particles

IMPACT FACTOR 2.3

CITESCORE 3.0

Article

Pattern Recognition with Artificial Intelligence in Space Experiments

Federica Cuna, Maria Bossa, Fabio Gargano and Mario Nicola Mazziotta

Special Issue

Advances in Space AstroParticle Physics: Frontier Technologies for Particle Measurements in Space, 2025 Edition

Edited by

Dr. Matteo Duranti and Dr. Valerio Vagelli

MDPI

https://doi.org/10.3390/particles8040099

particles

MDPI

Article

Pattern Recognition with Artificial Intelligence in Space Experiments

Federica Cuna [1,2,*], Maria Bossa [2,3], Fabio Gargano [1,2] and Mario Nicola Mazziotta [1]

[1] Istituto Nazionale di Fisica Nucleare—Sezione di Bari, Via Giovanni Amendola, 173, 70126 Bari, Italy; fabio.gargano@ba.infn.it (F.G.); marionicola.mazziotta@ba.infn.it (M.N.M.)
[2] ICSC—Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, Via Magnanelli, 40033 Casalecchio di Reno, Italy; mbossa@na.infn.it
[3] Istituto Nazionale di Fisica Nucleare—Sezione di Napoli, Strada Comunale Cinthia, 80126 Napoli, Italy
* Correspondence: federica.cuna@ba.infn.it

Abstract: The application of advanced Artificial Intelligence (AI) techniques in astroparticle experiments represents a major advancement in both data analysis and experimental design. As space missions become increasingly complex, integrating AI tools is essential for optimizing system performance and maximizing scientific return. This study explores the use of Graph Neural Networks (GNNs) within the tracking systems of space-based experiments. A key challenge in track reconstruction is the high level of noise, primarily due to backscattering tracks, which can obscure the identification of primary particle trajectories. We propose a novel GNN-based approach for node-level classification tasks, specifically designed to distinguish primary tracks from backscattered ones within the tracker. In this framework, AI is employed as a powerful tool for pattern recognition, enabling the system to identify meaningful structures within complex tracking data and to discriminate signal from backscattering with higher precision. By addressing these challenges, our work aims to enhance the accuracy and reliability of data interpretation in astroparticle physics through the advanced deep learning techniques.

Keywords: Artificial Intelligence; Graph Neural Networks; HPC; space experiments

Academic Editors: Matteo Duranti, Valerio Vagelli and Armen Sedrakian

Received: 30 September 2025
Revised: 11 November 2025
Accepted: 1 December 2025
Published: 10 December 2025

Citation: Cuna, F.; Bossa, M.; Gargano, F.; Mazziotta, M.N. Pattern Recognition with Artificial Intelligence in Space Experiments. *Particles* **2025**, *8*, 99. https://doi.org/10.3390/particles8040099

1. Introduction

In recent years, Artificial Intelligence (AI) has made remarkable progress across various domains, including finance, social media, and natural language processing. Its impact in the field of physics is rapidly growing, with a wide array of AI techniques—particularly machine learning and deep learning—being increasingly adopted in experimental research. These methods, originally developed for tasks such as image recognition, have proven highly effective in enhancing data analysis in high-energy physics [1].

One significant example is the transformation of astrophysical research through machine learning, especially in automating the classification of astronomical objects. Convolutional Neural Networks (CNNs), for instance, have been successfully applied to galaxy classification based on morphological features [2].

More recently, advancements in machine learning and deep learning algorithms have extended their influence to space-based astroparticle physics. A notable example is the DAMPE (Dark Matter Particle Explorer) mission (https://dpnc.unige.ch/dampe/index.html) (accessed on 10 October 2025), an orbiting satellite since 2018, whose primary objective is to detect electrons and photons in the search for potential dark matter signatures. DAMPE has pioneered the implementation of fully machine-learned track reconstruction

pipelines, achieving substantial improvements over traditional methods. These advanced techniques not only increase the accuracy of particle tracking over a broad energy spectrum but also enable high-precision measurements of cosmic-ray proton and helium spectra, reaching energies up to the PeV scale [3].

This work focuses on applying AI techniques to space-based experiments, with the goal of developing robust, AI-driven algorithms capable of effectively handling real-world experimental data. Specifically, we explore the use of Graph Neural Networks (GNNs) for pattern recognition tasks [4], where pattern recognition refers to the process of identifying and grouping signals that originate from the same particle, thereby enabling the reconstruction of its trajectory within the tracking system. The HERD (High-Energy cosmic-Radiation Detection) facility [5] serves as the testbed for this study, since its simulation software framework, HerdSoftware [6], offers a detailed and accurate modeling of the detector, making it particularly suitable for the development and validation of AI-based approaches.

2. Materials and Methods

2.1. The HERD Experiment

The HERD (High-Energy cosmic-Radiation Detection) experiment is a forthcoming space mission dedicated to the direct detection of cosmic rays, scheduled for deployment on the Chinese Space Station in 2027.

HERD will feature a homogeneous, isotropic, and deeply segmented 3D calorimeter, surrounded by multiple sub-detectors dedicated to charge identification, timing, and particle tracking. Thanks to its innovative design, HERD will be capable of detecting particles from virtually all directions, ensuring a large geometric acceptance. Combined with excellent energy resolution, HERD's geometric factor will exceed that of current space-based experiments focused on proton and electron detection by approximately an order of magnitude. This enhanced sensitivity will enable HERD to measure the cosmic-ray proton flux up to energies of 1 PeV.

