Advancing Blazar Science with Very-High-Energy Gamma-Ray Telescopes

Aryeh Louis Brill

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

Abstract

Advancing Blazar Science with Very-High-Energy Gamma-Ray Telescopes

Aryeh Louis Brill

Blazars, active galactic nuclei with relativistic jets pointed almost directly at Earth, are powerful and highly variable sources of nonthermal electromagnetic radiation, including very-high-energy gamma rays. We can detect these gamma rays with arrays of imaging atmospheric Cherenkov telescopes (IACTs), including the Very Energetic Radiation Imaging Telescope Array System (VERITAS) and the upcoming Cherenkov Telescope Array (CTA). After reviewing the science of blazars and the methods used by IACTs, we investigate how gamma-ray variability can provide insight into blazars' physical properties while also complicating efforts to understand these sources as a population. We first present a study of three flaring blazars observed with VERITAS and analyze these sources' spectral and variability characteristics, taking into account data at other wavebands, including that of the Large Area Telescope aboard the *Fermi* space telescope (*Fermi*-LAT). Next, after laying out how observing biases and intrinsic variability can confound blazar population studies with IACTs, we propose methods to account for these effects, and use simulated data to report expectations for a blazar luminosity function measurement with VERITAS. Sophisticated new instruments and data analysis methods can further expand the frontier of gamma-ray blazar science. To that end, we design a camera software system to enable safer and more efficient operations of a next-generation IACT being developed for CTA, the prototype Schwarzschild-Couder Telescope (pSCT). Finally, we develop methods to apply deep neural networks to the analysis of IACT data and employ these methods to reject background events detected by simulated arrays of IACTs.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my advisor Reshmi Mukherjee, whose tireless support and generous wisdom are a constant source of encouragement and wonder.

As a member of the Very Energetic Radiation Imaging Telescope Array System (VERITAS) and Cherenkov Telescope Array (CTA) collaborations, I have been lucky to be surrounded by mentors, professors, colleagues, and friends. I am especially grateful for the mentorship of the senior members of the VERITAS/CTA group at Columbia and Barnard, particularly Qi Feng, Brian Humensky, Daniel Nieto, and Marcos Santander.

I would like to thank all of my colleagues, many but not all in VERITAS and CTA, who eagerly shared their knowledge, provided helpful guidance and feedback, and engaged in productive and enjoyable discussions: Colin Adams, Massimo Capasso, Paolo Coppi, Manel Errando, Olivier Hervet, Deirdre Horan, Bryan Kim, Tjark Miener, Andriy Petrashyk, Martin Pohl, Deivid Ribeiro, Lorenzo Sironi, and Janeth Valverde, among many others.

I want to thank my parents, my family, my friends both at Columbia and elsewhere, and Sarah Meyer for the years of patience, understanding, belief, and love.

Dedication

To my normal force

who holds me up when I'm weighed down.

Chapter 1: Introduction

1.1 Very-High-Energy Astrophysics

Very-high-energy (VHE; between roughly 30 GeV and 300 TeV, often referred to simply as TeV) gamma rays are a window into the Universe's most extreme environments, providing a vital probe of astrophysics, cosmology, and fundamental physics. Gamma rays in this energy band can be indirectly detected on the ground using arrays of imaging atmospheric Cherenkov telescopes (IACTs), which detect the Cherenkov light emitted from air showers produced by VHE gamma rays when they are absorbed by the atmosphere. Indirect detection of VHE gamma-ray emission with ground-based instruments is complemented by direct detection in the high-energy (HE; between roughly 30 MeV and 300 GeV, often referred to simply as GeV) band with space-based telescopes. VHE astrophysics began in earnest with the detection by Weekes et al. (1989) of the brightest steady TeV source, the Crab nebula. The Crab is now used as a "standard candle" in VHE astrophysics, with fluxes often reported in Crab Units, that is, as a percentage of the Crab nebula flux. With current-generation IACTs, the Crab nebula can be detected at five standard deviations in about one minute. The known TeV gamma-ray sources are reported in the TeVCat catalog[1], and as of this writing, over 230 TeV sources have been detected, comprising numerous source classes and enabling a wide variety of scientific studies (e.g. CTA Consortium, 2019). Figure 1.1 shows a map of the known TeV sources and their classifications.

The majority of extragalactic TeV sources are blazars, a class of active galactic nuclei (AGN) characterized by luminous, rapidly variable, non-thermal broadband emission,

[1] `http://tevcat.uchicago.edu/`

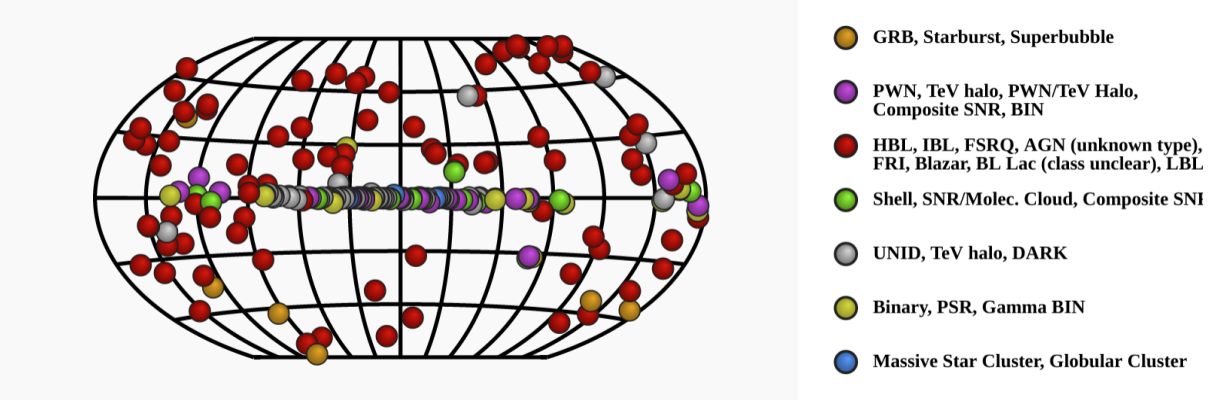


Figure 1.1: A map of the known TeV sources in Galactic coordinates, from TeVCat.

thought to originate in relativistic jets that are oriented nearly along our line of sight. Most TeV blazars belong to the class of BL Lac objects, although several flat spectrum radio quasars (FSRQs) have also been detected. IACTs often try to conduct blazar observations during flares, or short periods of elevated flux often observed at multiple wavelengths, which have been associated with high-energy neutrino production and can provide high signal-to-noise datasets enabling probes of AGN structure and searches for new fundamental physics. Blazars typically appear in the gamma-ray band as point sources with no resolvable spatial structure. A handful of detected radio galaxies (Archer et al., 2020) and starburst galaxies (VERITAS Collaboration et al., 2009) round out the extragalactic TeV source catalog.

Gamma-ray observatories play a key role in the emerging era of multimessenger astrophysics (for a review, see Mészáros et al., 2019), combining signals associated with the four forces of nature: electromagnetic radiation, gravitational waves, neutrinos (weak force), and cosmic rays (strong force). Cosmic rays have been observed at energies of over 10^{20} eV, and while the highest-energy cosmic rays are believed to be extragalactic in origin, their sources are unknown (Aab et al., 2020). Gamma-ray observations can help identify the sources of ultra-high-energy cosmic rays, which are charged particles and are therefore deflected by electromagnetic fields en route to the Earth. One recent discovery exemplifying the multi-

messenger paradigm is the coincident detection of gravitational waves and multiwavelength electromagnetic radiation produced by a short gamma-ray burst and subsequent kilonova caused by a binary neutron star merger (B. P. Abbott et al., 2017). IACTs have recently made the groundbreaking detections of gamma-ray emission from a flaring blazar associated with a highly energetic neutrino detected by the IceCube Neutrino Observatory (M. G. Aartsen et al., 2018; Abeysekara et al., 2018) and of TeV emission produced by a gamma-ray burst (Acciari et al., 2019).

Observations of blazars in the VHE band, especially in a multiwavelength and multi-messenger context, can help answer numerous scientific questions. Due to their head-on orientation, blazars produce relativistically beamed emission, presenting an opportunity to study AGN at the highest energies, greatest apparent luminosities, and fastest variability timescales. VHE blazar observations can help answer key open questions, including:

- Where is the gamma-ray emission region located, and what is its size and shape?
- How is the central black hole's energy transported to the emission region?
- What particle acceleration and radiative processes give rise to gamma-ray and multi-wavelength emission, and are blazar flares neutrino sources?
- What causes blazars' extreme variability on all observed timescales?
- Can blazar flares be predicted on any timescale, and what is their duty cycle?
- Do flares originate from different physical processes than quiescent emission?
- What physical mechanisms explain the observed differences between FSRQs and BL Lac objects? Do these blazar classes have an evolutionary relationship?

We can further use VHE gamma-ray emission from blazars at cosmological distances as a laboratory to study cosmology and particle physics, taking advantage of the fact that VHE

gamma rays may be absorbed via photon-photon interactions with intergalactic radiation fields. Measuring the extent and products of these absorption processes provides a probe of the extragalactic background light and intergalactic magnetic field, giving insight into how stars and large-scale magnetic fields formed over the history of the Universe. We can also use these measurements to constrain new physics, including potential interactions with axion-like particles (de Angelis, Galanti, and Roncadelli, 2011) and energy-dependent Lorentz invariance violation predicted by theories of quantum gravity (Abdalla et al., 2021).

Closer to home, Galactic TeV sources come in a diversity of types, including supernova remnants, pulsar wind nebulae, and gamma-ray binaries, often displaying complex morphologies; in addition, complex structures and large-scale diffuse emission are present throughout the Galactic plane and in the Galactic ridge in particular (Abdalla et al., 2018; Archer et al., 2016). The Crab nebula is a pulsar wind nebula. In addition to detections of TeV sources, constraints have been placed on hypothesized sources of TeV gamma-ray emission, such as gamma rays produced by the annihilation or decay of dark matter in dwarf galaxies (Archambault et al., 2017a).

1.2 Ground-Based Very-High-Energy Gamma-Ray Detectors

The first atmospheric Cherenkov telescope was built by Galbraith and Jelley (1953), who by placing a photomultiplier tube (PMT) at the focus of a parabolic mirror were able to observe pulses caused by cosmic-ray-induced Cherenkov showers. Although it was quickly realized that Cherenkov telescopes could also be used to observe gamma-ray showers, over three decades would elapse until the Crab nebula was successfully detected using the Whipple 10 m telescope by Weekes et al. (1989). This breakthrough was made possible by introducing imaging, allowing gamma-ray and cosmic-ray showers to be efficiently separated based on their morphology. The next major improvement to be introduced, by the current generation

of observatories, was the use of arrays of IACTs, dramatically improving event reconstruction and enabling the cosmic-ray background to be cut down still further. A history of the field of VHE astrophysics from the early days through the current generation of instruments is given by Hillas (2013).

Three major IACT observatories are currently in operation: the High Energy Stereoscopic System (H.E.S.S.), located in Namibia (F. Aharonian et al., 2006); the Major Atmospheric Gamma Imaging Cherenkov (MAGIC) telescope array, located on the island of La Palma in the Canary Islands, Spain (Aleksić et al., 2016); and the Very Energetic Radiation Imaging Telescope Array System (VERITAS), located at the Fred Lawrence Whipple Observatory (FLWO) in southern Arizona (Holder et al., 2006; Holder, 2011). H.E.S.S. has four 12 m telescopes as well as one 28 m telescope, which provides enhanced low-energy sensitivity, while MAGIC has two 17 m telescopes. VERITAS, the successor to Whipple, has four 12 m telescopes. Figure 1.2 shows a photograph of the VERITAS observatory.

Figure 1.2: A photograph of the VERITAS Observatory.

Building on the experience gained from these instruments, the next-generation observatory for VHE gamma-ray astronomy, the Cherenkov Telescope Array (CTA), is currently under development (Actis et al., 2011). CTA, with approximately 100 telescopes located in

two separate arrays in the Northern and Southern Hemispheres, will improve on the capabilities of the current-generation observatories by an order of magnitude. The CTA project includes the development of a novel dual-mirror IACT design, the Schwarzschild-Couder Telescope (SCT). A prototype SCT (pSCT) has been constructed at the VERITAS site. In addition to the major observatories, the First G-APD Cherenkov Telescope (FACT), located on La Palma, operates as a single telescope to perform automated monitoring of bright blazars (Dorner et al., 2019).

Observations with IACTs are complemented by data from water Cherenkov detector arrays, which detect air showers using the Cherenkov light emitted as shower particles pass through water tanks. Observatories using this technique include the High-Altitude Water Cherenkov (HAWC) observatory (Abeysekara et al., 2017) and the Large High-Altitude Air Shower Observatory (LHASSO; Bai et al., 2019). While these instruments have much higher duty cycles and larger fields of view compared to IACTs, they are less efficient at rejecting background cosmic-ray showers and are only sensitive to gamma rays with energies greater than about 10 TeV.

1.3 Space-Based High-Energy Gamma-Ray Detectors

Above the Earth's atmosphere, direct gamma-ray detection is possible. Typical astrophysical gamma-ray sources have steeply falling power-law spectra, making observations with space telescopes (with collection area limited to $\sim$1 m^2) feasible only for the relatively higher photon fluxes in the HE band. The first high-energy gamma-ray source was the quasar 3C 273, which was detected by the COS-B satellite (Swanenburg et al., 1978). Following that discovery, a large-scale survey in the HE band was conducted with the Energetic Gamma Ray Experiment Telescope (EGRET) instrument aboard the Compton Gamma Ray Observatory (CGRO) satellite, which was sensitive to gamma rays between 30 MeV and 30 GeV.

Figure 1.3: *Left*: Artist's rendition of the *Fermi* spacecraft. *Right*: Diagram of the LAT detector, with cutouts for illustration. ACD means anticoincidence detector. Both images are from NASA.

EGRET detected 271 sources, including over 60 blazars, during its operations from 1991 to 1996 (Hartman et al., 1999).

EGRET's successor is the Large Area Telescope (LAT) aboard the *Fermi* Gamma-ray Space Telescope (*Fermi*-LAT; Atwood et al., 2009). Since its launch on June 11, 2008, *Fermi*-LAT has been highly successful, having detected over 5700 gamma-ray sources (Abdollahi et al., 2020; Ballet et al., 2020), including approximately 3000 blazars (Ajello et al., 2020; Lott, Gasparrini, and Ciprini, 2020). *Fermi*-LAT is sensitive to gamma rays from 20 MeV to over 300 GeV and is capable of observing the entire sky in about three hours when operated in survey mode. The *Fermi* telescope and LAT instrument are illustrated in Figure 1.3.

The LAT is a pair-conversion telescope that detects the e^+e^- pairs produced by gamma rays incident on the detector (Ackermann et al., 2012). The LAT has three detector subsystems. First, a tracker, consisting of 18 silicon detector planes alternating with tungsten converter layers, promotes the conversion of gamma rays to e^+e^- pairs and enables reconstruction of the arrival directions of the incident particles. Next, a calorimeter, consisting of cesium iodide scintillation crystals stacked in eight layers, measures the particle energies and

facilitates background rejection by providing three-dimensional shower profiles. Finally, an anticoincidence detector consisting of scintillating tiles surrounds the tracker and calorimeter, used for rejection of showers caused by charged cosmic rays.

Building on *Fermi*'s success in the GeV band, a number of MeV-scale gamma-ray detectors have been proposed, promising to open up a new era of medium-energy gamma-ray astronomy (e.g. McEnery et al., 2019; Buckley et al., 2019).

1.4 About This Thesis

This thesis pursues two objectives. First, we aim to improve our understanding of the physical processes driving the luminous, variable gamma-ray emission from blazars by using data from IACTs, as well as from space-based telescopes. Second, we develop new instrumentation and data analysis methods to maximize the effectiveness of observations with IACTs. In Chapter 2, we discuss the taxonomy and observational properties of blazars, focusing on the gamma-ray context, and review the physical processes at work in these sources. In Chapter 3, we describe how IACTs perform gamma-ray observations, focusing primarily on VERITAS and on CTA, and on the SCT in particular.

In Chapter 4, we examine three luminous, highly variable FSRQs observed by VERITAS and *Fermi*-LAT, 3C 279, PKS 1222+216, and Ton 599. We constrain physical characteristics of these sources including the timescales of processes in their accretion disks, the jet Doppler factors and gamma-ray emission region locations, and the potential for these sources to produce PeV-scale neutrinos during TeV flares. Next, in Chapter 5, we consider population studies of TeV blazars. Such studies are challenged by variability and observational biases. In order to evaluate the prospects for measuring the TeV luminosity function of BL Lac objects with VERITAS, we develop methods to address some of these challenges.

In Chapter 6, we describe the design of a control and monitoring software system to

enable safe and efficient operation of the camera of the pSCT, a prerequisite for any scientific measurement. In order to improve IACT performance in general, in Chapter 7, we develop methods to apply deep neural networks to the analysis of data from IACT arrays. By making full use of the rich information present in IACT images, deep neural networks have the potential to improve the sensitivity of IACT observatories, particularly to the highly background-dominated soft spectra typical of blazars. Finally, we conclude with thoughts on future research directions flowing from this work.

Chapter 2: Blazars

2.1 Active Galactic Nuclei

AGN are energetic phenomena in the central regions of a few percent of galaxies, powered not by stars, but by accretion onto a supermassive black hole (SMBH). VHE gamma-ray blazars are the most extreme form of AGN, exhibiting highly relativistically beamed emission and offering the possibility to study fundamental physics. As a subclass of AGN, blazars belong to a complex taxonomy classifying objects based on historical observational accidents, apparent differences due to orientation, and true intrinsic differences between types of objects. Figure 2.1 shows a chart of the observational classifications of AGN.

At a high level, AGN can be divided into radio-quiet and radio-loud objects, with about 90% of all AGN being radio-quiet. The two subclasses of radio-quiet AGN are Seyfert galaxies and quasars. Seyfert galaxies have bright nuclei and strong emission lines in their optical spectra (Seyfert, 1943) and can be divided into two classes based on the presence or absence of broad lines, in addition to narrow lines, in their optical spectra. Quasars are highly variable objects that resemble stars in the optical band due to the relative faintness of the surrounding galaxy; for this reason, quasars are also referred to as quasi-stellar objects or QSOs. In general, quasars may be defined observationally as objects with a very broad spectral energy distribution (SED) explainable only by non-thermal processes (Peterson, 1997). Although most quasars are radio-quiet, about 10% are radio-loud. The bolometric luminosity of a typical quasar is about 100 times greater than that of a typical Seyfert galaxy. Despite their historical division into two discrete classes, it is now believed that

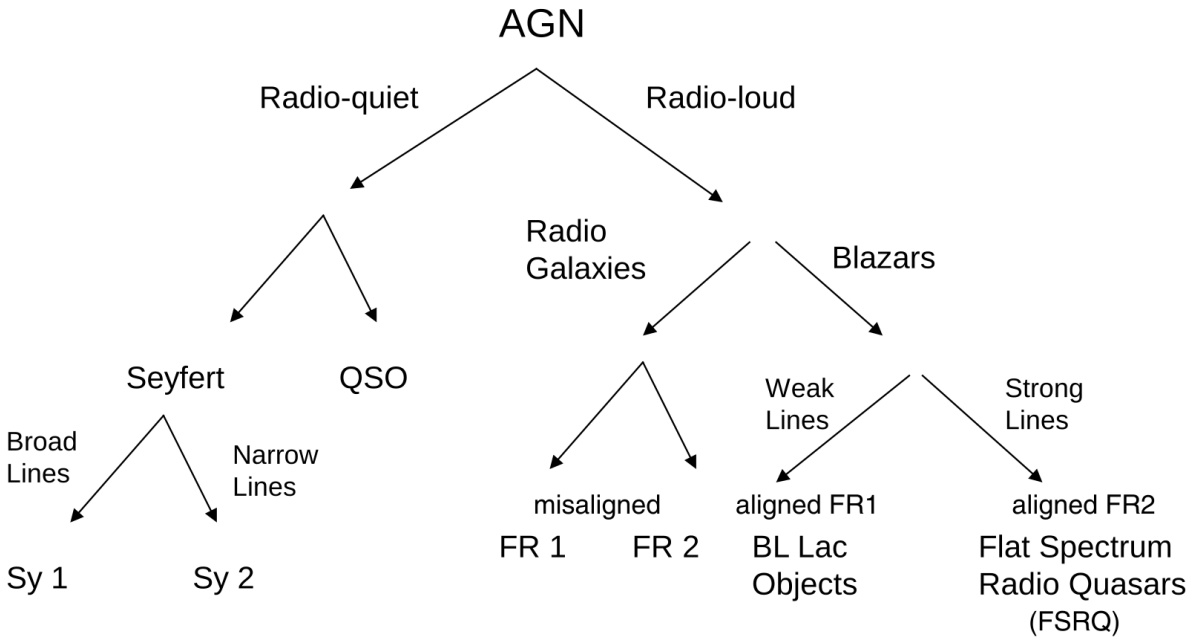


Figure 2.1: Chart of observational classifications of AGN, from Dermer and Giebels (2016). QSO are quasi-stellar objects; Sy 1 and Sy 2 are Seyfert 1 and 2 galaxies; FR 1 and FR 2 are Fanaroff-Riley 1 and 2 galaxies.

Seyfert galaxies and radio-quiet quasars probably form a continuous sequence in luminosity.

Radio-loud AGN, while making up a minority of AGN, include almost all of the gamma-ray-emitting objects. Observationally, radio-loud AGN fall into two main categories, radio galaxies and blazars. Fanaroff and Riley (1974) observed that radio galaxies can be divided into two classes: low-luminosity sources in which the luminosity decreases going outwards from the center (FR-I), and high-luminosity sources in which the luminosity increases going outwards (FR-II). Blazars are AGN displaying very rapid variability in flux and polarization, extremely high luminosities, and apparent superluminal velocities of compact radio cores (Urry and Padovani, 1995). Like radio galaxies, blazars can be divided into two classes: low-luminosity BL Lac objects and high-luminosity FSRQs. An FSRQ can be identified by its broad optical spectral lines and flat radio spectrum, with a spectral index $\alpha \approx 0$, while the optical spectra of BL Lac objects are practically featureless. Radio-loud AGN typically

occur in elliptical galaxies, unlike Seyfert galaxies, which are mostly spiral galaxies.

AGN can be understood as consisting of a number of discrete structures, pictured in Figure 2.2 (Urry and Padovani, 1995). An accretion disk surrounds the SMBH at the center. The broad and narrow spectral lines are associated with separate regions of ionized gas, referred to as the broad-line region (BLR) and narrow-line region. These regions are ringed by a dusty molecular torus, which occludes the BLR when the AGN is viewed off-axis. In radio-loud AGN, a relativistic jet is also present, with the jet axis perpendicular to the plane of the torus. However, the low-luminosity BL Lac objects and FR-I radio galaxies are believed to lack most of these structures besides the accretion disk and jet. To put these AGN components in context, Table 2.1 summarizes their approximate length scales with the sizes of other astronomical structures for comparison.

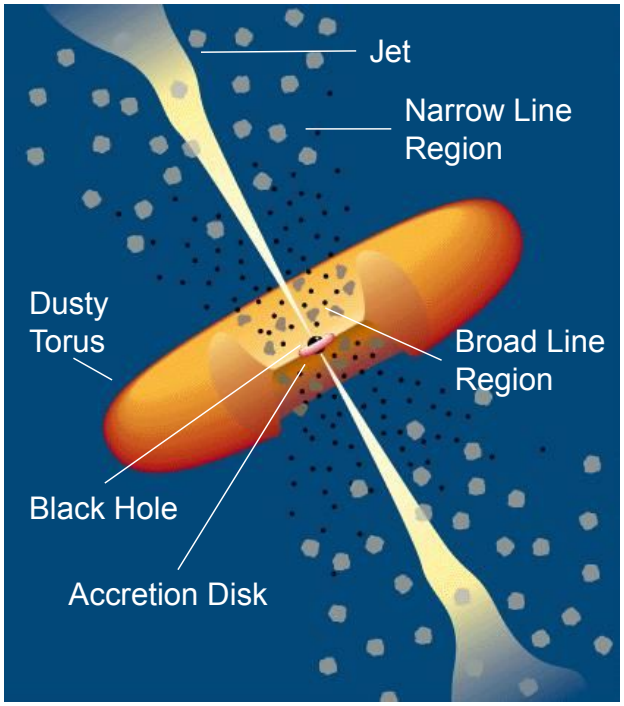


Figure 2.2: Schematic diagram of an AGN (not to scale), adapted from Urry and Padovani, 1995.

