





JAVIER DUARTE
ASPEN WINTER CONFERENCE
MARCH 28, 2023

MLFOR TRIGGERING

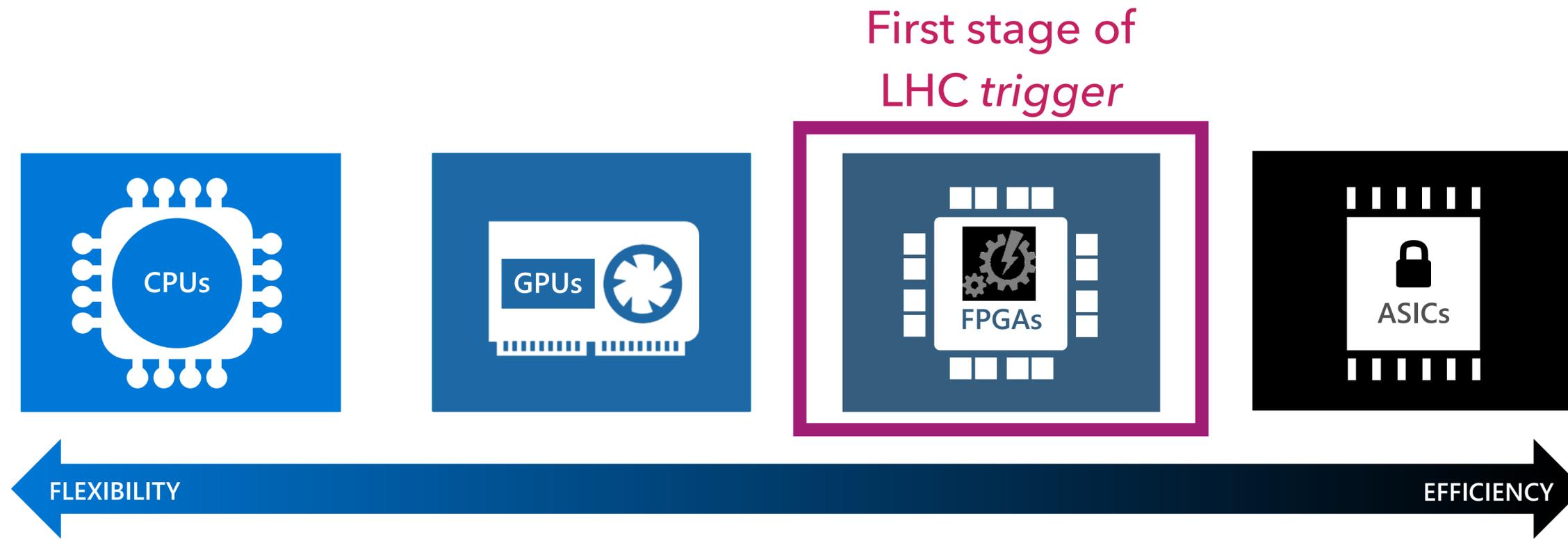
An aerial night view of a snowy mountain town, likely a ski resort, with lights from buildings and streets illuminating the valley. The background shows snow-covered mountains under a dark blue sky.

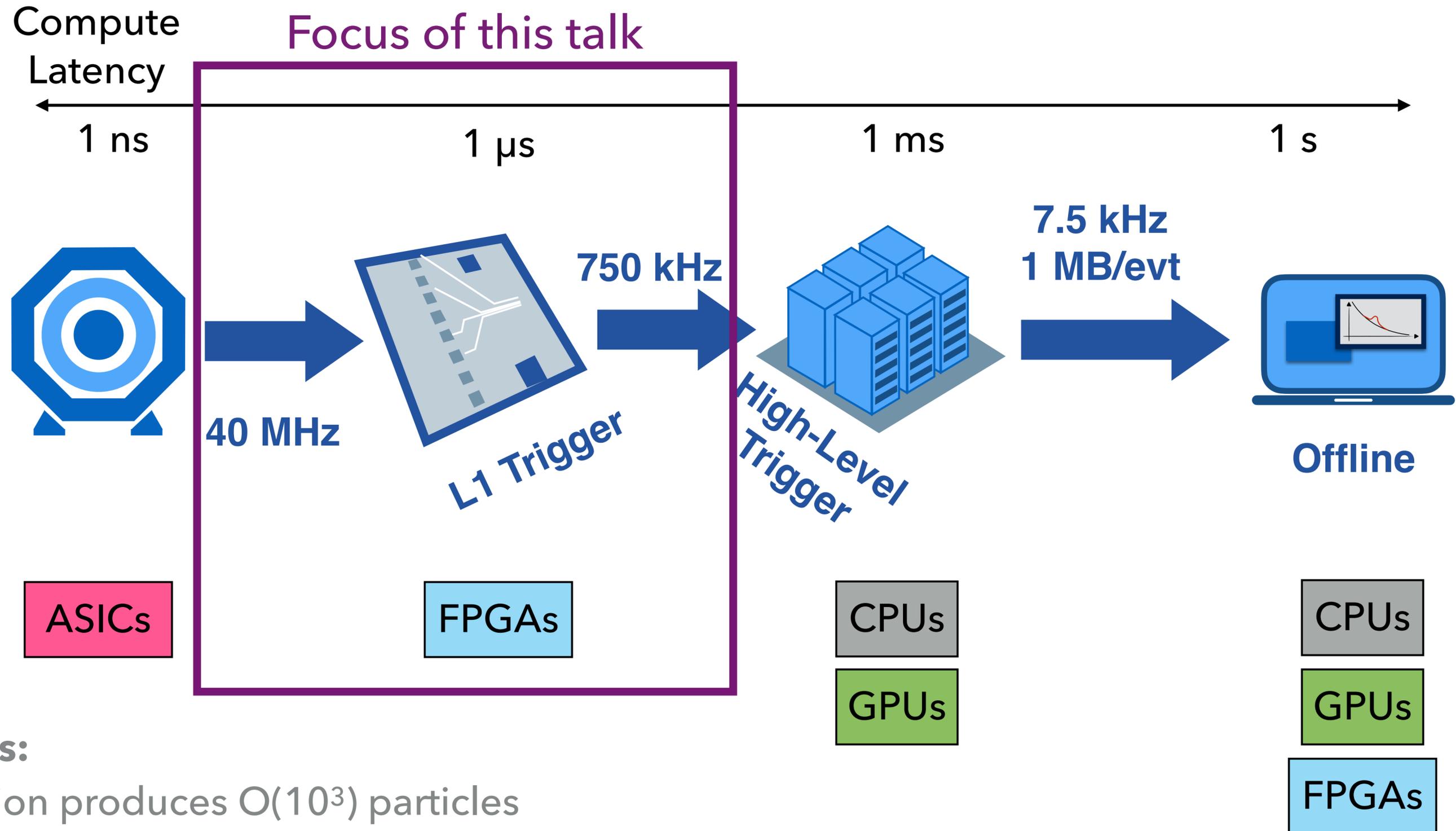
I. INTRO & MOTIVATION

II. COMPRESSION

III. HARDWARE

IV. APPLICATIONS





Challenges:

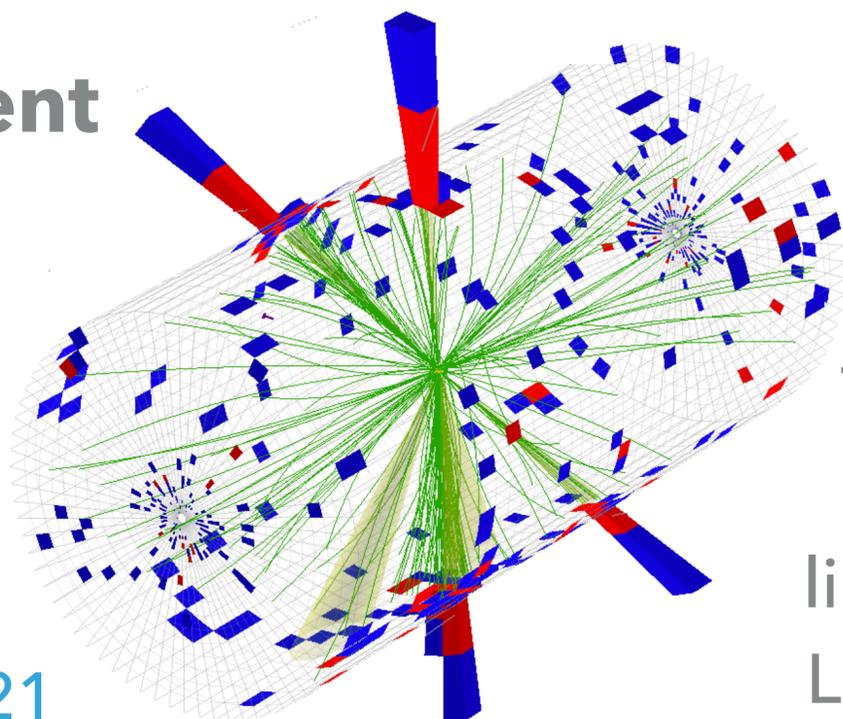
- Each collision produces $O(10^3)$ particles
- The detectors have $O(10^8)$ sensors
- Extreme data rates of $O(100 \text{ TB/s})$

Exabyte-scale datasets

SIMPLIFIED HL-LHC L1 TRIGGER MENU

- ▶ Single/double/triple muons/electrons
- ▶ Photons
- ▶ Taus
- ▶ Hadronic
- ▶ Missing transverse energy
- ▶ "Cross" triggers (not shown)

4-jet event

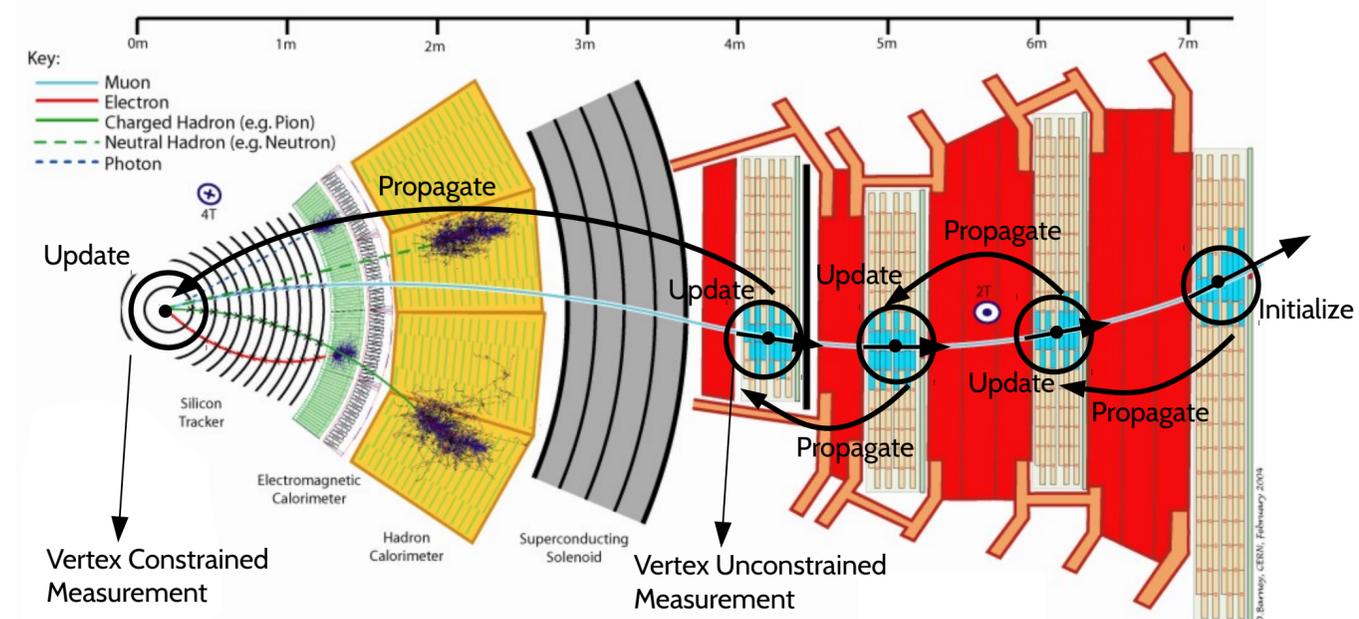
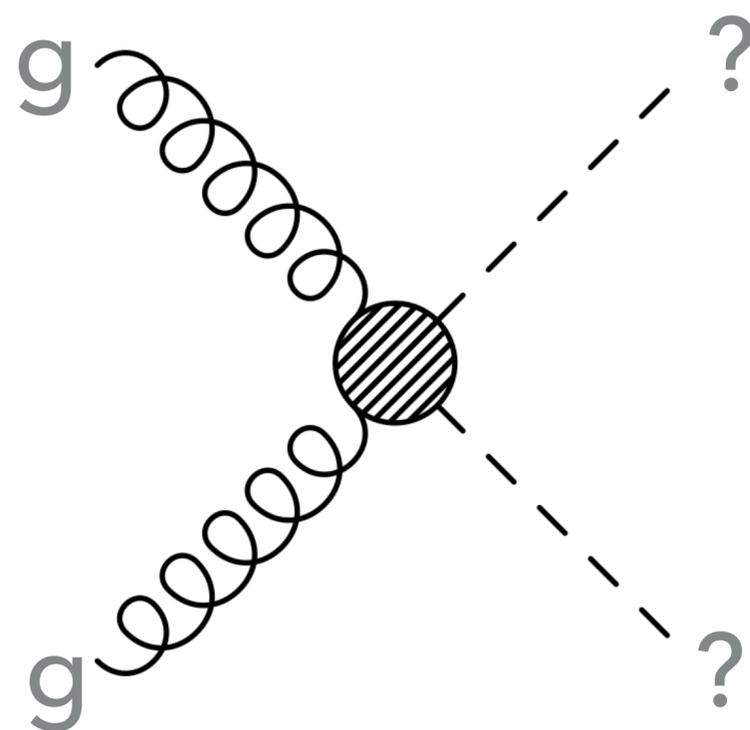
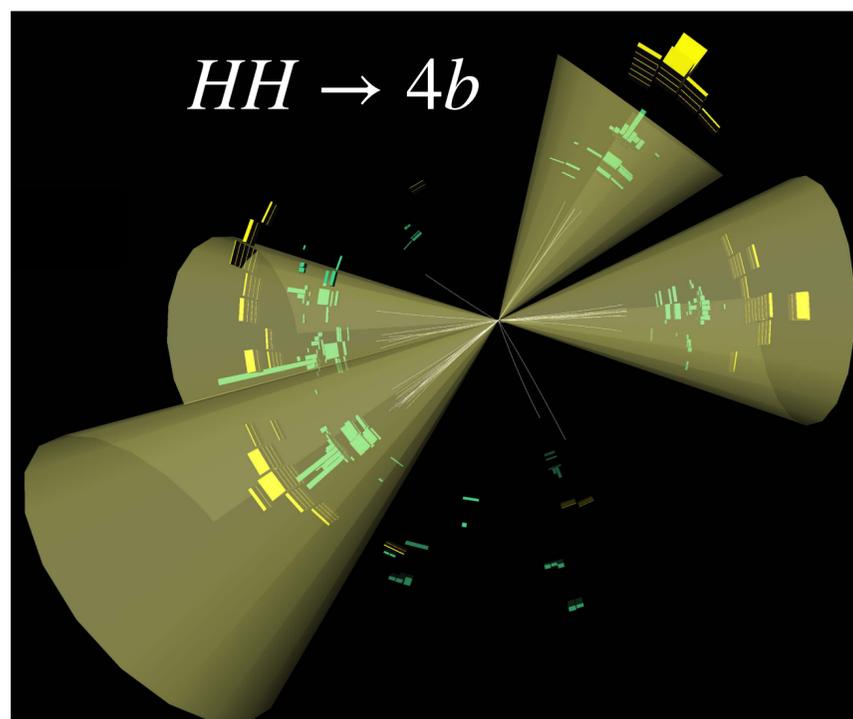
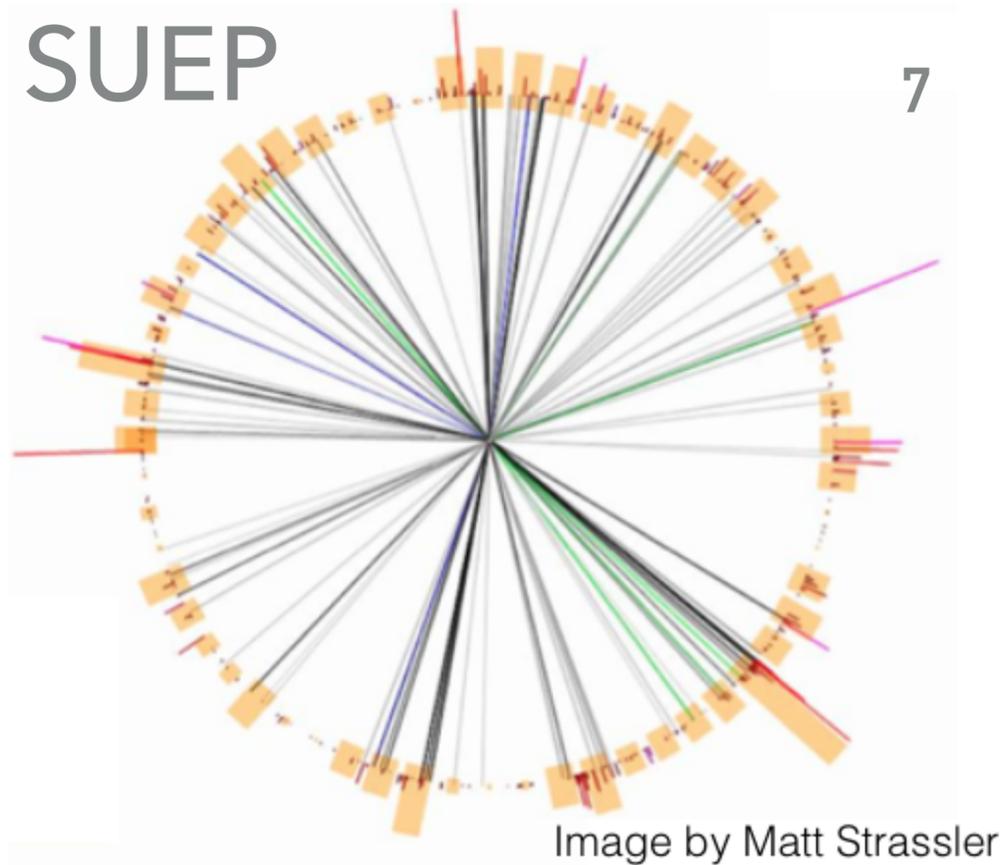


Thresholds set by backgrounds, limited resolution @ L1, and rate budget

Trigger	Threshold [GeV]
1 μ	22
2 μ	15, 7
3 μ	5, 3, 3
1 e	36
2 e	25, 12
1 γ	36
2 γ	22, 12
1 τ	150
2 τ	90, 90
1 jet	180
2 jet	112, 112
H_T	450
4 jet + H_T	75, 55, 40, 40, 400
p_{T}^{miss}	200

WHAT COULD WE BE MISSING?

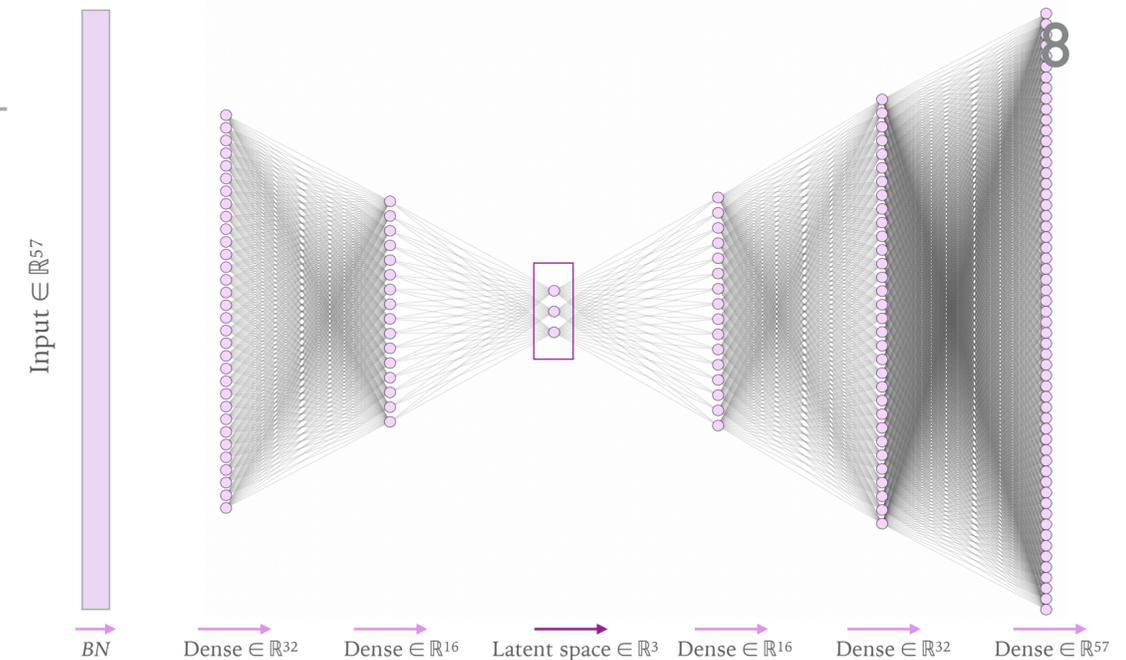
- ▶ How can we trigger on more complex low-energy hadronic signatures? Long-lived/displaced particles?
- ▶ What if we don't know exactly what to look for?
- ▶ What if our signatures require complex multivariate algorithms (e.g. b tagging)?
- ▶ How can we improve on our traditional (often slow) reconstruction algorithms?



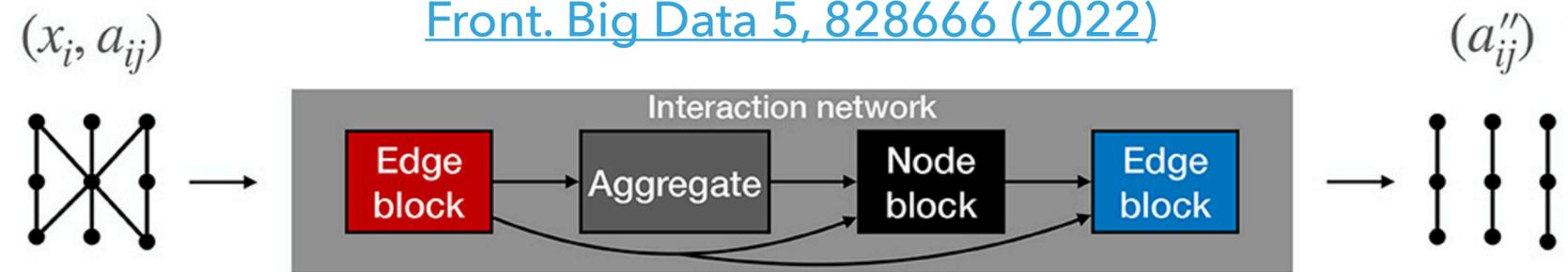
ML IN THE TRIGGER

- ▶ (Variational) autoencoders for anomaly detection
- ▶ 1D convolutional neural networks for b-tagging
- ▶ Graph neural networks for tracking

Nat. Mach. Intell. 4, 154 (2022)



Front. Big Data 5, 828666 (2022)

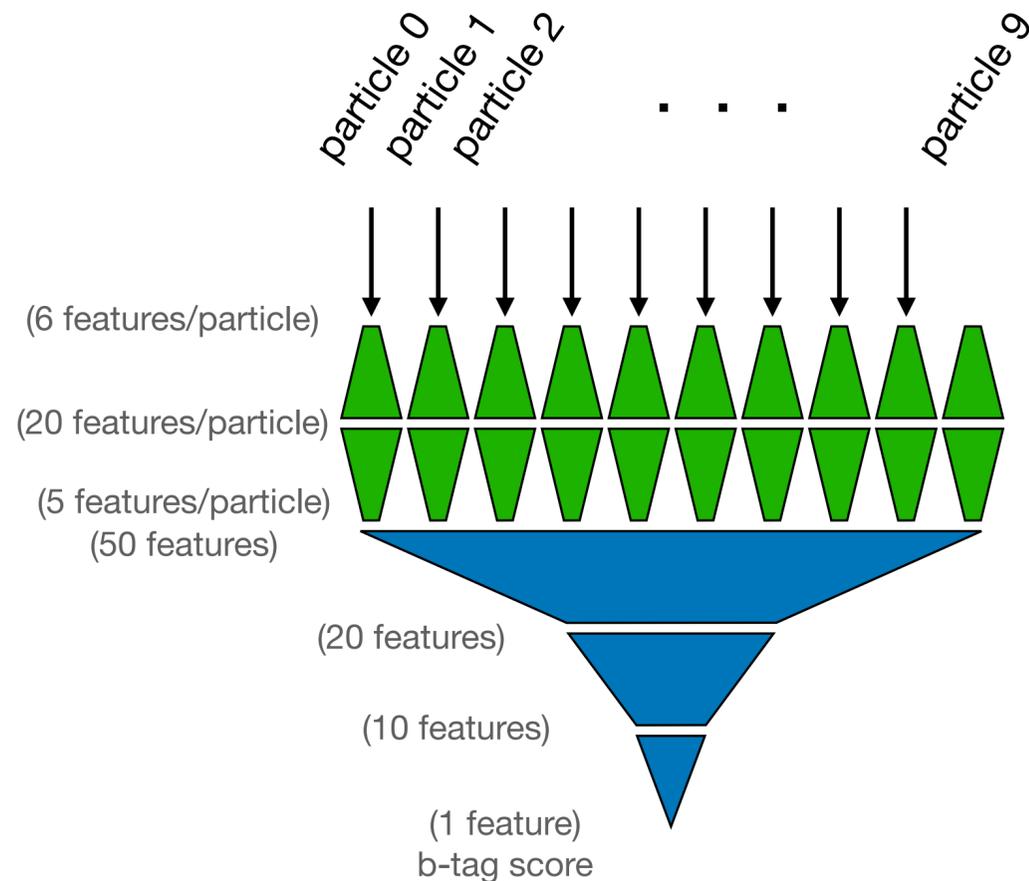
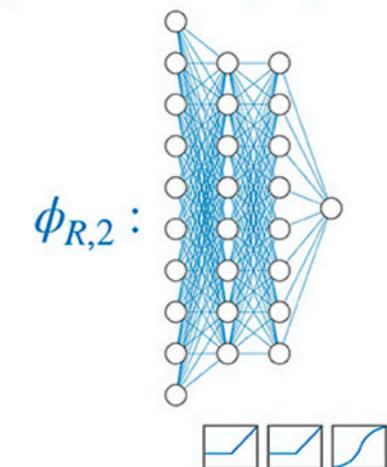
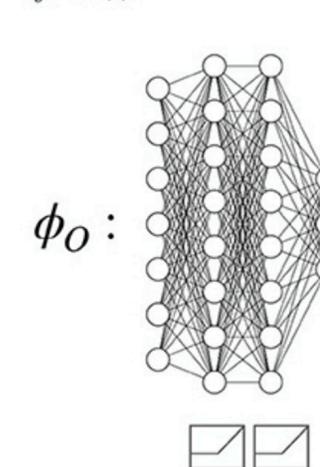
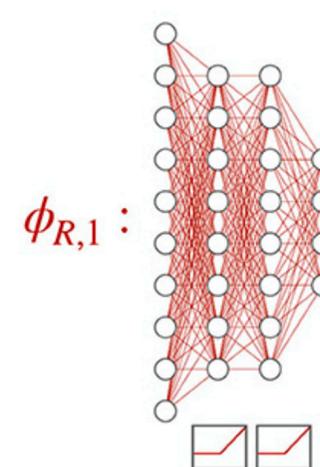


$$a'_{ij} = \phi_{R,1}(x_i, x_j, a_{ij})$$

$$x'_i = \phi_O(x_i, \bar{a}_i)$$

$$\bar{a}_i = \sum_{j \in N(i)} a'_{ij}$$

$$a''_{ij} = \phi_{R,2}(x'_i, x'_j, a'_{ij})$$

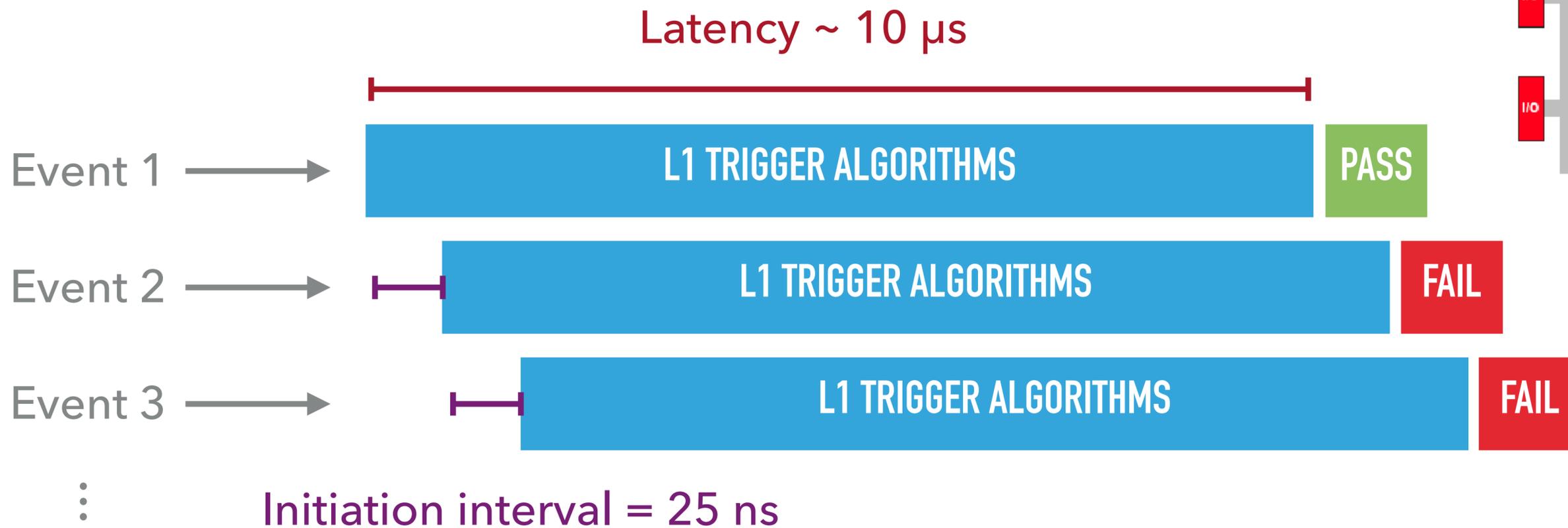
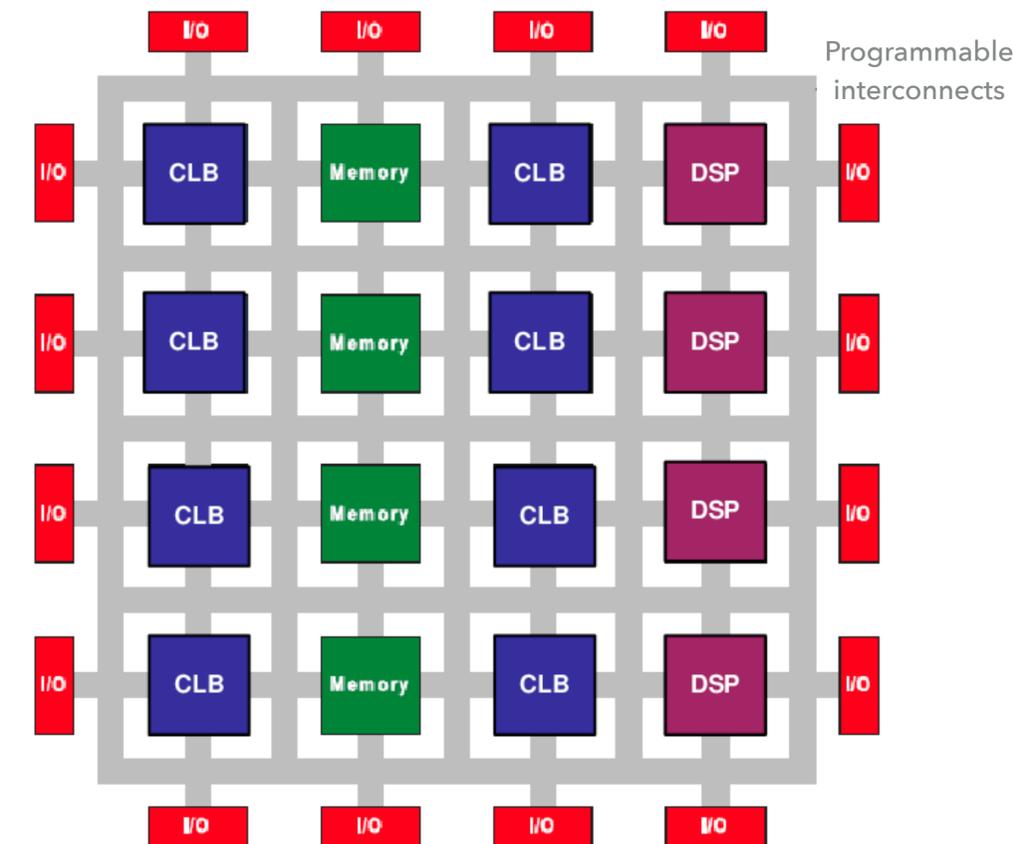


Pointwise convolution (per particle dense layer)

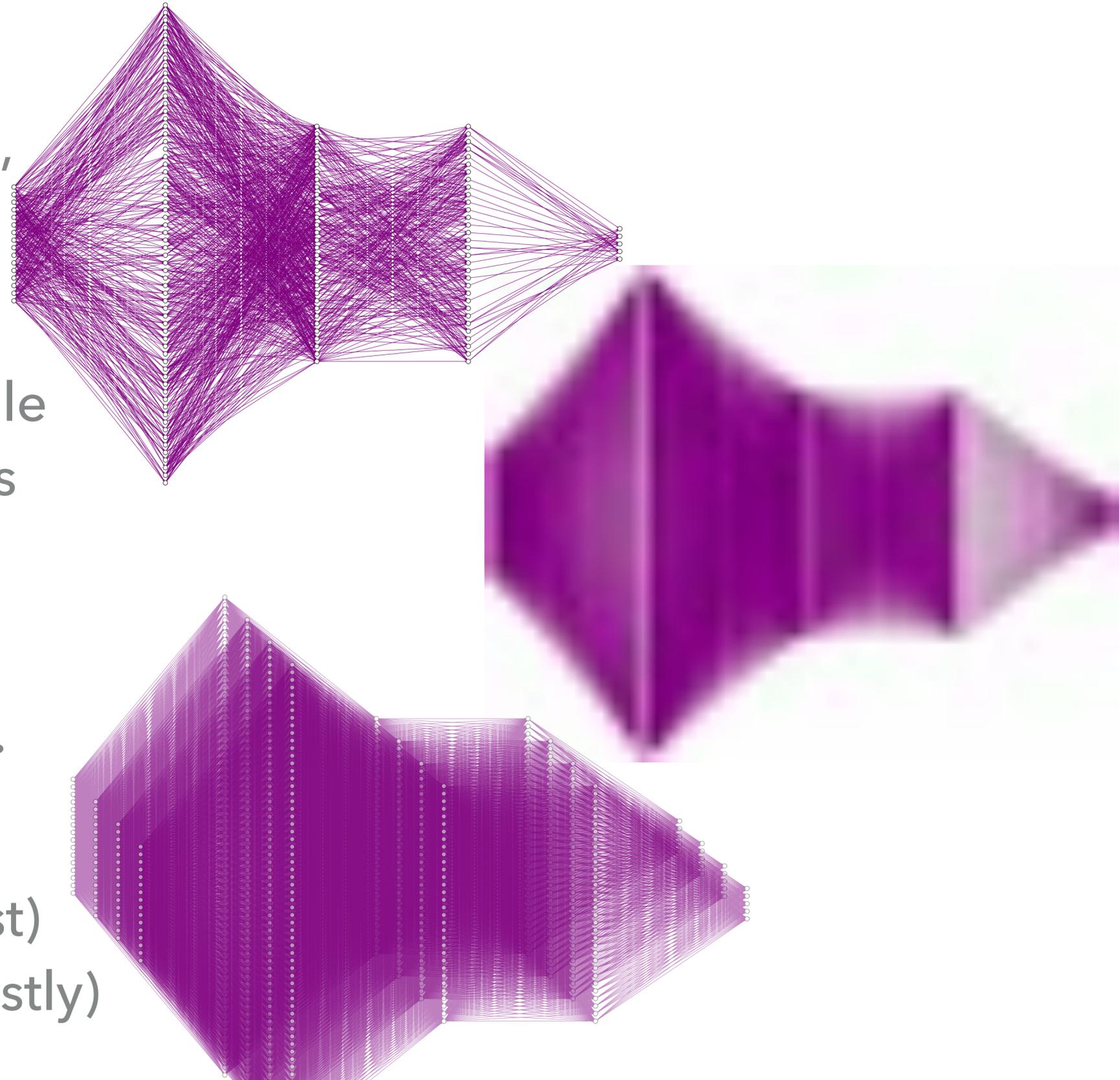
Dense layer

WHAT MAKES THIS HARD?

