Department of Physics and Astronomy
Heidelberg University

Bachelor Thesis in Physics
submitted by

David Tobias Karres

born in Bonn (Germany)

2024

Feasibility Study
of a Graph-Neural-Network-Based
Cosmic Muon Trigger for the Mu3e Experiment

This Bachelor Thesis has been carried out by David Tobias Karres at the
Physikalisches Institut Heidelberg
under the supervision of
Prof. Dr. André Schöning

Abstract

The Mu3e experiment at the Paul-Scherrer-Institute will lead the search for the lepton flavour violating muon decay $\mu^+ \to e^+e^-e^+$. The planned single event sensitivity of up to $\mathcal{O}(10^{-16})$ requires an extreme level of precision in both vertex finding and momentum resolution. Consequently, the silicon pixel tracking detector requires precise alignment. This is achieved with track-based alignment which presents a powerful form of active detector alignment. However, weak modes pose a challenge for this alignment strategy and require fundamentally different particle trajectories. This led to the proposition of utilizing cosmic ray muons for alignment at Mu3e, as it is also commonly done in other high energy physics experiments. Currently, reconstruction of cosmic muon tracks is not realized within the online filter farm. Because of that and the low incident rate of around $10\,\mathrm{Hz}$, among an overwhelming amount of background, a cosmic muon trigger is developed.

This thesis investigates the viability of a cosmic muon trigger using graph neural networks to reconstruct and classify events accordingly. Using this approach, an aceptance of background acceptance of 6.2×10^{-5} is achieved, at a cosmic muon reconstruction efficiency of 95.6%.

Zusammenfassung

Sobald das Mu3e Experiment am Paul-Scherrer-Institut in Betrieb geht, wird es die Suche nach dem leptonenzahlverletzenden Myon Zerfall $\mu^+ \to e^+e^-e^+$ anführen. Bei einer geplanten Auflösung des Verzweigungsverhältnisses von bis zu $\mathcal{O}(10^{-16})$ ist ein hohes Maß an Präzision gefordert, besonders in den Bereichen Vertexrekonstruktion und Impulsauflösung. Um diesen Anforderungen gerecht zu werden, muss der Silizium Pixel Detektor exakt ausgerichtet sein. Mit Hilfe von spurbasiertem Alignment lässt sich die geforderte Auflösung im Prinzip erreichen, jedoch bleiben sogenannte Weak Modes für diese Alignmentstratgeie unsichtbar. Um diese Fehlausrichtungsmoden zu beheben, werden grundlegend verschiedene Teilchenspuren benötigt. Deshalb soll bei Mu3e, wie auch bei anderen Experimenten zuvor, auf kosmische Myonen zurückgegriffen werden. Aktuell ist jedoch noch keine Rekonstruktion kosmischer Myonenspuren im Mu3e Auslesesystem vorgesehen. Aus diesem Grund und da kosmische Myonen nur mit einer Einfallsrate von etwa $10\,\mathrm{Hz}$ zwischen Hintergrundevents untergehen, soll ein kosmischer Myonentrigger entwickelt werden.

Diese Arbeit untersucht die Durchführbarkeit eines solchen Triggersystems unter Verwendung von Graph Neural Networks, mithilfe derer eine Rekonstruktion und Klassifikation durchgeführt werden soll. Als Resultat ergibt sich eine Akzeptanz von Untergrundframes von 6.2×10^{-5}, bei einer Rekonstrutionseffizienz kosmischer Myonen von 95.6%.

Contents

Contents

1 Introduction & Theory

The chess board is the world, the pieces are the phenomena of the universe, the rules of the game are what we call the laws of nature. The player on the other side is hidden from us. We know that his play is always fair, just and patient. But we also know, to our cost, that he never overlooks a mistake, or makes the smallest allowance for ignorance.

Thomas Henry Huxley

Experimental particle physics probes the smallest size scales possible in order to expand the knowledge about the most fundamental mechanisms in the universe. While the best current model, the *Standard Model of Particle Physics* (SM), is able to make many accurate predictions, the search for new physics beyond our current understanding is continuously driven further.

The *Mu3e experiment* at the Paul Scherrer Institute in Villigen, Switzerland does just that. Aiming at either directly observing the *lepton flavour violating* (LFV) decay $\mu^+ \to e^+ e^- e^+$ or improving current SM predictions for the likelihood of it occuring, Mu3e stands at the forefront of research of LFV decays.

The precision of experimental measurements is inherently dependent on factors, such as thermal expansion, deformations, or the mechanical accuracy of the detectors themselves. This is achieved not only through careful construction but also through active *detector alignment* strategies. One of the strategies envisioned at Mu3e is *track-based alignment*, which makes use of particle tracks. When charged particles traverse a tracking detector, they leave measureable energy deposits (hits) in the layers of the detector allowing for reconstruction of tracks.

Global deformations of the tracking detector give rise to misalignment modes invisible to track-based alignment, called *weak modes*. Identification and correction requires the utilisation of different particle tracks to the ones found at Mu3e. *Cosmic ray muons* present a viable solution due to their high momentum, even distribution across the whole detector and low scattering cross section. These properties allow them to traverse the entire detector. By reconstructing their tracks, it is possible to determine systematical deviations in the detector's alignment and correct them in software. As a result, their application in

an online alignment system for the Mu3e experiment is under investigation.

The main challenge associated with using cosmic ray muons is their low incident rate of approximately one muon per square centimeter per second. To detect these muons among the overwhelming abundance of electrons and positrons inside the detector, occuring at a decay rate of $\mathcal{O}(10^8\,\text{Hz})$, is a complex task.

While reconstruction of electron and positron tracks is already implemented, cosmic muons currently are not accounted for. This necessitates a *trigger system* capable of deciding whether an event contains a cosmic muon, and if so, retaining it for alignment purposes.

By efficiently reconstructing the tracks of cosmic muons, a trigger system could quickly discrimminate between events containing a cosmic muon. This thesis explores an approach for a cosmic muon trigger based on *Graph Neural Networks* (GNNs). GNNs are a subclass of *machine learning* models in the field of *geometric deep learning* and allow to operate on graph-structured datasets. *Graphs* consist of a set of *nodes* and *edges* connecting nodes. By representing particle tracks as graphs it is possible to capture the spatial relations of the hits and thus provide a natural way of working with tracking detector data.

The GNNs used during the course of this thesis assign an *edge score* to each edge, reflecting the likelihood that the nodes connected by the edge represent two subsequent hits of a particle track.

The aim of this work is to evaluate the performance and feasability of edge-classifying GNNs in the context of a cosmic muon trigger system for Mu3e. By reconstructing tracks from simulated data at Mu3e, the performance in terms of *track reconstruction efficiency* and *background acceptance* are evaluated.

The thesis is divided into three main sections. Chapter *Introduction & Theory* contains the theoretical background of the physics needed for this thesis (Section 1.1) as well as a section on the Mu3e experiment (Section 1.2) and an introduction to GNNs in the light of particle track reconstruction (Section 1.3).

The chapter *Methodology* explains the simulated data samples used (Section 2.1), gives an overview of the GNN tracking pipeline employed (Section 2.2) and finishes on the metrics used for evaluating the models (Section 2.3). Finally, the chapter *Results* presents the findings from inferring different GNN models on the simulated data samples (Section 3.1.1) as well as the evaluation of the track reconstruction of cosmic muon tracks (Section 3.1.2). This includes the comparison of acceptances, fake track ratios and reconstruction efficiencises for different combinations of networks and inference data. The last section of the thesis summarizes the work done, draws a conclusion on the feasibility of a GNN-based cosmic muon trigger and gives an outlook on possibilities to expand on this thesis (Section 3.2).

1.1 Theoretical Background

1.1.1 The Standard Model of Particle Physics

The Standard Model of Particle Physics (SM) is the most successful *Quantum Field Theory* (QFT) of our time. Developed throughout the 20th century, it describes the fundamental particles and interactions between them through an exchange of *gauge bosons*. The SM has made numerous predictions that have been confirmed experimentally, with one of the most notable ones being the discovery of the *Higgs boson* at the *Large Hadron Collider* (LHC) in 2012 [5].

However the SM does not explain some phenomena such as gravity, the non-zero neutrino mass and dark matter. Because of that, experimentally probing nature for physics beyond the SM is especially intriguing as it holds the potential for discovering new physics.

The two basic types of fundamental particles are *fermions* and *bosons*. The SM includes twelve different fermions (and their corresponding *antiparticles*), four types of *gauge bosons* and one *scalar boson*.

Fermions are spin-1/2 particles that follow *Fermi-Dirac statistics*, thus obeying the *Pauli exclusion principle*, which states that two fermions in a system cannot occupy the same quantum state simultaneously.

Fermions are the constituents of matter and are further classified into *leptons* and *quarks*. Both leptons and quarks come in three generations that differ in mass, with the first generation being the lightest and the third generation the heaviest.

Each lepton generation consists of a charged lepton: the *electron* (e^-), *muon* (μ^-) and *tauon* (τ^-); and a neutral lepton: the *electron neutrino* (ν_e), *muon neutrino* (ν_μ) and *tau neutrino* (ν_τ). The corresponding antiparticles are the *positron* (e^+), *anti-muon* (μ^+), *anti-tau* (τ^+) and *anti-neutrinos* ($\bar{\nu}_{e,\mu,\tau}$). All leptons interact via the *weak interaction* while only the charged leptons also interact *electromagnetically* due to their electric charge. The three generations of quarks all consist of an up type and a down type quark: up (u) and down (d), charm (c) and strange (s), top (t) and bottom (b). Quarks interact via the electromagnetic and weak interaction and, due to their *color charge*, also through the *strong interaction*. Because of *color confinement*, quarks can not exist in isolation, meaning they *hadronize* immediately after creation to form *mesons* (quark-antiquark pairs) or *baryons* (quark or anti-quark triplets) which include neutrons and protons, the constituents of atomic nuclei.

Bosons on the other hand are particles with integer spin and follow *Bose-Einstein statistics*. They are responsible for mediating the fundamental interactions with the exception of gravity (potentially mediated by the hypothetical *graviton*). In the SM, there are four kinds of gauge bosons (also called *vector bosons*) and one scalar boson. All gauge bosons have spin 1 and mediate one of the fundamental forces.

The massless *photon* (γ) carries the electromagnetic force. Also massless, eight *gluons* with different color configurations carry the strong force between quarks. The weak interaction is mediated by the electrically charged W^+ or W^- bosons and the neutral Z^0 boson. Finally, the only scalar boson, the Higgs boson (H^0) is a quantum excitation of the *Higgs field*, a scalar field permeating the universe. Through the *Higgs mechanism* particles acquire their mass, by interacting with the Higgs field. The stronger the interaction, the heavier the particle. Figure 1.1 shows all particles included in the SM as well as the hypothetical graviton.
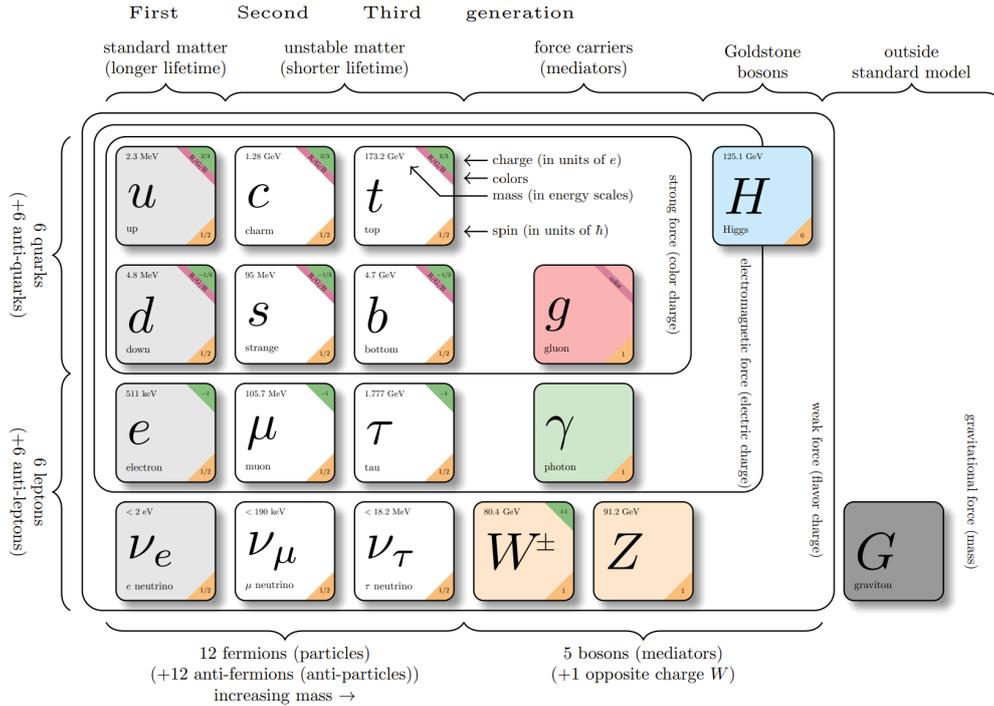

Figure 1.1: The Standard Model of Particle Physics. Figure taken from [3].

1.1.2 Muon Decay in the Standard Model

Muons mostly decay via the so-called *Michel decay*, which occurs through the weak interaction, mediated by the exchange of a W boson. In this process, a muon (μ^-) decays into an electron (e^-), a muon neutrino (ν_μ), and an anti-electron neutrino ($\bar{\nu}_e$). The lowest order *Feynman diagram* of this process is shown in Figure 1.2.

Similarly, an anti-muon (μ^+) decays into a positron (e^+), an anti-muon neutrino ($\bar{\nu}_\mu$), and an electron neutrino (ν_e). The *lepton flavor number* remains conserved in both decays.

With a branching ratio of close to 100% [20], the Michel decay represents the overwhelming majority in muon decays. The *branching ratio* $BR(j)$ of a decay mode j is defined as the *decay rate* of the mode $\Gamma(j)$ divided by the total decay rate Γ.

$$BR(j) = \frac{\Gamma(j)}{\Gamma} \tag{1.1.1}$$

$$\mu^- \quad \rightarrow \quad e^- \bar{\nu}_e \nu_\mu \qquad\qquad \mu^+ \quad \rightarrow \quad e^+ \nu_e \bar{\nu}_\mu \qquad\qquad (1.1.2)$$
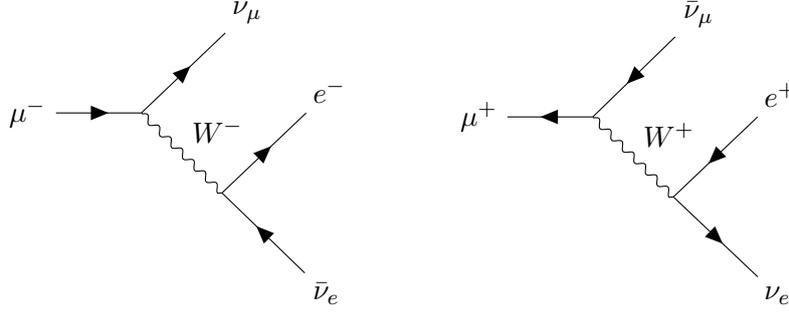
Figure 1.2: Feynman diagrams of the Michel decay.

There also are two sub-modes where the emitted electron/positron emits a photon which in turn might even produce an electron-positron pair called *radiative Michel decay* (see Figure 1.3).

$$\mu^- \quad \rightarrow \quad e^- \bar{\nu}_e \nu_\mu \gamma \qquad\qquad \mu^+ \quad \rightarrow \quad e^+ \nu_e \bar{\nu}_\mu \gamma \qquad\qquad (1.1.3)$$
$$\mu^- \quad \rightarrow \quad e^- \bar{\nu}_e \nu_\mu e^+ e^- \qquad\qquad \mu^+ \quad \rightarrow \quad e^+ \nu_e \bar{\nu}_\mu e^+ e^- \qquad\qquad (1.1.4)$$
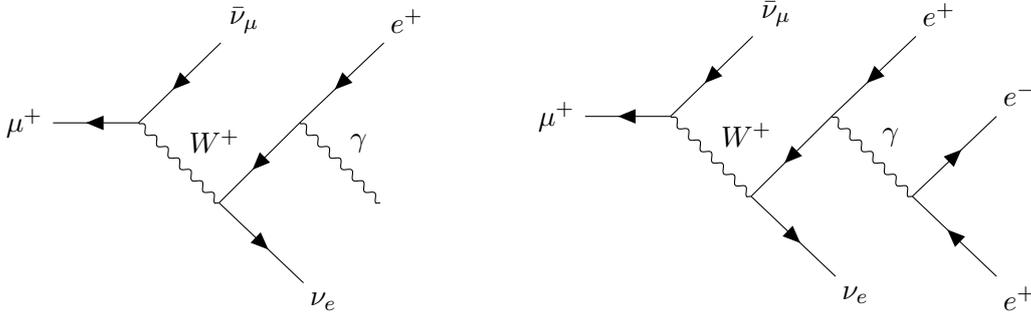
Figure 1.3: Feynman diagrams of radiative Michel decays.

1.1.3 Lepton Flavour Violation

In the SM the conservation of lepton flavour is expected at tree level. This causes lepton flavour violating (LFV) decays such as $\mu^+ \rightarrow e^+ e^+ e^-$ to be heavily suppressed with $BR(\mu \rightarrow eee) < 10^{-55}$.

The lepton number is defined as the difference between the number of leptons and the number of antileptons. This quantity is normally conserved for every generation of leptons at all vertices. However higher order *loop diagrams* allow LFV decays via *neutrino mixing*. The leading order Feynman diagram is shown in Figure 1.4. The process of neutrino mixing (also called *neutrino oscillation*) is described by the *Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix*, which is the analogue of the *Cabibbo-Kobayashi-Maskawa (CKM) matrix*,

describing the interaction strength of quark mixing [5]. A deeper theoretical discussion on this topic is not relevant for the following sections and is thus omitted.

Directly observing an LFV decay such as $\mu^+ \rightarrow e^+e^-e^+$ would provide an indication for additional decay channels impossible in the SM and would thus point towards *physics beyond the SM*. A theoretical mechanism allowing LFV decays is via *super-symmetric particles* (see Figure 1.5), these kinds of particles, however, have not been confirmed so far and remain hypothetical [14].
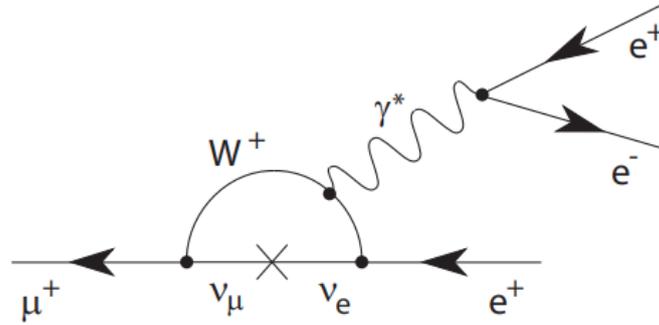

Figure 1.4: Feynman diagram of the decay $\mu^+ \rightarrow e^+e^-e^+$ via neutrino mixing. Figure taken from [4].
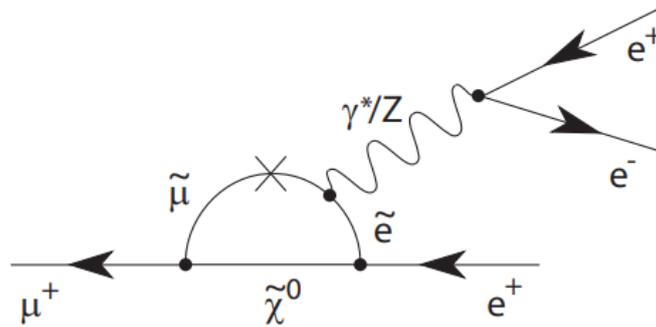

Figure 1.5: Possible Feynman diagram of the decay $\mu^+ \rightarrow e^+e^-e^+$ via super-symmetric particles. Figure taken from [4].

1.1.4 Particle Interactions with Matter

Charged particles propagating inside a material experience *Coulomb interactions* with atoms. Because energy is deposited in form of exictation or ionization of the involved atoms, the incident particle slows down and gets deflected. This fact is used in particle tracking detectors to record hits in multiple detector layers to then reconstruct the trajectory.

Heavy charged particles

Heavy charged particles, i.e. particles with $m \gg m_e$ such as muons, mainly scatter off of electrons of the material. These inelastic collisions cause excitations or ionizations of the materials atoms. The energy loss $\frac{dE}{dx}$ of a singly charged particle with velocity $v = \beta c$,

traveling a distance x into a material of electron number density n, with atomic number Z, can be described by the *Bethe-Bloch Formula* [5]:

$$\frac{\mathrm{d}E}{\mathrm{d}x} \approx -4\pi\hbar^2 c^2 \alpha^2 \frac{nZ}{m_e v^2} \left[\ln\left(\frac{2\beta^2\gamma^2 c^2 m_e}{I_e} \right) - \beta^2 \right]. \tag{1.1.5}$$

Where c corresponds to the speed of light, $\hbar$ to the reduced Planck constant, α to the fine structure constant, m_e to the electron rest mass, $\gamma = \frac{1}{\sqrt{1-\beta^2}}$ to the Lorentz factor and I_e to the effective ionisation potential of the material, averaged over all atomic electrons.
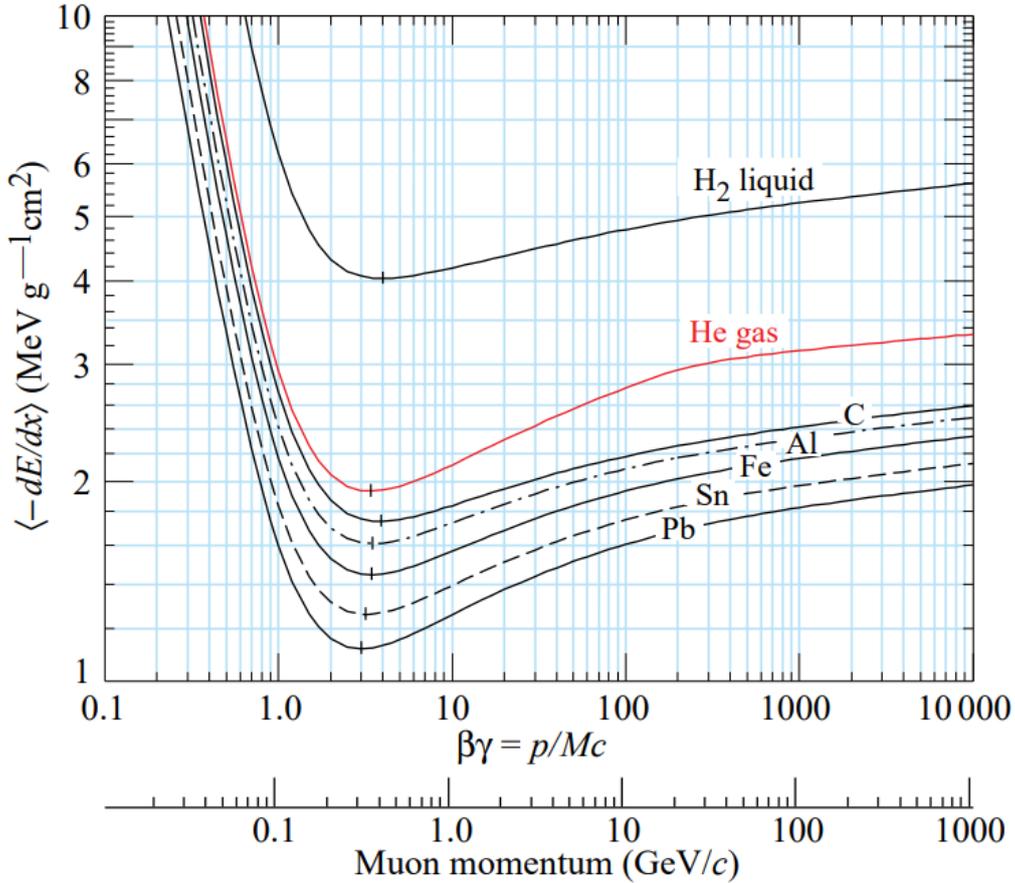

Figure 1.6: Mean energy loss inside of different materials at different particle energies. Figure taken from [20].

