

## Article

# Scattering-Based Machine Learning Algorithms for Momentum Estimation in Muon Tomography

Florian Bury <sup>1,\*</sup>  and Maxime Lagrange <sup>2,\*,†</sup> <sup>1</sup> Particle Physics Department, University of Bristol, Bristol BS8 1TL, UK<sup>2</sup> Centre de Cosmologie, de Physique des Particules et de Phénoménologie, Université catholique de Louvain, 1348 Ottignies-Louvain-la-Neuve, Belgium

\* Correspondence: florian.bury@cern.ch (F.B.); maxime.lagrange@uclouvain.be (M.L.)

† MODE Collaboration: <https://mode-collaboration.github.io/>.

**Abstract:** Muon tomography leverages the small, continuous flux of cosmic rays produced in the upper atmosphere to measure the density of unknown volumes. The multiple Coulomb scattering that muons undergo when passing through the material can either be leveraged or represent a measurement nuisance. In either case, the scattering dependence on muon momentum is a significant source of imprecision. This can be alleviated by including dedicated momentum measurement devices in the experiment, which have a potential cost and can interfere with measurement. An alternative consists of leveraging information on scattering withstood through a known medium. We present a comprehensive study of diverse machine-learning algorithms for this regression task, from classical feature engineering with a fully connected network to more advanced architectures such as recurrent and graph neural networks and transformers. Several real-life requirements are considered, such as the inclusion of hit reconstruction efficiency and resolution and the need for a momentum resolution prediction that can improve reconstruction methods.

**Keywords:** muon scattering tomography; momentum estimation; machine learning; supervised learning; regression



Academic Editors: Tommaso Dorigo,  
Roberto Ruiz de Austri Bazan, Jose  
Salt and Pietro Vischia

Received: 31 January 2025

Revised: 24 February 2025

Accepted: 3 April 2025

Published: 14 April 2025

**Citation:** Bury, F.; Lagrange, M.  
Scattering-Based Machine Learning  
Algorithms for Momentum  
Estimation in Muon Tomography.  
*Particles* **2025**, *8*, 43. <https://doi.org/10.3390/particles8020043>

**Copyright:** © 2025 by the authors.  
Licensee MDPI, Basel, Switzerland.  
This article is an open access article  
distributed under the terms and  
conditions of the Creative Commons  
Attribution (CC BY) license  
(<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Muon tomography encompasses imaging methods based on the measurement of absorption or deflection of atmospheric muons. The two main electromagnetic phenomena muons are subjected to, absorption by energy loss and multiple Coulomb scattering through matter, can be separately exploited by the two types of muon imaging: muon transmission and muon scattering tomography. In the past decade, applications of muon tomography have expanded across a wide range of fields, including geosciences [1], nuclear safety [2], archaeology [3], civil engineering [4], and security applications [5], and the technology has been gradually moving from high-energy physics to concrete engineering applications. While significant advancements have been made in detector design and reconstruction algorithms, current muon detection systems still lack the ability to measure muon momentum and rely solely on track measurements for imaging tasks.

This paper is organized as follows: in Section 2, we introduce the origin and flux of muons, how their momentum impacts both scattering and transmission tomography, and how it can be measured through scattering in known material; in Section 3, we discuss multiple machine learning (ML) algorithms trained to exploit the detector hit information to regress the momentum and improve its reconstruction accuracy; finally, in Section 4,

we study a few requirements needed to apply these models to the constraints of a real-life experiment.

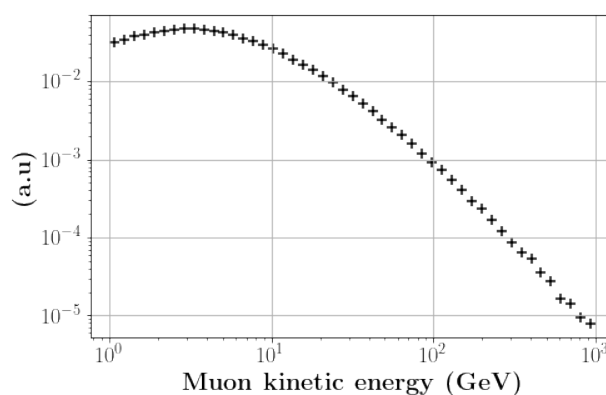
## 2. Muon Scattering and Momentum

### 2.1. Muon Flux at Sea Level

Both types of muon scattering and transmission muon tomography take advantage of the atmospheric muon flux. Atmospheric muons are abundantly produced by the interaction of primary cosmic rays within the upper atmosphere and represent approximately 80% of all charged particles reaching sea level, with a flux through a horizontal surface of around one particle per  $\text{cm}^2$  per minute. An example of the cosmic ray muon flux is given by the Gaisser parametrization [6]

$$\frac{dN}{dEd\Omega} = 0.14E^{-2.7} \left( \frac{1}{1 + \frac{1.1E \cos\theta}{115\text{GeV}}} - \frac{0.054}{1 + \frac{1.1E \cos\theta}{850\text{GeV}}} \right) \text{cm}^2\text{s}^{-1}\text{sr}^{-1}\text{GeV}^{-1}. \quad (1)$$

At sea level, the muon energy spectrum through a horizontal surface peaks at 4 GeV, as shown in Figure 1 in a wide momentum interval from hundreds of MeV up to thousands of GeV. The width of the momentum spectrum translates to different behavior in the interaction with matter. While high-momentum muons (100 GeV and above) are useless to muon scattering tomography as they traverse matter without undergoing measurable deflections, they are crucial to muon transmission tomography, as they can traverse large thicknesses of material and allow for probing kilometre-scale structures. The effect of the momentum on muon scattering and transmission tomography is further detailed in Section 2.2 and Section 2.3, respectively.



**Figure 1.** The distribution of atmospheric muons' energy obtained at sea level through a horizontal surface with the CRY cosmic-ray particle shower library [7], the crosses represent the bin content.

### 2.2. Muon Momentum in Muon Scattering Tomography

Like other charged particles, when going through matter, muons collide with electrons and nuclei, thereby losing energy and undergoing deflections. This process is called multiple Coulomb scattering and can be approximated by a Gaussian distribution. The differential distribution of scattered muons can therefore be written as

$$\frac{dN}{d\theta} = \frac{\theta}{\sqrt{2\pi}\theta_0} e^{-\theta^2/2\theta_0^2}, \quad (2)$$

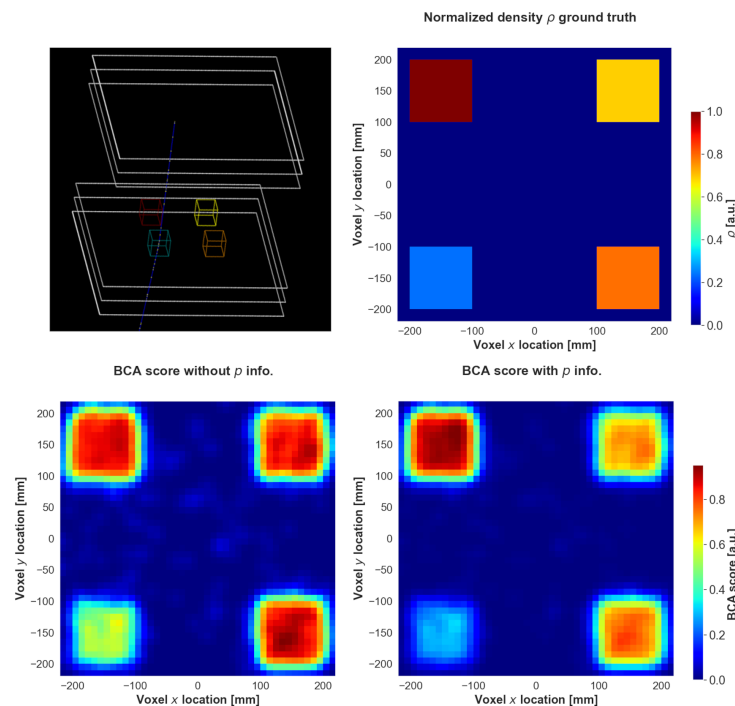
where the standard deviation  $\theta_0$  is given by Lynch and Dahl [8]:

$$\theta_0 = \frac{13.6 \text{ MeV}}{p\beta c} \sqrt{\frac{x}{X_0}} \left[ 1 + 0.038 \ln \left( \frac{x}{X_0\beta^2} \right) \right]. \quad (3)$$

$p$  and  $\beta c$  are the muon's momentum and speed, and  $x/X_0$  is the thickness of the scattering medium in radiation length. Radiation length is defined as the mean length of the material at which the energy of an electron is reduced by a factor  $1/e$ , and it can be found in the PDG [9]. This equation only holds under the small-angle scatter assumption, as hard scatterings on nuclei can cause non-Gaussian tails.

The scattering angle is thus a function of both the material properties ( $X_0$ ) and muon momentum ( $p$ ). In the context of a muon scattering tomography experiment, individual deflections of high-momentum muons through dense material and low-momentum muons through light material are indistinguishable. Since image reconstruction and material inference algorithms used in muon scattering tomography utilize measured scattering angles as input, they suffer from a lack of knowledge on muon momentum.

