Fast Machine Learning Inference for Scientific Discovery

Elham E Khoda

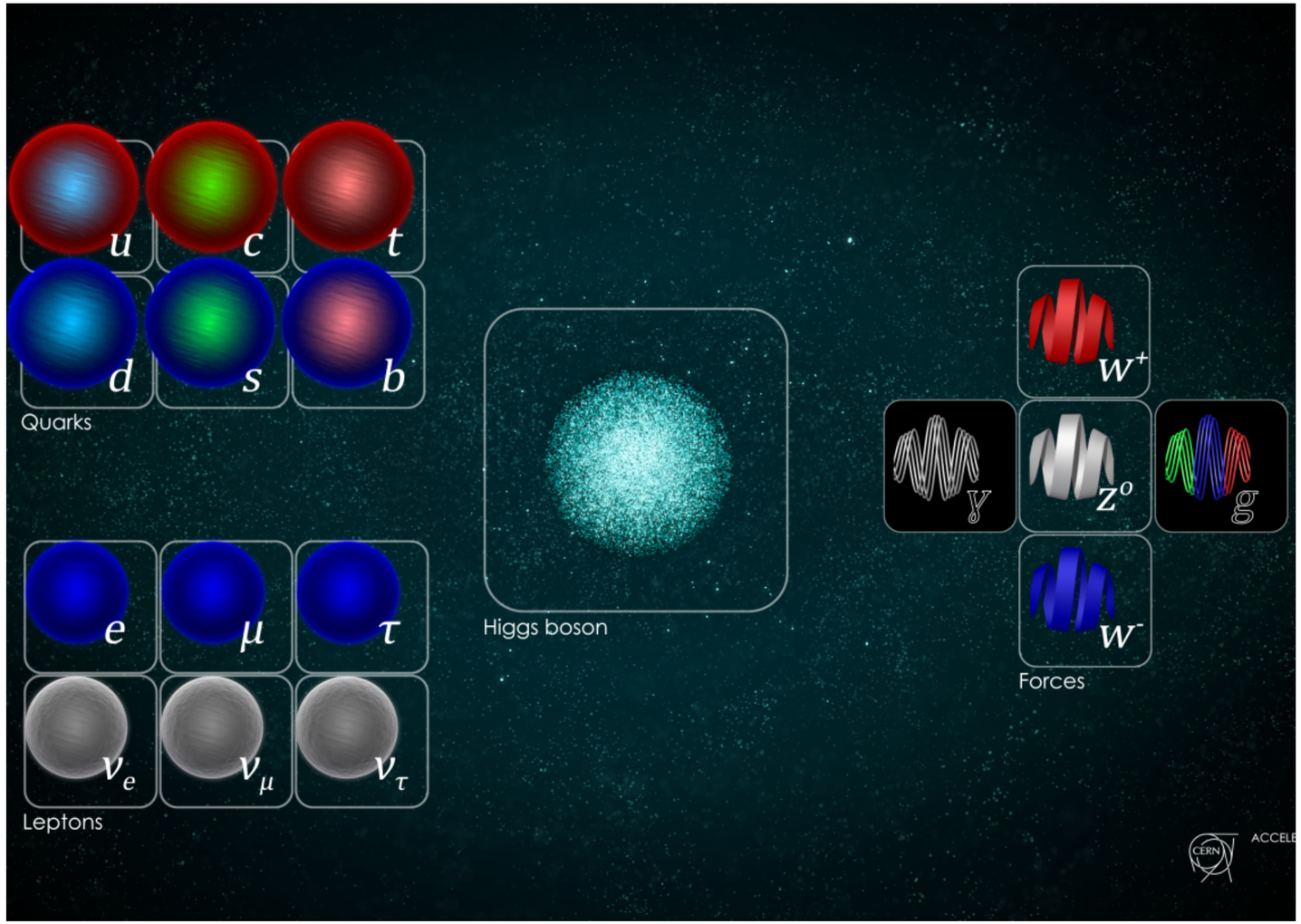
Particle Physics Seminar, University of Hawaii, Manoa

March 28, 2024

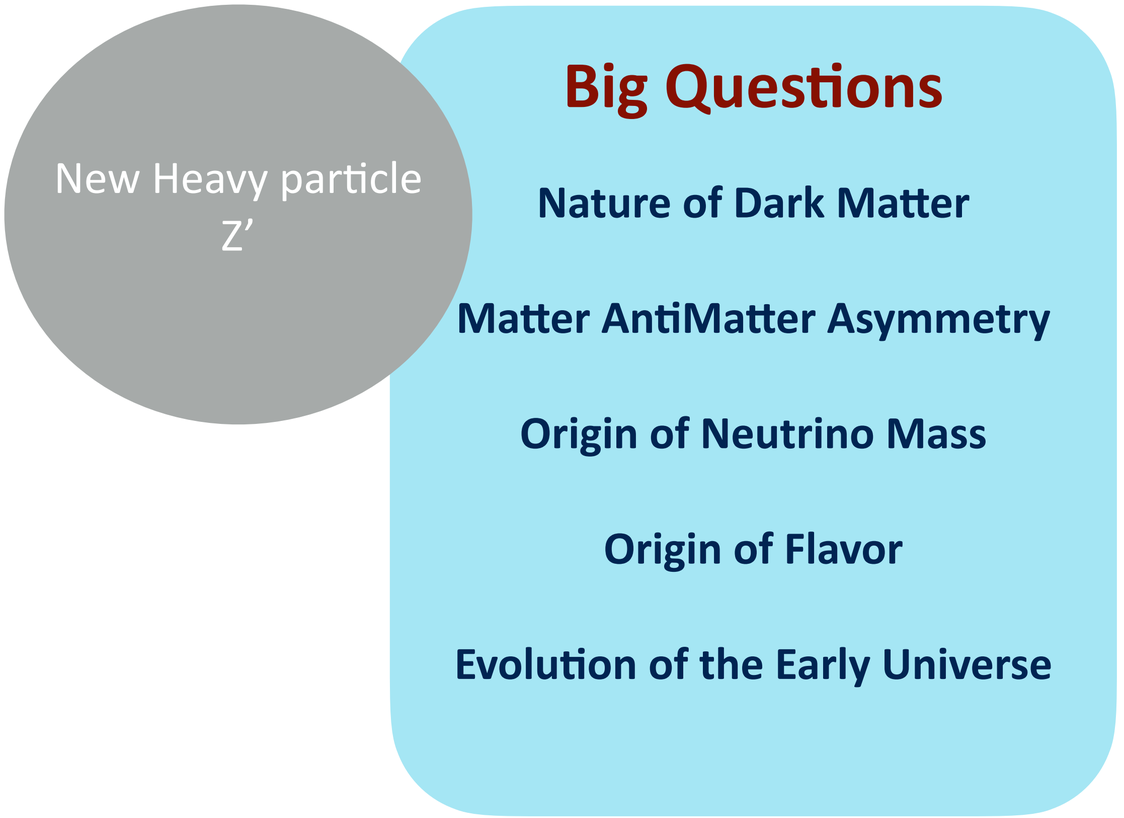
ATLAS EXPERIMENT

a3d3 Accelerated AI Algorithms for Data-Driven Discovery

Standard Model of Particle Physics

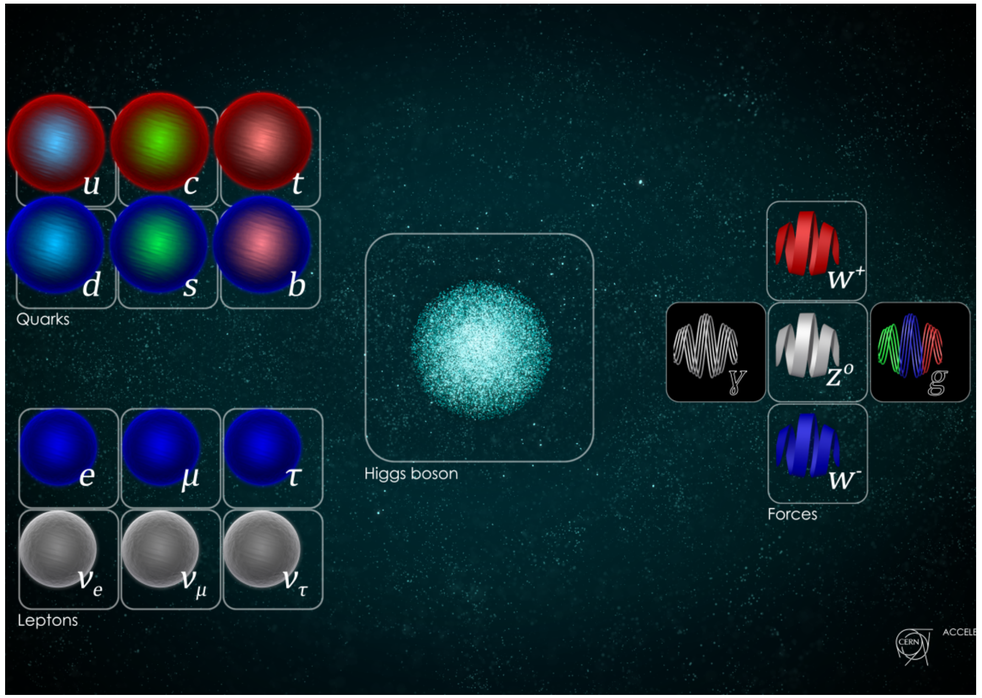


Physics Beyond the Standard model



Mediator to the Dark Sector

Standard Model



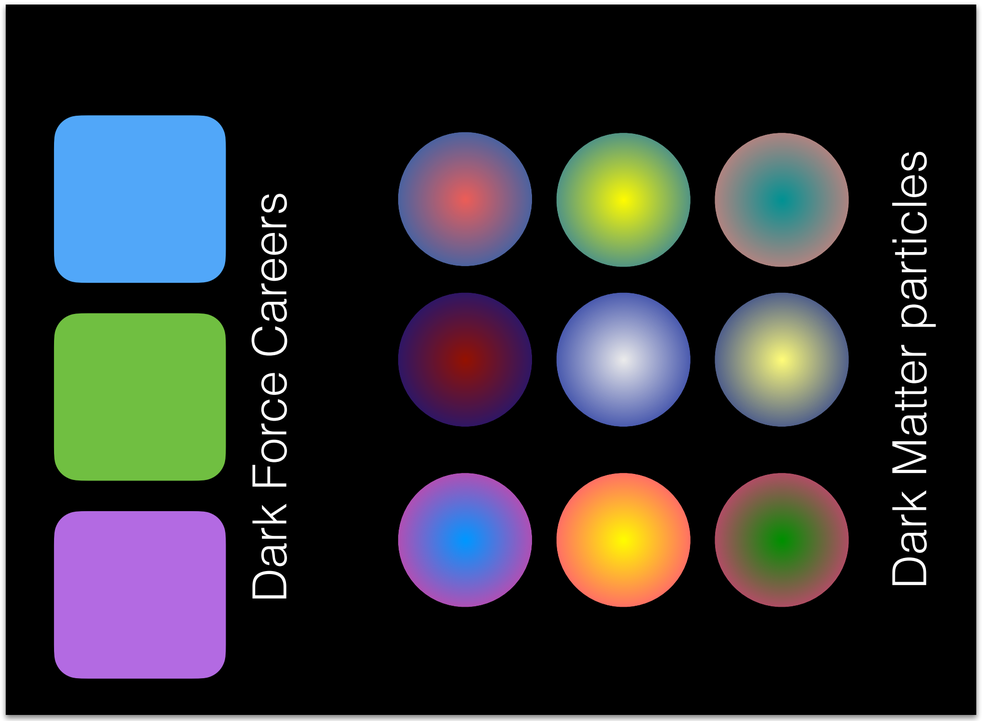
Mediators

?

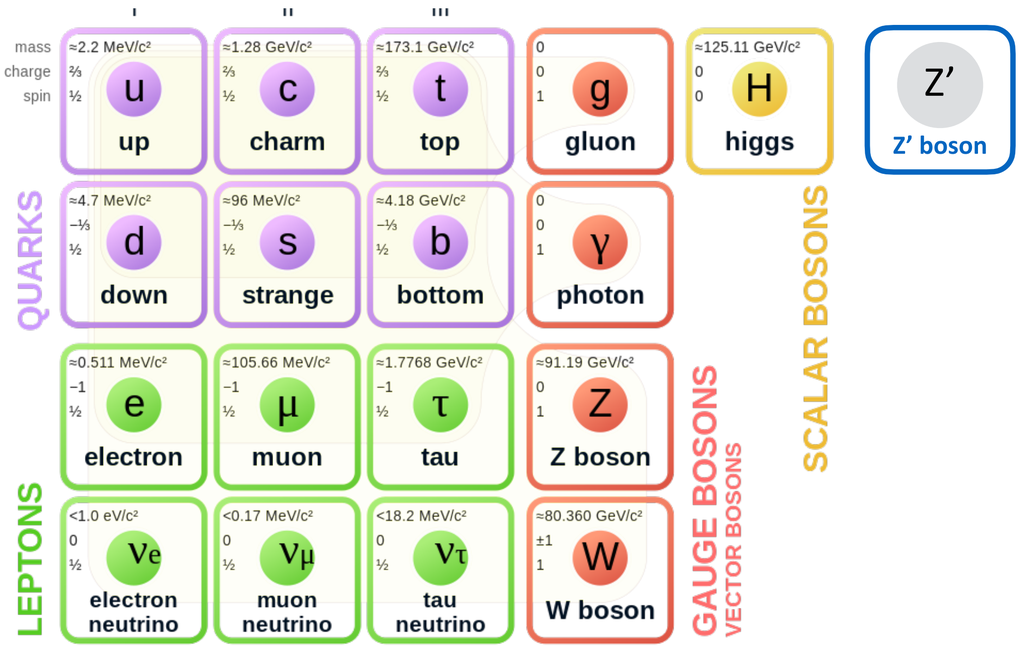
Z'

Z' boson

Dark Sector



Extension to the SM



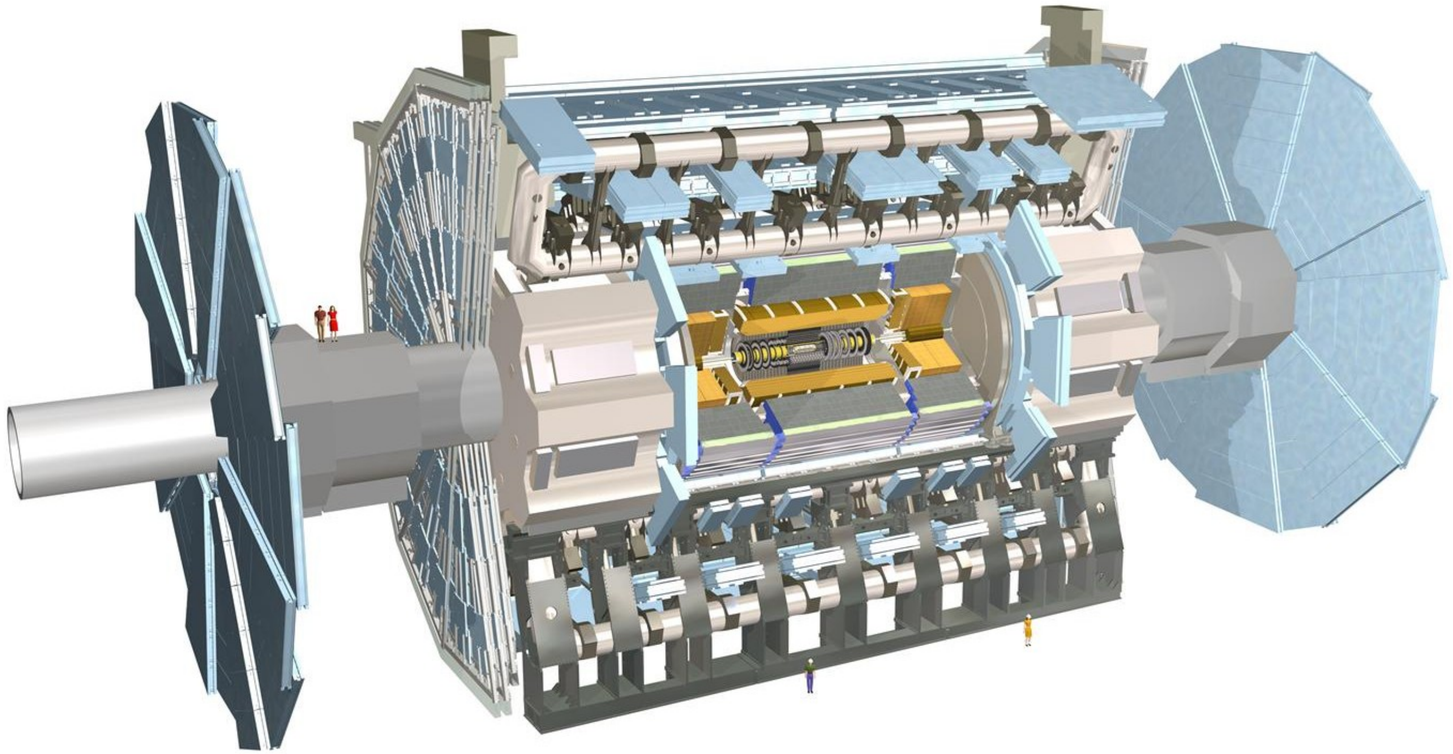
How to search for such Mediators in Collider Experiments?

Large Hadron Collider (LHC) and the ATLAS experiment

- World's largest and most powerful particle collider
- Collides protons bunches (~10^{11} protons) spaced by 25 ns

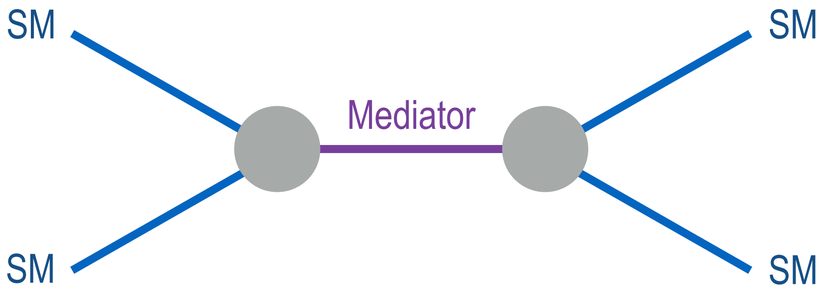
The ATLAS Experiment

ATLAS is a general purpose detector

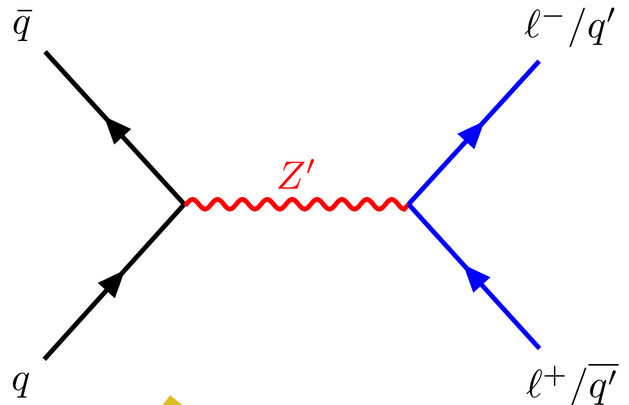


How to search for such Mediators in the ATLAS Experiment?

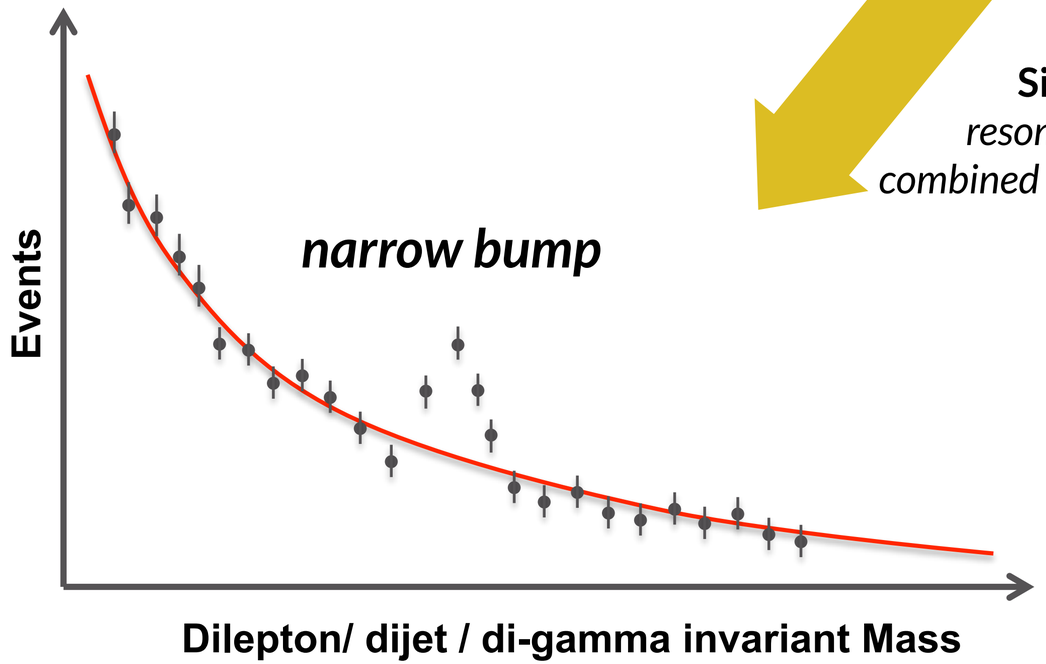
Dijet Final State



Example
2-lepton / 2-quark final state



Signature
resonance on the combined mass distribution



Theory model-driven Discovery Regime

Model-dependent Search

Most sensitive approach for a particular new physics scenario

→ Unlikely to probe a different scenario

Big Questions

Nature of Dark Matter

Matter AntiMatter Asymmetry

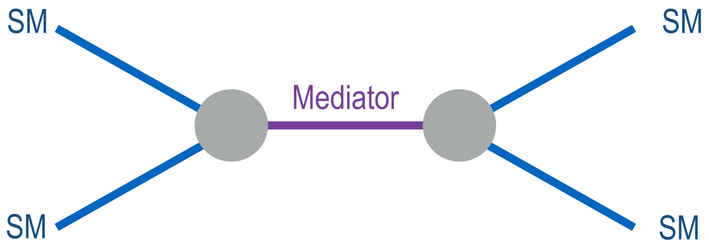
Origin of Neutrino Mass

Origin of Flavor

Evolution of the Early Universe

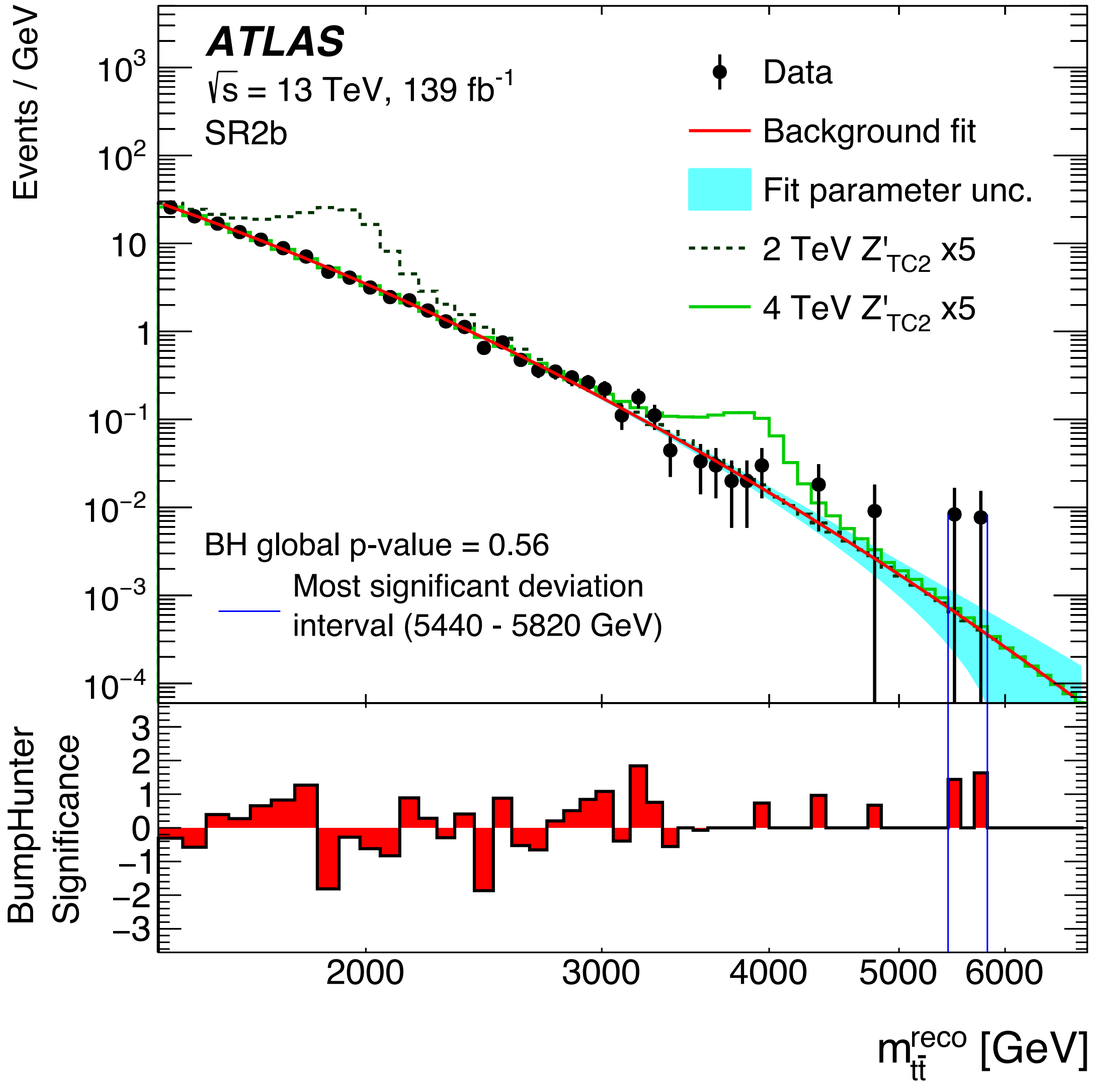
New Heavy particle
Z'

Physics Scenario
Signal model



Maximixe Sensitivity

Example: 2-jet final state

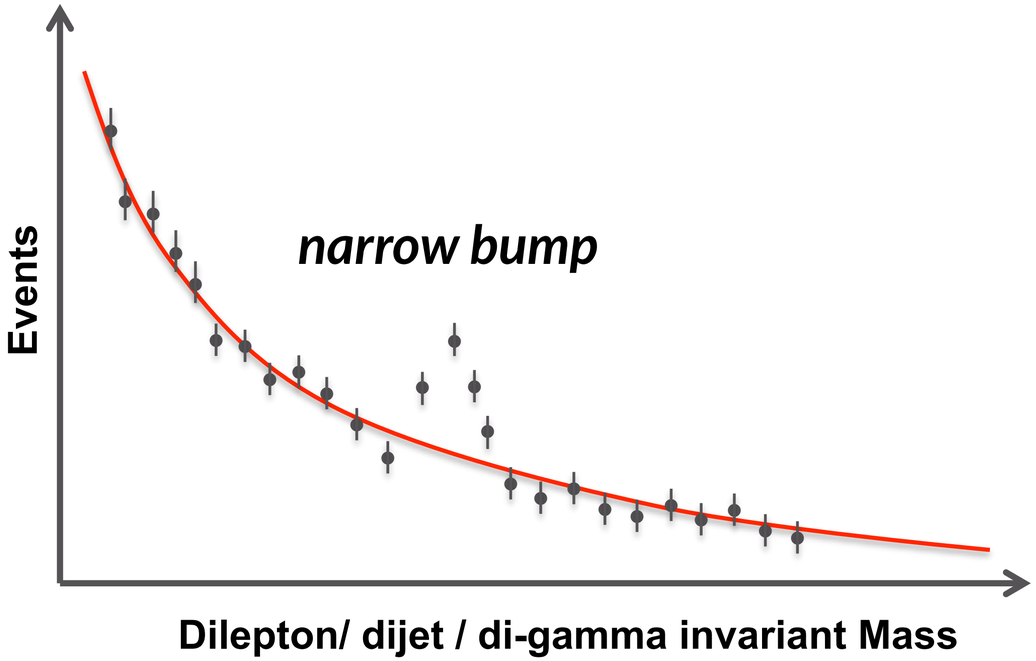


Data-driven Discovery Regime

Anomaly detection

Identify data with features that appear inconsistent with those of the majority of the dataset

For HEP application: Interested in an ensemble of events rather than a single outlier



Resonant Anomaly Detection Search

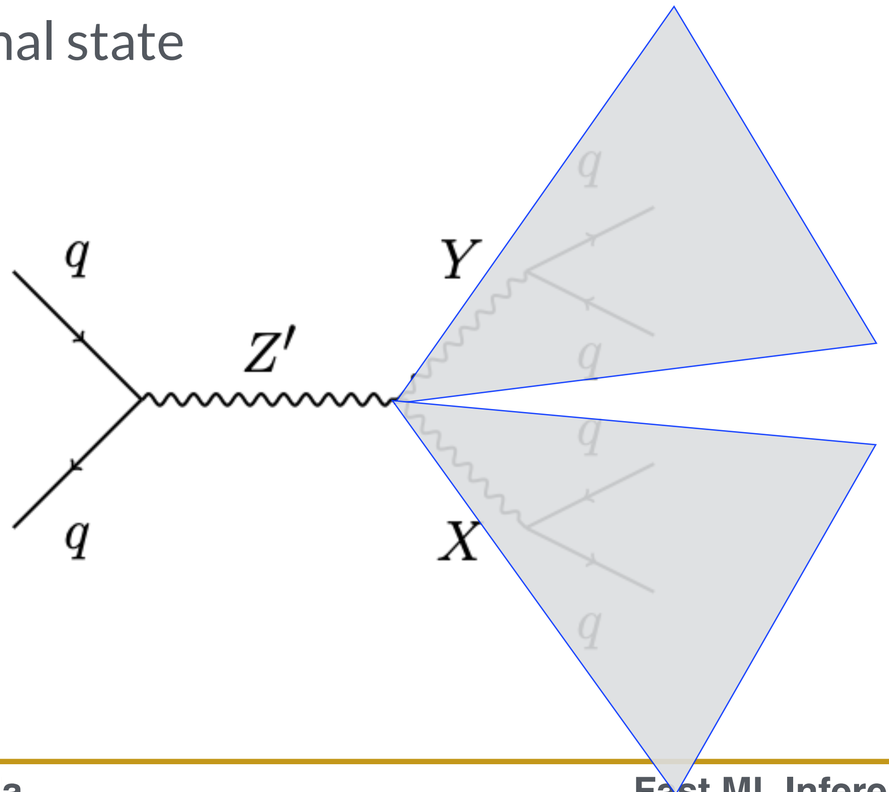
Assumption:

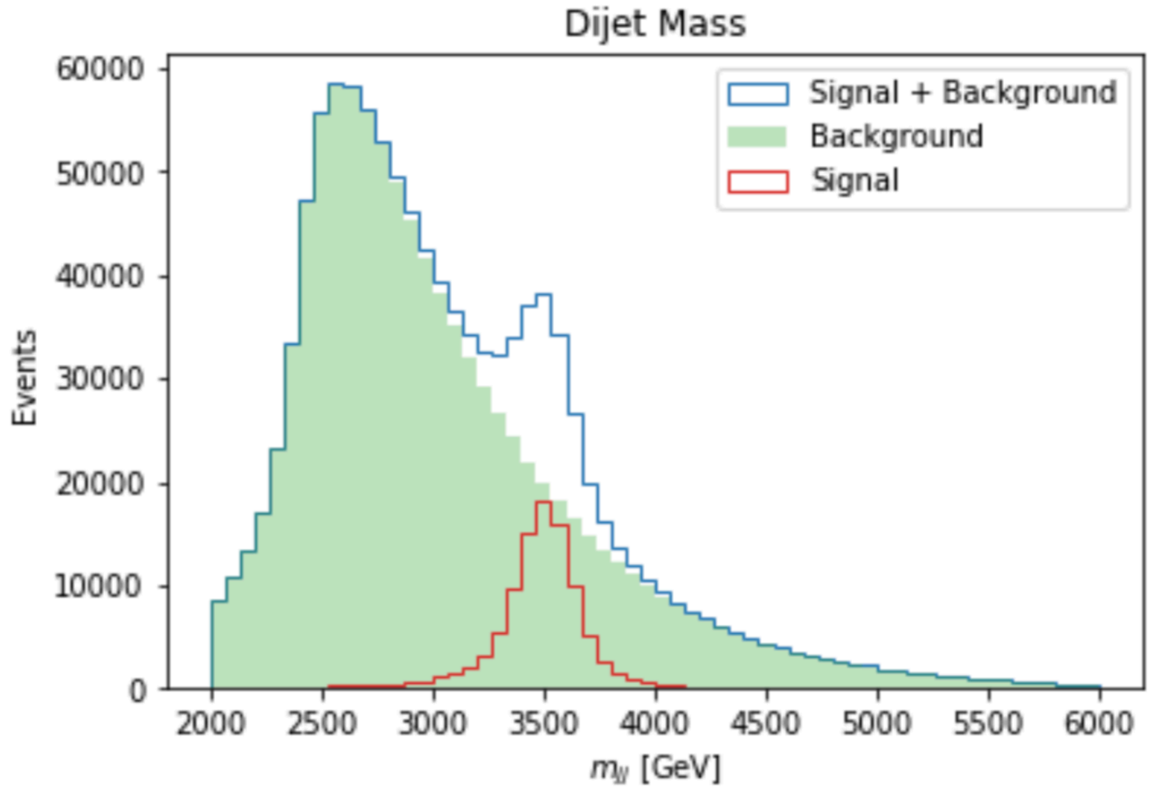
Signal is localized at least in one of the feature spaces (x)

$p_{\text{signal}}(x)/p_{\text{background}}(x)$ is high

Example:

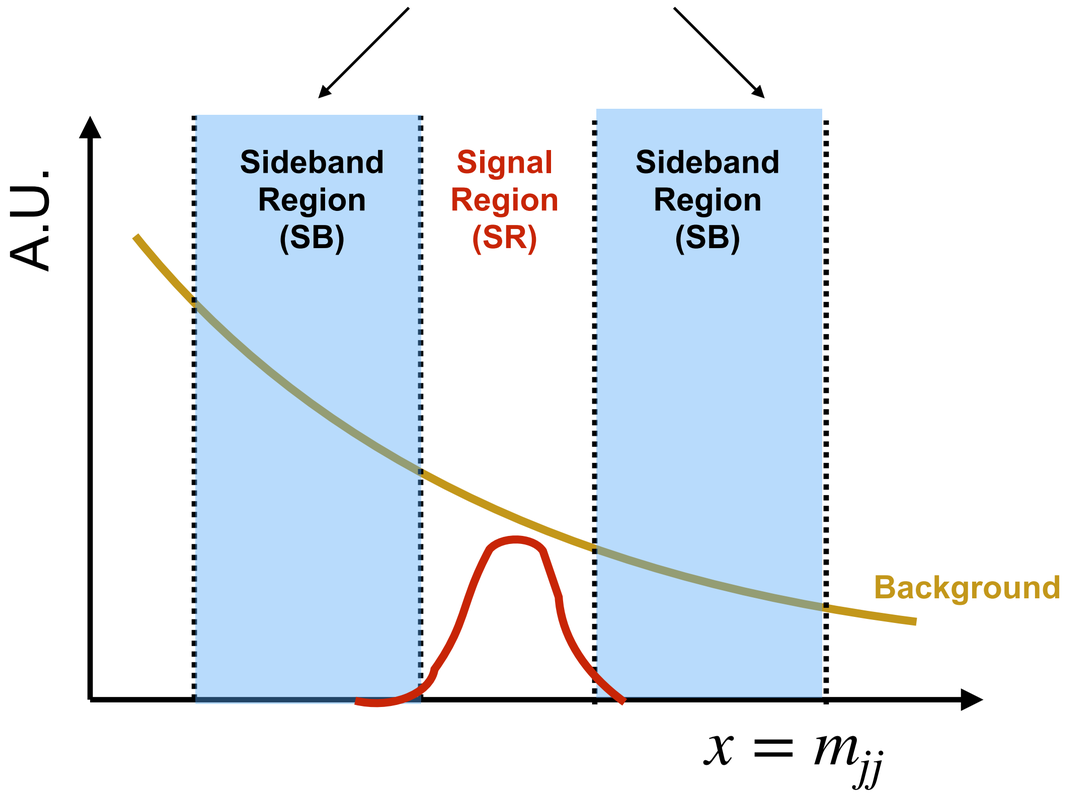
2 jet final state



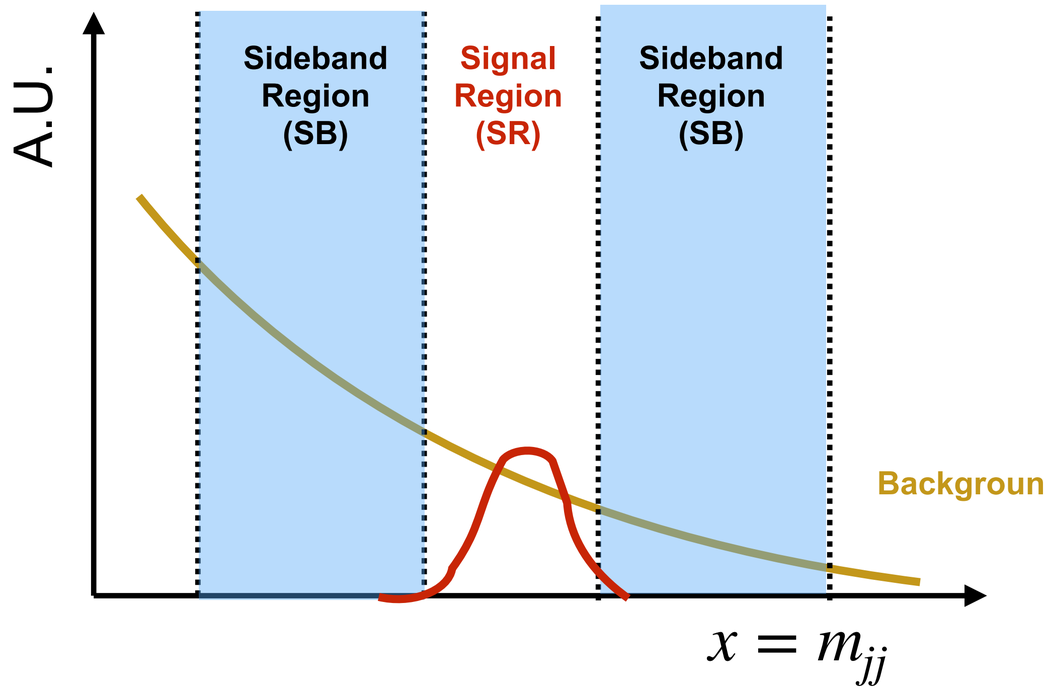


Increase Signal Sensitivity

Estimate background
with Generative algorithms

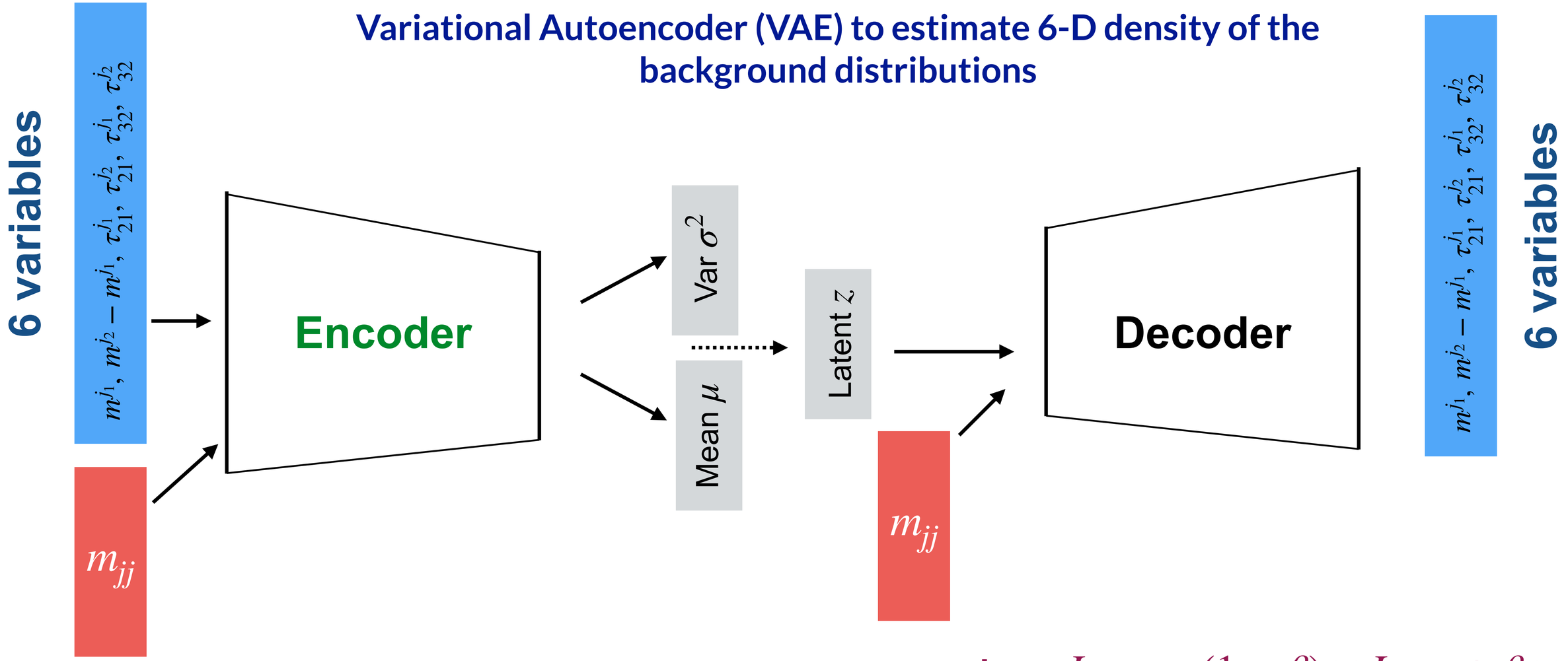


Reduce background keeping as
much signal as possible



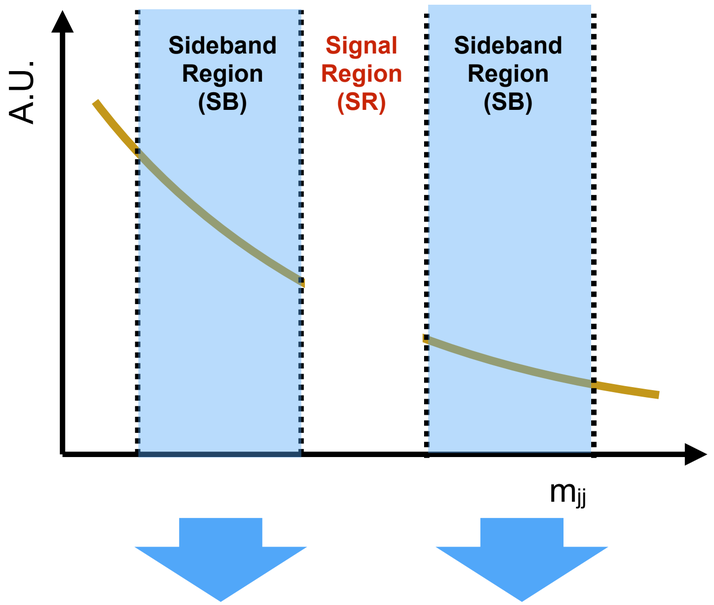
VAE as density estimator

Variational Autoencoder (VAE) to estimate 6-D density of the background distributions

