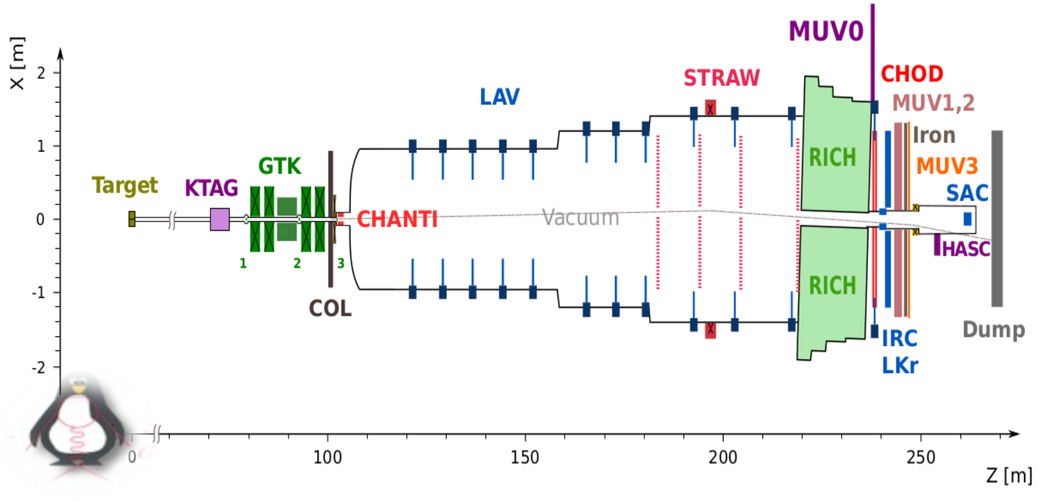
Explainable Artificial Intelligence (XAI) in HEP MUCCA Project

Joe Carmignani

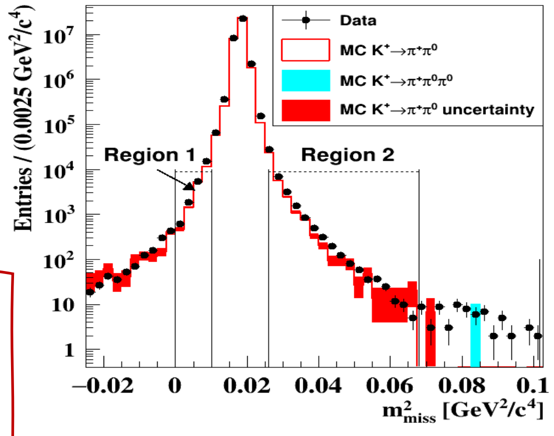
@UoL since December2021

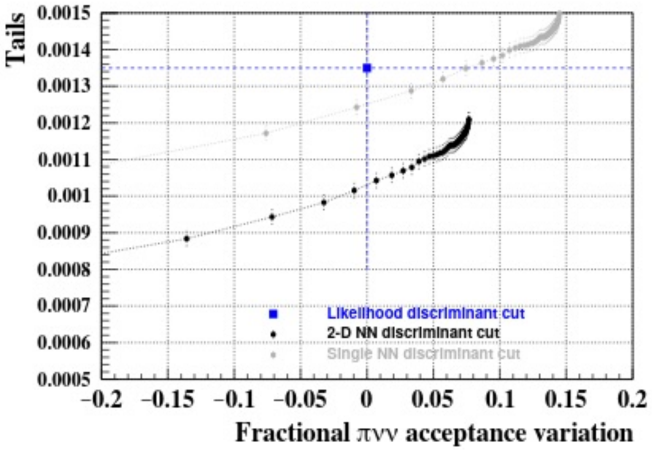
Previously on NA62 (PhD)

Carried out work on Neural Network Model + XAI-NPUTS and applications to NA62 analysis



- NA62 aims to measure precisely the BR of $K^+ \rightarrow \pi^+ \nu \bar{\nu}$
- Background must be kept extremely low due to Open Kinematics in m2miss
- A proper Track matching is a must





- One Instance: Track Matching Metrics to Tame dominant Background

1. Tails: the final fraction of $K^+ \rightarrow \pi^+\pi^0$ events entering signal regions of $\pi\nu\nu$
2. Fractional Acceptance Variation: The relative difference between the number of normalization events selected in the standard analysis and the ones selected using the NN K−π matching algorithm

- Best Values for Tails (Lowest) with cuts applied on both 2-D NN discriminant and FAV (NN-based High level Variable)
- We obtained 35% lower $\pi^+\pi^0$ background
- XAI metrics helped ameliorate Signal Acceptance

Now ATLAS in MUCCA: CHIST-ERA Project

Multi-disciplinary Use Cases for Convergent new Approaches to AI explainability

Monica D'Onofrio (PI), Cristiano Sebastiani and myself since Dec2021

Goal: *quantifying strengths* and solving weaknesses of new and state of the art XAI methods

Strategy: study XAI in *heterogenous use cases* from High Energy Physics (HEP), medical imaging, diagnosis of pulmonary, tracheal and nasal disease, neuroscience

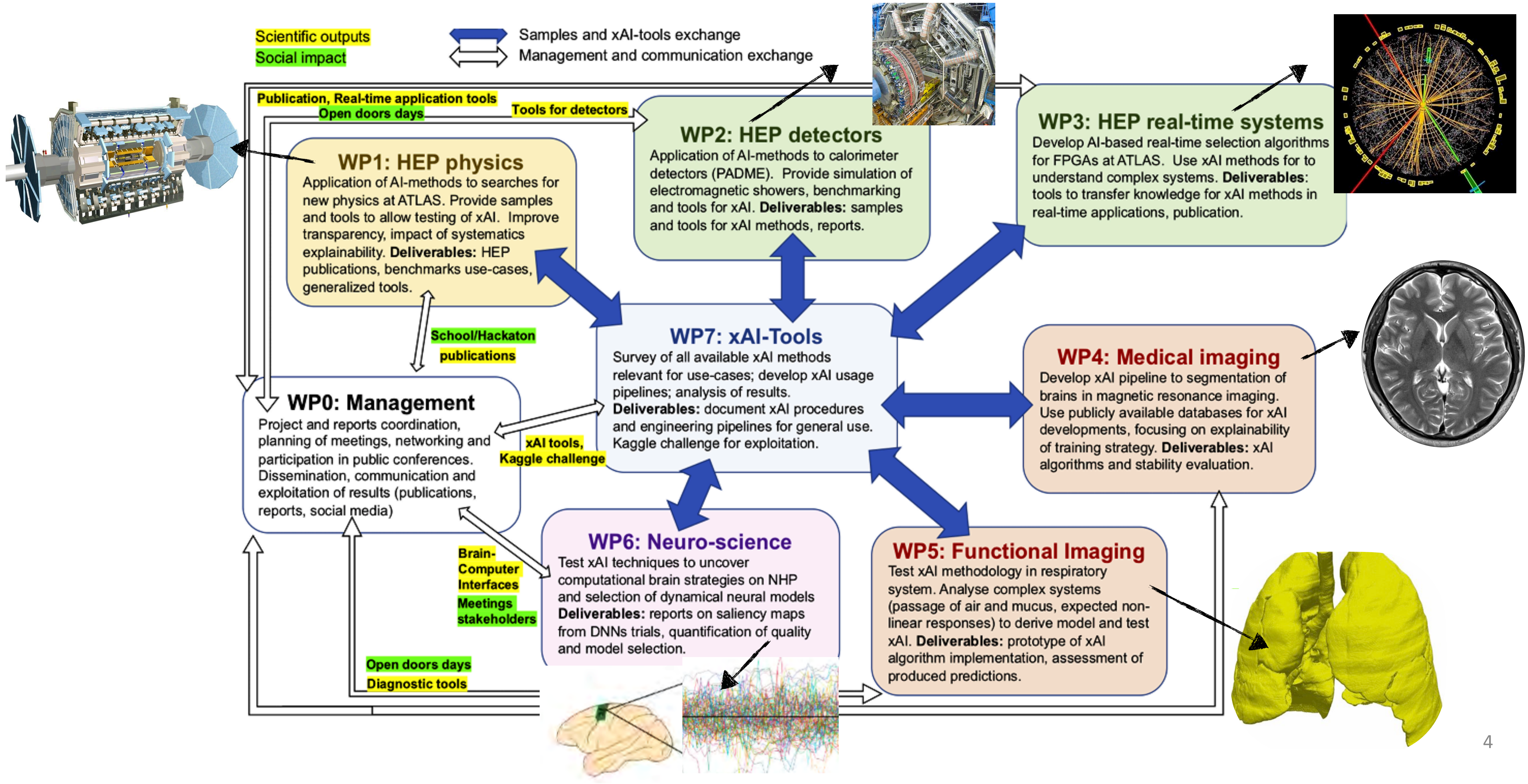
Three phases:

1. **Apply XAI-NPUT techniques**
2. **Identify shortcomings and metrics**
3. **Get new transparent algorithms**

CHIST-ERA 2022 1st Prize winning Video:

http://widgixeu-responseuploads.s3.amazonaws.com/fileuploads/90010018/90420529/140-2f7727dc2c9debd53715877981b60bce_MUCCA_videO_720p.mp4

MUCCA: Work Plan



HEP1-SUSY & DARK SECTOR

- Search for dark matter candidates resulting from the decay of new particles predicted by Supersymmetry the typical HEP case for ML classification (ATLAS analysis – by Hamish, now Dr, and Monica):
  - Extract small signal of interest from large SM background
  - Subtle/complex differences in variable correlations distinguish signal from background
  - Build ML discriminator to distinguish backgrounds from SUSY signals, trained on simulated Monte Carlo samples
  - Use classifier output score as discriminant variable for Hypothesis Testing (HIGH Level variables)

- Search for “dark” photons, LIGHT LL PARTICLES BELONGING TO A NEW HIDDEN SECTOR not yet discovered because too feebly interacting with ordinary matter (ATLAS analysis - by Cristiano, Alessandro – PhD, and Monica):
  - In this case, signal leaves different signature in the detector than background
  - ML discriminator use image classification trained to distinguish background processes from signal mapping clusters of hadrons (jets) in 3D coordinates

  → In order to extract information from the whole calorimeter, from a single jet 3D objects (Very Low Level)

More on the actual results from ATLAS in Cristiano's talk tomorrow

Approach to SUSY analysis

Analysis selects events with missing transverse momentum, a lepton and b-jets possibly coming from the decay of a Higgs boson