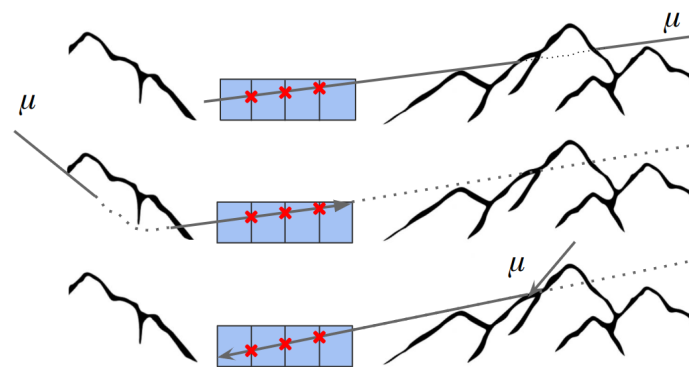
Some of these algorithms, like Angle Statistic Reconstruction [10], Binned Clustered Algorithm (BCA) [11], or momentum-aware POCA [12–14], can utilize momentum knowledge if available. To illustrate the impact of including the true momentum—as used in the simulations—in the density reconstruction, an example of a scattering muon tomography setup simulation is shown in Figure 2, with the predictions of the BCA with the true momentum—as used in simulation—and without any momentum information. Their performances are quantified through the mean squared error  $MSE = \frac{1}{N} \sum_{i=0}^N (Y - \hat{Y})^2$  and mean absolute error  $MAE = \frac{1}{N} \sum_{i=0}^N |Y - \hat{Y}|$  of the normalized density prediction  $\hat{Y}$ . The figure shows that including true momentum information during density inference not only reduces noise but also improves the quality of the inference. While the improvement depends on the algorithm, the integration of a momentum measurement system into any muon scattering tomography experiment is extremely relevant.



**Figure 2.** Muon scattering tomography setup (top left) of four cubes of different materials (top right): lead (red), copper (orange), iron (yellow), and aluminium (blue). The muon flux is simulated using the CRY library [7], and the muon propagation and scattering with GEANT4 [15]. The predicted densities, after eight hours of data taking, using standard BCA (bottom left), have performance scores of  $MAE = 0.085$  and  $MSE = 0.046$ , while the momentum-aware density predictions (bottom right) have  $MAE = 0.058$  and  $MSE = 0.028$ . The inclusion of the true momentum information helps to discriminate between the most dense materials, for example, between lead and iron. This study was conducted by the authors of this work.

### 2.3. Muon Momentum in Muon Transmission Tomography

Muon transmission tomography produces radiographic images by measuring the absorption of the cosmic muon flux through an object. Several experiments in the geoscience sector intend to infer the in-depth kilometre-scale volcano structures like the Puy-de-Dome [16] in France, the Mount Vesuvius in Italy [17], or Soufrière in Guadeloupe [18]. Given the large thicknesses of rock that muons have to traverse before reaching the detection system, only high-momentum muons (roughly above 100 GeV) can traverse the object without decaying and thus constitute most of the signal. On the other hand, low-momentum muons (5 GeV and below), which represent around 90% of the total muon flux, can scatter and enter the detector acceptance without interacting with the region of interest, thus mimicking high-momentum muons and biasing the muon flux absorption measurement, as shown in Figure 3. Since the flux of high-zenith-angle muons (near-horizontally) decreases by 2 to 4 orders of magnitude after a few kilometres of rock, it becomes comparable with the flux of the non-penetrating muons and makes the rejection of such events determinant.



**Figure 3.** An illustration of the different kinds of muons entering a muon detector and their acceptance: high-momentum muons traversing the target volcano (top) and low-momentum muons entering the detector by coincidental scattering, where the muon's direction is indicated by the arrow (middle and bottom).

While some experiments use a lead block to shield their detectors from low-momentum muons [17,18], reject them from the goodness of the track fit [19], or rely on coarse time of flight measurements to reject background muon tracks [20], the inclusion of a muon momentum measurement module in the detection system could further improve the background rejection capabilities.

### 2.4. Muon Momentum Measurement

Many particle physics experiments rely on the measurement of charged particle momentum. While momentum inference from the particle trajectory in a strong magnetic field is used extensively in high-energy physics experiments at the LHC, it is not suitable for muon tomography. Generating a high magnetic field is extremely costly as it requires large and complex magnets powered by high-power sources, which also imposes strong logistical constraints. It also interferes with the image reconstruction algorithms, as it becomes harder to disentangle muon scattering from the deflections caused by the magnetic field.

Two main techniques have been investigated in the literature, some exploiting Cherenkov radiation of muons through dedicated detectors [21,22] and others exploiting multiple Coulomb scattering of muons through passive layers of high-density and high-Z materials (e.g., lead, iron) [19,23–25]. The work presented in this paper focuses on the latter technique.

Figure 4a gives an example of a momentum measurement module built with three dense material layers (e.g., lead) of thickness  $\Delta s$  used to scatter the muons. The muon

hits are measured by eight detector layers separated by a gap  $\Delta g$ . The simulated detector panels are non-physical infinitely thin detectors recording the exact position of the muon. As such, muon scattering only occurs when propagating through the lead panels. This is a reasonable approximation as detector panels are typically thinner and have a lower density—e.g., gas mixtures, scintillators—than the scattering material. The selection of tracks is limited to those leaving a hit in each of the eight detector panels. From those eight hits, four linear segments can be determined from the hits, and three scattering angles  $\Delta\theta_i$  can be reconstructed. Contrary to the typical muon scattering tomography task, where one wants to infer the density from the measured scattering angle, momentum can be estimated by inverting Equation (3):

$$p = \frac{13.6 \text{ MeV}}{\theta_0} \sqrt{\frac{x}{X_0}} \left[ 1 + 0.038 \ln \left( \frac{x}{X_0} \right) \right], \quad (4)$$

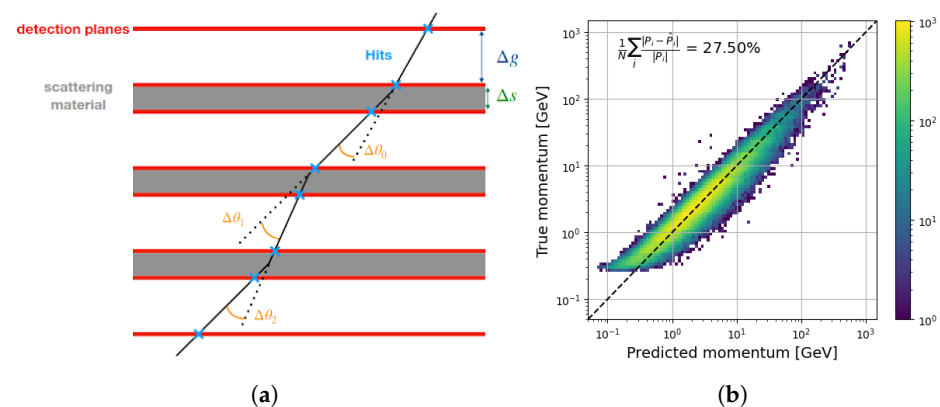
where muons are assumed to be relativistic ( $\beta \simeq 1$ ). In this context, the scattering material  $X_0$  is pre-defined, and the distance traversed by the muon  $x$  is estimated from the track measurement. Measuring the scattering angle of the muons through  $N$  layers of material gives a rough estimate of the scattering angles' standard deviation  $\theta_0$ :

$$\theta_0 = \frac{1}{\sqrt{2}} \theta_{RMS}^{scatt} = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^N (d\theta_i^{scatt})^2}. \quad (5)$$

Together with Equation (4), the scattering angles' standard deviation  $\theta_0$  provides an estimation of the muon momentum, as illustrated in Figure 4b. The mismeasurement tail at low momentum originates from the fact that the muon energy loss through 4 cm of lead has the same order of magnitude as its momentum, for example, a 200 MeV muon loses about 55 MeV through 4 cm of lead. This could be circumvented by the use of Kalman filters, as demonstrated in Ref. [26].

While such an approach has satisfactory results, it remains suboptimal as only a fraction of the information accumulated by the detector is used. Only scattering angle information can be plugged in Equation (3), representing three measurable parameters, while the detectors measure the muon position at eight locations.

In Section 3, different machine learning algorithms are trained to improve the momentum prediction accuracy. The dataset used during the training follows the setup of Figure 4a. The CRY library [7] is used to generate the muon flux, and the muon propagation and scattering are handled by GEANT4 [15].



**Figure 4.** (a) The experimental setup used in this paper: three layers of  $\Delta s = 4$  cm thick lead panels serve as scattering material, separated by  $\Delta g = 25$  cm of air, and the hits are detected by eight two-dimensional planes—agnostic of the detector technology. (b) The result of the muon momentum inference using Equations (4) and (5). The Mean Absolute Percentage Error (MAPE) is shown at the top of the figure.

### 3. Machine Learning Methods

Although physically motivated, using the sole scattering angles in Equation (4) represents a limited use of the information available from the detector. At the lowest level, all the information that can be gathered is contained within the hits registered in the different active layers. The ambition of this section is to explore different ML algorithms and compare their performance.

Historically, the first ML algorithms used in particle physics were boosted decision trees (BDTs) and neural networks (NNs). These simple models could be easily trained with the limited training data of the time, but they required the physicist to carefully craft variables to optimize the discrimination power. Such variables were often physically motivated, for example, an invariant mass or specific angular distance. In a way, this is similar to the use of the scattering angles motivated by Equation (4). Then, riding on the deep learning revolution and helped by the ever-growing availability of high-fidelity simulations, physicists figured out that they could increase their discrimination performance by supplementing these manually crafted variables with a larger set of lower-level information and providing this set to a deep neural network (DNN). This was especially visible in the context of jet tagging in LHC experiments [27], a topic that was and still is a main driver of ML development. Including additional information from each particle identified within a jet provided significant performance improvement and opened the door to a wide range of measurements thought to be impractical until then.

