Computing and Software for Big Science (2023) 7:8
https://doi.org/10.1007/s41781-023-00102-z

RESEARCH

Convergent Approaches to AI Explainability for HEP Muonic Particles Pattern Recognition

Leandro Maglianella[1] · Lorenzo Nicoletti[1] · Stefano Giagu[2] · Christian Napoli[1,3,4] · Simone Scardapane[5]

Received: 20 January 2023 / Accepted: 28 July 2023

Abstract

Neural networks are commonly defined as 'black-box' models, meaning that the mechanism describing how they give predictions and perform decisions is not immediately clear or even understandable by humans. Therefore, Explainable Artificial Intelligence (xAI) aims at overcoming such limitation by providing explanations to Machine Learning (ML) algorithms and, consequently, making their outcomes reliable for users. However, different xAI methods may provide different explanations, both from a quantitative and a qualitative point of view, and the heterogeneity of approaches makes it difficult for a domain expert to select and interpret their result. In this work, we consider this issue in the context of a high-energy physics (HEP) use-case concerning muonic motion. In particular, we explored an array of xAI methods based on different approaches, and we tested their capabilities in our use-case. As a result, we obtained an array of potentially easy-to-understand and human-readable explanations of models' predictions, and for each of them we describe strengths and drawbacks in this particular scenario, providing an interesting atlas on the convergent application of multiple xAI algorithms in a realistic context.

Keywords Explainable Artificial Intelligence · High-energy physics · Saliency maps methods · Intrinsically interpretable Decision Trees · Tracing gradient descent

Introduction

The proliferation of Artificial Intelligence (AI) has led it to be widely used even in important decision-making situations in very different contexts, for instance in advertisement [20], transportation, healthcare, military [9, 33], finance and legal applications [1]. In this kind of critical domains, it is crucial for the used Machine Learning (ML) method to be *transparent*, meaning that it must be possible to understand the reasons behind its outputs: this is important not only to ensure human control on AI but also to comprehend more deeply the reasoning process modeled by a machine, enabling further improvements and discoveries. To address these desiderata, Explainable Artificial Intelligence (xAI) aims at finding and implementing techniques that can merge high performances and highly explainable capabilities to understand decisions made by ML algorithms [18]. Due to the fact that explanations are usually qualitative and subjective, and given the heterogeneity of currently available xAI tools, instead of limiting the research to one single xAI method, in this paper we explore an array of techniques addressable to a varied range of end-users, including both experts with prior knowledge and beginners with no preparation on the study field, introducing the concept of *convergent* approaches in xAI applied to a real-world scenario to guarantee wider and more comprehensive understandings of automatic models' decision process.

✉ Stefano Giagu
stefano.giagu@uniroma1.it

[1] Department of Computer, Control and Management Engineering Antonio Ruberti, Sapienza University of Rome, Rome, Italy

[2] Department of Physics, Sapienza University of Rome, Rome, Italy

[3] Institute for Systems Analysis and Computer Science, Italian National Research Council, Rome, Italy

[4] Department of Computational Intelligence, Electronics and Telecommunications, Czestochowa University of Technology, Czestochowa, Poland

[5] Department of Information Engineering, Electronics and Telecommunications, Sapienza University of Rome, Rome, Italy

Published online: 03 August 2023

Use-Case: HEP and Muons

Our domain concerns a high-energy physics application to analyze particle trajectories detected in muon spectrometer detectors. We adopt as use-case a Level-0 trigger system of one of the Large Hadron Collider (LHC) experiments at CERN [12] for the high luminosity phase of the LHC (HL-LHC). This contribution extends a previous research from [14], an application studying muonic patterns and focusing on generating real-time models through compression and distillation techniques: its innovation concerns the study of suitable explainability methods applicable to muon pattern recognition. Starting from the specifications already defined in [14], we used the toy simulation of the detector and trigger response developed in that work to simulate and record trajectories and signal released on the RPC detector of the ATLAS muon spectrometer for high-energy muon particles, unstable subatomic particles representing much of the cosmic radiation reaching the surface of the Earth. The simulation includes realistic effects related to resolution and noise.

The criticality of this domain consists in estimating particle parameters while being able to efficiently distinguish muonic events from noisy observations due to random hit background in the detector produced by electronic noise, and beam-induced background sources. The objective of this work is, consequently, to accurately estimate the measures associated with patterns and simultaneously to provide understandable explanations for models' predictions.

Contributions of This Work

Explainability studies have received increasing interest in the last few years. This work provides a new and challenging applicability scenario, guaranteeing interpretations of ML decisions in estimating and recognizing muonic particle patterns. This research has involved a variegate array of xAI techniques, differing in explanation format and targeted category of end-users, from non-experts to researchers with knowledge in particle physics. In particular, we make the following contributions to the use-case under consideration:

1. Attribution algorithms [31, 32, 34, 37] provide explanations that are based on saliency maps, or heatmaps, graphically showing the most discriminative regions in an image that have influenced the output decision of the model, based on gradient information. Simple saliency maps methods tend to be noisy and may contain artifacts [36], which has motivated the design of a wide family of attribution algorithms such as Regression Activation Maps (RAM), Integrated Gradients (IntGrad), and SmoothGrad. However, these methods require background knowledge on how they are implemented and their assumptions may not align to a specific use-case. As an example, SmoothGrad [31] averages multiple saliency maps computed on noisy versions of the original input image. This noise is typically chosen to be Gaussian, which is not adequate to our use-case, since adding Gaussian noise to binary images results in images lying outside the manifold of the input data. In this paper, we provide two customized variants of SmoothGrad [31] and Integrated Gradients (IntGrad, [32]), taking into consideration the characteristics of the input data, which are directly applicable to our use-case and provide a more interpretable (i.e., sparse) explanation.
2. Intrinsically interpretable Decision Trees and variants [3, 7, 15, 17, 24, 35, 36] are models that do not require external explanatory methods since they have transparent structures that immediately make users understand their decision flow and are, therefore, said to be self-explained. From the simple logic rule-based Decision Tree, several variants have been proposed, combining an interpretable tree-shaped architecture with neural network components, such as neurons and activation functions, and introducing a 'soft' mechanism, in Soft Decision Trees (SDTs), to weigh 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. In this work, we have adopted one of the latest modifications enriched with convolutional layers, resulting in the so-called Convolutional Soft Decision Tree (ConvSDT), and implemented a visualization procedure to immersively explore the decision-making process of this explainable structure.
3. Example-based xAI techniques, also known as data attribution methods, provide explanations by collecting the most influential examples from the training set that support the output prediction for a test sample. Based on how influence is defined, different data attribution methods can be obtained. In this paper, we focus on analyzing the results of Tracing Gradient Descent (TracIn) [29], which computes an approximate influence function by analyzing the gradient correlation across different samples. This allows to identify what are called *proponents*, i.e., items that have reduced the loss at training time and are positively correlated with the sample to explain, and *opponents*, namely elements that instead, increasing the loss, have caused errors during the training phase and are negatively correlated. Observing Proponents and Opponents, one should be able to understand the logical connection with the input and whether the output prediction has been generated coherently. We have applied

TracIn in our domain and tested its potentialities on our complex-formatted data, way different from the ‘toy’ datasets used in the original work.

Our results will testify how such xAI methods, if carefully and appropriately customized, can be safely adopted even in crucial scenarios and the associated explanations can be addressed to different categories of end-users, both experts and not. In addition, while most works in the past have focused on applying one selected xAI approach to each problem, in this paper we focus on what we call a *convergent* application of xAI methods, belonging to different families, highlighting how in a difficult scientific scenario the heterogeneity of the algorithms can be used to provide qualitatively different insights on the problem under consideration. We believe this kind of varied analysis, while still underexplored in high-energy physics (with a few exceptions, e.g., [2, 10, 11, 13, 19, 25]), can provide valuable insights also into the application of xAI techniques to other scientific fields, ranging from medicine to mathematics.

Materials and Methods

Datasets

The ATLAS Level-0 RPC trigger system for HL-LHC aims to collect all the particle hit information from the fast RPC detectors in a given sector (i.e., a solid angle region of the detector) and tries to find a muon candidate—a collection of hits identified as a track in the detector—and measure its properties. The interesting quantities are the muon track spatial parameter inside the experiment (typically represented in terms of pseudorapidity η), and the transverse momentum of the muon p_{T} (expressed in giga-electron-volt, GeV).

When a muonic particle passes through the ATLAS RPC detector its trajectory can be conveniently represented into a fixed image-like 2D mesh, simulating a bi-dimensional grid. The bi-dimensional grid can then be used as input for ML models, specifically convolutional architectures, particularly suitable to find patterns like the muon tracks in this test scenario. The generated sample does not contain only the effective trajectory of the muon but also a varying amount of noise randomly due to electronic noise and environmental radioactive artifacts caused by beam-induced radioactivity from the materials surrounding the detector.

More in detail, the images are binary gray-scaled pictures with mainly 0-valued pixels and only a few lit 1-valued pixels, meaning that the associated sensors of the detector have registered the passage of the muon or have been affected by noise. The noise is generated by emulating the expected particle rate conditions in the future phases of the experiment during the HL-LHC. The toy simulation only accounts for the average hit rate in the spectrometer RPCs, therefore it does not consider correlated backgrounds. We did not aim to perfectly reproduce the experimental conditions but to give a proof of principle of the explainability methods in the context of a high-energy physics experiment. The pattern of the muon depends on the angle with which particles enter into the detector, expressed through the pseudorapidity η, and the p_{T} of the muon. These two parameters compose the labels to be estimated.

Each image is modeled as a (9×384) bi-dimensional array, where the 384 horizontal pixels map the values of η to the x-axis of the sample, while the 9 vertical pixels correspond to detector layers (from the bottom: 3 detector layers for the inner trigger station, 4 for the middle, and 2 for the outer station) [14]. In this convenient representation, an infinite momentum muon appears in the image as a vertical pattern of pixels, independently of the pseudorapidity η, while lower momentum muons appear ideally as tilted pixel patterns with slopes inversely proportional to the muon p_{T}. An example of a dataset item is visualized in Fig. 1, where we can distinguish the component with the actual pattern of the muon and the additional radioactive noise attached therein. Our ML models were trained using only the noisy images while the denoised ones were employed for support at the explanation phase. The derived dataset is composed of 945k images divided in a 90-10 train–test split.

Another interesting and additional dataset that has been involved in this work is composed of images of *only noise* and no muonic traces. These samples have been used post-training, during the explanation phase, to test the model and observe its behavior with items whose characteristics, zeroed p_{T} and η, were absent at training time. This freely accessible dataset, similarly to the previous one, is composed of 945k (9×384)-shaped images directly derived from the original pictures of the first noisy dataset; obviously, the associated denoised version is by definition an empty image.

