

# Unmasking the Gamma-Ray Sky

Comprehensive and Reproducible Analysis for Cherenkov Telescopes

by  
Kai Brügge

A document submitted in partial fulfillment of the requirements for the degree of

*Dr. rer. nat.*

at

TECHNISCHE UNIVERSITÄT DORTMUND

Supervised by

Prof. Dr. Dr. Wolfgang Rhode and Prof. Dr. Kevin Kröniger





## Abstract

Imaging atmospheric Cherenkov telescopes (IACT) observe the sky in the highest energy ranges. From the remnants of cataclysmic supernovae to jets powered by supermassive black-holes in the center of distant galaxies, IACTs can capture the light emerging from the most extreme sources in the universe.

With the recent advent of multi-messenger astronomy it has become critical for IACTs to publicly share their data and software. For the first time since the inception of IACT technology, in a combined effort of the H.E.S.S., MAGIC, VERITAS, and FACT collaborations, observations of the Crab Nebula were made available to the general public in a common data format. The first part of my thesis demonstrates the viability of the common data format by performing a spectral analysis of the Crab Nebula on the published datasets. The text gives detailed descriptions and mathematical formalizations of instrument response functions (IRFs) and the statistical modeling used for typical spectral analyses. This is essential to understand the measurement process of IACTs. The ultimate goal of this part of the thesis will be to use Hamilton Markov Monte Carlo methods to test spectral models and unfold flux-point estimates for the Crab Nebula.

The common data format paves the road for the operation of the upcoming Cherenkov Telescope Array (CTA). Once CTA has been constructed, it will be the largest and most sophisticated experiment in the field of ground-based gamma-ray astronomy. It will be operated as an open observatory allowing anyone to access the recorded data. The second part of my thesis concentrates on reproducible analysis for the Cherenkov Telescope Array (CTA). Once operational, CTA will produce a substantial amount of data creating new challenges for data storage and analysis technologies. In this part of the thesis I use simulated CTA data to build a comprehensive analysis chain based on fully open-source methods. The goal is to create a pipeline that rivals the physics performance of CTA's closed-source reference implementation. Every step of the analysis, from raw-data processing to the calculation of sensitivity curves, will be optimized with respect to complexity, reproducibility and run-time.

# Contents

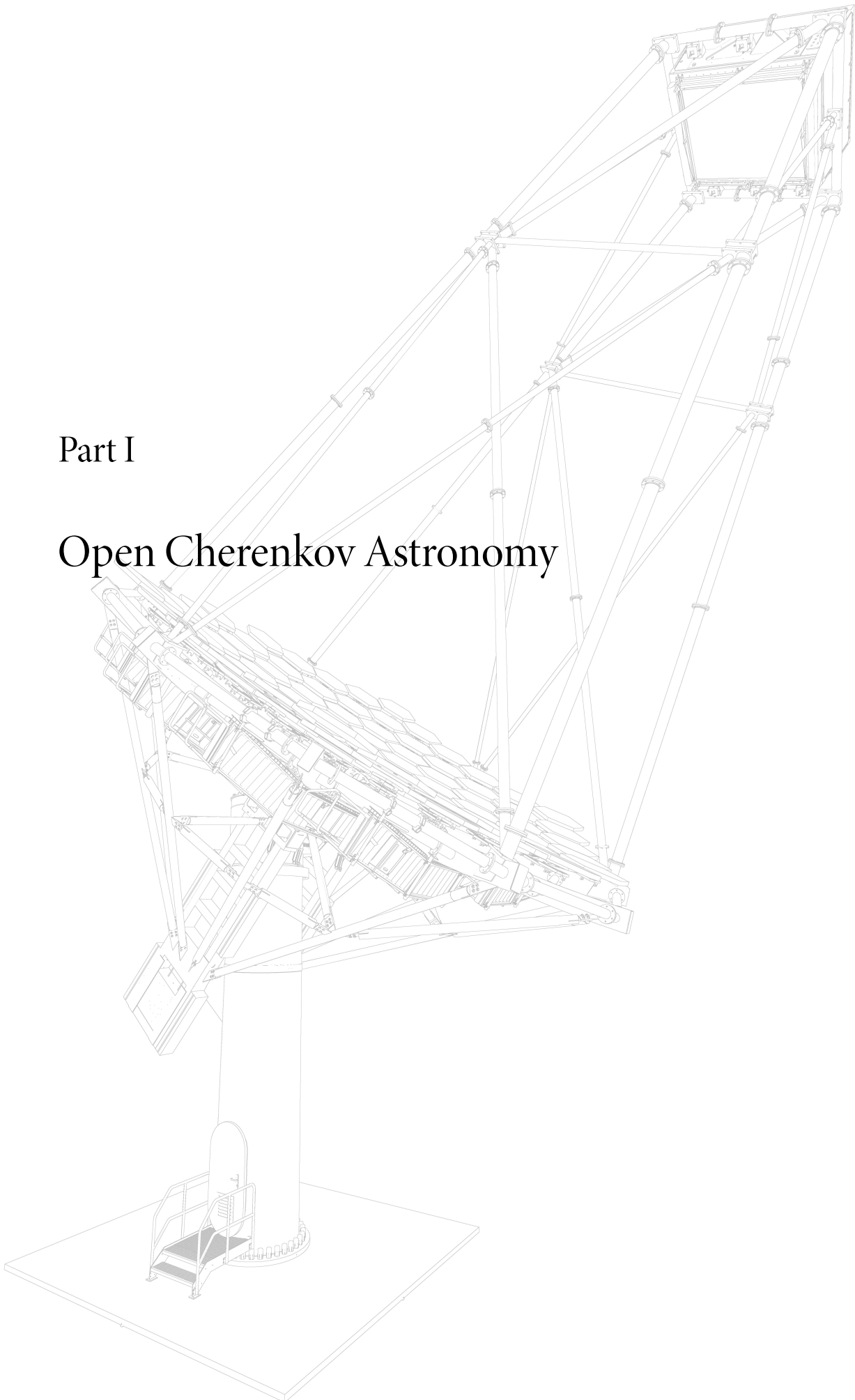
<b>I</b>	<b>Open Cherenkov Astronomy</b>	<b>1</b>
<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Acceleration of Cosmic Rays and Gamma Rays</b>	<b>5</b>
2.1	Fermi Acceleration . . . . .	8
2.2	Synchrotron Emission . . . . .	10
2.3	Inverse Compton Emission . . . . .	11
2.4	Log-Parabolic Energy Distributions . . . . .	13
<b>3</b>	<b>Observation of Very-High-Energy Gamma Rays</b>	<b>17</b>
3.1	Satellite Experiments . . . . .	17
3.2	Imaging Atmospheric Cherenkov Telescopes . . . . .	18
3.3	Water-Cherenkov Experiments . . . . .	23
<b>4</b>	<b>Modeling the Crab Nebula Emission</b>	<b>25</b>
<b>5</b>	<b>Spectral Analysis of IACT Data</b>	<b>31</b>
5.1	Event Data . . . . .	33
5.2	Instrument Response Functions . . . . .	34
5.3	Background Estimation . . . . .	41
5.4	Statistical Modeling . . . . .	42
5.5	Posterior Sampling using Markov Chain Monte Carlo . . . . .	45
5.6	Fitting Spectral Models . . . . .	48
5.7	Unfolding of Flux Points . . . . .	56
5.8	Notes on Uncertainty Propagation . . . . .	61
<b>6</b>	<b>Conclusion</b>	<b>63</b>

<b>II Reproducible Data Analysis for CTA</b>	<b>65</b>
<b>7 Introduction</b>	<b>67</b>
<b>8 Processing CTA Data</b>	<b>71</b>
8.1 Preprocessing Pipeline and Simulated Datasets . . . . .	72
8.2 Raw Data Processing . . . . .	75
8.3 Image Feature Extraction . . . . .	77
8.4 Geometrical Shower Reconstruction . . . . .	79
<b>9 Machine Learning</b>	<b>85</b>
9.1 Model Validation . . . . .	88
9.2 Decision Trees and Ensemble Learning . . . . .	91
9.3 AICT-Tools . . . . .	94
9.4 Application to CTA Data . . . . .	95
<b>10 A Prototype for Real Time Analysis</b>	<b>101</b>
<b>11 Sensitivity Computation</b>	<b>105</b>
11.1 Event Weights . . . . .	106
11.2 Background Estimation . . . . .	108
11.3 Detection Significance . . . . .	110
11.4 Sensitivity Computation . . . . .	113
<b>12 Notes on Reproducibility</b>	<b>119</b>
<b>13 Conclusion</b>	<b>121</b>
<b>Appendix</b>	<b>123</b>
<b>Bibliography</b>	<b>143</b>



Part I

# Open Cherenkov Astronomy







# 1 Introduction

The discovery of cosmic rays during the daring balloon flights of Victor Franz Hess in 1912 [80] opened up an entirely new window into the universe. This elusive radiation, which so relentlessly bombards us from outer space, carries a wealth of information about the most violent processes in the cosmos. Probing the gamma-ray sky is crucial to understanding the processes which drive the cosmic-ray acceleration. While charged cosmic rays are deflected by magnetic fields, gamma rays pinpoint back to the source and allow us to image these objects. Only the combination of data from multiple facilities can help to unravel the inner workings of cosmic-ray sources. The success of these joint campaigns has become apparent recently through the first observational evidence of neutrino emission from the blazar TXS 0506+056 [2]. This collective effort used data from the IceCube neutrino observatory in Antarctica as well as data from the gamma-ray experiments Fermi and MAGIC. From its very beginning, the Fermi collaboration made all of its recorded data and software available to the general public. With the recent advent of multi-messenger astronomy it has become critical for imaging atmospheric Cherenkov telescopes (IACT) to openly share their data as well. This first part of my thesis deals mainly with the computation of energy spectra and flux point estimates in an open and reproducible manner and tries to motivate the need for public IACT data. The text aims to convey all the essential information that is needed in order to understand the measurement process of IACTs.

The galactic cosmic ray population is seeded by remnants of cataclysmic supernova events. The Crab Nebula is the prototypical object of this kind. It is a steady source of bright TeV gamma-ray emission in the northern sky and is continuously observed by radio, X-ray, and gamma-ray observatories. It was the first gamma-ray source detected by an imaging atmospheric Cherenkov telescope (IACT) in 1989 [143]. In chapter 2 I describe the typical emission processes prevalent in supernova remnants (SNR) and describe how log-parabolic energy spectra emerge in many sources of cosmic rays. Chapter 3 gives an overview of experimental techniques used to observe gamma rays. In chapter 4 I show that the spectral energy distribution of the Crab Nebula, in an energy range from a few keV to tens of TeV, can be described with a single electron population of log-parabolic shape. In that chapter I model the synchrotron, inverse Compton, and Synchrotron Self-Compton emission using the `naima` [148]

## 1 Introduction

program and fitted the model to data from six different experiments using Markov chain sampling. This emphasizes the importance of open data in the Cherenkov astronomy community. Without open access to flux data from multiple wavelengths, no model assumptions can be validated.

Chapter 5 goes through all the harrowing details needed in order to understand the measurement process of IACTs, a critical part of which is the computation of the instrument response functions. In section 5.2 these response functions are formalized in an instrument-agnostic way, the ultimate goal of which is the construction of a common data format that can be used by any IACT instrument. Traditionally, all IACT collaborations use their own proprietary software and data formats to produce high-level results like flux points. In a mutual endeavor between the MAGIC, H.E.S.S., VERITAS, and FACT collaborations to change the current state of affairs, a small data sample of Crab Nebula observations was made public in this common data format. Using this data we published the first joint and fully open-source analysis of combined IACT data [122]. In section 5.6 I use the published event lists and instrument response functions to fit a log-parabolic spectral model to the IACT data using Hamilton Markov chain sampling. Section 5.7 shows how that same data can be used to unfold the flux points for each individual instrument.

## 2 Acceleration of Cosmic Rays and Gamma Rays

Earth's atmosphere is constantly being bombarded by very-high-energy particles of cosmic origin. Among these are photons, neutrinos, electrons, protons and heavier nuclei. The astroparticle community usually refers to the hadronic particles as *cosmic rays*, while neutrinos and high-energy photons are viewed as separate entities. The discovery of cosmic rays and their byproducts in the atmosphere is attributed to the balloon experiments performed by Victor Hess in the year 1912 [38, 80]. The term *cosmic rays* was coined by Robert A. Millikan 19 years later [115] after performing his own observations of ionizing radiation at several different altitudes. Cosmic rays cover a vast range of energy spanning 15 orders of magnitude from a few keV to several EeV. One major goal in the field of cosmic-ray astronomy is to learn more about the origin of these cosmic messengers. The key mechanism that drives cosmic rays is believed to be shock acceleration. Satellites have gathered direct evidence of particle acceleration in the keV to MeV range in interplanetary shocks in our solar system. Even in our direct neighborhood, Earth's bow shock with the surrounding medium, particles are accelerated to higher energies [69]. Interactions in the hot and dense plasma ejected by solar flares accelerate particles up to GeV energies [67, page 236]. Beyond GeV energies, however, the engines for cosmic particle acceleration lie outside of our solar system. The population of galactic cosmic rays is most likely driven by shock acceleration in supernovae remnants within the Milky Way. At even higher energies, beyond EeV, galactic sources cannot account for the observed fluxes. Protons at these energies are not confined by the magnetic fields in our galaxy and can escape into the intergalactic regions. The confinement of a particle is governed by its radius of gyration. A particle with rest mass  $m_0$  and Lorentz factor  $\gamma$  has kinetic energy  $E_{\text{kin}}$

$$E_{\text{kin}} = (\gamma - 1)m_0c^2.$$

Given the cosmic ray energy of 1 EeV and rearranging for  $\gamma$  yields

$$\gamma = \frac{E_{\text{kin}}}{m_0c^2} + 1 \approx 1.1 \times 10^9.$$

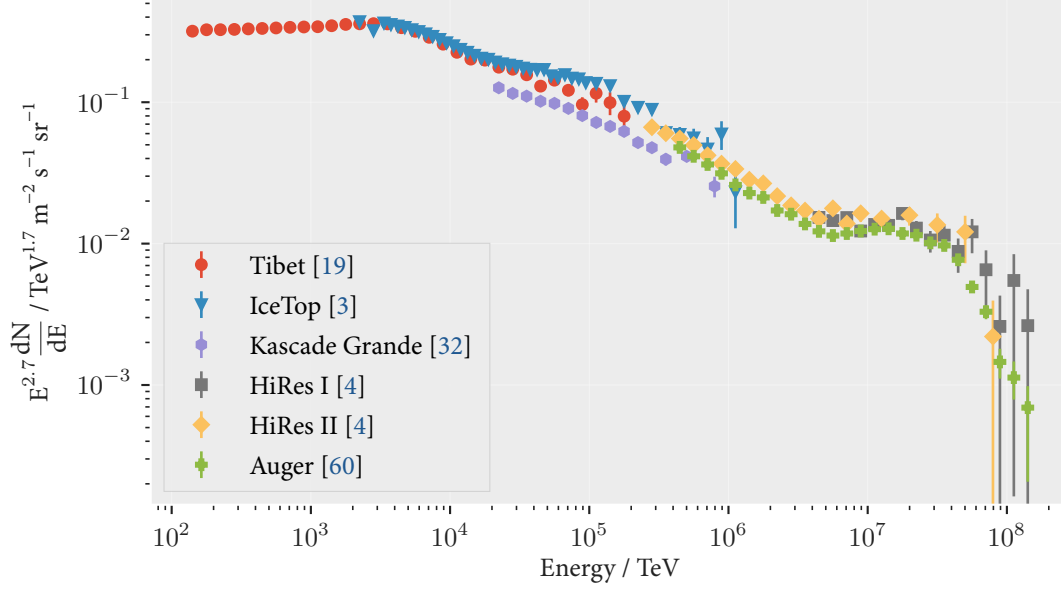
## 2 Acceleration of Cosmic Rays and Gamma Rays

The gyroradius  $r$  of a charged particle with charge  $q$  which is moving at velocity  $v$  perpendicular to a magnetic field of strength  $B$  is

$$r = \frac{p}{qB} = \frac{\gamma m v}{qB}.$$

Assuming a mean magnetic field of  $B = 5 \times 10^{-10}$  T within the Milky Way [78], results in a gyration radius of approximately  $r = 5 \times 10^{18}$  m = 528 ly. At these energies particles can escape the local arm of the galaxy. Hence, particles with higher energies most likely originate in extragalactic sources. Jets of active galactic nuclei (AGN) are widely accepted to be the source of these particles. Figure 2.1 shows the differential energy spectrum of cosmic rays. By multiplying the flux with  $E^{2.7}$  the steep power law becomes flat. The first bend in the spectrum around  $1 \times 10^3$  TeV is often called the *knee*. The second bend is called the *ankle* near  $3 \times 10^6$  TeV and its often associated with the appearance of extra-galactic particles. The shape of the spectrum between the knee and the ankle is still matter of some debate. The steep cutoff visible near  $5 \times 10^7$  TeV is assumed to be due to the Greisen–Zatsepin–Kuzmin (GZK) cutoff [67, page 209]. The origin and composition of cosmic rays beyond the cutoff is still unclear. Shock acceleration can be modeled by a process called *Fermi acceleration* and a *plasma dynamic* approach. Both approaches predict power-law like flux. They are briefly described in section 2.1.

The production and acceleration of cosmic rays is strongly intertwined with the production of gamma rays and neutrinos. Differential energy spectra, or flux curves, show the energy resolved light emission of an astrophysical source. In the gamma-ray community they are usually given in units of  $\text{cm}^{-2} \text{s}^{-1} \text{TeV}^{-1}$ . The flux is often scaled by the square of the photon energy  $E^2$  for better visual representation leading to units of  $\text{cm}^{-2} \text{s}^{-1} \text{TeV}$ . This scaled quantity is often called a spectral energy distribution (SED) [67, page 3]. Both  $\nu f_\nu$  and  $E^2 \frac{dN}{dE}$  are common shorthands for these SEDs. I will use the latter designation in this text. SEDs are of major interest to many astronomers, as their shape can be used to validate or invalidate models of acceleration mechanisms in these sources. SEDs often span more than 15 orders of magnitude in photon energy and combine a multitude of instrument technologies and disciplines. The gamma rays in the very-high-energy range of the SEDs, above 100 GeV, are produced by either synchrotron radiation or inverse Compton (IC) scattering on photon fields. Sections 2.2 and 2.3 give a little more detail about the shape of synchrotron and IC spectra for supernova remnants. Bright sources are extensively monitored by radio, infrared, optical, X-ray and gamma-ray telescopes. The most prominent source of gamma radiation within our galaxy is the Crab Nebula. It is a supernova remnant which exploded in 1054. The event was recorded by Chinese astronomers which reported a “Guest Star” which had been



**Figure 2.1:** All particle cosmic-ray spectrum measured by the Auger, IceTop, Tibet, Cascade, and HiRes experiments. The energies span 10 orders of magnitude and range from a few GeV to some 100 EeV. The plot only shows the high-energy end of the spectrum starting range from 100 TeV. Cosmic rays can reach energies equivalent of a baseball flying at 50 km/h. The plotted flux is multiplied by  $E^{2.7}$  to highlight the structural features of the spectrum. The *knee* and *ankle* are visible at approximately 1 PeV and 5 EeV.

visible for three weeks during daytime [81]. A scan of the original report by the astronomer and a translation can be found in appendix A.3. The matter ejected during the explosion has since expanded into a shell of hot plasma with a radius of about 1.5 lightyears [81]. Extensive observations from in the radio, optical, X-Ray, and gamma-ray bands, have made it one of the most popular objects for astronomers to study. For very-high-energy (VHE) gamma-ray astronomers in particular, the Crab Nebula is of major interest due to its steady and bright emission of photons. In Cherenkov astronomy it is common to test new data analysis techniques on data from *the Crab*, as it is often called in the community. The analysis I performed for this thesis is no exception.

## 2.1 Fermi Acceleration

In 1949 Enrico Fermi published a model to explain the high energies observed by cosmic-ray detectors and the power-law shape of the cosmic-ray spectra [61]. Prior to Fermi's publication the origin of the cosmic rays at the highest energies was unclear. Fermi summarized the situation as follows:

The argument against the conventional view that cosmic radiation may extend at least to all the galactic space is the very large amount of energy that should be present in form of cosmic radiation if it were to extend to such a huge space. Indeed, if this were the case, the mechanism of acceleration of the cosmic radiation should be extremely efficient.

The principal idea published by Fermi is that particles, on average, gain energy by *collisions* with randomly moving magnetic fields. These *magnetic mirrors* were believed to be clouds of plasma moving at high velocities in random directions. As to the origin of these fields, no explanation was given. The power-law energy spectrum for this type of particle acceleration follows from stochastic arguments [67, §12.2.1]. Assume some test particle gains energy  $\Delta\gamma = \alpha\gamma$  in each collision with the magnetic mirror. After  $n$  collisions the energy will be

$$\begin{aligned}\gamma_n &= \gamma_{n-1} + \gamma_{n-1}\alpha \\ \gamma_n &= \gamma_{n-1}(1 + \alpha) \\ \gamma_n &= \gamma_{n-2}(1 + \alpha)(1 + \alpha) \\ &\vdots \\ \gamma_n &= \gamma_0(1 + \alpha)^n,\end{aligned}$$

where  $\gamma_0$  is the initial energy of the particle. Solving this equation for  $n$  yields

$$n = \frac{\log\left(\frac{\gamma_n}{\gamma_0}\right)}{\log(1 + \alpha)}.$$

In this scenario, the particle keeps bouncing around between the moving magnetic fields until it escapes the acceleration region. Let  $P_{\text{esc}}$  be the constant probability for a particle to escape its confinement at any given time. Then, after  $n$  collisions, the probability for the particle



to remain within the acceleration region is  $P_n = (1 - P_{\text{esc}})^n$ . Substituting  $n$  and using the identity  $x^{\log(y)} = y^{\log(x)}$  results in

$$\begin{aligned}
 P_n &= (1 - P_{\text{esc}})^n \\
 &= (1 - P_{\text{esc}})^{\frac{\log\left(\frac{\gamma_n}{\gamma_0}\right)}{\log(1+\alpha)}} \\
 &= \left( (1 - P_{\text{esc}})^{\log\left(\frac{\gamma_n}{\gamma_0}\right)} \right)^{\frac{1}{\log(1+\alpha)}} \\
 &= \left( \frac{\gamma_n}{\gamma_0} \right)^{\frac{\log(1-P_{\text{esc}})}{\log(1+\alpha)}} \\
 &= \left( \frac{\gamma_n}{\gamma_0} \right)^{-s},
 \end{aligned}$$

where  $s = \log\left(\frac{1}{1-P_{\text{esc}}}\right)/\log(1+\alpha)$ . This acceleration model results in an energy spectrum which follows the power-law distribution so ubiquitous in astroparticle physics. Fermi's model was able to successfully explain the experimental data recorded by cosmic-ray researchers at the time.

Another approach to the problem, which is perhaps a little more motivated by physics rather than statistics, can be build from the so-called *diffusion-loss* equation. In this explanation, the magnetic fields described by Fermi are due to turbulent plasmas giving rise to "random" movements and strong magnetic field gradients. Following the notation in [103, page 566] the diffusion equation is defined as

$$\frac{dN(\gamma)}{dt} = D\nabla^2 N(\gamma) - \frac{d}{d\gamma}(\dot{\gamma}N(\gamma)) - \frac{N}{t_{\text{esc}}} + Q(\gamma), \quad (2.1)$$

where  $D$  is the diffusion coefficient,  $Q(\gamma)$  is a particle source term and  $t_{\text{esc}}$  is the typical escape time for a particle. Fermi's approach assumes a steady-state configuration without diffusion or source terms. So  $Q(\gamma) = 0$  and  $D\nabla^2 N = 0$ . As seen above, the energy-gain term in the Fermi approach is postulated as  $\frac{d\gamma}{dt} = \alpha\gamma$ . This results in the simplified equation

$$\begin{aligned}
 -\frac{d}{d\gamma}(\alpha\gamma N(\gamma)) &= \frac{N}{t_{\text{esc}}} \\
 N(\gamma) + \gamma \frac{dN(\gamma)}{d\gamma} &= -\frac{N(\gamma)}{\alpha t_{\text{esc}}} \\
 \frac{dN(\gamma)}{d\gamma} &= -\left(1 + \frac{1}{\alpha t_{\text{esc}}}\right) \frac{N(\gamma)}{\gamma}.
 \end{aligned} \quad (2.2)$$

Equation (2.2) has a solution of the form

$$N(\gamma) \propto \gamma^{-\left(1 + \frac{1}{\alpha t_{\text{esc}}}\right)} \propto \gamma^{-s}. \quad (2.3)$$

The diffusion equation results in a power-law spectrum just like the stochastic approach. The scenario Fermi described does introduce some problems, a discussion of which can be found in [103, page 566]. The energy gain per collision using the original Fermi model is proportional to the square of the magnetic field velocity  $V$  i.e:  $\Delta E \approx \beta^2 = (V/c)^2$ . Hence, it is often called *second-order* Fermi acceleration. This process alone is not efficient enough to explain the abundance of cosmic rays. The *first-order* Fermi mechanism was discovered in the 1970s and takes place in the presence of strong shock waves [103, page 569]. Supernova remnants (SNR) provide the perfect conditions for shock wave acceleration i.e. first order Fermi acceleration. Exploding stars hurl massive amounts of matter into the space. The discarded shell expands rapidly into the surrounding interstellar medium. A detailed discussion on acceleration in planar shock waves can be found in [67, §12.2.2]. Hence, supernova remnants like Cassiopeia A and the Crab Nebula are the perfect test objects to study models of cosmic-ray acceleration.

## 2.2 Synchrotron Emission

A charged particle moving through a magnetic field radiates energy in form of light. The breeding grounds of cosmic rays inevitably produce photons which can be observed on Earth. Hot gas and matter radiates thermal blackbody radiation. The largest part of the observed SEDs is produced by synchrotron radiation. It dominates the electromagnetic energy output over a broad range of wavelengths. Suppose a single relativistic electron with Lorentz factor  $\gamma$  moving in a magnetic field produces a synchrotron radiation spectrum  $F(\nu)$ . Then an electron population with distribution  $dN(\gamma)/d\gamma \propto \gamma^{-s}$  leads to a radiated synchrotron spectrum of

$$\frac{dN(\nu)}{d\nu} \propto \int dN(\gamma)/d\gamma F(\nu) d\gamma \quad (2.4)$$

$$\propto \int \gamma^{-s} F(\nu) d\gamma. \quad (2.5)$$

The synchrotron spectrum of a single relativistic electron with Lorentz factor  $\gamma$  moving in a magnetic field peaks strongly at

$$\nu_s = \gamma^2 \nu_c = \gamma^2 \frac{Be}{2\pi m_e} \quad [67, \text{page 284}]. \quad (2.6)$$

If the power radiated per differential energy by a single electron is approximated by its peak frequency we get

$$\frac{dN(\nu)}{d\nu} \propto \int \gamma^{-s} \delta(\nu - \gamma^2 \nu_c) d\gamma$$

Substituting  $x = \gamma^2 \nu_c$  and  $d\gamma = 2\gamma \nu_c dx = 2\left(\frac{x}{\nu_c}\right)^{1/2} \nu_c dx$  yields

$$\begin{aligned} \frac{dN(\nu)}{d\nu} &\propto \int \gamma^{-s} \delta(\nu - x) dx \\ &\propto \int \left(\frac{x}{\nu_c}\right)^{\frac{-s}{2}} \delta(\nu - x) \frac{1}{2\left(\frac{x}{\nu_c}\right)^{1/2}} dx \\ &\propto \int \left(\frac{x}{\nu_c}\right)^{\frac{-s-1}{2}} \delta(\nu - x) dx \\ &\propto \left(\frac{\nu}{\nu_c}\right)^{-\frac{s+1}{2}}. \end{aligned} \quad (2.7)$$

The synchrotron spectrum emitted by the electrons will follow the electrons' power-law shape with a modified spectral index. The energy loss introduced to the original electron population due to synchrotron radiation changes the injected power-law spectrum. Suppose a Fermi process injects a power-law electron distribution with spectral index  $s$ . Above some fixed break energy  $\gamma_{\text{break}}$  the synchrotron loss steepens the electrons spectrum to  $\gamma^{-(s+1)}$ . The injected electron spectrum hence changes to

$$\frac{dN(\gamma)}{d\gamma} \propto \begin{cases} \gamma^{-s} & \text{if } \gamma < \gamma_{\text{break}} \\ \gamma^{-(s+1)} & \text{else.} \end{cases} \quad (2.8)$$

This in turn changes the spectrum of the synchrotron emission as seen in (2.7). Many SEDs of active galactic nuclei and supernova remnants show a very distinct synchrotron *bump*. Measuring the synchrotron spectra gives direct evidence of the electron population in the source. The second major feature seen in many spectra is a consequence of the inverse Compton effect.

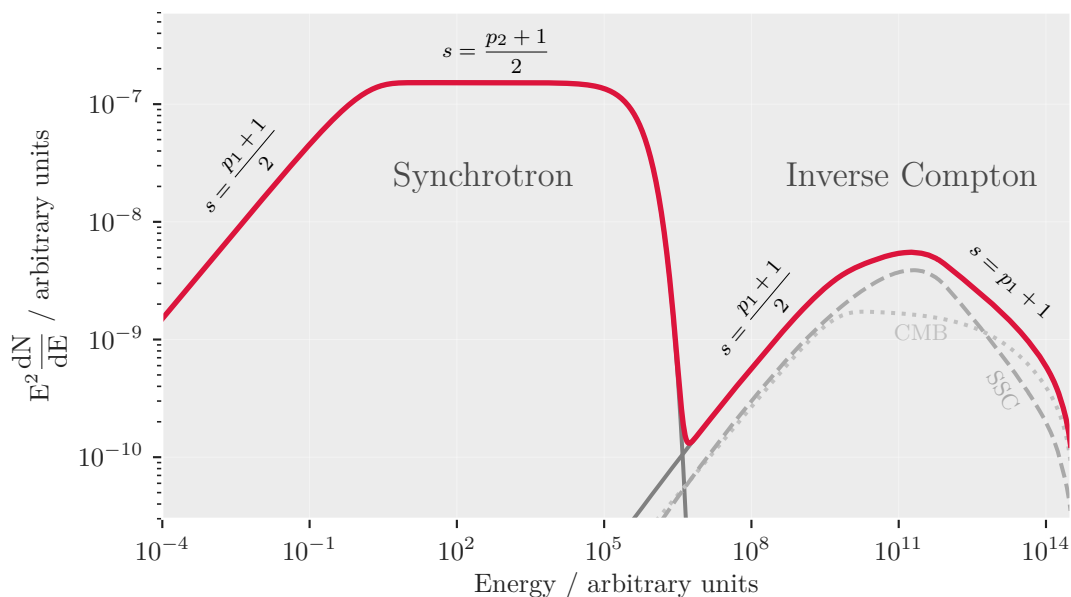
## 2.3 Inverse Compton Emission

Inverse Compton (IC) interaction is the driving mechanism for very-high-energy gamma rays detected from supernova remnants. Cherenkov telescopes almost exclusively observe

the inverse Compton emission of these sources. The Compton effect describes the scattering of a photon with an electron. Given an electron at rest, the incident photon will change its direction and wavelength during the scattering process. It is named after Arthur Compton, who was the first to publish a quantitative explanation of the effect in 1923. The Compton effect is of some historical importance since it has no satisfactory explanation in a pure wave-like description of light. The wavelength shift of the scattered photon can only be explained with the particle nature of light. In the inverse Compton effect, the electrons are no longer considered to be at rest. A fast-moving electron transfers its energy to a photon during scattering. High-energy electron populations in SNRs can interact with any of the surrounding photon fields this way, be it background starlight, infrared emission from dust clouds, or the cosmic microwave background (CMB). The photons radiated via synchrotron emission can also seed the inverse Compton process. In sources like the Crab Nebula, this so-called *Synchrotron Self-Compton* (SSC) interaction is the key component in the VHE gamma-ray emission measured by Cherenkov telescopes. Approximating the seed photon fields as monochromatic, the shape of the inverse Compton spectrum roughly follows by a broken power law as shown by [34, 64, 98]

$$\frac{dN(\nu)}{d\nu} \propto \begin{cases} \left(\frac{\nu}{\nu_c}\right)^{-\frac{s+1}{2}} & \text{for } \nu h \ll m_e c \\ \left(\frac{\nu}{\nu_c}\right)^{-(s+1)} & \text{else.} \end{cases} \quad (2.9)$$

Here  $m_e c$  is the electron's rest energy. The inverse Compton effect produces the second large *bump* in the SEDs of many sources. Combining the shape of the synchrotron photon spectra with the IC photon spectra we can now draw the SED of a typical gamma-ray source like the Crab Nebula. Figure 2.2 shows a drawing of an SED as it would be observed from a hypothetical source along with its main features and dependencies.



**Figure 2.2:** A sketch of a typical SED with Synchrotron Self-Compton emission as observed in many supernova remnants. The SED was simulated using `na.ima` [148]. The underlying electron spectrum was assumed to be distributed according to a simple power law with index  $p_1 = 2$  before the electron population starts to cool and index  $p_2 = 3$  after [91]. The cutoff energy at which the cooling sets in is fixed at  $10^{12}$  eV. As shown in (2.7) the synchrotron spectrum follows the electron spectrum with a modified index. The right side of the figure shows the inverse Compton bump. The IC emission from the CMB photons and the synchrotron photons is drawn separately as dotted and dashed lines respectively. This figure was adapted from Stefan Funk’s review article “Ground- and Space-Based Gamma-Ray Astronomy” [66].

## 2.4 Log-Parabolic Energy Distributions

The inverse Compton emission measured by Cherenkov telescopes often shows remarkable curvature. Extra-galactic sources like Markarian 421 as well as SNRs within our own galaxy show a curved photon spectrum at high energies. The approximation made in section 2.3 describing the spectrum as a power law seems to be too crude. In the previous section I assumed an electron population that was produced by a *Fermi-like* process resulting in a power-law distribution. Each *collision* with a magnetic field resulted in an energy gain of  $\Delta\gamma = \alpha\gamma$ . We can adapt the energy gain processes slightly to account for some randomness during the process. Following ideas from [141], I consider a single charged particle in prox-

## 2 Acceleration of Cosmic Rays and Gamma Rays

imity to moving magnetic fields. We can express the energy of the particle using its Lorentz factor after each collision with the magnetic field as

$$\begin{aligned}
 \gamma_n &= \epsilon_n \gamma_{n-1} \\
 &= \epsilon_n (\epsilon_{n-1} \gamma_{n-2}) \\
 &= \epsilon_n (\epsilon_{n-1} (\epsilon_{n-2} \gamma_{n-3})) \\
 &\vdots \\
 &= \gamma_0 \prod_{i=1}^n \epsilon_i,
 \end{aligned}$$

where  $\epsilon_i$  is the energy gain received by the particle in collision  $i$  and  $\gamma_i$  is its Lorentz factor. Suppose the particle starts with a low kinetic energy, we can set  $\gamma_0 = 1$  and rearrange the equation a bit

$$\begin{aligned}
 \gamma_n &= \prod_{i=1}^n \epsilon_i \\
 &= e^{\ln(\prod_{i=1}^n \epsilon_i)} \\
 &= e^{\sum_{i=1}^n \ln(\epsilon_i)} \\
 &= e^{\sum_{i=1}^n X_i},
 \end{aligned}$$

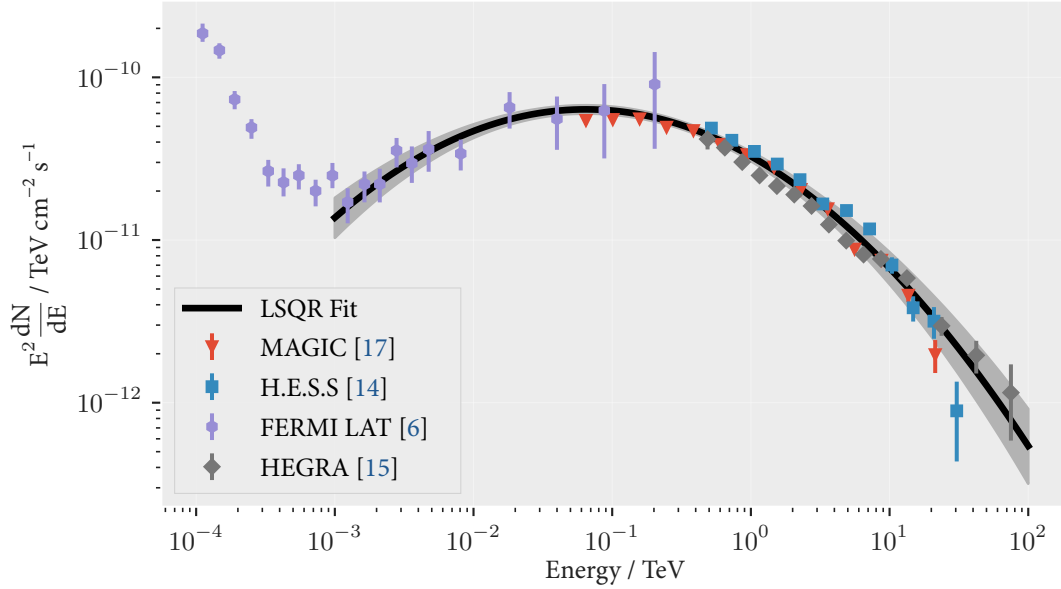
where we set  $X_i = \ln(\epsilon_i)$ . The  $X_i$  are assumed to be identically distributed, independent of each other and have finite variances. This may sound like a bold claim at first. However, the mechanisms taking place in each collision are bound to the same physical processes and completely uncorrelated with each other. Hence, the random variables  $X_i$  fulfill the conditions for applying the central limit theorem. It follows that for large  $n$  the sum  $\chi = \sum_{i=1}^n \ln(\epsilon_i)$  converges in distribution to a normal distribution  $\chi \sim \mathcal{N}(\mu, \sigma)$  with  $\mu = n\mu(\ln(\epsilon_i))$  and  $\sigma^2 = n\sigma^2(\ln(\epsilon_i))$ . In consequence, the energy distribution of the particles  $f(\gamma)$  will follow a log-normal distribution [50, page 312]

$$\ln(f(\gamma)) \sim \mathcal{N}(\mu, \sigma^2).$$

Transforming  $\chi$  into a standard normal variable using  $Z = \frac{\chi - \mu}{\sigma}$  we can write

$$f(E) \sim e^{\mu + \sigma Z}.$$





**Figure 2.3:** Simple least-squares fit of a log-parabolic model to observations of the Crab Nebula. The best fitted values are  $A = (3.27 \pm 0.13) \cdot 10^{-11} \text{TeV cm}^{-2} \text{s}^{-1}$ ,  $\alpha = 2.48 \pm 0.03$  and  $\beta = 0.20 \pm 0.02$ . Errors on the parameters are estimated from the diagonal of the covariance matrix resulting from the least squares fit. The error band in the plot has been estimated by sampling models from a gaussian with the same covariance matrix. The band indicates the 5<sup>th</sup> and 95<sup>th</sup> percentile of all 10 000 sampled models that are drawn from the gaussian. For this source the coverage of flux measurements in the transition region between synchrotron and inverse compton emission is provided by the Fermi satellite. For many other sources, mostly active galactic nuclei, the transition region is shifted to lower energies.

In literature the model often takes the form of an exponential function with three parameters  $A$ ,  $\alpha$  and  $\beta$

$$N(E) = A \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log_{10} \left( \frac{E}{E_0} \right)}. \quad (2.10)$$

As shown in [110, §5] and [98], the inverse Compton emission will approximately follow the shape of the electron spectrum at high energies. Hence, a log-parabolic electron distribution will lead to a log-parabolic gamma-ray emission. Figure 2.3 shows a fit of equation (2.10) to observed fluxes from the Crab Nebula.



# 3 Observation of Very-High-Energy Gamma Rays

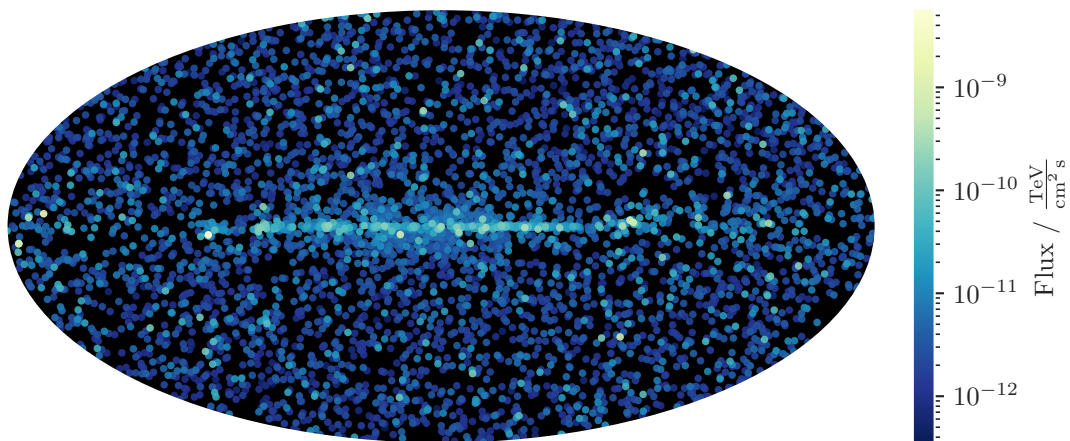
Cosmic rays have been observed using space-born as well as ground-based scintillation detectors for several decades. The highest-energy cosmic rays have been recorded by the AUGER instrument reaching up to  $10^{20}$  eV or 16 J of kinetic energy. While their isotropic arrival directions indicate their extragalactic origin, no point-source of cosmic rays could be identified as of today [1]. Measuring gamma-rays or neutrinos presents one useful advantage over observations of electrically charged cosmic rays. Photons and neutrinos are oblivious to electric and magnetic fields in space and travel in a straight line from their source to the observer. This makes them prime candidates for learning more about the processes and objects in which they are created. Neutrinos, however, are notoriously hard to capture due to their low mass and neutral charge. They only interact through the weak force requiring large detection volumes. So far no extragalactic point source of neutrinos could be identified though strong hints indicate they are produced in active galactic nuclei [2]. The observation of high-energy gamma rays requires much lower detector volumes and allows for observations into energy ranges of many TeV. Since the first successful gamma-ray astronomy missions in the early 1960s, the improvement of space-faring technology and sensor equipment now allows us to identify thousands of distinct sources of high-energy gamma radiation.

## 3.1 Satellite Experiments

The first gamma-ray telescope was launched into space onboard the Explorer XI mission in 1961 [95]. It carried a sandwiched scintillator with an area of only  $45 \text{ cm}^2$  and detected about 1000 gamma-rays during its 7-months-long mission. While not being able to pinpoint any sources, it is considered to be the first measurement of gamma rays of cosmic origin. Gamma-ray bursts (GRBs) were serendipitously discovered by satellites of the Vela mission in 1967, which were originally built to monitor the atmosphere for nuclear blasts [93]. More gamma-ray missions followed. Among them the Compton Gamma-Ray Observatory (CGRO), Swift,

### 3 Observation of Very-High-Energy Gamma Rays

AGILE and the Integral mission. The most recent mission, the Fermi Gamma-ray Space Telescope, was launched in 2008 and put into orbit about 550 km above Earth's surface. It is equipped with two detectors, the Gamma-ray Burst Monitor (GBM) and the Large Area Telescope (LAT). The GBM's primary purpose is to detect gamma-ray bursts within a large field of view. It can detect photons with energies between 8 keV and 40 MeV. As the name suggests, it was built to detect short bursts of gamma radiation in the sky. The bursts can last anywhere from minutes to seconds and are among the most powerful and bright events ever seen in the sky. About 200 to 300 GRBs are detected per year by the GBM [112]. The LAT detector onboard FERMI covers an energy range of 20 MeV to 300 GeV with a field of view of 2.4 sr covering about 20 % of the sky any one time. It was designed to catalog and monitor several thousand sources of gamma rays and record the energy spectra [25]. During its active years it has become an invaluable instrument for the cataloging and discovery of gamma-ray sources. The most recently published catalog [24] contains more than 5000 sources. Figure 3.1 shows a map of all sources found in the 4FGL catalog along with their gamma-ray flux between 100 MeV and 100 GeV.

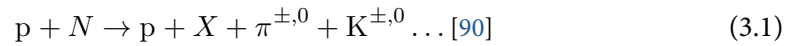


**Figure 3.1:** A sky map depicting each distinct source found in the 4FGL catalog combining data measured by the Fermi-LAT instrument over the course of 8 years. This is a Mollweide projection in galactic coordinates clearly showing a clustering of sources around the galactic plane. The colors indicate the total flux emitted by each source in an energy range of 0.1 GeV to 100 GeV.

### 3.2 Imaging Atmospheric Cherenkov Telescopes

Imaging Atmospheric Cherenkov Telescopes (IACTs) use the atmosphere as their detection medium. Each cosmic ray or gamma ray hitting Earth's atmosphere interacts with the nuclei in the surrounding gas. At sufficient energies, this interaction will create secondary particles,

which in turn interact with the surrounding medium. This process kicks off a cascade of particles moving towards the surface. This so-called *air shower* keeps growing until the particles' energies are insufficient to produce new offspring. Very energetic primary particles can induce cascades that reach the Earth's surface. The interactions governing the air shower induced by a primary hadron are severely different from those induced by a primary gamma ray or electron. An incoming electron or gamma ray interacts mainly through bremsstrahlung and pair production. This type of air shower is of purely electromagnetic nature. An incoming electron radiates a gamma-ray through bremsstrahlung in the presence of a nucleus. The new photon emerging from this *collision* either reaches the ground and gets absorbed or, given enough energy, creates a new electron-positron pair. This process continues until no new particles can be formed below an energy of roughly 1 MeV. A cosmic hadron, i.e. a proton or heavier nucleus, also starts a process of successive collisions. The number of sub-particles created in each collision depends on the parent particles' energy [71, page 78]. Equation 3.1 represents a proton-nucleon interaction, where  $N$  is the target nucleus in the atmosphere and  $X$  represents some remaining fragments of  $N$ .



