

MIŁOSZ ZDYBAŁ  
MARCIN KUCHARCZYK  
MARCIN WOLTER

## MACHINE LEARNING BASED EVENT RECONSTRUCTION FOR THE MUonE EXPERIMENT

**Abstract** *A proof-of-concept solution based on the machine learning techniques has been implemented and tested within the MUonE experiment designed to search for New Physics in the sector of anomalous magnetic moment of a muon. The results of the DNN based algorithm are comparable to the classical reconstruction, reducing enormously the execution time for the pattern recognition phase. The present implementation meets the conditions of classical reconstruction, providing an advantageous basis for further studies.*

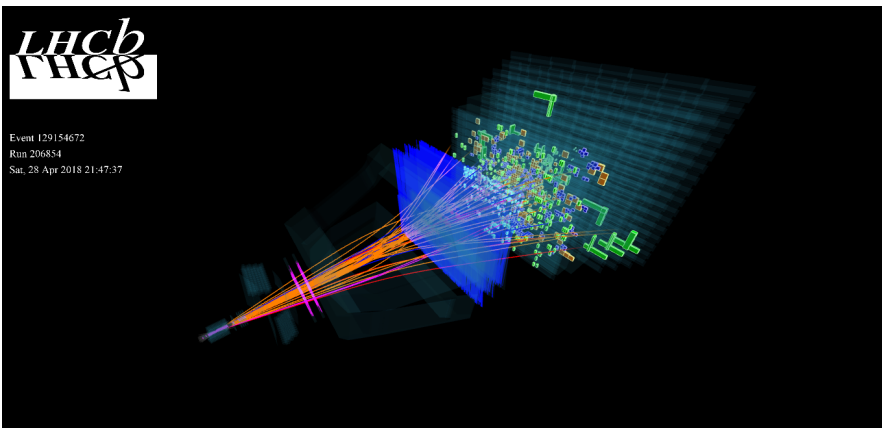
**Keywords** machine learning, artificial neural networks, track reconstruction, high energy physics

**Citation** Computer Science 25(1) 2024: 147–167

**Copyright** © 2024 Author(s). This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

## 1. Introduction

Significant developments have been applied during the last decades in the field of High Energy Physics (HEP) experiments, including computing technologies. Searches for New Physics phenomena, being an expansion of the so-called Standard Model, i.e. current incomplete theoretical knowledge about the basic behavior of the fundamental constituents of nature and the interactions between them, lead to experimental studies carried out at ever increasing energies. The number of particles created by the interaction of two particles (collision event) is generally increasing with the collision energy. As a consequence a huge number of charged particles have to be reconstructed (e.g. in proton-proton collisions), resulting in much more complex event patterns. A typical event in proton-proton collision showing the tracks of multiple particles passing through the detector is presented in Figure 1, where the particles leave energy deposits (hits) in consecutive detector layers, being a basis for further track reconstruction. In order to enable the search for rare interesting collision events immersed in a huge background of events exhibiting well-known physics, the data rates related to the detector luminosity<sup>1</sup> have increased enormously (e.g., 40 MHz readout rate in LHC). It has to be reduced online by more than five orders of magnitude before the information from an event is written on mass storage for further analysis.



**Figure 1.** Example of an event in High Energy Physics experiment, showing tracks of multiple particles passing through the detector [31]

This paper aims to review the machine learning based approach applied in crucial stages of the data analysis process in HEP experiments, i.e. the procedure to determine basic kinematic parameters of charged particles at their point of production and the procedure to establish the location of these production points. They are

---

<sup>1</sup>Luminosity translates to the number of collisions per second and it is related to track density.

commonly called track and vertex reconstruction. High density of tracks in a single collision event (detector occupancy) in operating and planned high-energy physics experiments results in a large combinatorics of hits in the event pattern recognition. Therefore, a novel machine learning based event reconstruction algorithms have been developed and tested within a framework of the MUonE experiment [2] in order to maximize the statistical power of the final physics measurement. The results of the DNN based algorithm are comparable to the classical reconstruction, allowing not only to reduce execution time of the pattern recognition phase, but also to improve the precision and efficiency of the track and vertex reconstruction.

## 2. Particle track reconstruction in High Energy Physics experiments

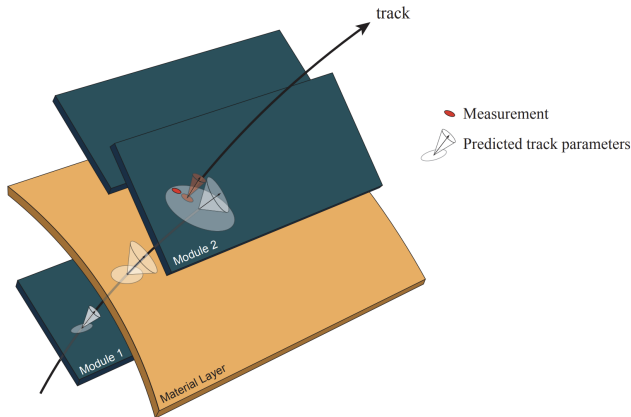
### 2.1. State of the art

In High Energy Physics experiments the reconstruction of charged particle trajectories in the detectors is the most crucial process as it constitutes the major part of reconstruction time of the whole event. Tracking algorithms partition a collection of position measurements into groups corresponding to the hits originating from the same particle traversing through the detector. In the next step the parametrized trajectories are fitted to these collections to extract particle kinematics and locations of interaction vertices. The obtained results are later combined with measurements from other detector systems, like calorimeters measuring the particle energy, to construct a complete physical model of an event. In the last stage of the data analysis the reconstructed events are used to extract the physical quantities.

Traditional tracking algorithms have been used with great success in the HEP experiments. However they suffer from serious limitations that motivate for searching new solutions. These algorithms are inherently serial, and scale poorly with detector occupancy. In particle physics the measurement of charged particle parameters is one of the most computationally-intensive processes. This process relies on measurements of particle tracking detectors to construct a particle trajectory by combining the detected hits and resolving the particle momentum via fitting the trajectory points using the Kalman filter [24]. The Kalman filter processes a set of discrete measurements to determine the internal state of a linear dynamical system (see Fig. 2). Both the measurements and the system can be subjected to independent random perturbations or noise. By combining predictions based on the previous state estimates with subsequent measurements, the impact of these perturbations on the following state estimates can be minimized.