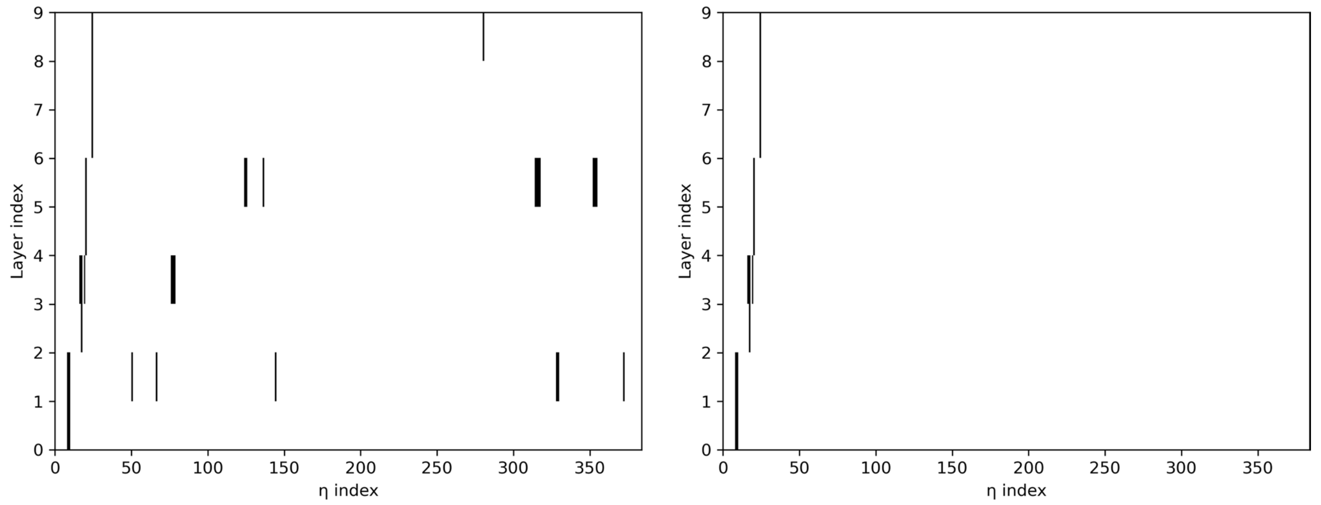
Metrics

The regression problem of estimating muonic parameters has been evaluated by minimizing the average prediction error both on p_{T} and η in terms of *mean absolute error* (MAE):

$$MAE(p_{\mathrm{T}}, \eta, \hat{p}_{\mathrm{T}}, \hat{\eta}) = |p_{\mathrm{T}} - \hat{p}_{\mathrm{T}}| + |\eta - \hat{\eta}|, \tag{1}$$

where $\hat{p}_{\mathrm{T}}$ and $\hat{\eta}$ are the predictions of the neural network.

In addition, specific physics-related metrics have been introduced to evaluate the performances of the networks. To this aim, two measures to quantify the effectiveness of our trained models have been adopted and, we must notice, both of them act on p_{T} only. The definitions of these metrics are:



(a) Image with the muon pattern and additional noise.

(b) Denoised image with the actual muon pattern only.

Fig. 1 Sample image

- **Spread:** The averaged value of the difference between real and predicted momenta in the range [7, 13] GeV (around the nominal trigger threshold of 10 GeV); this quantity should be minimized. Assuming that y_{p_T} and $\hat{y}_{p_\mathrm{T}}$ are, respectively, the vectors containing ground-truth and estimated p_T, it is possible to define $S = \{s \mid y_{p_\mathrm{T},s} \in [7, 13]\ \mathrm{GeV}\}$ as the set of vector indices corresponding to real momenta in the range of interest; the spread is then mathematically expressed as follows:

$$\mathrm{spread}(y_{p_\mathrm{T}}, \hat{y}_{p_\mathrm{T}}) = \frac{\sum_{s\in S} |y_{p_\mathrm{T},s} - \hat{y}_{p_\mathrm{T},s}|}{\#(S)}, \tag{2}$$

  where $\#(\cdot)$ represents the cardinality of a generic set of elements, namely the number of items composing the set.
- **Plateau efficiency** (from now on simply efficiency): The number of muons concurrently having $p_\mathrm{T} > 15$ GeV and $\hat{p}_\mathrm{T} > 15$ GeV over the total number of muons with $p_\mathrm{T} > 15$ GeV. This normalized value should be, instead, maximized. It can be straightforwardly derived through the definition of two sets of vector indices, $E_{num} = \{e \mid y_{p_\mathrm{T},e} > 15\ \mathrm{GeV} \wedge \hat{y}_{p_\mathrm{T},e} > 15\ \mathrm{GeV}\}$ and $E_{den} = \{e \mid y_{p_\mathrm{T},e} > 15\ \mathrm{GeV}\}$ as the ratio between the cardinality of these two collections:

$$\mathrm{efficiency}(E_{num}, E_{den}) = \frac{\#(E_{num})}{\#(E_{den})}. \tag{3}$$

As intrinsically noticeable from the definition of the efficiency metric, the ML algorithm should be then able to select muons with momentum above a fixed and high threshold ($p_\mathrm{T} \geq 15$ GeV) and simultaneously reject particles below the same threshold ($p_\mathrm{T} < 15$ GeV). This separation implies the consideration of four different regions for a predicted event: the cases when a muon is correctly selected or rejected, defined from now on as True Positives (TP) for selection and True Negatives (TN) for rejection, and the complementary cases when the model 'misclassifies' a pattern locating it in the False Positives (FP), when it erroneously outputs a momentum above the threshold, or in the False Negatives (FN), when it wrongly rejects the muonic trace. The notion of True/False Positives/Negatives, explicitly defining a 4-sector confusion matrix, will be of great interest during the explanation phase because the study will focus on representative events belonging to each of these four regions.

Models and Experimental Settings

Our model architectures include two Convolutional Neural Networks (CNNs), one named *Attribution CNN*, used with the three attribution-based saliency maps methods, and the other called *TracIn CNN* associated with the homonym xAI technique, and a *Convolutional Soft Decision Tree* [3]. The CNNs are fairly simple: after an *Input* layer trivially having as size the images' dimensions, they are made up of a succession of four convolutional layers, respectively, provided with 16, 32, 64 and 128 output filters. Every layer uses a kernel of size (5×5), a *same* padding is employed and a *ReLU* activation function is applied after each convolution. The output of this convolutional body is passed to a Global Average Pooling (GAP) layer and, finally, the regression prediction is performed by the head of the model: the Attribution CNN's head is made up of three *Dense* layers having, respectively, 1024, 512 and 2 neuronal units. The penultimate and the third last layers use *ReLU*-s activation functions while the final one uses a *Linear* activation function.

Table 1 Structure and number of parameters of the adopted CNN

Layer	Output shape	# params
Input layer	(9, 384, 1)	0
Conv2D	(9, 384, 16)	416
Conv2D	(9, 384, 32)	12832
Conv2D	(9, 384, 64)	51264
Conv2D	(9, 384, 128)	204928
GAP	(128)	0
Dense	(1024)	132096
Dense	(512)	524800
Dense	(2)	1026
Trainable params	Non-trainable params	Total params
927,362	0	927,362

The '*Output shape*' column refers to a single-image input

The structure of this model and its related trainable parameters are summarized in Table 1. The TracIn CNN's head only consists of a final *Dense* layer with 2 units; this design choice is motivated by the fact that for efficiency we use a fast TracIn implementation that requires a one-layered fully connected head to simplify the gradient computations [29].

The other model typology refers to the intrinsically interpretable ConvSDT, namely a Soft Decision Tree hierarchical structure positioned right after a sequence of convolution operations, whose implementation design has been freely inspired by the work of [3, 15, 24]. In particular, we used three convolutional layers with, respectively, 64, 32, and 16 filters with a kernel size of (3×3). Between each pair of consecutive layers, we introduced a *Batch Normalization-ReLU-Max Pooling* procedure to gradually reduce the dimensionality of the input image. This is motivated by the fact that (Soft) Decision Trees have often reported a decrease in performances with high-dimensional data [36]; with convolutions, we have exploited its advantages with images and have simultaneously facilitated the training of the tree thanks to such dimensionality reduction process: from a $9 \times 384 = 3456$-dim input vector without convolution, we are able to decrease and simplify it to a 768-dim vector. In the second phase, instead, the convoluted image, now a flattened 768-dim input vector, enters into the SDT. Its hierarchical structure is a priori built by defining the fixed depth d of the tree, namely this also implies the presence of $(2^d - 1)$ inner nodes and of 2^d leaf nodes. By imposing $d = 5$, the decision tree is then modeled with two *Linear* layers with, respectively, $(768 \times (2^5 - 1)) = (768 \times 31)$ and $(2^d \times 2) = (32 \times 2)$ input–output features. The 'inner' layer is followed by a *Sigmoid* activation function to simulate the binary splitting flow of the tree, while the 'leaf' layer is initialized with a normal distribution, as experimented in [24]. The output of the first layer is manipulated in order to compute the path probabilities to reach every leaf: these probabilities are then used to "weight" the predictions of the leaves, namely the output of the second and last layer [15, 17]. It is important to mention that a helpful pre-processing step scales the original labels y to $y' = \frac{y-\mu}{\sigma}$, where μ and σ are, respectively, the mean and the standard deviation of y in the training set [24]. Label scaling aims at reducing the interval of predictable values and at making the model able to predict more accurately. The resulting architecture of this model and its related parameters are summarized in Table 2. It is important to mention that the complexity of the model is incredibly low if compared to the CNNs described above: just observing the different number of parameters in the two tables, the dimensionality of the ConvSDT is about 20 times smaller than the one of the CNN for attribution methods, implying also faster predictions and lower memory required for storage.

In addition to training this model alone, we have also experimented a re-training from scratch exploiting a simple technique of *Knowledge Distillation*, a branch of ML that aims at transferring knowledge between different architectures [16]. The procedure, already addressed in [15] and re-proposed in this work, is straightforward: instead of training the soft tree as usual, namely by using the actual labels as ground-truth values, the hierarchical model has been trained replacing labels with the predictions generated from a teacher model, here the previously trained Attribution CNN. The resulting model will be denoted as *Distilled ConvSDT* in the rest of the discussion to differentiate it from the 'plain' version of the decision tree.