The unified model of AGN makes sense of this assortment of classifications and subtypes (Urry and Padovani, 1995). In this model, schematically represented in Figure 2.3, the

Structure	Approximate Size[a]	Comparison
SMBH event horizon	~ 0.01 – 10 AU	Earth-Sun distance, 1 AU
Disk inner radius	~ 0.01 – 60 AU	
Disk outer radius	~ 1 – 1000 AU	Outer radius of Kuiper belt, 50 AU[b]
Broad-line region	~ 0.01 – 1 pc	Distance to Alpha Centauri, 1.34 pc[c]
Molecular torus	~ 1 pc	
Narrow-line region	$\sim 10^2$ – 10^4 pc	Distance to Galactic Center, 8×10^3 pc[d]
Jet	$\gtrsim 10^5$ pc	

Table 2.1: Approximate sizes of AGN component structures. [a]Beckmann and Shrader (2012). [b]Stern and Colwell (1997). [c]Bailer-Jones et al. (2018). [d]Binney and Merrifield (1998).

appearance of an AGN depends essentially on three parameters: the presence or absence of a relativistic jet, the power of the central engine, and the orientation of the observer. Radio-loud AGN are defined by the presence of a relativistic jet. Another real difference exists between the high-power and low-power radio sources, which may be caused by a transition from retrograde to prograde spin of the central SMBH (Garofalo, Evans, and Sambruna, 2010). The remaining apparent differences are explained by the contingent orientation of the object with respect to the observer, which produces two effects. First, narrow-line objects result when we view them at large inclination angles so that the BLR is occluded by the molecular torus. Second, we observe a blazar when the jet is aligned with our line of sight, producing relativistically beamed emission. In this paradigm, BL Lac objects and FSRQs are the aligned counterparts of FR-I and FR-II radio galaxies, respectively.

2.2 Physical Processes in AGN

2.2.1 Accretion Power

An AGN is ultimately powered by the gravitational energy of a SMBH at the center of its host galaxy. The kinetic energy of a test particle of mass m falling onto a compact object from infinity can be estimated in the classical limit as (Longair, 1994)

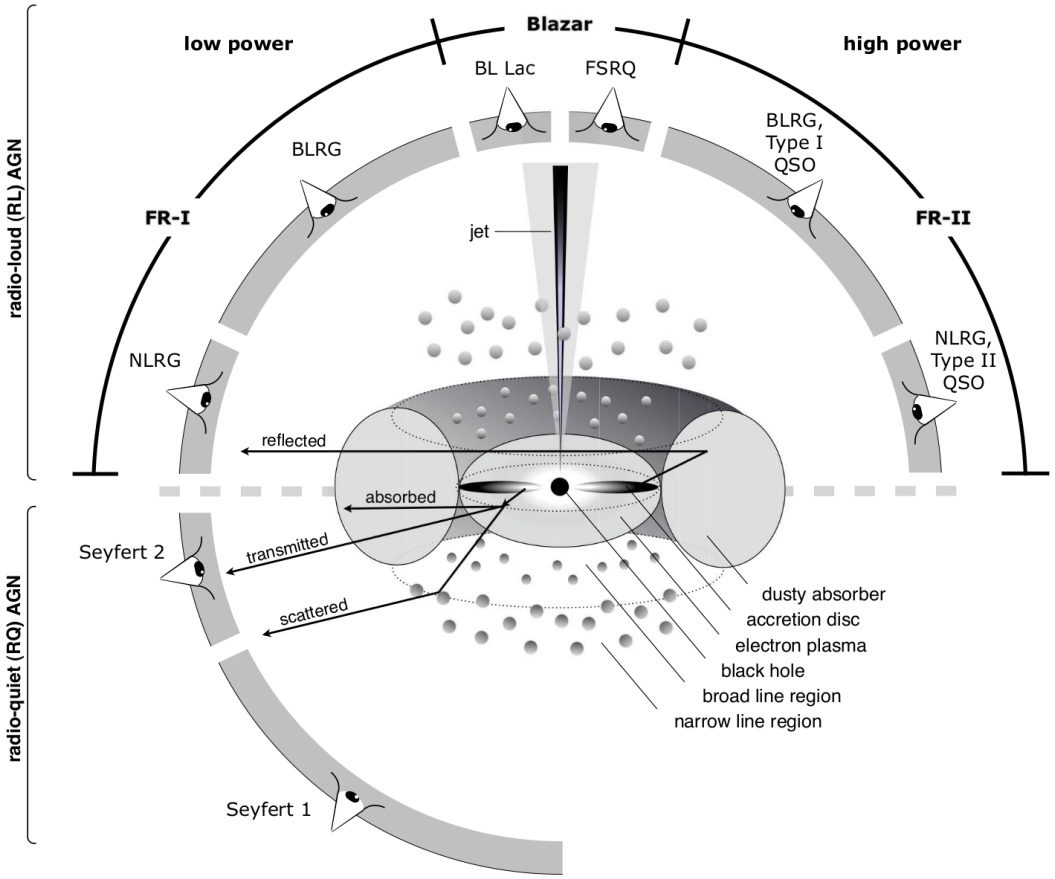


Figure 2.3: Schematic representation of the unification scheme of AGN, from Beckmann and Shrader (2012). The type of object observed depends on the presence or absence of a jet, the power of the central engine, and the orientation of the observer.

$$E \approx \left(\frac{r_S}{2R}\right) mc^2, \tag{2.1}$$

where $r_S \equiv 2GM/R$ is the Schwarzschild radius. Letting $\dot{m}$ be the accretion rate onto the central object, the luminosity of the source is

$$L \approx \left(\frac{r_S}{2R}\right) \dot{m}c^2 = \xi \dot{m}c^2, \tag{2.2}$$

where ξ is the efficiency of release of the accreting matter's rest-mass energy. For a non-rotating Schwarzschild black hole, the innermost stable orbit occurs at $R = 3r_S$, giving $\xi \approx 0.1$. A relativistic calculation yields $\xi = 0.057$ for a Schwarzschild black hole and a whopping $\xi = 0.426$ for a maximally rotating Kerr black hole. These are very high efficiencies. For comparison, the fusion of hydrogen into helium yields only $\xi = 0.007$.

However, black holes have no surface on which to actually dissipate the energy of infalling matter into heat. To see how this energy can be released, we observe that because of conservation of angular momentum, matter cannot fall directly into the black hole. Instead, infalling matter forms an accretion disk about the black hole's rotation axis. Viscous frictional forces in the disk then act to dissipate the angular momentum of infalling matter, while also heating the disk. At the same time, the luminous non-thermal emission observed in quasars is believed to be produced by a relativistically outflowing jet ejected from the black hole (Blandford and Königl, 1979). The energy to power the jet can be extracted from a rotating black hole via the mechanism of Blandford and Znajek (1977), in which the twisting of ambient magnetic fields by the black hole's rotation extracts energy and angular momentum, producing an energetic outflow.

As the luminosity due to accretion of ionized plasma increases, so does the resulting radiation pressure, until the outward force balances the inward pull of gravity. The resulting luminosity upper bound is called the Eddington luminosity, and is given by

$$L_{\rm E} = \frac{4\pi G M m_{\rm p} c}{\sigma_{\rm T}} \approx 1.3 \times 10^{38} \left(\frac{M}{M_\odot}\right) \text{ erg s}^{-1}, \tag{2.3}$$

where $m_{\rm p}$ is the proton mass and $\sigma_{\rm T}$ is the Thomson cross section (Eq. 2.12). The Eddington luminosity limit is only applicable for a steady state.

2.2.2 Particle Acceleration

Particle acceleration mechanisms transfer the energy extracted from the central engine into a population of relativistic particles. Two main such mechanisms have been proposed to be at work in blazars: shock acceleration and magnetic reconnection.

Shock acceleration has its basis in a model proposed by Fermi (1949). In this model, particles repeatedly collide with magnetic fields randomly moving with speed V, gaining or losing energy with each collision, before exiting the collision zone on some characteristic timescale. The energy gains from head-on collisions slightly outweigh the losses from overtaking collisions because head-on collisions occur more often. The average fractional energy increase with each collision is given by

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{8}{3} \left(\frac{V}{c}\right)^2. \tag{2.4}$$

Because the average energy increase is second order in V/c, this mechanism is referred to as second-order Fermi acceleration. As a result, the acceleration is somewhat slow and inefficient. A first-order Fermi acceleration model in which all collisions are effectively head-on was proposed by Bell (1978), called diffusive shock acceleration. In this model, relativistic particles repeatedly cross a shock front, with turbulence isotropizing the particle velocities on both sides. As a result, the particles gain energy when they cross in either direction, with the average energy increase per round trip now given by

$$\left\langle \frac{\Delta E}{E} \right\rangle = \frac{4}{3}\left(\frac{V}{c}\right). \tag{2.5}$$

In either case, we can calculate the spectrum expected from this process, following Longair (1994). Suppose that a population of N_0 particles is accelerated such that $E = \beta E_0$ is the energy after one collision and each particle remains in the acceleration zone with probability P after each collision. Then after k collisions, we have $E = E_0\beta^k$ and $N = N_0 P^k$. Eliminating k between these expressions and taking the derivative yields

$$\frac{dN}{dE} \propto \left(\frac{E}{E_0}\right)^{-1+\frac{\ln P}{\ln \beta}}. \tag{2.6}$$

In diffuse shock acceleration, $\ln P \approx -4V/3c$ and $\ln \beta = \ln(1 + 4V/3c) \approx 4V/3c$, so $\ln P/\ln \beta \approx -1$ and we have $p = 2$ for the spectral index.

The fact that the Blandford and Znajek (1977) mechanism produces jets that are magnetically dominated at their base suggests an alternative scenario in which the dissipation of magnetic fields rather than a shock transfers energy to particles. In this model, called magnetic reconnection, magnetic field lines of opposite polarity annihilate in an electron-position plasma, forming magnetic islands or "plasmoids" where particles are efficiently accelerated to produce a power-law spectrum with index $1 \lesssim p \lesssim 4$ that hardens with increasing magnetization (Guo et al., 2015; Sironi and Spitkovsky, 2014). Reconnection is most efficient in a highly magnetized plasma. However, because reconnection dissipates magnetic fields and shocks concentrate them, both processes tend to produce downstream emission regions in a rough equipartition between magnetic and particle energy densities, $U_B/U_e \sim 1$ (Sironi, Petropoulou, and Giannios, 2015). Particles can be accelerated when two plasmoids collide, producing fast flares accompanied by polarization swings (Hosking and Sironi, 2020).

2.2.3 Radiative Processes

Next, the accelerated particles generate the electromagnetic radiation that we detect with our telescopes. The most important radiative processes in blazars are synchrotron radiation and inverse Compton scattering, in which, as we will see, leptons (i.e. electrons and positrons) play the dominant role. However, there are several mechanisms by which relativistic protons may also radiate, in so-called hadronic processes.

Synchrotron radiation is emitted when a relativistic charged particle is accelerated by a magnetic field. A particle of mass m and charge e moving with Lorentz factor γ in a magnetic field $\mathbf{B}$ will exhibit helical motion about an axis parallel to $\mathbf{B}$ and emit radiation in a narrow cone of opening angle $1/\gamma$, as illustrated in Figure 2.4. The gyration frequency of the helical motion is

$$\nu_g = \frac{eB}{2\pi\gamma mc}, \tag{2.7}$$

and the emitted power is

$$P = \frac{2}{3}\frac{e^4}{m^2c^3}\beta^2\gamma^2 B^2 \sin\alpha^2, \tag{2.8}$$

where α in this context is the pitch angle between the particle velocity vector and B. Since $P \propto m^{-2}$, we immediately see that the emission from electrons will dominate that from protons. We therefore take m and e to refer to the electron mass and charge, respectively.

Averaging over all pitch angles yields

$$P = \frac{4}{3}\sigma_{\mathrm{T}} c\beta^2\gamma^2 U_B, \tag{2.9}$$

where $U_B = B^2/8\pi$ is the energy density of the magnetic field and the Thomson cross section σ_{T} is defined below. This expression is valid for a single particle, but in an astrophysical

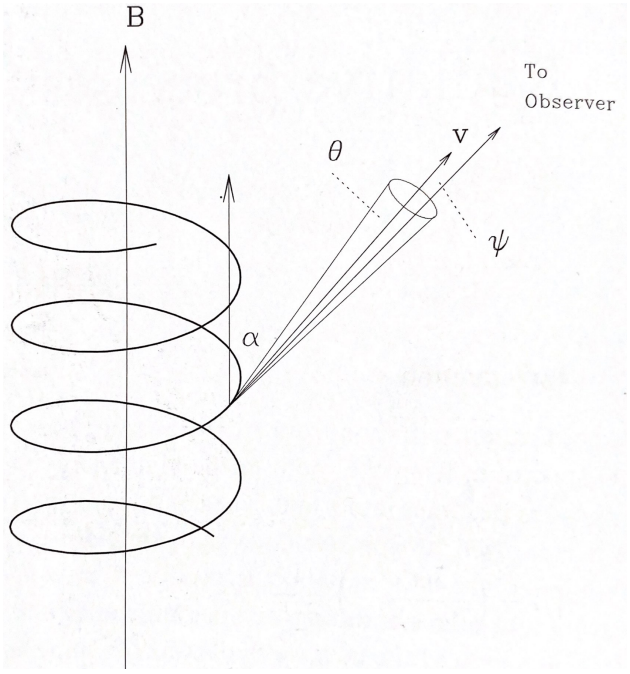


Figure 2.4: Diagram of a charged particle moving in a magnetic field and emitting synchrotron radiation, from Kembhavi and Narlikar (1999).

source, we expect the electron energies to have a power-law distribution with spectral index γ. In that case, the emitted spectrum will also have a power-law form,

$$P(\nu) \propto \nu^{-\alpha}, \alpha = \frac{p-1}{2}, \tag{2.10}$$

where α is now the energy spectral index. It is conventional in high-energy astrophysics to report spectra in terms of photon counts, in which case the photon spectral index is equal to $\Gamma = \alpha + 1$.

The synchrotron cooling timescale can be estimated by setting $dE/dt = -P$, giving

$$t_{1/2} = \frac{6\pi mc}{\sigma_{\mathrm{T}} \gamma B^2} \approx 7.74 \times 10^8 \ \gamma^{-1} \left(\frac{B}{1\ \mathrm{G}}\right)^{-2} \ \mathrm{s}, \tag{2.11}$$

where $t_{1/2}$ is the time needed for the electron to lose half of its energy.

Another important radiative process in blazars is inverse Compton scattering, in which a

relativistic electron can transfer energy to a photon. In the nonrelativistic limit, this process reduces to Thomson scattering, in which the photon is elastically scattered with no change in energy. The Thomson scattering cross section is frequency-independent, and is given by

$$\sigma_{\rm T} = \frac{8\pi e^4}{3m^2c^4} \approx 6.65 \times 10^{-25}\ {\rm cm}^2. \tag{2.12}$$

In the general case, the cross section is given by the Klein-Nishina formula,

$$\sigma_{\rm KN} = \frac{3\sigma_{\rm T}}{4}\left\{\frac{1+x}{x^3}\left[\frac{2x(1+x)}{1+2x} - \ln(1+2x)\right] + \frac{\ln(1+2x)}{2x} - \frac{1+3x}{(1+2x)^2}\right\}, \tag{2.13}$$

where $x \equiv \epsilon/mc^2$, with ϵ being the energy of the incident photon. For $\epsilon \gg mc^2$, the Klein-Nishina cross section falls rapidly to zero, and for $\epsilon \ll mc^2$, it reduces to the Thomson cross section. The Compton power is given by

$$P = \frac{4}{3}\sigma_{\rm T} c\beta^2\gamma^2 U_{\rm rad}, \tag{2.14}$$

where $U_{\rm rad}$ is the energy density of electromagnetic radiation. Comparing Eq. 2.14 to Eq. 2.9, we see that these expressions have the same form, with

$$\frac{P_{\rm Comp}}{P_{\rm synch}} = \frac{U_{\rm rad}}{U_B}. \tag{2.15}$$

This similarity can be explained by interpreting synchrotron radiation as a scattering process with the virtual photons of the magnetic field (Kembhavi and Narlikar, 1999). The ratio $P_{\rm Comp}/P_{\rm synch}$ is known as the Compton dominance, generally measured as the ratio of peak Compton to synchrotron luminosities from the SED (Section 2.3.1). If the electron energy distribution is a power law, the result of Eq. 2.10 holds for inverse Compton emission as well (Blumenthal and Gould, 1970).

In hadronic models, some or all of the radiation is due to relativistic protons emitting via photohadronic processes, proton synchrotron radiation, or other mechanisms. In hadronic processes, relativistic neutrinos may be emitted in addition to electromagnetic radiation. One such process is the synchrotron radiation of high-energy protons, which has been used to explain the high-energy emission from blazars (e.g. F. Aharonian, 2000; Mücke and Protheroe, 2000). Alternatively, neutrinos may be produced by the photohadronic interaction of a proton with a photon, producing pions that quickly decay to gamma rays and neutrinos (Dermer and Menon, 2009). The dominant photopion production channel occurs via the $\Delta^+(1232)$ resonance:

$$p + \gamma \rightarrow \Delta^+ \rightarrow \begin{cases} p + \pi^0 \rightarrow p + 2\gamma \\ n + \pi^+ \rightarrow n + e + 3\nu \rightarrow p + 2e + 4\nu. \end{cases} \tag{2.16}$$

The $p\pi^0$ and $n\pi^+$ rates occur in a 1:2 ratio via the Δ^+ resonance, but when the direct production channel of $p\gamma \rightarrow p\pi^0$ is also included, the rates become roughly equal. The secondary neutrino energies are about 5% of the initial proton energy, with the proton losing about 20% of its energy to the decay products. The resonance occurs when the invariant energy of the interaction is $\epsilon_{p\gamma,\mathrm{th}} \sim 0.3$ GeV. When this energy is higher, the main decay channel is multipion production, with π^+, π^0, and π^- particles produced in equal numbers. Photopion production of PeV-scale neutrinos requires a target photon population in the X-ray band. The $p\gamma$ process may co-occur with leptonic gamma-ray emission, and under this scenario, FSRQs may be sources of relativistic neutrinos at PeV or even EeV energies (e.g. Gao, Pohl, and Winter, 2017; Righi et al., 2020).

2.2.4 Relativistic Motion

The jet's bulk relativistic motion strongly modifies the observed emission from blazars. Relativistic Doppler boosting of the jet along an axis close to our line of sight is needed to explain many observational characteristics, with the apparent superluminal motions observed in radio jets and the ability of strong gamma-ray fluxes to escape compact emission zones without being absorbed via pair production being two of the most striking.

Many blazars display superluminal motion in their jets when observed using very-long-baseline interferometry (VLBI), often with apparent velocities of $\sim$5–10c or more (e.g. Jorstad et al., 2001). For example, the VLBI measurements of the FSRQ 3C 279 shown in Figure 2.5 reveal a radio structure moving twenty-five light-years in only seven years. This phenomenon was predicted by Rees (1966), who explained it as a geometrical relativistic effect resulting when a jet is moving with a substantial component of its velocity towards the observer so that it almost catches up with its emitted light.

Figure 2.6 shows the setup in which apparent superluminal motion can occur. Consider a source ("blob") at a distance D from the observer, moving relativistically with velocity βc at an inclination angle θ with respect to the observer's line of sight. In a time interval $\Delta\tau$, the blob moves a distance $\Delta x = \beta c \Delta\tau \sin\theta$ tangential to the observer's line of sight, which is the apparent distance traveled, and a distance $\Delta y = \beta c \Delta\tau \cos\theta$ towards the observer. The time measured by the observer for the blob to traverse the distance Δx is

$$\Delta\tau_{\rm app} = \left(\Delta\tau + \frac{D - \Delta y}{c}\right) - \frac{D}{c} = \Delta\tau(1 - \beta\cos\theta). \tag{2.17}$$

The apparent velocity measured by the observer is then

$$\beta_{\rm app} = \frac{\Delta x}{c\Delta\tau_{\rm app}} = \frac{\beta\sin\theta}{1 - \beta\cos\theta}. \tag{2.18}$$

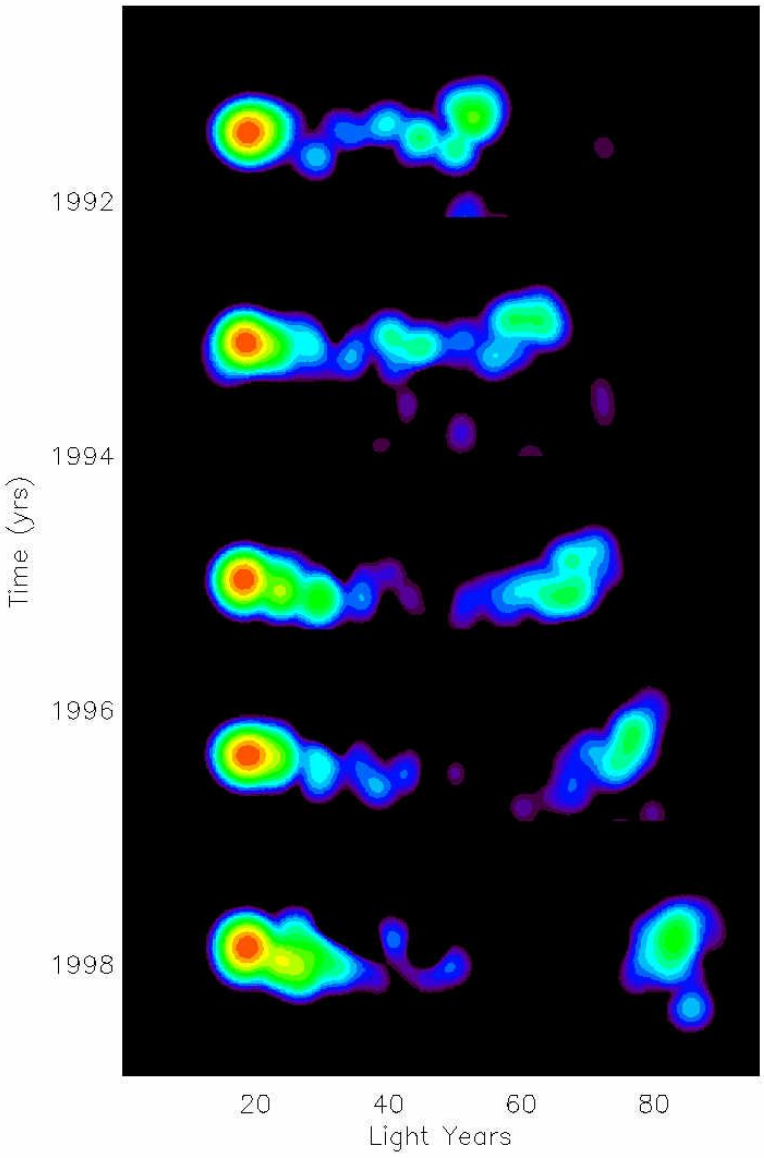


Figure 2.5: Apparent superluminal motion in the radio jet of 3C 279, from Wehrle et al. (2001).

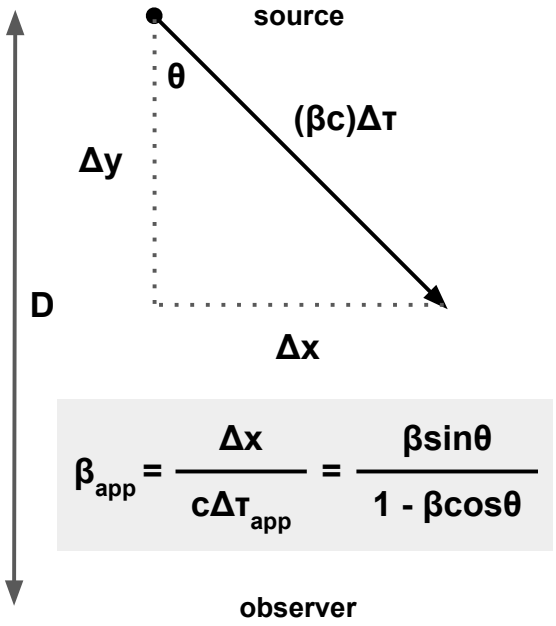


Figure 2.6: Diagram showing the setup leading to apparent superluminal motion. A radiating relativistic source moves distances Δx and Δy tangential and parallel to the observer's line of sight, respectively, in the time interval $\Delta\tau$.