Tested multiple ML classifiers: BDT, NN

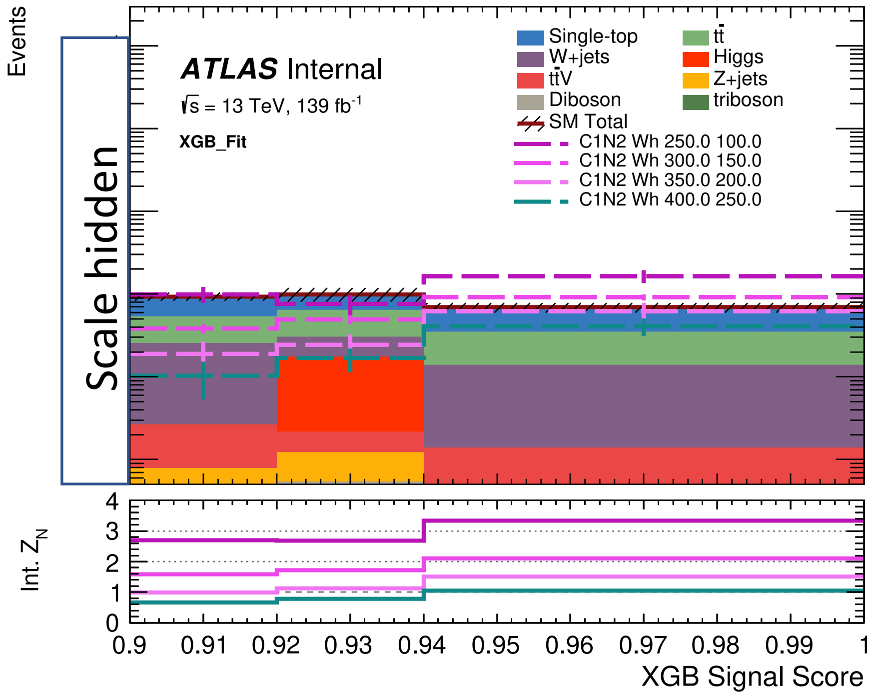
Use BDT (XGBoost) for reduced complexity, constructed from regression tree functions, using multi-classification with output scores containing the predicted probability of an event being in each class.

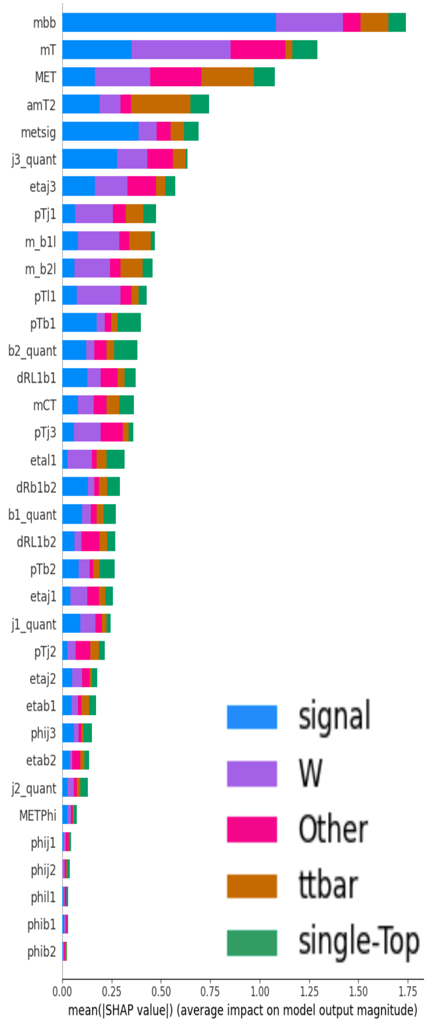
- **Used SHAP** (SHapley Adaptive exPlanations, 2017) to identify variables with largest impact for signal and that are most different when comparing simplified vs full reconstruction samples

✓ Me: Build eXplainable Graph data, Use GNNs and New SOTA Metrics

✓ Goal: Reduce dependencies on modelling from input variables

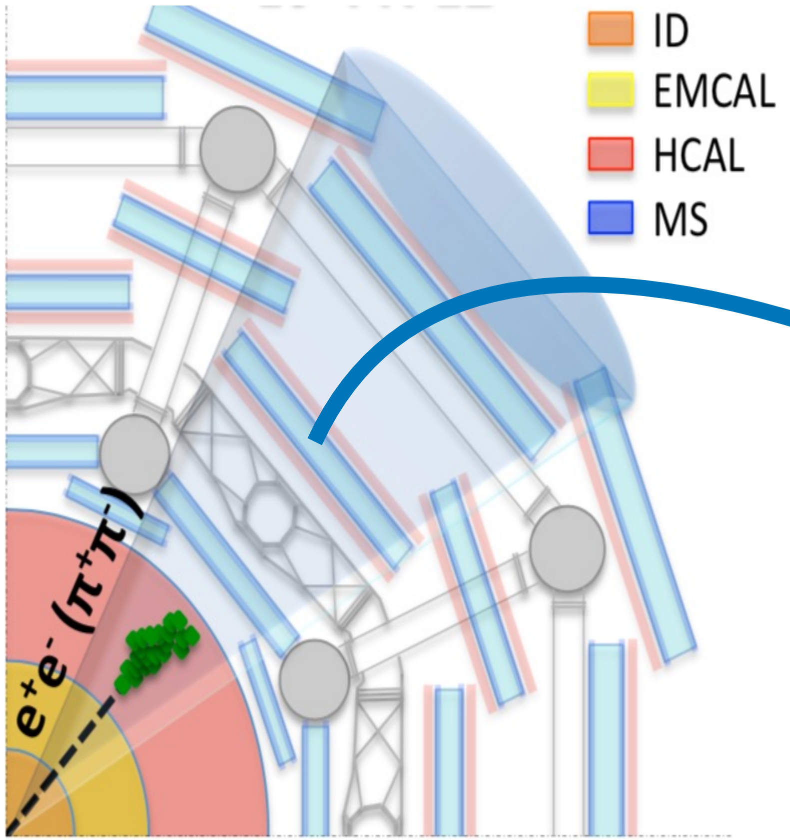
(more in Hamish' thesis)

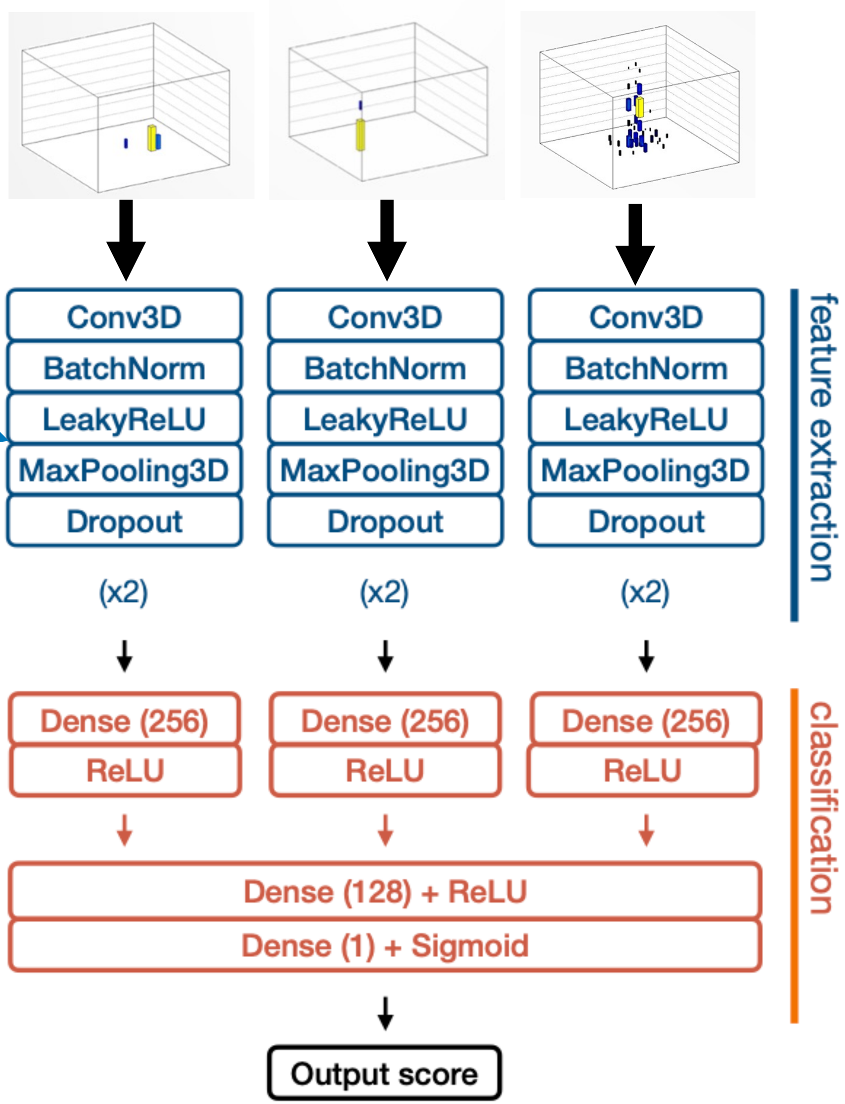




Dark Photon Analysis: 3D-CNN Inputs and structure

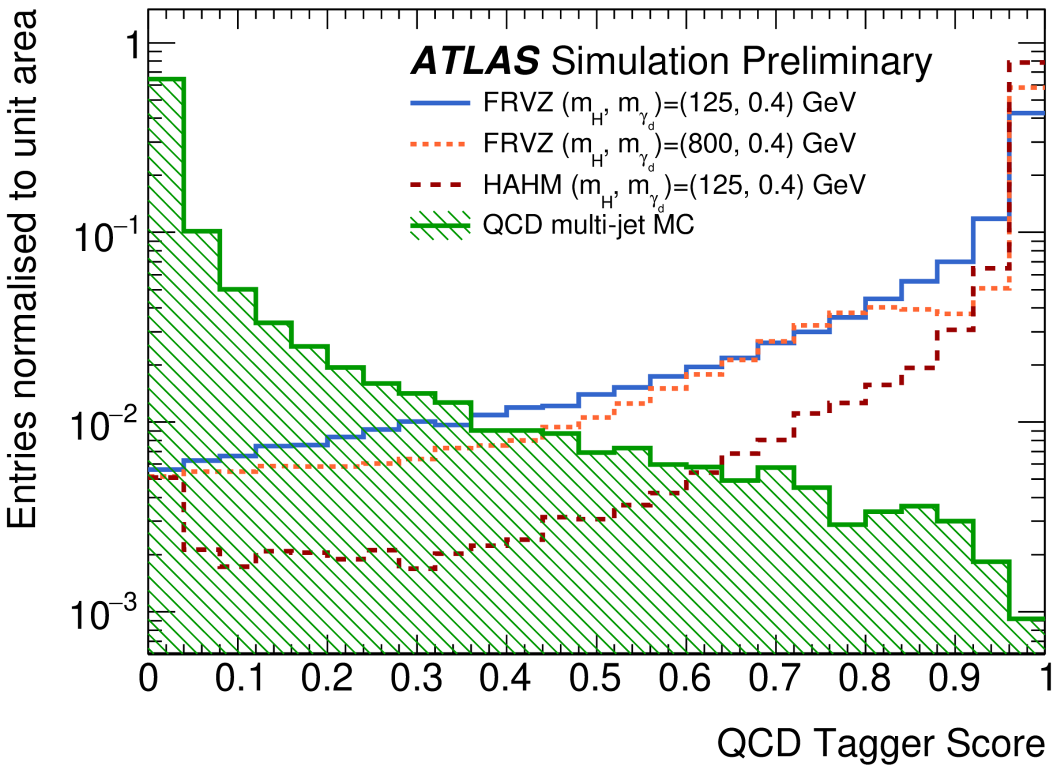
- Three 3D-CNNs processing low level information (images)
- Results combined to obtain **single output**
- Training datasets from MC events:
  - ~400k events for each dataset, signal and background