- ▶ Reconstruct all events and reject 98% of them in $\sim 10 \mu\text{s}$
 - ▶ Algorithms have to be $< 1 \mu\text{s}$ and process new events every $(25 \text{ ns}) \times N_{tmux}$
- ▶ Latency necessitates all **FPGA** design
 - ▶ Algorithms have to fit on < 1 FPGA
- ▶ How can we satisfy these constraints?



- ▶ **Codesign:** intrinsic development loop between ML design, training, and implementation
- ▶ Pruning
 - ▶ Maintain high performance while removing redundant operations
- ▶ Quantization
 - ▶ Reduce precision from 32-bit floating point to 16-bit, 8-bit, ...
- ▶ Parallelization
 - ▶ Balance parallelization (how fast) with resources needed (how costly)



An aerial night view of a snowy mountain town, likely Aspen, Colorado. The town is illuminated with warm yellow and orange lights, contrasting with the dark blue night sky and the white snow on the mountains. The text is overlaid on the image.

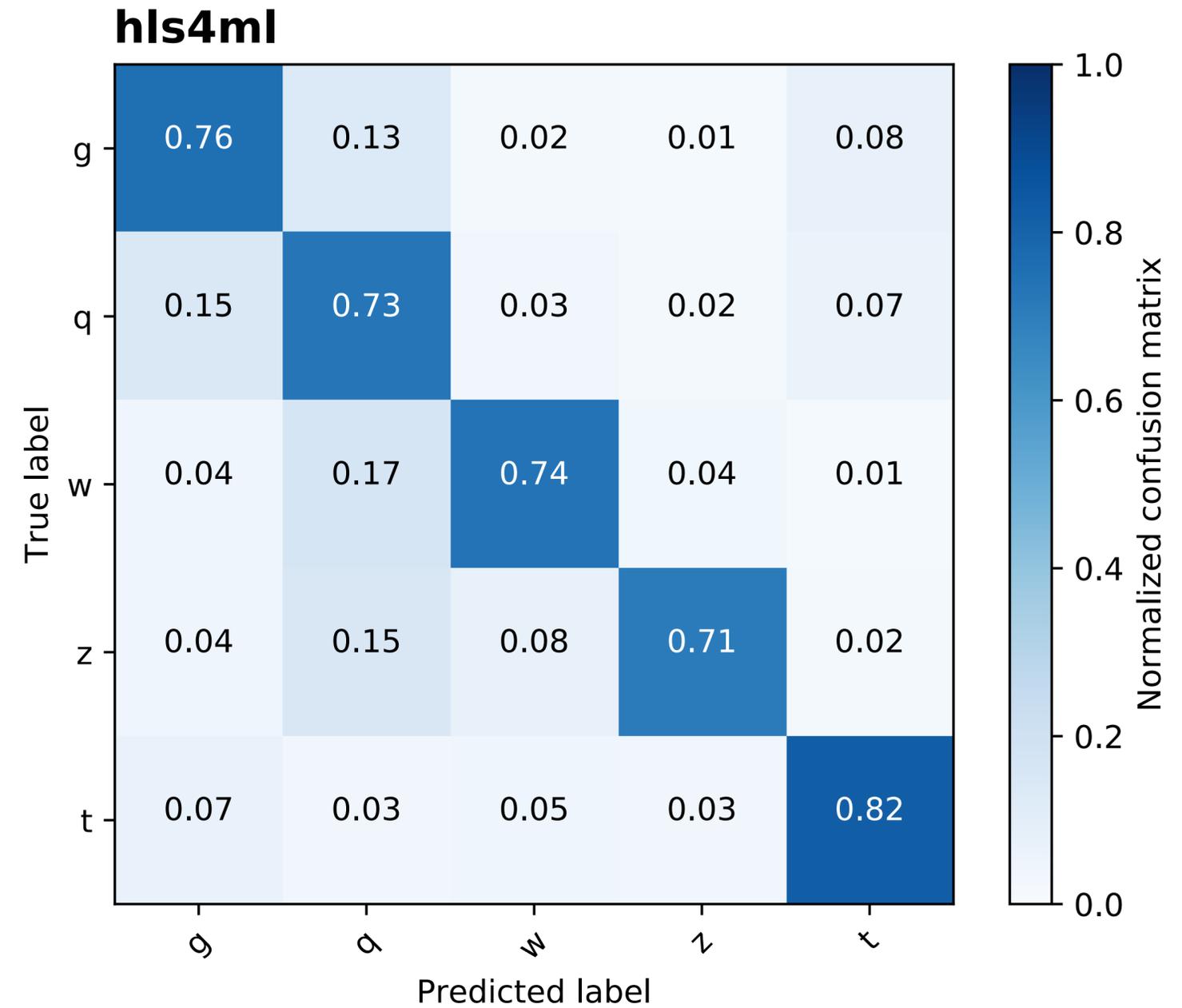
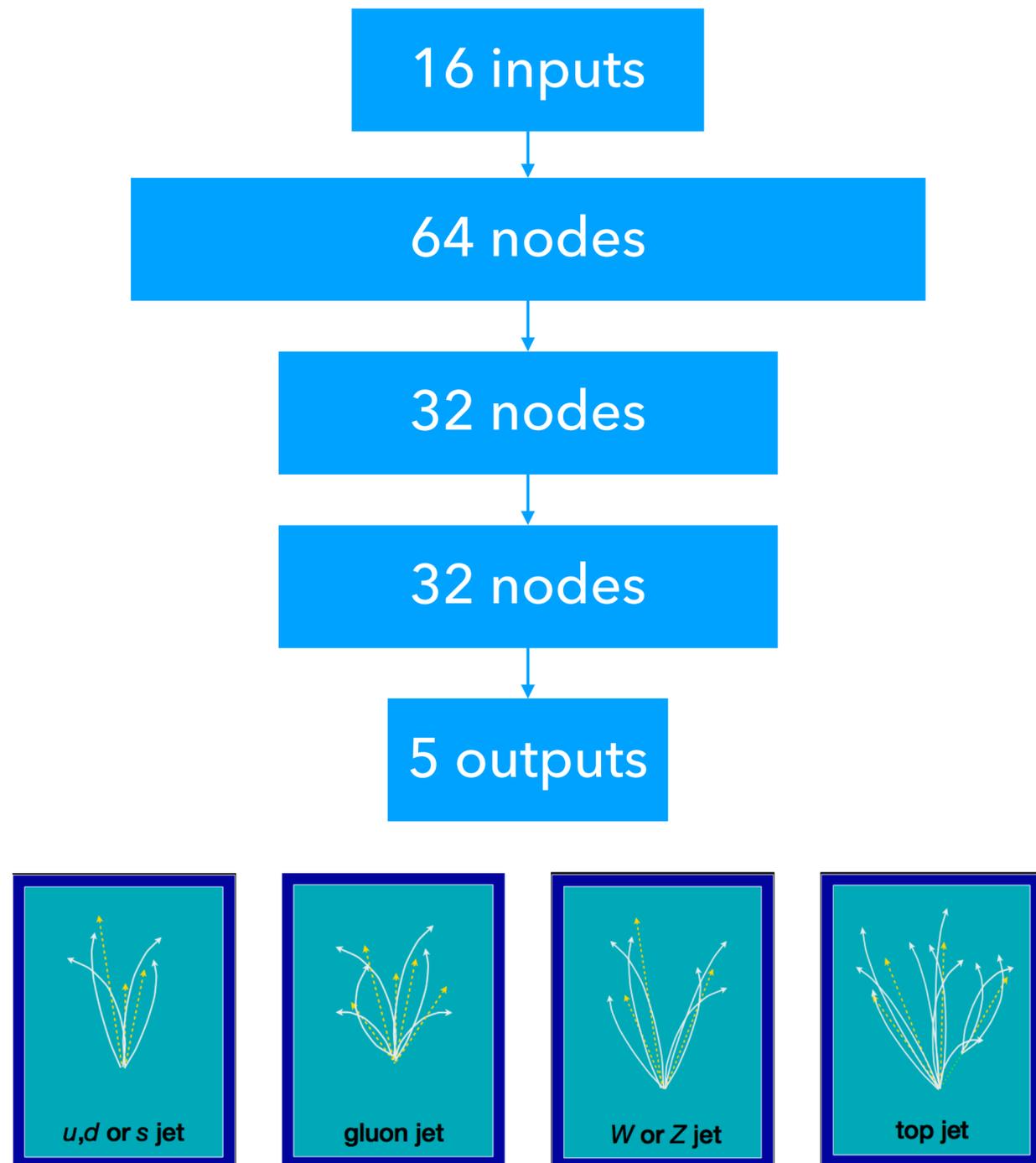
I. INTRO & MOTIVATION

II. COMPRESSION

III. HARDWARE

IV. APPLICATIONS

Small NN benchmark correctly identifies particle "jets" 70-80% of the time



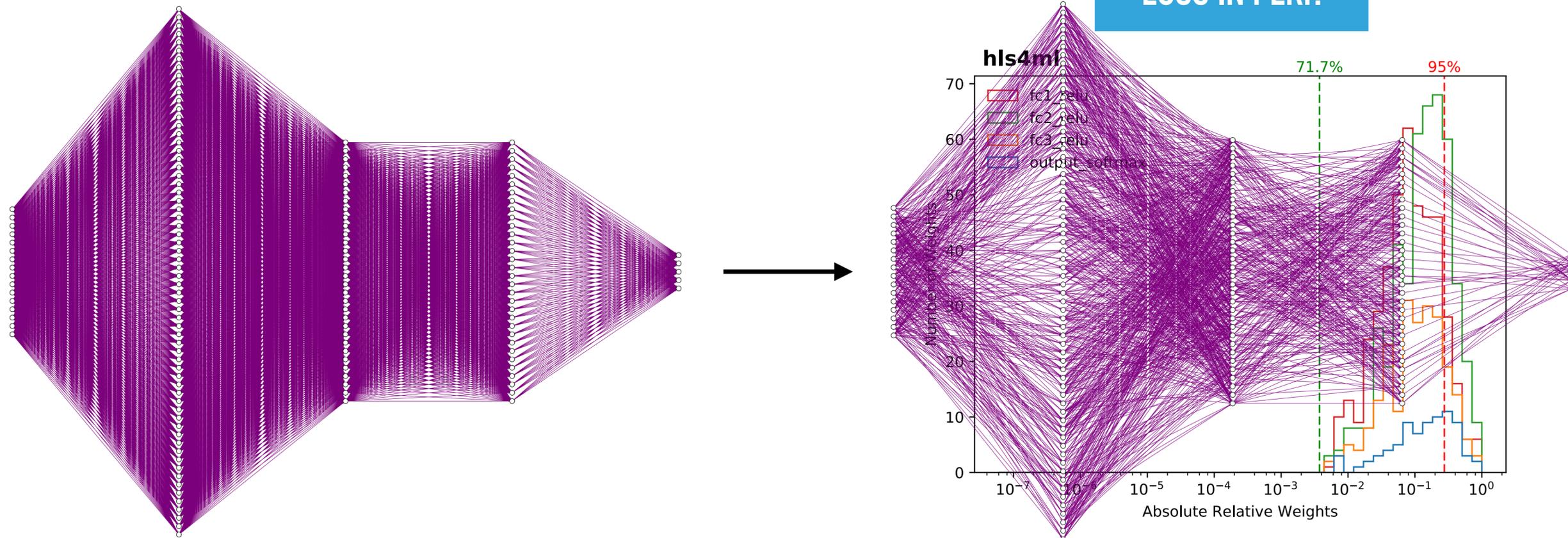
- ▶ Train with **L₁ regularization** (down-weights unimportant synapses)

Used in
[CMS-DP-2022-020](#)
for NN vertex finder

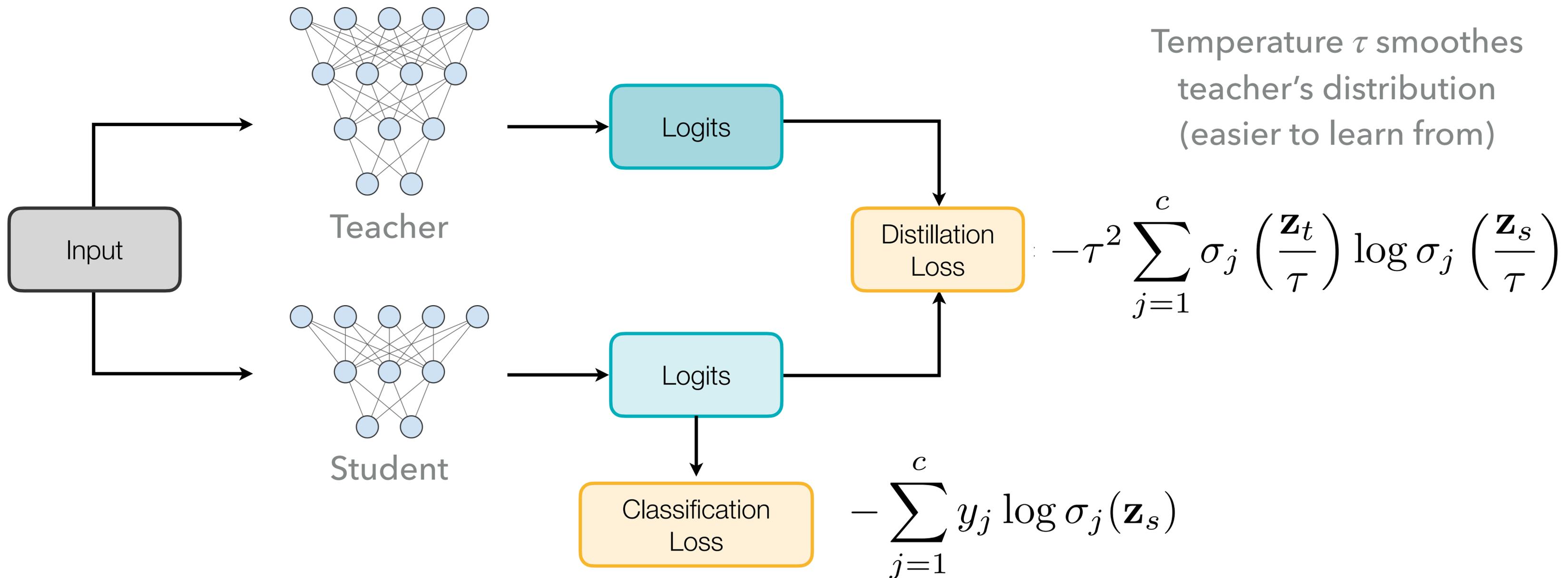
$$L_{\lambda}(\mathbf{w}) = L(\mathbf{w}) + \lambda \|\mathbf{w}\|_1$$

$$\|\mathbf{w}\|_1 = \sum_i |w_i|$$

- ▶ Remove **smallest** weights
- ▶ Iterate

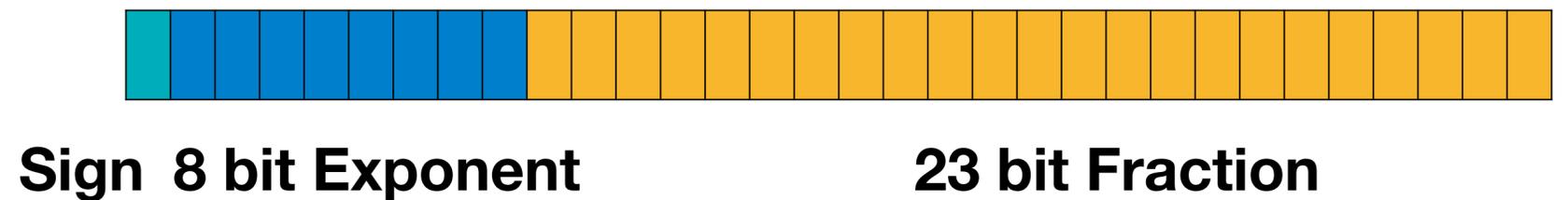


- ▶ Can we compress the architecture as well?
- ▶ Knowledge distillation: training a small **student network** to emulate a larger **teacher model** or ensemble of networks

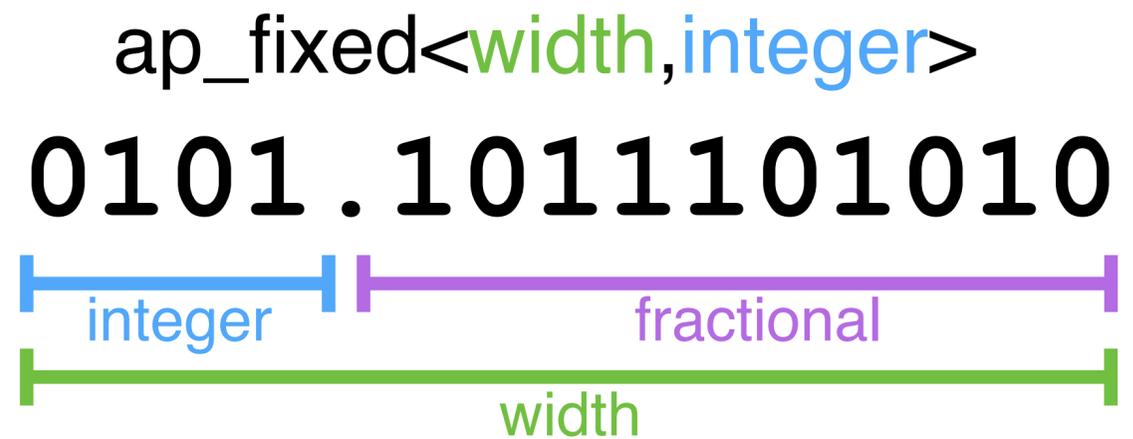


- ▶ Quantization: using reduced precision for parameters and operations

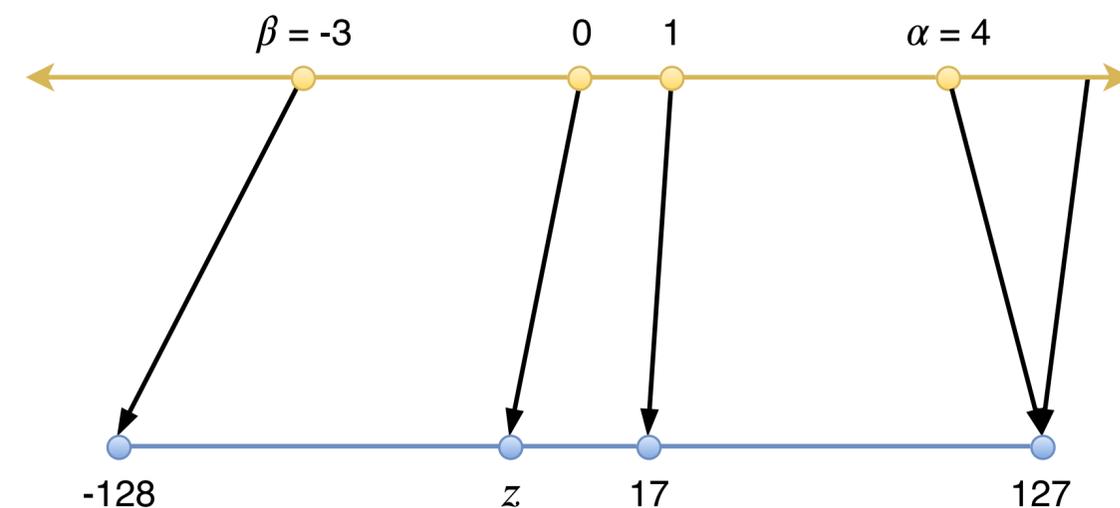
- ▶ Baseline: 32-bit floating-point precision

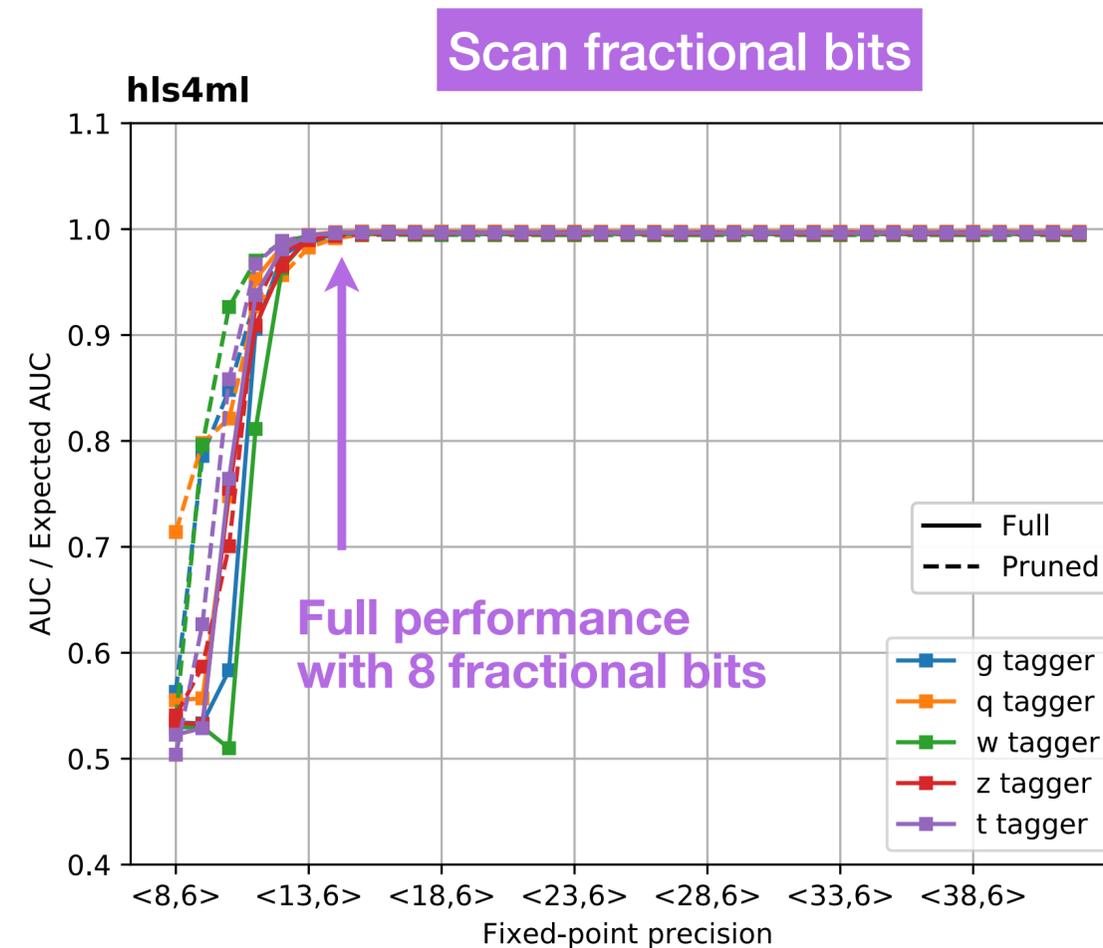
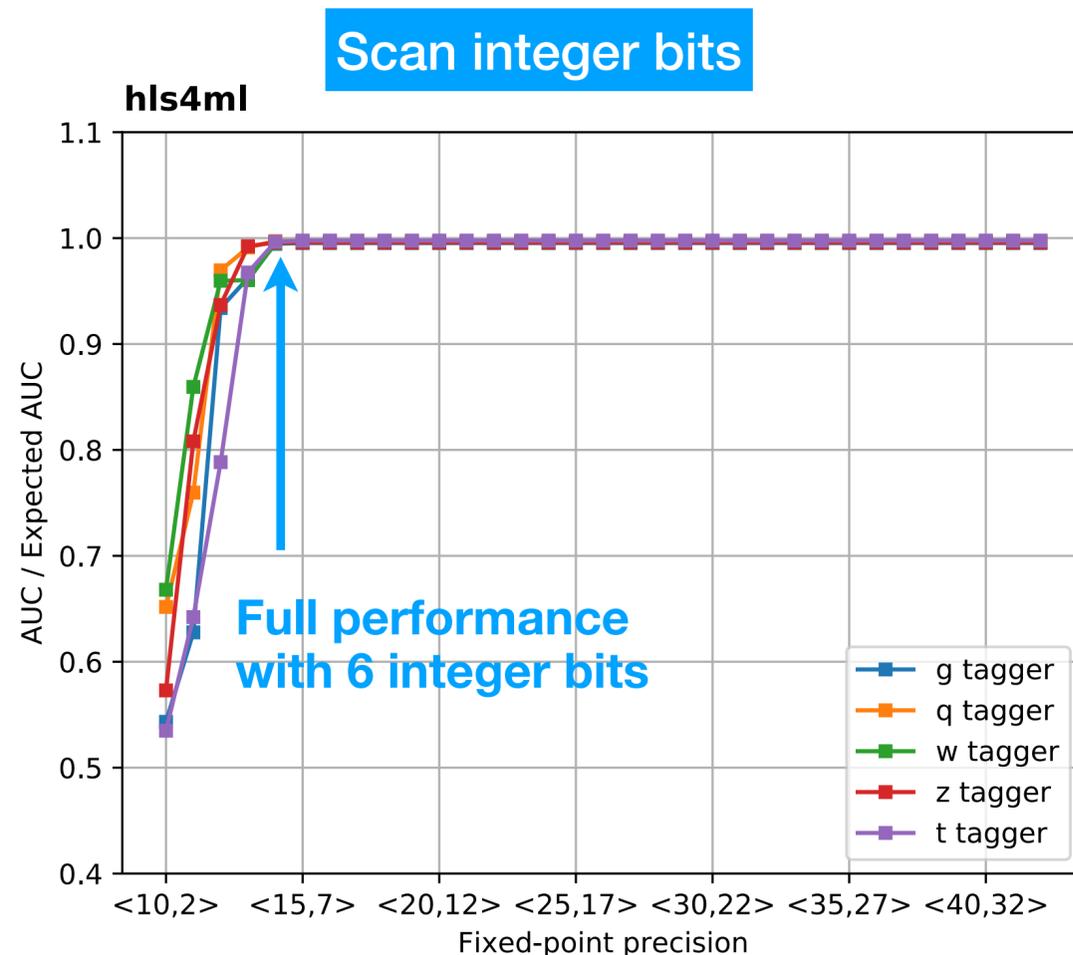