The energy loss clearly depends on the energy of the particle as well as the material it is traversing. Figure 1.6 shows the energy loss inside of multiple mediums at different particle energies. Going from low to high momentum, the energy loss first rapidly decreases until the minimum of around $\beta\gamma \approx 2.5$ and then increases logarithmically. Particles at or near the minimum are called minimum ionizing particles and lose the least amount of energy in the material. In addition to energy loss due to Coulomb interactions the effects of additional *radiative losses* become more dominant at higher energies. In the case of muons this begins at $\mathcal{O}(100\,\mathrm{GeV})$[9].

Electrons

The much lighter electrons on the other hand experience significant radiative effects at much lower energies. The process of electrons losing some of their energy by emitting a photon when interacting with the Coulomb field of the nucleus is called *Bremsstrahlung*. For electrons, Bremsstrahlung starts to dominate at $\mathcal{O}(10\,\text{MeV})$. The radiative energy loss is directly proportional to the electron energy

$$-\frac{\mathrm{d}E}{\mathrm{d}x} = \frac{E}{X_0},\tag{1.1.6}$$

where X_0 is the *radiation length*, i.e the distance inside the material after which an electron has lost $\approx 37\%$ of its original energy.

1.1.5 Cosmic Ray Muons

The earth is constantly hit by highly energetic ($E < \mathcal{O}(100\,\text{TeV})$) particles originating from outer space [10]. These cosmic rays are comprised of around 99% fully ionized nuclei (90% protons, 9% α-particles and 1% heavier nuclei) and 1% electrons.
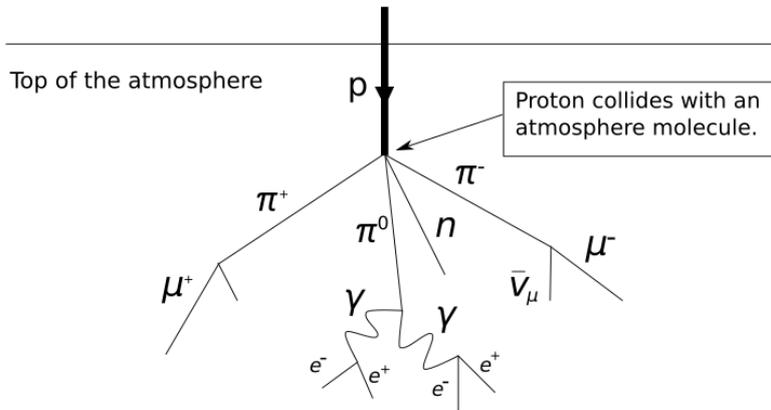

Figure 1.7: Interactions of cosmic ray particles with atmospheric molecules producing muons. Figure taken from [22].

Collisions with particles in the upper atmosphere at an altitude of around 15km may produce secondary particles such as *pions* or *kaons* (see Figure 1.7). These secondaries are highly unstable and quickly decay further. Charged pions and kaons in particular constitute the main source of *cosmic ray muons*. Table 1.1 shows the most prominent decays involved in cosmic ray muon production and their respective branching ratios.

Particle	Decay products	Branching ratio [%]
π^-	$\mu^- \bar{\nu}_\mu$	99.98770 ± 0.00004
π^+	$\mu^+ \nu_\mu$	
K^-	$\mu^- \bar{\nu}_\mu$	63.56 ± 0.11
	$\pi^0 \pi^- \rightarrow \pi^0 \mu^- \bar{\nu}_\mu$	20.67 ± 0.08
K^+	$\mu^+ \nu_\mu$	63.56 ± 0.11
	$\pi^0 \pi^+ \rightarrow \pi^0 \mu^+ \nu_\mu$	20.67 ± 0.08

Table 1.1: Most relevant decay channels involved in the production of cosmic ray muons. Values taken from [20].

Losing about $2\,\mathrm{GeV}$ on their descent, the mean energy of cosmic muons at sea level is $\approx 4\,\mathrm{GeV}$ [20]. The momentum spectrum lies almost flat below $1\,\mathrm{GeV}$ and then falls off gradually around $10 - 100\,\mathrm{GeV}$ (see Figure 1.8).
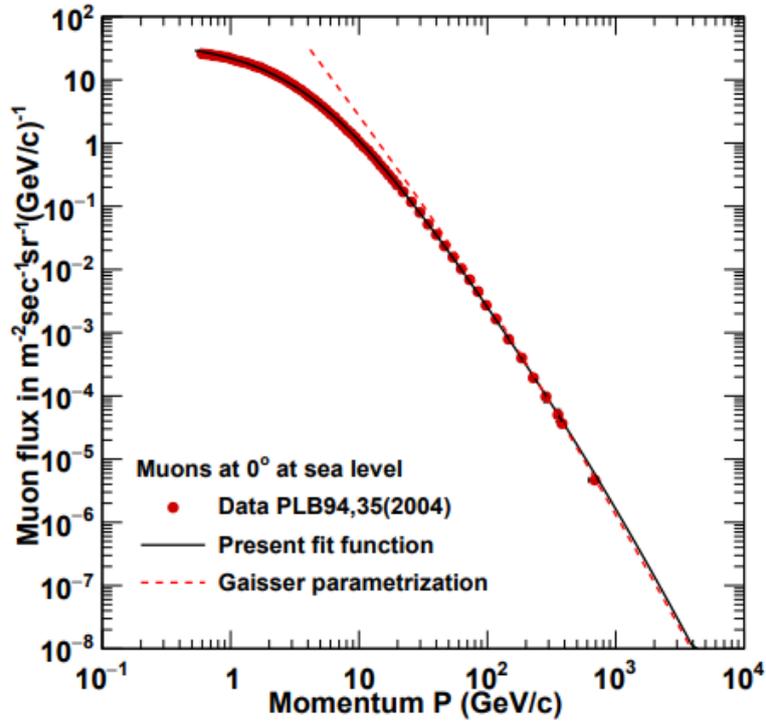

Figure 1.8: Momentum distribution of cosmic muons, measured at sea level and $0°$ zenith angle. Figure taken from [10].

The expected muon rate of vertical, $E > 1\,\mathrm{GeV}$ muons at sea level is $I \approx 70\,\mathrm{m}^{-2}\,\mathrm{s}^{-1}\,\mathrm{sr}^{-1}$. Muons with energies around $3\,\mathrm{GeV}$ show a $\cos^2(\theta)$ angular distribution, with θ being the zenith angle.

1.2 The Mu3e Experiment

The Mu3e experiment [14] at the *Paul Scherrer Institute* (PSI) in Villingen, Switzerland searches for the LFV decay of a muon into three electrons. This decay is forbidden in the SM on tree level and heavily suppressed to $BR = \mathcal{O}(10^{-55})$ when considering neutrino mixing in higher order diagrams. In particular the goal is to observe the process $\mu^+ \to e^+e^-e^+$ either directly, if the branching ratio is larger than 10^{-16}, or to set a new lowest limit for the branching ratio of $< 10^{-16}$ at the 90% confidence level. With an expected reconstruction efficiency of 20%, more than 10^{17} muons have to be stopped in the detector. Additionally, any background has to be supressed below the targeted accuracy of 10^{-16} [14].

The experiment is planned to run in two phases. In phase I, the $\pi E5$ muon beam at PSI with an intensity of 10^8 muons per second will be used to achieve an event sensitivity of $2 \cdot 10^{-15}$. Phase II is envisioned to use the high-intensity muon beam (HiMB), which is currently under development, with a projected intensity of up to $2 \cdot 10^9$ muons per second. With that the single event sensitivity will be enhanced by a factor of 20 to reach the ultimate goal of 10^{-16} [14].

If successful or by observing the decay, Mu3e will improve the current limit set by SIN-DRUM (1988) by four orders of magnitude [2].

1.2.1 Signal and Background

In order distinguish *signal* from *background*, energy and momentum conservation are considered, leading to the following criteria:

1. Since the muons are stopped in a target and decay into i particles at rest in the laboratory frame *3-momentum conservation* is satisfied if:

$$|\mathbf{P_{tot}}| = \left|\sum_i \mathbf{p}_i\right| = 0. \tag{1.2.1}$$

2. The *invariant mass* must be equal to the muon mass, which is the sum of the energies of the decay particles in this instance:

$$m_{inv} = m_\mu = \left|\sum_i p_i\right| = \sum_i E_i. \tag{1.2.2}$$

3. All decay products lie within the same plane.

4. All decay particles are emitted simultaneously.

An exemplary topology that satisfies these criteria is shown in Figure 1.9
Such a topology, however, can be mimicked by processes other than the desired LFV decay. With these backgrounds being nearly indistinguishable from a LFV event signal, precise vertex reconstruction along with high momentum and time resolution is essential.
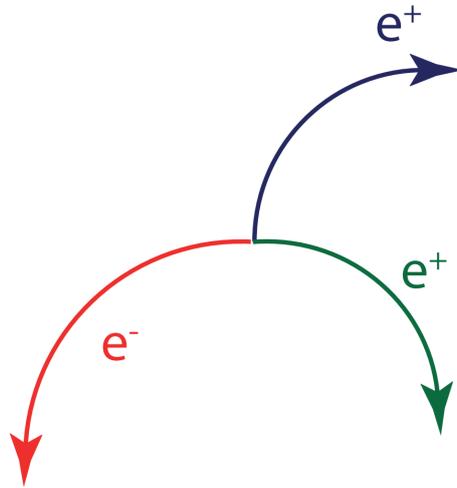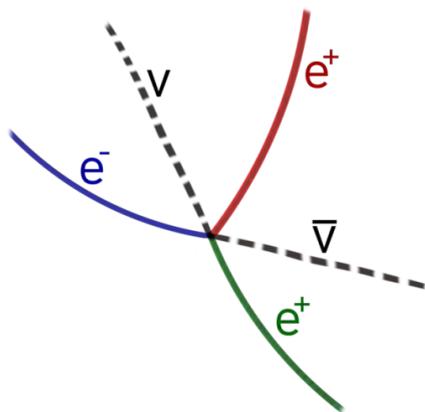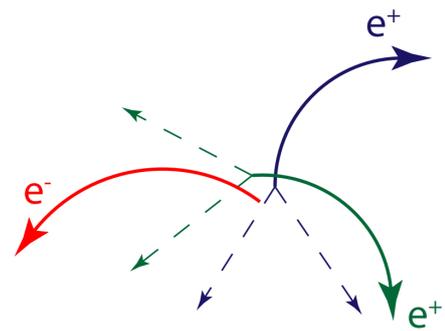
Figure 1.9: Topology of the signal decay $\mu^+ \rightarrow e^+ e^- e^+$. Figure taken from [11].

(a) Topology of an internal conversion event.

(b) Exemplary topology of an accidental background event.

Figure 1.10: The two main sources of background event at the Mu3e experiment. Figure taken from [11].

Internal Conversion

Radiative muon decays ($\mu^+ \rightarrow e^+ e^- e^+ \nu_e \bar{\nu}_\mu$) closely resemble the signal event, with the difference being the two additional neutrinos (shown in Figure 1.10a). Although neutrinos cannot be detected by the Mu3e detector directly, they carry away some of the momentum. Therefore, Equations 1.2.1 and 1.2.2 no longer hold when considering only the electrons momenta. As is clear from Equation 1.2.2 an excellent mass reconstruction resolution is needed. The required sensitivity of $2 \cdot 10^{-15}$ for phase I can only be achieved, if the mass reconstruction resolution is better than $1\,\text{MeV}$ [14].

Combinatorial Backgrounds

Combinatorial backgrounds arise from events that appear similar to the signal decay but consist of multiple overlapping processes like in Figure 1.10b. The Michel decay ($\mu^+ \rightarrow e^+ \nu_e \bar{\nu}_\mu$) by far is the most common decay channel of the muon, however through these decays only positrons are created. Consequently combinatorial backgrounds occurs when a Michel decay is overlayed with a process producing negatively charged particles, such as *Bhabha scattering*, *photon conversion* or *Compton scattering*.
In the case of a Michel decay overlapping with Bhabha scattering, two of the three particles even originate from the same point in space. To effectively distinguish between these events, it is clear that good vertex and time resolution are needed.

1.2.2 Detector Concept

As discussed in the previous section, an excellent vertex and momentum resolution is crucial. These are mostly limited by the spatial resolution of the detector itself and by multiple Coulomb scattering in the detector material. Since the energies of the decay products of the stopped muons are limited to half the muon mass ($\approx 53\,\text{MeV}$), these particles are highly susceptible to multiple Coulomb scattering. The pixel sensors, with a pixel size of $80\,\mu\text{m}$, make multiple scattering the dominant limiting factor.
The scattering angle due to multiple Coulomb scattering can be modeled by the following equation:

$$\theta_0 = \frac{13.6\,\text{MeV}}{\beta c p} Z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \left(\frac{x}{X_0} \right) \right), \tag{1.2.3}$$

where p, βc and Z correspond to the particle's momentum, velocity and charge number, respectively. A schematic view of this is shown in Figure 1.11. The term x/X_0 represents the thickness of the medium in units of the radiation length [20]. From this, it is clear that the thickness of the tracking sensors must be as thin as possible. Additionally, the detector has to be able to handle the high event rates envisioned for the Mu3e experiment, hence only leaving ultrathin silicon pixel sensors as a viable option.
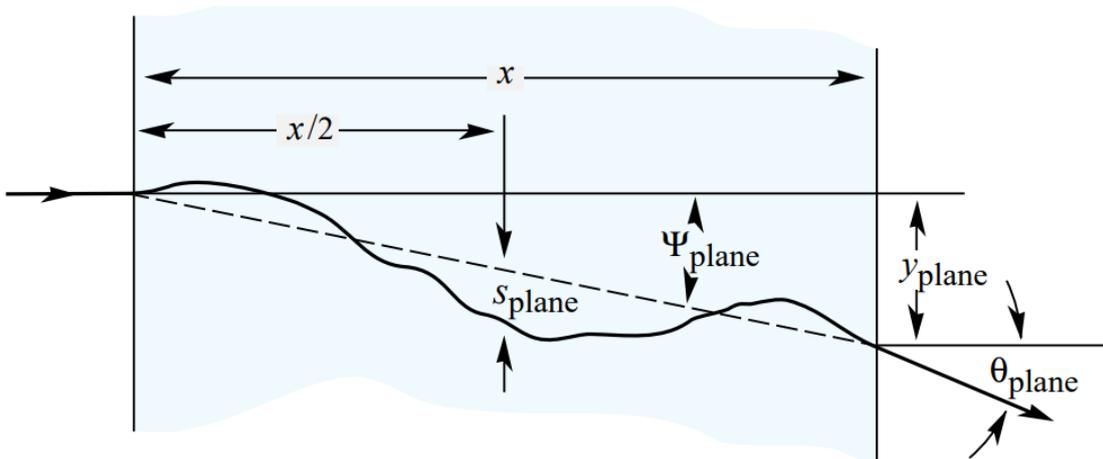
Figure 1.11: Schematic view of multiple Coulomb scattering of a charged particle travers-
ing a material. The trajectory is altered by an offset y and scattering angle
θ through a series of scattering events. Figure taken from [20].

The momentum of a charged particle is determined by measuring its bending radius in a
magnetic field. The *Lorentz force*

$$\mathbf{F}_{\mathrm{L}} = q\mathbf{v} \times \mathbf{B} \qquad (1.2.4)$$

acts on a particle with charge q and velocity $\mathbf{v}$ in a magnetic field $\mathbf{B}$. With the force
being perpendicular to the particle's velocity vector, the charged particle follows a *helical
path*. From the radius ρ of the helix the *transverse momentum* p_{T} of the particle can be
determined via

$$p_{\mathrm{T}}[GeV c^{-1}] \approx 0.3B[\mathrm{T}]\rho[\mathrm{m}]. \qquad (1.2.5)$$

To achieve good momentum resolution, the particle tracks have to be reconstructed as
accurately as possible to precisely calculate its bending radius.

Detector Geometry

The Mu3e detector setup consists of a cylindrical *silicon tracking detector*. In order to
ensure combinatorial background suppression, two kinds of timing detectors - *scintillating
tiles* and *sctintillating fibres* - are used [14].
The whole detector is divided into a *central station* situated around the *hollow double cone
target* as well as two *recurl stations*, which are placed up- and downstream next to the
central station respectively.
The central station comprises two double layers of *high voltage - monolithic active pixel
sensors* (HV-MAPS) with a scintillating fibre layer between them. The outer double layer
is extended up- and downstream and joined with a layer of scintillating tiles to form both
recurl stations. This is shown schematically in Figure 1.12. The length of the detector is
$1200\,\mathrm{mm}$ with a diameter of $180\,\mathrm{mm}$.
To make the momentum reconstruction possible, the whole detector is placed inside a
solenoidal magnetic field with $B = 1\,\mathrm{T}$.
The smallest units of the tracking detector are thinned down HV-MAPS, called *MuPix*.
They measure $50\,\mu\mathrm{m}$ in thickness with a radiation length of $x/X_0 \approx 0.115\%$ per layer, to
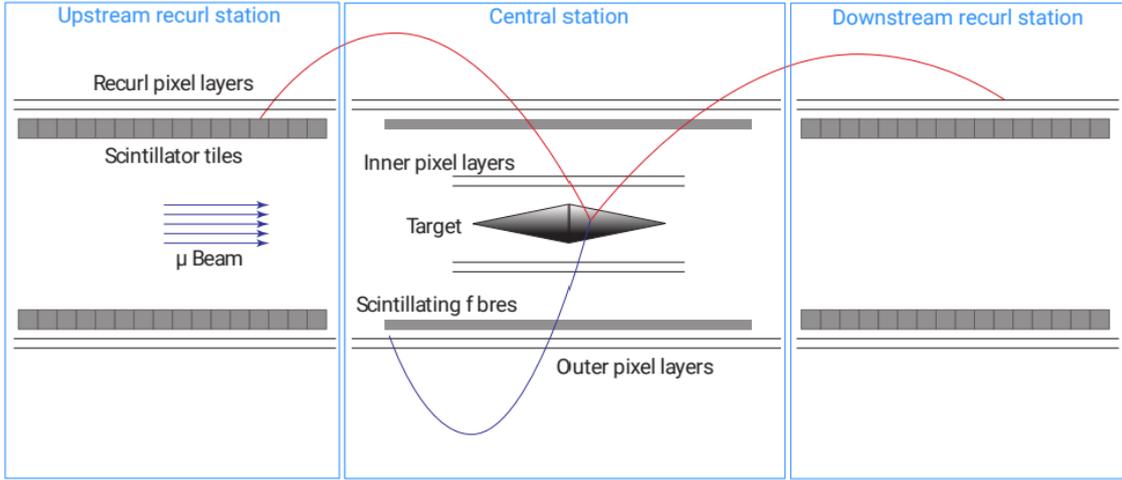
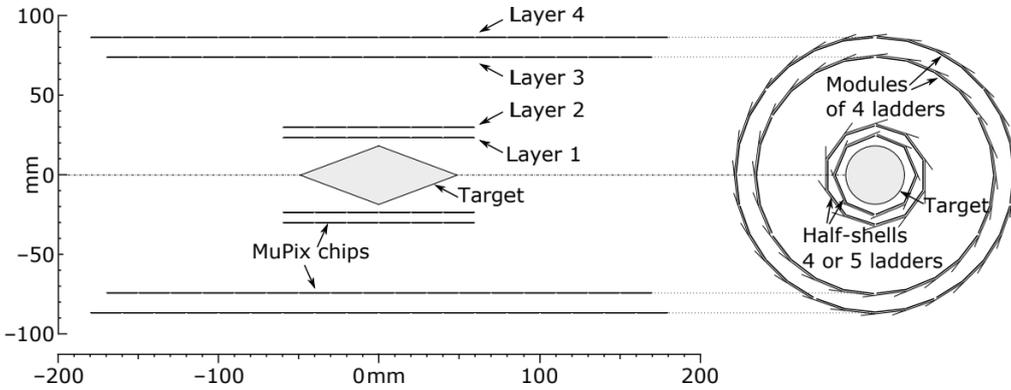Figure 1.12: Schematic of the Mu3e detector setup. Figure taken from [11]

Figure 1.13: Schematic illustration of the tracking detector. Figure taken from [14].

reduce multiple scattering and provide an active area of $20.48 \times 20.00\,\text{mm}^2$. With a pixel size of $80 \times 80\,\mu\text{m}^2$ this corresponds to 256×250 pixels.

Bonded to *high density interconnect* (HDI) circuits, multiple MuPix sensors form a ladder. The individual layers are constructed by building modules or half-shells from the ladders as is illustrated in Figure 1.13. Layers are assembled so that there is a lateral overlap of 0.5mm. This not only increases acceptance but also provides additional hits for track reconstruction.

Layer	1	2	3	4
Number of modules/half-shells per layer	2 half-shells	2 half-shells	6 modules	7 modules
Number of ladders per layer	8	10	24	28
Number of sensors per ladder	6	6	17	18
Ladder length [mm]	124.7	124.7	351.9	372.6
Minimum radius [mm]	23.3	29.8	73.9	86.3

Table 1.2: Geometrical parameters of the Mu3e tracking detector. Layers are counted from innermost to outermost. The radius is the distance of a MuPix sensor to the beam line. Based on [14].
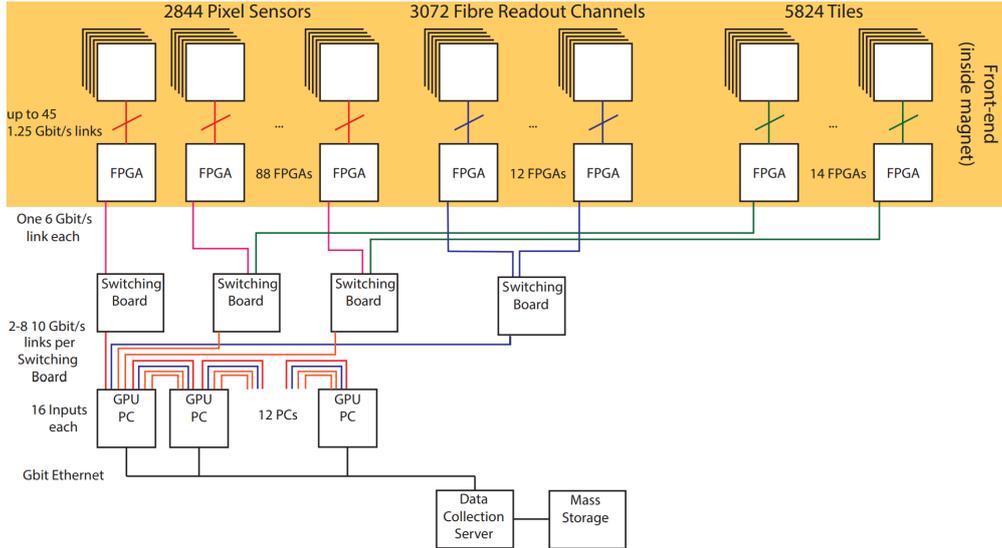
Figure 1.14: The Mu3e readout scheme. Figure taken from [14].