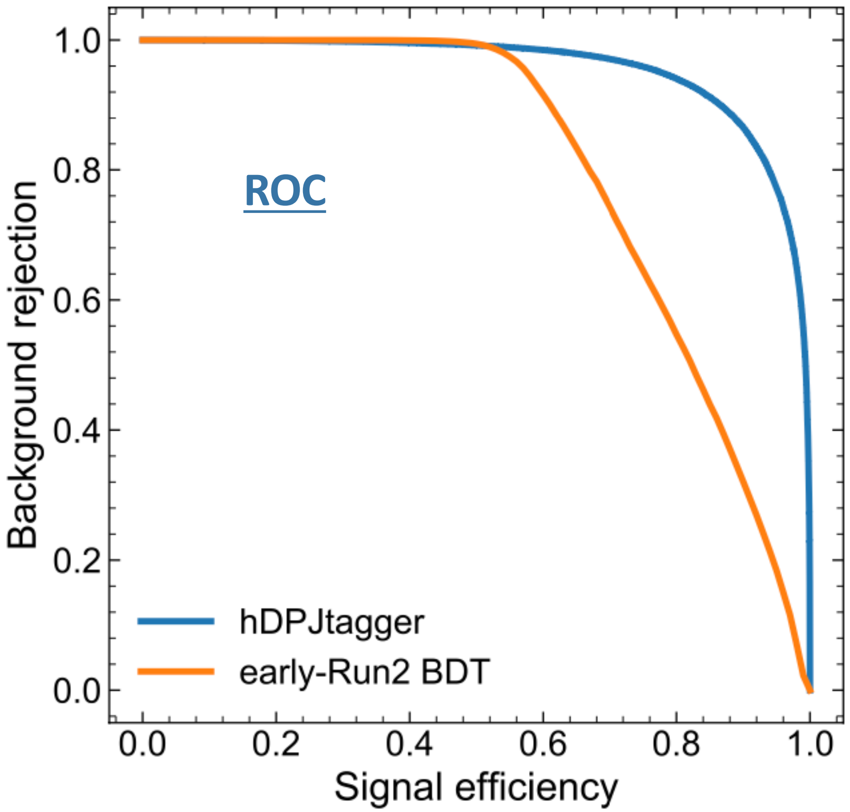




Cristiano, Alessandro - PhD, Monica
ATLAS-CONF-2022-001

Output score





Note: a Dense Neural Network (DNN) is also developed to discriminate signal from candidates that originate from the cosmic-ray background. The DNN is implemented using Keras with the Tensorflow backend and classifies each stand–alone object potential referrable to the signal based on low-level inputs including timing variables

Implementing Graphs

Dark Photon Jet MC for signal, and background from QCD jet ATLAS data with relevant kinematics.

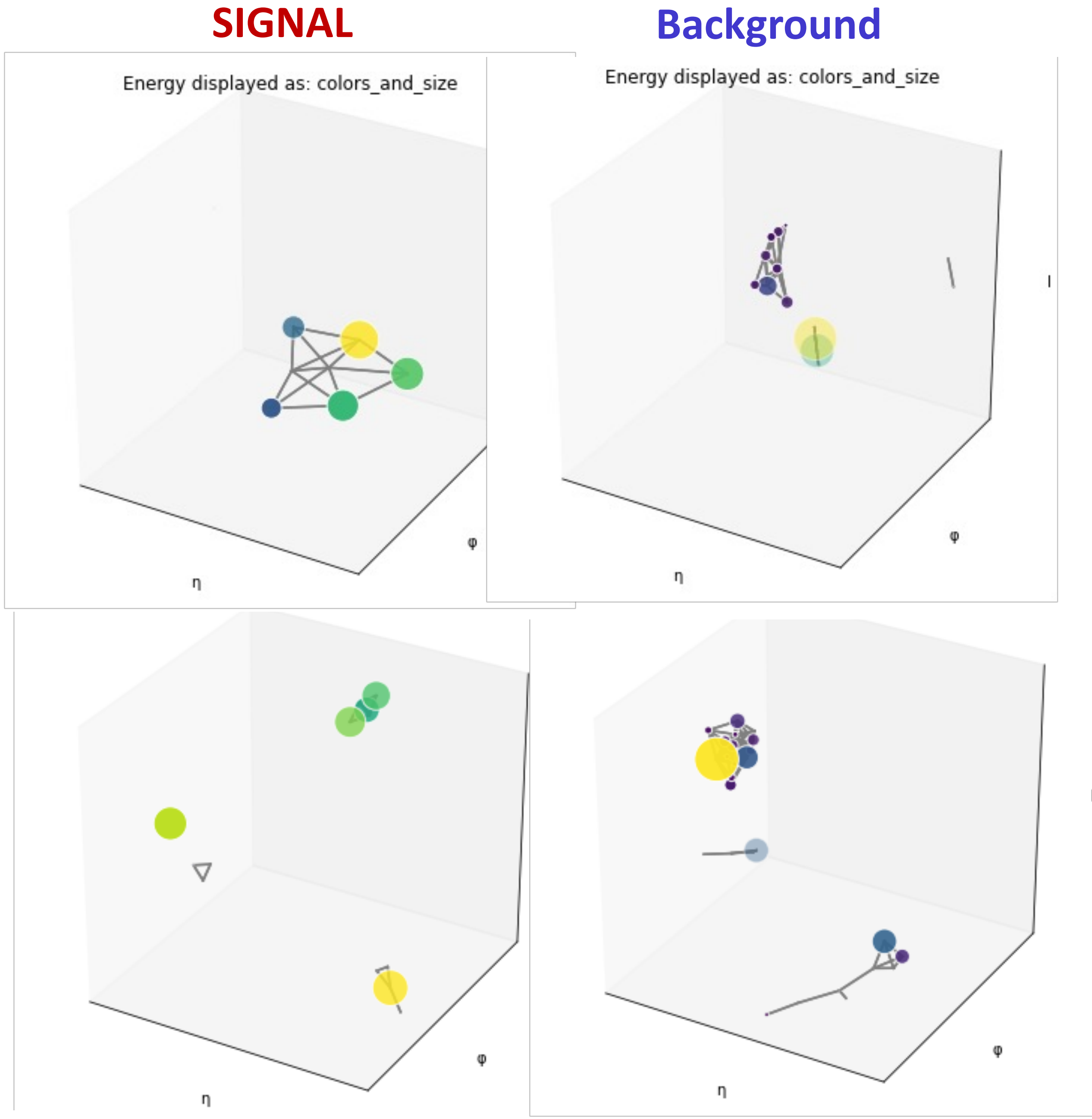
Nodes are individual clusters in all layers of calorimeter sampling

A single Attribute: Normalised Energy deposit/Cluster (max scaled)

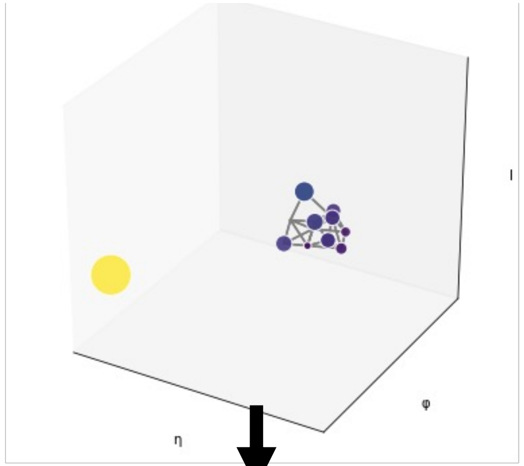
3-D coordinates to spread the nodes in the graphs accordingly (Eta, Phi, Sampling Layer)

"Hertz Probability Distribution" and TaxiCab metric were used in the radius threshold of Networkx "geometric graph" generator

MP potential: Building the edges with covariant distance as weight (p=2 norm between nodes)