In addition, HERD will measure the electron and positron flux up to several tens of TeV. It will search for potential indirect signatures of dark matter and, through the detection of high-energy photons, will investigate candidate sources of high-energy cosmic rays [7].

HERD consists of several sub-detectors, arranged from the innermost to the outermost layers:

- CALO: The calorimeter is the core of the experiment. It measures the energy of incoming particles and helps distinguish electrons from protons and nuclei. It consists of approximately 7500 cubic LYSO scintillating crystals, each with a side length of 3 cm [7].
- FIT: The FIber Tracker surrounds CALO on the top and on the four lateral sides. It is primarily designed for particle tracking and charge measurement. FIT is composed of five tracking sectors, each made of seven X-Y double layers of scintillating fibers, tightly arranged in parallel, providing seven independent position measurements for each traversing charged particle [8].
- PSD: The Plastic Scintillator Detector is composed of bars of plastic scintillators. It measures particle charge and serves as a trigger for low-energy gamma rays [9].
- SCD: The Silicon Charge Detector is a silicon micro-strip detector used for both tracking and charge identification of particles [9].
- TRD: The Transition Radiation Detector is located on one side of the detector. It is composed of Thick Gaseous Electron Multipliers (THGEMs) and is designed for the calibration of TeV-energy nuclei [9].

A schematic figure of the detector is shown in Figure 1.

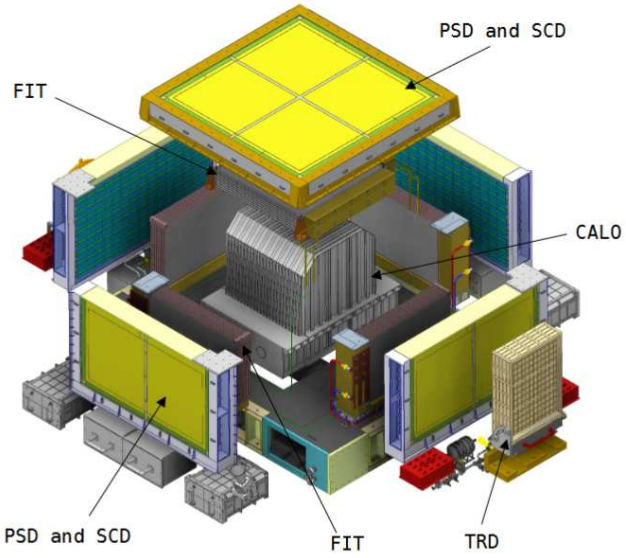


Figure 1. Schematic picture of HERD apparatus [7].

2.2. Tracking in High-Density Track Environment

Cosmic rays are of fundamental importance in understanding high-energy astrophysical processes in our galaxy and beyond. These energetic particles offer a unique window into the most extreme environments in the Universe, including supernova remnants, active galactic nuclei, and potential dark matter interactions. Studying the origin, composition, and propagation of cosmic rays requires precise measurements—feasible only outside Earth's atmosphere. At ground level, the atmosphere acts as a natural calorimeter, preventing direct detection of primary cosmic rays.

To overcome this limitation, satellite-based missions such as DAMPE, HERD, and FERMI [10] have been developed to perform direct cosmic-ray measurements in space. These missions aim to collect high-precision data on charged particles and photons across a wide energy range, thereby addressing fundamental questions in high-energy astrophysics and particle physics. Among their key objectives are the investigation of particle acceleration mechanisms in astrophysical sources, the study of cosmic-ray propagation in the Milky Way, and the search for new physics phenomena such as dark matter annihilation or decay [11].

Employing deep calorimeters for the measurement of supra-TeV cosmic charged radiation (CCR), HERD faces significant challenges related to the loss of tracking efficiency at high energies. This degradation arises primarily from the experimental noise caused by the copious production of backscattered secondary particles within the calorimeter, a phenomenon already observed in large-acceptance space-based calorimetric experiments. In conventional silicon and fiber detectors, hits generated by energy deposits from backscattered secondary particles are indistinguishable from those produced by the primary particle. Consequently, both hit clustering and particle tracking efficiencies are adversely affected. This effect becomes increasingly pronounced with the rising number of backscattered particles, and hence grows more severe as the energy of the primary particle increases [12]. A pictorial view of the backscattering phenomenon is presented in Figure 2, while a typical event generated by a proton track is shown in Figure 3.

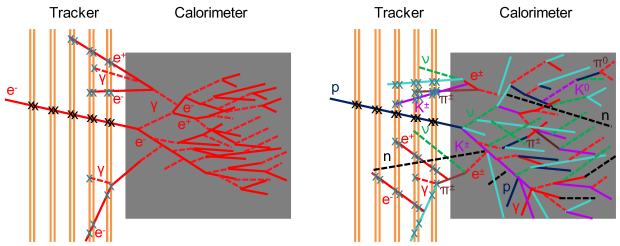


Figure 2. Representation of secondary particle tracks in a tracking detector positioned upstream of a calorimeter. Primary electrons (**left**) induce an electromagnetic shower within the calorimeter, producing primarily ultra-relativistic $e^{\pm}$ and γ backscattered secondaries detectable in the upstream tracking layers. In contrast, primary protons and nuclei (**right**) initiate a hadronic shower (schematically illustrated here for clarity) inside the calorimeter, which can generate a population of slower backscattered secondary particles detectable by the upstream tracker [12].