- ▶ Fixed-point precision

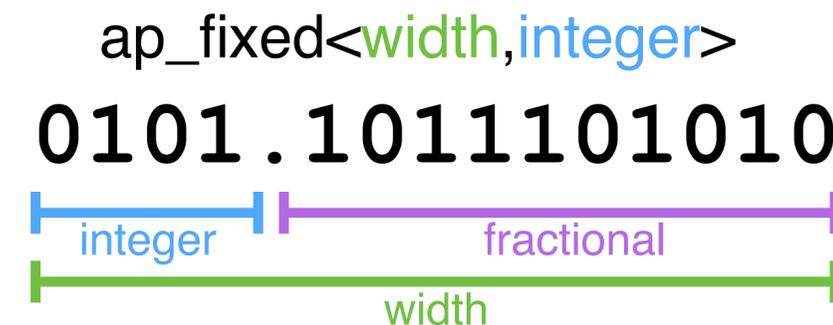


- ▶ Affine integer quantization

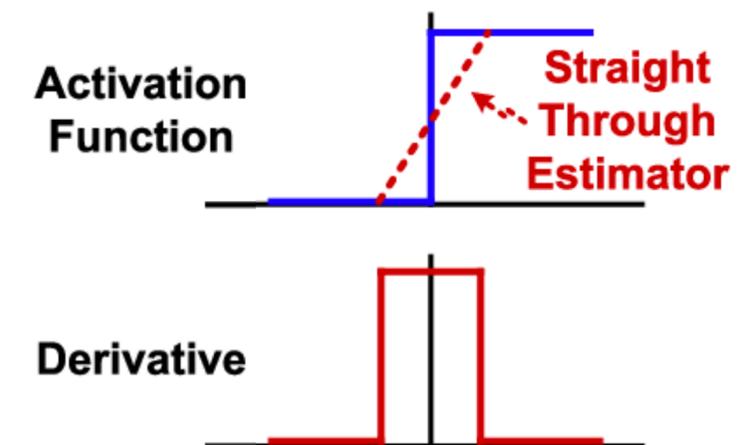
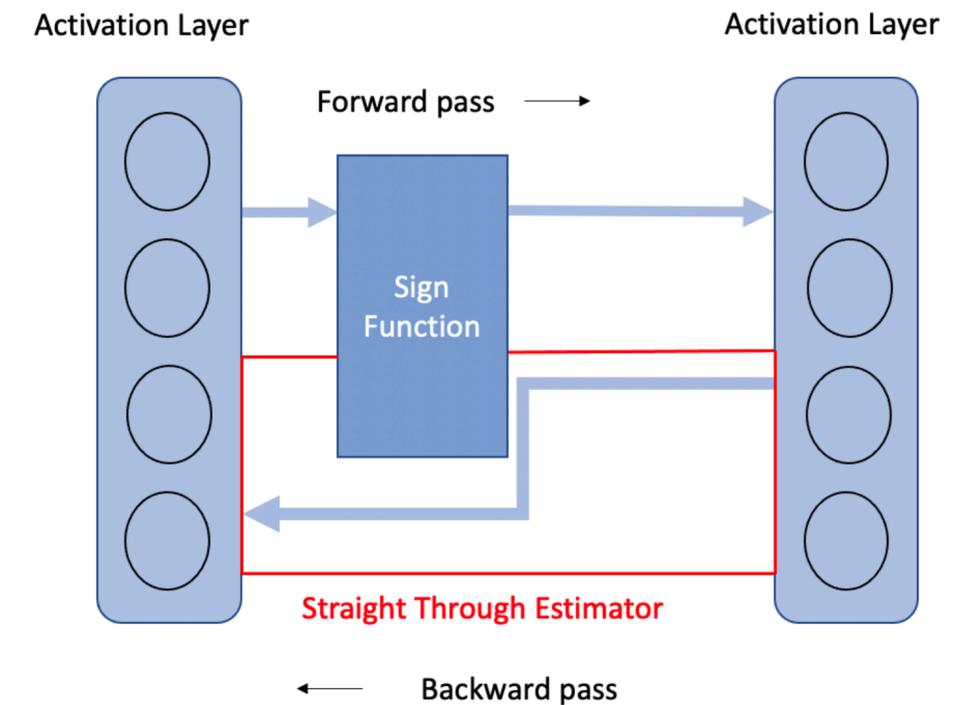
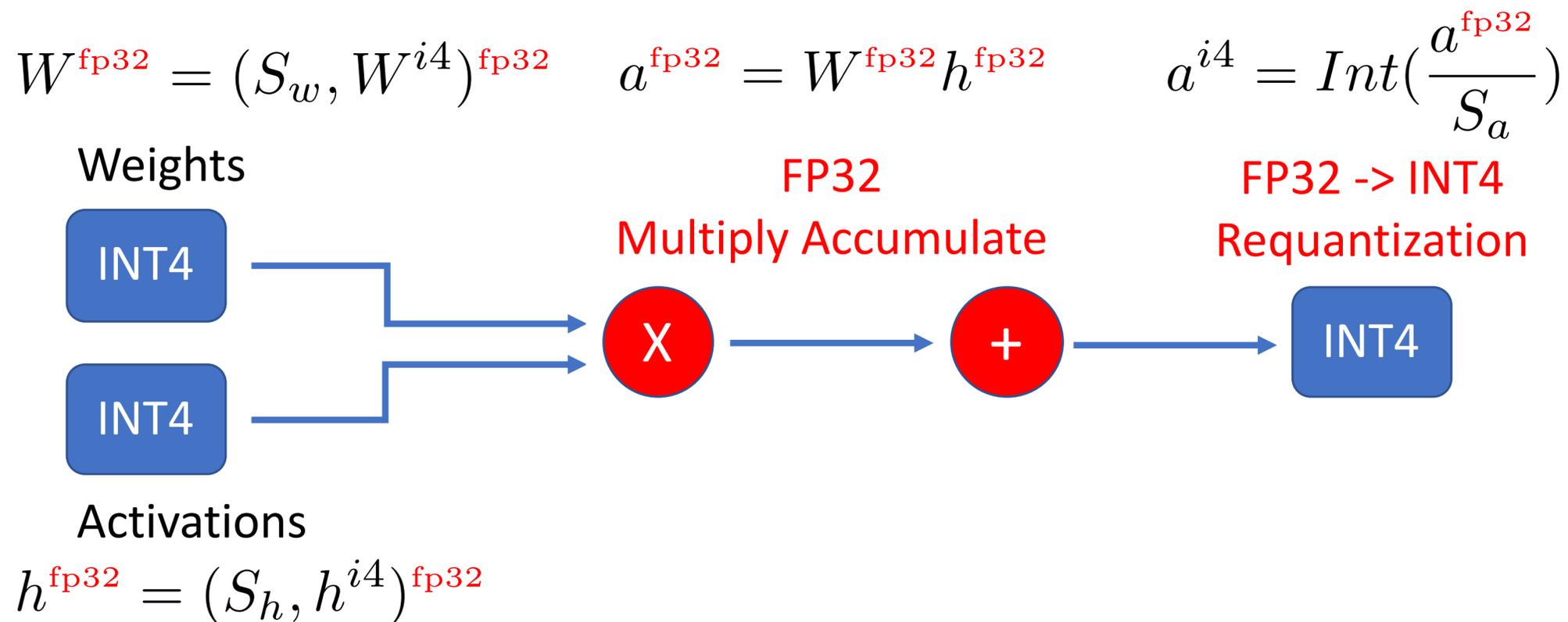




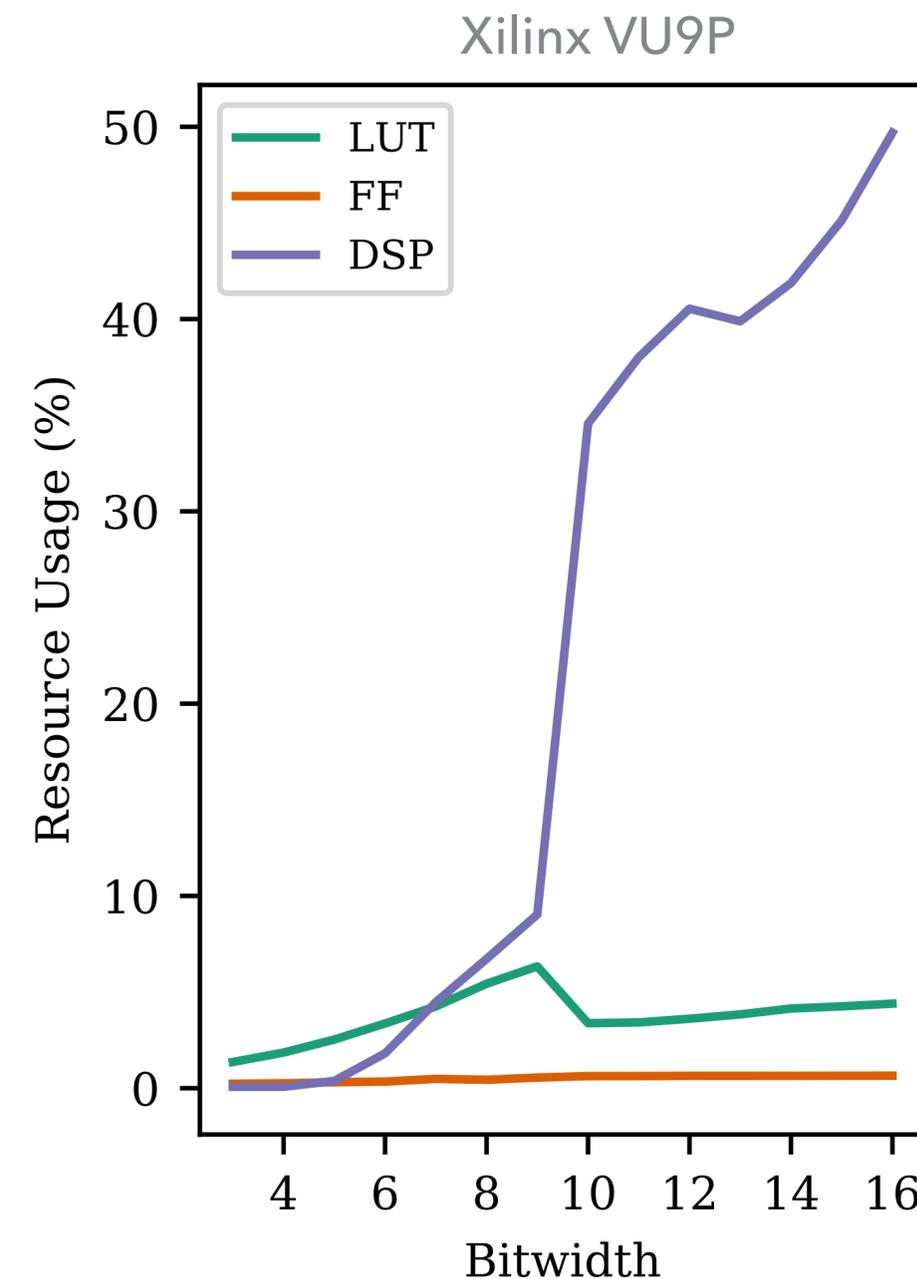
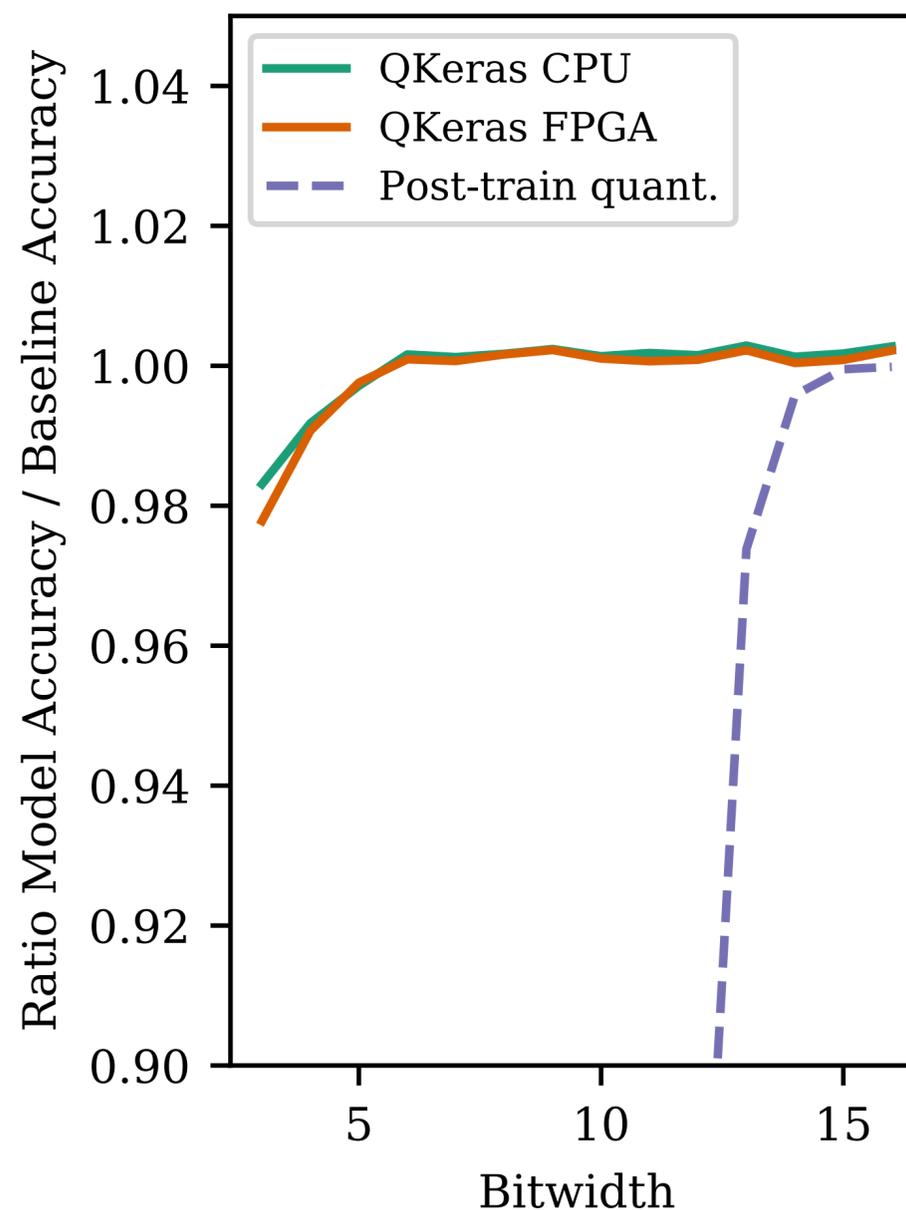
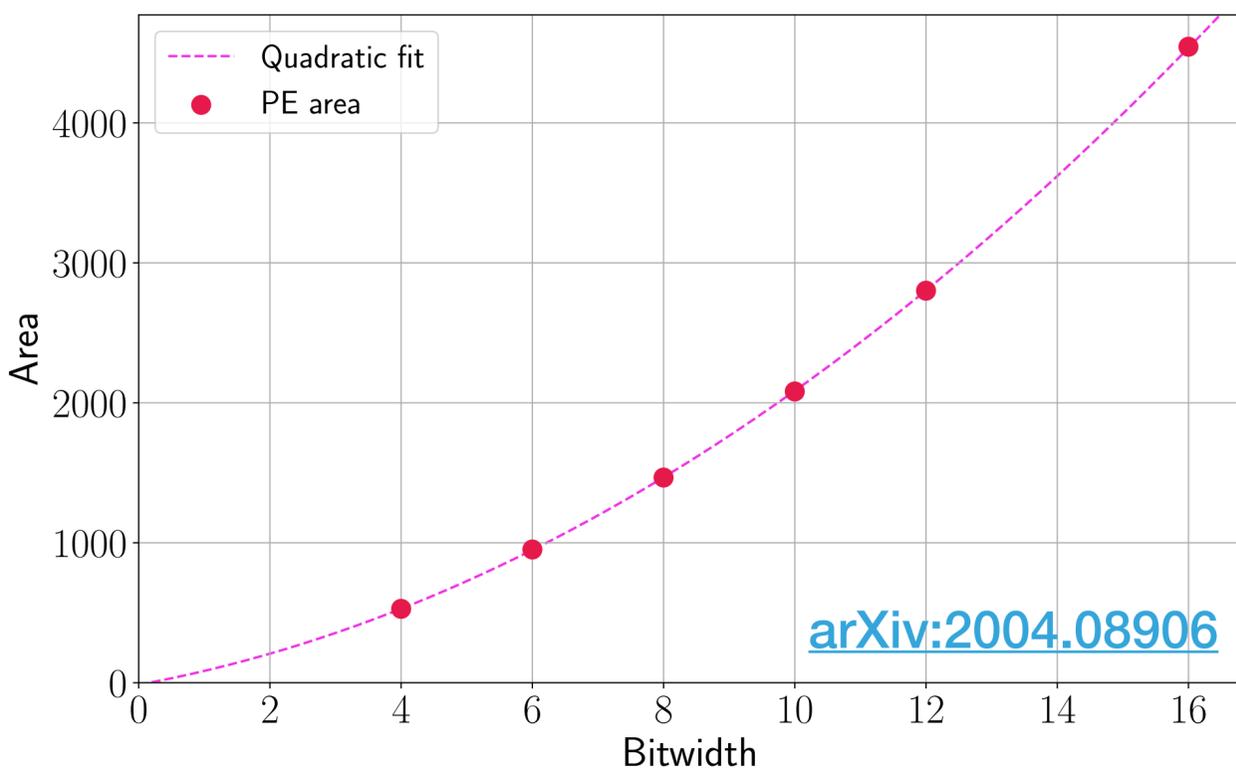
- ▶ General strategy: avoid overflows in integer bit
- ▶ Then scan the fractional bit width until reaching optimal performance



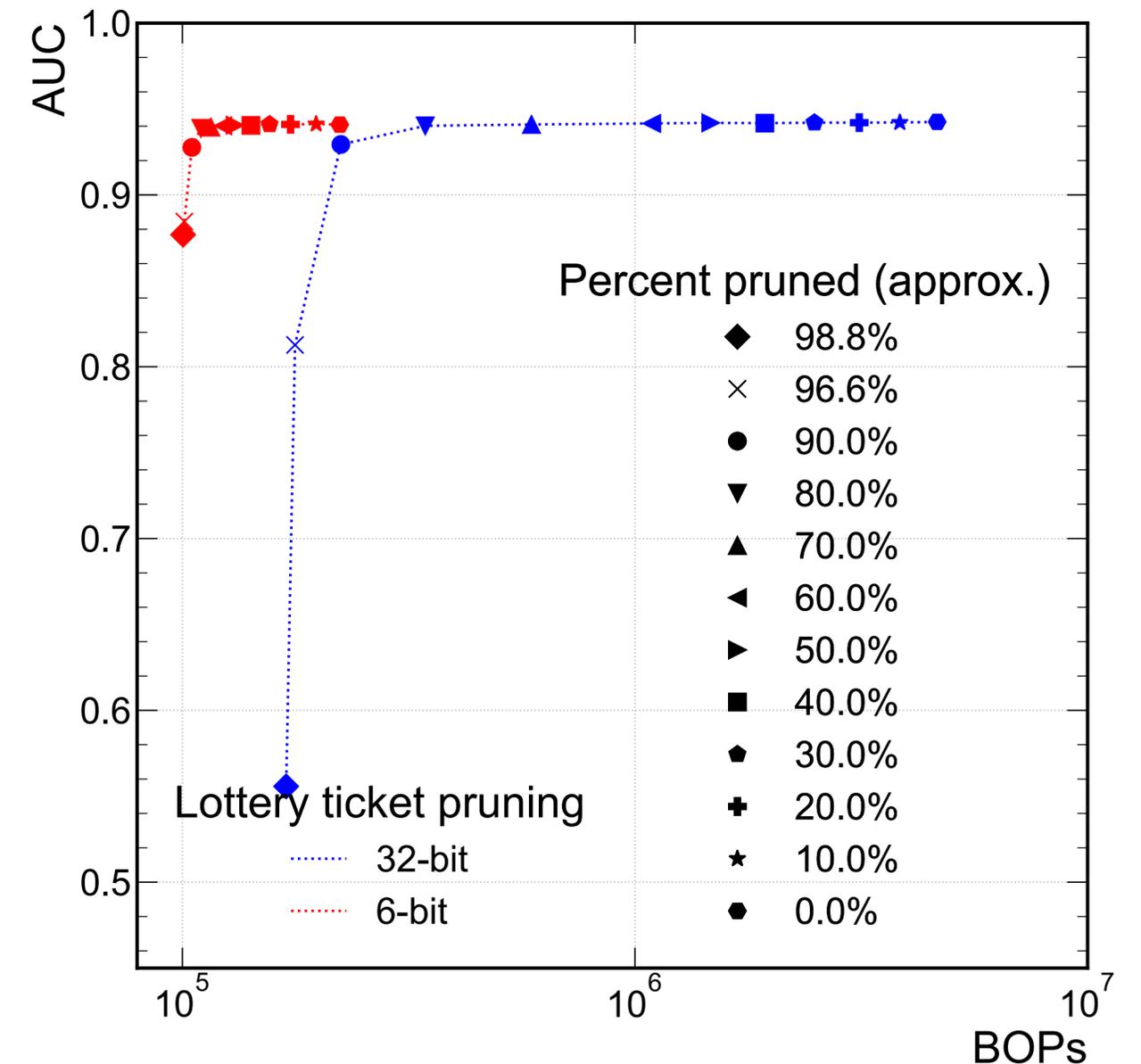
- ▶ Fake quantization: using 32-bit floating-point under the hood
- ▶ Straight-through estimator: during backpropagation, ignore quantization operation (treat as identity)



- ▶ Full performance with 6 bits instead of 14 bits
- ▶ Much smaller fraction of resources
- ▶ Area & power scale quadratically with bit width



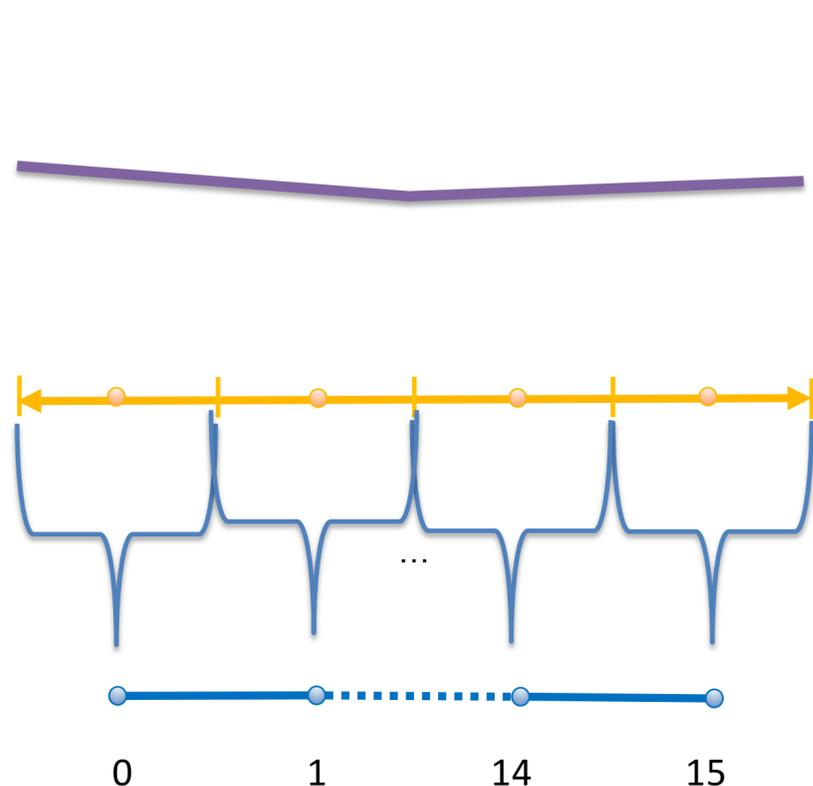
- ▶ Quantization-aware pruning (QAP): iterative pruning can further reduce the hardware computational complexity of a quantized model
- ▶ After QAP, the 6-bit, 80% pruned model achieves a factor of 50 reduction in BOPs compared to the 32-bit, unpruned model
- ▶ Study using [Brevitas](#)



Bit operations (BOPs) definition:

[arXiv:1804.10969](#)

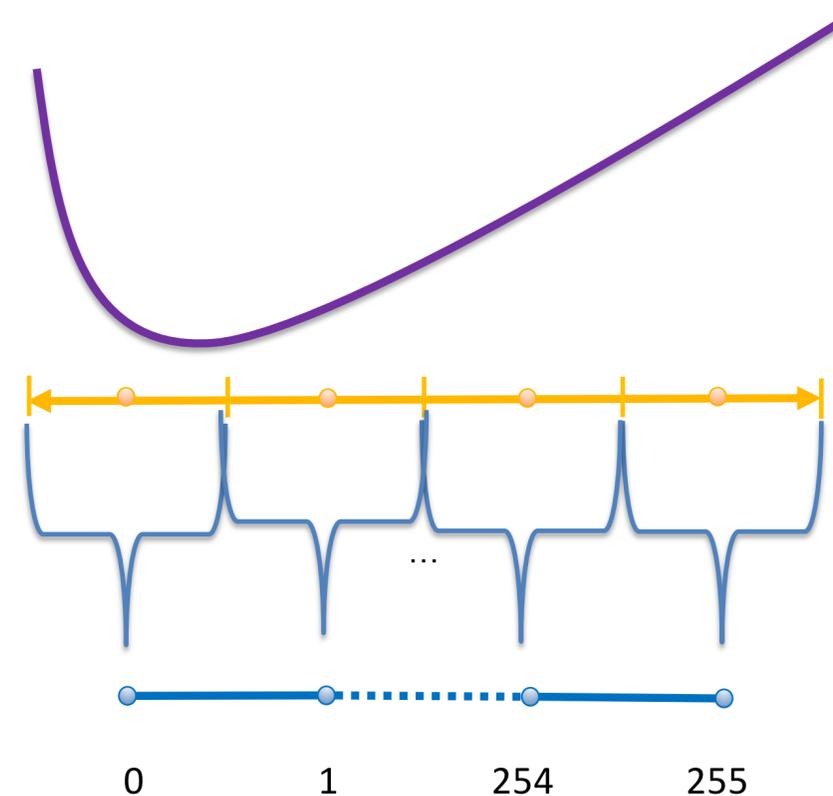
- ▶ Hessian of loss can provide additional guidance about quantization!
- ▶ Flat loss landscape: Lower bit width
- ▶ Sharp loss landscape: Higher bit width



Flat Loss Landscape

Floating Point values

4-bit Quantization



Sharp Loss Landscape

Floating Point values

8-bit Quantization

An aerial night view of a snowy mountain town, likely Aspen, Colorado. The town is illuminated with warm yellow and orange lights, contrasting with the dark blue night sky and the white snow on the mountains. The text is overlaid on the top half of the image.

I. INTRO & MOTIVATION

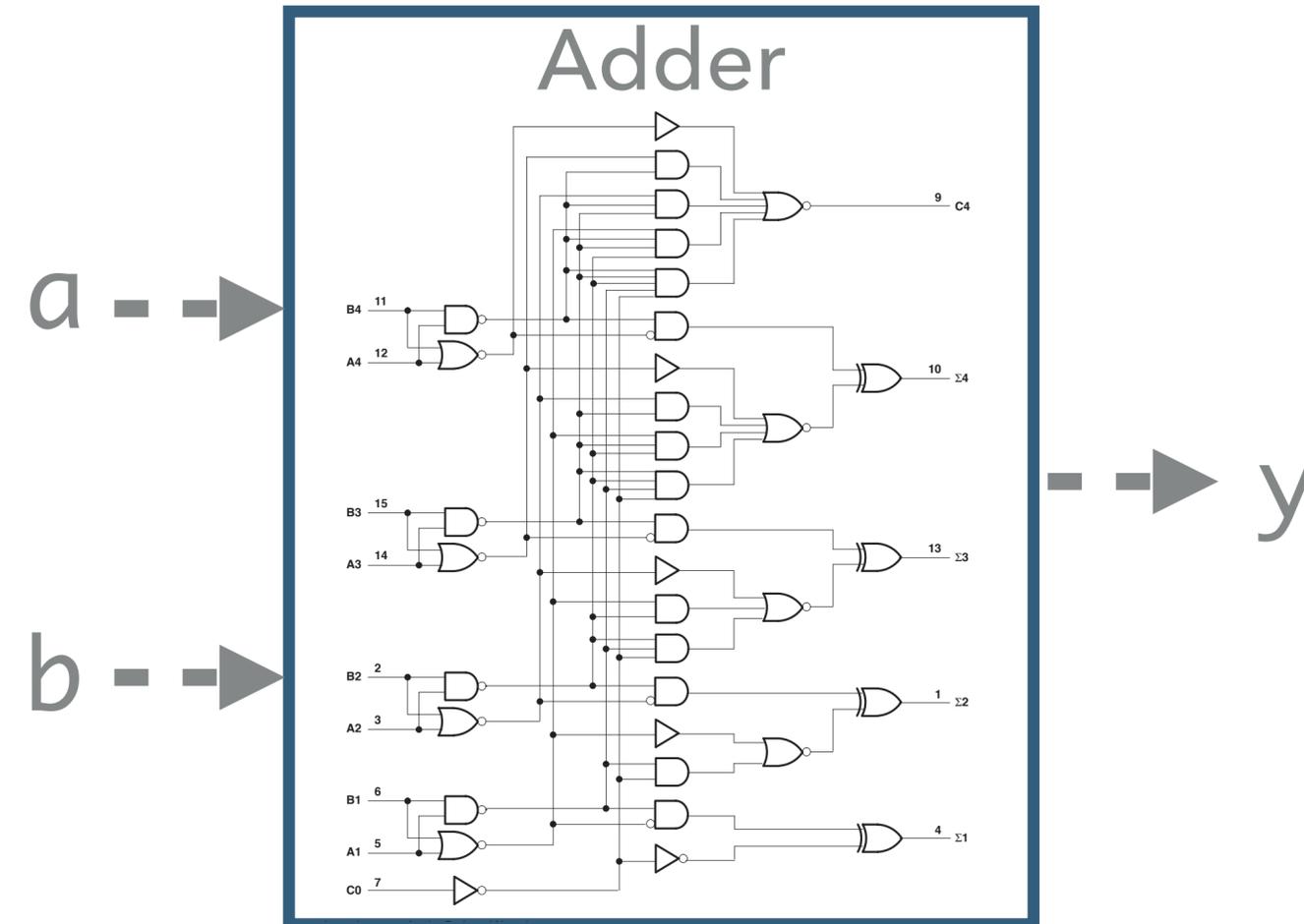
II. COMPRESSION

III. HARDWARE

IV. APPLICATIONS

- ▶ Say you want to program an "adder" function on an FPGA

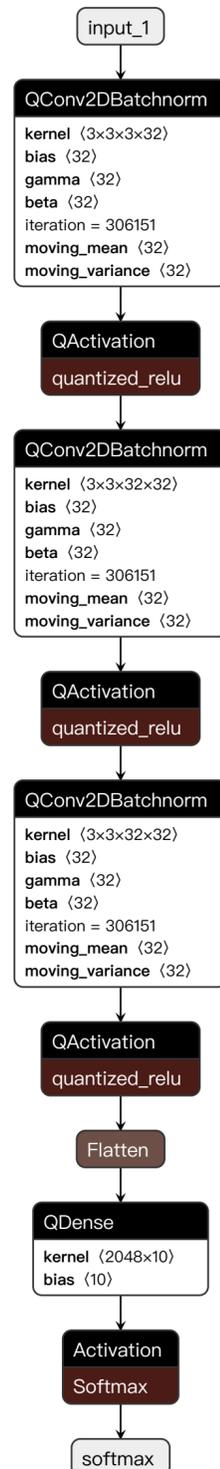
```
module adder(  
    input wire [4:0] a,  
    input wire [4:0] b,  
    output wire [4:0] y  
);  
    assign y = a + b;  
endmodule
```



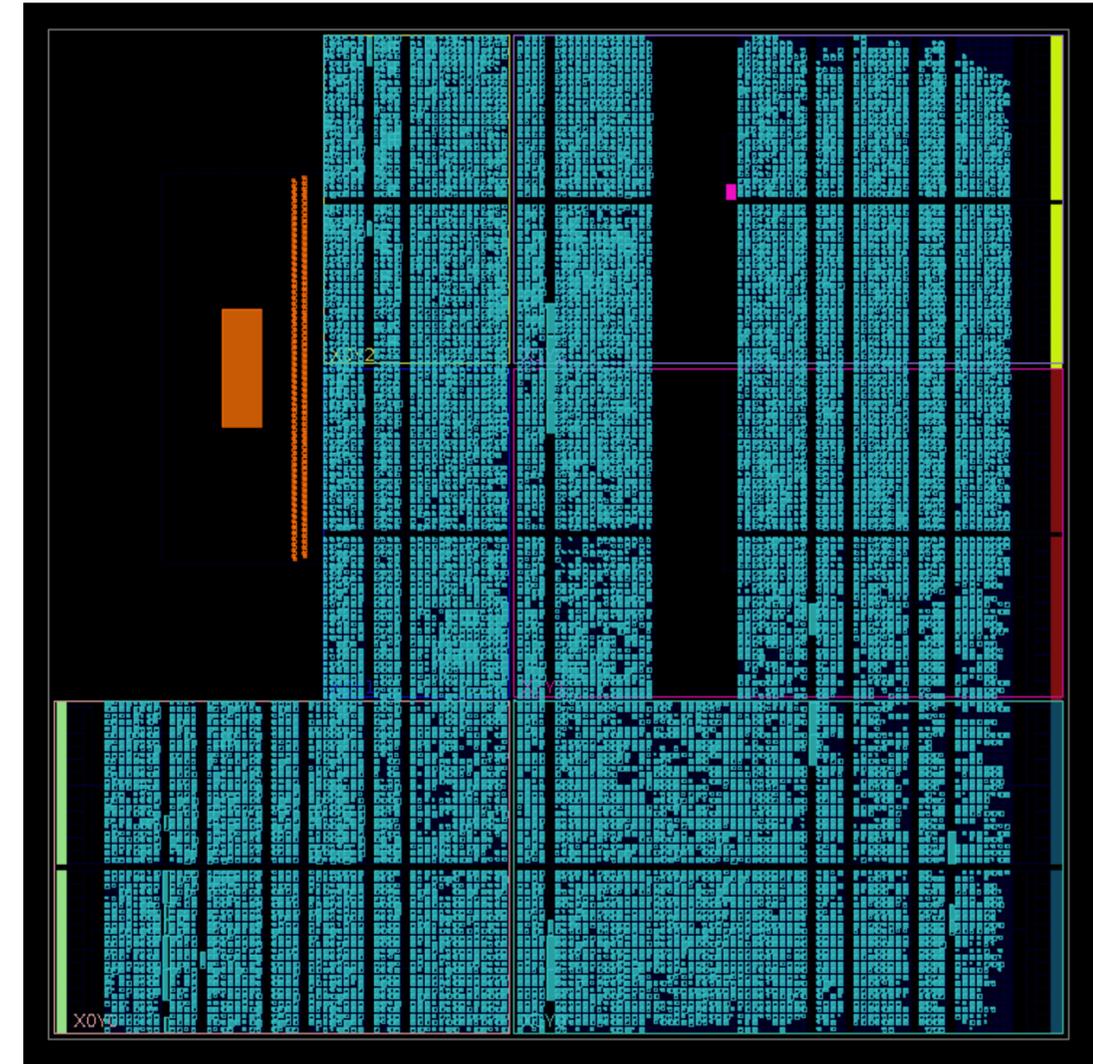
- ▶ Register transfer-level (RTL) code is "synthesized" into gates

