Fast Deep Learning on FPGAs for the Phase-II L0 Muon Barrel Trigger of the ATLAS Experiment

ACAT2019 - Saas Fee
March 13th 2019

Simone Francescato on behalf of the ATLAS Collaboration

Motivations

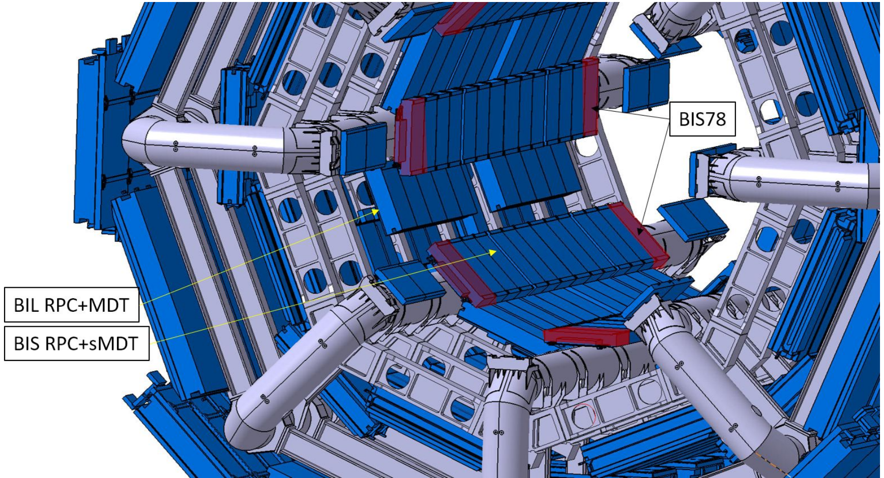
ATLAS Level-0 Muon Trigger will be fully upgraded for HiLumi-LHC

Machine parameters:

- Higher pile up: 30-40 → 200
- Higher lumi: (2 → 7.5) × 10^{34} $cm^{-2}s^{-1}$
- Higher trigger rate: up to 600 Hz/cm^2

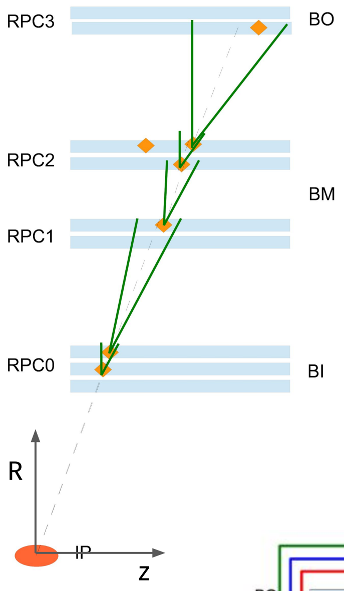
ATLAS upgrade:

- New Sector Logic
  - FPGA based system
    - Virtex UltraScale+ XCVU13P
    - System Logic Cells (K): 3780
    - Memory (Mb): 455
    - GTY Transreceivers (32.75 GB/s): 128
    - I/O Pins: 832
- New trigger station
  - New RPC layer
- Improved trigger algorithm
  - Must be extremely fast and flexible
  - NN as a valid opportunity
  - Possibility to develop dedicated triggers non-pointing and displaced muons



ATLAS-TDR-026

Standard trigger strategy



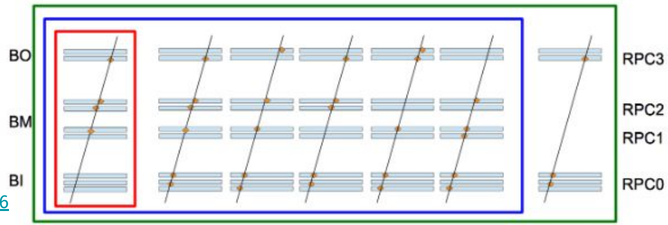
Standard algorithm:
- Check for coincidences in encapsulated windows

👍 Stable and reliable
👍 Good performances

👎 Coinc. windows need to be tuned “by hand”
- Windows depends on needed momentum threshold
- Strong dependency on coinc. schemes and geometry

