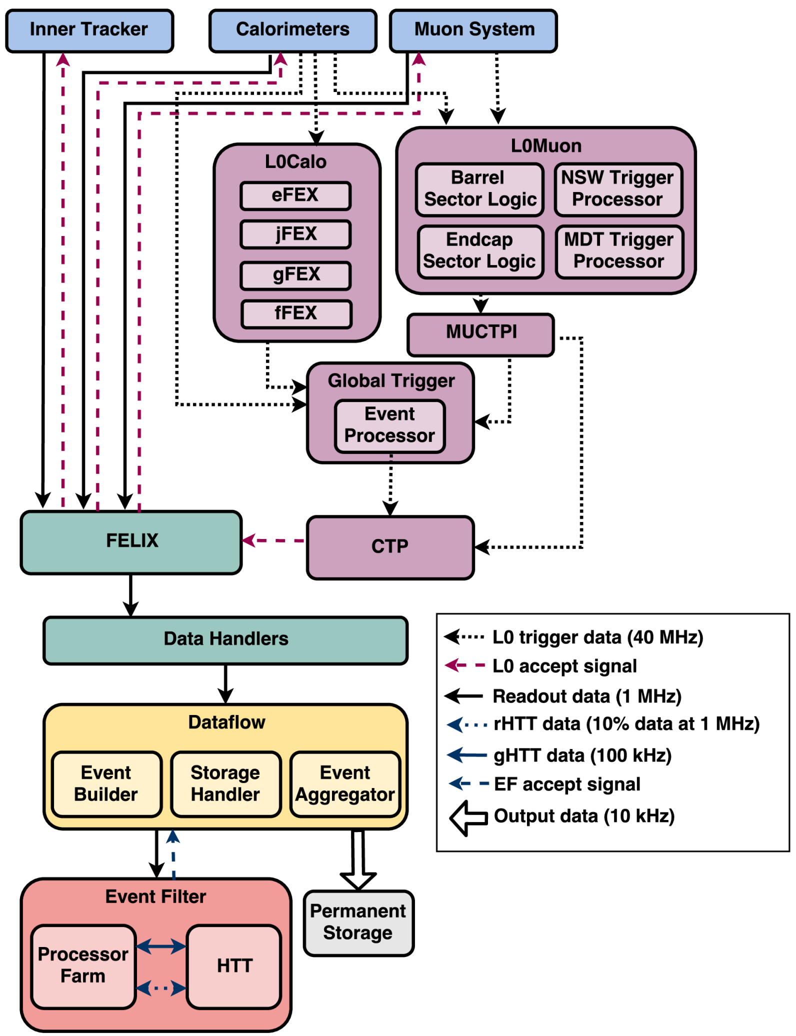
Quark/Gluon Tagger for GEP

Santosh Parajuli

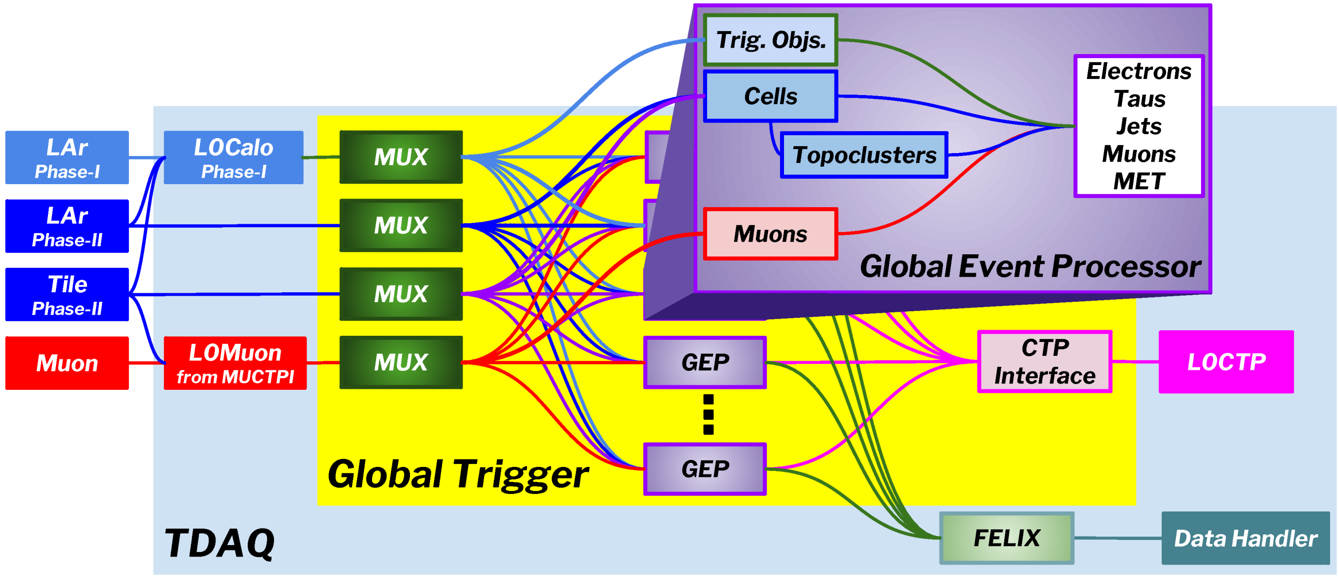
Southern Methodist University

22 May 2023

Global Trigger and Global Event Processer(GEP) in ATLAS



TDAQ System after the Phase-II upgrade (baseline design)