Synthesis

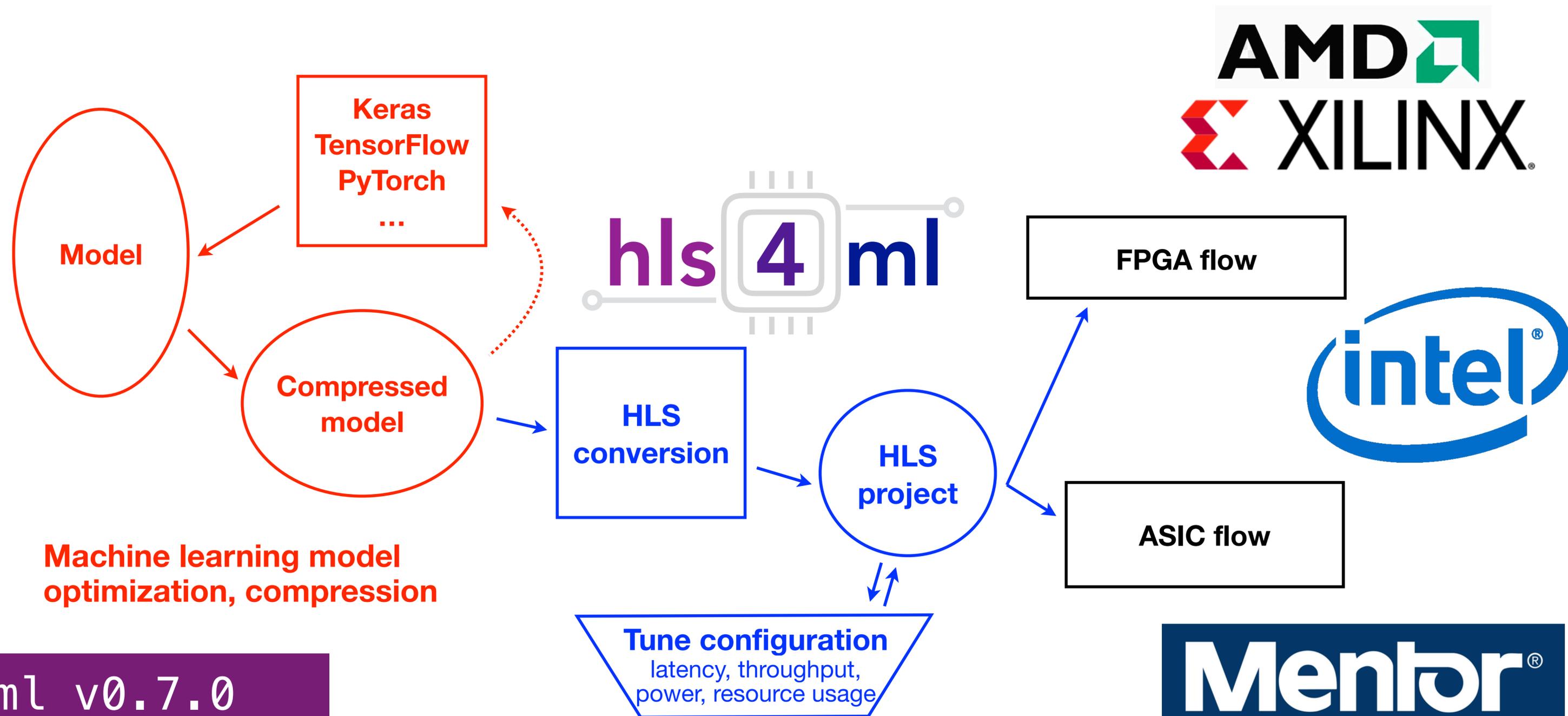
- ▶ What if instead we specify an AI model



High-Level Synthesis



- ▶ [hls4ml](#) for scientists or ML experts to translate ML algorithms into RTL firmware



hls4ml v0.7.0
coming this week!

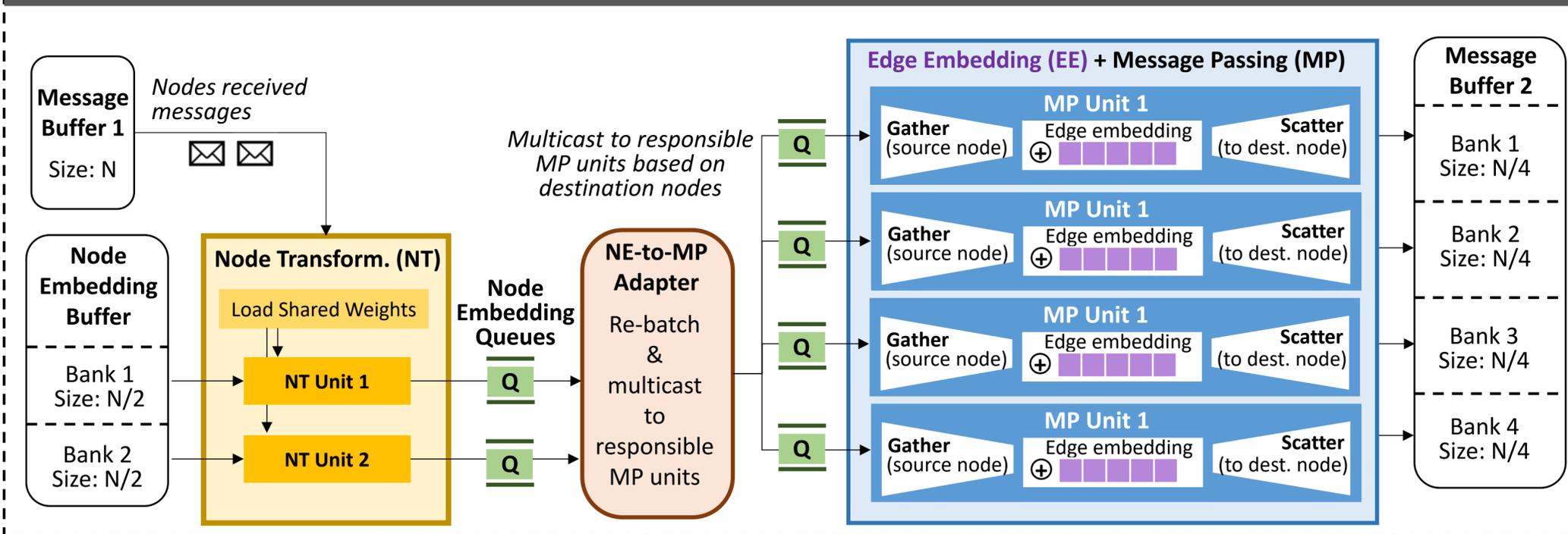


MANY TOOLS WITH DIFFERENT STRENGTHS

- ▶ FINN (NNs): <https://finn.readthedocs.io/en/latest/>
- ▶ Confier (BDTs): <https://github.com/thesps/conifer>
- ▶ fwXMachina (BDTs): <http://fwx.pitt.edu/>
- ▶ FlowGNN: <https://github.com/sharc-lab/flowgnn>



(b) FlowGNN Architecture with Multiple Node Transformation, Multiple Message Passing, and parallelized Edge Embedding



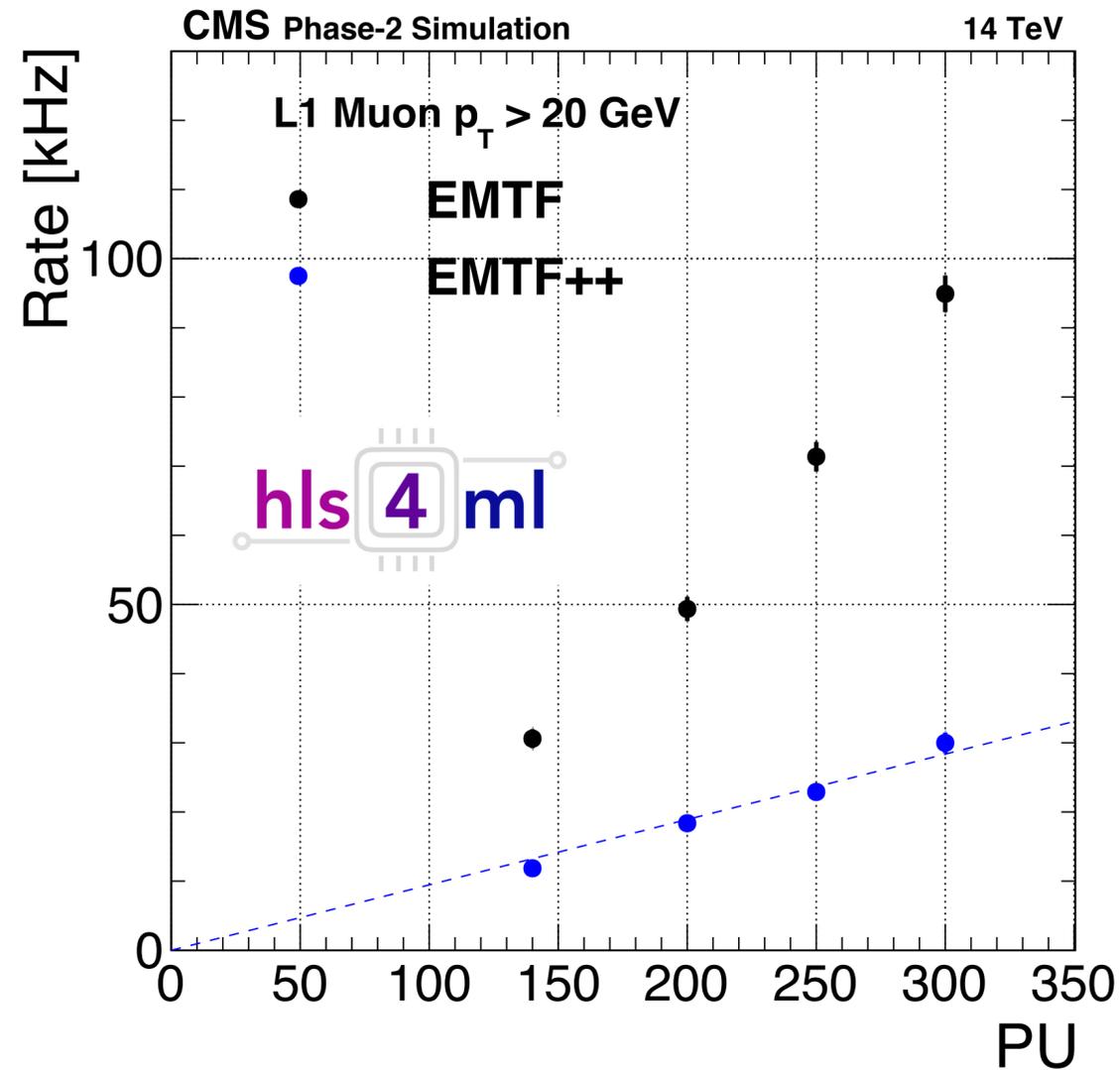
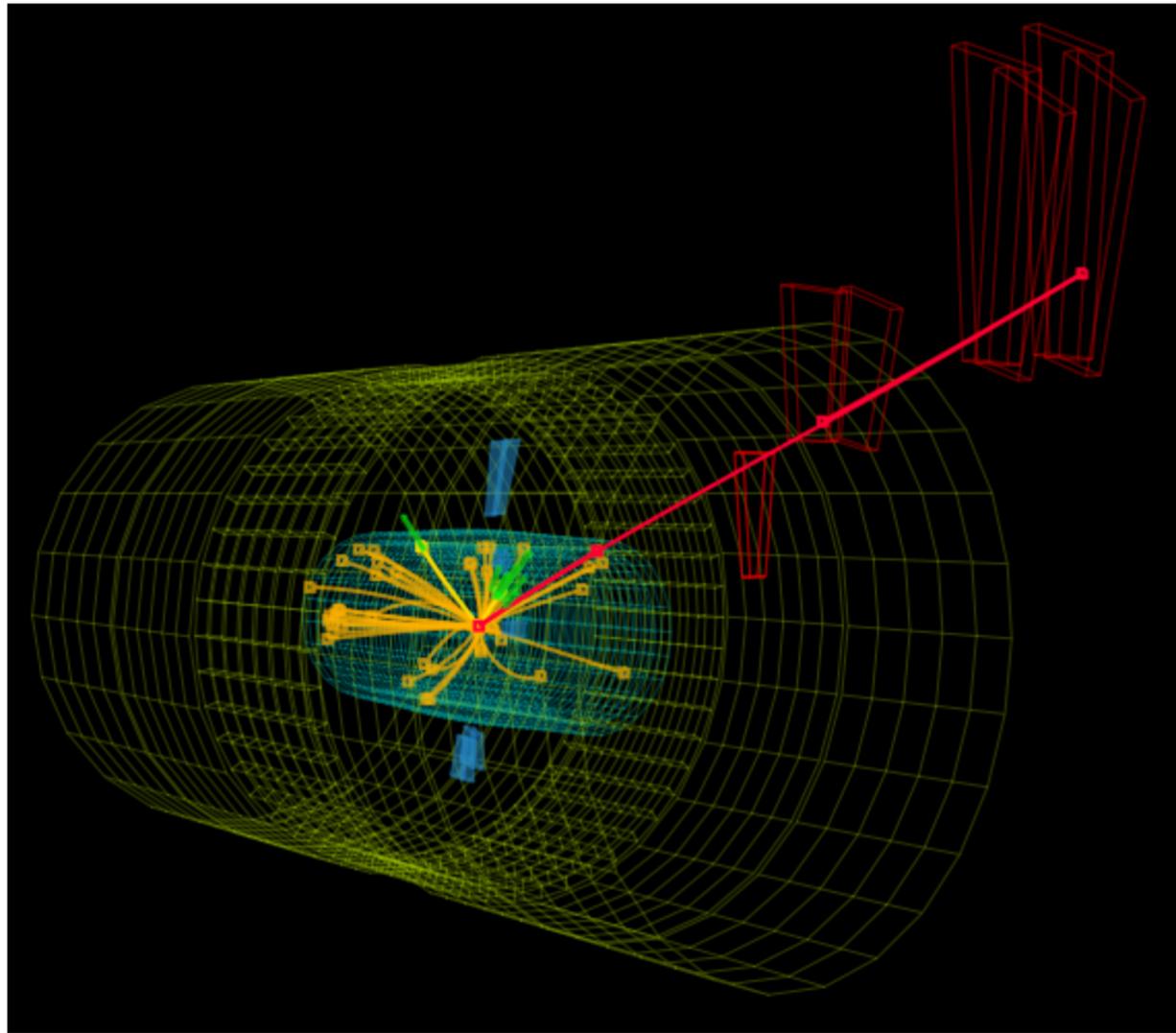
An aerial night view of a snowy mountain town, likely a ski resort, with lights illuminating the buildings and streets. The background shows snow-covered mountains under a dark blue sky.

I. INTRO & MOTIVATION

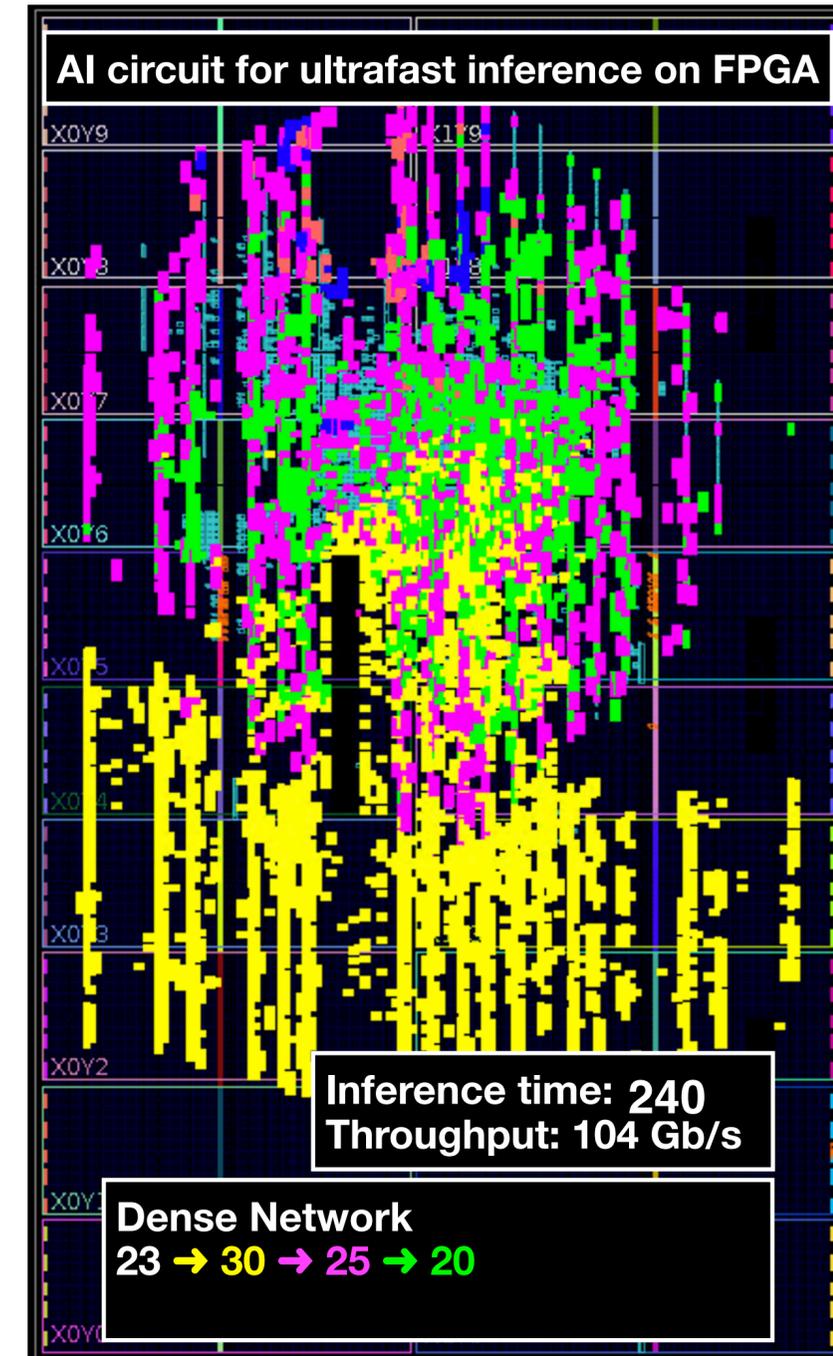
II. COMPRESSION

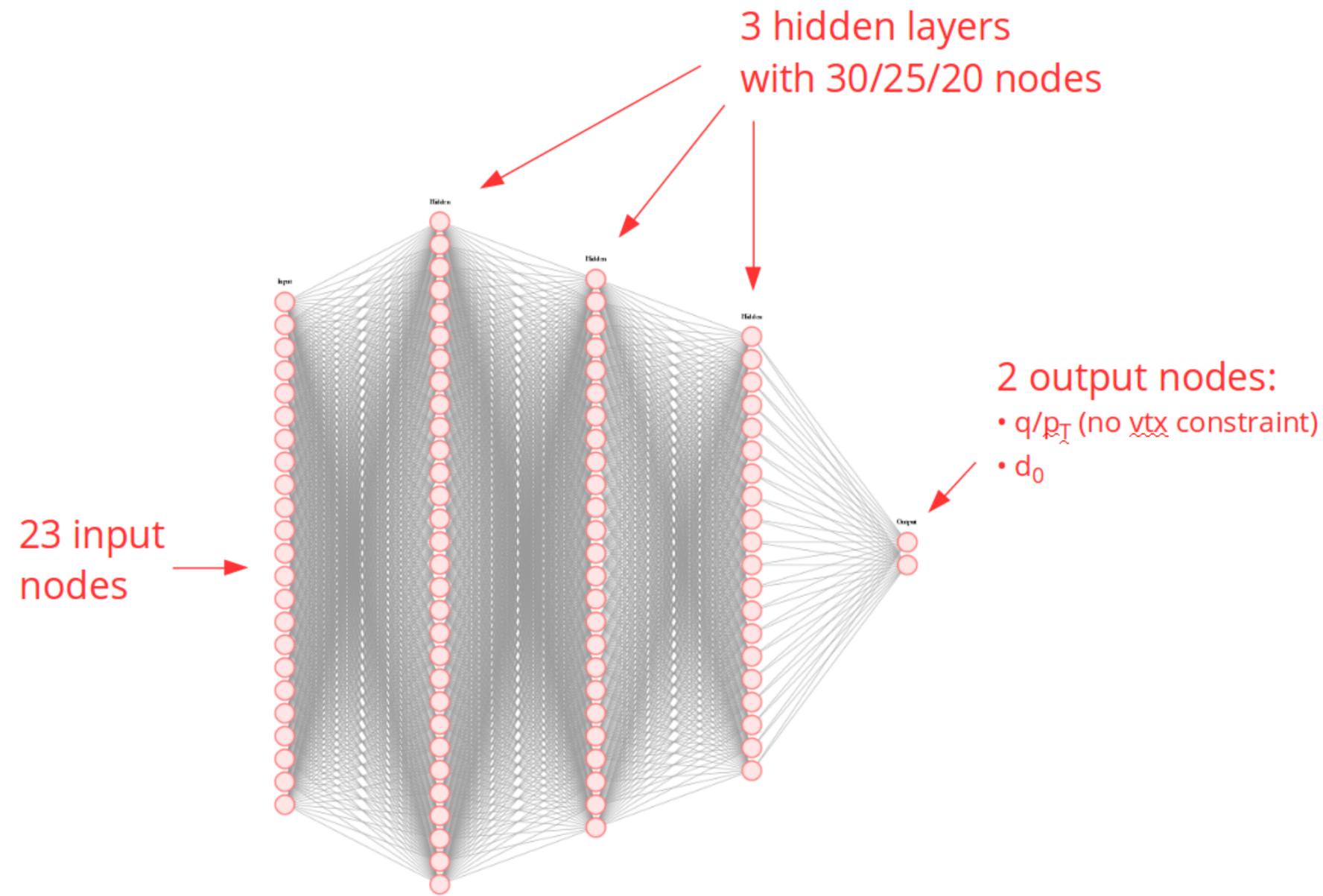
III. HARDWARE

IV. APPLICATIONS

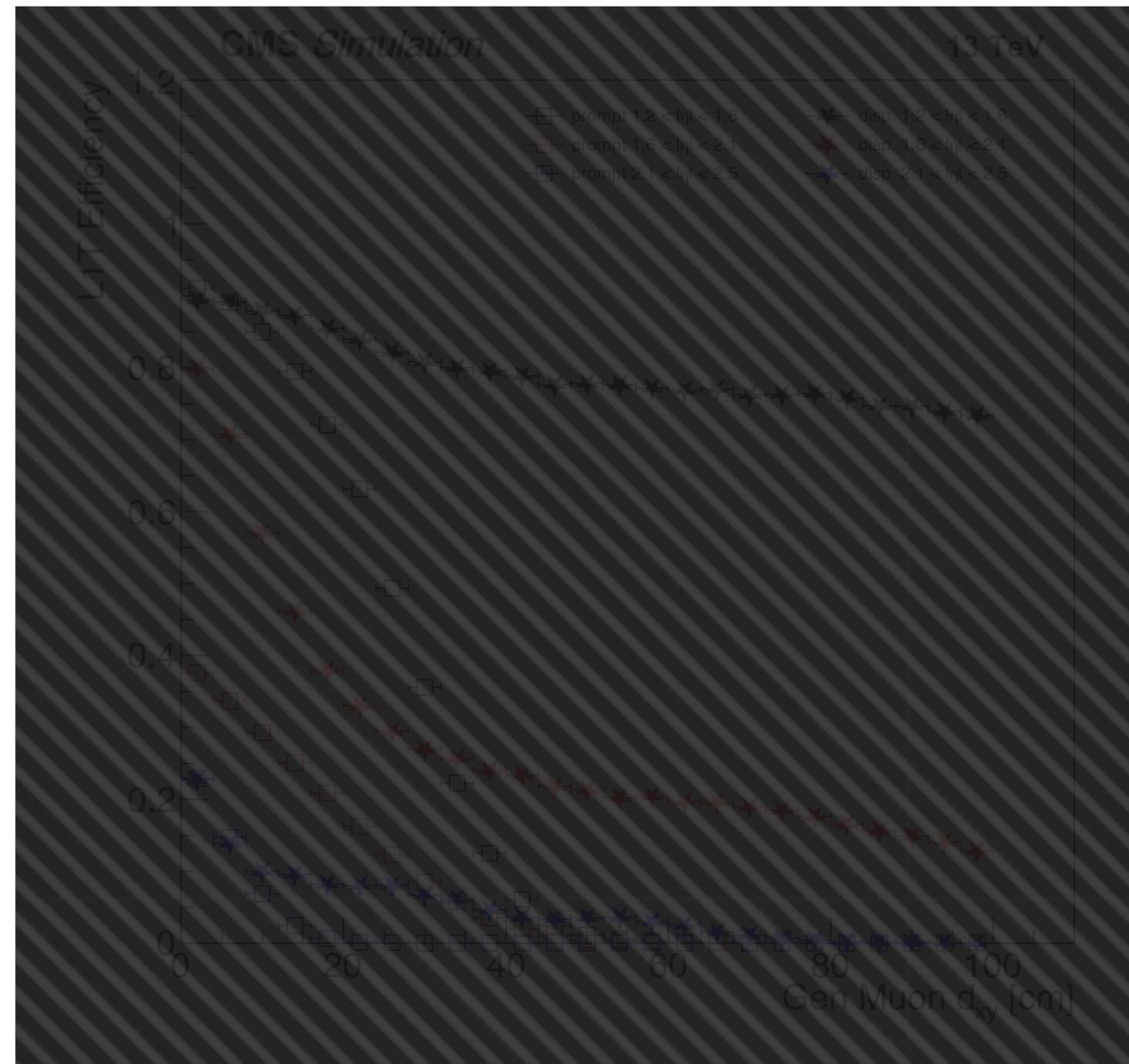


- ▶ NN measures muon momentum
 - ▶ 3× reduction in the trigger rate for NN!
- ▶ Fits within L1 trigger latency (240 ns!) and FPGA resource requirements (less than 30%)

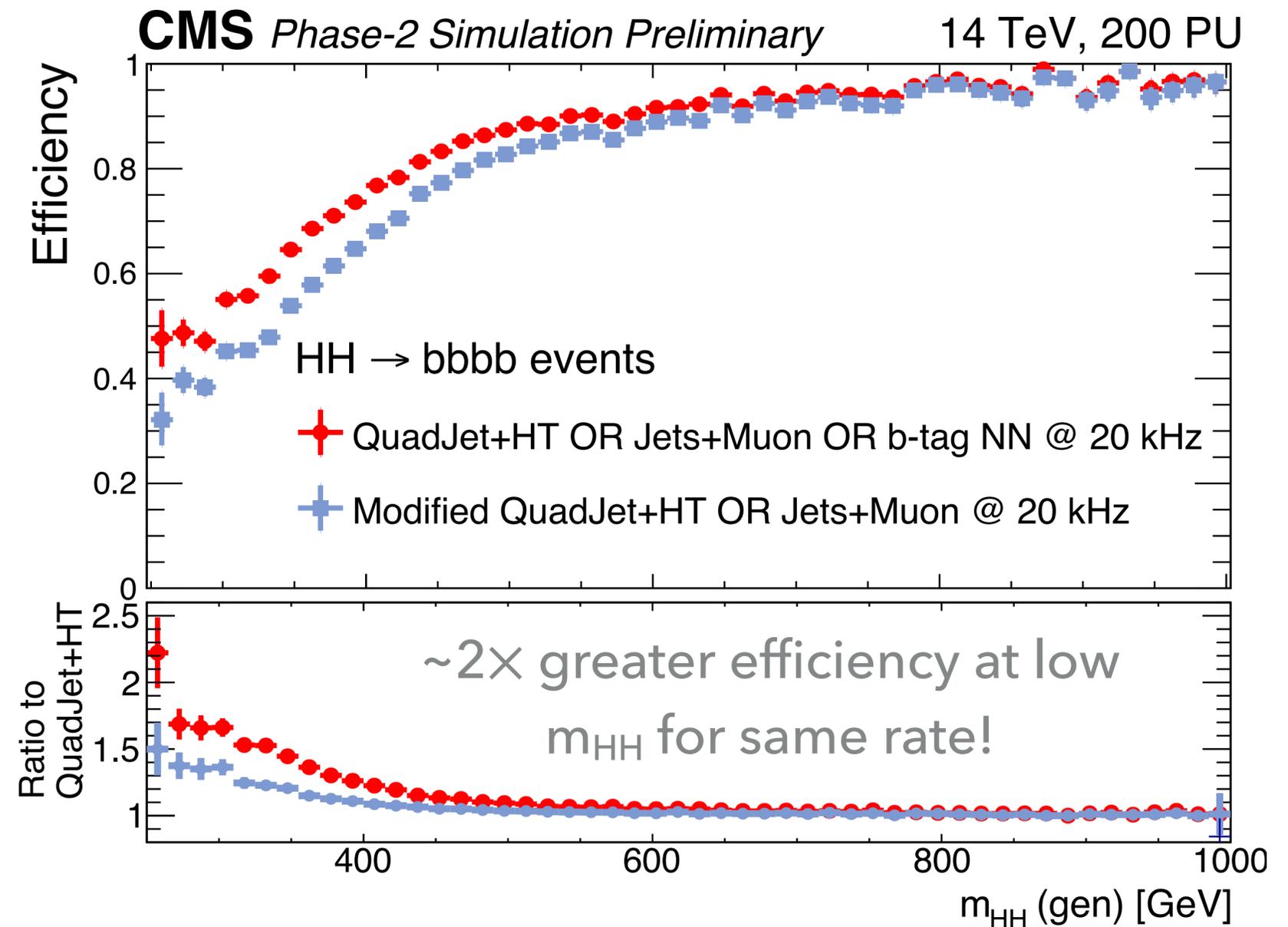
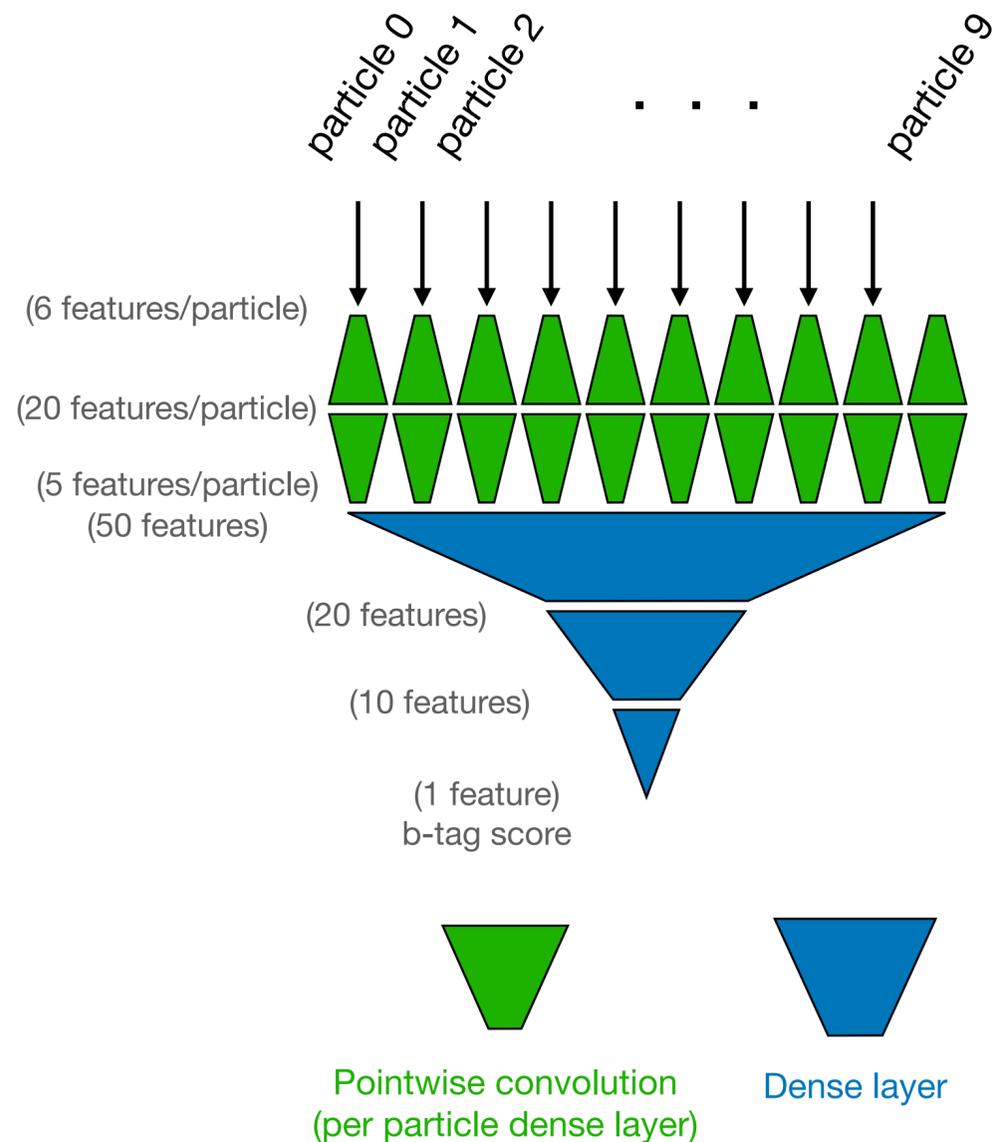




