τ Triggering with Deep Learning

Hadar Cohen

7/5/2020

Supervisor: Prof. Erez Etzion

TL;DR

Using deep learning methods to significantly improve hadronic tau trigger (L1Calo) performance at ATLAS

Outline

1. Physics motivation
2. The problem
3. The data (ATLAS Calorimeter)
4. Solutions
   a. CNN
   b. DeepSet
5. Results
6. Summary

Why Tau?

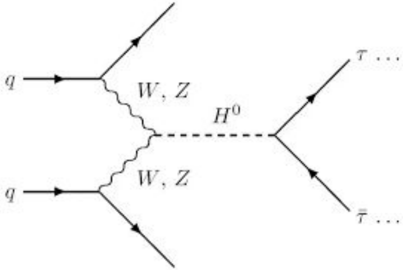
ELSEVIER

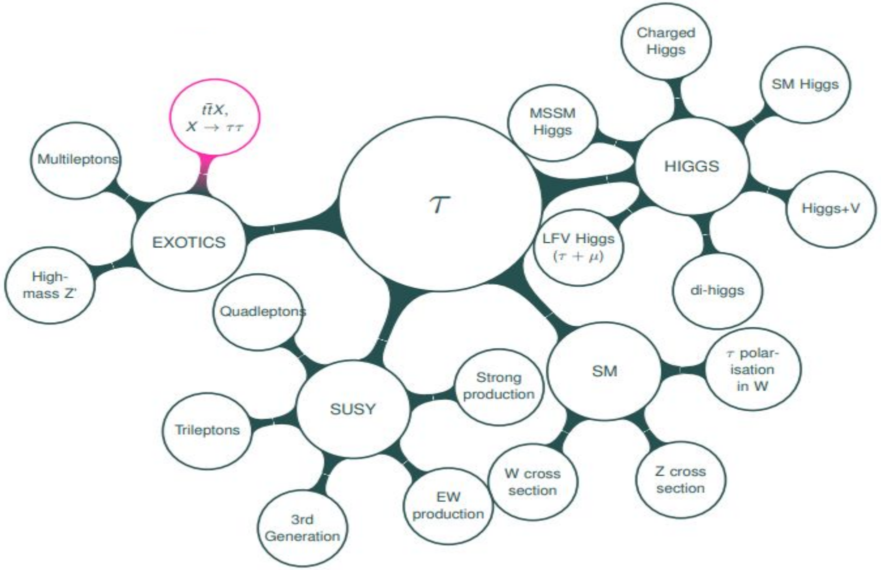
Physics Letters B
Volume 716, Issue 1, 17 September 2012, Pages 1-29

Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC ☆

Cross-section measurements of the Higgs boson decaying into a pair of τ-leptons in proton-proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector

M. Aaboud *et al.* (ATLAS Collaboration)
Phys. Rev. D **99**, 072001 – Published 10 April 2019





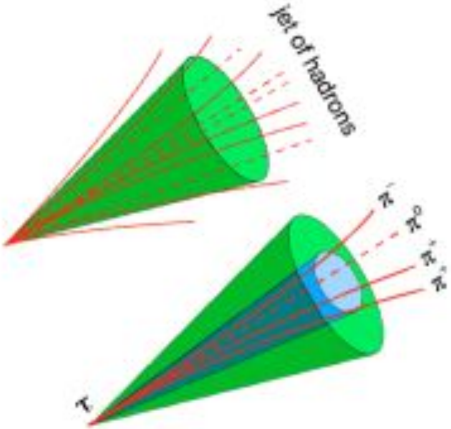
Triggering efficiently on hadronic τ leptons is crucial in order to achieve the physics goals of ATLAS:

- Measurements of Higgs coupling properties
- BSM Higgs search is limited to higher than 200GeV in ATLAS

Event Topology

- Leptonic decays: hard to distinguish from prompt e/μ - single track & short tau lifetime
- Identifying tau hadronic decays (65%) requires good understanding of the detector and event topology.
  - Provides narrower jets vs QCD wider jets but no unique enough

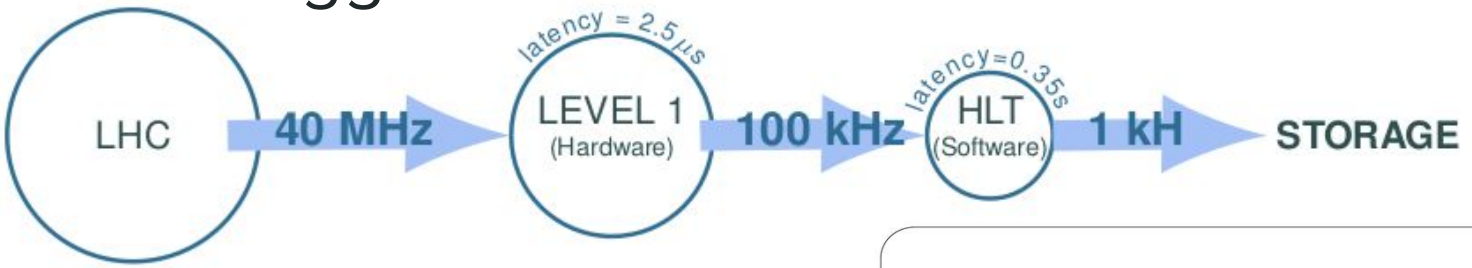
- Low track multiplicity
- Strong EM component due to π^0s in tau decays.
- Very challenging in the high luminosity environment



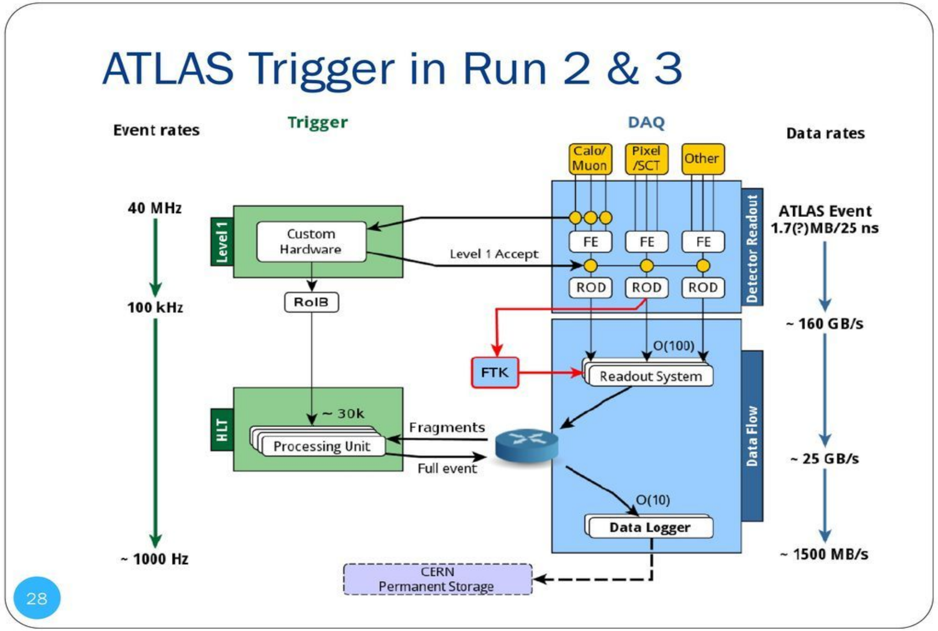
pdg

decay mode	fit result (%)
$\mu^-\bar{\nu}_\mu\nu_\tau$	17.3937 ± 0.0384
$e^-\bar{\nu}_e\nu_\tau$	17.8175 ± 0.0399
$\pi^-\nu_\tau$	10.8164 ± 0.0512
$K^-\nu_\tau$	0.6964 ± 0.0096
$\pi^-\pi^0\nu_\tau$	25.4941 ± 0.0893
$K^-\pi^0\nu_\tau$	0.4328 ± 0.0148
$\pi^-2\pi^0\nu_\tau$ (ex. K^0)	9.2595 ± 0.0964
$K^-2\pi^0\nu_\tau$ (ex. K^0)	0.0647 ± 0.0218
$\pi^-3\pi^0\nu_\tau$ (ex. K^0)	1.0429 ± 0.0707
$K^-3\pi^0\nu_\tau$ (ex. K^0, η)	0.0478 ± 0.0212
$h^-4\pi^0\nu_\tau$ (ex. K^0, η)	0.1118 ± 0.0391
$\pi^-\pi^-\pi^+\nu_\tau$ (ex. K^0, ω)	8.9868 ± 0.0513
$\pi^-\pi^-\pi^+\pi^0\nu_\tau$ (ex. K^0, ω)	2.7404 ± 0.0710

