Methods for Estimating Mass-Sensitive Observables of Ultra-High Energy Cosmic Rays using Artificial Neural Networks

To obtain the academic degree

Doctor of Science

from the Faculty of Physics of the
Karlsruhe Institute of Technology (KIT)
and
from the Instituto de Tecnología "Prof. Jorge A. Sábato" de la
Universidad Nacional de San Martín (UNSAM)

submitted

dissertation

of

Steffen Traugott Hahn
born in Heilbronn

Day of the oral exam: 16.12.2022

Referent: Prof. Dr. Ralph Engel

Co-referent: Prof. Dr. Brian Wundheiler

Supervisor(s): Dr. Markus Roth, Dr. Darko Veberič

8.11.2022

Methods for Estimating Mass-Sensitive Observables of Ultra-High Energy Cosmic Rays using Artificial Neural Networks

Para optar por el título de

Doctor en Ciencias Naturales

del Instituto de Tecnología "Prof. Jorge A. Sábato" de la
Universidad Nacional de San Martín (UNSAM)

y

de la facultad de física del
Instituto de Tecnología de Karlsruhe (KIT)

presentado

disertación

de

Steffen Traugott Hahn

nacido en Heilbronn

Día del examen: 16.12.2022

Director: Prof. Dr. Ralph Engel

Co-Director: Prof. Dr. Brian Wundheiler

Colaborador(es): Dr. Markus Roth, Dr. Darko Veberič

8.11.2022

Keywords: cosmic rays, air-shower physics, deep neural network analysis

54 68 65 20 6D 6F 73 74 20 6D 65 72 63 69 66 75 6C 20 74 68 69 6E
67 20 69 6E 20 74 68 65 20 77 6F 72 6C 64 2C 20 49 20 74 68 69 6E
6B 2C 20 69 73 20 74 68 65 20 69 6E 61 62 69 6C 69 74 79 20 6F 66
20 74 68 65 20 68 75 6D 61 6E 20 6D 69 6E 64 20 74 6F 20 63 6F 72
72 65 6C 61 74 65 20 61 6C 6C 20 69 74 73 20 63 6F 6E 74 65 6E 74
73 2E 20 57 65 20 6C 69 76 65 20 6F 6E 20 61 20 70 6C 61 63 69 64
20 69 73 6C 61 6E 64 20 6F 66 20 69 67 6E 6F 72 61 6E 63 65 20 69
6E 20 74 68 65 20 6D 69 64 73 74 20 6F 66 20 62 6C 61 63 6B 20 73
65 61 73 20 6F 66 20 69 6E 66 69 6E 69 74 79 2C 20 61 6E 64 20 69
74 20 77 61 73 20 6E 6F 74 20 6D 65 61 6E 74 20 74 68 61 74 20 77
65 20 73 68 6F 75 6C 64 20 76 6F 79 61 67 65 20 66 61 72 2E 20 54
68 65 20 73 63 69 65 6E 63 65 73 2C 20 65 61 63 68 20 73 74 72 61
69 6E 69 6E 67 20 69 6E 20 69 74 73 20 6F 77 6E 20 64 69 72 65 63
74 69 6F 6E 2C 20 68 61 76 65 20 68 69 74 68 65 72 74 6F 20 68 61
72 6D 65 64 20 75 73 20 6C 69 74 74 6C 65 3B 20 62 75 74 20 73 6F
6D 65 20 64 61 79 20 74 68 65 20 70 69 65 63 69 6E 67 20 74 6F 67
65 74 68 65 72 20 6F 66 20 64 69 73 73 6F 63 69 61 74 65 20 6B
6E 6F 77 6C 65 64 67 65 20 77 69 6C 6C 20 6F 70 65 6E 20 75 70 20
73 75 63 68 20 74 65 72 72 69 66 79 69 6E 67 20 76 69 73 74 61 73
20 6F 66 20 72 65 61 6C 69 74 79 2C 20 61 6E 64 20 6F 66 20 6F 75
72 20 66 72 69 67 68 74 66 75 6C 20 70 6F 73 69 74 69 6F 6E 20 74
68 65 72 65 69 6E 2C 20 74 68 61 74 20 77 65 20 73 68 61 6C 6C 20
65 69 74 68 65 72 20 67 6F 20 6D 61 64 20 66 72 6F 6D 20 74 68 65
20 72 65 76 65 6C 61 74 69 6F 6E 20 6F 72 20 66 6C 65 65 20 66 72
6F 6D 20 74 68 65 20 64 65 61 64 6C 79 20 6C 69 67 68 74 20 69 6E
74 6F 20 74 68 65 20 70 65 61 63 65 20 61 6E 64 20 73 61 66 65 74
79 20 6F 66 20 61 20 6E 65 77 20 64 61 72 6B 20 61 67 65 2E

48 2E 20 50 2E 20 4C 6F 76 65 63 72 61 66 74

I have used the DALL·E 2 [C:1] to create the pictures found at the beginning of each chapter. Since DALL·E 2 is a software framework based on neural networks that generates images from text prompts, I included these pictures since they demonstrate the potential of neural networks.

English Abstract

Ultra-high-energy cosmic rays are the most energetic, naturally occurring particles known to humankind. Being two orders of magnitude beyond the energy scale of the current generation of accelerators, the questions of how the cosmic rays obtain their energy and where they originate from are still a mystery. Moreover, the excess of muons in the decay products of ultra-high-energy cosmic rays in Earth's atmosphere also challenges our understanding of hadronic interactions at the highest energies. To solve the mysteries around cosmic ray physics, it is integral to identify the masses of the incoming cosmic rays. The separation of heavy cosmic rays from light cosmic rays allows for studies of the arrival directions to match minimally-deflected cosmic rays to potential source candidates. In addition, since the number of nucleons is directly related to the number of muons produced, it also grants us the ability to cross-test hadronic interaction models on real measurements.

Due to the low flux of ultra-high-energy cosmic rays, large detectors for the indirect detection of cosmic rays are required to obtain enough statistics. The interaction of a cosmic ray with Earth's atmosphere triggers a cascade of secondary particles called air shower. The Pierre Auger Observatory is the largest cosmic-ray observatory in the world, covering an effective detection area of over $3000\,\mathrm{km}^2$ specifically designed to detect air showers. The observatory has a hybrid design, using the atmosphere as a calorimeter to observe the longitudinal shower development with fluorescence detectors and a surface detector of regularly arranged detector stations on ground level to measure the amount of produced secondary particles. Commonly, the particles arriving at the ground are referred to as shower footprint.

The recent rise in the popularity of artificial neural networks granted physicists new, easy-to-use tools to approach physics problems in a data-driven way. Using neural networks presents an opportunity to relate detailed data, such as shower footprints, to physical observables, such as the energy of an air shower, without the need for analytical model building.

The central objective of this thesis is the investigation of methods based on neural networks to extract information from data taken by the surface detector of the Pierre Auger Observatory that is correlated to the mass of the cosmic rays. To achieve this, I considered two different approaches and explored the benefit of using neural networks. The first approach is based on the extraction of the muon content in each station of the surface detector and the second approach is based on the measurement of the entire shower footprint. Using Monte Carlo simulation studies, I discarded the first approach in favor of the second, which showed more promising results. From this simulation study, I selected three different neural network models trained to predict the depth of the shower maximum, relative muon content, and logarithmic mass from shower footprints. As a final step, I applied the networks on air showers measured by the Pierre Auger Observatory to estimate the mass composition of ultra-high-energy cosmic rays.

German Abstract

Die ultrahochenergetische kosmische Strahlung besteht aus den energiereichsten, natürlich vorkommenden Teilchen, die der Menschheit bekannt sind. Da sie zwei Größenordnungen jenseits der Energieskala der derzeitigen Generation von Teilchenbeschleunigern liegt, sind die Fragen, wie die kosmische Strahlung ihre Energie erhält und woher sie stammt, noch immer ein Mysterium. Darüber hinaus stellt der Überschuss an Myonen in den Zerfallsprodukten der ultrahochenergetischen kosmischen Strahlung in der Erdatmosphäre auch unser Verständnis der hadronischen Wechselwirkungen bei höchsten Energien in Frage. Um die Rätsel der kosmischen Strahlung zu lösen, ist es unerlässlich, die Massen der einfallenden kosmischen Strahlen zu bestimmen. Die Trennung der schweren kosmischen Strahlung von der leichten kosmischen Strahlung ermöglicht Untersuchungen der Einfallsrichtungen, um minimal abgelenkte kosmische Strahlung mit potenziellen Quellen in Verbindung zu bringen. Da die Anzahl der Nukleonen direkt mit der Anzahl der erzeugten Myonen zusammenhängt, können wir dadurch außerdem die Modelle der hadronischen Wechselwirkung anhand realer Messungen überprüfen.

Aufgrund des geringen Flusses der ultrahochenergetischen kosmischen Strahlung sind große Detektoren für den indirekten Nachweis der kosmischen Strahlung erforderlich, um genügend Statistik zu erhalten. Das Pierre-Auger-Observatorium ist das größte Observatorium für kosmische Strahlung auf der Welt und deckt einen Fläche von über $3000\,\mathrm{km}^2$ ab. Der Zerfall eines hochenergetischen Teilchens in der Atmosphäre löst eine Kaskade von Sekundärteilchen aus, die als Luftschauer bezeichnet wird. Das Observatorium ist speziell für den Nachweis von solchen Luftschauern konzipiert worden. Dabei wird die Atmosphäre als Kalorimeter genutzt, um die Entwicklung der Schauer in Längsrichtung mit Fluoreszenzdetektoren zu messen. Zusätzlich wird ein Oberflächendetektor mit regelmäßig angeordneten Detektorstationen eingesetzt, um die Menge der erzeugten Sekundärteilchen zu messen. Diese Sekundärteilchen werden üblicherweise als Schauer-Fußabdruck bezeichnet.

Die zunehmende Popularität künstlicher neuronaler Netze in jüngster Zeit hat Physikern neue, einfach zu handhabende Werkzeuge gegeben, um physikalische Probleme datengesteuert anzugehen. Die Verwendung neuronaler Netze bietet die Möglichkeit, detaillierte Daten, wie z. B. Schauer-Fußabdrücke, mit physikalischen Beobachtungsgrößen, wie z. B. der Energie eines Luftschauers, in Beziehung zu setzen, ohne dass ein analytisches Modell erstellt werden muss.

Das Hauptziel dieser Arbeit ist die Untersuchung von Methoden, die auf neuronalen Netzen basieren, um aus den Daten des Oberflächendetektors des Pierre-Auger-Observatoriums Informationen zu extrahieren, die mit der Masse der kosmischen Strahlung korrelieren. Um dies zu erreichen, habe ich zwei verschiedene Ansätze in Betracht gezogen und die Nützlichkeit der Verwendung neuronaler Netze untersucht. Dabei basiert der erste Ansatz auf der Extraktion des Myonengehalts in jeder Station des Oberflächendetektors und der zweite Ansatz auf der Messung des gesamten Schauer-Fußabdrucks. Anhand von Monte-Carlo-Simulationsstudien habe ich den ersten Ansatz zugunsten des zweiten verworfen, da letzterer vielversprechendere Ergebnisse lieferte. Aus dieser Simulationsstudie habe ich drei verschiedene neuronale Netze ausgewählt, die für die Vorhersage der Tiefe des Schauer-Maximums, des relativen Myonengehalts und der logarithmischen Masse von Schauer-Fußabdrücken trainiert wurden. In einem letzten Schritt benutzte ich die Netzwerke, um die Massenzusammensetzung der ultrahochenergetischen kosmischen Strahlung aus Messungen vom Pierre Auger Observatorium zu bestimmen.

Spanish Abstract

Los rayos cósmicos de ultra alta energía son las partículas de origen natural mas energéticas conocidas por la humanidad, llegan hasta dos órdenes de magnitud por encima del límite de la actual generación de aceleradores terrestres. Cómo los rayos cósmicos obtienen su energía y en dónde se originan siguen siendo un misterio. Además, el exceso de muones en los productos de desintegración de los rayos cósmicos en la atmósfera terrestre, desafía nuestra comprensión de las interacciones hadrónicas en las más altas energías. Para resolver los misterios en torno a la física de los rayos cósmicos, es fundamental identificar su composición química. La separación entre los rayos cósmicos pesados y ligeros permite estudiar sus direcciones de arribo para asi poder relacionar a los rayos cósmicos menos desviados con sus posibles fuentes. Por otro lado, dado que el número de nucleones está directamente relacionado con el número de muones producidos, también somos capaces de someter a prueba a los modelos de interacción hadrónica de ultra alta energía a la luz de las medición.

La desintegración de un rayo cósmico en la atmósfera desencadena una lluvia de partículas secundarias llamada cascada. Debido al bajo flujo de los rayos cósmicos ultra energéticos, se requieren grandes detectores para asi obtener suficiente estadística de la detección indirecta de los primarios a través de las lluvias. El Observatorio Pierre Auger, localizado en el hemisferio sur, es el mayor observatorio de rayos cósmicos del mundo, con un área de detección efectiva de más de 3000 km^2, diseñado específicamente para detectar las cascadas de partículas inducidas. Para ello, utiliza un diseño híbrido, empleando la atmósfera como calorímetro para observar el desarrollo longitudinal de la cascada con detectores de fluorescencia, y un detector de superficie con estaciones ordenadas regularmente a nivel del suelo para medir el desarrollo lateral de las partículas secundarias producidas. Comúnmente, las partículas que llegan al suelo conforman la denominada "huella de la cascada".

El reciente aumento de la popularidad de las redes neuronales artificiales ha proporcionado nuevas herramientas y de fácil implementación para abordar los problemas de la Física. El uso de redes neuronales ofrece la oportunidad de relacionar datos detallados, como las huellas de las cascadas, con observables físicos, como la energía de una cascada, sin necesidad de construir modelos analíticos.

El objetivo central de esta tesis es la investigación de métodos basados en redes neuronales para extraer información de los datos tomados por el detector de superficie del Observatorio Pierre Auger que esté correlacionada con la masa de los rayos cósmicos. Para ello, he considerado dos enfoques diferentes y explorado la viabilidad del uso de redes neuronales. Así, el primer enfoque se basa en la extracción del contenido de muones en cada estación del detector de superficie y el segundo enfoque se basa en la medición de toda la huella de la cascada. Utilizando estudios de simulación de Monte Carlo, descarté el primer enfoque en favor del segundo, que mostró resultados más prometedores. A partir de este estudio basado en simulaciones, seleccioné tres modelos de red neuronal diferentes entrenados para predecir la máxima profundidad de la cascada, el contenido relativo de muones y la masa logarítmica a partir de las huellas de la cascada. Como paso final, apliqué las redes a datos medidos en el obsevatorio para estimar la composición química de los rayos cósmicos primarios.

Contents

Abstracts

English Abstract i
German Abstract iii
Spanish Abstract v

Main Content

1 Introduction **1**

2 High-Energy Cosmic Ray Physics **5**
2.1 Historic overview 5
2.2 Phenomenology of UHECRs 6
2.2.1 Origin and Propagation 7
2.2.2 Cosmic Ray Flux 8
2.2.3 The Shower Cascade 10
2.2.4 Longitudinal Shower Profile 13
2.2.5 Lateral Particle Distribution of Extensive Air Showers 14
2.2.6 Simulation of Air Showers 14
2.3 The Pierre Auger Observatory 15
2.3.1 The Fluorescence Detector 16
2.3.2 The Surface Detector 17
2.3.3 Environmental monitoring 18
2.3.4 Upgrade: AugerPrime 18
2.4 Science case: key results and open questions 20
2.4.1 Muon deficit 20
2.4.2 High-energy suppression of the flux and anisotropy 21
2.4.3 Neutral messengers 21
2.4.4 Mass composition 22

3 Current Reconstruction in Use **25**
3.1 Offline analysis framework 26
3.2 Standard reconstruction 26
3.2.1 Overview of the FD reconstruction 27
3.2.2 SD reconstruction using the WCD stations 29
3.2.3 SD energy calibration via *Golden Hybrid* events 34
3.2.4 Reconstruction using SSD information 35
3.3 Reconstructions of mass sensitive parameters 36
3.3.1 Delta method: analysis of rise times 36
3.3.2 Air shower Universality 37

4 Using Artificial Neural Networks for the Reconstruction of Shower Properties **39**
4.1 Conventions and notation 41
4.2 Neural networks as tool for analyzing physics data 42
4.2.1 Training procedure 43
4.2.2 Important architectural concepts 47
4.2.3 Basic building blocks in neural networks 49

4.2.4 Notes on classification . . . 54
4.3 Status of neural network based analysis in auger . . . 55
4.3.1 Extraction of the local muon signal using SD detector stations . . . 55
4.3.2 Footprint analyses using deep neural networks . . . 56
4.4 Used evaluation metrics to compare NN predictions . . . 59
4.4.1 Metrics for the comparison of predictions of NNs . . . 59
4.4.2 Methods of comparing different distributions . . . 60
4.4.3 Estimating the spread of predictions from multiple models . . . 60

5 Data sets and Data Preparation 63
5.1 Overview of air shower simulation data sets . . . 64
5.1.1 Simulations based on the *Napoli-Praha* shower library . . . 64
5.1.2 Simulations based on the *Karlsruhe* shower library . . . 69
5.2 Overview of data measured by the Pierre Auger Observatory . . . 70
5.2.1 SD data set . . . 70
5.2.2 *Golden Hybrid* data set . . . 71
5.2.3 Differences between simulations and measurements . . . 73
5.3 Preparation of input and output data . . . 73
5.3.1 Preparation procedures of scalar data and traces . . . 73
5.3.2 Encoding of SD grid into rectangular space . . . 75
5.3.3 Geometrical standardization of shower footprints . . . 75
5.3.4 On-the-fly data augmentation . . . 79
5.4 Additional features generated from base data . . . 81
5.4.1 Timing relative to the planar shower front . . . 82
5.4.2 Energy dependence and width of the shower depth distribution of different primaries . . . 82
5.4.3 Muon content of a shower . . . 83

6 Extraction of Local Shower Properties from SD Station Signals 87
6.1 Reference models for local shower analysis . . . 88
6.1.1 Inputs and outputs for the station-level analysis . . . 88
6.1.2 Reference models . . . 89
6.1.3 Performance of reference models . . . 90
6.1.4 Importance of the input features on the prediction . . . 92
6.1.5 Stability of model parameters . . . 92
6.2 Extraction of total muon signal and comparison to baseline models . . . 93
6.2.1 Quality cuts, data transformation, and split strategy . . . 95
6.2.2 Training procedure . . . 98
6.2.3 Verification of network . . . 98
6.2.4 Comparison of reference models and neural network . . . 99
6.2.5 Estimating the muon fraction . . . 111
6.3 Extraction of muon time signal . . . 118
6.3.1 Using only WCD signals . . . 118
6.3.2 Adding SSD information . . . 120
6.4 Extracting number of muons from UMD traces . . . 122

7 Extraction of Global Shower Properties from Simulated Shower Footprints 129
7.1 Effect of the variation of hyperparameters on neural network predictions . . . 130
7.1.1 Baseline architecture for event-level prediction . . . 130
7.1.2 Shower footprint standardization . . . 132
7.1.3 Effect of varying the NN architecture . . . 133

7.1.4 Hyperparameters related to the network training . . . 136
7.1.5 Hyperparameters related to the encoding of the event-level data . . . 140
7.1.6 Using additional observables as inputs . . . 143
7.2 Evaluation of neural network predictions on event-level targets . . . 146
7.2.1 Selection process for NN models . . . 147
7.2.2 Prediction of the depth of the shower maximum . . . 148
7.2.3 Predictions of the zenith angle and the shower energy . . . 152
7.2.4 Prediction of relative muon content . . . 154
7.2.5 Direct predictions of the logarithmic mass number . . . 155
7.3 Estimating uncertainty of neural network predictions . . . 161
7.3.1 Study of predictions of NN on the same CORSIKA shower . . . 161
7.3.2 Uncertainty due to the training process . . . 165
7.3.3 Other sources of uncertainties . . . 167
7.4 Effect of using *AugerPrime* simulation data . . . 171
7.4.1 Effect of the UUB . . . 171
7.4.2 Effect of SSD information on NN predictions . . . 174
7.5 Classification of Photon Events . . . 180

8 Extraction of Global Shower Properties from the SD and Golden Hybrid Data Set 185
8.1 Transition from predictions trained on simulations to measurements . . . 185
8.1.1 Direct application of NN on measurements . . . 186
8.1.2 Derivation of corrections . . . 187
8.1.3 Correction for relative muon content and logarithmic mass number . . . 194
8.2 Estimation of mass composition of UHECRs . . . 195
8.2.1 Quality selection for the SD data set . . . 195
8.2.2 NN prediction of mass sensitive targets . . . 196

9 Conclusion 207

Acronyms

Bibliography
Internal References . . . 213
Physics References . . . 214
Computer Science References . . . 220
Other References . . . 221

SUPPLEMENTARY INFORMATION AND MATERIAL

A Definitions and Derivations 223
A.1 Notes on additional statistical quantities . . . 223
A.2 Definition of commonly used fit functions . . . 224
A.3 Standardization for an arbitrary choice of unit vectors . . . 224
A.4 Derivation of SmeLu . . . 225
A.5 Second moment of depth of shower maximum from simulation data . . . 225
A.6 Definition of the baseline architecture . . . 225
A.7 Definition of zenith-energy cut . . . 225

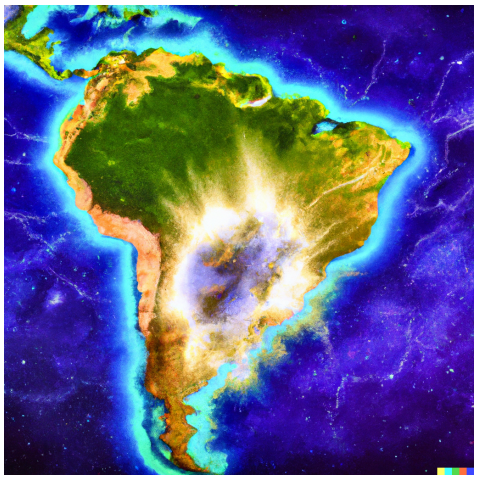
B Additional Content 229
B.1 Advanced non-standard layers . . . 229
B.2 Example for group equivariant convolutions . . . 231

B.3 Additional studies on the variation of the baseline architecture 232
B.4 Effect of model averaging 236
B.5 Estimation of noise due to input data 238
B.6 Direct energy estimator 238

C Specifications and Snippets **241**
C.1 GPU machine specifications 241
C.2 Snippets 241

D Additional Material **243**
D.1 Tabulated data 243
D.2 Additional figures 245

1 Introduction



Always go too far, because that's where you'll find the truth.

(Albert Camus, Philosopher)

DALL·E 2 prompt:

The country Argentina on Earth being hit by a high energy cosmic ray that came from another galaxy, oil painting[.]

Cosmic Rays (CRs) are particles accelerated by extraterrestrial sources. The energy spectrum of CRs is ranging from energies about 10^9 to over 10^{20} eV, spanning over 11 orders of magnitude. The exact origins and acceleration mechanisms of CRs above 10^{18} eV are still unknown. Their high energy beyond the scale of human-made accelerators challenges the theoretical models of astrophysics and particle physics. To unravel the mysteries of CRs, we require exact knowledge of the distribution of masses of the CRs arriving on Earth. Moreover, if we were be able to distinguish the CRs at highest energies on a particle-by-particle basis, we would gain new insights into non-understood problems, such as the sources of the CRs and the over-abundance of muons at highest energies. Gaining knowledge about the mass composition is the main objective of this thesis.

The flux of CRs at 10^{18} eV is roughly one CR per km^2 per year. This low flux makes direct detection unfeasible, favoring indirect methods based on the decay of CRs after interacting with the air molecules in Earth's atmosphere. Due to the high energy of the CRs, the interaction yields a cascade of secondary particles called air shower, which propagates into the momentum direction of the initial CR. Ground-based detectors can observe the development of these air showers, making it possible to gain information about the longitudinal development and lateral particle distribution at ground level.

At the time of writing, the Pierre Auger Observatory (Auger) is the world-wide largest ground-based detector for the indirect detection of CRs. The observatory follows a hybrid detector concept using fluorescence telescopes to observe the longitudinal shower profile and an extensive array of surface detector stations to measure the shower footprint. The hybrid detection of CRs makes it possible to cross-calibrate measurements taken by both independent detectors.

The Surface Detector (SD) of Auger consists of over 1600 evenly spaced Water-Cherenkov detectors (WCDs) arranged in a triangular grid. In total, Auger covers an area of about 3000 km^2. Currently, the observatory is undergoing an upgrade called *AugerPrime*. One of the additions is the deployment of Surface Scintillator Detector (SSD) on top of the WCD stations. The two independent measurements allow for a better disentanglement of the signal induced by muons and other particles.

In this work, we focus on analyzing shower footprints measured by the detector stations of the SD to extract shower observables correlated to the mass of the CR. Motivated by the phenomenology of air shower physics, the most promising candidates are the atmospheric depth of the shower maximum and the number of muons produced during the cascade. Both observables, however, are properties of an air shower that are hard to access via conventional analysis. The Fluorescence Detector (FD) allows for direct measurement of the depth of the shower maximum. Due to its sensitivity, however, the FD can only be used during moonless nights, reducing its potential uptime to about 15%. Moreover, there is still no direct measurement method for the number of muons at the highest energies. Therefore, we develop methods that relate the shower footprint measurements of the SD stations to the aforementioned quantities.

The data of air shower simulations and measurements is highly complex making the search for conventional analytic approaches exceptionally hard. For this reason, we use a data-driven approach to analyze the air shower data in this work. We select the tool set provided by Artificial Neural Networks (ANNs) for this purpose. This choice is baded on the recent successes of ANN-based approaches in hard-to-solve tasks, such as image and speech recognition. Due to similarity of footprints to images and time signals to audio signals, insights from these fields of computer science can be directly transferred to the analysis of the SD detector. Moreover, using ANN relieves us from a large part of the model building process allowing us to focus more on the mass estimation itself. Henceforth, we drop the 'artificial' from the acronym ANN solely refering to it as Neural Network (NN).

We attempt to reconstruct the observables from shower footprint data with two different approaches that coincide with two preceding analyses based on NNs. First, we link local shower information from measurements of the stations in the SD to the local signal induced by muons. While doing this, we reproduce and validate the results from the previous analysis, use reference models to contextualize the predictions of the NN, and show how much the NN-based ansatz improves the reconstruction of the local muonic content. In addition, we conduct a feasibility study to investigate whether it is possible to use an NN-based approach to extract the muon numbers from the binary signal traces of the Underground Muon Detector (UMD). Secondly, we proceed to analyze the entire shower footprint to extract global shower properties, such as the depth of the shower maximum and the shower energy. To do this, we use a NN architecture based on an earlier version used in the other preceding analysis. We boost the performance of this NN by standardizing the measured footprints to remove certain symmetries encoded in the triangular grid of the SD detector. We optimize the NN architecture and identify beneficial shower observables to improve the inference of the depth of the shower maximum, and demonstrate that NN-based approaches can be used to predict various global shower properties. In addition, we discuss various sources of uncertainty of the NN-based predictions and derive procedures to obtain uncertainty estimates from multi-network and multi-shower studies. Moreover, we conduct another feasibility study for photon event identification. In both of these analyses, we extensively use Monte Carlo (MC) air shower simulation studies. We also analyze how the SSD affects the quality of the predictions of the NNs used.

In the end, we apply the NN for the shower footprint analysis on measurements of Auger predicting the depth of the shower maximum and muon content of the showers. We correct for differences between simulations and measurements, such as atmospheric conditions, estimate systematic uncertainties for the method, and calibrate and verify the reconstruction with FD measurements. Based on this we obtain an estimate for the composition of CRs at highest energies.

Thesis philosophy This thesis is a case study. We want to ask in which way NNs may improve the prediction of specific high-level shower properties, such as the depth of the shower maximum and the primary particle energy, and how we can exploit this to gain a better understanding of the mass composition. The thesis is split into two separate analyses: the reconstruction of localized shower properties and the reconstruction of global shower properties. We only apply the latter analysis to measurements taken by Auger.

Thesis structure In Chapter 2, we survey the basic principles of CR physics and key milestones in its history. Since it is the centerpiece of this work, we discuss Auger and its most important contributions to modern science. We follow this up in Chapter 3 with the standard reconstruction of air shower events in the main framework of the Auger collaboration. Moreover, we discuss advanced reconstruction techniques. In Chapter 4, we introduce NNs as a tool for physics analysis. We also analyze the most prevalent analyses in Auger, which are based on NNs.

In Chapter 5, we discuss the datasets we simulated for this thesis and how we prepare them to use them as inputs for our NNs. In Chapter 6, we use air-shower simulations to compare NNs used to extract the muon content from single surface detector stations with reference models to contextualize the predictions of the NNs. This analysis leads to two additional studies. We try to exploit the new scintillator detectors to improve the predictions and check if an NN-based approach could work with data taken by the UMD. In Chapter 7, we switch to the NN-based reconstruction of global shower properties using all necessary event-level information. We systematically investigate the effect of selected hyperparameters of our network architecture to achieve a superior result. In addition, we discuss how we estimate uncertainties in the predictions of our networks, check the effect of introducing SSD data into our inputs, and check if an NN-based approach helps in distinguishing air showers induced by photons. Afterward, we apply this knowledge to measurement data in Chapter 8. We demonstrate how NN-based models trained on air shower simulations can be used on SD data taken by Auger. Moreover, we predict the depth of the shower maximum, the relative muon content, and the logarithmic mass number from shower footprint data with three different NNs obtaining insights into the mass composition at the highest energies. Finally, in Chapter 9, we draw a conclusion. We present the most important results and give a short outlook of future projects and possible improvements for further studies.

2 High-Energy Cosmic Ray Physics



It was just a colour out of space – a frightful messenger from unformed realms of infinity beyond all Nature as we know it;

(The Colour out of Space, H. P. Lovecraft)

DALL·E 2 prompt:

The physicist [V]ictor [H]ess wearing his glasses sitting in a hot air balloon with an electrometer explaining extensive air showers by Picasso[.]

Cosmic Rays (CRs) are particles originating from various extraterrestrial sources. Their spectrum covers more than *eleven* orders of magnitude [P:1] from below 1 GeV up to over 100 EeV. In the low-energy regions, CRs are mainly produced by coronal ejections from nearby stars, such as our Sun [P:2]. At higher energies, however, their exact origins are still a mystery and part of active research. CRs with energies above 10^{18} eV are called Ultra-High Energy Cosmic Rays (UHECRs). They are especially interesting since human-made accelerators, such as the Large Hadron Collider (LHC) [P:3], cannot accelerate particles to such high energy. They give us a way to probe particle physics beyond the energy scales of modern colliders. Like in the pre-accelerator era, Cosmic Ray Physics (CRP) provides us with further insights into the most extreme microscopic and macroscopic processes of our Universe.

At the highest energies, around 100 EeV, UHECRs have a flux of about one particle per km^2 per century (see Fig. 2.3). This low flux makes direct detection unfeasible. Hence, we need huge detector arrays to observe them in significant quantities in a reasonable time.

We start this chapter with a historical overview of CRP in Sec. 2.1. Afterward, in Sec. 2.2, we discuss the theory required to understand the basics of CRP. Then, in Sec. 2.3, we introduce the Pierre Auger Observatory. Up to date, this is the world's largest detector of UHECRs. Finally, in Sec. 2.4, we give an overview of the current science case of the observatory and CRP.

2.1 Historic overview

Over the last millennia, our ancestors gazed into the night sky, trying to comprehend its all-encompassing sea of stars. It fueled their imagination, sparking religion [T:A] and ancient philosophy [T:B] alike. Only very recently have we started to understand it in a scientific way. Every time we have looked closer, we have found something new and unforeseen. One of those unexpected finds has been CRs.

After the discovery of spontaneous radioactive decay in the 20th century by Becquerel [P:4],

measurements indicated that ionizing radiation was constantly present in Earth's atmosphere. Electrometers discharged even without obvious radioactive materials nearby [P:5]. Initially, many scientists believed Earth itself was the only source of this radiation. They drew this conclusion from electroscope measurements at different heights near the ground level, such as in [P:6]. The main idea was that radioactive elements underground were its source. Domenico Pacini contradicted this view by placing electroscopes underwater [P:7, P:8]. He found a decrease in the ionization rate. This measurement implied that the water shielded the radiation from sources other than Earth itself. However, initial measurements at higher altitudes, such as at the top of the Eiffel tower [P:9], did not support a cosmic origin. Higher altitudes were required. Using balloon experiments at various times of the day, Victor Hess [P:10] cemented this idea. Up to a certain altitude, the radiation decreased; then, it increased exponentially with height. Since the measurements did not vary too much between day and night, the Sun was ruled out as the primary source of this radiation. Hence, Hess concluded that the cosmos itself must be the source.

Finding their non-terrestrial origin, however, did not explain the nature of the radiation. Initially, gamma rays were the favored candidate [P:11], which gave rise to the name *cosmic ray* we use today. This changed after the dependence of the CR flux on the magnetic field of the Earth, which had been predicted by Rossi [P:12], was discovered. Later, Rossi concluded that the CRs produced secondary particles which are able to traverse the atmosphere [P:13] without too much energy loss. The discovery [P:14] of the muon μ revealed the nature of these secondary particles.

In 1939, Auger and colleagues arranged Geiger counters 300 m apart in a flat plane [P:15] outdoors. Finding a higher rate of coinciding triggers than expected, they concluded that the CRs interacted with air molecules in the atmosphere creating cascades of secondary particles. Moreover, they found that CRs followed a steeply decreasing energy spectrum of the form E^{-2}. In 1954, Heitler developed a theoretical framework that explained cascades induced by high-energy photons [P:16] (see Sec. 2.2.5). This led to the construction of large CR detector arrays, such as the Vulcano Ranch experiment in New Mexico. During the experiment, Linsley and colleagues observed the first UHECR with an energy of more than 10^{20} eV [P:17]. In 1966, Greisen, Zatsepin, and Kuzmin predicted a theoretical upper limit of the energies of proton UHECRs [P:18, P:19]: the Greisen–Zatsepin–Kuzmin (GZK) cutoff. Due to interaction with photons of the Cosmic Microwave Background (CMB), protons would lose energy during their travel to Earth, reducing the maximum distance to their potential sources significantly and softening the particle spectrum.

All these findings motivated the construction of more giant detectors. The next-generation experiments yielded contradictory results. While High Resolution Fly's eye (HiRes) – a fluorescence telescope – reported a steepening of the spectrum at very high energies [P:20], Akeno Giant Air Shower Array (AGASA) [P:21] – a ground-based detector – did not observe a suppression at the highest energies. To resolve this tension, an even larger observatory was needed, which led to the construction of the Pierre Auger Observatory (Auger) (see Sec. 2.3). Up to date, Auger is still the largest CR detector in the world. After ten years of operation, it verified the existence of a suppression of the spectrum at the highest energies. It is, however, still not clear if this suppression is partially due to the GZK cutoff or primarily due to the scarcity of sources.

2.2 Phenomenology of UHECRs

In this work, we define UHECRs as elementary particles and nuclei having a primary particle energy equal to or above 1 EeV. Due to their extremely small flux, direct detection is infeasible. However, when UHECRs enter Earth's atmosphere, they interact with air molecules.

This results in a shower cascade of billions of secondary low-energy particles. A part of these excites the nitrogen in the atmosphere leaving a longitudinal profile of fluorescence light. In addition, the footprint of the shower cascade can be measured at ground level.

The first couple of interactions of the primary particle are primarily of hadronic nature. Hence, they are governed by the theory of strong interaction: Quantum Chromodynamics (QCD). Due to the precise measurements of today's accelerators, we understand high-energy hadronic interactions quite well [P:22]. However, at low energies, soft QCD still poses many problems due to the breakdown of perturbative approaches. Since soft QCD is required to describe the development of hadronic air showers accurately, this leads to a mismatch between air shower measurements and simulations in the number of hadronic particles produced in a shower. To resolve this mismatch with particle accelerators, they would need to be of an astronomical size[1] to reach the required center-of-mass energies of up to 10^{20} eV. Therefore, the current procedures rely on the extrapolation of LHC data to approximate the properties of the interaction of CRs with the atmosphere.

2.2.1 Origin and Propagation

The exact processes that could accelerate UHECRs to such high energies are still a mystery. Until now, no experiment could uncover UHECR point sources.

Still, we are able to quantify some properties of the required sources. For charged CRs, the Lamor radius R_L needs to be smaller than the scale of the magnetic turbulences in the source R_s to confine them. This yields

$$R_s > \frac{E}{\beta Z e B} = R_\mathrm{L}, \tag{2.1}$$

where B is the magnetic field strength of the source, Z is the charge number of the particle, e is the unit charge, and $\beta = v/c_0$ is the characteristic velocity of the scattering centers. We call Eq. (2.1) the Hillas-Criterion for sources [P:2, P:23]. Note that this condition does not guarantee that the energy is achievable. It only is a rough estimate. The maximum energy achievable still depends on the accelerator itself.

Fig. 2.1(a) shows the Hillas diagram. It combines the characteristic size and magnetic field strength of different source classes. The lines represent the minimum product $R_s B$ required for the acceleration to 10^{20} eV for different values of β. From this plot, we already see that some sources, such as normal galaxies and supernovae, are not able to satisfy Eq. (2.1) and are therefore ruled out as the origins of UHECRs. However, even if a source passes the Hillas-Criterion, it still requires the necessary energy budget to produce enough UHECRs to explain the observed flux. In Fig. 2.1(b), we show the number density for various sources using their characteristic luminosities. The solid line represents the luminosity required to satisfy the energy budget if the CR luminosity equals that of the source. Since the UHECR luminosity does not need to be the same as the source luminosity, the dashed lines show the deviation for an over- and underproduction. Both Fig. 2.1(a) and Fig. 2.1(b) convey that the jets of Active Galactic Nuclei (AGN) sources, such as nearby radio galaxies, are very promising candidates to act as accelerators for UHECRs [P:24].

Since hadronic CR possess a charge, they change their trajectory through the extragalactic and galactic magnetic fields. During their long propagation, even small magnetic fields cause deflections that add up significantly. Since heavier nuclei carry more charge than light ones, they are deflected more. Hence, light CRs, such as protons, are very interesting in searching for sources. By modeling the magnetic fields, the CRs traverse, they might be used as pointers toward UHECR sources.

[1] For an equivalent center of mass energy of a primary of 10^{20} eV interacting with a fixed target, we would more or less need the circumference of mercury with LHC technology.

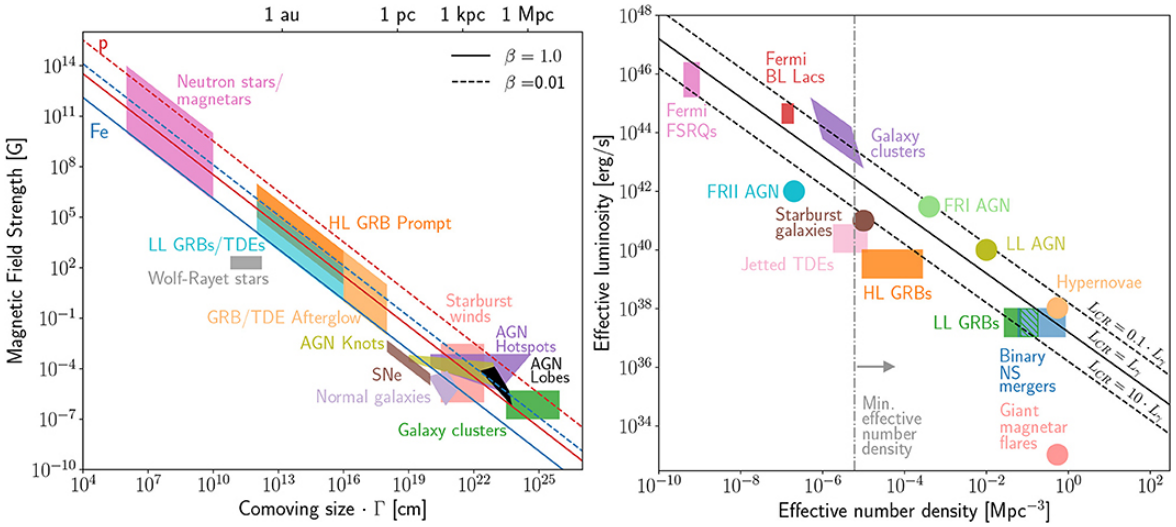


Figure 2.1: *Left*: Sizes and magnetic field strengths of various CR source candidates. The solid and dashed lines represent the confinement given by the Hillas criterion (see Eq. (2.1)) beyond the acceleration of particles to an energy of 10^{20} eV is possible. The magnetic field strength is given in the comoving frame of the source, and Γ is the Lorentz factor to account for cosmic expansion. *Right*: Effective number density and luminosity for various CR source candidates. The sources that satisfy the confinement conditions also must be luminous enough to explain the observed number densities. Taken from [P:24].

2.2.2 Cosmic Ray Flux

Assuming that the sources of cosmic rays are uniformly distributed over the sky, we are able to calculate the raw flux $J_{\text{raw}}(E)$ of CRs by "counting" cosmic ray events and correcting with the exposure of the used detector. We have depicted the flux in Fig. 2.2. Since detectors have a finite energy resolution and the flux is monotonically decreasing, energy measurements are more likely to pile up in higher energy bins. This migration effect distorts the spectrum. Therefore, $J_{\text{raw}}(E)$ is solely a prior for the real flux $J(E)$. To obtain the $J(E)$ from $J_{\text{raw}}(E)$ we need to do a complex unfolding process that corrects this migration effect [P:25, P:26].

The spectrum of UHECRs covers a vast energy range of more than two orders of magnitude. We describe different regions via local power laws of the form

$$J(E) = \frac{\mathrm{d}N}{\mathrm{d}E} \propto_{\text{loc}} E^{-\gamma_{\text{loc}}}, \tag{2.2}$$

where "loc" stands for local, N is the number of observed particles, and γ is the spectral index of the leading contribution inside the local energy interval. Between these regions, we find the so-called features of the cosmic ray flux. These features are sudden changes of the flux transitioning into a different dominant power law. The position of these features and the switch from one power law to another indicates changes in the main contributors to the flux and, hence, the underlying physical processes.

In this section, we assume that the flux is isotropic, even though this is not strictly true for all regions of the energy individually [P:27, P:28]. We discuss this in Sec. 2.4.2.

Currently, there is a consensus that the spectrum above an energy of 10^{17} eV contains three notable features: the second knee[2], the ankle, and the suppression at the highest energies. The exact origin of these high-energy features is still subject to active research and not completely understood. Here, we discuss the most common explanations.

[2]There is another knee at around $3{\times}10^{15}$ eV that is due to the decrease of light elements [P:30].

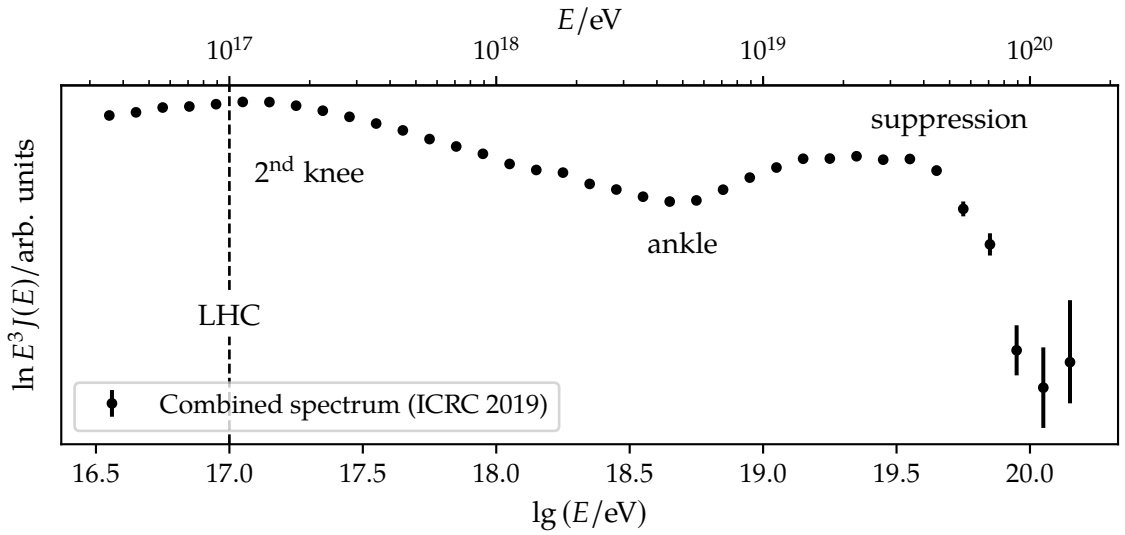


Figure 2.2: Scaled all-particle cosmic ray flux measured by the Pierre Auger Observatory. The flux is scaled by E^3 to highlight notable features. Most of the spectrum is well above the equivalent collision energy of the LHC. There are three notable features: the second knee, the ankle, and the suppression at the highest energies which are all reviewed in Sec. 2.2.2. Reproduced from data taken from [A:1]; official publication [P:29].

The second knee is a sudden steepening of the all-particle spectrum [P:29] at an energy of about $1.5{\times}10^{17}$ eV. Similarly to the first knee, this is (most likely) due to a decrease in population in the cosmic ray flux from our galaxy. In this case, that of the heavier elements. At this energy, the local galactic acceleration mechanisms are most likely not "extreme enough" to fuel the spectrum anymore [P:31].

The ankle is a sudden hardening of the CR spectrum. At this energy, the cosmic rays approach the threshold being able to escape their galactic boundaries. Hence, it most likely marks the transition from the dominance of galactic to extra-galactic cosmic rays and, in addition, a transition from light to heavier elements [P:32]. There are many models trying to explain this. Notable explanations are provided by the dip model [P:33] and the mixed composition model [P:34]. However, it is not entirely clear if one of them is the correct one.

The suppression is a strong decrease in flux at energies above $3{\times}10^{19}$ eV. The nature of the suppression is still a subject of heavy debate. A straight-forward way of giving an explanation is the GZK effect [P:18, P:19]. At about $5{\times}10^{19}$ eV, protons start to interact with the photons of the CMB. The CMB is a remnant of the era of recombination [P:35, P:36]; the point in time when the Universe had expanded enough to allow for the formation of neutral hydrogen. The CMB is a photon gas exhibiting a black body spectrum of a temperature of about 2.7 K [P:37]. In the rest system of the high-energy protons, this gas is heavily blueshifted. This allows photons with energies higher than (about) $1.2{\times}10^{9}$ eV to excite the nucleon. This yields π^0 and π^+ via the Δ resonance

$$p + \gamma_{\text{CMB}} \rightarrow \Delta^+ \rightarrow \left\{ \begin{array}{l} p + \pi^0 \\ n + \pi^+ \end{array} \right\} \rightarrow p + \text{secondary particles}, \tag{2.3}$$

reducing the energy of the proton during its travel to Earth. Although this is a convenient explanation, it is also possible that the steep decrease at the end of the spectrum indicates the

limit of acceleration of UHECR sources. At this point, there could just not be enough accelerators in our Universe that are capable of accelerating particles to these energies. To uncover which of these explanations is correct, it is necessary to reconstruct the mass composition at the highest energies from the high-energy data.

The resonance described by the GZK covers only protons. To achieve a similar nucleon resonance in heavier nuclei, the necessary energy for a Δ resonance would increase if we assume that the nucleus is just an ensemble of nucleons of equal energy. However, there are other effects, such as the giant resonance [P:38], that yield a similar effect than the GZK for protons.

2.2.3 The Shower Cascade

When high-energy[3] cosmic rays enter the atmosphere, they interact with air molecules inducing a cascade of millions of secondary particles. The atmosphere acts as an inhomogeneous calorimeter. The probability of interaction is proportional to the air density and increases towards the ground. A convenient way to describe traversed matter is given by the integrated slant column density

$$X(h) = \frac{1}{\cos\theta} \int_h^\infty \mathrm{d}h' \, \rho(h'), \tag{2.4}$$

where $\rho(h)$ is the local air density at height h above sea level and θ is the zenith angle[4] of the air shower event. The secans factor accounts for the longer path it takes through the atmosphere if the event is not perfectly vertical. X is referred to as slant depth. Using X as the variable of progression, the shower cascade follows approximately the differential equation [P:2]

$$\frac{\mathrm{d}N_i}{\mathrm{d}X} = -\left(\frac{1}{d_i} + \frac{1}{\lambda_i}\right) N_i(E_i, X) + \sum_j \int_{E_i}^\infty \mathrm{d}E_j \, N_j(E_j, X) Y_{j\to i}(E_i, E_j), \tag{2.5}$$

where λ_i is the interaction length of the particle i in air, d_i is its probability of decay in the infinitesimal element $\mathrm{d}X$, $N_j \mathrm{d}E_j$ is the flux of particles of type i, and $Y_{j\to i}$ is the yield of a particle of type i from particles of type j.

Due to momentum conservation, the shower cascade propagates into the direction given by the momentum vector of the initial cosmic ray. We denote the straight line defined by the first point of interaction and the momentum vector as the shower core.

We put the sub-components of the shower cascade initiated by a hadron primary in four different classes (see Fig. 2.3). First, we have the hadronic shower component. It consists primarily of pions but also other mesons. These decay relatively fast and, therefore, are mostly found near the shower core and during the early stages of the shower. Secondly, we have the muonic component. It is comprised of muons and anti-muons that are generated by the decay of charged pions and kaons. Due to their long mean lifetime (~2 µs) in the atmosphere, most muons reach ground level without interacting. They are related to the hadronic interactions in the shower (see Sec. 2.2.3.B). Therefore, having accurate measurements gives us insights into the mass of the primary particle. Thirdly, we have the electromagnetic component. It consists of the photons, electrons, and positrons generated by the secondary mesons and muon decays. It also inherits around 90% of the initial energy of the primary. The atmosphere quickly absorbs this component. One of the processes involved in this is the excitation

[3] We define high-energy CRs as CRs which exhibit an energy above 1 PeV coinciding with the first knee.
[4] Angle between the CR momentum vector and Earth's normal surface vector.

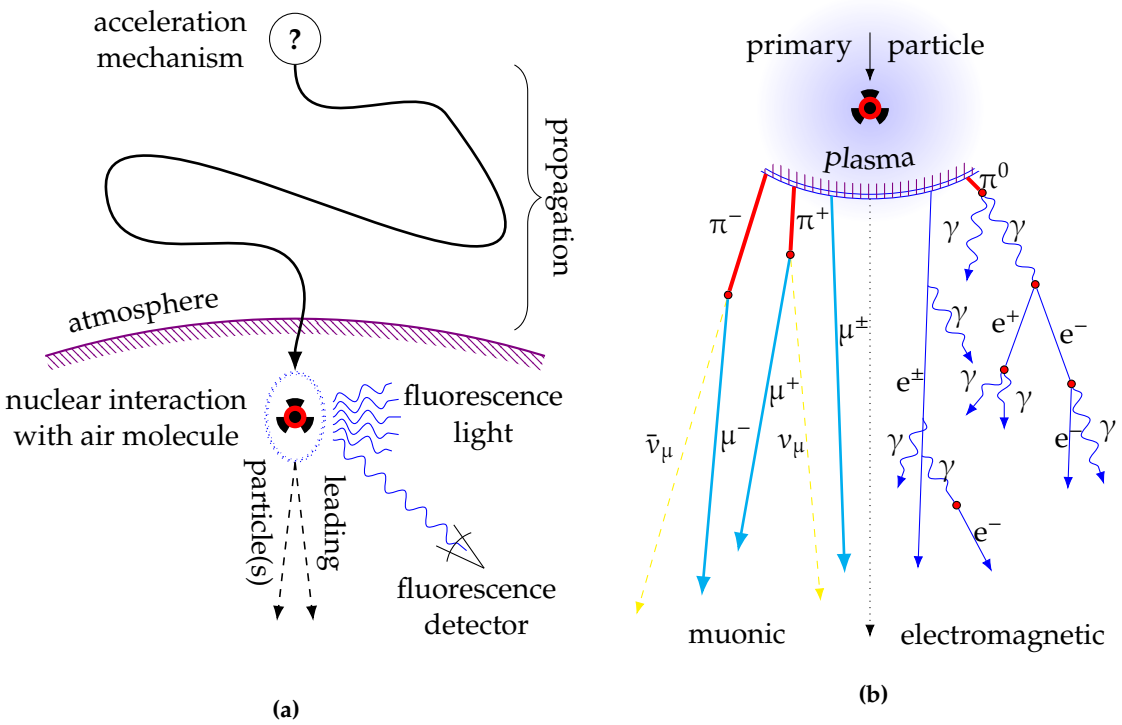


Figure 2.3: *Left*: Schematic representation of the path CRs take from their source to Earth's atmosphere. *Right*: Important decay processes in the hadronic shower cascade. Both the sources of UHECRs and their exact propagation processes are still a mystery. When they enter the atmosphere, the atmosphere acts as an inhomogeneous calorimeter. The electromagnetic subcomponent of the shower interacts with the nitrogen molecules of the atmosphere, producing fluorescence light in the UV region, which can be measured by specialized fluorescence detectors. This sub-component is driven by the π^0 produced in the shower cascade.

of nitrogen molecules (see Fig. 2.4). The de-excitation of nitrogen yields isotropic fluorescence light with a wavelength in the ultraviolet (UV) region, which is detectable by fluorescence telescopes. Models like shower universality (see Sec. 3.3.2) even differentiate between the electromagnetic component produced from muon decays and that of other sources. Finally, there is an "invisible" component[5]. This sub-component is comprised of particles, such as neutrinos, which are not detectable by most of the employed detector setups. We only account for the first three components and assume the losses due to the last component are comparably small.

A Heitler model

The Heitler model is an approximate solution of Eq. (2.5) for purely electromagnetic showers [P:16] induced by photons. The model provides us with a view of the electromagnetic sub-component discussed in the last subsection. The main idea is that only two dominant effects govern the development of the electromagnetic shower cascade. Photons produce electron-positron pairs via pair production, and electrons and positrons, on the other hand, produce photons via bremsstrahlung (see Fig. 2.4). Therefore, after each interaction, the number of particles doubles. Using the electromagnetic interaction length λ, the number of particles at X is

$$N(X) = 2^{X/\lambda}. \tag{2.6}$$

[5]In fact, the component is only very hard to detect.

At a critical energy of $E_c \approx 87\,\mathrm{MeV}$, ionization losses take over the radiation losses. Particles in the shower cascade below this energy do not add additional particles to the cascade. Hence, the maximum number of particles is determined by

$$N_{\max} = \frac{E}{E_c}, \tag{2.7}$$

where E is the initial energy of the CR. Using Eq. (2.6) and Eq. (2.7), we can find out at which shower depth the shower starts to "die out". This depth of the shower maximum is

$$X_{\max} = \lambda \log_2 \frac{E}{E_c}. \tag{2.8}$$

Essentially, for electromagnetic showers, the number of particles and depth of the shower maximum scale with energy and logarithmic energy, respectively. Usually, in Eq. (2.8), the natural logarithm is used instead of the binary logarithm absorbing the conversion constant into λ. We only keep it for the sake of clarity.

B Heitler-Matthews model

The Heitler-Matthews model is an extension of the Heitler model to approximate showers originating from primary nuclei [P:39]. It assumes that only the pions play a role for the development of the shower cascade. Since there are three kinds pions – two of which are charged – the decay ratio is 2:1 (charged:neutral). All of them are unstable. The charged pions decay approximately after the hadronic interaction length λ_I. Neutral pions decay very fast with a very high probability into two photons. Hence, they transfer energy into the electromagnetic cascade. In contrast to this, charged pions interact again building up the hadronic part of the shower. Afterward, when they reach a critical energy threshold, they decay into a muon and a neutrino. The former are the origin of the muonic sub-component.

We assume that each interaction produces N_π pions. After n interactions the average energy of the pions is $E_\pi = E/N_\pi^n$. In the case that the distance to the next interaction point exceeds the decay length of a pion the production of further pions is suppressed. The decay length depends on the energy of the pion. In the framework of the Heitler-Matthews model, this critical energy is fixed to $E_{hc} = 20\,\mathrm{GeV}$ [P:39], despite being dependent on the primary particle energy. The maximum number of interactions is

$$n_{\max} = \left\lfloor \frac{\ln(E/E_{hc})}{\ln N_\pi} \right\rfloor, \tag{2.9}$$

where $\lfloor \cdot \rfloor$ is the floor operator. Since only charged pion decays generate muons, in this model

$$N_\mu = \left(\frac{2}{3}N_\pi\right)^{n_{\max}} = \left(\frac{E}{E_{hc}}\right)^{\frac{\ln \frac{2}{3}N_\pi}{\ln N_\pi}}, \tag{2.10}$$

are produced. The number of produced muons is a power law of the initial energy with an exponent lower than unity. Moreover, studying the neutral pions, we are able to obtain an estimate on the shower maximum. Since $N_\pi/3$ neutral pions are produced at each interaction, and each neutral pion decays into two photons, the energy for the photons is $E/(2N_\pi)$. Assuming that the proton first interacted at X_0 we can modify Eq. (2.7) accordingly:

$$X_{\max} = X_0 + \lambda \log_2 \frac{E}{2N_\pi E_c}. \tag{2.11}$$

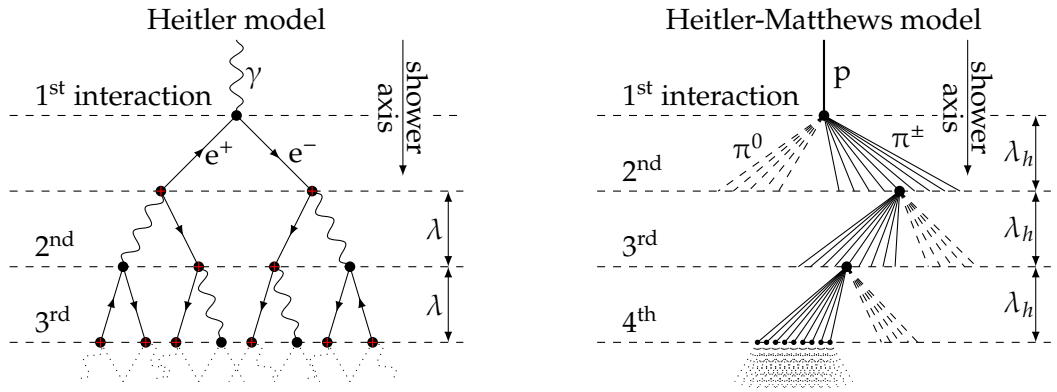


Figure 2.4: Illustrations of shower development according to the Heitler model (*left*) and the Heitler-Matthews model (*right*). The Heitler model considers a purely electromagnetic cascade spawned by a very high-energy photon. In the atmosphere, the photon decays into an electron-positron pair via pair production. The electron and positron then produce photons via bremsstrahlung. This process yields an exponential increase of particles in the cascade until a critical threshold energy is reached. In the Heitler-Matthews model, a proton decays in the atmosphere into a number of pions due to hadronic interactions. In turn, the charged pions decay into many lower energy pions. Neutral pions are the drivers of the electromagnetic component of the hadronic shower decaying with a branching fraction of almost 99% into two photons [P:22].

Again, at this shower depth, the shower begins to die out because the ionization losses take over.

To account for heavier primary nuclei, we are able to generalize this model by using the superposition model since the initial energy is much larger than that of the binding energy of the primary particle. We treat the shower cascade as A simultaneously induced proton shower cascades and ignore the inter-nuclei interactions. Using the substitutions $E \longrightarrow E/A$, we generalize Eq. (2.11) to

$$X_{\max}(E, A) = X_0 + \lambda \log_2 \frac{E}{2N_\pi E_c A} \stackrel{!}{=} X_{\max}(E, 1) - \lambda \log_2 A \tag{2.12}$$

and Eq. (2.10) to

$$N_\mu(E, A) = A N_\mu(E/A, 1) = A^{1-\frac{\ln \frac{2}{3} N_\pi}{\ln N_\pi}} N_\mu(E, 1). \tag{2.13}$$

We expect that a shower initiated by heavier nuclei for the same primary energy produces more muons and has a shallower $X_{\max}$. Although the Heitler-Matthews model gives only a qualitative explanation of the hadronic shower cascade, it describes the main effects quite well. The results do not change significantly, even if we account for the fragmentation of heavier nuclei [P:40].

2.2.4 Longitudinal Shower Profile

The electrons and positrons in the shower cascade experience constant energy losses by ionizing nitrogen molecules. During de-excitation, the molecules emit UV light isotropically. Consequentially, the shower cascade has a longitudinal fluorescence light profile that gives us the possibility to estimate the depth of the shower maximum and even the energy of the shower. Using X as from Eq. (2.4), the shower profile approximately follows the Gaisser-Hillas (GH) function [P:41]. It is given by

$$\frac{\mathrm{d}E}{\mathrm{d}X} \propto N(X) = N_{\max} \left[\frac{X - X_1}{X_{\max} - X_1} \right]^{\frac{X_{\max} - X_1}{\Lambda}} \exp\left[\frac{X_{\max} - X}{\Lambda} \right], \tag{2.14}$$

where Λ is treated as a pure fit parameter for hadronic showers, X_1 is the depth the of first interaction, and $N_{\max}$ is the number of contributing particles at the depth of the shower maximum. Eq. (2.14) describes a purely electromagnetic shower very well. For an electromagnetic air shower, Λ is the interaction length. Using Eq. (2.14), the extraction of $X_{\max}$ is straightforward. We provide an overview of the procedure in Sec. 3.2.1.

After determining the free parameters in Eq. (2.14) by measurements, we obtain the calorimetric energy E_{cal} of the shower by integration. This is not the energy of the primary particle since muons and other non-ionizing components are not accounted for. Hence, converting E_{cal} to the actual energy of the primary particle requires a model for this invisible part of the shower cascade. Consequentially, a fluorescence measurement depends indirectly on theoretical models for the calibration process.

It is noteworthy that the longitudinal shower profiles follow a universal behavior [P:42, P:43]. By changing coordinates from X to the slant depth $\Delta X = X - X_{\max}$ in Eq. (2.14), we can approximately align the profiles of different primaries. For protons[6], the shower-to-shower fluctuations are very large due to the randomness of X_1. However, the coordinate change removes this dependency. The framework of Universality uses this as a starting point for a "pattern-matching process" (see Sec. 3.3.2).

2.2.5 Lateral Particle Distribution of Extensive Air Showers

Another way to indirectly detect UHECRs is by measuring the particle content in one slice of the shower cascade. Most commonly, this slice is at ground level. In this case, we call this the shower footprint and the reason Extensive Air Showers (EAS) have their name. For electromagnetic showers, the Nishimura-Kamata-Greisen (NKG) function describes this footprint very well [P:44, P:45]. It is given by

$$\rho_{\text{NKG}} \propto \left(\frac{r}{r_m}\right)^{s-2} \left(1 + \frac{r}{r_m}\right)^{s-4.5}, \tag{2.15}$$

where r is the distance to the shower axis, r_m is the Molière unit[7], and s is the shower age which is defined as

$$s = \frac{3X}{X + 2X_{\max}}. \tag{2.16}$$

We define r as the shortest distance between the straight line defined by the shower axis and a point on ground level. Hence, it is the distance between a detector and the shower axis in the shower-detector plane.

To describe hadronic shower footprints more accurately, we use a modified version of Eq. (2.15). The modified NKG function reads

$$\rho_{\text{mNKG}} \propto \left(\frac{r}{r_{\text{opt}}}\right)^{\beta} \left(\frac{r + r_{\text{s}}}{r_{\text{opt}} + r_{\text{s}}}\right)^{\beta+\gamma}, \tag{2.17}$$

where r_{opt} and r_{s} are parameters that depend on the used detector, and r is, again, the distance to the shower axis. It is a purely phenomenological extension that is convenient to use for hadronic showers.

2.2.6 Simulation of Air Showers

If we want to test our analysis methods or detector setups, we need simulations of the physical processes we are going to encounter. This is especially important since air showers are

[6] For more massive nuclei the fluctuations are reduced due to the superposition.
[7] A constant connected to spread due to Coloumb scattering.

stochastic processes. It would be incredibly hard to develop a purely analytical model that describes shower-to-shower fluctuations. To simulate EAS, we rely on the COsmic Ray SImulations for KAscade (CORSIKA) [T:C] software. The most commonly used version, CORSIKA 7, is a single-core program written in `Fortran`. It computes the propagation of air shower cascades through the atmosphere in a step-by-step fashion. This involves following all individual particles that are above a certain energy threshold and tracking all of their interactions and decays. Without this threshold, a proton shower at the highest energies would take years to simulate [P:46]. Still, due to the sheer number of particles, CORSIKA simulations are computationally expensive. The higher the energy, the higher the number of particles that need to be followed. This, in turn, means that the execution time of CORSIKA scales with the energy of the initial particle (see Sec. 2.2.3.B).

Moreover, to simulate hadronic air showers, we need an accurate model of the hadronic interactions at the target energies. Currently, there are three different models used: EPOS-LHC [P:47], QGSJetII-04 [P:48], and Sibyll2.3 [P:49]. They all follow slightly different design philosophies producing roughly similar results. We only use the first two in this thesis in Chapters 6 to 7. Energy-dependent quantities, such as cross sections, are far out of the reach of current human-made accelerators, such as the LHC (see Fig. 2.2). Hence, these high-energy hadronic models have to rely on extrapolations motivated by collider physics.

Although all of the models follow different core principles, they all share an unfortunate similarity. They predict too few muons on ground level if compared to recent measurements [P:50]. This makes them all inevitably part of active research. Still, the simulations enable us to do essential cross-checks and, after simulating the detector responses (see Sec. 3.1), to test our models.

CORSIKA produces two important output files that correspond to the longitudinal shower development and the particle distribution at the ground level. Both are important for the Fluorescence Detector (FD) and Surface Detector (SD) simulation, respectively. We denote the former as `.long` files and the latter as `.part` files.

2.3 The Pierre Auger Observatory

The flux of UHECRs is extremely low. Hence, we need a huge effective detection area to detect a statistically significant amount of them making direct detection methods unfeasible. However, we are able to exploit the properties of the shower cascade (see Sec. 2.2.4 and Sec. 2.2.5) to detect UHECRs indirectly favoring large-scale ground-based detector arrays, which use the atmosphere as a calorimeter.

Up-to-date Auger is the largest CR detector, specifically designed to measure UHECRs and their decay products at energies above 10^{18} eV [P:51, P:52]. It covers an area of roughly 3000 km^2 in the Argentinean Pampa Amarilla (see Fig. 2.5) [A:1]. Being far away from any industrial center, this very flat part of South America provides a stable, clear air condition at a high altitude of about 1400 m above sea level. The data-taking process started in 2004 during the construction and is still ongoing. Auger follows a hybrid detector concept: In addition to solely measuring the longitudinal profile of the ionized atmosphere with the FD, like in experiments like HiRes [P:20], there is a complementary SD. The SD consists of different grids of Water-Cherenkov detectors (WCDs). They allow for point-like measurements of the shower footprint. The benefits of this design are two-fold: First, it allows for a cross-calibration of the two detector setups. Secondly, we obtain a reliable way to increase the uptime of the detector since the FD alone only operates during clear moonless nights, which occur only about 15% of the time [P:53]. In Fig. 2.5, an illustration of the observatory is provided.

This hybrid detector concept is easily extendable. By adding more detector systems that

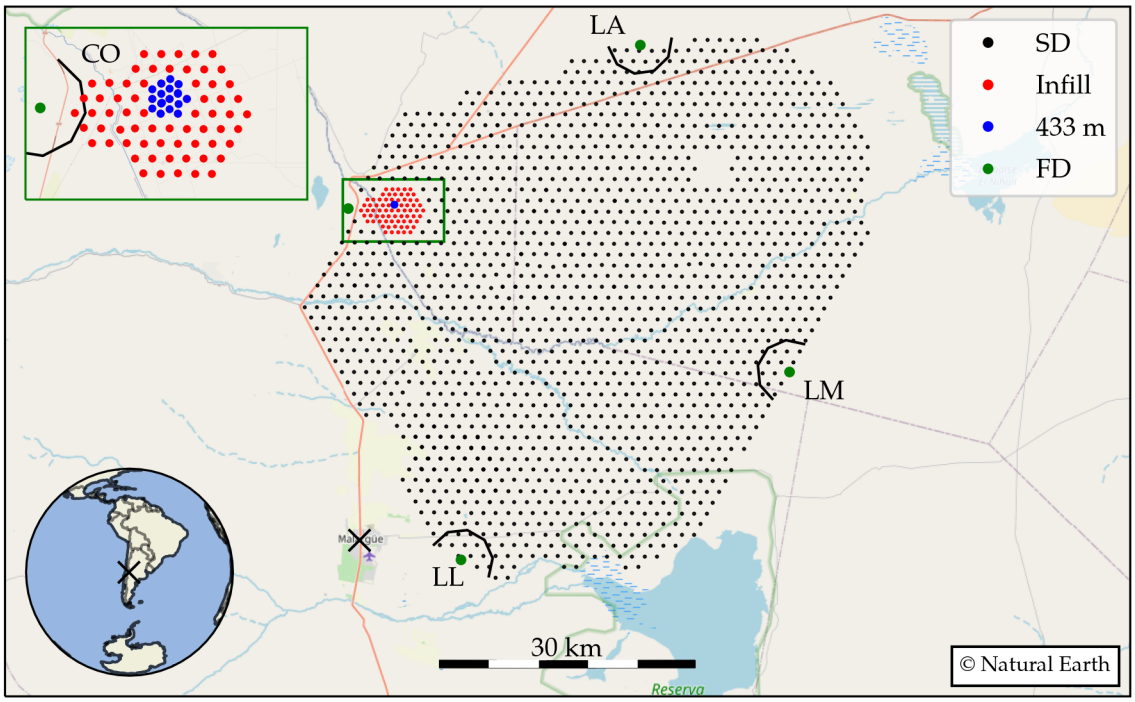


Figure 2.5: Overview of the FD and SD of Auger. The black segmented lines around the FD stations represent the azimuth viewing angles of the 24 florescence telescopes. The green box provides a zoomed in version of the Infill and 433 m array. For visibility we removed the regular SD detectors. The cross marks the position of the central data acquisition center in Malargüe. The background map is from [T:D].

measure and focus on different subsets of the shower, we are more likely to understand its particle content fully. Hence, at the moment of writing, the observatory has been going through an upgrade process. The upgraded observatory, denoted as AugerPrime (AP), adds, among other things, various new detectors to the SD and uses upgraded electronics. We describe AP in Sec. 2.3.4.

2.3.1 The Fluorescence Detector

The FD system of Auger consists of four detector buildings housing six independent fluorescence telescopes each [P:52]. All four detector sites face toward the center of the surface detector array, covering a field of view of 180° in azimuth and roughly 30° in altitude. The name of the sites is Coiheco (CO), Los Leones (LL), Loma Amarilla (LA), and Los Morados (LM). Their positions position relative to the SD array is depicted in Fig. 2.5. Due to their very high sensitivity, it is only possible to use them during moonless nights. Otherwise, their electronics would take significant amounts of damage because of the high amplification used. Hence, they follow a duty cycle of up to 15%.

In Fig. 2.6, an illustration of one of the 24 fluorescence telescopes is depicted. The fluorescence light emitted by the shower cascade enters the telescope through a circular UV-transmitting filter shielding the detector from background light. A segmented mirror redirects the light on a camera in the focus point. Depending on the site, the mirror consists of either rectangular or hexagonal pieces. The camera consists of a 22×20 grid of 440 hexagonal Photomultiplier Tubes (PMTs). Each of these PMTs represents one pixel in the final digitized image read out at a rate of about 12 mHz [P:52, P:53].

Due to its direct detection principle, the FD sets the energy scale for the entire experiment.

Consequentially, it is highly important to calibrate the FD telescopes regularly to ensure valid and consistent measurements. To do this, we need an absolute and a relative calibration. The absolute one is a time-consuming process. For each FD telescope, a well-understood light source has to emit light directly onto the mirrors. Previously, a giant cylinder called drum [P:54] had to be mounted to the aperture. Due to the extreme amount of effort [A:2], a new calibration system will be used for AP. It is called the XY-scanner [P:55], which is a small spherical, Lambertian light source that can be moved over the entire detector area. For the relative calibration, a special LED installed in each FD site is used. The relative calibration keeps track of changes between the absolute ones [P:56].

2.3.2 The Surface Detector

The surface detector array consists of over 1600 WCDs that are arranged in a triangular grid with a spacing of 1500 m [P:57]. Due to its design, the main array achieves full efficiency at about $10^{18.5}$ eV. At an energy of 10^{19} eV, on average, at least five stations trigger for each air shower event. In addition, there are two smaller sub-arrays called the 750 m- and the 433 m-array, which have a spacing of 750 m and 433 m, respectively. They target lower energy ranges than the main array and are of no importance for this thesis.

The WCDs employ a cylindrical body containing 12,000 l of ultra-pure water with a height of 1.2 m and a diameter of 3.5 m. Inside the water, each station has three PMTs mounted symmetrically at the top of the tank (see Fig. 2.6). They are evenly spaced out. Each one has a distance of 1.2 m from the center of the detector. In this way, the measured signal of all PMTs reduces the azimuthal dependence of the Cherenkov light emitted from the charged relativistic particles. The inside of the tanks is coated with a liner to reflect all of the Cherenkov light isotropically. The electronics of the WCD stations – called Unified Board (UB) – digitizes the PMT responses with a sampling rate of 25 ns into 10 bits. There are two signals for each PMT: the high-gain and the low-gain signal. The former is gained from the last dynode, and the latter from the anode directly. The time signals recorded for an air shower event are 19.2 µs long which corresponds to 768 bins.

The onboard electronics calibrate the signals by using the signal deposited by background muons [P:58]. Using a measurement over a 61 s time window the PMT responses fill a charge histogram. The charge histogram has two peaks. The second[8] one is related to the response of a vertically centered muon traversing through the tank after a geometrical correction. Conveniently, we call it the Vertical Equivalent Muon (VEM) charge Q_{VEM}. Total signals from the tanks use VEM as the unit since the charge measurement is related to the measured particle density. Henceforth, we will use VEM as a (regular) unit. For the trigger process, there is another kind of calibration. By putting the pulse heights of the measurement into another histogram, the most likely pulse height can be obtained. We denote it as I_{VEM}. This again relates to the pulse height a vertical muon would produce. This allows finding consistent threshold levels for slightly different PMTs. A commonly used quantity is the area over peak value a_{p} defined via

$$a_{\mathrm{p}} = \frac{Q_{\mathrm{VEM}}}{I_{\mathrm{VEM}}}. \tag{2.18}$$

In measurements, this quantity varies for each PMT and depends on the detector age [A:3].

Saturation of the electronics due to many particles traversing the WCDs is a common problem occurring, especially near the shower core at ground level. If signals are below 600 VEM, the low-gain channel of the PMTs is still usable. However, above this threshold, signals have to be recovered by methods such as those presented in [P:59].

[8] The first peak comes from other electromagnetic particles.

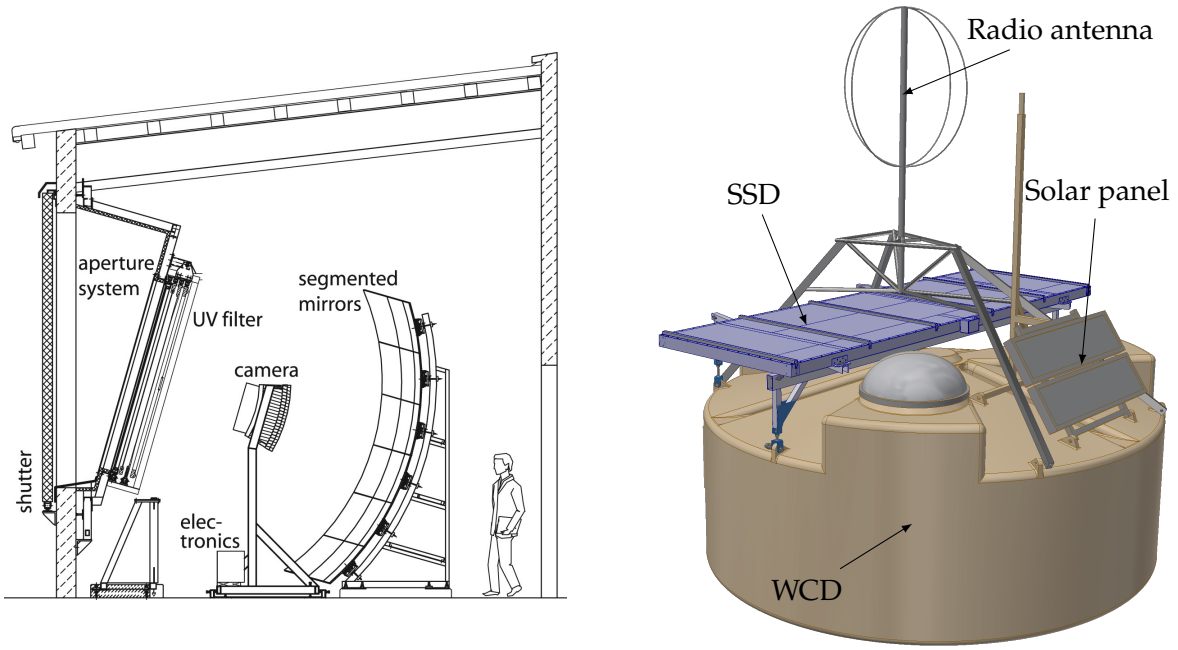


Figure 2.6: *Left*: Illustration of one of the 24 FD telescopes (taken from [P:52]). *Right*: Rendering of upgraded WCD station (modified from [P:60]). The original tank was missing the SSD and radio antenna.

The SD stations complete the hybrid detector concept of the observatory. However, due to the mismatch between the uptime of the SD and the FD, the latter covers only a subset of the phase space of the former. Hence, we have to account for differences in hybrid data sets.

2.3.3 Environmental monitoring

Since the atmosphere acts as a huge, inhomogeneous calorimeter for EAS, both the FD and the SD measurements depend on the environmental conditions [P:61]. Although atmospheric monitoring is much more critical for the former, these conditions also affect the SD. For example, on very cold days, the flat SD detector plane has a larger atmospheric depth X. Consequentially, the showers observed are in a later stage of development than usual. Hence, an essential part of the operation of the observatory is the constant survey of environmental parameters, such as the air density and the temperature.

Weather stations constantly measure temperature and atmospheric pressure at different parts of the observatory. At the FD sites, there are cloud cameras installed to estimate cloud coverage. To estimate the atmospheric profile, two laser facilities [P:62] are employed: the central laser facility (CLF) and the eXtreme laser facility (XLF). They emit light at a well-defined intensity and frequency. All FD sites detect this light to calibrate the energy estimate.

This procedure can be cross-checked via the Light Detection and Ranging (lidar) stations at each FD site [P:63]. These send pulses of laser light into the sky and measure the light back-scattered. The lidar stations are able to scan the entire sky over the observatory. Hence, they obtain data about cloud coverage and cloud height.

2.3.4 Upgrade: AugerPrime

Auger has been taking data since 2004. Initially, it was funded until 2015. However, to further benefit from the established infrastructure, the observatory is currently undergoing an upgrade, called AugerPrime (AP), to extend its lifespan [P:64]. This endeavor is focused not

only on improving available detectors but also on adding new detectors to the SD. These should provide complementary measurements of the shower to better separate the muonic and electromagnetic sub-component. In the best-case scenario, this separation would make it possible to estimate the cosmic ray mass on a shower-to-shower basis. The upgrade extends the operation time of Pierre Auger Observatory well into the 2030s, allowing for a significant increase in statistics.

For this work, the most important parts of the upgrade are the addition of the Surface Scintillator Detector (SSD) on top of the WCD stations (see Fig. 2.6) and the replacement of the UB (see Sec. 2.3.2) with the Upgraded Unified Board (UUB).

A Upgraded Unified Board

The additional detectors make it necessary to upgrade the old UB electronics. This upgrade comes in form of the UUB, which is a more powerful version of the old UB. It increases the sampling rate by a factor of three to 8.3 ns and comes with a larger digitization size of 12 bits for each Fast Analog/Digital Converter (FADC) bin. In addition, the new board also yields an increase in computation power and memory by a factor of ten [P:65]. As a consequence, the upgrade also allows for the employment of advanced trigger concepts such as that Neural Network (NN) triggers [A:4].

B The Surface Scintillator Detector

The SSD provides a direct upgrade to the station-based detection principle of Auger (see Sec. 2.3.2). It runs in slave mode to the WCD and provides an alternative measurement of the shower footprint. Each SSD is a 4 m by 2 m flat detector mounted on top of the WCD stations. It consists of multiple scintillator bars containing optical wavelength-shifting fibers that transfer the light yield to a central PMT. The calibration of the SSDs follows that of the WCDs. However, the signal unit, in this case, is called Minimum-Ionizing Particle (MIP).

In contrast to the WCD, the response of the detector is almost independent of the track length of the particles passing through the detector. Hence, the particles of the electromagnetic sub-component of the shower cascade (see Sec. 2.2.3) produce similar signals as those of the muonic sub-component. By using the signals of the WCD and the SSD, it might be possible to disentangle the signal of the different shower components.

C Other features of the upgrade

Additionally, apart from the SSD, there are other significant improvements implemented during the upgrade process. These focus primarily on enhancing the hybrid detection principle even further. In general, there is no reason why the presented methods in the following chapters could not be used for these new parts of the observatory to improve the reconstruction of physical quantities. The major upgrades are listed here in no particular order.

Small Photomultiplier Tube The saturation limit of the WCD is relatively low. Since the main interest of the observatory lies in the detection of UHECRs, this is rather inconvenient. The saturated stations introduce biases in all of the available analyses that rely on station signals. A simple workaround for this is to add another PMT in the WCD to complement the three regular ones. This Small Photomultiplier Tube (SPMT) will extend the dynamic range of the WCD greatly, allowing for signals with up to 20 kVEM [P:65]. In addition, this matches the dynamic range of the SSD, aligning both of those detectors further [P:66].

Undergound Muon Detector The Underground Muon Detector (UMD) is an extension to the 750 m-array (see Fig. 2.5) and, in particular, to the Auger Muons and Infill for the Ground Array (AMIGA) [P:67]. Its target energy range lies around the ankle in the CR spectrum. It is a direct muon detector using the soil as a natural filter for the electromagnetic sub-component. This allows for an unbiased estimation of the lateral muon density for muons above 1 GeV [P:68].

Radio Detector Even though all of the improvements discussed previously enhance the hybrid detection, this is only true for showers with zenith angles below 60°. Horizontal air showers are hard to detect with the current setup, and specialized analyses are required to gain any knowledge from them [P:69]. This is due to the strong attenuation of the electromagnetic component of the air shower by the atmosphere for very inclined showers. However, this component is still detectable by radio antennas. The radio detector is another addition to the WCD tank (see Fig. 2.6) which measures the coherent radio emission of air showers [P:70]. An advantage of this detection principle is that it provides an alternative energy scale for CR to that given by the FD [P:71].

2.4 Science case: key results and open questions

Understanding the physics behind CRs provides insights into the most extreme microscopic and macroscopic processes in the Universe. On the one hand, we can analyze at particle energies of UHECRs that are currently out of reach for modern particle accelerators. This, in turn, could yield new advances in microscopic theories such as QCD. On the other hand, in the future, we probably will understand what kind of accelerators in which galaxies could be the source for these particles.

Although the recent advances and discoveries in UHECR physics are driven by the new generation of CR detectors, such as Auger [P:72], there are still many open questions that need to be answered. In the following, we highlight a subset of these results and mysteries that are important in the context of this thesis.

2.4.1 Muon deficit

One of the most challenging questions that UHECRs pose today is the over-abundance of muons. If Monte Carlo (MC) air shower simulations (see Sec. 2.2.6) are compared to measurements of shower footprints, the number of predicted muons is much lower than the measured number [P:50, P:73].

If we trust modern hadronic models, we would obtain a mass composition dominated by nuclei heavier than iron at very high energies. The quantification of this effect is one of the key results of Auger. To measure the muon deficit, we need data from shower footprints dominated by the muon sub-component. One method of doing this with the SD main array is the data pre-selection of very inclined air showers [P:50] (see Fig. 2.7) for which the electromagnetic component is attenuated by the atmosphere. The obtained signals are then compared to the simulations using the modern hadronic models (see Sec. 2.2.6).

The muon deficit also poses unique problems to newly developed analyses. Before using them on measurements, usually, these need to be tested on simulations to quantify their systematic errors and predictive power. Due to the mismatch of the simulations, however, this is somewhat tricky. The process of going from simulations to measurements is non-trivial and has to be newly explored for each employed method.

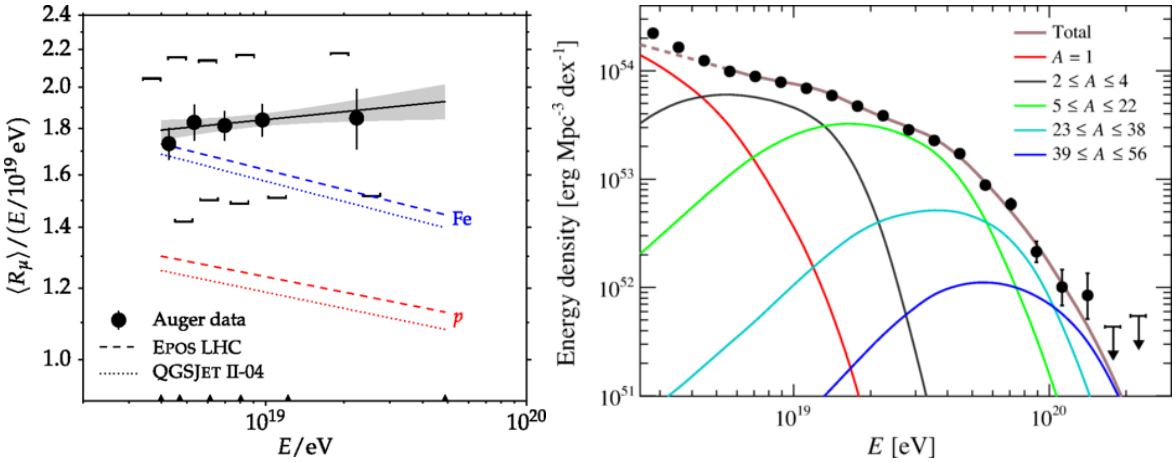


Figure 2.7: *Left*: Relative muon content as a function of the reconstructed energy for very inclined air showers compared to expected muon contents from CORSIKA simulations using different hadronic interaction models. The shower cascade of very inclined air showers is, due to the attenuation, mostly comprised of the muons at the ground level. Using these measurements reveals a substantial deficit of muons if compared to the simulations of QGSJetII-04 and EPOS-LHC (taken from [P:50]). *Right*: Best fit results of a mass composition fit to the energy spectrum published by Auger (taken from [P:74]). In this model, the amount of protons decreases strongly at around $1.1{\times}10^{19}$ eV just around the start of the GZK suppression region (see Sec. 2.2.2). Afterward, the composition gets heavier and heavier.

2.4.2 High-energy suppression of the flux and anisotropy

Initially, one of the driving ideas behind Pierre Auger Observatory was to check if there really is a high-energy suppression (see Sec. 2.2.2) in the spectrum. Relatively early in the lifetime of the observatory, ten years of taken data have shown that this is unambiguously the case [P:75]. The suppression is clearly visible in the spectrum at about 40 EeV [P:76] (see Fig. 2.2). This disproves the results of other CR detectors, such as AGASA [P:21].

Although the suppression is very close to the value predicted by the GZK-effect, it is still unclear if the GZK is the reason for this. The exact fraction of protons at this high energy is unknown. Hence, an acceleration limit of local sources could also explain the suppression. We could unravel this if we had better knowledge about the mass composition of CRs above 30 EeV, such as methods that could estimate the mass at a shower-to-shower level.

It is also notable that Auger has provided the most precise measurement of the ankle region [P:26, P:74] up to date. One of the more recent results from Auger is the slight anisotropy in the flux [P:27, P:28]. For energies above the position of the ankle, there is a 5σ dipol in the flux pointing outwards of our Galaxy (see Fig. 2.8). This result supports the assumption that from this energy on UHECRs are mainly extra-galactic. Even though there are feasible candidates, no definite source of these overdensities could be identified [P:24].

2.4.3 Neutral messengers

EASs are not restricted to charged nuclei. Neutral particles, such as photons and neutrinos, can also induce air showers (see Sec. 2.2.3.A). These exotic air showers are fundamentally different from the hadronic ones. Hence, it is much easier to distinguish them from the hadronic air showers.

Currently, Auger focuses on providing limits for the flux of neutral messengers. These flux limits help rule out exotic non-standard physics, such as special Lorentz invariance violation (LIV) models [P:77]. Furthermore, since magnetic fields do not affect them, it is possible to

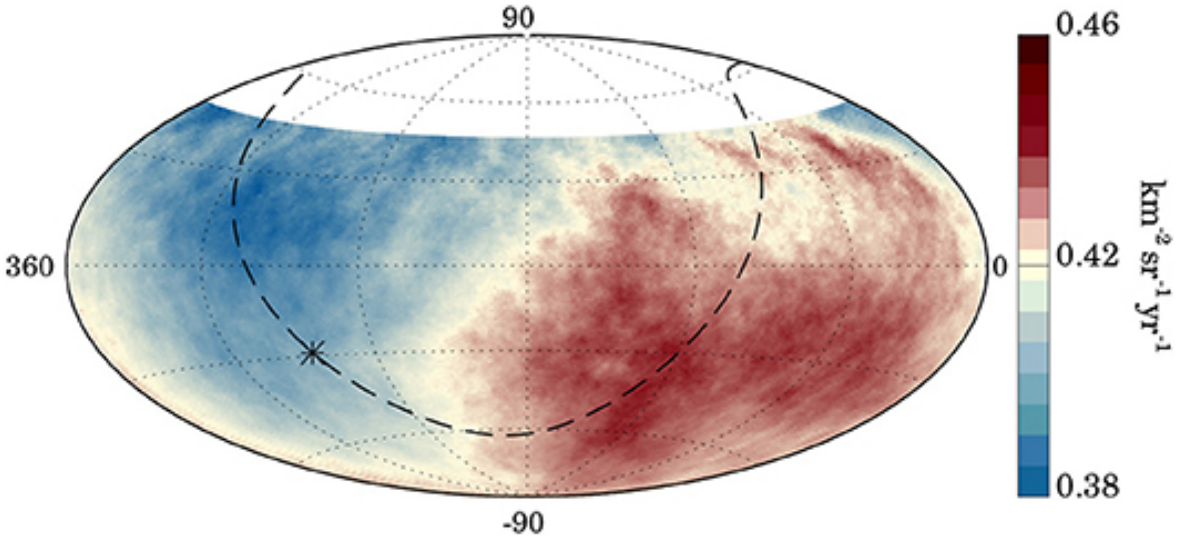


Figure 2.8: Dipol anisotropy of CR spectrum for CRs above 8 EeV (equatorial coordinates). Auger alone does not cover the entire sky due to its fixed position in Argentina. This is seen in the white region on top of the plot. Taken from [P:24], originally from [P:27].

point toward their source more easily. This is especially interesting in the age of Gravitational Wave (GW) telescopes. After the detection of GW events, high-energy neutrinos or photons could give a follow-up signal.

2.4.4 Mass composition

To estimate the mass composition of UHECRs, mass-sensitive high-level observables are of particular interest. Comparing their statistical moments to theory yields knowledge about the gradual change of the composition itself. One such parameter is the depth of the shower maximum: on average, proton showers are deeper than iron showers. This has the advantage that it is directly measurable with FD. In Fig. 2.9, we show the first and second moments in bins of energy compared to the theoretical prediction of QGSJetII-04. Over the target energy range, the composition changes from a light to a more heavy composition. Another – even more – mass-sensitive parameter is the muon content of the shower. Currently, we are only able to measure it via highly inclined events. However, future works are going to benefit benefit from AMIGA and UMD enabling direct detection.

To address most of the open questions in UHECR physics, we must have an accurate knowledge of the mass composition at the highest energies. This boils down to the development of novel analysis methods that, at the time of writing, can potentially tap into the power of the new detectors added by AP. To test such methods is also one of the objectives of this thesis.

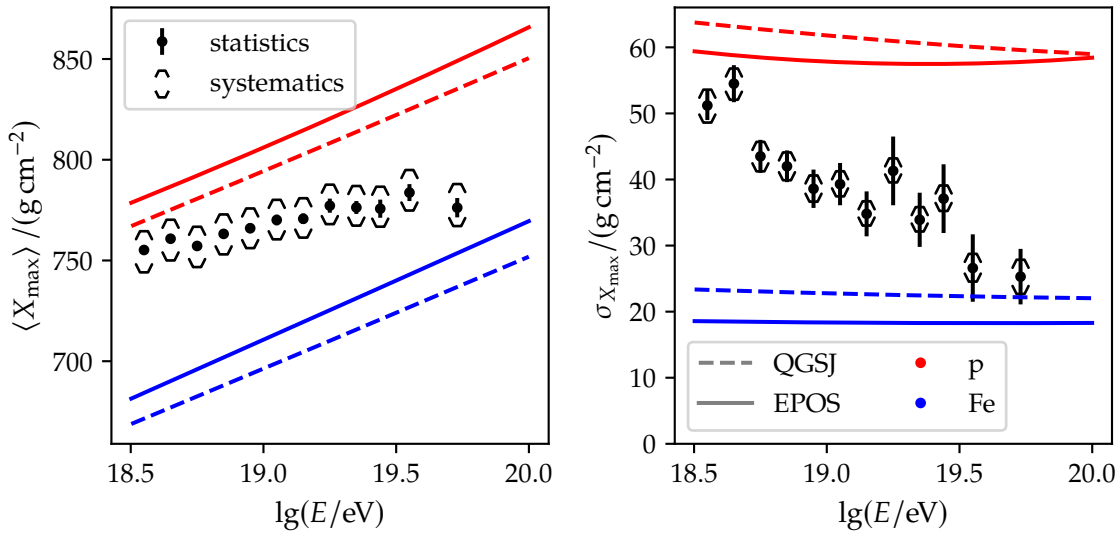


Figure 2.9: First (*left*) and second moment (*right*) of the depth of the shower maximum X_{max} measured by FD binned in energy. The red and blue dashed line depict the expected values (see Sec. 5.4.2) from QGSJetII-04 for protons and iron showers, respectively. We see a clear indication that the mass composition changes from light to heavy towards higher energies. Adapted from [P:78].

3 Current Reconstruction in Use



Words can be communicative only between those who share similar experiences.

(Alan Watts, Philosopher)

DALL·E 2 prompt:

A 3D rendering of a cylindrical water-Cherenkov tank having an antenna and a solar panel built on top[.]

To contextualize the NN-based models used in this thesis, we have to discuss the regular reconstruction of high-level shower observables in simulations and measurements taken by Auger. We define these observables as generic shower properties that are not directly measured by the participating detectors, such as the inclination (zenith) angle θ or the primary particle energy E. We use these reconstructed high-level observables in three ways: as direct inputs for our NN, as a tool for simplifying our input phase space, and as a direct comparison to our predictions. For example, to normalize our input data, we use geometric shower properties to align our showers in the same way (see Sec. 5.3.2).

In addition to standard methods, we review advanced methods that try to estimate mass-sensitive observables of showers since they have the same goals as the presented NN analysis. These methods distinguish themselves by not being a direct part of the commonly used analysis chain of the $\overline{\text{Off}}\underline{\text{line}}$ framework. This framework is currently considered the primary analysis tool of the Pierre Auger collaboration.

In Sec. 3.1 we shortly present the $\overline{\text{Off}}\underline{\text{line}}$ analysis framework and the standard data format used by the Auger. Afterward, in Sec. 3.2, we dive into selected parts of the standard reconstruction following the implementation in $\overline{\text{Off}}\underline{\text{line}}$. This part primarily focuses on the SD side of things since we use the SD array in the later analysis as well. In the end, we review the advanced non-NN-based reconstruction methods in Sec. 3.3.

3.1 $\overline{\text{Off}}\underline{\text{line}}$ analysis framework

Currently, the $\overline{\text{Off}}\underline{\text{line}}$ Analysis Framework ($\overline{\text{Off}}\underline{\text{line}}$) is the main[1] analysis tool of Auger [T:E]. Its primary purpose is the up-to-date standard reconstruction of high-level shower observables from measurements taken by the various detectors of the observatory. Furthermore, it is also used for standard analyses, to simulate events from air-shower simulations, such as CORSIKA or AIRshower Extended Simulations (AIRES), and various data preparation tasks.

At its core, $\overline{\text{Off}}\underline{\text{line}}$ is a modularly structured program written mainly in the `C++` programming language. Essentially, it provides an easy way of running different subprograms, called modules, in a certain order via `.xml` steering files, called bootstraps. Each module can be invoked via these bootstraps, where standard parameter values and standard processes can be overridden. This makes it simple to use – even for non-programmers. In addition, $\overline{\text{Off}}\underline{\text{line}}$ provides all of its functionality in the form of a `C++` library to enable advanced usage of its algorithms in custom applications.

$\overline{\text{Off}}\underline{\text{line}}$ saves information[2] in the `ROOT` [T:F] data format. This format provides a tree-like data structure which is an ideal way to represent event-based measurements of physics experiments. In the case of Auger, the topology of the data trees is defined via a self-written `C++` class hierarchy which is called Auger Data Summary Trees (ADST). In general, an ADST contains a jagged data structure. Different shower events trigger different amounts of stations. Consequentially, for each "event row", we have a differently sized chunk of memory. Hence, despite the efficiency of the format the "non-linearity" adds complexity which makes it potentially slower for high-throughput processes, like in the case of NN. Therefore, before using the data reconstructed with $\overline{\text{Off}}\underline{\text{line}}$, we transform all data into chunks of Hierarchical Data Format (HDF5) files. We have chosen the HDF5 data format due to its wide-spread use, its great support in almost all programming languages, and its straightforward way of saving data in rectangular memory. To convert ADST files to HDF5 files, we use a converter written in `C++` that acts as an interface between the $\overline{\text{Off}}\underline{\text{line}}$ ADST and the HDF5 library.

To identify certain fields in the ADST data set we use the following convention. We name variables via a combination of the name of the class they belong to and the name of the variable indicated by its getter method:



. (A)

Almost all quantities used in the later stages of this thesis come directly from the ADST files. They only have to be pre-processed (see Sec. 5.3). However, some quantities have to be first computed from these base quantities, and others are extracted from the CORSIKA files directly.

Note that all air shower events used in this thesis (see Chapter 5) have been produced by a recent version of $\overline{\text{Off}}\underline{\text{line}}$ in the active development branch. All high-level variables, such as the zenith angle θ, are taken from the reconstruction algorithms also used in data analyses.

3.2 Standard reconstruction

Before we discuss the standard reconstruction, we first introduce the most essential terminology. In this thesis, we work exclusively with showers that have been detected by the WCD stations of the SD. This data set is commonly called the SD data set. Another important data

[1] There is/was another independent reconstruction framework called built into the CDAS software. However, this framework is not in development anymore. Only legacy version remain.

[2] There are ways in $\overline{\text{Off}}\underline{\text{line}}$ to save the output in other file formats. However, `ROOT` files are the most common.

set is the Hybrid data set, consisting of events for which the FD also detects a signal. To calibrate the SD energy predictor (see Sec. 3.2.3), this Hybrid data set needs to be pre-selected. We want only high-quality measurements which can be reconstructed by both detectors independently and sufficiently well. For example, events for which the depth of the shower maximum X_{max} is not in the field of view (FOV) of the FD camera could have an undetermined bias and must be removed. This process yields the subset called the Golden Hybrid data set.

A large source of uncertainty in the standard analysis lies in the so-called shower-to-shower fluctuations. These are stochastic in nature and caused mostly by the randomness of the first couple of particle interactions. Two showers induced under the same conditions would show different shower footprints. This is especially true for lighter primaries, which are not subject to the superposition of multiple nuclei (see Sec. 2.2.3.B).

Since the main topic of this thesis is the application of an Artificial Neural Network (ANN) to extract information from shower footprints, we will only discuss the standard reconstructions involving the WCD of the SD and selected parts of the standard reconstruction of the FD. At the time of writing, no standard method of using the SSD in the reconstruction was available. We only provide basic information in this case.

3.2.1 Overview of the FD reconstruction

The FD measures the (binned) time signals induced by fluorescence light along a projection of the longitudinal shower profile [P:52]. Hence, the distance to the shower axis and its exact propagation direction is initially unknown. However, due to the (signal) timing information, it is still possible to determine the direction the shower came from, assuming the trajectory of the shower is well approximated by a straight line. We call this line the shower axis. It is collinear with the impulse of the primary particle pointing towards its extended trajectory. To estimate the shower axis, we need to know the shower detector plane (SDP). The SDP is the unique plane that is spanned by the detector position and the shower core (see Fig. 3.1). Assuming that the emitted light is traveling with the speed of light c and the shower core lies on a straight line, the trigger time t_i of pixel i of one of the telescopes of the FD can be modeled as

$$t_i = t_0 + \frac{R_{\mathrm{p}}}{c} \tan\left(\frac{\chi_0 - \chi_i}{2}\right), \tag{3.1}$$

where χ_i is the angle corresponding to pixel i, χ_0 is the angle between shower axis and ground plane in the SDP, and R_{p} is the minimum perpendicular distance between telescope and shower at the time t_0. Using χ^2-square minimization, we obtain estimates for χ_0, t_0, and R_{p}, which together define the shower plane (see Fig. 3.1).

After extracting the geometrical information, we are able to evaluate the time signal information to estimate the longitudinal shower profile (see Sec. 2.2.4). However, the measured signals represent only the photon flux. Hence, we need to convert this flux into the deposited energy $\mathrm{d}E/\mathrm{d}X$ as a function of the slant depth X (see Eq. (2.4)). This process is quite challenging. First, we need to disentangle the flux into different sub-components, and afterward, the attenuation needs to be accounted for. In the end, we convert the corrected flux by using the fluorescence yield [P:79].

To estimate X_{max} and the calorimetric energy E_{cal} of the shower, the algorithm performs a fit of the resulting profile to the Gaisser-Hillas function in Eq. (2.14) (see Fig. 3.1). The shower depth of the shower maximum X_{max} is a free fit parameter. We obtain it directly. It should be mentioned, as discussed in Sec. 2.2.5 and Sec. 2.2.4, that this is the only way of measuring the shower depth of the shower maximum, which makes it invaluable for the cross-test of other methods that try to reconstruct it from indirect measurements at Auger. To extract the shower energy, we need to perform an additional step. We obtain the calorimetric energy

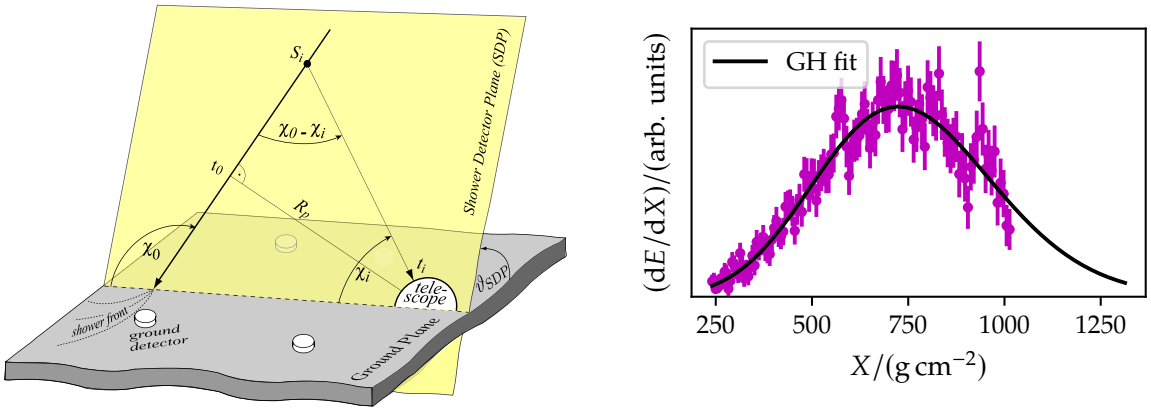


Figure 3.1: *Left*: Illustration of geometry reconstruction using FD (see Eq. (3.1)) and the SDP. Taken from [P:53]. *Right*: Energy deposit per slant depth (dE/dX) of simulated longitudinal shower profile (magenta points). The black line shows a best fit using the GH function (see Eq. (2.14)).

of the shower from an integral of the longitudinal profile function over X. To convert this integral into the primary particle energy, we must account for the fraction of shower particles that are invisible to FD, such as muons which hardly interact with the atmosphere [P:80].

Note that although Eq. (2.14) fits the profiles quite well, it has limitations. Sometimes the fit gives parameter values – usually associated with physical processes – that are unphysical, such as negative values of X_1. The fit function is only phenomenological, having problems reproducing the shower profile at the earliest interactions. This is not only the case for measurements that are subject to noise and imperfect measurement conditions but also for simulations (see Sec. 2.2.4).

To obtain the shower profile from the FD, we have to follow the reconstruction procedure described above. Each step introduces systematic uncertainties due to the assumptions and physical measurements. All uncertainties that affect the profile also directly affect the estimation of X_{max}. Accounting for all these uncertainties, we obtain a precision of about $15\,\text{g/cm}^2$ at the highest energies [P:81]. In addition, to estimate the energy, we also have to account for systematic errors introduced by the integration and the transformation from calorimetric to shower energy. These uncertainties on the FD energy scale amount to 14% [P:82, P:83, P:84].

This energy measurement is instrumental since it can be used to determine the energy scale of the SD of Auger. Without this "more" direct measurement, the energy predictors would need to be calibrated against simulation data. This, in turn, would give us much larger systematic uncertainties because of inconsistencies, such as the muon deficit (see Sec. 2.4.1) and differences between the hadronic models.

Since FD is based on the direct observation of the shower profile, the quality of the reconstruction relies on the amount of longitudinal profile which is inside the FOV of the FD telescope (see Fig. 3.2). However, we cannot directly remove events exhibiting incomplete profiles. This could yield a composition bias due to distortion of the measured X_{max} distribution. For the FD reconstruction, a data-driven approach, called fiducial FOV cut, is used to ease these problems [P:81, P:85]. The acceptance boundary $[X_{\text{l}}, X_{\text{u}}]$ is parametrized [P:86] in such a way as to ensure that the mean shower depth value in any given energy interval does not deviate more than $5\,\text{g/cm}^2$ when shrinking the interval.

By further requiring a precision below $40\,\text{g/cm}^2$ and a minimum observation angle above 20°, we obtain a high-quality data set that can be used for calibration purposes (see Sec. 3.2.3). In Sec. 5.2, we discuss the effect of this quality selection on the amount of data.

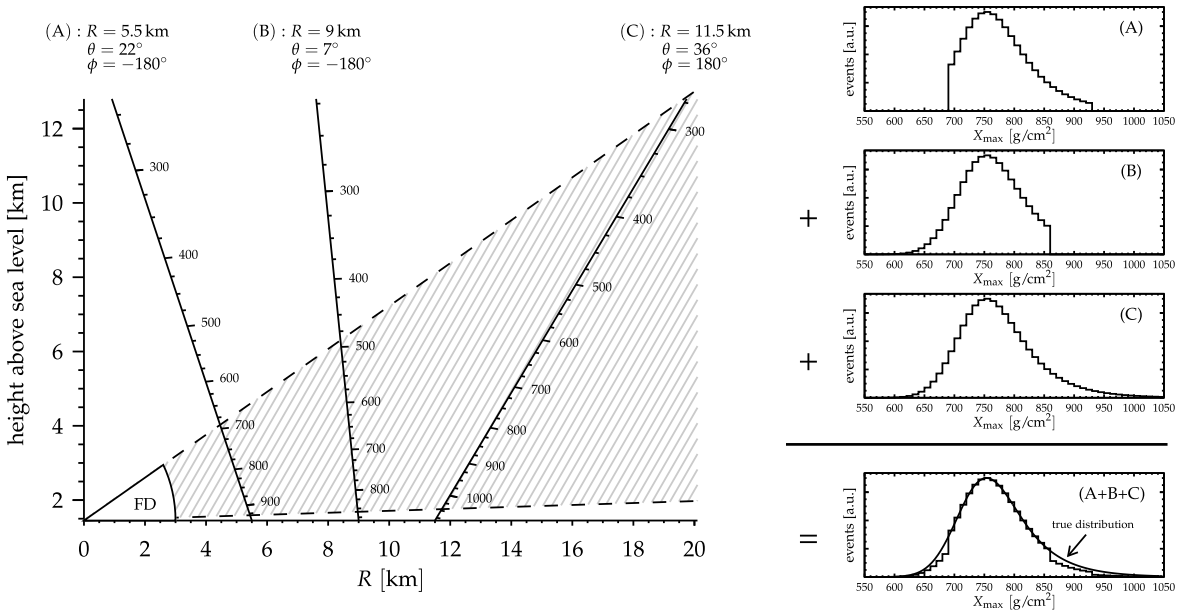


Figure 3.2: Illustration of the influence of the FOV of the FD telescopes on the sampling and subsequent reconstruction of the longitudinal shower profile. It depends on both the shower geometry and the distance to the observing telescope. If these are unfavorable, the measured profiles are truncated. Our preferred event geometry is that denoted in panel (C). Taken from [P:81].

3.2.2 SD reconstruction using the WCD stations

Fundamentally, while the FD measurement represents a projection of the longitudinal component of the shower, the SD observes points of the time-dependent footprint at the ground. The footprint represents a planar cut through the space-time profile of the shower cascade at the ground. Hence, the effective size[3] and the shape of the footprint are dependent on the inclination angle θ of the shower. For increasing θ, the more elliptic and larger the effective area is. Moreover, the effective size of this footprint also scales with primary particle energy E. The larger this size, the higher the number of stations that will potentially trigger. Hence, E and θ correlate directly to the effective footprint size.

Commonly, the station with the largest deposited signal is called the Hottest Station (HS). Due to the triangular structure of the SD-array grid (see Sec. 2.3.2) the stations around this station form differently sized hexagons (see Fig. 3.4). We call these unique hexagons "crowns" if the center of the hexagon is the HS. Therefore, the first crown is formed by the six[4] stations closest to the HS. All following crowns are built in the same manner. The (maximum) number of stations in crown n is $6n$. With high probability, the HS is also the closest station to the impact point of the shower core on the ground.

As for the FD, the reconstruction of SD events – after the triggering process – has two steps. In the first step, the shower geometry needs to be reconstructed. Afterward, we use this knowledge to perform a fit of the Lateral Distribution Function (LDF) (see Eq. (2.17)).

A High-level trigger conditions of the SD

Each WCD[5] constantly measures the deposited signal. This comes mainly from background particles. If all of the working PMTs of a station measure a coincident signal above a certain threshold, the local station triggers, sending this information to the CDAS. If at least three of

[3] In the sense of the area that where the signal is above the WCD threshold.
[4] In certain parts of the array, e.g. the edge of the SD, this number can be smaller.
[5] This is also true for any other detector on top of the WCD.

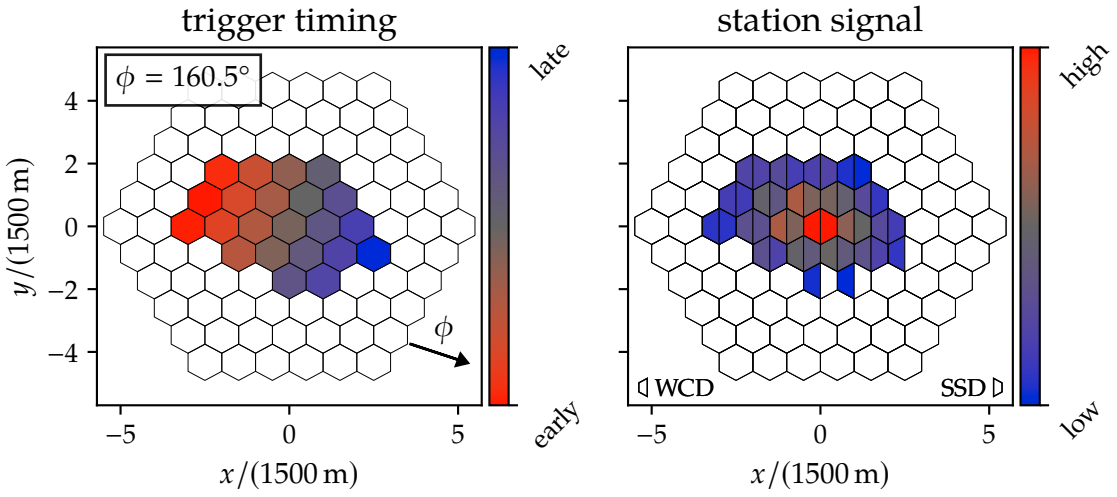


Figure 3.3: Representation of a shower footprint for a shower induced by a simulated iron primary. We placed the HS in the center of the triangular grid. Colored hexagons signify triggered stations. *Left*: Timing of the WCD triggers relative to the HS. The arrow indicates the direction the shower propagates to and ϕ is the corresponding azimuth angle. *Right*: Signals of the WCD and SSD normalized to the WCD and SSD signal in the HS. The left part of a hexagon shows the WCD, and the right part the SSD signal strength. A partially white hexagon indicates that the corresponding detector exhibits no signal. Since the SSD has a smaller effective detection area, it is possible that it does not detect a signal even if the WCD does.

such triggers happen in a time window of 50 µs, CDAS checks for their spatial correlation. To build an event from the triggers, they must satisfy specific trigger patterns depicted in Fig. 3.4.

These events – called T3 events – are potentially just coincidences that might happen when, for example, atmospheric muons trigger stations. Hence, they have to pass additional, more strict criteria[6] before being considered for reconstruction. A necessary condition is that the trigger times of the stations in a T3 event must be compatible with a planar shower front moving with the speed of light. If all criteria are met, the event is promoted to a T4: a physics event.

To ensure that T4 events are of high quality, there is another high-level trigger. This trigger can be applied before or after the reconstruction of the event. We focus only on the latter. After reconstruction, the station nearest to the impact point of the shower core must have at least n functioning stations in its first crown. If this is the case, the event is promoted to an nT5, where $n \in [4, 5, 6]$. The most valuable of these are the 6T5 events. Among other things, they are used in the SD energy calibration.

B Geometric reconstruction of showers using their footprint

The shower geometry is entirely defined by the shower core (see Sec. 2.2.3). Since it is a straight line, we need to find its direction vector and one unique point the line intersects. For the former, we want to use the collinear unit vector and denote it as the shower axis $\hat{\boldsymbol{a}}$. As a unique point, it is convenient to use the impact point of the shower core on ground-level $\boldsymbol{r}_\mathrm{c}$.

We are able to obtain $\hat{\boldsymbol{a}}$ and $\boldsymbol{r}_\mathrm{c}$ by a fit to the shower front. The shower front is a sphere

[6] The procedure identifies approximately 99% of physics events. For example, the conditions reject T3 events induced by lightning.

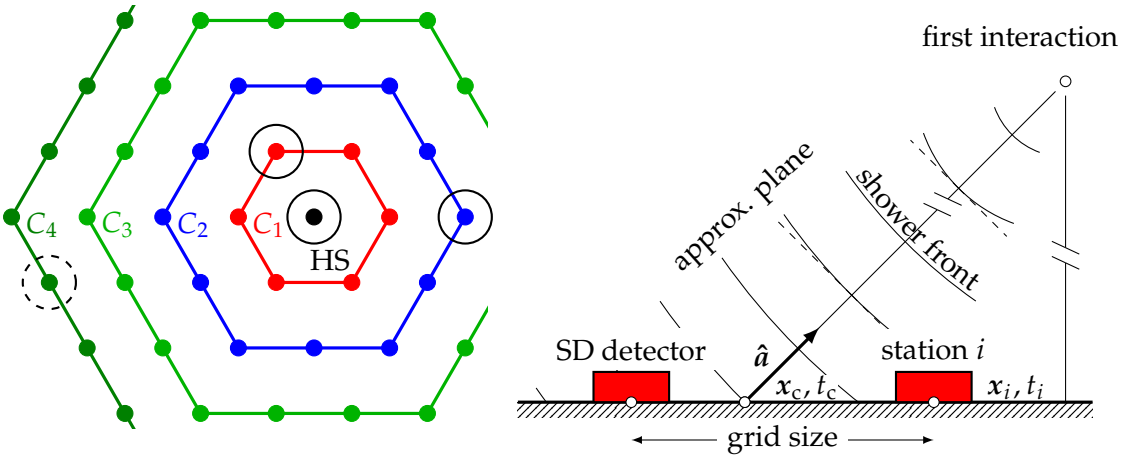


Figure 3.4: *Left*: Two examples of coincidence trigger (T3) pattern. The different colored lines make up concentric hexagons around the HS. The nth hexagon is called the nth crown. We denote it as C_n. The solid circles represent an example topology for a 3-fold coincidence. For such a coincidence, all stations must satisfy a time-over-threshold trigger and at least one triggered station must be in C_1 and a second one not further than C_2. If we also add the dashed circle, we obtain an example topology for a 4-fold coincidence. For such a coincidence all, participating stations can be any valid single station level trigger where the additional station can be as far as in C_4. *Right*: Sideview of SD setup for shower plane measurements (see Eq. (3.2)). Far away from the point of first interaction, the shower plane can be approximated by a plane front.

inflating with c originating from the point of first interaction. Far from it (see Fig. 3.4), the shower front is roughly planar. We exploit this to gain prior knowledge of $\hat{\boldsymbol{a}}$.

Let $(\boldsymbol{x}_i, t_i)$ be the four-vector inside the shower plane at station i which triggered at the time t_i. Since the shower plane propagates with constant speed c, the four-point of station i and station j must fulfill

$$c(t_i - t_j) = \hat{\boldsymbol{a}} \cdot (\boldsymbol{x}_i - \boldsymbol{x}_j). \tag{3.2}$$

Therefore, we can use a direct fit by choosing an arbitrary reference point. From the fit, we obtain the reference time and the shower axis for the plane front approximation. We denote it as $\hat{\boldsymbol{a}}^{\mathrm{pf}}$. If an event contains only three stations, the fall-back value is derived by solving Eq. (3.2) exactly by constructing a linear system of equations.

To describe the curved shower front, we need to slightly modify Eq. (3.2). We want to find a (virtual) point of origin $(\boldsymbol{x}_\mathrm{o}, t_\mathrm{o})$. Assuming the shower front is spherical, we obtain

$$c_0(t_i - t_\mathrm{o}) = |\boldsymbol{x}_i - \boldsymbol{x}_\mathrm{o}|. \tag{3.3}$$

The point of origin is obtained via a fit that accounts for the uncertainty of the timing. To estimate the shower axis $\hat{\boldsymbol{a}}$, another point on the shower core is required. For this, we use $\boldsymbol{r}_\mathrm{c}$. A rough approximation of $\boldsymbol{r}_\mathrm{c}$ is the barycenter of all stations weighted by their signals. This is used as an initial value in the LDF fit (see Sec. 3.2.2.D).

C Time signals of the UB-WCD

The total signal S_tot of WCD station i is the integral over the time signal called traces, detected by the PMTs. A UB trace has 768 bins (see Sec. 2.3.2). However, the signals are normally much shorter than this time window. The signal start bin b_s (`SdRecStation.SignalStartSlot`) marks the start of a signal. This is also the bin that approximately corresponds to the trigger time of the detector. In Analog/Digital Converter (ADC) counts, the total trace signal of a

PMT is

$$S_{\mathrm{ADC}} = \sum_{b=b_{\mathrm{s}}}^{b_{\mathrm{e}}} (S_{\mathrm{ADC}}(b) - B(b)), \tag{3.4}$$

where $B(b)$ is the baseline of the FADC [P:58] and b_{e} is the end bin of the trace given by $\overline{\mathrm{Off}}\underline{\mathrm{line}}$ (`SdRecStation.SignalEndSlot`). To convert this to a value that is independent of the used electronics, we require the station calibration value Q_{VEM} (see Sec. 2.3.2) that yields the equivalent of integrated counts to the average signal deposited by a vertically through-going muon. Using the signals of all working PMTs, we obtain the total signal via

$$S_{\mathrm{tot}} = \left\langle \frac{S_{\mathrm{ADC}}}{Q_{\mathrm{VEM}}} \right\rangle. \tag{3.5}$$

Historically[7], time traces $S_{\mathrm{off}}(t)$ taken from $\overline{\mathrm{Off}}\underline{\mathrm{line}}$ are given in units of the VEM peak, I_{VEM}. Hence, the integral over $S_{\mathrm{off}}(t)$ does not correspond to the deposited signal. Hence, we convert it using the area over peak a_{p} (see Eq. (2.18)),

$$S(t) = S_{\mathrm{off}}(t)/a_{\mathrm{p}}. \tag{3.6}$$

In Fig. 3.5, we show a trace taken from a simulated oxygen event. We highlighted the start and end bins via vertical dashed lines.

In simulations, we can access the sub-traces coming from the sub-components of our shower (see Fig. 3.5 (right)). These play a vital role in the Universality method discussed in Sec. 3.3.2. Due to the fact that the total and the component traces are simulated independently, the sum of all sub-traces does not match precisely the total trace.

The trace is made up of the deposited signal of many particles. Each particle j produces a detector response at the time t_j that follows roughly a one-sided exponential decay [P:87]. Therefore, the expected trace follows

$$S(t) = \mathrm{const} \sum_j R_j \Theta(t - t_j) \exp\left(-\frac{t}{\tau}\right), \tag{3.7}$$

where R_j is the magnitude of the response, which is proportional to the deposited energy of the particle, Θ is the Heaviside function, and τ is a decay constant that depends mainly on the absorption properties of the water in the WCD and the reflectivity of the liner. Moreover, τ also partially depends on the electronics of the PMTs.

D Lateral distribution function of total signals

Since the triggered SD stations only sample parts of the shower footprint, we need to define a self-consistent way to estimate the "magnitude" of the corresponding shower. We denote this estimate as the shower size.

Since signals correspond to particles traversing through our detectors, we parameterize the LDF of the signals on the ground level with the modified NKG function (see Eq. (2.17)). Keeping its functional form, we obtain

$$S(r) = S(r_{\mathrm{opt}}) f_{\mathrm{mNKG}}(r) = S(r_{\mathrm{opt}}) \left(\frac{r}{r_{\mathrm{opt}}}\right)^{\beta} \left(\frac{r + r_{\mathrm{s}}}{r_{\mathrm{opt}} + r_{\mathrm{s}}}\right)^{\beta+\gamma}, \tag{3.8}$$

where r_{s} is set to 700 m for all arrays due to its strong correlation with β and r is the perpendicular distance to the shower core called the shower-plane distance. We define $S(r_{\mathrm{opt}})$ as the shower size. The choice of r_{opt} depends on the array geometry.

[7] In this way, a single muon would on average have an amplitude of 1.

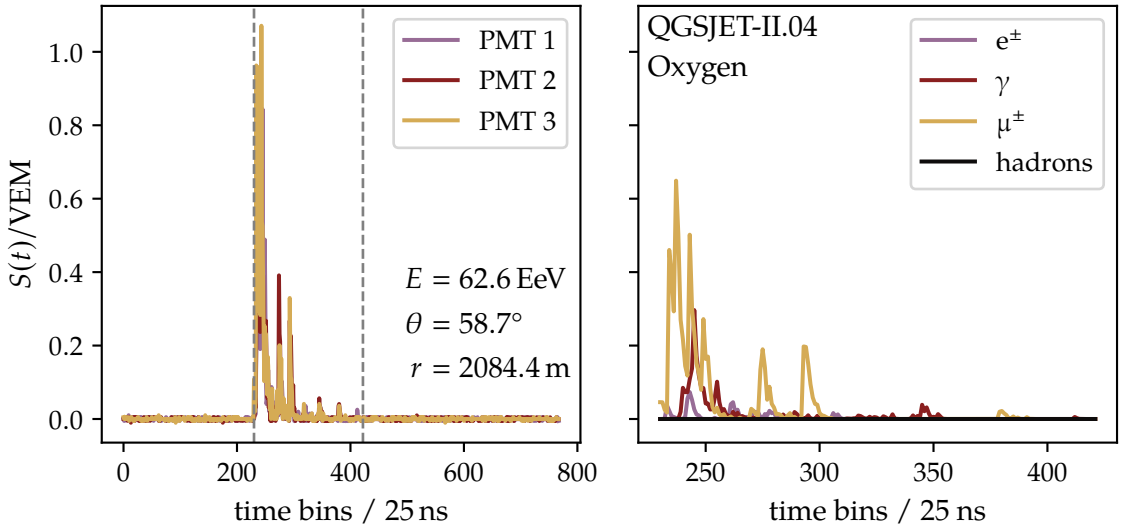


Figure 3.5: Example UB PMT traces of a triggered detector for a simulated oxygen-induced shower using the QGSJetII-04 model of hadronic interactions (left) and averaged traces of sub-components in the region of interest (right). The response of all PMTs differs only marginally for this inclined event. The signal is dominated by muons. This is due to the high inclination of the shower and the large distance to the shower axis.

Since we want to build an estimator, we want the signal at r_{opt} to be only minimally fluctuating. For the Auger SD main array, this distance is 1000 m, for the 750 m-array it is 450 m. To fit the LDF an elaborated log-likelihood minimization is used that also takes non-triggered stations and saturated stations into account [P:52]. We call candidate stations, all those that trigger due to the air shower. To enable the reconstruction of low-multiplicity events, β and γ are parameterized using high-multiplicity events. The LDF fit (see Fig. 3.6) provides us with the expected signals at each distance to the shower axis, assuming no asymmetry between up- and downstream stations[8].