Schematic view of the Global Trigger System

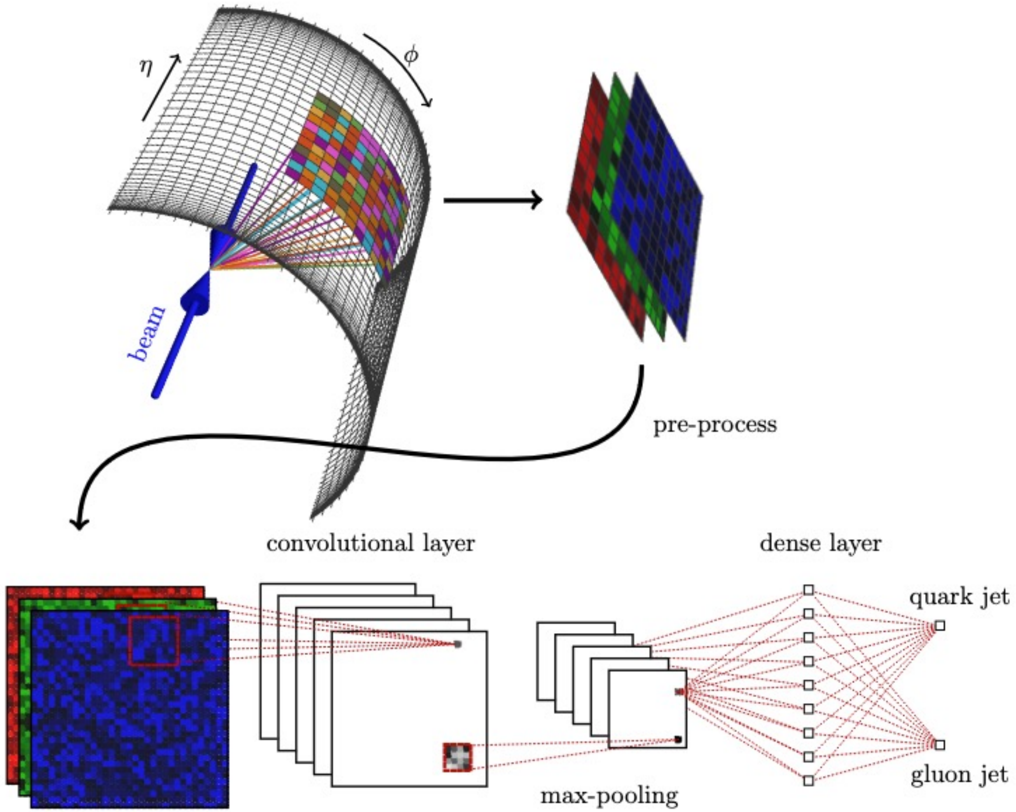
- Quark/Gluon tagging algorithm for GEP.

- Take the topocluster information (eta, phi, et), make an image of jet in eta-phi space.

- Use that image to train/test the CNN Model : binary classification.

- Latency constraint $< 1.2\,\mu s$.

Quark /Gluon Tagger



An illustration of the deep convolutional neural network architecture.

Sample Production

- **Framework:-** https://twiki.cern.ch/twiki/bin/view/Atlas/GlobalTriggerPerformance

- **Multi-jet Sample AODs:**
  - mc15_14TeV.800292.Py8EG_A14NNPDF23LO_jetjet_JZ2WithSW.recon.AOD.e8185_s3595_s3600_r12065---------------→ Using JZ2 sample only for this presentation.
  - mc15_14TeV.800293.Py8EG_A14NNPDF23LO_jetjet_JZ3WithSW.recon.AOD.e8185_s3595_s3600_r12065----------------→ These samples have very low number of events.
  - mc15_14TeV.800294.Py8EG_A14NNPDF23LO_jetjet_JZ4WithSW.recon.AOD.e8185_s3595_s3600_r12065
  - mc15_14TeV.800295.Py8EG_A14NNPDF23LO_jetjet_JZ5WithSW.recon.AOD.e8185_s3595_s3600_r12065
  - mc15_14TeV.800296.Py8EG_A14NNPDF23LO_jetjet_JZ6WithSW.recon.AOD.e8185_s3595_s3600_r12065

- Jets (R=0.4) reconstructed with Antikt algorithm using 422 topoclusters.

- AntiKt4emtopoCalo422Jets and AntiKt4TruthJets are saved in output ntuple.

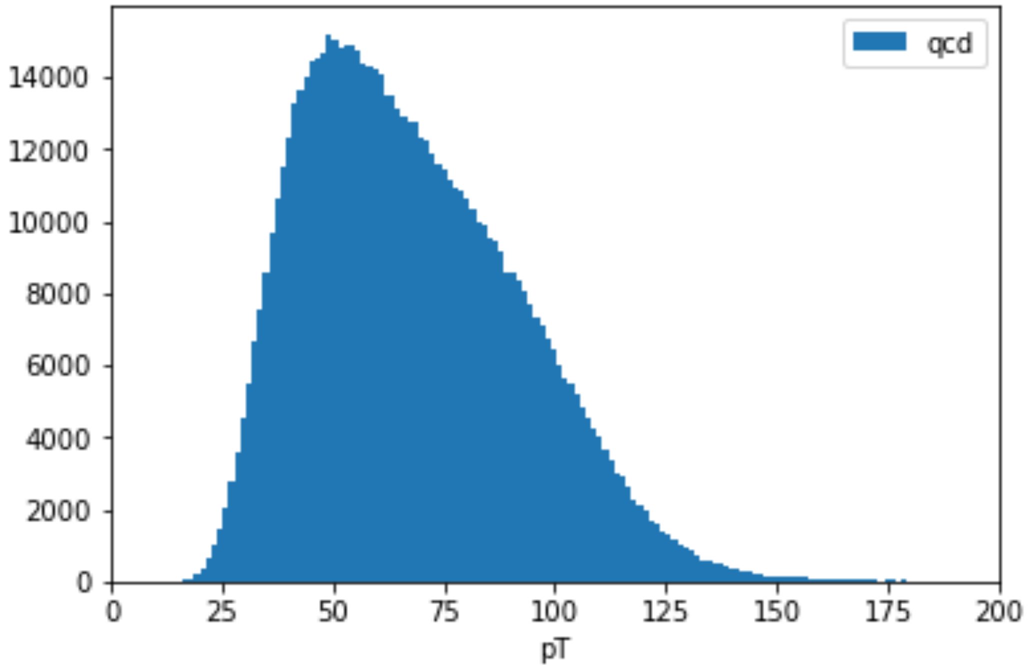
- Select the leading jet in each events for both AntiKt4emtopoCalo422Jets and AntiKt4TruthJets.