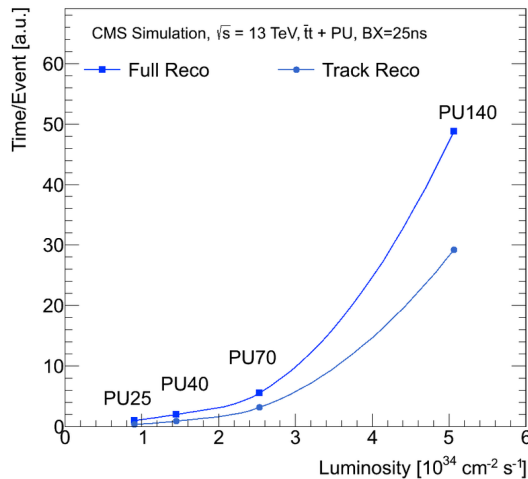
In high luminosity experiments, where multiple particles are produced as a result of an interaction, the process of isolating detector hits for each particle trajectory relies on considering each combination of hits that can potentially form a track and then fitting each hypothesis to determine which one represents a valid trajectory. Also the noise, always present in particle tracking detectors and resulting in additional hits

in the detectors, increases the number of possible combinations. This process can be time-consuming, amounting to the most significant part of the total data post-processing time. For such methods based on the Kalman filter the CPU needed for track reconstruction grows rapidly with the luminosity (see Fig. 3). Therefore, more advanced methods of finding particle trajectories using the measurements from all active detector elements can be investigated.



**Figure 2.** Simplified illustration of a typical extrapolation process within a Kalman filter.

The track representation on the detector module 1 is propagated onto the next measurement surface, which results in the track prediction on module 2 [37]



**Figure 3.** Expected CPU time per event as a function of instantaneous luminosity collected by the CMS experiment, for both full reconstruction and the dominant tracking part. The pile-up (PU) is the number of interactions per beam crossing. PU25 corresponds to the data taken in 2012, and PU140 corresponds to the HL-LHC era. The CPU time of the reconstruction is dominated by the track reconstruction [12]

## 2.2. Machine learning based track reconstruction

Machine learning methods such as deep neural networks have some promising characteristics that could prove effective for particle tracking. Neural networks are known to be very good at finding patterns and modeling non-linear dependencies in data. They also involve highly regular computation that can run effectively on parallel architectures such as GPUs (Graphics Processing Units).

Neural Networks are therefore widely used in High Energy Physics not only as a classification tool, but also for other tasks like intelligent data reduction and time series analysis [27] or reconstruction of a pulse shape from the front-end electronics [26]. They are used as well to optimize processes in various environments (Reinforcement Learning), for example automate the management of resources in a computing cloud [20].

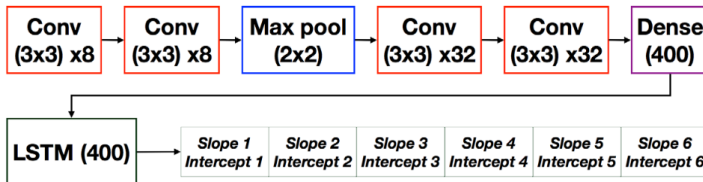
The first approach to use neural networks for particle track reconstruction was done already in the 1980s [34]. However modern techniques based on deep learning have started to be studied in the last years. Two categories of machine learning solutions, image-based and point-based models were investigated [17].

The Convolutional Neural Networks (CNNs) (see for example [13] for description of various types of neural networks) proved to be a very efficient tool in image recognition. As such they are widely used in physics research, one of the examples is the CREDO experiment offline trigger using CNN to tag artefacts appearing in the CREDO database [35].

The computer vision techniques based on CNN such as semantic segmentation and image captioning have inspired the image-based models of particle track recognition. In this approach the detector data is treated as an image and the convolutional and also recurrent neural networks are applied to detect tracks.

### Image based track reconstruction

In the image-based approach multiple track finding problem might be treated in a similar fashion to the image captioning, where the descriptions of the tracks (i.e., track parameters) are analogous to the text captions assigned to the various patterns seen in the image [39]. For this purpose, the long/short-term memory (LSTM) [21] layer is used.



**Figure 4.** Convolutional deep neural network with convolutional layers followed by dense and LSTM layers. Network is trained to reconstruct track parameters for multiple track events [29]

In the case of track reconstruction, the Convolutional Neural Network (CNN) sequentially reconstructs consecutive tracks, which are the sequential input for the LSTM layer. Example of such a network suited to reconstruct events with multiple tracks is shown in Figure 4.

The image-based models map nicely onto well-studied problems in computer vision and sequence modeling. However, when scaling up to the realistic complexity of particle physics experiments they are suffering from high dimensionality and sparsity.

### **Point based track reconstruction**

In the point-based models, the continuously distributed spacepoint hits are used. They are structured in a list or tree for learning how to group them into track candidates. A recurrent neural network acts as an iterative filter similar to a Kalman Filter. Therefore the model predicts the position of the point on the next detector layer. It can be used to build tracks by selecting the closest spacepoint to the prediction at every layer and searches until a complete track is found. This architecture might use an LSTM layer and fully connected layer with linear activation function to reconstruct multiple tracks.

### **Graph Neural Network models**

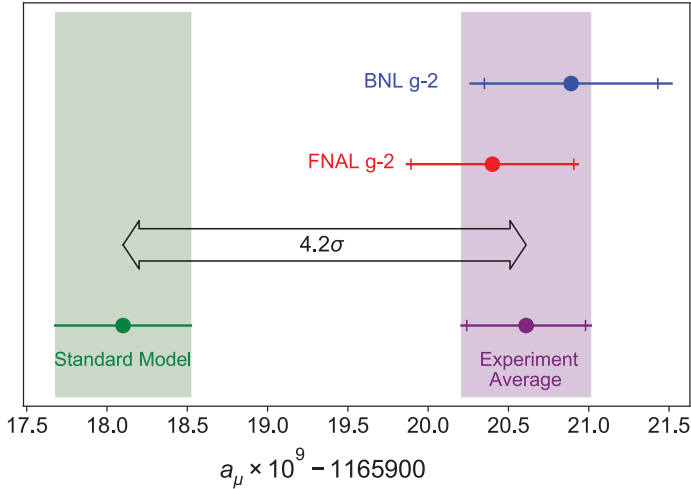
In 2020 the Exa.TrkX project [22] has demonstrated the applicability of Geometric Deep Learning (GDL) methods to particle tracking [14] (specifically Graph Neural Networks (GNNs) [38]). GNNs have already proven to be successful in computer vision applications [28]. Such a network is concerned with learning representations of data that have complex geometrical relationships and no natural ordering, which corresponds well with hits in the detector. In addition such models are naturally parallel and therefore well-suited to run on hardware accelerators and GPUs. The training of such a network might be computational demanding, but an answer of a trained network is fast and the computer time needed increases linearly with the number of tracks.

In applications to track finding graphs are constructed from the cloud of hits in each event. Edges are drawn between hits that may come from the same particle track according to some loose heuristic criteria. The GNN model is then trained to classify the graph edges as real or fake, giving a pure and efficient sample of track segments which can be used to construct full track candidates. Advanced studies concerning the application of GNNs for track reconstructions were presented by both CMS [9] and ATLAS [11] experiments.

It was shown [23], that within the simplifying assumptions, the GNN based track finding algorithm can meet the tracking performance requirements of current, high luminosity collider experiments. This performance should be robust against systematic effects like detector noise, misalignment, and pile-up. The GNN based algorithms are promising and growing in popularity.

### 3. MUonE experiment

A very promising opportunity to search for New Physics in the sector of muon's anomalous magnetic moment  $a_\mu$  has appeared with the new results from  $g - 2$  experiments [6, 8], which measured the anomaly with respect to the Standard Model prediction at the level of 4.2 standard deviations (see Fig. 5).



