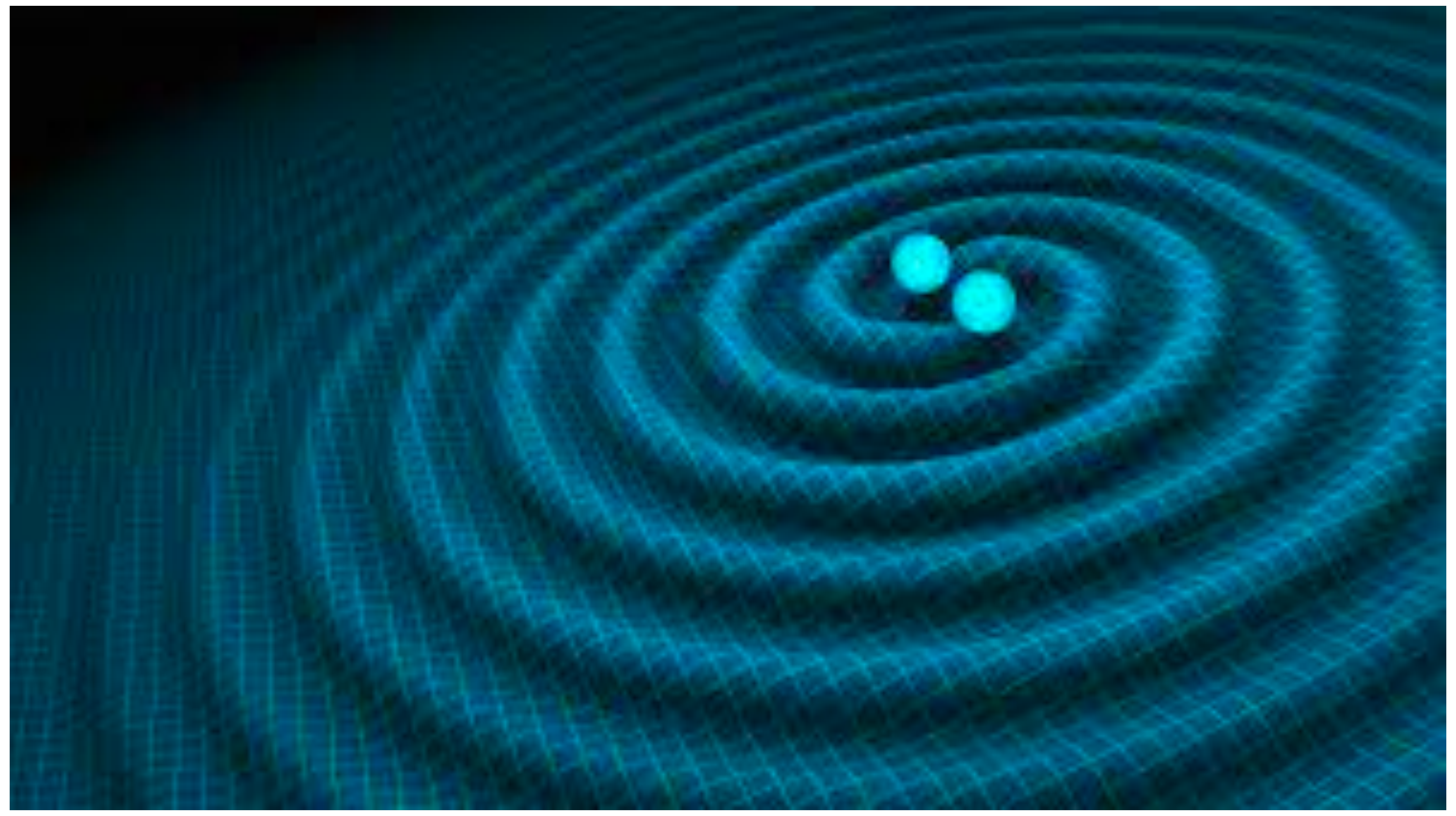
Machine learning for low-latency inference

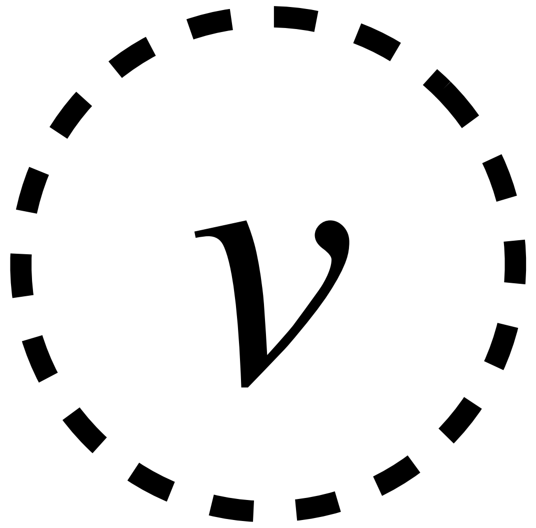
Dylan Rankin [UPenn]
Lepton Photon 2023
July 20th, 2023

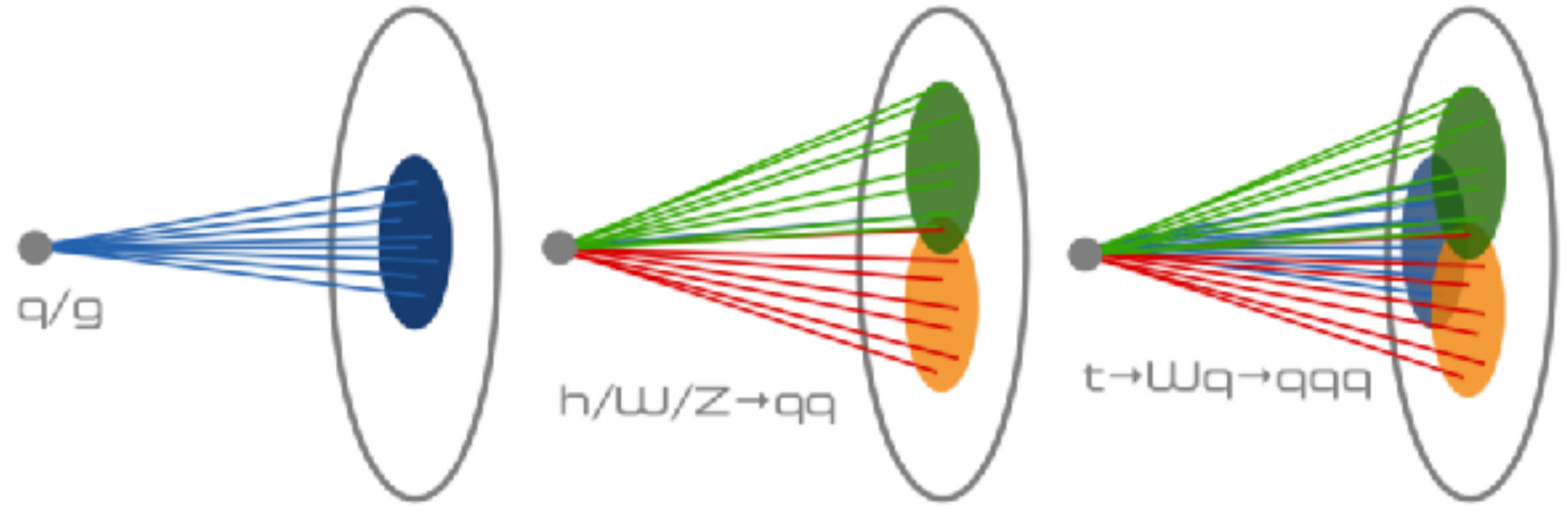
Introduction

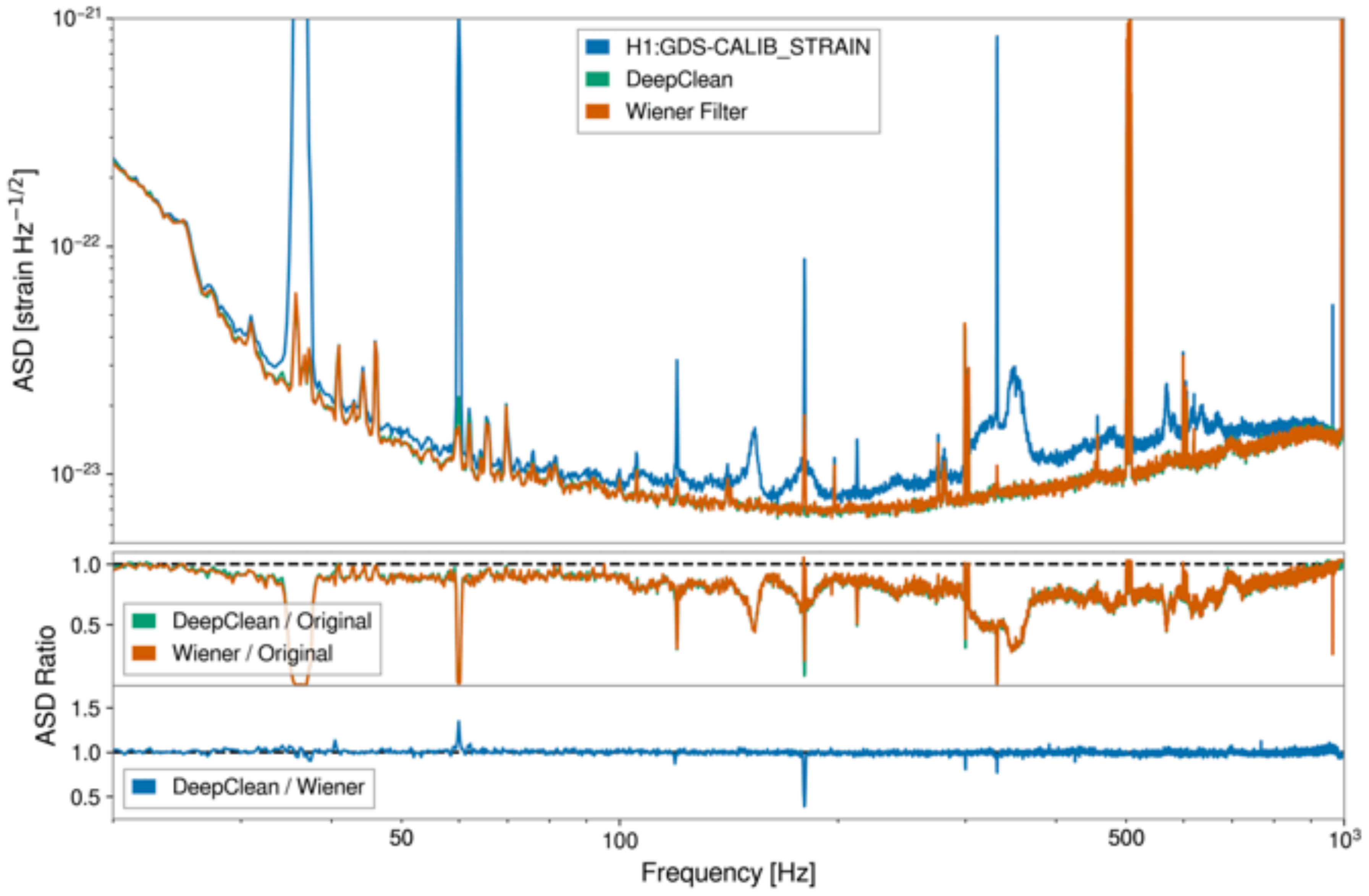
- Machine learning (ML) is becoming more and more popular
  - Availability of CPUs, GPUs, software has accelerated adoption
- What about low-latency (FPGA/ASIC-based) systems?
  - (How) can ML inference be run effectively in O(100 ns)?
- Applications
- Future

Machine Learning (ML) is Everywhere

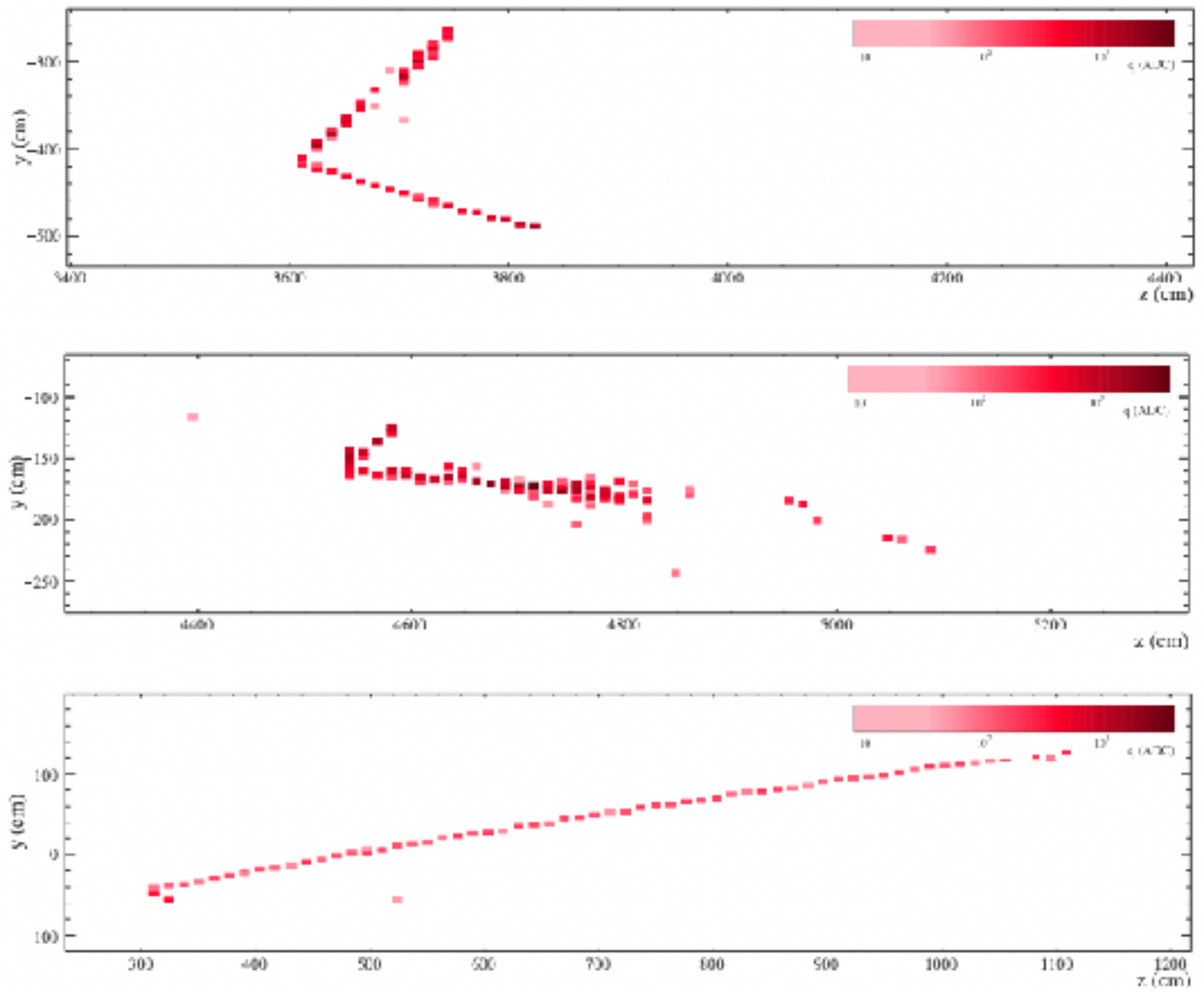




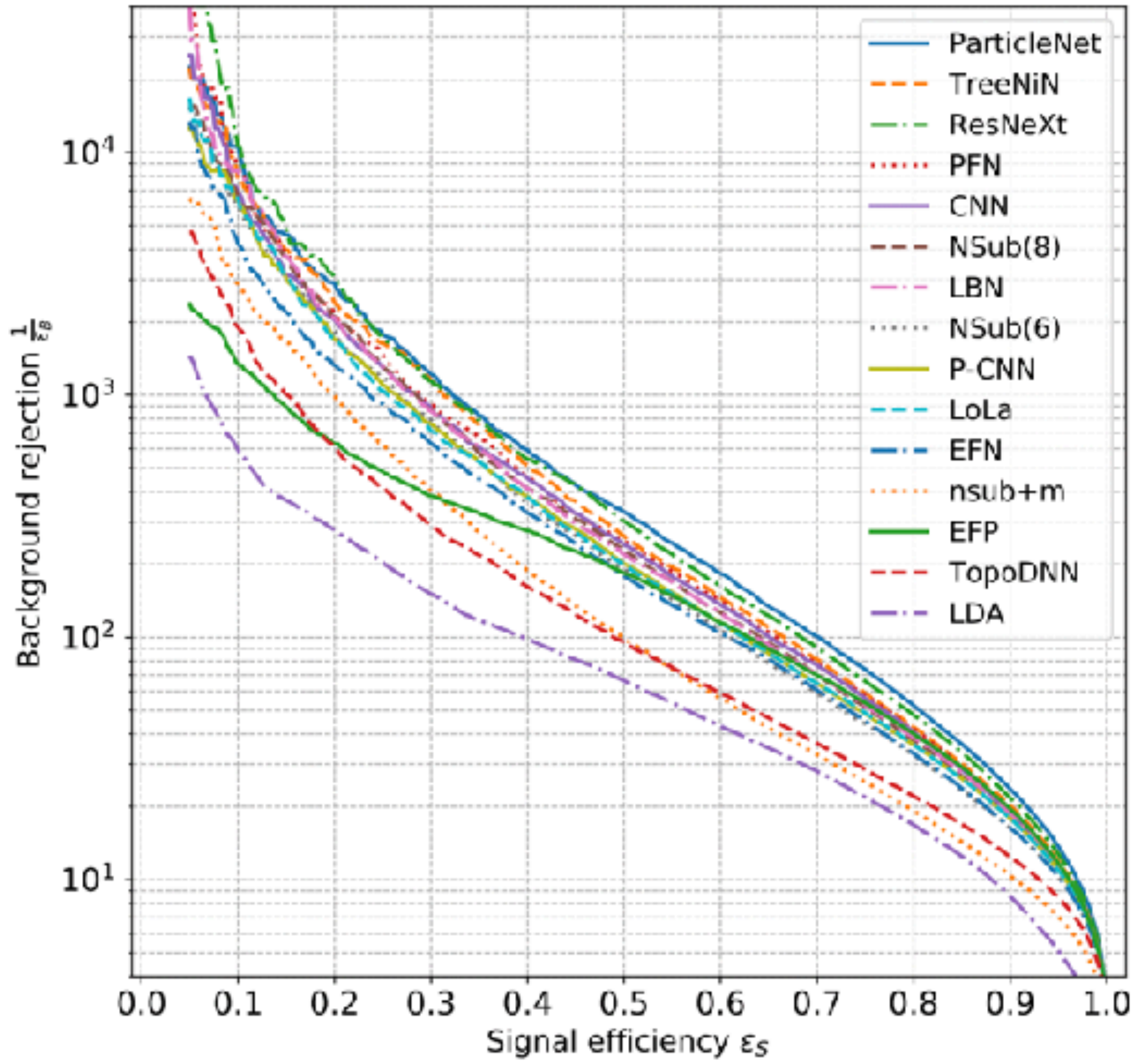




arXiv:2005.06534



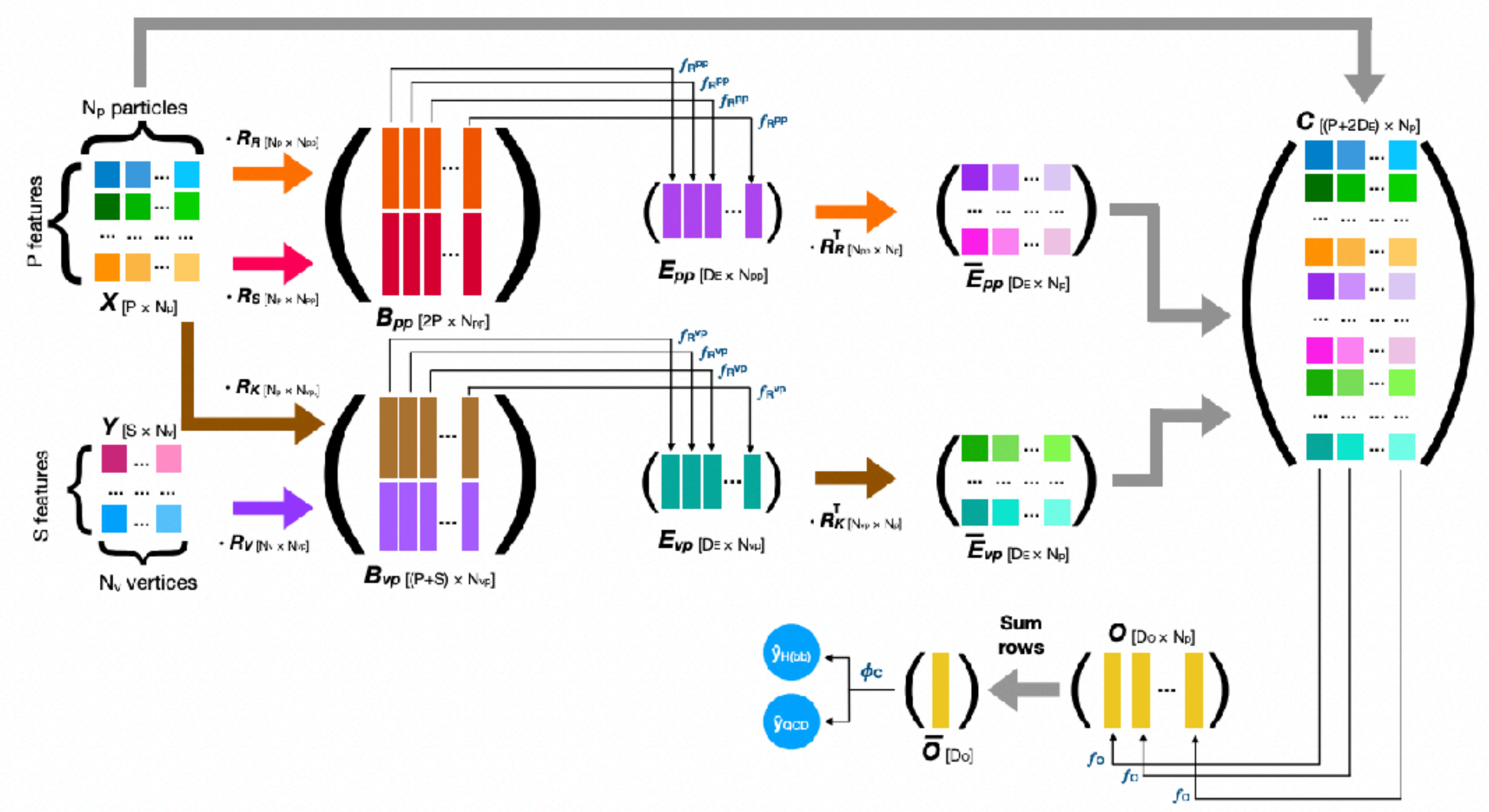
arXiv:1604.01444



arXiv:1902.09914

Machine Learning is (almost) Everywhere

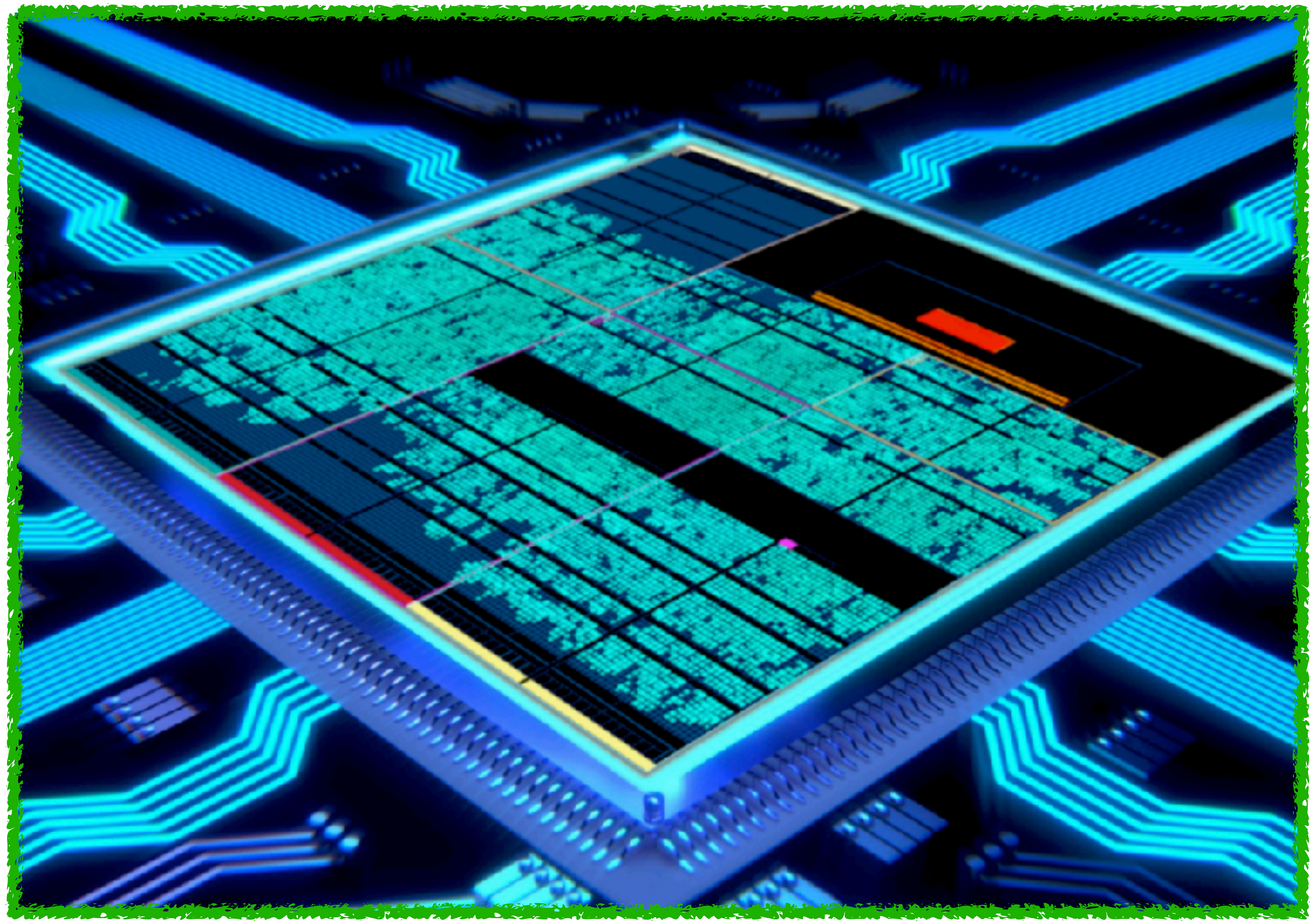
- Trends in ML towards bigger and more complicated models, more computing (GPUs)
- → **Majority of ML in physics is "off detector"**
  - System latency limits are typically soft (if at all)
  - No radiation
  - Issues do not impact data collection
    - Can re-run algorithms/workflows