This expression is maximized when $\cos\theta = \beta$, or $\theta \approx 1/\Gamma$, where $\Gamma = 1/(1-\beta^2)^{1/2}$ is the Lorentz factor corresponding to the bulk motion of the jet. At this inclination angle, apparent superluminal motion will be observed for any $\beta > 1/\sqrt{2}$.

Other evidence for relativistic motion in blazars comes from observations of gamma-ray emission with both high luminosity and fast variability. This is surprising because gamma rays naturally undergo absorption by pair production ($\gamma + \gamma \rightarrow \mathrm{e}^+ + \mathrm{e}^-$) with target photons of energy $h\nu_{\mathrm{target}} \sim (m_{\mathrm{e}}c^2)^2/h\nu_\gamma$. The optical depth to pair production is given by $\tau_{\gamma\gamma} \lesssim \sigma_{\mathrm{T}} n_\gamma R$, where R is the size of the source and n_γ is the number density of gamma rays, which can be estimated as $L_\gamma/4\pi R^2 m_{\mathrm{e}}c^3$. Associating the source's variability timescale (assuming no beaming) with its light-crossing time, we have $R \sim ct_{\mathrm{var}}$. Defining the dimensionless *compactness parameter* (Guilbert, Fabian, and Rees, 1983),

$$\ell \equiv \frac{L_\gamma}{R}\frac{\sigma_{\mathrm{T}}}{m_{\mathrm{e}}c^3}, \tag{2.19}$$

gamma rays can escape the source only if (Urry and Padovani, 1995),

$$\tau_{\gamma\gamma} \sim \frac{\ell}{40} \ll 1. \tag{2.20}$$

A bright blazar (e.g. 3C 279, Maraschi, Ghisellini, and Celotti, 1992) can have a typical HE gamma-ray luminosity of $L \sim 10^{48}$ erg s^{-1} with variability observed on daily or shorter timescales, yielding an optical depth of

$$\tau_{\gamma\gamma} \approx 3 \times 10^2 \left(\frac{L_{\rm obs}}{10^{48}\ {\rm erg\ s^{-1}}}\right) \left(\frac{t_{\rm var,obs}}{1\ {\rm day}}\right)^{-1}, \tag{2.21}$$

clearly in conflict with the large observed gamma-ray flux! Relativistic beaming resolves this tension by altering both the required intrinsic luminosity and variability timescale. First, the observed luminosity produced by relativistically beamed emission, L, is related to the intrinsic luminosity $\mathcal{L}$ by

$$L = \delta^p \mathcal{L}, \tag{2.22}$$

where the Doppler factor δ is given by

$$\delta = \frac{1}{\Gamma(1 - \beta\cos\theta)}, \tag{2.23}$$

where θ is the viewing angle between the velocity vector and the observer's line of sight. The exact value of the exponent p depends on the physical processes in the source. For the basic case of jet emission from a relativistic blob radiating isotropically in its rest frame, $p = 3 + \alpha$, which can be interpreted as a factor of δ^2 amplification of the flux by relativistic aberration, a factor of δ increase in the number of photons arriving per unit time caused by time dilation, and a factor of δ^α increase in energy caused by the blueshifting of the emitted spectrum (Urry and Shafer, 1984). For a symmetrical two-sided relativistic jet with $\Gamma \sim 10$,

$\theta \sim 1/\Gamma$, and $\beta \sim 1$, the ratio of observed luminosities of the approaching and receding components is $\gtrsim 10^9$, so the contribution of the receding component can be safely neglected.

When relativistic beaming is accounted for, the size of the source can be estimated from the observed variability timescale as

$$R \sim \frac{\delta}{1+z} c t_{\rm var}, \tag{2.24}$$

where the effect of the cosmological redshift z is also included in Equation 2.24. The optical depth due to pair production is thus reduced by a factor of δ^{p+1}, with $(p+1) \sim 5$, and the observed gamma-ray emission can be satisfactorily explained.

2.3 Observed Characteristics of Blazars

2.3.1 Spectral Energy Distribution

Figure 2.7 illustrates the basic properties of blazar SEDs. An SED plots the energy flux (Section 3.4.5) as a function of frequency, or equivalently, of energy. A blazar SED has a double-humped structure, with a low-energy component caused by synchrotron radiation emitted by relativistic electrons and a high-energy gamma-ray component that may be produced by one or more different mechanisms. Modeling a blazar SED can give insight into its physical structure and multiwavelength emission mechanisms. In leptonic models, the gamma-ray component is explained by relativistic electrons scattering via the inverse Compton process off of a population of lower-energy seed photons. One possibility is that the seed photons are the same synchrotron photons that the electrons themselves emitted, in what is called the synchrotron self-Compton process (SSC; Maraschi, Ghisellini, and Celotti, 1992). Alternatively, the seed photons may be radiation from an external structure, such as accretion disk radiation reprocessed in broad-emission-line clouds, in what is called the external inverse Compton process (EIC; e.g. Ghisellini and Madau, 1996). The EIC seed photons are

commonly taken to be radiation fields in the BLR, although this picture has been challenged by the lack of characteristic BLR absorption features in the average gamma-ray spectra of *Fermi*-LAT FSRQs (Costamante et al., 2018). The presence or absence of EIC emission can give inform our understanding of the gamma-ray emission region location relative to radiating structures in the AGN.

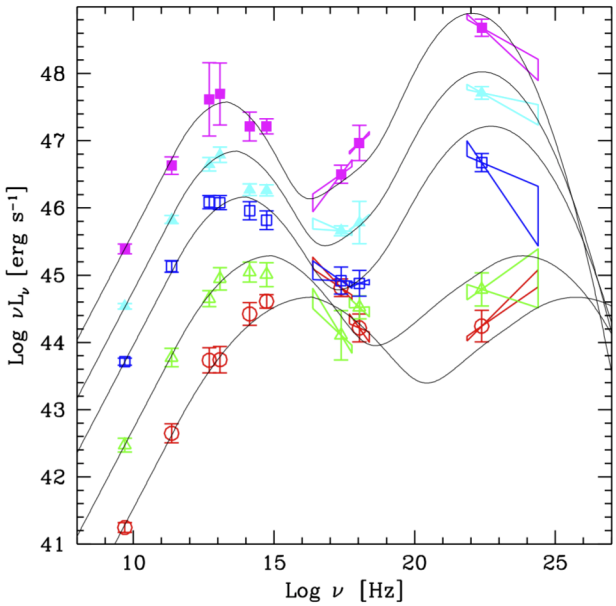


Figure 2.7: Sample-averaged blazar SEDs with model fits, from Donato et al. (2001), using the blazars studied by Fossati et al. (1998). The double-humped structure and anticorrelation of peak frequency with luminosity are clearly visible.

Another important property, known as the blazar sequence, is the inverse relationship of the peak frequencies of the humps with the bolometric luminosity (Section 2.3.3). The SED of an FSRQ is typically dominated by its gamma-ray emission component, which peaks in the HE band – in other words, FSRQs are Compton-dominated. On the other hand, BL Lac objects are usually less luminous overall as well as less Compton-dominated, but have a gamma-ray emission component that peaks in the VHE band.

2.3.2 Variability

Blazars exhibit extreme variability on all observed timescales, limited only by the duration and sensitivity of the observations, down to several minutes in both the GeV and TeV gamma-ray bands (Ackermann et al., 2016; F. Aharonian et al., 2007). The physical mechanisms that drive this variability are not well understood. It is possible that different processes, possibly originating at different locations in the AGN, drive variable emission occurring at different timescales. While short variability timescales observed in blazars suggest that the emission may be connected to processes in the central engine or accretion disk, the ability of TeV gamma rays to escape the AGN without being absorbed implies that the emission originates further out in the jet, such as in the molecular torus, as recently suggested by Harvey, Georganopoulos, and E. T. Meyer (2020). The observed variability directly constrains the size of the emission region (Eq. 2.24). Blazar power spectral densities in the gamma-ray band can usually be described as a power law with an index of ∼1-3 (Finke and Becker, 2014), indicating that their flux variability is at least partly stochastic in origin.

Variability at the intermediate (∼days–months) timescale is of particular importance for modeling the size, shape, location, and dynamics of the emission region (or regions). It therefore plays a key role in disentangling the sources of the emission observed at multiple wavelengths and with different astrophysical messengers. Notably, at this timescale blazars undergo "flares" in which their optical, X-ray, or gamma-ray flux can increase by as much as an order of magnitude. Blazar flares may last for just a few days or weeks, but observing them is critical to coordinate multiwavelength observations, obtain high signal-to-noise spectra and light curves, and probe the unique physics that may be occurring during these events. Knowledge of the flare duty cycle in blazars, or fraction of time these sources spend in a flaring state, is critical for planning observations with pointed telescopes and relating these sources' electromagnetic emission to other messengers. The duty cycle of elevated flux states

has been estimated to be about 5-10% in Fermi AGN (Ackermann et al., 2011).

Despite their prime observational importance, flares are vaguely defined and poorly understood physically. While flares can be thought of intuitively as rare, bright events lasting for a few days or weeks, they do not have a standard objective definition in the community. Flares display a diversity of structures, durations, and luminosities. They are difficult to separate from the supposed quiescent background level, and it is an open question whether such a separation is physically meaningful at all. It is not clear whether flares are generated by a physical process distinct from the source of the quiescent emission, or whether they are merely upward fluctuations resulting from a continuous stochastic process. Naturally, these fundamental difficulties complicate the calculation of the flare duty cycle as well.

Observations of blazar variability on other timescales are important as well. Observations of rapid ($\sim$minutes–hours) variability can put strong constraints on the Doppler factor and the particle acceleration processes producing the emission due to the strong constraints placed on the size of the emission region. Various models have been proposed to explain this fast variability, including relativistic magnetic reconnection (Petropoulou, Giannios, and Sironi, 2016) and red giant stars interacting with the jet close to the central SMBH (Barkov et al., 2012). Over long ($\sim$annual) timescales, blazar variability can be studied using the flux distribution, which describes the relative frequencies of different flux levels. Blazar flux distributions exhibit long tails, and have been fit using log-normal models (e.g. Giebels and Degrange, 2009; Sinha et al., 2017; Shah et al., 2018), which could indicate evidence of an underlying multiplicative physical process. Light curves with a log-normal flux distribution have the property that their amplitude of variability is linearly proportional to their mean flux (Uttley, McHardy, and Vaughan, 2005; but see Scargle, 2020). M. Meyer, Scargle, and Blandford (2019) studied six bright FSRQs and modeled their flux distributions using a broken power law, though a log-normal distribution was also compatible with their data. As we will discuss in Chapter 4, Tavecchio, Bonnoli, and Galanti (2020) have proposed an

alternative model to describe the flux distributions of those same objects, which is based on a stochastic differential equation. Gamma-ray observations may also be used to search for evidence of periodic or quasi-periodic processes in blazars, such as OJ 287, which displays optical outbursts with an approximately 12 year period (Kushwaha, Sahayanathan, and Singh, 2013; O'Brien, 2017).

2.3.3 Populations of Gamma-ray Blazars

Blazars can be classified according to the peak frequency of their synchrotron emission. The conventional definitions are high-synchrotron-peaked (HSP) objects with $\nu_{\rm synch} > 10^{15}$ Hz, intermediate-synchrotron-peaked (ISP) objects with 10^{14} Hz $< \nu_{\rm synch} < 10^{15}$ Hz, and low-synchrotron-peaked (LSP) objects with $\nu_{\rm synch} < 10^{14}$ Hz. FSRQs virtually all belong to the LSP class. BL Lac objects are found in all of the synchrotron peak classes, and following Padovani and Giommi (1995), we refer to HSP, ISP, and LSP BL Lac objects as HBLs, IBLs, and LBLs, respectively.

Empirically, a blazar's synchrotron peak frequency is inversely correlated with its luminosity. This anti-correlation has been interpreted as a continuous *blazar sequence* (FSRQ – LBL – IBL – HBL) parameterized by the decreasing intrinsic luminosity of the jet (Fossati et al., 1998; Ghisellini et al., 2017). More recent work has disfavored the intrinsic validity of the blazar sequence, explaining it as an artifact of Doppler boosting (Nieppola et al., 2008). In particular, the apparent blazar sequence may in fact be merely the high-luminosity envelope of two separate strong-jet and weak-jet populations, shown in Figure 2.8 (E. T. Meyer et al., 2011; Keenan et al., 2021). Regardless of its physical interpretation or lack thereof, the blazar sequence remains a good empirical description of the average characteristics of blazar populations.

Accordingly, while FSRQs typically have bolometric luminosities greater than those of BL Lac objects, they are rarely detected at TeV energies. Instead, HBLs are the dominant

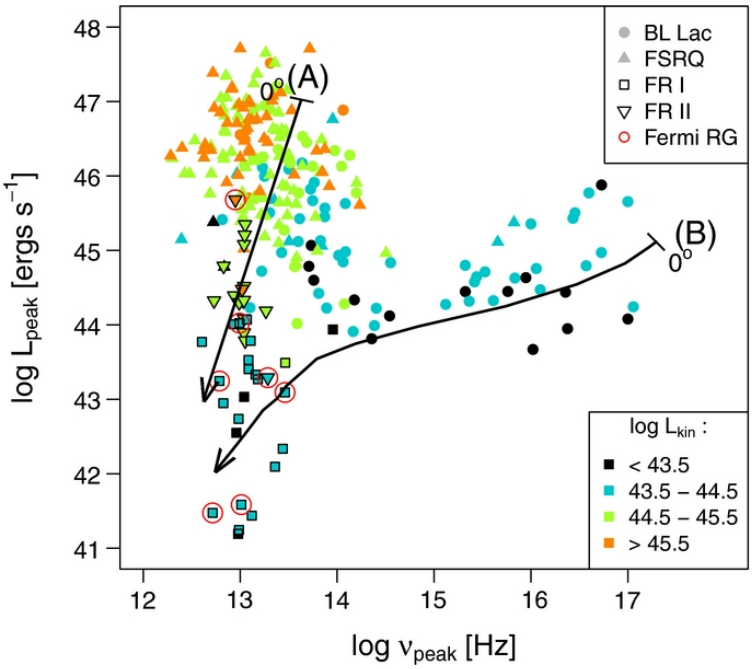


Figure 2.8: The "blazar envelope" of E. T. Meyer et al. (2011). The effect of orientation on two separate strong-jet (A) and weak-jet (B) populations is shown with black arrows. The apparent blazar sequence results from viewing only the high-luminosity envelope of the two populations.

VHE source population. Figure 2.9 shows the skymap and counts by source class of the 82 AGN that have been detected by IACTs at TeV energies to date. Over 60% of the objects are HBLs, with FSRQs making up only about 10%. By contrast, FSRQs are more commonly detected at GeV energies, comprising approximately 700 of 3100 AGN seen by *Fermi*-LAT (Ajello et al., 2020; Lott, Gasparrini, and Ciprini, 2020), and dominating the blazar population detected by EGRET (Mukherjee, 2001).

Figure 2.10 shows the redshift distribution of the TeV-detected blazars. 90% of the detected blazars have $z < 0.5$, with 65% of the objects having $z < 0.2$. By contrast, *Fermi*-LAT has detected multiple blazars with redshifts of $z > 3$ (Ajello et al., 2020). Part of this discrepancy can be attributed to the fact that the BL Lac objects that dominate the TeV source catalog have become more common at recent cosmological times, which we discuss further in Section 5.1, as well as to the larger sample size available to *Fermi*. However, the

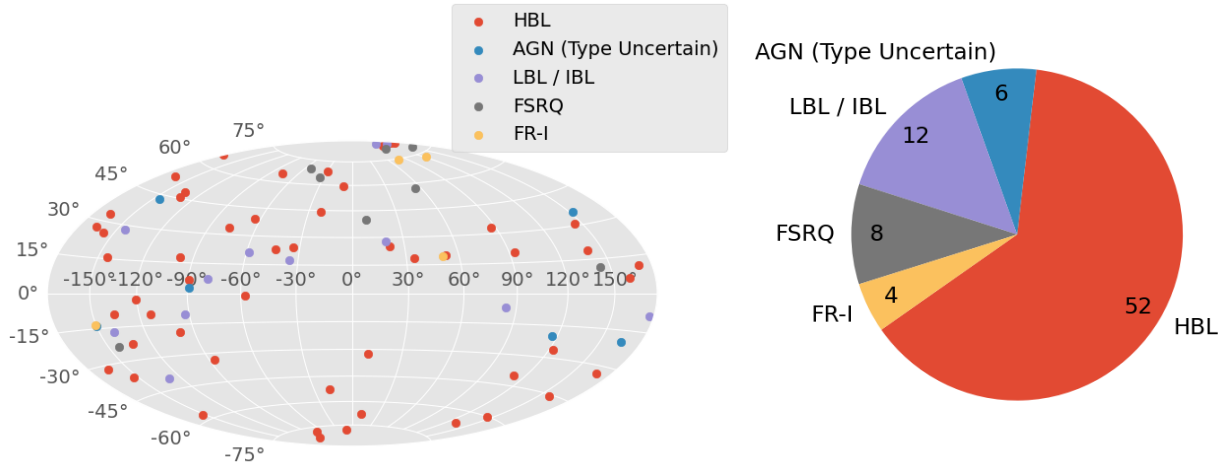


Figure 2.9: *Left*: Skymap in Galactic coordinates of the TeV-detected AGN reported in TeVCat. *Right*: Counts of TeV-detected AGN by source class.

absence of TeV-detected objects at high redshifts also results from an important selection effect, which we discuss next: the absorption of VHE gamma rays by intergalactic radiation fields.

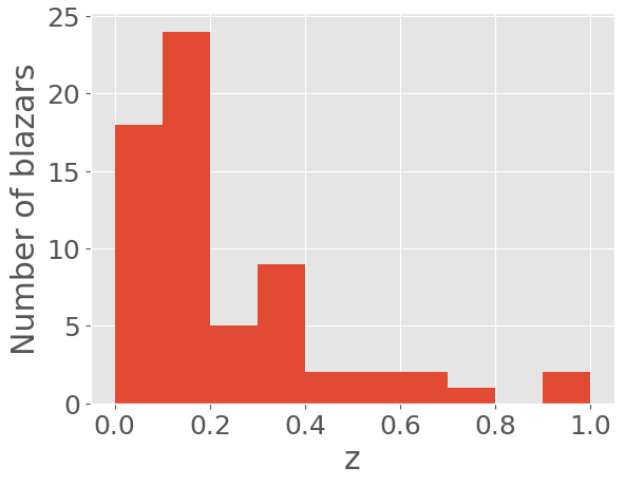


Figure 2.10: Redshift distribution of the TeV-detected blazars reported in TeVCat.

2.4 Blazars as a Cosmological Probe

2.4.1 Extragalactic Background Light

The star formation history of the Universe is encoded in the extragalactic background light (EBL), a radiation field consisting of all light ever emitted by stars. The EBL comprises an optical/ultraviolet component from unprocessed starlight and an infrared component from starlight absorbed and reradiated by interstellar dust. Gamma-ray photons can undergo absorption via pair production with the EBL, leaving an energy- and redshift-dependent imprint on blazar spectra (for a review, see Dwek and Krennrich, 2013). Figure 2.11 shows a schematic of the spectral energy distribution of the EBL along with that of the cosmic microwave background (CMB).

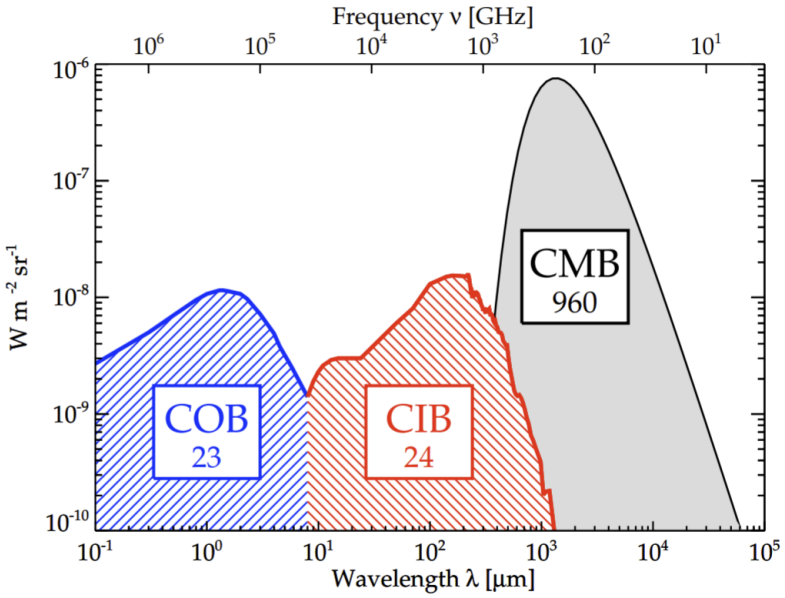


Figure 2.11: Spectral energy distributions of the EBL, including its cosmic optical background (COB) and cosmic infrared background (CIB) components, as well as the CMB, from Dole et al. (2006). The approximate intensity of each background in nW m^{-2} sr^{-1} is written in the boxes.

Multiple theoretical models have been developed to predict the EBL intensity as a func-

tion of wavelength and redshift (Finke, Razzaque, and Dermer, 2010; Domínguez et al., 2011; Gilmore et al., 2012; Franceschini, Rodighiero, and Vaccari, 2008; Franceschini and Rodighiero, 2017), taking into account star and galaxy formation and evolution, models of the dust distribution in galaxies, and radiative transfer. These models provide the expected optical depth of pair production with the EBL for a gamma ray of a given energy and source redshift. Directly measuring the EBL is difficult due to interplanetary and Galactic foregrounds, but lower limits can be placed on it by summing the expected light from resolved galaxies, and upper limits can be set by measuring the attenuation of gamma-ray spectra from a conservatively assumed intrinsic spectrum, particularly for a large sample of sources at different redshifts (e.g. Fermi-LAT Collaboration et al., 2018; Abeysekara et al., 2019).

2.4.2 Intergalactic Magnetic Field

An intergalactic magnetic field (IGMF) is present on scales at least as large as galaxy clusters, but its origin is unclear (for a review, see Durrer and Neronov, 2013). The IGMF is believed to have been produced by the amplification of a seed magnetic field by gravitational collapse during cosmological structure formation. The origin of the seed field is an open question, and it may have been formed during the early universe or by astrophysical processes contemporaneous with the amplification during structure formation. These scenarios can be told apart by constraining the IGMF strength and coherence length, although no direct measurements of these quantities exist. Observations of gamma-ray blazars can be used to probe the IGMF. When TeV gamma rays are attenuated by the EBL, they produce e^+e^- pairs that can inverse Compton scatter CMB and EBL photons, inducing an electromagnetic cascade (Protheroe and Stanev, 1993). Due to this process, TeV blazars would have an excess at GeV energies observable by *Fermi*-LAT unless the pairs are deflected by a sufficiently strong IGMF or dissipated by some other means such as plasma instabilities (Broderick, P. Chang, and Pfrommer, 2012; but see Sironi and Giannios, 2014). If the IGMF strength is

only moderate, the GeV emission could show up as a halo around TeV blazars. The non-detection of such excess GeV emission on or around TeV blazars has been used to place lower limits on the IGMF field strength (Finke et al., 2015; Ackermann et al., 2018). IACTs have excluded IGMF field strengths around 10^{-15} to 10^{-14} G through the non-detection of magnetically broadened emission from extremely hard-spectrum blazars where the cascade emission would appear in the VHE range (e.g. Archambault et al., 2017b), with the limits likely to improve further in the CTA era (M. Meyer, Conrad, and Dickinson, 2016).