**Figure 5.** Comparison of the measurements of anomalous muon magnetic moment  $a_\mu$  with the Standard Model prediction [6], where the discrepancy of 4.2 standard deviation between theory and experiment can be observed. In order to improve the visibility of the discrepancy the unit on the  $x$ -axis corresponds to a given  $a_\mu$  value multiplied by  $10^9$  and subtracted with 1165900

As the main limitation of eventual discovery comes from the precision of the theoretical Standard Model predictions, dominated by the uncertainty related to the hadronic interactions, the idea is to use the process of elastic muon scattering on electrons for the precise estimation of the hadronic contribution to  $a_\mu$ . The experiment dedicated to measure precisely such a hadronic contribution is the MUonE project [2], designed to determine the hadronic part of the running of the electromagnetic coupling constant in the space-like region by the scattering of high-energy muons on atomic electrons in a low- $Z$  target through the elastic process  $\mu e \rightarrow \mu e$  [4]. A result with significantly suppressed statistical uncertainty can be achieved on the hadronic contribution to  $a_\mu$ , which, together with the results from running [6] or planned [5]  $g - 2$  experiments supposed to measure directly  $a_\mu$ , would increase the significance of observed discrepancy up to the level of 7 standard deviations. In order to measure the hadronic contribution with a required accuracy, a significant boost in precision and event statistics is necessary. This can only be achieved by accurate performance of the both the trigger and tracking system of the MUonE experiment.

### 3.1. Experimental setup

The data samples of  $\mu e \rightarrow \mu e$  elastic scattering will be collected in MUonE experiment using 150–160 GeV muons impinging on the atomic electrons of Beryllium targets. The upgraded M2 muon beam at the CERN SPS [15] will be used for this purpose, delivering high energy and high-intensity muon and hadron beams, and also low intensity electron beams for calibration. The beams are conducted in the following way. First, the SPS primary proton beam of 450 GeV impinges on a primary Beryllium production target, where mainly secondary protons, electrons, pions and kaons are produced. In the next step the secondary particles are transported in a beam line allowing the pions and kaons to decay into muons. At this stage a 9.9 m thick Beryllium absorber stops the left-over hadrons, allowing the muons at the same time to pass basically unharmed. Next, such muons are momentum-selected employing large magnetic collimators, and finally the muons with momenta in the range of 100 and 225 GeV/c are selected. The typical maximal intensity for a beam energy of 160 GeV is  $5 \cdot 10^7 \mu/\text{sec}$ .

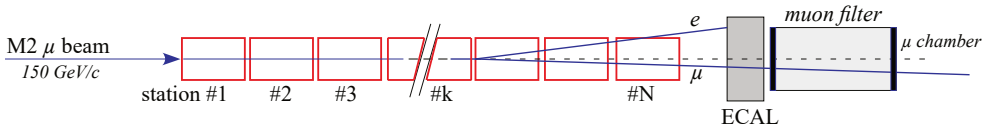


Figure 6. Schematic view of the MUonE experimental apparatus [2]

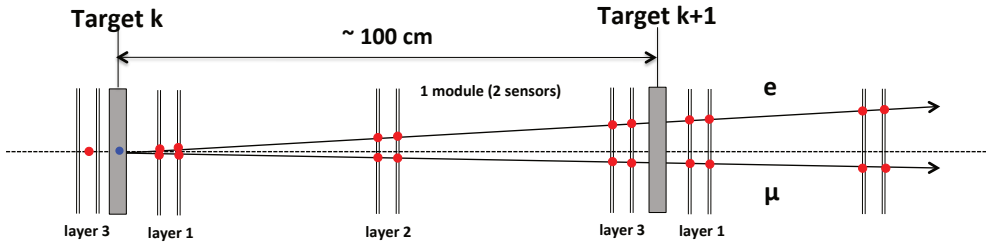


Figure 7. Schematic view of a single tracking station [2]

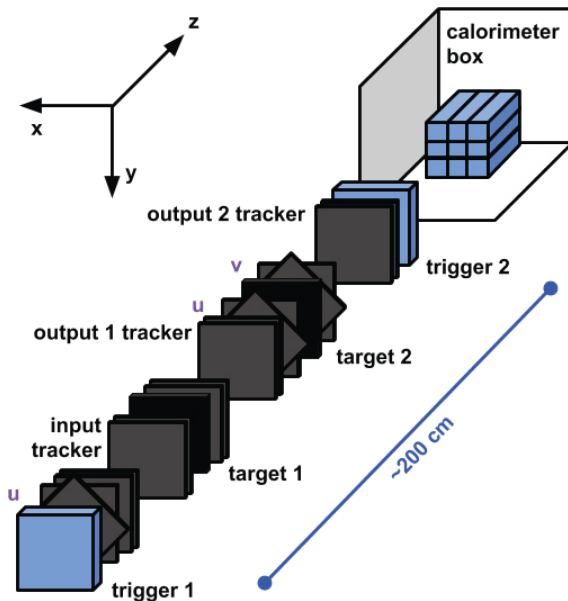
The main detectors of the MUonE experiment [2] are specified in Figure 6 and 7. The tracking system will provide the precise measurement of the scattering angles of the outgoing electron and muon, with respect to the direction of the incoming muon beam. It will contain 40 identical stations (see Fig. 6), each consisting of a 3 cm thick layer of Beryllium coupled to 3 Si layers (see Fig. 7) located at a relative distance of about one meter from each other and spaced by intermediate air gaps. Such an arrangement provides both a distributed target with low- $Z$  and the tracking system. The silicon strip sensors for the MUonE project are characterized by a large active area sufficient to cover the full MUonE required acceptance, together with appropriate



spacial resolution. They can also support the high readout rate of 40 MHz required for MUonE with their accompanying front-end electronics. The downstream particle identifiers are planned to be installed, required to solve the muon-electron ambiguity. That will be a calorimeter for the electrons and a muon filter for the muons. A homogeneous electromagnetic calorimeter placed downstream all the tracker stations will be used, in order to accomplish the physical requirements, i.e. particle identification, measurement of the electron energy and event selection.

### 3.2. MUonE test beam in 2018

In 2018 the MUonE test run was performed with the aim to provide information for the design of the final MUonE detector setup [3]. The apparatus used was situated in the EHN2 experimental area, located behind the COMPASS spectrometer. The 187 GeV positive muon beam was obtained from decays of pions, which were stopped in the beam dump in the posterior part of the spectrometer. A  $10 \times 10 \text{ cm}^2$ , 8 mm thick graphite target was followed by the tracking system with 16 microstrip layers consisted of a  $9.293 \times 9.293 \times 0.041 \text{ cm}^3$  single-side sensor with 384 channels (see Fig. 8). Each tracking layer measured one hit coordinate,  $x$  or  $y$ . Despite that, stereo stations rotated by an angle  $\pm\pi/4$  were added. A calorimeter located at the end of the system, composed from BGO tapered crystals, covered an angular acceptance of about 15 mrad on each side from the center of Si layers.