Arguably, a second revolution occurred when physicists started leveraging the symmetries of the problem they were facing directly within the ML model architecture. One such model was based on convolutional neural networks (CNNs), which were initially developed in the context of image processing and leverage local information through a kernel of weights applied across the image. It represents a jet as an image, from the unfolded barrel detector, allowing us to exploit the geometrical information and the translational symmetry [28].

In contrast, the use of recurrent neural networks (RNNs) [29,30], and by extension recursive neural networks [31], focused on the hierarchy relations between the jets' constituents. While both approaches provided significant improvement, notably the ability to solely use low-level information and handle a variable number of inputs—a major limitation of DNNs—they both had limitations. For example, the binning into cells required it to produce an image coming with information loss in a CNN, and the ordering of the objects to be fed to an RNN can introduce hard-to-learn long-range dependencies, despite the introduction of long-short term memory (LSTM) and gated recurrent units (GRUs).

From these considerations started a wide search for models exploiting more useful symmetries and making use of as much information as physicists could consider. Many models were considered, but two of them are of particular note. The first is called ParticleNet [32] and expands the concept of convolution to a graph neural network (GNN). In the context of jet tagging, this is much more appropriate than a CNN because it can make better use of the sparsity of a jet representation. The other model is called ParT [33] and is built from the transformer architecture [34] that replaced RNNs thanks to their self-attention mechanism, better suited to handle any range of dependency. Notably, both models natively include a permutation symmetry, which is better suited to jets and could explain why they outperformed the previous ones.

This section will follow this historical evolution by starting with a DNN, augmenting the input variables, and then exploring the aforementioned more advanced architectures. The implementations of all models were performed using the standard Pytorch [35], PyG [36] for the GNN, and their training through lightning [37].

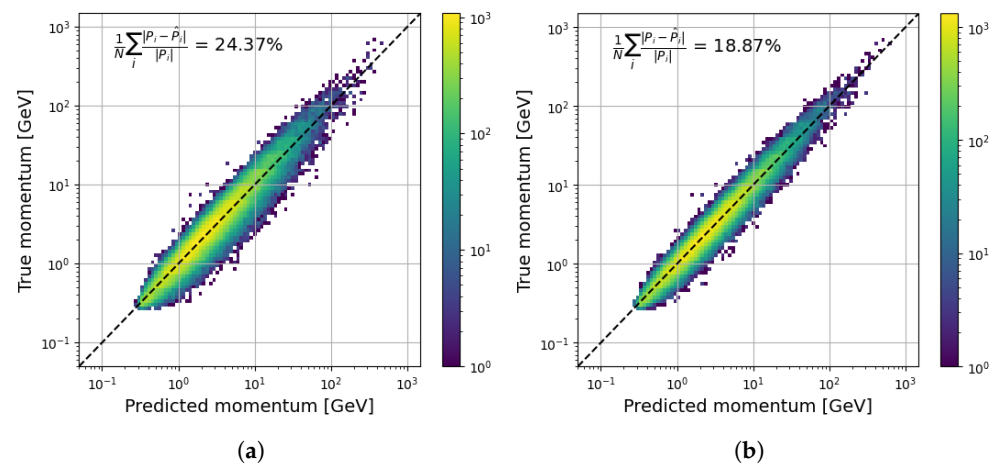
All model hyperparameters can be found in Tables A1–A4. Additionally, all the models are trained with the RAdam optimizer [38], a batch size of 1024, mean squared error (MSE) loss function, and ReduceLROnPlateau learning rate scheduler—starting at  $10^{-3}$  or  $10^{-4}$  depending on the model. Around a million muons were generated in the setup of Figure 4a; 70% are used directly in the training, 10% are used for validation, and the remaining 20% to produce all the illustrations in the following sections, therefore providing unbiased performance estimation.

### 3.1. Deviations as Features

As a first attempt at leveraging more information than through Equation (4), a DNN will be trained on some features to try to regress the muon momentum. The momentum distribution follows the expected flux of Figure 1, convoluted with the detector acceptance. As this distribution spans many orders of magnitude, the actual target of the regression is  $\log_{10} P$ , which will be discussed further in the following.

To have a direct comparison with Figure 4b, only the three scattering angles—as featured in Figure 4a—will be used as input. Similarly to the target, a  $\log_{10}$  operation is applied beforehand, and then they are preprocessed by removing the mean and scaling to unit variance. The result of the regression in Figure 5a shows that the DNN is able to exploit more information from these mere three inputs than Equation (4). There might, however, be more variables we can extract from the hits, for example, the deflection angles and the track residuals, as illustrated in Figure 6. When these features are added into the regression, while keeping the same model,  $\log_{10}$ , and preprocessing, the gain is significant, as illustrated in Figure 5b.

The application of a  $\log_{10}$  to both inputs and targets came from the empirical observation that spans several orders of magnitude and that the angles scale non-linearly from it. High-momentum muons suffer very few deflections, and it becomes hard for the DNN to distinguish them above 100 GeV. The  $\log_{10}$  was then chosen to disentangle them, effectively “zooming in” on that high-momentum region. This, however, should not come as a surprise since Equation (3) implies that  $\log_{10} \theta \propto -\log_{10} P$ , as shown in Figure 7.

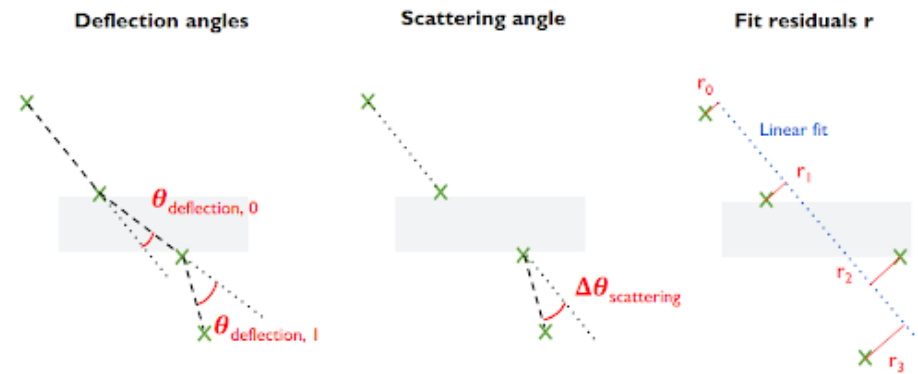


**Figure 5.** Muon momentum regression using a DNN with a different set of features: only scattering angles (a); scattering angles, deflection angles, and fit residuals (b), as shown in Figure 6.

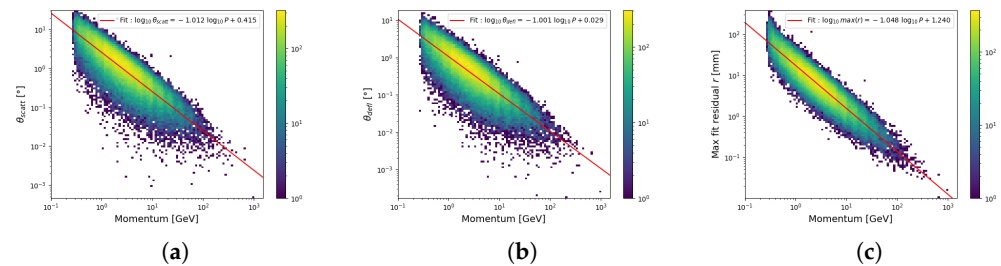
The results of Figure 5 suggest that more information can be obtained by using raw information, namely, detector hits. However, the actual information about the muon momentum is contained in the scattering of the muon compared to a linear track, as implied in Equation (3). High-energy muons have hits that almost form a perfect line, while low-momentum muons typically exhibit a curve or helix around a linear track fit, as shown

in Figure 8a in three dimensions, and their projections on both horizontal axes are shown in Figure 8b. This feature comes from the fact that muons are deflected within a cone whose zenith angle is determined by Equation (2) and with a uniform azimuthal angle, giving the impression of a rotation around the linear track axis.

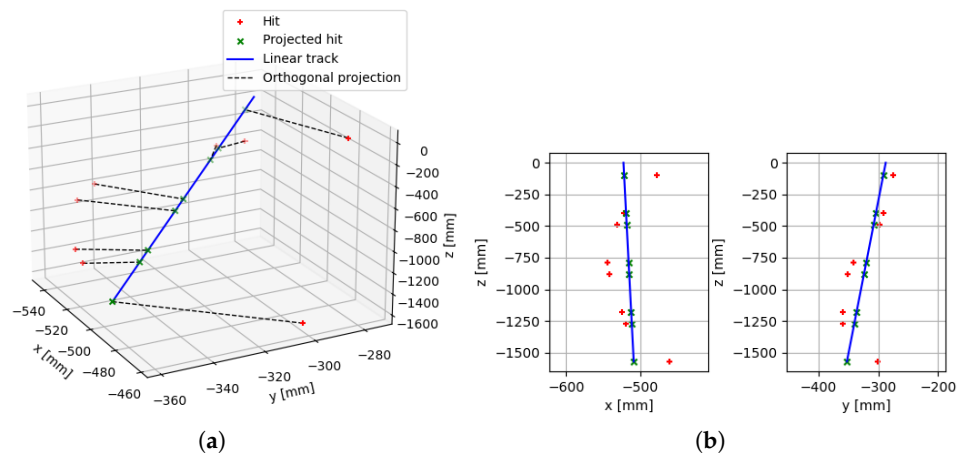
Therefore, what actually matters are the distances from the hits to their orthogonal projections onto the linear track. Instead of considering only the absolute distance, however, to avoid any loss of information, the two-dimensional vector in the perpendicular plane of the track will be used and referred to as a *deviation*.