During the training procedure, the Adam optimizer [21] with a *learning rate* of 0.001 has been used for the Attribution CNN and the ConvSDT, while the TracIn CNN was trained with a SGD [5] optimizer, with a *learning rate* of

Table 2 Structure and number of parameters of the adopted ConvSDT

Layer	Output shape	# params
Conv2D	(64, 9, 384)	640
BatchNorm2D	(64, 9, 384)	128
ReLU	(64, 9, 384)	0
MaxPool2D	(64, 4, 192)	0
Conv2D	(32, 4, 192)	18,464
BatchNorm2D	(32, 4, 192)	64
ReLU	(32, 4, 192)	0
MaxPool2D	(32, 2, 96)	0
Conv2D	(16, 2, 96)	4624
BatchNorm2D	(16, 2, 96)	32
ReLU	(16, 2, 96)	0
MaxPool2D	(16, 1, 48)	0
Flatten	(768)	0
Linear	(31)	23,839
Sigmoid	(31)	0
Linear	(2)	64
Trainable params	Non-trainable params	Total params
47,855	0	47,855

The '*Output shape*' column refers to a single-image input

0.01, because the associated xAI algorithm is more correctly approximated with this specific type of optimizer [29].[1] The CNN models were trained for 30 epochs dividing the dataset into 32-sized batches, instead the ConvSDT was trained for 5 epochs using a batch size of 64.

Compared Explainability Techniques

In this section, we describe the explainability techniques we applied on the methods described in "Materials and Methods" section, and any customization we made when necessary. For each class of xAI methods, the selection of a particular method was made based on a combination of factors that include their popularity, stability of the underlying software libraries, and simplicity. We expect that our results will extend to other methods from the same class, and we refer to published surveys, e.g., [1], for a broader exposition on xAI methods.

Due to the fact that images composing our datasets are extremely different from standard RGB or gray-scaled images, some of the proposed xAI approaches have required specific adaptations to suitably deal with the proposed HEP use-case. The changes have not involved all five approaches: some of them have required particular care, while others were already applicable to muonic pattern images.

Attribution Methods

Regression Activation Map [34] is a common saliency map method for regression outputs, and it was found to be already compatible with the HEP domain. We implemented RAM to produce two heatmaps for each sample image, one for the transverse momentum and the other for the pseudorapidity variable, highlighting the discriminative regions for both labels separately.

On the other hand, IntGrad [32] and SmoothGrad [31] had to be customized. IntGrad is a state-of-the-art saliency map method that computes a smoothed map by interpolating between an empty image and the original input image. For our use-case, we re-define the original interpolation routine to produce a sequence of images such that, starting from an empty baseline with all zeroed pixels and ending with the original image, the in-betweens are realized by iteratively adding one lit 1-valued pixel for each new interpolation. For instance, given the noisy image depicted above (Fig. 1) and composed of $n = 33$ lit pixels, the resulting interpolation will instantiate a set of $n + 1 = 34$ images, including the empty baseline. The pixels are added by rows in a top-down fashion. A sampling of this sequence is extracted in Fig. 2 to show its evolution over the interpolation steps.

[1] The TracIn approximation sustaining the whole method empirically seems to work also with optimizers not based on gradient descent (e.g., Adam), however we preferred to introduce SGD to maintain the optimal setup.

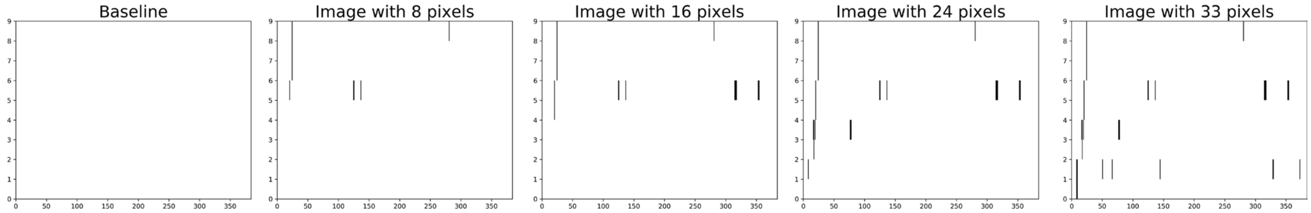


Fig. 2 Example of customized interpolation given an image containing 33 lit pixels: the result is a sequence of 34 images starting from the empty baseline. Here, a sampling of the sequence at steps 8, 16, and 24

As mentioned in "Introduction" section, SmoothGrad's adaption is related to the way noise is randomly added to the picture. Constrained by the fact that pixels have binary values only, we have replaced the Gaussian with a *salt-and-pepper* noise distribution to add 1s according to a predefined noise percentage σ. The two hyperparameters used by the method, the number of smoothing iterations n and σ, have been chosen through a Grid Search over values for $n \in [10, 20, 50, 80, 100]$ and for $\sigma \in [0.5\%, 1\%, 2\%, 5\%, 10\%]$. Evidence showed that setting $n = 100$ and $\sigma = 2\%$ represented, for our use-case, the best trade-off between level of noise, computational cost, and quality of the explanatory results. Finally, SmoothGrad has exploited IntGrad itself as saliency method integrated into its functioning. Future work will consider the inclusion of additional important attribution methods, commonly used also in HEP use-cases, such as Layer Relevance Propagation [4, 6, 8, 26, 27, 30] or Shapley values [23].

Convolutional Soft Decision Trees

The architectural aspect of ConvSDTs has been already described in the previous section about model structures and parameters; however, it is important to mention that this work also provides a graphic visualization of their decision flow by re-implementing the typical hierarchical drawing of Decision Trees inspired by the Scikit-learn library [28] and adapting it to sketch the decisions of each node from the root to the leaves with their associated probabilities, similarly as done in [15] but with augmented visual information and insights relative to the model's functioning: explanations will support and motivate the whole process of soft trees' probabilistic decision-making that, starting from inner nodes visualized as filters to highlight the contributing pixels at each depth layer, will also reach and show the probability distribution of leaves to understand the level of confidence of the associated prediction together with potential errors taken along the path from the root. Representative examples will be later discussed in the results section.

Training Data Influence Estimation

The last xAI approach adopted in this work is a *data influence* method [29], where the term 'influence' represents the impact that training samples have on the prediction of a test sample (as opposed to single elements of the test sample itself). The most influential Proponents and Opponents identified by the TracIn algorithm [29] will act as examples motivating the reasons why the model has produced that specific prediction given a fixed test item to explain.

In the TracIn formulation, influential data points are selected by studying the effect they have on the loss function during the entire optimization process [29]. Ideally, the method should trace all model parameters at all training points and for each iteration of the train procedure but, due to the clear computational impracticality of tracking all these quantities, an easily integrable approximation is based on the possibility of simulating the training routine in the algorithm by saving only a subset of k checkpoints, current configurations of parameters, while training and then loading them to calculate gradient influence off-line. More details about the procedure can be found in [29].

A few adaptations regarding the algorithm have focused on the associated CNN, by using the more appropriate SGD optimizer at training time, as explained before, and the choice of which checkpoints to save and use in the approximated version of the approach. Authors of [29] usually recommend selecting particularly important checkpoints in terms of loss improvements; therefore, the picked neural networks have been chosen among the stored ones according to this criterion and also to their occurrence with respect to the whole training process, trying to select them uniformly in the interval of the train epochs.

Results and Discussion

Regression Stage

After the training procedure previously described, the performances of the four models are collected in Table 3,

Table 3 Evaluation metrics: MAE loss, spread, and efficiency for all the adopted models

Model	Loss	Spread	Efficiency	%
Attribution CNN	0.61	1.12	0.88	–
ConvSDT	**0.57**	**1.05**	0.85	–
Distilled ConvSDT	0.65	1.27	**0.89**	+4
TracIn CNN	0.73	1.31	0.84	–

Bold entries in the table denote the best performance among the four models, such as the lowest values for Loss and Spread in their respective columns, and the highest value for Efficiency in its corresponding column

The '%' column indicates the percentage improvement in efficiency using Knowledge Distillation

showing results for the regression task in terms of loss, spread and efficiency. Starting from the first metric, losses are comparable after an inverse-scaling on ConvSDT's output predictions. Attribution CNN has registered a lower error than the similar TracIn CNN, while Knowledge Distillation apparently seems to have no benefits if monitoring this quantity only since the undistilled tree has reached a minimized value with respect to the distilled version of the model and, in general, the minimum loss among all the tested models. Moving to the physics-related metrics, the spread has registered an almost-optimized[2] value in the plain ConvSDT, but also the other models have registered satisfactory achievements concerning this evaluative measure. The most interesting considerations at this stage of results can be derived by observing the efficiency: the 88% of efficiency certifies Attribution CNN as one of the most performing considering this criterion, considerably higher than the 85% of ConvSDT and 84% of TracIn; however, when we used this network as a teacher for the convolutional tree student, Distilled ConvSDT has reached and even surpassed Attribution CNN, reporting a consistent increase of +4%, resulting in a final 89%-efficiency.

An additional and interesting key feature of ConvSDT is given by its reduction in terms both of the number of parameters and memory storage, an advantage that makes such a model preferable for real-time applications. Exploiting Knowledge Distillation and remembering that ConvSDT requires a total number of parameters to be tuned almost 20 times smaller than the CNN, the performances of the two architectures remain comparable while the former model is also minimizing the required memory storage: a saved checkpoint of Attribution CNN takes 10.6 MB of memory space, more than 55 times bigger than the 196 KB needed for the ConvSDT checkpoint.

Finally, models' performances can be evaluated through the confusion matrix indicating the rate of muonic event selection/rejection. The resulting three matrices, one for each model except ConvSDT, discarded for its poorer efficiency with respect to Distilled ConvSDT, are represented in Fig. 3. In general, matrices reflect previous achievements but it is also interesting to notice that the convolutional networks, Attribution CNN and TracIn CNN, are able to minimize the number of False Positives at the cost of increasing False Negatives; vice versa, Distilled ConvSDT has reduced the False Negative events while causing a slightly opposite effect on False Positives: one can choose to use a model rather than others if the objective is to minimize one specific region of errors among the two.

We also tested our models with the supplementary dataset composed exclusively of images containing only noises and no muonic traces. Out of 945k images, models have selected the following amount of events:

- Attribution CNN, 1389 images, corresponding to 0.15% of the noised dataset;
- Distilled ConvSDT, 2 images, corresponding to less than 0.01% of the noised dataset;
- TracIn CNN, 11 images, corresponding to less than 0.01% of the noised dataset.