- ▶ Extends idea to measure muon displacement as well as p_T
- ▶ *Stay tuned for Run 3 results*

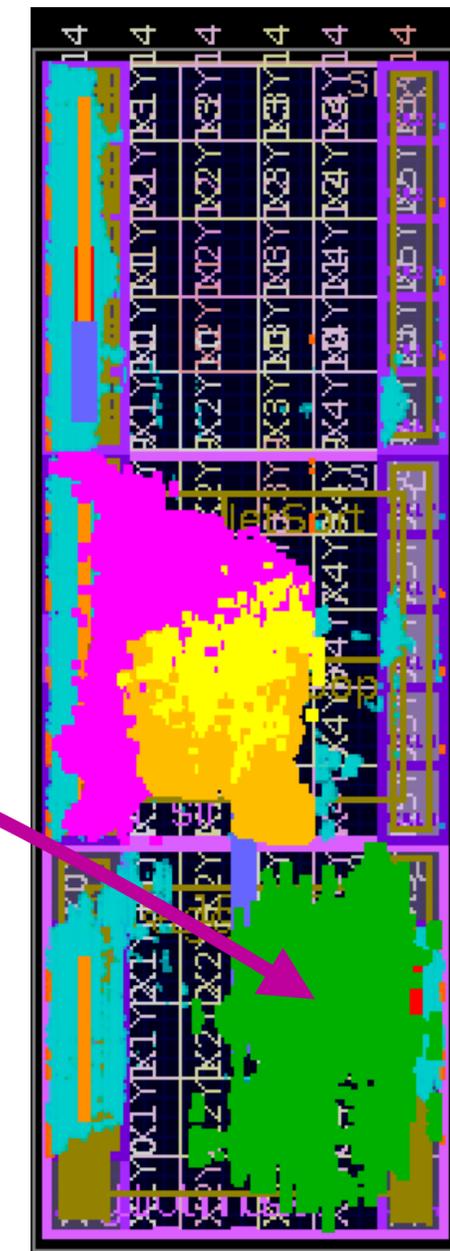
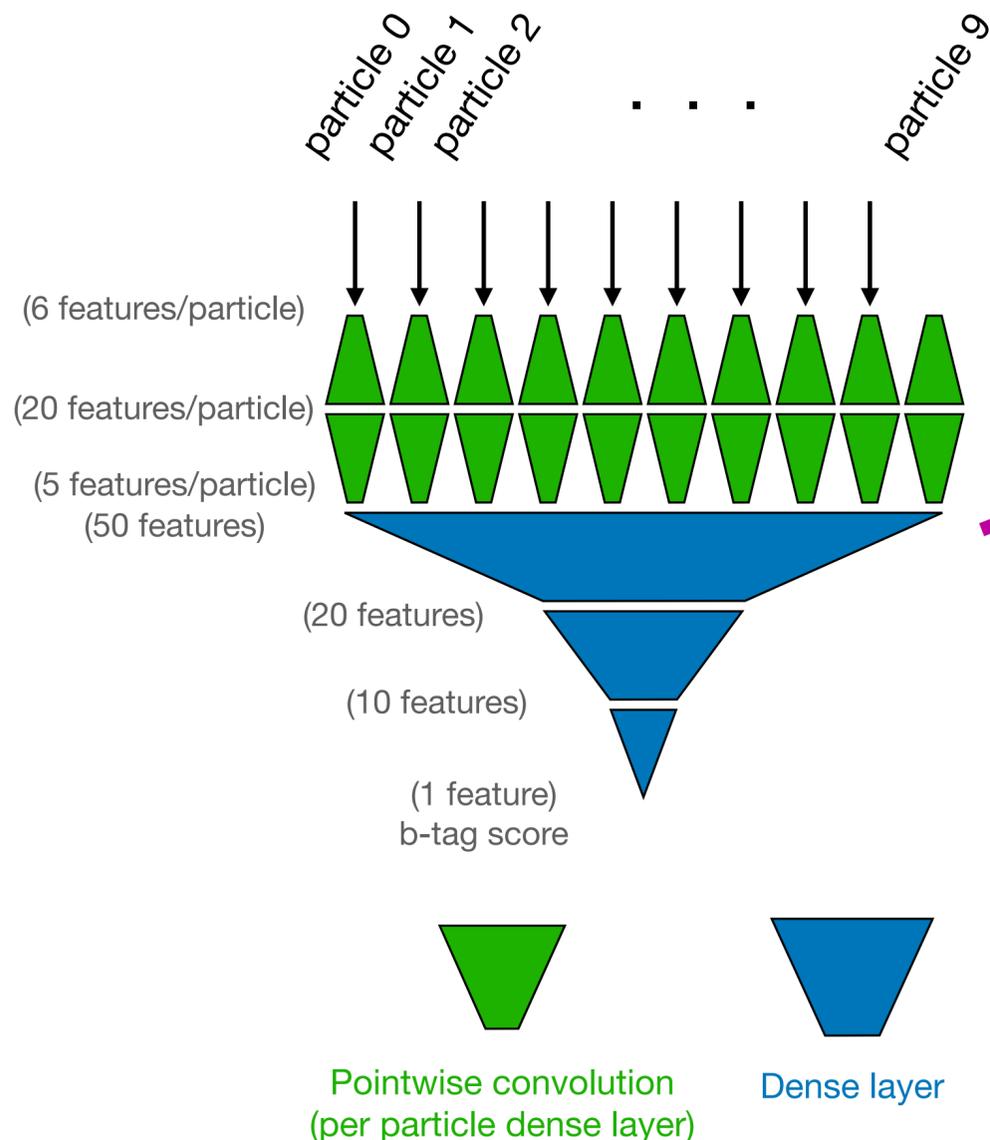
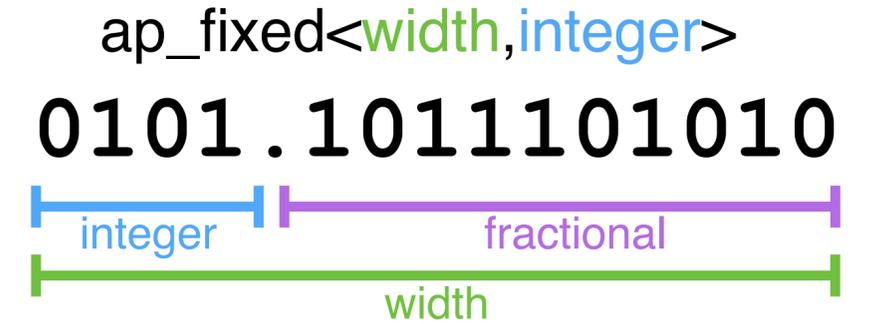


- ▶ Upgraded HL-LHC level-1 track trigger information enables b-tagging with a **neural network** to improve the $HH \rightarrow 4b$ search
- ▶ Input features for 10 particles within each jet: particle type, momentum, and vertex information

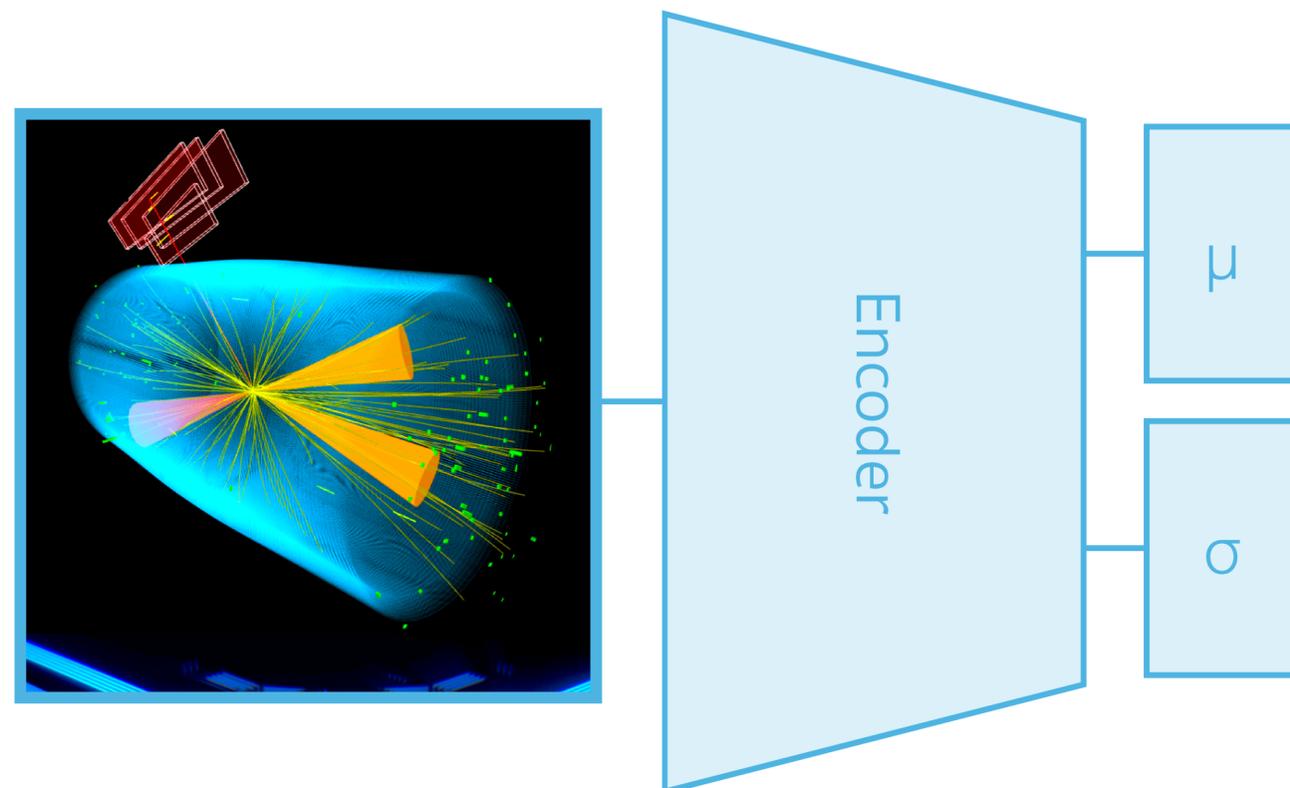


IMPLEMENTATION: B-TAGGING @ L1

- ▶ But does it fit and meet timing?
- ▶ After **quantization**, can implement NN with 9 bits
- ▶ Latency of 60 ns, II of 5 ns per jet, and <12% of FPGA



- ▶ Challenge: if new physics has an unexpected signature that doesn't align with existing triggers, precious BSM events may be discarded at trigger level
- ▶ Can we use unsupervised algorithms to detect non-SM-like anomalies?
 - ▶ Autoencoders (AEs): compress input to a smaller dimensional latent space then decompress and calculate difference
 - ▶ Variational autoencoders (VAEs): model the latent space as a probability distribution; possible to detect anomalies purely with latent space variables



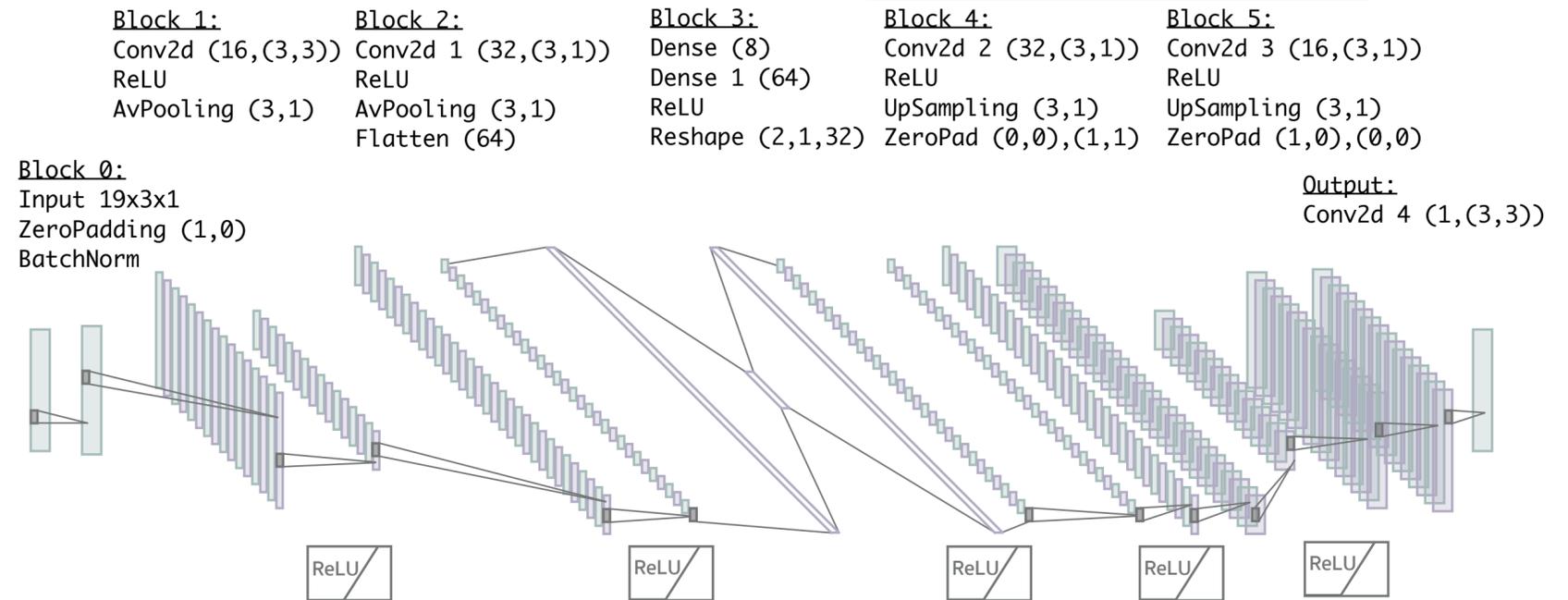
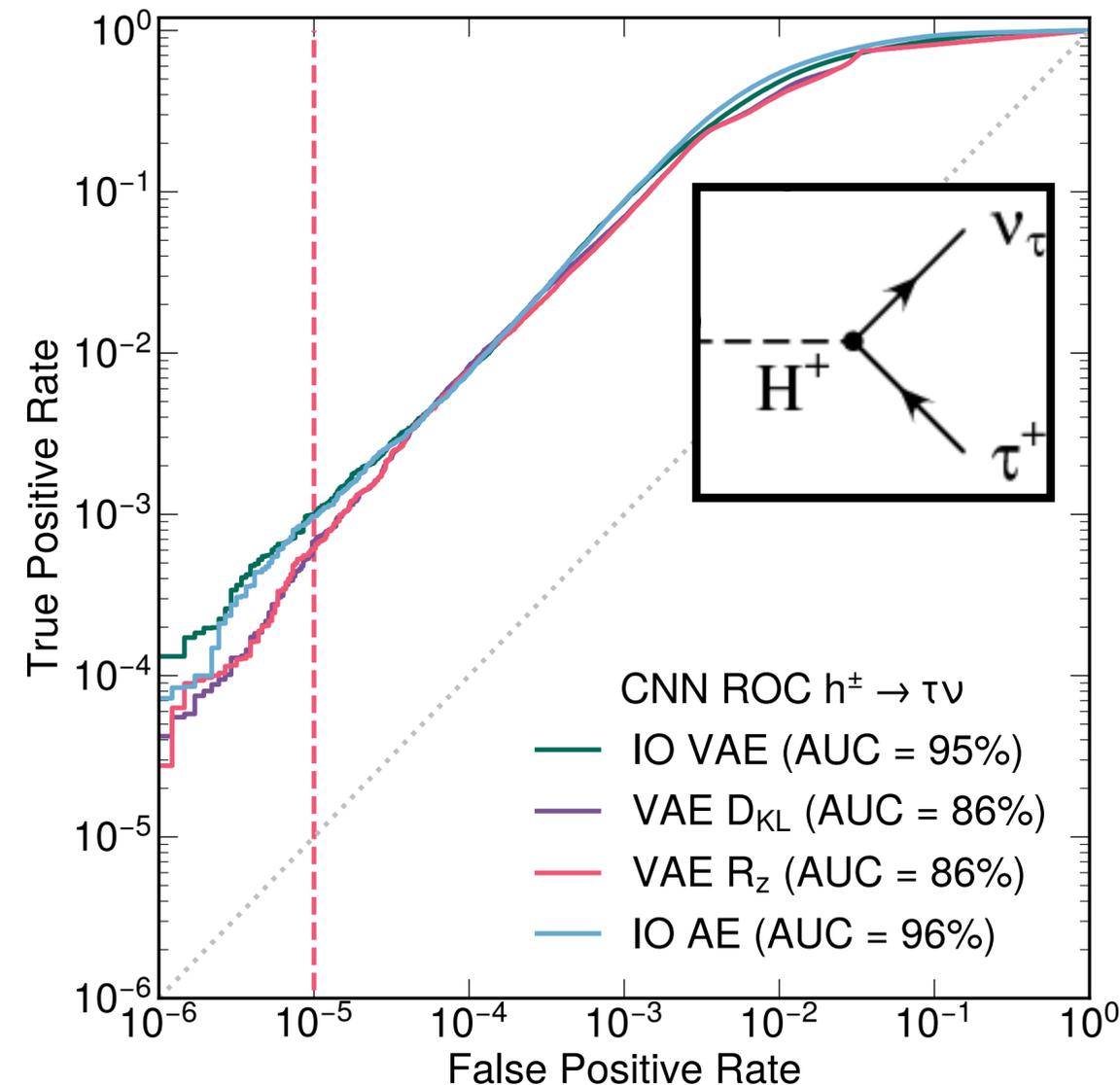
Key observation: Can build an anomaly score from the latent space of VAE directly!
No need to run decoder!

$$R_z = \sum_i \frac{\mu_i^2}{\sigma_i^2}$$

- ▶ CNNs as the basis for (V)AEs for anomaly detection
- ▶ Good anomaly detection performance for unseen signals ($LQ \rightarrow b\tau, A \rightarrow 4l, h^\pm \rightarrow \tau\nu, h^0 \rightarrow \tau\tau$)
- ▶ **VAE** fits in latency and resource requirements for HL-LHC!

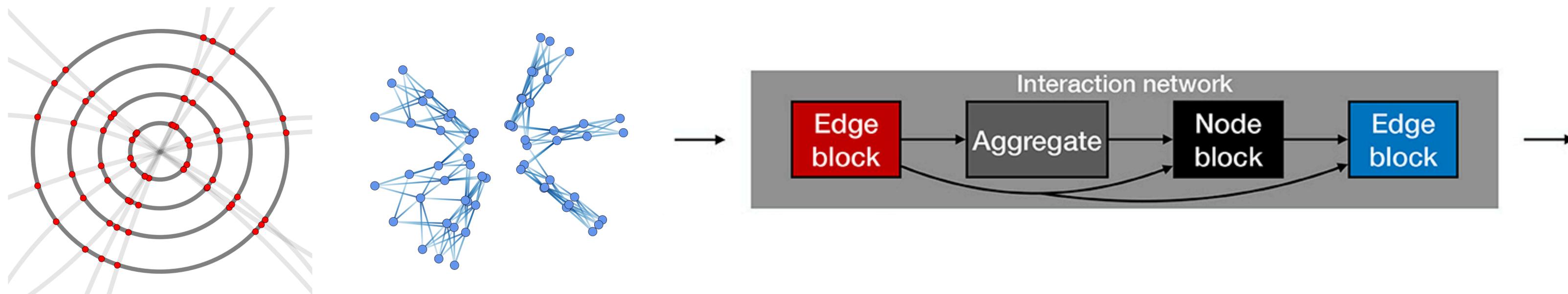


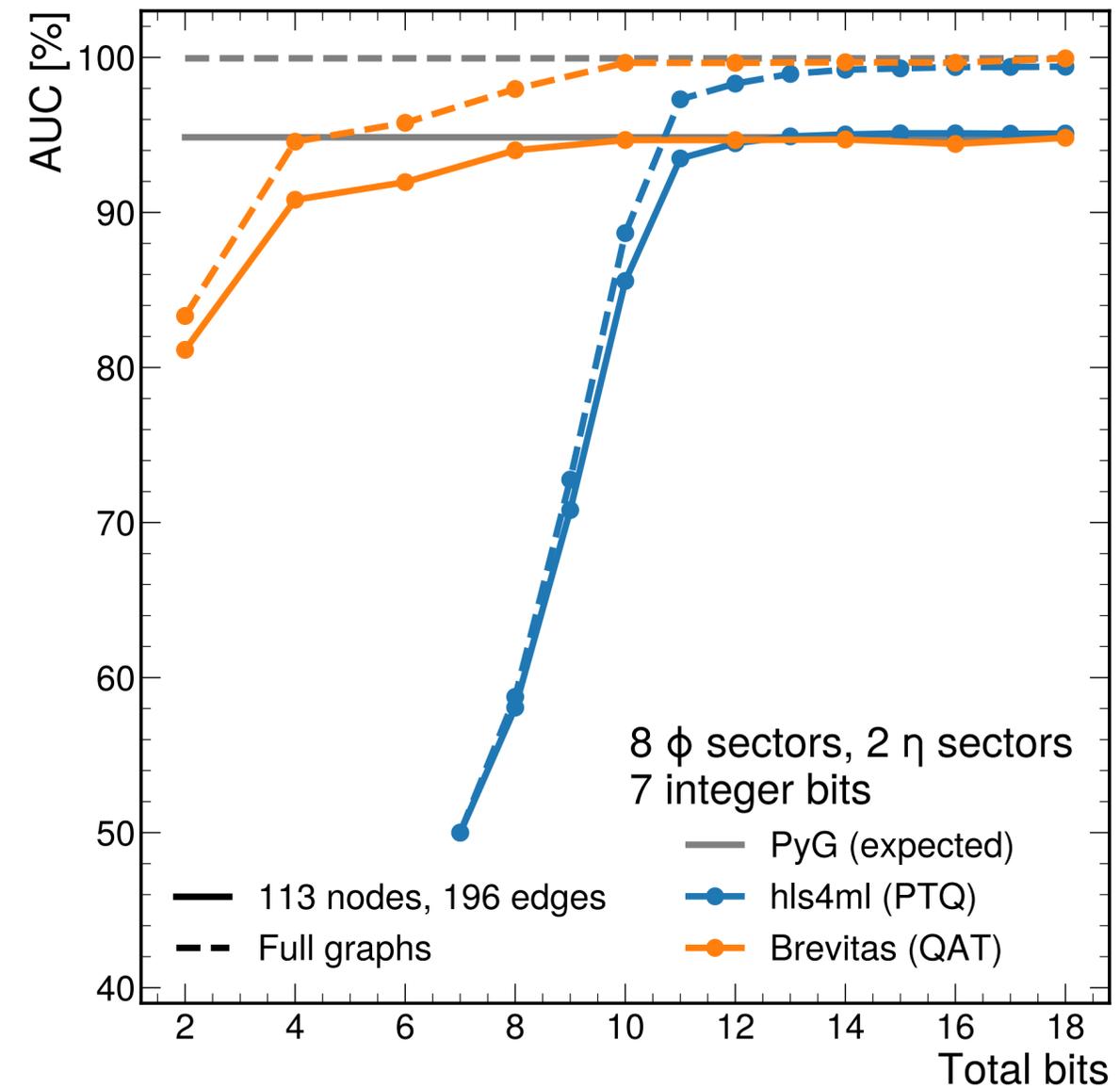
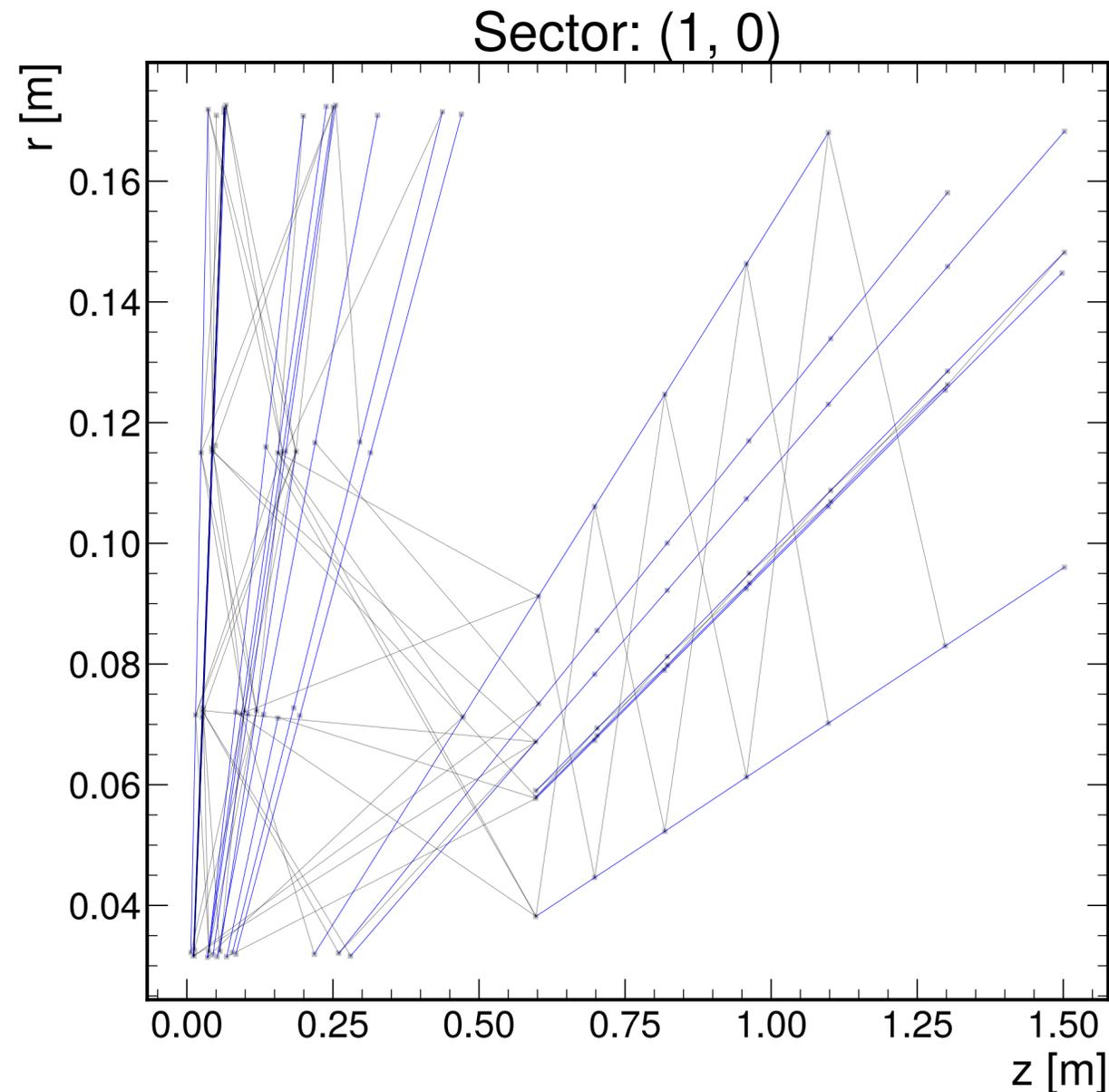
Stay tuned for Run 3...