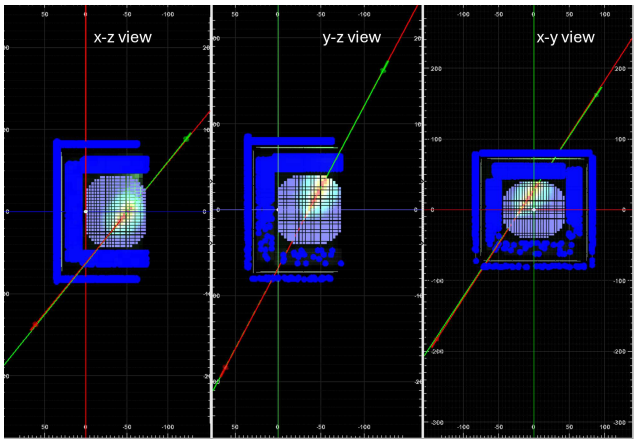


Figure 3. Event display of proton track inside HERD detector.

3. Methodology

This section introduces the methodology followed to develop the GNN pattern recognition algorithm in the HERD FIT detector.

3.1. Pipeline for AI Pattern Recognition Algorithm

Since tracking data can naturally be represented as graphs, which is a collection of nodes and links, as it explained in detail next, we explore the use of a GNN architecture to reconstruct tracks within the FIT, by selecting the pattern of signal clusters over the backscattering ones.

Figure 4 illustrates the pipeline used for developing the GNN tracking algorithm.

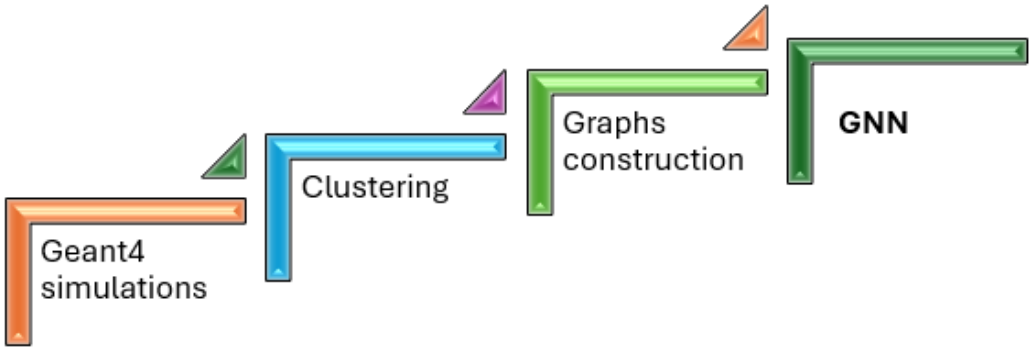


Figure 4. Visual illustration of the GNN tracking algorithm definition.

The process begins with the simulation of the data set, followed by two data preprocessing steps (clustering and graph construction). Finally, the pipeline culminates in the definition of a GNN-based pattern recognition algorithm: it consists of a GNN classification of signal clusters over backscattering clusters.

A final linear fit stage to retrieve the track parameter (angular coefficient and intercept) will be implemented to create a fully tracking pipeline.

The next subsection provides a detailed description of each step.

3.2. The HERD Simulated Dataset

The full dataset, generated using the custom HerdSoftware simulation framework, consists of proton tracks simulated within a power-law energy spectrum E^{-1}, spanning an energy range from

- 1 GeV to 100 GeV (66,293 simulated tracks);
- 100 GeV to 10 TeV (204,229 simulated tracks).

Tracks are distributed within a spherical region surrounding the HERD detector.

3.3. Data Preprocessing

To develop a high-performance AI tracking algorithm, a meticulous data preprocessing procedure was carried out. Tracks passing through the fiber tracker produce energy deposits identified as "hits". These hits, detected within each side and layer of the FIT, are grouped using a traditional clustering algorithm that aggregates neighboring activated silicon strips. The charge barycenter of each cluster is then computed.

Taking as reference the reconstructed shower calorimeter axis, clusters lying outside a cylindrical region of 3.5 cm radius are discarded.

Data are structured in a graph format, where each node corresponds to a cluster and edges represent inter-layer connections between clusters. For this preliminary analysis, only tracks traversing the top side of the FIT have been considered.

An example of a track described by a graph is shown in Figure 5.