arXiv:1909.12285

What if…

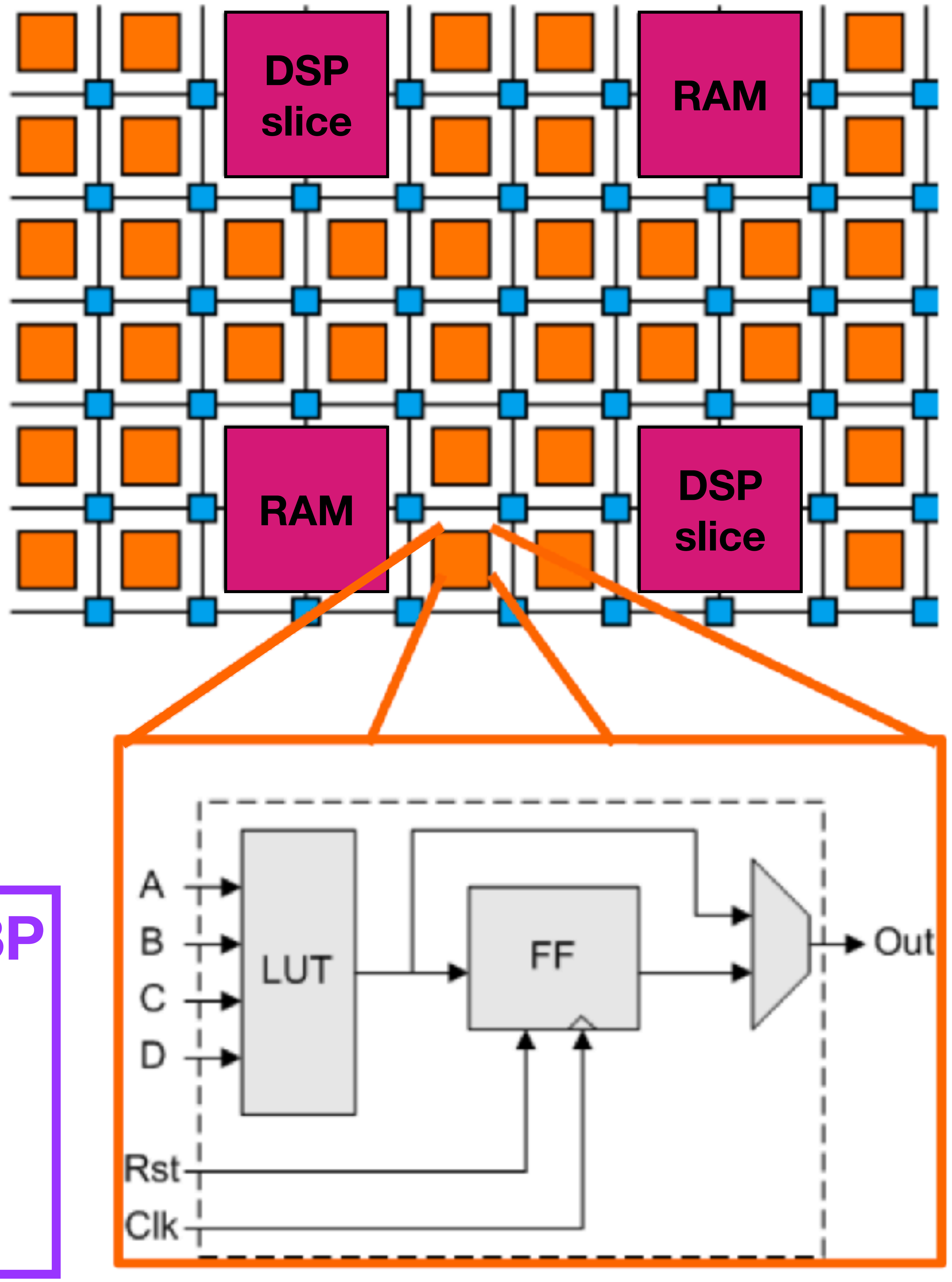
- What if:
  - **System latency limits are low?** ($\lesssim \mu$s)
  - High radiation?
- Requires dedicated hardware
  - **FPGAs** (or ASICs)



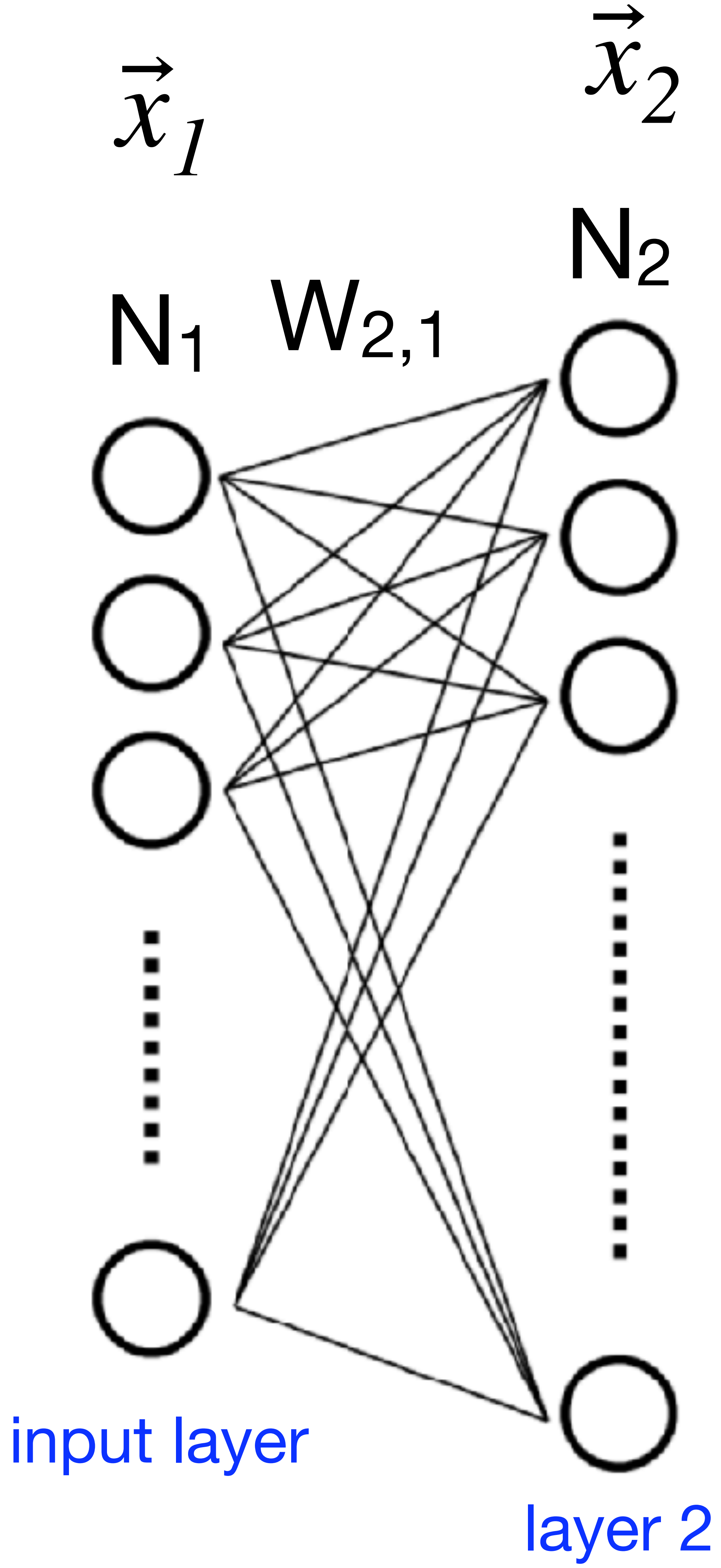
What is an FPGA?

- **F**ield-**p**rogrammable **g**ate **a**rray
- Building blocks:
  - **Multiplier units (DSPs) [arithmetic]**
  - **Look Up Tables (LUTs) [logic]**
  - **Flip-flops (FFs) [registers]**
  - **Block RAMs (BRAMs) [memory]**
- Algorithms are wired onto the chip
  - Can only use the resources on the chip
- Run at high frequency: hundreds of MHz, O(ns) runtime

Xilinx Virtex Ultrascale+ VU13P
12288 Multipliers
1.7M LUTs
3.4M FFs
95 Mb BRAM

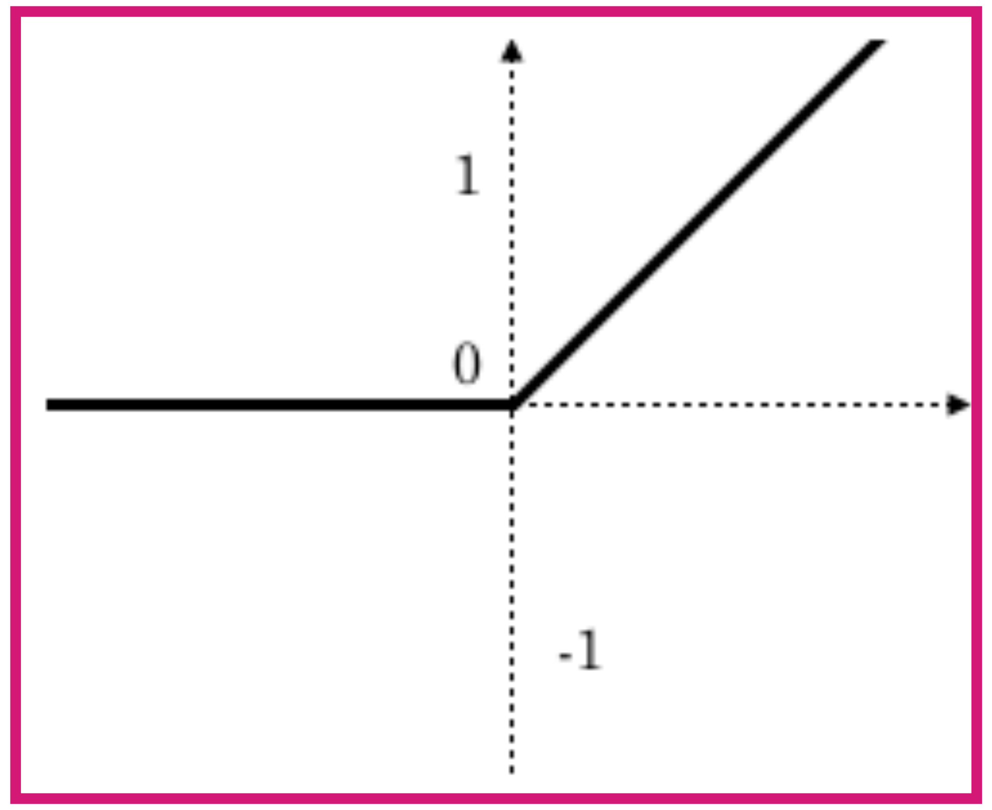


What is a Neural Network?



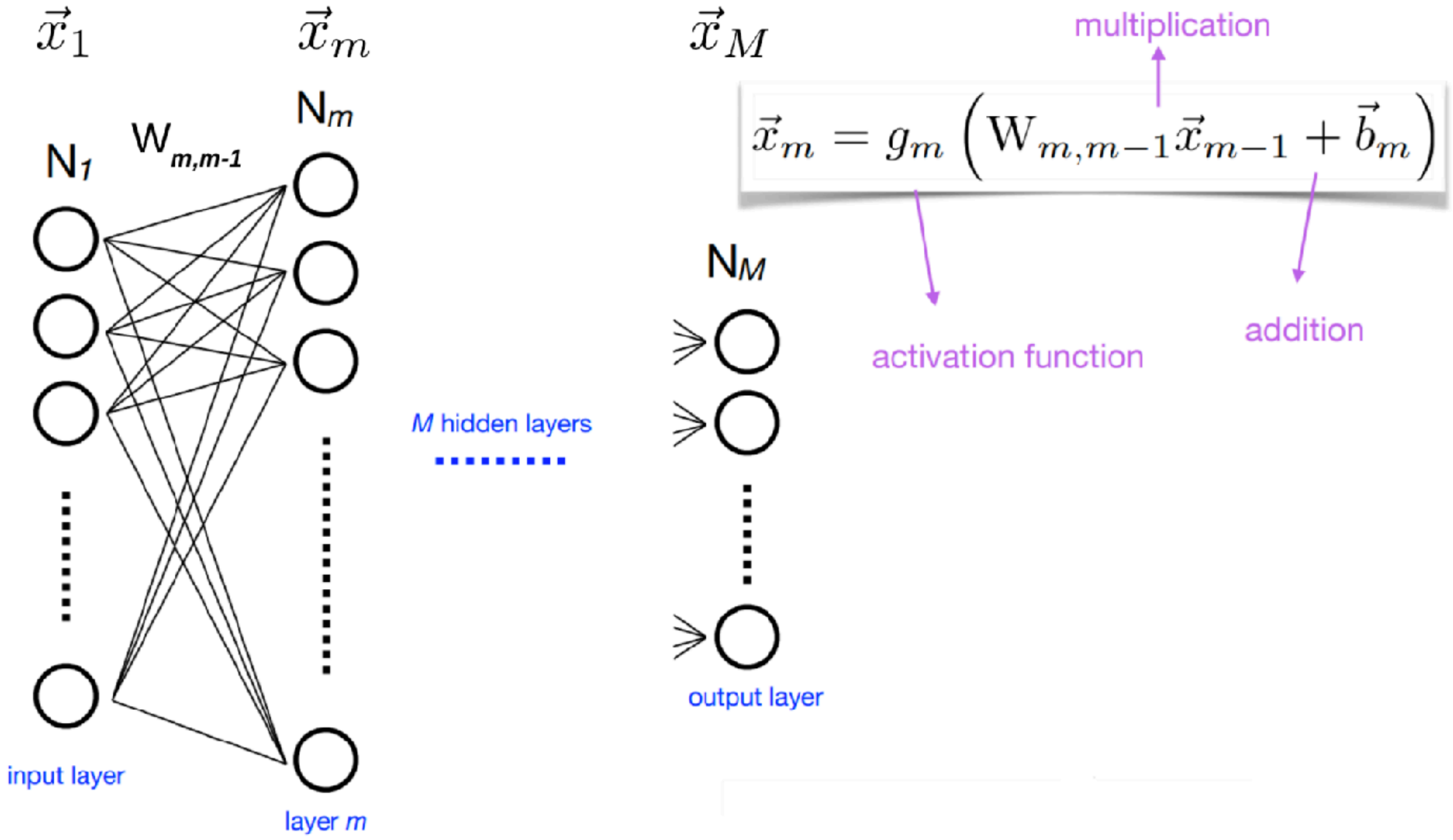
$$\vec{x}_1 = \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ \vdots \\ x_{1,N_1} \end{pmatrix} \qquad W_{2,1} = \begin{pmatrix} w^{(2,1)}_{1,1} & w^{(2,1)}_{2,1} & w^{(2,1)}_{3,1} & & w^{(2,1)}_{N_1,1} \\ w^{(2,1)}_{1,2} & w^{(2,1)}_{2,2} & w^{(2,1)}_{3,2} & \cdots & w^{(2,1)}_{N_1,2} \\ w^{(2,1)}_{1,3} & w^{(2,1)}_{2,3} & w^{(2,1)}_{3,3} & & w^{(2,1)}_{N_1,3} \\ & \vdots & & \ddots & \\ w^{(2,1)}_{1,N_2} & w^{(2,1)}_{2,N_2} & w^{(2,1)}_{3,N_2} & & w^{(2,1)}_{N_1,N_2} \end{pmatrix} \qquad \vec{b}_2 = \begin{pmatrix} b_{2,1} \\ b_{2,2} \\ b_{2,3} \\ \vdots \\ b_{2,N_2} \end{pmatrix}$$

$$g_2(\cdot) =$$

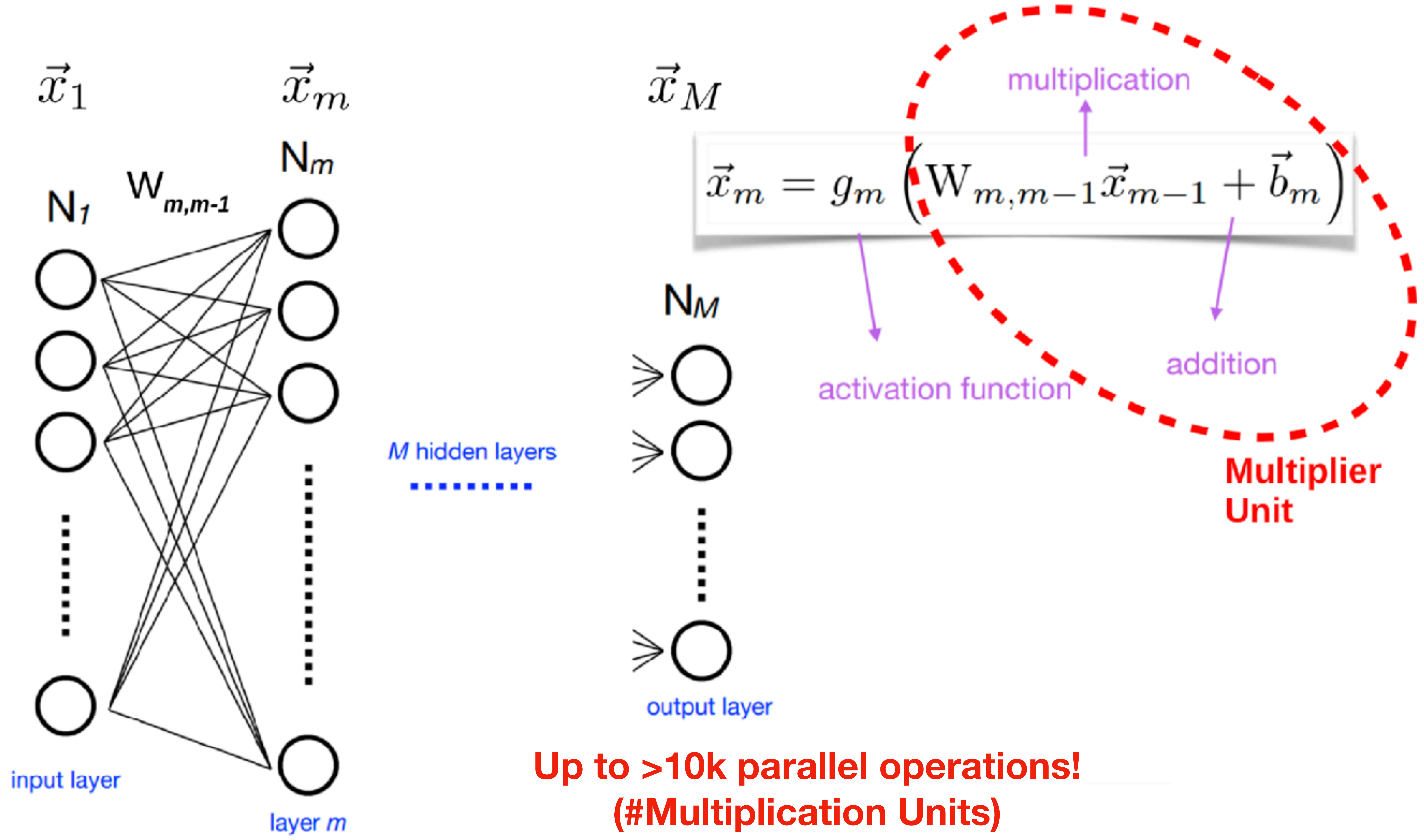


$$\vec{x}_2 = g_2(W_{2,1}\vec{x}_1 + \vec{b}_2)$$

What is a Neural Network?

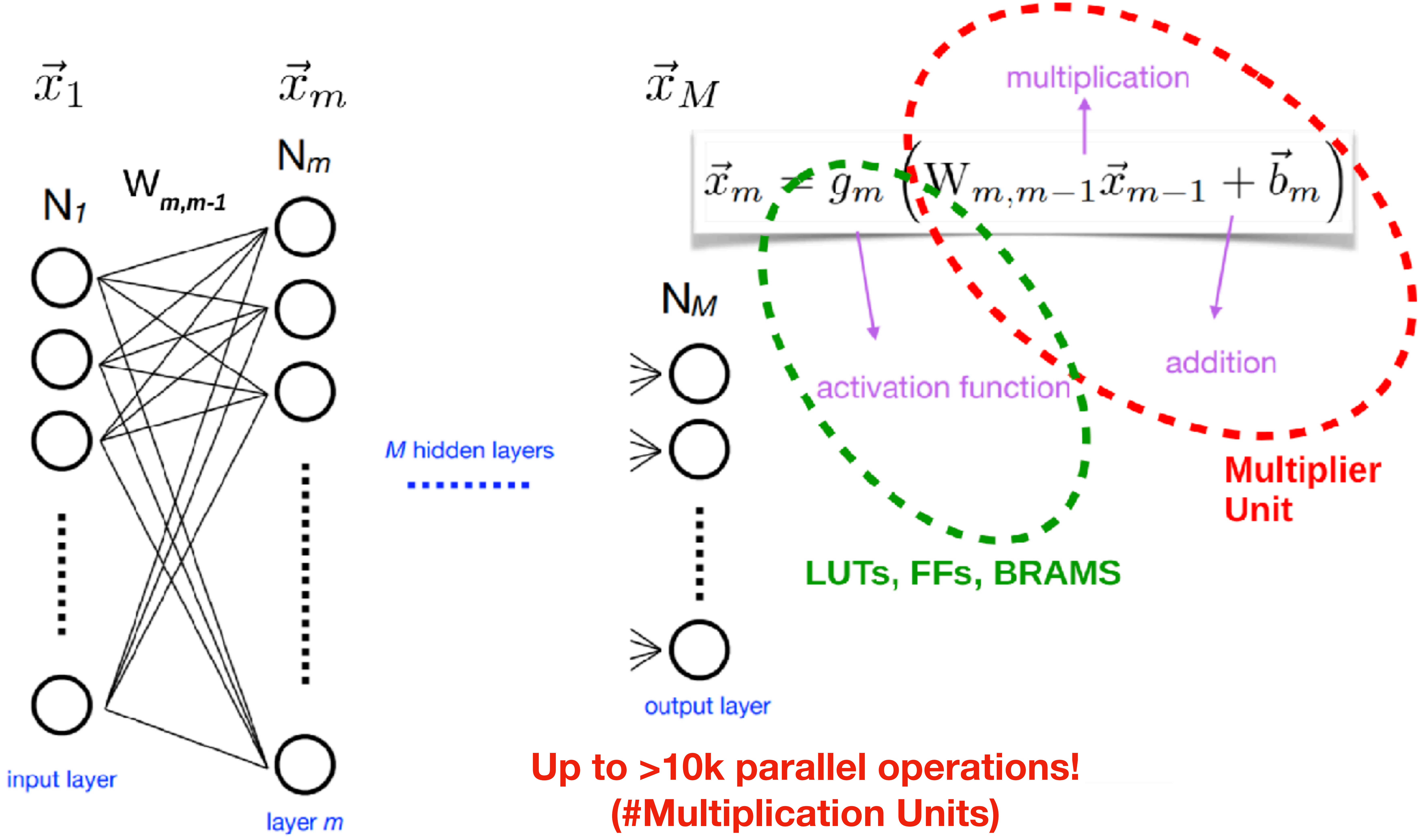


Inference on FPGAs



Up to >10k parallel operations!
(#Multiplication Units)