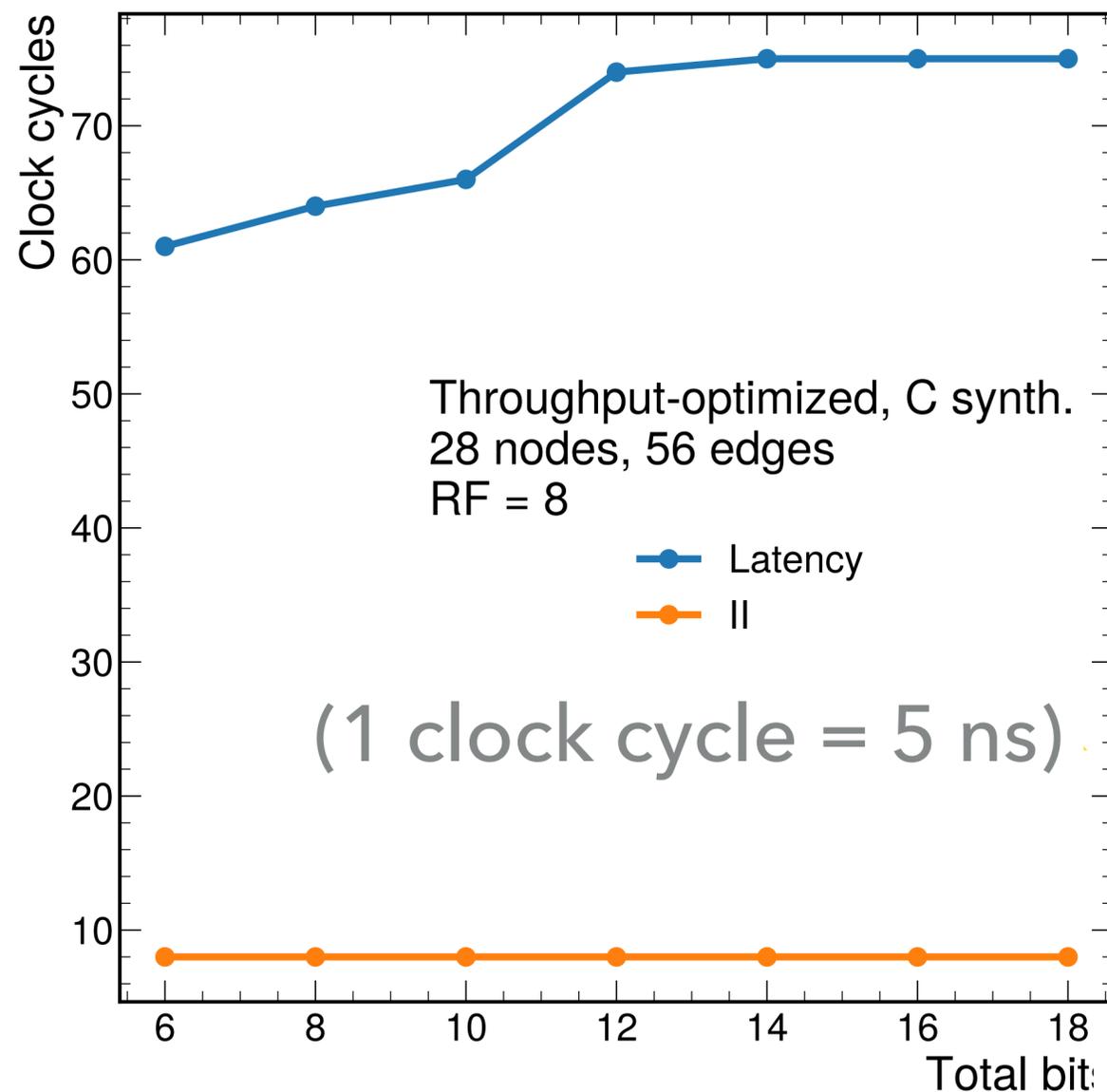
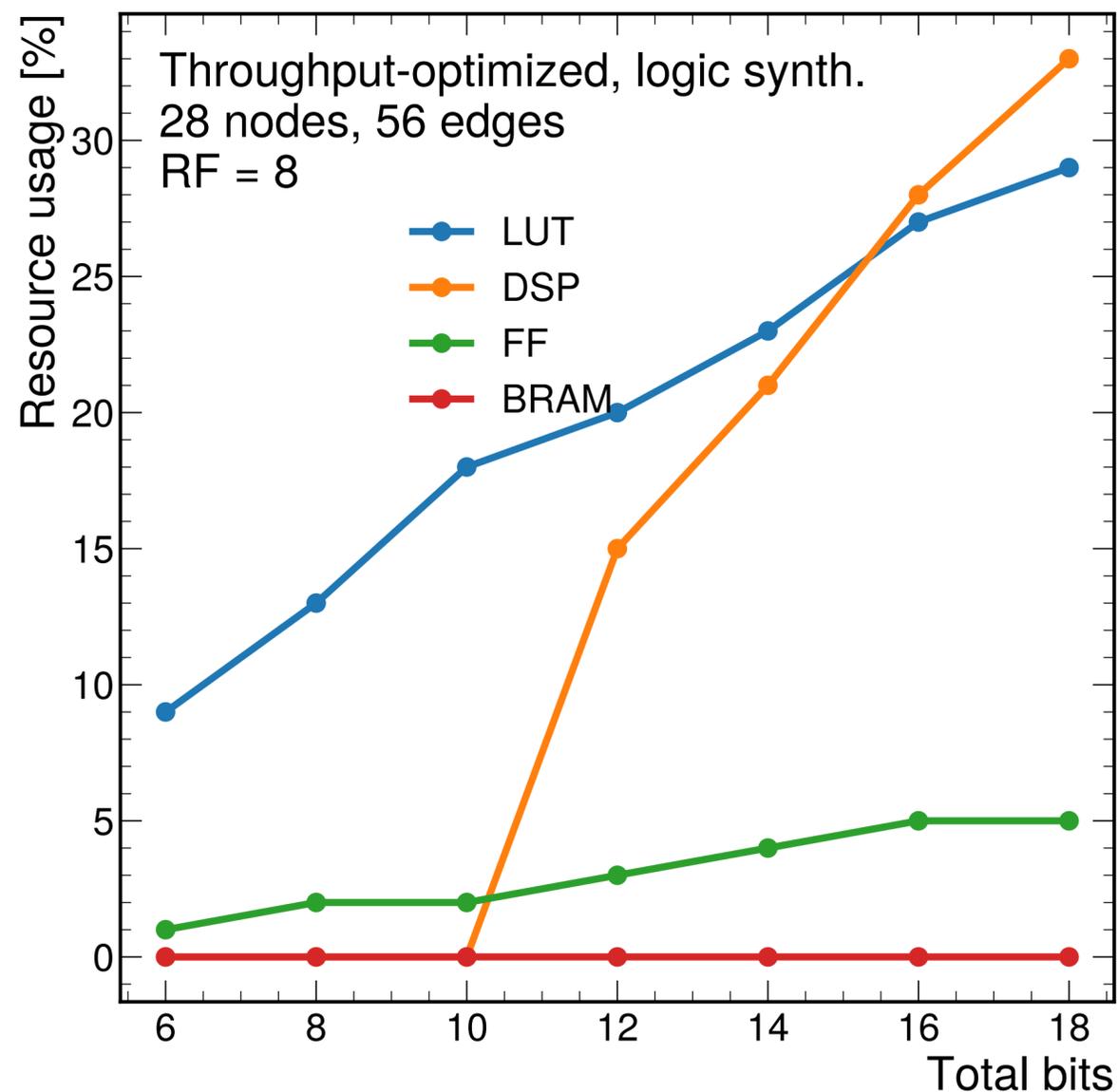
Model	DSP [%]	LUT [%]	FF [%]	BRAM [%]	Latency [ns]	II [ns]	AUC [%]	TPR @ FPR=10 ⁻⁵
CNN VAE								
R_z	10	12	4	2	365	115	86	0.06%

- ▶ Traditional tracking algorithms scale quadratically with the number of hits
- ▶ New algorithms (based on **graph neural networks**) may be able to do better
- ▶ Proof of concept study: use GNN to classify good track segments (edges)
 - ▶ Can this fit on an FPGA?

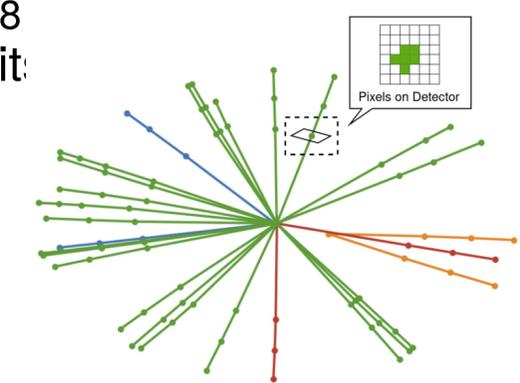
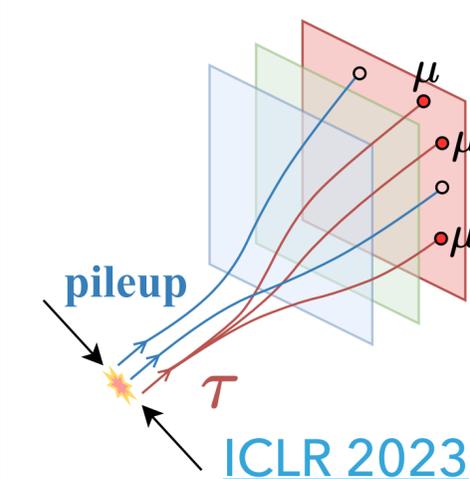




- ▶ Build realistic (segmented) graphs for L1 trigger applications
- ▶ ≤ 8 -bit quantized GNN can achieve good edge classification performance

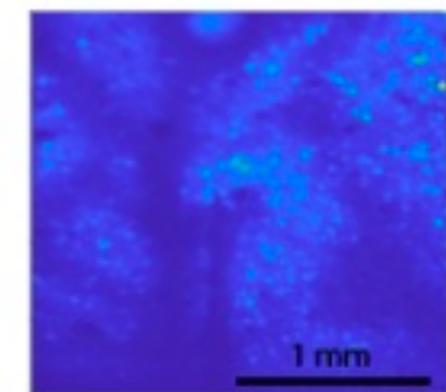
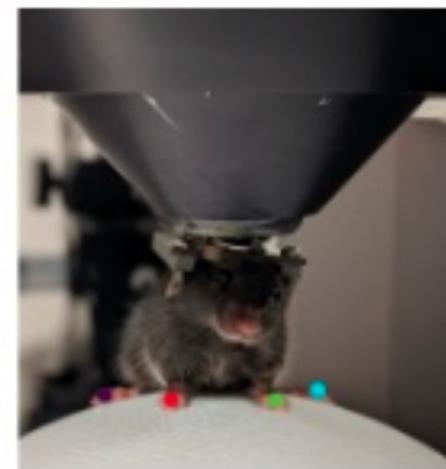
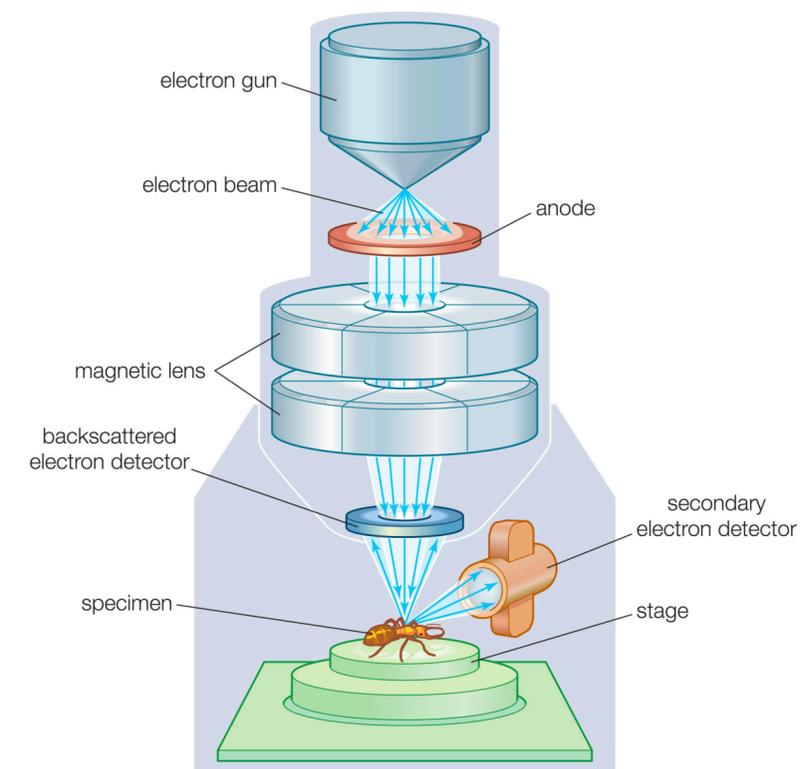
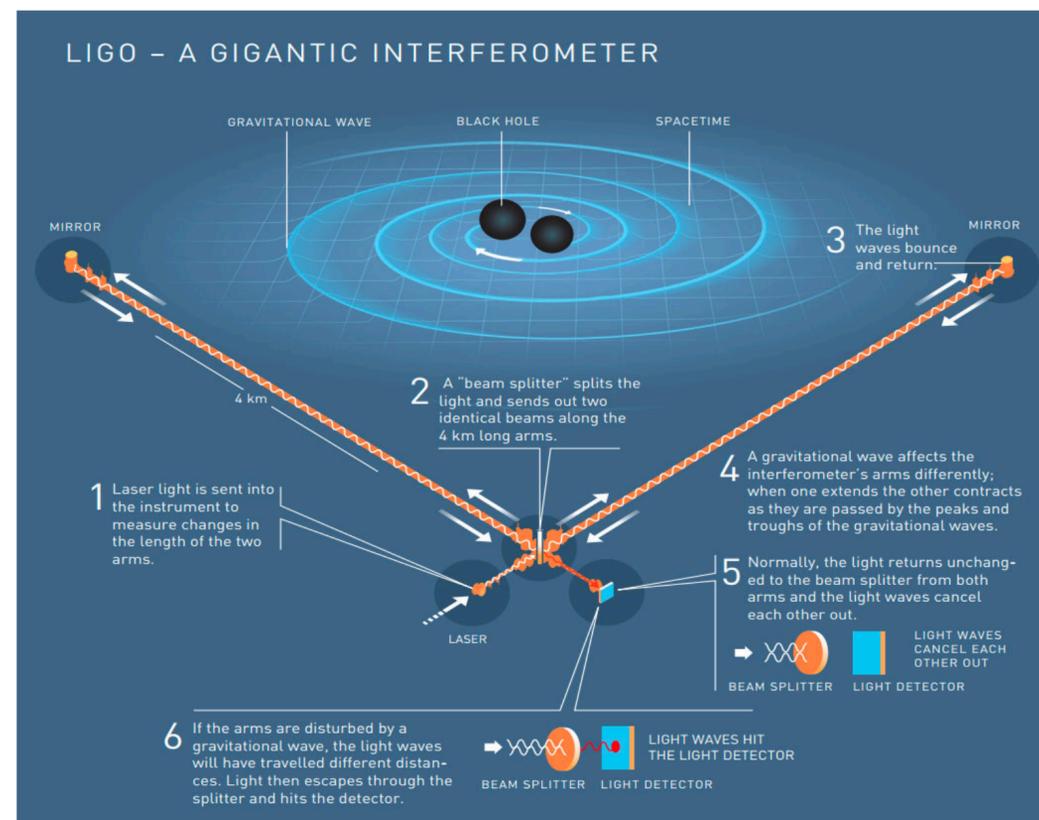
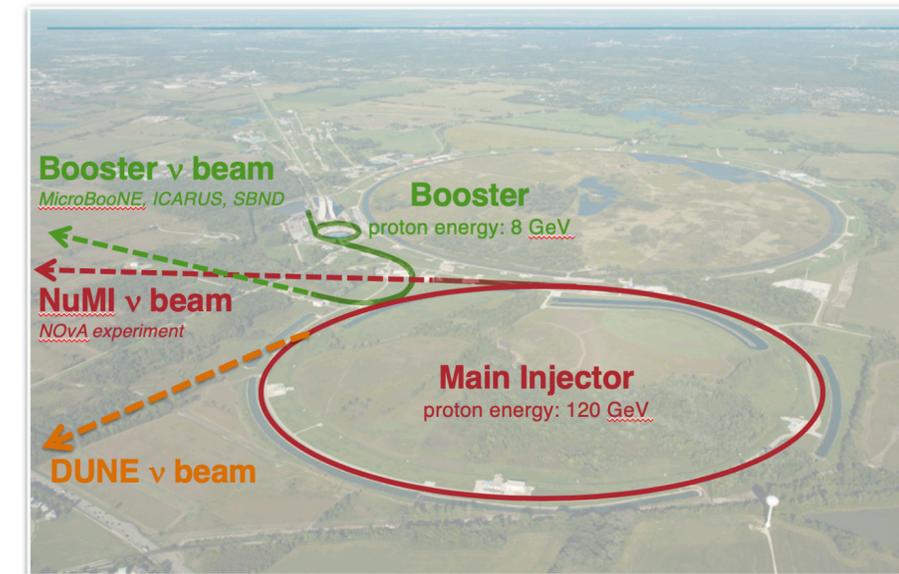
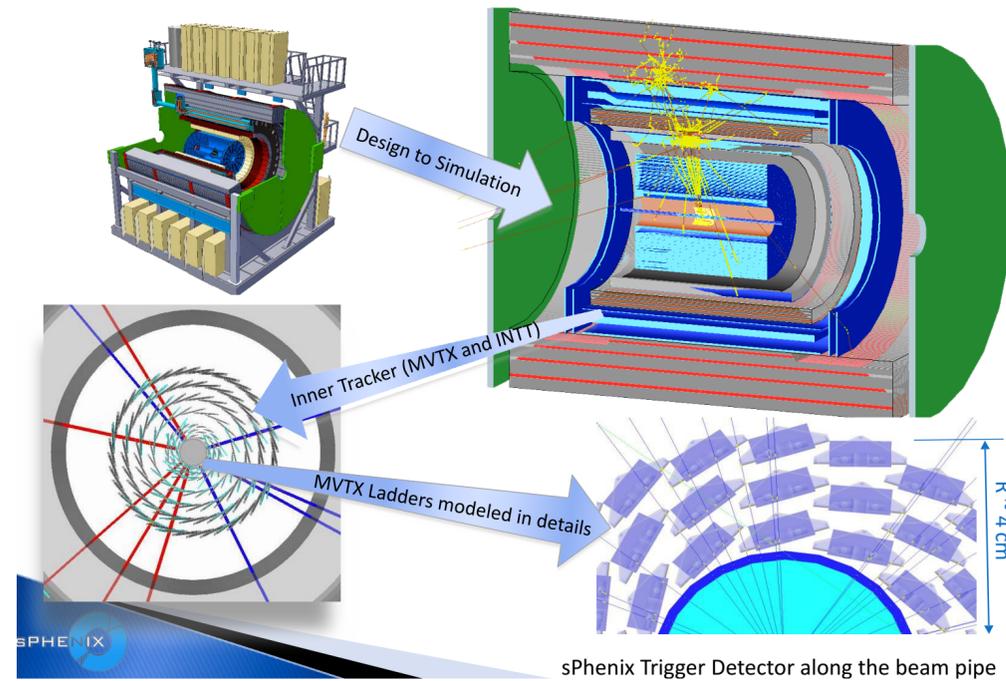


Similar algorithms for $\tau \rightarrow 3\mu$ @ LHC and tracking in sPHENIX

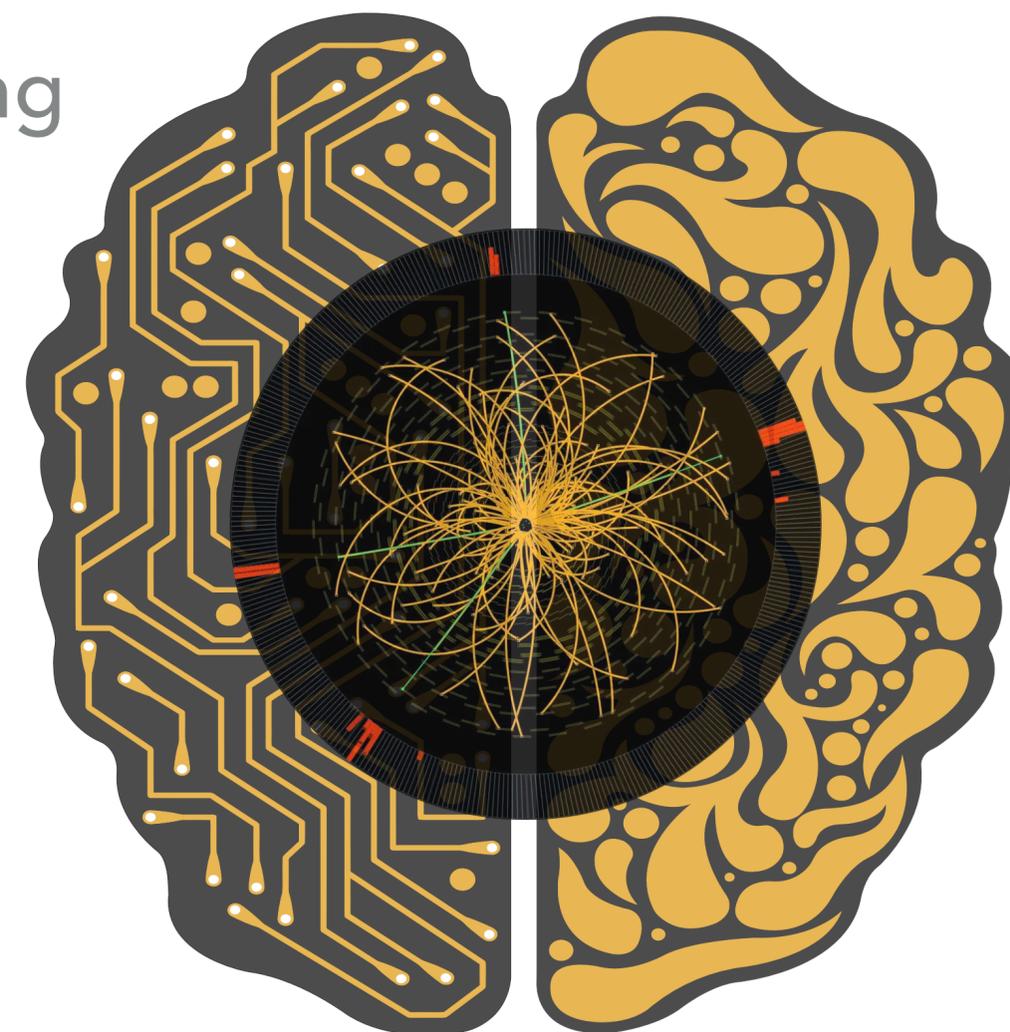


- ▶ Small graphs (~30 nodes, ~60 edges) easily fit on 1 FPGA
- ▶ Within L1T latency (300 ns) and II (50 ns) requirements

- ▶ Nuclear physics
- ▶ Accelerator control
- ▶ Neutrino physics
- ▶ Multi-messenger astronomy
- ▶ Electron & X-ray microscopy
- ▶ Neuroscience



- ▶ ML allows us to better reconstruct our data and save potentially overlooked data
- ▶ **Codesign** principles can enable ML on hardware with stringent constraints
- ▶ Community (fastmachinelearning.org, e-group hls-fml@cern.ch) and Institute (a3d3.ai) developing open-source tools and techniques to enable this
 - ▶ [hls4ml](#): expanding open-source toolkit for translating ML into hardware aimed at trigger applications and more...
- ▶ Applications range from momentum regression, to b-tagging, tracking, and more!
 - ▶ Enhance **future particle physics program**

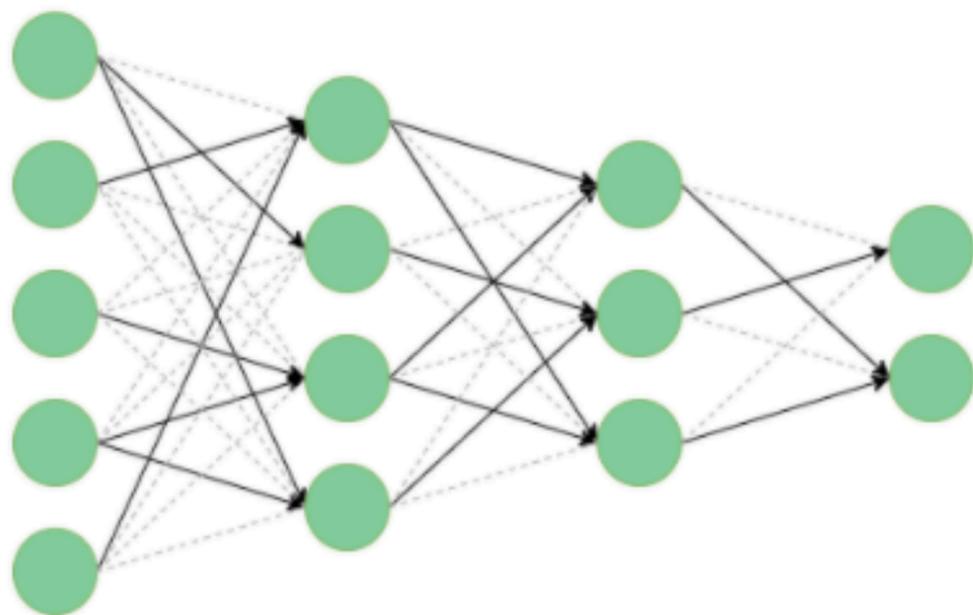




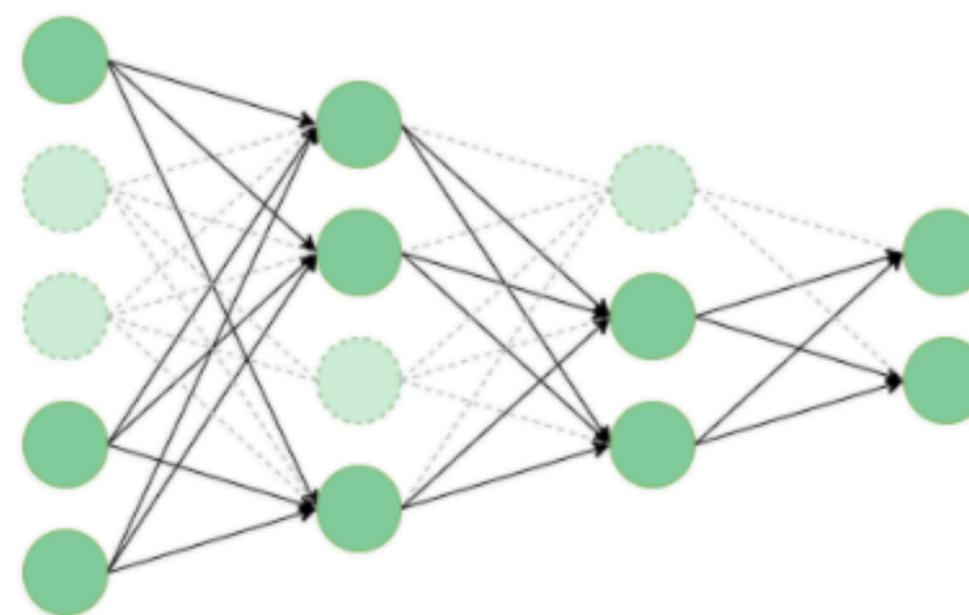
BACKUP

- ▶ Unstructured pruning: removing some connections regardless of placement
- ▶ Structured pruning: removing all input/output connections of particular nodes

Unstructured Pruning

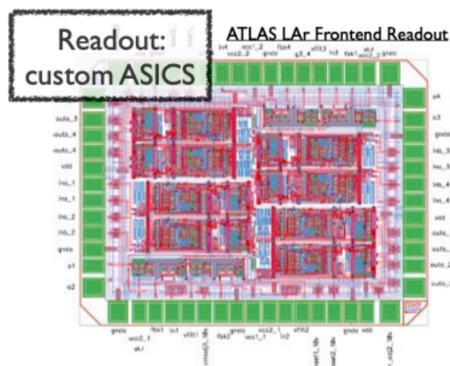


Structured Pruning



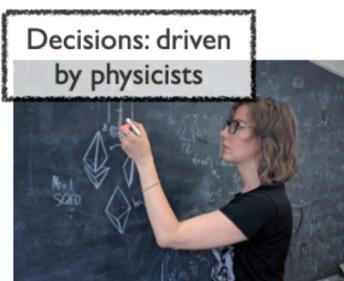
▶ Excellent overview talks for reference

Why Fast ML



Machine learning has hugely impacted analysis at the LHC: cornerstone of our work now

The challenge of the HL-LHC **requires** us to revise the entire data-flow pipeline



Hugely increased complexity of events: machine learning can help address every aspect!

ORGANIZING COMMITTEE

MOHAMMED ABOELELA
SOUTHERN METHODIST UNIVERSITY

SUSAN BATAJU
SOUTHERN METHODIST UNIVERSITY

THOMAS COAN
SOUTHERN METHODIST UNIVERSITY

ALLISON MCCARN DEIANA
SOUTHERN METHODIST UNIVERSITY

JASMINE JENNINGS
SOUTHERN METHODIST UNIVERSITY

FRED OLNESS
SOUTHERN METHODIST UNIVERSITY

SANTOSH PARAJULI
SOUTHERN METHODIST UNIVERSITY

REAGAN THORNBERRY
SOUTHERN METHODIST UNIVERSITY

SCIENTIFIC COMMITTEE

ALLISON MCCARN DEIANA
SOUTHERN METHODIST UNIVERSITY

JAVIER DUARTE
UNIVERSITY OF CALIFORNIA SAN DIEGO

PHIL HARRIS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SCOTT HAUCK
UNIVERSITY OF WASHINGTON

BURT HOLZMAN
FERMI NATIONAL ACCELERATOR LABORATORY

SHIH-CHIEH HSU
UNIVERSITY OF WASHINGTON

SERGO JINDARIANI
FERMI NATIONAL ACCELERATOR LABORATORY

MIA LIU
PURDUE UNIVERSITY

MARK NEUBAUER
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

MAURIZIO PIERINI
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH (CERN)

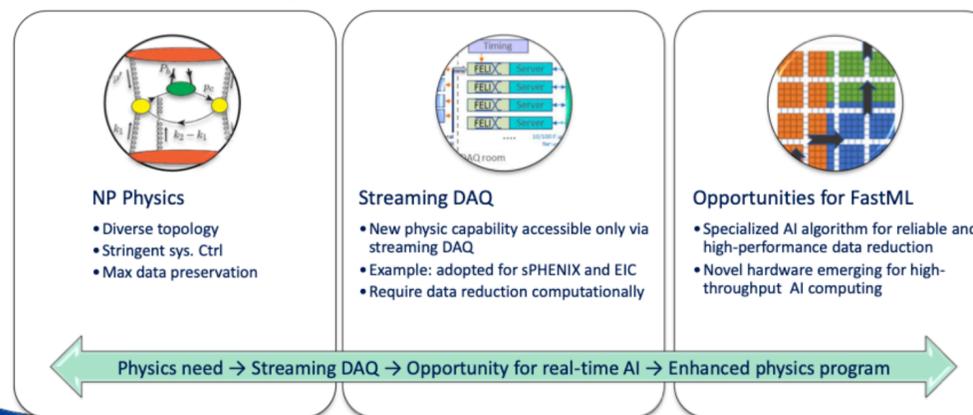
NHAN TRAN
FERMI NATIONAL ACCELERATOR LABORATORY

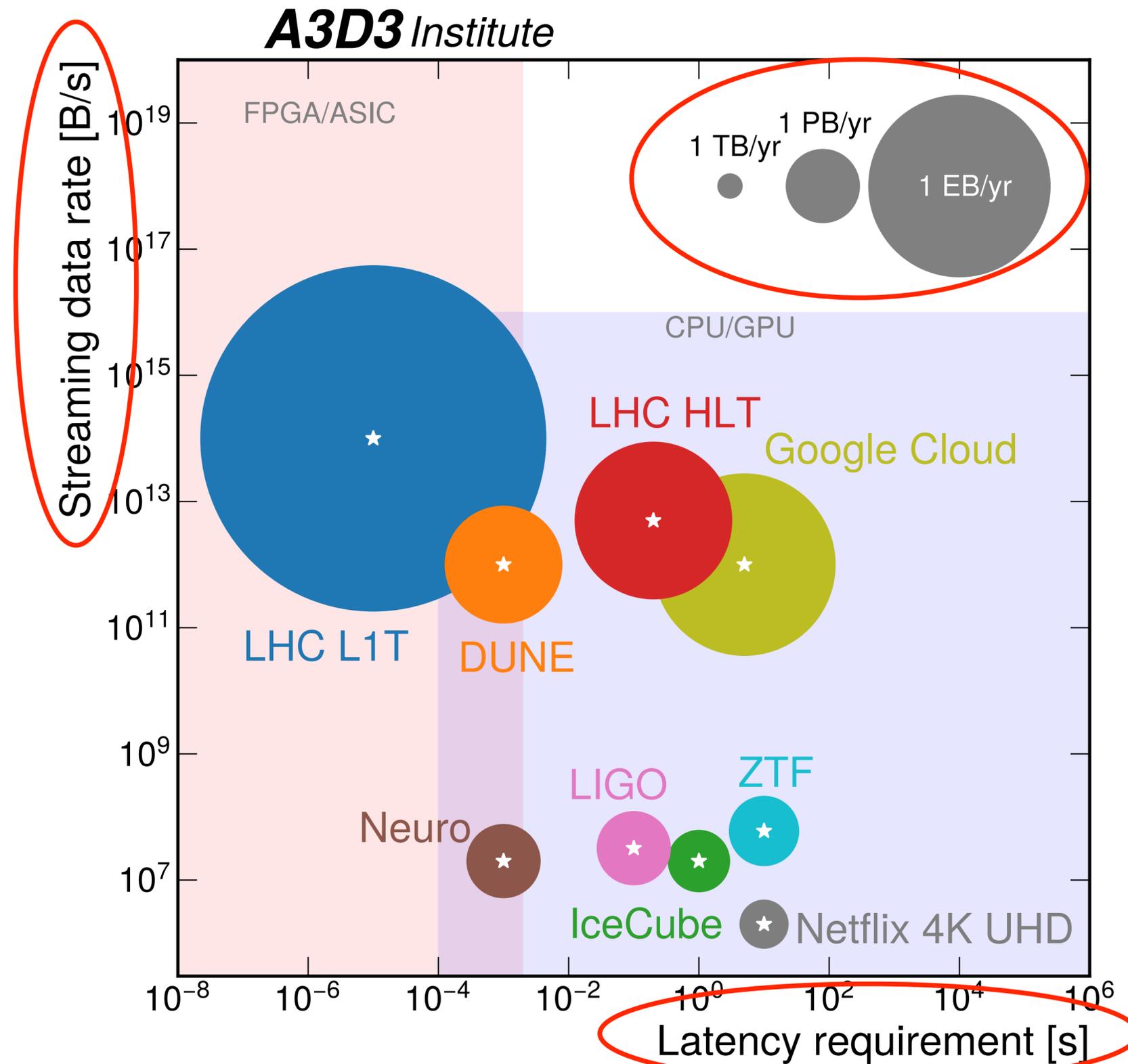
SOUTHERN METHODIST UNIVERSITY

OCTOBER 3-6, 2022

Scan the QR Code or visit the link below for registration information
<https://indico.cern.ch/e/fml2022>

Streaming DAQ and real-time AI: A new paradigm shift for experiments in next NP LRP