ATLAS Trigger

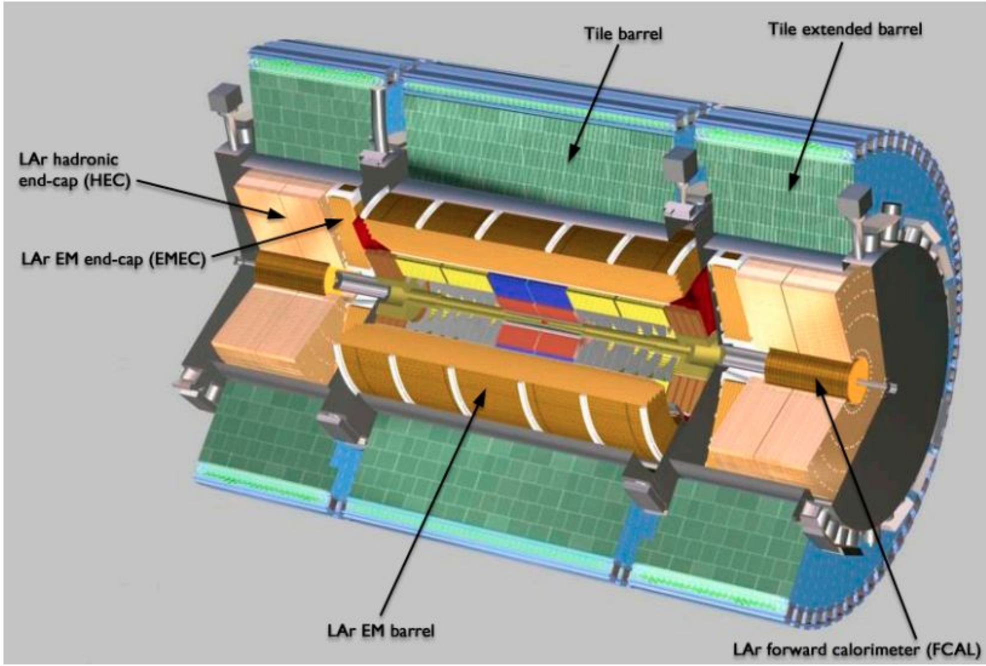


- Online selection is vital to collect the most interesting collisions out of the large data volume.
- The ATLAS experiment utilizes a trigger system that consists of a hardware L1 and a software based HLT to reduce to rate to a mangable one.



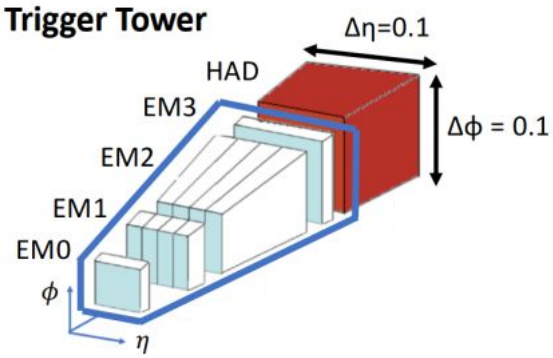
ATLAS Calorimeter System

- The ATLAS calorimeter system consists of two components, LAr and Tile calorimeters.
- Covers the barrel regions + endcaps up to $|\eta|=4.9$
- Increase of luminosity and pileup, degrade the calorimeter resolution and the isolation of single particles
- We need to explore new approaches to keep the trigger thresholds as low as possible.



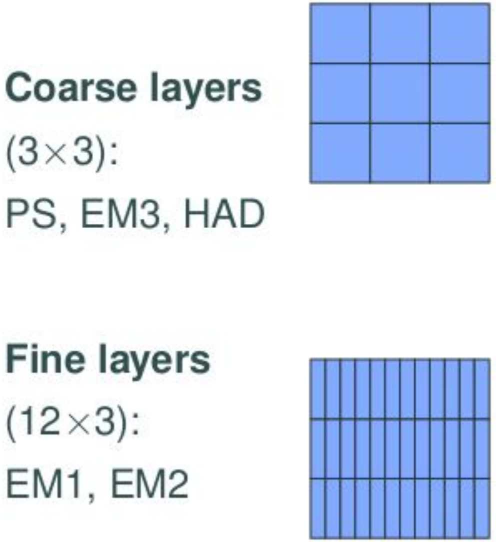
L1Calo

- L1 trigger based on calorimeter data - LAr and Tile systems
- Increased granularity in Run3 upgrade
- FPGA based hardware



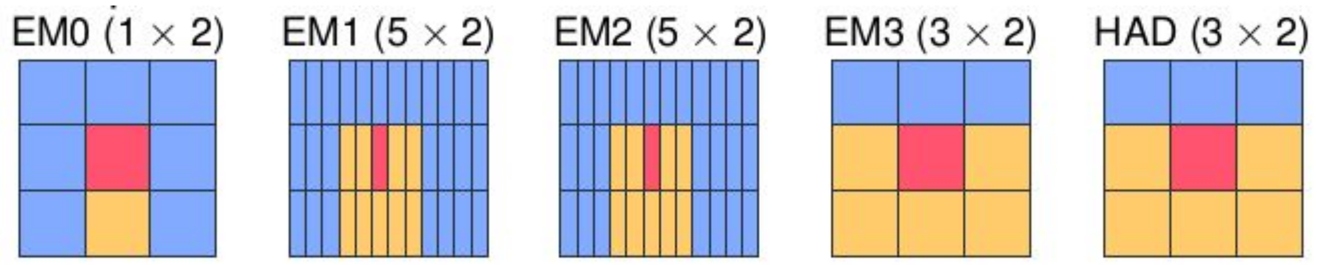
Data

- Raw ATLAS calorimeter energy deposits - E_T
- Five layers(99 cells): EMCALO + HADCALO
- MC: Z->ττ vs di-jet QCD



Benchmarks Algorithms

- Predefined recipe using different shapes/window over the different layers

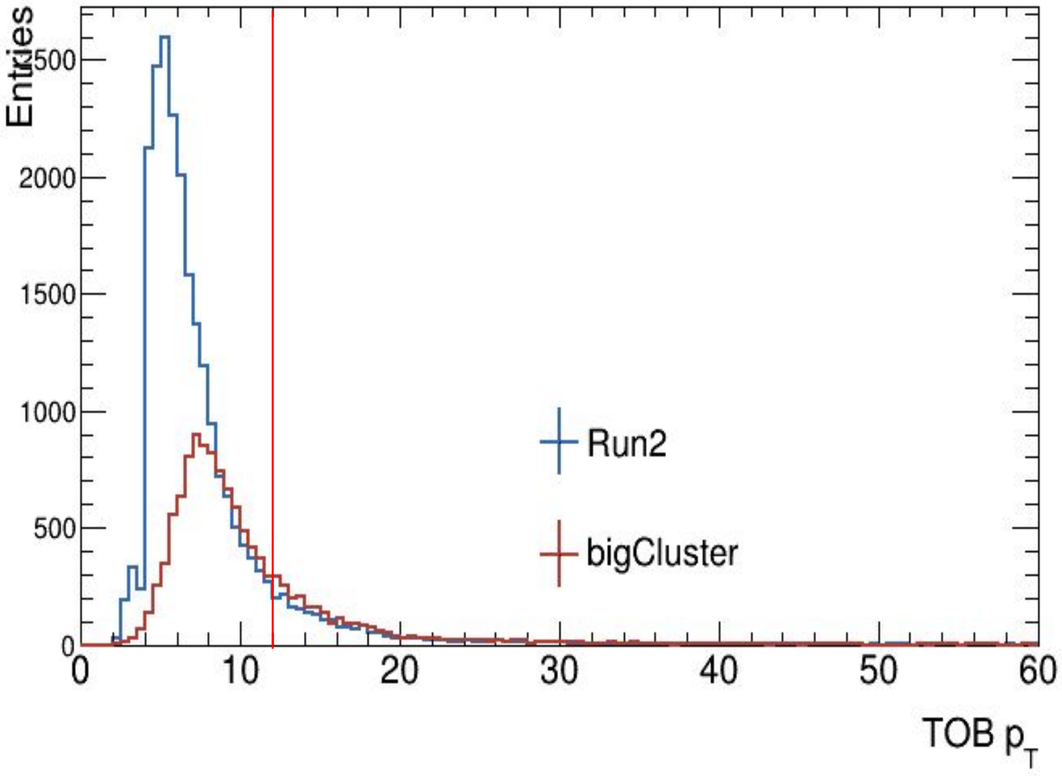


- Searching for hottest cell in EM1+EM2, clustering and adding from adjacent layers around it
- Try to evaluate a tau energy deposit in order to do a threshold trigger

Fake Rate Constraint

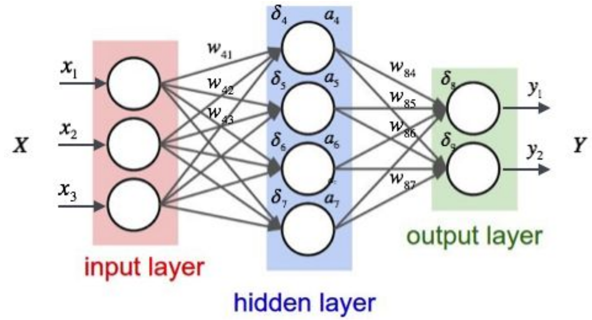
- We are strictly limited by trigger rate, meaning every decision making algorithms we design must not yield ou a higher fake rate than the current Run2 one.