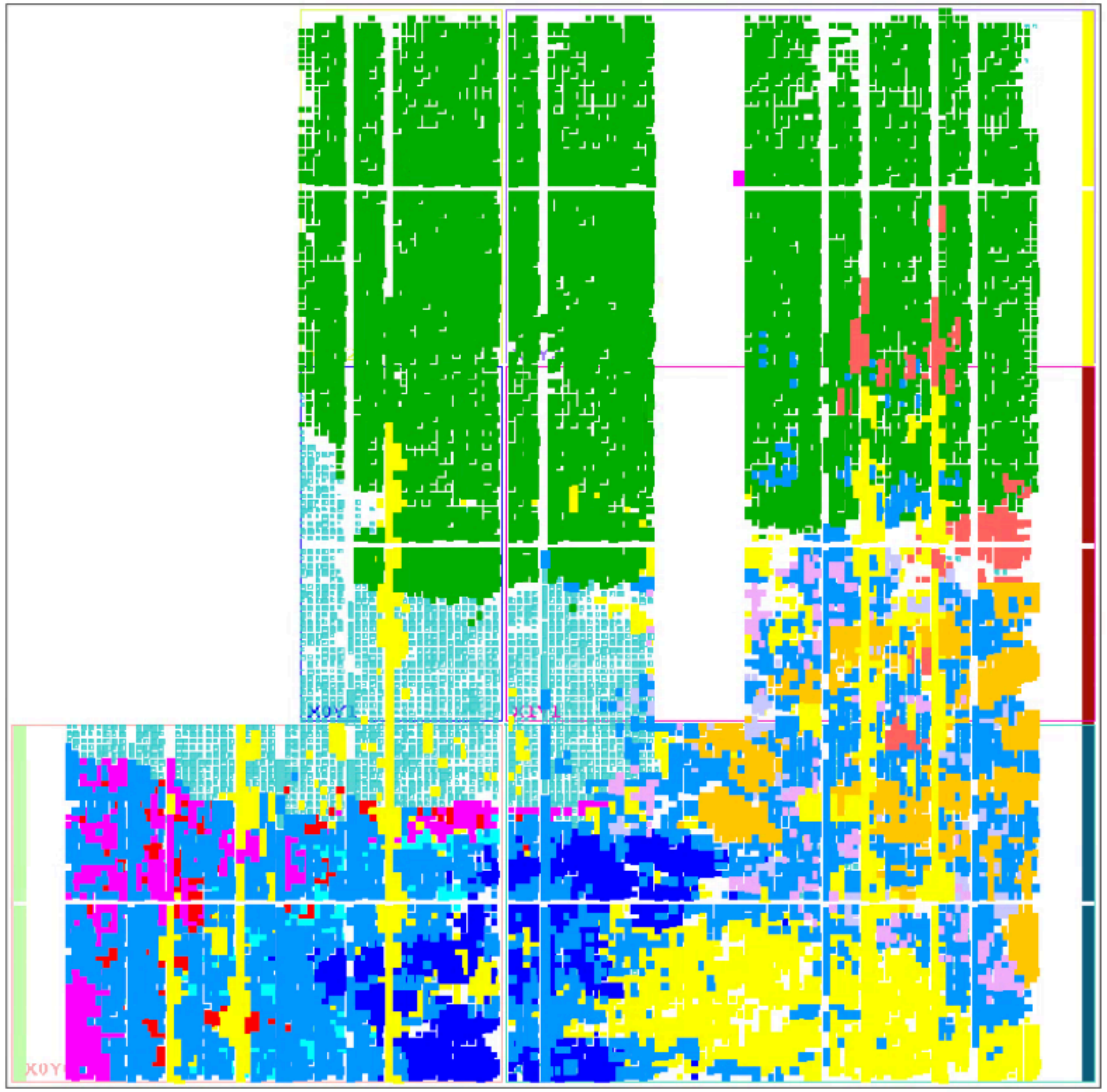
Inference on FPGAs

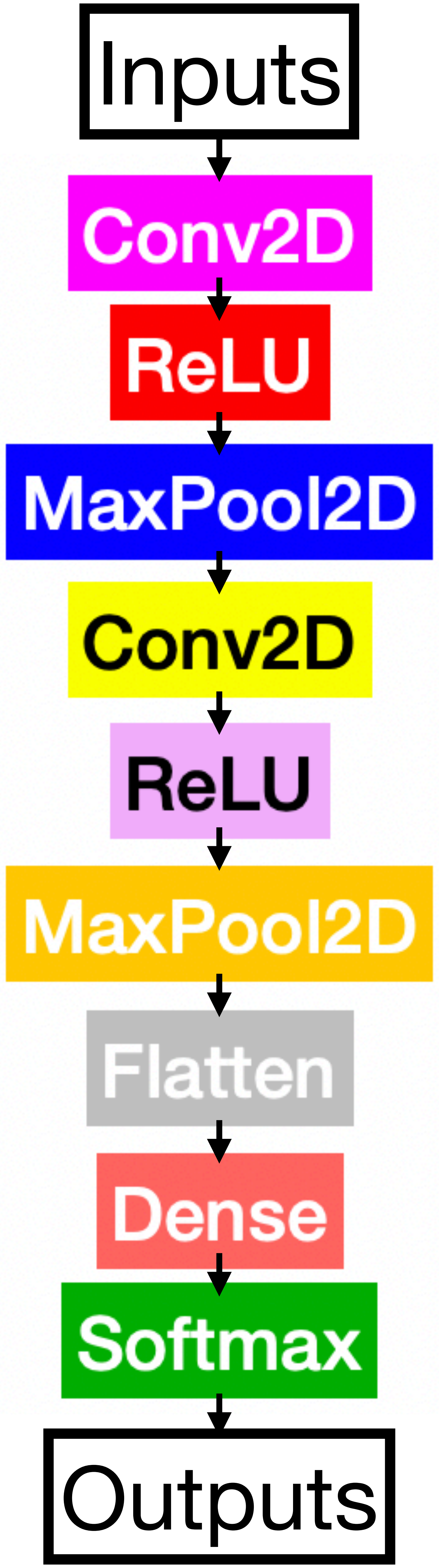


Up to >10k parallel operations!
(#Multiplication Units)

Inference on FPGAs

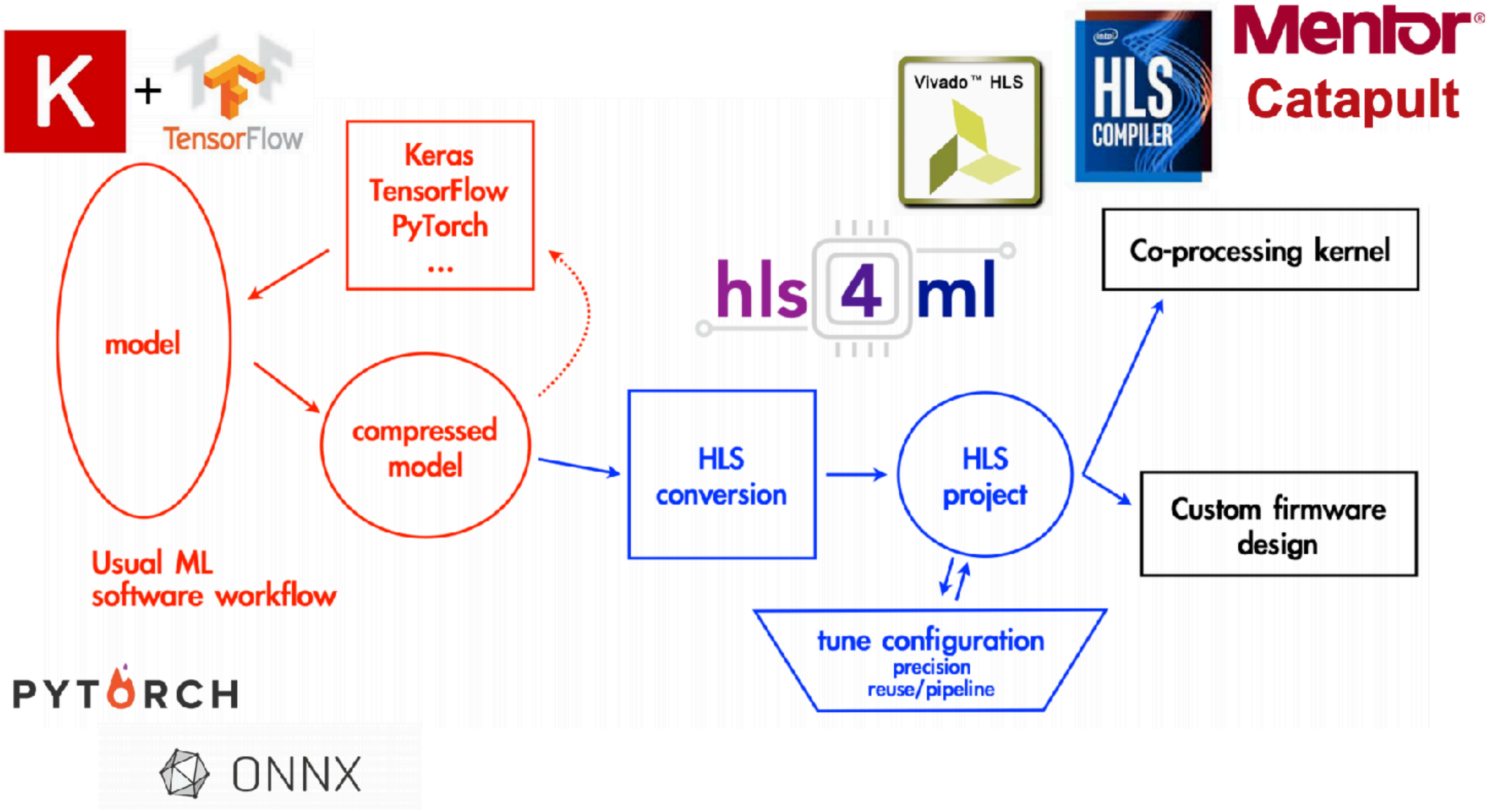
- Each part of network must be placed on the FPGA, connected together
- Cannot implement an algorithm if there are no resources left



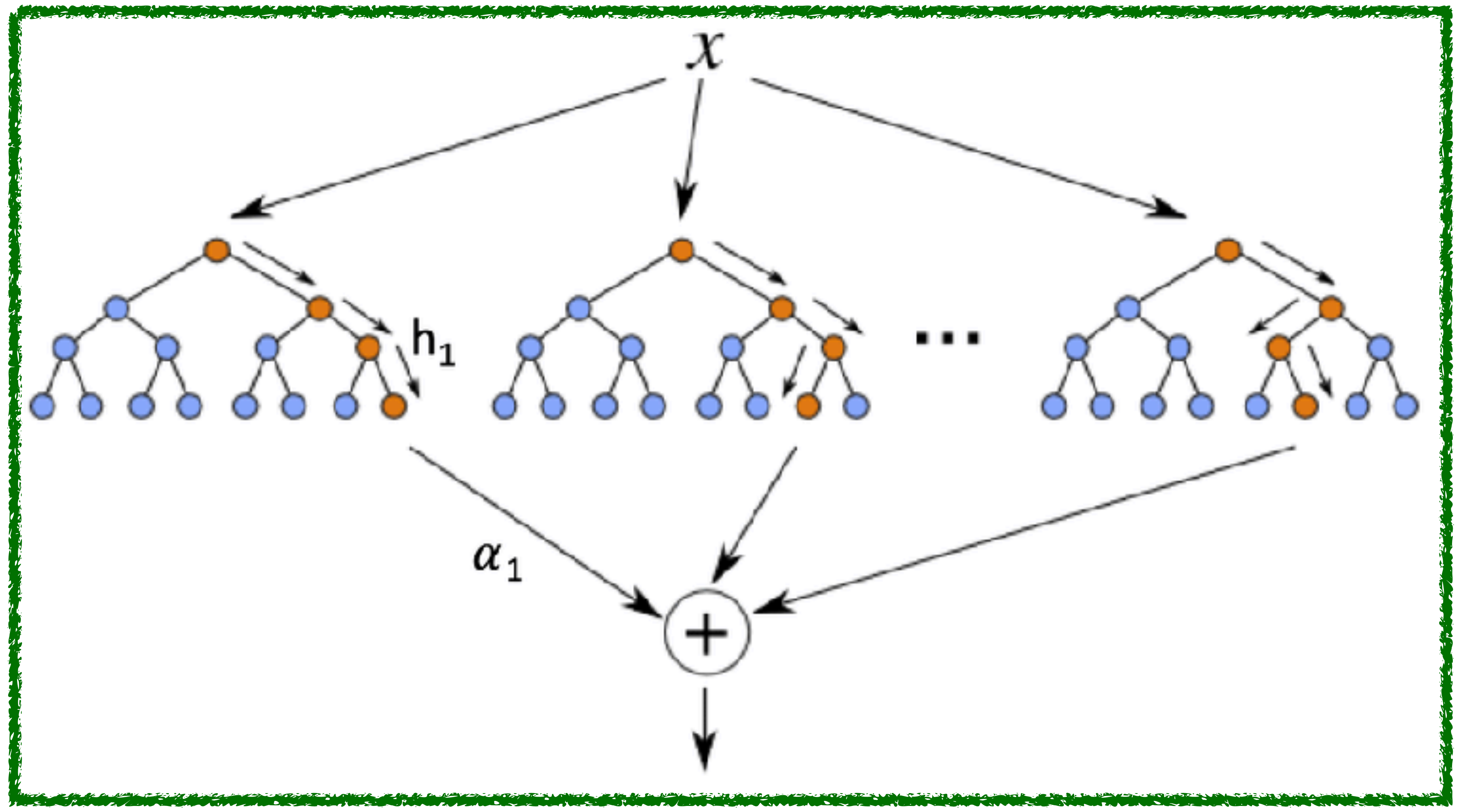


- hls4ml is a software package for automatically creating implementations of neural networks for FPGAs and ASICs
    - https://fastmachinelearning.org/hls4ml/ [arXiv:1804.06913]
    - pip installable
- Supports common layer architectures and model software (keras, tensorflow, pytorch, ONNX)
- Part of larger Fast Machine Learning collaboration

hls4ml Workflow



Many Others



- NNs: FINN

arXiv: 2004.03021

- Boosted Decision Trees (BDTs): Conifer, FWXMachina

arXiv: 2002.02534

arXiv: 2104.03408

- Entirely non-exhaustive list…

ML Size / Complexity

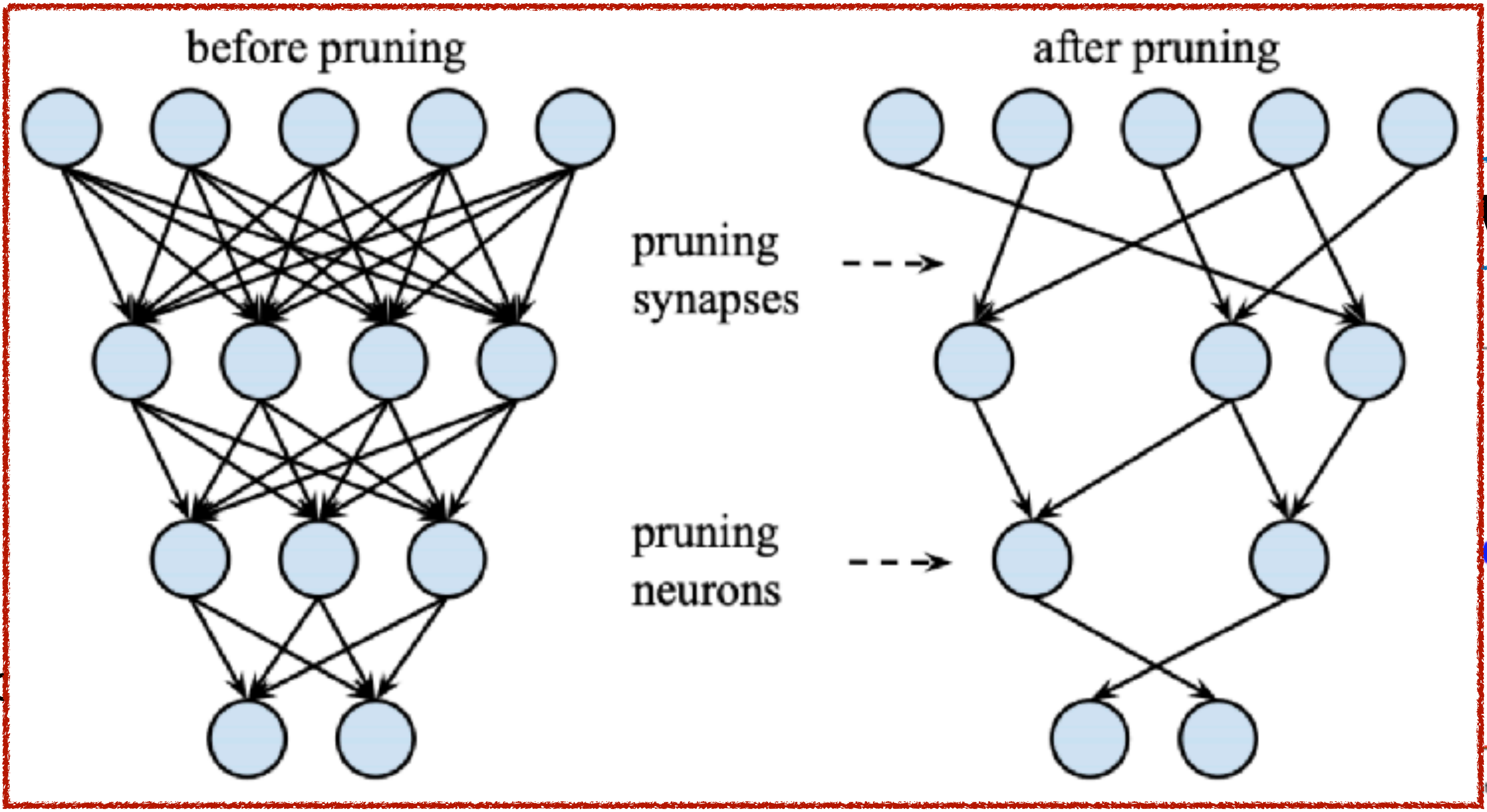
- Regardless of toolkit, limitation of doing low latency ML is FPGA size
  - Bigger FPGA → more resources → more computation

Xilinx Virtex Ultrascale+ VU13P
12288 Multipliers
1.7M LUTs
3.4M FFs
95 Mb BRAM

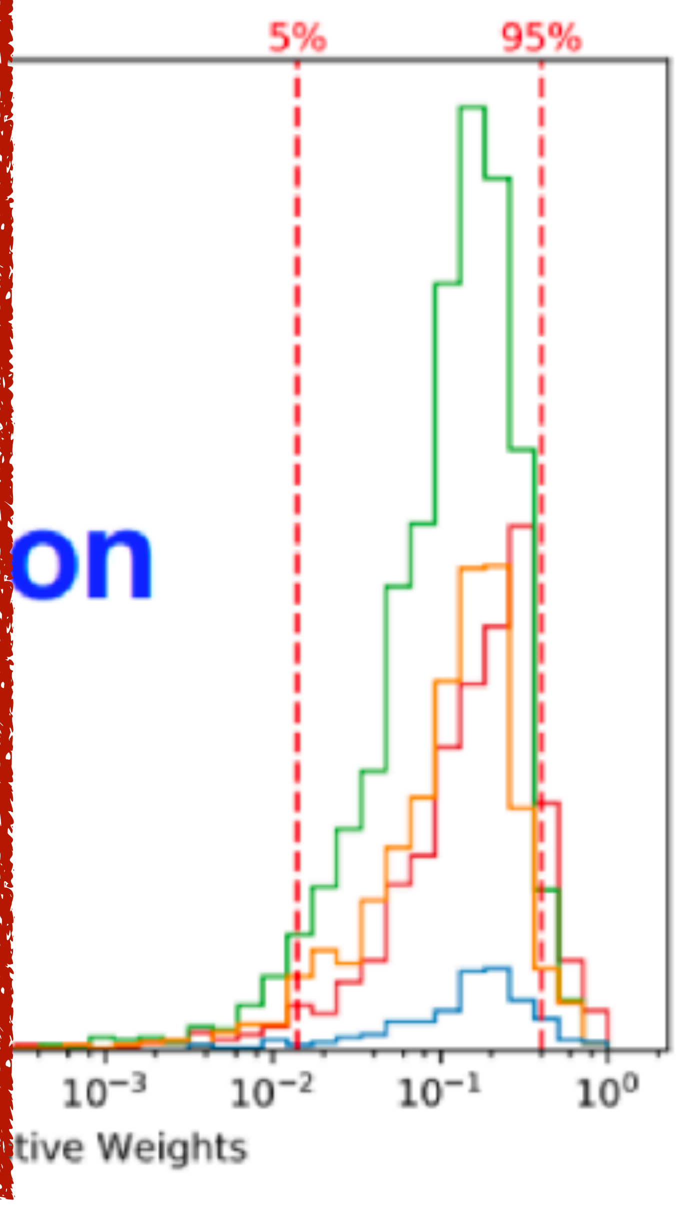
- ***Pruning*** and ***quantization*** are ways to reduce resources
  - Challenge is maintaining performance

Pruning

- Are all the pieces a given network necessary?
- M
- M
- M
fr

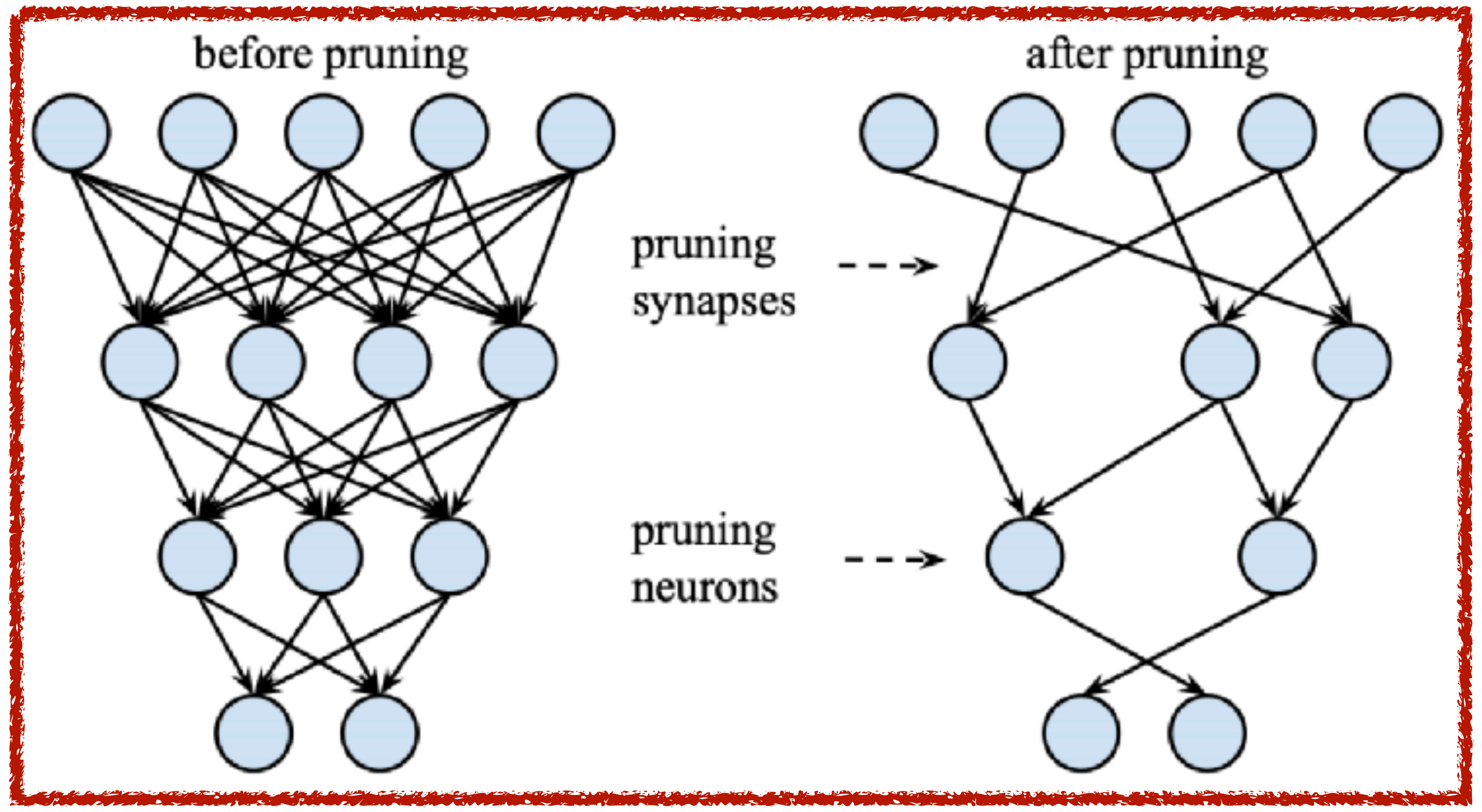


$) + \lambda||\mathbf{w}||$

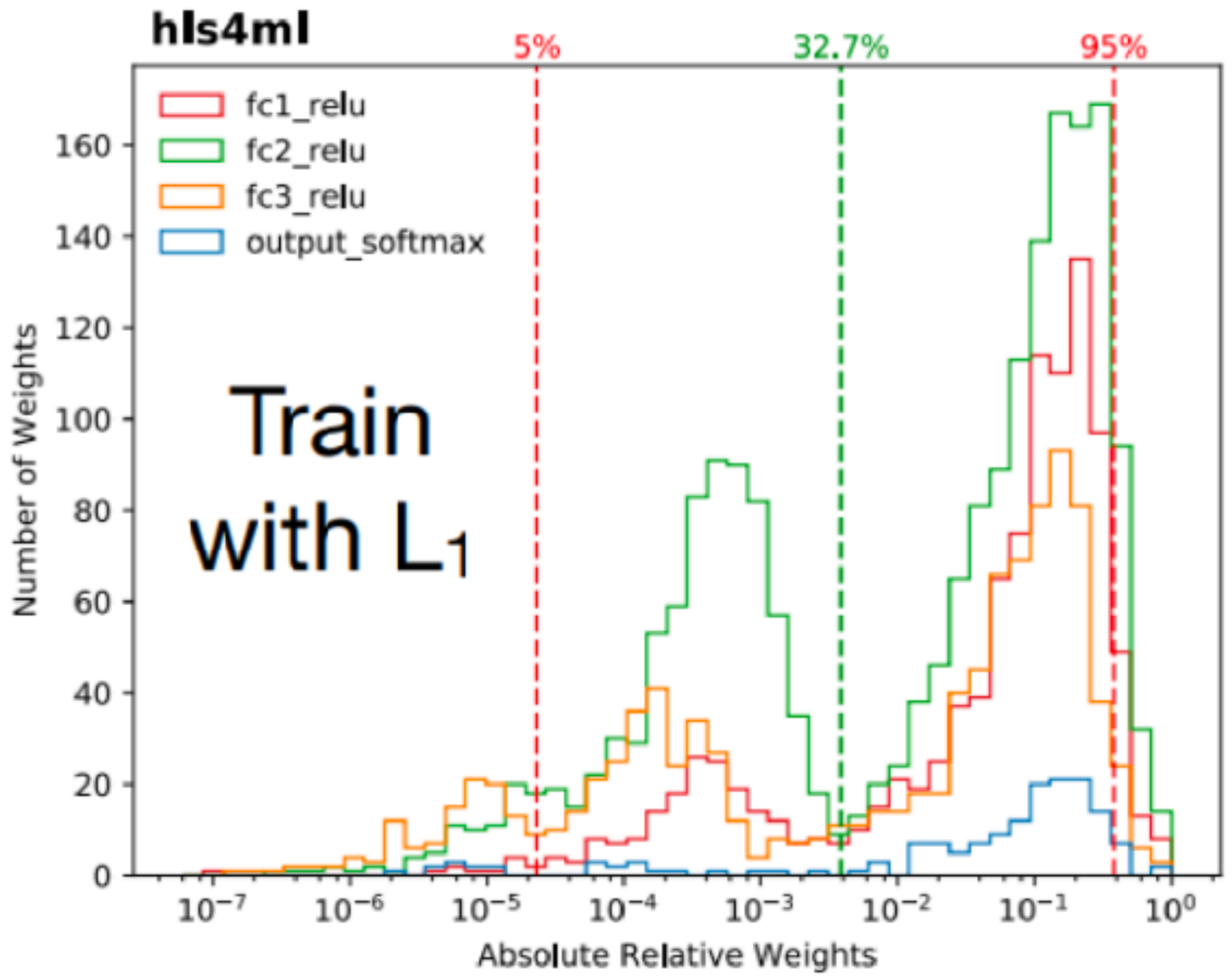


Pruning

- Are all the pieces a given network necessary?
- Many different types of pruning
- Magnitude-based:
    - Use regularization (penalty term in loss function for large weights)
    - Remove smallest weights
    - Repeat
- Multiplications by 0 can be completely removed from FPGA design

