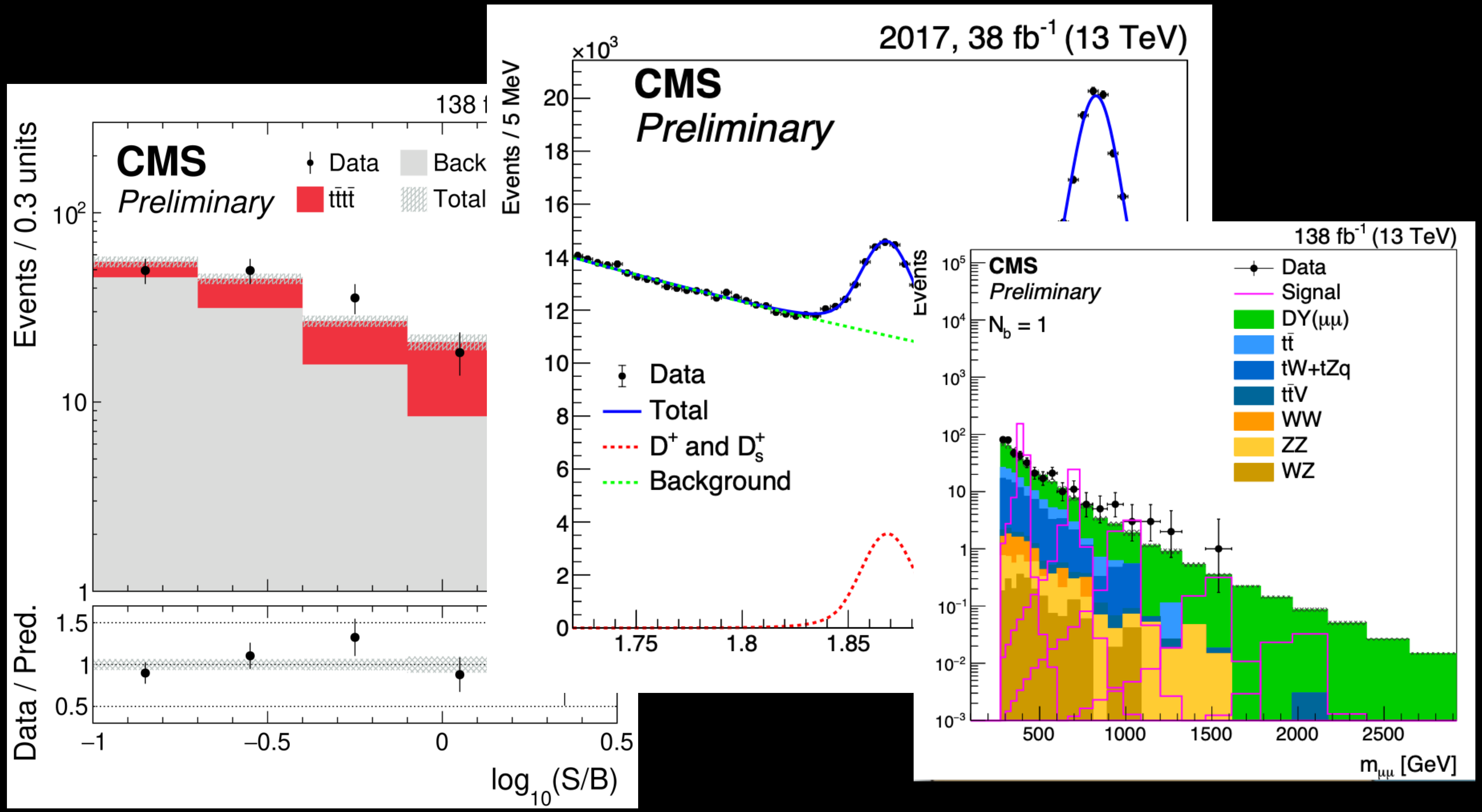
Input data

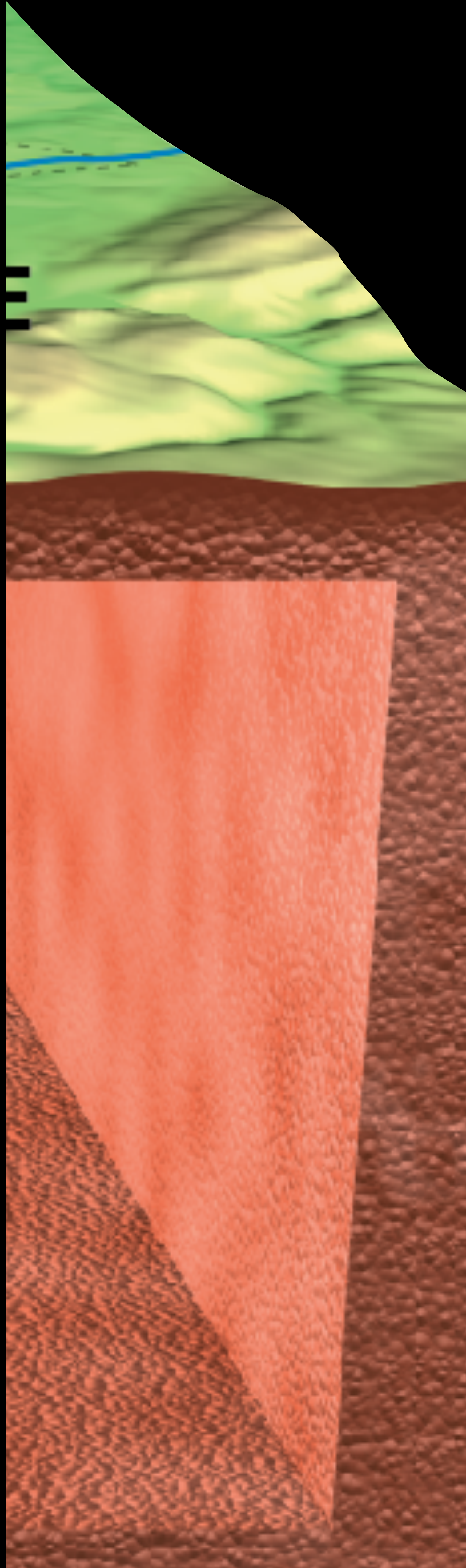
- 2 Tb/s → 63 Tb/s

Latency

- 4 μs → 12 μs







To make sure we select "the right" 0.0025%, algorithms must be

- Fast (get more data through)
- Accurate (select the right data)


Geneva
Geneva Lake
LHCb
ATLAS
ALICE
CMS
EDGE INFERENCE
Fast inference on specialised hardware
ASIC inference
FPGA inference
Level-1 trigger:
Latency O(1) μs
Detector:
40 MHz
~Pb/s

Programming an FPGA

C/C++ algorithm

Constraints/ Directives

Xilinx

Vitis HLS

Vivado HLS

Vivado Accelerator

Intel

Intel HLS

Mentor

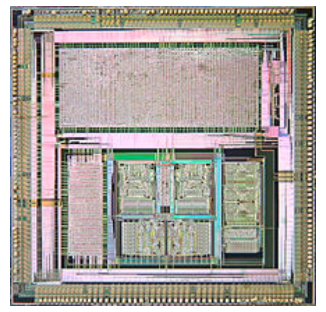
Catapult HLS

(CMS, ATLAS)

VHDL/Verilog

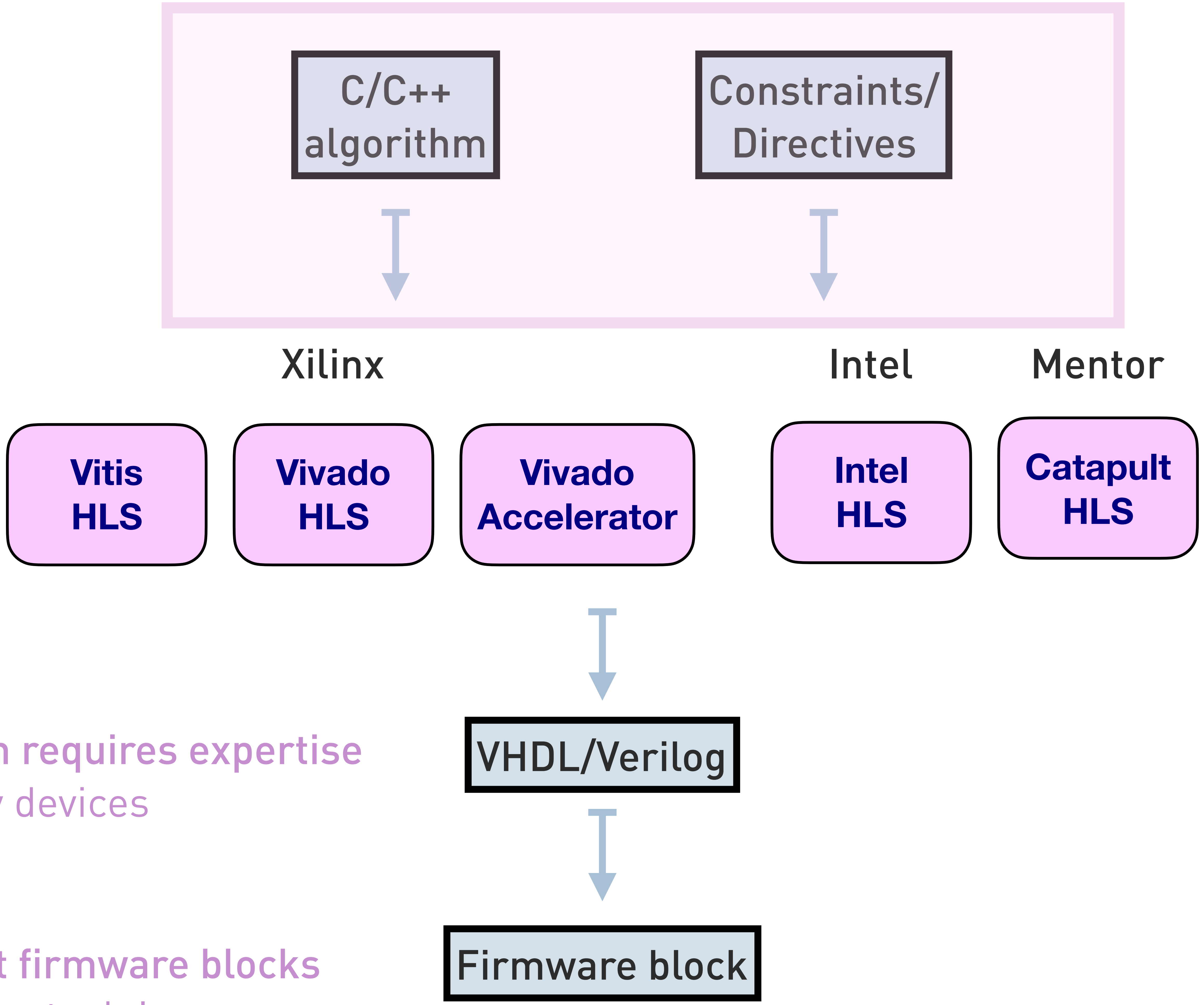
Firmware block

(LHCb, ATLAS?)



(ASICs)

Programming an FPGA



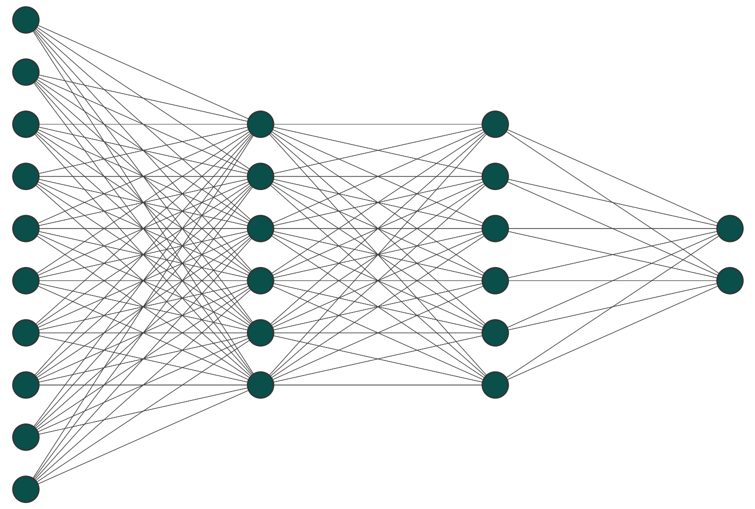
Efficient L1T firmware design requires expertise

- FPGA deployment in busy devices
- ≪ 1μs latency target

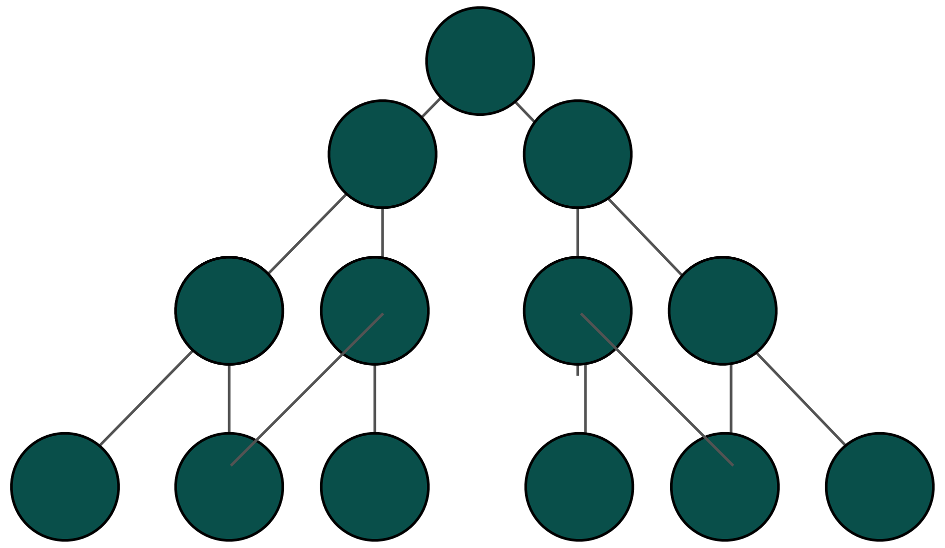
Translating NNs into efficient firmware blocks

- Not well served by industry tools!

TensorFlow / TF Keras / PyTorch / ONNX



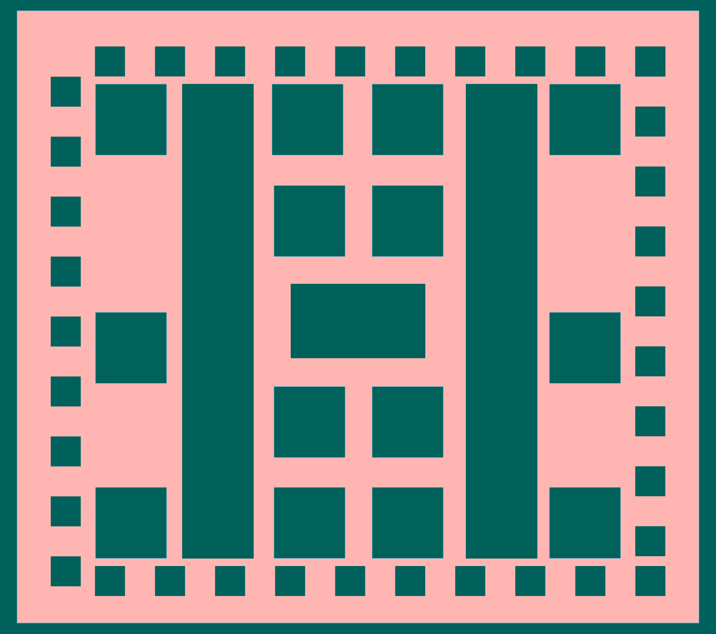
scikit-learn / XGBoost / TMVA

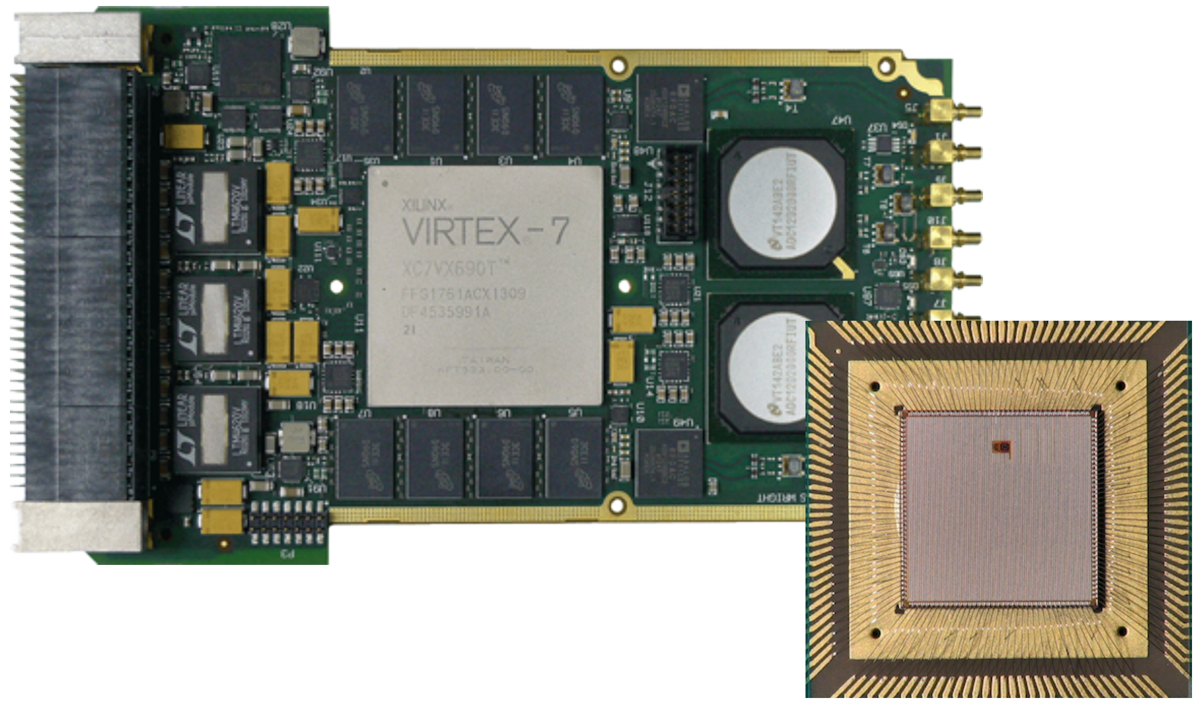


→ hls4ml →

→ Conifer →

HLS project:
Xilinx Vitis HLS, Intel Quartus HLS,
Mentor Catapult HLS






```
pip install hls4ml
pip install conifer
```

K + TensorFlow

ONNX

PYTORCH

Model (quantized/pruned)

Quantized:

QONNX

QKERAS

hls4ml

Keras + TensorFlow

ONNX

PYTORCH

Model (quantized/pruned)

→

Convert model to internal representation

Write HLS project targeting specified backend (configurable parallelization/ quantization)

Run emulation

Run synthesis

Quantized:

QONNX

QKERAS

Vivado™ HLS

Mentor® A Siemens Business

Catapult (soon!)

intel HLS COMPILER

ONNX

PYTORCH

Model (quantized/pruned)

Quantized:

QONNX

QKERAS

hls4ml

Convert model to internal representation

Write HLS project targeting specified backend (configurable parallelization/ quantization)

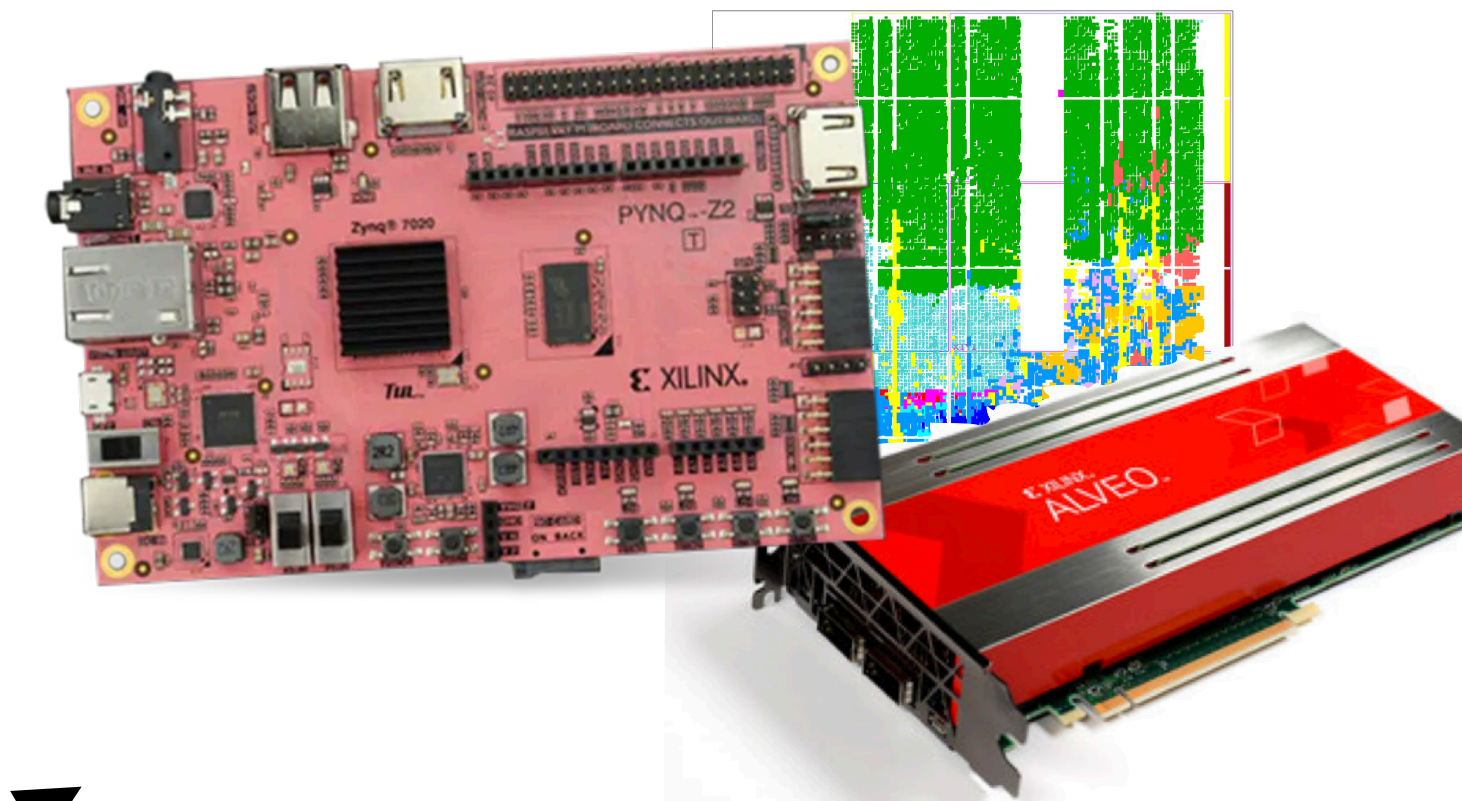
Run emulation

Run synthesis

Vivado™ HLS

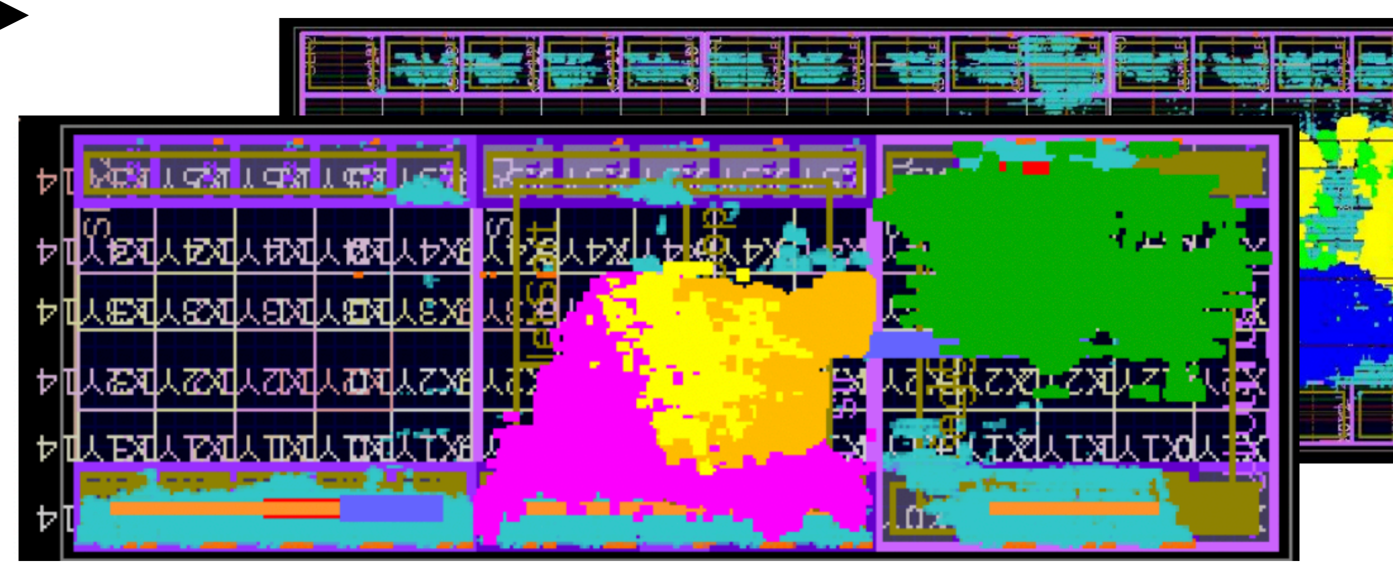
Mentor
A Siemens Business
Catapult (soon!)

HLS COMPILER

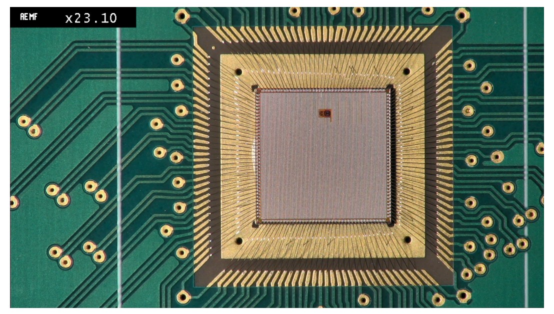


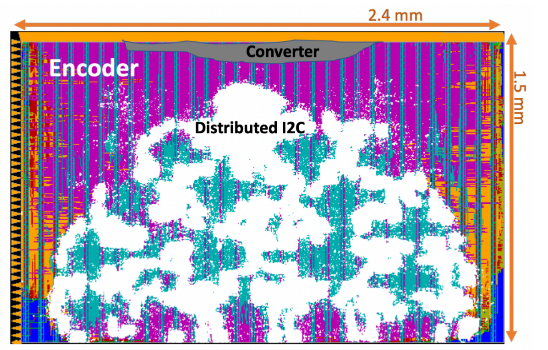
Co-processing kernel (Xilinx accelerators/SoCs)

FPGA custom designs (eg trigger algorithms)



ASICs

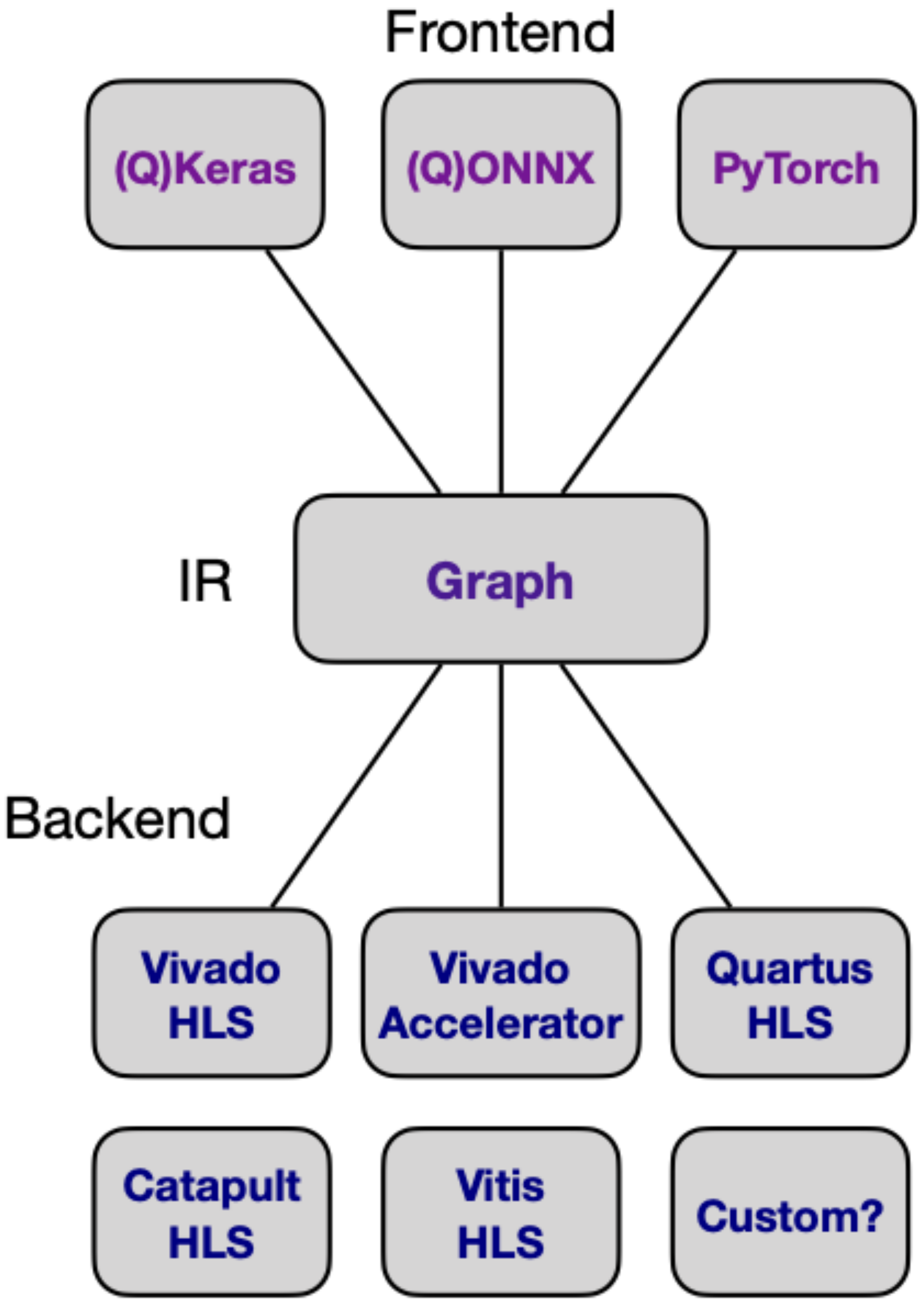




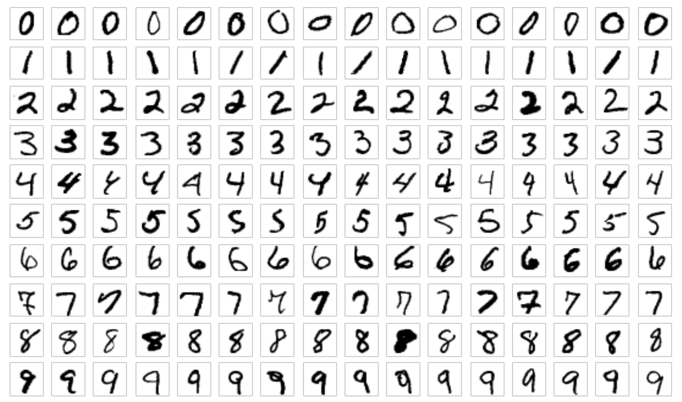
hls4ml:

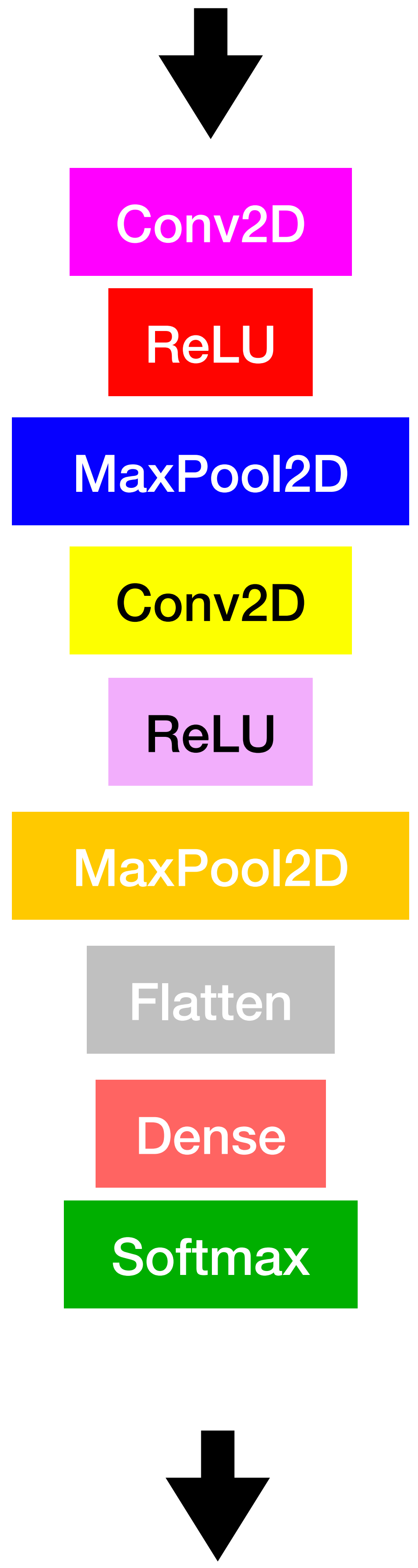
Very broad frontend (ML library) and backend (HLS compiler) to cover most use cases

Should be a library all LHC experiments can use!



hls4ml

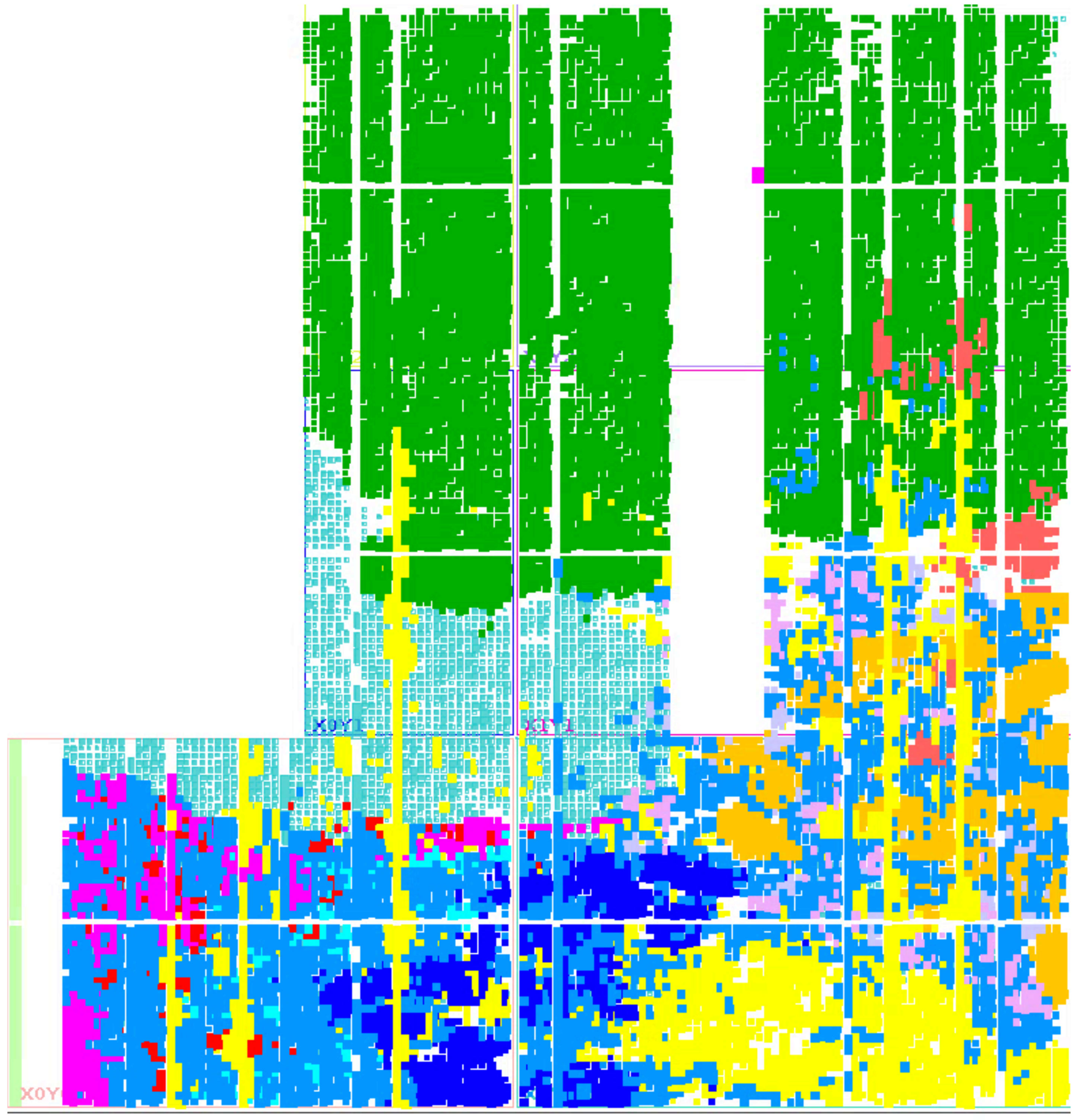




Prediction

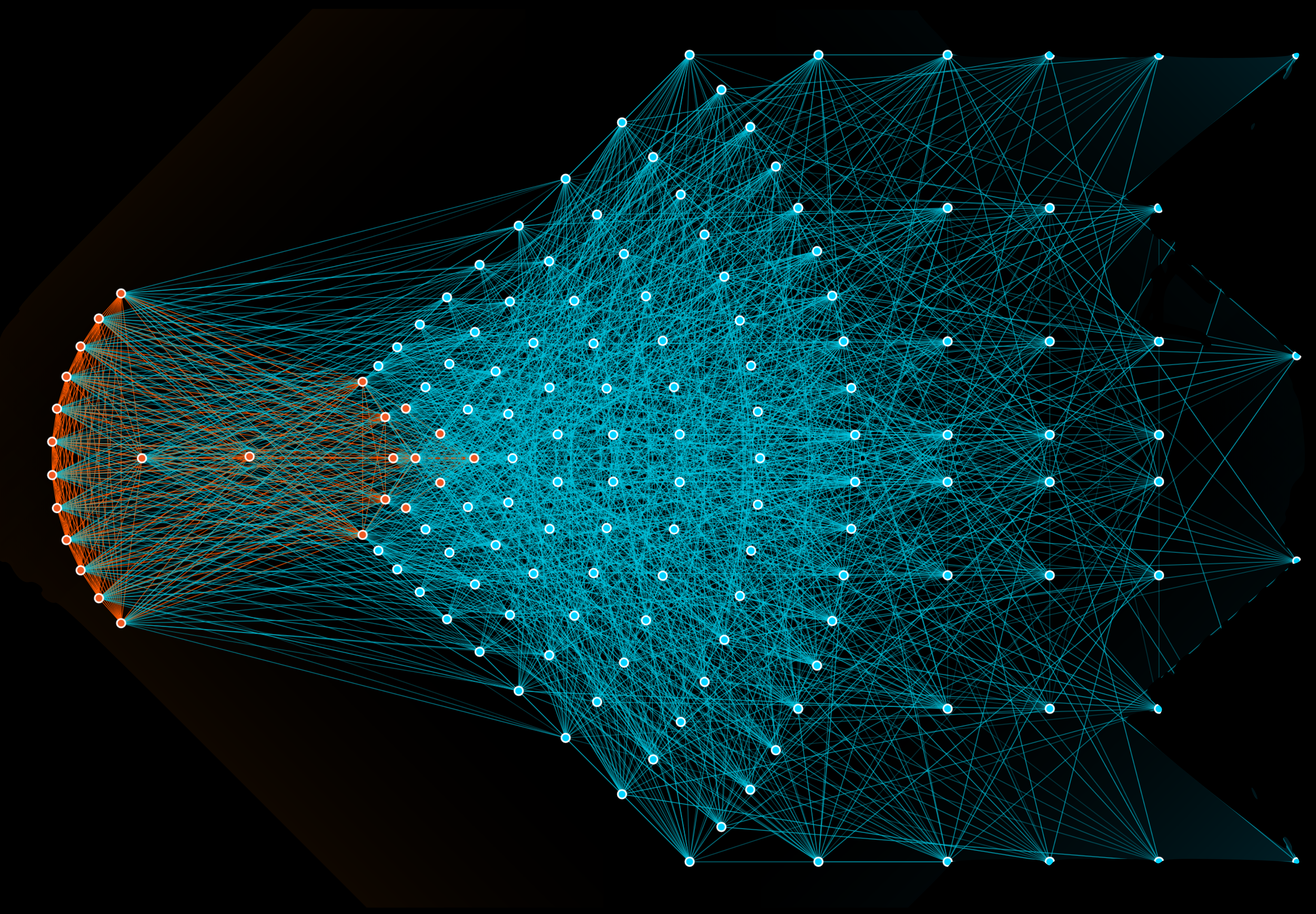
```
from hls4ml import …
import tensorflow as tf

# train or load a model
model = … # e.g. tf.keras.models.load_model(…)

# make a config template
cfg = config_from_keras_model(model,
granularity='name')

# tune the config
cfg['LayerName']['layer2']['ReuseFactor'] = 4

# do the conversion
hmodel = convert_from_keras_model(model, cfg)

# write and compile the HLS
hmodel.compile()

# run bit accurate emulation
y_tf = model.predict(x)
y_hls = hmodel.predict(x)

# do some validation
np.testing.assert_allclose(y_tf, y_hls)

# run HLS synthesis
hmodel.build()
```

Fully on-chip

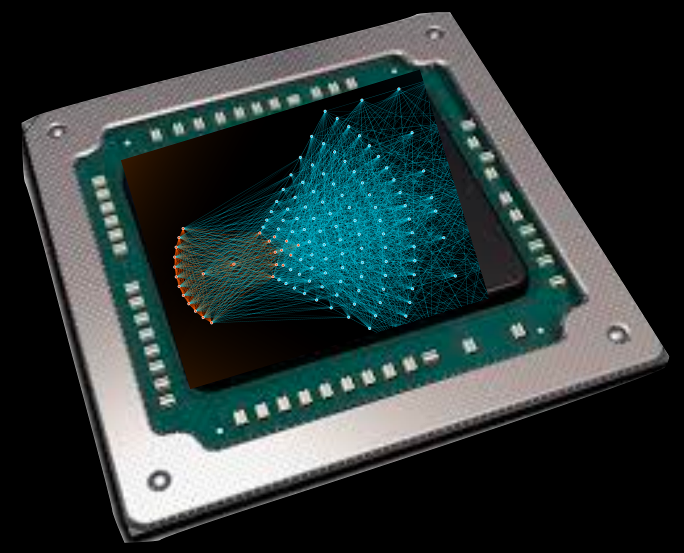


pynq-z2 floorplan

(from Sioni S Summers)



Ideally



Reality

Edge inference

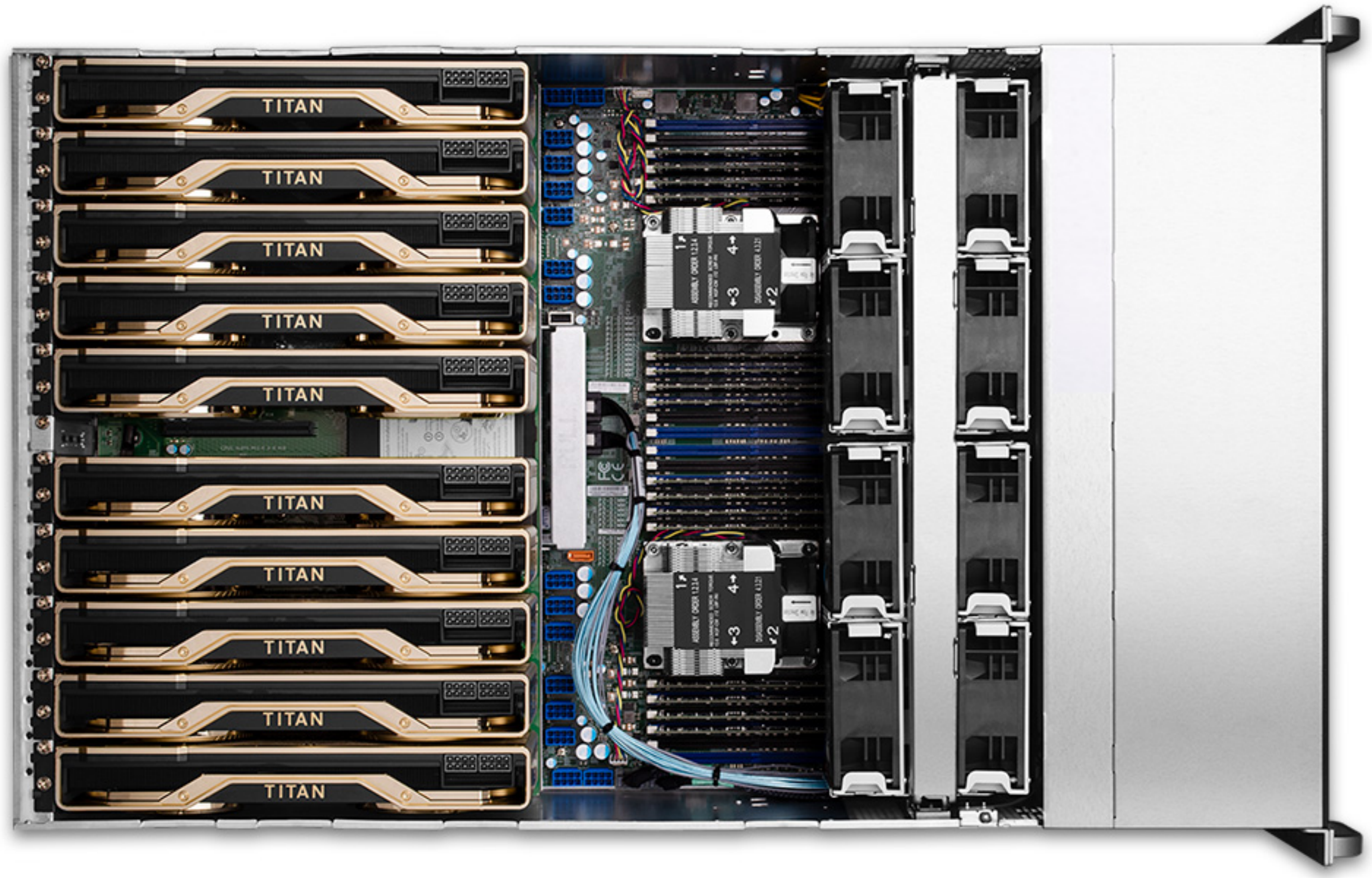
Before deploying any DNN on the edge, must make it efficient!