Fake Rate (di-jets)



ML

Why Use Machine Learning? (UAT)

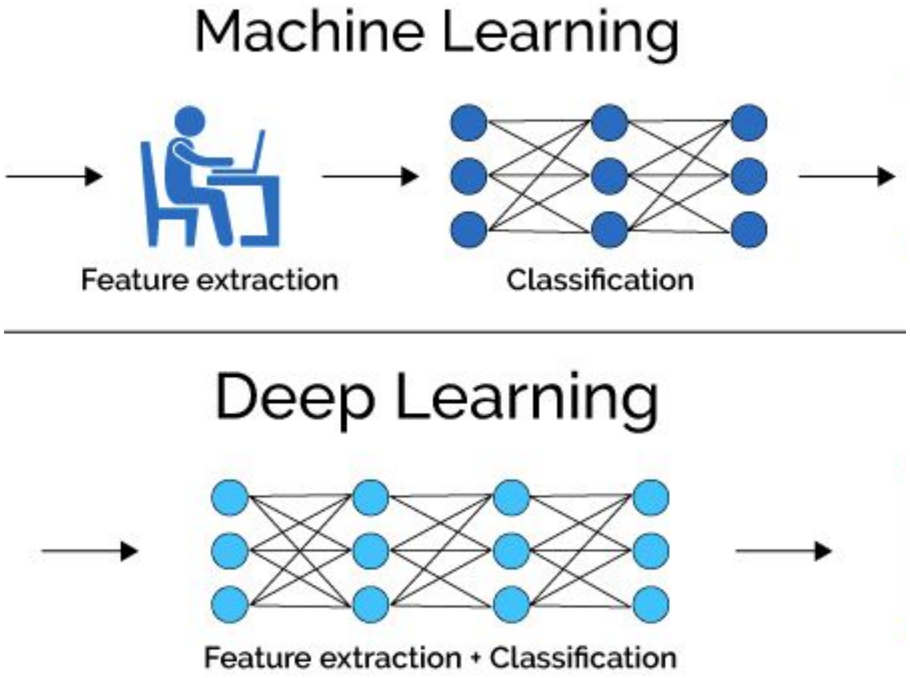


From wikipedia:

"..the universal approximation theorem (UAT) states that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions..."

If we accept most classes of problems can be reduced to functions, the UAT statement implies a neural network can, in theory, solve any problem.

All that is left for us, is to find a function that maps between the data to a probability of that data being a tau. ,



Why is Deep Learning so successful?

- **The network extracts features directly from the data.**
  - As opposed to feature engineering with classical machine learning.
- In our case we wish it to learn features from the geometrical structure of the data
- The input? Only raw data from the calorimeter itself

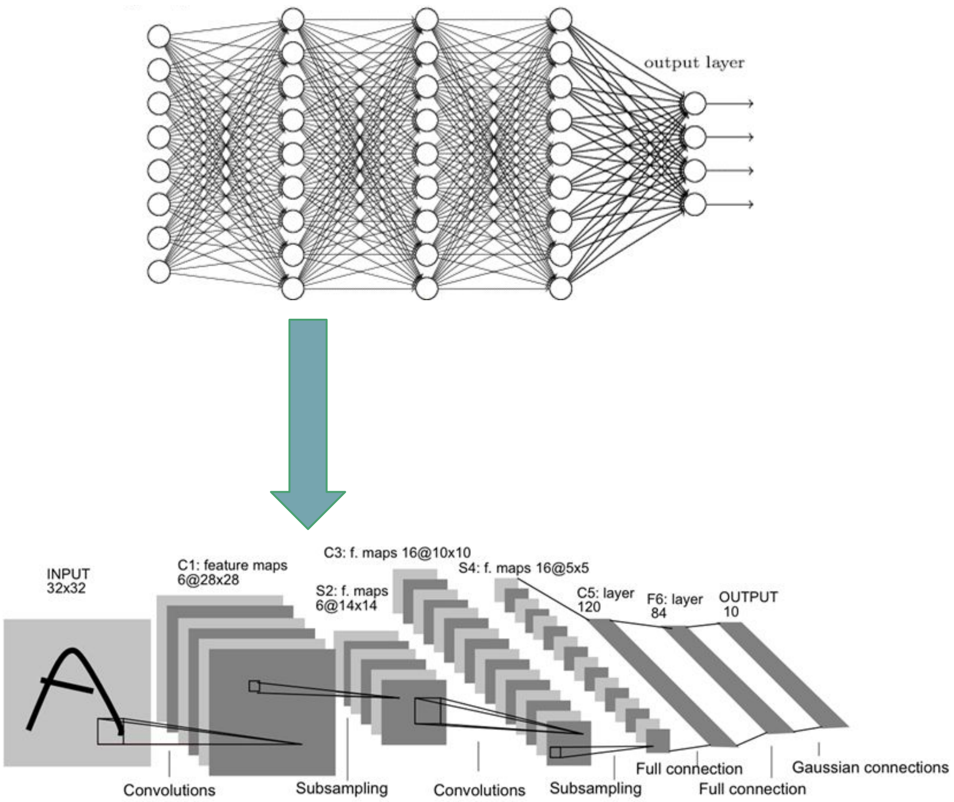
Solutions

CNN & DeepSets

Convolutional Neural Net (CNN)

What and Why?

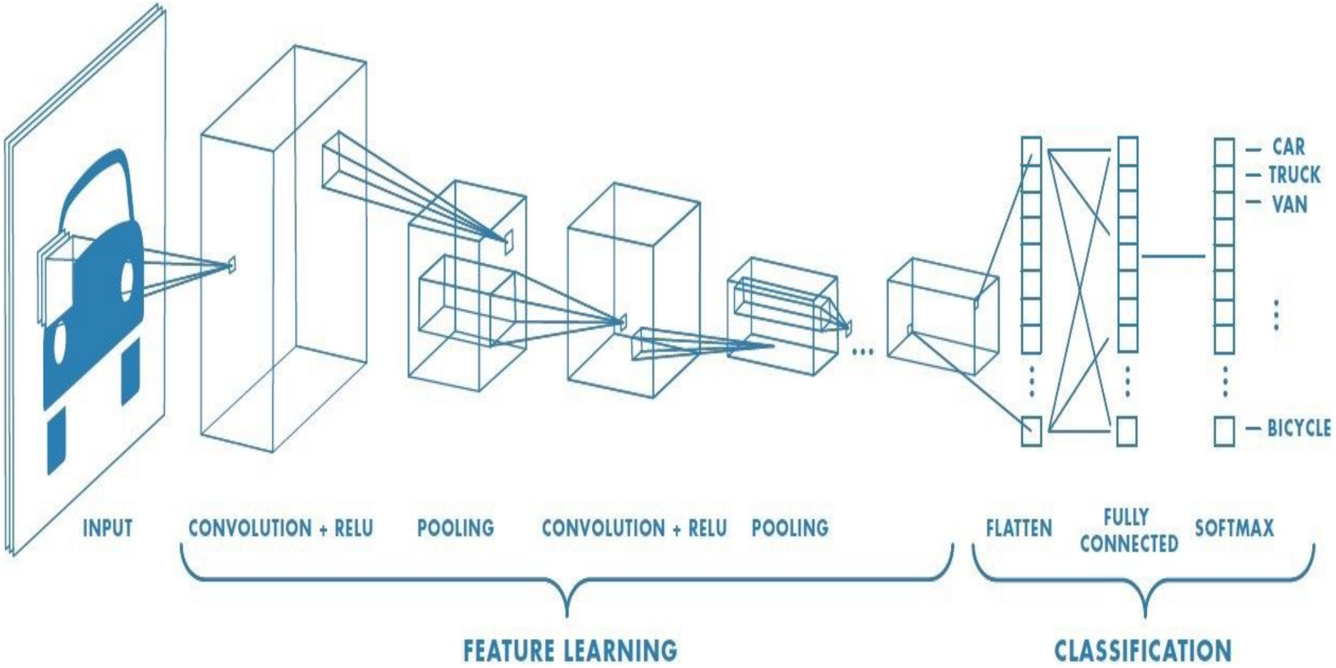
- Convolutional Neural Nets(CNNs) use convolution operation to extract information from the data.
- Main usage - Image processing
    - Go from single values to a 2D image
- Weights -> Conv filters
- Highly successful in learning complex shapes