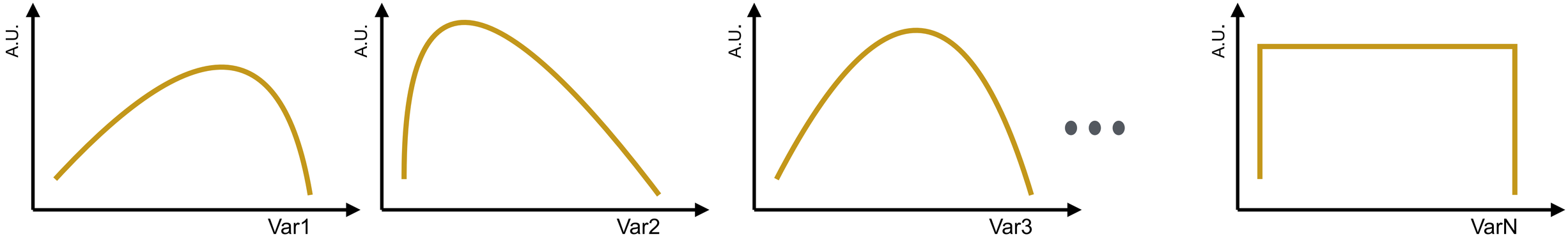


Loss: $L_{VAE} = (1 - \beta) \times L_{MSE} + \beta \times KL$

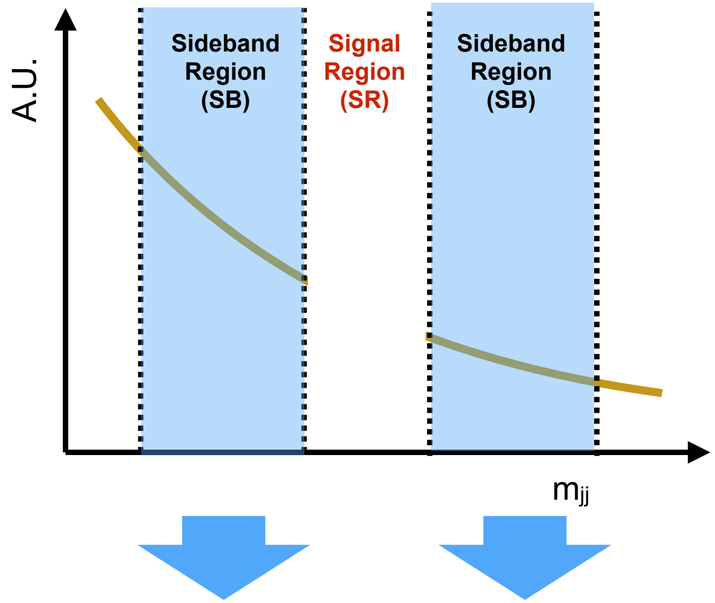
Predict the background in the signal region



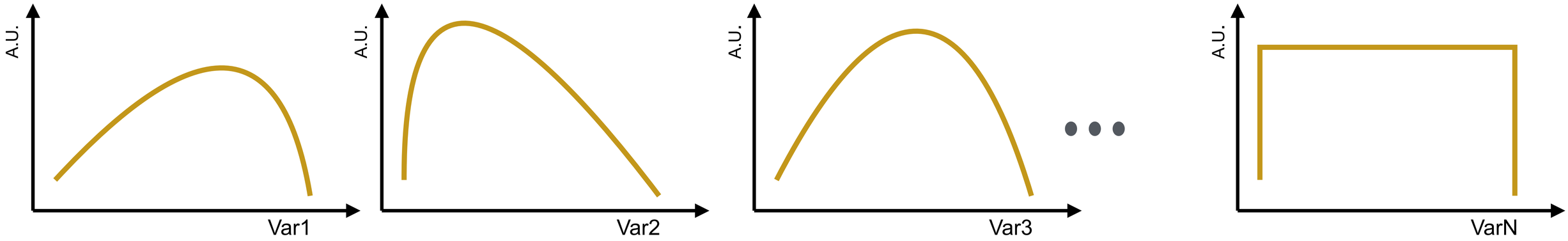
Model the multiple observables in the sideband regions



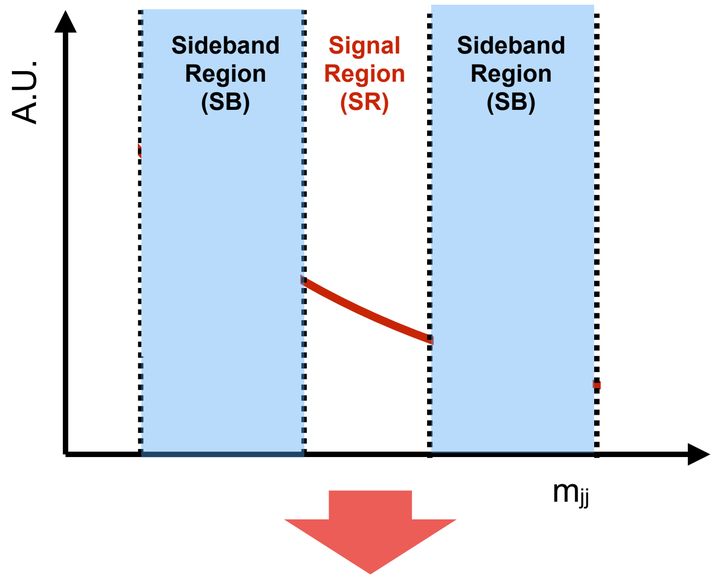
My Strategy to estimate the density



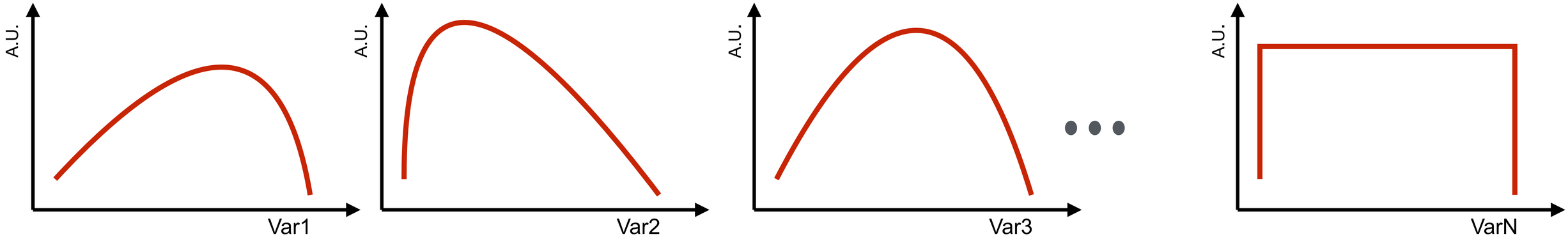
Generative models (Generative Adversarial Network or Variational Autoencoder) to estimate the densities



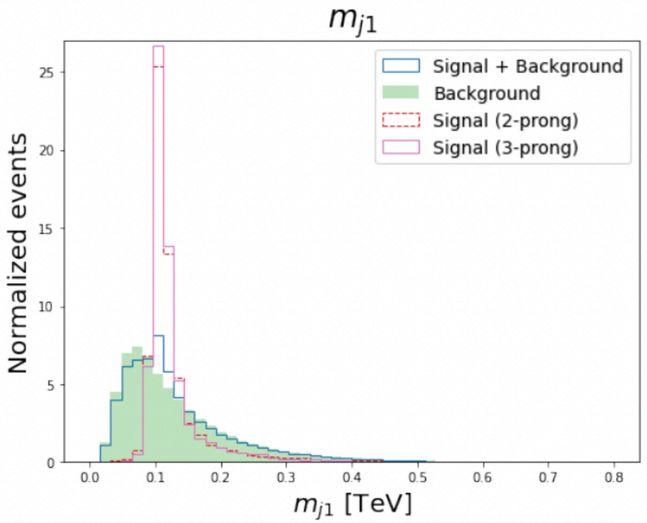
Model background in the side-band

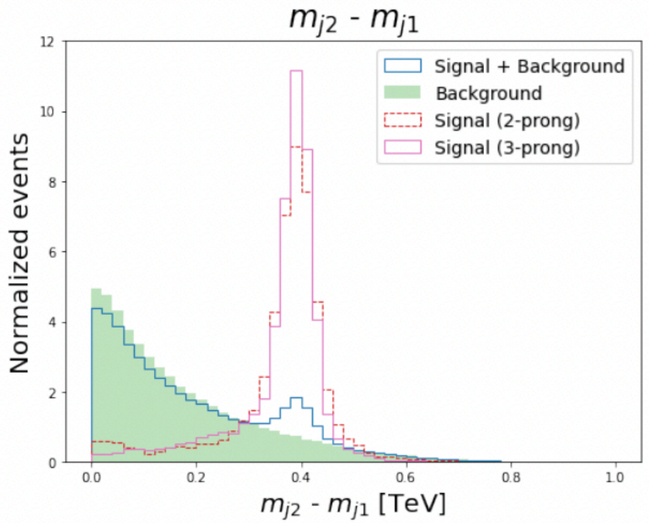


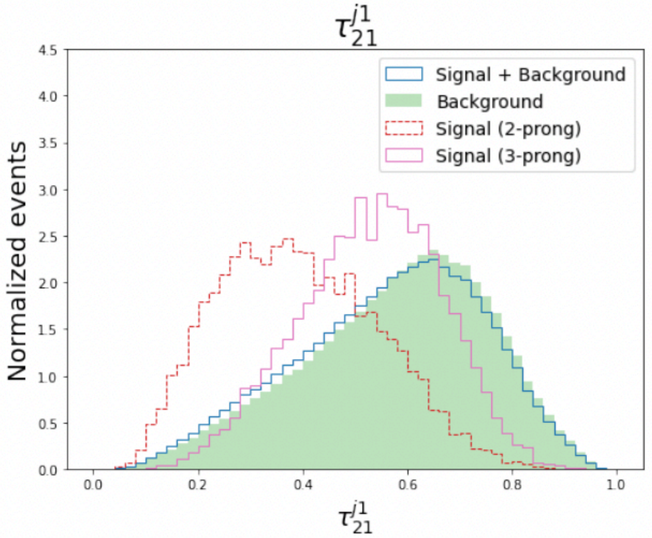
Predict them in the signal region

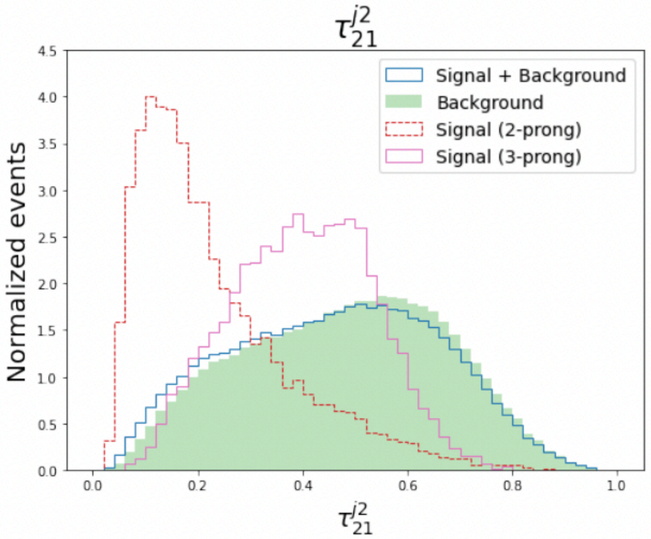


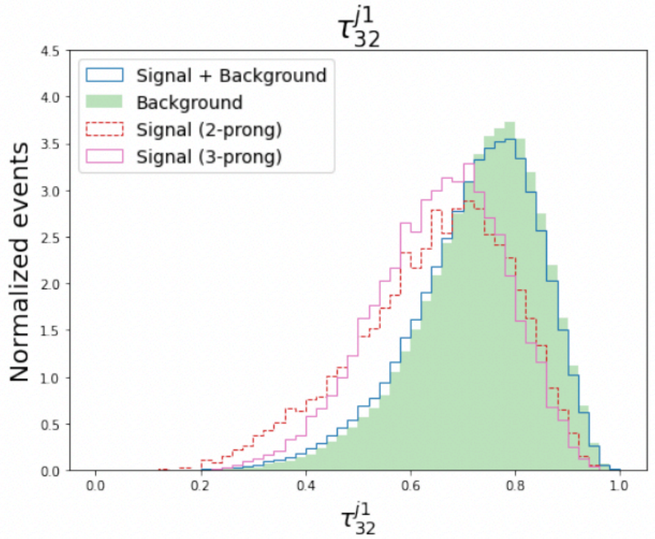
Features in the SR

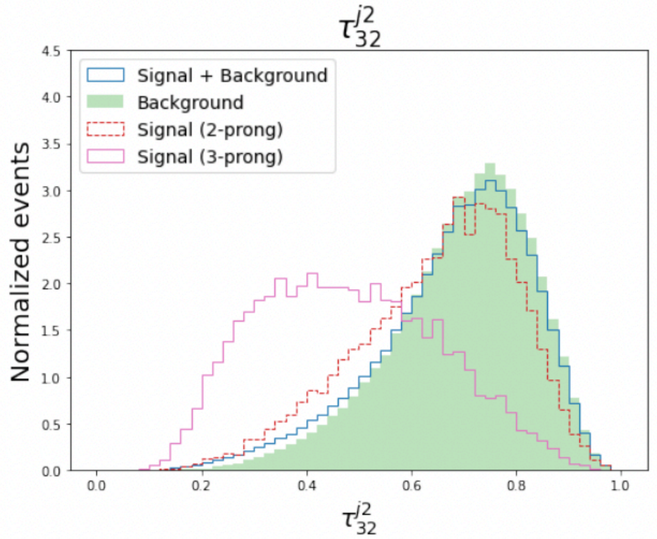




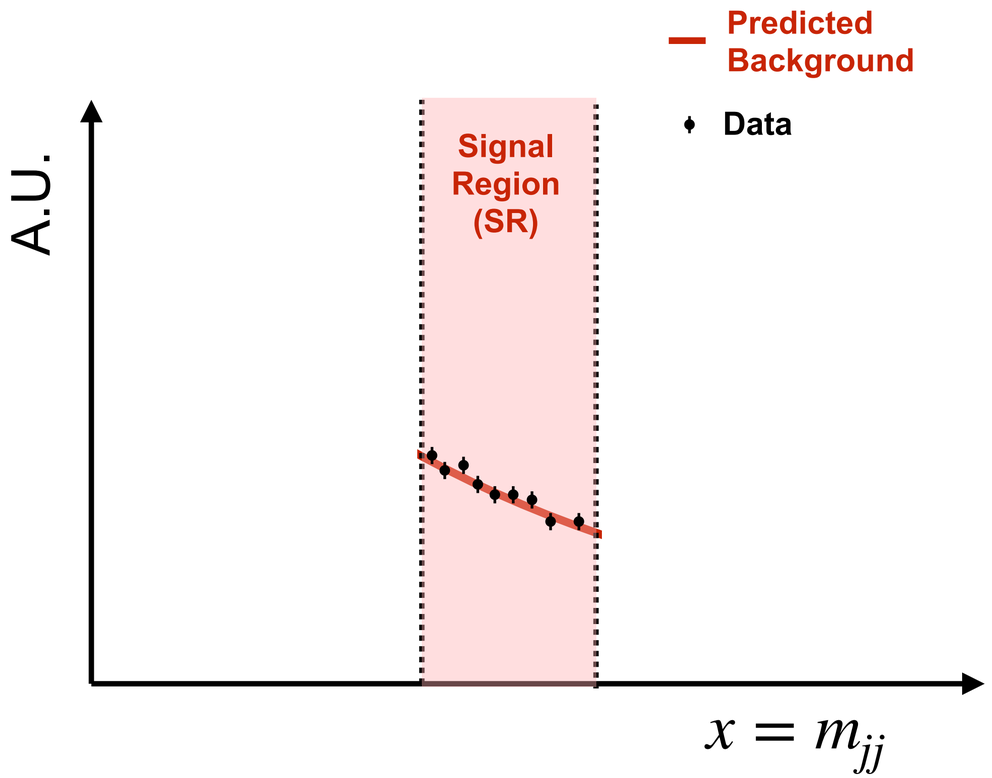






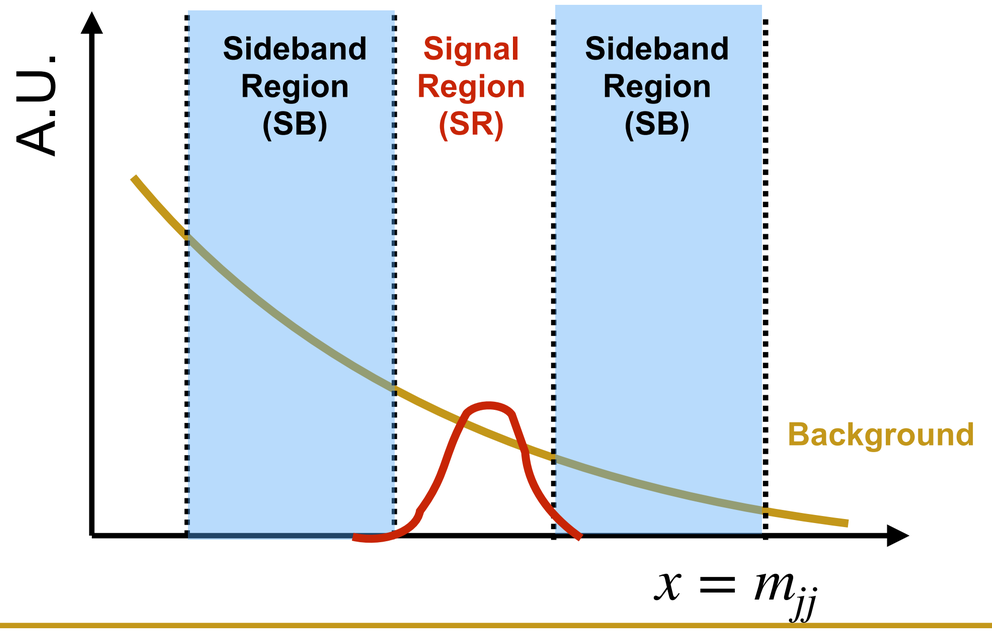


Compare Data and Prediction



Compare

Example:
ML classifier trained to differentiate signal-rich data vs background



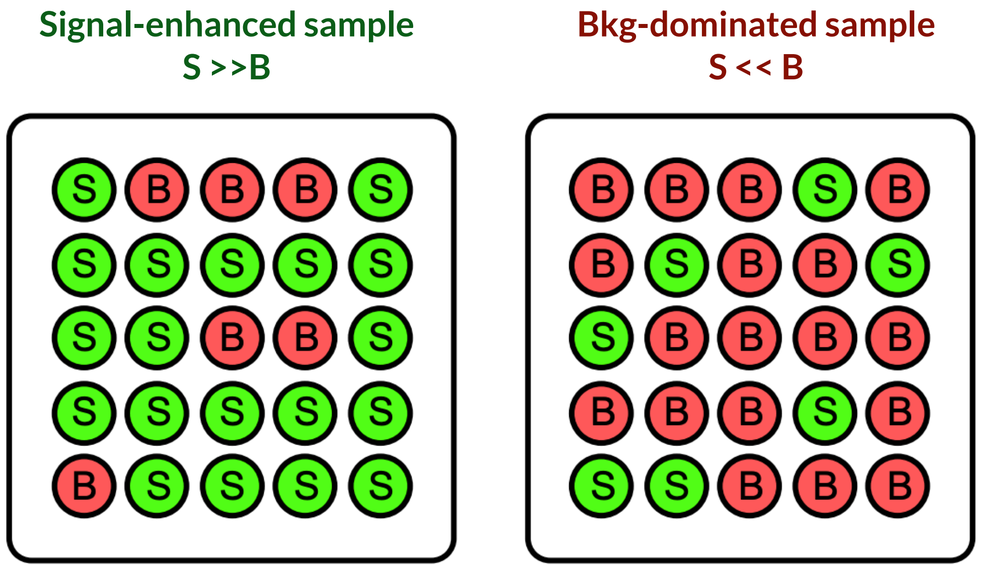
Weakly Supervised Method

Weakly supervised = noisy labels

Let's assume we have:
Two mixed samples of events with no label information

CWoLa (Classification Without Labels) method shows:
It is possible to train a classifier to distinguish red from green

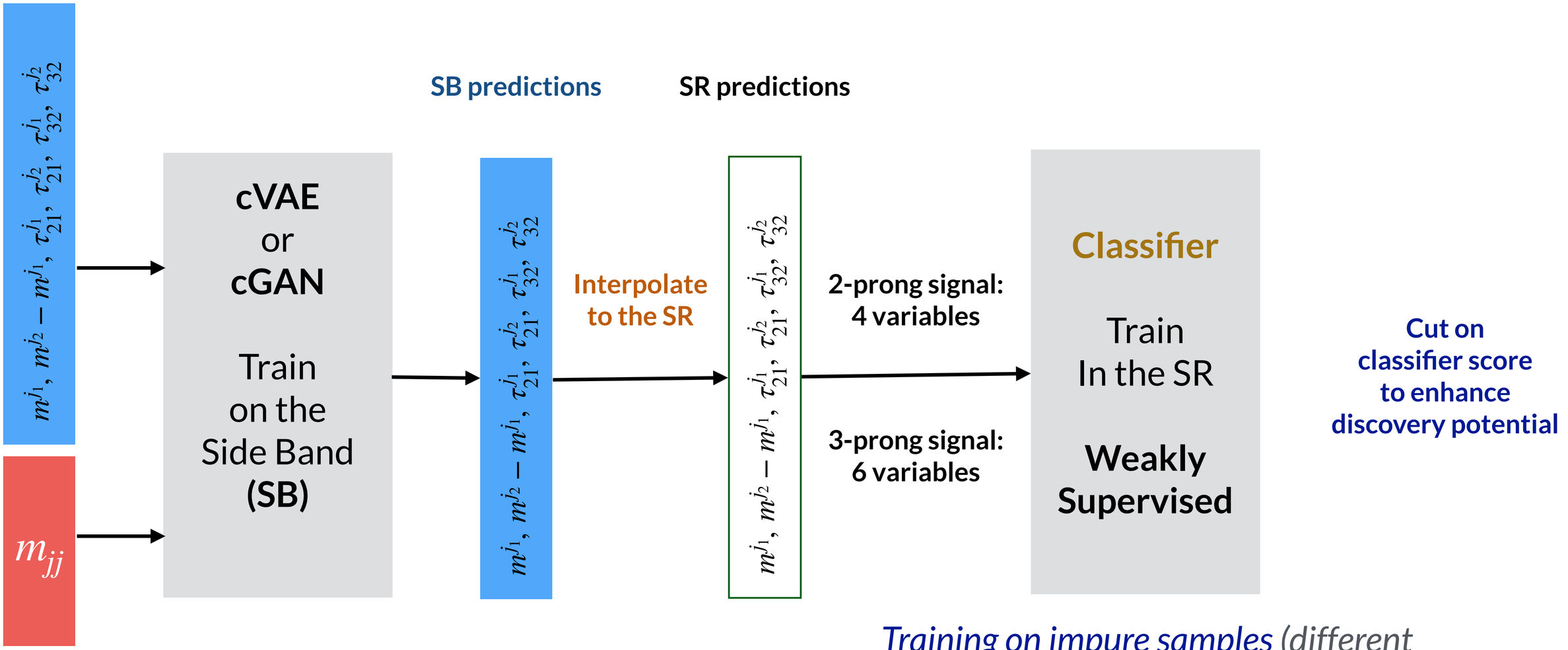
Training on impure samples (different admixture of S and B) is asymptotically equivalent of training on pure samples



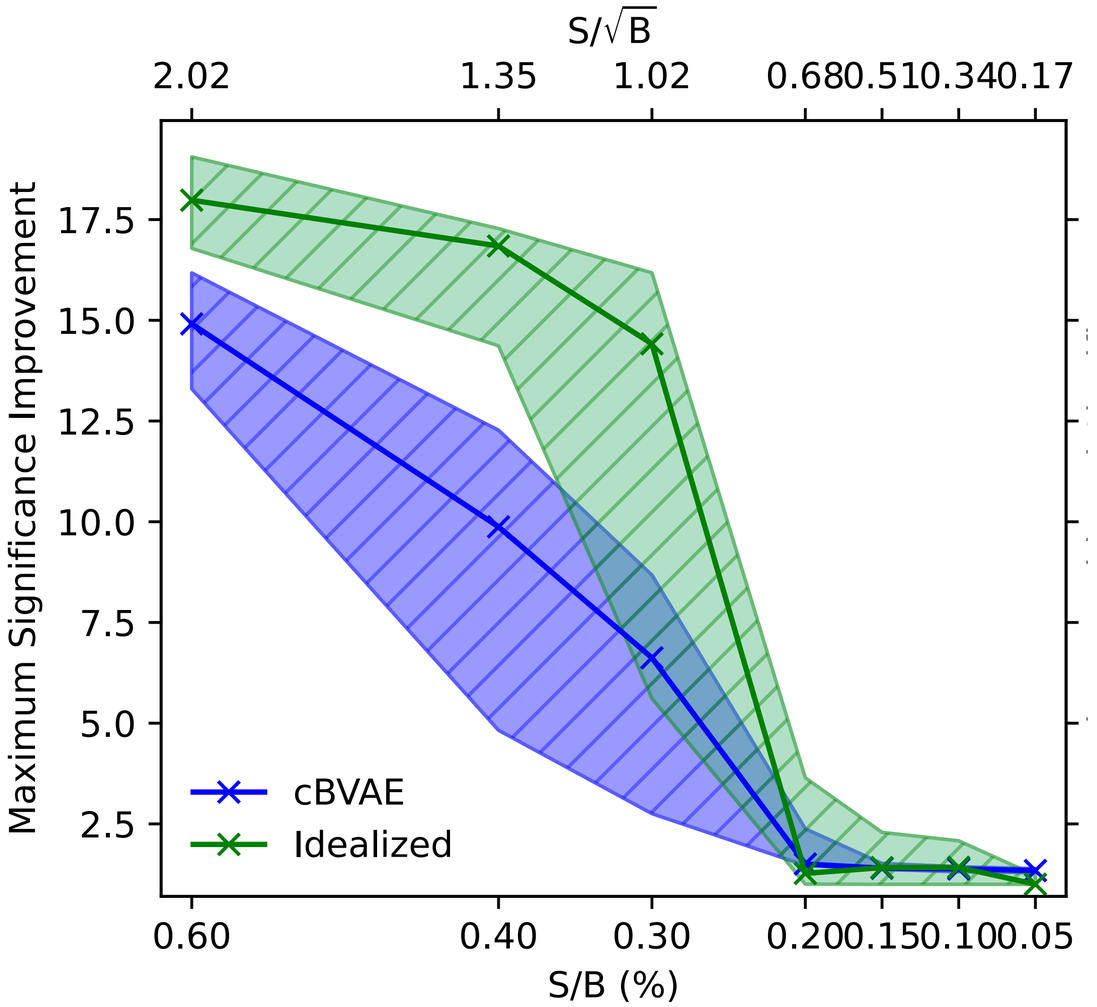
Classifier

JHEP 10 (2017) 174

Let's use Generative Models



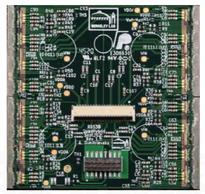
Training on impure samples (different admixture of S and B) is asymptotically equivalent of training on pure samples

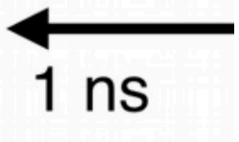

S/√B
2.02
1.35
1.02
0.68
0.51
0.34
0.17
Maximum Significance Improvement
17.5
15.0
12.5
10.0
7.5
5.0
2.5
cBVAE
Idealized
0.60
0.40
0.30
0.20
0.15
0.10
0.05
S/B (%)

How Fast Machine Learning contributes to such Scientific Discovery?

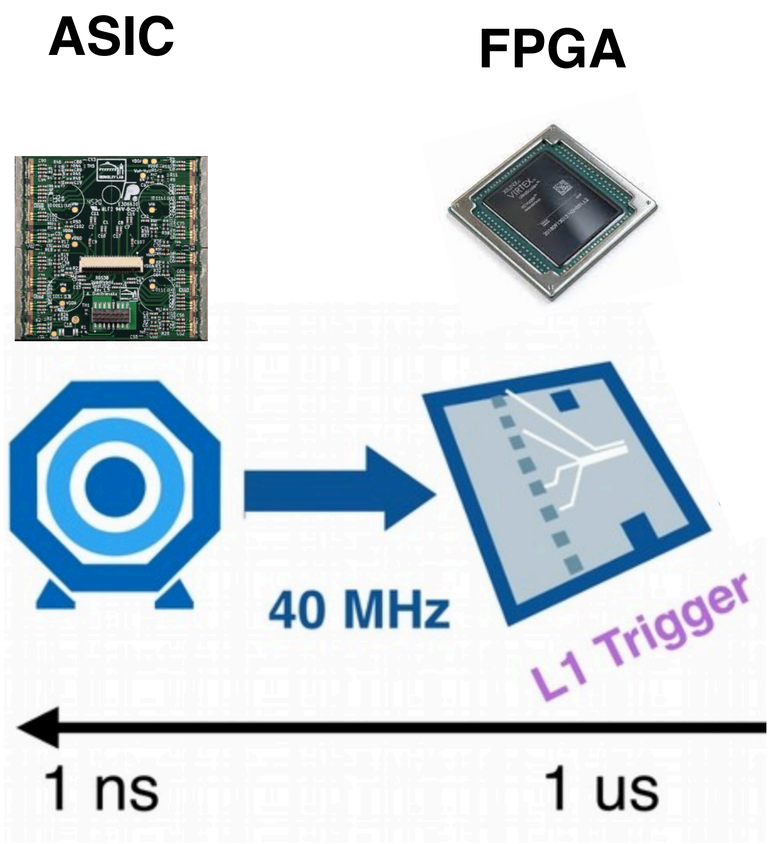
ATLAS Run-3 Data Processing

ASIC

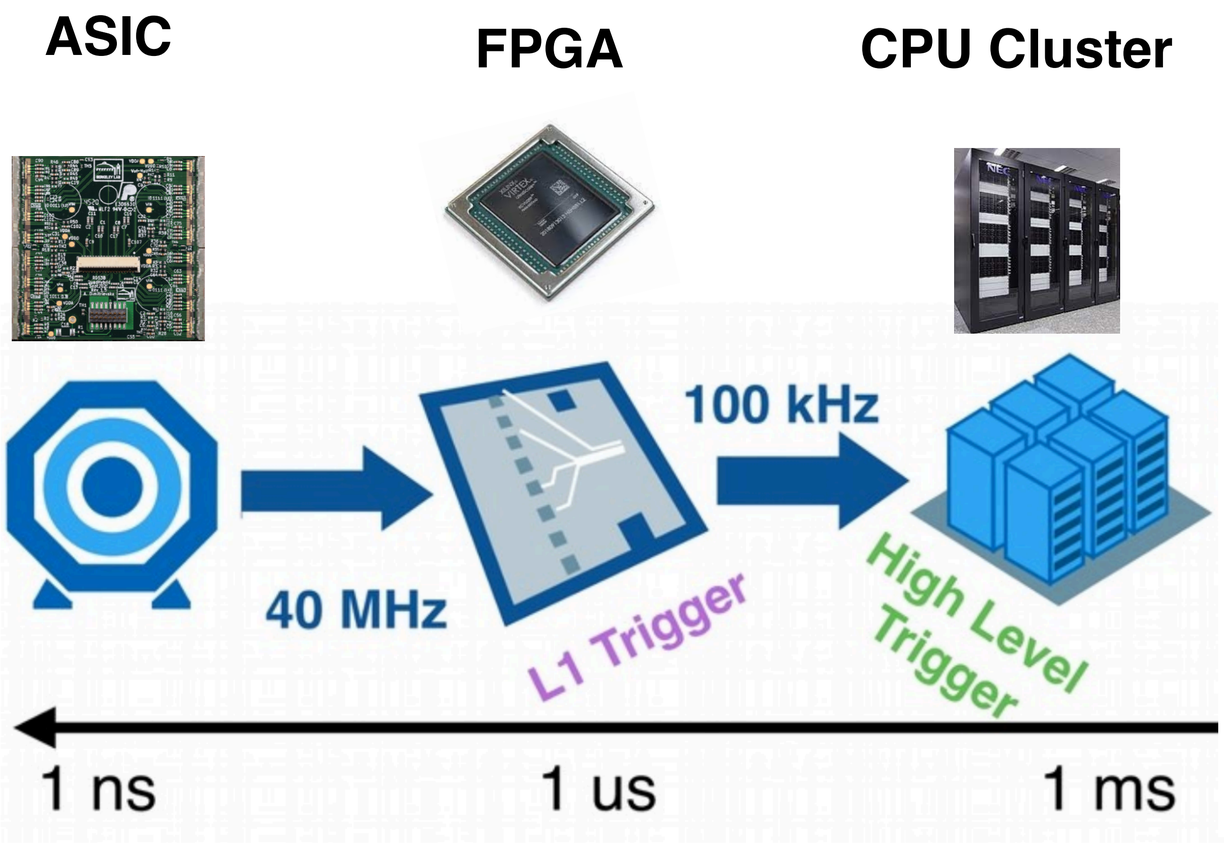




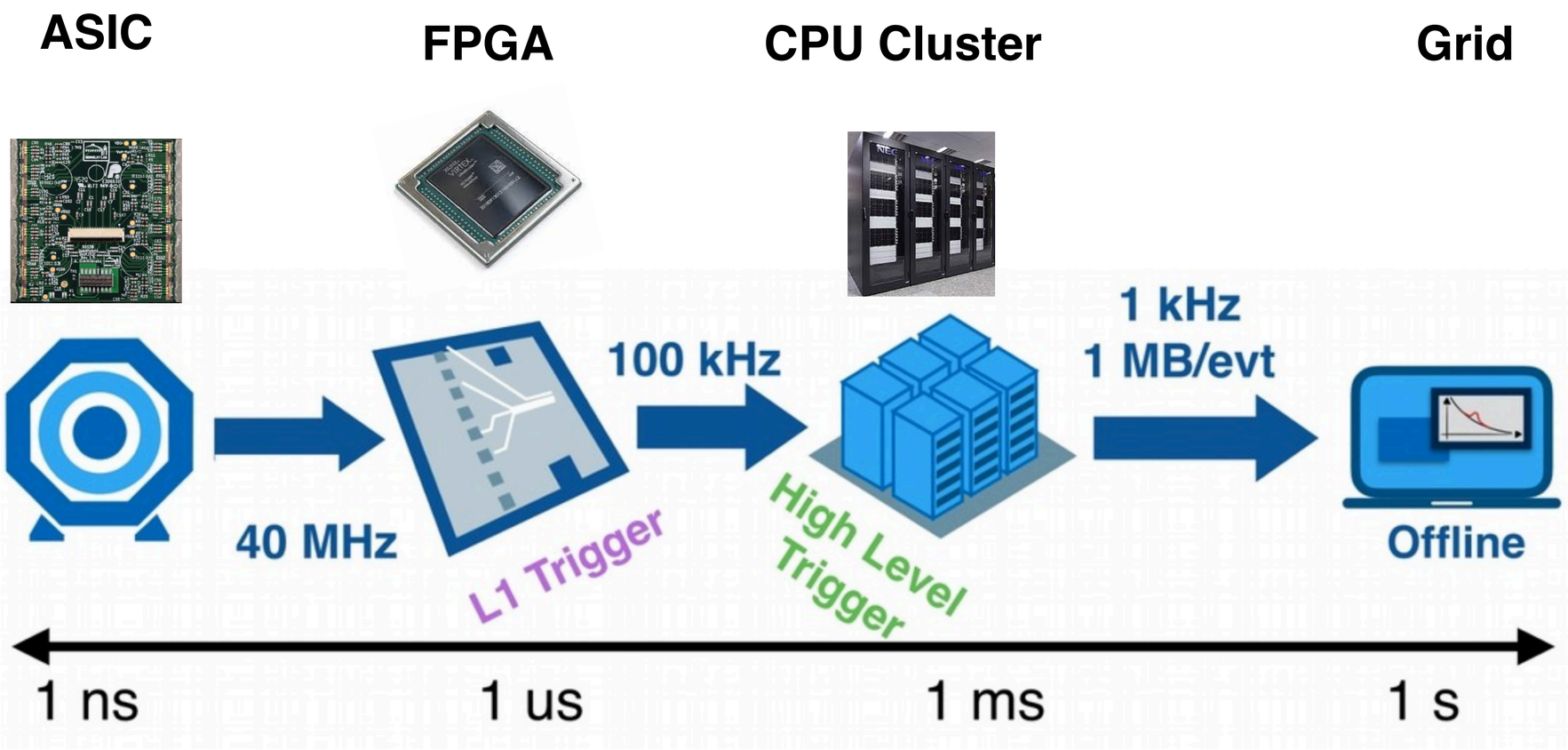
ATLAS Run-3 Data Processing



ATLAS Run-3 Data Processing

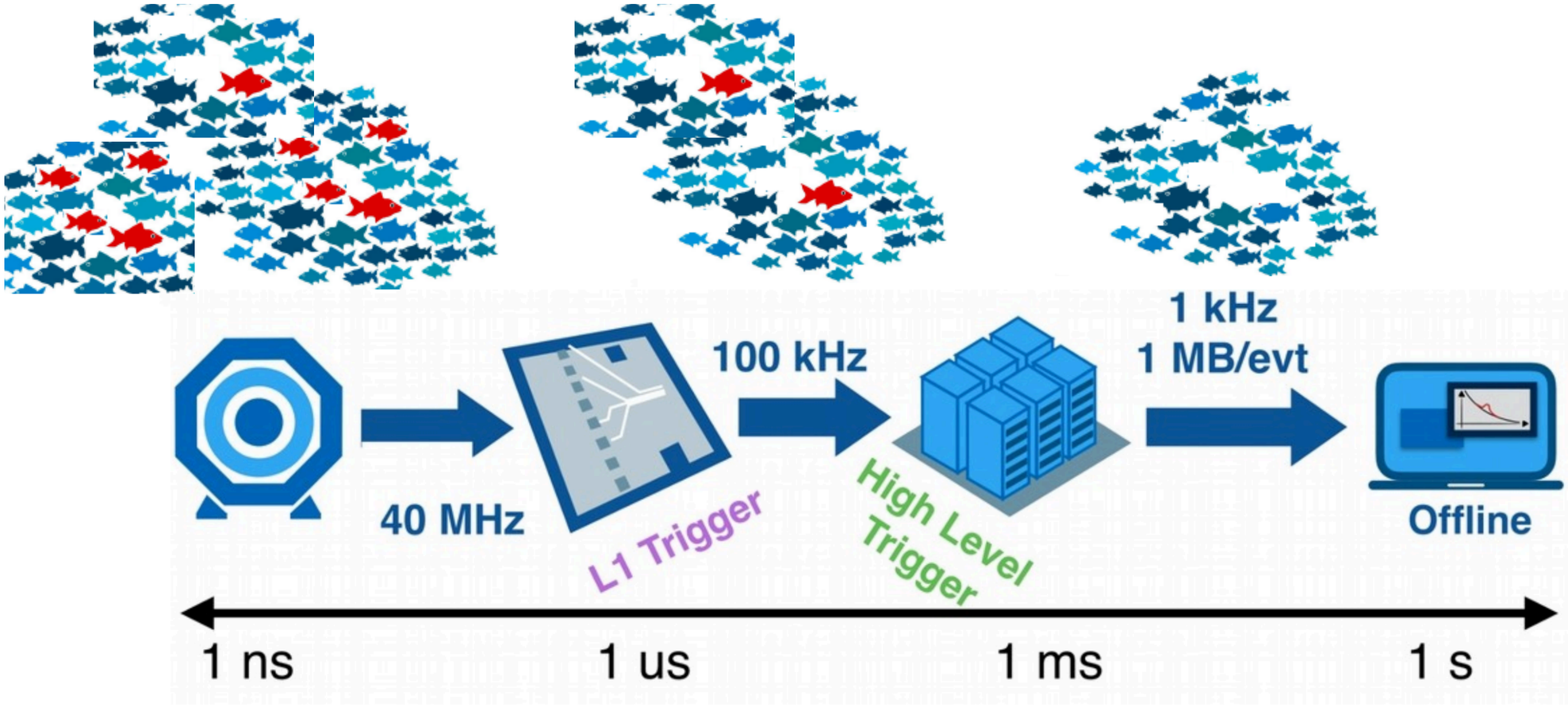


ATLAS Run-3 Data Processing

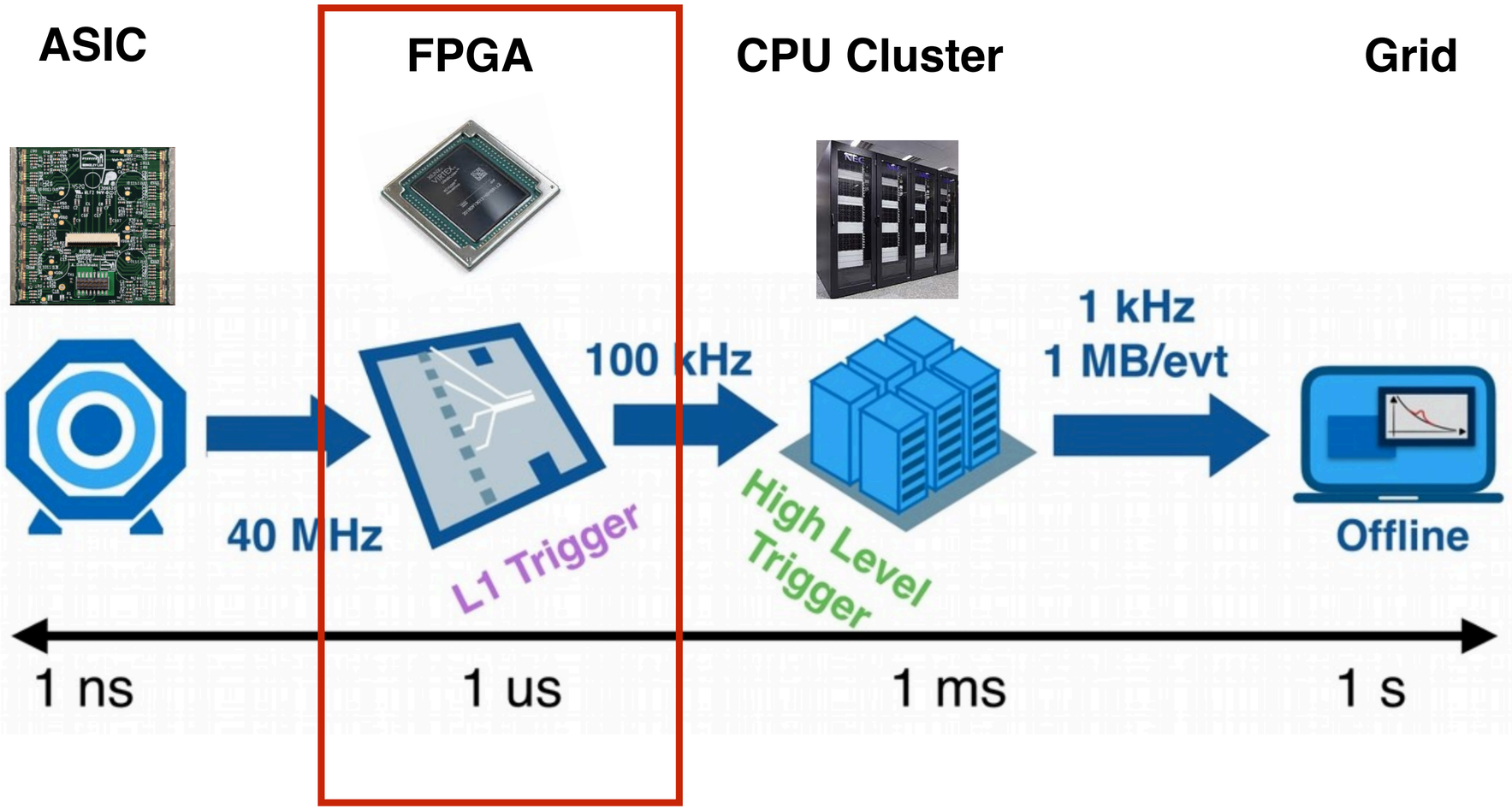


Are we storing them?