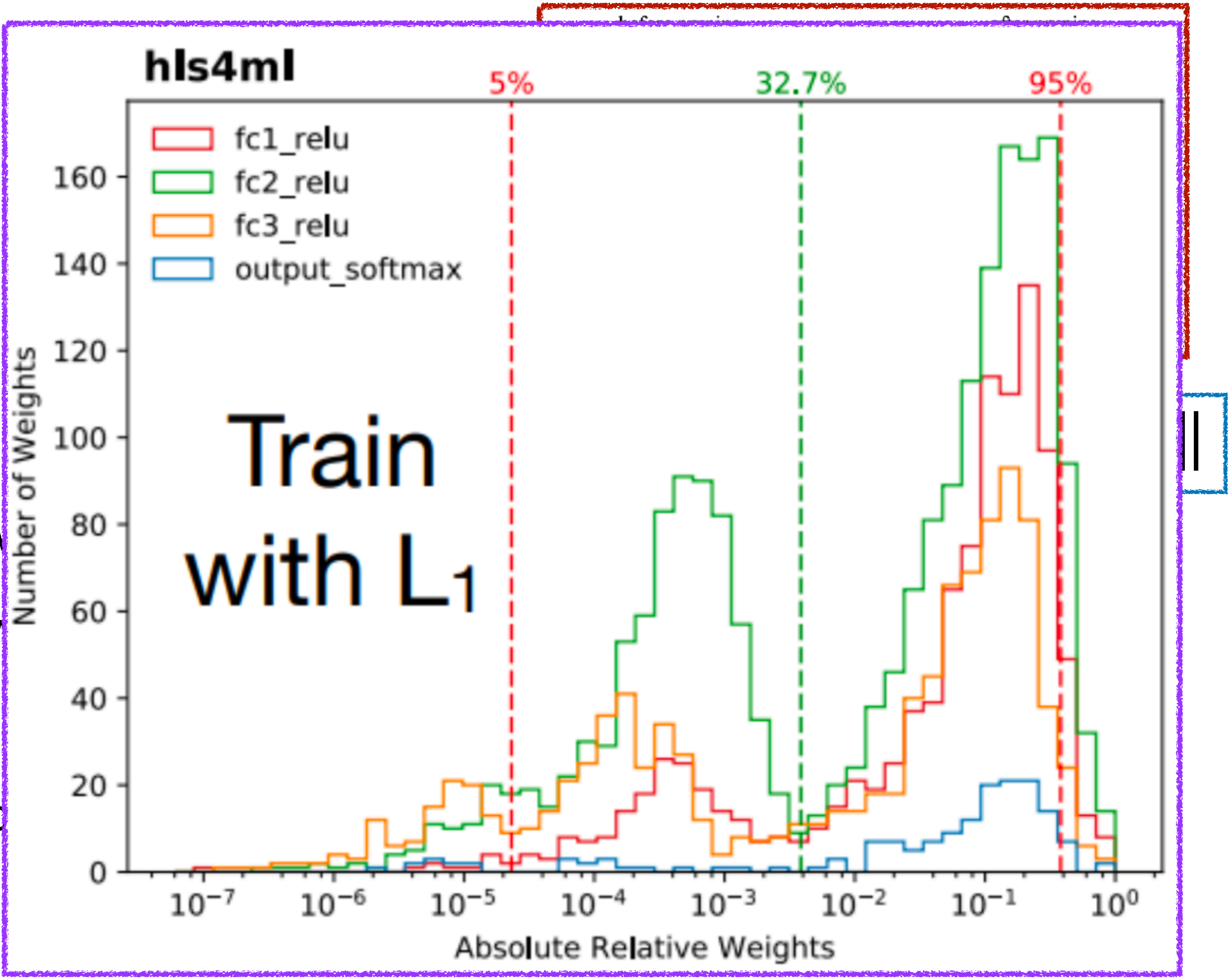


$$L_\lambda(\mathbf{w}) = L(\mathbf{w}) + \lambda||\mathbf{w}||$$



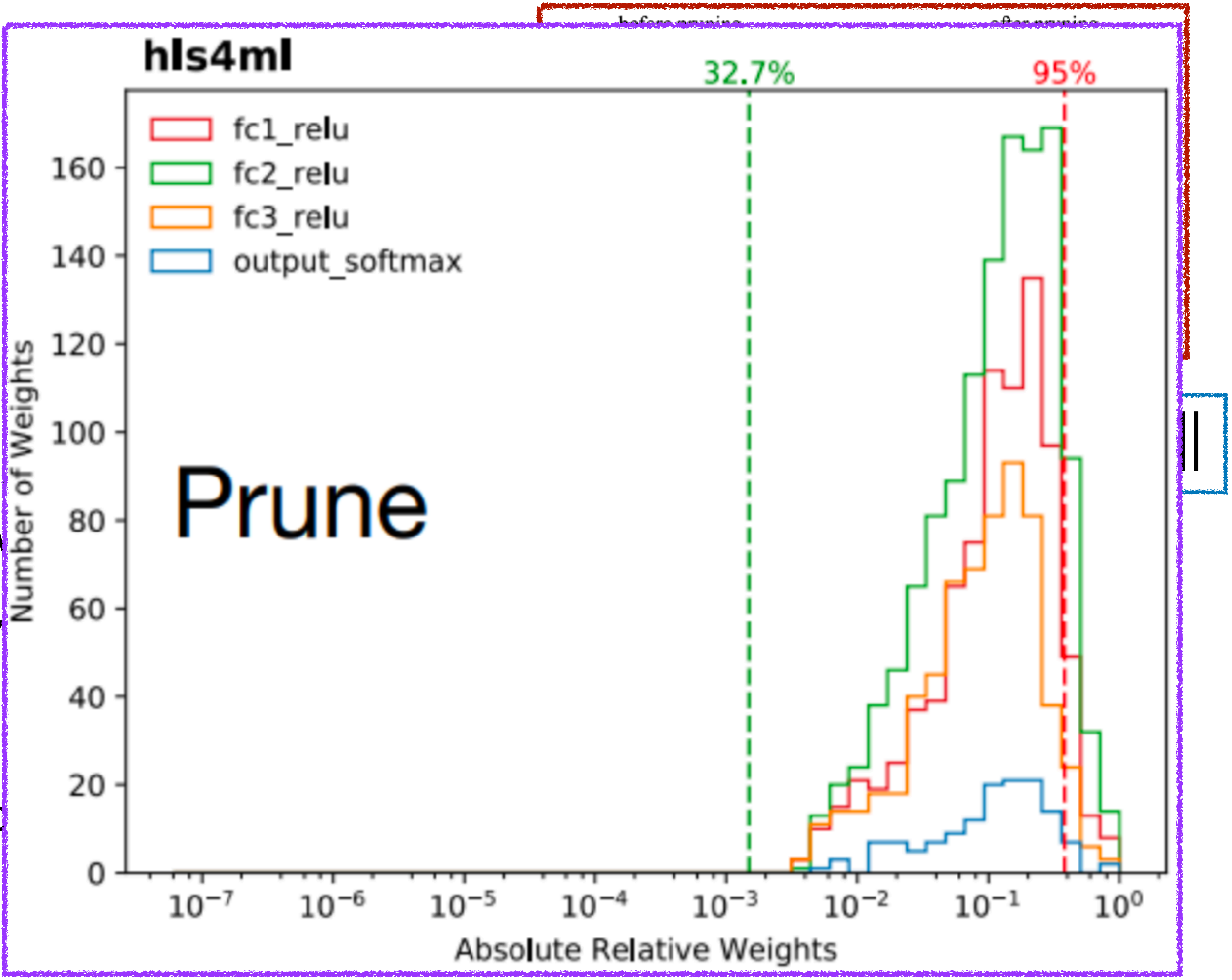
Pruning

- Are all the pieces a g
- Many different types
- Magnitude-based:
    - Use regularization
      function for large w
    - Remove smallest w
    - Repeat
- Multiplications by 0 c
  from FPGA design



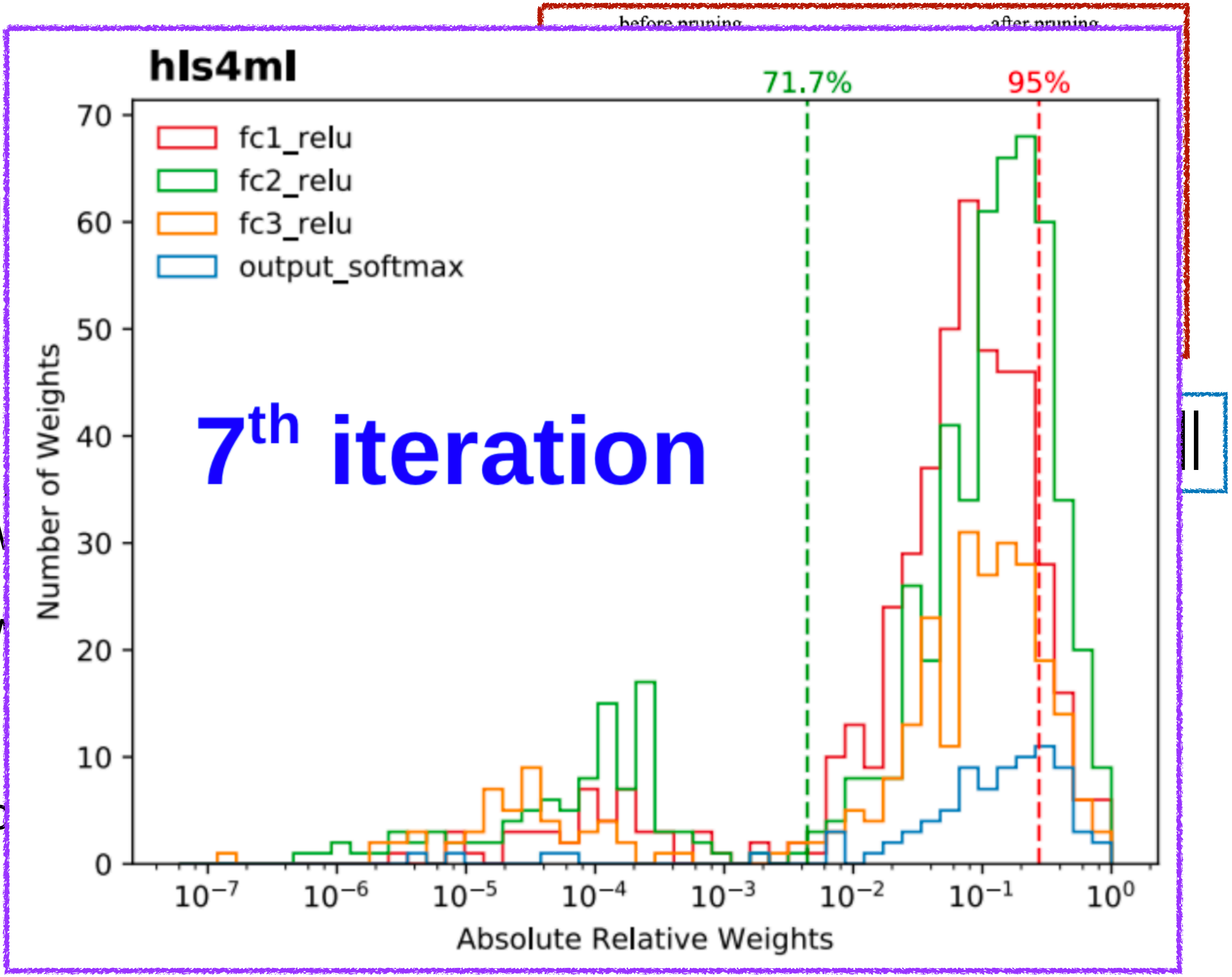
Pruning

- Are all the pieces a g
- Many different types
- Magnitude-based:
  - Use regularization
    function for large w
  - Remove smallest w
  - Repeat
- Multiplications by 0 c
  from FPGA design



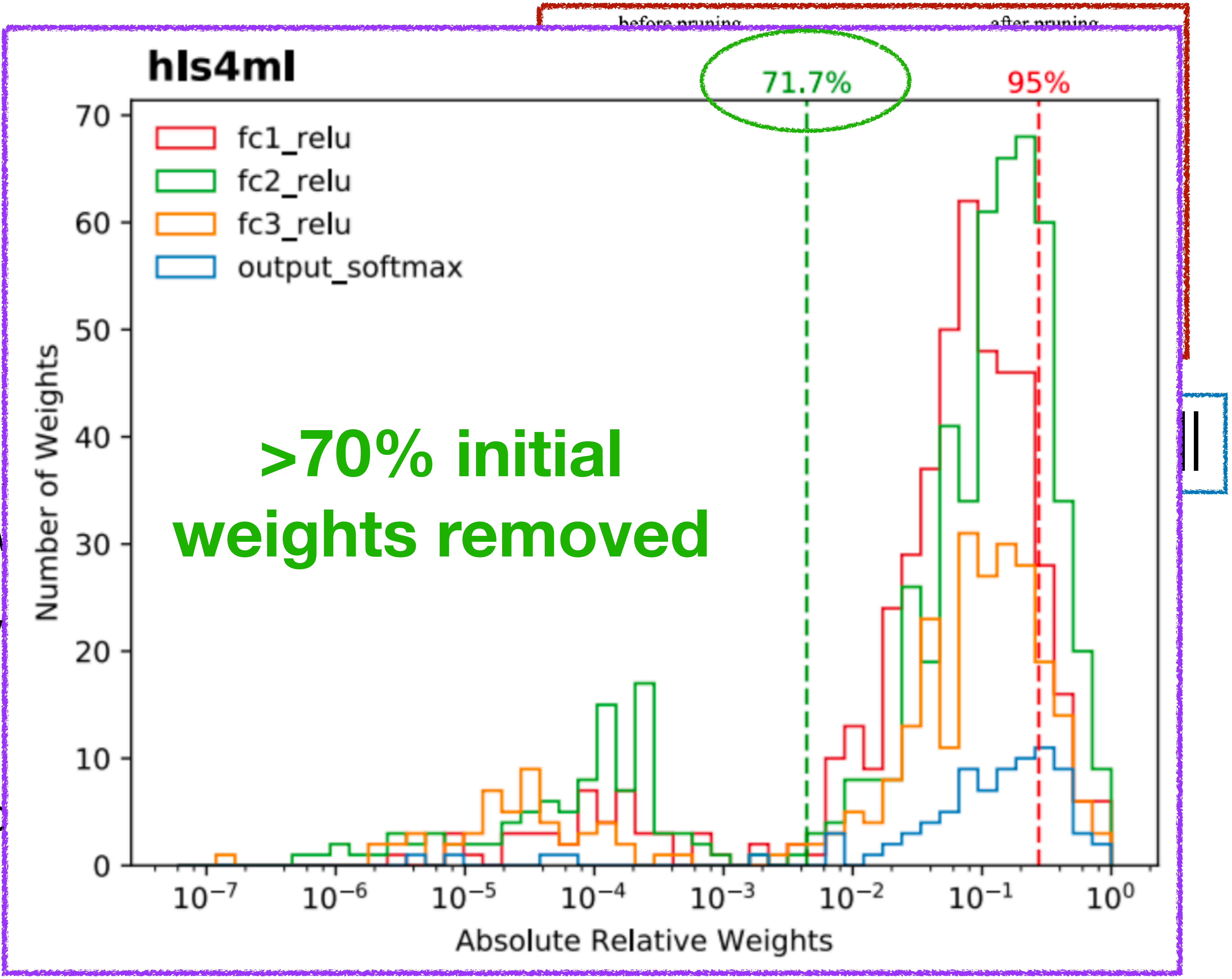
Pruning

- Are all the pieces a g
- Many different types
- Magnitude-based:
  - Use regularization function for large w
  - Remove smallest w
  - Repeat
- Multiplications by 0 c from FPGA design



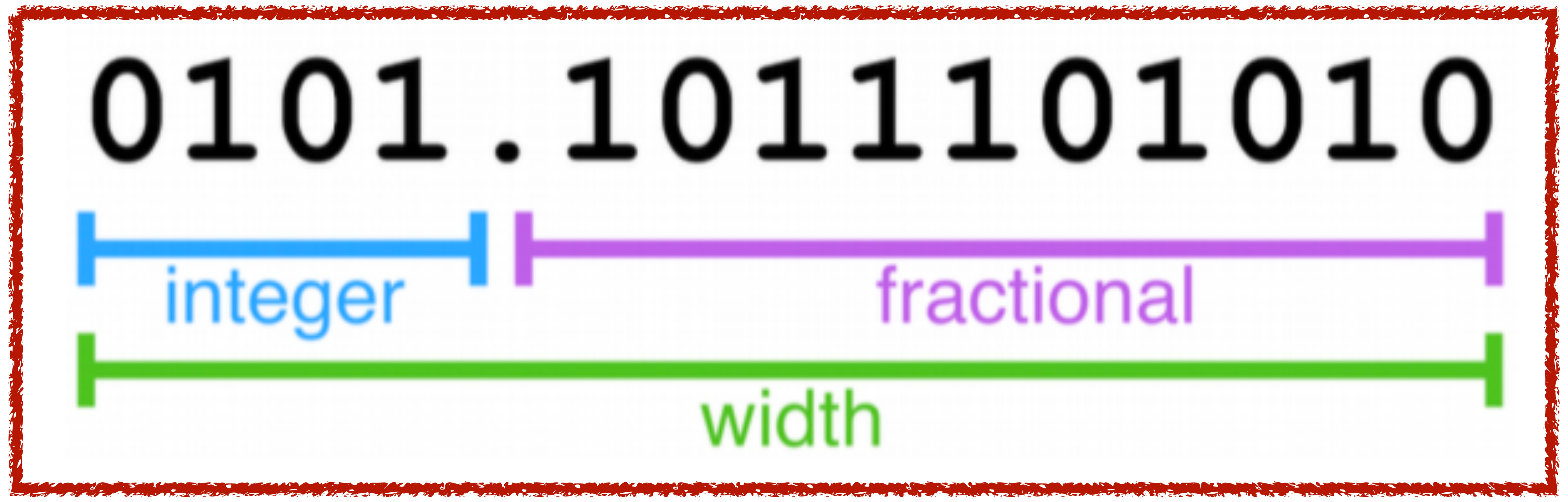
Pruning

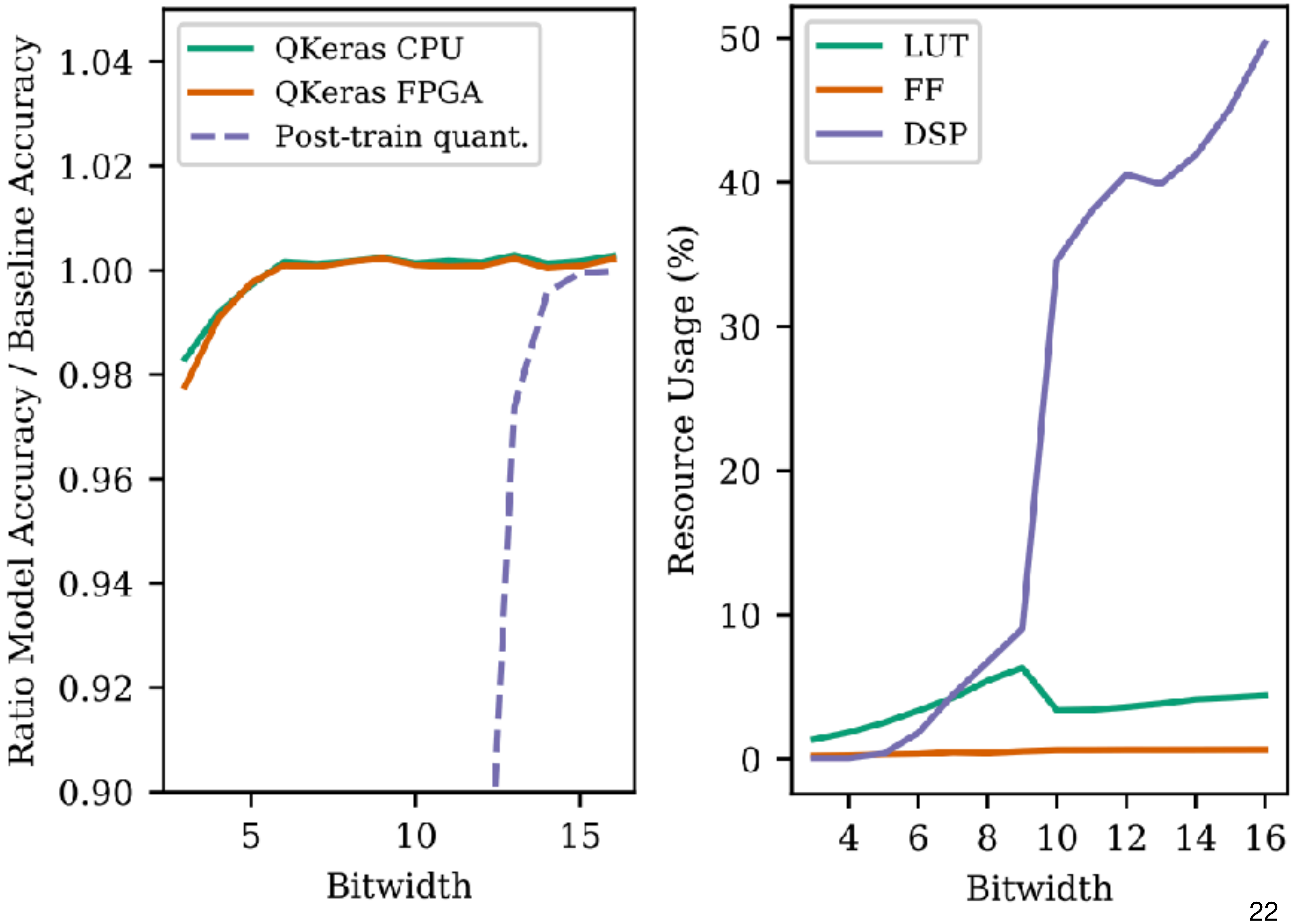
- Are all the pieces a g
- Many different types
- Magnitude-based:
    - Use regularization function for large w
    - Remove smallest w
    - Repeat
- Multiplications by 0 c from FPGA design