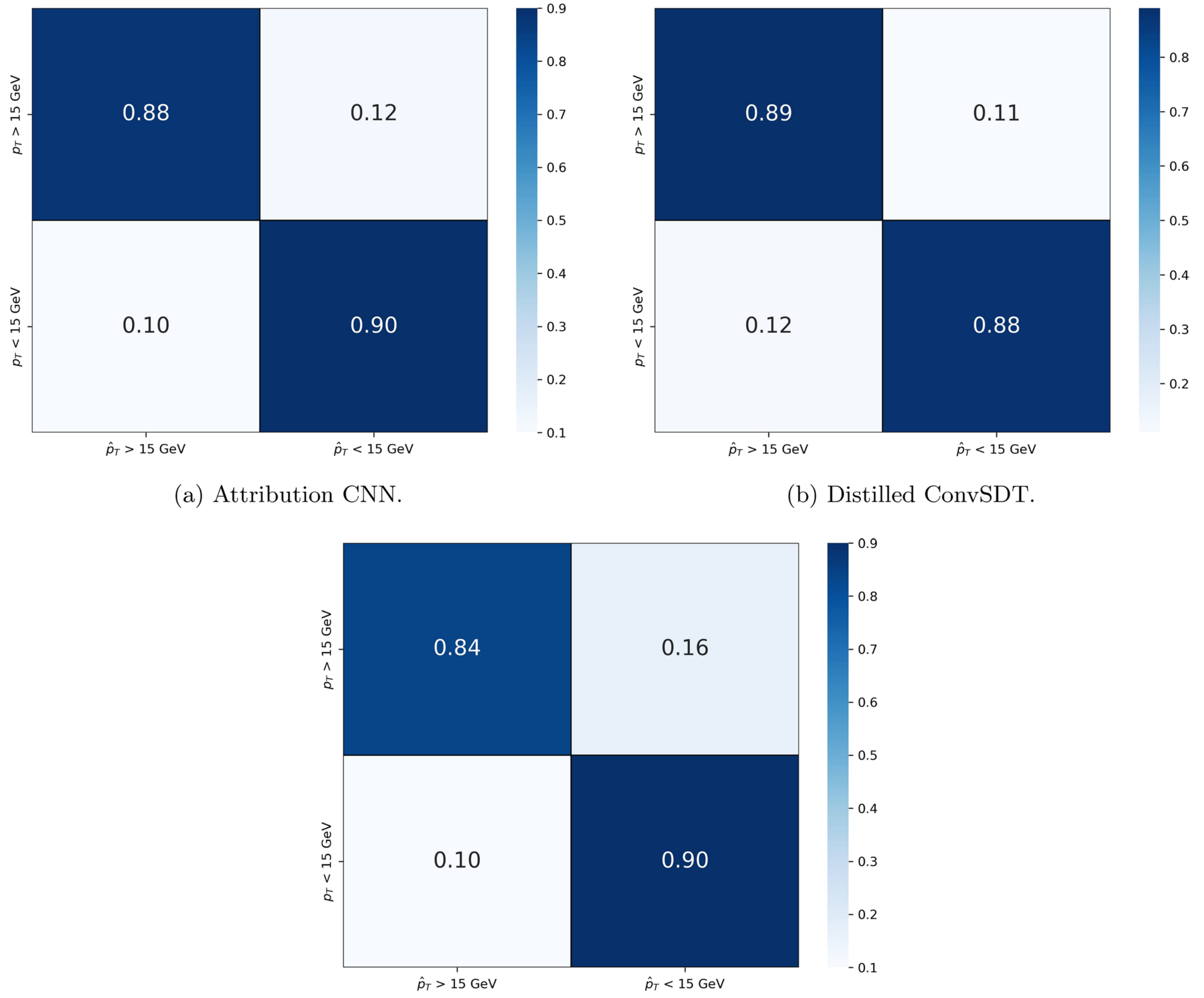
The requirement of the project for the real-time muonic selection system of the ATLAS experiment demands the fake rate, namely the fraction of erroneously selected only-noise events, to be $< 0.2\%$[3]; this sub-goal is successfully accomplished here, as testified by the above percentage values. Such a result can be considered in an even more positive way reminding that the training dataset did not contain any only-noise image, meaning that the three models did not expect to have as input 0-valued p_T and η.

Explainability Stage

After assessing the performances of our models from a quantitative point of view, the analysis of results focuses on explainability, to provide justifications for models' decisions and, consequently, ensure the validity of their predictions. Due to the fact usually users do not share the same knowledge about Physics, HEP and ML, our array of techniques is intentionally built to guarantee explainability irrespective of the experience end-users have in the

[2] The spread value expressed in GeV has an intrinsic minimum value of about $\approx 1-2$ GeV due to the intrinsic resolution through which the detector measures the particle momentum.

[3] This precise threshold is motivated by the maximum rate of supportable events at this stage of selection required by the L0 muon trigger of the ATLAS experiment at HL-LHC.



(a) Attribution CNN.

(b) Distilled ConvSDT.

(c) TracIn CNN.

Fig. 3 Confusion matrices

topic. Therefore, the three attribution methods are thought for beginners or non-experts thanks to the visual intuitiveness of heatmaps; the self-interpretable Convolutional Soft Trees are again easily understandable, however, some insights or additional details may be better perceived with a prior comprehension of particle kinematics and muon characteristics hence the method has a medium-difficulty level of comprehensibility; TracIn is instead designed mainly for physicists because the relationship between the input sample and its Proponents and Opponents is not always immediately perceived and further studies are often required. Coherently with this increasing-difficulty direction, we will explore the results of our set of methods, showing how the convergence of xAI is able to support users investigating models' decisions, detect erroneous predictions by consulting multiple explanations at once and understand diversified aspects of the faced critical domain.

xAI Use-Case 1: All Methods Agree

Starting from a first case where our methods support each other, resulting in an enhanced reliability of models' outputs, examples of explanations are grouped in Fig. 4, Fig. 5 and Fig. 6, which we discuss in-depth in the following and in respective captions. Initially focusing on Attribution CNN, the explainability study we performed concerned the investigation of the most relevant discriminative regions provided by our set of saliency maps methods.

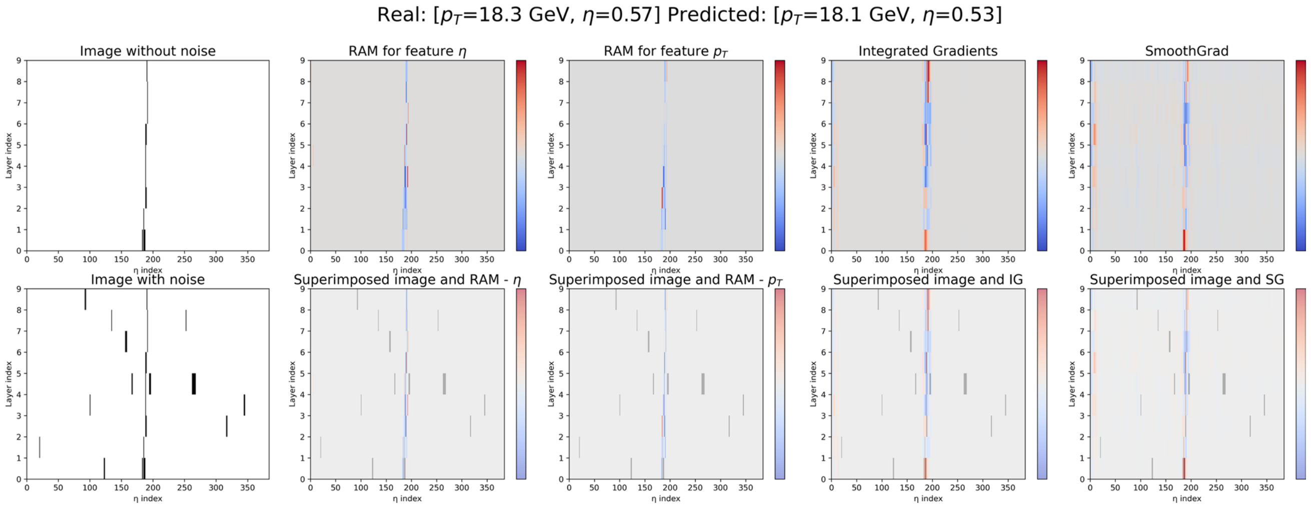


Fig. 4 Explanation through attribution methods: five columns showing, from left to right, the image passed as input to the model labeled as ‘Image with noise’ and its denoised version as ‘Image without noise’; the heatmap generated by RAM for the feature η and its superposition with the input image; the heatmap generated by RAM for the feature p_T and its superposition with the input image; the heatmap generated by IntGrad (IG) and its superposition with the input image; the heatmap generated by SmoothGrad (SG) and its superposition with the input image. In the title, it is possible to compare the associated ground-truths (‘Real’) and the predictions of the model (‘Predicted’). In this example, a TP straight-lined muonic pattern corresponding to a muon with true p_T of 18.3 GeV and true η of 0.57, by comparing the denoised image and the heatmaps in the first row we can assess that the model is able, for each of the three attribution methods, to select the right muonic trajectory. Assuming that the denoised image and the ground-truths are unavailable, users observing the superimpositions in the second row can reasonably state that the CNN has probably individuated the real particle trajectory and ignored other noisy random hits in the image, resulting in a reliable prediction

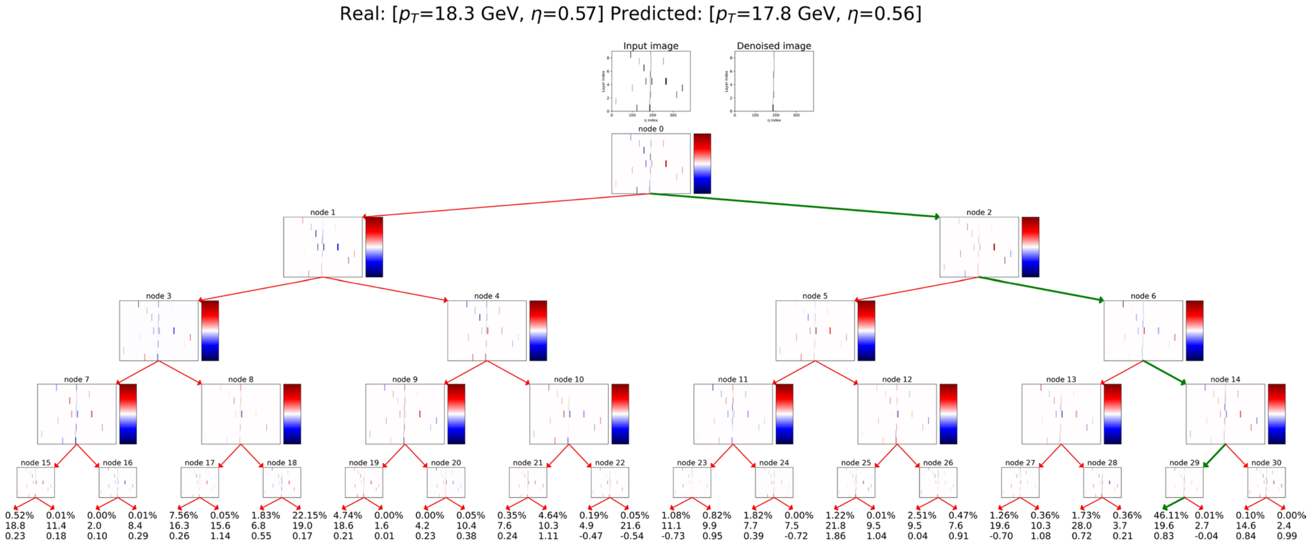


Fig. 5 Explanation through ConvSDT: the visualization shows the flow of the input image (on top, next to its denoised version) into the tree. Each inner node is represented as a correlation map with the input; a leaf node, instead, is characterized by three textual components: the probability (in percentage) of reaching it and the corresponding predictions of p_T and η to be weighted accordingly. There exists a *maximum probability path* in green, leading to the leaf with the highest probability, and minor red paths with lower probability. In the title, it is possible to compare the associated ground-truths (‘Real’) and the predictions of the model (‘Predicted’). In this example, a TP straight-lined muonic pattern, ConvSDT outputs a high-valued momentum mainly due to the 46.11%-maximum probability path leading to a p_T of 19.6 GeV, meaning that the prediction is made with high confidence. Moreover, the maps of the inner nodes along the maximum probability path, showing the pattern of hits more correlated (red color) or anticorrelated (blue color) with the splitting probability, suggest that the pixels of the muonic pattern are always in correlation with the nodes, meaning that the tree has considered consistent hits while predicting