- Truth match AntiKt4emtopoCalo422Jets with AntiKt4TruthJets (If DeltaR between two jets < 0.4).

- Assign the parton truth label for truth matched AntiKt4emtopoCalo422Jets

- Use parton truth label to classify leading R=0.4 jets as quark jets and gluon jets.

Plots of leading Jets

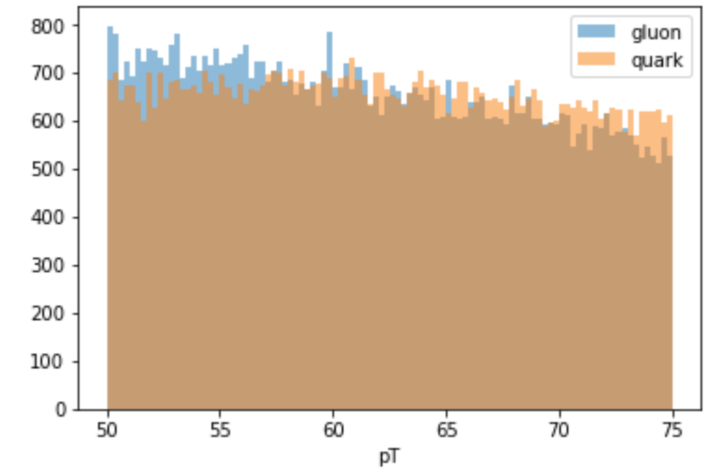


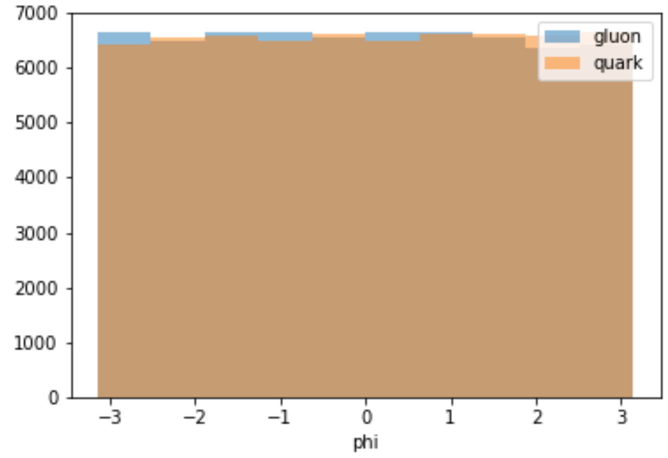
- Only JZ2WithSW sample has sufficient number of events that can be use in training while selecting leading jets for each events.
- Choose a slice in pT range
  - pT Range -> 50-75 GeV
  - pT Range-> 75-100 GeV

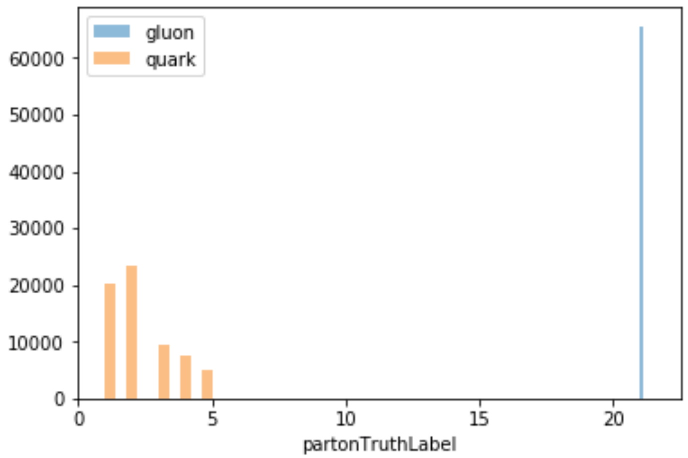
Plots of leading Jets (50 < pT < 75)

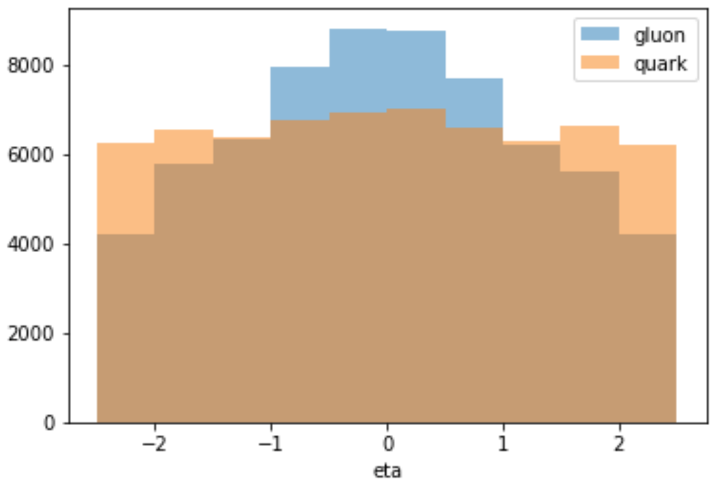
(65,611*2 total events)