Chapter 3: Imaging Atmospheric Cherenkov Telescopes: VERITAS and CTA

VHE gamma rays decay in the Earth's atmosphere into extensive air showers. Measuring these air showers with IACTs allows us to reconstruct the properties of the primary gamma rays that produced them. In this chapter, we introduce the concepts behind the imaging atmospheric Cherenkov technique, review how IACT data is analyzed, and describe the VERITAS and CTA observatories, including the development of the prototype SCT.

3.1 Extensive Air Showers

3.1.1 Gamma-ray Air Showers

VHE gamma rays decay via interactions with the nuclei in air, forming an extensive air shower. Above the critical energy $E_0 \approx 84$ MeV, the dominant decay process is pair production occurring in the electromagnetic field of a nucleus, $\gamma \rightarrow e^+ + e^-$. The presence of a nucleus is necessary to conserve energy and angular momentum. Ionization losses become dominant below E_0. The secondary particles split the energy of the primary particle. The secondary electrons radiate primarily through Bremsstrahlung radiation in the presence of air nuclei to produce secondary gamma rays. These processes alternate until the secondary particle energy drops below E_0. The particles also undergo Coulomb scattering, which causes the shower to take on a small lateral extent. The only particles produced are photons, electrons, and positrons. The characteristic length scale of energy loss by these processes in a given material is given by the radiation length, which has units of density times distance.

One radiation length is the distance required for an electron to lose all but $1/e$ of its energy by Bremsstrahlung, and is 7/9 of the mean free path for pair production. The radiation length in air for an electron is $X_0 \approx 37$ g cm^{-2}.

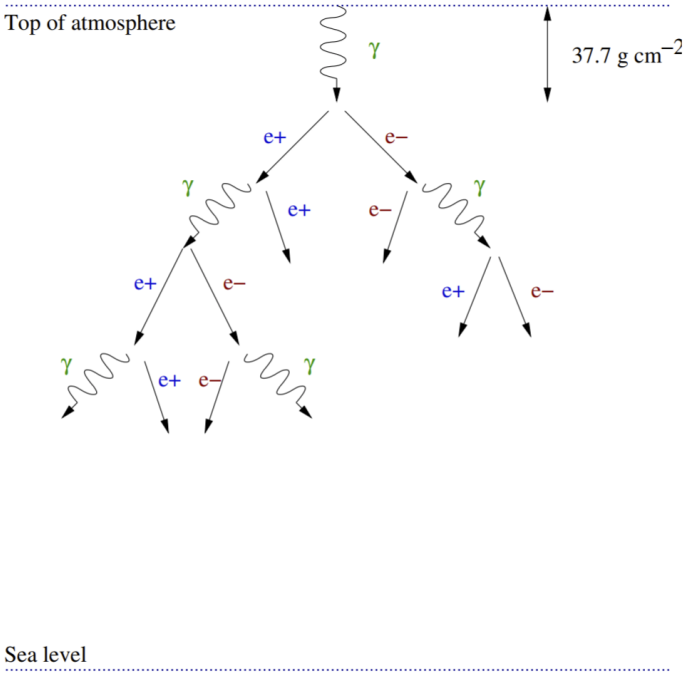


Figure 3.1: Schematic illustration of a gamma-ray-initiated air shower in the atmosphere, from Gammell (2004).

Figure 3.1 illustrates the development of an electromagnetic air shower. A simple model of electromagnetic air shower development was proposed by Heitler (1954). The process starts with a primary gamma ray of energy E, which enters the atmosphere and reaches its height of first interaction after an average of one radiation length. After every radiation length, each particle in the shower splits into two secondary particles, each with half as much energy, until the particle energy falls below E_0. The maximum number of particles generated in the shower is $N_{\max} = E/E_0$ and the point of maximum shower development is $X_{\max} = \log_2(E/E_0)X_0$. The Heitler model tells us that $N_{\max} \propto E$ and $X_{\max} \propto \log E$, suggesting that these observable variables can be used to reconstruct the shower energy.

3.1.2 Cosmic-ray Air Showers

Cosmic rays also create extensive air showers when they enter the atmosphere. In fact, the vast majority of all air showers detectable by IACTs are initiated by cosmic rays, except at the highest energies. Cosmic rays are predominantly hadrons, mostly protons, along with some heavier nuclei. A subdominant population of cosmic-ray electrons is also present. Figure 3.2 illustrates the development of hadronic air showers, which is much more complex than that of electromagnetic showers. Interactions with nuclei in the air produce secondary nucleons and many pions. The pions subsequently decay into gamma rays, neutrinos, and muons (cf. Eq. 2.16). While low-energy muons quickly decay into electrons, high-energy muons can survive long enough to reach the ground because of time dilation. The transverse momenta of secondary particles in hadronic showers can be much larger than in electromagnetic showers, and hadronic showers may produce multiple electromagnetic sub-showers. Due to these diverse possible interactions, the appearance of cosmic-ray showers can vary greatly from event to event. Hadronic showers are initiated at lower altitudes than electromagnetic showers, as the mean free path for a proton in air is $X_0 \approx 80$ g cm^{-2}.

3.1.3 Cherenkov Radiation

The energetic charged particles in extensive air showers move faster than the speed of light in air and therefore emit Cherenkov radiation. This phenomenon is illustrated in Figure 3.3. A charged particle passing through a dielectric medium excites the surrounding particles, which emit photons upon returning to the ground state. If the particle moves slowly, the emitted radiation is spherically symmetric and cancels itself out, but if the particle is moving faster than the speed of light in the medium, the emission interferes constructively, producing a strong, narrow wavefront in the direction of the particle's motion. The Cherenkov radiation appears as a narrow cone with opening angle $\theta = 1/\beta n$, where the particle velocity is βc and

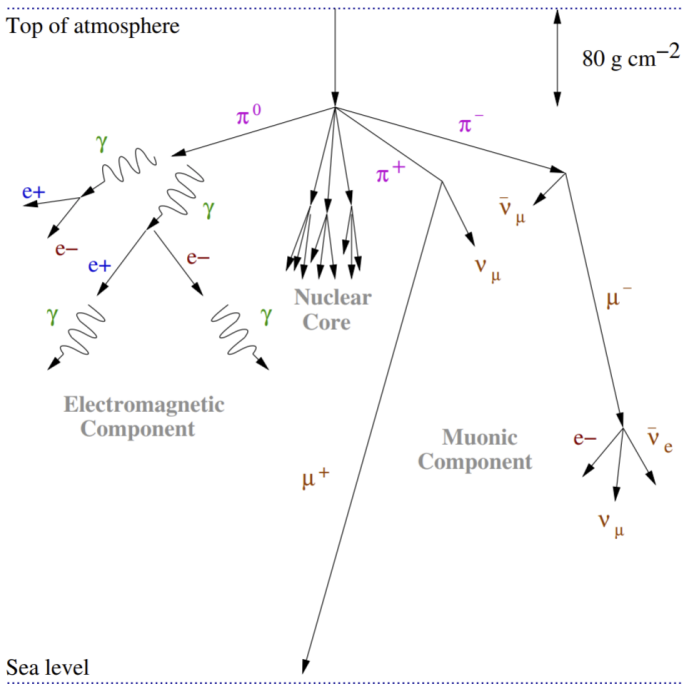


Figure 3.2: Schematic illustration of a hadronic air shower in the atmosphere, from Gammell (2004).

n is the refractive index of the medium. At sea level, $\theta \approx 1.3°$.

The charged particles in an extensive air shower have nonzero transverse momenta, and their collective Cherenkov light forms a light pool on the ground. The refractive index of air scales with density, causing the Cherenkov opening angle to increase as the particle descends. As a result, the lateral size of the Cherenkov light pool is roughly fixed at all altitudes, with a radius on the ground of about 130 m, as demonstrated using Monte Carlo simulations in Figure 3.4. The Cherenkov light arrives on the ground mostly within the span of a few nanoseconds, with a spectrum peaking in the ultraviolet, at about 350 nm, once atmospheric extinction is taken into account. A image of the Cherenkov shower taken from within its light pool resembles an ellipse, with major and minor axes determined by the longitudinal and lateral development of the shower, respectively, modified by the position and orientation of the shower relative to the telescope.

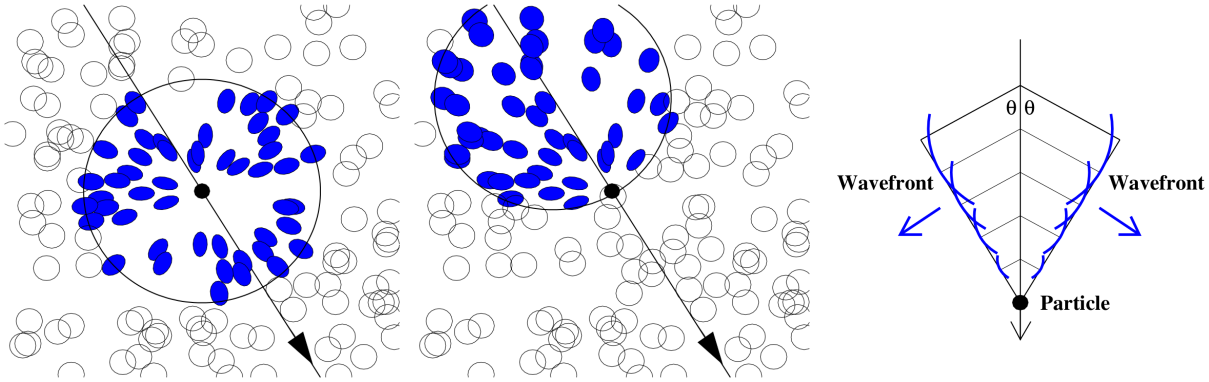


Figure 3.3: Conceptual diagram of Cherenkov radiation, from Cogan (2006). *Left*: Destructive interference from a non-relativistic particle. *Middle*: Constructive interference from a relativistic particle. *Right*: Geometric representation of the wavefront produced by Cherenkov radiation.

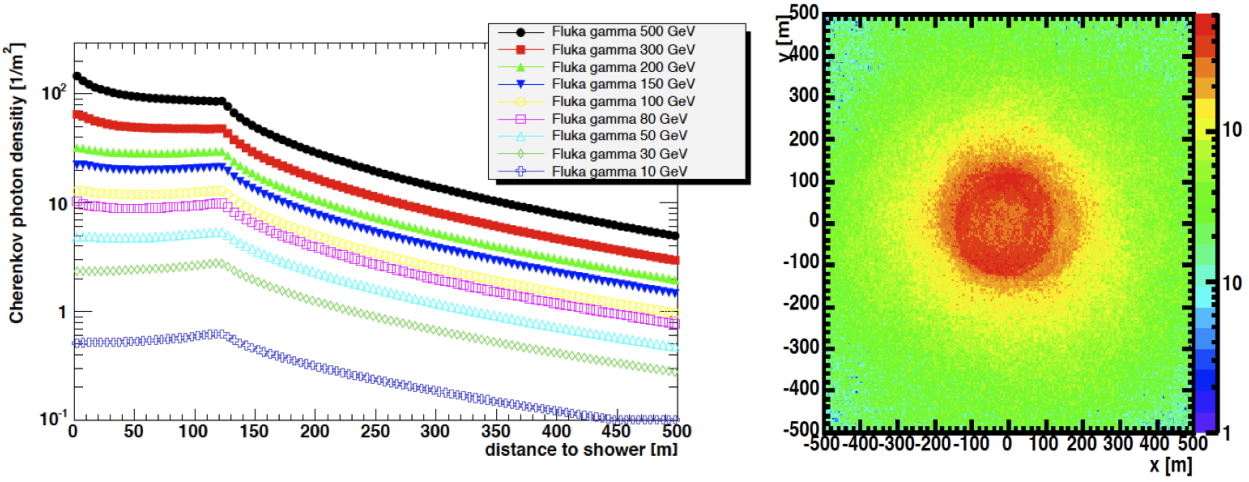


Figure 3.4: *Left*: Cherenkov photon density as a function of distance from the shower core derived from simulations, showing an approximately constant density within 130 m and a rapid decrease beyond that. *Right*: Simulated positions on the ground of Cherenkov photons generated by a 300 GeV primary. Figures by Gernot Maier.

3.2 The Imaging Atmospheric Cherenkov Technique

In the IACT technique, a mirror is used to focus Cherenkov light onto a pixelated camera which captures an image of the shower. In one sense, IACTs are essentially optical telescopes, and operate in a similar manner. Observations must be conducted at night, under clear, dark skies. Clouds or rain prevent observations, as does bright moonlight. However, IACTs also have very different design requirements from standard optical telescopes. In particular, in order to image Cherenkov showers, nanosecond time resolution is required, as is a very large field of view (FoV) of several degrees or more. On the other hand, the angular resolution demanded of the mirrors and camera is much less than for an optical telescope, as shower fluctuations place a fundamental limit of about a few arcminutes on the useful optical angular resolution[1] (Hofmann, 2006). For more information on the methods used by imaging atmospheric Cherenkov telescopes, the reader is referred to the review by Holder (2015).

In measuring an air shower event with an IACT, the key observable properties of the primary particle are the particle type, arrival direction, core position, and energy. Figure 3.5 shows an example image taken by a VERITAS telescope. We discuss in Section 3.4.4 how the shower morphology can be used to characterize the primary particle type. The telescope's physical position relative to where the shower landed on the ground (the shower core position) determines where the image is located in the camera. At the same time, the coordinates in the camera plane correspond to angular coordinates on the sky. The major axis of an elliptical parameterization of the image should pass through the angular coordinates of the shower origin. A single shower image therefore constrains the shower arrival direction to a track along the sky.

Triangulating the images taken by multiple telescopes breaks this degeneracy (Hofmann et al., 1999). The principle of stereoscopic event reconstruction is illustrated in Figure 3.6.

[1] This fundamental limit is approximately matched by the pixel spacing of the SCT (Section 3.6).

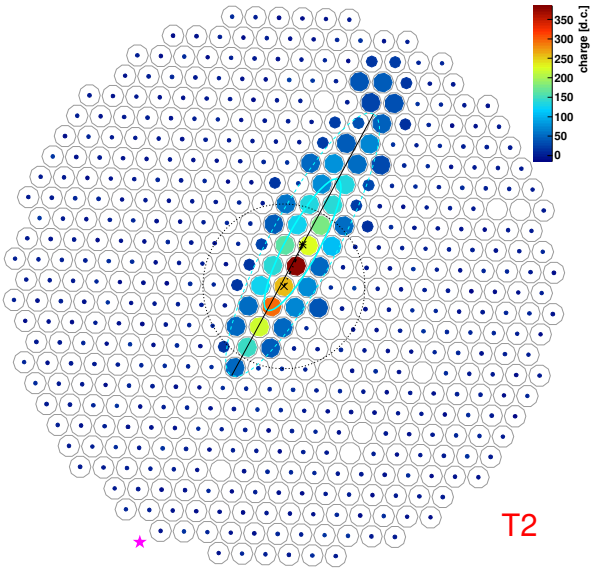


Figure 3.5: An example shower image from a VERITAS telescope. The image has been cleaned to remove the signal in pixels without any Cherenkov light, and a simple moment-based elliptical fit is overplotted in light blue.

With multiple telescopes capturing different views of the same event, the intersection points of the images' major axes determine the event arrival direction and core position when the intersection is performed in camera coordinates and in physical coordinates, respectively. The brightness of an image increases with shower energy and decreases with distance to the shower core position. A good estimate of the core position is therefore needed to accurately measure the shower energy, which is proportional to the primary particle energy. If the shower core position is far from the telescopes, the images will be close to parallel. In this case, or when doing single-telescope analysis, the arrival direction can be estimated from the shower morphology using the *displacement* method, which exploits the geometrical relationship between the shower's elongation and the angle at which it is viewed (Lessard et al., 2001).

The more telescopes in an IACT array, the better. Studies on the design and layout of IACT arrays have found that the sensitivity increases roughly as the square root of the

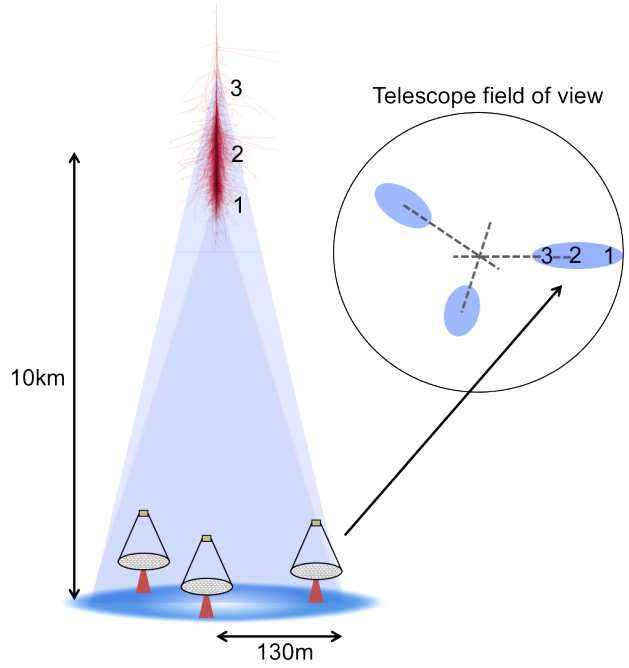


Figure 3.6: Illustration of event reconstruction with an array of IACTs, from Holder (2015).

number of telescopes, and that the optimal spacing depends on the energy, with a wider spacing preferred for higher energies (Hofmann et al., 1999; Bernlöhr et al., 2013). It is important to realize that because a shower can be imaged from anywhere within its light pool, the effective area of an IACT array is not limited by the total mirror area, but rather by the characteristic size of the Cherenkov light pool along with the number and layout of telescopes in the array. IACTs therefore have effective areas of order $\pi(130\text{ m})^2 \sim 10^5\text{ m}^2$, compared to $\sim 1\text{ m}^2$ for space telescopes! This fact allows IACTs to remain sensitive to astrophysical sources that have steeply falling power-law spectra. However, the mirror area is also important, as it determines the minimum energy threshold of the detector.

3.3 VERITAS

The VHE gamma-ray data analyzed in this work was collected using the VERITAS array of IACTs. The four 12-meter VERITAS telescopes at FLWO ($+31°\ 40'\ 30.21''$, $-110°\ 57'\ 7.77''$)

have a Davies-Cotton optical design (Davies and Cotton, 1957). The optical system of each telescope consists of 350 identical, hexagonal, aluminum-coated glass mirror facets aligned to produce an approximately spherical reflector. This design is relatively inexpensive to produce, but has the downside of inducing spread in the arrival time of the Cherenkov photons on the order of a few nanoseconds. Each telescope's camera consists of a hexagonally spaced array of 499 photomultiplier tubes (PMTs). The empty space between PMTs is recovered with Winston light cones (Winston, 1970), shown in Figure 3.7. The VERITAS camera has a 3.5° FoV with a pixel spacing of 0.15°. Data readout is performed using flash analog-to-digital converters that sample the signals from the PMTs at a 500 MHz rate. In the standard operating mode, 24 samples, corresponding to 48 ns, are read out for each event, with an 8-bit dynamic range per sample.

Figure 3.7: A view of the Winston light cones on top of the PMTs in the VERITAS camera. In normal operation, all PMTs are covered with light cones. From the VERITAS webpage, `https://veritas.sao.arizona.edu/about-veritas/atmospheric-cherenkov-technique-and-veritas-technologies`.

Fluctuations in the night sky background level as well as Cherenkov light radiated by high-energy cosmic-ray muons close to the ground can produce spurious background signals

in individual telescopes. VERITAS uses a three-level hardware trigger system to eliminate these backgrounds, referred to as L1, L2, and L3 triggers. The L1 trigger is a pixel-level trigger, in which each PMT is connected to a constant-fraction discriminator (CFD) that issues a trigger if the charge depositied in the PMT passes a pre-determined threshold. The CFD threshold is optimized for sensitivity to the faintest possible Cherenkov showers while still ignoring night sky background fluctuations (a similar procedure is performed for the pSCT; see Section 6.1.1). The L2 trigger further cuts down on noise by requiring three neighboring pixels to trigger within a coincidence window of about 5 ns (Zitzer and VERITAS Collaboration, 2013). In order to eliminate the background from local muons, the L3 trigger requires that a signal be present in at least two telescopes for an event to be recorded (Weinstein, 2008).

VERITAS is sensitive to gamma rays from about 100 GeV to over 30 TeV, although the energy threshold increases for sources observed at low elevation. The energy resolution is about 20%, and the gamma-ray angular resolution, defined as 68% containment radius, is about 0.08° at 1 TeV, increasing to 0.13° at 200 GeV. The sensitivity of VERITAS as a function of exposure time is shown in Figure 3.8. VERITAS is located at an altitude of 1268 m and, given its latitude of about 31°, is able to observe sources with declinations between approximately $-10°$ and 70°. VERITAS typically acquires about 70-100 hours of data per month, totaling approximately 1000 hours per year. The observatory does not operate during Arizona's summer monsoon season. About 20% of the data that is normally collected annually is taken with a higher energy threshold under partial moonlight conditions.

VERITAS regularly monitors known and candidate TeV blazars and follows up flaring events seen in its own and multiwavelength observations (e.g. Benbow, 2019). In particular, VERITAS preferentially performs observations of FSRQs when they exhibit an elevated flux in other wavebands, as a flare at TeV energies might also be occurring. VERITAS also carries out short monitoring observations of FSRQs. Because these sources are not believed to be

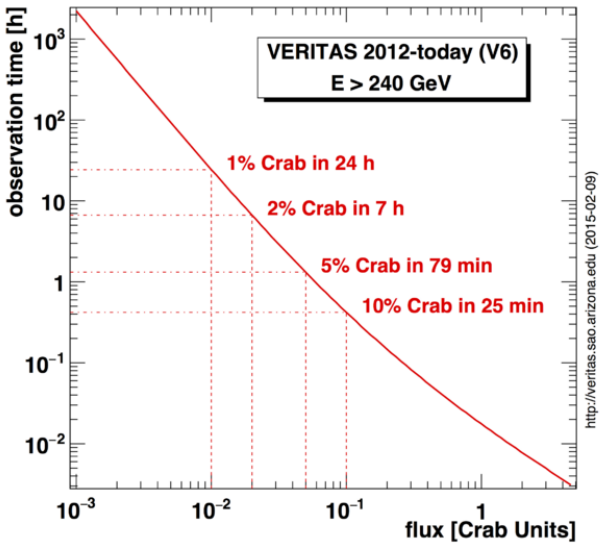


Figure 3.8: Sensitivity of the VERITAS array as a function of exposure time, from the VERITAS webpage, `https://veritas.sao.arizona.edu/about-veritas/veritas-specifications`.

strong emitters of TeV gamma rays except during flares, the primary aim of these monitoring observations is to self-trigger on serendipitous flares.

3.4 Analyzing IACT Data

IACT data analysis can be factorized into independent stages. Each stage can be thought of as transforming the data from a lower to a higher data level (Deil et al., 2017): raw traces or waveforms (DL0); calibrated images and, usually, extracted parameters (DL1); reconstructed showers (DL2); gamma-ray event lists (DL3); science products such as sky maps, spectra, and light curves (DL4); and source catalogs (DL5).

The analysis of VERITAS data is performed using two different software packages, EventDisplay (Maier and Holder, 2017) and the VERITAS Gamma-ray Analysis Suite (VEGAS; Cogan, 2008). The VERITAS analysis presented in this work was performed using EventDisplay. It has been cross-checked with independent analyses using VEGAS. EventDisplay consists of three tools, *evndisp*, *mscw_energy*, and *anasum*, that perform different stages of

the analysis. Table 3.1 breaks down the analysis stages, giving their corresponding EventDisplay tools and data levels. For CTA data analysis, the *ctapipe* software package is being developed (Kosack et al., 2020b).

Analysis stage	EventDisplay tool	Data level
Calibration	*evndisp*	DL0 → DL1
Image Analysis	*evndisp*	DL1
Event Reconstruction	*mscw_energy*	DL1 → DL2
Background Rejection	*anasum*	DL2 → DL3
Results Extraction	*anasum*	DL3 → DL4

Table 3.1: Stages of an IACT data analysis.