The process will propagate through the atmosphere until no more sub-particles can be created below the rest energy of the pion near 140 MeV. Each hadronic air shower also has an electromagnetic sub-shower due to the gamma rays produced in pion decay



The air showers produced from hadronic primaries can be distinguished from the purely electromagnetic counterparts through various observables. Most important is the *lateral* spread of the shower and the time development of the number of particles present in the shower, both of which can be observed by ground-based telescopes. The charged component of an air shower produces flashes of Cherenkov light and makes it possible to *take an image* of the shower as its moving through the atmosphere. Early on, scientists working with radioactive material noticed a faint blue glow in water near strong radioactive sources [71, page 835]. Pavel Cherenkov began studying it systematically in 1934. He later shared the Nobel price for its discovery with Frank and Tamm in 1958 [137]. Cherenkov light is emitted by charged particles moving through a medium at superluminal speeds. While the speed of light in vacuum  $c_0$  is constant in all reference frames, the speed of light in transparent media is slower. The speed of light in a material is expressed in terms of its refraction index  $n = c_0/v$ . The charged high-energy particles present in an air shower can have velocities higher than the speed of

### 3 Observation of Very-High-Energy Gamma Rays

light in air. If they do, they radiate Cherenkov light. This bluish light is emitted along the direction of the moving particle. The opening angle of the Cherenkov light cone generated by a single charged particle of velocity  $v$  is given as

$$\theta = \arccos\left(\frac{1}{\beta n(h)}\right) \quad (3.3)$$

where  $\beta = v/c_0$  and  $n(h)$  is the refractive index of air at height  $h$  above sea level. As the refraction index  $n(h)$  increases with pressure, the opening angle of Cherenkov light decreases at high altitudes. At an altitude of 10 km, the opening angle will be close to  $0.8^\circ$ . At sea level the angle is closer to  $1.35^\circ$ . Hence, the Cherenkov light is strongly focused along the trajectory of the moving charge. A typical light flash produced by an air shower glows for approximately 20 ns to 30 ns. Capturing this faint and fast glimpse of an air shower requires sensitive instruments. Cherenkov telescopes require purpose-built cameras with single-photon resolution and fast readout systems to image air showers. The first detections of a point-source of cosmic gamma radiation was performed with the Whipple telescope in Arizona in 1989 [143]. It was the first detection of the Crab Nebula in the TeV range of light. The next generation of IACT experiments followed promptly with the HEGRA [15], MAGIC [18], H.E.S.S. [83], VERITAS [85] and FACT [20] projects, of which all but HEGRA are still operating. Before the first successful observation of the Crab Nebula, many challenges and problems inherent to IACTs had to be overcome. The first IACTs struggled to differentiate between showers induced by cosmic rays from those induced by gamma rays. Even for very bright sources of gamma rays, the amount of cosmic rays triggering the telescope is many orders of magnitude higher than the desired signal. Cosmic rays effectively act as the major source of background noise in IACT data. Development of methods to perform effective background suppression took extensive work on simulations of air showers. These simulations are necessary since there is no artificial source of gamma rays or protons in the energy ranges probed by IACTs. Experiments sensitive to lower energies can be calibrated in a laboratory setting, where their response to incoming particles can be measured in great detail. For IACTs simulations are the only way to gauge the instrument's response. Today, background suppression, or gamma-hadron separation, is performed using machine-learning algorithms which have been trained on simulations. Chapter 9 goes into more detail about the algorithms used. By today, the success of Cherenkov astronomy is self-evident given its huge contribution to our understanding of active galactic nuclei, supernova remnants and the gamma-ray sky in general. One recent result from the H.E.S.S. collaboration [73] is displayed in figure 3.2. It shows a gamma-ray view of the galactic plane in energies above 1 TeV. Chapter 7 contains details on the next-generation Cherenkov Telescope Array (CTA) project and its data analysis.

The following sub-sections describe the four currently operating Cherenkov telescopes and the astronomer's apparent fondness for intricate abbreviations.

#### 3.2.1 MAGIC

The Major Atmospheric Gamma Imaging Cherenkov Telescopes [18], or MAGIC, are a pair of Cherenkov telescopes located on the Canary island of La Palma off the west coast of Africa. It is part of the Roque des los Muchachos observatory on the island's volcanic remnant at a height of approximately 2200 m above sea level. The two telescopes feature large segmented mirrors with a total diameter of 17 m per telescope with a field of view of  $3.5^\circ$ . The first of the two telescopes was operated in monoscopic mode from 2004 to 2009 until the second telescope was ready for operations. MAGIC can detect relatively faint sources due to its large mirrors.

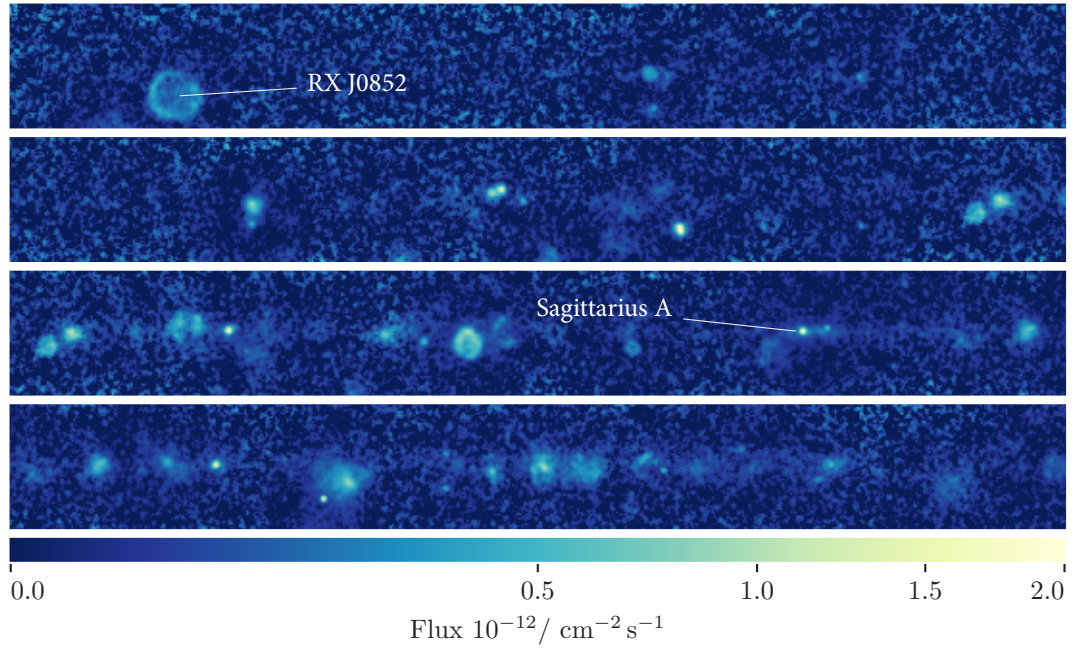
#### 3.2.2 VERITAS

VERITAS, the Very Energetic Radiation Imaging Telescope Array System [85], is an array of four telescopes. Each telescope has a mirror with a diameter of 12 m and a field of view of  $3.5^\circ$ . The VERITAS telescopes are located at the Fred Lawrence Whipple Observatory, Arizona, in just about 3 hours driving distance from the state's capital Phoenix. The four telescopes are located on the corners of a rectangle approximately 100 m apart from each other. VERITAS observes both active galactic nuclei as well as sources in our own galaxy.

#### 3.2.3 FACT

The First G-APD Cherenkov Telescope (FACT) was the first of its kind to employ silicon photo-multipliers for VHE gamma-ray astronomy. It is located next to the MAGIC telescopes and shares much of its infrastructure with it. FACT is a single IACT with a small mirror with a diameter of 4 m and a field of view of  $4.5^\circ$ . FACT is dedicated to observe and monitor bright active galactic nuclei on the northern sky. FACT is fully autonomous and does not require operators on site. It observes a list of predefined sources at night and automatically contacts a remote shifter via a phone call in case any problems appear during observation [124].





**Figure 3.2:** The H.E.S.S. galactic plane survey. H.E.S.S. measured the diffuse gamma-ray flux in the galactic plane over several years and published the data in 2018 [73]. The colors show the integral flux emitted in energies above 1 TeV [73, §4.3.2]. The figure shows a region near the galactic plane in longitudes  $260^\circ$  to  $45^\circ$  and latitudes  $-2.5^\circ$  to  $2.3^\circ$ . On the left-hand side of the upper panel, a supernova remnant RX J0852.0-4622 is clearly recognizable due to its shell-like structure. Sagittarius A in the center of our galaxy is marked in the third panel.

### 3.2.4 H.E.S.S.

The High Energy Stereoscopic System [83], H.E.S.S., is the largest of all currently operating Cherenkov telescopes and the only one operating south of the equator. It consists of 5 telescopes situated at an altitude of 1800 m in the Namibian highlands. Its location in the southern hemisphere allows for long-term observations of the center of the Milky Way as seen in figure 3.2. The largest of H.E.S.S.'s telescopes has a diameter of 28 m covering an area of about  $615 \text{ m}^2$ . The sizable aperture reduces the depth of field considerably. To adjust the focus of the optical system, the camera can be moved along the optical axis of the telescope electronically. The four smaller telescopes each have a diameter of 13 m. These four telescopes are arranged in a square with an edge length of 120 m with the large telescope placed in its center. The first phase of H.E.S.S. operations began in 2004. The fifth telescope was added in 2012, reducing the energy threshold of H.E.S.S. to several tens of GeV. A unique design feature of H.E.S.S. is its ability to quickly dismount the cameras from the telescopes. This simplifies maintenance works and hardware upgrades.



#### 3.2.5 CTA

The Cherenkov Telescope Array (CTA) will be the largest of all earth-bound gamma-ray observatories. It is currently in its planning stage with the first prototype telescopes under construction. Its current design includes over 100 telescopes that will be stationed at La Palma and the Paranal observatory in Chile. More details about CTA will follow in chapter 7. Suffice it to say at this point, it will be the largest array of optical telescopes that has ever been operated. With an estimated cost of over € 300 million [48], it will also be the most expensive operation in the history of ground-based astroparticle physics.

### 3.3 Water-Cherenkov Experiments

The particles produced in air showers can be captured using scintillation water tanks on the surface. A dense spacing of tanks and light sensors helps to measure details about the shower structure, which can help to differentiate between air showers induced by hadrons and air showers induced by photons. The High Altitude Water Cherenkov Experiment (HAWC) [8] is the latest gamma-ray observatory using water tanks. It is located in Mexico at an altitude of 4100 m and has been operating since 2015. HAWC consists of 300 water tanks with a total water content of 56 million liters. Each tank is fitted with 4 photomultiplier tubes. The charged component of air showers produces Cherenkov light in the tanks. The arrival time of the shower front in the tanks is the prime indicator for the direction of the incident shower. This method of water Cherenkov detection makes it easy to fill relatively large detector volumes hence making HAWC more sensitive to high gamma-ray energies beyond the capabilities of the FERMI satellite. HAWC's detection capability peaks near 10 TeV of gamma-ray energy. The absorption of high-energy photons due to interaction with extragalactic background light reduces the maximum distance at which HAWC can detect sources. Thus, HAWC is best suited for observing bright sources within our own galaxy. One considerable advantage of HAWC is its large field of view and its capability to operate during daytime.



# 4 Modeling the Crab Nebula Emission

Modeling the broadband photon emission of gamma-ray sources is an important tool to study the driving forces behind cosmic-ray acceleration. Accurate modeling can give valuable clues about the magnetic fields present in these sources as well as matter distribution and composition. Having access to multi-wavelength data from many instruments is pivotal to SED modeling. Unfortunately collecting this data is often tedious if not outright impossible. These data points are often proprietary and not available online. Even data published in journal papers is often solely provided as a plot without accompanying tables or data files. This problem will become an even larger problem in the future of multi-messenger astronomy. In an effort to change the status quo, flux data of the Crab Nebula has been collected by [15, 41, 114] and published within the open-source `gammapy` project. I used this data to evaluate a simple model of the acceleration processes in the Crab Nebula. Previous analyses [23, 114] indicate that two distinct electron populations are present in the source. In this case I am only interested in the high-energy part of the photon spectrum above  $10^7$  eV. Here, a single electron population suffices to describe the data. I use a Synchrotron Self-Compton model to calculate the gamma-ray flux given the shape of the electron distribution and the magnetic field strength. I assume that the magnetic field is of constant strength  $B$  and isotropic throughout the entire acceleration region. The electron population is modeled to be distributed according to a log-parabolic energy spectrum

$$N(E) = H(E, E_{\min}, E_{\max}) A \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log_{10} \left( \frac{E}{E_0} \right)}, \quad (4.1)$$

where  $H$  is a step function describing a cutoff beyond maximum and minimum electron energies of  $E_{\min}$  and  $E_{\max}$ .

$$H(E) = \begin{cases} E : & E < E_{\min} \text{ and } E \geq E_{\max} \\ 0 : & \text{otherwise} \end{cases}$$

The electrons produce synchrotron emission in the nebula's magnetic field as described in 2.2 and inverse Compton emission on existing photon fields as described in 2.3. Following [23],

four photon fields are assumed to seed the IC process. First, there is the photon field of the cosmic microwave background (CMB), which is modeled as blackbody radiation with a temperature of 2.7 K. Observations in the optical and infrared regime show glowing filaments of gas and dust inside the nebula. This second photon field is also assumed to be an isotropic blackbody radiator. The dust has an approximate temperature of 70 K and an energy density of  $0.5 \text{ eV cm}^{-3}$ . The third field is due to galactic background starlight with a temperature of 5000 K and a density of  $1 \text{ eV cm}^{-3}$ . The most important seed for IC emission, however, is the high-energy photons produced by the synchrotron emission within the nebula itself. All photon fields are assumed to have uniform number density within the nebula. While certainly accurate for the CMB photons, more accurate modeling of the photons' spatial distributions might improve results. This SSC model has six free parameters, of which five describe the shape of the electron distribution  $A$ ,  $E_{\min}$ ,  $E_{\max}$ ,  $\alpha$ ,  $\beta$ , and one describes the magnetic field strength  $B$ . The flux points to which this model is fitted was recorded by 6 different telescopes. The hard X-Ray and soft gamma-ray fluxes up to  $\approx 10^{-5}$  TeV were observed by the INTEGRAL satellite and the SPI instrument onboard the Comptel satellite. Fluxes from  $10^{-4}$  TeV to  $10^{-1}$  TeV were recorded by the Fermi satellite. Above  $\approx 10^{-1}$  TeV ground-based IACTs measure the highest gamma-ray energies.

The synchrotron and IC emission was calculated using the numerical approximations implemented in the `naïma` [148] package. Fitting was performed using Markov-Chain Monte-Carlo sampling (MCMC) on the posterior using the `emcee` [63] sampler. By default, `naïma` uses a Gaussian likelihood assuming independent measurement errors on the data

$$\mathcal{L}(\mathbf{F} | \mathbf{p}) = \prod_i^N \mathcal{N}(\mathbf{F}_i | \mu = \text{SSC}(\mathbf{p}), \sigma = \sigma_i), \quad (4.2)$$

where  $\mathbf{F}$  is a vector of flux measurements with corresponding uncertainties  $\sigma$  and  $\mathbf{p}$  is the parameter vector for the SSC model. The independence assumption for the flux errors is almost certainly not correct. It is, however, a pragmatic approach when working with flux data which is often published without further information about possible correlations between the points. All priors were assumed to be uniform, or uniform in logarithmic space.

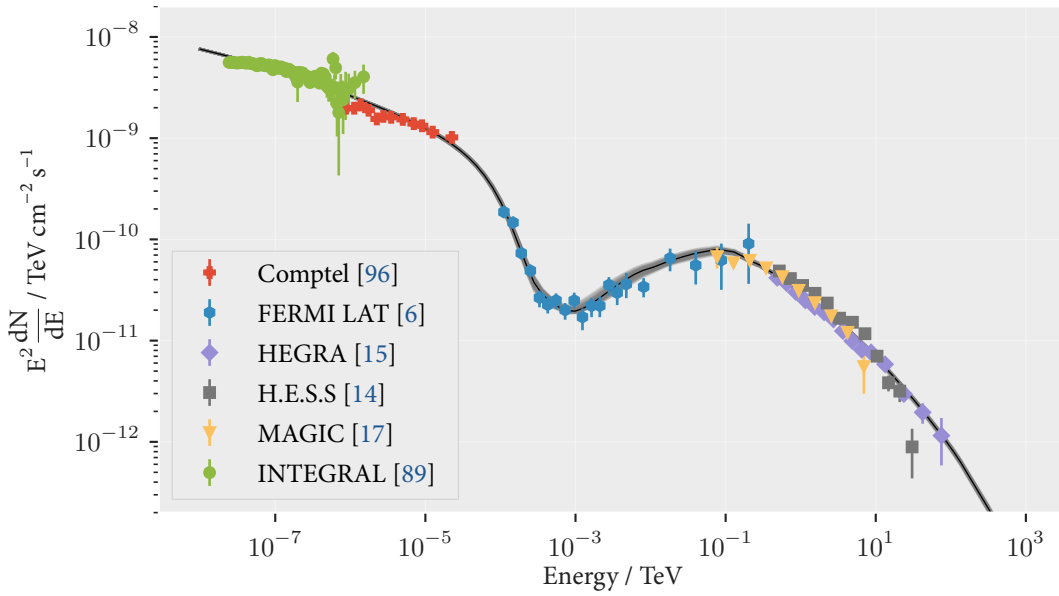
Even though implementation of radiative models in `naïma` is relatively efficient, evaluating the model on hundreds of thousands of Markov chain samples takes hours. To speed up the sampling, I have built a lookup table of values of the SSC model evaluated on a grid of 1 000 000 parameter combinations. Building the lookup table takes several hours on a large machine with 24 CPU cores. The samplers then evaluated the model using linear inter-

polation between the grid points. Sampler settings can then be adapted and tuned without recalculating the lookup table. For the final fit a total of 1 500 000 samples were taken in 300 parallel chains. The resulting fit values are calculated from the median of the marginalized posterior distributions. The provided errors are taken from the 16<sup>th</sup> and 84<sup>th</sup> percentile.

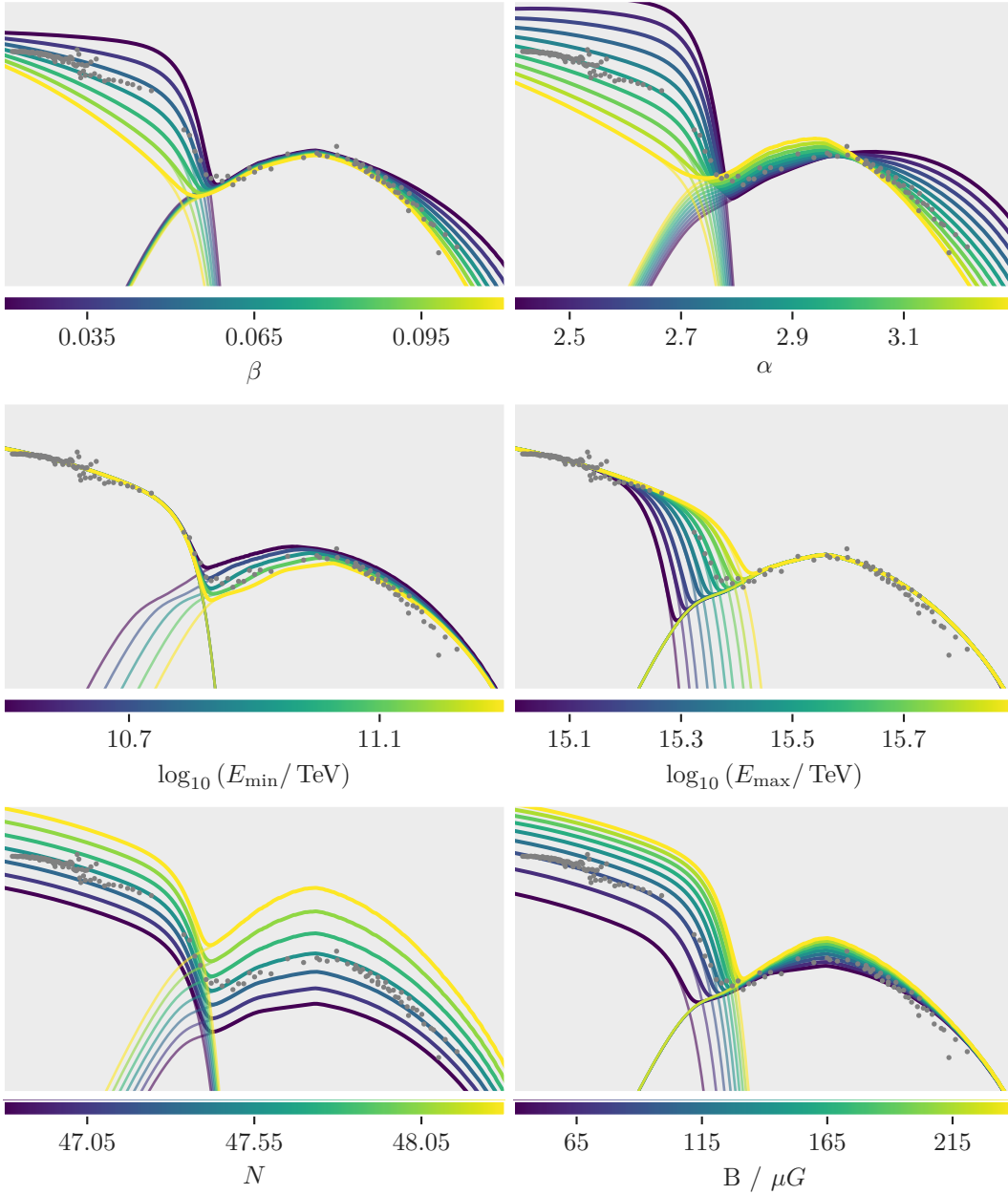
$$\begin{aligned}
 \log_{10}\left(\frac{N}{\text{erg}}\right) &= 47.51^{+0.02}_{-0.02} & \alpha &= 3.03^{+0.01}_{-0.01} \\
 \log_{10}\left(\frac{E_{\text{max}}}{\text{TeV}}\right) &= 15.55^{+0.01}_{-0.01} & \beta &= 0.045^{+0.001}_{-0.001} \\
 \log_{10}\left(\frac{E_{\text{min}}}{\text{TeV}}\right) &= 11.11^{+0.05}_{-0.05} & \frac{B}{\mu\text{G}} &= 107.8^{+1.6}_{-1.5}
 \end{aligned}$$

Figure 4.1 shows the fitted model together with the measured flux data. Even though this model uses relatively simple physical assumptions, it accurately represents the observed fluxes in the high-energy end of the SED. Figure 4.2 shows the influence of each single parameter on the shape of the model spectrum. Table A.1 in the appendix shows the values of the fitted parameters together with images of the marginalized posterior distributions and the sampled chains. Figure A.1 in the appendix shows the correlation between each of the fitted parameters. More details on MCMC methods in general, will be given in section 5.5. The code to reproduce these results is available at

[https://github.com/kbruegge/simple\\_ssc\\_model](https://github.com/kbruegge/simple_ssc_model) 



**Figure 4.1:** The full SSC (Synchrotron Self-Compton) model plotted together with the observed data. The colored error bars show the observed fluxes by the six different instruments. The black line shows the median of the values sampled by the Markov chain. The error band around the black line is built from randomly chosen samples in the chain. For each sampled parameter set, the SSC model is drawn as a gray transparent line. Burn-in samples have been discarded before producing this plot. Despite the simple model assumptions, the SED accurately matches the data in the VHE gamma-ray end.



**Figure 4.2:** The full SSC model plotted together with measured fluxes. In each image one of the parameters of the model is varied while the others remain fixed. As stated in the text, this model assumes one population of high-energy electrons which are distributed according to a log-parabolic energy spectrum with high and low energy cutoffs. Each of the six free parameters is varied as indicated by the colorbars.





# 5 Spectral Analysis of IACT Data

In the previous chapter I fitted a model to existing flux points published by various IACT experiments. In this chapter I will describe how to estimate the flux of a gamma-ray point source using IACT data.

Observational astrophysics is performed by measuring the energy output of an object in some form or another. IACT analysis is no different. The success of multi-wavelength observations of the sky are self-evident. The combination of images from multiple wavelengths is key to unlocking some of the mysteries of modern astrophysics. In the 1970s, in an effort to overcome the technical challenges involved when sharing images among operating systems, the FITS file format was invented. The Flexible Image Transport system (FITS), first standardized in 1981 [144], quickly became the image format of choice among astronomers. Since the 1990s, with support from NASA's High Energy Astrophysics Science Archive Research Center (HEASARC), many additions to the FITS format were developed to stimulate the interchange between experiments. The FITS format has become the de-facto standard for storing high-level results in astronomy in all wavelengths, be it radio, infra-red, optical or X-Ray. The VHE gamma-ray community, the community of ground-based Cherenkov observatories, is a considerable exception. Their methodology was inherited from particle physics. For decades the standard software and file format in traditional particle physics has been the ROOT project [40]. Storage and data analysis software for H.E.S.S., MAGIC, and VERITAS is programmed within the confines of the ROOT framework. Despite the common framework, the internal file structures and softwares are proprietary and no sharing of code or data takes place among the experiments. The upcoming next-generation Cherenkov Telescope Array (CTA) will break the status quo. The CTA project is currently under construction and will consist of at least 80 telescopes of 4 different types. The new generation of astroparticle physicists in the CTA collaboration are trying to move away from the closed-source ROOT-based approach to modern open-source scientific solutions like the `gammapy` [51] project. Even though the internal file structures and programs are different, all Cherenkov telescopes produce their data in similar ways in terms of physics. So similar in fact, that a joint effort was started to find a common file format for storing and representing them in FITS format. The project was initiated by Christoph Deil in 2016 in a face-to-face meeting near Paris. Members from all

IACT collaborations, including myself, discussed commonalities and differences between the data produced by the different experiments. A description of a common file structure was drafted and presented at the Gamma 2016 conference [53] in Heidelberg. High-level datasets from the MAGIC, H.E.S.S., VERITAS and FACT telescopes were made available to the public in FITS format. These datasets contain observations of the Crab Nebula and corresponding instrument response functions (IRFs). Under the auspices of Cosimo Nigro and Christoph Deil, we used this data to publish a joint paper titled “Towards open and reproducible multi-instrument analysis in gamma-ray astronomy” [122] in the *Astronomy & Astrophysics* journal. In the paper we perform a multi-instrument likelihood fit for a log-parabolic spectral model on the Crab Nebula data from all participating telescopes. The joint analysis in our paper uses open source software with public data and allows anyone to recreate the results. Users can execute the analysis by downloading all data and the relevant source code under a fixed digital object identifier (DOI) <https://doi.org/10.5281/zenodo.2381863>. It is the first paper in the field of gamma-ray astronomy which provides this level of reproducibility. In this chapter I perform a similar spectral analysis on the data similar to the one we presented in the paper. The statistical model used in my analysis is modified to contain nuisance parameters describing the irreducible background. The statistical model is described in section 5.4. Section 5.3 outlines how the background is estimated from measurements. Instead of minimizing the likelihood as we did in the paper, I sample the posterior distribution using Markov-Chain methods. Section 5.5 gives a short overview of the Monte-Carlo methods used to sample distributions. There are two different ways to perform flux estimation of astrophysical sources. The model-dependent approach, which I will describe in section 5.6, depends on an analytical description of the sources energy dependent flux. The model-independent approach, which is explained in 5.7, is often called *unfolding*. While both approaches use the same underlying statistical model and are solved with the same techniques, they differ fundamentally in terms of semantics. Fitting spectral models is useful for validating or invalidating physical assumptions. Unfolding is useful for creating model-independent flux points which can be used in conjunction with measurements and models which use different assumptions. The main difficulty, for both approaches, lies in building the instrument response function (IRF). Motivations for building a common data format and details about the datasets are given in section 5.1. Details about the exact shape and functionality of the IRFs are given in section 5.2. The pipeline that processes the raw telescope data into a high-level event lists usable for astrophysical analysis will be the topic of chapter 8. For the remainder of this chapter, the datasets I refer to are the published observations used in our joint paper [122]. The loading of the event lists and instrument responses as well as the estimation of background counts is performed with code from the *gammapy* project if not mentioned otherwise.

## 5.1 Event Data

Unlike traditional optical telescopes, the data recorded by IACTs is not of *image-like* dimensions. While an optical telescope exposes some region in the sky for a long period of time to create an image, IACTs record a set of distinct *events*. In the community's parlance, one event refers to the recorded data of one unique air shower. For a single telescope this results in one image recorded per event, for an array of multiple telescopes more images get recorded. The data in its raw form is not usable for astrophysical analysis. The cameras used for Cherenkov telescopes are highly specialized devices sensitive to single photons. Precise calibrations have to be performed and applied to the data in order to distinguish Cherenkov photons from background light or sensor artifacts. As discussed before, even after preprocessing and reconstruction of the primary particles' energy and direction, most of the observed air showers are induced by cosmic rays. These are rejected by machine-learning methods before the final event list is produced. The algorithms are trained on simulated data to differentiate between gamma-ray and cosmic-ray air showers. Even then, much unwanted cosmic-ray background remains. Chapter 9 will discuss these methods in detail.

The common data structure we propose for IACT data is based on the FITS format. The BINTABLE extension to the FITS standard allows for storage of table-like data with associated meta data stored in a header. The common data format requires just 5 columns to perform spectral analysis: the **ENERGY** column, which contains estimated energy of the primary particle; the **RA** and **DEC** columns containing the reconstructed point of origin of the event on the sky in equatorial coordinates; the **TIME** column, which stores each event's time of recording, and a unique identifier called **EVENT\_ID**. Additional information, such as the pointing position of the telescope or the total observation time, is stored in the attached header [52]. It is important to keep in mind that *estimated energy* means different things for different instruments and analysis methods. In the hypothetical situation that both FACT and MAGIC see an air shower induced by a gamma ray with 1 TeV kinetic energy, the estimated energies of FACT and MAGIC can differ wildly. The energy migration matrices covered in the next section can only ever consider expected values of energy distributions. For a single air shower, no hard statements can be made. To make matters worse, for the reasons outlined in 5.2, the instrument responses can only be built from simulations. The response functions for observed data might look different. This uncertainty caused by using simulations can introduce biases in the result if the simulation does not resemble reality accurately. While it is certainly possible to model some of this systematic uncertainty, it is prohibitively expensive in terms of computing and man power. Table 5.1 gives an overview over the four datasets used

**Table 5.1:** The data used in the fit from MAGIC, H.E.S.S., VERITAS and FACT. All the data was made public and can be downloaded as a Zenodo bundle [122] including exploratory Python notebooks to read the data. The altitude range shows the minimum and maximum pointing angle with respect to the local horizon.

Telescope	Year	Duration / h	Number of Events	Altitude Range
H.E.S.S.	2004	1.87	30129	41.4° to 44.6°
FACT	2013	10.33	27306	60.1° to 83.6°
MAGIC	2013	0.66	12109	69.7° to 74.3°
VERITAS	2011	0.67	3482	80.1° to 80.7°

to fit the Crab Nebula energy spectrum. The number of events seen by the telescopes depends strongly on how the events were selected during the analysis. Even so, the small mirror size of FACT clearly means that more observation time is required to gather a sufficient amount of data.

## 5.2 Instrument Response Functions

The telescope's probability to trigger an incoming air shower, a so-called *event*, depends on environmental conditions like humidity, ambient temperature, background light, cloud coverage and pointing direction. Even for fixed environmental conditions, the probability still depends strongly on the primary particle's true energy and point of origin within the field of view. Not all gamma rays can be properly reconstructed by the telescope's analysis. This might be simply due to the fact that they are too dim or undistinguishable from sensor noise. Some events get miss-classified, i.e. an air shower induced by a gamma ray is erroneously recognized as a hadron-induced shower. Events may also fail to pass some predefined quality requirement. After accounting for the triggering probability and all analysis-related effects we are left with total acceptance probability  $P_a(E_T, \Theta, \Psi, \Phi)$ . The angles  $\Psi$  and  $\Phi$  define the pointing direction of the telescope in horizontal coordinates. The angle  $\Psi$  is the azimuth angle with respect to the north pole. At different azimuth angles the observed air showers are aligned differently with respect to the local geomagnetic field above the telescope site.  $\Phi$  is the pointing altitude above the local horizon. Air showers detected at low altitude angles have to travel a much greater distance through the atmosphere than air showers coming from the zenith directly above the telescope's location. Hence, both angles influence the telescope's acceptance probability. The angle  $\Theta$  defines the offset angle between the center of the telescope's field of view and a potential gamma-ray source. The acceptance probability drops off rapidly towards the edge of the field of view (FoV). Aberration effects in the telescope's

optical system increase at larger distances from the center of the FoV. More importantly, air showers originating near the edge of the FoV are less likely to be detected. Near the edge, only one-half of all air showers will remain inside of the telescope's visible slice of atmosphere. Measuring the  $P_a$  directly is impossible since no artificial source of gamma rays exists in the required energy range. Simulations are the only way to gauge the instrument response to air showers. For each generated gamma ray, the simulation needs to propagate all constituents of the resulting air shower through the atmosphere. For each charged particle in the air shower, the Cherenkov photons must be generated and their trajectory through the atmosphere calculated. Simulated Cherenkov photons reaching the telescope are mapped onto the camera's sensor plane via ray tracing. All components that potentially interact with the photons must be taken into account. The reflection on the mirrors, the shadowing by the telescope's frame and refraction of the light guides in front of the pixels are all important components that have to be simulated as close to reality as possible. The resulting photon distribution on the camera's sensor is then used as input for the simulation of the detector's electronics. The trigger logic, sensor effects, and electronic artifacts are simulated up to the point where the telescope's data acquisition system writes all its data to a file. The CORSIKA [79] software is responsible for the air-shower physics part of the simulation. It has been under active development for more than 20 years and is used for all major astroparticle experiments. The collaborations building Cherenkov telescopes or cosmic-ray detectors then use their own in-house detector simulation software to process the output of CORSIKA. Each simulated air shower which triggers the telescope will be put through the telescope's analysis pipeline just like observed data. During the analysis, noise removal, background suppression and event selection is performed. This includes the filtering of air showers which are likely of hadronic origin. The number of events remaining after the analysis is smaller than the number of triggered events and much smaller than the amount of simulated showers

$$N_{\text{Simulated}} > N_{\text{Triggered}} > N_{\text{Accepted}}. \quad (5.1)$$

The number of accepted events in the result is then used to estimate the acceptance probability  $P_a = \frac{N_{\text{Accepted}}}{N_{\text{Simulated}}}$ . A Cherenkov telescope can only observe air showers whose primary particles' trajectory hits the ground around the telescope within a maximum distance  $R_{\text{max}}$ . As seen in equation (3.3) the Cherenkov emission of an air shower is directed strongly along the shower's trajectory. The photons hit the ground within an energy dependent radius of the shower's impact point on the ground. In order to detect an air shower, the telescope has to be situated within that radius. This defines the maximum area of impact points  $A = 2\pi R_{\text{max}}^2$  around the telescope in which air showers are still observable. The acceptance probability is

multiplied by the area in which showers are scattered during simulation  $A_{\text{sim}}$  resulting in the *effective area*  $A_{\text{eff}} = P_a A_{\text{sim}}$ . The true, time-dependent, gamma-ray flux of a source, either simulated or real,  $N(E_T, t)$  is transformed into a distribution of accepted events by

$$f(E_T, \Theta) = \int_{t_{\text{obs}}} N(E_T, t) P_a(E_T, \Theta, \Psi(t), \Phi(t)) A_{\text{sim}} dt. \quad (5.2)$$

The pointing angles are time dependent when the telescope tracks a source. During the observation time  $t_{\text{obs}}$  the Earth's rotation forces the pointing position to adjust. The offset parameter  $\Theta$  remains free so that any potentially extended source in the FoV can be observed. In practice, the explicit time dependence is removed by splitting the measurement into smaller distinct observation windows, so-called runs. The lengths of one run is chosen such that the pointing angles  $\Psi, \Phi$  do not change significantly during the run. The effective area is calculated on a per-run basis. For spectral analysis several observations of the same source are aggregated which implicitly averages the luminosity of the gamma-ray source over time

$$f(E_T, \Theta) = \sum_{\text{run}} N(E_T) A_{\text{eff,run}}(E_T, \Theta), \quad (5.3)$$

where  $N(E_T) = \int_{t_{\text{obs}}} N(E_T, t)$  is the sources flux integrated over time. Typical run durations lie anywhere between 5 and 20 minutes. From a technical standpoint this brings an additional advantage. It reduces load on the data acquisition and computing infrastructure as it avoids writing huge files for long observation campaigns.

The gamma ray's true kinetic energy  $E_T$  can only be estimated by the amount of Cherenkov light emitted by the air shower. Much care is taken to use all available information in the recorded Cherenkov images to estimate the primary particle's energy. As described in chapter 9, machine-learning methods are employed to get the best estimator possible. Nevertheless, no energy estimation is perfect. All models are inherently flawed. Due to the stochastic nature of the air shower, the correspondence between true and estimated energy can never be completely accurate. The relation between the true energy  $E_T$  and the estimated energy  $E_{\text{Est}}$  is given by the dispersion function  $p_j(E_{\text{Est}}, E_T, \Theta, \Psi, \Phi)$ . As with the effective area, the explicit dependency on time and the pointing angles can be dropped when splitting the observation into shorter runs. The dispersion function transforms distributions in true energy  $f(E_T)$  into distributions of estimated energy  $g(E_{\text{Est}})$ . The function  $f$  has compact support since  $P_a$  has a minimum and maximum energy outside of which no events can be

accepted. This means that  $f$  can be normalized and interpreted in a stochastic context. In this sense  $f$  is the probability density function of the random variable  $E_T$

$$\Pr(E_T \in \Delta E_T) = \int_{\Delta E_T} f(E) dE.$$

In the same manner,  $g$  can be defined as the probability density function of the random variable  $E_{\text{Est}}$ . Each simulated air shower has a true and an estimated energy. Provided sufficient simulations are available, the distributions  $g$ ,  $f$  and the joint distribution  $p_j(E_T, E_{\text{Est}})$  can be estimated either by kernel density estimation or straightforward histogramming of the data. Marginalizing  $p_j$  with respect to  $E_T$  results in a function which connects the true event energy with its estimated energy

$$g(E_{\text{Est}}, \Theta) = \int_{E_T} p_j(E_{\text{Est}}, E; \Theta) dE = \int_0^\infty p_D(E_{\text{Est}} | E; \Theta) f(E) dE. \quad (5.4)$$

The dispersion function acts as the conditional distribution of the estimated energy given the true energy  $p_D(E_{\text{Est}} | E_T)$ . This relation also holds if  $f$  is not normalized to be a proper density function, but instead is a function relating true energy to actual event counts or rates. Now all necessary ingredients are available to define the measurement process of an IACT. By substituting  $f$  with the expression from equation (5.3), the distribution of event counts for a single run can now be written as

$$G(E_{\text{Est}}, \Theta) = B(E_{\text{Est}}, \Theta) + \int_{E_T} p_D(E_{\text{Est}} | E_T; \Theta) N(E_T) A_{\text{eff}}(E_T, \Theta) dE_T, \quad (5.5)$$

where  $B(E_{\text{Est}}, \Theta)$  is the remaining background after all analysis steps have been performed. This background consists of showers started by cosmic electrons or hadrons which could not be discriminated from the actual gamma-ray signal by the analysis. The background can be modeled in the same way as the signal by extensive simulations and calculation of acceptance probabilities and migration functions. However, the simulation of hadronic air showers is quite expensive in terms of computing time. As detailed in section 5.3 the background  $B(E_{\text{Est}}, \Theta, \Psi, \Phi)$  can be measured directly while observing gamma-ray sources.

In practice, the effective area and the migration function are calculated in some discretized form. The effective area for a single observation is calculated by creating a two-dimensional histogram of accepted events in bins of true energy  $E_T$  and offset angle  $\Theta$ . If  $S_a$  is the set of accepted events and  $S$  the set of simulated events, then we get the effective area matrix  $\mathbf{A}$  by

dividing the number of accepted events in bin  $\Delta E_T$  by the total number of simulated events in that bin

$$A_{\Delta E_T} = A_{\text{sim}} \frac{\sum_{E_T \in S_a} \mathbf{1}_{\Delta E_T}(E_T)}{\sum_{E_T \in S} \mathbf{1}_{\Delta E_T}(E_T)}, \quad (5.6)$$

where the indicator function  $\mathbf{1}_R$  is defined as

$$\mathbf{1}_R(x) := \begin{cases} 1 & \text{if } x \in R, \\ 0 & \text{else.} \end{cases} \quad (5.7)$$

This admittedly unwieldy notation describes the division of two histograms, which is a relatively simple operation in many programming languages. An astrophysical source model  $N(E)$  can be discretized into a vector containing counts in true energy  $c$  where each entry in the vector is calculated by integrating over the energy range in the bin

$$c_{\Delta E_T} = \int_{\Delta E_T} N(E) dE.$$

The count vector of accepted events can then be calculated in analogy to equation (5.2)

$$f_{\Delta E_T} = c_{\Delta E_T} A_{\Delta E_T} t_{\text{obs}}.$$

The subscripts can be dropped when expressing  $f$ ,  $c$ , and  $A$  as vectors containing the values for all energy bins. The equation can then be written in terms of vectorized operations as

$$f = c \odot A t_{\text{obs}},$$

where  $\odot$  is the element-wise multiplication of two vectors, the Hadamard product.

The energy dispersion function for one observation is discretized in a similar way. A two-dimensional histogram is build to estimate the joint probability density of true and estimated energy

$$\mathbf{p}_{j_{\Delta E_{\text{Est}} \Delta E_T}} = \sum_{(E_T, E_{\text{Est}}) \in S_a} \mathbf{1}_{\Delta E_T}(E_T) \mathbf{1}_{\Delta E_{\text{Est}}}(E_{\text{Est}}).$$

Per definition of conditional probabilities,  $p_D(E_{\text{Est}} | E_T)$  can be calculated from the joint distribution by dividing it by the probability density of  $E_T$

$$p_{\Delta E_T} = \sum_{E_T \in S_a} \mathbf{1}_{\Delta E_T}(E_T).$$



resulting in the final dispersion matrix

$$\mathbf{P}_{D\Delta E_{\text{Est}}\Delta E_T} = \frac{\mathbf{P}_{j\Delta E_T\Delta E_{\text{Est}}}}{p_{\Delta E_T}}.$$

In analogy to equation (5.4) the relation between estimated and true energy can then be expressed using matrix multiplication of  $\mathbf{p}_D$  with count vectors  $g$  and  $f$

$$g = \mathbf{p}_D f.$$

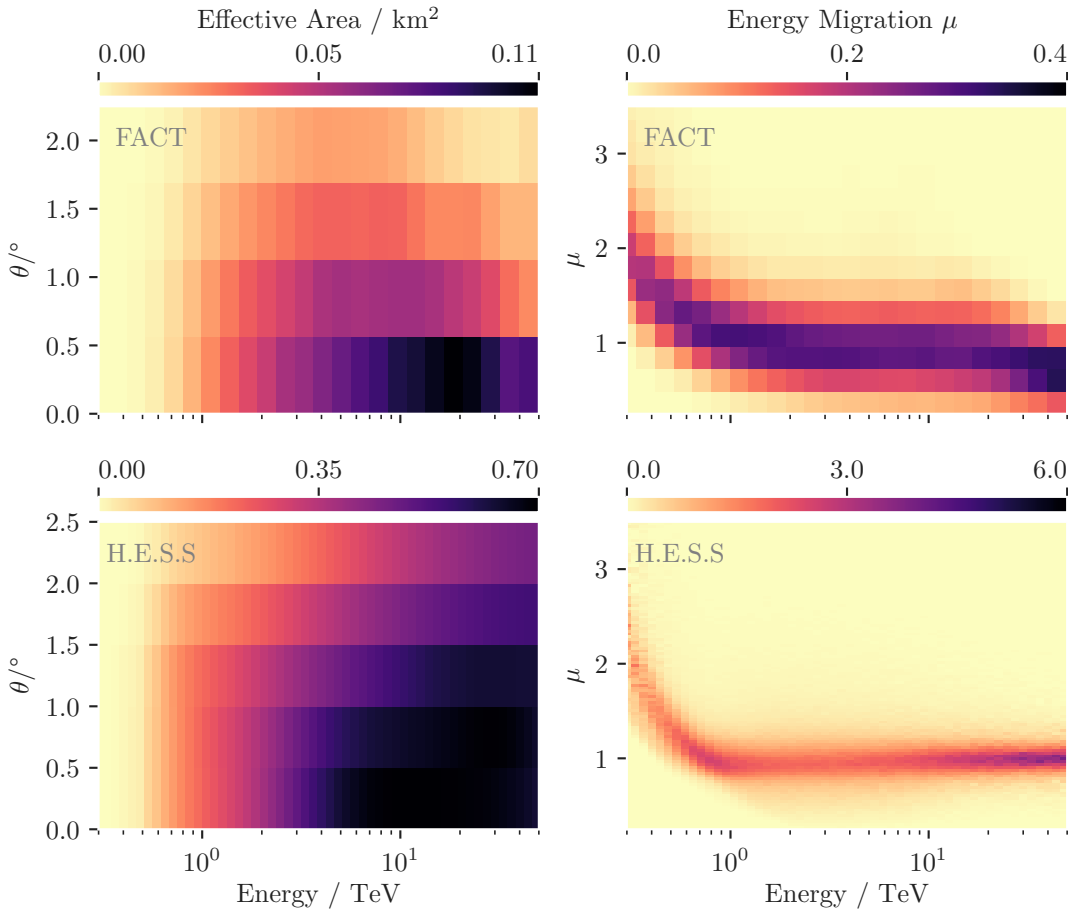
Now the measurement process can be written down in discretized form equivalent to equation (5.5)

$$g = B + \mathbf{p}_D (c \odot A) t_{\text{obs}}. \quad (5.8)$$

The instrument response functions,  $\mathbf{p}_D$  and  $A$  are stored for each offset bin  $\Theta$ . The data format introduced in the previous section 5 stores the energy dispersion information in a modified form. The energy migration  $\mu = \frac{E_{\text{Est}}}{E_T}$  is calculated for each simulated event and then distributed into bins of width  $\Delta\mu$

$$\mathbf{P}_{M\Delta\mu\Delta E_T} = \frac{\sum_{(\mu, E_T) \in S_a} \mathbf{1}_{\Delta\mu}(\mu) \mathbf{1}_{\Delta E_T}(E_T)}{p_{\Delta E_T}}. \quad (5.9)$$

Speaking in terms of histograms: The IRF contains a two-dimensional histogram in bins of  $\Delta\mu$  and  $\Delta E_T$ . The right-hand side of figure 5.1 shows FACT's energy migration matrix in the offset range  $\Delta\theta = [0^\circ, 1.125^\circ)$ . The left side of figure 5.1 shows a plot of the effective area of the FACT telescope. The IRFs are stored according to the open gamma-ray astronomy data description (OGA) which I mentioned in the previous section. The OGA defines column names, units, dimensions and header keywords necessary to store the IRFs into FITS files. The IRFs are stored as multidimensional arrays in binary tables according to the BINTABLE extension of the FITS standard [136, S.7.3] and appendix B of [47]. More details are given in the official OGA documentation [52].



**Figure 5.1:** The figures in the left column show the effective area of the FACT and H.E.S.S. telescopes. The effective area is split into offset bins between  $0^\circ$  and radius of the field of view. It is largest near the center of the field of view, where images of the air showers are still fully contained within the telescope’s camera. The effective area for FACT takes into account only those events which have been reconstructed to be originating within a radius of  $0.17^\circ$  from the true, simulated, source position. The effective area for H.E.S.S. applies no directional cut. The right-hand figures show energy migration. FACT’s migration is valid between  $0^\circ$  and  $1.125^\circ$  and H.E.S.S.’ between  $0^\circ$  and  $0.5^\circ$ . Each column in the migration matrix sums to unity as defined in (5.9). At the time of producing the FACT IRFs, not much simulated gamma-ray data was available and so the matrices for H.E.S.S. use a much finer binning in both energy and offset.