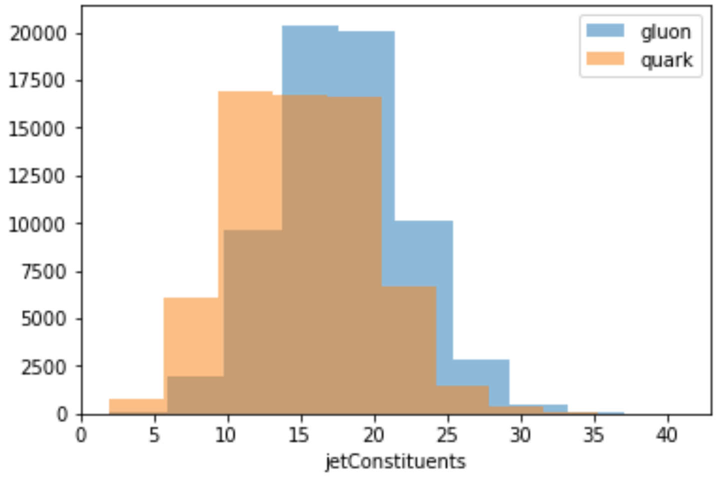
- **Selection: |eta| < 2.5 & 50 < pT < 75**











Quark and Gluon jets Selection

- For Quark: partonTruthLabel >=1 and <=5 are selected.
- For Gluon: partonTruthLabel = 21 is selected.
- |eta| < 2.5
- 65,611 signal(quark jets) and 65,611 Background (gluon jets) events are used here. Total-> 131,222 events for 50 < pT < 75 GeV jets.

Making the Jet Image in eta-phi space

- Topological cluster inputs from calorimeter are used to make the jet image.

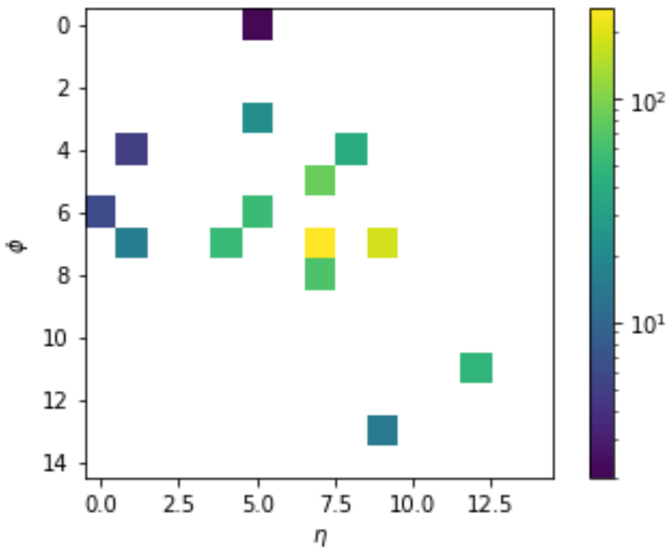
- The constituents inside a jet are translated in eta and phi so that the jet's center is located at center in eta-phi space.

- 15 * 15 pixel image in eta-phi space.

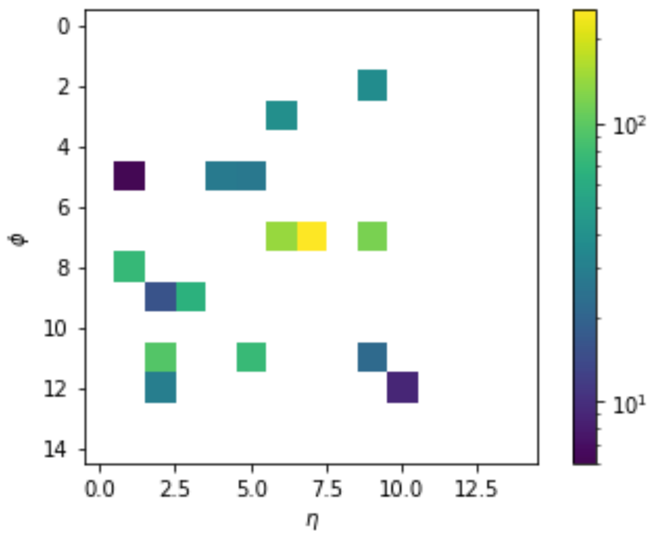
- The intensity of each pixel is the total cluster energy with in the pixel.

- Pixel value is then normalized by dividing it by the value of the hottest (maximum) pixel in the image. This scaling ensures that the pixel values of the entire image are between 0 and 1. Then, the pixel values are scaled to a range between 0 and 255, this is done by multiplying each pixel value by 255.