- New physics is clearly very good at hiding from us
- Depending on anomaly, we could have none left in recorded data



ATLAS Run-3 Data Processing



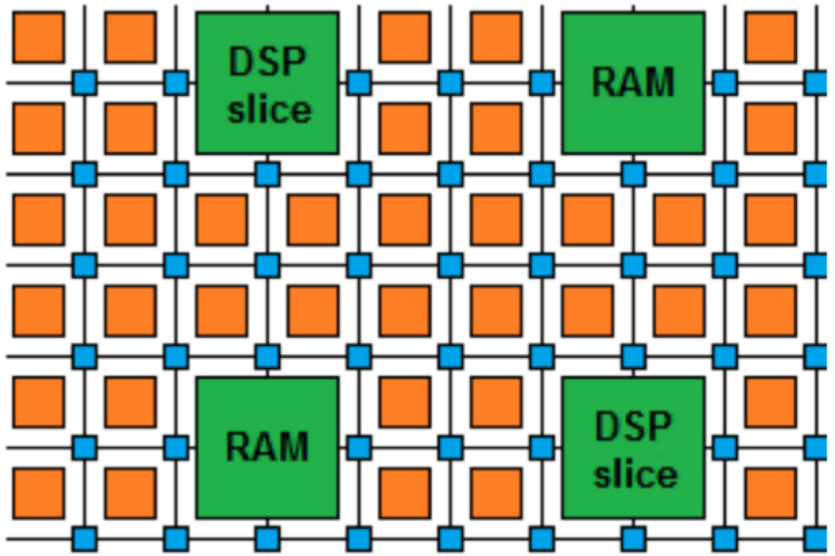
More ML algorithms here?

How to run an ML algorithm on FPGAs?

What is an FPGA?

Field **P**rogrammable **G**ate **A**rrays (FPGAs) are reprogrammable integrated circuits

- Contain many different building blocks (‘resources’) which are connected together as you desire
- Originally popular for prototyping ASICs, but now also for high performance computing



Building blocks:

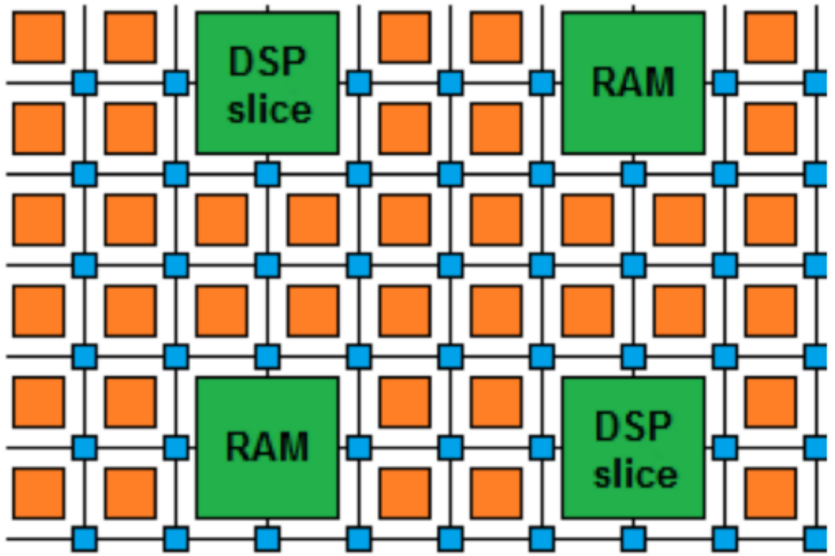
- **Multiplier units (DSPs)** [arithmetic]
- **Look Up Tables (LUTs)** [logic]
- **Flip-flops (FFs)** [registers]
- **Block RAMs (BRAMs)** [memory]

What is an FPGA?

- Run at high frequency - *O(100 MHz)*
  - Can compute outputs in O(ns)

- Low-level Hardware Description Language for programming
  Verilog/VHDL

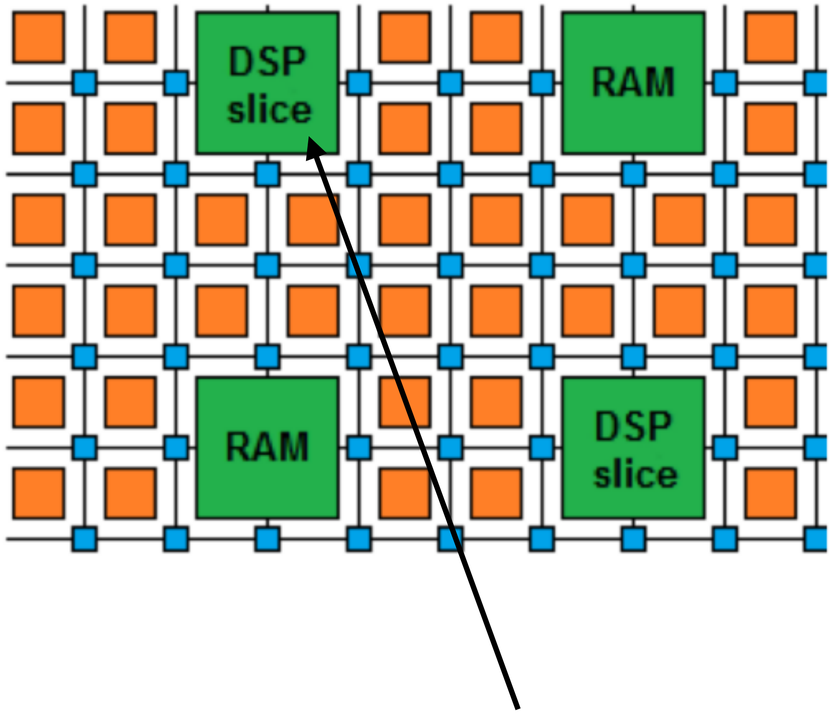
- Possible to translate C/C++ → Verilog/VHDL using High **Level Synthesis (HLS)** tools



Building blocks:
- **Multiplier units (DSPs)** [arithmetic]
- **Look Up Tables (LUTs)** [logic]
- **Flip-flops (FFs)** [registers]
- **Block RAMs (BRAMs)** [memory]

What is an FPGA?

- **DSPs** (Digital Signal Processor) are specialized units for multiplication and arithmetic
- DSPs are often the most scarce for NNs
- Faster and more efficient than using LUTs for these types of operations



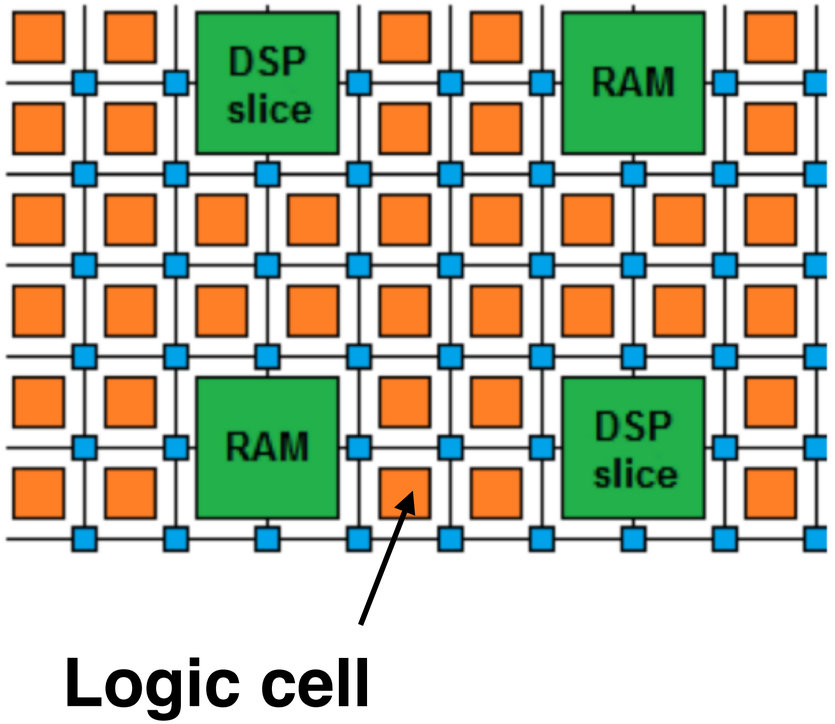
DSP
(multiplication)

Building blocks:
- **Multiplier units (DSPs)** [arithmetic]
- **Look Up Tables (LUTs)** [logic]
- **Flip-flops (FFs)** [registers]
- **Block RAMs (BRAMs)** [memory]

What is an FPGA?

- **Logic cells / Look Up Tables** perform arbitrary functional operations on small bit-width inputs (2-6)
  - boolean, arithmetic
  - small memories

- **Flip-Flops** control the flow of data with the clock pulse



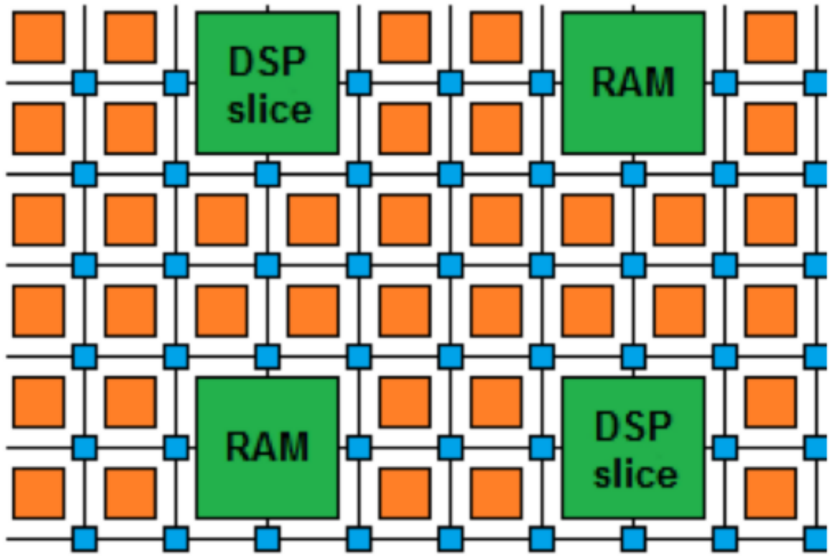
Building blocks:

– **Multiplier units (DSPs)** [arithmetic]
– **Look Up Tables (LUTs)** [logic]
– **Flip-flops (FFs)** [registers]
– **Block RAMs (BRAMs)** [memory]

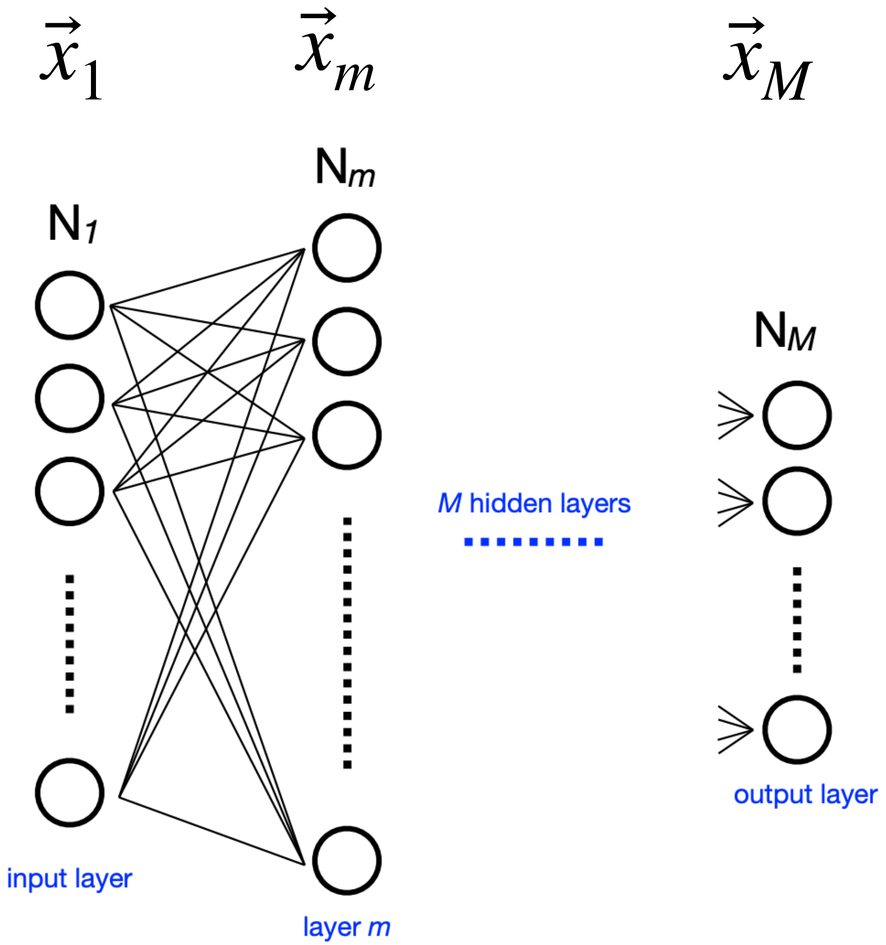
Example: AMD Xilinx FPGA

AMD Xilinx VIRTEX UltraScale+ VU13P

- 12288 Multipliers
- 1.7M LUTs
- 3.4M FFs
- 95 Mb BRAM



Inference on an FPGA



$$\vec{x}_m = g_m\left(W_{m,m-1}\vec{x}_{m-1} + \vec{b}_m\right)$$

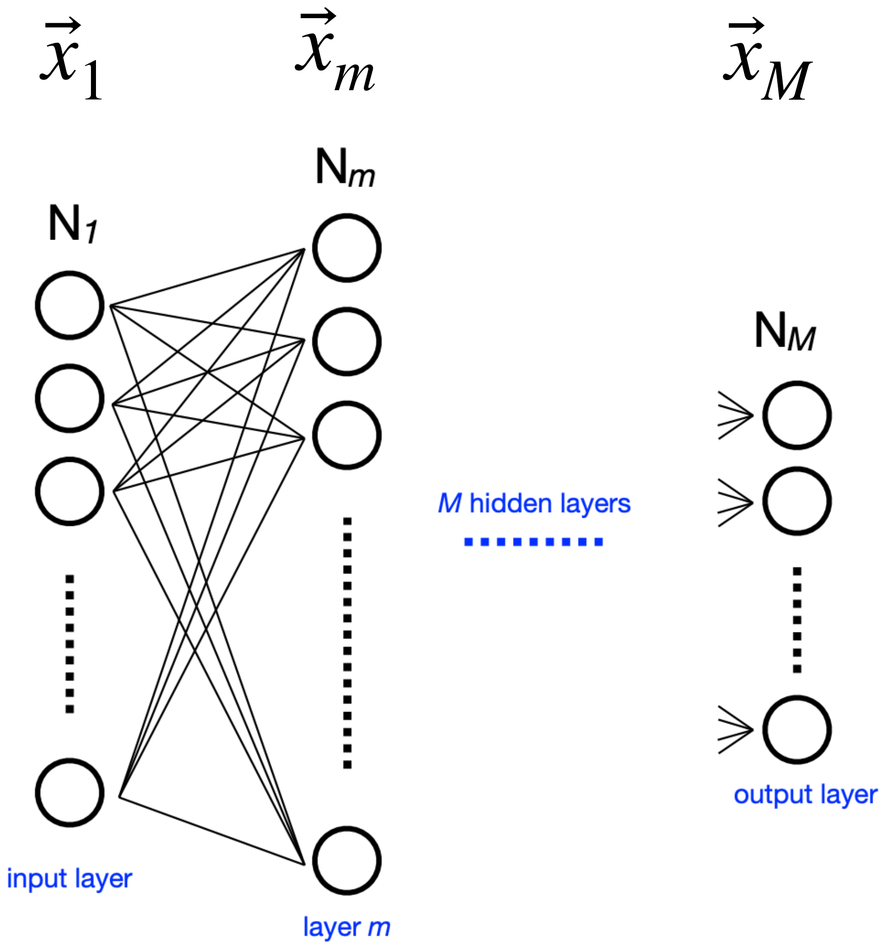
multiplication

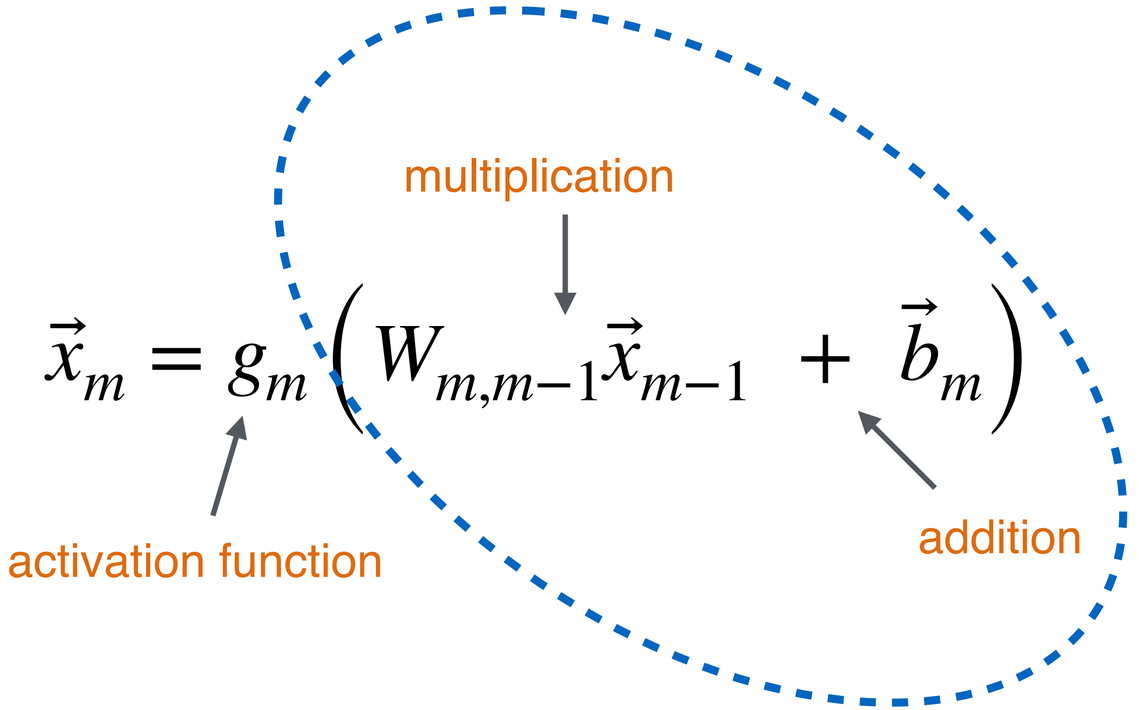
activation function

addition

Credit: Dylan Rankin

Inference on an FPGA



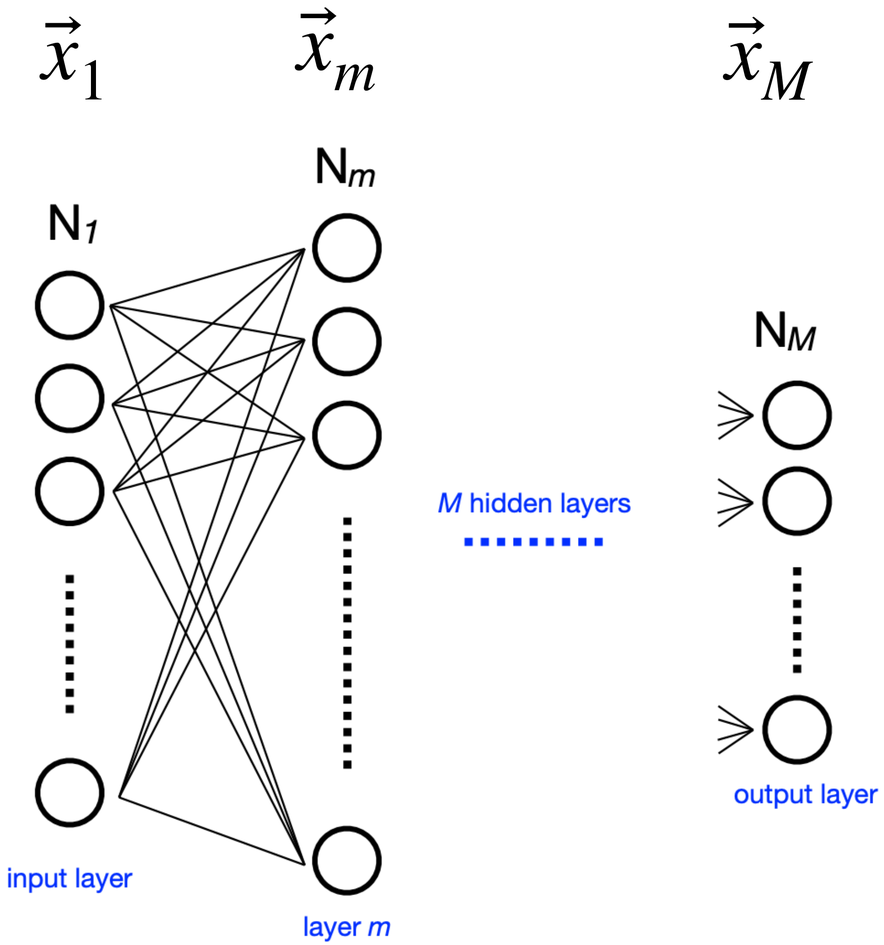


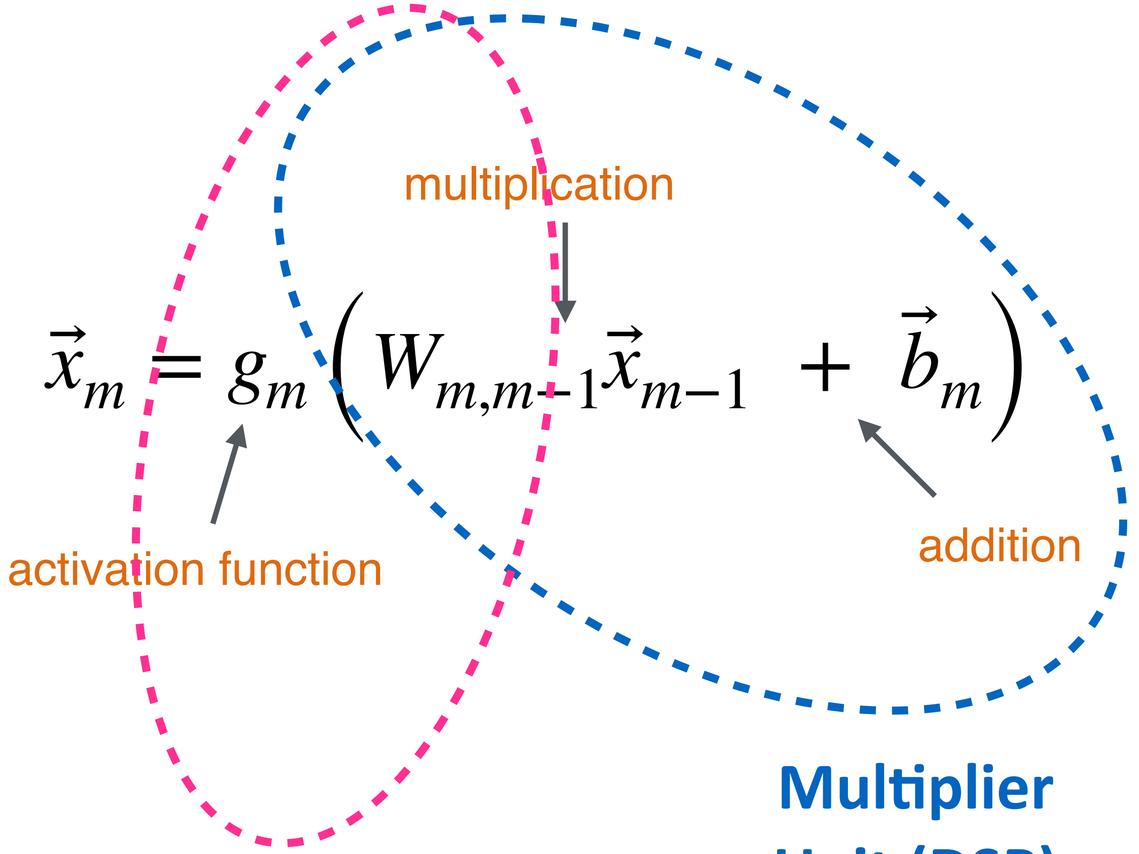
Multiplier Unit (DSP)

up to ~6k parallel operation (VU9P)

Credit: Dylan Rankin

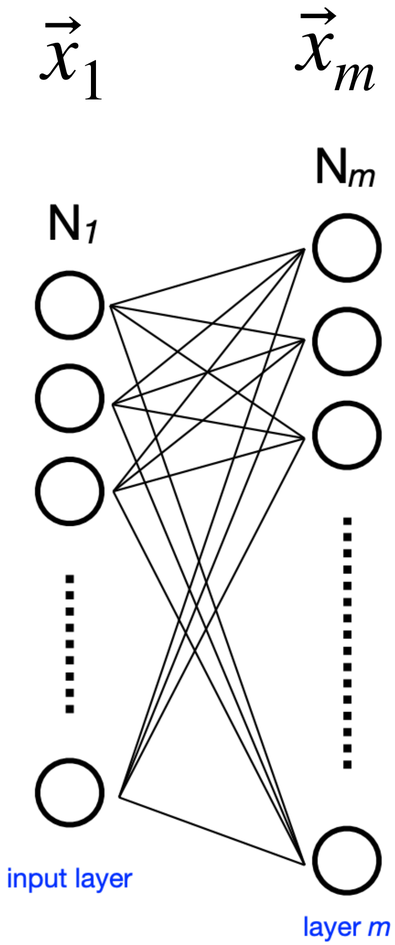
Inference on an FPGA





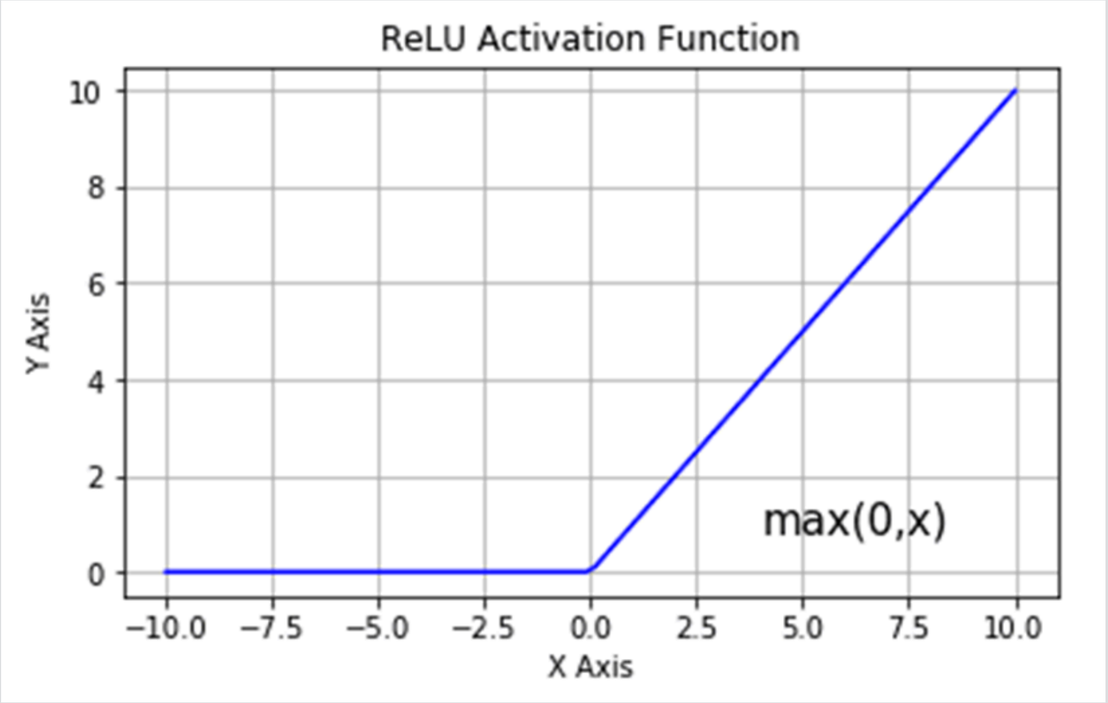
Credit: Dylan Rankin

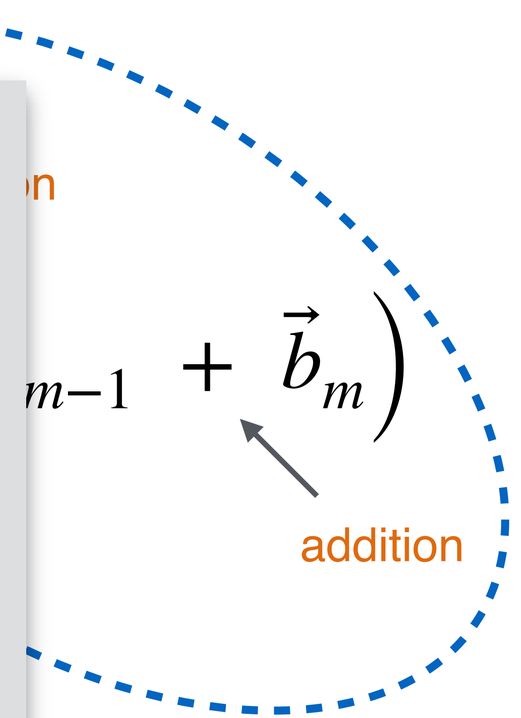
Inference on an FPGA



M hidden layers

Hardware efficient activation function





Multiplier Unit (DSP)

Credit: Dylan Rankin

High Level Synthesis with Machine Learning (hls4ml)

Usual ML software workflow

(Q) Keras + TensorFlow

(Q) ONNX

PYTORCH

hls 4 ml

HLS Conversion

HLS Project

Tune

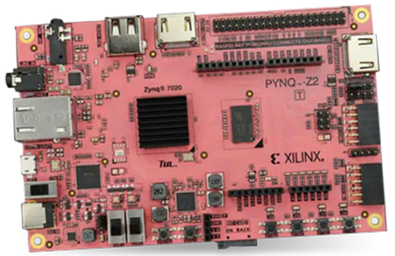
Optimized HDL code

Vivado™ HLS

intel HLS COMPILER

VIVADO

intel Quartus® Prime Design Software



Bit Files for FPGA

https://fastmachinelearning.org/hls4ml/
arXiv:2103.05579

High Level Synthesis with Machine Learning (hls4ml)

https://fastmachinelearning.org/hls4ml/
arXiv:2103.05579

A software interface for implementing Neural Networks on an FPAG

- Supports many common layer like DNN, CNN, RNN, Graph NN etc
- Transformers were not implemented until December, 2023

All these ML algorithms could be used for several low-level tasks
Example:
- Jet Energy Calibration
- Missing Transverse Energy reconstruction

Resource Limitation

Regardless of toolkit,
FPGA size is the limitation of doing low latency ML

- Bigger FPGA → more resources → more computation → larger networks

AMD Xilinx VIRTEX UltraScale+ VU13P

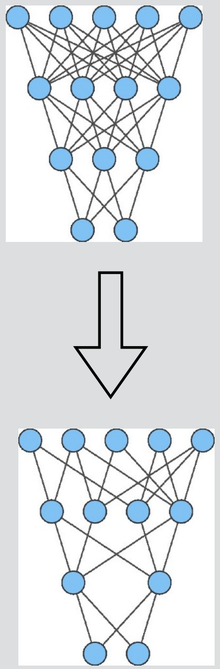
- 12288 Multipliers
- 1.7M LUTs
- 3.4M FFs
- 95 Mb BRAM

So efficient model design to reduce weights is crucial

Efficient Model Design

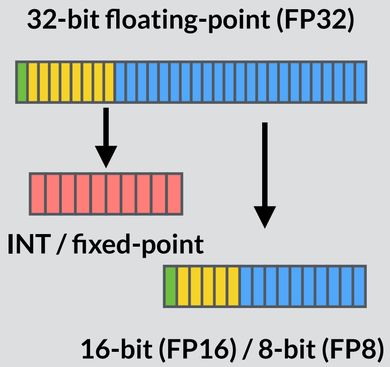
Pruning

Remove synapses and neurons



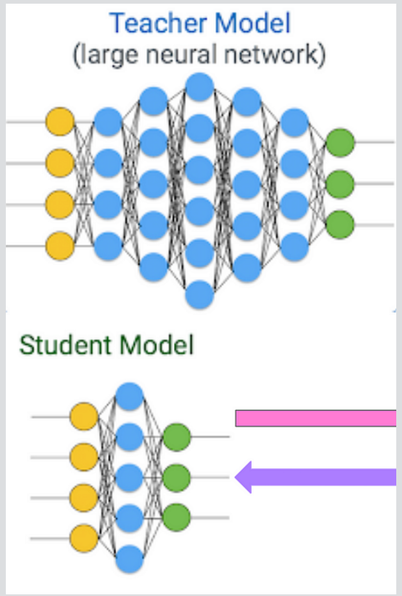
Quantization

Reducing precision



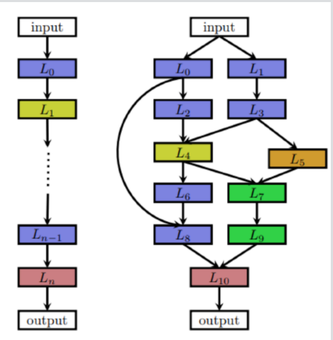
Knowledge Distillation

Train a smaller model using a bigger model



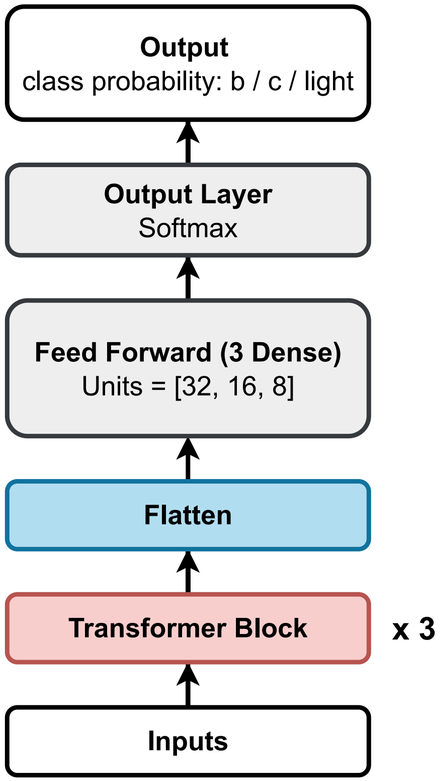
Neural Architecture Search

Finding the optimal model architecture



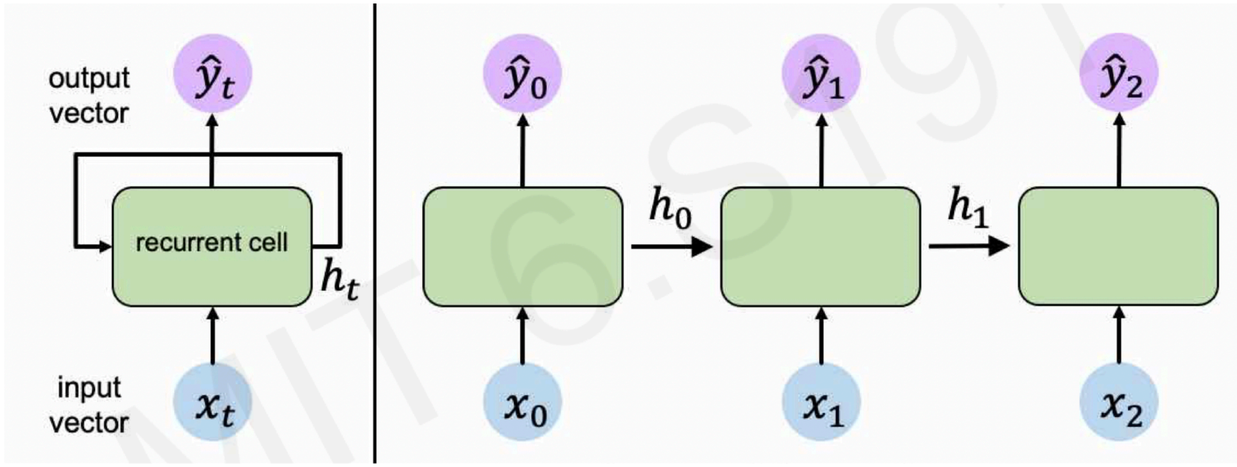
ML Inference with FPGA

Tranformers



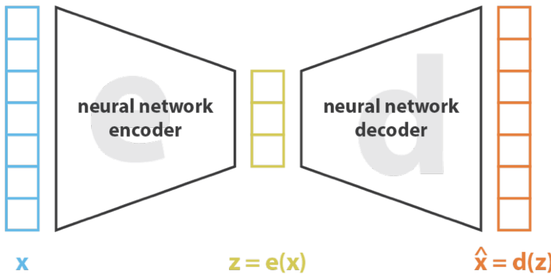
EK et al, NeurIPS 2023, arXiv 2402.01047