### 5.3 Background Estimation

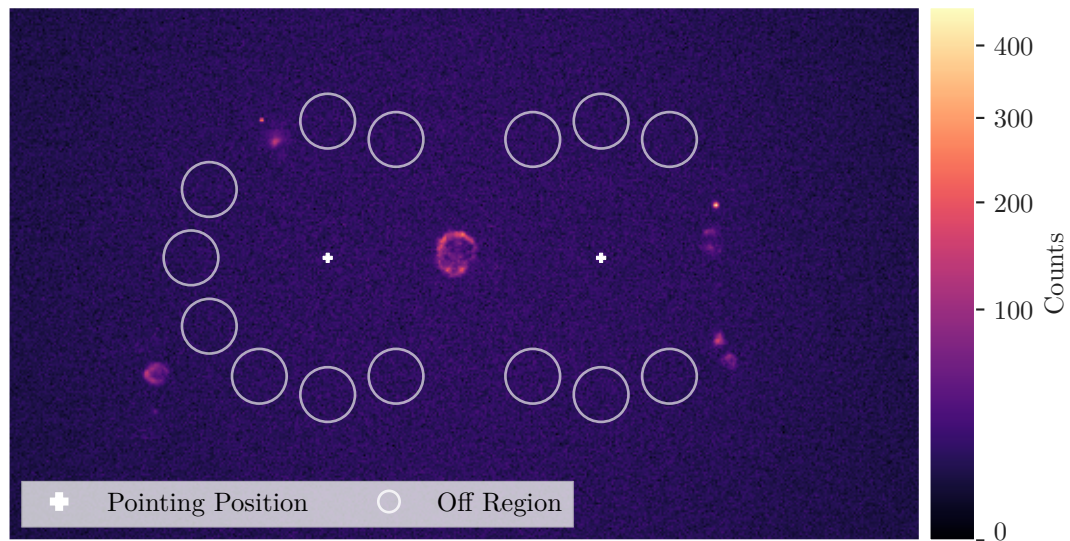
The first step performed when measuring the energy spectrum of a point source is to create a *count spectrum*. In most cases the positions of the point sources are known from measurements in other wavelengths, most often radio and optical. As described before, an IACT produces a list of reconstructed energies and directions. All events within a predefined radius of the point source's position are selected and distributed into bins of estimated energy  $E_{\text{est}}$  building the count spectrum. This count spectrum is called  $N_{\text{on}}$  as it counts the number of events in the signal region. Despite best efforts to reduce the background in the measurements, IACT data is always contaminated by background events, either due to miss-classified events that were induced by hadrons or due to air showers started by electrons which are notoriously hard to discriminate. Hence, the count spectrum inevitably contains unwanted background counts

$$N_{\text{on}} = N_{\text{signal}} + N_{\text{background}}.$$

One way to estimate the background is to perform *off measurements*. For these measurements, a region in the sky is selected with similar conditions to that containing the source. This might be a region with similar amounts of background starlight or similar atmospheric conditions. The large disadvantage of this method is the large amount of wasted observation time. For each signal observation, some time has to be allotted for off measurements. The WHIPPLE telescope operated in this manner until the HEGRA collaboration established the *wobble* observation mode. The idea is to point the Cherenkov telescope not directly at the source, but with a small offset. During observation of point-sources, large fractions of the telescope's field of view are free from gamma-ray sources. This fact can be used to estimate the background contamination of a count spectrum. Background events are sampled from this *off region* using the same criteria that were applied when selecting the data in the signal region. This will result in a measured count spectrum for the off-region

$$t_{\alpha} N_{\text{off}} \approx N_{\text{background}},$$

where  $t_{\alpha}$  is the fraction of area covered by the on and off-region and their respective exposure times. Choosing off-regions is not trivial. The signal, or *on*, region position is fixed due to the location of the source. The radius of the region depends on the angular resolution of the directional reconstruction. The off-region needs to be large enough to get enough statistics and at the same time have a similar instrument response as the on-region. As visible in figure 5.1, the instrument response function varies strongly within the field of view. The selection



**Figure 5.2:** A typical observation in wobble mode. In this hypothetical observation, the supernova remnant in the center of the image is the object of interest. Two pointing positions are marked by white pluses. The small white circles indicate the off-regions. They are positioned on the edge of a circle around their respective pointing position. This way the background can be measured in the same field-of-view offset  $\Theta$  as the gamma-ray source. Regions in the sky which contain other gamma-ray sources are excluded from the choice of off-regions.

process needs to take the inhomogeneous shape of the IRF into account. In the simple case of radially symmetric acceptance probability, which is roughly the case for many observations, one can select regions with a similar offset  $\theta$  as the on-region. This method is called *reflected regions* and is just one of many ways to estimate the background. A comparison of methods can be found in [30]. Figure 5.2 shows how the reflected regions are selected for wobble observations. Once the off-regions have been selected, the background contamination of the signal's count spectrum can be estimated and we are left with  $N_{\text{on}}$  and  $N_{\text{off}}$ . Now we can build a statistical model to estimate the energy spectrum of the observed source.

## 5.4 Statistical Modeling

The count spectra measured for both  $N_{\text{on}}$  and  $N_{\text{off}}$  can be modeled by Poisson distributions. Each bin in estimated energy, i.e. each entry in  $N_{\text{on}}$  or  $N_{\text{off}}$ , is the result of a counting process. Each value in  $N_{\text{off}}$  is the number of accepted events in the off-region within a fixed range of estimated energies  $\Delta E_{\text{est}}$ . Hence, each entry is distributed as

$$P_{\text{off},i} \sim \text{Poisson}(\mu_{b,i}).$$

The Poisson distribution models the number of independent events in a fixed space or time range. Its probability mass function is defined as

$$P_{\text{Poisson}}(k) = \frac{\mu^k e^{-\mu}}{k!}, \quad (5.10)$$

where  $k$  is the number of events and  $\mu$  is the mean of the Poisson distribution and its only parameter [50, page 287]. The counts in the on-region

$$N_{\text{on}} = N_{\text{signal}} + N_{\text{background}} = N_{\text{signal}} + t_{\alpha} N_{\text{off}}$$

are the result of two independent poisson processes. The mean of the counts in the on-region is the sum of the mean signal count  $\mu_s$  and the mean background count  $t_{\alpha}\mu_b$ . Consequently, the counts in the on-region are distributed as

$$P_{\text{on},i} \sim \text{Poisson}(\mu_{s,i} + t_{\alpha}\mu_{b,i}).$$

We are interested in finding the mean of the signal counts  $\mu_s$  from the observed data vectors  $N_{\text{on}}$  and  $N_{\text{off}}$ . The likelihood of the observed data given the parameter vectors  $\mu_s, \mu_b$  can be expressed as the joint distribution of the background and signal. Since these processes are independent of each other, the joint distribution is simply the product of the two likelihoods

$$\mathcal{L}(N_{\text{on}}, N_{\text{off}}, t_{\alpha} \mid \mu_s, \mu_b) = \prod_i P_{\text{on},i}(N_{\text{on},i} \mid \mu_{s,i} + t_{\alpha}\mu_{b,i}) \cdot P_{\text{off},i}(N_{\text{off},i} \mid \mu_{b,i}). \quad (5.11)$$

The number of free parameters in  $\mathcal{L}$  depends on the number of energy bins in the count spectra

$$n_{\text{params}} = \dim(\mu_s) + \dim(\mu_b) = \dim(N_{\text{on}}) + \dim(N_{\text{off}}).$$

We are only interested in finding the parameters which best describe the data under the given likelihood. Hence, the absolute value of the likelihood is of no importance. It is often easier to work in the equivalent formulation of the problem by applying the logarithm and dropping all terms which are constant with respect to the parameters

$$\begin{aligned} \log(\mathcal{L}) &= \log \left( \prod_i \frac{(\mu_{s,i} + t_{\alpha}\mu_{b,i})^{N_{\text{on},i}} e^{-(\mu_{s,i} + t_{\alpha}\mu_{b,i})}}{N_{\text{on},i}!} \cdot \frac{\mu_{b,i}^{N_{\text{off},i}} e^{-\mu_{b,i}}}{N_{\text{off},i}!} \right) \\ &= \sum_i N_{\text{off},i} \log(\mu_{b,i}) + N_{\text{on},i} \log(\mu_{s,i} + t_{\alpha}\mu_{b,i}) - (t_{\alpha} + 1)\mu_{b,i} - \mu_{s,i}. \end{aligned} \quad (5.12)$$

The  $\mu_s$  can be calculated by assuming a spectral model and calculating the gamma-ray flux of true energy  $N(E_T)$  and feeding the result through the detector's response function. This essentially reduces the number of free parameters in the likelihood to the number of parameters in the spectral model. Section 5.6 will describe this process in detail. Alternatively, the count spectrum can be fitted directly by applying the energy migration matrix to a count vector as outlined in the section about unfolding 5.7. Often no such instrument responses can be supplied for the background counts as it would require expensive simulations of hadronic interactions in the atmosphere for different observation conditions. In many use cases, the likelihood parameters for the background counts  $\mu_b$  are of no particular interest in terms of physics. The value of  $\mu_b$  carries no information about the actual gamma-ray source that is being observed. Hence, the  $\mu_b$  can be considered as a nuisance parameter to this statistical model. There are three ways to deal with this.

First, interpret the background measurements as some form of God-given *truth* and discard any uncertainty about it. The likelihood in this case only consists of a single Poisson term

$$\mathcal{L}(N_{\text{on}}, N_{\text{off}}, t_\alpha \mid \mu_s, ) = \prod_i P_{\text{on},i}(N_{\text{on},i} \mid \mu_{s,i} + t_\alpha N_{\text{off},i}).$$

Second, disregard the  $\mu_b$  by building the profiled, or concentrated, likelihood and expressing the  $\mu_b$  as a function of the other parameters. The profiled likelihood only considers the subspace of the likelihood where  $\frac{\partial \log(\mathcal{L})}{\partial \mu_b} = 0$ . Intuitively speaking, the profiling selects only the *ridge* in the likelihood landscape in which the nuisance parameters maximize the likelihood [35, page 188]. Applying the partial derivative with respect to  $\mu_b$  and setting it to zero yields

$$\frac{\partial \log(\mathcal{L})}{\partial \mu_b} = \frac{N_{\text{on}}}{\mu_b} + \frac{N_{\text{on}} t_\alpha}{\mu_s + t_\alpha \mu_b} - t_\alpha - 1 = 0. \quad (5.13)$$

This equation can be solved for  $\mu_b$  resulting in an expression which depends on  $N_{\text{on}}, t_\alpha, \mu_s$  and  $N_{\text{off}}$

$$\mu_b = \frac{N_{\text{off}} t_\alpha + N_{\text{on}} t_\alpha - t_\alpha \mu_s - \mu_s - \sqrt{K}}{2 t_\alpha (t_\alpha + 1)},$$

where

$$K = t_\alpha^2 (N_{\text{off}}^2 + 2 N_{\text{off}} (N_{\text{on}} + \mu_s) + N_{\text{on}}^2 - 2 N_{\text{on}} \mu_s + \mu_s^2) + 2 t_\alpha \mu_s ((N_{\text{off}} - N_{\text{on}}) + \mu_s) + \mu_s^2.$$

This expression can be substituted into (5.12) resulting in the profile likelihood which has no dependence on  $\mu_b$  anymore. This profiled likelihood is known as *WSTAT* among X-ray astronomers [65] and some H.E.S.S. members. It is a common choice for fitting spectral

models to IACT data. Literature about it is sparse, however. Johannes King's dissertation gives an exhaustive review of its properties and potential biases [92].

The third way to deal with the  $\mu_b$ , is to allow them to remain free during likelihood maximization. While this increases the number of parameters in the fit, it properly takes any statistical uncertainty in the background measurement into account. In the following sections I will use the full likelihood as it is given in (5.11) including all nuisance parameters. Likelihoods like these are very common in high-energy physics. Often they are maximized directly using iterative numerical optimization methods. Errors on the resulting parameters are then estimated by building frequentist confidence intervals using likelihood-ratio tests. For high-dimensional problems, these methods quickly reach their limitations in terms of computing time and robustness. Especially when dealing with many nuisance parameters, the Bayesian approach can be easier to use and interpret. In section 5.6 I use Markov-Chain Monte Carlo (MCMC) sampling on the full posterior distributions including all nuisance parameters from the background to fit a spectral model. Some basics about MCMC are given in the next section.

## 5.5 Posterior Sampling using Markov Chain Monte Carlo

There is a plethora of specialized algorithms to sample random variates from common univariate distributions. The Ziggurat algorithm [108], for example, is one of these specialized methods for sampling numbers from a Gaussian distribution. Typical likelihoods and their posterior distributions, like the ones build in the previous section, cannot be sampled directly as they do not resemble any common probability distribution and are highly problem-specific. Additionally, these likelihoods are often expensive to compute, which makes simple rejection sampling highly problematic in terms of computing cost. The likelihood computation sometimes includes the entire detector simulation or numerical integration routines over the instrument response functions. The next section 5.6 will show an example of that. Markov Chain Monte Carlo (MCMC) methods can be used to sample arbitrary multivariate probability distributions. In this case we want to create samples from the posterior distribution defined by our likelihood and our prior assumptions on its parameters. The posterior distribution is build from the likelihood using Bayes' theorem

$$p(\mu | \mathbf{x}) = \frac{\pi(\mu)\mathcal{L}(\mathbf{x} | \mu)}{g(\mathbf{x})}$$

where  $\mu$  is the parameter vector,  $\mathbf{x}$  is the data vector,  $\mathcal{L}$  is the likelihood, and  $\pi(\mu)$  is the prior distribution of the parameters. The normalization  $g$  is often omitted from the equation. Its calculation  $g(\mathbf{x}) = \int \pi(\mu)\mathcal{L}(\mathbf{x} | \mu) d\mu$  is expensive when  $\mu$  has many dimensions and, maybe more importantly, the shape of the posterior does not depend on the normalization.

A simple form of a Monte Carlo method to tackle this problem is the method of **rejection sampling**. Rejection sampling makes use of a *proposal density*  $Q$ , which is *simpler* than  $p(\mu)$  in the sense that we can directly draw samples from it. Following the definition in [104], we assume for some constant  $c$

$$cQ(\mu) > p(\mu) \quad \forall \mu.$$

We draw two random numbers. First,  $\mu'$  is sampled from  $Q$  and then a uniformly distributed number  $u \in [0, cQ(\mu')]$  is chosen. Add  $\mu'$  to the set of samples if  $u \leq p(\mu')$  and reject it otherwise. The larger the target density  $p(\mu)$  with respect to  $\mu$ , the more samples get accepted. If  $Q$  and  $p$  are of similar shape the process is relatively efficient. If not, less samples get accepted and more time is spent on computing rejected samples. This simple approach gets problematic in high dimensions. Finding a reasonable proposal density  $Q$  might prove to be impossible without choosing large values of  $c$ .

The **Metropolis-Hastings** [77, 113] method is a Markov Chain Monte Carlo method which alleviates some of the problems mentioned above. The samples generated by Markov Chain methods are generally not independent of each other. As the name suggests, the samples are building a connected chain. Now the proposal density  $Q(\mu; \mu(t))$  depends on the current state  $t$  of the chain  $\mu(0), \mu(1), \dots, \mu(t)$ . In this case the shape of  $Q$  is more or less irrelevant to the problem. As [104, page 365] suggests,  $Q$  might be standard normal distribution centered on the current position  $\mu(t)$ . Again, we generate a proposal  $\mu'$  from  $Q$  and either accept or reject it based on a probability. The probability to accept  $\mu'$  as the new  $\mu(t+1)$  for the Metropolis-Hastings algorithm is

$$p_{\text{accept}} = \min\left(1, \frac{p(\mu')}{p(\mu(t))} \frac{Q(\mu(t); \mu')}{Q(\mu'; \mu(t))}\right) \quad [63, 104, \text{page 366}]. \quad (5.14)$$

If a new sample gets accepted, it will be added to the chain. If not, the old state is used again. In this case the  $\mu(t+1) = \mu(t)$  and the chain contains the same sample twice. It can be shown that for  $t \rightarrow \infty$ , the chain  $\mu(t)$  converges in distribution to  $kp(\mu)$ , where  $k$  is an arbitrary constant [100, S.21]. Determining the speed of convergence of a MCMC chain is difficult. Even making a reliable statements about whether a chain has converged is not an easy task. In practice the number of required samples in a chain is estimated using



its autocorrelation length [63, S.3]. A modified variant of a Metropolis-Hastings algorithm was used in chapter 4 to fit the SED model of the Crab Nebula to flux points. This MCMC algorithm uses an ensemble of samplers to generate proposals in parallel. It is implemented in the open-source `emcee` project [63]. The algorithm implemented in `emcee` is very robust for many use cases and easily executed in parallel. If the evaluation of the model  $p(\mu)$  is expensive or the acceptance fraction is too low, building a long enough chain of samples might still take too long. The proposal distribution  $Q$  essentially *guesses* the direction in which the chain should move. We can use information about the shape of  $p$  to avoid the *random walk* behavior of the chain.

**Hamilton Monte Carlo** methods use the gradient of the (log) posterior with respect to  $\mu$  to generate proposals. This method is inspired by the physics of hamiltonian dynamics. In addition to the parameter vector  $x := \mu$  we sample a momentum vector  $m$ . We interpret the target distribution as a physical system with a Hamiltonian for the position variable  $x$  and the momentum variable  $m$

$$H(x, m) = E(x) + K(m), \quad (5.15)$$

where  $E$  is the posterior distribution of interest, the “potential energy” in terms of Hamilton dynamics, and  $K \propto m^T m$  models the “kinetic energy” of an object bound to the surface of  $p(x)$ . We sample the *canonical distribution* of system states

$$P(x, m) \propto e^{-H(x, m)} = e^{-E(x)} e^{-K(m)}. \quad (5.16)$$

Any target distribution  $p(x)$  can be expressed as a canonical distribution over energies by setting  $E(x) = -\log(p(x))$ . The MCMC chain now consists of two variables per time step. The momentum variable  $m$  is of no interest to the inference made on the model. It is simply a tool to apply Hamilton dynamics to this system. Since  $P(x, m)$  can be separated into marginalized distributions, the  $m_i$  in the chain can simply be discarded once calculation is complete. Two proposals are generated per time step to create the next sample  $(x(t+1), m(t+1))$  in the chain. A momentum  $m_s$  is proposed from a standard normal distribution. A path of length  $L$  starting from  $(x_t, m_s)$  is followed according to the constraints set by Hamiltonian dynamics

$$\begin{aligned} \dot{x} &= m \\ \dot{m} &= -\frac{\partial E(x)}{\partial x}. \end{aligned} \quad (5.17)$$

This can be done using any solver which preserves the volume of the parameter space. The position and momentum found at the end of the path is the new proposal  $(x', m')$ . Equivalent to the Metropolis step shown in equation 5.14, this proposal gets accepted with probability

$$p_{\text{accept}} = \min\left(1, \frac{P(x', m')}{P(x(t), m_s)}\right) = \min\left(1, e^{-H(x', m') + H(x(t), m_s)}\right) \quad [84, 104, \text{page 388}]. \quad (5.18)$$

In a physical interpretation this implies that samples whose total energy  $H$  is smaller than that of the the previous sample will always get accepted.

In the following section I use the PyMC3 [131] project to sample the full model as specified in equation 5.11 using a variant of the Hamilton MCMC method. Explicit knowledge about the gradient of the likelihood is still needed. This includes the forward propagation of an astrophysical flux model through the detector's instrument response function as discussed in the next section.

## 5.6 Fitting Spectral Models

PyMC3 provides a convenient high-level user interface to create statistical models by combining and nesting elementary distributions. The stochastic model is transformed into a computational graph by the Theano [138] backend. The graph connects all input variables, the spectral parameters in this case, to the outputs, the value of the posterior distribution. Like many of its competitors, e.g. PyTorch [128] and Tensorflow [109], to name just two, Theano optimizes runtime by pruning paths in its computational graph. Once the graph is optimized, Theano creates C++ code for execution on the CPU or CUDA [120] code for execution on the GPU. The resulting chain of operations is then automatically differentiated using the backpropagation algorithm [72, 102]. This way PyMC3 can use the full gradient of the posterior with respect to the spectral parameters to run Hamilton Monte Carlo algorithms. PyMC3 [131] uses the NUTS [84] algorithm to solve equation 5.17. It requires no additional tuning of path length or step size. The PyMC3 project supports the usage of custom operations in order to model more complex processes. This is needed to model the measurement process of an IACT. The main hurdle that needs to be overcome in order to use PyMC3 is the implementation of a custom differentiable Theano operation which describes the interaction of the spectral model with the instrument response. The following steps are applicable to all sorts of spectral models; however, I will concentrate on the log-parabolic spectral shape introduced in section 2.4. Fitting this model means to find those parameters which are most likely to

represent the observed data according to the statistical model we defined in section 5.4 in particular in equation (5.11). For a log-parabolic spectrum we are fitting three model parameters  $N(E; N_0, \alpha, \beta)$  as defined in equation (2.10). The likelihood has two parameter vectors  $\mu_s$  and  $\mu_b$ . Each  $\mu_s$  models the expected number of observed counts as predicted by the assumed spectral model applied to the instrument's response function. In other words, the entries in the  $\mu_s$  vector represent the expected counts in the corresponding bin of estimated energy as seen by the telescope's analysis. As discussed before, the background parameters  $\mu_b$  are considered nuisance parameters. They do not carry information about the gamma-ray source under observation. As seen in section 5.2 the number of expected events  $\mu_s$  can be calculated from the model  $N$  when both the migration matrix  $\mathbf{p}_D$  and the effective area  $A_{\text{eff}}$  are known. As per equation (5.8) the expected number of counts from the gamma-ray source for an observation of length  $t_{\text{obs}}$  is

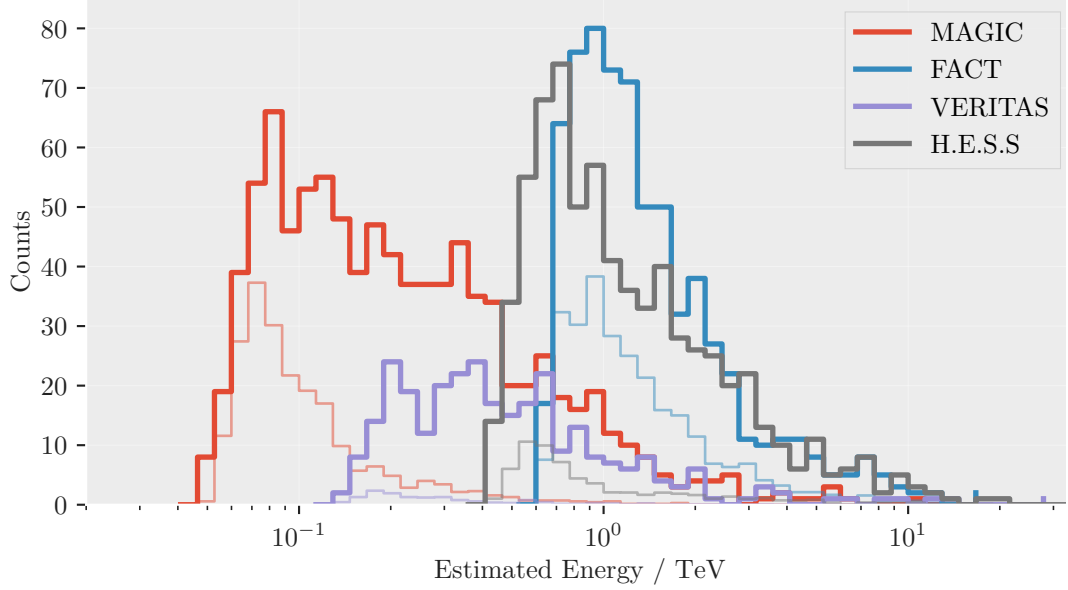
$$\mu_s = \mathbf{p}_D (c \odot A_{\text{eff}}) t_{\text{obs}}, \quad (5.19)$$

where  $\odot$  is the element-wise multiplication operator and  $c$  is the expected gamma-ray flux from the discretized spectral model. The spectral model is discretized per energy bin  $\Delta E_T$  to get the flux within an energy range from the differential flux

$$c_{\Delta E_T} = \int_{\Delta E_T} N(E; N_0, \alpha, \beta) dE. \quad (5.20)$$

Equation (5.19) can only be applied if the size and number of energy bins matches those of  $A_{\text{eff}}$  and  $\mathbf{p}_D$ . The same has to be true for the observed data  $N_{\text{on}}$  and  $N_{\text{off}}$  in order to calculate the likelihood. In case the binning does not match, the IRFs need to be recomputed or interpolated and re-binned to fit the discretized spectral model. The energy bins are always logarithmically spaced. This is necessary for two reasons. First, as discussed in chapter 2, gamma-ray distributions from astrophysical source roughly follow power-law with a negative index. The number of observed events steeply declines with increasing energy. Uniformly spaced energy bins would mostly be empty. Second, the simulated events also follow a power-law distribution. This has practical reasons. The higher the primary particle's energy, the larger the number of secondary particles in the shower and consequently the computing requirements. Hence, the simulated particle spectrum is often built as a power-law distribution. Logarithmic bins in some variable  $E$  and basis  $b$  are defined as

$$\log_b(E_{i+1}) = \log_b(E_i) + w, \quad (5.21)$$



**Figure 5.3:** Counts from the signal and background regions of the telescope's published datasets of the Crab Nebula. The thick colored lines show the counts in the signal region. The thin lines show the corresponding background counts  $\propto N_{\text{off}}$ . The number of bins is the same for all data for both signal and background regions. The MAGIC and VERITAS telescopes are more sensitive in the lower energy ranges than FACT. FACT, being only a single telescope with a 4 m mirror, collects less light than the other telescopes. H.E.S.S. can only observe the Crab Nebula at very low altitude angles because it is located far south of the equator in Namibia. This makes H.E.S.S. less sensitive to this particular source. Note that the *estimated* energy can differ wildly between telescopes. Energy estimators have inherent biases and finite resolutions. A 1 TeV shower seen by FACT will have a different estimated energy than the energy estimated by the MAGIC telescope on the same shower.

where  $w$  is the constant, logarithmic bin width. The bin edges can hence be calculated as

$$E_{i+1} = b^{\log_b(E_i)+w} = b^{\log_b(E_i)} b^w = E_i b^w. \quad (5.22)$$

This form of binning is used for both estimated and true energy in the stored IRFs and the observed count vectors. The data I use to fit the Crab Nebula is binned with 20 bins per decade of energy. It is important to take all bins into account in which the telescope still has a non-zero trigger probability. Event counts can migrate between bins when the predicted count vector in true energy ( $c \odot A_{\text{eff}}$ ) is multiplied by the dispersion matrix. For the data fitted here a total of 80 bins in true energy are used. Figure 5.3 shows the on and off counts for the Crab Nebula data published for the joint analysis in the previously mentioned paper [122]. The figure shows the number of recorded events per bin in *estimated* energy. This data will be used to fit a flux model of the Crab Nebula.

The vector of expected signal counts  $\mu_s$  used in the likelihood defined in equation (5.11) depends on the parameters of the spectral model. In order to use Hamilton Markov Chain methods, the gradient of the posterior  $p(\mu_s(N_0, \alpha, \beta, ), \mu_b | N_{\text{on}}, N_{\text{off}})$  with respect to the spectral model parameters  $N_0$ ,  $\alpha$ , and  $\beta$  has to be calculated. I use the automatic differentiation capabilities of Theano and PyMC3 to build the gradient. In a slightly oversimplified view, automatic differentiation finds the gradient of a function by repeated application of the chain rule. Doing so for the likelihood  $\mathcal{L}$  is not problematic as it consist of elementary functions, the Poisson density, with known gradients. The signal parameter  $\mu_s(N_0, \alpha, \beta, )$  includes the calculation of the integral in equation (5.20). The integral can be solved analytically. To simplify, I replace the energy  $E$  in equation (2.10) with a unitless variable  $x$  and drop the normalization  $E_0$ . The integral becomes

$$\begin{aligned} \int N(x; N_0, \alpha, \beta) dx &= N_0 \int x^{-\alpha-\beta \log_{10}(x)} dx \\ &= N_0 \int x^{-\alpha-\beta' \log(x)} dx \\ &= N_0 \int \exp(-\alpha \log(x) - \beta' \log^2(x)) dx, \end{aligned}$$

where  $\beta' = \frac{\beta}{\log(10)}$  and the substitution  $x = e^{\log(x)}$  was used in the last line. Now the integration variable is substituted with  $u = \log(x)$  and  $du = \frac{1}{x} dx$ . Solving for  $du$  and replacing  $x = e^{\log(x)} = e^u$  yields

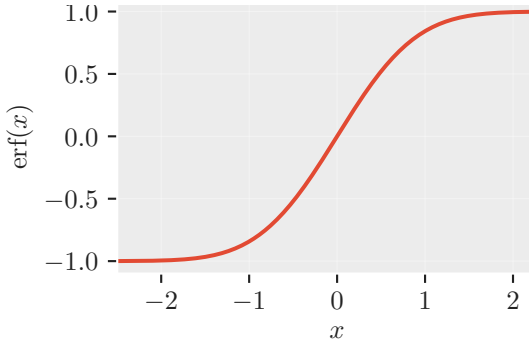
$$\begin{aligned} \int N(x; N_0, \alpha, \beta) dx &= N_0 \int \exp(-\alpha u - \beta' u^2) \exp(\log(u)) du \\ &= N_0 \int \exp(-((\alpha - 1)u + \beta' u^2)) du \\ &= N_0 \frac{1}{2} \sqrt{\frac{\pi}{\beta'}} \exp\left(\frac{(\alpha - 1)^2}{4\beta'}\right) \operatorname{erf}\left(\log(x) \sqrt{\beta'} + \frac{(\alpha - 1)}{2\sqrt{\beta'}}\right), \quad (5.23) \end{aligned}$$

where  $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$  is the error function. The last step in equation (5.23) is the solution to integral number 2.325.13 in the integral table compiled by Gradshteyn and Ryzhik [70, page 108]. The Theano library includes a symbolic tensor operation to compute the error function and its derivatives. Hence, equation (5.23) can be used to sample the posterior distribution with PyMC3 having all gradients available via automatic differentiation. Unfortunately, the integral's solution is very sensitive to round-off errors. To calculate the definite

integral of  $N(x; N_0, \alpha, \beta)$  between points  $a$  and  $b$ , the difference between two error functions has to be evaluated

$$\int_a^b N(x; N_0, \alpha, \beta) dx \propto \left[ \operatorname{erf}\left(\log(b)\sqrt{\beta'} + K\right) - \operatorname{erf}\left(\log(a)\sqrt{\beta'} + K\right) \right], \quad (5.24)$$

where  $K = (\alpha-1)/2\sqrt{\beta'}$ . The error function converges quickly to  $\lim_{x \rightarrow \infty} \operatorname{erf}(x) = 1$  for large values of  $x$  and, since erf is odd,  $\lim_{x \rightarrow -\infty} \operatorname{erf}(x) = -1$  for small values of  $x$  respectively.



**Figure 5.4:** The error function  $\operatorname{erf}(x)$  quickly converges to 1. For values of  $x \gtrsim 6$ , the implementation of the error function returns 1 due to the limited precision of 64 bit floating-point arithmetic.

The quick convergence becomes apparent in figure 5.4, which shows the error function for values close to 2. During Markov chain sampling, unfavorable combinations of  $\alpha$  and  $\beta'$  can get evaluated where  $\beta'$  is much smaller than  $\alpha$ . For these configurations the argument of the error function easily reaches values larger than 6, at which point  $1 - \operatorname{erf}(6) \approx 10^{-17}$ . This value lies far below the machine epsilon for IEEE floats with 64 bits of approximately  $2.2 \times 10^{-16}$ . For this reason, large values of  $a$ ,  $b$ , or  $K$  lead to rounding errors when evaluating the error function. For these values the error function will be rounded to unity  $\operatorname{erf}(6) \stackrel{64 \text{ bit}}{=} 1$ . Hence, the evaluation of  $\int_a^b N(x; N_0, \alpha, \beta) dx$  wrongly returns 0 for large parts of the sampled parameter space due to the limited floating point precision. The problem can be circumvented by approximating the integral in equation (5.20) by numerical methods.

Despite the fact that Hamilton Monte Carlo needs less sample values than a non-Hamilton approach, several thousand samples are still needed for this particular posterior. Since the gradient gets evaluated for each energy bin independently, the total number of integral computations quickly reaches many millions. While it is a relatively crude way to approximate the area under a curve, I implemented the trapezoidal rule since it can be easily vectorized to compute the integral in all energy bins at once. The trapezoidal rule approximates the definite integral  $\int_a^b f(x) dx$  by partitioning the integration interval into several smaller regions in which  $f(x)$  is assumed to be linear. Any function  $f(x)$  can be approximated by a combination of linear functions. The finer the partitioning, the better the approximation. The area

under the linear function in partition  $i$  is simply the width of the partition  $\Delta x_i = x_{i+1} - x_i$  times the average value of  $f(x)$  in that partition

$$\int_a^b f(x) dx \approx \sum_{i=0}^{N-1} \frac{1}{2} (f(x_i) + f(x_{i+1})) \Delta x_i, \quad (5.25)$$

where  $N$  is the number of partitions and  $a \leq x_i < x_{i+1} \leq b$ . This can be easily generalized to integrate vector functions  $\mathbf{f} : \mathbb{R}^M \rightarrow \mathbb{R}^M$ , where  $M$  is the number of energy bins in the model. Using the trapezoidal rule, the gradients  $\frac{\partial c}{\partial \alpha}$ ,  $\frac{\partial c}{\partial \beta}$ , and  $\frac{\partial c}{\partial N_0}$  can now be expressed as a sum of elementary functions. In addition to the potential speed improvements, this allows the use of automatic differentiation to find the derivatives of the posterior distribution with respect to the model parameters. At this point we have all necessary ingredients to define the model priors and sample the posterior distribution using PyMC3. The priors on the spectral parameters are chosen to be uniform distributions  $U$  which have their lowest bound at 0 to avoid negative values during sampling. In this context the uniform prior conveys the lack of knowledge about these model parameters. The calculation of the  $\mu_s$  vector given in equation (5.19) is performed separately for each observation taking into account the different instrument responses per observation. The complete statistical model can be summarized as

$$N_0 \sim U(a=0, b=\infty)$$

$$\alpha \sim U(a=0, b=\infty)$$

$$\beta \sim U(a=0, b=\infty)$$

$$\mu_s \sim \sum_{\text{Observations}} \mathbf{p}_D(c(N_0, \alpha, \beta) \odot A_{\text{eff}}) t_{\text{obs}}$$

$$\mu_b \sim U(a=0, b=\infty)$$

$$p \sim \text{Poisson}(\lambda = \mu_s + t_\alpha \mu_b \mid N_{\text{on}}) \cdot \text{Poisson}(\lambda = \mu_b \mid N_{\text{off}}).$$

This set of equations is an almost verbatim copy of the PyMC3 code used to define the model in Python. I sample the model with the  $N_{\text{on}}$  and  $N_{\text{off}}$  data as depicted in figure 5.3 for each telescope separately. The Markov chain for this fit was built from 10 000 samples in 6 chains. On recent multi-core CPUs the sampling will take less than a minute per dataset when the chains are sampled in parallel. Section 5.6 shows the results of the sampled parameter values. The lines in figure 5.5 show the fitted models together with the SSC model calculated in chapter 4. The fit ranges for the single telescope were chosen according to the ranges selected

**Table 5.2:** Fit results for a log-parabolic model as defined in equation (2.10) for the statistical model defined in section 5.4 including all nuisance parameters. The numbers are computed by calculating the 16<sup>th</sup>, 50<sup>th</sup> and 84<sup>th</sup> percentile of the marginalized sample distributions. These numbers are compatible with the ones presented in our paper [122], which used a numerical likelihood minimization to find the best fit and a likelihood-ratio test to determine the errors. As shown in figure 5.5, these parameters closely match the shape of the Crab Nebula SED fitted in chapter 4.

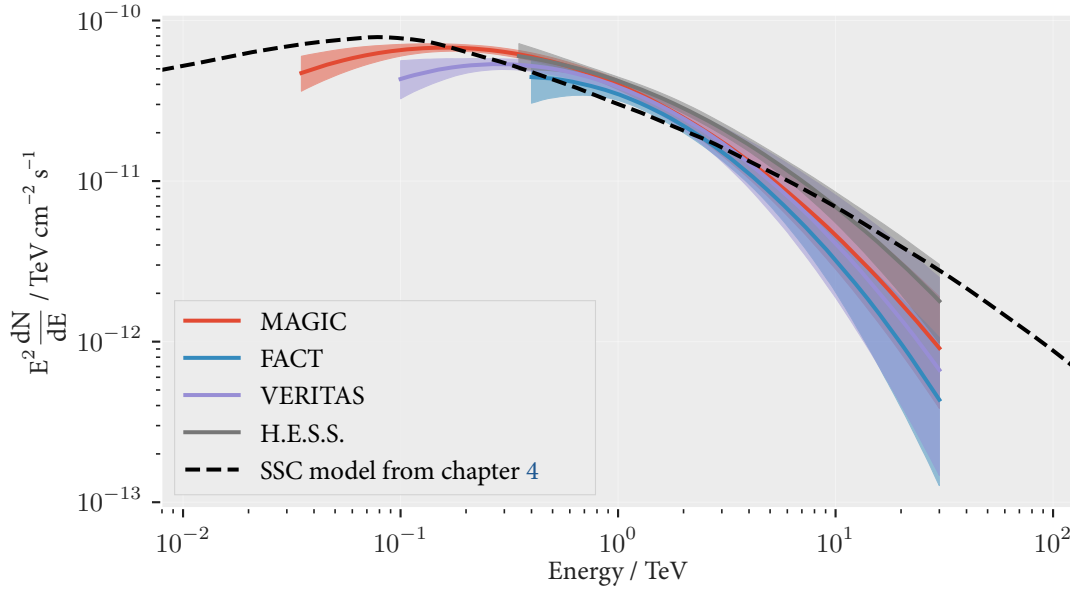
Telescope	$N_0/10^{-11}\text{TeV}^{-1}\text{cm}^{-2}\text{s}^{-1}$	$\alpha$	$\beta$
FACT	$3.47^{+0.31}_{-0.30}$	$2.48^{+0.21}_{-0.24}$	$0.56^{+0.38}_{-0.29}$
MAGIC	$3.99^{+0.29}_{-0.28}$	$2.58^{+0.10}_{-0.09}$	$0.36^{+0.10}_{-0.10}$
H.E.S.S.	$4.21^{+0.23}_{-0.22}$	$2.48^{+0.11}_{-0.12}$	$0.31^{+0.18}_{-0.16}$
VERITAS	$3.86^{+0.38}_{-0.35}$	$2.51^{+0.14}_{-0.12}$	$0.46^{+0.23}_{-0.20}$

in our paper [122] to make the results comparable. Appendix A.5 gives some implementation details for the PyMC3 and Theano programs used to produce these results. All code necessary to reproduce these results is available at

[https://github.com/tudo-astroparticlephysics/ll\\_experiments](https://github.com/tudo-astroparticlephysics/ll_experiments) 

The repository contains two command line applications which can read the published data and store the sampled Markov chains to disk. A configuration file allows to modify the energy ranges and number of bins.





**Figure 5.5:** Results of the PyMC3 fit to the published data used in our paper [122]. The best fit is taken to be the median of the sample values in the Markov chains. The colored lines indicate the log-parabolic spectral model corresponding to these parameters. The error bands around the spectral lines are computed by evaluating the spectral model for 5000 randomly chosen parameter combinations from the Markov chain. Of these values the area between the 16<sup>th</sup> and 84<sup>th</sup> percentile is shown. As a visual reference, the dashed black line shows the best fit result of the SSC model from chapter 4. Note that even though flux points from H.E.S.S. and MAGIC are used in chapter 4, the data used here comes from different observations. The SSC model takes a much larger energy range into account and uses data from a different observation period. It does, however, match the fitted spectra remarkably well. The pivot point for the inverse-compton emission of the SSC model is lower than that of the fitted log-parabola models. This could indicate a bias in the assumed spectral model for the fit for which the pivot energy was fixed at 1 TeV. This effect might be mitigated if the IACT could detect lower energies. The high-energy part of the fitted models also drops off more quickly than the SSC model. This could indicate a bias in the SSC model due to the meager amount of flux points in the TeV range used to fit its parameters.

## 5.7 Unfolding of Flux Points

In the previous section an astrophysical source model was fitted to measured count spectra. The predictions of the assumed model were fed through the detector's response to imitate the process of observing the source with a telescope. The resulting count spectrum was then compared to the measured count spectrum. The unfolding approach is independent of a physical source model. Unfolding is a ubiquitous problem throughout particle physics and image processing. Detectors in high-energy physics often observe the by-products of particle interactions, as is the case for IACT measurements. The primary particles properties are obfuscated by finite detector resolution and the stochastic nature of particle interactions. Wherever counts of particles are measured, limited acceptance probabilities obscure the true particle distributions. In order to find the true distribution of the primary particles with respect to some observable, e.g. kinetic energy, the detector effects have to be *reversed*. Literature about unfolding often describes the measurement process in terms of an integral equation

$$g(\mathbf{x}) = \int_{\Omega} A(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y} = (L f)(\mathbf{y}). \quad (5.26)$$

In this abstract representation,  $A$  describes the detector's response to particles with respect to the observable  $\mathbf{y}$  and true particle distribution  $f(\mathbf{y})$ . The linear operator  $L$  describes the operation in terms of an integral transform. The distribution of observed events with respect to some estimated variable is  $g(\mathbf{x})$ . We are interested in finding the true distribution  $f$  from the observed  $g$ . In mathematics this kind of task is called an *inverse problem*. Research about inverse problems is plentiful. With the rise of medical imaging techniques and the availability of cheap imaging sensors, interest in solving these problems has increased over the recent decades. The excellent review article "Modern Regularization Methods for Inverse Problems" provides a rather extensive overview of the historical development and current state-of-the-art methods [29]. Popular methods for tackling inverse problems, especially in the field of high-energy physics, include the expectation-maximization method [54], variational methods, building the SVD inverse, and even machine-learning approaches like DSEA [42] and network-based approaches like [13, 45, 88]. The machine-learning based methods require simulated, labeled, data to build a model. No simulated data is available in the open datasets used in this chapter and a variational method is employed. Problems such as equation (5.26) are often ill-posed as no unique solution exists. The general idea to solve the ill-posed problem is to approximate it by well-posed ones [29, page 3]. To ensure the existence of a unique solution, the space of possible solutions is constrained via *regularization*. Variational methods solve equation (5.26) by minimizing a loss function  $F(g, (L f)(\mathbf{y}))$ , which in the case of least-

squares regression is simply the  $L^2$  norm of  $(Lf)(\mathbf{y}) - g(\mathbf{x})$ . The shape of the solution can be constrained by including a regularization term in the loss function

$$\hat{g} = \arg \min_f \|(Lf)(\mathbf{y}) - g(\mathbf{x})\| + T(f(\mathbf{y})), \quad (5.27)$$

where  $T(f(\mathbf{y}))$  is the so-called Thikonov regularization term. The Thikonov function  $T$  suppresses undesirable solutions and often depends on a parameter  $\alpha$  which controls the *strength* of regularization. A common choice for the regularization term is  $T(f(\mathbf{y})) = \|\alpha f(\mathbf{y})\|$  which penalizes solutions with large norms. This special form of regularization is called  $L^2$  regularization or ridge regression in the machine-learning community's vernacular. Another way to simplify the inverse problem is by approximating the solution  $f(\mathbf{x})$  and the operator  $L$  by a discretized form. The equation above then simplifies to a matrix multiplication  $\mathbf{x} = A\mathbf{y}$ . This can be understood as a form of regularization, where the solution is forced into the real vector space  $\mathbb{R}^{\dim(\mathbf{x})}$ . In the matrix formulation the problem can be solved by variational methods just as well. The loss function  $F$  can be motivated from statistical arguments as well. The likelihood of observing  $g$  when given the true distribution  $f$  together with a a-priori probability density  $\pi(f)$  can be used to build the posterior distribution  $p \sim \pi \mathcal{L}$  as seen in section 5.5. In a statistical interpretation this posterior distribution acts as the loss function. The attentive reader will notice that this exact problem, in its discretized form, has already been solved in the previous section 5.6. The equation (5.26) above corresponds to equation (5.4) which in its discretized form becomes equation (5.19). In the previous section we regularized the problem further by constraining the vector of true counts  $c$  to follow the shape of the log-parabolic spectral model by integrating the model in energy bins as stated in equation (5.20). For IACT measurements we can solve the inverse problem to directly produce discrete flux points. In this case the flux of the source is assumed to be constant within an energy bin so that equivalently to equation (5.20)

$$c_{\Delta E_T} = \int_{\Delta E_T} N(E) dE = N_{\Delta E_T} \Delta E_T,$$

where  $N_{\Delta E_T}$  is the flux of the source in energy bin  $\Delta E_T$ . All that has changed in comparison to the previous section is the shape of the spectral model. The new spectral model has as many parameters as the number of bins in true energy, which makes the problem harder to solve. Despite the fact that the model-dependent approach and the unfolding approach can be interpreted, and implemented, as one and the same variational problem, high-energy physicists make a clear distinction between the two methods. The input data remains the same, but is binned differently. I use 5 bins per decade of true energy and 20 bins per decade

of estimated energy. Two additional *overflow bins* are used on the true energy axis in order to cover the entire range of energies in which the trigger probability is not zero. The fit range is selected from the range in which the number of excess counts  $N_{\text{excess}} = N_{\text{on}} - t_{\text{obs}}N_{\text{off}}$  meets a threshold of at least 4 counts. More energy bins can be used for longer observations with more data or when constraining the likelihood with stronger priors. The newly binned count spectra from the signal and background regions in the sky are entered into the likelihood from equation (5.11). The vector of expected counts from the background region  $\mu_b$  is directly constrained by the entries in  $N_{\text{off}}$ . Just as we did in the previous section, the background counts  $\mu_b$  are considered to be nuisance parameters without further physical meaning. They are only needed to calculate the expected counts in the on-region  $\mu_s + t_{\alpha}\mu_b$ . As in the previous section I use PyMC3 to sample the posterior using Markov chains. The prior for the expected flux  $N$  is an almost flat normal distribution which is truncated at 0 to avoid sampling of negative numbers. A simplified view of the model is given by the equations below.