**Figure 6.** An example of variables that can be extracted from muons' hits in the detector layers. Green crosses represent measured hits, dashed line segments join two consecutive hits, the dotted line is the track linear fit, and the solid rectangles are the scattering material blocks. (Left) Deflection angles are the angles between two consecutive segments. (Middle) Scattering angles are the angles between two segments on each side of the scattering material. (Right) Residuals are the distances between the hits and their orthogonal projections on the fitted linear track.



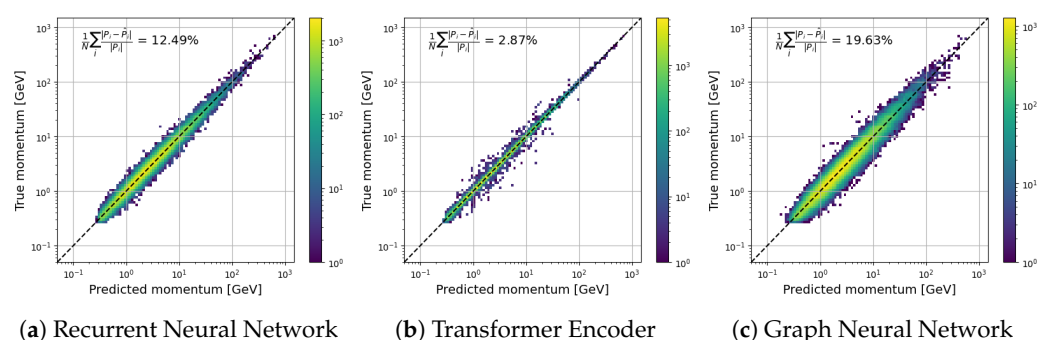
**Figure 7.** The distributions of the different DNN features as defined in Figure 6 as a function of the momentum: scattering angle (a), deflection angle (b) and the fit residuals (c). The result of a linear fit on the logarithmically scaled variables is shown in red.



**Figure 8.** Representation of the muon's deviations in 3D (a) and in XZ and YZ projections (b). The hits measured by the detector planes are used to fit a linear track and then projected onto that track.

### 3.2. Deviations as a Sequence

In order to be fed to an RNN, an order needs to be provided for the deviations. Fortunately, a natural choice consists of following the trajectory of the muon, in this case, from top to bottom. To exploit the azimuthal symmetry of the scattering in Equation (2), the track direction is used as the basis  $z$ -axis, with the deviations expressed in polar coordinates  $(r, \theta)$  instead of Cartesian  $(x, y)$ . This choice of cylindrical basis improved the regression performance when used with sequential algorithms. Additionally, to circumvent the logarithmic relationship with the momentum illustrated in Figure 6, the *log-modulus* function  $f(x) = \text{sign}(x) \log_{10}(|x| + 1)$  is applied on the magnitude of the deviation  $r$ , while preserving the angle. This function is used instead of a pure logarithm to avoid changing the sign of the deviation when  $|r| < 1$ , effectively “zooming” in on the high-energy muon while preserving the shape of the deviations, as illustrated in Figure A1. Finally, the tracks are rotated around their axes, so that the polar angle starts at  $\theta = 0$ . After passing through the RNN layers, the embedding of the last object is fed to a DNN in a *many-to-one* scheme. The result of the regression when the cylindrical coordinates  $(r, \theta, z)$  are provided to the RNN is in Figure 9a.



**Figure 9.** Muon momentum regression using advanced ML algorithms.

The transformer on the other hand is naturally permutation-invariant, which needs to be broken to keep the hierarchy of the deviations along the muon track. In language processing, this is performed through position encoding. Several methods exist; initially, absolute encoding—using sine and cosine functions—is employed, before other methods such as relative encoding are used. In order to fully make use of the three-dimensional geometrical information, the positions of the hits projected on the linear track are fed to a DNN to provide a learnable position encoding. The deviations expressed as  $r$  and  $\theta$  are used as features and encoded using a similar DNN embedding. Both positions and features are first passed through a batch normalization layer, and then their respective encodings are summed and passed through a transformer encoder. Similarly to CaiT vision transformer [39], a token is passed through several cross-attention layers together with the encoder output to summarize its information, and then it is passed to a DNN. The result of the regression is in Figure 9b.

Treating the deviations as a sequence could be exploited to identify upcoming tracks of low-momentum muons acting as backgrounds in the context of transmission tomography since they lose energy when they pass through dense material and their deviations grow larger toward their exit side. This is partially visible in Appendix A, with a visible upward trend in the deviations in  $r$  for low-momentum muons. This effect is, however, extremely limited for high-momentum muons, and it could potentially limit the usefulness of this feature. As this is more relevant as a background rejection proposal, it was deemed outside the scope of this study.

### 3.3. Deviations as a Geometry

To use the geometric information in a CNN, the deviations of Figure A1 need to be binned into pixels. The main issue comes from the resolution of such pixels and the memory footprint. Figure 7 shows that no matter the basis used, the deviations span several orders of magnitude. Using a pixel size that would be fine enough to be able to measure high-momentum muon deviations—around 0.1 mm at 100 GeV—while still having enough pixels to catch the range of low-momentum muons—a few cms at 1 GeV—would mean thousands of pixels per dimension. This would not fit in memory to be used by a three-dimensional CNN. An attempt was made to use two-dimensional projections in the  $x - y$  and  $x - z$  planes, akin to multi-view CNNs from Ref. [40]. This essentially consists of turning the first two graphs of Appendix A into an image, with a single pixel per layer in the  $z$  direction to limit the image memory footprint. Still, to accommodate the different ranges, two CNNs per projection were needed with different resolutions, similar to the idea in Ref. [41]. While this setup did yield a decent regression, it was barely able to beat Equation (4). This was expected considering the information losses due to the projection and binning, as well as the sparsity of the image. However, it did prove that the geometric information could still be exploited.

This led to the use of a GNN instead, which, contrary to a CNN, is perfectly suited to sparse inputs. To build a graph from a track, the hits are rotated and then translated so that the track axis is vertical and centred on  $(x = 0, y = 0)$ . As also performed for the sequence processing, the log-modulus is applied to the distance between each hit and the track, while keeping the same angle to not modify the shape of the graph. The hits are then identified as the graph nodes, with the number of edges connecting each node with the closest neighbour treated as a hyperparameter. An example of graphs when each node is connected to the two closest nodes—unless the node is at the edges of the graph, in which case it only connects to its closest neighbour—is shown in Figure A2 with and without the log-modulus function.

Convolutions in a GNN can be reformulated in the message passing scheme, as implemented in PyG, where edge features of  $\mathbf{x}_i^{(k)}$  at node  $i$  and layer  $k$  can be obtained from the previous layer  $k - 1$  as

$$\mathbf{x}_i^{(k)} = \gamma^{(k)} \left( \mathbf{x}_i^{(k-1)}, \bigoplus_{j \in \mathcal{N}(i)} \phi^{(k)}(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, \mathbf{e}_{ij}) \right), \quad (6)$$

where  $\mathcal{N}(i)$  is the set of neighbours of node  $i$ ,  $\mathbf{e}_{ij}$  is the edge attribute between nodes  $i$  and  $j$ ,  $\bigoplus$  is a differentiable aggregation operation, and  $\gamma$  and  $\phi$  are optional, complex, and differentiable functions, typically DNNs. In particular, for this application, the model used is edgeConv [42], specifically designed for point cloud convolutions, where Equation (6) is particularized to

$$\mathbf{x}_i^{(k)} = \bigoplus_{j \in \mathcal{N}(i)} \phi^{(k)}(\mathbf{x}_i^{(k-1)} || \mathbf{x}_j^{(k-1)} - \mathbf{x}_i^{(k-1)}). \quad (7)$$

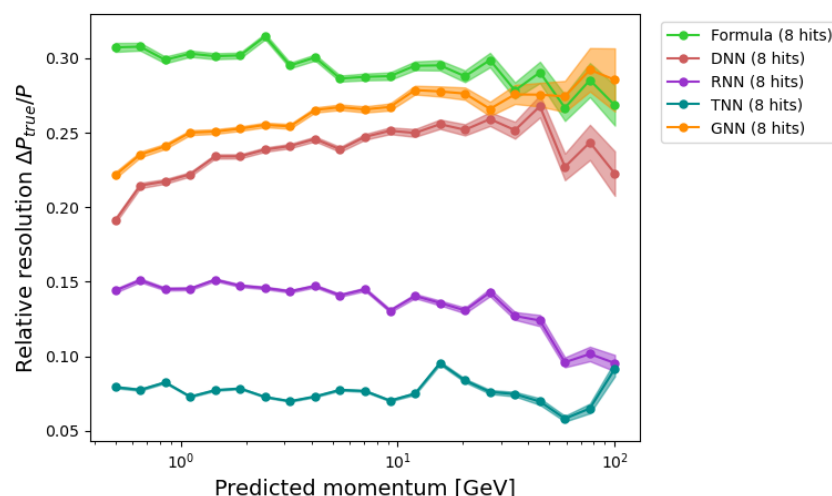
In this particular message passing mode, there is no edge attribute and the DNN  $\phi$  acts on the concatenated tensor of the node tensor and its difference with other connected nodes. For the graphs defined in Figure A2,  $x$  is just the position in the graph, and the mean pooling is used for  $\bigoplus$ .