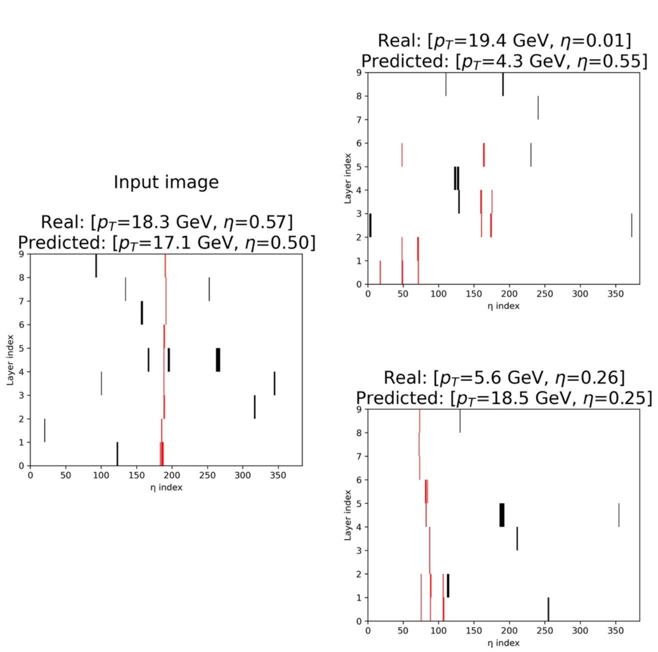


Fig. 6 Explanation through TracIn method: on the left-hand side, the image passed to the model labeled as 'Input image'; on the right-hand side, two distinct groups of images containing the 3 most influential Proponents, on the top, and Opponents, on the bottom. Each image comprehends in the same plot both the noisy hits in black and the actual pattern of the muon in red and is also equipped with a title containing the associated ground-truths ('Real') and the predictions of the model ('Predicted'). In this example, a TP straight-lined muonic pattern, it is possible to notice that all the individuated Proponents are characterized by very high real momenta and show analogies in their actual muonic trajectories. Opponents instead, are events with low real p_{T}, and predicted $\hat{p}_{\mathrm{T}}$ similar to the real p_{T} of the input image, that show muonic patterns very close to the one in the input image. This testify how the model has been particularly influenced by samples similar to the test input during the training and, consequently, implying the prediction has been performed coherently

Each attribution explanation is organized in five columns and two rows (Fig. 4) that, from the left, show:

- the input image without noise (clearly not given to the model) in the upper row, and the same image but with noise passed as input to the model, in the lower row;
- the feature heatmap outputted using the RAM method on the label η (upper row) and its superimposition with the input image with noise (lower row);
- the feature heatmap outputted using the RAM method on the label p_{T} (upper row) and its superimposition with the input image with noise (lower row);
- the feature heatmap outputted using the IntGrad method (upper row) and its superimposition with the input image with noise (lower row);
- the feature heatmap outputted using the SmoothGrad method (upper row) and its superimposition with the input image with noise (lower row).

The purpose of superimposed images is to graphically visualize whether noisy pixels have impacted on model's predictions by observing if heatmap and noise coincide in some points of the sample. Moreover, color bars have been placed next to heatmaps to show the level of 'heat' for each pixel: positively and negatively valued heats indicate areas that have affected model's decisions and belong, respectively, to the red and blue color intervals; background is gray-colored and corresponds to the 0 value indicating no influence.

In this example, having access to the denoised version of the input, we can assess the image is characterized by an almost straight-lined pattern with true $p_{\mathrm{T}} = 18.3$ GeV and predicted $\hat{p}_{\mathrm{T}} = 18.1$ GeV, which testifies this is a correctly selected event and, according to our nomenclature, a True Positive. Noise components are randomly present in both left and right sides of the image, next to the real trajectory of the muon. Focusing on the first row of Fig. 4, we can see that the model is consistently able, for each of the three attribution methods, to select the right muonic pattern. Assuming that the denoised image is missing and no clues on ground-truth labels or on the actual muonic pattern are available, if we observe the superimpositions in the second row, we can state with a reasonable level of confidence that the CNN has probably individuated the real particle trajectory, ignored other random hits in the image due to noise and that the associated high-momentum prediction is coherent with the perturbations in the heatmaps depicting an almost vertical path of the muon.

Our work has also shown how different attribution methods can be used together: here the three techniques have generated similar heatmaps without contradictions, making the whole validity statement more reliable. Therefore, even if many other works complained about disagreements between techniques like in [22], in our case all these methods agree and choosing only one among them could be enough for the evaluation through attribution.

Moving to explanations derivable from soft trees, the interpretability provided by the hierarchical structure of the ConvSDT allows the user to understand the model's decision flow. We have implemented a visualization modality in order to graphically render the explanation sustaining a prediction (Fig. 5). Our visualization function portrays the phases of the process applied to the input image during its passage through the depth levels of the tree. Each inner node is represented as a 2D grid showing the correlation between the kernel of the multi-layer perceptron and the input and is equipped with a color bar to qualify the positive and negative correlation influence of the node's pixels/weights. The leaf nodes, instead, are characterized by the probability of reaching them together with the associated predictions of p_{T} and η to be weighted accordingly. Moreover, there will exist a *maximum probability path*, marked in green, leading to the leaf with the highest and heaviest contribution and other $2^d - 1 = 2^5 - 1 = 31$ red paths with lower probabilities whose leaves' predictions will act as a refinement to the final prediction. In this way, the possibility to explain is given in a double way: leaves quantify the probability as confidence in predicting certain values rather than others using it to weigh their associated outputs, while inner nodes certify that such confidence is based on coherent pixels or areas of the correspondent correlation images.

In the previous analysis with heatmaps, attribution has suggested that the prediction was correct by associating the high value of $\hat{p}_{\mathrm{T}}$ with the highlighted straight pattern in the maps. Exploiting convergence of methods to the same input sample, these considerations are supported by the ConvSDT's output as well: the final prediction is again a high-valued transverse momentum of 17.8 GeV with the maximum probability path leading to a leaf predicting $p_{\mathrm{T}max} = 19.6$ GeV with 46.11% of confidence, which can be seen by following the green path in Fig. 5. Such percentage can be also "inflated" by considering the second and third most likely paths, respectively 22.15% reached from node 18 and 7.56% from node 17, pointing to predictions close to $p_{\mathrm{T}max}$. In the end, the hypothesis from attribution methods is here strengthened and confirmed by the high level of confidence that the tree has when directing the search toward values close to the predictions. Moreover, the kernels of the inner nodes along the maximum probability path (and along the second and third paths) suggest that the vertical line of the input image, highlighted in the heatmaps and individuated as the possible pattern of the muon, is always subject to correlation with the weights of the nodes, ensuring even more the correctness of the conclusions drawn so far. Physicists and HEP experts may also understand the role played by the noisy hits spread in the input and visibly correlated with the inner nodes of the ConvSDT.

The last part of the study on convergent interpretability of muon patterns has concerned TracIn and was performed using three checkpoints uniformly spread among the 30 epochs of training of the model, specifically picking the saved models at epochs 5, 15 and 25. The gradients were traced using 32-dimensional batches, iterating both on the training and the test subsets.

We have restricted our research to the 3 most affecting Proponents and Opponents that we empirically found to make the interpretation sufficiently exhaustive; each explanation is composed of these two sets together with the image to be explained. The visualization comprehends in the same plot both the noisy hits in black and the actual pattern of the muon in red. Each picture is also equipped with a title containing the real and predicted labels (Fig. 6). Evidence and consultation with HEP experts not involved in this study, have suggested that TracIn is able to individuate Proponents coherently: their level of true p_{T} and true η is consistent with the one of the predicted image and the actual pattern shows similarities with the explaining picture. Opponents, also as expected, show pattern similarities with the tested image, but very different true values of the p_{T}, consistently with the definition of an opponent event that has caused errors during the training phase, increasing the loss, and so is negatively correlated with the test image.

Exploiting the advantages of convergence, we can assert that the considerations on the quality of the prediction for this input sample are correct. In this scenario, the last explanation through examples of TracIn definitely confirms the conclusions of the previous xAI algorithms: also TracIn CNN generates a high-momentum prediction of $\hat{p}_{\mathrm{T}} = 17.1$ GeV.

We note that even if the insights derivable from this technique are only partial due to the fact that, in this application, the method is mainly designed for experts, it is still possible to notice that the individuated Proponents are characterized by very high real momenta with $p_{\mathrm{T}} > 18$ GeV in all the train examples, testifying how the model has been particularly influenced by samples similar to the test input in terms of label values during the training and, consequently, implying the prediction has been performed coherently.

