Cluster di FPGA: use-case ATLAS

Persone coinvolte:
- S. Giagu (Sapienza), N. Stocchetti (Sapienza, lau.), G. Russo (Sapienza, phd), E. Martino (Sapienza, lau.)
- A. Coccaro (Genova), F. di Bello (Genova), L. Rambelli (Genova, phd)
- B. Spisso (Napoli)

CN1 Spoke2 WP2/WP4 - 17.3.2023

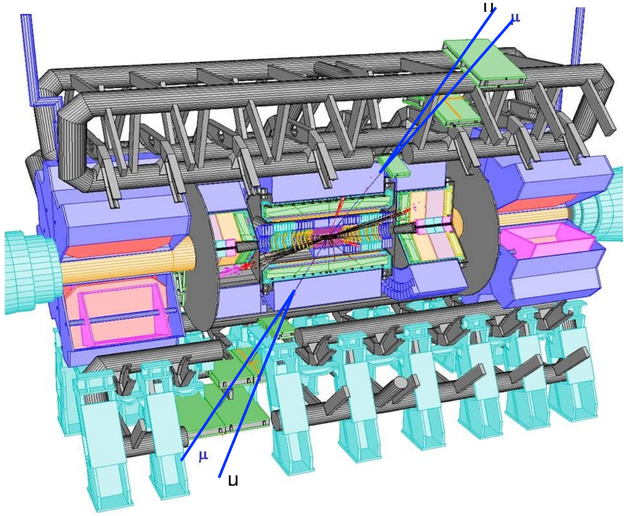
ATLAS EXPERIMENT

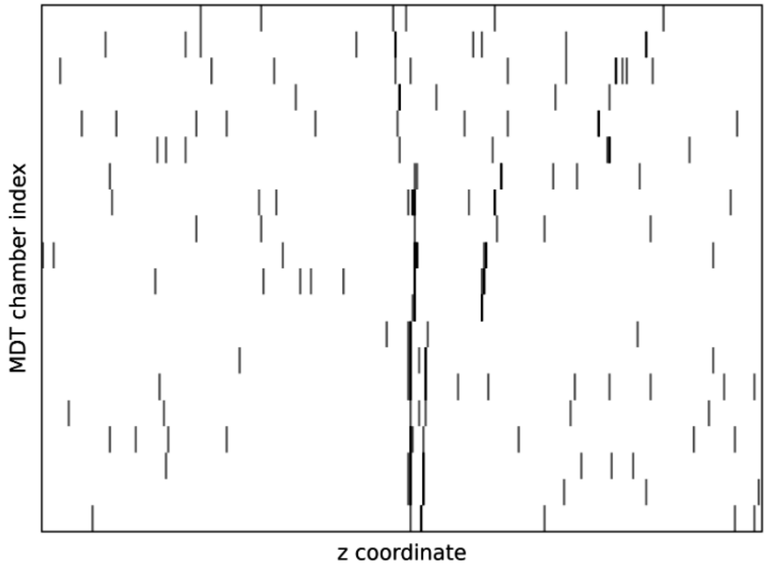
INFN Istituto Nazionale di Fisica Nucleare

SAPIENZA UNIVERSITÀ DI ROMA

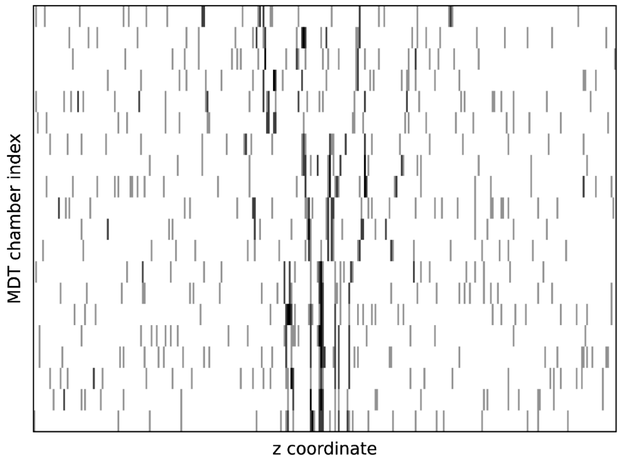
ACTIVITY 1: IDENTIFICAZIONE LLP NELLO SPETTROMETRO A MUONI

- sviluppo di algoritmi EF-Muon basati su DNN per l'identificazione nello spettrometro muonico di decadimenti di nuove particelle esotiche a lunga vita media
  - benchmark-NP: settori dark e hidden valley (v-pions)
  - segnature: LLP neutre con lunghezze di decadimento comparabili con le dimensioni del rivelatore ATLAS che decadono in "jet" collimati di leptoni o fermioni





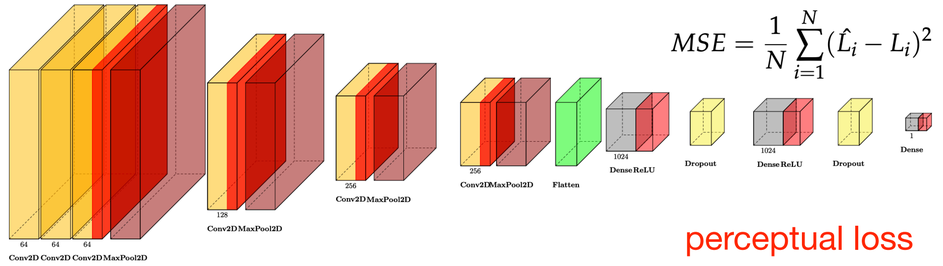
$\gamma_d \to \mu^+\mu^-$



$\pi_V \to b\bar{b} \to$ N $\in [2,10]$ tracce cariche nello spettrometro

si prestano bene ad essere analizzate con DNN di tipo convoluzionale o tramite GNN/Transformers …

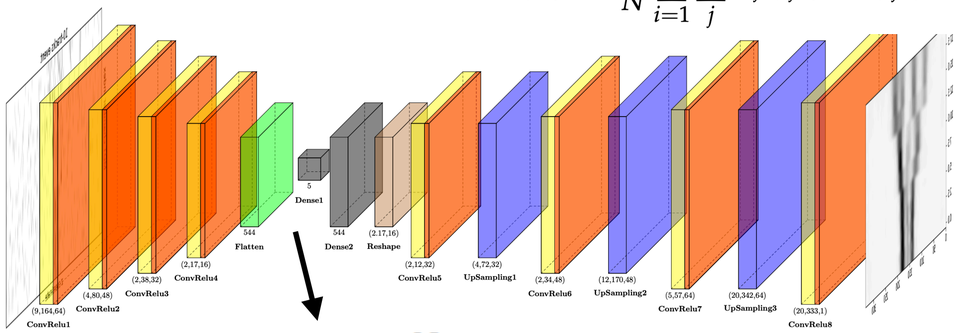
- tre tipologie di architetture “benchmark” in studio:
  - reti convoluzionali (CNN) addestrate in modo supervisionato a predire il vertice di decadimento della LLP a partire dai pattern di hit nello spettrometro