During training

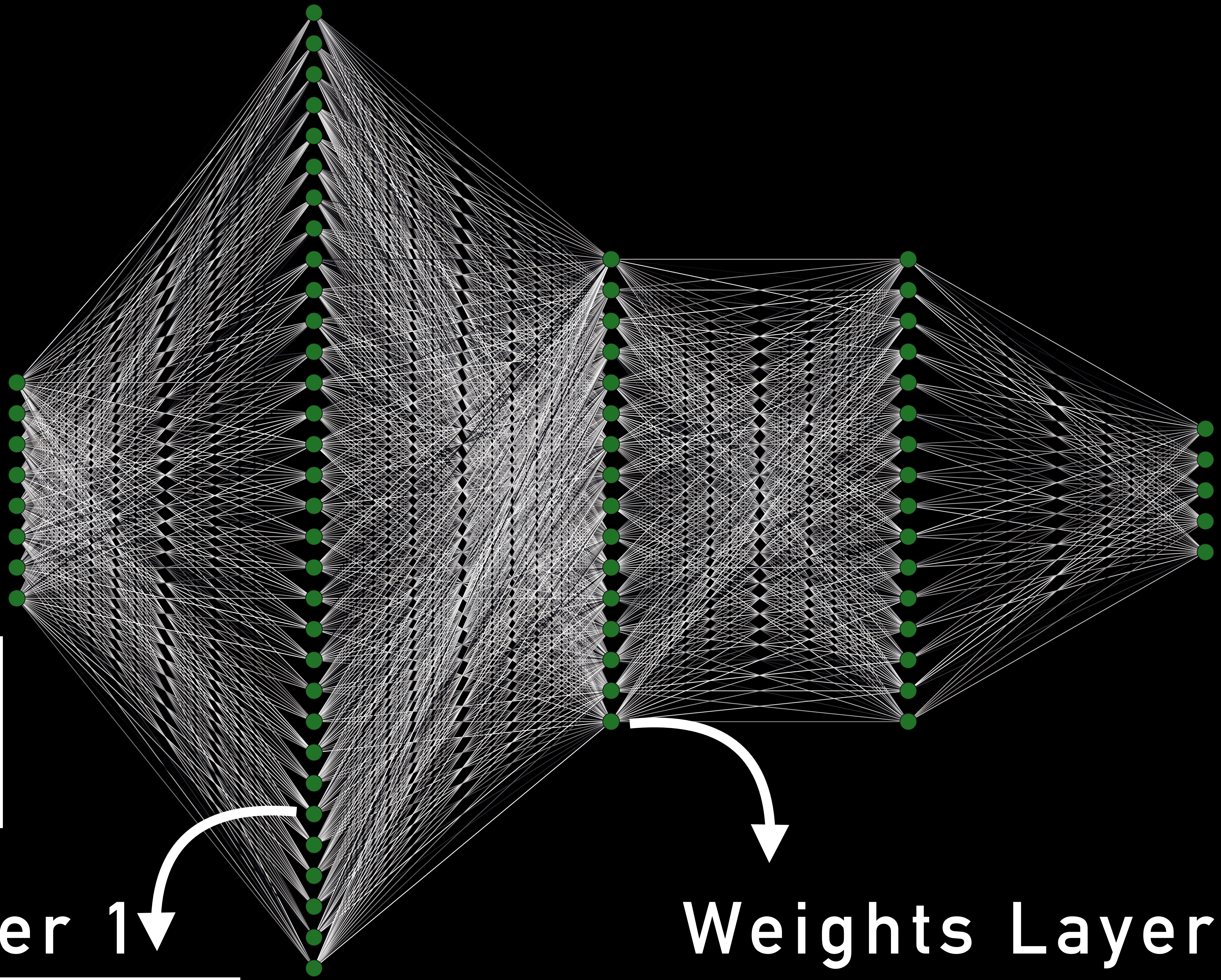
- **Quantization:** do you really need 32-bit FP precision?
- **Pruning:** removal insignificant synapses
- (**Knowledge distillation:** train large network, deploy small)

Post-training

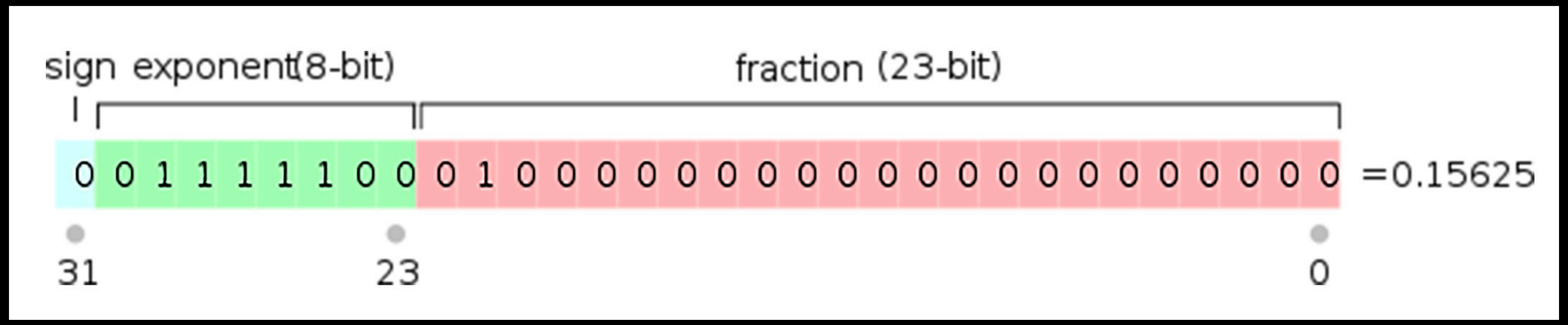
- **Parallelise**:
  all computation that can be done in parallel, do in parallel!