Quantization

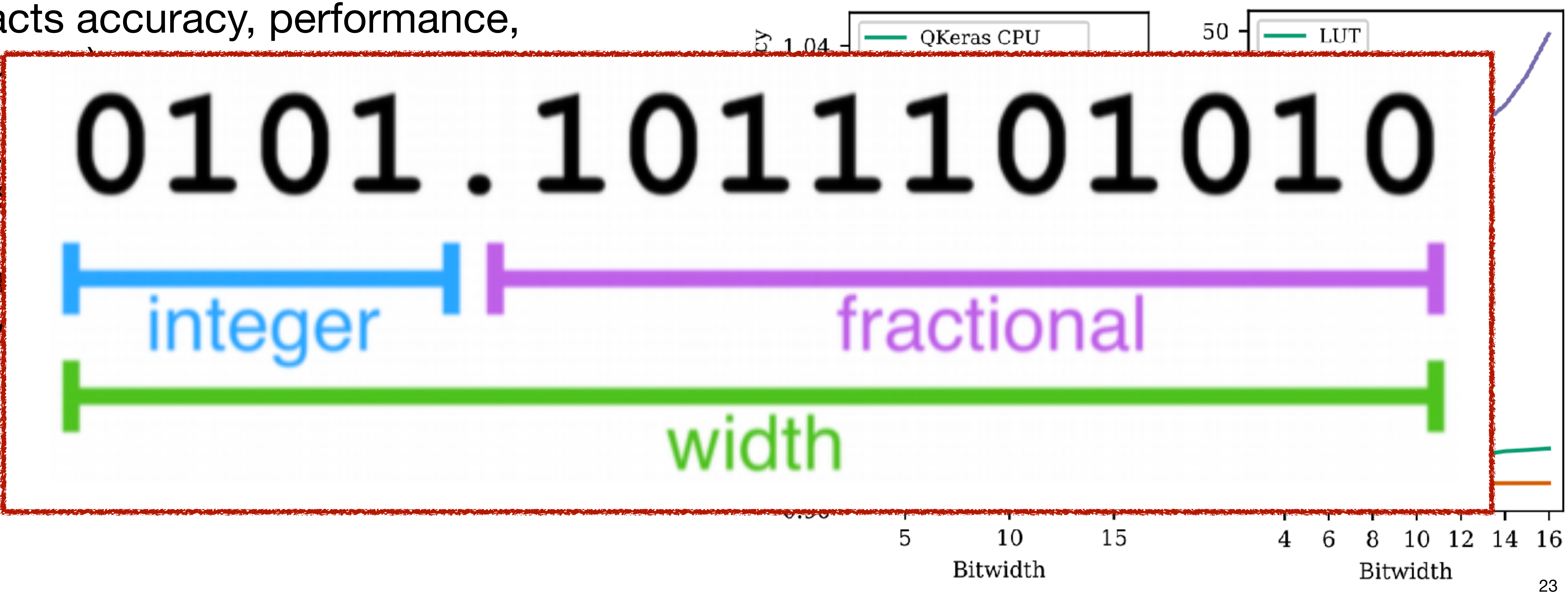
- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, resources)
  - Can be combined with other customizations
- Quantization-aware training [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization





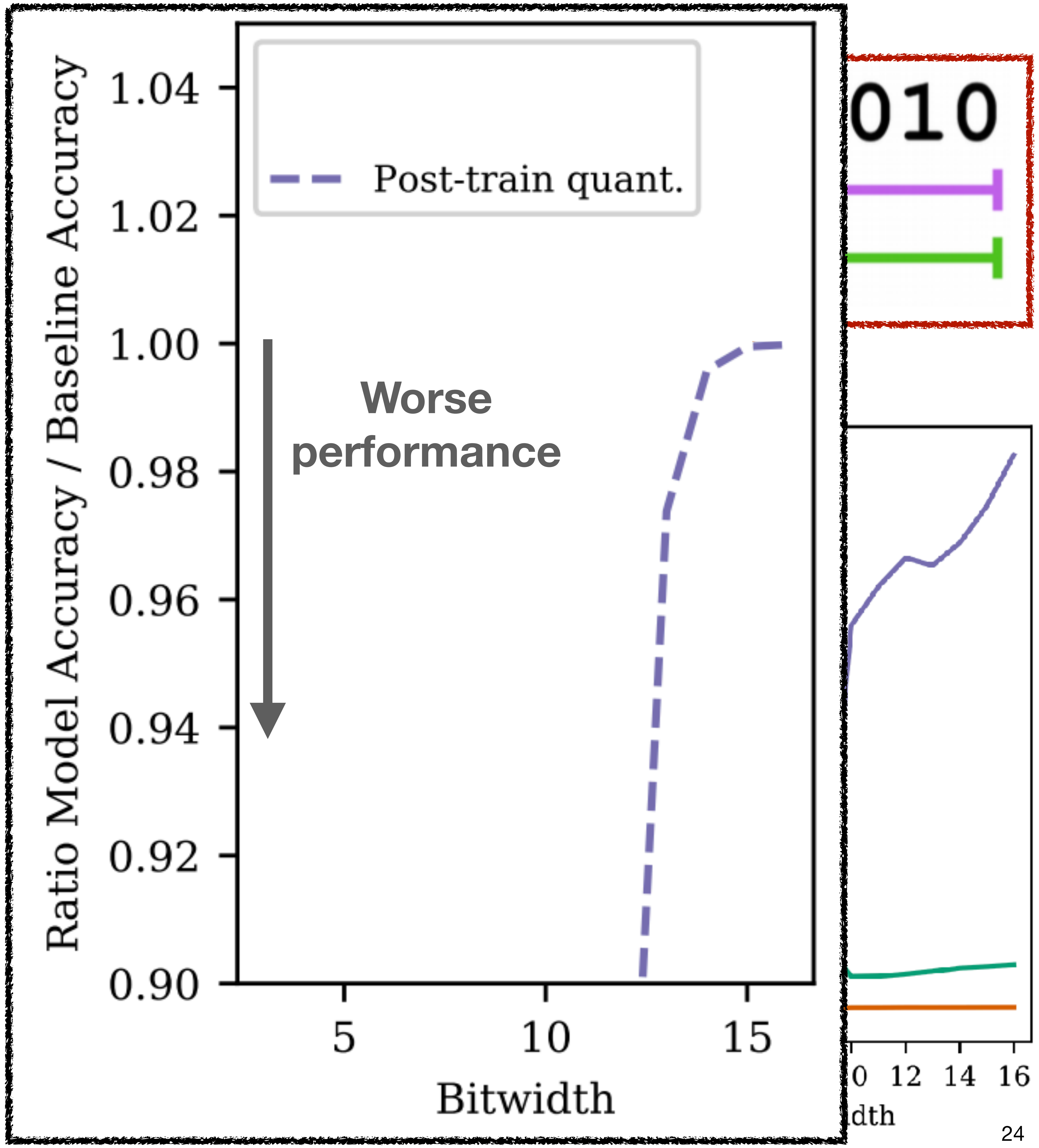
Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, resou
  - Ca
    cus
- Quan
  [arXiv
  - Ca
    tra



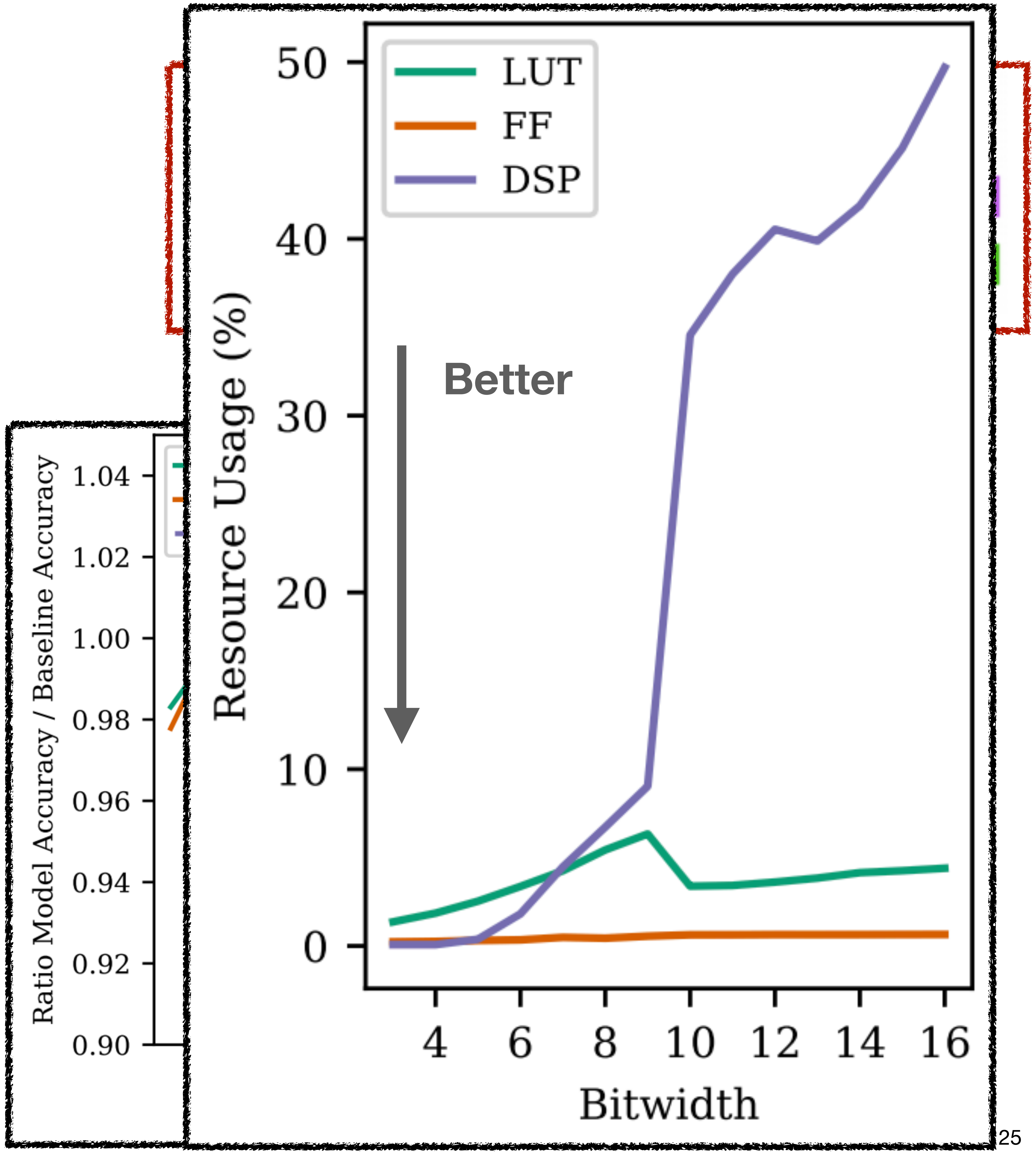
Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, **performance**, resources)
    - Can be combined with other customizations
- Quantization-aware training [arXiv:2006.10159]
    - Can greatly reduce size of network by training with knowledge of quantization



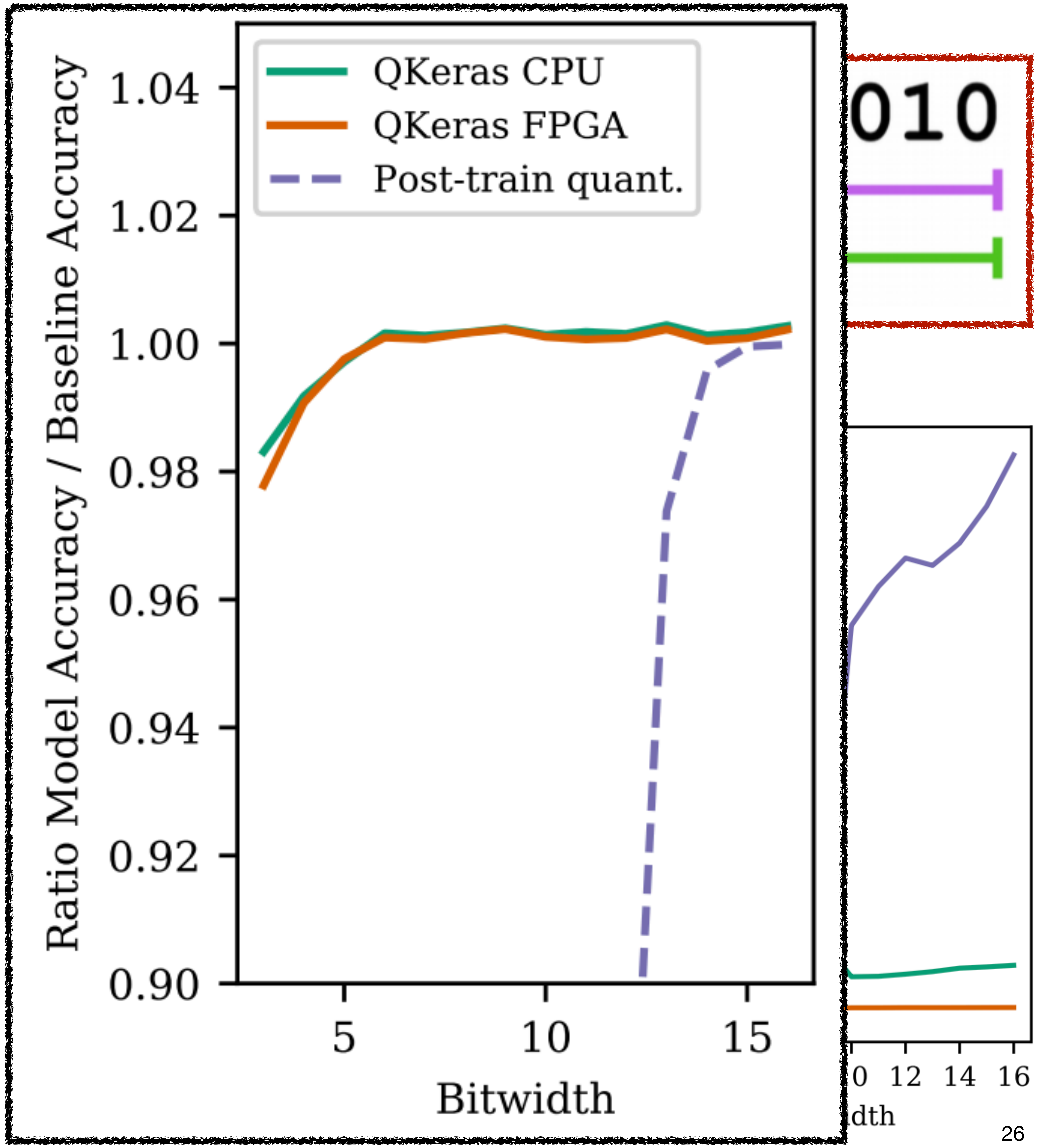
Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, **resources**)
  - Can be combined with other customizations
- Quantization-aware training [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization



Quantization

- FPGAs are well suited to fixed-point numbers, not floating point
- Bitwidth can be adjusted as needed (impacts accuracy, performance, resources)
  - Can be combined with other customizations
- **Quantization-aware training** [arXiv:2006.10159]
  - Can greatly reduce size of network by training with knowledge of quantization



An Application

Large Hadron Collider (LHC)

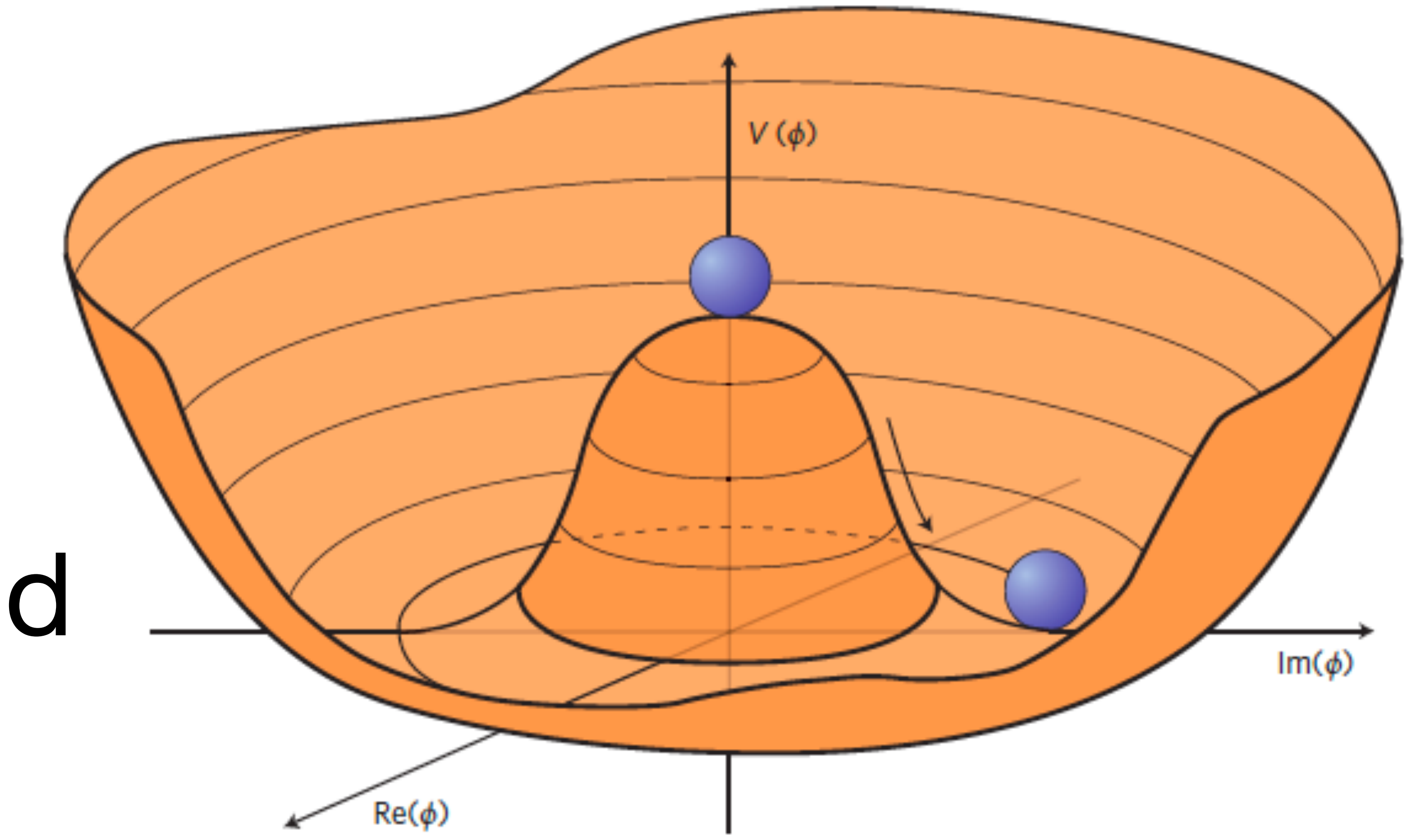
LHC Data Processing / Readout

LHC Data Processing / Readout

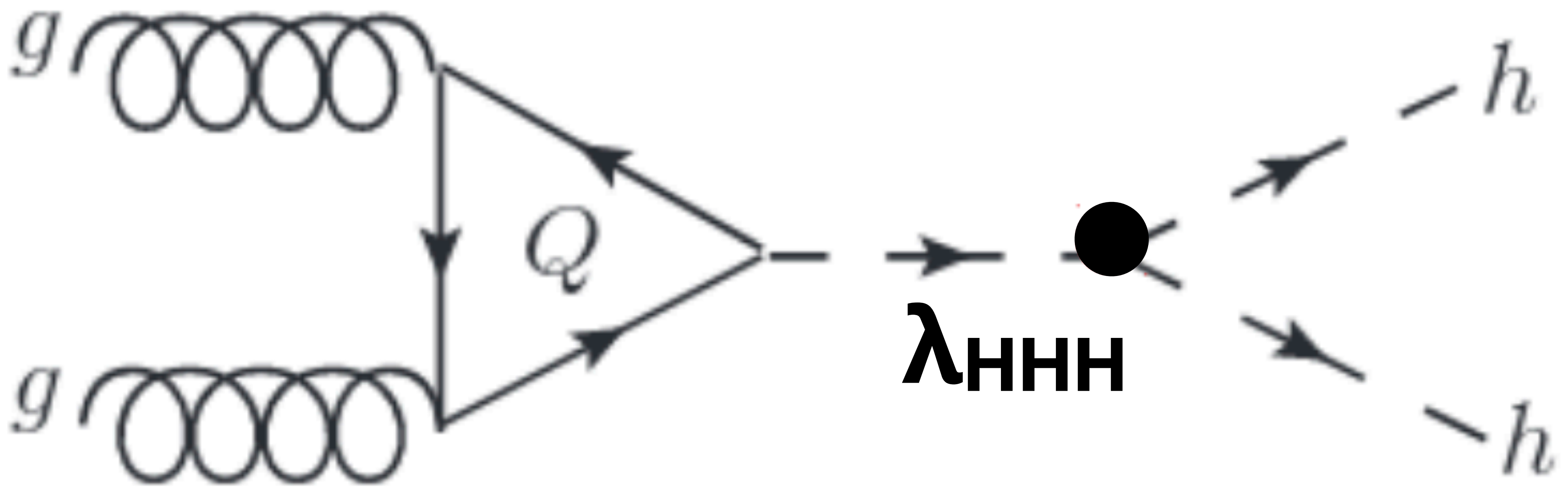
Must use FPGAs, enforced by latency requirement

Need to make decisions ultra fast or physics suffers!

Di-Higgs Production



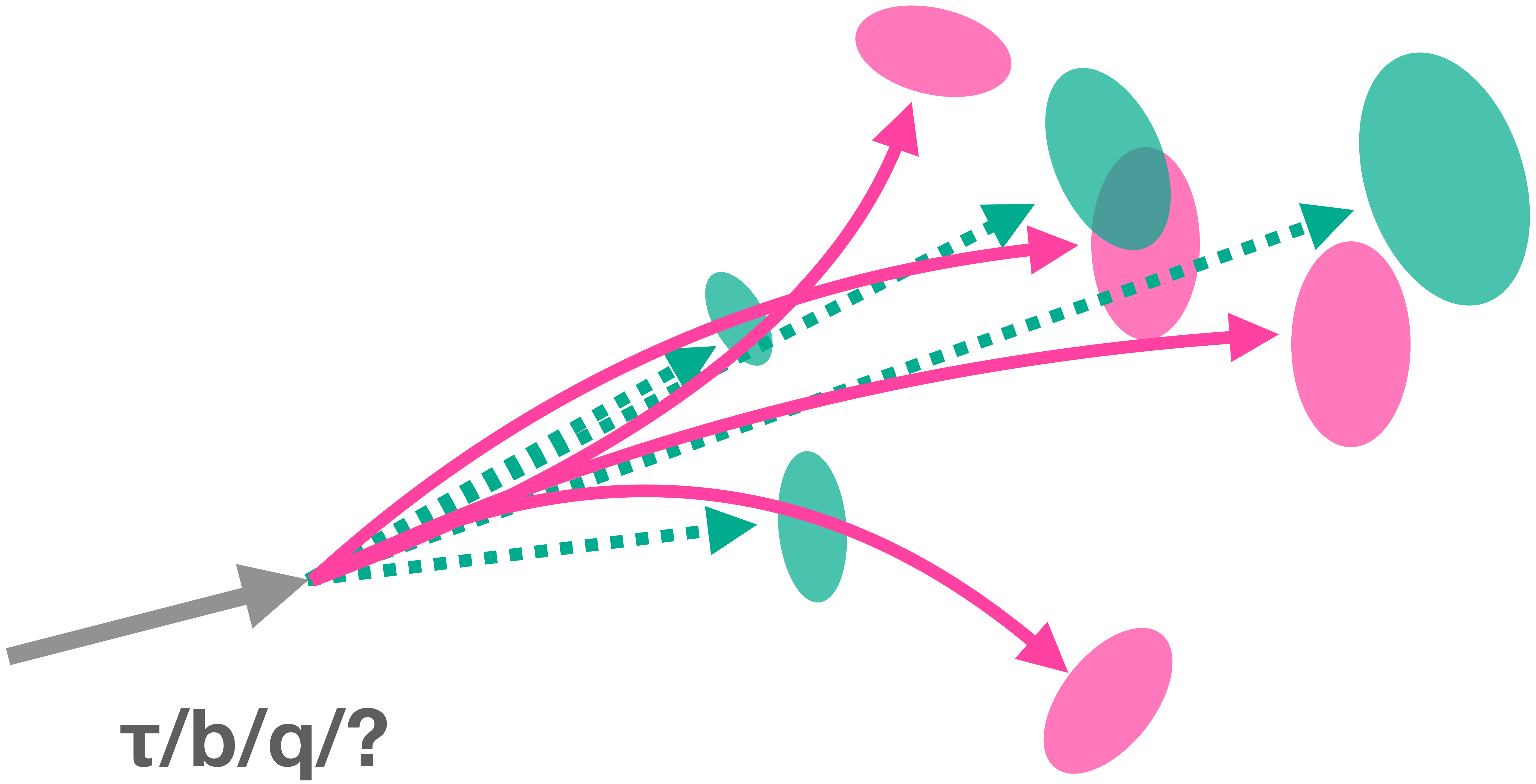
- HH is best way to measure scalar potential of Higgs field
  - Higgs self-coupling: λ_{HHH}
- Major target for CMS and ATLAS experiments
- **bbbb**, **bbWW**, **bbττ** have largest branching ratios
  - Lots of background that mimics these signals → very difficult to record