Recurrent Neural Networks



EK et al, Mach. Learn.: Sci. Tec. 4 025004

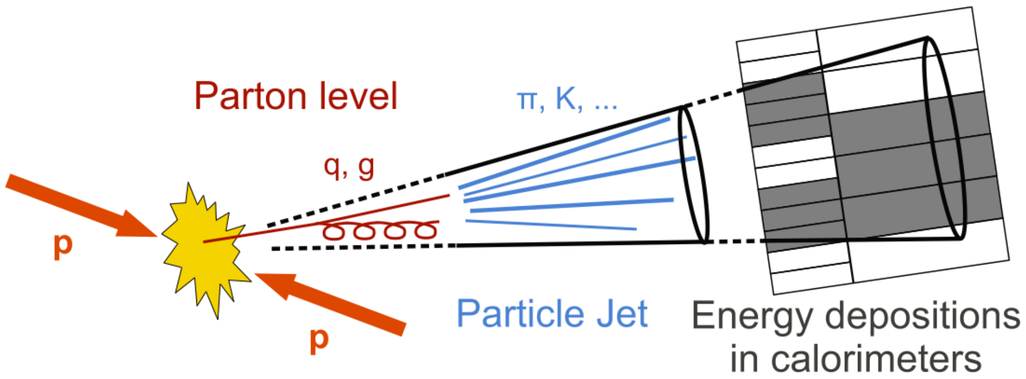
Autoencoders and Variational Autoencoders

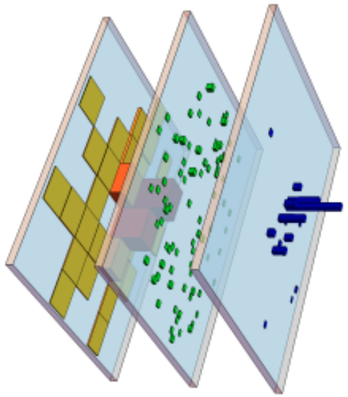


loss = $||x - \hat{x}||^2 = ||x - d(z)||^2 = ||x - d(e(x))||^2$

EK et al, FastML@ICCAD 2023
arXiv 2402.04274

Example: Jet tagging





Fixed Set of variables
Dense Neural Network

Variable sets
Deep sets model

Images
Convolution NN

Sequence of particles
Recurrent NN, Transformers

Graphs
Graph Neural Network

b-tagging with transformer model

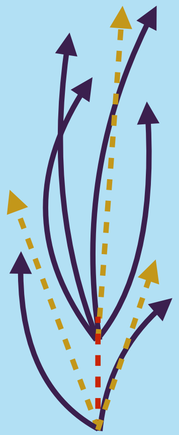
Identify b/c jets from light-flavored jets

Jets are treated as sequence of particles

b / c jets

Jets from
b- or c-quark

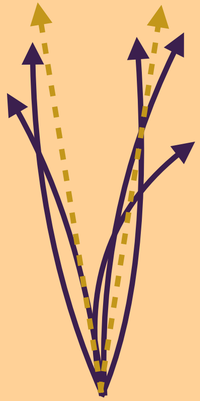
→ Displaces
vertex



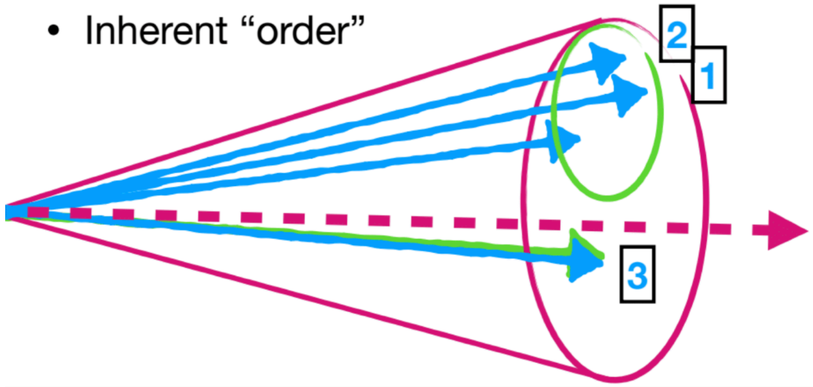
light jets

Jets from
u/d/s-quarks

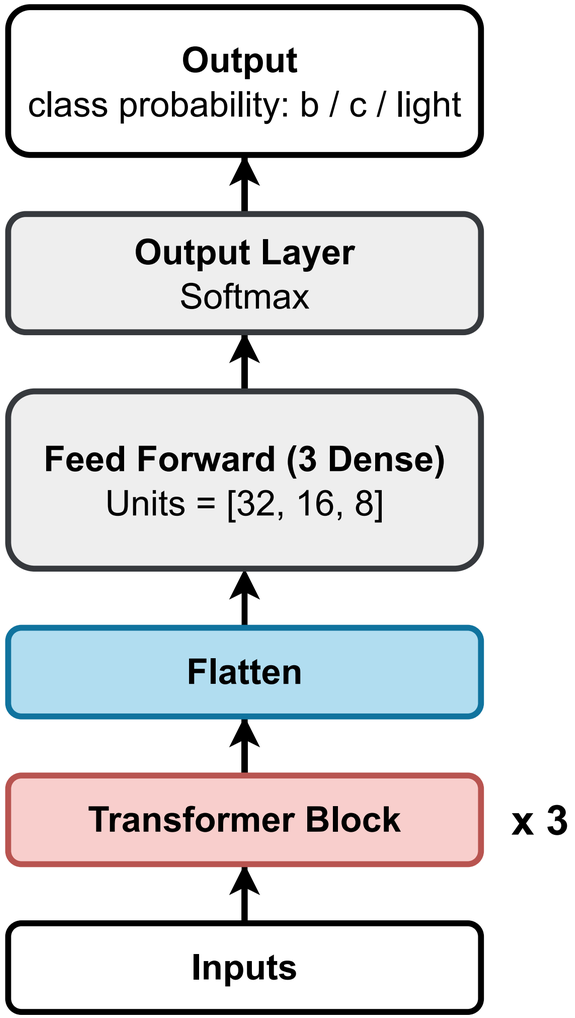
→ no specific
structure



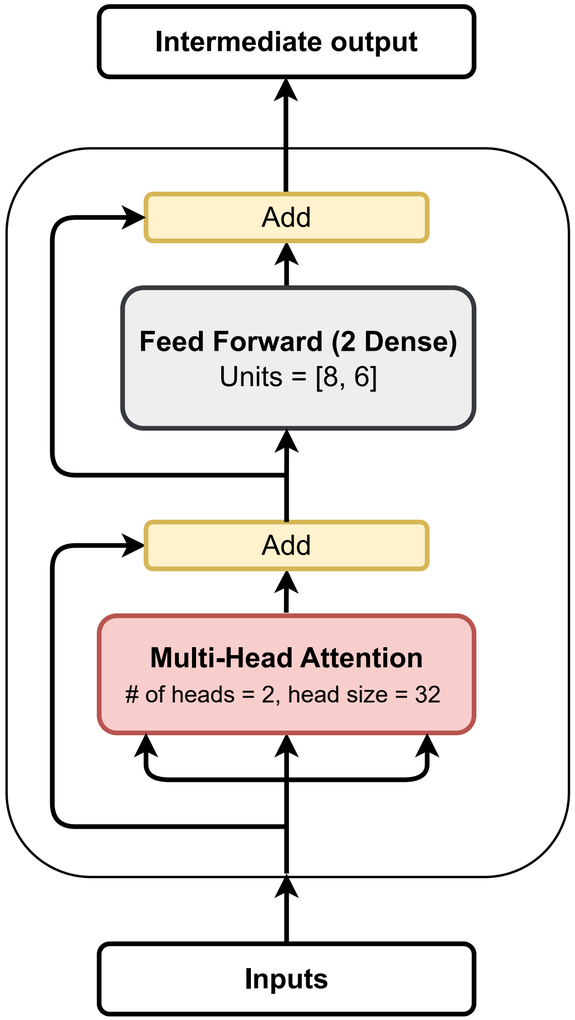
- Very effective for natural sequences (collection of particles)
  - Inherent “order”



Transformer Architecture



x 3



No. of parameters ~9k

Note:

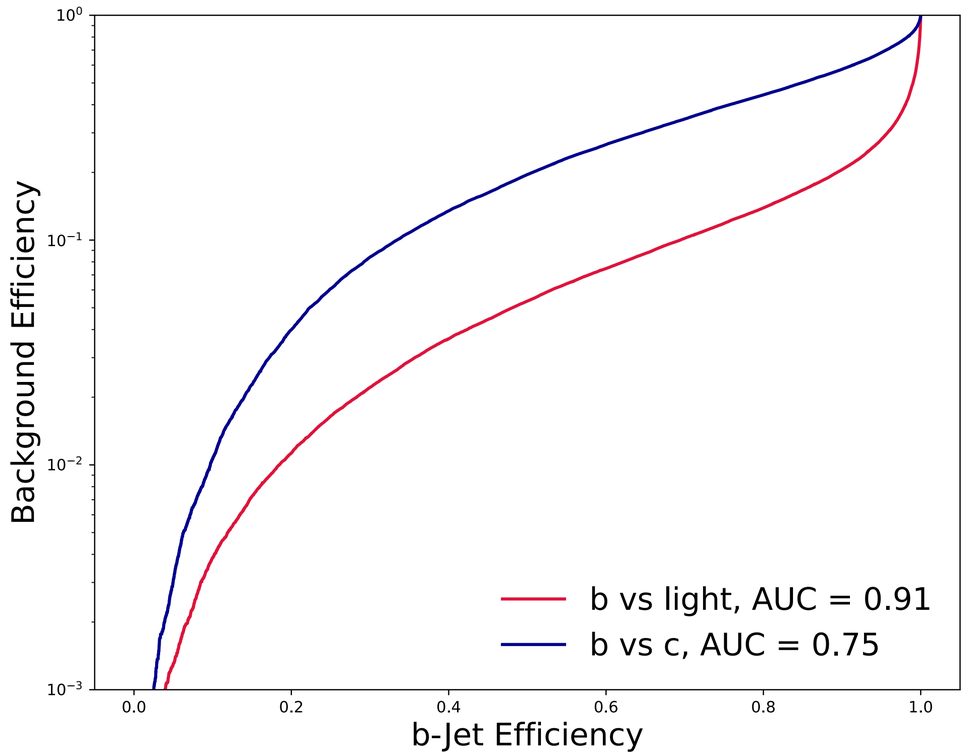
Positional encoding was not used for this model

- It is not crucial for this application

$$O_h = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Model performance: ROC

- All the benchmark models are trained using **Keras + TensorFlow**
- Weights and biases are represented by 32 bit floating point numbers



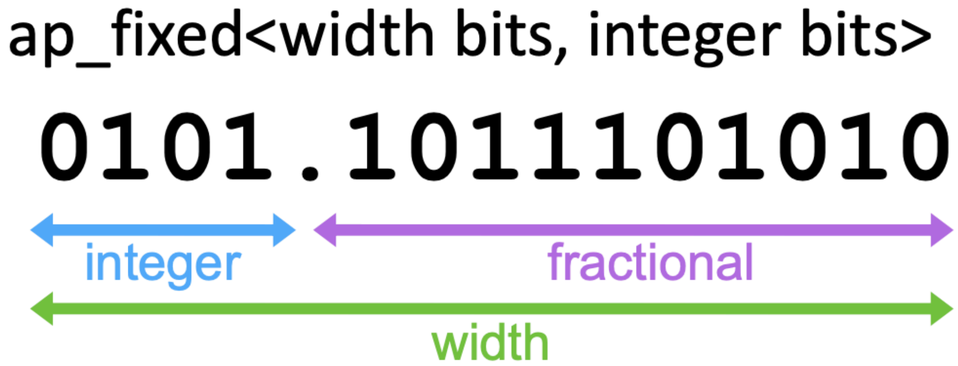
Quantization

Quantization – Reducing the bit precision used for NN arithmetic

Why this is necessary?

- Floating-point operations (32 bit numbers) on an FPGA consumes large resources
- Not necessary to do it for desired performance

- **hls4ml uses fixed-point representation for all computations**
  - Operations are integer ops, but we can represent fractional values



Example:

<20, 10>

Total Width

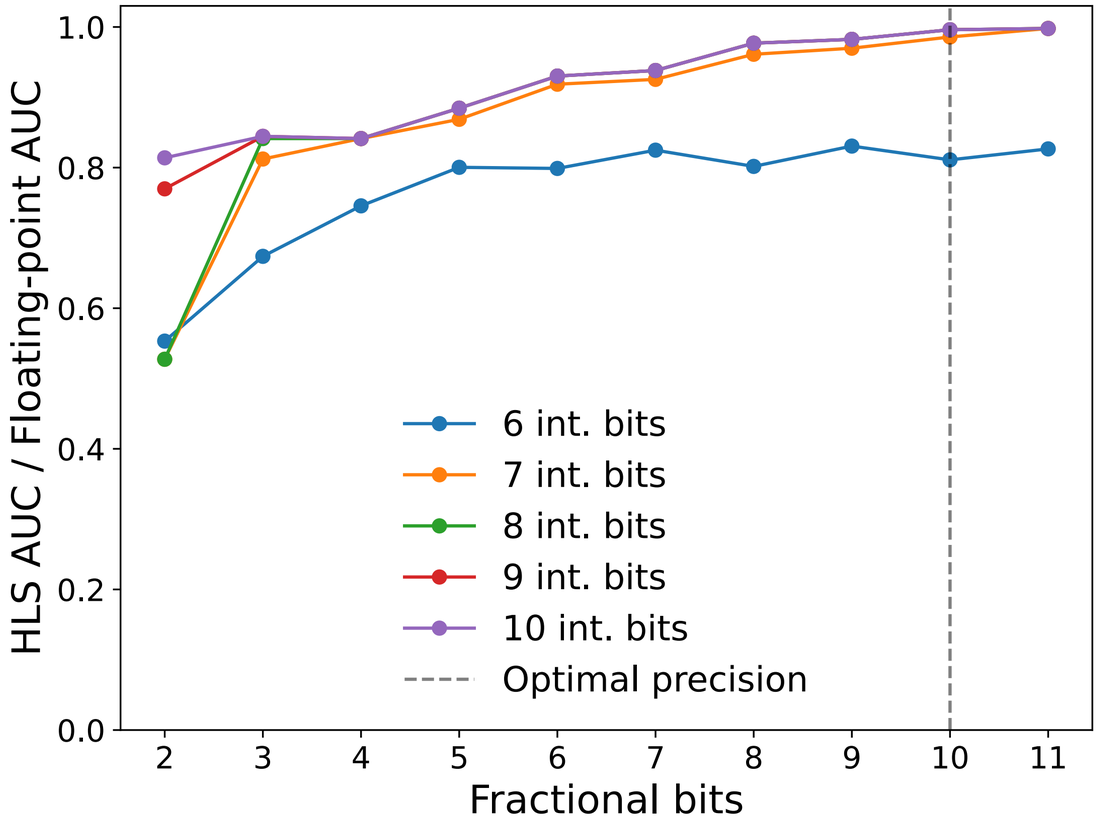
Integer

AUC after HLS conversion

$$\text{Relative AUC} = \frac{\text{HLS AUC}}{\text{Floating} - \text{point AUC}}$$

- Post-training quantized **Transformer models** (with optimal precision) performs similar to the floating-point models

Best Config
Integer bits = 10
Fractional bits = 10



Quantization Strategies

Usual ML software workflow

(Q) Keras + TensorFlow

(Q) ONNX

PYTORCH

Post Training Quantization

hls4ml

HLS Conversion

(hls4ml)

Initial Model

Usual ML software workflow

(Q) Keras + TensorFlow

(Q) ONNX

PYTORCH

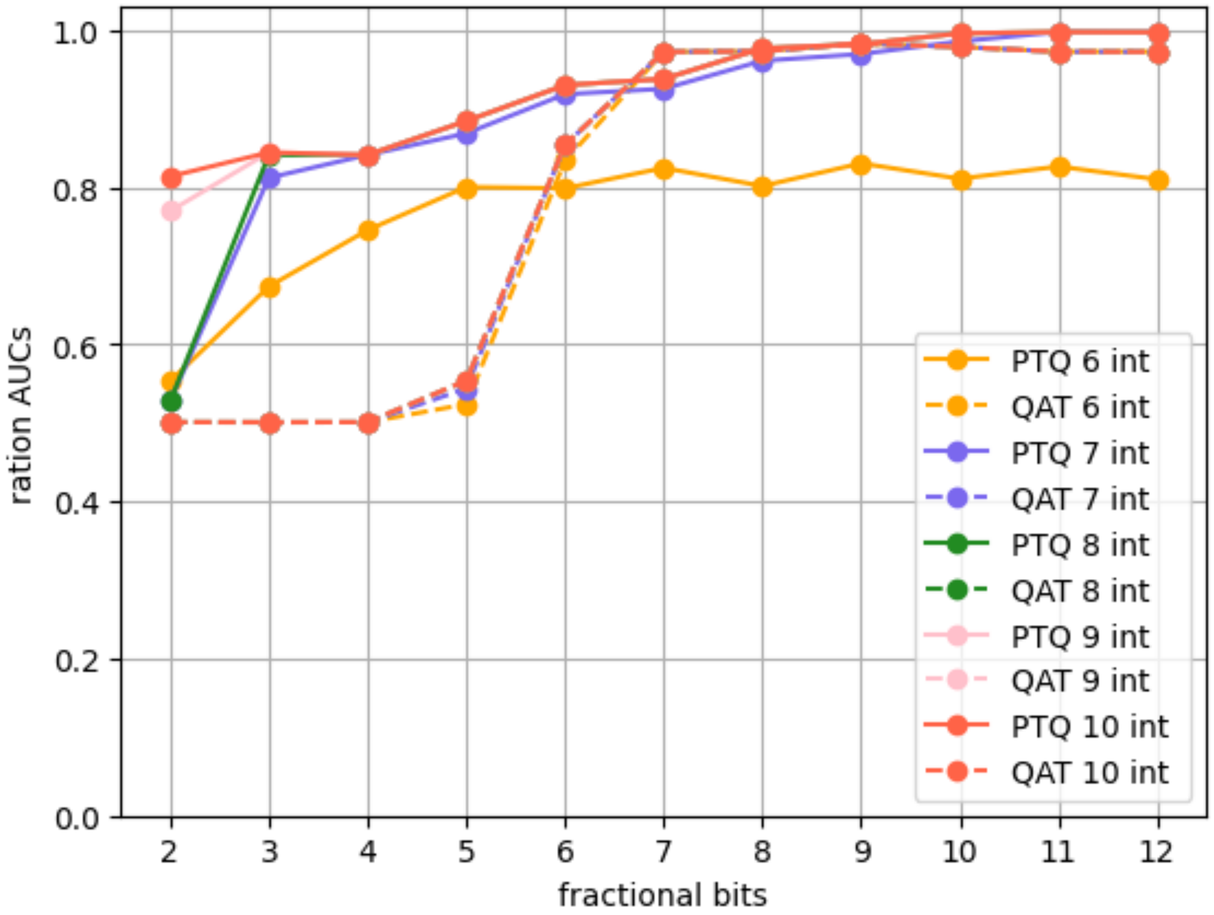
Quantization-Aware Training

- QKeras
- PyTorch (limited options)
- TensorFlow (limited options)
- QONNX (in development)
- Bravitas

HLS Conversion

(hls4ml)

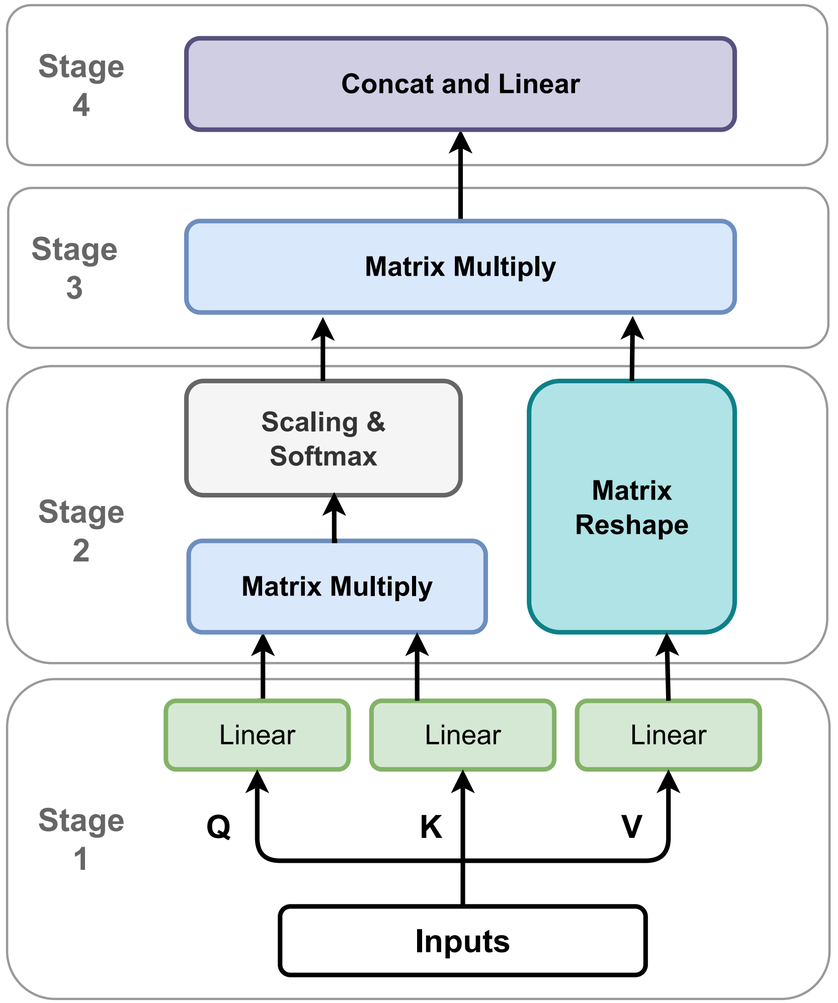
AUC after HLS conversion



PTQ = Post training Quantization
QAT = Quantization-aware Training

Implementation Details

- Pipelining the MHA into 4 stages
- Using FIFOs for the data stream across stages to save the FF usage
- Enable parallel processing for multiple heads
- Applying reuse factor to optimized the resource usage for DSP, and other resources
- Using an optimized softMax layer in stage 3

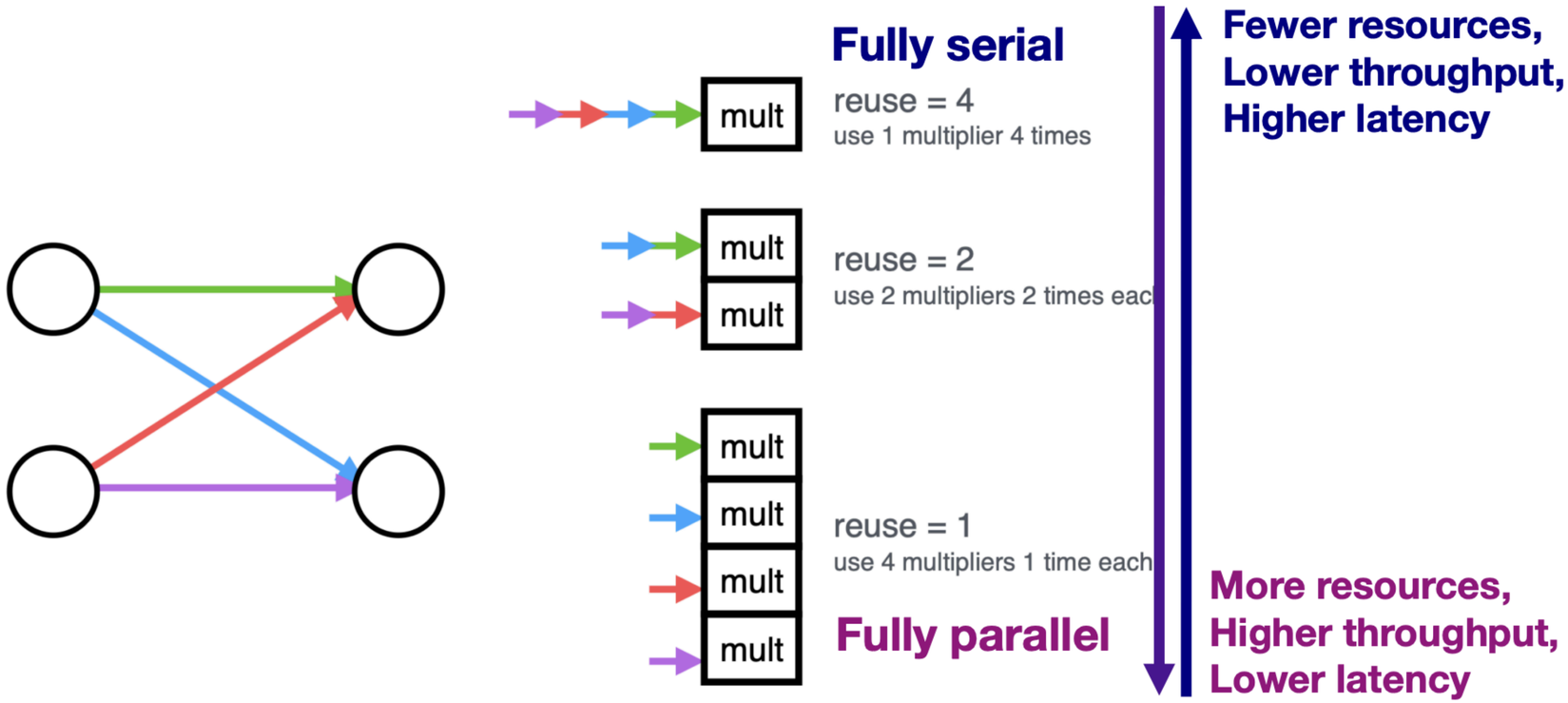


Single-head attention

$$O_h = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Parallelization

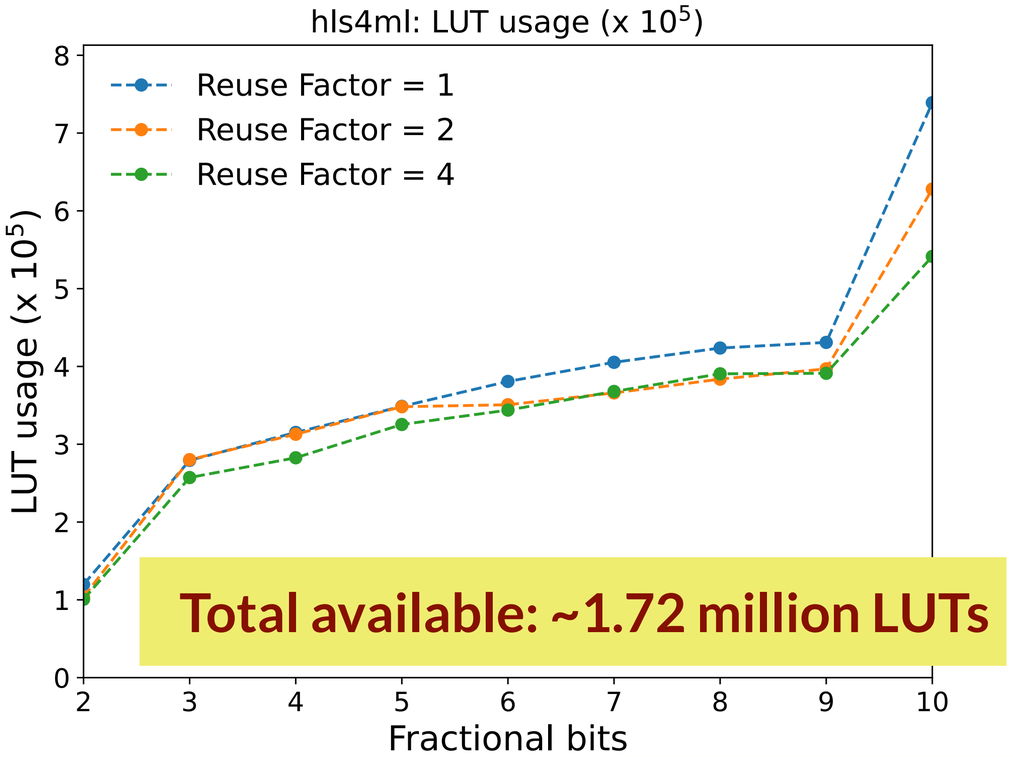
- Trade-off between latency and FPGA resource usage determined by the parallelization of the calculations in each layer
- Configure the "reuse factor" = number of times a multiplier is used to do a computation

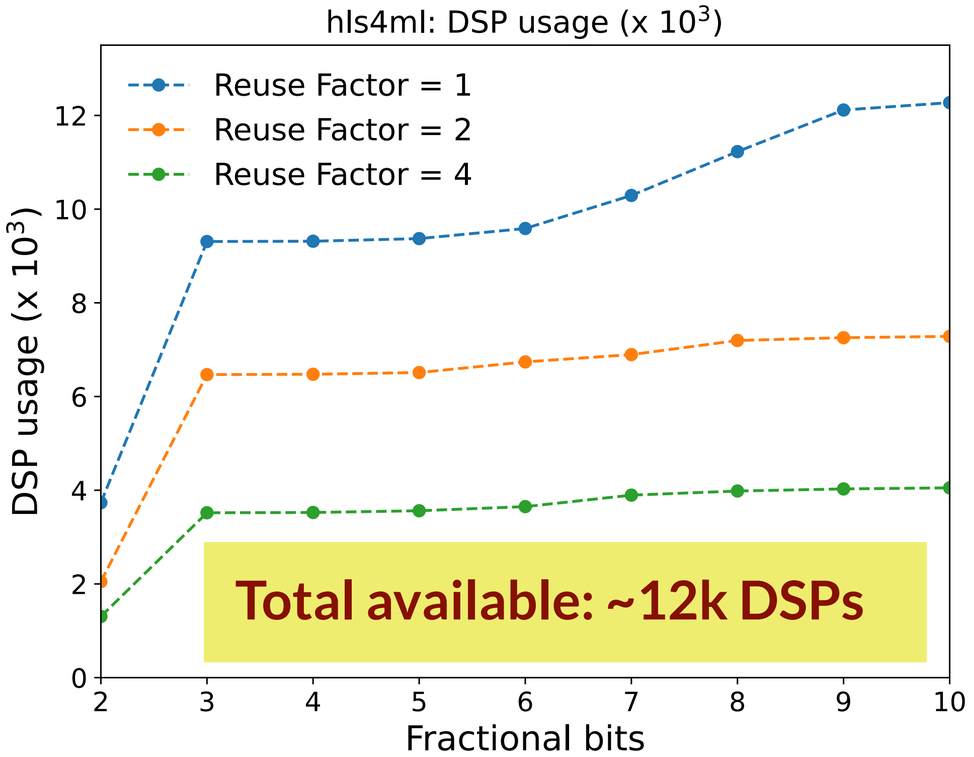


HLS synthesis: DSP and LUT usage

- **DSP usage** as a function of **Total bit width** after HLS synthesis
- The Jumps correspond to DSP input width

Synthesized using Xilinx Kintex UltraScale
FPGA part: xcvu13p-fhga2104-2L-e





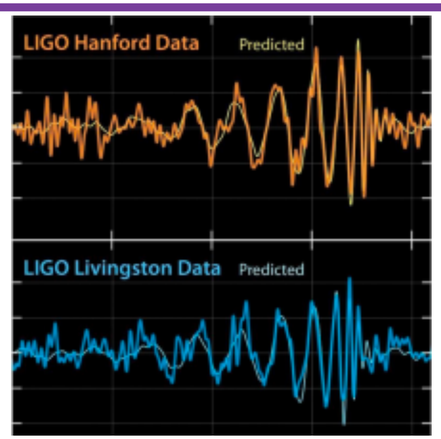
Observed Inference Latency ~ 2-10 μs

Latency

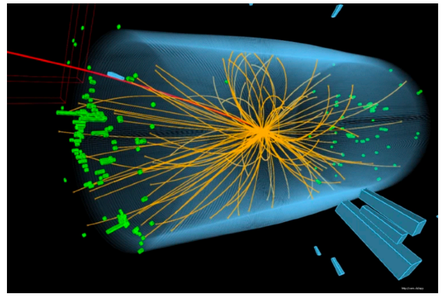
Observed Inference Latency ~ 2-6 μs

Reuse and clk	Interval (cycle)	Latency (cycles)	Latency(time)
R1 (6.577 ns)	49	269	2.077 us
R2 (6.215 ns)	65	449	3.467 us
R4 (4.723 ns)	100	768	5.853 us

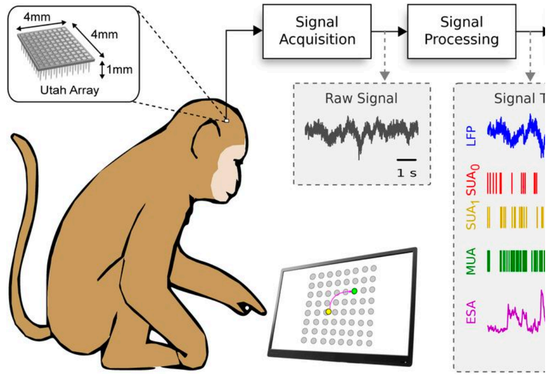
Transformer Models: beyond particle physics



Gravitational Wave Data

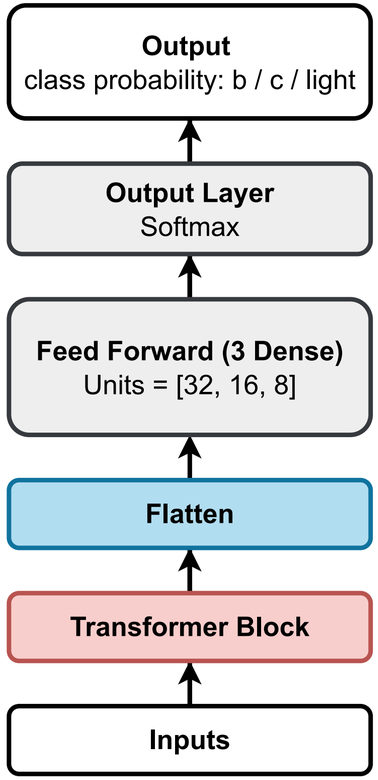


Particle Physics Data



Neural Spike Data

Transformer Models



x 3

Real-time Inference on Hardware

Custom Hardware

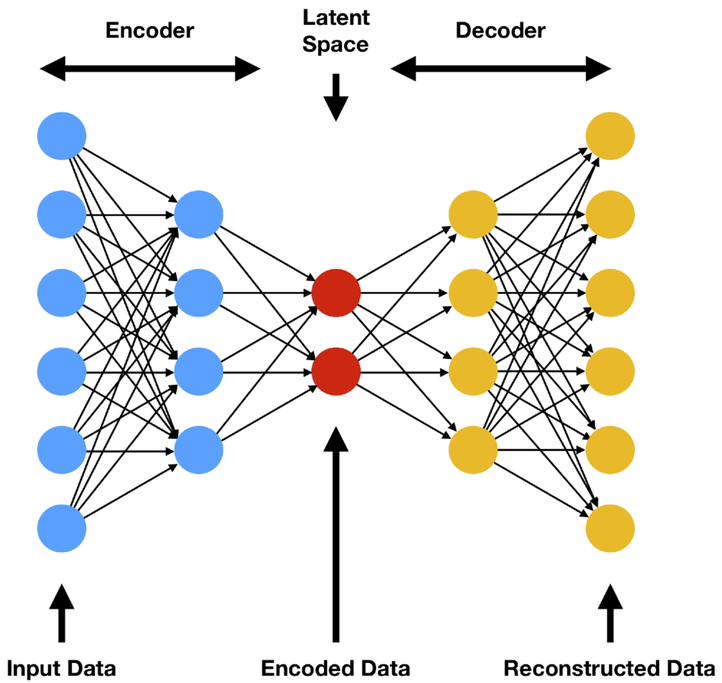
Custom Chip: ASIC

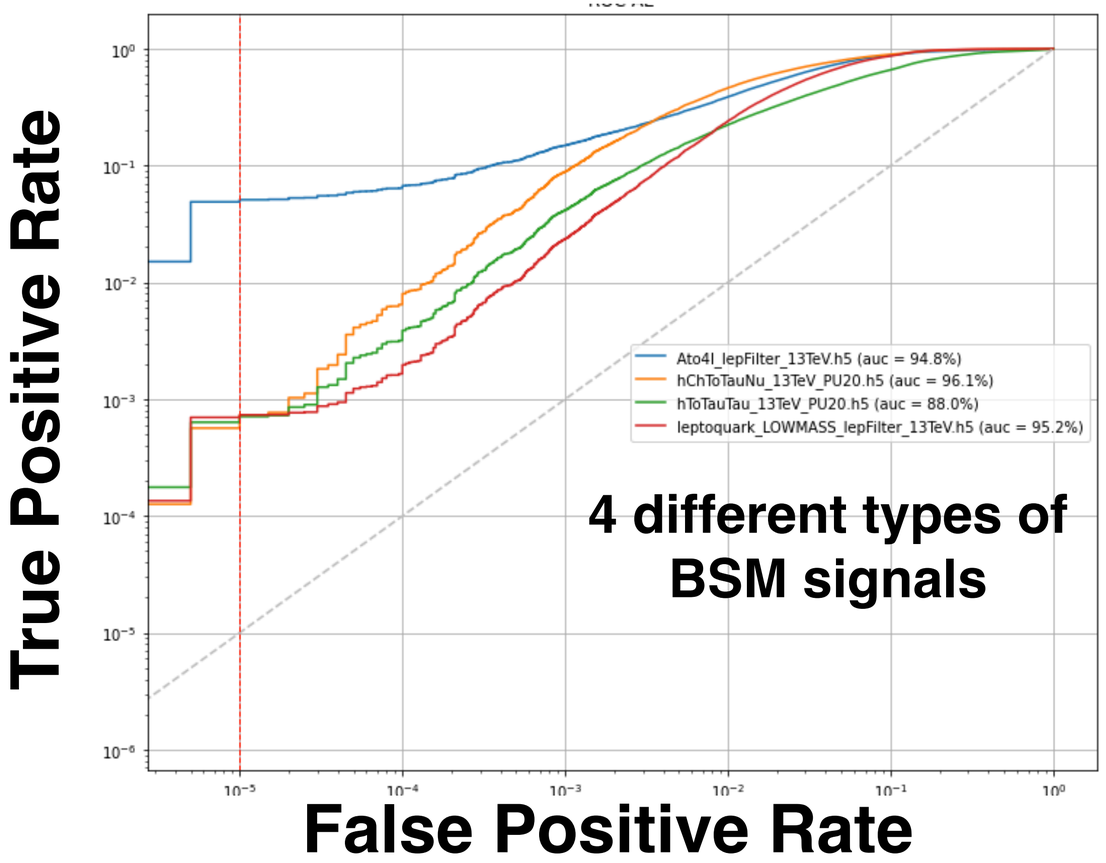
FPGA

One Possible Method: Autoencoder

Autoencoders or Variational Autoencoders

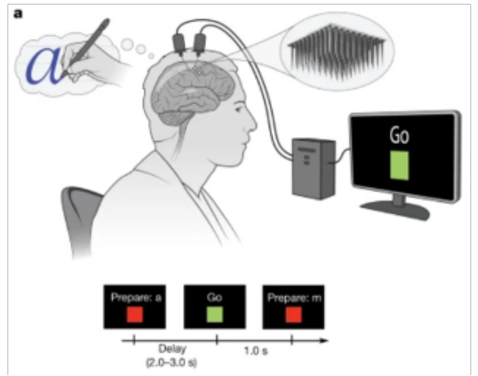
- Reconstruction loss between input and output could be used as anomaly score
- CMS is already using this idea in their Run-3 trigger