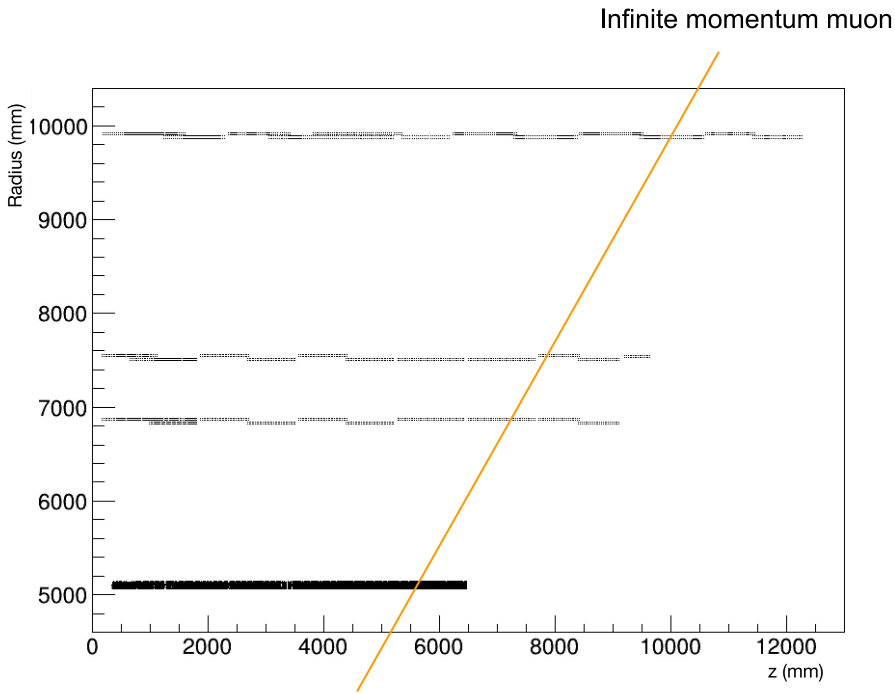
👎 Assume pointing to the primary vertex
- Displaced vertex decays?

👎 It does not give directly a momentum estimation



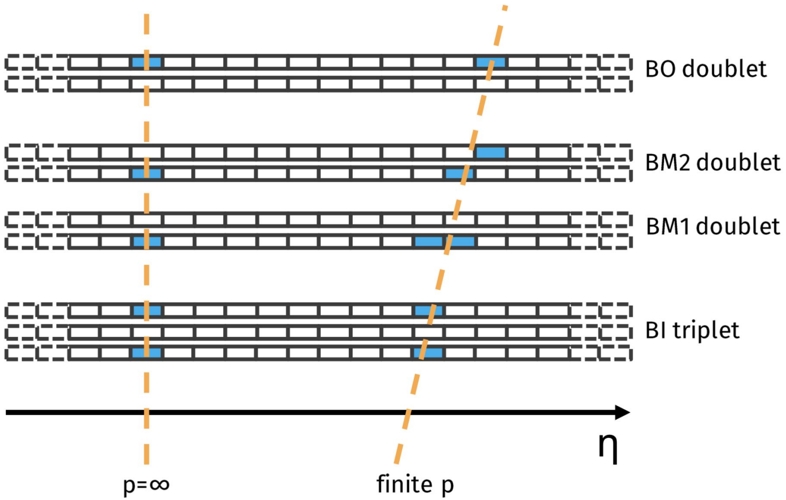
ATLAS-TDR-026

RPC Detector info

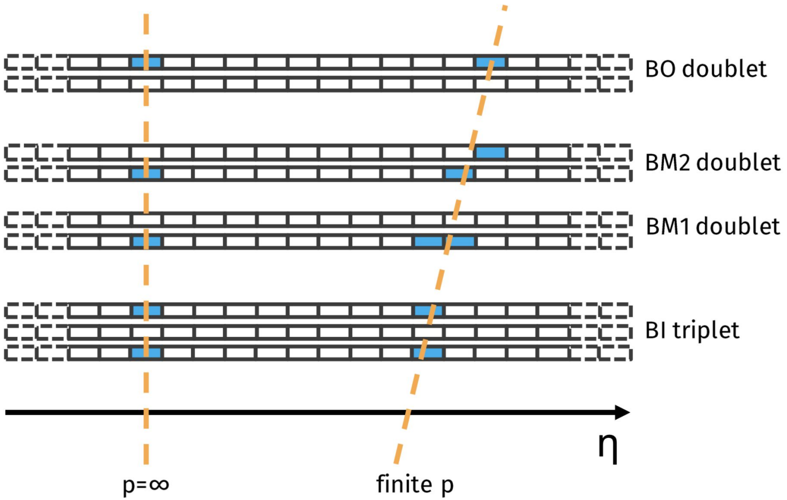


Sector =
- Divide ATLAS in 2 sides
- Divide ϕ coordinate in 16 slices
- Take all the strips of all the RPC