Detector Readout & Data Aqusition

In phase I, at a muon stopping rate of 10^8 muons per second, the raw hit data produced by the tracking and timing detectors is expected to exceed 80 GBit/s [8]. Permanently storing these amounts of data presents a major challenge and requires immense computational capacity and storage space. This is not only impractical but also expensive. Consequently the data needs to be filtered first before transferring it to permanent storage. The Mu3e *data acquisition* (DAQ) system performs an online reconstruction and filters out background events, thus mostly keeping signal candidate events.

The front-end reads out hit positions and timing data from the detectors using an array of *front-end field-programmable gate arrays* (FPGAs). During this process, the data is sorted chronologically and organized into 64 ns time frames. Next, the datastreams are directed to the *switching boards*, merging data from the tracking and timing detectors. After that the online reconstruction is performed on 12 DAQ PCs, comprising a *graphics processing unit* (GPU) and an FPGA each. While the FPGA performs a first hit selection, the GPU takes selected data and runs track- and vertex reconstruction. For track reconstruction, only short tracks are considered, i.e. tracks with four hits, going outwards from the target. Because of that, cosmic muons cannot be reconstructed and need a separate system. In the next step, candidate events are selected for storage, while the rest is discarded. With this, the datastream is reduced to <100 Mbit/s [14]. The complete readout scheme is shown in Figure 1.14.

1.2.3 Alignment

Because the Mu3e experiment relies on high precision position measurements, each active detector part needs to be assembled and *aligned* to a very high accuracy.

The *misalignment* of each pixel inside a sensor is determined by the manufacturing process. In this case, the tolerances are expected to be $< \mathcal{O}(100\,\text{nm})$ which is negligible in

comparison to the other sources of misalignment. In addition to that, the whole detector is subject to deformations of all sort, such as bending, twisting or shifting. Furthermore, thermal effects and the magnetic field itself have an influence on the detector. Due to the low material budget, the detector becomes more sensitive with respect to these forms of misalignment.

Therefore, alignment requires the knowledge of the positions, orientations and deformations of all active detector parts to calculate and apply correctional transforms to the data in software.

Alignment Approaches at Mu3e

There are two methods of alignment planned for the Mu3e experiment. The first one is a *position monitoring system* containing multiple high-resolution cameras tracking alignment marks on larger detector parts. Such a setup allows to obtain the relative positions of the recurl stations to the central station. Nonetheless, misalignment on smaller scales cannot be detected by this method. This is done by the second alignment approach: *track-based alignment*. As the name suggests, individual particle tracks are used to acquire correctional parameters. This poses a minimization problem, where the residuals, i.e. the difference between measured hits and the fit prediction, are minimized by varying track and alignment parameters.

Weak Modes

Track parameters in the fitting process are calculated by minimizing the following χ^2 function [11]:

$$\chi^2(\mathbf{q}_j, \mathbf{p}) = \sum_j^{\text{tracks}} \sum_i^{\text{hits}} \left(\frac{m_{ij} - f(\mathbf{q_j}, \mathbf{p})}{\sigma_{ij}} \right)^2 . \tag{1.2.6}$$

In this instance $\mathbf{q}_j$ represents the local track parameters, $\mathbf{p}$ the global parameters, m_{ij} is a hit measurement, f is the prediction describing the measurements and σ_{ij} is the measurement uncertainty.

The obvious choice of trajectories used for track-based alignment are particle tracks originating from the target. Some misalignment modes leave the χ^2 defined above unchanged and are thus called *weak modes*. The trivial case is translation in space, where the whole detector is moved thus allowing to fit a track with the same χ^2 to the original detector position. Weak modes mostly appear as global deformations of the tracker as shown in Figure 1.16.

When the detector is deformed along a weak mode, the true track of the particle scores a worse χ^2, compared to fitting the track in an undeformed detector. However, it possible to reconstruct a track with the same χ^2 as the true track in an undeformed detector (see Figure 1.15). Consequently one cannot distinguish whether the track is correctly reconstructed or not.

In order to identify and suppress weak modes, tracks with a different *topology* have to be used.
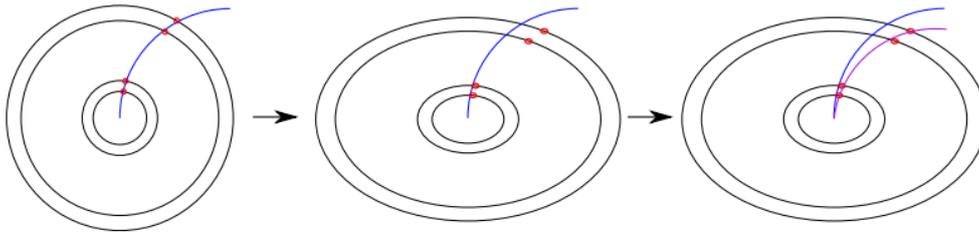
18

Figure 1.15: Schematic illustration of the Mu3e tracking detector in the transverse plane. The red markers correspond to particle hits inside the tracking detector, the physical particle track in blue and a reconstructed track including the deformation in purple. The ideal, perfectly round detector is shown on the left with the true track of the particle in blue. When introducing an elliptical deformation (center), the χ^2 of the true track is worse than without the deformation. However the reconstruction algorithm can fit a track (purple) with the same χ^2 as the true track as seen in the rightmost picture. Figure taken from [11].

Cosmic Ray Muons for Alignment

As discussed in Section 1.1.5, cosmic ray muons (also called cosmic muons or cosmics in short) posses energies far larger than the electrons found in the Mu3e experiment. Because of that, cosmics are able to traverse the entire detector with almost no deflection in the magnetic field and experience virtually no multiple scattering, thus allowing for precise track reconstruction. Furthermore, due to them originating from outside the detector, each muon hitting the detector leaves at least two hits per layer, one on the entry and one on the exit. Additionally, cosmics are able to connect detector parts with a track, that are normally not connected. The most important advantage of cosmic muon tracks is that, because of their fundamentally different track topology, weak modes can be identified (see Figure 1.17). Another advantage of using cosmic muons for alignment is their even distribution across the whole detector. When performing alignment with electrons and positrons from the target, there are far more tracks inside of the central station than in the recurl stations. This leads to worse statistics and hence alignment in the recurl stations. Cosmics on the other hand allow for alignment across the entire detector. Even though cosmics provide great value for alignment and weak mode identification in particular, they only occur at a relatively low rate compared to the beam intensity. The cosmic muon rate present at the Mu3e detector is $\mathcal{O}(10\,\mathrm{Hz})$ while the decay particle rate will be up to $\mathcal{O}(10^8\,\mathrm{Hz})$. Considering that the readout data is split into 50 ns bins, i.e. 2×10^7 frames per second, this means that only about one in 10^6 frames contain a cosmic muon. In addition to that, a large number of cosmics to improve statistics in alignment is desirable. These circumstances call for a *trigger system* that is able to identify frames containing a cosmic muon while remaining computationally inexpensive.

This thesis will explore the possibility of exploiting graph neural networks for this task which are discussed in the next section.
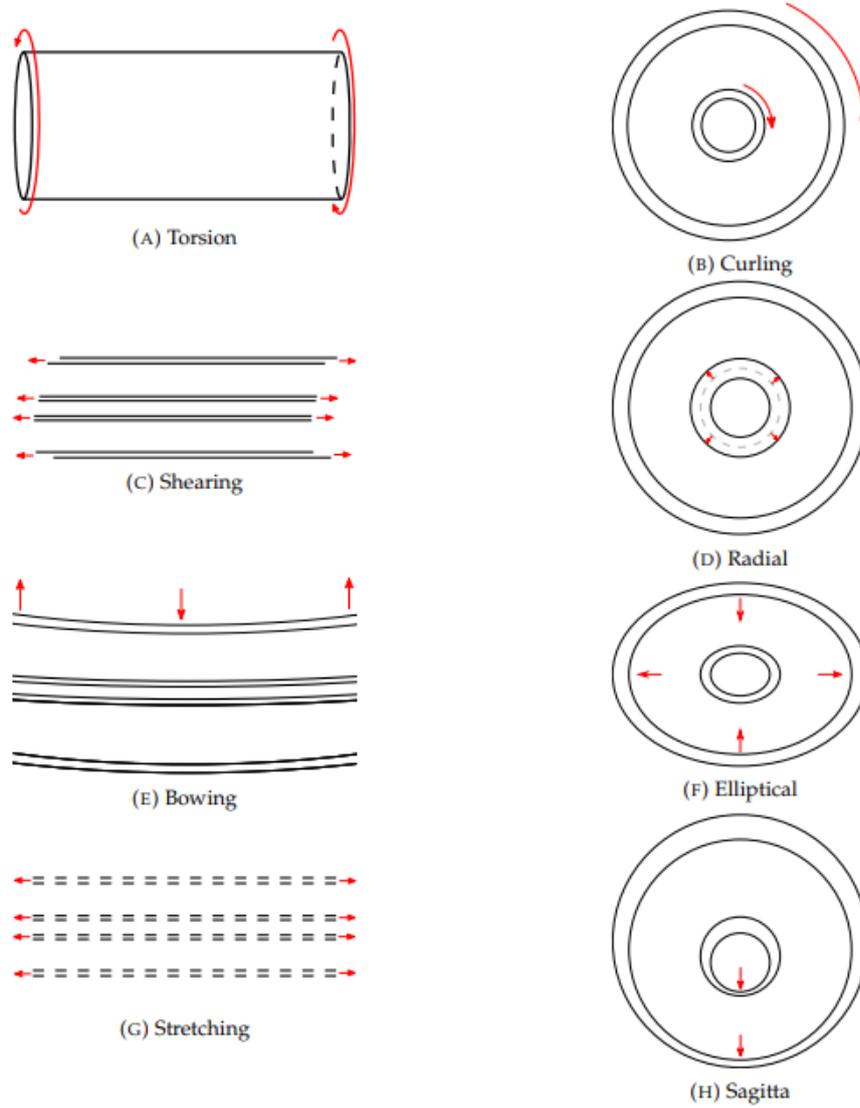
Figure 1.16: Possible weak modes for the Mu3e detector. The figures on the left show the r-z-plane and the figures on the right the x-y-plane. Figure taken from [11].
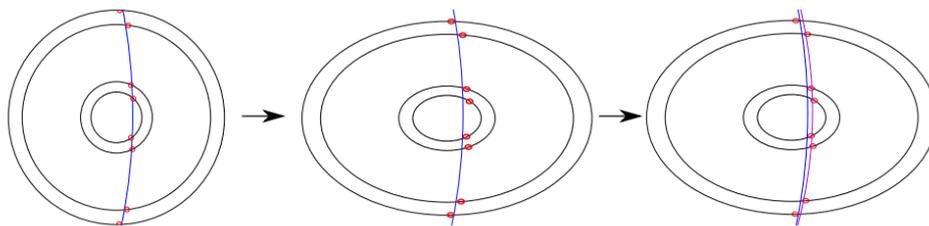

Figure 1.17: A cosmic muon traversing an elliptically deformed detector. The χ^2 of the reconstructed track is different than the original true track. Figure taken from [11].

1.3 Graph Neural Networks for Particle Tracking

Nowadays, high energy physics is impossible to imagine without *machine learning* (ML) methods. Among other applications, ML based algorithms find use in the context of charged particle track reconstruction (*tracking*). Developments of recent years have identified *graph neural networks* (GNNs) as a promising ML solution for track finding.

The following chapter will give an introduction to GNNs in the context of tracking and explain the possible advantages associated with encoding hits of charged particles inside of a tracking detector as graph objects.

1.3.1 Graph-structured Data

There are many instances in which *graphs* can be used to describe a set of objects and their connections. Examples for graph structured data can be found in social networks, transportation networks but also in computer vision for example.

A graph represents the relations between a set of *nodes* by connecting them with *edges*. The data encoded in a graph can be divided into three levels of information:

1. Node attributes: e.g. spatial coordinates, number of neighbours,...

2. Edge attributes: e.g. distance, edge weight,...

3. Graph attributes: e.g. number of nodes, number of edges,...

Graphs may be *directed*, i.e. edges distinguish between source and destination nodes. For example, in a graph describing a train network there may be an edge between station A and station B, i.e. a train goes from A to B, but not neccessarily from B to A.

Additionally, it is no requirement for all nodes to be connected to each other. The *connectivity* of a graph is one of its fundamental properties. *Fully connected graphs* are characterized by each node being connected to every other node in the graph. There are also *connected graphs* which are defined as graphs, where a path exists between any two nodes, i.e. there are no isolated nodes or collections of nodes. Analogously, a *disconnected graph* allows for two nodes to not be connected by a path. Furthermore, a *connected component* is defined as a connected subgraph of an unconnected graph, where every node of that component exclusively belongs to that component. This implies, that a graph is only connected if it has exactly one connected component.

Edge connectivity can be encoded in an *adjacency matrix A* of dimension $n_{\mathrm{nodes}} \times n_{\mathrm{nodes}}$ with entries $a_{ij} \in \{0, 1\}$ depending on the existence of an edge (as shown in Figure 1.18). However, using an adjacency matrix has some drawbacks. The number of entries scales quadratically with the number of nodes, which may become unfeasible as some graphs contain thousands or millions of nodes. This representation contains redundant information for undirected graphs, because $a_{ij} = a_{ji}$, which is space inefficient.

A more concise way of defining edges is a $2 \times n_{\mathrm{edges}}$ *edge set*, where each pair corresponds to a specific edge.
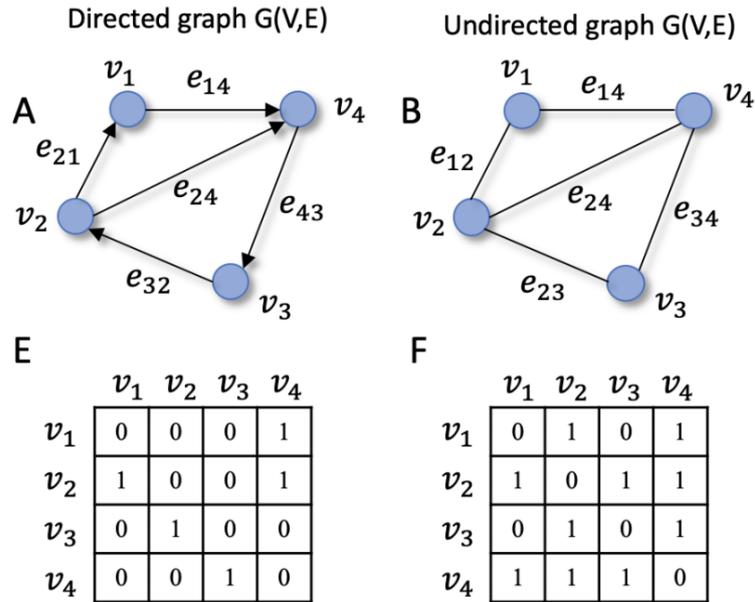
Figure 1.18: Adjacency matrix representation of a directed graph (left column) and an undirected graph (right column). The adjacency matrix of the directed graph is asymmetric. Figure taken from [13].

Tracking Detector Hits as Graphs

Modern silicon pixel tracking detectors contain millions to billions of individual pixels. When a particle traverses the detector, only a small fraction of pixels are hit, which leads to extremely sparse data. By associating every detector hit (i.e. the position measurement) with a graph node and the corresponding track segment with an edge, it is possible to find a natural graph representation of tracking detector data. This not only makes use of the inherent sparsity of the data but also allows for encoding of the spatial relations of the tracker hits.

1.3.2 Multi Layer Perceptrons

Neural networks (NNs) represent a large sub-field of machine learning models. They are roughly inspired by the structure of biological brains. On a fundamental level, NNs are constructed from interconnected *neurons*, typically organized in *layers*. While there are a number of different layer-types, the following will only deal with *fully connected layers*. In these layers, all neurons of the last layer are connected to all neurons of the next layer. A neuron receives inputs (i.e. real numbers) from other neurons or the input data itself, computes a weighted sum over all inputs, applies a non-linear *activation function* to that sum and returns the result as an output. The activation function is needed to capture non-linear features of the data at hand.

As mentioned, neurons are organized into layers. There are three types of fully connected layers: the *input layer*, *hidden layers* and the *output layer*. As the name suggests, input and output layers are the first and last layer of a network. The input layer receives the input data, while the output layer returns the prediction made by the network, which is

dependent on the given task. Finally, the hidden layers lie between input and output and are responsible for extracting more complex patterns with increasing depth (number of layers) and width (number of neurons per layer). NNs with at least three fully connected layers (i.e. at least one hidden layer) are called *Multi Layer Perceptrons* (MLPs). Figure 1.19 shows an MLP of depth 4 and width 5 in the hidden layers.

Training an MLP means adjusting the *weights* associated to every connection between neurons so that the discrepancy between a defined target and the network output is minimized. This is realized by numerically minimizing the error between the networks prediction and the truth information. This discrepancy is encoded in a so-called *loss function.*
The minimization process is achieved by performing a *forward pass* through the network to calculate the error of the prediction, i.e. the loss function. The goal is to adjust the network weights such that the loss function is minimized. This is done by performing *back-propagation*, where the gradients of the loss function with respect to all network weights are computed. Finally, the *gradient descent* algorithm is used to update the weights towards minimizing the loss function:

$$w_i' = w_i - \eta \frac{\partial Loss(\mathbf{w})}{\partial w_i}, \tag{1.3.1}$$

where, w_i' is the updated weight, $Loss(\mathbf{w})$ the loss function depending on all weights of the network and η the *learning rate.* The learning rate is a hyperparameter which is chosen before the training. Large η perform large modifications of the weights and thus are preferred in early training, but a lower learning rate is helpful for finer adjustments. Because of that the learning rate often decays periodically after a fixed number of training steps.
One way of using MLPs is to embed a given dataset into a highly dimensional *latent space* to then perform some other operations (e.g. passing it to a classifying network).
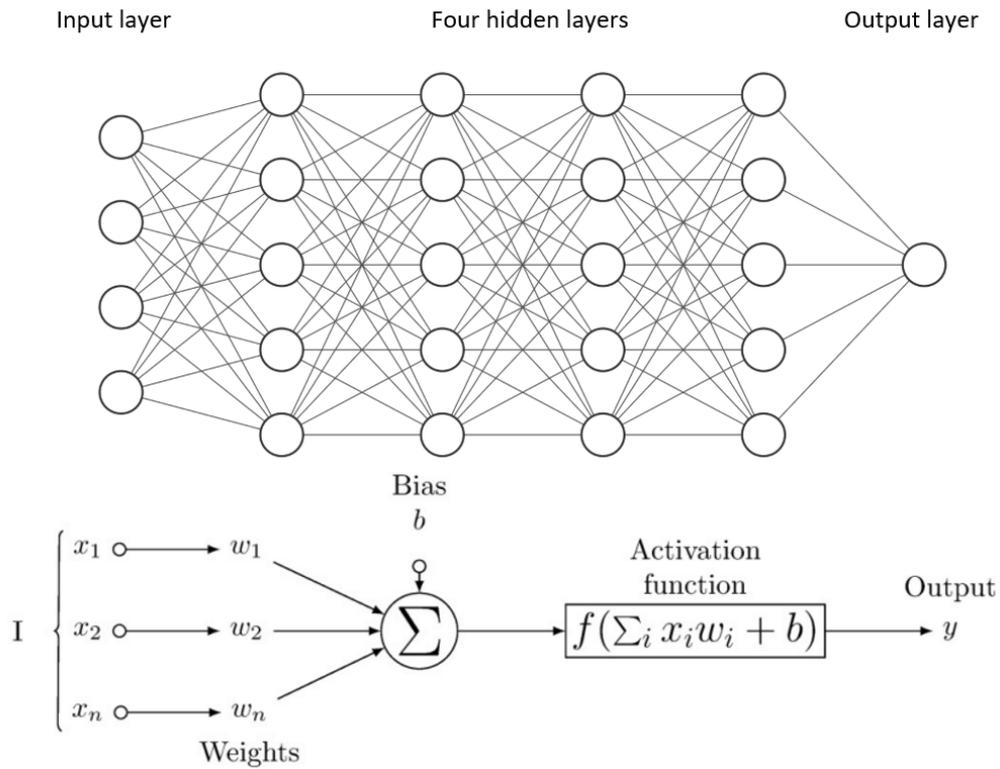
Figure 1.19: Schematic view of an MLP with a four node input layer, four hidden layers with five nodes each and a one node output. The process of calculating the output of a single neuron with n inputs x_i is shown on the bottom. The inputs are aggregated in a weighted sum with the connection weights w_i and added to an optional neuron bias b. The activation function f then calculates the output y. Figure taken from [12].

1.3.3 Graph Neural Networks

Other than the most common neural networks, operating on lists or grids of data, GNNs are designed to capture structural information in graph-structured datasets. As the recent years showed, GNNs present a powerful class of *geometric deep learning models* in a number of fields, including charged particle tracking in high energy physics [16, 19].

Similar to other neural networks, GNNs process graph-structured data by passing information through multiple network layers to produce predictions. Like previously discussed, information is encoded on three levels in a graph: node level, edge level and graph level. Similarly, a GNN can make predictions on each of these levels. In the context of particle tracking, an example for a graph-level task is determining if an event contains a specific particle. Distinguishing if that edge belongs to a particle track on the other hand is an edge-level task.

A relatively simple GNN for example, makes use of separate MLPs to embed all graph features. This is called a GNN layer. Analogously to MLPs with more hidden layers, the data can be fed through the GNN layer multiple times to extract higher level patterns. It is important to note that a GNN layer does not change the graphs structure (i.e. edges or nodes) but rather operates on the features associated to each of the graphs components. The last step, i.e. making the actual prediction (node, edge or graph level), requires a final decoder MLP. In some cases, a *pooling* step is added before the classification MLP. For example, in a node level prediction task, it is possible to aggregate edge-level embeddings of connected edges (usually by summming) to gather information on node level.

Interaction Network

The GNNs used during the course of this thesis are *interaction GNNs* [6], which are used to predict an *edge score* of every edge of the graph. The three building blocks of an interaction GNN are as follows:

1. Encoder MLPs: Node and edge level features are embedded with two separate MLP encoders.

2. Interaction network: Extracts the actual geometric patterns encoded in the graph by repeatedly performing message passing steps (see Section 1.3.3).

3. Decoder MLP: Takes the output of the interaction network and transforms the features of each edge to a singular score for every edge.

Message Passing

The mechanism at the heart of an interaction network is called *message passing*. By iteratively adjusting the node and edge embeddings, it is possible to learn more and more geometric patterns in the graph. Message passing in an interaction network is realized with two separate MLPs: the edge update MLP and node update MLP (or edge- and node block).