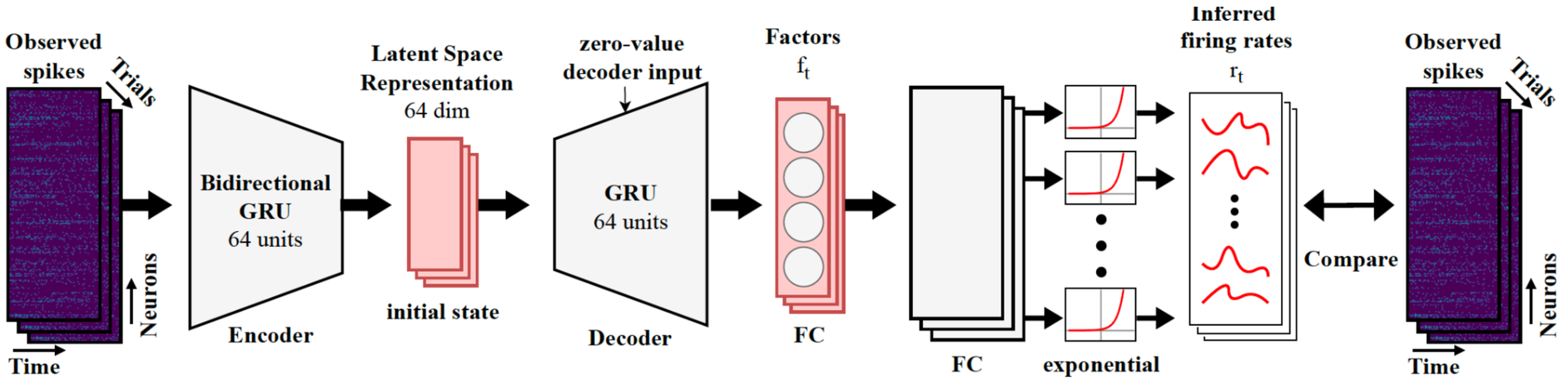


Latent Factor Analysis via Dynamic Systems (LFADS)

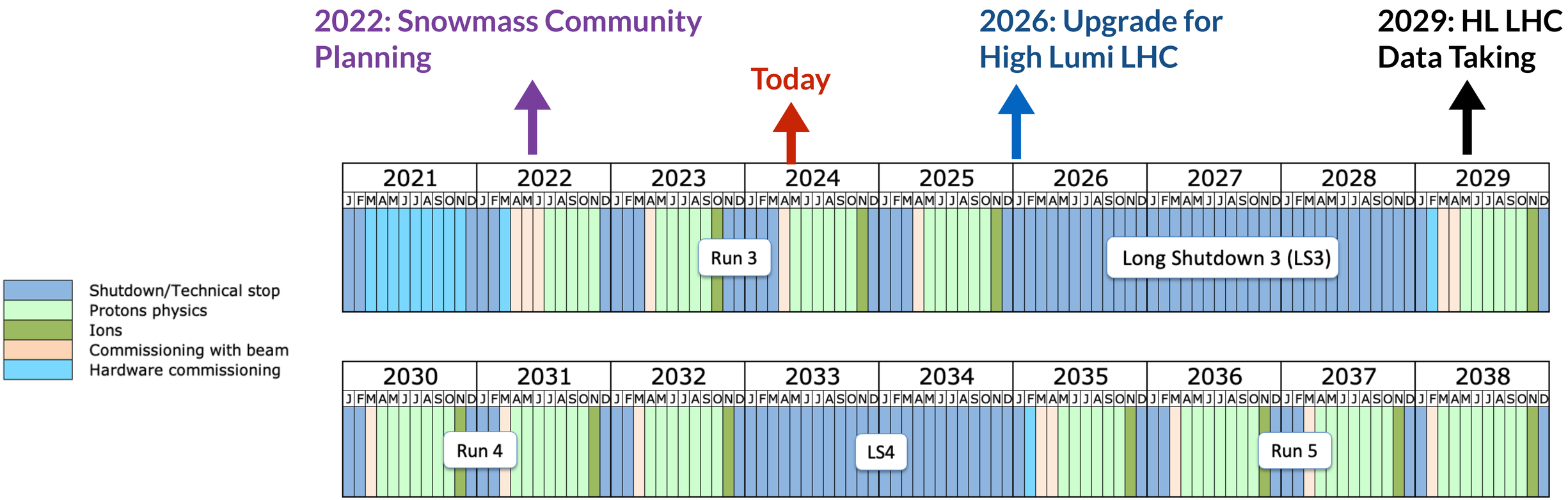
LFADS is a sequential model based on VAE



- LFADS assumes the observed spikes are samples from a Poisson process with firing rates
- Decoder learns the firing rates a function of time
- Training objective: Decoder is trained to infer a reduced set of latent dynamic factors



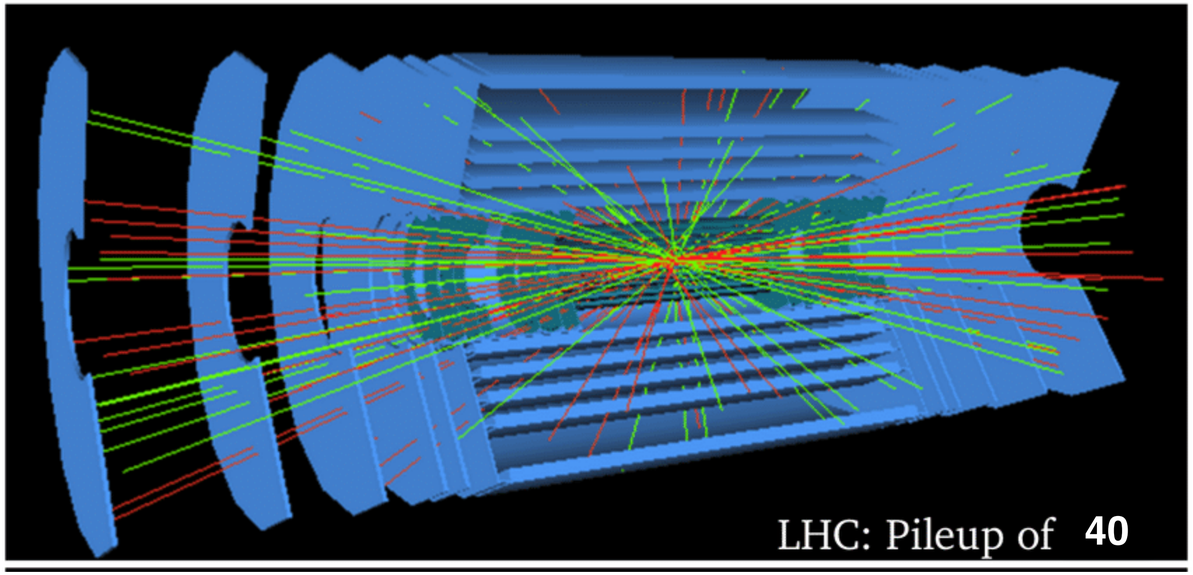
Future of the LHC



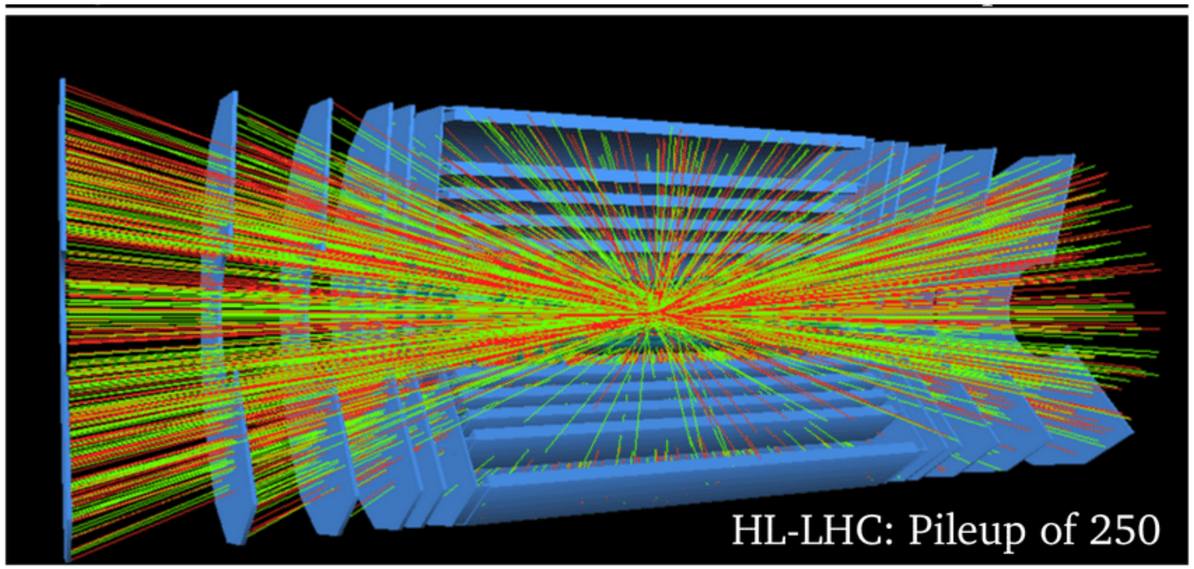
- High Luminosity LHC (HL-LHC) in ~2029:
  - 4 times the current data taking rate

4x more collision rate

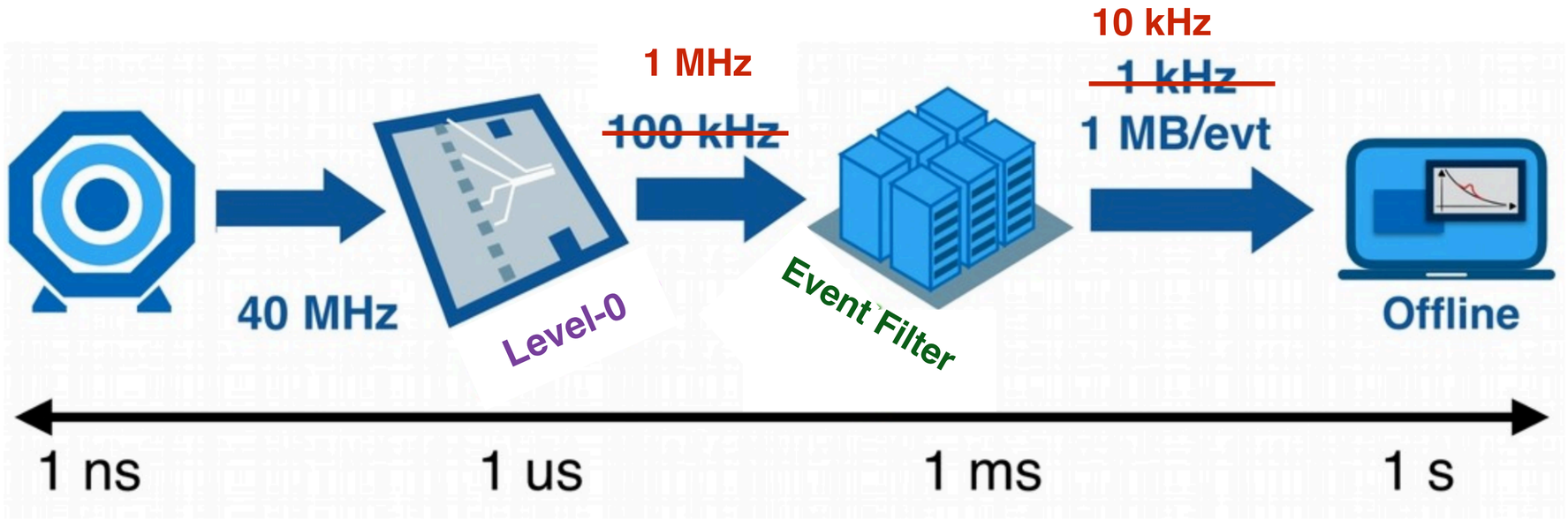
LHC



HL-LHC



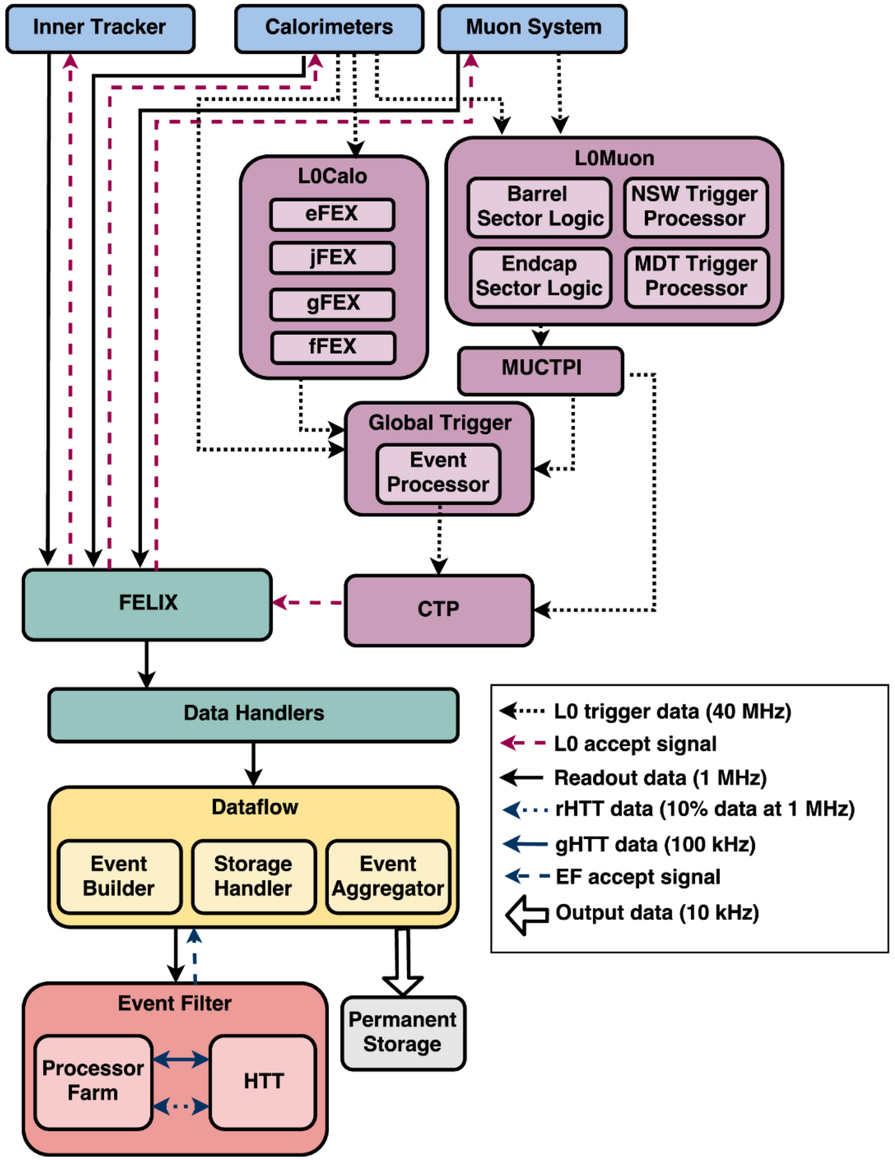
ATLAS HL-LHC Data Processing: Online



ATLAS detector upgrade:
Many subsystems will be upgraded to be compatible with high occupancy / trigger rates

Upgrade in Detector Readout
- *10x faster data collection*
- Better hardware in Level-0 and EventFilter

Phase II ATLAS Trigger Overview



Online Jet Energy Calibration in ATLAS

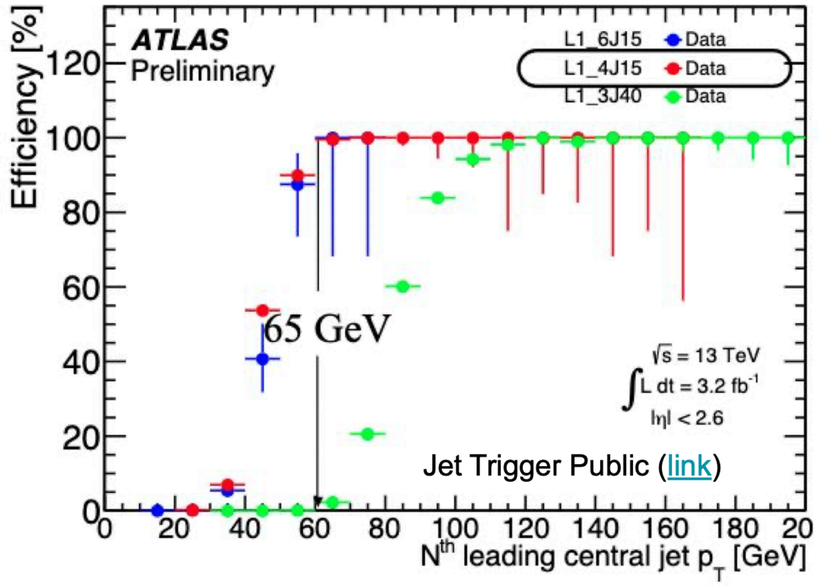
- BDT based regression model to predict the truth energy
- Could improve several physics analysis like di-Higgs

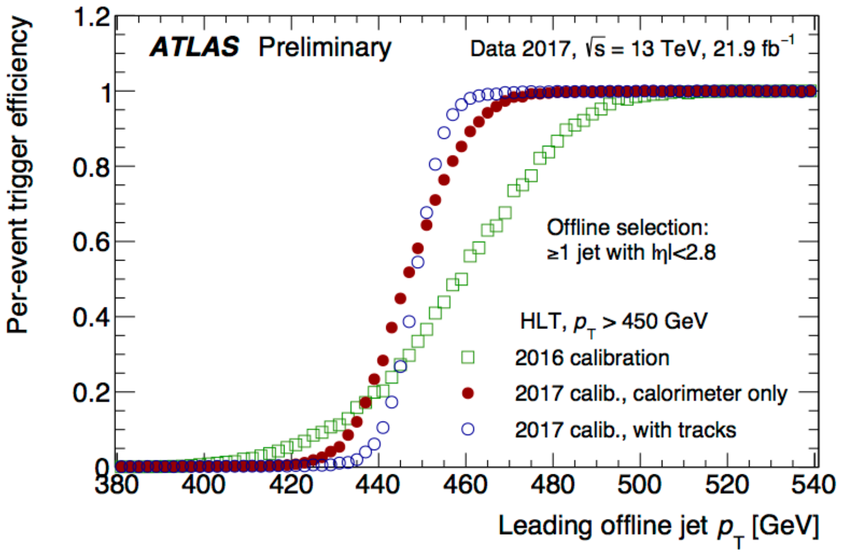
Work-in-progress

- Preliminary BDT model is synthesized using FwX tool
- 25 ns average Latency
- Minimal resource usage: ~1% LUTs

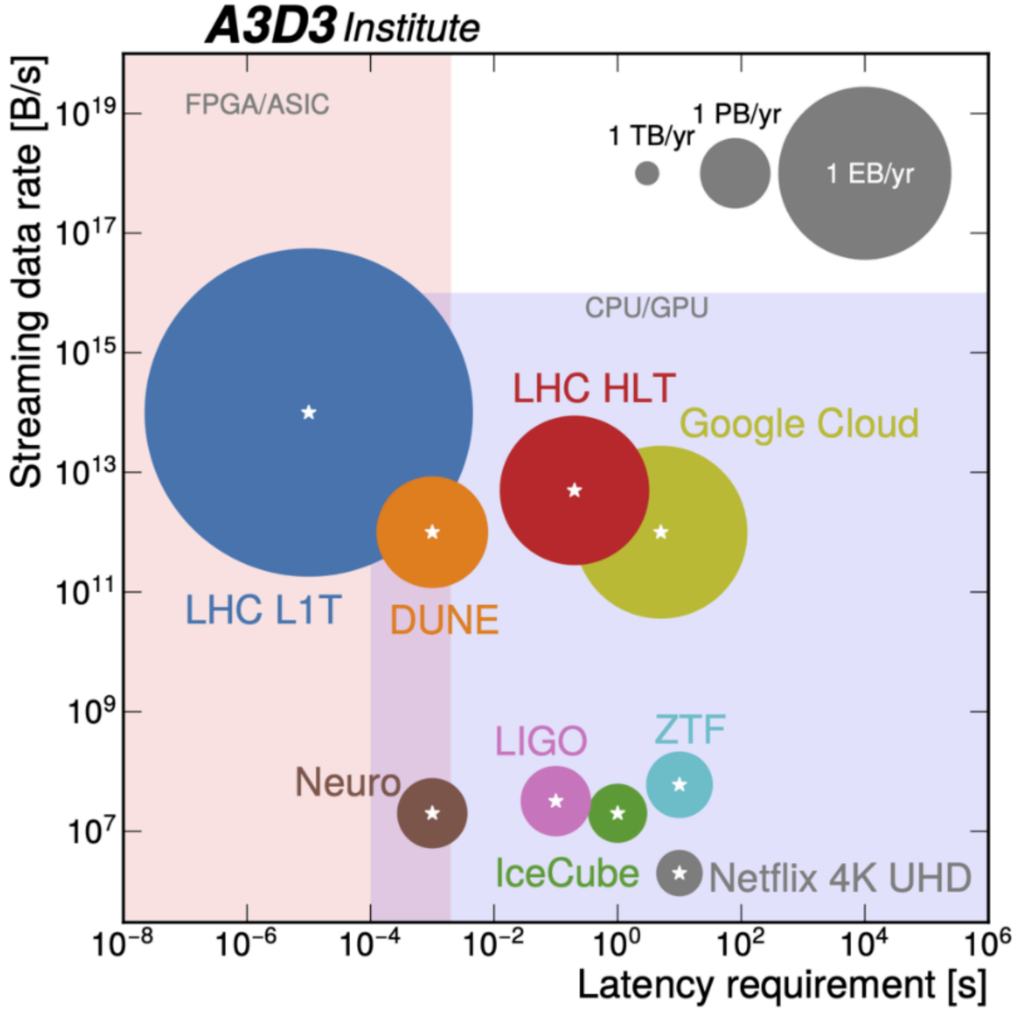
Synthesized using Xilinx Kintex UltraScale

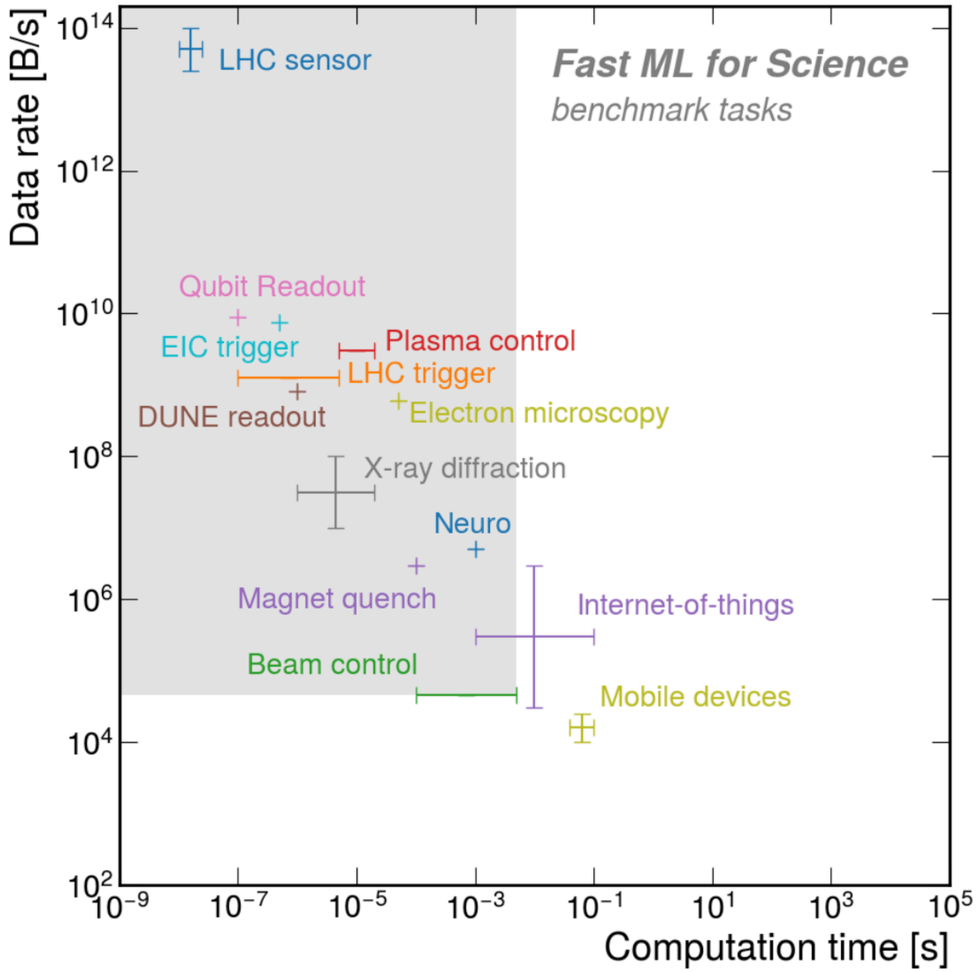
FPGA part: xcvu9p





Beyond High Energy Physics



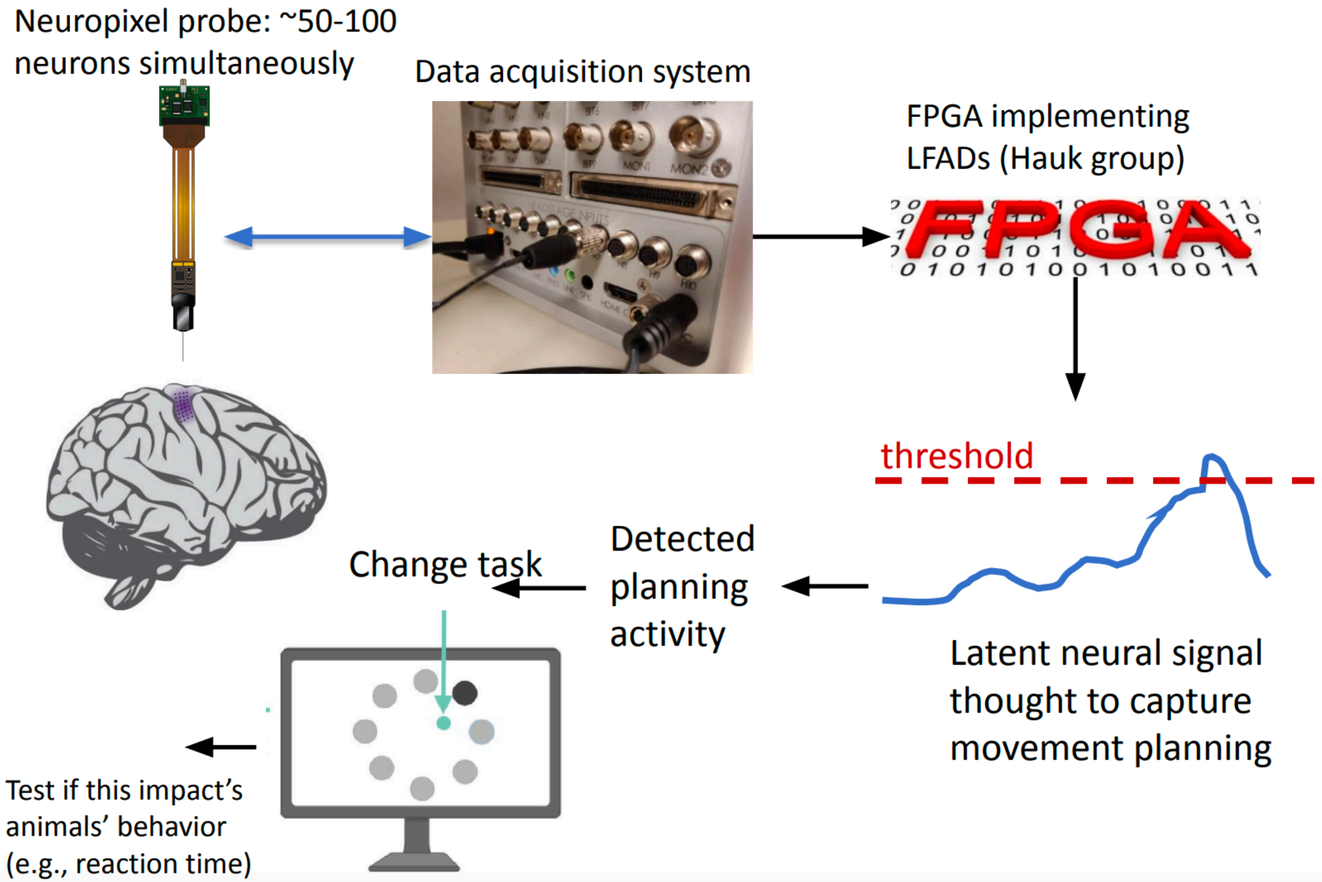


Application: Neuroscience

UW demo: Real time detection of neural states from high-density electrophysiology measurements to drive closed-loop manipulations

Motivation:

- Measure individual neurons
- But computations are performed by groups of neurons
- Auto-encoder models learn latent factors that seem to better capture the "computational units" in the brain
  - LFADs algorithm
- Don't have many tools to compute these in real time,
- Will help us do experiments to test if they are meaningful computational units



Summary

- Increasingly possible to perform low latency inference of ML models
  - Also low-power, high radiation
- Many cases, ML offers improved performance over traditional algorithms
- Fast ML could help enable discovery!
- Applications in many fields, areas

Real-time AI mini-course
at IEEE NPSS 2023 (Vancouver)

Many thanks to my Collaborators!

Thank You!

Extra Slides

Softmax Optimization

General approach:
each SoftMax output Si requires the calculation of the exponent of the difference between zj and zi , summed over all elements, and then inverted

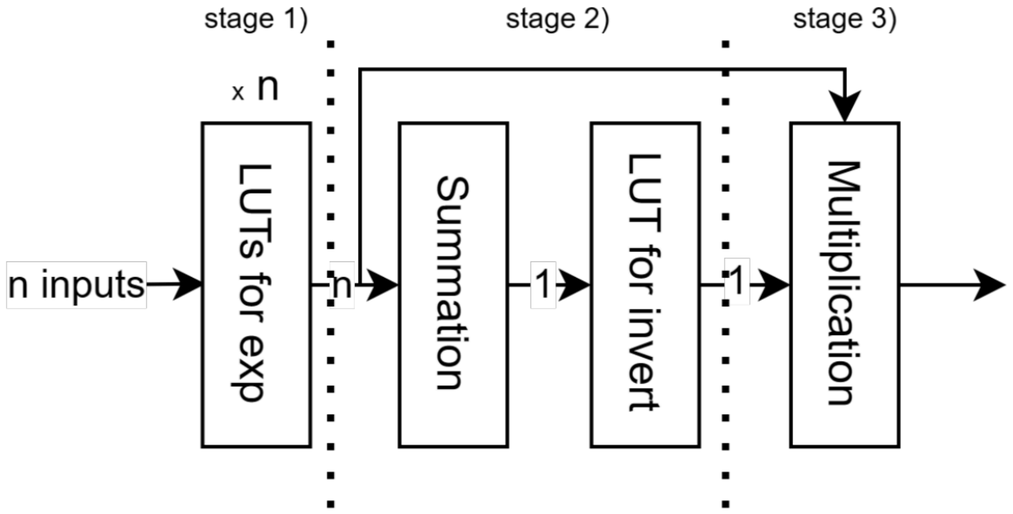
$$S_i = (\sum_{j=1}^{k} (e^{(z_j - z_i)}))^{-1}$$

Modified approach:

Stage 1:
Element-wise exponent computation

Stage 2:
Sum of all the exponents
-> Inverted using inverse look-up table
-> stored in register

Stage 3:
Element-wise multiplication



gFEX: Phase I Trigger upgrade

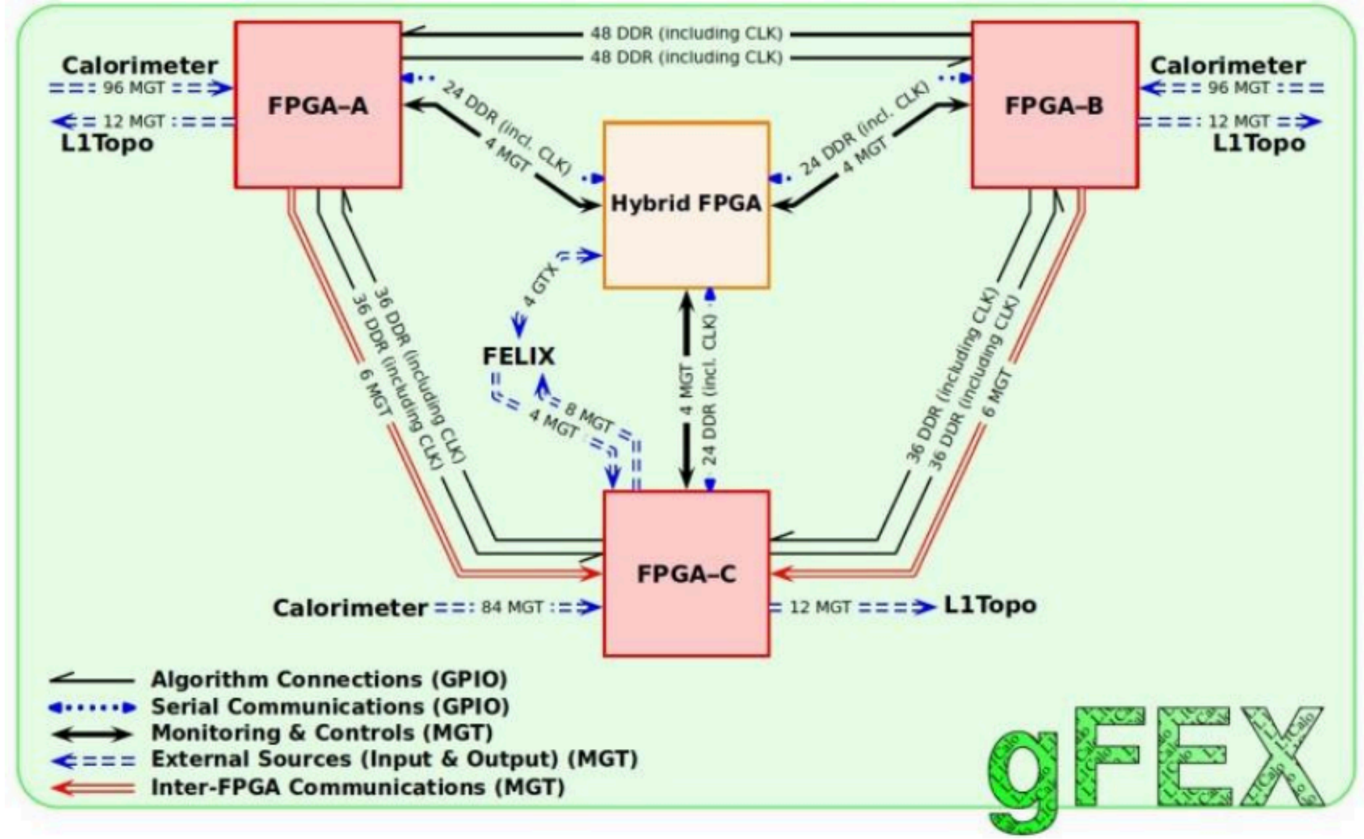
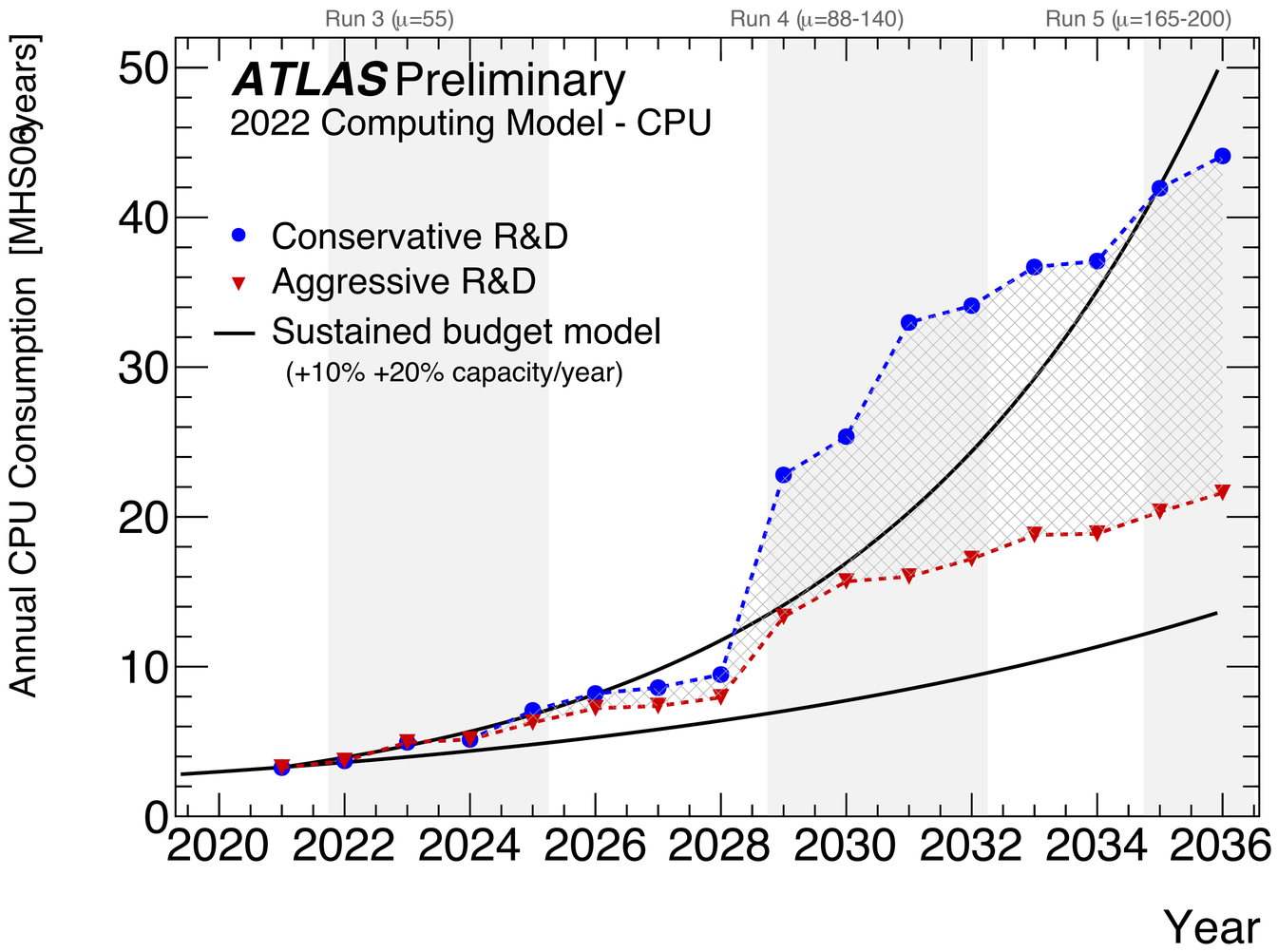


Fig. 4. Block diagram of gFEX. FPGA A, B, and C are the same ultra-scale FPGAs; the Hybrid FPGA is ZYNQ FPGA. The FELIX is a PCIe module designed for ATLAS System.