Similarly to what was performed in Ref. [42], the output of each convolutional layer is concatenated to obtain a tensor with varying levels of abstraction, a method often referred to as jumping knowledge [43]. Residual connections are also included to help the training, as suggested in Ref. [44]. Before being fed to the DNN for the regression, the information from each node needs to go through a final differentiable aggregation operation. To that end,

several options were tried, using single operators (mean, min, max, standard deviation) or a mix of them, concatenated. These operators are permutation-invariant, which, besides the convolution operation, implies that the deviations are treated as unordered, which explains the limited performance shown in Figure 9c. More complex and permutation-variant aggregation operations could potentially improve the regression [45,46].

### 3.4. Model Comparison

The real measure of the regression performance, besides the prediction bias, resides in the prediction resolution. This is illustrated as a function of the momentum in Figure 10 for the different models developed in this section. The sizes and inference times of the different models are in Table 1. The best model, the transformer (TNN), has a relative resolution below 10% and an inference time not far from the DNN thanks to GPU parallelization.



**Figure 10.** The relative resolution of the true momentum as a function of the predicted momentum. The true momentum resolution  $\Delta P$  is estimated from the standard deviation of the true momentum distribution for muons whose predicted momentum falls within a range around the  $x$ -axis marker within 10%. The uncertainty band illustrates the limited number of muons used to calculate the standard deviation.

**Table 1.** A comparison of the size of the different ML models in terms of the number of parameters and memory and inference time per muon (in a batch size of 1024 on a GPU).

| Model | # Params | Size [MB] | Time [ $\mu$ s] |
|-------|----------|-----------|-----------------|
| DNN   | 19 K     | 0.07      | 12              |
| RNN   | 165 K    | 0.63      | 50              |
| TNN   | 420 K    | 1.61      | 17              |
| GNN   | 100 K    | 0.40      | 25              |

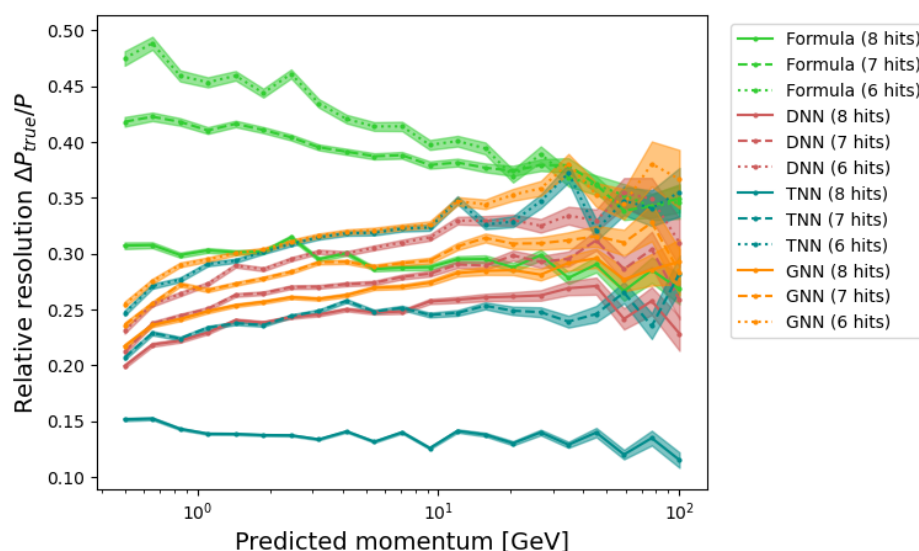
## 4. Real-Life Experiment Considerations

Besides good performance, there are a few other considerations for an ML model to be truly useful in a real-life environment. Section 3 only focused on an ideal detector, assuming all hits were detected and their position measured with infinite precision. In addition, the predicted momentum does not come with any prediction resolution, which would improve muon tomography inference. This section will address how these experimental factors can be measured or improved by re-training the models of Section 3 with the same simulated data, but this time incorporating a more realistic detector response.

#### 4.1. Hit Detection Efficiency

The fact that some hits might not be detected needs to be accounted for in the momentum inference. This can be circumvented by applying a veto on any track that does not leave a hit in every detector layer. However, this would increase the necessary acquisition time to keep the same number of tracks. For example, with eight detector layers and a 95% efficiency—a value in the ballpark of most muon tomography detectors—34% of the tracks would have to be discarded. Additionally, relaxing the trigger selection to events having fewer hits than the total number of panels could increase the overall acceptance, as muon tracks missing the last few panels could now be recorded.

To study the effect of missing hits, several datasets are derived from the one used to train the models in Section 3. For each track, one or several hits are excluded from the linear fit and flagged as missing for the different momentum inference methods. With eight detector planes, it will be assumed that a veto is placed on tracks with less than six hits; therefore, datasets with two, one, and no missing hits will be included in the training. Their performance is in Figure 11.



**Figure 11.** The relative resolution of different models as a function of the predicted momentum. Each ML model has been trained on a combination of datasets with 8 (all hits detected), 7, and 6 hits. In datasets with missed hits, the track fit, computation of angles in Figure 6, and deviations such as in Figures A1 and A2 are performed with detected hits only. When two hits are missed, in around 50% of the cases the formula of Equation (4) cannot be applied as no scattering angle can be measured, and these muons are excluded.

Any hit missed on either side of a scattering material block will make it impossible to measure the scattering angle, but Equations (4) and (5) can still be used with fewer values, albeit with limited precision. However, in some unfortunate cases, no scattering angles can be measured and no momentum can be inferred, which is another limitation of that method.

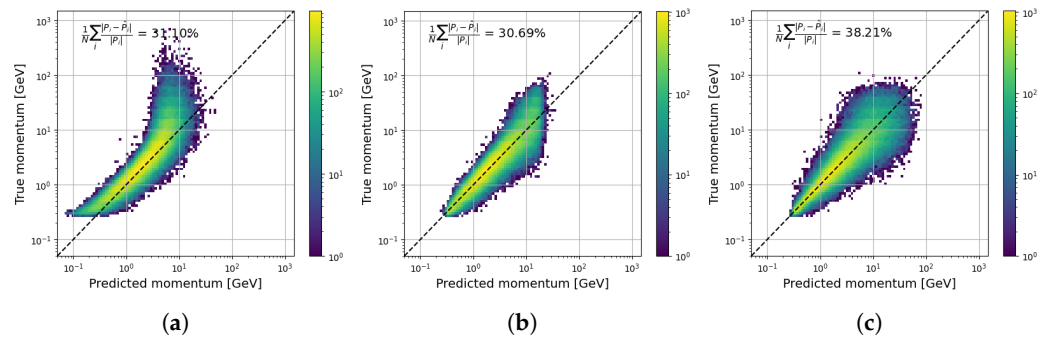
The DNN model is more resilient to missed hits because more deflection angles and residuals can be measured. The main drawback of DNNs is that the number of input features is fixed. When a hit is missed, one alternative is to use a zero value for the features associated with the missed hit; this is called *zero-padding*. An increase in prediction resolution, as observed in Figure 11, is expected from the loss of information, but the mixed training setting seems to be able to mitigate that.

The transformer, as well as the GNN, can handle variable-length inputs and does not necessitate zero-padding. There is, however, a larger degradation in the resolution of the

transformer compared to the DNN, which indicates that the model struggles to handle the change in the deviations' representation. Comparing the case with no missed hit with Figure 10 additionally shows that the mixed training causes confusion. This seems to be linked to the cylindrical representation—and especially the polar angle—on Appendix A that presents large variations when the track parameters obtained from a fit with fewer hits are modified. The GNN, on the other hand, even if less performant overall, suffers less from that effect as it uses a Cartesian representation (Appendix A). The DNN is also less affected because it relies more on the deviation angles.

#### 4.2. Hit Spatial Resolution

To simulate the hit resolution, a Gaussian smearing is applied to the hit position in both the  $x$  and  $y$  directions. The width is chosen as 1 mm. This is a typical, albeit slightly pessimistic, value for resistive plate and drift wire chamber detectors. The features and deviations are extracted from these smeared detector hits, and the different models are then trained on this modified dataset. The regression results are shown in Figure 12. The loss of performance matches the expectation from Figure 7c, as even for a muon with infinite momentum, the smeared hits will emulate deviations of a muon with a momentum of a couple dozen GeVs. While ML models perform better than Figure 12b because they leverage more features, they hit a plateau relatively early in the training, and it is unlikely that the information loss from the training can be recovered. In the transformer case in Figure 12c, if the training is performed for longer, the overall performance decreases, but it is able to predict higher-momentum muons.



**Figure 12.** The result of the regression for a detector with 1 mm spatial resolution using Equation (4) (a), the DNN of Section 3.1 (b), and the transformer of Section 3.2 (c).

#### 4.3. Momentum Resolution

As shown in Figure 2, the density estimation performance can already gain from the momentum knowledge. Supplementing the prediction of our models with a measure of uncertainty could further increase that accuracy, especially in methods using statistical inference.