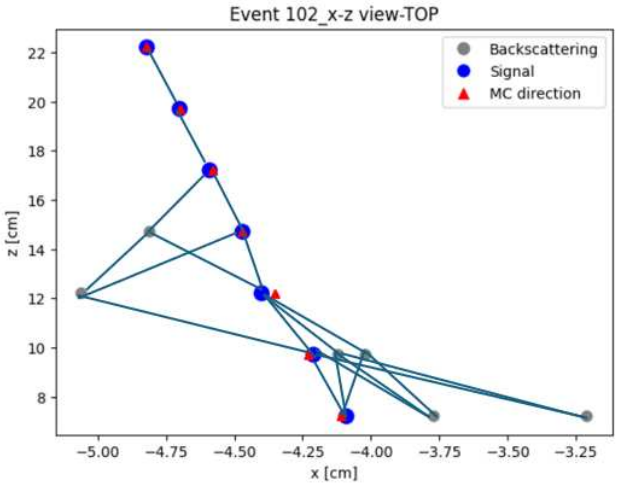


Figure 5. An example of track data represented as a graph. The grey dots represent the backscattering clusters (class 0), the blue ones represent the signal clusters (class 1), and the red triangles represent the primary track direction.

3.4. The Graph Neural Network

The representation of a track as a graph allows the implementation of a powerful class of methods from Geometric Deep Learning: the Graph Neural Network.

A GNN model can learn on this representation and solve tasks with predictions over the graph nodes, edges, or global state.

In GNNs, data is represented as a graph G, consisting of

- Nodes: Represent entities with specific attributes.
- Edges: Represent connections or relationships between nodes, carrying attributes as well.
- Global attributes: Represent information about the entire graph.

GNNs operate through Graph Network (GN) blocks, which iteratively update node, edge, and global attributes using update and aggregation functions. These functions pass messages along edges, aggregate them at nodes, and compute global properties, enabling the network to learn from local and global graph structures [13].

GNNs are highly flexible and can generalize to unseen graphs by focusing computations locally (at edges and nodes). When multiple layers are used, the network can capture more abstract and long-range patterns, making it ideal for tasks involving graph-structured data.

There are different types of GNNs, like the Graph Convolutional Neural Network [14], GraphSAGE and SageConv [15], Graph Attention Networks [16], and interaction network [17].

3.5. The Interaction Network

The architecture used in this paper to develop the AI pattern recognition algorithm is an interaction network [17], a variant of the Graph Neural Network, whose implementation is similar to the one developed by the HEP.TrkX project (https://heptrkx.github.io/).

It consists of an encoder stage and two core modules: an edge network and a node network.

The edge network computes edge weights using the features of the start and end nodes; the node network updates each node's features based on the aggregated edge-weighted features of neighboring nodes. The aggregation is performed separately for connections to the previous and next detector layers, and is combined with the node's current features.

Both the EdgeNetwork and NodeNetwork are implemented as Multi-Layer Perceptrons (MLPs).

Two versions of the encoder stage have been tested. The encoder stage has the goal of initializing the graph representation. The first version applies two independent two-layer fully connected networks to transform the raw input features of the nodes and edges, and the second version uses a Multi-Head Feature Attention mechanism. Instead of directly projecting input features through fully connected layers, this encoder employs a multi-head attention module to learn more expressive representations of the input node features. Specifically, the encoder maps each node's input features into multiple attention heads, enabling the model to capture different types of interactions in parallel. Each head performs scaled dot-product attention, and the resulting representations are then concatenated and passed through a final linear projection layer. An activation function is applied to introduce non-linearity.

The node and edge network are applied recursively for a fixed number of iterations N. At each iteration i, edge weights are recomputed by the edge network, and node features are updated by the node network using the aggregated information from neighboring nodes. After a fixed number of iterations, the final node representations are passed to the output module, which consists of a linear classifier with dropout regularization. This module outputs classification scores for each node, identifying whether it belongs to the primary track or to a backscattered track.

Figure 6 shows a diagram of the IN network.

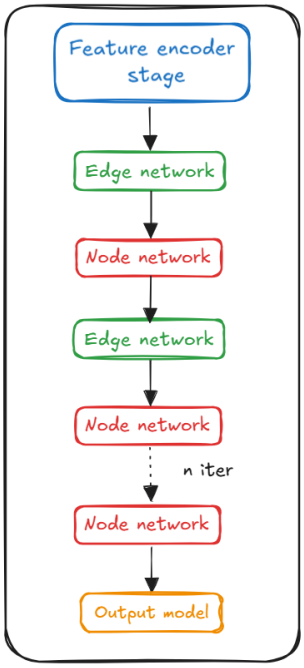


Figure 6. Diagram of the Graph Neural Network model which begins with an encoder stage and has a number of recurrent iterations of alternating EdgeNetwork and NodeNetwork components. In this case, the final output layer is the NodeNetwork, making this a node classifier model.

This GNN-based approach enables robust and context-aware classification of tracker clusters, allowing the model to effectively distinguish primary clusters from backscattered ones induced by backscattering, thus significantly improving tracking accuracy in high-noise environments.

3.6. Loss Function for the Imbalanced Dataset

Tracking data are highly imbalanced since there are many more backscattered clusters than there are primary ones, especially at the higher-energy ranges.

In this scenario, a model trained with standard cross-entropy loss tends to be biased toward the majority class, often failing to adequately learn the minority class, which results in suboptimal performance on underrepresented samples.
Beyond the different methods available in order to address the data imbalance, we decided to test the implementation of the focal loss function [18].

By introducing a scaling factor α_t, focal loss dynamically adjusts the learning focus toward harder, misclassified examples. This mechanism not only addresses class imbalance, but also emphasizes samples that the model finds most challenging, thereby improving learning efficiency and robustness. The equation of the focal loss is

$$L_{focal}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{1}$$

p_t is the model's predicted probability for the correct class: if p_t is low (the model is uncertain), the scaling factor $(1-p_t)^{\gamma}$ will be large, making the loss higher for this sample. So this extra term controls the focus on misclassified samples.