Computing Chalenges in ATLAS

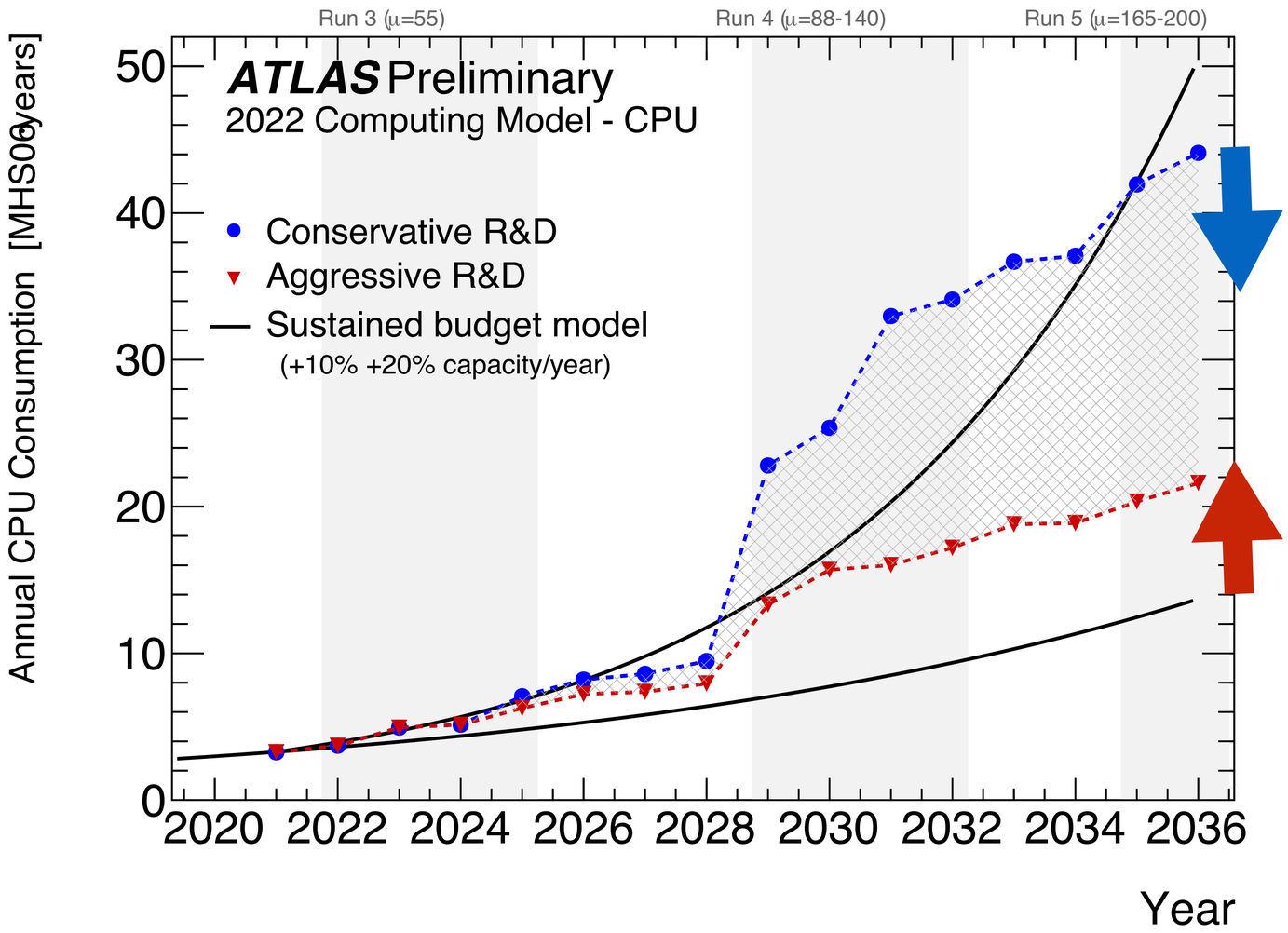


To preserve current physics:

- 4 times the current data taking rate

Lacking sufficient budget to sustain required computing

Closing the Gap



Smarter algorithms: ML

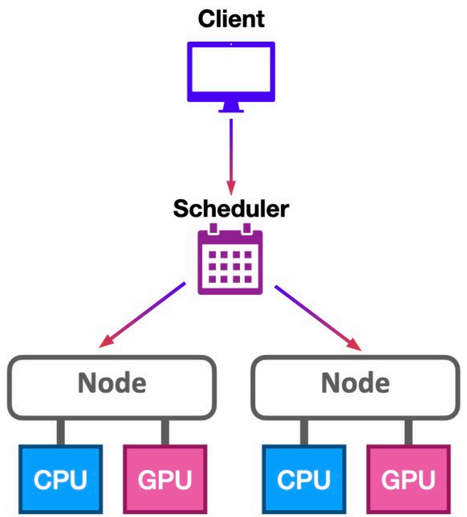
Faster Hardware: FPGAs, GPUs

Heterogeneous Computing Model

- Support for different type of hardware is becoming important

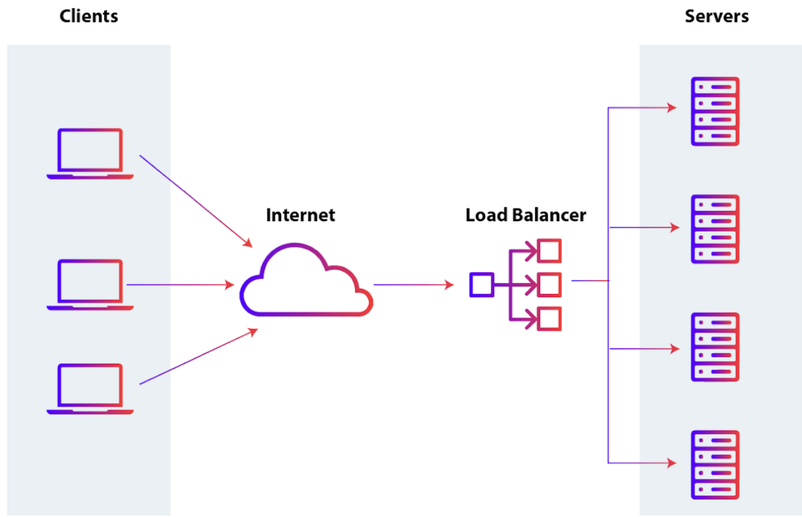
Direct Connection

CPUs and GPUs are connected



As a Service

No need to have a local GPU



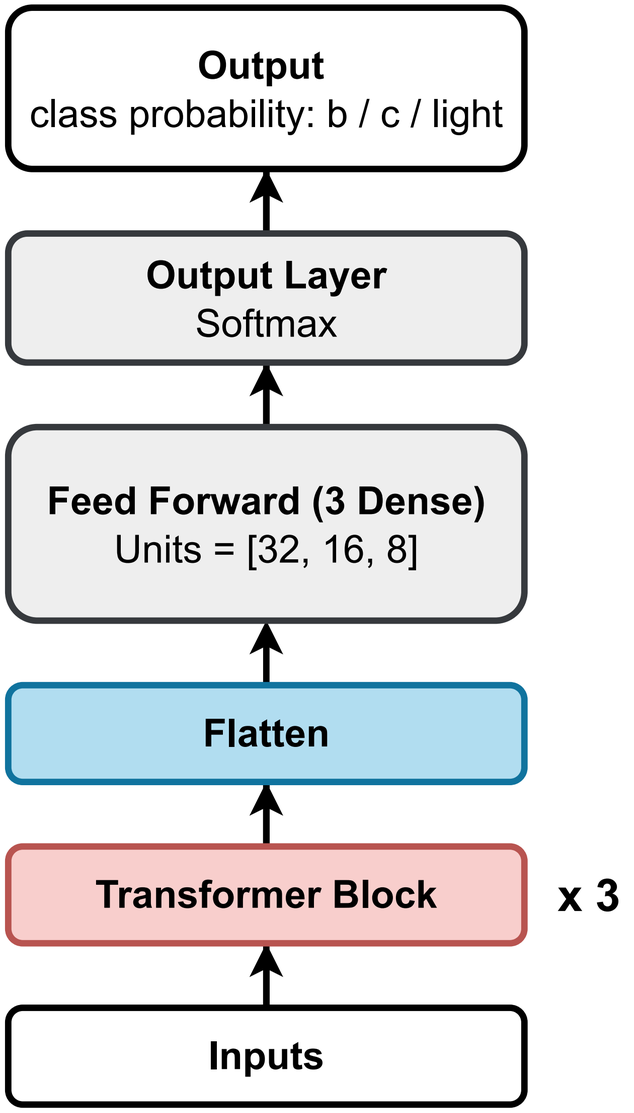
- Simple support for mixed hardware
- Scaleable
- Throughput optimization for multiple-core

Summary: Transformers

Three benchmark cases

1. Binary classifier: ~3.3k parameters
   - Car engine anomaly detection with Ford time series data
2. 3-class classifier: ~10k parameters
   - particle physics application
3. 4-class classifier: ~3.3k parameters
   - LIGO gravitational wave detection application

Observed Inference Latency ~ 2-10 μs

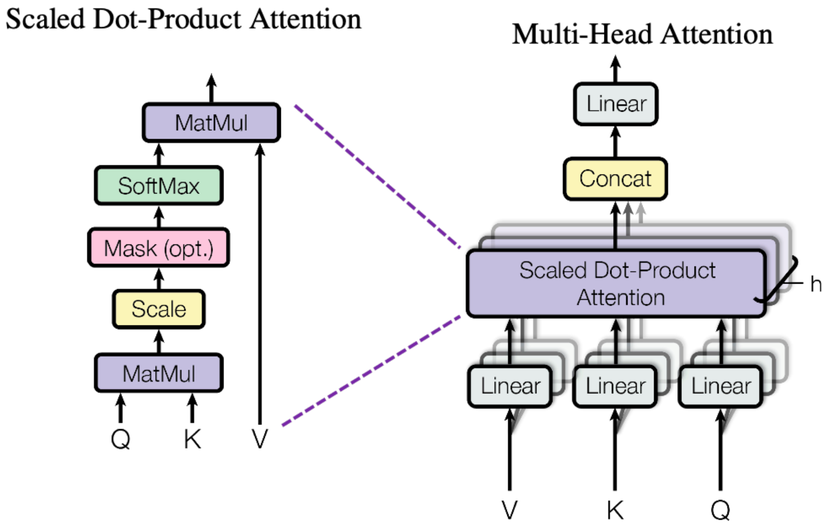


Other Future applications:

Modeling Neural population Dynamics with Transformer-based architecture

What's Next? Transformers?

- Good for long sequences
- Potentially more useful for Gravitation Wave applications
  - Using on LIGO signal - background classification model

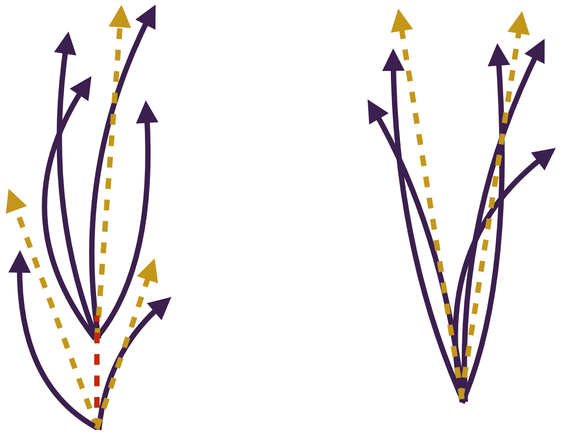


Involves a lot of large matrix multiplication

B-tagging Model

3-class classifier: ~10k parameters

Classify b / c / light jets



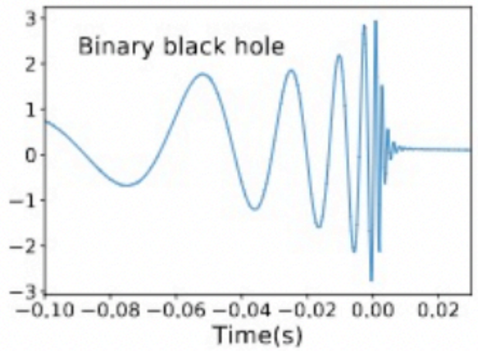
b/c

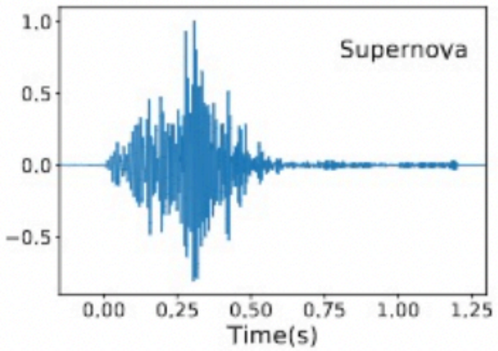
light

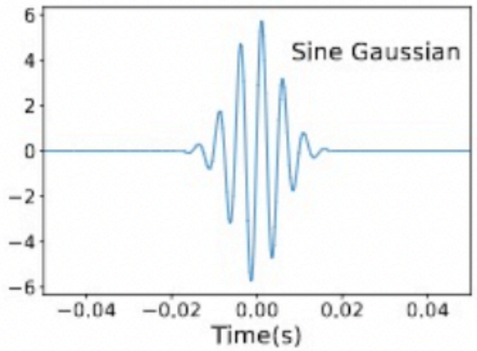
- Implemented the on an FPGA
- Successfully synthesize
- Latency of 2-3 μs

Other Applications: LIGO and Neuroscience

LIGO Gravitational Wave Experiment



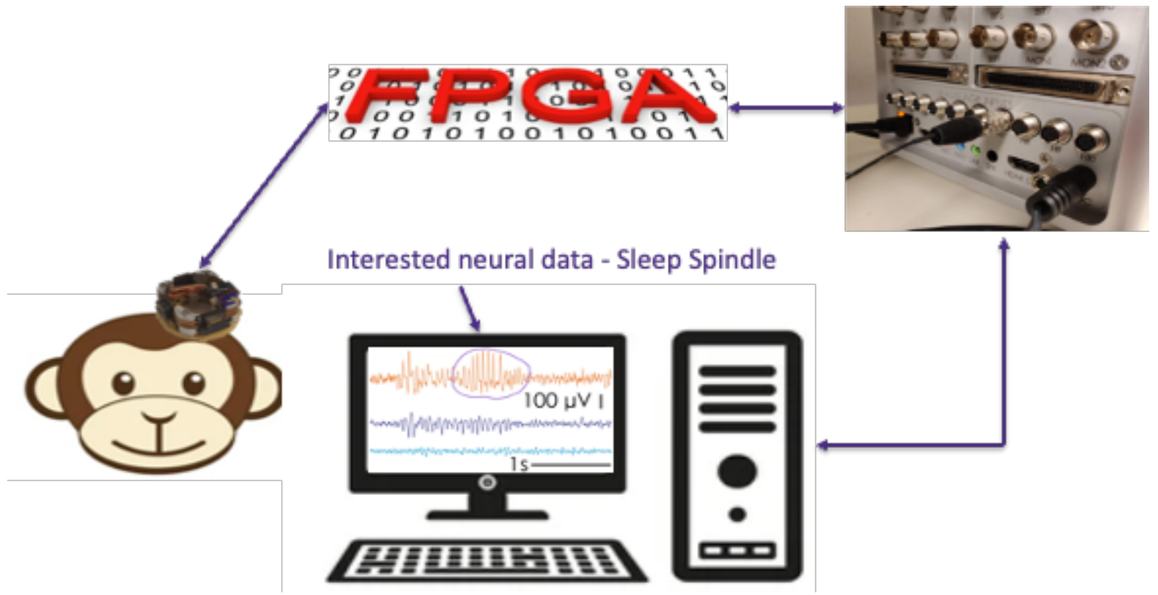




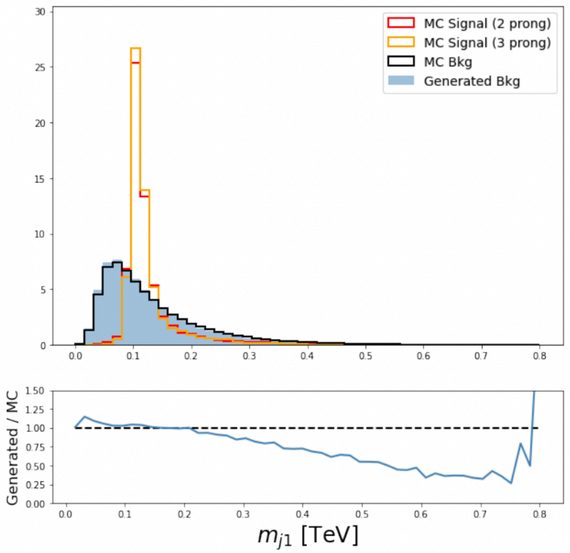
Classification:
Signal, Glitch, Background

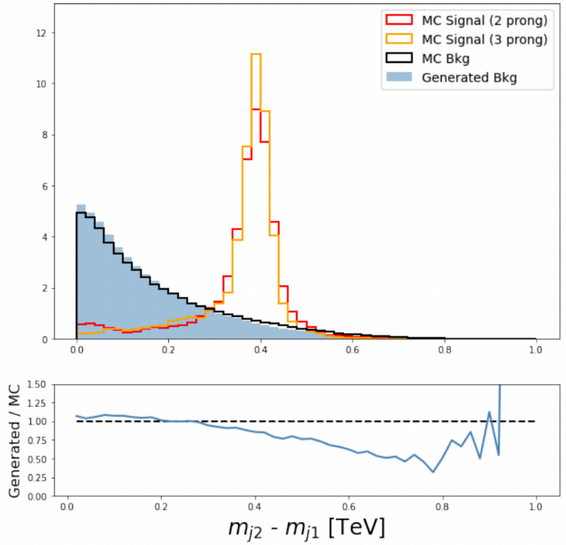
Neuroscience: Detecting brain activities

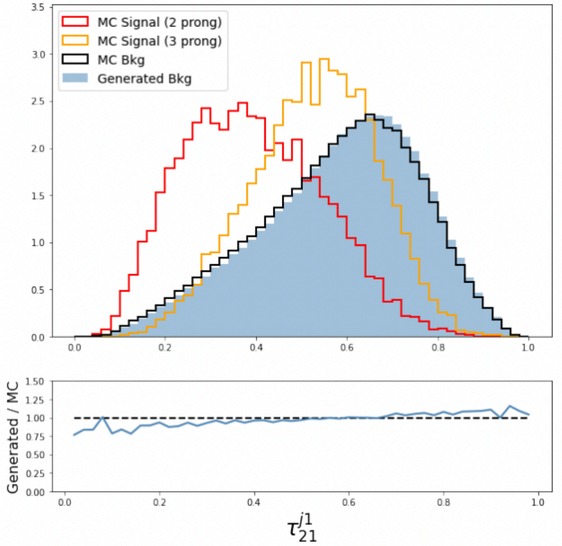
Real time inference based on neuron activities

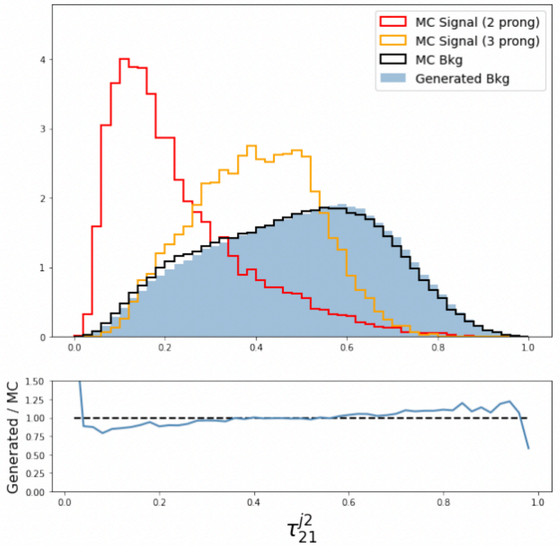


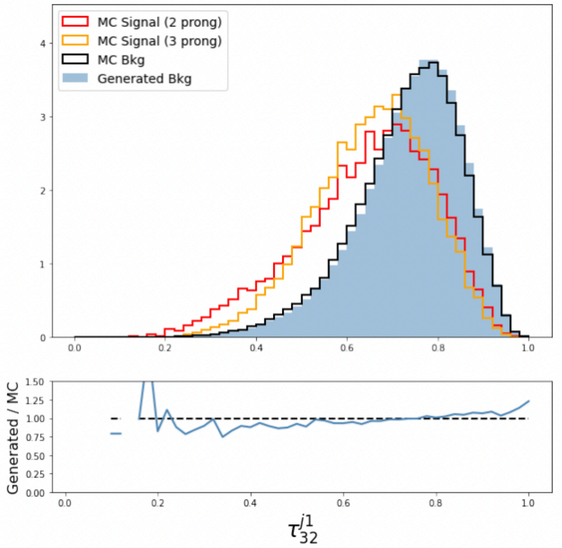
Generated Events Signal Region

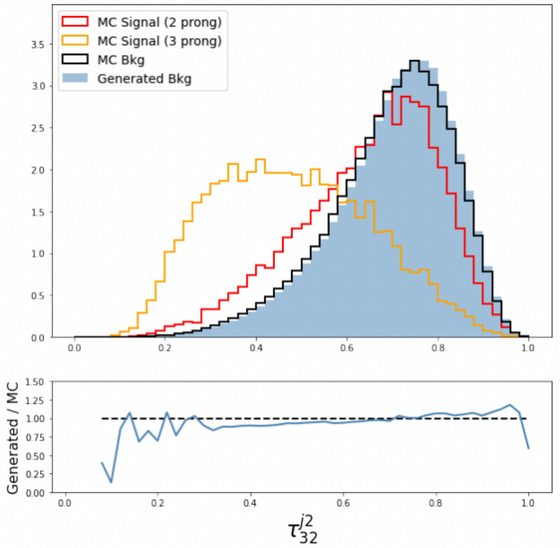




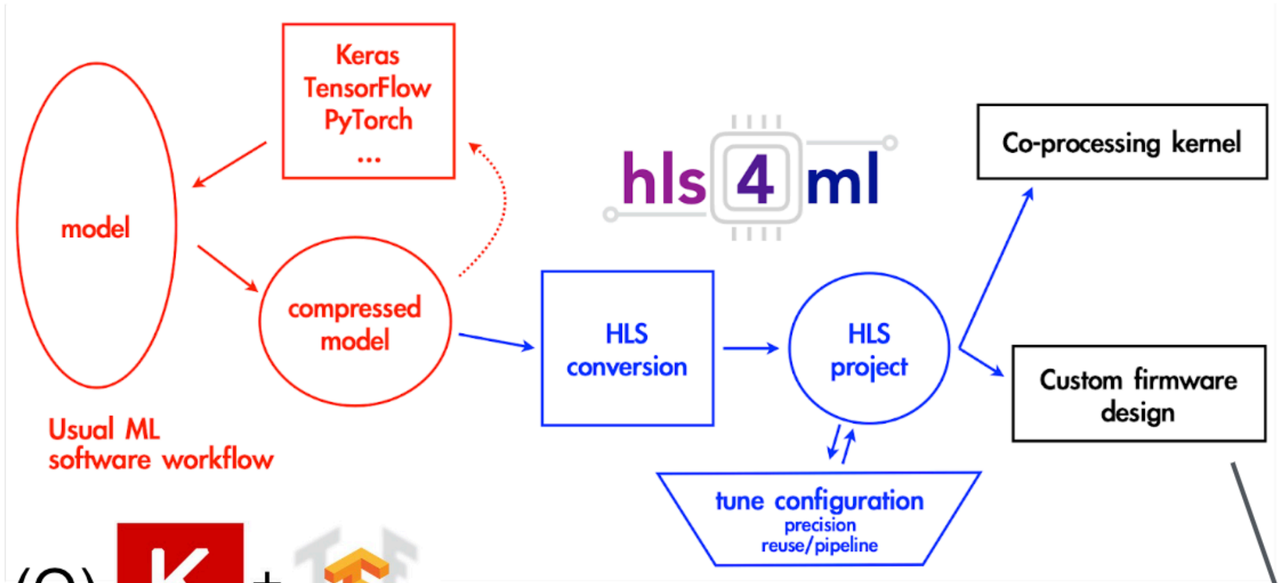


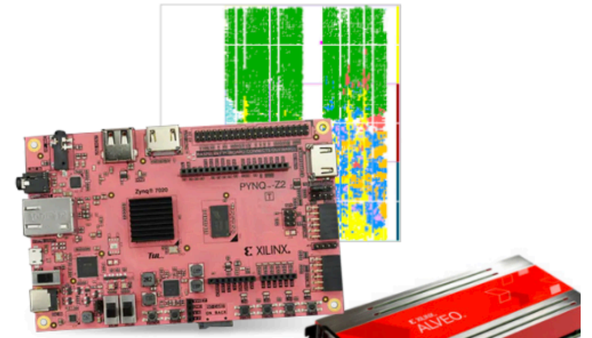




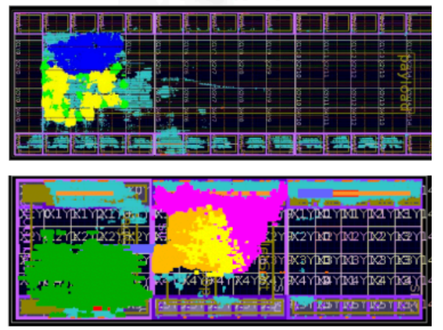


High Level Synthesis with Machine Learning (hls4ml)



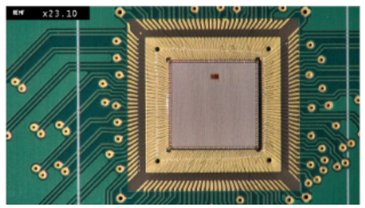


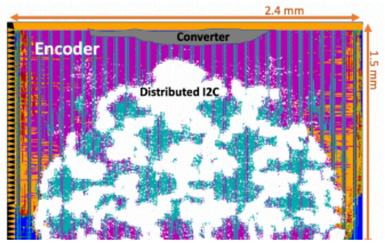
VivadoAccelerator
for Xilinx accelerators / SoCs



Custom design
(talk 1, 2)

ASICs (talk 1, 2)





(Q) K + TensorFlow

(Q) ONNX

PYTORCH

Vivado™ HLS

intel HLS COMPILER

Mentor®
Catapult
Coming soon

Many Other Tools

Neural Networks

arXiv: 2004.03021

SLAC Neural Network Library (SNL)

arXiv: 2305.19455

Boosted Decision Trees (BDTs)

**** ATLAS is using both***

arXiv: 2002.02534

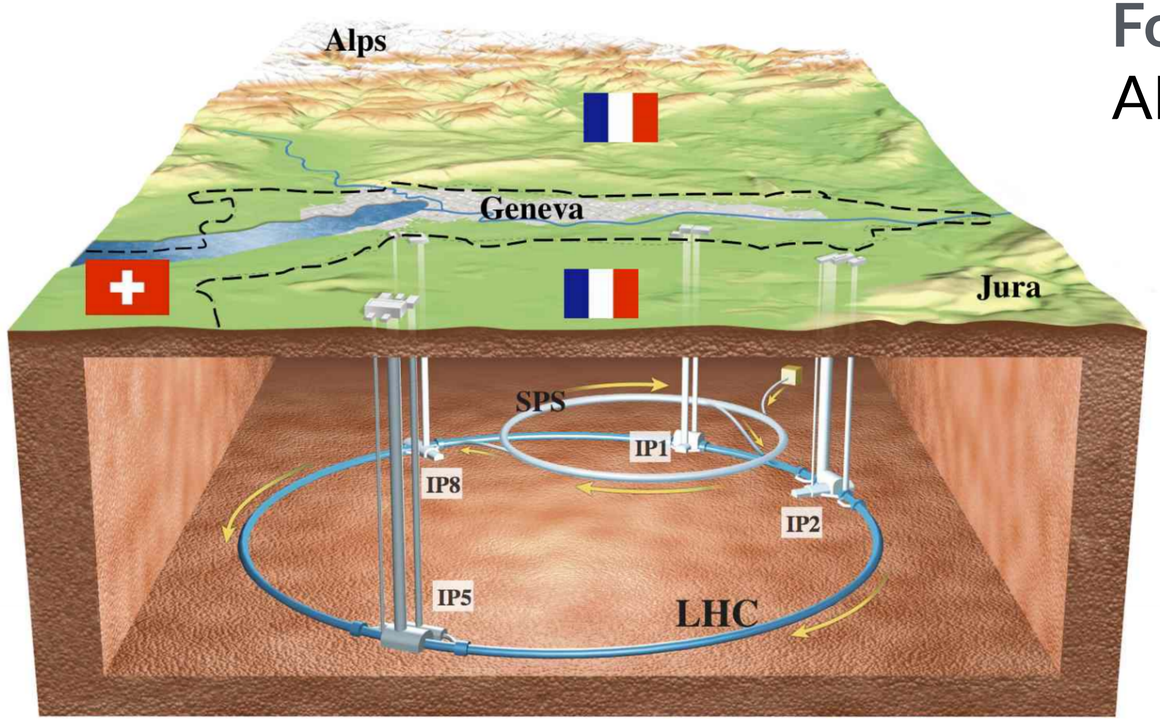
arXiv: 2104.03408

This is not a exhaustive list

Mostly from Physics

Large Hadron Collider at CERN

- World's largest and most powerful particle collider
- Collides protons (most of the time) bunches (~10^{11} protons in a bunch) spaced by 25 ns



Four major experiments on the LHC ring:
ALICE, **ATLAS**, CMS, LHCb

centre of mass energy

2011: 7 TeV
2012: 8 TeV
2015 - 18 : 13 TeV
2022 - 25 : 13.6 TeV
2028 - : High Lumi LHC

In this talk

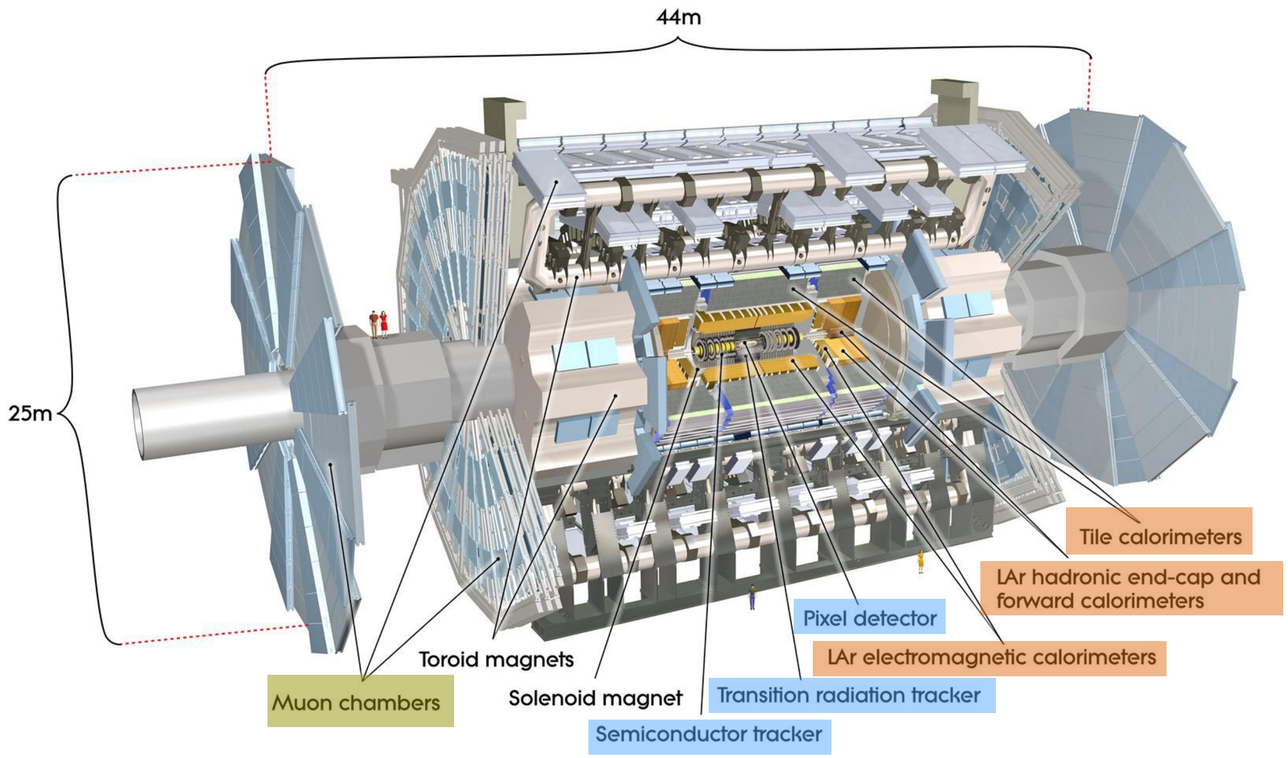
The ATLAS Experiment

General purpose detector

Muon Spectrometer:
Four different detector technology

Calorimeter:
Electromagnetic (Liquid Argon), Hadronic (Liquid Argon (endcap) & Tile (barrel))

Solenoid Magnet: 2.0 T



Inner Detector:
Three different detector technology

1. Silicon Pixel
2. Silicon Strip
3. Straw Tubes: Transition Radiation Tracker (TRT)

Particle Reconstruction

Particles are reconstructed combining signatures from different sub-detectors

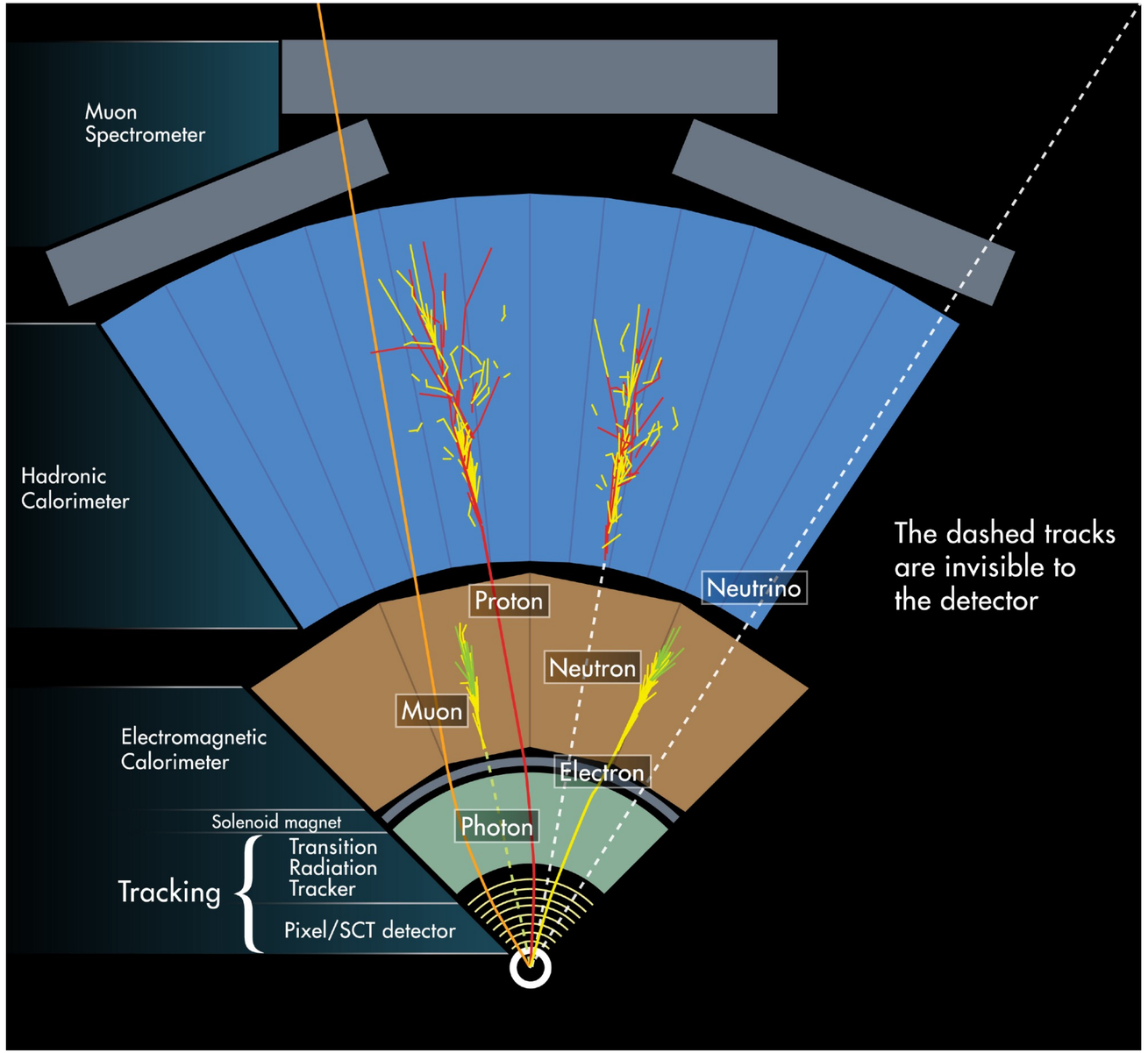
Electrons: Inner detector track + calorimeter deposit

Photons: Calorimeter deposit

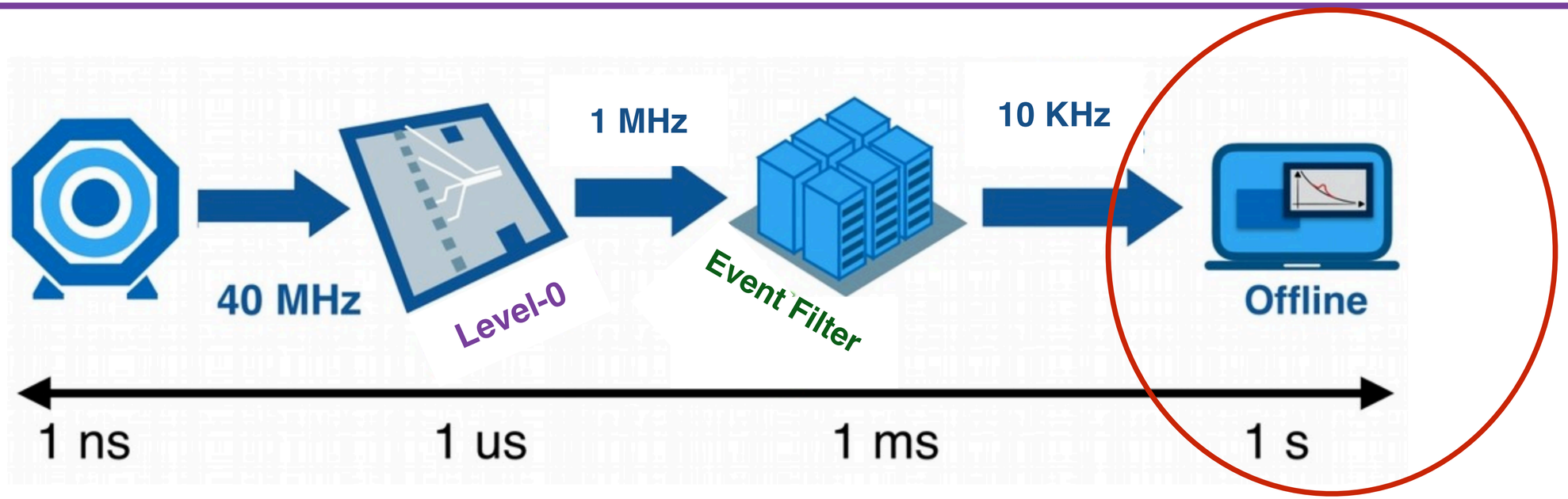
Jets: Inner detector tracks (charged), Calorimeters

Muons: Inner detector and Muon spectrometer tracks

Neutrinos: Cannot detect, mostly resolved using missing transverse energy



ATLAS Phase-II Data Processing

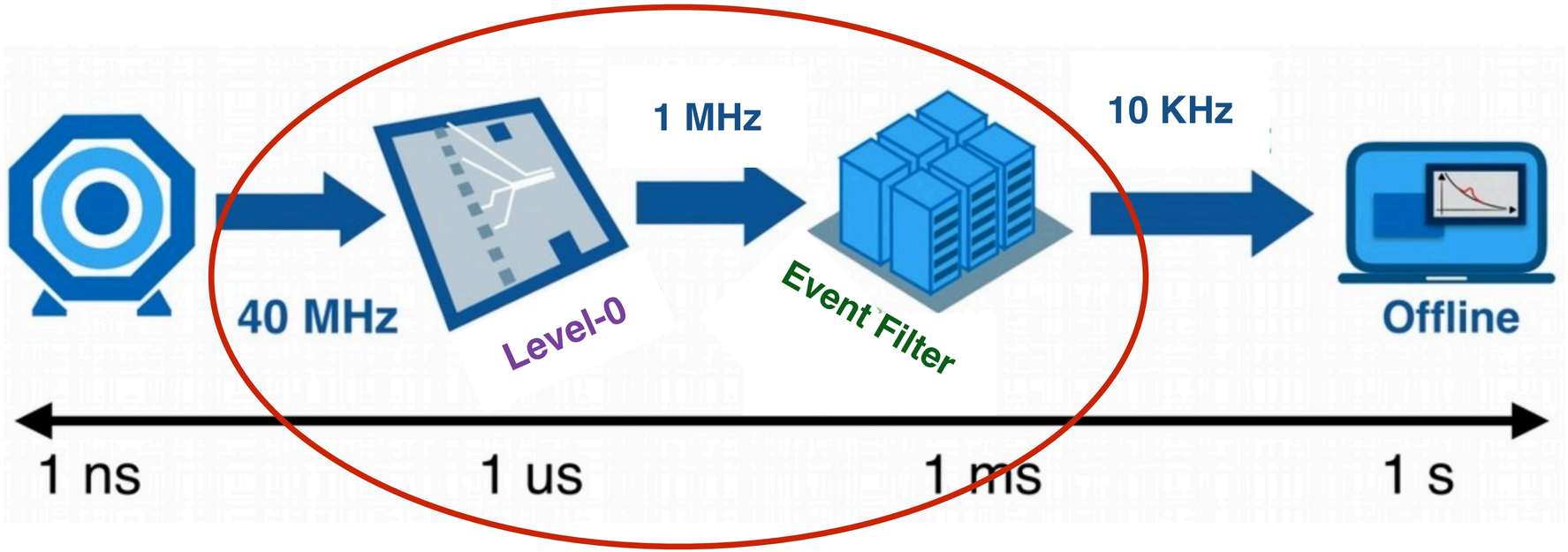


- Usage of ML is growing over time

- Active R&D
  Further improvements driven by more complicated algorithms

Usage of GPUs will be beneficial for the future LHC Runs (after 2026)