In a message passing step, an edge is selected and all features of connected nodes and the
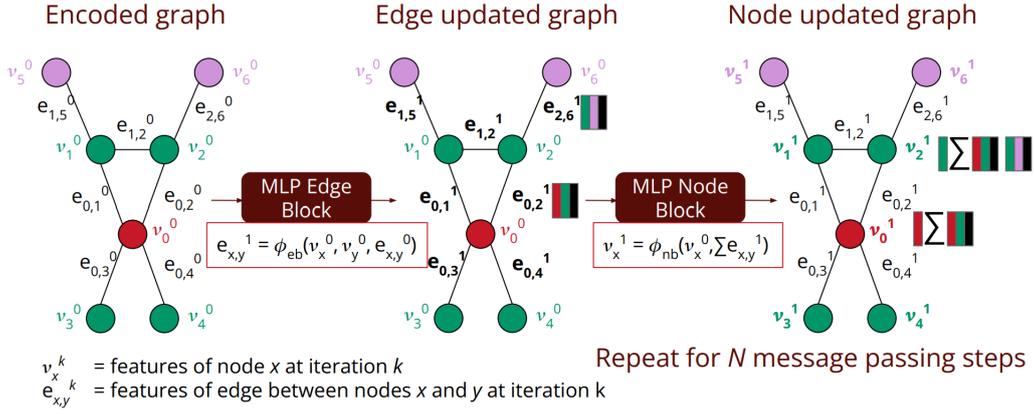
Figure 1.20: Schematic representation of a message passing step inside an interaction network. The edge block is denoted by ϕ_{eb}, the node block by ϕ_{nb} and the aggregation function is represented by Σ. Figure taken from [23].

edge features itself are passed to the edge block which produces an updated edge feature vector. This is repeated in parallel for all edges in the graph. Once all edge features are updated, a similar procedure is performed to update all node features. A node is selected and the current node features, as well as a permutation invariant *aggregation* (typically a sum) of all connected edge features is passed to the node block to update the node feature vector. Again, this is repeated in parallel for all nodes until every node is updated. This whole process is one message passing step and can be repeated N times. Figure 1.20 shows the process of a message passing step for a simple example graph.

1.3.4 Track Reconstruction with Graph Neural Networks

The application of GNNs for track reconstruction is primarily an edge level task. A graph is fed into the network, which assigns an edge score to every edge, ranging from 0 to 1. An edge score of 1 signifies that the edge is a track segment, whereas 0 means, that the edge in question is false. On this scored graph, a score cut is performed to finally label edges as either true or false.

Finally a connected components algorithm is applied to the classified graphs to produce *track candidates*. A track candidate is essentially a list of nodes belonging to a track.

1.3.5 Graph Construction

The data obtained from the tracking detector comes in lists. A GNN however, needs this data to be structured as graphs. This is done with a *graph construction algorithm*. In order to reduce computation time, the total number of edges after graph construction should be as small as possible, while containing all track segments.

In the following, graph size will correspond to the total number of edges found in the graph.

Fully Connected Graphs

The most simple way of construction a graph is done by creating a fully connected graph. Clearly, such a graph contains all track segments. However, the graph size rises quadratically with increasing number of nodes which makes fully connected graphs only feasible for smaller event sizes.

Considering an undirected graph without self-loops, the graph size n_{total} scales with the number of nodes n_n according to the following equation:

$$n_e = 1 + 2 + ... + (n_n - 1) = \frac{n_n^2 + 1}{2}. \qquad (1.3.2)$$

Because of the quadratic scaling this graph construction method is only feasible for graphs with a comparatively low number of nodes.

2 Methodology

This chapter is dedicated to explaining the methods employed for analyzing the feasibility of a *GNN-based cosmic muon trigger*.

First, the simulated data samples used throughout the course of this thesis are discussed. Then, the functionality of the GNN tracking pipeline is layed out, as well as the concrete GNN models used and their training process. Finally, the chapter concludes with motivating and defining the metrics to evaluate the perfomance of GNNs for cosmic muon track reconstruction.

The scope of this thesis is to explore if a GNN-based cosmic muon trigger presents a viable solution for detecting time frames containing cosmic muons for alignment. Because of the overwhelming amount of background, mainly in the form of Michel decays, the aim is to reconstruct as many cosmic muon frames as possible, while keeping the rate of falsely labeled frames down.

2.1 Simulation Data

All results presented in this thesis are based on simulated events from the *Mu3e simulation package* [14]. At the heart of the simulation lies the *Monte-Carlo*-based *Geant4 library*, commonly used to produce simulation data for high energy physics experiments.

The Mu3e coordinate system has its origin in the muon stopping target, with the z axis pointing along the beam direction, the y axis upward and forms a right-handed coordinate system with the x axis. The azimuthal angle φ of a point is the angle between the x axis and the line from the origin to the projection of the point onto the $x - y$ plane.

The simulation contains the detector geometry discussed in Section 1.2.2, including the sensors, HDIs and support structures. The magnetic field is modeled as a symmetric cylindrical field in 10mm step sizes, based on calculations of the manufacturer. It is planned to replace this with a measured field map in the future [14]. Furthermore, the magnet itself is also simulated when performing cosmic muon simulations.

Cosmic muons are simulated as originating from a parabolic cloud between $1000\,\text{mm}$ and $100\,\text{mm}$ above the detector (as shown in Figure 2.2) with Gaussian momentum distributions in x- and z-direction and purely negative momentum in y-direction. Michel electrons and positrons are generated by simulating muon decays on the target in the center of the detector.
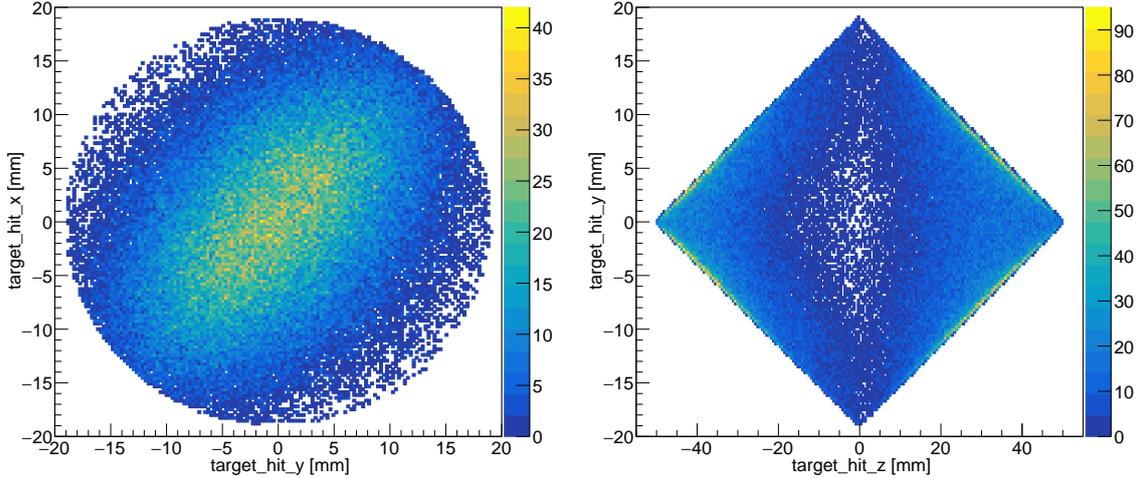
(a) Heatmap of the vertex positions of the simulated Michel decays in the x-y plane. The uneven distribution stems from the simulated beam profile.

(b) Heatmap of the vertex positions of the simulated Michel decays in the y-z plane.

Figure 2.1: Vertex positions of electrons and positrons generated in simulated Michel decays. The double cone shape of the target is clearly visible. The plots are generated using the Mu3e simulation package [14].
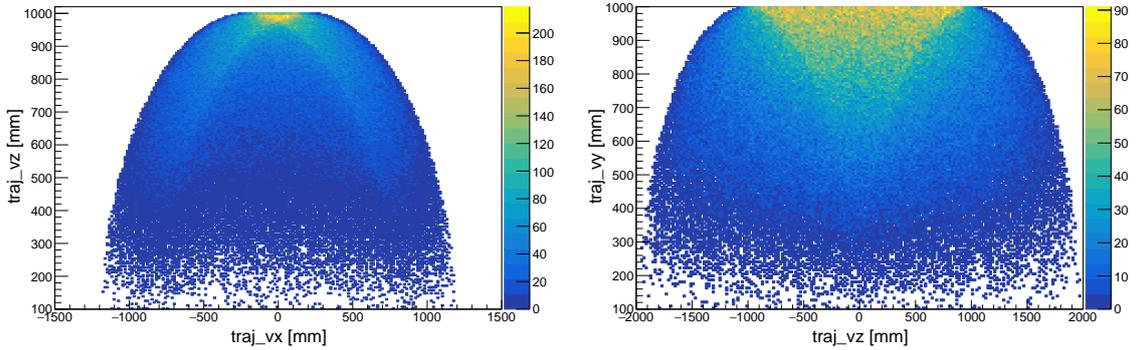
(a) Heatmap of the vertex positions of the simulated cosmic muons in the x-y plane.

(b) Heatmap of the vertex positions of the simulated cosmic muons in the y-z plane.

Figure 2.2: Vertex positions of simulated cosmic muons above the detector. The plots are generated using the Mu3e simulation package [14].

The simulation data studied during the course of this thesis can be divided into three categories:

1. Frames containing a cosmic muon (beam-off)

2. Frames only containing Michel decays (beam-on, no cosmics)

3. Frames containing one cosmic muon overlayed with Michel decays

The cosmic-muons-only samples are by far the smallest by the number of hits and thus require the least amount of computational power to perform analysis on.

Michel-only samples on the other hand are much larger than cosmics-only samples, due to the higher overall number of particles and low momentum electrons recurling and leaving multiple hits per layer. With no cosmic muons being found in these samples, they can be used to verify the training of GNNs by checking for fake tracks. This is especially important for obtaining an estimate for the background rejection, i.e. how often the trigger would select a background frame.

Finally, the samples with Michel decays also containing a cosmic muon are essentially the type of data a cosmic muon trigger is expected to be able to deal with. After all, the goal is to find cosmic muon tracks among the immense background of electrons and positrons from Michel decays to mark these frames for alignment purposes.

2.1.1 Distributions of the Simulation Data

To get an understanding of the data at hand, a collection of histograms is shown, illustrating different properties of the samples. These include hit positions in the tracking detector, particle momenta, vertex locations as well as the number of hits per track and particle constitution of the samples.

Figure 2.3 shows the distributions of the hit positions inside the Mu3e tracking detector, both in Cartesian and cylindrical coordinates. The x and y components show peaks around the radii of the tracking layers with Michel electrons and positrons producing around one order of magnitude more hits than the cosmic muons with the same number of frames. The distribution of z positions clearly shows the separation between the three stations of the Mu3e detector. For Michel frames, by far the most hits are registered in the central station, whereas the abundance of hits decreases with increasing distance to the central station in the recurl stations. The assymmetric distribution in the recurl station is caused by the simulated stopping distribution of the target (see Figure 6.3 in [14]). The cosmic muon histogram shows an even distribution of cosmics over the entire detector, as mentioned in Section1.2.3.

When converting Cartesian coordinates to cylindrical coordinates, the layer structure of the tracking detector becomes visible. Cosmic muons leave more than an order of magnitude more hits in the outer layers, compared to the inner layers. This is owed to the much smaller surface area of the inner layers in the $x - z$ plane. Furthermore, there is a clear dependence of hit positions and the azimuthal angle φ for cosmic muons while Michel decay frames seem to be uniformly distributed as expected. This is due to the larger projected detector area in the x-z plane on the top and bottom of the detector ($\varphi = \pm\frac{\pi}{2}$).

Hit Position Distributions of 10000 Events per Dataset in Cartesian and Cylindrical Coordinates
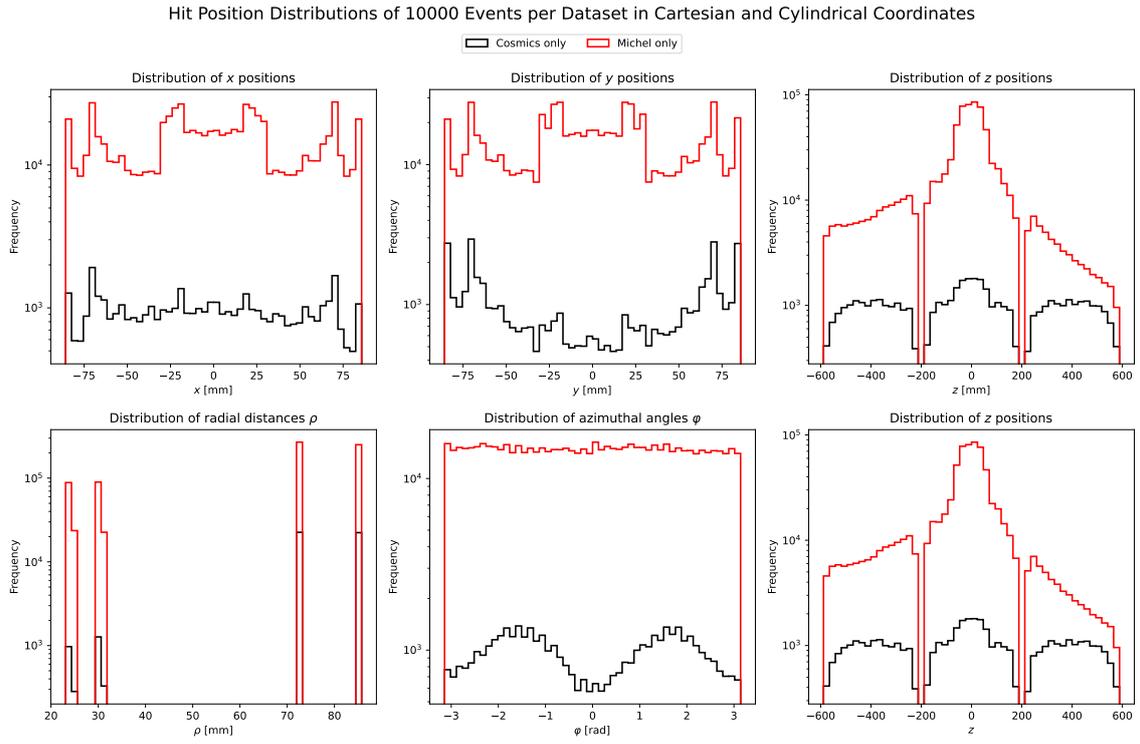

Figure 2.3: Hit position distributions for the cosmics-only sample (black) and the Michel-only sample (red) in Cartesian coordinates (top row) and cylindrical coordinates (bottom row) for 10000 frames per sample. The two histograms of the z position are identical.

Figure 2.4 shows the total and transverse momentum, as well as the momentum components of muons and electrons in the data samples. As expected, cosmic muons posses far larger momenta of up to $2\,\mathrm{TeV}$, while the total momentum of Michel electrons are cut off at the Michel edge of $\approx 53\,\mathrm{MeV}$. The clear edge at the lower end of the cosmic muon momentum distribution at $2.5\,\mathrm{GeV}$ stems from a cut in the generation of the cosmic muons in the simulation because the behaviour low energy muons are not well enough modeled. The bottom row of plots shows that the electron momenta are similarly distributed in all components, the same goes for the x and z of muon momenta, however the y component clearly shows a bias towards high negative momenta (top right plot).

The blue curves in Figures 2.4 and 2.5 correspond to a few in-flight decays and scattering events producing $e^{\pm}$ in the pure cosmic muon sample. However, they only make up less than 1% of the total number of tracks.
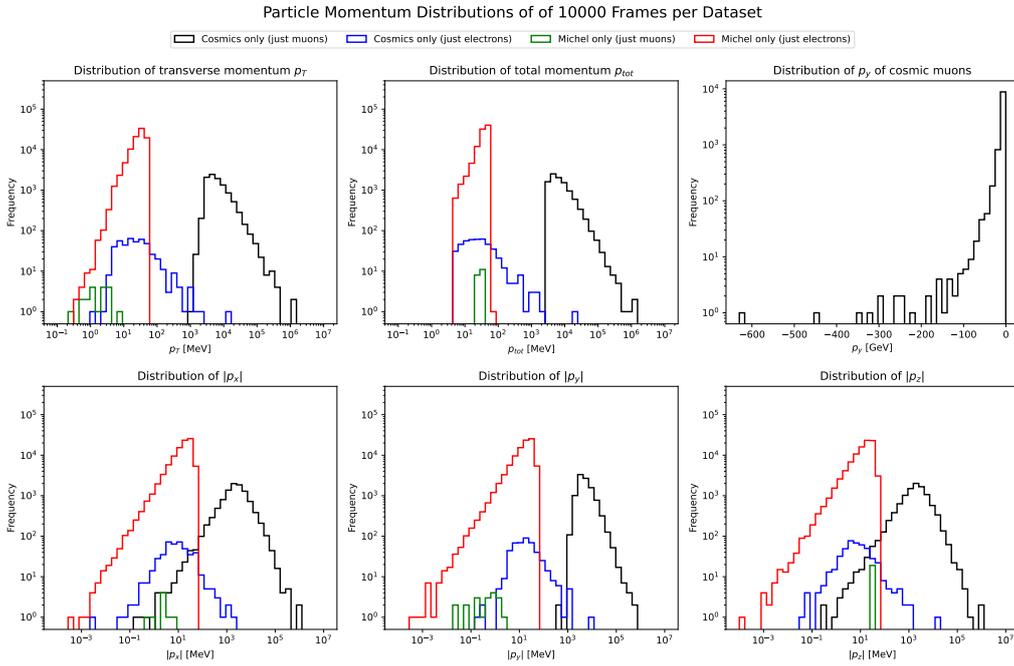

Figure 2.4: Top row: Histograms of transverse momentum p_T, total momentum p_{tot} and the y components of cosmic muon momenta.
Bottom row: Histograms of the absolute values of the momentum components for both $e^{\pm}$ and μ.
Muons from the cosmics-only sample are shown in black, electrons from the same sample in blue, while the few beam muons from the Michel-only sample are shown in green and electrons in red.
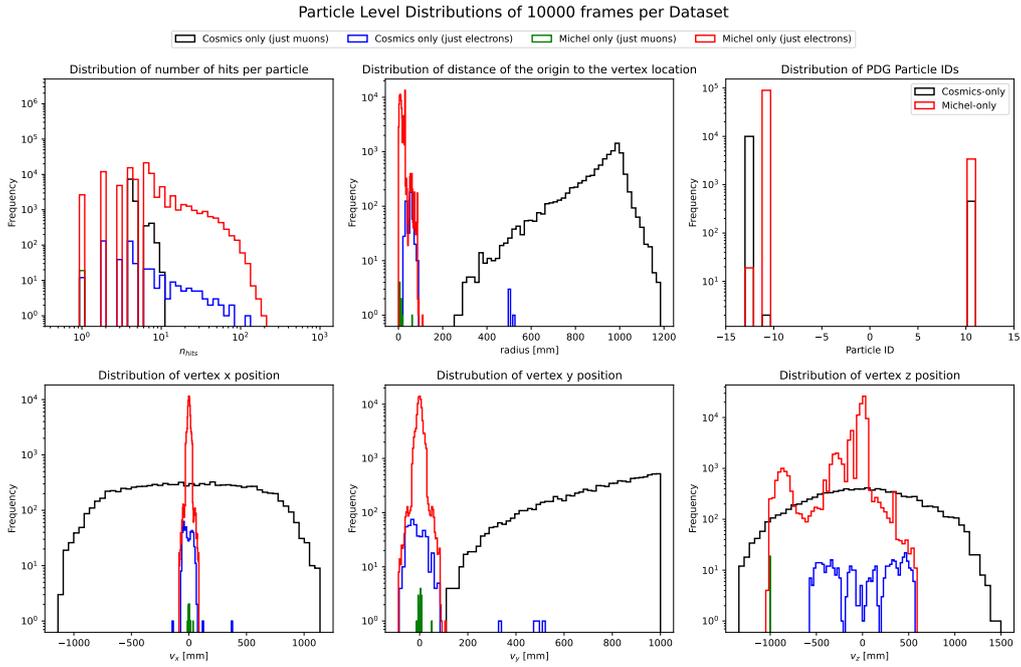
Figure 2.5: Top row: Histograms of the number of hits in the detector per particle, the distance to the vertex location and the particle ID according to the PDG Monte-Carlo numbering scheme. Bottom row: vertex positions in Cartesian coordinates.
Muons from the cosmics-only sample are shown in black, electrons from the same sample in blue, while the few beam muons in the Michel-only sample are shown in green and electrons in red.

Figure 2.5 shows the number of hits each particle left in the detector. Electrons and positrons are able to produce up to 200 hits per track in the given dataset, while cosmic muons leave 10 hits at most. Also, when considering the vertex positions and the distance between detector and the vertex, the different origins of cosmic muons and electrons are shown. While most electrons originate from the center of the detector, muons are generated between $1000 \, \mathrm{mm}$ and $200 \, \mathrm{mm}$ above the detector with Gaussian distributions in x and z. Finally, the distribution of particle IDs shows the constitution of the two samples considered. The cosmic muon sample almost exclusively consists of μ^+, particle ID -13, with only a few secondary electrons (particle ID ± 11).

2.2 Graph Neural Network Tracking Pipeline

Originally developed for charged particle reconstruction of collision events in the inner tracker (ITk) of the ATLAS experiment at LHC, the Acorn track reconstruction framework [15, 17, 18, 21] provides most of the functionality needed for this work. The pipeline is based on the PyTorch Geometric (PyG) library [27], which is built upon the popular deep learning library PyTorch [26] to be able to conveniently build and train GNNs .

2.2.1 Stages of the Pipeline

The reconstruction pipeline works in three stages as outlined in Figure 2.6:

1. Graph Construction: Raw frames from the tracking detector are converted into a graph representation. Hits in the detector comprise the nodes and may be connected by an edge. An edge between two nodes suggest, that these nodes might belong to the same particle track. Furthermore, a set of node- (e.g. space point coordinates, tracker module IDs, etc.) and edge-level attributes (e.g. distance between nodes) are associated to each node and edge.

2. Edge Labeling: A GNN is inferred on these graphs assigns an edge classification score between 0 and 1 to each edge in each graph. High scores suggest that the two nodes connected by the edge are associated to successive hits of a target particle's trajectory. Edges with low scores correspond to everything else, i.e. fake edges or non-target particle tracks.

3. Graph Segmentation: Track candidates, i.e. a list of nodes belonging to the track, are generated from the scored graph by applying minimum score thresholds. The track candidate can then be run through a track fitting algorithm to obtain the track parameters.
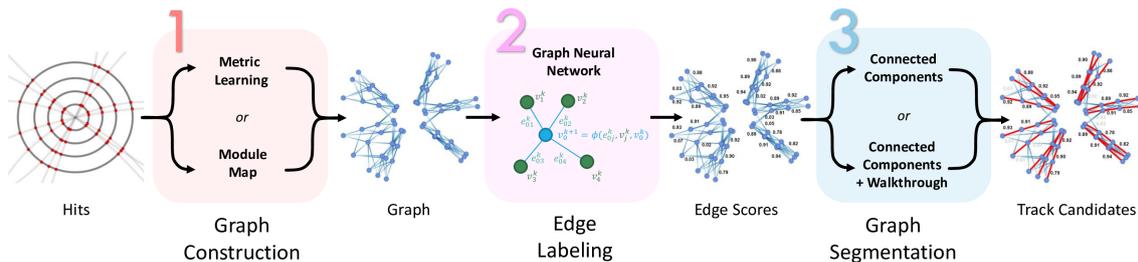

Figure 2.6: Schematic overview of the GNN-based track finding pipeline. The graph construction step is modified for Mu3e frames by implementing fully connected graph construction. Figure taken from [17].

Additionally, the pipeline also provides functionalities to read-in data files with truth information on particles and their hits in the detector. Every event is saved in a separate file and then passed to the data reading stage. During the data reading stage, a PyG file containing the truth graph is created, for every frame file, i.e. graphs containing only true track edges as well as other node and graph level features, such as hit positions, momenta

or particle type. Lastly, there are also evaluation metrics employed for every stage of the pipeline which are discussed in Section 2.3.