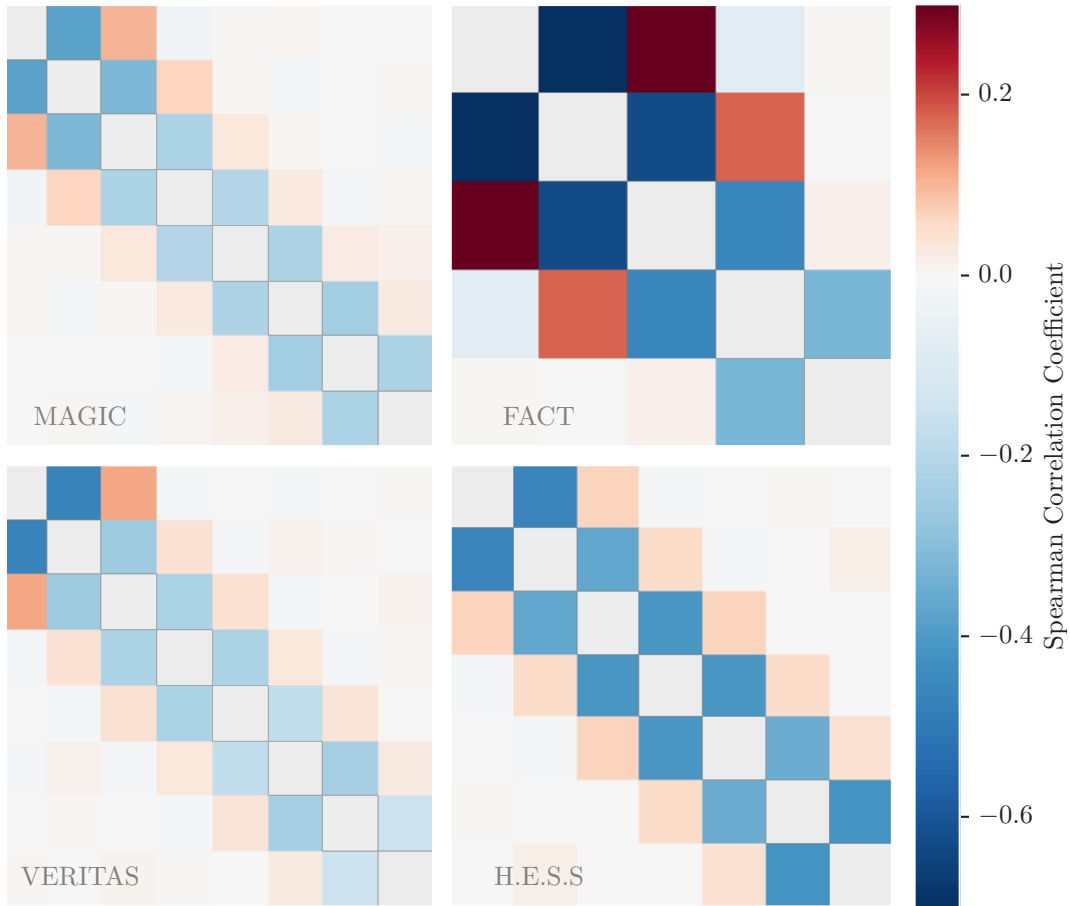
$$N \sim \text{TruncatedNormal}(\mu=0, \sigma=50)$$

$$\mu_b \sim \text{U}(a=0, b=\infty)$$

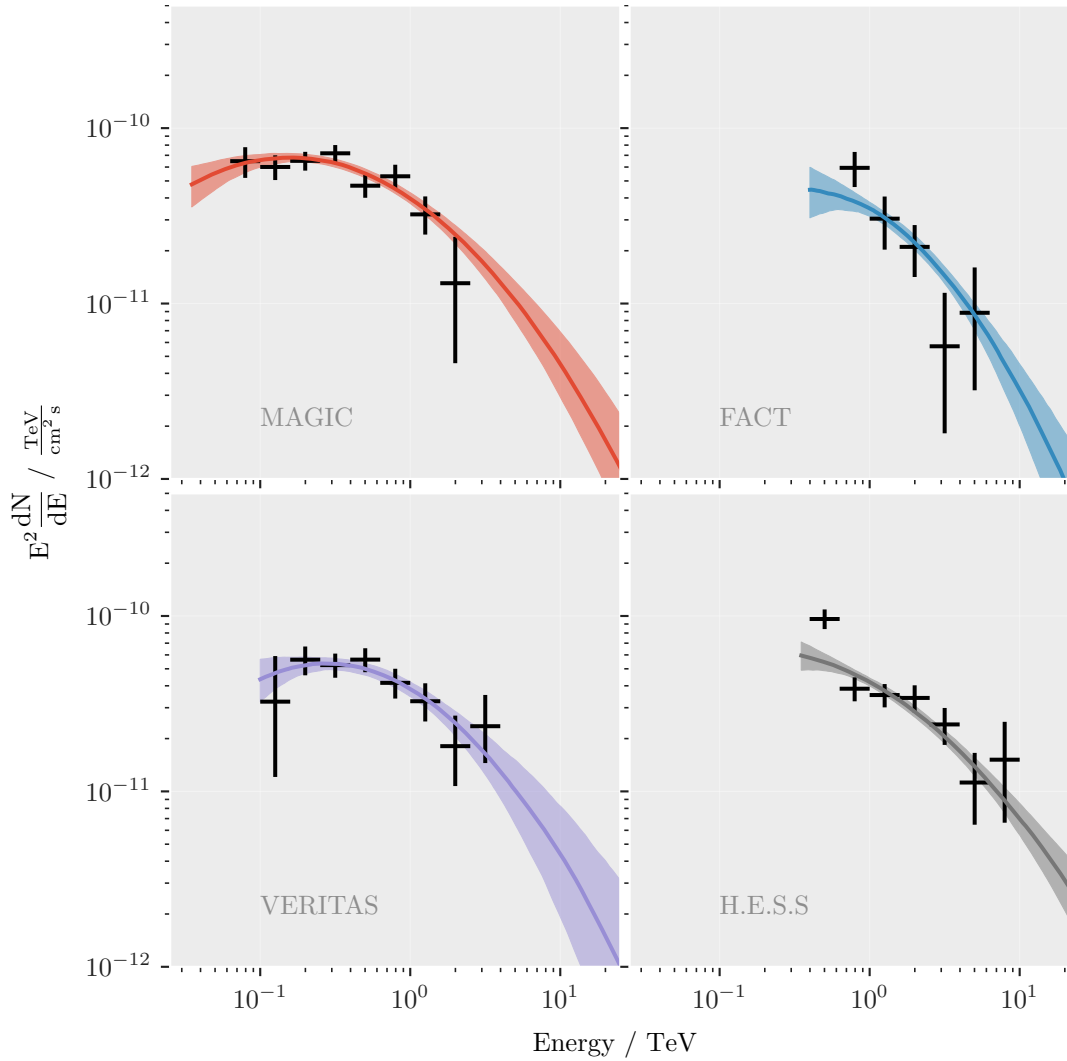
$$\mu_s \sim \sum_{\text{Observations}} \mathbf{p}_D(N \Delta E_T \odot A_{\text{eff}}) t_{\text{obs}}$$

$$p \sim \text{Poisson}(\lambda = \mu_s + t_{\alpha}\mu_b \mid N_{\text{on}}) \cdot \text{Poisson}(\lambda = \mu_b \mid N_{\text{off}})$$

Figure 5.7 shows the resulting flux points overlaid on the spectral models fitted in the previous section 5.6. The Markov chain was built from 3000 samples in 6 chains. Table 5.3 lists the flux points for each telescope including their 16<sup>th</sup> and 84<sup>th</sup> percentile. The fitted fluxes are correlated with their neighboring bins as shown in figure 5.6. Correlations are calculated between the samples in the Markov chain using Spearman's correlation coefficient.



**Figure 5.6:** The figure shows the correlation between the parameters as sampled by PyMC3 for each of the 4 telescopes. For unfolding there are as many parameters as energy bins. These correlations show the interdependence between neighbouring bins in the resulting fluxes. Bins with lower energies are on the top-left of each matrix. High correlations indicate a strong migration between bins in energy ranges, where the dispersion matrix  $\mathbf{p}_D$  has a rather broad distribution. This effect is stronger towards the lower end of the energy range, at which point the energy estimation has a larger resolution. The diagonal elements in the matrices are all equal to 1 and masked out in these pictures. The overflow bins are not included in this representation. The low-energy bins for the FACT dataset show the highest correlation in this comparison. Indeed, the comparatively large amount of background events in the dataset and the large energy resolution visible in figure 5.1 have an adverse effect on the accuracy of the unfolding, not to mention the fact that the energy dispersion matrices  $\mathbf{p}_D$  for the telescopes are calculated from simulations which cannot represent real data perfectly.



**Figure 5.7:** Results of the PyMC3 fit to the published data used in our paper [122]. The black error bars show the unfolded flux points. As before, these values are calculated from the 16<sup>th</sup>, 50<sup>th</sup> and 84<sup>th</sup> percentile of the marginalized posterior distributions gained from the Markov sampler. The colored lines indicate the log-parabolic spectral model as fitted in the previous section 5.6. The fit ranges for the unfolding approach are evidently smaller than those used for the spectral fit. As mentioned in the text, the fit ranges were found by selecting the bin edges in which the distribution of excess events had at least 4 counts. This rather simple heuristic to find smaller fit ranges is required to help the PyMC3 sampler to converge. Intuitively speaking, this is a consequence of the fact that the spectral model regularizes the inverse problem and constraints the shape of the solution. Hence, the direct fitting of flux points has larger uncertainties and can only give reliable estimates in the energy range where sufficient statistics are available. Still, the fitted flux points match the spectral models from the previous section quite well considering the fact that the underlying model assumptions are utterly different.

**Table 5.3:** Results of the PyMC3 *unfolding*. Similar to the results presented in the previous section 5.6, these values are calculated from the 16<sup>th</sup>, 50<sup>th</sup> and 84<sup>th</sup> percentile of the marginalized sample distributions. The input data, count vectors in estimated energy, is discretized into 20 bins per decade in estimated energy. The unfolded flux points are distributed into 5 bins per decade in true energy. The additional *overflow bins* are not shown in the table. The Markov chain consists of 3000 samples in 6 chains and can be sampled in several seconds on a modern CPU.

Energy / TeV		Flux / TeV <sup>-1</sup> s <sup>-1</sup> cm <sup>-2</sup>			
		MAGIC	FACT	VERITAS	H.E.S.S.
0.06	0.10	(1.03 <sup>+0.21</sup> <sub>-0.20</sub> ) 10 <sup>-8</sup>			
0.10	0.16	(3.79 <sup>+0.63</sup> <sub>-0.60</sub> ) 10 <sup>-9</sup>		(2.05 <sup>+1.68</sup> <sub>-1.29</sub> ) 10 <sup>-9</sup>	
0.16	0.25	(1.63 <sup>+0.21</sup> <sub>-0.20</sub> ) 10 <sup>-9</sup>		(1.42 <sup>+0.27</sup> <sub>-0.26</sub> ) 10 <sup>-9</sup>	
0.25	0.40	(7.19 <sup>+0.82</sup> <sub>-0.79</sub> ) 10 <sup>-10</sup>		(5.25 <sup>+0.86</sup> <sub>-0.80</sub> ) 10 <sup>-10</sup>	
0.40	0.63	(1.87 <sup>+0.30</sup> <sub>-0.27</sub> ) 10 <sup>-10</sup>		(2.25 <sup>+0.36</sup> <sub>-0.34</sub> ) 10 <sup>-10</sup>	(3.83 <sup>+0.51</sup> <sub>-0.47</sub> ) 10 <sup>-10</sup>
0.63	1.00	(8.42 <sup>+1.38</sup> <sub>-1.27</sub> ) 10 <sup>-11</sup>	(9.41 <sup>+2.18</sup> <sub>-2.10</sub> ) 10 <sup>-11</sup>	(6.59 <sup>+1.34</sup> <sub>-1.23</sub> ) 10 <sup>-11</sup>	(6.10 <sup>+0.98</sup> <sub>-0.93</sub> ) 10 <sup>-11</sup>
1.00	1.58	(2.03 <sup>+0.54</sup> <sub>-0.47</sub> ) 10 <sup>-11</sup>	(1.92 <sup>+0.65</sup> <sub>-0.64</sub> ) 10 <sup>-11</sup>	(2.06 <sup>+0.55</sup> <sub>-0.48</sub> ) 10 <sup>-11</sup>	(2.24 <sup>+0.35</sup> <sub>-0.33</sub> ) 10 <sup>-11</sup>
1.58	2.51	(3.28 <sup>+2.75</sup> <sub>-2.13</sub> ) 10 <sup>-12</sup>	(5.28 <sup>+1.75</sup> <sub>-1.73</sub> ) 10 <sup>-12</sup>	(4.55 <sup>+2.24</sup> <sub>-1.85</sub> ) 10 <sup>-12</sup>	(8.57 <sup>+1.54</sup> <sub>-1.48</sub> ) 10 <sup>-12</sup>
2.51	3.98		(5.71 <sup>+5.81</sup> <sub>-3.89</sub> ) 10 <sup>-13</sup>	(2.35 <sup>+1.19</sup> <sub>-0.90</sub> ) 10 <sup>-12</sup>	(2.41 <sup>+0.58</sup> <sub>-0.57</sub> ) 10 <sup>-12</sup>
3.98	6.31		(3.53 <sup>+2.87</sup> <sub>-2.25</sub> ) 10 <sup>-13</sup>		(4.47 <sup>+2.13</sup> <sub>-1.90</sub> ) 10 <sup>-13</sup>
6.31	10.00				(2.40 <sup>+1.55</sup> <sub>-1.35</sub> ) 10 <sup>-13</sup>

## 5.8 Notes on Uncertainty Propagation

Error bands like the ones calculated for figure 2.3 and 5.6 cannot easily be estimated by using standard Gaussian uncertainty propagation. This form of error propagation is often used when some function  $f(\mathbf{x})$  depends on parameters  $\mathbf{x}$  whose probability distributions are assumed to be Gaussian and can be described using the covariance matrix  $\mathbf{V}_x$ . In case the function  $f$  is linear in  $\mathbf{x}$  with coefficients  $\mathbf{B}$  it can be written in matrix form as  $f(\mathbf{x}) = \mathbf{B}\mathbf{x}$ . For any two random variables  $X, Y$  and scalars  $a$  and  $b$  the covariance can be transformed like  $\text{Cov}(aX, bY) = ab \text{Cov}(X, Y)$ . The elements in the covariance matrix contain the variance and covariance between the entries in the parameter vector  $\mathbf{x}$ . Hence, the covariance of the parameters  $\mathbf{V}_x$  can be transformed into the covariance matrix for the function  $f$  by using a simple matrix transform

$$\mathbf{V}_f = \mathbf{B}\mathbf{V}_x\mathbf{B}^T.$$

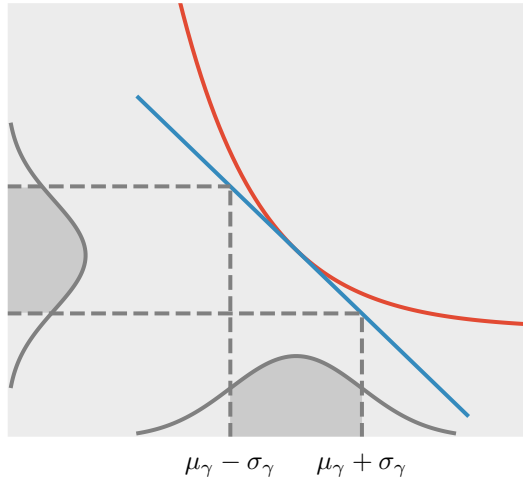
In the non-linear case the function  $f$  is usually linearized using its truncated Taylor approximation to

$$f \approx f(\mu_x) + \mathbf{J}_f(\mathbf{x} - \mu_x),$$

where  $K$  is some constant and  $\mathbf{J}_f$  is the Jacobian matrix of  $f$ . The covariance of  $f$  can then again be written as a matrix operation

$$V_f \approx \mathbf{J}_f \mathbf{V}_x \mathbf{J}_f^T.$$

This approximation works reasonably well in most cases. It can be used whenever the truncated Taylor expansion is close to the true value of  $f$  within the region of uncertainty of  $\mathbf{x}$  [21].



**Figure 5.8:** This figure, recreated from figure 2 in [21], shows the possible bias introduced by standard uncertainty propagation.

When propagating uncertainties of fitted spectral parameters for energy distributions in astrophysics, this assumption does not always hold. Even if the assumption of Gaussian uncertainties on the input  $\mathbf{x}$  holds, the energy spectra under scrutiny are highly non-linear and the linear Taylor approximation is biased. Figure 5.8 shows a power-law  $E^{-\gamma}$  for fixed energy and different values of the index  $\gamma$  on the x-axis. The figure depicts the bias that can be introduced when approximating a power-law with a linear Taylor series. Energy spectra in astrophysics often have high curvature with respect to the fitted parameters. Here a power-law with spectral index  $\gamma$  is shown together with its first-order

Taylor expansion. In this case the uncertainty of  $\gamma$ , as indicated by the distribution on the x-axis, is biased. The true distribution of  $E^{-\gamma}$  is shifted and non-symmetric. The error bands for the spectrum plots shown here and in our paper are build by sampling a multivariate Gaussian with the covariance estimated by the fitting procedure  $\mathbf{V}_x$ . The model is then evaluated for each sampled parameter combination. At each evaluated energy, the percentiles are noted and the area in-between is marked. This way non-symmetric error estimation can be performed.



# 6 Conclusion

Supernova remnants like the Crab Nebula are the driving forces of cosmic-ray acceleration in our own galaxy. These objects emit electromagnetic radiation in a vast range of wavelengths. Observation from radio, infrared and optical to X-ray and very-high-energy gamma ray are combined into the objects' characteristic spectral energy distribution. Most of the observed light is due to synchrotron emission as discussed in section 2.2. Cherenkov telescopes, however, observe the inverse Compton emission of these sources. A log-parabolic energy distribution emerges from simple stochastic arguments as shown in section 2.4. This is compatible with the spectral shape of the inverse Compton emission in SNR as measured by IACTs shown in figure 2.3. In order to build a spectral model for the Crab Nebula I assumed a single electron spectrum and homogenous magnetic field strength. I modeled the synchrotron, inverse Compton, and Synchrotron Self-Compton emission of the Crab Nebula using the open-source `na.ima` software. Figure 4.1 shows that this simple model fits the X-ray and gamma-ray data well. The resulting magnetic field in the Crab Nebula was estimated to be  $B = 107.8^{+1.6}_{-1.5} \mu\text{G}$ .

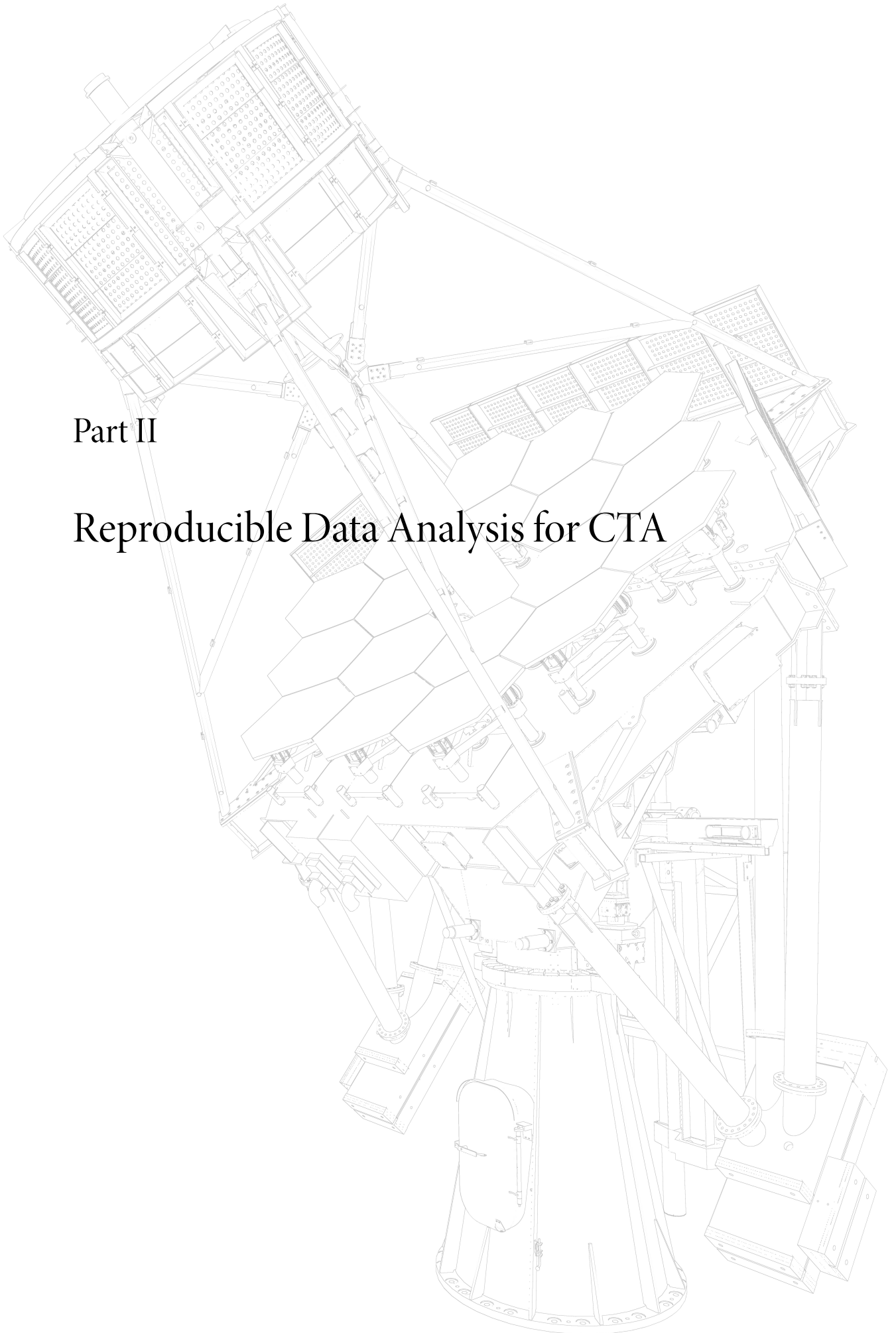
Open software and common data formats are the key ingredients to reproducible science. In a joint undertaking between the FACT, MAGIC, VERITAS, and H.E.S.S. telescope collaborations, Crab Nebula data was made public in a common data format. This open dataset contains the event lists and instrument response functions for observations of the Crab Nebula. In section 5.6 I fitted a log-parabolic spectral model to the observations in this open dataset. The statistical model that describes the measurement process assumes Poissonian distributions for the events in both the signal and background region of the sky. The expected values for the distribution have to be calculated from the telescope's instrument response function. The full model, taking into account all nuisance parameters, was given in equation (5.11). The posterior probability of the model is then sampled using Hamilton Markov chain techniques. In order to use Hamiltonian sampling, the gradient of the posterior with respect to the free parameters needs to be calculated. This includes the integral term over the energy spectrum and the application of the instrument response. The analytical solution to the integral of the log-parabolic spectrum as given in equation (5.23) suffers from numerical problems. It is possible to circumvent these numerical difficulties by approximating the

## 6 Conclusion

integral with the trapezoidal rule. I then use Theano's automatic differentiation techniques to find the gradient for the posterior. Hamiltonian sampling is performed by the PyMC3, which allows me to sample the model several hundred times per second and CPU core. The resulting spectra are compared to the full SSC model in figure 5.5. Section 5.7 demonstrates how the same sampling techniques and the same model assumptions can be used to to unfold the flux points for each individual telescope.

In the future the PyMC3 project will change its backend from the now defunct Theano library to *TensorFlow Probability* [56]. This might allow for even quicker sampling, which becomes important when thousands of distinct observations are being considered. Once the Cherenkov Telescope Array is deployed, this will become a relevant use case.

CTA will be the first experiment in high-energy physics that operates as a public observatory. The large user community requires an open and programming-language agnostic format definition. While the small data sample is a great test bench for new software and algorithms, more observational data must be made public in the future.



Part II

## Reproducible Data Analysis for CTA



# 7 Introduction

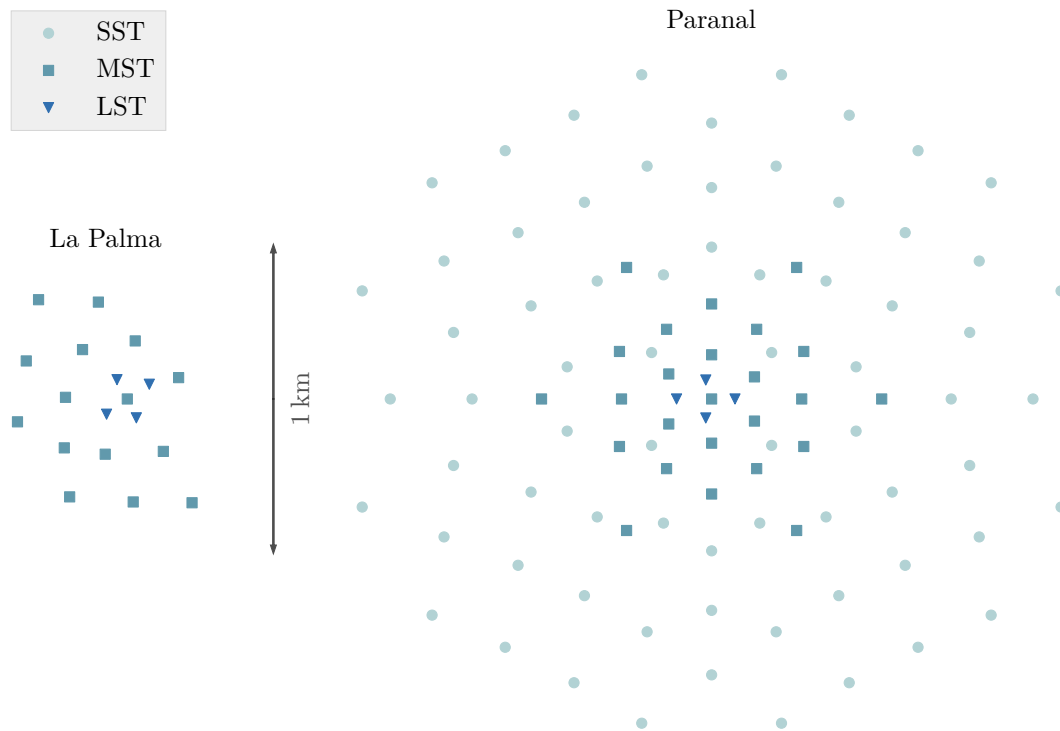
The Cherenkov Telescope Array will be the most sophisticated experiment in the field of ground-based gamma-ray astronomy to date. The book “Science with the Cherenkov Telescope Array” published in 2019 [134] gives an exhaustive review of the scientific opportunities that arise with the CTA facility. Once completed, CTA will be able to map the gamma-ray sky in an energy range from  $\sim 20$  GeV to at least  $\sim 300$  TeV. The unprecedented energy range will open a new window onto the very-high-energy sky. The lowest energies are invaluable to study distant extra-galactic sources of gamma rays. CTA’s sensitivity in the GeV range avoids the absorption of distant sources by interaction with the extra-galactic background light. The largest energies will allow CTA to search for the extreme accelerators in our own galaxy which boost cosmic rays into the PeV energy range. CTA will be operated as a public observatory that serves the entire astronomy community. This is a new development for the astroparticle community that opens up the possibilities for coordinated multi-wavelengths campaigns and exposure to the general public. Previously both observation schedules and observed data were only accessible to collaboration members.

Observation of serendipitous transient events and response to multi-messenger alerts necessitates a wide field of view combined with a small angular resolution. The need for timely follow-up observations in the era of multi-messenger astronomy motivated the stringent requirements for the short timescale capabilities of CTA. The telescopes in the array are designed to reposition themselves to any point in the sky within 90 seconds at most. In response to the recent observation of gravitational waves with the LIGO and VIRGO detectors [5], the slewing times will be increased even further. The study of active galactic nuclei (AGN) will benefit immensely from the improved sensitivity in the lower energy range and the fast slewing times. The observation of these distant highly variable objects can lead to a better understanding of black hole and jet dynamics. A full survey of the galactic plane and the Large Magellanic Clouds with CTA is planned. The quality of these surveys is limited by available observation time and angular resolution. CTA’s improved angular resolution compared to current instruments allows for better source separation in the densely populated Cygnus region in the center of the Milky Way. Other science goals probe the frontiers of current physics. CTA will search for Dark Matter, Axions, and quantum gravity effects. To cover such a large span

of observable energy ranges, CTA will host telescopes of three different sizes. The Large Size Telescopes (LSTs) are, as the name suggests, the largest CTA telescopes. The LSTs have a mirror diameter of 23m and a field of view of  $4.5^\circ$ . They are sensitive to the lowest part of the observed energy spectrum down to several GeV. The first LST was built and officially inaugurated in October 2018. It is currently in its commissioning phase and its mechanical structure and camera are being tested. Shower images have been successfully recorded by the LST camera. The Medium Size Telescopes (MSTs) have a diameter of 10 m to 12 m and a field of view (FoV) of  $6^\circ$  to  $8^\circ$ . Their highest sensitivity lies in the energy range of approximately 100 GeV to 1 TeV. Two prototypes for the MST are under investigation at the moment. One is based in Adlershof near Berlin, where its mechanical structure is being overhauled. The MST-SCT dual-mirror prototype is located at the Fred Lawrence Whipple Observatory in Arizona and recently received funding to construct a first iteration of its 11 000 pixel camera. The Small Size Telescopes (SSTs) are built to observe the highest energies. They have large FoV of about  $10^\circ$  to image the brightest showers in the highest energy range above 1 TeV. Three SST prototypes have been built so far, two of which are already recording shower images. The SST-1M telescope in the city of Kraków in Poland, the ASTRI telescope on the slopes of Mount Etna in Sicily, and the SST-GCT just outside of Paris.

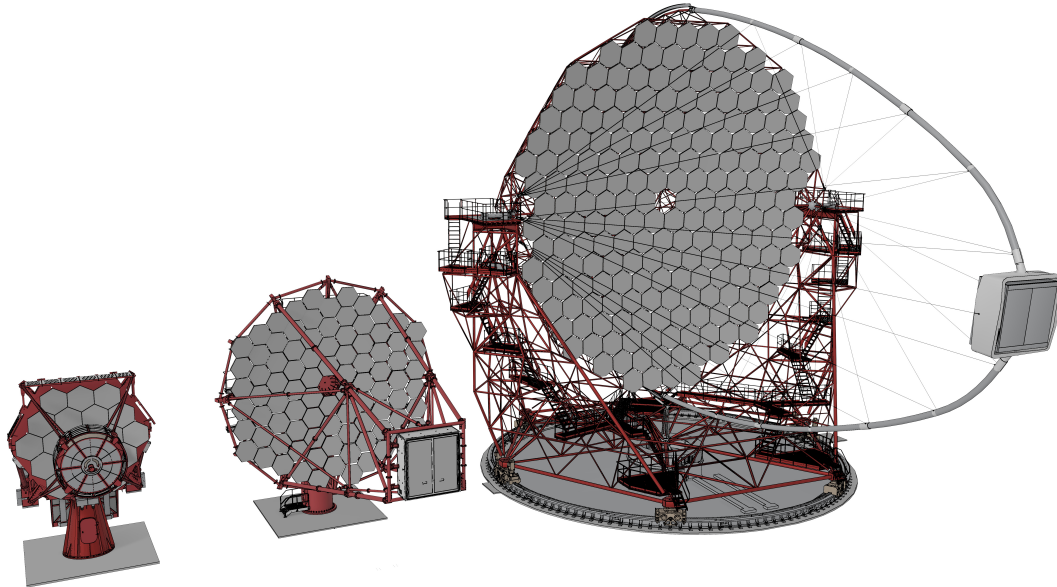
The final CTA telescopes will be built at two observation sites, one in the northern hemisphere on the island of La Palma and one in Chile's Atacama dessert at the Paranal observatory. The array in the northern hemisphere will observe extra-galactic sources and some local pulsar wind nebulae. The southern array will concentrate on galactic sources and features a larger field of view for a chance to catch rare galactic supernova explosions. Due to the nature of the ubiquitous power-law spectra, large collection areas are needed to catch the rare high-energy events above 10 TeV. This is why CTA's southern array layout covers about  $4.5 \text{ km}^2$  and consists of 50 to 70 small SST type telescopes. The northern array is made up of 4 large LSTs and 10 to 15 medium-sized MST type telescopes. Figure 7.1 shows the planned layouts of the northern and southern site. Figure 7.2 shows renderings of three CTA telescope prototypes.

So far, the data for the Cherenkov Telescope Array only exists in simulated form. The goal of the coming chapters will be to create a fully configurable and reproducible pipeline for CTA data analysis that matched the physics performance of previous CTA reference analyses. This new analysis is based on the official CTA pipeline prototype `ctapipe`, a Python-based program which is developed in a public repository. Chapter 8 lists all preprocessing steps that are needed to be performed on the raw, simulated, CTA data. In chapter 9, the machine-learning methods for background suppression and energy estimation are explained. The chapter also introduces the `aict-tools` software, which was developed to facilitate config-



**Figure 7.1:** Planned array layout of the Paranal and La Palma sites. This is the layout used for the simulations throughout this document. The CTA internal designation for this layout is HB9. The southern array consists of a total of 99 telescopes of which 70 are SST, 25 are MST, and 4 are LST type telescopes. The northern array on La Palma is made up of 15 MSTs and 4 LSTs. The array layout for the northern site is more irregular as it is located on a mountain top and has to adapt to the geological features of the surface. The Paranal site in Chile, on the other hand, is relatively flat. The layouts were optimized using large-scale simulations described in a paper by Tarek Hassan and others [75].

urable machine learning for IACTs. A fully working prototype for a CTA real-time analysis is introduced in chapter 10. It is capable of processing tens of thousands air shower events per second and can easily handle CTA’s full data rate under real-time demands. In chapter 11 a full sensitivity curve for my CTA analysis is shown and compared to CTA’s reference curve. Finally, chapter 12 contains a few remarks on the reproducibility of these results.



**Figure 7.2:** Renderings of an LST, SST, and MST telescope. The LST on the right side has about four times the mirror area of the MST on the left. A prototype of the MST structure was built in Berlin Adlershof where it currently undergoes mechanical testing. Of the three SST prototypes, the ASTRI model will most likely be the one deployed for CTA. The ASTRI telescope is a two-mirror design that allows for a very wide field of view without introducing strong aberration effects. All telescopes use active-mirror-control technology that continuously adjusts each individual mirror facet. This helps to compensate for the deformation of the telescope’s structure while sources in the sky are being tracked. The LST structure is mounted on bogies running on a flat track of rail with a diameter of approximately 24 m. It weighs a total of 103 tons, of which the camera alone makes up just under 2 tons [46]. Its slender-looking design is made possible by the use of carbon fiber composite materials. This image was rendered using Blender 2.8 [33] with models provided by G. Pérez Diaz of the Instituto de Astrofísica de Canarias.



# 8 Processing CTA Data

The physics performance of the CTA project can only be gauged from simulated data. Simulations for Cherenkov telescopes as described in section 5.2 track single particles and their secondary products through the atmosphere. Each single Cherenkov photon produced in an air shower is propagated to the virtual camera using ray tracing. For a project as large as CTA, with its vast collection area on the ground, this becomes especially challenging. The distribution of Cherenkov photons on the ground are too large to store on disk. Instead, the data is piped directly into the simulation of the telescope's optics and electronics. With a total of 99 telescopes on the ground at the Paranal site, this takes up large parts of the entire CTA computing infrastructure. Air-shower simulations for CTA, as for every IACT, are performed by the CORSIKA software [79]. The detector simulation is computed by a program called `sim_telarray`. This software is maintained by Konrad Bernlöhner and was previously used to simulate data for the HEGRA and H.E.S.S. telescopes [31]. So far, CTA data for the final layouts only exists in simulated form. However, the development of analysis software already is well underway. The result of the simulations resemble the data from real telescopes as much as possible. The simulated data essentially produces uncalibrated raw data. The task of the data processing for CTA is to read the raw data and reconstruct information about the primary particles for each air shower. Early CTA analysis was performed by the `Eventdisplay` [107] and `MARS` [117] programs. These ROOT and C++ based projects have originally been developed for the VERITAS and MAGIC projects. Both packages are proprietary to their respective collaborations and were used in the early stages of CTA's development. The new software, `ctapipe`, is a Python project developed under open-source licenses. The intention behind the design of `ctapipe` is to create a fully configurable analysis pipeline for all CTA telescopes that adheres to all provenance requirements set by the CTA consortium. As of now, `ctapipe` can perform preprocessing of simulated data, noise removal, extraction of image features, and reconstruction of the air shower's direction, as I will describe in the coming sections. Section 8.1 gives details about the `ctapipe`-based preprocessing pipeline and the datasets used throughout the next chapters. Details on the feature extraction methods, the so-called Hillas analysis, are given in section 8.3. Section 8.4 elaborates on the direction reconstruc-

tion implementation in `ctapipe`. Both background suppression and energy estimation are performed by machine-learning methods which will be the topic of chapter 9.

### 8.1 Preprocessing Pipeline and Simulated Datasets

The `ctapipe` project in version 0.6.2 does not yet provide a fully featured and configurable application to process simulated raw data. While `ctapipe` comes with basic tools to build configurable command line applications, this part of the code was only recently overhauled and is still in an early test phase. In `ctapipe` a *container* metaphor is used to encapsulate data. Rudimentary support for input and output of these container types is already available. Using these features for large productions is, unfortunately, still problematic at this point. This is mostly due to the fact that no common file standards and formats have been agreed on in the CTA collaboration. The `ctapipe` preprocessing pipeline reduces the raw data to a tabular structure which can be used to train machine-learning models for energy estimation and background suppression. This tabular structure is called data level 2 (DL2) in the CTA vernacular. The predictions of the trained models are then appended as new columns to the rest of the DL2 data. Consequently, one of the requirements for a DL2 storage format is the capability to efficiently add, select, and remove columns. The DL2 data is also the input for the computation of the instrument response functions as described previously in section 5.2, and therefore has to contain meta information about the air-shower simulations. The search for an official DL2 data format comes down to a tradeoff between columnar and row-wise storage. In machine-learning use cases, the data is often queried and operated on in a column-wise manner. Hence, a storage format is preferable that only reads the selected columns into memory. On the other hand, the processing of raw data happens event-wise. Therefore, appending rows to existing files has to be as efficient as possible. The file standard currently under consideration for CTA is HDF5, which supports both storage modes. The HDF5 standard provides hierarchical binary storage with built-in compression capabilities [135]. The `ctapipe` solution writes its data in row-wise form. This might prove to be useful in the future once data needs to be written under real-time constraints. For the analysis of simulated data, however, the runtime performance of the append operation is of lesser importance. For my use case, analysis of simulated data, I chose column-wise storage which allows for memory-efficient machine learning on a typical desktop computer. This format allows me to store single telescope information, array-wide event information and simulation settings within a single file. I use the methods implemented in `ctapipe` to build

a custom preprocessing pipeline which can be configured using `yaml` [28] files. The code and example configurations for the preprocessing pipeline can be found online at

[https://github.com/tudo-astroparticlephysics/cta\\_preprocessing](https://github.com/tudo-astroparticlephysics/cta_preprocessing) 

The list below gives an overview of the steps performed by the pipeline.

### Raw Data Calibration

Sensor artifacts and electronic noise is removed from the raw voltage curves with the help of calibration data. In the simulated data only a rudimentary calibration is performed.

### Integration

The calibrated voltage curves are integrated below their peak to find the estimated number of photons that hit the camera's pixel. The location of the peak is used as an estimator for the mean arrival time of the Cherenkov photons.

### Image Cleaning

The group of pixels which have been hit by Cherenkov light are retained, while others are discarded. An example of applied image cleaning can be seen in figure 8.1.

### Image Parametrization

The cleaned image is reduced to a list of descriptive features based on the shape of the Hillas ellipse as described in section 8.3.

### Shower Reconstruction

The point of origin and trajectory of the shower are reconstructed by using images from multiple telescopes at once. Section 8.4 explains how this stereoscopic information is used by the implemented algorithms.

### Output

The final results of the pipeline are written to disk. The final output contains telescope-wise as well as event-wise information. In order to support the computation of instrument responses, meta data about the simulation settings are stored per simulation run.

CTA's air-shower simulations are performed by the CORSIKA software. Its development has been ongoing since the late 1980s and was originally designed to simulate hadronic interactions in the atmosphere above the KASCADE cosmic-ray experiment in Karlsruhe [32]. The propagation of the Cherenkov photons to the ground is usually performed with the IACT/ATMO extension to CORSIKA. The extension follows the production of charged particles in the atmosphere and calculates the emitted Cherenkov light in each track segment. The extension was mainly written by Konrad Bernlöhner and is still maintained by him. CTA's detector simulation is performed by the `sim_telarray` [31] program. The history of `sim_telarray`

started with the HEGRA array and is still used today by the H.E.S.S. telescopes. The crucial part of any detector simulation is the realistic behavior of the telescope's trigger. A biased simulation will lead to instrument response functions which do not correspond to the actual behavior of the telescope. In particular the calculation of the effective area, as seen in section 5.2, is sensitive to errors in the simulations. The detector simulation uses the trajectory of each single Cherenkov photon as computed by CORSIKA and traces their path through the optical components of the telescope such as mirrors and light-guides. Once the photons have been ray traced to the virtual camera pixel, the detector's sensor electronics are simulated. Additionally, potential background light sources such as scattered light in the atmosphere have to be taken into account. Then the trigger logic is applied and the data from the CORSIKA is either written to disk or discarded. This is the point in the analysis that takes up a majority of the computing time. The calculation of the shower propagation is a slow process and most of the simulated showers are discarded. Research is ongoing into adapting CORSIKA by allowing it to stop early in the propagation process [26]. These approaches try to predict whether the telescope's trigger logic will discard the shower or not. Both CORSIKA and `sim_telarray` are long-running software projects with decades of history. Large parts of these legacy code bases have become unmaintainable and derelict over the recent years. Efforts are ongoing to modernize both projects by rewriting or replacing them. The roadmap to a new and modern air-shower simulation software has already been formalized in the CORSIKA 8 white paper [59].

To gauge CTA's physics performance on observed data, both signal and background data has to be simulated. Large amounts of showers have to be simulated to test the preprocessing pipeline and benchmark the quality of the reconstruction and machine-learning algorithms. As mentioned in section 3.2 the majority of triggered air showers are induced by the hadronic component, protons and heavier nuclei, of the cosmic rays. Another background component in the low GeV energy range comes from cosmic electrons. Both particle types are simulated separately. While air showers that were induced by protons can be separated from gamma-ray showers by the shape of the shower, electrons create electromagnetic cascades indistinguishable from gamma-ray showers. The incoming primary particles can be instantiated in two different ways. Diffuse simulations scatter the origin of the primary particle on the sky. The protons and electrons are simulated in a diffuse manner, so that their point of origin is uniformly distributed across the field of view. This is in contrast to point-like simulations, where the point of origin is fixed in the center of the field of view. Table 8.1 shows the simulated datasets together with their associated simulation settings. These datasets will be used throughout the rest of the document. I only consider data simulated for the southern

**Table 8.1:** Datasets used for the CTA analysis. The table below lists information about the number of simulated showers, as well as the number of remaining events after the telescope’s trigger simulation and preprocessing have been applied. During processing events get dropped if the image cleaning does not select any pixels or the image parametrization fails for numerical reasons. In this dataset all telescopes point in the same direction, due south with an elevation of 70°. In the point-like gamma-ray simulation the virtual source is situated right in the center of the array’s field of view. I exclusively use data from the southern array in my analysis. All 99 telescopes are participating in the trigger. I selected the simulated *LSTCam*, *NectarCam*, and *DigiCam* prototype hardware for these datasets.

### Paranal Array HB9

	<b>Gamma</b>	<b>Diffuse Gamma</b>	<b>Proton</b>	<b>Electron</b>
Energy (TeV)	0.003 – 330.0	0.003 – 330.0	0.004 – 600.0	0.003 – 330.0
Mean Multiplicity	5.4	4.5	4.7	4.7
Pointing Alt/Az (°)	70, 180	70, 180	70, 180	70, 180
Processed Events	1 245 513	1 858 688	4 326 085	594 198
Scatter Radius (m)	2500	3000	3000	3000
Simulated Showers	32 440 000	201 480 000	927 300 000	70 900 000
Spectral Index	-2.0	-2.0	-2.0	-2.0
View Cone (°)	0	10	10	10
CORSIKA Runs	1622	10 074	9273	3545

site for the analysis. CTA’s detector simulation places multiple virtual camera prototypes into a single telescope. I selected the *LSTCam*, *NectarCam*, and *DigiCam* prototype hardware for my analysis in accordance with previous reference analyses.

## 8.2 Raw Data Processing

The data recorded by imaging Cherenkov telescopes is contaminated with noise and sensor artifacts. Some of that noise originates in background light due to stars or other diffuse light hitting the mirror. Other noise is produced by the sensor itself or by the camera electronics. The raw data from the pixel sensors, be it silicon based photo-multipliers or traditional photo-multiplier tubes, consists of a series of voltages over time. Data from the sensors is only transferred when a group of pixels in the camera reaches a certain voltage threshold. In addition to these single telescope triggers, CTA uses a stereoscopic trigger system. Only when two or more telescopes have triggered coincidentally, the data will be stored permanently. The collection of telescope data corresponding to one coincident trigger is called an *array-event* throughout this chapter. The part of an array-event belonging to one distinct telescope

**Table 8.2:** Cleaning levels used for the CTA analysis. I selected the simulated LSTCam, NectarCam, and DigiCam prototype hardware for the datasets. In the first sweep the tail-cut cleaning method selects all pixels above the “core threshold”. In the second step, all pixels are added whose light content is above the “neighbor threshold” and which have at least “min pixels” neighboring pixels that were selected in the first step.

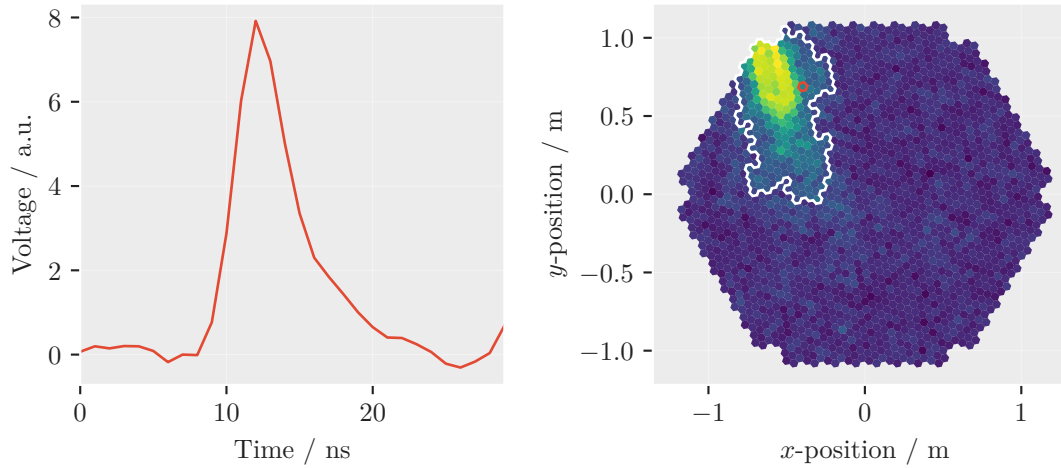
	LSTCam	NectarCam	DigiCam
Neighbor Threshold	3.5	3.0	2.0
Core Threshold	7.5	5.5	4.5
Min Pixel	2.0	2.0	2.0

is called a *telescope-event*. Converting the time series of voltages in each pixel to images is the first step of the `ctapipe` pipeline. This step is sometimes called signal, or image, extraction.

For the standard CTA analysis, two numbers are extracted per pixel from the voltage curves. The number of recorded photons are estimated by integrating over the length of an adaptively selected time window. The result is then multiplied by the gain factor of the corresponding pixel which is known from calibration measurements. The mean arrival time of photons per pixel is estimated by finding the rising edge of the signal. Smoothing methods are usually applied to reduce the influence of electronic noise in the signal. This step is highly dependent on the sensor technology and circuitry used in the camera. The observed data will have to be carefully calibrated to compensate for any environmental effects like temperature and humidity. CTA simulations make the rather optimistic assumption that all telescope are well cross-calibrated.