Unfortunately, due to attenuation, S_{1000} strongly depends on the amount of traversed atmosphere and, consequentially, on the zenith angle θ. To make shower events relatable, we want to obtain the expected value of S_{1000} at some reference zenith value. Assuming that the event distribution is isotropic, the number of detected events N follows

$$\frac{\mathrm{d}N}{\mathrm{d}\sin^2\theta} = \frac{\mathrm{d}N}{\mathrm{d}\cos^2\theta} = \text{const.} \tag{3.9}$$

The trigger efficiency of the array remains over 99% up to a zenith angle of 60°. The median of the angular distribution in this zenith range lies at 38°. Hence, we chose this as the reference angle θ_{ref}, which defines the signal S_{38}.

Assuming the attenuation depends only[9] on θ, we can exploit the relation in Eq. (3.9). This is called Constant Intensity Cut (CIC) method [P:41]. Then, S_{38} is defined as

$$S_{38} = \frac{S_{1000}}{f_{\text{CIC}}(\theta)} = \frac{S_{1000}}{1 + ax + bx^2 + cx^3}, \tag{3.10}$$

where a, b, and c are fit parameters and $x = \sin^2\theta - \sin^2\theta_{\text{ref}}$ (see Fig. 3.7). In the past, a, b,

[8] Stations that triggered before the shower front reached the core position and stations that triggered after it was reached.

[9] Only recently, an energy dependence has been introduced into this parameterization.

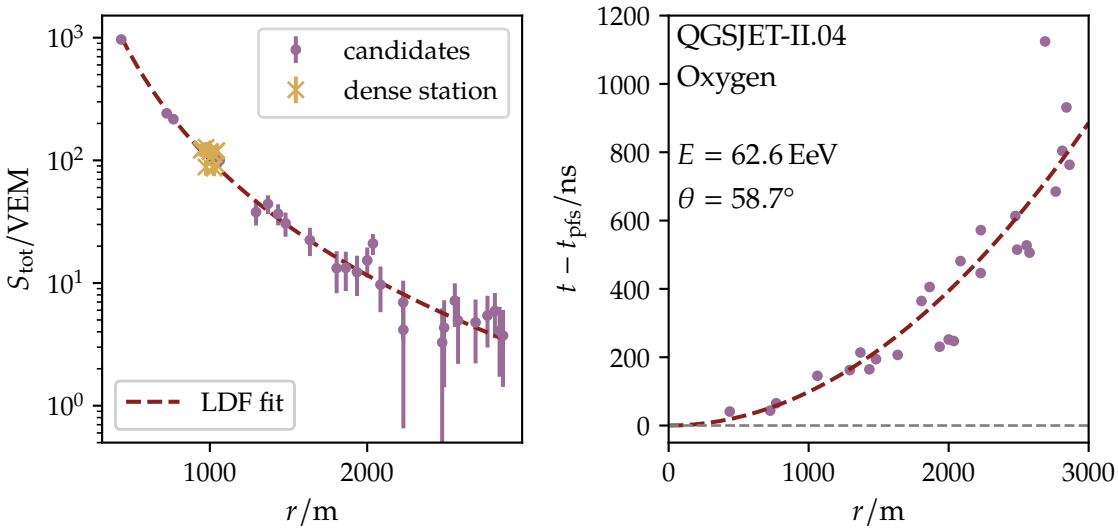


Figure 3.6: *Left*: LDF fit (black dashed line, see Eq. (3.8)) of total signal S_{tot} of candidate stations (magenta points, see Sec. 3.2.2.D) for a simulated oxygen-induced shower using the QGSJetII-04 model of hadronic interactions. For the simulation, we added 10 dense stations (orange points) at a perpendicular distance of 1000 m to the (true MC) shower core. The signal of these (off-grid) stations matches the expected signal given by the LDF fit at r = 1000 m. *Right*: Difference (magenta points) of trigger time t of a candidate station and the time it would take a planar shower front t_{psf} to reach that station. We have shifted the differences that it is zero at r_c and have fitted a quadratic function to the points (black dashed line). This shows the curvature of the shower front.

and c have been derived via a counting method [A:5] exploiting Eq. (3.9). Presently, a more advanced method based on Poisson χ^2 minimization is used [A:6]

Offline enables us to add SD stations, which are not part of the SD grid, during an air shower simulation. We call these off-grid stations dense stations. The position of a dense station is defined by its distance to the (true MC) shower core and an azimuth angle using the impact point of the shower core on the ground as the origin. The dense stations are still placed on the ground level. Among other things, the dense stations are used to estimate the deviations in the shower size. For the 1500 m-array, they are positioned at a distance of 1000 m for a set of evenly spaced azimuth angles. Thus, these stations are arranged on the unique ellipse at ground level, whose main axis lies on the projection of the shower axis to the ground.

3.2.3 SD energy calibration via *Golden Hybrid* events

The basic idea behind the SD energy measurement is the construction of the zenith-independent standard signal S_{38}, which represents the effective shower size at 38°. It combines a signal measured at a standardized distance, which is optimized to the geometry of the detector, a correction based on the atmospheric column, which has been traversed by the shower, and the zenith-dependent response of the employed detector. Hence, following Sec. 3.2.2.D we now have an indicator for the energy of the shower. The energy is related to S_{38} via a power law, which reads

$$E_{SD} = A \left(\frac{S_{38}}{\text{VEM}} \right)^B , \tag{3.11}$$

where A and B are constants inferred from the calibration process (see Fig. 3.7).

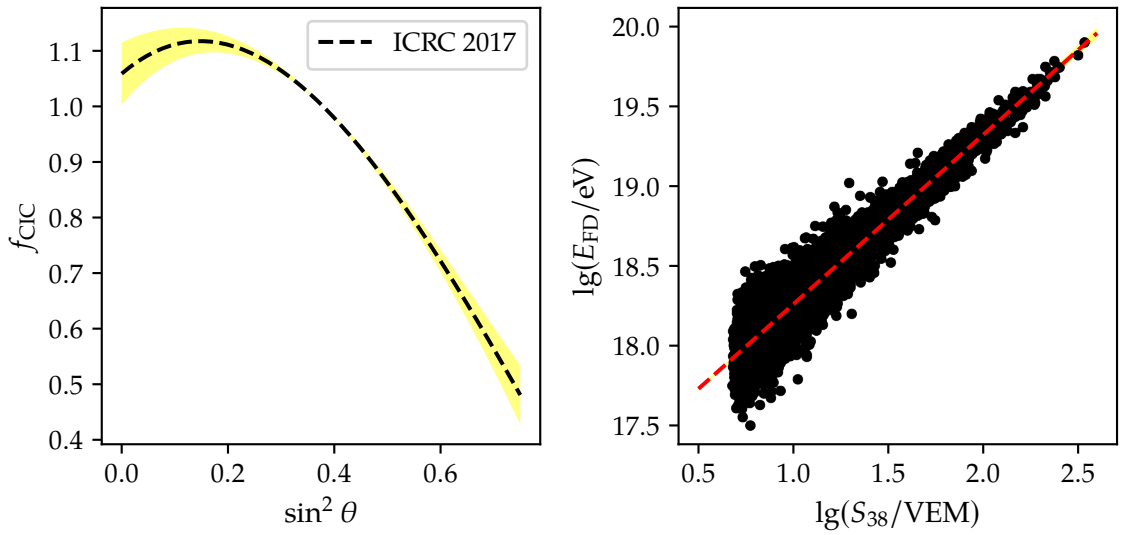


Figure 3.7: *Left*: CIC function f_{CIC} computed from Golden Hybrid data up to 2017 (see Eq. (3.10)). The yellow regions represent the 95% confidence bands of the f_{CIC} parameters. *Right*: FD energy vs S_{38} for Golden Hybrid events up to 2020 (see Eq. (3.11)).

For *Golden Hybrid* events, we can perform the SD and FD reconstruction independently of each other. Therefore, for each value of S_{38}, we have a corresponding value of E_{FD}, which is one of the main advantages of a hybrid detector design like that of Auger. Using the energy E_{FD} from *Golden Hybrid* events (see Sec. 5.2.2), we can fix the parameters A and B with a log-likelihood minimization. This log-likelihood accounts for various effects, such as the steeply decreasing spectrum, the shower-to-shower fluctuation, and the detector resolutions [P:88].

3.2.4 Reconstruction using SSD information

At the moment of writing, the procedure of using the SSD signals for a hybrid reconstruction is not fully fixed. Hence, we only will give general remarks.

Similarly, as for the WCD stations, we can construct an LDF using the total signals $S^{\mathrm{S}}_{\mathrm{tot}}$ of the SSD stations. This SSD LDF should exhibit a slightly different behavior than the LDF of the WCD, providing us with extra information. Using this together with S_{1000} potentially enables us an event-by-event mass separation for pure SD events [P:89, P:90] in addition to a better energy estimate.

Furthermore, having two measurements at the same position of the shower for two different types of detectors could also allow us to separate muonic from electromagnetic signals for each tank position. One – heavily investigated – method to do this is the so-called matrix formalism which assumes that the fluxes of the sub-components of the shower relate to the measured signals in both detectors via a simple matrix:

$$\begin{pmatrix} S^{\mathrm{W}}_{\mathrm{tot}} \\ S^{\mathrm{S}}_{\mathrm{tot}} \end{pmatrix} = \begin{pmatrix} \ddots & \\ & \ddots \end{pmatrix} \begin{pmatrix} \mathcal{F}_{\mu} \\ \mathcal{F}_{\mathrm{em}} \end{pmatrix}, \tag{3.12}$$

where $S^{\mathrm{W}}_{\mathrm{tot}}$ is the total signal in the WCD and $\mathcal{F}_{\mu}$ and $\mathcal{F}_{\mathrm{em}}$ are the muonic and electromagnetic fluxes [A:7], respectively. Inverting the matrix yields the relation between the signals in the WCD and SSD and the signals from the muonic and electromagnetic flux.

3.3 Reconstructions of mass sensitive parameters

A central topic in CR physics is the need for accurate knowledge of the primary particle masses (see Sec. 2.4) of air showers. These masses, however, are incredibly hard to reconstruct due to their close relation to the reconstructed energy of an air shower. When focusing on the total signal distribution, a proton-induced air shower can appear like a less energetic iron-induced air shower.

A workaround for this lies in the identification and reconstruction of mass-sensitive parameters. The most commonly used mass sensitive parameters are the depth of the shower maximum $X_{\max}$, which we have discussed in Sec. 2.2.4, and the relative muon content R_μ (see Sec. 2.4). We define the relative muon content by

$$R_\mu = \frac{N_\mu}{\left\langle N_\mu^{\mathrm{p}} \right\rangle}, \tag{3.13}$$

where N_μ is the number of muons at the ground and $\left\langle N_\mu^{\mathrm{p}} \right\rangle$ is the average number of muons produced in a proton shower with the same energy at the ground. For protons, Eq. (3.13) is by construction on average 1; for iron showers, R_μ is around 1.3 to 1.4.

Here, we introduce the most commonly used analytic approaches using non-NN-based methods to extract mass-sensitive parameters. Note that there are currently also data-driven analyses in this direction. We discuss them in Sec. 4.3 after introducing NNs.

3.3.1 Delta method: analysis of rise times

The shape of the early part of the time trace is related to the early particles arriving at the detectors. These early particles are dominated by muons since the particles of the electromagnetic shower component experience more scattering in the atmosphere. The spread of the muons depends mainly on the origin of their production.

A quantity that describes this is the so-called risetime $t_{1/2}$ of the WCD traces. It is defined as the duration the signal needs to reach from 10% to 50% of its total integral [P:91, P:92]. Similarly to R_μ, we relate $t_{1/2}$ to a benchmark value that quantifies its station-wise relative deviation. For each station i, we obtain

$$\Delta_i(r, \theta) = \frac{t_{1/2}^{(i)} - t_{\mathrm{b}}(r, \theta)}{\sigma_{t_{1/2}}(r, \theta)}, \tag{3.14}$$

where t_{b} is the expected benchmark rise time value and $\sigma_{t_{1/2}}$ its second-order moment, measuring the uncertainty. The average value of the Δ_i for an event depends on the underlying primary. Therefore, it can be used as a mass estimator [A:8]. This is called the Δ method.

Phenomenologically, the average value over the Δ_i can be also used to modify the relation in Eq. (2.12). To first order, the depth of the shower maximum is given by

$$X_{\max}^{\Delta} = a + b \left\langle \Delta \right\rangle + c \lg(E_{\mathrm{SD}}/\mathrm{eV}), \tag{3.15}$$

where a, b, and c are fit parameters and $\langle \Delta \rangle$ represents the average value of Δ_i for all stations. The fit parameters are determined by calibrating with the FD measurement of the depth of the shower maximum [P:93].

The benchmark relates the measurement with the muonic average. Therefore, it is problematic using simulations to get the parameters in Eq. (3.15) [P:92]. Due to the muon deficit, it would yield a shift, if compared to real data, that cannot be dealt with by means of calibration since there are no direct mass measurements. Furthermore, the correlation between the depth of the shower maximum, as defined in Sec. 2.2.3.A, and as defined in Eq. (3.15) is not unique due to its dependence on the muons. In spite of all these deficiencies, the Δ method remains a straight-forward way to estimate $X_{\max}$ from SD event data.

3.3.2 Air shower Universality

As addressed in Sec. 2.2.4, air showers induced by photons show a universal behavior [P:42, P:94]. For hadronic showers, this observation can be generalized in a phenomenological way since certain aspects, such as the energy spectra of the electromagnetic sub-component [P:95] or the universal shower profile (see Sec. 2.2.4), appear universal.

Air shower universality exploits this similarity by parameterizing the (main-contributing) shower sub-components, decoupling the components in the process [P:96]. We quickly reiterate (see Sec. 2.2.3) them here: For Universality, we differentiate between

- the muonic component μ,
- the electromagnetic component e$\gamma(\mu)$ from muon decay,
- the electromagnetic component from hadronic component e$\gamma(\pi)$,
- the electromagnetic component eγ from all other sources.

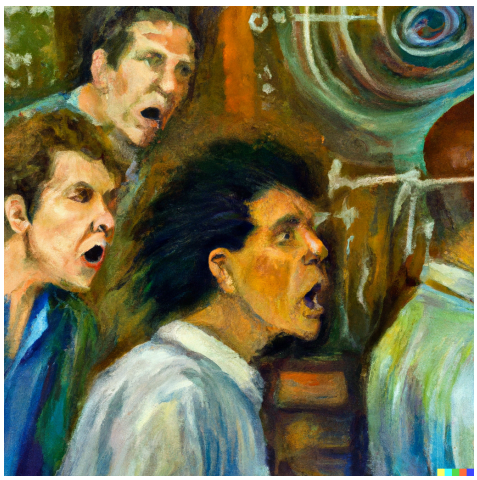
For each sub-component, a model of the expected signal to be deposited has to be developed. These models describe the signal for each sub-component measured in the SD stations. This signal model is accompanied by a time model, which essentially describes the expected trace shapes in the stations (see Sec. 3.2.2.C).

All of these parametrizations depend only on a subset of global shower parameters. This subset usually consists of the core position $\boldsymbol{r}_c$, the relative plane front time t_{pf}, the zenith angle θ, the azimuth angle ϕ, the depth of the shower maximum X_{max}, the energy of the shower E, and the relative muon content R_μ (see Eq. (3.13)). The reconstruction of X_{max} and R_μ to estimate the primary particle mass is the main objective of the Universality method. Moreover, Universality also acts as an independent estimator for the shower energy. Normally, this is done by a global log-likelihood minimization which accounts for the entire SD event.

At its, core Universality is an elaborate template-matching algorithm. By performing the Universality fit procedure, we try to find the appropriate shower that would explain the measured observables of this shower by finding appropriate total signals and signal shapes. This is the most significant advantage of the method since it is entirely analytic. Although due to its complexity, all of the parametrizations are derived from simulations, we are still able to scale parameters, such as R_μ, for which simulations do not match with measurements, recovering its predictive power.

In recent years, many different groups and people have [P:97, P:98, P:99, P:100] tried to tackle Universality. This popularity shows another advantage of Universality: it is (in principle) independent of the underlying detector type. Theoretically, it can be generalized for any combination of detectors opening a plethora of exciting applications. However, in each case, a new parametrization is needed. This yields complexity and has many nuances. Therefore, slight changes in the model could potentially have large effects on the outcome and predictions. In this work, we view the Universality reconstruction as the analytic counterpart of the NN-based approaches.

4 Using Artificial Neural Networks for the Reconstruction of Shower Properties



My CPU is a neural-net processor; a learning computer. The more contact I have with humans, the more I learn.

(Terminator 2: Judgment Day)

DALL·E 2 prompt:

Physicists shocked about a neural network that extracts information from physics data without human interference, oil painting[.]

ANNs are information processing algorithms modeled loosely similar to how the human brain and its neurons thought to process data. Their structure is – generally speaking – well represented by a (specially designed) directed graph. Henceforth, we refer to the topology of nodes and edges as the architecture of the network.

In the simplest implementation, each node in the graph represents a neuron that is able to weigh and pass information[1] received from neurons (or itself) to other neurons. The exact way how each of these neurons mixes and transfers the information is learned during a process called training. During training, information is propagated through the network. The response of the NN is then evaluated against some pre-defined metric, and the network is adjusted in such a way as to fulfill this metric. It has been shown that a special class of ANNs are universal approximators [C:2]. If they are large enough, they can approximate any analytical function with arbitrary precision. The huge advantage of this is that it is not necessarily required to know the form of the analytic function which belongs to the information-response pairs mentioned before. Therefore, we do not need to model the dependencies ourselves. This simplification also means that NN-driven algorithms do not have to be explicitly programmed. We assume that [C:2] is approximately true for any large enough NN. Due to the high complexity[2] and the training procedure, methods based on NNs belong to the field of computer science of machine learning.

In the last couple of years, the area of machine learning based on NNs has surged in popularity and usefulness[3]. An increase in affordable, parallel computing power, efficient training algorithms, and the general availability of data allowed the use of much larger NNs architectures called DNNs. The first well-known application of such DNNs has been in the

[1] Normally, encoded as floating point numbers.

[2] Let N be a part of a network that consists of a set A of n and a set B of m nodes. If we connect all nodes from A and B we would need $n \cdot m$ (initially) independent weights.

[3] Using GoogleScholar, the number of research papers containing 'deep learning' and 'physics' in their titles was around 7000 in 2015. Now, in 2022 it is about 33 000.

hard-to-solve task of image recognition [C:3]. A CNN [C:4], dubbed AlexNet, outperformed every model submitted to the ImageNet Challenge 2010 by over 10 percent points [T:G]. Only a few years later, a new network [C:5] surpassed human-level classification performance. Nowadays, many tasks humans reigned supreme are performed by DNNs in a much faster way. Even creative processes, usually hard to tackle with computer algorithms, are not safe anymore from these advances. DNN-based image generation from text prompts, such as DALL·E 2 [C:1], show this quite well.

DNNs have many different applications outside of the classification. They are also used in tasks such as

1. the extraction of novel features from input data,
2. the (fast) generation of new data samples,
3. the interpretation of situations based on audio-visual data,
4. and numerical regression.

These applications have various use cases in physics data analysis. The main focus, however, will be on the last point, which is especially interesting for the analysis of air showers. Using a NN-driven approach, we can relate complex properties of air showers, e.g., the shower footprint, to high-level observables, e.g., the energy or the depth of the shower maximum.

The normal research workflow of scientists is based on the utilization of analytic functions to model experimental results. This approach works fine when we have 'easy-to-exploit' dependencies or know all of the underlying physics. However, the data may contain complex correlations that are a priori unknown and very hard to uncover. Approaches based on DNN tackle this problem by working in a way that is driven by the data itself and solely guided by the pre-defined metric in the training process. This data-driven analysis, in turn, benefits the regular analysis since the NN result gives us a threshold of for how good the approach can become. For example, we could construct a data-driven form of the purely analytic Universality framework to check the parametrization of the used signal model (see Sec. 3.3.2).

In this work, we consider DNNs as a tool to analyze our physics data. In the case that established analytic approaches yield the same result as carefully[4] crafted DNN, it is reasonable to assume that the data has no correlations anymore that we can exploit. From our point of view, we obtain a threshold analysis and simultaneously a powerful predictor out of regular training.

In Sec. 4.1 we introduce conventions and handy notations to simplify the explanations in the rest of the chapter. Afterward, we discuss the basics of NNs in Sec. 4.2. We explain the training process, basic architectural concepts, and a couple of advanced network layers. Finally, we demonstrate how we apply NNs to classification tasks. All of this discussion gives us the tools to review the current main analysis done with NNs in Auger in Sec. 4.3. We show the basic ideas and architectures which we use and modify in the subsequent chapters. Finally, in Sec. 4.4, we discuss how we gauge the quality of our network predictions and in which way we compare the predictions of different NNs that predict the same shower observable.

[4] Unfortunately, this is hard to quantify.

4.1 Conventions and notation

In this thesis, we work with multidimensional arrays. These act as information mediators between different parts of NNs. Each variable x is an element of $\mathbb{R}^{n_1 \times \cdots \times n_M}$ where M is the number of array-like dimensions and the different n_i correspond to the sizes of the components. For example, the complete information of all time signals of the PMTs of N_s WCDs of one event (Sec. 3.2.2.C) could be represented by a multidimensional array x with $x \in \mathbb{R}^{N_s \times 768 \times 3}$, where N_s is the number of triggered stations.

To simplify our notation, we define a couple of operators which act on such these multidimensional arrays. Let $x \in \mathbb{R}^{n_1 \times \cdots \times n_M}$, we define $\mathcal{O}_D$ as

$$\mathcal{O}_D x = n_1 \times \cdots \times n_M \quad \text{and} \quad \mathcal{O}_{D,j} x = n_j. \tag{4.1}$$

Therefore, it retrieves the 'size' of x. Sometimes it is useful to know how many independent elements a multidimensional array has. We define the counting operator #

$$\#x = n_1 \cdots n_M. \tag{4.2}$$

Sometimes, instead of a single x we work with sets of differently sized multidimensional arrays, e.g., the trace information together with global shower properties, such as the zenith angle. In this case,we write $\{x^{(1)}, \ldots, x^{(n_t)}\}$, where n_t is the number of elements in the set. Henceforth, we omit the adjective multidimensional for the sake of brevity.

In this work, we focus only on supervised learning (see Sec. 4.2.1). We denote x (X) as the array (set of arrays) of the input, p (P) as the array (set of arrays) of the output, and y (Y) as the real values which correspond to p. We denote y (Y) as labels or targets for the inputs x (X). For example, x could represent the shower footprint and y the maximum of the shower depth. Normally, $\mathcal{O}_D y = \mathcal{O}_D p$, however there could also be a post-processing step that transforms p into the shape of y.

We differentiate between architectures $\mathcal{AR}$ and models $\mathcal{M}$. An architecture defines how a set of free parameters control the flow of information between the inputs x (X) and outputs p (P) and also in which way these parameters are adjusted during the training process. We differentiate between two types of such parameters: Hyperparameters Υ and regular model parameters[5] η. Hyperparameters are parameters that define the topology of a NN, the used inputs, and the training process. They are set before the regular training process. For example, the learning rate α (see Sec. 4.2.1) that defines how much the network parameters η are adjusted in each step of the training process is a typical hyperparameter. Regular parameters η are estimated at training time. They affect the propagation of information itself. If we compare this to regular fit models, the degree of a polynomial and the choice of minimization algorithms would be typical hyperparameters Υ, whereas their coefficients would be regular parameters η. Using this definition, we denote an architecture $\mathcal{AR}$ as a set of fixed hyperparameters. From an architecture $\mathcal{AR}$, we obtain a model $\mathcal{M}$ by fixing all of the parameters η via the training process. Consequentially, a single architecture can spawn many different models depending on how η has been fixed.

If we have a fixed sets of hyperparameters Υ and regular parameters η, we obtain the output P by using X as an input for $\mathcal{M}$

$$P = \left\{p^{(1)}, \ldots, p^{(N_\mathrm{o})}\right\} = \mathcal{M}\Big(X; \Upsilon, \eta\Big) = \mathcal{M}\Big(\left\{x^{(1)}, \ldots, x^{(N_\mathrm{i})}\right\}; \Upsilon, \eta\Big), \tag{4.3}$$

where X and P are sets of N_i and N_o arrays, respectively. Normally, we have N input-output pairs $X = \{X_i\}$ and $Y = \{Y_i\}$ for which we obtain a set of predicitons $P = \{P_i\}$. Henceforth, we use x, y, and p interchangeably with X, Y, and P.

[5] Commonly, in the field of statistics and machine learning the variable θ is used for the parameters. However, to avoid confusion with the zenith angle, we decided to use η instead.

Let $\mathcal{D}$ be a data set with N elements. We define it as

$$\mathcal{D} = \{(x_j, y_j)\} = \left\{\left(\{x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(N_\mathrm{i})}\}, \{y_j^{(1)}, \ldots\}\right)\right\}, \tag{4.4}$$

where $x_j^{(i)}$ are N_i inputs and $y_j^{(i)}$ are N_o different outputs (see Sec. 4.1). In Eq. (4.4) the inputs and outputs which share the same j belong to each other.

If the dimension of one of our arrays has the form

$$O_D x = n_b \times n_{d_1} \cdots \times n_{d_n} (\times n_t) \times n_c \tag{4.5}$$

we call n_b the batch dimension (see Sec. 4.2.1.A), n_{d_x} the spatial dimensions, n_t the time dimension, and n_c the channel dimension. For example, a mono-color video file would have two spatial dimensions, a single time dimension, and one channel dimension. The sizes of the spatial dimensions correspond to the width and height of the image. The time and channel dimensions correspond to the length of the video and the color channels, respectively. This definition is similar to that necessary to encode the SD time signals of the shower footprint. The batch dimension corresponds to the number of videos passed to our model simultaneously. Without loss of generality, we generalize Eq. (4.3) to be able to take an arbitrary number of inputs to accommodate the batch dimension.

4.2 Neural networks as tool for analyzing physics data

As already mentioned, NNs are essentially complex graphs through which information is passed via multidimensional arrays. The nodes of this graph are not restricted to simple operations. Each node of the graph could also be replaced by a new (sub-)graph exhibiting the same input and output edges (see Fig. 4.1).

We call a set of nodes that form a logical unit a network Layer. A common procedure is to stack many of these Layers to build a more complex architecture without the need to define every node by itself. Consequentially, if it is reasonable, we call logical units of multiple Layers sub-networks. Sub-networks perform a specific task that distinguishes them from other parts of the network. For example, in Chapter 7, we use a sub-network that is designed to extract features from encoded traces.

Essentially, the process from having enough raw data to a functioning NN model is as follows: First, we have to collect and pre-process the to-be-used data. The pre-processing step depends strongly on the problem and the chosen architecture. By doing this, we want to prevent, for example, extremely high values in the input data that could yield unexpected results and divergences. In this section, we assume that this step (see Sec. 5.3) is already performed for our inputs and outputs. Secondly, we have to construct a reasonable architecture. Most of the time, an NN architecture is constructed by inspecting previous networks that try to achieve similar goals adjusting or extending them for the specific problem. We take an existing architecture and fine-tune it based on our problem. Thirdly, we use the data to train the network via a suitable metric. Both, the training process and the choice of the metric are non-trivial. Lastly, we have to do a quality control of the resulting NN model. This quality control can yield an iterative process that starts again at the first step with slight modifications.

In this work, we use TensorFlow (TF) [T:H] to implement the NNs. TF is an open-source library providing all the necessary tools to build, train, and evaluate NNs with pre-defined inputs. It works by building directed graphs evaluated efficiently on Central Processing Units (CPUs) or Graphics Processing Units (GPUs). All (graph-)functions in the framework are automatically differentiable, which is highly beneficial for the training process (see Sec. 4.2.1). Due to its efficiency, we use the `Python` interface to build our NNs.

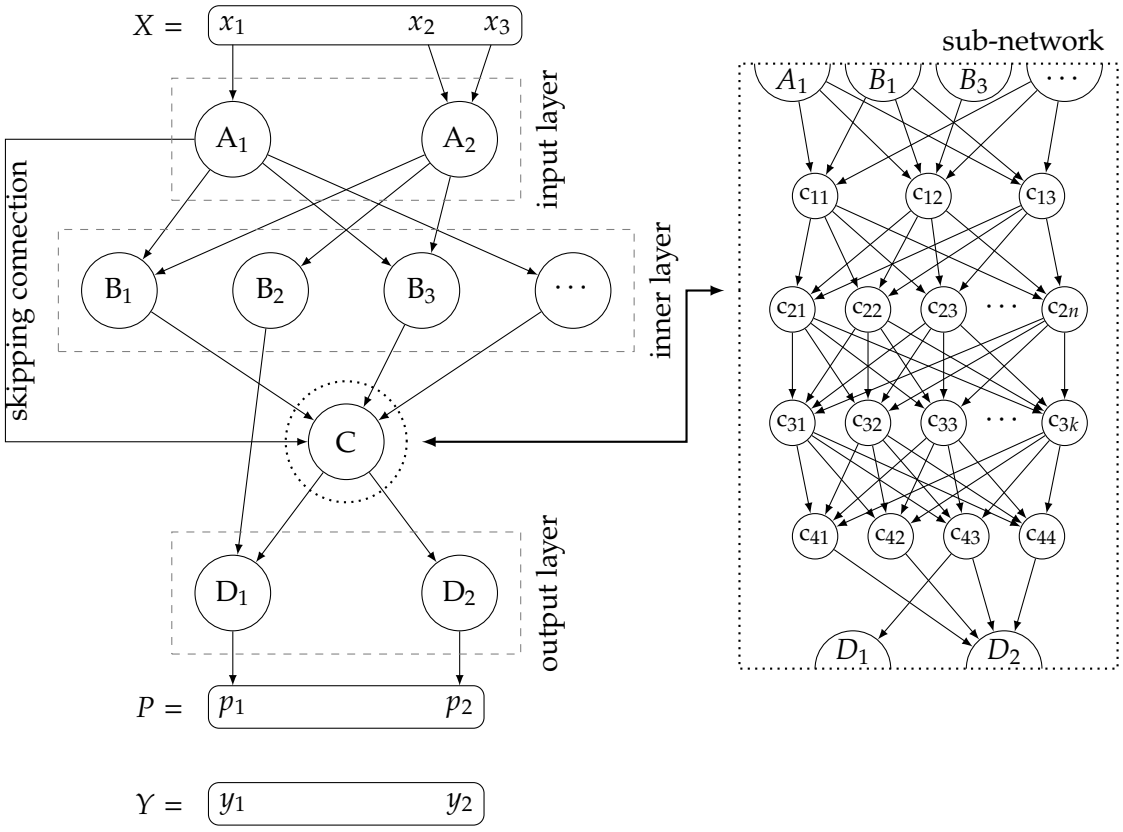


Figure 4.1: Illustration of the graph representation of a simple ANN showing some of the terminology described in Sec. 4.2.

4.2.1 Training procedure

We distinguish between different types of machine learning depending on how the model learns from data during the training process. In this work, we focus on the most common one called supervised learning [C:6]. During training, we supply the network model with a set of inputs and outputs. Henceforth, we refer to each pair as one data point[6]. For each data point, we obtain one prediction that depends on the choice[7] of the network parameters η. In the context of NN, this calculation is called a forward pass. Using a metric, we then evaluate if the prediction is close to the expected output and adjust the parameters η in such a way as to improve the prediction.

Hence, we optimize η by minimizing the metric. We obtain the optimal values of model parameters $\hat{\eta}$ via

$$\hat{\eta} = \underset{\eta}{\operatorname{argmin}}\, \mathcal{L}[\Upsilon](p, y; \eta) = \underset{\eta}{\operatorname{argmin}}\, \mathcal{L}[\Upsilon](\mathcal{M}(x; \eta), y; \eta), \tag{4.6}$$

where $\mathcal{L}$ is the cost functional. For a fixed set of Υ it generates the metric for our network training called *loss function*. Normally, the loss function is a convex function. For the sake of brevity, we drop the y on the hyperparameters Υ in Eq. (4.6) and move the dependecy of η into $\mathcal{M}$, leaving us with the following minimization problem

$$\hat{\eta} = \underset{\eta}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{M}(x; \eta), y). \tag{4.7}$$

[6]From the machine learning task of image recognition, this is also generally referred to as input X_j with the label y_j.

[7]To a certain degree the predictions also depend on the hyperparameters Υ.

Simulations of detector responses of CORSIKA shower using Offline (see Sec. 3.1) are a prime candidate for supervised learning since we have access to MC values. Beyond supervised learning, there is also unsupervised learning. Its prominent use cases are in data clustering, anomaly detection, and finding novel features. In each of these applications, there is usually no easy access to labeled data. Moreover, hybrid versions of training called semi-supervised learning combine supervised and unsupervised learning to some degree.

A Data preparation of inputs and outputs

Since NN models use a high number of free parameters, we have to prevent over-training. Over-training occurs when a NN does not learn correlations in a data set but memorizes it instead. Then, although it performs well on the previously seen data, it cannot generalize. Therefore, we need methods to check the prediction of the NN models during and after the training procedure (see Eq. (4.7)) and ensure that they do not deviate from their "ought-to-be" path. One way of doing this is by splitting the data set $\mathcal{D}$ into three non-intersecting parts. First, we require a data set used for tests after the training itself. It acts a control group on which we do all of our cross-tests to evaluate the quality of the predictions of our networks. We call this part the test data set (TeDs). Secondly, we have the training data set (TrDs). As its name suggests, the TrDs is the part of our data we use to train the model itself. It is the only part of our main data set that is directly used to adjust the model parameters η. Finally, we have a validation data set (VaDs). This is either a part sampled[8] from the TrDs or a third independent data set. The VaDs is not used to evaluate the performance in the testing stage. It is only used during the training at specific points to see if the model is still able to generalize outside of the TrDs. Over-training is detected by tracking the performance of the TrDs and VaDs. If the loss of the VaDs plateaus or strongly increases while that of the training decreases, it is very likely that the network is over-training. We denote the number of elements in each of the three data sets as N_{te}, N_{tr}, and N_{va}. If the VaDs is drawn from the TrDs, we will give the validation fraction f_{va}[9] called validation split. In the following, we use the subscripts tr, va, and te if we refer to any of the subsets.

Since data sets for the training of NNs are usually enormous, it is convenient to sub-divide our base data sets used as input in the neural network. This reduces the computational cost dramatically by enabling parallelization. We call these subsets batches. The number of samples $N_{B,b}$ in each batch b are not necessarily constant. Let D_B the function[10] that draws in each training step $N_{B,b}$ batches from $\mathcal{D}$ We define

$$\left\{(x_j, y_j)\right\}_b = \{D_B(\mathcal{D}; b)_i\}_b = D_B(\mathcal{D}; b) = D_B\left(\{(x_i, y_i)\}\,; b\right). \tag{4.8}$$

The sequence of data points (x_i, y_i) taken for each batch depends on the implementation. During training, it is common to randomly draw a constant number of elements without replacement. When this process exhausts the data set or after a pre-defined number of batches, we restart the process. We call this an epoch[11]. Batches between epochs do not need to be necessarily the same. Commonly, the end of an epoch marks the point at which we process the VaDs to check the training process.

B Stochastic gradient descent and weight updates

For NNs, direct minimization is unfeasible. There are no efficient, stable algorithms to account for all of their parameters. However, due to their design and implementation, they are

[8] Note that these data points are then not used for the training.

[9] $N_{\text{va}} = f_{\text{va}} N_{\text{tr}}, N_{\text{tr}} \to (1 - f_{\text{va}}) N_{\text{tr}}$

[10] In many implementations a `generator function` D_B provides these batches to the NN framework.

[11] Actually, we should modify Eq. (4.8) by the epoch number e as another input for D_B. However, out of convenience we neglect it.

differentiable (see Sec. 4.2). Hence, the standard approach is an iterative process based on the gradient descent algorithm [C:7] and modifications thereof. In each optimization step i, we compute the direction of the steepest descent of the loss function via a first-order Taylor approximation with respect to the current values of η_i. The most basic update step for one data point can be written as (see Eq. (4.8))

$$\eta_{i+1} \to \eta_i - \alpha \left.\frac{\partial}{\partial \eta} \mathcal{L}(\mathcal{M}(x_i;\eta), y_i)\right|_{\eta=\eta_i}, \tag{4.9}$$

where α is a hyperparameter called the learning rate. We have illustrated this process in Fig. 4.2. Although using Eq. (4.9) also works for larger NN, it would take an extraordinary amount of time to run an optimization on a single data point basis. To make this iteration faster by using parallelization, we can use an entire batch of training data (see Sec. 4.2.1.A) to make the update. This generalization is called Stochastic Gradient Descent (SGD). It generalizes Eq. (4.9) to

$$\eta_{i+1} \to \eta_i - \frac{\alpha}{N_b} \sum_{k=1}^{N_b} \left.\frac{\partial}{\partial \eta} \mathcal{L}(\mathcal{M}(x;\eta), y)\right|_{(x,y)=D_B(\mathcal{D};b)_k, \eta=\eta_i}. \tag{4.10}$$

Common problems of SGD[12] are overshoots due to a wrong choice of the learning rate α in very flat solution spaces, getting stuck because of a too little learning rate, and instabilities due to noise. We counteract these by modifying Eq. (4.9) slightly. Under the assumption that the training process is behaving well, we will come closer to a global minimum in each training step. Hence, it makes sense to reduce the learning rate the longer the training runs. This successive reduction is called learning rate decay. Learning rates can also be scaled to account for local minima or plateaus in the loss. This is referred to as adaptive learning rate. Moreover, to prevent sudden changes in training rate and gradient direction due to noise or special samples in the data set, we introduce a step-dependent momentum term. This term tracks previous learning steps to steer the training in the right direction. After adding all these ideas to Eq. (4.10), the update from i to $i+1$ reads

$$\eta_{i+1} \to \eta_i - \beta_i - \frac{\alpha_i}{N_b} \sum_{k=1}^{N_b} \left.\frac{\partial}{\partial \eta} \mathcal{L}(\mathcal{M}(x;\eta), y)\right|_{(x,y)=D_B(\mathcal{D};b)_k, \eta=\eta_i}, \tag{4.11}$$

where α_i is the step-dependent rate of learning and β_i is step-dependent momentum. The exact implementation of α_i and β_i depends on the algorithm. The very commonly used algorithm Adaptive Moment Estimation (Adam) [C:8] can combine all ansatzes discussed and, additionally, adds terms that prevent the collapse of the network training in the start phase. In this work, we use Adam for all network training processes if not mentioned otherwise. There are also even more advanced algorithms based on exponential decay and specialized momentum terms [C:9, C:10].

C Backpropagation

The gradient calculation goes from the output to the input of the neural network, exactly the other way around as the forward pass (see Sec. 4.2.1). Hence, it is called backward pass through our model $\mathcal{M}$. Since NNs are directed graphs the gradient calculation is straightforward. During the forward pass, we obtain for the outgoing edge k of node N

$$y_k = N(\{x_k\}; \{\eta_1, \ldots\}), \tag{4.12}$$

[12] This is also true for the regular gradient descent algorithm.

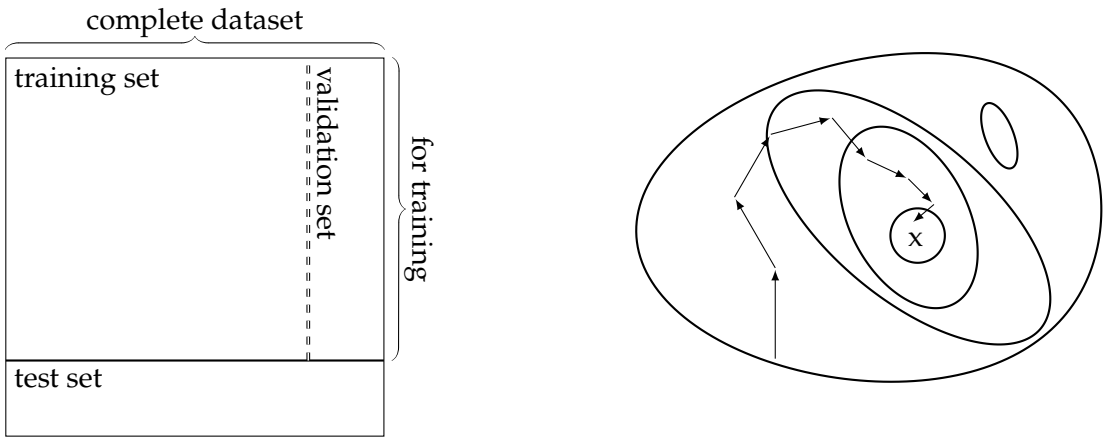


Figure 4.2: *Left*: Illustration of the subdividing of the data set. In the first step, we split the data set into one that is for the training procedure and one that is completely separated from it (thick solid line). For the training, it is common to subdivide the used data further (double dashed line). *Right*: Illustration of the gradient descent algorithm. The cross marks a local minimum in the "solution space". The black lines are contour lines for the loss function. The arrows indicate the steps in the gradient descent algorithm (see Sec. 4.2.1.B). In this particular case, the step size is reduced after each step, as in Eq. (4.11).

where x_k are the outputs of previous nodes that are required to produce the output y_k from the local model parameters $\{\eta_1, \ldots\}$. For each node we can find a path from the loss function by following edges against their direction. If there would be a path for going from the loss to node i via the nodes $i+2$ and i, we obtain

$$\partial_{\eta^{(i)}}\mathcal{L} = \partial_{\eta^{(i)}}N^{(i+2)} = \partial_{N_1^{(i+1)}}N^{(i+2)}\partial_{\eta^{(i)}}N^{(i+1)} = \partial_{N_1^{(i+1)}}N^{(i+2)}\partial_{N^{(i)}}N^{(i+1)}\partial_{\eta^{(i)}}N^{(i)}. \tag{4.13}$$

The algorithm which stores the function calls of the forward pass and the first-order derivatives of the backward pass is called backpropagation. It is a cost efficient way of computing the gradient of the loss term in Sec. 4.2.1.B.

All of these weight update algorithms assume that the initial weights have already been set. However, there is no easy way to find an ideal set of starting values. Commonly, the initial weights are randomly set using specialized distributions for each layer, such as the Glorot uniform distribution, which depends on the number of available weights [C:11]. Finding a "good" distribution for certain layers is still part of active research.

D Loss functions

For regression problems, we usually use the Mean Squared Error (MSE) as a loss function since it minimizes the bias and variance of Δy and is computationally inexpensive. It reads

$$\mathcal{L}_{\text{mse}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left(\mathcal{M}(x_i;\eta) - y_i\right)^2. \tag{4.14}$$

However, the MSE also is prone to overweight outliers. A natural extension of the MSE loss lies in the assumption that Δy follows a Gaussian distribution. By adding another output to our network, we transform Eq. (4.14) into

$$\mathcal{L}_{\text{e-mse}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left[\frac{\left(\mathcal{M}_1(x_i;\eta) - y_i\right)^2}{2\mathcal{M}_2(x_i;\eta)^2} - \ln M_2(x_i;\eta)\right], \tag{4.15}$$

where $\mathcal{M}_1$ and $\mathcal{M}_2$ are two NN outputs. We interpreted $\mathcal{M}_2$ as the σ of the prediction. Therefore, the network has the possibility to reduce the effect of hard-to-predict values and gives us feedback on this.

Another advantage of the MSE loss in Eq. (4.14) is that it is easily expandable for factorizable and unevenly distributed problems. Sometimes, it is helpful to weight our data to counteract the effects of unequal distributions where the training could ignore parts with insufficient amounts of data. Moreover, sometimes we are able to categorize our data into multiple different kinds of classes. Modifying Eq. (4.14) accordingly, we get

$$\mathcal{L}_{\text{m-mse}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{1}{\sum_j^{N_b} \delta(x_j \in C_i)\lambda(x_j)} \sum_j^{N_b} \delta(x_j \in C_i)\lambda(x_j) \left(\mathcal{M}(x_j;\eta) - y_j\right)^2 , \tag{4.16}$$

where $\delta(x_j \in C_i)$ is one if x_j is part of the class i and zero otherwise, $\lambda(x_j)$ is a weight for point j, and N_c the number of classes. In CR simulations we use the primaries as natural classes in later chapters.

Since we are especially interested in a mass-independent predictor, we can also add a term that penalizes non-zero means of the classes:

$$\mathcal{L}_{\text{m-mse+extra}} = \mathcal{L}_{\text{m-mse}} + \lambda_B \sum_i^{N_c} \sum_j^{N_b} \left\langle \mathcal{M}(x_j;\eta) - y_j \right\rangle^2_{j \in C_i} , \tag{4.17}$$

where λ_B is a hyperparameter to adjust the importance of the extra term. The average goes over all j belonging to class C_i.

E Early stopping and automatic decrease of learning rate

Wrong choices for the learning rate yield instabilities in the training process. For example, if we use a too-high learning rate, we are likely to jump over local minima. To counteract this, we modify the training on-the-fly. If, after a pre-defined number of epochs, the prediction of the validation set did not improve anymore, we stop the training directly without waiting to finish the remaining epochs. We use the weights from the epoch that gave us the best value on the validation set. Before this forced stop of the training, we try to recover the training process by reducing the learning rate by a pre-defined factor. In most cases, the network is then able to find a better local minimum.

4.2.2 Important architectural concepts

In physics problems, we usually exploit symmetries and correlations to simplify our analyses. To obtain a working model predicting physics data, we have to reflect these symmetries and correlations in the architecture. Finding a suitable architecture is, therefore, not trivial. Often we have to repeat this process for each individual task we want to tackle. To simplify this process, we regularly use "atomic" building blocks of layers that connect different parts of the network. Here, we will shortly take a view over the most common of these and other important parts of a NN.

In general, a NN architecture acts as a funnel for "important" information. As a rule of thumb, in each successive layer the amount of passing information is slightly reduced. In the best-case scenario, this leaves only a couple of final nodes that are then combined to reach a final prediction.

A Activation functions

Many of the standard layers provide only linear transformations between their input and output. If we would mindlessly stack these layers, we would end up with a giant, overly complex linear model. Such a model could certainly be collapsed into a much simpler version, and it would make it impossible to catch non-linear correlations in the base data.

To circumvent this, we add a source of non-linearity to each of the layers. This can be achieved by different means[13]. The easiest way to do this, however, is by introducing non-linear functions called activation functions. These are added to each layer separately, being applied on each output value of the layer. Activation functions directly influence the back-propagation process (see Eq. (4.13)) by adding an additional term to the derivative chain. Choosing an unsuitable activation function results in bad convergence rates, instabilities, and a slow inference process. For example, a well-known problem for activation functions, such as the sigmoid or tanh function, is the vanishing gradient problem [C:12, C:13]. The derivative of these functions for high and low values is close to zero. Since the weight updates directly scale to the derivatives (see Eq. (4.13)), the updates of the η become suppressed. This slows down the training or potentially even stops it.

The most common activation function which solves this basic problem is the Rectified Linear Unit (ReLU). We have depicted it in Fig. 4.3. It is a linear function that is zero for negative values[14]. Its non-linear behavior is solely due to the jump of its derivative from one to zero at $x = 0$. Because of this, ReLUs are computationally inexpensive. A problem with the basic ReLU is that it cuts information partially away, which could yield instabilities zeroing all of the data. Therefore, most generalizations of ReLU try to soften the hard cutoff at the origin to prevent this "dying" ReLU problem [C:14].

Notable generalizations that tackle these problems are the Parametric Rectified Linear Unit (PReLU) [C:5] and the Scaled Exponential Linear Unit (SELU) [C:15]. The former just adds a negative ReLU to ReLU scaled by a trainable factor. This factor is usually chosen to be very small. However, it is enough to reduce the risk of information being blocked in some nodes. The SELU replaces the zero in the negative half-plane with a scaled exponential that saturates to some constant negative value for very low x. It also scales the slope of the straight line in the positive half-plane. These changes are made in such a way that a zero mean and unit variance is achieved in each layer which regularizes the training process.

Another disadvantage of the ReLU activation function lies in its exacerbation of the irreproducibility of models and results [T:I]. Mainly, this is due to the jump in the function. This problem can be tackled by a Smooth Rectified Linear Unit (SMELU). The SMELU connects the zero value in the negative half-plane with the linear function in the positive half-plane by a quadratic function. This quadratic function is chosen in such a way that its minimum is at the left transition point and its slope is one at the right transition point. We derived the functional form in Appendix A.4. We obtain

$$A_{\text{SmeLu}}(x;\beta) = \begin{cases} 0 & x < -\beta \\ x & x > \beta \\ (x+\beta)^2/(4\beta) & \text{other} \end{cases} . \tag{4.18}$$

Since no complex functions are used, and its derivative is known, it is extremely fast to calculate

We mainly rely on the standard ReLU in the employed networks. However, at some point, we cross-check our results with more complex activation functions.

[13] For example, we could build networks that use higher moments of inputs or other functional representations. This method is similar to a kernel approach.

[14] Therefore, the ReLU is essentially a hard high-pass filter.

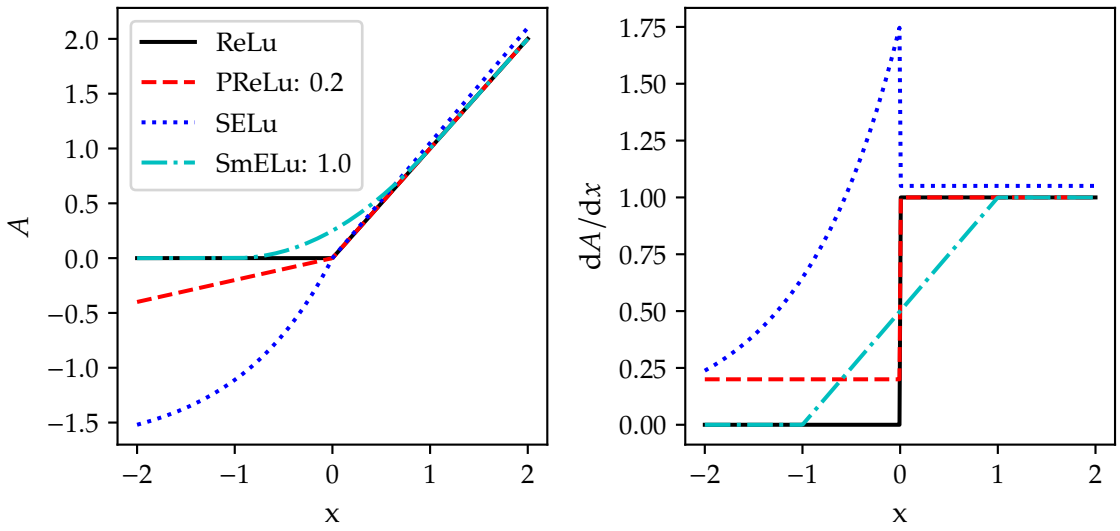


Figure 4.3: Visual representation of common activation functions and their derivatives. All extension trying to circumvent the dying ReLU problem by allowing for non-zero values or a smooth transition between the linear function and the zero in the negative half-plane.

B Skipping connections

The update steps of Eq. (4.11) are proportional to the derivative of all activation functions and weights between the loss function and a node (see Eq. (4.13)). If a network is extremely deep, it is possible that some of its parameters can not be properly updated anymore. To stabilize the training process, we add shortcuts, called skipping connections, to our architecture. These connect "earlier"[15] parts to "later" parts of the network, skipping layers in between. Such a shortcut enables the network to decide during training if the skipped layers are required for the inference process.

Most commonly, such shortcuts are realized in two ways. The output of a layer could just be added to the output of a later layer. This is called residual connections [C:16]. Another idea is to concatenate the output of a layer to the output of a later layer. In this way different levels of abstraction can be analyzed together in later parts of the network. Sub-networks using this scheme are referred to as densely connected [C:17].

4.2.3 Basic building blocks in neural networks

Since we use TF to build and train the networks, we do not have to implement any of the standard layers. They are already part of the software framework and are already highly optimized for parallel execution. In the following, we discuss all layers used in the network architectures in this work.

A Fully connected layer

One of the simplest layer types used commonly in NNs are fully connected layers, called a dense layer. In a dense layer with m nodes, all inputs are connected to all of the m nodes. Then the inputs are weighted linearly, resulting in m outputs. It can be represented by a

[15] We refer here to parts nearer to the input.

matrix multiplication

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1n} \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{m1} & \eta_{m2} & \cdots & \eta_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} \eta_{b1} \\ \eta_{b2} \\ \vdots \\ \eta_{bm} \end{pmatrix}, \tag{4.19}$$

where η_{ij} are trainable weights, and η_{bi} is a trainable constant vector referred to as bias. We did not add an activation function for visualization purposes.

Since a dense layer connects all inputs with all outputs, theoretically, it should be able to catch all possible correlations if we make it large enough. It provides the network training process with the freedom to decide which of these is most important to make accurate predictions. This, however, comes with a cost. The standard dense layer represented in Eq. (4.19) has $nm+m$ free parameters. Therefore, it has a quadratic $O(nm)$ memory consumption. This can go out of hand rather quickly. Also, a high number of trainable weights also translates to a more challenging training process which potentially can be dominated by noise. A visual representation of a dense layer is found on the right side in Fig. 4.1.

B Convolution layer

If the inputs are a priori uncorrelated, dense layers are a good choice. However, if our data consists of time series or images, which potentially exhibit local correlations, we can exploit this by using a different approach. In this case, convolution layers are more suitable. They work like regular discrete convolutions put in parallel whose set of filters f are model parameters fixed in the training process.

Let $O_D x = N_b \times \cdots \times K$ and $O_D y = N_b \times \cdots \times L$ (see Eq. (4.1)). Then, the convolution operation is given by

$$y_l = \left(x * f_l\right)(\boldsymbol{u}) = \sum_{\boldsymbol{v}\in\mathbb{Z}^n} \sum_{k=1}^{K} x_k(\boldsymbol{v})\, f_l^k(\boldsymbol{u}-\boldsymbol{v}), \tag{4.20}$$

where $\boldsymbol{u}$ ($\boldsymbol{v}$) is an n-dimensional integer index and f_l^k the filter corresponding to input k. Therefore, for each individual filter i, there is a separate filter k for the input k. We assume that all filters have the same dimensions. Therefore, we have $KL(\#f)$ free parameters. These are much fewer free parameters than a dense layer would have. At the borders of our inputs, the behavior of the convolution operation is defined by the user in a similar way as in all common implementations of the fast Fourier transform. In regular-mode the convolutions are only performed at positions where the entire filter lies inside of x. In the same-mode, zeros are padded in such a way to the input x that there is a valid filter position for each original position. In this case, x and y differ only in the channel dimension. In Fig. 4.4, we depicted the convolution operation in the one-dimensional case for two steps.

Sometimes, it is beneficial to further reduce the number of free parameters by using separable convolution layers. These use lower-dimensional filters to construct a higher-dimensional filter by using the Kronecker product. In two dimensions this is similar to doing a matrix multiplication with a transposed n-dimensional vector and another n-dimensional vector yielding an $n \times n$ matrix. In this way, we save $n^2 - 2n$ free parameters per filter.

Far away from the border regions, convolutions are translation invariant. Let Λ_L be the shifting operator. Without loss of generality, we denote for a one-dimensional filter with

[15] If there is a multidimensional input the behavior depends on the framework. In TF this is only true for trailing dimensions.

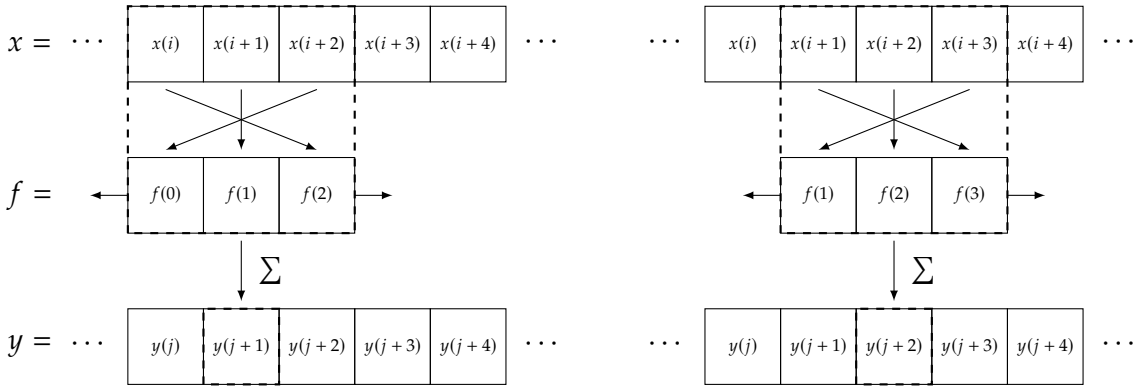


Figure 4.4: Representation of the operation defined in Eq. (4.20) for the one-dimensional case and a filter of size 3. The filter is moved step-by-step over the input combining 3 successive inputs.

$K = 1$

$$\Lambda_L y_l = (x * f^l)(u + s) = \sum_{v \in \mathbb{Z}} x(v) f^l(u + s - v) \tag{4.21}$$

$$= \sum_{v' \in \mathbb{Z}} x(v' + s) f^l(u - v') = ([\Lambda_L x] * f^l)(u). \tag{4.22}$$

Therefore, a filter can identify patterns on the whole available space.

Instead of running the convolution over the entire input, it is also possible to use it only on a subset. Usually[16], this is done by defining a stride length l_s. Instead of performing the operation defined in Eq. (4.20) for all valid integer indices $\boldsymbol{u}$ we increment[17] $\boldsymbol{u}$ by the stride length. This directly reduces the size of the output by the stride length: $O_D y = N_b \times n/l_s \cdots \times L$.

We can compute the size of the output after applying the filter via

$$O_{D,j} y = 1 + \frac{O_{D,j} x - K_j + 2l_p}{l_{s,j}}, \tag{4.23}$$

where j is the spatial or temporal dimension the filter is used on and l_p is a padding size. This padding can be used to allow for the filter to be used also on non-valid points at the corners of the input.

C Dense convolution layer

We can combine the concept of succeeding convolution layers (see Sec. 4.2.3.B) and densely connected shortcuts (see Sec. 4.2.2.B). In this way, in each layer, all features extracted by all previous layers remain accessible by the next convolution. These kinds of shortcuts simplify the training and allow for larger NN architectures.

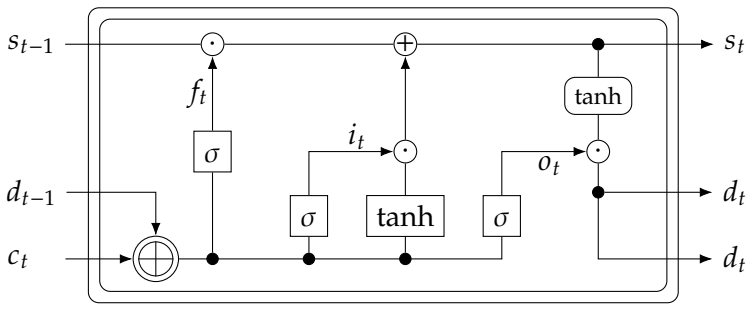
Commonly, dense convolution layers are built by applying a convolution that conserves the non-channel dimensions of the input. Afterward, the output of this layer is concatenated to the input and then used for the next iteration. Each time the number of channels increases by the filter size of the used convolution.

[16] Another common idea is to connect non-neighboring parts of the inputs by using dilated convolutions.
[17] Note that a stride length of 1 would be just a regular convolution.

D Long short-term memory layer

Until now, we have only discussed acyclic nodes and layers. Recurrent layers allow the output of a node to be used as another input for the same node affecting subsequent inputs. Like the convolution layers, they are tools to analyze sequential data, such as time signals. In this case, the internal state of the layer is updated for each time bin. This enables the layer to correlate a large number of bins which makes them superior in many direct applications since we do not have to standardize our input time series or specialize our architecture. Networks using a recurrent architecture are usually referred to as Recurrent Neural Networks (RNNs).

An advanced example of a recurrent layer is the Long Short-term Memory (LSTM) layer. The (internal) architecture of a LSTM resembles



(4.24)

where s_t is the t-th inner state, d_t the t-th output of the cell, called hidden state, c_t the t-th input bin, σ the sigmoid function, and tanh the tangens hyperbolicus. The tanh ensures that it is possible to add and delete memory. Part of the inner state is forgotten via the forget gate (f_t), and afterward, more information from the current bin is added via the input gate (i_t). We interpret the t as a time step. The inner state s_t acts as the memory of the LSTM cell. All of the rectangular σ and tanh nodes in Eq. (4.24) perform the following transformation

$$g_t = A_g(W_g d_{t-1} + U_g c_t + b_g), \tag{4.25}$$

where W_g and U_g are specific weight matrices, b_g is a bias, and A_g is either a σ or tanh. The σ-type are called gates. They act as funnels for the information content passed to other parts of the cell. For the new inner state s_t follows

$$s_t = f_t s_{t-1} + i_t \tanh(W_s d_{t-1} + U_s c_t + b_s). \tag{4.26}$$

Similarly, the new hidden state d_t is

$$d_t = o_t \tanh s_t. \tag{4.27}$$

An LSTM can return either only the hidden state of the last step or all of the hidden states building a new time series with the same length as that of c_t. The initial inner and hidden states of an LSTM are usually set to zero. However, we can set them to any arbitrary values.

Due to this complexity and the recursive nature of LSTMs, their execution is hard to parallelize. As a consequence, networks that use LSTMs are slower than equivalent networks which use convolutional layers. Still, they are potent tools that only have a small set of to-be-adjusted hyperparameters, which are applicable to inputs of any length. They are the core reason for improvements in specific machine-learning tasks, such as natural language processing [C:18, C:19] and human-level competence in complex games [C:20].

A regular LSTM goes only in one direction (from t to $t + 1$). To allow for the correlation between later and earlier bins, we can use bidirectional LSTMs. Basically, they are just two

LSTMs that go in different orders through the time series, one from t to $t+1$ and one from t to $t-1$. In this way, future bins can be correlated with past bins.

In this work, we will use only LSTM layers as recurrent layers in our networks. Hence, every time we refer to an RNN, it is based upon those.

E Other important layers

Due to the popularity of NN based models, there is a great number of standardized layers. Here, we quickly discuss a number of standard layers which are commonly used.

Dropout: To prevent over-fitting during training and enable the learning of different combinations of features it can be beneficial to randomly set inputs of other layers to zero. A layer that performs this task is called a dropout layer. It just sets a fraction x of its inputs to zero. In many implementations, such as in TF, the remaining inputs are then multiplied by a scale factor of $1/(1-x)$. This action prevents overtraining. During inference, this layer is deactivated.

Reshape: A reshape layer rearranges the dimensions of the input arrays. We have $O_D x = N_b \times A_1 \times \ldots \times A_n$ to $O_D y = N_b \times B_1 \times \ldots \times B_q$ where $\#x = \#y$. This is useful to prepare different inputs for advanced layers.

Flatten: A flattening layer collapses all non-batch dimensions. Hence, $O_D x = N_b \times N_1 \times \ldots N_d$ transforms to $O_D y = N_b \times N_1 \cdots N_d$. Usually, this kind of layer is one of the last layers in neural networks to funnel all values of the output array into a final dense layer.

Concatenate: Sometimes, the outputs from different parts of the network represent different kind of features that have to be connected but not yet mixed in any way. This is the task of the concatenation layer. For example, if we have two inputs $O_D x_1 = N_b \times N_1 \times \ldots \times N_d \times A$ and $O_D x_2 = N_b \times N_1 \times \ldots \times N_d \times B$ we obtain an output of the form $O_D y = N_b \times N_1 \times \ldots \times N_d \times (A+B)$.

Pooling: A pooling layer of size $(s_1, s_2, \ldots)$ reduces the size of its inputs by coupling neighboring values in one or multiple of its dimensions. In TF there are two modes for this: average and max. In the one-dimensional case, s neighboring values are taken and the operation of the mode is applied. Afterward, the next non-intersecting s neighboring values are processed. Looking at the input and output dimensions we have $O_D x = \ldots \times N_p$ and $O_D y = \ldots \times N_p/s$, respectively.

Lambda: A lambda layer is a common concept that allows users to create layers on the fly. It works like a lambda function where users can provide their own methods that are executed by the network. For example, it is very convenient to add singleton dimensions to inputs with a lambda layer. These singleton dimensions are sometimes needed to satisfy the shape conditions of layers, such as convolutional layers.

In Appendix B.1 we discuss additional non-standard layers used in the network discussed in Sec. 4.3.2.

4.2.4 Notes on classification

In Chapter 7, we use NN-based classifiers to identify events induced by different primary particles. Let N_c be the number of distinct classes, then the output of a classifier is, usually, N_c rational numbers. Each of these numbers indicates how likely the input corresponds to one of the classes. To obtain the final prediction, we have to define a decision function D_c, which estimates the most-likely primary particle from these N_c numbers. For example, a simple decision function would be the argmax function which returns the class exhibiting the largest value in the N_c NN output values. For a binary classification problem, it is enough that a classifier predicts a single scalar output. A common decision function, in this case, would be a step function at a pre-defined threshold. If the network output lies below the threshold, the input is classified as the first class, and if it lies above the threshold as the second class.

To evaluate the quality of our predictions, we use the accuracy $\mathcal{A}$. The accuracy is the fraction of correctly predicted classes after using the decision function D_c on the NN predictions. We define the accuracy $\mathcal{A}$ on the TeDs of a prediction via

$$\mathcal{A}(y, p) = \frac{\#(y \cap D_c(p))}{N_{\text{te}}} = \frac{1}{N_{\text{te}}} \sum_{i}^{N_{\text{te}}} \delta_{y_i p_i} \in [0, 1], \tag{4.28}$$

where δ is the Kronecker delta. For N_c classes a random predictor has an accuracy of $\mathcal{A} = 1/n_c$. A $\mathcal{A} < 0.5$ for a binary class implies that the chosen decision function should be inverted.

For classification, a powerful tool for visualization of the performance of our classifier is the confusion matrix C. We define it as follows

$$C = \overbrace{\begin{pmatrix} n_{1,1} & n_{1,2} & \cdots & n_{1,N_c} \\ n_{2,1} & n_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ n_{N_c,1} & \cdots & \cdots & n_{N_c,N_c} \end{pmatrix}}^{\text{classes}} \Bigg\} \text{predictions} \tag{4.29}$$

where $n_{a,b}$ represents the number of values belonging to class a while predicted to be in class b. In this matrix, the sum over the diagonal is the number of input and output pairs that are classified correctly. The lower left part (blue) and upper right part (red) contain the numbers of miss-classified values. For the binary classification case, Eq. (4.29) simplifies to

$$C_{\text{binary}} = \begin{pmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{pmatrix}, \tag{4.30}$$

where True Positive (TP) and True Negative (TN) are the number of correctly classified inputs and False Positive (FP) and False Negative (FN) are the number of wrong classifications.

4.3 Status of neural network based analysis in auger

There are two methodological different NN-based ansatzes that have been recently published by Auger [P:101, P:102]. Like the advanced conventional methods in Sec. 3.3, both focus on the reconstruction of shower properties, which are essential for estimating the primary particle mass. However, they differ in the scope of their inputs and used architectures. The first approach uses mainly localized information to predict a muon signal in on WCD station. The second one uses global shower-level information to reconstruct global shower properties. We define shower-level information as all MC and reconstructed properties that describe the shower and its propagation through the atmosphere. For example, the shower inclination angle θ is a shower-level property. Moreover, we define station-level shower properties as all quantities that depend on a particular position on ground level, such as the signal in the PMT of the WCD stations of the SD.

A caveat of both of these networks is their dependence on simulation data for the training process. Due to the mismatch between simulations and measurements, a direct application of the NNs on measurement data yields erroneous results. Therefore, proper corrections and cross-tests are required.

In Chapters 7 to 8, we use network architectures based on those found in both analyses [P:101, P:102]. Therefore, we provide, in this section, a short overview of both. Our primary focus, thereby, lies on the used network architectures.

4.3.1 Extraction of the local muon signal using SD detector stations

A shower event has a plethora of station-level information for each of the triggered SD stations. It is a valid question if and in which way we are able to exploit this information to reconstruct station-level shower properties, such as the signal produced by the muonic subcomponent in the WCD. Knowing it for each of the stations benefits the estimation of the primary particle mass since the muon content depends on the mass of the primary particle (see Sec. 3.3).

Historically, this ansatz started with the estimation of the total muon signal [A:9, A:10] which was followed up by mixing the predictions with results from the delta method [A:11]. From this point, the analysis evolved reconstructing the entire muon trace [P:101, A:12].

A Extraction of total muon signal

To extract the total muon signal S_μ, a NN purely made of dense layers (see Sec. 4.2.3.A) has been employed [A:9, A:10] which uses nine shower parameters as inputs: The decadic logarithm lg of the Monte Carlo energy of the shower

$$\mathcal{E}_{\mathrm{mc}} \equiv \lg\left(E_{\mathrm{mc}}/\mathrm{eV}\right), \tag{4.31}$$

the total signal S_{tot}, the distance to the shower axis r, the zenith angle θ, the angle between the to-the-ground projected shower-axis and the detector φ_{ad}, the rise time $t_{1/2}$, the fall time[18] t_f, the difference between start and end time t_{d} ($\propto (b_{\mathrm{e}} - b_{\mathrm{s}})$), and the ratio r_{p} of the maximum of the total trace and the total signal. We define r_{p} as

$$r_{\mathrm{p}} = \frac{S_{\mathrm{tot}}\left(b_{\mathrm{max}}\right)}{S_{\mathrm{tot}}}, \tag{4.32}$$

where b_{max} is the bin in which most signal was deposited in the detector. Thus, in total, two event- and seven station-level input parameters are utilized.

[18] Similar to the rise time. But in this case the time for the integrated signal to reach 50% to 90%.

In total, we have two shower-level and seven station-level input parameters. The difference in scales of all inputs was removed by standardization. For standardization we transform each variable x via

$$x \to \frac{x - \langle x \rangle}{\sigma_x} \tag{4.33}$$

setting the mean to zero and standard deviation one.

The base data set for this analysis has been comprised of the WCD station information from events simulated using the hadronic interaction model QGSJetII-04. Only stations have been kept that are non-saturated and have a total signal of above 10 VEM. Energies below the full efficiency of the SD have not been included. The zenith θ was kept below 45°.

An evolutionary algorithm [C:21] has been used as an optimization tool to find the best-suited architecture. The algorithm had the choice between layer widths, number of layers, and activation functions. This resulted in the architecture depicted in Fig. 4.5 [P:103]. As optimizer for the network training, Adam has been selected (see Sec. 4.2.1.B). The final network model reconstructed the muon signal with a relative error of below 10%. Moreover, it did not significantly lose performance when switching to simulations using the hadronic interaction model EPOS-LHC. Hence, the station-level approach is almost independent of the hadronic interaction model.

The final architecture can be classified as a densely connected, multilayer, feed-forward network (see Fig. 4.5). In the original work [A:9], the first layer of the network is incorrectly described. If we implemented it like stated there, we would reduce our statistics by a factor of two because of the ReLU functions directly after the input and the used standardization procedure. Therefore, we believe that either another nine-unit dense layer is in between the input or the 18-unit dense layer or that the input layer has just a linear activation function. The entire network has (without the 9-unit layer) 2002 free parameters. This is compared to image classifiers like ResNet [C:22] small.

B Extraction of the muon signal trace

The signal traces of the WCD might contain a rich set of hidden information that we do not exploit in standard analysis. Therefore, the next step to improve on the analysis described in Sec. 4.3.1.A lies in the inclusion of the signal traces. To accomplish this, a different architecture is required since we want to account for the local temporal correlations. Using LSTMs (see Sec. 4.2.3.D) as a core component of such an architecture (see Fig. 4.5) is a natural choice.

For this analysis, an advanced network has been designed in [P:103] (see Sec. 4.3.1.A). Instead of predicting the total muon signal, the NN predicts the parts of the muon signal trace. It uses the standardized secans of the zenith angle, the standardized shower distance, and 200 bins[19] of a trace as inputs [P:103] . The muon trace analysis uses a relaxed version of the cuts in Sec. 4.3.1.A. The zenith angle cut is at 60°, and the total signal cut is at 5 VEM.

Again, the prediction of the trained model does not strongly depend on the hadronic interaction model giving satisfactory results. Moreover, it could also be shown that lateral distributions of the muon sub-component and the electromagnetic sub-component match well with the predictions [P:103].

4.3.2 Footprint analyses using deep neural networks

To reconstruct event-level shower properties, it is reasonable to use all of the available data gained from the SD measurements. Similar to the traces of the SD stations, the correlations between station-level information of the shower footprint may contain hidden information previously not exploited by standard analysis. To test this, we can use NNs.

[19] From b_s on. If the trace is not long enough it is zero-padded.

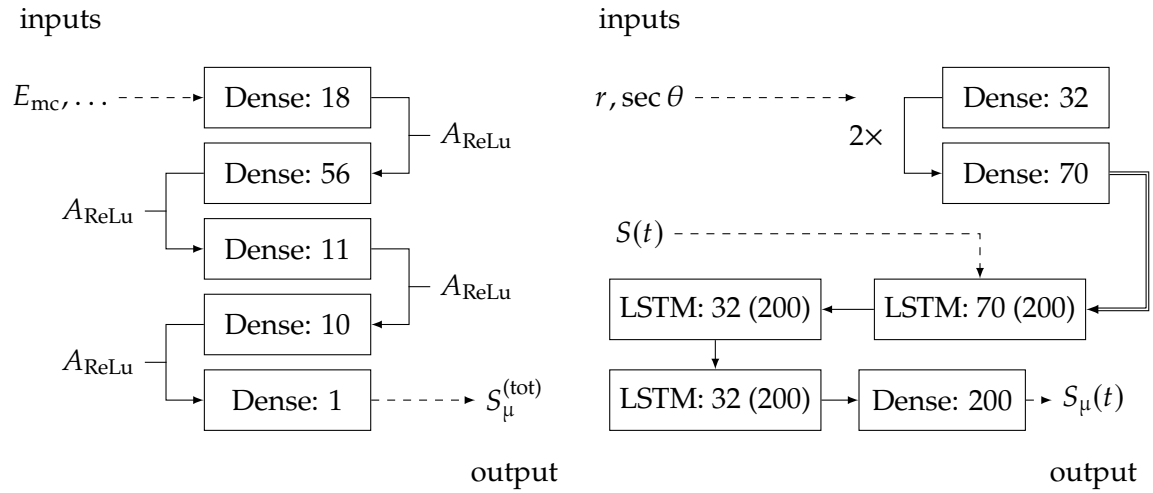


Figure 4.5: *Left*: Illustration of the architecture used for the estimation of the total muon signal $S_{\mu}^{(\mathrm{tot})}$ (see Sec. 4.3.1.A). The architecture consists only of stacked dense layers. The number after the colon indicates the number of nodes. *Right*: Illustration of the architecture used for the estimation of the muon signal trace $S_{\mu}(t)$ (see Sec. 4.3.1.B). The first number behind the colon for the LSTM layers represents the number of LSTM units, and the number in the parentheses is the size of the time dimension (see Sec. 4.2.3.D). Hence, we use the complete output of the LSTM. The output of the two dense layers is used as initial values for the inner and hidden states.

The Air Shower Extraction Network (AixNet) architecture was designed exactly for this use case [A:13, A:14]. Since we use NNs based on this architecture in Chapters 7 to 8, we discuss in this section the general architecture of AixNet and review the most recent changes in each of the end steps [P:102, P:104]. We have depicted AixNet in Fig. 4.6 The core design idea of the architecture is to separate the air shower reconstruction into three smaller networks with different scopes.

A Architecture of *Aixnet*

The first sub-network (N1) condenses the information of signal traces of length L_t into n_f features. We call it Trace Feature Extractor (TFE). Since the physics of the detecting traces is the same for each WCD station the same, the TFE uses the same weights for each trace individually. Due to this weight-sharing, the TFE is independent of the size of the encoded footprint. Hence, this sub-network extracts longitudinal shower information of the shower development from the shower footprint. In the current iteration of AixNet, the TFE is built from two sequential LSTM layers (see Sec. 4.2.3.D). One bidirectional that correlates past and future time steps of all PMT traces with 120 bins and one that takes the output to generate ten unique features.

In the next step, the trace features are concatenated to n_a additional, encoded station-level parameters creating a shower footprint feature map. This feature map is used as input in the second sub-network (N2). Henceforth, we denote it as the Spatial Correlation Analyzer (SCA). Its main objective is to correlate the spatial information of the footprint feature map. Therefore, it is the core part of the footprint analysis. Since the objective is similar to the analysis of images, a mix of densely connected (see Sec. 4.2.3.C) 2D-convolution layers (see Sec. 4.2.3.B) is used. Currently, the implementation uses group equivariant convolutions with hexagonal 3×3 convolutions (see Appendix B.1). In this way, the rotational symmetry of the triangular grid is directly encoded in the network architecture.

In the last step, the output of the SCA is used as input for a prediction unit. Henceforth,

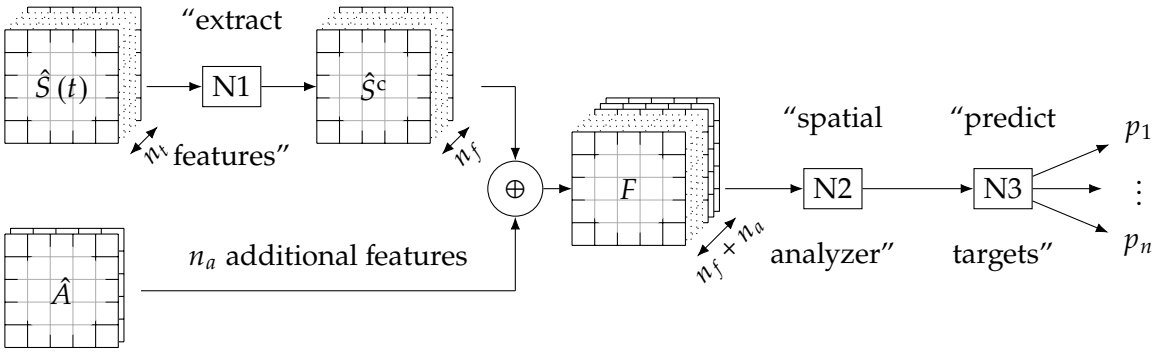


Figure 4.6: Illustratiuon of the AixNet architecture. The network can be separated into three different sub-networks marked by boxes. N1 is a trace feature extraction network. It extracts n_f features from encoded event traces of the length L_t. Each trace is treated the same way by using weight sharing. The novel features are then concatenated to n_a additional human-engineered features creating a feature map F. The feature maps F are used as input for the SCA. The SCA correlates the features in each of the bins and between different bins of the shower footprint encoding. It tries to find correlations between the station positions. Its output is then used as input in the last sub-network (N3). This last part can be one single or multiple prediction units that funnel the outputs SCA into dense layers. One of the main advantages of the architecture of AixNet is the modular design.

we refer to this as Feed-Forward Predictor (FFP) (N3). For each event-level target to-be-reconstructed, the FFP uses the same[20] sub-network architecture to funnel the output of the SCA into a dense layer. The number of units of this final dense layer depends on the target, e.g., for the shower energy, we need only one unit, and for the shower-axis vector, we need three. In the newest design [P:105], the funneling is realized by using residual shortcuts and pooling layers to reduce the spatial dimensions of the input before flattening it.

Due to its modular design, the base architecture of AixNet can be used to relate a shower variable to the shower footprints. This makes it easy to reuse the architecture for each scalar target we want to predict. We use an architecture based on AixNet in Chapter 7 and Chapter 8.

B Training and performance

The loss function used for the training of AixNet is a linear combination of the MSE loss functions of the predictors. The weighting was determined by studying the training process. For its X_{max} predictor a version of Eq. (4.16) has been used to reduce the inter-primary bias of the predictions [P:104]. The training data is comprised of reconstructed events of air shower simulations based on the hadronic interaction model EPOS-LHC.

For all targets that have an equivalent in the SD standard reconstruction, such as the shower energy and the zenith angle, the network performs at least as well as the standard reconstruction on simulation data [P:104]. Predicting on different hadronic interaction models the network has not been trained on results in similar precision but global biases. After correcting for differences between simulations and measurements the X_{max} prediction of the network shows similar behavior than the FD measurement [P:106].

[20] However, the weights for each of these parallel sub-networks are not the same.

4.4 Used evaluation metrics to compare NN predictions

After training a neural network, we have to estimate how well the predictions of the network model match the target values on the TeDs. We do this by evaluating the predictions on a fixed set of metrics. In this way, we are able to estimate the quality of the predictions and lay the groundwork for the comparison of different models which predict the same target.

4.4.1 Metrics for the comparison of predictions of NNs

We start by making basic definitions of statistical quantities following [T:J]. Let $\{x_1, \ldots, x_N\}$ be a set of N independent values drawn from an arbitrary distribution and x a random variable corresponding to this distribution. We denote $\langle x \rangle$ as the sample mean of the set

$$\langle x \rangle = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4.34}$$

and σ_x^2 as the sample variance

$$\sigma_x^2 = \mathrm{VAR}(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \langle x \rangle)^2. \tag{4.35}$$

Using the variance, we define the standard error SE of x as [T:J]

$$\mathrm{SE}(x) = \sqrt{\mathrm{VAR}(x)}. \tag{4.36}$$

In Appendix A.1 we dedicated a part of the section to less used quantities, such as the standard error of the standard deviation (see Eq. (4.36)) which we need for our analysis in Sec. 7.3.

To evaluate the quality of the predictions of NN we require a set of metrics that we use to compare the prediction p with the target value y. Usually, we compute these metrics in bins over other quantities, e.g., the MC energy E_{MC}. Since we want the prediction value p to be very close to the value of y, we define the difference

$$\Delta y = \langle y - p \rangle = \frac{1}{N} \sum_{i=1}^{N} \Delta_i y, \tag{4.37}$$

where $\Delta_i y = y_i - p_i$, to check for potential biases in the predictions of the NN. A small bias corresponds to a good accuracy. Since the width of the distribution of $\Delta_i y$ is a measure of the precision of the predictions, we utilized the unbiased standard deviation

$$\sigma_{\Delta y} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\Delta_i y - \Delta_y)^2} \tag{4.38}$$

of the set of differences $\Delta_i y$. We want that our NN predictions do not depend on the underlying primary. To estimate the primary-dependent bias, we use

$$\Delta_{\text{p-Fe}} y = \left| \left\langle \Delta_{\mathrm{Fe}} y \right\rangle - \left\langle \Delta_{\mathrm{p}} y \right\rangle \right|, \tag{4.39}$$

where $\Delta_{\mathrm{Fe}} y$ and $\Delta_{\mathrm{p}} y$ are the average differences for protons and iron (see Eq. (4.37)), respectively.

In later chapters the variable symbol y will be replaced by the symbol of the target. We use the naming convention that y^p is the name of symbol for the prediction.

There are many ways to assess the performance of a regression model. As an addition to the above performance metrics, we define the following three metrics to be used in Chapter 6. First, we use a modified version of the MSE. Since we compare muon signals, we modify Eq. (4.14) via

$$m_2 = \left\langle \left(S_\mu - S_\mu^{\mathrm{p}} \right)^2 / \mathrm{VEM}^2 \right\rangle = m_{\Delta S_\mu, 2} / \mathrm{VEM}^2 \tag{4.40}$$

to remove the units. We test the dependence on the training split and training procedure solely with this metric. However, because the muon signal covers multiple scales, its absolute value is hard to interpret. Therefore, we also have to define relative metrics. We define the relative error of the predicted muon signal as

$$m_R = \left\langle \frac{S_\mu - S_\mu^p}{\epsilon + S_\mu} \right\rangle, \tag{4.41}$$

where ϵ is a small constant that prevents instabilities due to very low muon signals. As final metric, we use an estimate for the precision of the S_μ predictions of the model. For this, we use the standard deviation σ of the difference between predicted and real muon signal value normalized to the averaged real muon signal

$$m_\sigma = \frac{\sigma_{\Delta S_\mu}}{\left\langle S_\mu \right\rangle}, \tag{4.42}$$

which we interpret as a measure for the resolution.

4.4.2 Methods of comparing different distributions

Let w and z be two sets of values which we have to compare. To obtain an estimate of the linear correlation between w and z, we use the Pearson correlation coefficient. We define it via

$$\rho_{\mathrm{P}}(w, z) = \frac{\langle (w - \langle w \rangle)(z - \langle z \rangle) \rangle}{\sigma_w \sigma_z}. \tag{4.43}$$

The figure of merit $\mathrm{mf}(w, z)$ defined as

$$\mathrm{mf}(w, z) = \frac{|\langle w \rangle - \langle z \rangle|}{\sqrt{\sigma_w^2 + \sigma_z^2}}. \tag{4.44}$$

is a quantity often used in Auger, which gives a rough idea about the separation of two distributions. For example, w and z could be the proton and iron subset of a mass separating variable x. The higher $\mathrm{mf}(x_{\mathrm{p}}, x_{\mathrm{Fe}})$ is, the better the separation between the masses. As a shorthand notation, we use $\mathrm{mf}_{\text{p-Fe}}(x)$.

We have to be careful when comparing figure of merits from different sources. A much higher value might suggest that we have a much better separation. However, this is not necessarily true (see Fig. 7.70). Especially if w and z follow non-gaussian distributions, we could increase and decrease the figure of merit without changing the separation at all.

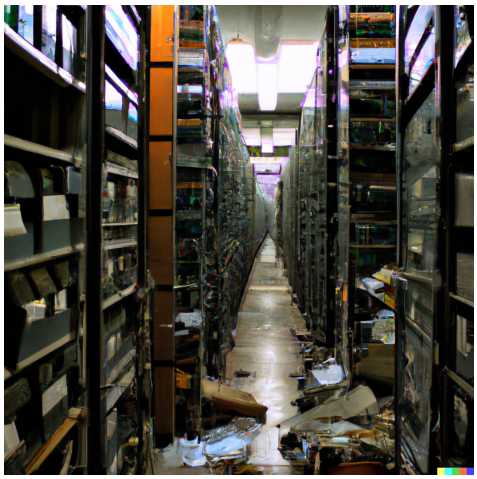
4.4.3 Estimating the spread of predictions from multiple models

Let p_{ji} be the i-th prediction for the real value y_j. Using $\Delta_{ji} = y_j - p_{ji}$, we define

$$\varsigma_j^2 = \frac{1}{N-1} \sum_i \left(\Delta_{ji} - \bar{\Delta}_j \right)^2, \qquad \text{where} \quad \bar{\Delta}_j = \frac{1}{N} \sum_i \Delta_{ij}, \tag{4.45}$$

as the variance of the predictions for the target j in a TeDs. This is an estimate on how accurate the predictions of a network or multiple networks are. In Sec. 7.3 we use this prediction spread to assess the uncertainty of the predictions of a NN under certain conditions. For example, we use Eq. (4.45) to obtain the spread of the prediction of multiple models derived from the same architecture $\mathcal{AR}$ under the same conditions.

5 Data sets and Data Preparation



It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.

(Sherlock Holmes, Arthur Conan Doyle)

DALL·E 2 prompt:

An old library cramped full of computer hard drives with extremely high, rusted shelves and endless hallways, photo[.]

For the training of NNs via supervised learning, we need large amounts of labeled data (see Sec. 4.2.1). In physics, these data sets are provided by MC simulations. To get a detector simulation for Auger, we adopt a two-step process. First, we simulate an air shower using one of the hadronic interaction models discussed in Sec. 2.2.6. Usually, CORSIKA is used for this task. Afterward, we simulate the detector responses of all participating detectors and reconstruct the shower with $\overline{\text{Off}}\underline{\text{line}}$. In the following, to distinguish between both types of simulation, we denote the CORSIKA simulations as air shower simulations and the detector simulations simply as simulations.

Since, at the highest energies, CORSIKA simulations are computationally expensive, we rely on already existing[1] UHECR air shower libraries. Due to the low statistics in the shower libraries, we use each unique shower multiple times by randomizing the impact point of the shower in the $\overline{\text{Off}}\underline{\text{line}}$ simulation. From the detector simulation, we produce simulation libraries to be exploited in the network training and subsequent testing process.

After training and validating our NNs on simulation data, we have to assess how they perform on measurements done by Auger. Due to the expected differences between detector simulations and air shower measurements, we have to correct the predictions of our NNs. For this evaluation, we use $\overline{\text{Off}}\underline{\text{line}}$ to obtain the reconstruction of the air shower measurements. We focus the reconstruction only on SD events and *Golden Hybrid* events.

To improve the training process, we have to pre-process the input data. An vital step to allow for event-level analysis is to encode the shower footprints from the triangular arrangement of the SD into a rectangular grid. The encoding step enables us to use regular TF algorithms designed for rectangular memory. In addition, by exploiting the symmetries of the triangular grid, we can reduce the phase space of the problem during the encoding.

At the beginning of this chapter, we review all simulation data sets used in this work (Sec. 5.1). Here, the CORSIKA air shower libraries and the $\overline{\text{Off}}\underline{\text{line}}$ detector simulations have

[1] Specifically, for this purpose a simulation task force exists in Auger.

been tabulated. In addition, we have defined subsets of the simulation libraries used as training inputs. Afterward, in Sec. 5.2, we focus on the measurements taken by Auger which we use in Chapter 8 to test the NNs defined in Chapter 7. In addition, we examine the most striking differences between air shower simulations and the measurements. In Sec. 5.3, we discuss the procedures we use to pre-process our data. Finally, in Sec. 5.4, we introduce additional variables used in the subsequent analyses as inputs and outputs for our NNs.

5.1 Overview of air shower simulation data sets

In this work, we only use air shower simulations based on the hadronic interaction models QGSJetII-04 and EPOS-LHC (see Sec. 2.2.6) for training and testing our NN-based models. We have selected simulations based on QGSJetII-04 to generate the training data sets for our NN models, as done in the station-level analysis (see Sec. 4.3.1). The event-level analysis discussed in Sec. 4.3.2 makes use of air shower simulations based on the hadronic interaction model EPOS-LHC. We have chosen QGSJetII-04 to directly obtain a cross-check of the results in the event-level analysis based on AixNet. Henceforth, we denote the collections of air showers simulated with CORSIKA shower libraries and the SD detector simulations based on these showers as simulation libraries.

Since we are interested in investigating the effect of the UUB electronics and the addition of the SSD on the predictions of the NN, two SD simulation libraries for each of the hadronic interaction models have been generated using the same version of $\overline{\text{Off}}\underline{\text{line}}$. All of the detector simulations use an ideal version of the 1500 m array of ideal[2] WCD tanks (and SSD detectors). The ideal array is depicted in Fig. 5.1. Note that there are no holes in the array, in contrast to the real array (see Fig. 2.5). In addition, we have simulated dense stations for the data sets based on QGSJetII-04 (see Sec. 3.2.2.D). The dense stations are located at a shower plane distance of 1000 m and employed only for cross-checks, meaning that we do not use them as inputs for our NNs.

For all produced data sets, we have used the development version[3] of the development version of $\overline{\text{Off}}\underline{\text{line}}$ of the `SVN` revision 34 510. The latest change of the repository has occurred on 22.11.2021.

5.1.1 Simulations based on the *Napoli-Praha* shower library

For the current work, we opted for simulations based on the Napoli air shower library with Praha extension (NapLib) shower library [A:15] and its *Praha* extension [A:16] to train our networks. This choice is motivated by mainly three reasons: First, the NapLib shower library is large enough for the network training. Secondly, is has been used in the past, which allows a comparison with previous results. Finally, the air shower events are distributed uniformly in the logarithmic energy given by 10^{18} eV to $10^{20.2}$ eV and in $\sin^2\theta$ for the zenith interval given by 0° to 65° for four separate primaries: proton, helium, oxygen, and iron. For each primary particle, the NapLib shower library is continuous, and in each of the CORSIKA simulations, the same standard atmosphere has been used. Note that the simulation data also covers parts of the phase space at which the 1500 m SD array is not fully efficient anymore. We do not cut away these parts since NN models perform usually worse at the boundaries of the phase space. Therefore, we want this region not to be coinciding with the 100% efficiency points of $10^{18.5}$ eV and 60°.

[2] Meaning that they have the same detector response under the same conditions.

[3] During the generation of the data sets, the Auger collaboration migrated to `Git`. However, the `SVN` history should not have been affected by this.

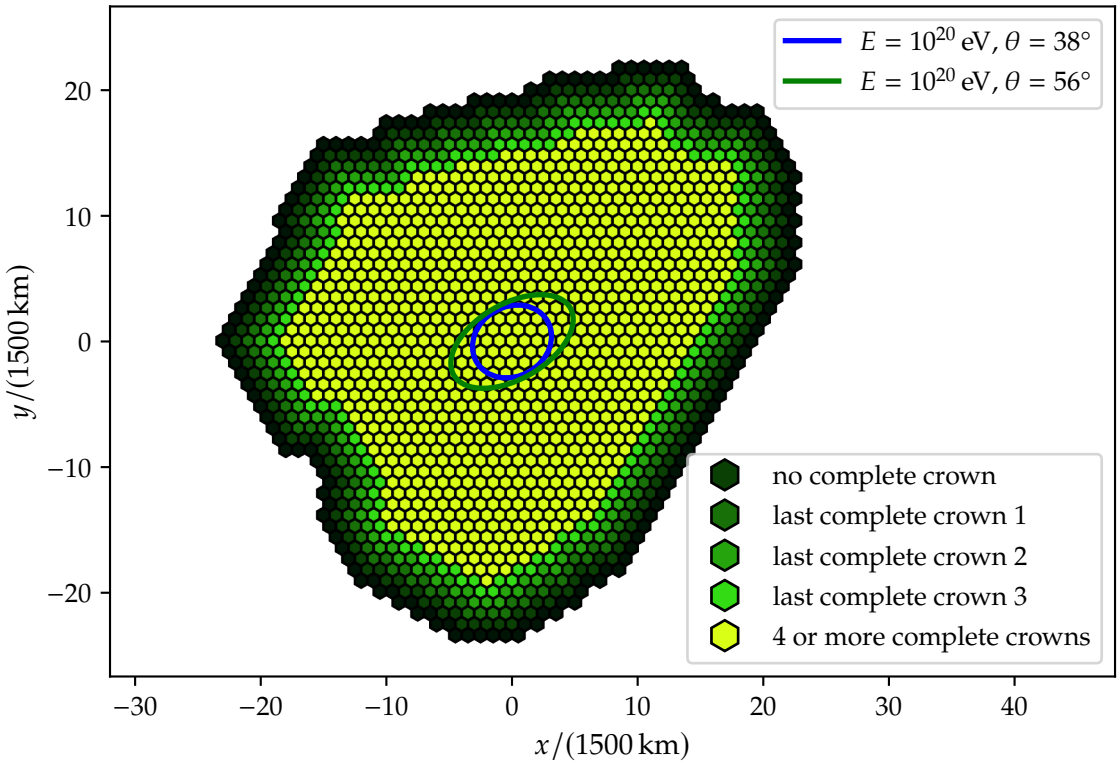


Figure 5.1: Color-coded Wigner-Seitz cells of the triangular grid used in the Offline detector simulations. Each hexagon represents one of the SD stations that are part of the 1500 m grid. The ellipses show the effective size of differently inclined proton shower footprints at 10^{20} eV. The lines represent the position at which a signal of 1 VEM is expected. The ellipses are from the LDF (see Eq. (3.8)) computed in the *Karlsruhe* library (see Sec. 5.1.2). The colors of the hexagons indicate the number of complete crowns around a given station.

Table 5.1: Overview of the local mirror of the NapLib shower library. The original data is hosted at IN2P3. The number of CORSIKA showers varies slightly due to data corruption and copying errors.

		18-18.5	18.5-19	19-19.5	19.5-20	20-20.2	size / TiB
QGSJetII-04	p	4934	5472	4998	5153	2012	6.22
	He	5068	4986	5007	5065	2008	6.51
	O	5225	5000	5006	5010	2003	7.03
	Fe	4725	4997	5027	5090	2000	7.47
EPOS-LHC	p	5861	4686	5002	5013	2006	6.47
	He	4910	4955	5011	5009	2013	6.70
	O	4923	4919	5007	5005	2003	7.16
	Fe	4877	5019	4874	5001	2010	7.60
	γ	9955	10000	9999	9984		5.32

Table 5.2: Overview of UB simulation libraries based on the NapLib shower library. The file sizes of the EPOS-LHC data set are much smaller because we did not simulate dense stations (see Sec. 3.2.2.D).

		18-18.5	18.5-19	19-19.5	19.5-20	20-20.2	size / GiB
QGSJetII-04	p	36084	48686	51228	49691	18112	178.4
	He	34666	50218	50384	49768	19968	183.2
	O	37520	51779	49793	49517	19917	187.1
	Fe	38967	46912	50623	50010	19904	188.9
EPOS-LHC	p	33208	56695	48645	48544	19502	118.2
	He	37687	47489	48674	48711	19556	117.8
	O	39084	47687	48565	48625	19480	120.0
	Fe	41098	47330	48551	47296	19533	121.2
	γ	22512	74172	95752	85616		130.5

The contents of the locally stored version of the NapLib are summarized in Table 5.1. Because of copying errors and corrupted data at the main storage hosted by the IN2P3 Computing Center in Lyon (IN2P3) in Lyon [T:K] the number of files varies slightly between primaries.

As mentioned in the second paragraph of Chapter 5, we reuse the CORSIKA air showers multiple times to increase the number of simulated events. For each of our available hadronic models, a unique shower is simulated ten times for the UB and the UUB electronics. The content of resulting simulation libraries is summarized in Table 5.2 and Table 5.3.

The base data sets require large amounts of disk space. As a conseunece, we cannot afford to use the full data set for each training process since it would make the training extremely slow on our computing cluster, especially, if multiple networks would be trained at the same time on the same machine. Even after extracting only the necessary information from the `.adst` files, this is still a problem. To circumvent this issues, we generate subsets from our base data set by randomly drawing N_{ds} events without replacement.

For each of the generated subsets, we define non-overlapping training, validation, and test sets by drawing from the subsets randomly (see Sec. 4.2.1.A). Since we choose N_{ds}, we are able to generate smaller data sets. The greatest advantage of these data sets is the reduced training time and the reduced resource requirements enabling us to train many networks

Table 5.3: Overview of UUB simulation libraries based on the NapLib shower library. Again, the sizes of the EPOS-LHC data set are smaller because we did not simulate dense stations (see Sec. 3.2.2.D). The increased number of events in the lowest energy bin of QGSJetII-04 is due to saving each use of a CORSIKA shower in a single file.

		18-18.5	18.5-19	19-19.5	19.5-20	20-20.2	size / GiB
QGSJetII-04	p	52730	49055	51234	49722	19148	284.7
	He	48591	50414	50388	49832	19086	287.2
	O	49157	51961	49854	49817	18945	294.0
	Fe	49387	46971	50638	50049	19894	296.9
EPOS-LHC	p	25997	57735	50126	50020	19434	184.5
	He	28849	48510	50007	50110	20130	192.4
	O	33192	48870	49968	50070	20030	197.7
	Fe	34623	48645	50009	48739	19477	192.7
	γ	14422	64280	93387	84832		169.7

in short amounts of time. We use such 'lightweight' data sets later for various cross-checks and the analysis of the variation of hyperparameters providing us with information about network configurations.

In local storage, there is also an older simulation library based on NapLib simulations which is mainly used in the non-UUB analysis of Sec. 6.1 and Sec. 6.2. This is for historic reasons. This library has been simulated with version `r2v9p5` of $\overline{\text{Off}}\underline{\text{line}}$ (see Sec. 3.1) for which no UUB simulation is available. In this simulation library, each of the showers of NapLib has been used six times.

The existence of this old library might raise the question, why we generated additional simulated libraries if we had access to an already finished one. The reason for this lies in the older version of $\overline{\text{Off}}\underline{\text{line}}$. At this point in time, $\overline{\text{Off}}\underline{\text{line}}$ was not capable of simulating the UUB electronics and the SSD detector. Furthermore, the data set was missing the energy ranges of $[10^{18}\,\text{eV}, 10^{18.5}\,\text{eV}]$ and $[10^{20}\,\text{eV}, 10^{20.2}\,\text{eV}]$. Consequentially, we re-simulated the NapLib library to remain consistent. However, in the end, there are no large differences between our UB simulation and that of the older library. Henceforth, we refer to the library that has already existed before this work as the old NapLib simulations. Note that even though this library has been created by a much older version of $\overline{\text{Off}}\underline{\text{line}}$, we still can use the trunk version defined in the last paragraph of the introductory text in Sec. 5.1. In general, `.adst` files are backwards compatible.

In Table 5.4, we have summarized the data sets used in the station-level analysis in Chapter 6. Due to historic reasons, we use the old napoli library for the initial analysis in Sec. 6.2 (see Row 5.4.a and Row 5.4.b). At the time of doing the analysis, the new libraries had not been created. For the second part, however, we use a data set sampled from the new UUB library based on QGSJetII-04 air shower simulations (see Row 5.4.c).

For the event-level analysis in Chapter 7, we solely use the data sets drawn from Table 5.2 and Table 5.3. In Table 5.5, we have tabulated all of the data sets. Most of the data sets are used in Sec. 7.1 to fix the architecture that we use in Chapter 8.

In this work, we do not provide the exact stations (see Table 5.4) and shower events (see Table 5.5) used in our data sets. We expect that using a similar sampling strategy as described in the corresponding analysis chapters is enough to reproduce the results. Even if we would provide the exact ids of the events, the training would – most likely – result in different predictors due to the non-determinism of parallel computing. In Sec. 7.3.2, we show that even

Table 5.4: Overview of the data sets used in Chapter 6 for the station-level analysis. The first column provides a unique identifier for the data set. The numbers refer to the amount of SD-stations taken into account. From one shower event, there could be multiple stations. We use QGSJ and EPOS as abbreviations for QGSJetII-04 and EPOS-LHC, respectively. We added old and new in parentheses to signify from which of the base data sets the data is drawn. New refers to data sets drawn from the newly simulated data (see Table 5.2 and Table 5.3) and old refers to the simulations already available. If the size of the validation set is given in form of a floating point number, it refers to the validation fraction that is used if not statet otherwise (see Sec. 4.2.1.A). In this case, the validation set is sampled randomly from the training data set and not fixed before the training process.

	model	primaries	energy range	TrDs	VaDs	TeDs
a	UB, QGSJ (old)	p, He, O, Fe	[18.5, 20.0]	72000	48000	144282
b	UB, EPOS (old)	p, He, O, Fe	[18.5, 20.0]	-	49691	90462
c	UUB, QGSJ (new)	p	[19.0, 19.5]	246767	0.2	27419

Table 5.5: Overview of our data set used in Chapter 7 for the event-level analysis. The first column provides a unique identifier for the data set. We use QGSJ and EPOS as abbreviations for QGSJetII-04 and EPOS-LHC, respectively. For this analysis, we use only data sets from the new simulation libraries described in this section. If the size of the validation set is given in form of a floating point number, it refers to the validation fraction that is used if not syayey otherwise (see Sec. 4.2.1.A). In this case, the validation set is sampled randomly from the training data set and not fixed before the training process.

	model	primaries	energy range	TrDs	VaDs	TeDs
a	UB, QGSJ	p, He, O, Fe	[18.0, 20.2]	95989	0.1	23998
b	UB, QGSJ	p, Fe	[18.0, 20.2]	95976	0.1	23994
c	UB, QGSJ	p, He, O, Fe	[18.0, 20.2]	395950	0.1	43995
d	UB, QGSJ	p, He, O, Fe	[18.0, 20.2]	479938	0.1	119985
e	UB, QGSJ	p, He, O, Fe	[18.0, 20.2]	444391	0.1	55549
f	UB, EPOS	p, He, O, Fe	[18.0, 20.2]	479879	0.1	119970
g	UUB, QGSJ	p, He, O, Fe	[18.0, 20.2]	95982	0.1	23996
h	UB, EPOS	p, Fe, γ	[19.0, 20.0]	96000	0.2	24000
i	UUB, EPOS	p, Fe, γ	[19.0, 20.0]	95998	0.2	24002

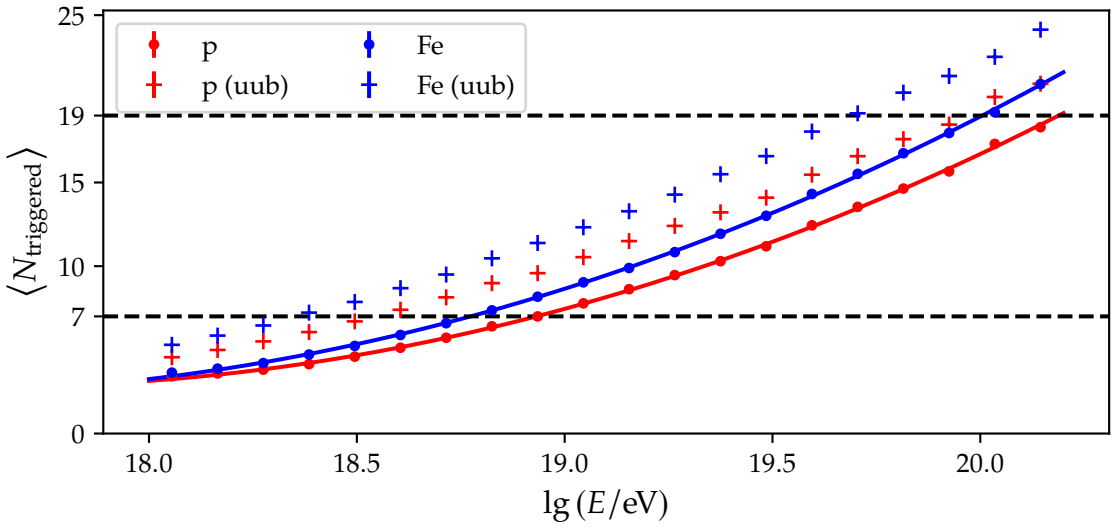


Figure 5.2: Average number of triggered stations binned in logarithmic MC energy. The two dashed horizontal lines represent the break points at which entire crowns can be filled. In the UUB data set, more stations trigger due to a bug in the UUB triggers. The trend of the triggered stations can be parametrized by a polynomial of second order. Hence, the effective trigger area and, therefore, the effective size of the shower footprint increases roughly linear with energy.

under very strict initial conditions, the predictions of networks with the same architecture can deviate from each other after training. However, even though single predictions are different, on average the networks perform quite similar.

Due to a bug in the UUB triggers of the used version of $\overline{\text{Off}}\underline{\text{line}}$, the number of triggered stations $N_{\text{triggered}}$ is – on average – much higher than that of UB (Fig. 5.2). We have depicted this number of average triggered stations in bins of logarithmic energy for the data sets of Table 5.2 and Table 5.3.

5.1.2 Simulations based on the *Karlsruhe* shower library

In contrast to the NapLib, the Karlsruhe air shower library (KarLib) consists only of proton and iron showers at fixed zenith angles and fixed energies. For each hadronic model, primary, zenith angle, and energy, the library has 120 unique CORSIKA showers. These 120 showers are simulated for 12 different atmospheres, which correspond to the 12 months of the year. For the library, the fixed energy values $10^{18.5}$ eV, $10^{19.0}$ eV, $10^{19.5}$ eV, and $10^{20.0}$ eV have been used. The fixed zenith angles are 0°, 12°, 22°, 32°, 38°, 48°, 56°, and 65°. In total, we have 3840 unique CORSIKA showers for each primary and hadronic model.

The KarLib provides us with an ideal testing ground to check the dependence of our models on shower-to-shower fluctuations and on different atmospheric conditions. Since KarLib is only used for cross-checks after the models have been already trained, we do not need to create subsets like in Sec. 5.1.1. For a test, we just use the entire library corresponding to the continuous training set.

We have tabulated the detector simulations of the KarLib in Table 5.6. In this, the UB QGSJetII-04 part of the library is primarily used. We have used each CORSIKA shower 10 times giving us up[4] to 10 events. Note that we did not list the UUB simulation library due to data corruption during copying.

[4]Sometimes, trigger conditions are not met and some of the showers do not produce events.

Table 5.6: Overview of the new simulation libraries based on CORSIKA showers of the KarLib. We use QGSJ and EPOS as abbreviations for QGSJetII-04 and EPOS-LHC, respectively. We only list the total number of events for each primary due to the simplicity of the libraries.

	model	electronics	proton	iron	no. of reuses
a	QGSJ	UB	37356	39000	10
b	EPOS	UB	22230	22403	6

5.2 Overview of data measured by the Pierre Auger Observatory

As basis for our analysis of real measurements with NN in Chapter 8, we use the latest, official data reconstruction performed by the Observer Task Force (Observer) [A:17] and a small extension thereof. The Observer task force usually produces an official data set for each of the bi-yearly International Cosmic Ray Conference (ICRC). The ICRC is considered the most important conference in the field of CR physics. This data set contains all possible combinations of reconstructed detector events up to a pre-defined point in time before the conference. For this purpose the Observer task force uses a special tagged version of $\overline{\text{Off}}\underline{\text{line}}$ (see Sec. 3.1), which is outside of the active development cycle. The standard Observer application of this officially tagged version of $\overline{\text{Off}}\underline{\text{line}}$ is used for the reconstruction.

The last time the collaboration has been successfully pulled through this process was for the ICRC 2019. As a consequence, the most recent official data set does not contain the full range of all reconstructed events. To access the full range, we have complemented this data set by the reconstruction of the most recent events under the same conditions using the tagged version of $\overline{\text{Off}}\underline{\text{line}}$. Since our main interest lies in the analysis of shower footprints, we only need shower events that have triggered SD stations. Furthermore, we only need high-quality FD observations. Therefore, we only reconstructed all possible *Golden Hybrid* and SD events (see Sec. 3.2). We use the *Golden Hybrid* events to re-calibrate predictions of our NNs that can be directly measured by FD, such as the depth of the shower maximum.

Due to the restrictions of our simulation data sets used for training (see Sec. 5.1.1), we have to filter the reconstructions of the real data. We only use events that have a reconstructed SD energy E_{SD} higher than 1 EeV and a reconstructed zenith angle θ_{SD} of less than 60°.

5.2.1 SD data set

Our SD data set spans the period from 2004 to 2021 which is three years more than the original data set used for the ICRC 2019. We filter our raw data set of reconstructions with the cuts defined in Appendix C.2 using the `selectADSTEvents` application from $\overline{\text{Off}}\underline{\text{line}}$. We use only periods which have not been flagged as bad periods [T:L]. Bad periods are time frames in which it can not be ensured that the SD array functions properly.

In Fig. 5.3, we display the distribution of our data set in bins of reconstructed logarithmic energy $\lg(E_{\text{SD}}/\text{eV})$ and reconstructed zenith angle θ_{SD} in terms of $\sin^2$. The number of events in each energy bin is steeply decreasing with energy. Comparing the rate of decrease, different aspects of the spectrum in Fig. 2.2 are visible such as the flattening at about $10^{18.5}$ eV and a steeping at $10^{19.5}$ eV. The distribution of all events shows clear deviations from the expected uniform distribution in $\sin^2$. This is caused by the reduced detector efficiency at energies below $10^{18.5}$ eV. For events with reconstructed energies above $10^{18.5}$ eV, no deviation is observed and the data is uniformly distributed. In total we have 1,935,850 events. Due to the steeply falling distribution we only have 21,566 events with a SD energy E_{SD} above 10 EeV.

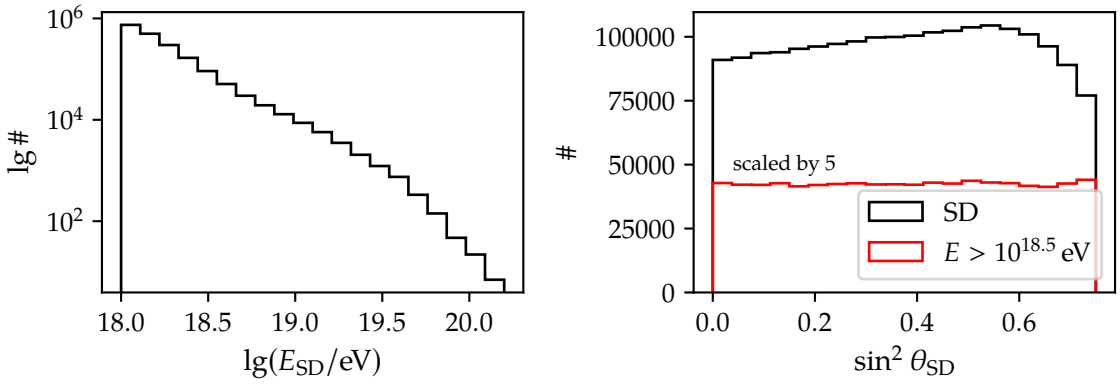


Figure 5.3: Distribution of the SD data set in bins of reconstructed logarithmic energy $\lg(E_{SD}/eV)$ (*left*) and reconstructed zenith angle θ_{SD} in terms of $\sin^2$ (*right*). The black histogram represents the complete data set while the red histogram includes only events with a reconstructed E_{SD} energy above $10^{18.5}$ eV. To improve visibility, we scaled the latter histogram by a factor of 5.

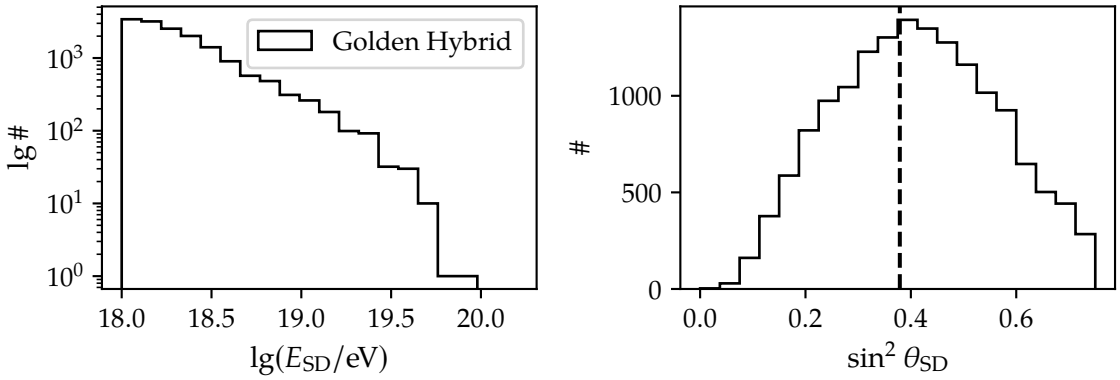


Figure 5.4: Distribution of the *Golden Hybrid* data set in bins of reconstructed logarithmic energy $\lg(E_{SD}/eV)$ (*left*) and reconstructed zenith θ_{SD} in terms of $\sin^2$ (*right*). The vertical dashed line marks 38°. The striking difference in the zenith distribution compared to the distribution of SD events in Fig. 5.3 is due to the fiducial field cut (see Fig. 3.2)

5.2.2 *Golden Hybrid* data set

Since 2019, there are no updated versions of the atmospheric databases [A:18], our extended version of the *Golden Hybrid* has only one additional year of data. The *Golden Hybrid* data set consists of shower events detected by both the SD and FD that can be reconstructed by each detector separately. Therefore, the *Golden Hybrid* data set consists of observations of the longitudinal shower development and the shower footprint simultaneously. In addition to the cuts in Appendix C.2, we also apply the cuts for the FD reconstruction used for the energy calibration (see Appendix C.2) to ensure a high-quality FD measurement (see Sec. 3.2.3).

The distribution of events in the *Golden Hybrid* data set differs from the distribution in the SD data set (Fig. 5.4). For example, the energy distribution of the *Golden Hybrid* is slightly flatter around 10^{18} eV. In the energy bins above $10^{19.5}$ eV, the distribution exhibits a much stronger decrease of event numbers, most likely caused by the low flux above this energy in combination with the FOV cut. The difference in the distribution of zenith angles is due to the fiducial field cut (see Fig. 3.2). In total, we have 15,850 events in the entire energy range and only 683 above a reconstructed SD energy of 10 EeV.

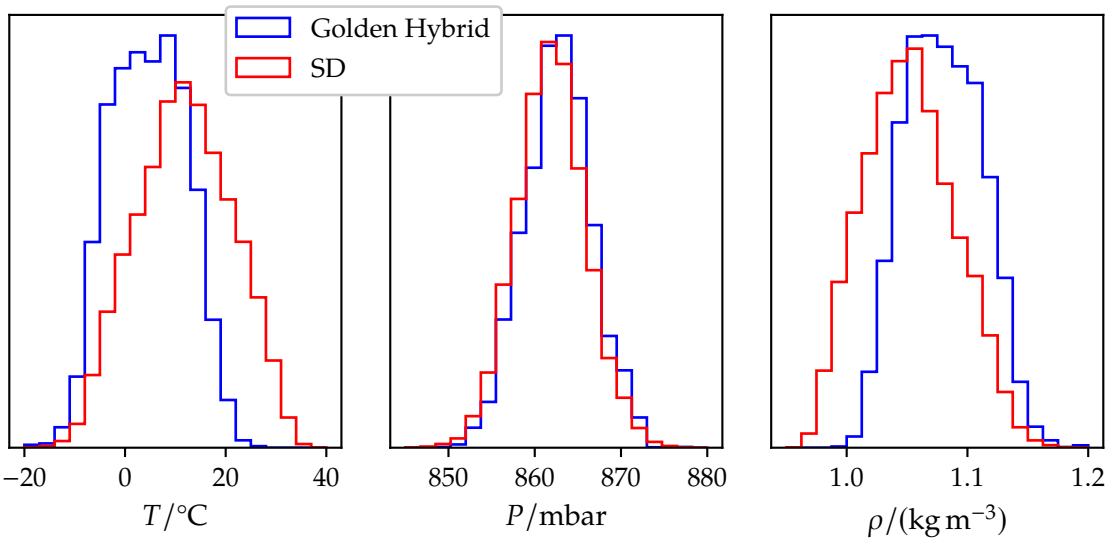


Figure 5.5: Normalized distribution of number of *Golden Hybrid* events (blue) and SD events (red) in bins of temperature (*left*), atmospheric pressure (*middle*), and air density (*right*) at the observatory. The temperature and air density distributions of the *Golden Hybrid* and SD events differ clearly showing the different atmospheric conditions during the data taking of *Golden Hybrid* hybrid events.

A Handling of multi-FD-telescope data

We want to use *Golden Hybrid* events because we can cross-check our predictions for observables, such as X_{max}.

If more than one FD telescope detects a shower event, we have to define how to combine their observations. The FD telescopes measure showers independent from each other. After the FD reconstructions, we obtain sets of estimations for shower observables, such as the shower energy E_{fd} and the depth of the shower maximum X_{max}, and uncertainty estimates for the estimations. Therefore, we use the variance-weighted average (see Eq. (A.9)) to combine these measurements. In the case of X_{max}, we would obtain

$$X_{\mathrm{max,FD}} = \frac{\sum_i X_{\mathrm{max},i}/\sigma^2_{X_{\mathrm{max},i}}}{\sum_i 1/\sigma^2_{X_{\mathrm{max},i}}}, \tag{5.1}$$

where the sum goes over the telescopes.

B Differences between *Golden Hybrid* and SD data set

The *Golden Hybrid* events can only be detected in moonless nights due to the restrictions of the FD (see Sec. 2.3.1). As a consequence, the *Golden Hybrid* data set covers only a subset of the phase space spanned by the SD data set. This is not only caused by the additional energy calibration cuts (see Appendix C.2). Atmospheric conditions that occur during the day, such as hot temperatures due to direct sunlight, are not accounted for in the *Golden Hybrid* data set. Comparing the normalized event distributions for temperature, pressure, and air density for both types of events (see Fig. 5.5), there is a visible shift to lower temperatures and higher air densities for *Golden Hybrid* events. Hence, if we do a correction of our predictions based on *Golden Hybrid* events, we have to ensure that our corrections work in regions outside of the *Golden Hybrid* events.

5.2.3 Differences between simulations and measurements

There are plenty of differences between the measurements and the simulation libraries which we use as training data (see Table 5.2) for our NN. Such mismatches could yield problems in the inference process and give us biased predictions. We want to validate and correct the predictions of our NN by checking for non-physical dependencies in the predictions that clearly stem from the differences between the data sets.

Auger is not an ideal detector. The SD stations and the PMTs inside are subject to ageing which changes the detector response over time. Due to the on-board calibration (see Sec. 3.2.2.C), the ageing does only affect the measurement of total signals minutely. However, the ageing changes the shape of the measured time traces. Since one of our intentions of using NNs is to exploit exactly these time traces, it could result in shifts in the predictions. We account for this aging by analyzing the dependence of our predictions on the runtime of the array, average runtime of the stations triggered in an event, and the area over peak a_{p}, which is for UB simulations fixed to 3.2. Another consequence of the detector aging is the loss of working PMTs in some stations of the array. An incomplete set of PMTs in a SD station causes the signal measurement to be more dependent on the azimuth direction of the shower.

Unlike the ideal grid used in simulations (see Fig. 5.1), the real detector grid is non-perfect. The SD station positions have different heights and are not perfectly aligned to the exact grid positions. Furthermore, even when using the 6T5 cut, we still encounter holes and non-functioning stations outside of the first crown. For example, this could introduce biases if the network counts triggered stations to obtain a prior for the energy.

Our simulation library used for training is based on an air shower library that uses only a single standard atmosphere (see Sec. 5.1.1). Therefore, the NN cannot account for changes in the footprint due to weather conditions, daily changes, and yearly variations. For example, a shower in a very dense atmosphere should experience a higher attenuation of its electromagnetic component changing the total signals and the signal shape. This, in turn, would most likely yield in a wrong prediction of the neural network.

5.3 Preparation of input and output data

Before we use the data in the NN-based models, we have to prepare it in such a way that it benefits the training process. Since there is no metric to find the perfect pre-processing, we have to rely on what-worked-before, trial-and-error, and ingenuity. If a pre-processing method is not applicable, usually, it can be seen in the predictions of the models and during the training process.

5.3.1 Preparation procedures of scalar data and traces

There are two popular methods of preparing data for machine learning models. Both of them make the data more accessible for machine learning algorithms by using linear transformations. The first method is the already mentioned standardization via Eq. (4.33). The second method is called normalization. We squash our data into a predefined interval. Most commonly, this interval is $[0, 1]$. We transform the scalar values x by using the transformation

$$x \to \frac{x - \min x}{\max x - \min x}. \tag{5.2}$$

Note that doing this may cause problems when performed independently on data sets with very different distributions. Therefore, we use the mean and standard deviation of the data set used in the training process in cases like this.

A Standardization of station trigger timing

We follow the same standardization procedure for the trigger times as in [P:104]. We subtract from the trigger times the average trigger time of the event. Afterward, we divide the result by the standard deviation of the distribution of average trigger times for our entire training data set:

$$t_{\text{trigger}} \to \frac{t_{\text{trigger}} - \left\langle t_{\text{trigger}} \right\rangle_{\text{ev}}}{\sigma_{t_{\text{trigger}} - \left\langle t_{\text{trigger}} \right\rangle_{\text{ev}}, \text{data set}}}, \tag{5.3}$$

where $\langle \cdot \rangle_{\text{ev}}$ is the average over the trigger times of all stations in an event and σ is the standard deviation of the differences between the trigger time and the corresponding average trigger time. Hence, we use an event-level parameter and a parameter derived from all events.

B Pre-processing of scalar data using non-linear transformations

Most of the time, using either normalization or standardization is sufficient to prepare data for training and prediction of NN. However, sometimes it is beneficial to perform a non-linear transformation before or after the normalization/standardization procedure.

If $f(x)$ is a distribution of the input values x, then $f(y)|\partial_y x|$ is the distribution of $y = g(x)$. Hence, we can transform the distribution of input values. This can prevent that the network focuses on over-densities during the training, such as that of the energy distribution in the training data. The energy is distributed like $1/E$ (see Sec. 5.1). However, it is uniform in decadic logarithmic energy. Therefore, $\lg E$ is preferably used as a network input rather than E. We perform such a "flattening" also for the zenith angle θ, which is distributed uniformly in $\sin^2$, assuming an isotropic distribution of CR over the sky.

If there are multiple scalar inputs which are independent from each other, we can also generate 'higher-order' features from the inputs by multiplying powers of different input features in different combinations. These kind of features can be generated as follows: Let $\{x_1, x_2, \dots x_n\}$ be the set of our input features. We choose an order P. Then, we search all unique combinations $x_1^{p_1} x_2^{p_2} \cdots x_n^{p_n}$ that satisfy $p_1 + p_2 + \cdots p_n < P$. Doing this yields a new set of input features of the form

$$x_{\text{new}} = \{x_1, x_2, \dots, x_n, x_1^2, x_2^2, \dots x_n^2, x_1 x_2, \dots\} \tag{5.4}$$

C Preparation of traces

The raw UB and UUB traces contain regions that do not contain information content. To reduce the memory size, allow[5] a more focused training, and align our input traces, we trim them on a trace-by-trace basis. In Offline, the index i of the bin of the real signal is saved in `SdRecStation.SignalStartSlot`. We remove all bins that have smaller indices. Then, if our target trace length is L_t bins, we take bins until the index $i + L_t$. If the trace is not long enough we pad with zeros.

Furthermore, we also believe that the baseline fluctuations do not contain any important information for our analysis. Hence, we perform a threshold cut to remove this perceived noise from our input data set. For this, we use a constant value of 10^{-3}.

For a better comparison with the previous work on event-level NN analysis in Sec. 4.3.2 [P:104], in addition, we will transform our signals via the decadic logarithm:

$$s(b) = \frac{\lg(1 + S_{\text{off}}(b)/\text{VEMPeak})}{\lg(1 + 100)}. \tag{5.5}$$

[5] We are sure that the network would eventually learn that these parts of the data do not matter for the output.