One option to include the resolution prediction is through the loss function. Instead of producing a single value, the ML model can include predictions for the quantiles of the output distribution. For example, one can add two quantile losses  $\rho$  to the MSE loss, to regress the 25% and 75% quantiles as

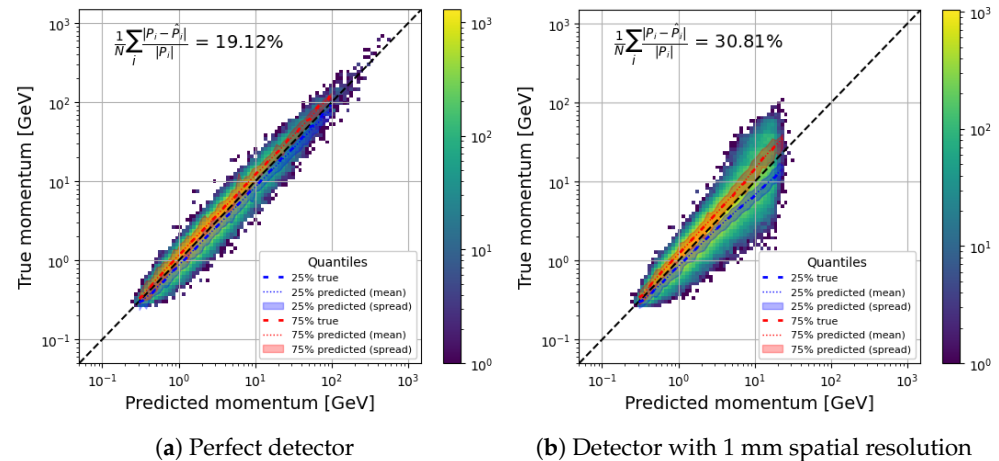
$$\text{Loss}(y, \hat{y}) = \text{MSE}(y, \hat{y}) + \rho_{25\%}(y, \hat{y}_{25\%}) + \rho_{75\%}(y, \hat{y}_{75\%}) \quad (8)$$

$$\rho_{\tau}(y, \hat{y}) = \begin{cases} \tau(y - \hat{y}) & , \quad \hat{y} \leq y, \\ (1 - \tau)(\hat{y} - y) & , \quad \hat{y} > y, \end{cases} \quad (9)$$

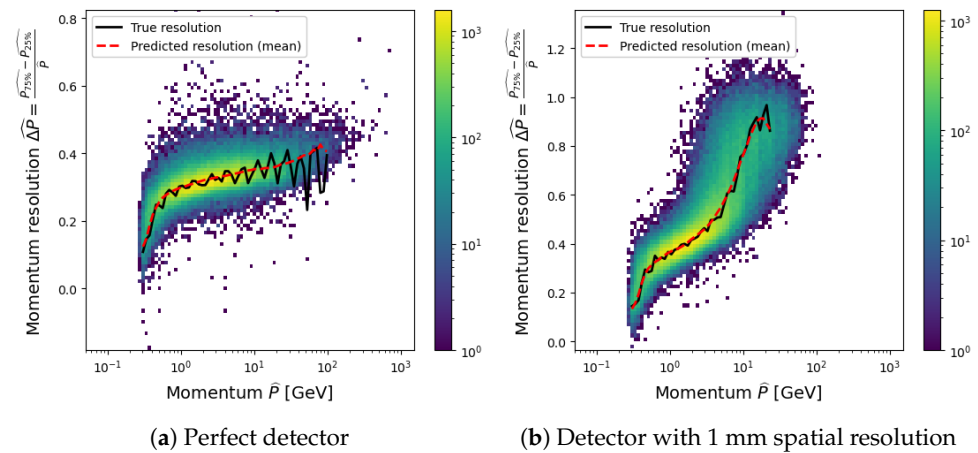
where  $y$  is the prediction target,  $\hat{y}$  is the ML model prediction,  $\tau$  is the quantile value, and  $\hat{y}_{25\%}$  and  $\hat{y}_{75\%}$  are the 25% and 75% quantile predictions, respectively. From  $y = \log_{10} p$ , the corresponding quantiles for the momentum can be obtained, and the resolution is defined as

$$\hat{\Delta P} = \frac{\hat{P}_{75\%} - \hat{P}_{25\%}}{\hat{P}}. \quad (10)$$

The DNN model of Section 3.1 is modified with three output nodes—for  $\hat{y}$ ,  $\hat{y}_{25\%}$ , and  $\hat{y}_{75\%}$ —and the extended loss function of Equation (9). Any of the models discussed in Sections 3.2 and 3.3 could be modified easily, but the larger resolution of the DNN prediction allows a clearer illustration of the method, as shown in Figure 13 for both perfect and imperfect detectors. The resolutions can then be extracted using Equation (10), as shown in Figure 14.



**Figure 13.** Muon momentum regression using the DNN described in Section 3.1 modified for the quantile regression, on two detector setups. The 25% and 75% quantiles on the momentum prediction are shown in blue and red, both the true value (as calculated on the vertical columns for the bin of predicted momentum) and the mean and spread of the quantile outputs (from the same bin).



**Figure 14.** Momentum prediction resolution defined as Equation (10) evaluated from the quantile outputs of the modified DNN, on two detector setups. The true resolution (black) is measured from the regression results of Figure 13, and the mean of the DNN prediction spread is shown for comparison.

These results show that with a very small modification of the training procedure, one can obtain a resolution as an ansatz for a prediction uncertainty. This resolution, however, has to be understood as a distribution property, not as an individual uncertainty for each muon. However, contrary to what was performed in Section 3.4, it can be obtained on a

per-muon basis, not measured on a certain interval of predicted momentum. One possible extension of the method is to get rid of the MSE component and use multiple quantile predictions to obtain some information about the  $f(p|\hat{p})$  posterior distribution, although generative methods for probability density estimation might be more appropriate.

## 5. Outlook

Muon tomography, in both the transmission and scattering modes, is becoming a valuable tool for both scientific and civil purposes. As a non-invasive, large-scale, and passive detection method, it occupies a unique place in the wide range of imaging techniques. One factor limiting muon tomography performance is the dependence on the unknown momentum of the muon.

There are many ways to obtain that information, and in this paper we focused on inverting the scattering problem, using materials with known parameters to infer the momentum from the scatterings. We have compared the classical way of using physics' first principles to several machine learning algorithms. We have shown that they not only improve the performance but are also able to deal more efficiently with experimental constraints of data taking, together with a proposal to augment the reconstruction algorithms with a prediction uncertainty.

To go beyond the stage of the proof of concept presented in this paper, there are several avenues we would like to address. First, the hyperparameters of the advanced algorithms have been manually selected and could be further optimized by a wider automatic search. In particular, the graph neural network presented here is only one model out of a wide range of models proposed in the literature. It might also be worth exploring whether both the sequence and geometry aspects of the deviations could be tackled by a combined model. Second, the use of the deviations from a linear track may have been successful, but it still represents a processing of the raw hit position data that the ML algorithms might not need. There might be a way to relax that step to improve both performance and inference time. Third, the use of the quantile regression can be used to obtain an estimate of the prediction resolution, and by extension a point-wise description of the posterior distribution. This would probably be better tackled by a generative model more suited to probability density estimation. Finally, the work presented in this paper is still far from practical application. Secondary charged particles such as electrons can be produced when muons pass through dense material; the additional hits they would generate in the detector have not been taken into account in our simulations. This effect, together with additional scattering, should also be reflected in more realistic detection planes. However, all these considerations—in simulation, tracking, and rejection algorithms—would need to be tailored for a specific detector technology. Additionally, the detector setup used throughout this paper is not particularly compact nor cheap in terms of material, which might be a limiting factor for its practical use. In order to accommodate the logistical constraints of an in situ deployment, the detector setup has to move toward a simpler configuration, with fewer detector panels and thinner absorbers. The methods developed in this paper should be more thoroughly tested on many different setups with possibly fewer detection layers and a smaller detector volume. And while active experiments can perform their own training on their specific geometry, it would be worth exploring a general-purpose algorithm that could be conditioned on a geometry that might not have been seen during training.