Concluding this first case of evaluation, each xAI method has led to a conclusion similar to the ones of the other algorithms, strengthening the validity of the whole set of explanations: convergent xAI approaches have guaranteed a more

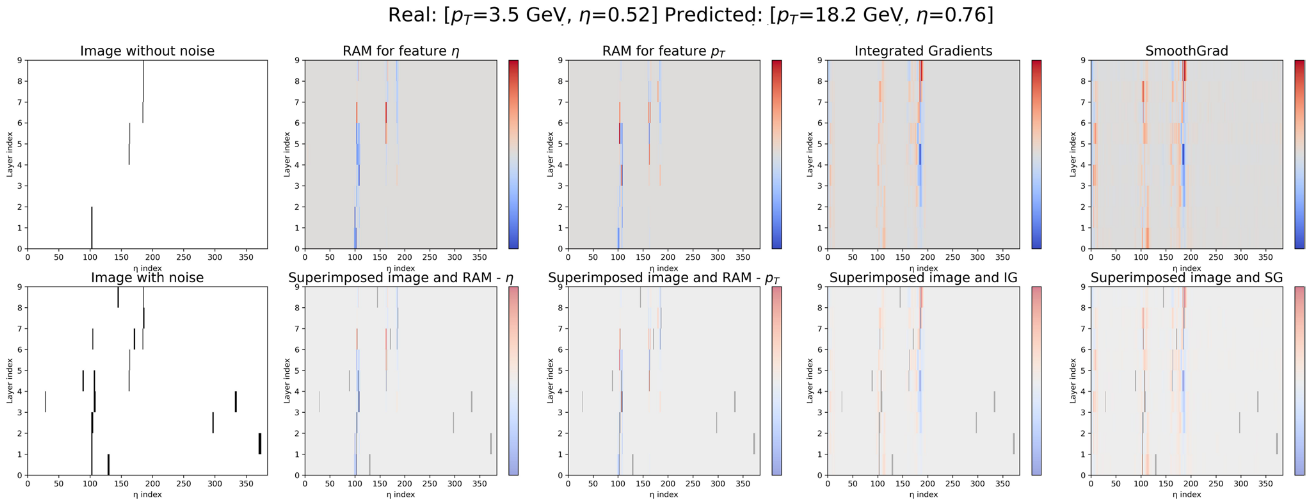


Fig. 7 Explanation through attribution methods: five columns showing, from left to right, the image passed as input to the model labeled as 'Image with noise' and its denoised version as 'Image without noise'; the heatmap generated by RAM for the feature η and its superposition with the input image; the heatmap generated by RAM for the feature p_{T} and its superposition with the input image; the heatmap generated by IntGrad (IG) and its superposition with the input image; the heatmap generated by SmoothGrad (SG) and its superposition with the input image. In the title, it is possible to compare the associated ground-truths ('Real') and the predictions of the model ('Predicted'). In this example, the denoised pattern is unusual and observations between layers 3 and 4 of the RPC chambers are missing, making the trajectory extremely difficult to recognize and resulting in a FP event. We can explain this erroneous outcome by observing the generated heatmaps: the model has attributed importance to noise and even background pixels that should be instead zeroed, resulting in multiple highlighted vertical lines, responsible for the high predicted momentum. Users observing these maps and not having access to the denoised image with its labels should conclude that here attribution methods are not able to find a unique pattern and that the associated prediction is therefore confused and cannot be considered reliable

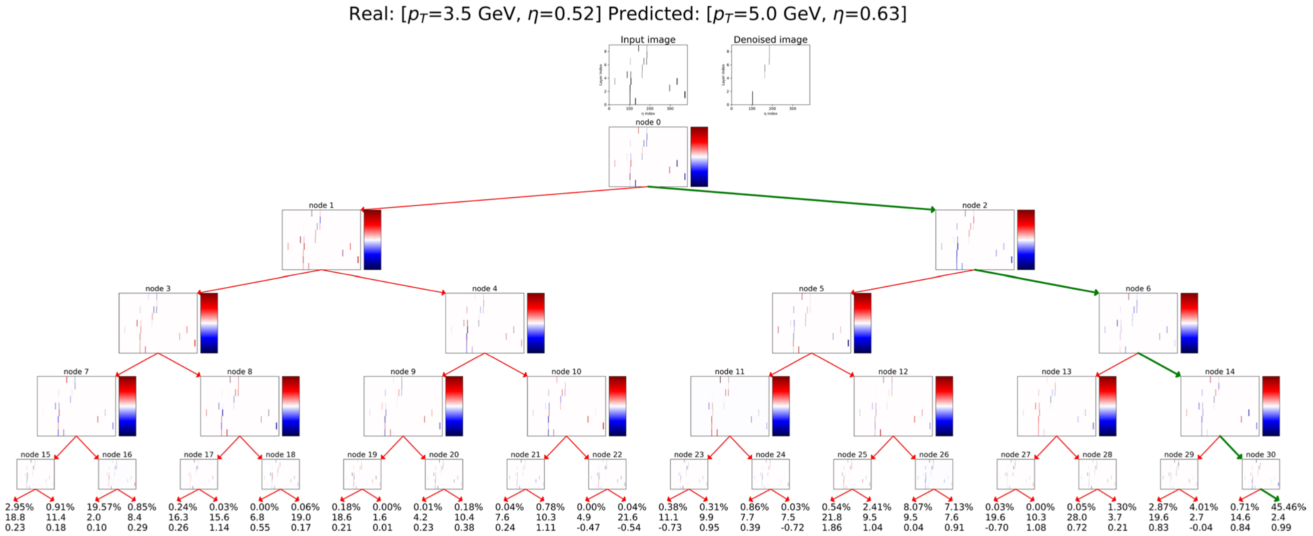


Fig. 8 Explanation through ConvSDT: the visualization shows the flow of the input image (on top, next to its denoised version) into the tree. Each inner node is represented as a correlation map with the input; a leaf node, instead, is characterized by three textual components: the probability (in percentage) of reaching it and the corresponding predictions of p_{T} and η to be weighted accordingly. There exists a *maximum probability path* in green, leading to the leaf with the highest probability, and minor red paths with lower probability. In the title, it is possible to compare the associated ground-truths ('Real') and the predictions of the model ('Predicted'). In this example, the denoised pattern is unusual and observations between layers 3 and 4 of the RPC chambers are missing, making the trajectory extremely difficult to recognize. Nonetheless, the soft tree generates a low predicted momentum, produced with the main contribution of the maximum probability path of 45.46% leading to 2.4 GeV, correctly classifying the event as TN. In addition to the high confidence of the tree for this case, a further marker of reliability is given by the fact that the actual pattern is well considered by looking at the correlations of the maps associated to the inner nodes along the maximum path, showing the pattern of hits more correlated (red color) or anti-correlated (blue color) with the splitting probability

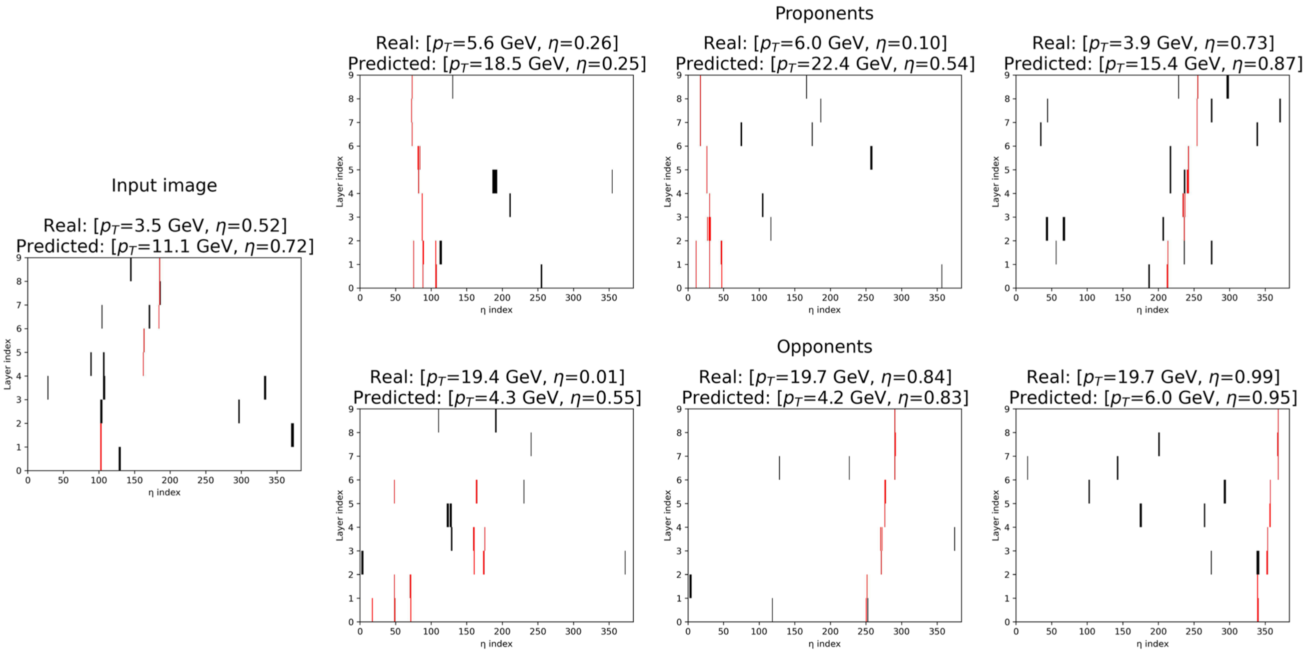


Fig. 9 Explanation through TracIn method: on the left-hand side, the image passed to the model labeled as 'Input image'; on the right-hand side, two distinct groups of images containing the 3 most influential Proponents, on the top, and Opponents, on the bottom. Each image comprehends in the same plot both the noisy hits in black and the actual pattern of the muon in red and is also equipped with a title containing the associated ground-truths ('Real') and the predictions of the model ('Predicted'). In this example, the denoised pattern is unusual and observations between layers 3 and 4 of the RPC chambers are missing, making the trajectory extremely difficult to recognize. Nonetheless, TracIn CNN outputs a medium momentum of 11.1 GeV, still in the correct range to classify this event as TN. This classification can be considered reliable thanks to the fact that the associated Proponents are reporting low real momenta too, demonstrating that these coherent samples have positively influenced the prediction of the model. Opponents instead, show high real p_T, and a predicted $\hat{p}_T$ close to the real p_T of the input image

powerful explainability of models' outcomes and a deeper comprehension of the decisions taken during the inference process.

xAI Use-Case 2: Disagreement Between Methods

The second case of convergent explanations addresses the scenario when methods do not share a uniform direction for their outputs like in the previous example, resulting in conflicts and misinterpretations. Here, convergence has the ability to detect erroneous decisions, understand sources of error and possibly recover the right result with the other methods of the array.

Similarly to the previous section, a representative example of this case is shown in Fig. 7 (attribution methods), Fig. 8 (soft tree methods), and Fig. 9 (data attribution). Starting again from the heatmaps grouped in Fig. 7, it is interesting to notice that the denoised pattern of the muon is unusual and observations between layers 3 and 4 of the RPC chambers are missing, due to simulated malfunctioning in the detector, making the trajectory extremely difficult to recognize. This results in a False Positive event where a low-momentum sample with $p_T = 3.5$ GeV is predicted with a high $\hat{p}_T = 18.2$ GeV. The erroneous outcome is explained observing the generated heatmaps: model has attributed importance (or heat) to noise and even background pixels that should be instead zeroed, resulting in multiple highlighted vertical lines, responsible for the high predicted p_T. An end-user observing these maps and not having access to the denoised image with its labels should conclude that attribution methods are not able to find a unique pattern like in the previous case: the associated prediction is therefore confused and can not be considered a certain output of the CNN.