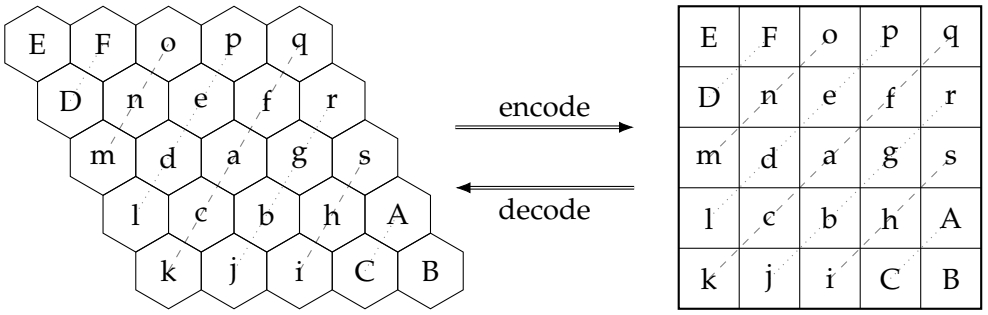


Figure 5.6: Illustration of the encoding of the triangular grid into rectangular memory. The letters represent the position in both grids. The number of nearest-neighbors in the triangular grid is different from that of the rectangular grid. To properly encode it, we have to add non-complete crowns in our rectangular representation if we want to use our memory efficiently. These are represented by uppercase letters (A-F).

This choice linearizes the exponential decay of the trace signal putting the trace scales into context. In contrast to [P:104], we use the traces directly from Offline for historic reasons. These are given in units of VEMPeak and can be converted via Eq. (3.6). Since a_p is constant for simulation, it is only a scaling factor and should not change the result of the analysis.

5.3.2 Encoding of SD grid into rectangular space

The detector stations of the surface detector of Auger are arranged in a triangular grid. We can describe the positions of the stations in the grid via

$$\boldsymbol{d}(i,j) = i\boldsymbol{c}_1 + j\boldsymbol{c}_2 = \left[i \begin{pmatrix} 1 \\ 0 \end{pmatrix} + j \frac{1}{2} \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix} \right] d, \tag{5.6}$$

where $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are an arbitrary choice of base vectors of the triangular grid and d is the lattice constant (for the SD main array $d \approx 1500\,\mathrm{m}$). The integer coordinates i and j uniquely identify each detector of the array. Coincidentally, positive integer coordinates are how regularly the elements of 2D-arrays are referenced. Therefore, by shifting i and j to i^+ and j^+, we get a direct map between (computer) memory and SD station positions.

To make this encoding comparable for all events, we have to choose a fix point. An ideal choice is to put the hottest station of an event at its center at $(i,j) = (0,0)$ and arrange all others around it. To encode the n-th crown completely, we need at least an array of the size $(2n+1) \times (2n+1)$ (see Fig. 5.6). Hence, we chose this quadratic memory format as basis for our encoding. To represent (i,j) in memory, we just have to choose one corner and use it as a reference point. The memory coordinates are then $(i^+, j^+) = (i+n, j+n)$. Without loss of generality, we can use this procedure to encode any information from the stations. Henceforth, we denote M_s as the width and height of our rectangular encoding. We call it the memory layout size.

5.3.3 Geometrical standardization of shower footprints

Assuming that all stations lie on the same plane, that there are no corner effects, and that there are no dependencies on the azimuth on shower propagation, we exploit the symmetries of the grid to reduce the effective phase space of our input data. This is similar to Appendix B.1 and Sec. 4.3.2.A. However, instead of imprinting the symmetry on our NN, we want to remove it entirely. We choose grid coordinates i and j from Eq. (5.6) as the basis of

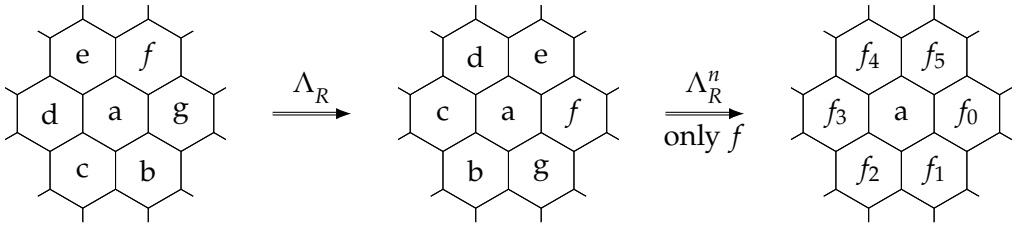


Figure 5.7: Illustration of the rotations defined in Eq. (5.8) on the first crown. The center point of the rotation is at the grid point which lies in the cell "a". Applying the rotation matrix Λ_R on the left grid, rotates all stations clockwise around "a". Following only the cell f, the effect of all other rotation matrices is visible (see Eq. (5.8)). In the right grid, the index of f corresponds to the used matrix n.

our investigations. Again, we use the hottest station as the fix point $(0, 0)$; For our standardization procedure, we want to find all distance-conserving transformations that preserve the underlying grid around this fixed grid point.

A SD grid-conserving transformations

Intrinsically, a triangular grid is invariant under rotations of multiples of 60°. Using the grid coordinates, we express the 60° rotation as

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \Lambda_R(60) \begin{pmatrix} x_i \\ y_i \end{pmatrix} \equiv \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \tag{5.7}$$

The matrix $\Lambda_R(60)$ rotates all grid points by 60° clock-wise (see Fig. 5.7) for our choice of the base vectors in Eq. (5.6). Henceforth, we will denote $\Lambda_R(60)$ as Λ_R. In Appendix A.3, we show how Λ_R is computed for arbitrary unit vectors. By multiplying Λ_R with itself n times we obtain the n-th "grid-conserving" rotation matrix

$$\Lambda_R(60n \pmod{360}) = \Lambda_R^n. \tag{5.8}$$

Hereby, the unit matrix corresponds to $n = 0$ (or $n = 6$). To gain a better understanding of the matrices in Eq. (5.8), we have depicted their transformations in Fig. 5.7.

In addition to the rotational symmetries, certain reflections also conserve the grid. If we restrict the reflection axis to go through a station, we find that there are six possible ones. Those correspond to the reflection symmetries of a hexagon. Fortunately, we require only one of them since all other reflections can be constructed from the rotations and the reflection choice (see last paragraph). Therefore, from a general point of view, the choice of this axis is also arbitrary. For convenience, we choose the straight line defined by $\boldsymbol{c}_2$ (see Eq. (5.7)) because it gives us a simple reflection matrix:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \Lambda_M(60) \begin{pmatrix} x_i \\ y_i \end{pmatrix} \equiv \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \tag{5.9}$$

We illustrate the effect of $\Lambda_M(60)$ in Fig. 5.8. Note that the elements of $\Lambda_M(60)$ are only true if we use the unit vectors of Eq. (5.6). In Appendix A.3, we show how to obtain them for an arbitrary choice.

Completeness of transformation: Our choice of transformations covers all possible symmetries of the triangular grid. Since reflections and rotations are the only isometries in the two-dimensional plane, we only have to show that we can construct all other reflections with

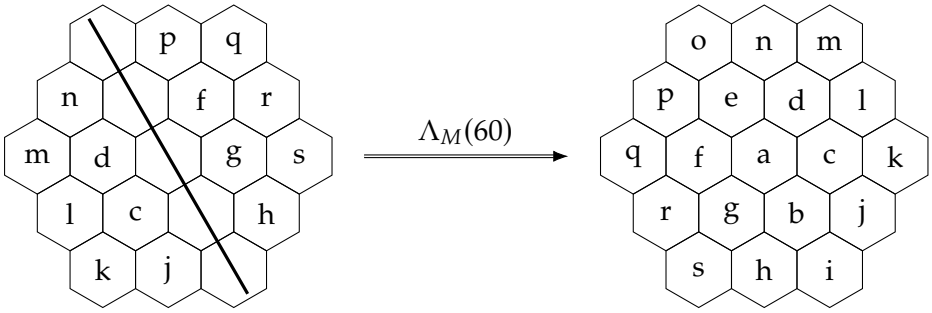


Figure 5.8: Illustration of reflection defined in Eq. (5.9) on the first and second crown. The reflection axis connected to $\Lambda_M(60)$ is defined by $\boldsymbol{c}_2$ (see Eq. (5.6)).

the matrices of Eq. (5.8) and Eq. (5.9). We denote the six distinct reflection operations as $\Lambda_M(0), \Lambda_M(30), \ldots \Lambda_M(150)$ where the argument corresponds to the angle of the reflection axis with respect to the (horizontal) x-axis. The set of $\Lambda_M(0)$, $\Lambda_M 60$, $\Lambda_M(120)$ and $\Lambda_M(30)$, $\Lambda_M(90)$, $\Lambda_M(150)$ lie 60° apart from each other. If we chose one transformation of each set, we are able to construct the others via the already defined rotations. Consequentially, we only have to connect those both groups of matrices. Using $\Lambda_M(30) = \Lambda_R(-30)\Lambda_M(0)\Lambda_R(30)$, we get

$$\begin{aligned}\Lambda_R(60)\Lambda_M(30) &= \Lambda_R(30)\Lambda_M(0)\Lambda_R(30) = \Lambda_R(30)(\Lambda_M(0)\Lambda_R(30)\Lambda_M(0))\Lambda_M(0)\\ &= \Lambda_R(30)\Lambda_R(-30)\Lambda_M(0) = \Lambda_M(0),\end{aligned}$$

proofing that this connection exists.

Reduction of phase space Showers are (to first order) uniformly distributed in the interval $[0°, 360°)$ over the azimuth ϕ. By using the transformation, we can rotate and mirror all detector responses into an arbitrary 30° slice. This effectively reduces the input phase space by a factor of 12 and simultaneously makes it unnecessary to choose a special NN architecture to account for those symmetries.

B Memory efficiency

Due to the encoding of the shower footprint, we need $M_s^2 = (2n+1)^2$ times the size of the station inputs we want to use as input in an NN. If the fifth crown should be completely ($n = 5$) represented in the trace, and traces of 120 bins be used, the data for one event would already amount to ~10 kB in memory. With new methods and the SSD data, this value increases even further. Because there is only a finite amount of memory, the choice in which direction the showers are standardized is not arbitrary (see Fig. 5.6). Depending on the chosen base vectors in Eq. (5.6), we want to align the showers in such a way that we maximize the information in our memory. Since all twelve choices to standardize the shower are equivalent from a physics standpoint, we choose the one that synergizes with the memory layout.

Stations have a higher probability of triggering when they lie nearer to the shower core. Therefore, shower measurements are elongated in this direction. By using the diagonal from top left to bottom right, we put the bins for not completed crowns near the shower axis (see Fig. 5.6). Not setting those to zero allows us to increase the information density for similar memory layout sizes. In Fig. 5.9, we compare the fraction of stations we lose by choosing a certain grid size for this standardization, no standardization, and an ineffective choice of standardization axis. The effect is especially prevalent for lower memory sizes. The effective

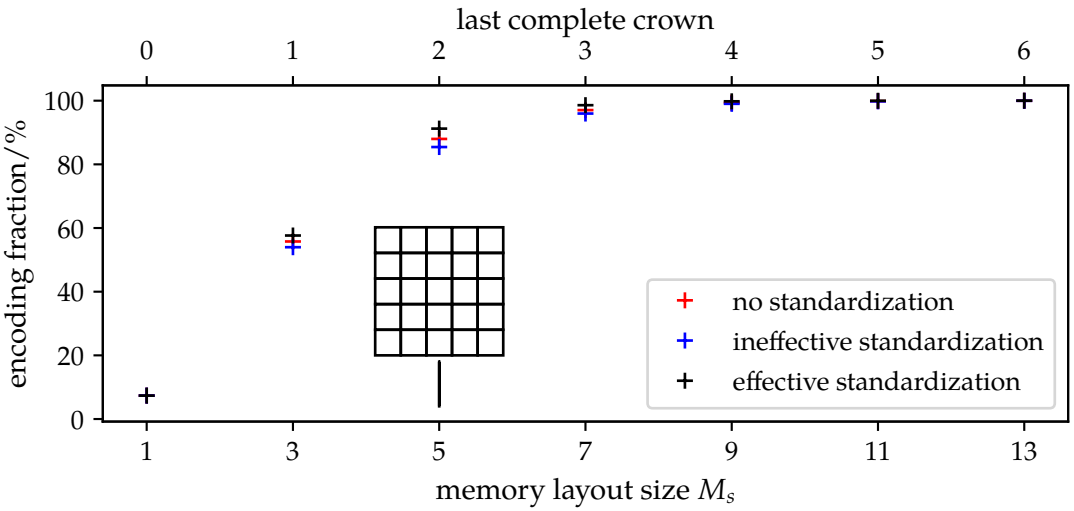


Figure 5.9: Fraction of stations in the data set that can be encoded inside a square memory layout with a memory layout size M_s for different memory configurations. At a memory layout size of 7 over 90% of stations can be encoded. The standardization using an effective layout increases the fraction of stations. Therefore, it is more memory efficient.

standardization beats both other methods[6]. Since this increase of information is free, we recommend it. Note that with the same argument, the information density could also be increased by trying different input shapes that are not quadratic.

C Check of the validity of the standardization procedure

If we want to use this standardization procedure, we need sufficient knowledge about the value of the azimuth angle ϕ of a shower event. Therefore, we face an intrinsic systematic error due to the geometrical reconstruction by the SD standard algorithm. To get a handle on this error, we compare the MC azimuth ϕ_{MC} with the reconstructed azimuth ϕ_{SD} taken from $\overline{\mathrm{Off}}\underline{\mathrm{line}}$. We want to use ϕ_{SD} for the standardization procedure. For this analysis, we analyze proton and iron events which are uniformly distributed in logarithmic energy range $[18.5, 20]$.

There are two main error sources: On the one hand, the azimuth difference $\Delta\phi$ could be greater than our "projection window" of 30°, on the other, we expect ambiguities for azimuths near the corners of the windows. Both errors are intertwined but conceptually different. In the former case, we face the problem of a highly erroneous reconstruction. Fortunately, $\Delta\phi$ value outside of the target range are quite uncommon (see Fig. 5.10). The polar plot shows a histogram of the full possible range of $\Delta\phi$ in logarithmic scale. Even though the difference distribution exhibits a fat-tail (see Fig. 5.10), from about 300,000 reconstructed showers only about 1,000 are outside of the valid region. Therefore, we assume that the error through reconstruction is marginal and can be neglected. Most likely, those erroneous reconstructions are due to vertical showers. There is a considerable amount of events that are matched to the wrong rotation matrix (see Fig. 5.11). Unfortunately, this does not change if we demand that all stations around the hottest station must have triggered. This means that this problem is not purely related to showers with a small footprint.

What happens when we include the reflection in the mix? Let us take a look at the differ-

[6] Even though the ineffective standardization loses stations, it will perform better than a non-standardized data set due to the standardization itself.

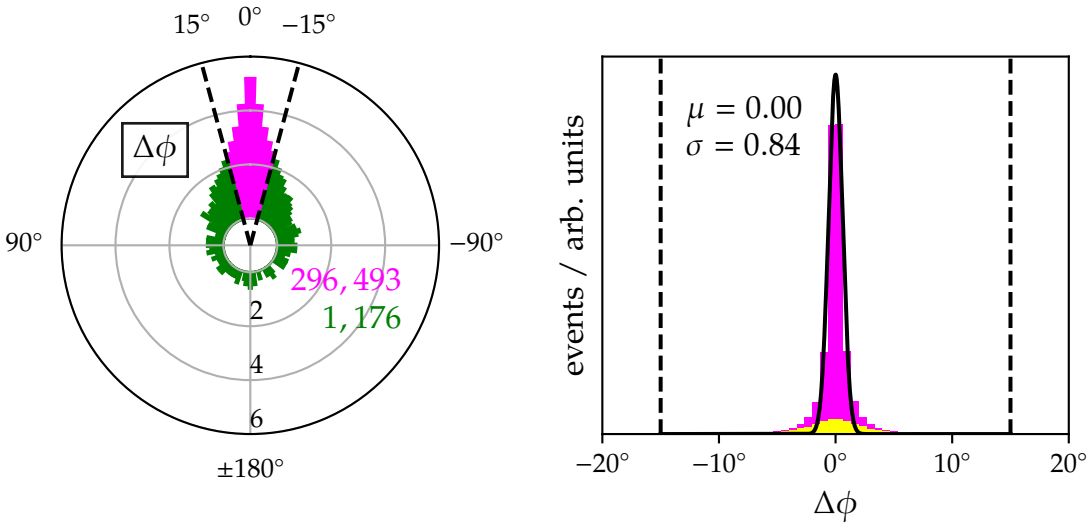


Figure 5.10: Difference of the simulated and reconstructed azimuth angle $\Delta\phi$ for all possible angles (left, logarithmic) and values near the valid interval (right). We find that from about 300,000 events only about 1000 are reconstructed outside (green area, left plot). The outer ring indicates the average zenith angle in the bin range. Therefore, these are primarily vertical events. From 1176 events over 95% are from events with zenith angles below 20°. Therefore, even though the difference distribution exhibits because of this a fat tail, which can be seen in the Gaussian fit (right), we conclude that the reconstruction error does not greatly influence the standardization procedure. The colorful band in the right plot indicates the average zenith angle. In the region of interest, it is almost exclusively about 38°.

ences between the transformed MC and the reconstructed azimuth angle. The azimuth of the transformed shower footprint is given by

$$\phi_{\text{tr}} = 60° - \Bigg| \underbrace{\phi - \left\lfloor \frac{\phi - 30°}{60°} \right\rfloor 60°}_{\text{rotation}} - 60° \Bigg| , \tag{5.10}$$

where $\lfloor . \rfloor$ denotes the floor operation[7]. As seen in Fig. 5.12, even when the wrong rotations are used, the reflection counteracts this. Larger differences stem from very vertical showers. For low zenith angles, the azimuth angle barely plays a role in the shape of the footprint. As a consequence, the erroneous transformed shower footprint is very similar to the correctly transformed shower footprint in these cases. Moreover, even if we would only regard the rotation, this effect should be minimal. We conclude that this error is also negligible.

Other error sources that are not completely accounted for in the simulations are the variation of the ground height, the geomagnetic effect, and the borders of the array. All of these effects break the symmetries we are exploiting in this standardization procedure. Henceforth, we assume that all of them are relatively small.

5.3.4 On-the-fly data augmentation

To enhance the possibility of generalization of our networks, we perform on-the-fly augmentations of our input data sets. This enables us to generate data sets dynamically that, in

[7]Please note that $\lfloor -i.jk \ldots \rfloor = -i - 1 (i \leq 0)$. Therefore, $\lfloor -0.jk \ldots \rfloor = -1$.

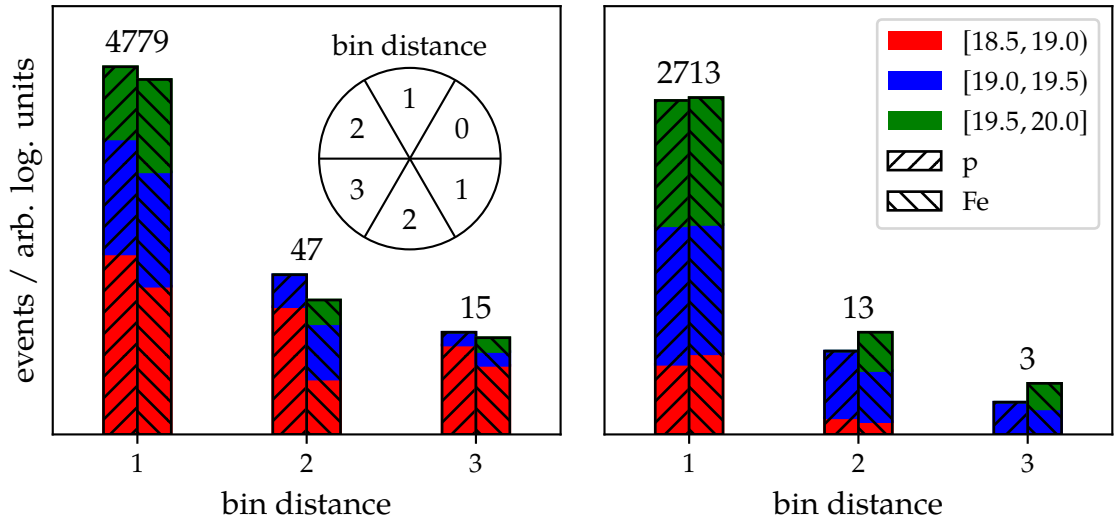


Figure 5.11: The number of events sorted in neighboring, wrong bins resulting in a wrong rotation for all events (*left*) and only events that survive a 6T5 cut (*right*). From all events only about 2% are rotated incorrectly.

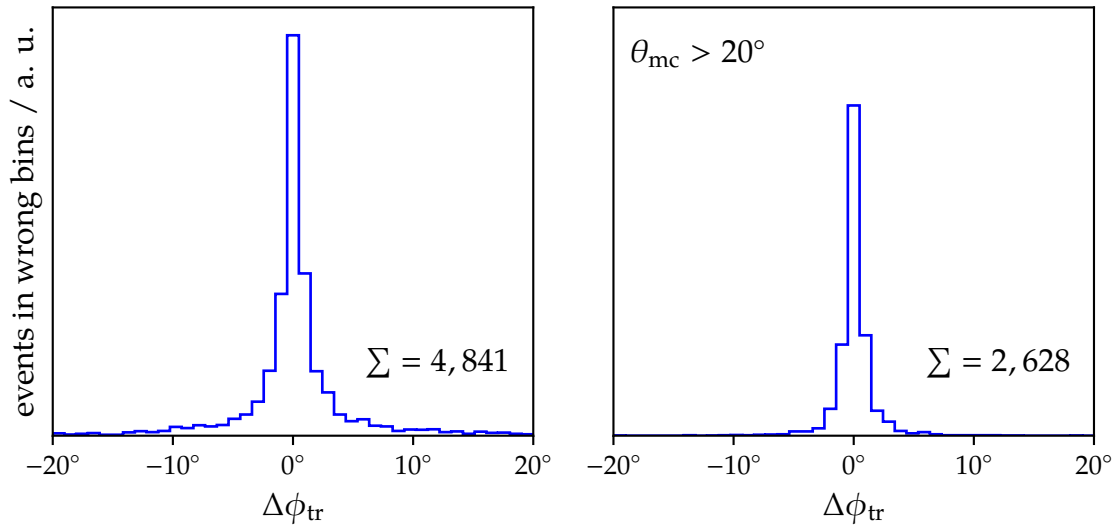


Figure 5.12: Difference distribution of the fully transformed (see Eq. (5.10)) Monte-Carlo and reconstructed azimuth. Even though the footprints shown here are rotated wrongly, the reflection diminishes this error greatly. By cutting away low zenith angles, we are able to remove the highly erroneous values.

theory, we are able to reproduce by fixing the seed. We implemented it via threading. During training, we run a parallel thread that chooses the indices for a fixed number of batches of our input data. The data in the batches is then modified and stored in memory in such a way that the training algorithm has easy access.

A Automatic garbage collection

The `Python` programming language is very memory-demanding. Common workarounds for very large data sets, such as memory mapping, fail and still fill (in many implementations) the memory step-by-step to reduce the Input-Output (IO) on the hard drive. Although much faster than loading it in the beginning, it still does not solve the problem of data sets being too huge to fit into the memory. Other workarounds in TF directly use a batching process, which makes it impossible to randomize data points between those super batches from which the regular batches for network training are created.

Since we feed our networks via an additional thread, we also have more control over which data stays in memory without being too invasive. After a pre-defined number of batches, we wipe the allocated space clean to ensure that Python cannot "optimize" at the wrong places.

B In-place dimensional reduction of input data

Another trivial (but useful) application of on-the-fly data augmentation is the modification of the input data outside of the network. In this way, we are able to reuse data sets that contain larger shower footprints and longer trace lengths saving disk space in the process. Moreover, we can use this to allow for an easy way to check the effect of the dimensionality of our input data.

C Black tank augmentation

At each moment of running the Observatory, there are a certain number of SD stations in the array that do not respond to CDAS. They are denoted as black tanks. There are many reasons for something like this to happen, e.g., broken detector station or no power in the onboard battery due to the weather conditions. We can obtain information about the black tank rate from [T:M]. The average number of black tanks in the array is of the order of 30. Moreover, the SD array has holes (see Fig. 2.5). At this regular grid positons, no stations have been placed. Both of these effects are not included in regular simulations.

To make the network more robust against this kind of "corrupted data", we have included the option to remove a certain percentage of randomly chosen non-hottest stations of each event. The default value of this fraction is 5%, which is much bigger than expected from the array monitoring. In addition, this kind of (in-training) data augmentation also suppresses potential over-trainin g due to the implicit increase of different shower footprints.

5.4 Additional features generated from base data

In addition to the shower observables provided by $\overline{\text{Off}}\underline{\text{line}}$ and CORSIKA, we also compute a hand full of the event- and station-level quantities that are not directly accessible. These can be novel input features or output targets for predictions. Then, in the course of this work, we will analyze how some of these new features could improve the predictions of our NN. The outputs are mostly related to the muon content of the shower: a local or global observable that we are highly interested in.

5.4.1 Timing relative to the planar shower front

The timing of the triggers corresponds to a curved shower plane (see Eq. (3.3)). The curvature of the shower front is an indicator for the propagation processes and the shower maximum depth $X_{\max}$. However, without exact positioning, it is hard to be inferred from the base data.

Nevertheless, we can put this curvature into perspective. We compare the real trigger times to the plane front model (see Sec. 3.2.2.B). Using the position of the core as the fix point where the shower plane is at $t = 0$, we can rewrite Eq. (3.2) into

$$t_i = \hat{\boldsymbol{a}} \cdot (\boldsymbol{x}_\mathrm{c} - \boldsymbol{x}_i), \tag{5.11}$$

where $\boldsymbol{x}_\mathrm{c}$ is `SdRecShower.CoreSiteCS`, $\hat{\boldsymbol{a}}$ is `SdRecShower.AxisSiteCS`, and t_i is the time of the shower plane at detector position $\boldsymbol{p}_i$. Using the trigger times t_i relative to the core time t_core, we can construct the time relative to the shower front

$$t_{\mathrm{pf},i} = (t_i - t_\mathrm{c}) - t_i^\mathrm{pf}. \tag{5.12}$$

This process encodes the curvature in the times and might be a more efficient way to use as an input for the NNs. From Universality we know that this relative timing gave a boost in the predictive power of the model [A:19].

5.4.2 Energy dependence and width of the shower depth distribution of different primaries

Since we have only a finite amount of simulation data, we use the following procedure to approximate the distribution of shower depths depending on energy and primary. We assume that the underlying distribution of the shower depth is approximately similar to the "skew normal distribution" [C:23] (see Fig. D.1), which introduces a skewness via the Error function erf:

$$\mathrm{pdf}_\text{skew-normal}(x;\mu,\omega,\eta) = \mathrm{pdf}_\mathrm{Gauss}(x;\mu,\omega)\left[1 + \mathrm{erf}\left(\nu\frac{x-\mu}{\sqrt{2}\omega}\right)\right], \tag{5.13}$$

where μ, ω, and ν are fit parameters that correspond roughly to mean, spread, and skewness, respectively. They depend on the primary and the energy of the shower. Assuming that they do not vary greatly over the target energy range, we parametrize them in $\lg(E/\mathrm{eV})$ by a linear, quadratic, and linear function[8], respectively. Hence, the corresponding Probability Density Function (PDF) has 2 + 3 + 2 parameters. We obtain them by minimizing

$$\ln\mathcal{L} = -\sum_x \ln \mathrm{pdf}_\text{skew-normal}(x; p_1, \ldots, p_7) \tag{5.14}$$

via the implementation of the Nelder-Mead algorithm [C:24] from `scipy` [T:N] using 3000 iterations. For each primary, we perform a separate minimization. In Table 5.5 we summarized the dataset.

We want to emphasize that the choice of distribution is arbitrary. Therefore, we want to validate this approach. To do this, we compare the values of statistical quantities inside energy bins with that of the value given by the distribution Eq. (5.13) using the parameters from the minimization (see Eq. (5.14)). We want to investigate the moments of the distribution and compare them to binned values. The mean

$$\langle x\rangle = \mu + \sigma\sqrt{\frac{2}{\pi}}\frac{\nu}{\sqrt{1+\nu^2}} \tag{5.15}$$

[8] $f(\lg(E/\mathrm{eV})) = a + b\lg(E/\mathrm{eV}) + \ldots$

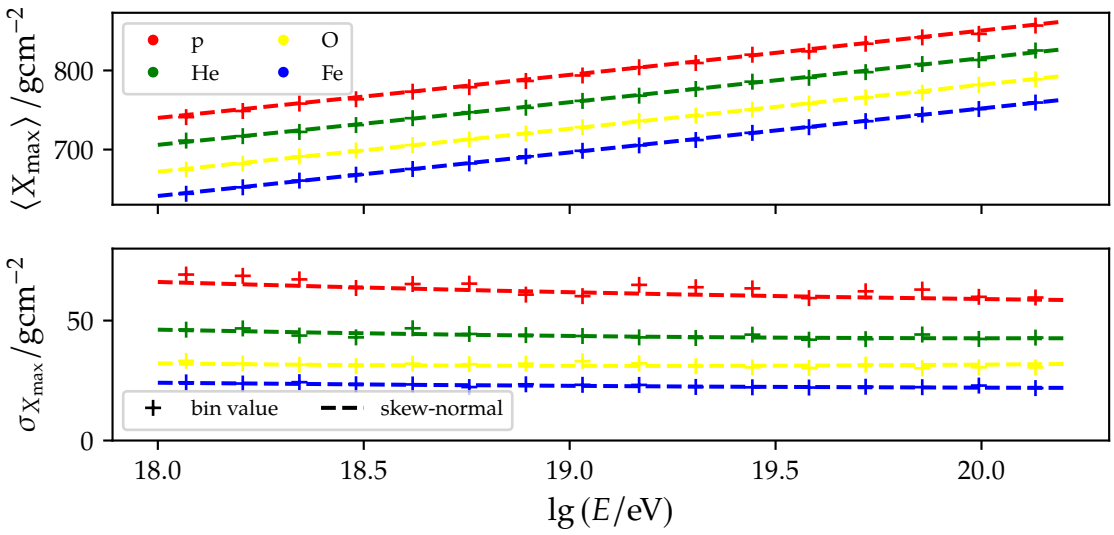


Figure 5.13: Energy dependence of the average value of X_{max} (*top*) and of shower-to-shower fluctuations (*bottom*) for different primaries of the QGSJetII-04 napoli library. We find a good agreement with the binned values for both, the averages and the standard deviations. The agreement is better for higher masses due to the increased penetration depth of lower ones. This explains the slight under prediction of fluctuations belonging to protons.

and variance

$$\sigma_x^2 = \sigma^2 \delta = \sigma^2 \left(1 - \frac{2\nu^2}{\pi(1+\nu^2)}\right) \tag{5.16}$$

are the analytic values for the skew normal distribution. In Fig. 5.13, we show the fit over the entire energy range for QGSJetII-04.

5.4.3 Muon content of a shower

Since the muon content of a shower on the ground level is closely connected to the primary particle, it is a prime target for all advanced reconstruction methods. This is true for both, the local and the global muon number. However, the number of muons and quantities derived from it, such as the muon signal, have a certain disadvantage. They are unbound and have a steeply falling PDF.

A Muon fraction at station-level

Since Offline saves the sub-component traces in simulations, we could directly take the muon signal as an indicator for the local muon density. Normalizing this to the total signal, we get the signal muon fraction

$$f_\mu^S = \frac{S_\mu}{S_{tot}}, \tag{5.17}$$

where S_μ is the total muon signal. However, since the baseline fluctuations for all sub-component traces are computed independently from each other, we sometimes obtain unphysical values for f_μ^S, such as muon fractions above one.

A more secure method of obtaining the fraction of muons is by counting the number of photo-electrons induced by different particles in the PMTs. Counting the responses in the

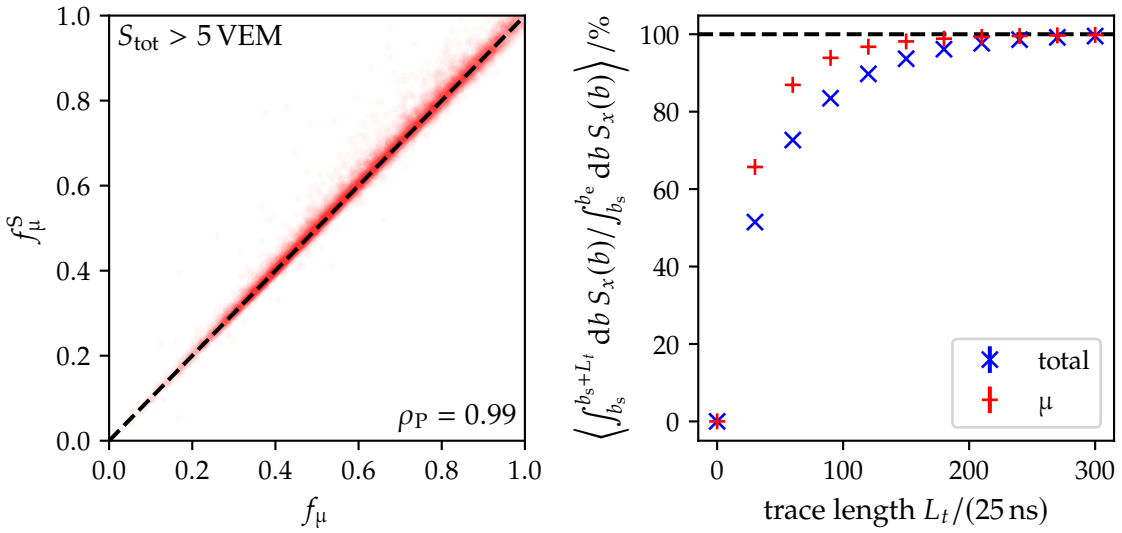


Figure 5.14: *Left*: Fraction of muon and total signal f_μ^{S} (see Eq. (5.17)) versus muon fraction f_μ computed from the photo-electron counts of the PMTs (see Eq. (5.18)) for not-high-gain-saturated stations satisfying $S_{\text{tot}} > 5\,\text{VEM}$. The dashed black line shows the identity relation. The linear correlation between both muon fractions is at 0.99. We cut all unphysical f_μ^{S} by restricting the plot area to the interval $[0, 1]$. The deviation between f_μ and f_μ^{S} is due to the independent baseline fluctuations of the sub-component traces. *Right*: Average ratio of the integral over the trace from start bin b_{s} to $b_{\text{s}} + L_{\text{t}}$ and total signal integral from b_{s} to b_{e} for different sub-trace lengths L_{t}. We used the same cuts as in the *left* figure. We compare the total trace (blue crosses) and the muon sub-component trace (red pluses). The muon signal converges more quickly to 100% since most of the muon signal is located in the beginning of the trace. If we want to compute the muon fraction from cut traces, we have to be careful. Even at a sub-trace length L_t of 200, we would over-predict the ratio of muonic signal to total signal.

PMTs, we get

$$f_\mu = \frac{\sum_i N_i \delta(i \text{ from } \mu^\mp)}{\sum_i N_i}, \tag{5.18}$$

where δ is the Kronecker delta that only equals one if the i-th bunch of photo-electrons comes from a muon. In practice, both, f_μ and f_μ^{S}, give us equivalent results (see Fig. 5.14).

A large part of the total muon signal comes from the beginning of the trace (see Fig. 5.14) since muons arrive at the SD stations almost unhindered. Therefore, we have to be careful when computing the muon fraction from integrals over sub-traces. By using too-small sub-trace lengths, we obtain muon signal that is too-large compared to the total signal.

B Relative muon content at event-level

Previous analyses have estimated the relative muon content R_μ (see Eq. (3.13)) via specialized simulations. By adding an increased number of virtual, off-grid stations (see Sec. 5.1), the muon signal on the ground can be approximated. However, such specialized simulations are time-consuming. Depending on how many stations are added, they are usually much longer than the simulation of the event itself.

Here, we propose a different method: Instead of taking the signal estimations, we rely only on the raw CORSIKA data. In the `.long` files (see Sec. 2.2.6), there is a field of the particle numbers at different shower depths. For each SD event simulated from the corresponding `.part` file, we use these numbers at ground level as a prior value for N_μ. We claim that

Table 5.7: Fit parameters for the fit model defined in Eq. (5.19) for the local mirror of the Napoli library. The values in the brackets are the leading digit errors. Due to the smallness of the errors, we assume that the fit models represent the average behavior well. The results for both hadronic interaction models QGSJetII-04 and EPOS-LHC only differ marginally for the energy dependency. However, there is a clear difference in the zenith dependency. This is most likely an effect of the difference in attenuation for both models.

Model	$\lg N_0^{\mathrm{p}}$	γ	a	b	c
QGSJetII-04	6.710(2)	0.945(2)	1.172(1)	−1.064(4)	−1.496(19)
EPOS-LHC	6.750(2)	0.947(2)	1.196(2)	−1.085(5)	−1.707(22)

this number is proportional to the actual number of muons. The conversion factor might dependent on θ and E, which is removed by the normalization to the proton R_μ.

To get R_μ, we need to estimate $\left\langle N_\mu^{\mathrm{p}} \right\rangle$ (see Eq. (3.13)). Since the attenuation of the muons of the shower is universal, we assume that we can disentangle the average proton muon number in an energy- and a zenith-dependent part via

$$\left\langle N_\mu^{\mathrm{p}} \right\rangle = N_0^{\mathrm{p}} f_\mu^{\mathrm{p}}(\theta) \left(\frac{E}{10^{18}\,\mathrm{eV}} \right)^{\gamma} = f_\mu^{\mathrm{p}}(\theta) \left\langle N_\mu^{\mathrm{p}}(E) \right\rangle , \tag{5.19}$$

where N_0 and γ are fit parameters. The choice of 10^{18} eV is arbitrary. This procedure is very similar to the Heitler-Matthews model (see Eq. (2.10)). To capture the zenith dependence, we use a CIC-like (see Eq. (3.10)) function[9]

$$f_\mu^{\mathrm{p}}(\theta) = a + bx + cx^2, \tag{5.20}$$

where $x = \sin^2\theta - \sin^2\theta_{\mathrm{ref}}$. We use $\theta_{\mathrm{ref}} = 38°$ as in Eq. (3.10). The $\left\langle N_\mu^{\mathrm{p}}(E) \right\rangle$ is dependent on the used zenith range. However, this dependence is caught by the a fit parameter of Eq. (5.20).

In Table 5.7, we show the fit values for the parametrization. The fit is a simple least square fit. Although Eq. (5.19) is purely phenomenological, it reasonably reproduces the average behavior (see Fig. 5.15). After calibration, the muon number is almost independent in energy for zenith bins (see Fig. 5.15). We conclude that it is sufficient to use this calibrated quantity to estimate the zenith dependence f_μ^{p}. In Fig. 5.16, we show the fit versus the average binned values. Again, the fit reproduces the behavior very well; errors on the fit parameters are negligible (see Table 5.7). After full calibration, R_μ shows a good separation for proton and iron values (see Fig. 5.16). We have attached the results for EPOS-LHC in Fig. D.2 and Fig. D.3.

[9] However, we allow for $f_\mu^{\mathrm{p}}(\theta_{\mathrm{ref}}) \neq 1$.

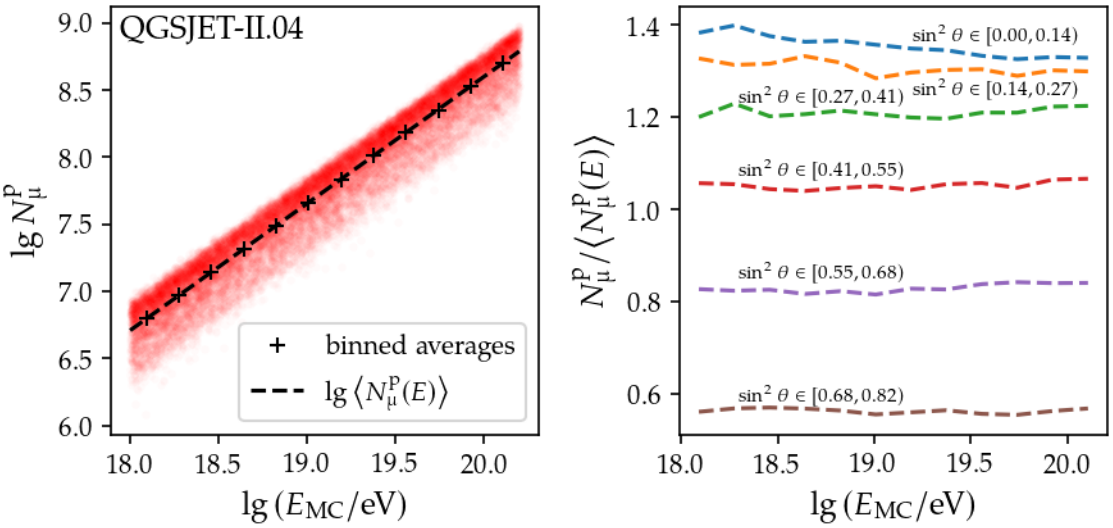


Figure 5.15: Energy dependence of the number of muons (μ^+ and μ^-) for proton CORSIKA files simulated with QGSJetII-04 of the Napoli library before (*left*) and after (*right*) removal of the energy dependence. We use the logarithmic N_μ^P due to the uniform distribution in the energy. The energy dependence is in good agreement with a linear model. After the removal, the dependence is (almost) flat for different zenith intervals.

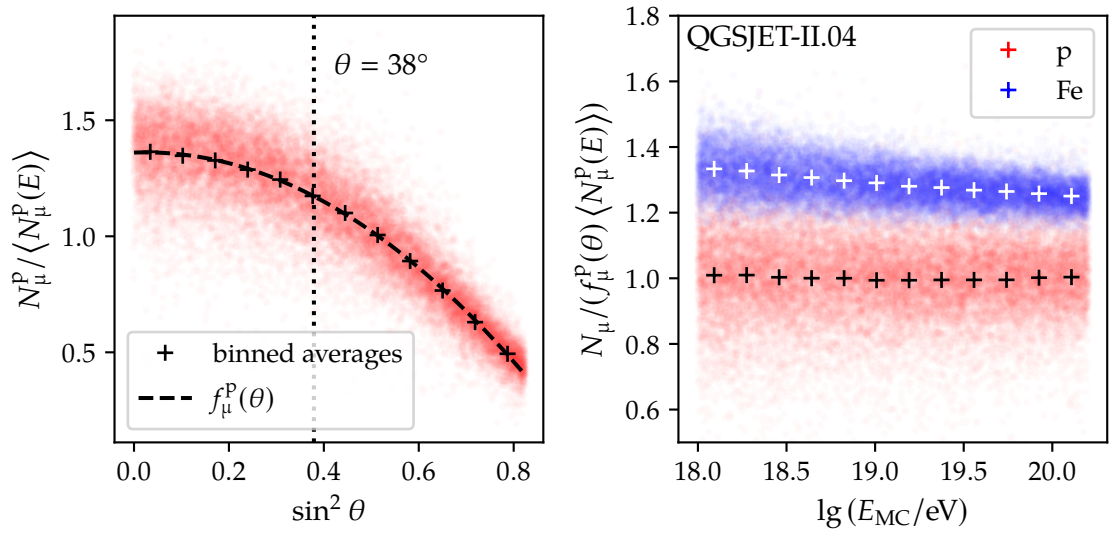


Figure 5.16: Zenith dependence of the number of muons divided by the expected number at the energy for proton CORSIKA files simulated with QGSJetII-04 of the Napoli library (*left*) and resulting R_μ for proton and iron of the same dataset (*right*). Again, we find a good agreement with the proposed model in Eq. (5.20). Using the zenith dependence, we can compute R_μ for the proton and iron showers. We find a visible separation.

6 Extraction of Local Shower Properties from SD Station Signals



Imagination was given to man to compensate him for what he is not; a sense of humor to console him for what he is.

(Francis Bacon, "father of empiricism")

DALL·E 2 prompt:

Screaming mad scientist on a bridge, [M]unch[.]

Analyses by the Auger collaboration utilize event-level observables to reconstruct important properties of air showers. However, the information contained in the responses of individual detector stations is not usually considered as such but only as part of the ensemble of detectors triggered by the overall event. It is a legitimate research question in what ways station-level information could be used to improve our understanding of air showers.ch way station-level information could be used to improve our understanding of air showers. Studies of this kind have been performed in [A:9, A:10, A:11, A:12]. Furthermore, it has been shown in [A:20, A:21] that using single station information can benefit the reconstruction of air-shower events.

We define local shower properties as quantities and observables, which directly belong to a spatially localized measurement of the shower cascade[1]. The time-dependent signal of a PMT (trace), for example, which is spatially localized measurement of the temporal distribution of shower particles, can be categorized as a local shower property. Time traces are an ideal candidate for machine learning-based analyses, because they are highly complex one-dimensional signals. Currently they are used in classical analysis approaches to extract a different observables and features, such as the integrated, total signal S_{tot}, and the empirically motivated risetime, $t_{1/2}$, to estimate the depth of the shower maximum (see Sec. 3.3.1). One of the most interesting local observable, however, is the signal produced only by the muonic sub-component of the shower (see Sec. 4.3.1), since it is directly dependent on the nuclear mass of the primary particle (see Eq. (2.10)). Therefore, we focus in this chapter on examining the local muon signal S_μ of a detector.

Machine learning is one of the most emerging disciplines of high-energy physics, and thus also for Auger. To test for consistency with previous works, we first focus on the reproduction of similar NN based results (see [A:9, A:12]) and compare the analyses with reference models, which use fewer free parameters than a standard NN. This allows us to better judge the

[1] Local shower properties are not necessarily independent of the underlying shower.

quality of the NN for reconstruction purposes. In addition, we want to examine the added value of the UUB data and SSD trace information for the analysis of the time traces.

Note that, in this chapter, the short notation $\mathcal{E}$ for the logarithmic energy $\lg(E/\mathrm{eV})$ is used (see Eq. (4.31)). This is primarily done because of spacing issues, e.g., Fig. 6.1. In this way, we are able to keep the terminology consistent. We switch back to the long form of $\mathcal{E}$ in Sec. 6.3.

First, we define reference models in Sec. 6.1, which are going to be compared to the NN based approach described in Sec. 4.3.1.A. In Sec. 6.2, the results of [P:107] are examined. Moreover, the added value of additional free parameters of the model is examined using previously defined reference models. The NN based approaches are tested for their overall potential and for the most promising case of application. In Sec. 6.3, instead of using solely scalar inputs, we use a part of the time traces as an input for a NN and examine the performance. In the course of this study, additional information provided by the SSD are used to optimize the prediction of the muon component signal. Finally, in Sec. 6.4 a preliminary study of feasibility is performed on the application of NN based approaches for the counting of muons passing trough the UMD.

6.1 Reference models for local shower analysis

The main focus of this chapter lies on the extraction of the muon signal S_μ from local shower parameters. To evaluate the performance of the NN models, their predictions are compared against models that do not use NN based inference, which are called reference models. The same inputs as defined in [A:9] are used to make the models comparable to the network defined in this work.

6.1.1 Inputs and outputs for the station-level analysis

A fraction f_μ of each signal deposited in a WCD[2] is due to muons traversing the water (see Sec. 5.4.3.A). We consider all other sub-components contributing to the signal as noise of the muon signal. As a consequence, all models utilized in this section produce only the muon signal S_μ or the muon fraction f_μ as a single output.

The total muon signal can be computed in two ways. The more straightforward way involves the time signal traces of the muon sub-component. Using the sub-component, the total muon signal is given by

$$S_\mu^\mathrm{s} = \int_{b_\mathrm{s}}^{b_\mathrm{e}} \mathrm{d}b\, S_\mu(b)\,, \tag{6.1}$$

where b_s and b_e are the start and the end bin, respectively. However, since the baseline fluctuations in $\overline{\mathrm{Off}}\underline{\mathrm{line}}$ for each sub-component are independently computed, the sum of the sub-components does not equal the total signal. If we want to account for the baseline fluctuations more appropriately [A:22], we have to use the indirect way given in Eq. (5.18). Hence, we define S_μ via

$$S_\mu = f_\mu S_\mathrm{tot}, \tag{6.2}$$

using the total signal S_tot and f_μ computed via counting the photo electrons.

In this analysis, the same nine input parameters are used as defined in [P:107]. They have been already discussed in Sec. 4.3.1.A. In Fig. 6.1, the individual parameters are summarized and examined in the form of a linear correlations plot for the simulation data set, with and without quality cuts (see Sec. 6.2.1). The Pearson correlation coefficient ρ_P (see Eq. (4.43)) is

[2] Indeed, this fraction can be zero. However these cases are rare in the investigated phase space.

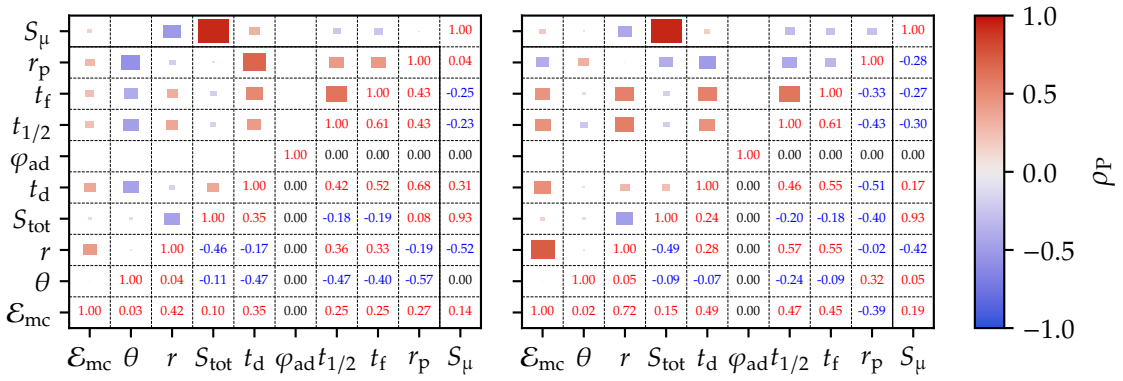


Figure 6.1: Pearson correlation coefficient (see Eq. (4.43)) with an absolute value above 10^{-3} for all possible combinations of the inputs and outputs of Sec. 6.1.1. The left plot shows correlations for the entire data set without cuts and the right plot for the cuts made in [P:107] (see Sec. 6.2.1). The color-coded boxes are just representations of this coefficient. They scale with its absolute value. Except of φ_{ad} and θ all input-parameters are linearly (anti-)correlated with the output for the cut data set.

shown for all combinations of input and output parameters, individually. Surprisingly, φ_{ad} and θ seem to have almost no linear correlation with the other parameters. All other input parameters show at least a slight correlation. With a wide margin, S_{tot} is most correlated with S_μ in both cases, the total data set and the data set with quality criteria applied. This is not surprising, because S_μ is on average a large part of the total signal, independently of the event-level or station-level values of f_μ.

Each cosmic ray shower with a high-enough energy ($E \gtrsim 3\,\mathrm{EeV}$) reliably triggers multiple of the WCDs of the SD. The average number of triggered stations increases with energy (see Fig. 5.2). Therefore, less station-level inputs are available for lower energies. This potentially impacts the training and inference process of the NN by implicitly underweighting traces of events, in which data from only few detectors are available.

6.1.2 Reference models

In this section the reference models to compare NN based analyses against are briefly introduced. The naming convention for input parameters given in Eq. (4.3) is followed. In the case of extracting the muon fraction, X is the vector of one set of standardized input values corresponding to the simulated muon signal S_μ. To remain comparable to the NN based analyses of [A:9, A:10], the muon signal is standardized according to the method presented in Eq. (4.33); the standardized muon signal is denoted as s_μ. The models are tuned to predict s_μ. The predicted standardized muon signal are written as s_μ^p. The relation between the total predicted muon signal S_μ^p and the standardized muon signal is given by,

$$S_\mu^p = \sigma_{tr}\left(S_\mu\right) s_\mu^p + \left\langle S_\mu \right\rangle_{tr}, \tag{6.3}$$

where the average muon signal $\langle S_\mu \rangle$ and standard deviation of the muon signals $\sigma_{tr}(S_\mu)$ are taken from the training data set, and therefore indexed *tr*.

The simplest alternative model is a constant predictor without any input parameters. By averaging the test set, an unbiased prediction for the muon signal in the whole data set can be achieved. In case a model to estimate the (standardized) muon signal performs worse than the constant predictor, it has to be excluded.

Furthermore, the total signal is highly correlated with the muon signal (see Fig. 6.1). The constant-predictor reference model can therefore naturally be extended by adding powers of S_{tot} as an input for a NN. Instead of a constant, we obtain a polynomial in terms of S_{tot}. We fix its degree with help of the validation set (see Sec. 6.2.1).

To access the "amount of linear information" in the input data, we use a purely linear model. Thereby, all of the inputs defined in Sec. 6.1.1 are used in a least-square fit. This leaves us with ten parameters. Such a model tells us what can be learned alone from the other less important parameters. Finally, we use an extension of this linear model. The first layers of dense neural networks mix their inputs in such a way that 'good' features are favorably weighted. Hence, during training, it automatically creates new, better suitable inputs. We want to emulate this behavior to a certain degree. As additional inputs for our least squares fit, we use all unique multiplications between each of the inputs[3] (see Sec. 5.3.1.B). Hence, we generate polynomial features up to a certain degree. This increases the number[4] of coefficients to 55. In the following, we call this the quadratic model. We do not introduce ways of restricting the fits in any way to reduce possible fine-tuning biases. We expect that the performance of these models increases with the number of available parameters.

6.1.3 Performance of reference models

In this section, the performance of the reference models to compare the NN analyses against is evaluated. For this purpose, the split strategy from Sec. 6.2.1 is applied with the help of the metrics defined in Sec. 4.4.1. A lower value according to one of the defined metrics directly is associated with a better performance of the model. The results of the models for a complete training are tabulated in Table 6.1. Results are obtained by fitting each model to 20 different splits with a training size of 22 000 and a validation fraction of 0.1.

The distribution of signals S_μ is of similar shape to the distribution of all total signals S_{tot}: both distributions show a sharp peak with an exponential tail. Therefore, the predictions of the constant predictor[5] are quite poor and cannot compete with one of the other models. All models that outperform the constant predictor contain at least some sort of information about S_μ in their output. The model that receives expansion in powers of S_{tot} as inputs works acceptably. The maximum power of S_{tot} for each of the splits is fixed separately by fitting polynomials of the degrees 2 to 9 and using the best performing one. Using 4 as the maximum power proved to be sufficient. Adding higher powers of S_{tot} does not notably improve the performance. In Fig. 6.2, the performance (see Sec. 4.4.1) is depicted for all reference models as a function of the available size of the training data. For small training data sets, the S_{tot} expansion model is highly unstable.

Adding different parameters acts as a regularization procedure. The linear model does not only improves the result, but it is also more stable with respect to the S_{tot} power expansion model (see Fig. 6.2) and only few more free parameters are required. Nevertheless, for bad choices of training data it is expected to experience instabilities for the extreme values of the parameters space of the available input. Fortunately, such instabilities do not affect the global outcome of this study. As expected, the performance improves significantly by employing the quadratic model.

In the following, we focus on the last three models in depth, since they show reasonable results and individually outperform the constant predictor model.

[3] If we would have a, b, and c: we would additionally get $a^2, ab, ac, b^2, \ldots$ as inputs.

[4] There is one parameter for the constant, nine for the linear model and $45 = 9 + 9 \cdot 8/2$ for all unique input multiplications.

[5] This would not be the case if done when f_μ would be used.

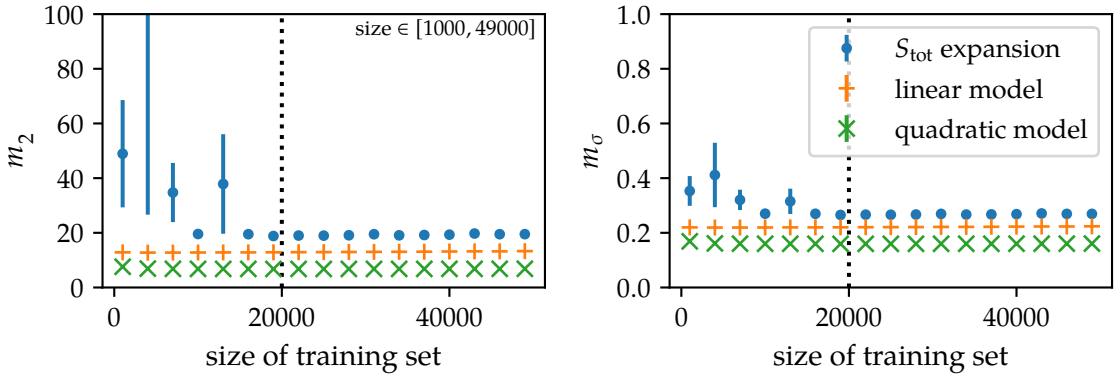


Figure 6.2: Dependence of m_2 (*left*, see Eq. (4.40)) and m_σ (*right*, see Eq. (4.42)) on the size of the training set for the fit-dependent models of Sec. 6.1. For each point twenty different training sets have been sampled. All of the tested models perform on a wide range of training size better than a constant predictor (see Table 6.1). At about $N_{\mathrm{tr}} = 20000$ the models saturate. Only the S_{tot} expansion seems slightly unstable after this point. However, these effects are minor.

Table 6.1: Values of comparison metrics (see Sec. 4.4.1) computed on QGSJetII-04 test sets for a regular fit of the reference models (see Sec. 6.1.2). The values are obtained by splitting the data set 20 times for a training size of 22 000 and averaging over the output of the resulting models. The maximum order of the S_{tot} expansion is fixed by taking the best result of polynomials from orders 2 to 9. As expected, the more we use the entire set of input parameters in different representations, the better the results.

data set	model	coefficients	m_2	m_R	m_σ
QGSJetII-04	constant	1	142.515	-0.618	0.732
QGSJetII-04	S_{tot} expansion	5	19.003	0.069	0.266
QGSJetII-04	linear	10	12.914	0.065	0.221
QGSJetII-04	quadratic	55	6.777	0.032	0.160

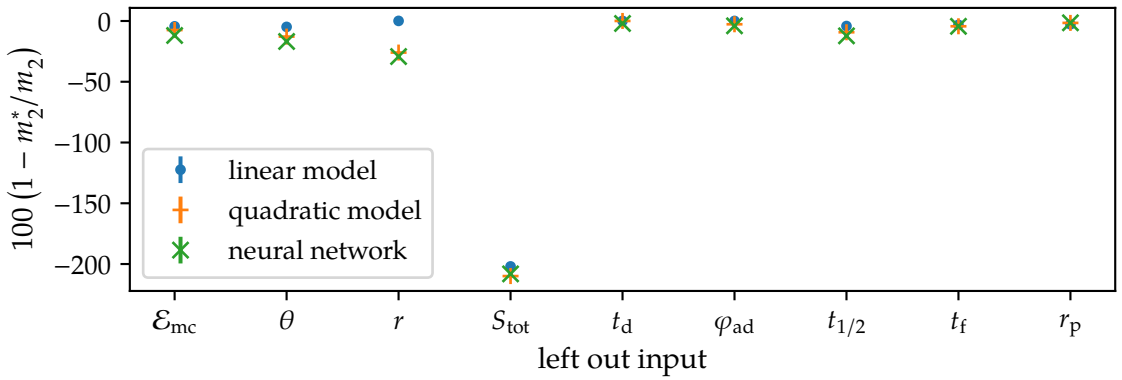


Figure 6.3: Change of value of m_2 for linear, quadratic, and neural network model because of the removal of one input parameter. Thereby, m_2^* is the value of the metric without the input on the x-axis. As expected from Fig. 6.1 the total signal S_{tot} plays a central role in the predictions. Ignoring it reduces the performance of all models. Furthermore, the quadratic and neural network model behave very similar. For each datapoint they are almost on top of each other.

6.1.4 Importance of the input features on the prediction

The performance of an estimator or a model is strongly dependent on the quality of its input parameters and the input data. Triggered by Fig. 6.1, we therefore analyze the dependence of the models on the set of the available input parameters. We only study the linear, quadratic, and – for completeness – the NN model by fitting and training them on the regular splits (see Sec. 6.2.1). While doing this, however, we exclude one of the inputs and study the effect on the global value of m_2 (see Fig. 6.3). Consequentially, in case of the quadratic model, all higher-order terms containing the particular input parameter are removed as well. As can be seen already in Fig. 6.1, the total signal S_{tot} plays a central role for every model. Discarding it, dramatically reduces the performance of all models equally. Furthermore, other inputs such as θ, have a non vanishing influence, even though their contribution is relatively small compared to S_{tot}. It is noticeable that the quadratic and neural network model behave very similar.

Therefore, we analyze the inputs of the quadratic model[6] in Fig. 6.4. This time we turn off one of the 55 inputs and look at those which change the metrics value by 3%. The result is similar to that in Fig. 6.3 for the linear contributions (first 10 values). From the higher order inputs only two seem to be relevant in this analysis. Again, both are connected to S_{tot}. We conclude that most of the higher order terms are not contributing and should under normal circumstances be left out. Furthermore, we expect that higher order fits - if they are stable - should improve the performance even further.

6.1.5 Stability of model parameters

To assess the stability of the linear and quadratic models, we analyze their coefficients when fit in different parts of the phase space. We perform this extra step to check if they exhibit an energy dependence or if the coefficients themselves are more or less universal. If the latter is true, we could these models with a fit to low energy data from AMIGA or the UMD and circumvent most of the problems we are facing in our simulations (see Sec. 2.4.1).

For this study, we train the linear and quadratic model with training data taken from the three MC energy intervals $(18.5, 19.0]$, $(19.0, 19.5]$, and $(19.5, 20.0]$. Again, for each interval,

[6] This is not possible for the neural network.

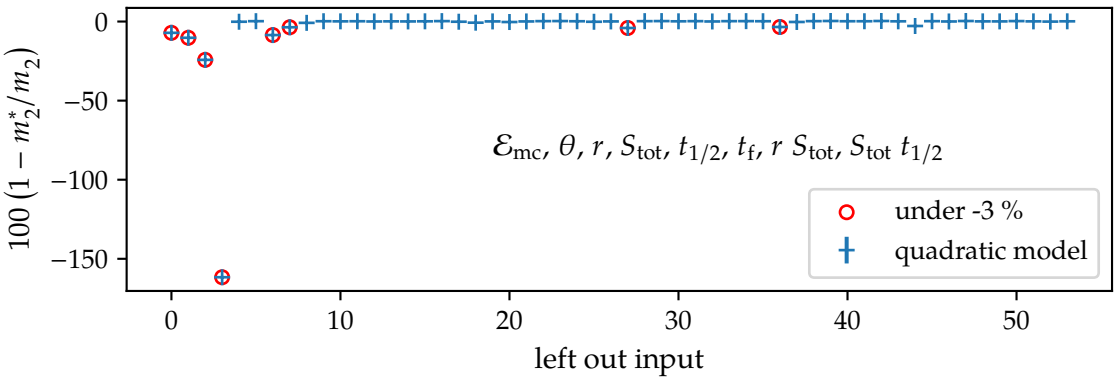


Figure 6.4: Change of value of m_2 of the quadratic model because of the removal of one input component. The components' names which decrease the metrics value by 3% are given in the center of the plot. Their corresponding datapoints are marked by a red circle. Due to fluctuations, sometimes removing a component yields a slightly better result. As expected from Fig. 6.3 components connected to the total signal are very important for the prediction of the quadratic model.

we take 20 test splits to account for fluctuations. For each of the splits, we obtain a unique value for the coefficient. We use the average of these values. The linear model shows (see Fig. 6.5) that there is, in fact, an energy dependency. However, comparing the coefficients using the entire energy range and the different intervals shows us that for almost every coefficient, the second energy bin has a similar value to that of the entire energy range. It is possible that we could correct the behavior in some way. Especially since some of the coefficient changes seem to be almost linear. The quadratic model shows a much stronger dependence on the energy interval (see Fig. 6.6). For the most important parameters, such as the total signal, there is a visible spread. We expect that here the situation is much more complicated, and a simple "upscaling" would take a much more detailed analysis. In the case such a study would be performed, we recommend using the linear model as a primer or even going back to the expansion in total signal S_{tot}.

6.2 Extraction of total muon signal and comparison to baseline models

In the following, we want to complement the analysis done in [P:107]. We begin by reproducing their results to ensure that we use the same setup. Afterward, we study the networks and compare them to the baseline models defined in Sec. 6.1. Our main goal is to find out if the increase in complexity using a NN-based approach is significant. Without comparing the NN with simpler or conceptually different models, we are unable to fully appraise their predictive power [C:25]. Hence, this approach will put the NN results into context. In this case, this is comparably easy since the used architecture is quite simple, having only around 2000 free parameters.

To get a similar number of samples as in [P:107], we use two of them from the old QGSJetII-04 data set (see Sec. 5.1.1). This could yield potential problems with shower degeneracy and, as a consequence, unexpected behavior. To exclude this, we use only unique showers in our EPOS-LHC data set (see Table 6.2). It acts primarily as a cross-test which is not used in the fitting/training steps. Additionally, this also reduces the problem of the non-uniform energy distribution of the WCD events (see Fig. 6.7).

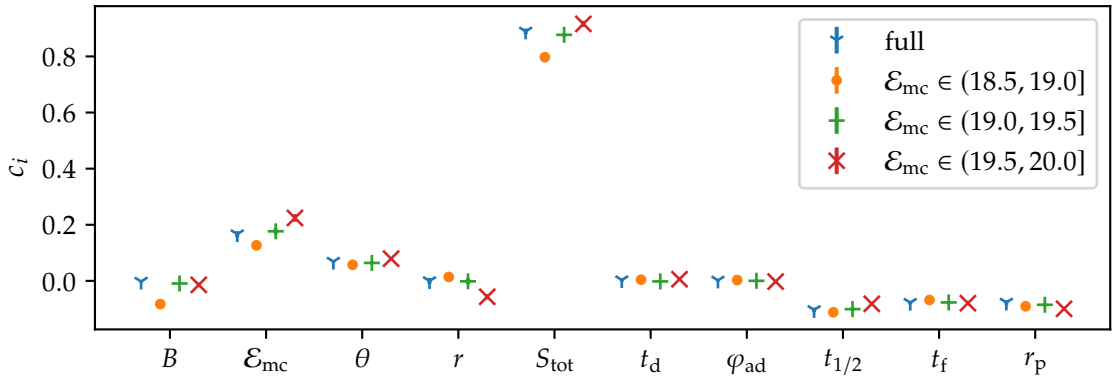


Figure 6.5: Average value of coefficients of linear model when trained in various energy intervals for 20 different splits of the QGSJetII-04 data set. The x-axis label B denotes the intercept of the linear fit. The fit coefficients show a dependence on energy. Using the full energy range yields almost the same coefficient than the second energy range.

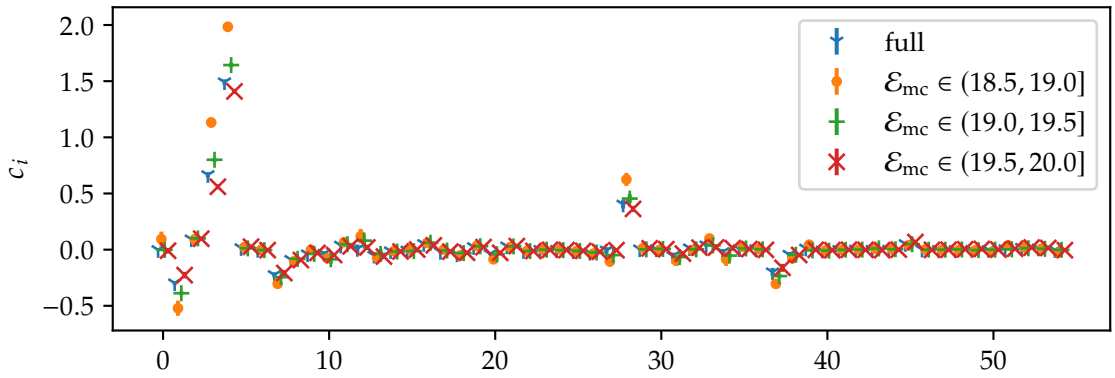


Figure 6.6: Average value of coefficients of quadratic model when trained in various energy intervals for 20 different splits of the QGSJetII-04 data set. As in Fig. 6.5 there is a clear dependence on energy for the fit coefficients. Unfortunately, it is much stronger than in the linear model.

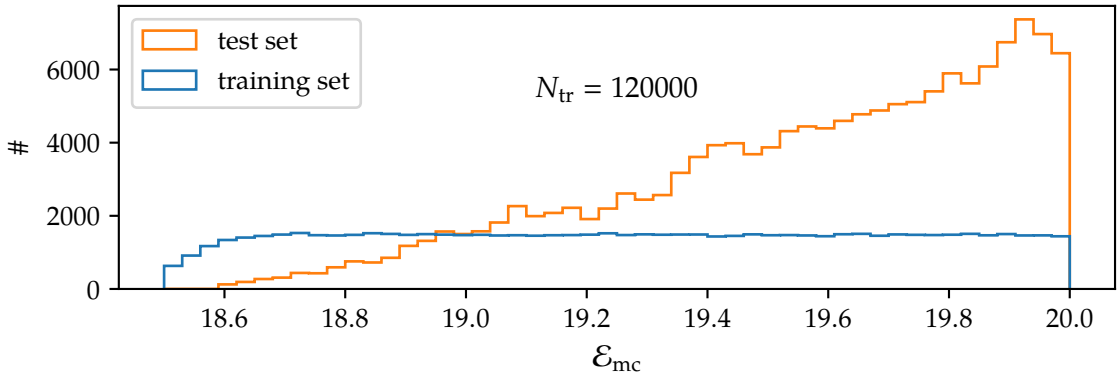


Figure 6.7: Energy distributions of test and training set for the QGSJetII-04 data set. The deviations from the uniform distribution of the training set are due to the sampling procedure. It uses a different bin size. The scarcity at lower energies is due to efficiency of the array. Showers are distributed uniformly in energy. However, the number of detectors triggered depends on the energy. The non-linear part in the lower energies is caused by the non-100% efficiency of detector array in this region.

Table 6.2: Summary of the data sets used and loss due to the cuts defined in Sec. 6.2.1. For QGSJetII-04 we work with two simulations of one shower and for EPOS-LHC with only one. CORSIKA files are from the old NapLib. The cuts remove almost 80% of the data. Furthermore, only about 10% of the combined detector signal remains. This is a consequence of the implicit cuts on the distance to the shower axis. All detectors near the shower core are discarded.

had. model	events	WCD data		data loss	
		before cuts	after cuts	amount / %	signal / %
QGSJetII-04	116123*	1208091	264282	78.12	89.22
EPOS-LHC	59407	626791	138533	77.90	89.18

* This is not to be confused with the number of showers.

6.2.1 Quality cuts, data transformation, and split strategy

We use the same quality cuts as proposed in [P:107]. First, we limit $\mathcal{E}_{mc}$ to the interval $[18.5, 20.0]$. Secondly, only stations in events with inclination angles below 45° and stations with $S_{tot} > 10\,\text{VEM}$ are used. Lastly, we remove all low- and high-gain saturated stations from the data set. All cuts are performed via MC parameters. Unlike [A:10], we limit ourselves to simulated data because we want to have full control and perfect information for this study.

The cuts are extensive and reduce the data set significantly (see Table 6.2). Over 70% of the data is lost. In Fig. 6.8, we illustrate the impact on the distributions of the two linearly most important input parameters (see Fig. 6.1). For both parameter combinations, the complexity of the distribution is reduced. In the case of the muon fraction plots, it almost transforms to a peaked distribution centering around 0.5. In Fig. 6.9, the projection of the 2D distributions on the y-Axis is shown. We find a peaked distribution for all of the different primaries.

Before feeding the inputs and output in the models for the training process, we standardized them (see Sec. 5.3.1). An advantage of this is that the transformation takes care of the units of the inputs.

Similar to [P:107], we get the training and validation set by the following procedure: For

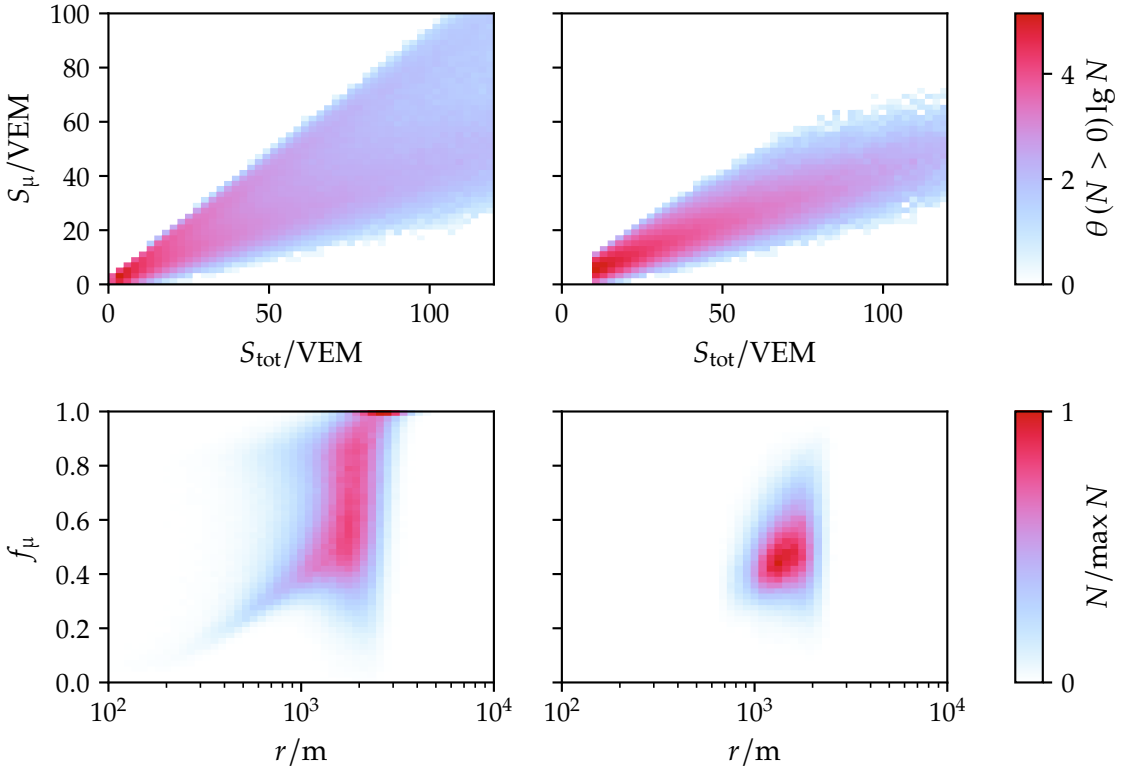


Figure 6.8: Impact of the cuts defined in Sec. 6.2.1 on the two-dimensional distributions of S_μ and S_{tot} in logarithmic scale (*top*) and f_μ and r in linear scale (*bottom*). The plots on the left side show the full QGSJetII-04 data set (see Table 6.2). The other exhibit the distributions after the cutting process. In both cases the cuts reduce the complexity. The spread of S_μ is greatly reduced. Since S_{tot} is the most person-correlated input parameter (see Fig. 6.1), we assume that this increases the predictive powers of models. Moreover, the cuts break the degeneracy for low values of f_μ. Only a localized distribution around $f_\mu \approx 0.5$ remains. Instead of the muon fraction defined in Eq. (6.1) f_μ is here computed via counting of the photoelectrons in the PMTs. We use this quantity because we believe that it is actually the right way to define the fraction. However, both quantities differ only marginally. We do not expect any loss in generality by choosing either.

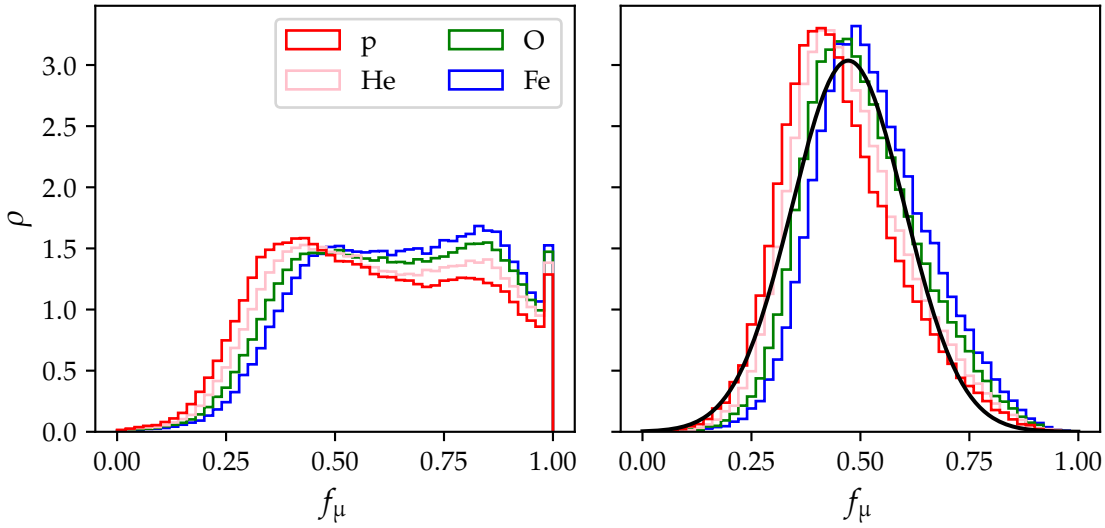


Figure 6.9: Distribution of f_μ for the different primaries of the used data set before (*left*) and after (*right*) applying the cuts. The y-axis shows some arbitrary density. The black line in the left plot is from a Gaussian fit to the sum of all component distributions. The Gaussian is centered around 0.472 and has a spread of 0.129.

Table 6.3: Number of samples in the different splits of the old NapLib data set divided into the primaries used. The primaries are not equally distributed. This is due to the energy sampling algorithm (see Sec. 6.2.1) and should not cause any loss in generality. The asymmetry in total number of primaries is due to their different masses.

hadronic model	primary	fit	validation	test	total
QGSJetII-04	p	18075	11925	32922	62922
QGSJetII-04	He	18102	11898	34911	64911
QGSJetII-04	O	17856	12144	37744	67744
QGSJetII-04	Fe	17967	12033	38705	68705
		72000	48000	144282	264282

each primary in the data, we draw random values in such a way that, in the end, they have a flat energy distribution[7] in the logarithmic energy interval [18.5, 20.0] (see Fig. 6.7). Therefore, we undertrain our models slightly near to 18.5. Fortunately, this gives us the advantage of being able to estimate errors due to irregularities in the training distributions.

We set the size of the TrDs N_{tr} to 120 000. The TeDs consist of the remaining[8] data samples N_{te}. Like in [P:107], we use 72 000 data points for the training itself, leaving us with 48 000 data points for the VaDs corresponding to a validation fraction of 40%. To ensure that this is sufficient for the models defined in Sec. 6.1, we train with different N_{tr} and compare the global results (see Fig. 6.2). We find that at a size of about 20 000 training samples, this stability requirement is met and the result saturates. Therefore, we use in this section a reduced training set size for further studies to reduce computational time. Later from Sec. 6.2.3 on, we use the real training size of 120 000. In Table 6.3, the splits of the data set are shown.

[7] We split the interval into different energy bins and just fill them (randomly).
[8] There are no intersections between test and trainings set.

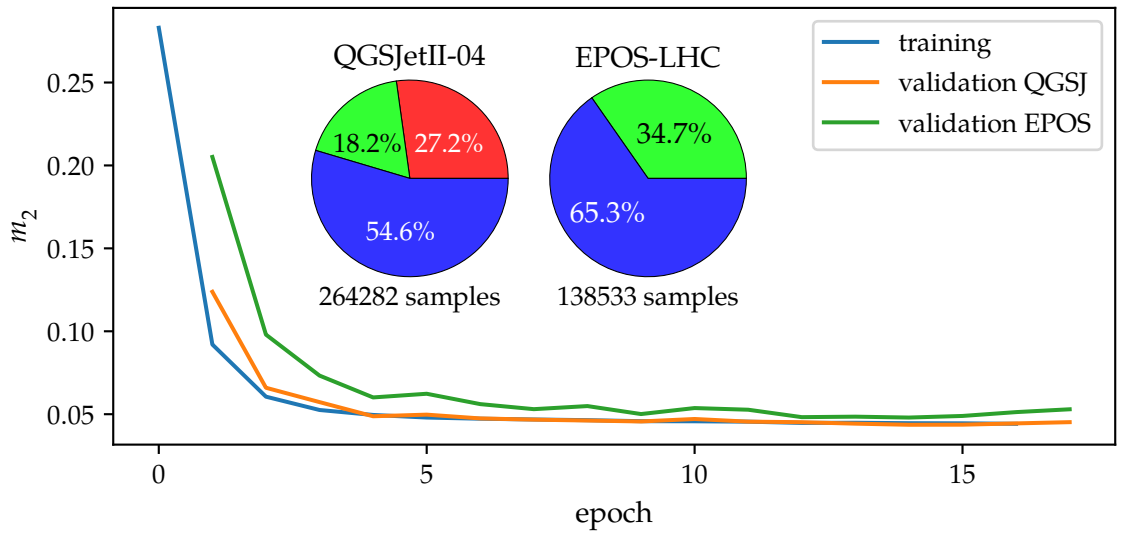


Figure 6.10: Development of the loss value during the training process for the training and the two validation sets. We moved the validation datapoints of the validation sets one epoch, because they are computed at the end of an epoch. For completeness, we added two pie charts displaying the relative amount of data in each of our data sets. The red, green, and blue slice correspond to the TrDs, VaDs, and TeDs, respectively. The number in the slices is the percentage of the total data set. Since, we train on the QGSJetII-04 data set we do not have training data for the EPOS-LHC data set. As mentioned in Sec. 6.2.2 the value of m_2 is computed from the standardized inputs. Therefore, there is a large difference to the values in Table 6.1. Since there are no extreme changes or outliers, we expect that the training procedure is valid and the network has not been overfitted.

6.2.2 Training procedure

Since in [P:107] there are no mentions of a special training procedures. Therefore, we keep the process as simple as possible. The network is trained batch-wise (batch size $N_b = 32$) for a maximum number of 100 epochs. We use MSE as the target loss function. Again, we use the standardized values of S_μ. Therefore, the values of m_2 are slightly different from that in Sec. 6.1.3. To prevent overfitting, we stop the training if one of two conditions are met: First, when the loss computed on the VaDs does not decrease after 3 epochs. In this case, we would overfit, finetuning the weights η of the network to the TrDs. Secondly, when the loss computed on the training set does not decrease after 3 epochs. We see this as a cross-check if something unexpected happens. Almost all training stops are due to the first condition. In addition to a regular VaDs, we use a second VaDs from the old EPOS-LHC data set. Doing this consistently yields a stable training process (see Fig. 6.10) in which the NN does not seem to do something unexpected.

6.2.3 Verification of network

Most probably, even if our training set has a very similar structure to that described in [P:107], we are not able to reproduce exactly the same results of [P:107]. We expect a landscape of minima in the solution space. It is unlikely that we would land in the same one with a different data set. Another complication is that we cannot make a one-to-one comparison because the weights are not published. Nevertheless, it is sufficient to show comparable results of averaged binned values which achieve similar performance. Moreover, in this way, we see if the architecture works on other data sets.

For the verification, we use the same two types of binned bias and relative error plots utilized in [P:107]. Like there, we show them for r, $\mathcal{E}_{\mathrm{MC}}$, $\sec\theta$, and S_{tot} for both the QGSJetII-04 and EPOS-LHC test data set. The distributions of those quantities are shown in Fig. 6.11. Only for $\sec\theta$ and S_{tot} the shapes of the fit and test set are similar. For the other parameters, we encounter deviating distributions due to the energy dependence of the number of triggered stations at a certain distance. We want to use a uniform binning in logarithmic energy to ensure that stations corresponding to low energies are still represented in our training data. Furthermore, a perfect predictor should not strongly depend on the used input distribution.

In Figs. 6.12 to 6.15, the bias for each of the primaries in the data set is shown. One difference compared to the original work is that we use Oxygen instead of Nitrogen. In all cases, we see that the bias and relative error in the different particle types diverges in some bins. Therefore, we have a certain degree of dependence on the mass in the network. We find that the network predictions for heavier components correlate to higher bias values and relative errors. Moreover, comparing Figs. 6.12 to 6.15 with Fig. 6.11 we see that in regions with little training data the output of the NN becomes much worse. Especially, the S_{tot} plot in Fig. 6.15 displays this behavior well. Both QGSJetII-04 and EPOS-LHC do not show large deviations. Although the network performs slightly worse on EPOS-LHC, we conclude that the NN approach for local information is almost hadronic model independent. This is - most likely - due to the heavy reliance on (directly)[9] shower-independent quantities like S_{tot}, which could also be obtained from pure tank simulations.

Apart from some minor differences, we consistently obtain similar results to that described in [P:107]. This verifies the used approach.

6.2.4 Comparison of reference models and neural network

We split the comparison study between the reference models defined in Sec. 6.1 and the NN from Sec. 6.2.2 into two/three parts. First, we do a back-to-back comparison of binned biases and relative errors like in Sec. 6.2.3. This gives us an overall feeling, if there are direct differences. Secondly, we compare other important quantities like the resolution and the spread in the bins. Finally, we investigate how the network probably works.

A Bias and relative error

In Sec. 6.2.3, we have seen that the predictions of the NN do only marginally depend on the primary particle mass. Therefore, we do not split up our QGSJetII-04 and EPOS-LHC test sets for the following analysis. As mentioned in Sec. 6.1, we compare the NN only to the expansion in terms of S_{tot}, the linear model, and the quadratic model. It is reasonable to use the same approach as in Sec. 6.2.3. However, this time we check only the results of r, $\mathcal{E}_{\mathrm{mc}}$, and S_{tot} (see Figs. 6.16 to 6.18). We moved the result of θ into the appendix because it shows almost no interesting behavior (see Fig. D.5). Moreover, we added a S_{μ} plot in the appendix (see Fig. D.6).

All bias and relative error plots show that in most cases, the S_{tot} expansion and the linear model give worse predictions than the quadratic model and the NN, which is expected. The linear and quadratic models exhibit better performance in the bias plots. This is most likely due to the fitting procedure and the additional information gained by the other parameters. We see this in Fig. 6.18, where the missing "regularization" makes the prediction of S_{μ} too small This is not true for the NN. The predictions perform not as well because we project to the solution space via an update of its weights. Again all models show similar performance

[9] Indirectly, the distributions of these quantities depend on that of the showers.

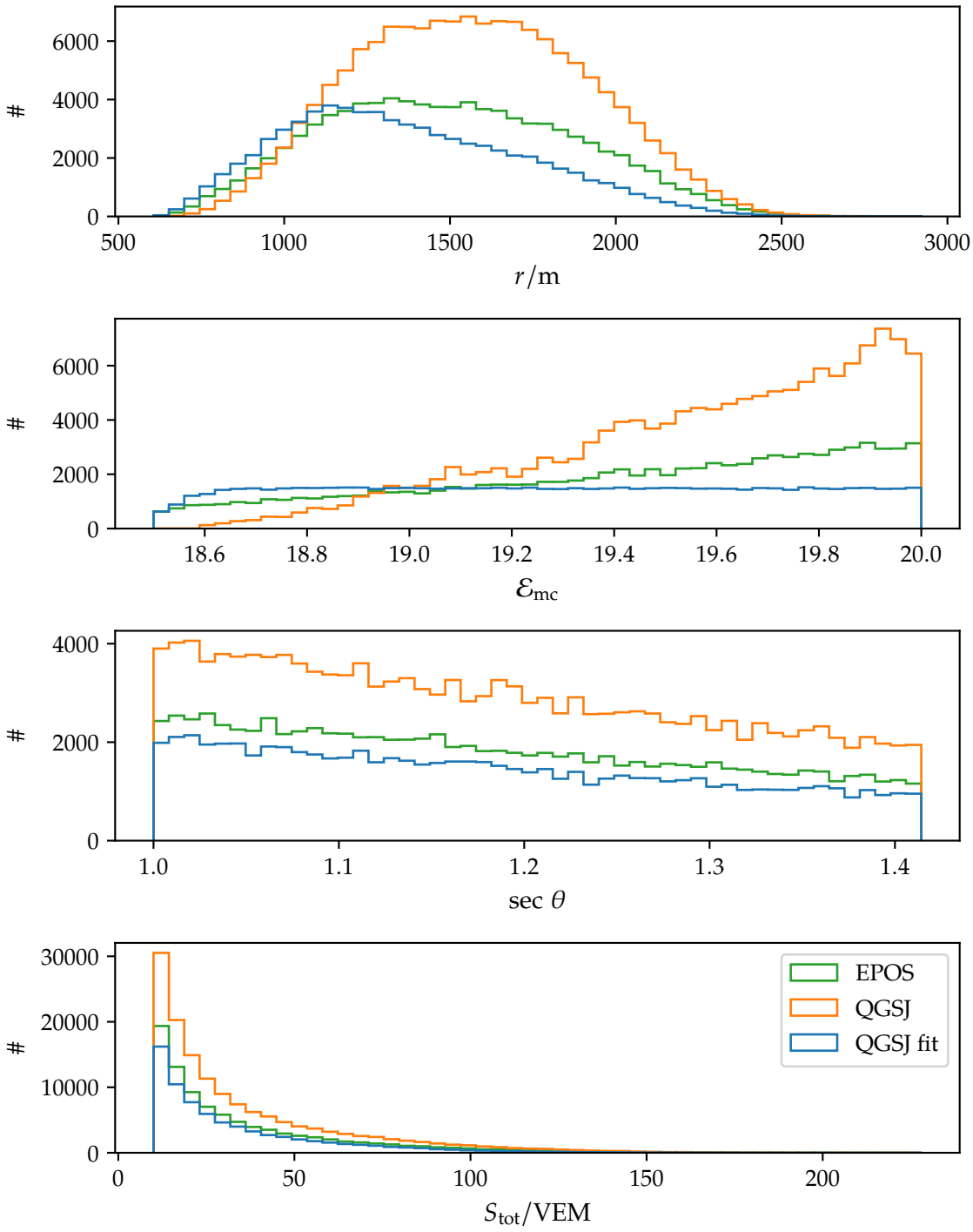


Figure 6.11: Distributions of fit and both test data sets for r, $\mathcal{E}_{mc}$, $\sec\theta$, and S_{tot}. Except in the case of r at least one of the test sets' distributions is similar to that of the fit distribution. The reason for this is visible in the $\mathcal{E}_{mc}$ distribution. We have an excess at higher energies which correlates with more triggered stations farther away from the shower core.

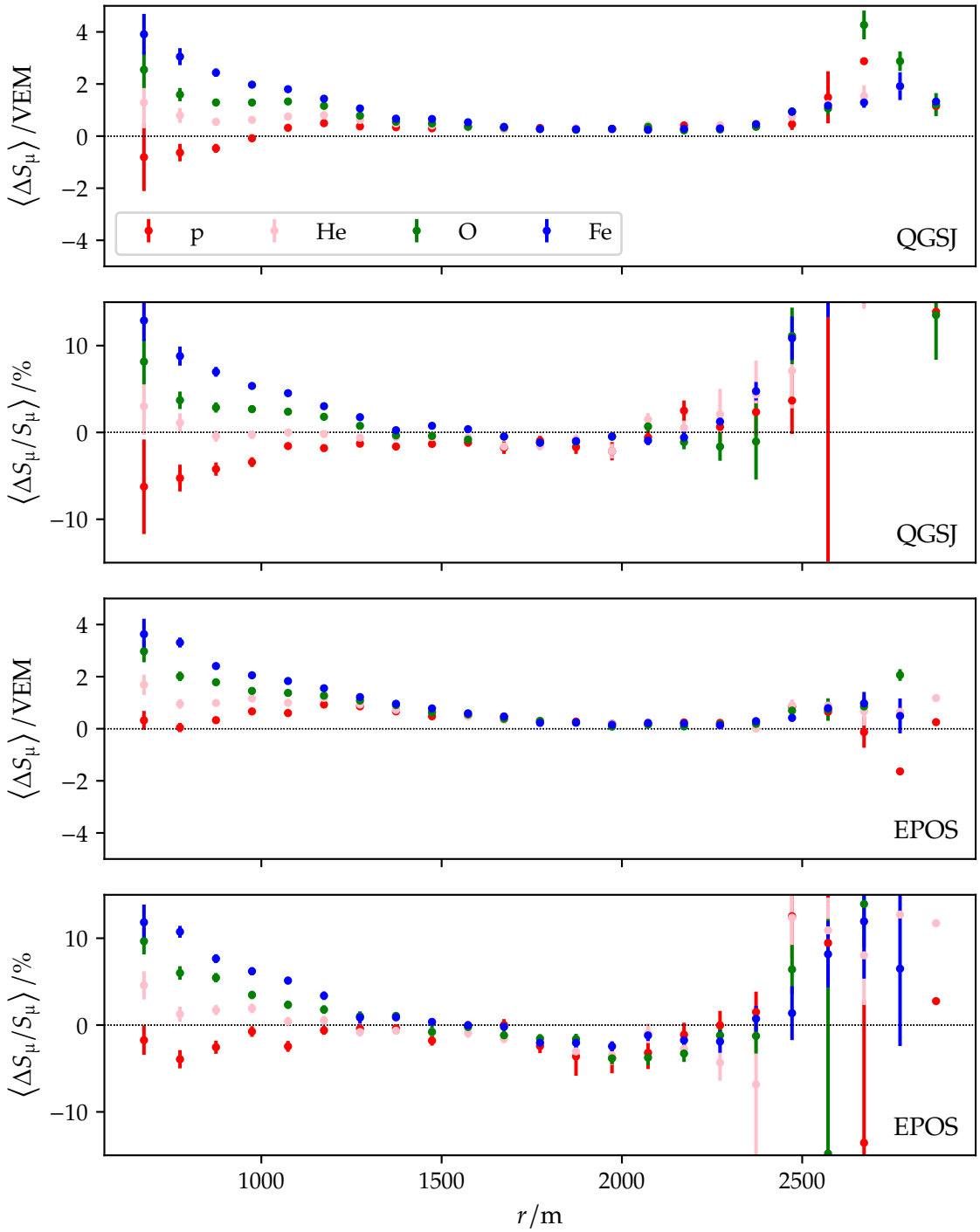


Figure 6.12: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of r. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. With the exception of minor differences, such as the use of Oxygen instead of Nitrogen, this is an attempt to replicate the plots shown in [P:107]. The network performs similar. It shows almost no bias or relative error in the interval [1200 m, 2000 m]. Although, its predictions appear slightly worse on EPOS-LHC, we see no great outliers or deviations. Comparing the results of both hadronic models they even have a similar shape.

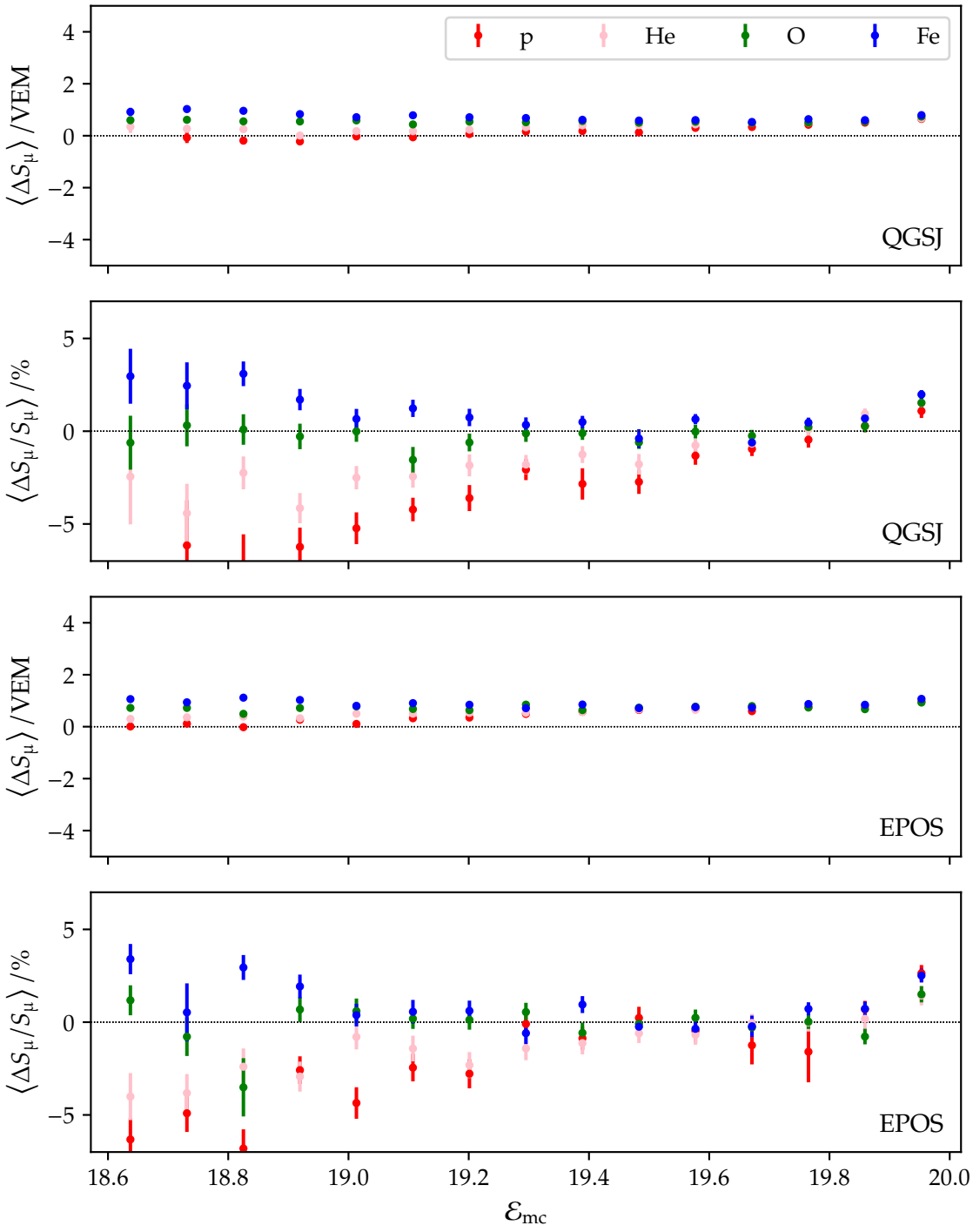


Figure 6.13: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of $\mathcal{E}_{\mathrm{mc}}$. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. Again, the network performs similar to that in [P:107]. For both hadronic models the bias is small over the entire energy range. The bad performance in the low energy part of the relative error is most likely due to the slightly smaller amount of training samples (see Fig. 6.11). Moreover, it could be a consequence of the increase of stations with energy.

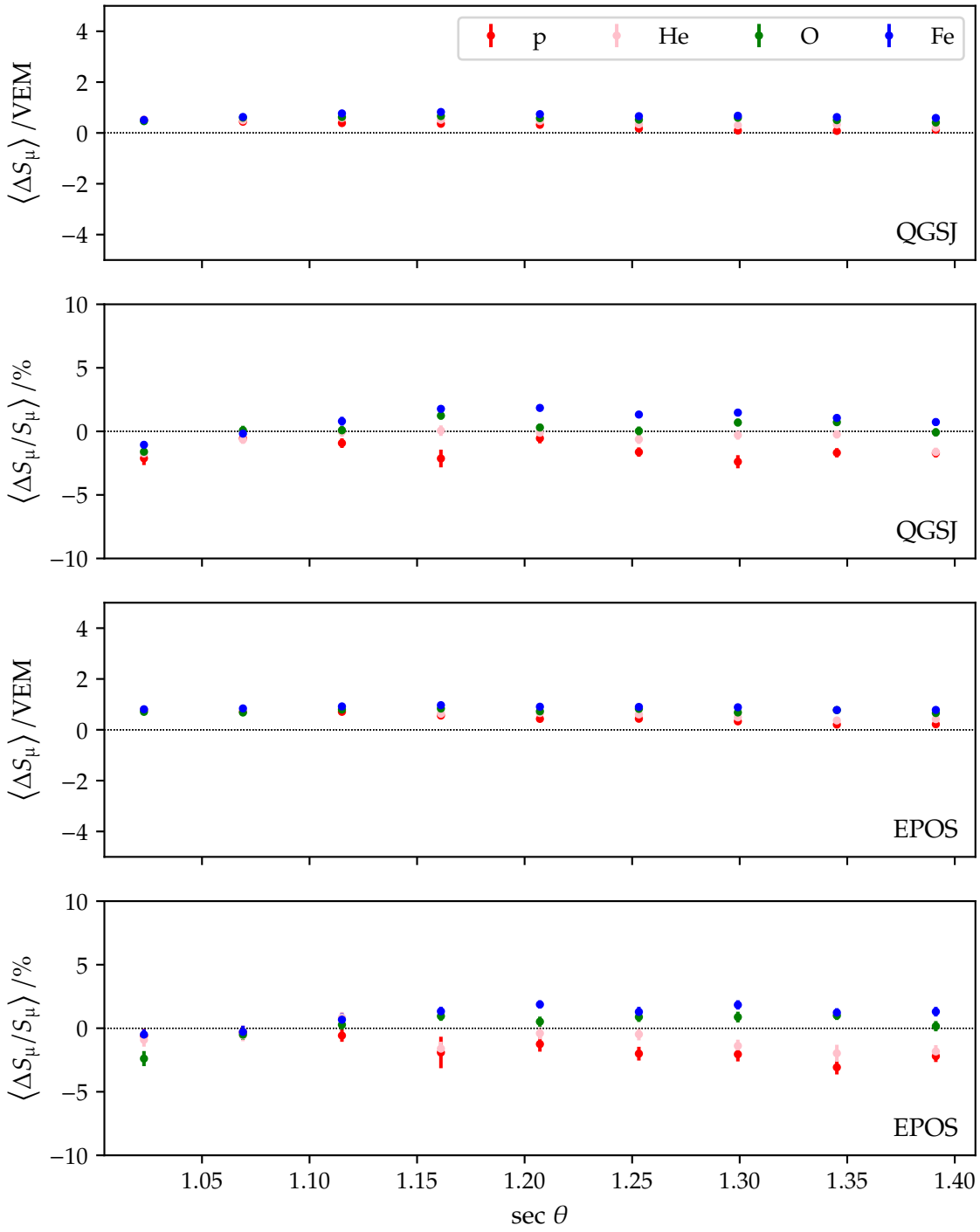


Figure 6.14: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of sec θ. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. Again, the network performs similar to that in [P:107]. Overall there are only small positive biases and centralized relative errors. Therefore, the predictions of the neural network are, on average, almost independent of the chosen zenith angle. Some of the larger deviations of the relative errors are potentially due to the θ sampling (see Fig. 6.11).

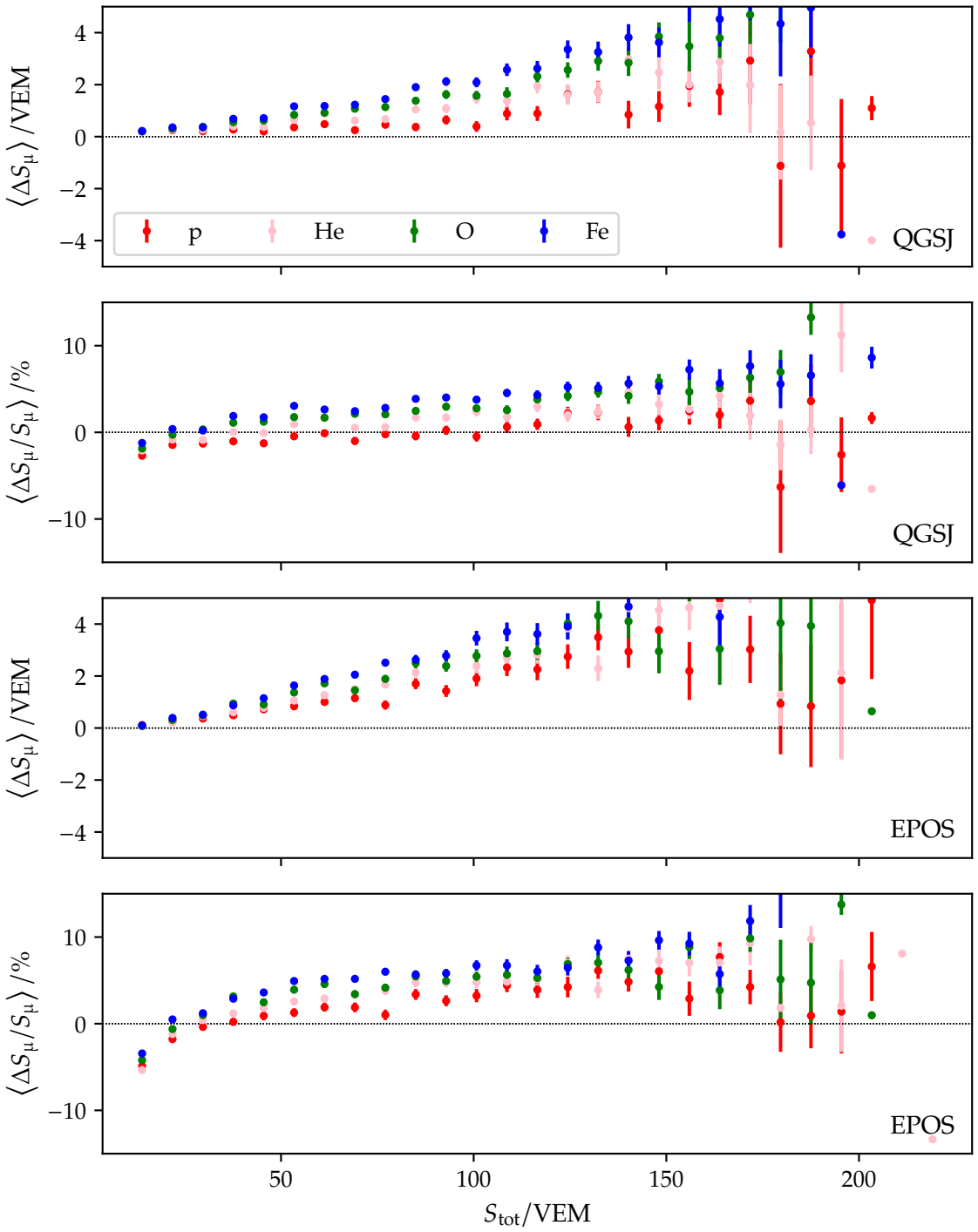


Figure 6.15: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of S_{tot}. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. Again, the network performs similar to that in [P:107]. To get a better picture over the entire range of total signals we have plotted a slightly larger interval. At very high signals ($S_{\mathrm{tot}} > 150$ VEM) the predictions become increasingly bad. This is most likely due to the missing training samples in this region (see Fig. 6.11).

on QGSJetII-04 and EPOS-LHC. We find only in Fig. 6.17, a noticeable difference between the NN approach and the reference models. There the relative error of the NN prediction is much better. This is an indicator that the NN predicts small values of S_μ better. The performance of the NN for small values of S_{tot} (see Fig. 6.18) substantiates this.

In conclusion, we find that there are no huge performance gaps between the models. Nevertheless, it is hard to draw a valid conclusion only on basis of bias and relative error. Even in bad models, both could have consistently small values if the output distribution is chosen correctly.

To get a better feeling of the biases the model exhibit, we compare the average bias for the both best performing models in bins of $\mathcal{E}_{mc}$ and r (see Figs. 6.19 to 6.20). In both cases the fringe regions exhibit almost random biases ΔS_μ which are more prevalent at low distances. Regularly, there the color scale is saturated. Near this region both models tend to underpredict the muon signals. The bias of the NN prediction is overall positive which coincides with Figs. 6.16 to 6.18.

B Variance and resolution

Alongside determining the accuracy of the prediction, it is also crucial to evaluate the precision of the employed models. We analyze the precision in two ways. First, we compare the standard deviation computed from the predictions of the different models in the same manner as in Sec. 6.2.4.A. Even if the S_μ distribution is very un-gaussian this gives us at least a rough idea of the quality of the predictions. Secondly, we use Eq. (4.42). The basic idea is to account for the strong variations in S_μ inside the bins.

Like in Sec. 6.2.4.A, we only check r, $\mathcal{E}_{MC}$, and S_{tot}. In Figs. 6.21 to 6.23, the square root of the variance is shown for the NN and the reference models. This time we are able to sort them in order of their overall performance. As expected, the S_{tot} expansion and the linear model are almost everywhere worse than the quadratic model and NN. It makes sense that the number of model coefficients is somehow correlated to the model variance. Surprisingly, the quadratic model is almost on top of the values given by the NN. The latter is only marginally better overall.

Most of the statements are also applicable to Figs. 6.24 to 6.26. For the quadratic model and the NN, the defined resolution lies roughly between 10% and 20%. It is noticeable that - again - both approaches show a similar performance. As a consequence, we believe that for very small and very high signals, the models have not only bad precision but also bad accuracy. The increase in performance for the quadratic model and NN are mainly due to better predictions in the intermediate range.

C Correlation of true values and prediction

In Secs. 6.2.4.A to 6.2.4.B, we have shown that in terms of bias, relative error, and resolution the quadratic model and NN outperform the other two reference models. Therefore, from this point on, these two models are not discussed anymore.

In Figs. 6.27 to 6.28, we show the correlation of the muon signal vs. its prediction in two different styles (quadratic model, NN)[10]. The left plots are regular, logarithmic histograms showing us the general form of the distribution. In both cases, they follow the identity. Thereby, the prediction of the NN appears coarser around the edges. This is most likely due to the training procedure. On the right plots, we show a row-wise normalized version of the distribution. For this, we divide each row by its maximum value. This shows us bias of the predictions much better. In this way, we are able to substantiate the claims of Sec. 6.2.4.A

[10] For completeness, we show the result of the linear model in Fig. D.7.

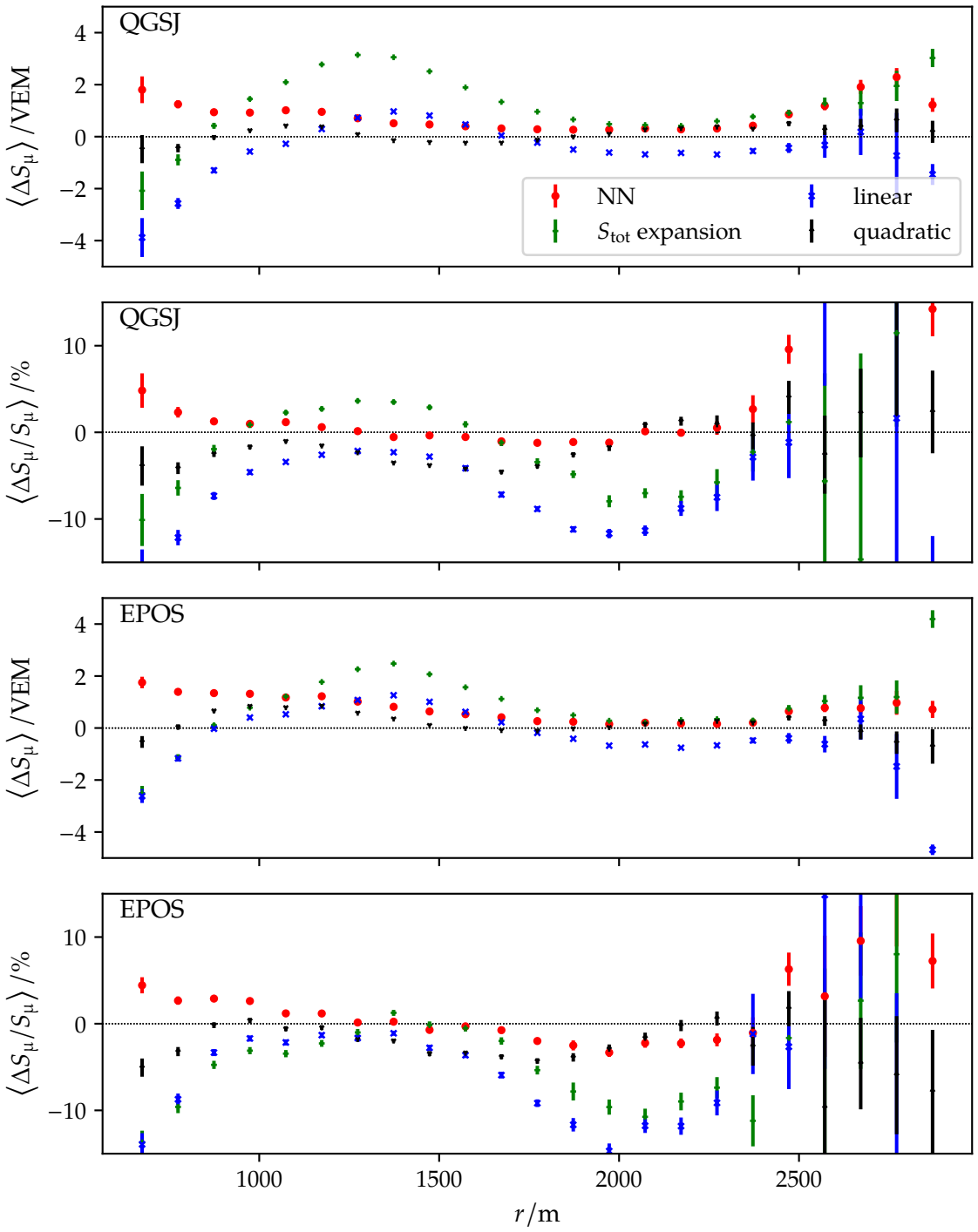


Figure 6.16: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of r. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. The performance of each model depends on the metric applied which is nicely visible in the S_{tot} expansion. Having a large bias compared to the linear model it achieves better values for the relative errors. Both the NN and the quadratic model perform well for both metrics.

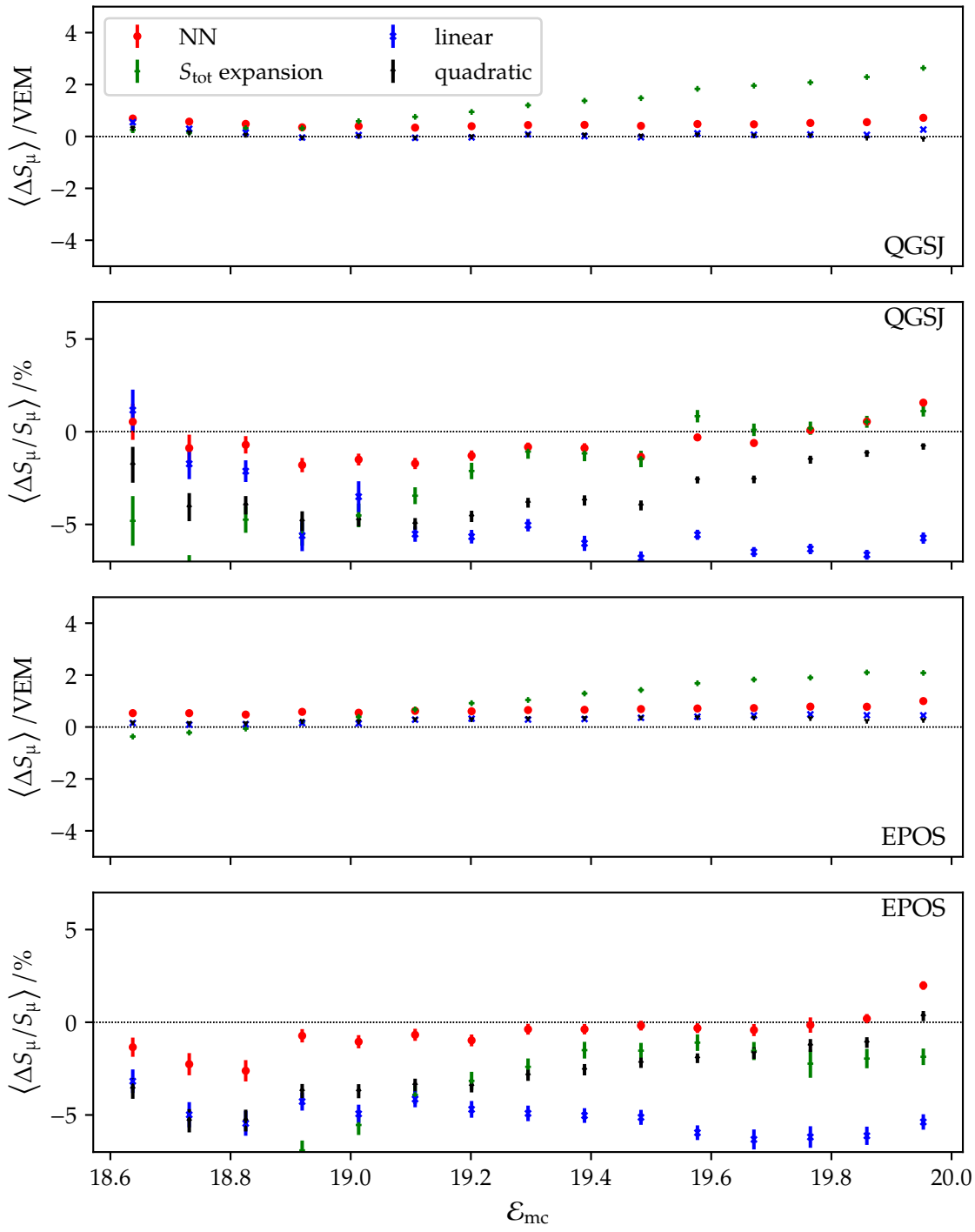


Figure 6.17: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of $\mathcal{E}_{\mathrm{mc}}$. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. In the bias plots only the S_{tot} expansion shows a divergence from zero for higher values of $\mathcal{E}_{\mathrm{mc}}$. Here, the NN achieves a much better result than the linear and quadratic model for the relative error. Again there are almost no differences between QGSJetII-04 and EPOS-LHC.

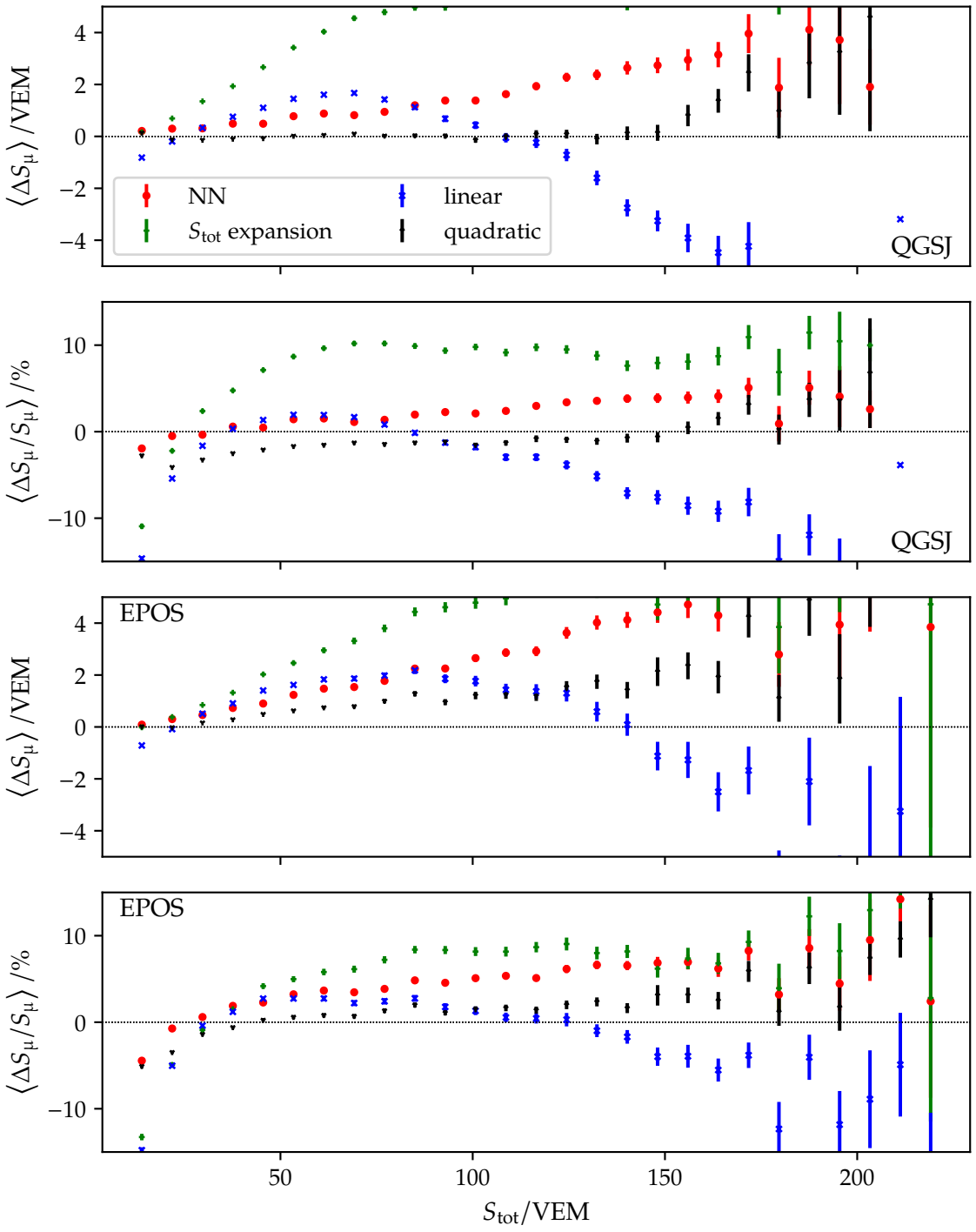


Figure 6.18: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of S_{tot}. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. Here, the S_{tot} expansion fails for higher values of S_{tot}. This is because of the missing regularization. Its coefficients must be relatively small to prevent the predictor from exploding for higher values of S_{tot}.

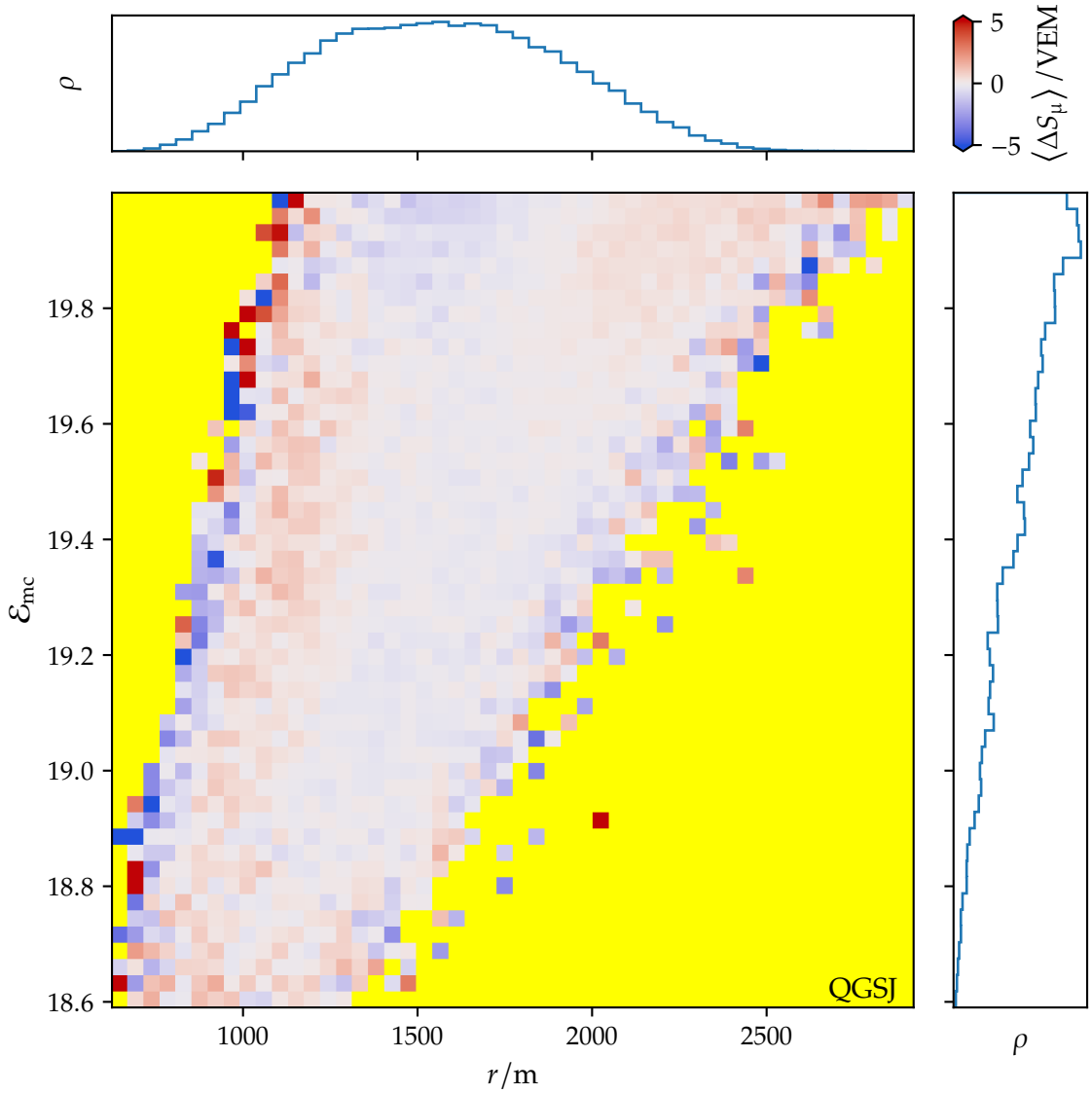


Figure 6.19: Average bias ΔS_μ of quadratic model in bins of $\mathcal{E}_{mc}$ and r. A yellow colored bin indicates the absence of data in that particular bin. The central region of the distribution is - on average - unbiased. Blue and red regions keep approximately in check. Only at the fringes (to the non-data regions) we find distinct outliers, mostly towards smaller distances. The reason for this is most likely that higher signals are expected there.