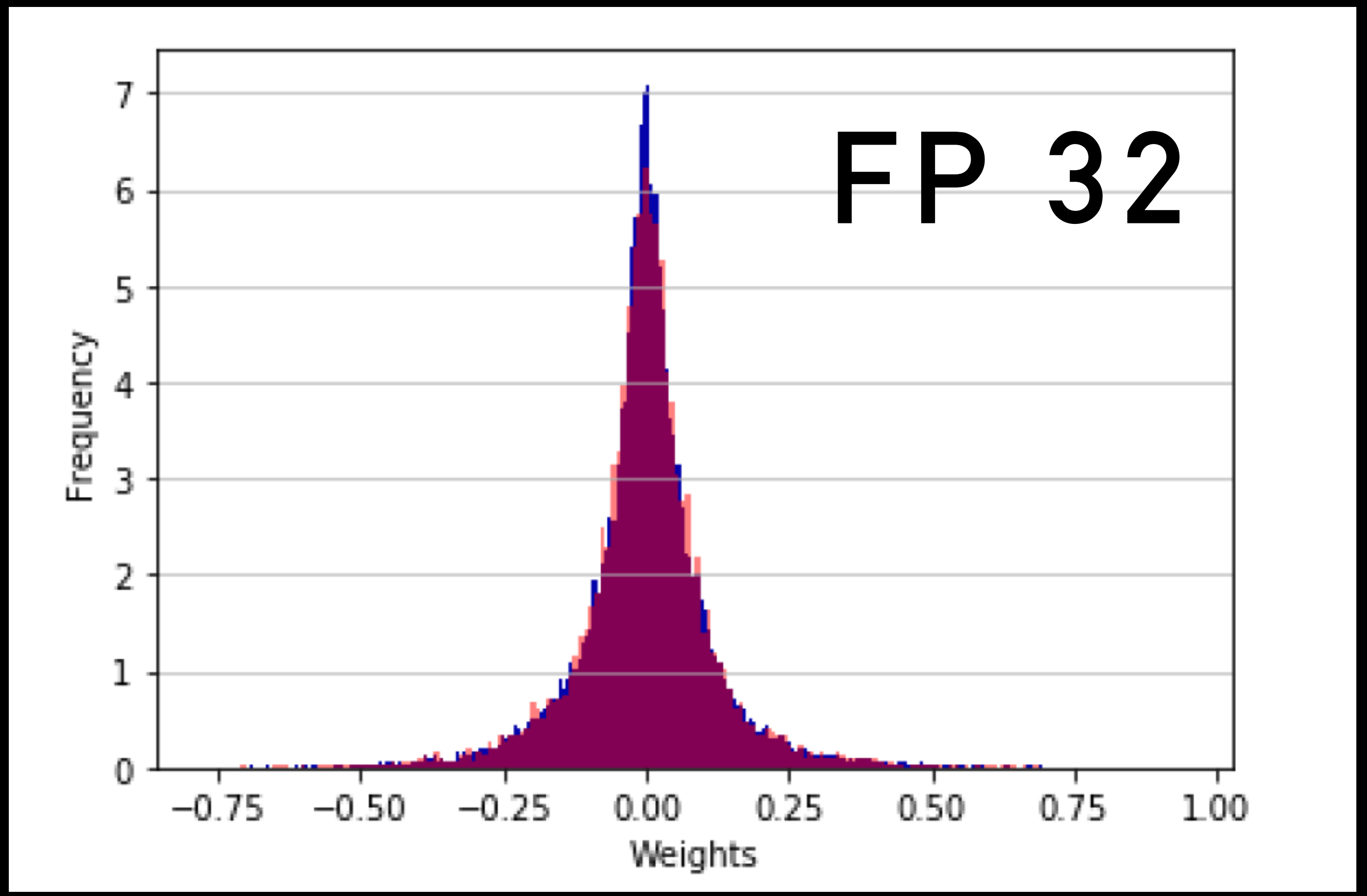
Quantization



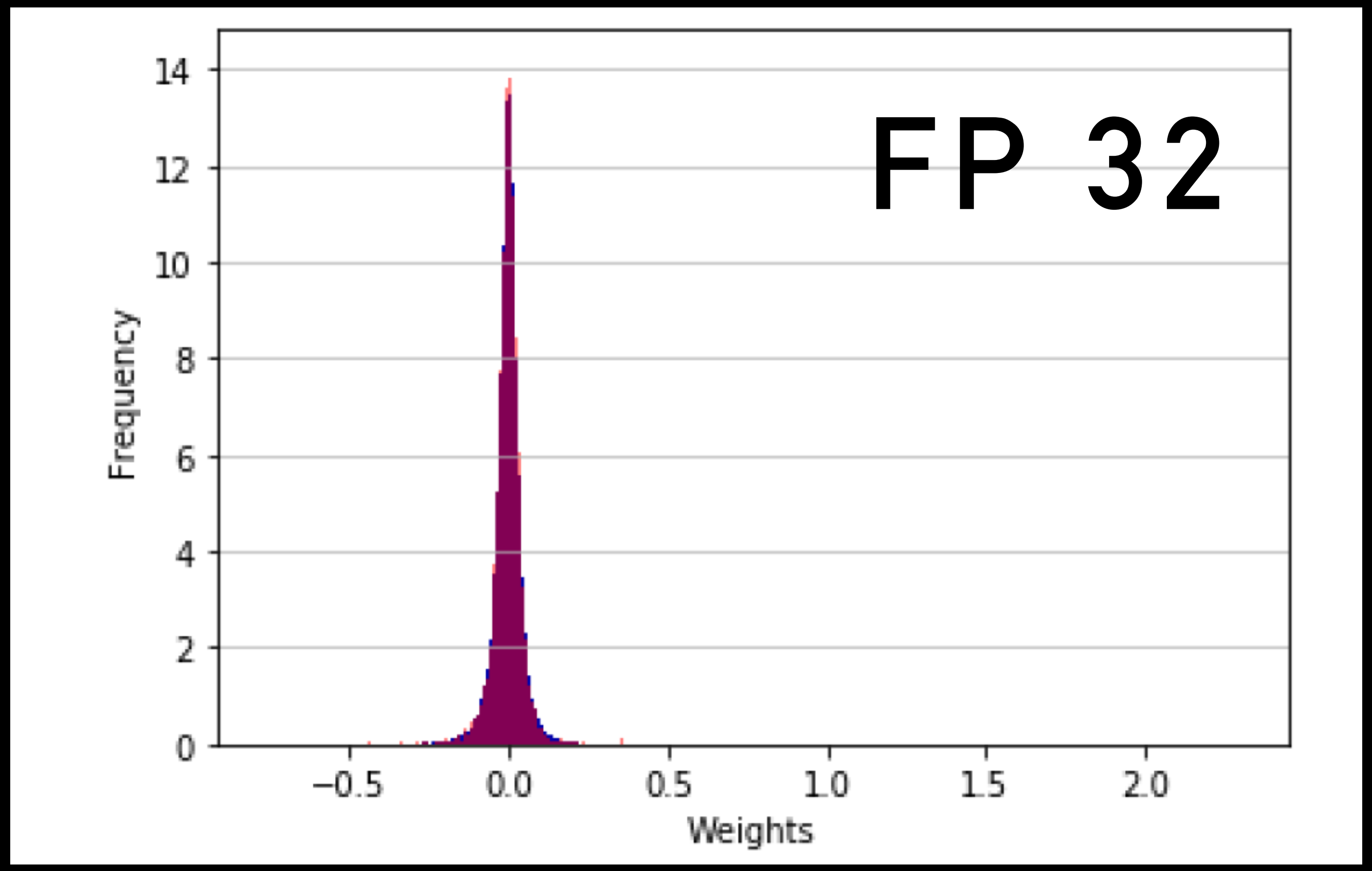
Floating point 32



Weights Layer 1

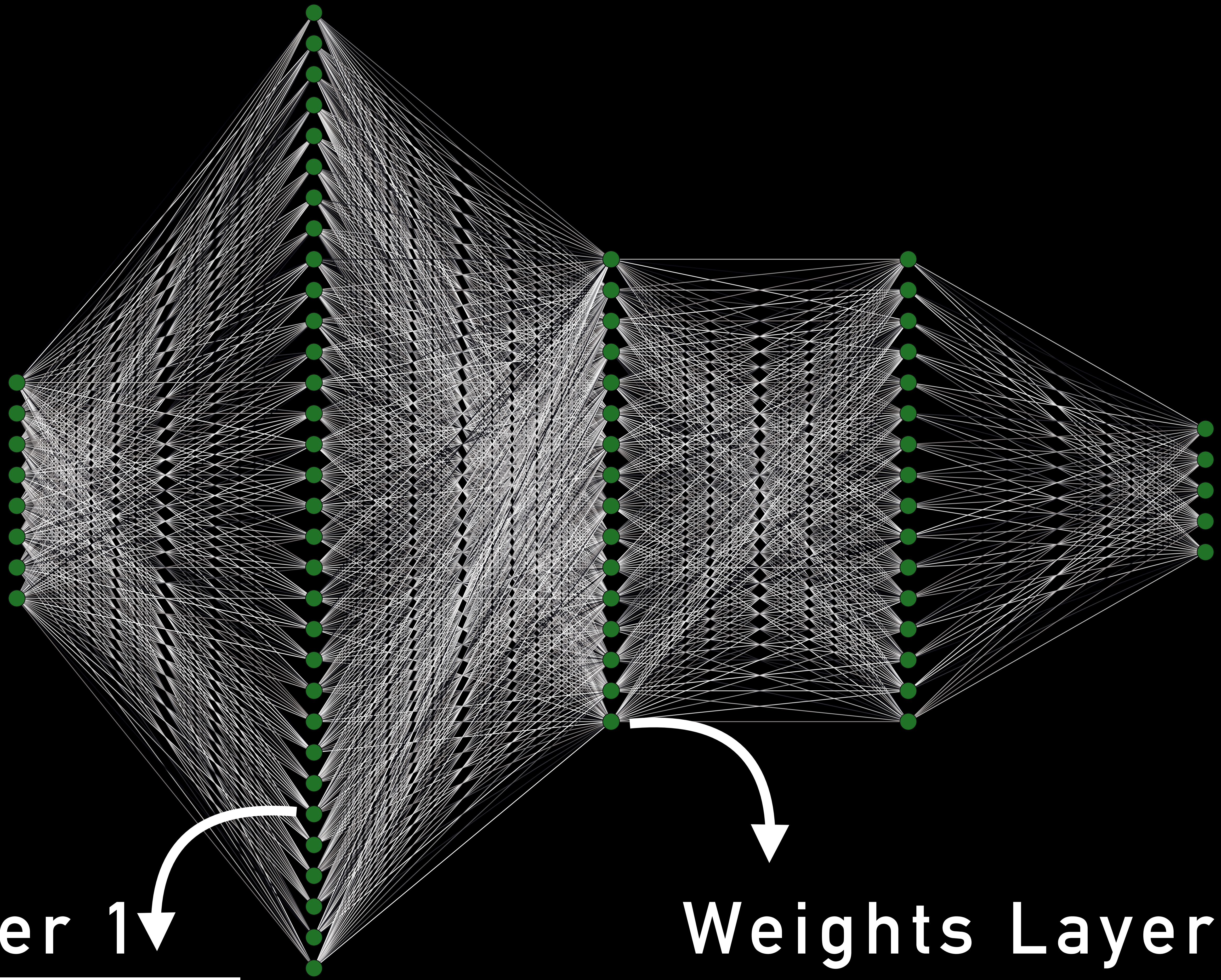


Weights Layer 2

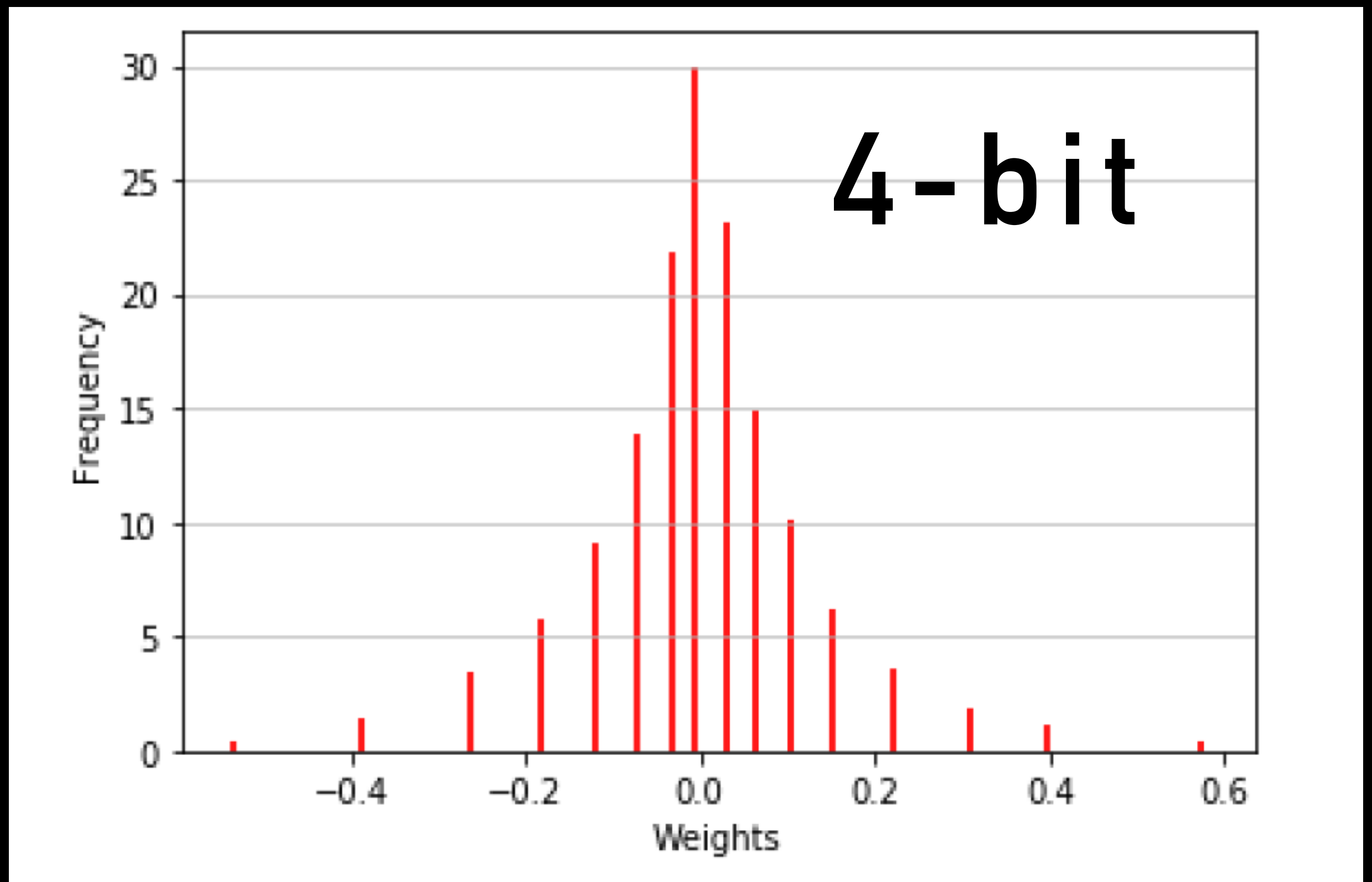


Quantization

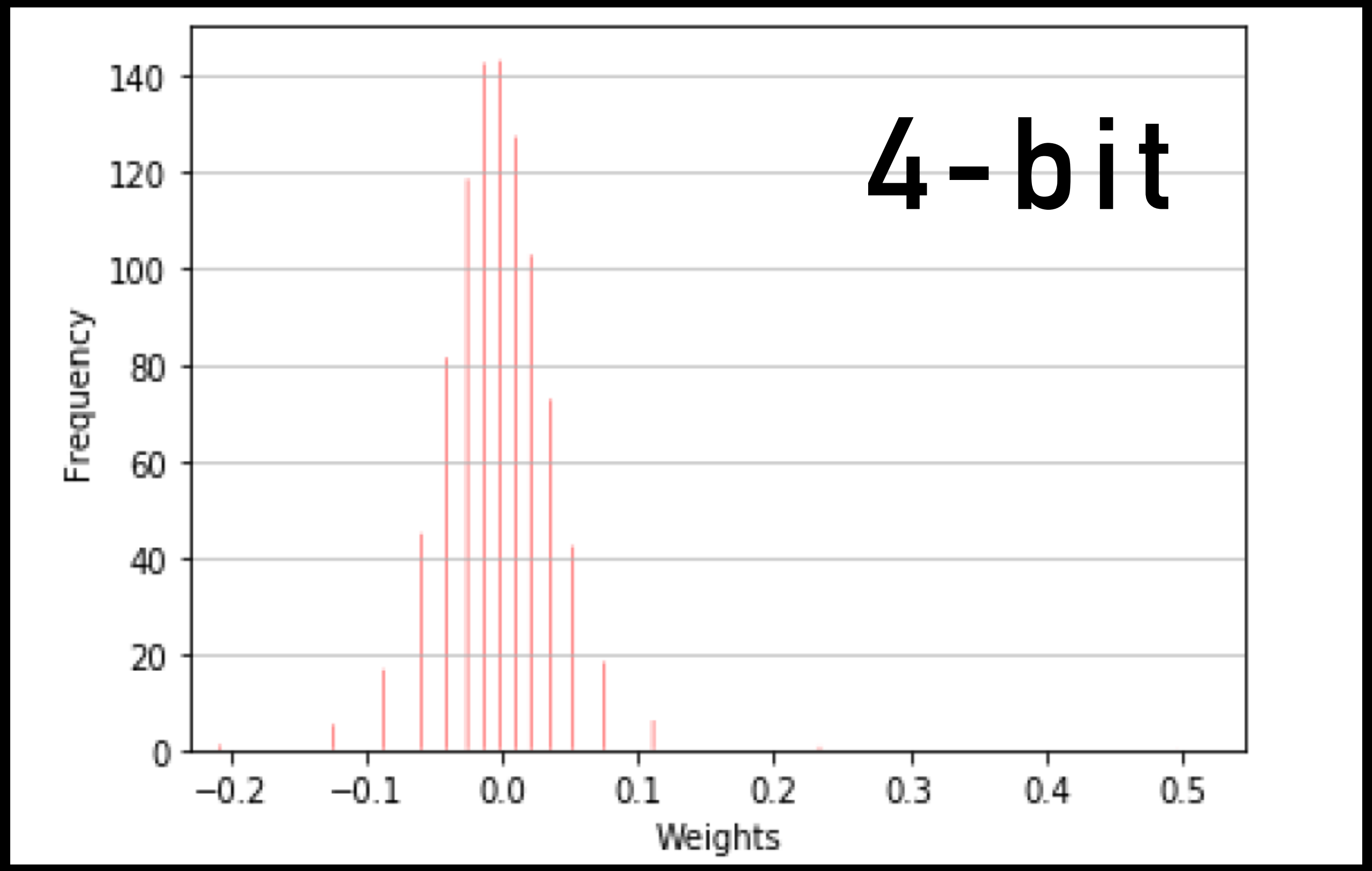
Fixed point



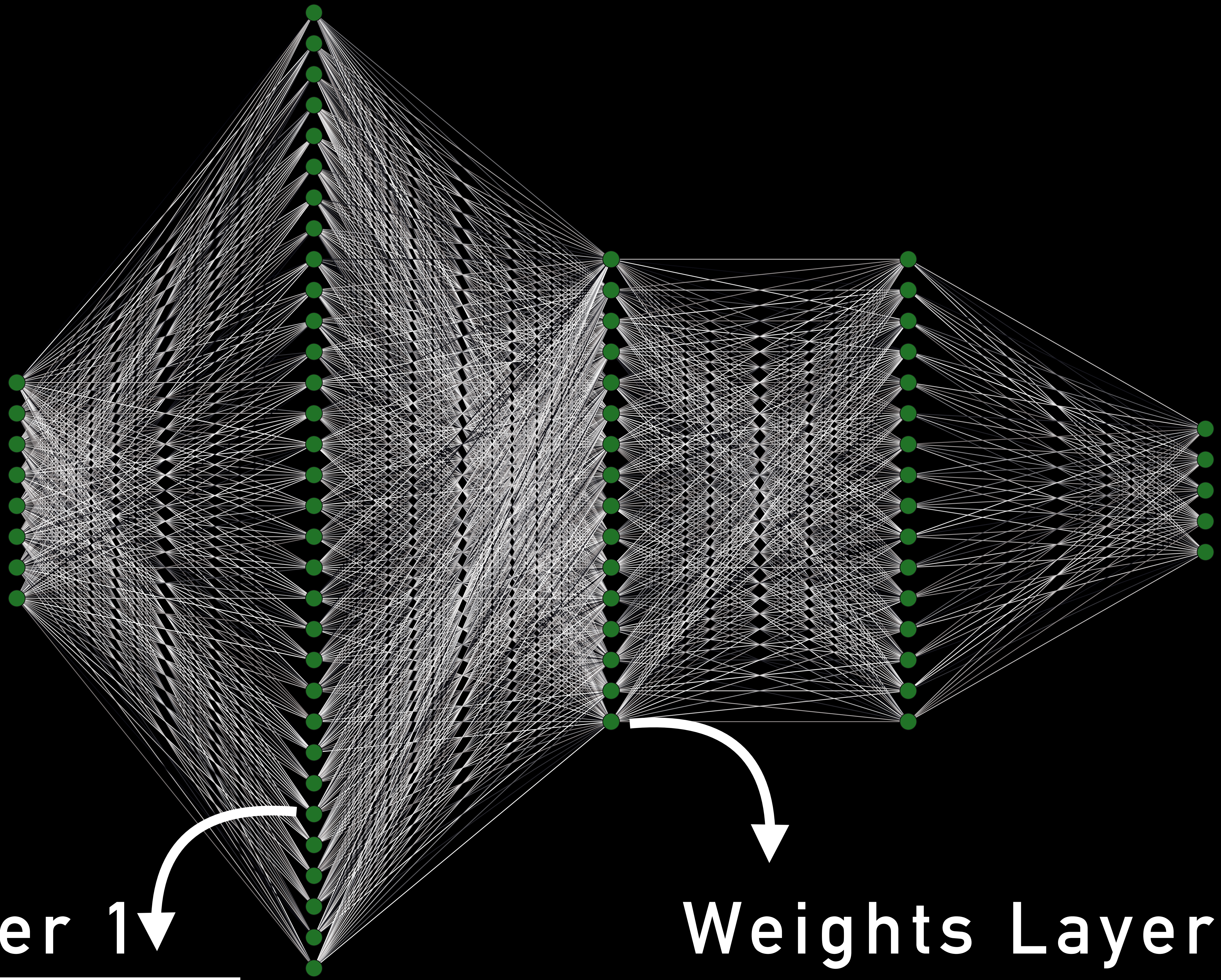
Weights Layer 1



Weights Layer 2



Quantization



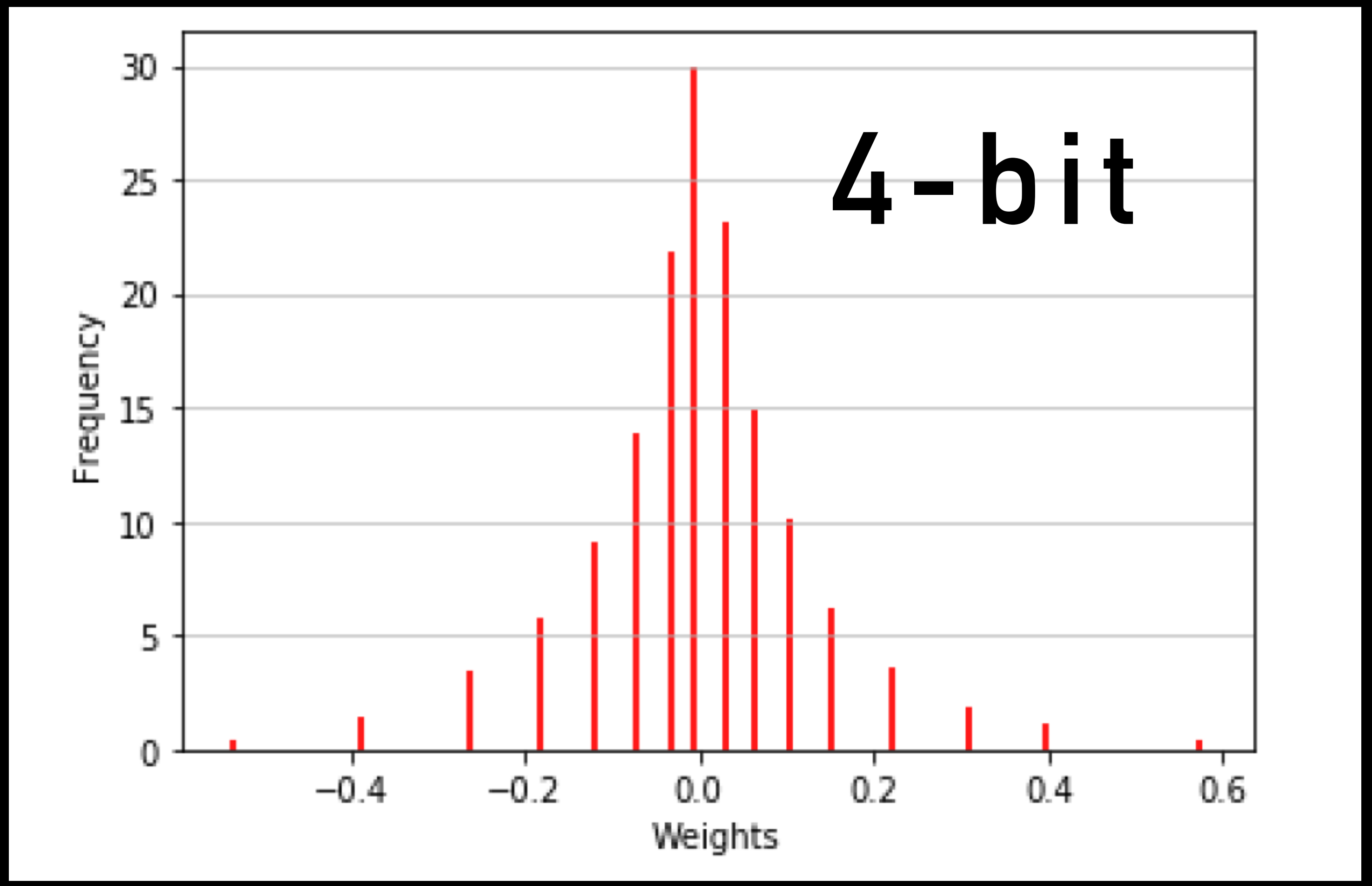
Fixed point

0101.1011101010

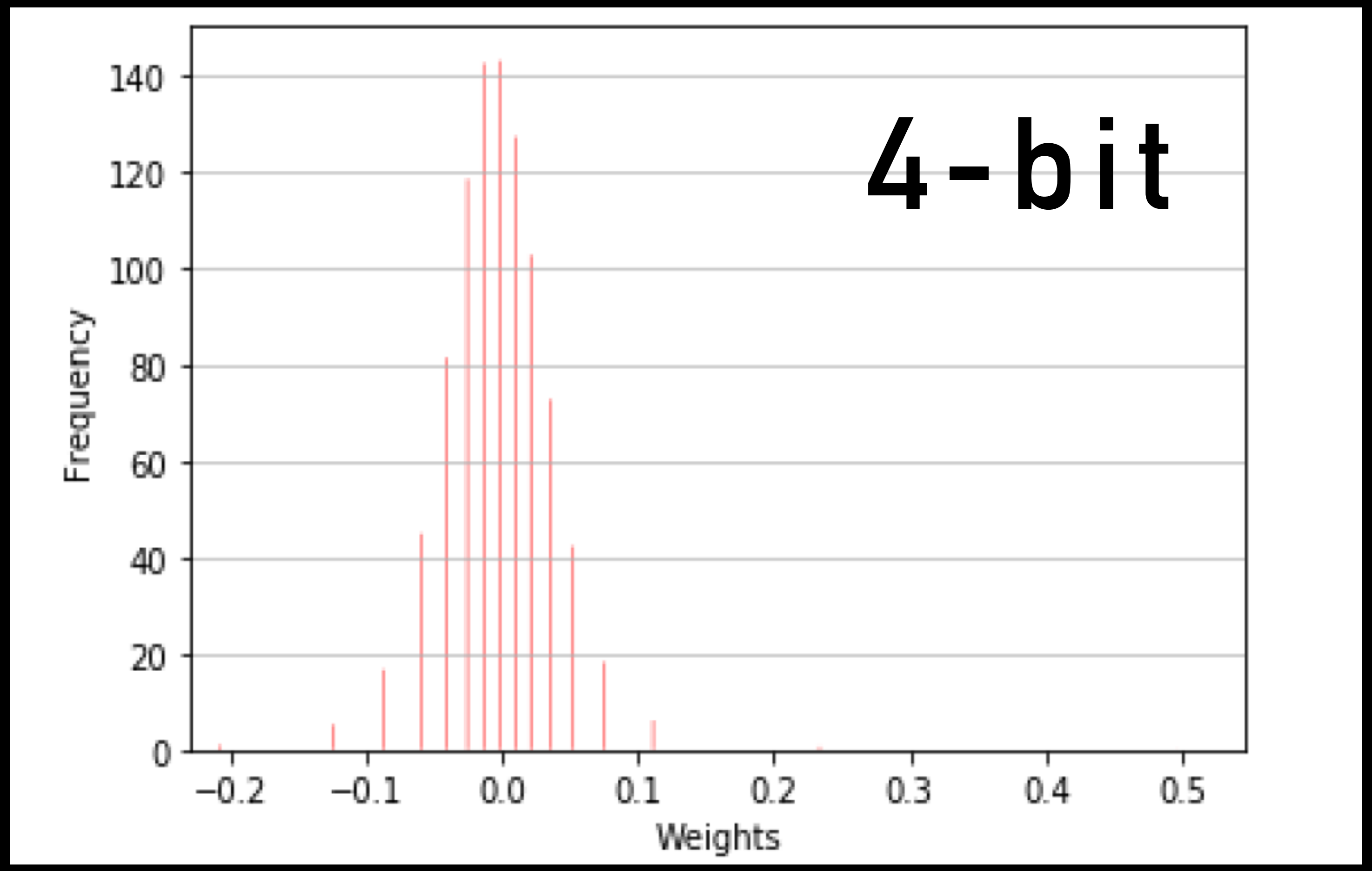
integer | fractional

width

Weights Layer 1

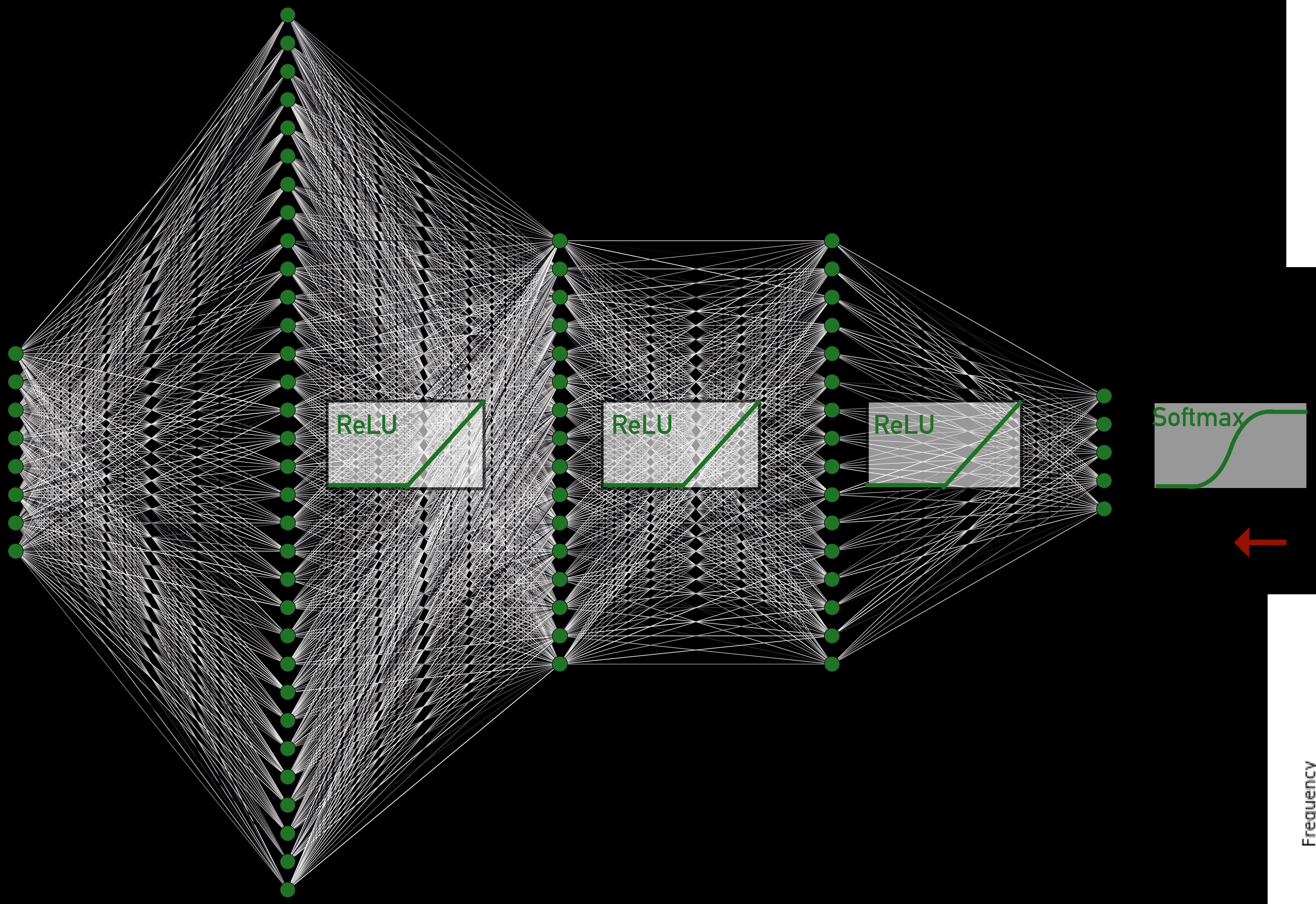


Weights Layer 2

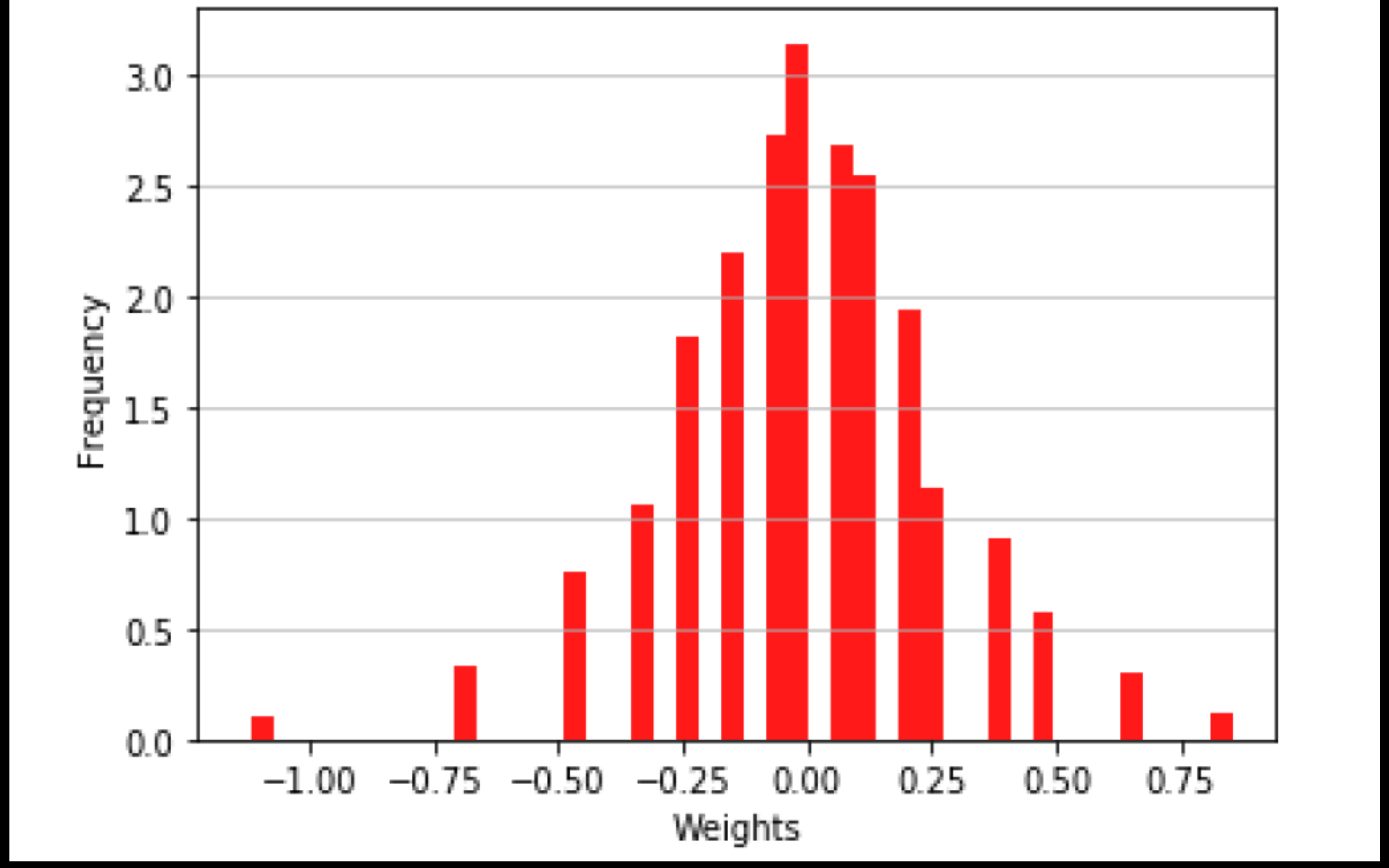


hls4ml + Google

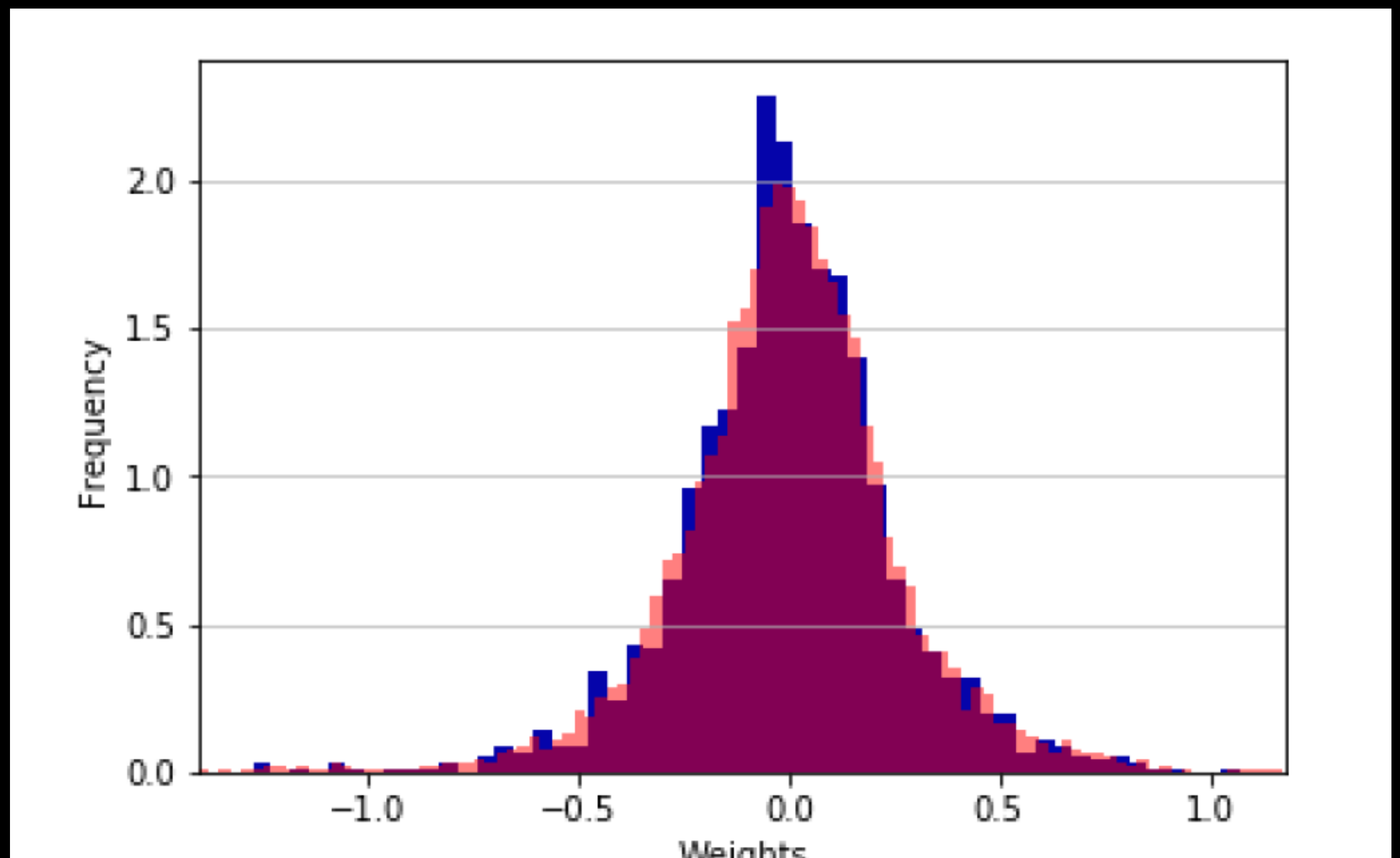
Quantization-aware training



Forward pass →



← Back propagation



Nature Machine Intelligence 3 (2021)

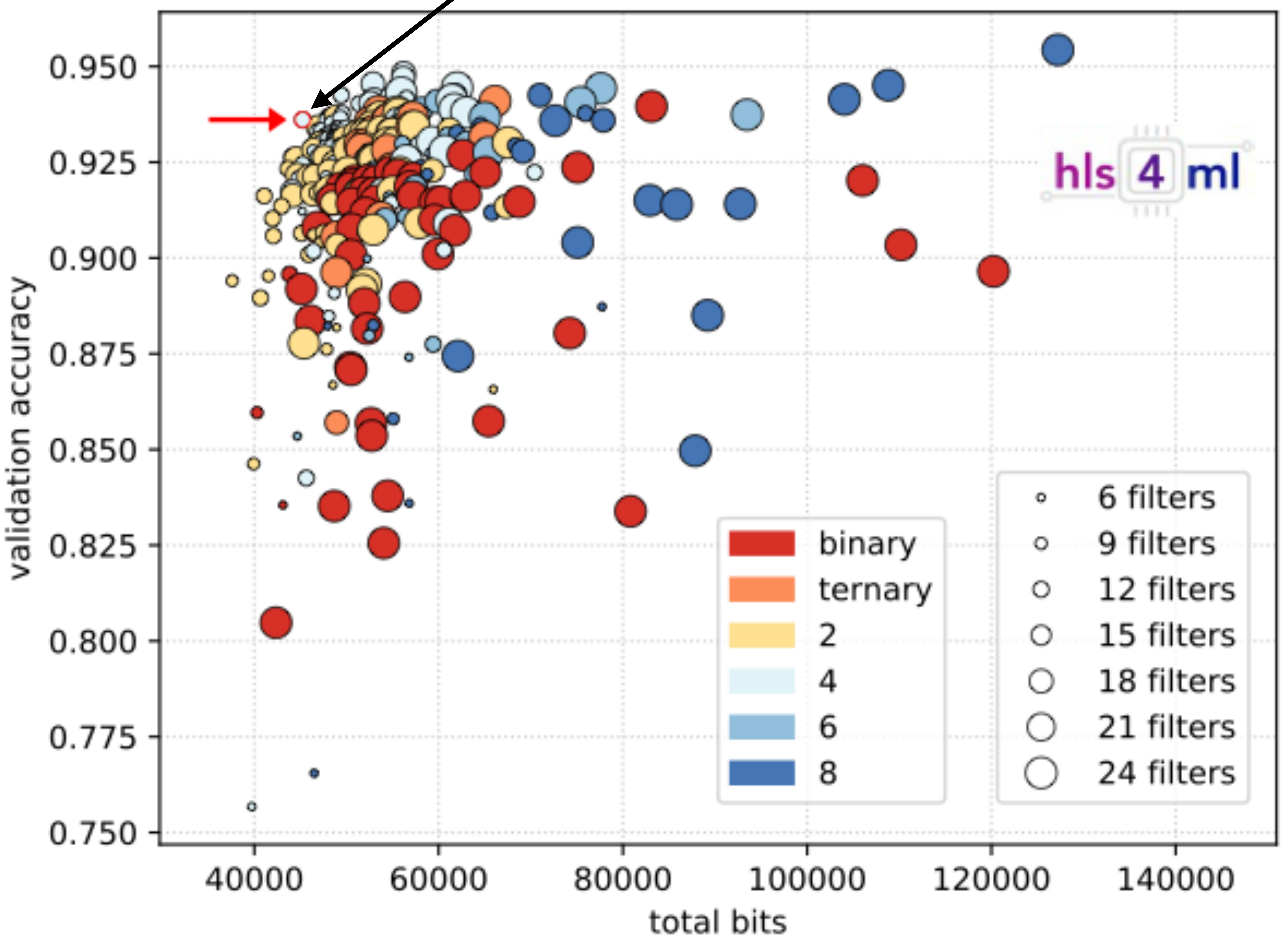
DOI 10.1088/2632-2153/ac0ea1

Automatic quantization

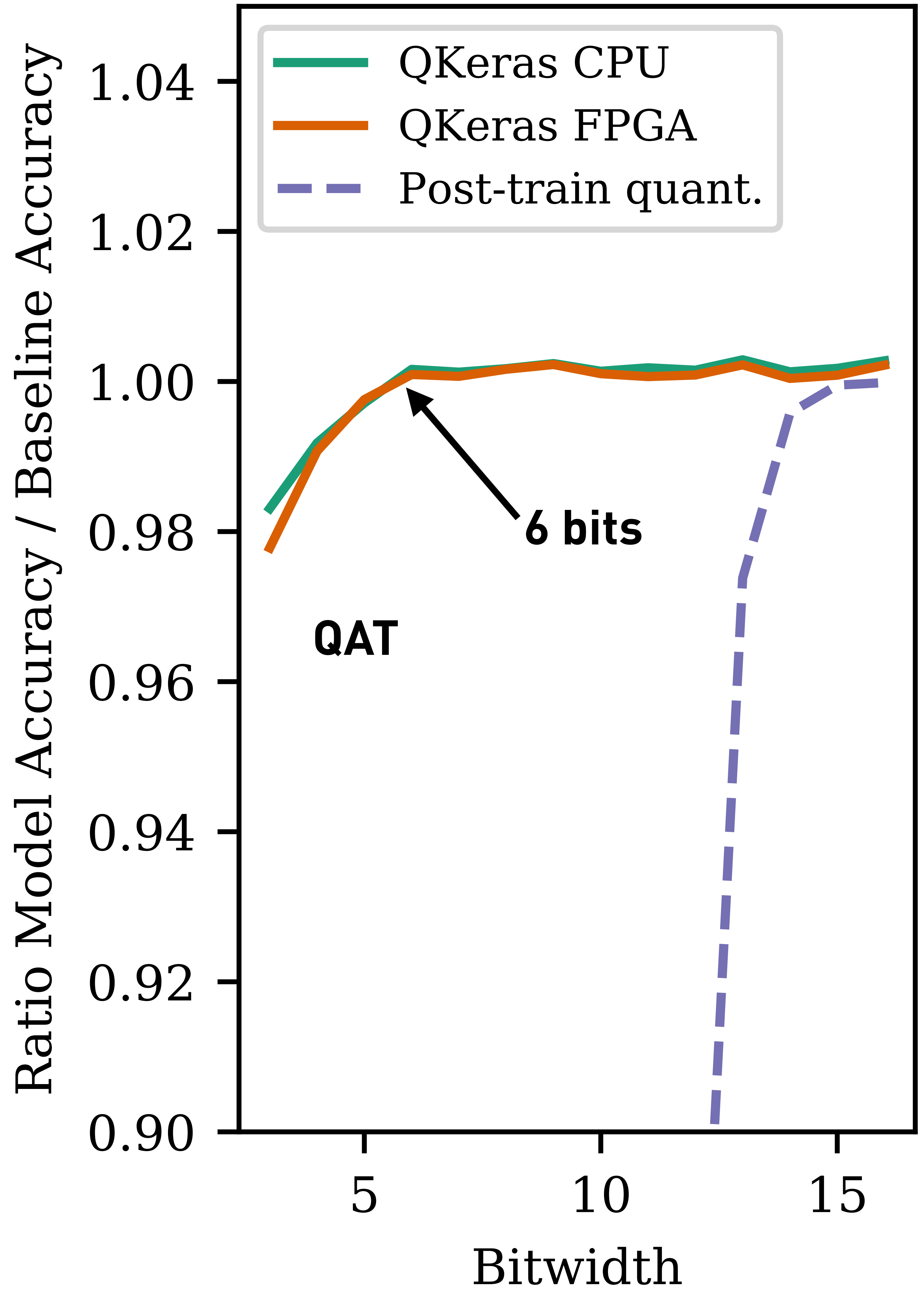
As optimization progresses, best model accuracy/size trade-off is found!

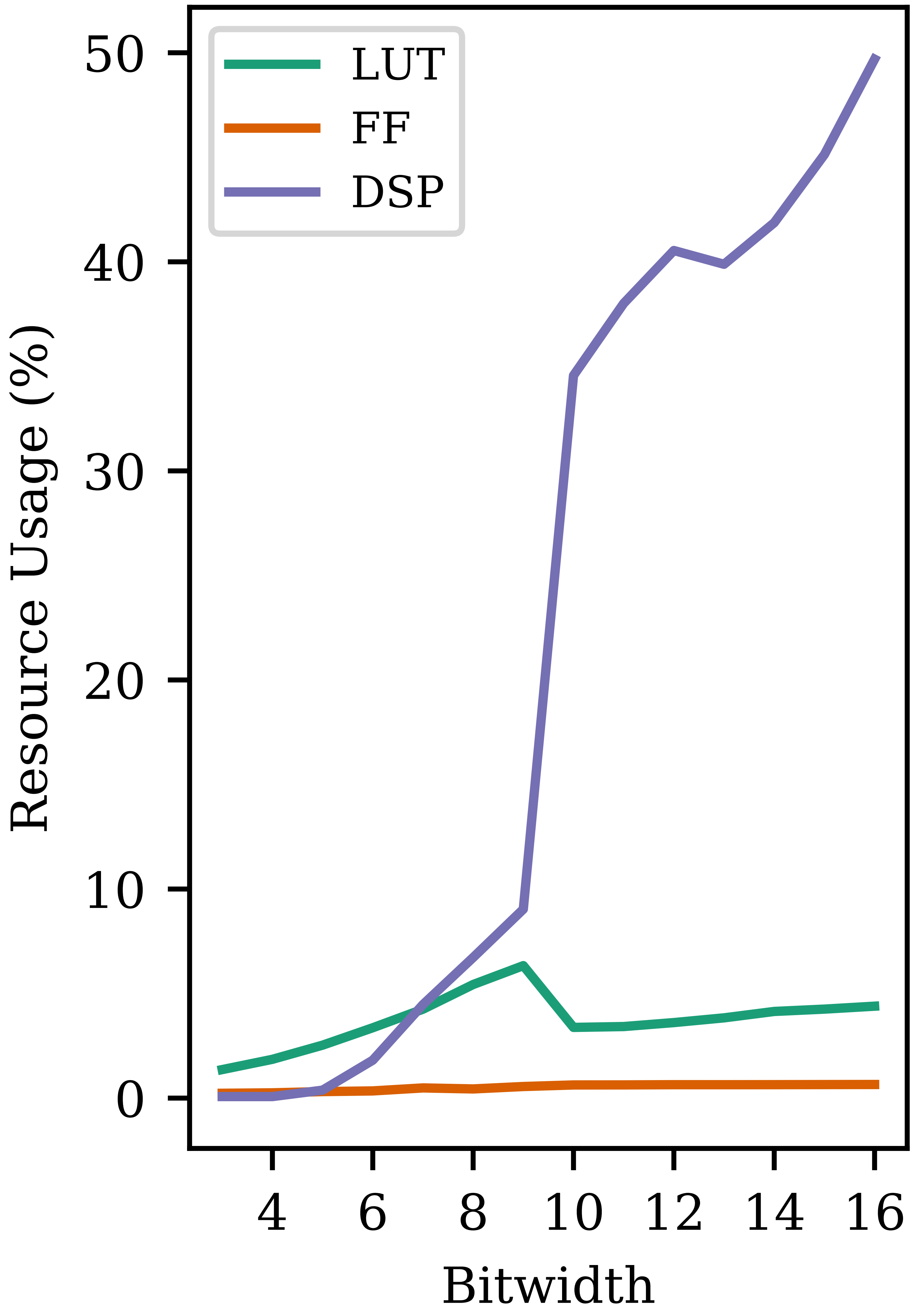
AutoQ Bayesian optimization

- Simultaneously scan quantizers and architecture (often less/more filters/neurons needed when quantizing)



FPGA performance





OpenReview.net

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

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

PDF

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 ($\sim$10K samples) even without accounting

Cornell University

the Simo

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)

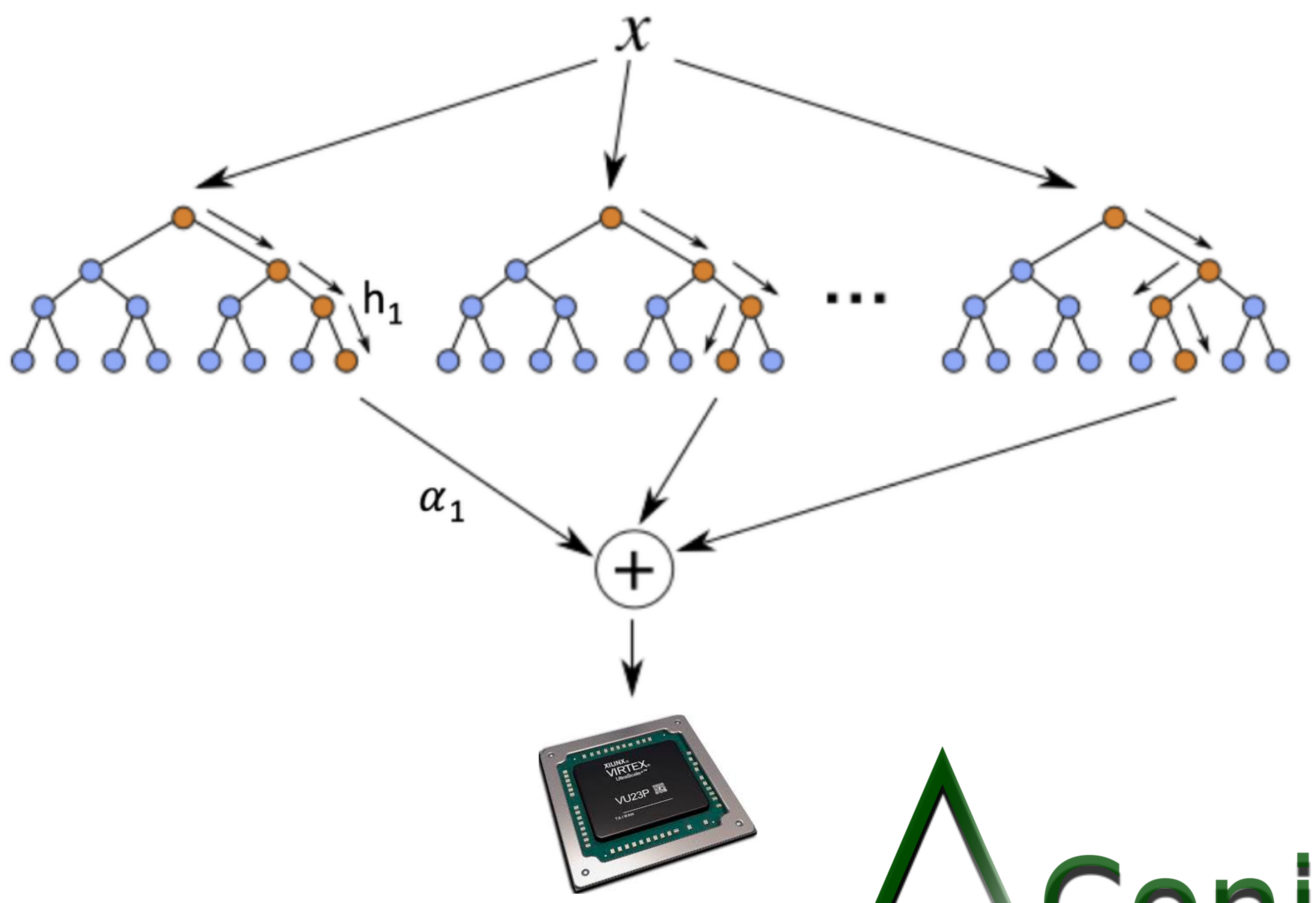
[v2] Mon, 17 Oct 2022 15:18:14 UTC (224 KB)

[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%

Google AI
TensorFlow
Decision Forests



CERN

Conifer

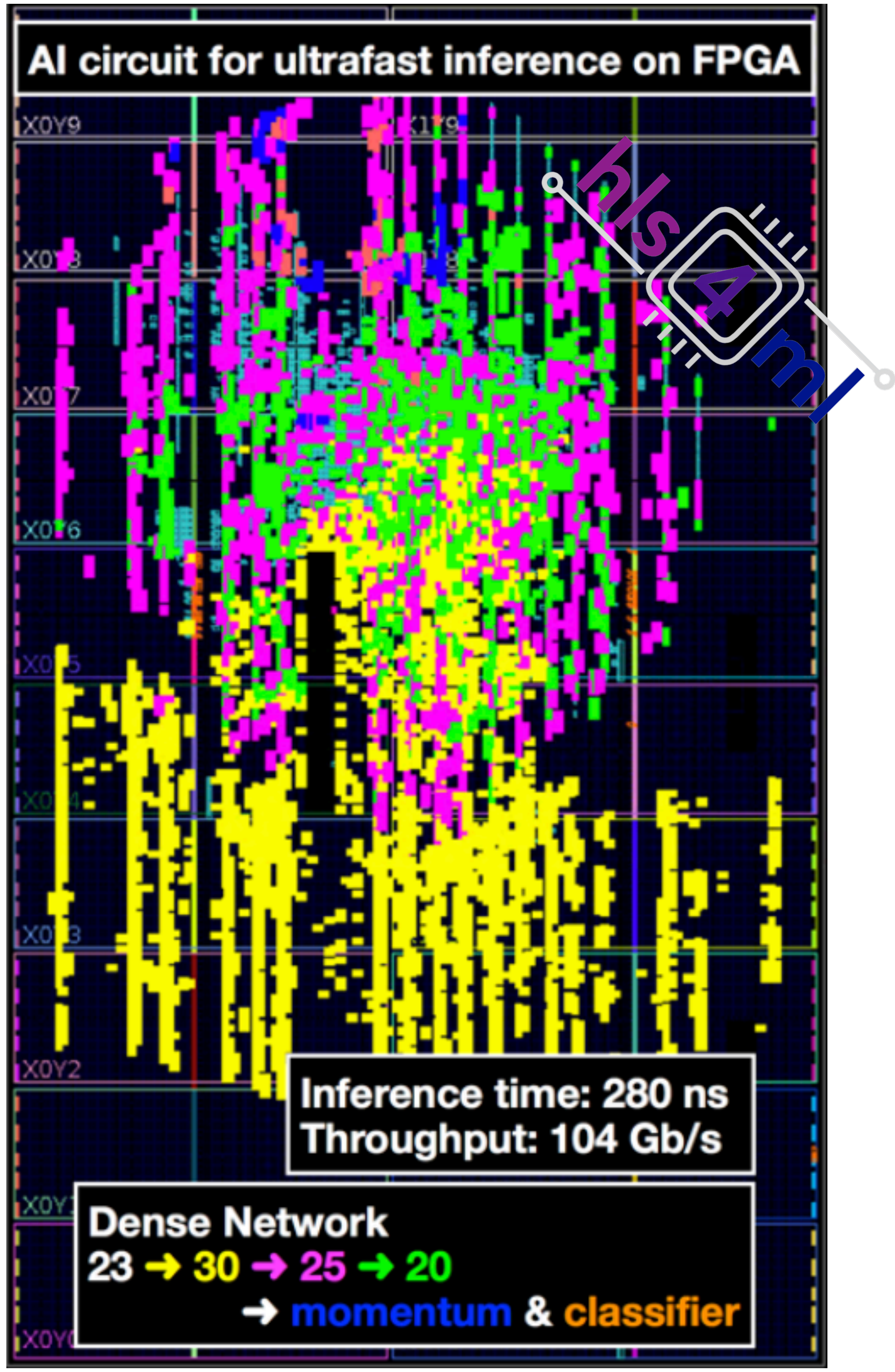
Making the most out of Run 3

Deep Learning at L-1, S. Jindariani and N. Tran

Run 3: Explore ML-based triggering algorithms
to improve physics quality of our last LHC data

- First hl4ml models running in CMS for Run 3!
- Better reconstruction of displaced muons:
  DNN for displaced muon p_T assignment and PU discriminant

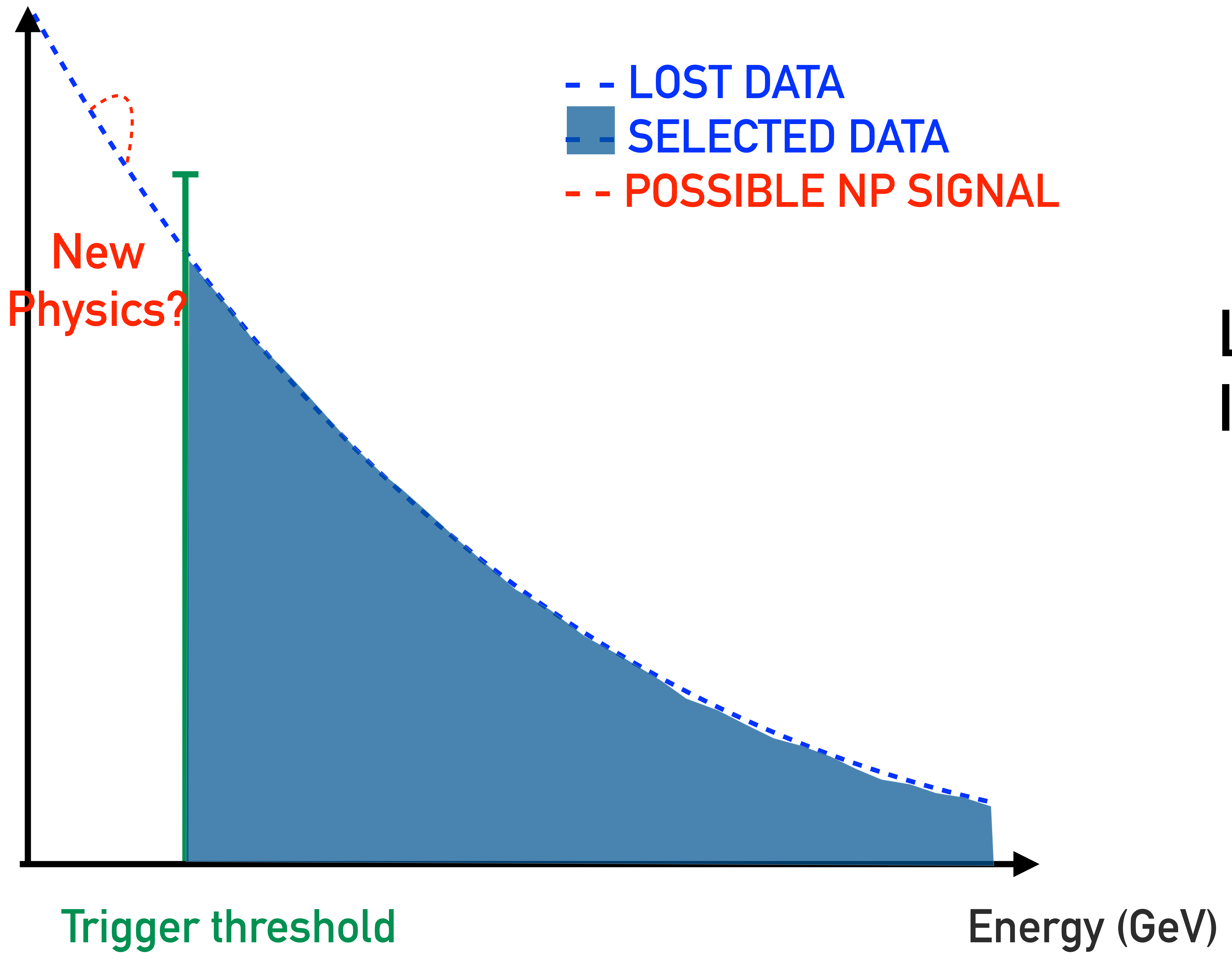
Also a chance to do something new….



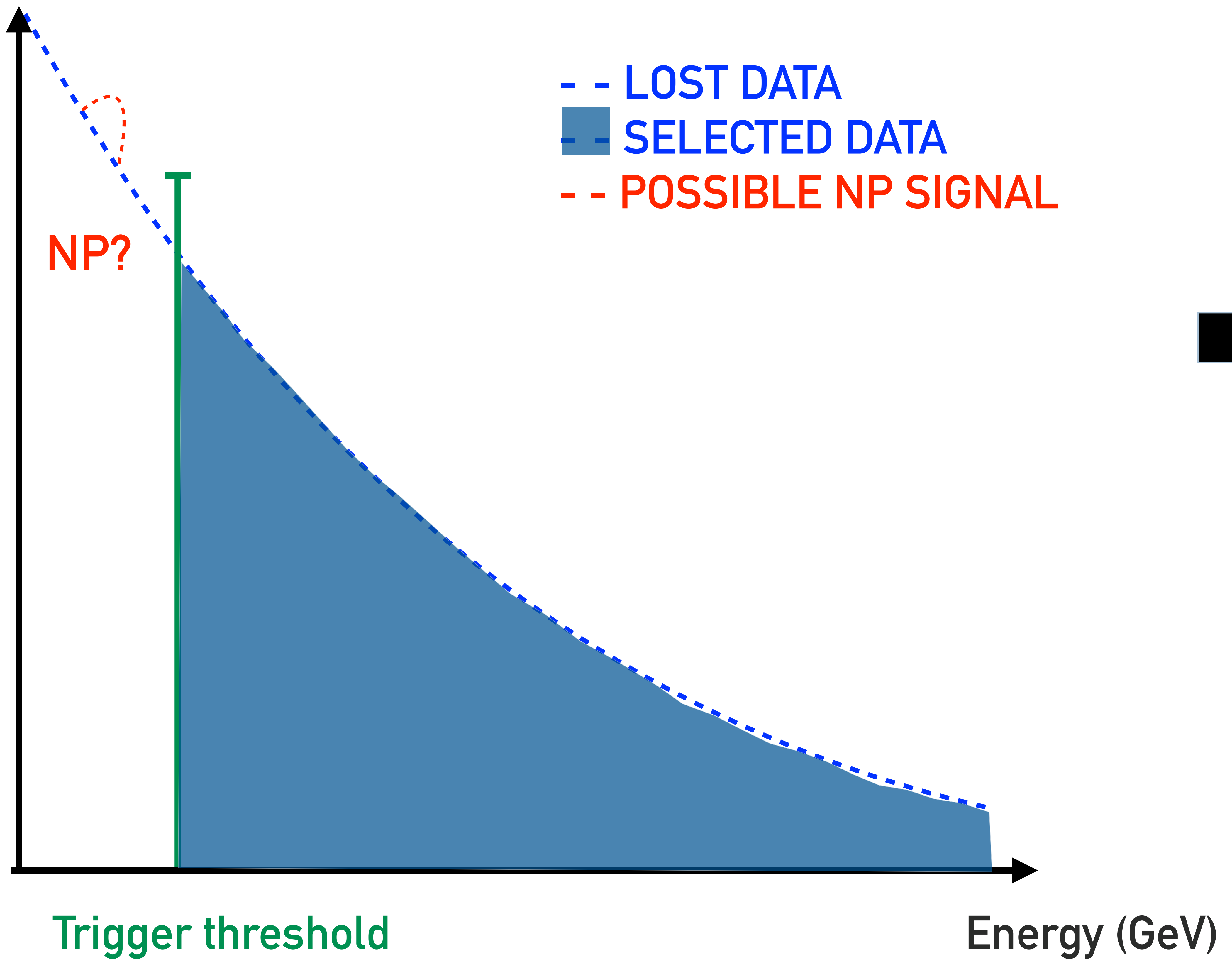
Searches for new particles at LHC

ARE WE LOOKING IN THE WRONG WAY?

Limitations of current trigger

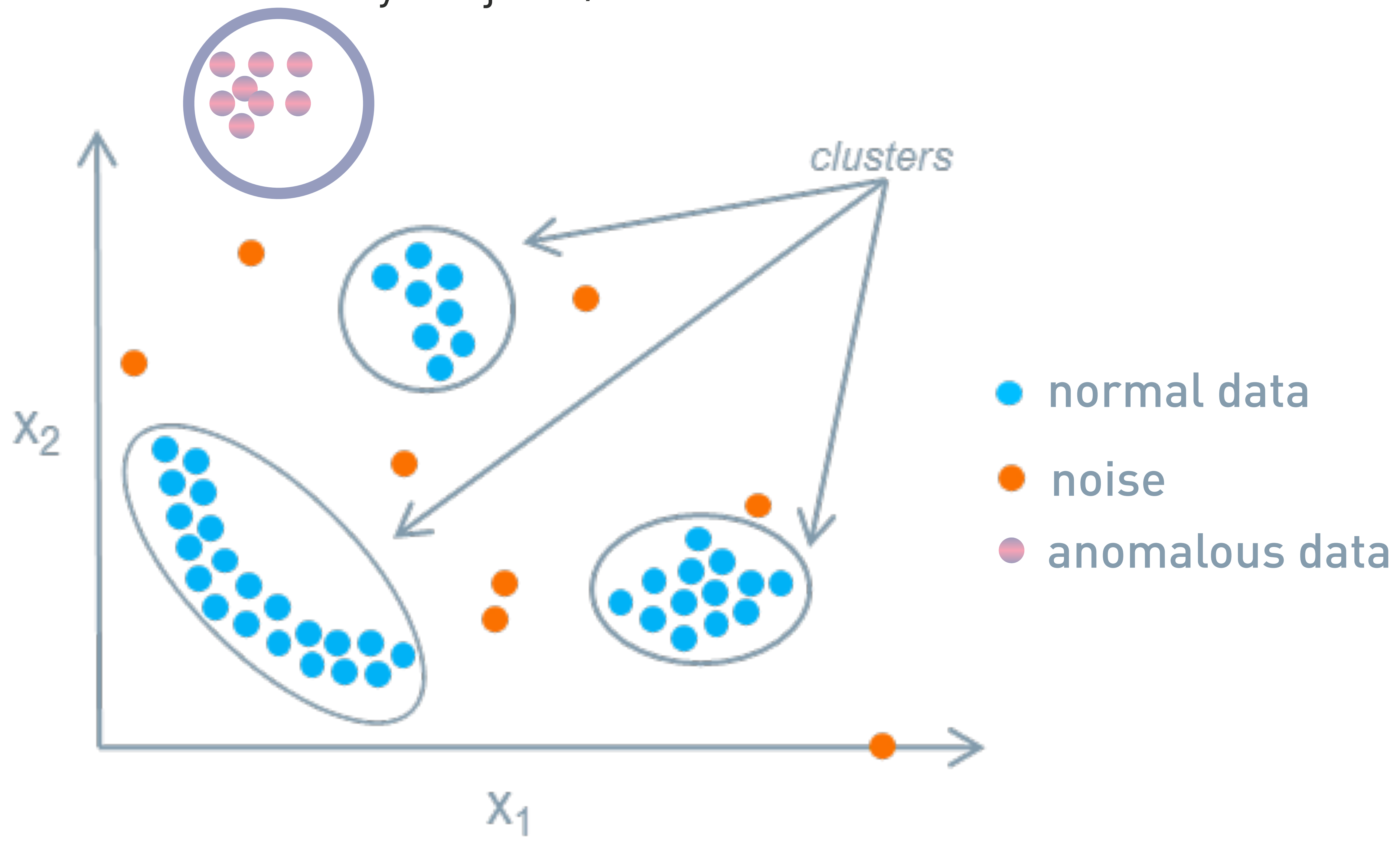


Level-1 rejects >98% of events!
Is there a smarter way to select?