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

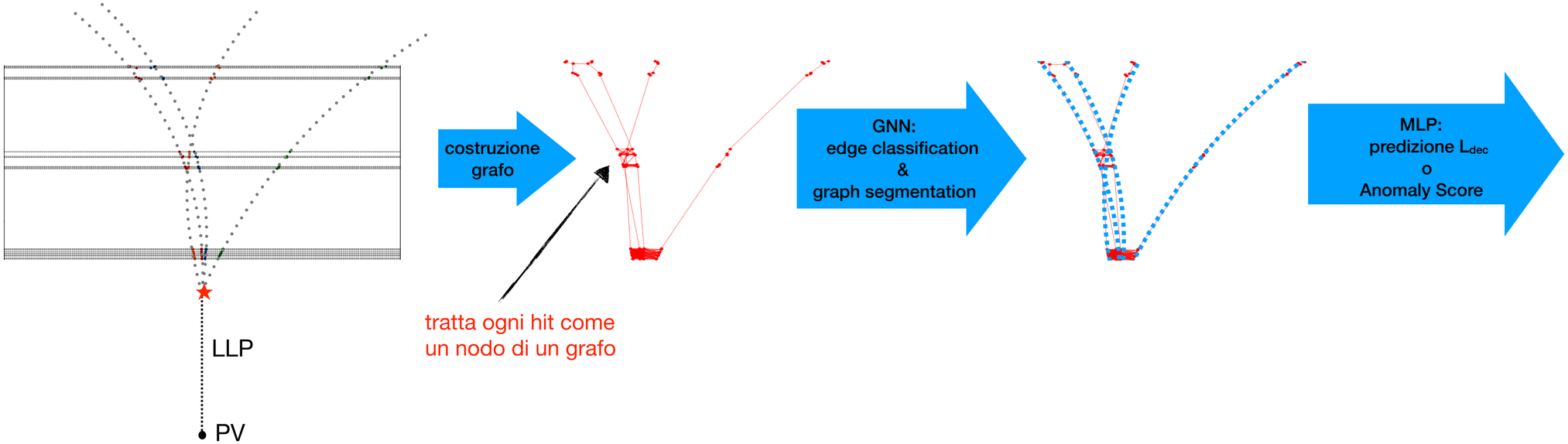
perceptual loss

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



$$M = \sum_i^N z_i$$ anomaly score

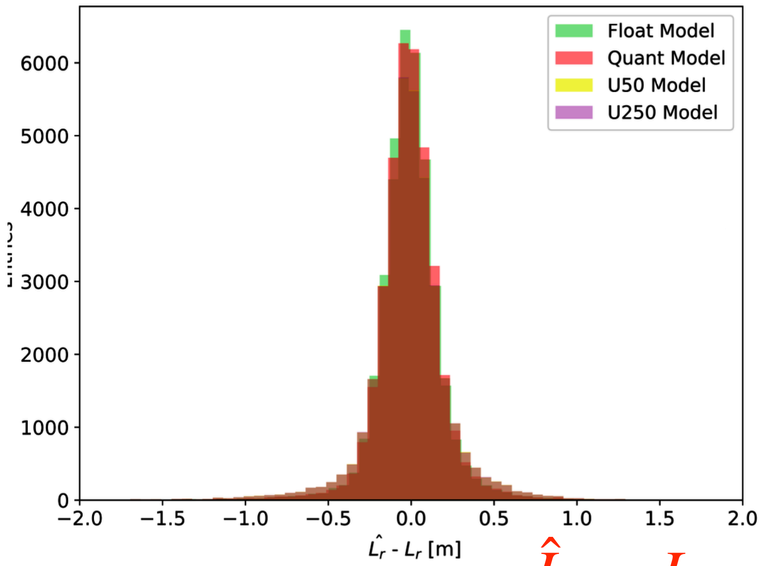
- architetture GNN / GNN-Transfomers / Transformers sia supervisionate che come anomaly detector



ESEMPIO PRESTAZIONI: SUPERVISED REGRESSION CNN

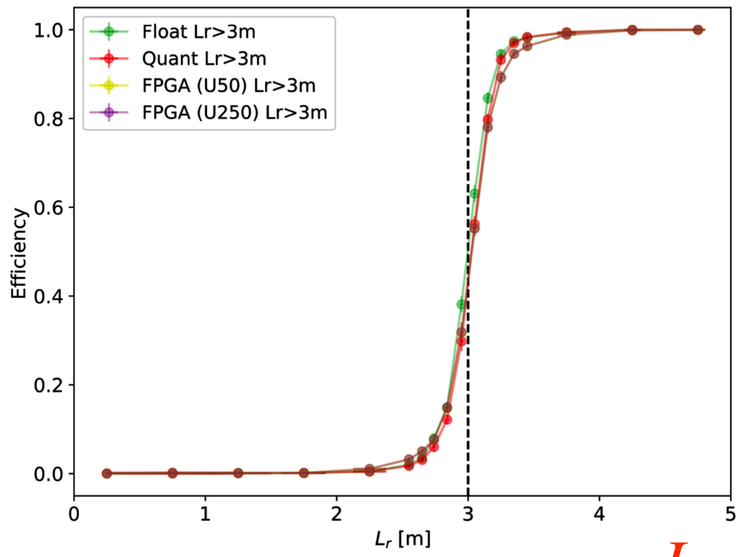
- CNN addestrata a predire la lunghezza di decadimento radiale (L_r) della LLP, su eventi generati con una simulazione semplificata della geometria e della risoluzione del rivelatore MDT di ATLAS

Distribuzione dei residui



$\hat{L}_r - L_r$

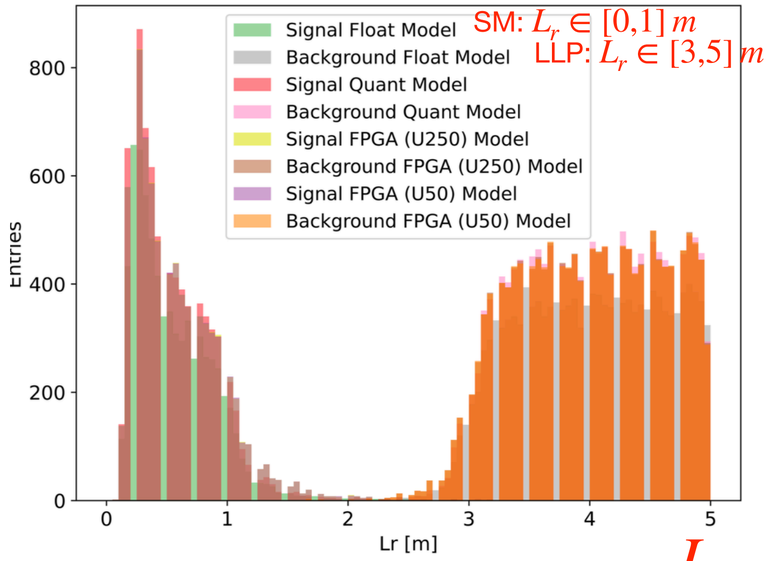
Efficienza di selezione del segnale per un taglio @ Lr>3m