The resulting images still contain noise due to background light. Only those pixels which have been hit by Cherenkov light are of consequence for the analysis. Others are discarded. At first, pixels above a certain threshold are selected as *core* pixels. In a second pass adjacent pixels above a second, lower, threshold are selected. This process is known as *tail-cut* cleaning. The set of selected pixels create an optical image of the air shower. These selected pixels can then be used to extract the geometrical properties of the air shower. The cleaning is a crucial step in the analysis. There is a trade-off between the goal to retain as many Cherenkov photons from the air shower as possible while discarding noisy pixels which can bias the reconstruction of the shower’s properties. The cleaning levels chosen for the preprocessing are listed in [table 8.2](#)

[Figure 8.1](#) shows a simulated gamma-ray event as seen by an LST type telescope. The left side shows the voltage curve of a single pixel which is part of the air-shower image as can be seen in the right image. The right side shows each pixel of the camera with the estimated number



**Figure 8.1:** A simulated air shower induced by a gamma ray as seen by one of the four LST cameras to be built at the Paranal observatory. The left side of the image shows the voltage curve for a single pixel. The selected pixel lies on the edge of the shower and is marked by a red edge in the right image. The right-hand side depicts the image in the LST camera that shows the estimated number of photons on the color axis. The contour around the brightest pixels shows the group of pixels  $C$  that have been selected by the tail-cut cleaning method. This gamma ray was simulated with an energy of 1.86 TeV and triggered a total of 17 telescopes in the array.

of photons on the color axis. The pixels selected by the tail-cut method are marked by the red outline. This part of the analysis process is clearly the most data intensive task as it needs to process the lowest level of data from many telescopes at once.

### 8.3 Image Feature Extraction

Properties of the incoming primary particle can only be inferred from the air-shower's Cherenkov emission. The classical IACT analysis uses the pixels selected by the cleaning step to calculate the so-called Hillas-parameters which describe the shape of the Cherenkov emission. In his seminal conference proceeding for the International Cosmic-Ray Conference 1985, Michael Hillas [82] used air-shower simulations for the WHIPPLE telescope to find parameters which allow for the separation between air showers from gamma rays from those started by cosmic rays. This early work is noteworthy due to the fact that it led to the first observation of a TeV gamma-ray source, the Crab Nebula, and started the success story of IACT technology. Hillas proposed to approximate the shape of the air showers by an ellipse and use its geometric parameters for further analysis. In particular the *width* and *length* of the observed ellipse serve as a discriminating feature. They describe the standard deviation of the Cherenkov photon distribution along the major and minor axis of the ellipse. Early

papers described the calculation of the parameters by using rather complex equations gained from analytical least-squares fitting of the ellipse's major axis and then rotating the coordinates before calculating the standard deviations along the axes. One example can be found in appendix of the paper describing the first successful observation of the Crab Nebula by the WHIPPLE telescope [143]. A simpler, and quicker, calculation of the Hillas ellipse can be performed by diagonalizing the covariance matrix of the photon distribution. Let  $p = (X, Y)^T$  be a two-dimensional vector of random variables  $X$  and  $Y$  describing the position of the Cherenkov photons on the camera. Then their covariance matrix is defined as

$$\mathbf{V}_p = \begin{pmatrix} \text{Var}(X) & \text{cov}(X, Y) \\ \text{cov}(X, Y) & \text{Var}(Y) \end{pmatrix}.$$

The decomposition of the covariance matrix of  $p$  yields a set of orthogonal eigenvectors. The vector associated with the largest eigenvalue points into the direction of largest variance i.e the major axis of the Hillas ellipse. The eigenvalues are the variances of the distributions along these directions. Hence, the Hillas *width* and *length* are then simply calculated as the square root of the eigenvalues. This process is also known as *Principal Component Analysis*.

The true distribution of Cherenkov photons  $p$  can be approximated from the cleaned camera image. Each pixel in the camera collects the photons in the area defined by its entry window. If the tail-cut cleaning selects the pixels which mostly contain Cherenkov photons, the resulting camera image can be interpreted as a binned measurement of the true photon distribution. The values in the cleaned image correspond to frequency weights. Given camera pixels of equal area, the covariance matrix for  $p$  can be calculated by the weighted variance of the selected pixel set  $C$

$$\text{Var}(X) \approx \frac{1}{W} \sum_{c \in C} w_c (x_c - \bar{x}_w)^2,$$

where  $\bar{x}_w$  is the weighted mean  $x$ -position,  $w_c$  is the weight of pixel  $c$ , and  $W$  is the sum of all weights in  $C$ . The weighted covariance is defined accordingly

$$\text{Cov}(X) \approx \frac{1}{W} \sum_{c \in C} w_c (x_c - \bar{x}_w)(y_c - \bar{y}_w).$$

The mean coordinate of the shower,  $(\bar{x}_w, \bar{y}_w)$ , is often called the center of gravity (cog)

$$p_{\text{cog}} = \frac{1}{W} \sum_{c \in C} w_c \begin{pmatrix} x_c \\ y_c \end{pmatrix}$$



and describes the mean of the photon distribution. The angle  $\psi$  defines the orientation of the major axis with respect to the horizontal axis of the camera. It is calculated from the first and second component of the covariance matrix's eigenvector  $\mathbf{v}$  with the largest eigenvalue

$$\psi = \arctan\left(\frac{\mathbf{v}_y}{\mathbf{v}_x}\right).$$

Note that the ambiguous definition of  $\arctan$  is used here instead of  $\arctan2$ , as no preferred direction is defined for the ellipse itself. The sum of weights  $W$  encodes another important property of the shower which is often called *size* or *intensity*. It is a proxy for the air shower's total brightness in terms of emitted Cherenkov radiation. It correlates strongly with the primary particles initial kinetic energy. Higher order moments along the shower's axis can be calculated once the eigenvectors and  $\psi$  have been calculated. The skewness of the light distribution, i.e. the third moment along the major axis, is an indicator for the travel direction of the shower. Similar information can be extracted from the arrival time of the photons in each pixel. Another feature is the *leakage* of the image. It is defined as the number of pixels, or sum of weights of pixels, which lie on the outer edge of the camera. This feature is useful to discard images which are not fully contained within the camera.

## 8.4 Geometrical Shower Reconstruction

As explained in section 5.3, the cosmic-ray background is isotropically distributed across the sky. Gamma-ray sources, either extended or point-like, can be distinguished from the cosmic-ray background only when the reconstruction of the gamma-ray direction is accurate. The better the reconstruction of the gamma ray, the higher the significance with which a source can be detected. The optical system of an IACT is focused on the upper parts of the atmosphere, where most air showers emit their Cherenkov light. In general, the effective area and energy range of an IACT is limited by its field of view and its mirror size respectively. A larger mirror helps collect more light and allows recording very dim showers. Unfortunately, a large mirror, and hence a large aperture, reduce the depth of field of the telescope and only parts of the shower are in focus. New detector types, like the *Cherenkov-Plenoscope* [119], are proposed to remedy this problem. Similar to a thin optical lens, aberration effects and distortions can negatively impact the image quality. Compared to a typical imaging telescope, the requirements for an IACTs optical system are less stringent in terms of mirror precision. For any digital imaging system, a point-like light source has to be mapped onto the area of a single pixel in the focal plane in order to produce a sharp image. As the typical pixel size

for an IACTs is in the order of centimeters, mirror precision is less crucial. However, since IACTs are completely exposed to the elements, their durability and stability is of much higher importance. In addition, these large structures which hold the mirrors and cameras of the telescope are not completely stiff. Each change in elevation angle requires a correction of the mirror alignment. All large IACTs use active mirror control to constantly align their mirrors with the camera. The geometric information of the shower can be reconstructed with an IACT due to the fact that its mirrors act much like a thin lens. The mirrors of an IACT uniquely map coordinates in the sky to coordinates in the focal plane of the telescope. As seen in the previous section, the major axis of the Hillas ellipse points along the main axis of the air shower which in turn points along the primary particles trajectory. In the image of a single telescope, the major ellipse axis does not uniquely specify a point of origin in the sky. In fact, the showers point of origin can lie anywhere on the line defined by the major axis. The reconstruction of the primary particles direction has a great impact on the telescopes sensitivity. For a single telescope, extensive shower simulations are needed to use higher order features about the image shape to determine the point of origin along the line of the major axis. This is known as the *disp*-method among Cherenkov astronomers [57].

A CTA array-event will always have the information of at least two telescopes available to describe the observed air shower. The stereoscopic view augments the Hillas parameters with additional information about the showers shape and direction. A simple stereoscopic reconstruction technique was introduced by the HEGRA experiment [49]. It works by superimposing the images of each telescope onto a common camera coordinate system. The intersection point between each pair of major ellipse axes is calculated and averaged to determine the point of origin on the sky.

During my research stay at CEA Paris, I adapted the HEGRA methods for implementation in `ctapipe`. Previous implementations in `ctapipe` used numerical minimization algorithms to find the point of intersection. In my implementation the intersection is found by linear least-squares methods. In contrast to HEGRA, CTA consists of telescopes with different focal lengths. The combination of different telescope sizes requires the transformation to a common coordinate frame relative to the local horizon. The coordinate frames and transformations in `ctapipe` were implemented by Maximilian Nöthe. They rely on `astropy`'s coordinate API, which allows for transitive transformation operations between coordinate frames. The definition of the altitude and azimuth angles in `ctapipe` follows `astropy`'s conventions in which an azimuth angle of  $0^\circ$  points due north and  $90^\circ$  points east. An altitude angle of  $0^\circ$  points parallel to the ground. The telescopes in the simulated datasets all point in the same directions with an altitude angle of  $70^\circ$  and an azimuth of  $180^\circ$ . Given the pointing

direction of a telescope in the horizontal frame, every point in the telescope's camera can be transformed into a tuple of altitude and azimuth coordinates. In analogy to the HEGRA method, the major axis of the ellipse in each telescope is transformed into the horizontal coordinate frame. This is achieved by selecting two points on the major axis in the camera. The first selected point is the center of gravity. Per definition it has to lie on the ellipse's major axis. The second point  $p_t$  is offset from the center of gravity by an arbitrary distance  $a$  in the direction  $\psi$  along the main axis

$$p_t = p_{\text{cog}} + a \begin{pmatrix} \cos(\psi) \\ \sin(\psi) \end{pmatrix}.$$

The transformed points  $p_t$  and  $p_{\text{cog}}$  together with the telescope's position on the ground define a *plane* in the euclidean space. A plane, in mathematical terms, is given by two vectors and a point of origin. Each telescope participating in the triggered event defines such a plane with an accompanying normal vector  $\mathbf{n}$ . The intersection between two, non-parallel, planes is given by a line whose direction is found by taking the cross product between the planes' normal vectors. The intersection between the planes points along the direction of the recorded air shower. Similar to the HEGRA approach, each ordered pair of normal vectors  $(\mathbf{n}_i, \mathbf{n}_j)$  is used to find an intersecting line which is then combined with a weighted sum

$$\hat{d} = \sum_{(i,j) \in S} c_i c_j (\mathbf{n}_i, \mathbf{n}_j),$$

where  $S$  contains all combinations of indices  $i$  and  $j$  for which  $i < j < N$ . For an event with  $N$  participating telescopes,  $N(N - 1)$  intersections are evaluated. In the current ctapipe implementation (version 0.6.2) the weights are calculated as  $c = W \frac{l}{w}$ , where  $W$  is the total intensity of the shower and  $w$  and  $l$  are the width and length of the Hillas ellipse. This simple heuristic puts emphasis on shower images that are bright and elongated for which the shower orientation  $\psi$  can be reconstructed accurately. Figure 8.2 shows the angular resolution of this reconstruction method for bins in simulated energy. The angular resolution is defined as the 68<sup>th</sup> percentile of the distance between estimated and simulated source position and is indicated by the blue line. The histogram in the background shows the underlying distribution of the simulated gamma rays.

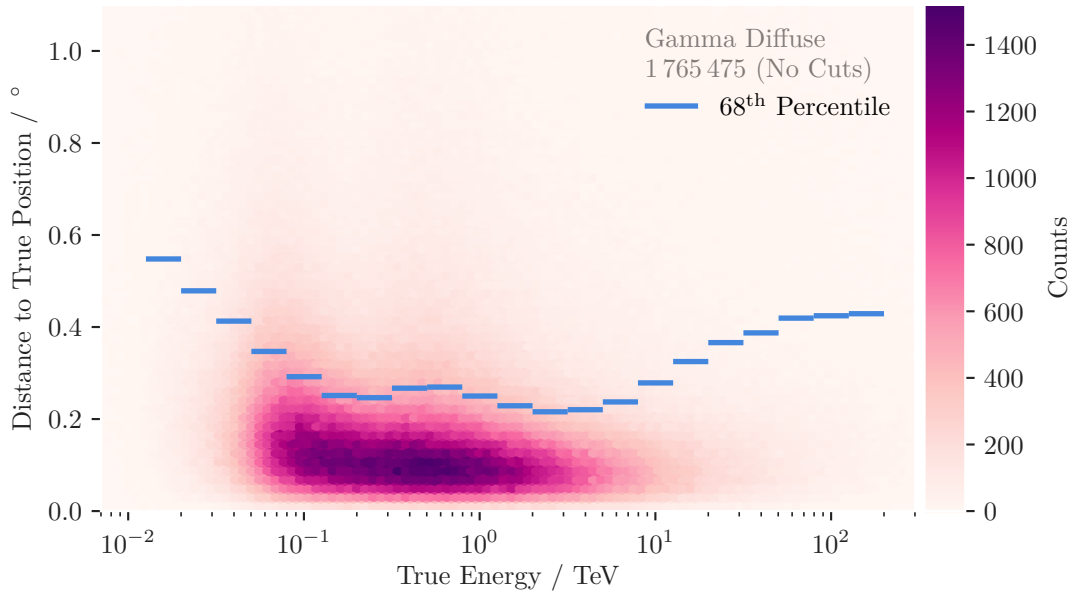
In addition to the direction of the shower, the height of the showers maximum lateral distribution  $H_{\text{max}}$  can be estimated. The  $H_{\text{max}}$  variable describes the point along the trajectory of the air shower at which the amount of produced secondaries starts to decrease. This corresponds to the point in the camera image in which the Hillas ellipse has its largest extension i.e. its

center of gravity. The  $H_{\max}$  feature correlates with the primary particles energy and type. The more energy the primary particle had, the longer the trajectory of the air shower before the all energy has dissipated. Therefore, the height above ground at which the shower was brightest can be used for energy estimation as well as background suppression. In ctapipe the estimation of  $H_{\max}$  is performed using a least-squares method. The center of gravity  $p_{\text{cog}}$  in each shower image is transformed into the local horizontal frame and then into the euclidean coordinates. This vector  $\mathbf{v}_i$  together with the telescope's position  $\mathbf{v}_{\text{tel},i}$  defines a line running from the telescope to the brightest part in the shower. In a second step, the closest point to a common intersection point between all  $N$  lines is estimated. Following arguments from [140] the point closest to all other lines can be found using matrix methods. In the first step the matrix  $\mathbf{M}_i = \mathbf{1} - \mathbf{v}_i \mathbf{v}_i^\top$  is build for all participating telescopes in the event. The closest mutual point  $p_{H_{\max}}$  to all lines is then found by calculating

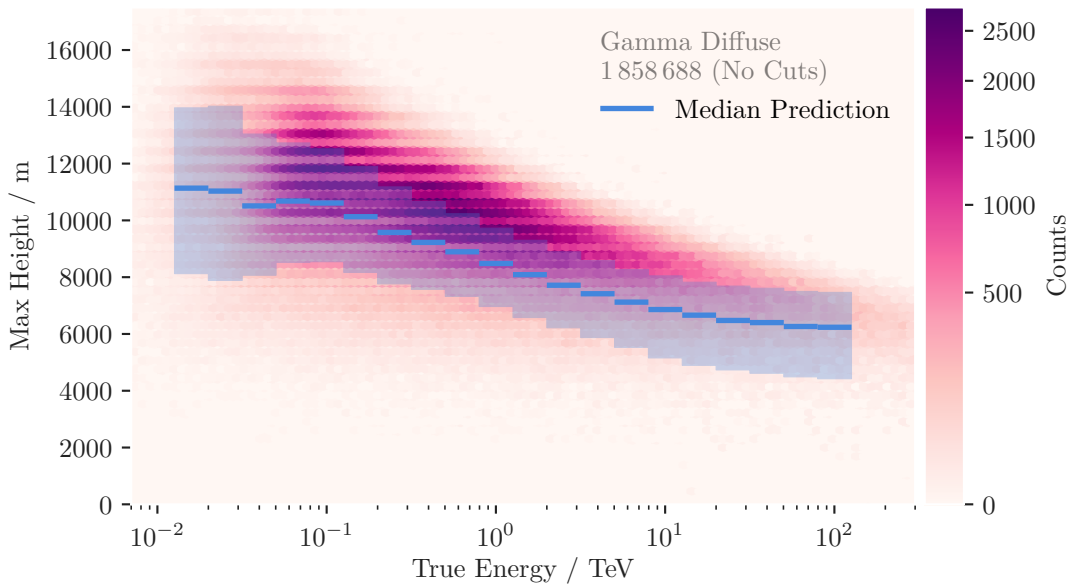
$$p_{H_{\max}} = \left( \sum_i \mathbf{M}_i \right)^{-1} \sum_i \mathbf{M}_i \mathbf{v}_{\text{tel},i}. \quad (8.1)$$

Figure 8.3 shows the estimated  $H_{\max}$  for simulated diffuse gamma rays together with the true height. While the estimator is clearly biased and systematically underestimates the true  $H_{\max}$ , the trend of decreasing  $H_{\max}$  with increasing energy is followed. Hence, it can be used as a useful feature for training the energy estimator.

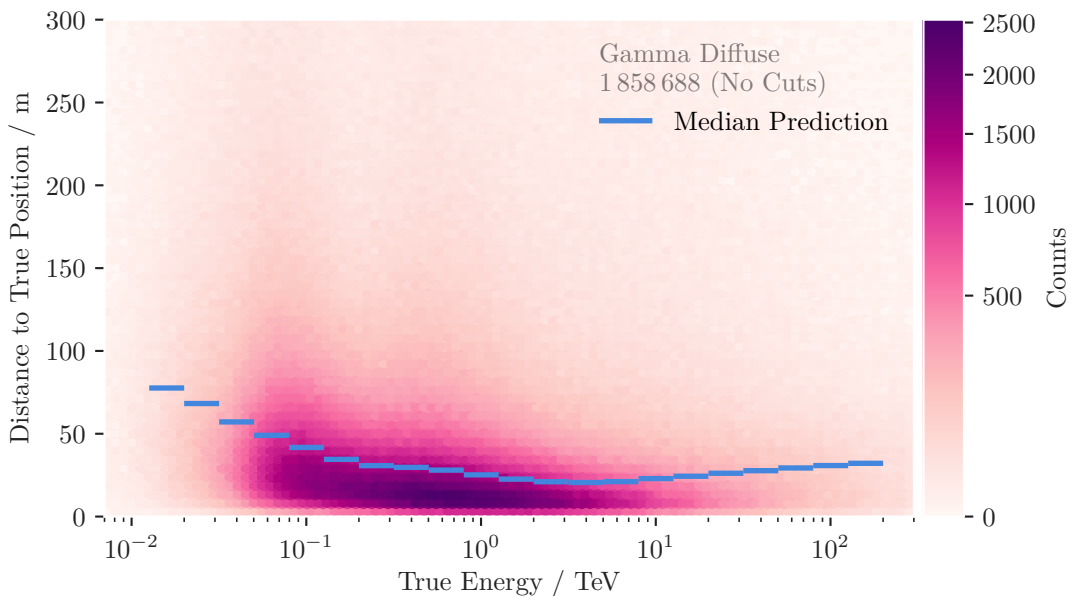
Another important shower parameter is the impact point of the shower trajectory on the ground. It is calculated using the lines defined by the reconstructed major axis of the shower in each camera. As before the closest mutual point, the closest point to a common intersection, can be found using the least square solution from equation (8.1). The distance of each telescope to the impact point is another helpful feature for energy estimation. Showers which trigger the camera despite large distances, have to emit a lot of light and therefore have a large energy content.



**Figure 8.2:** Angular resolution for diffuse gamma rays. The blue line shows the 68<sup>th</sup> percentile of the distance between estimated and simulated source position in each energy bin. The hexagonal histogram in the background shows the event distribution. From the minimum energy up to approximately 2 TeV the angular resolution improves. This is expected behavior as the reconstruction of the major axis orientation  $\psi$  improves with increasing image brightness. At even higher gamma-ray energy the shower images are often not fully contained within the telescope’s field of view and cannot be reconstructed properly. The distribution of events is clearly skewed towards large distances. The peak of the distribution is located near  $0.1^\circ$ . For most events, the direction is reconstructed accurately. Many of these outliers would be removed for the actual analysis, whose event selection is optimized for a specific use case. An example will be shown in chapter 11. It is important to note that the distance is calculated between points given in a horizontal coordinate frame which is spherical. The distance between two points on a sphere can be calculated using the Vincenty formula [142]. For this plot, and similar plots following in the coming sections, the `astropy` implementation of spherical distance computation is used. The dataset shown in the figure contains 1 858 688 array-events with a total of 8 171 917 single telescope-events.



**Figure 8.3:** The blue line shows the estimated height above ground at which the lateral distribution of Cherenkov photons is the largest. The shaded blue area indicates the 16<sup>th</sup> and 84<sup>th</sup> percentile of the estimation. The distribution in the background shows the simulated  $H_{\max}$  values. The horizontal lines in the  $H_{\max}$  distribution are due to binning effects in CTA's air-shower simulation. The estimator follows the same shape as the  $H_{\max}$  distribution, making it a good feature for energy estimation and background suppression.



**Figure 8.4:** The two-dimensional histogram shows the distribution of the distance between the true impact position and the estimated impact position of the shower on the ground for bins in simulated energy. The blue lines indicate the median of the distribution in each bin.

# 9 Machine Learning

Machine learning for this CTA analysis is used for energy estimation of the primary gamma ray and background suppression. As mentioned in section 3.2, cosmic-ray showers, i.e. hadronic showers, build the main background component in IACT data. A model which can separate air-showers induced by gamma rays from those started by hadrons, boosts the detection significance of an IACT considerably. Energy estimation is necessary to learn about a sources energy spectrum and the acceleration mechanisms taking place in it. From simulated data, where the true values for the particles type and energy are available, a model can be built which takes image parameters as input and predicts the output values on new, unlabeled, observations.

In the early days of IACT analysis, these model functions weere often hand-crafted. For background suppression a set of *cuts* were defined in the image parameter space. By looking at the distribution of the hillas parameters for simulated gamma rays and protons, the location of a cut can be fixed and then applied to observed data. The HEGRA analysis from 2004 [15], for example, used a simple cut in the width of the Hillas ellipse to reduce the amount of background events in their data. Energy estimation for HEGRA used a similar approach. Correlation plots between image parameters and true energy were used to find the shape of analytical function whose parameters  $\mathbf{p}$  were fitted. These approaches are flawed. The choice of parameters in which to place the cuts and the choice of analytical functions for energy estimation is utterly subjective. Not to mention the fact that the capacity of human brains to work in many dimensions is limited. These shortcomings can be overcome by supervised machine-learning algorithms.

Here I mostly follow the notations and definitions from the book “Elements of statistical Learning” by Trevor Hastie and others [76]. Given a set of  $N$  observations with  $p$  variables the matrix  $\mathbf{X}$  contains the full information needed to train a machine-learning model. In this so-called *data matrix* each column corresponds to an observed variable. Each row maps to a specific observation. The columns in  $\mathbf{X}$  are called differently depending on context and personal preference. Common choices are observables, variables or features. One specific observation  $x \in \mathbf{X}$  is written as a small letter. The  $j^{\text{th}}$  entry, or feature, in one observation is

indexed by  $x^{(j)}$ . In a similar manner, a single uppercase letter such as  $X$  denotes a generic random variable. The  $i^{\text{th}}$  observed value of  $X$  is written as  $x_i$ . Vectors containing entries for each of the  $N$  observations are set in bold lowercase  $\mathbf{x}$ . In this specific use case, the columns in  $\mathbf{X}$  map to the image parameters and the rows to the observed air showers. If  $X_w$  is the random variable describing underlying the distribution of observed Hillas widths, then  $x_i^{(w)}$  is the Hillas width from one observed air shower in one telescope and  $\mathbf{x}^{(w)}$  is the vector containing all observed Hillas widths. For the each observation we also have an associated *output* or *truth*  $y$  stored in the vector  $\mathbf{y}$ . The true label values in the CTA simulations come in two distinct forms. The energy of a particle  $y \in \mathbb{R}$  is a continuous parameter, whereas the type of the particle is a discrete variable that can take only one of two values  $g \in \{\text{gamma}, \text{hadron}\}$ . The labels values are encoded as numbers by convention so that  $g \in \{0, 1\}$ . For the sake of simplicity I do not always differentiate between the continuous  $y$  and the categorical  $g$  in some of the following definitions. Roughly following the definition in [76, page 10] we can now specify the task of supervised machine learning as follows:

Given a  $N \times p$  matrix  $\mathbf{X}$  and some associated output vector  $\mathbf{y} \in \mathbb{R}^N$ , find the function  $f(\mathbf{X}) = \hat{\mathbf{y}}$  that takes a vector  $\mathbf{X} \in \mathbb{R}^p$  and returns a prediction for  $\mathbf{y}$  and minimizes some “loss function”  $L(\mathbf{y}, f(\mathbf{X}))$  over the data and the true label.

If  $\mathbf{y}$  is continuous, then  $f$  is called a *regressor*. If  $\mathbf{y}$  is categorical, then  $f$  is called a *classifier*. In this formulation supervised machine learning can be understood as a global optimization problem. The space of possible solutions has to be constrained to find valid solutions with any predictive power. Usually, the shape of the function  $f$  is fixed for a machine-learning algorithm. It is parameterized in some form by a vector  $\beta$  whose entries are found by minimization with respect to the loss. This step is often called *training* the classifier or regressor.

The textbook example for a choice of  $f$  and  $L$  is the least-squares regressor [76, page 11] which exemplifies some of the important properties of supervised machine-learning methods. Assuming  $f$  is a linear function in the data  $\mathbf{X}$  then it can be written as

$$f(\mathbf{X}; \beta) = \hat{\mathbf{y}} = \beta_0 + \sum_{j=1}^p \mathbf{x}^{(j)} \beta_j.$$

The loss function for the least-squares method is the residual sum of squares which can be written as

$$L(\mathbf{y}, f(\mathbf{X}), \beta) = \sum_{i=1}^N (y_i - f(x_i; \beta))^2 = \sum_{i=1}^N (y_i - x_i \beta)^2,$$



where the data matrix was implicitly modified to contain series of ones  $(1, 1, \dots, 1)$  as the first column and  $\beta$  is a column vector  $\beta = (\beta_0, \beta_1, \dots, \beta_p)^\top$ . From this expression an analytical solution for  $\hat{\beta} = \arg \min_{\beta} L(\mathbf{y}, f(\mathbf{X}), \beta)$  can be found by setting the derivative with respect to  $\beta$  equals zero

$$\begin{aligned} \frac{\partial L}{\partial \beta} \stackrel{!}{=} 0 &= \frac{\partial}{\partial \beta} \sum_{i=1}^N (y_i - x_i \beta)^2 \\ &= -2 \cdot \sum_{i=1}^N x_i (y_i - x_i \beta) \\ &= 2 \cdot \mathbf{X}^\top (\mathbf{y} - \mathbf{X}\beta). \end{aligned}$$

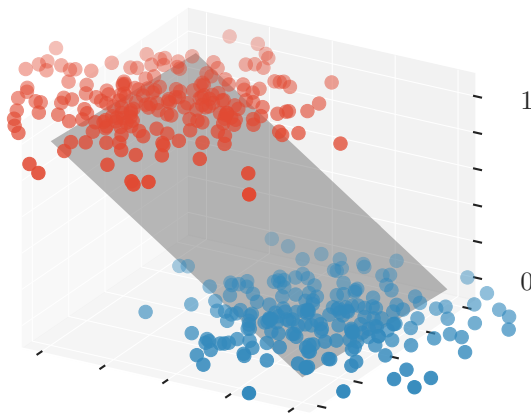
Solving for  $\beta$  yields

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.$$

The linear least-squares method essentially fits in hyperplane in the parameter space spanned by the provided data and true label  $y$ . If the values of  $\mathbf{y}$  are continuous, then  $\hat{\mathbf{y}} = f(\mathbf{X}; \hat{\beta})$  is the result of the regressor and can be used as is. In the case of classification, the fitted hyperplane, or the function  $f(\mathbf{X}; \hat{\beta})$ , can be used to define a decision boundary or threshold  $\alpha$ . The classification function is then defined as

$$\hat{g}_i = f'(x_i; \hat{\beta}) = \begin{cases} 0, & \text{if } f(x_i, \hat{\beta}) < \alpha \\ 1, & \text{else.} \end{cases}$$

In case the true label for the binary classification problem  $g$  or  $y$  was encoded using values  $\{0, 1\}$ , then the canonical value for the threshold is  $\alpha = 0.5$ . The decision boundary is hence defined by all points where  $f(\mathbf{X}; \hat{\beta}) = 0.5$ . Intuitively speaking, new observations are classified based on their location relative to the fitted hyperplane. The distance of a new observation to the decision boundary can be understood as a measure of ‘‘certainty’’ of the point belonging to either class. Figure 9.1 shows an artificial dataset on which a least-square regressor was trained. The points in the figure are colored to indicate the class to which they belong. It is also obvious from the figure that the raw output of the least-square regressor is not a proper probability density as it is neither bounded nor normalized. This makes the interpretation of the regressors output difficult to interpret. It can, however, be transformed into a value range resembling a probability density by wrapping the output of the least-squares predictor in a so-called sigmoid function  $S(f(\mathbf{X}; \hat{\beta}))$



**Figure 9.1:** The transparent plane in the figure shows the result of a least-squares fit to the data.

The family of sigmoid functions is loosely defined by their properties to be monotonically increasing and bounded by horizontal asymptotes for  $m \rightarrow \pm\infty$ . The error function shown in figure 5.4 is an example of a sigmoid function. If the sigmoid function is bounded by 0 and 1, the output of  $S(f(x_i; \hat{\beta}))$  can be interpreted as a probability density. Note that this is different from a *confidence level* for the prediction. Adjusting the classifier output to resemble a confidence is the topic of “classifier calibration” [121, 149]. Like the least-squares approach, most

classification algorithms provide a continuous output which is either bounded or can be wrapped by a sigmoid function. The adaption of the decision threshold  $\alpha$  is a key component in the analysis of IACT data. As explained in section 5.3, IACT data is heavily contaminated by cosmic-ray background. By increasing the prediction threshold the amount of background events can be reduced. This naturally comes with an inevitable decrease in selected signal events. The choice of  $\alpha$  is optimized for different physics use cases. In chapter 11 the prediction threshold will be optimized for the detection of gamma-ray point sources. The correct way to validate one’s classification model is often hotly debated within physics research groups. The validation methods I use for the CTA analysis are described in section 9.1. Section 9.2 explains the basics of decision tree algorithms on which both the background suppression and energy estimation are based. Results of the training and the application of the models to CTA data is shown in section 9.4. Section 9.3 describes the software we developed to create a reproducible machine-learning pipeline for the CTA and FACT telescopes.

## 9.1 Model Validation

Models for classification or regression can only be properly evaluated on labeled data. Some machine-learning algorithms tend to *overfit* on the training data. While the models minimize the loss  $L$  on the training data to a great extent, it does not generalize well to new data. In other words, it has weak predictive power on hold-out data. In literature this problem is known as bias-variance tradeoff, overtraining, or *overfitting*. A model fitted on some training

data  $\mathbf{X}_{\text{train}}$  with label  $\mathbf{y}_{\text{train}}$  needs to be applied to an independent dataset  $\mathbf{X}_{\text{test}}$  that comes with an associated label  $\mathbf{y}_{\text{test}}$  in order to get an estimate of its generalization capabilities.

A classifier's performance in terms of predictive power is estimated from its *confusion matrix*. In a classification problem with two categories, like the distinction of gamma-ray air showers from hadronic ones, the two classes are called “positive” and “negative” per convention. In IACT analysis the positive class is usually understood to be the gamma-ray events. The result of the model validation for a binary classification problem contains four numbers. The number of *true positives* ( $TP$ ) which are the events that have correctly been predicted to belong to the positive class, i.e. the number of correctly identified gamma rays. The amount of *false positives*, ( $FP$ ) the events that have incorrectly been predicted to belong to the positive class, i.e. the number of hadronic events that have been falsely classified as gamma-ray events. The number of *true negatives* ( $TN$ ) and *false negatives* ( $FN$ ) are defined accordingly. These four numbers, that make up the confusion matrix, completely characterize a classifier's prediction. A myriad of scalar quantities that try to summarize the classification can be derived from the entries in the confusion matrix. Popular choices include the accuracy, precision, F-score, recall or true positive rate, and the false positive rate just to name a few. Each come with their own advantages and disadvantages. The extremely common *accuracy* measure, for example, is defined as

$$\text{acc} = \frac{TP + TN}{TP + FP + TN + FN},$$

i.e. the ratio of correctly identified events with respect to the entire population. The accuracy is only useful in the case of balanced datasets where both classes are represented by an equal amount of data points. For imbalanced datasets the accuracy alone does not provide a good estimate on the classifier's “quality”. The balanced accuracy on the other hand is defined in terms of true positive and true negative rates instead of the absolute numbers

$$\text{acc}_{\text{bal}} = \frac{TPR + TNR}{2}$$

and is therefore not influenced by unbalanced class ratios. In the case of IACT analysis datasets are often imbalanced. The observed data contains orders of magnitude more background events than gamma-ray showers. The accuracy on the simulated data in itself does not help to gauge the performance of a telescope.

A popular method for calculating performance estimates is called cross-validation. A  $k$ -fold cross validation splits the training data into  $k$  independent subsets. One of the  $k$  subsets is

used as a validation set for a model that is trained on the union of the remaining  $k - 1$  subsets. During cross-validation the model is trained and evaluated  $k$  times. This yields  $k$  estimations of an algorithm's classification performance on a specific training dataset. This allows for robust error estimates on the entries in the confusion matrix. It does, however, increase the runtime of the model training by a factor of  $k$ . The increased runtime of the training step is tolerable considering that the application of the trained model to the CTA data takes about ten times as much time. This is due to the fact that the largest part of the datasets presented in table 8.1 is used to estimate the point-source sensitivity of CTA which will be the topic of chapter 11. Only a small part of the diffuse gammas and protons is used to train the classifier and energy estimator. It is this smaller sample which is split during cross validation.

For the CTA analysis I use the cross-validated receiver operating characteristic (ROC) to gauge the performance of my classification. The ROC curve is produced by drawing the false positive rate ( $FPR$ ) versus the true positive rate ( $TPR$ ) while varying the prediction threshold  $\alpha$ . The  $TPR$  and  $FPR$  for a fixed  $\alpha$  are defined as

$$TPR_{\alpha} = \frac{TP}{FN} \quad \text{and} \quad FPR_{\alpha} = \frac{FP}{FP + TN}$$

and are therefore independent of class balance. This curve is monotonically increasing towards larger values of  $FPR$ . Instead of comparing the ROC curves themselves, different classifiers trained on the same data can be compared by the area under their respective ROC curve. A hypothetically perfect classifier would create no false positives and classify only true positives. Hence, the ROC curve for the perfect has a constant value and the area under the curve (AuC) would equal one. A classifier that only randomly selects an outcome, would create a diagonal line with an AuC of 0.5. The ROC and the AuC are no perfect benchmarks. Their validity has long been the spotlight of discussions in the machine-learning community. There are plenty of alternative suggestions. See [74] for an example. When large datasets are available, which is the case for CTA analysis, these problems are somewhat diminished. Within CTA, the ROC AuC is used to the classification strengths of competing methods for background suppression.

For regression problems no confusion matrix in the classification sense can be constructed. Instead a matrix showing the correlation between the true label  $\mathbf{y}$  and the prediction  $\hat{\mathbf{y}}$  is created on an independent test dataset. A perfect regressor would only have entries on the diagonal of that matrix. To express the quality of the regression in one number, the coefficient of determination or  $R^2$  score is often used. The  $R^2$  score is calculated from the residual sum

of squares  $RSS = \sum_i (y_i - \hat{y}_i)^2$  as used for the least-squares loss and the variance of the target variable  $\mathbf{y}$ . The definition of the  $R^2$  square is

$$R^2 = 1 - \frac{RSS(\mathbf{y}, \hat{\mathbf{y}})}{N \text{Var}(\mathbf{y})},$$

where  $N$  is the number of entries in  $\mathbf{y}$ . It can be understood as a *goodness of fit* where a value of 1 indicates a perfect match between regressor and truth. If the regressor, on average, predicts the mean of the target, then the  $R^2$  score will have a value of 0 [55, page 486]. For IACT analysis the  $R^2$  score is not sufficient to explain the quality of the energy estimation. Equivalently to the calculation of the angular resolution, the relative difference between the predicted and true energy is calculated for bins in estimated or true energy.

## 9.2 Decision Trees and Ensemble Learning

Methods like least-squares or neural networks try to find the global minimum of a loss function which was in itself defined over the entire parameter space. Often it is not possible to find a satisfactory solution without further assumptions about the shape of the parameter space. The idea behind decision trees is to split the parameter space into subspaces where the problem is potentially easier to solve [76, page 305]. In each of the subspaces the loss function is then optimized independent of the data in the other subspaces. If no satisfactory value of the loss function is found, the subspaces are split again into even smaller sub-subspaces. Once the partitioning has finished, each subspace  $R_m$  is assigned a label  $c_m$ . In the regression case, the mean of the true labels from the training set in  $R_m$  is assigned. For classification, the majority class is assigned to the subspace. The decision function returns  $c_m$  for new point  $x_i$  located within the region  $R_m$

$$f(x) = \sum_{m=1}^M c_m \mathbf{1}(x \in R_m)$$

where  $\mathbf{1}$  is the indicator function. Finding the optimal partition of subspaces in order to construct a decision tree is an NP-complete [87] problem. Decision tree methods employ a greedy strategy to build the partition [76, page 307]. The tree is build by performing recursive binary splits of the subspace. Starting from the tree's root, the parameter space is split at a

into a set of “left” and “right” partitions. A variable  $j$  and a split value  $S$  are chosen which define the partition

$$R_L(j, S) = \{x \in \mathbf{X} \mid x^{(j)} \leq S\} \quad \text{and} \quad R_R(j, S) = \{x \in \mathbf{X} \mid x^{(j)} > S\}.$$

The best point  $S$  and variable  $j$  along which to partition the space is found by iterating through all possible splits for all columns in  $\mathbf{X}$  and choosing the one which minimizes the loss  $(j, S) = \arg \min_{(j,S)}(L)$ .

For classification tasks the *information gain loss* is a popular choice. It is known under different names such as the *cross-entropy* or just *entropy* criterion. The loss for the information gain (IG) is defined as

$$L = -\text{IG}(Y, R) = -(H(Y) - H(Y \mid R)).$$

The function  $H$  is known as the information entropy

$$H(Y) = - \sum_{g \in G} P(Y = g) \log_2 P(Y = g).$$

For the random variable describing the target distribution  $Y$ . The probabilities  $P$  are assumed to be uniform and are simply approximated by the class proportions. In a binary classification problem this simplifies to

$$H = -p \log_2(p) - (1 - p) \log_2(1 - p),$$

where  $p = \frac{|R_L|}{|R_L| + |R_R|}$ . The conditional entropy  $H(Y \mid R(j, S))$  is the entropy of  $Y$  after the split has been applied. It can be written in terms of probabilities as

$$H(Y \mid R) = \sum_{R \in \{R_L, R_R\}} P(X \in R) H(Y \mid x \in R),$$

where  $H(Y \mid x \in R)$  is the entropy of  $Y$  for all data points  $x$  in subspace  $R$ . Again, the probabilities are assumed to be uniform and can be approximated by their proportions

$$H(Y \mid R) = \frac{|R_L|}{|R_L| + |R_R|} H(Y \mid x \in R_L) + \frac{|R_R|}{|R_L| + |R_R|} H(Y \mid x \in R_R),$$

where  $|R_L|$  and  $|R_R|$  is the number of samples in the left and right partition respectively. Once the best split has been found, the process continues recursively until a stopping criterion is met. In the case of regression, where  $Y$  is continuous, the process is the same. The only

difference is the choice of loss function. For most regression problems, including the energy estimation performed here, the residual sum of squares is minimized in each partition

$$L \propto \sum_{R \in \{R_L, R_R\}} \text{Var}(Y \mid x \in R_L).$$

In other words, the split  $(j, S)$  which minimizes the total variance of the target in each partition is chosen.

Tree building algorithms try to find the optimal split criterion in some local region of the parameter space. Finding the best overall split in parameter space is computationally infeasible. This means the decision tree algorithm can run into a local optimum. When building a decision tree one needs to find a balance between the tree depth and its tendency to overtrain. In general, a decision tree whose final nodes, or leafs, contain too few samples, does not generalize well. The number of times a certain feature has been selected for splitting during optimization carries some notion of *feature importance*. This is a useful diagnostic tool for validating ones assumption regarding the classification.

The idea of *ensemble learning* is to train several weak classifiers on different subsets of the data and then combine them into one strong classifier by averaging their output. This makes the decision tree method relatively robust to overfitting without losing much runtime since the creation of the ensemble is trivial to parallelize. The most common way to build a decision tree ensemble was popularized by Breiman [36]. Here the training data is split into subsets using sampling with replacement, which is also known as bootstrapping. An additional source of randomness, or variance, was introduced by Breiman by limiting the choice of variables  $j$  in each partition to a random subset of features. This construction is called a “Random Forest” [37]. For use in ensemble methods another layer of randomization can be applied to the tree building process. Instead of iterating through all possible cuts for the  $n$  randomly selected features, only  $n$  randomly chosen splits are considered. These *extremely randomized trees* [68] can be used in bagged ensembles just like the original Random Forest, but with much faster training times.

For my CTA analysis I use an ensemble of extremely randomized trees for classification of particle type and regression of energy.

### 9.3 AICT-Tools

To ensure reproducibility for the entire machine-learning process, I started the open-source Python project called `aict-tools`. It consists of a fully configurable pipeline to perform the common machine-learning tasks encountered in IACT analysis. Given a set of input data and configuration files, it splits the data into test and training sets, performs on the fly feature generation, trains and applies the models, and creates the numbers and charts needed to gauge the performance of the trained models. Each task is dealt with by a single-purpose command line application. This was a design decision made to facilitate the use external tools such as `make` to model the data dependencies between each task. chapter 12 contains a short discussions on the use of Makefiles to ensure reproducibility. The classification and regression algorithms executed by the `aict-tools` are implemented in the `scikit-learn` library [129]. Arbitrary `scikit-learn` predictors can be used by the `aict-tools` as long as they produce continuous output. Everything is configured in human readable `yaml` files.

Originally build for the FACT project, I added new functionalities to the `aict-tools` to support the analysis of CTA data. The goal was to make it possible to work with data on a single telescope level, i.e. the telescope-events, as well as the array-event level. The output of the CTA preprocessing comes in the form of three tables stored within a single HDF5 file. The logical layout of this DL2 data file can be seen in table 9.1. This layout brings about some challenges concerning the I/O functionalities of the `aict-tools`. First and foremost, the splitting of the data into training and validation sets has to take the simulation information into account. The data belonging to a single simulation run must not be separated. Otherwise the calculation of instrument responses and event weights, as described later in section 11.1, would not be possible. Second, in order to train models on both single telescope features as well as array-wide information, the tables need to be merged. The tables are joined on the primary key which consists of the unique identifiers for the run, array-event, and telescope. The canonical way to support this kind of operation is by using the well known `pandas` library [111]. While the library is a powerful tool to perform operations on table-like data, it does not support memory mapping techniques or out of memory computing. The overhead in terms of required RAM are considerable. For the `aict-tools` I implemented a way to read and merge the CTA data in a chunk-wise manner. The `aict-tools` work with the full PROD3B analysis as well as the simulations for the prototype of the CHEC camera. The code, including example configurations, can be found at

<https://github.com/fact-project/aict-tools> 



**Table 9.1:** Logical structure of the DL2 data as produced by the CTA preprocessing script. All tables are stored in a single HDF5 file. The unique primary keys allow for merging and joining of information. This allows for a straightforward access to the simulated values for any given telescope-event. The telescope-event identified by the tuple (58, 4100, 92), for example, can easily be associated with the simulated energy, “mc\_energy”, of the array-event (58, 4100). For the training of machine-learning models, the array-event table is merged with the telescope-event table. For application of the models the data is read chunk-wise by reading the entire index into memory and then reading blocks of connected array-events. The table below shows the first exemplary lines of the diffuse simulated gamma-ray dataset described in table 8.1. Some column names were shortened in order to fit the table to the page. A full listing of column names can be found in the `aict-tools` configuration file in appendix B.

#### Runs

run_id	sim_index	num_showers		shower_reuse
14	-2	20000	...	10
148	-2	20000	...	10
160	-2	20000	...	10
	⋮			

#### Array Events

run_id	event_id	num_sst	mc_energy		total_intensity
14	1103	0	0.011	...	89.319
14	4305	0	0.077	...	1640.577
14	5705	0	0.015	...	174.140
	⋮				

#### Telescope Events

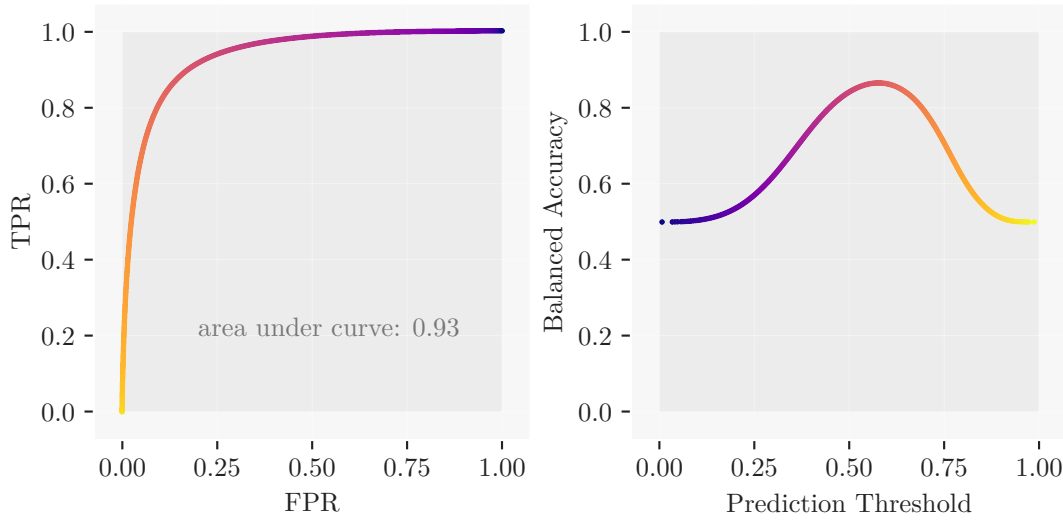
run_id	event_id	telescope_id	width	length		intensity
14	1103	4	0.018	0.024	...	56.402
14	1103	6	0.017	0.022	...	32.918
14	4305	4	0.027	0.042	...	283.853
	⋮					

## 9.4 Application to CTA Data

The datasets listed in table 8.1 are split into two parts. One for training the models and one for evaluating the physics performance. The classification model for background suppression is trained to separate diffuse gammas from proton simulations. The electron dataset is left as is and not used as input for the training step. All models are trained on a single telescope level. The array-events table is merged with the telescope-events table with an inner join.