The confusion derived by investigating outcomes of attribution is solved thanks to convergence by observing the other methods of the array. The soft tree (Fig. 8) generates a low predicted momentum of 5.0 GeV, clearly in conflict with the uncertain $\hat{p}_T$ from Attribution CNN, produced with the main contribution of the maximum probability path of 45.46% leading to 2.4 GeV; moreover, all the other leaf nodes with a consistent contribution, i.e., with a probability greater than 2%, are mainly directing the final output to the same interval of low momentum, meaning that the tree is particularly confident that the real value does not belong to the Positive range. Confirming

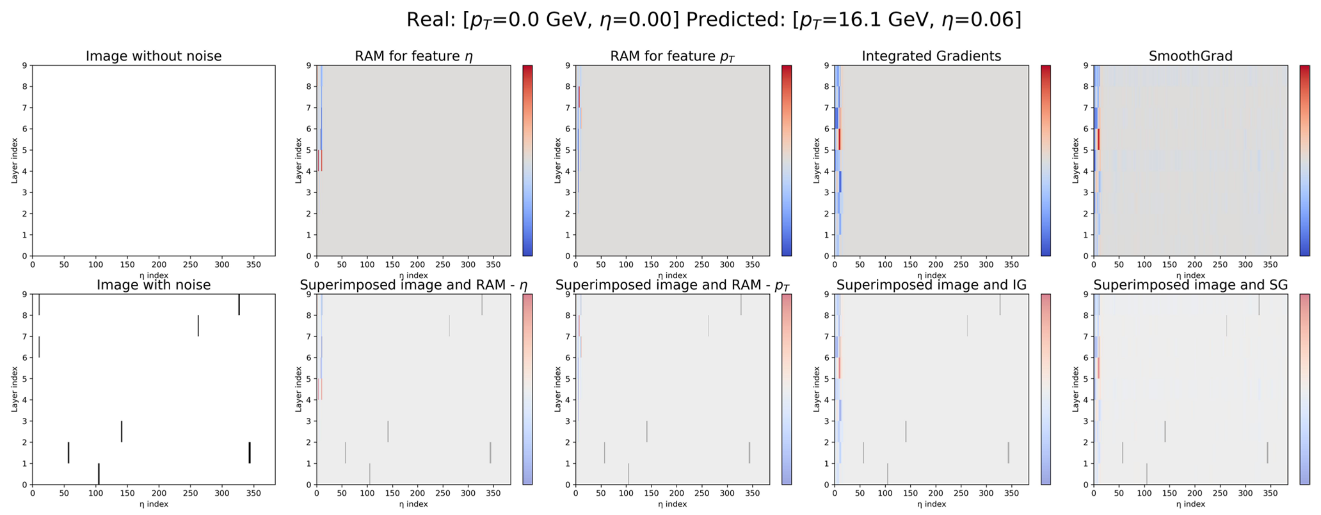

Real: [p_T=0.0 GeV, η=0.00] Predicted: [p_T=16.1 GeV, η=0.06]

Fig. 10 Explanation through attribution methods: five columns showing, from left to right, the image passed as input to the model labeled as ‘Image with noise’ and its denoised version as ‘Image without noise’; the heatmap generated by RAM for the feature η and its superposition with the input image; the heatmap generated by RAM for the feature p_T and its superposition with the input image; the heatmap generated by IntGrad (IG) and its superposition with the input image; the heatmap generated by SmoothGrad (SG) and its superposition with the input image. In the title, it is possible to compare the associated ground-truths (‘Real’) and the predictions of the model (‘Predicted’). In this example, an only-noise event, namely a scenario with no muonic traces with zeroed p_T and η, all three methods have individuated a straight line along the left side, motivating both the high predicted momentum and the ≈ 0 outputted pseudorapidity. In this case, the two hits in the upper left corner of the input image have wrongly suggested to the network the partial presence of a straight trajectory, deceiving the model to predict such high p_T. Users observing the explanation can understand this is an only-noise image because it is unlikely that a complete path in the heatmap is detected in the occurrence of only two hits in the image (from sensors 7 and 9): this is a clear situation when the confusion due to noise is directly individuated

that the actual pattern was considered by looking at the correlations of the inner nodes, the ConvSDT of this case is perfectly able to recover the problems of the initial explanations from attribution and provide clear motivations to correct the previous wrong decision of the CNN.

A further confirmation of this recovery is given by TracIn whose explanation is contained in Fig. 9: TracIn CNN outputs a momentum of 11.1 GeV, slightly greater than the one of ConvSDT but still belonging to the right region of the confusion matrix, and the most influential Proponents all have a low p_T in the Negative interval of momenta.

In the end, even if starting from a confusing explanation, the convergent set of approaches has managed to restore the coherence between predictions and interpretations of model’s decisions and to still guarantee a reliable explainability of the ML algorithms. A different conclusion would have been deduced when, on the opposite, all the methods did not produce clear explanations, for instance, if the tree had leaves with high probabilities pointing to contrasting predictions; in these situations, convergence is still useful to understand that the predictions should be discarded because the models were particularly inefficient with those specific input images.

xAI Use-Case 3: Prediction on Only-Noise Images

The third and last case of explainability results regards the behavior of the models with only-noise images as input. Previously, we already described how our architectures are perfectly able to reject this kind of events, i.e., classifying them as True Negatives; however, it is still important to study the instances erroneously predicted with a high p_T (for instance, $> 10--15$ GeV) that should provide information about how to reduce the fake muons rate, namely individuating which fictitious hits due to noise are mainly confused with the real path of the particle.

The related example we propose is represented in Figs. 10, 11 and 12 where an image with sparse noisy hits randomly spread in the picture is passed as input to the model. Interesting observations are derived when focusing on the heatmaps of Fig. 10: all three methods have individuated a straight and vertical line along the left side, thinner in RAMs and thicker in IntGrad and SmoothGrad, motivating both the high predicted momentum of 16.1 GeV and the ≈ 0 outputted pseudorapidity. In this case, the two hits in the upper left corner of the input image have wrongly suggested to the network the partial presence of a straight trajectory, deceiving the model to predict such high p_T. The user observing these explanations can intuitively understand this is the case of an only-noise image

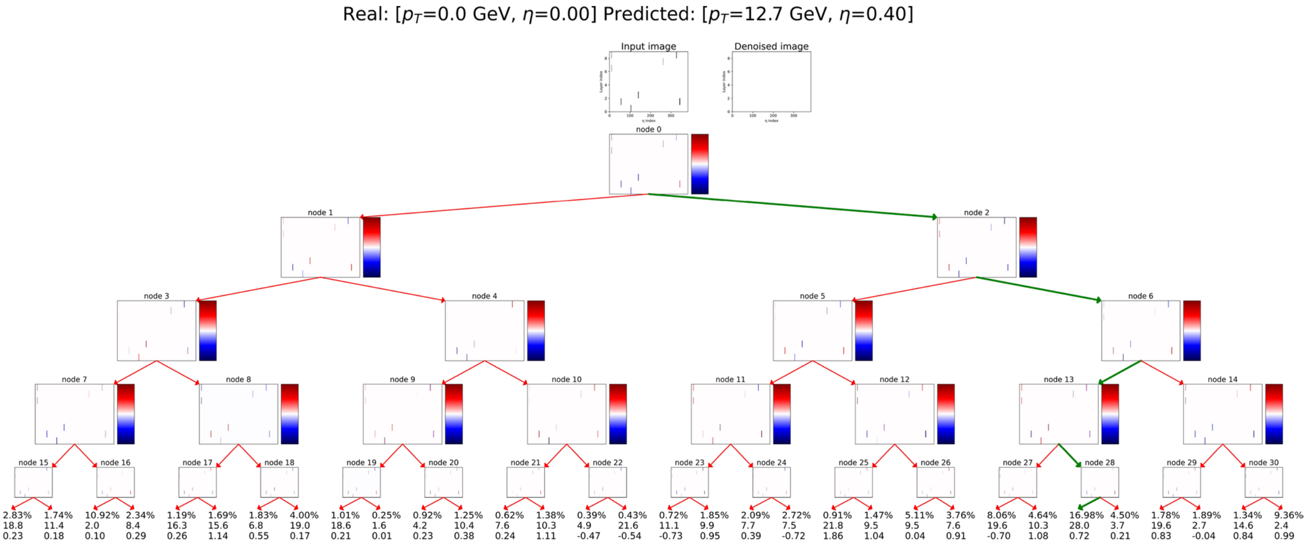
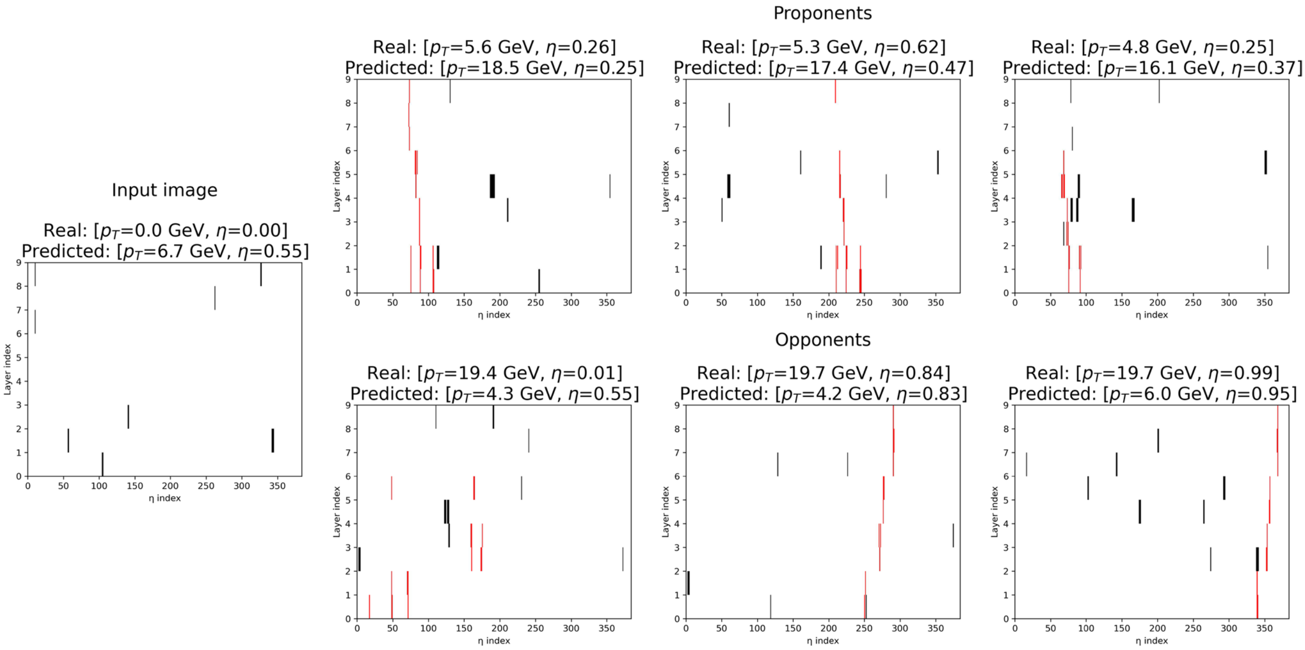