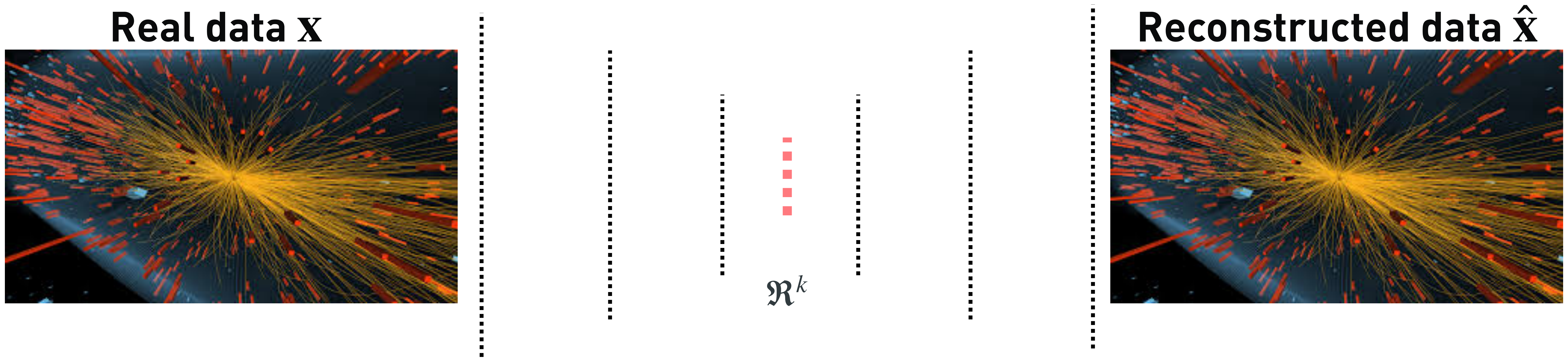
Look at **data** rather than defining signal hypothesis a priori

- Can we "classify" objects/events?

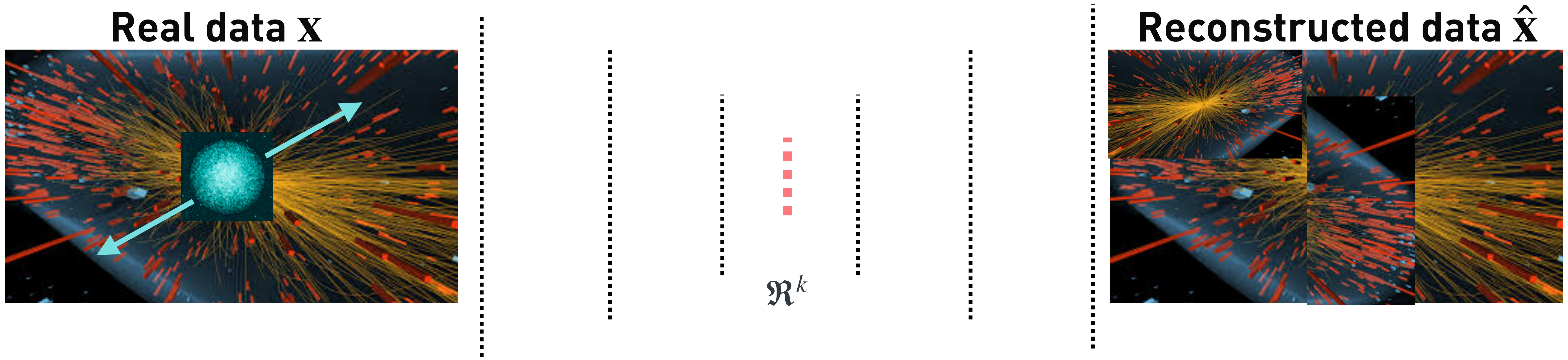


ML for anomaly detection

VAE: Learn from data

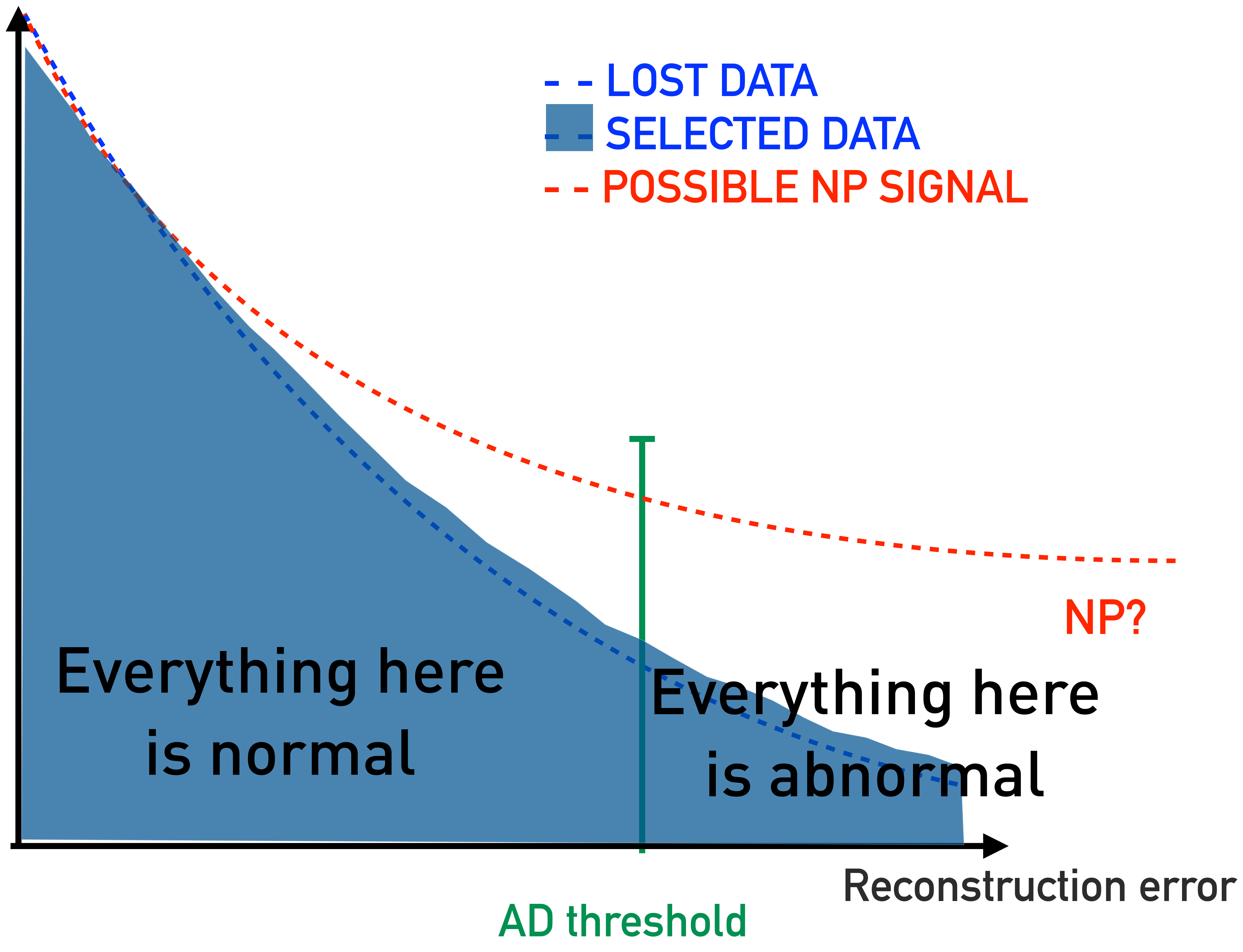


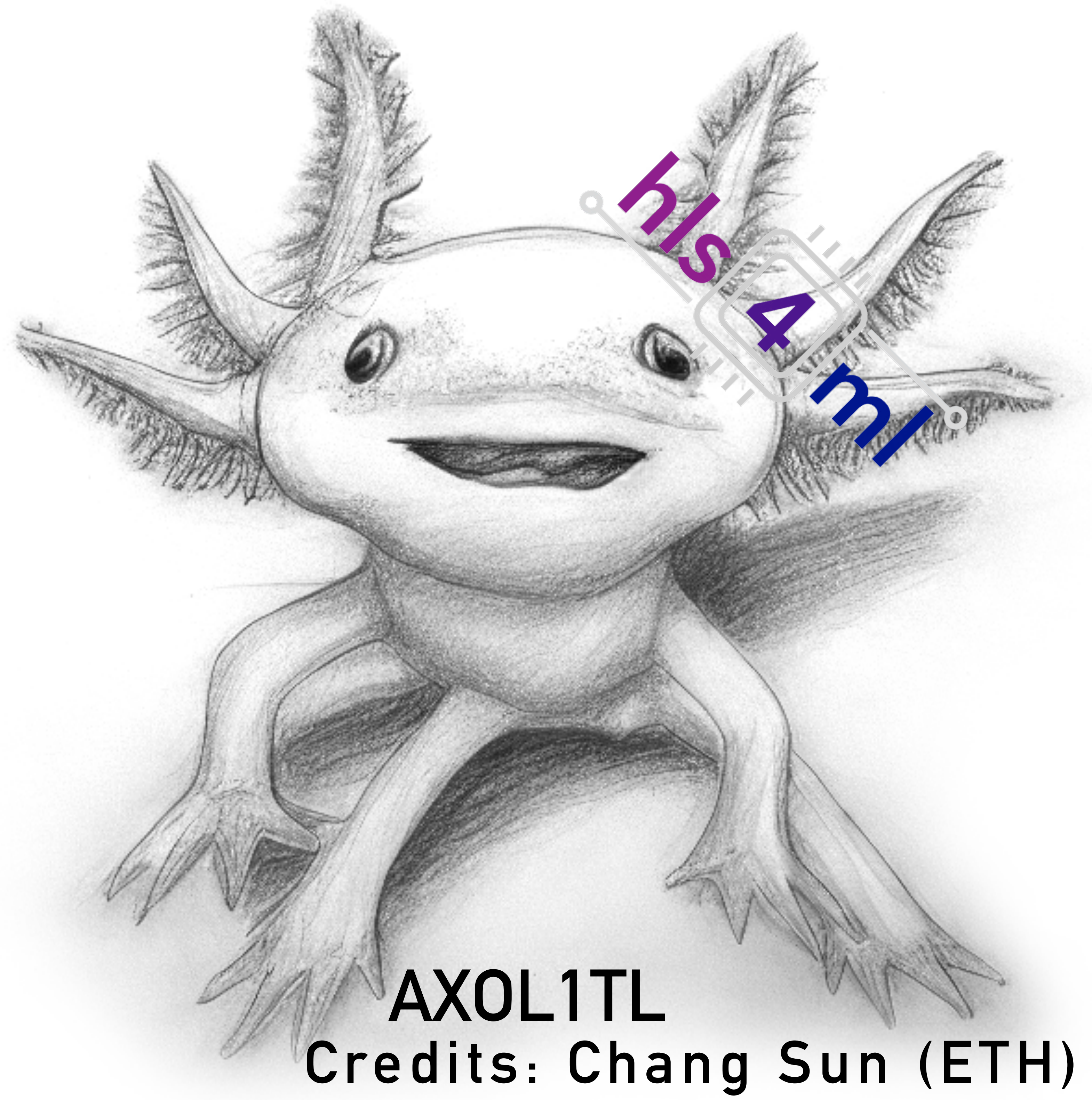
ML for anomaly detection



- Difference $\mathbf{x}$-$\hat{\mathbf{x}}$ defines "degree of abnormality"

ML for anomaly detection

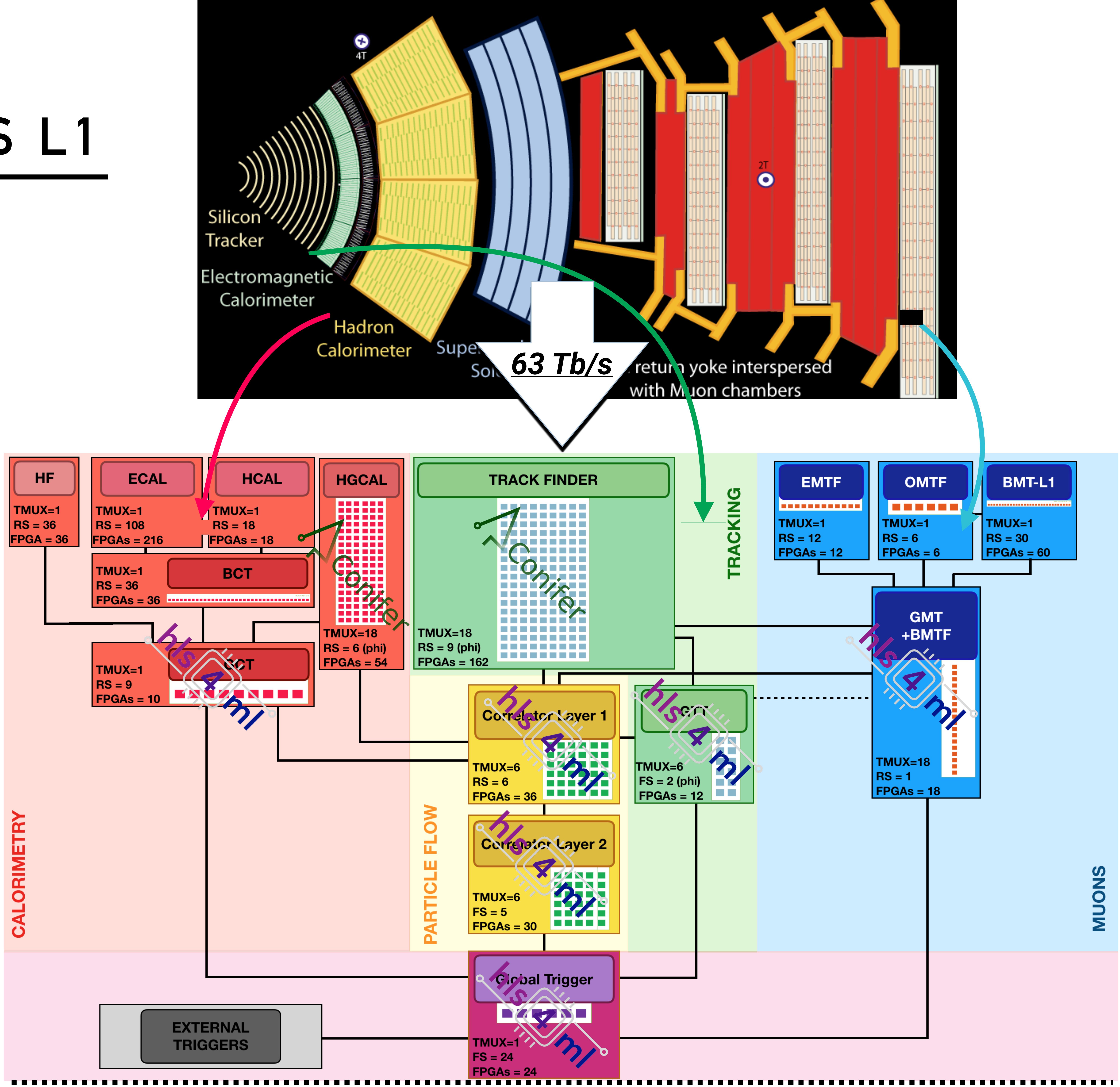




Nature Machine Intelligence 4, 154 (2022)

See also Wednesday talk on offline AD in ATLAS by Rui

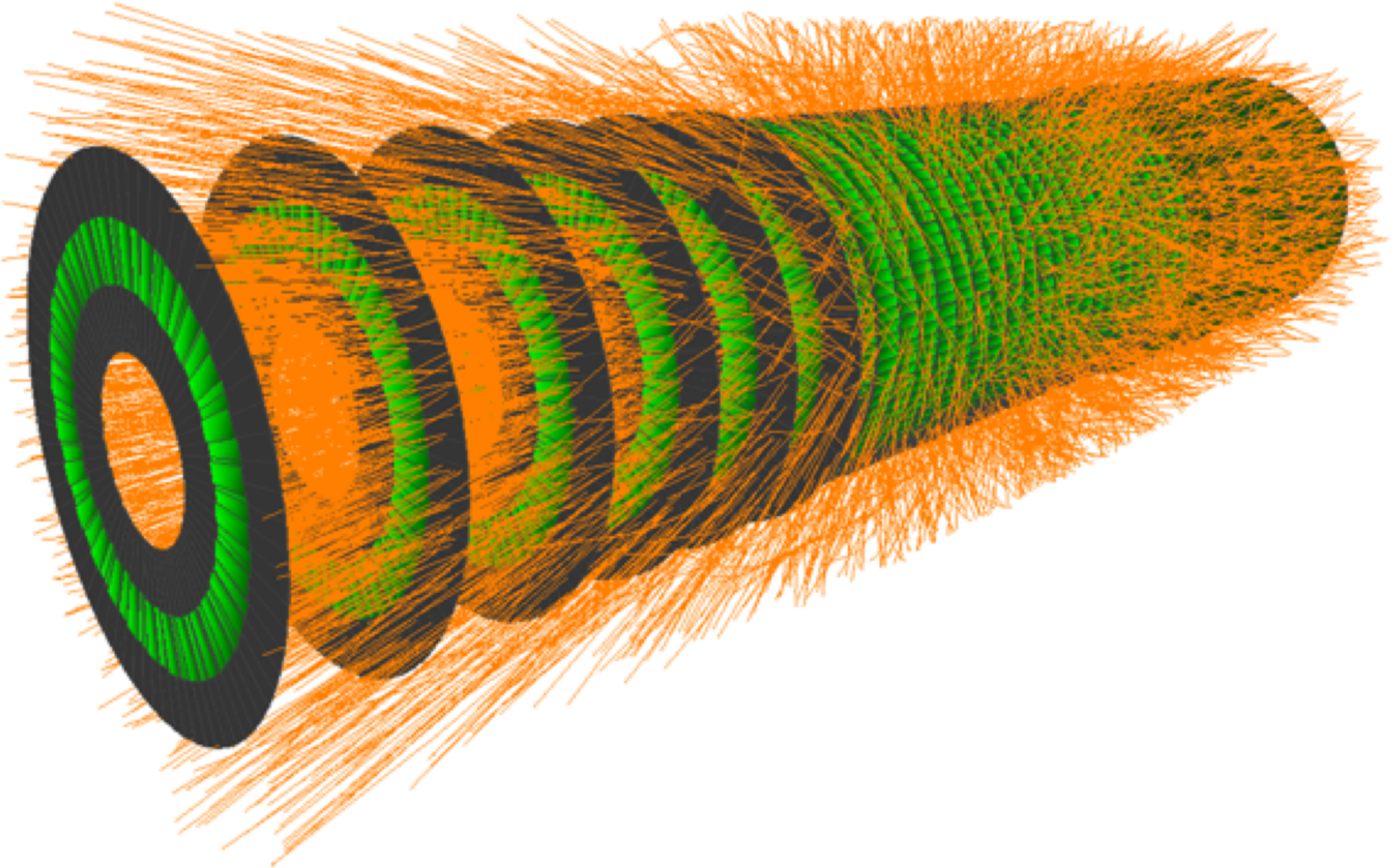
HL-LHC: CMS L1



4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
63 Tb/s
return yoke interspersed with Muon chambers
HF
TMUX=1
RS = 36
FPGA = 36
ECAL
TMUX=1
RS = 108
FPGAs = 216
HCAL
TMUX=1
RS = 18
FPGAs = 18
BCT
TMUX=1
RS = 36
FPGAs = 36
GCT
TMUX=1
RS = 9
FPGAs = 10
HGCAL
TMUX=18
RS = 6 (phi)
FPGAs = 54
TRACK FINDER
TMUX=18
RS = 9 (phi)
FPGAs = 162
Conifer
Conifer
TRACKING
Correlator Layer 1
TMUX=6
RS = 6
FPGAs = 36
Correlator Layer 2
TMUX=6
FS = 5
FPGAs = 30
GTT
TMUX=6
FS = 2 (phi)
FPGAs = 12
EMTF
TMUX=1
RS = 12
FPGAs = 12
OMTF
TMUX=1
RS = 6
FPGAs = 6
BMT-L1
TMUX=1
RS = 30
FPGAs = 60
GMT +BMTF
TMUX=18
RS = 1
FPGAs = 18
Global Trigger
TMUX=1
FS = 24
FPGAs = 24
EXTERNAL TRIGGERS
hls 4 ml
CALORIMETRY
PARTICLE FLOW
MUONS

DOI:10.3389/fdata.2022.828666

ML for tracking



In HL-LHC, will need to do track finding at L1

- O(1000) hits, O(100) tracks, 40 MHz rate, ~5 μs latency

Graph Neural Networks for fast charged particle tracking

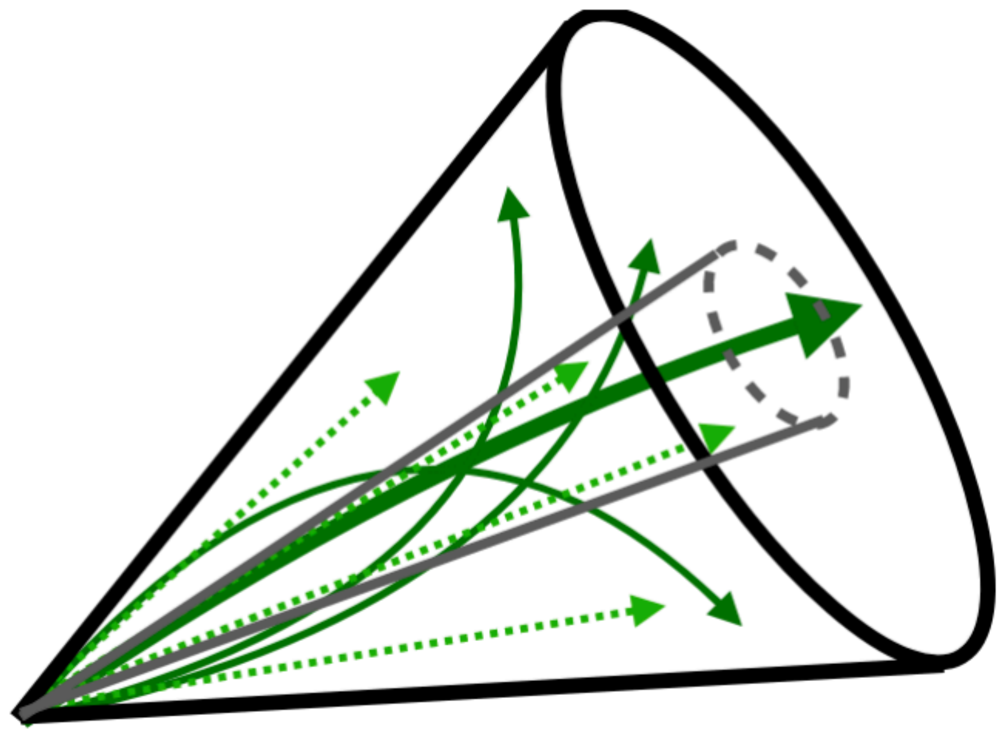
- Custom converter for PyTorch Geometric integrated in hls4ml

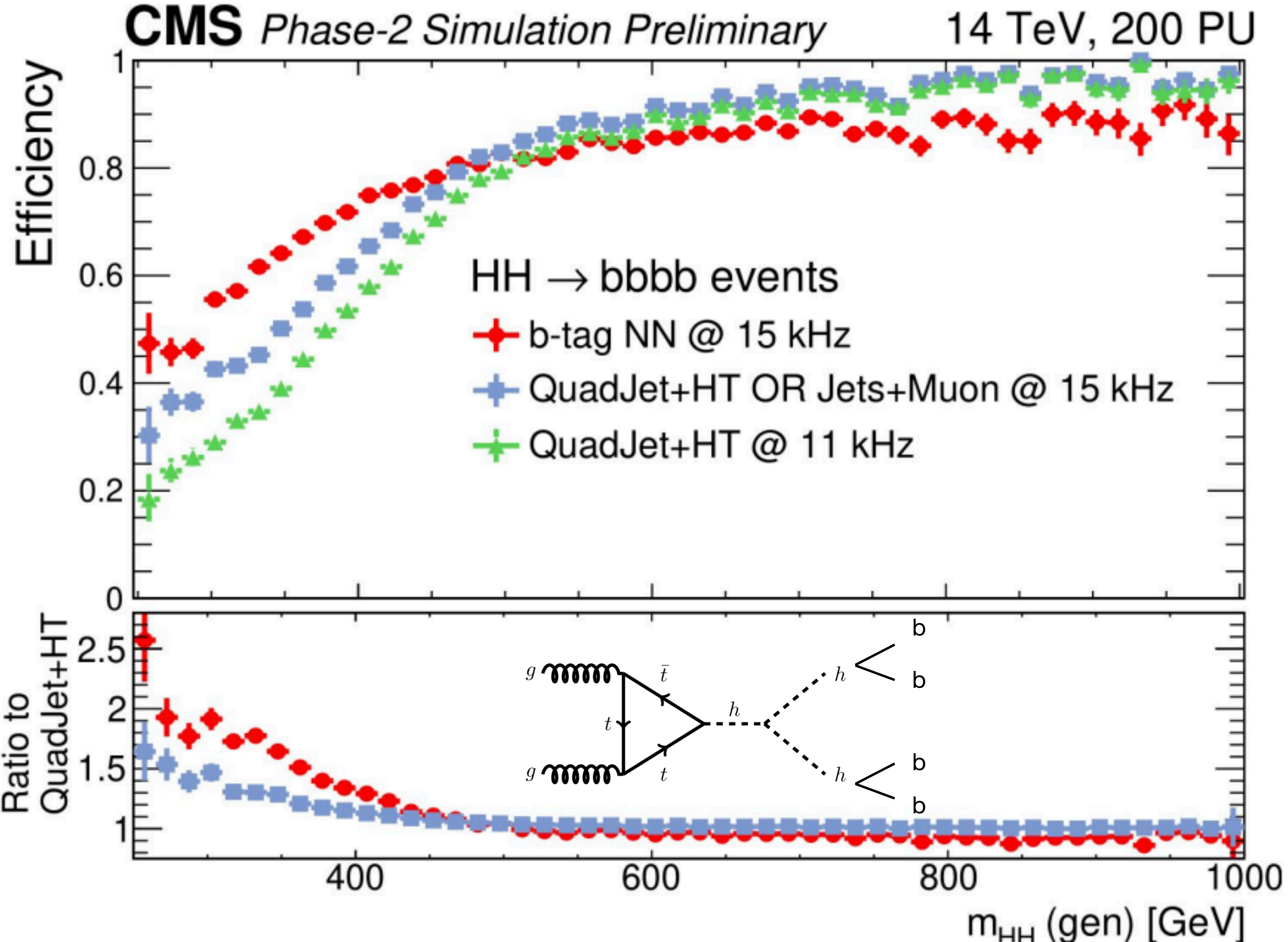
Throughput-optimized for L1 applications,
resource-optimised for co-processing

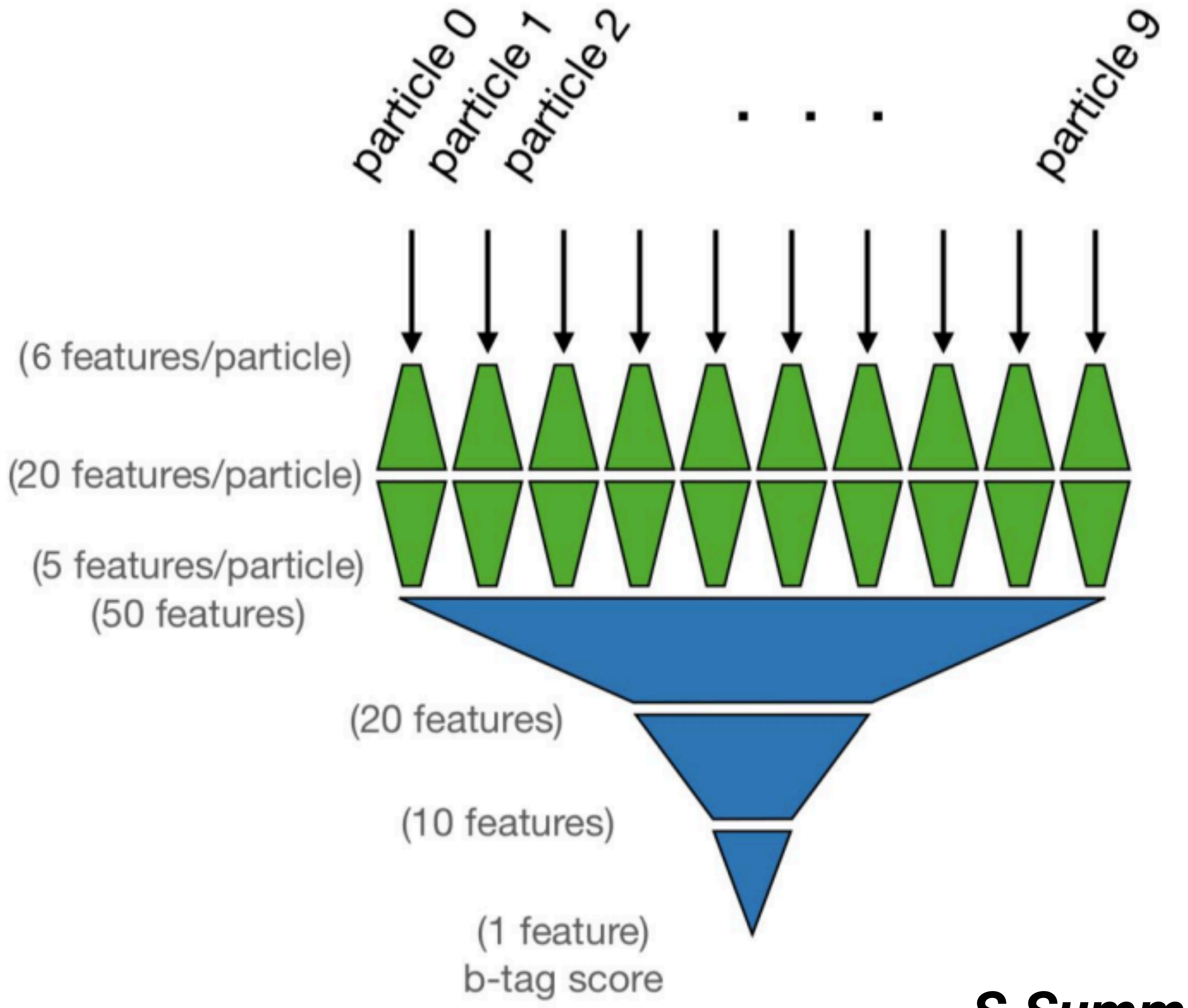
Design	$(n_{\text{nodes}}, n_{\text{edges}})$	RF	Precision	Latency [cycles]	II [cycles]	DSP [%]	LUT [%]	FF [%]	BRAM [%]
Throughput-opt.	(28, 56)	1	ap_fixed<14,7>	59	1	99.9	66.0	11.7	0.7
Resource-opt.	(28, 56)	1	ap_fixed<14,7>	79	28	56.6	17.6	3.9	13.1

ML for jet tagging

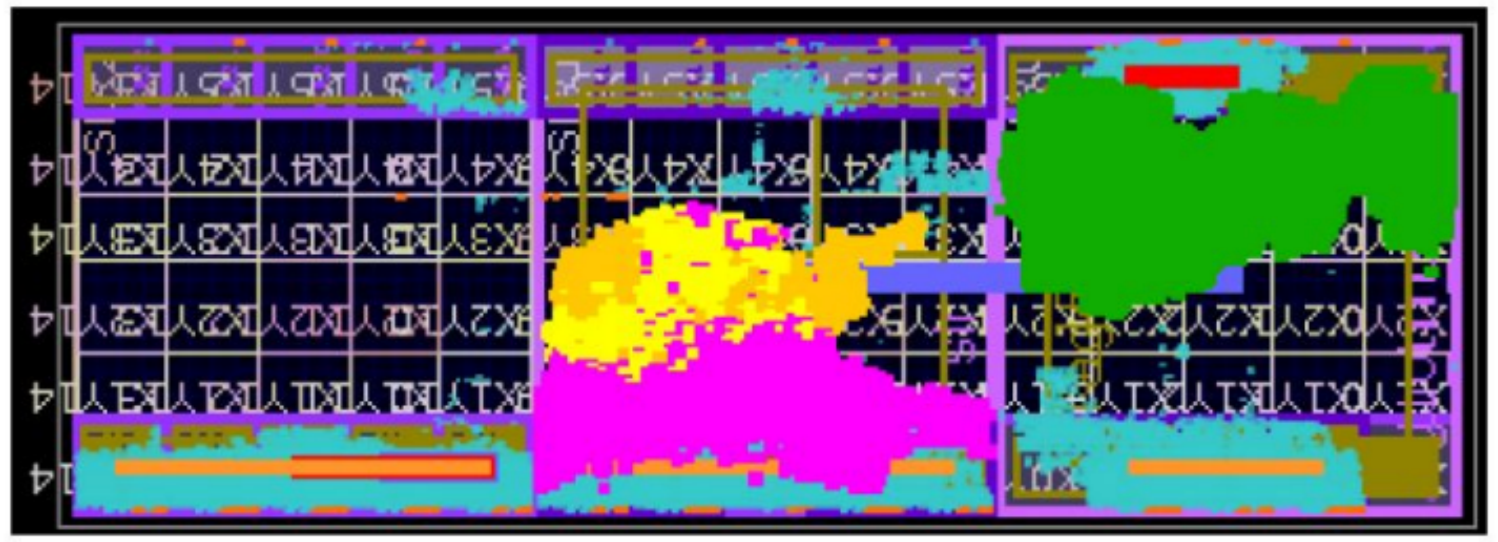
cds.cern.ch/record/2814728/







S.Summers

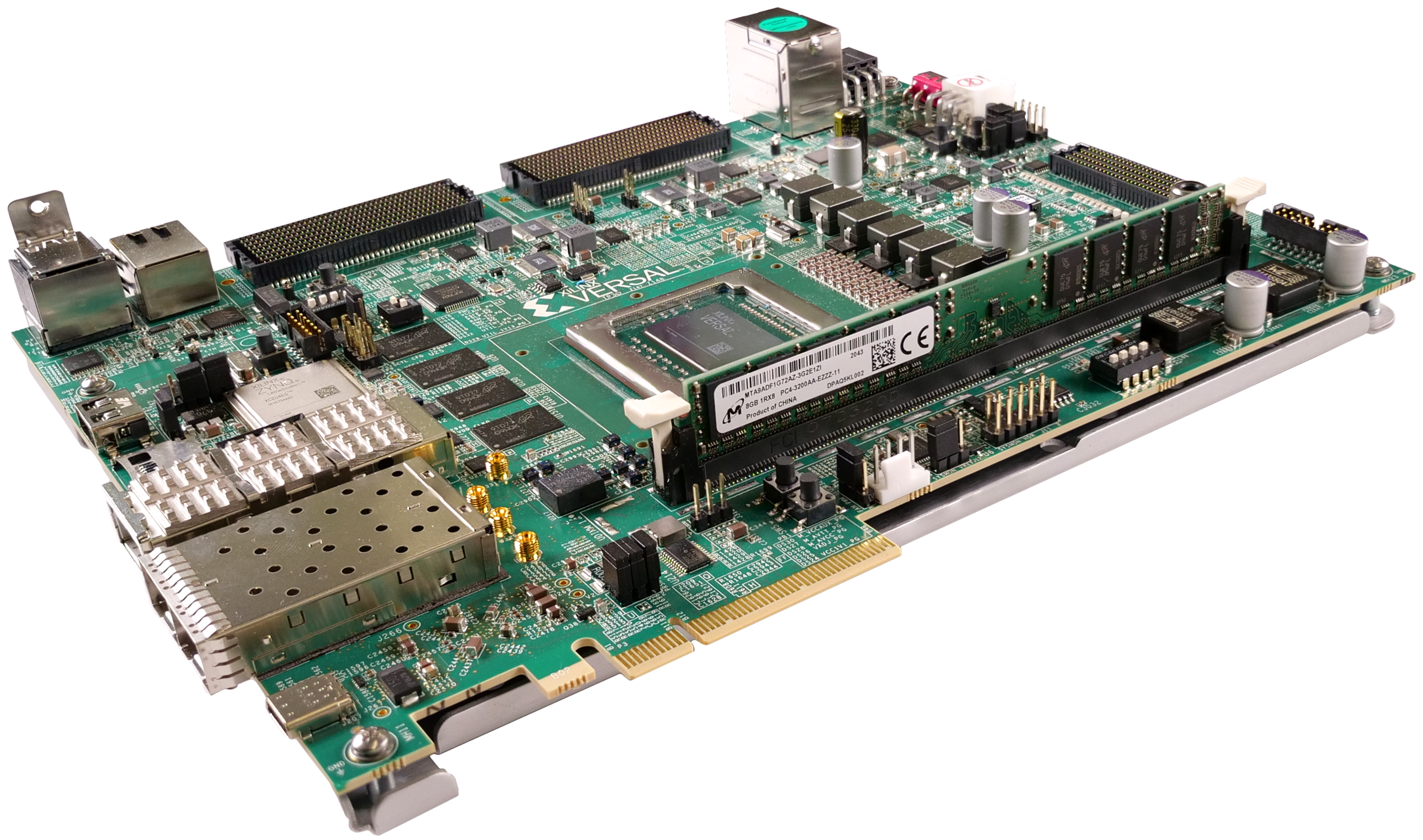


Jet Finding
b tag NN

AI engines

More and more dedicated AI processors on the market

- Can we utilise highly specialised ML hardware at CERN?