**Figure 8.** Schematic view of the apparatus used in the MUonE 2018 test run [3]

From the data collected at the last period of the 6 months run, after the final requirements on the presence of an incoming track and at least two outgoing tracks, the number of events used in the analysis was reduced to  $\sim 94 \cdot 10^3$ . The event reconstruction consisted of the following main steps: the pattern recognition, 2-dimensional track finding, combining 2-dimensional track candidates into a 3-dimensional track and finally, constructing a scattering event from three 3-dimensional tracks with a dedicated kinematic vertex fit based on a constrained least square method. Finally, this allowed to obtain a clean sample of  $\mu e$  elastic scattering events.

The angular resolution was determined with the simulation, which met the MUonE 2018 test beam configuration. A sample of  $\sim 100 \cdot 10^3$  events was produced, where the incoming muon beam was assumed to be a monoenergetic beam with energy of 187 GeV, and  $x$  and  $y$  distributions were adjusted to match the ones measured with data.

## 4. Machine learning based track reconstruction for MUonE

The measurement of the hadronic contribution to  $a_\mu$  in the MUonE experiment requires novel fast and efficient real-time based algorithms for the track and vertex reconstruction, together with a flexible trigger system. New event reconstruction methods developed for the MUonE experiment, based on the novel hardware-triggerless techniques or, alternatively, on machine learning methods implemented on parallel GPU processing, may become a standard approach in the future High Energy Physics experiments, facing an enormously tight execution time imposed by a fully software trigger system and achieving a maximum possible event reconstruction efficiency and precision. It will allow to efficiently reduce the size of data expected to increase fastly in the future experiments, and also to maximize the statistical power of the final physics measurement. In the MUonE experiment the algorithms of track finding based on machine learning techniques are being developed and tested in order to speed up the reconstruction process. This may lead to the significant acceleration of the execution of pattern recognition algorithms in the real-time event reconstruction algorithms. Moreover, the use of DNN techniques may significantly improve both the reconstruction efficiency and the precision of measuring the parameters that are crucial for final measurement.

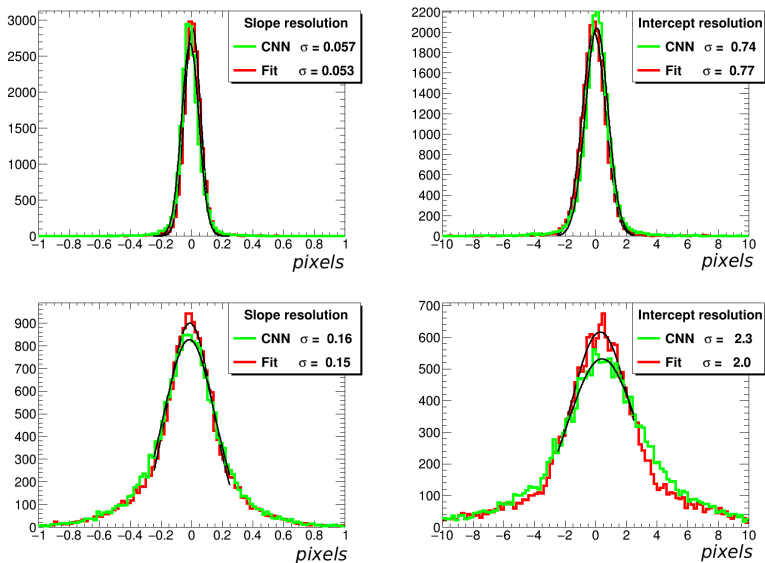
### 4.1. Two-dimensional machine learning based reconstruction

A first approach to use the machine learning techniques in context of the MUonE experiment [29] was based on an image-based model and a convolutional neural network [30]. This type of neural network is predominantly used in computer vision tasks, using a set of filters (called kernels) that analyze the image with a relatively small perception window (few pixels in size) that scans the input to produce the activation map of the filter. Network creates a set of filters sensible to different features in the input. Single filter can find multiple instances of the feature in the input image.

Training and testing datasets were generated using a two-dimensional toy-model. In total  $40 \cdot 10^3$  events corresponding to an elastic scattering signal with the MUonE 2018 test beam configuration (see Sec. 3.2) were produced. Each event contained one or two tracks reconstructed with the linear fit and was represented with a two-dimensional  $28 \times 28$  pixel image. Optionally, the noise was also included.

The neural network was implemented in KERAS [25] with TensorFlow [1] backend. It was trained to respond to the input image with slopes and intercepts of the tracks. Two convolutional layers were used with  $3 \times 3$  convolution window, followed by the MaxPooling layer and another two convolutional layers. The dropout layer was used to control the overtraining. Final regression was performed using the 1024-node dense layer. The full network had over 2 million trainable parameters. For multi-track events, long/short-term memory (LSTM) mechanism was used. It was inspired by the work of HEP.TrkX project [16, 18]. To make events more realistic, noise and pixel inefficiency was introduced. Noise could be defined in the 0–30% range, meaning a probability of pixel not belonging to the track to generate a signal. Pixel efficiency was lowered to 70% by changing the probability of track pixel to generate a signal.

Results provided by the CNN were used to find hits closest to the track candidates. The tracks were reconstructed using linear robust fit [10]. Differences between reconstructed and true tracks are shown in Figure 9, including 10% and 30% noise levels. Results were compared with the classical reconstruction algorithm (included in the plots). The neural network based approach proved to be successful and prompted the further development using three-dimensional approach.



**Figure 9.** Comparison of the distributions of the difference between the reconstructed and true track parameters: slope (left) and intercept (right), between CNN-based and classical track reconstruction. Noise level at 10% (top) and 30% (bottom). Figure taken from [29]

## 4.2. Three-dimensional DNN based track reconstruction

A natural next step after the work described in the previous section was an implementation of the machine learning based approach into three dimensions [40], that would meet much better the requirements of the MUonE experiment. In general an artificial neural network was designed to reproduce properly the track parameters. Based on the set of hit coordinates, the network's task is to predict the slope and the intercept of the track for each of two outgoing  $\mu$ - $e$  elastic scattering signal particles.

### 4.2.1. Learning dataset

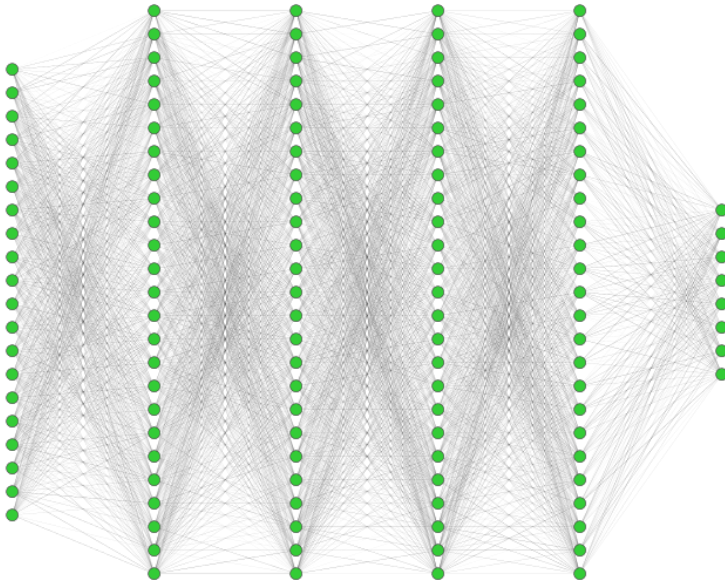
The tracking detector of the MUonE experiment (in the final detector configuration as well as in the 2018 test run) is based on silicon strip sensors which provide only one value for the measurement that can be combined with the sensor position along the beam axis to create two-dimensional representation of the hit position, i.e.  $(x, z)$  or  $(y, z)$ . An assumption was imposed that input data for the network will not include the information about the type of the hit ( $x$ ,  $y$ , or  $u$ ,  $v$  for stereo layers), but all the hits will be ordered by an increasing  $z$  coordinate.