Build a η vs. layer map of the RPCs



η vs layers mapping



Mapping prevents geometrical dependance:

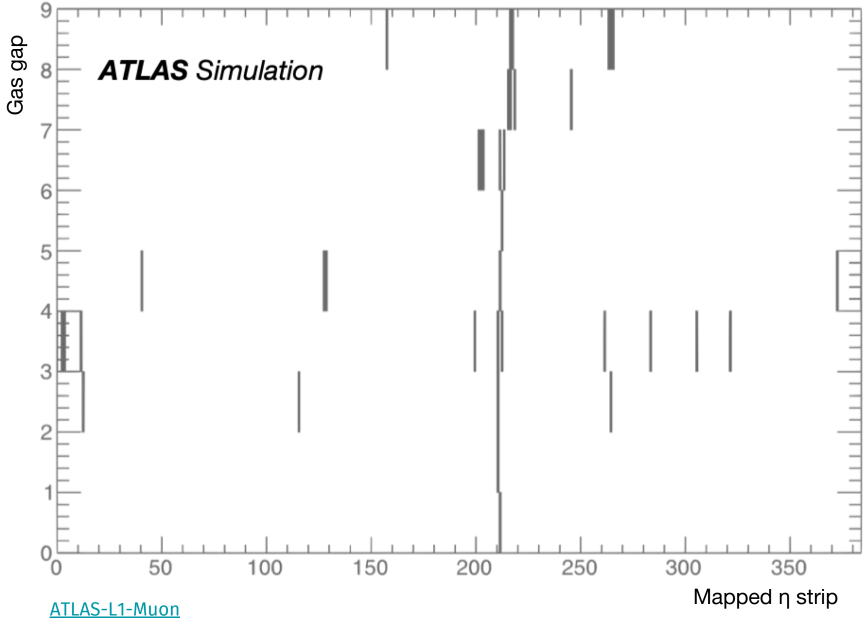
- Holes
- Chambers overlaps

Note that an **infinite momentum muon** is always a **vertical line**, independently on η

Strip mapping allows to increase/decrease algorithm granularity:

- To each real strip corresponds a dummy strip/bit
- Minimal case: #bits=#strips

From RPC Maps to Images



ATLAS-L1-Muon

Example of a muon with p_T=19 GeV + noise

We can build image for a NN:

- Minimal case 384x9 pixel
- a muon appears in the image as a straight line

We want to do regression on muon p_T

A CNN is a promising solver:

- “Muon tracks” are invariant under η translation

Problem:

- NN coping with highly sparsified images

Output targets

The Level-0 trigger is required to pass out up to 3 muon candidates for final trigger decision and for each one an estimate of their positions

A CNN can be used to:

- Trigger a muon candidate (=at least one with p_T>threshold)
- Do **regression** on leading muon p_T and η
- Do **regression** on subleading muon p_T and η → BONUS!
- Check (**classify**) if there are more than 2 muons → BONUS!

A NN perform a real regression on the track at L0:

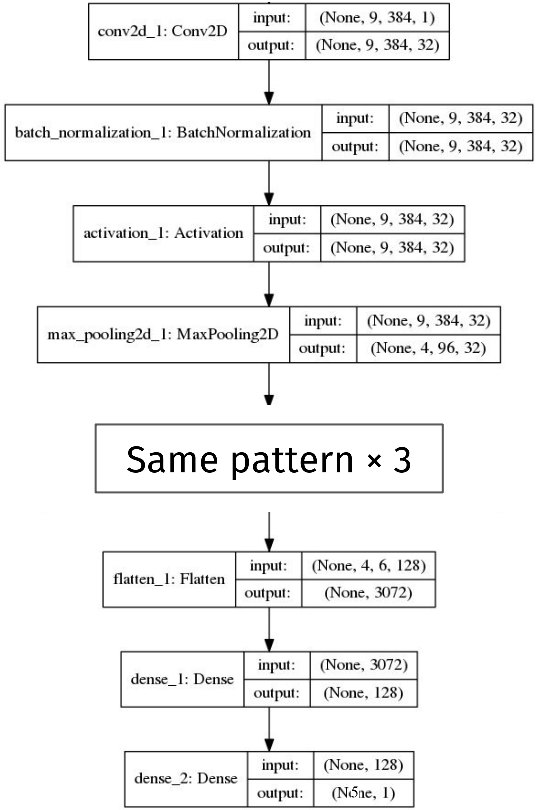
- Standard trigger can only give a rough estimate of p_T