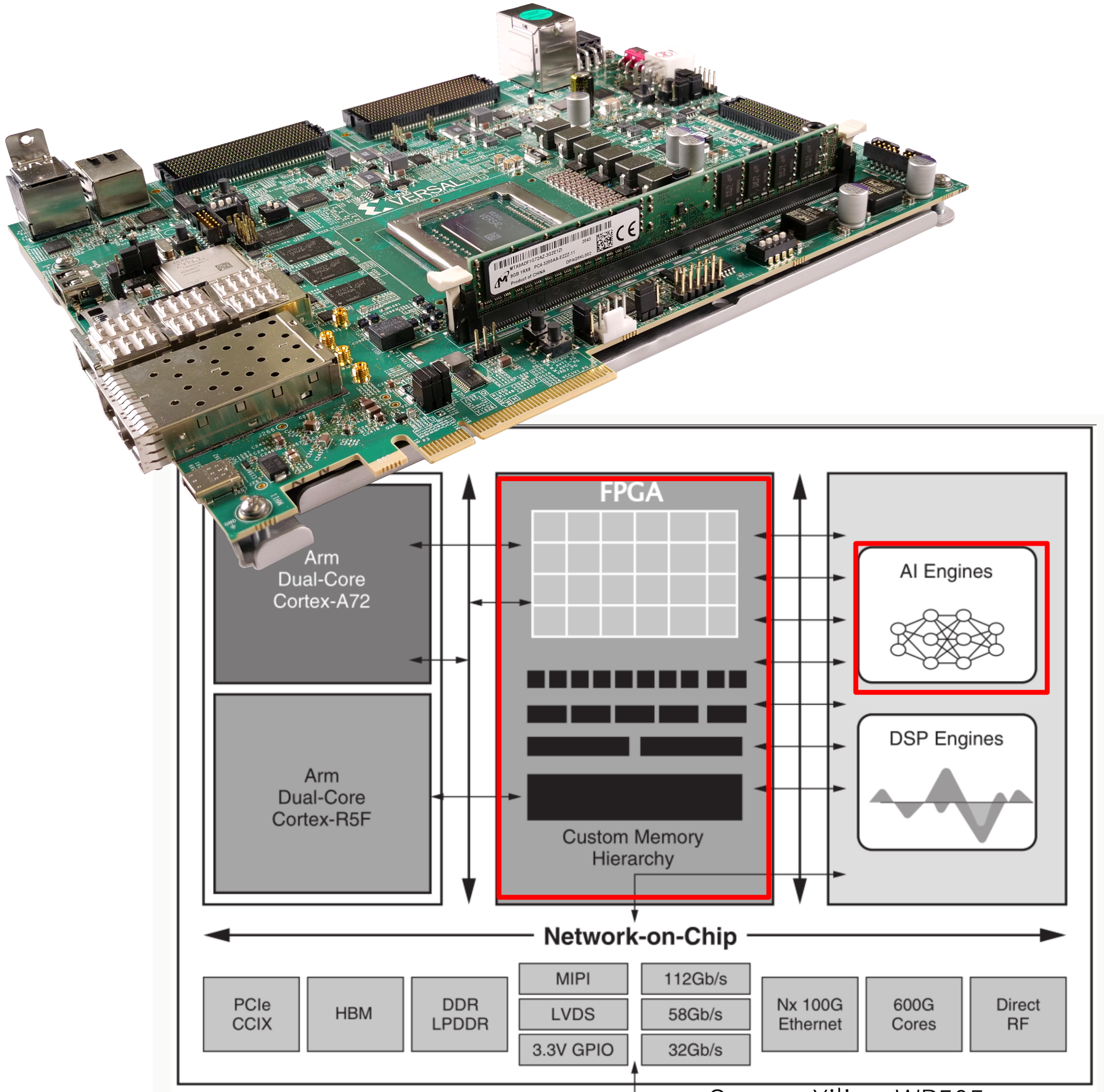
AI engines

GNNs with Versal AI, P. Schwaebig

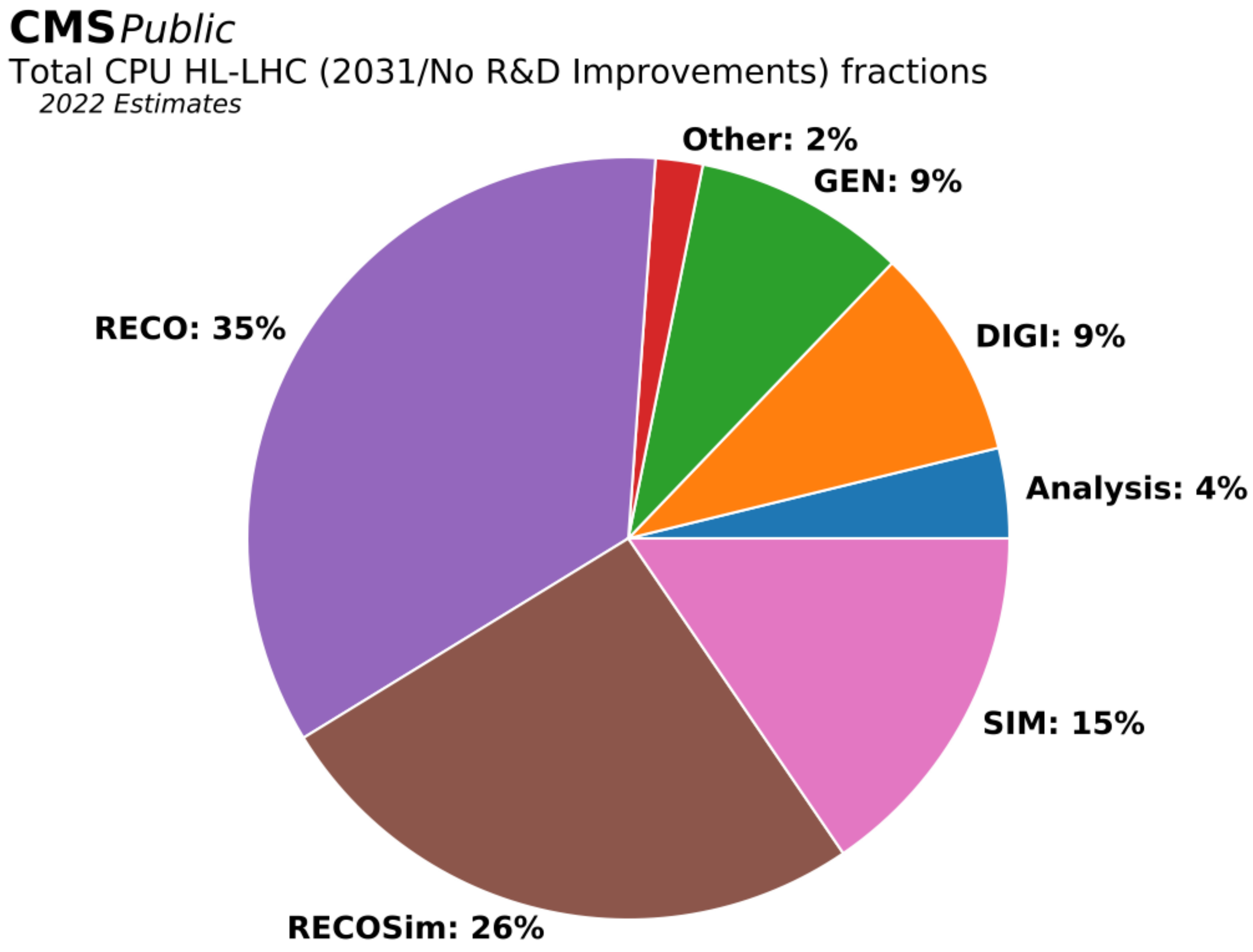
More and more dedicated AI processors on the market
- Can we utilise highly specialised ML hardware at CERN?

Xilinx Versal AI processors
- 400 AI processors, ~2M logic cells (FPGA),
  2k DSPs, Arm CPU, Arm RPU
- Data can move back and forth between AI Engines and FPGA

Explored for real-time tracking in trigger application
- GNN for pattern recognition



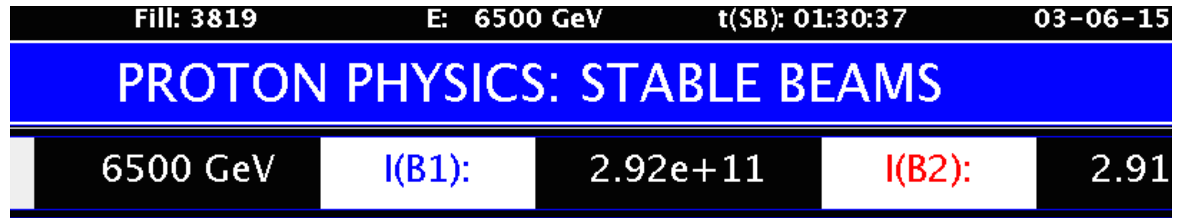
Source: Xilinx, WP505,

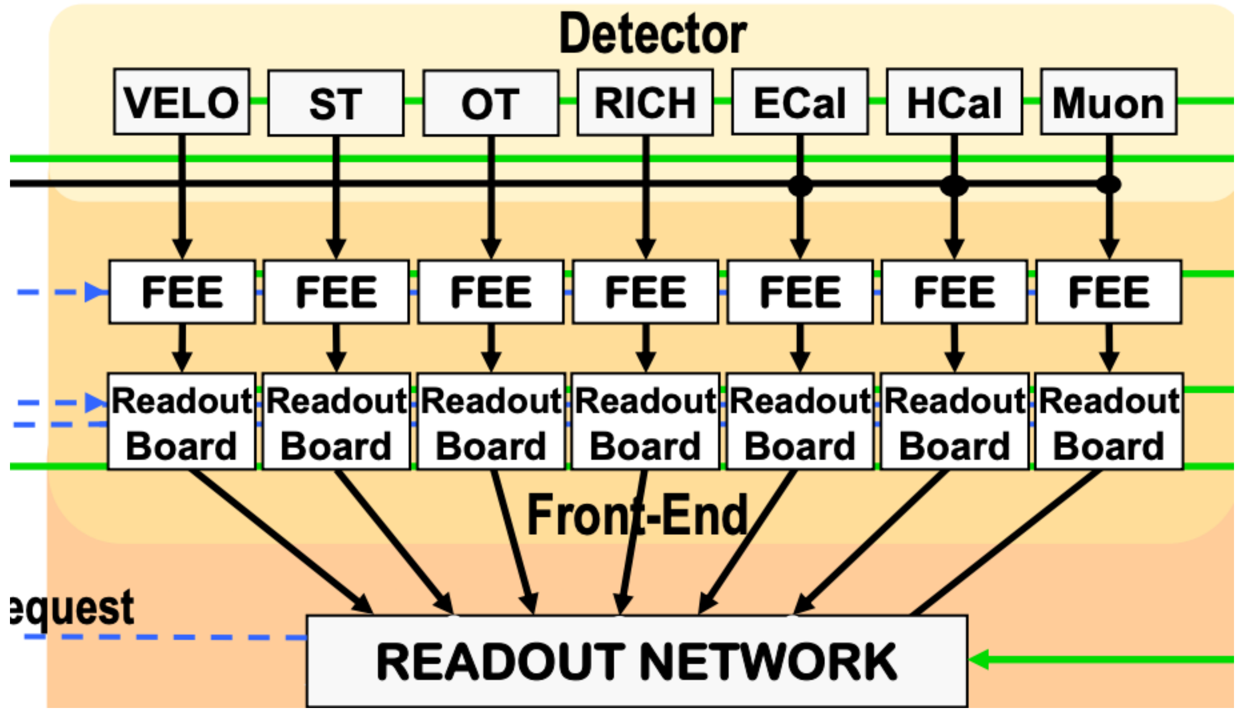


See Vangelis Kourlitis' talk

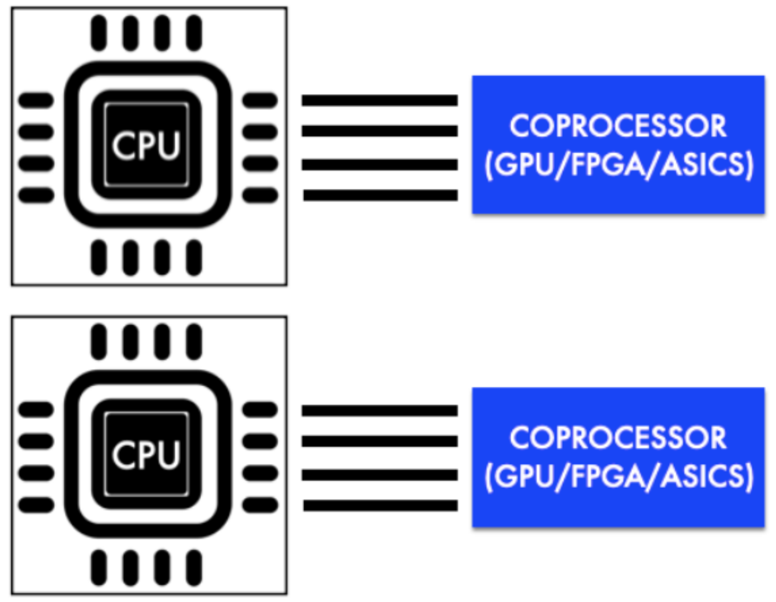
FPGAs as accelerators

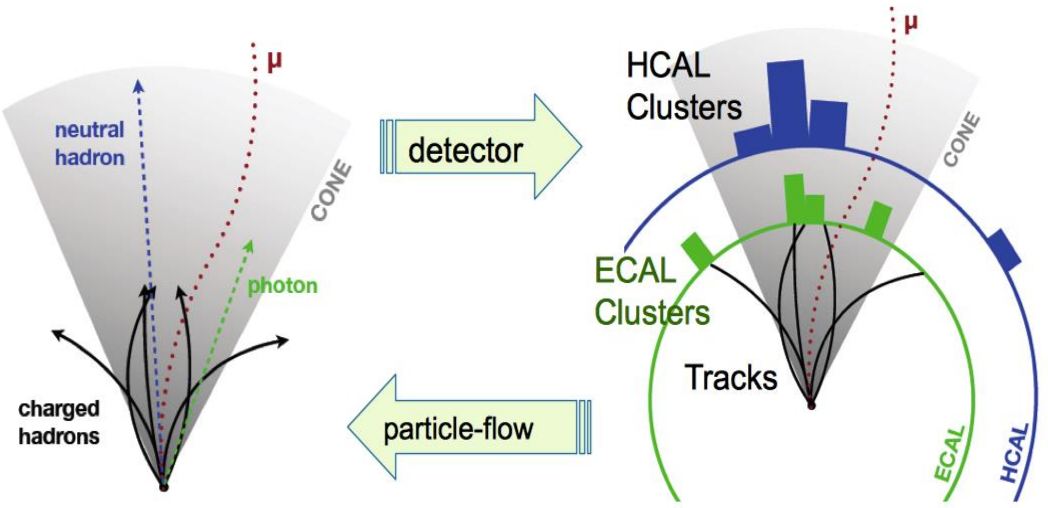
DAQ





CO-PROCESSING



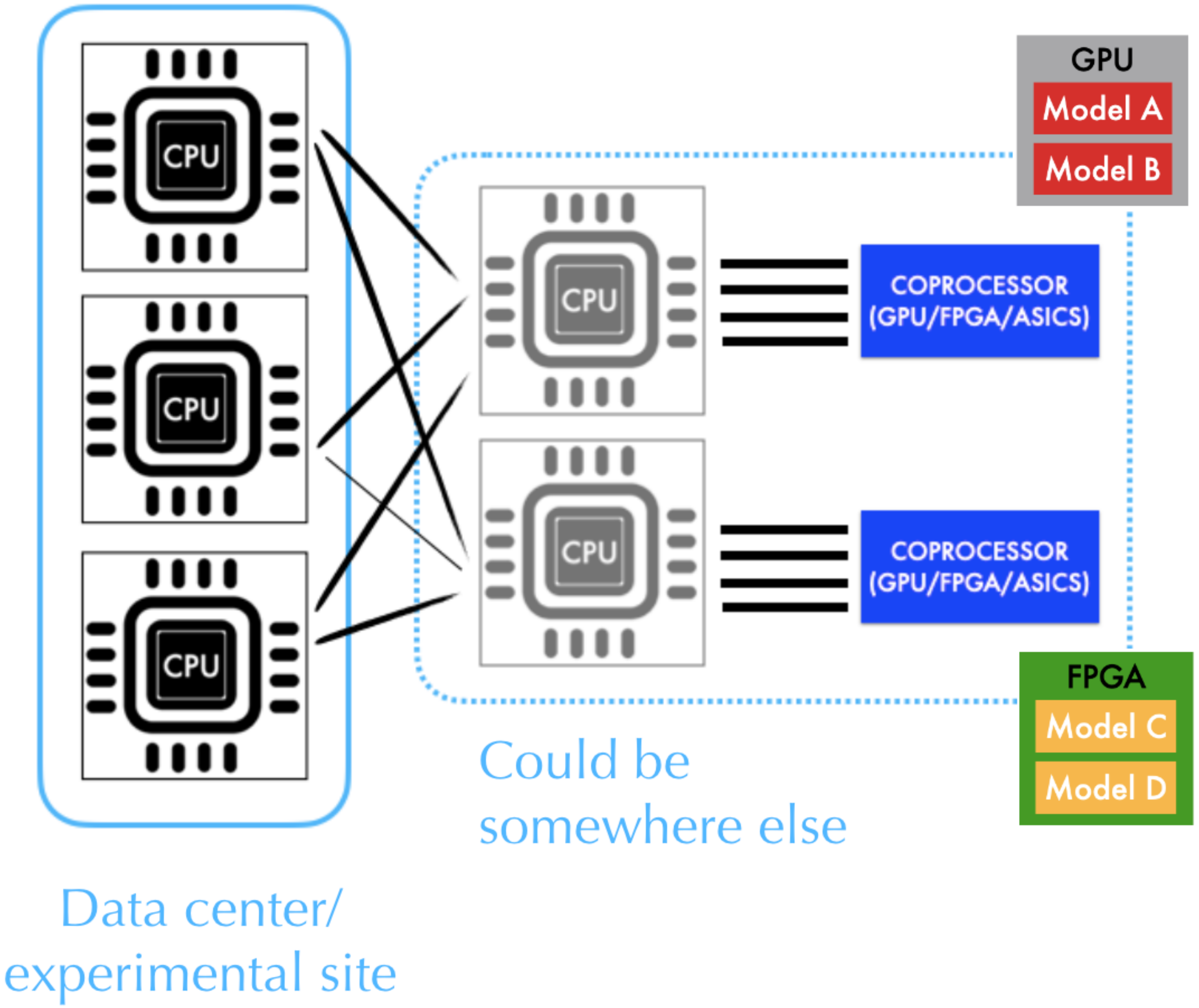


C. Beteta, I. Bezshyiko, N. Serra

FPGAs as accelerators

Alternative: FPGA-as-a-Service toolkit for Cloud inference

- Use hls4ml to deploy large models on FPGA
  →run inference in the cloud



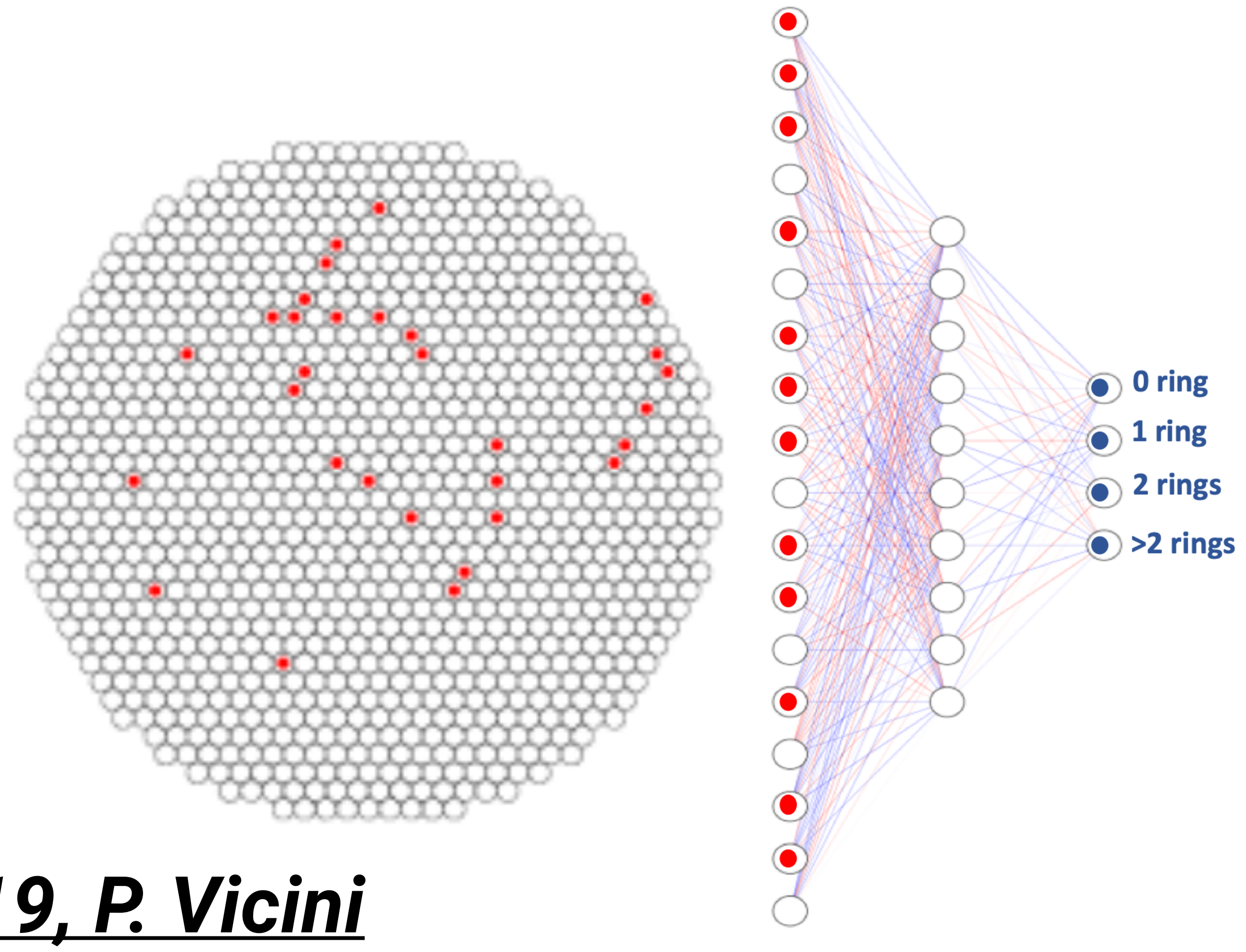
Algorithm	Platform	Number of Devices	Batch Size	Inf./s [Hz]	Bandwidth [Gbps]
FACILE	AWS EC2 F1	1	16,000	36 M	23
FACILE	Alveo U250	1	16,000	86 M	55
FACILE	T4 GPU	1	16,000	8 M	5.1

FaaST, D. Rankin et. al

hls4ml in other CERN experiments

NA62: Measuring BR($K^+ \to \pi^+ \nu \bar{\nu}$) = O($10^{-11}$)

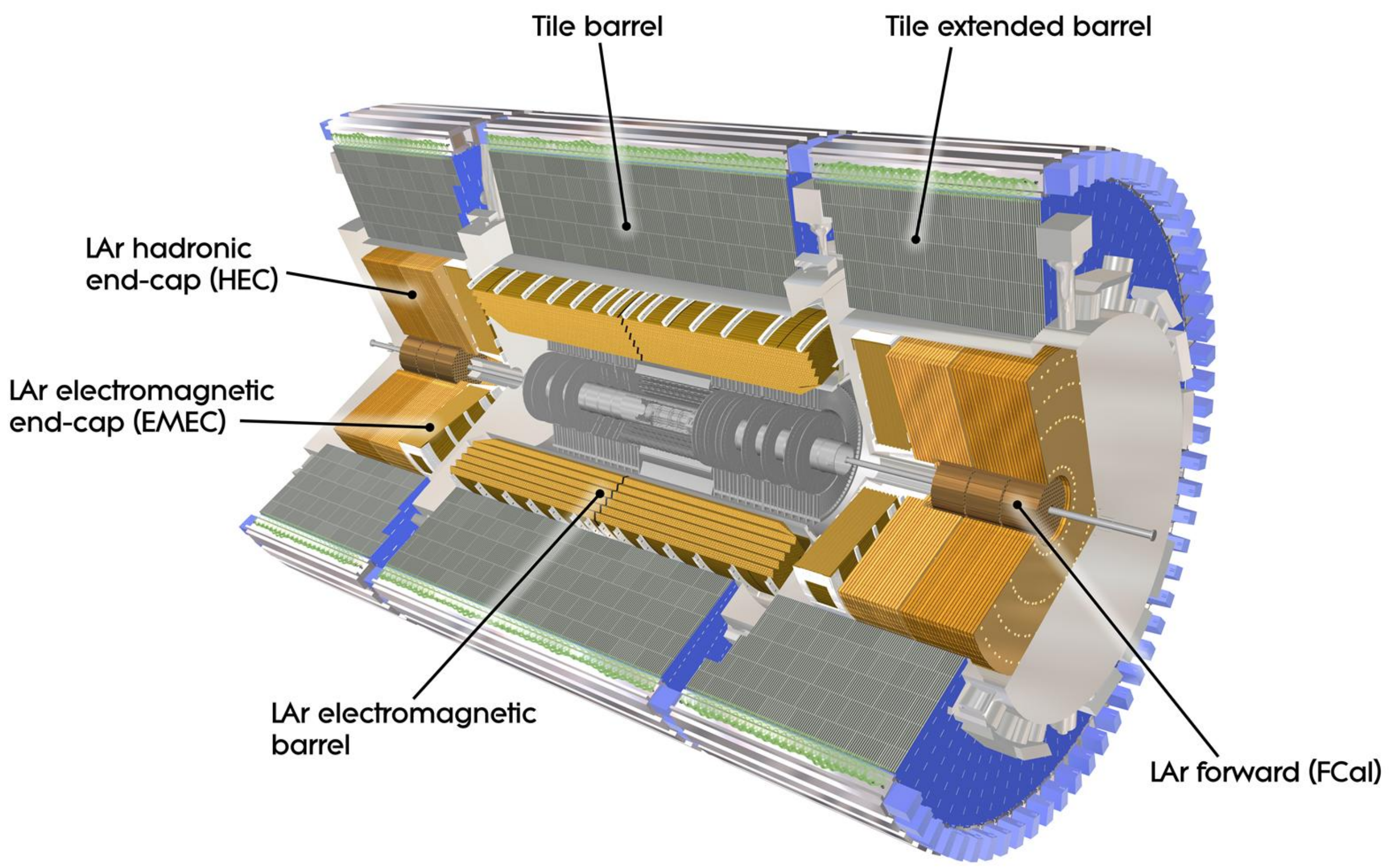
- FPGA trigger 800 MHz→1 MHz



CHEP 2019, P. Vicini

ATLAS Liquid Argon Calorimeter (R&D)

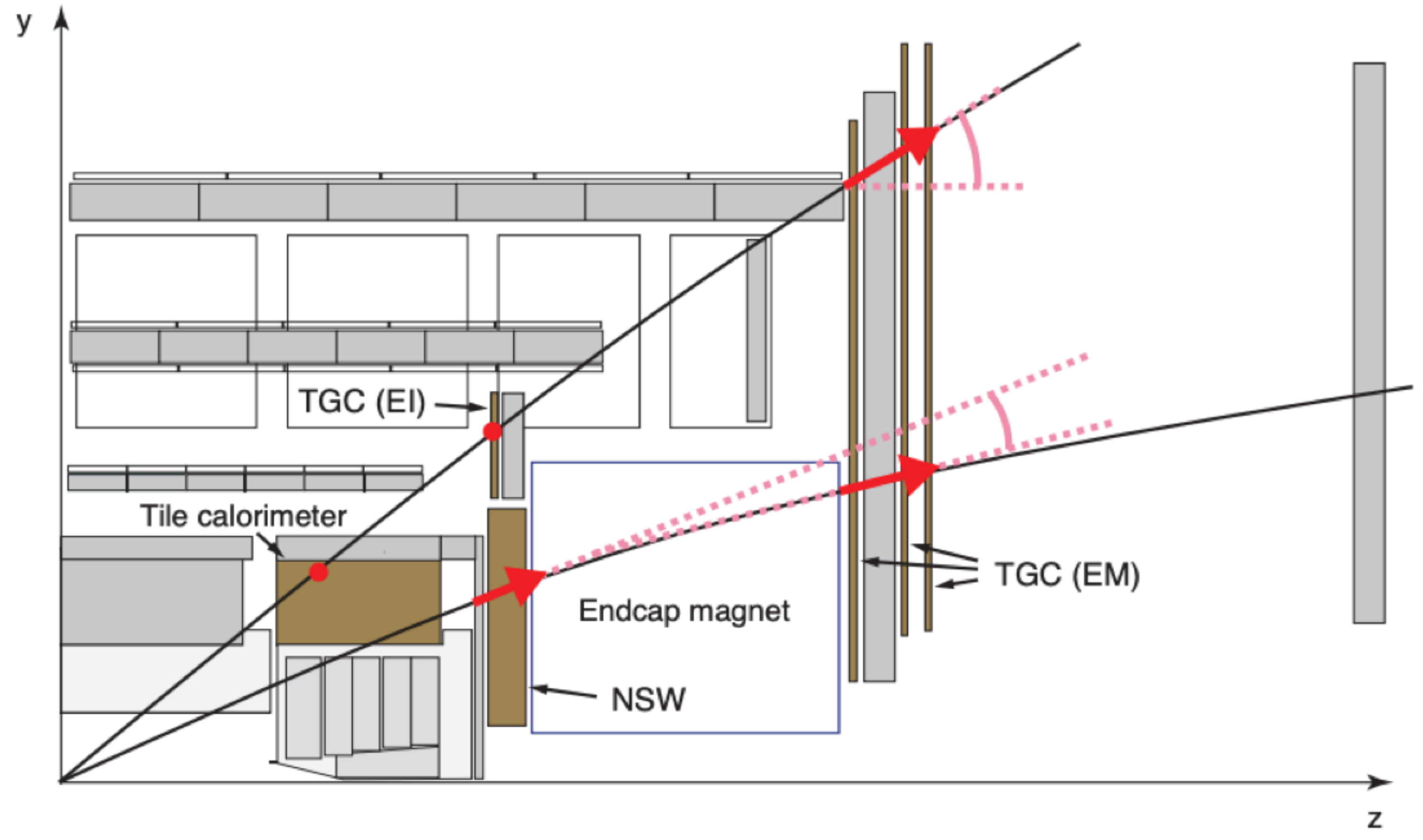
- RNN for real-time energy reconstruction
- ~200 ns on Intel Stratix-10 FPGA



DOI:10.1007/s41781-021-00066-y

ATLAS small wheel muon segment finding and reconstruction (R&D)

- Regression of muon position and angle
- 400 ns budget



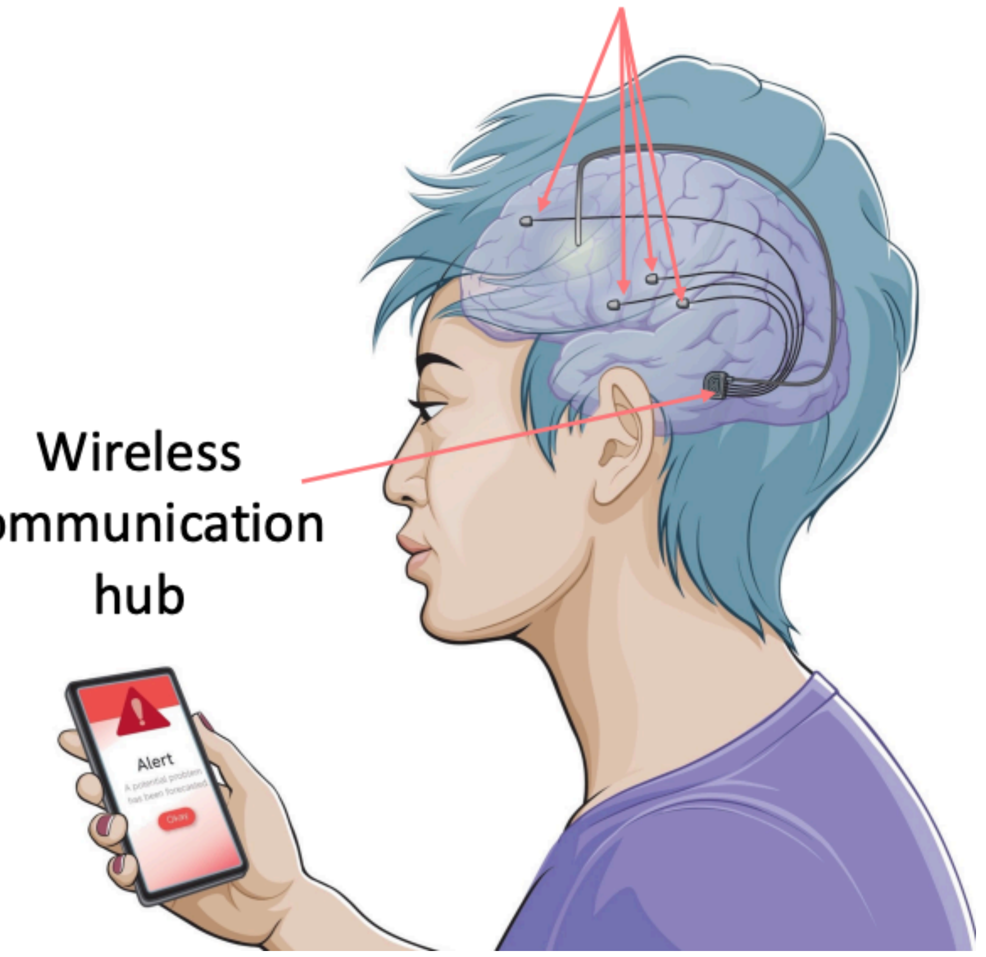
R. Teixeira de Lima, R Rojas Caballero et al.

...and outside of HEP

Semantic segmentation for autonomous vehicles

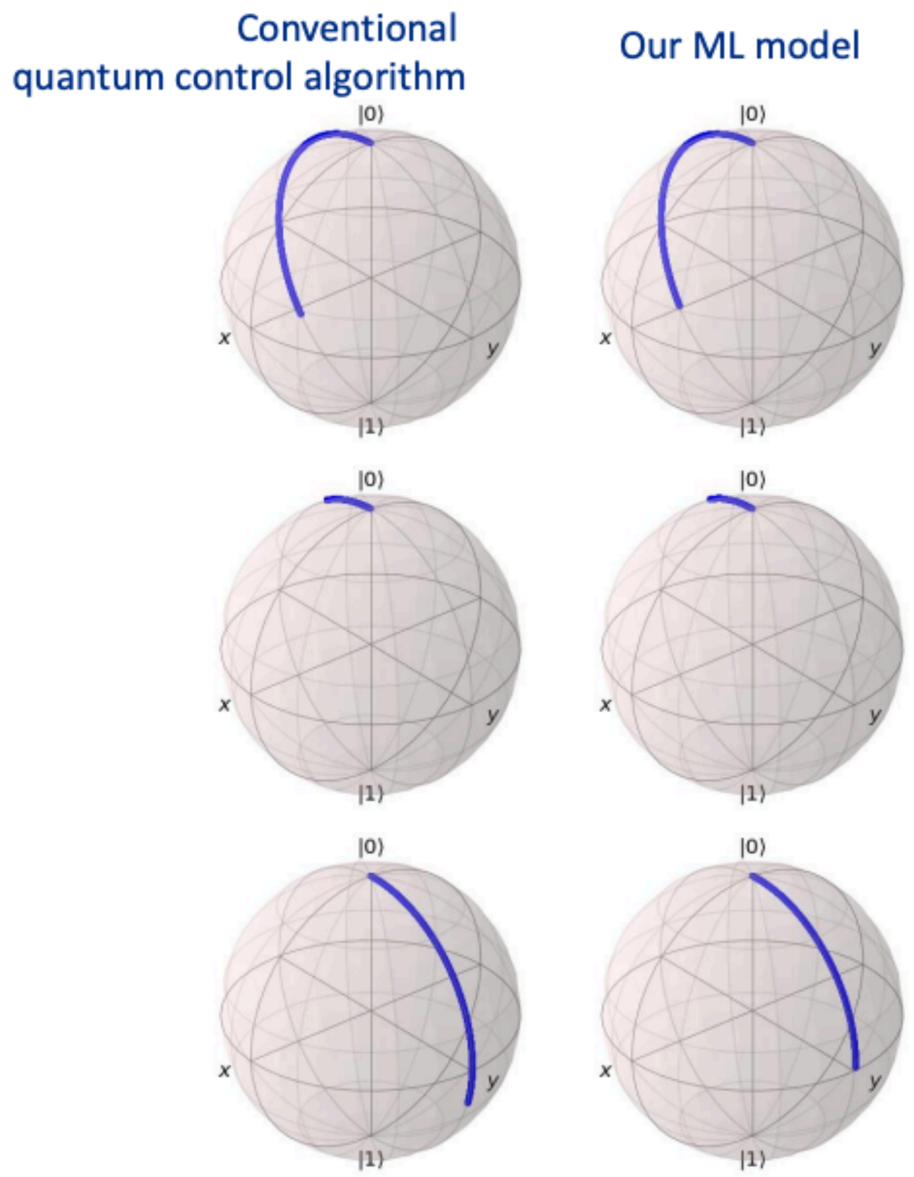
N. Ghielmetti et al.

Seizure Predicting Brain Implant



W. Lemaire et al.

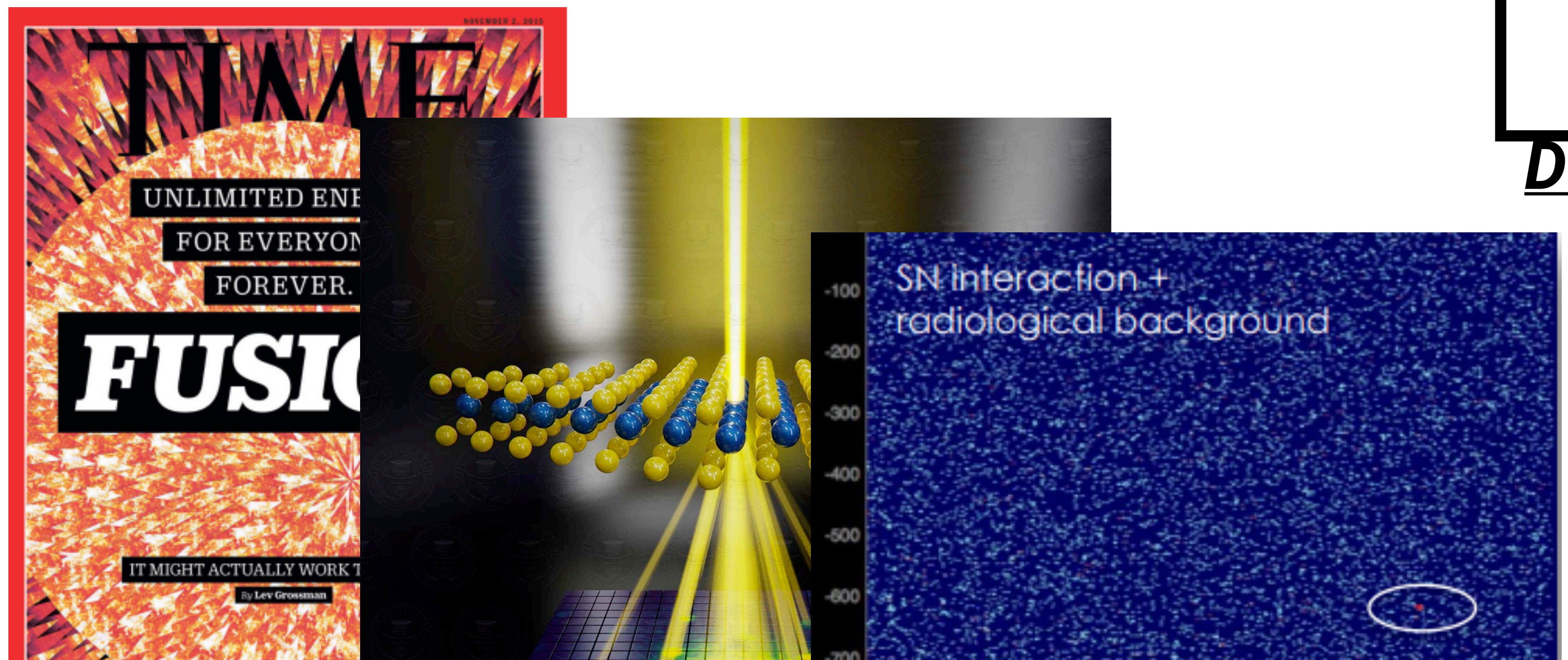
NN accelerator for quantum control



D Xu et al.

Other examples

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

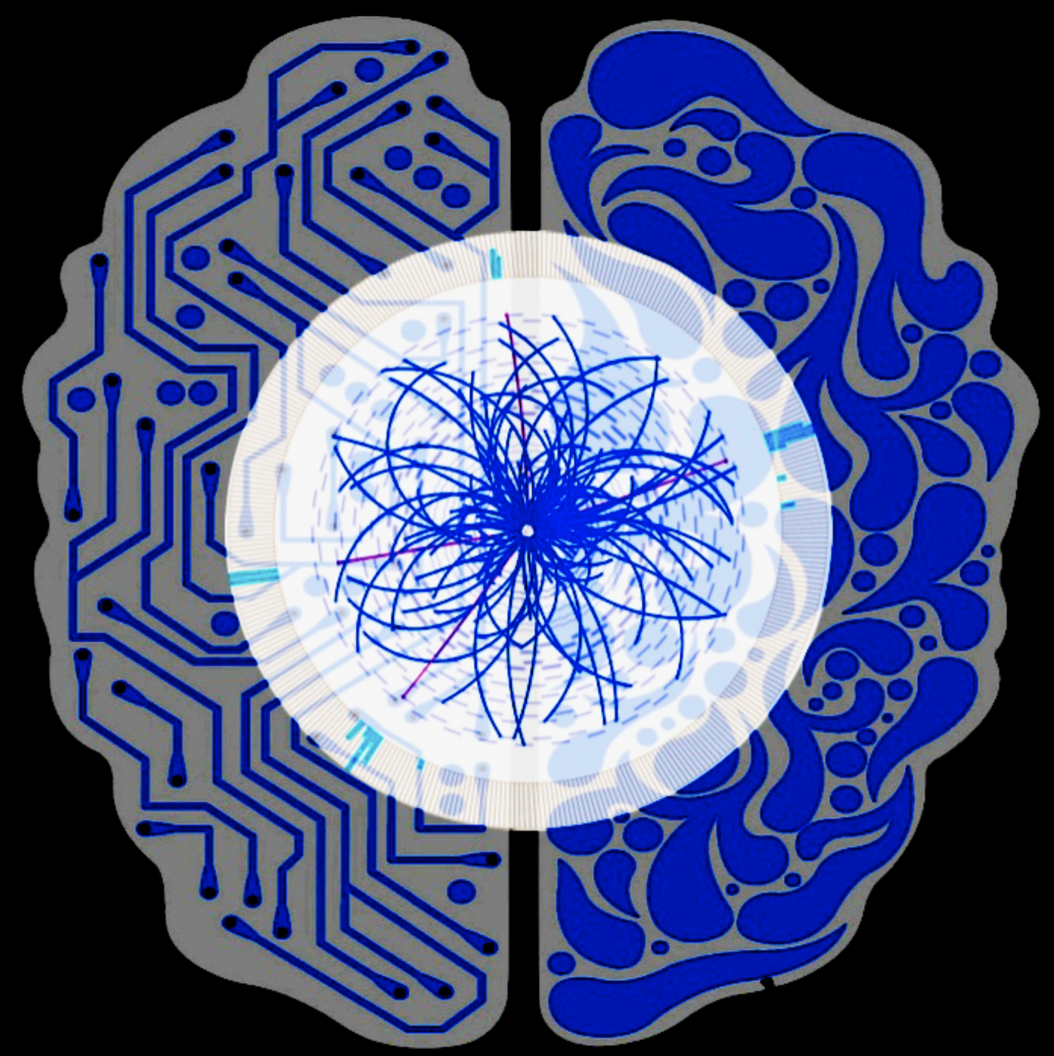


Conifer

Join the community:

fastmachinelearning.org

Sign up to the hls-fml group



BACKUP

What are FPGAs?