The parameter γ in focal loss acts as a focusing mechanism. When $\gamma = 0$, the focal loss reduces to standard cross-entropy loss. As γ increases, the contribution of well-classified examples—those with high predicted probability—is progressively down-weighted, while misclassified or hard examples receive greater emphasis. This modulation enables the model to concentrate more effectively on the samples it finds difficult to classify, introducing a level of adaptive learning not present in conventional cross-entropy loss. The parameter

α_t in focal loss serves as a class-balancing factor, allowing independent weighting of each class. This is particularly useful in scenarios with severe class imbalance, where the majority class can dominate the loss function. By assigning a higher α_t to the minority class and a lower value to the majority class, the model can be guided to pay more attention to underrepresented samples. In this work, rather than fixing α_t a priori, it is automatically computed for each batch based on the class distribution, ensuring that the weighting dynamically adapts to the data seen during training.

3.7. Training of the Network

To achieve optimal performance for the AI-based tracking algorithm, a critical component of the model development was the hyperparameter tuning. This is performed by using the RayTune framework [19]. Hyperparameter optimization plays a pivotal role in machine learning tasks: it allows for fine-tuning the model's settings to achieve the best possible generalization and efficiency. Given the complexity of the track reconstruction in the HERD experiment, the choice of hyperparameters significantly influences the model's ability to learn from the data, mitigate overfitting, and handle the high-dimensional and noisy nature of the input features. The iterative tuning process, therefore, ensures that the model operates at peak accuracy and performance across varying experimental conditions.

To conduct this hyperparameter search, two distinct hardware setups were employed to test the model's scalability and efficiency. The first setup involved distributed training across three NVIDIA (ITM: informatica e telematica meridionale)A100 GPUs, each with 40 GB of memory, each delivering substantial computational power for parallel processing and model training across large datasets. The second setup utilized a single NVIDIA H100 GPU with 94 GB of memory, selected for its high-performance capabilities in training deep learning models on a single node.

The hyperparameters which have been tuned are as follows:

- The number of iterations of edge and node networks;
- The activation function;
- The gamma parameter for the focal loss;
- The number of layers inside the node and edge network;
- The hidden dimensions of each layer;
- The number of heads for the Multi-Head Feature Attention approach;
- The learning rate;
- The batch size.

In addition to the hyperparameter tuning, an essential component of the training process was the integration of a learning rate scheduler, designed to dynamically adjust the learning rate during the training process based on the model's performance. Specifically, we employed a learning rate scheduler that reduces the learning rate by a factor of 0.5 whenever the loss does not improve over a given number of epochs. In our case, a patience of 5 epochs was used, a choice found empirically to provide stable convergence in our tests, meaning the learning rate is halved only after five consecutive epochs without improvement. This strategy enables finer control over the training process, helping prevent overfitting and promoting more stable convergence by lowering the learning rate when model performance reaches a plateau.

The best model has been chosen as the one that maximizes the f1 score.

The training procedure has been conducted in the first two energy ranges, from 1 GeV up to 100 GeV and from 100 GeV up to 10 TeV.

4. Results

Tables 1 and 2 summarize the performance of the two variants of the interaction network in the energy ranges of 1 GeV–100 GeV and 100 GeV–10 TeV, respectively.

Table 1. Evaluation metrics for the 1 GeV–100 GeV energy range.

Metrics	IN with Linear Encoder	IN with Attention Encoder
Accuracy	80.14	80.11
Precision	80.20	81.15
Recall	86.43	80.67
F1-score	80.19	80.18

Table 2. Evaluation metrics for the 100 GeV–10 TeV energy range.

Metrics	IN with Linear Encoder	IN with Attention Encoder
Accuracy	81.85	82.11
Precision	77.68	79.65
Recall	71.31	71.29
F1-score	74.11	73.97

The learning curves for the 1 GeV–100 GeV energy range are reported in Figure 7.