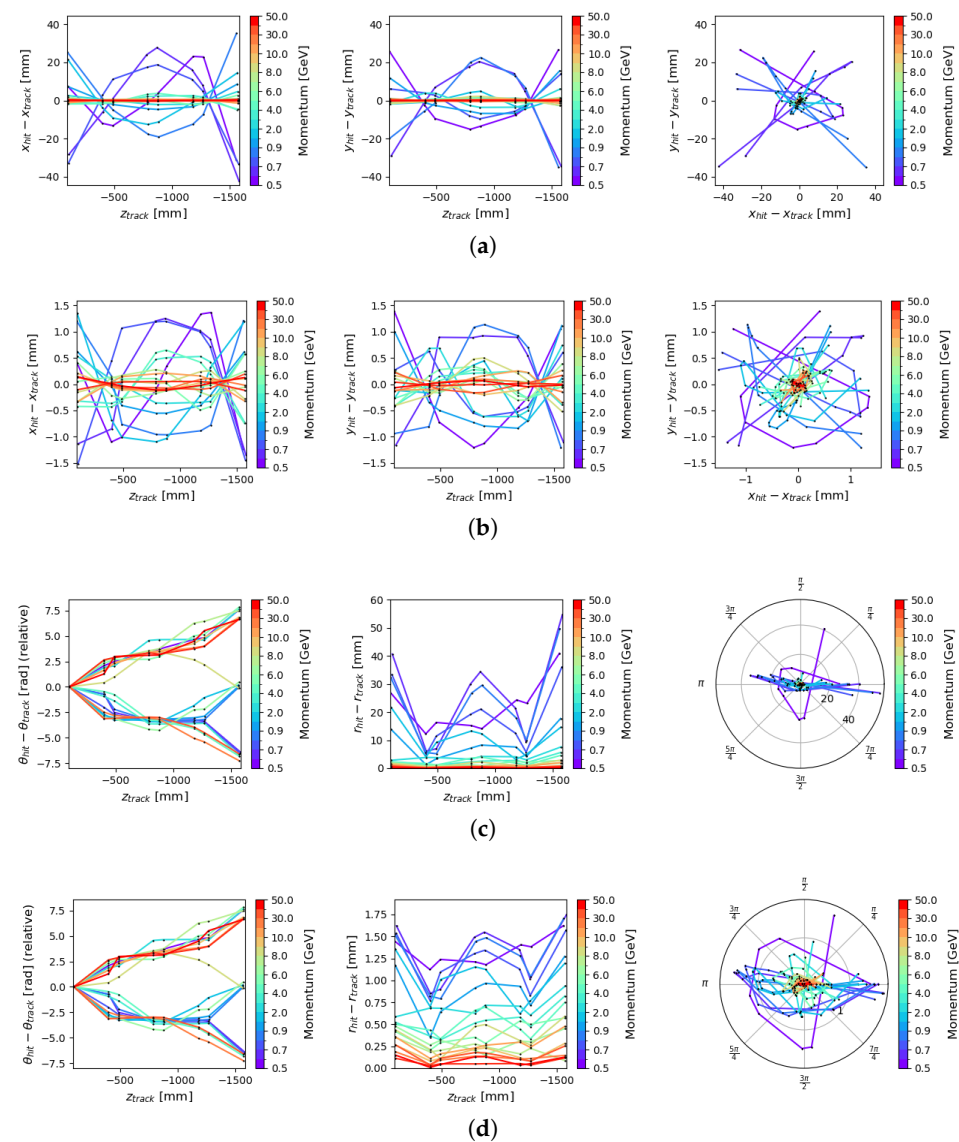
**Author Contributions:** Conceptualization, F.B. and M.L.; methodology, F.B. and M.L.; software, F.B. and M.L.; validation, F.B. and M.L.; data curation, F.B. and M.L.; writing—original draft preparation, F.B. and M.L.; writing—review and editing, F.B. and M.L.; visualization, F.B. and M.L.; funding acquisition, F.B. and M.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors acknowledge funding from the EU Horizon 2020 Research and Innovation Programme under grant agreement no. 101021812 (“SilentBorder”), by the Fonds de la Recherche Scientifique—FNRS under Grants No. T.0099.19 and J.0070.21 and the Science and Technology Facilities Council (STFC). Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

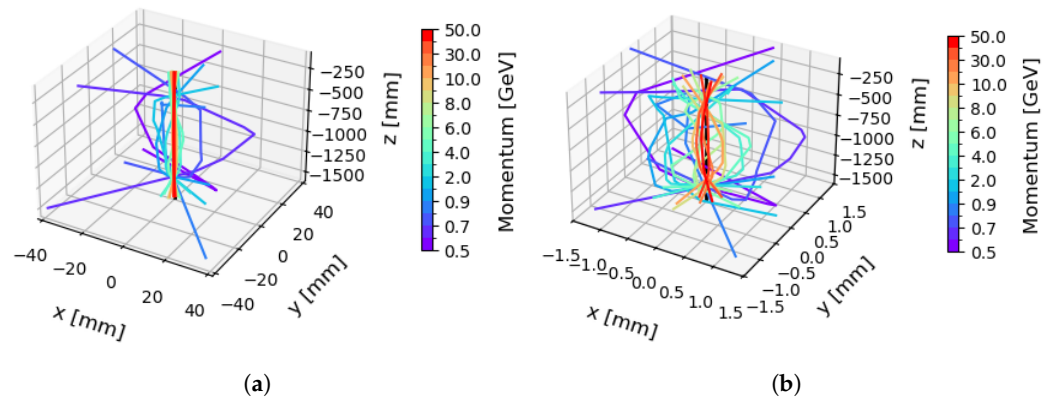
**Data Availability Statement:** The datasets presented in this article are not readily available because the data are part of an ongoing study. Requests to access the datasets should be directed to the authors.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A. Deviations’ Bases



**Figure A1.** The deviations of the muons’ hits from a fitted linear track expressed in different bases: Cartesian (a), Cartesian with the log-modulus function applied to both  $x$  and  $y$  (b), cylindrical (c), and cylindrical with log-modulus applied to the radius  $r$  while keeping the polar angle  $\theta$  constant (d).



**Figure A2.** The deviations of the muons' hits from the fitted linear track, defined as a graph along the track axis (black) using different scales: linear (a) and with log-modulus applied on the magnitude while keeping the same angle for each point (b).

## Appendix B. Network Hyperparameters

**Table A1.** DNN hyperparameters.

| Parameter         | Value |
|-------------------|-------|
| Number of layers  | 6     |
| Neurons per layer | 64    |
| Hidden activation | GELU  |
| Output activation | None  |
| Batchnorm         | True  |

**Table A2.** RNN hyperparameters.

| Parameter                    | Value |
|------------------------------|-------|
| <b>Embedding</b>             |       |
| Normalize inputs (batchnorm) | True  |
| Neurons for each layer       | 32,64 |
| Activation                   | ReLU  |
| <b>Recurrent cell</b>        |       |
| Cell type                    | GRU   |
| Dimension                    | 64    |
| Layers                       | 6     |
| Bidirectional                | False |
| <b>Output (DNN)</b>          |       |
| Layers                       | 3     |
| Neurons per layer            | 64    |
| Hidden activation            | ReLU  |
| Output activation            | None  |

**Table A3.** Transformer hyperparameters.

| Parameter                          | Value  |
|------------------------------------|--------|
| <b>Embedding/position encoding</b> |        |
| Normalize inputs (batchnorm)       | True   |
| Neurons for each layer             | 32, 64 |
| Activation                         | GELU   |
| <b>Encoder</b>                     |        |
| Dimension                          | 64     |
| Encoder layers/class layers        | 6/2    |
| Heads                              | 8      |

**Table A3.** *Cont.*

| Parameter              | Value |
|------------------------|-------|
| Feed-forward dimension | 128   |
| Layer norm             | True  |
| <b>Output (DNN)</b>    |       |
| Layers                 | 3     |
| Neurons per layer      | 64    |
| Hidden activation      | GELU  |
| Output activation      | None  |

**Table A4.** GNN hyperparameters.

| Parameter                 | Value         |
|---------------------------|---------------|
| <b>Graph blocks</b>       |               |
| Block type                | EdgeConv      |
| Feature dimension         | 64            |
| Number of blocks          | 6             |
| DNN feedforward dimension | 128           |
| DNN batchnorm             | True          |
| Graphnorm                 | True          |
| Activation                | GELU          |
| Node aggregation          | Mean          |
| <b>Graph pooling</b>      |               |
| Aggregation               | Mean          |
| Jumping knowledge         | Concatenation |
| <b>Output (DNN)</b>       |               |
| Layers                    | 3             |
| Neurons per layer         | 64            |
| Hidden activation         | GELU          |
| Output activation         | None          |

## References

- Oláh, L.; Tanaka, H.K.M.; Varga, D. (Eds.) *Muography: Exploring Earth's Subsurface with Elementary Particles*; American Geophysical Union: Washington, DC, USA, 2022. [\[CrossRef\]](#)
- Poulson, D.; Durham, J.; Guardincerri, E.; Morris, C.; Bacon, J.; Plaud-Ramos, K.; Morley, D.; Hecht, A. Cosmic ray muon computed tomography of spent nuclear fuel in dry storage casks. *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **2017**, *842*, 48–53. [\[CrossRef\]](#)
- Morishima, K.; Kuno, M.; Nishio, A.E.A. Discovery of a big void in Khufu's Pyramid by observation of cosmic-ray muons. *Nature* **2017**, *552*, 386–390. [\[CrossRef\]](#) [\[PubMed\]](#)
- Niederleithinger, E.; Gardner, S.; Kind, T.; Kaiser, R.; Grunwald, M.; Yang, G.; Redmer, B.; Waske, A.; Mielentz, F.; Effner, U.; et al. Muon tomography of a reinforced concrete block—First experimental proof of concept. *arXiv* **2020**, arXiv:2008.07251. [\[CrossRef\]](#)
- Barnes, S.; Georgadze, A.; Giammanco, A.; Kiisk, M.; Kudryavtsev, V.A.; Lagrange, M.; Pinto, O.L. Cosmic-Ray Tomography for Border Security. *Instruments* **2023**, *7*, 13. [\[CrossRef\]](#)
- Gaisser, T. *Cosmic Rays and Particle Physics*; Cambridge University Press: Cambridge, UK, 1990; Chapter Muons; p. 69.
- Hagmann, C.; Lange, D.; Wright, D. Cosmic-ray shower generator (CRY) for Monte Carlo transport codes. In Proceedings of the 2007 IEEE Nuclear Science Symposium Conference Record, Honolulu, HI, USA, 26 October–3 November 2007; Volume 2, pp. 1143–1146.
- Lynch, G.R.; Dahl, O.I. Approximations to multiple Coulomb scattering. *Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. Mater. Atoms* **1991**, *58*, 6–10. [\[CrossRef\]](#)
- Navas, S.; Amsler, C.; Gutsche, T.; Hanhart, C.; Hernández-Rey, J.J.; Lourenço, C.; Masoni, A.; Mikhasenko, M.; Mitchell, R.E.; Patrignani, C.; et al. Review of particle physics. *Phys. Rev. D* **2024**, *110*, 030001. [\[CrossRef\]](#)
- Stapleton, M.; Burns, J.; Quillin, S.; Steer, C. Angle Statistics Reconstruction: A robust reconstruction algorithm for Muon Scattering Tomography. *J. Instrum.* **2014**, *9*, P11019. [\[CrossRef\]](#)
- Thomay, C.; Velthuis, J.J.; Baesso, P.; Cussans, D.; Morris, P.A.W.; Steer, C.; Burns, J.; Quillin, S.; Stapleton, M. A binned clustering algorithm to detect high-Z material using cosmic muons. *J. Instrum.* **2013**, *8*, P10013. [\[CrossRef\]](#)