Field Programmable Gate Arrays

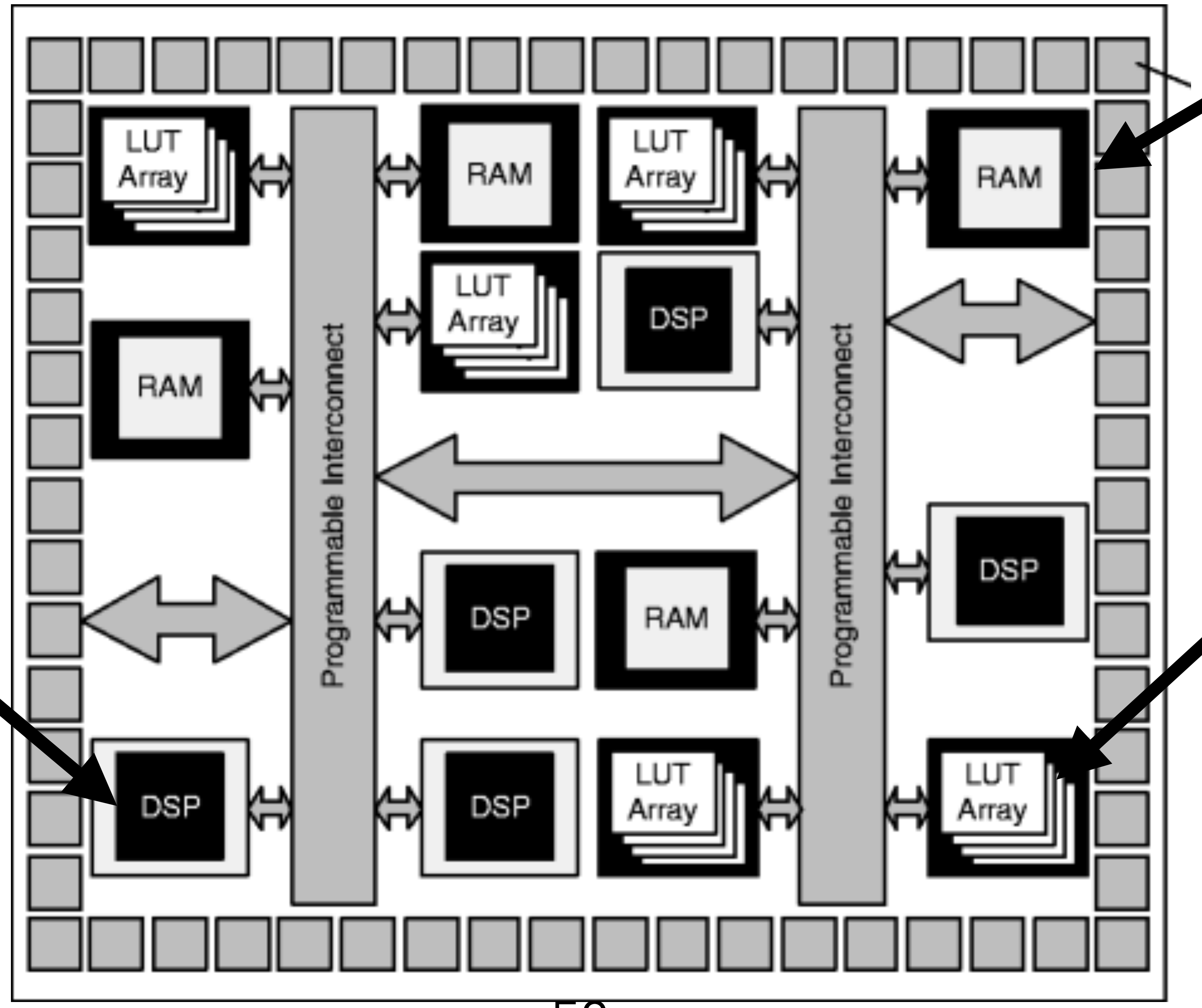
- Different resources with programmable interconnects (reprogrammable)
- Originally ASIC prototyping, now also for high performance computing

What are FPGAs?

Field Programmable Gate Arrays

- Different resources with programmable interconnects (reprogrammable)
- Originally ASIC prototyping, now also for high performance computing

Digital signal processors (DSPs):
specialised for multiplication



Memory (BRAM)

Logic cells/lookup tables (LUTs):
perform arbitrary functions

flip-flops (FF):
registers data in time with clock pulse

Why FPGAs at LHC?

Why FPGAs at LHC?

High parallelism = **Low latency**

- Can work on different data simultaneously (pipelining)! **High bandwidth**

Why FPGAs at LHC?

High parallelism = **Low latency**

- Can work on different data simultaneously (pipelining)! **High bandwidth**

Power efficient

- FPGAS ~x10 more power efficient than GPUs
  (for Phase-2, FPGAs dissipate heat of ~7W/cm2 while processing 5% of total internet traffic!)

Why FPGAs at LHC?

High parallelism = **Low latency**
- Can work on different data simultaneously (pipelining)! **High bandwidth**

Power efficient
- FPGAS ~x10 more power efficient than GPUs
  (for Phase-2, FPGAs dissipate heat of ~7W/cm2 while processing 5% of total internet traffic!)

Latency deterministic
- CPU/GPU has processing randomness, FPGAs **repeatable and predictable latency**

https://github.com/Xilinx/brevitas

QPYTORCH ?

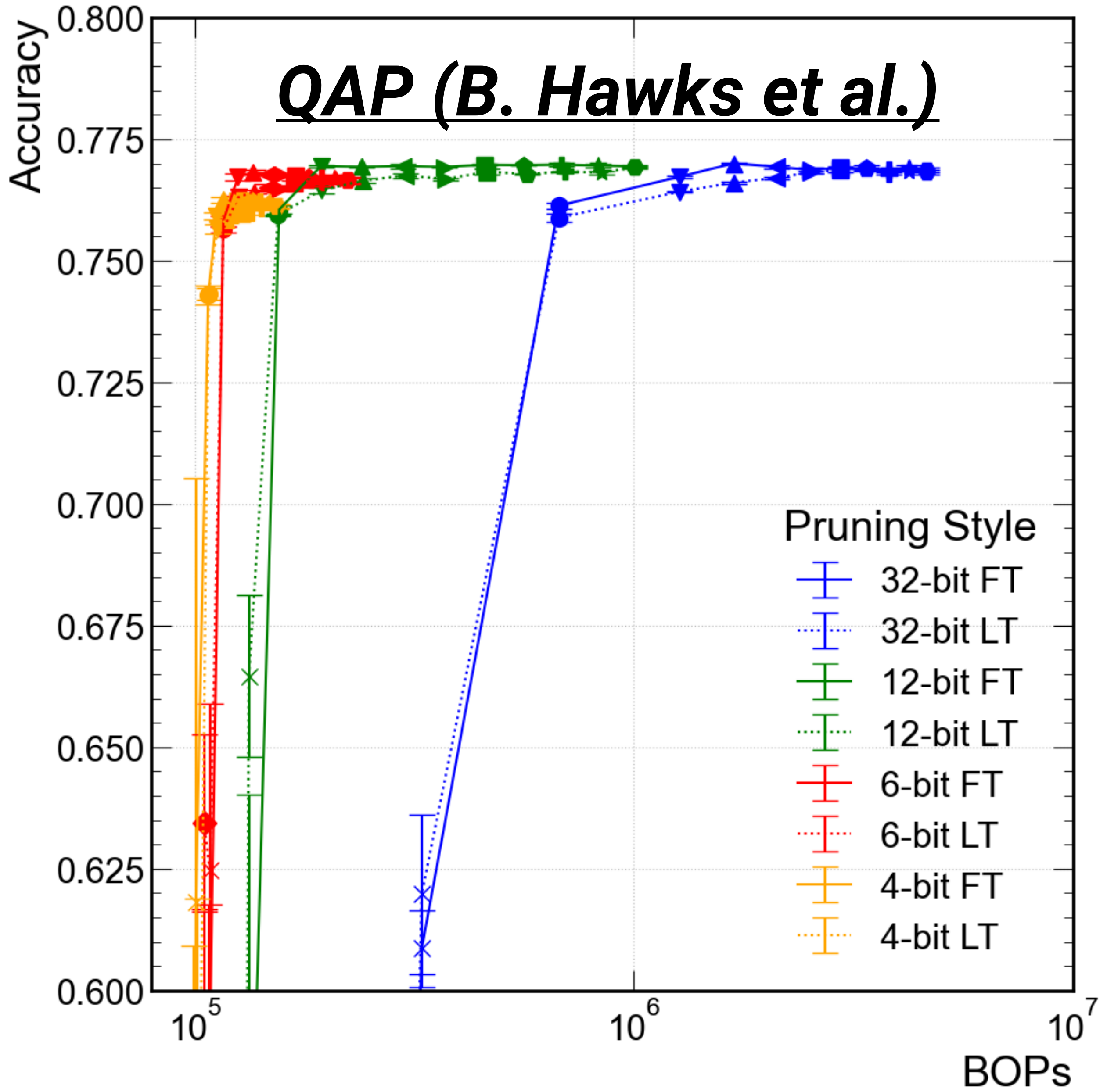
```
import brevitas.nn as qnn
qnn.
quant_bn (brevitas.nn)
QuantCat                       brevitas.nn.quant_eltwise
QuantTanh                      brevitas.nn.quant_activation
ScaleBias                      brevitas.nn.quant_scale_bias
quant_conv (brevitas.nn)
hadamard_classifier (brevitas.nn)
quant_accumulator (brevitas.nn)
quant_activation (brevitas.nn)
quant_avg_pool (brevitas.nn)
quant_convtranspose (brevitas.nn)
quant_dropout (brevitas.nn)
quant_eltwise (brevitas.nn)
quant_linear (brevitas.nn)
quant_max_pool (brevitas.nn)
quant_scale_bias (brevitas.nn)
quant_upsample (brevitas.nn)
BatchNorm1dToQuantScaleBias    brevitas.nn.quant_bn
BatchNorm2dToQuantScaleBias    brevitas.nn.quant_bn
ClampQuantAccumulator          brevitas.nn.quant_accumulator
```

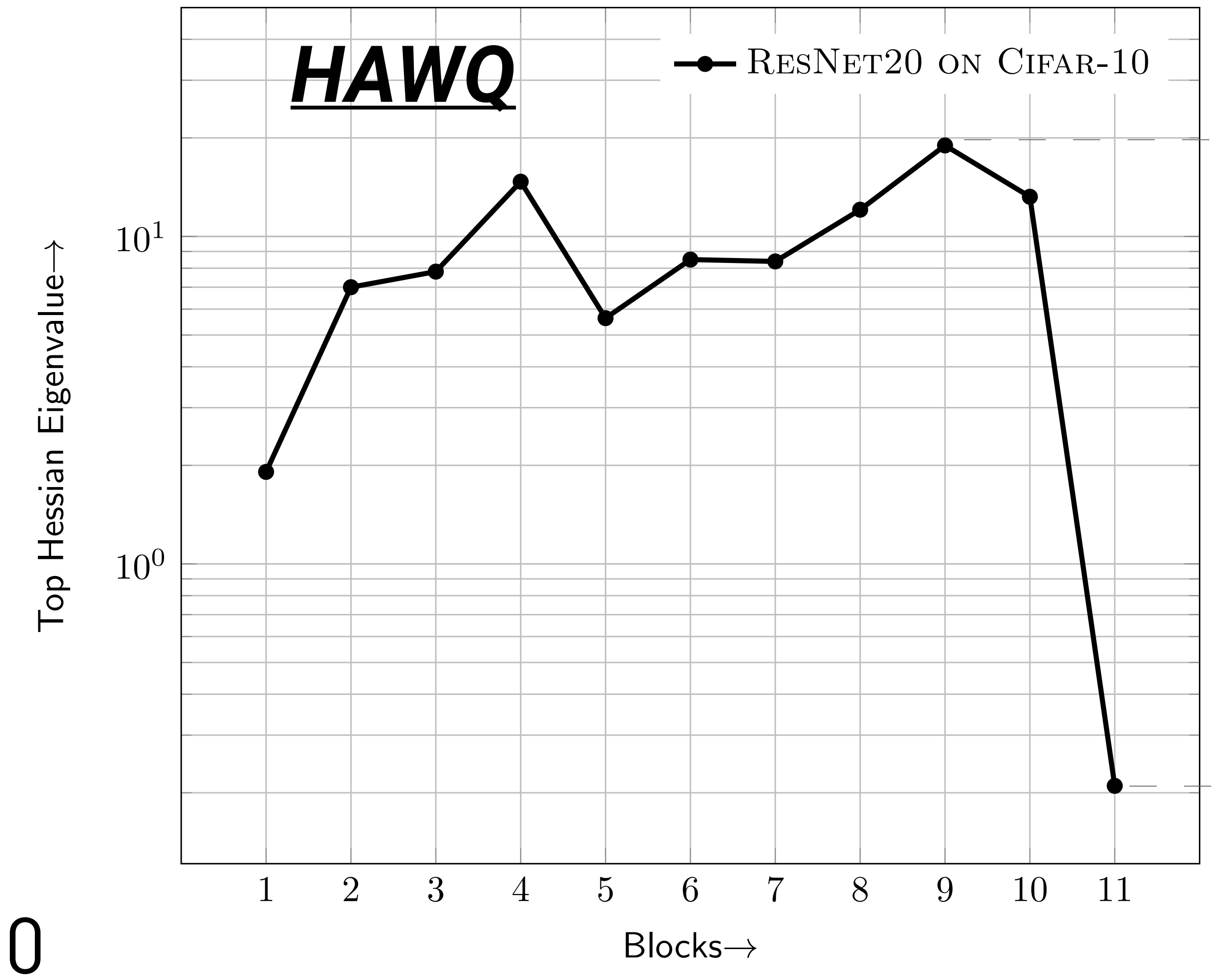
Brevitas like QKeras, but for PyTorch

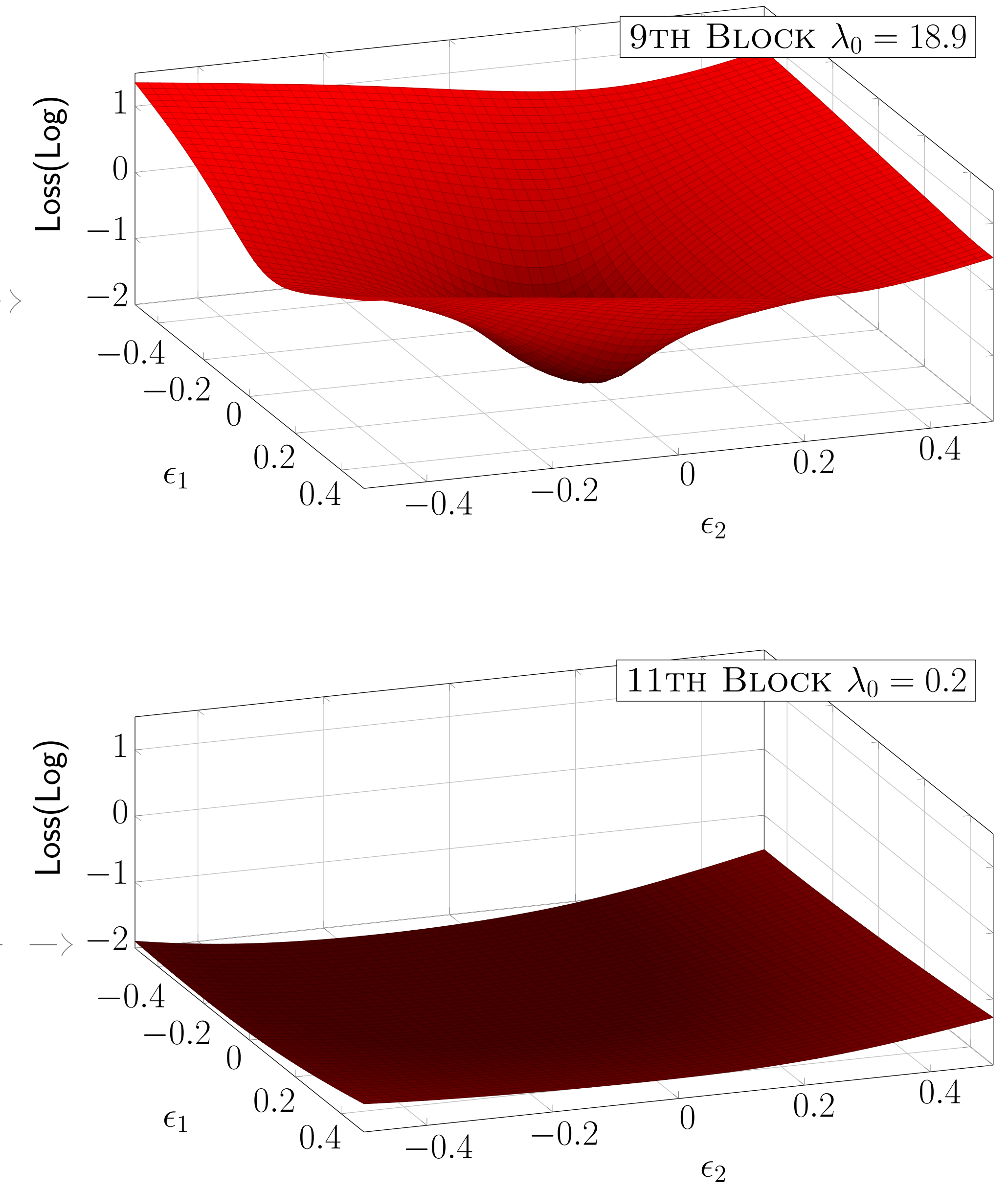
- QAT library
- Support most common layers and activation functions

Other quantization techniques:

- ***HAWQ: Hessian AWare Quantization***
- ***Quantization Aware Pruning (B. Hawks et al.)***







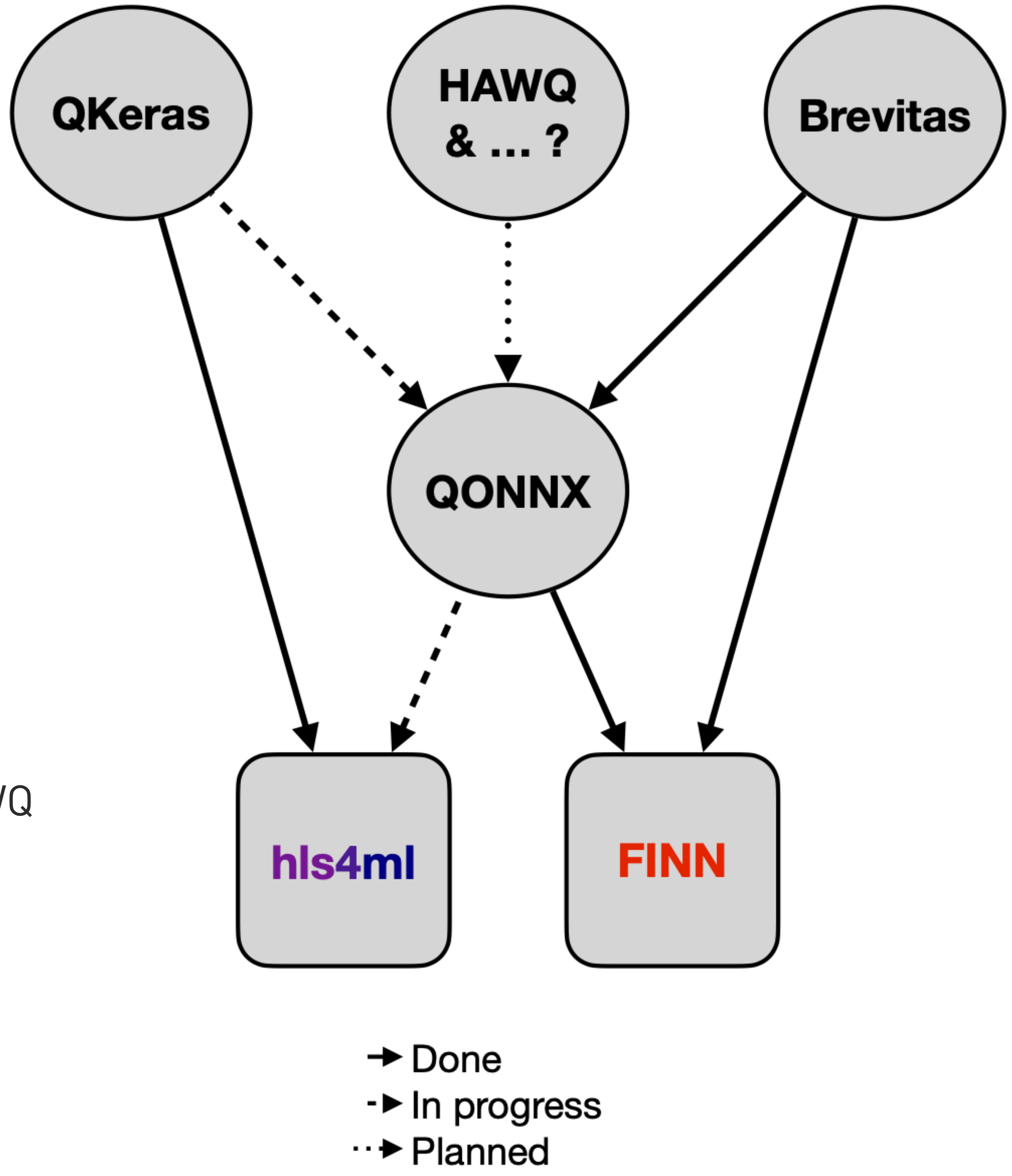
QPYTORCH ?

Brevitas like QKeras, but for PyTorch
- QAT library
- Support most common layers and activation functions

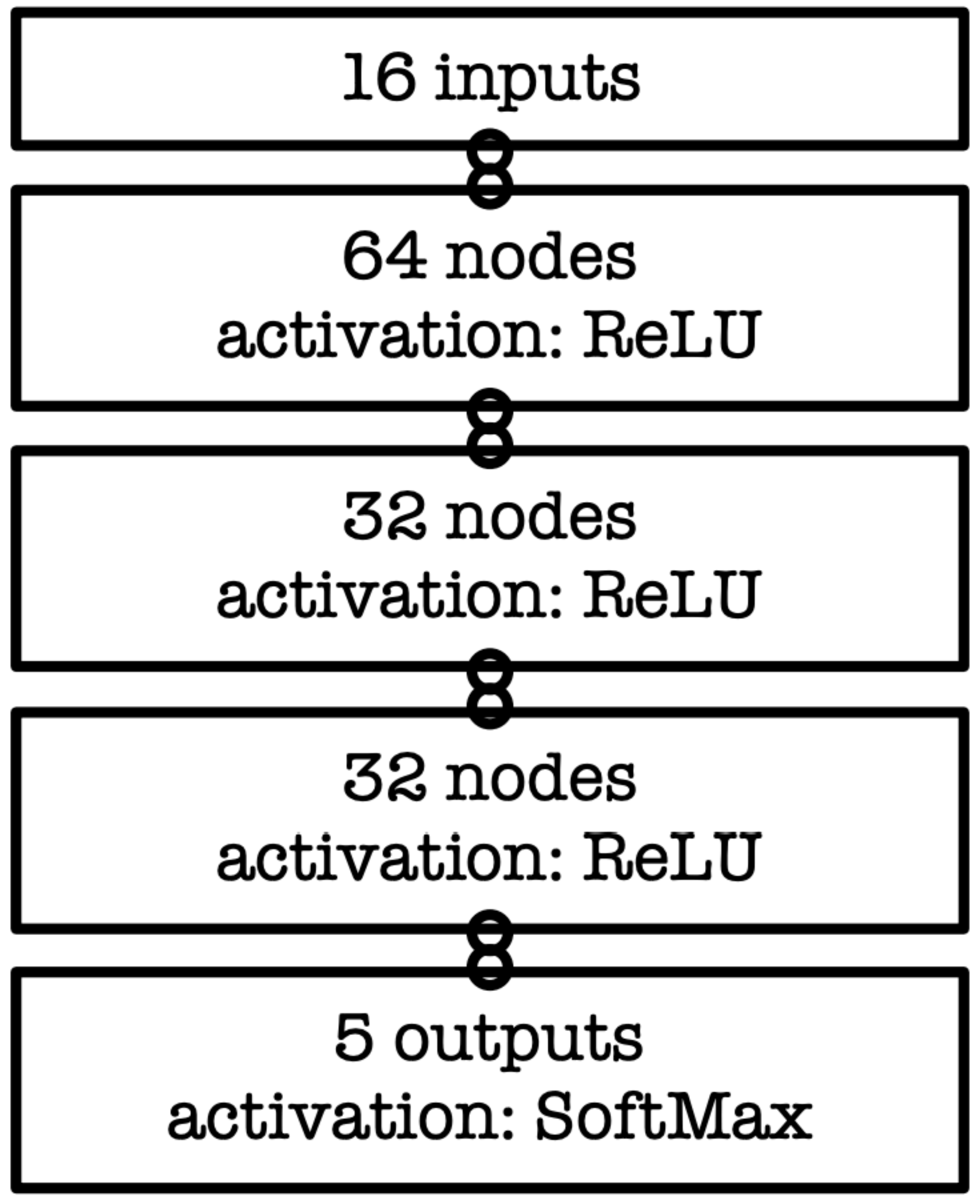
Other quantization techniques:
- ***HAWQ: Hessian AWare Quantization***
- ***Quantization Aware Pruning (B. Hawks et al.)***

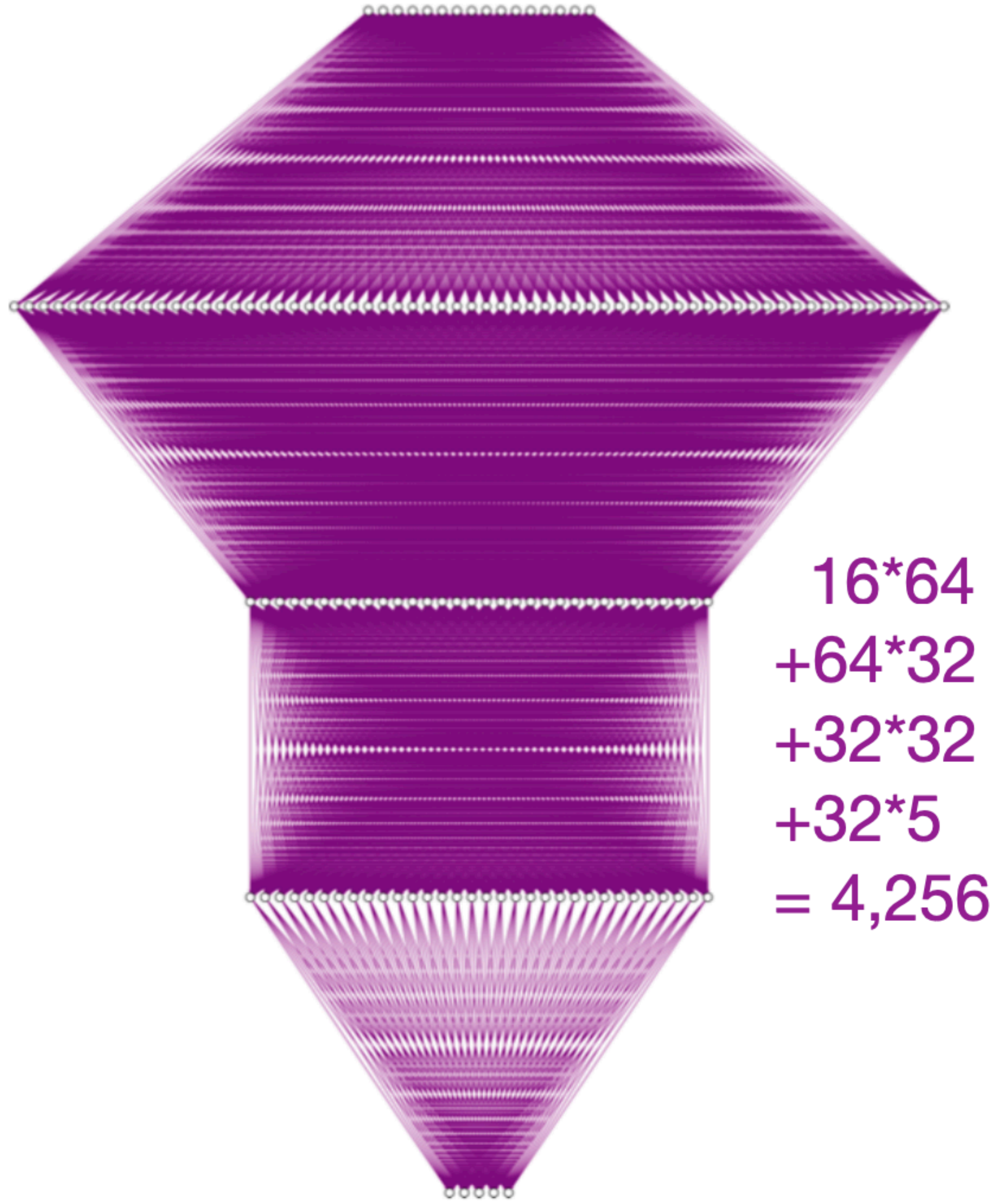
hls4ml collaborate with Xilinx Research Labs to develop QOONX
- Introducing 'Quant' node to ONNX graph
- Brevitas (PyTorch) and QKeras (Keras) can export QONNX (HAWQ export in progress): then hls4ml and FINN can import QONNX

Quantized ONNX (QONNX), J. Mitrevski et. al



Compression





16*64
+64*32
+32*32
+32*5
= 4,256

$$\mathbf{x}_n = g_n(\mathbf{W}_{n,n-1}\mathbf{x}_{n-1} + \mathbf{b}_n)$$

activation function — precomputed and stored in BRAMs

multiplication — DSPs

addition — logic cells

Network size limited by N multiplications!

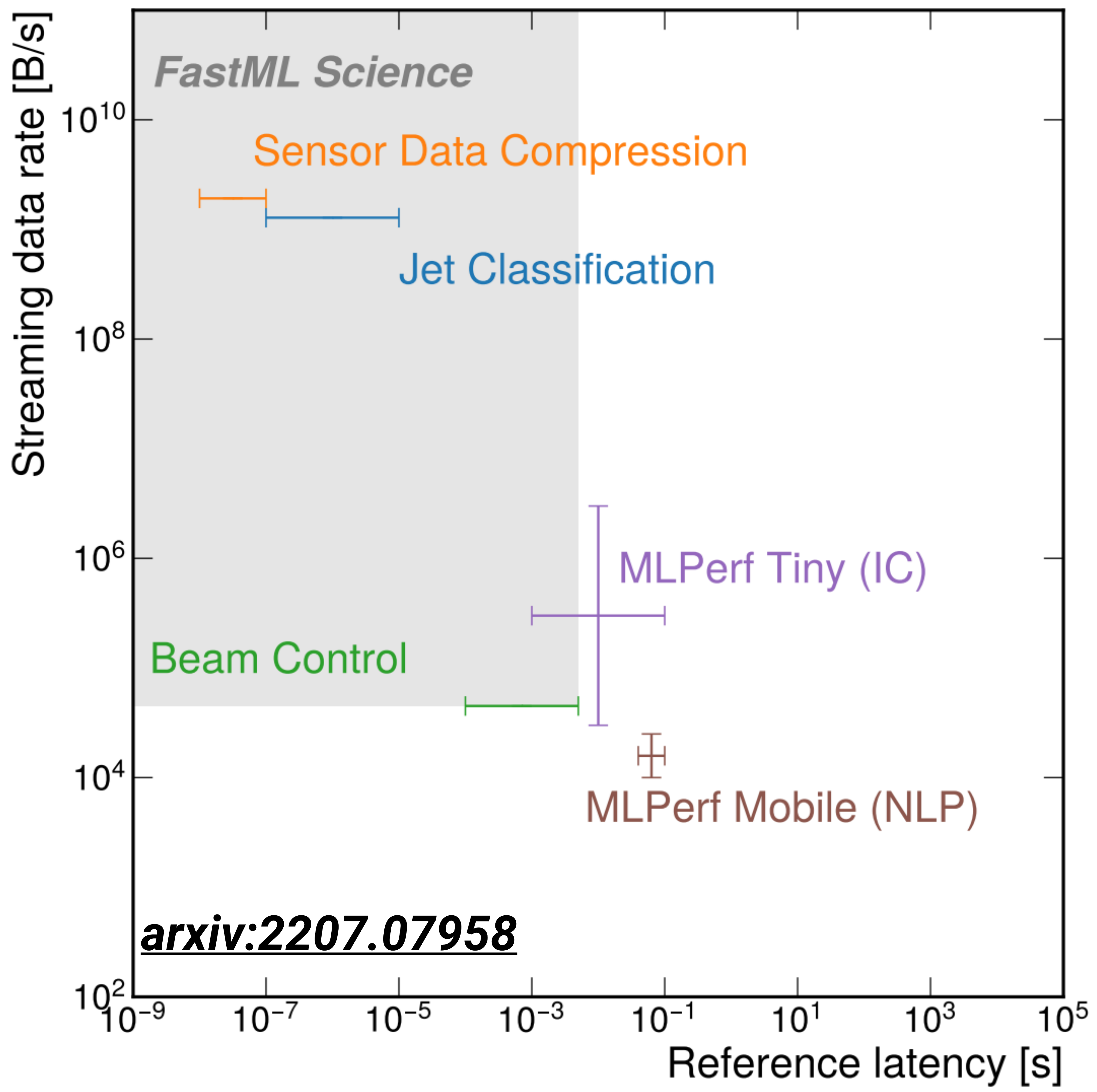
- E.g, simple DNN, 4256 multiplications!
- Typical FPGA at LHC usually has 4-6000 DSPs

Benchmarking

Datasets: Common FastML Science Benchmarking datasets
- guide design of edge ML hardware and software for sub-microsecond inference!

Algorithms: hls4ml-FINN benchmarked in MLPerf™
- how fast systems can process inputs and produce results
- demonstrate efficient and low-latency solutions on FPGAs with hls4ml and FINN

Consistently competitive (QKeras+hls4ml, semantic segmentation, MLPerf)



arxiv:2207.07958

https://mlcommons.org/en/inference-tiny-07/

Model	LUT		LUTRAM		FF		BRAM [36 kb]		DSP		Latency [ms]	Energy/inf. [μJ]
Pynq-Z2												
Available	53 200		17 400		106 400		140		220		–	–
IC (hls4ml)	28 544	53.7%	3 756	21.6%	49 215	46.3%	42	30.0%	4	1.8%	27.3	44 330
IC (FINN)	24 502	46.1%	2 086	12.0%	34 354	32.3%	100	71.4%	0	0.0%	1.5	2 535
AD	40 658	76.4%	3 659	21.0%	51 879	48.8%	14.5	10.4%	205	93.2%	0.019	30.1
KWS	33 732	63.4%	1 033	5.9%	34 405	32.3%	37	26.4%	1	0.5%	0.017	30.9

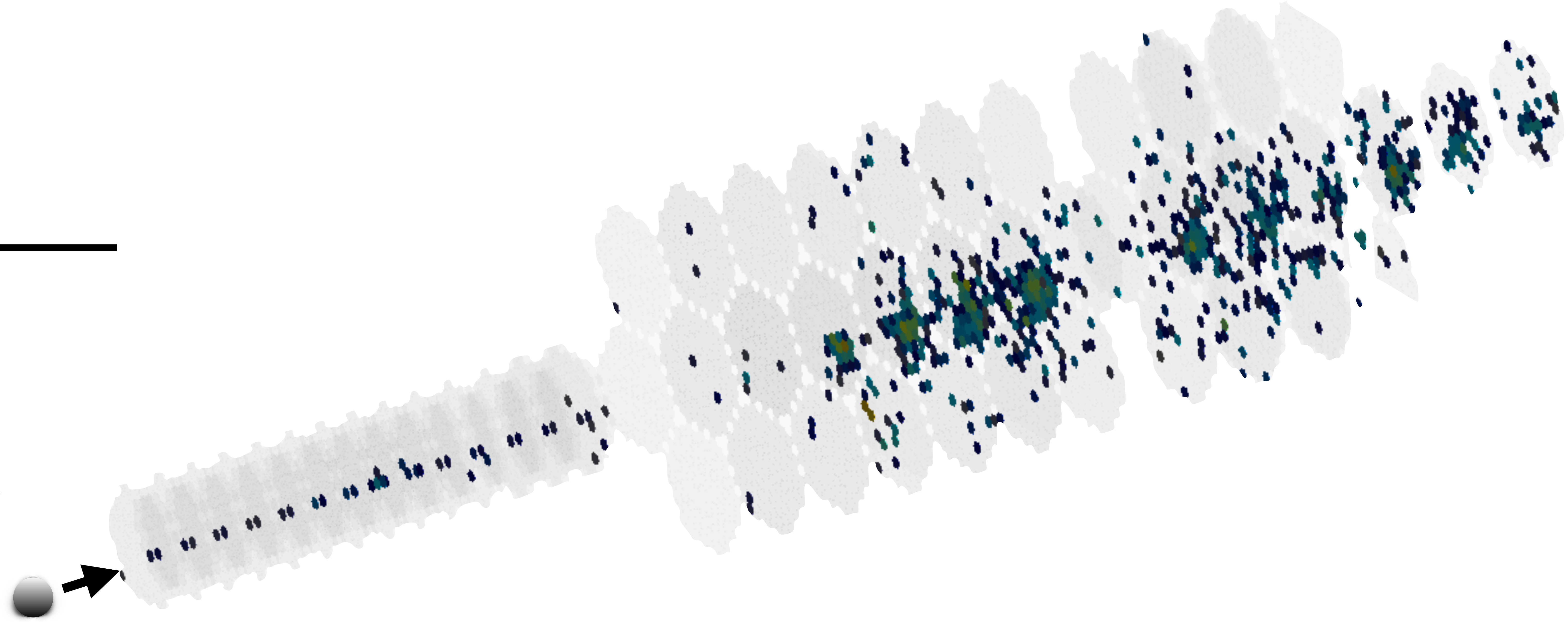
arxiv:2103.05579

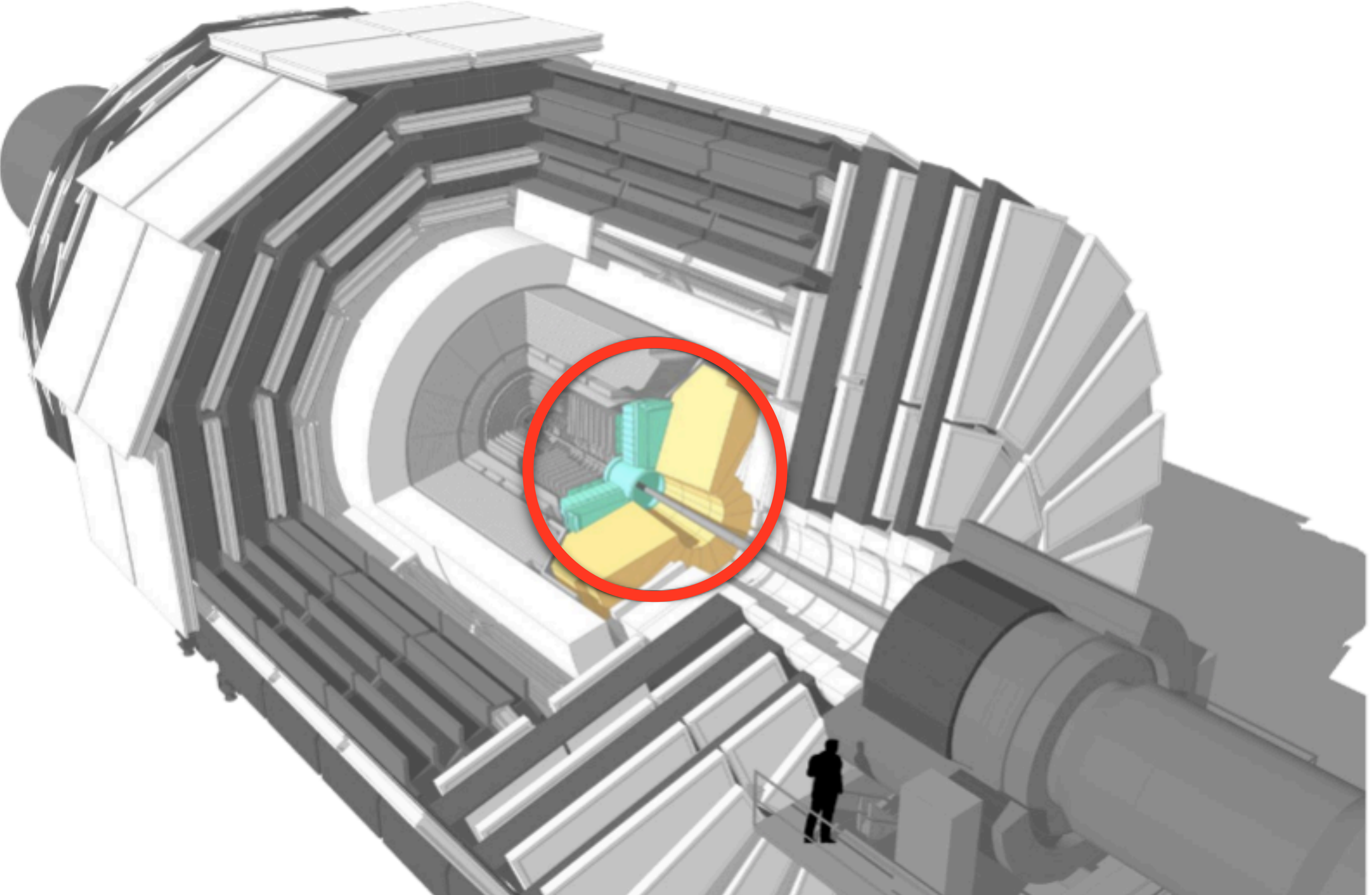
ML for compression

On-detector ML

CMS High Granularity calorimeter

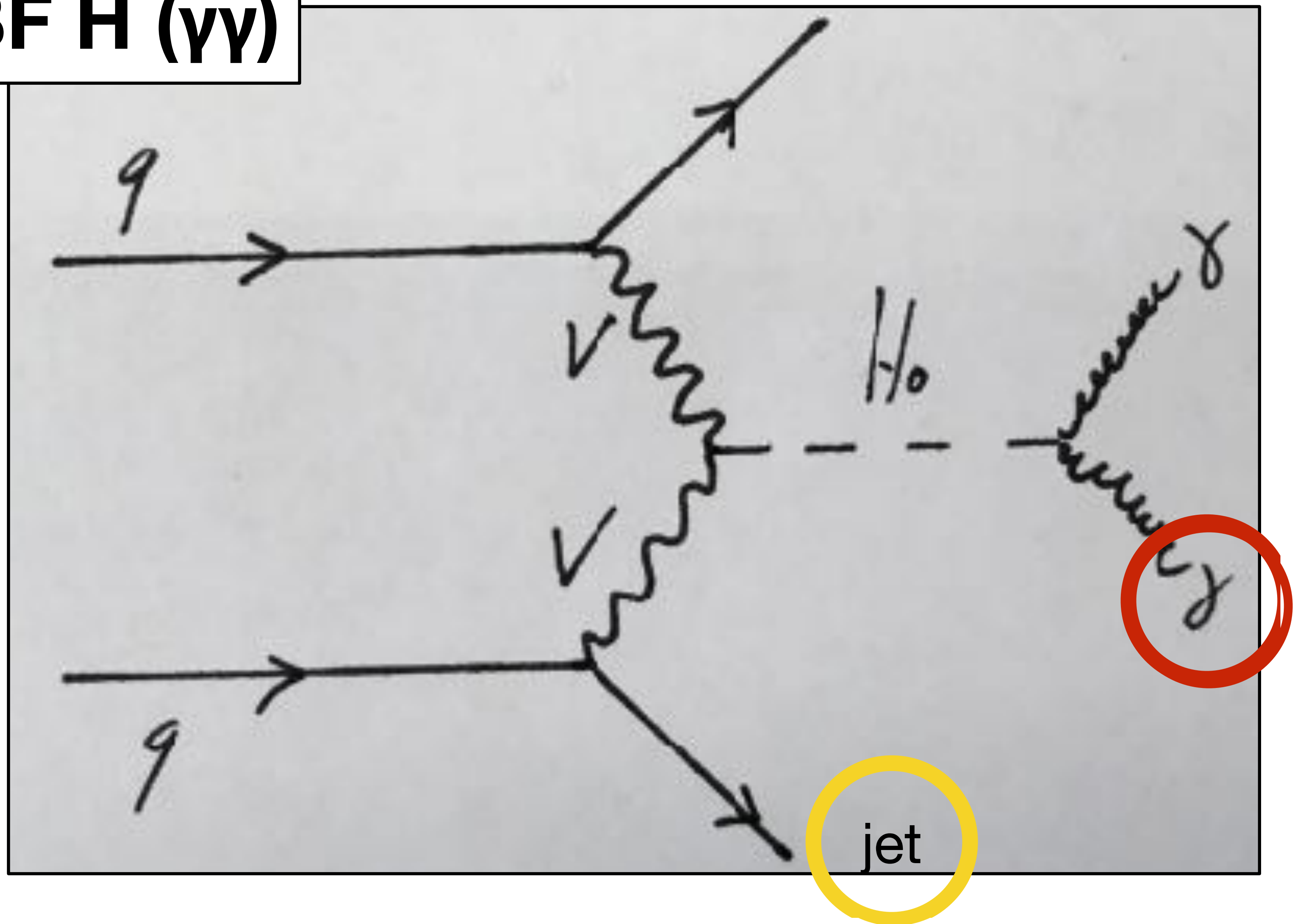
- 6.5 million readout channels, 50 layers