We train a CNN with a 5D output

$$(p_T^{lead} \quad \eta^{lead} \quad p_T^{sublead} \quad \eta^{sublead} \quad n^{muons}) \longleftarrow$$

0 → no muons
1/2 → one/two muons
3 → more than 2

Floating Point CNN structure



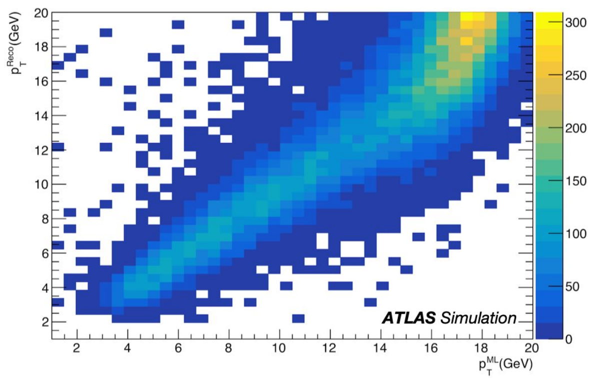
3 × 3 filter

2 × 4 pool.

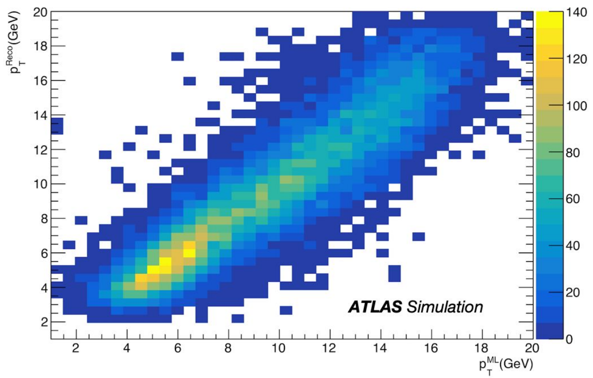
Model:
- (Conv2D + Batch Norm. + Max Pooling) x3
  - Conv. Layers are useful to cope sparsification
- Some dense layers to get the output (5 FP)