Quark and Gluon Images



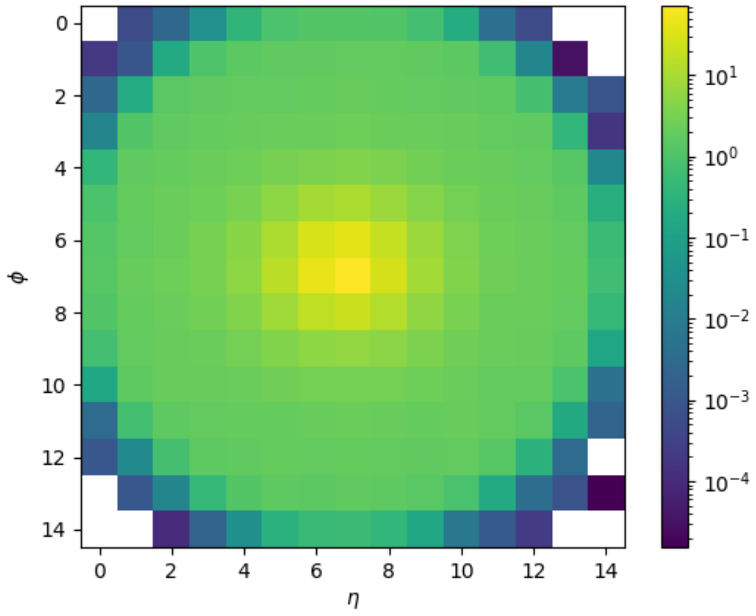
Quark Jets



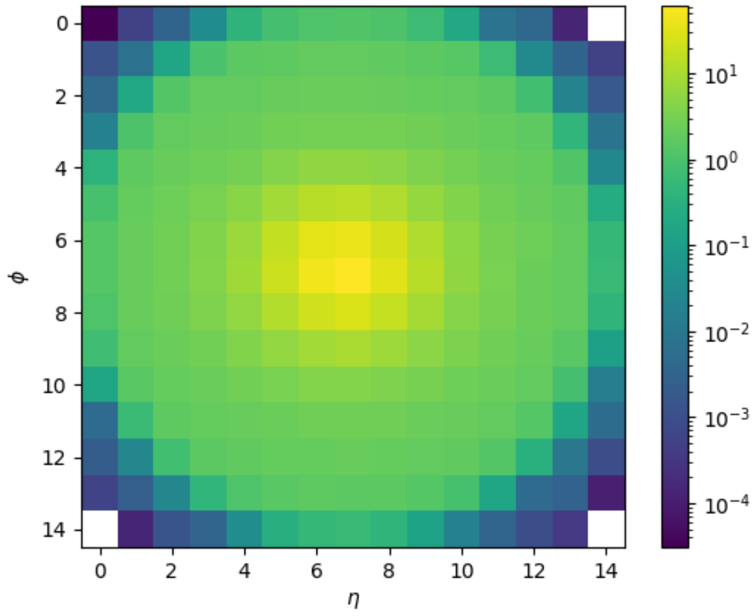
Gluon Jets

Comparison of Quark (left) and Gluon (right) jet images ($50 < pT < 75$ GeV).

Average Images



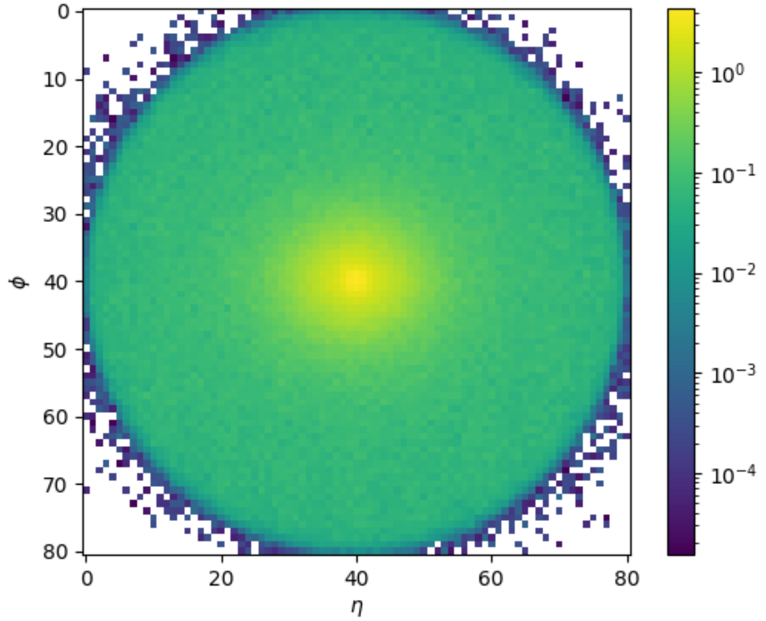
Quark Jets



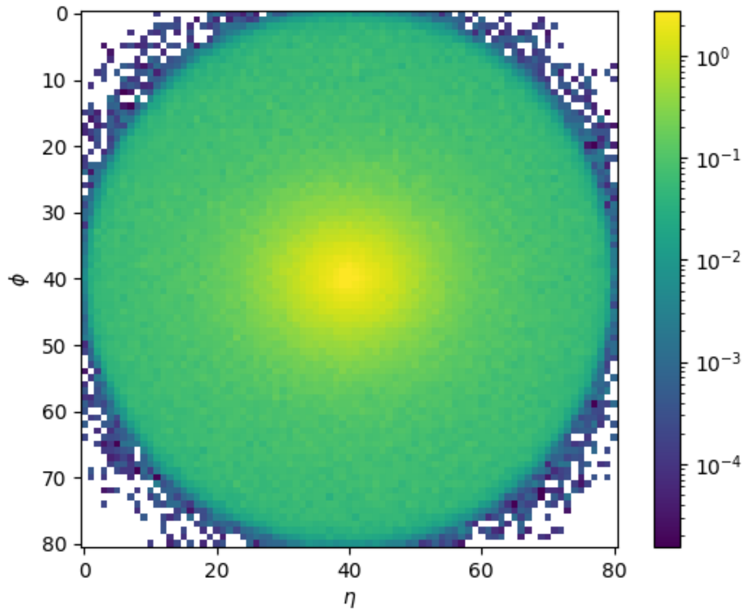
Gluon Jets

Comparison of Quark (left) and Gluon (right) jet images averaged over 65,611 jets (50 < pT < 75 GeV).