L_r

Distribuzione eventi SM e LLP predetti dai diversi modelli

SM: $L_r \in [0,1]\,m$
LLP: $L_r \in [3,5]\,m$



L_r

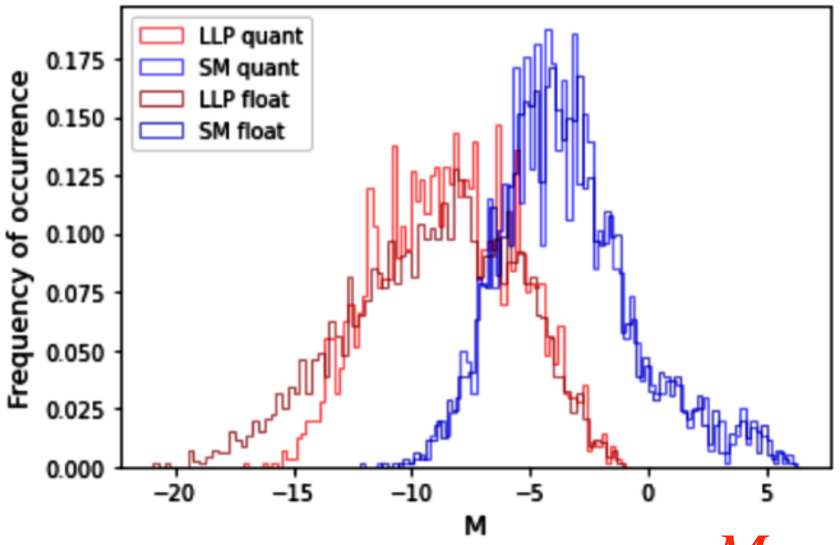
	CPU	GPU	U50	U250
Throughput [fps]	269.2 ± 0.4	1401.7 ± 22.6	950.3 ± 5.0	548.1 ± 4.1
Inference time [ms]	13.8 ± 3.2	100.8 ± 2.7	3.7 ± 0.1	12.8 ± 0.3

Throughput medio calcolato su 2000 frames con batch size 4 (CPU/GPU/U250) o 3 (U50)
Inference time per 1 batch di 4 frame (CPU/GPU/U250) o 3 frame (U50)
CPU FP32 all available cores w/ ONNX runtime engine
GPU FP32 tensorflow engine (no TensorRT)
FPGA INT8

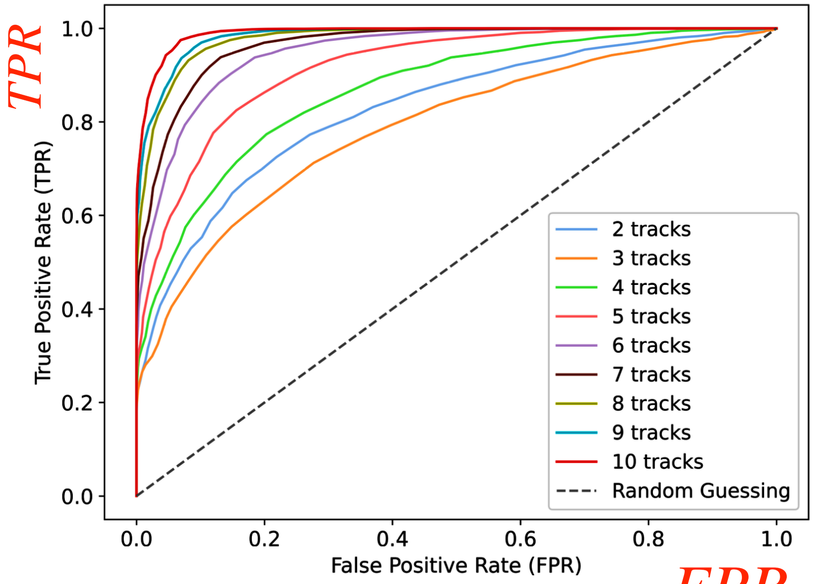
Board Setup:
ALVEO U50 con **Vitis-AI 1.4.1** (+ CPU **Intel Xeon E5-2698 2.2GHz**)
ALVEO U250 con **Vitis-AI 2.5** (+ 2 CPU **Intel Xeon Bronze 3204 1.9G**)
GPU **Nvidia Tesla V100** .

ESEMPIO PRESTAZIONI: ANOMALY DETECTION CAE

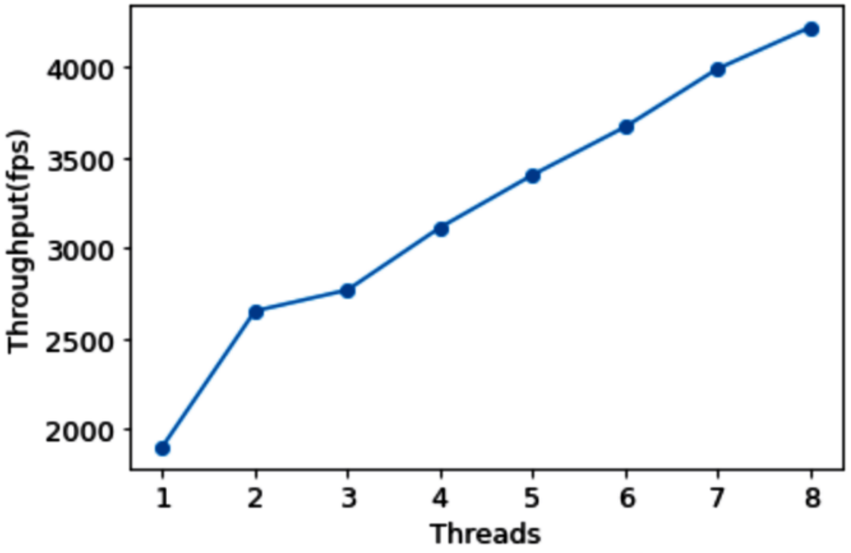
- CAE addestrata con eventi “prompt” e testata con eventi “prompt” e “displaced”
- anomaly score: aggregazione (sum) dello spazio latente appreso dall’auto-encoder



M



FPR



#threads

	CPU	GPU	FPGA
Throughput(fps)	1329.7 ± 21.0	1731.1 ± 28.5	1919.8 ± 3.8

Board Setup:
ALVEO U50 con **Vitis-AI 1.4.1** (+ CPU **Intel Xeon E5-2698 2.2GHz**)
GPU **Nvidia Tesla V100** .