Total numb. of parameters: 500k
- Lower than a pure dense NN

FP NN performances - Physical quantities

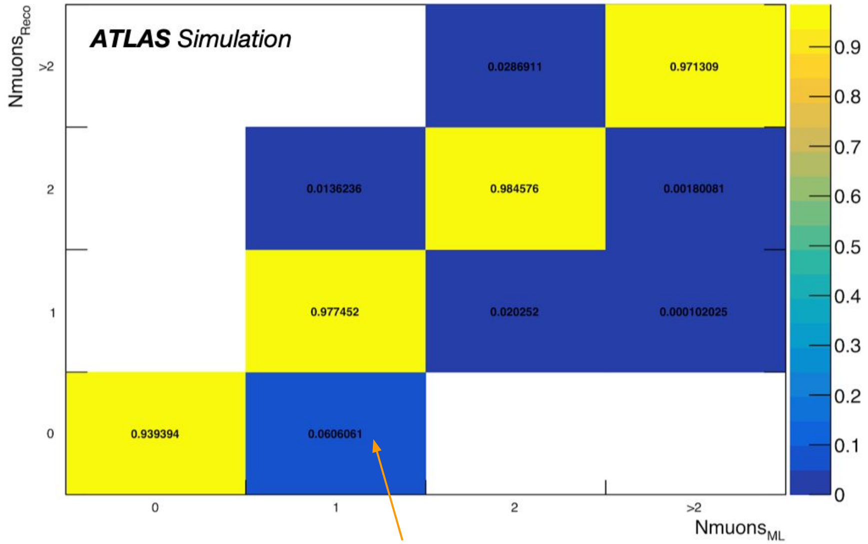


Leading (↑) and subleading (↓) muon



Interesting physical quantities are well fitted

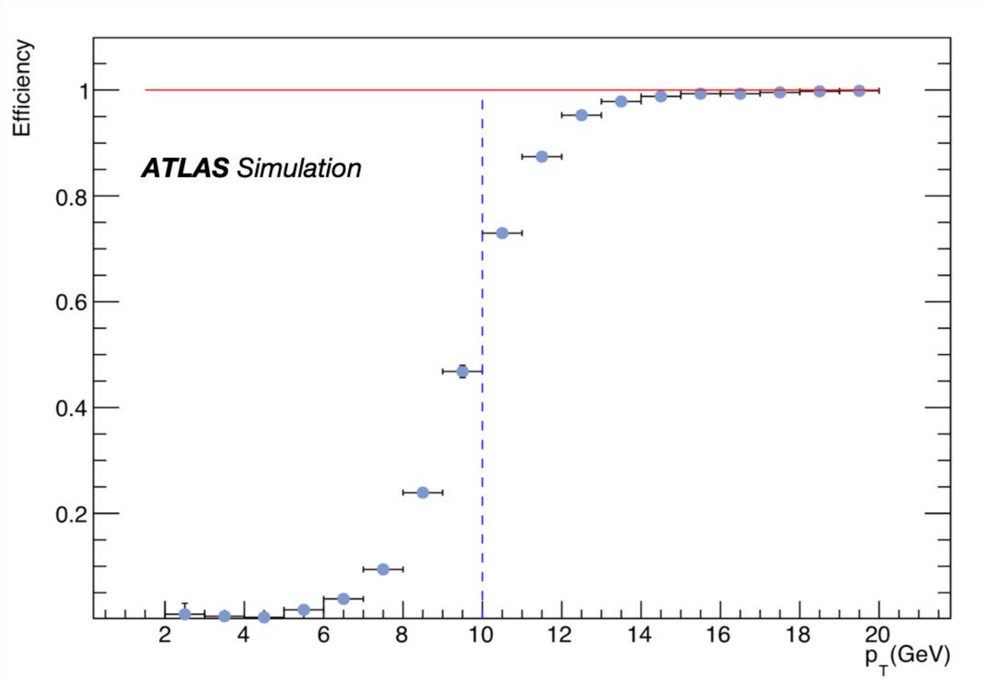
Events are well classified by n^{muons}



Mostly 0-2 GeV “muons” : removed by trigger

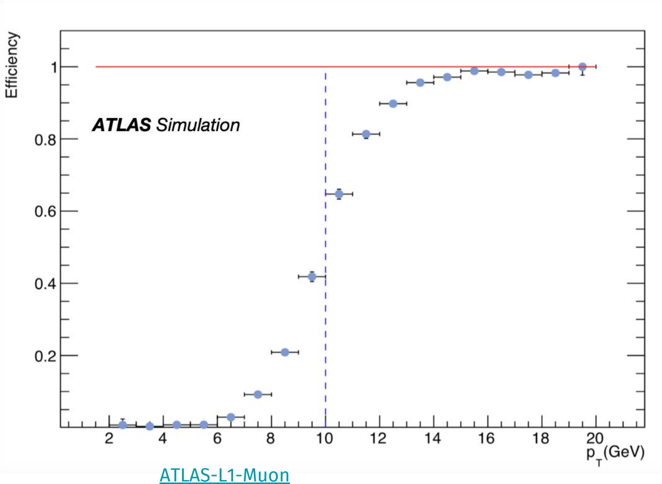
ATLAS-L1-Muon

FP NN performances - Trigger simulation



Leading (↑) and subleading (→) muon

Trigger efficiency curve is well shaped for both leading and subleading muons



ATLAS-L1-Muon

Ternary NN

Level-0 Sector Logic will run on FPGAs

We have to synthesize this NN on an FPGA
- Standard NN have floating point precision weights
- FP weights are not optimal
    - Waste of logical resources

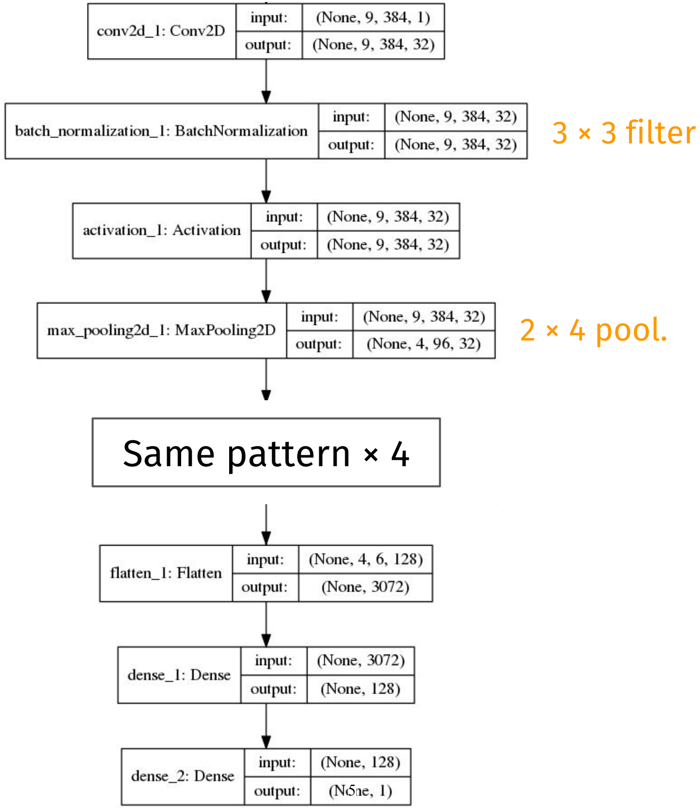
A ternary NN can be set up:
- Weights =-1, 0, +1 (only 2 bits required)