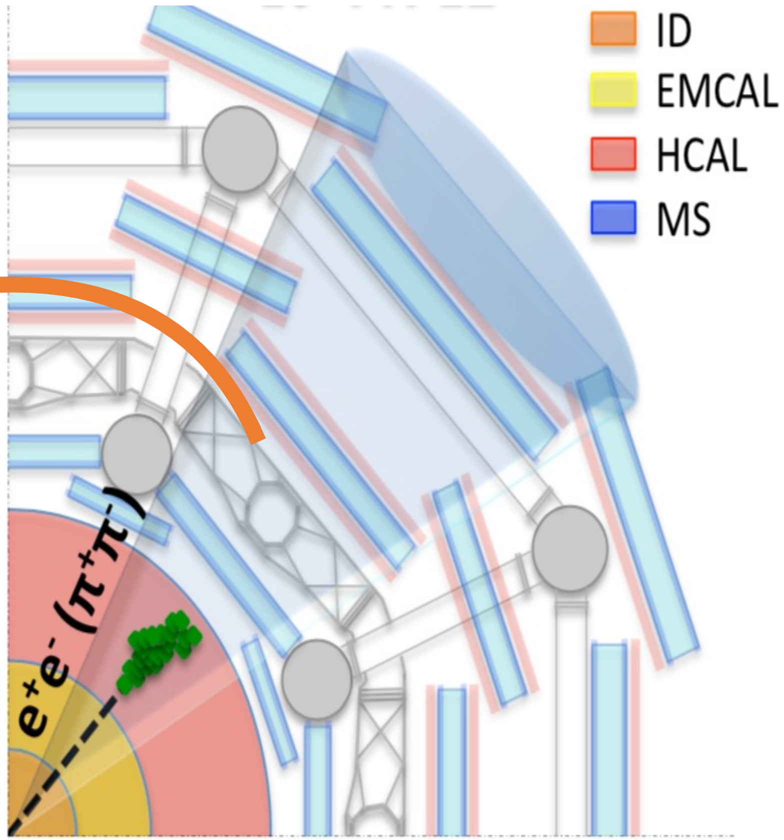


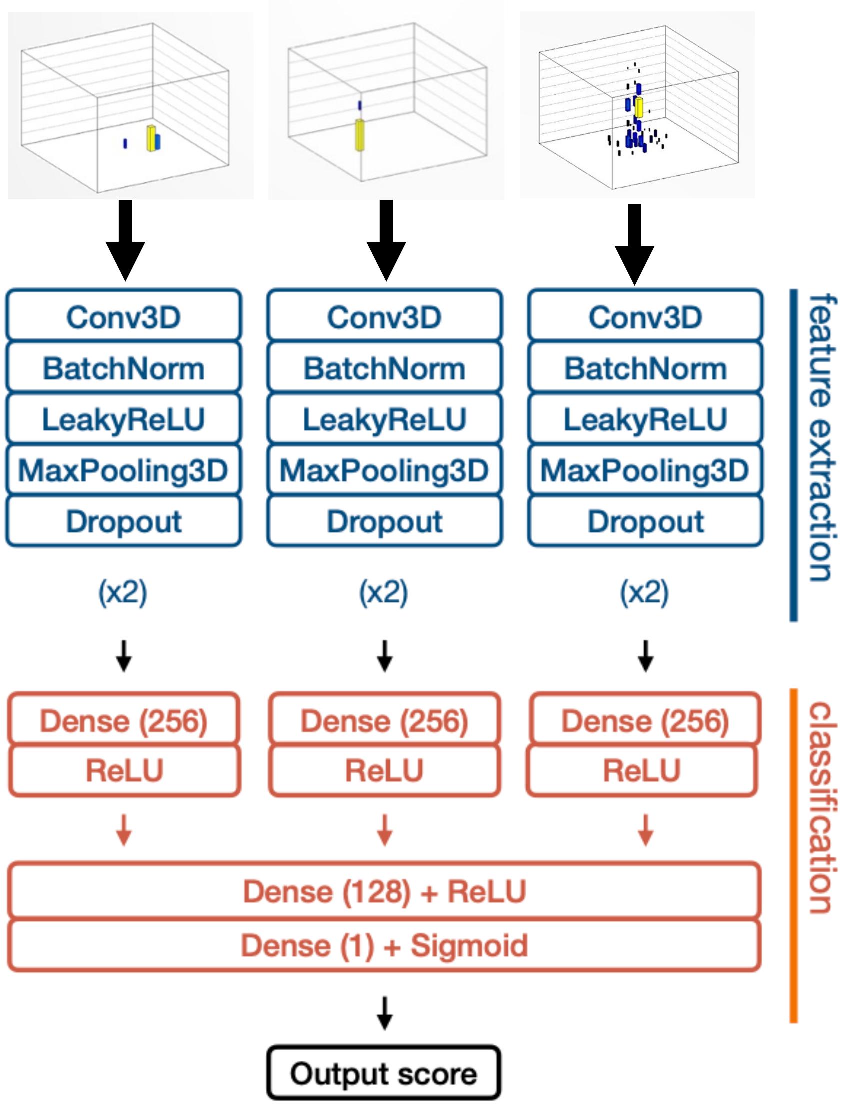
Dark Photon Analysis: ResGNN Inputs and structure



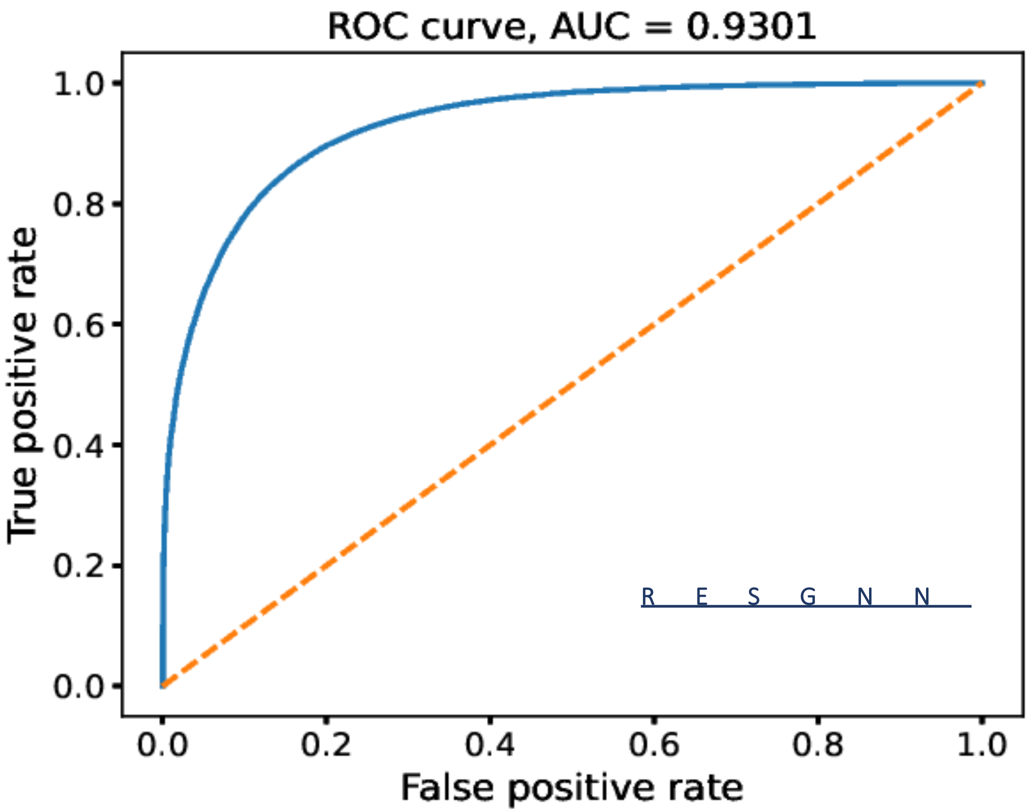
```
0 | conv1       | GraphConv         |
1 | conv2       | GraphConv         |
2 | conv3       | GraphConv         |
3 | conv4       | GraphConv         |
4 | conv5       | GraphConv         |
5 | conv6       | GCNConv           |
6 | conv7       | GCNConv           |
7 | lin         | Linear            |
8 | linout      | Linear            |
9 | loss_module | BCEWithLogitsLoss |
```