BENCHMARK E TEST PIANIFICATI

- **studio prestazioni e scaling su singolo acceleratore FPGA:**
  - latenza, utilizzo memoria/risorse, prestazioni fisiche trigger, …
  - dipendenza dall'architettura neurale: DNN vs RNN vs CNN vs GNN vs Transformers vs architetture ibride
  - scaling con la taglia del modello
  - scaling con numero threads, livello parallelizzazione, …
  - ottimizzazione dell'occupazione / trasferimento memoria: compressione (pruning, weight clustering, …)
  - quantizzazione (quantization aware vs tuned quantization)
  - dipendenza da diversi firmware/DPU disponibili sulle FPGA (latency vs throughupt optimized DPUs)

- **studio prestazioni e scaling su multi-acceleratore FPGA:**
  - 1 nodo con 2 FPGA, 2 nodi con 1 FPGA, 2 nodi con 2 FPGA
  - ottimizzazione/tuning per massimizzare le prestazioni
  - studio dettagliato bottleneck nel data flow vs processamento, bilanciamento tra carico CPU / FPGA

- **studio con FPGA di tecnologie diverse:** (sia dal punto di vista dell'hw che degli strumenti software): Acceleratori AMD/Xilinx vs Intel/Altera, ambienti di sviluppo/librerie VitisAI vs OpenAPI, vs hls4ml+HLS, vs Mipsology ZebraAI, vs …

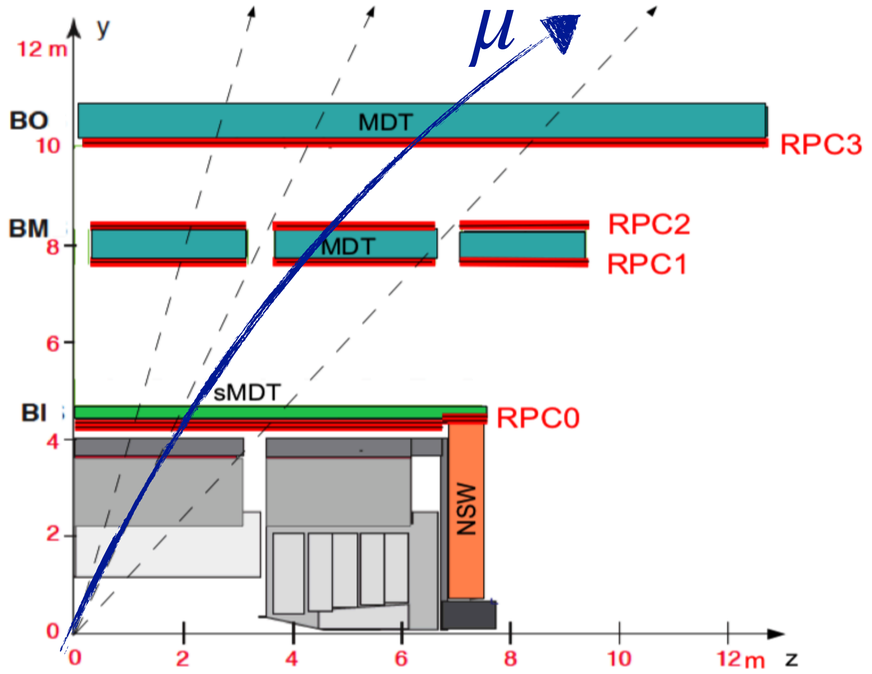
- **studio strategie ottimali per diversi trigger menu (segnali di fisica):** modelli ANN single task/obiettivo vs multi-task/obiettivo, precision physics vs NP, …

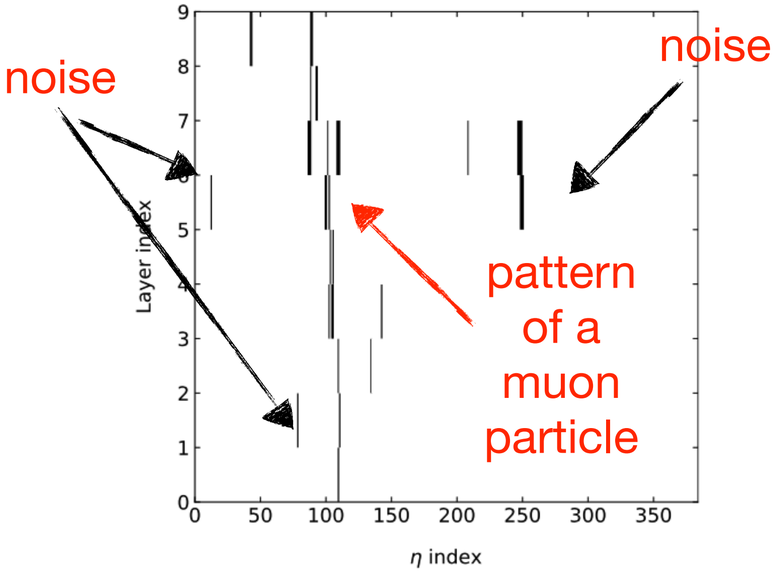
ACTIVITY 2: ULTRAFAST CNN ON FPGAS FOR THE LEVEL-0 MUON TRIGGER @HL-LHC

Goal: accurately reconstruct the momentum and angle of the muon track from the RPC detector hit information **in less than 400ns** (3 orders of magnitude faster than fastest AI models on CPUs and GPUs)

Latency and FPGA resource occupancy are in a trade-off relationship, while AI model performance strongly depends on the neural network scale