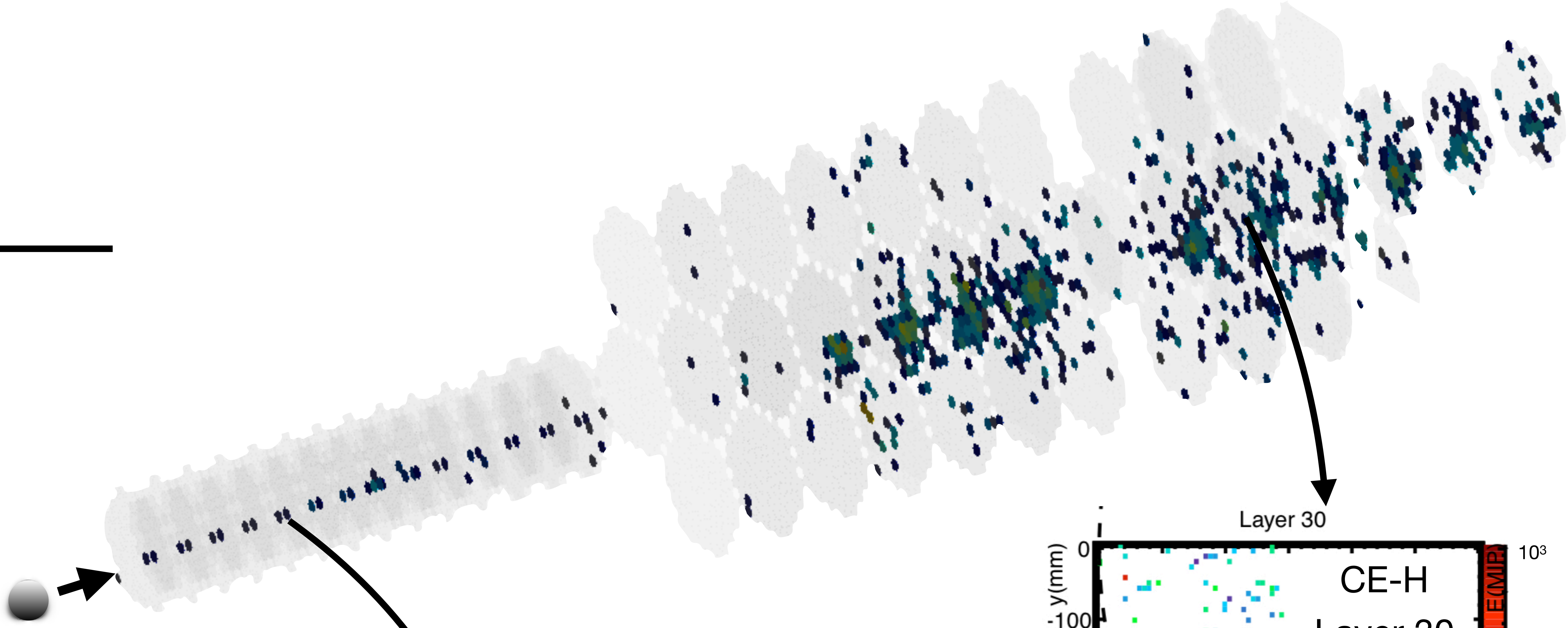


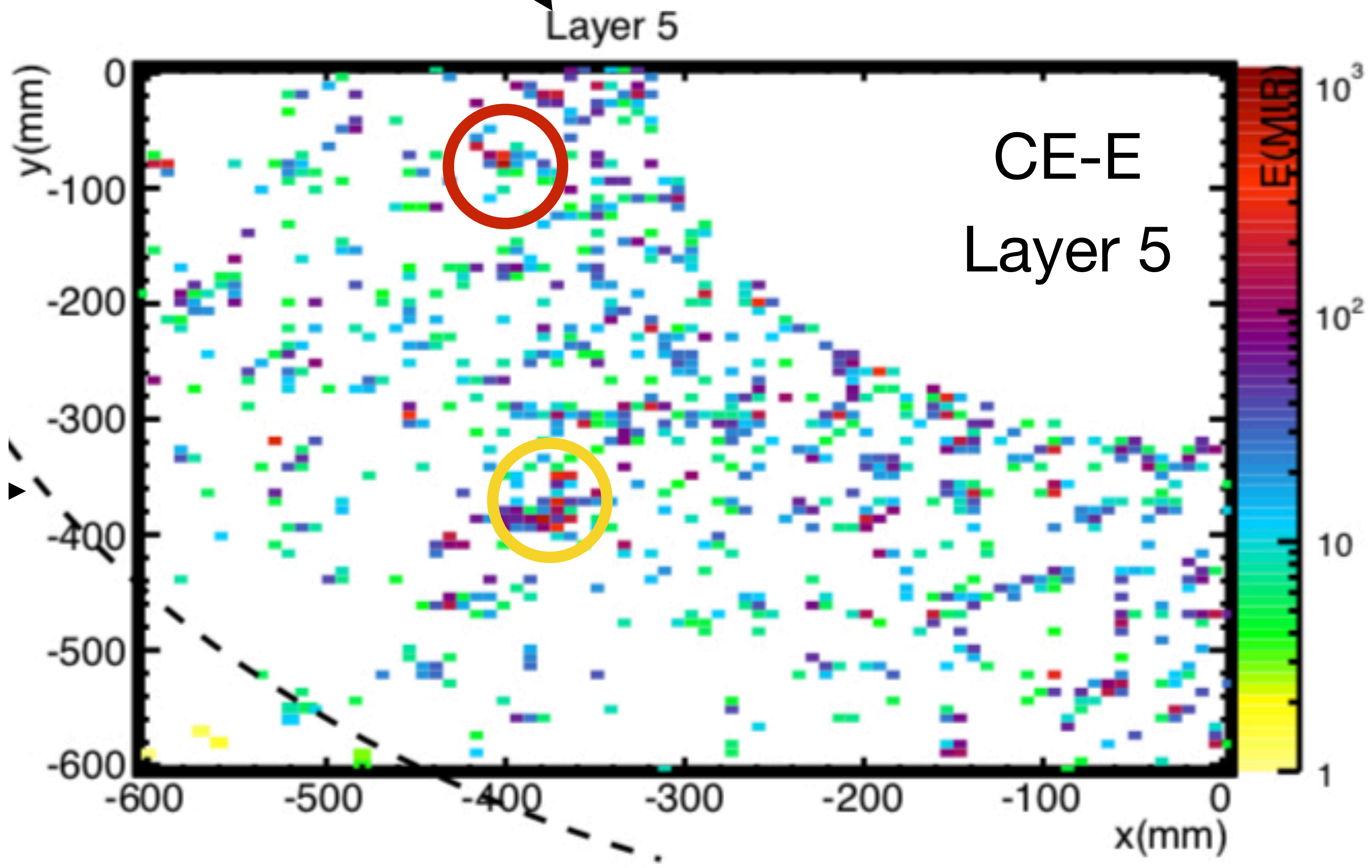
On-detector ML
AI for compression

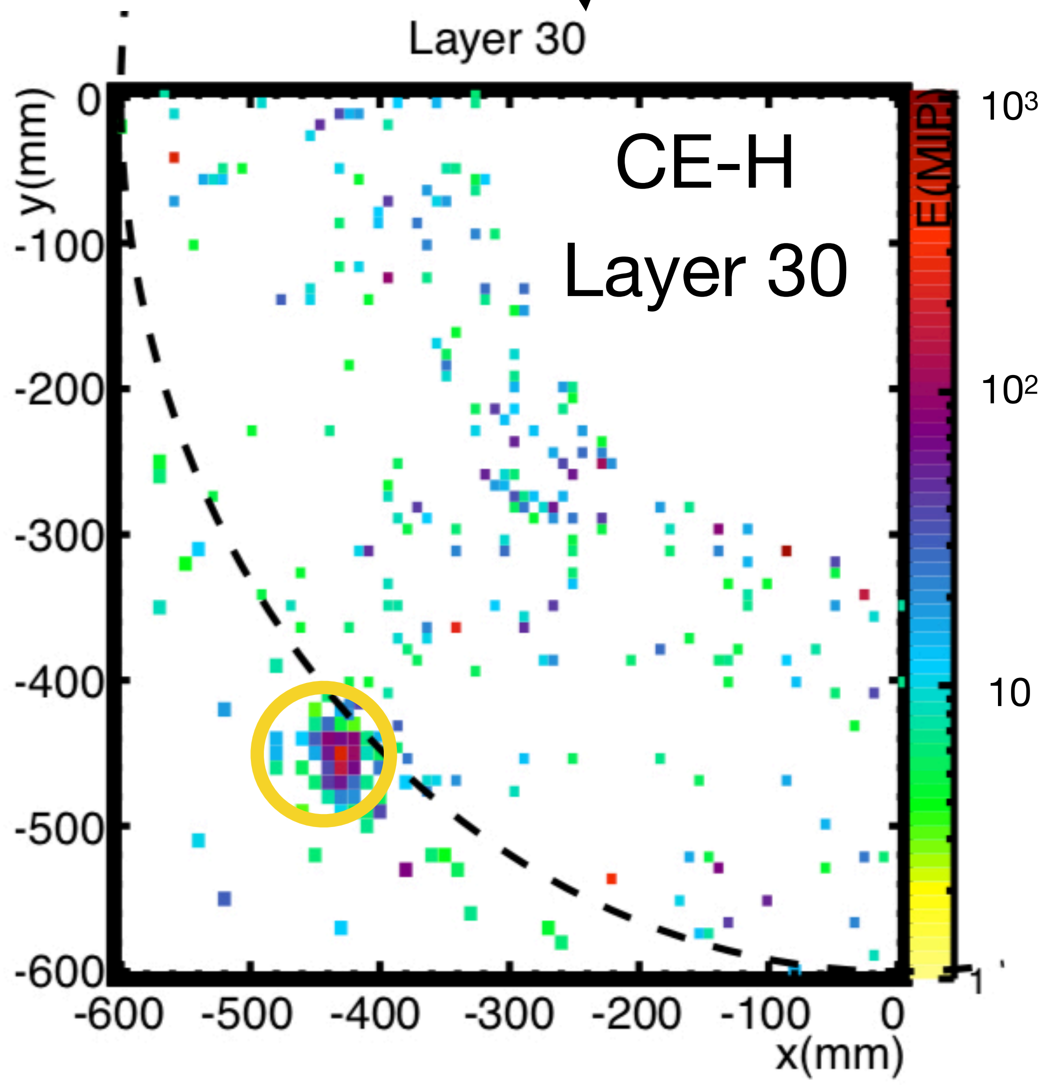
VBF H (γγ)



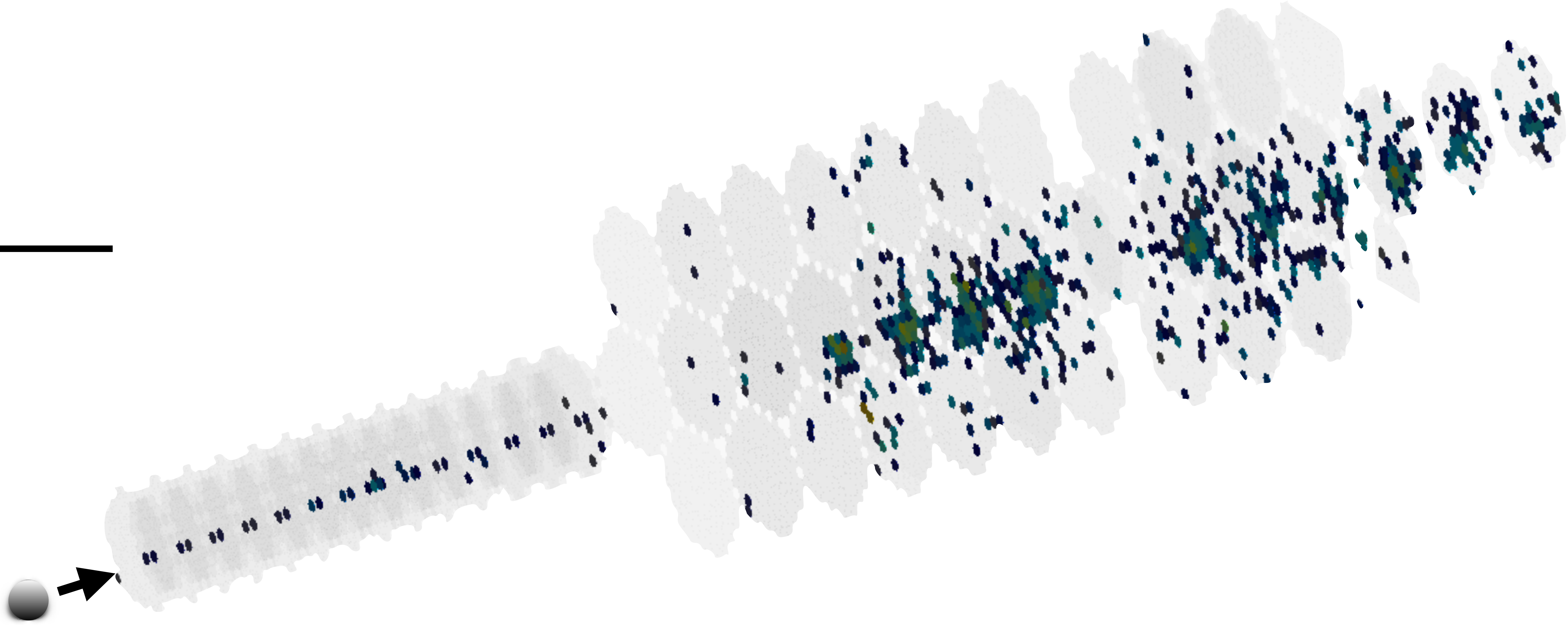
+



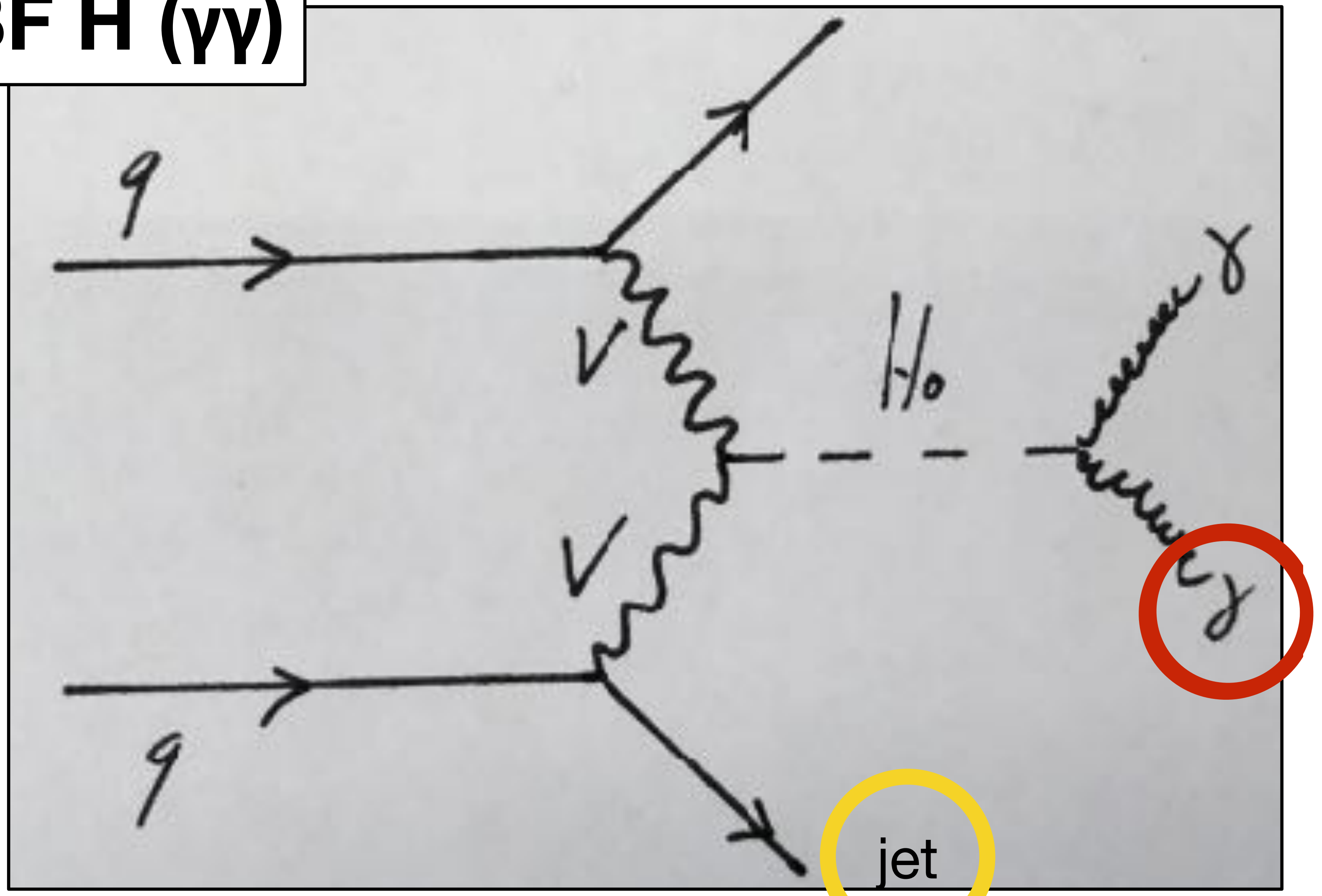




AI for compression / On-detector ML

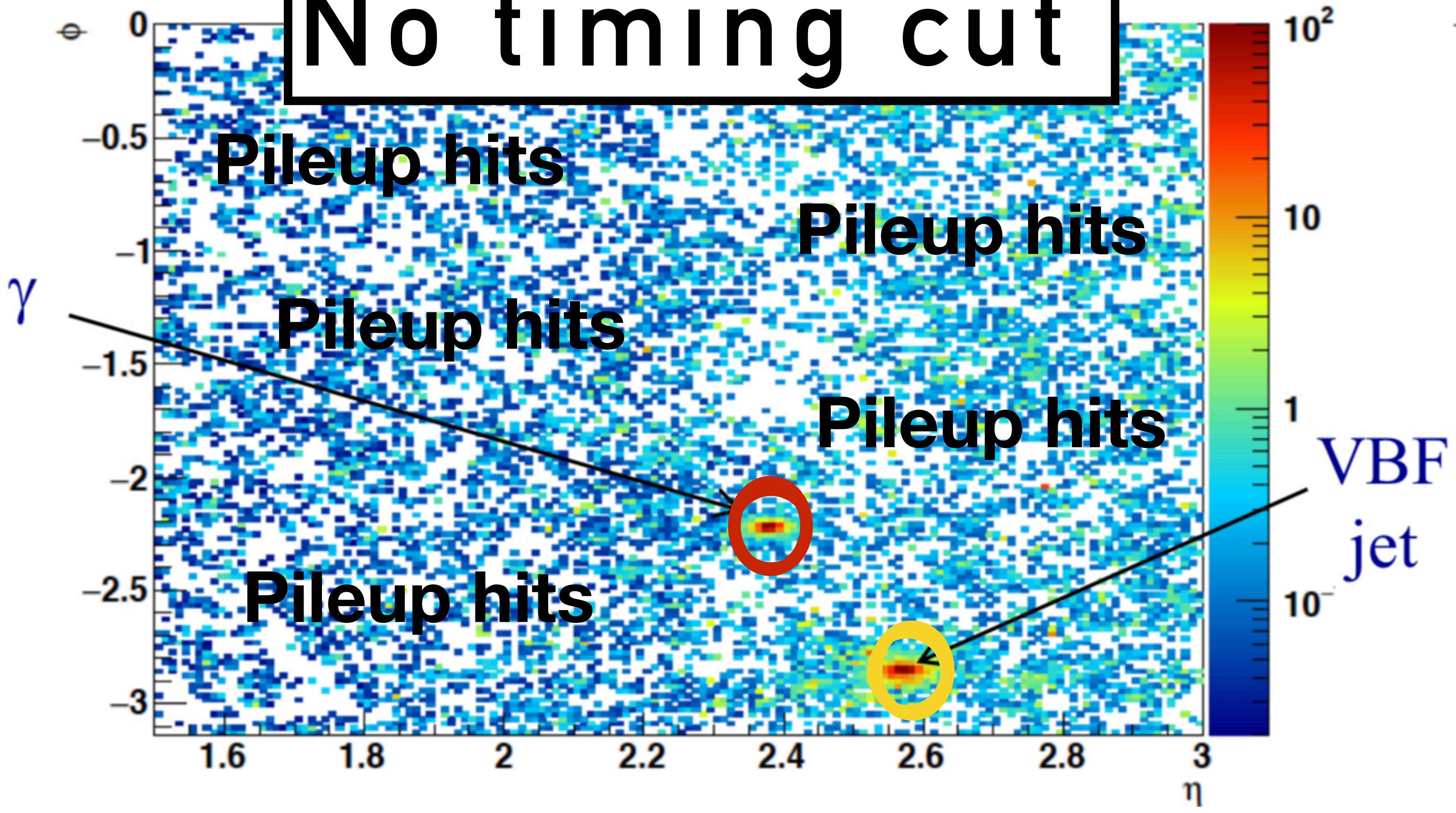


VBF H (γγ)

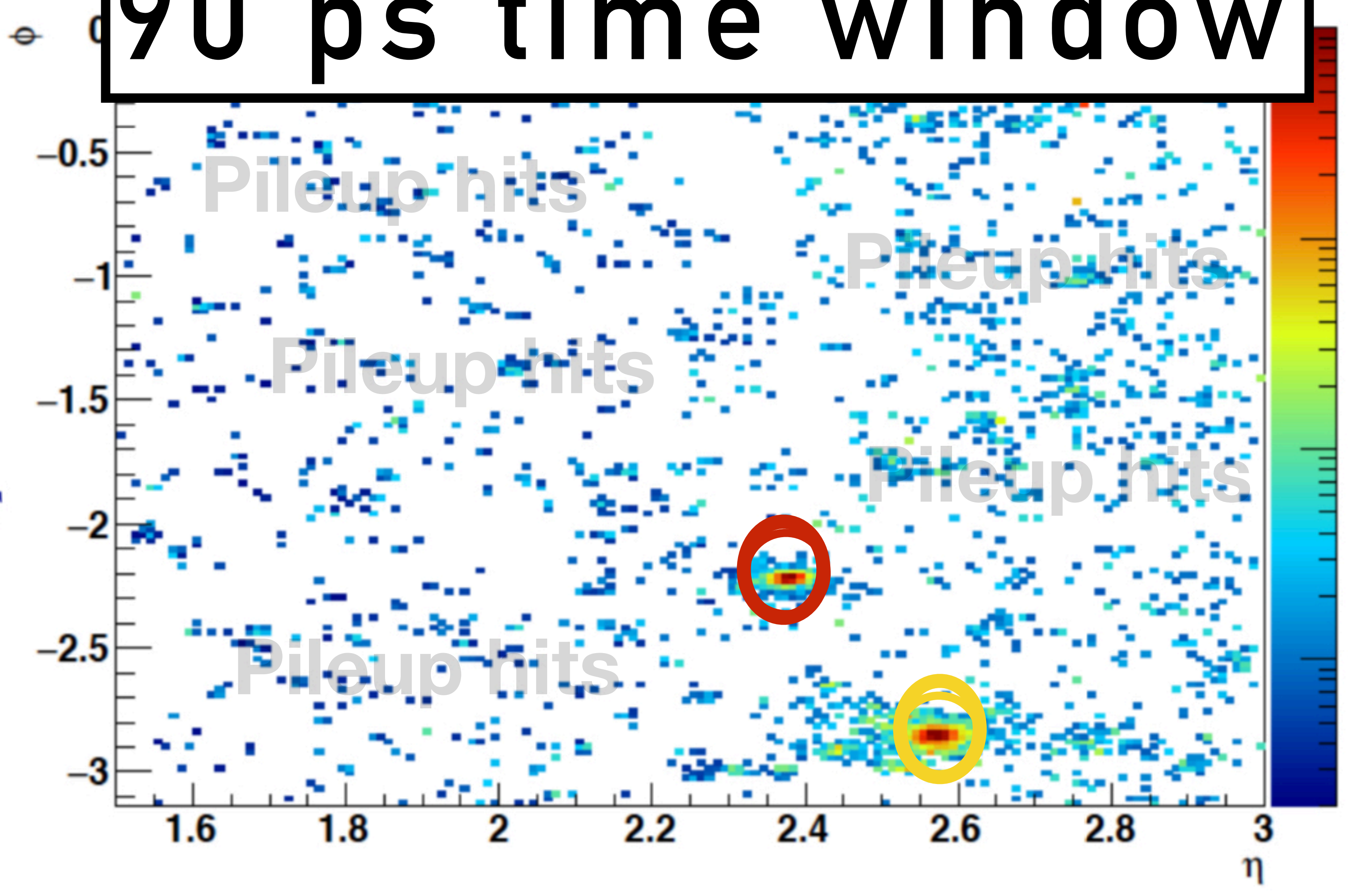


+

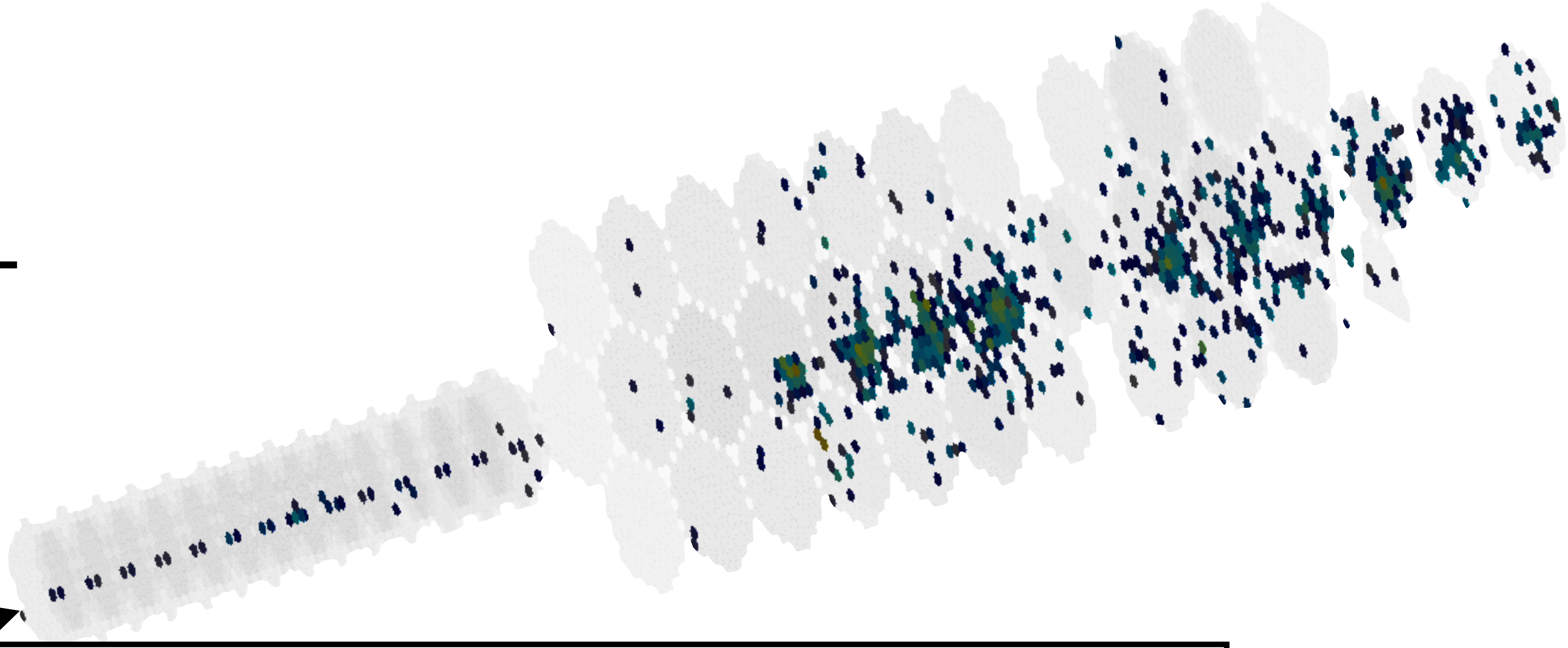
No timing cut



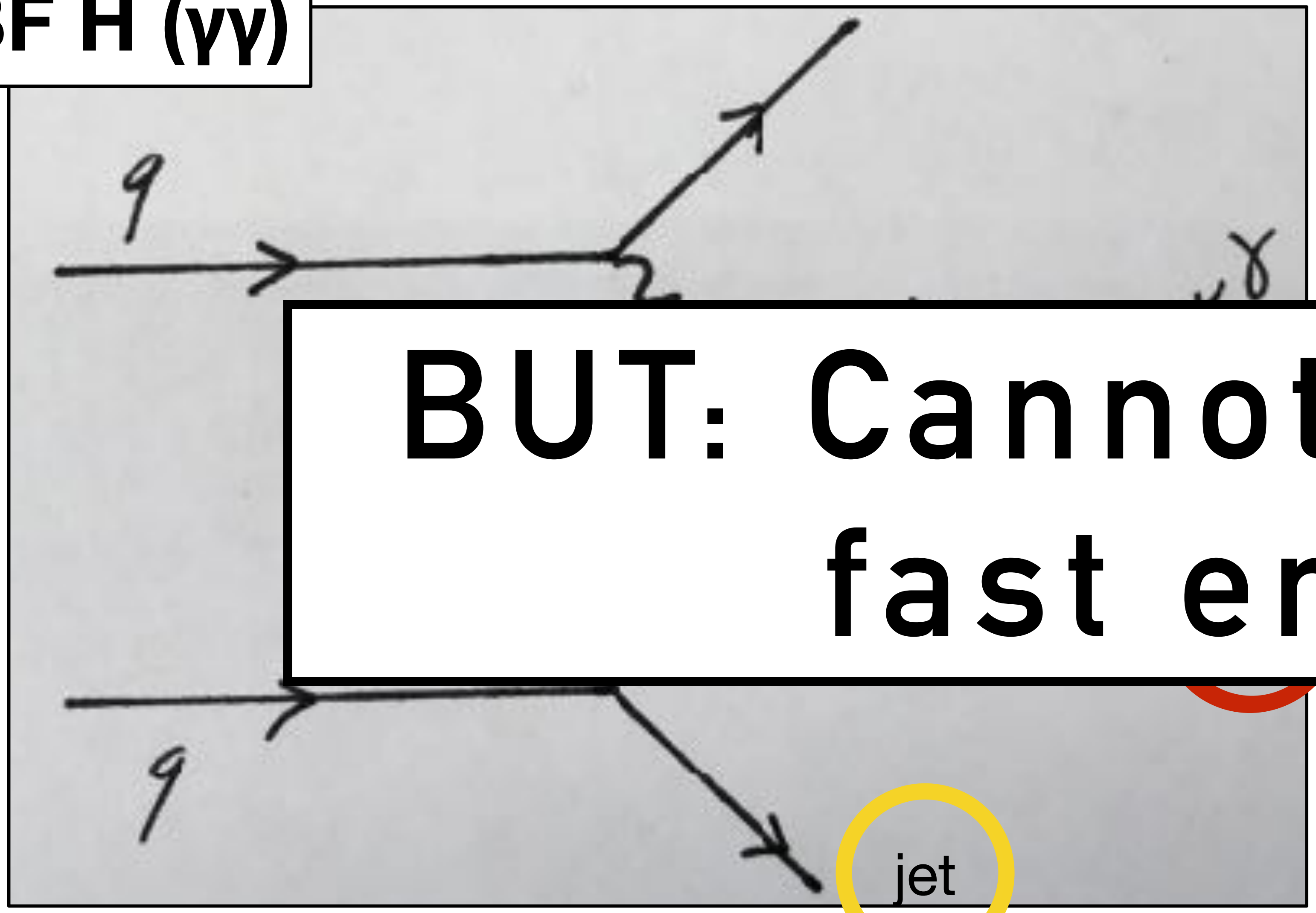
90 ps time window



On-detector ML
AI for compression



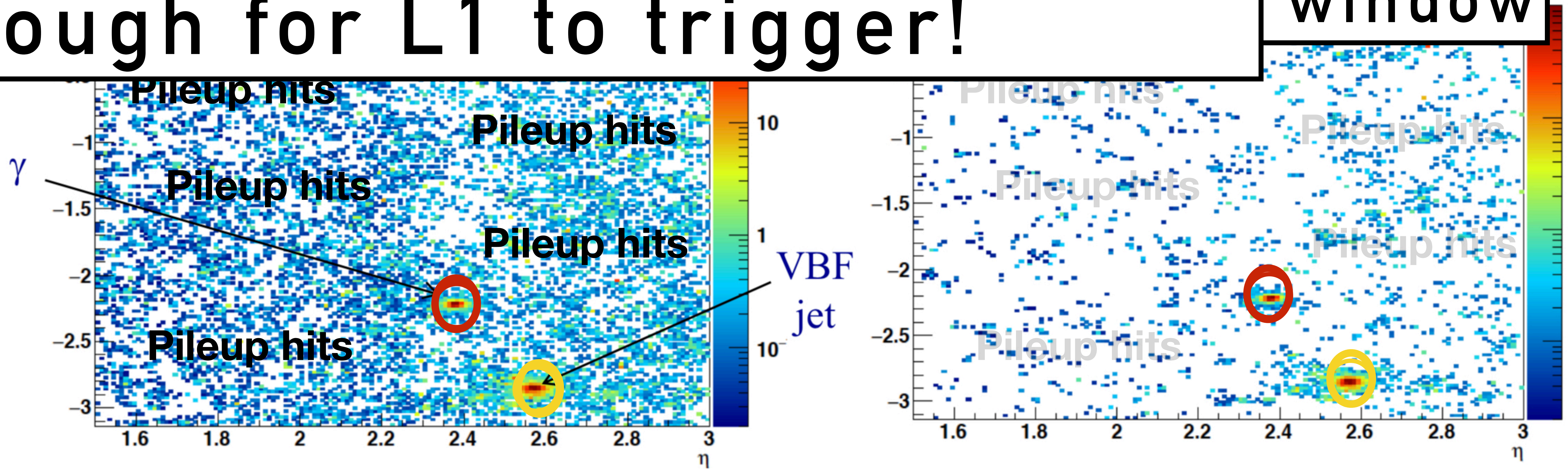
VBF H (γγ)



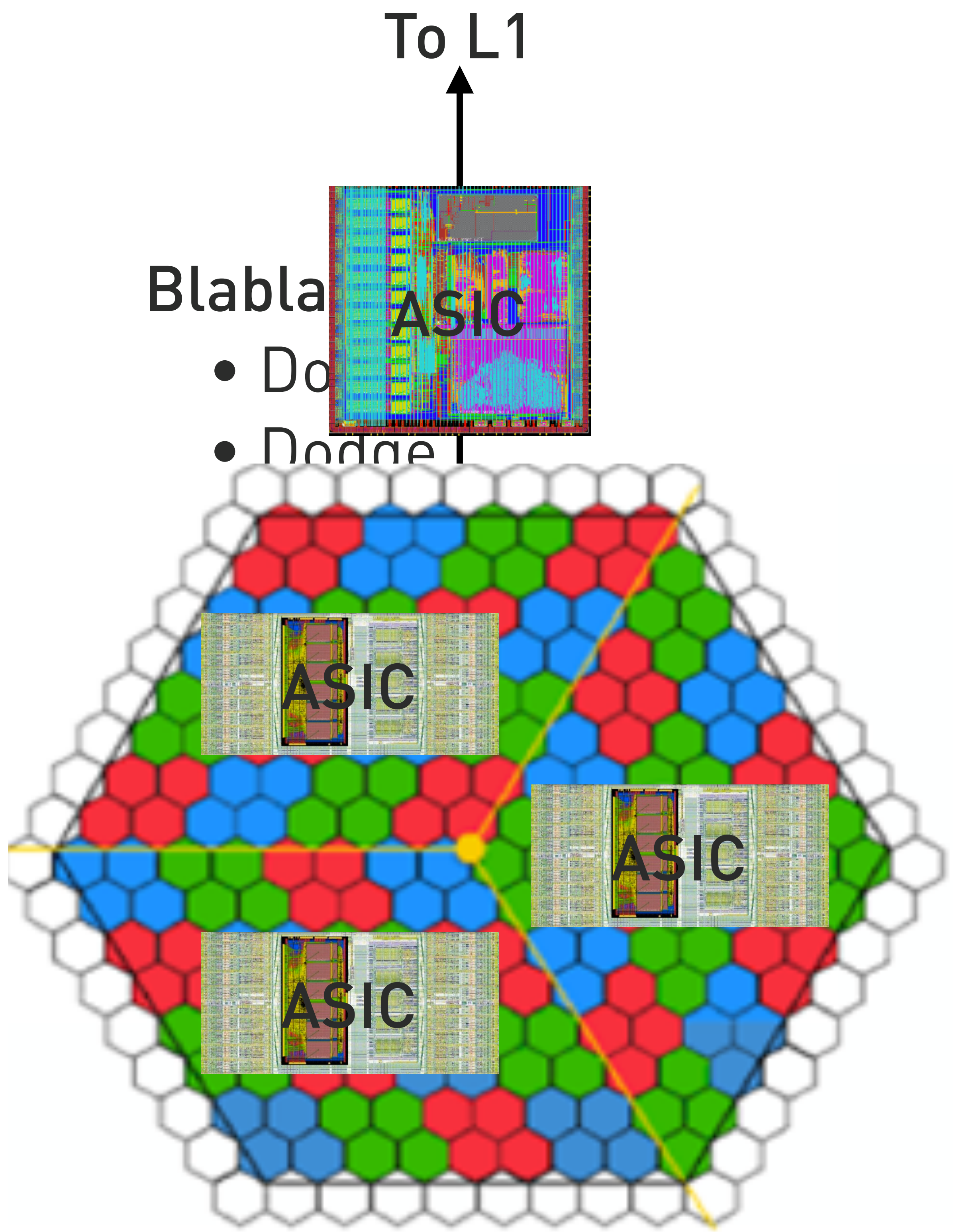
\+

BUT: Cannot read out all these channels fast enough for L1 to trigger!