Average Images



Quark Jets



Gluon Jets

Comparison of Quark (left) and Gluon (right) jet images averaged over 65,611 jets (50 < pT < 75 GeV). (80*80)

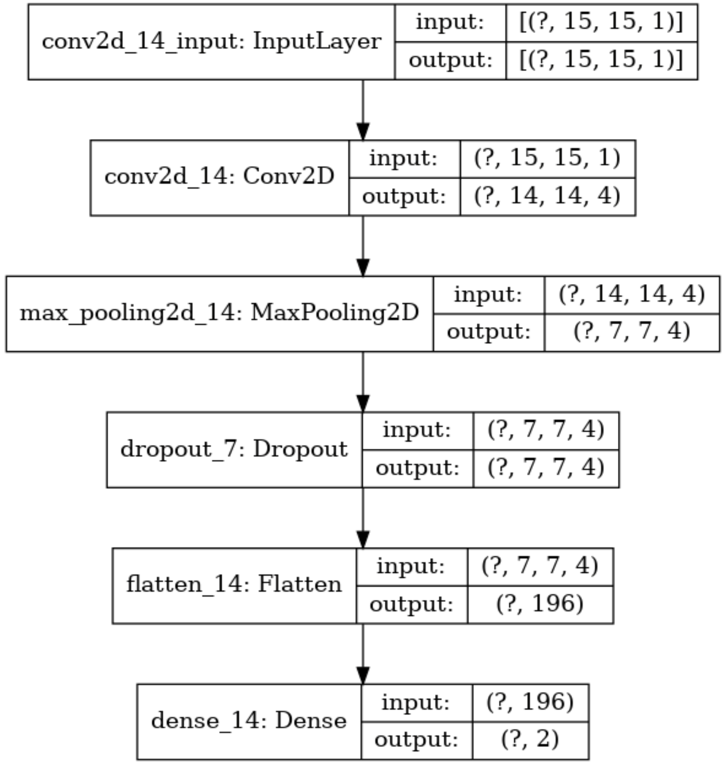
CNN Model and Performance:

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Dropout, Activation, Conv2D, MaxPooling2D
import tensorflow as tf

model_cnn = Sequential()
model_cnn.add(Conv2D(4, (2, 2), input_shape=(15, 15,1), activation='relu'))
model_cnn.add(MaxPooling2D(pool_size=(2, 2)))
model_cnn.add(Dropout(0.1))
model_cnn.add(Flatten())
model_cnn.add(Dense(2, activation='softmax'))

model_cnn.compile(loss='categorical_crossentropy',
                  optimizer=tf.keras.optimizers.Adam(learning_rate=1.e-4),
                  metrics=['accuracy'])

history_cnn = model_cnn.fit(x_train, y_train, validation_split=0.2, epochs=50,
                            batch_size=256, shuffle=True, verbose=1)

model_cnn.summary()
```

4 Filter

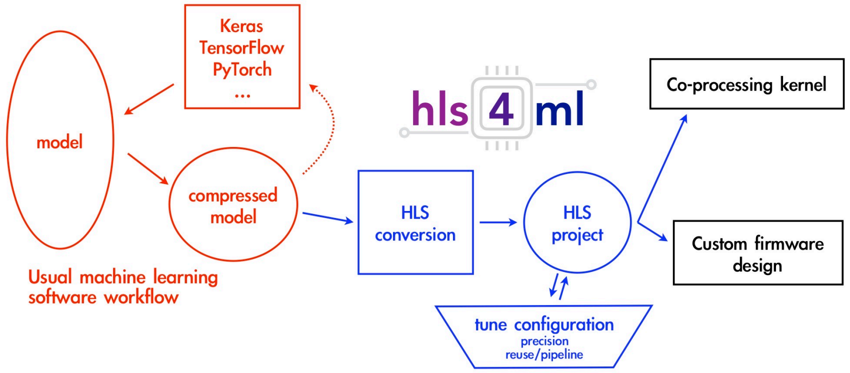
15*15 grayscale(1) image

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_14 (Conv2D)           (None, 14, 14, 4)         20        
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 7, 7, 4)           0         
_________________________________________________________________
dropout_7 (Dropout)          (None, 7, 7, 4)           0         
_________________________________________________________________
flatten_14 (Flatten)         (None, 196)               0         
_________________________________________________________________
dense_14 (Dense)             (None, 2)                 394       
=================================================================
Total params: 414
Trainable params: 414
Non-trainable params: 0
```