The training dataset was generated using a leading-order event generator with the detector simulation preformed with GEANT4 [7]. The sample contained about  $100 \cdot 10^3$  events corresponding to the MUonE 2018 test beam setup described in Sec. 3.2. Input vector of the neural network consisted of 20 floating point values representing the measurements of the detector. Hits were arranged by the increasing  $z$  coordinate, without distinction between  $x$ ,  $y$  and *stereo* hits. The  $z$  coordinates were not included in the input vector, as they were identical in all events. For each event a ground truth was provided in the form of slope and intercept of outgoing tracks, 12 values in total. The dataset was split into training and testing subsets in the 4:1 ratio.

### 4.2.2. Artificial neural network

The PyTorch [32] was chosen as the machine learning framework, as it incorporates tools needed for data handling, training process and inference, all with GPU support to accelerate underlying matrix-based computations. The input vector contained collection of hits described in the previous section. To reduce its size as well as the size of correlated network and its complicity, information not critical for the track reconstruction was removed. Hits were sorted by ascending  $z$  value, which made explicit use of this coordinate, repeating in all events, redundant. In addition, hits related to the incoming muon were skipped, as the algorithm focused on the reconstruction of outgoing tracks. Final input vector included 20 values, each representing a measurement made by a silicon strip sensor. There was no distinction among  $x$ ,  $y$  and *stereo* hits. The output vector contained slopes and intercepts of the two outgoing tracks, represented in  $x$ - $z$  and  $y$ - $z$  projections, totalling in 8 values. For ease of comparison, this is the same format as the ground truth was provided in.

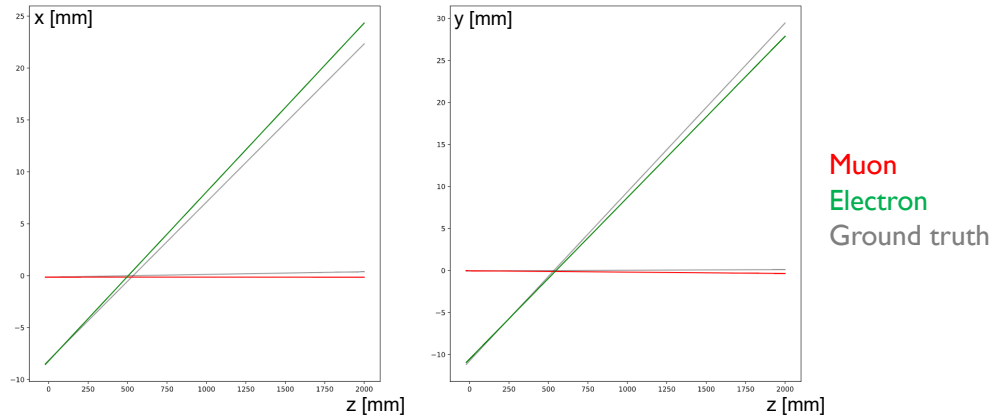
The artificial neural network consisted of four fully connected layers of 1000 neurons each, with additional layers for input and output (see Fig. 10). It is important to mention that at this stage of development no hyperparameter optimization was performed, i.e. parameters of the network were arbitrarily set to achieve acceptable results in reasonable time during the development. The *MSELoss* (Mean Squared Error Loss) [36] was chosen as the loss function, its implementation from the PyTorch package was used. In the process of training, the network was optimized in a way that minimized the mean squared error between the output and ground truth. In terms of the activation function, ReLU (rectified linear unit) was used, as being suitable for deep neural networks.



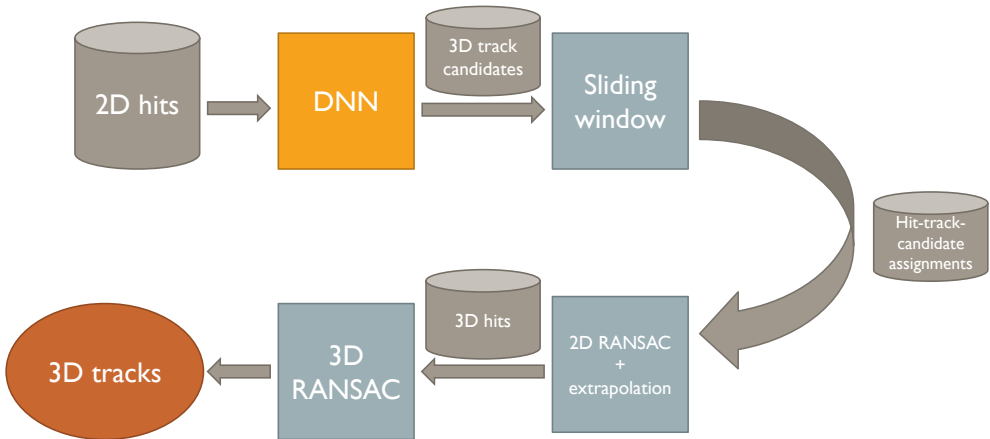
**Figure 10.** Architecture of the neural network used for track reconstruction. The input layer has 20 nodes, the following four hidden layers have 1000 nodes each and the output layer consists of 8 nodes. Number of nodes in the hidden layers in plot was reduced from 1000 to from 25 for clarity

### 4.2.3. Reconstruction algorithm

Comparison of the DNN-predicted tracks with the ground truth revealed that tracks are relatively close to each other (see Fig. 11), however reconstruction did not provide a precision required in the experiment. More complex algorithm had to be developed, where DNN was responsible only for the pattern recognition, being the most CPU time-consuming stage of the reconstruction in comparison to relatively fast linear fitting. Proposed algorithm is presented in Figure 12 and described in the following sections.