It follows the typical workflow of the pipeline:

1. Data reading stage: Every frame file is converted into PyG files containing the truth graph and all other features.

2. Graph construction stage: The truth PyG files are expanded by adding a graph according to a graph construction method of choice. These files are then used in training and inference of the GNN. When using ML based graph construction methods like metric learning, this is preceeded by training the graph construction model.

3. GNN training stage: After defining training parameters, the overall network architecture and a target particle, the GNN is trained using the truth information contained in the provided PyG files. During training, the progress is tracked and the best training checkpoint is continuously updated throughout the training.

4. GNN inference stage: The trained GNN is inferred on the graphs from the graph construction stage. In that process, an edge score is assigned every edge.

5. Track reconstruction: A connected components algorithm is employed, which performs a score cut on the edges to generate a track candidate (lists of connected nodes).

In between the pipeline stages, there are evaluation metrics employed, which are explained in Section 2.3.

2.2.2 Adaptations for Mu3e Frames

The data read-in and the graph construction stage have to be modified for use in the Mu3e experiment because of the difference in expected data rate.

Because the ATLAS experiment observes collision events of protons with center of mass energies several orders of magnitude higher than at Mu3e many more tracks per event are produced. Furthermore, the ITk has more detector layers than the Mu3e detector, which leads to more hits per track.

Graph Construction

Due to the much smaller frame size at Mu3e, fully connected graphs are a viable option. By constructing all possible combinations of nodes, without self-loops and directionality, a fully connected edge list is obtained.

Figure 2.7 shows the distributions of number of nodes and edges per fully connected graph, evaluated on 10^4 graphs. Clearly, the histogram of the number of hits is similar to the one shown in Section 2.1, with cosmic muons producing tracks with up to 10 nodes, while electron tracks reach up to $\mathcal{O}(10^2)$ nodes. The number of edges per fully connected graph scale quadratically with the number of node according to Equation 1.3.2.

Figure 2.7 illustrates the distribution of the number of signal edges (cosmic muon track segments) per graph in comparison to the number of background edges per graph. Per cosmics with Michel graph, there are on average 4.48 muon nodes and 3.48 muon edges. Electrons leave on average 75.25 hits and the mean number of background edges is 3952.
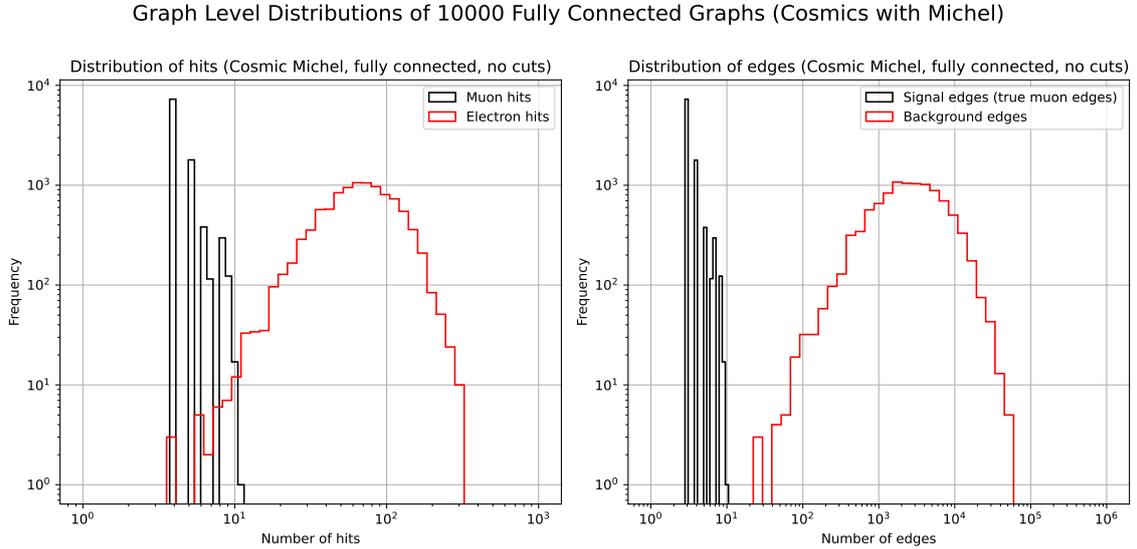

Figure 2.7: Histograms of the number of nodes for electrons and muons (left) and the number of signal and background edges (right) per fully connected graph.

2.2.3 Graph Neural Network Models

As mentioned in Section 1.3.3, the GNNs studied during this thesis are interaction GNNs. An examplary architecture is illustrated in Figure 2.8. Because one MLP alone needs many hyperparameters to define its architecture, the intitialization of the interaction GNN is simplified by using very similar architectures for the MLPs it is constructed from. All MLPs have the same number of nodes per hidden layer, this parameter is called hidden dimensions. The depth, i.e. the number of layers of each MLP is determined by two parameters, one for node related MLPs (node encoder and node block) and one for edge related MLPs (edge encoder, edge block and the decoder). Additionally, there are normalization layers in between every fully connected layer to help stabilize the training. Table 2.1 shows the MLP parameters used for the GNN studied during this work.
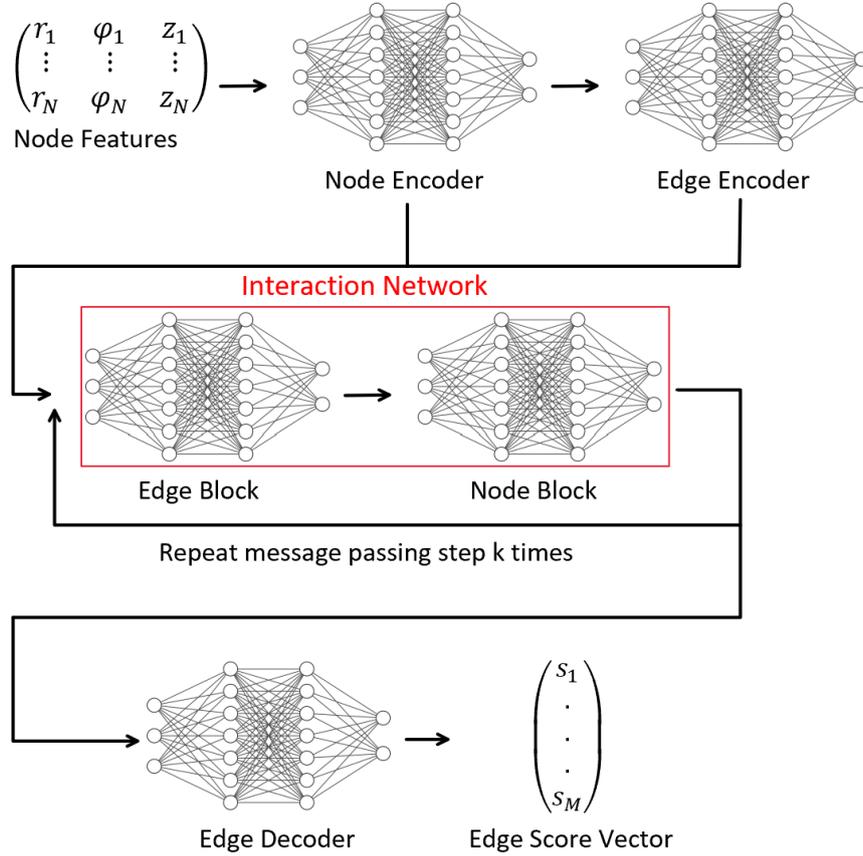
$$\begin{pmatrix} r_1 & \varphi_1 & z_1 \\ \vdots & \vdots & \vdots \\ r_N & \varphi_N & z_N \end{pmatrix}$$

Node Features

Node Encoder

Edge Encoder

Interaction Network

Edge Block

Node Block

Repeat message passing step k times

$$\begin{pmatrix} s_1 \\ \cdot \\ \cdot \\ \cdot \\ s_M \end{pmatrix}$$

Edge Decoder

Edge Score Vector

Figure 2.8: Schematic representation of the architecture of an edge classifying interaction GNN (not to scale). Node level features are encoded by separate MLPs to a latent space representation. The encoded features are passed to an edge encoder to obtain the edge features. Both the embedded node- and edge featues are fed into the interaction network. After k message passing steps where node- and edge features are being fed repeatedly into edge- and node block MLP, the datastream is transferred to the edge decoder MLP. It transforms the latent space representation to obtain an M dimensional edge score vector, containing values between 0 and 1. Figure inspired by [23].

MLP	Input size	Output size	Total number of layers
Node encoder	3	24	2
Edge encoder	48	24	2
Edge block	72	24	2
Node block	72	24	2
Edge decoder	72	1	3

Table 2.1: Parameters of the MLPs used to build up the GNN studied in this thesis. The number of layers refers to fully connected layers, normalization layers are not counted. This model employs layer normalization and performes four message passing steps per graph. Each layer uses ReLu activation functions. The input features are the cylindrical coordinates of the hit positions (r, φ and z).

All MLPs in the GNN are structured so that a fully connected network layer is followed by a layer normalization which is followed by the *Rectified Linear Unit* (ReLU) activation function, defined as $\text{ReLU}(x) = \max(0, x)$. The only exception is the edge decoder network which does not have an activation function or layer normalization after the last fully connected layer.

The in- and output sizes of the MLPs can be explained as follows. The node encoder receives the node features, i.e. cylindrical coordinates in this case. The node encoder embeds these features into a 24 dimensional latent space representation. Each edge is associated with two nodes, thus the edge encoder receives the embedded node features of both nodes it connects. After that, the edge block is passed a concatenation of the embedded node and edge features. The node block receives the embedded node features, as well as an aggregation (in this case a sum) of all outgoing and incoming message. Outgoing refers to messages where the node is sender and vice versa for incoming messages. At the end, the edge decoder once again receives all embedded node and edge features of the edge and outputs a single number, the edge score.

2.2.4 Graph Neural Network Training

Like for regular MLPs, the GNN training also hinges on a loss function. An edge-classifying interaction GNN performs, as the name suggests, a binary classification task. A commonly used loss function is Binary Cross Entropy (BCE):

$$\text{Loss}(y_{\text{pred}}) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_{\text{true}} \log(y_{\text{pred},i}) + (1 - y_{\text{true}}) \log(1 - y_{\text{pred},i}) \right]. \qquad (2.2.1)$$

The network prediction is denoted as y_{pred}, the truth label as y_{true} and the total number of samples in the dataset as N. Additionally, the BCE loss is modified by introducing specific weights to truth information by multiplying each term inside the sum by a number. Weighting the contributions in the BCE loss can help the network to focus on correctly classifying a certain class of predictions.

In the context of an edge classifying GNN, the sum runs over the number of edges in the graph. Including the weighting, the loss is calculated with:

$$\text{Loss}(y_{\text{pred}}) = -\frac{1}{N_{\text{edges}}} \sum_{i=1}^{N_{\text{edges}}} W_{\text{true}} \left[y_{\text{true}} \log(y_{\text{pred},i}) + (1 - y_{\text{true}}) \log(1 - y_{\text{pred},i}) \right], \quad (2.2.2)$$

where the weight factor W_{true} depends on the specific edge and takes values as shown above. Because the training is supposed to be focused on reconstructing cosmic muon tracks, their contribution to the loss is increased by a factor of three, while true electron edges and fake edges are treated equally as false with weight one.

Thus, the two cases are:

- True muon edges: $y_{\text{true}} = 1$ and $W_{\text{true}} = 3$. This leads to the following contribution to the sum:

$$\text{loss}_i = 3 \log y_{\text{pred},i} \quad (2.2.3)$$

- Non-target true edges and false edges: $y_{\text{true}} = 0$ and $W_{\text{true}} = 1$. This leads to the following contribution to the sum:

$$\text{loss}_i = \log 1 - y_{\text{pred},i} \quad (2.2.4)$$

The training itself is conducted by using the „$Adam$"optimizer [7], an optimized version of gradient descent. It employs using the exponentially weighted average of the error gradients like in Figure 1.19. The weights are updated via

$$w_{t+1} = w_t - \eta m_t, \quad (2.2.5)$$

where w_{t+1} is the new weight, w_t the current weight, η the learning rate and m_t is recursively defined as

$$m_t = \beta m_{t-1} + (1 - \beta)\frac{\partial Loss}{\partial w_t}, \quad (2.2.6)$$

with a constant parameter $0 < \beta < 1$.

Training Process

Before starting a network training, the dataset is split into three parts: train set, validation set and test set. This is common practice in neural network training to avoid and detect overfitting. The term overfitting in this instance refers to a neural network „memorizing"the given training data instead of finding higher level patterns to generalize.

This is detected by calculating the loss on data from the train set, updating the weights with gradient descent and then calculating the loss again for data from the validation set. By infering the network on data it was not directly trained on, the ability to generalize is tested. Theoretically, the training could be run indefinitely and the train loss would decrease. However, the validation loss would stagnate or increase at some point, i.e. the network does not improve on generalizing but rather focuses on memorizing the train set.

The test set is used to evaluate and test the network performance once training is complete A typical split of data is 80% train set, 10% validation set and 10% test set. This ensures sufficiently many training samples while achieving enough statistics for validation set and test set.

The training done during this work is structured such that only the network checkpoint with the lowest validation loss is saved. This essentially mitigates the risk of selecting an overfitted model. The training progress is monitored in real time using the Weights & Biases framework [28]

Figure 2.9 shows both training and validation loss plotted against the training epoch for an exemplary GNN training. The train loss tends to decrease even for later epochs. The validation loss stagnates around the 60th epoch which indicates slight overfitting.

Additionally, the edgewise target purity and efficiency (see Section 2.3.2) are monitored throughout the training process. Both metrics are shown as a function of the training epoch in Figure 2.10
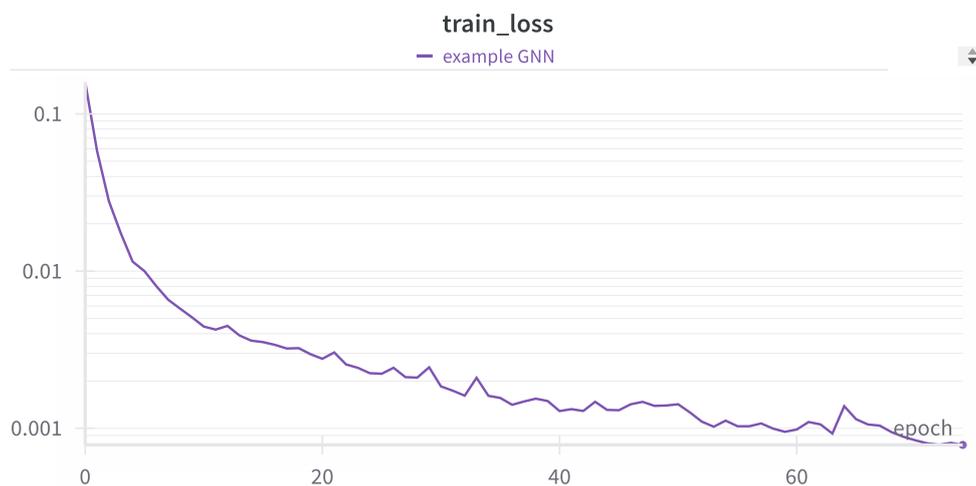
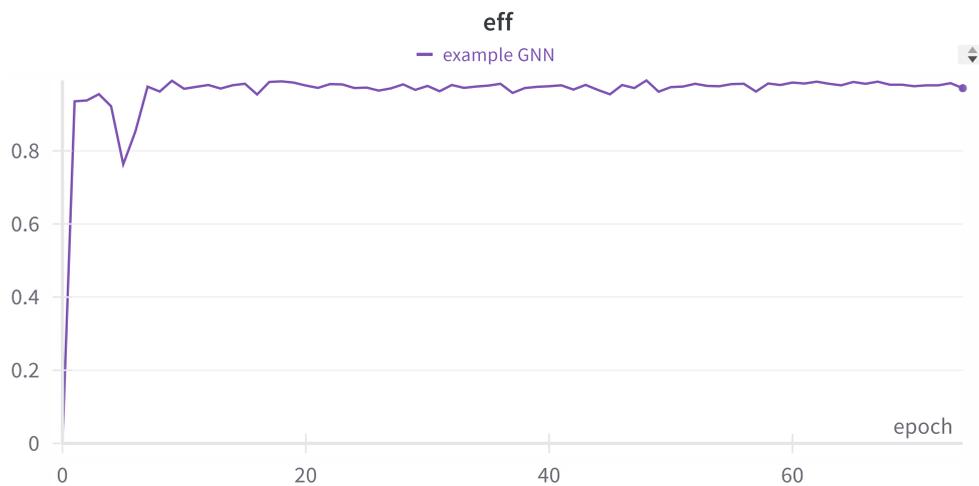(a) Training loss for every training epoch.
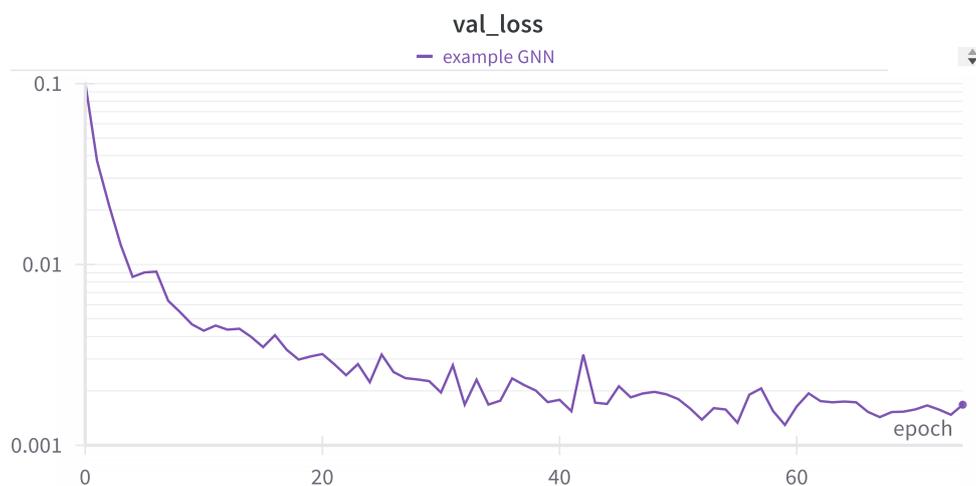
(b) Validation loss for every training epoch.

Figure 2.9: Loss curves during an exemplary GNN training.
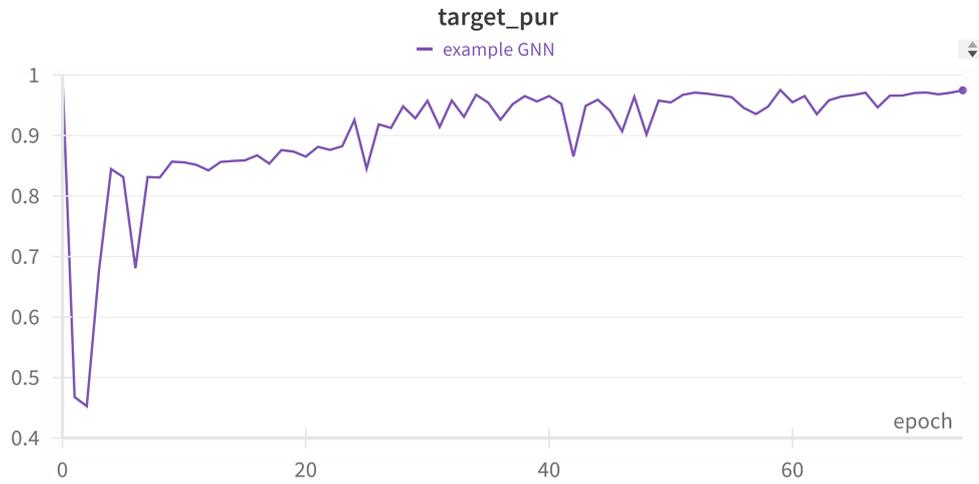
(a) Target purity for every training epoch using a score cut of 0.5.

(b) Target efficiency for every training epoch using a score cut of 0.5.

Figure 2.10: Target efficiency and purity at every training epoch. Target in this instance refers to true muon edges.

2.3 Evaluation Metrics

In order to make statements about the performance and to have a control instance for every stage of the pipeline, an assortment of performance metrics is applied.

2.3.1 Graph Construction Efficiency and Purity

The graph construction efficiency ϵ_{GC} and -purity π_{GC} are defined as

$$\epsilon_{GC} = \frac{n_{\text{true}}}{n_{\text{segments}}} \qquad (2.3.1)$$

$$\pi_{GC} = \frac{n_{\text{true}}}{n} \qquad (2.3.2)$$

where n_{true} denotes the number of track segments found in the edge list of the constructed graph, n_{segments} the number of track segments found in the truth graph and n the total number of edges in the edge list. Ideally, the efficiency should always be 1, i.e. every track segment is found in the edge list. When considering fully connected graphs, it is intuitively clear that this is indeed the case. It is also possible to perform edge cuts to improve purity, graph size and hence computation time but this comes reduces efficiency. The average graph construction efficiency and purity is computed by summing n_{true}, n_{segments} and n over all graphs of the sample, and using Equations 2.3.1 and 2.3.2.

2.3.2 Edgewise Target Efficiency and Purity of the Graph Neural Network

Continuous metrics are used to evaluate the GNN training process. By setting a score cut in the training configuration, i.e. a threshold value above which a target edge is classified as true, the edgewise efficiency and purity for every training epoch can be defined like

$$\epsilon_{target} = \frac{n_{\text{target,TP}}}{n_{\text{target}}} \qquad (2.3.3)$$

$$\pi_{target} = \frac{n_{\text{target,TP}}}{n_{\text{true}}}. \qquad (2.3.4)$$

In this context, $n_{\text{target,TP}}$ refers to the number of target edges that are correctly classified as true, n_{target} is the number of target edges and n_{true} is the total number of edges classified as true. Consequently, the ideal edge classifier correctly labels all target edges as true ($\epsilon_{\text{target}} = 1$) and no other edges as true ($\pi_{\text{target}} = 1$).