3.4.1 Calibration

Each pixel's raw digitized samples are converted into an integrated charge. This is done using a trace integration algorithm, which must be optimized to extract as much signal as possible without excessive dilution by background electronics noise (e.g. Cogan, 2006). This stage also involves applying relative gain corrections, which are determined by uniformly illuminating the camera with pulses from an LED "flasher" (Hanna et al., 2010), as well as timing corrections required due to differences in the cabling path length in each channel. Broken or noisy pixels are also excluded from the analysis at this stage.

3.4.2 Image Analysis

Next, the calibrated camera images are processed to extract relevant features for the analysis. In a standard analysis, the images are first cleaned and then parameterized. In the standard two-level cleaning method, a pixel is retained if its charge exceeds either a specified *image threshold* (making it an image pixel) or a lesser *border threshold* if it is adjacent to an image pixel (Hillas et al., 1998). Next, a number of parameters (Hillas parameters) are derived from each image (Hillas, 1985). The Hillas parameters, pictured in Figure 3.9,

include the second moments of the image - its *length* and *width* - along with measures of the shower's total size and its orientation and position in the camera.

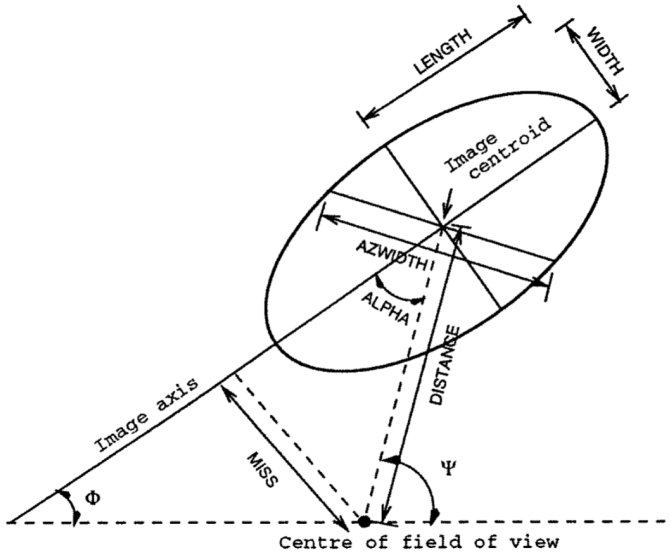


Figure 3.9: Hillas parameterization of a shower image, from D. J. Fegan (1997).

Hillas parameterization is a simple and computationally efficient way to summarize the basic characteristics of an IACT image, but it has shortcomings. Most importantly, it discards most of the detail in the image, potentially wasting information. This is particularly relevant when using a high-resolution camera. The image moments depend strongly on the cleaning level, potentially introducing a tradeoff between optimizing for event reconstruction and background rejection. Also, moment-based parameterization cannot handle incomplete images that are cut off at the edge of the camera, although this is less of an issue for likelihood-based fitting methods. For these reasons, it is worth studying methods potentially able to extract more useful information from the images. In particular, deep learning methods can operate directly on calibrated camera images, going straight from DL1 to DL3 data and skipping parameterization entirely. We explore these techniques in Chapter 7.

3.4.3 Event Reconstruction

Event reconstruction using telescope-level parameters can be done using stereoscopic reconstruction or the displacement method, as described in Section 3.2. Due to intrinsic shower fluctuations and measurement errors, the best reconstruction is not uniquely determined. The arrival direction and core position can be estimated in several ways, such as by finding the point that minimizes the distance to the image axes of all of the telescopes, or by taking the mean of pairwise intersection points. The contribution from each telescope or pair of telescopes should be weighted to favor telescopes with brighter, more informative images, as well as pairs of telescopes whose image axes are perpendicular.

3.4.4 Background Rejection

The sensitivity of IACTs depends strongly on efficiently rejecting the background of much more numerous cosmic-ray showers, which resemble those produced by gamma rays but tend to have a more complex morphology. Particle classification is therefore based on the image shape. Hadronic showers can be distinguished from gamma-ray showers by their larger and more irregular overall lateral extent, the possession of multiple electromagnetic sub-showers, and the presence of rings emitted by high-energy muons close to the telescope. Parametrically, the images of these showers tend to have a larger *width*. More subtle properties that could potentially be exploited by a high-resolution detector and sophisticated analysis method are illustrated in Figure 3.10, including low-intensity Cherenkov light emitted by the hadronic particles in the showers and direct Cherenkov emission from the primary particle emitted before the actual shower development (de Naurois and Rolland, 2009). Most cosmic-ray showers are initiated by protons, and the showers initiated by heavier nuclei are typically easier to distinguish from gamma-ray showers (Archer et al., 2018). Cosmic-ray rejection can therefore be approximated by the task of gamma/proton classification.

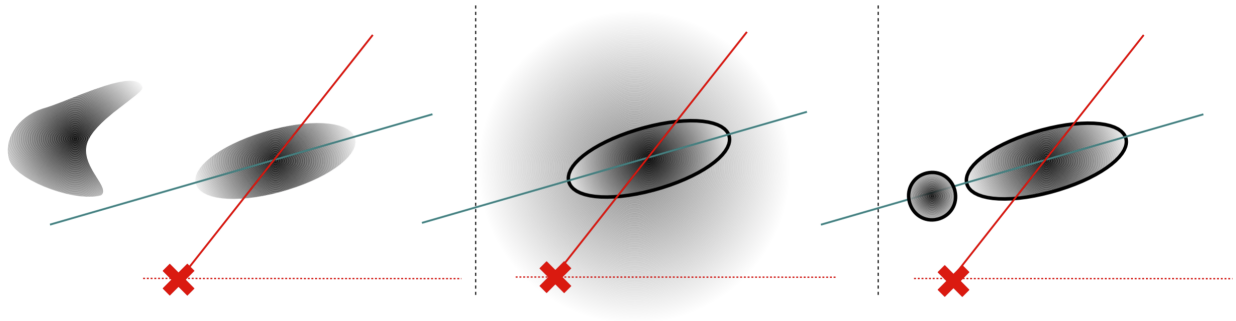


Figure 3.10: Illustration of features distinguishing hadronic from gamma-ray showers, from de Naurois and Rolland (2009). Left: image clusters corresponding to electromagnetic sub-showers. Middle: low-intensity Cherenkov light emitted by the hadronic portion of the shower. Right: Cherenkov emission emitted when the primary particle enters the atmosphere prior to the actual shower development.

Although most cosmic-ray showers are hadronic, at very high cut efficiencies, the showers initiated by cosmic-ray electrons can become a relevant population. These showers are purely electromagnetic, and therefore very difficult to distinguish from gamma-ray showers. Two subtle differences theoretically enable some statistical separation: first, electron-initiated showers start emitting Cherenkov light about one radiation length higher in the atmosphere, and second, these showers contain an additional component due to the Cherenkov radiation emitted by the primary electron (Hofmann, 2006). However, with current instruments, this background is usually considered irreducible.

Like event reconstruction, background rejection is generally performed using stereo variables constructed from the telescope parameters. In EventDisplay, the shower morphology is described using the mean reduced scaled width ($MRSW$) and length ($MRSL$), defined as (Krause, Pueschel, and Maier, 2017)

$$MRSW = \frac{1}{N_{\text{images}}} \sum_{i=1}^{N_{\text{images}}} \left(\frac{w_i - \hat{w}_{\text{MC}}(R_i, s_i)}{\sigma_{w_{MC}(R_i,s_i)}} \right) \left(\frac{\hat{w}_{\text{MC}}(R_i, s_i)}{\sigma_{w_{MC}(R_i,s_i)}} \right)^2, \tag{3.1}$$

and

$$MRSL = \frac{1}{N_{\text{images}}} \sum_{i=1}^{N_{\text{images}}} \left(\frac{l_i - \hat{l}_{\text{MC}}(R_i, s_i)}{\sigma_{l_{MC}}(R_i, s_i)} \right) \left(\frac{\hat{l}_{\text{MC}}(R_i, s_i)}{\sigma_{l_{MC}}(R_i, s_i)} \right)^2, \tag{3.2}$$

where for each telescope i, w and l are the Hillas *width* and *length* parameters, R is the distance to the shower core position, and s is the *size*, the total charge deposited in the camera, which is a proxy for shower energy. The $\hat{w}_{\text{MC}}$ and $\hat{l}_{\text{MC}}$ parameters are the medians of the width and length distributions derived from simulations, and $\sigma_{w_{MC}}$ and $\sigma_{l_{MC}}$ are the standard deviations of those distributions. Other useful parameters include the weighted mean height of maximum shower development, determined using the parallax distance calculated with all pairs of telescopes, and the size of the largest or second-largest image.

The simplest classification method using these parameters is to apply *box cuts* by placing an independent selection threshold on each parameter without taking into account any possible nonlinear interactions or correlations. To go beyond box cuts, a number of different classification methods based on machine learning have been developed, and are reviewed in Section 7.1. The cut thresholds that maximize the sensitivity depend on the strength and spectrum of the source. In EventDisplay, separate cuts are provided that are optimized for sources expected to have soft, medium, and hard spectra.

3.4.5 Results Extraction

We can now assess if the data contain a signal. In gamma-ray astronomy, the night sky background varies depending on the weather, telescope elevation, and positions of the Moon and stars. We must therefore estimate the background rate from the data and take the uncertainty on this estimation into account when calculating the statistical significance of the signal. If we measure N_{on} counts from the source region and N_{off} counts from a control region, the observed signal is

$$N_S = N_{\rm on} - \alpha N_{\rm off}, \tag{3.3}$$

where $\alpha = t_{\rm on}/t_{\rm off}$ is the ratio of exposure taken on each region. In practice, one usually estimates the "on" and "off" counts from the same run, using different regions of the FoV.

Two commonly used background models are ring background and reflected regions, demonstrated in Figure 3.11. In the ring background model, the control region is taken to be a ring around the source region, while in the reflected regions model, it consists of a set of separate regions at the same distance from the camera center as the source region. The camera acceptance is radially dependent, so energy-dependent acceptance corrections must be applied when using the ring background model. For this reason, the reflected regions model is a better choice when measuring energy spectra. When using these models, the regions surrounding any other potential gamma-ray sources in the FoV are excluded, as are the positions of bright stars that could produce spurious signals.

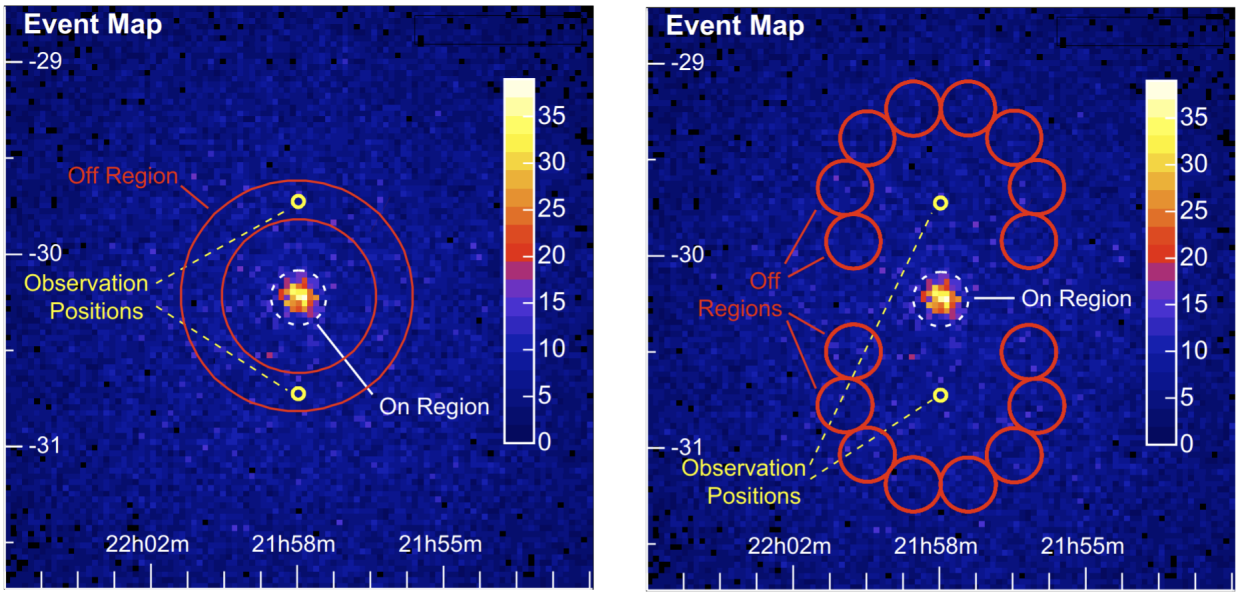


Figure 3.11: Schematic illustration of the ring background (*left*) and reflected regions (*right*) background models, from Berge, Funk, and J. Hinton (2007).

The significance can now be estimated using Equation 17 of T.-P. Li and Ma (1983),

$$S = \sqrt{2}\left\{N_{\rm on}\ln\left[\frac{1+\alpha}{\alpha}\left(\frac{N_{\rm on}}{N_{\rm on}+N_{\rm off}}\right)\right] + N_{\rm off}\ln\left[(1+\alpha)\left(\frac{N_{\rm off}}{N_{\rm on}+N_{\rm off}}\right)\right]\right\}^{1/2}. \tag{3.4}$$

A approximation to this formula, valid when the numbers of on and off counts are not too small, is Equation 9 of T.-P. Li and Ma (1983),

$$S = \frac{N_{\rm on} - \alpha N_{\rm off}}{\sqrt{\alpha(N_{\rm on}+N_{\rm off})}}. \tag{3.5}$$

If the significance meets the threshold for detection of $S = 5$ standard deviations, we can now estimate the flux of the source and derive high-level data products, such as a light curve, spectrum, and skymap. The effective area of the detector derived from simulations is used to convert the measured number of counts into a photon flux (cm^{-2} s^{-1}) or energy flux (erg cm^{-2} s^{-1}). Fluxes are sometimes also reported in Crab Units, where 1 Crab is the flux of the Crab Nebula in the energy range being considered.

3.5 Cherenkov Telescope Array (CTA)

CTA, the next-generation observatory for VHE gamma-ray astronomy, will improve on the sensitivity of current-generation instruments by an order of magnitude and cover energies from 20 GeV to more than 300 TeV using large arrays of telescopes in three sizes: large-sized telescopes (LSTs), medium-sized telescopes (MSTs), and small-sized telescopes (SSTs), designed to access gamma rays at low, medium, and high energies, respectively (e.g. Acharya et al., 2013; Hassan et al., 2017; Acharyya et al., 2019). The telescopes being developed for CTA are illustrated in Figure 3.12, and the expected sensitivity of CTA compared to current-generation instruments, including VERITAS and *Fermi*-LAT, is shown in Figure 3.13.

The CTA Observatory will consist of two separate arrays, allowing it to operate in both the Northern and Southern Hemispheres. The Northern site will be located at the Roque de los Muchachos Observatory on the island of La Palma in the Canary Islands, Spain, and will contain 4 LSTs and 15 MSTs in the baseline array configuration. The Southern site will be located near the Paranal Observatory in the Atacama Desert in Chile, and will contain 4 LSTs, 25 MSTs, and 70 SSTs in the baseline array configuration. The CTA Observatory will be operated as a open observatory, accepting guest observer proposals and publicly releasing data once a CTA Consortium proprietary period has elapsed.

The CTA telescopes feature upgraded technology compared to current-generation telescopes. The LST has a 23-meter-diameter parabolic mirror and a camera with 1855 PMTs, providing a 4.3° field of view (FoV). With its large mirror area, the LST is well suited to capture images from low-energy gamma rays that produce little Cherenkov light, making it the primary driver of the full system sensitivity to gamma rays between 20 and 150 GeV.

Several designs are being developed for the MST, CTA's "workhorse" telescope driving the sensitivity in the core energy range of 150 GeV – 5 TeV. An 11.5 m single-mirror telescope is being developed with two possible camera systems, NectarCam and FlashCam, each containing approximately 1800 PMTs and providing a FoV of approximately 7.5°. The other candidate MST for CTA is the Schwarzschild-Couder Telescope (SCT), which features an innovative dual-mirror optical system and high-resolution camera based on silicon photomultipliers (SiPMs). The SCT and its component technologies are discussed in detail in Section 3.6 below.

Finally, the SST will be both the smallest and most numerous of the CTA telescopes, making it the most suitable to detect the extremely bright but rare showers initiated by gamma rays of energies between 5 and 300 TeV. While the research described in this work was being conducted, three SST designs were being developed for CTA: ASTRI-Horn, GCT-CHEC, and SST-1M. SST-1M is a conventional single-mirror design using a PMT-based

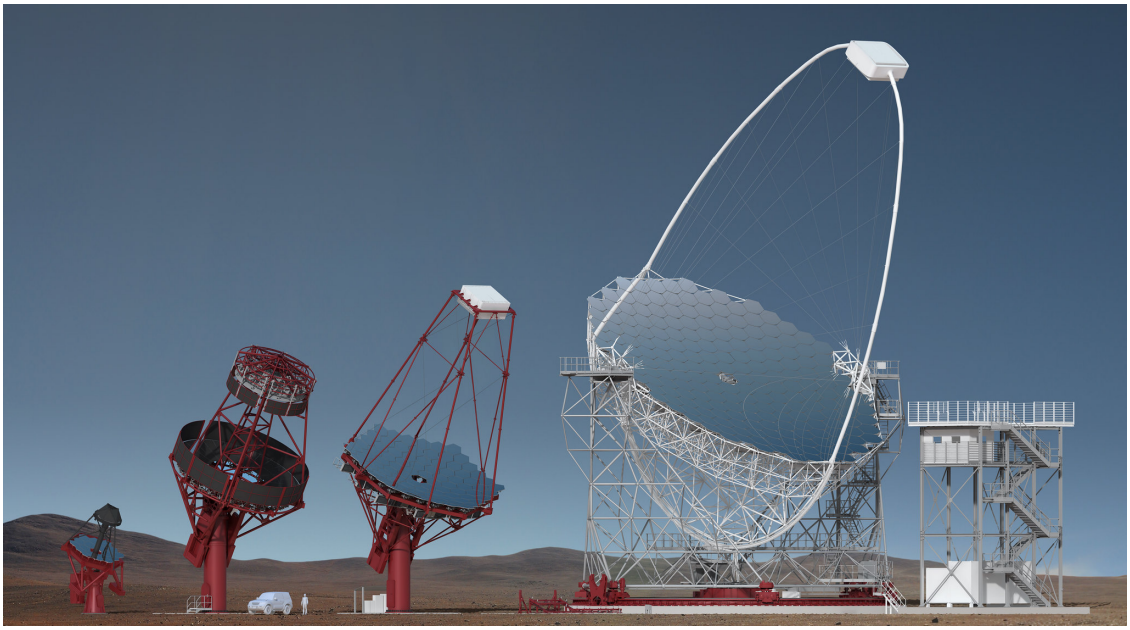


Figure 3.12: Rendering of the telescopes being developed for CTA, from Diaz (2020). Left to right: SST, SCT, single-mirror MST, LST.

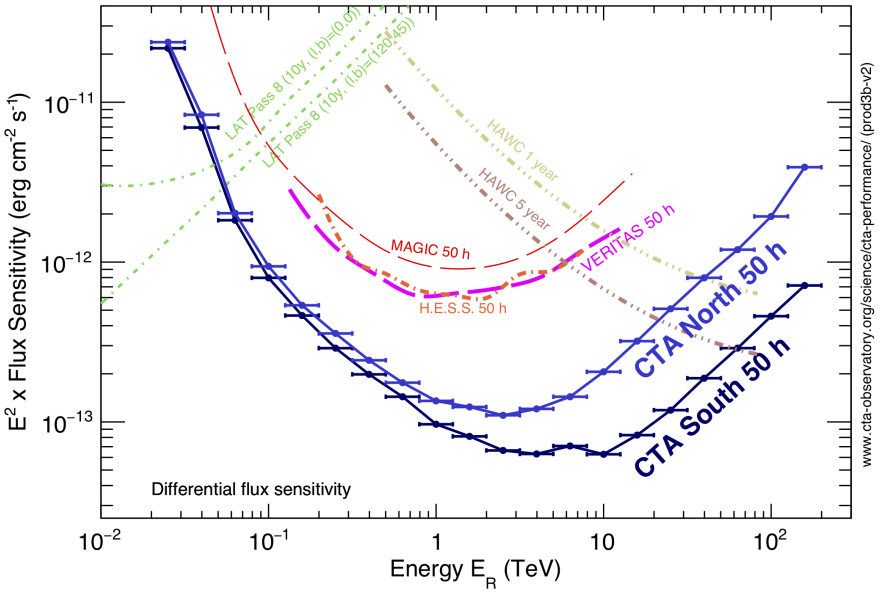


Figure 3.13: Differential flux sensitivity of CTA, compared to current-generation instruments, from CTA Observatory (2019). The differential sensitivity is defined as the minimum flux needed by CTA to obtain a 5σ detection of a point source. The CTA performance curves are derived from simulations similar to those presented in Bernlöhr et al. (2013).

camera. ASTRI-Horn, the telescope design of the *Astrofisica con Specchi a Tecnologia Replicante Italiana* (ASTRI) Project (Lombardi et al., 2020), and GCT-CHEC, a telescope using the Gamma Cherenkov Telescope (GCT) optical system (Blanc et al., 2018) and the Compact High-Energy Camera (CHEC) camera (Zorn et al., 2018), are dual-mirror designs using SiPM-based cameras, similar to the SCT. In June 2019, it was decided that a single SST design will be developed for CTA combining the ASTRI optical system and CHEC camera, taking into account the experience gained from all designs.

3.6 Schwarzschild-Couder Telescope (SCT)

The SCT has an optical system with two aspheric mirrors, in which light reflects off of the larger (9.7 m) primary mirror onto the smaller (5.4 m) secondary mirror and is in turn reflected onto the camera focal plane (Vassiliev, S. Fegan, and Brousseau, 2007). The optical system is segmented. The primary mirror is divided into an inner ring with 16 panels and an outer ring with 32 panels, and the secondary mirror is similarly composed of an 8-panel inner ring and a 16-panel outer ring. The alignment of these mirrors is a complex operation which is enabled by a system of stepper motors and edge sensors connected to the mirror panels (C. Adams et al., 2020a,b).

The Schwarzschild-Couder design has several advantages over Davies-Cotton telescopes. First, the optical system is aplanatic, or free of spherical and coma aberrations, which widens the potential FoV of the camera. In addition, the plate scale at the focal plane is greatly reduced, allowing the SCT to use a finely-pixelated camera composed of an array of high density SiPMs, which have higher photon detection efficiency compared to PMTs. The camera's 0.067° pixelation allows it to record images of Cherenkov showers at high resolution, improving the instrument's energy estimation, angular resolution, and background rejection, while maintaining a large 7.6° FoV. The improved angular resolution increases the sensitivity

to point sources, and in combination with the wide FoV, enables detailed mapping of spatially extended sources. The wide FoV also enhances the efficiency of survey observations and follow-up searches for poorly localized transient events, such as gamma-ray bursts, high-energy neutrinos, and gravitational waves. In particular, the off-axis performance of the SCT is greatly improved compared to the single-mirror MST. Replacing all MSTs in CTA with SCTs would improve the off-axis angular resolution of the array by 40% and halve the observation time required to achieve a given off-axis sensitivity, especially at low energies.

Figure 3.14: The prototype Schwarzschild-Couder telescope installed at the Fred Lawrence Whipple Observatory, from C. B. Adams et al. (2021).

A prototype SCT (pSCT) has been constructed alongside VERITAS at FLWO. Figure 3.14 shows a picture of the pSCT, which is situated about 35 meters from the nearest

VERITAS telescope. The pSCT was inaugurated and took first light data in January 2019, and detected the Crab Nebula during observations in January and February 2020 (C. B. Adams et al., 2021).

The SCT camera has a modular, hierarchical design, in which the focal plane is divided into nine sectors. Each sector contains either 25 modules or, in the corner sectors, 13 modules, for a total of 177. The modules are divided into two parts, separated by insulating foam: the focal plane module (FPM) holding the SiPM tiles and the front-end electronics (FEE) housed in an aluminum cage. Each FPM contains 64 image pixels, for a total of 11,328 pixels. For purposes of triggering, the pixels are grouped into fours, so that each FPM has 16 trigger pixels. To reduce astigmatism, the focal plane surface is curved. Figure 3.15 illustrates the hierarchical design of the SCT camera. A full description of the design and performance of the pSCT camera is given by Taylor (2021).