CNN Model

HLS4ML

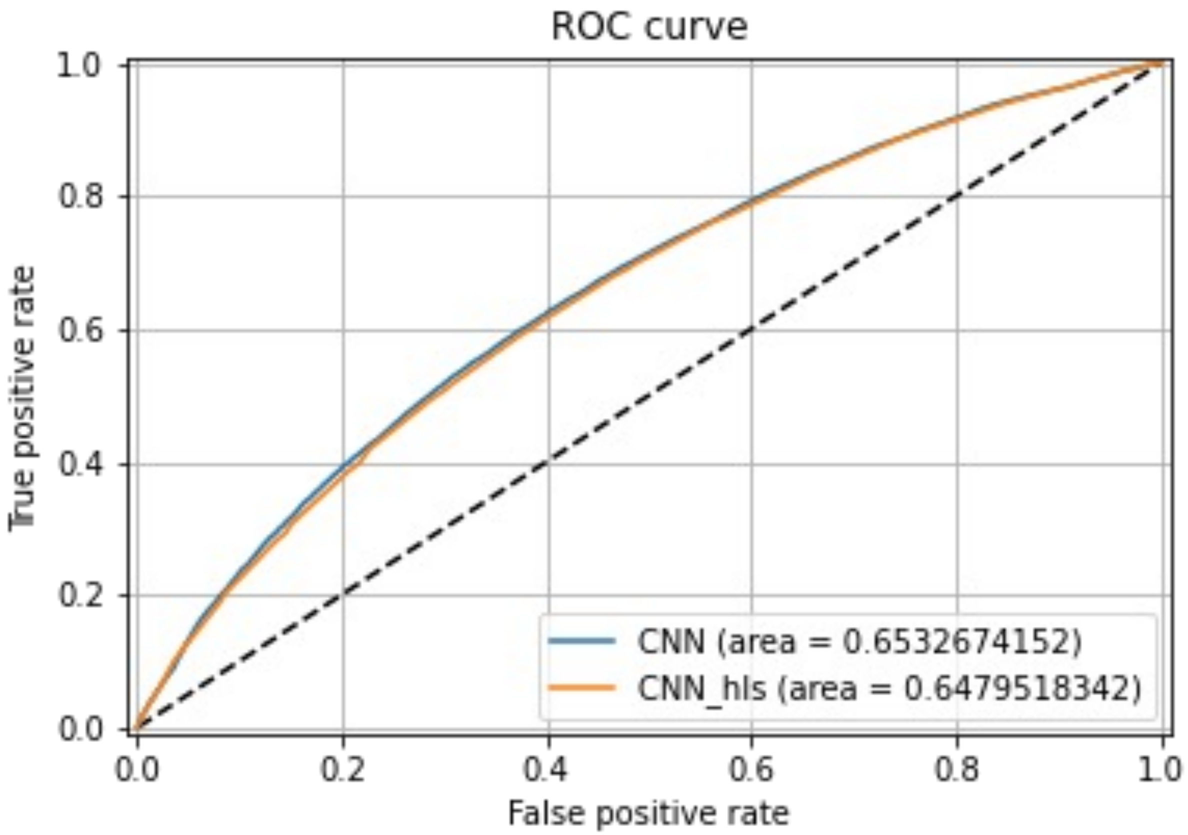


Workflow of hls4ml

- In the Global Event Processors (GEP), each trigger algorithm will be performed in an Algorithm Processing Unit (APU).

- At each Bunch Crossing, the detector would send the data to the GEP, and GEP would process those data using APU in pipelining.

HLS4ML Performance:

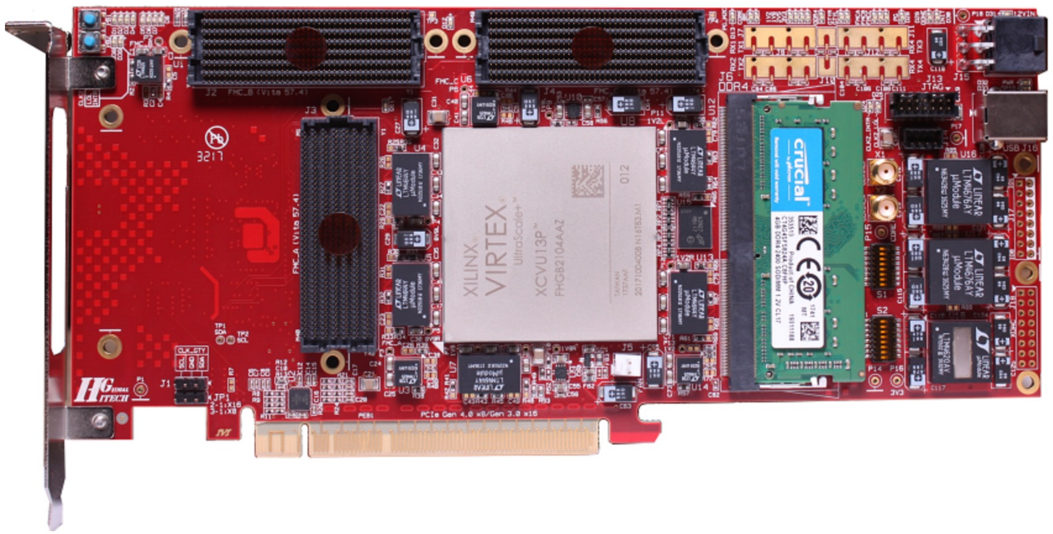


CNN Model/hls4ml Performance

HLS4ML Performance:

```
================================================================
== Vivado HLS Report for 'myproject'
================================================================
* Date:           Tue Apr 11 10:05:25 2023

* Version:        2018.3 (Build 2405991 on Thu Dec 06 23:56:15 MST 2018)
* Project:        myproject_prj
* Solution:       solution1
* Product family: virtexuplus
* Target device:  xcvu13p-flga2577-2L-e
```

```
================================================================
== Performance Estimates
================================================================
+ Timing (ns):
    * Summary:
    +--------+-------+----------+------------+
    |  Clock | Target| Estimated| Uncertainty|
    +--------+-------+----------+------------+
    |ap_clk  |   5.00|     4.321|        0.62|
    +--------+-------+----------+------------+

+ Latency (clock cycles):
    * Summary:
    +-----+-----+-----+-----+----------+
    |  Latency  |  Interval | Pipeline |
    | min | max | min | max |   Type   |
    +-----+-----+-----+-----+----------+
    |  233|  233|  229|  229| dataflow |
    +-----+-----+-----+-----+----------+
```



Latency ~1.17 μs

hls4ml Performance

HLS4ML Performance:

```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-------+---------+---------+------+
|       Name      | BRAM_18K| DSP48E|    FF   |   LUT   | URAM |
+-----------------+---------+-------+---------+---------+------+
|DSP              |        -|      -|        -|        -|     -|
|Expression       |        -|      -|        0|       32|     -|
|FIFO             |       12|      -|      458|      576|     -|
|Instance         |        2|    299|     5280|    14268|     -|
|Memory           |        -|      -|        -|        -|     -|
|Multiplexer      |        -|      -|        -|       36|     -|
|Register         |        -|      -|        6|        -|     -|
+-----------------+---------+-------+---------+---------+------+
|Total            |       14|    299|     5744|    14912|     0|
+-----------------+---------+-------+---------+---------+------+
|Available        |     5376|  12288|  3456000|  1728000|  1280|
+-----------------+---------+-------+---------+---------+------+
|Utilization (%)  |    ~0   |      2|    ~0   |    ~0   |     0|
+-----------------+---------+-------+---------+---------+------+
```

hls4ml Performance

Summary

- Studied CNN model for q/g for pT range 50 - 75 GeV.
- AUC in ROC is ~0.65.
- Latency to run model ~1.17 μs.