CNN Architecture

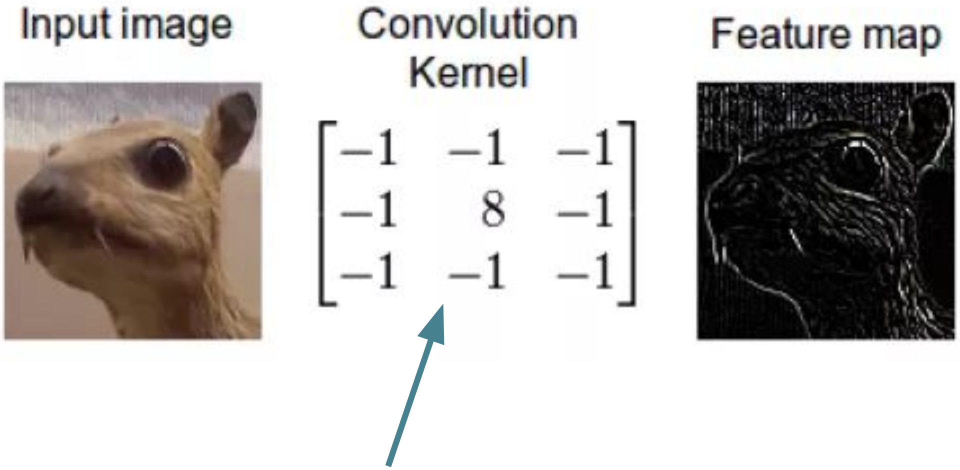
As with any DeepLearning method it contains two parts:

1. Feature learning: obtains meaningful information on different parts of the image. Different filters look for different structures
2. Classification: a fully connected neural net that operated on the extracted information to provide a classification.

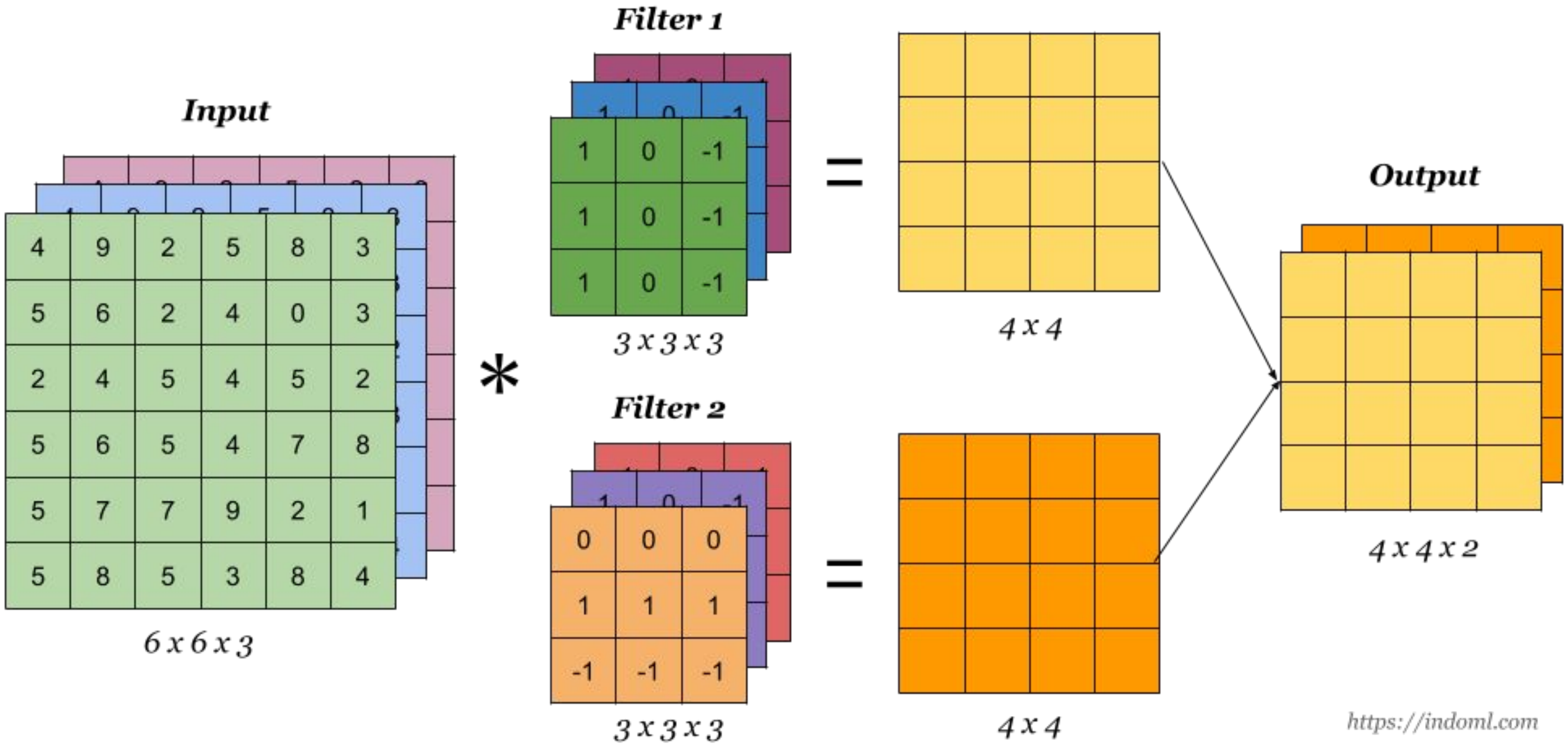


Feature Extraction

- An image is nothing but a matrix of pixel values
- We need to construct an algorithm that can operate on such input and provide a meaningful output.
- Conv operation over a picture yields a feature map
- Over time the network learns the best conv kernel to the problem



Learnable parameters of the model



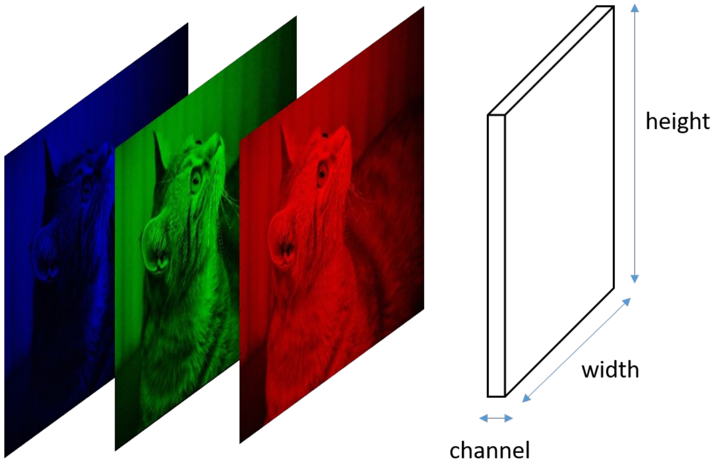
- The convolution step in a any CNN architecture (there are many) is to perform a 2D Convolution over the the 3 color channels
  - Meaning we take **N** 3x3x3 filters and apply them over the image.
  - These filter are the **weights** we learn over the process of training
  - Then we continue to non-lineraity and more

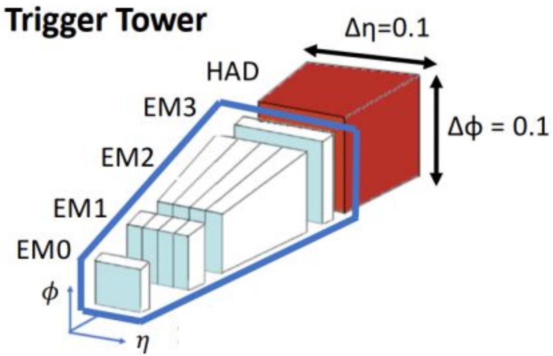
Image Like Representation

- A normal picture is a rank 3 tensor with a shape of

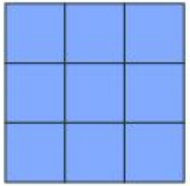
  Channels x Height x Width

- Grayscale images have only 1 channel while color images have 3 (RGB)





Coarse layers
(3×3):
PS, EM3, HAD



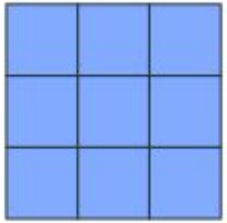
Fine layers
(12×3):
EM1, EM2

Preprocessing

Coarse layers

(3×3):