2.3.3 Edge Score Distribution

The GNN assigns each edge a score between 0 and 1, with the aim of assigning true target edges a high score and low scores to anything else. When evaluating the performance of a specific model it is important to know if the edge scores are indeed correctly predicted for most edges. To do that, a GNN trained to assign high scores to edges of tracks of a specified target particle is inferred on a new dataset (typically the test set). A histogram of the edge scores of target edges, fake edges and non-target true edges (edges containted within the truth graph, which do not correspond to a target particle) is plotted. The

ideal score distribution has all target edges in a bin around 1 and everything else in a bin around 0. Realistically, a GNN is not able to perfectly discrimminate true and fake edges, therefore the histogram for target edges bleeds into the low scores and vice versa. An exemplary score distribution of an edge classifying task is shown in Figure 2.11.
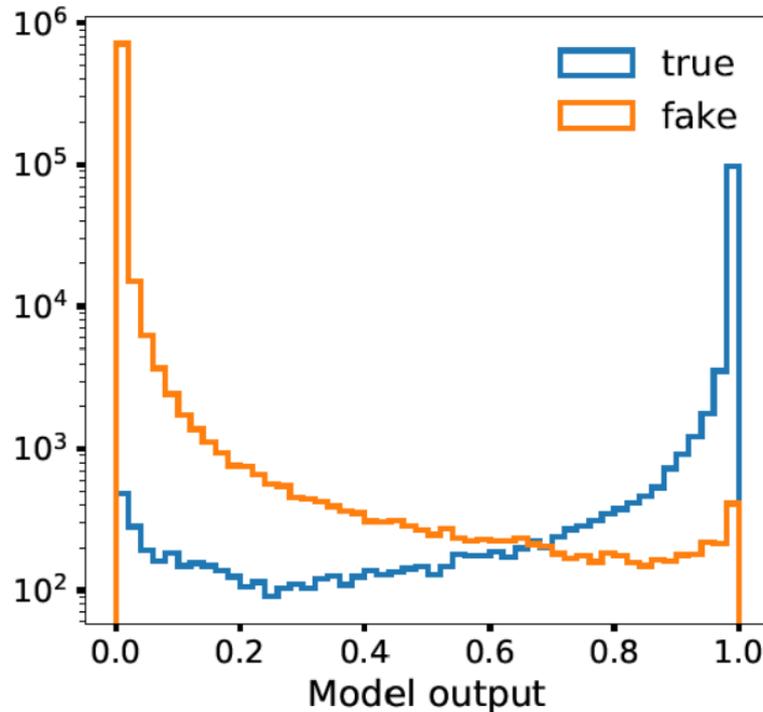

Figure 2.11: Example of a score distribution of an edge-classifying GNN. Most true target edges achieve high scores close to 1, while fake edges are scored close to 0. In some instances true edges obtain low scores and fake edges high scores. However, the abundance of these edges is $\mathcal{O}(10^3)$ smaller than the one of correctly scored edges. Figure taken from [25].

2.3.4 Receiver Operating Characteristic Curve and Area Under Curve

The Receiver Operating Characteristic (ROC) curve is a widely used metric illustrating the performance of a binary classifier. It is obtained by plotting the true positive ratio (TPR) against the false positive ratio (FPR) at different threshold values.

In the example of an edge classifying network, a score cut is repeatedly performed on a dataset of scored graphs, while varying the cut threshold value. Each score cut alters the TPR and FPR so that a ROC curve is obtained. The TPR in this case corresponds to the edgewise efficiency, while the FPR corresponds to fake- and non-target true edges mistakenly labeled as target true. An ideal ROC curve is just a point at $(0, 1)$, i.e. TPR=1 and FPR=0. Figure 2.12 shows ROC curves for models of different quality. A random, binary classifier has a 50% chance to correctly classify an edge, therefore the ROC curve will always produce a TPR close or equal to the FPR.

Sometimes it is impractical to look at the whole ROC curve, especially when comparing many different models. To still have a value to compare the models with, the area under curve (AUC) is used. As the name suggests, the AUC is defined as the area under the ROC curve. Since the ROC curve is bound by TPR and FPR $\in [0, 1]$, the AUC also takes a value between 0 and 1.
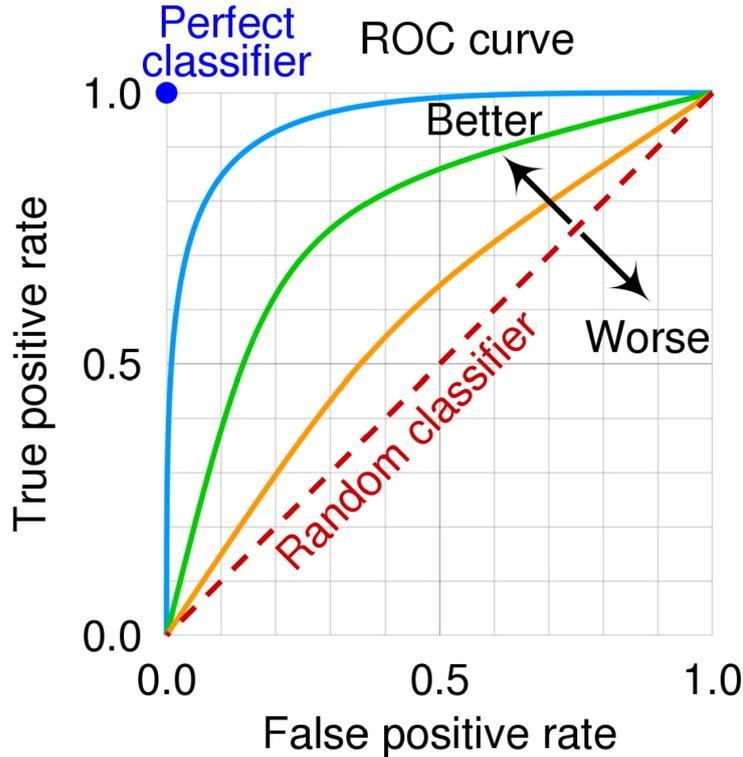

Figure 2.12: ROC curves as a evaluation metric of ML models. The AUC is defined as the area under the ROC curve. Figure taken from [24].

2.3.5 Track Reconstruction Efficiency and Fake Track Ratio

To quantify the track reconstruction stage, two metrics are introduced: the track reconstruction efficiency (in short: reconstruction efficiency) and the fake track ratio (in short: fake ratio). To define these metrics, a definition of a track (and fake track) is required first.

In the following, a track is a collection of N nodes which are connected by at least $N - 1$ edges. To perform a track fitting algorithm at least three nodes are needed. However, tracks with four nodes are more useful for redundancy reasons. A true target track in turn is defined as a track where every node stems from the target particle. Conversely, a fake track is a track with at least one non-target node.

A cosmic muon trigger system, shall trigger a frame if there is at least one true track found in the track candidates. Therefore, the number of reconstructed tracks n_{recon} is increased by 1 if at least one of the track candidates of a single frame is target true, whereas the number of fake tracks n_{fake} is increased by 1 if all track candidates are fake tracks.

The reconstruction efficiency ϵ_{recon} and the fake ratio r_{fake} in a set of k graphs are defined as:

$$\epsilon_{\text{recon}} = \frac{n_{\text{recon}}}{n_{\text{expected}}} \tag{2.3.5}$$

$$r_{\text{fake}} = \frac{n_{\text{fake}}}{k}, \tag{2.3.6}$$

where n_{expected} is the expected number of true target tracks in the dataset. For the cosmics only sample, this is the number of graphs, because every frame contains a cosmic muon track. The reconstruction efficiency is not defined for the Michel only sample because the number of expected true target tracks is zero.

Both rates depend on the minimum number of track hits N and the score cut, determining whether an edge is considered true or false. By scanning multiple score cut threshold values and checking different minimum track lengths, the fake track ratio in combination with the track reconstruction efficiency can be used to find a good working point for the construction of a cosmic muon trigger.

In the light of a cosmic muon trigger, the background rejection is a highly important characteristic. In this case, the background rejection is defined by the acceptance of the trigger, i.e. the percentage of frames that are triggered. This is the number of correctly reconstructed muon tracks, added to the total number of fake tracks, divided by the total number of frames. Consequently when infering on Michel-only frames, the fake ratio is equal to the acceptance of the trigger.

3 Results

In the following chapter, the results are presented in two sections. The first one, Section 3.1, deals with the evaluation of the inference and track reconstruction step for multiple GNNs and inference data sets. The main focus lies on the track reconstruction step at the end of the pipeline. By analyzing the track reconstruction, the most relevant metrics concerning a cosmic muon trigger are obtained, such as the frame acceptance and reconstruction efficiency. These metrics are paramount in assessing the feasibility of a GNN-based cosmic muon trigger.

The last Section 3.2 covers the summary of the results, drawing a conclusion whether a GNN-based cosmic muon trigger is worth further investigating and gives an outlook on possible improvements.

3.1 Graph Neural Network Evaluation

The GNN performance evaluation is carried out in two stages: quality of the inference step and track reconstruction performance. While the results from the track reconstruction yield the most expressivity concering trigger performance, it is also important to shine a light on the step before, as the track reconstruction founds on the results obtained in inference.

The performance of differently trained GNNs with respect to the mentioned metrics is evaluated by performing the inference step on the three data samples, cosmics-only, Michel-only and cosmics-with-Michel and subsequently reconstructing the cosmic muon tracks using the scored graphs. Because the GNN should only produce tracks if a cosmic muon is present in the graph, the inference on Michel-only samples is of special interest, as this allows for an estimation for the background rejection of the trigger. In total, twelve combinations of GNN and inference data sample are analyzed.

The GNNs all have the same architectural specifications listed in Table 2.1 and are trained on four different data sample mixtures. The first one is trained on cosmics-with-Michel frames exclusively, while the others are mixed with cosmics-only and Michel-only frames. This is done to investigate the effect of giving the GNNs additional samples Michel-only or cosmics-only frames. By adding cosmics-only, the hope is to boost the efficiency with the beam being turned off and by adding Michel-only frames a boost in background rejection is aimed for.

The training datasets are:

1. Cosmics-with-Michel

2. 50% Cosmics-with-Michel and 50% cosmics-only

3. 50% Cosmics-with-Michel and 50% Michel-only

4. 33.3% Cosmics-with-Michel, 33.3% cosmics-only and 33.3% Michel-only

In addition to that, another model is trained exclusively on the cosmics-only sample to investigate the performance of infering on cosmics-only frames for the beam-off case.

3.1.1 Inference

The four metrics employed to quantify the inference step are edgewise efficiency, the ROC curve, associated AUC and the score distribution itself.

Score Distributions

The task of the GNN is to assign an edge score to every edge of the graph. Therefore, showing the distributions of edge scores for the different combinations of datasets and GNNs begins to evaluate the network performance. A good score distribution shows a clear separation between target true edges and fake or non-target true edges, i.e. target true edges have scores close to 1, while any other edges have scores close to 0. Target true edges in this case are track segments of cosmic muon tracks, non-target true edges correspond to segments of other particle tracks and fake edges are anything not falling into either of these categories.

Figures 3.1, 3.2 and 3.3 each show two score distributions, to highlight the different performances of various GNNs.

The score distributions of the inference on the cosmics-only sample (see Figure 3.1) shows two different results. The network trained on a mixture of cosmics-with-Michel and cosmics-only (Figure 3.1a) produces a well separated score distribution, where non-target true and fake edges are scored low with only a small tail towards high scores while almost all true target edges are scored closely to 1. On the other hand, the network trained with additional Michel-only frames instead of cosmics (Figure 3.1b) is not able to distinguish true target edges from others and just assigns low scores to almost all edges. From all 12 combinations, this is the only one that failed the task. Presumably, this is due to the very much different graphs of Michel-only and cosmics-only frames. With only half of the total number of cosmic muons in the training set this must have led to poor convergence.

Inference on Michel-only frames (see Figure 3.2) shows very different score distributions. Because there are no target true edges in that sample, the expecation is a distribution that is populated exclusively on the low score end. This is indeed the case for inference of the GNN trained on cosmics-with-Michel and additional Michel-only frames (Figure 3.2b). Due to the significant tails towards high scores, the network trained on cosmics-with-Michel and additional cosmics-only frames (Figure 3.2a) does not perform as good as the previous one.

Finally, inference on cosmics-with-Michel frames also shows different performances (see Figure 3.3), depending on the training data of the GNN used for inference. The GNN trained on the fully mixed data (cosmics-with-Michel, Michel-only and cosmics-only) produces a separated score distribution with very low tails in the high score regime for non-signal edges (see Figure 3.3a). However, the true target edges are not as sharpy aggregated at the high score end of the distribution. The GNN trained on data without cosmics-only but added Michel-only frames (Figure 3.3b) produces a sharper peak for true target edges.
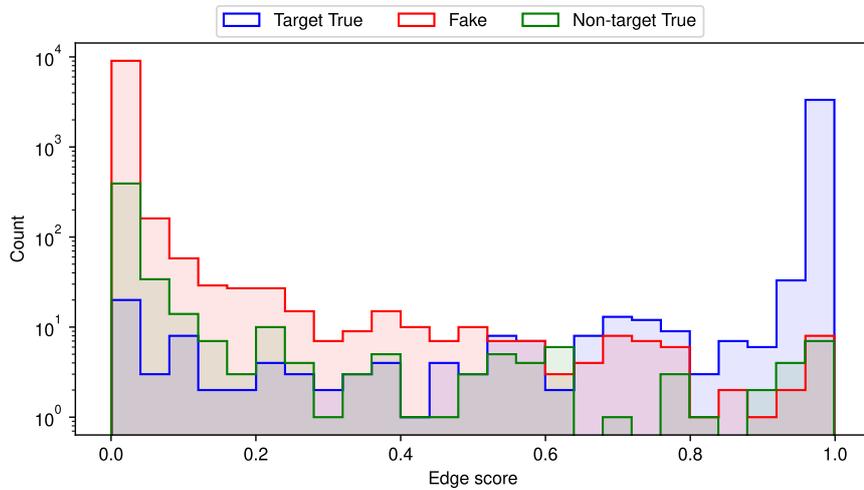
(a) Score distribution of the GNN trained on cosmics-with-Michel and cosmics-only, inferred on cosmics-only.
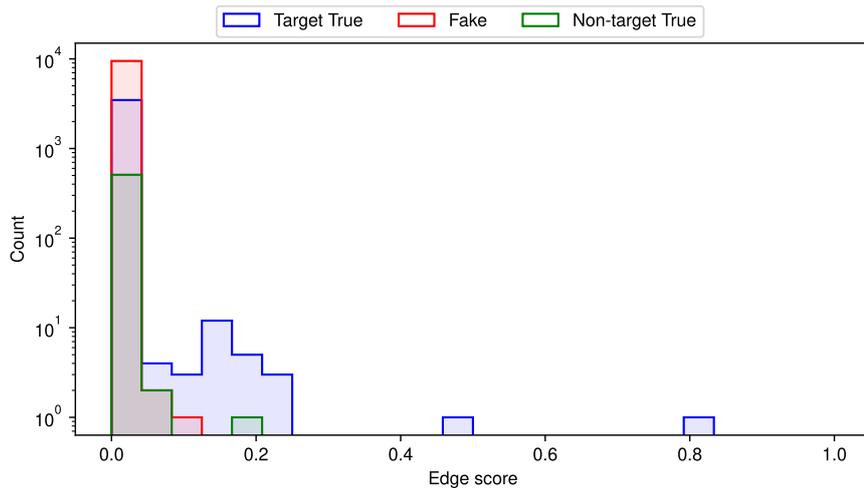
(b) Score distribution of the GNN trained on cosmics-with-Michel and Michel-only, inferred on cosmics-only.

Figure 3.1: Score distributions for 1000 test set frames of the cosmics-only sample for GNNs trained on cosmics-with-Michel with (a) added cosmics-only or (b) Michel-only frames respectively.
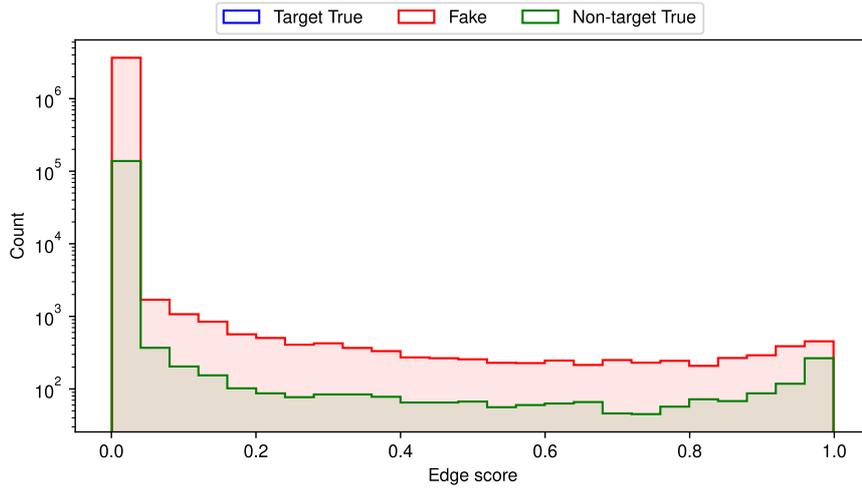
(a) Score distribution of the GNN trained on cosmics-with-Michel and cosmics-only, inferred on Michel decay graphs.
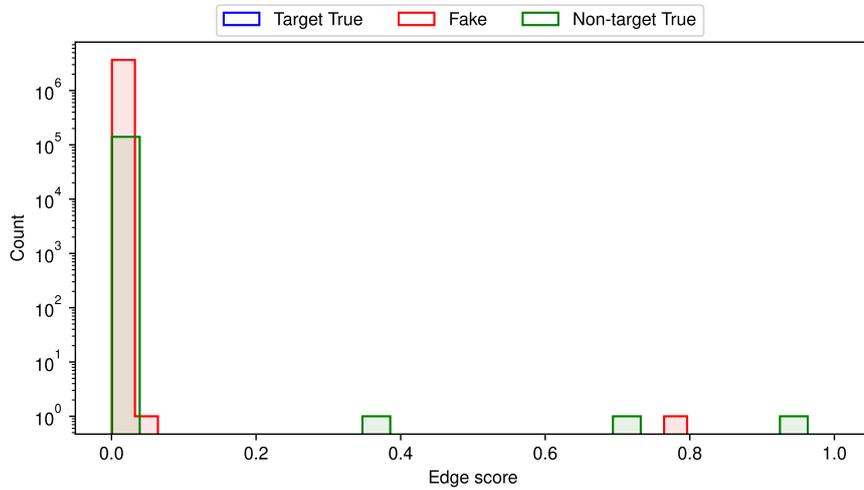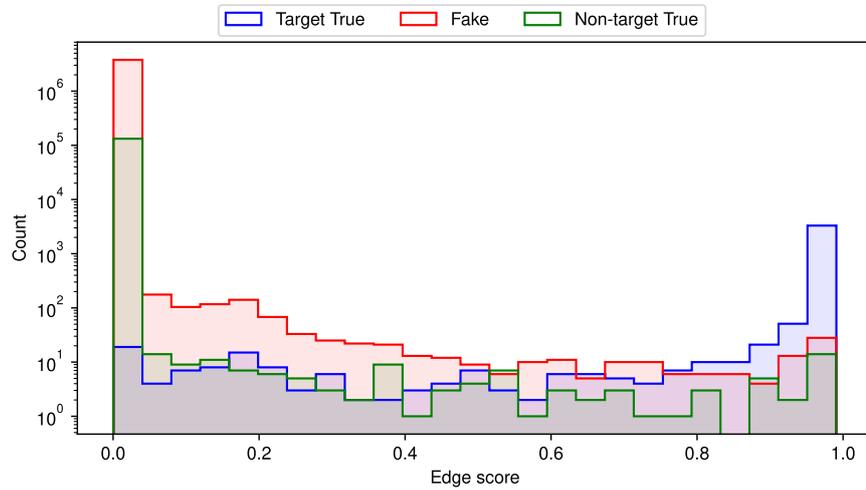

(b) Score distribution of the GNN trained on cosmics-with-Michel and pure Michel decays, inferred on Michel decay graphs.

Figure 3.2: Score distributions for 1000 test set frames of the Michel-only sample for GNNs trained on cosmics-with-Michel with (a) added cosmics-only or Michel-only frames respectively.
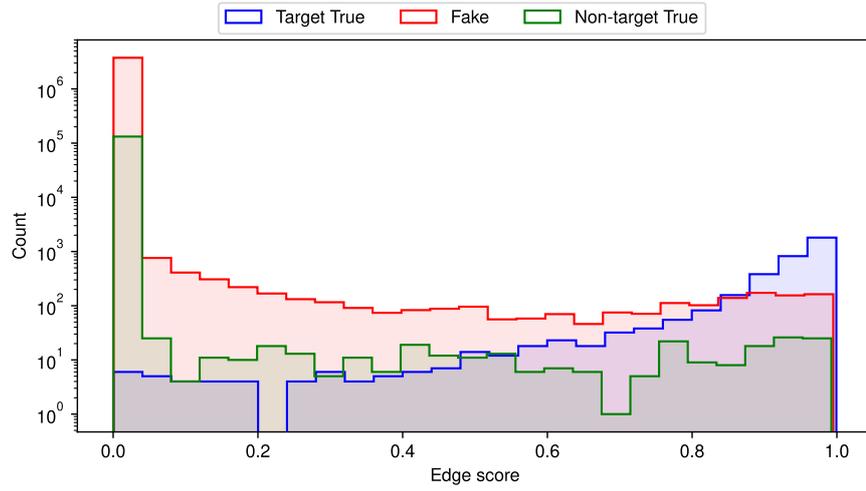
(a) Score distribution of the GNN trained on cosmics-with-Michel, Michel-only and cosmics-only, inferred on cosmics-with-Michel graphs.

(b) Score distribution of the GNN trained on cosmics-with-Michel and Michel-only, inferred on cosmic muon with Michel-only graphs.

Figure 3.3: Score distributions for 1000 test set frames of the cosmics-with-Michel sample for GNNs trained on cosmics-with-Michel with (a) added cosmics-only and Michel-only or (b) just Michel-only frames respectively.

Edgewise Efficiency

In Figures 3.4 and 3.5, the average edgewise efficiencies and purities are plotted against one minus the score cut thresholds ranging from 0.5 to 0.9999. Edges with higher scores than the threshold value are considered true. The number of true target edges, target edges and true edges are counted for every individual graph, summed over the dataset and the efficiencies and purities are calculated using Equations 2.3.3 and 2.3.4. The errors are calculated with the Clopper-Pearson interval at the 1σ confidence level [1].
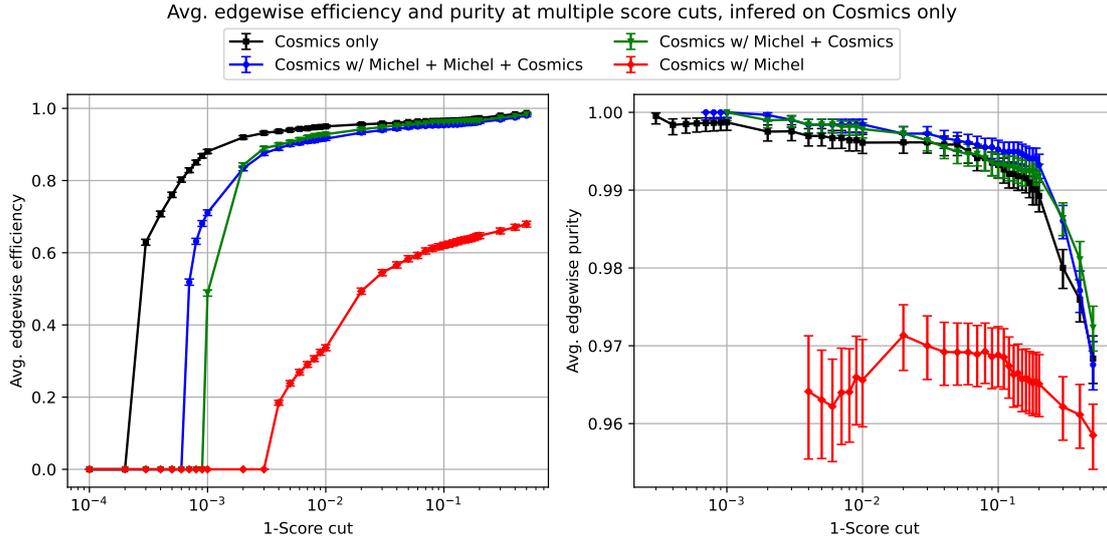
Figure 3.4: Average edgewise efficiency (left) and average edgewise purity (right) of different GNNs inferred on 1000 testset frames of the cosmics-only data sample as functions of $1-$the score cut threshold. The different colors correspond to the different training sets of the infered GNNs. Errors are calculated as the 1σ Clopper-Pearson confidence interval.

