2025 Activities

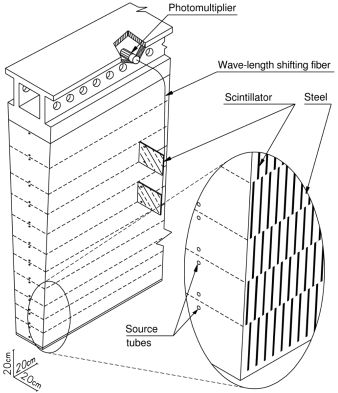
Wang Rui

Dec 16th, 2025

HORIZON-MSCA-COFUND-2022-1 GA 101126626

Introduction

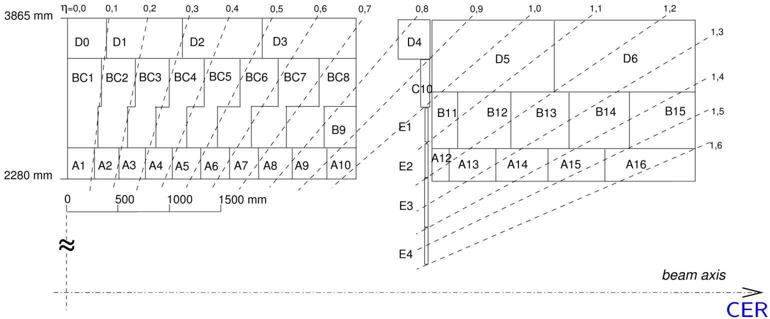
- In Tile Calorimeter (TileCal) at the ATLAS experiment, signals are processed online using the Optimal Filtering method.
- At the HL-LHC, signals will be processed per Bunch Crossing (BC) before passed to the first level of trigger
- Higher pileup and data rates, together with the need for real-time reconstruction, call for state-of-art algorithms
- **Develop ML algorithms to separate signal from noise before pulses are reconstructed**
- **Aim to reduce the time of energy reconstruction and improve energy resolution**
- The algorithms will be ultimately deployed on FPGAs



CERN-LHCC-2017-019

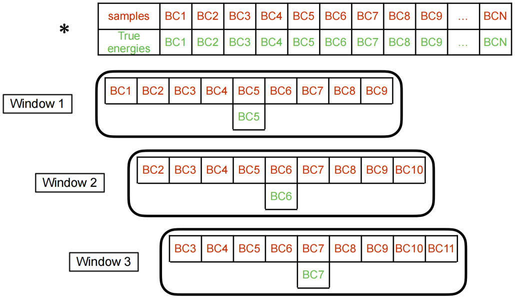
Data

- Simulated datasets are generated by the Pulse Simulator
- Samples used are ~1 million bunch crossings (BC) with minimum bias $<\mu>=200$ pile up only
- Only look at samples in the High Gain (HG)



CERN-LHCC-2017-019

Preprocessing



* credit to Francesco Curcio

- The ADC readings range between 0 and 4095
- Sliding window with a size 9
- Samples are simulated readout energy from the electronics, the inputs to the models
- Targets hold the information of true energy
- If the central BC has
  - $E_{true} < 10$ ADCs, the window is classified as noise
  - $E_{true} > 10$ ADCs, the window is classified as signal
- The window is dropped if any BC in the samples has an ADC count of 0 or 4095
- Train:validation:test split 6:2:2

Signal-noise separation: Data

- There is little gain to train a pre-filter for the signal-dominant cells
- The ratio of noise over signal for the preprocessed samples is checked for individual cells
- We are using data in noise-dominant cells in all four partitions (LBA, LBC, EBA and EBC)
- Samples from cells with a ratio ≥ 1.5 are used for training
- The ratio of all cells combined is approximately 3.92
- This imbalance is accounted for in the training

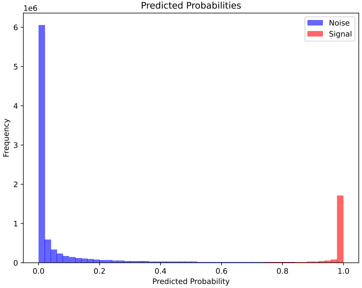
Cell	Ratio
A1	1.97
A2	1.86
A3	1.70
A4	1.68
A5	1.49
A6	1.40
A7	1.26
A8	1.04
A9	0.83
A10	0.84
BC1	5.51
BC2	5.33