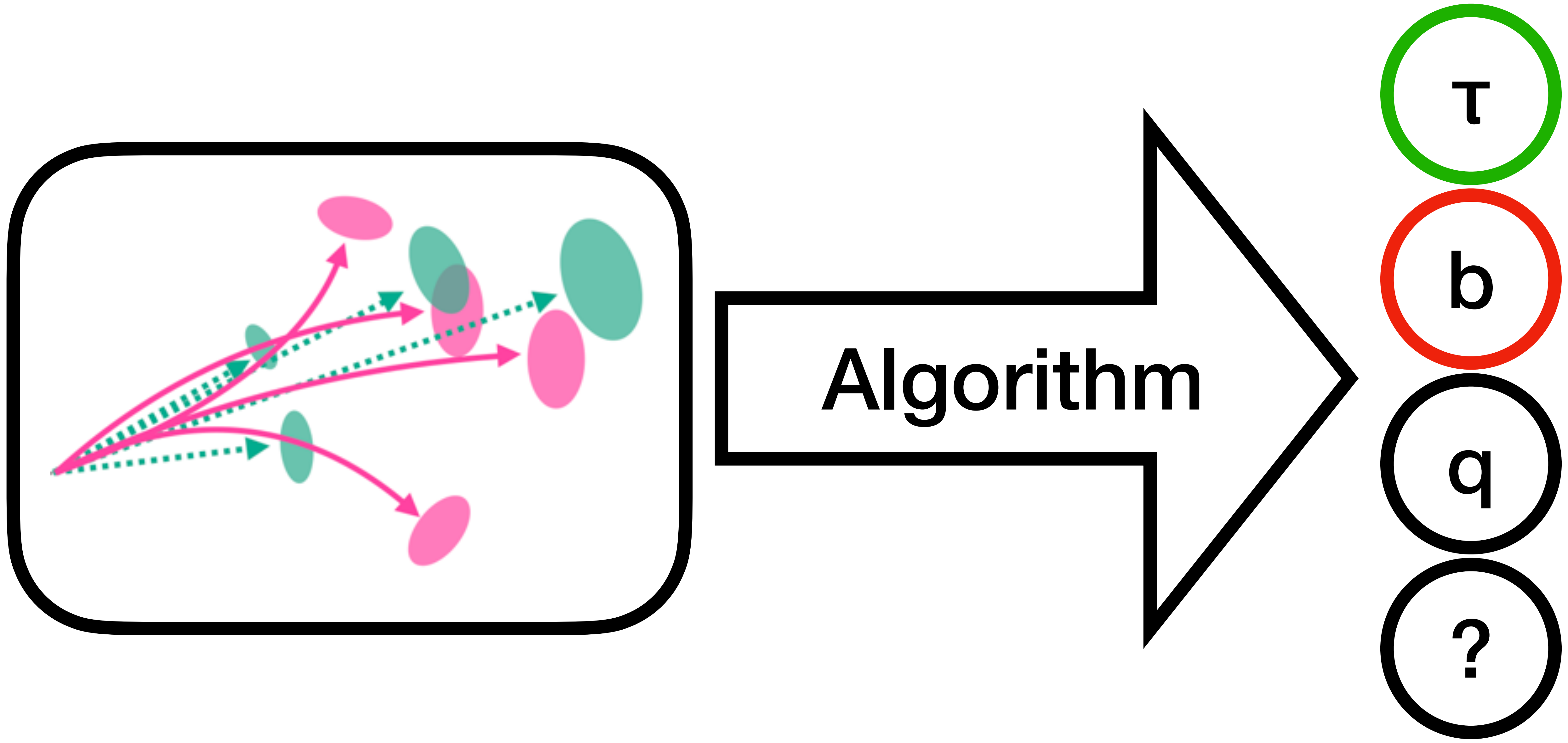


	bb	WW	ττ	ZZ	γγ
bb	33%				
WW	25%	4.6%			
ττ	7.4%	2.5%	0.39%		
ZZ	3.1%	1.2%	0.34%	0.076%	
γγ	0.26%	0.10%	0.029%	0.013%	0.0005%

Particle Identification

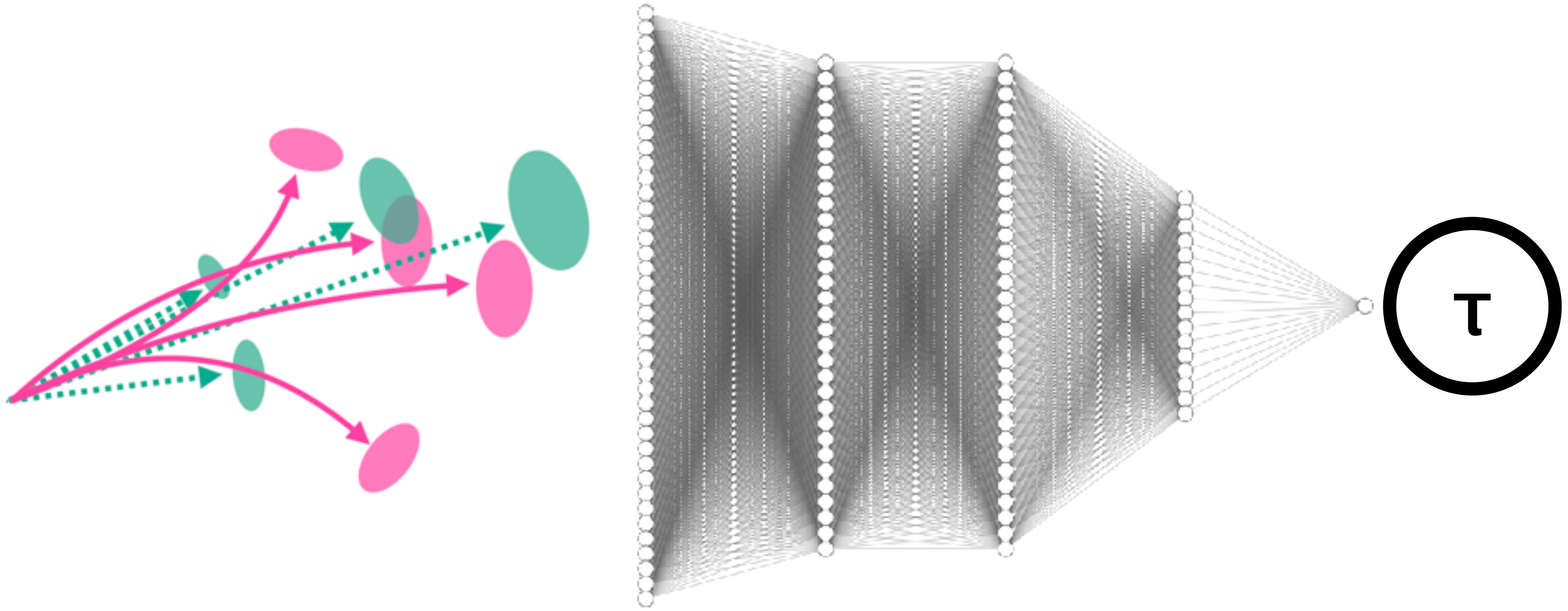
- HH → bbbb, bbττ
- Can we design algorithms to differentiate different collections of particles / detector signals
  - τ lepton, bottom quark
  - Light quarks, gluons, noise, combinatorics
- Can we do it every 25 ns on FPGAs?

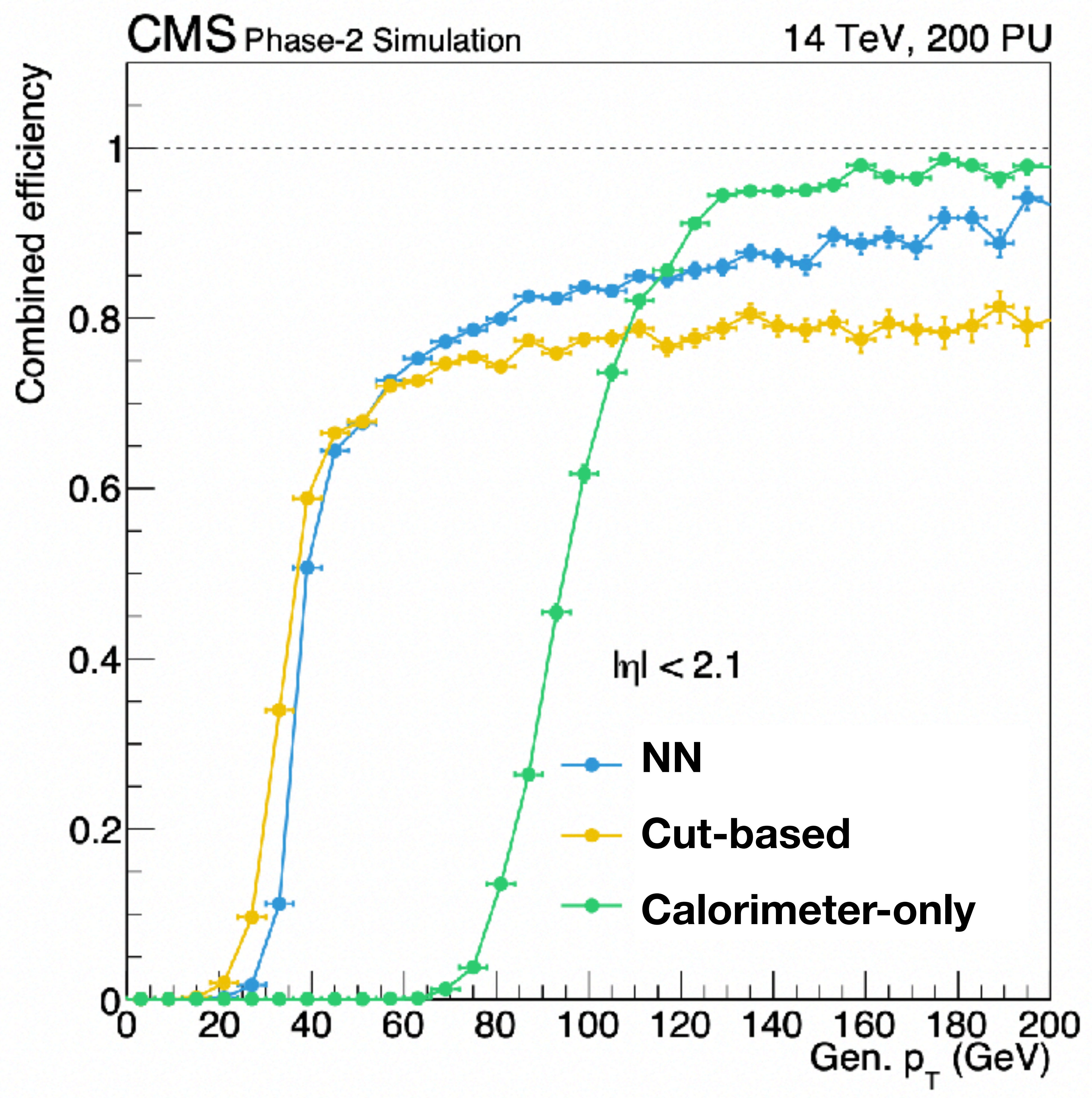




L1 τ Identification

- **NN algorithm** capable of accepting more τ leptons than traditional **cut-based method**
- Network is 3 layer dense model, uses information about particle p_T, η, ϕ, and type
- Outputs decision in 38 ns (9 clocks @ 240 MHz)

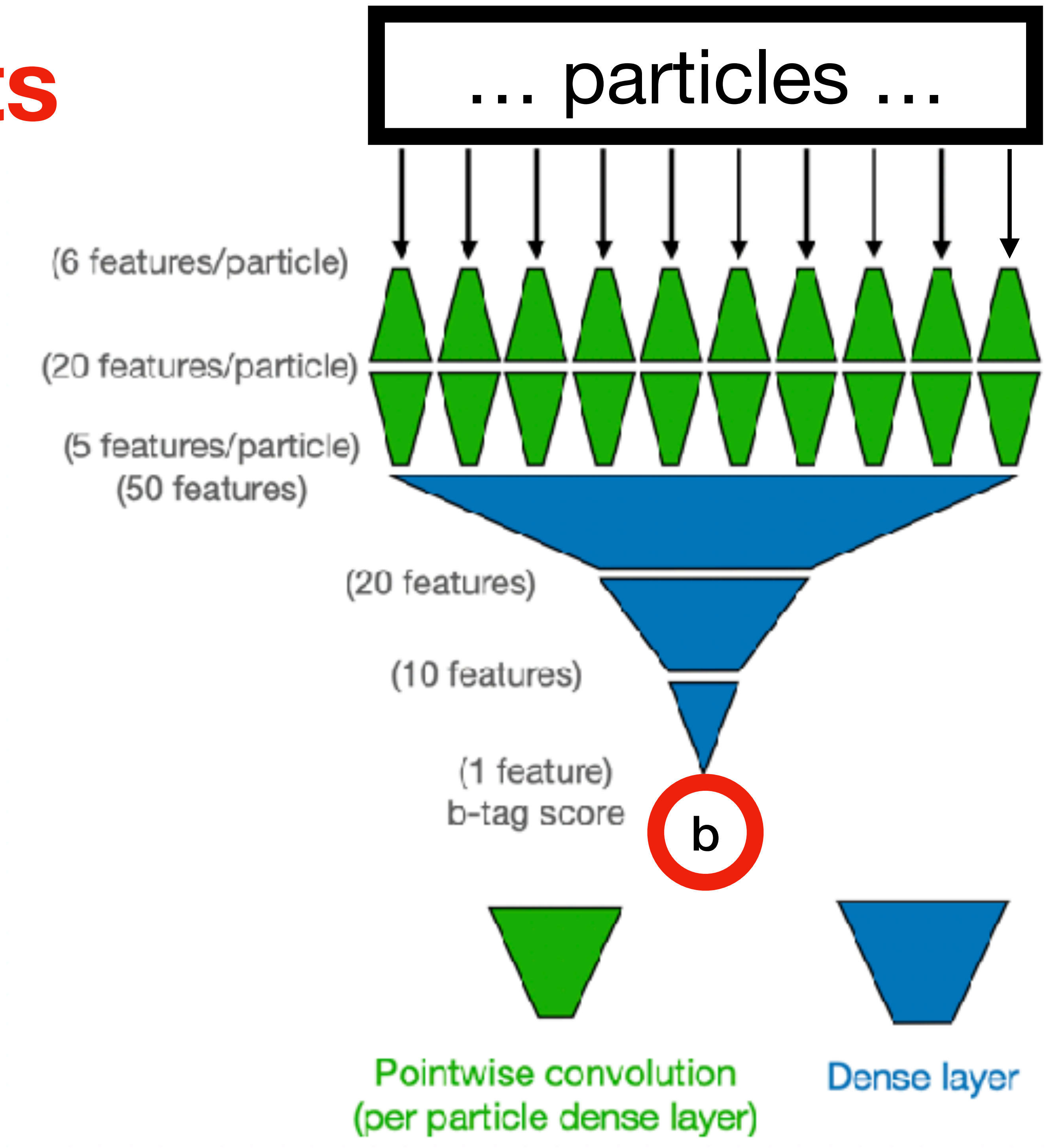


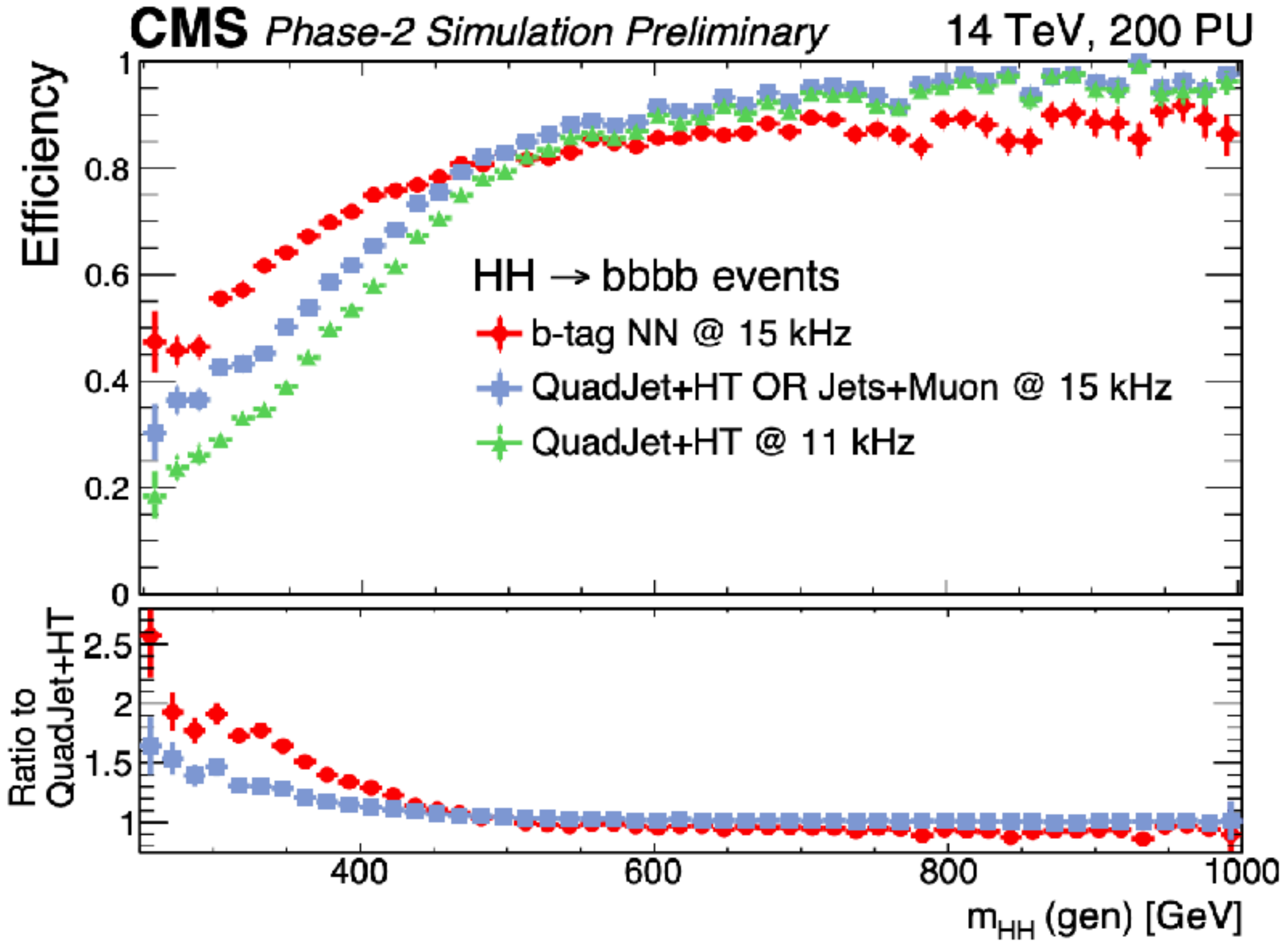


CMS TDR-021

L1 b-quark Identification

- NN trained to identify b-quarks using collection of particles
- Architecture includes featurizers that act on each particle individual
- **Significantly improved acceptance for HH→bbbb events with low m_{HH}** (compared to traditional cut-based methods)

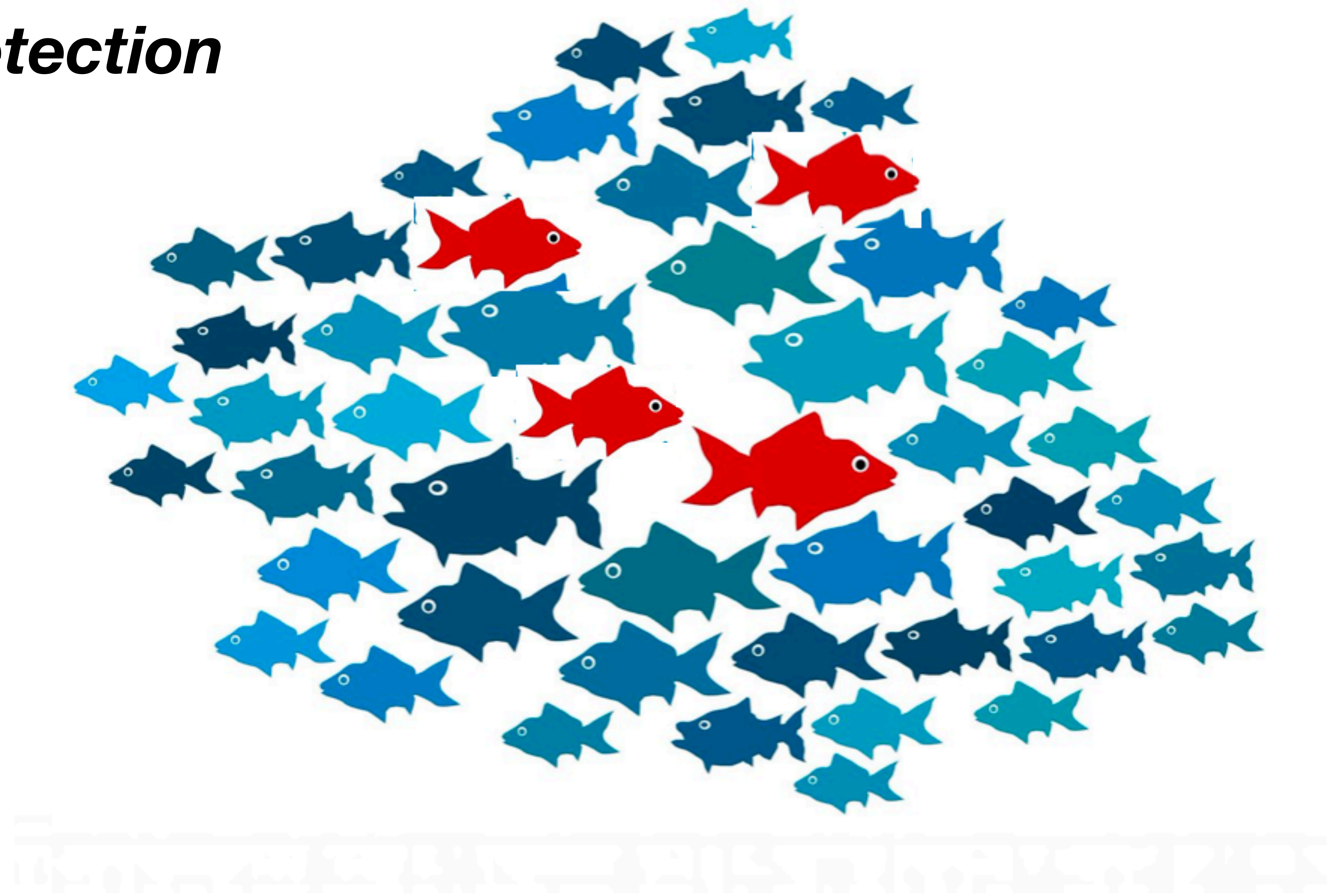




CMS DP-2022/021

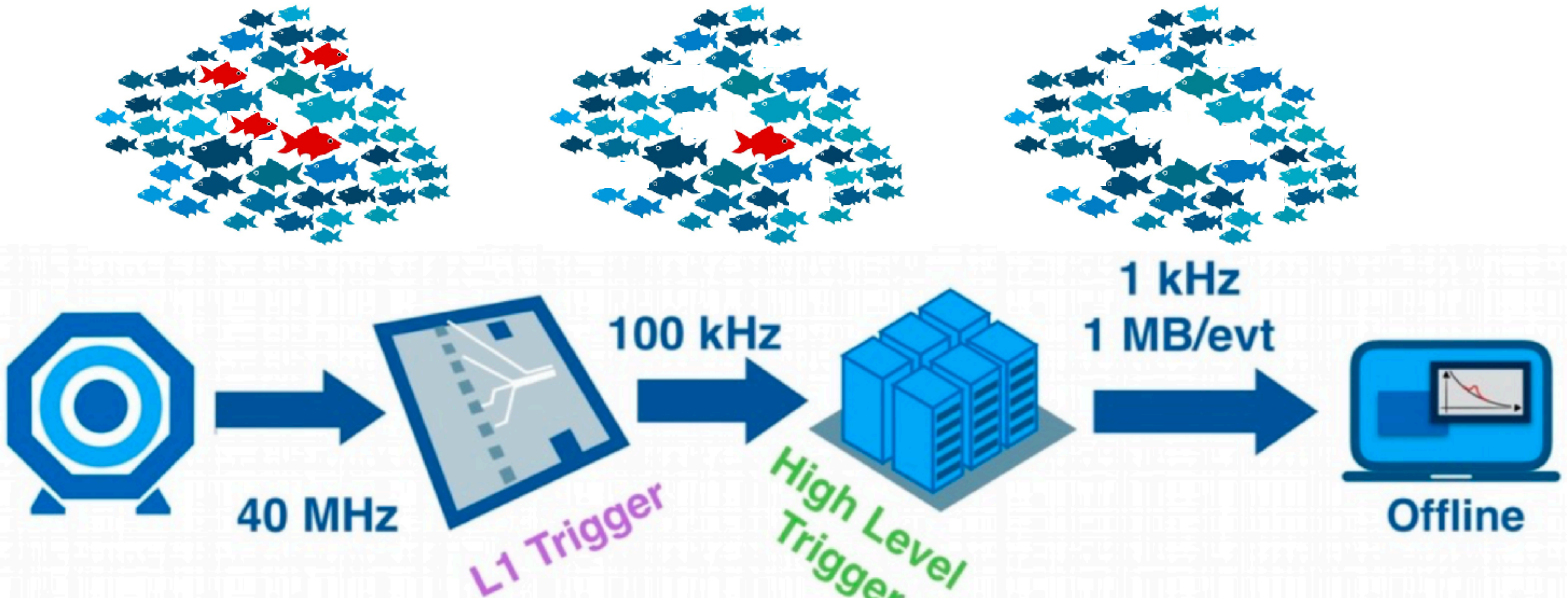
Anomaly Detection

- What if we don't know exactly what we are looking for?
- ML offers unique solution to this challenge (no traditional alternative)
  - Broad field of ***anomaly detection***



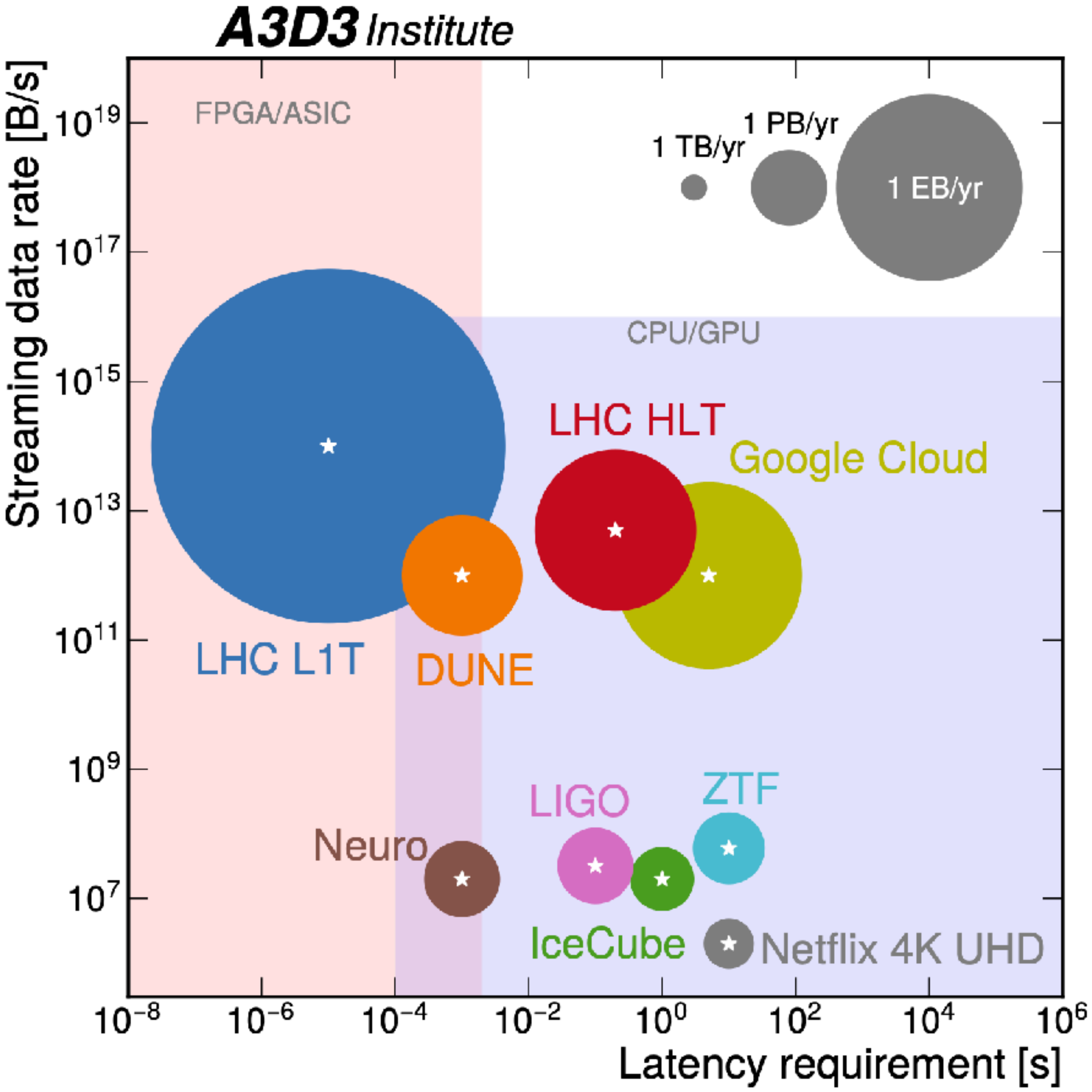
Fast Anomaly Detection?