Ternary NN means a **loss in precision**
- Few percent loss with respect to a same-structure NN

But it is **smaller**
- Lower logical resources consumption
- 16 times smaller than a FP32 NN

We can deepen it
- **More layers** to recover precision

Ternary CNN structure



Model:

- (Conv2D + Batch Norm. + Max Pooling) x4
  - Same pattern as FP case
  - Batch Norm is crucial for a ternary NN
- Some dense layers to get the output (5 FP)

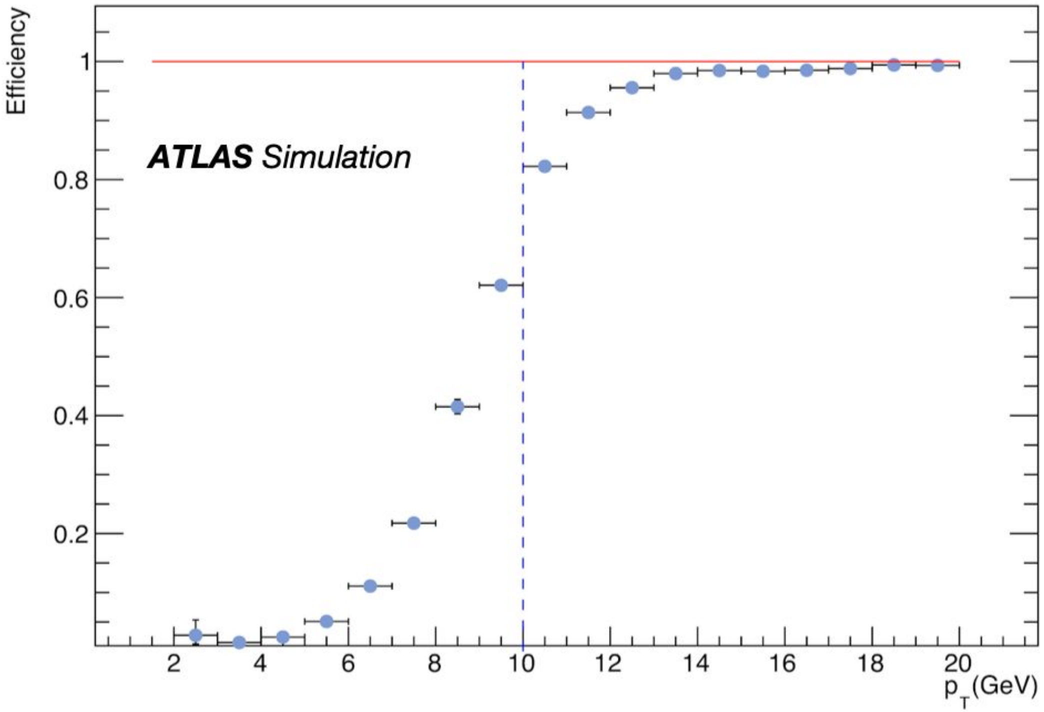
Total numb. of parameters: 1M

- Bigger than FP NN
- Here each weight can be just +1/0/-1
  - Potentially only 2 bits

In this TNN back-propagation is actually the same of the FP NN

- It is necessary only during training
- It can be removed during synthetization

Ternary NN - Performances



As well as trigger turn on curve for the leading muon

ATLAS-L1-Muon

NN on FPGA - How to implement

To implement a NN on an FPGA we need to translate traditional open-source machine learning package models into HLS that can be synthesize:

- We use HLS4ML tp create firmware implementations of ML algorithms

Common python models

↓

→ C/C++ model

C/C++

↓

Vivado HLS

↓

Time consuming programming → VHDL/Verilog

↓

Synthesis and Implementation

Current state of the work

NN can be used to implement the L0 muon trigger algorithm for the ATLAS Phase-II upgrade