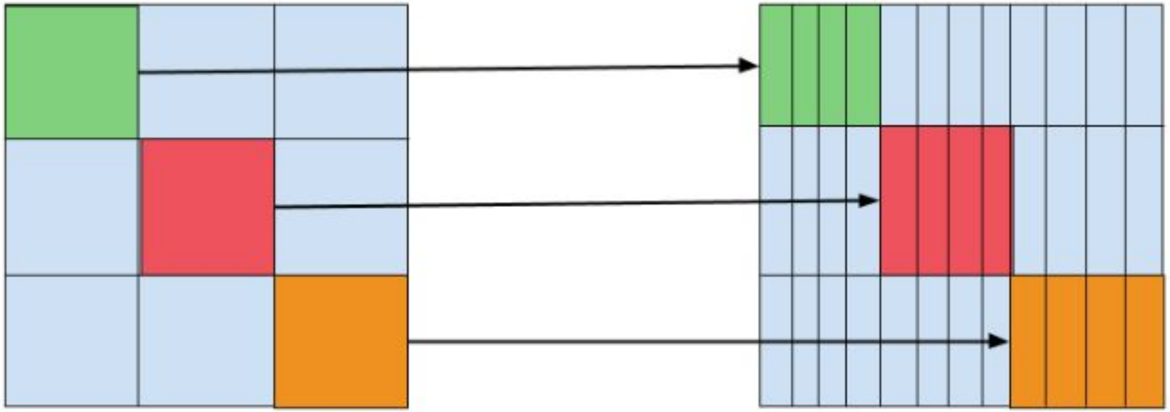
PS, EM3, HAD



Fine layers

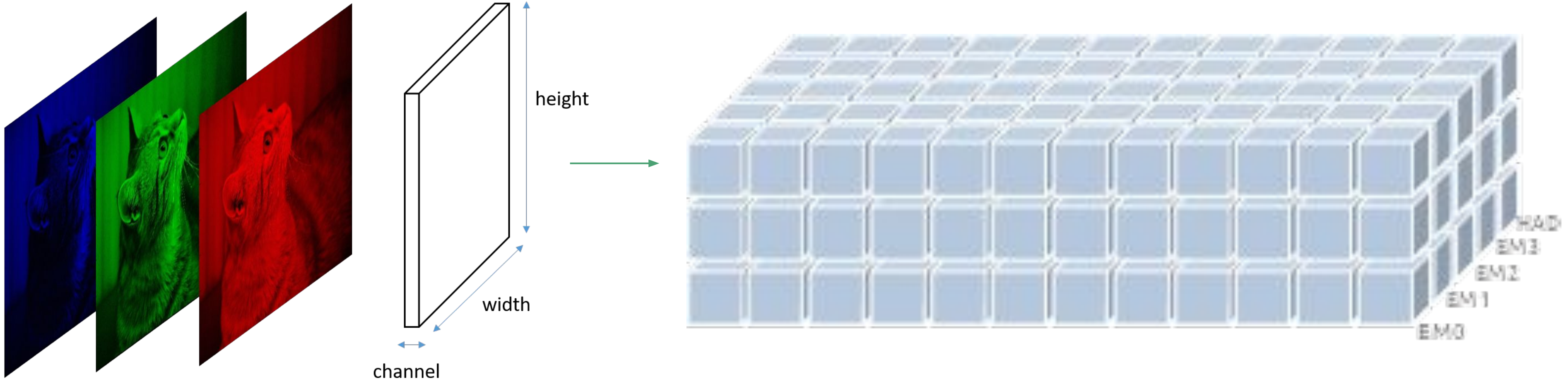
(12×3):

EM1, EM2



- “Stretching” coarser layers to have the same shape as finer layers
- Standard ML steps: filtering, normalizing, etc..

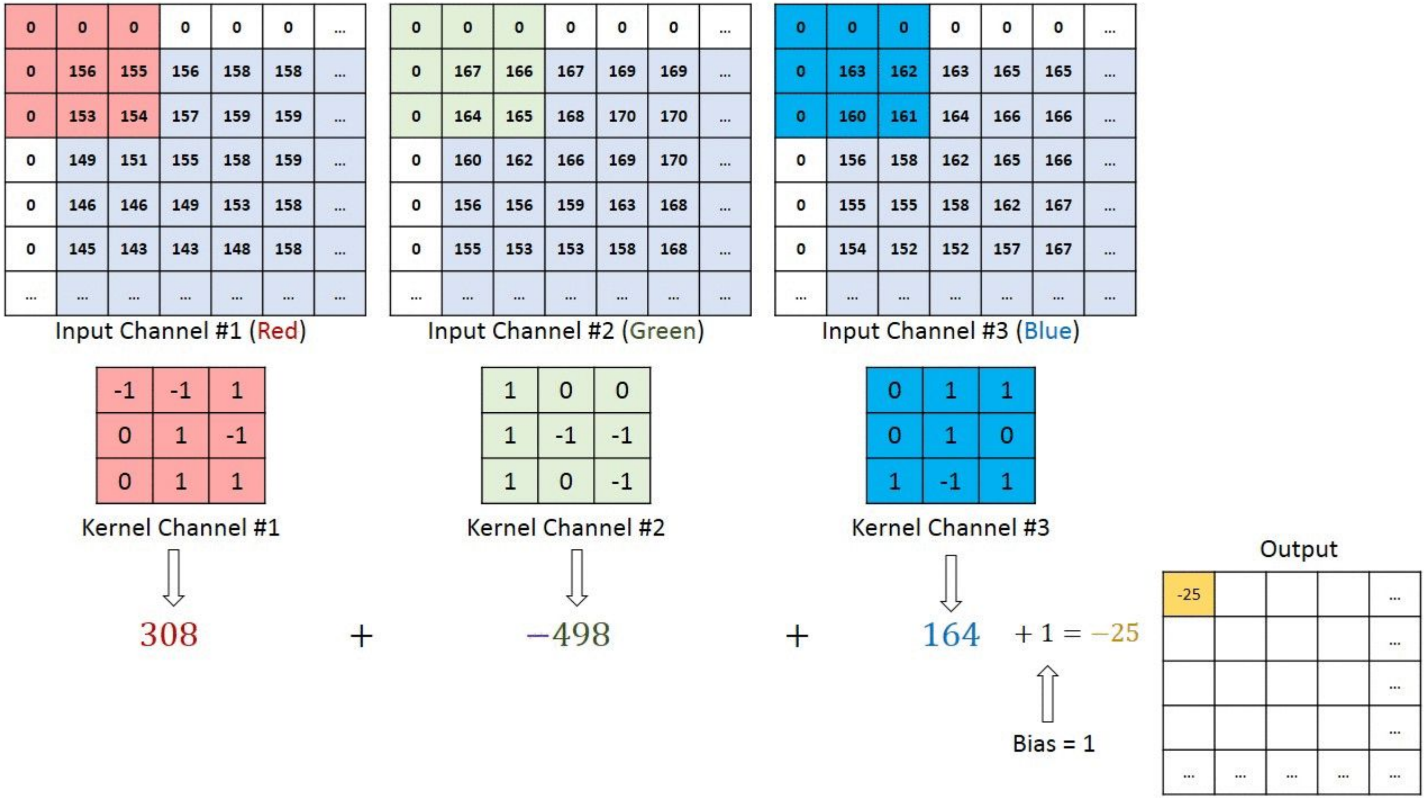
Image Like Representation



conv2d(3,nf) ----------------------> conv2d(5,nf)

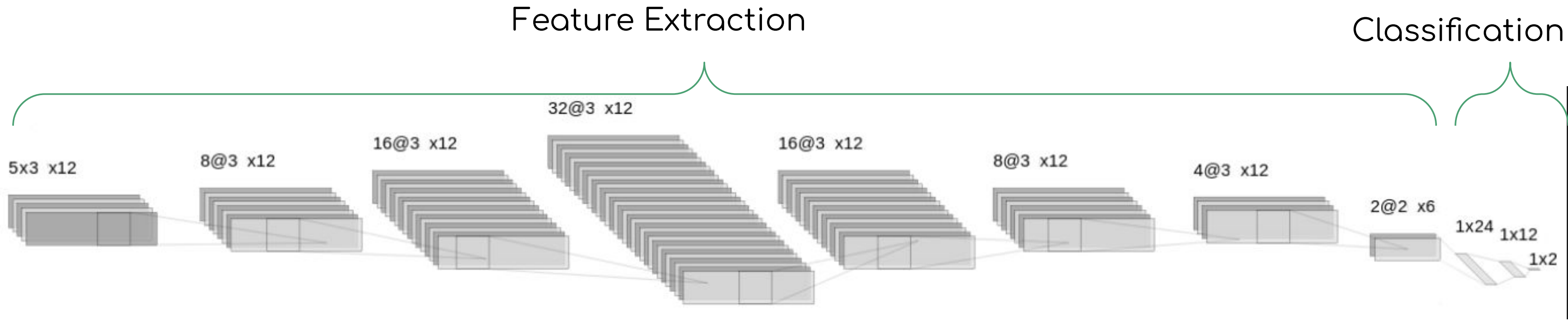
Intuition: if we can look at the 3D grid of energy deposits in the calorimeter as an image, we can find a pattern to distinguish between tau and a jet.

CNN



Intuition: if we can look at the 3D grid of energy deposits in the calorimeter as an image, we can find a pattern to distinguish between tau and jet events.