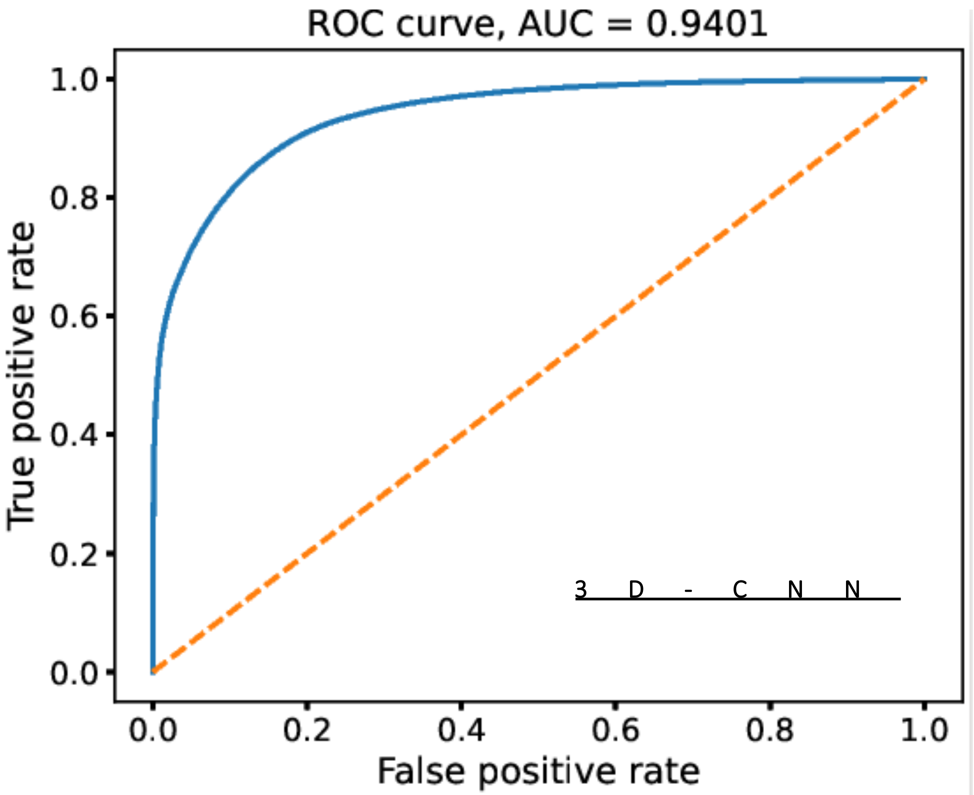
Output score





Demo: MPL ResGNN





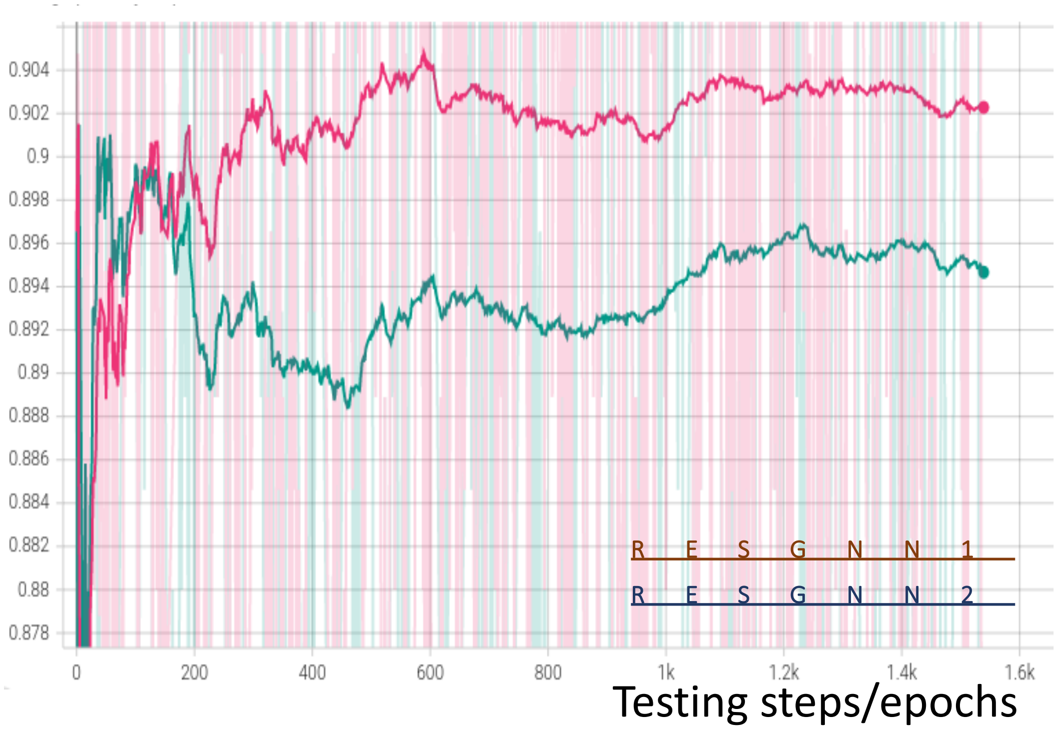
- ROC curves Nearly the same

Only 1% less for ResGNN

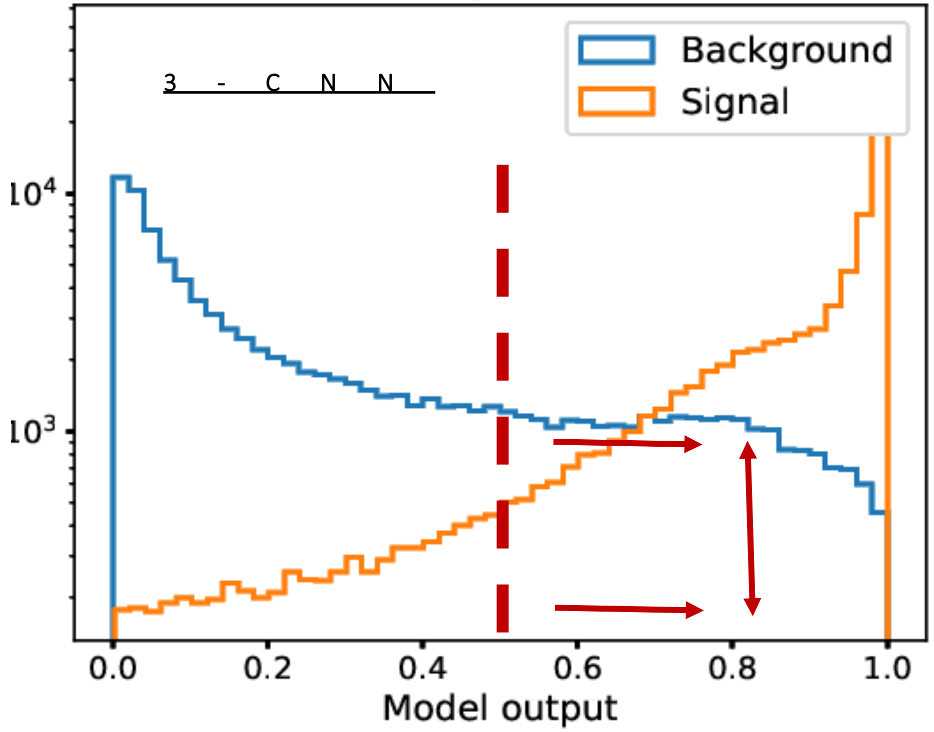
Demo: MPL ResGNN

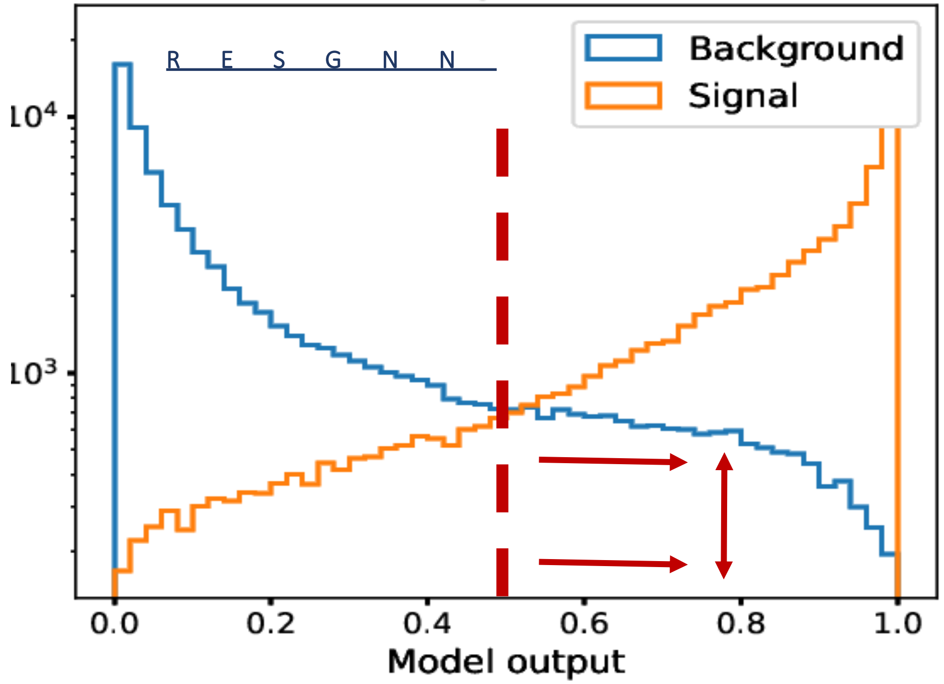
$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

90% TNR (True Negative Rate)
Or Background Purity



- 1 order of magnitude lower background tail overlapping Signal region (Less Mistag for ResGNN)

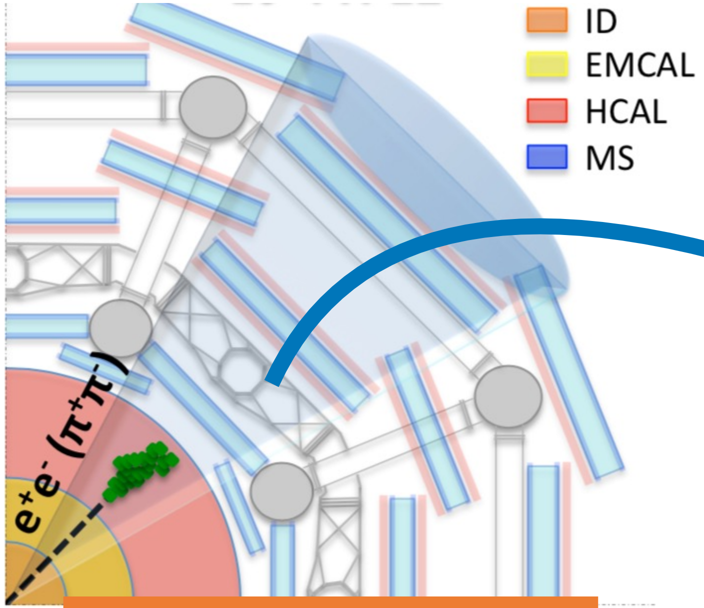




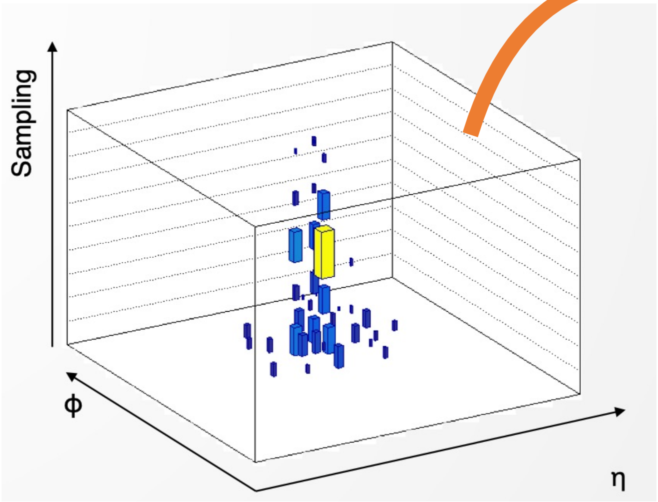
Low level inputs for jet discrimination

Extract low level information from the calorimeter geometry by singling out jets in either 3D images or graphs

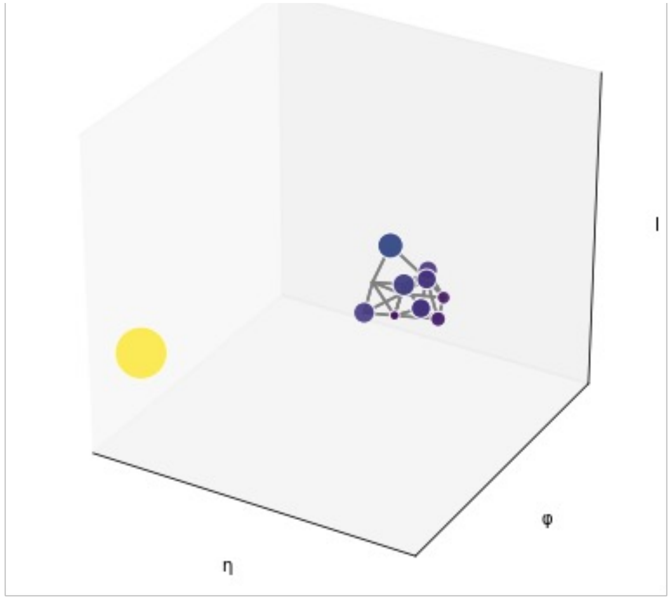
The ATLAS detector orthogonal view



Exploit the calorimeter granularity to parametrize the energy deposits: x, y, z, energy



3D jet images:
- Train a CNN used as reference for the study
- Very sparse images -> sub-optimal