Each array-event has at least two associated telescope-events. This means that array-wide information is simply duplicated for each row belonging to the same array-event. When the model is applied to new data, the predictions for the single telescopes are aggregated to build the estimate for the whole array-event. The application of nested machine-learning models show large potential for improvements [116]. For this analysis, the simple, unweighted average, is built for the prediction of the array-event. After the tables have been merged, new features are generated by the `aict-tools`. These new features are built by combining the values in different columns using elementary mathematical operators. The model settings are completely defined by the configuration file for the `aict-tools` which is listed in appendix B. The features that were selected for the model training as well as the definition of the generated features are included in the configuration.

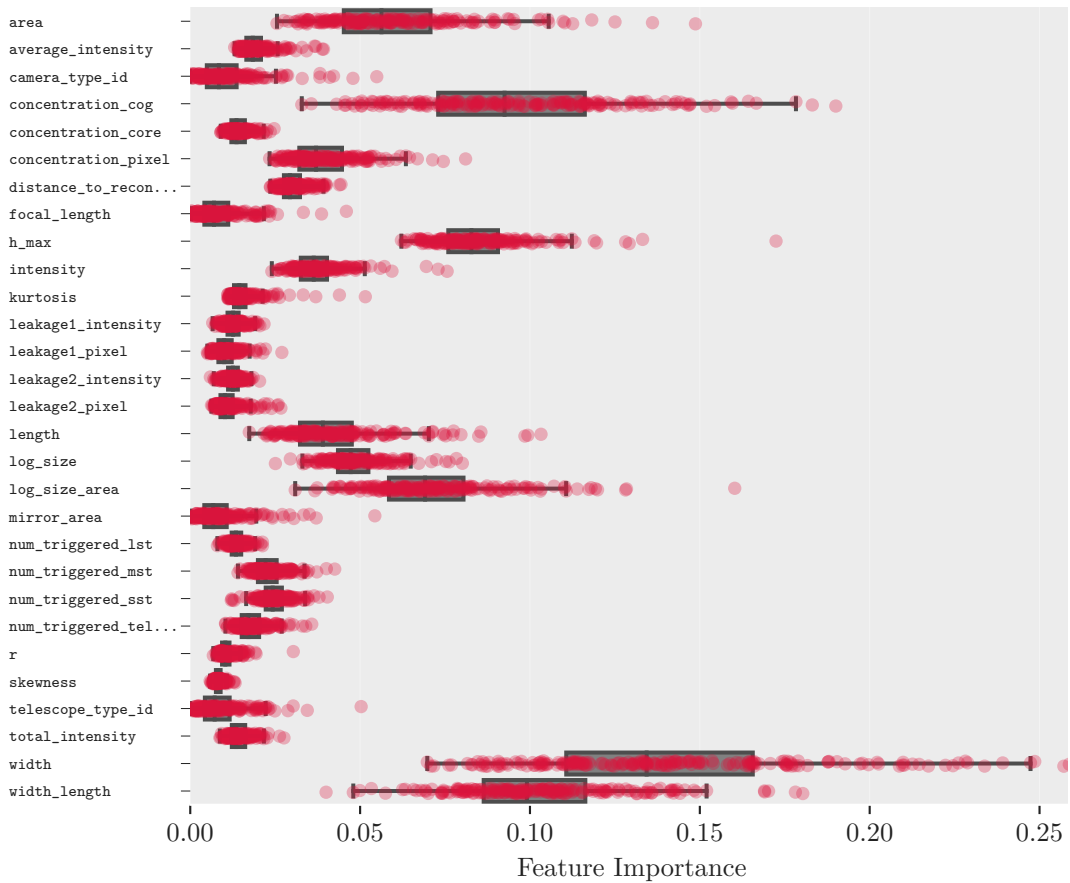
Of the original datasets, 1.5 % of the protons and 5 % of the diffuse gamma rays are used for training. That leaves 295 476 single telescope-events for the protons and 406 775 events for the gamma-ray data. No additional selection cuts are applied to the data prior to training the classifier. For this analysis an ensemble of 200 extremely randomized trees was trained. Each tree was limited in depth by only allowing further splits in nodes with at least 100 samples. During training a 5-fold cross validation is applied to evaluate the performance. The cross validated area under ROC curve is  $0.88 \pm 0.001$  which is in the same range as other analysis results previously circulated within the CTA consortium. It is difficult to make a rigorous comparison on this performance estimate alone. The classification strength depends strongly on the distribution of the data. Any sort of selection cut applied before training the model can improve the classification strength by a great deal. The same is true for the adaption of cleaning thresholds. Another layer of complexity is added when taking energy dependencies into account. In IACT analysis, it is common practice to look at the performance estimates in different simulated energy ranges, akin to figure 8.2 where the angular resolution was plotted with respect to the energy. For classification, this is difficult to justify as the energy distribution of the background events is unknown in observed data. The migration of the background events between the energy bins could be completely arbitrary, so that the classification strength measured on the simulated background energy might not resemble to performance on real observations at all. A comparison on the level of estimated energy might be more fair. However, this makes it hard to deconvolve the effects of two multivariate models with each other when comparing these figures between different analysis approaches. These problems have been recognized by the rest of the CTA collaboration as well and an effort has started to unify and define benchmarks between analyses [105]. Figure 9.2 shows the receiver operating characteristic (ROC) of the classifier on the entire energy range



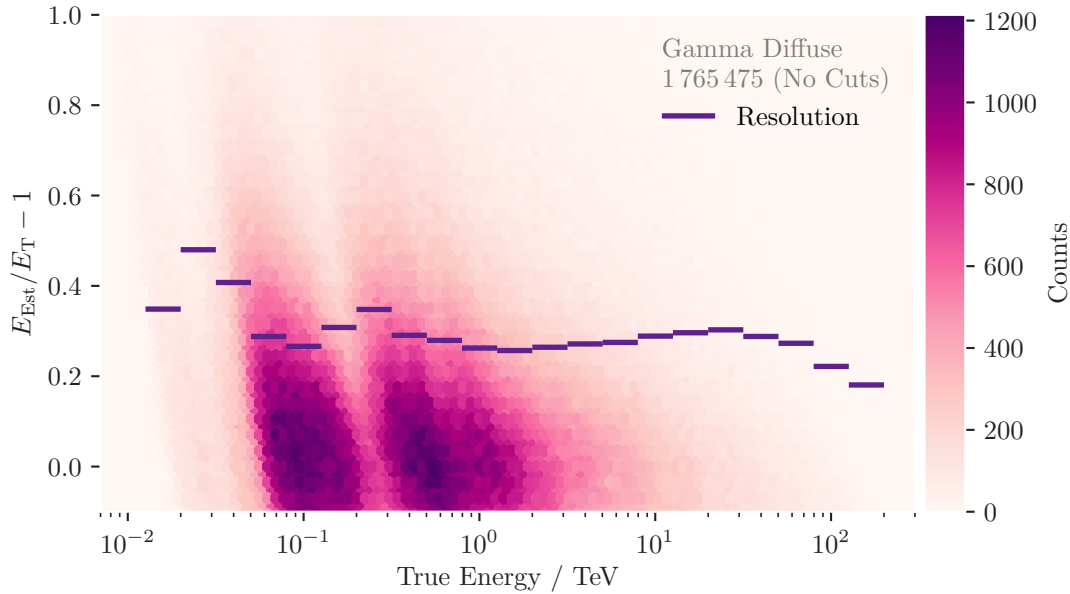
**Figure 9.2:** The left-hand side shows the receiver operating curve for the test data with 19 410 067 single-telescope proton events and 7 765 142 gamma events. The color scale indicates the prediction threshold corresponding to the pair of true and false positive rate. The figure on the right-hand side depicts the balanced accuracy of the classifier versus the prediction threshold. Both figures use the same color scale.

together with the balanced accuracy versus varying prediction thresholds. These numbers were calculated on the large test sets of diffuse gamma-ray and proton events. Figure 9.3 shows a list of all training features together with their feature importances as estimated by every tree in the ensemble.

Energy regression is handled in a very similar way to the classification. A total of 200 extremely randomized trees were trained on 29 features. The target variable, the simulated energy of the primary particle, was transformed by  $y' = \log_{10}(y)$  before training the classifier. This decreases the total range of the parameter and limits the dynamic range of the variance, or residual sum of squares, which is optimized during training. Using the transformed target, the estimator is able to put more weight on the low end of the energy range. The regressor was evaluated in a 5-fold cross validation. The mean  $R^2$  score for this model is  $0.759 \pm 0.007$ . In contrast to the classification, the behavior of the energy estimator with respect to the true energy is of major interest. The closer the estimated energy to the true energy, the lower the off-diagonal entries in the response matrix and in turn the calculation of fluxes and energy spectra. The energy resolution is calculated from the relative distance between the true and estimated energy  $d = \frac{E_{\text{Est}} - E_T}{E_T}$ . Two definitions for the energy resolution are customary in



**Figure 9.3:** A total of 29 variables are used to train the classifier. The box plot shows the inter-quartile range of the feature importances for each variable. Both single-telescope and array-wide features are deemed as relatively important by the decision tree ensemble. The classifier was also fed with instrument parameters such as `camera_type_id` and `mirror_area`. While it does not describe parameters of the air shower, it helps with learning the peculiarities in the different cameras and optical systems. The concentration parameters such as `concentration_cog` are often selected to split nodes in the decision tree. These parameters describe the amount of light collected in the center pixel with respect to the rest of the selected pixels in the shower.

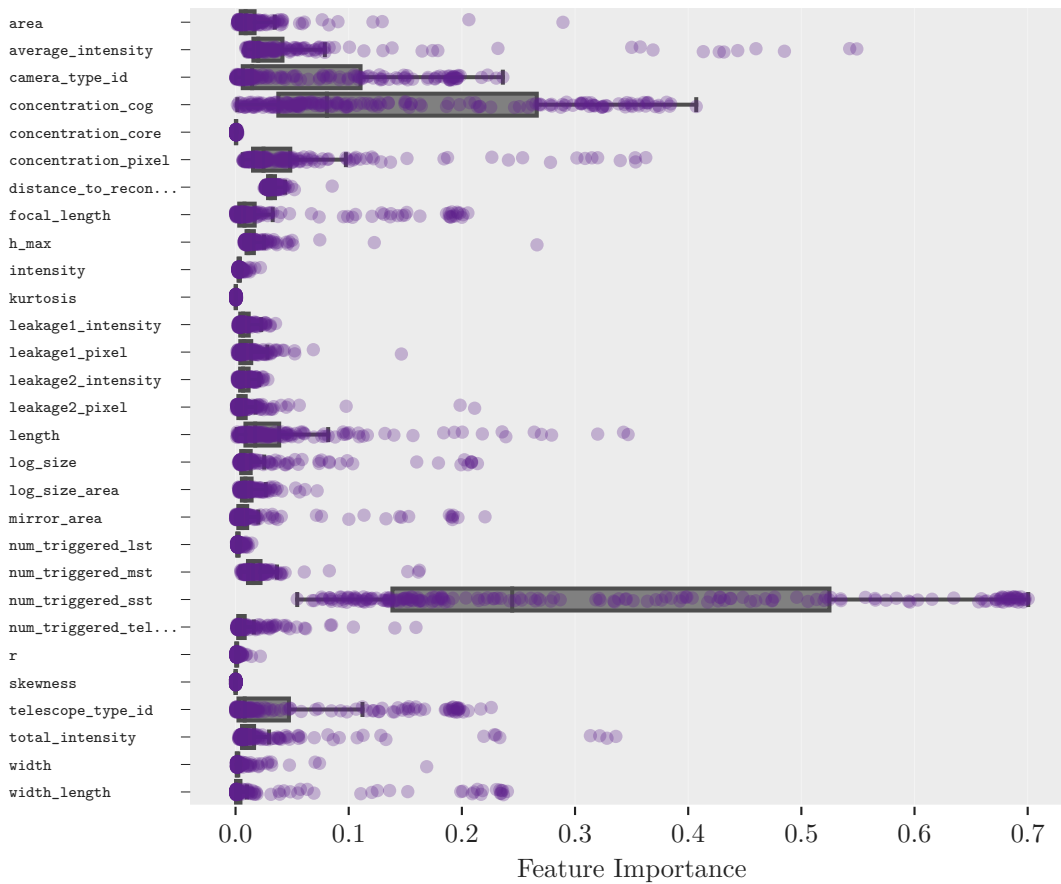


**Figure 9.4:** The figure shows the energy resolution of the regressor. It was trained on 406 775 gamma events with 29 features. The histogram in the background shows the distribution of  $d = \frac{E_{\text{Est}} - E_T}{E_T}$  with respect to the true energy. The peaks in the distributions correspond to the maximum acceptance probability of the different telescope types. Similar to the behavior of the angular resolution, the energy estimation improves with increasing energy up to the point where the lack of image containment prevents accurate energy reconstruction.

the IACT community. First, the 68<sup>th</sup> percentile of the relative distance and second, half the width of the central interval between the 16<sup>th</sup> and the 84<sup>th</sup> percentile

$$R_{E_{\text{Est}}} = \frac{1}{2}(Q_{84}(d) - Q_{16}(d)).$$

The second definition, the central interval, is used throughout this text. The *energy bias* is defined as the median of the relative distance distribution. Figure 9.4 shows the energy resolution of this regressor. Similar to benchmarking the classification strength, it is difficult to compare the energy resolution across different analyses. Selection cuts applied before applying the model have a large impact on the energy resolution. The optimization of the event selection cuts is strongly dependent on the physics use case. The next section will cover the optimization for a point-source analysis. Figure 9.5 shows the feature importance for every variable used by the regressor.



**Figure 9.5:** A total of 29 variables are used to train the regressor. The box plot shows the interquartile range of the feature importances for each variable. Somewhat surprisingly, the feature called `total_intensity`, which is the sum of the single telescope intensities, is not the most important observable. The `num_triggered_sst` variable is deemed most important by the regressor. This can be explained by the fact that the SST type telescopes can trigger on high-energy showers with large impact distances due to their large field of view. In addition, the small mirror excludes them from triggering very dim showers.

# 10 A Prototype for Real Time Analysis

One of CTA's primary goals is to facilitate multi-wavelengths observations [134, Chapter 9]. Observation of serendipitous events and response to multi-messenger alerts put strong constraints CTA's data acquisition system. The short timescale capabilities of CTA are a key element in the design process. The telescopes mechanical systems capable to target any point in the sky within 90 seconds. Transient events or flaring sources, e.g. active galactic nuclei that increase the energy output by orders of magnitude within tens of seconds, are of major scientific interests for CTA and the entire astroparticle community. CTA will run a continuous real-time analysis (RTA) which operates on the on-site computing infrastructure. This allows CTA to alert facilities operating in other wavelengths for fast follow-up observations. Official CTA requirements state that the on-site analysis must be able to notify operators of transient events within just 30 seconds of recording the data. Judging from the simulations used for my `ctape` based analysis, the rate for protons and electrons is approximately 21 371 events per second. That is *after* the preprocessing has been applied. The total trigger rate will be even higher. These staggering event rates signify the challenges which have to be overcome by the real-time analysis system. CTA's on-site computing resources are concentrated in one main cluster. Additionally, each telescope is equipped with a *camera server* that performs a low-level calibration of the data. The data will then be send from the corresponding camera server to a central trigger via ethernet. This software trigger bundles the single telescope-events into array-events if they arrive coincidentally and discards them otherwise. The RTA will operate on calibrated images. The process of integrating and calibrating the raw signal data is the responsibility of dedicated software which will run on the camera servers or some central facility. Current plans envision that CTA's real-time analysis will receive the calibrated images via multiple network endpoints. As is often the case in the world of high-energy physics, self-built solutions for handling these data streams are currently being developed by multiple physicists and engineers in CTA member institutions. Here I propose an alternative approach.

Over the past decade or so, a plethora of distributed computing frameworks have been popularized by the big data-driven companies like Google, Twitter, Facebook or Amazon. Some

of these solutions are slowly being introduced in places like the LHC at CERN [58]. In the astroparticle community, adoption of these new computing technologies is reluctant at best. All popular “big-data” frameworks such as Spark [150], Storm [139], and Heron [97] support fault tolerance computing and high availability mechanisms to recover from hardware or network failures. As part of my research stay at CEA Paris, I developed a prototype for analyzing data from the CTA array [39] using a well established open-source framework for distributed computing called Apache Flink [43]. The resulting program, dubbed `jayct`, can be executed in a distributed manner on heterogenous infrastructure without the need for hand-written parallelization routines. I chose the `flink` framework for `jayct` due to the simplified setup and more comfortable high-level API compared to other frameworks. The program performs image cleaning, calculates Hillas parameters, reconstructs the event’s direction, and applies pre-trained machine-learning models for particle and energy prediction. Like most big-data processing frameworks, `flink` is executed on the Java Virtual Machine (JVM). The big-data ecosystem almost exclusively relies on the Java runtime due to its remote debugging features and capability to execute compiled programs on any operating system and hardware platform. The reconstruction algorithms in `jayct` are essentially a Java re-implementation of FACT software and `ctapipe` methods.

Frameworks for distributed streaming such as `flink`, provide high-level abstractions that allow the user to model the dataflow as a graph in terms of sources and sinks. In this use case, the `flink` data sources output calibrated images which were generated beforehand. The source is connected to the sink nodes via the composition of map, filter, windowing, and aggregation operations. Each step in the computation is distributed to an arbitrary number of parallel *slots*. The location of the slots is not defined by the user, but instead is automatically delegated to any physical machine with sufficient resources. To emulate the behavior of CTA’s real-time analysis, we modeled multiple data sources in `jayct` which access the simulated images. The results are dumped into a single sink which writes the results to a `csv` file. The `jayct` program supports two modes for distributing the events between the test machines. In the first variant, each image in an event gets treated separately for the calculation of the image parameters. The image cleaning is performed on all images in a loop before they are split into separate data items. Then the Hillas parametrization, background suppression, and energy prediction is applied to each image separately before they get collected based on their event ID. This last step is performed in a windowed aggregation operation which accumulates the images within a fixed window of 5 seconds.

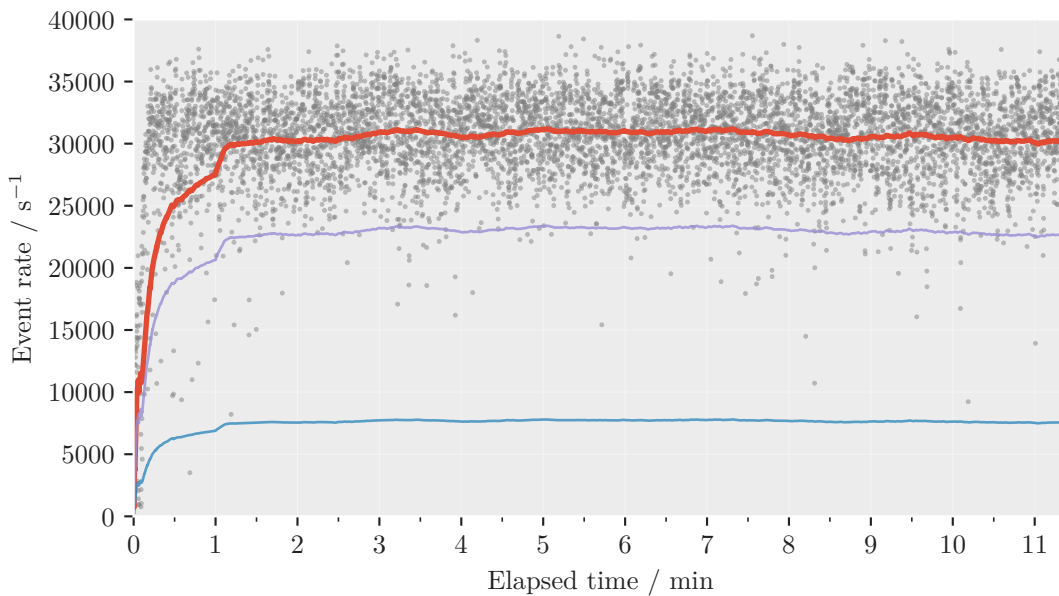
The second, and arguably easier, approach to distributing the computation, treats each array-event independently. In this case the images belonging to one array-event are not separated



and no windowed aggregation operation is necessary. Figure 10.1 shows the event rates achieved by this second variant on just two machines. The thick line shows the mean data rate achieved by jayct. It stays close to 30 000 events per second which is well above the estimated background event rate. In this, admittedly simplified, setup, two or three large machines will suffice to perform CTA's real-time analysis. Already at this point in time, with only one LST prototype operating, powerful computing infrastructure that consists of several hundred dedicated compute nodes is running on La Palma. Using established frameworks for distributed computing can help to bring down the cost of hardware, energy, and maintenance for CTA. The big-data industry has produced many battle tested solutions for these types of problems. There is no need to reinvent the wheel.

The source code for the jayct real-time analysis prototype can accessed at

<https://github.com/kbruegge/jayct> 



**Figure 10.1:** Two machines were used to benchmark the event rates reached by jayct. The gray dots indicate the sampled event rates in a 100 ms window. The bold line shows a running average over 1 minute. The two thinner lines show the mean event rates of the single machines. A total of 40 out of 96 available threads were used for this test, of which 30 were blocked on one machine and 10 on the other. The machines were not isolated from other users. Multiple workloads were performed by other users during the execution of this benchmark.



# 11 Sensitivity Computation

The sensitivity curve is a widely used tool to compare different telescope types and analyses in terms of their detection capabilities of gamma-ray point sources. This energy dependent curve shows the minimum flux required for the instrument to detect a source. CTA was specifically build to detect unknown and faint sources in the very-high-energy gamma-ray sky. The project will improve the sensitivity of TeV gamma-ray telescopes by an order of magnitude at least. The point source sensitivity curve is often considered to be the final proof of validity for a CTA analysis.

The reference analysis for CTA is implemented in the `Eventdisplay` and `MARS` software packages. Both are closed-source programs based on C++ and ROOT. The `Eventdisplay` [107] program was originally developed for the VERITAS project. It was adapted for CTA in order to calculate instrument responses during the early stages of the CTA development. The `MARS` [117] project is also based on ROOT and has been in use for the MAGIC telescope ever since its conception. The official CTA instrument responses are calculated using both `MARS` and `Eventdisplay`. Detailed information about the inner workings of both analyses is sparse. Some details about the simulations are given in [11]. The most detailed source of information is the internal report titled “Description of CTA Instrument Response Functions” [106]. Unfortunately this document is accessible to CTA members only. The latest official performance figures can be downloaded at the CTA observatory website [48]. Neither `MARS` nor `Eventdisplay` interoperate with a high-level programming language, let alone the official CTA pipeline prototype `ctapipe`. It was an explicit goal of this thesis to create an open-source pipeline based on `ctapipe` and other open-source tools that rivals the physics performance of the reference implementation. The significance and sensitivity assessments presented in this chapter are the ultimate benchmark for comparing the different analyses.

Calculating a sensitivity curve involves multiple intermediary steps and large amounts of simulated data. The full simulations used to produce the sensitivity curve for this analysis need approximately 20 TB of disk space. As mentioned previously, a fair comparison of these intermediate steps is difficult if not outright impossible. Many analyses apply a pre-selection of events to remove events that are likely to be miss-reconstructed. Images that are not fully

contained in the camera are often either removed by manual cuts or due to the way the reconstruction algorithms are implemented. This might consequently improve quantities like the angular and energy resolutions. An additional layer of complexity comes from the fact that many of the published performance curves are drawn with respect to estimated energy. An unbiased comparison is impossible without using the same model for energy regression. Official CTA performance curves, specifically angular resolution, energy resolution, and effective area, are based on an optimized event selection. The official event selection is performed by maximizing the point-source sensitivity in each energy bin independently.

In this chapter I reproduce parts of the CTA reference analysis in order to compare it to the performance of my, Python-based, open-source approach. As explained in section 5.2, the detection probability for a single gamma-ray shower is highly dependent on the particle's energy. To calculate the response of the detector to a real source, the simulated observations have to be reweighted to the spectrum of a real astrophysical source. Section 11.1 gives details on how these weights are calculated. Previously, in section 5.3 I described how the number of background counts in an IACT observation is estimated. This approach has to be modified for the CTA simulations used here. The adapted method is explained in section 11.2. The definition of sensitivity is strongly connected with the concept of detection significance. Section 11.3 describes how the detection significance of a point source is commonly defined in the IACT community. In that same section I optimize the event selection to find the best significance for a hypothetical observation of the Crab Nebula and show the infamous  $\theta^2$ -plot. Finally, section 11.4 presents the result of the optimized event selection and shows the resulting point-source sensitivity curve.

### 11.1 Event Weights

Events simulated by the shower simulation CORSIKA have their intrinsic energy distribution depend on some non-physical spectrum set during configuration. In order to produce realistic estimates for the telescope's performance in terms of real physical sources, these distributions have to be transformed. As mentioned before, the energy distribution of the simulated showers follows a power law. While a uniform energy distribution would be preferable in many situations, such as model training for energy estimation, the computing requirements increase with higher particle energy. The shower simulation has to track the interactions of each secondary particle in the atmosphere. The larger the primary particle's energy, the larger the number of secondary particles and hence, the computing time. In particular, we

want the distribution of simulated events to look like the distribution of events from a typical gamma-ray source in the sky e.g. the Crab Nebula. In addition to the different shapes of the simulated and real distributions, the absolute flux i.e. the total amount of particles per area and time, of the source also needs to be recreated. This is necessary to get meaningful estimates for a detection significance, e.g.  $5\sigma$  in a 10 min observation of the Crab Nebula, from simulations alone.

The reweighing has to be performed for the signal events, the simulated gamma rays, as well as the cosmic ray background events, the simulated protons and electrons. The CORSIKA simulation is configured by providing its minimum energy  $E_{\min}$ , maximum energy  $E_{\max}$ , the radius of the circular area on the ground in which particles are distributed  $A_{\text{sim}}$ , the opening angle of the cone in which particles from an extended source are produced  $\alpha$ , the total number of primary particles to simulate  $N_{\text{sim}}$ , and the spectral index  $\gamma$  of the power-law defining the shape of the energy distribution  $E^{-\gamma}$ . The flux  $F_{\text{sim}}(E) = \Phi_{\text{sim}} E^{-\gamma}$  in terms of physical units which corresponds to these configuration settings can be found by using the fact that the total number of expected counts from  $F(E)$  has to equal the number of simulated counts

$$N_{\text{sim}} \stackrel{!}{=} \int_{E_{\min}}^{E_{\max}} \Phi_{\text{sim}} E^{-\gamma} dE \cdot A_{\text{sim}} \cdot t_{\text{obs}} \cdot 2\pi(1 - \cos(\alpha)) = N,$$

where  $t_{\text{obs}}$  is the assumed observation duration. Solving for the only unknown variable,  $\Phi_{\text{sim}}$ , results in

$$\Phi_{\text{sim}} = \frac{N_{\text{sim}}}{N}.$$

Now we can calculate the energy dependent weight  $w(E)$  for each simulated event with respect to a target spectrum from an astrophysical source  $F_{\text{target}}$ . The weights have to be chosen so that the energy distribution of particles from the target spectrum  $P_{\text{target}}$  equals that of the simulated spectrum

$$P_{\text{target}}(E) = \int_0^E F_{\text{target}}(E') dE' \stackrel{!}{=} \int_0^E F_{\text{sim}}(E') dE' = P_{\text{sim}}(E),$$

which is satisfied when

$$w(E) = \frac{F_{\text{target}}(E)}{F_{\text{sim}}(E)}. \quad (11.1)$$

These weights are used when counting the simulated events or building event distributions using histograms. To compare the sensitivity and detection significance between different analyses and telescope designs, it is necessary to use the same assumptions about the tar-

**Table 11.1:** The spectral shapes chosen for reweighing the simulated events. The MAGIC spectrum for the Crab Nebula follows a log-parabolic spectral shape. The spectrum used to weigh the gamma rays has a strong influence on the event optimization. Some analyses use the HEGRA spectrum published in 2004, which follows a simple power-law. As seen in chapter 5 and the measurements published by MAGIC later, this seems to be inaccurate. Hence, the MAGIC spectrum is used here. The electron spectrum is the result of a fit to Fermi-LAT data released by CTA for internal comparisons. The function  $f_{\mathcal{N}}$  is the probability density of the normal distribution. The function is relatively complex as it tries to capture a “bump” in the spectral shape. The cosmic-ray spectrum was chosen in accordance with the CTA analysis working group to only include protons. A more realistic alternative could be achieved by using a cosmic-ray spectrum which includes heavier nuclei as well. The reference energy is fixed to  $E_0 = 1$  TeV. The spectra are given in units of  $\text{cm}^{-2} \text{s}^{-1} \text{GeV}^{-1} \text{sr}^{-1}$  for protons and electrons and  $\text{cm}^{-2} \text{s}^{-1} \text{GeV}^{-1}$  for the point-like gamma rays.

Particle	Publication	Shape
Gamma-Ray (Crab)	MAGIC (2015) [17]	$3.23 \times 10^{-8} \left(\frac{E}{E_0}\right)^{-2.47-0.24 \log_{10}(E/E_0)}$
Electron	Fermi-LAT (2009, 2010) [7, 12]	$2.38 \times 10^{-12} E^{-3.43} \left(1 + 1.95 \left(e^{f_{\mathcal{N}}(\log_{10}(E/E_0) \sigma=0.741, \mu=-0.101)} - 1\right)\right)$
Proton	BESS (2000) [132]	$9.6 \times 10^{-9} \left(\frac{E}{E_0}\right)^{-2.7}$

get spectra. Specifically which spectral shapes are used to weigh the signal and background events. Throughout this chapter the simulated gamma rays are weighted with the Crab Nebula spectrum as published by the MAGIC collaboration in 2015 [17]. The spectrum of cosmic-ray protons is taken from results of the BESS spectrometer [132] published in 2000. And last but not least, the spectrum of cosmic-ray electrons which was fitted to flux points published by the Fermi-LAT collaboration [7, 12]. The fit was performed by the authors of CTA’s reference analysis and published in the internal IRF report [106]. These spectra were chosen in agreement with CTA internal discussions which have yet to be formalized into a public document. Table 11.1 gives an overview over the selected spectra.

## 11.2 Background Estimation

Point-like simulated gamma rays are used to emulate a source in the sky for which the detection significance can be calculated. However, unlike the “wobble mode” observations discussed in section 5.3, the simulations place the source right in the center of the field of view. In this case, the reflected regions method for estimating the background counts is unsuitable. Since the acceptance probability drops off rapidly with increasing distance to the center, simply placing the off-regions in radially symmetric fashion around the center would

underestimate the number of background counts in the on-region. With simulated data it is, however, straightforward to disentangle the contributions to the counts on the on-region  $N_{\text{on}} = N_s + t_\alpha N_b$ , since each event is labeled with the primary particle type. The signal counts  $N_s$  are calculated from the set of point-like gamma events which meet the selection criterion given by the classification model that was trained beforehand. Then from this set of “gamma-like” events  $S$ , all events whose reconstructed source position lies within a radius of  $\theta_{\text{on}}$  to the true simulated source position are retained while all other events are discarded. Using the weights  $w_i$  calculated in the previous section, the number of signal counts can be calculated by

$$N_s = \sum_{w_i, \theta_i \in S} w_i \mathbf{1}(\theta_i < \theta_{\text{on}}) \quad (11.2)$$

The estimation of the background counts works in a similar manner. First all simulated electrons and protons which are selected as “gamma-like” by passing the prediction threshold of the classifier, are gathered into the set of background candidates  $B$ . The better the background rejection by the classifier, the lower the number of events in  $B$ . Even with a strong classifier, the large abundance of cosmic rays collected during observations, create a smooth distribution of background events in the field of view. In simulated data the amount of background data is limited by computational resources. In order to produce a finely binned sensitivity curve, a minimum amount of counts are required to avoid biases in the calculated significance due to fluctuations of the background counts. To alleviate this problem the radius around the true source position from which the background counts are estimated is enlarged to a radius of  $1^\circ$ . The number of counts in that region is then scaled to correspond to the size of the off-region. The off-region is larger than the on-region by a factor of  $1/t_\alpha$ . The scaling factor  $n$  between the size of the background region and the circle with a radius of  $1^\circ$  is given by

$$n = \frac{1}{t_\alpha} \frac{\pi \theta_{\text{on}}^2}{\pi 1^\circ{}^2} = \frac{\theta_{\text{on}}^2}{t_\alpha}$$

so that the total number of background counts in the scaled region can be estimated by

$$N_b = \sum_{w_i, \theta_i \in B} n w_i \mathbf{1}(\theta_i < 1^\circ) = \sum_{w_i, \theta_i \in B} \frac{\theta_{\text{on}}^2}{t_\alpha} w_i \mathbf{1}(\theta_i < 1^\circ). \quad (11.3)$$

The estimated numbers are strongly dependent on the prediction threshold  $\alpha$  and the size of the on-region  $\theta_{\text{on}}$ . For the observation of a single point source, these two parameters can be optimized to achieve the highest Li&Ma significance. CTA does not officially specify a method that defines how the background events should be estimated. This introduces another source of uncertainty when comparing analyses with each other. The choice of

background region has a large influence on the total number of background counts. Other CTA analyses parameterize the dependence of  $N_b$  on the prediction threshold in each energy bin by fitting functions to the background event distribution. The choice of function shape for these distributions is hard to justify and rather ad-hoc.

### 11.3 Detection Significance

The conventional definition of *detection* in the IACT community is a significance level of at least  $5\sigma$ . Most established analysis of IACT data follow the arguments by Li and Ma [101] to build a statistical model of the measurement process in order to calculate the significance of an observation. I will recount a shortened version of the Li & Ma argument in this section. Despite all the efforts taken to separate signal from background by using machine-learning models, a non negligible amount of background events remain as detailed in section 5.3. The number of observed counts in the signal region, the “on” region, is made up from the number of actual signal counts from the source and the number of approximated background counts  $N_{\text{on}} = N_{\text{signal}} + t_\alpha N_b = N_{\text{signal}} + t_\alpha N_{\text{off}}$ . The expectation for the number of events in the on-region is defined accordingly as  $\mu_{\text{on}} = \mu_s + t_\alpha \mu_b$ , where  $\mu_s$  and  $\mu_b$  are the expected signal and background counts and  $t_\alpha$  is the relative exposure of the off-region compared with the on-region. See section 5.4 for detailed definitions. We previously defined the likelihood of the data given the parameters  $\mu_s$  and  $\mu_b$  in equation (5.11). For a single energy bin it was defined as the product of two Poisson distributions

$$\begin{aligned} \mathcal{L}(N_{\text{on}}, N_{\text{off}}, t_\alpha \mid \mu_s, \mu_b) &= P_{\text{on}}(N_{\text{on}} \mid \mu_s + t_\alpha \mu_b) \cdot P_{\text{off}}(N_{\text{off}} \mid \mu_b) \\ &= \frac{(\mu_s + t_\alpha \mu_b)^{N_{\text{on}}} e^{-\mu_s - t_\alpha \mu_b}}{N_{\text{on}}!} \cdot \frac{\mu_b^{N_{\text{off}}} e^{-\mu_b}}{N_{\text{off}}!}. \end{aligned}$$

The significance of a detection is calculated from a likelihood-ratio test. The null hypothesis  $H_0$  is that all measured counts in the signal region are due to background events alone  $\mu_s^{(H_0)} = 0$ . Respectively, the alternative hypothesis is given by  $\mu_s^{(H_1)} > 0$ . The likelihood ratio test statistic  $\lambda$  is defined as the ratio of the null hypothesis divided by the alternative hypothesis

$$\lambda = \frac{\mathcal{L}\left(N_{\text{on}}, N_{\text{off}}, t_\alpha \mid \hat{\mu}_s^{(H_0)}, \hat{\mu}_b^{(H_0)}\right)}{\mathcal{L}\left(N_{\text{on}}, N_{\text{off}}, t_\alpha \mid \hat{\mu}_s^{(H_1)}, \hat{\mu}_b^{(H_1)}\right)},$$



where  $\hat{\mu}_s^{(H_1)}$ ,  $\hat{\mu}_b^{(H_1)}$  and  $\hat{\mu}_b^{(H_0)}$  are the maximum likelihood estimates for the null and alternative hypotheses. The maximum likelihood estimates for the alternative hypotheses follow immediately from the definition of the problem to  $\hat{\mu}_s^{(H_1)} = N_{\text{on}} - t_\alpha N_{\text{off}}$  and  $\hat{\mu}_b^{(H_1)} = N_{\text{off}}$ . Li and Ma argue that in the case of the null hypothesis, where no signal events are measured in the on-region, the estimate for the background can be improved by also taking into account the amount of counted background signals in the on-region. Following that line of reasoning, the maximum likelihood estimate for the expected background is  $\hat{\mu}_b^{(H_0)} = \frac{t_\alpha}{t_\alpha + 1}(N_{\text{on}} + N_{\text{off}})$ . Following Wilk's theorem [146] the test value can be transformed to follow a  $\chi^2$  distribution with one degree of freedom

$$-2 \ln(\lambda) \sim \chi^2 \quad \text{and} \quad \sqrt{-2 \ln(\lambda)} \sim |\mathcal{N}(\mu = 0, \sigma = 1)|$$

when the number of observations approaches infinity  $n \rightarrow \infty$ . The significance of the observation follows immediately from the value of  $S = \sqrt{-2 \ln(\lambda)}$ . Inserting the values for  $\lambda$  yields the expression commonly known as the Li&Ma significance [101, eq. 17]

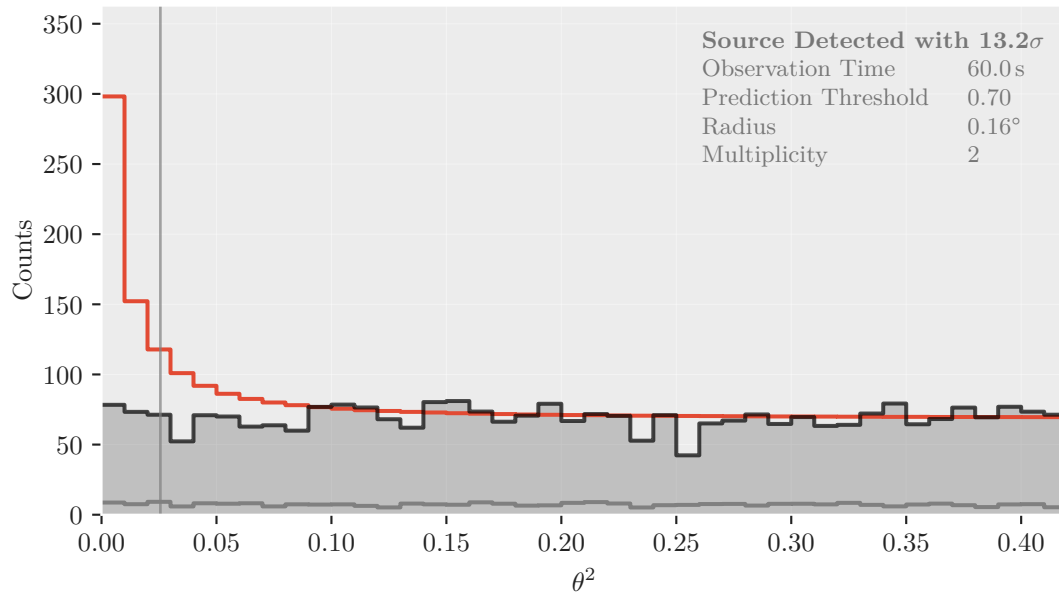
$$S = \sqrt{2N_{\text{on}} \ln\left(\frac{1 + t_\alpha}{t_\alpha} \frac{N_{\text{on}}}{N_{\text{on}} + N_{\text{off}}}\right) N_{\text{off}} \ln\left((1 + t_\alpha) \frac{N_{\text{off}}}{N_{\text{on}} + N_{\text{off}}}\right)}$$

The approximation by Wilk's theorem becomes problematic for low count statistics when the parameters for the Poisson distributions lie on the edge of the parameter space. Alternative solutions have been discussed by several authors [16, 44, 123]. The Li&Ma solution is used in almost all publications in Cherenkov astronomy whenever a detection significance is given.

The CTA analysis tries to maximize the sensitivity in each energy bin independently by varying the parameters for the event selection and hence changing the values for the event counts  $N_{\text{on}}$  and  $N_{\text{off}}$ . Three parameters are optimized in this analysis. The prediction threshold  $\alpha$ , the size of the on and off-region  $\theta$ , and the telescope multiplicity. The relative exposure is fixed at  $t_\alpha = 0.2$  throughout this chapter.

Figure 11.1 shows the infamous the  $\theta^2$ -plot. This image is a ubiquitous tool among Cherenkov astronomers to visualize source detections. The image shows the distribution of events from the on and off-regions with respect to the squared distance between estimated and true source position  $\theta^2$ . Here the entire energy range of the simulations is taken into account to produce the plot so that sufficient background events are available. The plot shows that the CTA analysis presented here would result in a definitive detection of the Crab Nebula within a fraction of a minute of observation time. To put this in perspective, the FACT telescope needs a few hours of observation time to achieve the same level of detection significance.

## 11 Sensitivity Computation



**Figure 11.1:** The  $\theta^2$  square plot for the Crab Nebula as seen by my CTA analysis. The red curve shows the distribution of the gamma rays together with the mean of the background counts. The shaded region indicates the total distribution of background events. The gray line towards the bottom shows the part of the background distribution which is due to electrons. The number of counts in the on and off-region correspond to the area under the red and black curves to the left of the vertical line. The larger the difference between the two, the stronger the detection. CTA will be able to detect gamma-ray sources much quicker than any previous IACT. As this result shows, sources as bright as the Crab Nebula can be detected in less than a minute with CTA. The multiplicity of the trigger, the radius of the on-region, and the prediction threshold were optimized to maximize the significance as given in section 11.3. Interestingly, the best prediction threshold in terms of significance roughly matches the location of the maximum of the balanced accuracy curve in figure 9.2. The faint vertical line near  $0.05^\circ$  shows the radius of the signal region. The  $\theta^2$ -plot gives a quick visual reference of the signal to noise ratio. It is a common sight in many publications [9, 10] and often displayed in the telescopes' control rooms.

## 11.4 Sensitivity Computation

The sensitivity of a telescope to a point-like gamma-ray source is defined as the minimum brightness the source must have in order for the telescope to detect it in a predefined time interval. The response IACTs is highly energy dependent. Hence, the sensitivity of an instrument depends on the shape of the source's emission. In high-energy Cherenkov astronomy the ubiquitous Crab Nebula is chosen as the prototypical source. Specifically for CTA, the sensitivity is defined as the minimum flux a source with a Crab Nebula like spectral shape must have so that it will be detected with a significance of  $5\sigma$  within a fixed observation time of 50 hours. This number is calculated in bins of estimated energy in order to produce an energy dependent sensitivity curve. This curve can then be compared to the spectral energy distribution of different sources and the sensitivities of similar instruments.

Given the counts in the on and off-region, the detection significance  $S(N_{\text{on}}, N_{\text{off}}, t_\alpha)$  is calculated according to section 11.3. In the simulated data, these numbers can be expressed in terms of signal and background counts  $N_{\text{on}} = N_s + t_\alpha N_b$  and  $N_{\text{off}} = N_b$ . At this point the signal and background events are already weighted according to their assumed physical spectra and observation time as described by equations 11.2 and 11.3. We are interested in finding the factor  $x$  by which the signal events have to be scaled so that the significance  $S$  reaches the required minimum significance level of  $5\sigma$ . This can be written as a minimization problem

$$\hat{x} = \arg \min_{N_s} (5 - S(N_{\text{on}} = xN_s + t_\alpha N_b, N_b, t_\alpha))^2$$

All values except  $x$  are fixed. This problem can be solved with a simple one dimensional minimization method. The result  $\hat{x}$  is the estimated *relative sensitivity*. If the relative sensitivity is larger than one, the source cannot be detected in that energy bin within the assumed observation time. If  $x$  is smaller than one, it describes the factor by which the source can be fainter and still be detected. Multiplying  $x$  with the flux of the target spectrum at the center of the energy bin yields the sensitivity in physical flux units. The statistical errors on  $x$  are computed by repeatedly sampling the  $N_s$  and  $N_b$  from a Poisson distribution and recomputing the sensitivity for each sample.

CTA introduces two additional constraints to mitigate the effects of statistical fluctuations. First, the number of excess counts has to be larger than 10, i.e  $N_{\text{on}} - t_\alpha N_{\text{off}} > 10$ . Second, the excess has to be larger than the uncertainty in the number of background counts by a factor of five. CTA assumes a modest 1 % of uncertainty leading to  $N_s > 5 * N_{\text{off}} * 0.01$ . For my analysis I supplement these constraints by requiring that the total number of unweighted

## 11 Sensitivity Computation

background events in a  $1^\circ$  radius is larger than 80 and assuming a background uncertainty of 2 %.

Energy dependent curves can only be compared if the energy binning is well defined. CTA uses 5 bins per decade of reconstructed energy starting at 0.02 TeV. The energy reconstruction differs between different analysis methods. Each regressor comes with an intrinsic energy bias in addition to its finite resolution. The relative difference between true and estimated energy  $d = \frac{E_{\text{Est}} - E_T}{E_T}$  as seen in figure 9.4 can be plotted versus true or estimated energy on the abscissa. The CTA reference analysis applies a so-called *bias correction*. Linear interpolation between the median values for  $d$ , the bias, creates a continuous function  $B(E_{\text{Est}})$  which can be used to correct the average offset in a bin of estimated energy. This function is then applied to the estimated energy values of the simulated signal and background events before they are distributed into the energy bins for further computations

$$E'_{\text{Est}} = \frac{E_{\text{Est}}}{B(E_{\text{Est}}) + 1}.$$

While the usefulness of this transformation is debatable at best, it is essential when comparing the sensitivity curves to the reference result.

To find the best sensitivity, the event selection is optimized in a brute force manner. A three-dimensional grid of test values is created for the minimum event multiplicity  $m$ , the radius of the on-region  $\theta$ , and the prediction threshold  $\alpha$

$$(m, \theta, \alpha) \in \{2, 3, \dots, 10\} \times \{0.01, 0.02, \dots, 0.17\} \times \{0.30, 0.35, \dots, 1.00\}.$$

This grid is then evaluated for each energy bin in parallel. From all tested combinations of  $(m, \theta, \alpha)$ , the event selection that produces the lowest relative sensitivity is chosen. Table 11.2 shows the resulting values for each energy range together with the calculated Li&Ma significance. In order to avoid overfitting, the step size of the grid is left intentionally large. No independent test data is available as the amount of available background simulations is limited. The results of the event selection is summarized in table 11.2 The selected thresholds in  $\alpha$  and  $\theta$  are interpolated linearly between energy bins in order to create continuous thresholding functions. The thresholds for the event multiplicity define a piecewise function which is constant between the edges of the energy bins. An event is then either accepted or rejected based on the value of the threshold functions at the event's estimated energy.

Figure 11.2 shows the angular resolution of the CTA analysis after applying the cuts from table 11.2. Unlike the figure 8.2 shown previously, this plots shows the performance on the

**Table 11.2:** The result of the grid search of the three-dimensional parameter space in minimum event multiplicity, on-region radius  $\theta$ , and prediction threshold  $\alpha$ . These values are used to select the events for the plots that compare the performance of this analysis to the reference analysis. As expected, the significance is lower in the very-high and very-low energy bins. Both due to limited statistics and inaccurate reconstruction of direction, particle type, and energy. The relative sensitivity is multiplied with the value of a flux model at the center of the energy range to produce a sensitivity curve. The shape of the curve changes depending on the shape of the target spectrum. I use the MAGIC log-parabolic spectrum, see table 11.1, to create the sensitivity curve. The last row is grayed out since the result does not match all required optimization constraints. One alternative approach to performing the event selection is to fix  $\theta$  to the value of the angular resolution as calculated on point-like simulations. An example is shown in A.6.