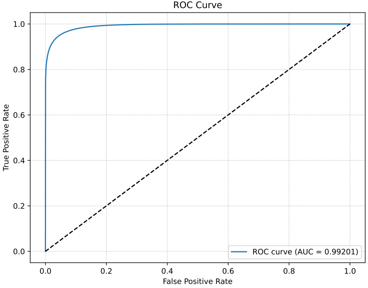
Cell	Ratio
BC3	5.22
BC4	4.84
BC5	4.58
BC6	4.06
BC7	3.59
BC8	2.11
B9	2.68
D0	53.70
D1	53.88
D2	50.78
D3	10.52
A12	0.42
A13	0.14

Cell	Ratio
A14	0.34
A15	1.34
A16	5.24
B11	0.94
B12	1.34
B13	2.37
B14	6.04
B15	18.52
C10	0.65
D4	3.73
D5	2.52
D6	24.13

Classification with large models

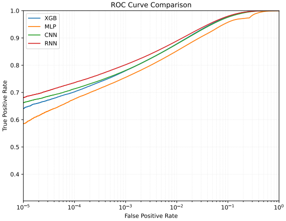
- Efforts started with models with very large number of model parameters
- Good classification was achieved
- Example: classification with XGBDT

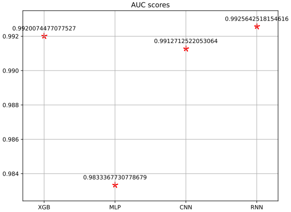


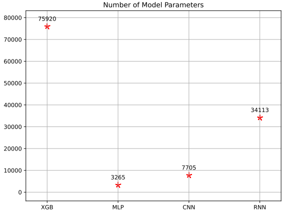


Classification with large models

- Comparison of different models



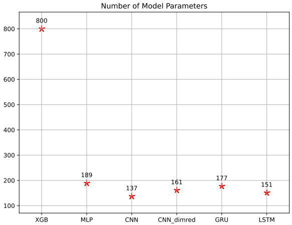


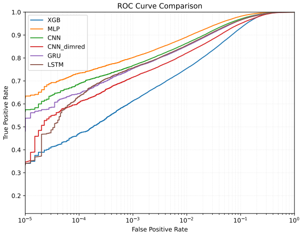


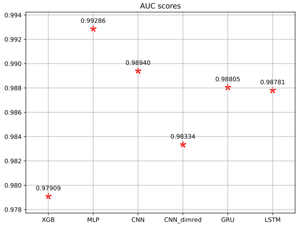
- Model sizes have to be reduced due to limitation of resrouces on FPGAs

Classification with small models

- A varity of model structures are studied, including BDT, MLP, CNN and RNN
- Number of parameters are constrained within 200
- Drop BDT and RNN due to worse performances and/or complexity of FPGA implementation
- Focus on MLP and CNN

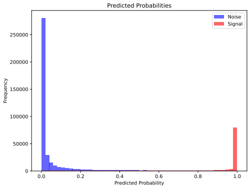


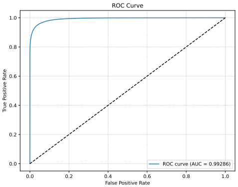




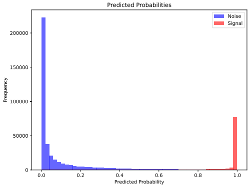
Classification with small models

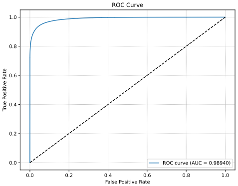
- Classification with MLP (optimised)





- Classification with CNN (unoptimised)

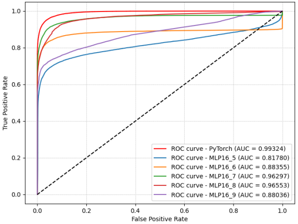


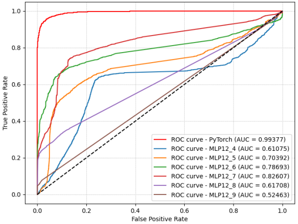


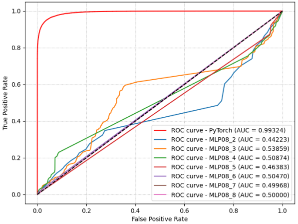
- Further optimisation of hyperparamters is ongoing

FPGA synthesis

- Two ways of synthesizing a neural network model for FPGA are studied:
  - hls4ml: python package, straightforward to use
  - FINN: has to be used within a docker container
- Three ways of quantising the trained model:
  - Post-training quantization (PTQ): train a model, then deploy it with quantisation
  - Quantization-aware fine-tuning (QFT): quantise the trained model and fine-tune it
  - Quantization-aware training (QAT): train a quantised model from scratch
- All three ways above are being investigated







Thanks to everyone!