CNN for Tau Architecture



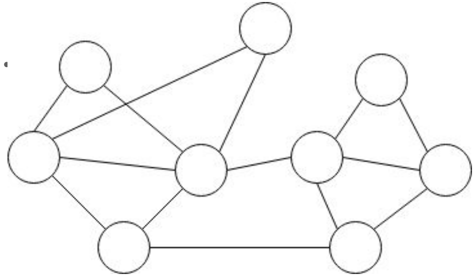
- Eight blocks of conv operations (each containing several + residual)
- Transforming the 5x3x12 calorimeter input into a single prediction

DeepSet

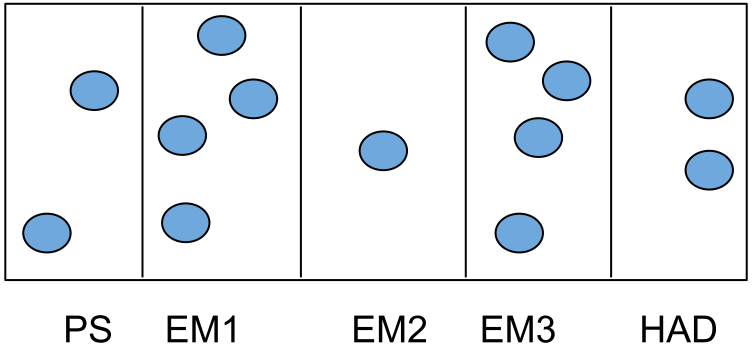
Data Structure

- CNN representation was a very sparse one, only 10% active.
- Lets try and look at the data from a different perspective:

  as nodes in space - a **graph**



- But, how should one define the graph’s edges? What is the adjacency matrix?
  - Is the PS(EM0) connected to the Hadronic calo?
  - Are all layer nodes (cells) fully connected?
  - Do we only connect nodes between layers?

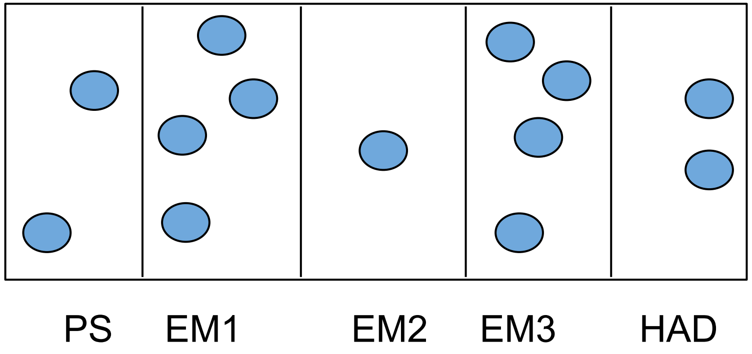


Graph → Set

- But, within a layer there is **no temporal order**, no single cell came before nor after a different one, there is no real meaning to some edges

  -> Thus, an event type should not be depended on the cells order (when evaluating) but merely on their properties and their existence.

- Taking that into account, we look at our data as a **SET.**



Set Representation

- We define our event as follow:
  - Each cell with an energy deposit is included in the set
  - Every member of the set has the basic raw features:
    - Energy
    - Coordinates - simple location in the calorimeter grid with the z-axis as layer number [0-4]
- But! A set is not fixed size, could be 15 or 33 cells.
- Neural nets work on a fixed size input. We need to adapt
- Our primary goal is to find a function that maps our input into a prediction
  - We need to find a **set function**

e	x	y	z
1.55	4	1	1
0.875	5	1	1
0.275	6	1	1
0.275	0	0	2
0.375	1	0	2
0.8	5	0	2
0.525	6	0	2
0.2	8	0	2
0.35	2	1	2
2.275	4	1	2
0.3	1	2	2

https://arxiv.org/pdf/1703.06114.pdf

DeepSets

Theorem 2 *A function $f(X)$ operating on a set X having elements from a countable universe, is a valid set function,* i.e., ***invariant*** *to the permutation of instances in X, iff it can be decomposed in the form $\rho\left(\sum_{x \in X} \phi(x)\right)$, for suitable transformations ϕ and ρ.*

- Replacing φ and ρ by universal approximators (UAT) leaves matters unchanged. Then, it remains to **learn** these approximators

$$f(X) = \rho\left(\Sigma_{x \in X} \phi(x)\right)$$

https://arxiv.org/pdf/1703.06114.pdf

DeepSets Architecture

- Each instance $x_m \forall 1 \leq m \leq M$ is transformed (possibly by several layers) into some representation $\phi(x_m)$.
- The addition $\sum_m \phi(x_m)$ of these representations processed using the ρ network very much in the same manner as in any deep network (*e.g.* fully connected layers, nonlinearities, *etc*).
- Optionally: If we have additional meta-information z, then the above mentioned networks could be conditioned to obtain the conditioning mapping $\phi(x_m|z)$.

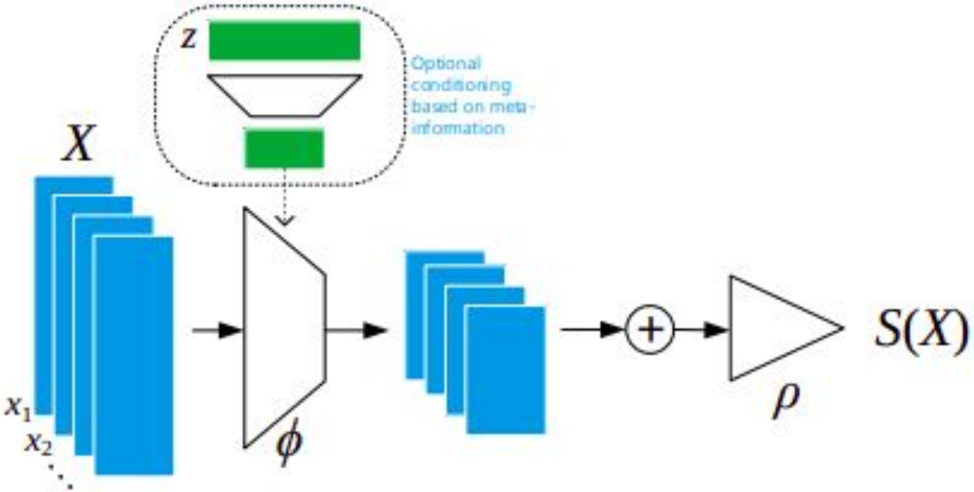


Figure 5: Architecture of DeepSets: Invariant

$$f(X) = \rho\left(\Sigma_{x \in X} \phi(x)\right)$$

DeepSets for Tau Trigger

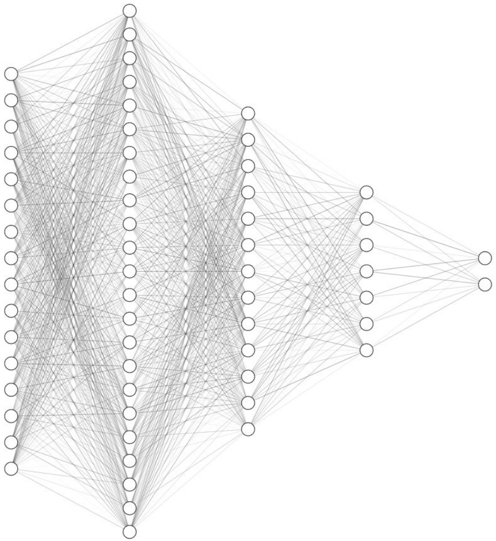
e	x	y	z
1.55	4	1	1
0.875	5	1	1
0.275	6	1	1
0.275	0	0	2
0.375	1	0	2
0.8	5	0	2
0.525	6	0	2
0.2	8	0	2
0.35	2	1	2
2.275	4	1	2
0.3	1	2	2

$$\overbrace{\text{Feature Extraction}}^{\phi(x_m)}$$

DeepSet Layer - 1D Conv for 256 features representation

$$\sum_m \phi(x_m)$$

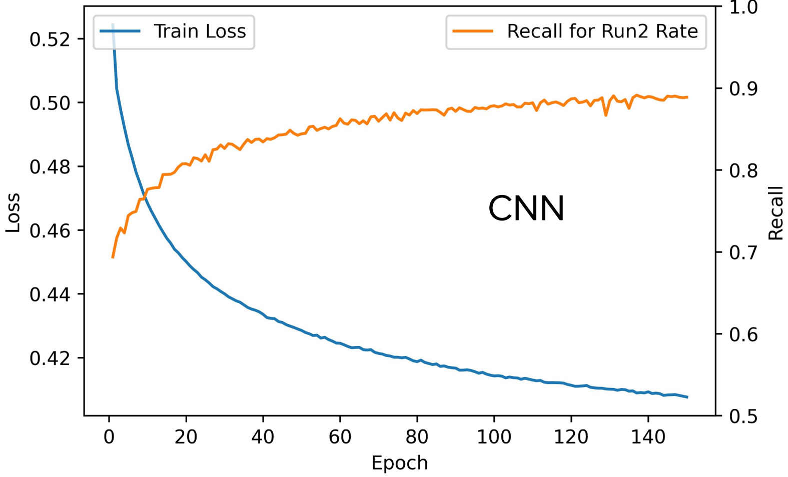
$$\overbrace{\text{Classification}}^{\rho}$$