- Good resolution and response
- Lead. and sublead. muons p_T and η regression

We developed a Ternary Convolutional NN that far exceeds the performance of the conventional trigger

- can be synthesised within the resource budget of the FPGAs planned to be used in the L0 Trigger algorithm

Simple Dense NN and 2D-Conv. NN have been successfully implemented into an FPGA

- Firmware model created using HLS4ML

Currently working on **firmware of our Ternary CNN**

F32 Dense NN with 3 layer

* Summary:

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6	-
FIFO	-	-	-	-	-
Instance	13	3376	44226	131419	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	36	-
Register	-	-	4450	-	-
Total	13	3376	48676	131461	0
Available	960	1824	433920	216960	64
Utilization (%)	1	185	11	60	0

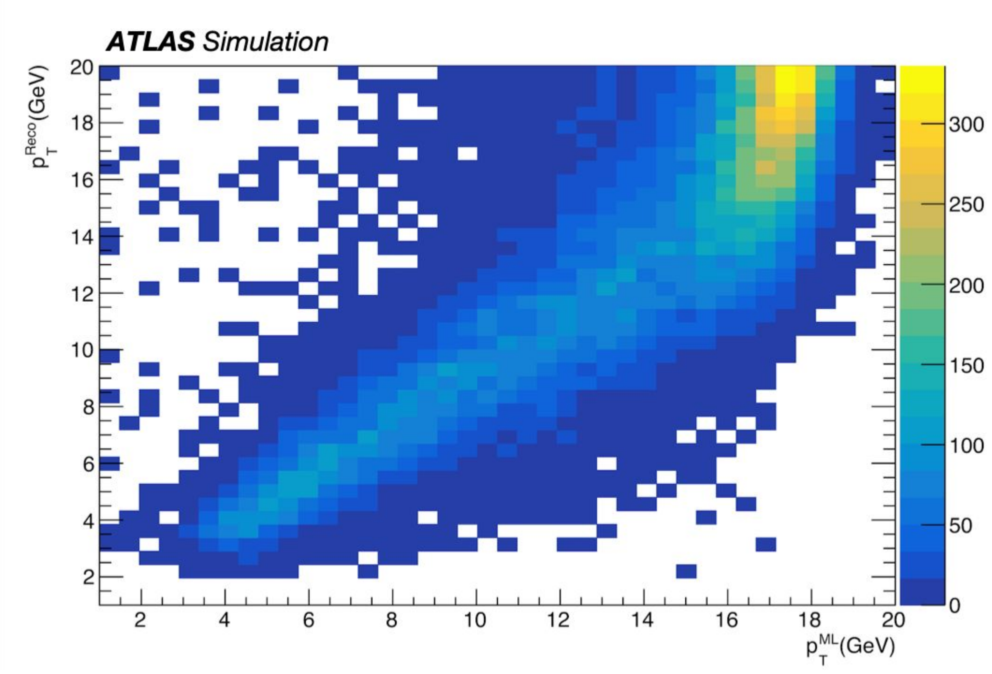
Ternary Dense NN with 3 layer

* Summary:

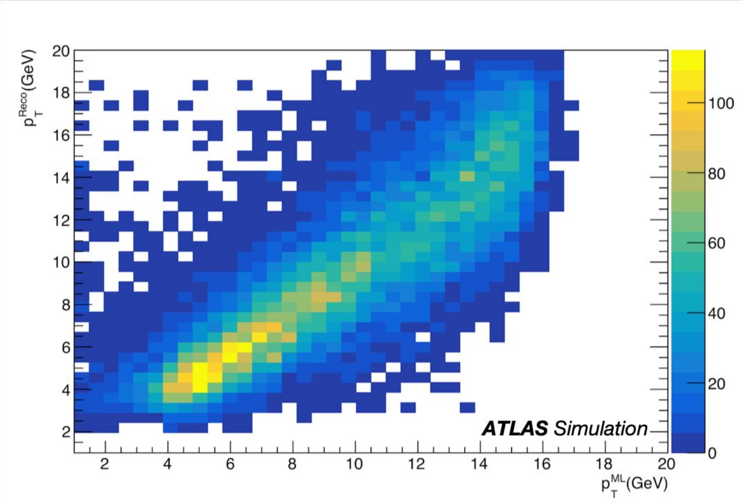
Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6	-
FIFO	-	-	-	-	-
Instance	-	123	9626	59344	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	36	-
Register	-	-	5906	-	-
Total	0	123	15532	59386	0
Available	960	1824	433920	216960	64
Utilization (%)	0	6	3	27	0