Energy Range / TeV		Multiplicity	$\theta/^\circ$	$\alpha$	Significance	Relative Sensitivity
0.02	0.03	4	0.010	0.70	7.1	$(6.91^{+1.12}_{-0.84}) 10^{-1}$
0.03	0.05	3	0.170	0.65	150.6	$(3.01^{+0.02}_{-0.02}) 10^{-2}$
0.05	0.08	3	0.170	0.70	257.6	$(1.58^{+0.01}_{-0.01}) 10^{-2}$
0.08	0.13	2	0.160	0.70	405.2	$(9.10^{+0.04}_{-0.04}) 10^{-3}$
0.13	0.20	3	0.110	0.75	428.7	$(4.97^{+0.04}_{-0.04}) 10^{-3}$
0.20	0.32	7	0.090	0.75	344.0	$(3.14^{+0.05}_{-0.06}) 10^{-3}$
0.32	0.50	8	0.070	0.80	290.1	$(2.16^{+0.08}_{-0.08}) 10^{-3}$
0.50	0.80	4	0.070	0.80	333.2	$(1.62^{+0.07}_{-0.07}) 10^{-3}$
0.80	1.26	8	0.060	0.80	230.5	$(1.49^{+0.11}_{-0.12}) 10^{-3}$
1.26	2.00	10	0.050	0.75	181.4	$(1.74^{+0.24}_{-0.22}) 10^{-3}$
2.00	3.17	9	0.050	0.75	155.0	$(1.76^{+0.28}_{-0.28}) 10^{-3}$
3.17	5.02	10	0.050	0.75	114.4	$(3.18^{+0.60}_{-0.47}) 10^{-3}$
5.02	7.96	7	0.070	0.75	98.9	$(3.40^{+0.52}_{-0.35}) 10^{-3}$
7.96	12.62	9	0.050	0.70	71.4	$(8.23^{+1.47}_{-1.57}) 10^{-3}$
12.62	20.00	6	0.050	0.70	53.0	$(1.19^{+0.22}_{-0.15}) 10^{-2}$
20.00	31.70	6	0.080	0.70	37.3	$(2.21^{+0.27}_{-0.16}) 10^{-2}$
31.70	50.24	5	0.060	0.65	26.1	$(4.39^{+0.84}_{-0.43}) 10^{-2}$
50.24	79.62	4	0.040	0.55	16.4	$(1.23^{+0.35}_{-0.26}) 10^{-1}$
79.62	126.19	10	0.040	0.40	7.1	$(6.03^{+2.74}_{-1.77}) 10^{-1}$
126.19	200.00	2	0.010	0.30	1.5	$(8.47^{+1.62}_{-5.65}) 10^0$

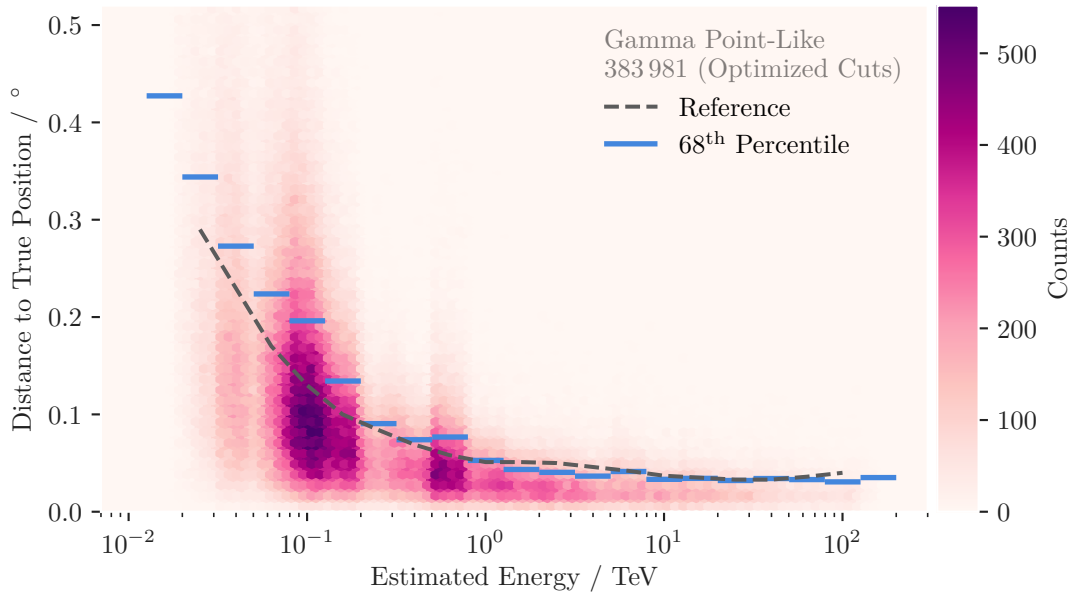
simulated point-like gamma-ray events. Beyond estimated energies of approximately 1 TeV, the angular resolution falls below  $0.05^\circ$  which is equivalent to the apparent radius of the Eagle nebula [118]. This low angular resolution allows CTA to map the TeV gamma-ray sky with unprecedented precision.

Similar to the angular resolution plot, figure 11.3 shows the energy resolution for the optimized event selection on point-like gamma rays. The resolution remains close to a value of 0.2 for the majority of the energy range. Like the reference analysis, the resolution improves with increasing energy and remains relatively constant until about 20 TeV. The bump near 20 TeV is due to outliers in the distribution which shift the percentile. In general, the energy regression seems to perform worse than the reference by at least a factor of two. This does not influence the point-source sensitivity to a large degree as the only consequence of this is the migration of events between energy bins in the curve. It might, however, have detrimental effects on the reconstruction of energy spectra.

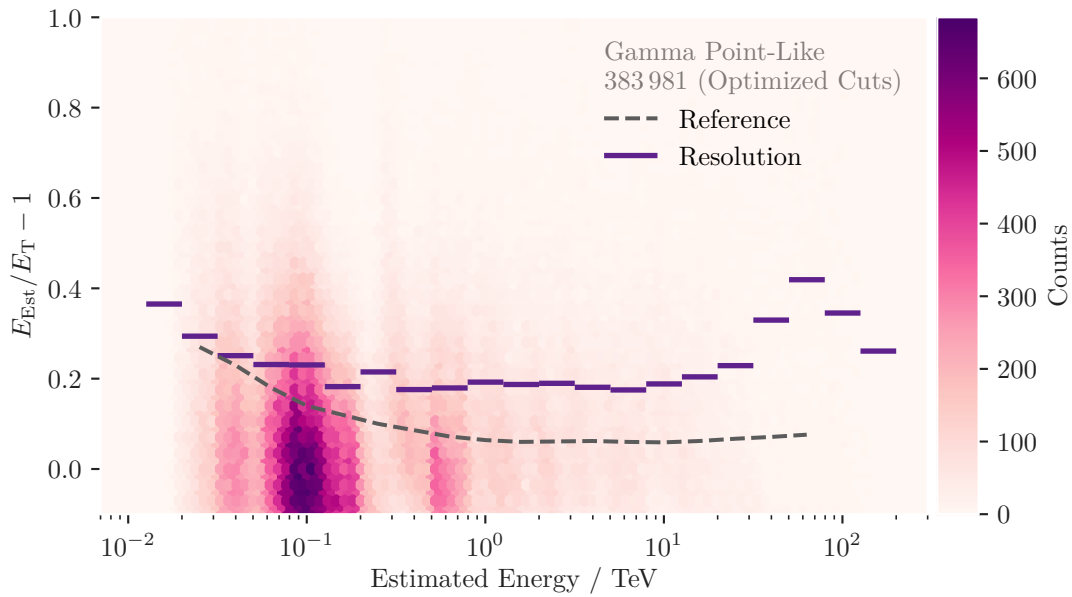
Figure 11.4 shows the effective area of this analysis compared to the reference analysis. As stated in section 5.2, the effective area is defined as the acceptance probability of the telescope multiplied by the maximum scatter area in the simulations. The curves match quite well in the lowest energy range. In the range above 1 TeV the effective area of the reference analysis is significantly larger. This might be due to the fact that I apply no “pre-selection” to the data whatsoever. These pre-selection cuts are traditionally used in IACT analysis to remove faint or non-typical events in a manual fashion. They are usually not well-motivated and often the result of subjective intuition of the researcher applying them. Hence, no manual selection takes place in the analysis presented here. The colors of the points indicate the prediction threshold that has been applied. The optimization chooses larger thresholds in the medium energy range. Towards the low and high-energy range separation between signal and background is not as effective and the chosen prediction threshold is lower.

Figure 11.5 shows the point source sensitivity curve for CTA analysis presented here. The gray line shows the official CTA requirement for the point source analyses. This result shows that the sensitivity requirements set by CTA can easily be met by the `ctapipe` based analysis in conjunction with the `aict-tools` package. In fact the sensitivity accurately matches the reference analysis and even outperforms it in some energy bins. The analysis presented here is currently the only competing alternative to the MARS and `Eventdisplay` analysis. All numbers and figures presented in this chapter are calculated by a package called `ctaplots` which is available at

[https://github.com/kbruegge/cta\\_performance\\_plots](https://github.com/kbruegge/cta_performance_plots) 

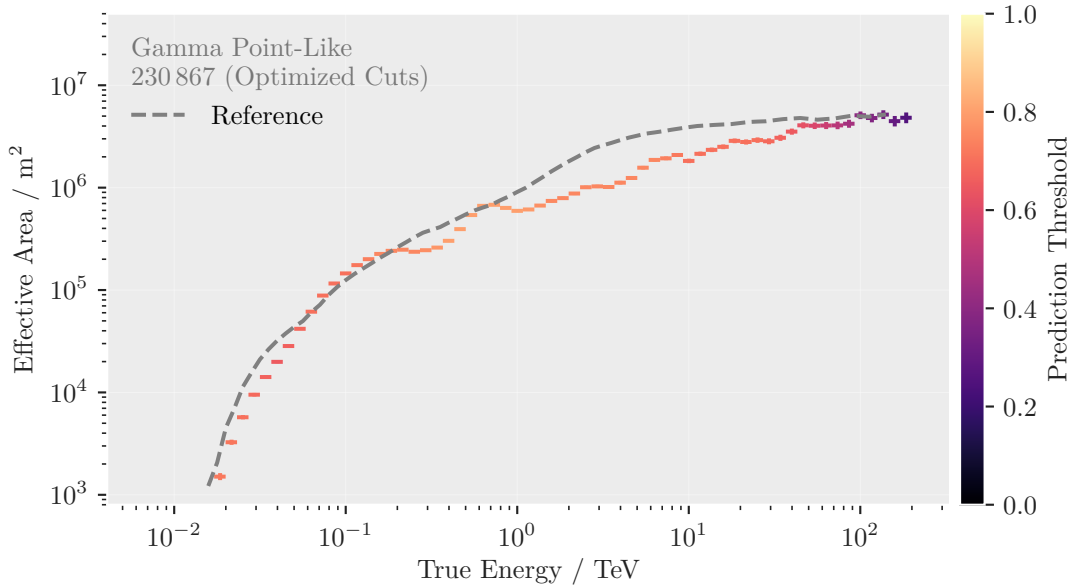


**Figure 11.2:** The distribution of distances between true and estimated source position. As discussed for figure 8.2 the angular resolution is defined as the 68<sup>th</sup> percentile of the distance between estimated and simulated source position. Unlike figure 8.2 the data here is plotted versus estimated energy to make it comparable to the reference curve, which is unfortunately not available in units of true event energy. Note that no cut in the distance to the true source position  $\theta$  has been applied.

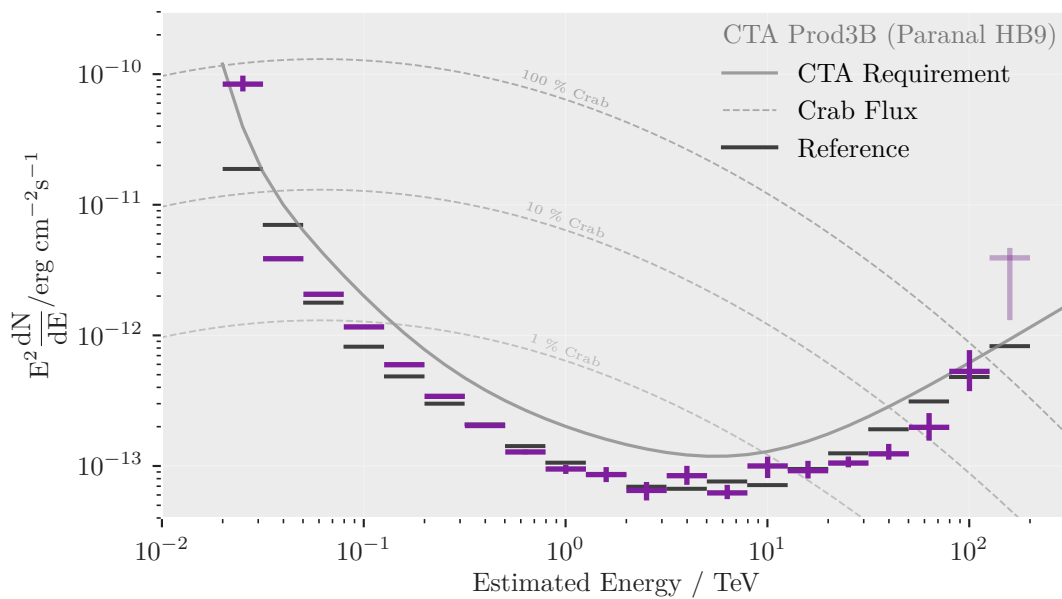


**Figure 11.3:** The distribution of the relative distances between true and estimated energy for point-like gamma-ray simulations. Unlike figure 9.4 the data here is plotted versus the estimated energy to make it comparable to the reference curve. The bump in curve near 50 TeV is due to the heavily tailed distribution of events in that range. The outliers might be due to the lack of image containment or similar effects. The median of the distribution, which is not shown here, does not exhibit this behavior and remains relatively flat. Note that no cut in the distance to the true source position  $\theta$  has been applied.

## 11 Sensitivity Computation



**Figure 11.4:** This figure shows the effective area *after* the optimized event selection has been applied. In conjunction with the definition of the reference curve, cuts in telescope multiplicity, prediction threshold, and direction are used. The color of each point shows the applied prediction thresholds as defined in table 11.2. The dashed gray line shows the result of the reference analysis. The additional cut in direction reduces the number of selected events by about half compared to figures 11.2 and 11.3. Its important to note that the reference analysis employs an additional smoothing operation to suppress noise which results in smoother curves.



**Figure 11.5:** The sensitivity curve of the ctape and aiCT-tools based analysis for the full southern layout at the Paranal observatory site. The sensitivity closely follows the reference shape indicated by the black bars. No error estimates are published for the reference results. The curved dashed lines show the flux of the Crab Nebula according to the log-parabolic spectrum as published by MAGIC [17]. The faded point on the very-high end of the energy range does not adhere to the full selection requirements.



# 12 Notes on Reproducibility

The computation of the optimized event selection is the final link in a long chain of analysis steps. Many terabytes of raw data are preprocessed into tabular data on which machine-learning algorithms can be trained. The data needs to be split into independent test and training sets to avoid biases. All meta data concerning the air-shower simulation has to be carried along so that the effective area can be calculated for any arbitrary subset of the data. The entire process is a composition of many different programs which are in principle self-contained. Without proper automatization, the whole construction is rather fragile, if not dangerously prone to error.

Each figure and table in this document has an implicit dependency on some input data. Table 11.2, for example, shows the results of the optimized event selection. The table can only be constructed if the machine-learning models described in chapter 9 have been applied to the test data which resulted from the preprocessing performed in chapter 8. The models themselves in turn depend on the configuration files and the training data. These data dependencies can be explicitly modeled with tools such as `make` [133] or `snakemake` [94]. I chose `make` because it is supported on essentially every operating system that is in use today. Data dependencies are described by so-called *Makefiles*. The `make` program builds a directed, acyclic graph from the *Makefile* and executes each step in topological order. This entire document can be build from a single call to the `make` program. Each machine-learning model, fit, figure, table, and automated  $\LaTeX$ snippet is part of the execution graph. Figure A.5 in the appendix shows a visual representation of the dependency graph for this document.

All software used for this thesis was written in the Python programming language in version 3.7.2. Each plot was created using the `matplotlib` [86] library version 3.1.0. A full list of Python dependencies can be found in the `requirements.txt` file that is attached in appendix B.2. The  $\LaTeX$ code for this document and all scripts needed to create the figures and tables will be uploaded after official publication to

<https://github.com/kbruegge/phd-thesis> 



# 13 Conclusion

The deployment of the Cherenkov Telescope Array project will ring in a new era of Cherenkov astronomy. Its unprecedented size brings new challenges to every aspect of traditional IACT analysis. This second part of my thesis shows that a fully open, reproducible and configurable analysis pipeline matches the performance of the previous CTA reference analysis.

The development of `ctapipe` as an open-source tool is a paradigm shift in the history of very-high-energy physics. The `ctapipe` project is a chance to bundle the expert knowledge from multiple telescope collaborations into a single place. I contributed to `ctapipe` implementation that performs the directional reconstruction of air showers. The resulting angular resolution as seen in figure 11.2 closely follows that of the reference analysis. To ensure reproducibility for the processing, I used the methods provided by `ctapipe` to implement a configurable pipeline which reads simulated CTA data and performs all steps necessary to perform background suppression and energy estimation. As of yet, no official data format has been agreed on to store the results produced by `ctapipe`. I used a column-based storage with unique identifiers on each row that allows me to perform database-like queries on the data. We developed the `aict-tools` package to handle the common machine-learning tasks encountered in IACT analysis. Efficient background suppression and energy estimation is maybe the most challenging part of any IACT analysis. In order to supply the multivariate methods with as much information as possible, the models were trained using per-telescope as well as array-wide features. The `aict-tools` support this by joining two tables before handing them to the model. For application to CTA data special care needs to be taken in order to ensure data consistency as the data can only be read in a batch-wise manner. The `aict-tools` perform remarkably well on CTA data as shown in figure 9.2.

The final benchmark for any CTA analysis is the sensitivity curve. The reference values published by CTA are computed with proprietary software from the MAGIC and VERITAS collaborations, for which no detailed documentation is available. I built an open-source toolkit to compute sensitivity curves and detection significances on a per-event basis. The resulting sensitivity curve is shown in figure 11.5. The image shows that the analysis I developed for my thesis performs just as well or event better than the reference.

### 13 Conclusion

The future development of `ctapipe` needs to concentrate on the definition of file formats so that an entire processing pipeline can be implemented. Once that is achieved, an official benchmark has to be defined in order to make different analyses comparable with each other. The performance of the `aict-tools` on CTA data could be improved in a future iteration by applying two nested models on the data: one on a per-telescope level, and one that summarizes the result of the single-telescope predictions for the entire array-event. A new production of CTA simulations has recently been started. This simulation has been adapted with data from the prototype telescopes that have been deployed during the last two years. CTA now has the chance to compute the next official performance numbers using a fully reproducible and open-source pipeline.

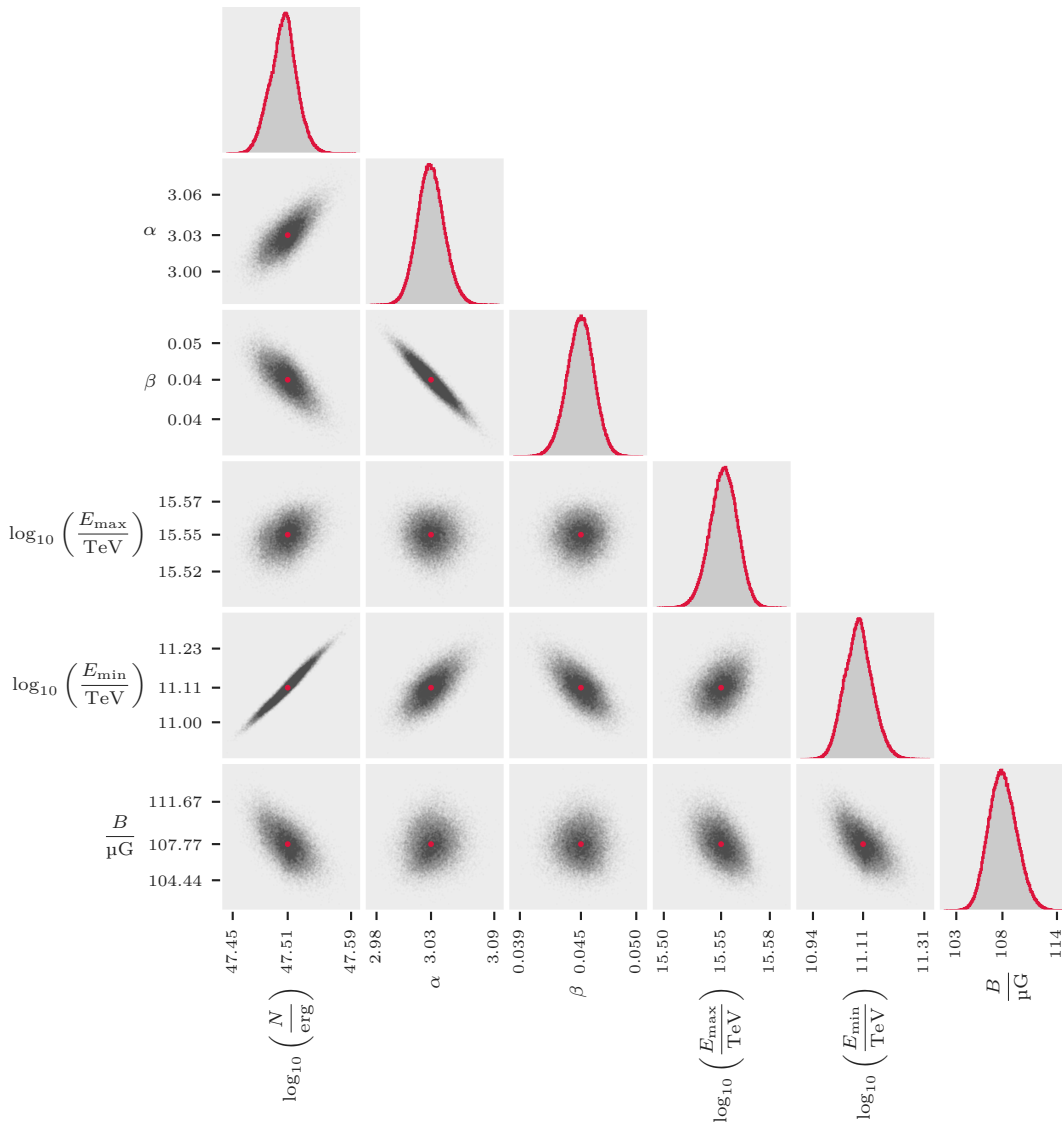
# A Additional Comments and Results

## A.1 Least Squares Fit to High Energy Crab Emission

In figure 2.3 I performed a least-squares fit to flux data for a log-parabolic energy spectrum with parameters  $\alpha$ ,  $\beta$ , and the amplitude  $A$ . The matrix below shows the covariance of the parameters as estimated by `scipy`'s `curve_fit` method. The `scipy` method uses the Levenberg-Marquardt [99] algorithm and estimates the covariance via the Hessian matrix in the minimum. The entries on the diagonal show the variance of the corresponding parameter.

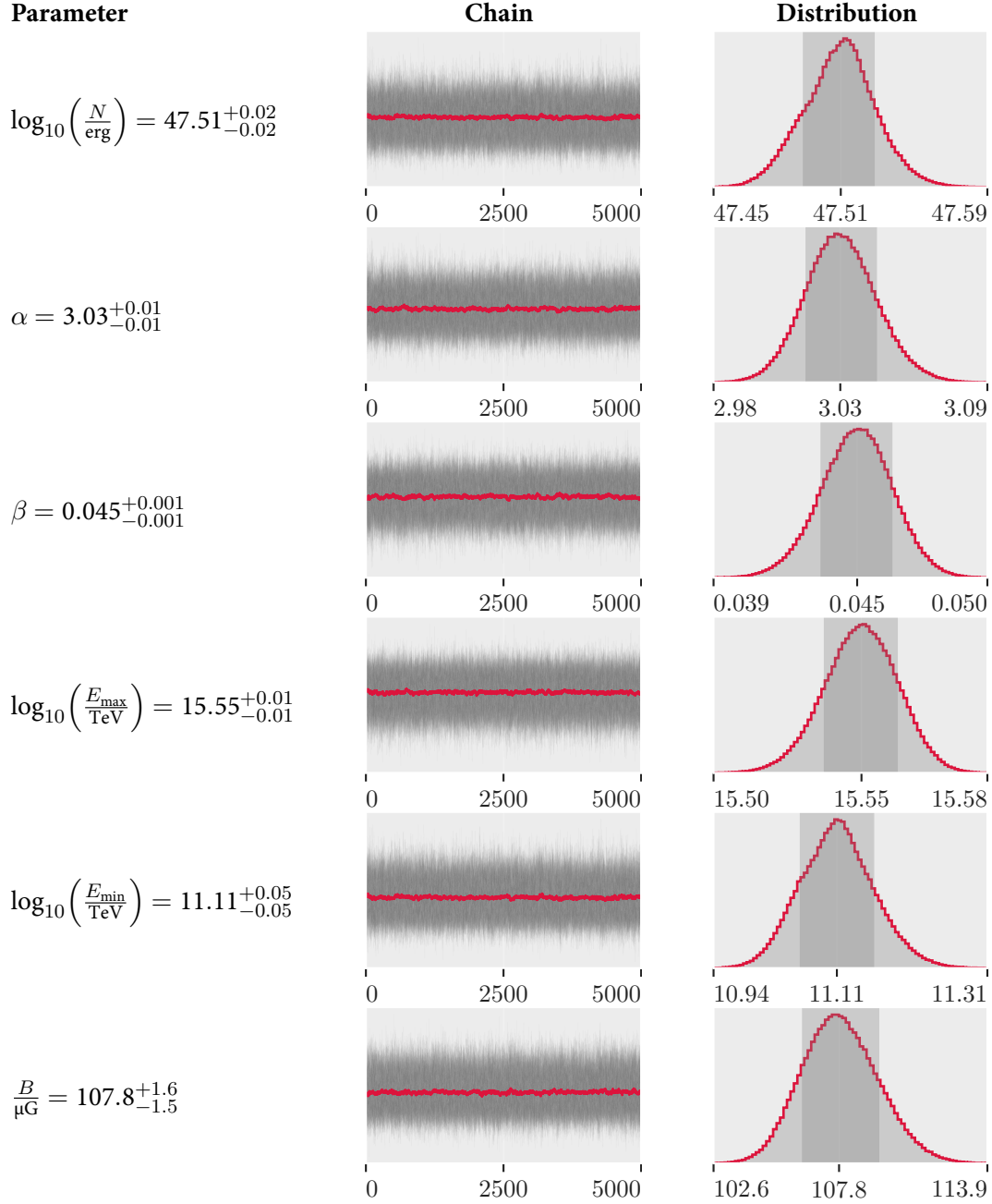
$$\begin{array}{c} A \\ \alpha \\ \beta \end{array} \begin{array}{ccc} A & \alpha & \beta \\ \left( \begin{array}{ccc} 1.819 \times 10^{-24} & -2.484 \times 10^{-14} & -5.267 \times 10^{-15} \\ -2.484 \times 10^{-14} & 1.169 \times 10^{-3} & 5.033 \times 10^{-4} \\ -5.267 \times 10^{-15} & 5.033 \times 10^{-4} & 2.783 \times 10^{-4} \end{array} \right) \end{array}$$

## A.2 SSC Fit to Crab Nebula Flux Data



**Figure A.1:** This figure shows the correlation between the sampled parameters for figure 4.1. The red dots indicate the median. Tick labels on the axis are placed at the 1<sup>th</sup>, 99<sup>th</sup> and 50<sup>th</sup> percentile. The histograms on the diagonal show the marginalized distributions of the parameters labeled on the lower axis. Some of these parameters show strong correlations, which need to be taken into account when interpreting the errors given in A.1

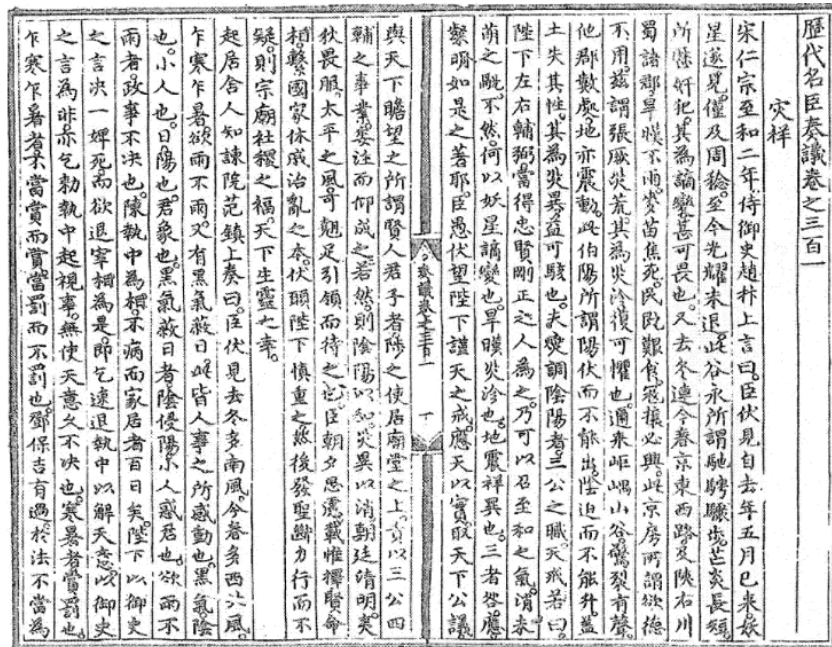
**Table A.1:** List of free parameters for the SSC model fitted to fluxes from the Crab Nebula. The left column shows the median sample values with their 16<sup>th</sup> and 84<sup>th</sup> percentiles. The center columns shows the parameter values for all sampled chains in gray and the median value of all chains in red. The rightmost column shows the marginalized posterior distribution of each parameter. The shaded gray area indicates the 16<sup>th</sup> and 84<sup>th</sup> percentiles.



### A.3 Historical Evidence of Crab Supernova

Chinese records provide relatively reliable descriptions of variable phenomena in the sky. Ancient Chinese astronomers speak of “Guest Stars” whenever such a variable star was observed. Compared to European records, these document are relatively reliable. The earliest records of Guest Stars in Chineses history date back to the Han dynasty approximately 200 B.C. [147]

The figure below shows the only known historical mention of the supernova that is now the Crab Nebula. A Chinese astronomer recorded the phenomenon in a letter to his emperor. The dates and description match that of a supernova consistent with the location of the Crab Nebula.



**Figure A.2:** Chinese records of the Lidai Mingchen Zouyi [145, page 535], which dates to 1414. The passage about the Crab Nebula supernova (SN 1054) as translated in [127]. The passage reads: “2<sup>nd</sup> year of the Zhihe reign period of Emperor Renzong of Song [1055]; Attendant Censor Zhao Bian submitted a letter saying: ‘Your servant considers that, since the 5th month of last year [when] the baleful star appeared, a full year has passed and until now its brilliance has not faded [lit. ‘retreated’]’. This is what Gu Yong meant by ‘its rapid movement, the variations in the length of its flaming rays, and the [asterisks] on which it has trespassed successively’, as a censorious anomaly it is greatly to be feared.”



## A.4 Profile Likelihood Solution

The full Poisson likelihood includes a nuisance parameter for each energy bin that describes the number of background counts. These nuisance parameters can be removed by “profiling” the likelihood. Solving (5.13) for  $\mu_b$  yields

$$\mu_b = \frac{N_{\text{off}}\alpha + N_{\text{on}}\alpha - \alpha\mu_s - \mu_s - \sqrt{K}}{2\alpha(\alpha + 1)} \quad (\text{A.1})$$

where

$$K = \alpha^2(N_{\text{off}}^2 + 2N_{\text{off}}(N_{\text{on}} + \mu_s) + N_{\text{on}}^2 - 2N_{\text{on}}\mu_s + \mu_s^2) + 2\alpha\mu_s((N_{\text{off}} - N_{\text{on}}) + \mu_s) + \mu_s^2.$$

This expression can be substituted into (5.11) resulting in the profile likelihood which has no dependence on  $\mu_b$  anymore. Special care has to be taken when implementing this formula for zero counts in the data. See [https://docs.gammapy.org/0.11/stats/fit\\_statistics.html](https://docs.gammapy.org/0.11/stats/fit_statistics.html) for some information about edge cases.

## A.5 Implementation for PyMC3 and Theano

The higher the sampling rate, the quicker the fit. Spending some time to optimize the code to integrate the spectral model brings a large speed improvement. The listing below shows how the trapezoidal operation is implemented in a vectorized way using the numpy [125] library.

```
# Define x and delta x. This can be pre-computed once.
d = np.tile(np.linspace(0, 1, num=num_nodes), num_bins).reshape(num_bins, -1)
xs = (d * bin_widths[:, None]) + bin_edges[0:-1, None]
delta_xs = np.diff(xs)

# Compute the integral of f using theano's symbolic sum operation.
y = f(xs, parameters)
integral = 0.5 * theano.tensor.sum((y[:, 0:-1] + y[:, 1:]) * delta_xs, axis=1)
```

This unassuming piece of code brings a dramatic speed increase of at least a factor of 10. Instead of building single scalar gradients for each energy bin, Theano can build the Jacobian for a single tensor object. The gradient of the integral with respect to the parameters  $N_0, \alpha, \beta$  can be build automatically. The PyMC3 model seems to sample faster when the gradients are precomputed. Since the integral limits are constant, we can apply Leibniz integral rule [62] and switch the order of the differential and the integration


$$\begin{aligned} \frac{\partial c}{\partial \alpha} &= \frac{\partial}{\partial \alpha} \int_{\Delta E_T} N(E; N_0, \alpha, \beta) dE = \int_{\Delta E_T} \frac{\partial}{\partial \alpha} N(E; N_0, \alpha, \beta) dE \\ &= \int_{\Delta E_T} \frac{\partial}{\partial \alpha} N_0 \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log_{10} \left( \frac{E}{E_0} \right)} dE \\ &= -N_0 \int_{\Delta E_T} \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log_{10} \left( \frac{E}{E_0} \right)} \log \left( \frac{E}{E_0} \right) dE. \end{aligned}$$

Equivalently for  $\beta$

$$\frac{\partial c}{\partial \beta} = - \int_{\Delta E_T} \frac{1}{\log(10)} \log^2 \left( \frac{E}{E_0} \right) \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log_{10} \left( \frac{E}{E_0} \right)} dE$$

and last but not least the amplitude parameter  $N_0$

$$\frac{\partial c}{\partial N_0} = \int_{\Delta E_T} \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log_{10} \left( \frac{E}{E_0} \right)} dE.$$

The PyMC3 model can now be sampled at a rate of several hundred samples per second. The listing below shows the definition of the PyMC3 model. The full code can be accessed at [https://github.com/tudo-astroparticlephysics/ll\\_experiments](https://github.com/tudo-astroparticlephysics/ll_experiments) .

```

amplitude = pm.HalfFlat('amplitude')
alpha = pm.HalfFlat('alpha')
beta = pm.HalfFlat('beta')

mu_s = forward_fold_log_parabola(integrator, amplitude, alpha, beta,
                                observations)

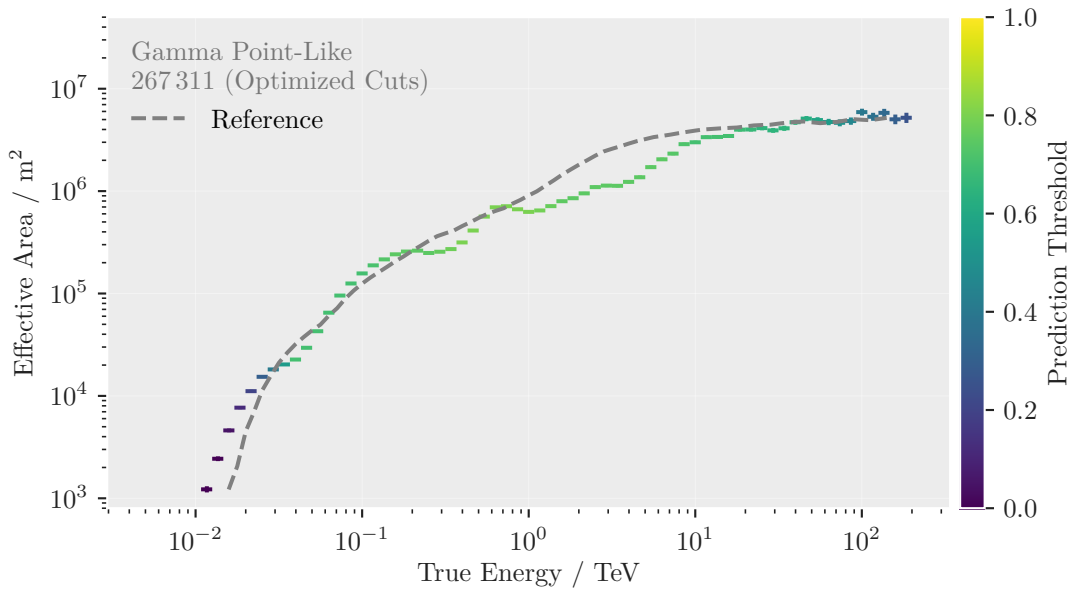
mu_b = pm.TruncatedNormal(
    'mu_b',
    lower=0,
    shape=len(off_data),
    mu=off_data,
    sd=5
)

pm.Poisson('background', mu=mu_b, observed=off_data, shape=len(off_data))
pm.Poisson(
    'signal',
    mu=mu_s + exposure_ratio * mu_b,
    observed=on_data,
    shape=len(on_data)
)

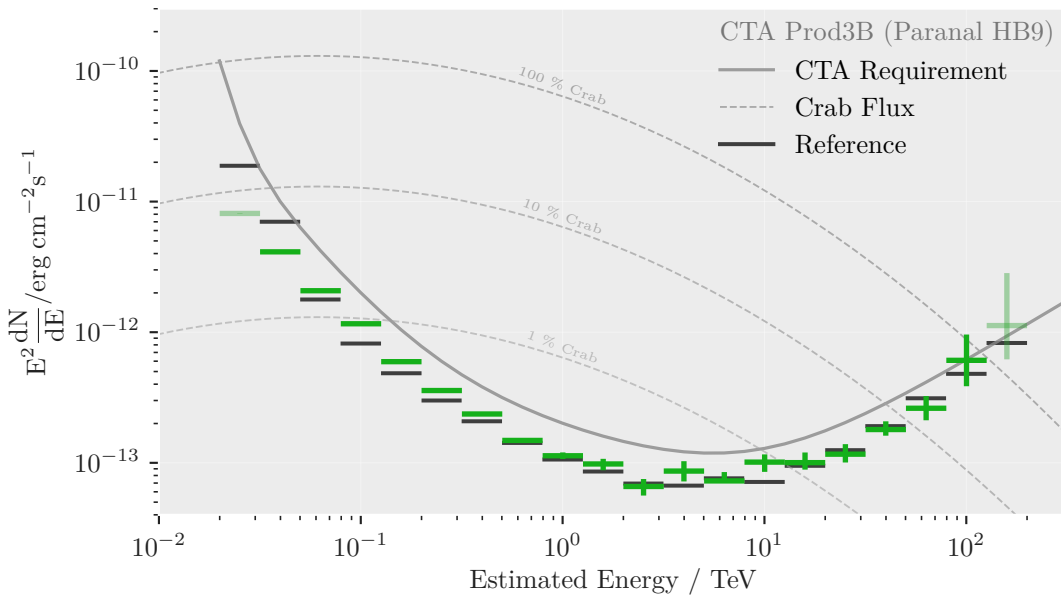
```

## A.6 Sensitivity and Effective Area for Fixed On Region

The sensitivity curve 11.5 and the effective area plot 11.4 were calculated on an optimized subset of the data. Combinations of prediction threshold, multiplicity, and on-region radius were tested to find the best sensitivity. This might introduce biases and *over-fitting* effects since no independent test data is available. To mitigate the effects to some degree, the size of the on-region,  $\theta$ , is defined in terms of angular resolution. This reduces the dimension of the optimization problem and increases runtime drastically.



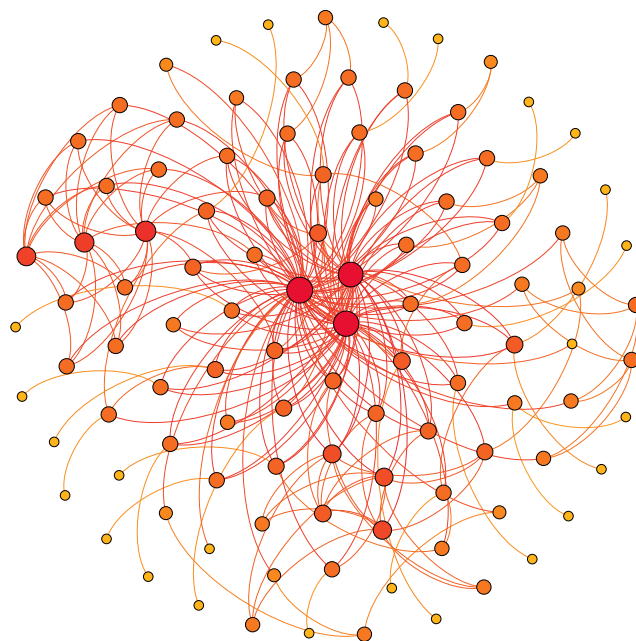
**Figure A.3:** Effective area for optimized event selection in multiplicity and prediction threshold. The on-region radius  $\theta$  was fixed to 50 % containment of the angular distance between true and estimated position in each energy bin. In comparison to the results shown in figure 11.4, the values match the reference curve better.



**Figure A.4:** Sensitivity curve for optimized event selection in multiplicity and prediction threshold. The on-region radius  $\theta$  was fixed to 50 % containment of the angular distance between true and estimated position in each energy bin. The results are slightly worse than the ones shown in figure 11.5, but still well within the expected performance range.

## A.7 Dependency Graph

The figure A.5 below shows the dependency of this document. It was created from the Makefile by the tool `makefile2graph`<sup>1</sup> written by Pierre Lindenbaum. The output was processed with the Gephi [27] software. Gephi is a graphical user interface to modify graphs for visualization. It supports various layout algorithms and search queries in large graphs. The figure below represents only a part of the dependency graph as one of the nodes calls into yet another Makefile recursively. Still, the figure summarizes most data dependencies needed for a proper CTA analysis. Managing this complicated construction by hand would be a herculean task. I urge anyone performing data analysis of any kind to use a workflow automation tool like `make` or `snakemake`.



**Figure A.5:** The dependency graph of this document as seen by `make`. The inner nodes correspond to the final targets. The final pdf file, which is this very document, has the largest degree of all nodes in this graph. Unfortunately, this figure itself cannot be created without human interaction and cannot be created fully automatically.

---

<sup>1</sup><https://github.com/lindenb/makefile2graph>



# B Configuration Files

## B.1 Configuration for the aict-tools

The listing below shows the contents of the configuration file used for the aict-tools. I'd like to take this opportunity and apologize for the terrible name of the aict-tools project. I was put on the spot by an approaching deadline and the creative fumes of the day were eluding me.

```
# seed for the random number generators,
# to make things reproducible
seed: 0
# telescope_type_key: telescope_type_name
telescope_events_key: telescope_events
array_events_key: array_events
runs_key: runs

multiple_telescopes: True

# config for the energy regression
energy:
  regressor : |
    ensemble.ExtraTreesRegressor(
      n_estimators=200,
      min_samples_split=100,
      n_jobs=-1,
    )
  log_target: True
  target_column: mc_energy
  n_cross_validations : 5
  # Define the name of the category you want to find.
  # It will be written as <class_name>_prediction
  class_name: gamma_energy
  # Define the name of the column that contains the
  # name of the telescope
  telescope_type_key: telescope_type_name
  features:
    - num_triggered_telescopes
    - width
    - length
    - skewness
    - kurtosis
    - intensity
    - camera_type_id
    - telescope_type_id
    - total_intensity
    - average_intensity
    - h_max
    - distance_to_reconstructed_core_position
    - num_triggered_lst
    - num_triggered_mst
    - num_triggered_sst
    - mirror_area
    - focal_length
    - leakage1_intensity
    - leakage2_intensity
    - leakage1_pixel
    - leakage2_pixel
    - concentration_pixel
    - concentration_core
    - concentration_cog
    - r
  feature_generation:
    needed_columns:
      - width
      - length
      - intensity
    features:
      area: width * length
      width_length: 1 - (width / length)
      log_size: log(intensity)
      log_size_area: log(intensity) / (width * length)

- width
- length
- intensity
features:
  area: width * length
  width_length: 1 - (width / length)
  log_size: log(intensity)
  log_size_area: log(intensity) / (width * length)

# config for the g/h separation
separator:
  classifier : |
    ensemble.ExtraTreesClassifier(
      n_estimators=200,
      min_samples_split=100,
      criterion='entropy',
      n_jobs=-1,
    )
  n_cross_validations : 5
  features:
    - num_triggered_telescopes
    - width
    - length
    - skewness
    - kurtosis
    - intensity
    - camera_type_id
    - telescope_type_id
    - total_intensity
    - average_intensity
    - h_max
    - distance_to_reconstructed_core_position
    - num_triggered_lst
    - num_triggered_mst
    - num_triggered_sst
    - mirror_area
    - focal_length
    - leakage1_intensity
    - leakage2_intensity
    - leakage1_pixel
    - leakage2_pixel
    - concentration_pixel
    - concentration_core
    - concentration_cog
    - r
  feature_generation:
    needed_columns:
      - width
      - length
      - intensity
    features:
      area: width * length
      width_length: 1 - (width / length)
      log_size: log(intensity)
      log_size_area: log(intensity) / (width * length)
```

## B.2 Python Requirements

I relied heavily on open-source software for every step presented in this thesis. The most important libraries are listed below in no particular order.