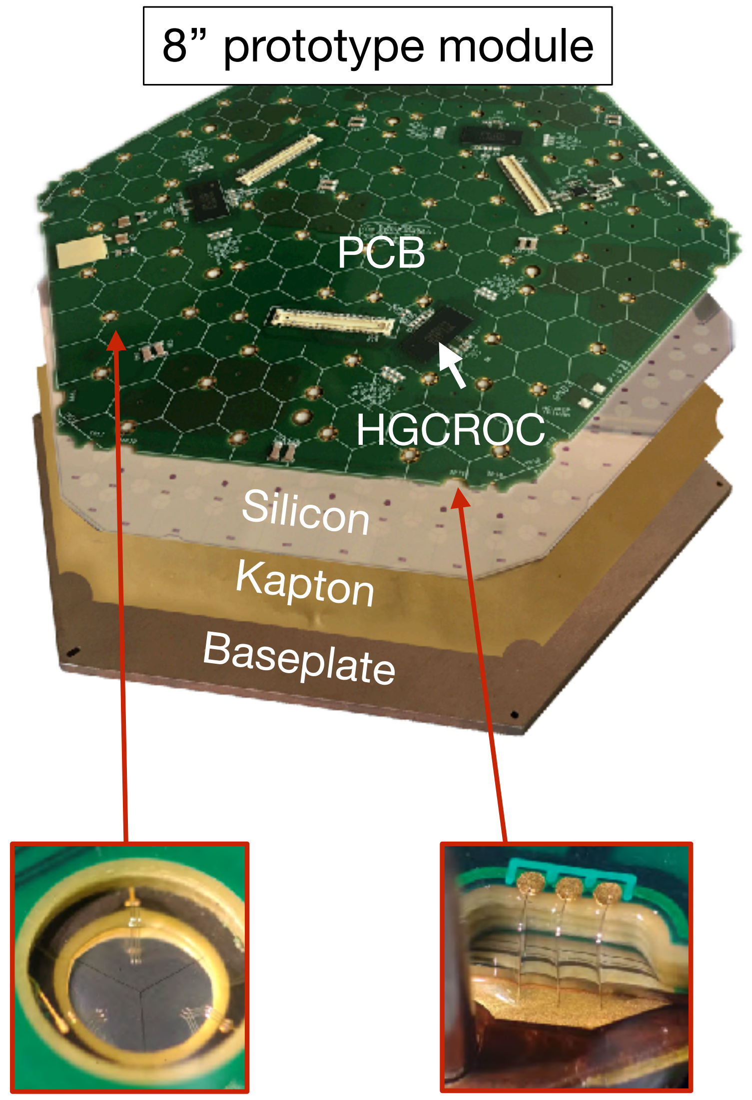
window



On-detector ML



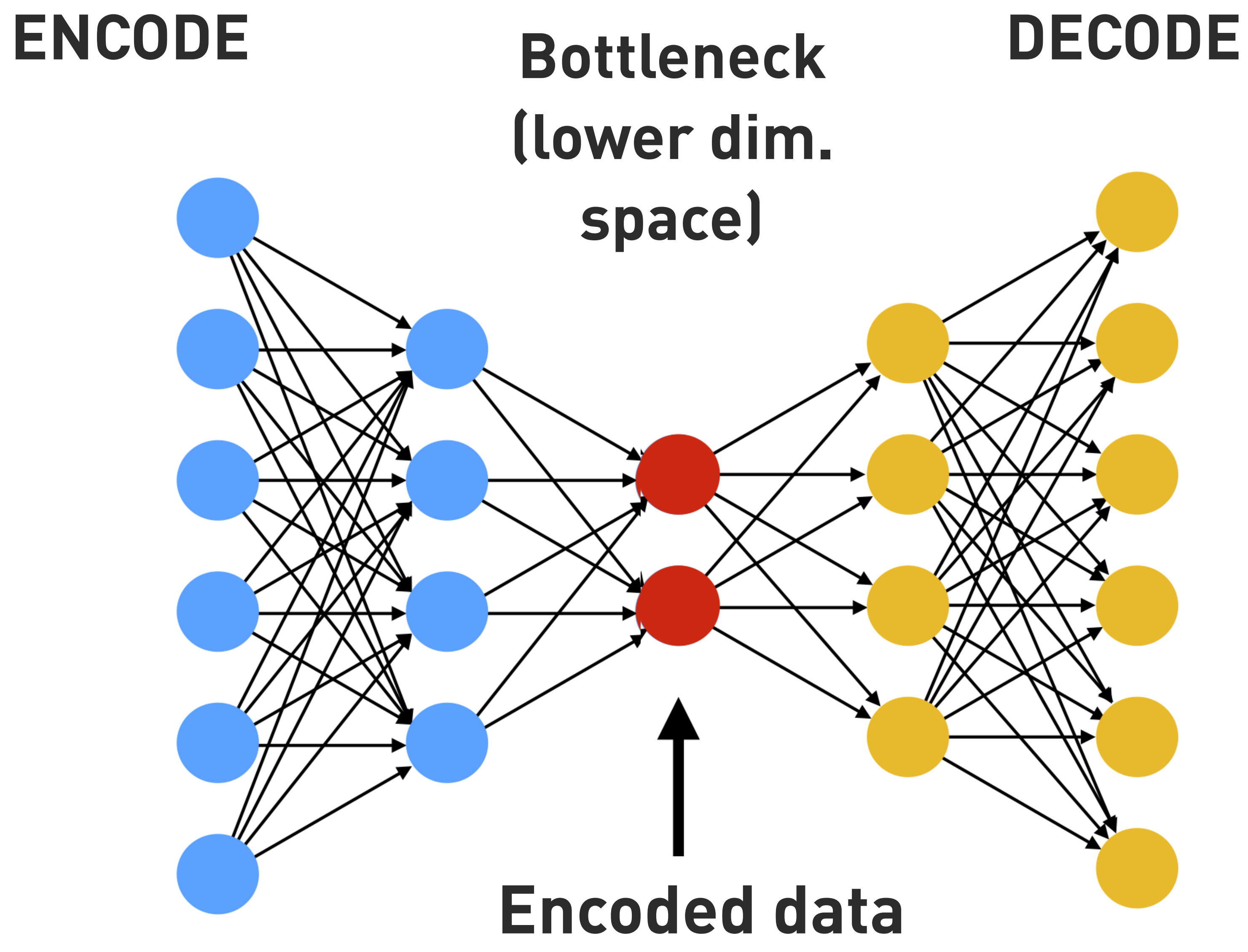
Blabla
- Do
- Dodge



Must compress ON DETECTOR
- High radiation
- Cooled to -30 → low power
- 1.5 μs latency

AI for compression
On-detector ML

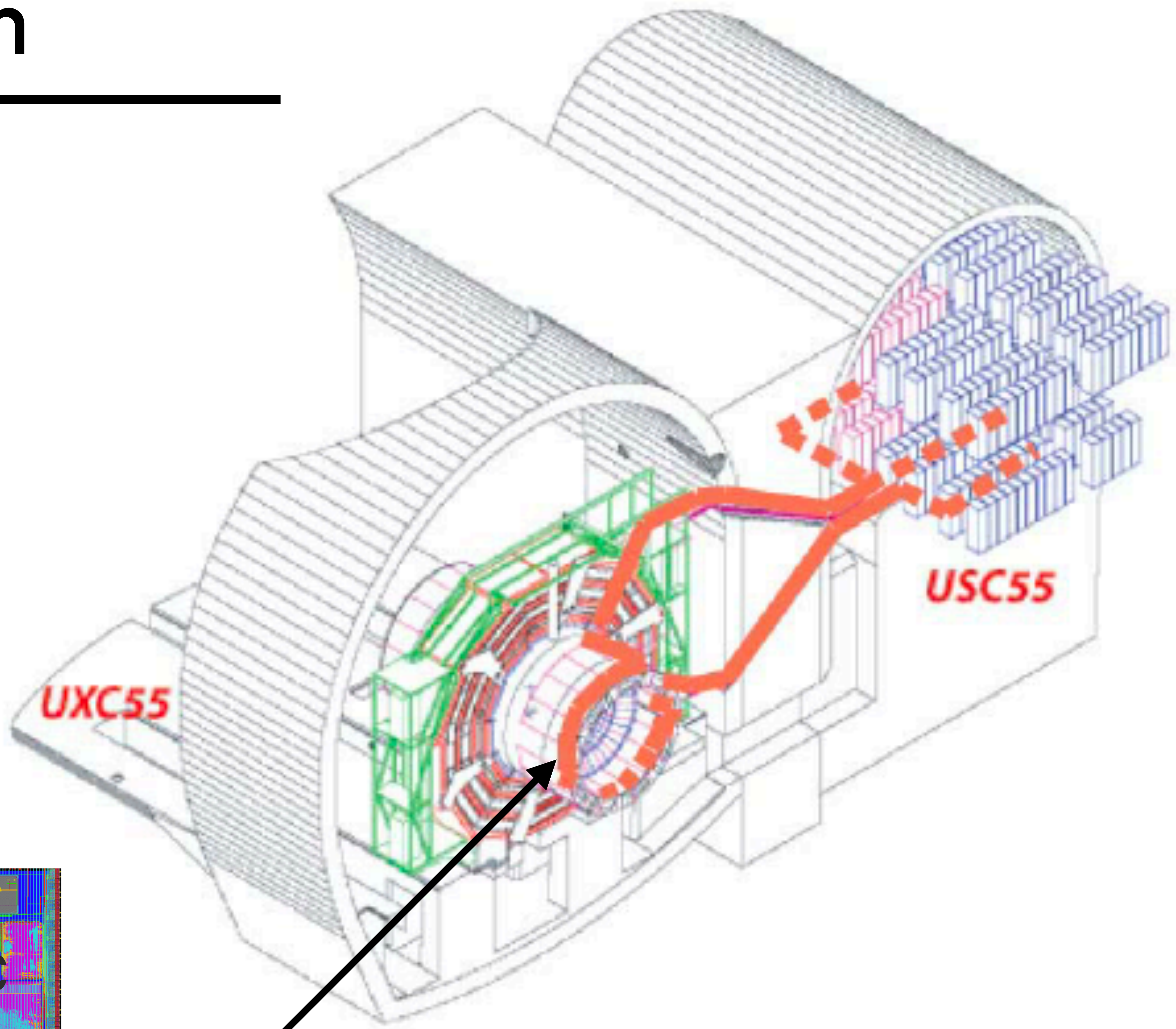
Blabla
- Dodge
- Dodge



Variational Autoencoder

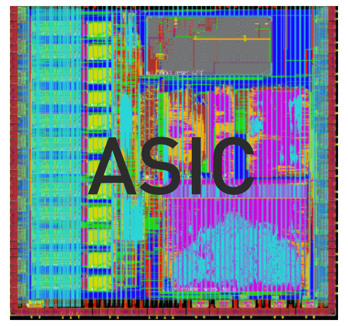
On-detector ML

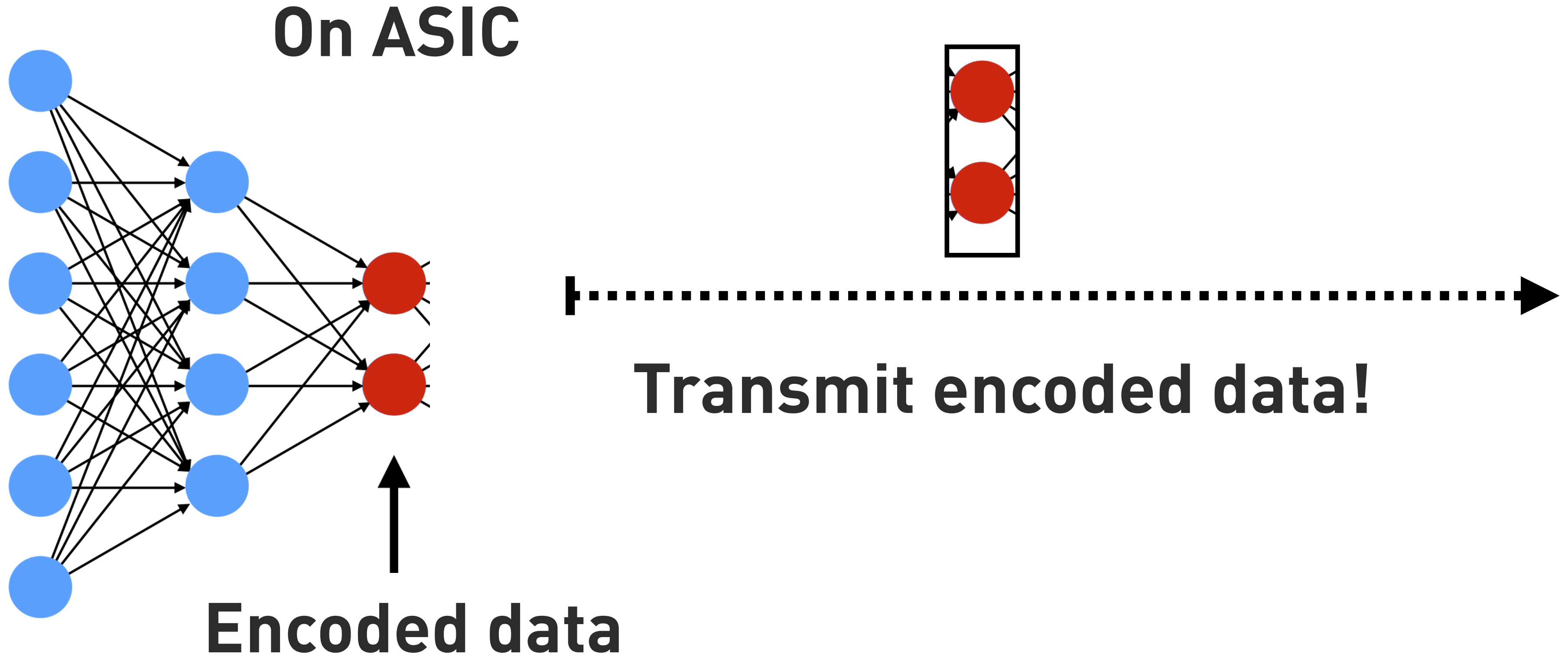
AI for compression



Blabla
- Dodge
- Dodge

Blabla
- Dodge
- Dodge

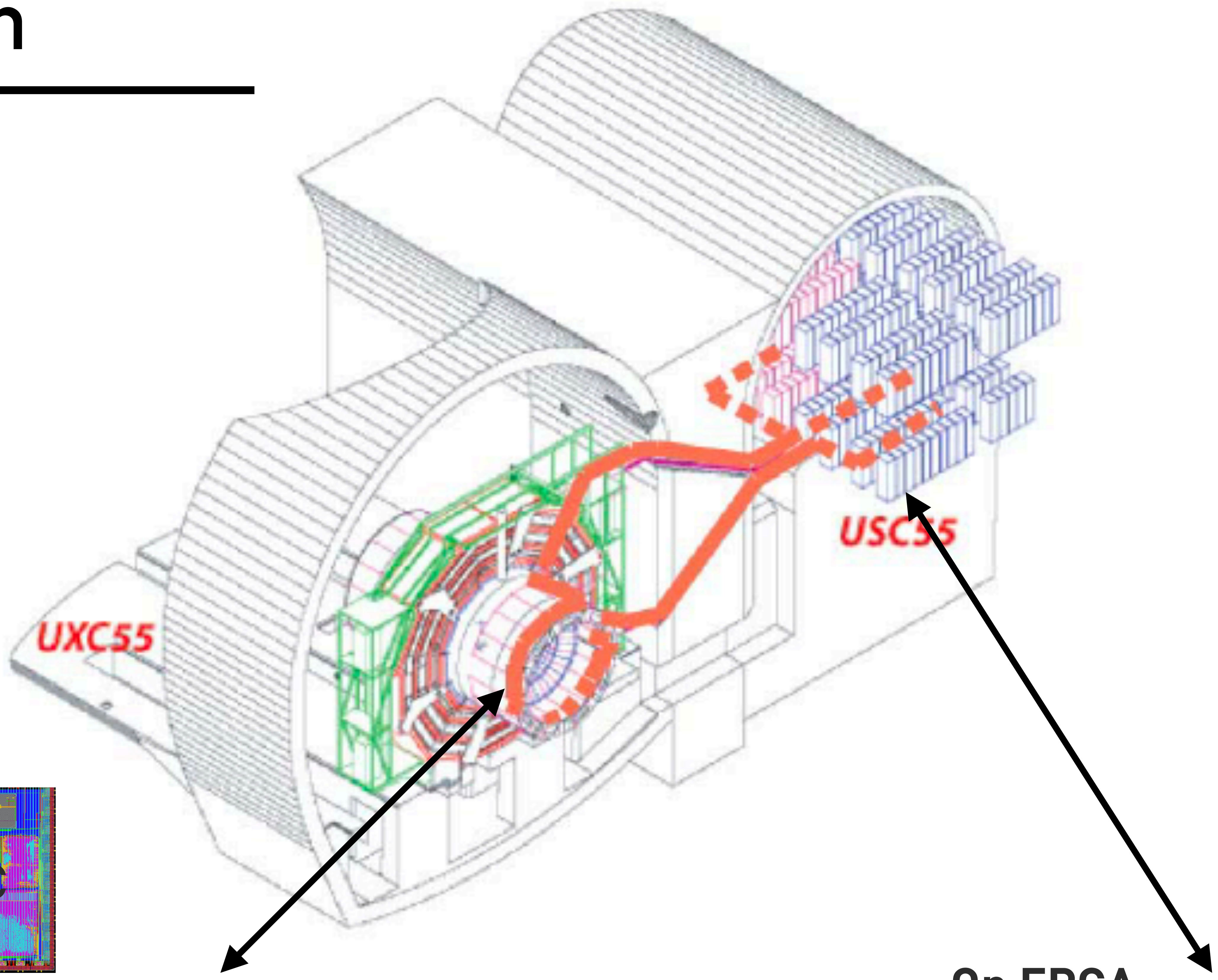




On-detector ML
Ah for compression

Blabla
- Dodge
- Dodge

Blabla
- Dodge
- Dodge



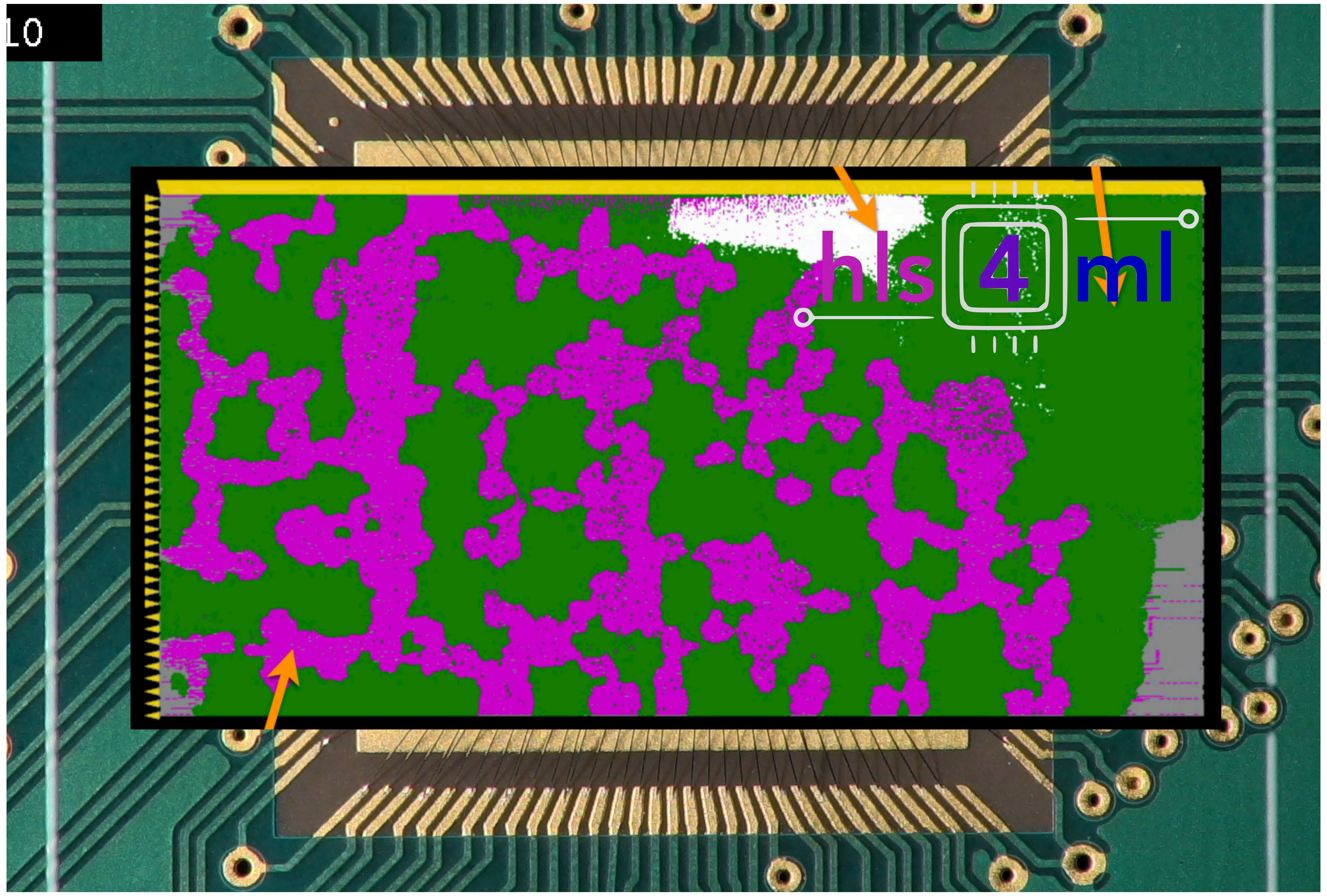
On-detector ML

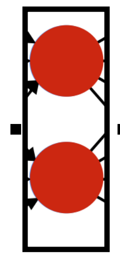
AI for compression

Blabla
- Dodge
- [

Blab
- [
- [

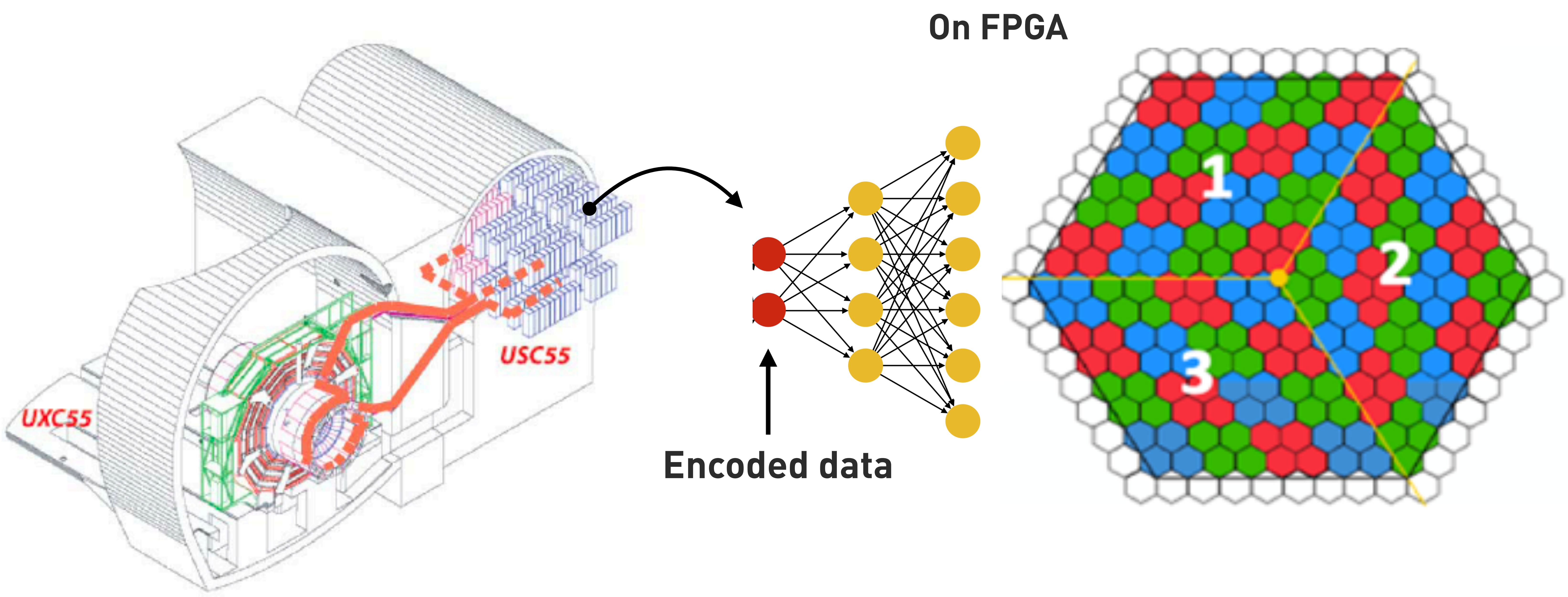
48 trigger cells





16 ReLU activated nodes

ML for reconstruction



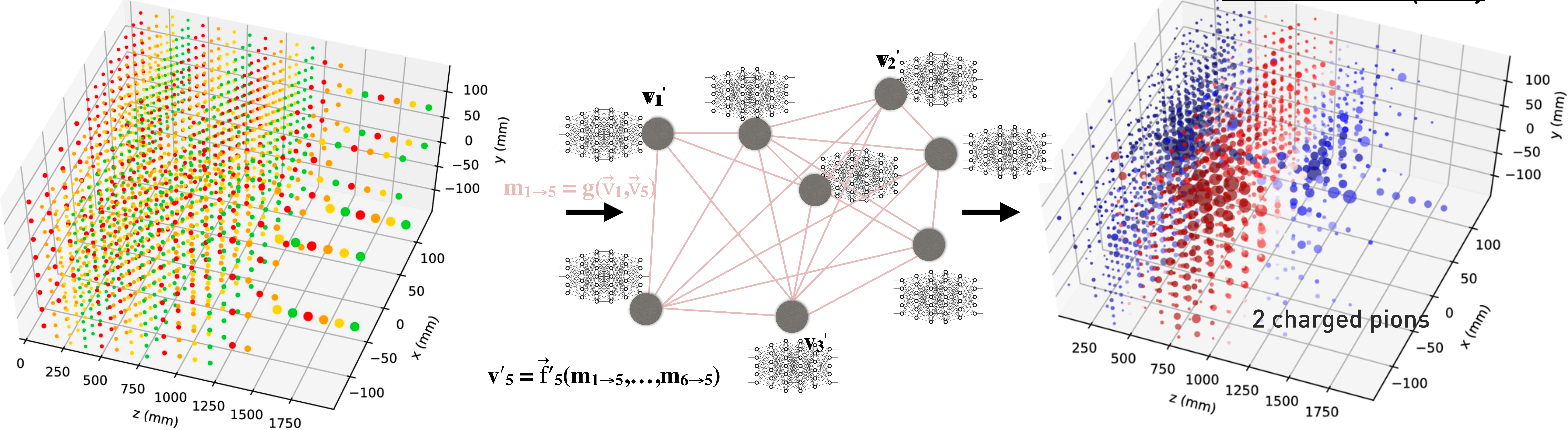
On FPGA: 3.5 μs to cluster energy deposits

ML for reconstruction

On FPGA: 3.5 μs to cluster energy deposits

ML for reconstruction

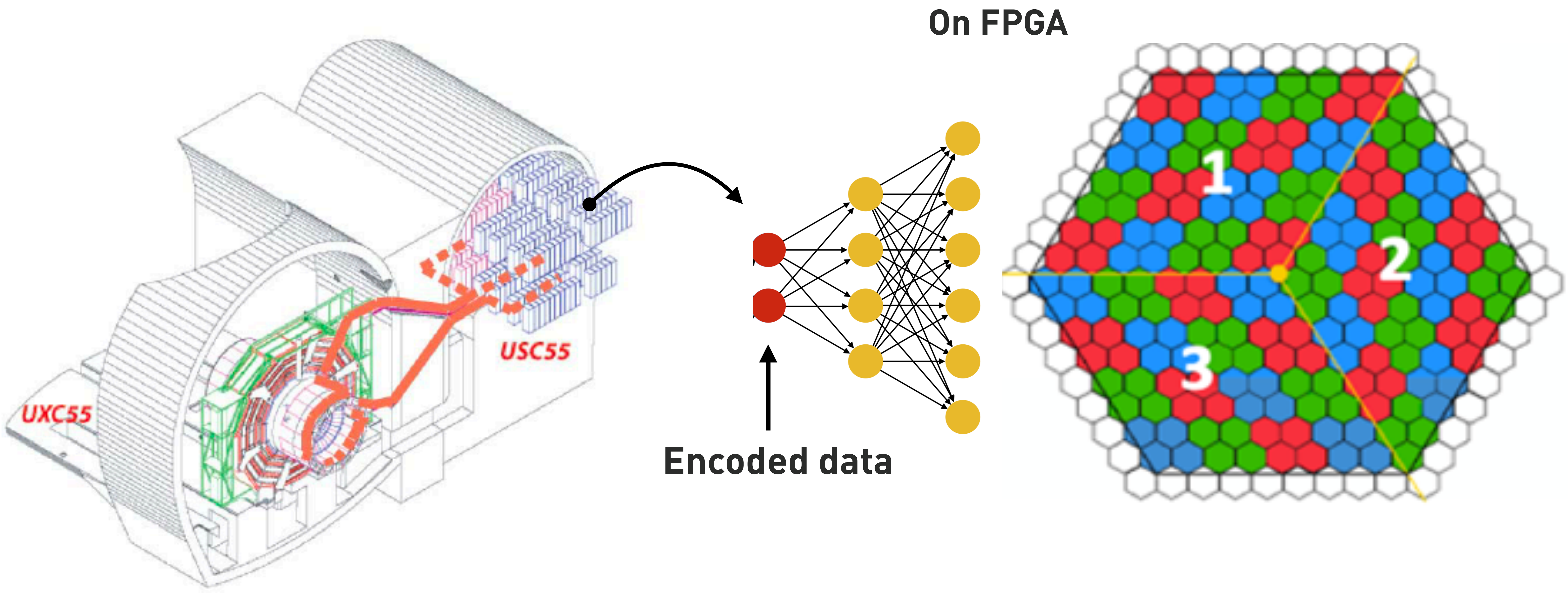
EPJC Vol 79 608 (2019)



On FPGA: 3.5 μs to cluster energy deposits

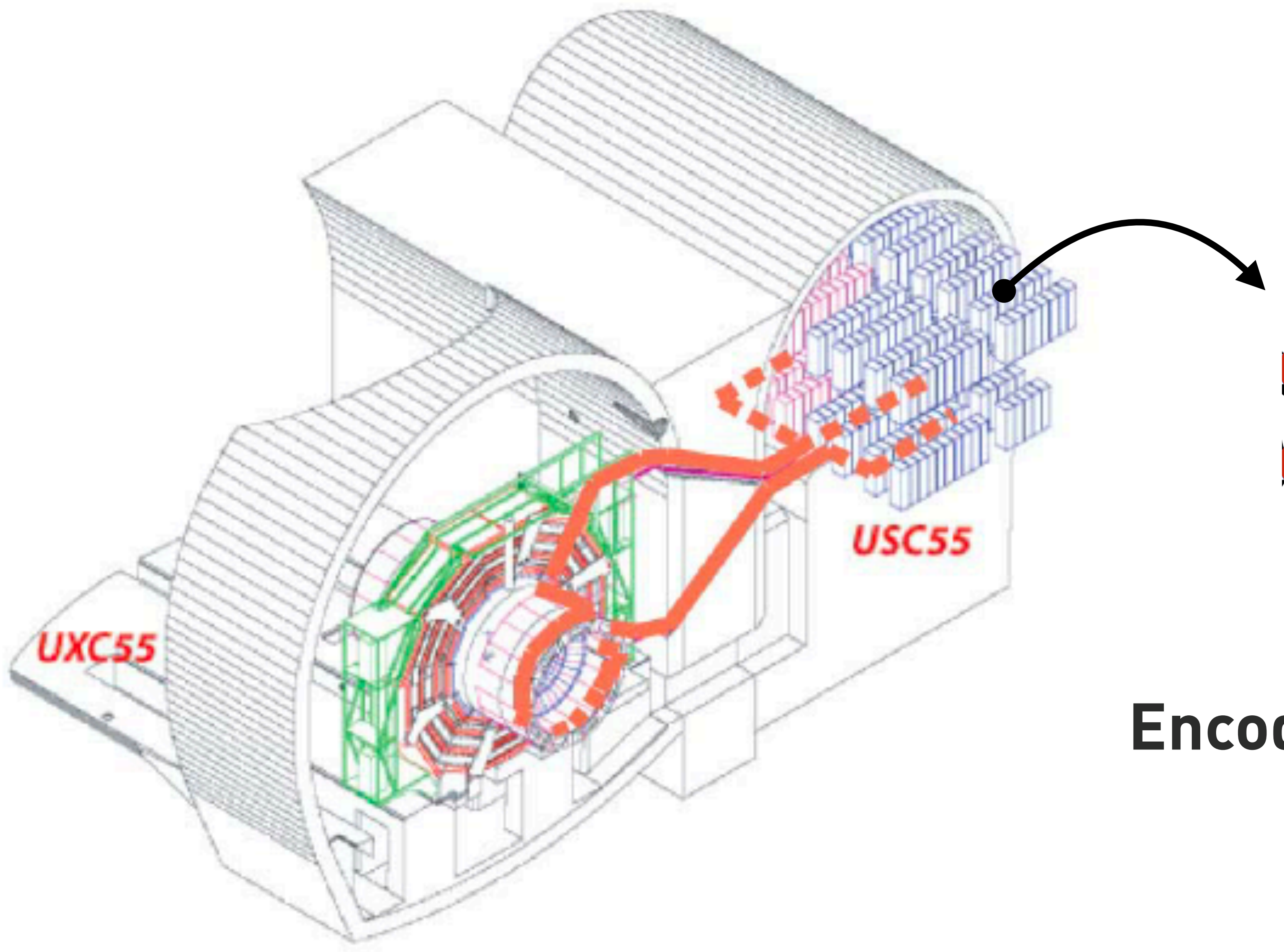
- Graph Neural Networks (GarNet/GravNet) for fast clustering of irregular geometry detectors
- hls4ml support for specific graph networks/layers (GarNet/JEDInet), but is moving to lower-level blocks for more generic support (PyTorch Geometric?)

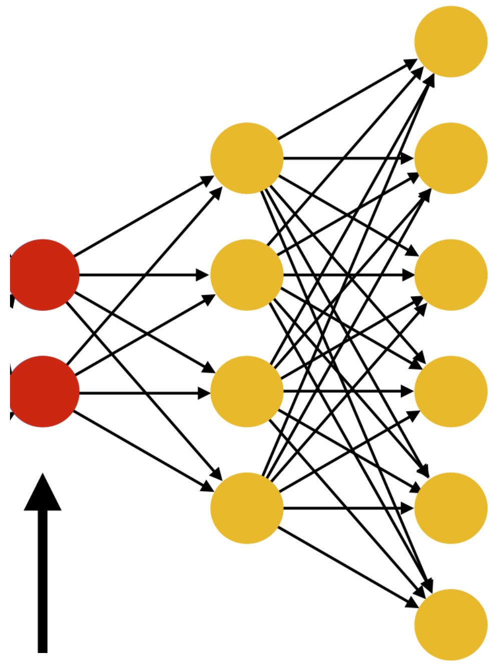
ML for reconstruction



On FPGA: 3.5 μs to cluster energy deposits

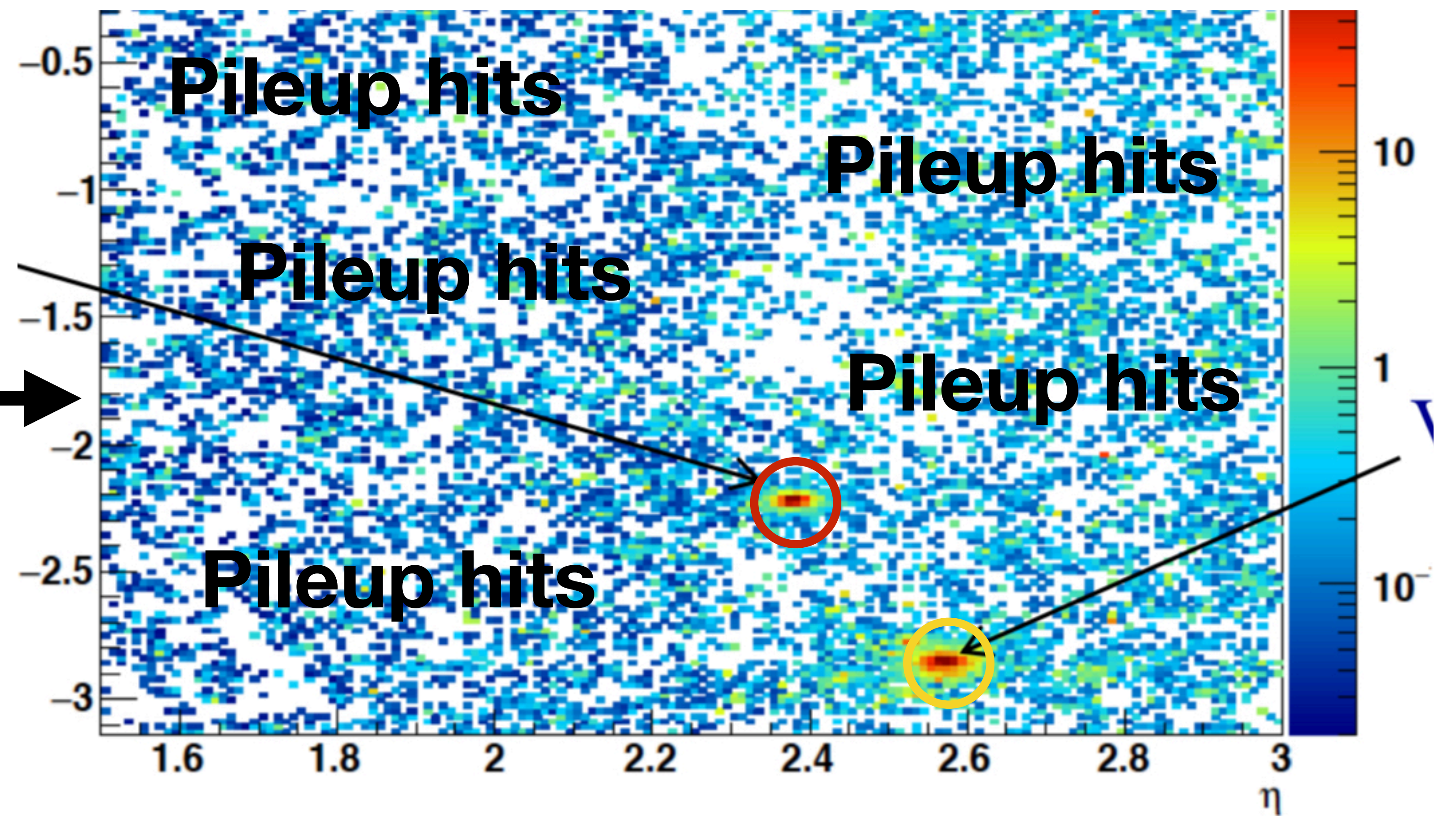
ML for reconstruction





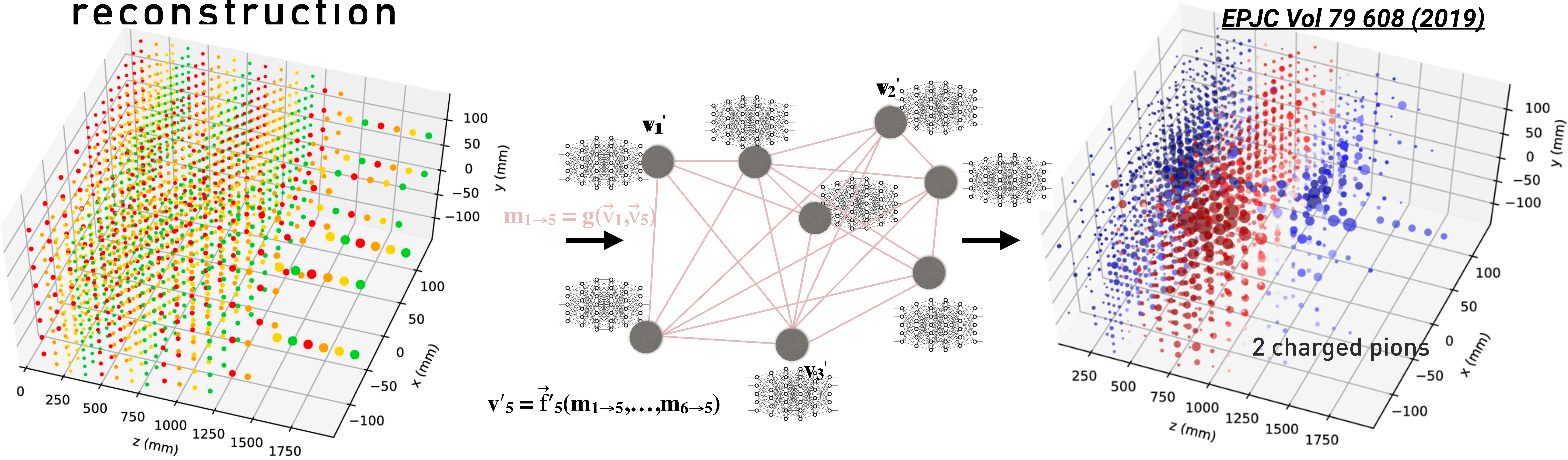
Encoded data

On FPGA



On FPGA: 3.5 μs to cluster energy deposits

ML for reconstruction



EPJC Vol 79 608 (2019)

On FPGA: 3.5 μs to cluster energy deposits

- Graph Neural Networks (GarNet/GravNet) for fast clustering of irregular geometry detectors
- hls4ml support for specific graph networks/layers (GarNet/JEDInet), but is moving to lower-level blocks for more generic support (PyTorch Geometric?)