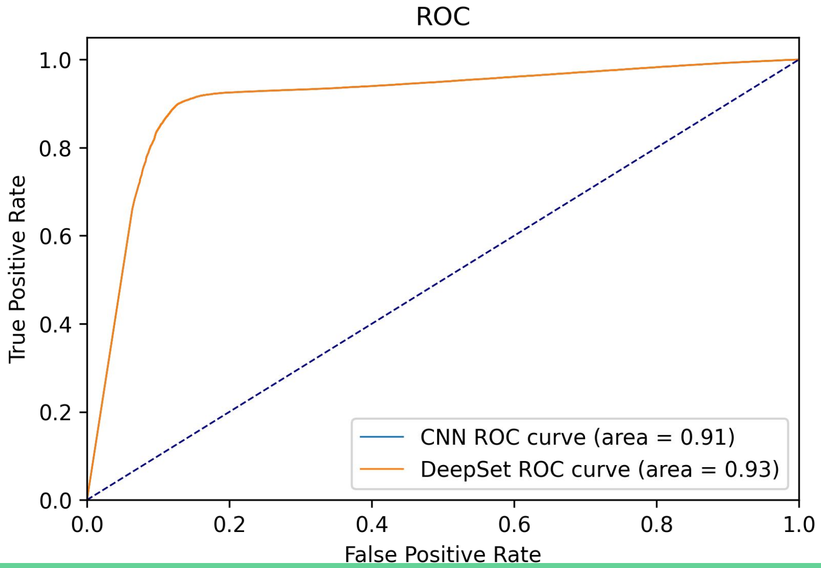


Results

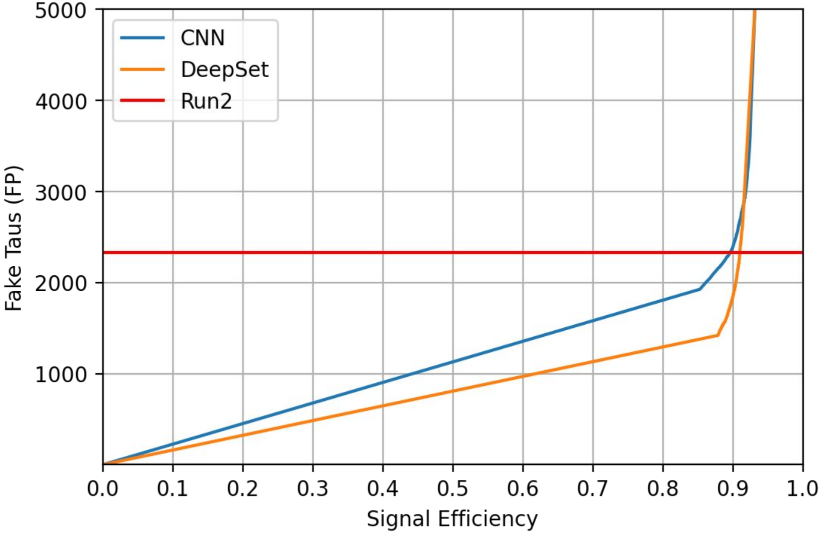
Training

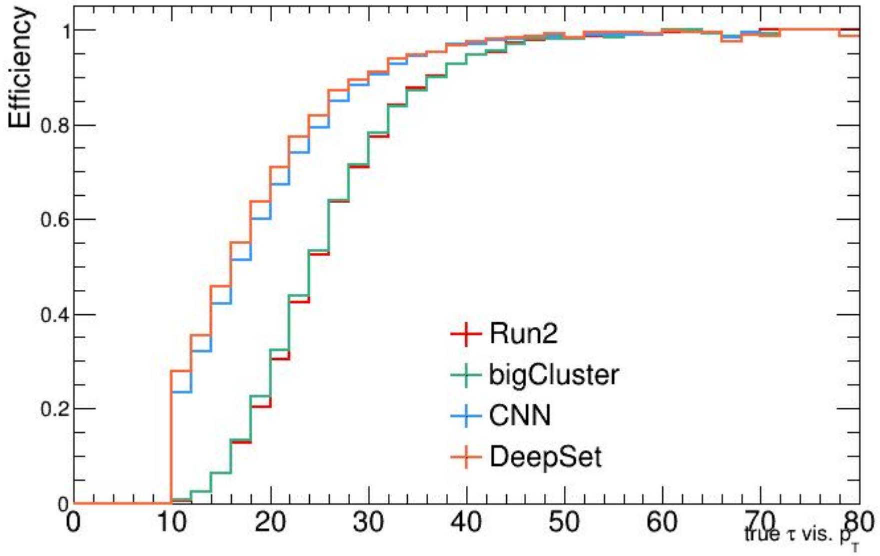
- Both methods, CNN & DeepSet were trained over nvidia's Tesla K80 GPUs using PyTorch
- Training metric: recall with a fixed Run2 fake rate
  - We are strictly limited by the fake rate, meaning how much tolerance we have for False-Positives
  - Prior to training, we estimate the Run2 fake rate and set that as a limit while training
  - We maximize the number of signal events (TP) while keeping the same fake rate (FP)
- Our final evaluation is a Turn On Curve - Efficiency vs True tau pT





Final Results

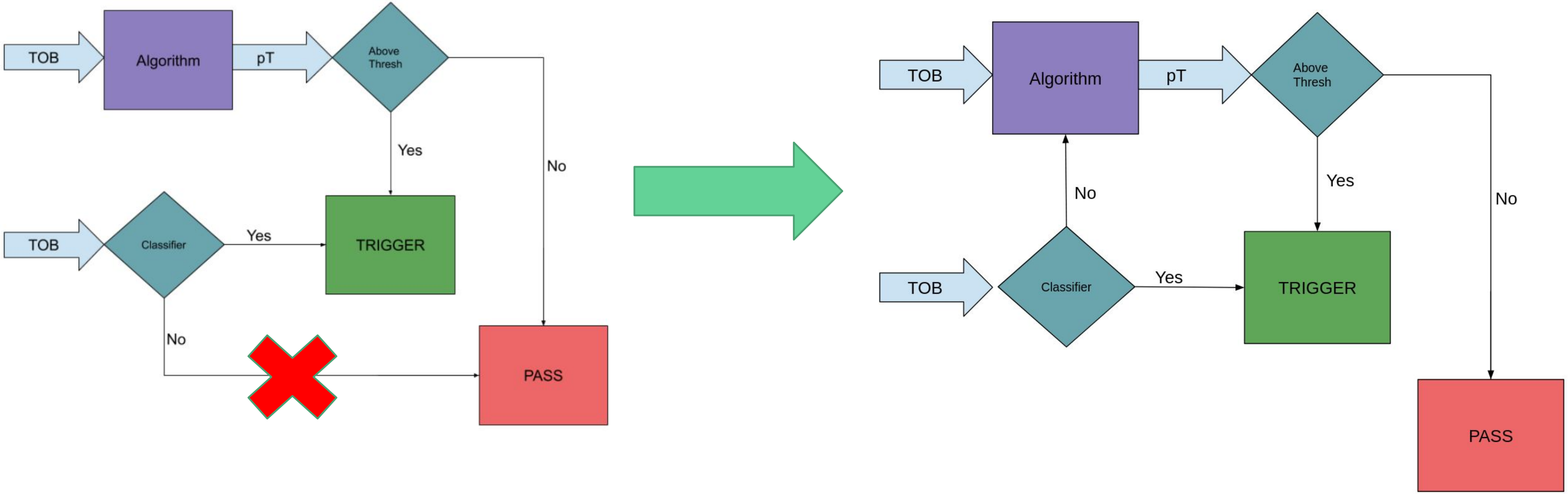




Selecting the highest signal efficiency with equal Run2 fake rate

Evaluating the efficiency w.r.t the truth tau pT. Significant improvement at low pT region