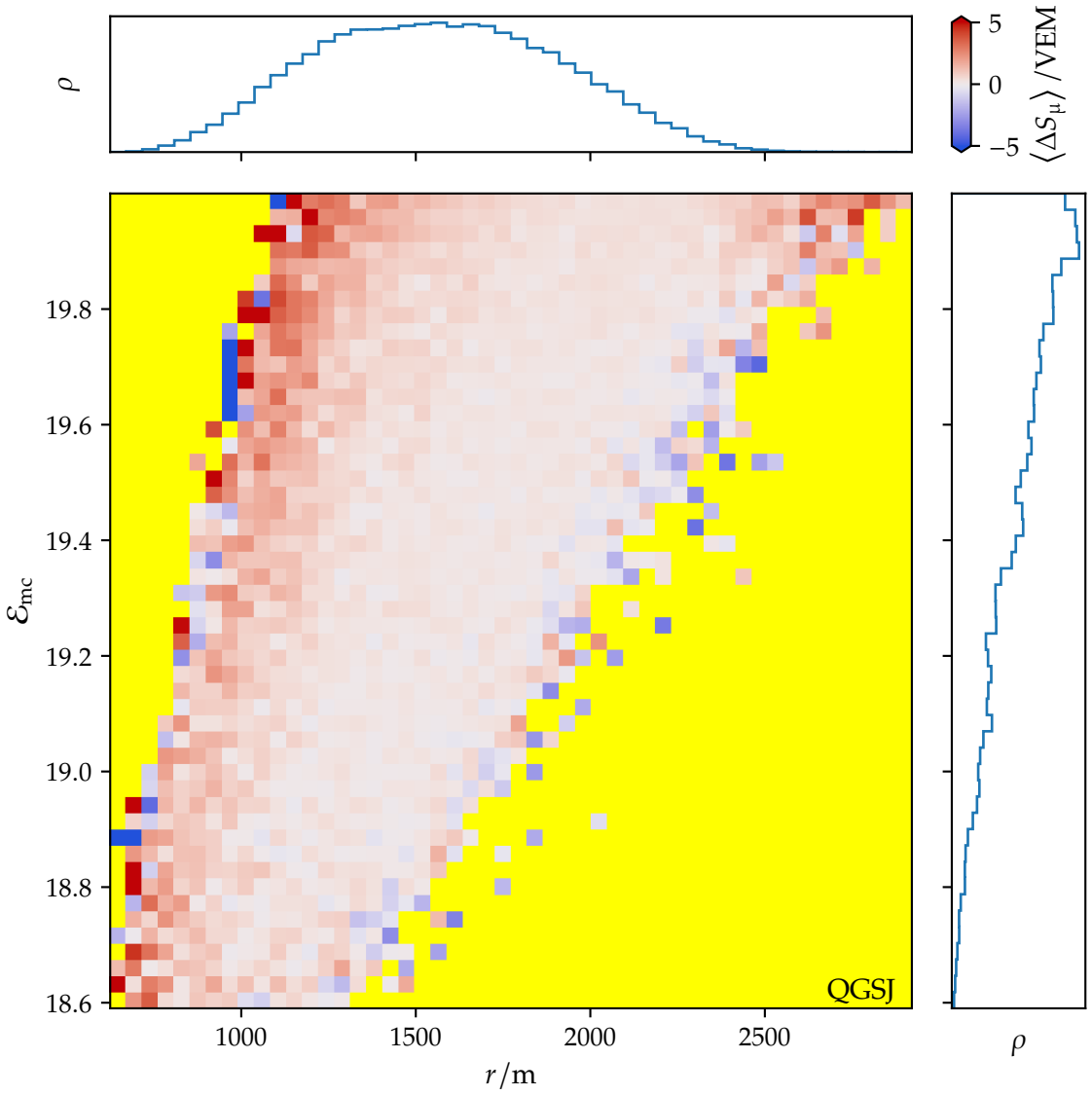


Figure 6.20: Like Fig. 6.19 only for the NN. This time he central region exhibits a slight bias. Therefore, the network tends to predict too small values of S_μ.

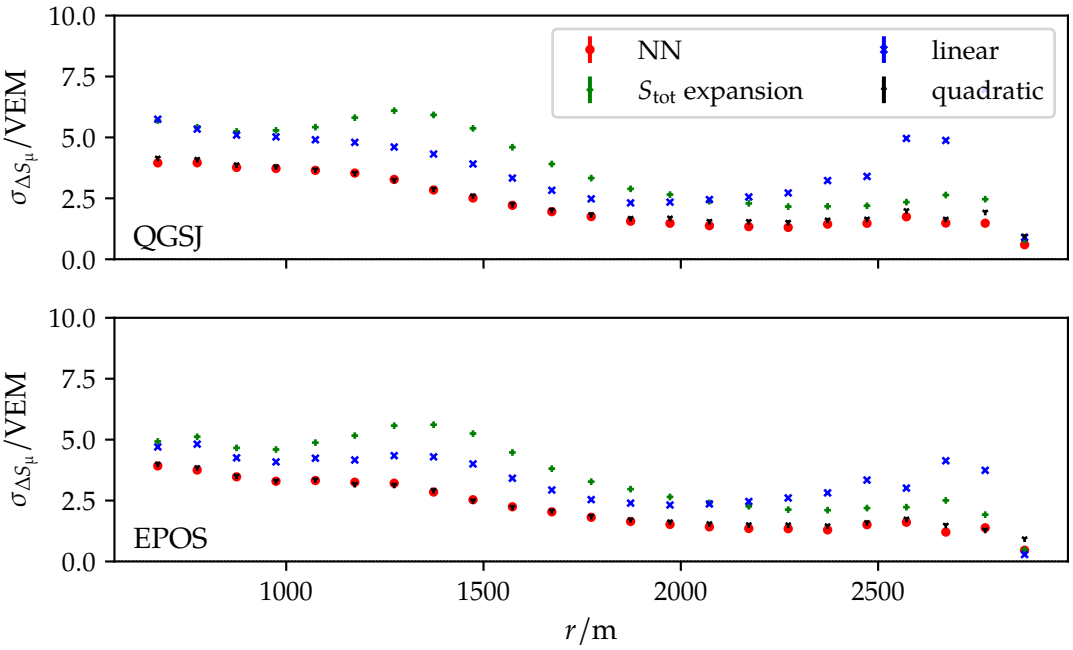


Figure 6.21: Precision of the predictions of reference models and the NN as a function of r for the QGSJetII-04 (*top*) and EPOS-LHC (*bottom*) test sets. There are no distinct outliers except that for higher distances the linear model seems to break down. Again, the quadratic model lies almost on top of the data points of the NN. However, the latter one shows a smaller spread for higher distances.

and Sec. 6.2.4.B. The predictions of the NN show problems in the low and high muon signal region. In the former, it is admittedly better than in the quadratic model. However, for higher muon signals the prediction of NN seems to break down. The network cannot predict too high values.

6.2.5 Estimating the muon fraction

The muon fraction f_μ and the muon signal are closely related quantities. Since f_μ lies on the unit interval, it is reasonable to check how the predictions of S_μ match with the real values of f_μ. In f_μ space, we are able to see deviations from its localized behavior (see Fig. 6.9).

First of all, we compare the distribution of f_μ on the EPOS-LHC based data set if compared to the transformed output of the quadratic model and the NN (see Fig. 6.29). Unfortunately, the distributions do not match. The outputs of the models are more similar to each other than to the underlying data. Their predictions pile up at the region around its peak. Also, the asymmetry between values above 0.6 and below 0.4 increases noticeably. There are almost no predictions below 0.3 anymore. Since the distributions change in favor of a more peaked one, we believe that the models have an 'average-towards-the-mean' problem. They try to compensate by predicting values near the mean value. This is enhanced by the not-matching training and test distributions. To test this, we combine Fig. 6.20 and Fig. 6.29 in Fig. 6.30. For different cuts in f_μ we see that the overall averaged bias gets worse. That is, the higher we cut, the higher the bias becomes. We conclude that this statement is also true for the quadratic model because both behave very similarly through all of the tests.

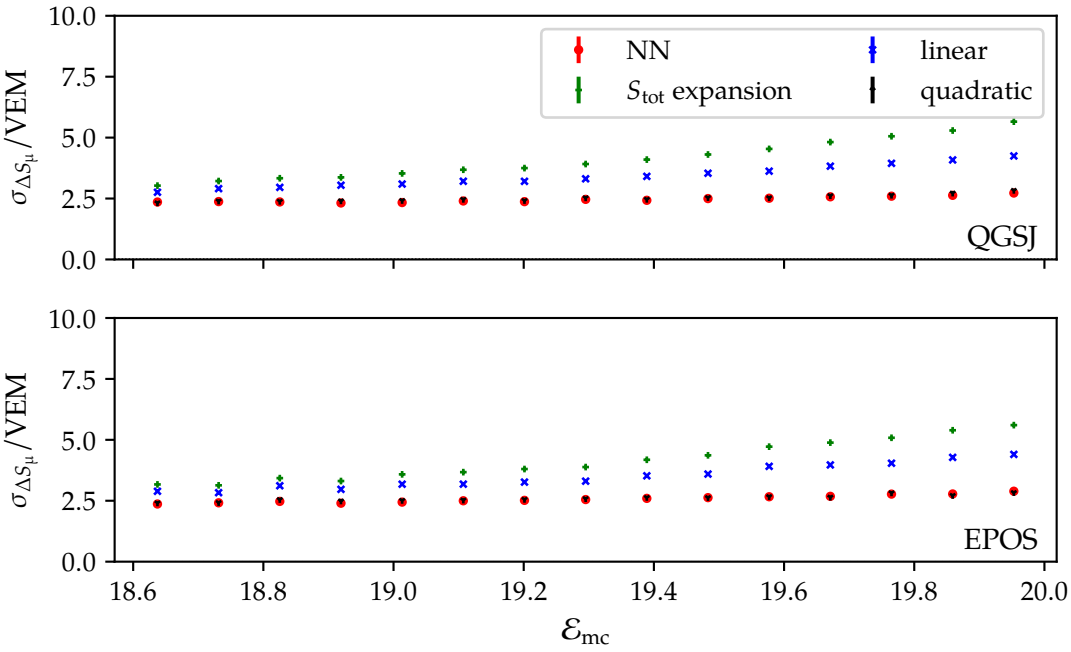


Figure 6.22: Precision of the predictions of reference models and the NN as a function of $\mathcal{E}_{\text{mc}}$ for the QGSJetII-04 (*top*) and EPOS-LHC (*bottom*) test sets. In each energy bin the distribution of S_μ is similar. Raising the energy increases the probability to encounter a larger amount of higher S_μ values. Therefore, we assume that the S_{tot} expansion and the linear model both are troublesome when trying to predict higher S_μ values. In contrast, the quadratic model and NN show an almost flat dependency.

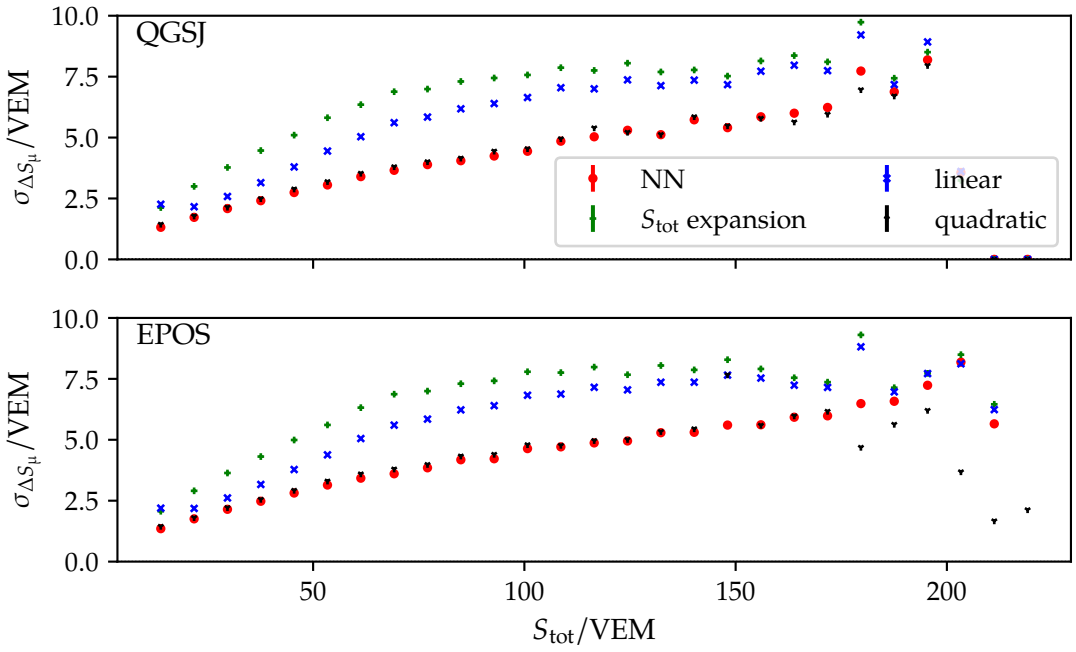


Figure 6.23: Precision of the predictions of reference models and the NN as a function of S_{tot} for the QGSJetII-04 (*top*) and EPOS-LHC (*bottom*) test sets. The instabilities in the high-signal region are most likely due to the scarcity of training samples. This is also the reason why all models behave more or less the same. In general, the standard deviation increases with energy. This behavior is almost linear for the quadratic model and the NN.

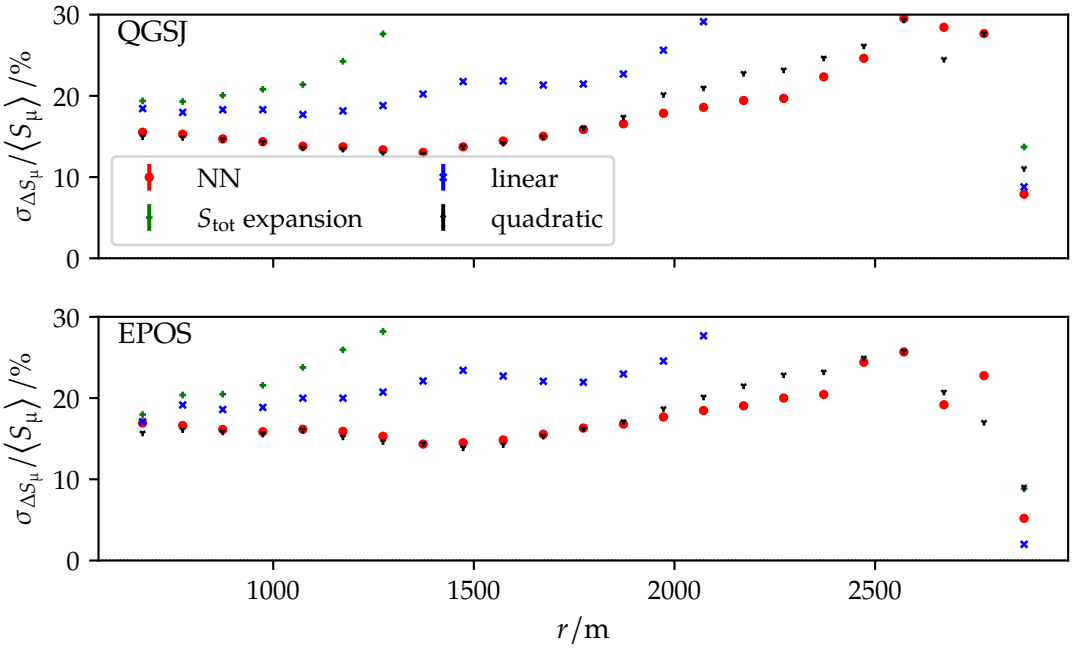


Figure 6.24: Resolution (Eq. (4.42)) of reference models and NN in different bins of r for the QGSJetII-04 (*top*) and EPOS-LHC (*bottom*) test sets. Since we weight each bin with the inverse mean signal we find that both the S_{tot} expansion and linear model both diverge for higher distances. Again, this indicates that for small values these models just predict almost random values. Even in this metric the resolution of the quadratic model almost coincides with that of the NN.

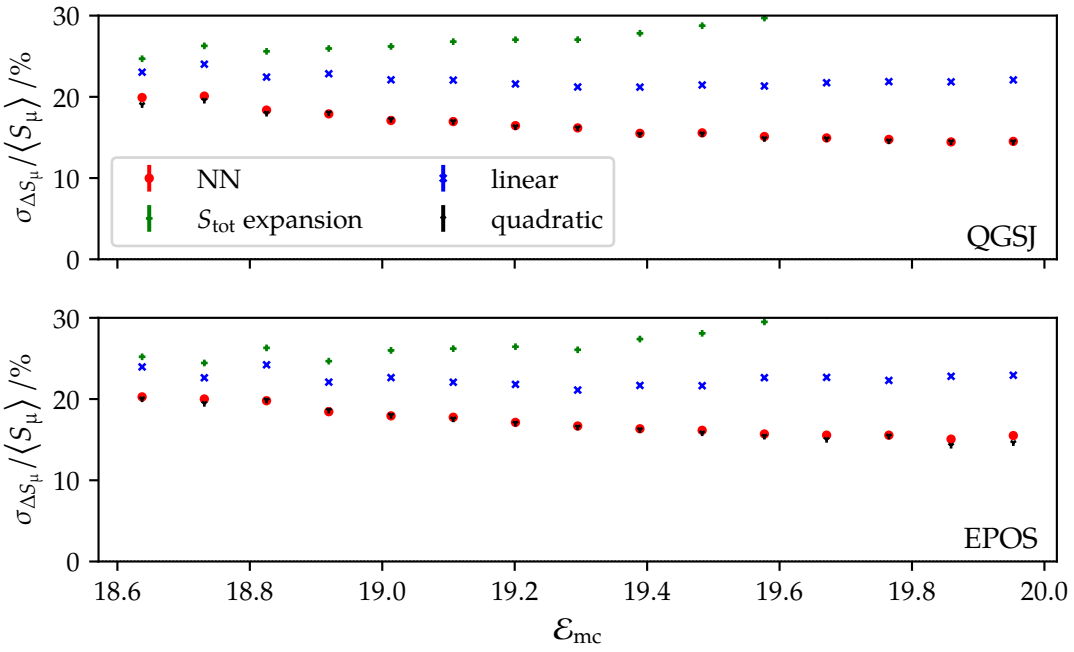


Figure 6.25: Resolution (Eq. (4.42)) of reference models and NN in different bins of $\mathcal{E}_{\text{mc}}$ for the QGSJetII-04 (*top*) and EPOS-LHC (*bottom*) test sets. We believe that same points as written in Fig. 6.22 hold.

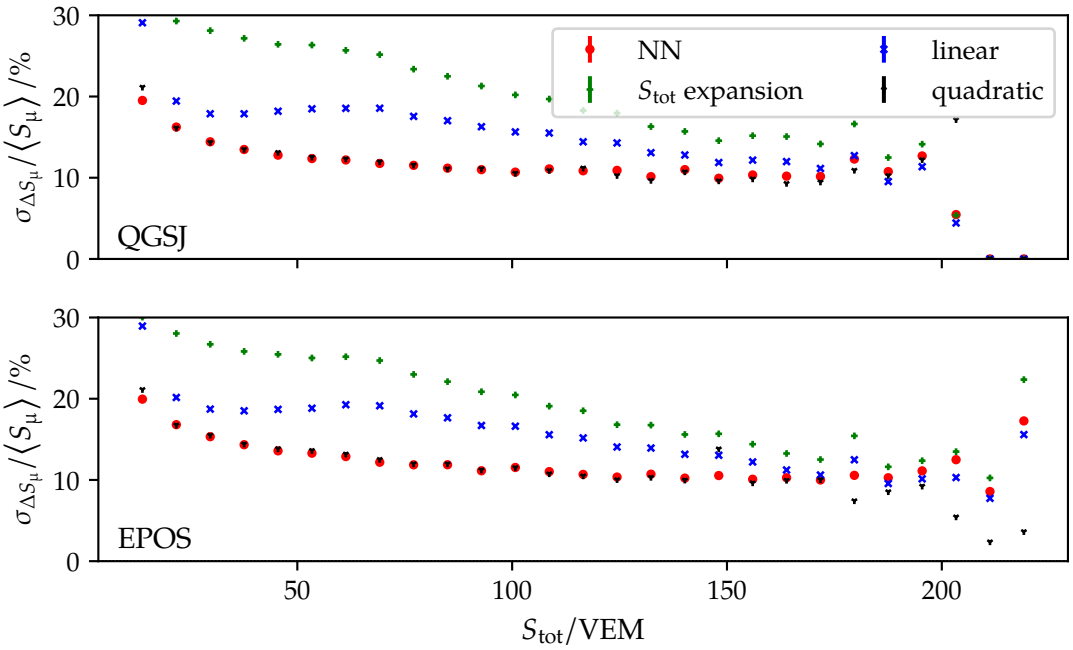


Figure 6.26: Resolution (Eq. (4.42)) of reference models and NN in different bins of S_{tot} for the QGSJetII-04 (*top*) and EPOS-LHC (*bottom*) test sets. All the models seem to converge at this point.

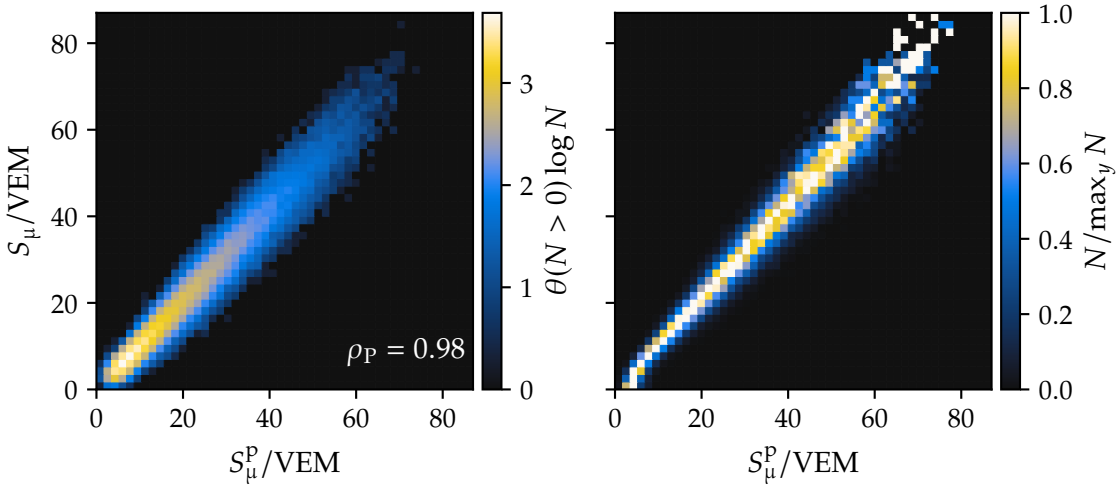


Figure 6.27: Histograms of S_μ value vs. the prediction of the quadratic model in logarithmic representation (*left*) and normalized to the maximum value of the row (*right*). In the density plot it is visible that the predictions follow the identity axis except for very small values of S_μ. There the model breaks down. This is substantiated by the normalized plot. We see that most low value predictions are wrong.

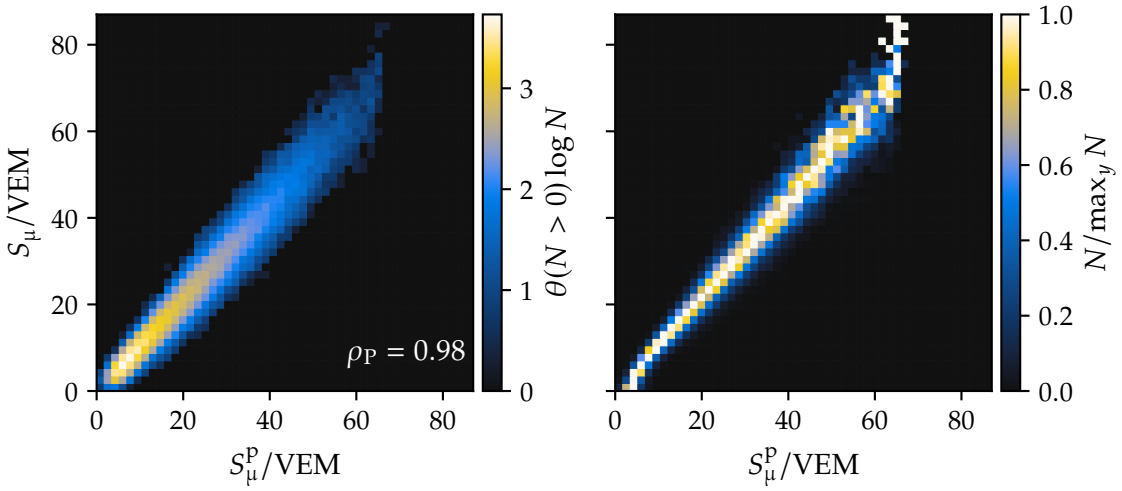


Figure 6.28: Histograms of S_μ value vs. the prediction of the NN in logarithmic representation (*left*) and normalized to the maximum value of the row (*right*). In contrast to Fig. 6.27, we find artifacts around the border region of the region around the identity axis. Comparing the right plot to the right plot of Fig. 6.27 we find that the NN has slightly better prediction for very low muonic signals. However, the NN seems to have a soft maximum for the muon signal prediction. and appears to be unstable in the central region.

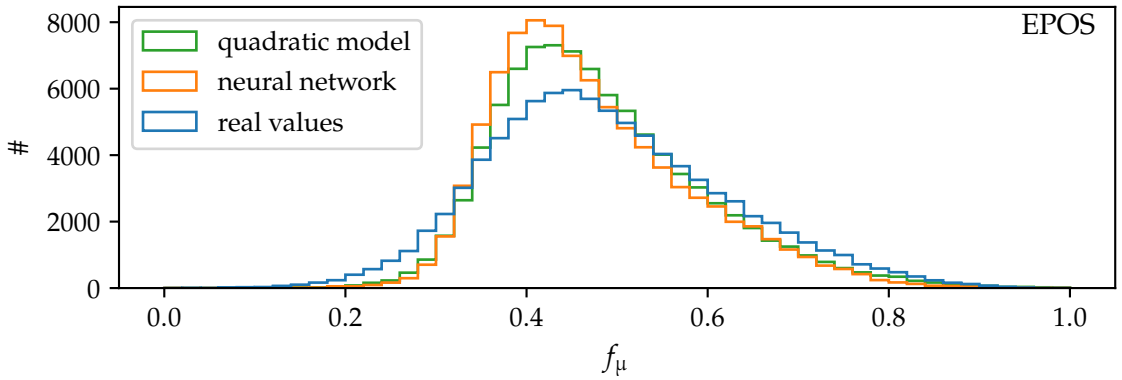


Figure 6.29: Distribution of muon signal fractions in the test data set based simulated from the hadronic interaction model EPOS-LHC (see Eq. (6.1)), the predictions of the quadratic model, and the NN predictions. The distributions do not match. For both models the predictions accumulate around the peak of the test data set. Despite much better than a mean prediction (see Table 6.1) this indicates that the models at least orientate themselves towards this mean prediction. Values at the edges of the distribution are not likely to be predicted. In this picture, they are in a state of averaging towards the mean.

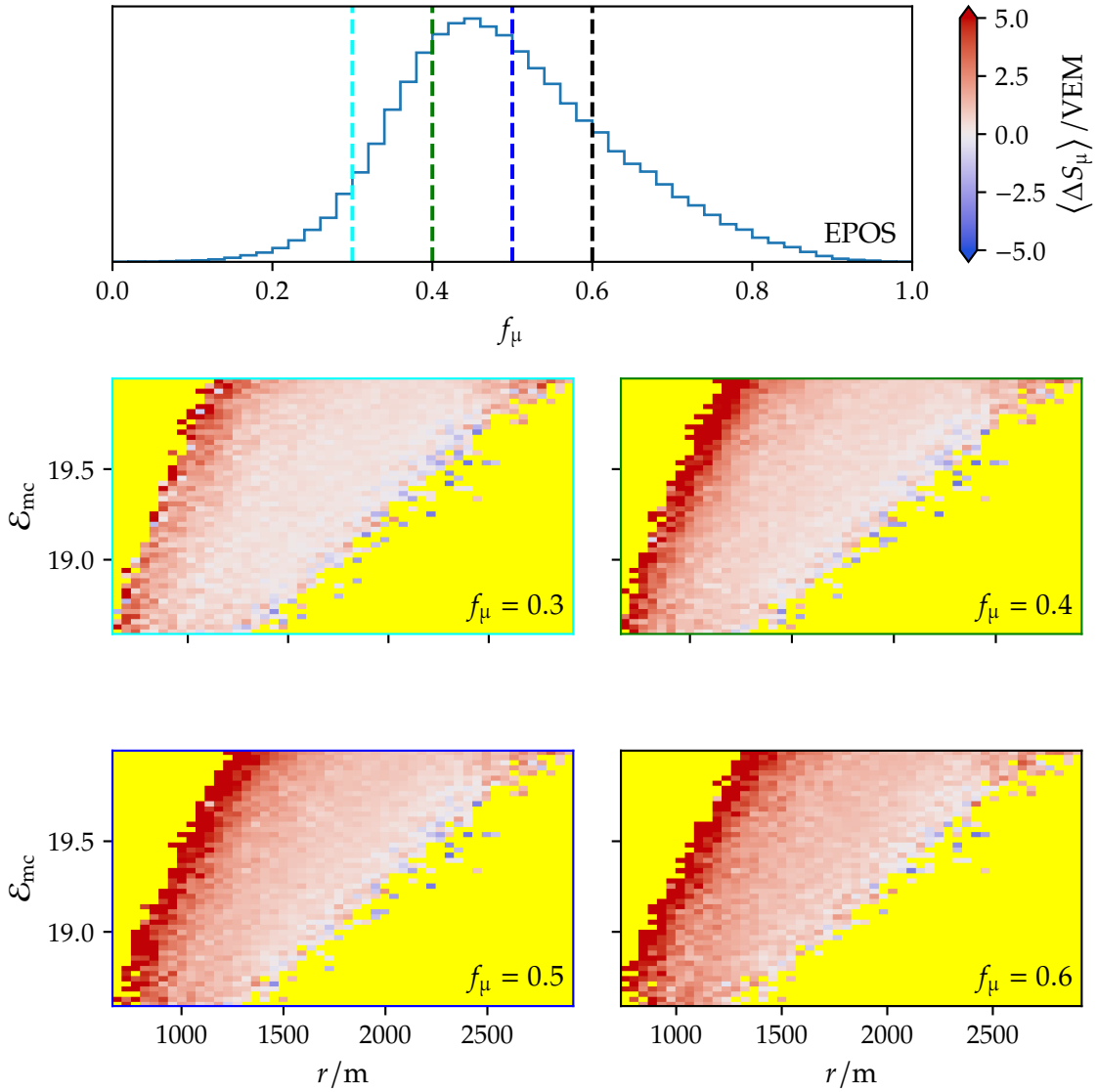


Figure 6.30: Effect of cuts in f_μ distribution on the average bias ΔS_μ of the NN in $\mathcal{E}_{mc} - r$ bins. The higher the f_μ cut is the more reddish the plots become overall. Moreover, the left border region starts noticeably to saturate. This indicates that the NN predicts too low values for higher muon contents supporting the claim made in Fig. 6.29. Seemingly, the NN is biased towards the mean value. We believe that this is also true for the quadratic model.

6.3 Extraction of muon time signal

In the previous section, we used only scalar inputs of the WCDs. It remains a valid question whether we can improve the prediction of the muonic signal S_μ by introducing trimmed traces to our problem or additional parameters derived from them. The time structure of the traces contains information that has not been captured by our choice of parameters. Some of this information could be highly correlated to S_μ. We use NNs to check this hypothesis.

Moreover, we also check the effect of adding SSD trace information to the mix of input parameters. Therefore, we switch from the old NapLib data set to the new UUB one defined in Sec. 5.1.1. We also use new cuts that are in line with that in [A:23] and in [P:101]. To simplify the analysis and the comparison to additional SSD signals, we only take a subset of the whole data set into account. We restrict ourselves to proton primaries in the logarithmic energy range of $[19.0, 19.5]$ and take only stations into account for which both[11] the WCD and SSD have triggered. In addition, we do not do any resampling of our distributions. We have summarized this data set in Row 5.4.c.

For trace-based predictions, the target is slightly different. Since we use trimmed traces, a part of the time signal might not be inside our region of interest (ROI). Hence, we define

$$S_\mu^{\mathrm{tr}} = c \sum_{b=b_\mathrm{s}}^{b_\mathrm{s}+L_\mathrm{t}} S_\mu(b)/a_\mathrm{p}, \tag{6.4}$$

where L_t is the size of the trimmed trace, a_p is the area over peak, and c is a conversion factor. We set the L_t for this analysis to 200 UB bins and use the a_p value directly from $\overline{\mathrm{Off}}\underline{\mathrm{line}}$. The conversion factor accounts for the difference in baseline fluctuations and the reduced trace length. We fix it by demanding

$$\langle S_\mu^{\mathrm{tr}} \rangle \stackrel{!}{=} \langle f_\mu S_{\mathrm{tot}} \rangle, \tag{6.5}$$

where S_{tot} is the total signal taken from $\overline{\mathrm{Off}}\underline{\mathrm{line}}$ and f_μ is the muon fraction computed from the photo-electrons (see Eq. (5.18)).

We further simplify this analysis by converting our UUB traces to UB equivalent ones by averaging over three consecutive bins. This procedure is not entirely correct due to a timing miss-match (25 ns to 8.3 ns) and the negligence of the new response of the electronics. However, we show in Sec. 7.4 that the UB-equivalent traces behave in an event-based analysis very similar to regular traces taken from UB simulations. By downsampling the traces, we ensure that any effect or change is due to the addition of the SSD information and, at the same time, that the method is comparable to the preceding analysis.

6.3.1 Using only WCD signals

This part is mainly a crosscheck of the results in [A:12], [A:23], and (by extension) in [P:101]. In [P:101], the test and validation sets have been drawn randomly from the ground data set in such a way that the events follow a uniform distribution in $\lg(E/\mathrm{eV})$ and $\sec\theta$. This yielded slightly different results due to the difference between the underlying data sets if this is not done (see Fig. 6.31). Nevertheless, the training data set has approximately the same form as the raw distributions. Therefore, it is sufficient to test our results on a non-sampled data set since the training process is untouched. We expect similar results to [A:12, A:23] and show that re-weighting in $\sec\theta$ yields similar results to [P:101]. As base architecture, we use the one found in [A:12] and in [P:101]. We believe that the more complicated architecture

[11] Since the SSD triggers only if the WCD has triggered, it is sufficient to check only the SSD signal.

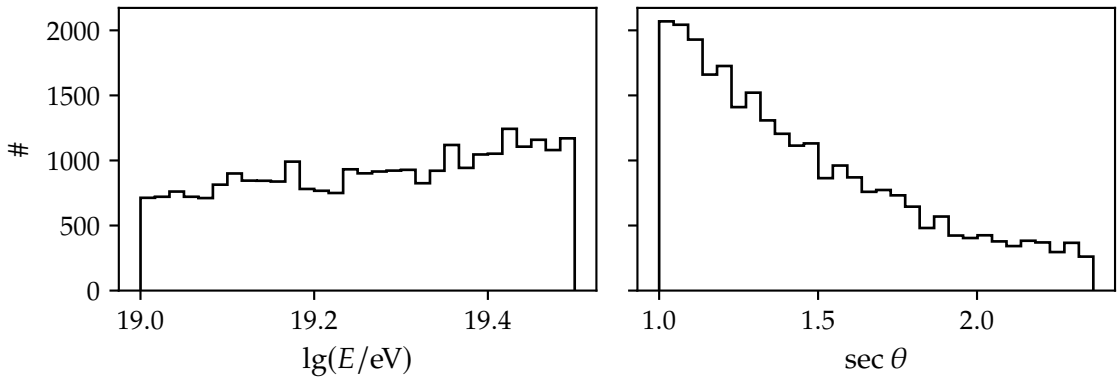


Figure 6.31: Distribution of logarithmic energy $\lg(E/\mathrm{eV})$ and the zenith angle θ in terms of sec for the data set defined in Row 5.4.c. Since the shower events in the underlying data set are distributed uniformly in $\lg(E/\mathrm{eV})$ and $\sin^2\theta$, we obtain non-uniform distributions for the station-level events. In contrast to [P:101], our data set is slightly leaning towards higher logarithmic energies and lower zenith values.

of [A:23] did not show better results since it has been reverted to the previous[12] architecture in [P:101].

Comparing the results to those presented in [P:101, A:12, A:23], we find reasonable agreement (see Fig. 6.32), taking $\sigma_{\Delta S_\mu}$ as a comparison metric. The deviation in the energy range is equivalent to that shown in [A:23] for the chosen bin in shower plane distance r and sec θ. Moreover, the performance of the predictions of the NN is very similar to that in [P:101] (see Fig. 6.32). Comparing both plots, the predictions of the NNs show better results for lower energies and higher values of sec θ. This explains the difference of the here-presented results to the results in [P:101]. It arises from differently weighted parts in the phase space (see Fig. 6.31).

In Eq. (6.4), we see that S_μ^t does not necessarily correspond to the actual muonic signal S_μ^r. Since we sum over 200 UB-equivalent bins (see Sec. 6.3) of the ideal muon trace $S_\mu(t)$ provided by $\overline{\mathrm{Off}}\underline{\mathrm{line}}$, we do not account for differences in baseline fluctuations or larger effective trace lengths. There is no easy way to fix the simulation result of the muon traces for our data set. Therefore, we recheck the performance of the NN on the muon signal obtained via the muon fraction f_μ (see Eq. (5.18)). The f_μ from the PMT photoelectrons is, in this case, a more robust estimate for f_μ^r.

Again, we compare the results of the NN to that of a simple alternative model. Due to the restrictive nature of our data set, we use only the best performing inputs identified in Sec. 6.1.4. Therefore, we end up with the total signal S_tot, the zenith θ, the shower plane distance r, the risetime $t_{1/2}$ and the falltime t_f. This time the only special input preparation lies in using the sec θ. Due to the increase in complexity of the NN compared to that in Sec. 6.2, we use a slightly more complex reference model. Instead of going to the second degree, we use the third when generating the polynomial features. We end up with 35 features[13] and call this the third-polynomnial upscaled reference model (P3). The model is fitted using S_μ as a target. Due to its simplicity, there can be negative predictions of S_μ. We zero these negative muon signals before further comparisons.

In Fig. 6.33, we show the direct comparison between both models and the effect of using S_μ^t and S_μ as targets. As expected, using S_μ^t yields a superior result for the NN. However,

[12] It has been first shown in [A:12].

[13] If we have a degree of n and m inputs, the number of unique features is: $\binom{n+m-1}{n}$.

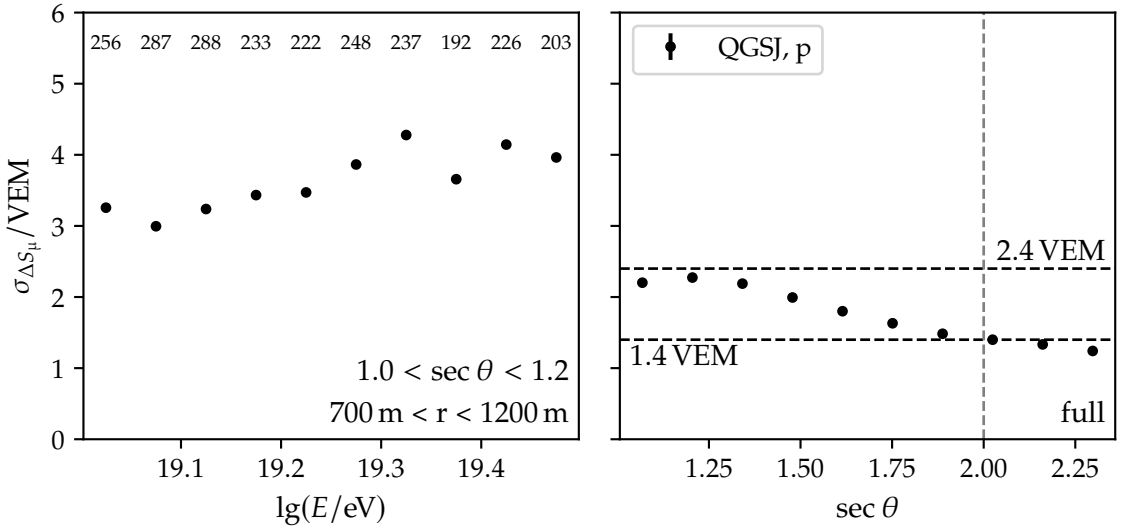


Figure 6.32: Precision $\sigma_{\Delta S_\mu}$ of the predictions of the network based on the architecture in [P:101] binned in logarithmic energy lg(E/eV) (*left*) and the zenith angle θ in terms of sec (*right*). For the *left* panel, we have used the same bins as in [A:23] to allow for a better comparison. We added small numbers to the plot to show the number of station-level events in each energy bin. The vertical dashed line in the *right* panel marks the zenith angle below the SD is fully efficient. The horizontal dashed lines mark the precisions of 1.4 VEM and 2.4 VEM. The network predictions become slightly worse for higher energies and better for lower values of sec θ.

this advantage vanishes when we go over to the better estimator S_μ. As expected, the network does not lose much of its predictive power if S_μ is used as a comparison value (see Sec. 5.4.3.A). This is most likely because the network performs better for low values of S_μ and S_μ^t is slightly larger than S_μ (see Fig. 6.33). We find in both cases comparable results. On S_μ, the advantage of the network is only barely visible. Due to its – most likely – more precise estimation of the real muonic signal S_μ^r, we use S_μ in the following as the target.

Still, using different metrics (see Fig. 6.34), we find that there are definitive differences between both models. As expected, the NN focuses on regions of the phase space that exhibit smaller signals. Below 18 VEM and at higher distances, the network performs much better than P3. The distributions at the bottom of the graphs are also the regions in which most of our events lie. We believe that by choosing a more sophisticated training procedure and more data at higher muon signals S_μ, we could surpass P3 in all regions. Still, it is not a huge leap from P3 to the results in NN if we take only larger muon signals into account.

6.3.2 Adding SSD information

The easiest way to extend the NN model is to provide the network with SSD information by adding it as a second channel to the RNN input of the base network (see Fig. 4.5). We use the same extraction and preparation procedure for the SSD traces as for the WCD traces: we trim to 600 bins (from the start bin) and take the mean of three consecutive bins.

In Fig. 6.35, the comparison of the WCD-based NN and the NN where we added SSD information is depicted. To ensure that we work in a suitable phase space for the SSD, we use only events that come from events below a zenith angle of 45°. Unfortunately, this straightforward way of introducing the SSD into the setup does not achieve any meaningful benefits. This result could have many reasons: First and foremost, the simple setup lets the network itself deal with the difference between VEM and MIP. This could yield a scaling problem.

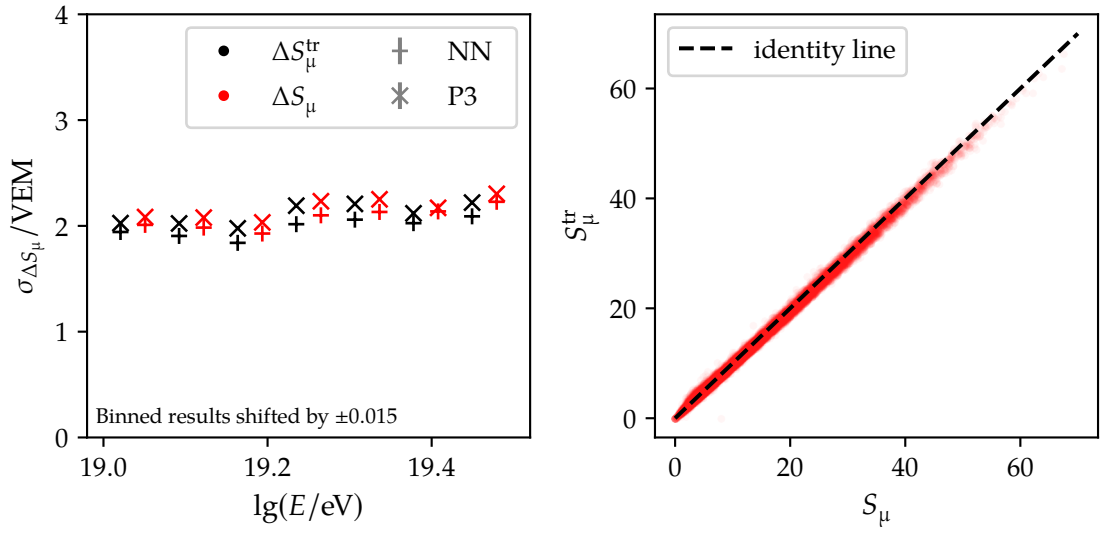


Figure 6.33: *Left:* Precision $\sigma_{\Delta S_\mu}$ of the predictions of the NN (pluses) and P3 (crosses) using the muonic signal computed from the trimmed traces S_μ^t (black, see Eq. (6.4)) and muon signal computed from photo electrons S_μ (red, see Sec. 5.4.3.A) as targets. For the sake of visibility, we shifted the binned values by 0.015 and −0.015 from the bin center for the different targets. Since we fit P3 on S_μ its predictions are less precise on S_μ^{tr}. For S_μ, which we believe to be the superior measure for the real muon signal, both models show very similar results. *Right:* Comparison between S_μ^{tr} and S_μ. The dashed line is the identity line. The result is comparable to Fig. 5.14.

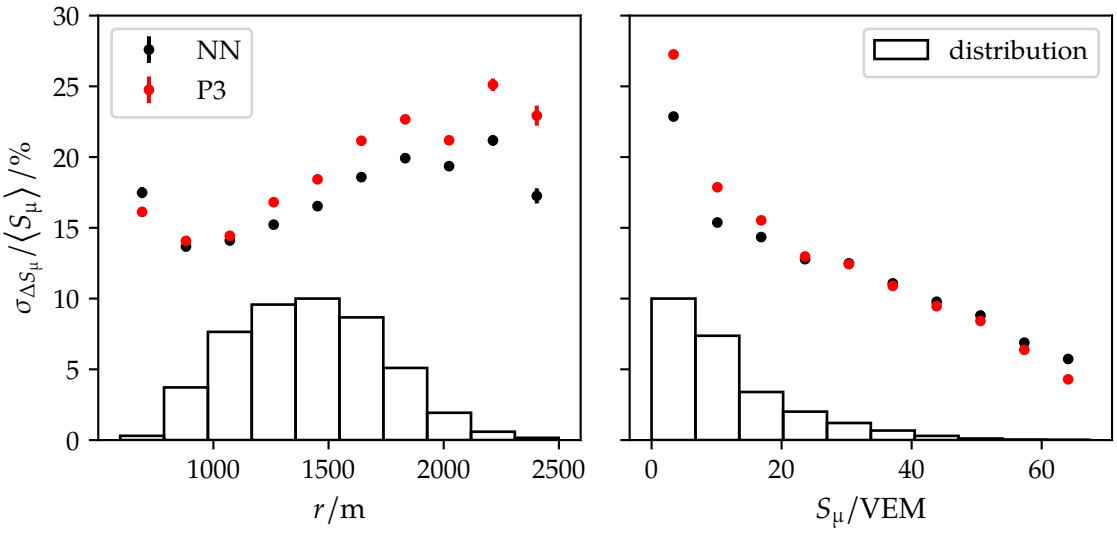


Figure 6.34: Resolution of the predictions of NN (black) and P3 (red) evaluated on $\sigma_{\Delta S_\mu}/\langle S_\mu \rangle$ as a function of the shower plane distance r (*left*) and the muonic signal S_μ (*right*). The bars at the bottom of both plots indicate the distribution of data. Their height is normalized to the highest bin and scaled to the 10% mark. The NN clearly outperforms P3 for high shower plane distances and low muon signals. This is also the region in which most of our data resides.

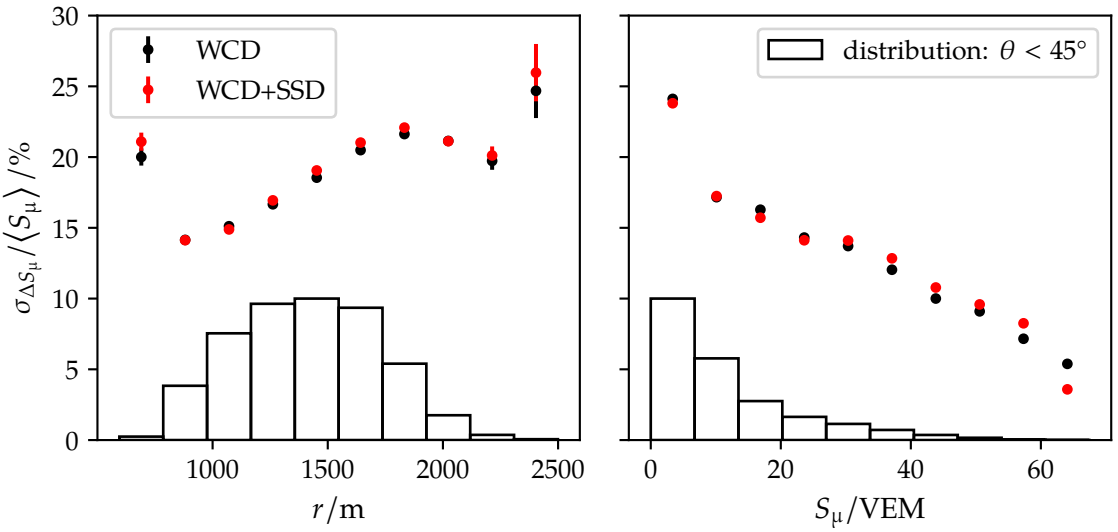


Figure 6.35: Resolution of the predictions of WCD- and WCD+SSD-based NN as a function of the shower plane distance r (*left*) and the muonic signal S_μ (*right*). We compute the bars like in Fig. 6.34. We use only events below 45° to test in the phase space in which the SSD works better. The performance of both networks is essentially the same. There is virtually no improvement of the NN using additionally the SSD over the NN from Sec. 6.3.1.

Furthermore, the architecture might be unsuitable for such kind of analysis. In the worst-case scenario the SSD trace carries no essential information in most of our chosen phase space and acts as additional noise in the training process.

To exclude the last hypothesis, we retrain a network and add an SSD parameter to the additional parameters. If the flux that goes through both detectors is homogeneous enough, we expect that their signals are (in some way) connected (see Sec. 3.2.4). We also expect that the response of both detector systems to the flux should be different. Consequentially, it makes sense to construct a feature that relates the total signal in both detectors. Since the SSD triggers only if the WCD triggers, it seems to be a good idea to use the signal fraction

$$f_S = \frac{S_{\text{tot}}^{\text{SSD}}}{S_{\text{tot}}^{\text{WCD}}} \tag{6.6}$$

as an additional feature [A:24]. In Universality, this already yielded a more unbiased result in the event-level analysis [A:25]. Indeed, adding this new feature to the mix improves the NN prediction considerably (see Fig. 6.36). Nearly all bins show a superior result. This indicates that SSD information potentially improves the predictions of such networks.

To conclude this section, we want to compare this new network with another simple model. As a baseline model, we use that of the last section and add the integrated signal of the SSD to the inputs giving us 56 unique features. In Fig. 6.37, we find a similar result to that in Fig. 6.34. The network clearly outperforms the alternative model for low S_μ. However, for higher values of S_μ, the network falls off.

6.4 Extracting number of muons from UMD traces

One of the only direct ways[14] to measure the local muon content of a hadronic shower lies in evaluating the UMD detector responses. Since they are positioned underground, only

[14] Highly inclined shower contain due to the attenuation mainly muons.

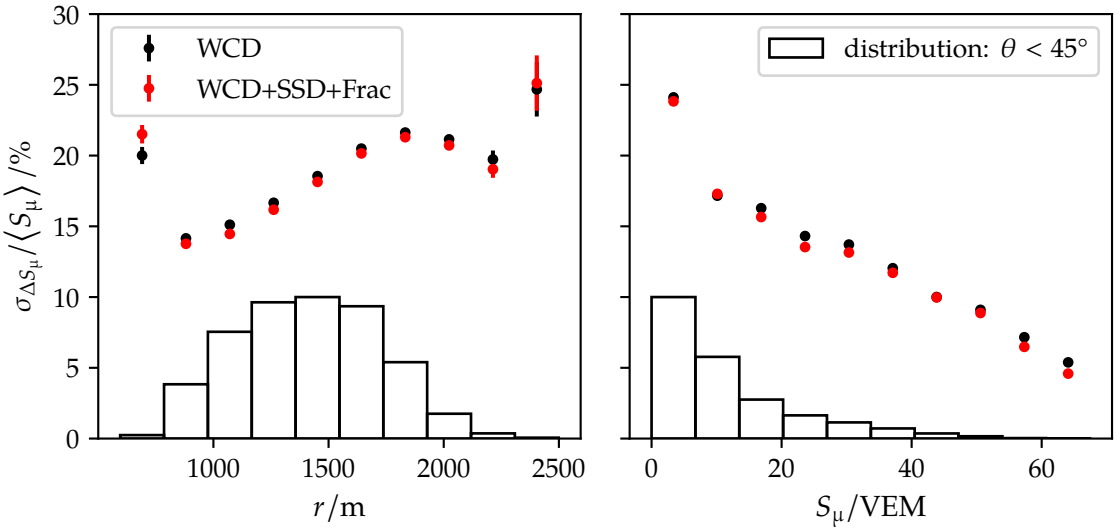


Figure 6.36: Resolution of the predictions of WCD- and WCD+SSD- (together with the f_S from Eq. (6.6)) based NN as a function of the shower plane distance r (*left*) and the muonic signal S_μ (*right*). We compute the bars like in Fig. 6.34. We use only events below 45° to test in the phase space in which the SSD works better. This time the performance of both network using the additional SSD information is slightly better.

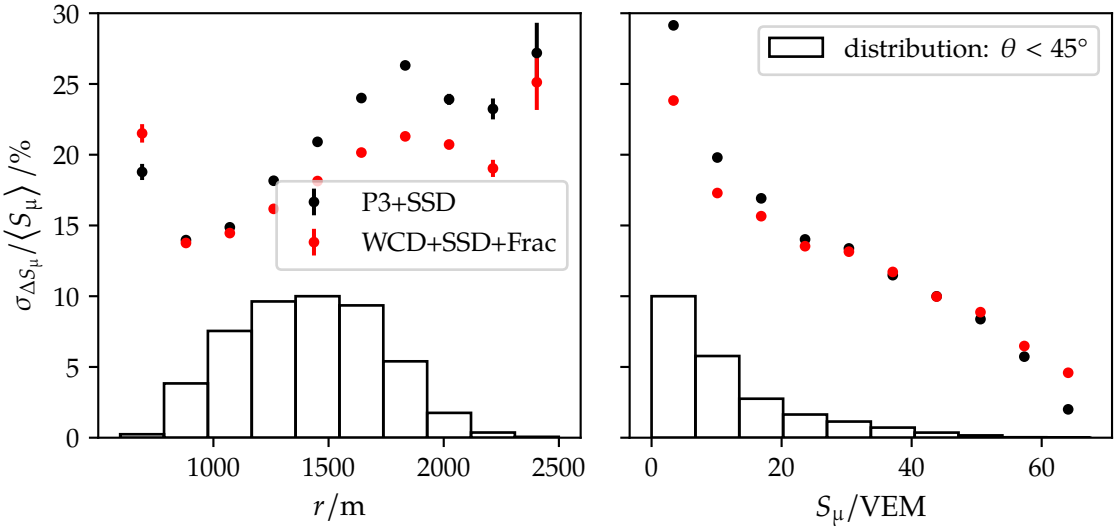


Figure 6.37: Resolution of the predictions of P3 and NN-based model as a function of the shower plane distance r (*left*) and the muonic signal S_μ (*right*) with added SSD features. We compute the bars like in Fig. 6.34. We use only events below 45° to test in the phase space in which the SSD works better. Again, we find that the NN clearly outperforms P3 for low S_μ. The result is very similar to that in Fig. 6.34.

muons above an energy threshold of about 1 GeV can reach the detector. Like in the WCD, single muons leave behind sharp peaks in the Silicon photomultipliers (SiPMs) of the UMD. This behavior can be used to count the number of muons traversing the detector during a shower event. Using a threshold, the regular Field Programmable Gate Array (FPGA) traces are converted into binary traces. These can be combed by a counting algorithm that searches for specific patterns which a muon would leave behind. Hence, this algorithm is called the muon counter[15] [A:27].

The UMD stations are scintillator detectors with 64 scintillator bars that can trigger independently of each other. The detector is rectangular, covering an area of $10\,\mathrm{m}^2$. These bars are arranged in two sets of 32 bars of 4 m in length aligned to the long side that lie on opposite sites of the detector [A:27]. For inclined showers that are not parallel to the bars, muons may traverse multiple neighboring bars, resulting in overcounting. We call this process corner clipping. To obtain an unbiased estimate of the number of traversed muons, we have to correct for corner clipping in the counting strategy which is called the corner clipping correction. A common approach is here to compare the timing of the signals of neighboring bars. If two of them have been triggered at the same time, there is a chance that this was done by one muon.

The base muon counter relies on pattern matching that only accounts for one bar. With the help of NN, we can check if (directly) correlating neighboring bar information is beneficial for this counting process. For this, we have to use a different network design to that of the networks used for analyzing WCD and SSD traces in the previous sections (see Fig. 6.38). We have to extend the network to correlate information of time bins of neighboring bars. It can be split into two sub-networks; one solely to extract information and a second one to use the newly gained information for inference. We use a setup based on convolutional layers for the first part. In the second part, we introduce important information to the NN, such as the zenith angle. Since, we have two groups of 32 bars we can use the same sub-network to extract information from both bars and then average over their outputs.

The data used in this section has been kindly provided by Joaquín De Jesús [A:28]. In the data set, a majority of the events do not contain muons. We use only detector responses that correspond to non-zero muons. Including them would overweight the non-muon case too much, making it much harder to get a valid training process. In Fig. 6.39, we show the distribution in our training data set. The highest weight lies in single muon signals.

The precision $\sigma_{\Delta N_\mu / N_\mu}$ of the relative muon content outperforms that of the corrected muon counter for a wide range of muon numbers (see Fig. 6.40). Unfortunately, the network does not handle small numbers of muons very well. Even though the precision is still better, we obtain a strong bias for the first bin. In Fig. 6.41, the same behavior is also seen in the bins over the distance to the shower axes. In Fig. 6.39, we find that the negative bias is most likely due to the low-muon cases in these regions.

Hence, the relative bias in both examples is consistent with a small negative value. The NN slightly over-predicts the number of muons (see Fig. 6.41). We find that an NN-based approach for counting the number of muons is at least equivalent to the regular correction for a larger number of muons. Consequentially, the network must have picked up on the corner clipping. This could yield insights into advanced algorithms for this kind of correction. Still, the performance of the network at low muon numbers is sub-par. However, this could most likely be fixed using a more appropriate loss and a better network design. Right now, it is more of a brute force approach designed to check the feasibility. Nevertheless, to ensure that all this is a valid statement we require much more simulations in a wider phase space!

[15] There is also a method based on integration of the ADC signal [A:26]. However, we do not focus on it.

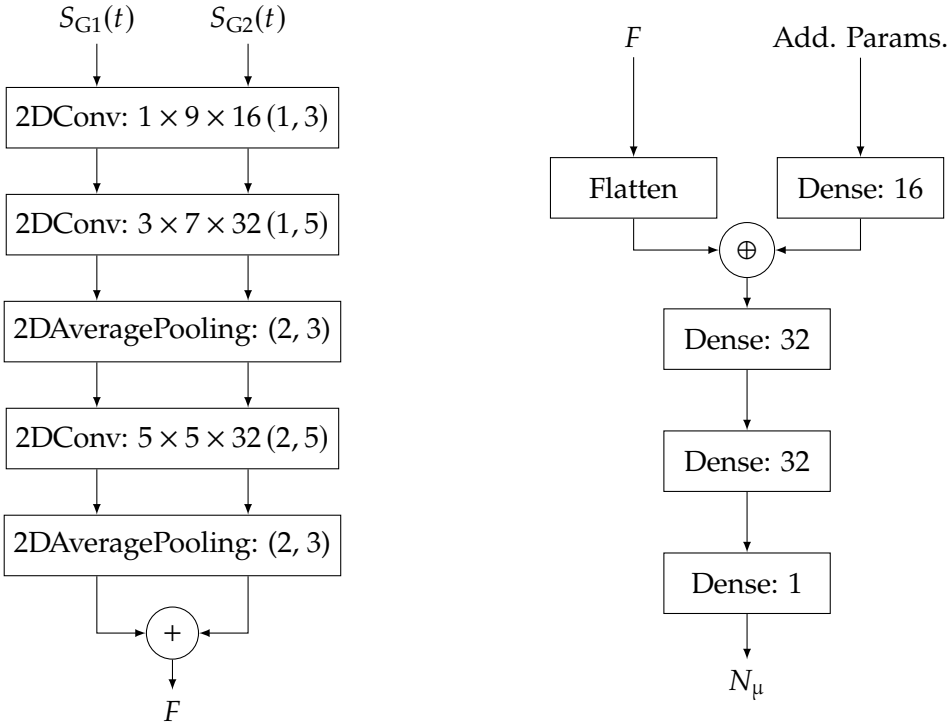


Figure 6.38: Illustration of NN architecture used for the feasibility study in Sec. 6.4. For readibility we have omitted the activation functions. *Left*: Sub-network architecture for the extraction of features from binary signals of the two groups G1 and G2 of bars in the UMD detector. The dimension and number of filters of each of the 2D convolution layers is indicated after the colon. The first two number represent the spatial and time dimension, respectively, and the last is the number of filters. In each convolution filter we use strides which are given in the parentheses. The numbers in the pooling layers signify the pooling sizes. The same weights are used for both groups. After the last layer both outputs are summed up giving us the extracted features F. *Right*: Subnetwork architecture used to predict the number of muons N_μ from the features extracted from the binary traces and the additional parameters defined in Sec. 6.4. The number of nodes in each dense layer is indicated by the number in the boxes. The $\oplus$ denotes a concatenation of the flattened F and the output of the dense layer.

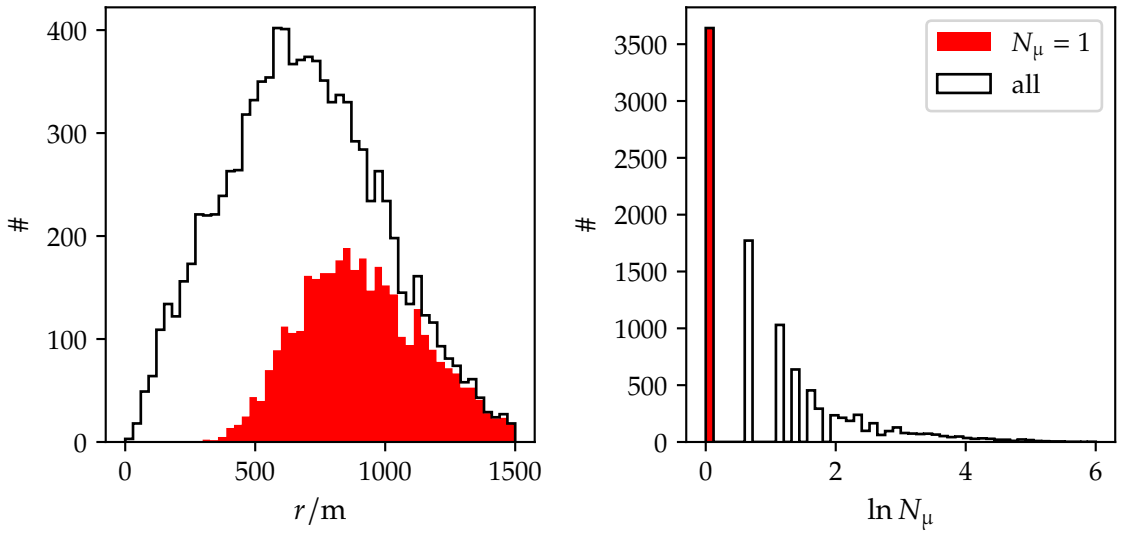


Figure 6.39: Distribution of the data set used in Sec. 6.4 in bins of the shower-plane distance r (*left*) and logarithmic muon number $\ln N_\mu$ (*right*). The red filled areas corresponds to the single muon part of our data set. They make up a large part of the data set.

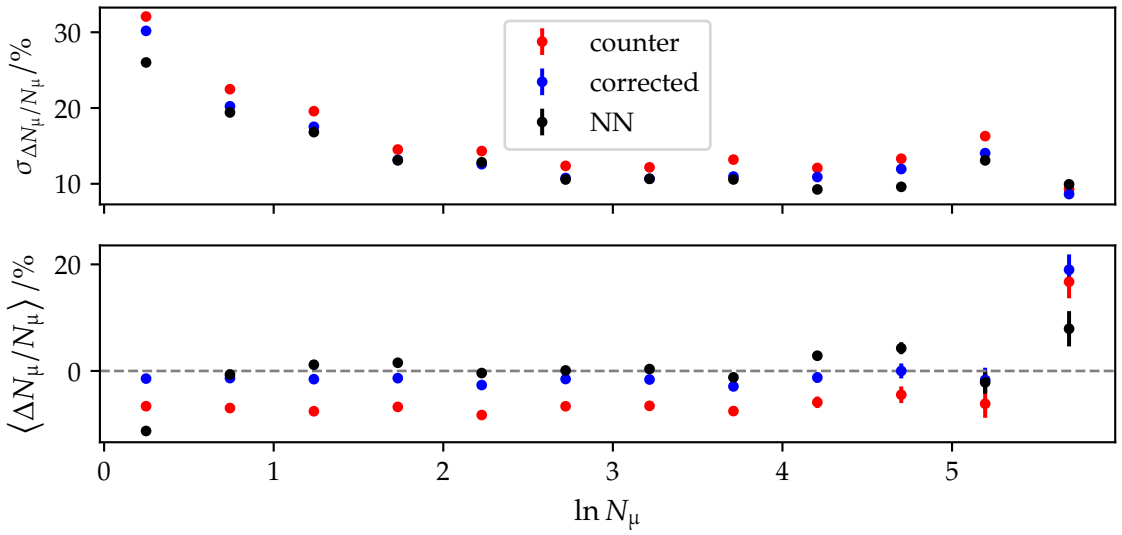


Figure 6.40: Precision of relative error (*top*) and mean relative error (*bottom*) in bins of $\ln N_\mu$. We cut the test data set to include only values below a shower distance of 1500 m. Overall the NN based approach performs slightly better in terms of precision than the regular muon counter and lies on par with the corrected version. Only in the first bin dominated by single muon events we find a strong negative bias.

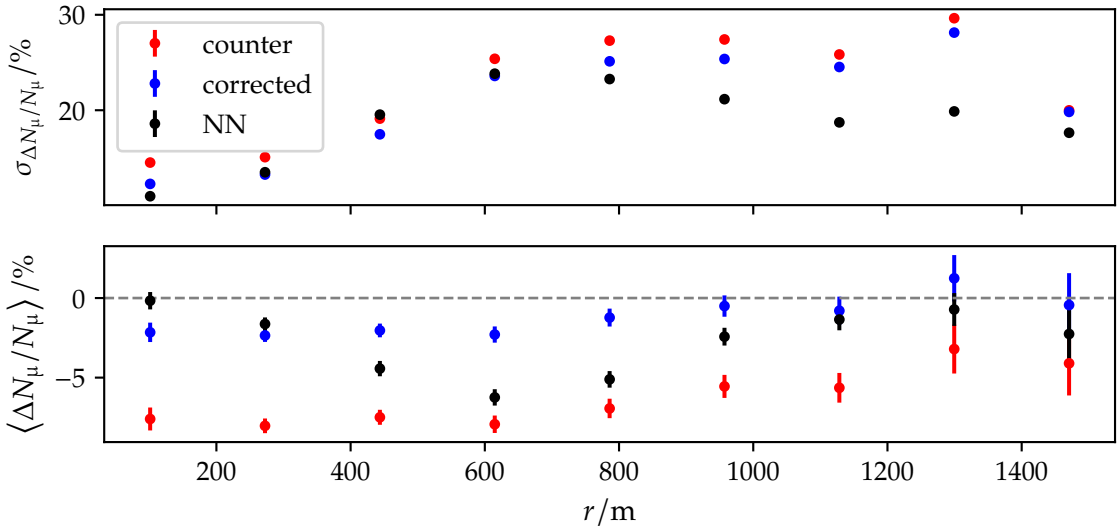
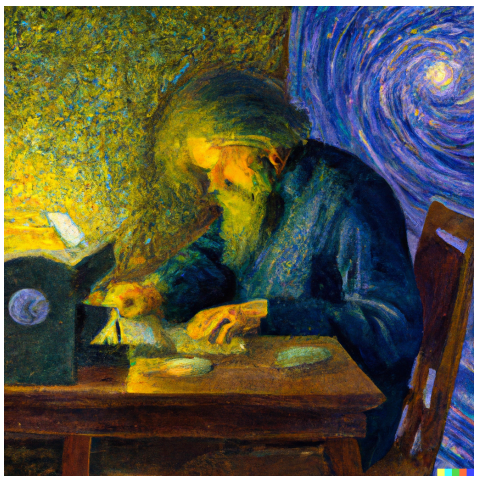


Figure 6.41: Precision of relative error (*top*) and mean relative error (*bottom*) in bins of r. As in Fig. 6.40 the network seems to produce very precise results. The far worse bias is an effect of the over-estimation of low muon numbers.

7 Extraction of Global Shower Properties from Simulated Shower Footprints



If we knew what it was we were doing, it would not be called research, would it?

(Albert Einstein)

DALL·E 2 prompt:

A van [G]ogh painting of an astroparticle physicist working on his computer to unravel the mysteries of the [U]niverse.

Shower footprints measured by the SD grid provide a rich set of complex data in the form of time signal traces. These time traces might contain hidden correlations which have not yet been discovered and exploited by regular analytic approaches. In this chapter, we perform a data-driven analysis of the shower footprints of air shower events using NNs.

Our main objective is to reconstruct various high-level observables that are related to the primary particle mass, such as the depth of the shower maximum X_{max}, directly from the shower footprint using only the time traces from triggered stations of the SD and shower observables reconstructed by the standard algorithms of the $\overline{\text{Off}}\underline{\text{line}}$ framework. MC input values are only used for cross-checks. The architecture of the NNs used in the analysis is based on AixNet and optimized during an exhaustive study of its hyperparameters to fit the shower footprint standardization approach discussed in Sec. 5.3.3.

We investigate the capability of NN-based approaches to reconstruct the zenith angle θ, the energy of the primary particle E, the depth of the shower maximum X_{max}, the relative muon content R_{μ}, and the logarithmic mass number $\ln A$. In all cases, similar base architectures are used. The study of the reconstruction of the zenith angle and primary particle energy is seen as cross-checks. To reconstruct the other observables, the NN based on the architecture need to be able to extract information about both observables since they are directly related to the signal distribution in the shower footprint.

We start the analysis in Sec. 7.1 by an exploration of the hyperparameter space of a NN architecture based on the architecture of AixNet. From this study, we obtain optimized versions of the base architecture and an improved training procedure which is used in the subsequent sections. We continue in Sec. 7.2 by confirming that the changes in the NN-setup have improved the quality of predictions on the targets defined above. In the course of this, we select the NNs best suited to be used on measurements. Afterward, in Sec. 7.3, we develop uncertainty estimates for the predictions of NNs based on the employed architecture and define systematic errors due to the unknown hadronic interactions at the highest energies and

the unknown composition of CRs. In Sec. 7.4, we complete the simulation study on the SSD detector with an event-level analysis. Adding it to the NN setup as an additional input, we study the improvement compared to NNs trained without the additional information. As in Sec. 6.4, we conclude this chapter with another feasibility study. In Sec. 7.5, we analyze in which way we are able to separate hadron-induced air showers from purely electromagnetic air showers, which have been induced by high-energy photons using the networks defined in the preceding sections.

7.1 Effect of the variation of hyperparameters on neural network predictions

There is no straightforward recipe or algorithm to find a suitable NN architecture for a given problem. We need to train multiple NNs and evaluate their predictions to evaluate an architecture. Since the training process for each NN takes a long time, it complicates grid searches or minimization procedures, making standard approaches unfeasible. Therefore, we use the following procedure to search for an architecture and training setup that produce 'well-performing' NNs. After defining a base architecture, we investigate how the change of a single hyperparameter Υ_j[1] impacts the predictions of a NN trained with the changed hyperparameter. For each value chosen for the hyperparameter Υ_j, we study ensembles of N_{ens} NN models and compare their predictions to ensembles of models trained with a different value of the hyperparameter. To evaluate the goodness of the predictions, we estimate the precision and accuracy of the predictions of each NN in the sets, using the standard deviation (see Eq. (4.38)) and the proton-iron bias (see Eq. (4.39)) of the differences between the target value and the prediction. In the end, we take the best-performing set of Υ, assuming that all of the network tests "factorize".

To reduce the training and inference time, we use, if not otherwise stated, the data set defined in Row 5.5.a for training and subsequent tests. Therefore, we evaluate each NN on the same validation and test set. The shorter training time enables us to train a large number of networks. All NNs in the subsequent tests are trained to predict the depth of the shower maximum X_{max} which is the only mass-separating observable of an air shower that is directly measurable. Hence, the predictions can be cross-tested using the measurements of the FD detector. We do not perform all of these tests on the other targets. We assume that all the dependencies we find in this section are also valid for other targets. This assumption holds for primary energy, [A:29].

7.1.1 Baseline architecture for event-level prediction

We encode the shower footprint data in a grid of size $M_s \times M_s$, as discussed in Sec. 5.3.2, and use a trace length of L_t bins. For the baseline architecture, we choose $M_s = 5$ and $L_t = 120$. If not mentioned otherwise, we use ReLU activation functions (see Sec. 4.2.2.A).

The network architecture used in this work is heavily inspired by the AixNet architecture. We split the baseline architecture into three sub-networks using the same naming convention as in Sec. 4.3.2.A: a Trace Feature Extractor (TFE), a Spatial Correlation Analyzer (SCA), and a Feed-Forward Predictor (FFP). We denote them as $\mathcal{AR}_{\text{i}}$, $\mathcal{AR}_{\text{ii}}$, and $\mathcal{AR}_{\text{ii}}$, respectively.

The primary task of $\mathcal{AR}_{\text{i}}$ is to extract n_f features from each of the traces in the encoded shower footprint. To treat every trace the same, we use a weight-sharing technique. For each trace, the same sub-network is used for the feature extraction. This procedure ensures that the extracted features are consistent. We will analyze two fundamentally different $\mathcal{AR}_{\text{i}}$.

[1] Note that we count data preparation of our inputs as one of hyperparameter.

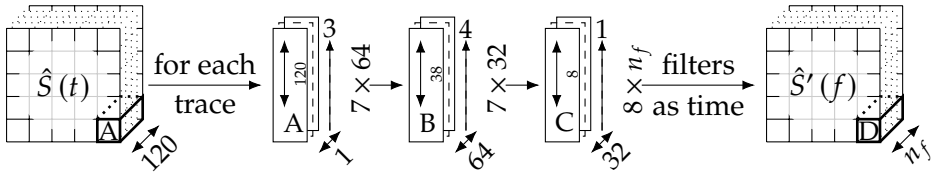


Figure 7.1: Illustration of the architecture of the CNN-based sub-network used for $\mathcal{AR}_\text{i}$. The network uses the same weights and layers for each trace. In total there are three convolution layers which gradually reduce the size of the time dimension to one. The $n \times m$ show the convolution filter size n (in the time dimension) and number of used filters m used in each layer. The number above the vertical arrow indicates the used stride and the small number right of the up-down array denotes the current size of the time dimension. We use the values in the channel dimension after the last convolution as new features and remove the remaining singleton dimension of the time dimension.

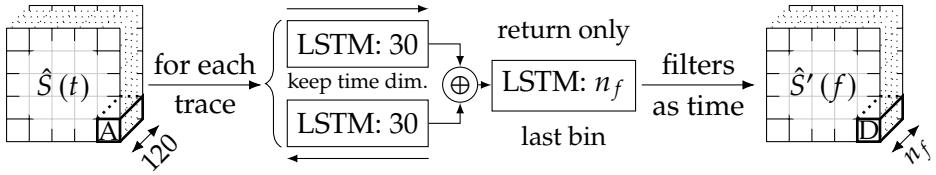


Figure 7.2: Illustration of the architecture of the RNN-based sub-network used for $\mathcal{AR}_\text{i}$. The network uses the same weights and layers for each trace. In total there are two LSTM layers. The number behind the colon indicates the number of units. The first LSTM is a bidirectional LSTM (see Sec. 4.2.3.D) which is indicated by the arrows. It returns the entire sequence for both directions. Hence, for any trace size L_t we obtain an output of the form $L_t \times 60$. The second LSTM layer uses this as input. It returns only the last element of the sequence. Therefore, the output has the size $1 \times n_f$. After dropping the singleton dimension, we use the channel dimensions as features.

First, a CNN-based sub-network (see Fig. 7.1) and a RNN-based sub-network (see Fig. 7.2). The CNN-based sub-network works only for traces of 120 bins due to the special choice of hyperparameters that reduce the 120 bins in the time dimension in three steps to 1. On the other hand, the RNN-based sub-network works for any trace length due to its recurrent architecture. Note that a RNN-based $\mathcal{AR}_\text{i}$ significantly increases the training and inference time due to the inferior potential for parallelization. Due to historical reasons, we have chosen two different values for the number of features n_f extracted by both sub-network types. The CNN-based and RNN-based sub-network extract 10 and 16 features, respectively. These features are saved in the channel dimension of the two-dimensional footprint.

The output of $\mathcal{AR}_\text{i}$ is concatenated along the channel dimension to the memory map of the n_a station- and event-level observables (see Fig. 4.6). Station-level features are encoded in the same manner as the time traces (see Sec. 5.3.2). In the baseline network, we use only the encoded trigger times (see Sec. 5.3.1.A) and a boolean mask that evaluates only to one at the position of triggered stations. Since event-level features, like the zenith angle, are scalar values, we encode them by setting each value in the rectangular encoding space to the value of the event-level feature. Before this mix of trace-based and additional features is used as input for the second part of the network, we apply an element-wise mask to it. We zero the value at all positions in the encoding that do not correspond to a triggered station. The architecture $\mathcal{AR}_\text{ii}$ is based on dense convolutions (see Sec. 4.2.3.C). In each step, we use a two-dimensional convolution with n_s filters of a filter size 3×3. We stack n_d of these dense convolutions increasing the channel dimension after each step by a factor of two. As final layer of the sub-network, we use a final two-dimensional convolution with n_x filters of the

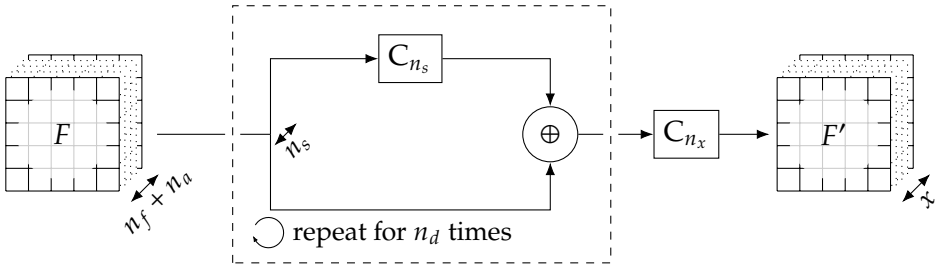


Figure 7.3: Illustration of the architecture of the SCA. The sub-network is comprised of multiple two-dimensional convolution layers. In contrast to Fig. 7.1, the filters have a spatial extend. They relate the different bins in the memory map (see Fig. 5.6). No strides are used in the convolution layers. For each convolution the padding is set to same to prevent the memory size from shrinking. The network uses a form of skipping connections. In each sub-layer the original input is concatenated to the output of the convolutional layer that uses the same number of filters as the time bins doubling them for each subsequent layer. This process is repeated until the output size is n_x. After another convolutional layer we end up with a output array of the size $M_s \times M_s \times x$ (see Eq. (5.6)).

filter size 3×3. The architecture of the spatial analyzer is depicted in Fig. 7.3.

We keep the architecture $\mathcal{AR}_{\text{iii}}$ of the prediction unit as simple as possible. In each network, we predict only a single target value. Hence, we flatten the output of $\mathcal{AR}_{\text{ii}}$, apply a dropout layer with a dropout fraction of d_f, and use the output of the dropout layer as the input for a final dense layer with one node.

We set up the basic training process as follows: We train on the advanced loss function defined in Eq. (4.17). We use batches of N_b events in each training step and train for a maximum of 100 epochs. After each epoch, the TF framework evaluates the loss on the validation set. If the validation loss does not improve for 4 sequential epochs, the training rate is automatically reduced by a factor of 0.8. Moreover, if it does not improve for 7 sequential epochs, the training process stops. This setup ensures that most NNs end the training process well below the 100 maximum epochs. We use the optimizer Adam with an initial learning rate α of 0.0022.

In Appendix A.6, we have summarized the base value of all hyperparameters that we vary in this section. Only if mentioned otherwise do we deviate from these baseline values.

7.1.2 Shower footprint standardization

In this section, we show that the footprint standardization procedure discussed in Sec. 5.3.3 improves the predictions of NNs. We do this by training 50 NNs on a non-standardized data set and 50 NNs on a standardized data set. For a direct comparison, we have selected the two networks of both ensembles that exhibit a minimal standard deviation of ΔX_{max} in the logarithmic energy range $[18.5, 20.2]$. In Fig. 7.4, we compare the precision of the two networks binned in logarithmic energy. In each energy bin, the precision of the NN trained on standardized footprints exceeds the precision of the NN trained on non-standardized footprints. The difference between both NNs is especially large for high energy values reaching consistently an improvement of over 10%. The dotted line in Fig. 7.4 shows the weighted average of the width of the underlying distribution of primaries. Since the values of the precision of both NN are for energy values above $10^{18.5}$ eV well below this line the networks perform much better than an average predictor on the data set.

The improvement between both sets of NNs becomes even clearer if we compare the precision of all NNs in both sets (see Fig. 7.5). The distribution of precision in each energy bin is due to the non-determinism of the training process. From every single training, we

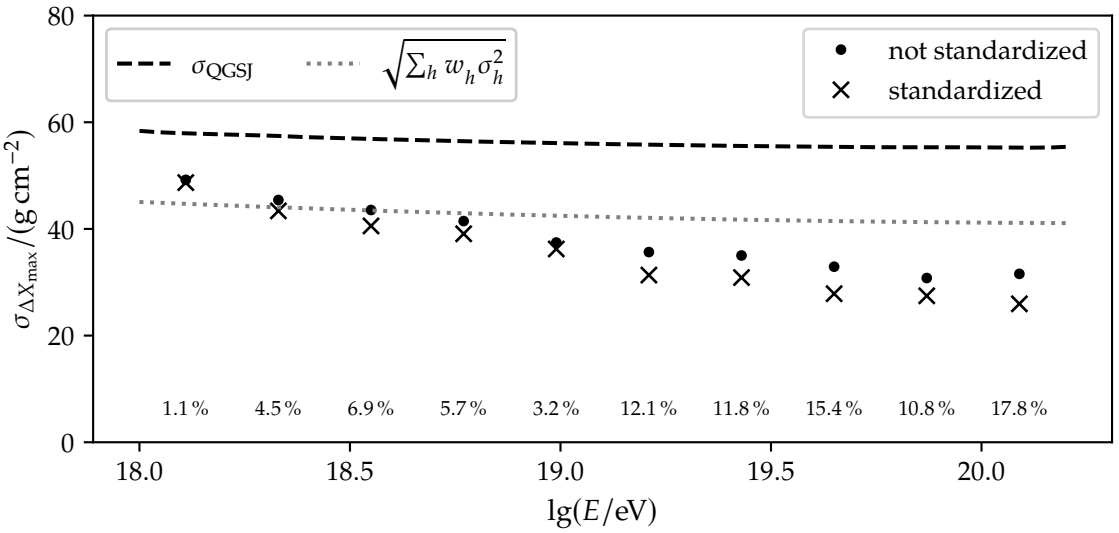


Figure 7.4: Precision of $\Delta X_{\max}$ for two networks trained on non-standardized (points) and standardized shower footprints (crosses) in bins of the logarithmic energy. We have standardized the footprint according to the procedure discussed in Sec. 5.3.3. Both NN models have been selected from two sets of 50 models. We chose the NN with the best precision in the logarithmic energy interval [18.5, 20.2]. To ensure that the predictions of both NNs are better than that of an average predictor, we have added σ_{QGSJ} (dashed line) and the weighted σ (dotted line) of the underlying primaries (see Appendix A.5). The black numbers in the bottom of the plot show the relative improvement of the value in the bin.

obtain slightly different weights, even if we chose the same architecture, training set, and start parameters (see Sec. 7.3.2). The distributions of the precision in each energy bin are shifted downwards, exhibiting on average better precision for the NNs trained on the standardized shower footprints. Therefore, without the requirement of additional resources, the predictions of the NN are likely to improve using the standardization procedure. We gain an average shift of about $2\,\mathrm{g/cm^2}$ over the entire energy range.

Because of the improvement, we use the standardized footprints for the following tests. Note that without training multiple NNs it is impossible to evaluate if a 'well-performing' network has been found.

7.1.3 Effect of varying the NN architecture

For the base architecture, we use very similar hyperparameter values than that of an earlier version of AixNet (see Appendix A.6). Therefore, we vary the hyperparameter values related to the architecture in this section to check if the novel standardization technique favors a certain architectural design.

In addition, to the architecture tests presented in this section, we added more studies in Appendix B.3. This are concerned with the values of the n_f, n_d, and n_s parameter of $\mathcal{AR}_{\mathrm{i}}$ and $\mathcal{AR}_{\mathrm{ii}}$.

A Influence of the architecture of the trace analyzer

The $\mathcal{AR}_{\mathrm{i}}$ part of the architecture described in Sec. 7.1.1 "extracts" relevant information from the traces. There are two very common strategies analyzing time-correlated data: CNN (see Sec. 4.2.3.B) and RNN (see Sec. 4.2.3.D). Hence, we directly compare both sub-networks to each other using the same approach as shown in Sec. 7.1.2.

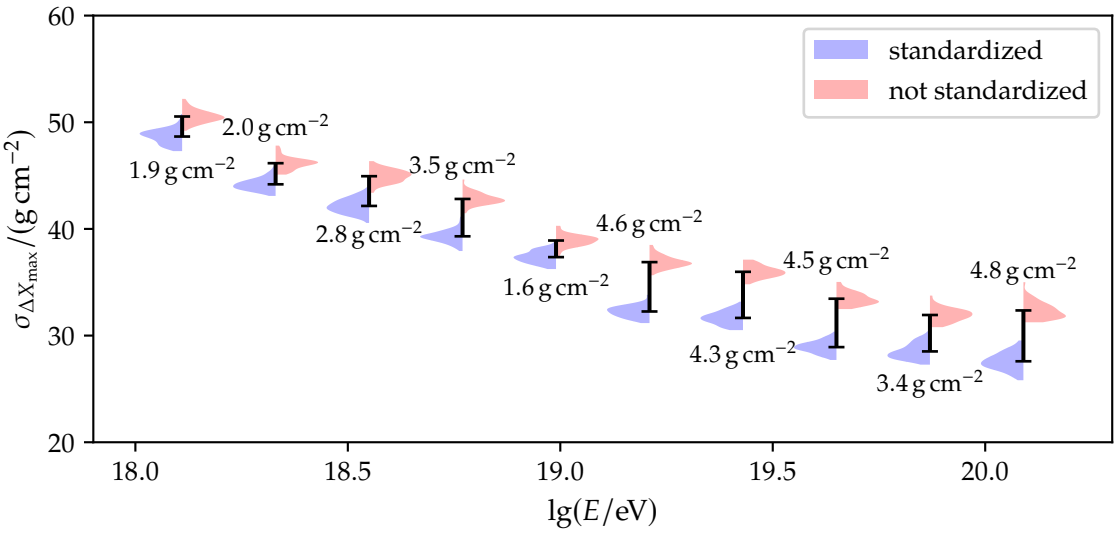


Figure 7.5: Distribution of precision of $\Delta X_{\max}$ for the two sets of NNs trained on non-standardized (red) and standardized (blue) shower footprints (see Sec. 5.3.3). In both sets are 50 unique NNs. The black line connects the average precision of both classes in each energy bin, and its length is quantified by the number next to it.

Since we already have 50 NNs from the study of the shower footprint standardization (see Sec. 7.1.2), we train 50 additional NNs using the RNN-based sub-network defined in Fig. 7.2 as trace analyzer. Selecting the NN with the best precision the logarithmic energy interval [18.5, 20.2] from the set of RNN-based NNs, we can directly compare the NN to that found in Sec. 7.1.2. The predictions of the RNN-based NN outperform that of the CNN-based NN. At energy values above 10^{19} eV the RNN-based NN exhibits an improvement of the precision of over 5% (see Fig. 7.6). However, the gain from using an entirely different sub-network is smaller than that from the standardization.

Similarly to Sec. 7.1.2, this improvement is also visible in the distributions of the precision of both sets of NNs (see Fig. 7.7). The distributions is shifted towards higher precisions for the RNN-based NNs compared to the CNN-based ones. We obtain an average improvement of around 1 g/cm^2 in each energy bin. However, even thought we have changed a major part of the baseline architecture the improvement is comparable small. The distributions of the precision overlap quite strongly which highlight the importance of the training of an ensemble of NNs. Using only on NN for each architecture, we could easily miss the improvement due to the change in architecture. Moreover, since the improvement is comparably small and we have to compare multiple hyperparameter values, we switch to a relative comparison in the following tests.

Even though, the RNN performs slightly better, there is one caveat we have to account for. On average, one of the machines defined in Appendix C.1 spends (4024.6±158.1) s for one network which uses a RNN-based TFE. This is more than 7 times longer than the average training time of the CNN-based sub-network which takes about (541.4±17.7) s. Therefore, we use CNN-based sub-networks if not mentioned otherwise, assuming that the improvements found for one type of TFE hold for the other one. We denote the choice of $\mathcal{AR}_{\mathrm{i}}$ by adding either CNN or RNN to the following plots.

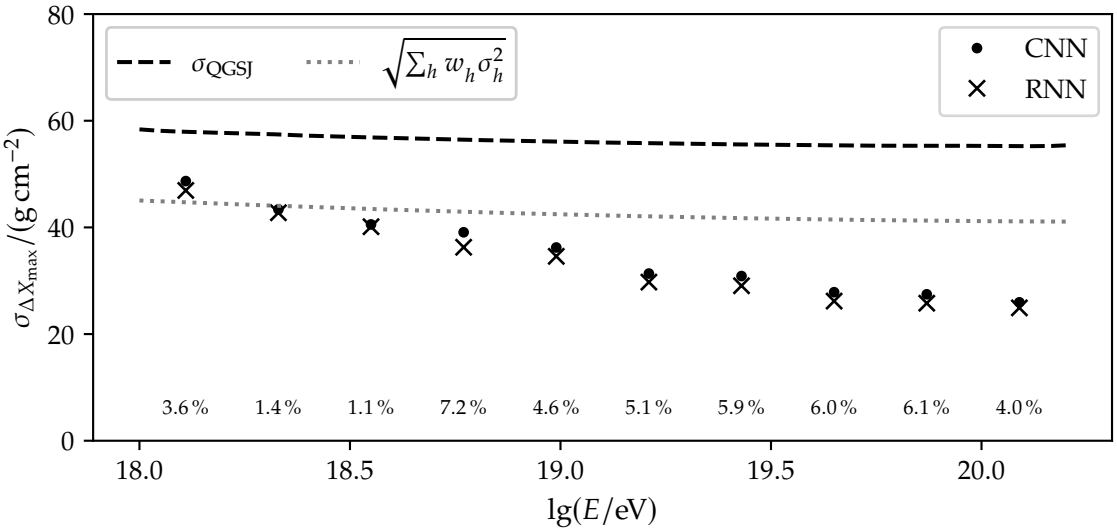


Figure 7.6: Precision of ΔX_{max} for two networks trained with a TFE based on a CNN sub-network (points) and an RNN sub-network (crosses) in bins of the logarithmic energy. Both NN models have been selected from two sets of 50 models by choosing the NN with the best precision in the logarithmic energy interval [18.5, 20.2]. The dashed and dotted lines are the same as discussed in Fig. 7.4. The black numbers in the bottom of the plot show the relative improvement of the value in the bin.

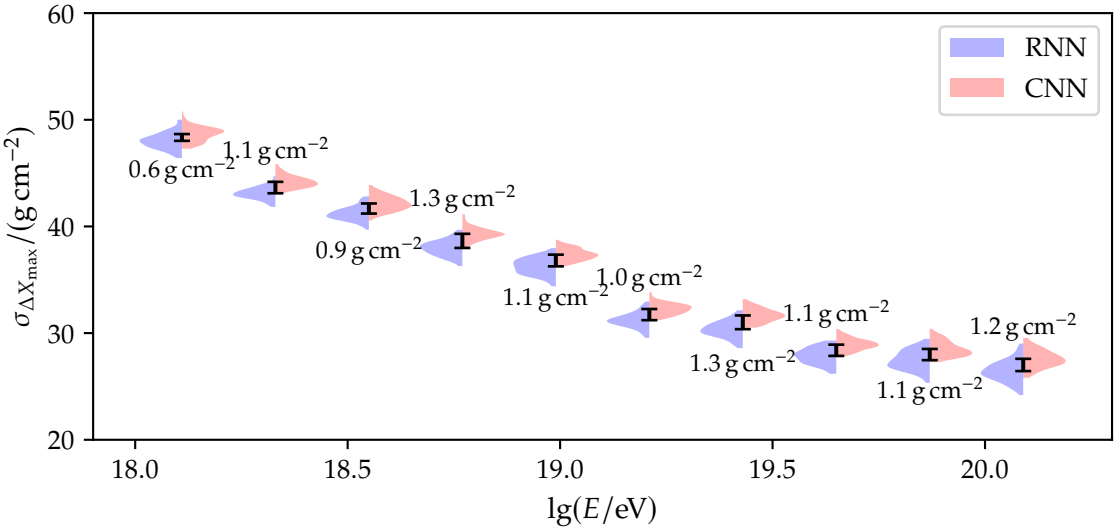


Figure 7.7: Distribution of precision of ΔX_{max} for the two sets of NNs trained with a TFE based on a CNN sub-network (red) and an RNN sub-network (crosses). In both sets are 50 unique NNs. The black line connects the average precision of both classes in each energy bin. The number indicates the distance from each other.

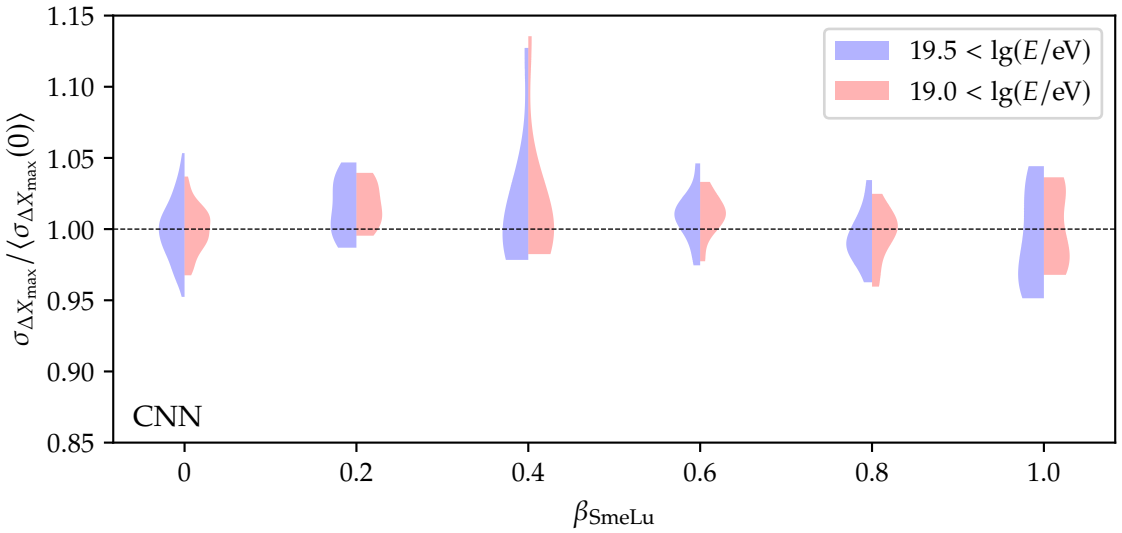


Figure 7.8: Ensembles of the precision $\sigma_{\Delta X_{\max}}$ for NNs trained with different values of β_{SmeLu} normalized to the average precision of the ensemble using $\beta_{\mathrm{SmeLu}} = 0$ which is equivalent to a regular ReLU activation function. We depict the precision in the two integral energy bins of events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

B Using SmeLu activation function in spatial analyzer

In Sec. 4.2.2.A, we have presented an alternative activation function that is claimed to improve the reproducibility of NN-based approaches. Here, we test if this holds for our NN architecture. We replace the ReLU activation functions in $\mathcal{AR}_{\mathrm{ii}}$ with SMELU activation functions investigating the effect of choosing different β parameters. Note that by construction (see Appendix A.4), the SMELU transforms into a ReLU for $\beta \to 0$.

If the SMELU improves the reproducibility, it should reduce the spread of the distribution of the precision or the proton-iron biases of the predictions of the ensembles of NNs. We cannot detect such a reduction (see Fig. 7.8 and Fig. 7.9). There is no noticeable difference in the distributions. Moreover, there is also no benefit for predicting the proton-iron bias. Therefore, we keep the simple ReLU in the final architecture.

C Variation of dropout fraction

In the baseline architecture, we have used a dropout fraction d_f of 0.2. Increasing it forces the NN to use different combinations of connections in the final layer of $\mathcal{AR}_{\mathrm{iii}}$ to predict the depth of the shower maximum. For the small data set we use in this study, increasing d_f also slightly improves the average precision of the networks (see Fig. 7.10). Moreover, there is a minor effect on the proton-iron bias (see Fig. 7.11) shifting the distributions downwards. Still, it is not clear if such an improvement translates to a much larger data set.

7.1.4 Hyperparameters related to the network training

Hyperparameters governing the training process, such as the batch size N_b, directly impact the length, stability, and result of the training process. In this section, we investigate the effect of varying hyperparameters related to the training process.

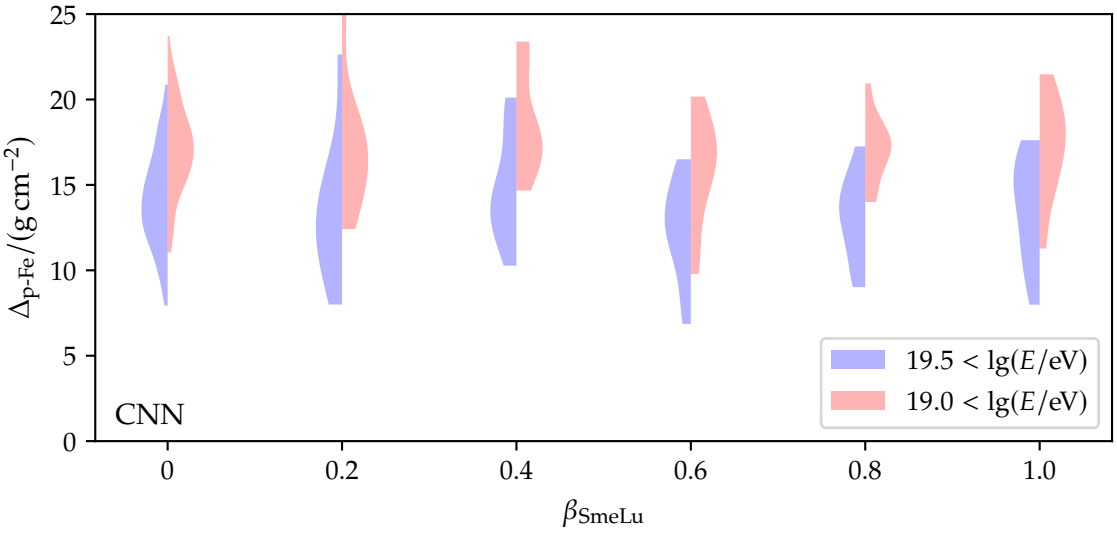


Figure 7.9: Ensembles of the proton iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with different values of β_{SmeLu}. We depict the precisions of two integral energy bins of events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

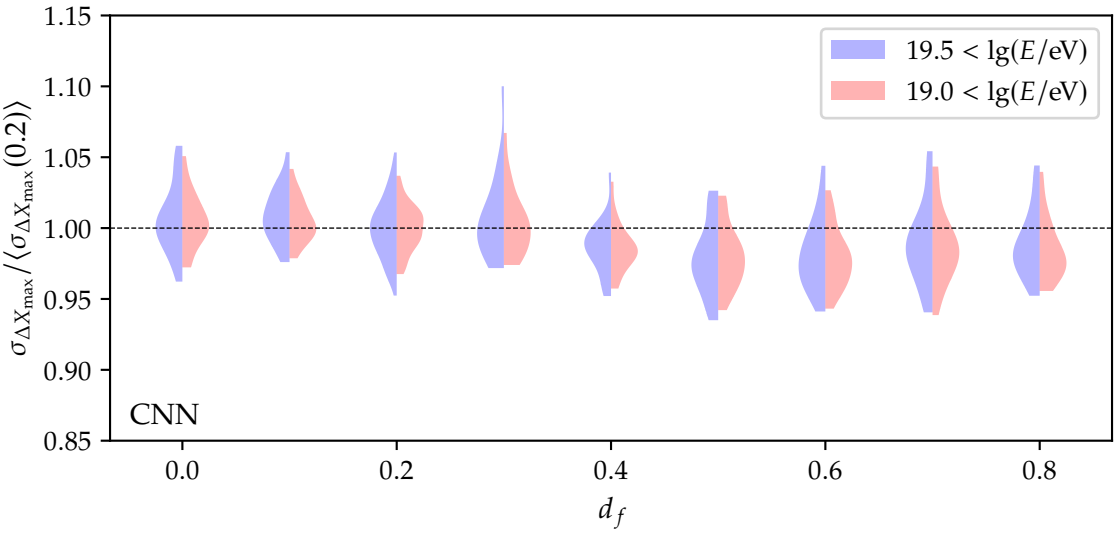


Figure 7.10: Ensembles of the precision $\sigma_{\Delta X_{\text{max}}}$ for NNs trained with different values of the dropout fraction d_f normalized to the average precision of the ensemble using $d_f = 0.2$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

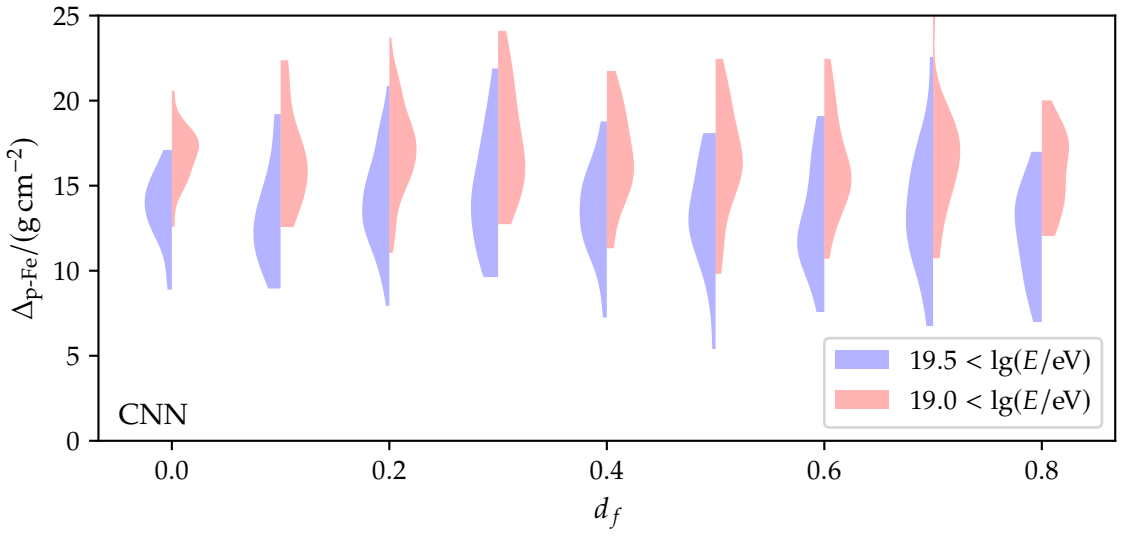


Figure 7.11: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with different values of the dropout fraction d_f for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

A Effect of variation of batch size

We vary the batch size N_b in multiples of two to take advantage of the memory size of the used GPUs. For NNs using the CNN-based TFE, an increase in the batch size improves the expected precision of the predictions (see Fig. 7.12). In the region of 512 up to 2048, there is a minimum of the average value of the precision achieving an improvement of up to 10% if compared to the baseline batch size of $N_b = 64$. For the batch sizes 2048 and 4096, the average precision starts increasing again.

To decide which batch size we use in the final architecture, we cross-check the result with the proton-iron bias $\Delta_{\text{p-Fe}}$ as a function of the batch size. Indeed, increasing the batch sizes reduces the average proton iron bias of the predictions (see Fig. 7.13). At a batch size of $N_b = 512$, the proton-iron bias plateaus. Since for a batch size of $N_b = 2048$, the predictions of NNs exhibit a high precision and low proton-iron biases, we chose $N_b = 2048$ for the final architecture. In addition, we also use $N_b = 512$ as a cross-check.

To confirm that this behavior is similar for NNs based on the RNN sub-network, we repeat the analysis for such networks (see Appendix B.3). In contrast to the CNN-based TFE networks, the NNs based on the RNN sub-network show the opposite behavior. The precision increases when the batch size is reduced (see Fig. B.6). Hence, we choose a batch size of $N_b = 32$ for this network.

B Effect of amount of training data

To estimate the dependence of the predictions of NNs on the amount of training data, we train multiple networks on different fractions f_D of the data set described in Row 5.5.c. As expected, the average precision improves with the available amount of training data (see Fig. 7.14). At around a fraction of 25%, the used data set is as large as the base data set in Row 5.5.a. Hence, the average precision using the base data set is about ten percentage points worse than when using 80% of the large data set. Given enough data, we expect that the precision should be saturated due to the constraints posed by the architecture and the noise in the underlying data set. However, such saturation is not visible in this analysis.

The proton-iron bias improves slightly until a fraction of 50% is reached (see Fig. 7.15). The

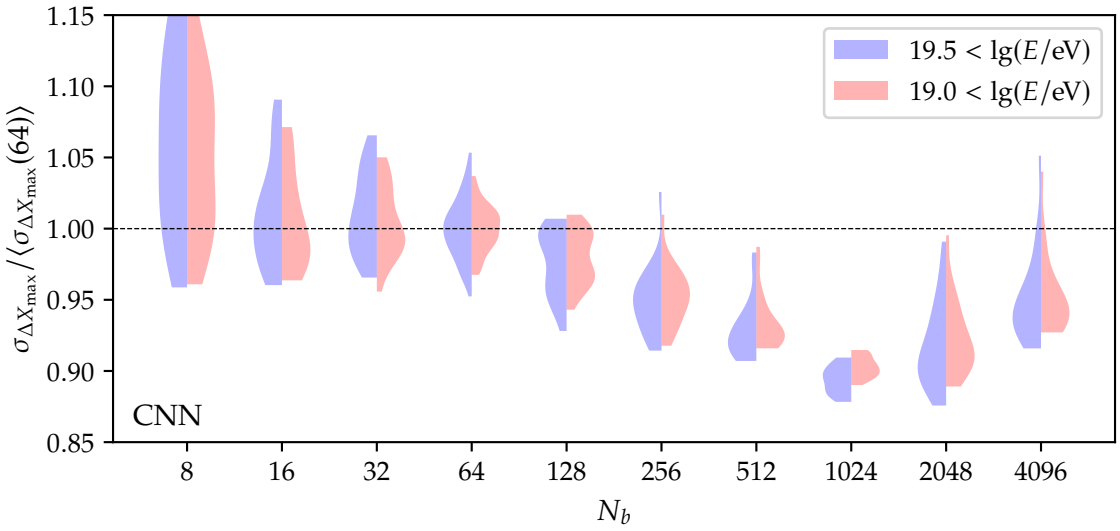


Figure 7.12: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained with different values of the batch size N_b normalized to the average precision of the ensemble using $N_b = 64$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

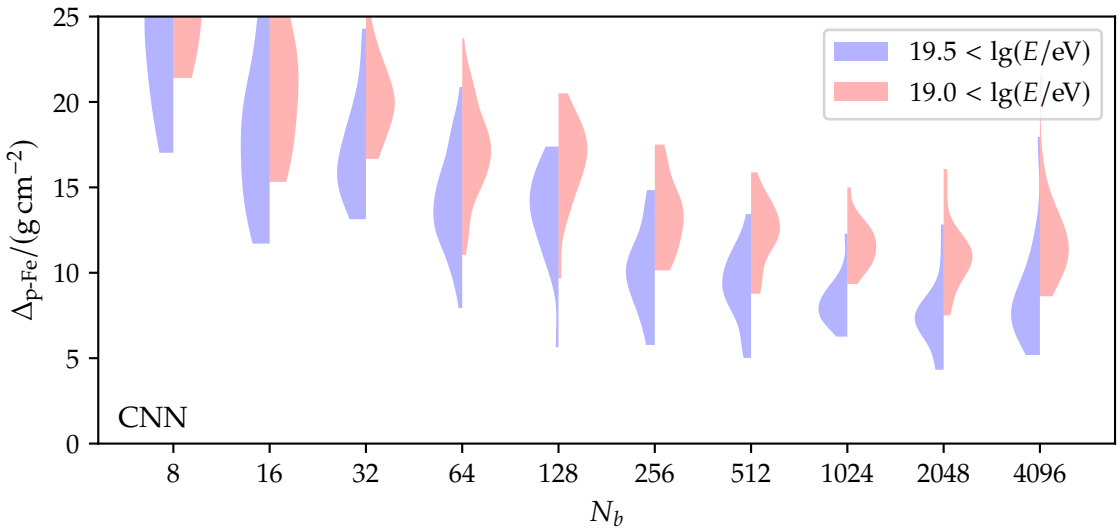


Figure 7.13: Ensembles of the proton-iron bias $\Delta_{p\text{-}Fe}$ for NNs trained with different values of the batch size N_b for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

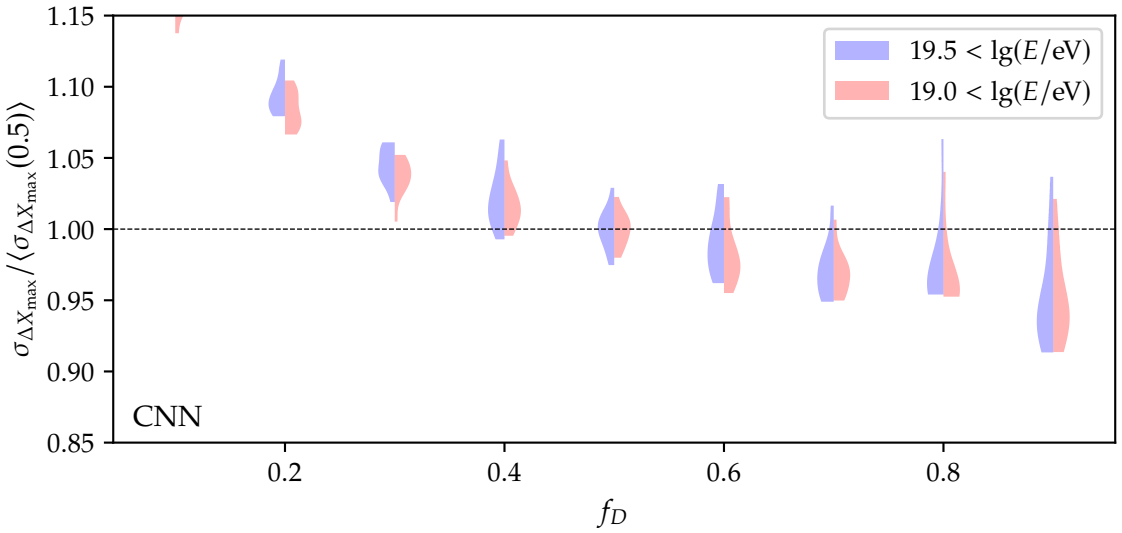


Figure 7.14: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained with different amounts of training data determined by the fraction f_D. The precision is normalized to the average precision of the ensemble using $f_D = 0.5$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

expected proton-iron bias for the fractions around 25% is noticeably worse than for a batch size of $N_b = 64$ in Fig. 7.13. This increase in proton-iron bias is due to the increased size of the test set, increasing the variability of the air showers.

7.1.5 Hyperparameters related to the encoding of the event-level data

The input data of our base network is encoded in a 5×5 grid using a trace length of 120 bins corresponding to 3 µs of UB trace data. We chose the encoding size of $M_s = 5$ to ensure a fast training process and the trace length comparable to the preceding analysis in [P:104]. In this section, we check how these values predict the predictions of ensembles of NNs. Since we vary the trace length, we use the RNN-based TFE in this section.

A Variation of trace length

Since there is no straightforward way to estimate the amount of important information, we need to know how much of a trace is required to achieve an optimal prediction. We probe this by training ensembles of NNs using traces of different trace lengths L_t. Due to the constraint of the architecture of the CNN-based TFE, we would need different architectures to reduce the time dimension over the convolutional layers (see Fig. 7.1). Consequentially, we only train NNs using the RNN-based sub-network (see Sec. 7.1.3.A).

In Fig. 7.16, we compare the distributions of the precision of the predictions of 20 networks trained with different trace lengths L_t. Above a trace length of 180 bins, the training process becomes partially unstable. For some NNs, the training prematurely stops yielding precision in the integral energy bins above 35 g/cm^2. We remove these networks from this analysis since they would create bi-modal distributions which distort the violin plots. This instability is a sign that the used architecture based on RNN is at its limit. Due to the increased trace length, correlating important information becomes more challenging. Also, for trace lengths below 100 bins, the ensembles become gradually worse. This decrease in precision is most notable for the networks trained on trace lengths of 30 bins indicating that in this region, there is too little information that the NNs could exploit. Between 120 and 150 bins is a sweet

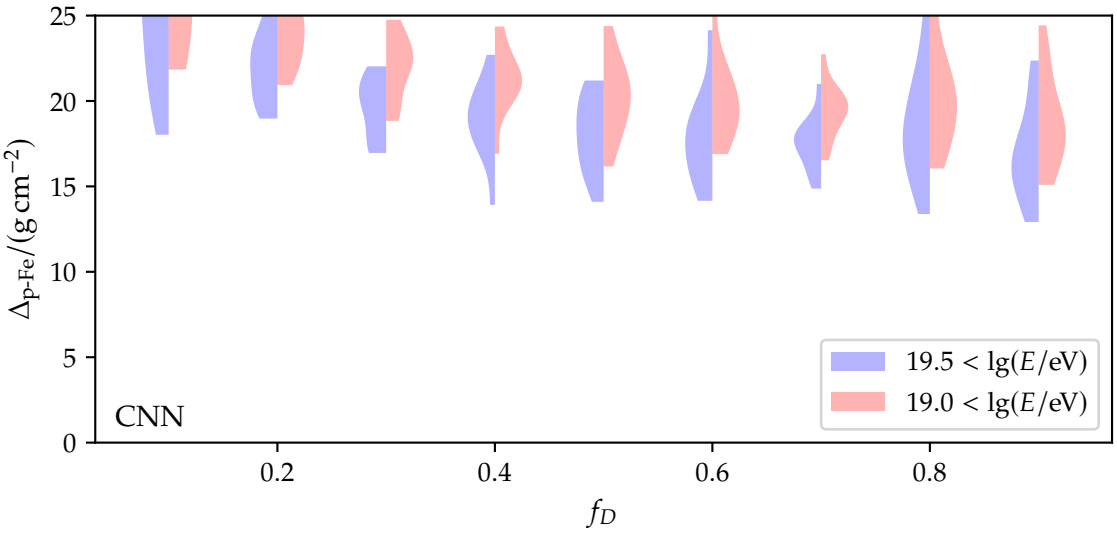


Figure 7.15: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with different amounts of training data determined by the fraction f_D. We show $\Delta_{\text{p-Fe}}$ for two different integral bins using only events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

spot. NNs trained with these trace lengths show no instability. The information content in the trace is large enough to make accurate predictions. Since there is no apparent difference between both trace lengths, we keep using 120 bins for the final architecture.

This choice is also driven by the trade-off between network training duration and performance. Using 150 bins for our study pushes us over the 1 h mark in training time for the reduced data set without showing considerable benefits.

B Variation of encoding size

For AixNetn an encoding size M_s of 13 has been used [A:13]. The large encoding size ensures that almost all triggered stations are encoded in the memory. However, the information content of stations that are far from the impact point of the shower core is greatly diminished. Hence, we favor smaller encoding sizes. To substantiate this reasoning, we have trained ensembles of NNs using different encoding sizes.

In Fig. 7.17, we compare the performance of ensembles of NNs trained on encoding sizes up to $M_s = 11$. For each case 10 networks have been used. Surprisingly, even an encoding size of $M_s = 5$ is unfavorable, yielding a smaller precision than the smallest possible footprint size encoding size of three. For encoding sizes greater than 5 the ensembles show similar results.

The improvement of the precision is also not caused by a trade-off with the proton-iron bias (see Fig. 7.18). This result is somewhat surprising. Adding more information of the shower footprint should be beneficial for predicting observables, such as the depth of the shower maximum. However, it also increases the difficulty of extracting relevant information. Stations far away from the shower core exhibit a lower signal-to-noise ratio than stations near the shower core. If a small amount of additional information is not required to make accurate predictions, the network potentially ignores bins in the encoding that are far from the hottest station. Still, the ignored bins impact the NNs in the TFE, adding noise to the overall prediction.

To back up this idea, we exclude the possibility that this reduction in the precision of the predictions is caused by an over-weighting of smaller shower footprints due to the low energy part of our training data set. We repeat the study for the smallest three encoding sizes for

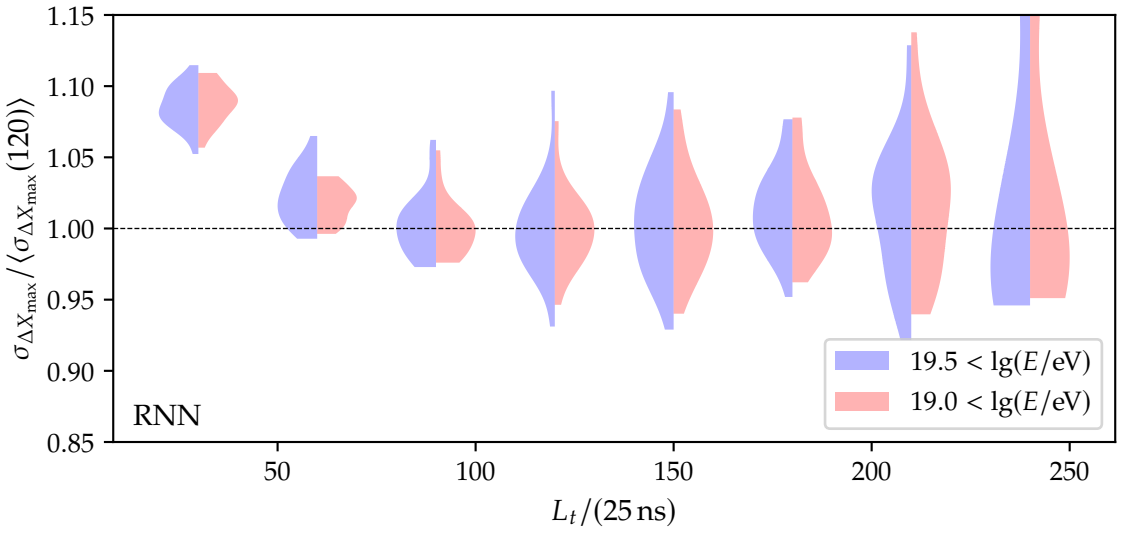


Figure 7.16: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained using traces of different trace lengths L_t. The precision is normalized to the average precision of the ensemble using $L_t = 120$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

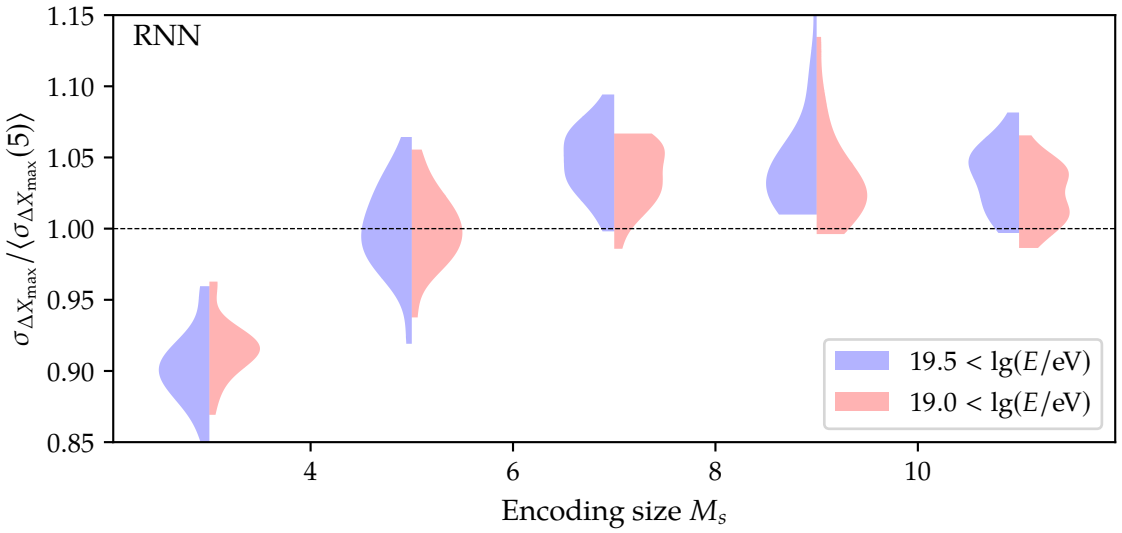


Figure 7.17: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained using different encoding sizes M_s. The precision is normalized to the average precision of the ensemble using $M_s = 5$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

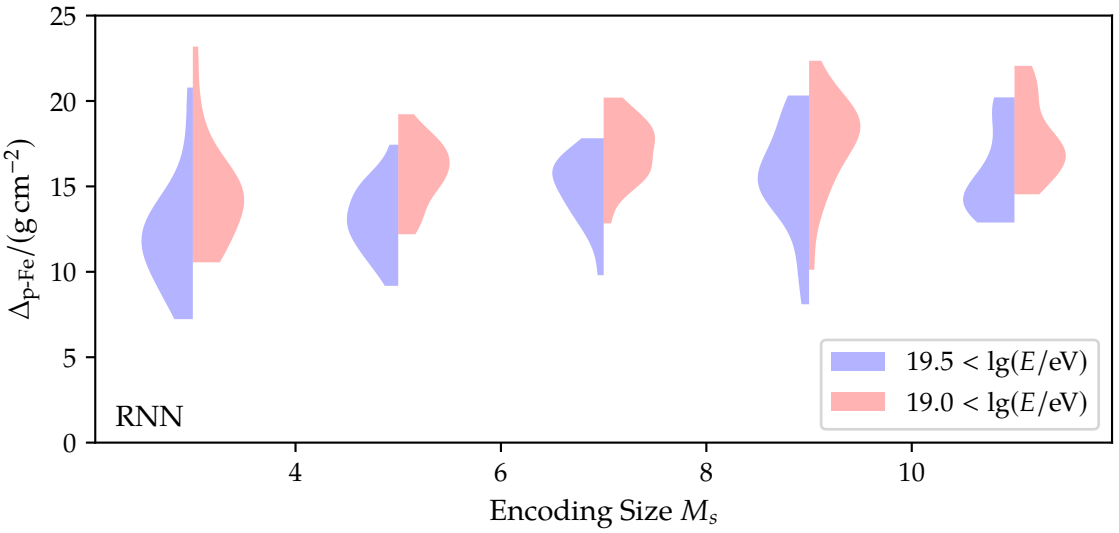


Figure 7.18: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained using different encoding sizes M_s for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

a specialized training set that contains only events with energies higher than 10 EeV. Below 10^{19} eV, many events have less than 7 triggered candidate stations. Hence events tend to not even fill up the first crown (see Fig. 5.2). Even for energies far above 10^{20} eV, the footprint for a 7×7 is, on average, not completely filled out. Despite using only high-energy events, we obtain a very similar result to that of Fig. 7.17 (see Fig. 7.19). The predictions of NNs using the smallest encoding size have higher precision than the NNs using the larger ones.