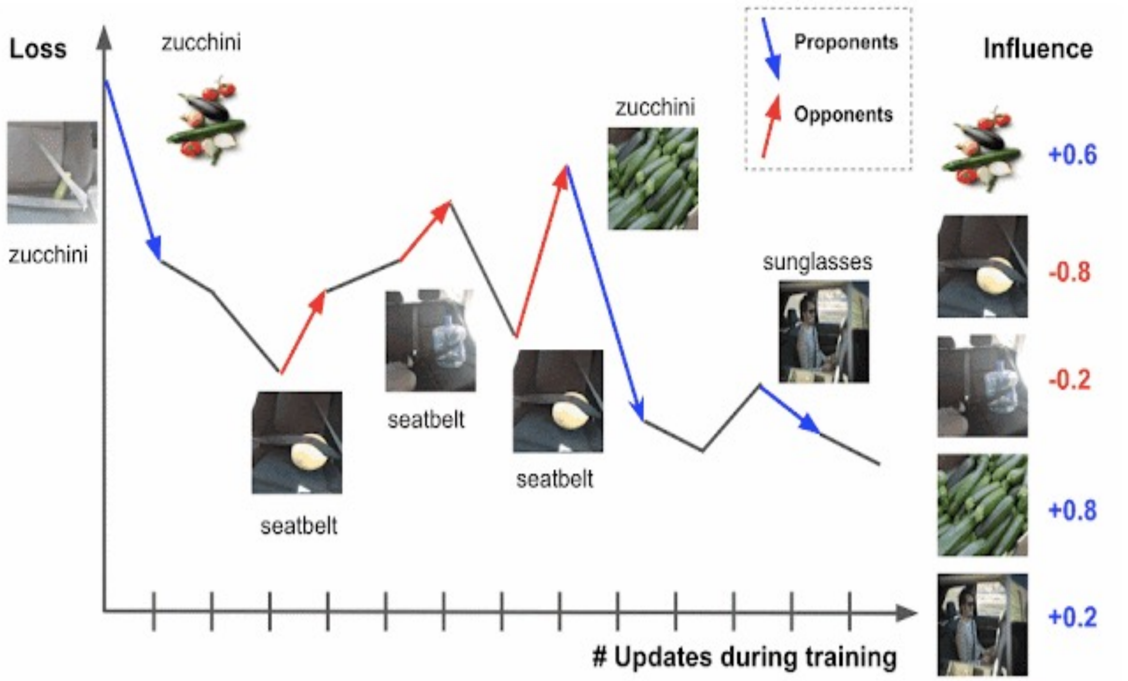
Graphs for XAI:
- **Train a fully optimized GNN**
- **Small cloud space objects**
- **Super Efficient Database and easy to manipulate**

Additional higher level variable can can be added as features to further improve the network performance, although the goal is to have them already 'learned' by the network by using only the low level inputs

Towards the X in X-AI

- Use innovative metrics sensitive to small changes in the input like TRAC-IN and Data-Models Implementation (see backup for references)
- To do so, we will focus on the GNN optimisation to fully exploit the input features and network capabilities:
  - Optimise graph attributes/weights to best balance (Performance vs. Computation)
  - Try other modules like Attention module with GATv2CONV Layers
  - Systematically train homogeneous modules Grid SWEEP-like Hyperparameter Tuning
- Use explainer layers: return subgraphs and/or subsets that mostly contribute to the prediction. (Captum packages for these metrics developed and added by WP7)

A typical workflow with Trac-in



https://ai.googleblog.com/2021/02/tracin-simple-method-to-estimate.html

Plans and next steps

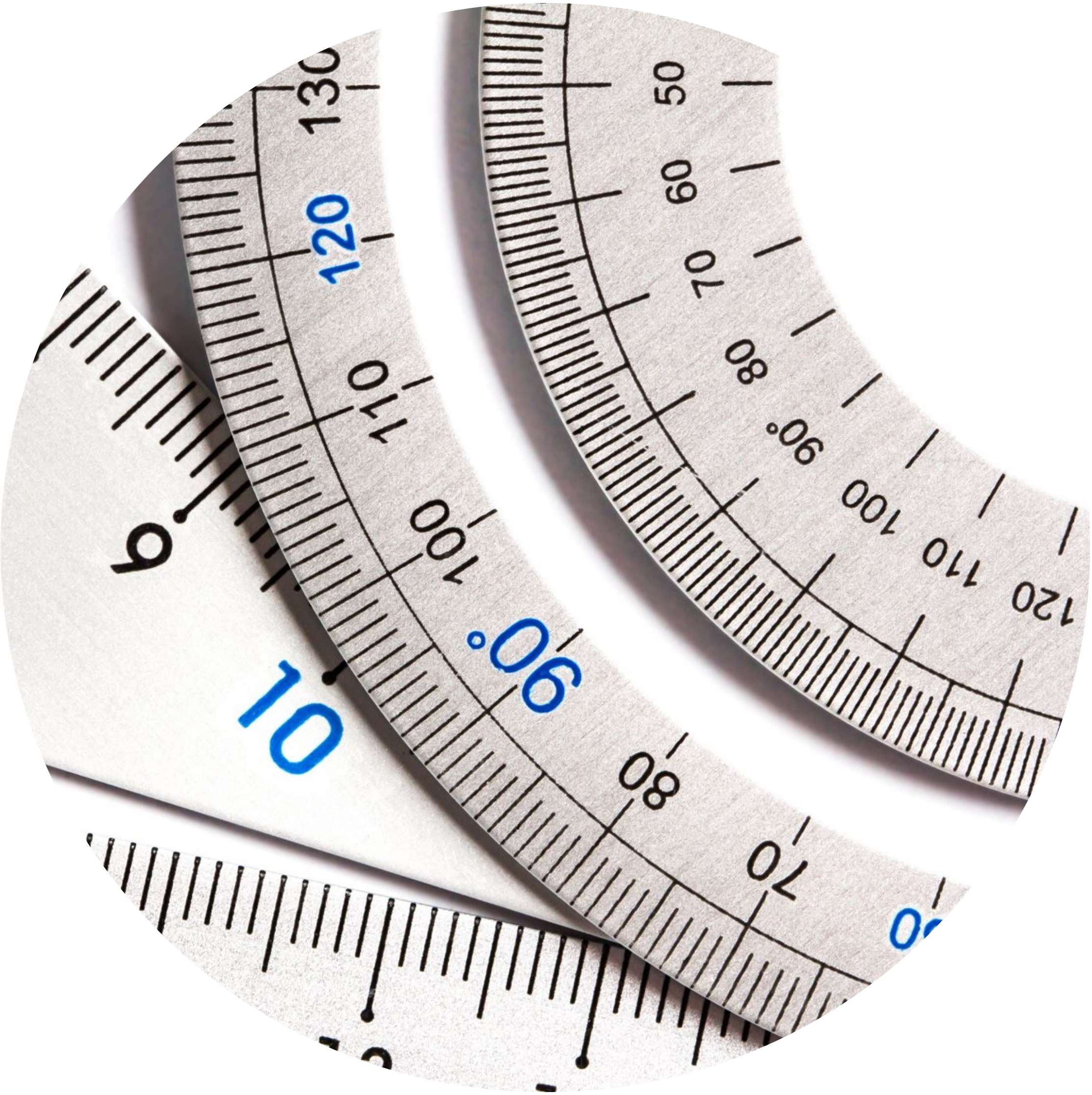
- Build a best optimised graph dataset and test a first GNN (with MPL ResGNN-like) implementation using only the same information exploited by the reference CNN
  - One-to-one performance comparison between the two
  - relate jet images directly to graphs to help explain the GNN predictions for a better AI explainability (e.g, understand background jets predictions in more detail)
- Rerun ATLAS CNN-based analysis with the new GNN to assess the improvement and publish open data to reproduce the study documented in a pub note (Service Task)
- Consider larger samples and apply similar approaches to SUSY case study
- Converge with all WPs to obtain a single XAI tool suitable for all cases

Dissemination:

- scientific publication, conferences
- open access toolkit
- Hackathon/School at Liverpool...

Multiple level impact:
1. Enable users to better understand XAI models and diagnosis limitation
2. Systematic understanding of which XAI methods better adapt to most applications
3. Skill development and training for young researcher

T E C H N I C A L S L I D E S



The Consortium

Sapienza University of Rome (IT)
Departments of Physics, Physiology, and Information Engineering

HEP: data-analysis, detectors, simulation AI: ML/DL methods in basic/applied research and industry, intelligent signal processing. Neurosciences: brain encoding of complex behaviours, ML in electrophysiology, multi-scale modelling approaches

Istituto Nazionale Fisica Nucleare (IT)
Rome group

Fundamental research with cutting edge technologies and instruments, applications in several fields (HEP, medicine imaging/diagnosis/prognosis/therapy)

Medlea S.r.l.s (IT)

High tech start-up, with an established track record in medical image analysis and high-performance simulation and capabilities of developing and deploying industry-standard software solutions

University of Sofia St.Kl.Ohridski (BG)
Faculty of Physics

extended expertise in detector development, firmware, experiment software in HEP

Polytechnic University of Bucharest (RO)
Department of Hydraulics, Hydraulic Equipment and Environmental Engineering

Complex Fluids and Microfluidics expertise: mucus/saliva rheology, reconstruction and simulation of respiratory airways, AI applications for airflow predictions in respiratory conducts

University of Liverpool (UK)
Department of Physics

physics data analysis at hadron colliders experiments, simulation, ML and DL methods in HEP

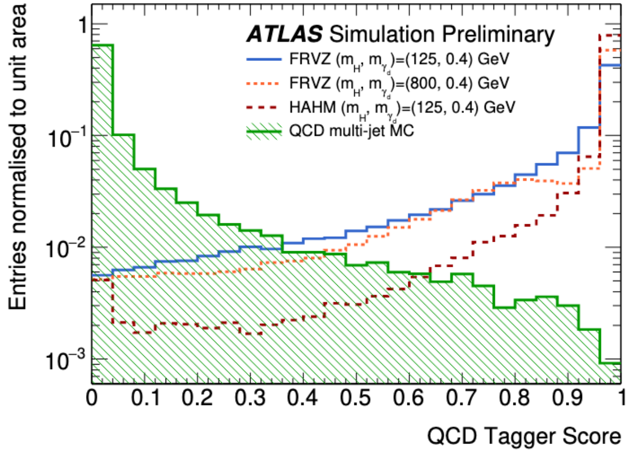
Istituto Superiore di Sanità

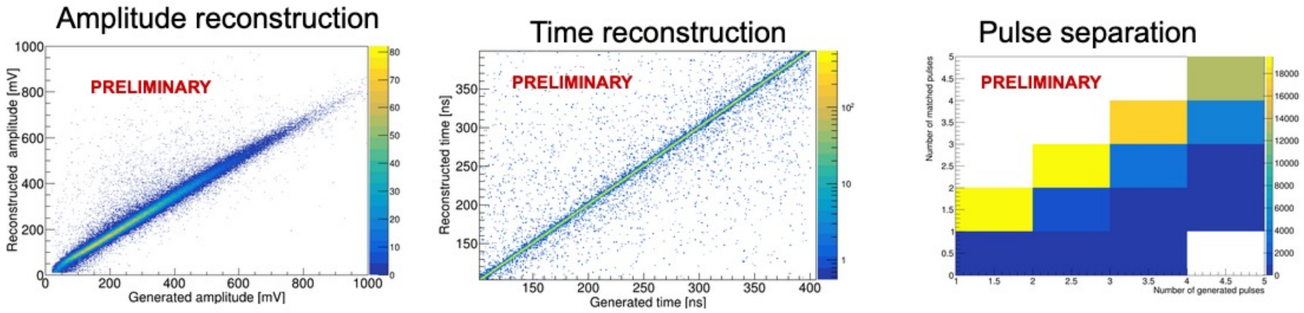
expertise in neural networks modeling, cortical network dynamics, theory inspired data analysis