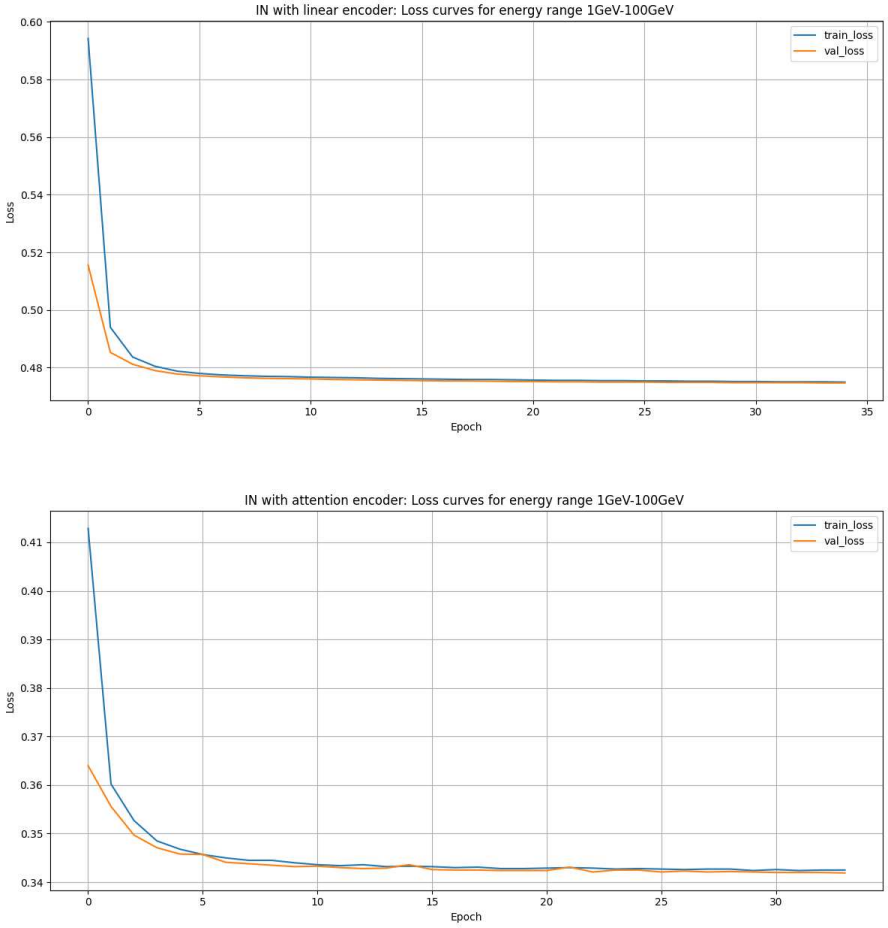


Figure 7. Learning curves for 1 GeV–100 GeV for interaction network with linear encoder (**top**) and interaction network with attention encoder (**bottom**).

The learning curves for the 100 GeV–10 TeV energy range is reported in Figure 8.

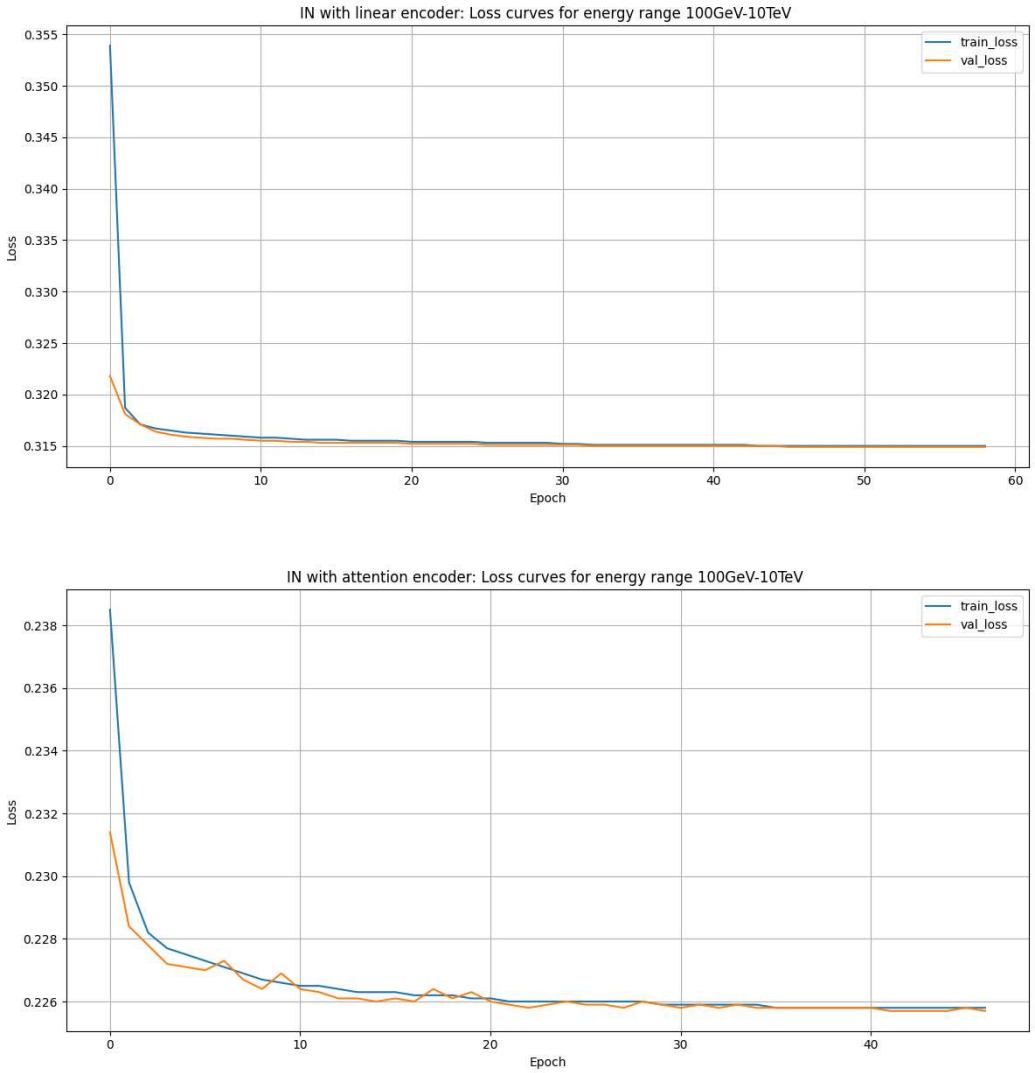
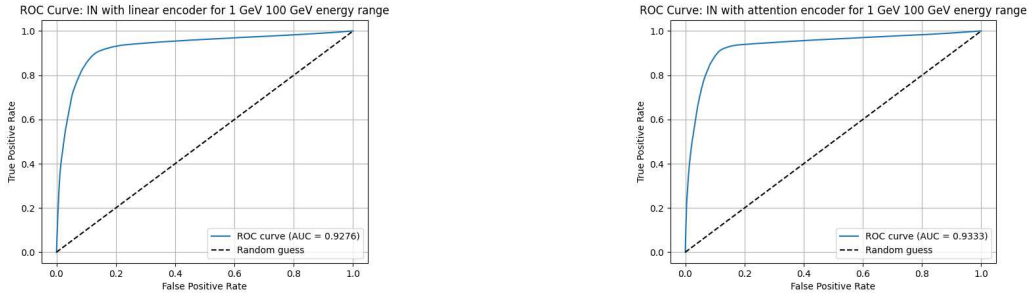