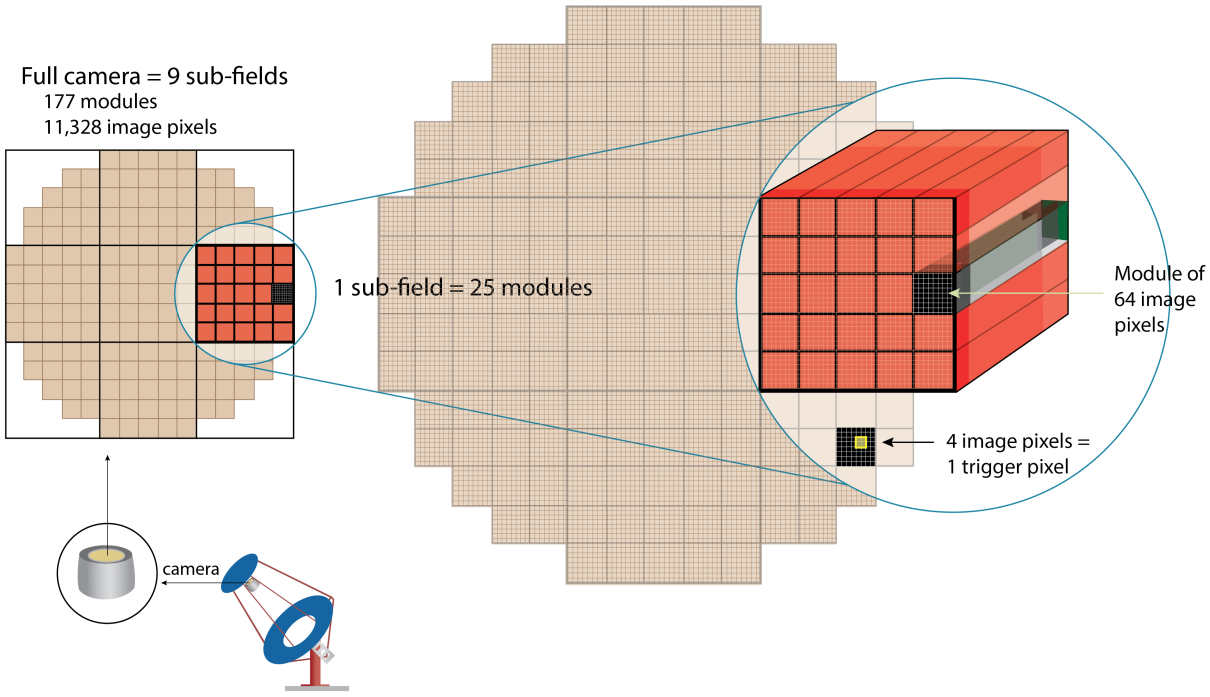


Figure 3.15: Diagram of the SCT camera showing its hierarchical design, from Taylor (2021).

In addition to the modules, the camera system also includes backend electronics and auxiliary subsystems. The modules in each sector are controlled for purposes of triggering and data readout through a backplane, which is a printed circuit board. The backplane contains two field-programmable gate arrays (FPGAs), consisting of a housekeeping FPGA that provides slow control and monitoring functions and a trigger FPGA that sends signals to the modules to trigger based on coincidence logic or an external command. Two custom data acquisition (DACQ) boards for each sector handle data acquisition by routing the data from the front-end electronics in the modules to a data server.

In the current pSCT camera, only the central sector is populated with modules, for a total of 25 modules and 1600 image pixels[2]. The backend electronics therefore consist of one backplane and two DACQ boards. The FEE of each module in the pSCT camera contains four custom TARGET 7 (seventh generation TeV Array Readout with GSa/s sampling and Event Trigger) application-specific integrated circuits (ASICs) for digitization and triggering (Funk et al., 2017). Figure 3.16 shows a block diagram of the pSCT camera subsystems.

[2]As of this writing, the central module is not installed in order to enable procedures for optical alignment, leaving 24 modules.

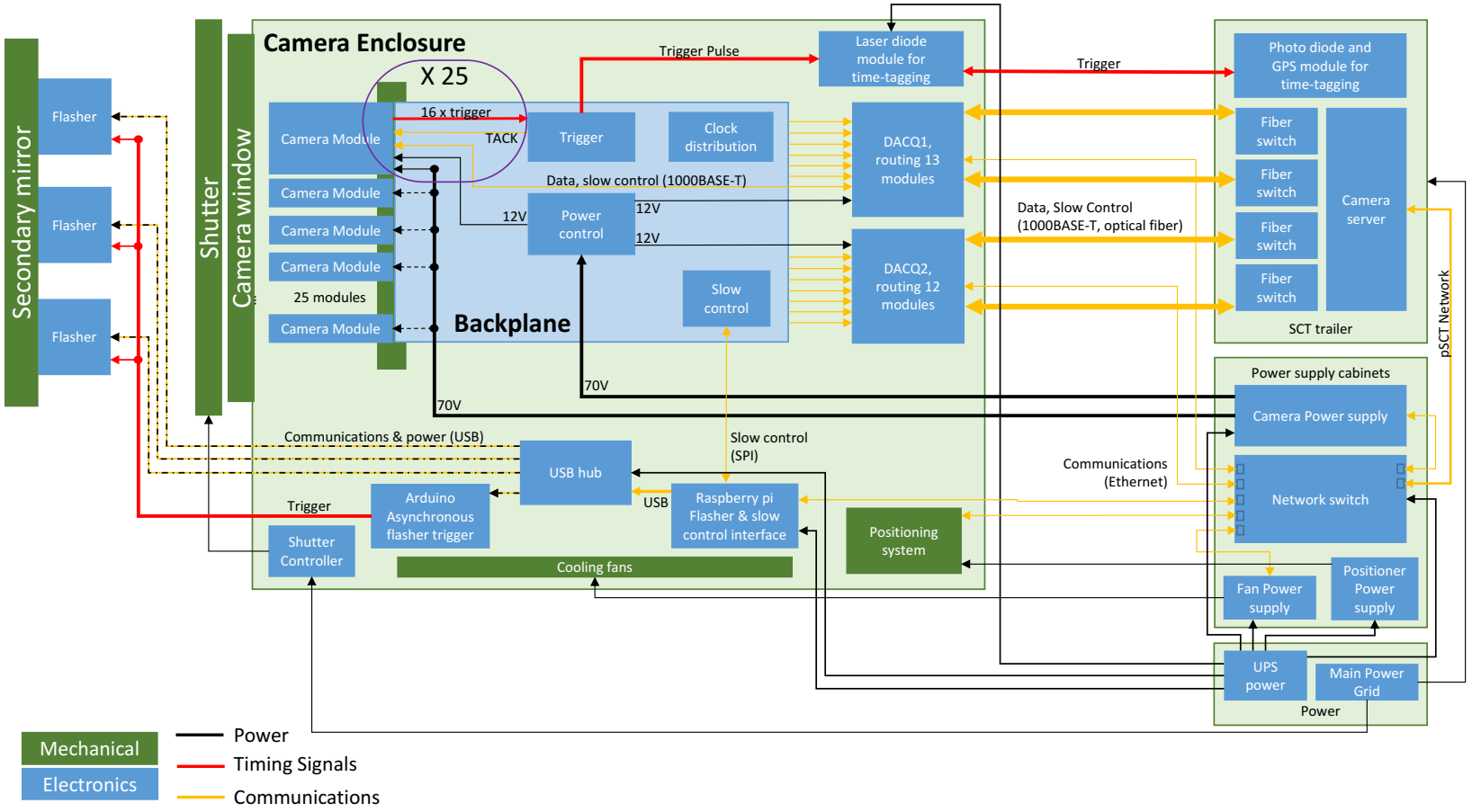


Figure 3.16: Block diagram of the pSCT camera subsystems, from Taylor (2021). Mechanical subsystems (green boxes) and electronics (blue boxes) are shown along with their connections via power (black lines), timing signals (red lines), and communications (yellow lines).

The camera system includes a number of auxiliary subsystems:

- **Power Supplies:** The power supplies for the camera components are mounted on a rack in a cabinet on the telescope, behind the camera. An Acopian power supply[3] provides power to the camera fans, while a Wiener PL506 power supply[4] powers the other components, including each module's FPM and FEE. The PL506 unit provides a main supply current and a high voltage (HV) current in two separate channels.

- **Chiller:** The module electronics and camera power supplies are cooled via heat exchange with a chilled coolant line supplied by a Dimplex SVO-5001M 60000 BTU chiller unit. From the chiller unit, the chilled coolant is sent through a manifold containing a pressure regulator which reduces the pressure to approximately 3 psi, compensating for differing pressures at different elevations of the telescope. The coolant temperature and pressure are monitored by sensors in the manifold, which are connected to a ControlByWeb X-320 web-based instrumentation module. The coolant then passes through heat exchangers connected to the FPMs and to the camera power supplies. The FPM temperatures are stabilized and monitored by Peltier micro-controllers.

- **Fans:** In addition to the cooling provided by the chiller, the camera is also cooled by eight Pabst EBM 6314 fans.

- **Shutter:** When not in use, the camera is protected from weather and daylight by a motorized commercial rollup shutter. The shutter is connected to a motor controller which can be controlled via a ControlByWeb X-301 WebRelay-Dual remote relay controller, as well as manually by using a remote control.

- **Flashers:** In order to calibrate the throughput of the camera and relative gains of the pixels, three LED flashers are installed on an optical table located at the center of the

[3] `https://www.acopian.com/single-s-1u-m.html`
[4] `http://www.wiener-d.com/sc/power-supplies/pl500/pl506.html`

secondary mirror. Each flasher has ten LEDs. The flashers are programmable, so that the pulse rate and LED activation pattern can be changed dynamically.

An upgrade to fully populate the camera with 9 backplanes and all 177 modules is planned for December 2022 (Meures, 2019). As part of this effort, many camera components will be redesigned and upgraded, including the backplane; the modules, encompassing both the SiPMs and the TARGET ASICs; and multiple auxiliary subsystems including the camera fans, shutter, and flashers.

Chapter 4: Variability and Spectral Characteristics of Three Flaring Gamma-ray Quasars Observed by VERITAS and *Fermi*-LAT

Flat spectrum radio quasars (FSRQs) are the most luminous blazars at GeV energies, but only rarely emit detectable fluxes of TeV gamma rays, typically during bright GeV flares. In this chapter, we explore the gamma-ray variability and spectral characteristics of three FSRQs that have been observed at GeV and TeV energies by *Fermi*-LAT and VERITAS, making use of almost 100 hours of VERITAS observations spread over 10 years: 3C 279, PKS 1222+216, and Ton 599, including VERITAS detections of PKS 1222+216 and Ton 599 during *Fermi*-LAT flares. Using these datasets, we study the variability of these FSRQs over long and short timescales and their behavior at GeV and TeV energies during fast flares. The contents of this chapter are based on a journal article led by the author, conducted jointly within the VERITAS and *Fermi*-LAT collaborations, that has been prepared for submission to the Astrophysical Journal (Brill et al., 2021). All figures and tables in this chapter, other than those in Section 4.2.1, have been taken from that work.

In Section 4.1 we describe the observations of the three FSRQs that we study in this chapter, and in Section 4.2 we report on the data analysis. In Section 4.3, we explain the GeV flux distributions of the sources in terms of a model derived from a stochastic differential equation and estimate the timescales of magnetic flux accumulation and stochastic instabilities in their accretion disks. In Section 4.4, we identify distinct flares using a procedure based on Bayesian blocks, and in Sections 4.5 and 4.6, we analyze their daily and sub-daily variability and gamma-ray energy spectra, respectively. In Section 4.7, using observations from VERI-

TAS as well as *Fermi*, *Swift*, and the Steward Observatory, we model the broadband spectral energy distributions of PKS 1222+216 and Ton 599 during VHE-detected flares in 2014 and 2017, respectively. This modeling places strong constraints on the jet Doppler factors, which we use in Section 4.8 to further constrain the gamma-ray emission region locations during these events. In Section 4.9, we place theoretical constraints on the potential production of PeV-scale neutrinos during these VHE flares. Finally, in Section 4.10, we describe the contributions of others to the work presented in this chapter.

Throughout this chapter, a flat ΛCDM cosmology was used, with $H_0 = 69$ km s^{-1} Mpc^{-1}, $\Omega_M = 0.286$, and $\Omega_\Lambda = 0.714$.

4.1 Observations of Three FSRQs

In this work, we investigate strong gamma-ray flares from three FSRQs at intermediate redshifts. 3C 279, at $z = 0.536$ (Lynds, Stockton, and Livingston, 1965), is one of the most well-studied blazars. It is among the brightest and most variable extragalactic objects in the gamma-ray sky, giving rise to one of the first large amplitude gamma-ray flares measured by EGRET in 1996 (Wehrle et al., 1998). In recent times, it underwent multiple bright gamma-ray flares in 2014, 2015, and 2018. Notably, during a flare beginning on June 16, 2015, it was detected by H.E.S.S., and *Fermi*-LAT observed minute-scale variability (Romoli et al., 2017; Ackermann et al., 2016). H.E.S.S. again detected 3C 279 during flaring states in January and June 2018 (Emery et al., 2019).

PKS 1222+216, at $z = 0.432$ (Osterbrock and Pogge, 1987) and also known as 4C +21.35, has exhibited periods of extreme variability in the VHE gamma-ray band, with VHE detections occurring during gamma-ray flares in June 2010 (Aleksić et al., 2011) and February and March 2014 (Holder, 2014).

Finally, Ton 599, at $z = 0.725$ (Schneider et al., 2010; see also Burbidge, 1968) and

Fermi-LAT

Source	z	Date Range [UT]	Energy Range [GeV]	Time Binning [day]	No. Bins	Flare Threshold (No. Flares) [ph cm^{-2} s^{-1}]
3C 279	0.5362	2008-08-04 – 2018-12-07	0.1-500	1	3471	4×10^{-6} (10)
PKS 1222+216	0.432	2008-08-04 – 2018-12-07	0.1-500	3	1158	5×10^{-7} (11)
Ton 599	0.725	2008-08-04 – 2018-12-12	0.1-500	7	512	5×10^{-7} (5)

VERITAS

Source	z	Energy Threshold [GeV]	Exposure [hr]	No. Obs.
3C 279	0.5362	200	54.4	139
PKS 1222+216	0.432	110	34.7	95
Ton 599	0.725	140	8.8	20

Table 4.1: Overview of the *Fermi*-LAT and VERITAS datasets presented in this work. For VERITAS, the energy threshold varies for different observations. A typical value is quoted for 3C 279 and the values during the VHE-detected flares are quoted for PKS 1222+216 and Ton 599.

also known as 4C +29.45 and B1156+295, entered a months-long HE high state in October 2017 (Cheung, Gasparrini, and Buson, 2017), leading to VHE detections on the nights of December 15 and 16 2017 (Mirzoyan, 2017; Mukherjee, 2017).

These three sources were continuously monitored by *Fermi*-LAT during the ten-year period from 2008 to 2018, and observed during periods of high gamma-ray activity by VERITAS. The VERITAS observations of 3C 279, PKS 1222+216 and Ton 599 that were simultaneous with the HE flares considered here were taken in response to the elevated fluxes reported by *Fermi*-LAT. For 3C 279 and PKS 1222+216, additional monitoring observations provided VERITAS data corresponding to low states observed by *Fermi*-LAT. Table 4.1 gives an overview of the gamma-ray data analyzed in this work. The *Fermi*-LAT light curves of the three sources and the periods of the VERITAS observations are shown in Figure 4.1. The LAT time binnings, reported in Table 4.1, were chosen for each source depending on its typical strength to avoid having an excessive number of bins with no detection.

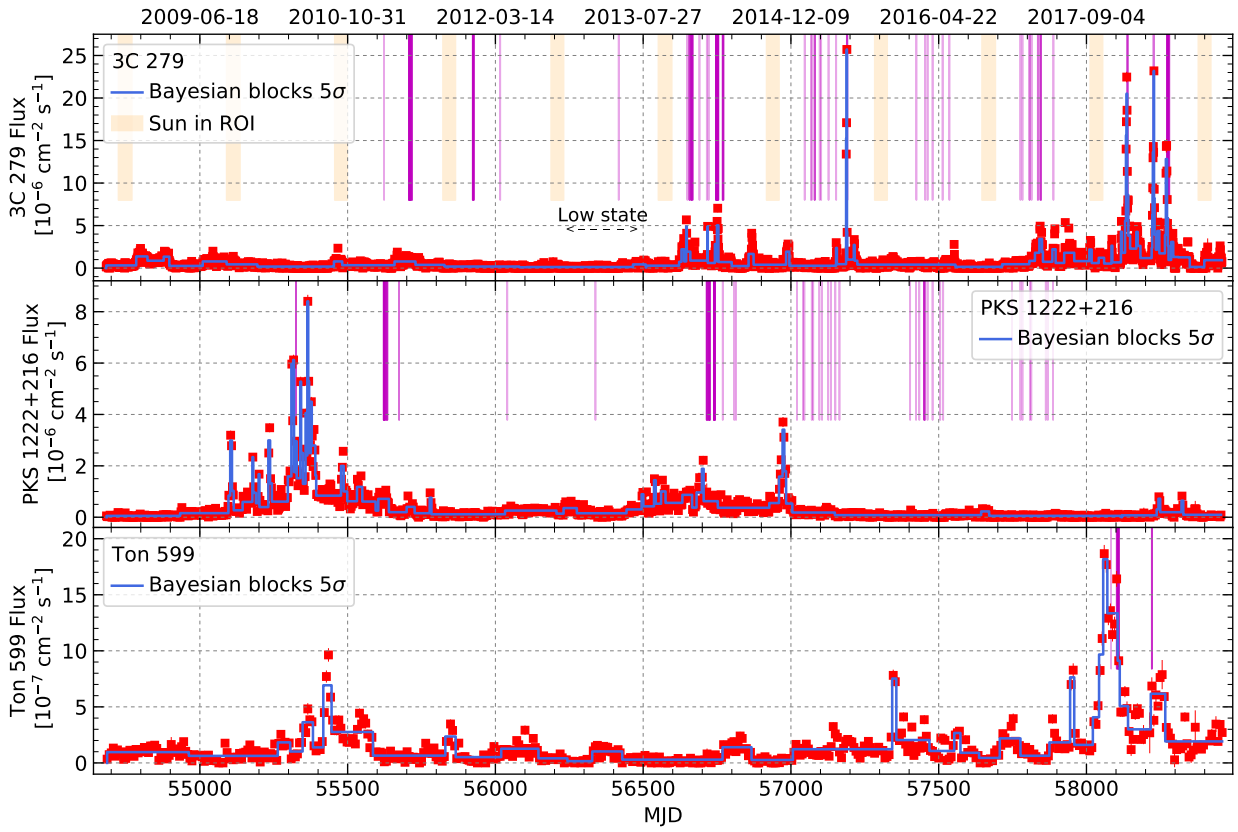


Figure 4.1: *Fermi*-LAT light curves of 3C 279 (top), PKS 1222+216 (middle), and Ton 599 (bottom). The flux points (red squares) are shown for 1, 3, and 7 day time bins for the three sources, respectively. $5\,\sigma$ Bayesian blocks are shown with blue lines. The time intervals in which VERITAS observed the sources are marked in magenta. For 3C 279, time intervals in which the Sun is less than 20° from the source are shown in orange and a *Fermi*-LAT low state from MJD 56230 – 56465 (see Section 4.6) is marked with a dashed line.

4.2 Data Analysis

4.2.1 VERITAS

The total exposure taken by VERITAS on each of the sources is reported in Table 4.1. The VERITAS data were inspected for data quality and the time intervals affected by weather or instrumental issues were excluded from the analysis. The data were analyzed using Event-Display (Maier and Holder, 2017) and independently cross-checked using VEGAS (Cogan, 2008). Boosted decision trees with soft selection cuts (appropriate for sources with a spectral index softer than $\Gamma \approx 3.5$) were used for separating gamma rays from background cosmic rays (Krause, Pueschel, and Maier, 2017). The reflected regions background model was used. The upper limits were calculated using the method of Rolke, López, and Conrad (2005). Preliminary analysis results of the VERITAS observations of 3C 279 and PKS 1222+216 in 2013 and 2014 were reported by Errando, 2014. These are superseded by the more updated analysis reported here.

Because VERITAS did not detect 3C 279, we defined discrete time periods to analyze based on the flares observed with *Fermi*-LAT. In Section 4.4, we describe the algorithm we used to define ten flare intervals and a "quiescent" non-flaring period. The time periods of these intervals are given in Table 4.7. Of the ten flares, five had corresponding VERITAS observations. The results of the VERITAS analyses for each flare of 3C 279 and the quiescent period are reported in Table 4.2. A representative sky map and significance distribution from one of the flares, Flare 3, is shown in Figure 4.2. The sky map is a correlated (smoothed) significance map, such that the significance at each point is calculated also using events in neighboring spatial bins, with the smoothing radius set equal to the size of the source exclusion region. The significance distribution (histogram of sky map bins) is well represented by a normal distribution with a mean of zero and a standard deviation of one, consistent with the background distribution expected for an empty field.

Event	t_{obs} [hr]	t_{live} [hr]	N_{on}	N_{off}	α	σ	Avg. El. [deg]	$E_{\text{thresh,obs}}$ [GeV]
Flare 3	8.07	5.94	225	2014	0.1	1.55	50.1	200
Flare 4	1.51	0.91	24	341	0.1	-1.75	40.4	500
Flare 6	1.80	1.19	27	223	0.1	0.92	51.2	250
Flare 8	1.00	0.74	18	135	0.1	1.11	51.1	200
Flare 10	4.00	3.51	109	769	0.1	3.26	49.3	250
Quiescent	42.23	35.23	1160	11620	0.1	-0.06	50.3	200

Table 4.2: VERITAS analysis results for 3C 279. t_{obs} is the total exposure taken on the source, while t_{live} is the total exposure minus deadtime (typically ∼10%) and time cuts for data quality, generally necessitated by adverse weather. $E_{\text{thresh,obs}}$ is the approximate energy threshold of the observations, which tends to increase at lower average elevations.

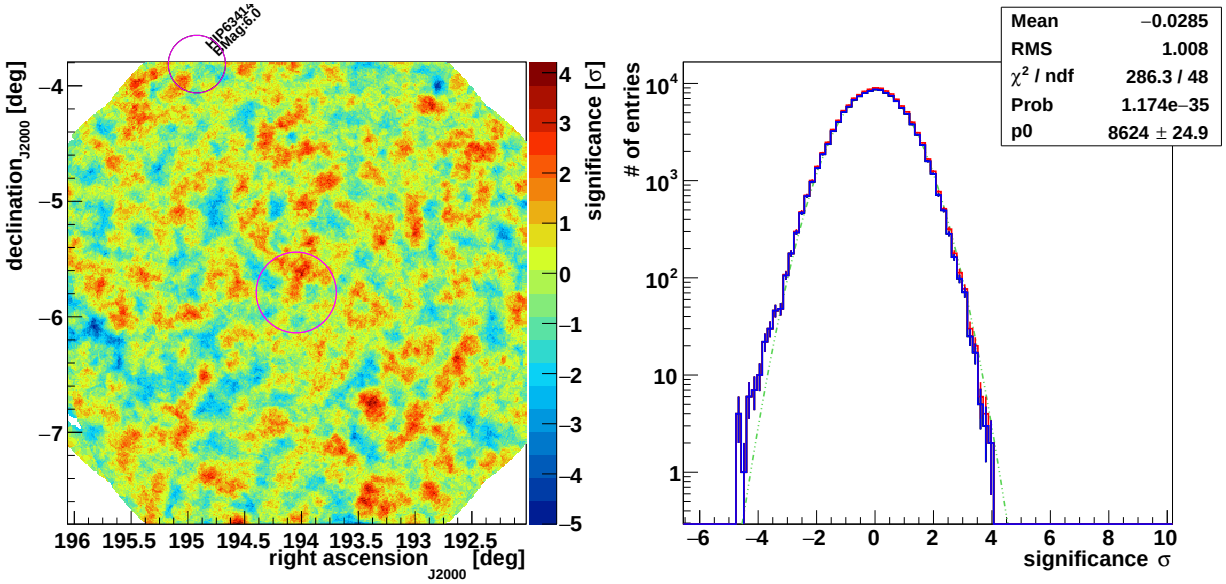


Figure 4.2: *Left*: sky map for a representative flare of 3C 279 (Flare 3). *Right*: significance distribution for the same event. Distributions are shown for all bins (red), the source and other exclusion regions excluded (black), and just the source region excluded (blue), along with a normal distribution with mean 0 and standard deviation 1 (green).