HEP Use-Cases

WP1: *developed AI algorithms* (CNN, Graph NN), *targeted to event classification* and process discrimination, for new physics and dark matter searches at ATLAS. *First review of suitable state-of-art xAI* algorithms performed

Public report atlas-conf-2022-001

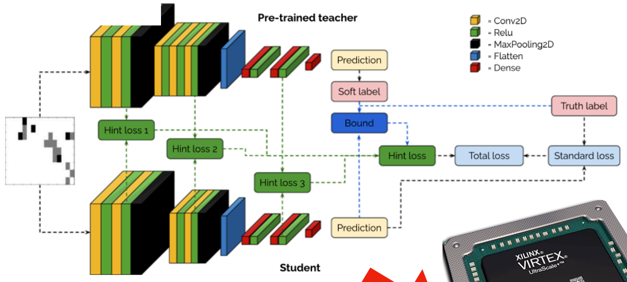




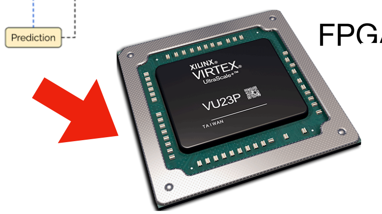
WP2: *AI algorithms* (CNN, autoencoder), *successfully developed* and applied to identify pulses, determing amplitude and time of arrival in close to reality simulated data of the PADME calorimeter

WP3: *developed complete pipeline for an AI based event selection* algorithm to expand physics potential of the ATLAS experiment. CNN model with compression and simplification strategies to make easier to interpret, and faster to execute the AI model, for the conversion and implementation in the firmware of FPGA accelerators. Obtained CNN inference in 80/150ns/image

Eur. Phys. J. C 81, 969 (2021)



model compression



FPGA

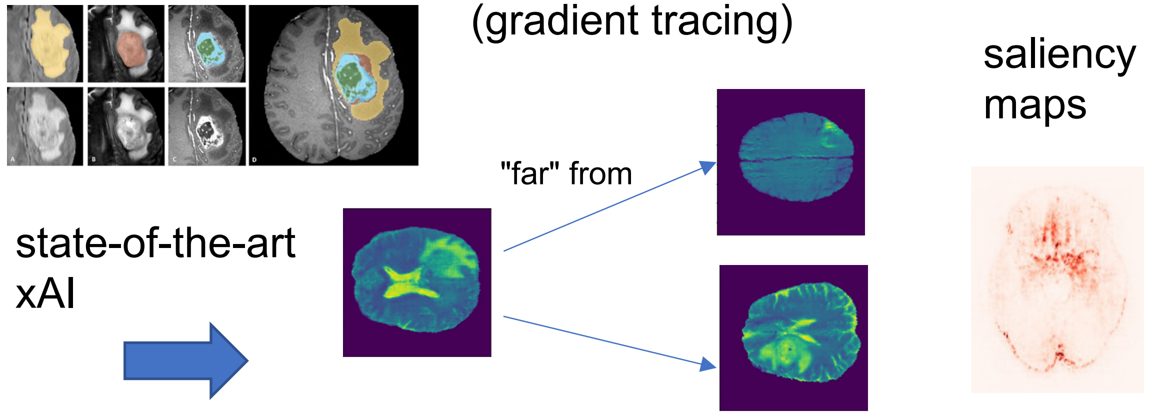
MED and NS Use Cases

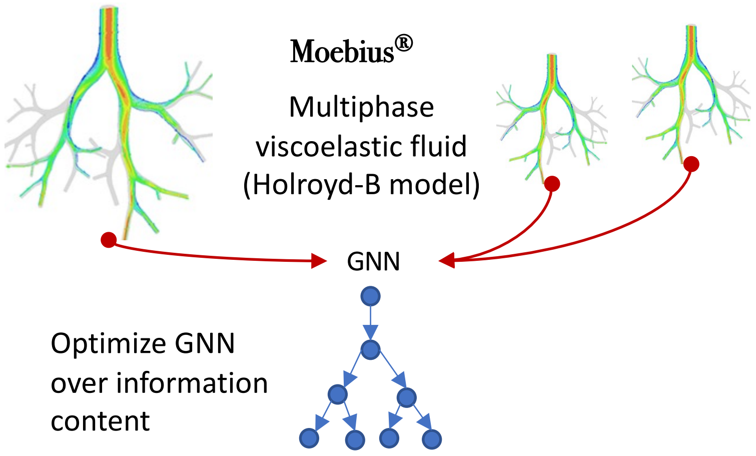
WP4: *Implemented AI models for the brain lesion segmentation* in the Brats17 MRI dataset (Unet2D, Resnet 3D). Data augmentation techniques to enhance performances tested. Selected state-of the art xAI algorithms, under implementation.

Brats17

training influence (gradient tracing)

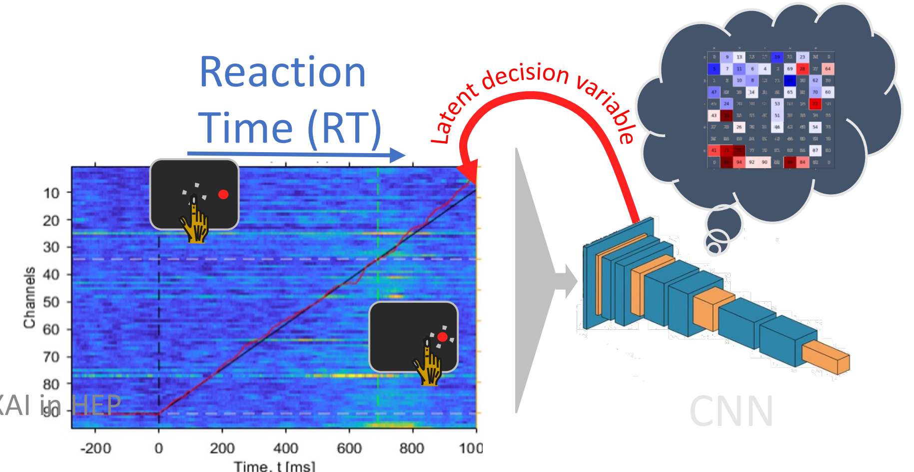
saliency maps





WP5: procedure for the realization of the prototypes of the trachea bifurcation (reconstruction of the geometry from the CT scan, numerical code) completed. Study of the GNN model for the simulation of the the air-flow

WP6: designed and realized a specific CNN (fed by electrophysiological signals) based on a ResNet to uncover an inner decision value increasing in time as a linear ramp eventually allowing to predict at single-trial level the onset timing of overt movements. Test of various xAI algorithms underway



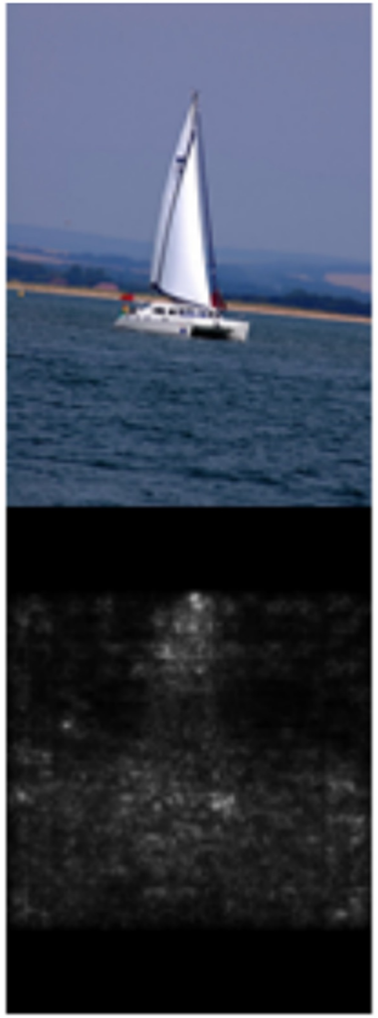
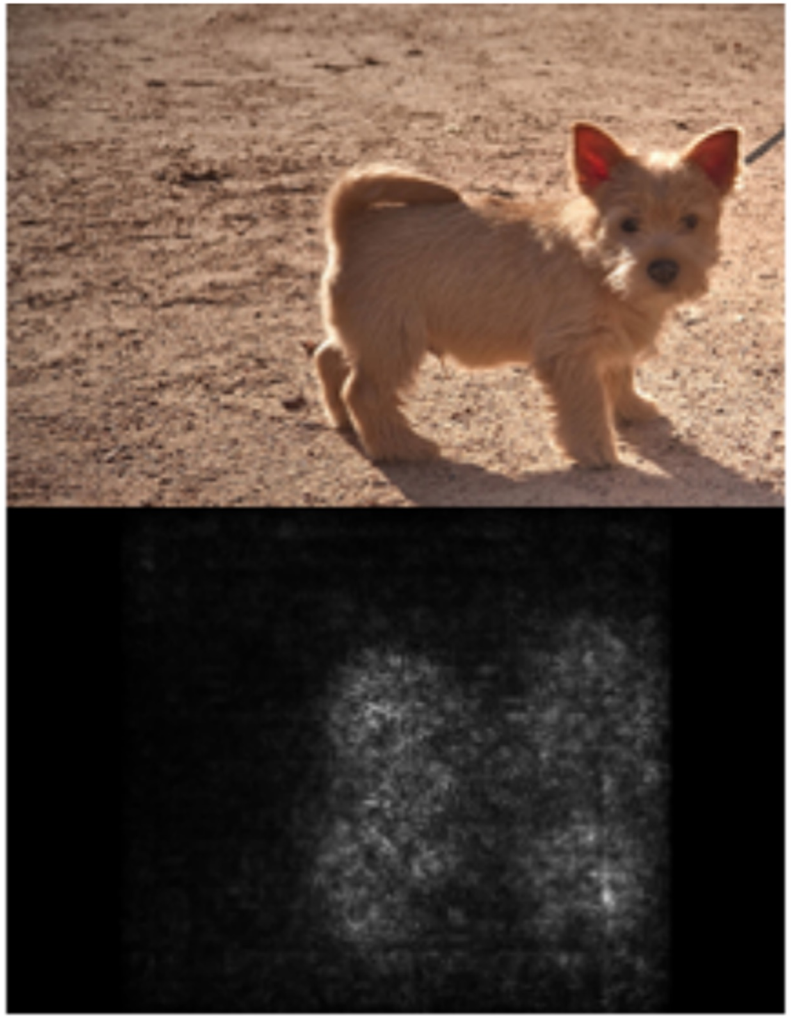
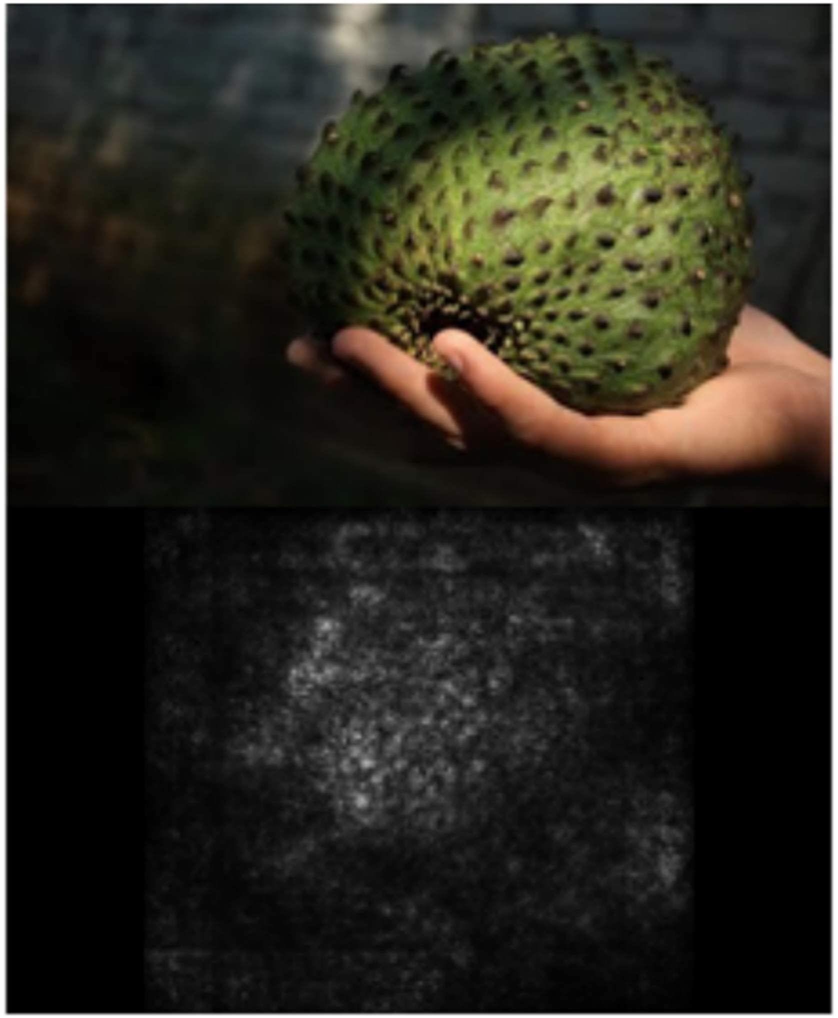
(Simonyan & al., 2013)