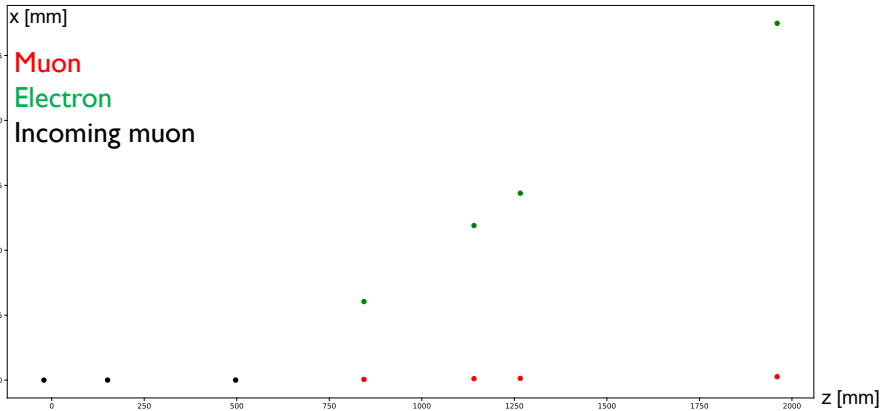
**Figure 11.** Example of the tracks reconstructed by the DNN, before applying further steps of the reconstruction algorithm. The ground truth shown in grey



**Figure 12.** The DNN-based algorithm for track reconstruction

- Deep neural network based pattern recognition.

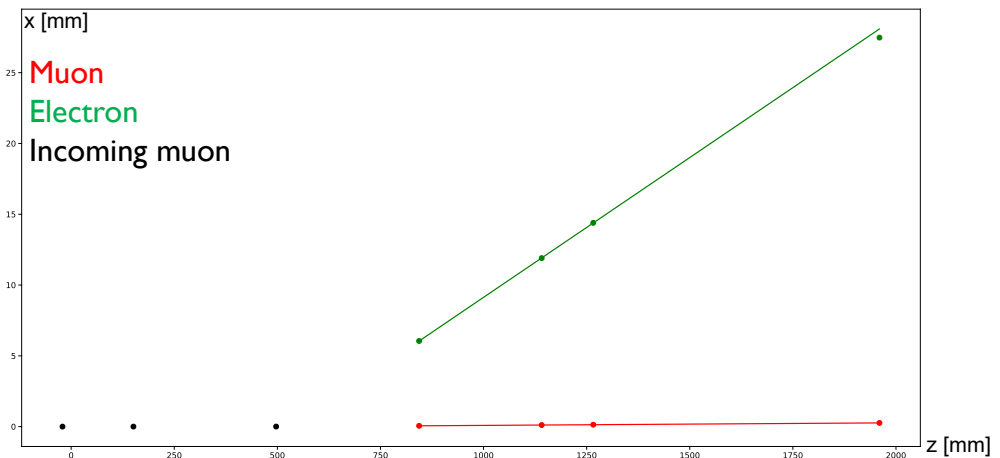
In the first step, the DNN machine learning model was used to turn the collection of hits representing  $\mu$ - $e$  elastic scattering signal event into three-dimensional track candidates. Every hit was then assigned to the DNN-reconstructed track that was the closest geometrically in its plane. Example of the event with hits assigned to the tracks is shown in Figure 13.



**Figure 13.** Example of the collections of hits corresponding to the  $\mu$ - $e$  elastic scattering signal tracks constructed based on the DNN-predicted track candidates. Points represent the hits in  $x$ - $z$  projection, colours correspond to the particle type

- Two-dimensional linear fit.

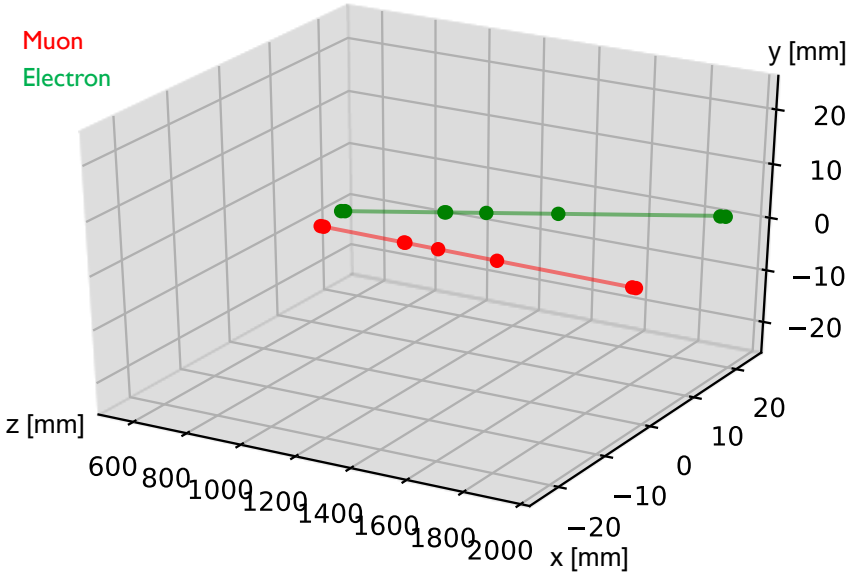
At this stage, RANSAC method (*Random Sample Consensus*) [19] was used to reconstruct two-dimensional temporary tracks in  $z$ - $x$  and  $z$ - $y$  projections. The RANSAC iterative algorithm was chosen as it is effective for outlier removal. Implementation provided in Scikit-learn package [33] was used. As a result, two 2D lines were established in both  $z$ - $x$  and  $z$ - $y$  planes. Example of temporary tracks resulting from the 2D linear fit in  $z$ - $x$  projection is shown in Figure 14.



**Figure 14.** Example of the result of the linear fit (solid lines) in  $x$ - $z$  projection for an outgoing muon and electron from the  $\mu$ - $e$  elastic scattering signal event. The points represent the hits in  $x$ - $z$  projection, colours correspond to the particle type

- Final 3-dimensional track fit.

Two-dimensional temporary tracks from the previous step were used to extrapolate the missing coordinate for each hit. With a collection of 3D hits assigned to every track, the final linear fit was performed with 3-dimensional RANSAC algorithm. Example of reconstructed three-dimensional tracks is shown in Figure 15.