Source	$t_{\rm obs}$ [hr]	$t_{\rm live}$ [hr]	$N_{\rm on}$	$N_{\rm off}$	α	σ	Avg. El. [deg]	$E_{\rm thresh,obs}$ [GeV]
PKS 1222+216	7.90	5.87	370	2602	0.1	6.06	75.1	110
Ton 599	2.50	2.15	203	780	0.1	11.0	70.3	140

Table 4.3: VERITAS analysis results for PKS 1222+216 and Ton 599 during their VHE flares. $t_{\rm obs}$ is the total exposure taken on the source, while $t_{\rm live}$ is the total exposure minus deadtime (typically $\sim$10%) and time cuts for data quality, generally necessitated by adverse weather. $E_{\rm thresh,obs}$ is the approximate energy threshold of the observations.

$E_{\rm bin}$ [GeV]	$E_{\rm min}$ [GeV]	$E_{\rm max}$ [GeV]	$E^2\,dN/dE$ [10^{-12} erg cm^{-2} s^{-1}]
141	112	178	6.3 ± 3.2
224	178	282	1.8 ± 1.0
531	282	1000	< 0.95

Table 4.4: VERITAS spectrum of PKS 1222+216 during its VHE flare. The last point is an upper limit at the 95% confidence level.

For PKS 1222+216 and Ton 599, we report the analysis results during their detections by VERITAS, which occurred between February 26 and March 10, 2014, and December 15 and December 16, 2017, respectively. The analysis results for the two sources are reported in Table 4.3. The sky map, significance distribution, and spectral points are given for PKS 1222+216 in Figure 4.3 and Table 4.4, and for Ton 599 in Figure 4.4 and Table 4.5.

$E_{\rm bin}$ [GeV]	$E_{\rm min}$ [GeV]	$E_{\rm max}$ [GeV]	$E^2\,dN/dE$ [10^{-12} erg cm^{-2} s^{-1}]
168	141	200	28.1 ± 9.8
237	200	282	14.1 ± 3.1
335	282	398	5.3 ± 1.9
631	398	1000	1.3 ± 0.7

Table 4.5: VERITAS spectrum of Ton 599 during its VHE flare.

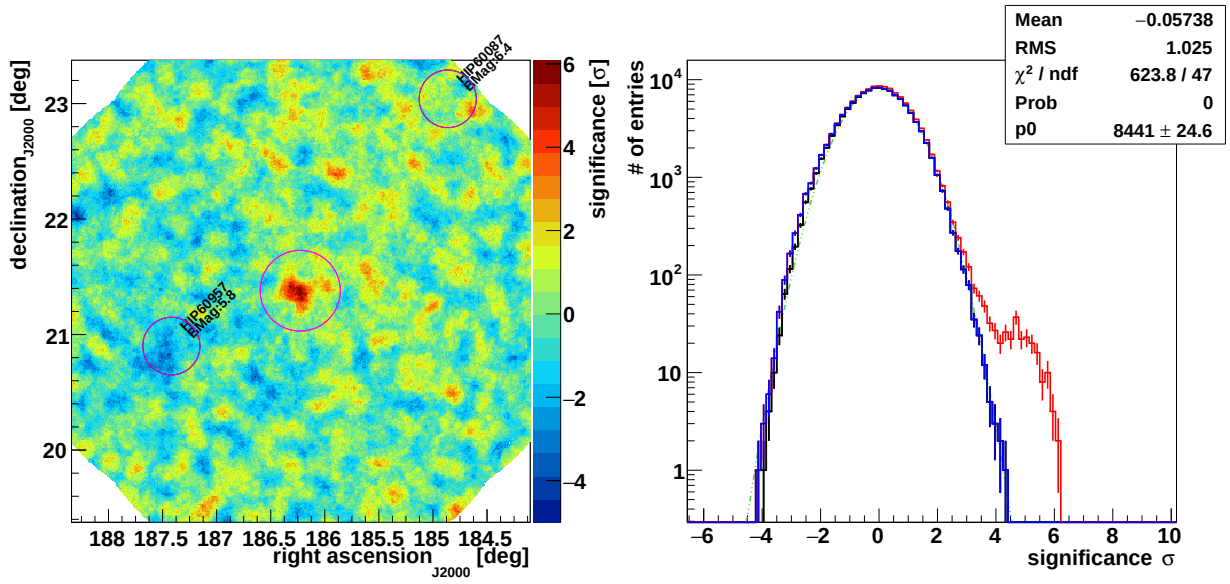


Figure 4.3: *Left*: sky map for PKS 1222+216 during its VHE flare. *Right*: significance distribution for the same event. Distributions are shown for all bins (red), the source and other exclusion regions excluded (black), and just the source region excluded (blue), along with a normal distribution with mean 0 and standard deviation 1 (green).

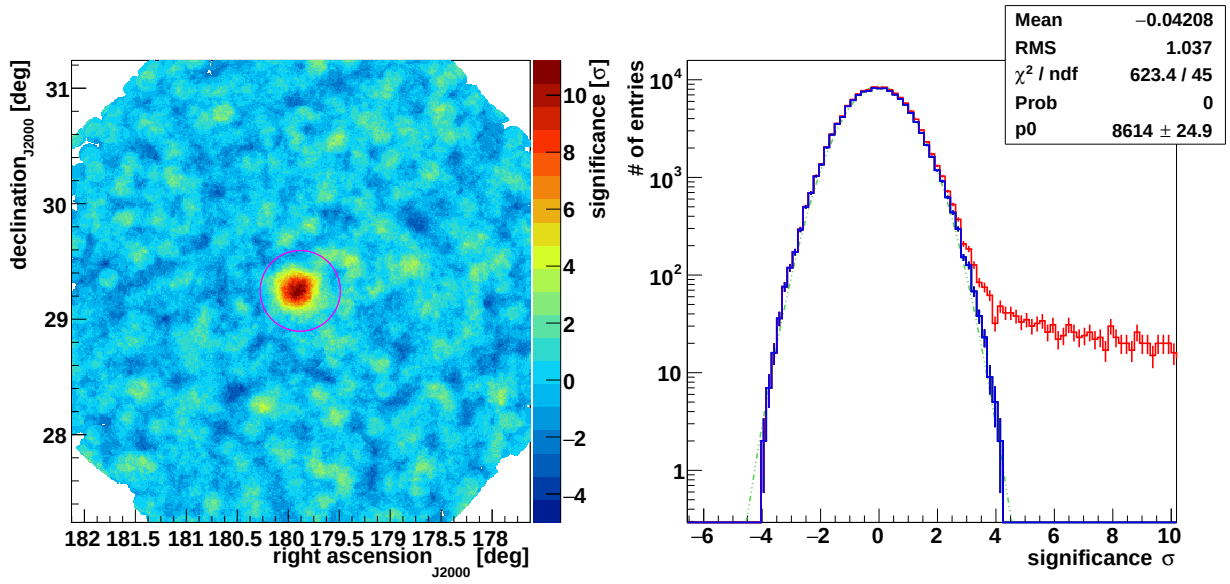


Figure 4.4: *Left*: sky map for Ton 599 during its VHE flare. *Right*: significance distribution for the same event. Distributions are shown for all bins (red), source and other exclusion regions excluded (black), and just the source region excluded (blue), along with a normal distribution with mean 0 and standard deviation 1 (green).

4.2.2 *Fermi*-LAT

We performed an unbinned likelihood analysis of the data using the LAT `Fermitools` 1.0.3 and instrument response functions `P8R3_SOURCE_V2`. We analyzed the data in the 10.3 year period starting on August 4, 2008 (MJD 54682.7), the start of the *Fermi*-LAT all-sky survey, as reported in Table 4.1. The energy range from 0.1 GeV to 500 GeV was analyzed, and photons with zenith angle $> 90°$ were excluded to reduce contributions from the Earth's limb. For each source, the region of interest (ROI) considered was the circle of radius 10° surrounding the catalog source position. The background model consisted of, along with galactic (`gll_iem_v06.fits`) and isotropic (`iso_P8R3_SOURCE_V2.txt`) diffuse emission models, all sources in the FL8Y catalog[1] within a 20° circle surrounding the source. This is to ensure that the model would include gamma-ray emission from sources outside the ROI that could extend into the ROI due to the size of the point spread function of the LAT, especially at low energies.

When performing the likelihood fit, we iteratively fixed the parameters of the least significant sources until convergence was reached. Sources with TS less than zero were removed from the model. When fitting individual light curve and SED points, the spectral parameters were kept fixed, either to their catalog values for global analyses or to values determined by analyzing the entire flare duration for flare analyses, with the diffuse background model normalization parameters left free. We checked that the background model we used is consistent with the 4FGL-DR2 catalog (Abdollahi et al., 2020; Ballet et al., 2020), finding no new bright, variable sources in the ROI of each of the three FSRQs that could significantly impact the analysis of our sources. We excluded time ranges corresponding to solar flares and gamma-ray bursts in the ROI from the analysis.

Since 3C 279 lies close to the ecliptic, the Sun and Moon contribute diffuse foreground

[1] `https://fermi.gsfc.nasa.gov/ssc/data/access/lat/fl8y/`

emission in the ROI of this source during certain periods (Abdo et al., 2011, 2012). We checked using a likelihood analysis containing extended templates for the Sun and Moon emission that the flux of 3C 279 does not change significantly for time bins in which the Sun or Moon is more than 5° from the source. Since both the Sun and Moon were more than 20° from 3C 279 during all of the flare states identified in Section 4.4, no contamination is expected during any of these periods.

4.2.3 *Swift*-XRT

The X-Ray Telescope (XRT) on the Neil Gehrels *Swift* observatory is a grazing-incidence focusing X-ray telescope, and is sensitive to photons with the energies between 0.2 and 10 keV (Gehrels et al., 2004; Burrows et al., 2005). *Swift*-XRT observed PKS 1222+216 and Ton 599 during the VHE flares of those sources.

The *Swift*-XRT data were extracted from the *Swift* data archive and analyzed using `HEASoft v6.24`. The fluxes and flux errors were deabsorbed using the fixed total column density of Galactic hydrogen $N_H = 2.29 \times 10^{20}$ cm^{-2} for PKS 1222+216 and 1.89×10^{20} cm^{-2} for Ton 599 (Kalberla et al., 2005; Willingale et al., 2013) and the photoelectric cross section $\sigma(E)$ to account for the effects of neutral hydrogen absorption. The deabsorbed X-ray spectrum was fitted with a broken power law model for PKS 1222+216 and a power law model for Ton 599.

4.2.4 *Swift*-UVOT

The ultraviolet/optical telescope (UVOT) on the Neil Gehrels *Swift* observatory is a photon counting telescope sensitive to photons with energies ranging from about 1.9–7.3 eV or 170 - 550 nm (Roming et al., 2005). *Swift*-UVOT observed PKS 1222+216 and Ton 599 approximately concurrently with *Swift*-XRT.

The UVOT data were extracted from the *Swift* data archive and analyzed using `HEASOFT`

`v6.28`. The counts from the sources and the background were extracted from regions of a radius of 5.0″ centered on the position of the sources and nearby positions without any bright sources, respectively. The magnitude values of the sources were computed using `uvotsource`, and converted to fluxes using the zero-points given by Poole et al. (2008). Extinction corrections were applied following Roming et al. (2009), using the reddening values $E(B-V) = 0.0199$ and 0.0171 (Schlafly and Finkbeiner, 2011) for PKS 1222+216 and Ton 599, respectively.

4.2.5 Steward Observatory

During the first decade of the *Fermi* mission, the Steward Observatory of the University of Arizona obtained optical polarimetry, photometry, and spectra of the LAT-monitored blazars and *Fermi* targets of opportunity (ToOs) using the SPOL CCD Imaging/Spectropolarimeter (Smith et al., 2009). We used the reduced photometric Johnson V and R band data, which are available online[2]. Six observations were taken of Ton 599 and two of PKS 1222+216 during their respective VHE flares. There was no significant variability during either event.

4.3 *Fermi*-LAT Flux Distributions

The distribution of the LAT fluxes observed from each of these FSRQs may provide a clue to the origin of the gamma-ray emission. The observed flux distributions of the three sources (scaled to form probability density histograms) are shown in Figure 4.5. Time bins that have a test statistic (TS) less than 9 or that occur when the Sun is less than 20° from the source were excluded.

To account for uncertainties from both the flux binning and the finite observation length, the flux histogram bin errors were calculated using a bootstrapping approach. 2,500 bootstrap samples were used, each consisting of the same number of flux points as the actual

[2] `http://james.as.arizona.edu/~psmith/Fermi/`

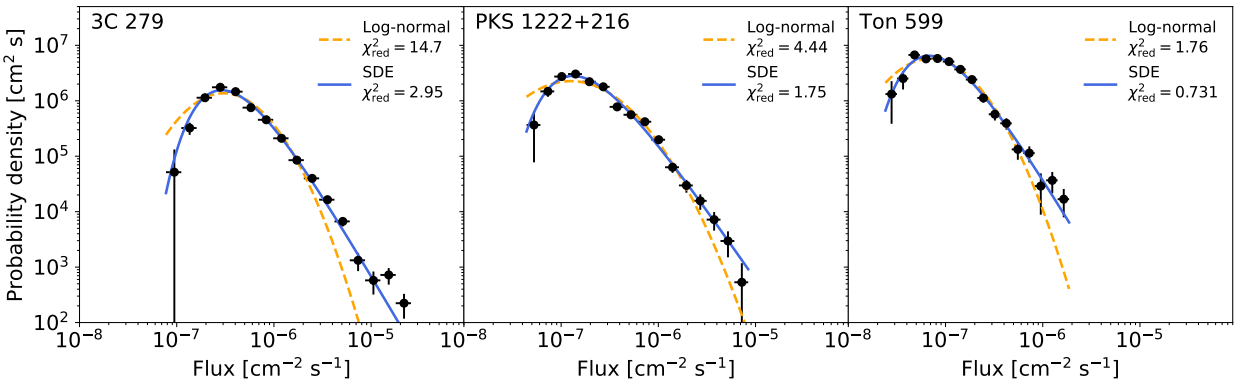


Figure 4.5: Flux distributions of the three FSRQs, scaled as probability densities. The distributions are fit with a log-normal PDF (dashed orange) and the stationary-state PDF corresponding to the SDE of Tavecchio, Bonnoli, and Galanti (2020) (solid blue). In all three cases, the SDE provides a better fit.

light curve. Each bootstrap sample was obtained by sampling from the set of actual flux points with replacement, so that a given flux point might be sampled multiple times or not at all. To include the uncertainties of the individual flux points, an error term was added to each sampled point in each bootstrap sample, determined by sampling from a Gaussian distribution with standard deviation equal to the measurement uncertainty of the respective sampled point. The bin errors were then defined as the standard deviations of the bin fluxes over all of the bootstrap samples binned using the same bins as the original dataset.

As discussed in Section 2.3.2, blazar flux distributions are often described using a log-normal distribution. An alternative model has been proposed by Tavecchio, Bonnoli, and Galanti (2020), based on a stochastic differential equation (SDE) with two terms modeling a deterministic tendency to return to equilibrium and stochastic fluctuations with amplitude proportional to the absolute flux level. The form of the SDE is motivated by an astrophysical scenario of stochastic disturbances perturbing a magnetically arrested accretion disk. In this model, the flux distribution is asymmetrical about a peak, falling off as a power law at high fluxes and exponentially at low fluxes, with the relative importance of the deterministic and stochastic components dictating the shape of the distribution. Figure 4.5 shows a

comparison between the best-fit probability density functions (PDFs) corresponding to a log-normal distribution and the stationary state of the SDE proposed by Tavecchio, Bonnoli, and Galanti, 2020.

The stationary-state PDF corresponding to the SDE (Tavecchio, Bonnoli, and Galanti, 2020, Appendix A) is

$$p(X) = \frac{(\lambda\mu)^{1+\lambda}}{\Gamma(1+\lambda)} \frac{e^{-\lambda\mu/X}}{X^{\lambda+2}}, \tag{4.1}$$

where X is a dimensionless random variable proportional to the flux, μ is a parameter representing the equilibrium value of X, λ is a parameter representing the relative weight of the deterministic and stochastic terms, and Γ is the gamma function. Here, X was related to the flux by a proportionality constant of 1×10^{-7} ph cm^{-2} s^{-1}. The distribution peaks at $X_{\max} = \mu\lambda/(\lambda+2)$. The stationary-state PDF is valid on timescales much longer than the timescale for the system to return to equilibrium, which is certainly the case for the ten-year periods considered here.

The PDFs were fit to the histogram bins with a nonlinear least-squares algorithm using SciPy (Virtanen et al., 2020). The best-fit parameters and reduced χ^2 values of the two models are reported in Table 4.6. In all three cases, the SDE PDF provides a better fit than the log-normal PDF. Both models have two free parameters. We verified that the preference for the SDE model is preserved if the histogram bins at the lowest fluxes, which might be affected by requiring light curve bins to have TS > 9, are excluded from the fit.

The SDE model PDF is parameterized by the shape parameter $\lambda \equiv 2\theta/\sigma^2$, where θ and σ are the coefficients of the deterministic and stochastic terms. These parameters can be interpreted by associating $1/\theta$ with the timescale of magnetic field accumulation in the accretion disk, while σ is related to the dynamics of the perturbative processes. A large value of λ therefore represents a high relative importance of the deterministic variability

Source	Log-normal			SDE			
	μ	σ	$\chi^2_{\rm red}$	μ	λ	$X_{\rm max}$	$\chi^2_{\rm red}$
3C 279	1.65±0.02	0.73±0.01	14.7	8.63±0.28	1.01±0.06	2.90±0.21	2.95
PKS 1222+216	1.07±0.03	0.92±0.03	4.44	6.27±0.60	0.54±0.08	1.33±0.23	1.75
Ton 599	0.21±0.04	0.75±0.03	1.76	2.11±0.19	0.94±0.15	0.68±0.13	0.73

Table 4.6: Best-fit parameters and goodness of fit ($\chi^2_{\rm red}$) for the log-normal and SDE PDF fits to the LAT flux distributions. The values of μ and σ for the log-normal distribution and μ for the SDE distribution have been normalized to 1×10^{-7} ph cm^{-2} s^{-2}. For the SDE PDF, the peak flux is determined by $X_{\rm max} = \mu\lambda/(\lambda + 2)$.

component compared to the stochastic one, while a small value indicates the opposite. To relate these timescales to the gravitational radii of the central supermassive black holes, $r_g = GM/c^2$, we adopt values of $\sim 5 \times 10^8$, 6×10^8, and $3.5 \times 10^8 M_\odot$ for the black hole masses of 3C 279, PKS 1222+216, and Ton 599, respectively (Hayashida et al., 2015; Farina et al., 2012; Liu, Jiang, and Gu, 2006).

One can estimate σ^2 from the light curve using the expression (Tavecchio, Bonnoli, and Galanti, 2020):

$$\sigma^2 \simeq \frac{1}{n} \sum_{i=0}^{n} \frac{(X_i - X_{i-1})^2}{X_{i-1}^2 (t_i - t_{i-1})}, \tag{4.2}$$

where X_i is the scaled flux at time step i. Using this expression, we obtain σ^2 equal to 0.354, 0.158, and 0.0616 day^{-1}, or 100, 185, and 820 r_g/c, for 3C 279, PKS 1222+216, and Ton 599, respectively. These values are consistent with the $\gtrsim$100 r_g/c variability timescale injected into the jet by magneto-rotational instability in the accretion disk estimated in theoretical work (Giannios and Uzdensky, 2019). Using the relation $1/\theta = 2/\lambda\sigma^2$, we can then constrain the physics of accretion flow in 3C 279, PKS 1222+216, and Ton 599 by estimating their magnetic flux accumulation timescales to be 200, 690, and 1750 r_g/c, respectively, within the magnetically-arrested disk scenario.

4.4 Flare Selection

Flare states were identified in the *Fermi*-LAT data using the following procedure:

1. Segment the data using Bayesian blocks. We set the false positive rate p_0 to the value equivalent to $5\,\sigma$ using Equation 13 of Scargle et al. (2013).

2. Choose a flux threshold above which the blocks are designated as flaring.

3. Designate each contiguous set of flare blocks as an individual flare state and all non-flare blocks as the quiescent state.

This empirical procedure reflects a picture of individual flares superimposed on a constant quiescent background, but identifies them purely as states of elevated flux, making no explicit assumptions about the flares' shape or spectra. Due to its basis on Bayesian blocks, it guarantees that states identified as flares have flux significantly greater than the states surrounding them. The flux threshold to identify flares must be tuned on a source-by-source basis. Choosing the flux threshold to identify flares involves a tradeoff between ensuring that dimmer flares are selected and avoiding misidentifying fluctuations in the quiescent background as flares. In addition, because the sources differ in average flux, the threshold must necessarily vary on an absolute level from source to source. Performing the flare selection procedure with the flare selection thresholds listed in Table 4.1 results in 10 flares selected for 3C 279, 11 for PKS 1222+216, and 5 for Ton 599, listed in Table 4.7.

Because the flux distributions are best fit by the single-component SDE model PDF, it is not natural to calculate a duty cycle of flares based on a division into baseline and flaring components (e.g. Resconi et al., 2009). The amount of time spent in the highest-flux states can be estimated directly from the flux distribution by defining the "typical" flux as the peak of the PDF, given in Table 4.6. 3C 279, PKS 1222+216, and Ton 599 have flux greater than 5 (10) times the typical flux 12% (4%), 19% (8%), and 13% (4%) of the time, respectively.

#	Date Range (MJD)	Approx. Date	Blocks	VHE Exp.
		3C 279		
1	56645.66 - 56647.66	Dec 2013	1	-
2	56717.66 - 56718.66	Mar 2014	1	-
3	56749.66 - 56754.66	Apr 2014	1	6.79 hr
4	57186.66 - 57190.66	Jun 2015	3	1.00 hr
5	58116.66 - 58119.66	Dec 2017	1	-
6	58130.66 - 58141.66	Jan 2018	4	1.38 hr
7	58168.66 - 58173.66	Feb 2018	1	-
8	58222.66 - 58230.66	Apr 2018	5	0.83 hr
9	58239.66 - 58247.66	May 2018	1	-
10	58268.66 - 58275.66	Jun 2018	2	3.95 hr
		PKS 1222+216		
1	55096.66 - 55114.66	Sep-Oct 2009	3	-
2	55144.66 - 55201.66	Nov-Dec 2009	5	-
3	55231.66 - 55594.66	2010	27	-
4	55603.66 - 55639.66	Feb-Mar 2011	4	5.38 hr
5	55777.66 - 55783.66	Aug 2011	1	-
6	56494.66 - 56500.66	Jul 2013	1	-
7	56536.66 - 56665.66	Sep 2013	5	-
8	56680.66 - 56752.66	Jan-Apr 2014	3	15.53 hr
9	56926.66 - 57004.66	Sep-Dec 2014	5	-
10	58243.66 - 58249.66	May 2018	1	-
11	58321.66 - 58327.66	Jul 2018	1	-
		Ton 599		
1	55417.66 - 55445.66	Aug-Sep 2010	1	-
2	57342.66 - 57356.66	Nov 2015	1	-
3	57944.66 - 57958.66	Jul 2017	1	-
4	58042.66 - 58140.66	Oct-Jan 2017/18	5	8.30 hr
5	58217.66 - 58266.66	Apr-May 2018	1	2.00 hr

Table 4.7: *Fermi*-LAT flares selected using the algorithm given in Section 4.4. For each enumerated flare, the date range in MJD, approximate calendar date, number of Bayesian blocks, and amount of VHE gamma-ray exposure taken by VERITAS (if any) are provided. All of the times in the date ranges given in Table 4.1 but not listed here are considered to be quiescent.