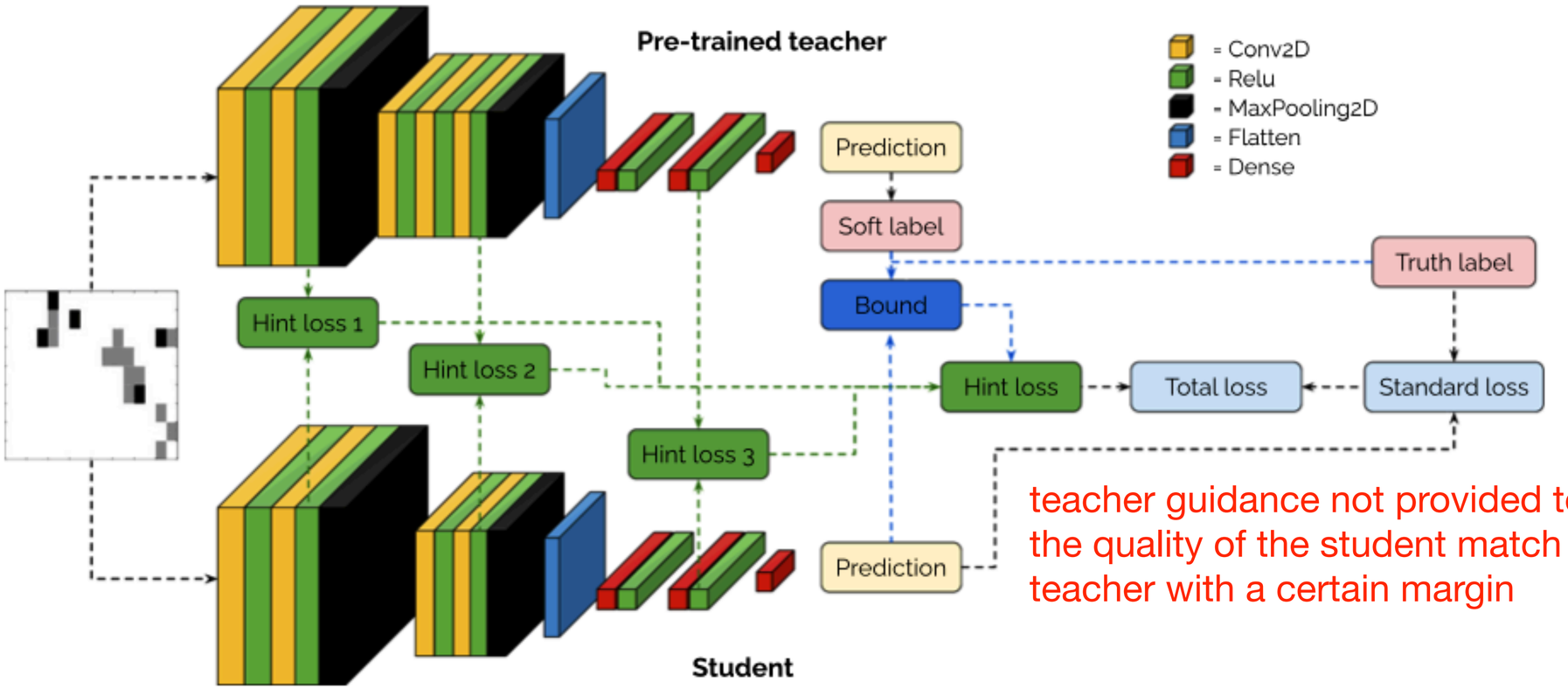
Strategy: multi-stage **AI model compression** and simplification based on **aggressive quantisation** and **knowledge transfer techniques** to avoid degradation of physics performances





KNOWLEDGE TRANSFER FOR CNN MODEL COMPRESSION

transfer knowledge learned by a larger neural network pre-trained for the same task to a smaller and quantised (4-bits per activations and weights) model



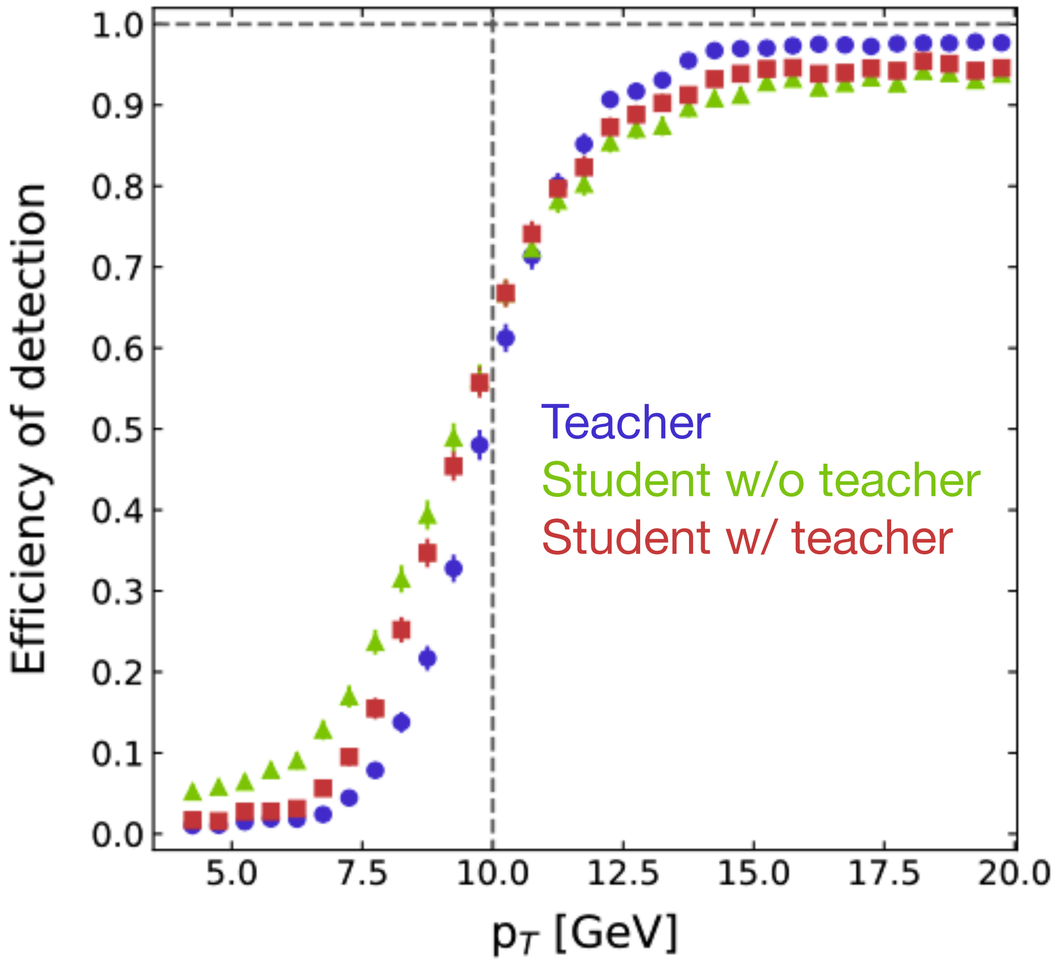
teacher guidance not provided to the student once the quality of the student match or surpass that of the teacher with a certain margin

obtained a reduction on size of the model of a factor 100 with only a limited reduction in performance

S. Francescato, S.Giagu, F. Riti, G.Russo, L.Sabetta, F.Tortonesi, Eur. Phys. J. C (2021) 81:969

PRELIMINARY PERFORMANCES

Single muon trigger efficiency curve
for a nominal threshold of 10 GeV



FPGA resource occupation

Table 3 Percentage occupancy relative to the total FPGA available resources (model xcvu13p-fhga2104-2L-e [14])