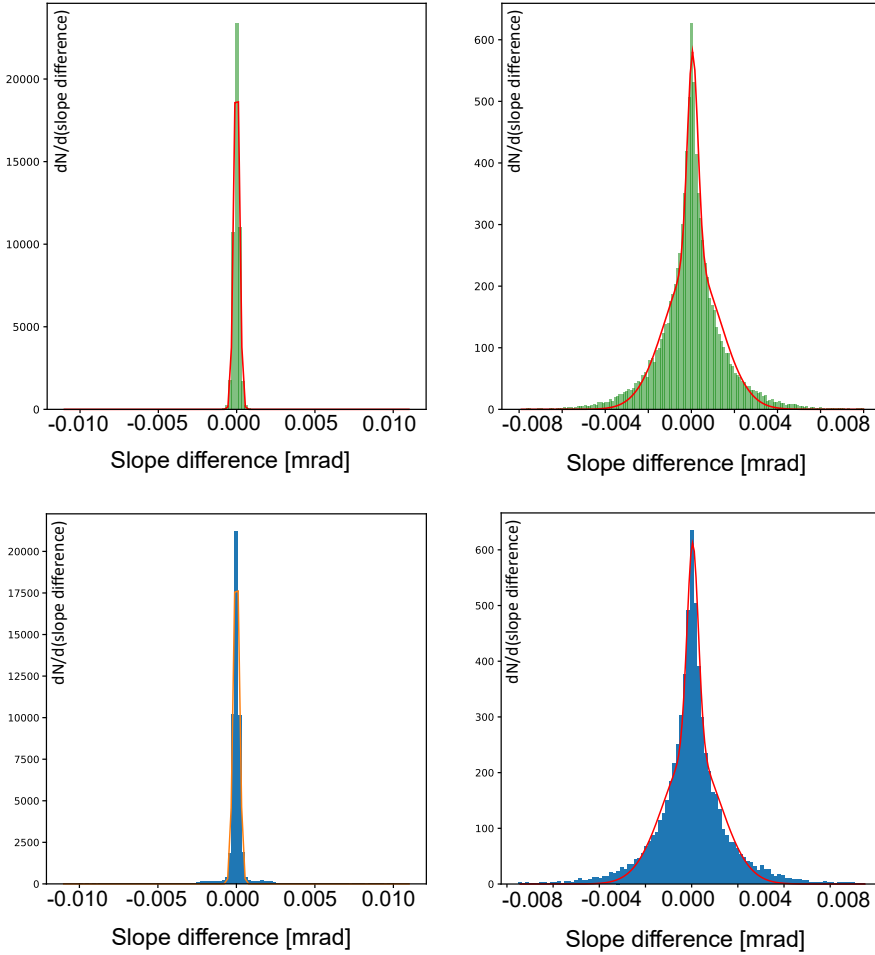
**Figure 15.** Example of the result of the 3D linear track fit (solid lines – red for muon and green for electron) for an outgoing muon and electron from the  $\mu$ - $e$  elastic scattering signal event. The points represent the hits, colours correspond to the particle type (red for muon and green for electron)

#### 4.2.4. Results

To assess the quality of the reconstructed tracks, the resolutions in track slopes were determined. To achieve this, histograms of difference in slopes between reconstructed track and ground truth were fitted with the Gaussian distribution, and the value of the standard deviation was interpreted as the resolution. In the case of electron track double Gaussian was used as this particle is more affected by the multiple scattering. The same procedure was performed for the tracks reconstructed with classical algorithm. Distributions of the slope differences are shown in Figure 16 and resolutions are summarized in Table 1. Additionally, efficiencies of the reconstruction algorithms were compared. Efficiency in this case is defined as the percentage of the tracks with the slope calculated with difference less than  $1 \cdot 10^{-2}$  when compared to the ground truth. Efficiencies of reconstruction algorithms are compared in Table 2. Results achieved by the DNN-based algorithm are on pair with the conventional algorithm. Differences, if



present, are not significant. This shows that the machine learning approach to track reconstruction has a great potential and should be investigated further.



**Figure 16.** Distributions of slope difference of reconstructed tracks (left for muons, right for electrons) in relation to the MC truth, for DNN-based algorithm (upper plots) and classical reconstruction (bottom plots)

**Table 1**

Slope resolutions for an outgoing muon and electron

Particle	DNN based	Classical
Muon	$\sigma = 0.000018$ mrad	$\sigma = 0.000019$ mrad
Electron	$\sigma_1 = 1.290$ mrad, $\sigma_2 = 0.245$ mrad	$\sigma_1 = 1.230$ mrad, $\sigma_2 = 0.244$ mrad

**Table 2**  
Efficiencies of reconstruction algorithms, as defined in the text

Particle	DNN based [%]	Classical [%]
Muon	100	99.98
Electron	99.66	99.38

## 5. Summary and outlook

The present DNN based algorithm prototype for the three-dimensional track reconstruction in the MUonE experiment proved to be competitive with the classical track reconstruction tasks in terms of quality, with potential performance benefits. Further development will involve the implementation of the neural network architecture based on Graph Neural Network. Graph Neural Networks (GNNs, subsection 2.2) ensure, among others, the inductive bias, reduction of number of parameters, more elaborated loss function, and above all a much more natural data representation. New event reconstruction methods being developed for the MUonE experiment, based on the novel hardware-triggerless techniques using machine learning methods may become a standard approach in the future High Energy Physics experiments, facing an enormously tight execution time imposed by a fully real-time reconstruction and achieving a maximum possible efficiency and precision. Although such techniques are not yet applied on a scale in any High Energy Physics experiments, they are intensively developed and planned to be employed in the near future.

### Acknowledgements

*This research was supported by the National Science Centre NCN (Poland) under the contract no. 2022/45/B/ST2/00318 and also supported in part by PL-Grid Infrastructure.*

### References

- [1] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., *et al.*: TensorFlow: Large-scale machine learning on heterogeneous distributed systems, *arXiv preprint arXiv:160304467*, 2016.
- [2] Abbiendi G.: Letter of Intent: the MUonE project, *CERN-SPSC-2019-026, SPSC-I-252*, 2019. <https://cds.cern.ch/record/2677471>.
- [3] Abbiendi G., Ballerini G., Banerjee D., Bernhard J., Bonanomi M., Brizzolari C., Foggetta L., *et al.*: A study of muon-electron elastic scattering in a test beam, *Journal of Instrumentation*, vol. 16(06), P06005, 2021. doi: 10.1088/1748-0221/16/06/p06005.
- [4] Abbiendi G., Carloni Calame C.M., Marconi U., Matteuzzi C., Montagna G., Nicosini O., Passera M., *et al.*: Measuring the leading hadronic contribution to the muon  $g-2$  via  $\mu e$  scattering, *The European Physical Journal C*, vol. 77(3), 2017. doi: 10.1140/epjc/s10052-017-4633-z.