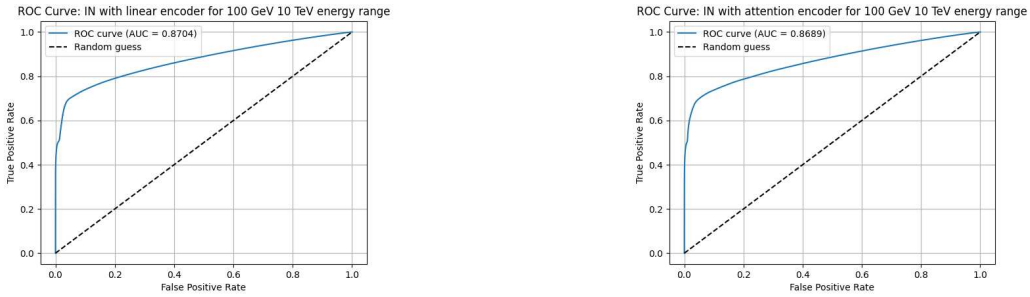
Figure 8. Learning curves for 100 GeV–10 TeV for interaction network with linear encoder (**top**) and interaction network with attention encoder (**bottom**).

Figure 9 shows the roc curves for the two variants of the interaction network in the energy ranges of 1 GeV–100 GeV and 100 GeV–10 TeV.



(**a**) (**b**)

Figure 9. *Cont.*



(**c**) (**d**)

Figure 9. ROC curves for the two variants of the interaction network across the tested energy ranges. (**a**) ROC curve for the interaction network with the linear encoder in the 1 GeV–100 GeV energy range. (**b**) ROC curve for the interaction network with the attention encoder in the 1 GeV–100 GeV energy range. (**c**) ROC curve for the interaction network with the linear encoder in the 100 GeV–10 TeV energy range. (**d**) ROC curve for the interaction network with the attention encoder in the 100 GeV–10 TeV energy range.

5. Discussion

The results obtained with the two interaction network architectures confirm the feasibility of applying GNN-based methods to tracking data in space experiments. The comparison between the linear encoder and the attention-based encoder across the two energy intervals (1–100 GeV and 100 GeV–10 TeV) highlights complementary behaviors that reflect the interplay between signal topology and background from backscattered particles.

At low energies (1–100 GeV), both models achieve competitive and balanced performance, as illustrated by the ROC curves in the top panels of Figure 9. The linear encoder shows the highest recall (86.43 %), demonstrating a stronger ability to correctly identify signal hits, while the attention encoder achieves higher precision (81.15%), reflecting a better suppression of false positives. These results suggest that the linear encoder favors sensitivity to signal clusters, whereas the attention mechanism is more effective at discriminating against backscattering.

In the higher-energy range (100 GeV–10 TeV), the challenge shifts significantly. With increasing energy, backscattered secondaries become more numerous, making the task of separating signal from backscattering more demanding. Despite this, both models maintain stable performance, with the attention encoder achieving slightly higher accuracy (82.11% vs. 81.85%) and precision (79.65% vs. 77.68%). Recall values remain nearly identical (around 71%), indicating that both approaches struggle to recover all true signal hits in the presence of the intense backscattering environment. This behavior is confirmed in the bottom panels of Figure 9.

The learning curves of both models in both energy ranges confirm smooth convergence, suggesting robustness of the training procedure even in the high-noise regime.

Overall, the two encoder variants display complementary strengths: the linear encoder achieves better sensitivity at low energies, while the attention encoder provides a modest but consistent gain in precision and overall accuracy at higher energies, where the complexity introduced by backscattering requires more expressive feature representations.

In future work, hybrid or ensemble strategies could be explored to combine these advantages and improve performance across the full energy spectrum.

At this stage, a direct comparison between the AI pipeline and traditional tracking approaches is not possible, as the HerdSoftware is no longer maintained and no conventional tracking pipeline is currently available. Nevertheless, in preliminary studies on a toy model of a space experiment, we found that processing 5000 tracks with the traditional pipeline, based on the χ^2 minimization, took approximately more than one hour, whereas

the AI-based pipeline completed the same task in about 250 ms, highlighting the substantial computational efficiency of the AI approach.