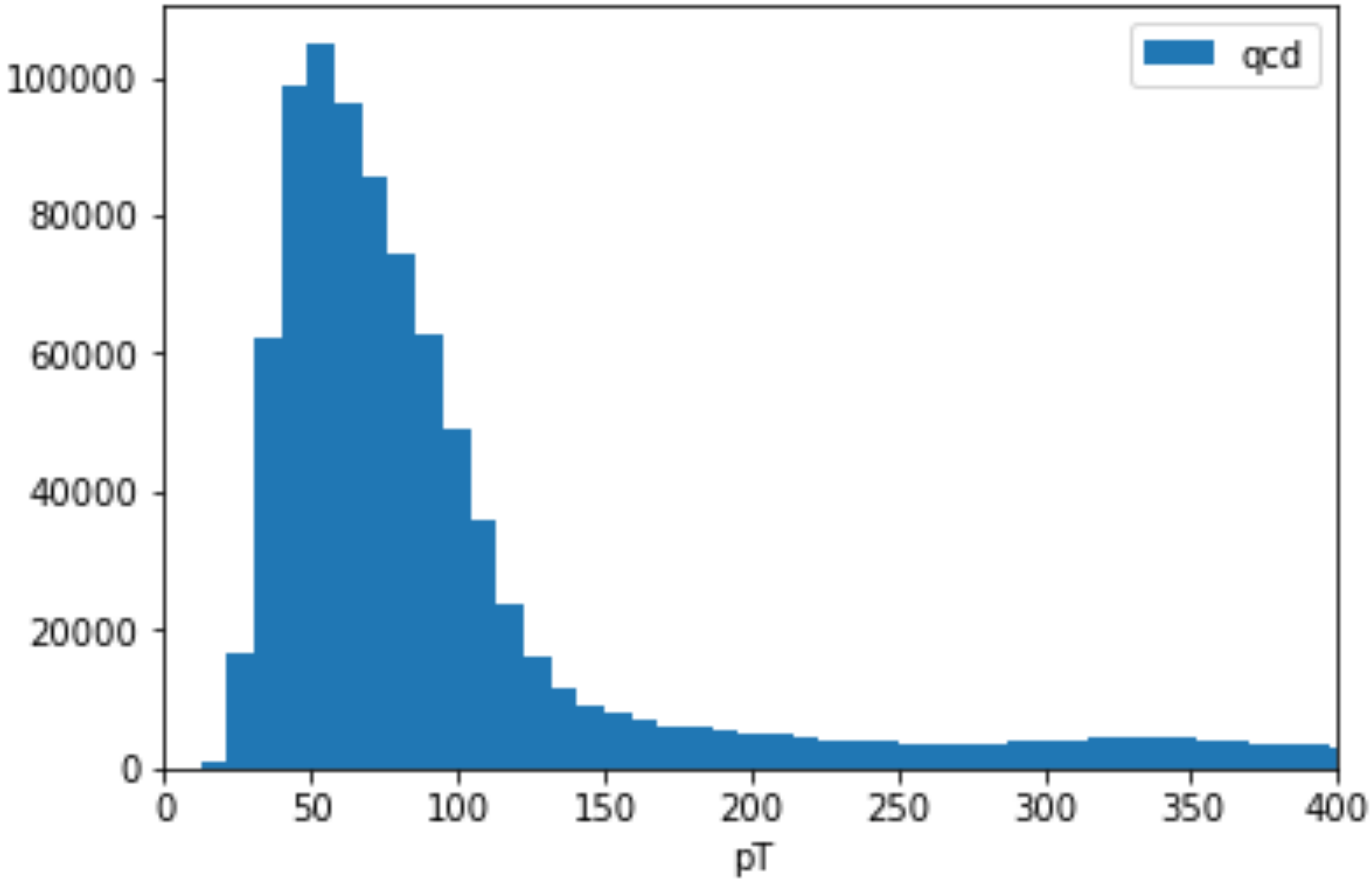
Machine learning evaluation in the Global Event Processor FPGA for the ATLAS trigger upgrade

Zhixing Jiang[1,*], Bowen Zuo[1], Ben Carlson[4], Yuval Frid[2], Maayan Tamari[2], Santosh Parajuli[5] Liron Barak[2], Boping Chen[2], Jeffrey Eastlack[3], Scott Hauck[1], Shih-Chieh Hsu[1], Rohin Narayan[5], Allison Deiana[5]

[1] *University of Washington*
[2] *Tel Aviv University*
[3] *Michigan State University*
[4] *Westmont College and University of Pittsburgh*
[5] *Southern Methodist University*
[*] *zhixij@uw.edu*

Abstract: In this paper, we present an architecture of the algorithm processing platform (APP) that supports fast data transmission, data synchronization with different clock speeds, and communication for multiple APPs. We utilize HLS4ML and fwX into the APU development process to significantly reduce the complexity of algorithm design and improve the APU's performance, as demonstrated by our successful implementation of various machine learning algorithms with low latency of 1.5 microseconds and less than 5% resource utilization on the FPGA XCVU13P when using ML algorithms in the APU. Finally, we implement a Global Event Processor (GEP)-defined algorithm into the APU and examine its actual performance in experimental applications with supporting data.
Our work shows the potential of using machine learning in the APU for high-energy physics applications. This can significantly improve the performance of the trigger system and enable the ATLAS experiment to collect more data and make more discoveries. The architecture and approach presented in this paper can also be applied to other applications that require real-time processing of large volumes of data. [1]

THANK YOU!



JZ2,3,4,5,6 combine