(Vanilla) Saliency maps

A **saliency map** is an object of the same dimensionality as the input, providing information about which features were most important for a given prediction.

Formally (*i* is the index of the class of interest):

$$\text{Saliency map} = \max_{\text{channels}} \left| \frac{\partial f_i(x)}{\partial x} \right|$$

Limits of saliency maps

Simple saliency maps have several issues that balances their simplicity:

1. They are highly **unstable** wrt small changes in the input.
2. They are not well **localized**.
3. They have no formal guarantees.

In particular, they do not respect a property called **sensitivity**: if two inputs differ for a single pixel but have different predictions, a saliency map is not guaranteed to highlight that pixel.

Gradient Tracing (**Pruthi & al., 2020**)

Consider an idealized training procedure where at iteration *t* we update the parameter vector as:

$$w_{t+1} = w_t - \eta \nabla l(w_t, z_t)$$

The **influence** of point z on point z' is defined as:

$$\mathrm{TracInIdeal}(z, z') = \sum_{t: z_t = z} l(w_t, z') - l(w_{t+1}, z')$$

Gradient Tracing (Pruthi & al., 2020)

By first-order approximation, it can be shown that:

$$\mathrm{TracInIdeal}(z, z') \approx \sum_{t: z_t = z} \eta \nabla l(w_t, z) \cdot \nabla l(w_t, z')$$

This can be approximated by storing *k* checkpoints during training and computing:

$$\mathrm{TracInIdeal}(z, z') \approx \sum_{i=1}^{k} \eta \nabla l(w_i, z) \cdot \nabla l(w_i, z')$$

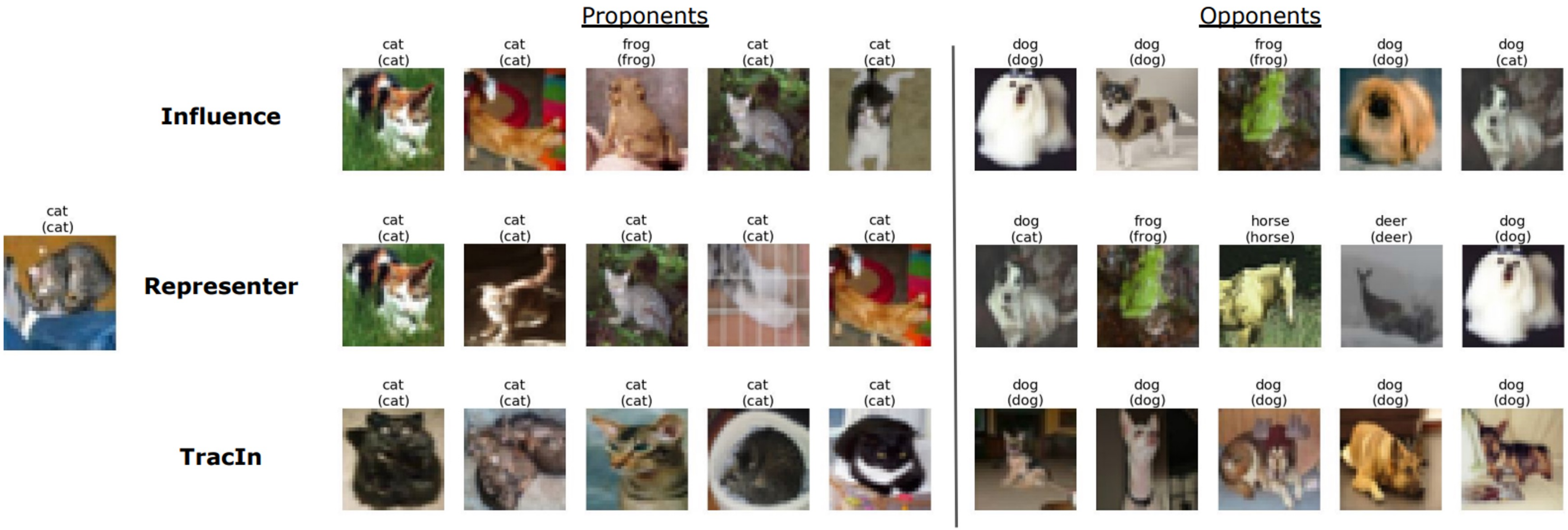


Figure 5: CIFAR-10 results: Proponents and opponents examples of a correctly classified cat for influence functions, representer point, and `TracIn`. (Predicted class in brackets)

Datamodels

(**Ilyas & al., 2022**)

Denote by f(x;S) the output of a network f on x after training on a set of data S. A **datamodel** is a model trained to approximate this function on a fixed x.

Suppose we sample uniformly subsets of the original training set, and train different models:

$$\{(S_1, f_{\mathcal{A}}(x; S_1)), \ldots, (S_m, f_{\mathcal{A}}(x; S_m))\}$$

Datamodels

Definition 1 (Datamodeling). *Consider a fixed training set S, a learning algorithm $\mathcal{A}$, a target example x, and a distribution $\mathcal{D}_S$ over subsets of S. For any set $S' \subset S$, let $f_{\mathcal{A}}(x; S')$ be the (stochastic) output of training a model on S' using $\mathcal{A}$, and evaluating on x. A datamodel for x is a parametric function g_θ optimized to predict $f_{\mathcal{A}}(x; S_i)$ from training subsets $S_i \sim \mathcal{D}_S$, i.e.,*

$$g_\theta : \{0,1\}^{|S|} \to \mathbb{R}, \qquad \textit{where} \qquad \theta = \arg\min_w \widehat{\mathbb{E}}^{(m)}_{S_i \sim \mathcal{D}_S} \left[\mathcal{L}\left(g_w(\mathbf{1}_{S_i}),\ f_{\mathcal{A}}(x; S_i) \right) \right],$$

$\mathbf{1}_{S_i} \in \{0,1\}^{|S|}$ is the characteristic vector of S_i in S (see (3)), $\mathcal{L}(\cdot,\cdot)$ is a loss function, and $\widehat{\mathbb{E}}^{(m)}$ is an m-sample empirical estimate of the expectation.

In practice, we can train linear datamodels:

$$g_\theta(\mathbf{1}_{S_i}) := \theta^\top \mathbf{1}_{S_i} + \theta_0$$

Implementing datamodels

A Pseudocode for Estimating Datamodels

Algorithm A.1 An outline of the datamodeling framework: we use a simple parametric model as a proxy for the entire end-to-end training process.

procedure ESTIMATEDATAMODEL(target example x, trainset S of size d, subsampling frac. $\alpha \in (0,1)$)
    $T \leftarrow []$ ▷ Initialize *datamodel training set*
    for $i \in \{1, \ldots, m\}$ **do**
        Sample a subset $S_i \subset S$ from $\mathcal{D}_S$ where $|S_i| = \alpha \cdot d$
        $y_i \leftarrow f_{\mathcal{A}}(x; S_i)$ ▷ Train a model on S_i using $\mathcal{A}$, evaluate on x
        Define $\mathbf{1}_{S_i} \in \{0,1\}^d$ as $(\mathbf{1}_{S_i})_j = 1$ if $x_j \in S_i$ else 0
        $T \leftarrow T + [(\mathbf{1}_{S_i}, y_i)]$ ▷ Update datamodel training set
    $\theta \leftarrow$ RUNREGRESSION(T) ▷ Predict the y_i from the $\mathbf{1}_{S_i}$ vectors
    return θ ▷ Result: a weight vector $\theta \in \mathbb{R}^d$