The current approach also stops at node-level classification, i.e., the pattern recognition stage, without yet including the final track fitting step (e.g., linear regression for trajectory extraction). Including this in future developments will be crucial for building a full end-to-end AI tracking pipeline.

Author Contributions: Conceptualization, F.G., F.C. and M.B.; methodology, F.C. and M.B.; software, F.C.; validation, F.C.; formal analysis, F.C.; investigation, F.C.; resources, F.C. and M.B.; data curation, F.C.; writing—original draft preparation, F.C.; writing—review and editing, F.C. and M.B.; visualization, F.C.; supervision, F.G. and M.N.M.; project administration, F.G.; funding acquisition, F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Italian Research Center on High Performance Computing Big Data and Quantum Computing (ICSC), project funded by the European Union—NextGenerationEU—and National Recovery and Resilience Plan (NRRP)—Mission 4 Component 2 within the activities of Spoke 3 (Astrophysics and Cosmos Observations).

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: A special thanks for the technical support goes to G. Donvito and G. Vino from the INFN Section of Bari.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Calafiura, P.; Rousseau, D.; Terao, K. *Artificial Intelligence for High Energy Physics*; World Scientific Publishing: Singapore, 2022.
2. Khalifa, N.E.M.; Taha, M.H.N.; Hassanien, A.E.; Selim, I.M. Deep Galaxy: Classification of Galaxies based on Deep Convolutional Neural Networks. *arXiv* **2017**, arXiv:1709.02245. [CrossRef]
3. Tykhonov, A.; Kotenko, A.; Coppin, P.; Deliyergiyev, M.; Droz, D.; Frieden, J.M.; Perrina, C.; Putti-Garcia, E.; Ruina, A.; Stolpovskiy, M.; et al. A deep learning method for the trajectory reconstruction of cosmic rays with the DAMPE mission. *Astropart. Phys.* **2023**, *146*, 102795. [CrossRef]
4. Pearce, A.; Wiltschko, A.; Sanchez-Lengeling, B.; Reif, E. A Gentle Introduction to Graph Neural Networks. *Distill* **2021**, *6*, e33. [CrossRef]
5. Zhang, S.N.; Adriani, O.; Albergo, S.; Ambrosi, G.; An, Q.; Bao, T.W. The High Energy Cosmic-Radiation Detection (HERD) Facility Onboard China's Space Station. *Proc. SPIE* **2014**, *9144*, 91440X.
6. Mori, N. GGS: A Generic GEANT4 Simulation Package for Small- and Medium-Sized Particle Detection Experiments. *Nucl. Instrum. Methods Phys. Res. A* **2021**, *1002*, 165298.
7. Betti, P. The HERD Experiment: New Frontiers in Detection of High Energy Cosmic Rays. *PoS* **2024**, *TAUP2023*, 142.
8. Perrina, C.; Frieden, J.M.; Azzarello, P.; Sukhonos, D.; Wu, X.; Gascón, D.; Bai, Y.L.; Bao, T.W.; Barbanera, M.; Barbato, F.C.T.; et al. The Scintillating-Fiber Tracker (FIT) of the HERD Space Mission from Design to Performance. *PoS* **2024**, *ICRC2023*, 147.
9. Liu, X.; Adriani, O.; Bao, T.W.; Berti, E.; Betti, P.; Bottai, S. Double Read-Out System for the Calorimeter of the HERD Experiment. *PoS* **2023**, *HERD2023*, 97.
10. Fermi Gamma-Ray Space Telescope—Mission Website. Available online: https://fermi.gsfc.nasa.gov/ (accessed on 10 October 2025.
11. Grupen, C. *Astroparticle Physics*; Springer: Berlin/Heidelberg, Germany, 2020.
12. Duranti, M.; Vagelli, V.; Ambrosi, G.; Barbanera, M.; Bertucci, B.; Catanzani, E.; Donnini, F.; Faldi, F.; Formato, V.; Graziani, M.; et al. Advantages and Requirements in Time Resolving Tracking for Astroparticle Experiments in Space. *Instruments* **2021**, *5*, 20. [CrossRef]
13. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M. Relational Inductive Biases, Deep Learning, and Graph Networks. *arXiv* **2018**, arXiv:1806.01261. [CrossRef]
14. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907. [CrossRef]
15. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *arXiv* **2018**, arXiv:1706.02216. [CrossRef]

16. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903. [PubMed]
17. Battaglia, P.W.; Pascanu, R.; Lai, M.; Rezende, D.; Kavukcuoglu, K. Interaction Networks for Learning about Objects, Relations and Physics. *arXiv* **2016**, arXiv:1612.00222. [CrossRef]
18. Yeung, M.; Sala, E.; Schönlieb, C.B.; Rundo, L. Unified Focal Loss: Generalising Dice and Cross Entropy-Based Losses to Handle Class Imbalanced Medical Image Segmentation. *Comput. Med. Imaging Graph.* **2022**, *95*, 102026. [CrossRef] [PubMed]
19. Ray Tune: Hyperparameter Tuning—Ray 2.50.1. Available online: https://docs.ray.io/en/latest/tune/index.html (accessed on 10 October 2025).