- `astropy` [22, 130]
- `numpy` [125]
- `pandas` [111]
- `scikit-learn` [129]
- `matplotlib` [86]
- `scipy` [126]

This document itself was built with MacTex-2018 on both macOS Mojave and macOS High Sierra. The listing below shows all the Python packages and their version numbers which were used to produce the results shown in this document, given that the Python installation is properly set

```
aict-tools==0.17.0
astropy==3.2.1
atomicwrites==1.3.0
attrs==19.1.0
bokeh==1.2.0
Click==7.0
colorama==0.4.1
corner==2.0.1
corsikaio==0.2.0
cta-plots==0.0.6
ctapipe==0.6.2.post173
ctapipe-extra==0.2.17
cyclcer==0.10.0
decorator==4.4.0
emcee==2.2.1
eventio==0.20.3
gammapy==0.10
h5py==2.9.0
iminuit==1.3.7
importlib-metadata==0.18
ipython-genutils==0.2.0
Jinja2==2.10.1
joblib==0.12.5
kiwisolver==1.1.0
llvmlite==0.29.0
MarkupSafe==1.1.1
matplotlib==3.1.0
mock==3.0.5
more-itertools==7.1.0
naima==0.8.3
numba==0.44.1
numexpr==2.6.9
numpy==1.16.4
packaging==19.0
pandas==0.24.2
patsy==0.5.1
peewee==3.9.6
Pillow==6.1.0
pluggy==0.12.0
psutil==5.6.3
py==1.8.0
pycrypto==2.6.1
pydot==1.4.1
pyfact==0.24.0
pymc-spectrum==0.1.0
pymc3==3.6
pymongo==3.8.0
PyMySQL==0.9.3
pyparsing==2.4.0
pytest==5.0.1
python-dateutil==2.8.0
pytz==2019.1
PyYAML==5.1.1
regions==0.4
ruamel.yaml==0.15.99
scikit-learn==0.20.3
scipy==1.3.0
seaborn==0.9.0
Shapely==1.6.4.post2
simple-crypt==4.1.7
six==1.12.0
SQLAlchemy==1.3.5
tables==3.5.2
Theano==1.0.4
tornado==6.0.3
tqdm==4.32.2
traitlets==4.3.2
wcwidth==0.1.7
wrapt==1.11.2
zipp==0.5.2
```



## B.3 Configuration for Preprocessing

The listing below shows the contents of the configuration file used for the preprocessing of the simulated CTA data. The list of telescope IDs selects the telescopes belonging to the so-called HB9 layout of the southern CTA site.

```

allowed_telescopes: [
  4, 5, 6, 11, 53, 54, 55, 56, 57, 60, 61, 64, 65,
  66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 88, 89, 90,
  91, 92, 93, 415, 416, 417, 418, 426, 427, 432, 433,
  438, 439, 440, 441, 442, 443, 448, 449, 450, 451, 458,
  459, 460, 461, 474, 475, 480, 481, 482, 483, 485, 486,
  500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510,
  511, 524, 525, 526, 527, 528, 529, 536, 537, 538, 539,
  540, 541, 542, 543, 544, 545, 550, 551, 552, 553, 554,
  555, 556, 557, 558, 559, 560, 561
]

names_to_id:
LSTCam: 1
NectarCam: 2
FlashCam: 3
DigiCam: 4
CHEC: 5

types_to_id:
LST: 1
MST: 2
SST: 3

integrator : NeighbourPeakIntegrator

cleaning_method: tailcuts_clean
cleaning_level:
  LSTCam: [3.5, 7.5, 2]
  NectarCam: [3, 5.5, 2]
  DigiCam: [2, 4.5, 2]

```



# Acknowledgements

My deepest gratitude goes to Prof. Dr. Dr. Wolfgang Rhode for giving me the opportunity to work in the field of astroparticle physics and for supervising this thesis. I also thank my second supervisor Prof. Dr. Kevin Kröniger for his interest in my research topic.

This work would never have been possible without the continuous support of my fellow colleagues and friends in the astroparticle physics group at Dortmund. It truly is a thoroughly pleasant environment to work in. Special thanks go out to Jens Buss, for essentially recruiting me and being the moral compass of the department, Lena Linhoff, for enduring my prolonged presence in her office, and Alexander Sandrock, for guiding me through horrible integral transforms. In particular, I'd like to thank my office mate and physics partner-in-crime Maximilian Nöthe. Thanks for teaching me innumerable things about Python,  $\LaTeX$  and programming in general, not to mention my appreciation for active-noise-canceling technology. I'd like to give special thanks to everyone who took on the daunting task of proofreading this document: Lena Linhoff, Maximilian Nöthe, and Linda Richerd.

Finally I can't help but acknowledge the Sonderforschungsbereich 876 for its funding and for giving me the opportunity to travel and meet all the terrific people that make up the CTA consortium. A special thanks to the SFB for allowing me to work at the CEA institute in Paris during the early stages of my graduate studies. Thanks to Thierry Stolarczyk for hosting me at CEA, Karl Kosack for many helpful discussions and supporting open science for CTA, and the entire group at CEA.

I owe additional gratitude to all my friends and family who did not get to see me much during the last few months. In particular I'd like to thank Hannah for putting up with my late-night antics and unusual dinner schedule.



# List of Figures

2.1	The all-particle cosmic-ray spectrum. . . . .	7
2.2	Schematic SED model . . . . .	13
2.3	Fit to the high-energy Crab emission . . . . .	15
3.1	Skymap of Fermi's 4FGL source catalog . . . . .	18
3.2	The H.E.S.S. galactic plane survey . . . . .	22
4.1	Fit of SSC model to Crab Nebula data . . . . .	28
4.2	SSC model with varying parameters . . . . .	29
5.1	Instrument response function for H.E.S.S. and FACT . . . . .	40
5.2	Depiction of wobble mod observations. . . . .	42
5.3	Measured counts in the open datasets. . . . .	50
5.4	Plot of the error function. . . . .	52
5.5	Results of the PyMC3 spectral fits . . . . .	55
5.6	Correlation between unfolded flux points. . . . .	59
5.7	Results of the PyMC3 unfolding . . . . .	60
5.8	Error propagation schematic explanation. . . . .	62
7.1	CTA layout at the northern and southern site . . . . .	69
7.2	Renderings of the individual CTA telescopes . . . . .	70
8.1	Preprocessing of CTA data . . . . .	77
8.2	Angular resolution for diffuse gamma rays . . . . .	83
8.3	Max height reconstruction accuracy. . . . .	84
8.4	Reconstruction of the impact distance . . . . .	84
9.1	Least squares machine learning example . . . . .	88
9.2	ROC curve and balanced accuracy. . . . .	97
9.3	Feature importance for the classifier . . . . .	98
9.4	Energy resolution . . . . .	99

*List of Figures*

9.5	Feature importance of the energy regressor . . . . .	100
10.1	Real-time analysis event rates . . . . .	103
11.1	$\theta^2$ square plot for the Crab Nebula . . . . .	112
11.2	Angular resolution with optimized event selection . . . . .	117
11.3	Energy resolution with optimized event selection . . . . .	117
11.4	Effective area of CTA analysis . . . . .	118
11.5	Sensitivity curve of the CTA analysis . . . . .	118
A.1	Correlation between fit parameters of the <code>na.ima</code> SED model. . . . .	124
A.2	Historical Chinese records about the Crab supernova . . . . .	126
A.3	Effective area for fixed values of $\theta$ . . . . .	130
A.4	Sensitivity curve for fixed values of $\theta$ . . . . .	130
A.5	Dependency Graph . . . . .	131

# List of Tables

5.1	Overview of the open data from MAGIC, H.E.S.S., VERITAS and FACT. . .	34
5.2	Results of the PyMC3 spectral fits . . . . .	54
5.3	Results of the PyMC3 unfolding . . . . .	61
9.1	Data structure for data level 2 . . . . .	95
11.1	Reference spectra for event weights . . . . .	108
11.2	Event selection optimization results. . . . .	115
A.1	Posterior distributions of the SED model parameters . . . . .	125





# Bibliography

1. A. Aab et al. “Searches for Large-scale Anisotropy in the Arrival Directions of Cosmic Rays Detected above Energy of  $10^{19}$  eV at the Pierre Auger Observatory and the Telescope Array”. *Astrophysical Journal* 794, 172, 2014, page 172.  
DOI: [10.1088/0004-637X/794/2/172](https://doi.org/10.1088/0004-637X/794/2/172). ARXIV: 1409.3128 [astro-ph.HE].  
Note: The error in the title of this was published as such.
2. M. G. Aartsen et al. (Fermi, MAGIC, AGILE, ASAS-SN, HAWC, H.E.S.S., INTEGRAL, Kanata, Kiso, Kapteyn, Liverpool Telescope, Subaru, Swift/NuSTAR, VERITAS, and VLA Collaboration). “Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922A”. *Science* 361:6398, 2018, eaat1378.  
DOI: [10.1126/science.aat1378](https://doi.org/10.1126/science.aat1378)
3. M. G. Aartsen et al. (IceCube Collaboration). “Measurement of the cosmic ray energy spectrum with IceTop-73”. *PhysRevD* 88, 042004, 2013, page 042004.  
DOI: [10.1103/PhysRevD.88.042004](https://doi.org/10.1103/PhysRevD.88.042004). ARXIV: 1307.3795 [astro-ph.HE]
4. R. U. Abbasi et al. “First Observation of the Greisen-Zatsepin-Kuzmin Suppression”. *PRL* 100, 101101, 2008, page 101101.  
DOI: [10.1103/PhysRevLett.100.101101](https://doi.org/10.1103/PhysRevLett.100.101101). ARXIV: astro-ph/0703099 [astro-ph]
5. B. P. Abbott et al. (LIGO and Virgo Collaboration). “Observation of Gravitational Waves from a Binary Black Hole Merger”. *Physical Review Letters D*. 116:6, 2016, page 061102.  
DOI: [10.1103/PhysRevLett.116.061102](https://doi.org/10.1103/PhysRevLett.116.061102). ARXIV: 1602.03837 [gr-qc]
6. A. A. Abdo et al. (Fermi Collaboration). “Fermi Large Area Telescope Observations of the Crab Pulsar And Nebula”. *Astrophysical Journal* 708, 2010, pages 1254–1267.  
DOI: [10.1088/0004-637X/708/2/1254](https://doi.org/10.1088/0004-637X/708/2/1254). ARXIV: 0911.2412 [astro-ph.HE]
7. A. A. Abdo et al. (Fermi Collaboration). “Measurement of the Cosmic Ray  $e^+ + e^-$  Spectrum from 20 GeV to 1 TeV with the Fermi Large Area Telescope”. *Physical Review Letters* 102, 18 2009, page 181101.  
DOI: [10.1103/PhysRevLett.102.181101](https://doi.org/10.1103/PhysRevLett.102.181101)
8. A. U. Abeysekara et al. “The 2HWC HAWC Observatory Gamma-Ray Catalog”. *Astrophysical Journal* 843:1, 40, 2017, page 40.  
DOI: [10.3847/1538-4357/aa7556](https://doi.org/10.3847/1538-4357/aa7556). ARXIV: 1702.02992 [astro-ph.HE]
9. V. A. Acciari et al. (MAGIC Collaboration). “A Fast Very High Energy  $\gamma$ -ray Flare from BL Lacertae during a Period of Multiwavelength activity in June 2015”. *Astronomy & Astrophysics*. 623, 2019, A175.  
DOI: [10.1051/0004-6361/201834010](https://doi.org/10.1051/0004-6361/201834010). ARXIV: 1901.01733 [astro-ph.HE]
10. V. A. Acciari et al. (MAGIC Collaboration). “Deep observations of the globular cluster M15 with the MAGIC telescopes”. *Monthly Notices of the Royal Astronomical Society* 484:2, 2019, pages 2876–2885.  
DOI: [10.1093/mnras/stz179](https://doi.org/10.1093/mnras/stz179). ARXIV: 1901.04367 [astro-ph.HE]

## Bibliography

11. A. Acharyya et al. “Monte Carlo studies for the optimisation of the Cherenkov Telescope Array layout”. *Astropart. Phys.* 111, 2019, pages 35–53.  
DOI: [10.1016/j.astropartphys.2019.04.001](https://doi.org/10.1016/j.astropartphys.2019.04.001). ARXIV: [1904.01426](https://arxiv.org/abs/1904.01426) [astro-ph.IM]
12. M. Ackermann et al. (Fermi Collaboration). “Fermi LAT observations of cosmic-ray electrons from 7 GeV to 1 TeV”. *Physical Review D* 82:9, 092004, 2010, page 092004.  
DOI: [10.1103/PhysRevD.82.092004](https://doi.org/10.1103/PhysRevD.82.092004). ARXIV: [1008.3999](https://arxiv.org/abs/1008.3999) [astro-ph.HE]
13. J. Adler and O. Öktem. “Solving ill-posed inverse problems using iterative deep neural networks”. *Inverse Problems* 33:12, 2017, page 124007.  
DOI: [10.1088/1361-6420/aa9581](https://doi.org/10.1088/1361-6420/aa9581)
14. F. Aharonian et al. “Observations of the Crab Nebula with HESS”. *Astronomy & Astrophysics* 457, 2006, pages 899–915.  
DOI: [10.1051/0004-6361:20065351](https://doi.org/10.1051/0004-6361:20065351). ARXIV: [astro-ph/0607333](https://arxiv.org/abs/astro-ph/0607333) [astro-ph]
15. F. Aharonian et al. “The Crab Nebula and Pulsar between 500 GeV and 80 TeV: Observations with the HEGRA Stereoscopic Air Cerenkov Telescopes”. *Astrophysical Journal* 614, 2004, pages 897–913.  
DOI: [10.1086/423931](https://doi.org/10.1086/423931). ARXIV: [astro-ph/0407118](https://arxiv.org/abs/astro-ph/0407118) [astro-ph]
16. M. L. Ahnen. “On the On-Off Problem: An Objective Bayesian Analysis”. In: *Proceedings of the 34<sup>th</sup> International Cosmic Ray Conference*. ICRC 2015. 2016, page 701.  
DOI: [10.22323/1.236.0701](https://doi.org/10.22323/1.236.0701)
17. J. Aleksić et al. (MAGIC Collaboration). “Measurement of the Crab Nebula spectrum over three decades in energy with the MAGIC telescopes”. *Journal of High Energy Astrophysics* 5-6, 2015, pages 30–38. ISSN: 2214-4048.  
DOI: [10.1016/j.jheap.2015.01.002](https://doi.org/10.1016/j.jheap.2015.01.002)
18. J. Aleksić et al. (MAGIC Collaboration). “The major upgrade of the MAGIC telescopes, Part I: The hardware improvements and the commissioning of the system”. *Astroparticle Physics* 72, 2016, pages 61–75.  
DOI: [10.1016/j.astropartphys.2015.04.004](https://doi.org/10.1016/j.astropartphys.2015.04.004). ARXIV: [1409.6073](https://arxiv.org/abs/1409.6073) [astro-ph.IM]
19. M. Amenomori et al. “The All-Particle Spectrum of Primary Cosmic Rays in the Wide Energy Range from  $10^{14}$  to  $10^{17}$  eV Observed with the Tibet-III Air-Shower Array”. *Astrophysical Journal* 678, 2008, pages 1165–1179.  
DOI: [10.1086/529514](https://doi.org/10.1086/529514). ARXIV: [0801.1803](https://arxiv.org/abs/0801.1803) [hep-ex]
20. H. Anderhub et al. (FACT Collaboration). “Design and operation of FACT – the first G-APD Cherenkov telescope”. *Journal of Instrumentation* 8:06, 2013, P06008–P06008.  
DOI: [10.1088/1748-0221/8/06/p06008](https://doi.org/10.1088/1748-0221/8/06/p06008)
21. K. O. Arras. *An Introduction To Error Propagation: Derivation, Meaning and Examples of Equation  $C_y = F_X C_X F_X^T$* . Technical report. École Polytechnique Fédérale de Lausanne, 1998.  
DOI: [10.3929/ethz-a-010113668](https://doi.org/10.3929/ethz-a-010113668)
22. Astropy Collaboration et al. “Astropy: A community Python package for astronomy”. *Astronomy & Astrophysics* 558, A33, 2013, A33.  
DOI: [10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068). ARXIV: [1307.6212](https://arxiv.org/abs/1307.6212) [astro-ph.IM]
23. A. M. Atoyan and F. A. Aharonian. “On the mechanisms of gamma radiation in the Crab Nebula”. *MNRAS* 278:2, 1996, pages 525–541.  
DOI: [10.1093/mnras/278.2.525](https://doi.org/10.1093/mnras/278.2.525)

24. W. B. Atwood, A. A. Abdo, et al. (Fermi Collaboration). “Fermi Large Area Telescope Fourth Source Catalog”. *arXiv e-prints*, 2019.  
ARXIV: 1902.10045 [astro-ph.HE].  
Note: Preliminary catalog publication. No journal publication yet.
25. W. B. Atwood et al. (Fermi Collaboration). “The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission”. *Astrophysical Journal* 697, 2009, pages 1071–1102.  
DOI: 10.1088/0004-637X/697/2/1071. ARXIV: 0902.1089 [astro-ph.IM]
26. D. Baack. *Data Reduction for CORSIKA*. Technical report. Astroparticle Department, TU Dortmund, 2016
27. M. Bastian, S. Heymann, and M. Jacomy. “Gephi: An Open Source Software for Exploring and Manipulating Networks”. In: *Proceedings of the International AAAI Conference on Weblogs and Social Media*. ICWSM 2009. San Jose, California, USA, 2009.  
DOI: 10.13140/2.1.1341.1520
28. O. Ben-Kiki, C. Evans, and I. Net. *YAML Specification v1.2*. 2019.  
<https://yaml.org/spec/1.2/spec.html>
29. M. Benning and M. Burger. “Modern regularization methods for inverse problems”. *Acta Numerica* 27, 2018, pages 1–111.  
DOI: 10.1017/S0962492918000016
30. D. Berge, S. Funk, and J. Hinton. “Background modelling in very-high-energy  $\gamma$ -ray astronomy”. *Astronomy & Astrophysics* 466:3, 2007, pages 1219–1229.  
DOI: 10.1051/0004-6361:20066674
31. K. Bernlöhr. “Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim\_telarray”. *Astroparticle Physics* 30:3, 2008, pages 149–158.  
DOI: 10.1016/j.astropartphys.2008.07.009
32. M. Bertina et al. “KASCADE-Grande energy spectrum of cosmic rays interpreted with post-LHC hadronic interaction models”. In: *Proceedings of the 34<sup>th</sup> International Cosmic Ray Conference*. Vol. 34. ICRC 2015. 2015, page 359.  
DOI: 10.22323/1.236.0359
33. Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. 2019.  
<http://www.blender.org>
34. G. R. Blumenthal and R. J. Gould. “Bremsstrahlung, Synchrotron Radiation, and Compton Scattering of High-Energy Electrons Traversing Dilute Gases”. *Reviews of Modern Physics* 42, 2 1970, pages 237–270.  
DOI: 10.1103/RevModPhys.42.237
35. B. M. Bolker. *Ecological models and data in R*. Princeton University Press, 2008.  
DOI: 10.2307/j.ctvcm4g37
36. L. Breiman. “Bagging predictors”. *Machine learning* 24:2, 1996, pages 123–140.  
DOI: 10.1023/A:1018054314350
37. L. Breiman. “Random forests”. *Machine learning* 45:1, 2001, pages 5–32.  
DOI: 10.1023/A:1010933404324

## Bibliography

38. B. Breisky. “When Victor Hess Discovered Cosmic Rays in a Hydrogen Balloon. On Its Centenary, Celebrating a Ride That Advanced Physics”. *NY Times*, 2012.  
<https://www.nytimes.com/2012/08/07/science/space/when-victor-hess-discovered-cosmic-rays-in-a-hydrogen-balloon.html>
39. K. Brügge, A. Egorov, C. Bockermann, K. Morik, and W. Rhode. “Distributed Real-Time Data Stream Analysis for CTA”. In: *Proceedings of the 27<sup>th</sup> conference on Astronomical Data Analysis Software & Systems*. ADASS XXVII. 2018.  
ARXIV: 1809.00581 [astro-ph.IM].  
Note: Proceedings to be published. Preprint published with permission.
40. R. Brun and F. Rademakers. “ROOT — An object oriented data analysis framework”. *Nuclear Instruments and Methods in Physics Research A* 389:1, 1997, pages 81–86.  
DOI: 10.1016/S0168-9002(97)00048-X
41. R. Bühler and R. Blandford. “The surprising Crab pulsar and its nebula: a review”. *Reports on Progress in Physics* 77:6, 2014, page 066901.  
DOI: 10.1088/0034-4885/77/6/066901
42. M. Bunse, N. Piatkowski, T. Ruhe, W. Rhode, and K. Morik. “Unification of Deconvolution Algorithms for Cherenkov Astronomy”. In: *5<sup>th</sup> International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2018, pages 21–30.  
Note: To be published.
43. P. Carbone et al. “Apache Flink™: Stream and Batch Processing in a Single Engine”. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 38, 2015, pages 28–38
44. D. Casadei. “Objective Bayesian Analysis of ON/OFF Measurements”. *The Astrophysical Journal* 798:1, 2014, page 5.  
DOI: 10.1088/0004-637x/798/1/5
45. H. Chen et al. “Low-dose CT via convolutional neural network”. *Biomed. Opt. Express* 8:2, 2017, pages 679–694.  
DOI: 10.1364/BOE.8.000679
46. J. Cortina and M. Teshima. “Status of the Cherenkov Telescope Array Large Size Telescopes”. In: *Proceedings of the 34<sup>th</sup> International Cosmic Ray Conference*. ICRC 2015. 2016, page 943.  
DOI: 10.22323/1.236.0943
47. W. D. Cotton, D. Tody, and W. D. Pence. “Binary table extension to FITS.” *Astronomy & Astrophysics Supplement* 113, 1995, page 159
48. CTA Project Office. *CTA Observatory Website*. CTA Observatory gGmbH. 2019.  
<https://www.cta-observatory.org/>.  
Note: IRF data can be downloaded at <https://www.cta-observatory.org/science/cta-performance/> // Current cost estimates are listed at <https://www.cta-observatory.org/project/industry/>
49. A. Daum et al. “First results on the performance of the HEGRA IACT array”. *Astroparticle Physics* 8, 1998, pages 1–11.  
DOI: 10.1016/S0927-6505(97)00031-5
50. M. DeGroot and M. Schervish. *Probability and Statistics*. Addison-Wesley, 2012. ISBN: 978-0-321-50046-5

51. C. Deil et al. “Gammapy - A prototype for the CTA science tools”. *International Cosmic Ray Conference* 301, 2017, page 766.  
DOI: [323/1.301.0766](https://doi.org/10.3233/1.301.0766). ARXIV: [1709.01751](https://arxiv.org/abs/1709.01751) [astro-ph.IM]
52. C. Deil et al. *Data formats for gamma-ray astronomy*. 2019.  
<https://gamma-astro-data-formats.readthedocs.io/en/latest/> visited on 05/14/2019
53. C. Deil et al. “Open high-level data formats and software for gamma-ray astronomy”. In: *6th International Symposium on High Energy Gamma-Ray Astronomy*. Vol. 1792. American Institute of Physics Conference Series. 2017, page 070006.  
DOI: [10.1063/1.4969003](https://doi.org/10.1063/1.4969003). ARXIV: [1610.01884](https://arxiv.org/abs/1610.01884) [astro-ph.IM]
54. A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. *Journal of the Royal Statistical Society: Series B (Methodological)* 39:1, 1977, pages 1–22
55. J. L. Devore. *Probability and Statistics for Engineering and the Sciences*. 8th. Brooks/Cole, 2011.  
DOI: [10.1002/9781119047063](https://doi.org/10.1002/9781119047063)
56. J. V. Dillon et al. “TensorFlow Distributions”. *CoRR*, 2017.  
ARXIV: [1711.10604](https://arxiv.org/abs/1711.10604)
57. E. Domingo-Santamariaa, J. Flixia, V. Scalzottob, W. Wittekc, and J. Ricoa (MAGIC Collaboration). “The DISP analysis method for point-like or extended  $\gamma$  source searches/studies with the MAGIC Telescope”. In: *Proceedings of the 29th International Cosmic Ray Conference*
58. D. Duellmann, K. Surdy, L. Menichetti, and R. Toebbicke. “Hadoop and friends - first experience at CERN with a new platform for high throughput analysis steps”. *Journal of Physics: Conference Series* 898, 2017, page 072034.  
DOI: [10.1088/1742-6596/898/7/072034](https://doi.org/10.1088/1742-6596/898/7/072034)
59. R. Engel et al. “Towards A Next Generation of CORSIKA: A Framework for the Simulation of Particle Cascades in Astroparticle Physics”. *Computing and Software for Big Science* 3:1, 2018, page 2. ISSN: 2510-2044.  
DOI: [10.1007/s41781-018-0013-0](https://doi.org/10.1007/s41781-018-0013-0)
60. F. Fenu and Pierre Auger Collaboration. “The cosmic ray energy spectrum measured using the Pierre Auger Observatory”. In: *Proceedings of the 35th International Cosmic Ray Conference*. Vol. 301. ICRC 2017. 2017, page 486
61. E. Fermi. “On the Origin of the Cosmic Radiation”. *Phys. Rev.* 75, 8 1949, pages 1169–1174.  
DOI: [10.1103/PhysRev.75.1169](https://doi.org/10.1103/PhysRev.75.1169)
62. H. Flanders. “Differentiation Under the Integral Sign”. *The American Mathematical Monthly* 80:6, 1973, pages 615–627. ISSN: 00029890, 19300972.  
<http://www.jstor.org/stable/2319163>
63. D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. “emcee: The MCMC Hammer”. *Publications of the Astronomical Society of the Pacific* 125:925, 2013, page 306.  
DOI: [10.1086/670067](https://doi.org/10.1086/670067). ARXIV: [1202.3665](https://arxiv.org/abs/1202.3665) [astro-ph.IM]
64. M. Fouka and S. Ouichaoui. “External Inverse Compton Spectra for Monoenergetic and Black-body Photon Fields Upscattered by a Power-Law Electron Distribution With a Finite Energy Range”. *Astrophysical Journal* 737:2, 2011, page 84.  
DOI: [10.1088/0004-637x/737/2/84](https://doi.org/10.1088/0004-637x/737/2/84)

## Bibliography

65. P. Freeman, S. Doe, and A. Siemiginowska. “Sherpa: a mission-independent data analysis application”. In: *Astronomical Data Analysis*. Vol. 4477. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. 2001, pages 76–87.  
DOI: [10.1117/12.447161](https://doi.org/10.1117/12.447161). ARXIV: [astro-ph/0108426](https://arxiv.org/abs/astro-ph/0108426)
66. S. Funk. “Ground- and Space-Based Gamma-Ray Astronomy”. *Annual Review of Nuclear and Particle Science* 65, 2015, pages 245–277.  
DOI: [10.1146/annurev-nucl-102014-022036](https://doi.org/10.1146/annurev-nucl-102014-022036)
67. T. K. Gaisser, R. Engel, and E. Resconi. *Cosmic rays and particle physics*. Cambridge University Press, 2016.  
DOI: [10.1017/CBO9781139192194](https://doi.org/10.1017/CBO9781139192194)
68. P. Geurts, D. Ernst, and L. Wehenkel. “Extremely randomized trees”. *Machine Learning* 63:1, 2006, pages 3–42. ISSN: 1573-0565.  
DOI: [10.1007/s10994-006-6226-1](https://doi.org/10.1007/s10994-006-6226-1)
69. J. T. Gosling, J. R. Asbridge, S. J. Bame, and W. C. Feldman. “Ion acceleration at the earth’s bow shock: A review of observations in the upstream region”. *AIP Conference Proceedings* 56:1, 1979, pages 81–99.  
DOI: [10.1063/1.32069](https://doi.org/10.1063/1.32069)
70. I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. 7th ed. Academic press, 2007. ISBN: 0-12-373637-4
71. P. K. F. Grieder. *Extensive Air Showers: High Energy Phenomena and Astrophysical Aspects - A Tutorial, Reference Manual and Data Book*. 2010.  
DOI: [10.1007/978-3-540-76941-5](https://doi.org/10.1007/978-3-540-76941-5)
72. A. Gunes Baydin, B. A. Pearlmutter, A. Andreyevich Radul, and J. M. Siskind. “Automatic differentiation in machine learning: a survey”. *Journal of Machine Learning Research* 18:1, 2017, pages 5595–5637. ISSN: 1532-4435.  
ARXIV: [1502.05767](https://arxiv.org/abs/1502.05767) [cs.SC]
73. H. E. S. S. Collaboration et al. “The H.E.S.S. Galactic plane survey”. *Astronomy & Astrophysics* 612, A1, 2018, A1.  
DOI: [10.1051/0004-6361/201732098](https://doi.org/10.1051/0004-6361/201732098). ARXIV: [1804.02432](https://arxiv.org/abs/1804.02432) [astro-ph.HE]
74. D. J. Hand. “Measuring classifier performance: a coherent alternative to the area under the ROC curve”. *Machine Learning* 77:1, 2009, pages 103–123.  
DOI: [10.1007/s10994-009-5119-5](https://doi.org/10.1007/s10994-009-5119-5)
75. T. Hassan et al. “Monte Carlo performance studies for the site selection of the Cherenkov Telescope Array”. *Astroparticle Physics* 93, 2017, pages 76–85. ISSN: 0927-6505.  
DOI: <https://doi.org/10.1016/j.astropartphys.2017.05.001>
76. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd ed. Springer Series in Statistics. Springer, New York, NY, USA, 2009.  
DOI: [10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7)
77. W. Hastings. “Monte Carlo Sampling Methods using Markov Chains and their Applications”. *Biometrika* 57:1, 1970, pages 97–109.  
DOI: [10.1093/biomet/57.1.97](https://doi.org/10.1093/biomet/57.1.97)

78. M. Haverkorn. “Magnetic Fields in the Milky Way”. In: *Magnetic Fields in Diffuse Media*. Springer Berlin Heidelberg, 2015, pages 483–506.  
doi: [10.1007/978-3-662-44625-6\\_17](https://doi.org/10.1007/978-3-662-44625-6_17)
79. D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw. *CORSIKA: A Monte Carlo code to simulate extensive air showers*. Technical report. Karlsruhe Institute of Technology, Department of Nuclear Physics, 1998
80. V. F. Hess. “Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten”. *Physikalische Zeitschrift* 13, 1912, pages 1084–1091
81. J. J. Hester. “The Crab Nebula: an astrophysical chimera”. *Annual Reviews in Astronomy and Astrophysics* 46, 2008, pages 127–155.  
doi: [10.1146/annurev.astro.45.051806.110608](https://doi.org/10.1146/annurev.astro.45.051806.110608)
82. A. M. Hillas. “Cerenkov Light Images of EAS Produced by Primary Gamma Rays and by Nuclei”. In: *Proceedings of the 19<sup>th</sup> International Cosmic Ray Conference*. Vol. 3. ICRC 1985. 1985, page 445
83. J. A. Hinton and the HESS Collaboration. “The status of the HESS project”. *New Astronomy Reviews* 48:5-6, 2004, pages 331–337.  
doi: [10.1016/j.newar.2003.12.004](https://doi.org/10.1016/j.newar.2003.12.004). ARXIV: [astro-ph/0403052](https://arxiv.org/abs/astro-ph/0403052) [astro-ph]
84. M. Hoffman and A. Gelman. “The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” *Journal of Machine Learning Research* 15:1, 2014, pages 1593–1623.  
<http://www.jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf>
85. J. Holder et al. “The first VERITAS telescope”. *Astroparticle Physics* 25:6, 2006, pages 391–401.  
doi: [10.1016/j.astropartphys.2006.04.002](https://doi.org/10.1016/j.astropartphys.2006.04.002). ARXIV: [astro-ph/0604119](https://arxiv.org/abs/astro-ph/0604119) [astro-ph]
86. J. D. Hunter. “Matplotlib: A 2D graphics environment”. *Computing in Science & Engineering* 9:3, 2007, pages 90–95.  
doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
87. L. Hyafil and R. L. Rivest. “Constructing optimal binary decision trees is NP-complete”. *Information Processing Letters* 5:1, 1976, pages 15–17.  
doi: [10.1016/0020-0190\(76\)90095-8](https://doi.org/10.1016/0020-0190(76)90095-8)
88. K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. “Deep convolutional neural network for inverse problems in imaging”. *IEEE Transactions on Image Processing* 26:9, 2017, pages 4509–4522.  
doi: [10.1109/TIP.2017.2713099](https://doi.org/10.1109/TIP.2017.2713099)
89. E. Jourdain and J. P. Roques. “The High-Energy Emission of the Crab Nebula From 20 keV to 6 MeV with INTEGRAL SPI”. *The Astrophysical Journal* 704:1, 2009, pages 17–24.  
doi: [10.1088/0004-637x/704/1/17](https://doi.org/10.1088/0004-637x/704/1/17)
90. K.-H. Kampert and A. A. Watson. “Extensive air showers and ultra high-energy cosmic rays: a historical review”. *European Physical Journal H* 37:3, 2012, pages 359–412.  
doi: [10.1140/epjh/e2012-30013-x](https://doi.org/10.1140/epjh/e2012-30013-x). ARXIV: [1207.4827](https://arxiv.org/abs/1207.4827)
91. N. S. Kardashev. “Nonstationarity of Spectra of Young Sources of Nonthermal Radio Emission”. *Soviet Astronomy* 6, 1962, page 317
92. J. King. “Hochenergetische Gammastrahlung aus dem Galaktischen Zentrum”. PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2018



## Bibliography

93. R. W. Klebesadel, I. B. Strong, and R. A. Olson. "Observations of Gamma-Ray Bursts of Cosmic Origin". *Astrophysical Journal* 182, 1973, page L85.  
DOI: [10.1086/181225](https://doi.org/10.1086/181225)
94. J. Köster and S. Rahmann. "Snakemake. A Scalable Bioinformatics Workflow Engine". *Bioinformatics* 28:19, 2012, pages 2520–2522. ISSN: 1367-4803.  
DOI: [10.1093/bioinformatics/bts480](https://doi.org/10.1093/bioinformatics/bts480).  
<https://doi.org/10.1093/bioinformatics/bts480>
95. W. Kraushaar et al. "Explorer XI Experiment on Cosmic Gamma Rays." *Astrophysical Journal* 141, 1965, page 845.  
DOI: [10.1086/148179](https://doi.org/10.1086/148179)
96. L. Kuiper et al. "The Crab pulsar in the 0.75-30 MeV range as seen by CGRO COMPTEL. A coherent high-energy picture from soft X-rays up to high-energy gamma-rays". *Astronomy & Astrophysics* 378, 2001, pages 918–935.  
DOI: [10.1051/0004-6361:20011256](https://doi.org/10.1051/0004-6361:20011256). ARXIV: [astro-ph/0109200](https://arxiv.org/abs/astro-ph/0109200) [astro-ph]
97. S. Kulkarni et al. "Twitter Heron: Stream Processing at Scale". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD '15. ACM, Melbourne, Victoria, Australia, 2015, pages 239–250.  
DOI: [10.1145/2723372.2742788](https://doi.org/10.1145/2723372.2742788)
98. E. Lefa, S. R. Kelner, and F. A. Aharonian. "On the spectral shape of radiation due to Inverse Compton Scattering close to the maximum cut-off". *Astrophysical Journal* 753:2, 2012, page 176.  
DOI: [10.1088/0004-637x/753/2/176](https://doi.org/10.1088/0004-637x/753/2/176)
99. K. Levenberg. "A method for the solution of certain non-linear problems in least squares". *Quarterly of Applied Mathematics* 2:2, 1944, pages 164–168.  
DOI: [10.1090/qam/10666](https://doi.org/10.1090/qam/10666)
100. D. A. Levin and Y. Peres. *Markov chains and mixing times*. Vol. 107. American Mathematical Society, 2017.  
DOI: [10.1090/mbk/107](https://doi.org/10.1090/mbk/107)
101. T.-P. Li and Y.-Q. Ma. "Analysis methods for results in gamma-ray astronomy". *Astroparticle Journal* 272, 1983, pages 317–324.  
DOI: [10.1086/161295](https://doi.org/10.1086/161295)
102. S. Linnainmaa. "Taylor expansion of the accumulated rounding error". *BIT Numerical Mathematics* 16:2, 1976, pages 146–160. ISSN: 1572-9125.  
DOI: [10.1007/BF01931367](https://doi.org/10.1007/BF01931367)
103. M. S. Longair. *High-Energy Astrophysics*. Cambridge university press, 2011. ISBN: 978-0-521-75618-1
104. D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN: 978-0-521-64298-9
105. G. Maier, J. Bregeon, T. Hassan, A. Moralejo, et al. *Discussions about a CTA Benchmark*. Technical report. 2019.  
<https://forge.in2p3.fr/projects/benchmarks-reference-analysis/wiki>.  
Note: There is no official document yet. One is being drafted by the CTA ASWG. The link above can only be accessed by CTA members.



106. G. Maier et al. *Description of CTA Instrument Response Functions*. Technical report. 2017.  
[https://forge.in2p3.fr/projects/cta\\_analysis-and-simulations/repository/show/DOC/InternalReports/IRFReports/released](https://forge.in2p3.fr/projects/cta_analysis-and-simulations/repository/show/DOC/InternalReports/IRFReports/released).  
 Note: This internal CTA report is not accessible to the general public.
107. G. Maier and J. Holder. “Eventdisplay: An Analysis and Reconstruction Package for Ground-based Gamma-ray Astronomy”. In: *Proceedings of the 35<sup>th</sup> International Cosmic Ray Conference*. ICRC 2017. 2018, page 747.  
 DOI: [10.22323/1.301.0747](https://doi.org/10.22323/1.301.0747). ARXIV: [1708.04048](https://arxiv.org/abs/1708.04048)
108. G. Marsaglia. “Generating a Variable from the Tail of the Normal Distribution”. *Technometrics* 6:1, 1964, pages 101–102.  
 DOI: [10.1080/00401706.1964.10490150](https://doi.org/10.1080/00401706.1964.10490150)
109. Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.  
<https://www.tensorflow.org/>
110. E. Massaro, M. Perri, P. Giommi, and R. Nesci. “Log-parabolic spectra and particle acceleration in the BL Lac object Mkn 421: Spectral analysis of the complete BeppoSAX wide band X-ray dataset”. *Astronomy & Astrophysics* 413, 2004, pages 489–503.  
 DOI: [10.1051/0004-6361:20031558](https://doi.org/10.1051/0004-6361:20031558). ARXIV: [astro-ph/0312260](https://arxiv.org/abs/astro-ph/0312260) [astro-ph]
111. W. McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. 2010, pages 51–56
112. C. Meegan et al. (Fermi GBM Collaboration). “The Fermi Gamma-ray Burst Monitor”. *Astrophysical Journal* 702, 2009, pages 791–804.  
 DOI: [10.1088/0004-637X/702/1/791](https://doi.org/10.1088/0004-637X/702/1/791). ARXIV: [0908.0450](https://arxiv.org/abs/0908.0450) [astro-ph.IM]
113. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. “Equation of State Calculations by Fast Computing Machines”. *Journal of Chemical Physics* 21:6, 1953, pages 1087–1092.  
 DOI: [10.1063/1.1699114](https://doi.org/10.1063/1.1699114)
114. M. Meyer, D. Horns, and H. S. Zechlin. “The Crab Nebula as a standard candle in very high-energy astrophysics”. *Astronomy & Astrophysics* 523, A2, 2010.  
 DOI: [10.1051/0004-6361/201014108](https://doi.org/10.1051/0004-6361/201014108). ARXIV: [1008.4524](https://arxiv.org/abs/1008.4524) [astro-ph.HE]
115. R. A. Millikan and G. H. Cameron. “The Origin of the Cosmic Rays”. *Physical Review* 32, 4 1928, pages 533–557.  
 DOI: [10.1103/PhysRev.32.533](https://doi.org/10.1103/PhysRev.32.533)
116. L. Möllerherm. “Verbesserung der Energieregression bei CTA”. Master’s Thesis. Astroparticle Department, TU Dortmund, 2018
117. A. Moralejo et al. (MAGIC Collaboration). “MARS, the MAGIC Analysis and Reconstruction Software”. In: *Proceedings of the 31<sup>st</sup> International Cosmic Ray Conference*. ICRC 2009. 2009.  
 ARXIV: [0907.0943](https://arxiv.org/abs/0907.0943) [astro-ph.IM]
118. E. F. E. Morales, F. Wyrowski, F. Schuller, and K. M. Menten. “Stellar clusters in the inner Galaxy and their correlation with cold dust emission”. *Astronomy & Astrophysics* 560, 2013, A76.  
 DOI: [10.1051/0004-6361/201321626](https://doi.org/10.1051/0004-6361/201321626). ARXIV: [1310.2612](https://arxiv.org/abs/1310.2612) [astro-ph.GA]
119. S. A. Mueller. “Cherenkov-Plenoscope”. PhD thesis. ETH Zürich, 2019.  
 DOI: [10.3929/ethz-b-000337911](https://doi.org/10.3929/ethz-b-000337911)

## Bibliography

120. J. Nickolls, I. Buck, M. Garland, and K. Skadron. “Scalable Parallel Programming with CUDA”. *Queue – GPU Computing* 6:2, 2008, pages 40–53. ISSN: 1542-7730.  
DOI: [10.1145/1365490.1365500](https://doi.org/10.1145/1365490.1365500)
121. A. Niculescu-Mizil and R. Caruana. “Predicting Good Probabilities with Supervised Learning”. In: *Proceedings of the 22<sup>nd</sup> International Conference on Machine Learning*. ICML '05. ACM, Bonn, Germany, 2005, pages 625–632.  
DOI: [10.1145/1102351.1102430](https://doi.org/10.1145/1102351.1102430)
122. C. Nigro et al. “Towards open and reproducible multi-instrument analysis in gamma-ray astronomy”. *Astronomy & Astrophysics* 625, 2019.  
DOI: [10.1051/0004-6361/201834938](https://doi.org/10.1051/0004-6361/201834938). ARXIV: [1903.06621](https://arxiv.org/abs/1903.06621) [astro-ph.HE]
123. D. Nosek and J. Nosková. “On Bayesian analysis of on–off measurements”. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 820, 2016, pages 23–33. ISSN: 0168-9002.  
DOI: [10.1016/j.nima.2016.02.094](https://doi.org/10.1016/j.nima.2016.02.094)
124. M. Nöthe, D. Neise, and S. A. Mueller (FACT Collaboration). “Towards Robotic Operation with the First G-APD Cherenkov Telescope”. *arXiv e-prints*, 2018.  
ARXIV: [1806.01542](https://arxiv.org/abs/1806.01542) [astro-ph.IM]
125. T. Oliphant. *NumPy: A guide to NumPy*. USA: Trelgol Publishing, 2006.  
<http://www.numpy.org/>
126. T. E. Oliphant. “Python for scientific computing”. *Computing in Science & Engineering* 9:3, 2007, pages 10–20
127. D. W. Pankenier. “Notes on translations of the East Asian records relating to the supernova of AD 1054”. *Journal of Astronomical History and Heritage* 9, 2006, pages 77–82
128. A. Paszke et al. “Automatic Differentiation in PyTorch”. In: *Autodiff Workshop at the 31st Conference on Neural Information Processing Systems*. NIPS 2017. Long Beach, CA, USA, 2017
129. F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research* 12, 2011, pages 2825–2830.  
ARXIV: [1201.0490](https://arxiv.org/abs/1201.0490)
130. A. M. Price-Whelan et al. “The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package”. *The Astrophysical Journal* 156, 123, 2018, page 123.  
DOI: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f)
131. J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. “Probabilistic programming in Python using PyMC3”. *PeerJ Computer Science* 2, 2016.  
DOI: [10.7717/peerj-cs.55](https://doi.org/10.7717/peerj-cs.55)
132. T. Sanuki et al. “Precise Measurement of Cosmic-Ray Proton and Helium Spectra with the BESS Spectrometer”. *Astrophysical Journal* 545:2, 2000, pages 1135–1142.  
DOI: [10.1086/317873](https://doi.org/10.1086/317873). ARXIV: [astro-ph/0002481](https://arxiv.org/abs/astro-ph/0002481) [astro-ph]
133. R. M. Stallman, R. McGrath, and P. D. Smith. *GNU Make: A Program for Directing Recompilation, for Version 3.81*. Free Software Foundation, 2004. ISBN: 978-1-882114-83-2
134. The CTA Consortium. “Science with the Cherenkov Telescope Array”. *World Scientific*, 2019.  
DOI: [10.1142/10986](https://doi.org/10.1142/10986)

135. The HDF Group. *Hierarchical Data Format, version 5*. 2019.  
<http://www.hdfgroup.org/HDF5/>
136. The IAU FITS Working Group. *Definition of the Flexible Image Transport System*. NASA, 2018.  
[https://fits.gsfc.nasa.gov/fits\\_standard.html](https://fits.gsfc.nasa.gov/fits_standard.html)
137. *The Nobel Prize in Physics 1958*.  
[http://nobelprize.org/nobel\\_prizes/physics/laureates/1958/index.html](http://nobelprize.org/nobel_prizes/physics/laureates/1958/index.html) visited on 05/14/2019
138. Theano Development Team. “Theano: A Python framework for fast computation of mathematical expressions”. *arXiv e-prints* abs/1605.02688, 2016.  
<http://arxiv.org/abs/1605.02688>
139. A. Toshniwal et al. “Storm@Twitter”. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’14. ACM, Snowbird, Utah, USA, 2014, pages 147–156.  
DOI: [10.1145/2588555.2595641](https://doi.org/10.1145/2588555.2595641)
140. J. Traa. *Least-Squares Intersection of Lines*. Technical report. University Illinois at Urbana Champaign, 2009.  
[http://cal.cs.illinois.edu/~johannes/research/LS\\_line\\_intersect.pdf](http://cal.cs.illinois.edu/~johannes/research/LS_line_intersect.pdf) visited on 05/16/2019.  
Note: Alternative link <https://docplayer.net/21072949-Least-squares-intersection-of-lines.html>
141. A. Tramacere, E. Massaro, and A. M. Taylor. “Stochastic Acceleration and the Evolution of Spectral Distributions in Synchro-Self-Compton Sources: A Self-consistent Modeling of Blazars’ Flares”. *Astrophysical Journal* 739, 66, 2011, page 66.  
DOI: [10.1088/0004-637X/739/2/66](https://doi.org/10.1088/0004-637X/739/2/66). ARXIV: [1107.1879](https://arxiv.org/abs/1107.1879) [astro-ph.HE]
142. T. Vincenty. “Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations”. *Survey Review* 23:176, 1975, pages 88–93.  
DOI: [10.1179/sre.1975.23.176.88](https://doi.org/10.1179/sre.1975.23.176.88)
143. T. C. Weekes et al. “Observation of TeV Gamma Rays from the Crab Nebula Using the Atmospheric Cerenkov Imaging Technique”. *Astrophysical Journal* 342, 1989, page 379.  
DOI: [10.1086/167599](https://doi.org/10.1086/167599)
144. D. C. Wells, E. W. Greisen, and R. H. Harten. “FITS - a Flexible Image Transport System”. *Astronomy & Astrophysics Supplement* 44, 1981, page 363
145. E. P. Wilkinson et al. *Chinese history: a manual*. Harvard Univ Asia Center, 2015. ISBN: 978-0-9988883-0-9
146. S. S. Wilks. “The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses”. *The Annals of Mathematical Statistics* 9:1, 1938, pages 60–62.  
DOI: [10.1214/aoms/1177732360](https://doi.org/10.1214/aoms/1177732360)
147. Z. Xu, W. Pankenier, and Y. Jiang. *East-Asian Archaeoastronomy: Historical Records of Astronomical Observations of China, Japan and Korea*. CRC Press, 2000. ISBN: 978-90-5699-302-3
148. V. Zabalza. “naima: a Python package for inference of relativistic particle energy distributions from observed nonthermal spectra”. In: *Proceedings of the 34<sup>th</sup> International Cosmic Ray Conference*. ICRC 2015. 2015, page 922.  
ARXIV: [1509.03319](https://arxiv.org/abs/1509.03319)

## *Bibliography*

149. B. Zadrozny and C. Elkan. “Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers”. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. 2001, pages 609–616. ISBN: 1-55860-778-1.  
<http://dl.acm.org/citation.cfm?id=645530.655658>
150. M. Zaharia et al. “Apache Spark: A Unified Engine for Big Data Processing”. *Communications of the ACM* 59:11, 2016, pages 56–65. ISSN: 0001-0782.  
DOI: [10.1145/2934664](https://doi.org/10.1145/2934664)