AI-assisted muon triggers for LLP with FPGA based inference

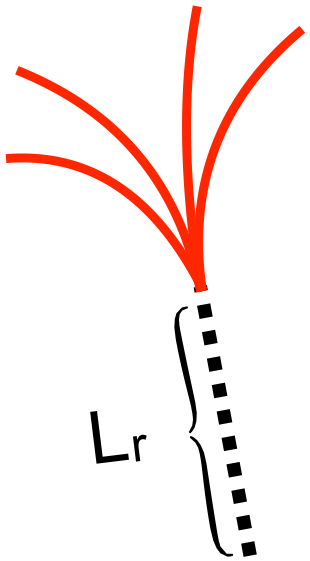
AI-assisted muon triggers for LLP

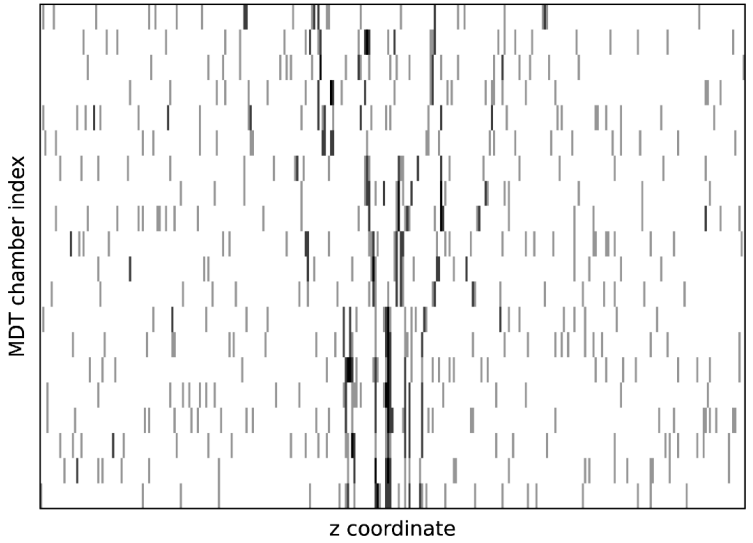
Case of study: “simplified” High Level Muon Trigger

DNN model trained to identify long-life particles (LLP) from hit patterns released in the MDT chambers

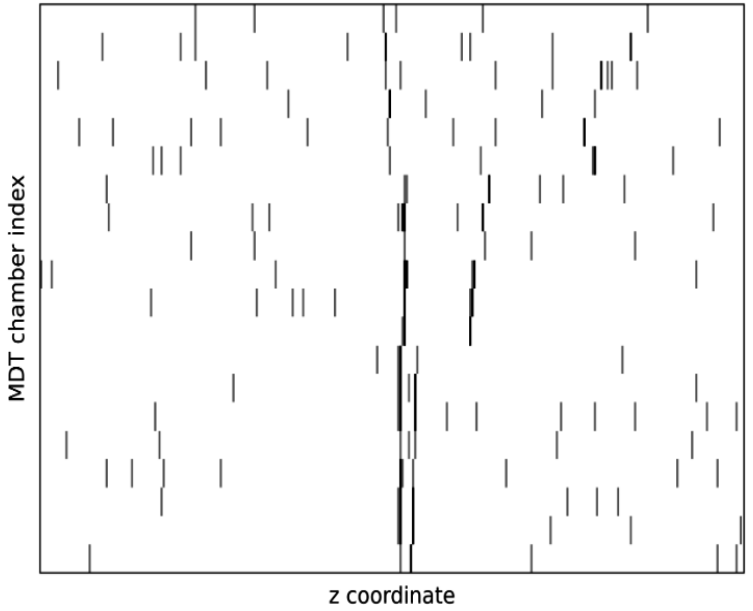
- neutral particles decaying into multi-muons at different distances from the primary vertex of interaction, hits patterns in the spectrometer are represented as binary images

- CNN trained to predict the radial decay length (Lr) of the LLP, on events generated with a **simplified** simulation of the geometry and resolution of the ATLAS MDT detector