Inference on the cosmics-only sample (see Figure 3.4) shows very high efficiencies of $> 90\%$ before rapidly decreasing around a score cut of 0.998, achieved by the GNNs trained on cosmics-only samples (black, blue and green). A significantly worse performance is exhibited by the cosmics-with-Michel trained GNN (red) with edgewise efficiencies of only up to 68%. The GNN trained on cosmics-with-Michel and Michel-only frames is not able to correctly score muon edges (as shown in Figure 3.1b), thus having efficiency 0.
Considering the average edgewise purities a similar picture as for the efficiencies is presented. Again, the networks trained on cosmics-only frames are able to achieve purities of $> 99\%$ until the drop-off. However, the statistical uncertainties are much larger, due to the low number of fake edges in the cosmics-only sample. Therefore, all GNNs trained on cosmics-only lie within each other's confidence interval. The GNN trained on cosmics-with-Michel frames performs significantly worse than the other GNNs at purities mostly below 97%. This shows that the addition of cosmics-only frames into the training set can boost the scoring performance at least on edge level.
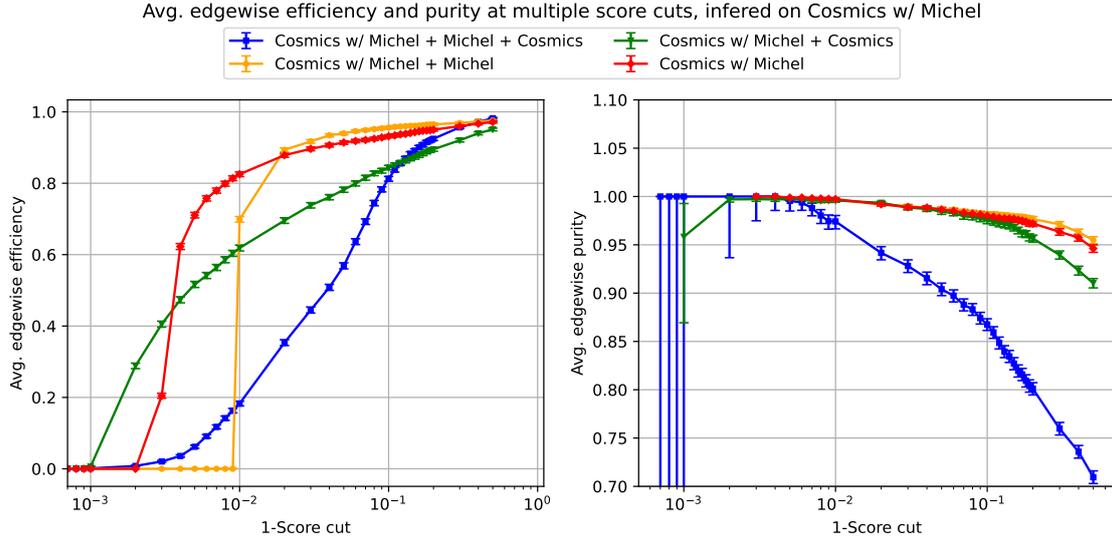
Figure 3.5: Average edgewise efficiency (left) and average edgewise purity (right) of different GNNs inferred on 1000 testset frames of the cosmics-with-Michel data sample as functions of $1-$the score cut threshold. The different colors correspond to the different training sets of the infered GNNs. Errors are calculated as the 1σ Clopper-Pearson confidence interval.

Figure 3.5 also shows the average edgewise efficiency and purity for different GNNs but for the inference case on cosmics-with-Michel. The GNN trained exclusively on cosmics-only (black) is replaced by the GNN trained on cosmics-with-Michel and Michel-only (yellow) to explore how and if adding Michel-only frames in training helps to supress background edges. The efficiencies of the cosmics-with-Michel GNN (red) and the cosmics-with-Michel + Michel-only GNN start off similarly for low score cuts. The efficiency for the red network is almost always about 2% lower than the yellow one. However, while the efficiency of the yellow network drops to zero at score cuts > 0.99, the red network only reaches zero at a score cut of 0.998.

A similar picture is seen for the two GNNs trained on cosmics-only frames as well (blue and green). The blue GNN falls of first at a score cut of only 0.9. Interestingly the green GNN has a lower overall efficiency than red and yellow, however, it reaches zero last at a threshold of 0.999. Concerning the edgewise purities, all networks except for the blue one perform similarly well, with red and yellow above the green one again.

In conclusion, networks which see cosmics-only events in training performed worse for classifying cosmics-with-Michel events. However a boost in performance is observed for inference on comsics-only. Conversely, adding Michel-only events to the training boosted background rejection but worsened inference performance on cosmics-only.

ROC Curve and AUC

The final metric that is considered for the inference step is the ROC curve, as well as the corresponding AUC. The Figures 3.6 and 3.7 show the ROC curves after infering on the cosmics-only sample and cosmics-with-Michel sample. Defining a ROC for Michel-only inference does not make sense because the efficency is not defined.

The ROC curves in Figure 3.6 show very similar performances except for GNNs whose training contained cosmics-only frames. With the AUC of the well-performing networks lying between 0.99850 and 0.99866, they clearly outperform the other two (0.92782 red, 0.82877 yellow), which is expected after considering the edgewise efficiency plots.

Inference on the cosmics-with-Michel sample produces the ROC curves shown in Figure 3.7. Other than the ROC curves for the cosmics-only sample, here the TPR keeps dropping for FPRs as low as $\mathcal{O}(10^{-7})$. Like before, the networks which are trained additionally on pure cosmic muon graphs (blue and green) generally performed worse than the others (red and yellow). The maximum AUC is recorded for the cosmics-with-Michel and Michel-only trained GNN (yellow), at 0.99996, closely followed by the one trained on cosmics-only with Michel-only frames (red) at 0.99995.
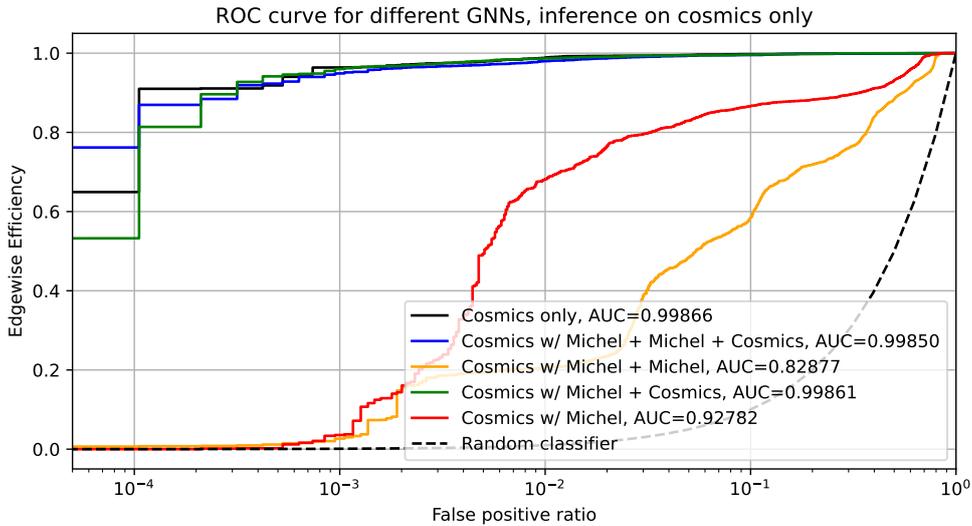

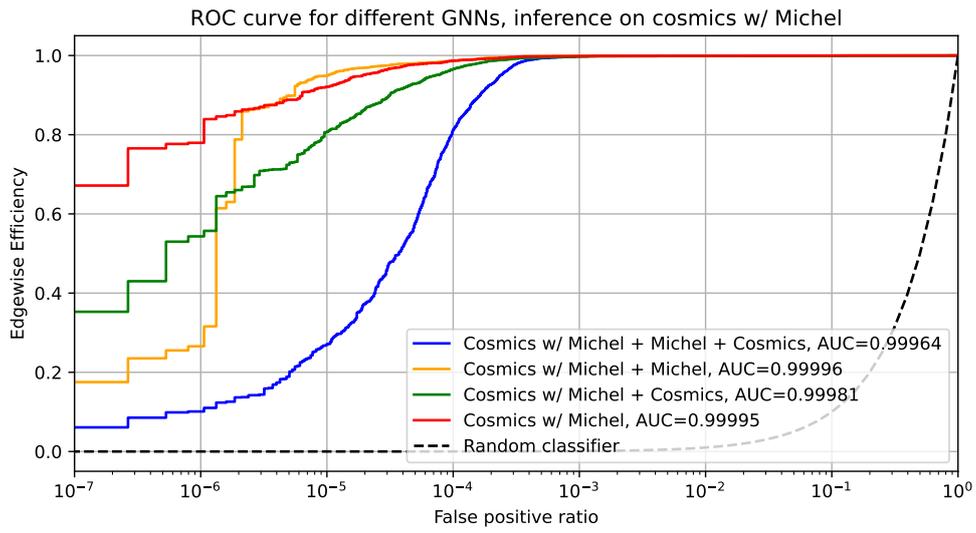Figure 3.6: ROC curves for inference on cosmics-only with differently trained GNNs.

Figure 3.7: ROC curves for inference on cosmics-only with differently trained GNNs.

3.1.2 Track Reconstruction

The final step of the GNN pipeline is the track reconstruction stage. Scored graphs are transformed to connected components to produce track candidates. In the light of testing GNN performance to develop a cosmic muon trigger both the fake track rate needs to be suppressed as good as possible while maintaining high track reconstruction efficiencies. Again, the 12 combinations of inference data set and GNN are considered. It is clear that the reconstruction efficiency, fake track ratio, as well as the acceptance all heavily rely on the chosen score cut and the minimum track length. The following section shows all three metrics in their dependence of the score cut and each other.

Reconstruction on Cosmics-only Graphs

Figure 3.8 shows the track reconstruction efficiency for different GNNs, for a minimum track length of 3 and 4. Again, the network trained on cosmics-with-Michel and Michel-only is replaced by the network trained on cosmics-only exclusively.
The track reconstruction efficiency is plotted against different score cuts for 1000 scored graphs of testset of the cosmics-only sample.
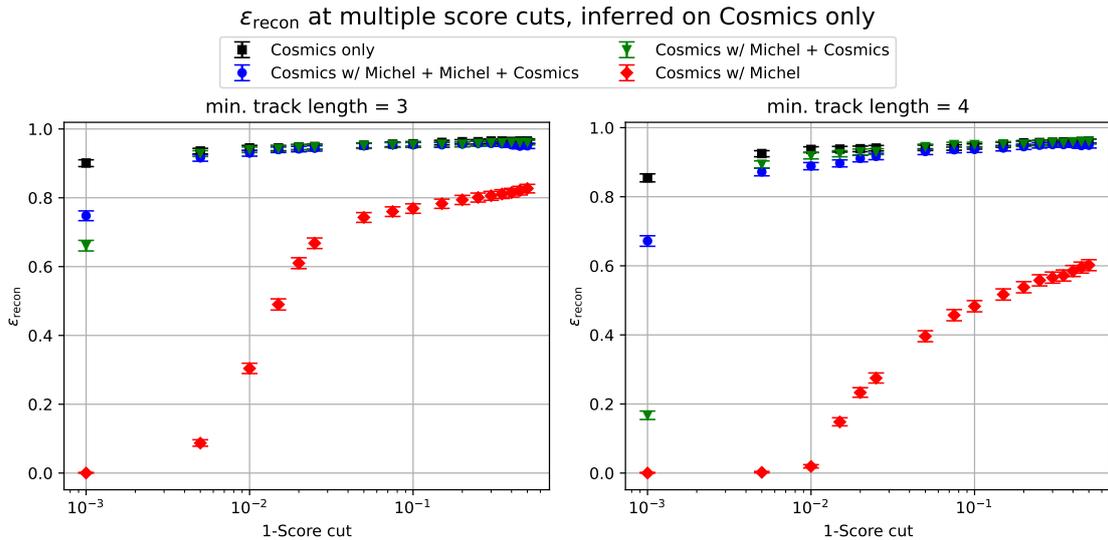

Figure 3.8: Reconstruction efficiency of 1000 cosmics-only testset samples as a function of 1 - the score cut for GNNs of the same architecture, trained on different data samples. The left plot shows the results for a minimum track length of 3, while the right shows the same for a minimum track length of 4. Errors are calculated as the 1σ Clopper-Pearson confidence interval.

The plots in Figure 3.8 show dropping reconstruction efficiencies with increasing score cuts. This is intuitively clear because more and more edges do not make the cut, thus causing the number of reconstructed tracks to drop and hence the reconstruction efficiency. The reconstruction efficiencies of the GNN trained on the fully mixed data (blue), the GNN trained on the cosmics-with-Michel and cosmics-only sample (green) and the one trained exclusively on the cosmics-only sample (black) are significantly higher than the one trained exclusively on cosmics-with-Michel frames (red).

The very different performances in reconstruction performance suggests the assumption, that GNNs which have seen pure cosmic muon tracks in training, provide better performance concerning the reconstruction of such tracks. The reason for this presumably lies within the fundamentally different graph topologies of cosmics-only and graphs containing Michel edges. While graphs from cosmics-only frames only have about 5 nodes and 13 edges on average, the graphs of Michel-only frames on the other hand have $\mathcal{O}(100)$ nodes and $\mathcal{O}(10^3)$ edges on average.

Similar to the plots in Figure 3.8, the fake track ratio can also be plotted as a function of the score cut threshold, see Figure 3.9. Fake tracks in the cosmics-only sample are produced by the few secondary electrons.
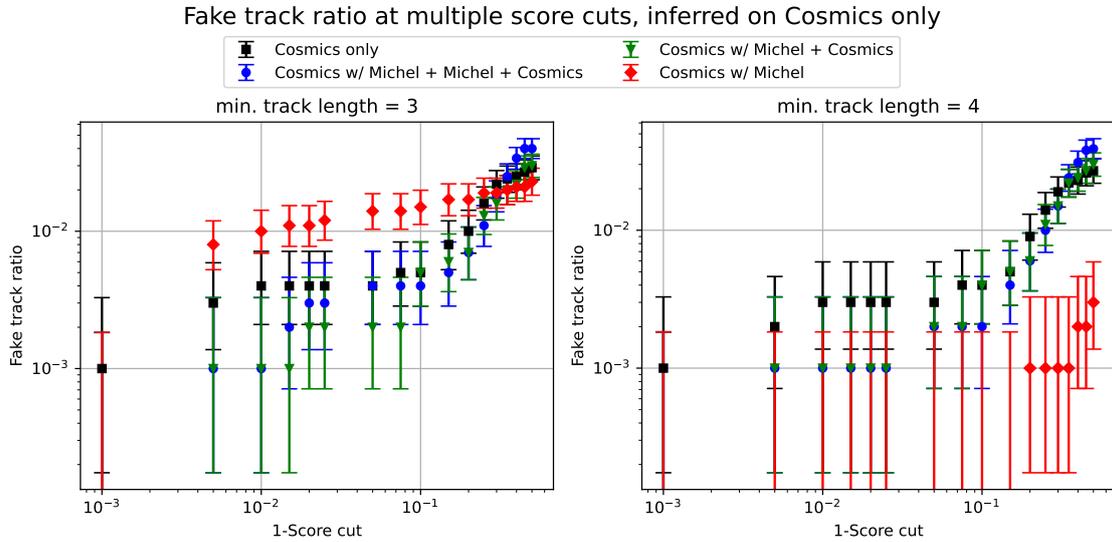

Figure 3.9: Fake track ratio of 1000 cosmics-only testset samples as a function of 1 - the score cut for GNNs of the same architecture, trained on different data samples. The left plot shows the results for a minimum track length of 3, while the right shows the same for a minimum track length of 4. Errors are calculated as the 1σ Clopper-Pearson confidence interval.

Like before, for the minimum track length 3 case, all GNNs except for the one trained just on cosmics-with-Michel produce low fake track ratio of down to $\mathcal{O}(10^{-3})$. However, for a minimum track length of 4. the red network outperforms the others by an order of magnitude. Due to the large errors in the fake ratio, the three well-performing GNNs deviate statistically insignificantly from each other.

By plotting the reconstruction efficiency against the fake track ratio (see Figure 3.10), the trade-off between both is illustrated. The aim is to achieve high reconstruction efficiencies while keeping the fake track ratio as low as possible. Because the errors in both reconstruction efficiency and fake track ratio are shown in Figures 3.8 and 3.9 they are left out for better visibility.
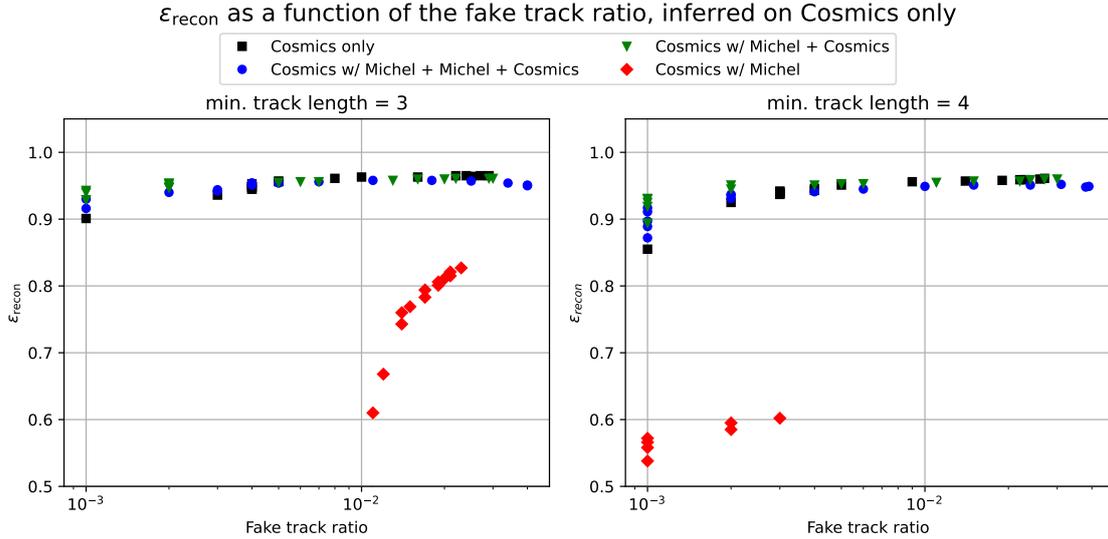
Figure 3.10: Reconstruction efficiencies of the cosmic muon sample as a function of the fake track ratio for GNNs of the same architecture, trained on four different data samples. The left plot shows the results for a minimum track length of 3, while the right shows the same for a minimum track length of 4. Error bars are omitted for visibility.

Consequently, a similar picture as in Figures 3.8 and 3.9 emerges. The GNNs trained on cosmics-only lie within their respective statistical error intervals at high efficiencies between 0.9 and 0.95 for all fake ratios. The GNN trained exclusively on cosmics-with-michel performs significantly worse with efficiencies below 0.85 and drops below 0.6 for a fake track ratio of about 10^{-2}.

Reconstruction on Michel-only Graphs

Michel-only frames constitute the vast amount of background frames, the trigger is not supposed to select. By testing the inference on Michel-only frames, the background rejection can be determined. As mentioned before, plotting the reconstruction efficiency for inference on the Michel-only data samples does not yield any results and is omitted.

In Figure 3.11, the acceptance is plotted against different score cut thresholds for the four GNNs, different data samples and track lengths. However, considering only 1000 frames showes heavy statistical limitations for both GNNs trained on additional Michel-only frames (yellow and blue), because both networks produce less than 10 fake tracks in 1000 graphs. To increase the resolution of the acceptance up to 6.2×10^{-5}, inference on a total of 16200 graphs is performed.
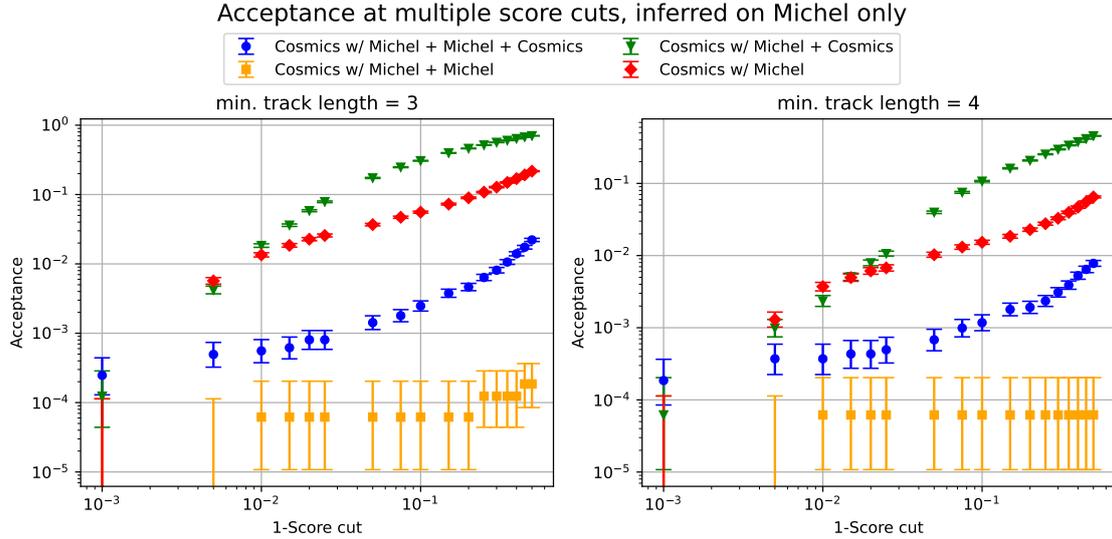
Figure 3.11: Acceptances of the Michel-only sample as a function of the score cut for GNNs of the same architecture, trained on four different data samples. The left plot shows the results for a minimum track length of 3, while the right shows the same for a minimum track length of 4. A total number of 16200 graphs is considered to improve statistics. Errors are calculated as the 1σ Clopper-Pearson confidence interval.

Figure 3.11 shows a similar picture as Figure 3.9 (fake ratio for inference on cosmics-only). GNNs that are trained on the kind of data samples used in inference, tend to perform significantly better.

The best network concerning acceptance of Michel-only frames is the GNN trained on cosmics-with-Michel with Michel-only frames (yellow). Sill statistically limited, only three false triggers are recorded in the 16200 test frames. Score cuts of 0.8 and above already hit the minimum acceptance of 6.2×10^{-5}. Second best is the GNN trained on the fully mixed training set (blue) with acceptances mostly below 10^{-2} and down to 2.4×10^{-4} for a score cut of 0.999. The networks without Michel-only frames in training (red and green) perform significantly worse than the ones discussed earlier.

Choosing a longer minimum track length of 4 helps lowering the acceptances of the red and green GNNs by about a factor 3. Further increasing the minimum number of hits per track, however, is not advisable when considering Figure 2.7: About 70% of muon tracks do not have more than 4 hits, thus making a cut on tracks longer than 4 would at least remove 70% of triggered frames, not considering faulty or partial reconstructions.

Reconstruction on Cosmics-with-Michel Graphs

In the following, the results from performing track reconstruction on the cosmics-with-Michel data sample are presented. These samples carry significance, because these are the kinds of graphs the cosmic muon trigger is supposed to select for alignment.