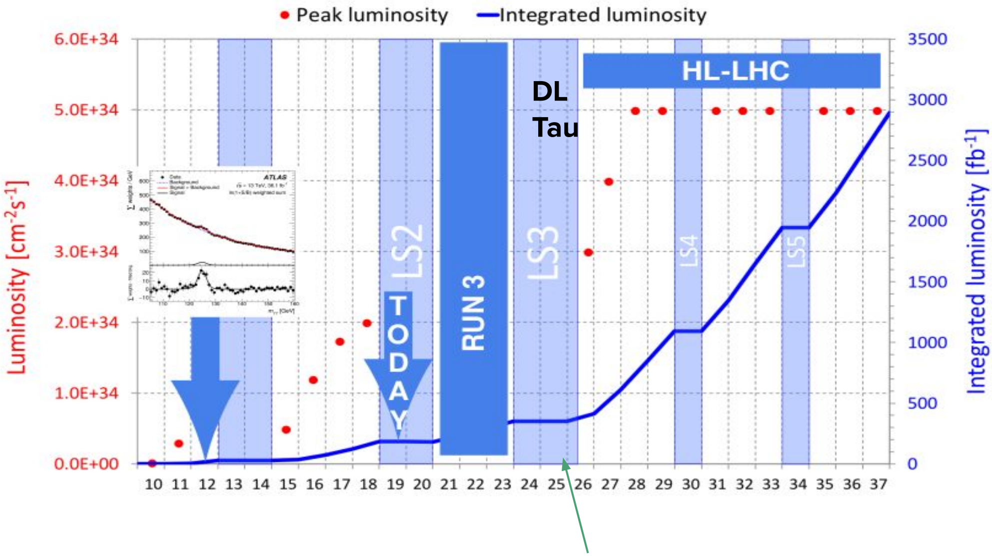
Hybrid Method



- Second chance - for any non triggered event higher than a certain pT, try using a the usual route
  - Pros: Safety net for misclassified higher pT events
  - Cons:
    - Higher rate - meaning we need to raise the classifier cut, thus lowering the signal efficiency
    - More calculation = higher latency

Future Prospects

- Optimize solutions
  - Slowly and carefully remove layers to minimize the computational cost
  - Minimize representation
- FPGA Implementation
  - ATLAS Phase I hardware is fixed with a Xilinix FPGA. Implementing such CNN/DeepSet within the latency constraints would be a significant trigger improvement
- Jet Physics
  - Explore the use of DeepSets within the realm of jet physics substructure



Run3 Hardware is fixed.

Best case will be for Phase-II, Run4 Using specific hardware..

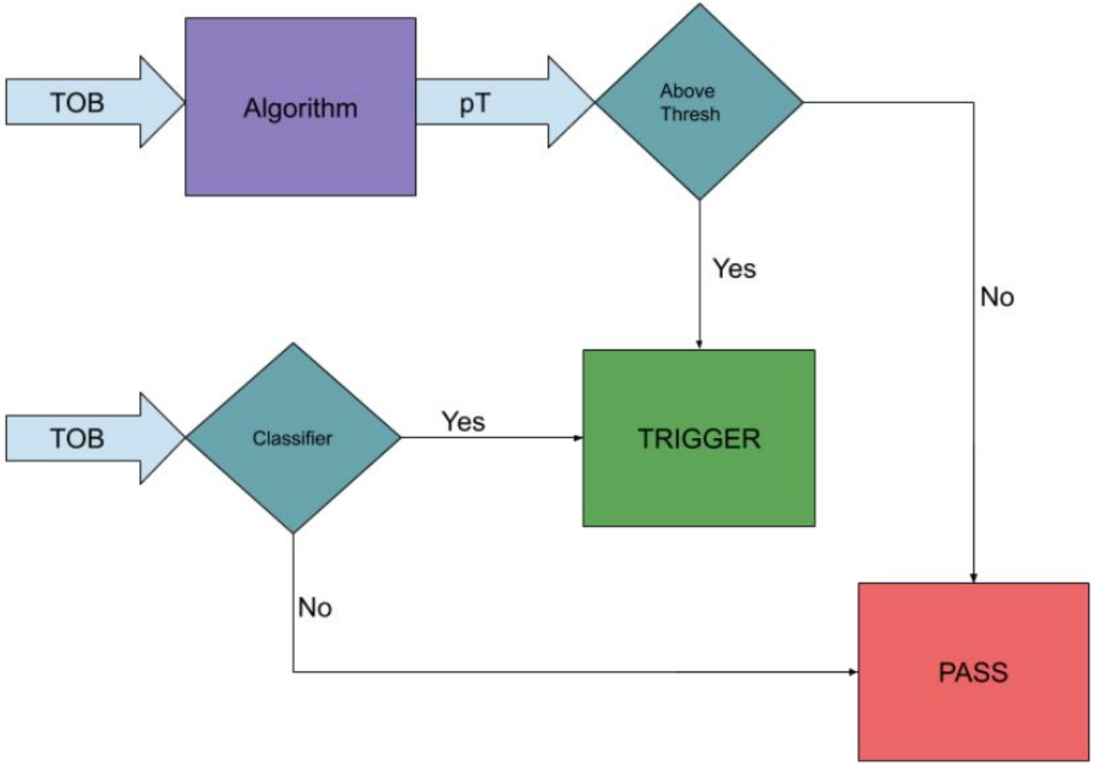
Summary

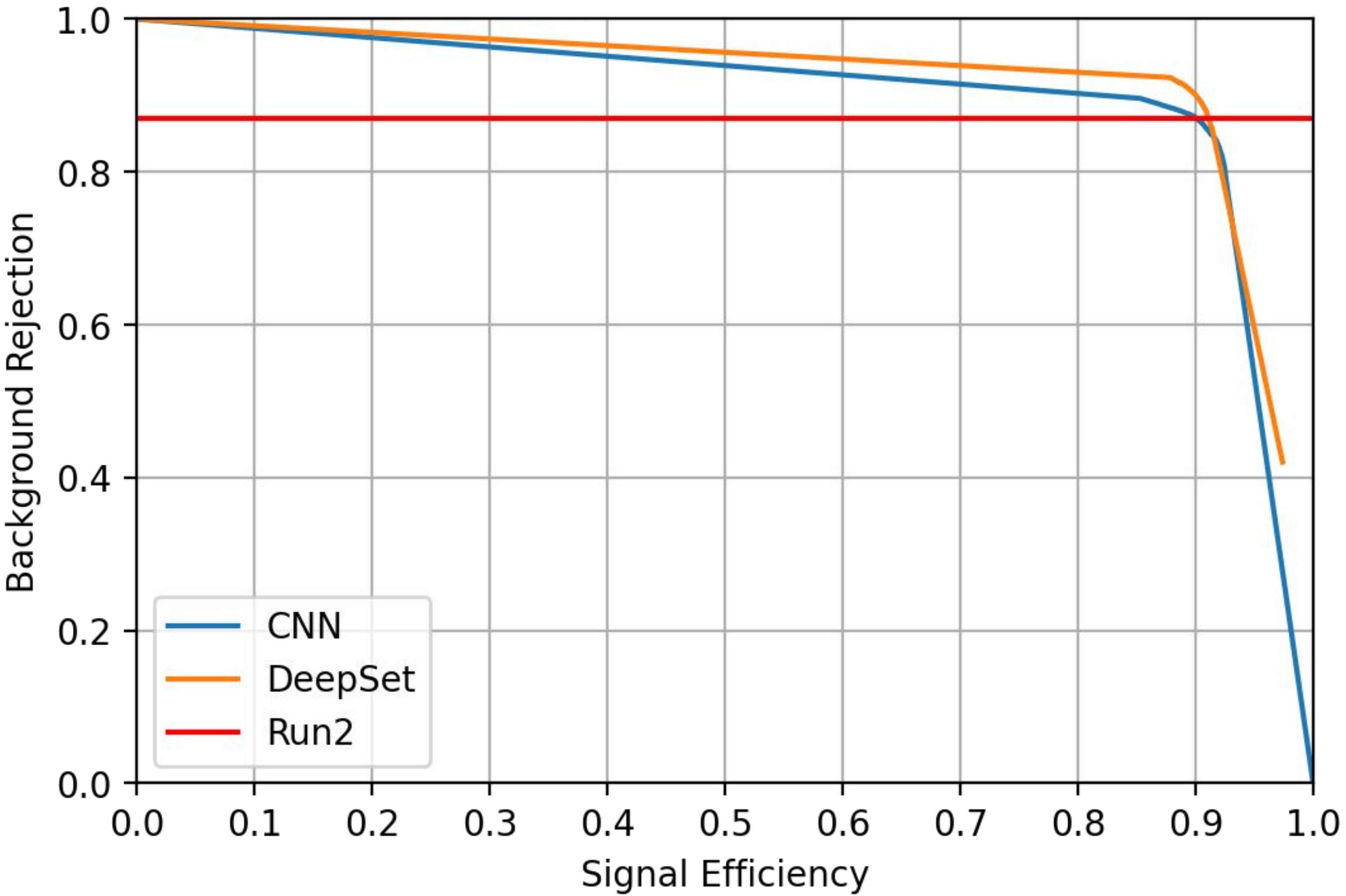
- L1 upgrade will help to cope with increase in luminosity and pileup in Run3
- Significant improvement in lower pT regions
- Probably no near future implementation due to latency issues.
- Real ATLAS MC
- Terrific problem

Hope this will **trigger** you to look for solutions in the other fields

Thank you.

New Way of Triggering





1.0
0.8
0.6
0.4
0.2
0.0
Background Rejection
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Signal Efficiency
CNN
DeepSet
Run2