This result indicates that the most essential part of the shower footprint is right around the impact point of the shower core. The signal in this region carries the information necessary for a good prediction of X_{max}. Potentially, we could build a network that focuses explicitly on these non-core regions of the shower footprint. However, this fine-tuning is beyond the scope of this thesis. We chose an encoding size of 3 and 5 for the shower footprints used for the final We keep the footprint size of 5 to ensure that the small amount of training data does not mainly drive this result.

7.1.6 Using additional observables as inputs

Until now, we solely varied hyperparameters related to the topology and training process. It is not a priori unclear which station- and event-level shower observables benefit the inference process. In this section, we study how additional inputs improve the overall performance of the predictions of NNs.

A Adding new event-level information

We selected the energy E and zenith angle θ as new potential event-level inputs. Both of these observables are directly related to the footprint size and signal strength in the core region. We transform the energy via lg and the zenith with $\sin^2$. These transformations make the distribution of both shower observables uniform. The signal detected in the SD stations depends on $\sec\theta$. Therefore, to test the choice of $\sin^2$, we also test sec as a potential transformation. To estimate how the reconstruction impacts these results, we use the MC values and the reconstructed SD values as inputs.

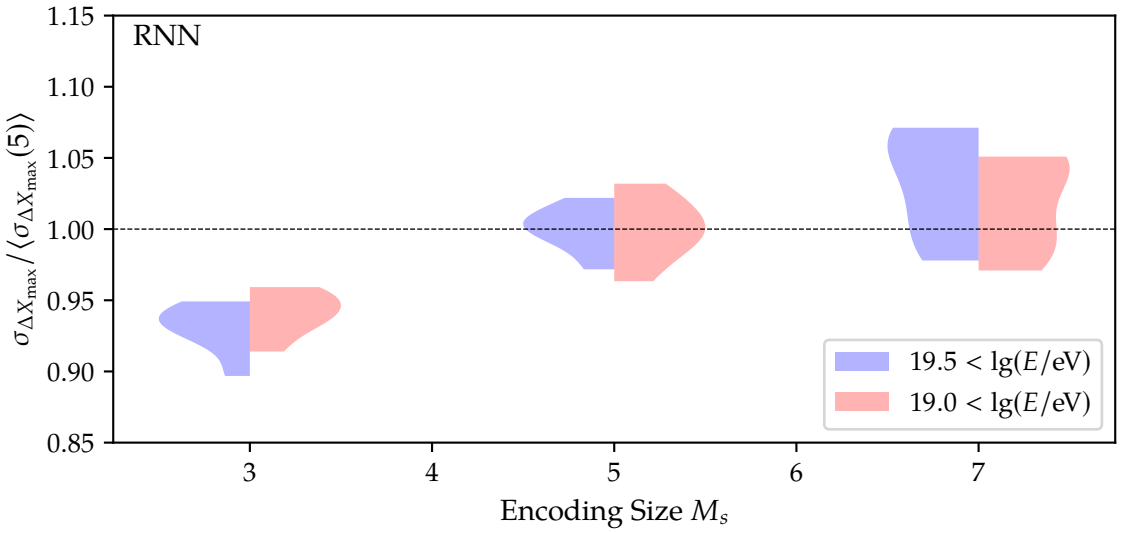


Figure 7.19: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained using different encoding sizes M_s for a data set using only events with an energy above 10^{19} eV. The precision is normalized to the average precision of the ensemble using $M_s = 5$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

Adding the energy to the network has no visible impact on the precision of the predictions (see Fig. 7.20). The precision is equivalent to NNs using no additional information. Since there is a direct relation between the energy to the average value of X_{max} (see Sec. 5.4.2), the networks most likely interpolate this relation from the trace signals. The zenith angle, on the other hand, improves the precision of the predictions of the NNs ensembles considerably. Using the MC values or the reconstructed values yields a visible downward shift of the distributions. The difference between using θ_{MC} and θ_{SD} is minimal. We conclude that the accuracy of the standard reconstruction for θ_{SD} is large enough for this analysis.

Comparing the different transformations used for θ_{SD}, it appears that by using sec the precision of the predictions of the NNs improves slightly. This improvement is most likely a trade-off between the precision and proton-iron bias. In Fig. 7.23, the roles of the $\sin^2$ and sec transforms are reversed. The distribution of the proton-iron bias reaches slightly lower values. We conclude that we should include θ_{SD} as an additional input. Most likely, providing the reconstructed zenith angle gives the network a better handle on the geometry of the shower. We select the $\sin^2$ transformation out of convenience.

B Adding new station-level information

Since there are many possible station-level observables, we have selected a subset based on the experience of working with the data. We classified the used additional inputs as follows. First, we have selected the high- and low-gain flags of each triggered station. We denote them as HG and LG, respectively. If the former is set, the trace is determined by the low-gain channel. If the latter is set, the trace is at least partially saturated. This saturation yields a plateau in the signal trace. Both flags signify that the trace signal behaves slightly differently. It could be beneficial to give this information to the network before the spatial information is correlated. Secondly, we added the risetime $t_{1/2}$ and total signal S_{tot} to the set of investigated station-level information. Both are features directly related to the integral over the trace. If the information of these trace features is essential for the prediction of X_{max}, adding them would free a value of n_f. Forcing the network to extract different features could yield a

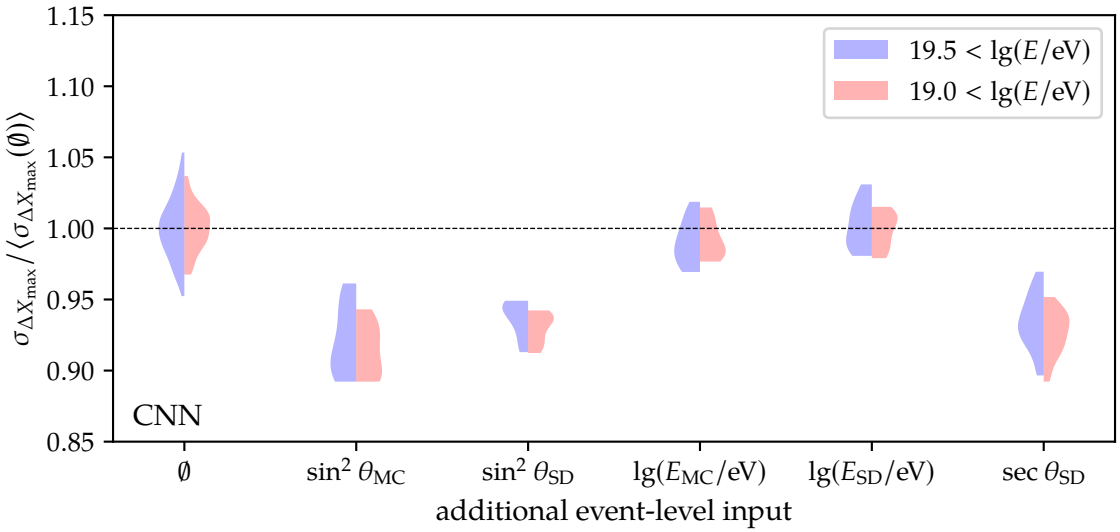


Figure 7.20: Ensembles of the precision $\sigma_{\Delta X_{\max}}$ for NNs trained with additional event-level inputs. The $\emptyset$ refers to the ensemble of the baseline network. The precision is normalized to the average precision of the ensemble using $M_s = 5$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

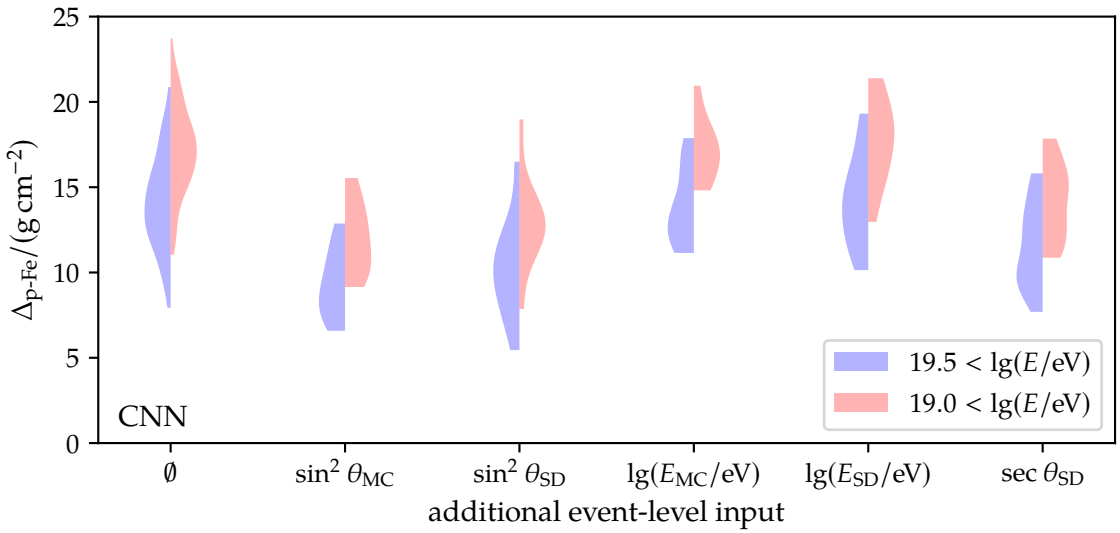


Figure 7.21: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with additional event-level inputs for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red). The $\emptyset$ refers to the ensemble of the baseline network.

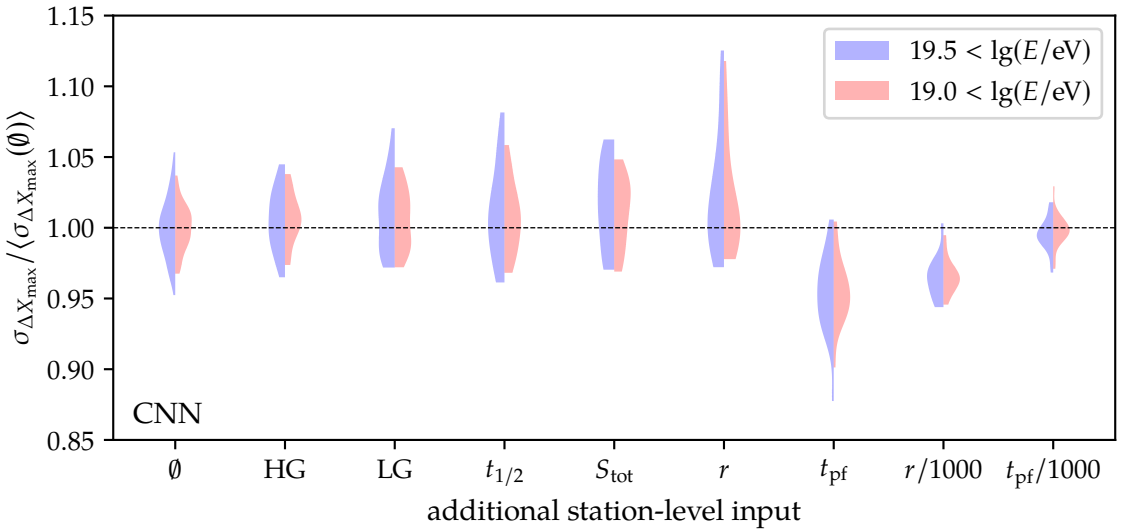


Figure 7.22: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained with additional station-level inputs. The $\emptyset$ refers to the ensemble of the baseline network. The precision is normalized to the average precision of the ensemble using $M_s = 5$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

slightly better result. Finally, we have added the shower plane distance r and the plane front time t_{pf} (see Sec. 5.4.1). Both station-level observables encode geometric information of the station positions. In Sec. 7.1.6.A, this improved the predictions of the NNs considerably. To demonstrate the effect of normalization of our input values, we also consider r and t_{pf} divided by a factor of 1000. Due to the results of the previous section, we do not perform the tests with MC values.

Adding observables from the first two classes does not affect the distributions of the precision or the distributions of the proton-iron bias (see Fig. 7.22 and Fig. 7.23). The precision of the predictions of the networks in the corresponding ensembles is similar to that of the ensemble of the baseline network. Only the ensembles of the networks trained with the plane front time t_{pf} and the normalized shower plane distance $r/1000$ show an increase in precision. This result shows us the great importance of correctly preparing our input data. Using the normalized plane front time $t_{pf}/1000$ or the shower plane distance r does not yield any visible improvement. In this case, this is most likely due to the smallness and largeness of the former and the latter observable, respectively. For example, if an input value is compared to the other inputs, overall substantial, it causes instabilities in the training process and hinders it. To ensure a stable training process, the proper normalization for the inputs has to be found and applied. This instability caused by the large values in r is most likely the reason for the long upward tail of the distribution of the precision of NNs using r directly as an additional input. In contrast to Fig. 7.21, the distributions of the proton-iron bias $\Delta_{\mathcal{E}}$ in Fig. 7.23 do not show significant differences if compared to each other. Only LG and $r/1000$ seem to show a reduced proton-iron bias.

Therefore, we select t_{pf}, $r/1000$, and LG as additional station-level inputs.

7.2 Evaluation of neural network predictions on event-level targets

In the last section, we have demonstrated that by changing various hyperparmeters the predictions of NNs become more precise and exhibit less proton-iron bias. Since we are not

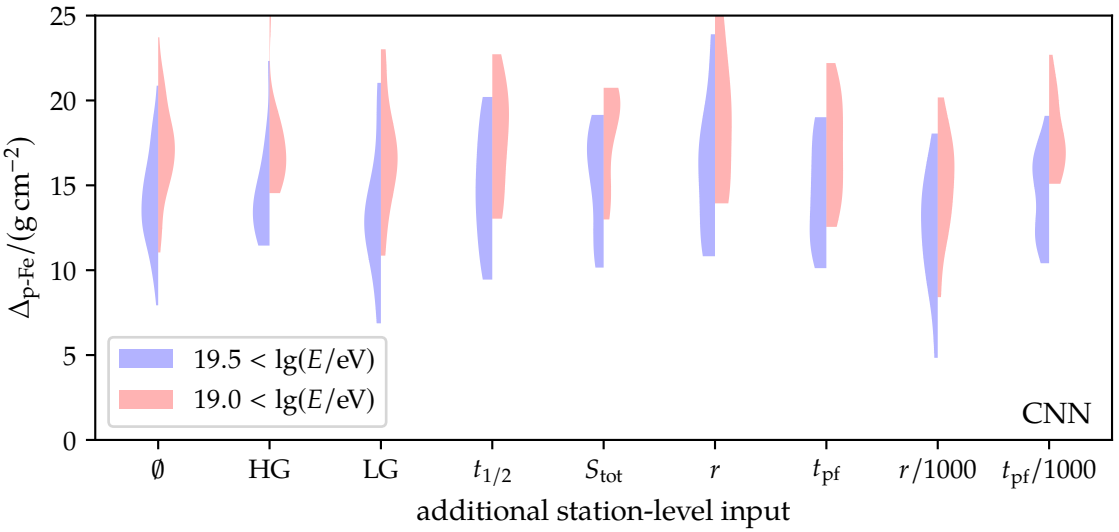


Figure 7.23: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with additional station-level inputs for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red). The ∅ refers to the ensemble of the baseline network.

able to train NNs for each unique combination of values of the hyperparameters, we implement all changes that improved the predictions for two CNN- and one RNN-based TFE. In addition, we train the modified architectures on the larger data set defined in Row 5.5.d.

In this section, we compare the NN predictions to the SD standard reconstructions, when available, and the predictions of NNs based on the corresponding baseline architecture. For each observable and architecture, at least 5 NNs have been trained. We summarized the setups used to train the NNs used in this section in Table 7.1.

7.2.1 Selection process for NN models

Due to the non-determinism of the training on GPUs (see Sec. 7.3.2), NNs trained under the same conditions yield slightly different predictions for the same events. A priori, it is impossible to deduce the model performance from the early parts of the training process. Consequently, for each analysis using an NN-based approach, multiple models need to be trained. From this set of models, suitable ones can be selected by additional criteria.

In this work, we perform the selection process using a metric that appraises all predictions of each model as a whole. For this thesis, the high-energy regions of the CR spectrum are of particular interest. Hence, models need to be selected in this part of the phase space that is as independent of the primaries as possible and exhibit a high-enough resolution. In addition, the chosen models are used solely in regions of maximum trigger efficiency. Defining the thresholds 10^{19} eV and 60°, the metric

$$d(\Delta x) = \sigma_{\Delta x} + \lambda_m \Delta_{\text{p-Fe}}(\Delta x), \tag{7.1}$$

where λ_m is a scale parameter and d is only evaluated for events that fulfill the conditions $E_{\text{MC}} > 10^{19}$ eV and $\theta_{\text{MC}} < 60°$, coincides with these requirements. The choice of λ_m depends on investigated problem. In this work we set $\lambda_m = 1$. Note that, for the metric, it is implicitly assumed that the biases of events induced by helium and oxygen primaries lie between events induced by proton and iron primaries. Despite being reasonable due to the dependence of the signal distribution on the primary particle, it is not a priori clear that this assumption remains valid for all targets.

Table 7.1: Overview of setups used to build and train the NNs in this section. We list only deviations from the base setup defined in Appendix A.6.

	$\mathcal{AR}_{\text{i–iii}}$			Training		Other		
	type	n_f	d_f	N_b	λ_B	M_s	add. inputs	data set
a	CNN	10	0.2	64	1	5	-	Row 5.5.a
b	RNN	16	0.2	64	1	5	-	Row 5.5.a
c	CNN	10	0.2	64	0	5	-	Row 5.5.a
d	CNN	10	0.2	64	1	5	-	Row 5.5.d
e	CNN	10	0.2	512	1	5	θ_{SD};LG,t_{pf},$\frac{r}{1000}$	Row 5.5.d
f	CNN	10	0.2	2048	1	5	θ_{SD};LG,t_{pf},$\frac{r}{1000}$	Row 5.5.d
g	CNN	10	0.2	512	1	3	θ_{SD};LG,t_{pf},$\frac{r}{1000}$	Row 5.5.d
h	CNN	10	0.2	2048	1	3	θ_{SD};LG,t_{pf},$\frac{r}{1000}$	Row 5.5.d
i	RNN	16	0.2	32	1	5	θ_{SD};LG,t_{pf},$\frac{r}{1000}$	Row 5.5.d
j	RNN	16	0.2	32	1	3	θ_{SD};LG,t_{pf},$\frac{r}{1000}$	Row 5.5.d

We have not used Eq. (7.1) for training because of the restrictions on the phase space and since it does not account for helium and oxygen primaries. Still, this metric is a purely phenomenological choice that happened to work well for our approach. A better metric could be uncovered by comparing different choices for the metric for numerous networks trained on multiple different training data sets sharing the same test data set, which contains a sufficient number of events. However, this wide-ranging study lies outside of the scope of this thesis.

7.2.2 Prediction of the depth of the shower maximum

Since all of the prior studies on the architecture have been carried out for NNs trained on the depth of the shower maximum X_{max}, we ensure in this section that the changes we have implemented improve the quality of the predictions of our networks. Hence, before comparing the NNs based on the optimized architectures, we compare the predictions of a baseline network (see Row 7.1.b) with the predictions of a network using an optimized architecture (see Row 7.1.e). The set of NNs corresponding to Row 7.1.b has been chosen over the set of NNs corresponding to Row 7.1.a since the predictions of the RNN-based architecture showed better results than the CNN-based one. Both networks are selected from a set of 99 baseline networks and 10 optimized networks using Eq. (7.1) with $\lambda_m = 1$. The predictions of the optimized network excel the predictions of the baseline network (see Fig. 7.24). In each energy bin, the predictions of the modified network show a better precision for events induced by proton and iron primaries. The proton-iron bias is below the best-performing base network reaching consistent values below $5\,\text{g/cm}^2$ for all energy values above $10^{19.3}\,\text{eV}$. The overall improvement is mainly due to the increase in performance at high zenith angles (see Fig. 7.25). The proton-iron bias reaches a minimum value at 44° for both networks. At this zenith angle lies the median value of the zenith for our data set going up to 65°.

Since the reconstructed azimuth ϕ_{SD} is used to standardize the shower footprints (see Sec. 5.3.3), we have to check if there are any dependencies on the MC azimuth. By doing the standardization, we have implicitly assumed that the shower footprints are rotational invariant. Earth's magnetic field and the orientation of the PMT in the WCD (see Sec. 2.3.2) breaks this symmetry. Both networks are likely not sensitive enough to pick up any dependence (see Fig. 7.26) over the investigated phase space. The resolution $\sigma_{\Delta X_{\text{max}}}$ and the proton-iron

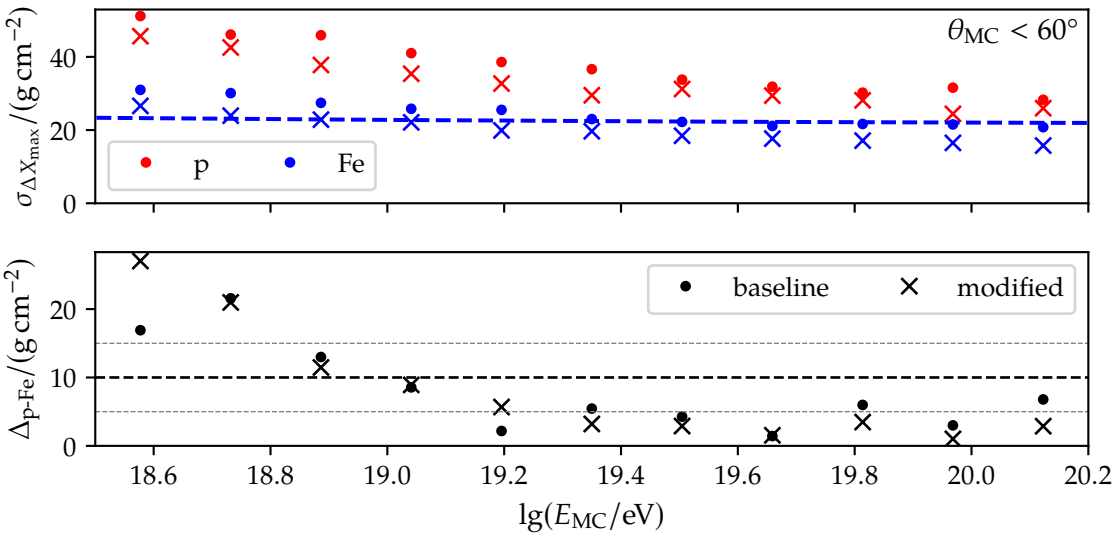


Figure 7.24: Comparison of the precision $\sigma_{\Delta X_{max}}$ (*top*) and of the proton-iron bias $\Delta_{p\text{-}Fe}(\Delta X_{max})$ (*bottom*) in bins of the logarithmic MC energy $\lg(E_{MC}/eV)$ between the predictions of the baseline network (dots) and the network (crosses) using the modified architecture described in Sec. 7.2. The dashed, blue line in the upper panel is the width of the X_{max} distribution of air showers induced by iron using the hadronic interaction model QGSJetII-04. We use only events with a MC zenith angle below 60°. The black lines in the bottom plot mark a proton-iron bias of 15 g/cm^2 (dotted), 10 g/cm^2 (dashed), and 5 g/cm^2 (dotted).

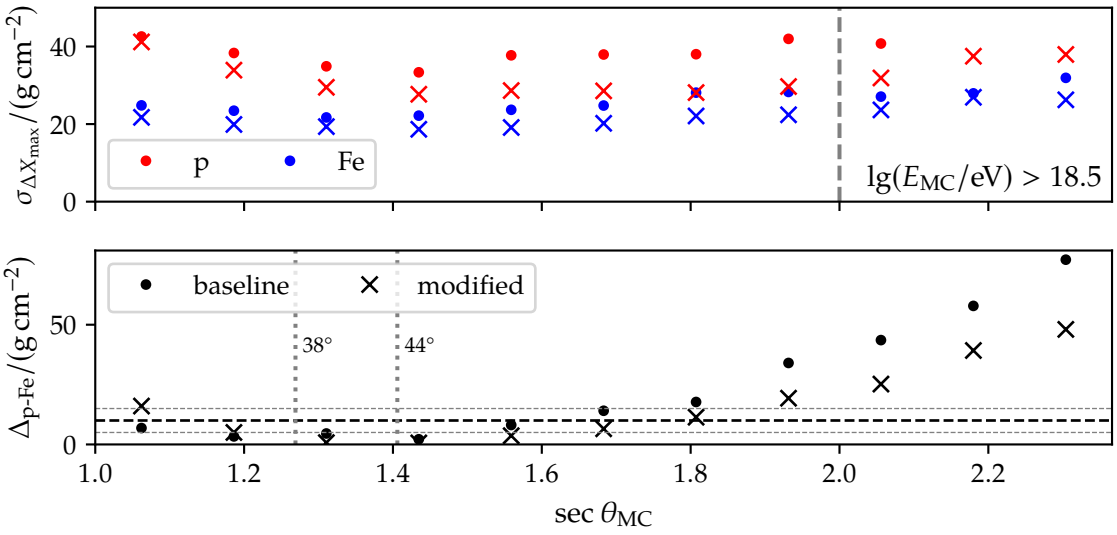


Figure 7.25: Comparison of the precision $\sigma_{\Delta X_{max}}$ (*top*) and of the proton-iron bias $\Delta_{p\text{-}Fe}(\Delta X_{max})$ (*bottom*) in bins of $\sec\theta_{MC}$ between the predictions of the baseline network (dots) and the predictions of the network (crosses) using the modified architecture. We use only events with a logarithmic MC energy of above 18.5. The vertical dashed line in the upper plot shows the position of $\theta_{MC} = 60°$. The black lines in the lower panel mark the proton-iron bias of 15 g/cm^2 (dotted), 10 g/cm^2 (dashed), and 5 g/cm^2 (dotted). The vertical lines in the lower panel show the position of $\theta_{MC} = 38°$ and $\theta_{MC} = 44°$. These angles are the median zenith values for a maximum zenith angle of 60° and 65°, respectively.

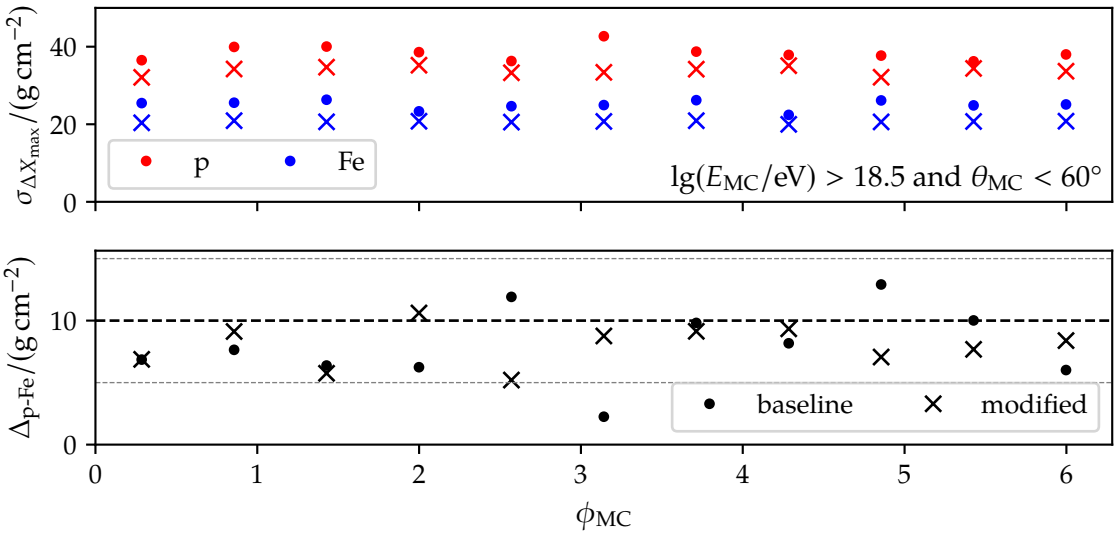


Figure 7.26: Comparison of the precision $\sigma_{\Delta X_{max}}$ (*top*) and of the proton-iron bias $\Delta_{p\text{-}Fe}(\Delta X_{max})$ (*bottom*) in bins of the MC azimuth angle ϕ_{MC} between the predictions of the baseline network (dots) and the predictions of the network (crosses) using the modified architecture. We use only events with a logarithmic MC energy of above 18.5 and a zenith angle of below 60°. The black lines in the bottom plot mark a proton-iron bias of 15 g/cm^2 (dotted), 10 g/cm^2 (dashed), and 5 g/cm^2 (dotted).

bias $\Delta_{\lg(E/eV)}$ are mostly flat and have no apparent dependence on the azimuth angle.

The effect of the asymmetry is more significant for higher zenith angles and higher signals. Cutting our test data set accordingly, the bias ΔX_{max} shows a slight azimuthal dependence appearing to be periodic for the modified network (see Fig. 7.27). The network predictions have a primarily negative bias up to 180° and a positive bias afterward. The absolute bias is below 2.5 g/cm^2 in both cases. Checking if this is not a statistical effect is outside of the scope of this thesis. Henceforth, we assume that our assumption of rotational invariance does not influence our predictions greatly.

Overall, these results indicate that the modifications used yield an improvement in the predictions in comparison to the baseline network. Both the precision, in terms of resolutions, and the accuracy, in terms of the proton-iron bias, of the predictions, are superior. This improvement is not only valid for the selected networks in the case but also for the entire sets of NNs (see Fig. D.10 and Fig. D.9).

During the training and selection of the modified setups tabulated in Rows 7.1.e to 7.1.j, a couple of differences to the results in Sec. 7.1 have been revealed. Despite the fact that the RNN-based architecture has shown promising results in Sec. 7.1.5.B for the smallest footprint size of $M_s = 3$, the NNs trained with the setup described in Row 7.1.j gave inferior results to those of the similar setup in Row 7.1.i. This is not the case for CNN-based architectures. In Sec. 7.1.4.A, increasing the batch size for the CNN-based architecture increased the overall precision and reduced the proton-iron bias for both investigated energy bins. However, the training of the setups described in Row 7.1.f and Row 7.1.h has been relatively unstable, also yielding inferior results to those from the setups in Row 7.1.e and Row 7.1.g. Using Eq. (7.1) with $\lambda_m = 1$, 6 NNs have been selected for each of the setups described in Rows 7.1.e to 7.1.j (see Table 7.2). The values of the chosen metric for most setups differ only about 1 g/cm^2 except for the NN in Row 7.2.f, which has a large value for the chosen metric. Comparing the values of $d(\Delta X_{max})$, only the predictions of the NNs in Row 7.2.c and Row 7.2.e have a value of below 27 g/cm^2. This reduction is mainly due to the low proton-iron bias of their predic-

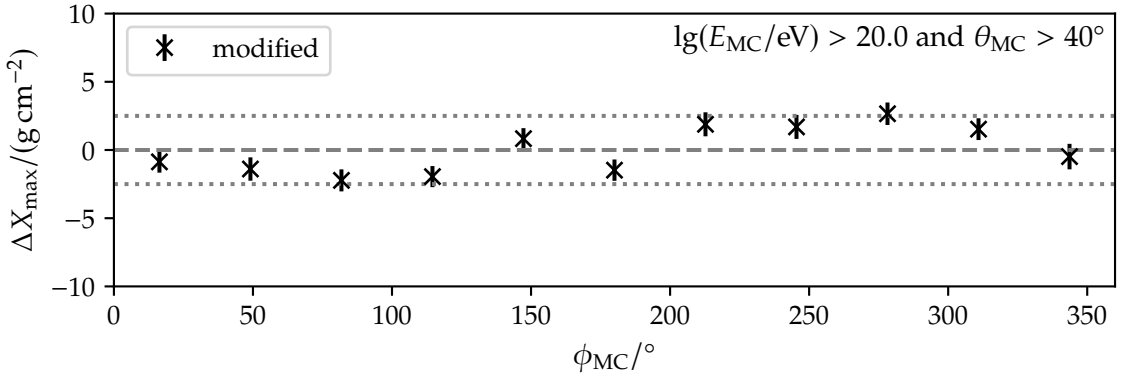


Figure 7.27: Bias ΔX_{max} in bins of the MC azimuth angle ϕ of predictions of the network (crosses). We use only events with a logarithmic MC energy of above 20.0 and a zenith angle of above 40°. The horizontal gray lines in the bottom plot mark a bias of 2.5 g/cm^2 (dotted), 0 g/cm^2 (dashed), and −2.5 g/cm^2 (dotted).

Table 7.2: Overview of the performance of the selected NN models on the metric defined in Eq. (7.1) and its sub-components. The last column is the number of NNs trained in the setup of the row.

	setup	$d(\Delta X_{\text{max}})/(\text{g/cm}^2)$	$\sigma_{\Delta X_{\text{max}}}$ / (g/cm^2)	$\Delta_{\text{p-Fe}}$ / (g/cm^2)	N
a	Row 7.1.e	27.24	23.72	3.52	10
b	Row 7.1.f	28.84	23.50	5.35	5
c	Row 7.1.g	26.77	24.33	2.44	9
d	Row 7.1.h	28.41	23.67	4.74	4
e	Row 7.1.i	26.71	24.06	2.66	10
f	Row 7.1.j	29.09	23.62	5.47	10
g	Row 7.1.d	27.43	25.18	2.26	20

tions lying below 3 g/cm^2. However, they also exhibit a resolution above 24 g/cm^2. Another candidate is the NN in Row 7.2.a, whose predictions have a resolution below the 24 g/cm^2 threshold and a proton-iron bias that is only slightly worse than the other networks. Since the standard error of the standard deviation is about 0.1 g/cm^2 and the standard error of the proton-iron bias is about 0.3 g/cm^2, we select Row 7.2.a as the model of choice. However, in Chapter 8, we perform additional cross-checks based on the NN in Row 7.2.c and Row 7.2.e.

One of the most significant improvements in resolution has been seen in Sec. 7.1.4.B by increasing the amount of training data. To disentangle the effects of adding more data from the architectural changes, NNs have been trained under the setup in Row 7.1.d. As expected, using a larger amount of training data, the predictions of NNs derived from the baseline architecture may achieve similar results to the modified architecture (see Row 7.2.g). However, comparing all NNs from Row 7.1.d the network described in Row 7.2.g is on the far left of the distribution of all d (see Fig. 7.28). Moreover, resolutions of under 24 g/cm^2 are not achieved. Therefore, we conclude that the architecture changes and the changes in the training procedure have increased the likelihood of obtaining a well-performing NN model.

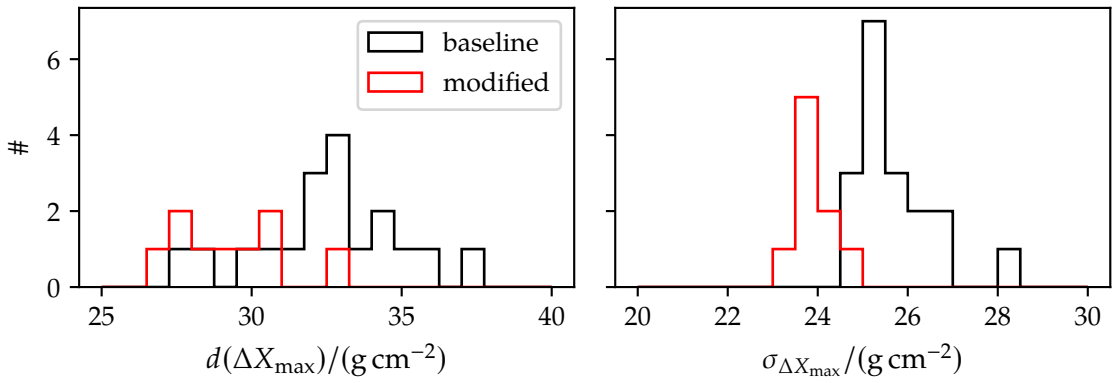


Figure 7.28: Distribution of $d(\Delta X_{\mathrm{max}})$ (*left*) defined in Eq. (7.1) and the precision $\sigma_{\Delta X_{\mathrm{max}}}$ computed above the energy 10^{19} eV for the NNs trained using the setup in Row 7.1.d (black) and NNs trained using the setup in Row 7.1.e (red).

7.2.3 Predictions of the zenith angle and the shower energy

To estimate mass-sensitive parameters from the signal distribution on the ground level, it is expected that a method needs to account for the zenith angle θ and the primary particle energy E since both observables have a considerable impact on the shower footprint. Hence, we check in this section if NNs using the architectures defined before yield sufficiently well predictions if trained on θ_{MC} and E_{MC}.

The zenith angle is an already well-reconstructed quantity since it is a geometrical shower property that is almost independent of the primary and hadronic model. Improving its precision does not yield a benefit for arrival direction studies due to the deflection during the propagating CRs in magnetic fields[2]. However, accurate knowledge is still very significant due to the attenuation dependence of the shower footprint. An improved prediction of the zenith angle could, in turn, improve indirectly the established analytic methods. Because θ_{SD} is used as an input for the optimized architectures and the SD reconstruction of the zenith angle is almost independent of the primary which induced the shower, we only use NNs based on the setup in Row 7.1.c. Because of the same reason, we selected the a NN using Eq. (7.1) with $\lambda_m = 0$. In total, 20 NNs have been trained. In addition, 5 NNs have been trained that use θ_{SD} as an additional input. In this way, it can be assessed if the architecture provides the possibility of refining the result of the SD reconstruction improving the overall predictions. The resolution of the predictions of the NN-based approaches is comparable to the resolution of the standard reconstruction (see Fig. 7.29). The decrease of resolution in the fourth energy bin is due to an 'outlier' in the SD reconstruction. The resolution of predictions of the NN using θ_{SD} as additional input improves slightly. It also makes the predictions subject to the reconstruction error of the standard reconstruction shown by the decrease in resolution in the second and fourth bin. Even though the resolution of the predictions from the NNs is slightly worse, the composition dependence of the NNs is overall larger (see Fig. 7.30) than the standard reconstruction. It is likely that this is due to the trade-off in the MSE used for the training process. Nevertheless, the predictions of the NN and the SD reconstruction agree with each other. Therefore, the chosen architecture is suitable for inferring the zenith angle of the shower, making it possible to use it for other predictions internally. Since the result of the refiner is similar to the baseline network, we do not compare the results to one of the advanced architectures.

Both the NN based on the Row 7.1.a as well as the NN based on Row 7.1.e show a su-

[2]The magnetic fields give us a limit at which resolution does not improve the anisotropy studies anymore.

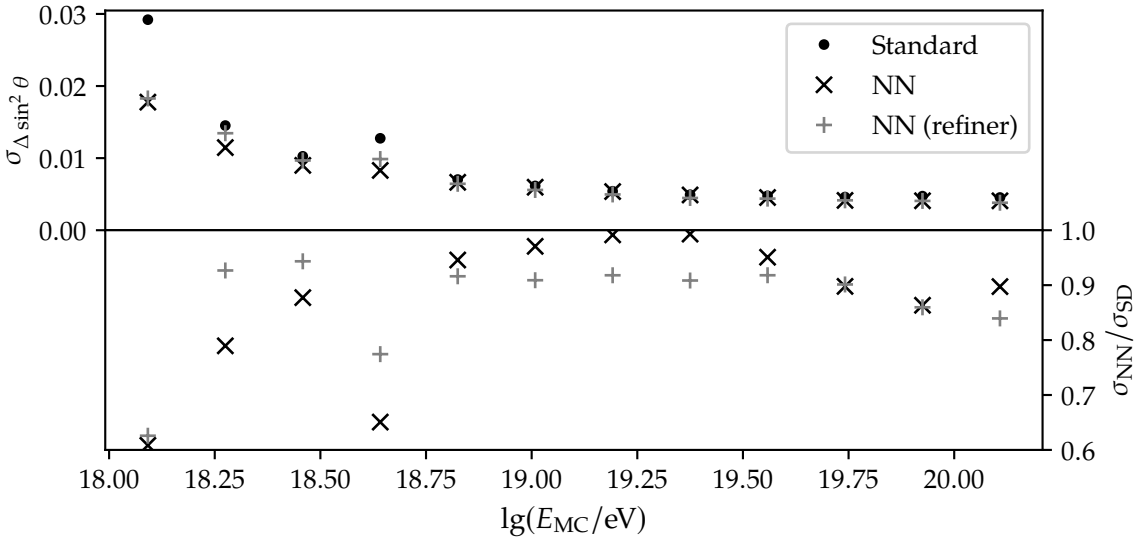


Figure 7.29: Precision of $\sigma_{\Delta \sin^2 \theta}$ of the predictions from two NNs models and from the standard reconstruction algorithm found in Offline (points, gray) in bins of the logarithmic MC energy. We have used two different kinds of networks based on the setup described in Row 7.1.c. The second network (refiner) uses the reconstructed θ_{sd} as an additional input. The third bin is an 'outlier' in the SD reconstruction.

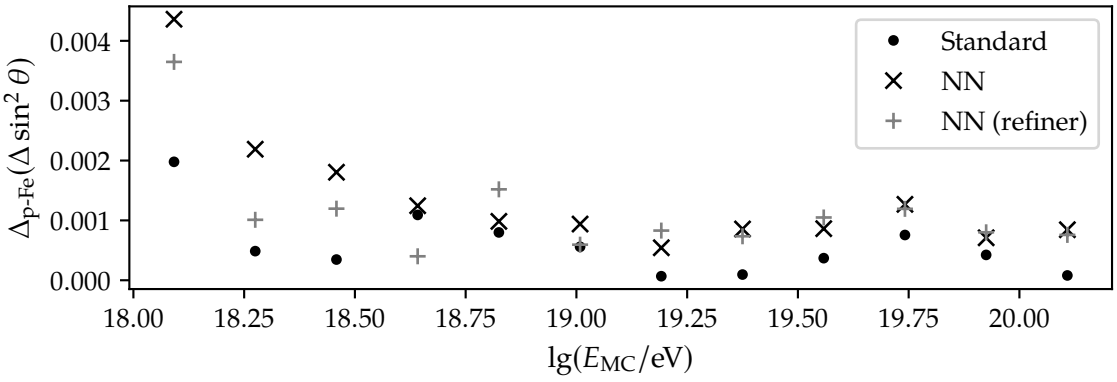


Figure 7.30: Proton-iron bias $\Delta_{p\text{-}Fe}(\Delta \sin^2 \theta)$ of the predictions from two NNs models and from the standard reconstruction algorithm found in Offline (points, gray) in bins of logarithmic MC energy. Two different kind of networks have been used based on the setup described in Row 7.1.c. The second network (refiner) uses the reconstructed θ_{sd} as an additional input.

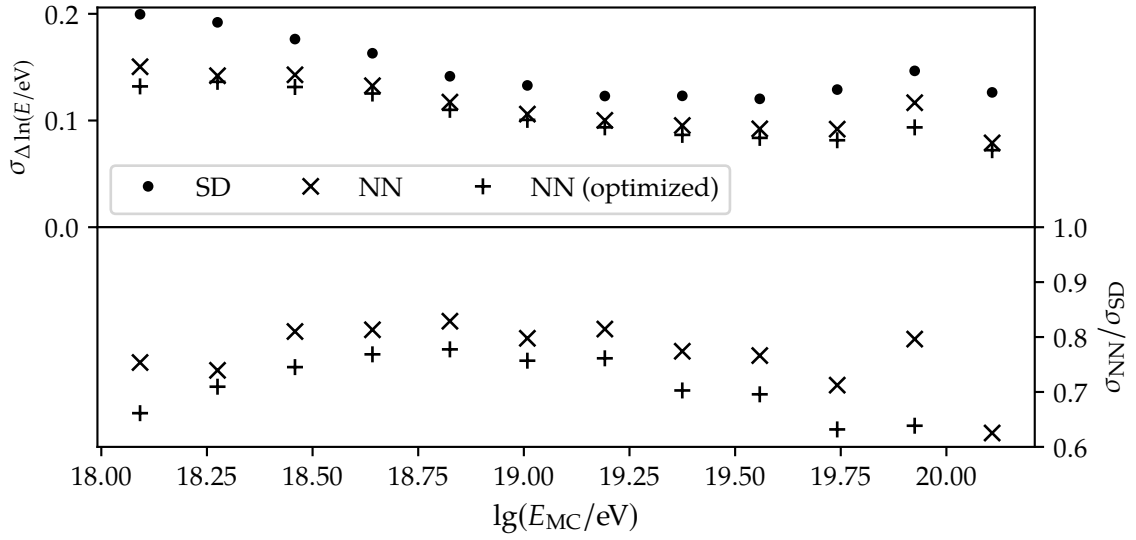


Figure 7.31: Precision of $\sigma_{\Delta \ln(E/\mathrm{eV})}$ of the predictions from two NNs models and from the standard reconstruction algorithm found in $\overline{\mathrm{Off}}\underline{\mathrm{line}}$ (points, gray) in bins of the logarithmic MC energy. Two different kind of networks have been used based on the setup described in Row 7.1.b (crosses) and Row 7.1.e (pluses), respectively.

perior resolution in all energy bins if compared to the SD reconstruction of the energy (see Fig. 7.31). Moreover, in this case, there is no direct trade-off between bias and resolution. The proton-iron bias in each energy bin is also significantly reduced for the NN-based networks compared to the standard reconstruction (see Fig. 7.32). Hence, NNs based on the used architectures also give an energy estimate for given shower footprints. An in-depth study of the NN-based prediction of the primary particle energy is done in [A:29].

Note that, even though the SD energy estimator is calibrated on measurements and therefore biased on simulations (see Fig. D.8), the resolution σ and proton-iron bias $\Delta_{\text{p-Fe}}$ are independent of global shifts. Hence, it is not expected that the result of this study would differ significantly, when we would use a model optimized for simulations (see Appendix B.6).

7.2.4 Prediction of relative muon content

There is no standard reconstruction to obtain the relative muon content R_μ at ground level. Moreover, since the definition used to compute R_μ is based on the muon number on ground level taken from the CORSIKA air shower simulation files, it is not directly comparable to the R_μ used in the latest Universality framework (see Sec. 3.3.2) in which is it based on the ground signal as measured by the WCD stations. Therefore, the predictions of the NN-based models are only compared with each other.

Like in Sec. 7.2.2, at least 5 NNs have been trained for the setups defined in Rows 7.1.e to 7.1.j. Due to the unsatisfactory performance of the CNN-based setups using $N_b = 2048$ in Sec. 7.2.2, we removed the models using this batch size from the analysis (see Table 7.3). Again, we select the NN using the setup in Row 7.1.e (see Row 7.3.a) for the comparison. Again, the predictions of the selected NN are more precise and have a lower proton-iron bias than a baseline network (see Fig. 7.33). Moreover, comparing the predicted distributions of proton and iron events at high energies, the distributions are more separated, reaching approximately 0.2 above an energy of 10^{19} eV. The improvement of the resolution stems from events with low zenith angles (see Fig. D.11). However, for very inclined showers, the NN using the modified setup exhibits a worse resolution. Still, the proton-iron bias of the

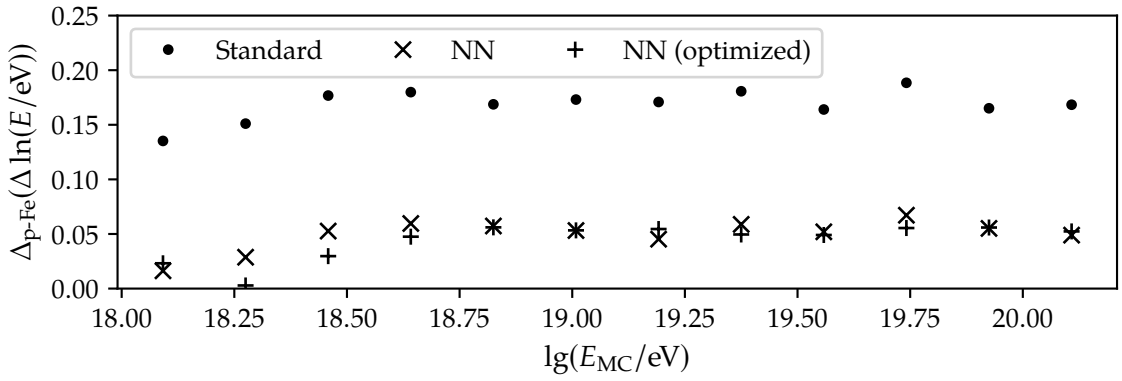


Figure 7.32: Proton-iron bias $\Delta_{\text{p-Fe}}(\Delta \ln(E/\text{eV}))$ of the predictions from two NNs models and from the standard reconstruction algorithm found in Offline (points, gray) in bins of the logarithmic MC energy. Two different kind of networks have been trained using the setup described in Row 7.1.b (crosses) and Row 7.1.e (pluses), respectively.

Table 7.3: Overview of the performance of the selected NN models trained to predict the relative muon content R_μ on the metric defined in Eq. (7.1) and its sub-components. The last column is the number of NNs trained in the setup of the row. Due to the instability of the networks using a batch size of 2048 we have excluded them from this analysis.

	setup	$d(\Delta R_\mu)$	$\sigma_{\Delta R_\mu}$	$\Delta_{\text{p-Fe}}(\Delta R_\mu)$	$\Delta_{\text{p-Fe}}(R_\mu)$	N
a	Row 7.1.e	0.178	0.108	0.070	0.201	5
b	Row 7.1.g	0.179	0.108	0.071	0.200	5
c	Row 7.1.i	0.226	0.111	0.115	0.156	5
d	Row 7.1.j	0.191	0.105	0.086	0.185	5

prediction of the modified setup in the same region is much better.

Analyzing the difference between the average predicted value for air shower events induced by proton and iron primaries (see Fig. 7.34) reveals that both networks do not separate the proton and iron events very well. The differences of the predictions of both NNs lie consistently below the expected difference computed from air shower simulations using the hadronic interaction model QGSJetII-04.

The distribution of predictions of the NNs trained on the baseline setup (see Row 7.1.a) exhibits a shift towards the average expected value of R_μ (see Fig. 7.35). This is an effect of the low amount of training data and the intermediate R_μ values from the helium and oxygen events. Using a data set with the same amount of events but only proton and iron primaries, the R_μ distributions of proton and iron events shift noticeably towards the expected R_μ values. We do not exploit this behavior in this work, preferring to employ a NN that is well-performing for all primaries. Still, it is an intriguing idea that is worth to be investigated in future works.

7.2.5 Direct predictions of the logarithmic mass number

Since there is no standard method of reconstructing $\ln A$, all comparisons are made between NNs of the baseline setup of Row 7.1.b and NNs of the setup Row 7.1.g. The setup Row 7.1.g has been chosen because of the performance in Table 7.4. From both sets of NNs, again, the two NN have been selected which have minimal $d(\Delta \ln A)$ (see Eq. (7.1)). The resolution of

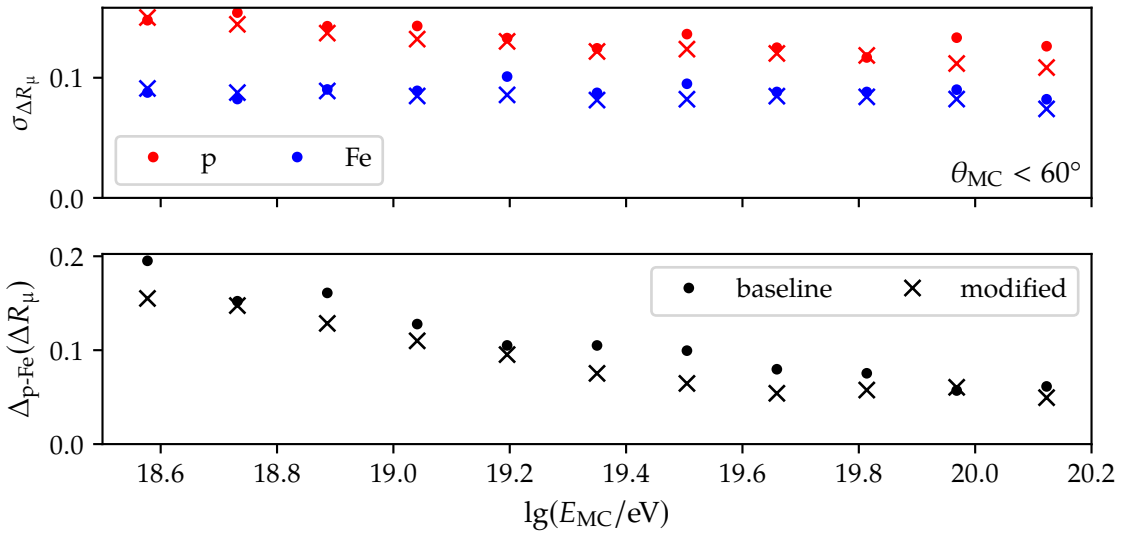


Figure 7.33: Precision $\sigma_{\Delta R_\mu}$ for proton (red) and iron (blue) events (*top*) and proton-iron bias $\Delta_{p\text{-}Fe}(\Delta R_\mu)$ (*bottom*) in bins of the logarithmic MC energy $\lg(E_{MC}/eV)$. We compare the predictions of the baseline network (dots) and the network (crosses) using the modified architecture described in Sec. 7.2. We have included only events with a MC zenith angle below 60°.

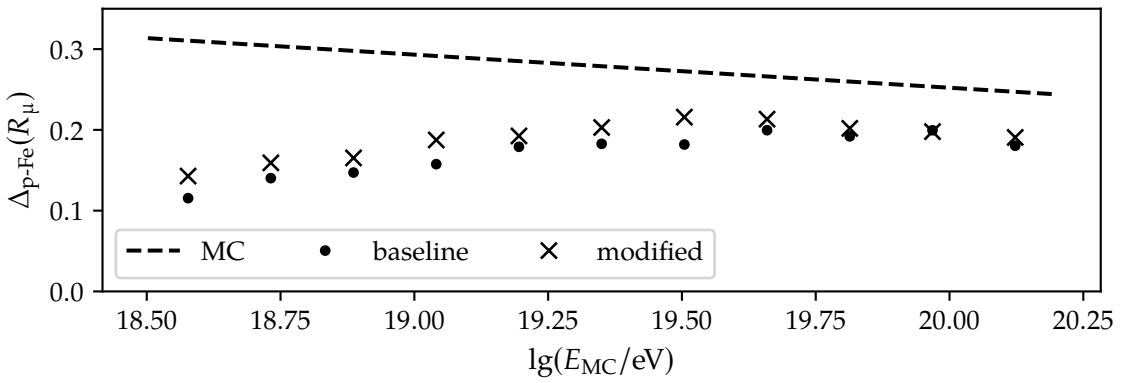


Figure 7.34: Difference of average prediction of proton and iron events $\Delta_{p\text{-}Fe}(R_\mu)$ for the predictions of the baseline network (dots) and the network (crosses) using the modified architecture in bins of the logarithmic MC energy. The dashed line is the expected average difference between the iron and proton R_μ for the hadronic interaction model QGSJetII-04.

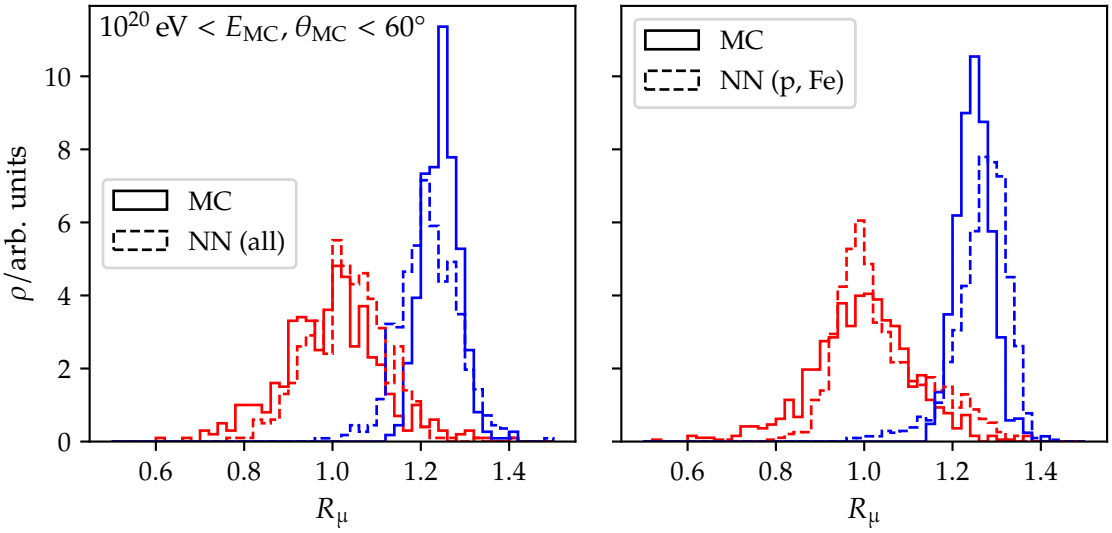


Figure 7.35: Predictions of R_μ from a NN trained with the setup described in Row 7.1.a (*left*) and a NN with the setup described in Row 7.1.d (*right*). Due to the training on all primaries the NN predictions in the *left* panel shows a shift of both distributions towards the mean. In contrast to this, the predictions of the NN trained only on proton and iron events has much less overlap in the center regions.

Table 7.4: Overview of the performance of selected NN models trained to predict the logarithmic mass number $\ln A$ on the metric defined in Eq. (7.1). We added the summands of Eq. (7.1) and the average difference between the predictions between proton and iron events. The last column is the number of NNs trained in the setup of the row. Due to instabilities, we have excluded NNs based on the RNN TFE.

	setup	$d(\Delta R_\mu)$	$\sigma_{\Delta R_\mu}$	$\Delta_{\text{p-Fe}}(\Delta R_\mu)$	$\Delta_{\text{p-Fe}}(R_\mu)$	N
a	Row 7.1.e	2.288	1.149	1.139	2.886	5
b	Row 7.1.f	2.236	1.128	1.108	2.917	5
c	Row 7.1.g	2.218	1.141	1.076	2.949	5
d	Row 7.1.h	2.284	1.129	1.155	2.870	5

the predictions Fig. 7.36 is for both of the NN comparable (see Fig. 7.36). For each energy bin, the predictions of the NN using the optimized setup have a slightly worse precision for proton events and better resolutions for iron events. However, the proton-iron bias of the NN using the optimized setup is lower. The resolution and proton-iron bias in bins of the zenith (see Fig. D.13) and the azimuth (see Fig. D.14) show similar behavior as in Sec. 7.2.4.

Even though the predictions of the NN trained on the modified setup show less proton-iron bias, the predictions are still shifted towards the average $\ln A$ value of the underlying distribution (see Fig. 7.37). The difference between the average predictions for proton and iron events is below the 4. For energy values above $10^{19.5}$ eV, the NN using the modified setup barely reaches a difference of 3. This behavior is most likely caused by regression towards the mean since the helium and oxygen events have values close to $\ln A = 2$.

Using a data set only comprised of proton and iron events improves the separation of the predicted values of $\ln A$ for proton and iron events (see Fig. 7.38). Still, as in Sec. 7.2.4, since no helium and oxygen events have been used in training, it is a priori not clear if the NNs trained on the proton and iron subset extrapolate between proton and iron events correctly.

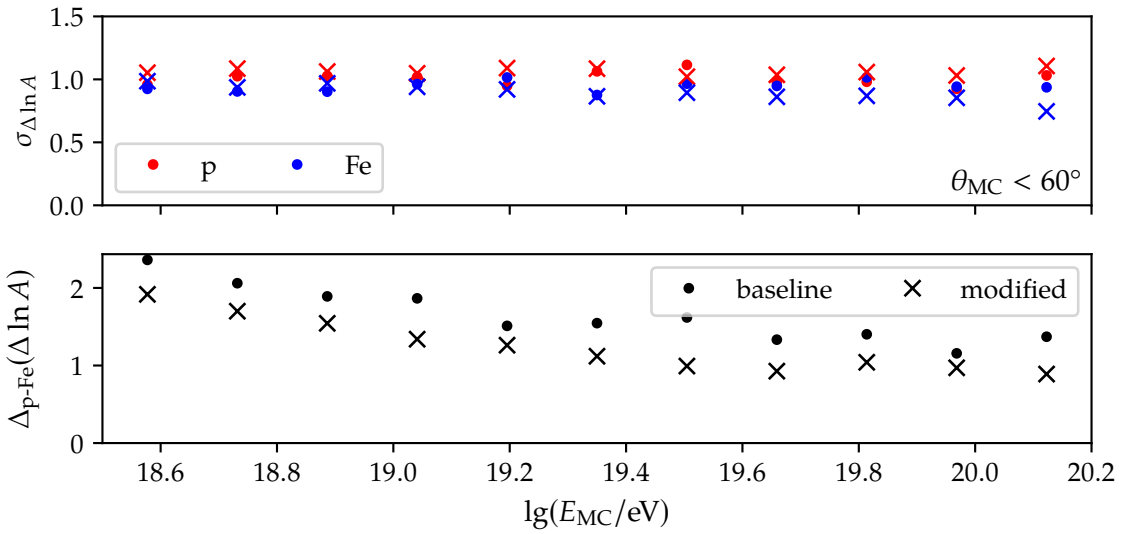


Figure 7.36: Precision $\sigma_{\Delta \ln A}$ for proton (red) and iron (blue) events (*top*) and proton-iron bias $\Delta_{p\text{-}Fe}(\ln A)$ (*bottom*) in bins of the logarithmic MC energy $\lg(E_{MC}/eV)$. We compare the predictions of the baseline network (dots) and the network (crosses) using the modified architecture described in Sec. 7.2. We have included only events with a MC zenith angle below 60°.

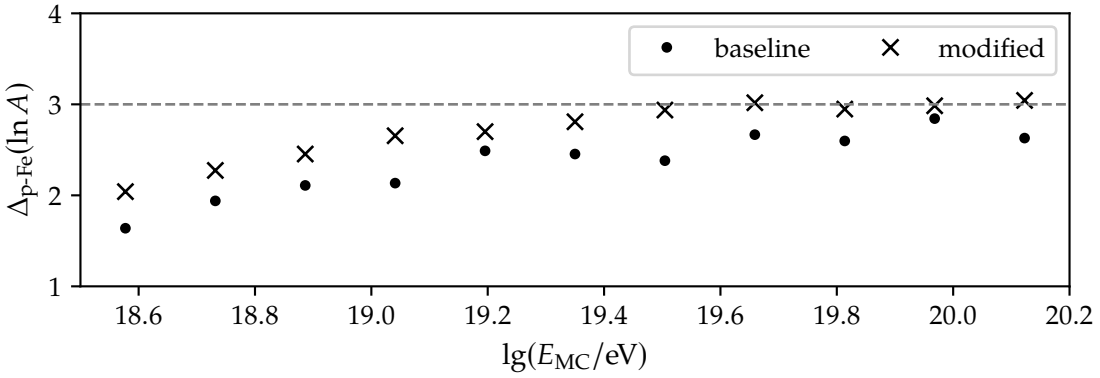


Figure 7.37: Difference of average $\ln A$ predictions for proton and iron events in bins of the logarithmic MC energy for the baseline NN (points) and the NN (crosses) trained on the modified setup. The dashed, gray line marks an average difference of 3.

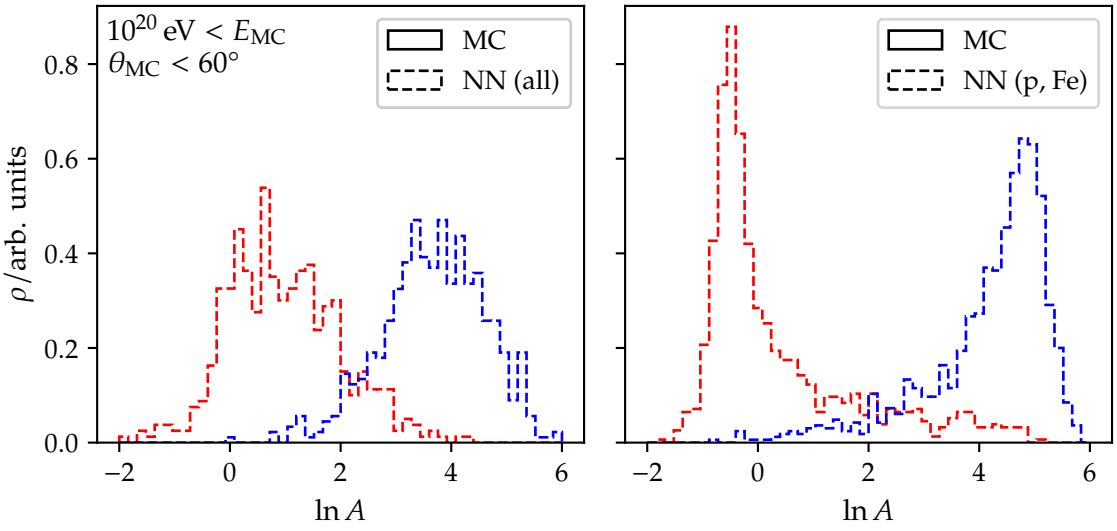


Figure 7.38: Predictions of $\ln A$ from a NN trained with the setup described in Row 7.1.a (*left*) and a NN with the setup described in Row 7.1.d (*right*). Due to the training on all primaries the NN predictions in the *left* panel shows a shift of both distributions towards the mean. In contrast to this, the predictions of the NN trained only on proton and iron events has much less overlap in the center regions.

Again, we leave the study based on this matter for future analysis since a careful exploration of the possibilities of modifying the distribution of input events is beyond the scope of this work.

The maximum shower depth X_{max} and relative muon content R_μ are phenomenologically related to the primary particle mass (see Sec. 2.2.3.B). Hence, instead of a direct prediction of $\ln A$, it is also possible to use X_{max} and R_μ directly estimate the most likely value of $\ln A$. Using Eq. (2.12) and Eq. (2.13) and assuming that X_0 is independent from the primary particle mass, the logarithmic mass number can be independently expressed as

$$\ln A[X_{max}] = \ln 56 \frac{X_{max} - \langle X_{max} \rangle_p}{\langle X_{max} \rangle_{Fe} - \langle X_{max} \rangle_p} \tag{7.2}$$

and

$$\ln A[R_\mu] = \ln 56 \frac{\ln R_\mu - \langle \ln R_\mu \rangle_p}{\langle R_\mu \rangle_{Fe} - \langle \ln R_\mu \rangle_p}. \tag{7.3}$$

To find an optimal linear combination of the estimates in Eq. (7.2) and Eq. (7.3), we use a linear fit and scale the fit result by a factor to ensure that the mean prediction for iron events is $\ln 56$. This procedure yields a robust predictor (see Fig. 7.39). Even when cutting in the phase space, the distributions do not shift significantly.

Nevertheless, the scaling reduces the precision of the method (see Fig. 7.40), making it much worse when compared to the results in Fig. 7.38. However, it also ensures that the average prediction $\ln A$ on the data set is correctly estimated (see Fig. 7.41). The estimation of $\ln A$ with R_μ and X_{max} serves as a crosscheck for the direct prediction of $\ln A$. In future studies, this ansatz can be improved by using a parametrization that depends on other shower observables. For example, in Fig. 7.39, there is a slight dependence on the zenith cut shifting the distributions that could be accounted for in this way.

To conclude this section, we estimate the upper limit on the separation between proton and iron events for the direct and indirect predictions of $\ln A$ as a function of logarithmic

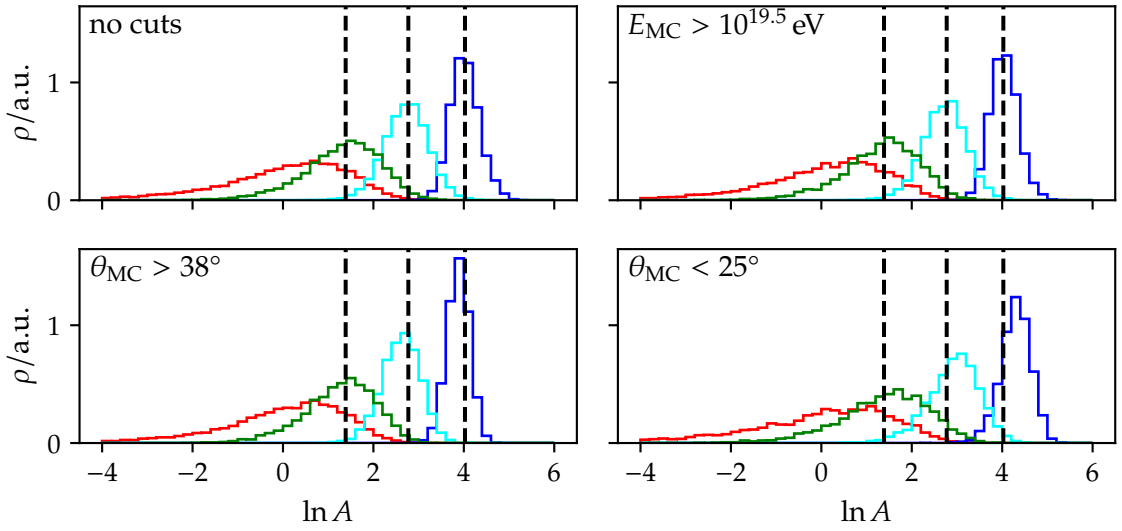


Figure 7.39: Distributions of the logarithmic mass number $\ln A$ predicted by a linear model based on X_{max} and R_μ split into the different primaries in the CORSIKA data set. The text in the upper right corner in the panels shows the used cut.

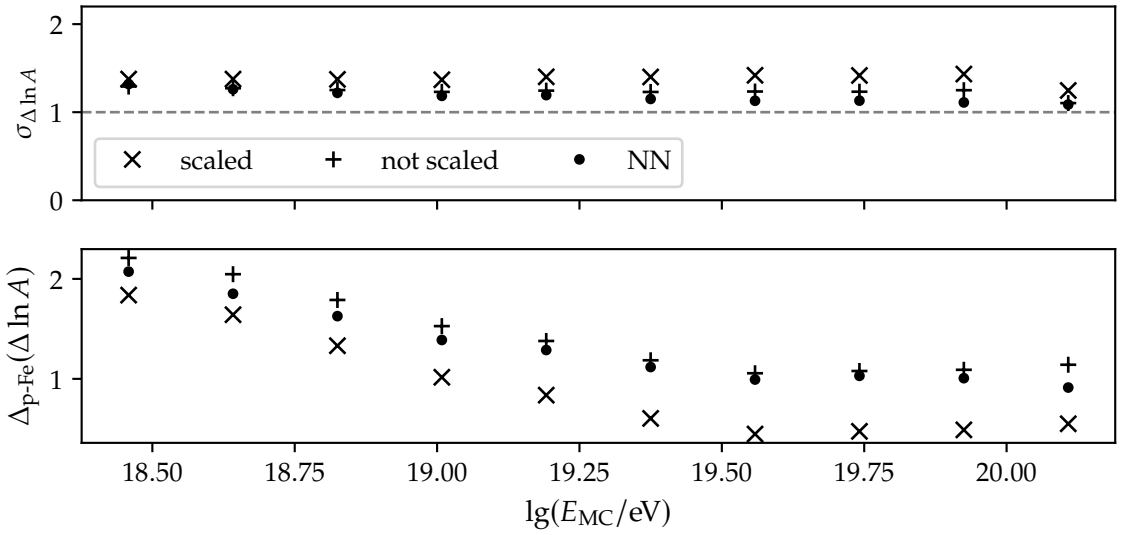


Figure 7.40: Precision $\sigma_{\Delta \ln A}$ (*top*) and proton-iron bias $\Delta_{p\text{-}Fe}(\ln A)$ (*bottom*) in bins of the logarithmic MC energy $\lg(E_{MC}/eV)$. We compare the predictions of the indirect prediction of $\ln A$ using the X_{max} R_μ predictions for the scaled (crosses) and not-scaled model (pluses) with the direct predictions of $\ln A$ (points) using the NN model defined in Sec. 7.2.5.

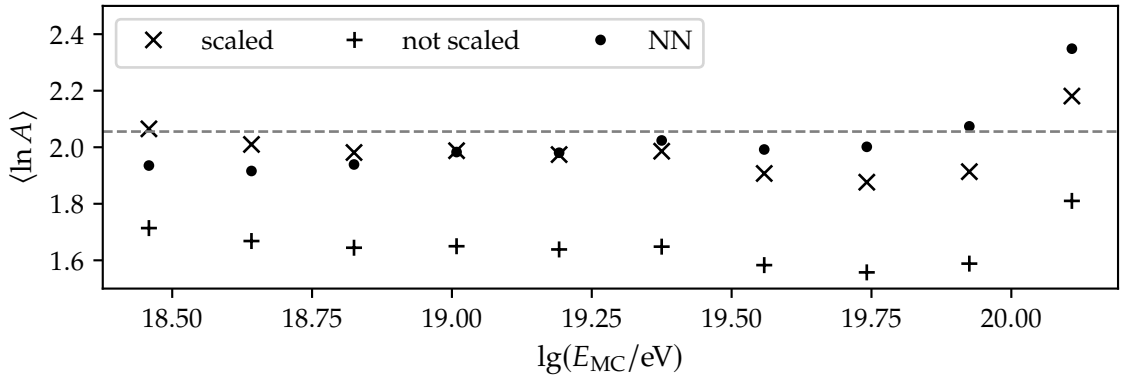


Figure 7.41: Average prediction of $\ln A$ for both indirect methods based on $X_{\max}$ R_{μ} predictions (crosses and plusses) and the direct method (points) using the NN model defined in Sec. 7.2.5. The dashed, gray line is the expected $\ln A$ value of the underlying data set.

energy. In measurements we do not have access to a primary-independent energy estimator like E_{MC}. Hence, we use the SD energy estimate E_{SD} in this estimation. We remove the global shift between $\lg E_{\mathrm{MC}}$ and $\lg E_{\mathrm{SD}}$ (see Fig. D.8) by subtracting the average difference of both energy values to E_{SD}:

$$\lg E_{\mathrm{SD}^*} = \lg E_{\mathrm{SD}} - \langle \lg E_{\mathrm{SD}} - \lg E_{\mathrm{MC}} \rangle. \tag{7.4}$$

The separation between proton and iron events of the direct predictions is superior to that of the indirect predictions (see Fig. 7.42). Over an energy of $10^{19.3}$ eV the direct estimation reaches consistently merit factors above 2.

7.3 Estimating uncertainty of neural network predictions

In this section, we focus on developing methods to estimate the uncertainty of NN-based predictions using the setup established in Sec. 7.1 and verified in Sec. 7.2. Again, we use the depth of the shower maximum $X_{\max}$ as the target all tests are performed due to the same reasons highlighted in Sec. 7.1.

We have split this uncertainty estimation into three parts. In the first two parts, we illustrate two useful approaches to find the uncertainty of a single network prediction. We have summarized the idea of both approaches in Fig. 7.43. Essentially, we analyze the predictions of one network predicting on multiple throws of the same CORSIKA shower and the predictions of multiple networks trained under the same conditions on the same shower. In the second part, we estimate the uncertainties arising from the high-energy hadronic interactions and the unknown composition.

7.3.1 Study of predictions of NN on the same CORSIKA shower

The positioning of the SD grid relative to the impact point of the shower core affects the signal distribution measured in the detector stations triggered by a shower. Since the event-level NNs described in the previous sections use this encoded signal information of the shower footprint, the prediction of the NNs depends on this relative positioning giving rise to uncertainty. We interpret this uncertainty as the combination of using the detector stations of the SD array and the NN-based method itself.

To capture this effect, we exploit the multiplicity of the *Karlsruhe* simulation library (see Sec. 5.1.2) based on the hadronic interaction model QGSJetII-04 (see Row 5.6.a). Since each

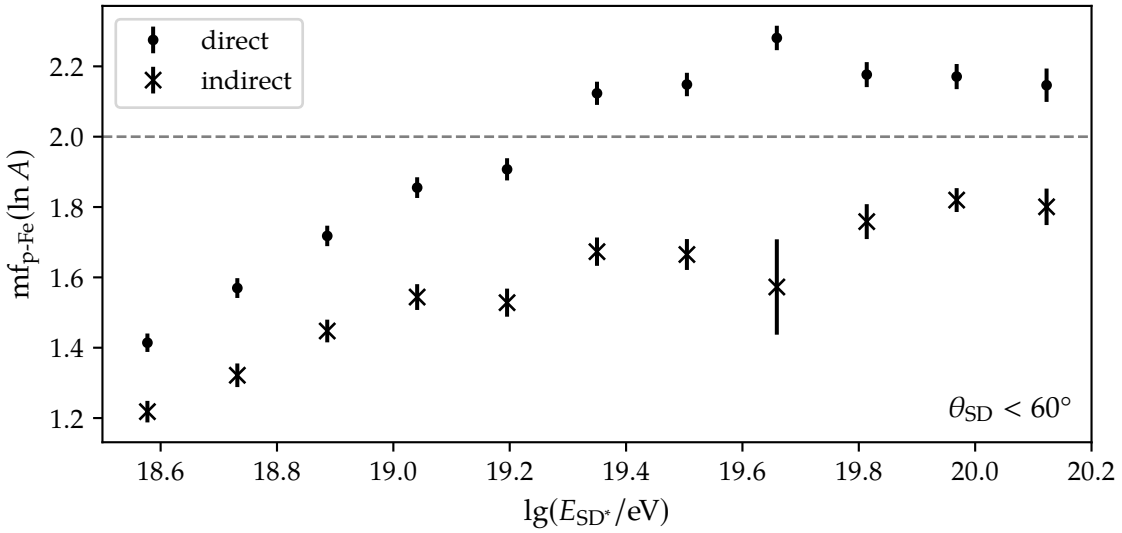


Figure 7.42: Merit factor of direct (points) and indirect (crosses) ln A predictions of proton and iron events binned in lg E_{SD^*} for events with a reconstructed zenith angle below 60°. The dashed, gray line marks a merit factor of 2.

CORSIKA shower is reused in this simulation library up to 10 times, we obtain for each of these showers up to 10 different measurements of the same shower footprint. Encoding these footprints and using the NN to predict the X_{max} grants us up to 10 different predictions. Hence, since the *Karlsruhe* library is a fixed library, we gather for each combination of primary, shower angle, and shower energy 120 sets of predictions. Using Eq. (4.45), we compute one value of ς for each set (see Fig. 7.43). Since in each of the sets, the same shower is predicted, shower-to-shower fluctuations do not affect the value of ς. We use the NN defined in Row 7.2.a for this study.

The 120 unique showers use 12 different atmospheric conditions corresponding to the conditions of the 12 months in a year. Since the atmospheric conditions also impact the signal distribution at ground level, the ς is not a pure estimate of the uncertainty from the SD detector and method itself. We assume that the contribution due to the different atmospheric conditions is small for ς because shifts in the prediction due to increased and decreased average signals do not affect its absolute value.

To obtain an estimate for the expected value of ς for each combination of primary, energy, and zenith angle, we use the variance-weighted mean. Due to the low statistic in each of the 120 sets, the single ς values are driven by outliers. We use Eq. (A.5) to compute the variance for each ς. Excluding the zenith angle of $\theta = 65°$, the value of ς depends slightly on the used zenith angle (Fig. 7.44). The dependence is more prevalent in proton events than in iron events. For different energy bins (see Figs. D.15 to D.17), the behavior is similar. The deviation from the average values becomes smaller for smaller energy values.

To simplify the analysis and obtain a rough estimate of the uncertainty as a function of the energy, we assume that most of the deviations from the average value in the bins below 60° are caused by the low statistics. This course of action is acceptable because around 38°, the dependence is almost flat. Henceforth, we neglect the zenith dependence in favor of the energy dependence.

Using only the zenith bins below 60°, we compute the expected value of ς again via the variance-weighted average (see Fig. 7.45). The ς for proton and iron events decreases with energy from 25 g/cm^2 to 17 g/cm^2 and 18 g/cm^2 to 12 g/cm^2, respectively.

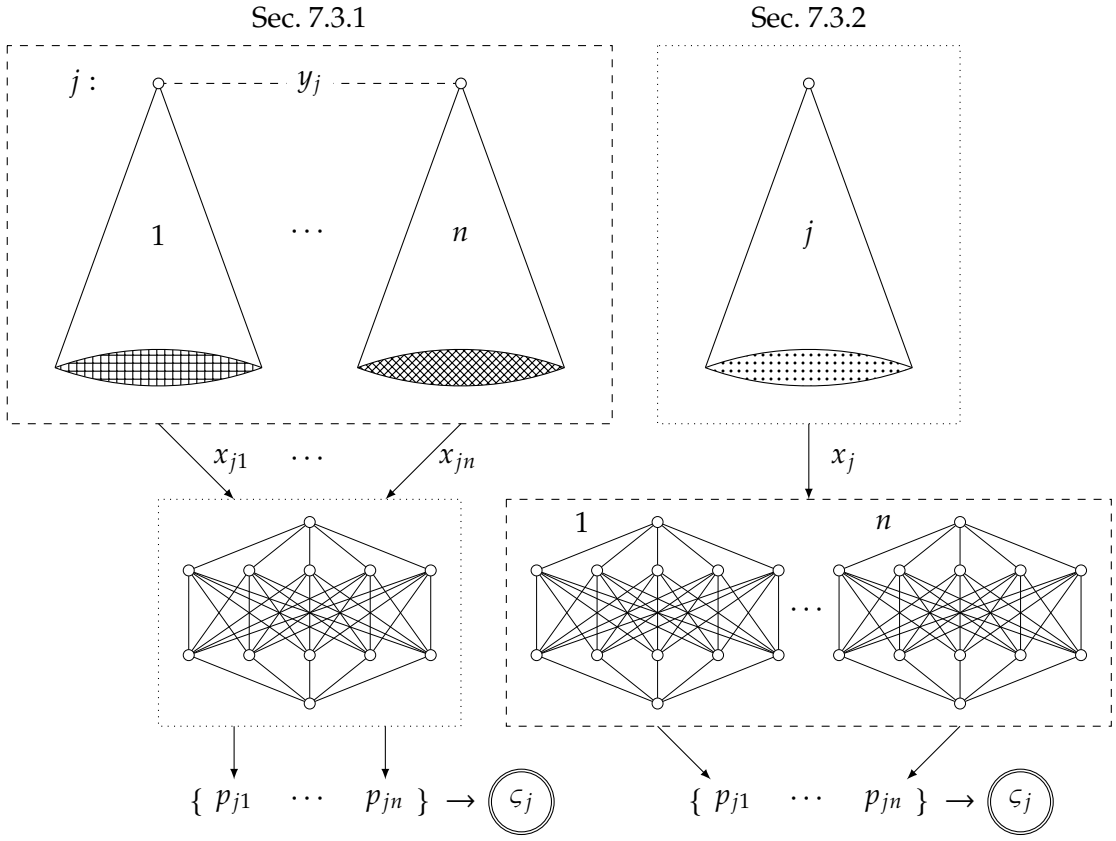


Figure 7.43: Illustration of the procedures followed in Sec. 7.3.1 (*left*) and Sec. 7.3.2 (*right*) to estimate the uncertainty of predictions based on NNs. In Sec. 7.3.1, we investigate how the prediction of one single NN changes when applied on the same CORSIKA shower at different positions of the SD grid. We simulate the shower j up to n times, giving us n network inputs. From these inputs x_{ij}, we obtain n predictions p_{ij}. We use these x_{ij} as input for the network, obtaining a set of n predictions p_{ji} for the target y_j. For the set of predictions, we compute ς_j (see Eq. (4.45)). In Sec. 7.3.2, we analyze how the training process and the minima in the solution space influence the predictions. We train n models under the same pre-defined conditions. Afterward, we obtain n predictions p_{ji} for each shower j. Again, we compute a ς_j for each of the showers.

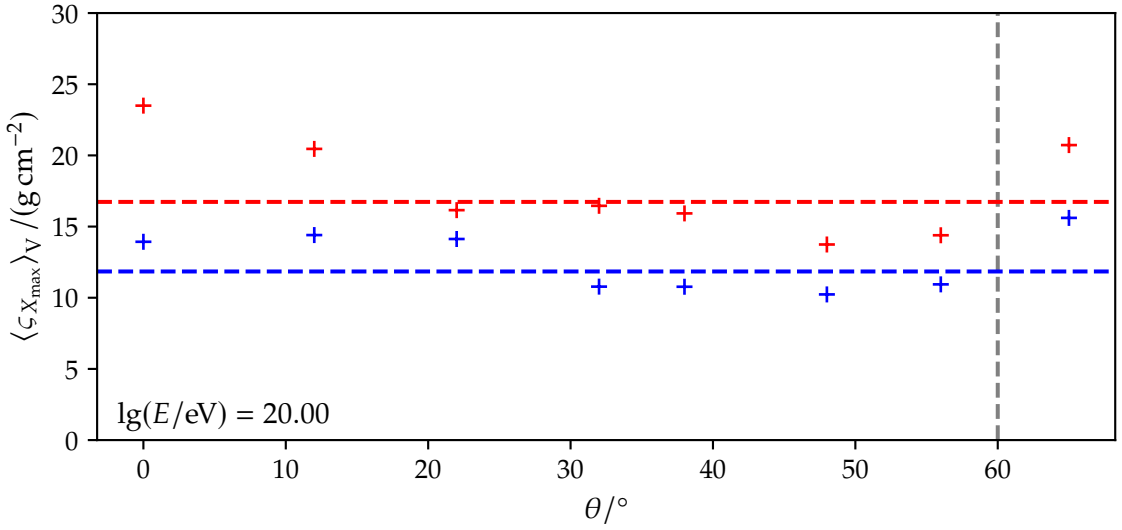


Figure 7.44: Variance-weighted average values of $\varsigma(\Delta X_{\max})$ for proton (points, red) and iron (points, blue) events for each zenith bin in the $\lg(E/\mathrm{eV}) = 20$ bin of the *Karlsruhe* library. The horizontal dashed lines show the average value of ς for the proton (red) and iron (blue) events without accounting for the last zenith bin. The vertical dashed line marks the 100% efficiency boundary of the SD detector.

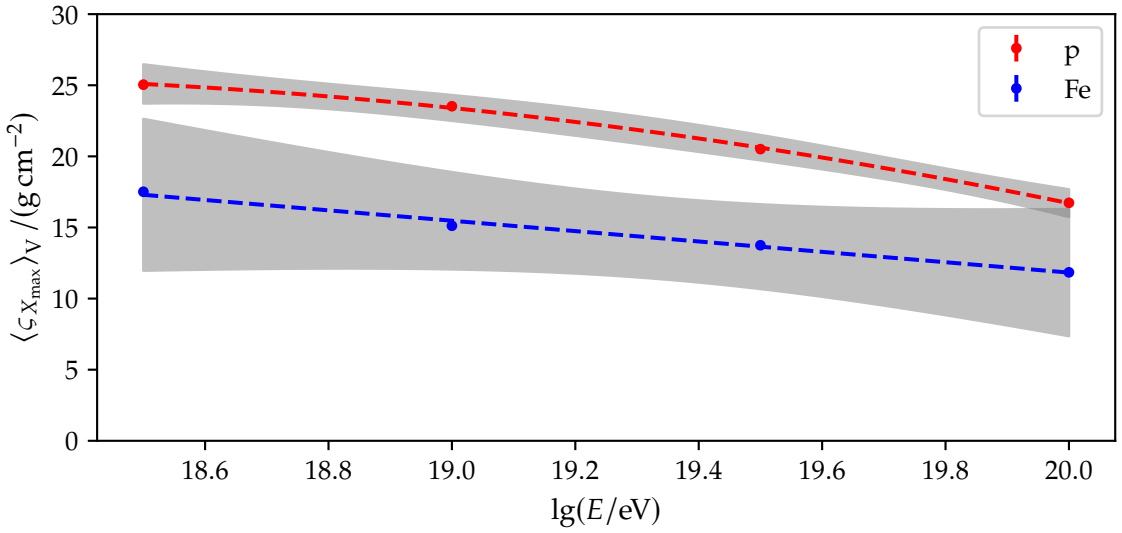


Figure 7.45: Variance weighted average values of $\varsigma(\Delta X_{\max})$ over all zenith bins for proton (points, red) and iron (points, blue) events for all energy bins of the *Karlsruhe* library. The color-coded dashed lines depict a quadratic and linear interpolation for the ς values of proton and iron events, respectively. The gray areas show the 95% confidence bands of the fit functions.

7.3.2 Uncertainty due to the training process

In Secs. 7.1 to 7.2, we trained multiple NNs for each model to ensure that a proper comparison between different models was possible. Due to the non-determinism of the NN training process, NNs trained under the same conditions yield different predictions due to small changes in the weight update process. This is caused by the parallelization of the computations. Since the order of computations is not strictly defined in during training and calculations are not exact, this introduces an irreducible noise during the training process. Nowadays, TF has an explicit – but experimental – option to enable determinism [T:O]. However, this is primarily for debugging purposes, slowing down the training.

Hence, training can be understood as a sampling process in which NN models are drawn from an unknown distribution. We want to probe this distribution using the procedure depicted in Fig. 7.43. Instead of taking one model to predict the same shower, we take the opposite approach. We train N NNs under the same pre-defined conditions and, afterward, predict with each of these NNs on the same test data set, gaining N different predictions for each event. From these predictions, we compute ς to determine the spread of the predictions (see Fig. 7.43). In the entire study, we use the data set defined in Row 5.5.e.

For this analysis, we use a continuous simulation library (see Sec. 5.1.1). We also assume that distribution of ς follows approximately a log-normal distribution due to the random processes involved in the NN training. We define it as

$$f_{\text{lo-no}}(x; A, B) = \frac{1}{xB\sqrt{2\pi}} \exp\left[-\frac{\ln^2(x/A)}{2B^2}\right], \tag{7.5}$$

where A and B are fit parameters. The mean value of a log-normal distribution is $\langle\varsigma\rangle = A\exp\left(B^2/2\right)$.

A Estimation of the effect of parallelization on NN predictions

We have trained a set of ten NNs under the following conditions

1. all models share the same architecture
2. all models give reasonable predictions without post-processing
3. all models share the same training data and predict on the same test data set
4. all models are trained with the same set of starting parameters and one GPU
5. all executing programs start from the same random seeds of all frameworks

The first condition is just for the sake of completeness. Condition two is a sanity condition. Mixing non-converging NNs into the study would warp our analysis. All other conditions have been chosen to have as little randomness as possible during the training. This leaves in the ideal case only the parallelization on the GPU.

In Fig. 7.46, we show the distribution of the ς for the models created using these conditions. There is a considerable spread between the predictions of different models for the same value of y_i. The assumption for a log-normal function (see Eq. (7.5)) is well met. Using the fit parameters, we obtain an average deviation of about $10\,\mathrm{g/cm^2}$. Since we removed all pseudo-random processes, this noise is irreducible.

Interestingly, even though the spread of predictions between models is relatively high, this has almost no effect on the precision of the models. The deviation of the precisions is

$$\sigma_{\text{GPU}} = 0.27\,\mathrm{g/cm^2}. \tag{7.6}$$

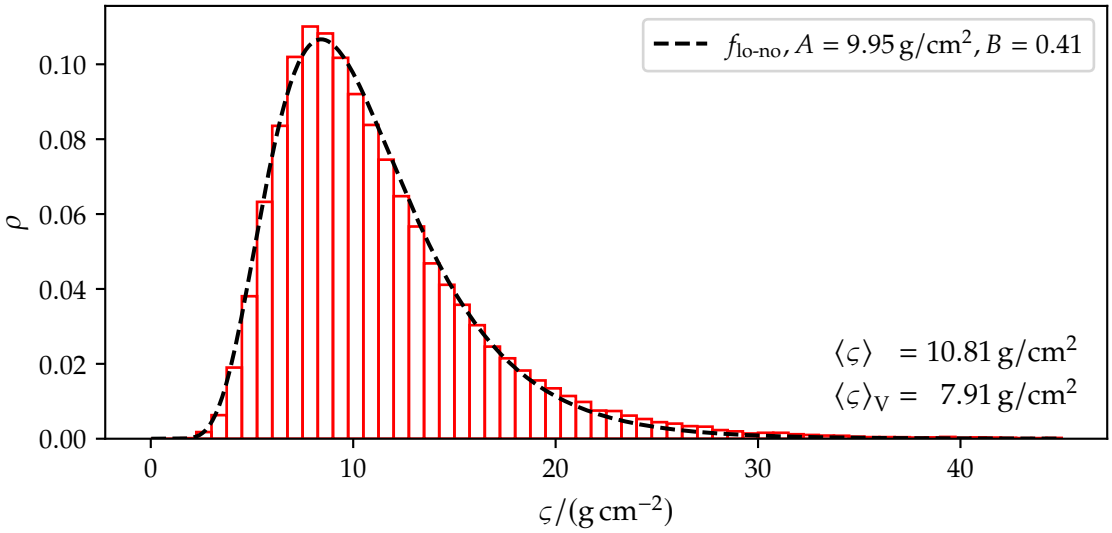


Figure 7.46: Distribution of ς (see Eq. (4.45)) for models created according to the conditions defined in Sec. 7.3.2.A. Even though we have reduced random effects in the training, there is a noticeable spread between the predictions of each of the models. Therefore, the random processes introduced due to the GPU training are significant.

Therefore, the spread of the predictions is not directly correlated to the "goodness" of the result.

Since the network uses a dropout layer in $\mathcal{AR}_{\text{iii}}$ to inhibit over-fitting, we have checked if removing it would have a profound impact on ς_i. This is not the case, as shown in Fig. D.18.

B Effect of adding pseudo-random number generation

We have trained a set of 40 NNs under the following conditions

1. all models share the same architecture
2. all models give reasonable predictions without post-processing
3. all models share the same training data and predict on the same test data set
4. all models are trained on one GPU

These are a relaxed version of the conditions in Sec. 7.3.2.A. The seeds for the random number generators are not fixed. As a consequence also, the starting weights between training runs are different. This has a direct effect on the spread of the predictions of the NNs. The random effects shift the distribution (see Fig. 7.47) slightly towards higher values. In addition, the processes involved also seem to make the distribution more complex. Even though the log-normal function does not correctly follow the distribution of ς it is still a good approximation. Using the fit parameters, we obtain a average deviation of about 11.6 g/cm^2. This is only about 1 g/cm^2 worse than the value we obtained in Sec. 7.3.2.A. Since we have included in this step all pseudo-random effects, we conclude that most of this deviation of the predictions is caused by the very complex solution space. Even small 'amounts of randomness' are sufficient to end up in different minima.

The pseudo-random processes also increase the spread of the precision slightly

$$\sigma_{\text{pseudo-random}} = 0.37\,\text{g/cm}^2. \tag{7.7}$$

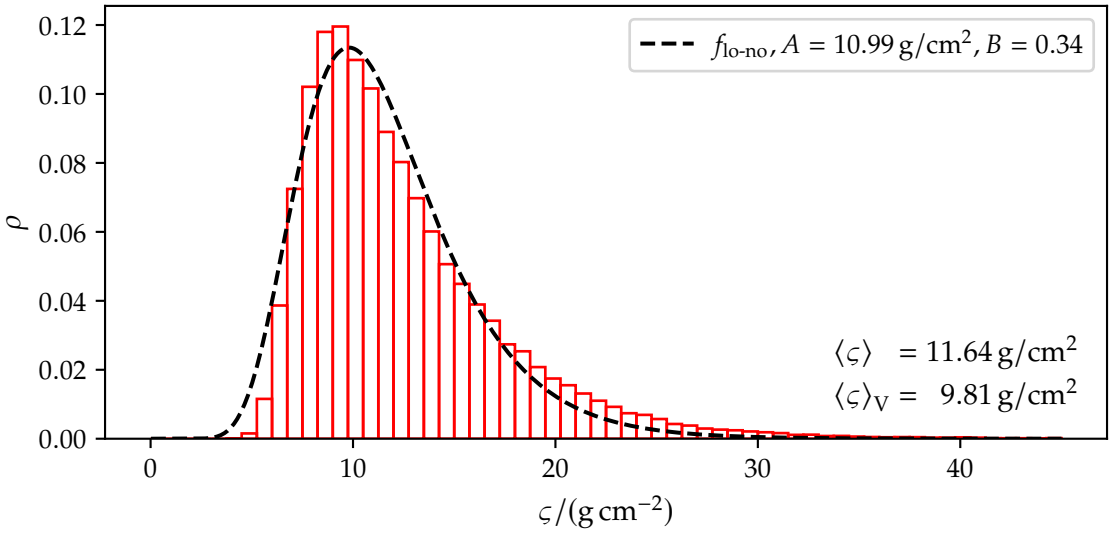


Figure 7.47: Distribution of ς for models created according to the conditions defined in Sec. 7.3.2.B. The pseudo-randomness we have introduced to the training process moves the center of mass of the distribution to higher values. The log-normal function does not catch this shift well indicating the increased complexity.

Unfortunately, it is unclear how to disentangle the effect caused by the GPU and by the pseudo-random processes correctly. Since the number of parameters of our NN is about $1{\times}10^6$, small changes in the start values yield significant differences in the final result.

C Constructing an uncertainty estimate from the previous results

Checking the effect of varying the input data additionally (see Appendix B.5), we obtain almost the same result as in Sec. 7.3.2.A and Sec. 7.3.2.B. Hence, we conclude that adding any noise on top of the irreducible noise of the NN training due to the GPU only increases the deviation of the predictions slightly.

Training only one network to estimate a physical quantity is, therefore, insufficient if NNs are used for the inference in physics. Overall the predictions might perform well, but single predictions are subject to this spread of prediction. Hence, we assume that the uncertainty we obtain from this analysis is systematic one that can not be reduced by the currently used setup. Since we performed all the studies with only small amounts of NNs, we approximate this uncertainty roughly by a value between the variance-weighted average and average value computed from the distribution of Sec. 7.3.2.B. We choose $10\,\text{g/cm}^2$ for single predictions.

7.3.3 Other sources of uncertainties

We know that the current hadronic interaction models do not reproduce the number of muons on ground-level correctly. In all of the models, slightly different strategies are utilized to simulate high-energy hadronic interactions, which result in slightly different predictions. Moreover, the exact energy dependence of the mass composition of UHECR is unknown. A bias that depends on the primary yields a shift in our predictions if low- or high-mass CRs dominate the spectrum. We have to account for both of these uncertainties.

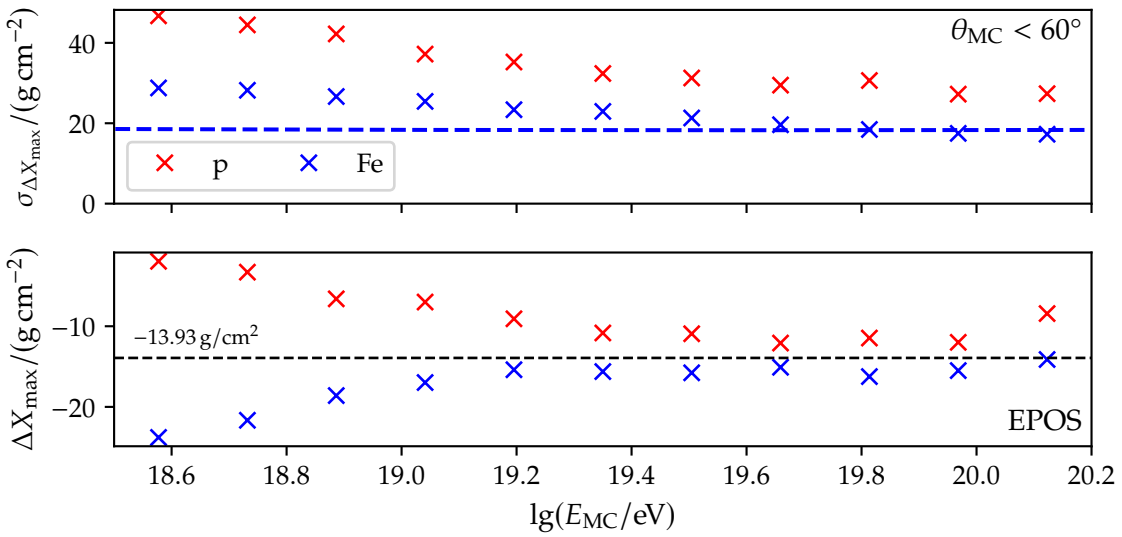


Figure 7.48: Precision $\sigma_{\Delta X_{max}}$ (*top*) and bias ΔX_{max} (*bottom*) for the NN predictions of the proton- and iron-part of the data set based on simulations using the hadronic interaction model EPOS-LHC. The NN has been trained on a data set using the hadronic interaction model QGSJetII-04. The curve in the *top* plot (blue, dashed) represents the width of the X_{max} distribution of the iron primaries. The precision of the predictions for the iron primaries is for most of the energy range above the width of the underlying X_{max} distribution of iron. The predictions of the NN have a global bias of about 14 g/cm^2. The horizontal line in the *bottom* plot (black, dashed) marks this global bias.

A Uncertainty due to hadronic interaction model

To estimate the dependence on the hadronic interaction model, we use the NN to predict on the data set based on air shower simulations simulated with the hadronic interaction model EPOS-LHC (see Row 5.5.f). However, the precision is in almost all energy bins larger than the width of the X_{max} distribution belonging to the iron primaries of the underlying data set. We attribute this to the difference between the X_{max} distributions for the different hardonic interaction models. More noticeably, we find a global bias of the predicted X_{max} values (see Fig. 7.48). On average, the predictions are shifted by −13.93 g/cm^2 (see Row D.1.a).

In Fig. 7.49, we compare the bin-wise precision of the predictions on the data sets simulated with different hadronic interaction models. We denote σ_{QGSJ} and σ_{EPOS} as the precision in the data sets defined in Row 5.5.d and Row 5.5.f. In each energy bin, the precision of the predictions for the data set simulated with the hadronic interaction model EPOS-LHC is smaller. Switching the interaction model reduces the precision by about 3 g/cm^2 at a low energy of $10^{18.6}$ eV and 1.5 g/cm^2 at energies above 1.5 g/cm^2 (see Row D.1.b).

B Uncertainty due to unknown composition

We use for all of the studies in this section the SD energy E_{SD} instead of the MC energy E_{MC} to account for the primary dependence of the energy estimator in measurements. Again, to account for the difference between simulations and measurements we shift E_{SD} like in Eq. (7.4).

The predictions of the NN depend on the primary particle which induced the shower (Fig. 7.50). We parameterize the difference between proton and iron events via two independent fits to the predictions on both event types. We utilize, in both cases, an exponential fit function (see Eq. (A.13)). The difference between both fits goes from 35 g/cm^2 at low energies

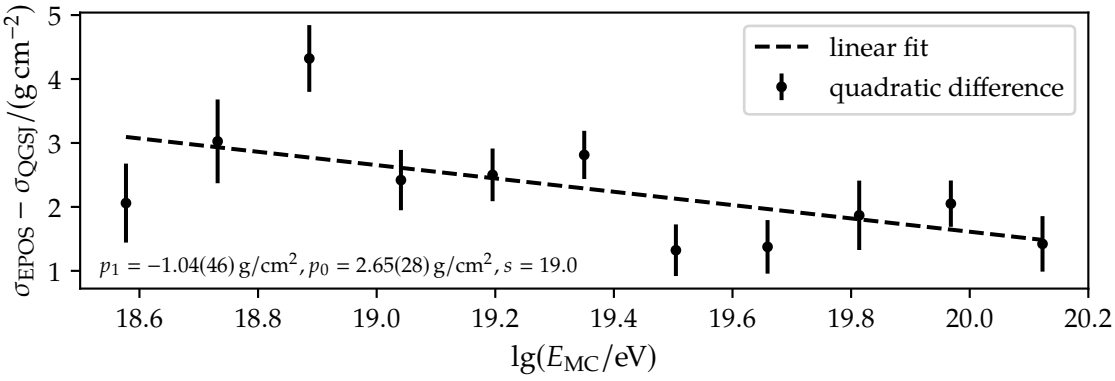


Figure 7.49: Difference of the precision of the NN-based predictions on the data sets defined in Row 5.5.d and Row 5.5.f in bins of energy. The line (black, dashed) is a linear fit to the difference (see Eq. (A.10)).

to 1 g/cm^2 at high energies (see Rows D.1.c to D.1.d).

To approximate the uncertainty on $\sigma_{X_{\mathrm{max}}}$ of the X_{max} distribution, we check the difference $\Delta\sigma_{X_{\mathrm{max}}}$ of the MC width $\sigma_{X_{\mathrm{max}}}$ and the width of the predictions for different compositions. We select the pure composition for each of the primaries, the composition containing only proton and iron primaries, and a composition of the underlying data set. In each bin, the under- and over-prediction are different for the different compositions. We attribute this variation to a systematic effect since the NN is not able to reproduce the distribution of X_{max} values correctly in the energy bins. We conservatively estimate the composition-dependent uncertainty by fitting two fit functions to the maximum and minimum values in each bin. For the maximum values, we use an exponential fit function (see Row D.1.e), and we choose a constant fit for the minimum values (see Row D.1.f).

C Uncertainty for relative muon content and logarithmic mass number

The uncertainty due to the unknown high-energy physics and the uncertainty due to the unknown composition for relative muon content R_μ and logarithmic mass number $\ln A$ are calculated the same way as presented in Sec. 7.3.3.A and Sec. 7.3.3.B. Hence, only differences between the approaches are highlighted. The results have been tabulated in Table D.1.

Relative muon content Instead of using exponential functions to parameterize the composition bias (see Fig. D.21) and the composition dependence of the resolution (see Fig. D.22), linear functions have been fitted to the corresponding data. Because of missing CORSIKA files, the dependence on the hadronic interaction model was fitted to a data set comprised of only proton and iron events. The fits are presented in Figs. D.19 to D.22. The fit parameters are recorded in Rows D.1.g to D.1.l.

Logarithmic mass number Like in the case for the R_μ parameterization, instead of exponential functions, we have used linear functions to fit the behavior of the composition dependencies. To estimate the resolution dependence, a quadratic model has been used. The fits are presented in Figs. D.23 to D.26. The fit parameters are recorded in Rows D.1.m to D.1.r.

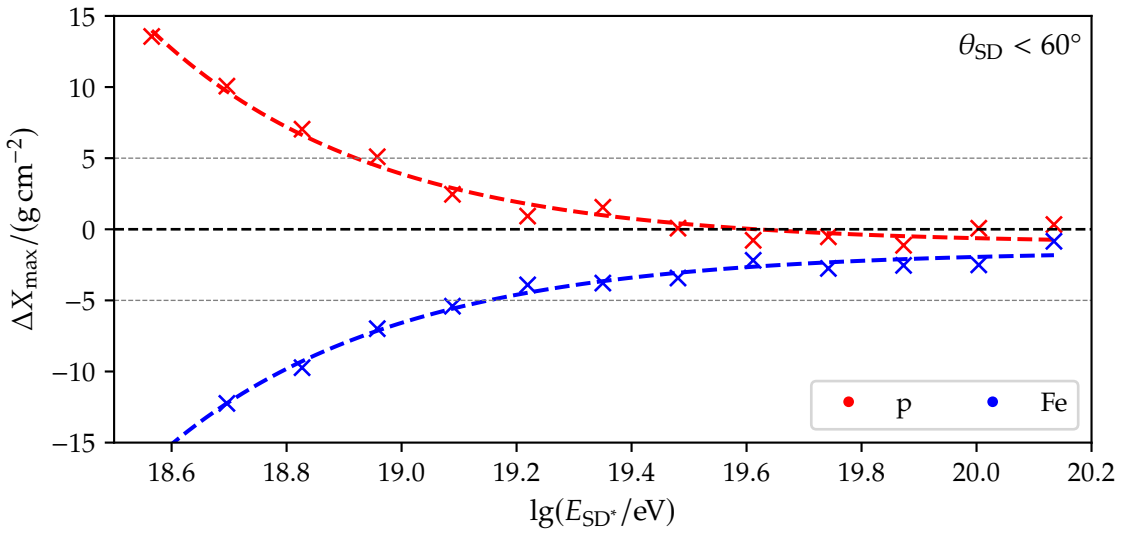


Figure 7.50: The bias ΔX_{max} for events belonging to proton (red) and iron (blue) primaries. The dashed lines are exponential fits to both of the sub-sets.

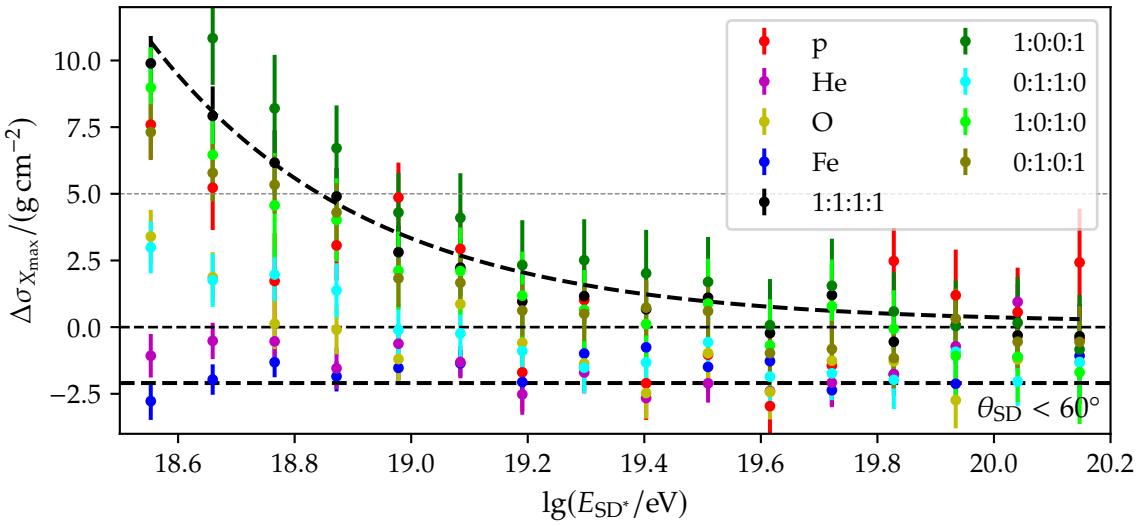


Figure 7.51: The difference of the width of the MC distribution and the width of the predictions for different compositions. We have fitted an exponential to three highest values in each bin and a constant to the minimum values.

7.4 Effect of using *AugerPrime* simulation data

To investigate the effect of the new UUB electronics and the additional SSD detector, we perform a similar study to that in Sec. 7.2. However, instead of using a complex NN, we fall back to the network architecture used in Sec. 7.1. In this way, we ensure that the only improvement is from the additional information of the UUB and the SSD. In addition to this restriction, we use the small data set defined in Row 5.5.g to speed up the training process and allow us to train more independent networks.

7.4.1 Effect of the UUB

The better sampling of the UUB electronics allows for a better sampling and measurement of the time signal direct after triggering. Since it is directly related to the muon sub-component of the air shower, this potentially gives us essential information about the first interaction.

A Effect of increased sampling rate

For the initial test, we use the RNN-based sub-network for $\mathcal{AR}_{\mathrm{i}}$. This choice allows us to use the same network architecture for different trace lengths L_t, making the tests more comparable. To cross-test the results, we compare the results of the NNs to the RNN-based baseline networks trained for the tests in Sec. 7.1.3.A. We have generated three ensembles of NNs consisting of 5 NNs each using the trace lengths: 120, 240, and 360.

Since the trace lengths are lower than the UB trace lengths 40, 80, and 120, we ensure that the performance of the UUB-based networks cannot come from extra bins. Despite having a much lower statistic, the networks trained on UUB data outperform the networks trained on UB data if more than 120 UUB bins are used (see Fig. 7.52). Moreover, the NN trained on data with trace lengths of 240 UUB bins also has a slightly lower proton-iron bias (see Fig. 7.53).

This improvement of the predictions substantiates the claim that the increased sampling rate increases the amount of useful information in the traces. We gain a more precise and accurate prediction compared to using less than 80 UB bins.

B Analysis of different downsampling strategies

Due to the increased sampling rate of the UUB electronics, UUB traces contain approximately three times as many bins as their UB counterpart. Since the increased sampling rate improves the precision and reduces the proton-iron bias, we have to transform the UUB traces into UB-like traces. In this way, we ensure that any improvement of the predictions of the NNs is due to the additional signal information, e.g., the signal of SSD detectors. In $\overline{\text{Off}}\underline{\text{line}}$, a procedure is implemented that generates UB equivalent traces from UUB traces by convolving the UUB traces with a predefined constant filter. We refer to this process as downsampling. In this section, we analyze other downsampling strategies which are computationally inexpensive when the on-the-fly data augmentation is used (see Sec. 5.3.4). In each analysis, an initial trace length of 360 UUB bins is used. All discussed downsampling strategies 'compress' these 360 bins to 120 bins.

Since three UUB bins are equivalent to one UB bin and the traces in the input data sets are trimmed to a trace length of L_t trace bins starting from the start bin, we analyze strategies that average over three consecutive bins. We start by comparing the arithmetic mean over three consecutive bins if computed from the normalized, logarithmic trace signals used as input of the NNs (see Sec. 5.3.1.C) and from the original traces. Since we have to transform the traces

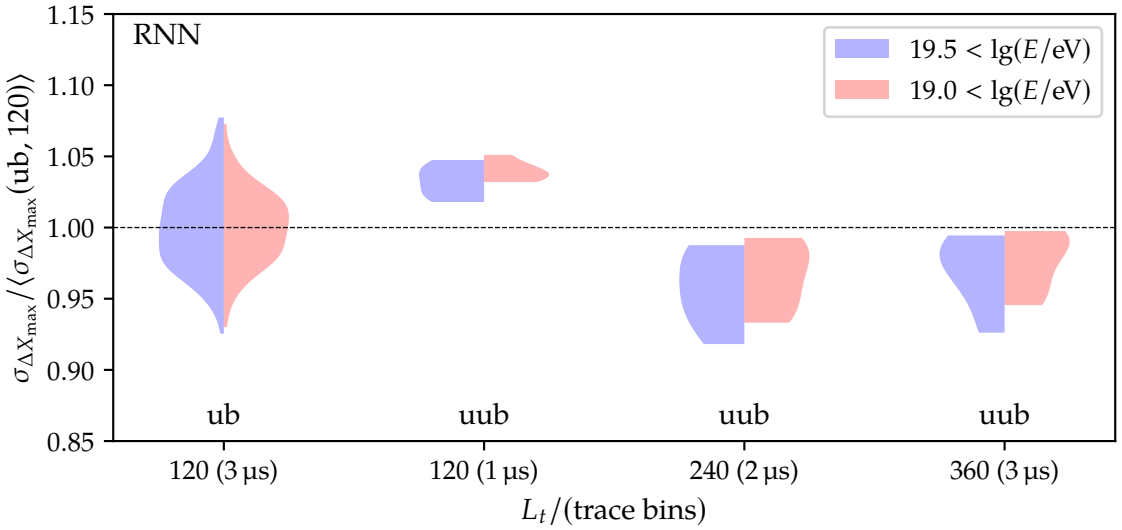


Figure 7.52: Effect of the input trace length L_t on the precision $\sigma_{\Delta X_{\max}}$ of the predictions for ensembles of NN-based predictions normalized to the average standard deviation of the network using 120 UB bins. We compare the predictions of the RNN-based NNs trained for Sec. 7.1.3.A to the ensembles of predictions of RNN-based NNs trained on data based on UUB simulations.

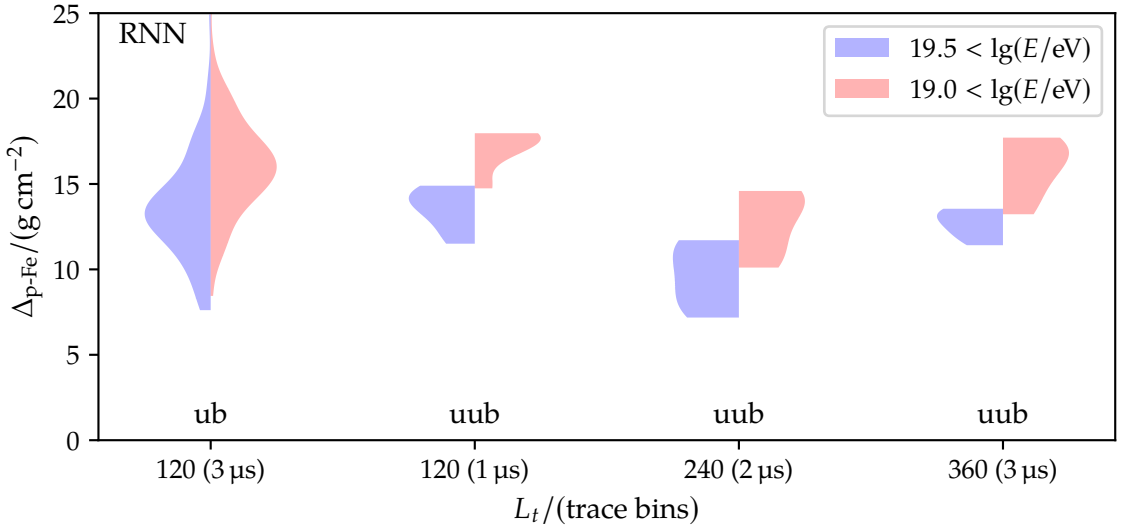


Figure 7.53: Distributions of proton-iron bias $\Delta_{\text{p-Fe}}$ for predictions of ensembles of NNs trained with different input trace lengths simulated with UB and UUB electronics. The numbers on the x-ticks denote the length of the input trace. The first number is the number of sampling bins and the second, in the parenthesis, denotes the physical trace length in µs.

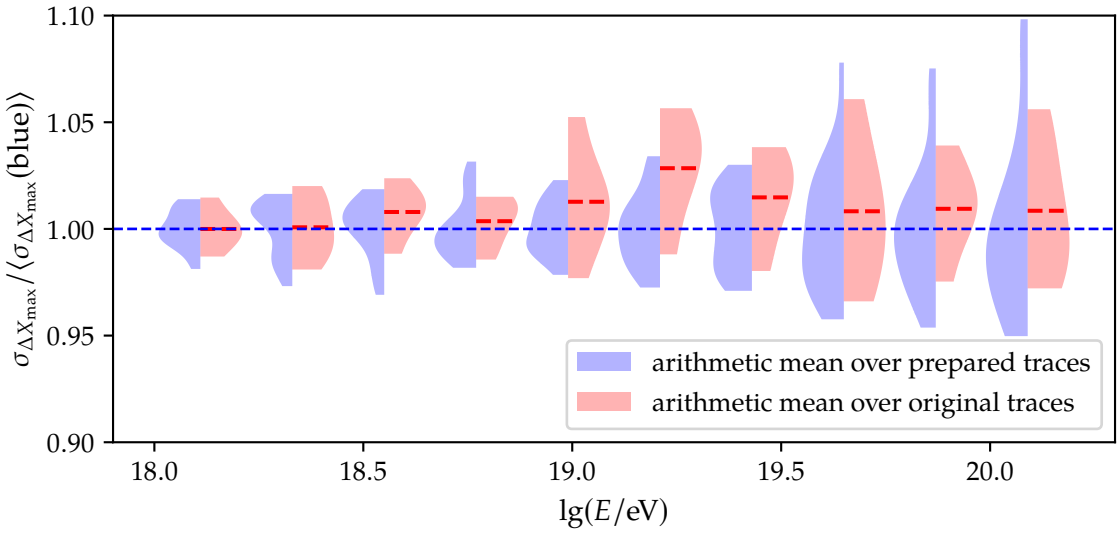


Figure 7.54: Comparison of ensembles of the resolution $\sigma_{\Delta X_{max}}$ of predictions of NNs trained with downsampled traces using the arithmetic mean of the prepared traces (blue) and the arithmetic mean of the original traces (red) relative to the average precision of the former averaging procedure.

back to compute the average of the latter, the former is computationally inexpensive. However, it is also the correct strategy. For high signals, the arithmetic mean over the logarithmic trace signals corresponds approximately to the geometric mean over three consecutive bins of the original trace. The predictions of ensembles of NNs trained on traces subject to either of the downsampling strategies show equivalent results (see Fig. 7.54 and Fig. D.27). The improvement in sub-1 g/cm^2 is due to fluctuations in the training and the small sample size. Hence, we can safely use the average over the prepared trace as a downsampling strategy.

The result in Fig. 7.54 raises the question of how the averaging over three consecutive bins influences the result. Computing the arithmetic mean over sequential data functions like a primitive noise filter that smooths the measurement. Since the noise is partially suppressed, one might presume that this would impact the training process, improving the predictions of the emerging NNs. This is not the case. To show this, we define three additional downsampling strategies that do not possess the filtering property. We split the trace into consecutive blocks of three bins. From these blocks, we can generate a trace with the desired trace length by selecting the n$^{\text{th}}$ bin in each of the blocks. If we compare the precision of 20 networks for each of the four downsampling strategies, there is no clear indication that averaging over multiple sequential bins yields a benefit (see Fig. 7.55).

In Sec. 7.4.1.A, we claimed that earlier parts of the traces contain more important information than later parts. We test this assertion with an importance-based downsampling strategy. Instead of considering groups of three sequential bins corresponding to a UB bin, the mapping between the initial trace bins and the downsampled trace bins is chosen in such a way as to increase the number of bins corresponding to earlier trace signals in the downsampled trace. Since PMT signals induced by particles exhibit an exponential decay (see Eq. (3.7)), a logarithmic map is a convenient way to achieve the desired importance sampling. In Appendix C.2, the indices of the initial trace bins, which have been used to generate the final trace, are documented. Even though the trace loses the temporal consistency, the ensemble of NNs trained on the traces which have been downsampled with the logarithmic map slightly outperforms the ensemble of NNs trained on the traces downsampled with the

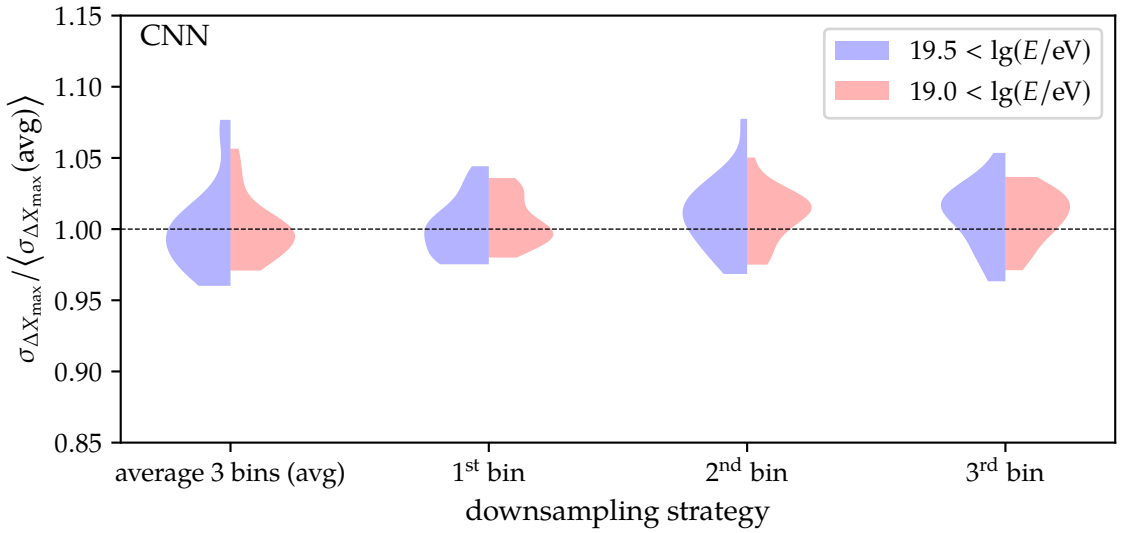


Figure 7.55: Comparison of ensembles of the resolution $\sigma_{\Delta X_{max}}$ of predictions computed for energies above $10^{19.5}$ eV (blue) and 10^{19} eV (red) of CNN-based NNs trained on four differently downsampled traces. All downsampling strategies shown act on blocks of three consecutive bins transforming a UUB trace of 360 bins to a length of 120. For the first strategy, the arithmetic average of each block is computed (see Fig. 7.54). The other strategies take the n^{th} bin of each block. The precisions are normalized to the average precision of the NNs trained with the downsampled traces generated by using the arithmetic mean.

arithmetic mean procedure (see Fig. 7.56). This implies that the earlier bins are more important for the overall prediction of the maximum shower depth. Nevertheless, using the logarithmic downsampling also increases the spread of precision distributions in every energy bin, indicating a more unstable training process. We conclude that even if it seems that the logarithmic sampling improves the predictions of the NNs slightly, more studies are needed to draw a final conclusion. Taking the networks with the best predictions, we find only a marginal improvement in the mid-energy range (see Fig. D.28). However, we cannot exclude that this is an artifact of our selection procedure to find the NN with the best-performing predictions (see Sec. 7.2.1).

7.4.2 Effect of SSD information on NN predictions

To determine the effect of adding SSD detector traces to the input of the NNs, we downsample traces from a UUB-trace length of 360 bins to a 120 UB-equivalent length by taking the arithmetic mean over three consecutive bins (see Sec. 7.4.1.B). In this way, we ensure that a direct comparison to the baseline networks in Sec. 7.2 is possible and any improvement is due to the inclusion of the SSD itself and not an effect of the sampling rate of the UUB (see Sec. 7.4.1.A). Hence, for each target, we have two sets of NNs. We denote the set of NNs using WCD and SSD trace inputs as WCD+SSD and the set of NNs using only WCD trace inputs as WCD. We exclude the phase space in which the effective size of the SSD detector is small. Only events in the test data sets are used that have a zenith angle of below 50°.