Model (9 × 16)	BRAM	DSPs	FF	LUT
Teacher (%)	20.9	258.0	69.4	15.3
Student 32 bit (%)	3.2	31.0	8.4	2.7
QStudent 4 bit (%)	0.2	0.05	0.4	1.7

Inference time per event on FPGA
Xilinx Ultrascale+ XCV13P

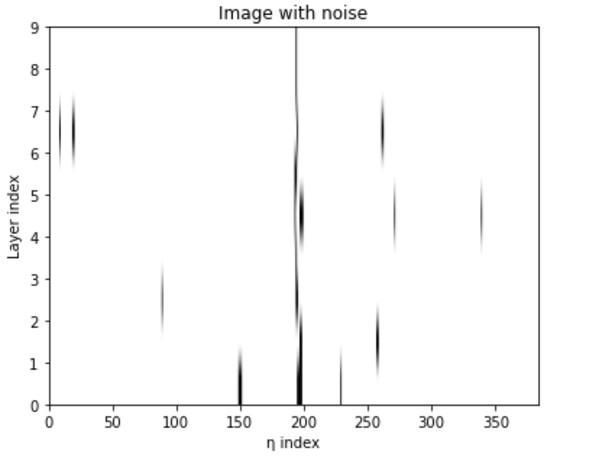
- Teacher fp32: 5 ms (Tesla V100 GPU)
- Student 4 bit: 438 ns (hls4ml)
- Student 4 bit: 84 ns (our VHDL implementation)

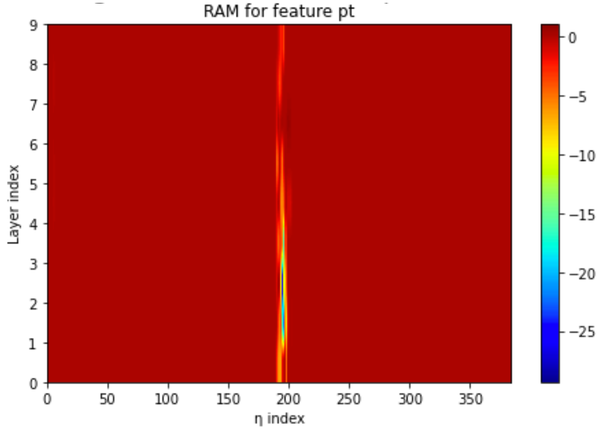
S. Francescato, S.Giagu, F. Riti, G.Russo, L.Sabetta, F.Tortonesi, Eur. Phys. J. C (2021) 81:969

xAI VIA ATTRIBUTION ALGORITHMS

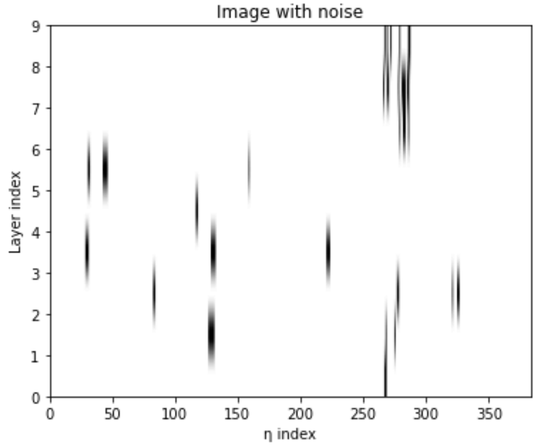
- provide explanations that are based on saliency maps (visualize pixels that have contributed the most to the track reconstruction)
- heat maps obtained with the RAM technique (regression activation maps (generalise grad-CAM for regression tasks))