$\pi_V \rightarrow b\bar{b} \rightarrow \mathrm{N} \in [2,10]$



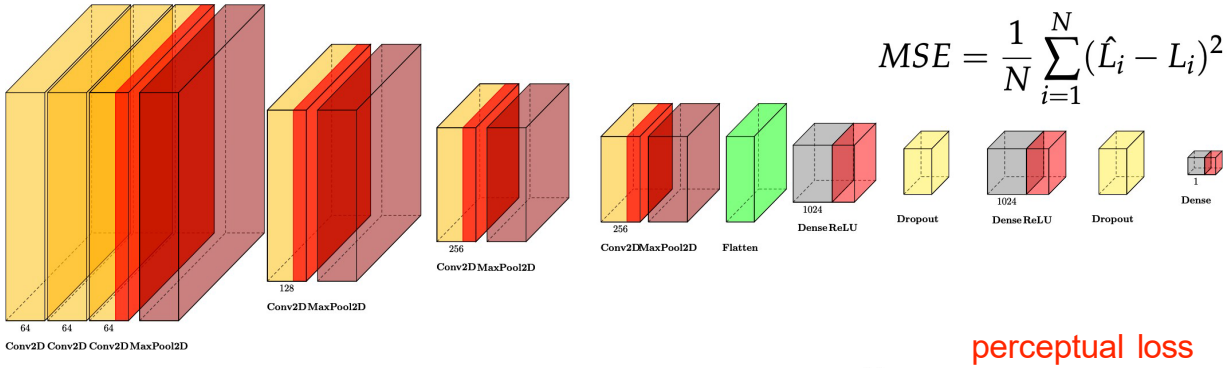
$\gamma_d \rightarrow \mu^+\mu^-$

AI-assisted muon triggers for LLP

- Different types of architecture considered:

- convolutional networks (CNNs) trained in a supervised manner to predict the decay vertex of the LLP from hit patterns in the spectrometer

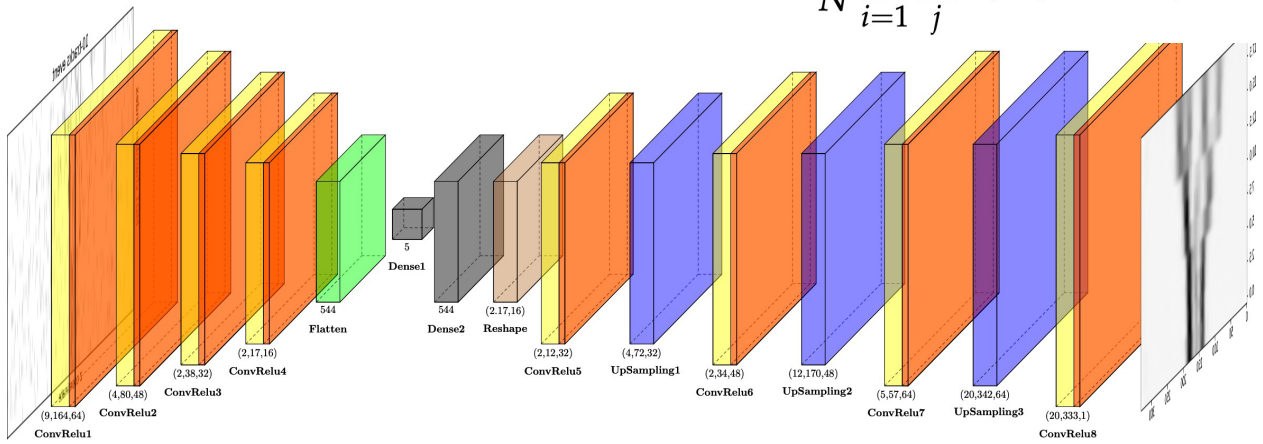
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{L}_i - L_i)^2$$



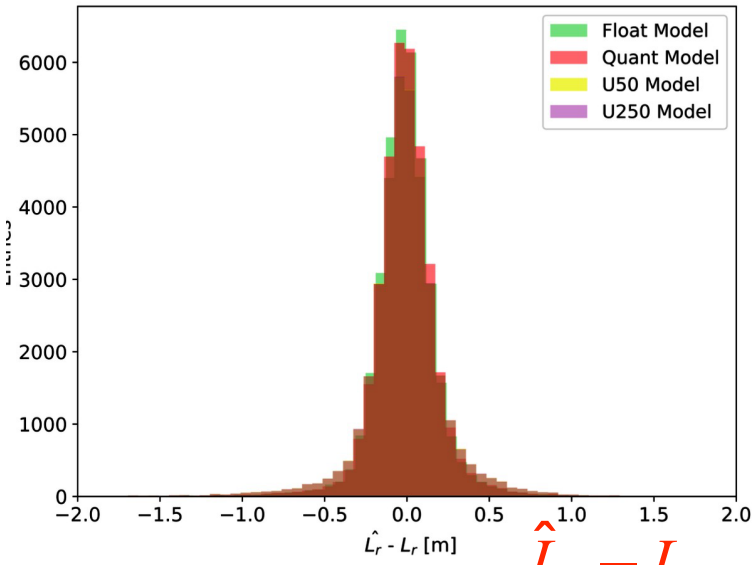
perceptual loss

$$L_{\mathrm{AE}} = \mathrm{BCE}(y,\hat{y}) + \frac{1}{N}\sum_{i=1}^{N}\sum_{j} w_j |\vec{\phi}_j(y) - \vec{\phi}_j(\hat{y})|$$