A Comparison of the predictions of directly accessible targets

We start by comparing the predictions of NNs of the zenith angle θ and the energy E in terms of $\sin^2$ and ln, respectively. For both targets, the precision of the predictions of the en-

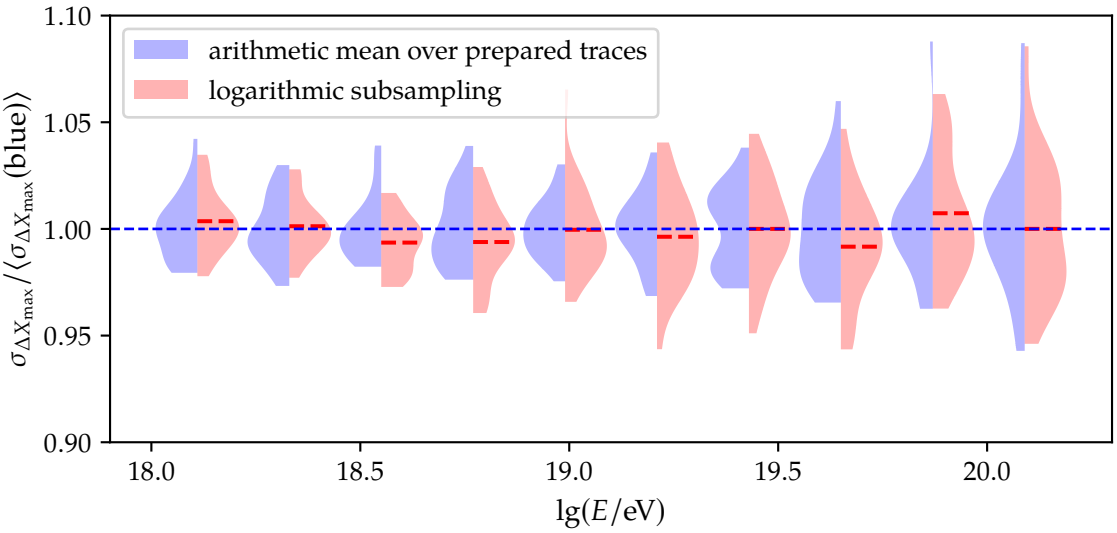


Figure 7.56: Comparison of ensembles of the precision $\sigma_{\Delta X_{max}}$ of predictions of NNs trained with downsampled traces using the arithmetic mean of the prepared traces (blue) and the logarithmic importance sampling (red) relative to the average precision of the former averaging procedure.

sembles of the NNs trained with the SSD trace data shows no clear improvements compared to that without the SSD trace data (see Fig. 7.57 and Fig. 7.58). The large distance between the centers of distributions in the second bin in Fig. 7.57 and the fourth and seventh bins in Fig. 7.58 are caused by outliers[3] in the corresponding energy bins. There is no striking trend in the ensembles of resolutions of the NNs using the SSD traces and NNs using only the WCD traces. The zenith predictions of both ensembles of networks show a proton-iron bias consistent with zero (see Fig. D.34). Therefore, they are both similarly unbiased for different primaries.

Fig. 7.59 shows the proton-iron bias of the energy predictions of the two corresponding sets of NNs. The decrease of bias at lower energies is due to the proximity to the edge of our data set in a region of low resolution. At high energies, the ensemble of NN predictions using additional SSD data shows an overall smaller proton-iron bias. Since the resolutions of both sets are the same in the same energy interval, we attribute this to the additional SSD data. The additional decrease in bias in the seventh bin is due to the outlier in the data set containing WCD and SSD traces (see Fig. 7.58).

The predictions of the depth of the shower maximum X_{max} of both sets NNs appear to have no apparent trends in the precision (see Fig. 7.60) and proton-iron (see Fig. 7.61) ensembles. The increase of precision at energies below $10^{18.6}$ eV is in a region in which the SD detector is not 100% efficient. In the interval $\left[10^{18.6}\,\text{eV}, 10^{19.1}\,\text{eV}\right]$, the shifts in the distribution of resolutions coincide with the shifts in the distributions of the proton-energy bias. However, since this improvement of the predictions is only observed for two bins in the full efficiency range and vanishes afterward, we cannot exclude that this is due to an artifact[4] of the test data set used to evaluate the NNs using only the WCD.

We depicted the precision of the predictions of the 'best-performing' NNs for each of the targets and sets in Figs. D.29 to D.31.

[3] We did not do any cuts on the test data except of the zenith cut mentioned in the beginning of the chapter since we want to show the overall behavior.

[4] We tried to remove extremely high-valued outliers from the data set used to evaluate the NNs using only WCD trace data. However, this procedure did not remove the shifts.

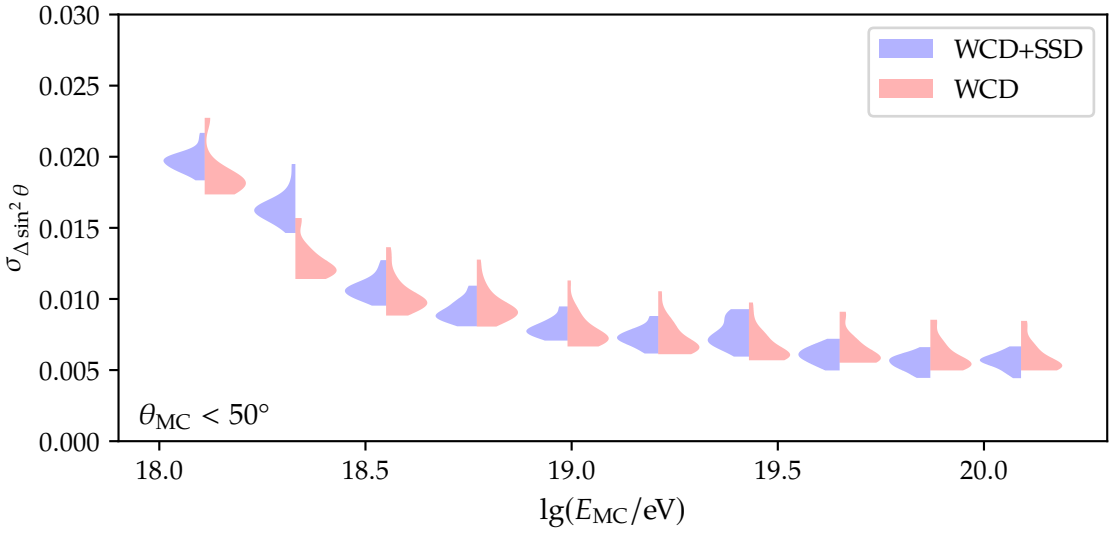


Figure 7.57: Ensembles of the precision of $\sin^2 \theta$ predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set with a zenith angle below 50° are considered.

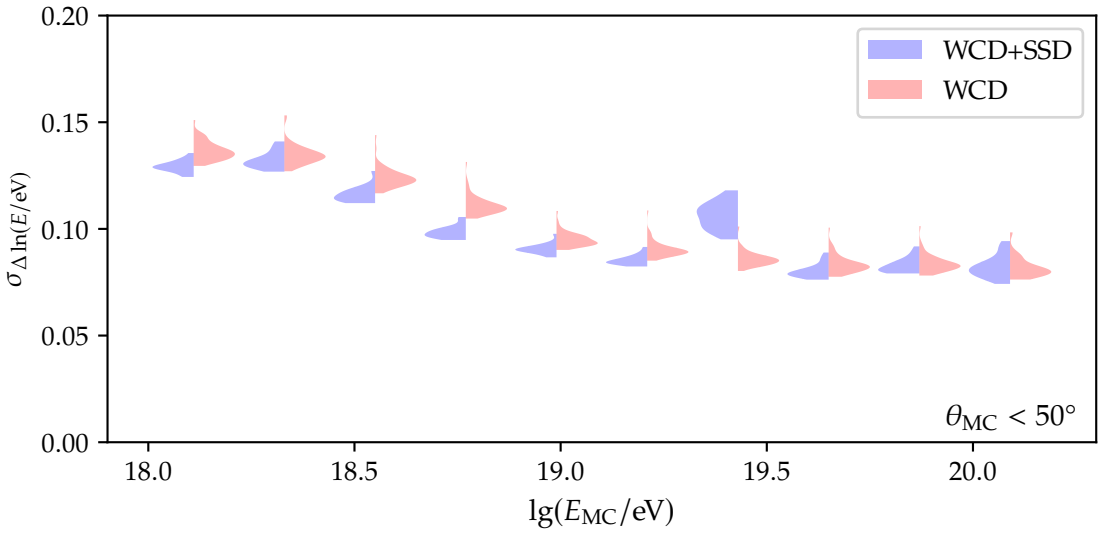


Figure 7.58: Ensembles of the precision of $\ln E$ predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set with a zenith angle below 50° are considered.

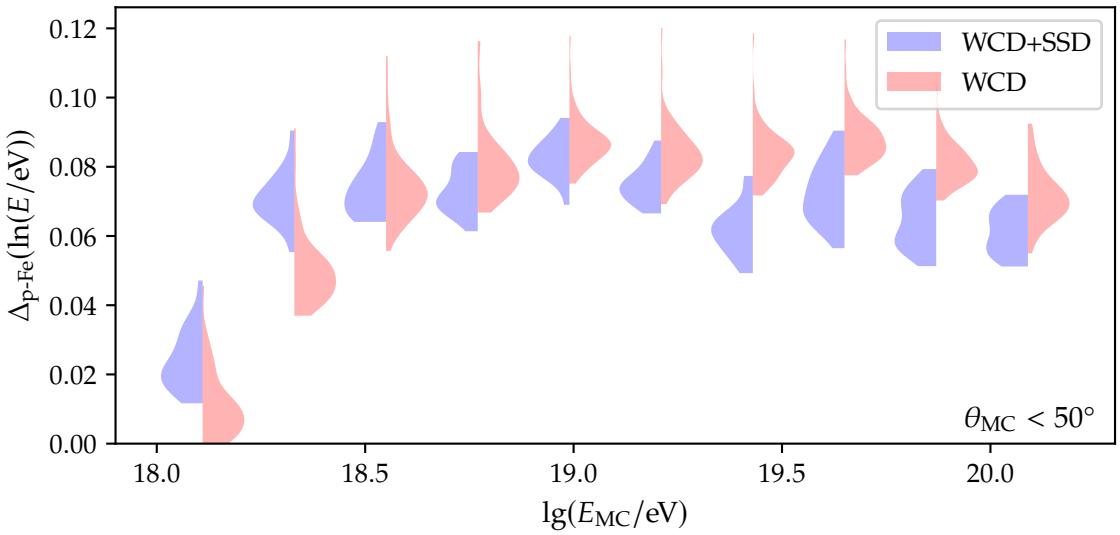


Figure 7.59: Ensembles of the proton-iron bias of $\ln E$ predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set with a zenith angle below 50° are considered.

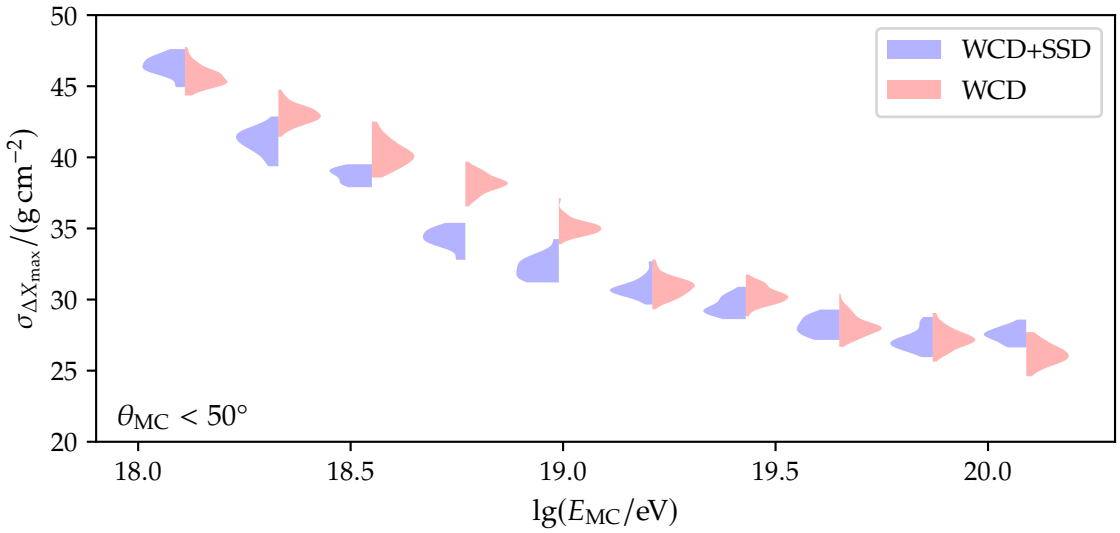


Figure 7.60: Ensembles of the precision of X_{max} predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set with a zenith angle below 50° are considered.

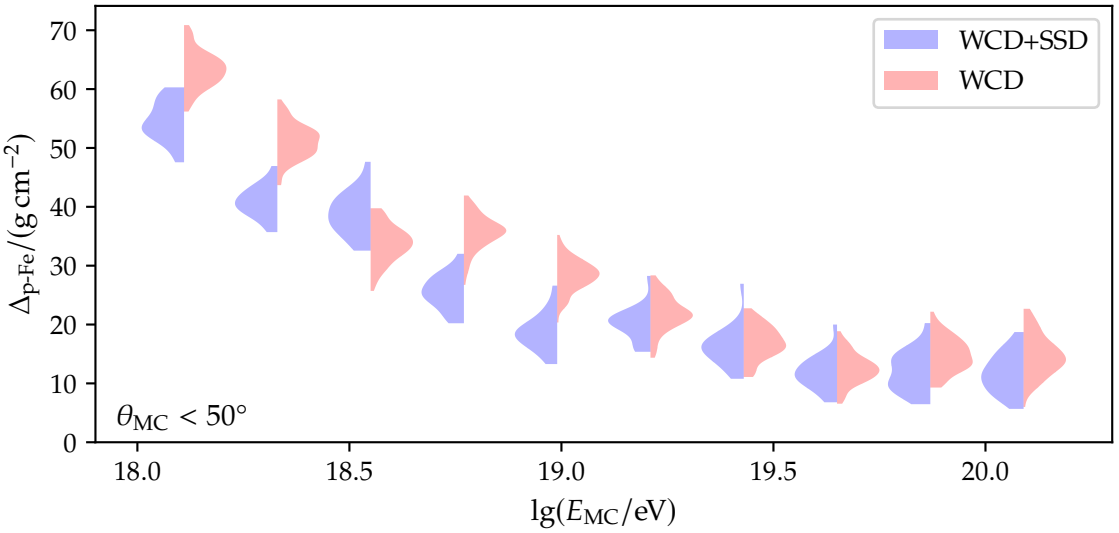


Figure 7.61: Ensembles of the proton-iron bias of X_{max} predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set with a zenith angle below 50° are considered.

B Effect on the prediction of targets without standard methods

The relative muon content R_μ and the logarithmic mass number $\ln A$ are inaccessible by regular air shower reconstructions. A direct way of predicting any of both air shower properties provides essential knowledge for composition studies. Hence, any improvement of reliable methods is beneficial for any of the goals of instruments such as the Auger observatory.

We start by comparing the ensembles of R_μ predictions from NNs using SSD information and from NNs using only WCD information. Except at the edges of the investigated phase space, the precision of the predictions of the former NNs is, on average, higher than the precision of the latter NNs (see Fig. 7.62). Since the relative muon content R_μ is almost constant for the entire energy range, the increased resolution corresponds to a better separation of proton and iron events. One way of substantiating this claim is to compute the average difference between the mean predictions of iron events and the mean predictions of proton events (see Fig. 7.63). In every energy bin, the separation of proton and iron predictions is more significant for the NNs using SSD information is larger than that of NNs using only WCD information. The error bars in Fig. 7.63 are the standard deviations of the underlying distributions. Hence, the distributions of the differences in each energy bin are significantly separated. The diminished resolution in the high-energy bins is likely an effect of the decrease in the average R_μ value for iron (see Sec. 5.4.3.B). Adding the ratio of the integral over the SSD and WCD traces as an additional input, like in Sec. 6.3.2, does not affect the precision and has only a minor effect on the proton-iron separation (see Fig. D.32 and Fig. D.33). Since the ensemble of NNs using this additional information does not improve noticeably, we conclude that the ratio is already extracted from the raw traces, if it would be beneficial for the predictions. As already implied by the results in Sec. 7.4.2.A, this result on R_μ demonstrates that by adding the SSD information, an NN-based approach achieves a reduced proton-iron bias.

We corroborate this statement further by a comparison of the $\ln A$ predictions of NNs for both input types. Both Fig. 7.64 and Fig. 7.65 show a similar result to the equivalent figures of R_μ. The predictions of NNs using SSD inputs exhibit a higher resolution and show a better separation of proton and iron values. Note that the average difference at high energy values

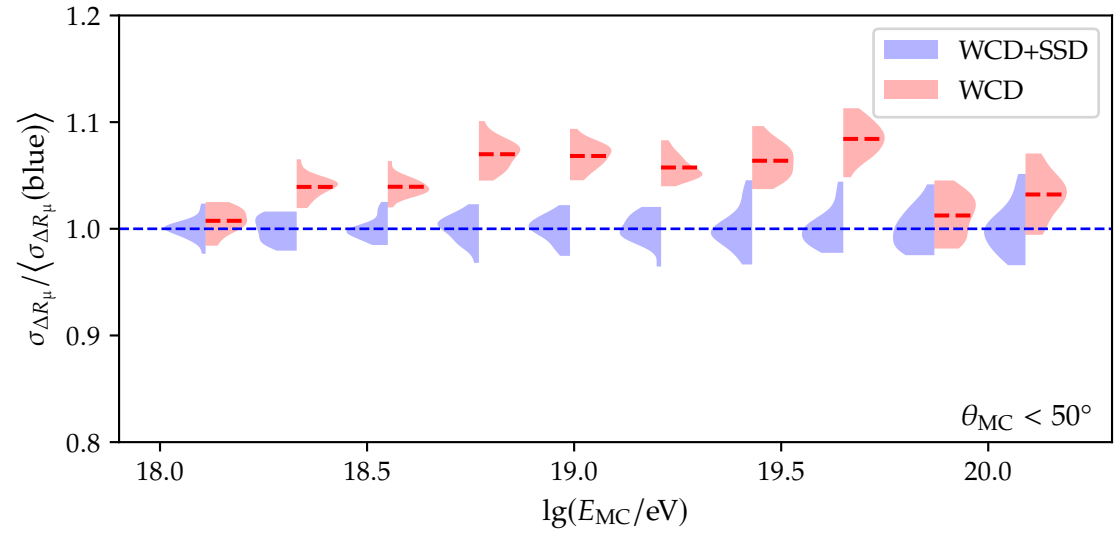


Figure 7.62: Ensembles of the precision of R_μ predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. The distributions are normalized to the average precision of the former set of NNs. The red dashed lines indicate the average value of the precision of the same colored distribution of resolutions in the same energy bin. Only events in the test data set with a zenith angle below 50° are considered.

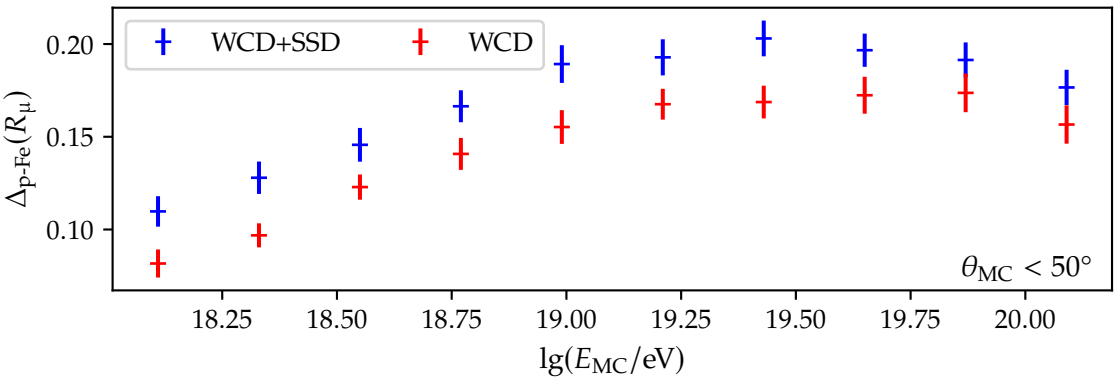


Figure 7.63: Average difference between R_μ predictions of iron events and proton events for predictions NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set exhibiting a zenith angle below 50° are considered. Note that for the error bars the standard deviations of the underlying distributions have been used.

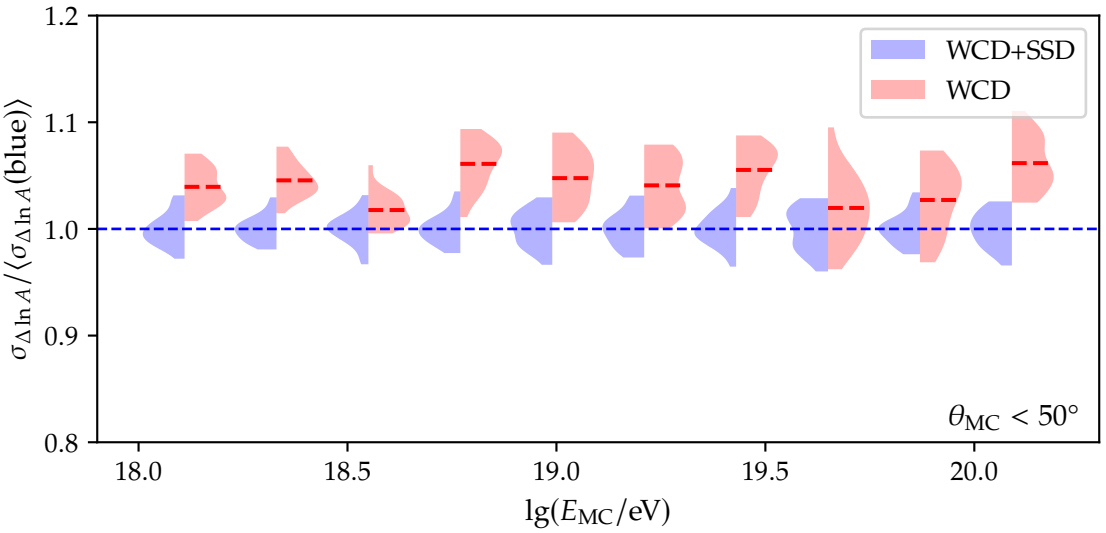


Figure 7.64: Ensembles of the precision of $\ln A$ predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces normalized to the average resolution of former set of NNs. The red dashed lines indicate the average value of the precision of the same colored distribution of resolutions in the same energy bin. Only events in the test data set with a zenith angle below 50° are considered.

of about 2.8 is still far below the ideal value of 4. Still, if we compare Fig. 7.65 to Fig. 7.37, we find a similar improvement over the result of the baseline network. Therefore, adding the SSD information is roughly similar to using an optimized architecture and training setup with over four times more training events. This improvement demonstrates the potential of the SSD detector.

We repeat the study of the merit factor done in Sec. 7.2.5 to corroborate this result. Using SSD information, shows an increase in the merit factor in most of the energy bins (see Fig. 7.66) compared to a NN using only WCD information. In almost all remaining bins, the merit factor of the former NN is inside the 1.2σ interval of the merit factor of the latter. Hence, we presume that most of these 'outliers' are due to statistical fluctuations and the small size of the used training data set. The predictions of the NN using SSD information, especially in the logarithmic energy range of $[19.6, 20.0]$, are comparable to those found in Sec. 7.2.5.

7.5 Classification of Photon Events

In Sec. 7.2.5, we have already seen that our here-employed architecture has been able to predict the logarithmic mass number $\ln A$. Moreover, for this specific task, the additional SSD input seemed to have a beneficial impact on the prediction quality (see Sec. 7.4.2.B). It is reasonable to test how far we can broaden this approach.

At least for lower energy ranges, a fraction of CRs are photons. Their shower cascade is solely electromagnetic (see Sec. 2.2.3.A). This should make it more or less distinct if compared to a hadronic shower. Consequentially, we can try to train a NN to differentiate between photonic and hadronic showers using our base architecture. Until now, we mainly used NNs in regression tasks. In this case, it makes sense to switch to a classification approach (see Sec. 4.2.4).

We give photon showers the label 0 and hadron showers the label 1. We draw only showers

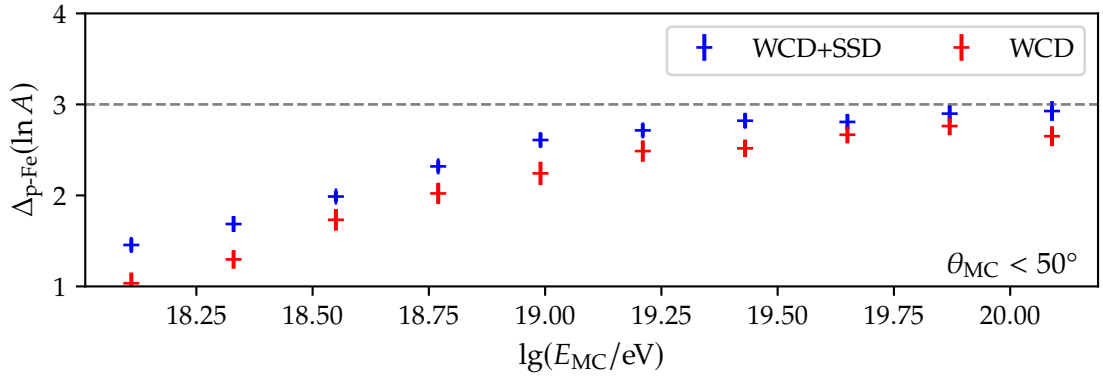


Figure 7.65: Average difference between $\ln A$ predictions of iron events and proton events for predictions NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set with a zenith angle below 50° are considered.

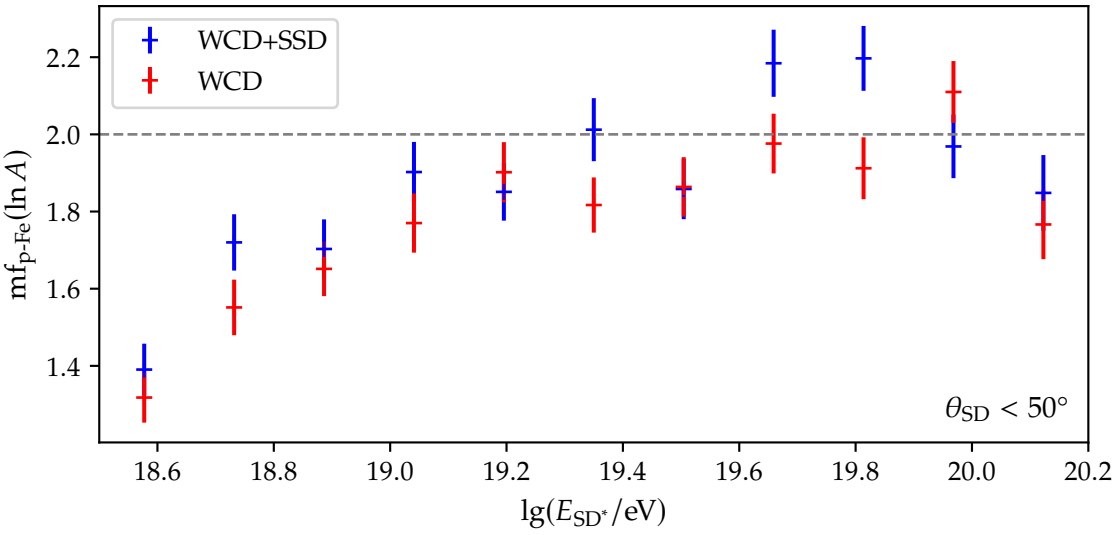


Figure 7.66: Merit factor of $\ln A$ predictions of proton and iron events binned in $\lg E_{SD^*}$ for events with a reconstructed zenith angle below 60°. The predictions are from an NN (blue) using WCD and SSD traces and an NN using only WCD traces as inputs. The dashed, gray line marks a merit factor of 2.

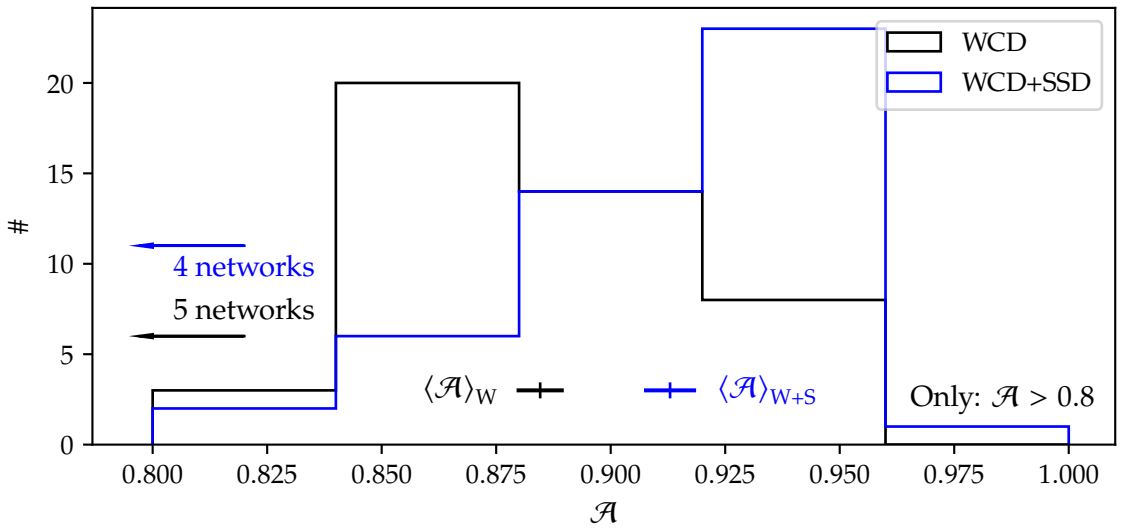


Figure 7.67: Histograms of the accuracy $\mathcal{A}$ of the predictions of 20 NNs trained on WCD and WCD+SSD data for the setup described in Sec. 7.5. We leave out all models that exhibit an accuracy of under 80% on our test set. If we solely compare the average result of all of the models, we obtain only a small improvement by adding the SSD traces to our network. However, the distribution of the models using the SSD is shifted visibly to higher values of the accuracy.

from an energy range of $\mathcal{E} \in [19.0, 20.0]$. We use a 2:1:1 fraction of photon, proton, and iron showers. Our training set consists of about 100 000 showers (see Rows 5.5.h to 5.5.i). The footprint size is 7×7 and the trace length is 120 bins. In the case of UUB data, we average over three consecutive bins (see Sec. 7.4.1.B) to get the same number of bins as in the UB case.

We generate 20 models for WCD and WCD+SSD. We use a single (scalar) network output. We do not squash this output into an interval of $[0, 1]$. Hence, it is only a real number that gives us a "tendency" to which class the underlying event belongs to. A high number relative to the average value implies that the event is hadronic and a low number indicates that it is photonic. We define the average value of the predictions as the decision boundary between photonic and hadronic events. Doing this gives us Fig. 7.67. Even though we have added the additional SSD information, the difference between WCD and SSD is (on average) not very large. However, there seems to be a shift in the distribution for the SSD case. We attribute this to the different response and sensitivity of the SSD to the electromagnetic sub-component of the shower. It is more likely to obtain a better working network if SSD information is added to the mix.

In Fig. 7.68, we show the predictions of the best-performing models (in terms of accuracy) for the WCD and WCD+SSD setup. We plot the output distribution of the corresponding NN for the different classes of primaries. As expected, the separation gets better for heavier primaries. This could be very beneficial for real data since the events appear to be heavier due to the muon excess. The confusion matrices (see Eq. (4.30)) are only marginally different for both. Still, we correctly classify additionally 150 events more in the second case. Moreover, the distributions for all primaries appear more peak-like.

In Fig. 7.69 and Fig. 7.70 we show the accuracy $\mathcal{A}$ and merit factor binned in logarithmic energy $\lg(E/\mathrm{eV})$ and $\sin^2(\theta)$ for both NNs which exhibit the most accurate predictions. The accuracy saturates with increasing energy to over 95%. Excluding the last bin, there is no overly large zenith dependency. In almost all bins the network using SSD information outperforms the network without. We believe that the WCD+SSD network works worse for high zenith angles because of the strongly diminished effective area of the SSD detector in

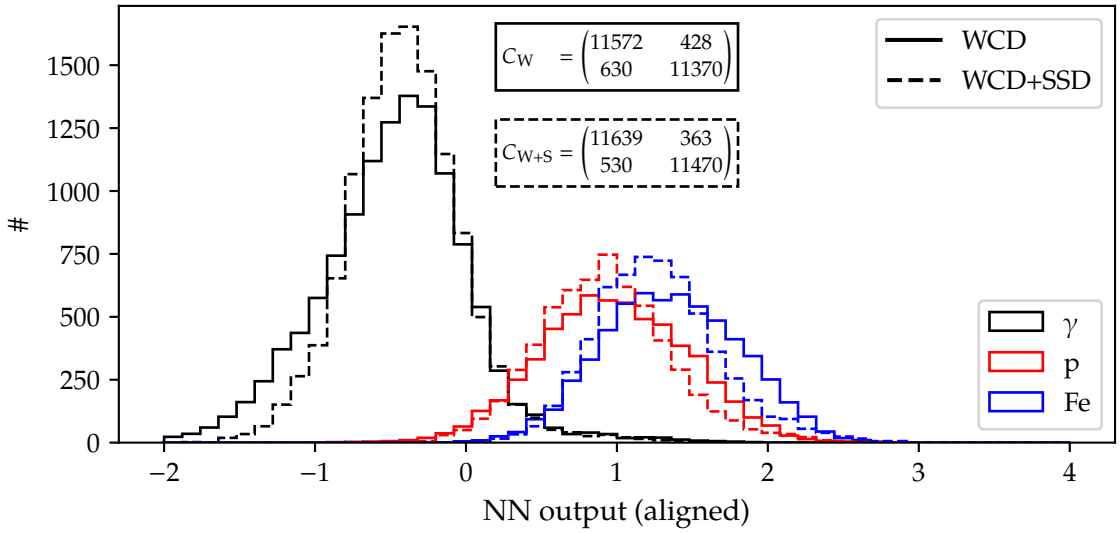


Figure 7.68: Distribution of NN output for the best performing models using the WCD and WCD+SSD setup split up into the primaries found in the test data set. To make the outputs comparable we have aligned them via the photon peak. In the center of the plot we added the confusion matrices (see Eq. (4.30)) for both setups. Even though they are only marginally different we see for the best case scenario an increase of the correctly classified events in the second setup.

this area. If during training, the network learns to rely on its signal to make predictions it could explain the worse performance at those zenith angles. Consequentially, we believe that SSD information is actually used to make the predictions and that this is not solely a statistical fluctuation of using too small networks. Still, we find also the non-SSD network very feasible. It remains to be investigated if these results could be extended to lower energies or in which phase space the networks perform best.

Moreover, comparing Fig. 7.69 and Fig. 7.70 we find that the merit factor is a quantity whose value is hard to interpret (see Sec. 4.4.2). Consistent values in accuracy correspond to large jumps in the merit factor for both types of networks. We could misinterpret the results in Fig. 7.70 thinking that our model performs much better. Therefore, we think it would be beneficial to never use the merit factor in the context of separation for future analyses in Auger again.

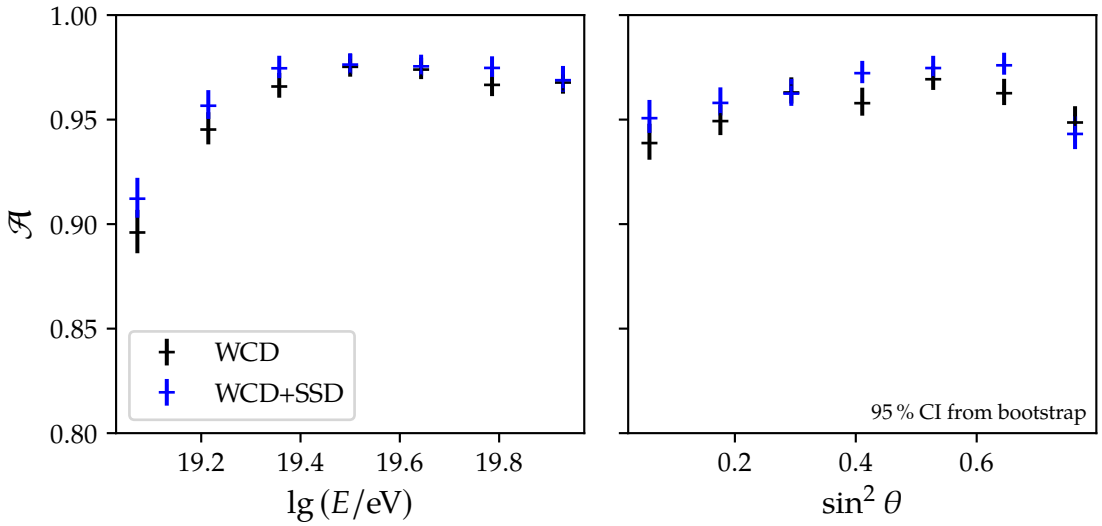


Figure 7.69: Accuracy $\mathcal{A}$ for bins in logarithmic energy $\lg(E/\mathrm{eV})$ (*left*) and $\sin^2(\theta)$ (*right*) for the best models in both setups. We used the bootstrap method to compute the 95% confidence intervals. Only for high zenith angles, the network using the SSD information does not outperform the other network. This is especially true for low energy events. We attribute the loss in accuracy for the last bin in the $\sin^2(\theta)$ binning as due to the low effective area of the SSD at this angles.

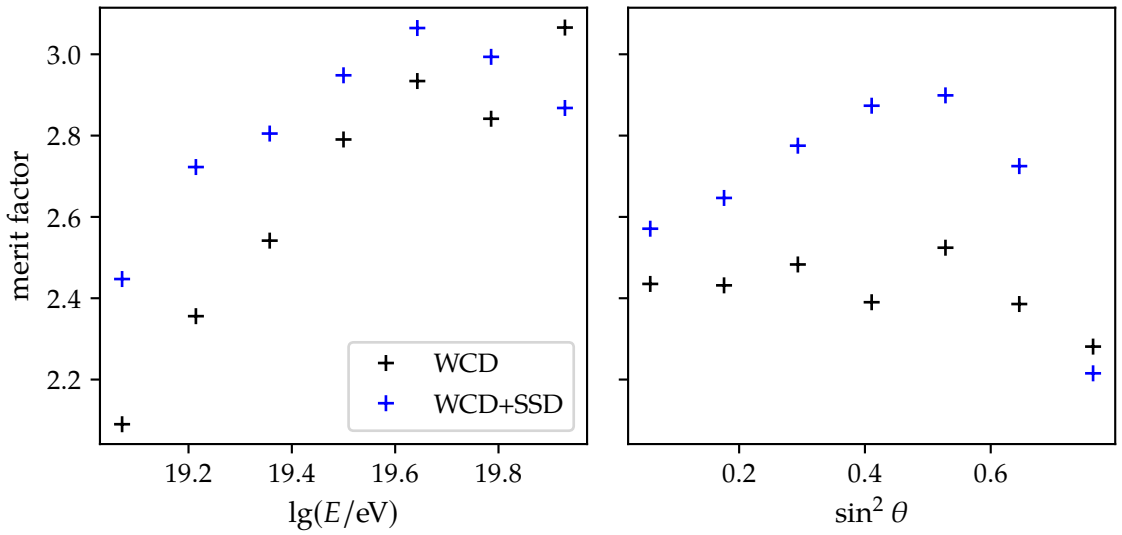


Figure 7.70: Merit factor for bins in logarithmic energy $\lg(E/\mathrm{eV})$ (*left*) and $\sin^2(\theta)$ (*right*) for the best models in both setups. Compared to Fig. 7.69 we find that the values suggest an extreme improvement using the SSD based model. This is especially dubious because of the last bin in the energy plot of Fig. 7.69. There the result of both models agree well while for the merit factor there is an increase. Comparing it to the previous bins the roles of both models switch only in this bin.

8 Extraction of Global Shower Properties from the SD and Golden Hybrid Data Set



Numero pondere et mensura Deus omnia condidit.

(Isaac Newton)

DALL·E 2 prompt:

A alien physicist on a treadmill that shoots out cosmic rays in the direction of Earth to fool all the scientists on Earth in the style of the mona lisa[.]

In Chapter 7, we have demonstrated that the NN-based approaches are suitable for predicting various event-level shower observables solely from the shower footprint. However, all of the preceding studies have been performed only on simulated air showers.

In this chapter, we present the application of selected NN models on air shower measurements taken at the Pierre Auger Observatory. Since we are especially interested in the mass composition of CRs, we concentrate on the NNs trained to predict the depth of the shower maximum X_{max}, the relative muon content R_{μ}, and the logarithmic mass number $\ln A$.

Due to the differences between air shower simulations and air shower measurements, it is necessary to correct the predictions of the NNs. To do this, we parametrize unphysical biases in Sec. 8.1 for each of our targets and remove them by subtraction. Afterward, in Sec. 8.2, we define high-quality subsets of our data sets and use the bias-corrected predictions in these subsets to find the first and second moment of X_{max}, R_{μ}, and $\ln A$.

8.1 Transition from predictions trained on simulations to measurements

There are fundamental differences between the air shower simulations of all hadronic models and measurements by Auger (see Sec. 2.4). Hence, a direct application of the NN-based models from Chapter 7 on measurements gives us only a raw, preliminary result. We have to correct for the most notable differences (see Sec. 5.2.3) and define cuts that remove measurements that cannot be adequately reconstructed by a method trained on simulation data, for example, measurements taken at extremely cold or hot temperatures. For this study, we use the NNs found in Sec. 7.2.

We base the corrections on unphysical dependence of the predictions from the NNs on other parameters. For example, the logarithmic mass number $\ln A$ does not depend on the time when the corresponding event has been detected. By parametrizing this dependence,

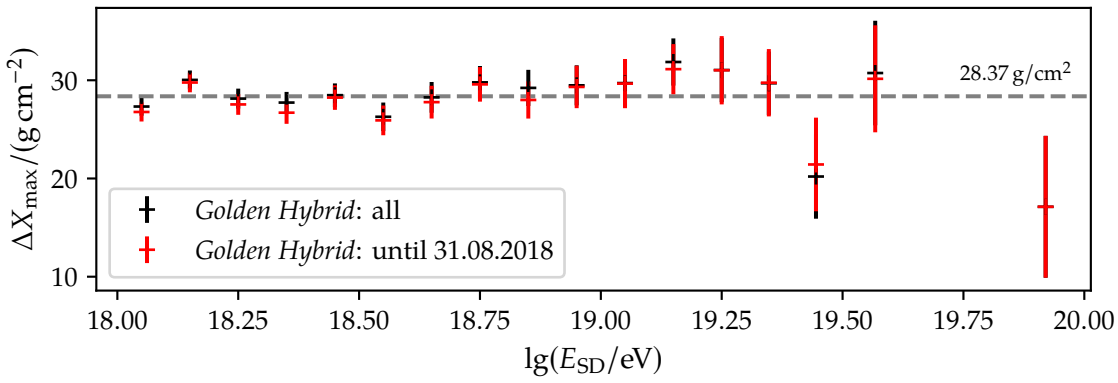


Figure 8.1: Difference $\Delta X_{\max}$ of raw NN predictions and $X_{\max}$ measurements by FD as a function of the reconstructed energy E_{SD}. The dashed gray line marks the average value of the differences. The black crosses mark the differences for the predictions on the entire *Golden Hybrid* data set and the red crosses the difference for all events measured before 31.08.2018.

we are able to correct each target individually by subtracting it from the original prediction. We divide our corrections loosely into three different categories: assumption, aging, and atmosphere.

In addition, we correct for the most striking difference between our air-shower simulations and the measurements. To generate the trace inputs for our networks, we have used the signal $S_{\mathrm{off}}(t)$ from the `SdRecStation:VEMTrace` vector in Offline. Consequentially, the trace is given in units of VEMPeak. In simulations, the VEM signal and $S_{\mathrm{off}}(t)$ differ only by a constant factor. That, however, is not the case for measurements due to the aging of the detector and the electronics. We correct for this by individually transforming the traces $S(t)$ from the measurements of all PMTs by

$$S(t) \to \frac{3.2}{a_{\mathrm{p}}} S(t), \tag{8.1}$$

where 3.2 is the fixed value of a_{p} in simulations.

8.1.1 Direct application of NN on measurements

Before the correction process, we ensure that the network predicts reasonable values for $X_{\max}$. We compare the prediction with the $X_{\max}$ measurements of FD on our *Golden Hybrid* data set (see Sec. 5.2.2). We find an almost constant shift over the entire investigated energy range (see Fig. 8.1), which is very similar to that described in [P:104]. Since the analysis in [P:104] has been conducted for *Golden Hybrid* events measured before , we added the differences for this period to check if our results are stable. We find only minimal changes at the evaluated energies, which are reasonably explained by statistical fluctuations. Averaging over all differences, we obtain a value of 28.37 g/cm^2 as the global shift between the raw network predictions and the FD measurements. Using the global shift, we calibrate the $X_{\max}$ predictions of our NN and compare them directly to that of FD.

In Fig. 8.2, the average reconstructed values of $X_{\max}$ as well as the standard deviation $\sigma_{X_{\max}}$ is depicted as a function of the reconstructed energy E_{SD}. The average values of the network predictions for both data sets[1] agree with the measurements of the FD. However, the predictions for the standard deviations differ. For energies below $10^{18.5}$ eV, the standard deviation

[1] A priori, it is not clear if the shift also works for the different phase space of the SD data set.

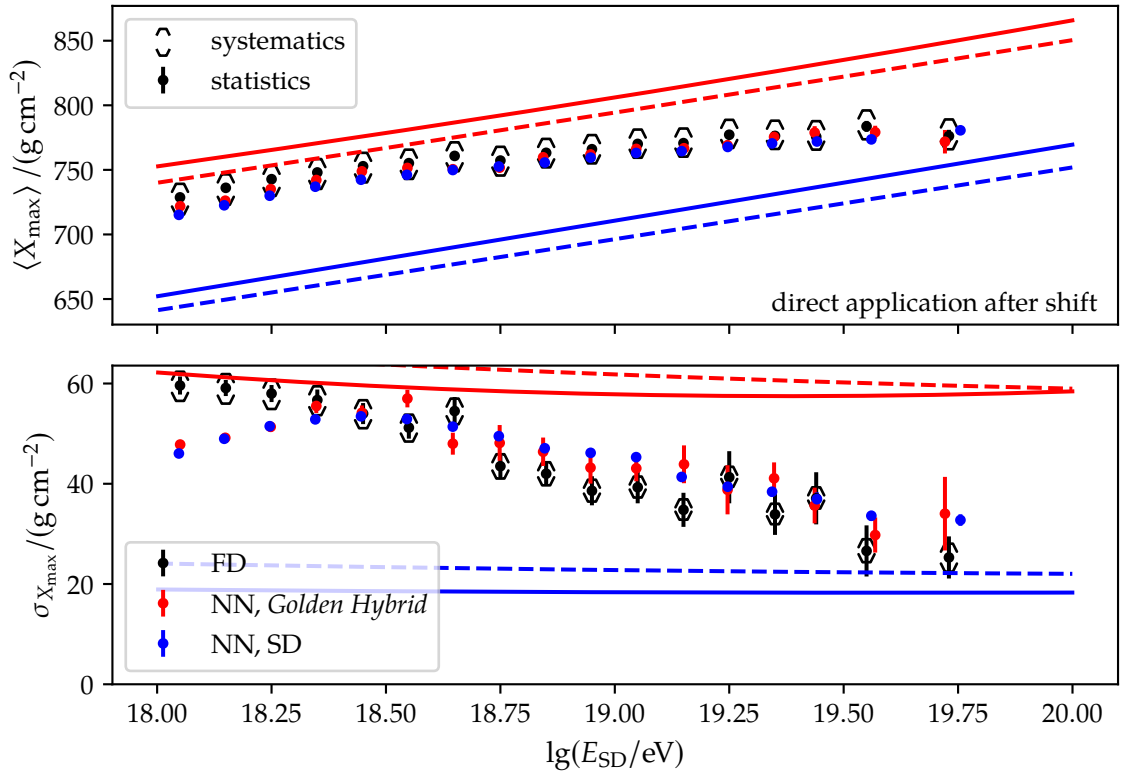


Figure 8.2: Average value of X_{max} (*top*) and standard deviation $\sigma_{X_{\mathrm{max}}}$ (*bottom*) as a function of the reconstructed energy E_{SD} for the FD measurement (black), the prediction of the NN on the *Golden Hybrid* data set (red), and the prediction of the NN on the SD data set (blue). The solid and dashed lines show the expected behavior (see Sec. 5.4.2) for air showers induced by protons (red) and irons (blue) using the hadronic interaction models EPOS-LHC and QGSJetII-04, respectively. The predictions of the NN on both data sets are very similar. Below a primary energy of $10^{18.5}$ eV, the standard deviation of the NN predictions lies below the values of the FD measurement. This under-prediction is an indicator that in this region the NN is predicting more likely the mean value.

of the network predictions is much lower than the FD measurements. We attribute this to regression towards the mean. In this region, the information in the shower footprint is not sufficient for the network to make accurate predictions falling back on predicting an average value. Hence, We discard the predictions for SD energies below a primary energy of $10^{18.5}$ eV. For higher energies, the standard deviation is notably higher than that of the FD measurements which is most likely due to missing corrections and missing quality cuts. Henceforth, we proceed with the shifted values of X_{max} to deduce further corrections.

8.1.2 Derivation of corrections

To correct for the difference between the air shower simulation libraries and measurements, we use the following procedure: For each investigated target T, we compute

$$\overline{T} = \langle T \rangle - T. \tag{8.2}$$

We denote $\overline{T}$ as the deviation from the average behavior. We test for any unphysical behaviour by analyzing the dependence of $\overline{T}$ on one observable that is independent of the value of T, e.g., the depth of the shower maximum X_{max} needs to be independent of the average a_{p}

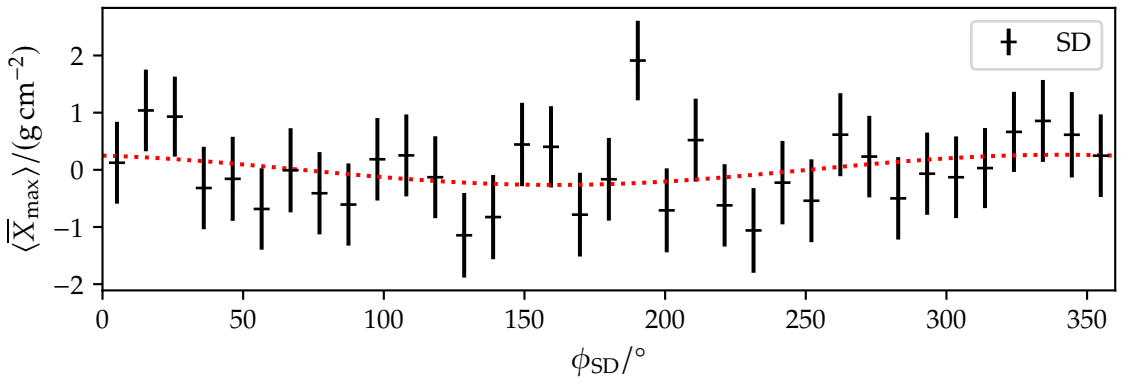


Figure 8.3: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) in bins of the reconstructed azimuth angle ϕ_{SD}. To test for a periodic dependence, we have fitted to a sine function (red, dashed). Since the amplitude is well below $1\,\mathrm{g/cm^2}$.

in an event. To visualize this process, we compute the average value of $\overline{T}$ as a function of B:

$$\langle \overline{T} \rangle (B) = \langle \langle T \rangle - T \rangle (B). \tag{8.3}$$

The quantity $\langle \overline{T} \rangle (B)$ is the deviation from the global average at B. In the following, we drop the argument B for better legibility. In each test, we use the SD data set. We use only events with a reconstructed energy above $10^{18.5}$ eV and reconstructed zenith angle below 60°. Each fit is performed on the entire data set and not on the binned values.

If possible, we compare the result on the SD data set to the result on the *Golden Hybrid* data set. In this case, we do not compute Eq. (8.3) for the *Golden Hybrid* events, we use

$$\overline{T} = \Delta T = T_{\mathrm{FD}} - T_{\mathrm{NN}}. \tag{8.4}$$

A Corrections for the SD geometry

Until now, we have assumed that we work with a planar, ideal detector and that there are no effects that would break the symmetries used for the shower footprint standardization (see Sec. 5.3.3). In this section, we test the validity of this assumption.

The magnetic field breaks the rotational symmetry of the triangular grid of the SD. Therefore, we have to check for the dependence of any observable on the azimuth angle ϕ_{SD} in our predictions (see Fig. 8.3). We test for periodic behavior using a sine fit function (see Eq. (A.11)). The amplitude of the sine-function is below $1\,\mathrm{g/cm^2}$ (see Table D.2). Therefore, we do not correct for this behaviour.

In Sec. 7.1.5.B, we showed that the most essential part of the shower footprint lies in the region of the hottest station, denoted as C_0, and the first crown C_1. In simulations, all PMTs of the WCD stations work. This is not the case in real data (see Fig. D.36). There are no events in which all PMTs in C_0 and C_1 in all stations operate at the same time. In Fig. 8.4, we demonstrate that not working PMTs have an impact on our predictions. Fewer working PMTs correlate with larger predictions of X_{max}. However, for more than 80% of working PMTs, which is the case for most events, the deviation is relatively small (see Fig. D.36). Therefore, in the region of interest, we find no strong dependence of the X_{max} prediction on the number of working PMTs of C_0 and C_1.

In the simulation study, we did not account for the individual height of the SD detectors above sea level. The real SD array, however, has a slight tilt. Due to attenuation, however, the signal distribution depends on the height of our detectors. Since the network also uses

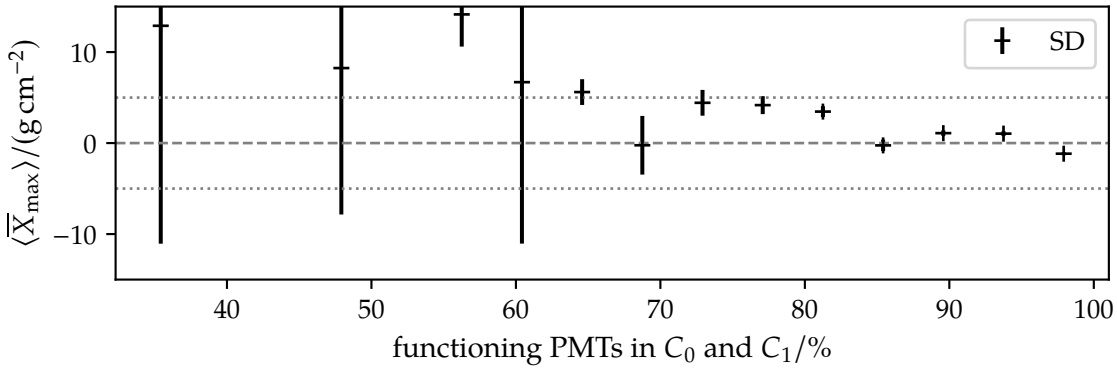


Figure 8.4: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) in bins of the fraction of working PMTs in the hottest station and the first crown, denoted as C_0 and C_1 respectively. The horizontal gray lines mark $5\,\mathrm{g/cm^2}$ (dotted), $0\,\mathrm{g/cm^2}$ (dashed), and $-5\,\mathrm{g/cm^2}$ (dotted). In the region of the main bulk of data, above 80%, we do not find noticeable deviations (see Fig. D.36).

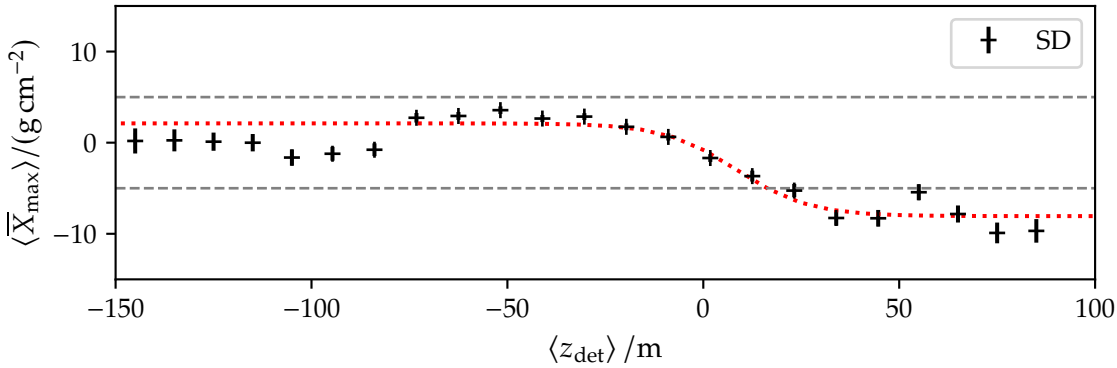


Figure 8.5: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) in bins of the average detector height $\langle z_{\mathrm{det}} \rangle$. The horizontal gray lines mark $5\,\mathrm{g/cm^2}$ (dashed) and $-5\,\mathrm{g/cm^2}$ (dashed). We find a step-like transition between high and low $\langle z_{\mathrm{det}} \rangle$. This behavior is well approximated by a tanh-fit function (red, dotted).

the signal distribution to predict the targets, we test for this effect by computing the average height of all participating detectors of an event $\langle z_{\mathrm{det}} \rangle$. In Fig. 8.5, we depict $\langle \overline{X}_{\mathrm{max}} \rangle$ as a function of $\langle z_{\mathrm{det}} \rangle$. We find a step-like dependence on the average detector height. For detectors at higher altitudes, X_{max} is under-predicted. For lower $\langle z_{\mathrm{det}} \rangle$, on the other hand, X_{max} is slightly over-predicted. We correct for this step-like deviation using a tanh-like function in Eq. (A.12).

The lower average predictions for X_{max} could arise from the fact, that the shower is more attenuated at stations at lower height above sea level, and thus the distance in slant depth of the detector the shower maximum is increased. However, the effect could also arise from the tilted geometry of the SD array. Only few events of the whole data set suffer from this effect. Hence, although, we could correct by using another step function, we refrain from doing it.

B Corrections due to detector aging

The signal shape of the WCD stations changes over time due to aging effects (see Sec. 5.2). The most simple shape parameter, which measures this change, is the so called area over

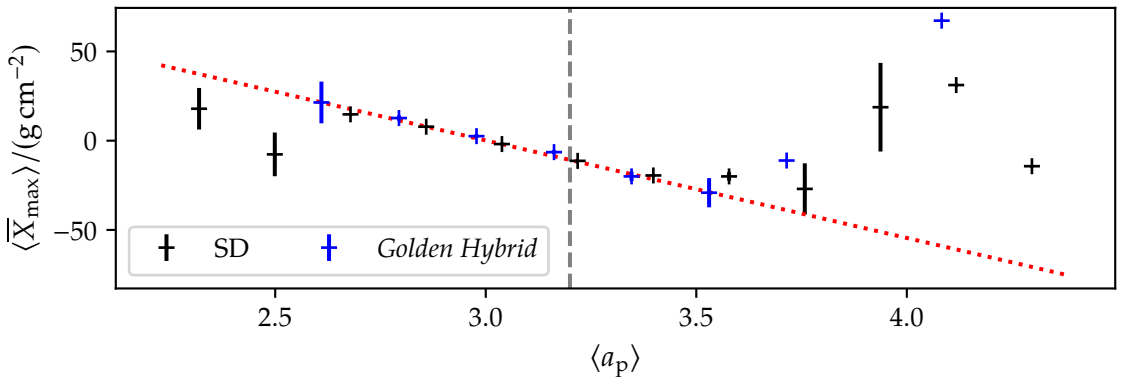


Figure 8.6: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) in bins of the average area over peak a_{p} for events in the SD (black) and *Golden Hybrid* (blue) data set. The vertical line (gray, dashed) marks the a_{p} used in our simulation library. In the region containing the most data, we find a linear relationship between the deviation and the average area over peak. We have fitted a linear function (red dashed) to correct for it.

peak a_{p}. Since each PMTs in an event has a unique value, we calculate the average value of all area over peak values $\langle a_{\mathrm{p}} \rangle$ and test for dependencies (see Fig. 8.6). We find a linear relationship between $\langle a_{\mathrm{p}} \rangle$ and $\langle \overline{X}_{\mathrm{max}} \rangle$ in the region containing over 95% of the data (see Fig. D.38) for the SD and the *Golden Hybrid* data set. To correct this behavior, we fit a linear function to $\overline{X}_{\mathrm{max}} \rangle$ as a function of a_{p} (see Eq. (A.10)). The values outside of the a_{p} interval $[2.7, 3.5]$ show substantial deviations from this linear relationship. Compared to Fig. D.38, this deviation is mainly caused by the low statistics. Since the SD and *Golden Hybrid* results agree, we disregard the regions in a_{p} without obvious linear behavior.

In the following, we show that the a_{p} correction removes the dependence on aging sufficiently. We compute the time an event has been detected by converting the Global Positioning System (GPS) time stamps taken from Offline to Universal Time Coordinated (UTC) time. Ignoring the difference in leap seconds[2], we use

$$\mathrm{UTC} = \mathrm{UTC}(1980, 1, 6) + \mathrm{GPS} \tag{8.5}$$

to perform the conversion. We find that the a_{p} correction removes the non-linear dependence on the time of detection almost completely (see Fig. 8.7).

C Corrections due to atmospheric conditions

All of the CORSIKA-based air shower simulation libraries used for the training of the NNs are simulated under the same atmospheric conditions. However, the temperature T, pressure P, and the air density ρ at the observatory are not constant. They depend on the local weather conditions and the climate in the Pampa Amarilla.

The atmospheric parameters depend on the day-night cycle (see Fig. 8.8) and the time of the year (see Fig. 8.9). In both cases, there is a periodic dependence of $\langle \overline{X}_{\mathrm{max}} \rangle$ on the time of the day and the day of the year, respectively. The effect of the day-night cycle is smaller than that of the seasonal variations. By fitting a sine function (see Eq. (A.11)) to the day-night cycle, we find an amplitude of $1\,\mathrm{g/cm^2}$. If we do the same to the seasonal variations, we obtain an amplitude of about $2\,\mathrm{g/cm^2}$. To generate the plots, we used the UTC time of each event (see Eq. (8.5)), ignoring the time zone. For the day-night cycle, we removed the date.

[2] Until 2020 there is only a difference of a couple of seconds.

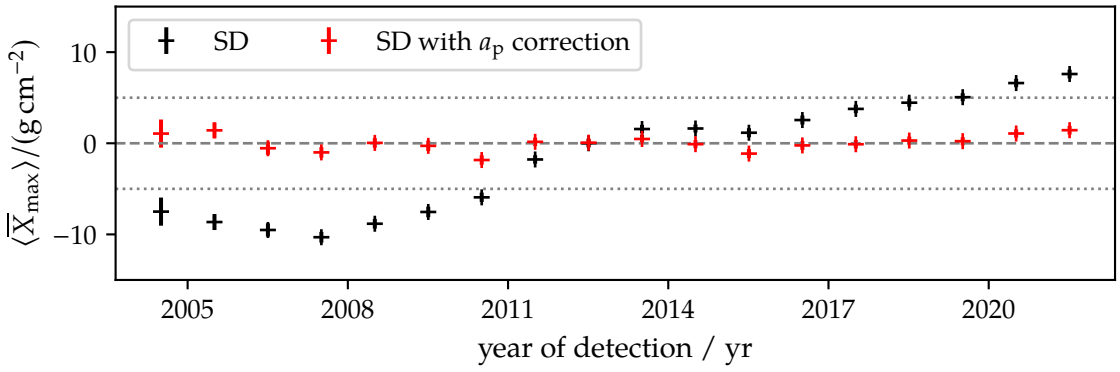


Figure 8.7: Deviation from average behavior $\langle \overline{X}_{max} \rangle$ (see Eq. (8.2)) in bins of the time of detection for the raw events (black) and events that have been corrected using the area over peak correction from Fig. 8.6 (red). The horizontal gray lines mark $5\,g/cm^2$ (dotted), $0\,g/cm^2$ (dashed), and $-5\,g/cm^2$ (dotted). After correcting for the area over peak dependence, the SD data set shows no dependence on the time of detection anymore.

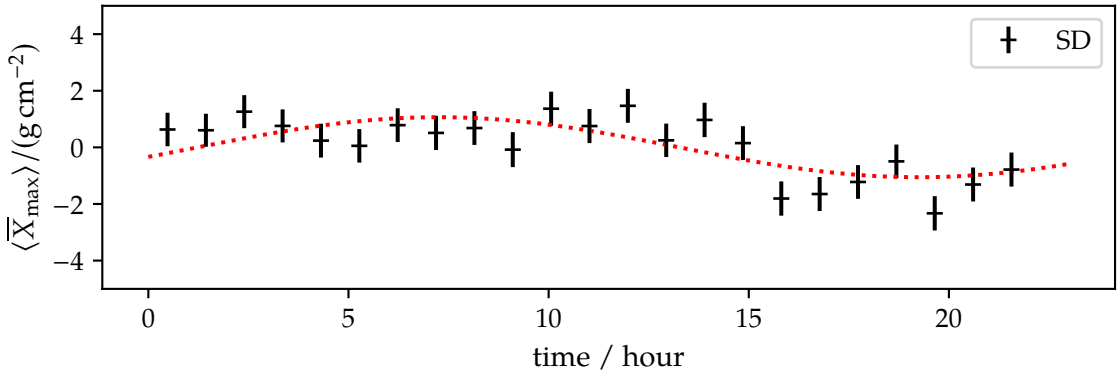


Figure 8.8: Deviation from average behavior $\langle \overline{X}_{max} \rangle$ (see Eq. (8.2)) as a function of the time of a day. We fitted a sine function (dotted, red) to the underlying data. We obtain an amplitude of $1\,g/cm^2$. Note that during calculation of the 'time of day' we ignored the time zone.

For the seasonal variations, we removed the year of the UTC time. Hence, in both cases, we have averaged over many different effects such as the changing lengths of the days and the aging of the detectors. Therefore, we account for these corrections by adding them to the systematic uncertainties instead of subtracting the parametrization.

Instead of analyzing the overall change of all atmospheric parameters, we also investigate the dependence on each one of the atmospheric parameters. We find no obvious functional dependence for any of the observables on the temperature T (see Fig. D.39) and the air density ρ (see Fig. D.40). In both cases, we obtain a result that is in agreement with zero in the regions containing a majority of the data (see Fig. 5.5). Only for extreme values of the pressure or the temperature, we experience slight deviations from the mean.

For the atmospheric pressure, $\langle \overline{X}_{max} \rangle$ shows non-linear dependence (see Fig. 8.10). At low atmospheric pressure, the NN predicts larger than average values of X_{max}. To ensure that this is not a natural effect due to the change in overall air density, we added ΔX_{max} from the *Golden Hybrid* data set to the analysis. Both data sets show similar behavior. This indicates that this is an effect we have to correct for. We use a quadratic function (see Eq. (A.10)) to

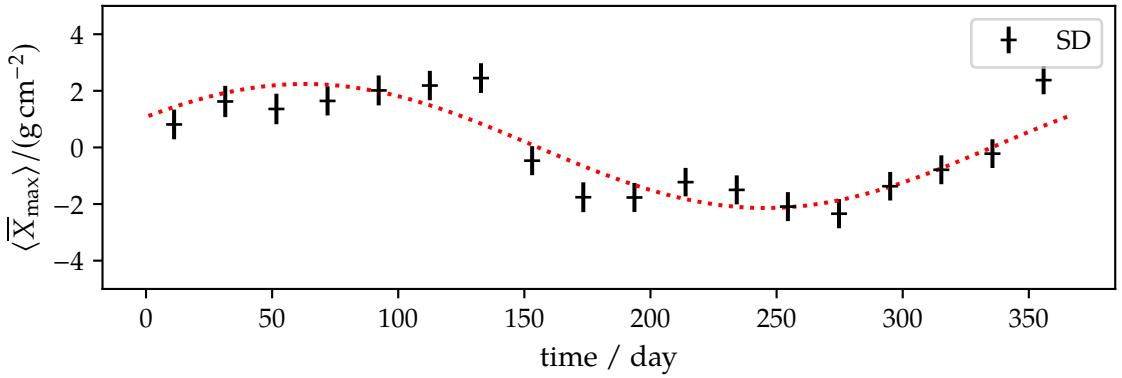


Figure 8.9: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) as a function of the day of a year. We fitted a sine function (dotted, red) to the underlying data. We obtain an amplitude of $2\,\mathrm{g/cm^2}$.

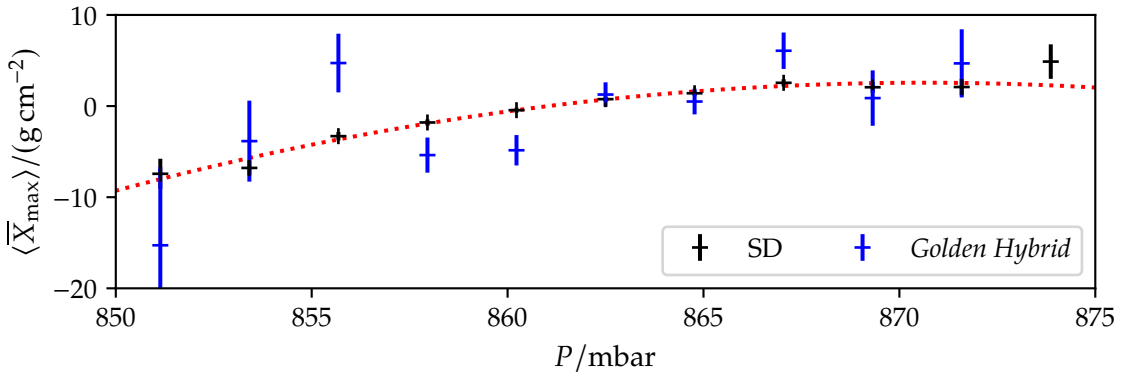


Figure 8.10: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) as a function of the pressure P (black, pluses). We added ΔX_{max} from the *Golden Hybrid* data set (blue, pluses) to show that this is an effect we have to correct for. We have fitted a quadratic function to the underlying data (red, dotted). It reproduced the average behavior very well.

interpolate the dependence.

The higher the zenith angle of a shower, the higher the attenuation of the electromagnetic sub-component. In the air shower libraries, a model has been used to account for this shower inclination. However, this model does not portray the atmosphere in Malargüe correctly. In Fig. 8.11, we see a substantial deviation at high zenith angles. Again, the predictions on *Golden Hybrid* events show similar behavior. We model this effect using an exponential function (see Eq. (A.13)) and correct by usual means.

D Application of corrections

In total, we correct for the dependencies in $\langle z_{\mathrm{det}} \rangle$, $\langle a_{\mathrm{p}} \rangle$, P, and $\sin^2\theta$. After applying all corrections, we analyze the bias of our predictions on *Golden Hybrid* data. We find a bias of about $30.5\,\mathrm{g/cm^2}$, which is only marginally larger than that found previously in Sec. 8.1.1. In addition, we have to account for the difference between the phase space of our SD and *Golden Hybrid* data set. We compare the average value of the predictions on both data sets. We obtain a difference of $2.22\,\mathrm{g/cm^2}$.

We again shift the corrected predictions using the calibration shift and the shift from the

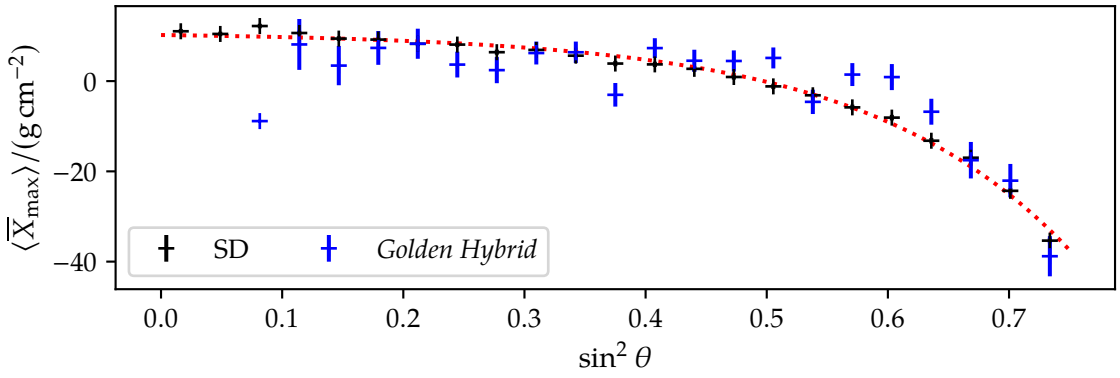


Figure 8.11: Deviation from average behavior $\langle \overline{X}_{\mathrm{max}} \rangle$ (see Eq. (8.2)) as a function of the zenith angle θ in terms of $\sin^2$. We added ΔX_{max} from the *Golden Hybrid* data set (blue, pluses) to show that this is an effect we have to correct for. We have fitted an exponential function to the underlying data (red, dotted).

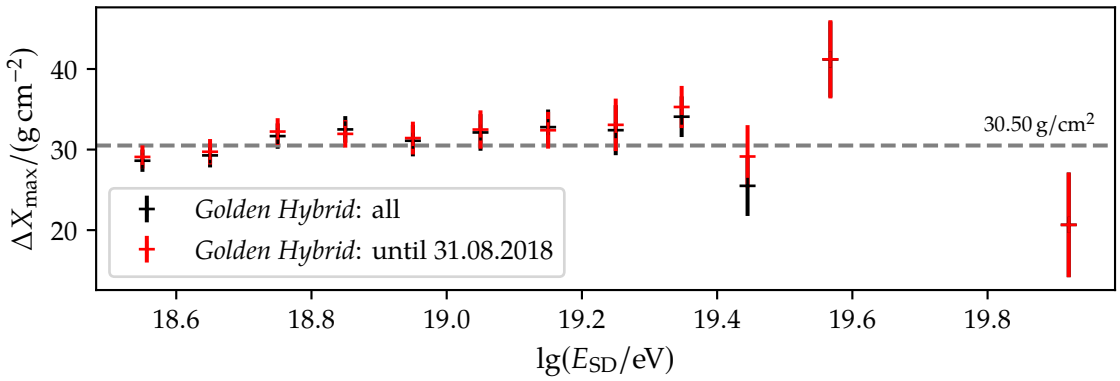


Figure 8.12: Difference ΔX_{max} of corrected NN prediction and X_{max} measurement by FD as a function of the reconstructed primary energy E_{SD}. The dashed gray line marks the average over all differences. The black crosses mark the differences for our entire *Golden Hybrid* data set and the red crosses the difference for all data taken before 31.08.2018.

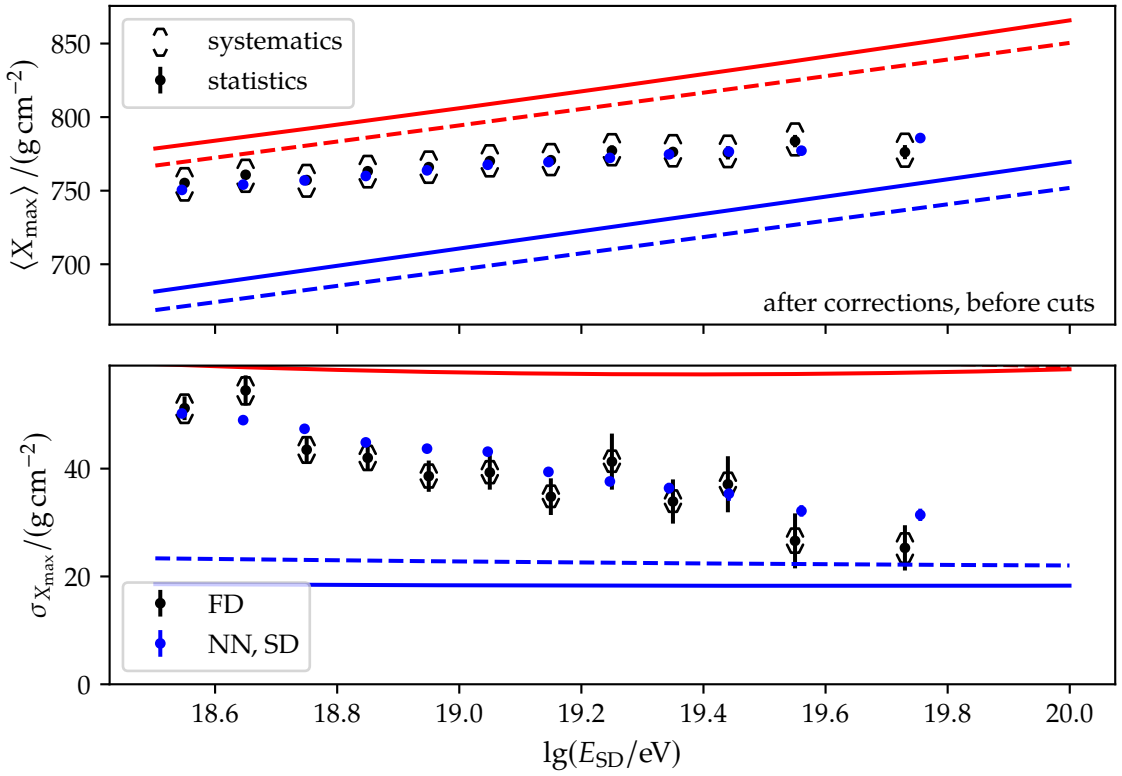


Figure 8.13: Average values of X_{max} (*top*) and standard deviation $\sigma_{X_{\mathrm{max}}}$ (*bottom*) as a function of the reconstructed primary energy E_{SD} for the FD measurement (black) and the corrected predictions of the NN on the SD data set (blue). The solid and dashed line show the expected behavior (see Sec. 5.4.2) for air showers induced by protons (red) and irons (blue) using the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively.

phase-space-mismatch (see Fig. 8.13). The first moment, $\langle X_{\mathrm{max}} \rangle$ agrees well with the prediction of the NN on the SD data set. The second moment, as predicted by the NN, still lies above that predictions of the FD.

8.1.3 Correction for relative muon content and logarithmic mass number

We follow the same procedure as discussed in Sec. 8.1.2 to find the corrections for the relative muon content R_{μ} and logarithmic mass number $\ln A$ prediction of the NNs. Since there is no direct measurement of any of the observables, we solely rely on Eq. (8.3) used on SD predictions and cannot cross-check any of the results with the *Golden Hybrid* data set. To shorten this section, we only discuss notable differences in the corrections of X_{max}.

Relative muon content We find a similar behavior for the relative muon content R_{μ} as observed for X_{max}. We have attached the figures for all corrections that show no difference from the X_{max} counterpart in Figs. D.41 to D.50. The y-axes are flipped due to the missing global pre-calibration shift in Sec. 8.1.1.

To correct R_{μ}, we use an additional correction (see Fig. 8.14). The deviation $\langle \overline{R}_{\mu} \rangle$ shows a dependence on the number of active PMTs in C_0 and C_1. We model the correction with an exponential fit function (see Eq. (A.13)).

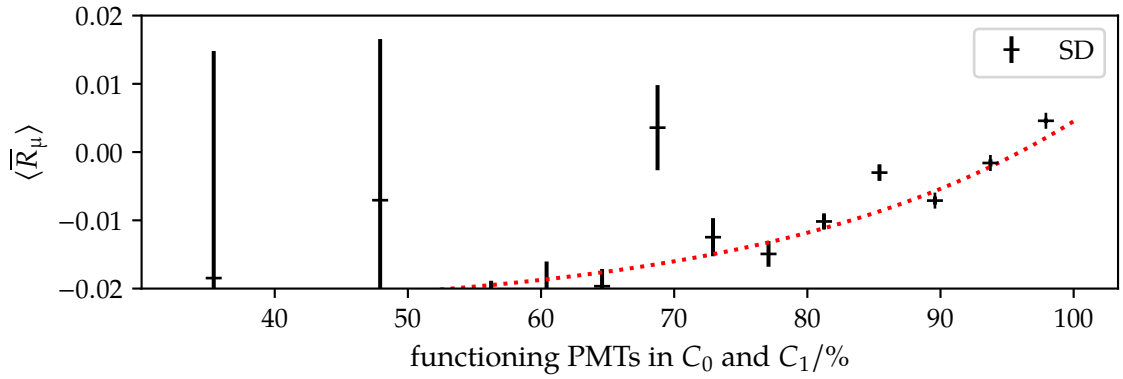


Figure 8.14: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ (see Eq. (8.2)) as a function of the fraction of working PMTs in the HS and the first crown, denoted as C_0 and C_1 respectively. This plot is equivalent to that in Fig. 8.4. For lower fractions the relative muon number R_μ is on average over-predicted. We model the correction with an exponential fit function (dotted, red).

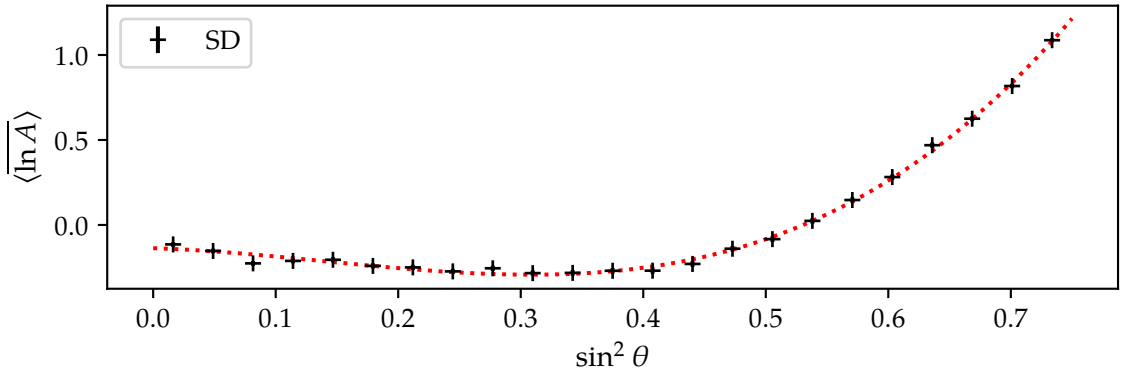


Figure 8.15: Dependence of the average predicted values of $\ln A$ on the zenith angle, θ (cf. Fig. 8.11). The red, dashed line is a best fit of a third-order polynomial fitted to the underlying data.

Logarithmic mass number Except for the zenith correction shown in Fig. 8.15, all corrections exhibit similar behavior to that found in Figs. D.51 to D.60. For the reconstructed values of $\ln A$, the zenith dependence is parameterized by a cubic polynomial (see Eq. (A.10)).

8.2 Estimation of mass composition of UHECRs

In this section, we use the corrected predictions derived in Sec. 8.1 to estimate the mass composition of UHECRs on a high-quality subset of the SD data set.

8.2.1 Quality selection for the SD data set

We base most of the quality cuts on regions of the phase space that lie outside of normalcy. If the Auger observatory works under extreme conditions, e.g., at too low or high temperatures or for average area-over-peak values outside of the core region, it is not clear how the signal distribution on the ground level is influenced. Most of the time, these regions coincide with regions of low statistics, which makes it impossible to account for the effects. We have

Table 8.1: Overview of the quality cuts on the SD data set. The first column is an identifier and the second column the name of the cut. In the third column is the number of remaining events in the data set. The fourth column shows the percentage of remaining events after the cut and the final column the percentage of remaining events if compared to a reference column. We reset the total loss percentage after the second row due to the severity of the energy cut. The quality cuts after the energy cut amount to a loss of 15% of the SD data set.

		size of data set	last cut/%	remaining events/%
a	SD data set	1935850	100	100
b	energy	169960	8.78	8.78
c	area over peak	164337	96.69	96.69
d	fraction pmts in C_0 and C_1	159177	96.72	96.07
e	height of detectors	157174	98.74	92.48
f	pressure	149567	95.16	88.00
g	density	145712	97.42	85.73
h	temperature	144528	99.19	85.04

tabulated the effect of all cuts in the order of appearance in Table 8.1.

We remove the events with an average area over peak value not inside the interval [2.75, 3.45] (see Row 8.1.c). We motivate this with the result in Fig. 8.6. Outside the chosen interval, the linear relationship does not hold. The interval coincides with that found in [P:104]. We only keep events where at least 80% of the PMTs inside the first crown are working (see Row 8.1.d). By doing this, we assume to obtain an accurate estimate of the signal traces. We remove events that exhibit very large or very low average detector heights (see Row 8.1.e). All events below −150 m or above 100 m are cut away. To ensure stable working conditions, we also cut the tails of the distribution of the atmospheric parameters (see Rows 8.1.f to 8.1.h). We cut away all events outside of the 2σ regions of the original distributions of pressure P, air density ρ, and temperature T (see Fig. 5.5). Since the atmospheric parameters are correlated, the effectiveness of the cut is reduced with each additional cut. In addition, we remove 16 events, which showed unusual behaviour in earlier work [P:98].

In contrast to [P:104], we do not perform a fiducial cut in the energy-zenith plane. This choice leaves us twice as many events after the cuts as in the preceding analysis.

8.2.2 NN prediction of mass sensitive targets

Applying the corrections derived in Sec. 8.1.2 on the predictions of the NNs and the quality cuts on the SD data, we obtain results for the depth of the shower maximum X_{max}, the relative muon content R_μ, and the logarithmic mass number $\ln A$. In this section, we review the systematic uncertainties for each observable independently and, afterward, discuss the predictions. The average value of one of the three observables is referred to as the first moment and the standard deviation of a target is referred to as the second moment.

A Prediction of the depth of the shower maximum

At first, the contributions of systematic uncertainties to the first moment of the X_{max} distribution are analyzed. Since we have calibrated the predictions of the NN against the *Golden Hybrid* data set, there is no need to account for all of the systematic uncertainties derived in Sec. 7.3.3. Instead, the uncertainties from the X_{max} measurement of FD are used. These sum up to approximately $10\,\mathrm{g/cm^2}$, varying only slightly with the energy (see [P:53]). For simplicity, we use a constant value of $10\,\mathrm{g/cm^2}$ for the entire energy range. This value is

much smaller than the systematic uncertainty we would have obtained from Sec. 7.3.3. In addition to the uncertainty of the X_{max} measurement, the uncertainties due to the calibration procedure have to be taken into account. The first calibration constant used to adjust the predictions using the *Golden Hybrid* data set has an uncertainty of about $0.6\,\mathrm{g/cm^2}$. The second calibration constant that accounts for the different phase spaces of *Golden Hybrid* and SD has an uncertainty of about $0.8\,\mathrm{g/cm^2}$. These are combined to a conservative estimate of $2\,\mathrm{g/cm^2}$. Moreover, the amplitudes from the daily (see Fig. 8.8) and seasonal (see Fig. 8.9) variations are considered as another source of systematic uncertainty in our predictions. They add up to about $3\,\mathrm{g/cm^2}$. Summing over the absolute values, we obtain a total systematic uncertainty of $15\,\mathrm{g/cm^2}$ over the entire energy range.

For the second moment $\sigma_{X_{\mathrm{max}}}$, there is no way to directly calibrate against the FD measurements. Therefore, we use the results of Sec. 7.3.3. To account for the uncertainty due to the high-energy hadronic interaction, we use the parameterized loss of precision shown in Fig. 7.49. The uncertainty ranges from $3.2\,\mathrm{g/cm^2}$ at $10^{18.5}\,\mathrm{eV}$ to $1.6\,\mathrm{g/cm^2}$ at $10^{20.0}\,\mathrm{eV}$. Similarly, the parameterizations in the simulation study shown in Fig. 7.51 are used as asymmetric uncertainties due to the unknown composition. We obtain a constant lower uncertainty of 2.1 eV and an energy-dependent upper uncertainty going from $12.3\,\mathrm{g/cm^2}$ at $10^{18.5}\,\mathrm{eV}$ to $0.4\,\mathrm{g/cm^2}$ at $10^{20.0}\,\mathrm{eV}$. We obtain the total lower and upper uncertainty of the second moment by summing up the absolute values treating the symmetric uncertainty from the high-energy hadronic interaction as an equally valued lower and upper uncertainty.

Using the estimates for the systematic uncertainty, we can compare the first and second moments of the corrected predictions for the quality-selected events to the FD reconstruction (see Fig. 8.16). The energy bins are updated to exhibit the behavior of the first and second moments of the NN predictions at the highest energies. To ensure enough statistics, each bin contains at least 40 events. The distributions of the NN predictions as a function of the primary energy are depicted in Fig. D.61. The first moment of the NN predictions agrees well with the first moment of the FD measurement at the available energy bins. At higher primary energies, the first moment of the NN predictions implies an increasingly heavier composition. This behavior is in agreement with previous studies [P:100, P:104].

The second moment of the NN predictions also agrees with that of the FD measurements within uncertainties. However, in most of the energy bins, the second moment of the predictions of the NN is slightly higher. This difference is especially noticeable in the last two FD energy bins, where the expected values of both second moments lie over $5\,\mathrm{g/cm^2}$ apart. At the highest energies, in the last four energy bins of the NN predictions, the width of the second moment seems even to increase again. This result is not in agreement with the study conducted in [P:104] (see Fig. D.62). All of the second moments computed from the predictions of the NN selected in this work are larger than that from the NN used in [P:104]. To exclude that this is an effect of the fiducial event selection, a similar selection process is discussed in Appendix A.7, using a slightly stricter condition to find a high-quality subset in the investigated phase space. However, applying this criteria to our data set does not resolve the tension between both results (see Fig. 8.17). From the last four bins, only the highest energy bin is noticeable affected.

There are many differences between the analysis chain used in this work and that used in [P:104], which makes it hard to point out an apparent cause for this issue. Especially since the predictions of both of the NNs have shown a similar precision and primary-dependent bias in simulation studies. The difference could be caused by training on a data set simulated with the hadronic interaction model QGSJetII-04. Compared to EPOS-LHC, which is used in [P:104], QGSJetII-04 produces larger values of fluctuations for X_{max}. This could impact the training process. Another explanation could be due to the different choices of X_{max}. For air shower simulations, $\overline{\mathrm{Off}}\underline{\mathrm{line}}$ saves two different reference values for X_{max}. One of these

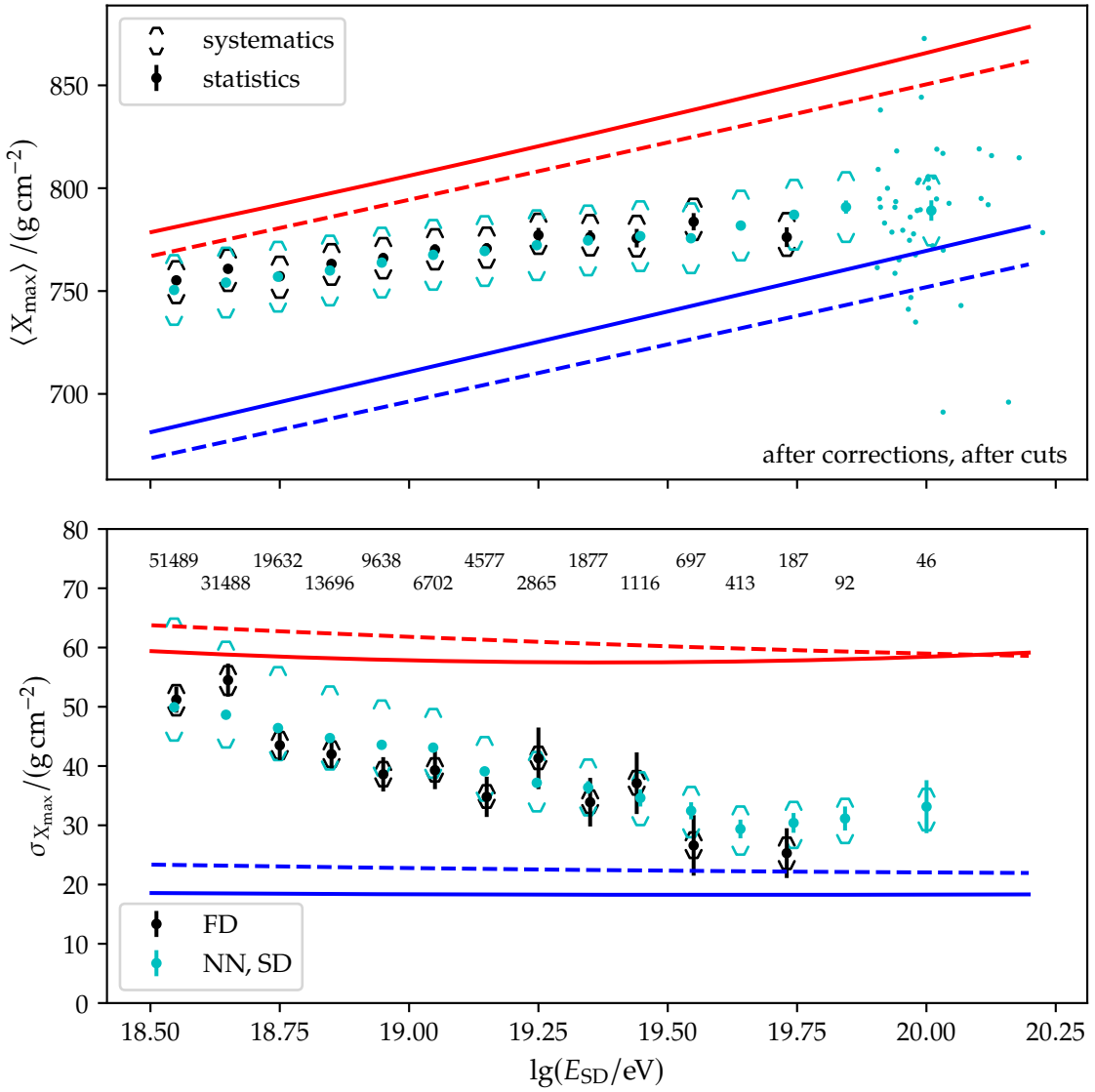


Figure 8.16: The first (*top*) and second moment (*bottom*) of X_{max} as a function of the reconstructed energy E_{SD} for the FD measurement (black) and the corrected predictions of the NN (cyan) on the quality-selected SD data set (see Table 8.1). The dashed and solid lines show the expected behavior (see Sec. 5.4.2) for air showers induced by protons (red) and irons (blue) using the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. The numbers in the *bottom* panel indicate the amount of events in each of the energy bins. The small cyan points in the *top* panel depict single events in the highest energy bin.

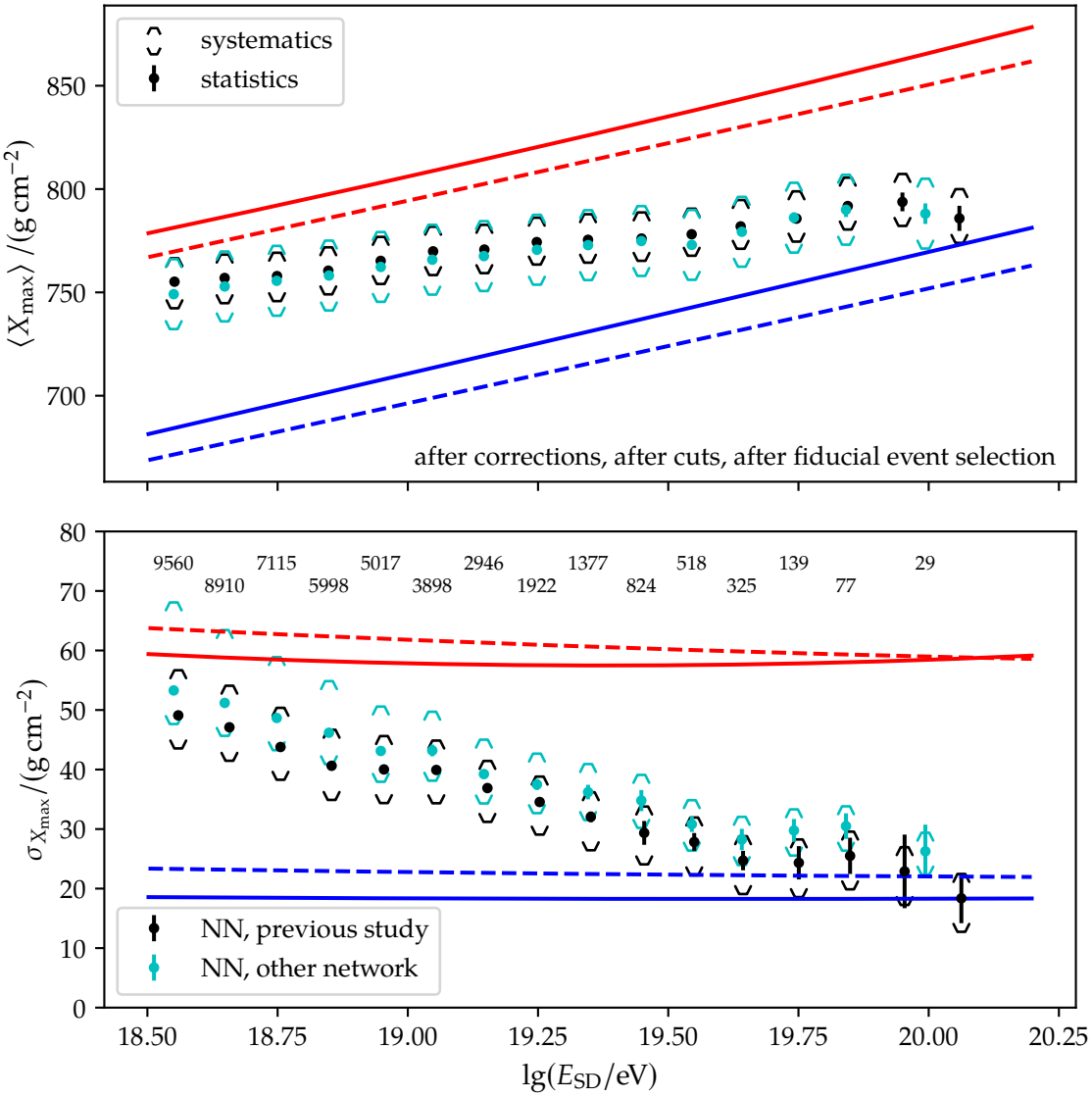


Figure 8.17: First (*top*) and second moment (*bottom*) of X_{max} as a function of the reconstructed energy E_{SD} for the NN of the study in [P:104] (black) and the corrected predictions of the NN (cyan) on the quality-selected SD data set (see Table 8.1) for events passing the fiducial event selection defined in Appendix A.7. The dashed and solid lines show the expected behavior (see Sec. 5.4.2) for air showers induced by proton (red) and iron (blue) primaries using the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. The numbers in the *bottom* panel indicate the amount of events in each of the energy bins.

X_{max} values is accessible via `GenShower.XmaxGaisserHillas`, denoted $X^{\mathrm{g}}_{\mathrm{max}}$, which is taken directly from the CORSIKA files. The other one, `GenShower.XmaxInterpolated`, denoted $X^{\mathrm{i}}_{\mathrm{max}}$, is computed in Offline. In this work, the former depth of the shower maximum was used as true value in the simulations. However, in [P:104], a mix of both X_{max} values has been used. Switching from $X^{\mathrm{g}}_{\mathrm{max}}$ to $X^{\mathrm{i}}_{\mathrm{max}}$ to evaluate the NN predictions reduces the bias of all primaries without changing the precision noticeably (see Fig. D.63 and Fig. D.65). Moreover, the discrepancy could be caused by the transition from simulation to measurements. In Fig. D.64 and Fig. D.66, the precision and bias for a NN trained on the setup described in Row 7.1.i are discussed. The network was disregarded in favor of other networks due to its sizeable proton-iron bias at the highest energies (see Sec. 7.2). Even though the predictions of the disregarded NN are worse than the predictions of the – for this study – selected NN on simulations, the second moment of the disregarded NN aligns much better with the second moment of the NN used in the preceding study for energy values above $10^{18.8}$ eV (see Fig. 8.18). To resolve the discrepancy in the second moment, more studies are required.

Even though there is a tension in the second moment, the first moments of both NN-based studies agree well, with and without the fiducial event selection (see Figs. 8.17 and D.62). Therefore, the NN-based approach discussed in this thesis can be considered equivalent to the previous approach. Since both NN approaches are fundamentally different, this allows for further cross-checks of the predictions of both methods. Furthermore, the much faster training time of the NNs setups in this thesis could also benefit the slower setup in [P:104], making hyperparameter searches more viable.

B Prediction of the relative muon content

In this section, the systematic uncertainties for the R_{μ} predictions are estimated. Then, results on reconstructed values for R_{μ} from the NN are discussed.

R_{μ} prediction of the NN cannot be calibrated, since there is no direct measurement over the entire energy range. Therefore, the results of the simulation study described in Sec. 7.3.3 are used to account for biases in the NN-based approach. We obtain a systematic uncertainty of 0.03 over the entire energy range (see Row D.1.g) due to the high-energy hadronic interaction (see Fig. D.19). Similarly, we treat the bias of proton and iron events in Fig. D.21 as the lower and upper uncertainties due to the unknown composition, respectively. Both are energy-dependent. The lower uncertainty ranges from 0.07 at $10^{18.5}$ eV to 0.01 at $10^{20.0}$ eV. The upper uncertainty ranges from 0.08 at $10^{18.5}$ eV to 0.03 at $10^{20.0}$ eV. In addition, we conservatively estimate the uncertainty due to daily and seasonal variation as 0.01. The same procedure as in Sec. 8.2.2.A is applied to estimate the systematic uncertainties of the second moment of the R_{μ} predictions. We obtain a symmetric, energy-dependent uncertainty for the high-energy interaction models, which goes from 0.016 at $10^{18.5}$ eV to 0.003 at $10^{20.0}$ eV, and an asymmetric uncertainty for the unknown composition. The lower uncertainty has a constant value of 0.02 over the entire energy range. The upper uncertainty ranges from 0.056 at $10^{18.5}$ eV to 0.019 at $10^{20.0}$ eV. In both cases, the total systematic uncertainties are computed in accordance to Sec. 8.2.2.A .

We present the first and second moments of corrected predictions of the quality selected data set (see Table 8.1) in Fig. 8.19. The distributions of R_{μ} for the individual events in the corresponding energy bins are depicted in Fig. D.68. The first moment of the predictions is in conflict with previous studies, e.g., the muon deficit observed in very inclined air showers (see Fig. 2.7). Even though the R_{μ} used for training is not entirely comparable to the R_{μ} in studies like [P:50], commonly, the relative muon contents are far above the predictions for iron of the hadronic interaction models. Hence, it is possible that due to the deficit of muons in simulations, the first moment of the predictions could be higher than the pure iron reference, represented by the dashed blue line. However, this is only the case for energies

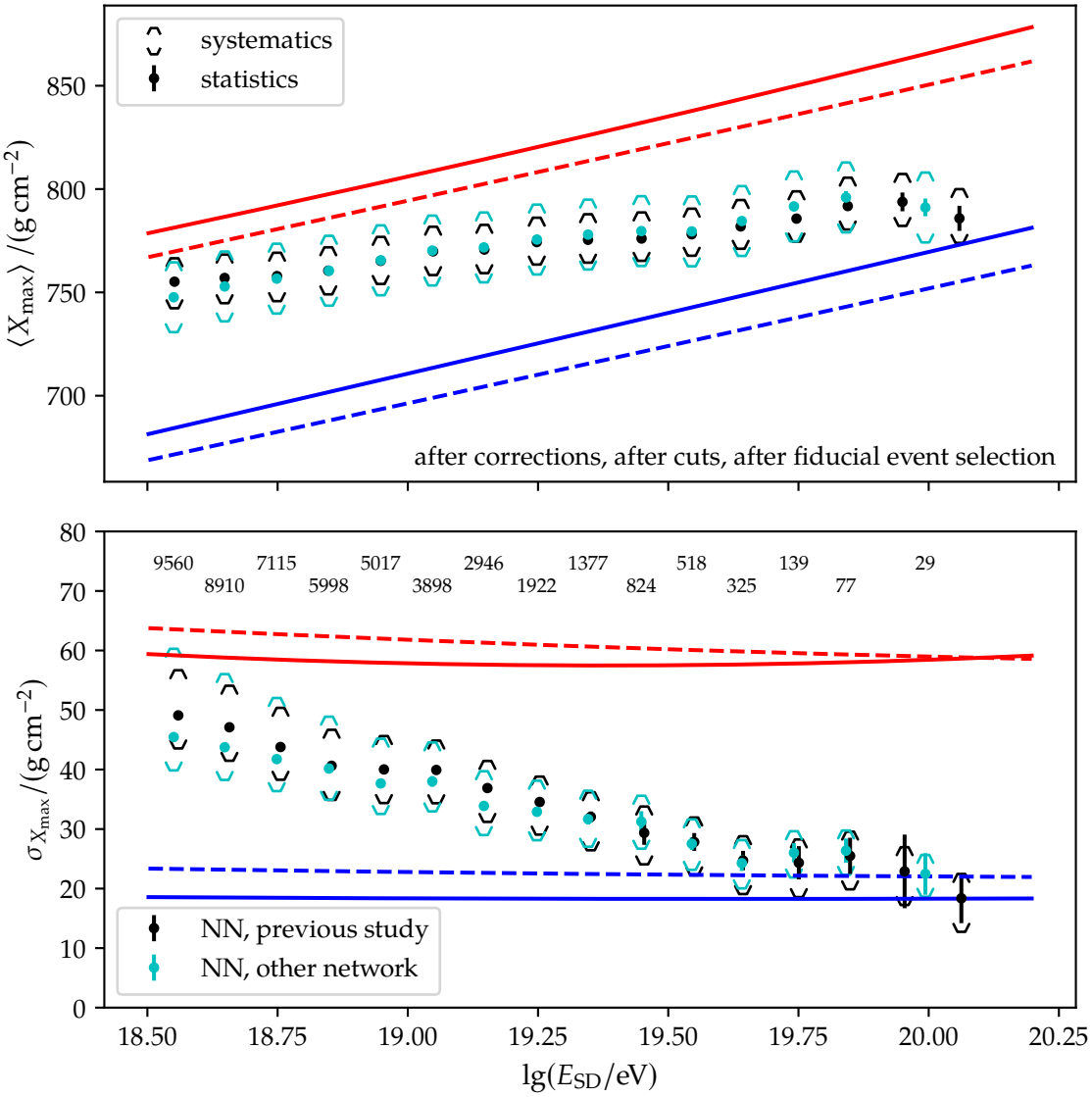


Figure 8.18: The first (*top*) and second moment (*bottom*) of X_{max} as a function of the reconstructed energy E_{SD} for the NN of the study in [P:104] (black) and the corrected predictions of a different NN (cyan) on the quality-selected SD data set (see Table 8.1) for events passing the fiducial event selection defined in Appendix A.7. The dashed and solid lines show the expected behavior (see Sec. 5.4.2) for air showers induced by protons (red) and irons (blue) using the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively. The numbers in the *bottom* panel indicate the amount of events in each of the energy bins.

above $10^{19.5}$ eV. Primarily, this is attributed to the 'averaging-towards-the-mean' behavior of the NN shown in Sec. 7.2.5. A potential cause of this lies in the data set used for training. High R_μ values are only found in few of the simulated air shower events. Only a fraction of about 30% of the iron events exhibits R_μ values above 1.3. The NN resists to predict values of R_μ outside of its known phase space to optimize the loss in the training process. This reasoning is supported by the behaviour depicted in Fig. D.67. Even at the highest energies, the network underestimates the relative muon content. Another reason for this behaviour could be the choice of reference for R_μ. By using a value directly taken from the CORSIKA files, it might be harder to relate the information of a simulated shower footprint to the muon content. In preceding investigations, R_μ has been determined by simulated detector signals. For subsequent studies, we propose the same approach to cross-check this result.

Even though the absolute value of the first moment seems to be too small, the overall trend of the first moment is in agreement with the result found in Sec. 8.2.2.A. The first moment of R_μ implies a trend towards a heavier composition with increasing energy. The second moment shows a sudden change above $10^{19.5}$ eV, increasing again, which could indicate a transition from a pure composition to a mixed composition at the highest energies. This, however, is speculative. More studies are needed to draw a final conclusion.

The results of this section imply that a new approach might be necessary to find NNs that can predict the muon content on measurements. A good starting point for a feasibility study is the new generation of physics informed NN [P:108] which follow not solely a naive data-driven approach but also encode physics information in the training setup.

C Direct prediction of logarithmic mass number

Similarly to the R_μ predictions in the previous section, there is no direct way to calibrate the $\ln A$ predictions over the entire energy range. Therefore, we account for the systematic uncertainties due to the high-energy interaction and unknown composition like in Sec. 8.2.2.B. This time, we only list the contributions. For the first moment, we obtain a symmetric uncertainty of 0.48 due to the hadronic interaction and an asymmetric, energy-dependent uncertainty due to the unknown composition, which is 0.82 and 0.97 at $10^{18.5}$ eV and 0.73 and 0.20 at $10^{20.0}$ eV for the lower and upper uncertainty, respectively. We do not add any contribution from the daily and seasonal variations due to their smallness. For the second moment of the $\ln A$ prediction, we obtain a symmetric uncertainty due to the hadronic interaction, which is 0.20 at $10^{18.5}$ eV and 0.02 at $10^{20.0}$ eV, and an asymmetric one to account for the unknown composition. The lower uncertainty is 1.12 over the whole energy range, and the upper uncertainty goes from 0.38 at $10^{18.5}$ eV to 0.09 at $10^{20.0}$ eV.

From the $X_{\max}$ measurements of FD, we estimate the first moment of $\ln A$ using the hadronic interaction model QGSJetII-04. We use this estimate to cross-test the direct NN prediction of $\ln A$. Due to the muon deficit, we expect an overestimation of $\ln A$ in data. We test this proposal by subtracting the value of the first moments in the energy interval of decadic logarithmic energies between $[18.7, 18.8]$ of the $\ln A$ derived from the $X_{\max}$ measurement from the raw NN predictions. The resulting discrepancy amounts to $\Delta \ln A = 2.27$. To compare the first moments of both methods, we subtract this difference from the raw predictions of the NN (see Fig. 8.20)). In Fig. D.69, we have depicted the distributions corresponding to the 15 bins of the raw $\ln A$ predictions. The shifted estimates of the first moment agree with that of FD within the uncertainties of both methods. Beyond the energy bins of the FD estimate, the first moment indicates a further increase in the average primary particle masses of CRs, supporting the findings presented in [P:100]. At the highest energies, we obtain an average logarithmic mass number of about 2.0 for the shifted predictions of the NN on the hadronic interaction model QGSJetII-04. The systematic uncertainty of the second moment is very large, thus making the interpretation uncertain. Still, if only the measured values are

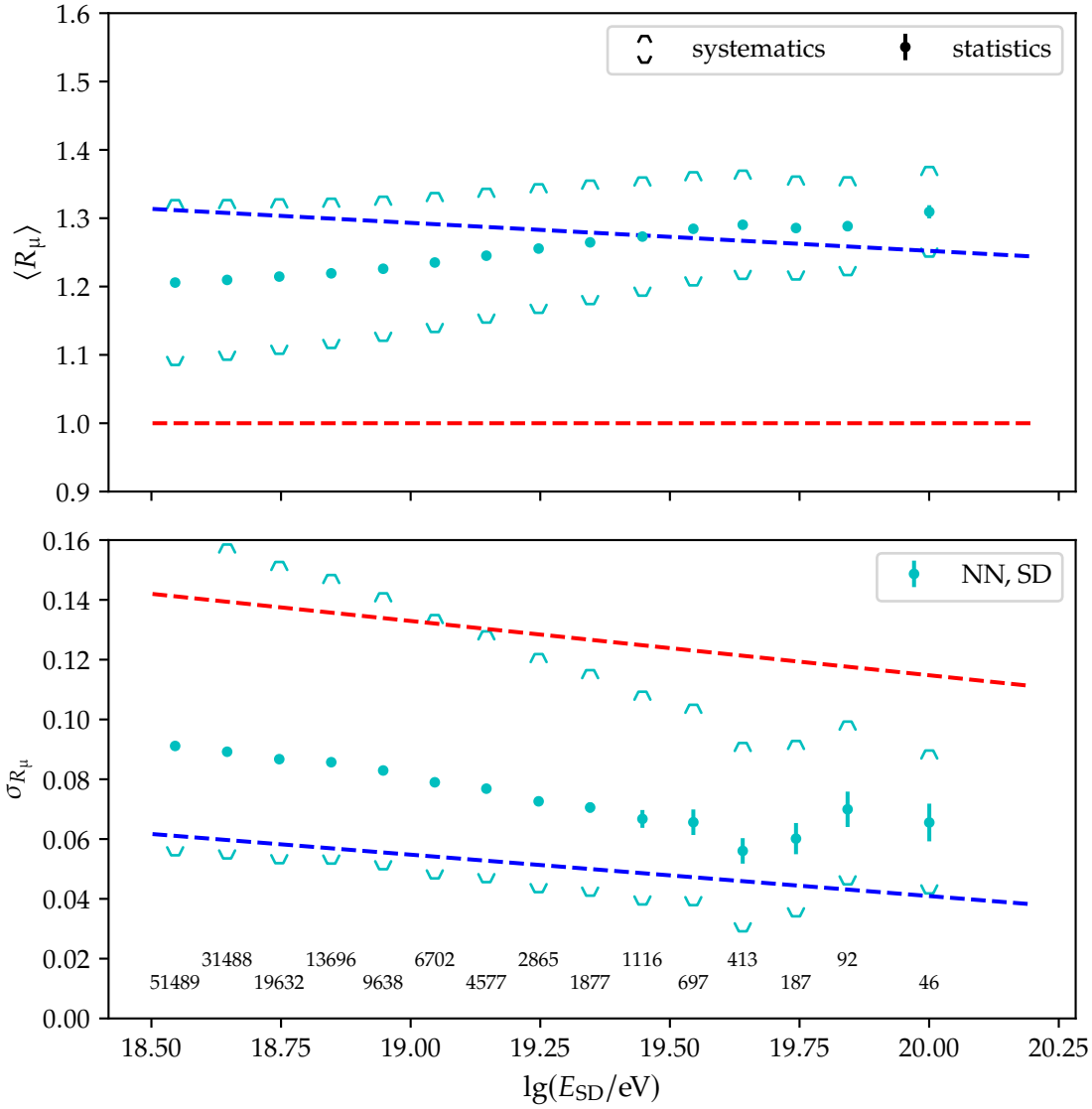


Figure 8.19: First (*top*) and second moment (*bottom*) of R_μ as a function of the reconstructed energy E_{SD} for the corrected predictions of the NN on the quality-selected SD data set (cyan, see Table 8.1). The dashed lines show the expected first and second moment (see Sec. 5.4.2) for air showers induced by protons (red) and irons (blue) using the hadronic interaction model QGSJetII-04, respectively. The numbers in the *bottom* panel indicate the amount of events in each of the energy bins.

considered, we are able to check for trends. Until an energy of $10^{19.7}$ eV, the second moment is continuously decreasing. If the first moment is taken into consideration, this trend supports a change towards a more pure composition comprised of more massive particles. For energies above $10^{19.7}$ eV, however, there is a sudden increase that coincides with the plateau in the first moment. This flattening implies a constant composition at the highest energies and is similar to the results in Sec. 8.2.2.B.

Using the reconstructed values of X_{max} from Sec. 8.2.2.A and R_{μ} from Sec. 8.2.2.B, we can compute additional estimates for $\ln A$ with the methods described in Sec. 7.2.5. We use R_{μ}, assuming that the predictions are suffering from the muon deficit in the form of a global shift by an unknown value. We obtain four different estimates for the logarithmic mass number (see Fig. 8.21). The absolute values of the estimates do not agree with each other. To test if the differences between the direct $\ln A$ predictions and the indirect estimations are due to a constant shift, we compute the differences between the NN-based direct prediction and the other methods. Excluding the last bin due to the insufficient statistic, we find that from all indirect estimators, the method using both the X_{max}- and R_{μ} predictions $\ln A$ seems to be the most stable. Moreover, analyzing the second moment, there is a good agreement with the second moment from the direct prediction (see Fig. 8.22). Since the second moment does not change under global shifts, we can estimate the average increase in the R_{μ} prediction which would reproduce the same absolute values of the first moment of $\ln A$ if we use the method based on X_{max} and R_{μ}. To do this, we use the calibrated values of X_{max}. We obtain a positive shift of 0.124 to reproduce the same average value as the direct $\ln A$ prediction. Hence, using this shift makes predictions of all independent NNs self-consistent.

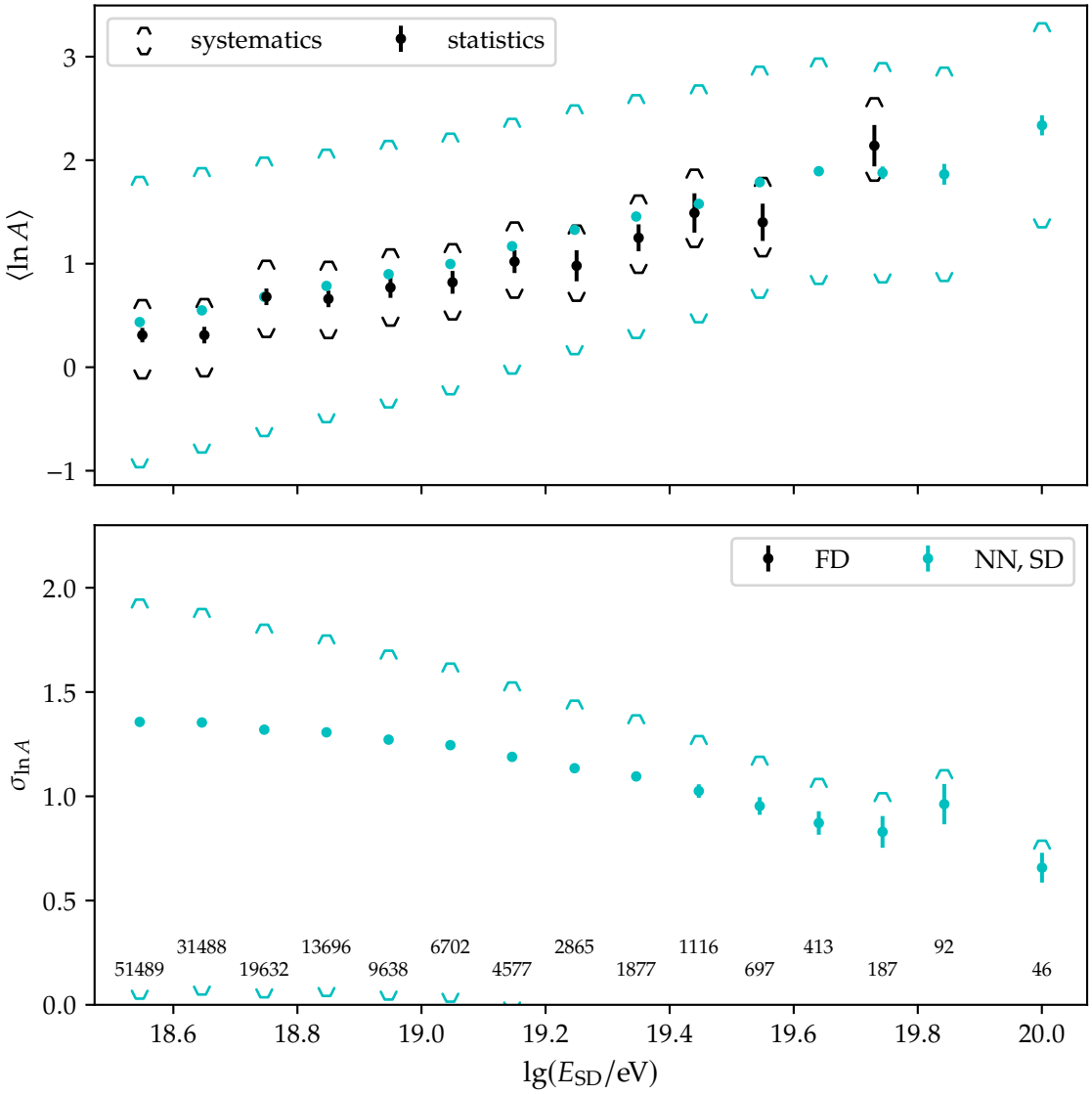


Figure 8.20: The first (*top*) and second moment (*bottom*) of $\ln A$ predicted by the NN (cyan) and from FD (black) using the hadronic interaction model QGSJetII-04 as a function of the reconstructed energy E_{SD} for the corrected predictions of the NN on the quality-selected SD data set (see Table 8.1). The predictions of the NN are shifted in such a way that the third bin of the first moment has the same value than the third bin of the FD estimate. The numbers in the *bottom* panel indicate the amount of events in each of the energy bins.

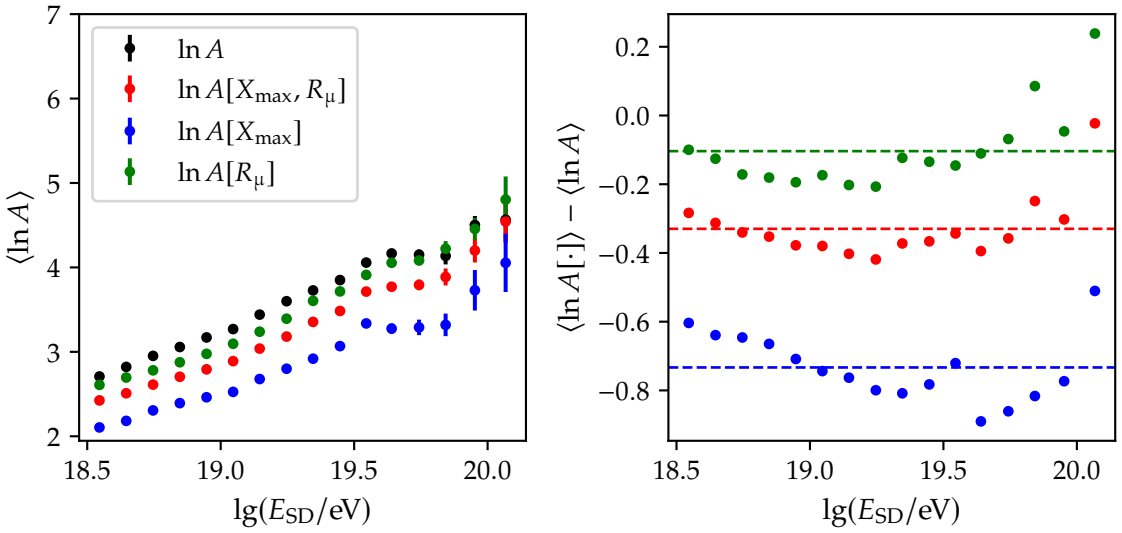


Figure 8.21: *Left*: Average reconstructed values of $\ln A$ on the quality-selected SD data set for different NN-based methods as a function of the reconstructed energy. The black markers are the raw predictions of the NN also used in Fig. 8.20. In this case, the values are not shifted. The other markers are indirect reconstructions using only the observables inside the square brackets. *Right*: Difference of the indirectly reconstructed average $\ln A$ values and the directly reconstructed average $\ln A$ value as a function of the reconstructed energy. The horizontal dashed lines are the color-coded average values of the data depicted in the same color.

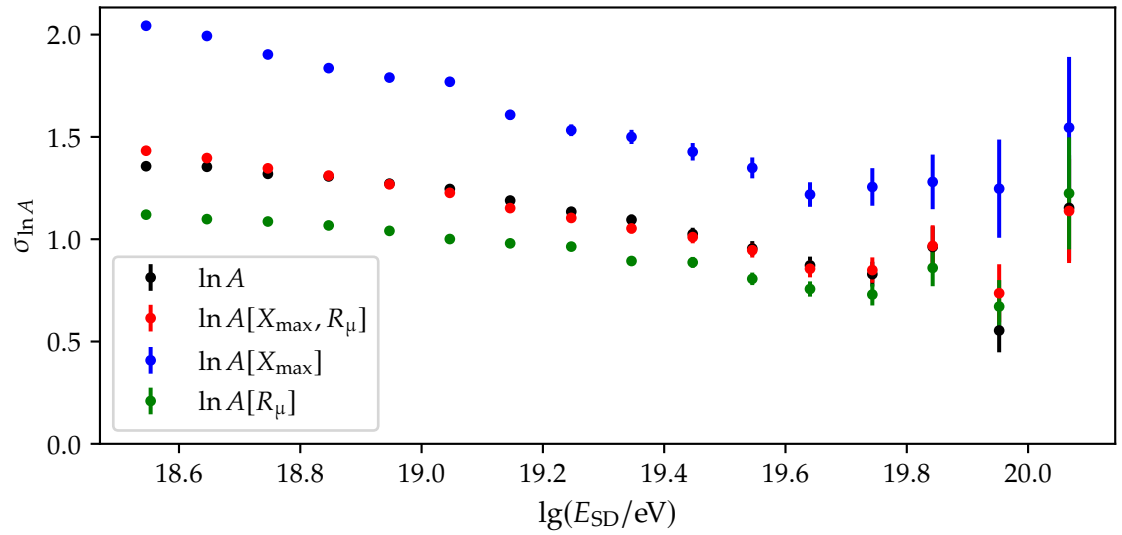


Figure 8.22: Second moment of $\ln A$ predicted by various NN-based methods in bins of the reconstructed energy. The black markers are the predictions of the NN also used in Fig. 8.20. The other markers represent second moments of the indirect estimators of $\ln A$ using only X_{max} (blue), using only R_μ (green), and using both X_{max} and R_μ (red). All of the quantities are reconstructed using the previous discussed NNs.

9 Conclusion



In order to understand the world, one has to turn away from it on occasion.