▶ **Tools**

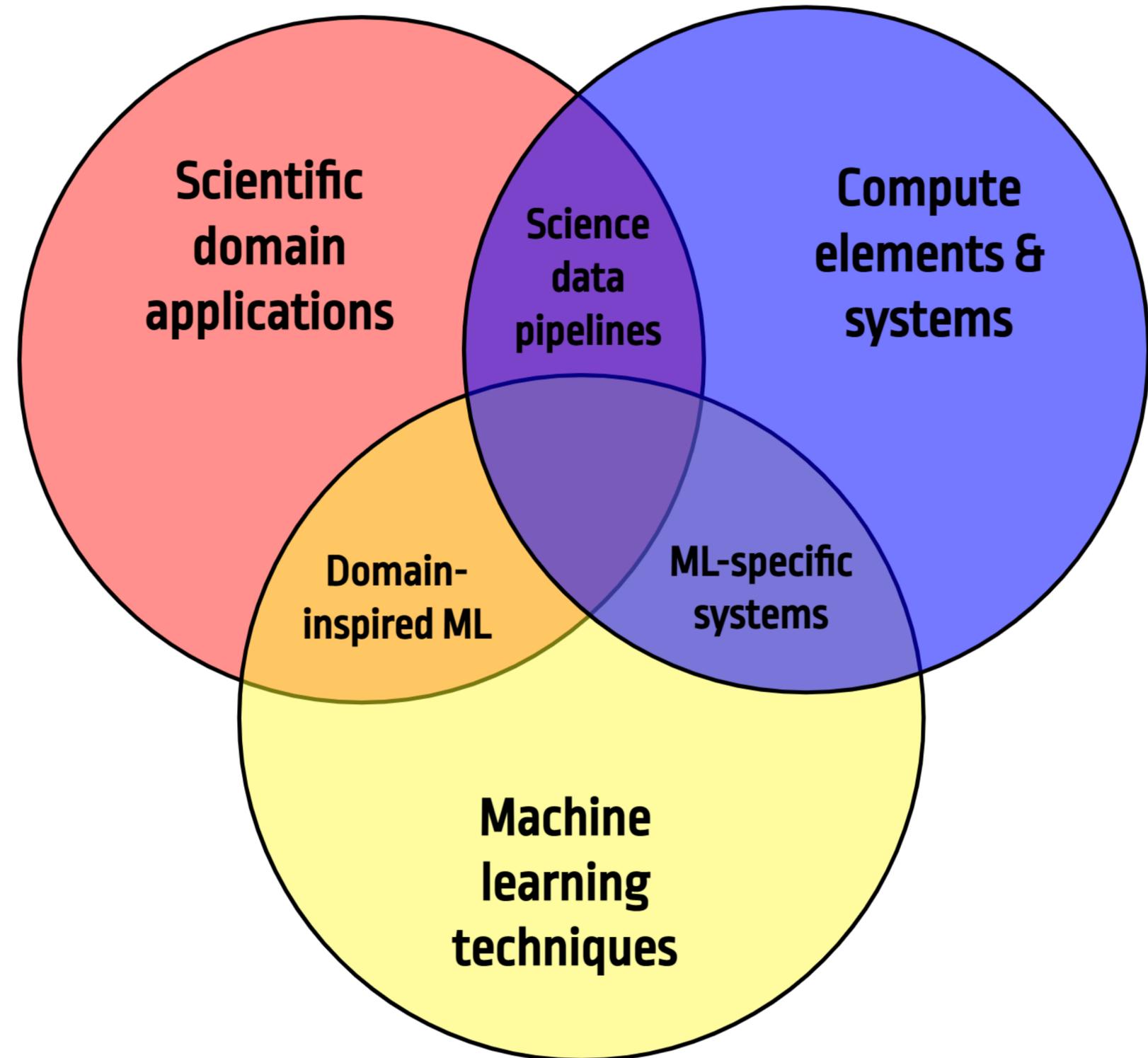
- ▶ Accessible workflows like HLS to make hardware more accessible domain scientists

▶ **ML techniques**

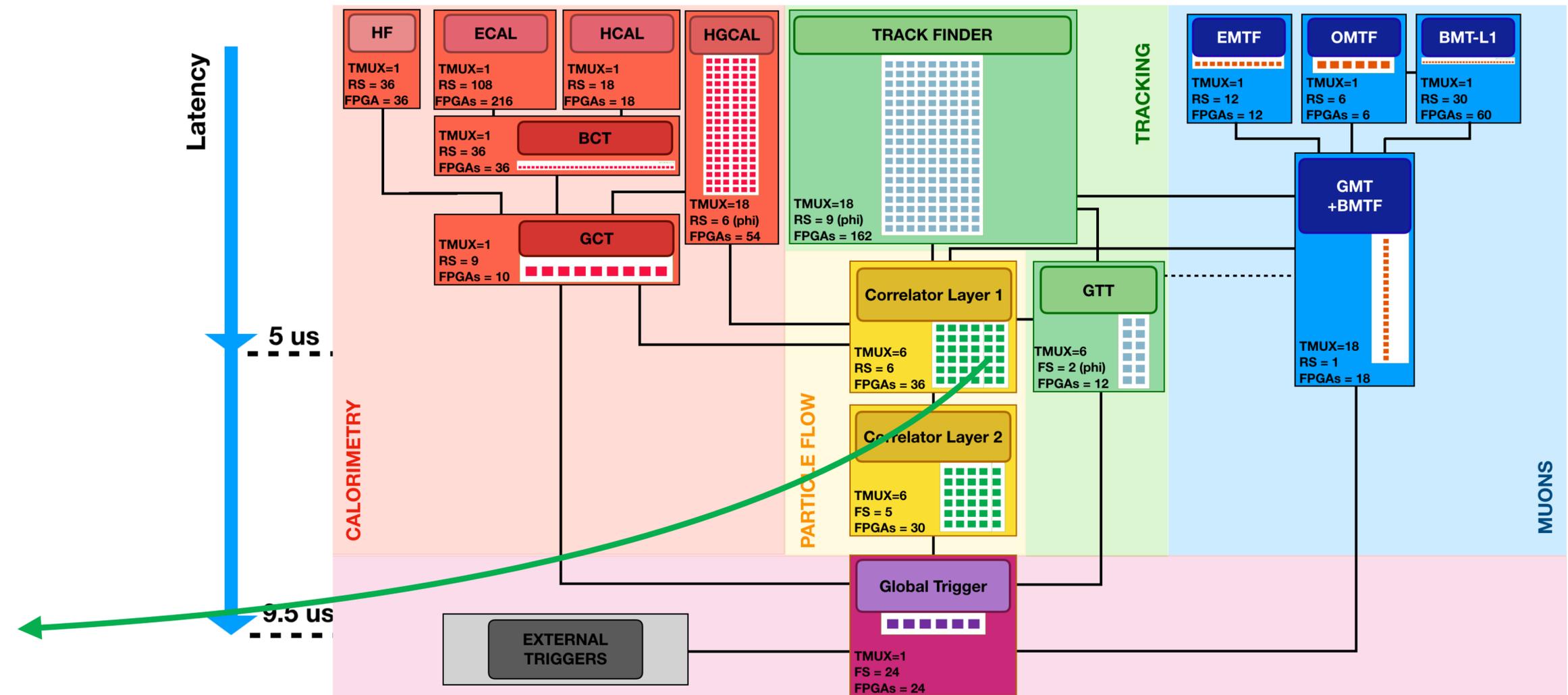
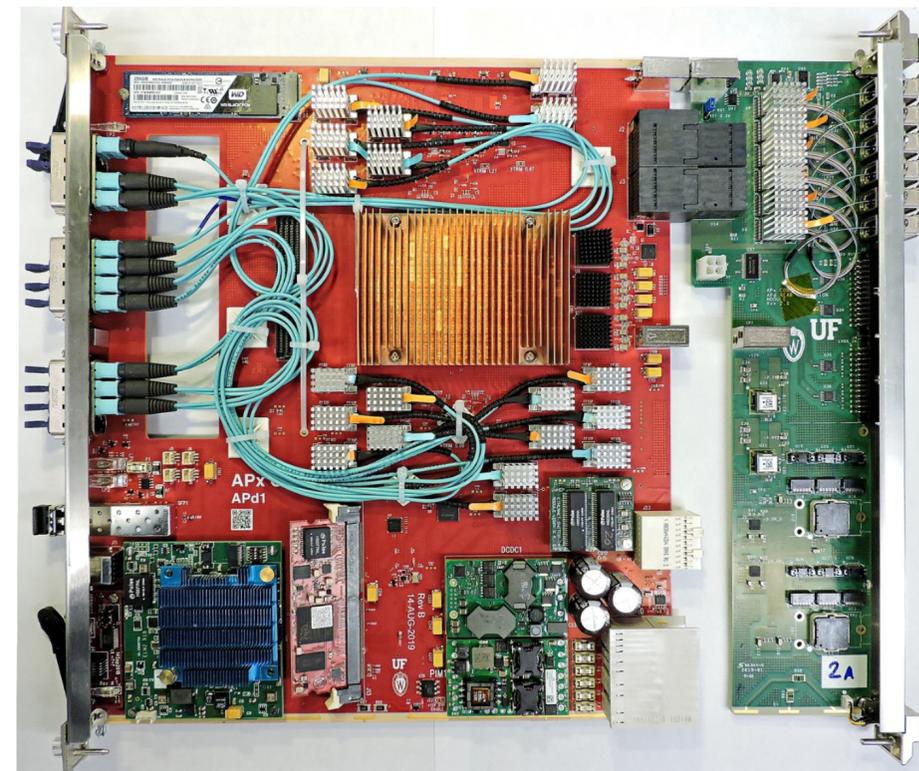
- ▶ Efficient training and implementation methods codesigned for specific hardware

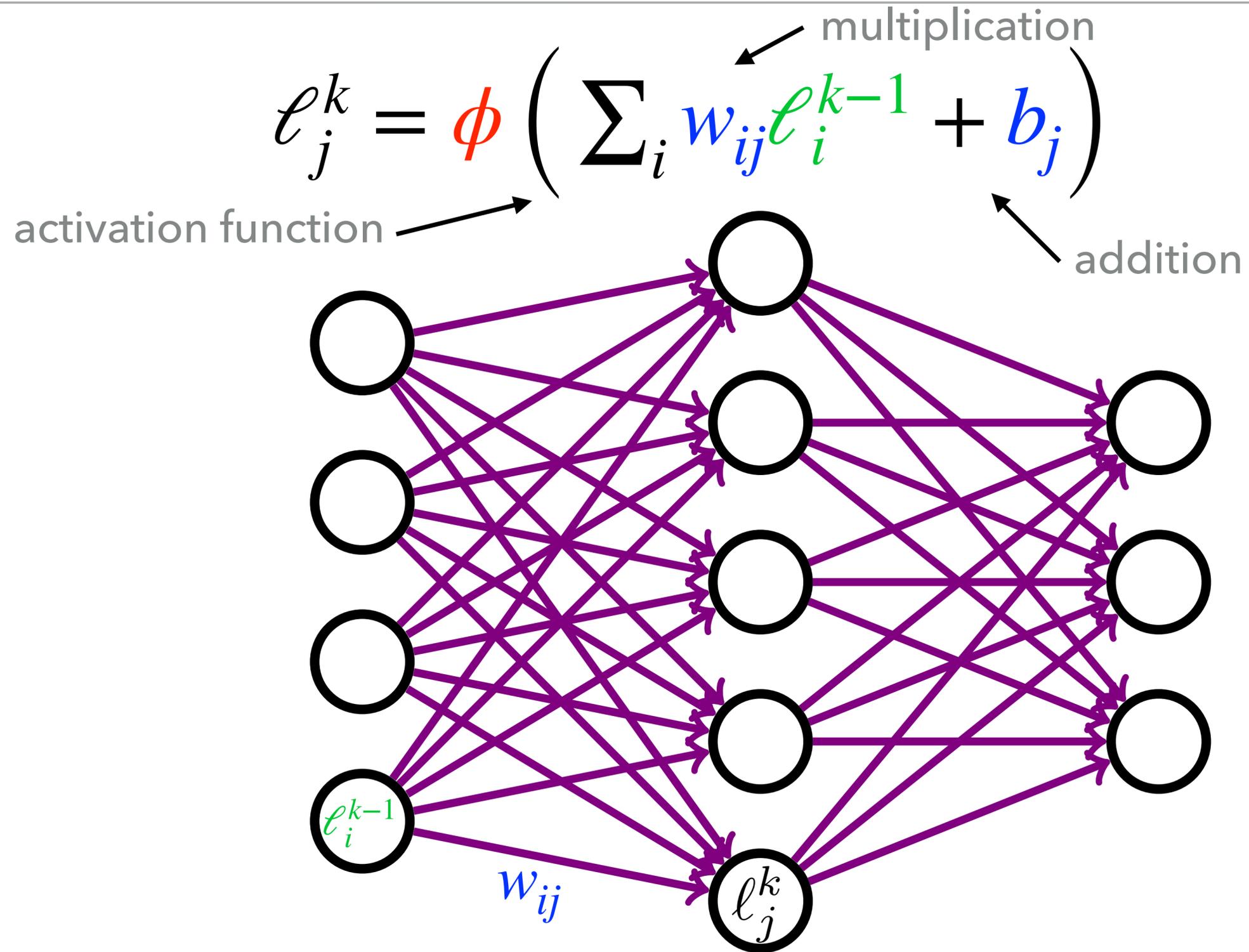
▶ **Hardware**

- ▶ Evolving compute platforms, e.g. power-law growth in FPGA logic

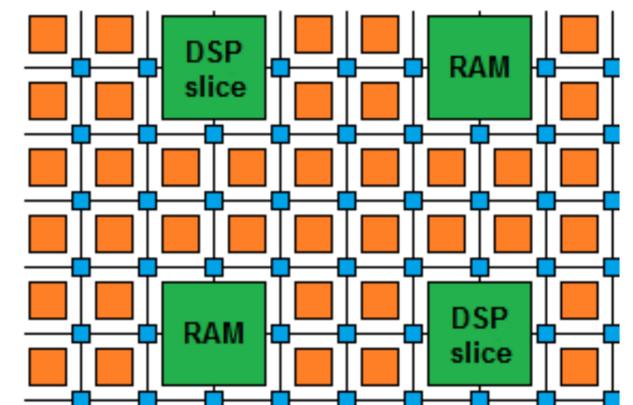


- ▶ Reconstruct all events and reject 98% of them in $\sim 12.5 \mu\text{s}$
 - ▶ Individual algorithms usually have to be $< 1 \mu\text{s}$ and keep up with new events every 25 ns
- ▶ Latency necessitates all **FPGA** design (many algorithms running on 729 FPGAs!)
 - ▶ Individual algorithms usually have to fit on < 1 FPGA





Maps nicely onto FPGA resources: high I/O, DSPs, LUTs, etc.



- ▶ Operations can be implemented with core operations (gates)

LUT

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

LUT

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

LUT

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

LUT

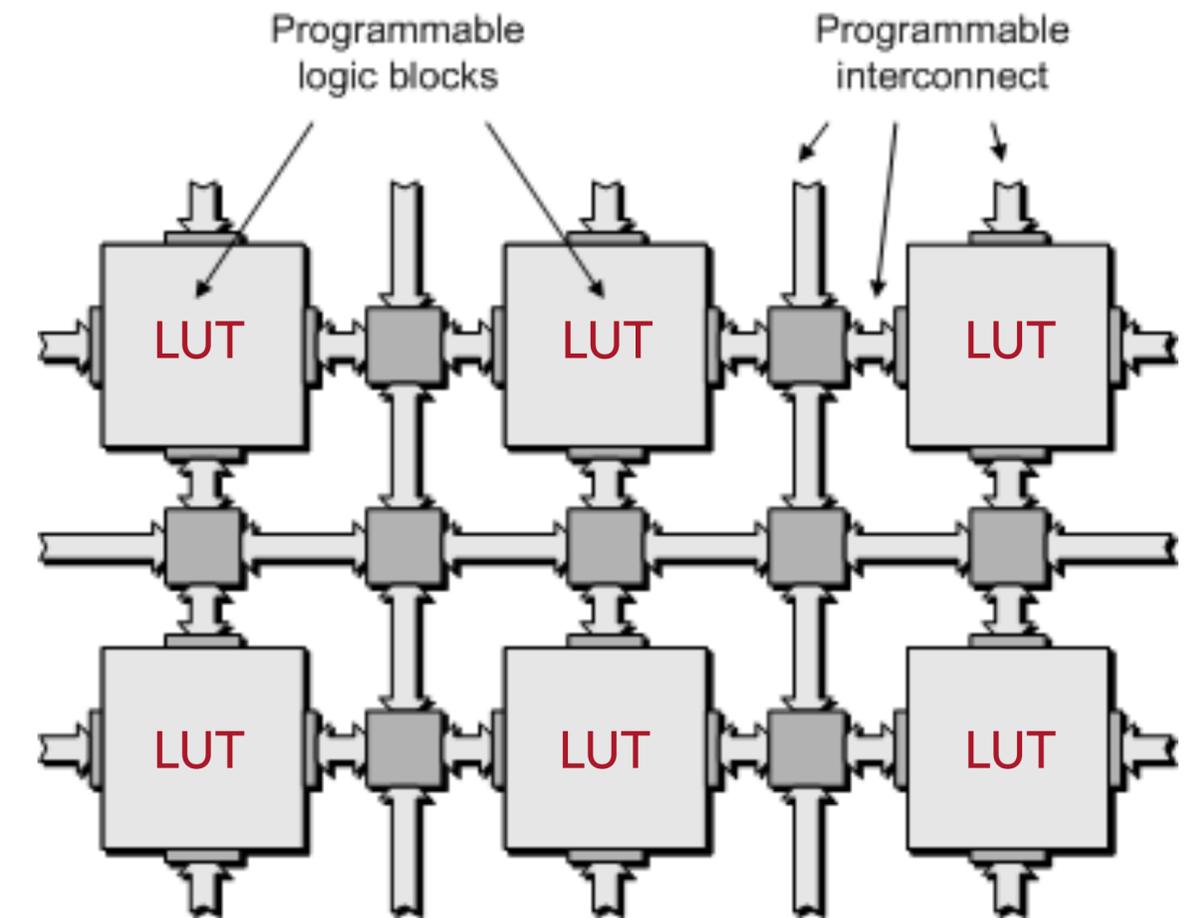
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

LUT

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

LUT

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



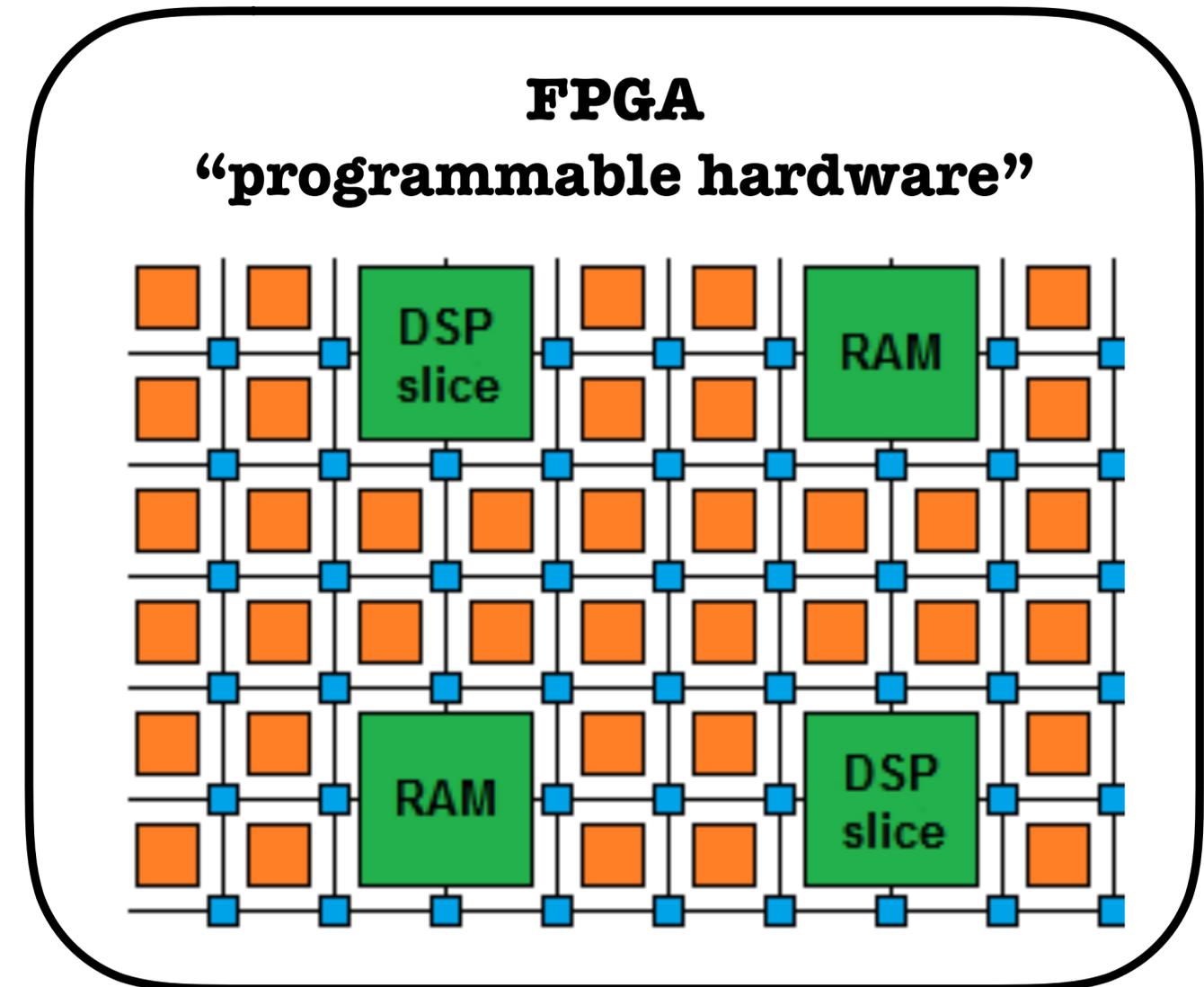
- ▶ Gates are like look-up tables (LUTs)
- ▶ If we can (re-)program arbitrary LUTs and (re-)connect them however we want, we can (re-)implement whatever algorithm we want!

▶ Pros:

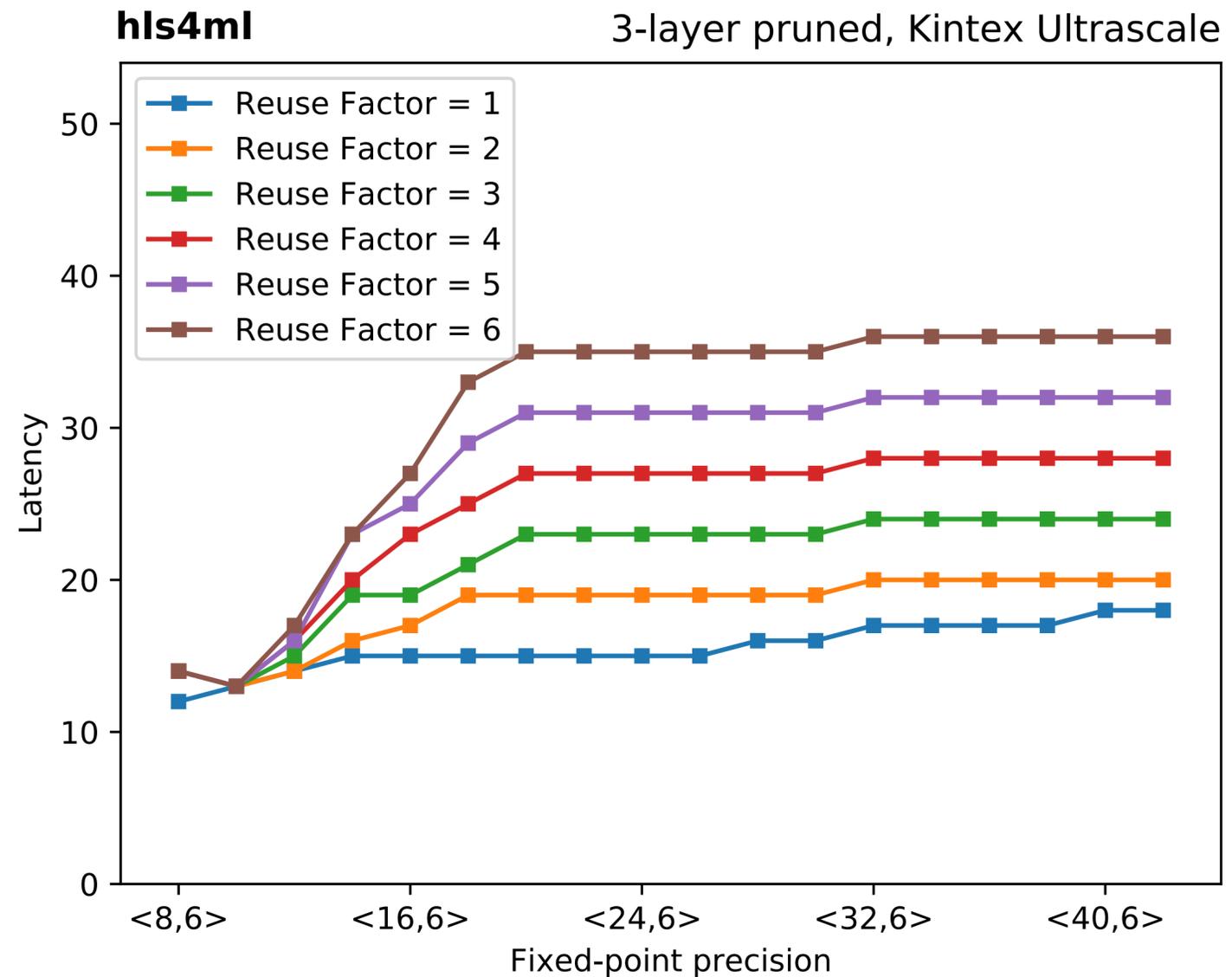
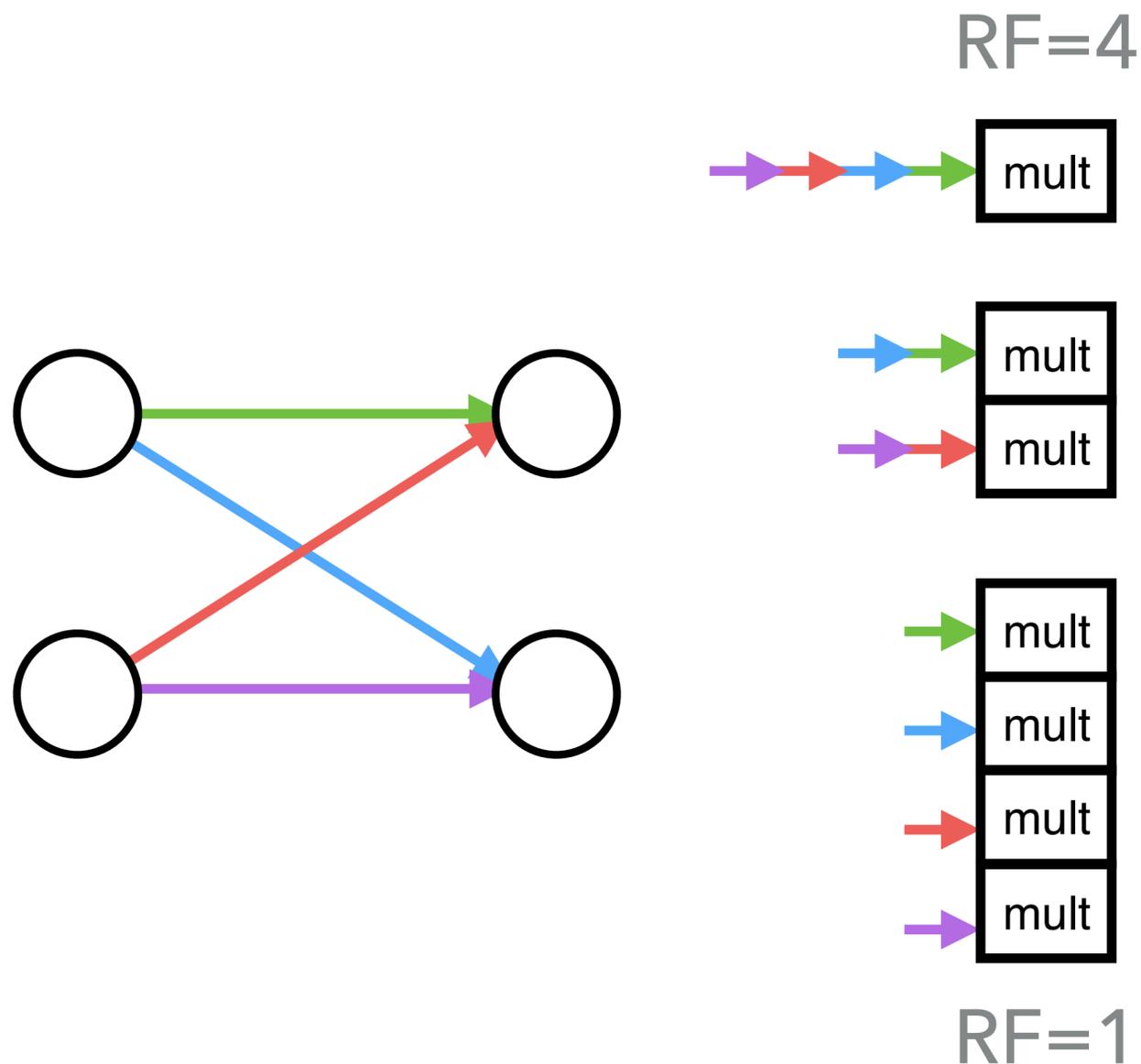
- ▶ Reprogrammable interconnects between embedded components that perform multiplication (DSPs), apply logical functions (LUTs), or store memory (BRAM)
- ▶ High throughput I/O: O(100) optical transceivers running at O(15) Gbps
- ▶ Massively parallel
- ▶ Low power

▶ Cons:

- ▶ Requires domain knowledge to program (using VHDL/Verilog)



- ▶ Decreasing reuse factor, increases parallelization and decreases latency



~35 clocks
@ 200 MHz
= 175 ns

~15 clocks
@ 200 MHz
= 75 ns

- ▶ Algorithm comfortably fits in latency requirements (<1 μ s)

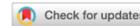
Jinst PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB
RECEIVED: May 10, 2018
ACCEPTED: July 17, 2018
PUBLISHED: July 27, 2018

Fast inference of deep neural networks in FPGAs for particle physics

J. Duarte,^a S. Han,^b P. Harris,^b S. Jindariani,^a E. Kreinar,^c B. Kreis,^a J. Ngadiuba,^d M. Pierini,^d R. Rivera,^a N. Tran^{a,1} and Z. Wu^e

ARTICLES
<https://doi.org/10.1038/s42256-022-00441-3>

nature
machine intelligence



Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider

Ekaterina Govorkova^{1✉}, Ema Puljak¹, Thea Aarrestad¹, Thomas James¹, Vladimir Loncar^{1,2}, Maurizio Pierini¹, Adrian Alan Pol¹, Nicolò Ghielmetti^{1,3}, Maksymilian Graczyk^{1,4}, Sioni Summers¹, Jennifer Ngadiuba^{5,6}, Thong Q. Nguyen⁶, Javier Duarte⁷ and Zhenbin Wu⁸

MACHINE
LEARNING
Science and Technology

Compressing deep neural networks on FPGAs to binary and ternary precision with hls4ml

Jennifer Ngadiuba¹, Vladimir Loncar¹, Maurizio Pierini¹, Sioni Summers¹, Giuseppe Di Guglielmo², Javier Duarte³, Philip Harris⁴, Dylan Rankin⁴, Sergio Jindariani⁵, Mia Liu⁵, Kevin Pedro⁵, Nhan Tran⁵, Edward Kreinar⁶, Sheila Sagar⁷, Zhenbin Wu⁸ and Duc Hoang⁹

A Reconfigurable Neural Network ASIC for Detector Front-End Data Compression at the HL-LHC

Giuseppe Di Guglielmo¹, Farah Fahim¹, Member, IEEE, Christian Herwig¹, Manuel Blanco Valentin, Javier Duarte², Cristian Gingu, Member, IEEE, Philip Harris, James Hirschauer³, Martin Kwok, Vladimir Loncar, Yingyi Luo, Llovizna Miranda, Jennifer Ngadiuba, Daniel Noonan, Seda Orgrenci-Memik, Maurizio Pierini, Sioni Summers, and Nhan Tran⁴

frontiers
in Big Data

ORIGINAL RESEARCH
published: 12 January 2021
doi: 10.3389/fdata.2020.598927



Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics

hls4ml: An Open-Source Codesign Workflow to Empower Scientific Low-Power Machine Learning Devices