Analogously to the sections on cosmics-only and Michel-only frames, the reconstruction efficiency plotted against different score cuts is shown in Figure 3.12.
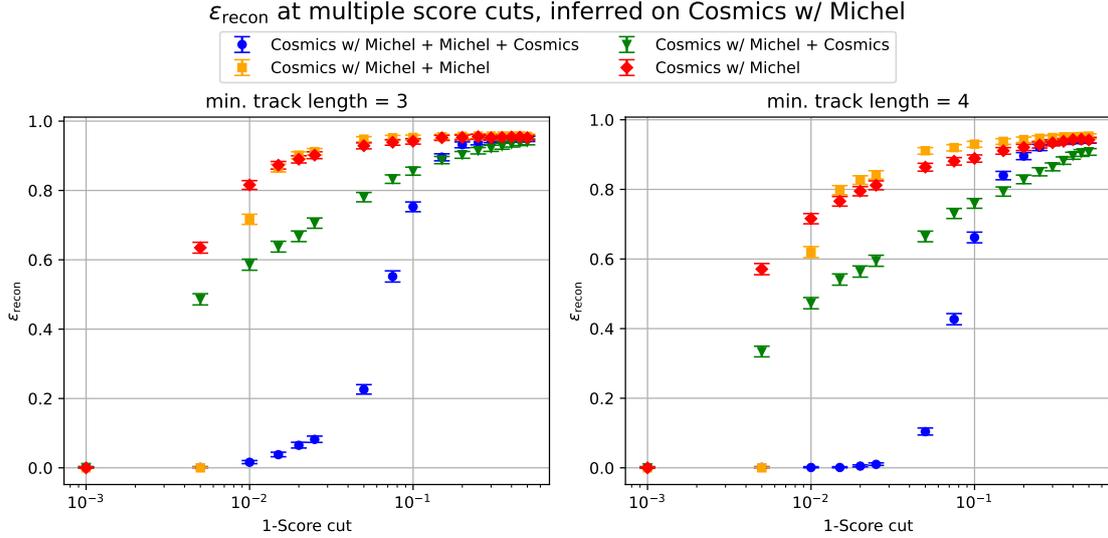

Figure 3.12: Reconstruction efficiencies of the cosmics-with-Michel sample as a function of the score cut for GNNs of the same architecture, trained on four different data samples. The left plot shows the results for a minimum track length of 3, while the right shows the same for a minimum track length of 4. Errors are calculated as the 1σ Clopper-Pearson confidence interval.

The best reconstruction efficiencies are achieved by the GNNs without Michel-only frames in training (red and yellow). The reached efficiencies lie at > 0.95 and only drop below 0.8 for score cuts above 0.99. Similar behaviour at lower overall reconstruction efficiencies is observed for the reconstruction of minimum length 4 tracks. The network trained on cosmics-with-Michel and cosmics-only (green) shows a similar progression but with lower overall efficiencies. Even though the fully mixed network started out like the red and yellow one, it quickly dropped in efficiency for score cuts greater than 0.9.

Considering the plot of fake track ratio versus score cut, all networks, except for the one trained on the fully mixed data (blue), exhibit very similar acceptances again lying within their statistical errors (see Figure 3.13).
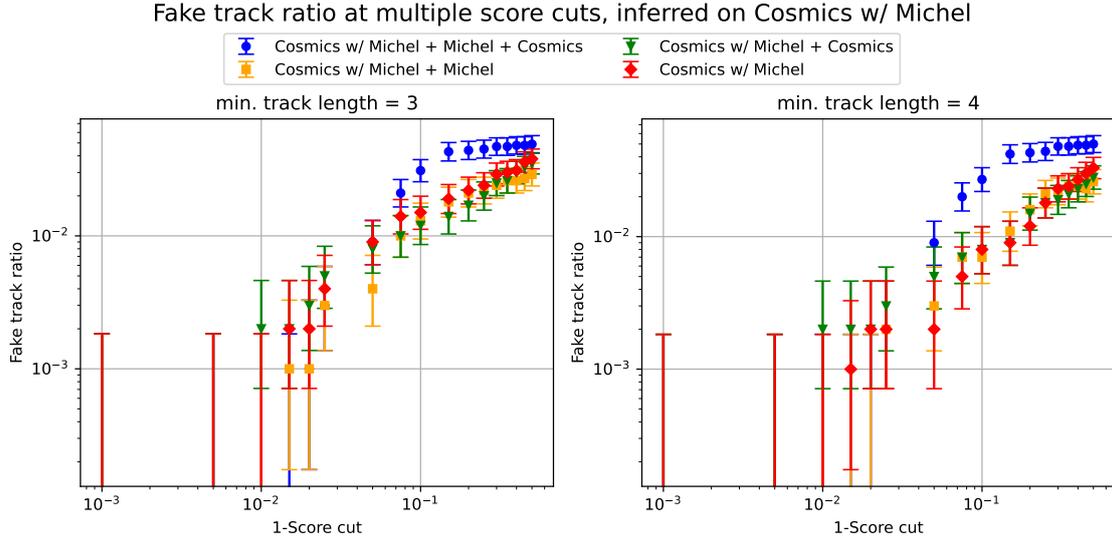
Figure 3.13: Fake track ratios of the cosmics-with-Michel sample as a function of the score cut for GNNs of the same architecture, trained on four different data samples. The left plot shows the results for a minimum track length of 3 while the right plot shows the same for a minimum track length of 4. Errors are calculated as the 1σ Clopper-Pearson confidence interval.
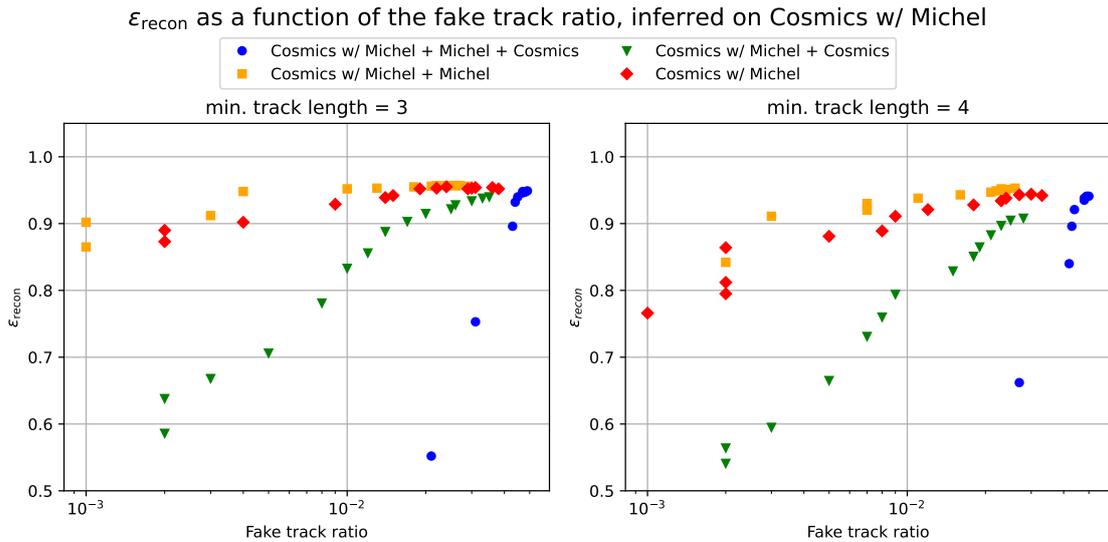

Figure 3.14: Reconstruction efficiencies of the cosmics-with-Michel decays sample as a function of the acceptance for GNNs of the same architecture, trained on four different data samples. The left plot shows the results for a minimum track length of 3, while the right shows the same for a minimum track length of 4. Error bars are omitted for visibility.

Figure 3.14 presents the reconstruction efficiencies of the four GNNs plotted against their corresponding fake track ratios for inference on cosmics-with-Michel. Like before, the networks trained on samples not containing Michel-only frames outperform the other two GNNs. They produce efficiencies mostly above 0.9 for the minimum length three tracks and above 0.8 for minimum length four tracks for the whole range of fake track ratios.

Second best is the network trained with additional cosmics-only (green), which only drops below 0.8 for fake track ratios of about 10^{-2}. Again, the network trained on the fully mixed sample (blue) exhibits the worst peformance.

Background Rejection and Signal Acceptance

By plotting the reconstruction efficiency for inference on the cosmics-with-Michel sample against the acceptance for inference on the Michel-only sample, a working point with high reconstruction efficiency and low acceptance can be found. This is shown in Figure 3.15.
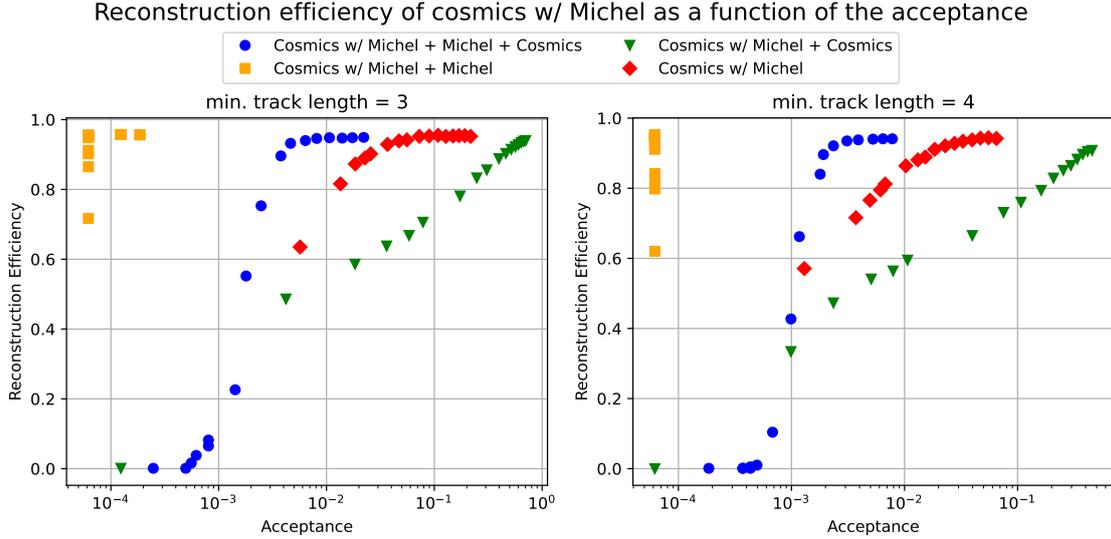

Figure 3.15: Reconstruction efficiencies for the cosmics-with-Michel sample plotted against the acceptance for inference on the Michel-only sample. Error bars are omitted for visibility.

The GNNs exhibited, show very different performances. While the reconstruction efficiency generally is quite high at over 90% in most instances, the acceptances are shifted by about an order of magnitude for all GNNs, except for the one trained on cosmics-with-Michel and Michel-only (yellow), which has a maximum acceptance of 1.9×10^{-4}. With the goal in mind to build a cosmic muon trigger, data points at the low end of the acceptance scale and at high reconstruction efficiencies are preferred. The clear favorite in that regard is the GNN trained on a mixture of cosmics-with-Michel frame and Michel-only frames. At a reconstruction efficiency of 95.6% for a minimum track length of 3 and 95.3% for a minimum track length of 4 the minimum acceptance of 6.2×10^{-5} is reached. Still being statistically limited and considering the curves of the other GNNs, the acceptance for reconstruction efficiencies above 90% is probably even lower.

The fully mixed GNN (blue) has the second best performance, with the reconstruction rate crossing the 90% only at acceptances of 3.8×10^{-3} (minimum track length 3) and 1.9×10^{-3} (minimum track length 4) respectively.

The conclusion from this is, that the GNN which is trained on additional Michel-only frames, highly outperforms the others regarding inference on cosmics-with-Michel. However, for the beam-off case, i.e. cosmics-only frames, this network completely fails. Therefore using a different model for alignment with no beam is advised.

3.2 Discussion of the Results

3.2.1 Summary

This thesis explores the viability of using graph neural networks for a cosmic muon trigger on Mu3e data. A GNN architecture is presented, which is able to correctly classify edges depending on them being part of a cosmic muon track or not. Furthermore, the influence of altering the training data by introducing mixed training samples is investigated. Finally, the most important parameters of a trigger system, the background rejection and signal efficiency, are estimated.

Inference

In Section 3.1.1 the inference step of the pipeline is evaluated on the basis of four GNNs with the same hyperparameters but trained on different mixtures of simulation data.

Inference on cosmics-only frames shows a boost in edgewise efficiency and purity for the networks, which are trained on mixtures of samples containing cosmics-only graphs. Conversely, GNNs which are trained on Michel-only events as well, either completely fail the task or show significant shrinkage in all metrics. Similarly, when inferring on cosmics-with-Michel frames, the GNNs seeing cosmics-only events during training perform much worse than their counterparts. GNNs which had Michel-only events added in the mix even show a slight boost in performance for inference on cosmics-with-Michel events.

The explaination for that lies within the fundamentally different graph structures of cosmics-only and cosmics-with-Michel/Michel-only. While the former only counts about 5 nodes with 13 edges on average, Michel-graphs contain about 80 nodes and 4000 edges on average. Because of the training, the network expects a certain graph structure. Thus, the network is unable to generalize to test data, which does not follow the expected structure.

Track Reconstruction

The section on the evaluation of the track reconstruction step (Section 3.1.2), shows significant differences in the performance for differently trained GNNs. The goal of rejecting as many Michel-only frames as possible is best achieved by a GNN trained on 50% cosmics-with-Michel and 50% Michel-only samples. The achieved acceptance is statistically limited to 6.2×10^{-5} at a reconstruction efficiency for cosmic muons of 95.6% for tracks of minimum length 3 and 95.2% for tracks of length 4 and above.

However, this network completely fails inference and track reconstruction of cosmics-only frames, thus providing no value for the use at the beam-off case. In that regard, either

of the GNNs trained with added cosmics-only frames performes well, with reconstruction efficiencies $> 95\%$ throughout.

In summary, it is shown that a highly specialized GNN (trained on cosmics-with-Michel and Michel-only) outperforms the others by far with respect to background rejection. A more versatile GNN, able to handle the beam-on and beam-off case is realized in the form of the fully mixed GNN. It reaches track reconstruction efficiencies of 89.6% at an acceptance of 3.8×10^{-3} for a minimum track length of 3) and 1.9×10^{-3} for a minimum track length of 4.

Due to the 30 times higher background rejection of 6.2×10^{-5} and under the assumption, that the real acceptance most like is even lower, the following working point for the GNN trained on cosmics-with-Michel and Michel-only for a minimum track length of 3 is proposed:

- Score cut: 0.8

- Acceptance: $\leq 6.2 \times 10^{-5}$

- Reconstruction efficiency (Cosmics-with-Michel): 95.6%

To put these numbers into perspective, assuming a frame rate of $\frac{1}{64\,\text{ns}} = 15.625 \times 10^6\,\text{s}^{-1}$, only around 1000 frames per second would have to be reconstructed, while 95.6% of cosmic muon frames are triggered.

For future analysis, a different training split where the percentage of cosmics-only graphs is reduced, while increasing Michel-only graphs, could be worth investigating further to produce a network capable of handling both beam-on and -off.

3.2.2 Conclusion

In conclusion, the employment of graph neural networks for the reconstruction of cosmic muon tracks is able to achieve high reconstruction efficiencies of over 95% while suppressing the number of accepted frames by a factor of $< 6.2 \times 10^{-5}$. With this being a first step towards a GNN-based cosmic muon trigger, improvements in terms of acceptance are to be expected. Because of that, the approach of employing graph neural networks for the identification and reconstruction of cosmic muon frames is feasible and should be pursued further.

3.2.3 Outlook

Training Data

The clear dependence of the GNN's performance on the training data is evident from the last chapter. When inferred on graphs including both cosmic muon and Michel electron edges, the networks which are trained on datasets with pure cosmic muon graphs in them performed significantly worse in all metrics except for the fake track rate. However, inference on cosmic muon graphs only showed massive improvements for the networks trained on such graphs.

To obtain a more well-rounded network, capable of reconstructing both cosmics-only and cosmics-with-Michel graphs, a better training data mixture has to be found. The mixed training data for the networks are comprised of equal mixtures of the three data samples (50/50 or 33/33/33). The findings above suggest that a split more in favour of cosmics-with-Michel and Michel-only graphs may improve the network performance. This for sure is an interesting aspect, worth further exploration by analyzing different mixing ratios and comparing their performances.

In this work, the influence of electronic noise is not taken into account. By comparing the acceptance of electronic noise frames and reconstruction efficiency of cosmics-only frames, a working point for the GNN handling the beam-off case can be found.

Throughput on a GPU or an FPGA

In the end, a cosmic muon trigger would be implemented as a part of the readout scheme on a GPU or FPGA. This neccessitates a sufficiently fast throughput of frames, able to keep up with the expected decay rate of up to $\mathcal{O}(10^8)$ muons per second for phase I.

With this thesis being focused on the physics performance of a GNN-based trigger, no analysis of the network throughput has been made yet. Should a GNN-based cosmic muon trigger be implemented at Mu3e, this will be a decisive metric which has to be investigated.

To give a rough estimate of the ressources needed, a few numbers during training and inference are listed in the following.

For a training of 100 epochs on 10^4 train set graphs, and 10^3 validation and test set graphs, using the network architecture from 2.1 on a NVIDIA RTX A6000, the training duration lies between 6 and 8 hours with a maximum memory usage of about 800 MB. Inference of 12000 Michel graphs takes around 220 s, while track reconstruction takes approximately 110 s. Of course these numbers do not represent precise estimates but rather give a sense of scale.

Metric Learning

This thesis exclusively dealt with fully connected graphs, because the event size is small enough and presented the most simplistic way of graph construction. As discussed, due to the quadratic scaling of graph size with respect to the number of nodes this method is not practical for larger events, e.g. events produced at high energy collider experiments such as ATLAS.

One method that can be used to obtain highly efficient graphs while keeping the total number of edges low is called metric learning. The working principle is to train a regular MLP to embed node features into a D-dimensional latent space, such that hits within a threshold radius r calculated as the Euclidian distance

$$r = \left(\sum_{i=1}^{D} (v_i - w_i)^2 \right)^{\frac{1}{2}},$$
(3.2.1)

where v_i and w_i are the embedded node feature vector components, are connected by edges. This is shown to decrease the number of edges per graph at ATLAS ITk events from $\mathcal{O}(10^{10})$ edges down to $\mathcal{O}(10^6)$ [23].

Assuming that cosmic muon tracks have a fundamentally different topology compared to Michel decays, it might be possible to train a metric learning model to create highly pure graphs.

Bibliography

1. Clopper, C. J. & Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* **26,** 404–413. ISSN: 0006-3444. eprint: `https://academic.oup.com/biomet/article-pdf/26/4/404/823407/26-4-404.pdf`. `https://doi.org/10.1093/biomet/26.4.404` (Dec. 1934).

2. Bellgardt, U. *et al.* Search for the Decay $\mu^+ \to e^+e^+e^-$. *Nucl. Phys. B* **299,** 1–6 (1988).

3. Galbraith, D. & Burgard, C. *Standard Model of Physics* 2012. `http://davidgalbraith.org/portfolio/ux-standard-model-of-the-standard-model/`.

4. Blondel, A. *et al. Research Proposal for an Experiment to Search for the Decay $\mu \to eee$* Jan. 2013. arXiv: `1301.6113 [physics.ins-det]`.

5. Thomson, M. *Modern Particle Physics* (Cambridge University Press, 2013).

6. Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D. & Kavukcuoglu, K. *Interaction Networks for Learning about Objects, Relations and Physics* 2016. arXiv: `1612.00222 [cs.AI]`. `https://arxiv.org/abs/1612.00222`.

7. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* 2017. arXiv: `1412.6980 [cs.LG]`. `https://arxiv.org/abs/1412.6980`.

8. Vom Bruch, D. *Pixel Sensor Evaluation and Online Event Selection for the Mu3e Experiment* PhD thesis (Heidelberg University, 2017). `http://www.ub.uni-heidelberg.de/archiv/23689`.

9. Lechner, A. Particle interactions with matter. *CERN Yellow Rep. School Proc.* **5,** 47. `https://cds.cern.ch/record/2674116` (2018).

10. Shukla, P. & Sankrith, S. Energy and angular distributions of atmospheric muons at the Earth. arXiv: `1606.06907 [hep-ph]`. `https://arxiv.org/abs/1606.06907` (2018).

11. Hartenstein, U. *Track Based Alignment for the Mu3e Pixel Detector* PhD thesis (Johannes Gutenberg University Mainz, 2019). `https://www.psi.ch/sites/default/files/2019-05/DissertationHartenstein.pdf`.

12. Pérez-Enciso, M. & Zingaretti, L. A Guide for Using Deep Learning for Complex Trait Genomic Prediction. *Genes* **10.** `https://doi.org/10.3390/genes10070553` (2019).

13. Xu, M. Understanding graph embedding methods and their applications. `https://arxiv.org/abs/2012.08019` (2020).

14. Arndt, K. *et al.* Technical design of the phase I Mu3e experiment. *Nucl. Instrum. Meth. A* **1014,** 165679. arXiv: `2009.11690 [physics.ins-det]` (2021).

15. Biscarat, Catherine, Caillou, Sylvain, Rougier, Charline, Stark, Jan & Zahreddine, Jad. Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC. *EPJ Web Conf.* **251,** 03047. `https://doi.org/10.1051/epjconf/202125103047` (2021).

16. DeZoort, G. *et al.* Charged Particle Tracking via Edge-Classifying Interaction Networks. *Computing and Software for Big Science* **5** (2021).

17. ATLAS Collaboration. *Track finding performance plots for a Graph Neural Network pipeline on ATLAS ITk Simulated Data* `https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/IDTR-2022-01/`.

18. Caillou, S. *et al. ATLAS ITk Track Reconstruction with a GNN-based pipeline* tech. rep. (CERN, Geneva, 2022). `https://cds.cern.ch/record/2815578`.

19. Duarte, J. & Vlimant, J.-R. in *Artificial Intelligence for High Energy Physics* 387–436 (WORLD SCIENTIFIC, Feb. 2022). ISBN: 9789811234033. `http://dx.doi.org/10.1142/9789811234033_0012`.

20. Navas, S. *et al.* Review of particle physics. *Phys. Rev. D* **110,** 030001. `https://doi.org/10.1103/PhysRevD.110.030001` (2024).

21. Atkinson, M. J. *et al. GNN4ITK* `https://github.com/GNN4ITkTeam/CommonFramework`.

22. *Atmospheric Collision* `https://commons.wikimedia.org/wiki/File:Atmospheric_Collision.svg`.

23. Dittmeier, S. *Track reconstruction for the ATLAS Phase-II Event Filter using GNNs on FPGAs* `https://indico.desy.de/event/42884/contributions/159264/attachments/88982/119459/dpg2024_dittmeier_final.pdf`.

24. Kılıç, İ. *ROC Curve and AUC: Evaluating Model Performance* `%7Bhttps://medium.com/@ilyurek/roc-curve-and-auc-evaluating-model-performance-c2178008b02%7D`.

25. Murnane, D. T. & Ju, X. *Tracking GNN Walk Through* `https://indico.cern.ch/event/852553/contributions/4062229/attachments/2128020/3583808/Metric%20Learning%20and%20GNNs%20for%20Tracking%20-%20ExatrkX.pdf`.

26. *PyTorch* `https://pytorch.org/`.

27. *PyTorch Geometric* `https://pytorch-geometric.readthedocs.io/en/latest/#`.

28. *Weights and Biases* `https://wandb.ai/site`.

Acknowledgements

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 07.10.2024,