(Albert Camus)

DALL·E 2 prompt:

The ennui of writing conclusions for scientific papers as an abstract sculptur[.]

In this thesis, I performed an analysis of extensive air shower data based on neural networks. I studied both station-level and event-level data focusing on the general development of methods to optimally estimate the mass composition of UHECRs. A central part of the analysis was performed on extensive simulations of the detector responses of the surface detector of the Pierre Auger Observatory. I validated and contextualized preceding studies, which focused on either station-level or event-level analysis of air-shower data. I supplemented previous work and investigated new applications for the NN-based approaches in air shower physics. In addition, I validated the added value of the scintillator surface detectors of *AugerPrime* for various NN-based reconstruction approaches. At last, I applied the NNs-based event-level reconstruction mechanism I developed to Auger data to determine the mass composition of UHECR based on three separate observables.

Station-level analysis The main objective of the station-level analysis was to extract the local muon signal from the information provided by single SD stations alongside global shower observables since the muon signal distribution at the ground contains valuable information about the primary mass of the cosmic ray starting the air shower. Two different approaches proved themselves adequate for this task.

First, I studied a set of scalar observables to estimate their relation and correlation to the signal of the muonic shower component deposited in the SD stations of the Auger Observatory. I revisited one network architecture in detail, which has been used in one part of the preceding analysis. I showed that the performance of the NN is compatible with previous results; in the process of validating this approach. In the course of this, I demonstrated which of the possible input parameters are most sensitive to the desired output. To estimate the added value of an NN-based approach compared to an analytical model, I created multiple reference models to benchmark the NN performance. I showed that the NN approach has a slightly increased performance with respect to a linear fit model based on polynomial features generated up to the second order from the same inputs used for the NN. The analytical model, however, requires much fewer parameters than an NN of comparable performance

and is thus a valid and straightforward alternative as a model.

Secondly, I revisited and extended another LSTM-based NN architecture, which has been utilized to infer the muon time signal in the WCDs in the past, to assess the effect of adding SSD signal information on the reconstructed muon signal. To do this, I used a restricted data set solely consisting of UUB detector station responses from air showers induced by protons using the hadronic interaction model QGSJetII-04. I confirmed that the NN-based approach works using UUB data sub-sampled to a UB representation, as shown in the preceding analysis. Again, I created a reference model to quantify the performance of the network on the estimation of the total muon signal. I found that the NN-based approach outperforms the reference model for low muon signals but has a comparable performance for larger signals. Future analyses can exploit this to construct hybrid models working in different intervals of the distance to the shower core.

Adding the SSD trace signal for the training and inference did not significantly change the performance of the NNs. However, I discovered that by adding the fraction of the trace integrals of the SSD and WCD signals, the NN predictions outperform the NN using only WCD information. Therefore, I identified this fraction as a potential candidate for further investigation.

In addition, I did a feasibility study on using NNs to count the muons passing through the UMD. I developed a new NN architecture based on CNN, which utilizes weight sharing to account for the design of the UMD. By applying it to the binary traces of a small data set, I obtained a result that is less biased and more precise than that of the standard muon counter. For intermediate values of the actual number of muons, it also outperformed the predictions of the corrected muon counter. The shortcomings of the NN-based method are most likely due to the extremely low amount of available data. Therefore, more studies have to be done to further probe the procedure.

Event-level analysis on simulations The primary goal of the event-level analysis was to provide a reliable NN-based analysis tool for data taken by the SD array of the Pierre Auger Observatory to estimate mass-sensitive observables, such as the depth of the shower maximum X_{max} and the relative muon content R_μ of air showers. As the starting point for the analysis, I utilized an earlier version of the NN architecture, called AixNet, from a more simple preceding event-level analysis. In contrast to later iterations, I did not encode the symmetries of the SD array in the network architecture but developed a novel standardization procedure for shower footprints. This procedure removes over 90% of the phase space of our input data which streamlines the training procedure, enables the use of a simple network architecture, and improves the quality of network predictions. I showed that by standardizing the shower footprint before the training, I obtained superior network predictions without the additional cost of a more complex architecture. I gained an overall improvement of about $2\,\mathrm{g/cm^2}$ over the entire investigated energy range. All NNs for the event-level analysis have been trained using data set simulated using the hadronic interaction model QGSJetII-04 to cross-test the results of the preceding analysis that used EPOS-LHC.

The simpler network design allowed the training of large numbers of NNs in a short amount of time which enabled the study of how different modifications of the NN architecture affect the precision and primary-dependent bias of the NN predictions. Based on this architectural studies, I crafted a modified network architecture using additional inputs, such as the plane-front shower time. Using these architecture, I trained multiple sets of NNs on the standardized footprint data for the reconstruction of various high-level observables. Namely, I analyzed the predictions of the zenith angle, the primary particle energy, the depth of the shower maximum, the relative muon content, and the logarithmic mass number. The NNs based on the modified architecture were able to reconstruct the zenith angle as well as

the primary particle energy better than the SD standard reconstruction. Moreover, I showed that NNs trained on the modified architecture outperform NNs trained on a baseline architecture for the other three targets if sets of NNs are compared to each other. To select one NN from each set for the application on measurements, a metric was used that accounts for the proton-iron bias and the precision of the predictions. The chosen NNs models all use a CNN-based sub-network. To estimate the uncertainty of the predictions of the NNs, I developed a method to estimate the uncertainty of a single network prediction due to the detector setup and the non-determinism of the training process. In addition, I assessed the systematic uncertainties arising from the high-energy hadronic interaction and the unknown composition. Predicting on a data set simulated with the hadronic interaction model EPOS-LHC, I found a shift of $14\,\mathrm{g/cm^2}$ for X_{max}, 0.03 for R_{μ}, and 0.48 for $\ln A$.

Afterward, I studied the effect of the UUB electronics and the addition of the SSD on the predictions of NN-based approaches. Due to the finer sampling of the UUB compared to the UB electronics, NNs using UUB information exhibit, on average, a slight improvement in resolution and inter-primary bias of predictions of the depth of the shower maximum. To estimate the effect of adding the SSD to the inputs of the baseline network architecture, I showed that various downsampling strategies yield the same result. Comparing the performance of NNs using additional, downsampled SSD traces to that of the previous results, I demonstrated that the additional SSD improves the predictions of the primary particle energy, the relative muon content, and the logarithmic mass number. In all of these cases, the enhancement of the predictions came from the better separation of the used primaries. This result shows the potential of the SSD for primary-independent studies of air-shower events.

In the end, I applied the results of the simulation analysis in a feasibility study. Using a small data set comprised of only events induced by proton, iron, and photon primaries and a classifier based on the baseline architecture, I showed that adding the SSD improves the separation of hadronic and photon events. Moreover, comparing two sets of NN trained with and without additional SSD information revealed that using the SSD improved the expected performance of NNs trained for this task. The average gain from analyzing the ensembles of NNs was about three percentage points. Even though the first results are promising, future studies are needed to determine if the approach can be applied to the search of photons in data.

Event-level analysis on measurements Finally, I applied the selected NN models for the prediction of the depth of the shower maximum X_{max}, the relative muon content R_{μ}, and the logarithmic mass number $\ln A$ from the simulated study on measurements taken by Auger. Because of the difference between air-shower simulations and data, I corrected all predictions for unphysical biases in all target observables separately. To make the final comparisons, I selected a high-quality subset of the available SD data.

From all of the studied targets, the depth of the shower maximum is the only one that can be directly measured. Using the expected value of high-quality FD data, the corrected predictions of the NN showed a mostly constant offset of $30\,\mathrm{g/cm^2}$ over the entire energy interval. After corrections, the first and second moments of the predictions from the neural network were discussed as a function of the energy and compared to FD measurements. Above $10^{19.6}\,\mathrm{eV}$, the values of the last four energy bins of the second moment are higher than in the previous study on NNs. Applying an additional fiducial event selection did not resolve the issue. I discussed potential reasons for these tensions that have to be investigated in future studies.

Estimating the first and the second moment of R_{μ} as a function of energy, the average value of R_{μ} consistently lies slightly below the expected relative muon number of air showers induced by iron primaries. This result is surprising, because it shows a smaller muon deficit

than expected from preceding analyses. I attributed this behavior to averaging-toward-the-mean. The selected NN can not predict extreme values of R_μ due to the lack of training data without muon deficit.

This study is the first time a NN has been used to predict the atomic mass of UHECRs in terms of $\ln A$ directly from shower footprints. Comparing the first of the corrected $\ln A$ predictions of the NN to that derived from the QGSJetII-04 transformed $X_{\max}$ values measured by FD yields an excellent agreement. At the highest energies, the larger number of analyzed events allows us to resolve a plateau showing an almost constant value of 1.9 between $10^{19.6}$ eV and $10^{19.9}$ eV. Moreover, using the results of the $X_{\max}$ and R_μ analysis to obtain additional predictions for $\ln A$, I could determine that using the calibrated value of $X_{\max}$ and shifting R_μ up by 0.12, the combined prediction of $X_{\max}$ and R_μ show a similar result to the direct $\ln A$ prediction.

Summarizing, all of the studies performed in this work imply that the composition of UHECR becomes increasingly heavy with increased primary energy, which is in agreement with previous analyses. However, the result of this work is also compatible with the interpretation that at the highest energies the composition of CR becomes more mixed again.

Acronyms

A | C | D | E | F | G | H | I | K | L | M | N | O | P | Q | R | S | T | U | V | W | X

A

Adam Adaptive Moment Estimation. 45
ADC Analog/Digital Converter. 31
ADST Auger Data Summary Trees. 26
AGASA Akeno Giant Air Shower Array. 6
AGN Active Galactic Nuclei. 7
AIRES AIRshower Extended Simulations. 26
AixNet Air Shower Extraction Network. 57
AMIGA Auger Muons and Infill for the Ground Array. 20
ANN Artificial Neural Network. 2, 27
AP AugerPrime. 16
Auger Pierre Auger Observatory. 1, 6

C

CIC Constant Intensity Cut. 33
CLF central laser facility. 18
CMB Cosmic Microwave Background. 6
CO Coiheco. 16
CORSIKA COsmic Ray SImulations for KAscade. 15
CPU Central Processing Unit. 42
CR Cosmic Ray. 1, 5
CRP Cosmic Ray Physics. 5

D

DEC Direct Energy Calibration. 239

E

EAS Extensive Air Showers. 14

F

FADC Fast Analog/Digital Converter. 19
FD Fluorescence Detector. 2, 15
FFP Feed-Forward Predictor. 58
FN False Negative. 54
FOV field of view. 27
FP False Positive. 54
FPGA Field Programmable Gate Array. 124

G

GEC Group Equivariant Convolution. 229
GH Gaisser-Hillas. 13
GPS Global Positioning System. 190
GPU Graphics Processing Unit. 42
GW Gravitational Wave. 22
GZK Greisen–Zatsepin–Kuzmin. 6

H

HDF5 Hierarchical Data Format. 26
HiRes High Resolution Fly's eye. 6
HS Hottest Station. 29

I

ICRC International Cosmic Ray Conference. 70
IN2P3 IN2P3 Computing Center in Lyon. 66
IO Input-Output. 81

K

KarLib Karlsruhe air shower library. 69

L

LA Loma Amarilla. 16
LDF Lateral Distribution Function. 29
LHC Large Hadron Collider. 5
lidar Light Detection and Ranging. 18

LIV Lorentz invariance violation. 21
LL Los Leones. 16
LM Los Morados. 16
LSTM Long Short-term Memory. 52

M

MC Monte Carlo. 2, 20
MIP Minimum-Ionizing Particle. 19
MSE Mean Squared Error. 46

N

NapLib Napoli air shower library with Praha extension. 64
NKG Nishimura-Kamata-Greisen. 14
NN Neural Network. 2, 19

O

Observer Observer Task Force. 70

P

P3 third-polynomnial upscaled reference model. 119
PDF Probability Density Function. 82
PMT Photomultiplier Tube. 16
PReLU Parametric Rectified Linear Unit. 48

Q

QCD Quantum Chromodynamics. 7

R

ReLU Rectified Linear Unit. 48
RNN Recurrent Neural Network. 52
ROI region of interest. 118

S

SCA Spatial Correlation Analyzer. 57
SD Surface Detector. 1, 15
SDP shower detector plane. 27
SELU Scaled Exponential Linear Unit. 48
SGD Stochastic Gradient Descent. 45
SiPM Silicon photomultiplier. 124
SMELU Smooth Rectified Linear Unit. 48
SPMT Small Photomultiplier Tube. 19
SSD Surface Scintillator Detector. 1, 19

T

TeDs test data set. 44
TF TensorFlow. 42
TFE Trace Feature Extractor. 57
TN True Negative. 54
TP True Positive. 54
TrDs training data set. 44

U

UB Unified Board. 17
UHECR Ultra-High Energy Cosmic Ray. 5
UMD Underground Muon Detector. 2, 20
UTC Universal Time Coordinated. 190
UUB Upgraded Unified Board. 19
UV ultraviolet. 11

V

VaDs validation data set. 44
VEM Vertical Equivalent Muon. 17

W

WCD Water-Cherenkov detector. 1, 15

X

XLF eXtreme laser facility. 18

Bibliography

We split our references in different classes of accessibility and topic. In Auger there is a private server for publishing Auger specific analyses and content in form of short papers. These are called GAP (Giant Array Project) notes and could be considered internal papers. We separated these and private communications from published papers giving them the prefix A. Since this is a physics work which uses many methods from computer science, we split the officially published references (articles, proceedings, books, ...) in two categories separately. We gave the former the prefix P and the latter the prefix C. All other references, for things like computer programs, tools, websites, and everything that does not really fit into the other sections, are in the last section.

Internal references

[A:1] *Official Website of the Pierre Auger Observatory*. `https://www.auger.org/`. Accessed: 2022-01-30.

[A:2] A. Dorofeev et al. "FD Absolute Calibration, April 2013" GAP note 058 (2013).

[A:3] P. Billoir. "Aging effects on the calibration of the Surface Detector through the Vertical Equivalent Muon" GAP note 047 (2015).

[A:4] P. Filip. "Preliminary title: Neural Network based Triggers for the Surface Detector of the Pierre Auger Observatory" Future GAP note (2023).

[A:5] H. Dembinski and M. Roth. "Constant Intensity Cut method revisited: Uncertainty calculation with the Bootstrap" GAP note 074 (2011).

[A:6] D. Veberič et al. "Constant Intensity Cut: Unbinned Estimation of the Signal Attenuation Function" GAP note 065 (2015).

[A:7] M. Pothast, C. Timmermans, and S. de Jong. "SSD and WCD signal model" GAP note 058 (2021).

[A:8] C. J. T. Peixoto et al. "Application of the Δ-method to the Phase I Data" GAP note 024 (2021).

[A:9] A. Guillén et al. "An estimation of the muon signal registered by the Surface Detector of the Pierre Auger Observatory using Deep Neural Networks" GAP note 014 (2018).

[A:10] A. Guillén et al. "An estimation of the muon signal registered by the Surface Detector of the Pierre Auger Observatory using Deep Neural Networks: An application to experimental data" GAP note 020 (2018).

[A:11] A. Bueno, J. M. Carceller, and A. A. Watson. "Modern analysis techniques meet old measurements" GAP note 027 (2018).

[A:12] A. Bueno et al. "Extraction of the Muon Traces Recorded by the Surface Detector" GAP note 004 (2019).

[A:13] T. Pan et al. "Mass Composition Studies of Extensive Air Showers with Deep Learning" GAP note 056 (2017).

[A:14] M. Erdmann and J. Glombitza. "Deep Neural Network for the Reconstruction of the Shower Maximum Xmax using the Water-Cherenkov Detectors of the Pierre Auger Observatory" GAP note 005 (2020).

[A:15] R. Colalillo, F. Guarino, and A. Yushkov. "Napoli + Praha Library: A Brief Guide to the Summary Trees, Including the Correct Way to Calculate Components of Signals in SD Stations." GAP note 069 (2019).

[A:16] E. Santos and A. Yushkov. "Extending the Naples CORSIKA shower library for Auger studies at $16.5 \leq \log_{10}[E/\text{eV}] \leq 18.0$" GAP note 043 (2018).

[A:17] *Official Auger Observer Website.* `https://web.ikp.kit.edu/observer/`. Accessed: 2022-07-18.

[A:18] V. Harvey. *Private Communication: FD Data from 2020 on.* University of Adelaide. 2022.

[A:19] M. Stadelmaier. *Private Communication: Shower front time improves the Universality reconstruction.* 2021.

[A:20] A. Bueno et al. "A New Observable to Infer the Chemical Composition of Cosmic Rays of Ultra-High Energy" GAP note 068 (2017).

[A:21] A. Bueno, J. M. Carceller, and A. A. Watson. "Study of the Shower-to-Shower Fluctuations using the Risetime Measurements of the Water-Cherenkov Detectors" GAP note 042 (2018).

[A:22] D. Schmidt. *Private Communication: Baseline-fluctuations.* 2019.

[A:23] A. Bueno and J. M. Carceller. "Extraction of the Muon Trace Recorded by the Surface Detector of The Pierre Auger Observatory using Recurrent Neural Networks" GAP note 006 (2020).

[A:24] R. Engel. *Private Communication: Fraction of SSD and WCD signals is a good starting point for analysis of the added value of the SSD.* 2022.

[A:25] M. Stadelmaier. *Private Communication: Fraction of SSD and WCD signals decreases bias in Universality reconstruction.* 2022.

[A:26] J. de Jesús, J. M. Figueira, and F. Sánchez. "Study on the optical fiber attenuation of the UMD using the integrator channel" GAP note 052 (2021).

[A:27] J. de Jesús, J. M. Figueira, and F. Sánchez. "Study on the optical fiber attenuation of the UMD using the binary channel" GAP note 034 (2021).

[A:28] J. D. Jesús. *Private Communication: UMD traces with additional variables.* 2022.

[A:29] F. Ellwanger. "Preliminary title: Neural Network based Prediction of Primary Particle Energy using the Surface Detector of the Pierre Auger Observatory" Future GAP note (2022).

Physics references

[P:1] S. Mollerach and E. Roulet, "Progress in high-energy cosmic ray physics", *Prog. Part. Nucl. Phys.* **98** (2018) 85–118.

[P:2] T. K. Gaisser, R. Engel, and E. Resconi. *Cosmic rays and particle physics*. Cambridge University Press, 2016.

[P:3] L. Evans and P. Bryant, "LHC machine", *J. Instrum.* **3**:08 (2008) S08001.

[P:4] H. Becquerel, "Sur les radiations émises par phosphorescence.", *C. R. Phys.* **122** (1) 420–421.

[P:5] J. R. Hörandel. "Early cosmic-ray work published in German." In: *AIP Conference Proceedings*. Vol. 1516. 1. American Institute of Physics. 2013, 52–60.

[P:6] T. Wulf, "On the radiation of high penetrating power that exists in the atmosphere", *Phys. Zeit* **1**:152-157 (1909) 124.

[P:7] D. Pacini, "Sulle radiazioni penetranti", *Rend. Acc. Lincei* **18** (1909) 123.

[P:8] A. De Angelis, N. Giglietto, and S. Stramaglia, "Domenico Pacini, the forgotten pioneer of the discovery of cosmic rays." , *arXiv: 1002.2888* (2010).

[P:9] T. Wulf et al., "Observations on the radiation of high penetration power on the Eiffel tower", *Z. Phys.* **11**:811 (1910) 2155–304.

[P:10] V. F. Hess, "Uber Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten", *Phys. Zeits.* **13** (1912) 1084–1091.

[P:11] W. F. Swann, "The history of cosmic rays", *Am. J. Phys.* **29**:12 (1961) 811–816.

[P:12] B. Rossi, "On the magnetic deflection of cosmic rays", *Phys. Rev.* **36**:3 (1930) 606.

[P:13] B. Rossi, "Über die Eigenschaften der durchdringenden Korpuskularstrahlung im Meeresniveau", *Z. Phys.* **82**:3 (1933) 151–178.

[P:14] S. H. Neddermeyer and C. D. Anderson, "Note on the nature of cosmic-ray particles", *Phys. Rev.* **51**:10 (12 1937) 884.

[P:15] P. Auger et al., "Extensive cosmic-ray showers", *Rev. Mod. Phys.* **11**:3-4 (1939) 288.

[P:16] W. Heitler. *The quantum theory of radiation*. Courier Corporation, 1984.

[P:17] J. Linsley, "Evidence for a primary cosmic-ray particle with energy 10^{20} eV", *Phys. Rev. Lett.* **10**:4 (1963) 146.

[P:18] K. Greisen, "End to the cosmic-ray spectrum?" *Phys. Rev. Lett.* **16**:17 (1966) 748.

[P:19] G. T. Zatsepin and V. A. Kuzmin, "Upper limit of the spectrum of cosmic rays", *JETP Lett.* **4** (1966) 78.

[P:20] R. Abbasi et al., "Observation of the ankle and evidence for a high-energy break in the cosmic ray spectrum", *Mod. Phys. Lett. B* **619**:3-4 (2005) 271–280.

[P:21] S. Yoshida et al., "The Cosmic ray energy spectrum above 3×10^{18} eV measured by the Akeno Giant Air Shower Array", *Astropart. Phys.* **3** (1995) 105–124.

[P:22] R. L. Workman et al., "Review of Particle Physics", *Prog. Theor. Exp. Phys.* **2022** (2022) 083C01.

[P:23] A. M. Hillas, "The origin of ultra-high-energy cosmic rays", *Annu. Rev. Astron. Astrophys.* **22** (1984) 425–444.

[P:24] R. Alves Batista et al., "Open Questions in Cosmic-Ray Research at Ultrahigh Energies", *Front. Astron. Space Sci.* **6** (2019) 23.

[P:25] D. Mockler. "Measurement of the Cosmic Ray Spectrum with the Pierre Auger Observatory." PhD thesis. Karlsruher Institut für Technologie (KIT), 2019.

[P:26] P. Abreu et al., "The energy spectrum of cosmic rays beyond the turn-down around 10^{17} eV as measured with the surface detector of the Pierre Auger Observatory", *Eur. Phys. J. C.* **81**:11 (2021) 1–25.

[P:27] A. Aab et al., "Observation of a large-scale anisotropy in the arrival directions of cosmic rays above 4×10^{19} eV", *Phys. Rev. Lett.* **101** (6 2008) 061101.

[P:28] A. Aab et al., "Large-scale cosmic-ray anisotropies above 4 EeV measured by the Pierre Auger Observatory", *Astrophys. J.* **868**:1 (2018) 4.

[P:29] O. Deligny, "The energy spectrum of ultra-high energy cosmic rays measured at the Pierre Auger Observatory and at the Telescope Array", *PoS* **ICRC2019** (2019) 234.

[P:30] T. Antoni et al., "KASCADE measurements of energy spectra for elemental groups of cosmic rays: Results and open problems", *Astropart. Phys.* **24**:1-2 (2005) 1–25.

[P:31] T. K. Gaisser, "The Cosmic-ray Spectrum: from the knee to the ankle", *J. Phys. Conf. Ser.* **47** (2006) 15–20.

[P:32] M. Unger, G. R. Farrar, and L. A. Anchordoqui, "Origin of the ankle in the ultrahigh energy cosmic ray spectrum, and of the extragalactic protons below it", *Phys. Rev. D* **92** (12 2015) 123001.

[P:33] R. Aloisio, V. Berezinsky, and A. Gazizov, "Transition from galactic to extragalactic cosmic rays", *Astropart. Phys.* **39** (2012). Cosmic Rays Topical Issue 129–143.

[P:34] D. Allard, "Extragalactic propagation of ultrahigh energy cosmic-rays", *Astropart. Phys.* **39** (2012). Cosmic Rays Topical Issue 33–43.

[P:35] S. Seager, D. D. Sasselov, and D. Scott, "A new calculation of the recombination Epoch", *Astrophys. J.* **523**:1 (1999) L1.

[P:36] S. Weinberg. *Cosmology*. Oxford University Press, 2008.

[P:37] D. Fixsen, "The temperature of the cosmic microwave background", *Astrophys. J.* **707**:2 (2009) 916.

[P:38] G. Baldwin and G. Klaiber, "Photo-fission in heavy elements", *Phys. Rev.* **71** (1 1947) 3–10.

[P:39] J. Matthews, "A Heitler model of extensive air showers", *Astropart. Phys.* **22**:5-6 (2005) 387–397.

[P:40] R. Engel. "Air Shower Calculations With the New Version of SIBYLL." In: *I26th International Cosmic Ray Conference (ICRC26)*. Vol. 1. International Cosmic Ray Conference. 1999, 415.

[P:41] T. K. Gaisser and A. M. Hillas. "Reliability of the method of constant intensity cuts for reconstructing the average development of vertical showers." In: *International Cosmic Ray Conference*. Vol. 8. International Cosmic Ray Conference. 1977, 353–357.

[P:42] P. Lipari, "The Concepts of 'Age' and 'Universality' in Cosmic Ray Showers", *Phys. Rev. D* **79**:6 (2009) 063001.

[P:43] P. Lipari, "Universality in the longitudinal development of Cosmic Ray showers", *Nucl. Part. Phys. Proc.* **279** (2016). Proceedings of the 9th Cosmic Ray International Seminar 111–117.

[P:44] K. Kamata and J. Nishimura, "The lateral and the angular structure functions of electron showers", *Progress of Theoretical Physics, Supplement* **6** (1958) 93–155.

[P:45] K. Greisen, "Progress in Cosmic Ray Physics", *JG Wilson, Amsterdam, Netherlands* (1956).

[P:46] J. Knapp et al., "Extensive air shower simulations at the highest energies", *Astropart. Phys.* **19**:1 (2003) 77–99.

[P:47] T. Pierog et al., "EPOS LHC: Test of collective hadronization with data measured at the CERN Large Hadron Collider", *Phys. Rev. C* **92** (3 2015) 034906.

[P:48] S. Ostapchenko, "QGSJET-II: towards reliable description of very high energy hadronic interactions", *NuPhS* **151**:1 (2006) 143–146.

[P:49] R. S. Fletcher et al., "SIBYLL: An Event generator for simulation of high-energy cosmic ray cascades", *Phys. Rev. D* **50** (9 1994) 5710–5731.

[P:50] A. Aab et al., "Muons in air showers at the Pierre Auger Observatory: Mean number in highly inclined events", *Phys. Rev. D* **91**:3 (2015) 032003.

[P:51] D. Zavrtanik and P. A. Collaboration, "The Pierre Auger Observatory", *Nucl. Phys. B* **85**:1-3 (2000) 324–331.

[P:52] A. Aab et al., "The Pierre Auger Cosmic Ray Observatory", *Nucl. Instrum. Methods Phys. Res. A* **798** (2015) 172–213.

[P:53] J. Abraham et al., "The fluorescence detector of the Pierre Auger Observatory", *Nucl. Instrum. Methods Phys. Res. A* **620**:2-3 (2010) 227–251.

[P:54] J. Brack et al., "Absolute photometric calibration of large aperture optical systems", *Astropart. Phys.* **20**:6 (2004) 653–659.

[P:55] J. F. Debatin. "Preparation of the Operation and Calibration of the Fluorescence Detector of AugerPrime." PhD thesis. Karlsruher Institut für Technologie (KIT), 2020.

[P:56] R. Knapik et al., "The absolute, relative and multi-wavelength calibration of the Pierre Auger observatory fluorescence detectors", *arXiv:0708.1924* (2007).

[P:57] I. Allekotte et al., "The surface detector system of the Pierre Auger Observatory", *Nucl. Instrum. Methods Phys. Res. A* **586**:3 (2008) 409–420.

[P:58] X. Bertou et al., "Calibration of the surface array of the Pierre Auger Observatory", *Nucl. Instrum. Methods Phys. Res. A* **568**:2 (2006) 839–846.

[P:59] D. Veberič, P. A. Collaboration, et al. "Estimation of the Total Signal in Saturated Stations of Pierre Auger Surface Detector." In: *International Cosmic Ray Conference*. Vol. 33. 2013, 0633.

[P:60] J. Hörandel et al., "A large radio array at the Pierre Auger Observatory", *Eur. Phys. J.* (2018).

[P:61] P. Abreu et al., "Description of atmospheric conditions at the pierre auger observatory using the global data assimilation system (gdas)", *Astropart. Phys.* **35**:9 (2012) 591–607.

[P:62] P. Abreu et al., "Techniques for Measuring Aerosol Attenuation using the Central Laser Facility at the Pierre Auger Observatory", *J. Instrum.* **8**:04 (2013) P04009.

[P:63] S. BenZvi et al., "The Lidar system of the Pierre Auger Observatory", *Nucl. Instrum. Methods Phys. Res. A* **574**:1 (2007) 171–184.

[P:64] A. Aab et al., "The Pierre Auger Observatory upgrade-preliminary design report." , *arXiv: 1604.03637* (2016).

[P:65] G. Marsella et al., "AugerPrime Upgraded Electronics", *tbp*, 230 (2022).

[P:66] A. Castellina, "The dynamic range of the AugerPrime Surface Detector: technical solution and physics reach", *PoS* **301** (2017). Ed. by D. Veberic 397–1.

[P:67] A. Etchegoyen. "AMIGA, auger muons and infill for the ground array." In: *International Cosmic Ray Conference*. Vol. 5. 2008, 1191–1194.

[P:68] A. Botti et al., "Status and performance of the underground muon detector of the Pierre Auger Observatory", *PoS* **ICRC2021** (2022).

[P:69] A. Aab et al., "Reconstruction of inclined air showers detected with the Pierre Auger Observatory", *JCAP* **08** (2014) 019.

[P:70] J. R. Hörandel. "Precision measurements of cosmic rays up to the highest energies with a large radio array at the Pierre Auger Observatory." In: *EPJ Web of Conferences*. Vol. 210. EDP Sciences. 2019.

[P:71] J. L. Kelley et al., "AERA: the auger engineering radio array", *Proc. of the 32nd ICRC* (2011).

[P:72] R. Alves Batista et al., "Open Questions in Cosmic-Ray Research at Ultrahigh Energies", *Front. Astron. Space Sci.* **6** (2019) 23.

[P:73] A. Aab et al., "Measurement of the fluctuations in the number of muons in extensive air showers with the Pierre Auger Observatory", *Phys. Rev. Lett.* **126**:15 (2021) 152002.

[P:74] A. Aab et al., "Features of the energy spectrum of cosmic rays above 2.5×10^{18} eV using the Pierre Auger Observatory", *Phys. Rev. Lett.* **125**:12 (2020) 121106.

[P:75] I. Valino, "The flux of ultra-high energy cosmic rays after ten years of operation of the Pierre Auger Observatory", *PoS* **ICRC2015** (2016) 271.

[P:76] J. Abraham et al., "Observation of the suppression of the flux of cosmic rays above 4× 10 19 eV", *Phys. Rev. Lett.* **101**:6 (2008) 061101.

[P:77] P. Abreu et al., "Testing effects of Lorentz invariance violation in the propagation of astroparticles with the Pierre Auger Observatory", *J. Cosmol. Astropart. Phys.* **2022**:01 (2022) 023.

[P:78] A. Yushkov et al. "Mass Composition of Cosmic Rays with Energies above $10^{17.2}$ eV from the Hybrid Data of the Pierre Auger Observatory." In: *36th International Cosmic Ray Conference*. Vol. 358. SISSA Medialab. 2021, 482.

[P:79] M. Ave et al., "Precise measurement of the absolute fluorescence yield of the 337 nm band in atmospheric gases", *Astropart. Phys.* **42** (2013) 90–102.

[P:80] A. Aab et al., "Data-driven estimation of the invisible energy of cosmic ray showers with the Pierre Auger Observatory", *Phys. Rev. D* **100**:8 (2019) 082003.

[P:81] A. Aab et al., "Depth of maximum of air-shower profiles at the Pierre Auger Observatory. I. Measurements at energies above $10^{17.8}$ eV", *Phys. Rev. D* **90** (12 2014) 122005.

[P:82] V. Verzi. "The Energy Scale of the Pierre Auger Observatory." In: *33rd International Cosmic Ray Conference*. 2013, 0928.

[P:83] B. Dawson, "The Energy Scale of the Pierre Auger Observatory", *PoS* **ICRC2019** (2019) 231.

[P:84] A. Aab et al., "The Pierre Auger Observatory: Contributions to the 36th International Cosmic Ray Conference (ICRC 2019)", *arXiv:1909.09073* (2019).

[P:85] J. Abraham et al., "Measurement of the depth of maximum of extensive air showers above 10^{18} eV", *Phys. Rev. Lett.* **104** (9 2010) 091101.

[P:86] A. Aab et al., "Depth of maximum of air-shower profiles at the Pierre Auger Observatory. II. Composition implications", *Physical Review D* **90**:12 (2014) 122006.

[P:87] D. Veberič et al. "On the need for unbiasing azimuthal asymmetries in signals measured by surface detector arrays." In: vol. ICRC2021. 2021, 435.

[P:88] H. P. Dembinski et al., "A likelihood method to cross-calibrate air-shower detectors", *Astropart. Phys.* **73** (2016) 44–51.

[P:89] D. Schmidt. "Sensitivity of AugerPrime to the masses of ultra-high-energy cosmic rays." PhD thesis. Karlsruher Institut für Technologie (KIT), 2019.

[P:90] Á. Taboada Núñez. "Analysis of the First Data of the AugerPrime Detector Upgrade." PhD thesis. Karlsruher Institut für Technologie (KIT), 2020.

[P:91] P. Sánchez Lucas. "The Δ Method: An estimator for the mass composition of ultra-high-energy cosmic rays." PhD thesis. Universidad de Granada, 2016.

[P:92] A. Aab et al., "Inferences on mass composition and tests of hadronic interactions from 0.3 to 100 EeV using the water-Cherenkov detectors of the Pierre Auger Observatory", *Phys. Rev. D* **96**:12 (2017) 122003.

[P:93] C. Peixoto, J. Todero, P. A. Collaboration, et al., "Estimating the depth of shower maximum using the surface detectors of the Pierre Auger Observatory", *PoS* **ICRC2019** (2019) 440.

[P:94] A. Hillas, "Angular and energy distributions of charged particles in electron-photon cascades in air", *J. Phys. G* **8**:10 (1982) 1461.

[P:95] F. Nerling et al., "Universality of electron distributions in high-energy air showers – Description of Cherenkov light production", *Astropart. Phys.* **24**:6 (2006) 421–437.

[P:96] M. Ave, M. Roth, and A. Schulz, "A generalized description of the time dependent signals in extensive air shower detectors and its applications", *Astropart. Phys.* **88** (2017) 46–59.

[P:97] M. Detlef. "Mass Composition of Ultra-High Energy Cosmic Rays Based on Air Shower Universality." PhD thesis. Karlsruher Institut für Technologie (KIT), 2013.

[P:98] A. Schulz. "Measurement of the Energy Spectrum and Mass Composition of Ultra-high Energy Cosmic Rays." PhD thesis. Karlsruher Institut für Technologie (KIT), 2016.

[P:99] J. Hulsman. "Hybrid Universality Model Development and Air Shower Reconstruction for the Pierre Auger Observatory." PhD thesis. Karlsruher Institut für Technologie (KIT), 2020.

[P:100] M. K. Stadelmaier. "On Air-Shower Universality and the Mass Composition of Ultra-High-Energy Cosmic Rays." PhD thesis. Karlsruher Institut für Technologie (KIT), 2022.

[P:101] A. Aab et al., "Extraction of the muon signals recorded with the surface detector of the Pierre Auger Observatory using recurrent neural networks", *J. Instrum.* **16**:07 (2021) P07016.

[P:102] A. Aab et al., "Deep-learning based reconstruction of the shower maximum X max using the water-Cherenkov detectors of the Pierre Auger Observatory", *J. Instrum.* **16**:07 (2021) P07019.

[P:103] J. M. Carceller López et al., "A study of the signals measured with the Water-Cherenkov detectors of the Pierre Auger observatory to infer the mass composition of ultra-high energy cosmic rays", (2020).

[P:104] J. Glombitza. "Deep-learning based measurement of the mass composition of ultra-high energy cosmic rays using the surface detector of the Pierre Auger Observatory." PhD thesis. RWTH Aachen University, 2021.

[P:105] J. Glombitza et al. "Air-shower reconstruction at the Pierre Auger Observatory based on deep learning." In: *Proceedings of the 36th International Cosmic Ray Conference*. Univerza v Novi Gorici. 2019.

[P:106] P. Abreu et al., "Event-by-event reconstruction of the shower maximum Xmax with the Surface Detector of the Pierre Auger Observatory using deep learning", *PoS* **ICRC2021** (2022) 359.

[P:107] A. Guillén et al., "Deep learning techniques applied to the physics of extensive air showers", *Astropart. Phys.* **111** (2019) 12–22.

[P:108] S. Cuomo et al., "Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next", *arXiv:2201.05624* (2022).

Computer science references

[C:1] A. Ramesh et al., "Hierarchical Text-Conditional Image Generation with CLIP Latents." , *arXiv: 2204.06125* (2022).

[C:2] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural. Netw.* **2**:5 (1989) 359–366.

[C:3] M. Z. Alom et al., "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches", *arXiv:1803.01164* (2018).

[C:4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Adv. Neural. Inf. Process. Syst.* **25** (2012).

[C:5] K. He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In: *Proceedings of the IEEE international conference on computer vision*. 2015, 1026–1034.

[C:6] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[C:7] H. B. Curry, "The method of steepest descent for non-linear minimization problems", *Q. Appl. Math.* **2**:3 (1944) 258–261.

[C:8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv:1412.6980* (2014).

[C:9] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond", *arXiv:1904.09237* (2019).

[C:10] T. Dozat, "Incorporating nesterov momentum into adam", (2016).

[C:11] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, 249–256.

[C:12] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen", *TUM* **91**:1 (1991).

[C:13] S. Hochreiter et al. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. Chapter of the book: A Field Guide to Dynamical Recurrent Networks. 2001.

[C:14] L. Lu et al., "Dying relu and initialization: Theory and numerical examples", *arXiv:1903.06733* (2019).

[C:15] G. Klambauer et al., "Self-normalizing neural networks", *Adv. Neural. Inf. Process. Syst.* **30** (2017).

[C:16] X. Cheng et al., "Pest identification via deep residual learning in complex background", *Comput. Electron. Agric.* **141** (2017) 351–356.

[C:17] G. Huang et al. "Densely connected convolutional networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 4700–4708.

[C:18] A. Graves, A.-r. Mohamed, and G. Hinton. "Speech recognition with deep recurrent neural networks." In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, 6645–6649.

[C:19] Y. Wu et al., "Google's neural machine translation system: Bridging the gap between human and machine translation", *arXiv:1609.08144* (2016).

[C:20] C. Berner et al., "Dota 2 with large scale deep reinforcement learning", *arXiv:1912.06680* (2019).

[C:21] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[C:22] K. He et al. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778.

[C:23] A. O'hagan and T. Leonard, "Bayes estimation subject to uncertainty about parameter constraints", *Biometrika* **63**:1 (1976) 201–203.

[C:24] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization", *Comput. J.* **7** (1965) 308–313.

[C:25] M. F. Dacrema, P. Cremonesi, and D. Jannach. "Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches." In: *Proceedings of the 13th ACM Conference on Recommender Systems*. RecSys '19. Copenhagen, Denmark: ACM, 2019, 101–109.

[C:26] T. Cohen and M. Welling. "Group equivariant convolutional networks." In: *International conference on machine learning*. PMLR. 2016, 2990–2999.

[C:27] L. Breiman, "Bagging predictors", *Mach. Learn.* **24**:2 (1996) 123–140.

Other references

[T:A] M. León-Portilla. *The Phoenix of the Western World: Quetzalcoatl and the Sky Religion*. Duke University Press, 1983.

[T:B] Plato. *Timaeus*. BoD – Books on Demand, 2019.

[T:C] D. Heck et al. *CORSIKA: A Monte Carlo code to simulate extensive air showers*. Tech. rep. 1998, 399. DOI: 10.5445/IR/270043064.

[T:D] *Natural Earth map data*. Accessed: 07.07.2022. URL: https://www.naturalearthdata.com.

[T:E] S. Argirò et al., "The offline software framework of the Pierre Auger Observatory", *Nucl. Instrum. Methods Phys. Res. A.* **580**:3 (2007) 1485–1496.

[T:F] R. Brun and F. Rademakers, "ROOT–An object oriented data analysis framework", *Nucl. Instrum. Methods Phys. Res. A.* **389**:1-2 (1997) 81–86.

[T:G] *Official Website of ImageNet Large Scale Visual Recognition Challenge*. https://image-net.org. Accessed: 15.07.2022.

[T:H] A. Martin et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", *arXiv:1603.04467* (2016). Software available from tensorflow.org.

[T:I] *Reproducibility in Deep Learning and Smooth Activations*. https://ai.googleblog.com/2022/04/reproducibility-in-deep-learning-and.html. Accessed: 16.07.2022.

[T:J] C. R. Rao et al. *Linear statistical inference and its applications*. Vol. 2. Wiley New York, 1973.

[T:K] *Official Website of IN2P3 Computing Centre in Lyon*. https://cc.in2p3.fr. Accessed: 15.07.2022.

[T:L] *Bad Periods for SD array*. https://www0.mi.infn.it/auger/files/BadPeriods. Accessed: 29.08.2022.

[T:M] *Monitoring website of detectors the Pierre Auger Observatory.* http://mon.auger.uni-wuppertal.de/pro. Accessed: 2022-10-18. July 2011. arXiv: 1107.4806 [astro-ph.IM].

[T:N] P. Virtanen et al., *Nat. Methods* **17** (2020) 261–272.

[T:O] *Experimental enable determinism option for TensorFlow 2.9.1.* https://www.tensorflow.org/api_docs/python/tf/config/experimental/enable_op_determinism. Accessed: 22.08.2022.

A Definitions and Derivations

> My name is immaterial,' she said. That's a pretty name,' said Rincewind.
>
> (The Colour of Magic, Terry Pratchett)

A.1 Notes on additional statistical quantities

We have copied the following definitions from [T:J]: Let $\{x_1, \ldots, x_N\}$ be a set of independent values drawn from an arbitrary distribution and x a random variable corresponding to this distribution. Then, the nth raw moment is

$$v_n = \langle x^n \rangle \,, \tag{A.1}$$

and the nth central moment is

$$\mu_n = \langle (x - v_1)^n \rangle \,, \tag{A.2}$$

where $\langle \cdot \rangle$ returns the expected value (see Eq. (4.34)). Moreover, we define the nth raw moment as

$$O_n = \frac{1}{N} \sum x_i^n. \tag{A.3}$$

and the nth corrected moment as

$$m_n = \frac{1}{N} \sum (x_i - O_1)^n. \tag{A.4}$$

For all possible distributions the variance VAR of the variance is then

$$\mathrm{VAR}(m_2) = \mathrm{VAR}\left(O_2 - O_1^2\right) = \frac{(N-1)^2}{N^3} \left(\mu_4 - \frac{N-3}{N-1}\mu_2^2\right). \tag{A.5}$$

From this, we obtain for all possible distributions

$$\mathrm{SE}\left(\frac{n}{n-1}(O_2 - O_1^2)\right) = \frac{1}{\sqrt{N}} \sqrt{\mu_4 - \frac{n-3}{n-1}\mu_2^2} \tag{A.6}$$

as standard error for the sample variance. In first-order approximation, the standard error of the standard deviation is then

$$\mathrm{SE}(\sqrt{\sigma^2(x)}) \approx \frac{1}{2\sqrt{\sigma^2(x)}} \mathrm{SE}(\sigma^2(x)). \tag{A.7}$$

Using uncertainty propagation, we obtain the variance of the standard deviation via

$$\mathrm{VAR}(\sqrt{\sigma^2(x)}) \approx \frac{1}{4\sigma^2(x)} \mathrm{VAR}(\sigma^2(x)). \tag{A.8}$$

If we estimate multiple standard deviations from independent subsamples with low sample sizes, we estimate the expected value of the standard deviation via the variance-weighted mean to ensure that we do not over-predict and obtain a biased value. We define the variance weighted average as

$$\langle \sigma \rangle_\mathrm{V} = \frac{1}{\sum \frac{1}{\mathrm{VAR}(\sigma_i)}} \sum \frac{\sigma_i}{\mathrm{VAR}(\sigma_i)}. \tag{A.9}$$

We also use the variance-weighted average to combine our measurements in the case when multiple FD telescopes measure the same shower (see Sec. 5.2.2.A).

A.2 Definition of commonly used fit functions

In this thesis, we use a hand full of simple fit functions for modeling the behavior found in data. Here, we want to introduce the most important of these fit functions and define the naming of the fit parameters.

We define polynomial fit functions as

$$P_N(x; p_0, \ldots, p_N) = \sum_{n=0}^{N} p_n x^n, \tag{A.10}$$

where n is the order of the polynomial P_N. We use

$$S(x; p_0, \ldots, p_3) = p_0 + p_1 \sin(p_2 x + p_3) \tag{A.11}$$

to fit periodic behaviors, such as dependencies on the azimuth angle. If we encounter step-like behavior, we use a tanh function

$$T(x; p_0, \ldots, p_3) = p_0 + p_1 \tanh(p_2 x + p_3). \tag{A.12}$$

We use an exponential function of the form

$$E(x; p_0, p_1, p_2) = p_0 + p_1 \exp(p_2 x), \tag{A.13}$$

if the underlying data decreases or increases 'very suddenly'.

A.3 Standardization for an arbitrary choice of unit vectors

The standardization procedure discussed in Sec. 5.3.3, depends on the arbitrarily chosen unit vectors (see Eq. (5.6)). Here, we provide a way to use this method for any possible set of unit vectors $\{\boldsymbol{u}, \boldsymbol{v}\}$ and transformation matrices. Using the regular, Cartesian rotation matrix we find

$$\boldsymbol{R}_{60} = \boldsymbol{T}^{-1} \boldsymbol{R}^{\mathrm{c}}_{60} \boldsymbol{T} \equiv \boldsymbol{T}^{-1} \begin{pmatrix} \cos 60° & \sin 60° \\ -\sin 60° & \cos 60° \end{pmatrix} \begin{pmatrix} \boldsymbol{u} & \boldsymbol{v} \end{pmatrix}, \tag{A.14}$$

where $\boldsymbol{T}$ is the transformation matrix which connects x, y (see Eq. (5.6)) with the Cartesian coordinate system and the superscript "c" marks a Cartesian rotation matrix (clockwise, 60°). A way to obtain $\boldsymbol{M}_j$ ($j \in \{0, 30, 60, 90, 120, 150\}$) is by transforming the Cartesian (x-axis) reflection matrix $\boldsymbol{M}^{\mathrm{c}}_0$:

$$\boldsymbol{M}_j = \boldsymbol{T}^{-1} \boldsymbol{M}^{\mathrm{c}}_j \boldsymbol{T} = \boldsymbol{T}^{-1} \boldsymbol{R}^{\mathrm{c}}_j \boldsymbol{M}^{\mathrm{c}}_0 \boldsymbol{R}^{\mathrm{c}}_{-j} \boldsymbol{T} = \boldsymbol{T}^{-1} \boldsymbol{R}^{\mathrm{c}}_j \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{R}^{\mathrm{c}}_{-j} \boldsymbol{T}. \tag{A.15}$$

Alternatively, the correct transformation matrices can be found by selecting a base and labeling the grid points up to the second crown with the corresponding integer coordinates. By checking which coordinates the points should have after a certain transformation, we can find four independent equations that can be used to determine the transformation matrix.

With Eq. (A.14) and Eq. (A.15), we are able to compute the transformed azimuth angle by using a unit vector which points in the original azimuth direction. Assuming that $\boldsymbol{F}$ is the final, combined transformation matrix we obtain the relation

$$\cos \phi_{\mathrm{tr}} = \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} \boldsymbol{T} \boldsymbol{F} \boldsymbol{T}^{-1} \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} \right]. \tag{A.16}$$

The reflection matrices for the basis are

$$\boldsymbol{M}_0 = -\boldsymbol{M}_{90} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; \boldsymbol{M}_{30} = -\boldsymbol{M}_{120} = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}, \text{ and } \boldsymbol{M}_{60} = -\boldsymbol{M}_{150} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{A.17}$$

A.4 Derivation of SmeLu

In the following, we derive the functional form of the SMELU activation function:

$$\begin{aligned} A_{\text{SmeLu}}(x;\beta) &\propto \int_{\mathbb{R}} \mathrm{d}y\, \text{ReLu}(y)\text{Box}(2\beta, x-y) \\ &= \int_0^{\infty} \mathrm{d}y\, y\Theta(\beta+(x-y))\Theta(\beta-(x-y)) \\ &= \int_{\max(0,\beta-x)}^{\max(0,\beta+x)} \mathrm{d}y\, y = \frac{1}{2}\left[\max\left(0,(\beta+x)^2\right) - \max\left(0,(\beta-x)^2\right)\right]. \end{aligned}$$

A.5 Second moment of depth of shower maximum from simulation data

Using the results in Sec. 5.4.2, we compute the average value for different compositions comprised of events originating from {p, He, O, Fe}. The mean value is then

$$\langle x \rangle = \sum_h w_h \langle x \rangle_h \,, \tag{A.18}$$

where w_h is the fraction of primary h and $\langle x \rangle_h$ is the average value of that primary. For the standard deviation we get

$$\sigma_x^2 = \sum_h w_h \sigma_{x,h}^2 + \sum_h w_h (\langle x \rangle_h - \langle x \rangle)^2, \tag{A.19}$$

where $\sigma_{x,h}$ is the standard deviation of the primary h.

A.6 Definition of the baseline architecture

The baseline architecture uses the following architecture:

- $\mathcal{AR}_{\text{i}}$: CNN, $n_f = 10$; RNN, $n_f = 16$
- $\mathcal{AR}_{\text{ii}}$: $n_s = 16$, $n_d = 4$
- $\mathcal{AR}_{\text{iii}}$: $d_f = 0.2$
- Training: $N_b = 64$, $\alpha = 0.0022$, $\lambda_B = 1$
- Other: $M_s = 5$, $L_t = 120$

We listed the CNN- and RNN-based TFE since we use both at different points of Sec. 7.1.

A.7 Definition of zenith-energy cut

The NNs trained using the setups described in Sec. 7.2 show primary dependent biases for low energies and low zenith angles in simulation data. The NNs cannot accurately predict the chosen targets in these regions due to the reduced shower footprint sizes. We investigate if these regions of worse performance influence the NN predictions on measurements by defining cuts that allow us to remove these parts of the phase space.

We estimate the cuts from the training set of the simulation library defined in Row 5.5.f following a similar procedure as discussed in [P:104]. We use a different hadronic model to ensure that we have enough statistics. In contrast to [P:104], we derive the cuts using the

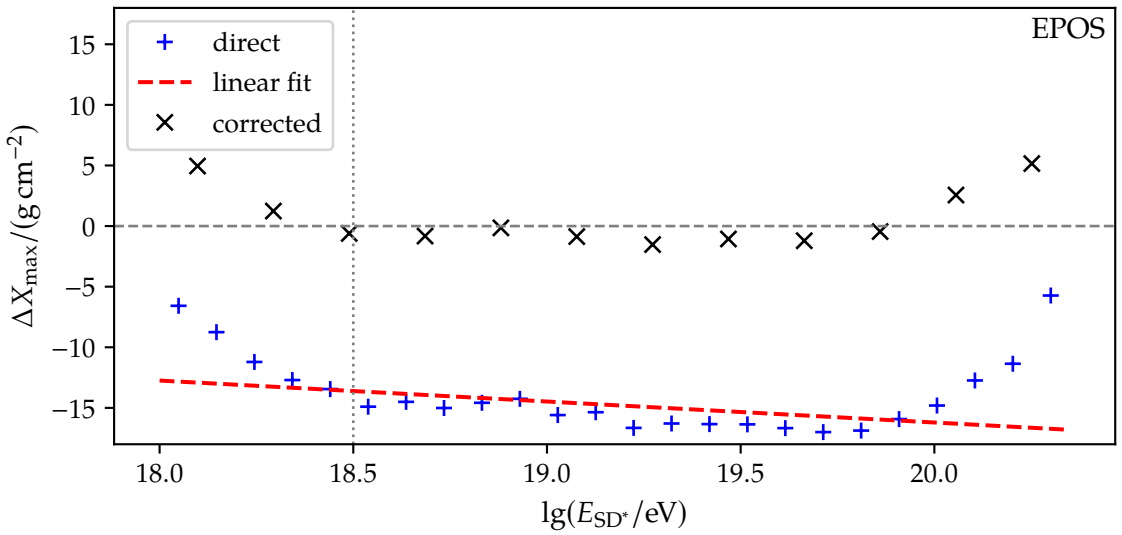


Figure A.1: Bias of the NN predictions if applied on a data set simulated with the hadronic interaction model EPOS-LHC as a function of the shifted reconstructed energy (see Eq. (7.4)). The blue plusses show the direct prediction and black crosses the corrected predictions. For the correction a linear function is fitted to direct predictions (dashed, red, line). The horizontal dashed line marks a bias of zero. The vertical dashed line marks the energy of full efficiency.

Table A.1: Fit parameters obtained in Appendix A.7. Row A.1.a refers to the shift of the predictions because the cuts have been estimated from a different hadronic interaction model than that which have been used for the training.

	name	fit function	p_0	p_1	p_2	shift
a	shift	Eq. (A.10)	$(-14.41\pm0.07)\,\mathrm{g/cm^2}$	$(-1.73\pm0.12)\,\mathrm{g/cm^2}$	-	19.0
b	upper cut	Eq. (A.10)	-2.20 ± 0.58	0.15 ± 0.03	-	-
c	lower cut	Eq. (A.10)	112.04 ± 32.71	-11.45 ± 3.38	0.29 ± 0.09	-

shifted reconstructed SD energy E_{SD^*} (see Eq. (7.4)) and zenith angle θ_{SD} to include the effects of the composition bias and the reconstruction. Since we trained the NN on simulations based on the hadronic interaction model QGSJetII-04, we have to correct for the bias found in Sec. 7.3.3.A. We model the bias with a linear function and subtract it from the predictions (see Fig. A.1).

We evaluate the cuts in the $\lg E - \sin^2\theta$ plane since the simulation data set is distributed uniformly over $[0, \sin^2 60°]\times[18.0, 20.2]$. Contrary to [P:104], we define the cut via the proton-iron bias $\Delta_{\text{p-Fe}}$ (see Fig. A.2) instead of the maximum absolute bias of proton and iron events. In this way, we account for the negative bias of the iron events. We have chosen $10\,\mathrm{g/cm^2}$ as the threshold value making the cut slightly more strict than in [P:104].

To estimate the region where the proton-iron bias is below the threshold, we use the following procedure. We bin the data set in the $\lg E - \sin^2\theta$ plane. For each energy bin, we determine the first upper and lower $\sin^2\theta$ that fulfill the condition. Therefore, we obtain an upper and lower cut. We fit a linear function to the upper bin values and a quadratic function to the lower bin values (see Fig. A.3). Only events whose reconstructed energy and zenith angle lie between both lines are inside the high-quality phase space. We discard all other events from the analyzed data set. We tabulated all fit results in Table A.1.

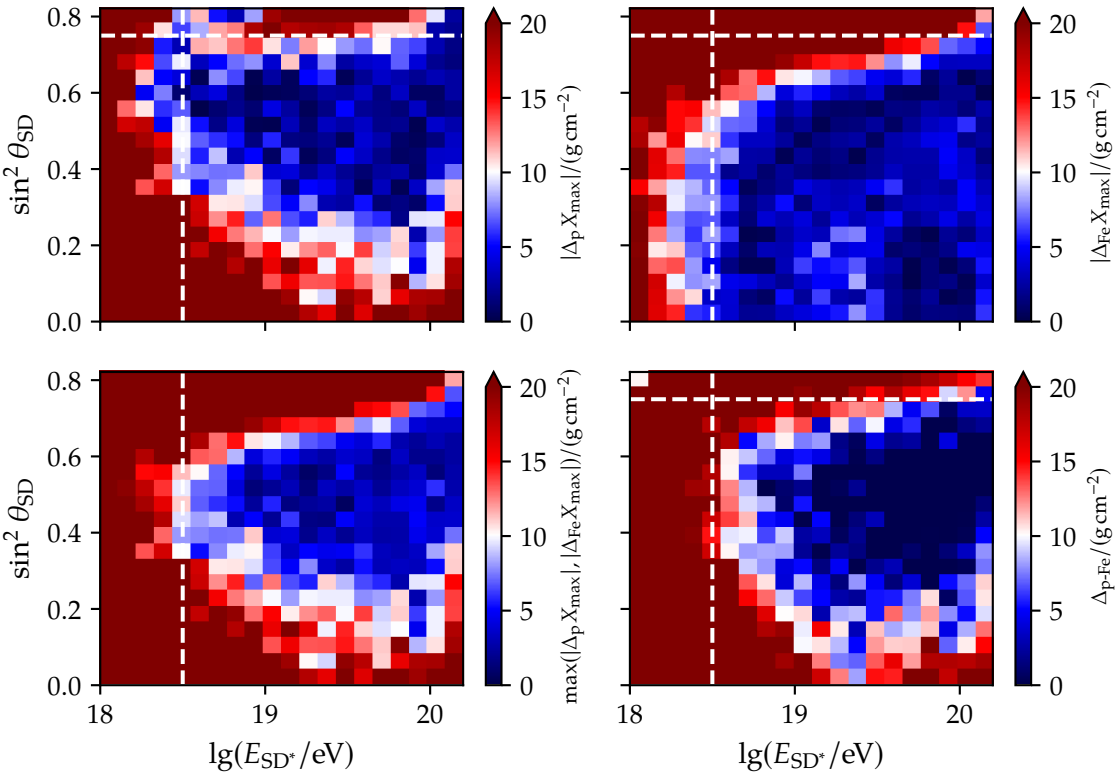


Figure A.2: Different bias estimates of the corrected predictions (see Fig. A.1) binned in the $\lg E - \sin^2\theta$ plane. The *top* panels show the absolute bias for proton (*left*) and iron (*right*) events. The *bottom left* panel depicts the maximum value of the biases in each of the *top* panels. The *bottom right* panel shows the proton-iron bias $\Delta_{\text{p-Fe}}$ for each bin. The white dashed lines mark the full efficiency boundary of the SD in the phase space.

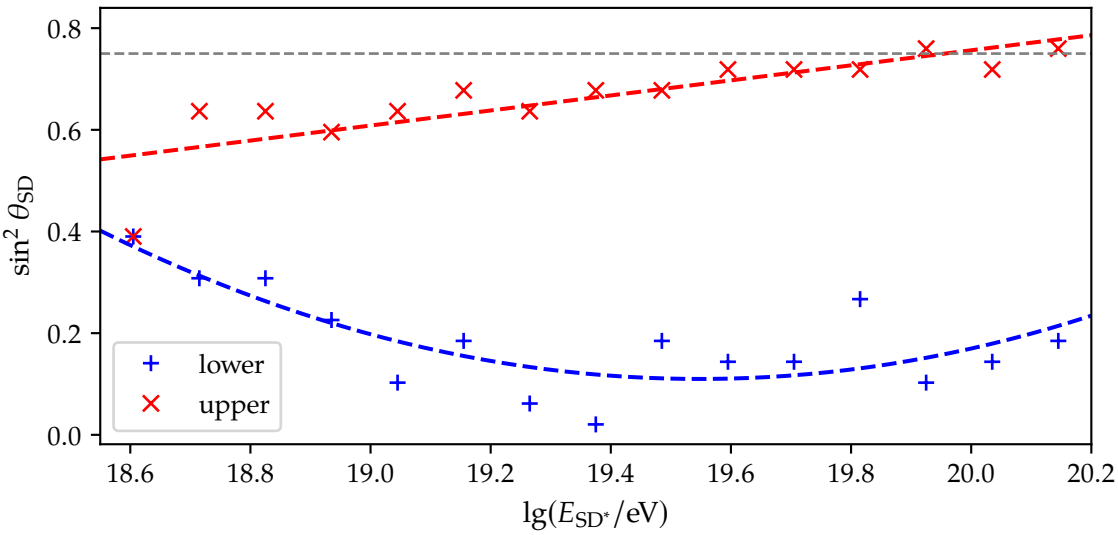


Figure A.3: Upper (red) and lower (blue) bound determined from the bins in Fig. A.2 with the method described in Appendix A.7. The color-coded dashed lines represent the fits to the corresponding $\sin\theta_{\mathrm{SD}}$ bins.

B Additional Content

> Some books are to be tasted, others to be swallowed, and some few to be chewed and digested.
>
> (Sir Francis Bacon)

B.1 Advanced non-standard layers

TF not only has a wide variety of building blocks to choose from, but also provides a big framework to create specialized layers and sub-models. Often these self-build layers and sub-models are special combinations of the basic layers in Sec. 4.2.3. This allows us to use the highly optimized algorithms of the underlying TF library. Nevertheless, most of these exhibit novel architectures that tackle different kind of problems in certain sub-fields of machine learning.

A common theme in recent years is to accommodate the special properties of input data via the general architecture of the NN. If the data follows some kind of symmetry, we want to exploit this as much as possible. In this way, the network does not need to find this symmetry during the training process stabilizing it and in the best case making the predictions more robust. This idea – of course – is very much an ordinary course of action if tackling physics problems.

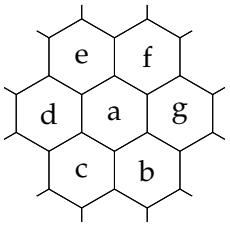
A Hexagonal convolution layer

The SD of Auger is arranged in a triangular grid (see Sec. 2.3.2). Hence, each of the grid stations has six neighbors. Directly encoding this setup in rectangular memory generates the problem that each pixel suddenly has eight neighbors. The filters of regular 2D-convolution layers (see Sec. 4.2.3.B) are rectangular. If we want to correlate spatial information of the encoded grid data via a convolution layer, we would suddenly correlate the information of non-neighboring stations.

Hexagonal convolution layers tackle this problem in a simple way. In general they are regular convolution layers with quadratic filters that have an odd filter size. However, in those filters, depending on the encoding, the parts that do not belong to a certain hexagonal grid size are zeroed. Consequentially, the stations not directly inside of the hexagon do not contribute to the calculation of Eq. (4.20). The number of masked pixels in the filter is $(2n + 1)^2 - (1 + 6n)$, where n is the number of crowns in the hexagon.

B Group equivariant convolutions

As discussed in Chapter 4, we have to design the neural network architecture to account for intrinsic symmetries and correlations of our physics data. Especially, exploiting symmetries reduce the effective phase space a neural network has to account for (see Sec. 5.3.3). A way of encoding symmetries into the neural network architecture are Group Equivariant Convolution (GEC) layers. GEC layers are a generalization of regular convolution layers [C:26] (see Sec. 4.2.3.B). If we know the transformation associated with a symmetry, we can construct a specialized GEC layer that is invariant under this transformation.



∅	e	f
d	a	g
c	b	∅

Figure B.1: Filter of a hexagonal convolution in terms of a regular convolution for direct encoding using integer grid positions. The ∅ represent the positions that are masked by zeroing the filter values. The letters represent the positions at which the convolution filter is used.

Let Λ_G be an arbitrary transformation, we want to construct a layer with the following property (see Eq. (4.21))

$$\Lambda_G y = \Lambda_G(x *_G f) = ([\Lambda_G x] *_G f), \tag{B.1}$$

where $*_G$ is the GEC. We define

$$(x *_G f)(g) = \sum_{h \in L} \sum_{k=1}^{K} x_k(h) f_k\left(g^{-1}h\right), \tag{B.2}$$

where

$$f_k\left(g^{-1}h\right) = \Lambda_G f_k(h) \tag{B.3}$$

is one of the filters and L is the group of the output of the previous layer. If there are no preceding GEC layers, L is usually $\mathbb{Z}_n$ corresponding to the n-dimensional integer indexes of the outputs of the preceding layer. Otherwise L is the group G itself. Since this concept is rather abstract, we have provided an example for a mirror transformation in Appendix B.2.

By analyzing Eq. (B.3), we are able to relate regular convolutions with the GEC. Comparing Eq. (B.2) to a regular convolution, we can store our K_l GEC filters in the multidimensional array F_l that satisfies

$$O_D F_l = K_l \times S_l \times K_{l-1} \times S_{l-1} \times n_f, \tag{B.4}$$

where n_f is the total filter size, K_{l-1} is the number of channels in the preceding layer $l-1$, and S_l and S_{l-1} are the group symmetry numbers of layer l and $l-1$, respectively. The group symmetry number is the number of unique transformations, e.g., four for all possible rotations in a rectangular grid. Due to the cyclic nature of the transformations, e.g., if Λ_M is the mirror operator $\Lambda_M \Lambda_M$ is the identity operator, some of the weights in Eq. (B.4) are shared. To find the indices of these shared weights, we define the invertible map $g(s, u, \ldots)$ which relates the indices $\hat{s}, \hat{\boldsymbol{u}}$ to the indices $s', s, \boldsymbol{u}$ via

$$\hat{s}, \hat{\boldsymbol{u}} = g^{-1}\left(g(s', \mathbf{0})^{-1} g(s, \boldsymbol{u} \ldots)\right). \tag{B.5}$$

For example, in the one-dimensional case described in Appendix B.2, we could use the matrix representations

$$g(s, u) = \begin{pmatrix} (-1)^s & u \\ 0 & 1 \end{pmatrix} \rightarrow g(s', 0)^{-1} g(s, u) = \begin{pmatrix} (-1)^{s+s'} & (-1)^{s'} u \\ 0 & 1 \end{pmatrix} \tag{B.6}$$

to compute the conversion above. Effectively, we can define a tensor E_l of the size

$$O_D F_l = K_l \times K_{l-1} \times S_{l-1} \times n_f, \tag{B.7}$$

that contains only unique filters and relate it with Eq. (B.4) via

$$F_l\left[i, s', j, s, \boldsymbol{u}\right] = E_l\left[i, j, \hat{s}, \hat{\boldsymbol{u}}\right]. \tag{B.8}$$

Since there is a direct relation between the filters of a regular convolution and the filters of a GEC convolution, we can use regular convolutions to construct the GEC. For the triangular grid we can define a GEC that is invariant under rotations of 60°. We encode the triangular grid by using axial coordinates (see Sec. 5.3.3.A) and use hexagonal filters as described in **??**. We denote Denoting Λ_R as the transformation operator and using Eq. (B.2) we obtain

$$y = (x *_G f^i)(s', \boldsymbol{u}) = \sum_{k=1}^{K_f} \sum_{h \in \mathbb{Z}_2} x_k\left(h\right) f_k^i\left(g^{-1}h\right) \tag{B.9}$$

$$= \sum_{k=1}^{K_f} \sum_{\boldsymbol{v}} x_k\left(\boldsymbol{v}\right) \Lambda_T \Lambda_R(s') f_k^i(\boldsymbol{v}), \tag{B.10}$$

and for subsequent layers

$$z = (y *_G q^i)(s', \boldsymbol{u}) = \sum_{k=1}^{K_q} \sum_{\boldsymbol{v}} \sum_{s} y_k(s, \boldsymbol{v}) \Lambda_T \Lambda_R q_k^i\left(s, \boldsymbol{v}\right) \tag{B.11}$$

$$= \sum_{k=1}^{K_q} \sum_{\boldsymbol{v}} \sum_{s} y_k(s, \boldsymbol{v}) \Lambda_T q_k^i\left(s + s', \Lambda_R \boldsymbol{v}\right), \tag{B.12}$$

where $\boldsymbol{v}$ and $\boldsymbol{u}$ are the integer coordinates, s is an integer coordinate associated with the group element, and f^i and q^i are the ith filters of both GEC layers. Both Eq. (B.10) and Eq. (B.12) have the form of a regular convolution. Detailing how this index ordering and transformation is performed, is outside of the scope of this thesis.

B.2 Example for group equivariant convolutions

For sake of simplicity we only use one feature map, a single filter, write $*$ instead of $*_G$, and remain in one dimension. Let x be a "feature array" of length three and f_1 and f_2 be two set of filters of length two. Using the following representation

$$x = \boxed{}\boxed{}\boxed{} \qquad f_1/f_2 = \boxed{}\boxed{} \tag{B.13}$$

we want to find the two GEC layers that transform

$$x \xrightarrow[f_1]{\text{GEC}} y \xrightarrow[f_2]{\text{GEC}} z\ . \tag{B.14}$$

We assume that our input is invariant under mirroring. Therefore, we choose the transformation $\Lambda_M x = -x$ as the starting point of the construction of the corresponding GEC layer.

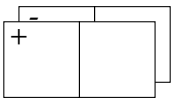
In the first step of Eq. (B.14) we have to transform from "real space" to the "group space" (see Eq. (B.2)). Therefore, L is equal to $\mathbb{Z}$.

$$y\left(g\right) \equiv \left[x * f_1\right](m, u) = \sum_{v \in \mathbb{Z}} x(v) f_1\left(g^{-1}v\right) = \sum_{v \in \mathbb{Z}} f(v)\left[\Lambda_T(u)\Lambda_M(m)\psi\left(v\right)\right], \tag{B.15}$$

where Λ_T is the translation operator (see Eq. (4.21)). Note that $\Lambda_T \Lambda_M = \Lambda_G$. Since we have two states in our transformation we get two feature arrays: y^+ and y^-. The second part of Eq. (B.14) can be written as

$$z(m, x) = [y * f_2](m, u) = \sum_{h \in G} y(h) [\Lambda_G f_2(h)] . \tag{B.16}$$

Therefore, we need a separate filter for each group element

$f_2 =$ 

. (B.17)

Then, we can expand Eq. (B.16)

$$z(\pm, u) = \sum_{v \in \mathbb{Z}} \left(y^+(v) [\Lambda_T \Lambda_M(\pm) f_2(+, v)] + y^-(v) [\Lambda_T \Lambda_M(\pm) f_2(-, v)] \right) \tag{B.18}$$

Now, we have to compute$\Lambda_M f_2$. Since, we have two states we can do this

$$\Lambda_M(\pm) f_2(\pm, v) = \phi(n \pm 1, \Lambda_M(\pm) v) = \phi(n \pm 1, \pm y) \tag{B.19}$$

$$= \begin{cases} \Lambda_M(\pm) f_2((1 \pm 1) \mod 2, \pm v) \\ \Lambda_M(\pm) f_2((0 \pm 1) \mod 2, \pm v) \end{cases} \tag{B.20}$$

$$= f_2(\mp, \pm v) , \tag{B.21}$$

where n is the number of inversions. We have only to pre-compute these filter maps for each layer.

B.3 Additional studies on the variation of the baseline architecture

In this section, we extend the study on the variation of hyperparameters related to the network architecture which has been done in Sec. 7.1.3.

Number of extracted features In Fig. B.2 and Fig. B.3, we show the precision and accuracy for the predictions for CNN-based NNs using different n_f values. Effectively, these tests if the choice of n_f is high enough. If not, we would see an improvement of precision or accuracy, since in the traces are more features than could be represented in the n_f channel dimensions. This seems to be not the case.

Depth and filter size of SCA In Fig. B.4 and Fig. B.5, we show the precision and accuracy for the predictions for CNN-based NNs using different n_d and n_s values. Even though the result of $n_d = 4$ and $n_s = 6$ show a slightly better precision than that of the baseline value $n_d = 4$ and $n_s = 16$, it is small enough to be a fluctuation. Moreover, since the amount of training data for the optimized networks are in general larger, we do not directly want to reduce the number of parameters of the network.

Variation of batch size for RNN-based sub-network In contrast to Sec. 7.1.4.A, the results for the RNN-based TFE are reversed (see Fig. B.6). Reducing the trace size seems to improve the precision of the predictions.

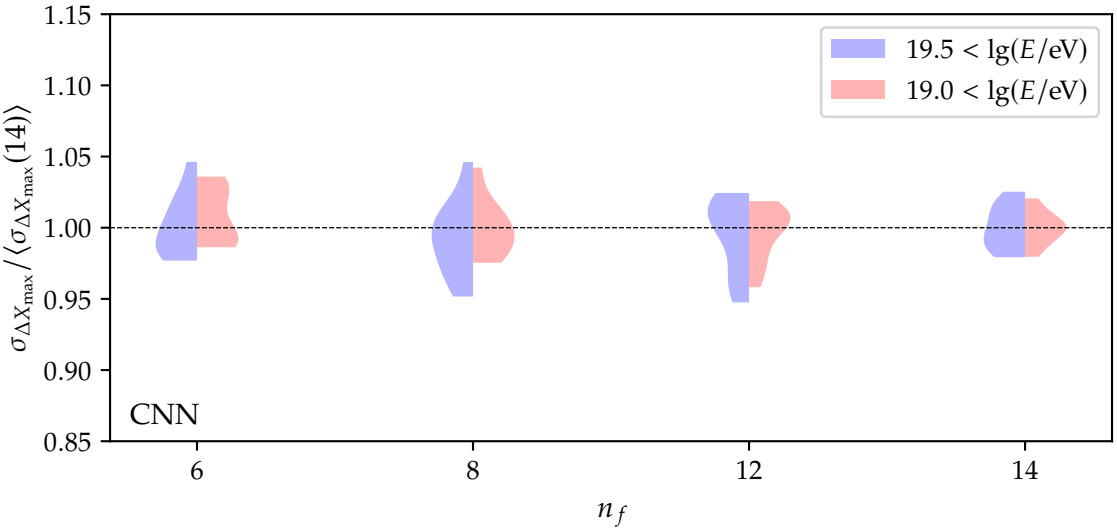


Figure B.2: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained with different values of n_f normalized to the average precision of the ensemble using $n_f = 14$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

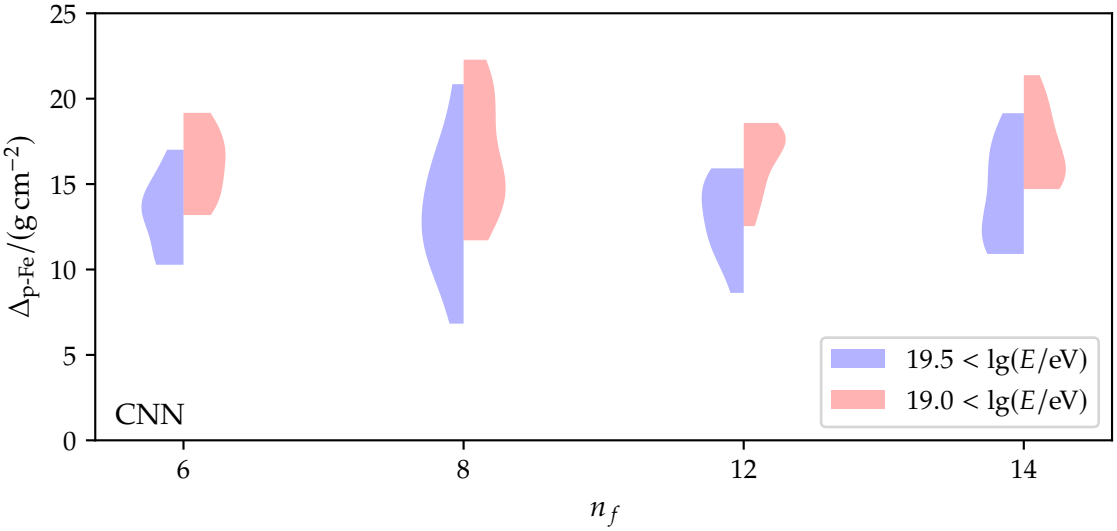


Figure B.3: Ensembles of the proton-iron bias $\Delta_{p\text{-}Fe}$ for NNs trained with different values of n_f for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

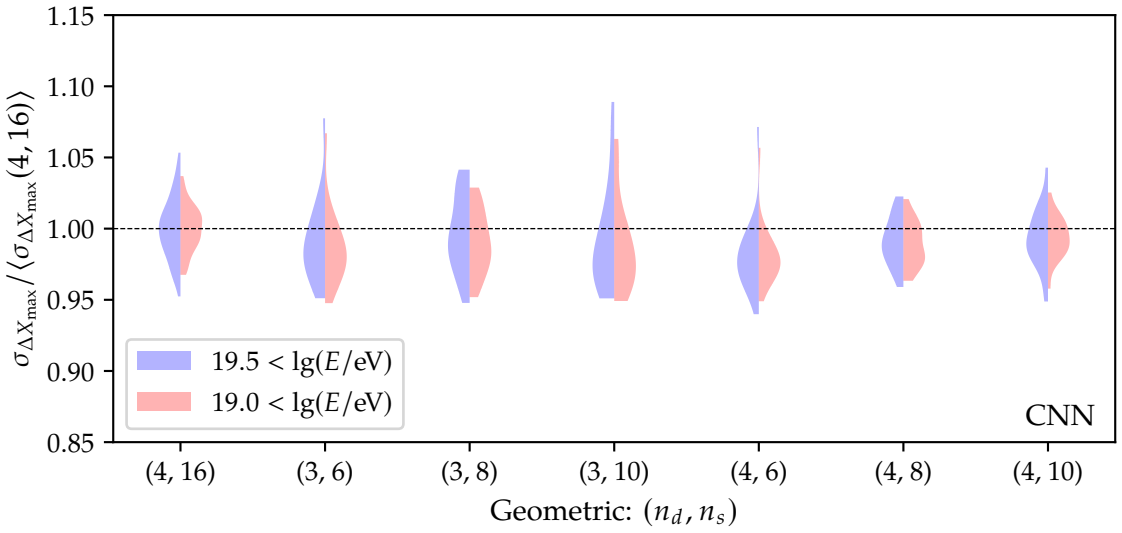


Figure B.4: Ensembles of the precision $\sigma_{\Delta X_{max}}$ for NNs trained with different values of n_d and n_s normalized to the average precision of the ensemble using $n_d = 4$ and $n_d = 16$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

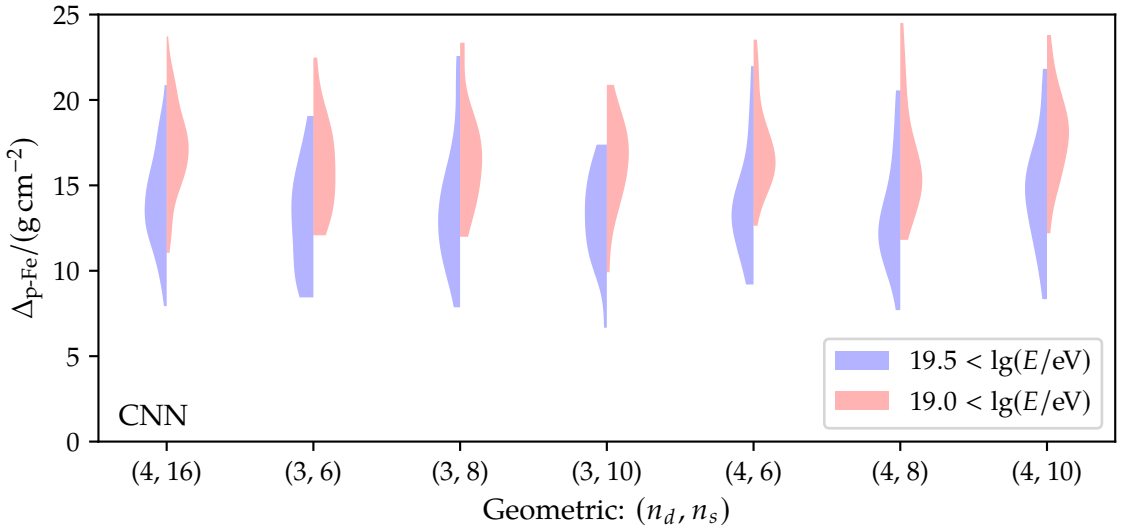


Figure B.5: Ensembles of the proton-iron bias $\Delta_{p\text{-}Fe}$ for NNs trained with different values of n_d and n_s for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

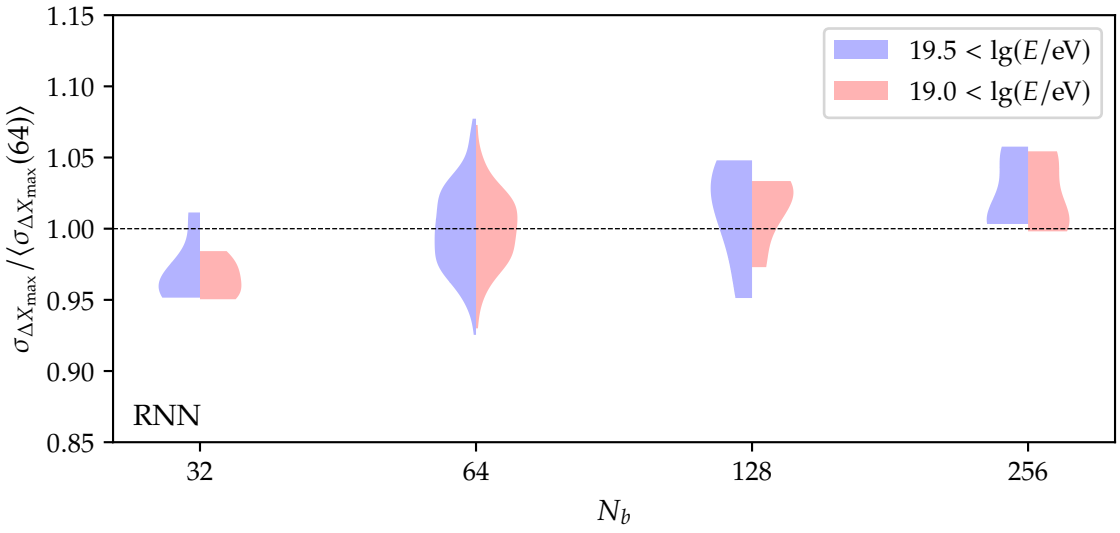


Figure B.6: Ensembles of the precision $\sigma_{\Delta X_{\max}}$ for NNs trained with different values of the batch size N_b normalized to the average precision of the ensemble using $N_b = 64$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red). In contrast to Fig. 7.12, the NNs trained for this study use the RNN-based TFE.

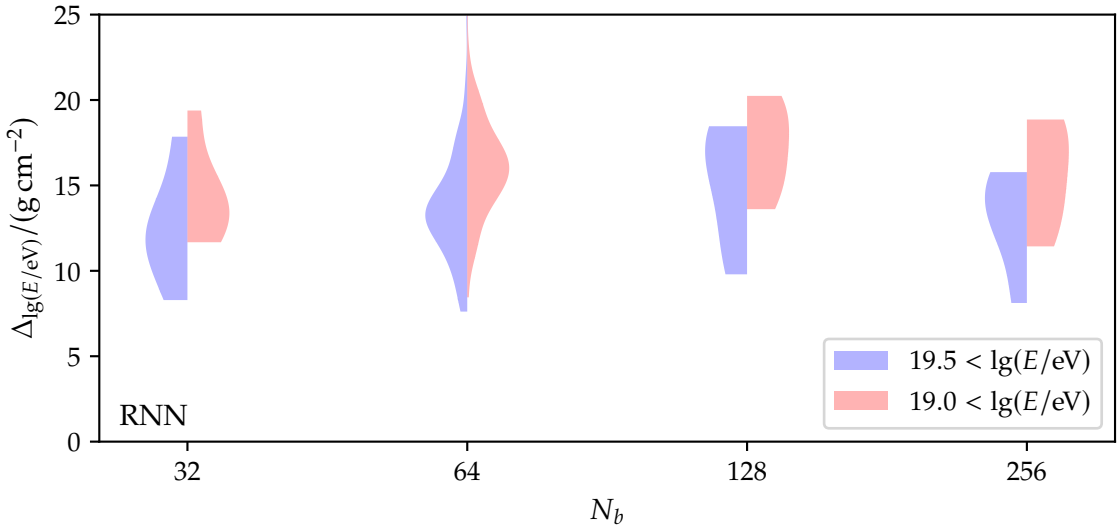


Figure B.7: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with different values of the batch size N_b for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red). In contrast to Fig. 7.13, the NNs trained for this study use the RNN-based TFE.

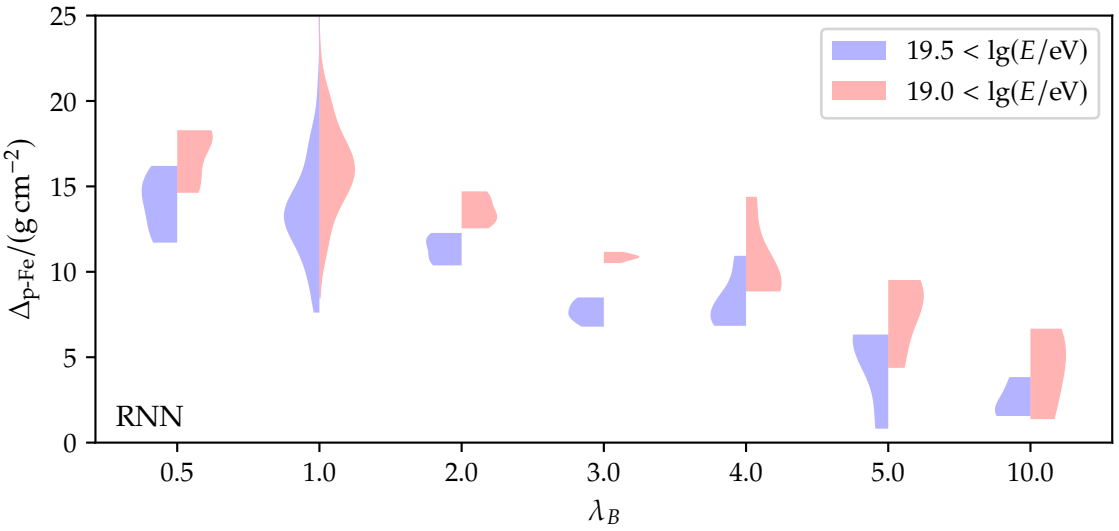


Figure B.8: Ensembles of the proton-iron bias $\Delta_{\text{p-Fe}}$ for NNs trained with different values of λ_B for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

Variation of the importance of the bias-penalizing term To ensure the correct implementation and demonstrate the effect of the bias-penalizing term in the used loss function, we show the effect of varying λ_B. This variation changes the relative importance of the bias during the minimization of the loss (see Eq. (4.17)). Due to a typo in one of our scripts we did this study for the RNN based architecture. As expected, increasing λ_B reduces the proton-iron bias considerably (see Fig. B.8). This can be considered as a proof of concept. Unfortunately, introducing this loss term also reduces the precision of the networks (see Fig. B.9). This effect is more dominant for higher energies since the blue distributions shift more upwards than the red ones.

B.4 Effect of model averaging

Due to the randomness involved in our network training (see Sec. 7.3.2), it is reasonable to check if we can use multiple models to stabilize our predictions. We do this by taking the average prediction of multiple models trained on the same data set. This averaging is analogous to a poor-man[1] version of bagging [C:27].

In Fig. B.10, we show the σ values for 50 draws of n models from our CNN models of Sec. 7.1.3.A. For a higher number of models drawn from the underlying 50, we obtain a better result. Moreover, also the width of the distributions shrinks. The more models we take for the averaging process, the less is the improvement of the precision. Note that for larger draw sizes, this is also an effect of the small underlying data set. However, we think that – at least for smaller draw sizes – this averaging reduces the effect of outliers reducing the variance and, in turn, the precision.

Even though this process improves the precision of the predictions of our composite models, it has a negative impact on the inter-primary spread. In Fig. B.11, we show the same approach for the distribution of $\Delta_{19.0}$ and $\Delta_{19.5}$ values. The averaging process centers the bias between proton and iron primaries. Therefore, we have to carefully evaluate if and – if yes – how many models we want to use for the model averaging.

[1] Instead of real bootstrapping we take the same data set over and over.

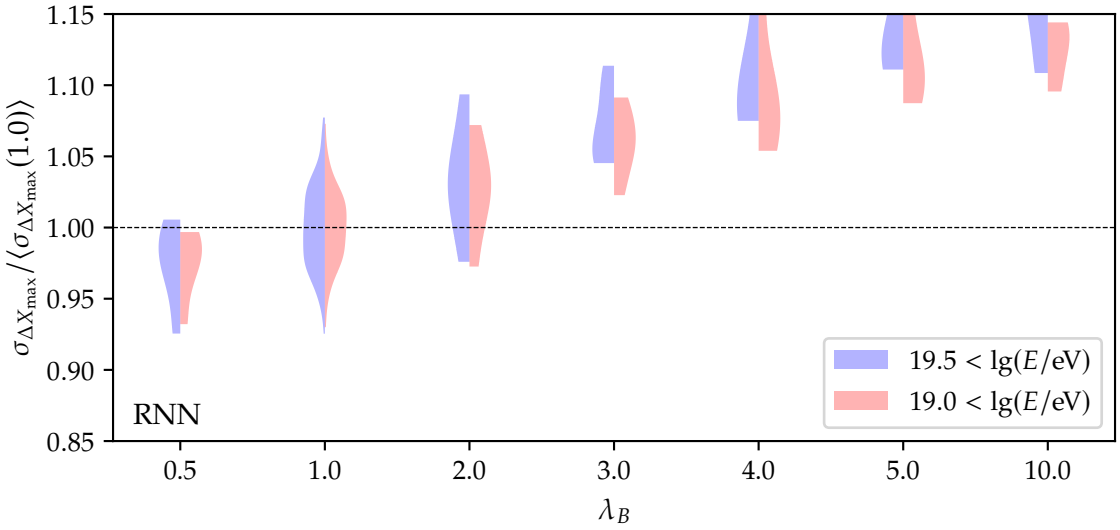


Figure B.9: Ensembles of the precision $\sigma_{\Delta X_{\max}}$ for NNs trained with different values of λ_B normalized to the average precision of the ensemble using $\lambda_B = 1.0$ for events with an energy above $10^{19.5}$ eV (blue) and $10^{19.0}$ eV (red).

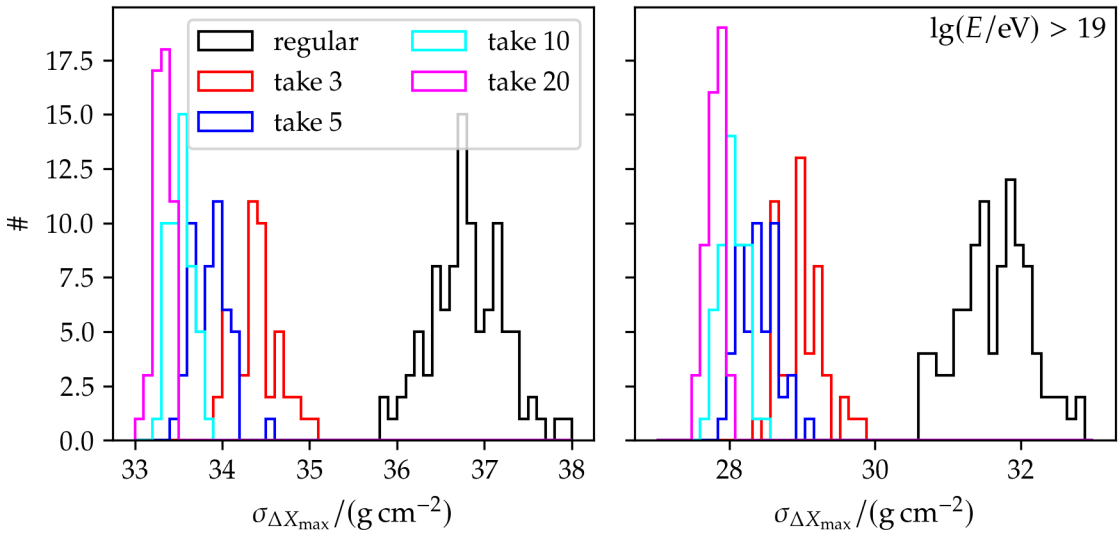


Figure B.10: Distribution of precision of 50 models trained on the setup discussed in Appendix B.4 compared to that of model averaging evaluated on the global standard deviation (*left*) and the standard deviation at high energies (*right*). For each average predictions we draw n unique models from the initial set of 50. The more models we take the better the average prediction gets. However, the effect decreases with the number of models.

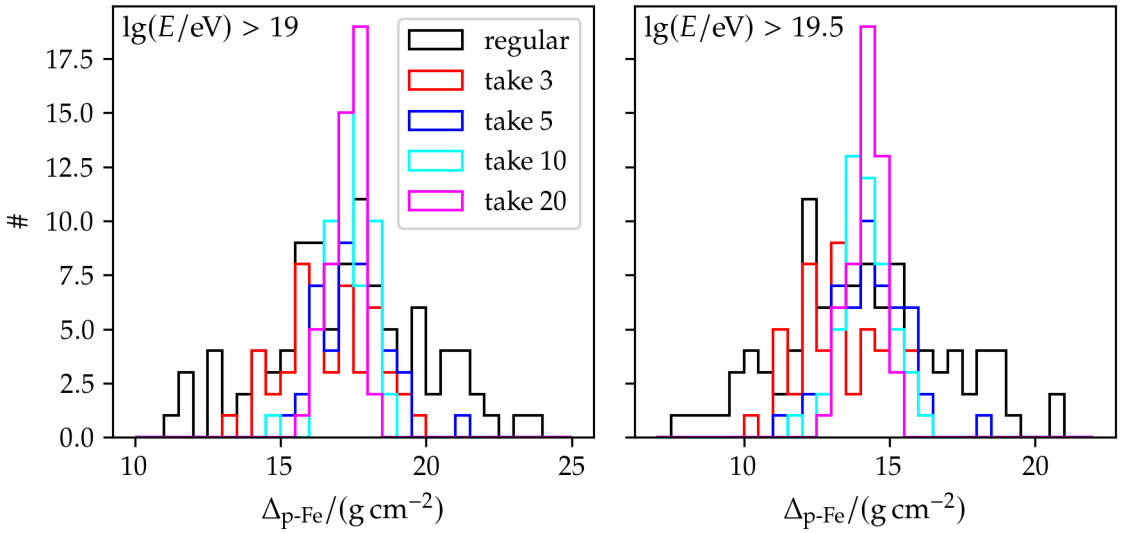


Figure B.11: Distribution of proton-iron bias of 50 models trained on the setup discussed in the beginning of the section compared to that of model averaging evaluated on events with an energy above 10^{19} eV (*left*) and an energy above $10^{19.5}$ eV (*right*). The more models we take into account the less is the possibility for models with a low Δ value.

B.5 Estimation of noise due to input data

We train eight models for six different data sets. The data is drawn from our global shower library and follow more-or-less the same distribution and compositions as that defined in Row 5.5.e. For this part we relaxed the conditions even more:

1. all models share the same architecture
2. all models give reasonable predictions without post-processing
3. all models are trained on one GPU

Since we have drawn the six data set from our base data set which is much larger, we would have to find the intersection of all data sets to compare at the spread between all models. However, the intersection between all test data sets has zero elements. Therefore, we take the intersections of all possible pairs of data sets to approximate the distribution of ς (see Fig. B.12). We find the same deviations from the distribution of Sec. 7.3.2.A as in Sec. 7.3.2.B. As expected the effects pile up. Even though we underestimate the importance of the choice of input data it still yields a larger average spread. This is also seen in σ. We obtain

$$\sigma_{\text{input-variation}} = 0.40\,\text{g/cm}^2 \tag{B.22}$$

for these conditions. Therefore, there is not a large difference using similar training and test distributions.

B.6 Direct energy estimator

The SD energy predictor is derived from measurements. Due to the differences between measurements and simulation data, it shows a bias if applied on simulation data. To check the energy prediction of our NN model, we could, therefore, create our own model based on the

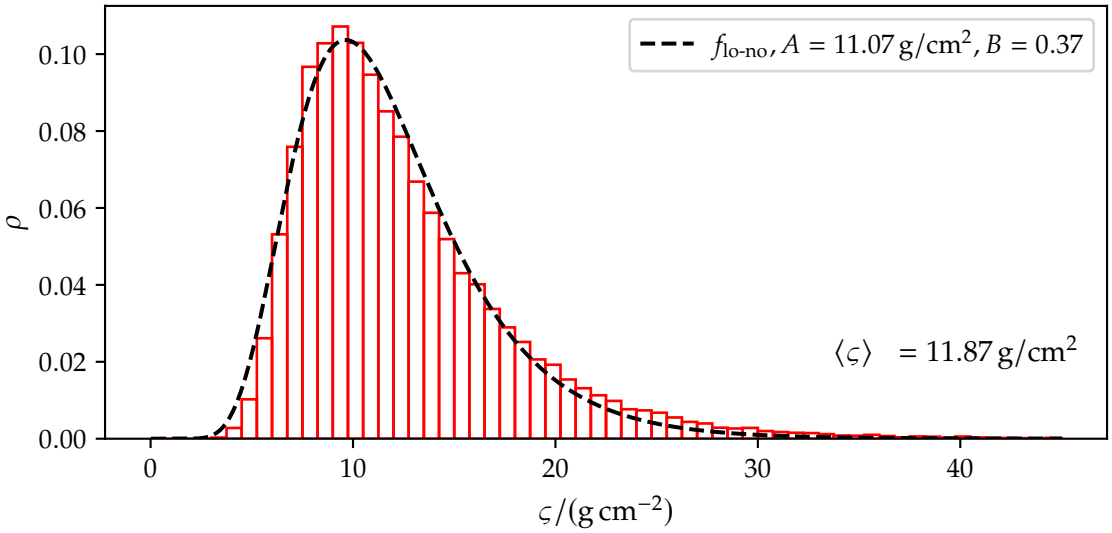


Figure B.12: Distribution of ς for models created according to conditions defined in Appendix B.5.

Table B.1: Fit parameters for the DEC defined in Eq. (B.23) fitted on UB data. We find that both UB-QGSJetII-04 and -EPOS-LHC exhibit similar energy calibration parameters but have slightly different attenuations.

Model	A	B	a	b	c
QGSJetII-04	17.3038(3)	1.0742(1)	1.6537(116)	−1.5716(32)	−1.0937(12)
EPOS-LHC	17.2621(3)	1.0767(1)	1.5358(117)	−1.6927(32)	−1.0499(12)

CIC procedure in Eq. (3.10). Since, our simulation data is evenly distributed in logarithmic energy and $\sin^2\theta$ we perform a least square fit to

$$\lg E_{\text{DEC}} = A + B \lg(S_{1000}/\text{VEM}) - \lg f_{\text{CIC}}(\cos^2\theta - \cos^2\theta_{\text{ref}}; a, b, c), \tag{B.23}$$

where A, B, a, b, and c are fit parameter. Therefore, we fit the CIC and the energy scale directly. We call this the Direct Energy Calibration (DEC). We have summarized the fit results in Table B.1.

In Fig. B.13, we compare the model obtained by a direct fit with the SD standard reconstruction of the energy. We shift the predictions of the latter model by the average global mean difference between mean logarithmic energies to align our distributions. Even though the DEC is slighly asymmetric, it performs better on a global level. Nevertheless, since both models behave approximately the same we think that the fit of DEC is reasonable. Moreover, such a model gives us the opportunity to find out how to perform the transition from simulations to real data.

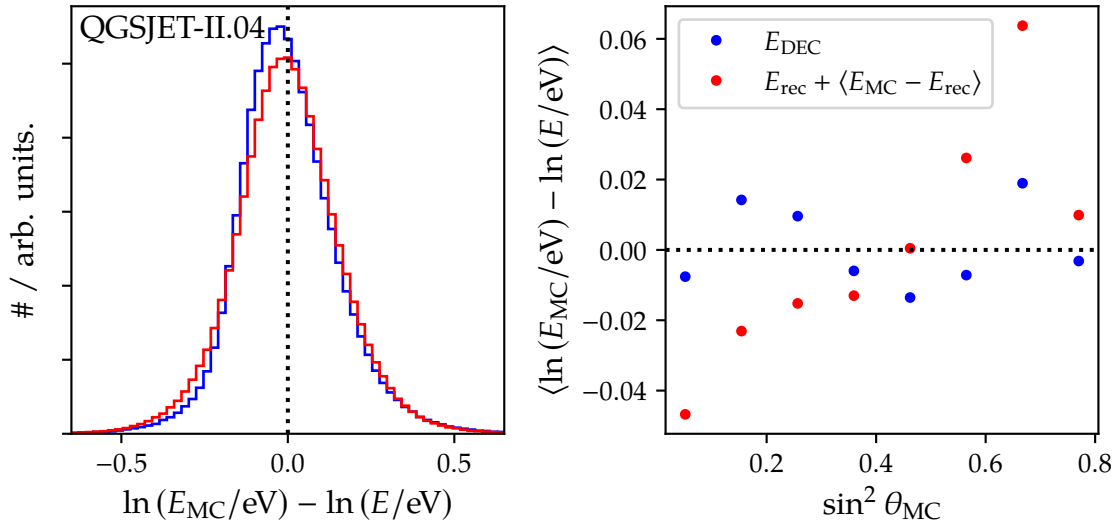


Figure B.13: Direct energy calibration method vs shifted SD energy estimate. Even though the direct method has a clear asymmetry it performs slighly better on MC data. Since this is only a reference model and the precision is good enough, we allow for this slight inaccuracy.

C Specifications and Snippets

> What is important is to spread confusion, not eliminate it.
>
> (Salvador Dalí)

C.1 GPU machine specifications

All networks in this thesis have been trained on two machines with the same configuration. Both machines have an Intel(R) Xeon(R) Gold 5122 CPU with a clock rate of 3.60 GHz using 125 GB of random-access memory. Each machine has access to two NVIDIA Tesla V100-32GB GPU and one NVIDIA Tesla V100S-32GB GPU. The machines are accessible via `HTCondor`.

C.2 Snippets

C.2.1 SD cuts

```
ADST cuts version: 1.0

# SD cuts
!lightning

minRecLevel           3   # see SdRecLevel.h
maxZenithSD          60.  # maximum zenith angle [deg.]
T4Trigger             2
T5Trigger             2   # 1: 5T5 post, 2: 6T5 prior, 3: T5Has

minLgEnergySD  18

badPeriodsRejectionFromFile
```

C.2.2 *Golden Hybrid* cuts

```
adst cuts version: 1.0

#==== reject laser events
!isCLF
!isXLF

#==== keep either CO/HEAT or HECO
heatOrientationUp
eyeCut 001111

#==== hardware status
badFDPeriodRejection
minMeanPixelRMSMergedEyes {  params: 17 6 110000 nMinusOne: 100  0 100 }
minMeanPixelRMSSimpleEyes {  params: 17 011111 nMinusOne: 100  0 100 }
!badPixels                 1
good10MHzCorrection
```

```
#==== atmosphere
hasMieDatabase
maxVAOD 0.1
cloudCutXmaxPRD14 { params: 1 nMinusOne: 21  -10.5 10.5 }

#==== full hybrid geometry
hybridTankTrigger        2
maxCoreTankDist 1500
maxZenithFD              90
minLgEnergyFD            1e-20
skipSaturated
minPBrass           0.9
maxPBrassProtonIronDiff  0.05

#==== FOV cuts
FidFOVICRC13 40 20

#==== quality cuts
xMaxObsInExpectedFOV { params: 40 20 }
maxDepthHole         20.
profileChi2Sigma   {  params:  3 -1.1 nMinusOne: 400  -20 20 }
depthTrackLength    200
xMaxError               40.0
energyTotError          0.12
```