- Depending on anomaly, we could have none left in recorded data
- Low-latency ML is the only option!



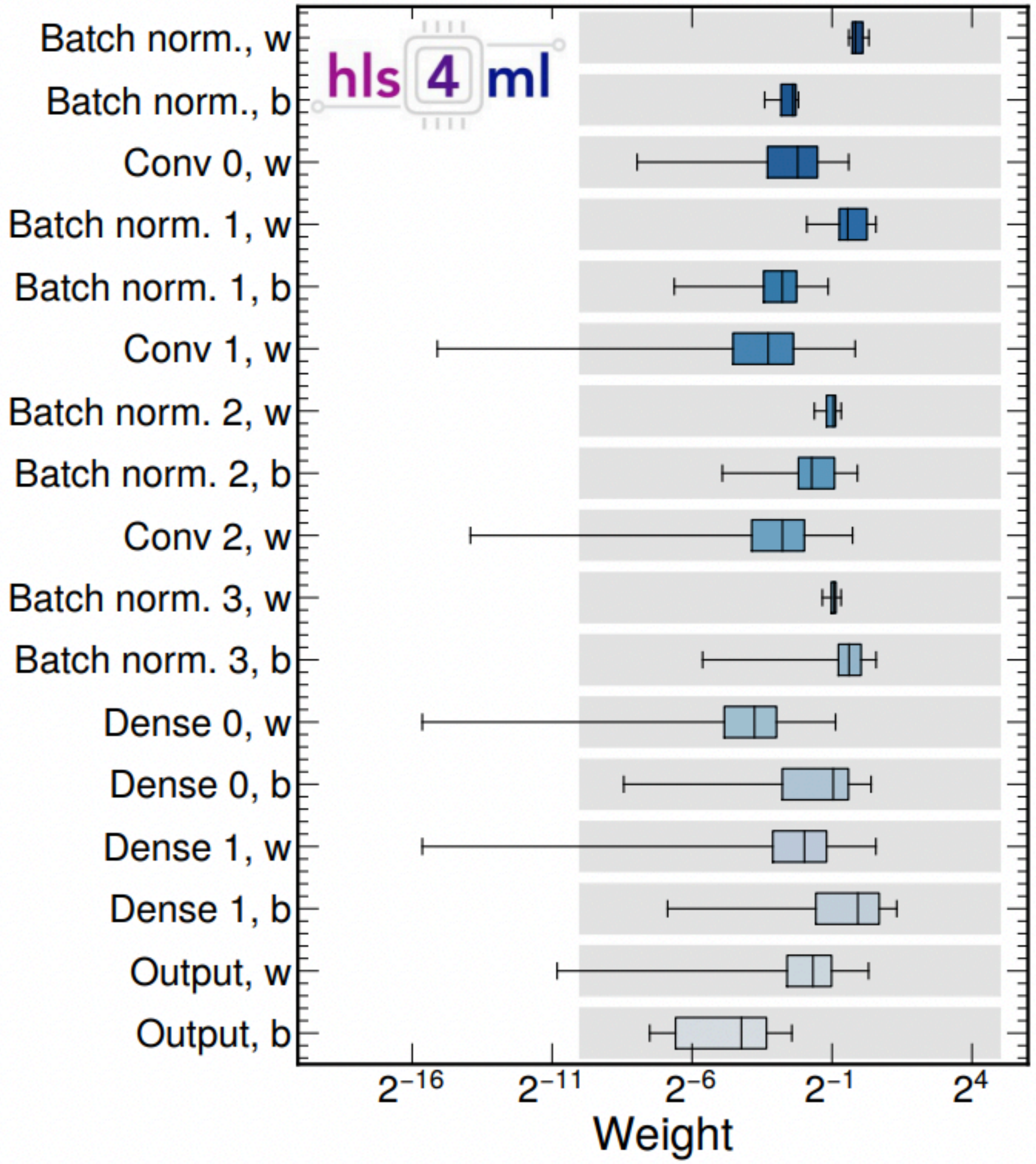
Conclusions

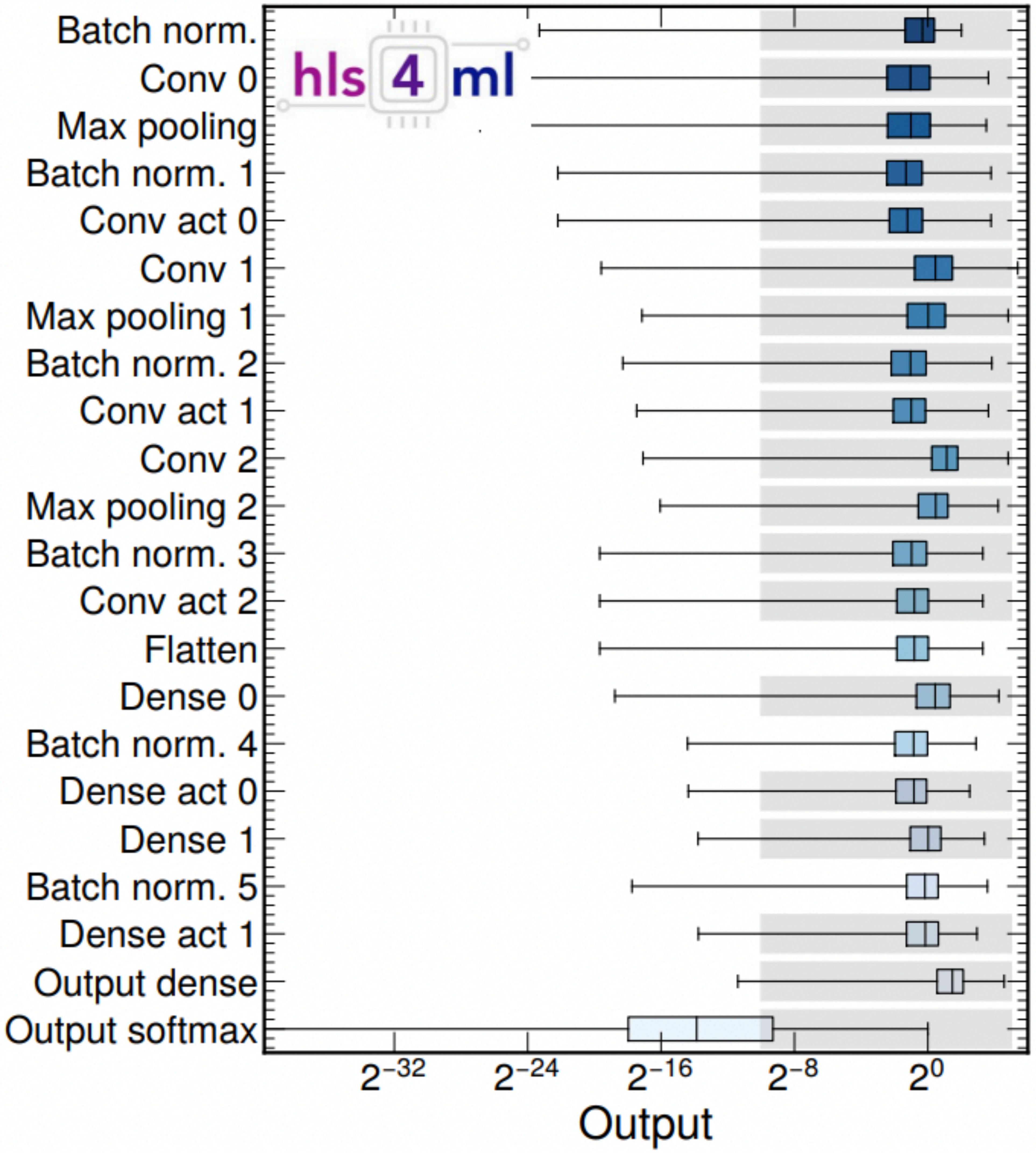
- Increasingly possible to perform low latency inference of ML models
  - Also low-power, high radiation
- ML offers improved performance over traditional algorithms
  - Potential for better alignment of offline and online algorithms
- Applications in many fields, areas



BACKUP

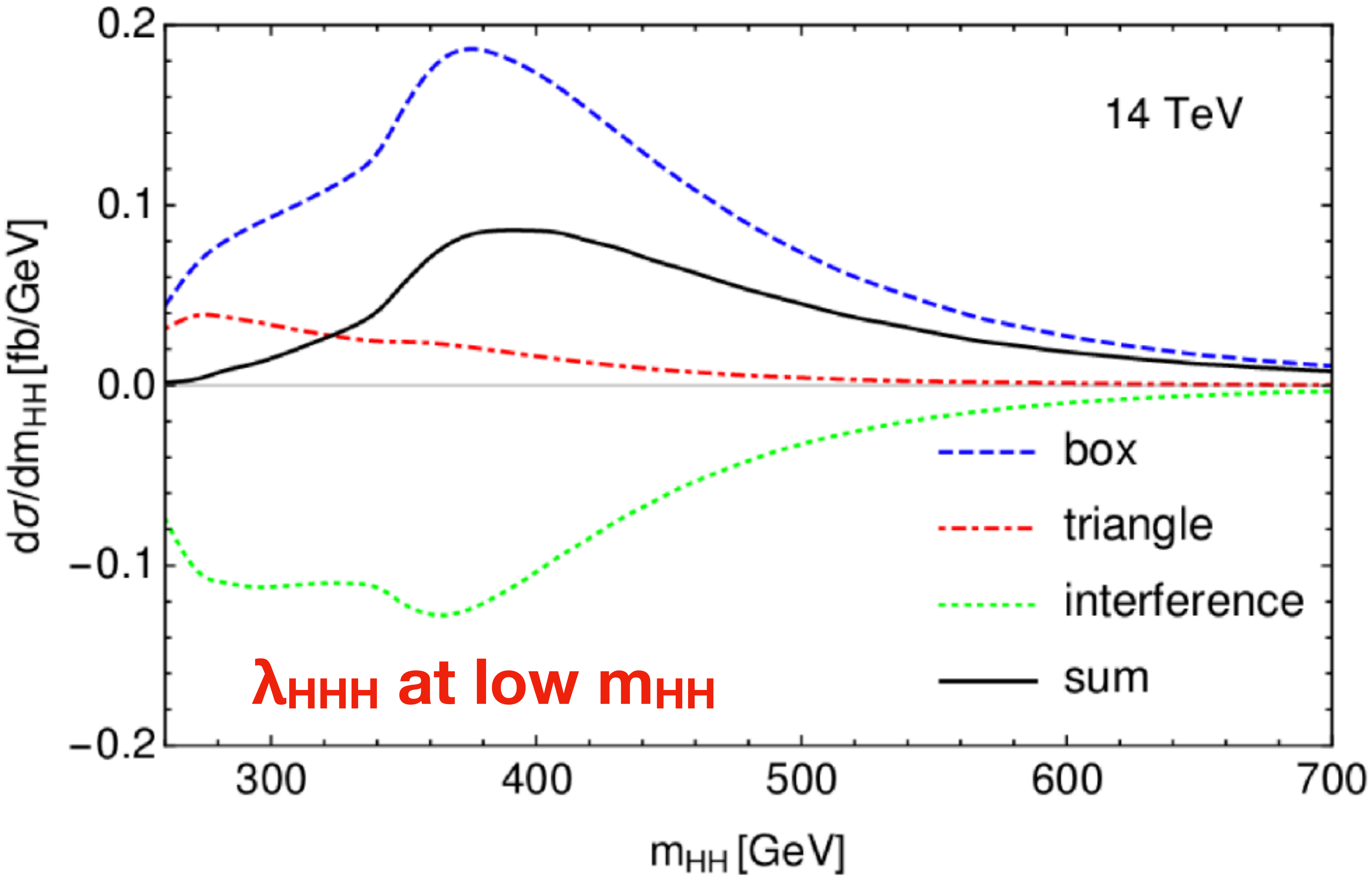
Precision

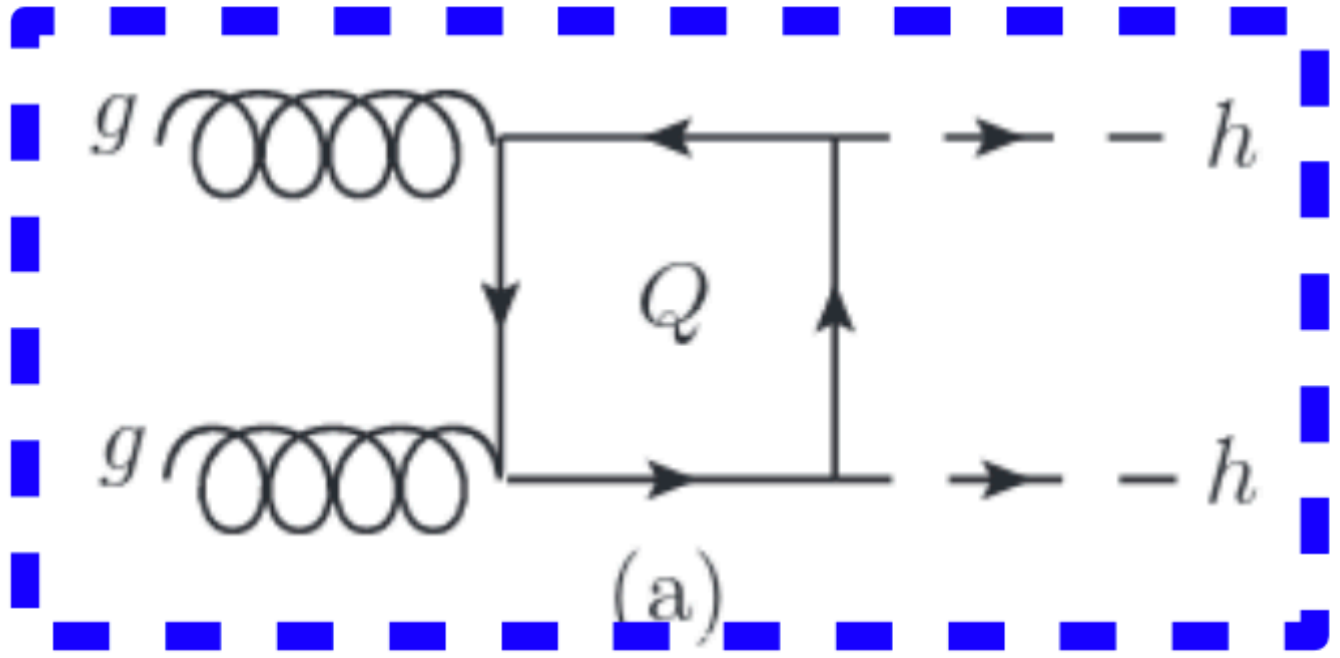

hls 4 ml
Batch norm., w
Batch norm., b
Conv 0, w
Batch norm. 1, w
Batch norm. 1, b
Conv 1, w
Batch norm. 2, w
Batch norm. 2, b
Conv 2, w
Batch norm. 3, w
Batch norm. 3, b
Dense 0, w
Dense 0, b
Dense 1, w
Dense 1, b
Output, w
Output, b
2^{-16}
2^{-11}
2^{-6}
2^{-1}
2^{4}
Weight

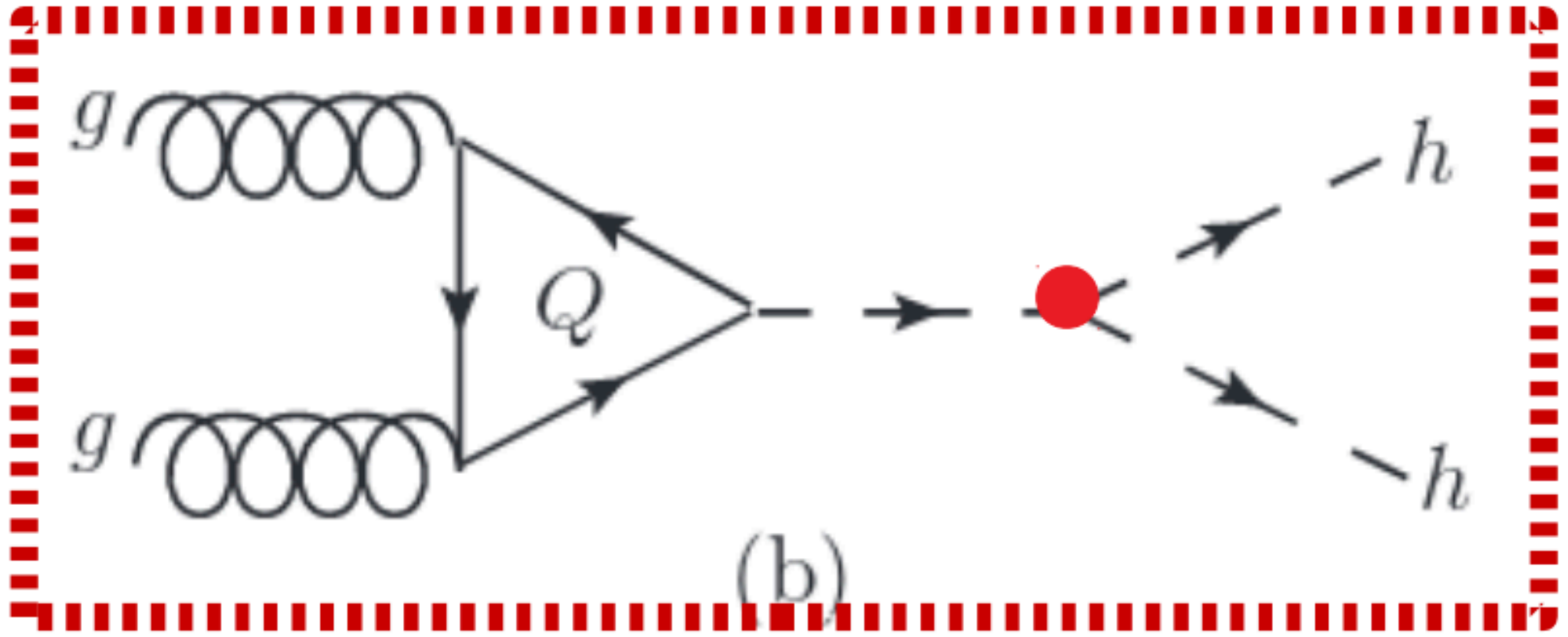

hls 4 ml
Batch norm.
Conv 0
Max pooling
Batch norm. 1
Conv act 0
Conv 1
Max pooling 1
Batch norm. 2
Conv act 1
Conv 2
Max pooling 2
Batch norm. 3
Conv act 2
Flatten
Dense 0
Batch norm. 4
Dense act 0
Dense 1
Batch norm. 5
Dense act 1
Output dense
Output softmax
2^{-32}
2^{-24}
2^{-16}
2^{-8}
2^{0}
Output

Di-Higgs Decays

- Lots of background that mimics these signals → very difficult to record
- Low m_{HH} is most critical, but produced object have lower energy





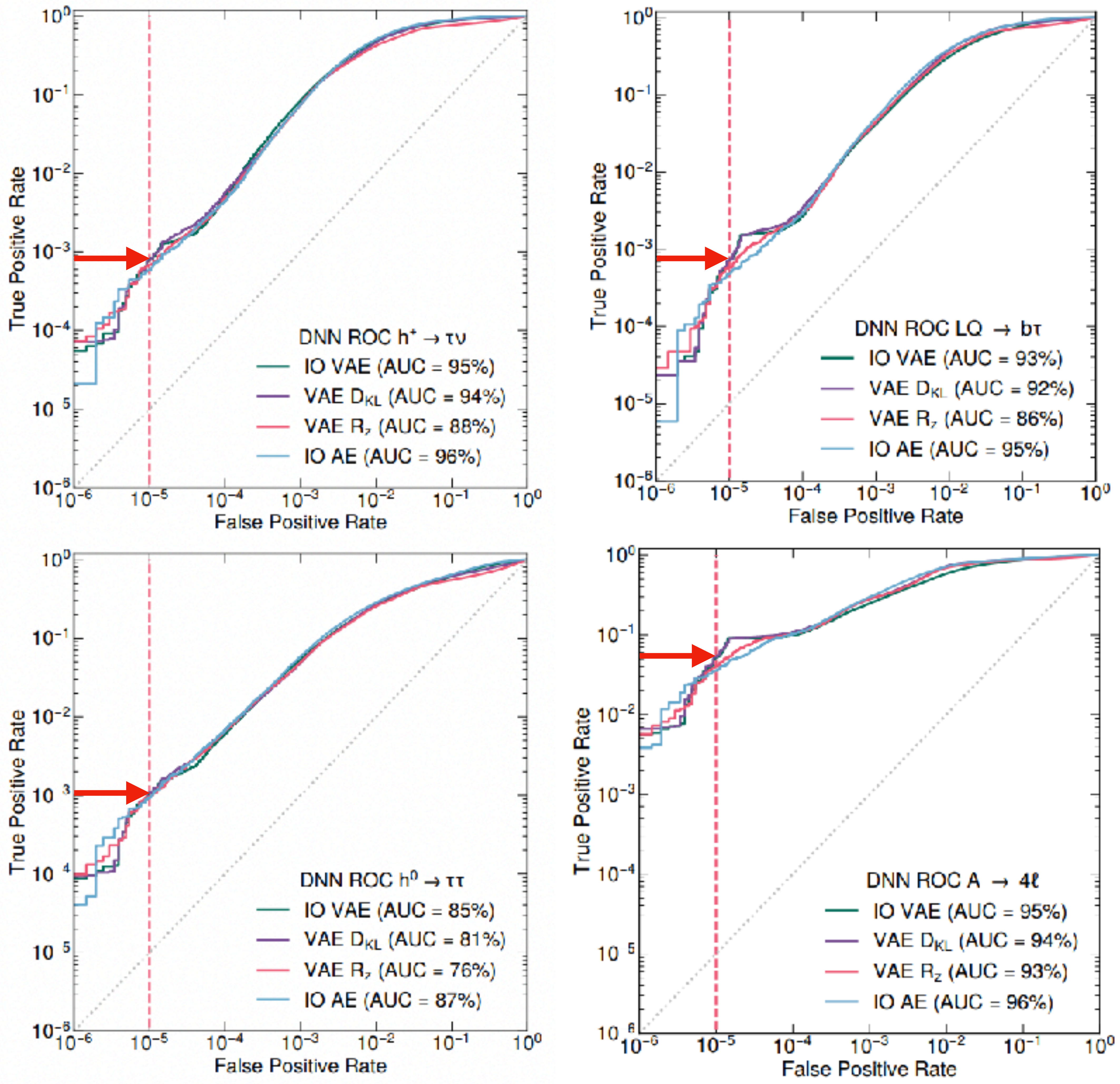


Fast Anomaly Detection

- Algorithm could take in relevant objects in each event

- Low latency is significant limitation on anomaly detection

- Performance depends on signals



Model	DSP [%]	LUT [%]	FF [%]	BRAM [%]	Latency [ns]	II [ns]
DNN AE QAT 8 bits	2	5	1	0.5	130	5
CNN AE QAT 4 bits	8	47	5	6	1480	895
DNN VAE PTQ 8 bits	1	3	0.5	0.3	80	5
CNN VAE PTQ 8 bits	10	12	4	2	365	115