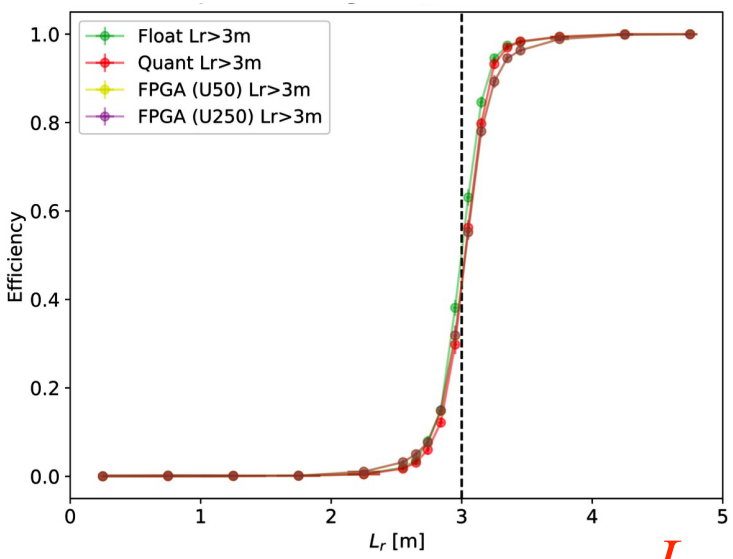
- convolutional auto-encoders (AE-NN) trained for anomaly detection in a partially supervised way (only "normal" events consisting of SM processes): provide a measure of how far an input event deviates from the learned representation for normal events



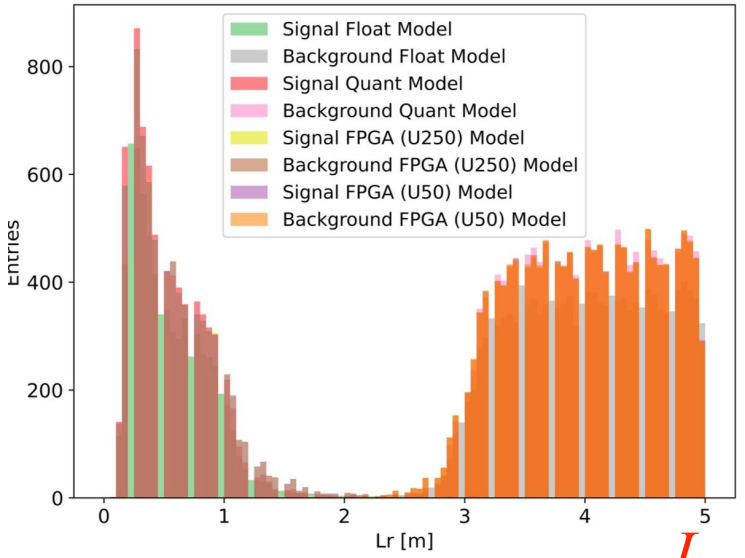
AI-assisted muon triggers on AMD FPGA preliminary results



$\hat{L}_r - L_r$



L_r



L_r

	CPU	GPU	U50	U250
Throughput [fps]	269.2 ± 0.4	1401.7 ± 22.6	950.3 ± 5.0	548.1 ± 4.1
Inference time [ms]	13.8 ± 3.2	100.8 ± 2.7	3.7 ± 0.1	12.8 ± 0.3

Board Setup:
ALVEO U50 con **Vitis-AI 1.4.1** (+ CPU **Intel Xeon E5-2698 2.2GHz**)
ALVEO U250 con **Vitis-AI 2.5** (+ 2 CPU **Intel Xeon Bronze 3204 1.9G**)
GPU **Nvidia Tesla V100** .

On going performance studies

- performance study and scaling on single FPGA accelerator
  - Latency, memory/resource usage, physical performance triggers, ...
  - Neural architecture dependency: DNN vs RNN vs CNN vs GNN vs Transformers vs hybrid
  - scaling architectures with model size with threads number, parallelization layer, ...
  - memory occupation/transfer optimization: compression (pruning, weight clustering, ...)
  - quantization aware vs tuned quantization
  - dependence on different firmware/DPUs available on FPGAs (latency vs throughput optimized DPUs)
- study with FPGAs of different technologies (both from the point of view of hw and software tools):
  - AMD/Xilinx vs Intel/Altera Accelerators, VitisAI vs OpenAPI Development Environments/Libraries, vs hls4ml+HLS, vs Mipsology ZebraAI, vs ...
- study of optimal strategies for different menu triggers (physics signals):
  - DNN single task/goal vs multi-task/goal models,

Future steps

- Performance study on production hardware :
  - Test the performance (and performance/power) of the DNN inference on production DAQ CPU cluster.

- Realistic training of the DNNs
  - Conduct the training with full realistic simulation of MDT chambers events Instead of a simplified simulation.
  - Test the inference performance against the real data.

- Performance study and scaling on multi-accelerators:1 node with 2 FPGAs, 2 nodes with 1 FPGA, 2 nodes with 2 FPGAs
  - **Optimization/tuning to maximize performance on a FPGA cluster**
  - detailed bottleneck study in data flow vs processing, balance between CPU / FPGA load.