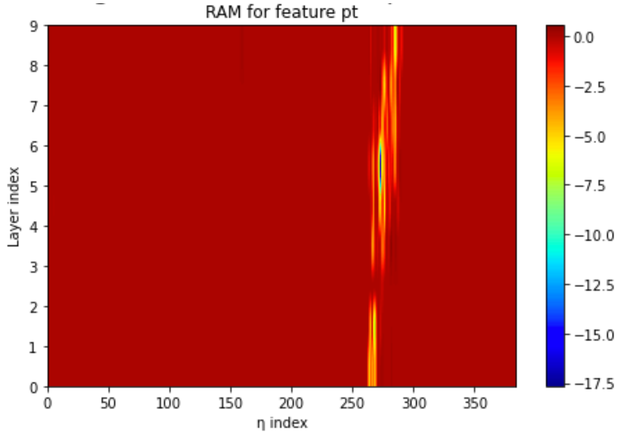
true positive case



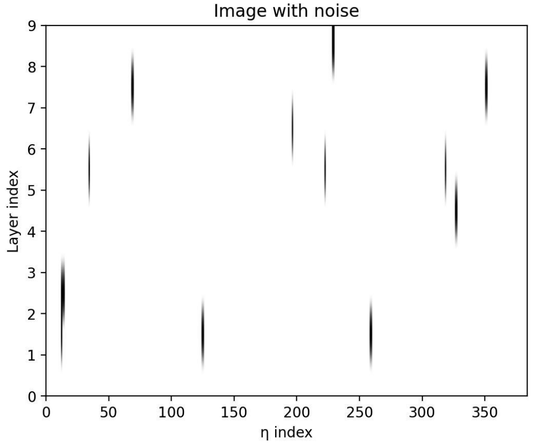


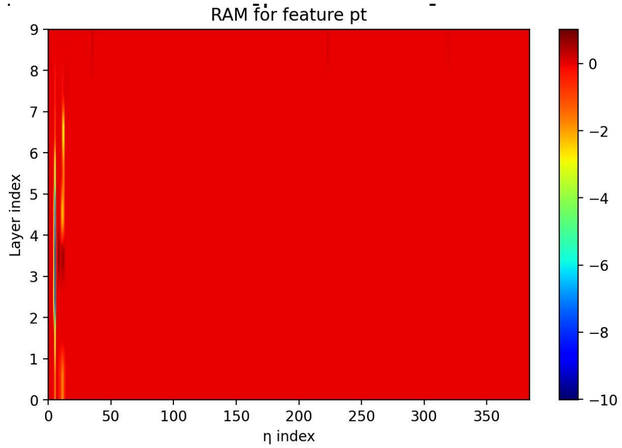
false positive case





noise-only FP case

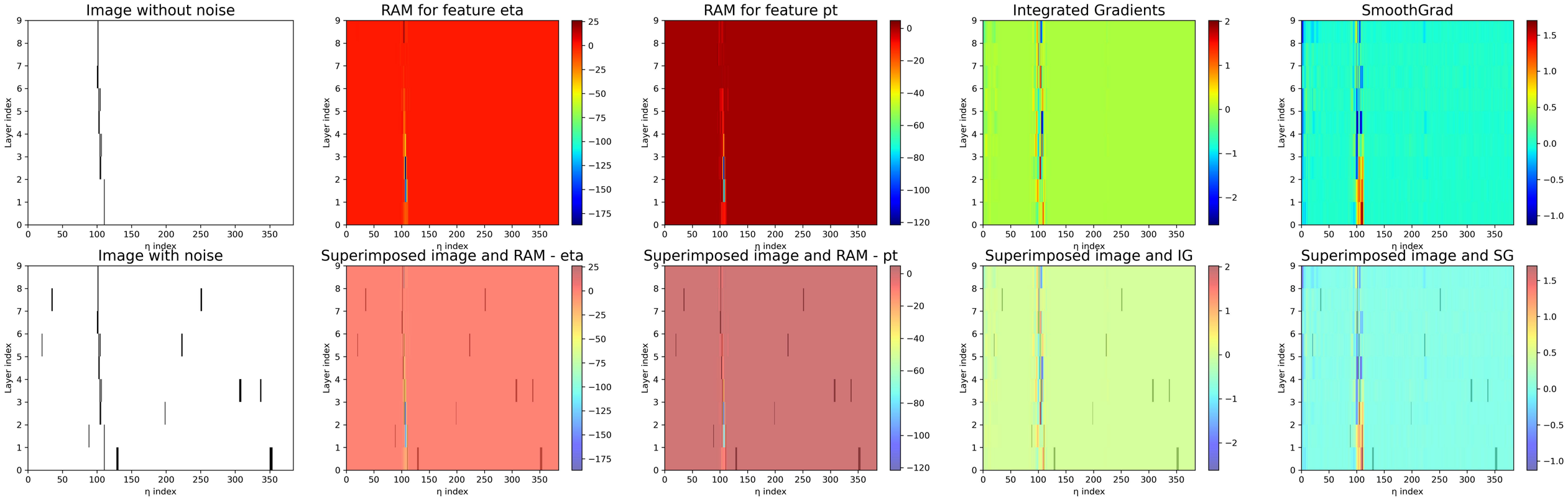




HEAT MAPS VARIATIONS

Regression Activation Maps VS Integrated Gradients VS Smooth Gradients

Real: [pt=14.108, eta=0.336] Predicted: [pt=13.172, eta=0.229]



consistent results among the methods

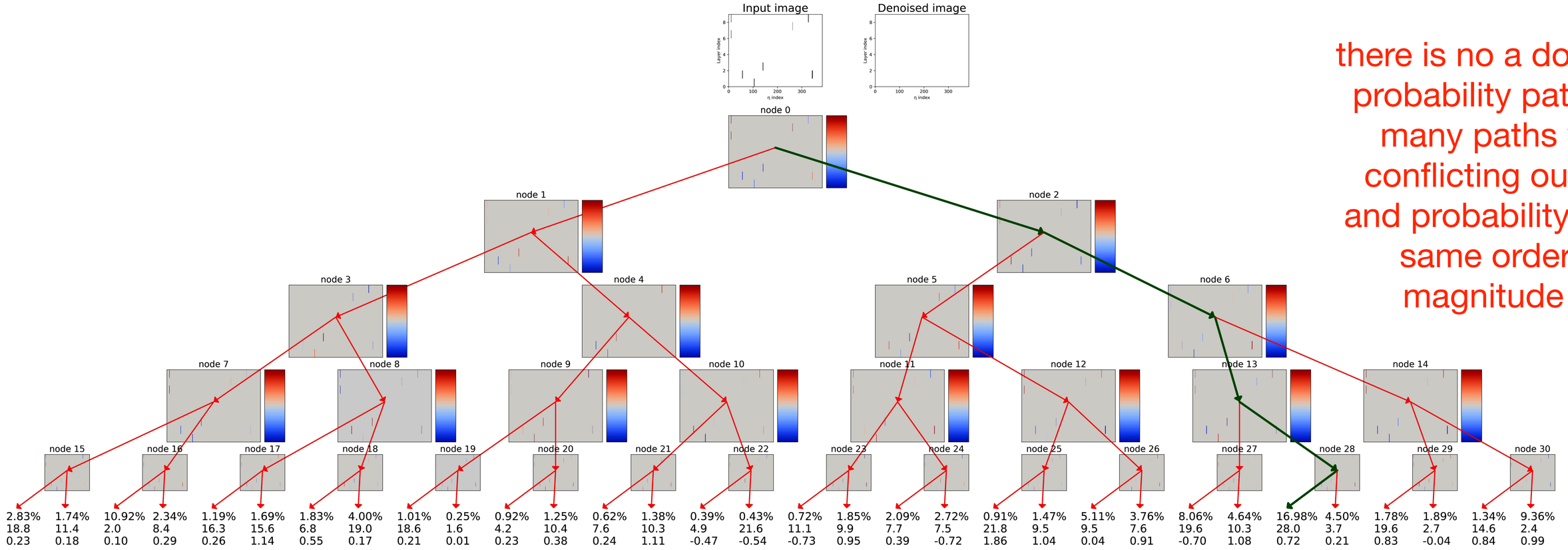
xAI VIA DISTILLATION TO CONVOLUTIONAL SOFT DECISION TREES

- teacher distilled to a intrinsically explainable student model: a decision tree (SDT: Soft Decision Tree)
- SDTs weights all nodes according to the probability to reach every leaf → instead of taking hard and exclusive decisions all nodes actively contribute to the model's final prediction
- ideas improved with Convolutional layers on top, to provide a latent representation of the input data to be passed to the hierarchical mixture of the trees

Real: [p_T=0.0 GeV, η=0.00] Predicted: [p_T=12.7 GeV, η=0.40]