C.2.3 Bins used for importance sampling

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20,
21, 23, 24, 25, 27, 28, 30, 31, 33, 34, 36, 37, 39, 40, 42,
44, 45, 47, 49, 51, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70,
72, 74, 77, 79, 81, 84, 86, 88, 91, 93, 96, 98, 101, 104, 106,
109, 112, 115, 118, 121, 124, 127, 130, 133, 136, 139, 143,
146, 149, 153, 156, 160, 164, 167, 171, 175, 179, 183, 187,
191, 195, 199, 204, 208, 213, 217, 222, 226, 231, 236, 241,
246, 251, 256, 262, 267, 272, 278, 284, 289, 295, 301, 307,
313, 319, 326, 332, 339, 345, 352, 359
```

D Additional Material

Abandon All Hope, Ye Who Enter Here.

(Divine Comedy, Dante Alighieri)

D.1 Tabulated data

Table D.1: Summary of the parametrization parameters for the systematic uncertainties of X_{max}, R_{μ}, and $\ln A$ due to the high-energy hadronic interaction and unknown composition (see Sec. 7.3.3). The last column is the shift that is applied on the independent variable which is in this case the logarithmic energy. The pattern of the rows is as follows. The first two rows for each of the observables list the fit parameters of the shift and the linear function corresponding to parametrizations used for the biases of the high-energy hadronic interaction model (see Sec. 7.3.3.A. The next two rows show the fit parameters of the proton and iron bias. The last two two rows for each of the observables are the parameters corresponding to the upper and lower uncertainty of the standard deviation due to the unknown composition (see Sec. 7.3.3.B.

	name	fit function	p_0	p_1	p_2	shift
	X_{max}	all coefficients in g/cm^2				
a		Eq. (A.10)	−13.929±0.097	-	-	-
b		Eq. (A.10)	2.654±0.275	−1.042±0.458	-	19.0
c		Eq. (A.13)	−0.999±0.448	4.891±0.668	−2.575±0.296	19.0
d		Eq. (A.13)	−1.515±0.338	−5.064±0.499	−2.469±0.210	19.0
e		Eq. (A.13)	0.154±0.288	3.177±0.439	−2.685±0.304	19.0
f		Eq. (A.10)	−2.097±0.126	-	-	-
	R_{μ}					
g		Eq. (A.10)	0.030±0.001	-	-	-
h		Eq. (A.10)	0.012±0.001	−0.009±0.002	-	19.0
i		Eq. (A.10)	0.039±0.004	−0.051±0.002	-	19.0
j		Eq. (A.10)	−0.037±0.005	0.064±0.003	-	19.0
k		Eq. (A.10)	−0.025±0.003	0.044±0.002	-	19.0
l		Eq. (A.10)	−0.019±0.001	-	-	-
	$\ln A$					
m		Eq. (A.10)	0.484±0.004	-	-	-
n		Eq. (A.10)	0.102±0.007	−0.161±0.020	0.080±0.025	19.0
o		Eq. (A.10)	−0.790±0.048	0.062±0.080	-	19.0
p		Eq. (A.10)	0.718±0.028	−0.512±0.046	-	19.0
q		Eq. (A.10)	0.286±0.025	−0.197±0.046	-	19.0
r		Eq. (A.10)	−1.117±0.009	-	-	-

Table D.2: Summary of the fit parameters for the corrections of the unphysical dependences described in Sec. 7.3.3. The penultimate column is the shift that is applied on the independent variable. The last column indicates if we correct for the dependency.

	name	fit function	p_0	p_1	p_2	p_3	shift	used?
	$X_{\max}$	all coefficients in g/cm^2, except of those with marked by $*$						
a	ϕ_{SD}	Eq. (A.11)	–0.001±0.122	0.264±0.173	$1^{\dagger *}$	1.906±0.656*	-	no
b	$\langle z_m \rangle$	Eq. (A.12)	–2.972±0.208	0.460±0.089	5.089±0.226*	–2.972±0.208*	-	yes
c	$\langle a_{\mathrm{p}} \rangle$	Eq. (A.10)	–10.793±0.192	–54.628±0.761	-	-	3.2	yes
d	seasonal	Eq. (A.11)	0.051±0.122	2.191±0.173	$\frac{2\pi}{365}^{\dagger *}$	0.488±0.079*	-	no
e	daily	Eq. (A.11)	0.005±0.122	1.060±0.173	$\frac{2\pi}{12}^{\dagger *}$	–0.327±0.163*	-	no
f	θ_{SD}	Eq. (A.13)	10.777±0.262	–0.564±0.059	5.931±0.144	-	-	yes
g	P	Eq. (A.10)	–0.584±0.149	0.592±0.037	–0.028±0.005	-	860	yes
	R_{μ}							
h	ϕ_{SD}	Eq. (A.11)	consistent with zero					no
i	$\langle z_m \rangle$	Eq. (A.12)	0.005±0.000	–0.009±0.000	0.326±0.133	–0.176±0.040	-	yes
j	C_0, C_1	Eq. (A.13)	–0.024±0.005	0.000±0.001	0.043±0.011	-	-	yes
k	$\langle a_{\mathrm{p}} \rangle$	Eq. (A.10)	0.034±0.000	0.170±0.001	-	-	3.2	yes
l	seasonal	Eq. (A.11)	below 3×10^{-3}					no
m	hour	Eq. (A.11)	below 2×10^{-3}					no
n	θ_{SD}	Eq. (A.13)	–0.098±0.002	0.040±0.002	2.134±0.053	-	-	yes
o	P	Eq. (A.10)‡	0.694±0.293	–0.869±0.072	0.049±0.010	-	860	yes
	$\ln A$							
p	ϕ_{SD}	Eq. (A.11)	below 5×10^{-3}					no
q	$\langle z_m \rangle$	Eq. (A.12)	0.084±0.005	–0.147±0.006	0.403±0.087	–0.072±0.011	-	yes
r	$\langle a_{\mathrm{p}} \rangle$	Eq. (A.10)	0.444±0.005	2.248±0.021	-	-	3.2	yes
s	seasonal	Eq. (A.11)	below 5×10^{-2}					no
t	daily	Eq. (A.11)	below 3×10^{-2}					no
u	θ_{SD}	Eq. (A.10)	–0.137±0.013	–0.233±0.150	–3.394±0.464	8.142±0.406	-	yes
v	P	Eq. (A.10)‡	10.592±4.126	–14.516±1.018	0.870±0.142	-	860	yes

† fixed to this value and not fitted, ‡ $p_n \times 10^3$ due to the smallness of the coefficients

D.2 Additional figures

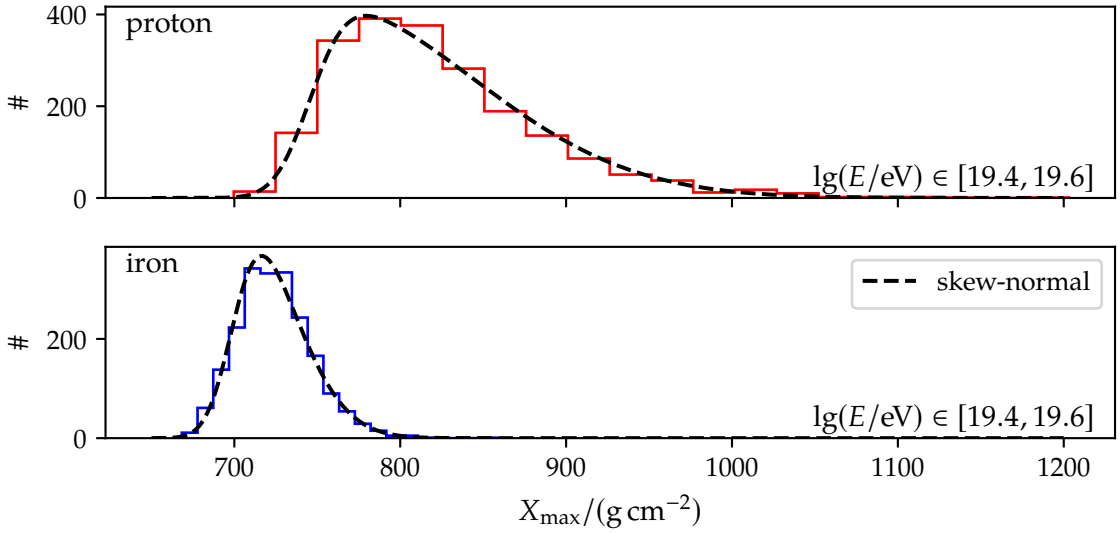


Figure D.1: Distribution of depth of the shower maximum X_{max} for proton (*top*) and iron (*bottom*) events in the logarithmic energy range [19.4, 19.6] simulated with the hadronic interaction model QGSJetII-04. The black dashed lines denote fits to the distributions using the skew normal distribution (see Eq. (5.13)).

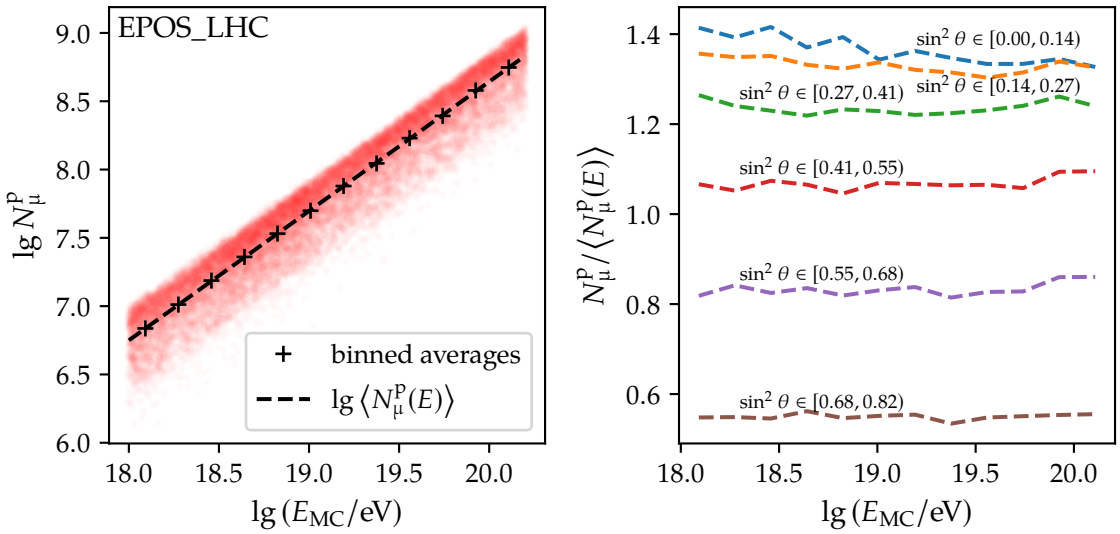


Figure D.2: Energy dependence of the number of muons (μ^+ and μ^-) for proton CORSIKA files simulated with EPOS-LHC of the Napoli library before (*left*) and after (*right*) removal of the energy dependency. We use the logarithmic N_μ^{P} due to the uniform distribution in energy. The energy dependence is in good agreement with a linear model. After the removal, the dependence is (almost) flat for different zenith intervals.

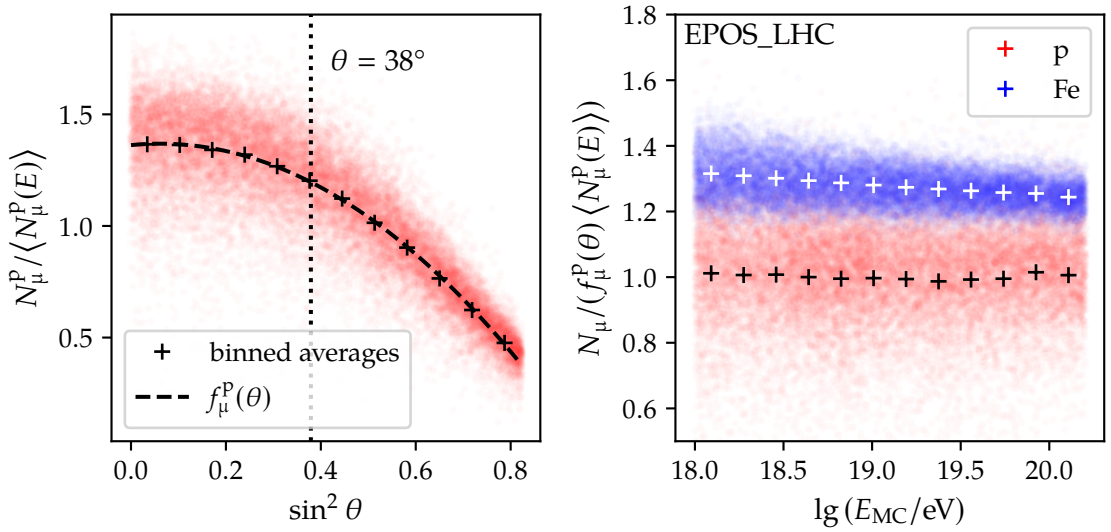


Figure D.3: Zenith dependence of the number of muons divided by the expected number at the energy for proton CORSIKA files simulated with EPOS-LHC of the Napoli library (left) and resulting R_μ for proton and iron of the same data set. Again, we find a good agreement with the proposed model in Eq. (5.20). Using the zenith dependence we can compute R_μ for the proton and iron showers. There is a visible separation between both events.

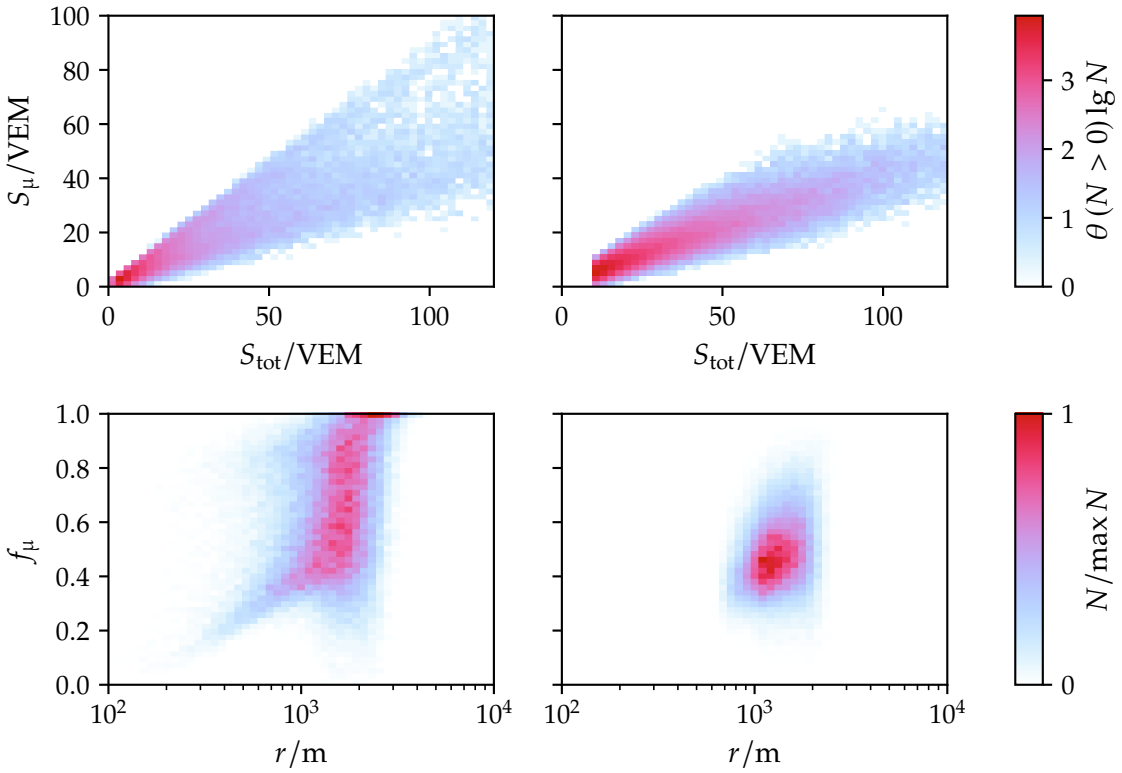


Figure D.4: Impact of the cuts defined in Sec. 6.2.1 on the distribution of S_μ and S_{tot} (*top*) and f_μ and r (*bottom*) when the sampling described in Sec. 6.2.1 is performed. The panels on the *left* side show the full QGSJ dataset (see Table 6.2) and the others show the distributions after the cutting process. We draw the same conclusions as in Fig. 6.8. The sampling has almost no effect on the distribution cuts.

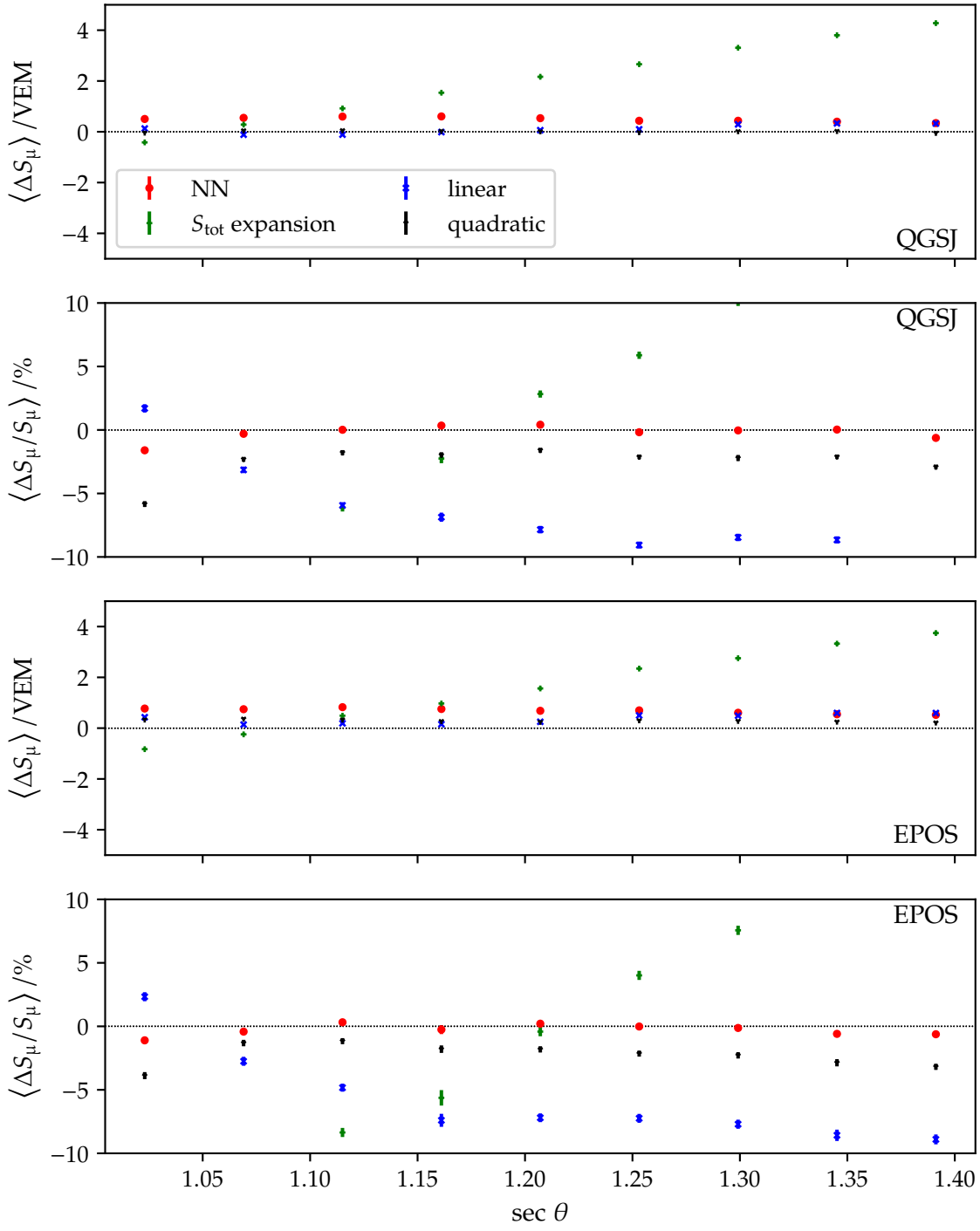


Figure D.5: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of sec θ. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively.

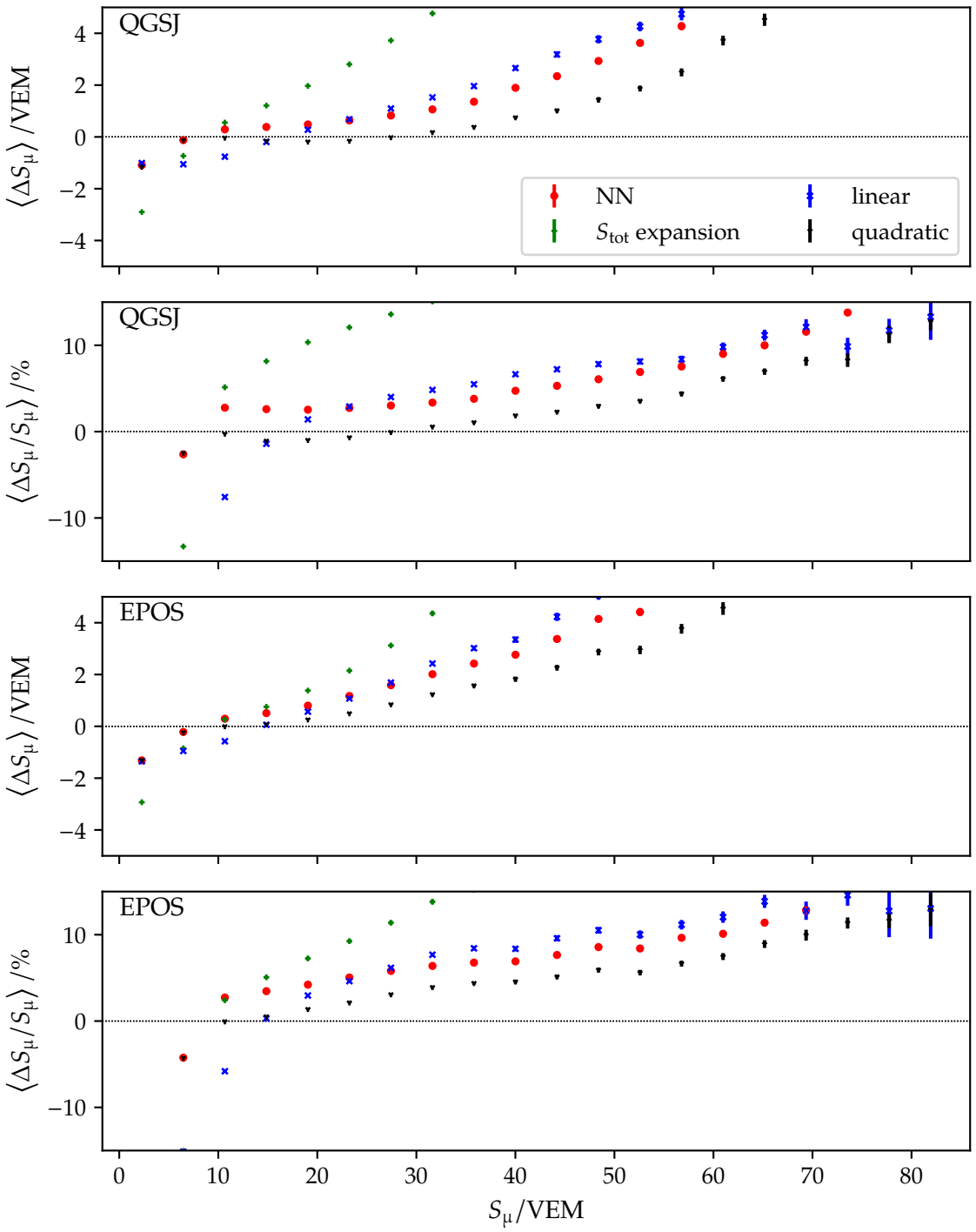


Figure D.6: Bias (*first*, *third*) and relative error (*second*, *fourth*) of NN as a function of S_μ. The *upper* two panels and the *lower* two panels are the results for the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively.

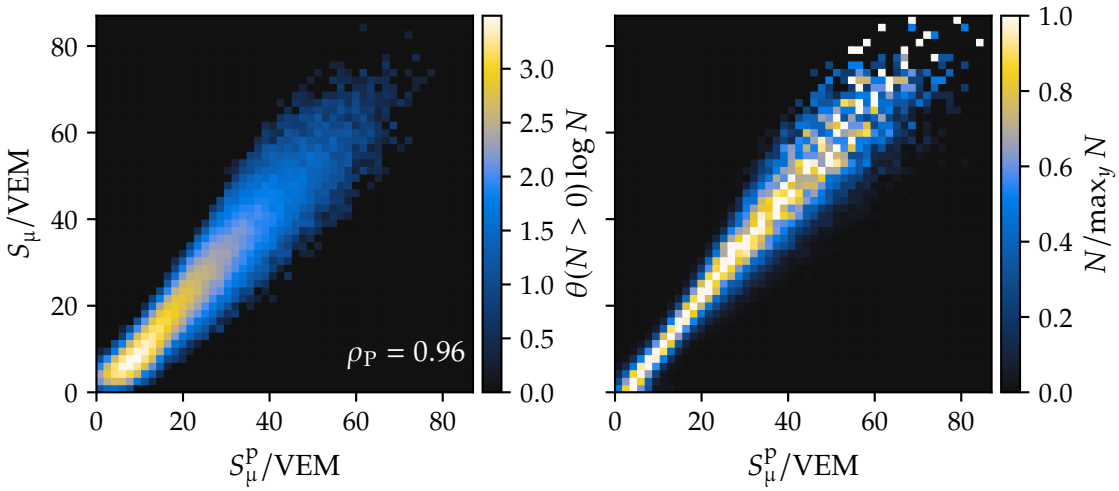


Figure D.7: Histograms of S_μ value vs the prediction of the linear model in logarithmic representation (*left*) and normalized to the maximum value of the row (*right*).

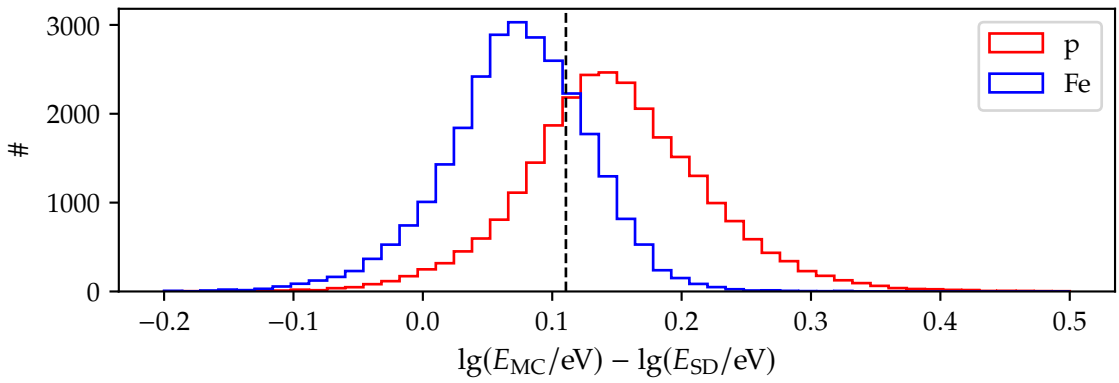


Figure D.8: Distribution of the difference between the logarithmic MC energy $\lg E_{\mathrm{MC}}$ and the logarithmic SD energy predictions $\lg E_{\mathrm{SD}}$ for the proton (red) and iron (blue) primaries. The vertical line (black, dashed) marks the global shift of the SD predictions on the MC data set (see Eq. (7.4)).

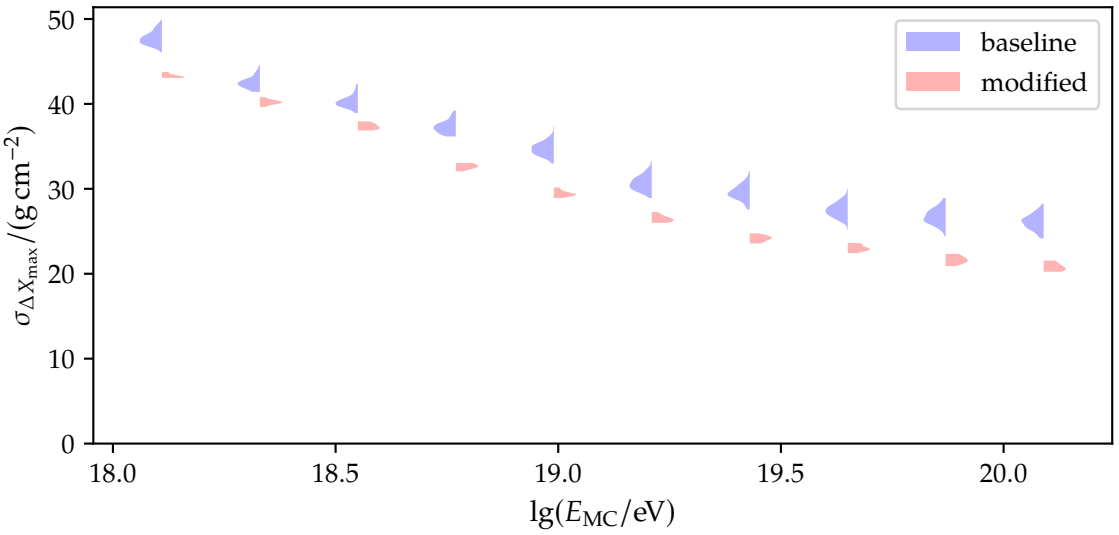


Figure D.9: Distribution of precision of ΔX_{max} for the predictions of the baseline network and the NN trained on the optimized setup in bins of logarithmic MC energy.

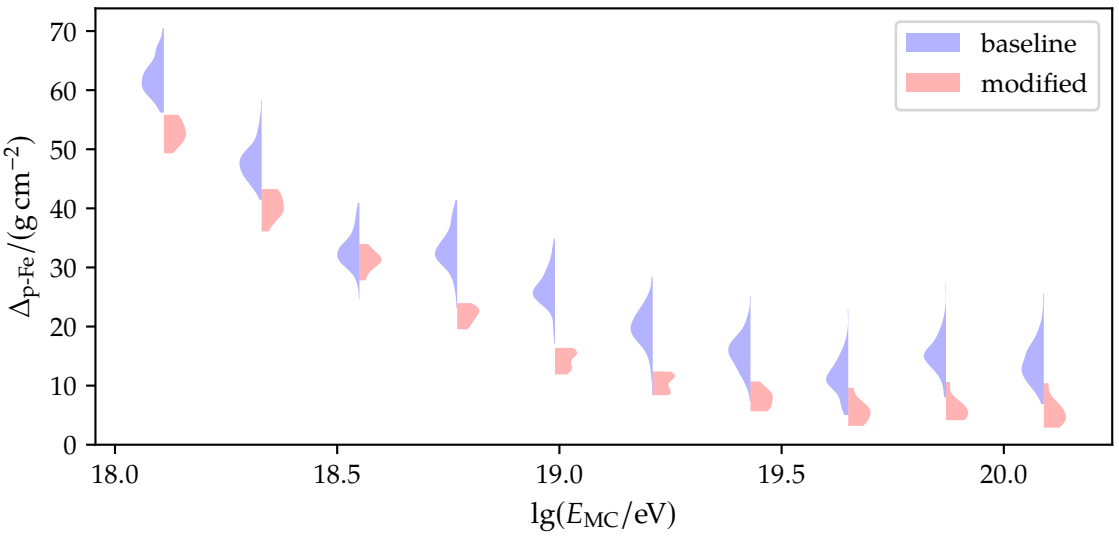


Figure D.10: Distribution of proton-iron bias of ΔX_{max} for the predictions of the baseline network and the NN trained on the optimized setup in bins of logarithmic MC energy.

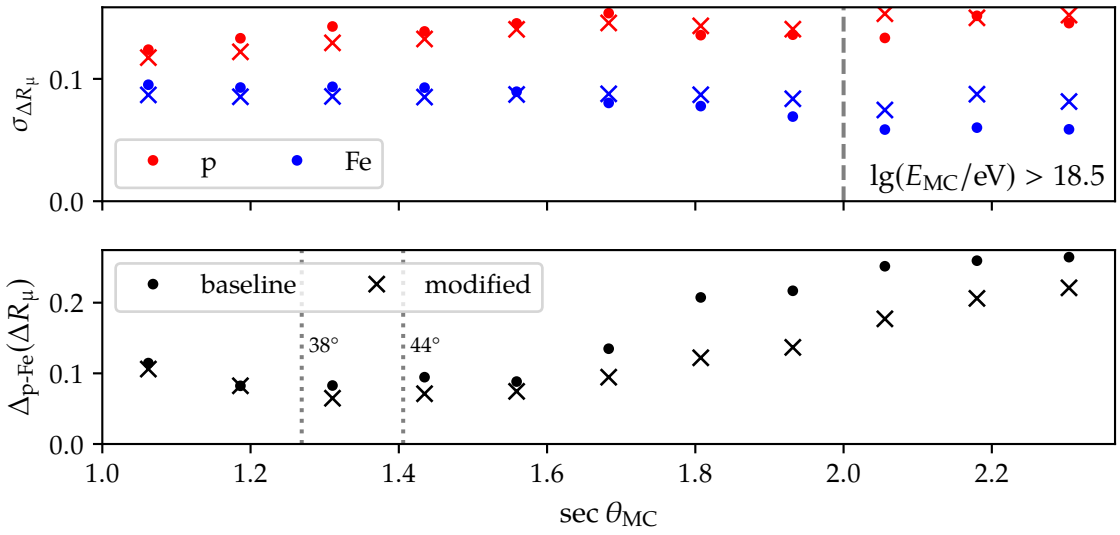


Figure D.11: Comparison of the precision $\sigma_{\Delta R_\mu}$ (*top*) and of the proton-iron bias $\Delta_{p\text{-}Fe}(\Delta R_\mu)$ (*bottom*) between the predictions of the baseline network (dots) and the predictions of the network (crosses) using the modified architecture (see Sec. 7.2.4) as a function of sec θ_{MC}.

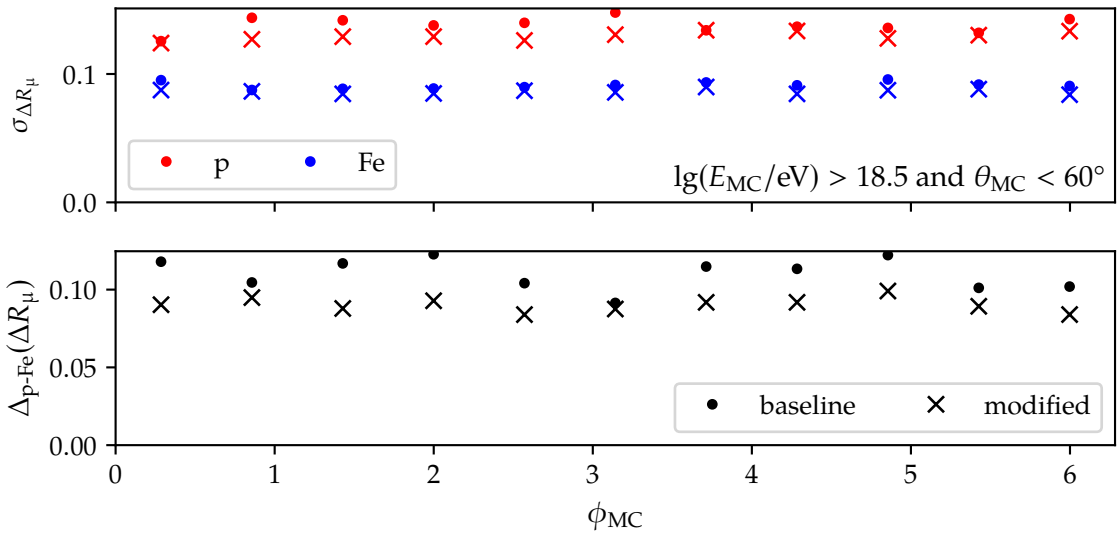


Figure D.12: Comparison of the precision $\sigma_{\Delta R_\mu}$ (*top*) and of the proton-iron bias $\Delta_{p\text{-}Fe}(\Delta R_\mu)$ (*bottom*) between the predictions of the baseline network (dots) and the predictions of the network (crosses) using the modified architecture (see Sec. 7.2.4) as a function of ϕ_{MC}.

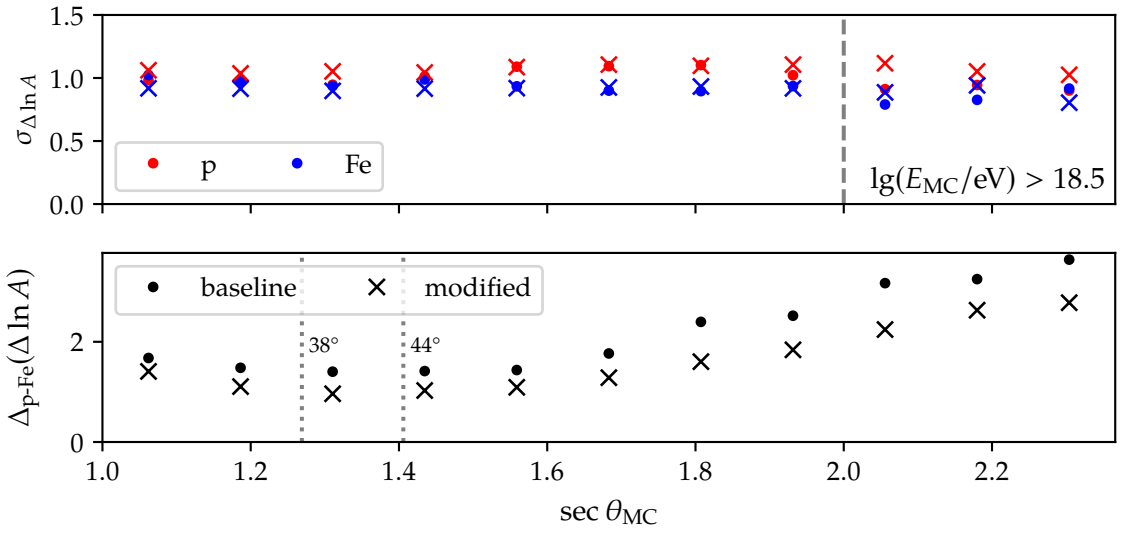


Figure D.13: Comparison of the precision $\sigma_{\Delta \ln A}$ (*top*) and of the proton-iron bias $\Delta_{\text{p-Fe}}(\Delta \ln A)$ (*bottom*) between the predictions of the baseline network (dots) and the predictions of the network (crosses) using the modified architecture (see Sec. 7.2.4) as a function of $\sec \theta_{\mathrm{MC}}$.

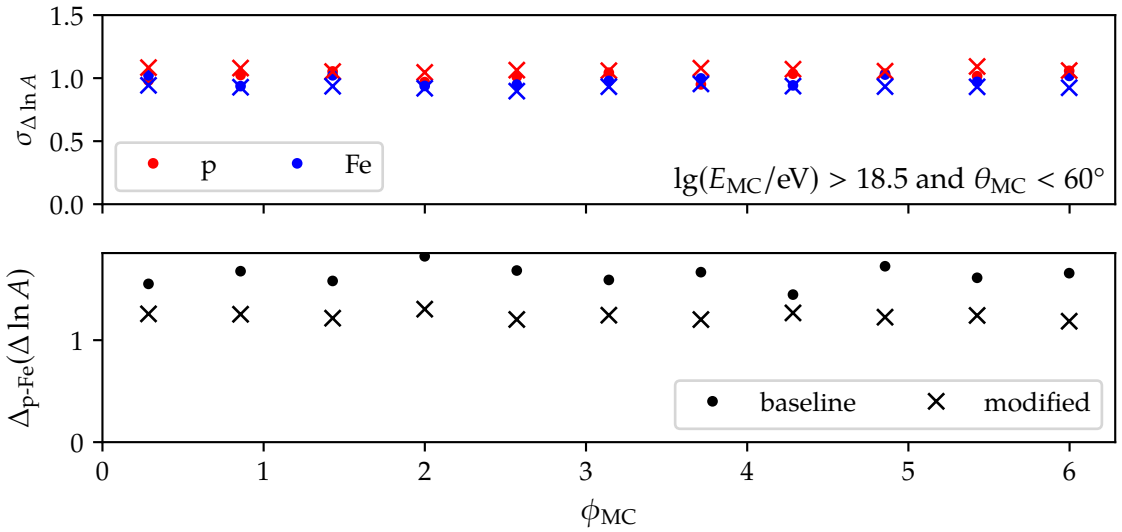


Figure D.14: Comparison of the precision $\sigma_{\Delta \ln A}$ (*top*) and of the proton-iron bias $\Delta_{\text{p-Fe}}(\Delta \ln A)$ (*bottom*) between the predictions of the baseline network (dots) and the predictions of the network (crosses) using the modified architecture (see Sec. 7.2.4) as a function of ϕ_{MC}.

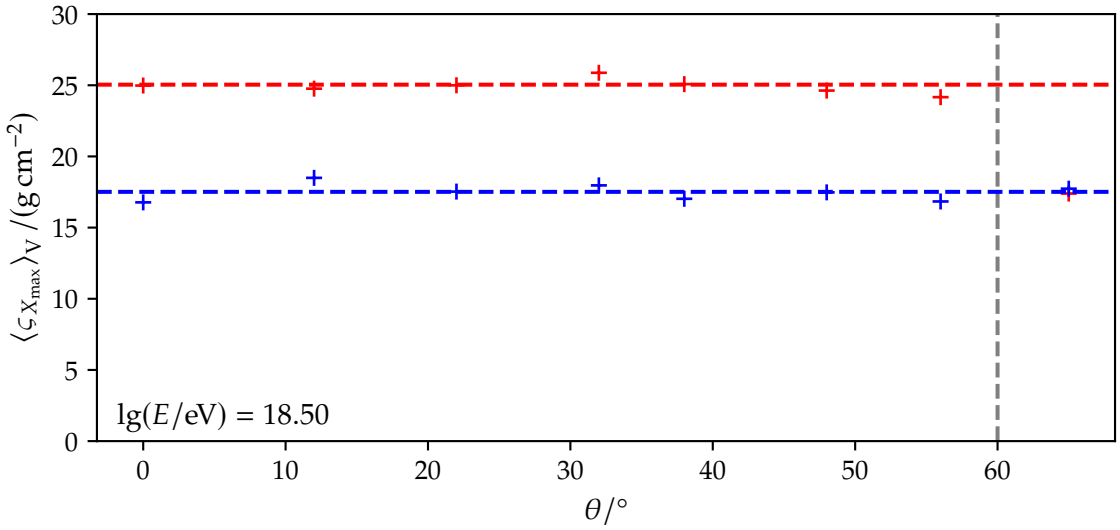


Figure D.15: Variance-weighted average values of $\varsigma(\Delta X_{max})$ for proton (points, red) and iron (points, blue) events for each zenith bin in the lg(E/eV) = 18.5 bin of the *Karlsruhe* library. The horizontal dashed lines show the average value of ς for the proton (red) and iron (blue) events without accounting for the last zenith bin. The vertical dashed line marks the 100% efficiency boundary of the SD detector.

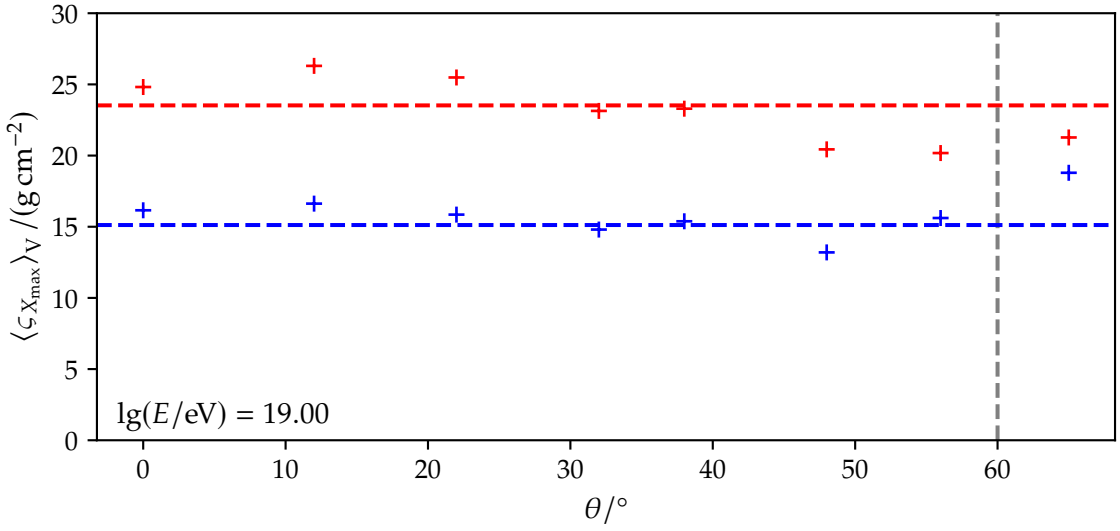


Figure D.16: Variance-weighted average values of $\varsigma(\Delta X_{max})$ for proton (points, red) and iron (points, blue) events for each zenith bin in the lg(E/eV) = 19.0 bin of the *Karlsruhe* library. The horizontal dashed lines show the average value of ς for the proton (red) and iron (blue) events without accounting for the last zenith bin. The vertical dashed line marks the 100% efficiency boundary of the SD detector.

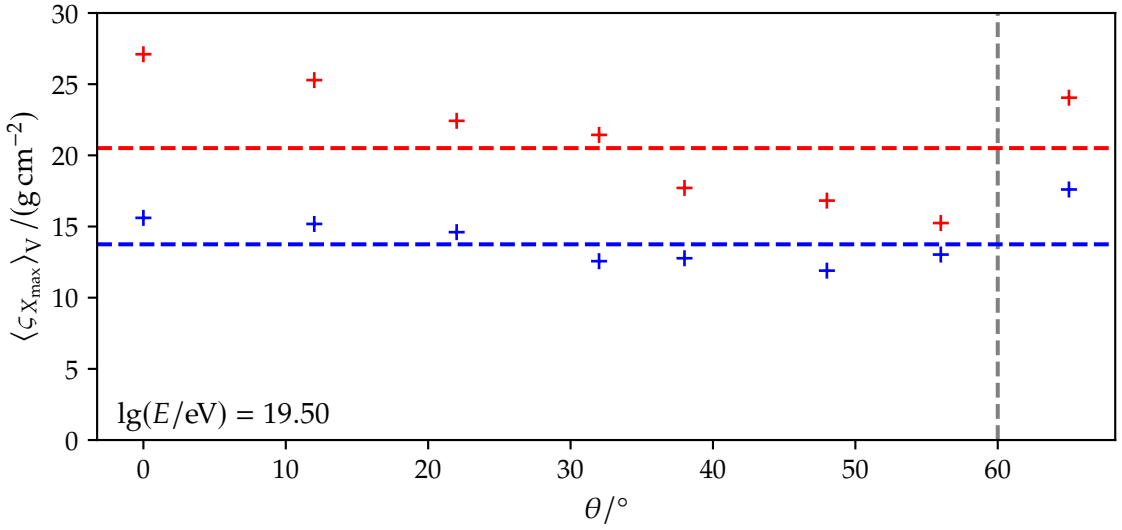


Figure D.17: Variance-weighted average values of $\varsigma(\Delta X_{\max})$ for proton (points, red) and iron (points, blue) events for each zenith bin in the $\lg(E/\mathrm{eV}) = 19.5$ bin of the *Karlsruhe* library. The horizontal dashed lines show the average value of ς for the proton (red) and iron (blue) events without accounting for the last zenith bin. The vertical dashed line marks the 100% efficiency boundary of the SD detector.

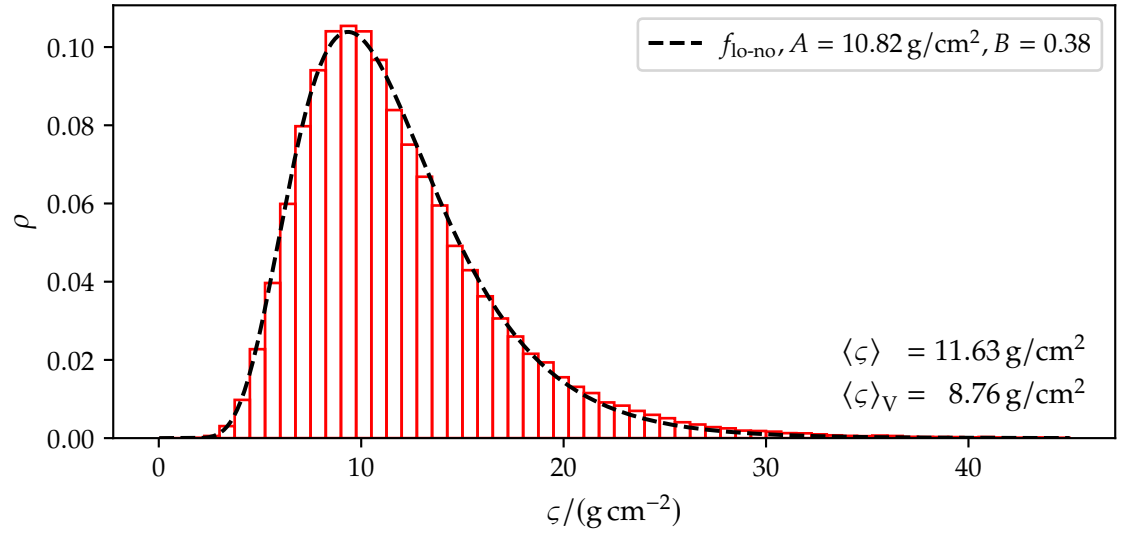


Figure D.18: Distribution of ς (see Eq. (4.45)) for models trained on the conditions defined in Sec. 7.3.2.A without the dropout layer.

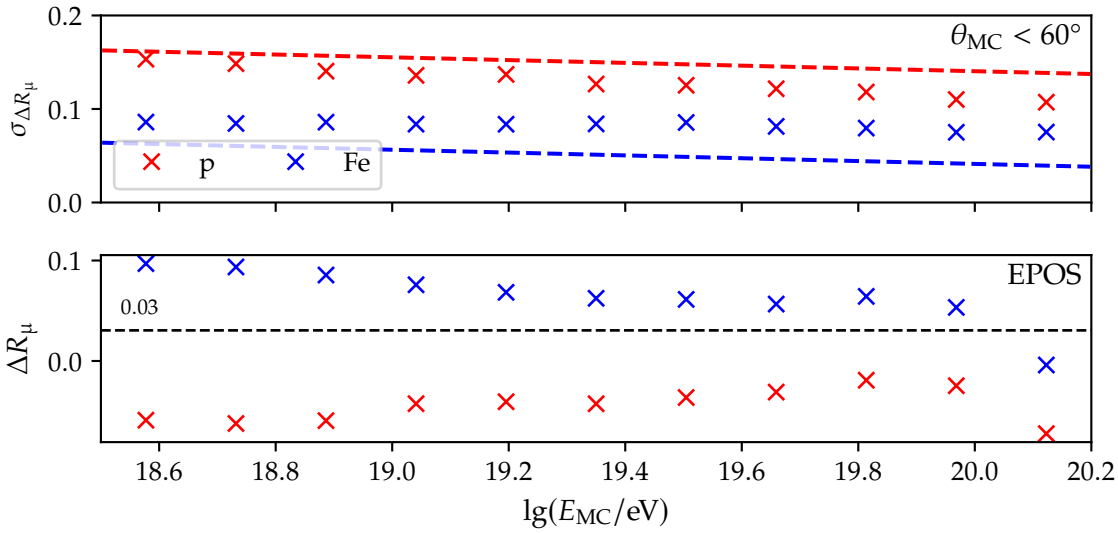


Figure D.19: Precision $\sigma_{\Delta R_\mu}$ (*top*) and bias ΔR_μ (*bottom*) for the NN predictions of the proton- and iron-part of the data set based on simulations using the hadronic interaction model EPOS-LHC. The NN has been trained on a data set using the hadronic interaction model QGSJetII-04. The dashed lines in the *top* panel represent the expected width of the R_μ distribution for pure proton (red) and pure iron (blue) events. The predictions of the NN have a global bias of about 0.03 (see Row D.1.g). The horizontal line in the *bottom* plot (black, dashed) marks this global bias.

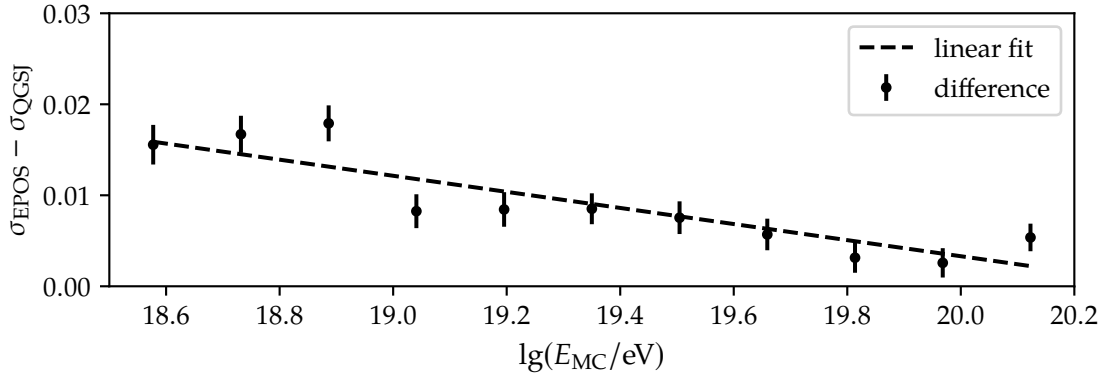


Figure D.20: Difference of the precision of the NN-based prediction of R_μ on the data sets defined in Row 5.5.d and Row 5.5.f in bins of energy. The line (black, dashed) is a linear fit to the difference (see Row D.1.h).

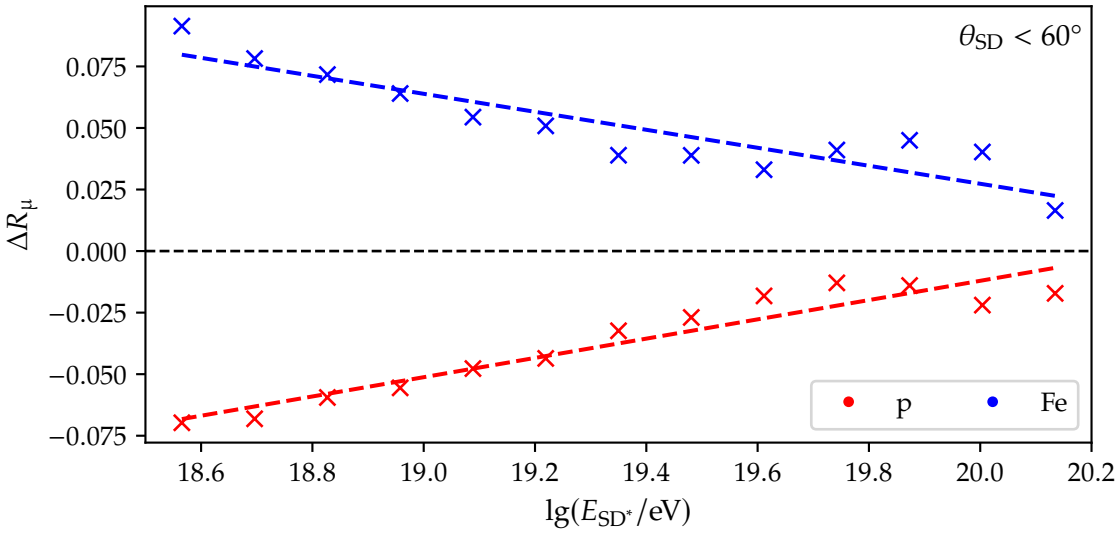


Figure D.21: The bias ΔR_μ for events belonging to proton (red) and iron (blue) primaries. The dashed lines are linear fits to both of the subsets (see Rows D.1.i to D.1.j).

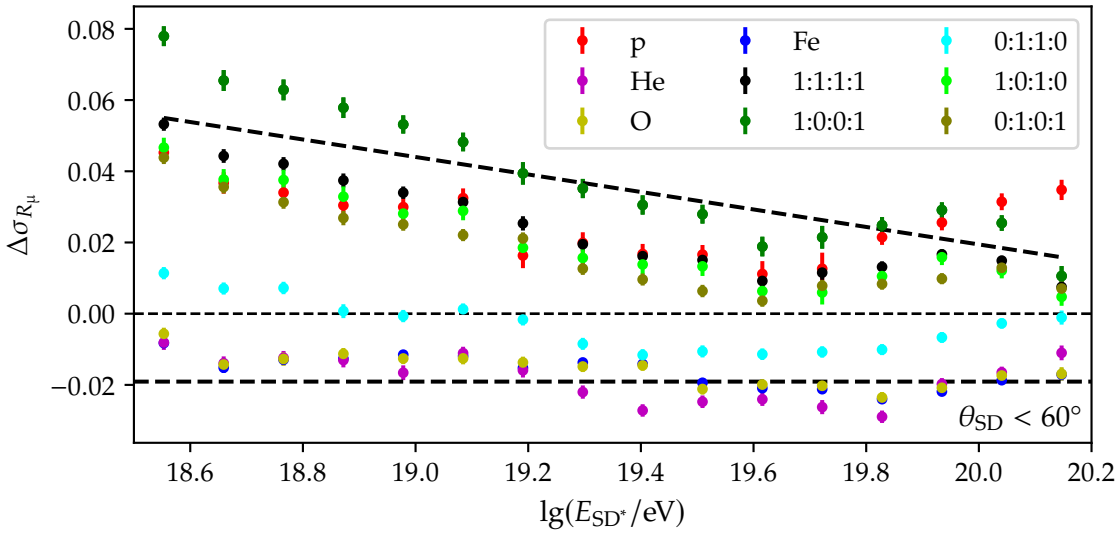


Figure D.22: The difference between the width of the MC distribution and the width of the predictions for different compositions. We have fitted an linear function to the three highest values in each bin and a constant to the minimum values (see Rows D.1.k to D.1.l).

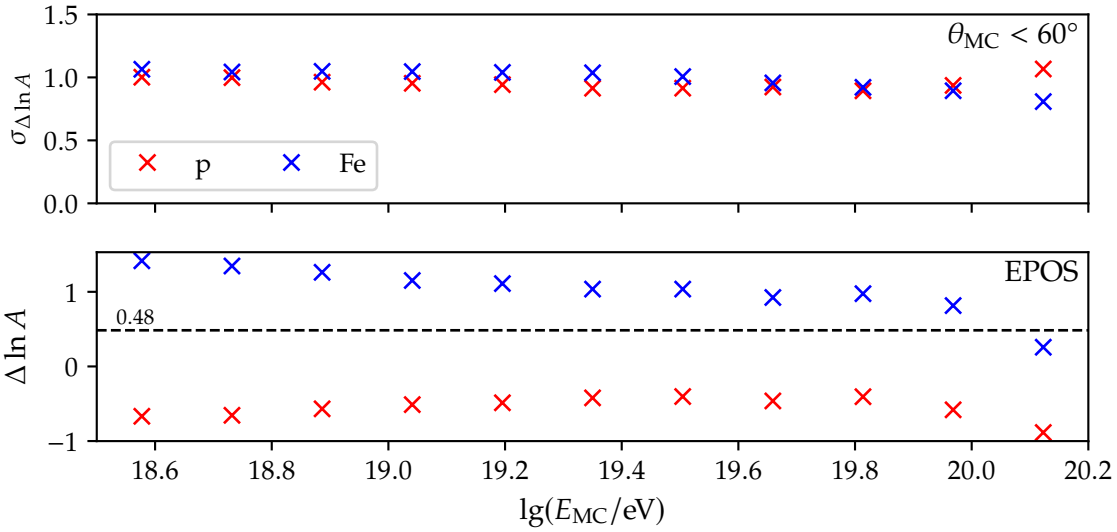


Figure D.23: Precision $\sigma_{\Delta \ln A}$ (*top*) and bias $\Delta \ln A$ (*bottom*) for the NN predictions of the proton- and iron-part of the data set based on simulations using the hadronic interaction model EPOS-LHC. The NN has been trained on a data set using the hadronic interaction model QGSJetII-04. The predictions of the NN have a global bias of about 0.48. The horizontal line in the *bottom* plot (black, dashed) marks this global bias (see Row D.1.m).

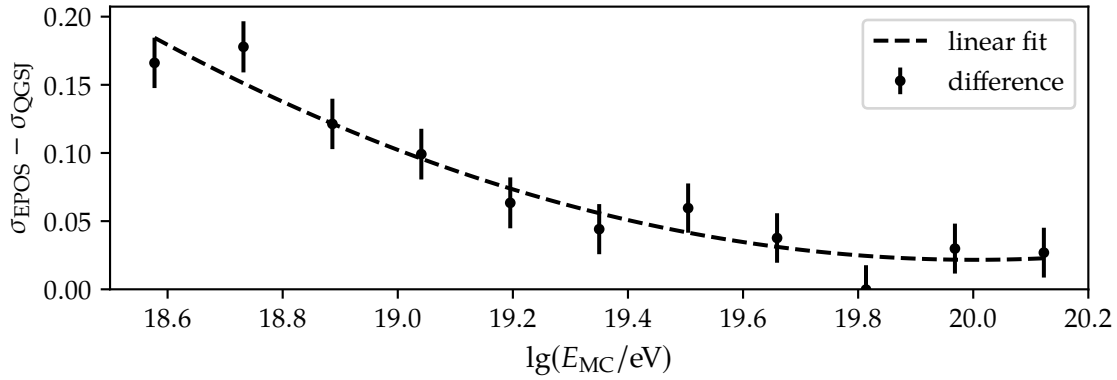


Figure D.24: Difference of the precision of the NN-based prediction of $\ln A$ on the data sets defined in Row 5.5.d and Row 5.5.f in bins of energy. The line (black, dashed) is a quadratic fit to the difference (see Row D.1.n).

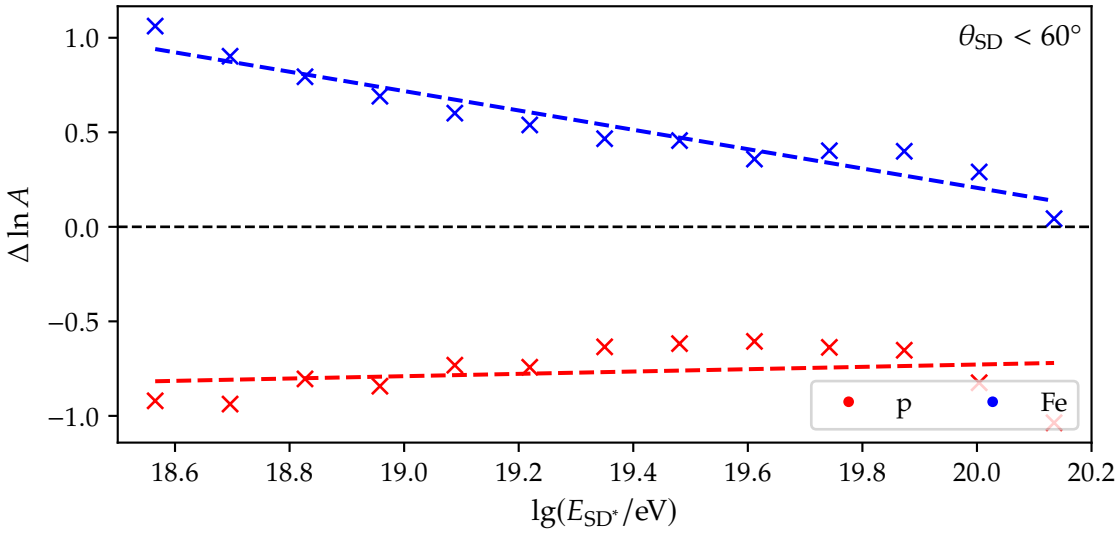


Figure D.25: The bias ΔX_{max} for events belonging to proton (red) and iron (blue) primaries. The dashed lines are linear fits to both of the subsets (see Rows D.1.o to D.1.p).

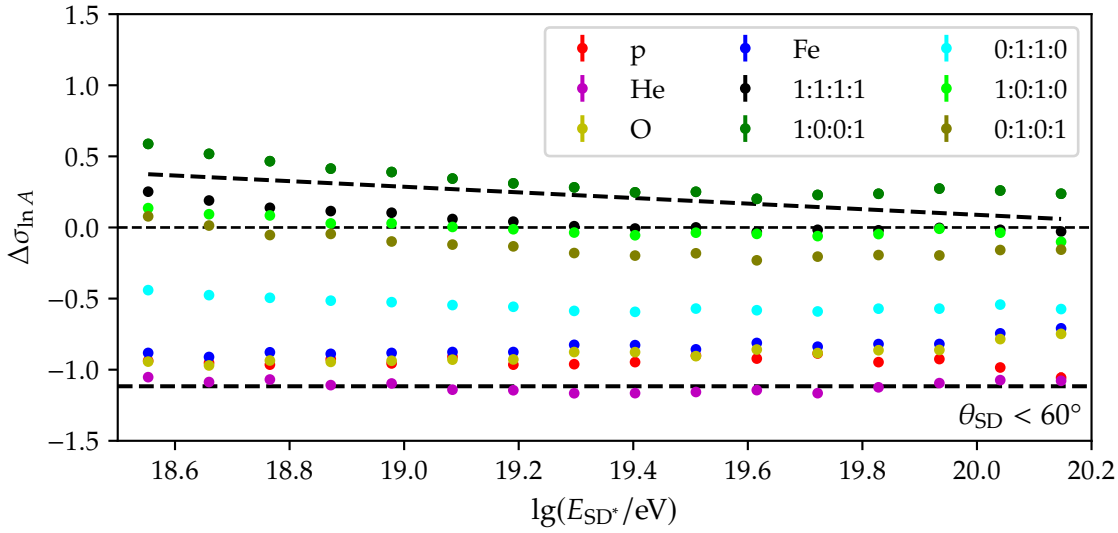


Figure D.26: The difference between the width of the MC distribution and the width of the predictions for different compositions. We have fitted an linear function to the three maximum values in each bin and a constant to the minimum values (see Rows D.1.q to D.1.r).

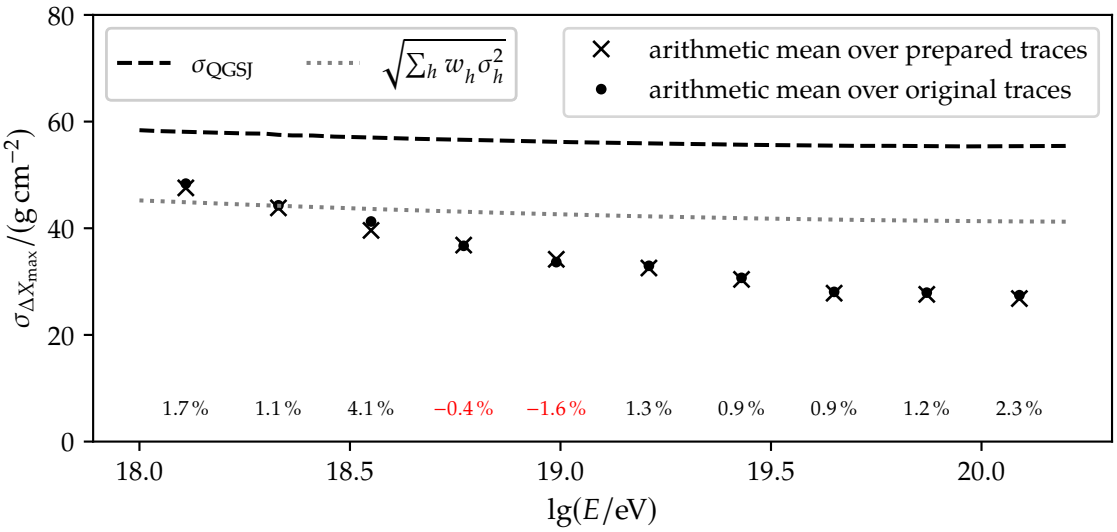


Figure D.27: Comparison of the precision $\sigma_{\Delta X_{\max}}$ of predictions of two NNs using differently downsampled UUB traces as a function of the logarithmic energy.

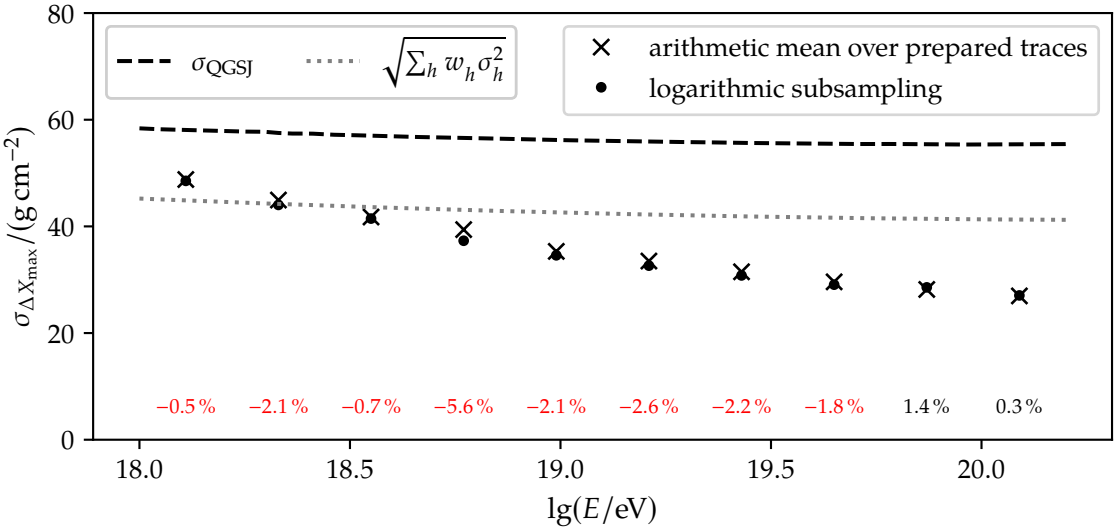


Figure D.28: Comparison of the precision $\sigma_{\Delta X_{\max}}$ of the predictions of two NNs using differently downsampled UUB traces as a function of the logarithmic energy.

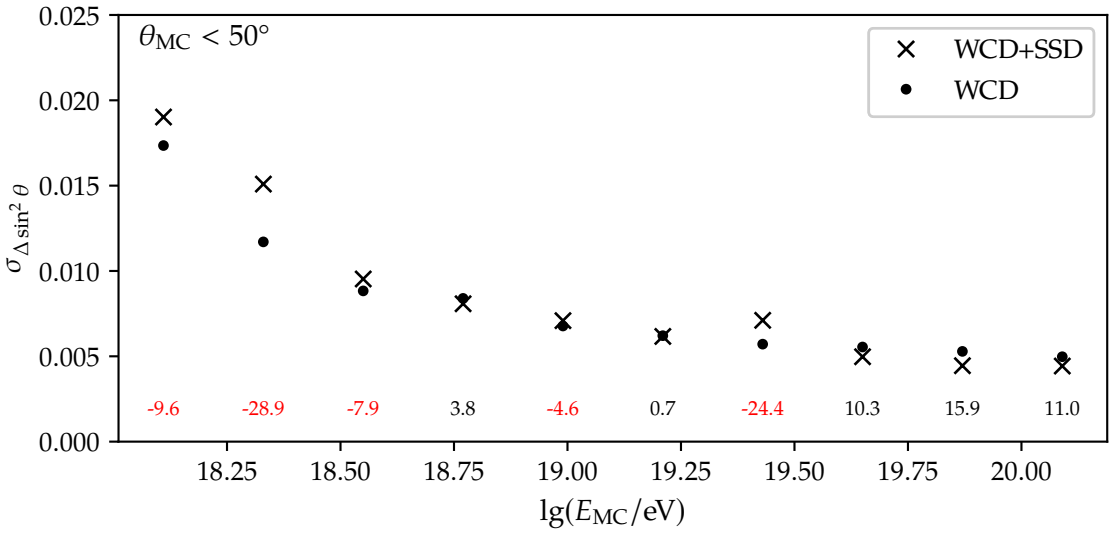


Figure D.29: Comparison of the precisions $\sigma_{\Delta \sin^2 \theta}$ of the predictions of two NNs as a function of logarithmic energy. The first NN (crosses) was trained using SSD and WCD traces, and the second NN was trained using only WCD traces.

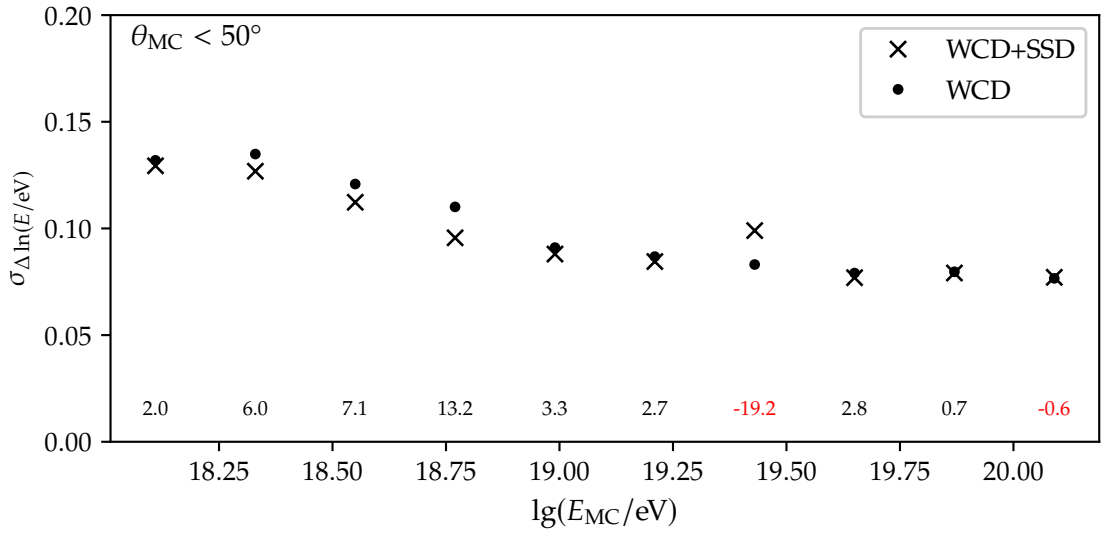


Figure D.30: Comparison of the precisions $\sigma_{\ln(E/eV)}$ of the predictions of two NNs as a function of logarithmic energy. The first NN (crosses) was trained using SSD and WCD traces, and the second NN was trained using only WCD traces.

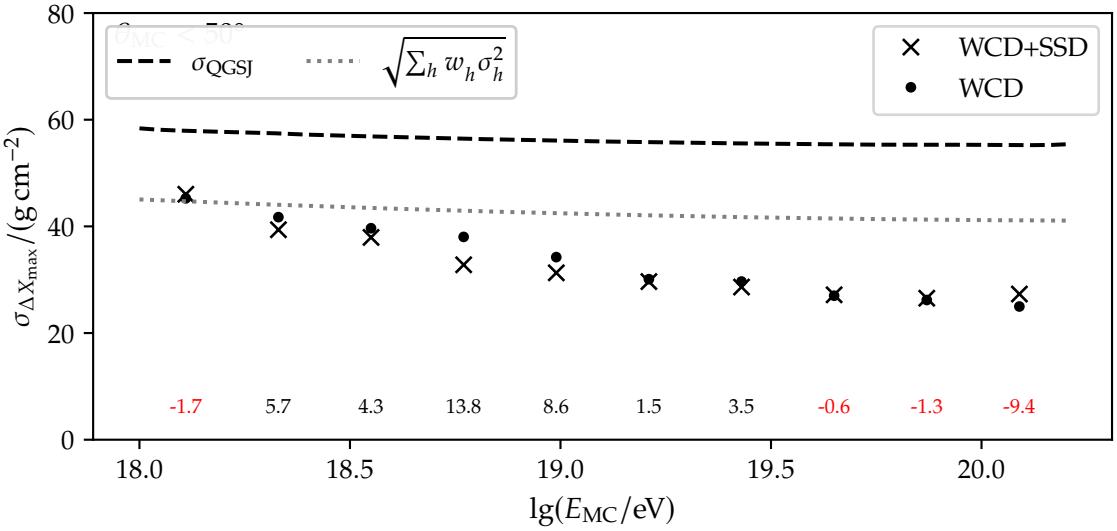


Figure D.31: Comparison of the precisions $\sigma_{X_{max}}$ of the predictions of two NNs as a function of logarithmic energy. The first NN (crosses) was trained using SSD and WCD traces, and the second NN was trained using only WCD traces.

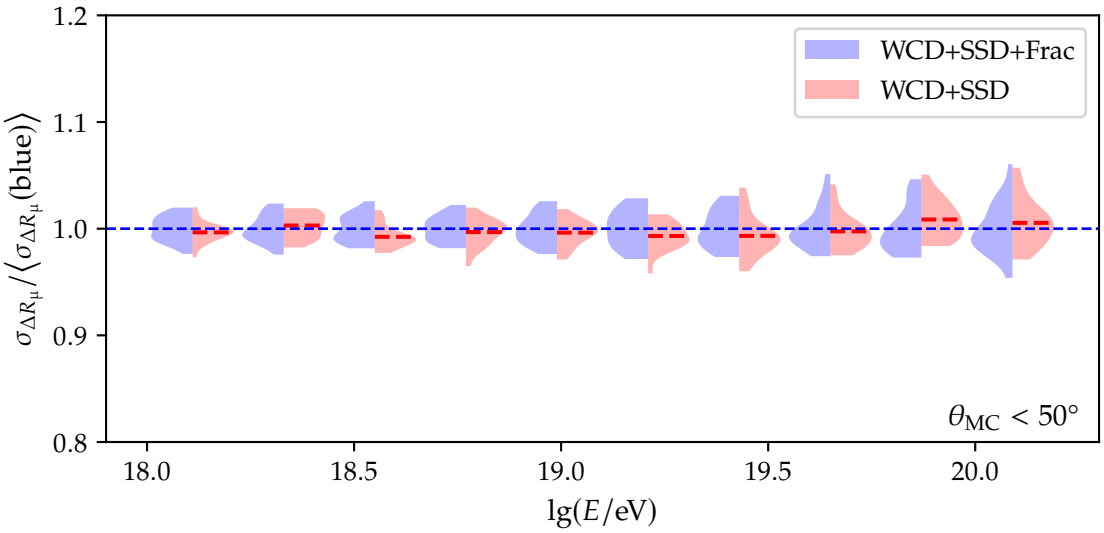


Figure D.32: Ensembles of the precision of R_μ predictions from NNs trained on WCD and SSD traces normalized to the average precision of the set of NNs corresponding to the blue distributions. The ensemble of NNs depicted in the blue distributions uses the ratio of the integrated traces as an additional input. The red dashed lines indicate the average value of the resolution of the same colored distribution of precisions in the same energy bin. Only events in the test data set exhibiting a zenith angle below 50° are considered.

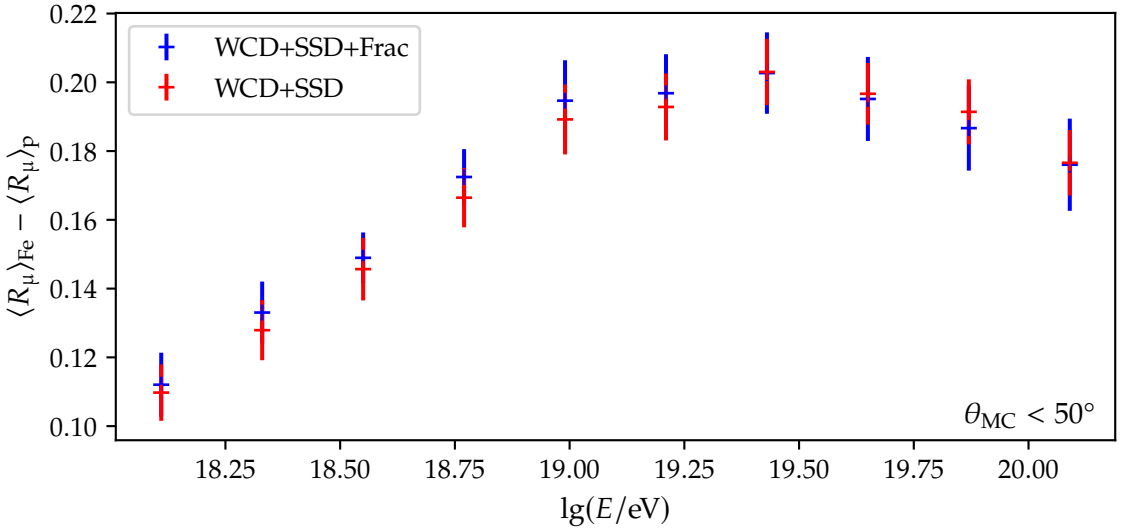


Figure D.33: Average difference between R_μ predictions of iron events and proton events from NNs trained using the ratio of the integrated traces as an addtional input (plusses, blue) and NNs without this extra information. Both types of network use the WCD and SSD traces as inputs. Only events in the test data set with a zenith angle below 50° are considered. Note that for the error bars the standard deviations of the underlying distributions have been used.

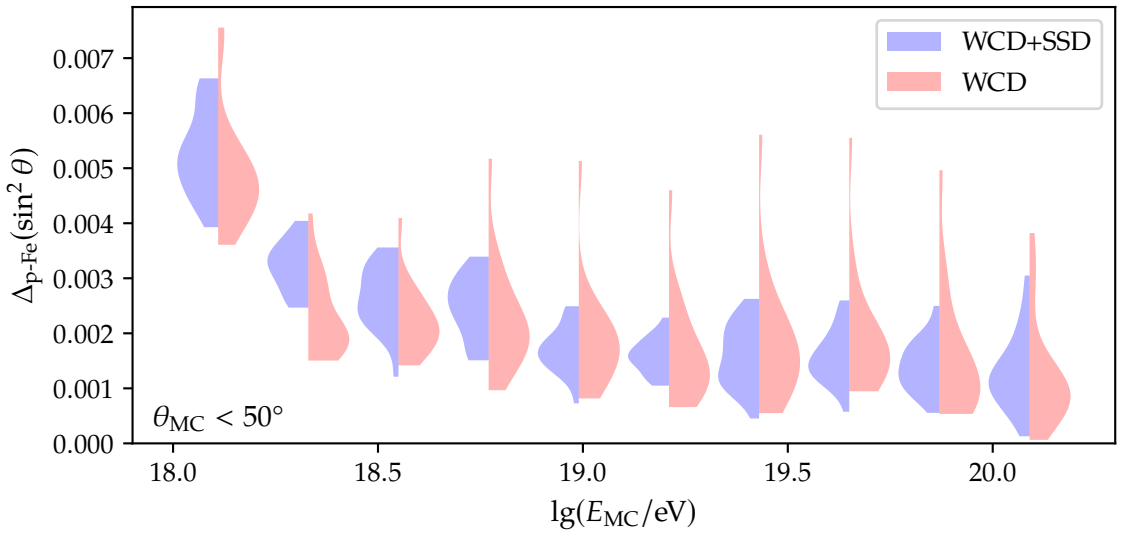


Figure D.34: Ensembles of the proton-iron bias $\Delta_{p\text{-}Fe} \sin^2\theta$ of predictions from NNs trained on WCD and SSD traces (blue) and only on WCD traces. Only events in the test data set exhibiting a zenith angle below 50° are considered.

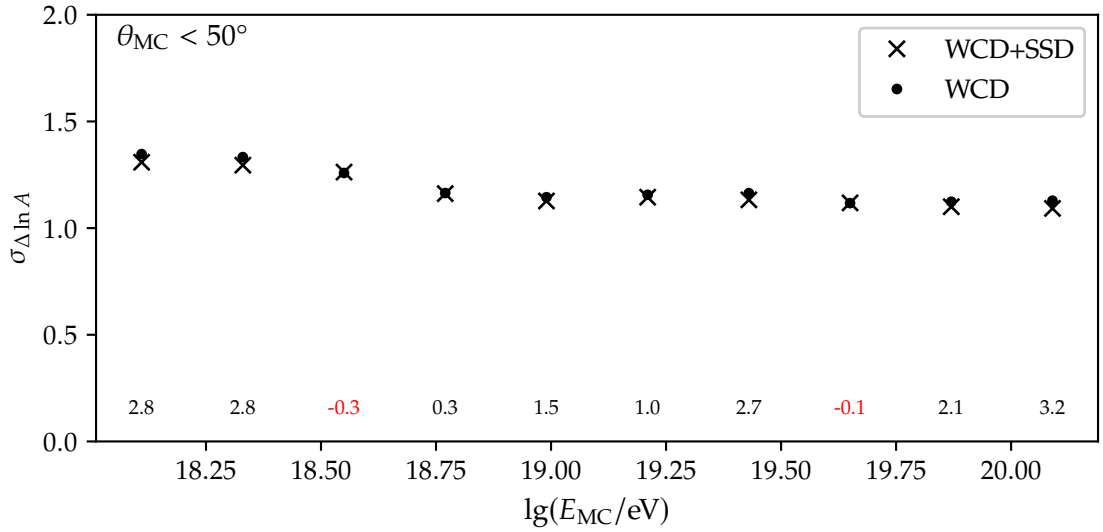


Figure D.35: Comparison of precision $\sigma_{\ln A}$ of the predictions of NNs as the function of energy. The first NN (crosses) was trained using SSD and WCD traces, and the second NN was trained using only WCD traces.

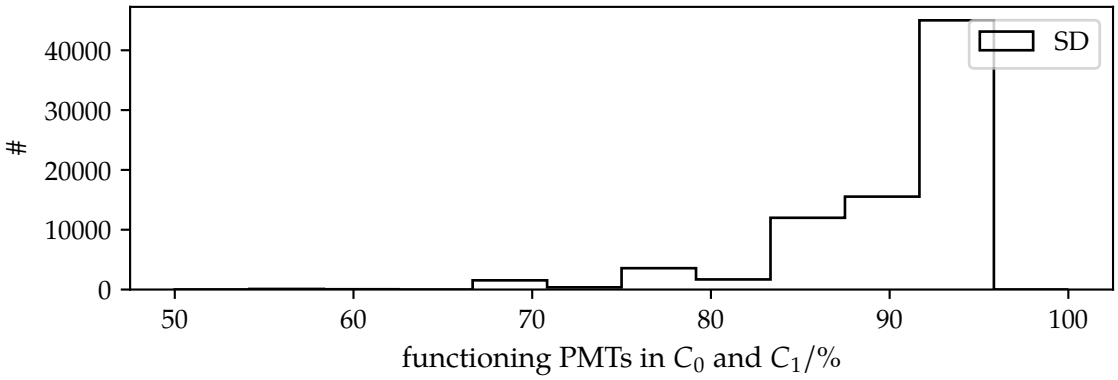


Figure D.36: Distribution of the fraction of working PMTs in the HS and the first crown for the events in the SD data set.

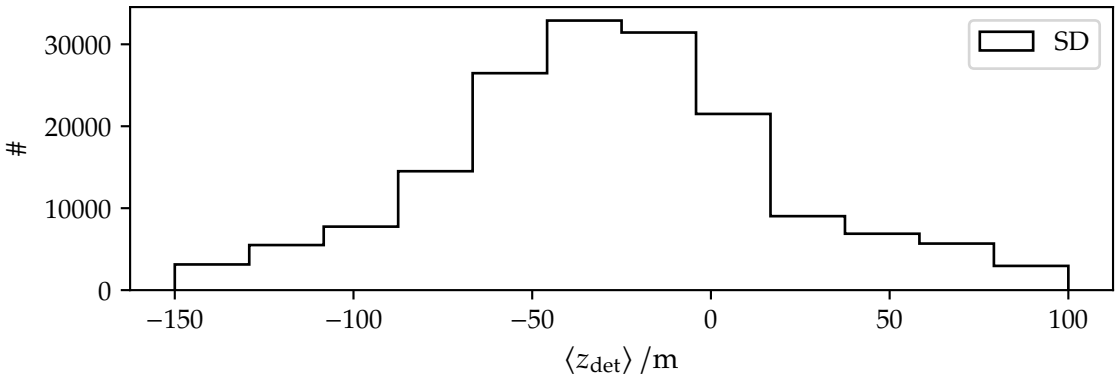


Figure D.37: Distribution of the average detector height $\langle z_{det} \rangle$ for the events in the SD data set.

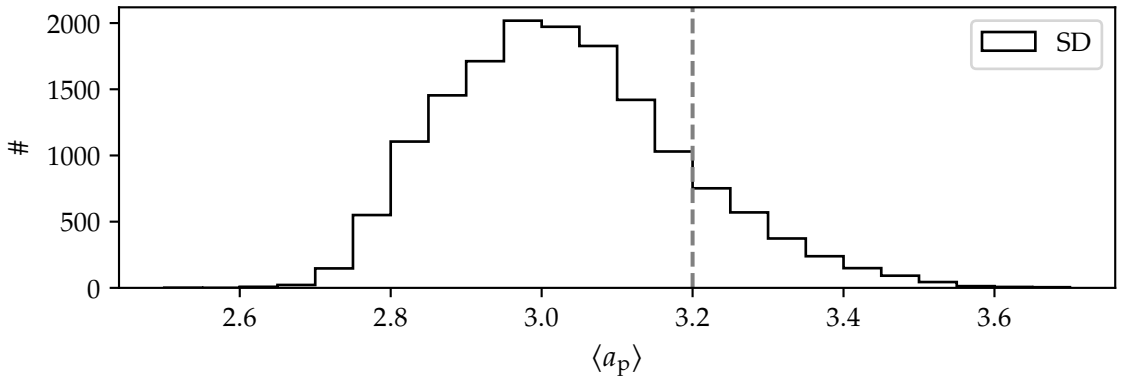


Figure D.38: Distribution of the average area over peak value for the events in the SD data set. The vertical line marks the value used in simulations.

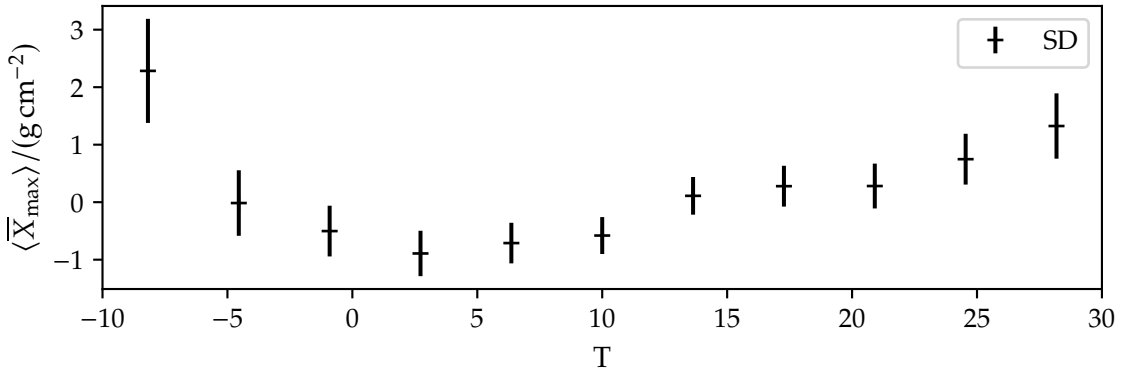


Figure D.39: Deviation from average behavior $\langle \overline{X}_{max} \rangle$ (see Eq. (8.2)) as a function of the temperature T.

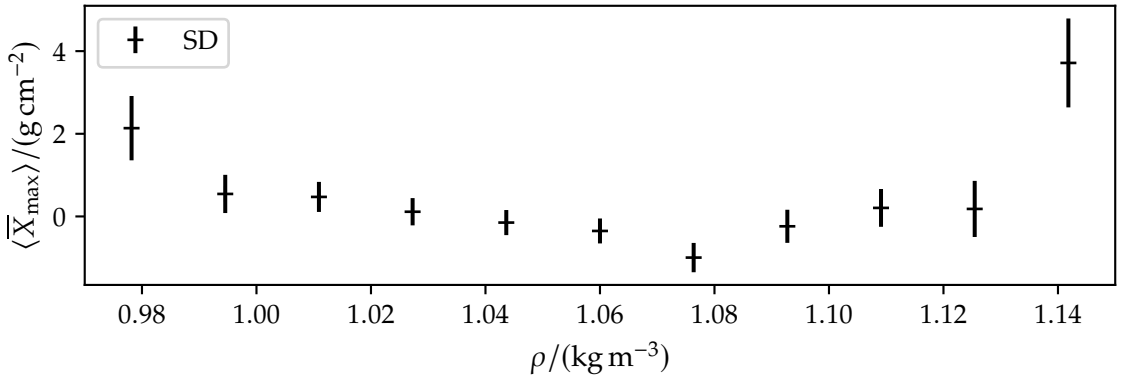


Figure D.40: Deviation from average behavior $\langle \overline{X}_{max} \rangle$ (see Eq. (8.2)) as a function of the air density ρ.

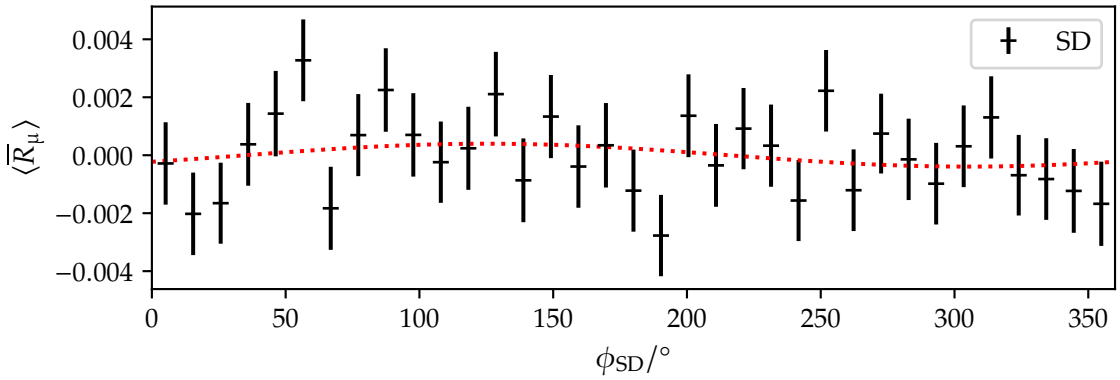


Figure D.41: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of ϕ_{SD} (cf. Fig. 8.3).

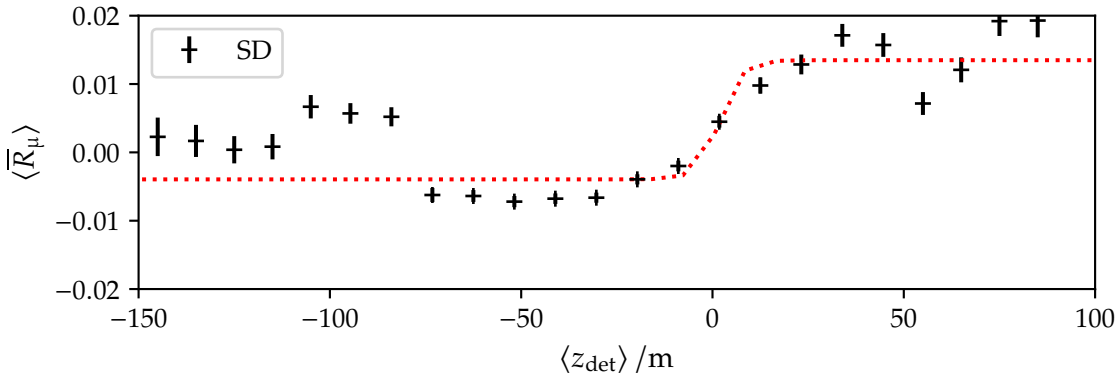


Figure D.42: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of $\langle z_{\mathrm{det}} \rangle$ (cf. Fig. 8.5).

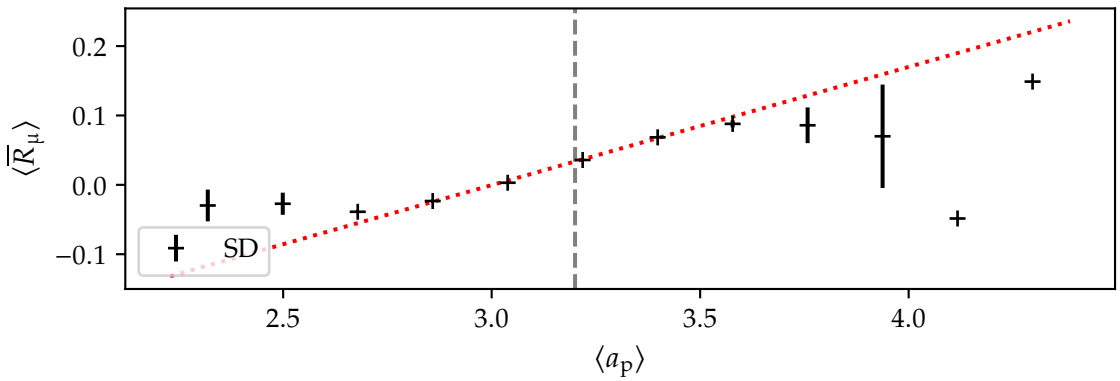


Figure D.43: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of $\langle a_{\mathrm{p}} \rangle$ (cf. Fig. 8.6).

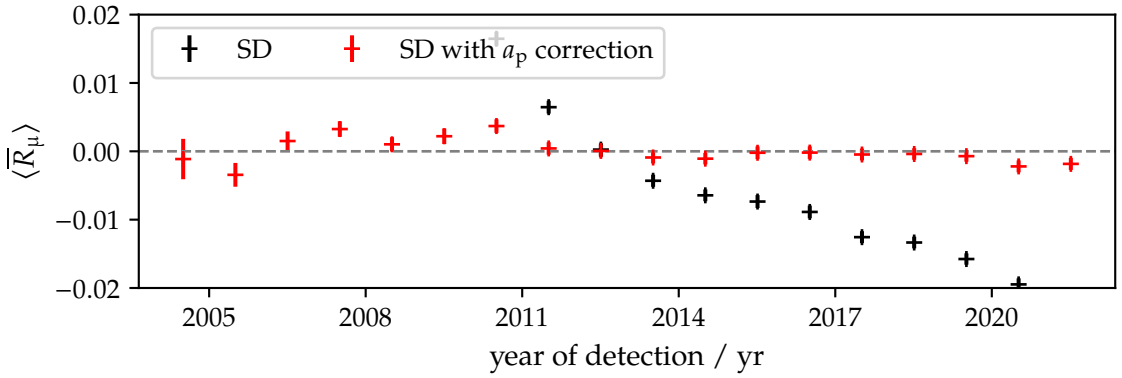


Figure D.44: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the time of detection (cf. Fig. 8.7).

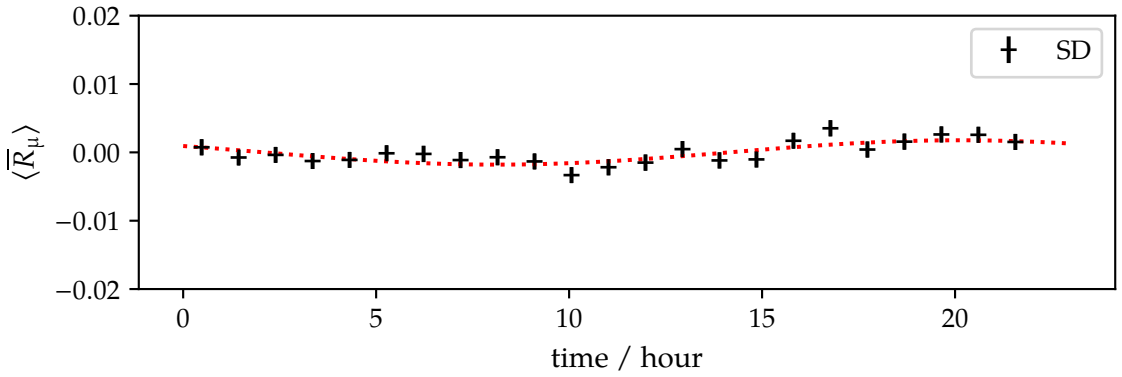


Figure D.45: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the time of day (cf. Fig. 8.8).

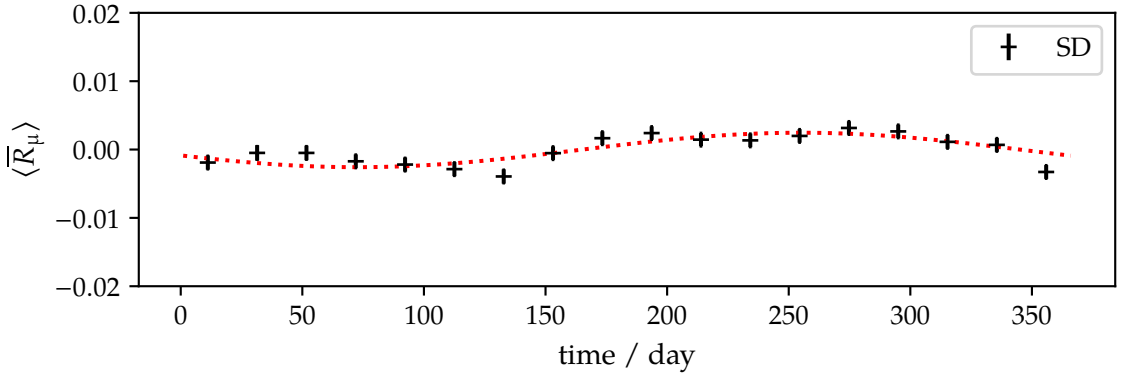


Figure D.46: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the day of the year (cf. Fig. 8.9).

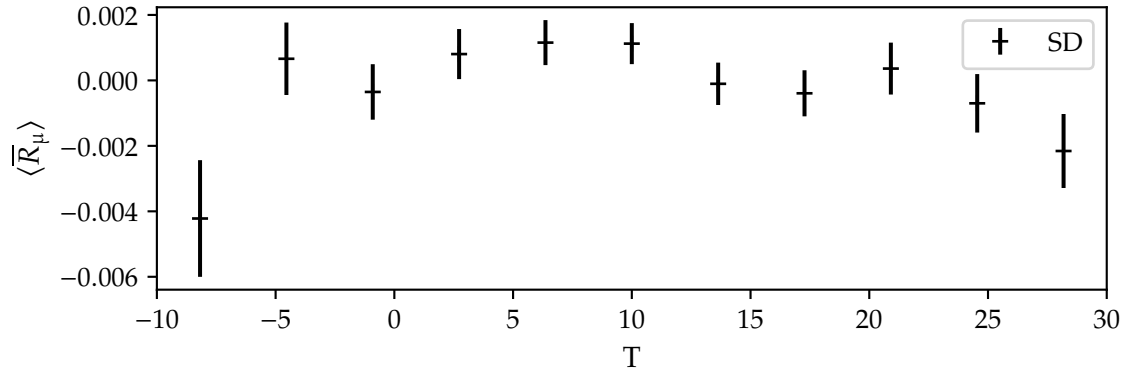


Figure D.47: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the temperature T (cf. Fig. D.39).

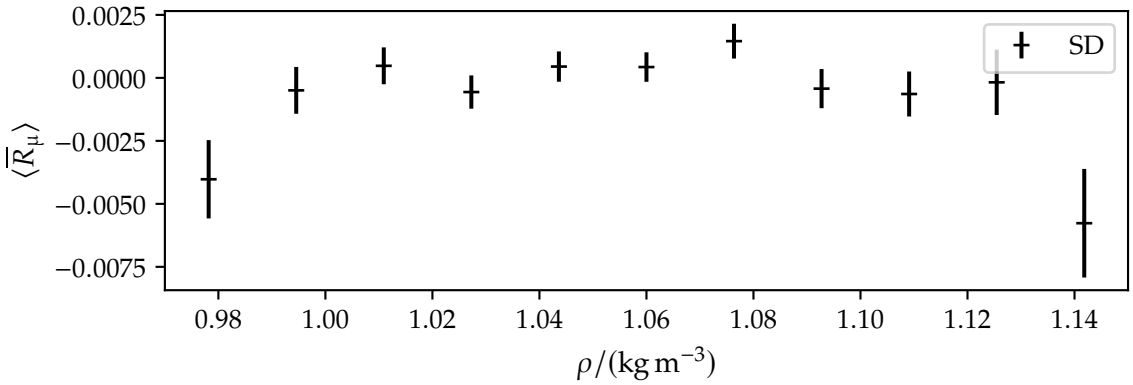


Figure D.48: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the air density ρ (cf. Fig. D.40).

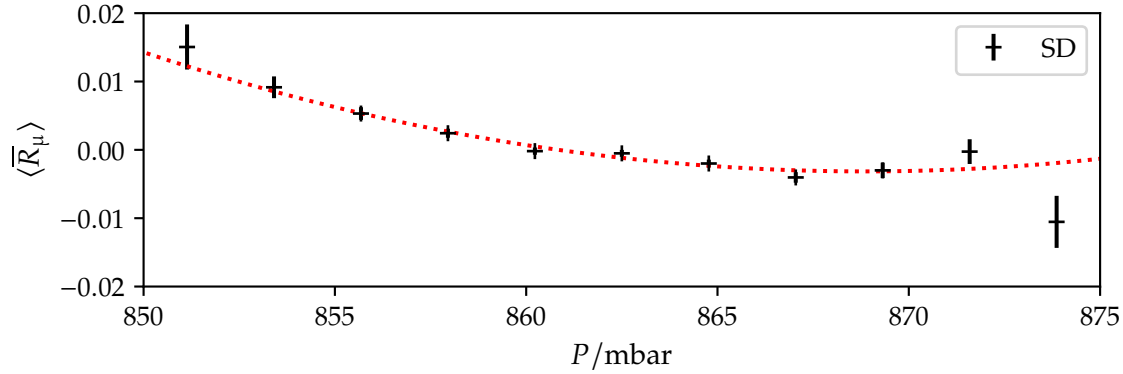


Figure D.49: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the atmospheric pressure P (cf. Fig. 8.10).

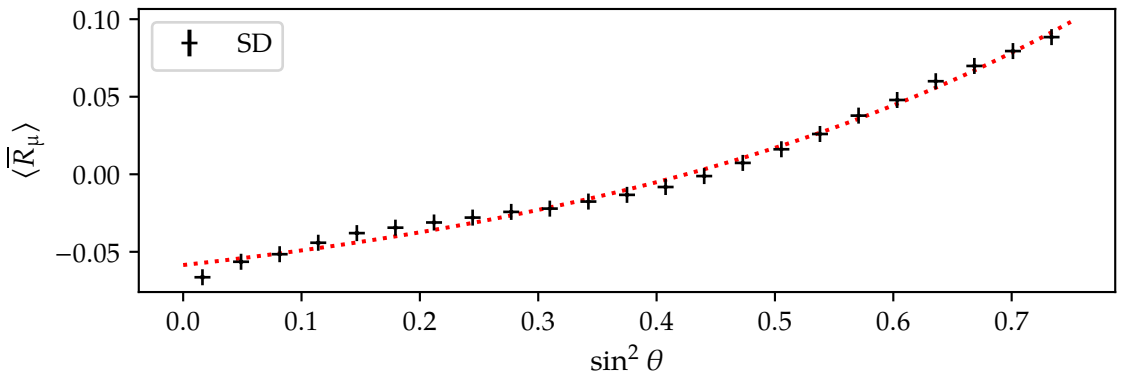


Figure D.50: Deviation from average behavior $\langle \overline{R}_\mu \rangle$ as a function of the reconstructed zenith angle in terms of $\sin^2 \theta_{SD}$ (cf. Fig. 8.11).

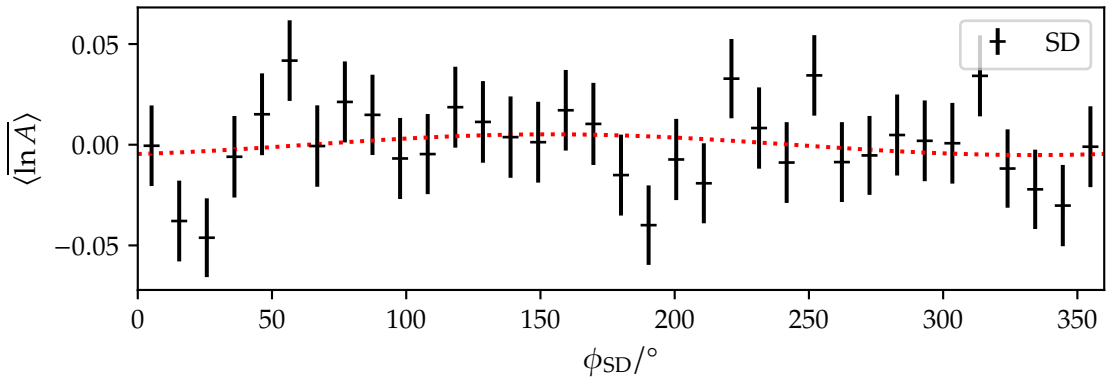


Figure D.51: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of ϕ_{SD} (cf. Fig. 8.3).

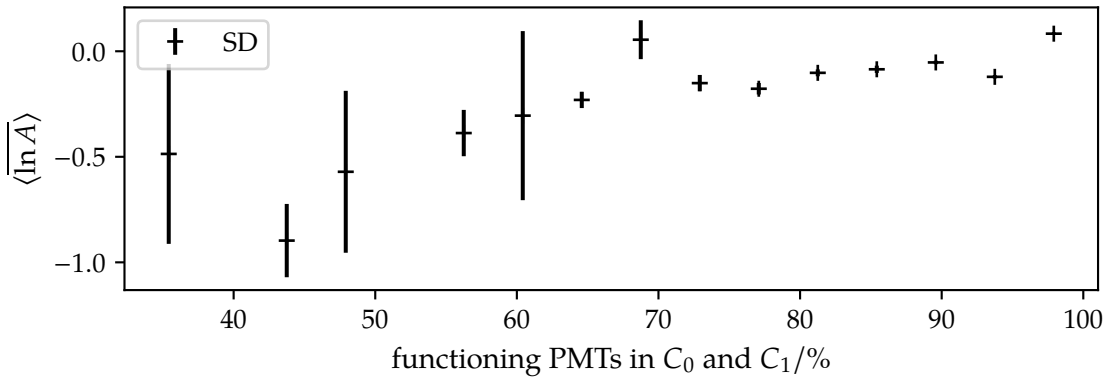


Figure D.52: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the fraction of operating PMTs in the HS and the first crown (cf. Fig. 8.4).

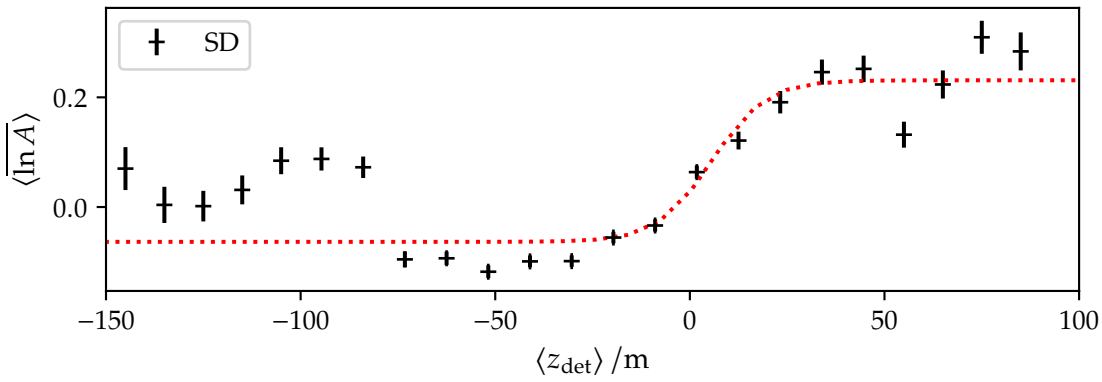


Figure D.53: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of $\langle z_{\mathrm{det}} \rangle$ (cf. Fig. 8.5).

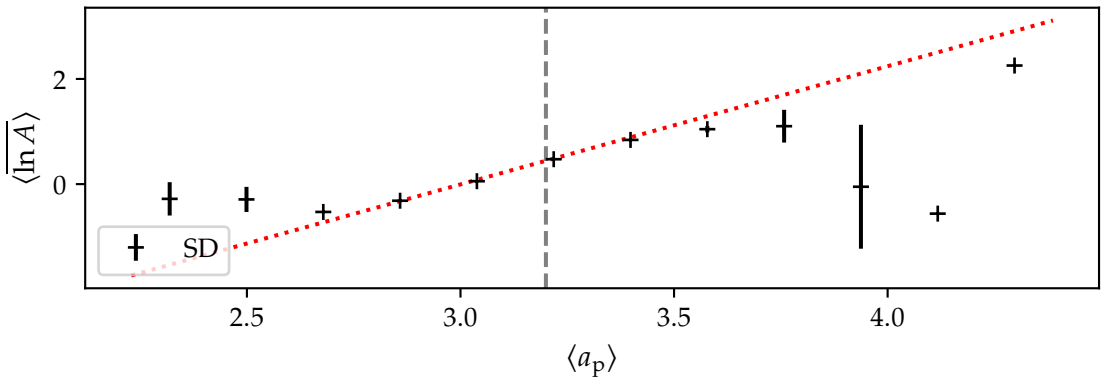


Figure D.54: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of $\langle a_{\mathrm{p}} \rangle$ (cf. Fig. 8.6).

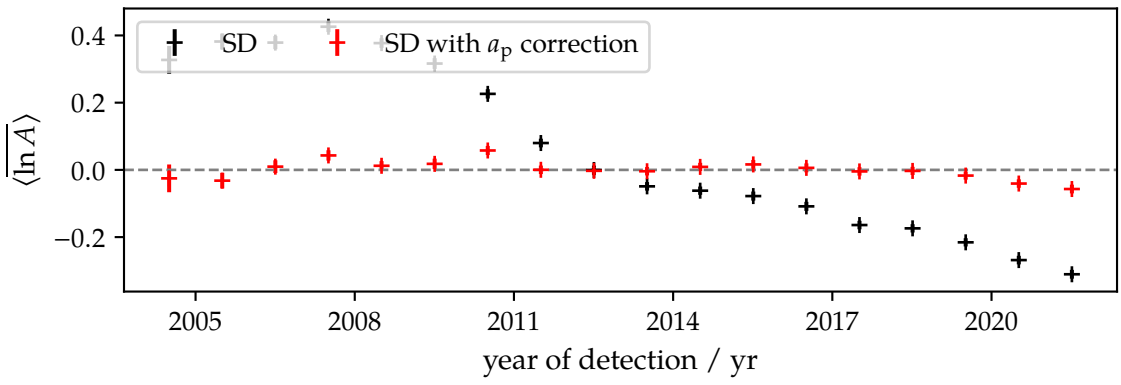


Figure D.55: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the time of detection (cf. Fig. 8.7).

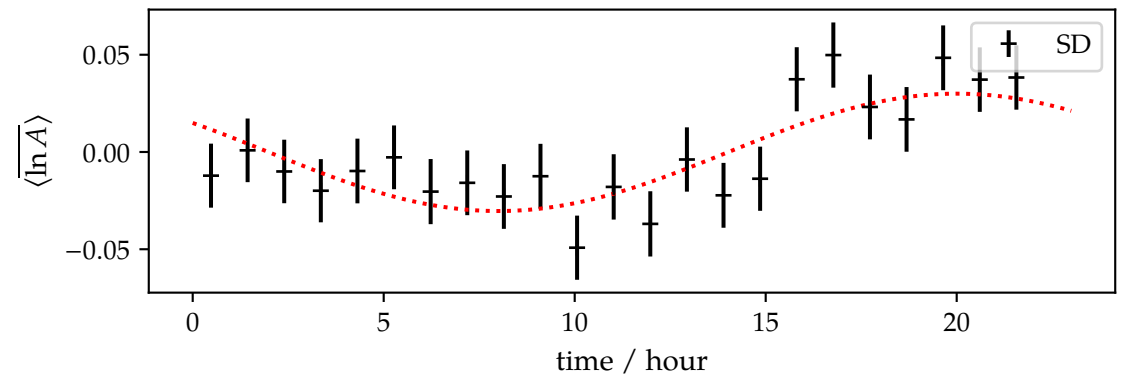


Figure D.56: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the time of day (cf. Fig. 8.8).

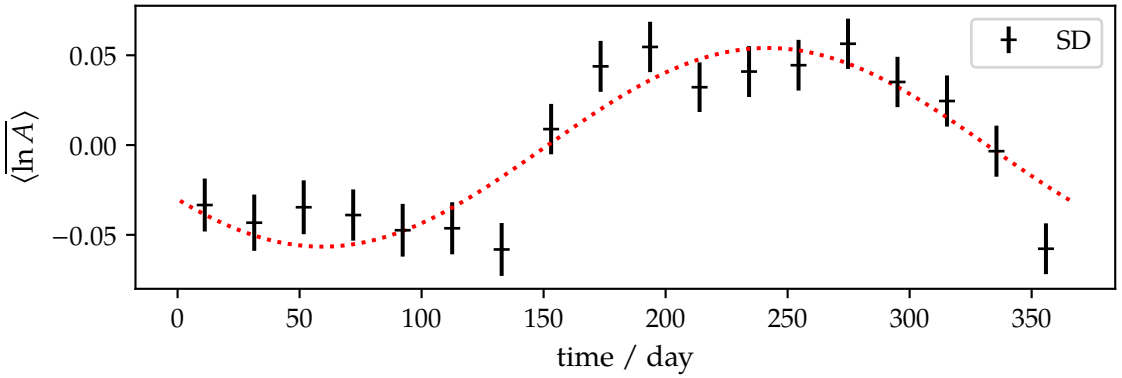


Figure D.57: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the day of the year (cf. Fig. 8.9).

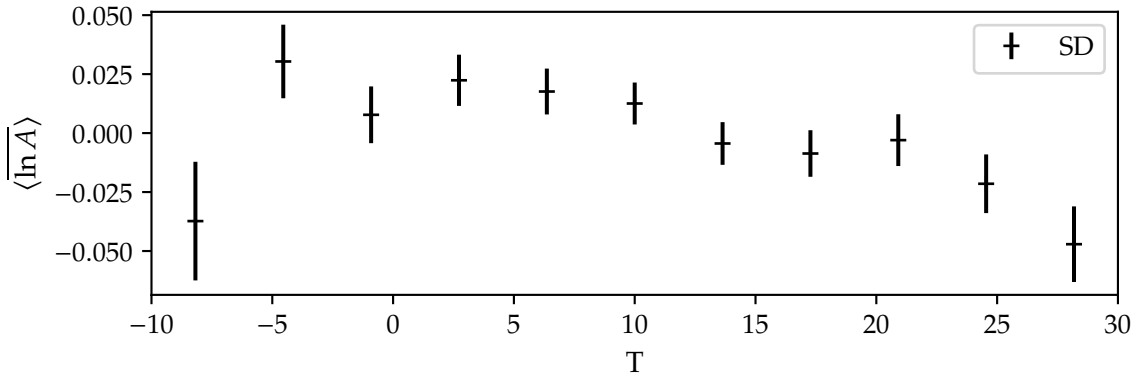


Figure D.58: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the temperature T (cf. Fig. D.39).

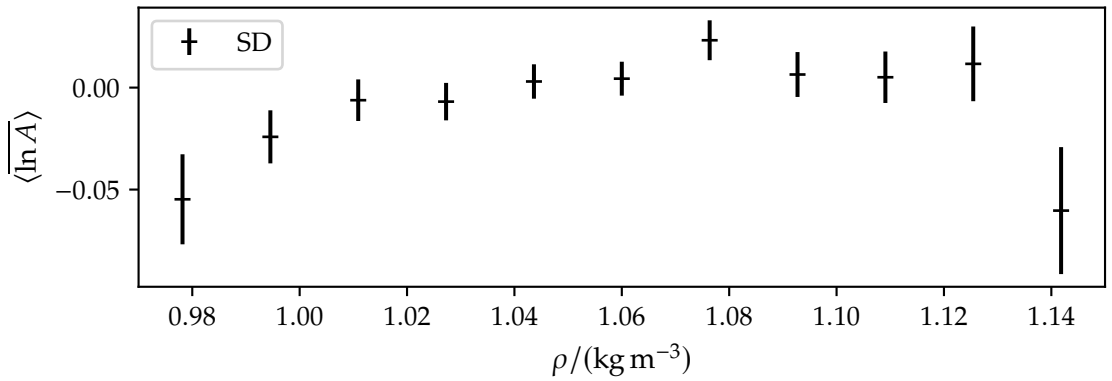


Figure D.59: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the air density ρ (cf. Fig. D.40).

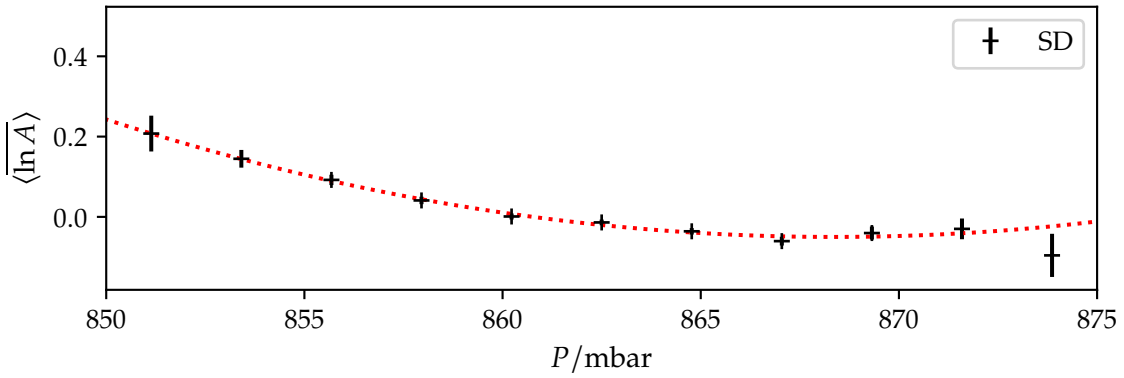


Figure D.60: Deviation from average behavior $\langle \overline{\ln A} \rangle$ as a function of the atmospheric pressure P (cf. Fig. 8.10).

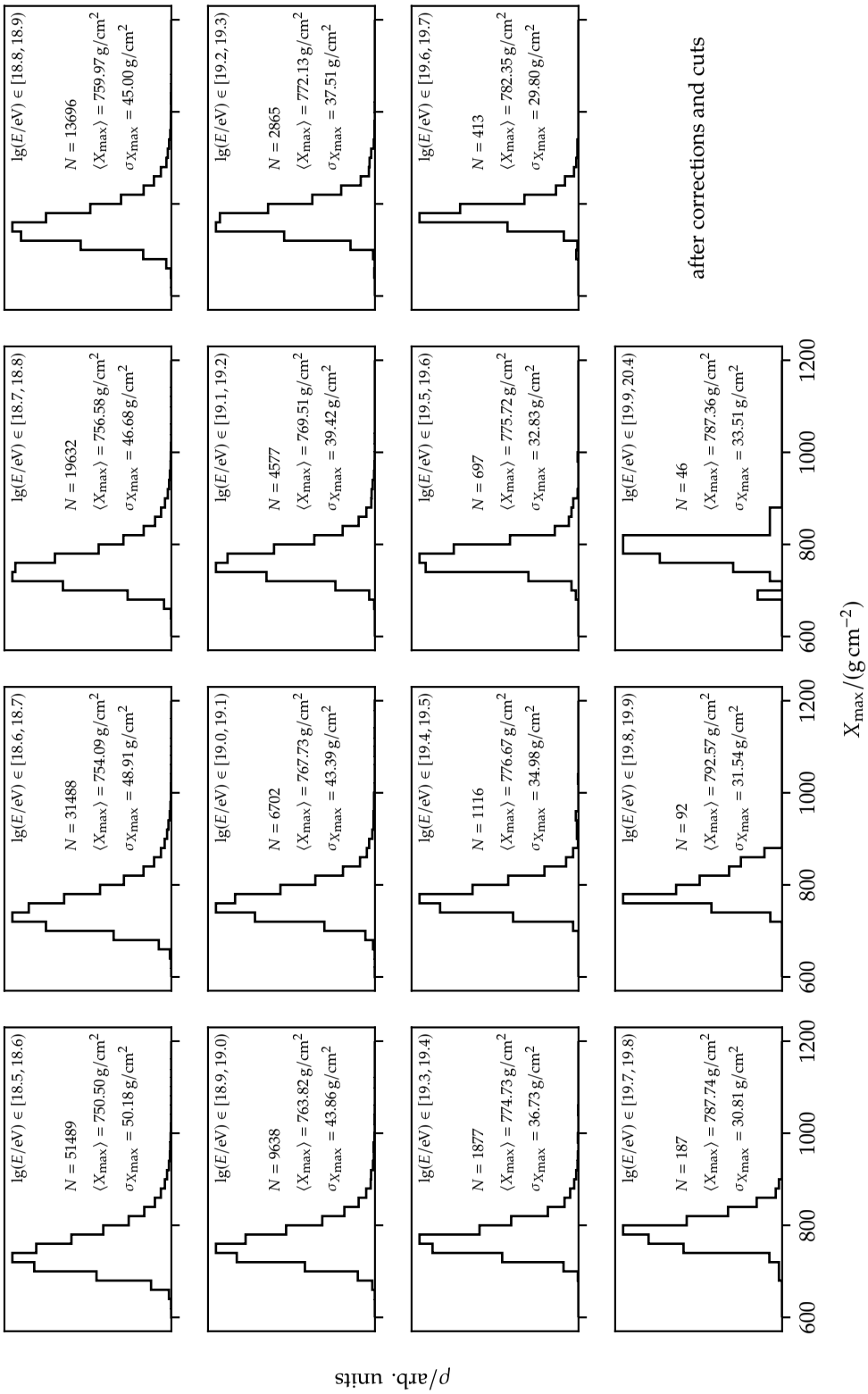


Figure D.61: Distribution of the $X_{\max}$ predictions in the bins shown in Fig. 8.16.

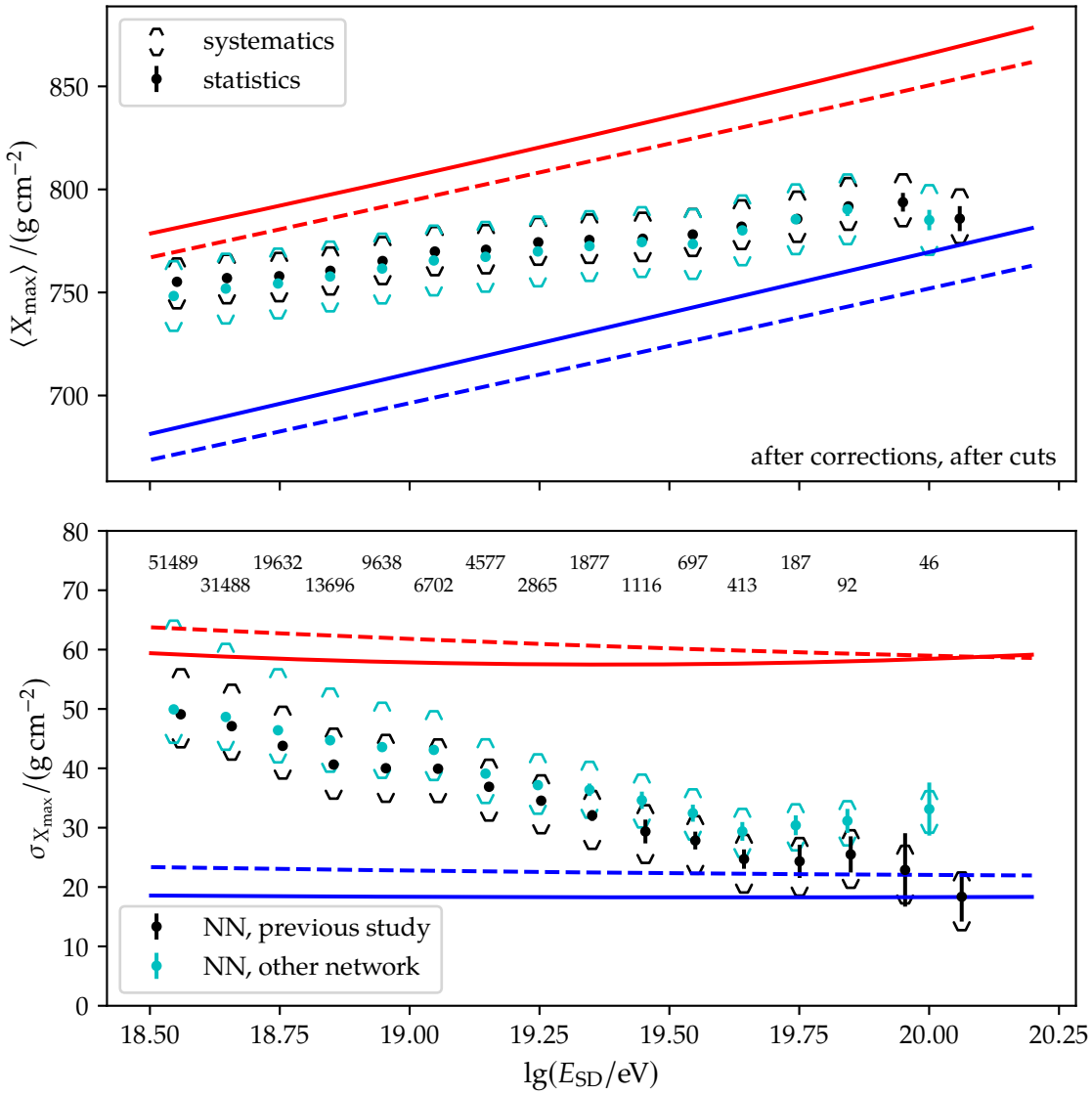


Figure D.62: First (*top*) and second moment (*bottom*) of X_{max} in bins of reconstructed energy E_{SD} for the NN of the study in [P:104] (black) and the corrected predictions of the NN on the quality-selected SD data set (cyan, see Table 8.1). The solid and dashed line show the expected behavior (see Sec. 5.4.2) for air showers induced by protons (red) and irons (blue) using the hadronic interaction models QGSJetII-04 and EPOS-LHC, respectively.

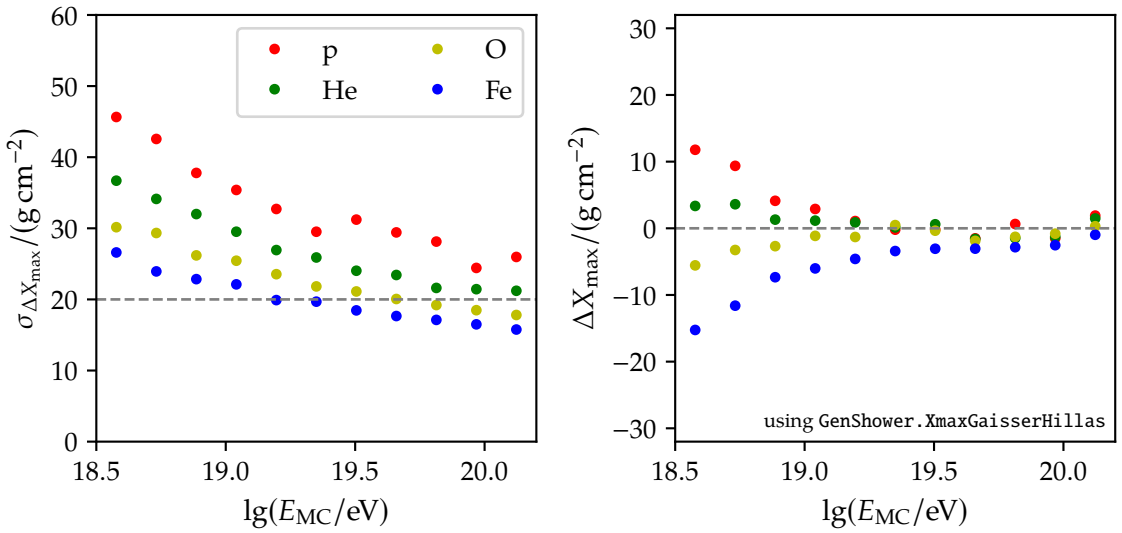


Figure D.63: Precision σ_{xmax} (*left*) and bias ΔX_{max} (*right*) for the predictions of the NN selected for the analysis in Sec. 8.2.2.A as a function of the logarithmic energy if `GenShower:XmaxGaisserHillas` is used as the true value for X_{max}.

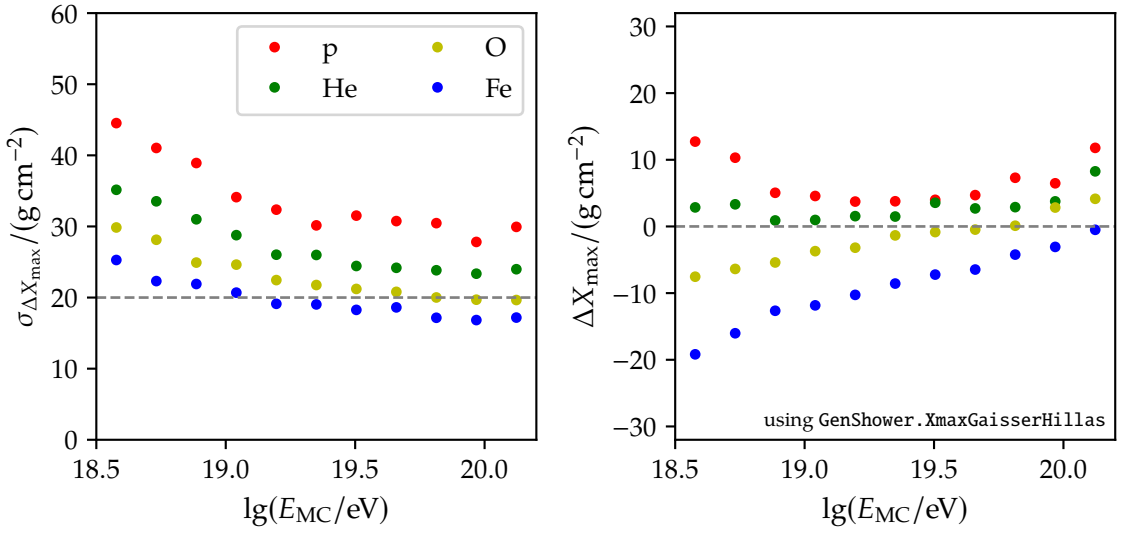


Figure D.64: Precision σ_{xmax} (*left*) and bias ΔX_{max} (*right*) for the predictions of an alternative NN that performs worse on simulations as a function of the logarithmic energy if `GenShower:XmaxGaisserHillas` is used as the true value for X_{max}.

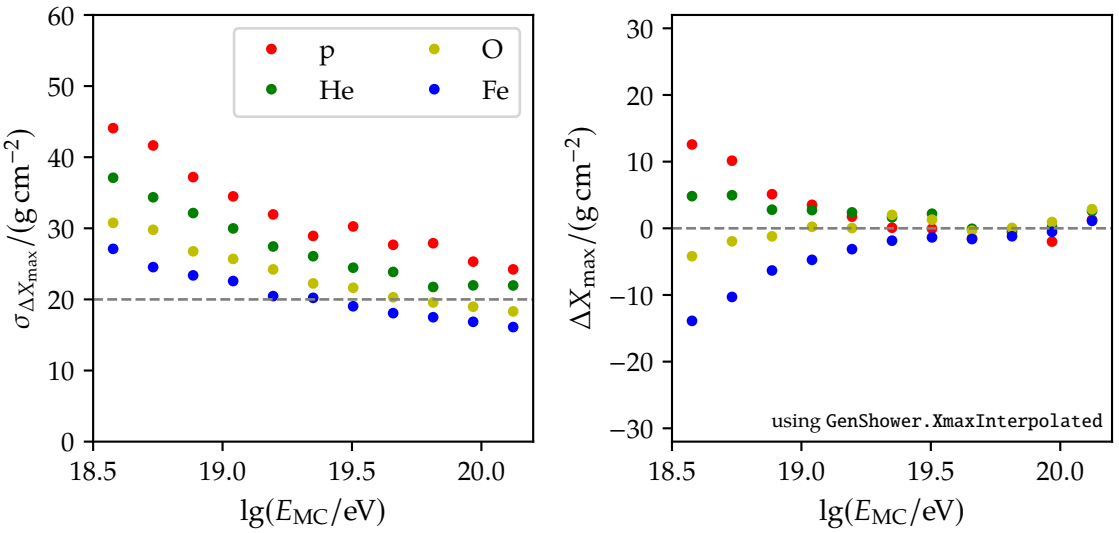


Figure D.65: Precision σ_{xmax} (*left*) and bias ΔX_{max} (*right*) for the predictions of the NN selected for the analysis in Sec. 8.2.2.A as a function of the logarithmic energy if `GenShower:XmaxInterpolated` is used as the true value for X_{max}.

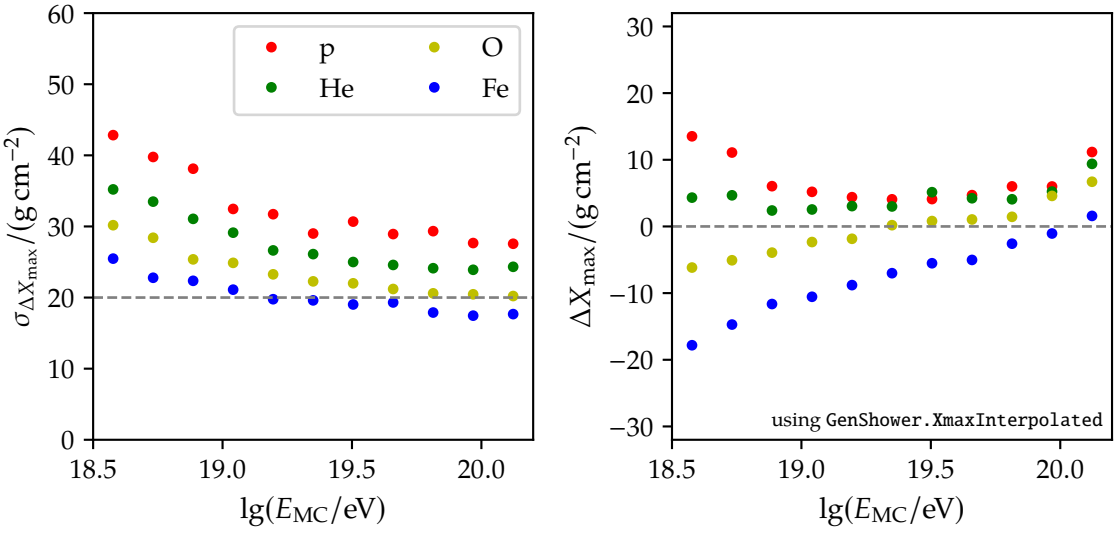


Figure D.66: Precision σ_{xmax} (*left*) and bias ΔX_{max} (*right*) for the predictions of an alternative NN that performs worse on simulations as a function of the logarithmic energy if `GenShower:XmaxInterpolated` is used as the true value for X_{max}.

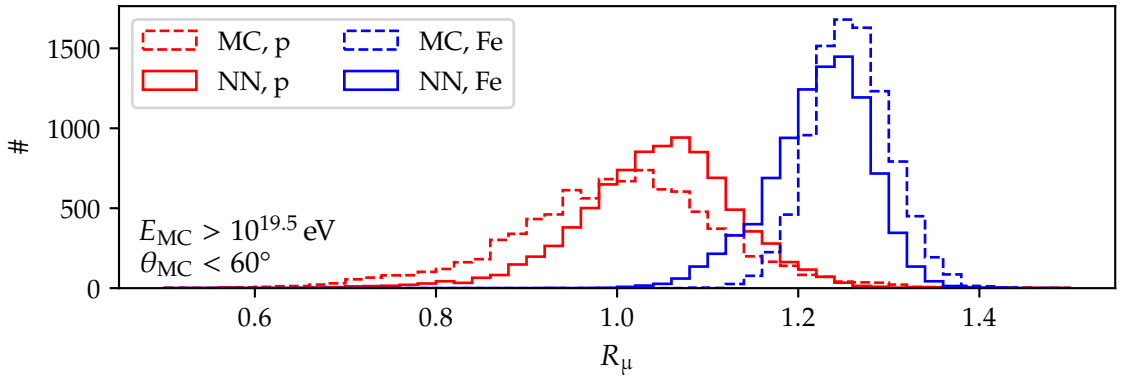


Figure D.67: Distribution of R_μ MC values (dashed) and NN predictions (solid) split into proton (red) and iron (blue) events.

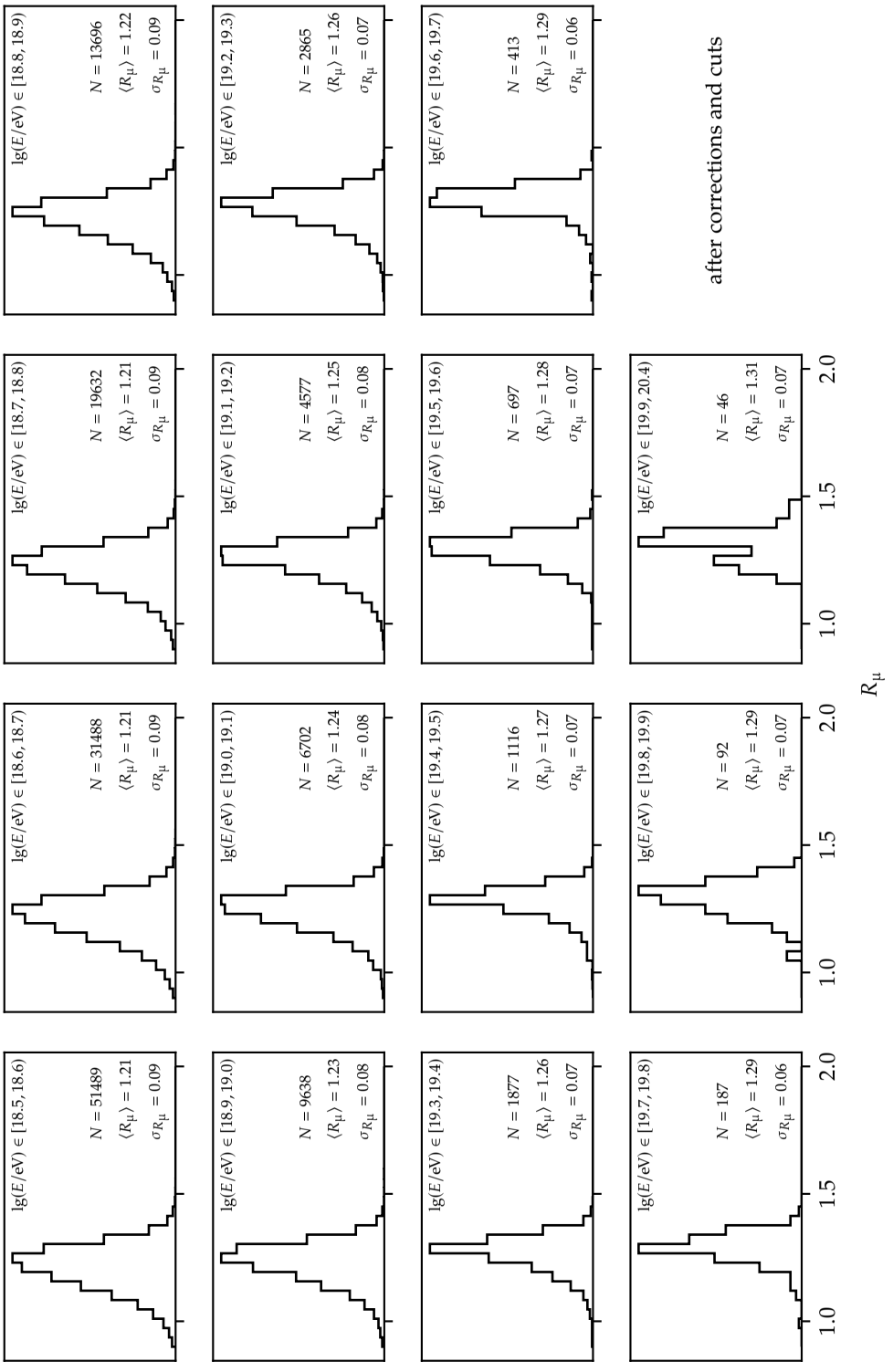


Figure D.68: Distribution of the R_μ predictions in the bins shown in Fig. 8.19.

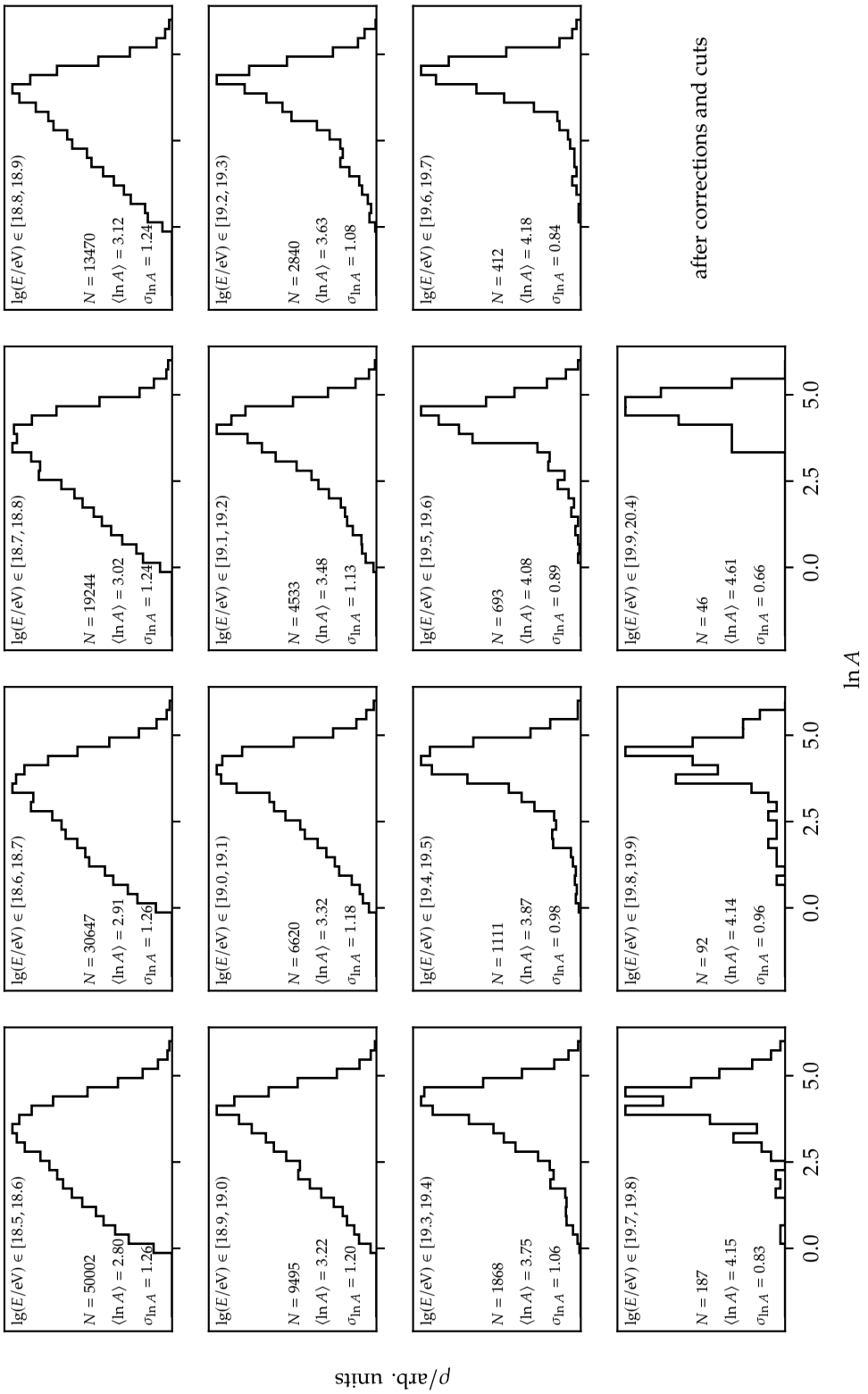


Figure D.69: Distribution of the raw $\ln A$ predictions in the bins shown in Fig. 8.20.