Thank you for your attention

Ternary NN - Performances

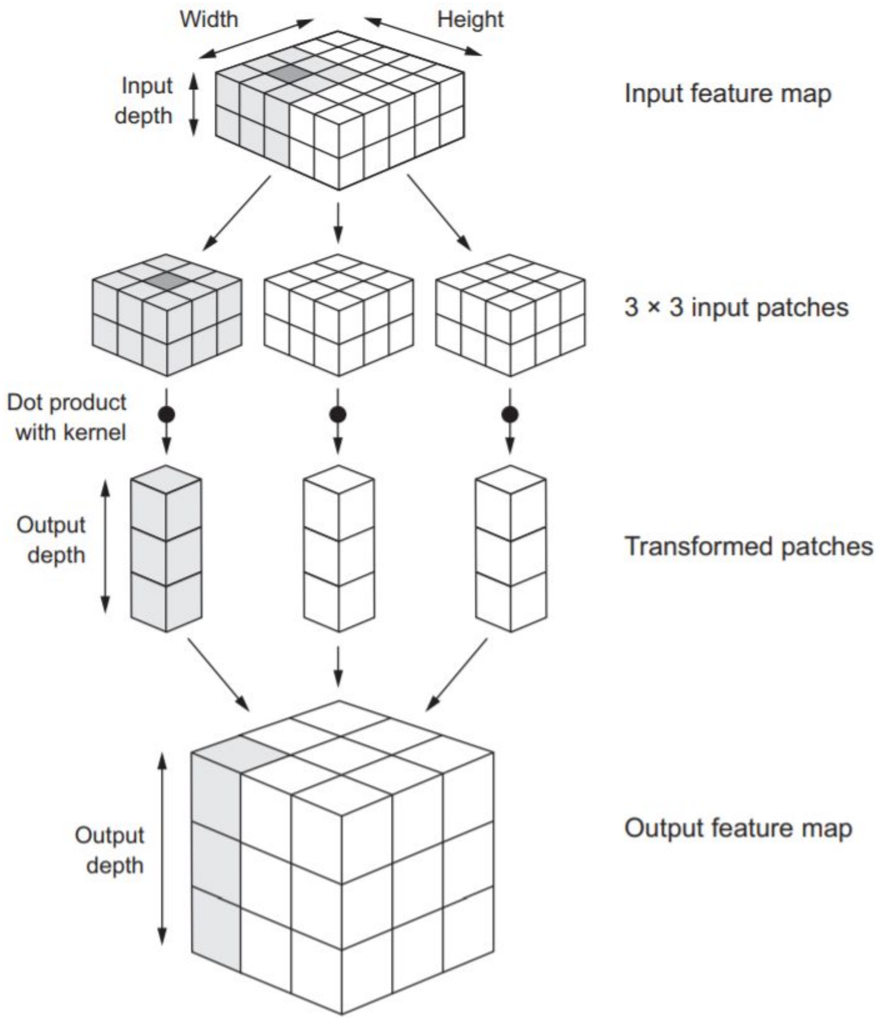


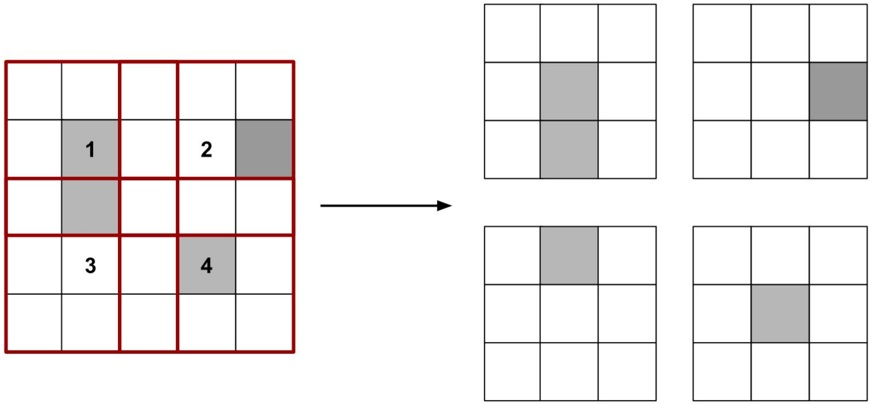
Physical quantities are well reproduced



Leading (↑) and subleading (→) muon

ATLAS-L1-Muon


Width
Height
Input depth
Input feature map
3 × 3 input patches
Dot product with kernel
Output depth
Transformed patches
Output depth
Output feature map


1
2
3
4