Real: [p_T=0.0 GeV, η=0.00] Predicted: [p_T=12.7 GeV, η=0.40]

Fig. 11 Explanation through ConvSDT: the visualization shows the flow of the input image (on top, next to its denoised version) into the tree. Each inner node is represented as a correlation map with the input; a leaf node, instead, is characterized by three textual components: the probability (in percentage) of reaching it and the corresponding predictions of p_T and η to be weighted accordingly. There exists a *maximum probability path* in green, leading to the leaf with the highest probability, and minor red paths with lower probability. In the title, it is possible to compare the associated ground-truths (‘Real’) and the predictions of the model (‘Predicted’). In this example, an only-noise event, namely a scenario with no muonic traces with zeroed p_T and η, the tree erroneously outputs a high momentum. However, users are warned that there is something causing confusion and that, therefore, the associated prediction is not certain: several leaf nodes with relevant contributions are leading to contrasting predictions. The maximum probability is indeed relatively low, 16.98%, and there are different other leaves with close weights that disagree with each other. This explains how the associated final prediction for this case cannot be considered trustworthy


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]

Fig. 12 Explanation through TracIn method: on the left-hand side, the image passed to the model labeled as ‘Input image’; on the right-hand side, two distinct groups of images containing the 3 most influential Proponents, on the top, and Opponents, on the bottom. Each image comprehends in the same plot both the noisy hits in black and the actual pattern of the muon in red and is also equipped with a title containing the associated ground-truths (‘Real’) and the predictions of the model (‘Predicted’). In this example, an only-noise event, namely a scenario with no muonic traces with zeroed p_T and η, the network correctly generates a low momentum prediction, rejecting the event. Observing the Proponents, it is possible to understand the reasons motivating the choice: the most positively influential training samples sustain the prediction with their low real momentum, strengthening the validity of the CNN’s decision

because it is unlikely that a complete path in the heatmap is detected in the occurrence of only two hits in the image (from sensors 7 and 9): this is a clear situation when the source of confusion from noise is directly individuated.

Proceeding with the soft tree intrinsically interpretable structure (Fig. 11), the user is again warned that there is something causing confusion and that, therefore, the associated prediction is not certain: several leaf nodes with relevant contributions are leading to contrasting predictions. The maximum probability is indeed relatively low, 16.98%, and there are different other leaves with close weights that disagree with each other. Consequently, the associated final prediction of 12.7 GeV cannot be considered trustworthy and the hypothesis the input is an only-noise image is now even more consistent, given that the confusion is explained by the fact the ConvSDT did not expect to predict this category of samples.

Concluding the convergent analysis with TracIn, the associated CNN is the only one able to predict a relatively low $\hat{p}_T$ of 6.7 GeV and the labels of the Proponents in Fig. 12 are close to the one predicted as output, indicating that this model was capable to perceive the low momentum associated to the input. In the end, the initial confusion remains because the other methods of our array have generated conflicting outcomes, the input sample of this example should be then marked as ambiguous and probably only-noised: even in this scenario, the convergence of results suggests users the most likely nature of the input event, managing to prove the inconsistency of the decisions of the models.

Conclusions

Concluding, the proposed high-energy physics use-case was addressed from multiple points of view concerning the accuracy of the methods, their efficiency, and a varied set of explainability tools. We showed that it is possible to implement performing ML models and to customize explainability approaches to guarantee predictions' explanations needed in the critical application domain of the real-time filtering system used in HEP. We were able to implement a large set of different xAI techniques based on cutting-edge methods ranging from input attribution to intrinsically explainable models, capable to guarantee competitive performance, while, most importantly, providing effective ways to explain their decisions. This sets an important step in our final objective to generate an array of convergent techniques able to guarantee explanations in different formats to be addressed by different categories of end-user and problems, providing at the same time a variegate comprehension of model's behavior. Moreover, as the developed methodology focuses both on correct and wrong predictions, it allows users to detect whether the model's outcomes can be considered reliable or to diagnose its limitations by investigating the causes of errors and further optimizing the ML algorithm.

Author Contributions LM: conceptualization, formal analysis, methodology, implementation, visualization, writing—original draft, writing—review and editing; LN: conceptualization, formal analysis, methodology, implementation, visualization, writing—original draft, writing—review and editing; SG: conceptualization, data curation, formal analysis, investigation, methodology, supervision, writing—review and editing; CN: conceptualization, formal analysis, methodology, supervision, writing—review and editing; SS: formal analysis, methodology, supervision, writing—review and editing.

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement. This research has been supported by the CHIST-ERA grant number CHIST-ERA-19-XAI-009, and by PNRR MUR project PE0000013-FAIR.

Availability of Data and Materials The datasets generated during and/or analyzed during this study are available in Zenodo at: https://zenodo.org/record/7810813 (DOI: doi.org/10.5281/zenodo.7810813).

Code Availability The code developed for this work is available at https://github.com/lorin98/xAI-HEP.

Declarations

Conflict of Interest The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Ethical Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication All authors gave final approval for publication and agreed to be held accountable for the work performed herein.

References

1. Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 6:52,138-52,160

2. Agarwal G, Hay L, Iashvili I et al (2021) Explainable AI for ML jet taggers using expert variables and layer-wise relevance propagation. J High Energy Phys 5:1–36
3. Ahmetoğlu A, İrsoy O, Alpaydın E (2018) Convolutional soft decision trees. In: Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Springer, pp 134–141
4. Alber M, Lapuschkin S, Seegerer P et al (2019) Investigate neural networks! J Mach Learn Res 20(93):1–8
5. Amari S (1993) Backpropagation and stochastic gradient descent method. Neurocomputing 5(4):185–196
6. Bach S, Binder A, Montavon G et al (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE 10(7):1–46
7. Balestriero R (2017) Neural decision trees. arXiv preprint http://arxiv.org/abs/1702.07360
8. Binder A, Bach S, Montavon G, et al (2016) Layer-wise relevance propagation for deep neural network architectures. In: Information science and applications (ICISA) 2016, Springer, pp 913–922
9. Bistron M, Piotrowski Z (2021) Artificial intelligence applications in military systems and their influence on sense of security of citizens. Electronics 10(7):871
10. Bradshaw L, Chang S, Ostdiek B (2022) Creating simple, interpretable anomaly detectors for new physics in jet substructure. Phys Rev D 106(3):035014
11. Chakraborty A, Lim SH, Nojiri MM (2019) Interpretable deep learning for two-prong jet classification with jet spectra. J High Energy Phys 7:1–36
12. Collaboration TA (2008) The ATLAS experiment at the CERN large hadron collider. J Instrum 3(08):S08,003-S08,003
13. Faucett T, Thaler J, Whiteson D (2021) Mapping machine-learned physics into a human-readable space. Phys Rev D. https://doi.org/10.1103/PhysRevD.103.036020
14. Francescato S, Giagu S, Riti F et al (2021) Model compression and simplification pipelines for fast deep neural network inference in FPGAS in hep. Eur Phys J C 81(11):969
15. Frosst N, Hinton G (2017) Distilling a neural network into a soft decision tree. In: CEX Workshop, 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017)
16. Gou J, Yu B, Maybank SJ et al (2021) Knowledge distillation: a survey. Int J Comput Vision 129:1789–1819
17. Irsoy O, Yıldız OT, Alpaydın E (2012) Soft decision trees. In: 21st International Conference on Pattern Recognition (ICPR2012), IEEE, pp 1819–1822
18. Islam MR, Ahmed MU, Barua S et al (2022) A systematic review of explainable artificial intelligence in terms of different application domains and tasks. Appl Sci 12(3):1353
19. Khot A, Neubauer MS, Roy A (2022) A detailed study of interpretability of deep neural network based top taggers. arXiv preprint http://arxiv.org/abs/2210.04371
20. Kietzmann J, Paschen J, Treen E (2018) Artificial intelligence in advertising: How marketers can leverage artificial intelligence along the consumer journey. J Advert Res 58:263–267
21. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: 3rd International Conference for Learning Representations (ICLR)
22. Krishna S, Han T, Gu A, et al (2022) The disagreement problem in explainable machine learning: a practitioner's perspective. arXiv preprint http://arxiv.org/abs/2202.01602
23. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. Adv Neural Inf Process Syst. https://doi.org/10.48550/arXiv.1705.07874
24. Luo H, Cheng F, Yu H et al (2021) SDTR: soft decision tree regressor for tabular data. IEEE Access 9(55):999–56011
25. Mokhtar F, Kansal R, Diaz D, et al (2021) Explaining machine-learned particle-flow reconstruction. In: Machine Learning for Physical Sciences Workshop, NeurIPS 2021
26. Montavon G, Lapuschkin S, Binder A et al (2017) Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recognit 65:211–222
27. Montavon G, Binder A, Lapuschkin S, et al (2019) Layer-wise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning pp 193–209
28. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
29. Pruthi G, Liu F, Kale S et al (2020) Estimating training data influence by tracing gradient descent. Adv Neural Inf Process Syst 33:19,920-19,930
30. Samek W, Wiegand T, Müller KR (2017) Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint http://arxiv.org/abs/1708.08296
31. Smilkov D, Thorat N, Kim B, et al (2017) Smoothgrad: removing noise by adding noise. arXiv preprint http://arxiv.org/abs/1706.03825
32. Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: 2017 International Conference on Machine Learning (ICML), PMLR, pp 3319–3328
33. Svenmarck P, Luotsinen L, Nilsson M, et al (2018) Possibilities and challenges for artificial intelligence in military applications. In: NATO Big Data and Artificial Intelligence for Military Decision Making Specialists' Meeting, pp 1–16
34. Wang Z, Yang J (2017) Diabetic retinopathy detection via deep convolutional networks for discriminative localization and visual explanation. arXiv preprint http://arxiv.org/abs/1703.10757
35. Yang Y, Morillo IG, Hospedales TM (2018) Deep neural decision trees. In: ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)
36. Zhang Y, Tiňo P, Leonardis A et al (2021) A survey on neural network interpretability. IEEE Trans Emerg Top Comput Intell 5(5):726–742
37. Zhou B, Khosla A, Lapedriza A, et al (2016) Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2921–2929