Farah Fahim^{*}
Benjamin Hawks
Christian Herwig
James Hirschauer
Sergo Jindariani
Nhan Tran^{*}
Fermilab
Batavia, IL, USA

Luca P. Carloni
Giuseppe Di Guglielmo
Columbia University
New York, NY, USA

Philip Harris
Jeffrey Krupa
Dylan Rankin
MIT
Cambridge, MA, USA

Manuel Blanco Valentin
Josiah Hester
Yingyi Luo
John Mamish
Seda Orgrenci-Memik
Northwestern University
Evanston, IL, USA

Thea Aarrestad
Hamza Javed
Vladimir Loncar
Maurizio Pierini
Adrian Alan Pol
Sioni Summers
European Organization for Nuclear
Research (CERN)
Geneva, Switzerland

Javier Duarte
UC San Diego
La Jolla, CA, USA
jduarte@ucsd.edu

Scott Hauck
Shih-Chieh Hsu
University of Washington
Seattle, WA, USA

Jennifer Ngadiuba
Caltech
Pasadena, CA, USA

Mia Liu
Purdue University
West Lafayette, IN, USA

Duc Hoang
Rhodes College
Memphis, TN, USA

Edward Kreinar
HawkEye360
Herndon, VA, USA

Zhenbin Wu
University of Illinois at Chicago
Chicago, IL, USA

arXiv:2103.05579v3 [cs.LG] 23 Mar 2021

ESP4ML: Platform-Based Design of Systems-on-Chip for Embedded Machine Learning

Davide Giri, Kuan-Lin Chiu, Giuseppe Di Guglielmo, Paolo Mantovani and Luca P. Carloni
Department of Computer Science · Columbia University, New York
[davide_giri, chiu, giuseppe, paolo, luca]@cs.columbia.edu

Jinst

PUBLISHED BY IOP PUBLISHING FOR SISSA MEDIALAB

RECEIVED: February 20, 2020

ACCEPTED: April 7, 2020

PUBLISHED: May 29, 2020

Fast inference of Boosted Decision Trees in FPGAs for particle physics

S. Summers,^{a,1} G. Di Guglielmo,^b J. Duarte,^c P. Harris,^d D. Hoang,^e S. Jindariani,^f E. Kreinar,^g V. Loncar,^{a,h} J. Ngadiuba,^a M. Pierini,^a D. Rankin,^d N. Tran^f and Z. Wuⁱ

nature
machine intelligence

ARTICLES

<https://doi.org/10.1038/s42256-021-00356-5>



Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors

Claudionor N. Coelho Jr¹, Aki Kuusela², Shan Li², Hao Zhuang², Jennifer Ngadiuba³, Thea Klaeboe Aarrestad^{4,5}, Vladimir Loncar^{4,5}, Maurizio Pierini⁴, Adrian Alan Pol⁴ and Sioni Summers⁴

CERN European Organization for Nuclear Research CERN-LHCC-2020-004
Organisation européenne pour la recherche nucléaire CMS-TDR-021
10 March 2020

The Phase-2 Upgrade of the CMS Level-1 Trigger Technical Design Report

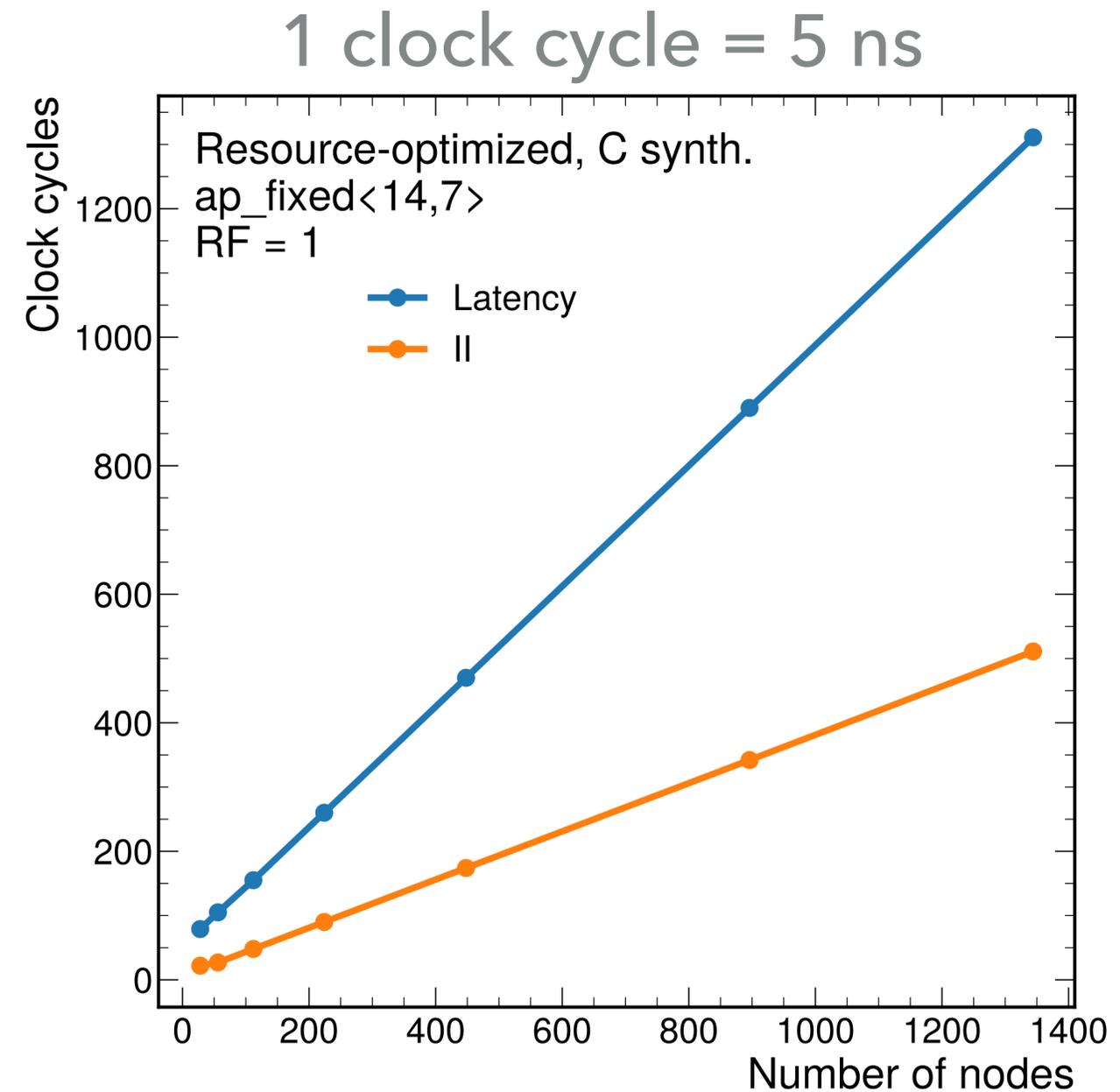
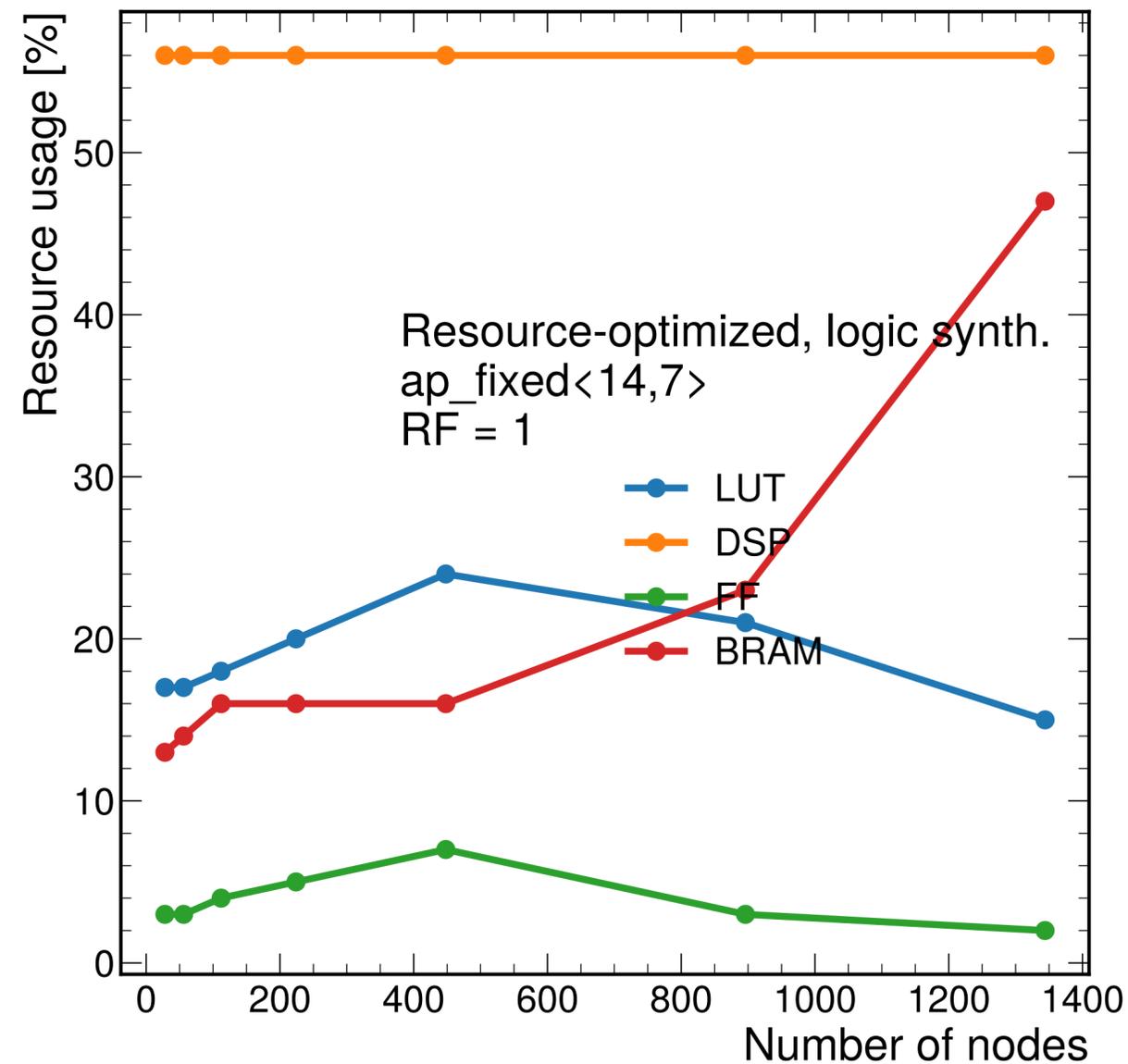
frontiers
in Big Data

ORIGINAL RESEARCH
published: 23 March 2022
doi: 10.3389/fdata.2022.828666



Graph Neural Networks for Charged Particle Tracking on FPGAs

Abdelrahman Elabd¹, Vesal Razavimaleki², Shi-Yu Huang³, Javier Duarte^{2*}, Markus Atkinson⁴, Gage DeZoort⁵, Peter Elmer⁵, Scott Hauck⁶, Jin-Xuan Hu³, Shih-Chieh Hsu^{6,7}, Bo-Cheng Lai³, Mark Neubauer^{6*}, Isobel Ojalvo⁵, Savannah Thais⁵ and Matthew Trahms⁶



- ▶ Modified design can scale to much larger graphs (~1400 nodes, ~2800 edges), for longer latency (6 μ s) and II (2 μ s)

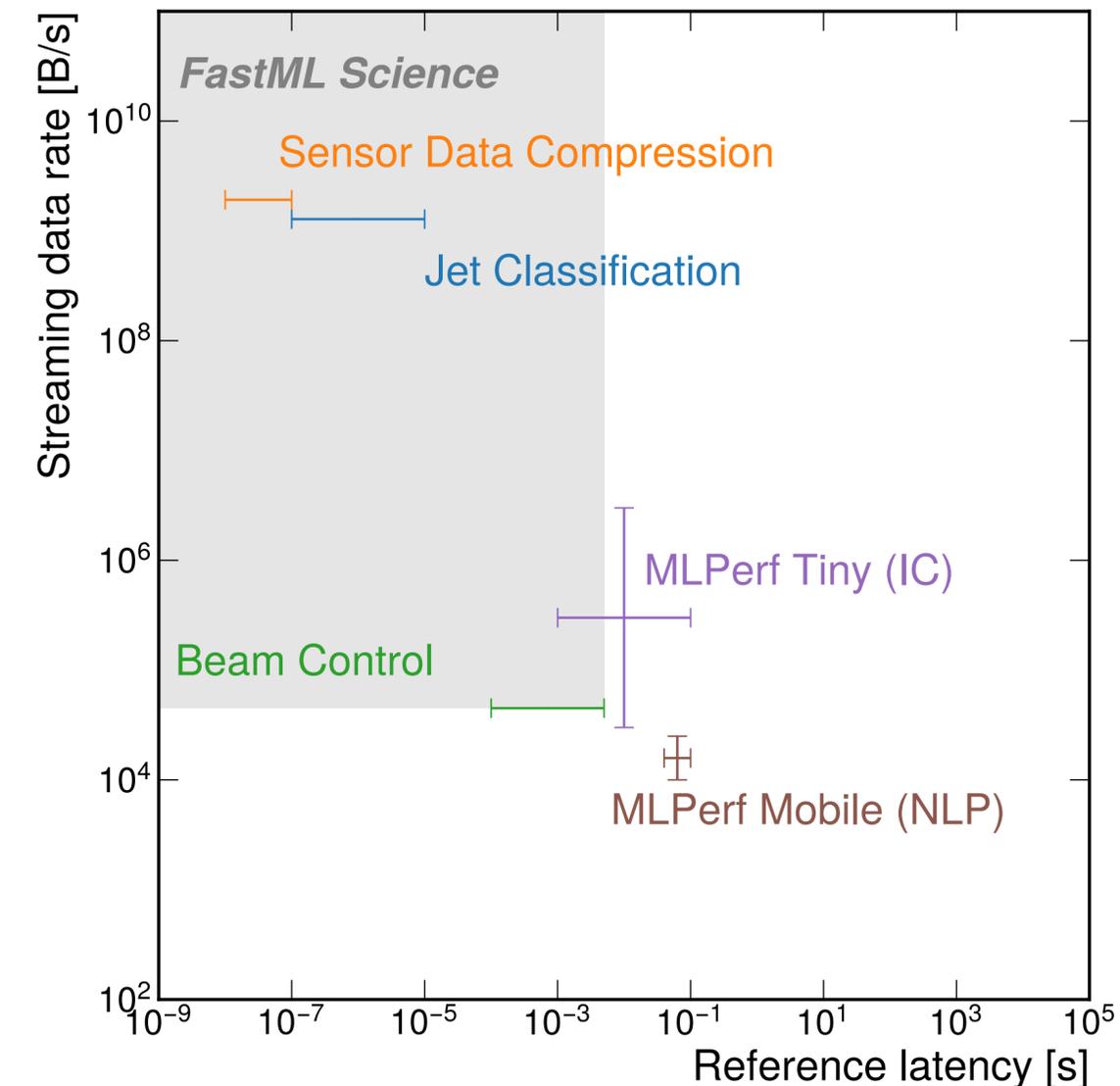
1. Define generic ML benchmarks for bespoke domain problems that attract interest from a broad community of system and ML experts
2. Design benchmarks to satisfy challenging scientific requirements that overlap with a number of systems

- ▶ Set of 3 benchmarks inspired by low-latency edge ML use cases in science
- ▶ Cover a wide range of latency/data rate constraints
- ▶ Unique set of qualities

	Formalized Benchmark	Scientific Workload(s)	Edge Computing	Real-Time Constraints
FastML Science Benchmarks (this work)	✓	✓	✓	✓
SciMLBench (Thiyagalingam et al., 2021)	✓	✓	✓	x
LHC New Physics Dataset (Govorkova et al., 2021)	x	✓	✓	✓
MLPerf HPC (Farrell et al., 2021)	✓	✓	x	x
BenchCouncil AIBench HPC (BenchCouncil, 2018)	✓	✓	x	x
MLCommons Science (MLCommons, 2020)	✓	✓	x	x
ITU Modulation Classification (ITU, 2021)	x	x	✓	✓

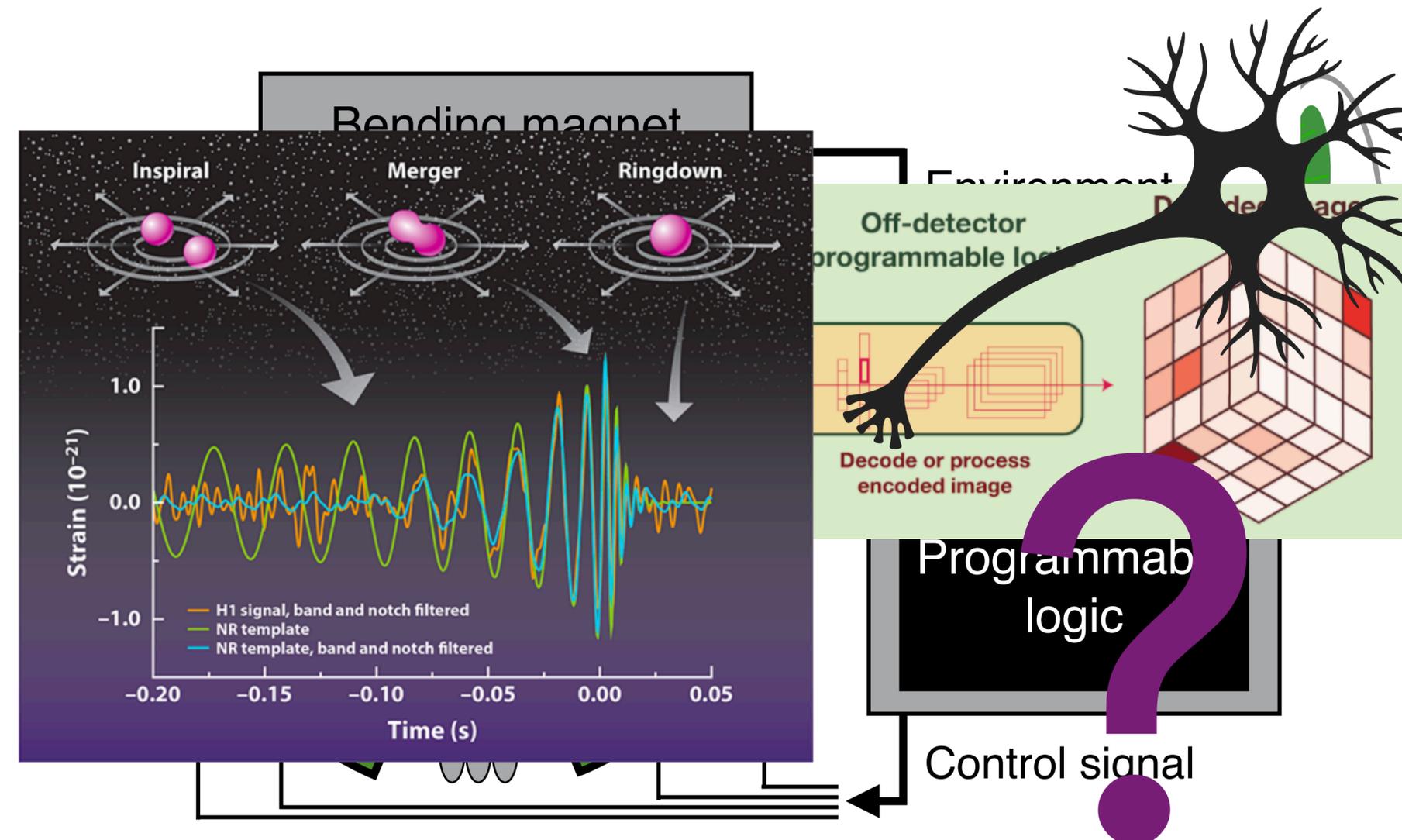
FASTML SCIENCE BENCHMARKS: ACCELERATING REAL-TIME SCIENTIFIC EDGE MACHINE LEARNING

Javier Duarte^{*1} Nhan Tran^{*2} Ben Hawks² Christian Herwig²
Jules Muhizi³ Shvetank Prakash³ Vijay Janapa Reddi³



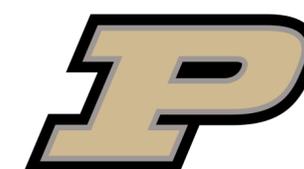
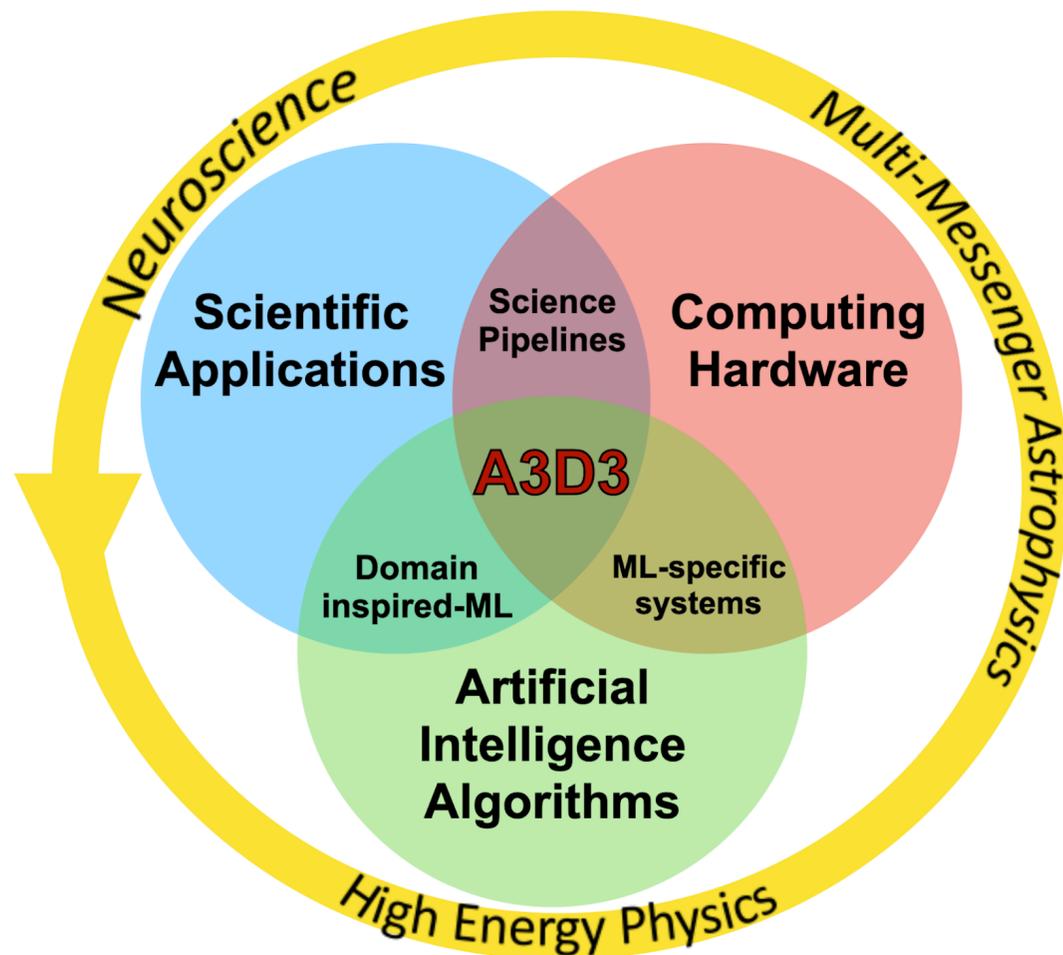
Type	Benchmark	Input Precision	Pipeline Rate	Real-time Latency	Misc. Req.	Baseline Model Parameters
Supervised Learning	Jet Classification	16b	150 ns	1 μ s	-	4,389
Unsupervised Learning	Sensor Data Compression	9b	25 ns	100 ns	area, power (65 nm)	2,288
Reinforcement Learning	Beam Control	32b	5 ms	5 ms	-	34,695

- ▶ Particle jet classification for level-1 trigger: $\sim 1 \mu$ s latency
- ▶ Sensor data compression: ~ 100 ns latency and additional area/power requirements
- ▶ Reinforcement learning for steering accelerator beams: ~ 5 ms latency
- ▶ *Future: Time sequence analysis for gravitational wave or neural data, and more?*

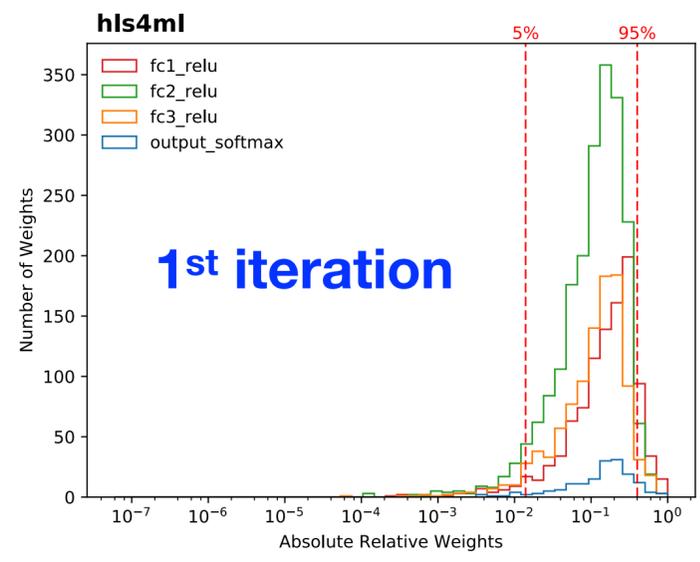


- ▶ Tightly coupled organization of domain scientists, computer scientists, and engineers that unite three core components which are essential to achieve real-time AI to transform science: AI techniques, Computing Hardware, Scientific Applications
- ▶ Collaborators welcome! Check the a3d3.ai for events

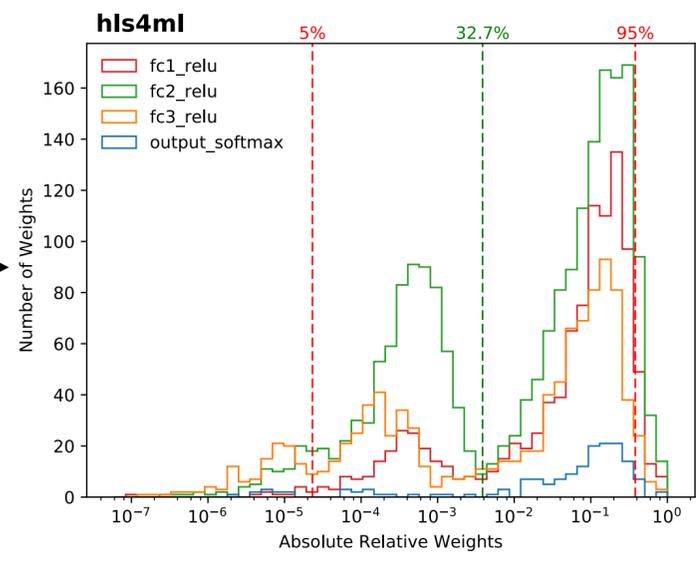
[OAC-2117997](https://www.nsf.gov/awardsearch/showAward?AWDNO=OAC-2117997)



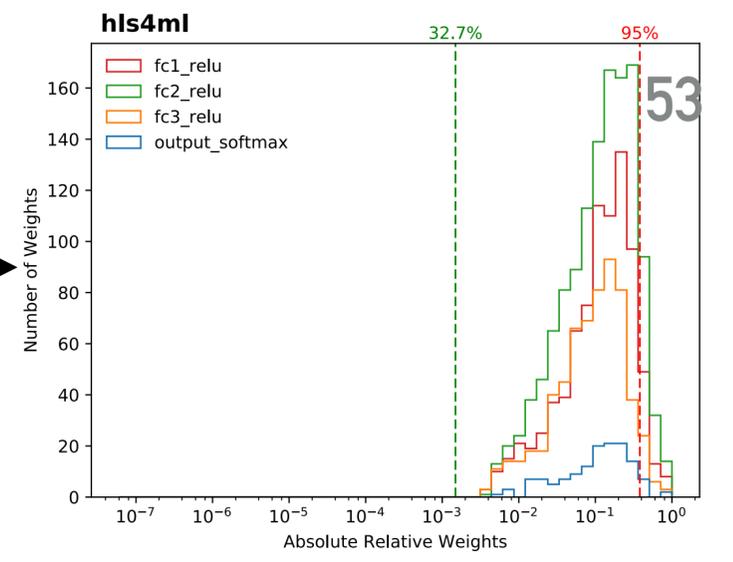
PRUNING



Train
with L₁

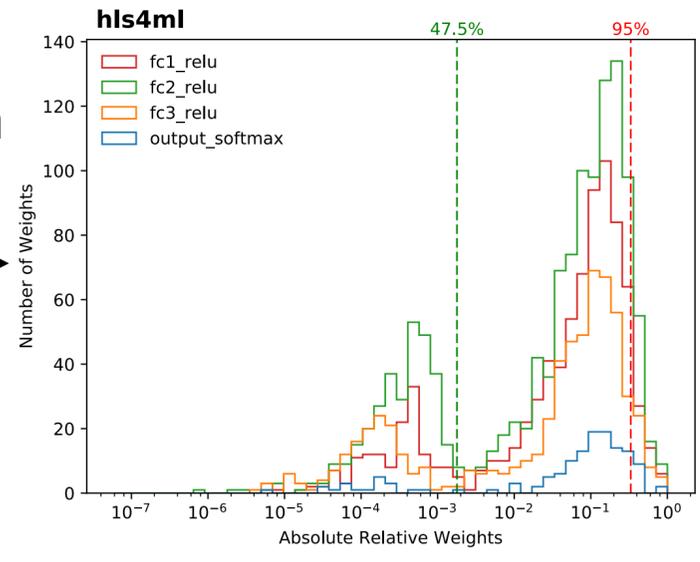


Prune

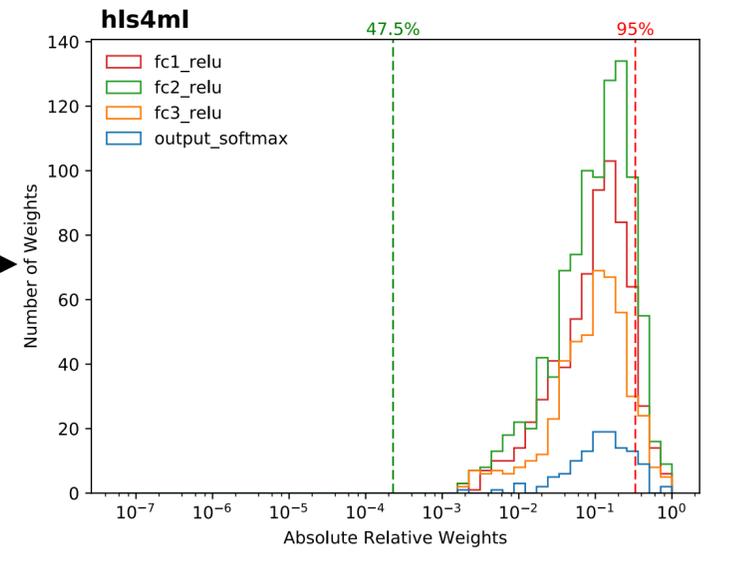


2nd iteration

Retrain
with L₁

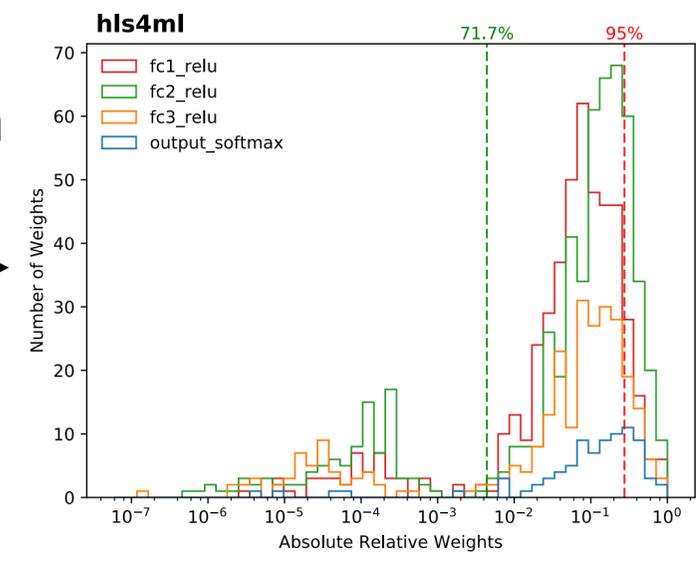


Prune



7th iteration

Retrain
with L₁



Prune