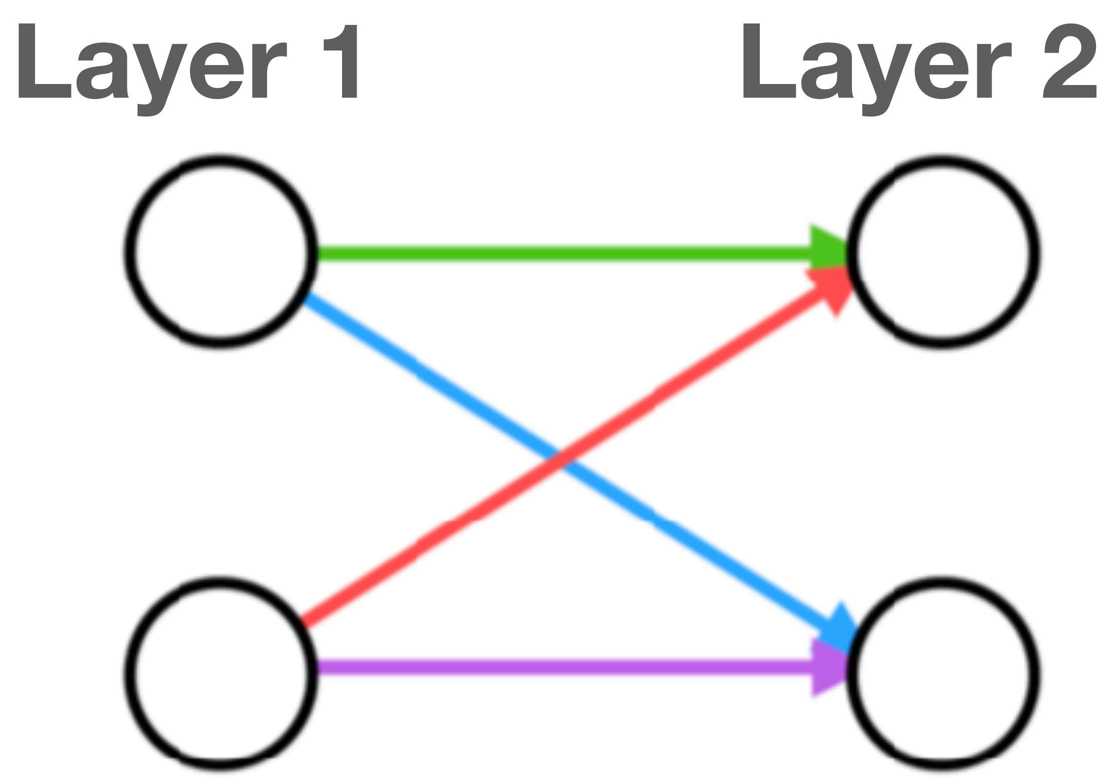
arXiv:2107.02157
arXiv:2108.03986

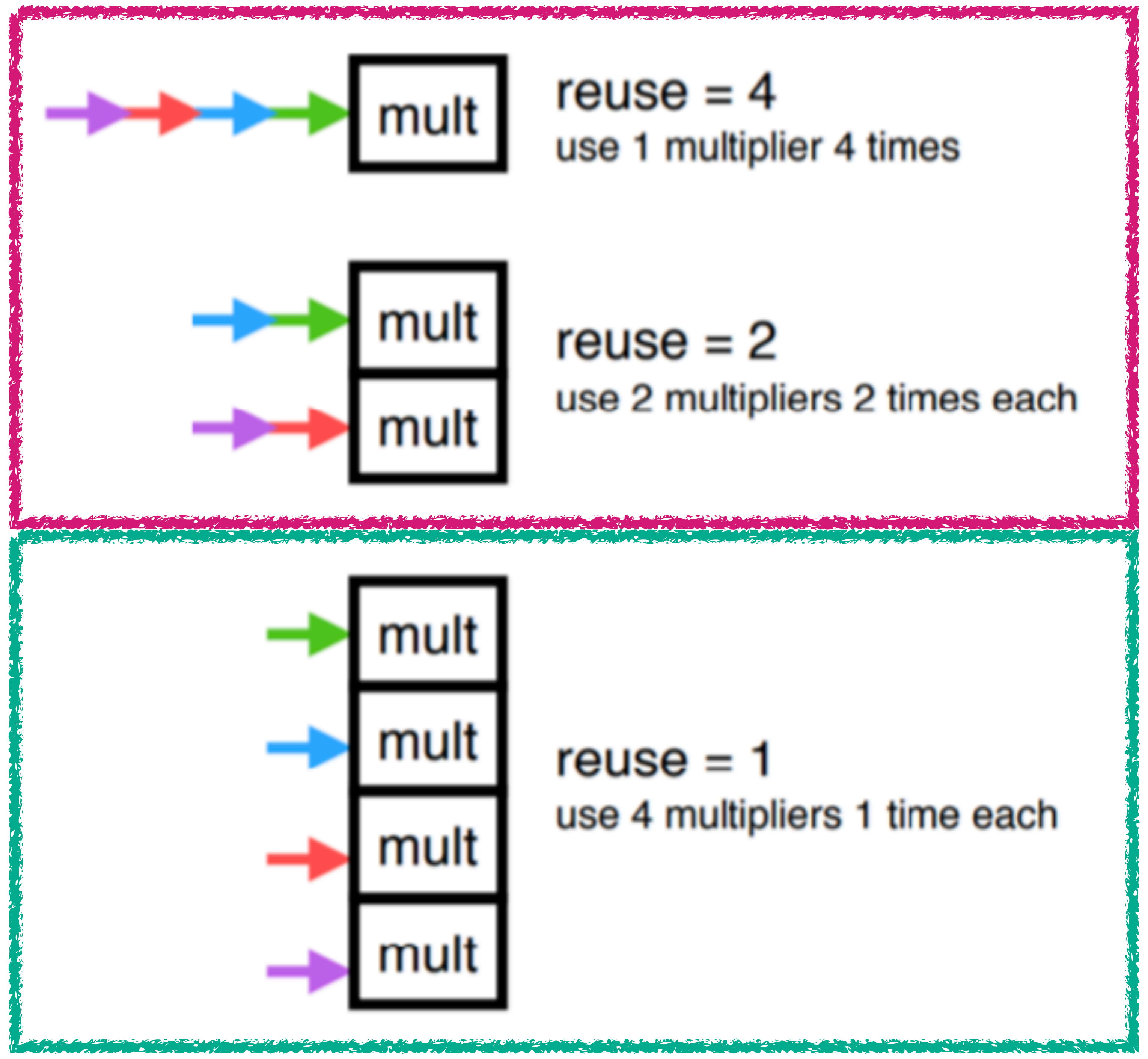
hls4ml Support

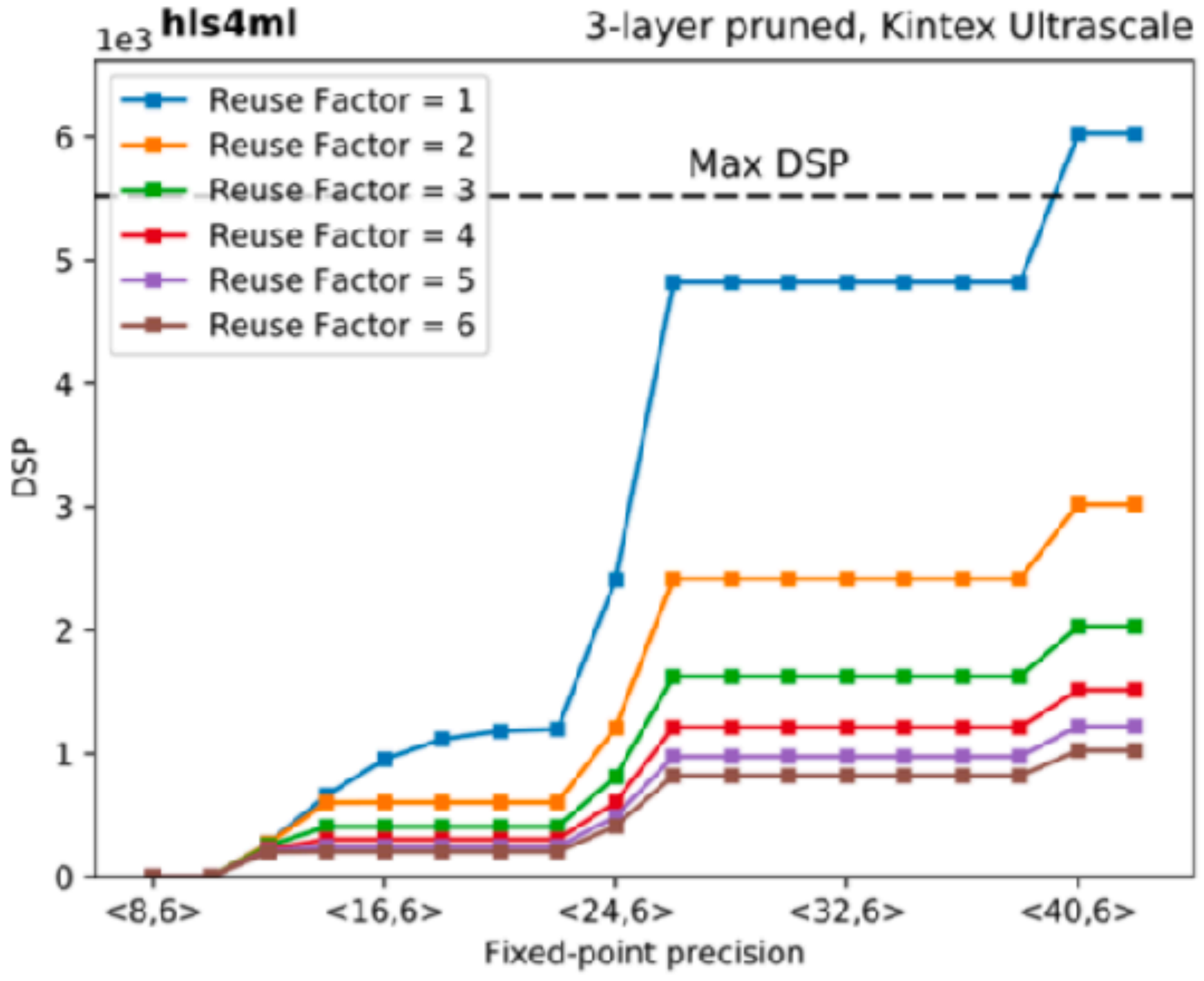
- Support for:
    - **MLPs, BDTs** [arXiv:2002.02534], **CNNs** [arXiv:2101.05108], **Binary & Ternary NNs** [arXiv:2003.06308], **Quantization-aware training (QKeras)** [arXiv:2006.10159], **Modified GarNet architecture (GraphNN)** [arXiv:2008.0360], **RNNs/LSTMs/GRUs** [arXiv:2207.00559]
- Active maintenance and development
    - Many applications for fast ML in physics (low latency, low power)

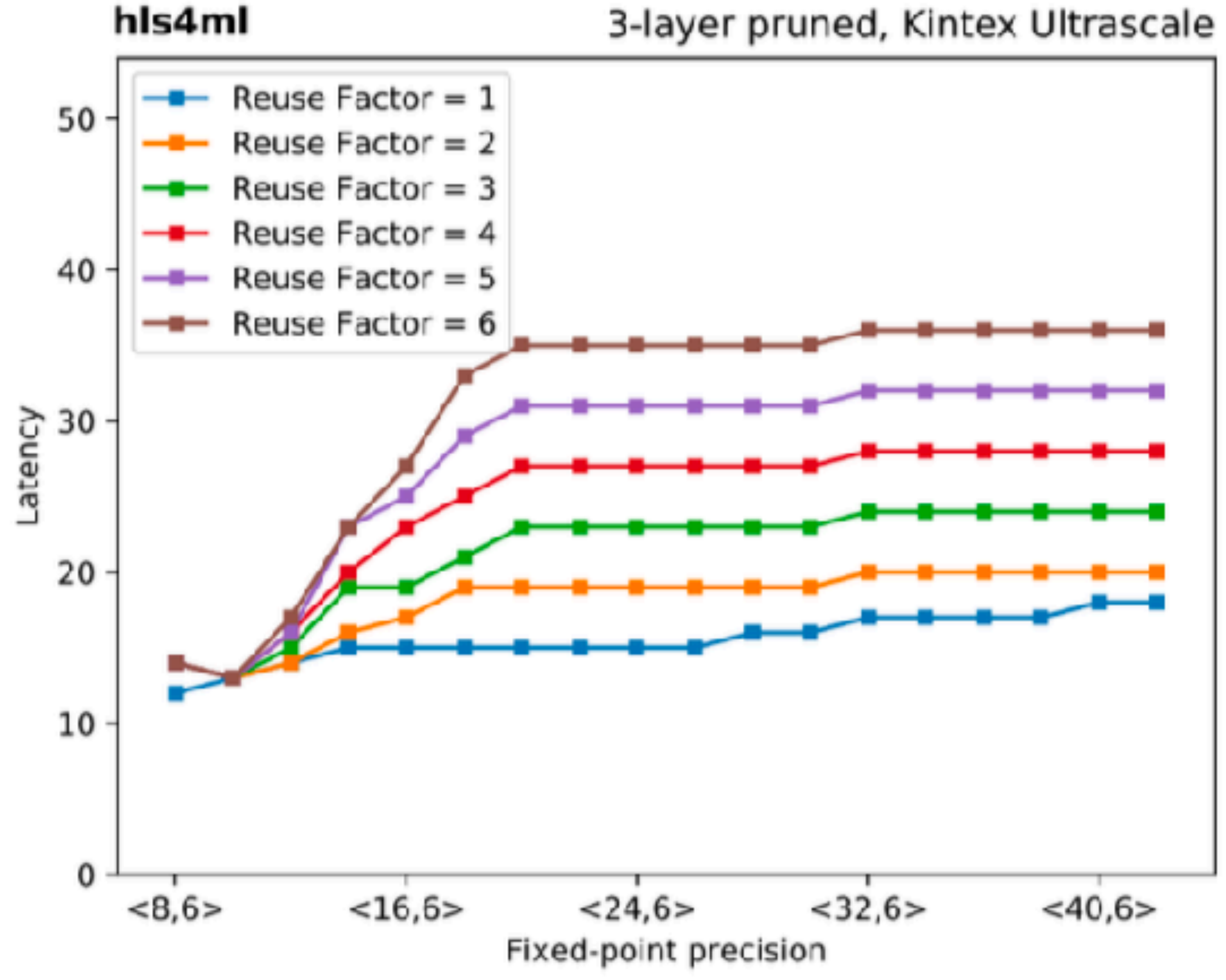
Reuse

- For lowest latency, compute all multiplications at once
  - Reuse = 1 (fully parallel) → latency = # layers)
- Larger reuse implies more serialization
- Allows trading higher latency for lower resource usage

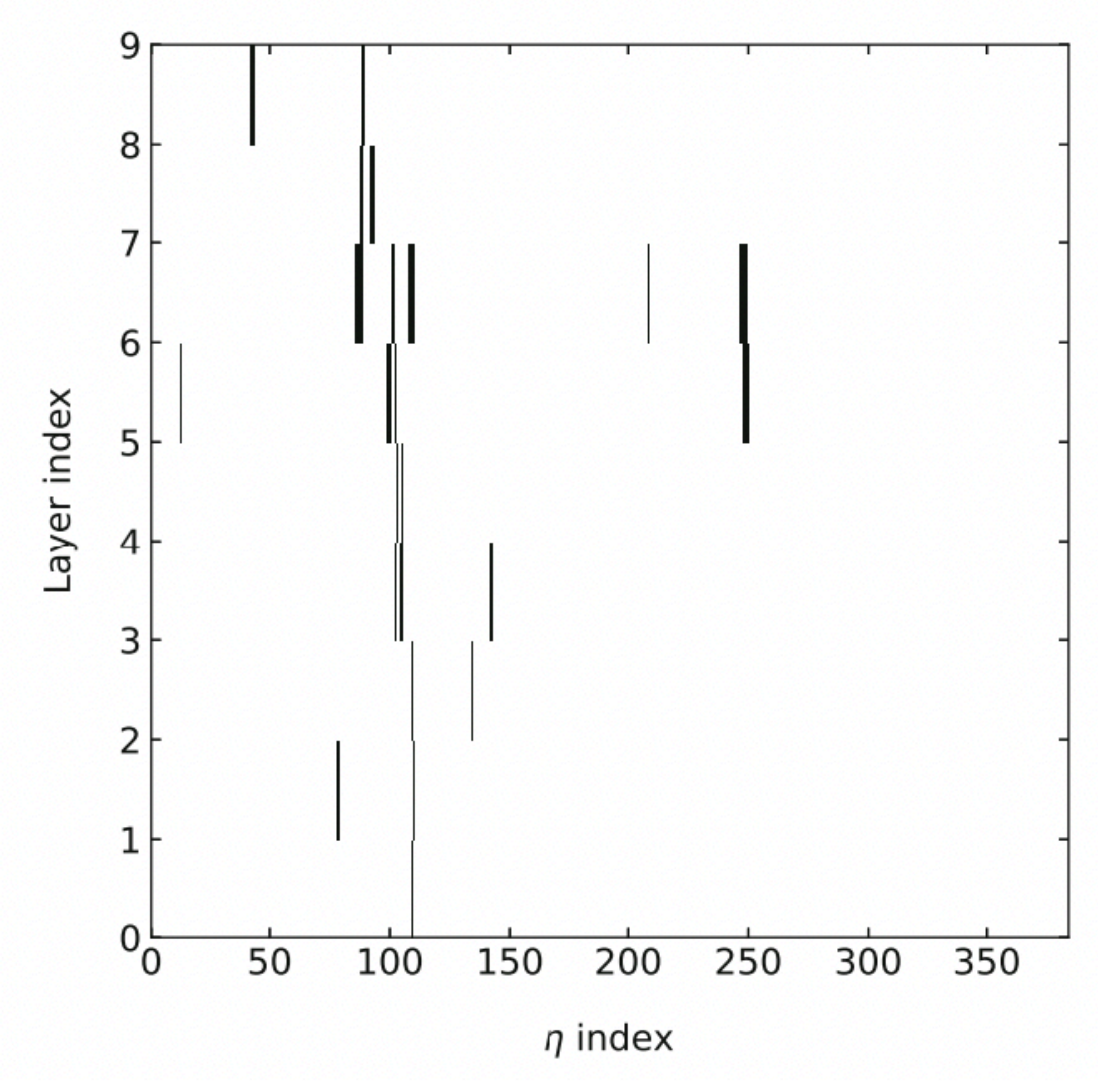




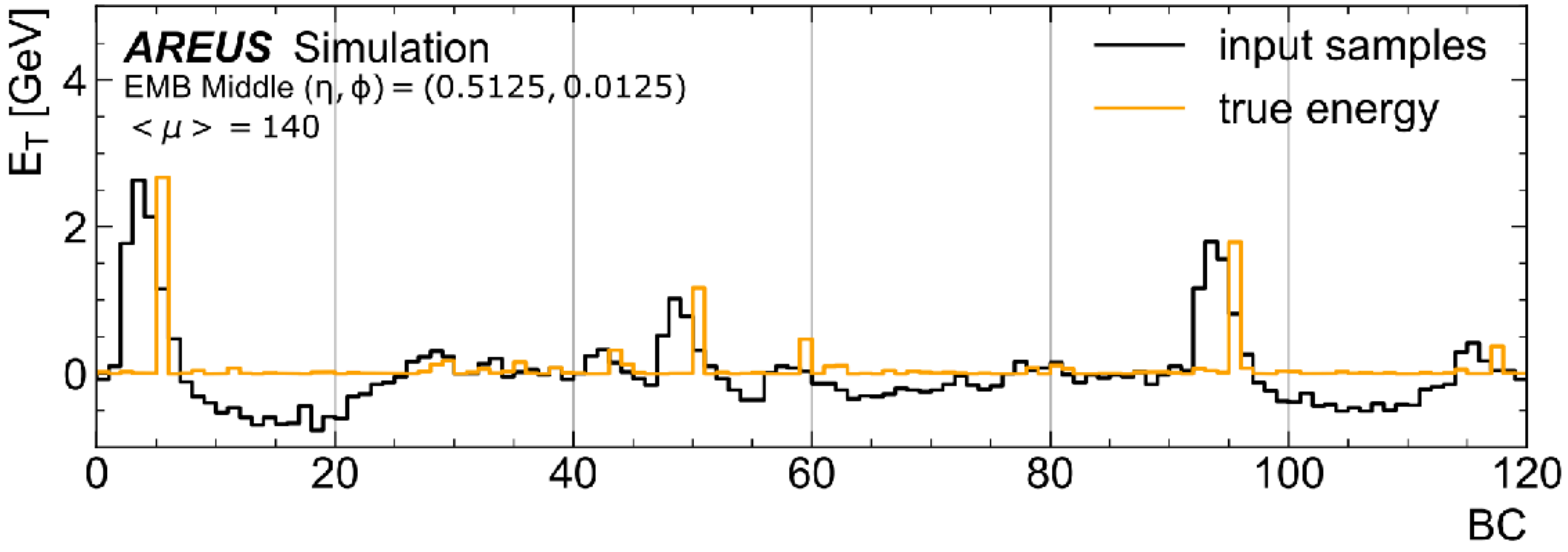




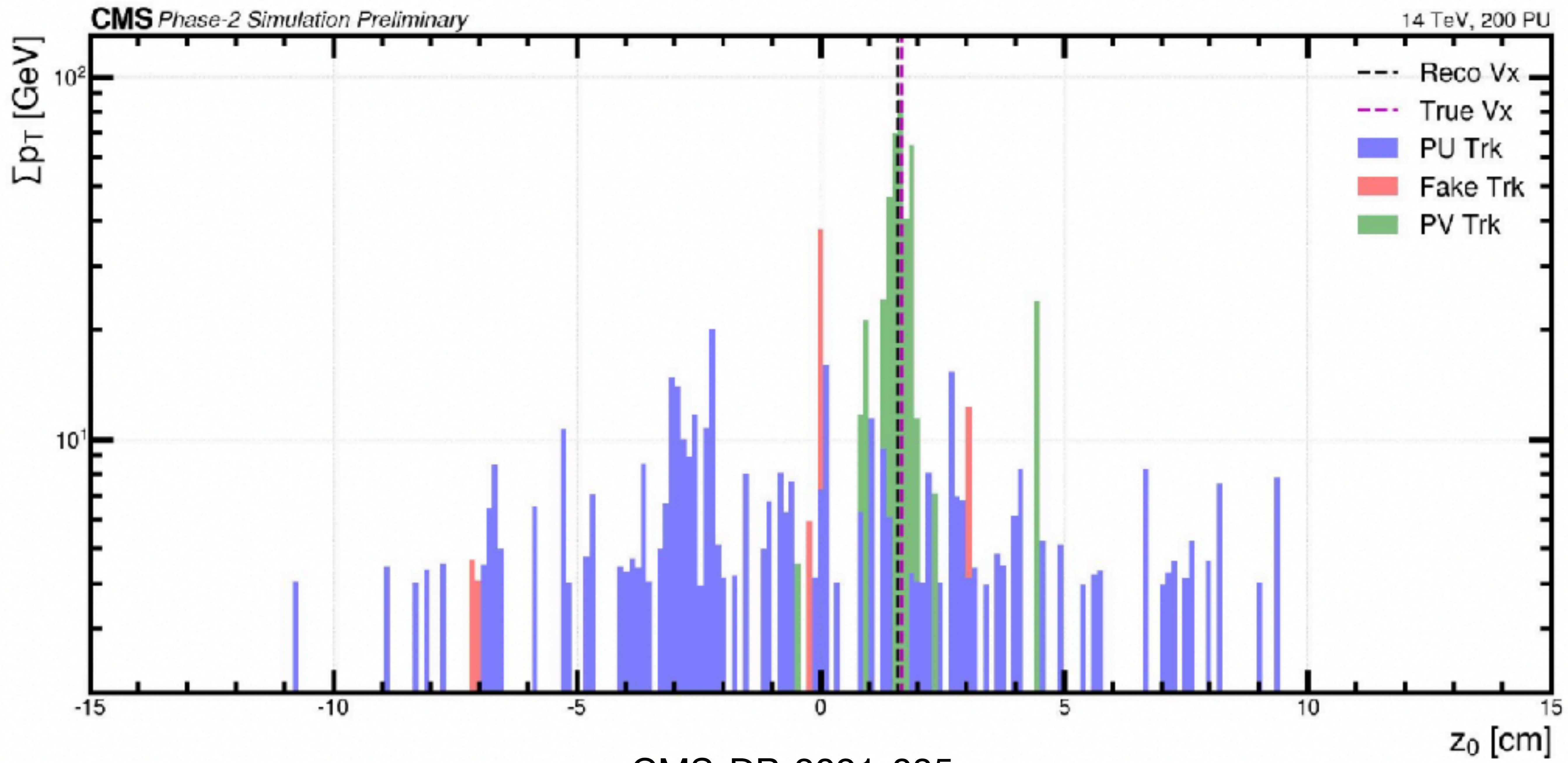
Applications



Eur. Phys. J. C (2021) 81 :969



arXiv: 2111.08590



CMS-DP-2021-035