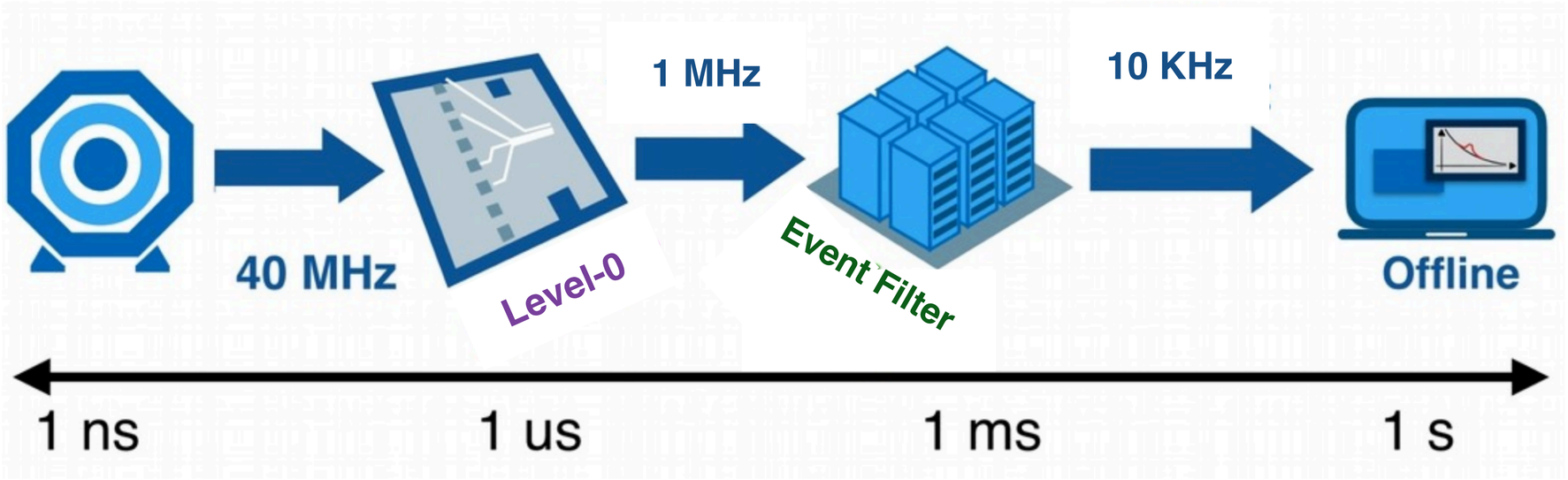
ATLAS Phase-II Data Processing



- ML has potential to improve physics performance in the trigger system
- **Strict latency requirements:** μs (ms) for **Level-0** (**Event Filter**)

  For **Level-0** trigger → we need to run ML on FPGAs

ATLAS Phase-II Data Processing



L0 Trigger (hardware: FPGAs) – *O(μs) hard latency*

- Typically coarse selections are applied

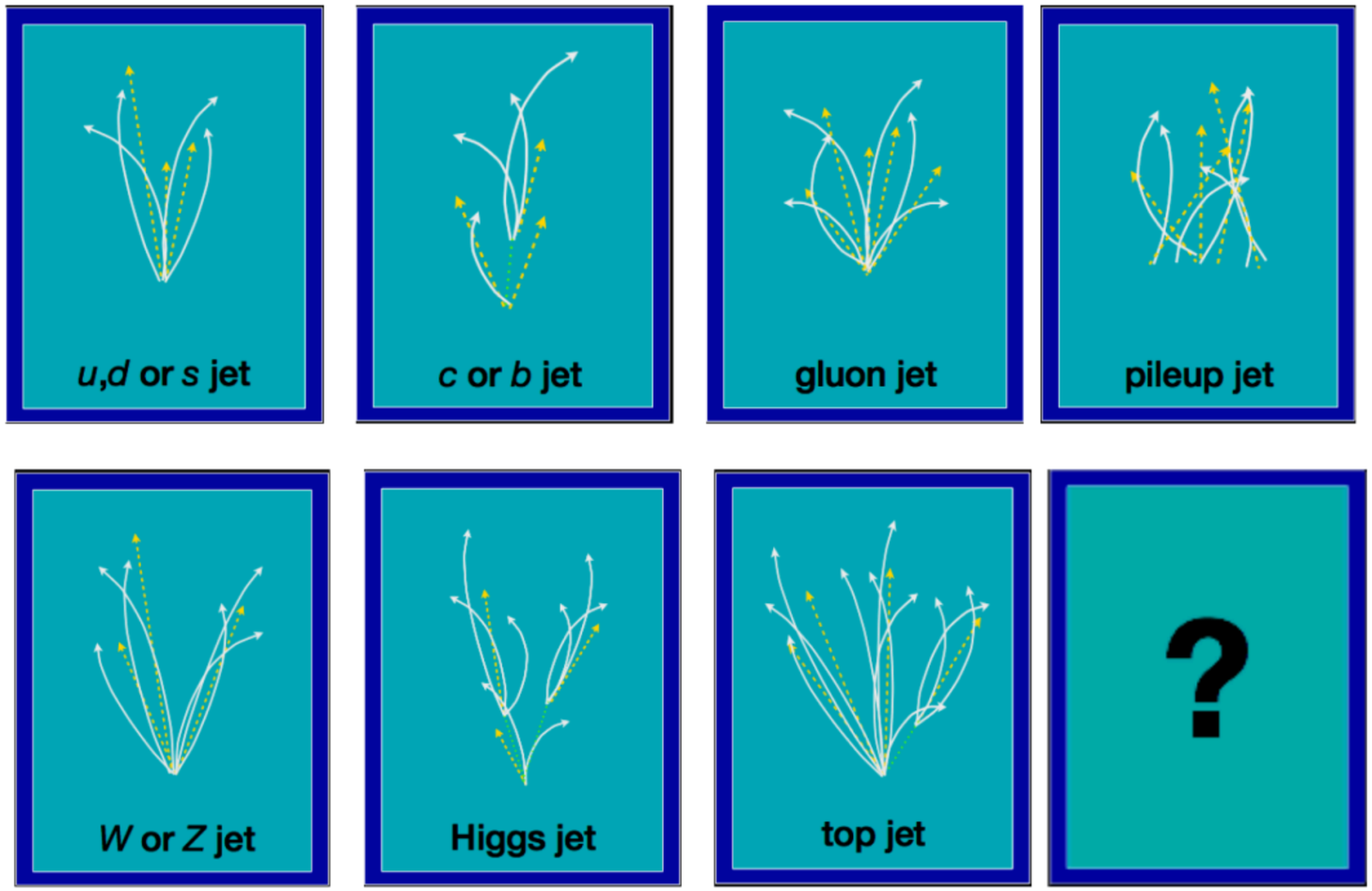
Event Filter (software: CPUs) – *O(100 ms) soft latency*

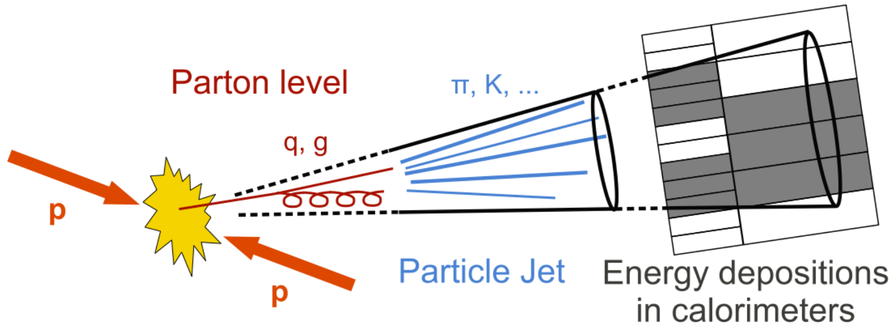
- More complex algorithms (full detector information available), some BDTs and DNNs used

Offline (software: CPUs)

- Full event reconstruction, bulk of machine learning usage in ATLAS/CMS

Example: Jet Classification



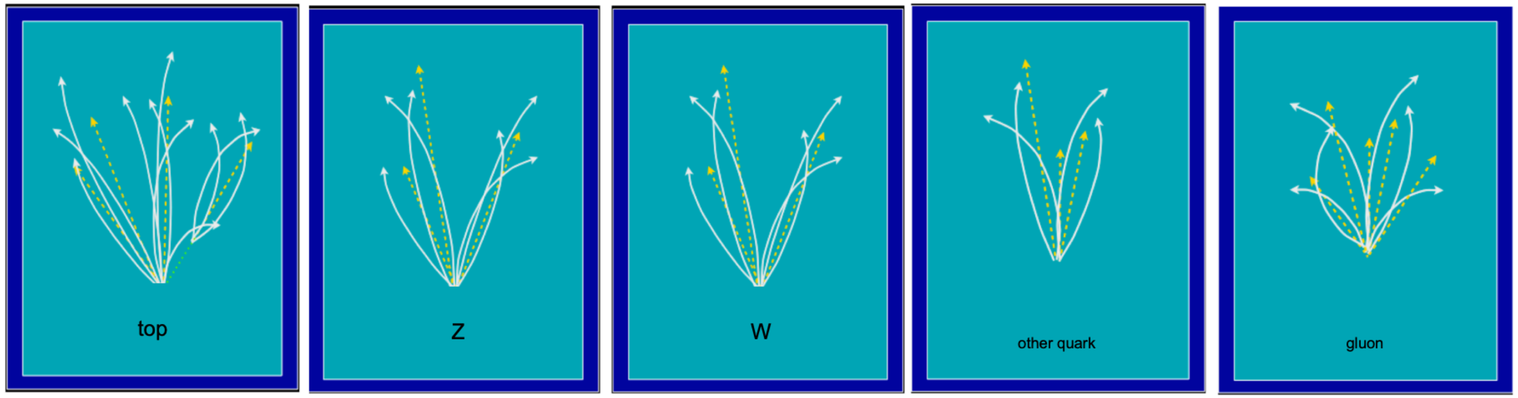


Perhaps an unrealistic example for L1 trigger, but lessons are useful

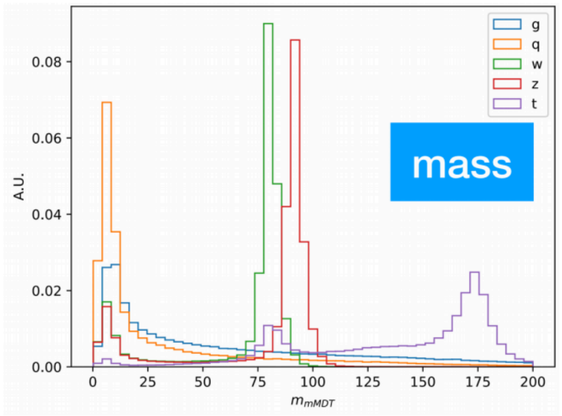
Jet Classification: 5-class classifier

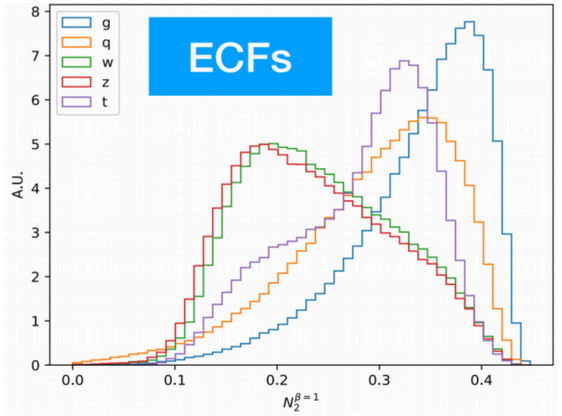
Five class classifier

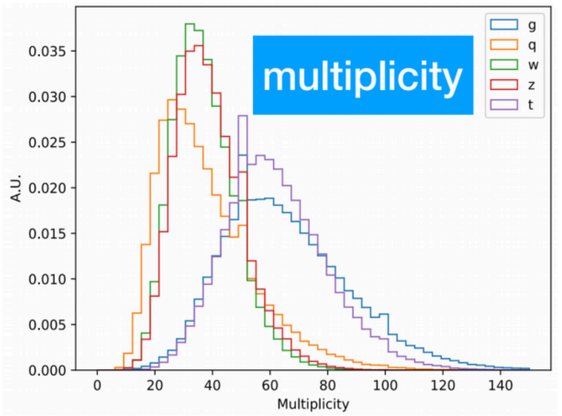
Sample: ~ 1M events with two boosted WW/ZZ/tt/qq/gg anti-kT jets



t→bW→bqq **Z→qq** **W→qq** **q/g background**







Observables

m_{mMDT}

$N_2^{\beta=1,2}$

$M_2^{\beta=1,2}$

$C_1^{\beta=0,1,2}$

$C_2^{\beta=1,2}$

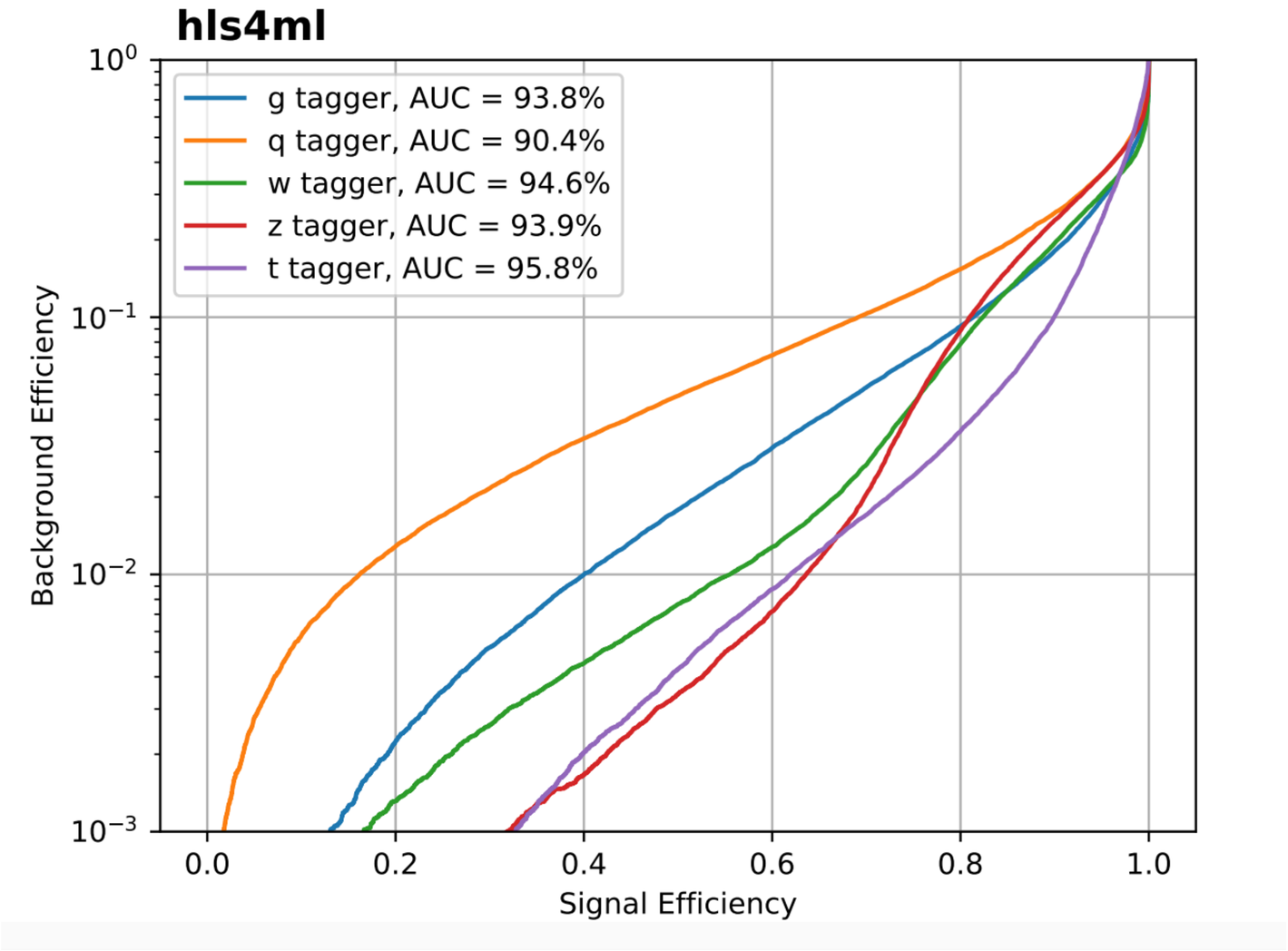
$D_2^{\beta=1,2}$

$D_2^{(\alpha,\beta)=(1,1),(1,2)}$

$\sum z \log z$

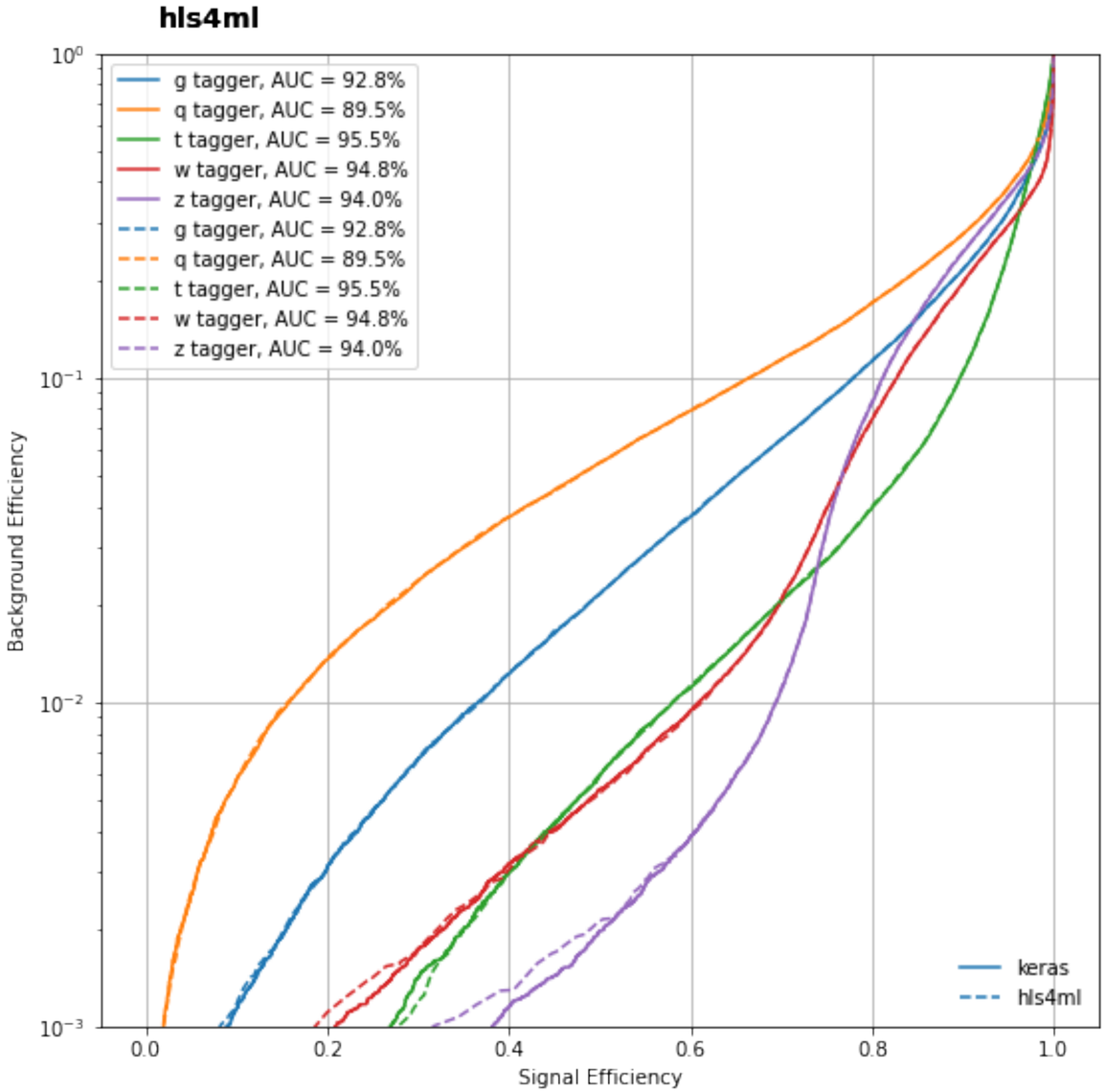
Multiplicity

Jet-tagging ROC



Jet-tagging ROC: Post Quantization

Used precision: <16,6>
Integet bits: 6
fractional bits: 10

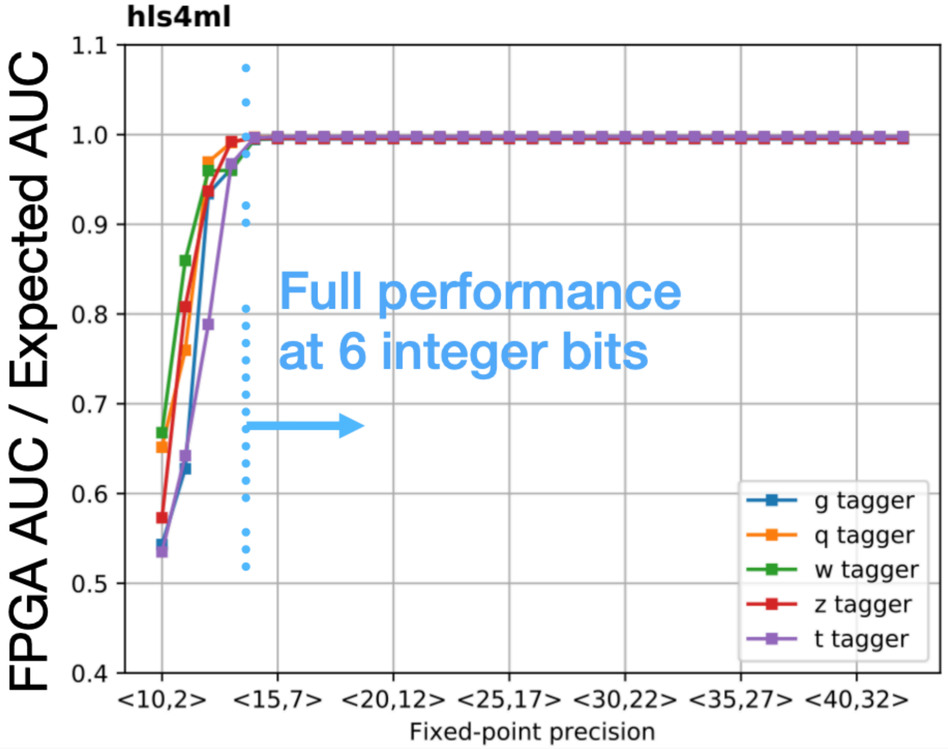


Scan to find optimal precision

ap_fixed<width bits, integer bits>

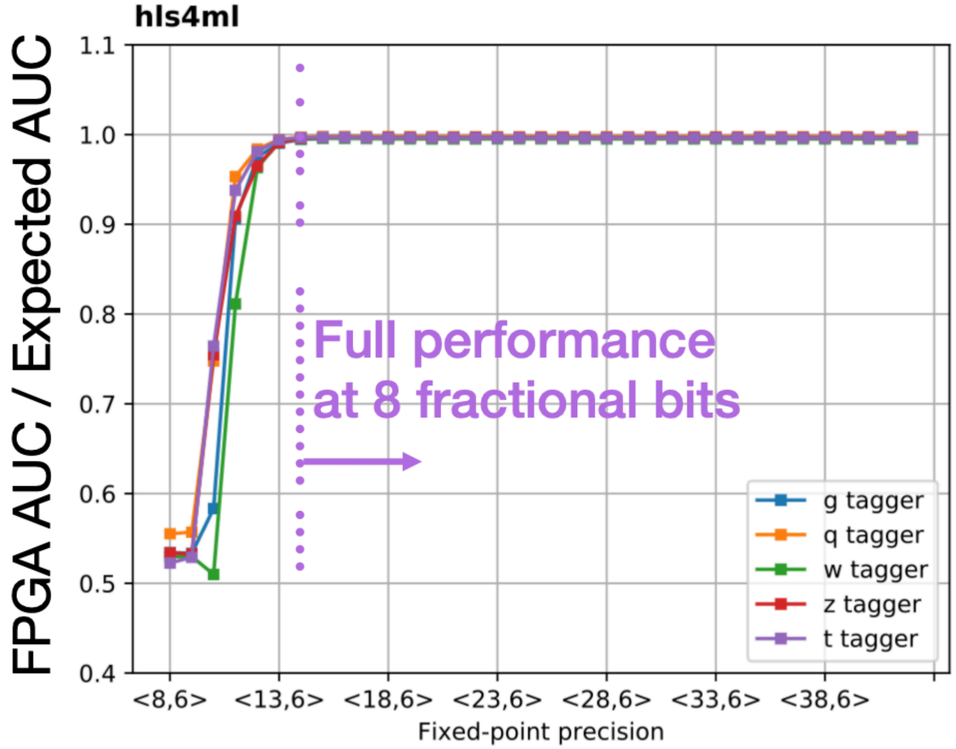
Scan integer bits

Fractional bits fixed to 8

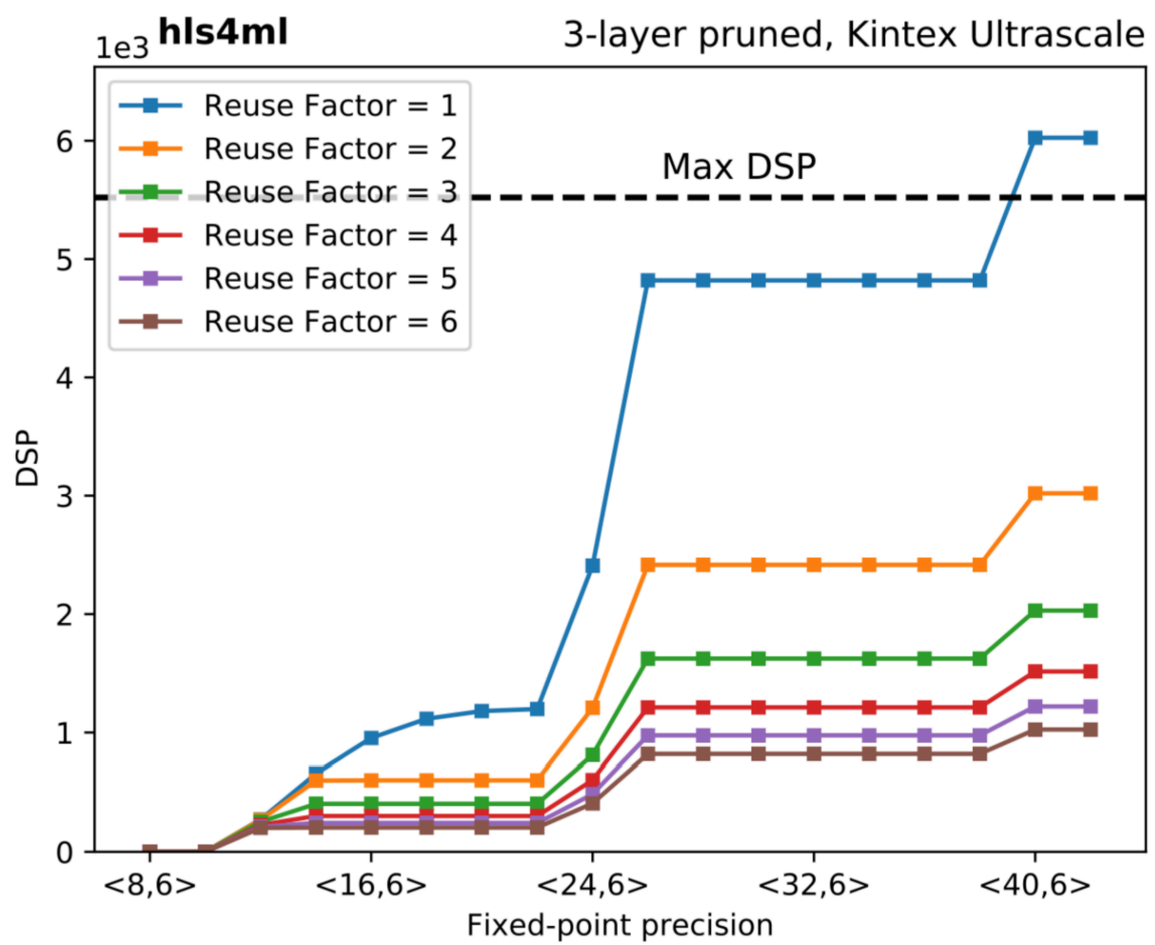


Scan fractional bits

Integer bits fixed to 6



(Example) 5-class jet-tagging: DSP usage



More resources

Fully parallel
Each mult. used 1x

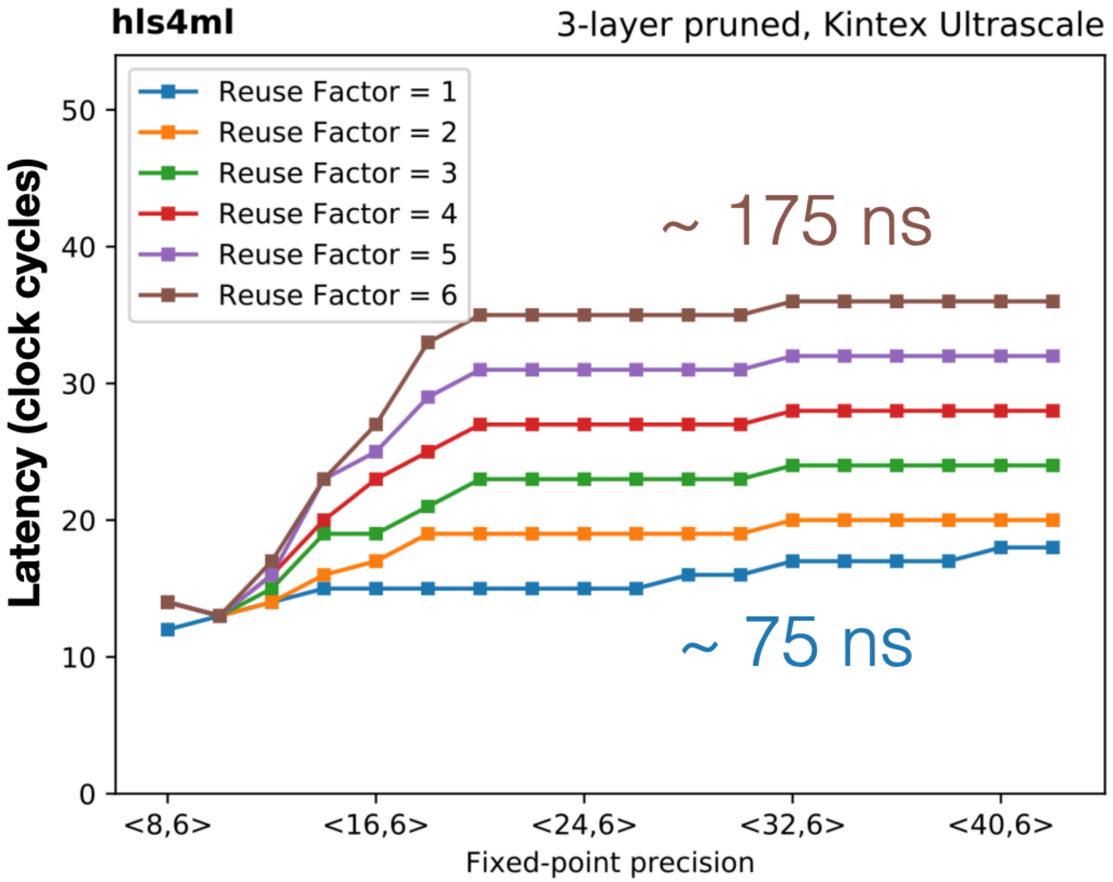
Each mult. used 2x

Each mult. used 3x

⋮

Longer latency

(Example) 5-class jet-tagging: Timing



Each mult. used 6x

⋮

Each mult. used 3x

⋮

Fully parallel
Each mult. used 1x

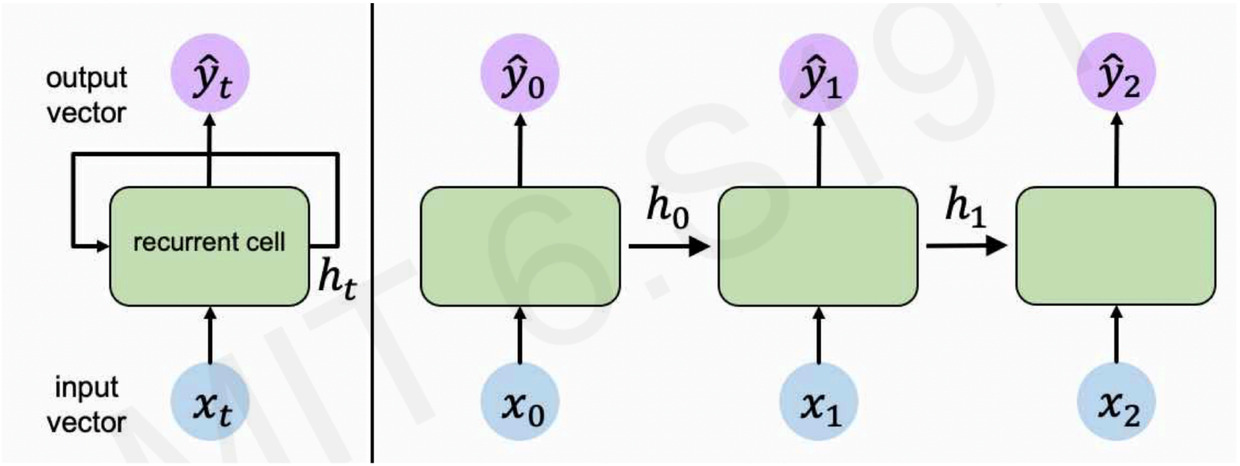
Longer latency

More resources

Recurrent Neural Network (RNN)

Recurrent Neural Networks

- Designed to work with sequential data
  - Text, audio, video, strokes, etc
- RNNs have a state, h_t, that is updated at each time step as the sequence is processed
- Recurrence relation at every time step



$$\hat{y} = f(x_t, h_{t-1})$$

Output Input past memory

$$h_t = f_W(x_t, h_{t-1})$$

cell state

Function with weights W

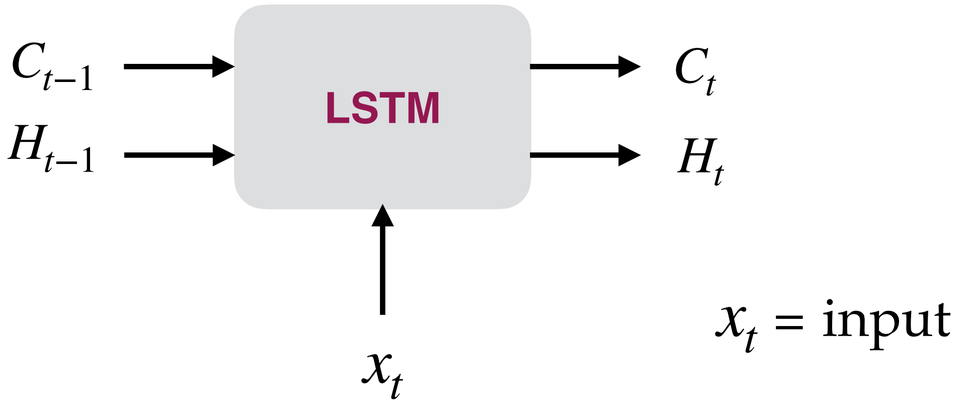
Input

old state

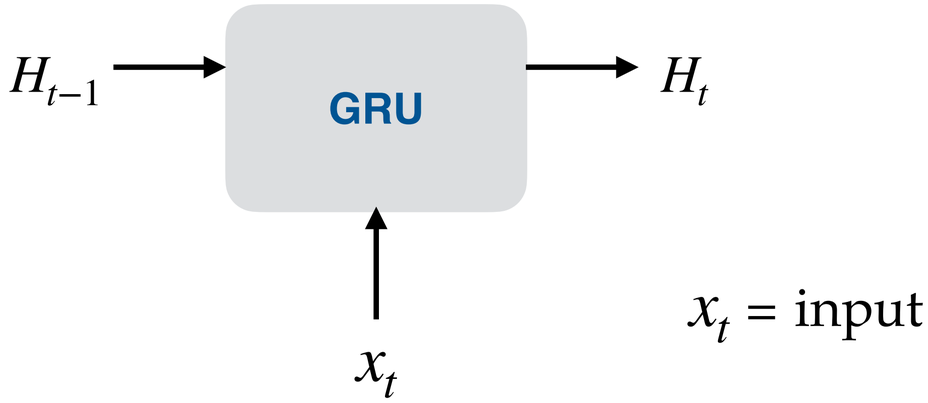
Implementation of RNN models:

- LSTM (Long Short-Term Memory)
- GRU (Gated Recurrent Unit)

LSTM vs GRU



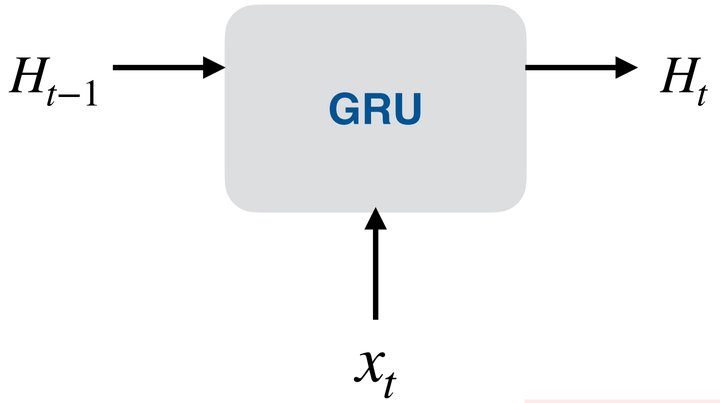
- **3 gates:** Input, Output, Forget
- **2 States:** Cell state (C_t) and Hidden state (H_t)

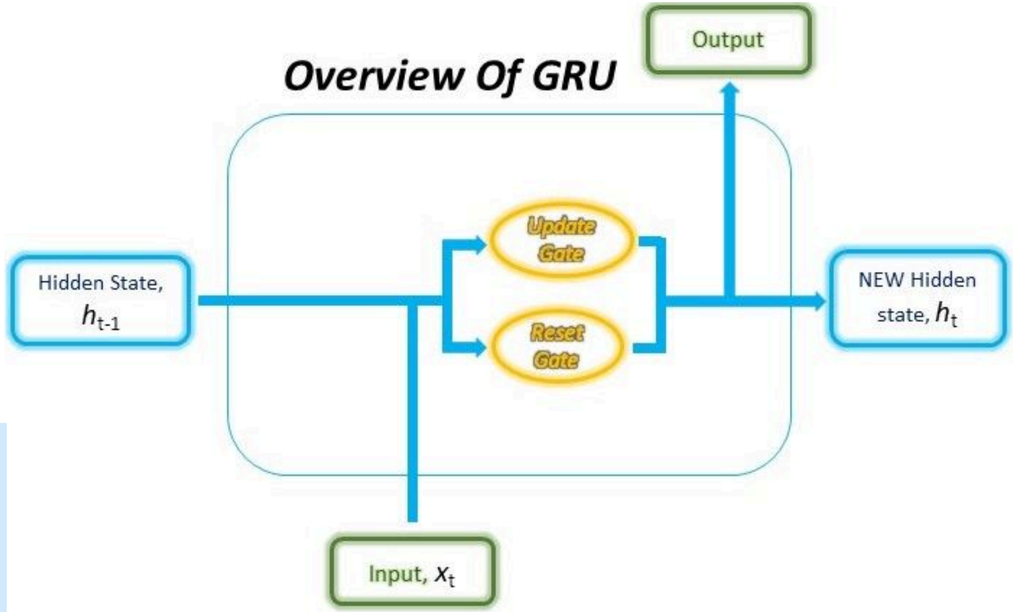


- **2 gates:** Update and Reset
- **Single** Hidden state (H_t)

- Less number of matrix multiplications
- Faster to train

Gated Recurrent Unit (GRU)





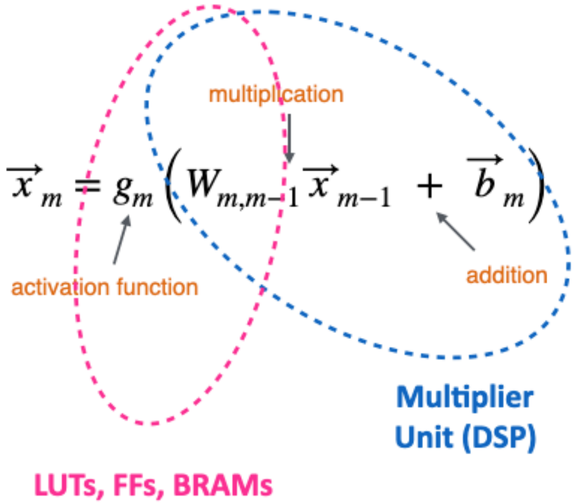
Dense **Dense**

Reset: $r_t = \sigma\ (W_{xr} \cdot x_t + b_r + W_{hr} \cdot h_{t-1})$

Update: $u_t = \sigma\ (W_{xu} \cdot x_t + b_u + W_{hu} \cdot h_{t-1})$

Candidate hidden state: $\tilde{h}_t = \tanh\ (W_{xh} \cdot x_t + b_h + (r_t \odot h_{t-1}) \cdot W_{hh})$