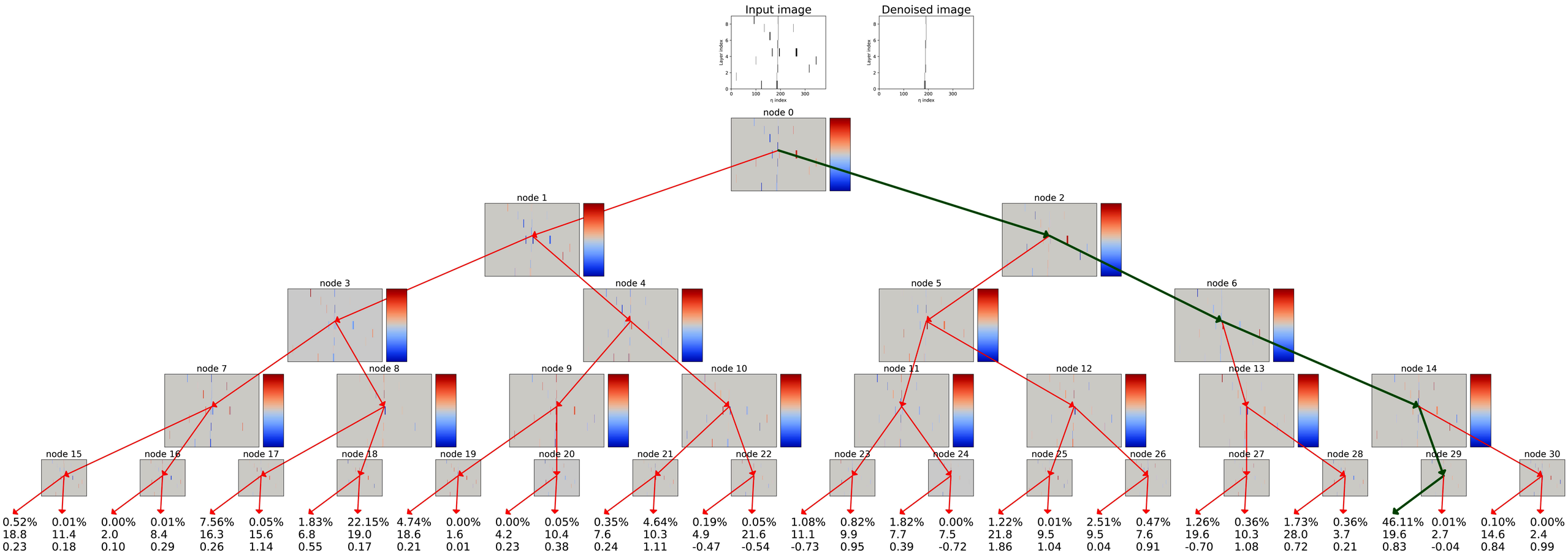
False Positive Example

there is no a dominant probability path, but many paths with conflicting outputs and probability of the same order of magnitude …



A True Positive Example

Real: [p_T=18.3 GeV, η=0.57] Predicted: [p_T=17.8 GeV, η=0.56]



in this example a single path (76.72% probability) dominate the muon momentum and pseudorapidity predictions

xAI VIA TRAINING INFLUENCE

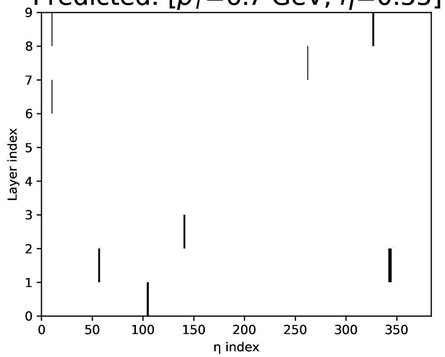
- TracIn leveraged to find the training examples most relevant for a given prediction on a test set event

Proponents: training examples that have reduced the loss at training time and are positively correlated with the sample to explain

Opponents: examples that have increased the loss, and are negatively correlated with the sample to explain

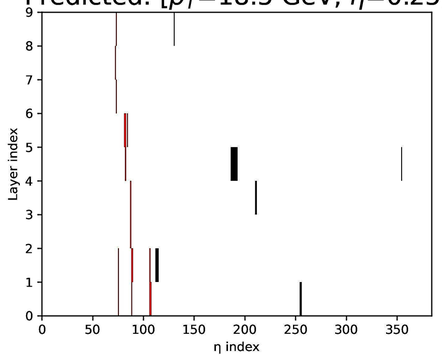
Input image

Real: [p_T=0.0 GeV, η=0.00]
Predicted: [p_T=6.7 GeV, η=0.55]

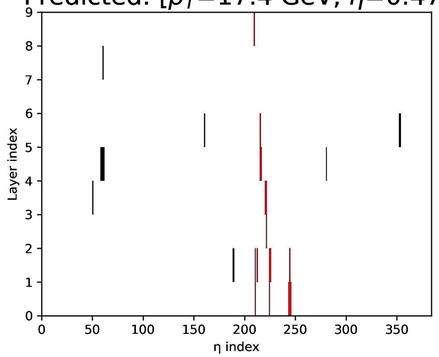


Proponents

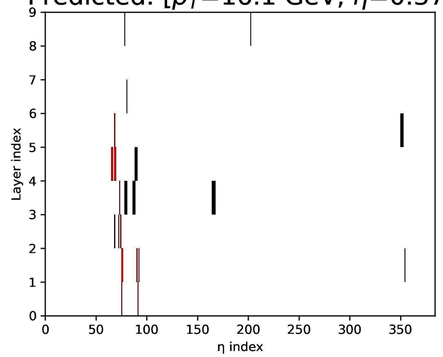
Real: [p_T=5.6 GeV, η=0.26]
Predicted: [p_T=18.5 GeV, η=0.25]



Real: [p_T=5.3 GeV, η=0.62]
Predicted: [p_T=17.4 GeV, η=0.47]

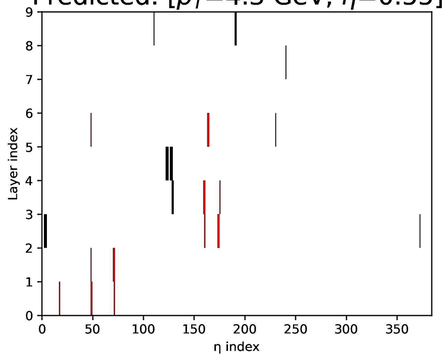


Real: [p_T=4.8 GeV, η=0.25]
Predicted: [p_T=16.1 GeV, η=0.37]

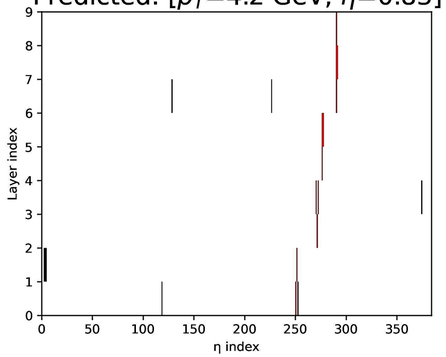


Opponents

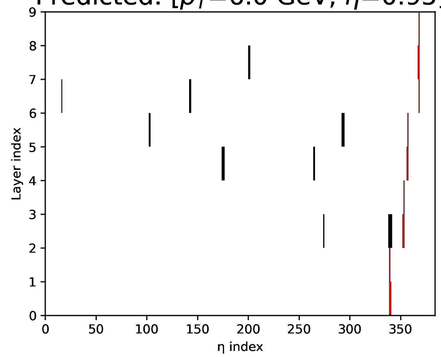
Real: [p_T=19.4 GeV, η=0.01]
Predicted: [p_T=4.3 GeV, η=0.55]



Real: [p_T=19.7 GeV, η=0.84]
Predicted: [p_T=4.2 GeV, η=0.83]



Real: [p_T=19.7 GeV, η=0.99]
Predicted: [p_T=6.0 GeV, η=0.95]



False Positive Example