12. Bae, J.; Montgomery, R.; Chatzidakis, S. Image reconstruction algorithm for momentum dependent muon scattering tomography. *Nuclear Eng. Technol.* **2024**, *56*, 1553–1561. [[CrossRef](#)]
13. Bae, J.; Montgomery, R.; Chatzidakis, S. Momentum informed muon scattering tomography for monitoring spent nuclear fuels in dry storage cask. *Sci. Rep.* **2024**, *14*, 6717. [[CrossRef](#)]
14. Bae, J.; Montgomery, R.; Chatzidakis, S. A New Momentum-Integrated Muon Tomography Imaging Algorithm. *arXiv* **2023**, arXiv:2304.14427.
15. Agostinelli, S.; Allison, J.; Amako, K.A.; Apostolakis, J.; Araujo, H.; Arce, P.; Asai, M.; Axen, D.; Banerjee, S.; Barrand, G.J.N.I.; et al. GEANT4—A simulation toolkit. *Nucl. Inst. Meth. A* **2003**, *506*, 250. [[CrossRef](#)]
16. Ambrosino, F.; Anastasio, A.; Bross, A.; Béné, S.; Boivin, P.; Bonechi, L.; Cârloganu, C.; Ciaranfi, R.; Cimmino, L.; Combaret, C.; et al. Joint measurement of the atmospheric muon flux through the Puy de Dôme volcano with plastic scintillators and Resistive Plate Chambers detectors. *J. Geophys. Res. Solid Earth* **2015**, *120*, 7290–7307. [[CrossRef](#)]
17. Giammanco, A.; Al Moussawi, M.; Hong, Y.; Ambrosino, F.; Anastasio, A.; Basnet, S.; Bonechi, L.; Bongi, M.; Borselli, D.; Bross, A.; et al. First Results, and Experimental Status of the MURAVES Experiment 2024. *J. Adv. Instrum. Sci.* **2024**, *2024*, 1–11. [[CrossRef](#)]
18. Gibert, D.; de Bremond d’Ars, J.; Carlus, B.; Deroussi, S.; Ianigro, J.C.; Jessop, D.E.; Jourde, K.; Kergosien, B.; Le Gonidec, Y.; Lesparre, N.; et al. Observation of the Dynamics of Hydrothermal Activity in La Soufrière of Guadeloupe Volcano with Joint Muography, Gravimetry, Electrical Resistivity Tomography, Seismic and Temperature Monitoring. In *Muography*; American Geophysical Union (AGU): Washington, DC, USA, 2022; Chapter 5, pp. 55–73. [[CrossRef](#)]
19. Oláh, L.; Tanaka, H.; Suenaga, H.; Miyamoto, S.; Galgóczi, G.; Hamar, G.; Varga, D. Development of Machine Learning-Assisted Spectra Analyzer for the NEWCUT Muon Spectrometer. *J. Adv. Instrum. Sci.* **2022**, *2022*. [[CrossRef](#)]
20. Lo Presti, D.; Gallo, G.; Bonanno, D.L.; Bonanno, G.; Ferlito, C.; La Rocca, P.; Reito, S.; Riggi, F.; Romeo, G. Three Years of Muography at Mount Etna, Italy. In *Muography*; American Geophysical Union (AGU): Washington, DC, USA, 2022; Chapter 7, pp. 93–108. [[CrossRef](#)]
21. Bae, J.; Chatzidakis, S. Momentum-Dependent Cosmic Ray Muon Computed Tomography Using a Fieldable Muon Spectrometer. *Energies* **2022**, *15*, 2666. [[CrossRef](#)]
22. Chen, J.; Li, H.; Li, Y.; Liu, P. Towards a muon scattering tomography system for both low-Z and high-Z materials. *J. Instrum.* **2023**, *18*, P08008. [[CrossRef](#)]
23. Vanini, S.; Calvini, P.; Checchia, P.; Garola, A.R.; Klinger, J.A.; Zumerle, G.; Bonomi, G.; Donzella, A.; Zenoni, A. Muography of different structures using muon scattering and absorption algorithms. *Philos. Trans. R. Soc. A* **2018**, *377*, 20180051. [[CrossRef](#)]
24. Rand, E.T.; Kamaev, O.; Valente, A.; Bhullar, A. Nonparametric Dense-Object Detection Algorithm for Applications of Cosmic-Ray Muon Tomography. *Phys. Rev. Appl.* **2020**, *14*, 064032. [[CrossRef](#)]
25. Anghel, V.; Armitage, J.; Baig, F.; Boniface, K.; Boudjemline, K.; Bueno, J.; Charles, E.; Drouin, P.L.; Erlandson, A.; Gallant, G.; et al. A plastic scintillator-based muon tomography system with an integrated muon spectrometer. *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.* **2015**, *798*, 12–23. [[CrossRef](#)]
26. Ankowski, A.; Antonello, M.; Aprili, P.; Arneodo, F.; Badertscher, A.; Baiboussinov, B.; Baldo Ceolin, M.; Battistoni, G.; Benetti, P.; Borio di Tigliole, A.; et al. Measurement of through-going particle momentum by means of multiple scattering with the ICARUS T600 TPC. *Eur. Phys. J. C* **2006**, *48*, 667–676. [[CrossRef](#)]
27. Mondal, S.; Mastrolorenzo, L. Machine learning in high energy physics: A review of heavy-flavor jet tagging at the LHC. *Eur. Phys. J. ST* **2024**, *233*, 2657–2686. [[CrossRef](#)]
28. Kasieczka, G.; Plehn, T.; Russell, M.; Schell, T. Deep-learning Top Taggers or The End of QCD? *J. High Energy Phys.* **2017**, *5*, 006. [[CrossRef](#)]
29. The ATLAS Collaboration. *Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment*; ATLAS collaboration; Technical Report; CERN: Geneva, Switzerland, 2017; ATL-PHYS-PUB-2017-003
30. Bols, E.; Kieseler, J.; Verzetti, M.; Stoye, M.; Stakia, A. Jet Flavour Classification Using DeepJet. *J. Instrum.* **2020**, *15*, P12012. [[CrossRef](#)]
31. Cheng, T. Recursive Neural Networks in Quark/Gluon Tagging. *Comput. Softw. Big Sci.* **2018**, *2*, 3. [[CrossRef](#)]
32. Qu, H.; Gouskos, L. ParticleNet: Jet Tagging via Particle Clouds. *Phys. Rev. D* **2020**, *101*, 056019. [[CrossRef](#)]
33. Qu, H.; Li, C.; Qian, S. Particle Transformer for Jet Tagging. *arXiv* **2022**, arXiv:2202.03772.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. [[CrossRef](#)]
35. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Sydney, Australia, 2019; pp. 8024–8035.
36. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. In *Proceedings of the ICLR Workshop on Representation Learning on Graphs and Manifolds*, New Orleans, LA, USA, 6–9 May 2019.

37. Falcon, W.; The PyTorch Lightning Team. PyTorch Lightning. Zenodo. 2019. Available online: <https://zenodo.org/records/3828935> (accessed on 28 March 2025).
38. Liu, L.; Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; Han, J. On the Variance of the Adaptive Learning Rate and Beyond. *arXiv* **2019**, arXiv:1908.03265. [[CrossRef](#)]
39. Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; Jégou, H. Going deeper with Image Transformers. *arXiv* **2021**, arXiv:2103.17239. [[CrossRef](#)]
40. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. *arXiv* **2015**, arXiv:1505.00880. [[CrossRef](#)]
41. Tumasyan, A.; Adam, W.; Andrejkovic, J.W.; Bergauer, T.; Chatterjee, S.; Dragicevic, M.; Escalante Del Valle, A.; Frühwirth, R.; Jeitler, M.; Krammer, N.; et al. Identification of hadronic tau lepton decays using a deep neural network. *J. Instrum.* **2022**, 17, P07023. [[CrossRef](#)]
42. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *arXiv* **2018**, arXiv:1801.07829. [[CrossRef](#)]
43. Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.I.; Jegelka, S. Representation Learning on Graphs with Jumping Knowledge Networks. *arXiv* **2018**, arXiv:1806.03536. [[CrossRef](#)]
44. Li, G.; Müller, M.; Thabet, A.; Ghanem, B. DeepGCNs: Can GCNs Go as Deep as CNNs? *arXiv* **2019**, arXiv:1904.03751. [[CrossRef](#)]
45. Buterez, D.; Janet, J.P.; Kiddle, S.J.; Oglic, D.; Liò, P. Graph Neural Networks with Adaptive Readouts. *arXiv* **2022**, arXiv:2211.04952. [[CrossRef](#)]
46. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *arXiv* **2017**, arXiv:1706.02216. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.