Output hidden state: $h_t = u_t \odot h_{t-1} + (1 - u_t) \cdot \tilde{h}_t$



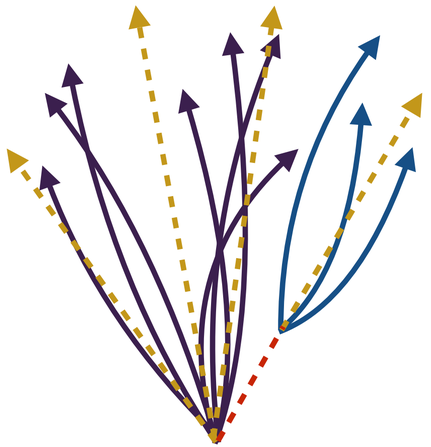
Benchmark Examples

Three benchmark cases

1. Binary classifier: ~4k parameters
   Identify top-quarks

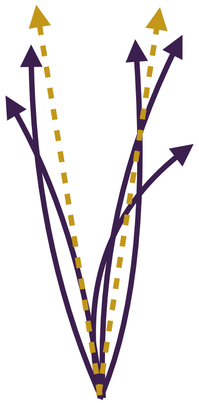
2. 3-class classifier: ~60k parameters
   Classify b / c / light jets

3. 5-class classifier: ~130k parameters
   QuickDraw dataset: differentiate between Bees, Butterflies, Mosquitos, Snails, Ants



top

b/c

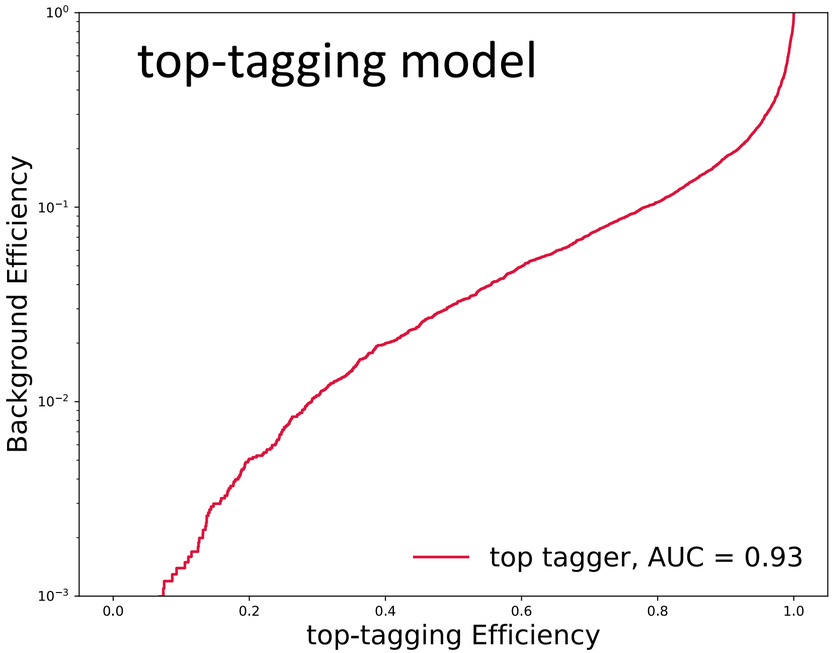


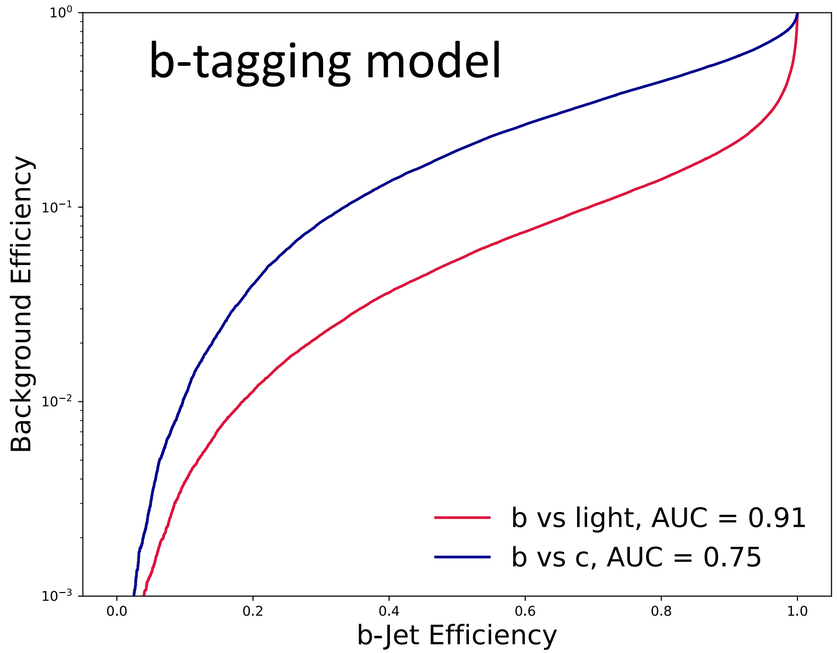
light

QuickDraw dataset

Model Performance: ROC

- All the benchmark models are trained using **Keras + TensorFlow**
- Weights and biases are represented by 32 bit floating point numbers



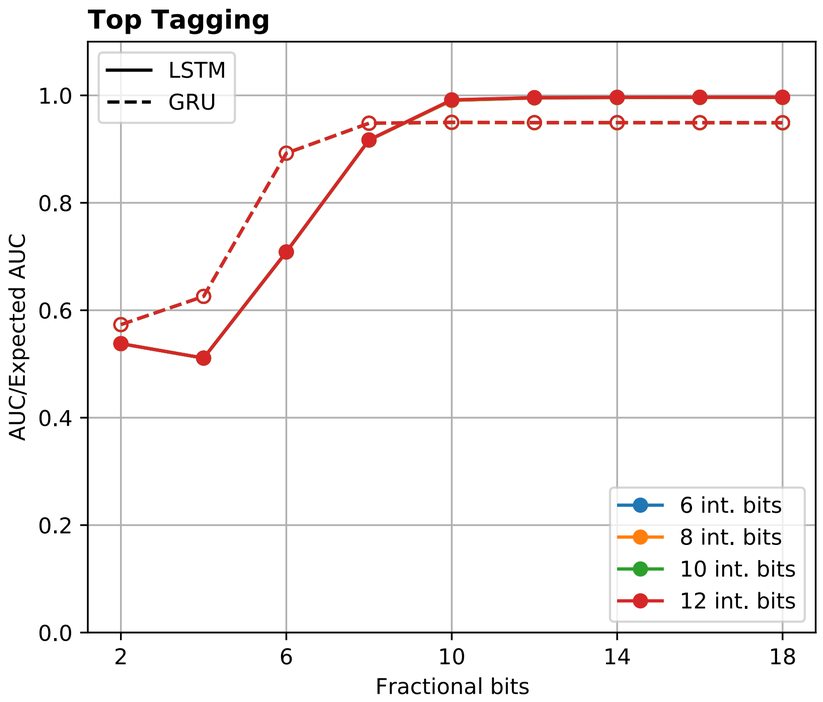


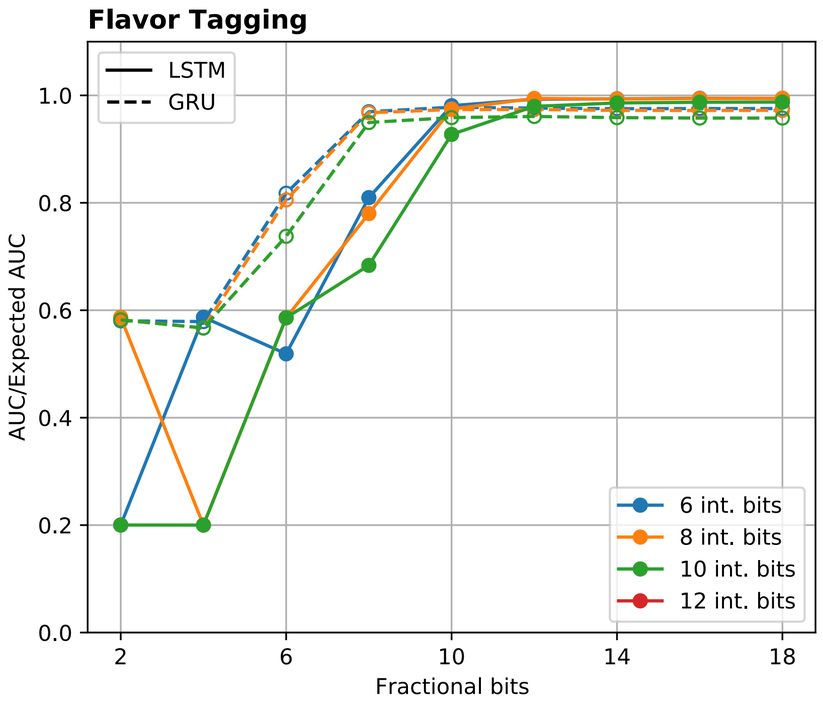
AUC after HLS conversion

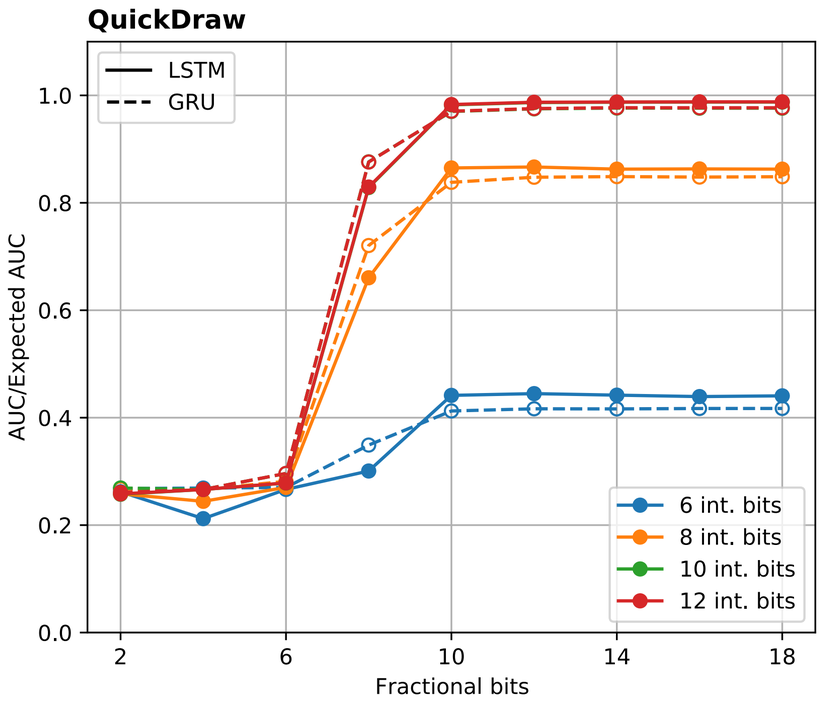
Mach. Learn.: Sci. Tec. 4 025004

$$\text{Relative AUC} = \frac{\text{AUC}_{\text{HLS}}}{\text{AUC}_{\text{Keras}}}$$

- Post-training quantized **LSTM models** (with optimal precision) performs similar to the floating-point models
- Small performance degradation (< 5%) in the **GRU models** after quantization





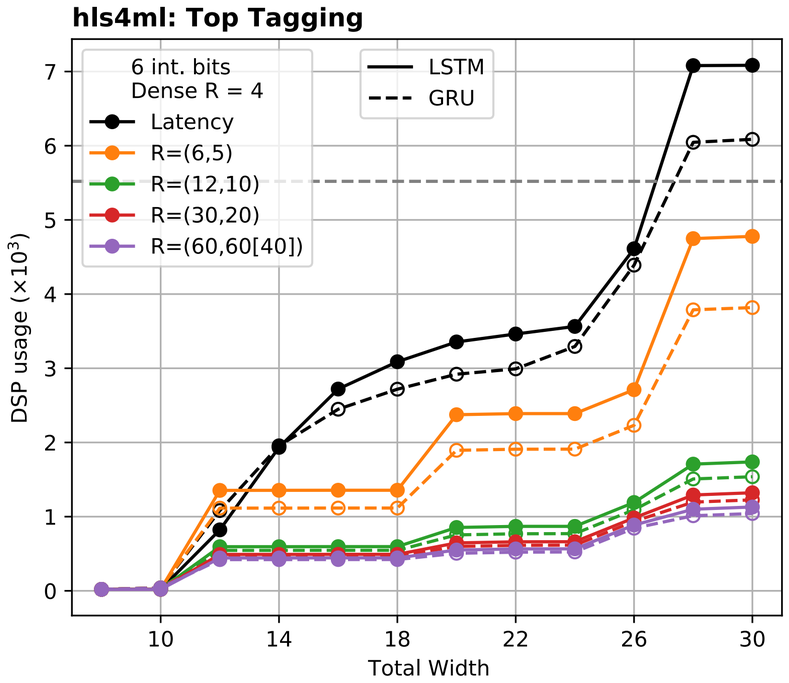


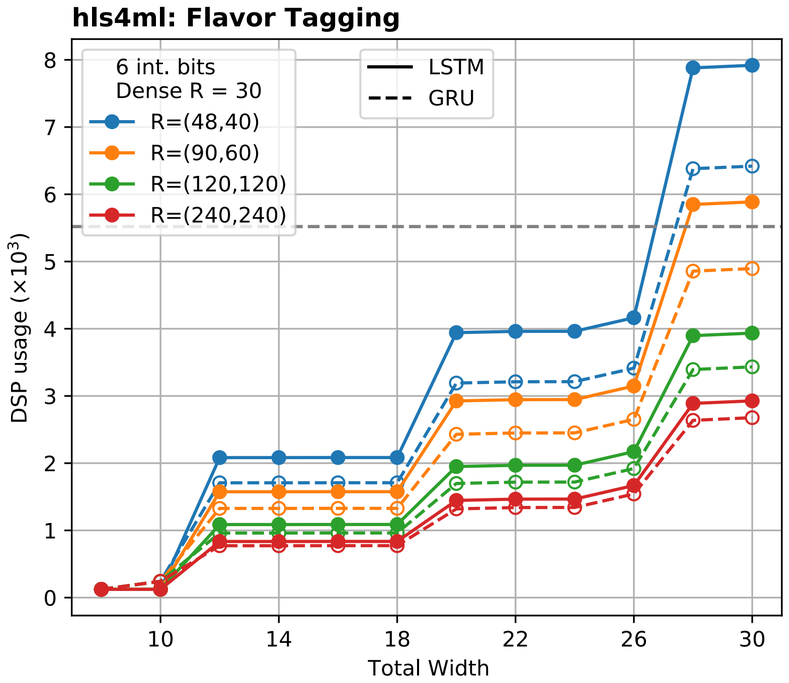
HLS Synthesis (RNN): DSP Usage

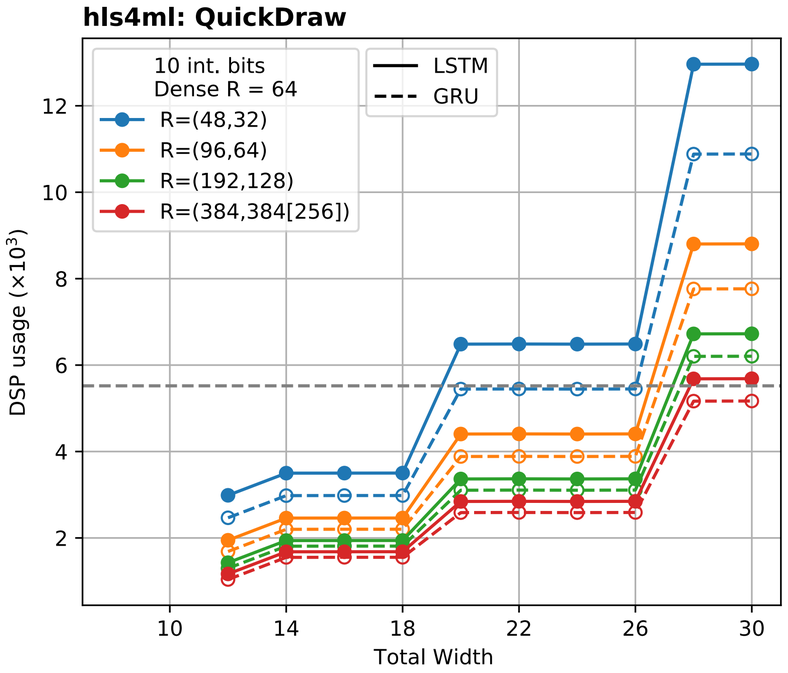
Mach. Learn.: Sci. Tec. 4 025004

- **DSP usage** as a function of **Total bit width** after HLS synthesis
- The Jumps correspond to DSP input width

Synthesized using Xilinx Kintex UltraScale FPGA
FPGA part: xcku115-flvb2104-2-i

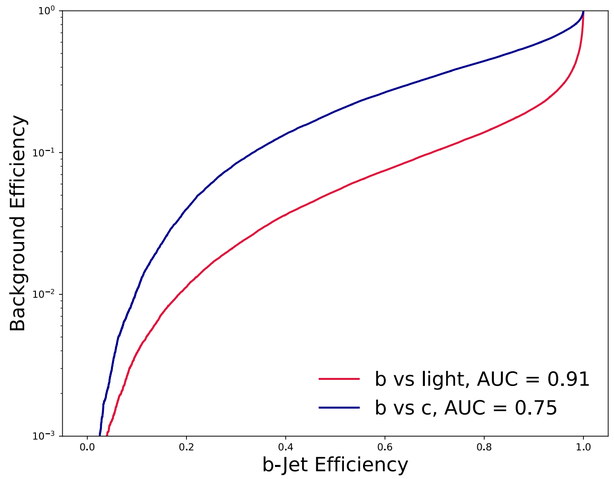






Summary: RNN Models

- RNN-based models could give good performance at trigger-level jet identification
- **Implemented LSTM and GRU layers inside hls4ml source code**
- **b-tagging cannot be done in ATLAS hardware trigger, but possible in CMS**



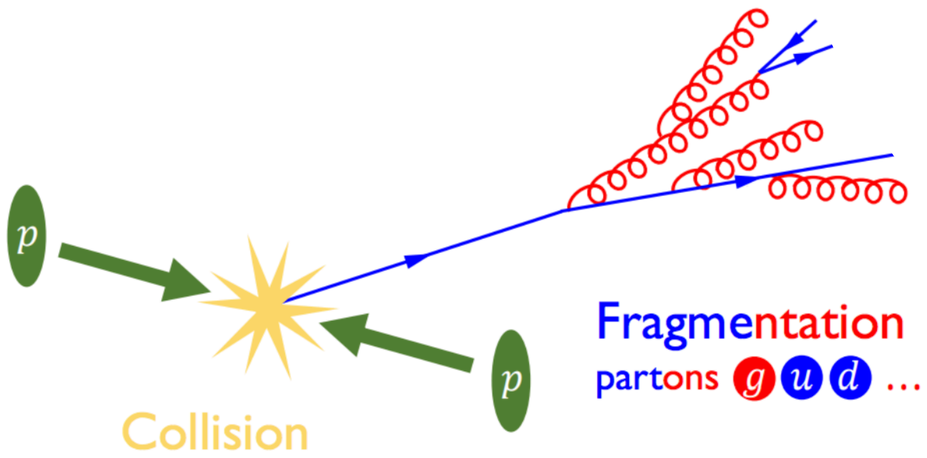
Jets

Due to QCD confinement we do not see quarks in isolation

→ *only exists in confinement of a hadron*

Parton Shower

Cascade of gluons



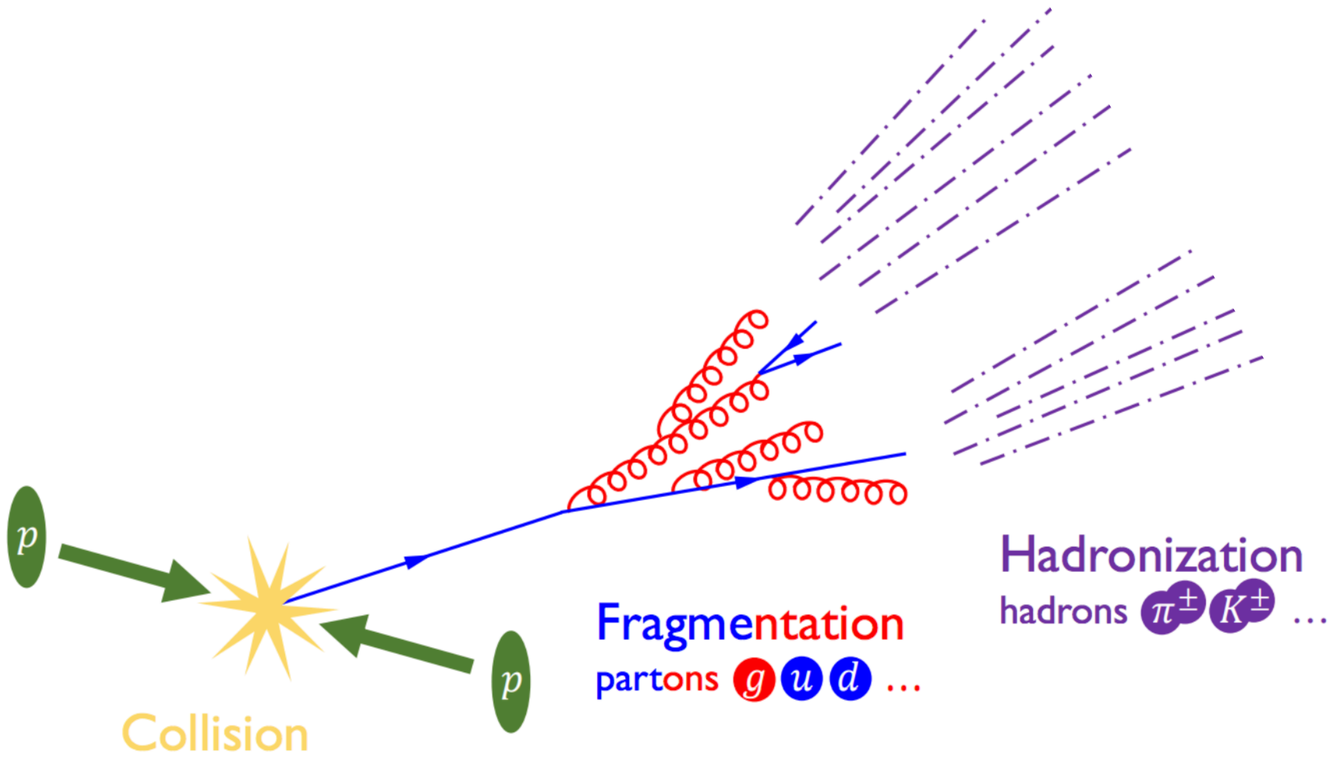
Jets

Due to QCD confinement we do not see quarks in isolation

→ *only exists in confinement of a hadron*

Parton Shower
Cascade of gluons

Jets:
Collection of particles



Jets

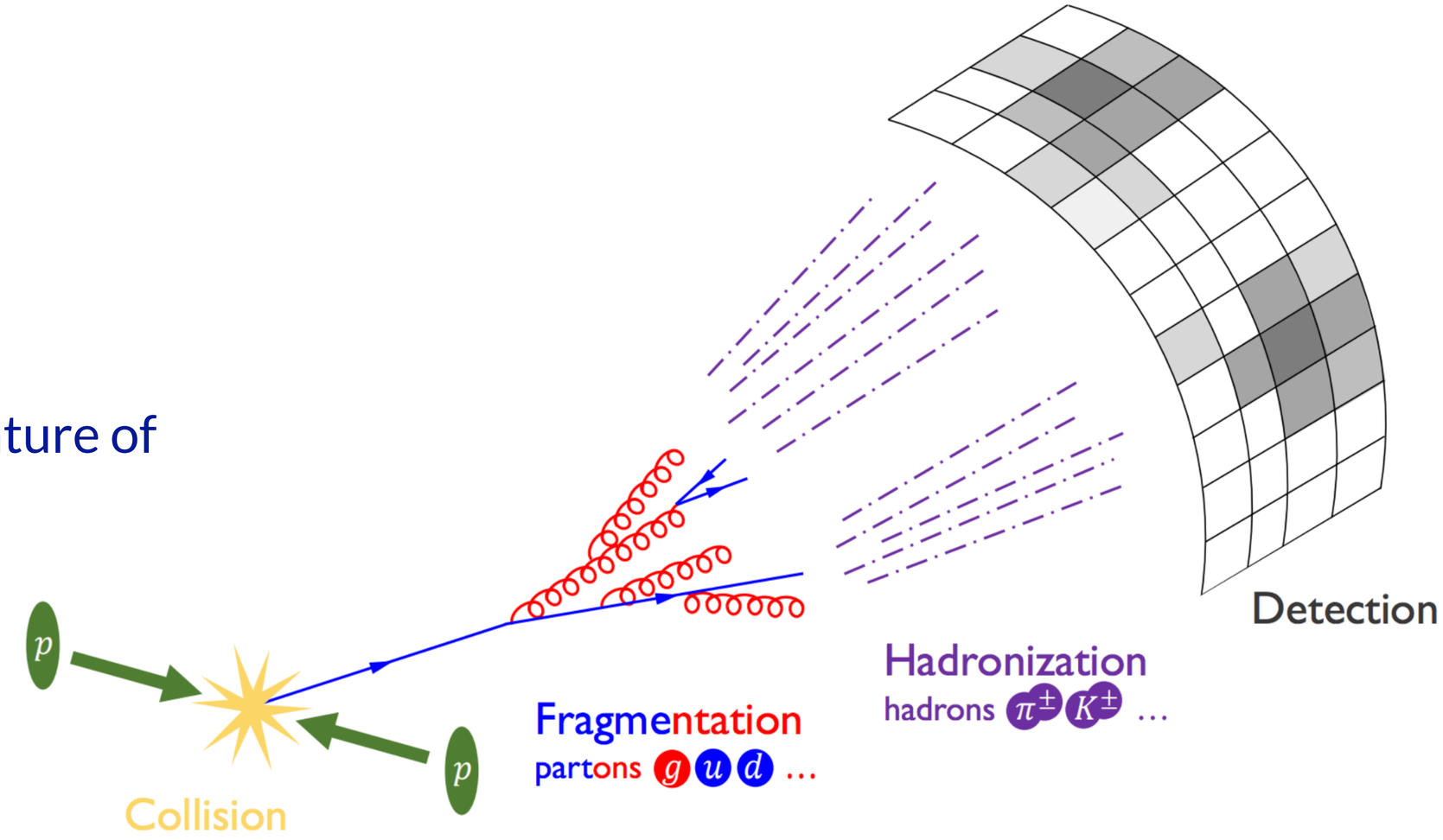
Due to QCD confinement we do not see quarks in isolation

→ *only exists in confinement of a hadron*

Parton Shower
Cascade of gluons

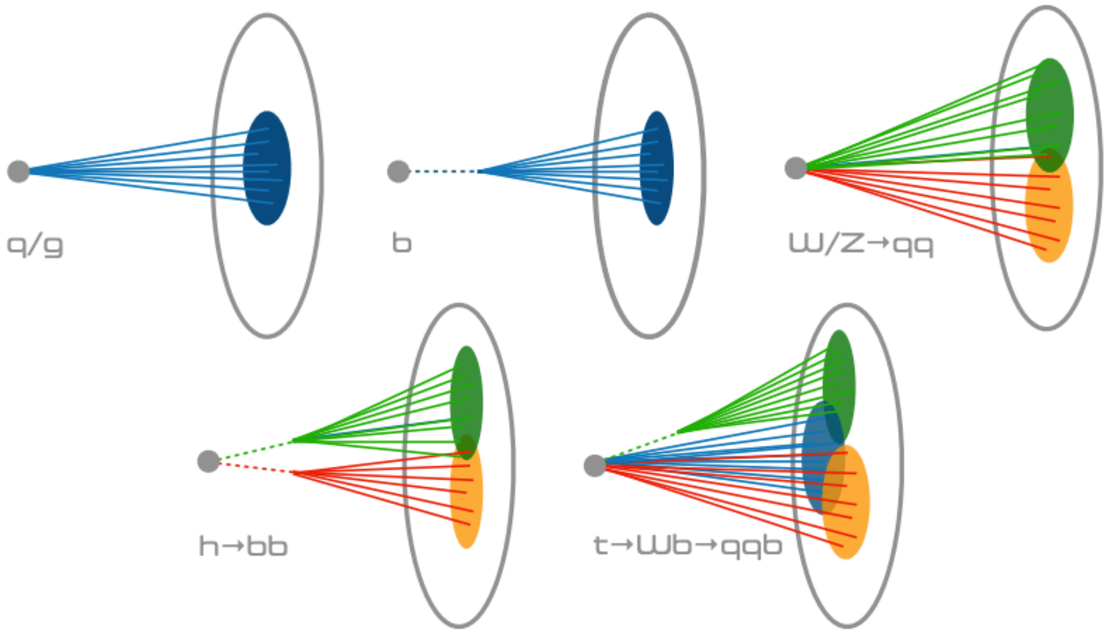
Jets:
Collection of particles

Jets are experimental signature of quarks and gluons

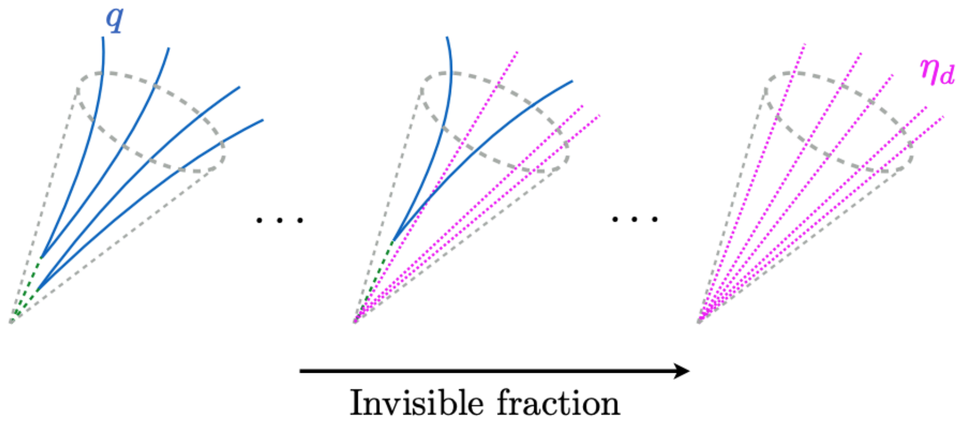


Structures within jets

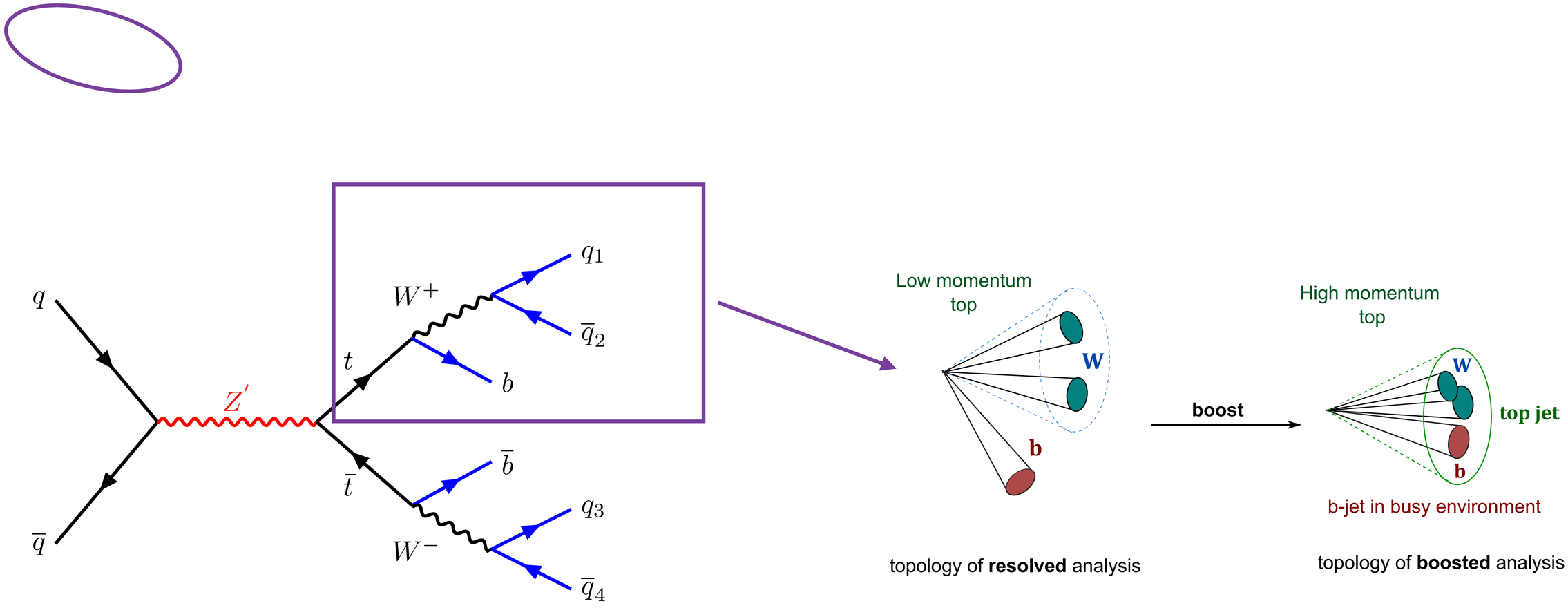
QCD jets
Could have different substructure



Semi-visible jets
Contains dark-hadrons



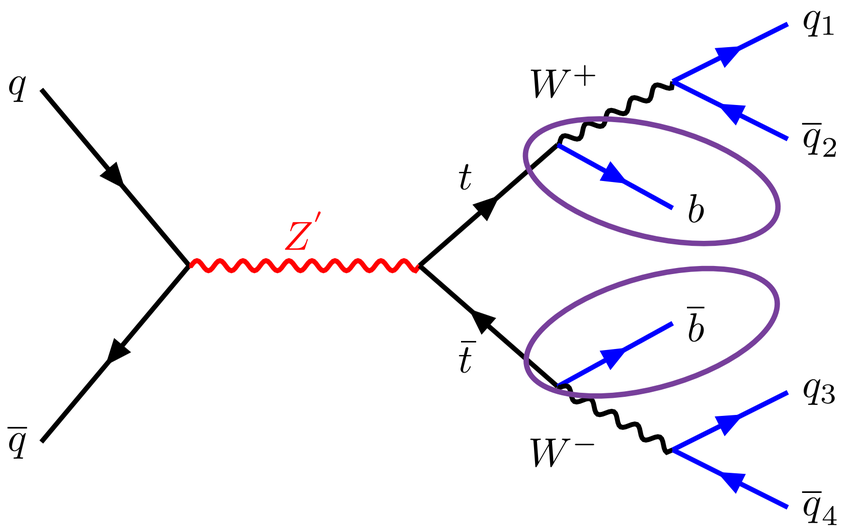
High Momentum Particles

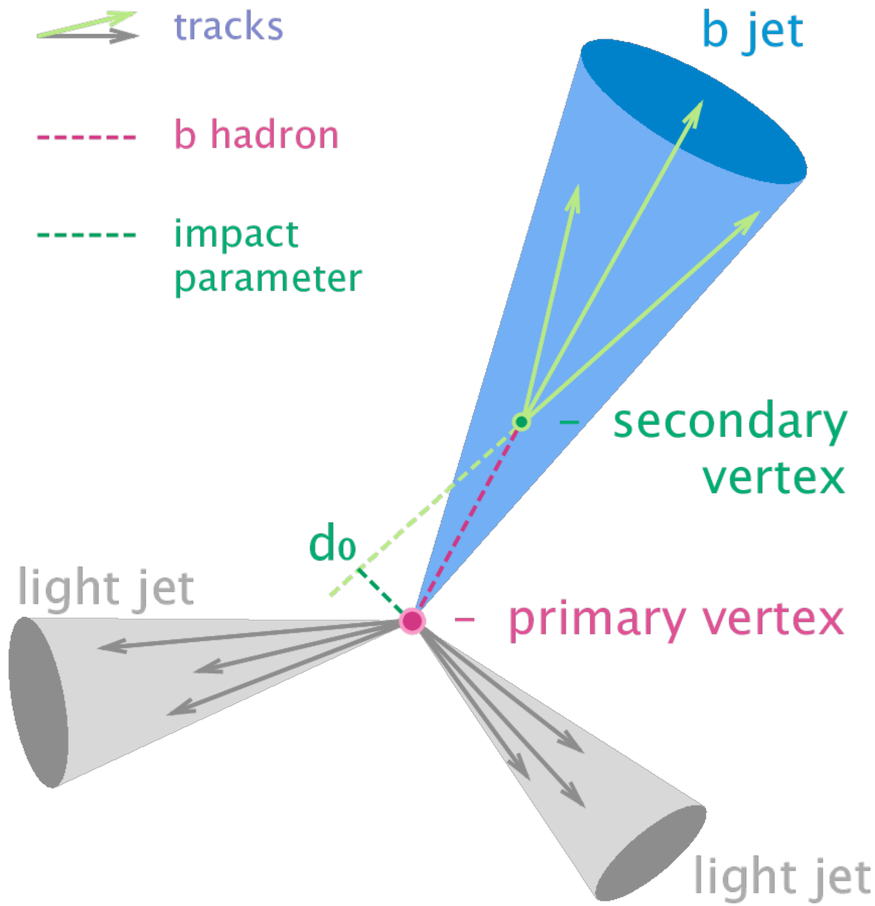


B-jet tagging

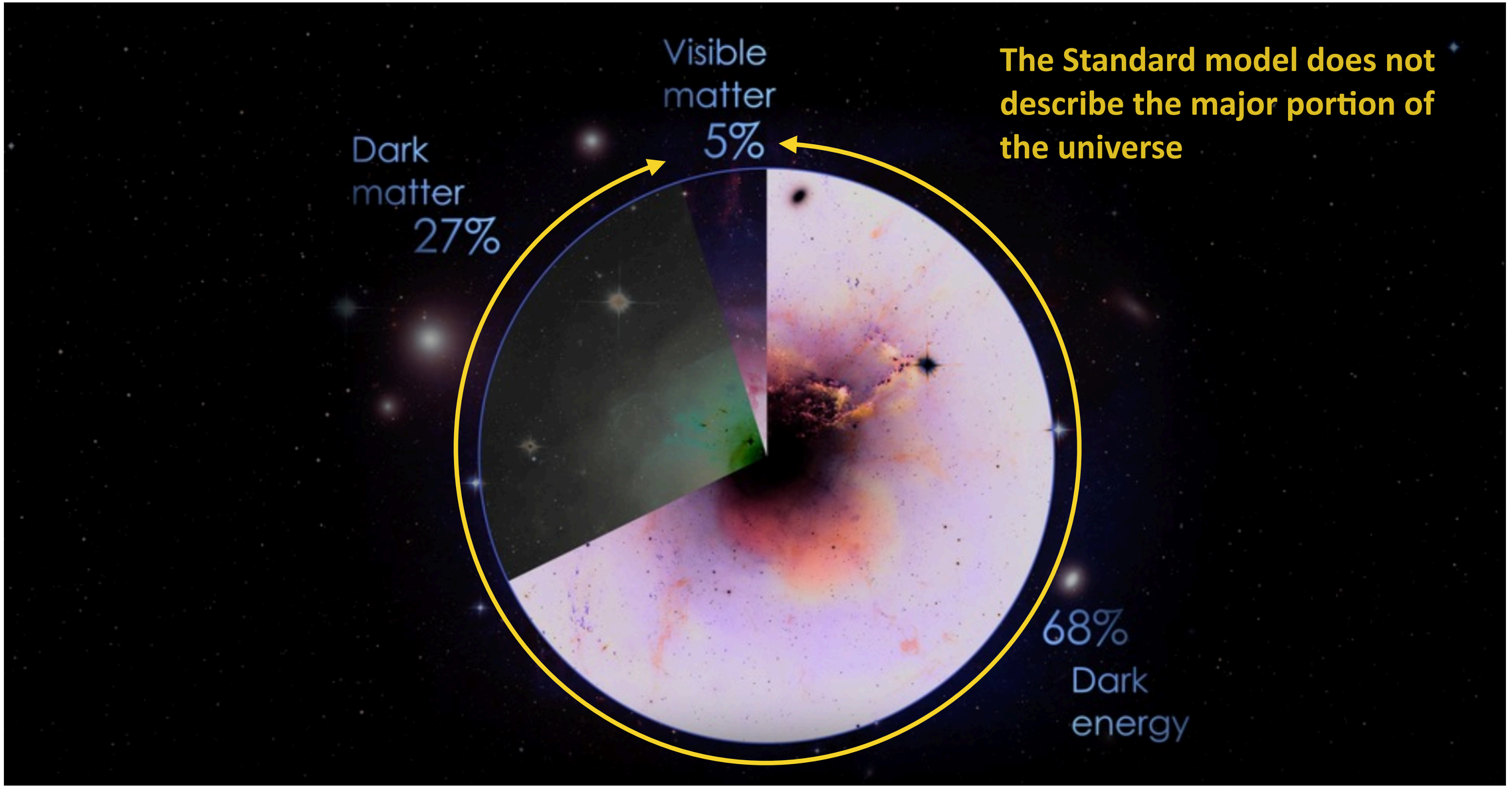
B-hadrons have measurable lifetime

- Creates displaced vertex
- Important quantity to identify b-jets





Physics Beyond Standard model



The Standard model does not describe the major portion of the universe