- [5] Abe M., Bae S., Beer G., Bunce G., Choi H., Choi S., Chung M., *et al.*: A new approach for measuring the muon anomalous magnetic moment and electric dipole moment, *Progress of Theoretical and Experimental Physics*, vol. 2019(5), 053C02, 2019.
- [6] Abi B., Albahri T., Al-Kilani S., Allspach D., Alonzi L.P., Anastasi A., Anisenkov A., *et al.*: Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm, *Physical Review Letter*, vol. 126, 141801, 2021. doi: 10.1103/PhysRevLett.126.141801.
- [7] Agostinelli S., Allison J., Amako K., Apostolakis J., Araujo H., Arce P., Asai M., *et al.*: GEANT4—a simulation toolkit, *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506(3), pp. 250–303, 2003. doi: 10.1016/S0168-9002(03)01368-8.
- [8] Bennett G.W., Bousquet B., Brown H.N., Bunce G., Carey R.M., Cushman P., Danby G.T., *et al.*: Final report of the E821 muon anomalous magnetic moment measurement at BNL, *Physical Review D*, vol. 73(7), 072003, 2006.
- [9] Bhattacharya S., Chernyavskaya N., Ghosh S., Gray L., Kieseler J., Klijsma T., Long K., *et al.*: GNN-based end-to-end reconstruction in the CMS Phase 2 High-Granularity Calorimeter, 2022. doi: 10.48550/ARXIV.2203.01189.
- [10] Brun R., Rademakers F.: ROOT – An object oriented data analysis framework, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 389(1), pp. 81–86, 1997. doi: 10.1016/S0168-9002(97)00048-X.
- [11] Caillou S., Calafiura P., Farrell S.A., Ju X., Murnane D.T., Rougier C., Stark J., Vallier A.: ATLAS ITk Track Reconstruction with a GNN-based pipeline. Technical report: ATL-COM-ITK-2022-057, 2022. <https://cds.cern.ch/record/2815578/>.
- [12] Cerati G., Elmer P., Krutelyov S., Lantz S., Lefebvre M., McDermott K., Riley D., *et al.*: Kalman filter tracking on parallel architectures, *Journal of Physics: Conference Series*, vol. 898, 042051, 2017. doi: 10.1088/1742-6596/898/4/042051.
- [13] Chollet F.: *Deep learning with Python*, Simon and Schuster, 2021.
- [14] Choma N., Murnane D., Ju X., Calafiura P., Conlon S., Farrell S., Cerati G., *et al.*: Track seeding and labelling with embedded-space graph neural networks, *arXiv preprint arXiv:200700149*, 2020.
- [15] Doble N., Gagnon L., Holtey von G., Novoskoltsev F.N.: The upgraded muon beam at the SPS, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 343(2–3), pp. 351–362, 1993. doi: 10.1016/0168-9002(94)90212-7.
- [16] Farrell S., Anderson D., Calafiura P., Cerati G., Gray L., Kowalkowski J., Mudigonda M., *et al.*: The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking, *EPJ Web of Conferences*, vol. 150, 00003, 2017. doi: 10.1051/epjconf/201715000003.

- [17] Farrell S., Calafiura P., Mudigonda M., Mr. Prabhat, Anderson D., Bendavi J., Spiropoulou M., *et al.*: Particle Track Reconstruction with Deep Learning. In: *31st Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. [https://dl4physicalsciences.github.io/files/nips\\_dlps.2017.28.pdf](https://dl4physicalsciences.github.io/files/nips_dlps.2017.28.pdf).
- [18] Farrell S., Calafiura P., Mudigonda M., Prabhat, Anderson D., Vlimant J.R., Zheng S., *et al.*: Novel deep learning methods for track reconstruction, 2018. doi: 10.48550/ARXIV.1810.06111.
- [19] Fischler M.A., Bolles R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Commun ACM*, vol. 24(6), pp. 381–395, 1981. doi: 10.1145/358669.358692.
- [20] Funika W., Koperek P., Kitowski J.: Continuous self-adaptation of control policies in automatic cloud management, *Concurrency and Computation: Practice and Experience*, vol. 35(20), e7371, 2023.
- [21] Hochreiter S., Schmidhuber J.: Long short-term memory, *Neural Computation*, vol. 9(8), pp. 1735–1780, 1997.
- [22] Ju X., Farrell S., Calafiura P., Murnane D., Prabhat, Gray L., Klijsma T., *et al.*: Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors. In: *33rd Annual Conference on Neural Information Processing Systems*, 2020.
- [23] Ju X., Murnane D., Calafiura P., Choma N., Conlon S., Farrell S., Xu Y., *et al.*: Performance of a geometric deep learning pipeline for HL-LHC particle tracking, *The European Physical Journal C*, vol. 81, 876, 2021.
- [24] Kalman R.E.: A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, vol. 82(1), pp. 35–45, 1960.
- [25] Keras, <https://keras.io>, 2015.
- [26] Kopciwicz P., Akiba K.C., Szumlak T., Sitko S., Barter W., Buytaert J., Eklund L., *et al.*: Simulation and optimization studies of the LHCb beetle readout ASIC and machine learning approach for pulse shape reconstruction, *Sensors*, vol. 21(18), 6075, 2021.
- [27] Kopciwicz P., Szumlak T., Majewski M., Akiba K., Augusto O., Back J., Bobulska D.S., *et al.*: The upgrade I of LHCb VELO – towards an intelligent monitoring platform, *Journal of Instrumentation*, vol. 15(06), C06009, 2020. doi: 10.1088/1748-0221/15/06/C06009.
- [28] Krzywda M., Lukasik S., Gandomi A.H.: Graph Neural Networks in Computer Vision-Architectures, Datasets and Common Approaches. In: *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, 2022.
- [29] Kucharczyk M., Wolter M.: Track Finding with Deep Neural Networks, *Computer Science*, vol. 20(4), 2019. doi: 10.7494/csci.2019.20.4.3376.
- [30] LeCun Y., Bengio Y., Hinton G.: Deep Learning, *Nature*, vol. 521, pp. 436–444, 2015. doi: 10.1038/nature14539.
- [31] LHCb Experiment, LHCb Collaboration: Event collected at the beginning of 2018 data taking, 2018. <http://cds.cern.ch/record/2315673>.

- [32] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., *et al.*: PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019.
- [33] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., *et al.*: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [34] Peterson C.: Track finding with neural networks, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 279(3), pp. 537–545, 1989.
- [35] Piekarczyk M., Bar O., Bibrzycki L., Niedźwiecki M., Rzecki K., Stuglik S., Andersen T., *et al.*: CNN-based classifier as an offline trigger for the CREDO experiment, *Sensors*, vol. 21(14), 4804, 2021. doi: 10.3390/s21144804.
- [36] PyTorch: MSELoss – PyTorch 1.11.0 documentation. <https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html>.
- [37] Salzburger A.: The ATLAS Track Extrapolation Package, Technical report ATL-SOFT-PUB-2007-005; ATL-COM-SOFT-2007-010, 2007. <https://cds.cern.ch/record/1038100>.
- [38] Scarselli F., Gori M., Tsoi A.C., Hagenbuchner M., Monfardini G.: The graph neural network model, *IEEE Transactions on Neural Networks*, vol. 20(1), pp. 61–80, 2008.
- [39] Vinyals O., Toshev A., Bengio S., Erhan D.: Show and tell: A neural image caption generator. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2015.
- [40] Zdybał M., Kucharczyk M., Wolter M.: DNN Based Prototype of the Track Reconstruction Algorithm for the MUonE Experiment. In: S.R. González, J.M. Machado, A. González-Briones, J. Wikarek, R. Loukanova, G. Katranas, R. Casado-Vara (eds.), *Distributed Computing and Artificial Intelligence, Volume 2: Special Sessions 18th International Conference*, pp. 202–205, Springer International Publishing, Cham, 2022.

## Affiliations

### Miłosz Zdybał

The Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences,  
<https://www.ifj.edu.pl>, [miłosz.zdybal@ifj.edu.pl](mailto:miłosz.zdybal@ifj.edu.pl)

### Marcin Kucharczyk

The Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences,  
<https://www.ifj.edu.pl>, [marcin.kucharczyk@ifj.edu.pl](mailto:marcin.kucharczyk@ifj.edu.pl)

### Marcin Wolter

The Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences,  
<https://www.ifj.edu.pl>, [marcin.wolter@ifj.edu.pl](mailto:marcin.wolter@ifj.edu.pl)

**Received:** 17.08.2023

**Revised:** 12.01.2024

**Accepted:** 12.01.2024