Our flare selection flux thresholds for 3C 279 and Ton 599 are comparable at 13.8 and 11.8 times their typical fluxes, consistent with their similar values of the PDF shape parameter $\lambda \approx 1$. For PKS 1222+216, our threshold is 3.8 times the typical flux. This source has a lower value of $\lambda \approx 0.5$, with a correspondingly harder power law of the flux distribution at high fluxes. This is perhaps reflected in the long epochs of high flux seen in this source's light curve, such as its Flare 3 in 2010 which is approximately a year in duration (Table 4.7). A relatively low threshold was therefore needed to also capture the smaller flares of the approximately weekly timescales that typically trigger VERITAS observations, consistent with the other two sources.

4.5 Daily and Sub-daily Variability

In order to deduce the smallest variability time around the rising and decaying periods of each flare selected according to the algorithm described in Section 4.4, we extracted sub-daily light curves of the three sources down to the smallest significant time-bin sizes. The bins ranged from 12 hours down to 1.5 hours for the brightest source, 3C 279. For PKS 1222+216 and Ton 599, the minimum bin sizes were 12 and 6 hours, respectively.

To characterize the flares with multiple peaks we used a sum of exponential profiles (Valtaoja et al., 1999; Abdo et al., 2010a), F_i, where each one has the form

$$F_i(t) = \begin{cases} F_{0_i}\, e^{(t-t_{\mathrm{peak_i}})/t_{\mathrm{rise_i}}}, & t \leq t_{\mathrm{peak_i}} \\ F_{0_i}\, e^{-(t-t_{\mathrm{peak_i}})/t_{\mathrm{decay_i}}}, & t > t_{\mathrm{peak_i}}. \end{cases} \tag{4.3}$$

For flares with a single peak we used:

$$F(t) = \begin{cases} F_0\, e^{(t-t_{\mathrm{peak}})/t_{\mathrm{rise}}} + F_{\mathrm{const}}, & t \leq t_{\mathrm{peak}} \\ F_0\, e^{-(t-t_{\mathrm{peak}})/t_{\mathrm{decay}}} + F_{\mathrm{const}}, & t > t_{\mathrm{peak}}, \end{cases} \tag{4.4}$$

including a constant term to avoid biasing the fit towards large rise and decay timescales.

The fitting procedure started by considering a single peak characterized by Equation (4.4). In order to evaluate the possibility of adding a second peak, a fit to the sum of two exponential profiles, as given by Equation (4.3), was performed and compared against the one-peak scenario using the reduced χ^2 method. The two-peak model was taken when a clear preference was observed over the one-peak function. More peaks were added following a similar procedure until a reasonable reduced χ^2 value was reached, or when the best fit values obtained no longer provided relevant information for constraining the variability timescales of the sources under study. The peaks were not required to match the Bayesian blocks used for flare selection, which were defined using the coarsely binned light curves.

The flare profiles of the three sources are shown in Figure 4.6. Two selected flares of 3C 279 are shown, as are the two flares each of PKS 1222+216 and Ton 599 that were observed by VERITAS. Profiles of all ten flares of 3C 279 are provided in Appendix A. In order to illustrate when VERITAS observed the source relative to the LAT flare peaks, the VERITAS daily-binned light curves for each of the flares are also shown in Figure 4.6.The fit results for the three sources are reported in Tables 4.8 and 4.9.

For 3C 279, each flare lasted between one and eleven days and consisted of between one and four separately resolved components, modeled using exponential profiles. Twenty-four distinct components are resolved within the ten flares. The rise and decay times range from timescales of days to less than one hour. The smallest resolved variability timescale was 36 ± 13 minutes, which occurred around MJD 58227.945, during the rising period of Flare 8 (MJD 58222.655 – 58230.655), indicated in boldface in Table 4.8.

For PKS 1222+216 and Ton 599, the variability timescales were of the order of days. Notably, for both sources, the fastest variability did not occur during the detected VHE flares. The shortest variability timescale observed by LAT during the VHE flare of PKS 1222+216 was 10.4 ± 6.2 days, which was the decay timescale of the coincident flare component. The

Amplitude (F_0) [10^{-9} erg cm^{-2} s^{-1}]	$t_{\rm peak}$ [MJD]	$t_{\rm rise}$ [min]	$t_{\rm decay}$ [min]
Flare 4 (MJD 57186.655 – 57190.655): χ^2/d.o.f.= 77.31/19 = 4.07			
12.07 ± 0.67	57187.446 ± 0.031	378 ± 46	1784 ± 147
9.79 ± 2.29	57188.425 ± 0.028	216 ± 101	155 ± 64
21.72 ± 1.59	57189.069 ± 0.008	137 ± 18	512 ± 55
12.41 ± 1.30	57189.532 ± 0.010	220 ± 63	77 ± 25
Flare 8 (MJD 58222.655 – 58230.655): χ^2/d.o.f.= 177.25/106 = 1.67			
5.29 ± 1.29	58224.773 ± 0.105	1996 ± 716	5899 ± 4035
17.70 ± 2.01	58227.945 ± 0.004	**36 ± 13**	329 ± 131
16.42 ± 1.87	58228.323 ± 0.012	140 ± 54	115 ± 48
5.59 ± 1.69	58227.139 ± 0.133	3816 ± 1450	4077 ± 2080

Table 4.8: Results of the LAT flare profile fits for 3C 279. The smallest variability time found is indicated in boldface.

shortest variability timescale of Ton 599 observed by LAT during its VHE flare was 11.8±1.1 days, which also was the coincident flare component's decay timescale. In the case of Ton 599, the VERITAS detection occurred over a period of 2 days, after which the observed VHE flux became insignificant. No significant intra-flare variability was observed by *Fermi*-LAT or VERITAS during either event. Therefore, in the remainder of this work, we take the most constraining variability timescales during the VHE flares of PKS 1222+216 and Ton 599 to be 10 and 2 days, respectively.

The symmetry or asymmetry of flares can provide information on the timescales of the particle acceleration and cooling processes in the emission region (e.g. Abdo et al., 2010a). If the cooling time is longer than the light travel time through the emission region, the decay time will be longer than the rise time, producing an asymmetric flare. If the cooling time is shorter than the light travel time, the flare will appear more symmetrical.

Figure 4.7 shows the fitted rise and decay times for each of the exponential flare components of 3C 279. No clear trend in the flare asymmetry is observable, whether overall, among components within a single flare, or between the components belonging to different flares.

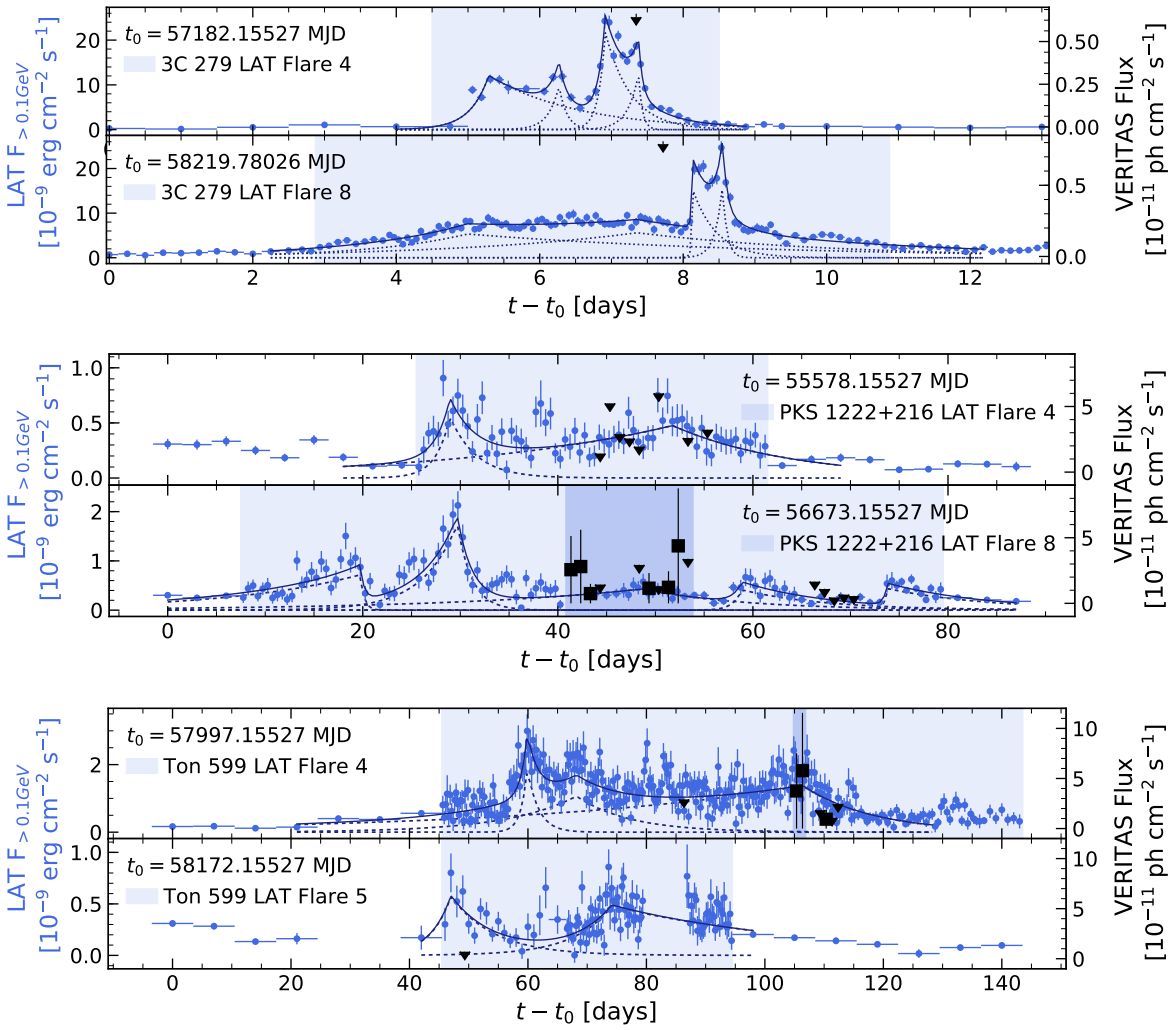


Figure 4.6: LAT daily and sub-daily light curves (blue points) around selected flaring episodes (light shaded areas). The dotted blue lines show the fitted exponential profiles, with their sums shown in solid blue. The dark shaded areas indicate the periods considered for the SED modeling (Section 4.7). The VERITAS data points and 95% upper limits are shown as black squares and downwards arrows.

	Amplitude (F_0) [10^{-9} erg cm^{-2} s^{-1}]	$t_{\rm peak}$ [MJD]	$t_{\rm rise}$ [days]	$t_{\rm decay}$ [days]
		PKS 1222+216		
	Flare 4 (MJD 55603.7 – 55639.7): χ^2/d.o.f.= 102.25/69 = 1.48			
	0.56 ± 0.09	55607.1 ± 0.3	1.5 ± 0.5	2.7 ± 0.8
	0.48 ± 0.03	55629.9 ± 1.2	22.3 ± 3.5	12.1 ± 2.2
	Flare 8 (MJD 56680.7 – 56752.7): χ^2/d.o.f.= 166.40/104 = 1.60			
	0.72 ± 0.09	56692.9 ± 0.1	13.6 ± 2.7	0.4 ± 0.3
	1.75 ± 0.17	56702.8 ± 0.2	3.6 ± 0.6	1.5 ± 0.3
*	0.43 ± 0.05	56721.9 ± 1.6	19.9 ± 8.5	**10.4 ± 6.2**
	0.41 ± 0.10	56732.0 ± 0.4	0.9 ± 0.8	8.6 ± 3.6
	0.44 ± 0.06	56746.9 ± 0.1	0.3 ± 0.3	10.3 ± 2.5
		Ton 599		
	Flare 4 (MJD 58042.7 – 58140.7): χ^2/d.o.f.= 456.78/296 = 1.54			
	1.89 ± 0.29	58057.1 ± 0.2	1.1 ± 0.3	1.9 ± 0.6
	1.06 ± 0.11	58065.4 ± 1.0	11.9 ± 2.6	9.0 ± 2.1
*	1.37 ± 0.06	58103.5 ± 0.8	47.0 ± 4.7	**11.8 ± 1.1**
	Flare 5 (MJD 58217.7 – 58266.7): χ^2/d.o.f.= 153.01/96 = 1.59			
	0.57 ± 0.12	58219.2 ± 1.3	3.5 ± 2.8	7.3 ± 2.1
	0.48 ± 0.03	58246.3 ± 0.7	6.6 ± 1.7	34.9 ± 5.6

Table 4.9: Results of the LAT flare profile fits for PKS 1222+216 and Ton 599. The flare components coincident with VHE flares are marked with a *, with corresponding smallest variability times indicated in boldface.

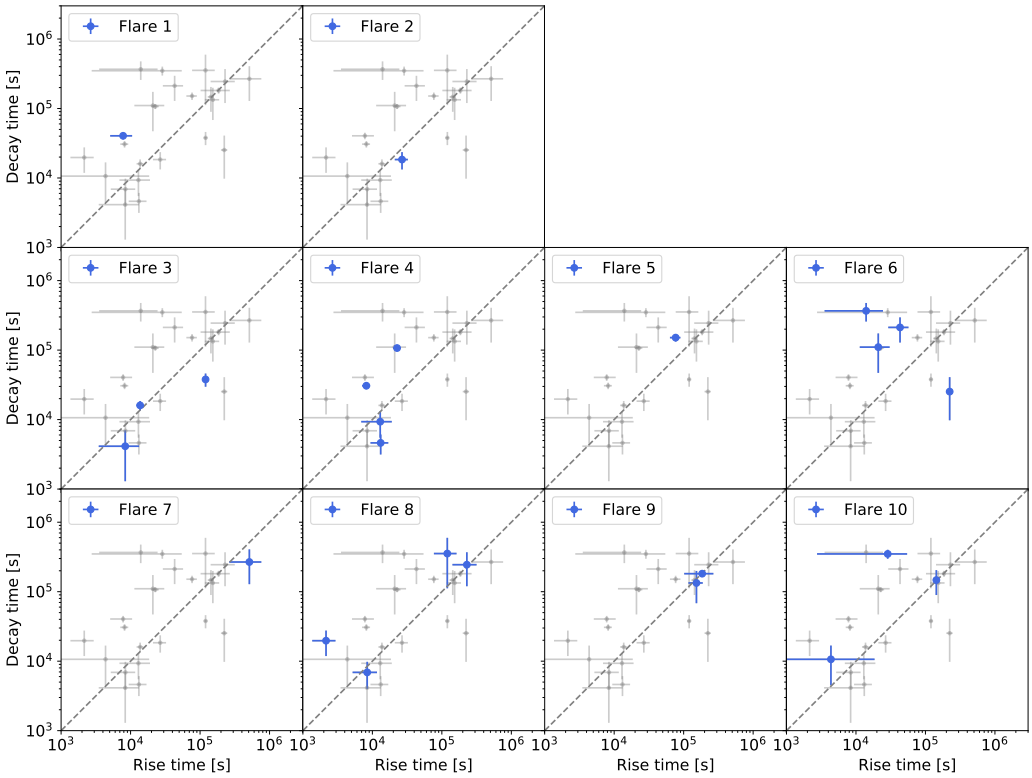


Figure 4.7: Decay time vs. rise time for each of the resolved exponential components in the flares of 3C 279. The points corresponding to all of the components are shown in gray. The dashed reference line shows where the rise and decay times are equal.

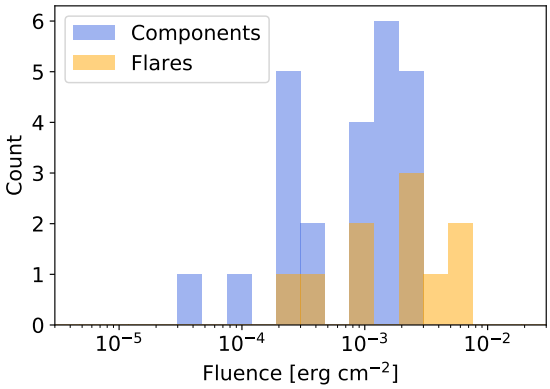


Figure 4.8: Fluence distributions of the twenty-four resolved flare components and ten flares of 3C 279.

Both longer decay times and longer rise times are observed, and many flares appear symmetric. A Wilcoxon signed-rank test (Wilcoxon, 1945) finds no significant preference ($p = 0.178$) for flares to have a faster rise time than decay time rather than the reverse. These findings are consistent with previous studies of gamma-ray flares in bright *Fermi* blazars (e.g. Abdo et al., 2010a; N. Roy et al., 2019).

Models of blazar flares powered by relativistic reconnection predict that flare components produced by large, non-relativistic plasmoids should have similar fluences to components produced by small, relativistic ones, so that flare components should have similar fluence regardless of their variability timescales (Petropoulou, Giannios, and Sironi, 2016). Figure 4.8 shows the distributions of fluences of the components of the ten flares and the twenty-four individual flare components of 3C 279. The fluence of a flare with exponential components F_i is given by:

$$\mathcal{F} = \sum_i F_i(t_{\rm rise} + t_{\rm decay}). \tag{4.5}$$

For 3C 279 Flares 1, 2, 5, and 7, the best fit is given by a single component plus a constant baseline flux. In these cases, the baseline flux is included in the fluence estimate

for consistency with the other flares, approximating the flare duration as $t_{\rm rise} + t_{\rm decay}$, so that the fluence is given by:

$$\mathcal{F} = (F_0 + F_{\rm const})(t_{\rm rise} + t_{\rm decay}). \tag{4.6}$$

The median flare fluence is 2.1×10^{-3} erg cm^{-2} and the median component fluence is 0.85×10^{-3} erg cm^{-2}. The observed component fluences range over about one order of magnitude, as do the flare amplitudes, while the rise and decay timescales span about two orders of magnitude. These dynamic ranges are generally compatible with the expectations for plasmoid-powered flares derived from particle-in-cell simulations of relativistic magnetic reconnection (Petropoulou, Giannios, and Sironi, 2016).

The long-term gamma-ray variability study of the three FSRQs presented here is compatible with the extensive flare characteristics study carried out recently by M. Meyer, Scargle, and Blandford, 2019 on the brightest flares detected by *Fermi*-LAT. A similar Bayesian blocks analysis was carried out to identify flares and look for variability on sub-hour timescales. Consistent with their findings, we find sub-hour-scale variability in 3C 279, where it was possible to resolve flares in finer time bins, suggesting that extremely compact emission regions may be present within the jet.

4.6 Gamma-ray Spectra

Figure 4.9 shows the LAT energy spectra corresponding to the entire data sets of each of the three sources, along with the VERITAS spectra for 3C 279. The best-fit spectral parameters are reported in Appendix B. Since all three sources were best fit by a log-parabola model in the 4FGL catalog (Abdollahi et al., 2020), we fit the LAT spectra with this model, parametrized as

$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_b}\right)^{-(\alpha+\beta(\log(E/E_b))}, \tag{4.7}$$

where $E_b = 457.698$ MeV.

We checked that the log-parabola model provides a better fit than a power-law model using the likelihood ratio test. A power-law sub-exponential cutoff model was also preferred over a power law, but we could not compare it directly to the log-parabola model using a likelihood ratio test, since the two curved models are non-nested, with neither being a special case of the other. We assumed a log-parabola spectrum for all subsequent LAT analyses. To facilitate comparison with the VERITAS points, the LAT model fits and butterfly contours were extended beyond the LAT maximum energy of 500 GeV, and extragalactic background light absorption was applied to them using the model of Franceschini and Rodighiero, 2017.

The global spectral shapes of the three sources are similar, with an index α of $\sim$2.1–2.3 and a curvature parameter β of $\sim$0.04–0.06, and they differ primarily by their normalization.

Using the data from 3C 279, we compared several methods to determine a baseline non-flaring spectrum. First, we defined a low state lasting from MJD 56230 – 56465 (see Figure 4.1), during which the flux was quiescent and stable in HE gamma rays, R-band optical, and X-rays. We checked publicly available Tuorla[3] data for the R-band light curve. For the X-rays, we analyzed the *Swift*-XRT light curve using the online data products generator[4]. To ensure low, stable gamma-ray emission, we selected the interval to span the Bayesian blocks with the lowest flux while excluding intervals with the sun in the ROI. The low-state LAT SED is shown in Figure 4.9. Only one VERITAS observation occurred during this interval, on MJD 56417. The corresponding VERITAS upper limits are not constraining and are not shown.

Next, using the algorithm described in Section 4.4, we designated all epochs of the LAT

[3] `https://users.utu.fi/kani/1m/3C_279_jy.html`
[4] `https://www.swift.ac.uk/user_objects/`

light curve other than the flaring episodes as quiescent. From those epochs, we extracted those LAT data strictly simultaneous with the VERITAS observations, integrating a total of 43.6 hours of observations. The resulting strictly simultaneous LAT and VERITAS spectra are shown in Figure 4.9. We then performed the same procedure for four flaring epochs during which a significant *Fermi*-LAT detection could be obtained strictly simultaneous with the VERITAS observations, which occurred on the nights of April 3, 2014; June 16, 2015; April 19, 2018; and June 3, 2018. These strictly simultaneous LAT and VERITAS SEDs are shown in Figure 4.10.

The spectral shapes of the 3C 279 low and quiescent states are similar to each other and to the global state, although the fit parameters have high uncertainties due to the low significance. The spectra differ primarily in flux normalization. The normalization of the low state is lower than that of the global state by design, while the normalization of the strictly simultaneous quiescent state is higher. This could result from the timing of the VERITAS monitoring and triggered observations which often follow up on *Fermi* flares and may tend to catch mildly elevated activity in *Fermi*-LAT even if the source is not actually flaring.

Finally, we derived LAT SEDs for all of the ten identified flares of 3C 279, using the entire flare time periods, irrespective of strict simultaneity with VERITAS, shown in Figure 4.11. The average flare spectrum is more strongly curved than the global spectrum, with $\alpha = 2.02 \pm 0.01$ and $\beta = 0.093 \pm 0.008$, compared to $\alpha = 2.228 \pm 0.004$ and $\beta = 0.061 \pm 0.003$ for the global state.

4.7 SED Modeling

Multiwavelength SED modeling can shed light on the mechanisms of gamma-ray production during TeV flares. TeV detections of FSRQs are particularly interesting because these sources possess external radiation fields that might be expected to produce increased Comp-

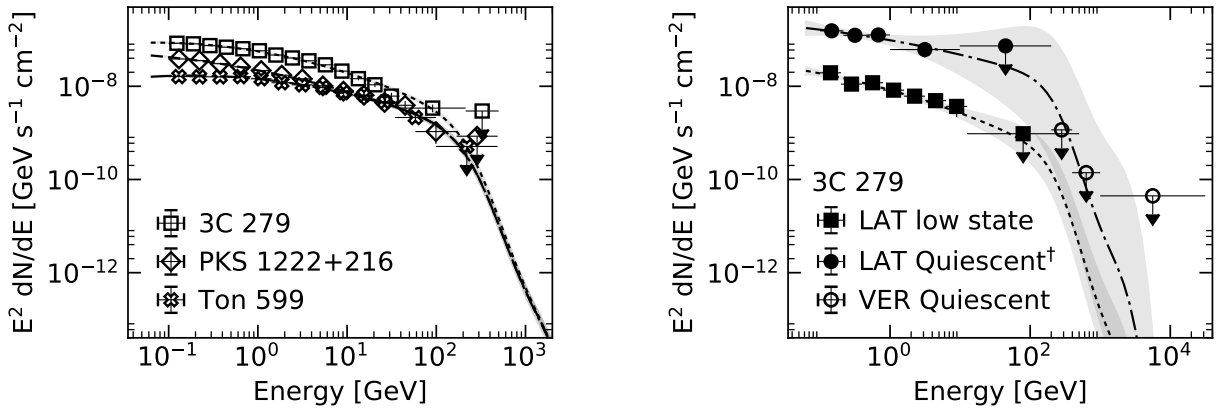


Figure 4.9: *Left:* Global *Fermi*-LAT spectra for 3C 279, PKS 1222+216, and Ton 599. The LAT spectra are extrapolated to the VERITAS energy range, incorporating EBL absorption. *Right:* Two baseline states of 3C 279. The *Fermi*-LAT spectrum corresponding to the multiwavelength low state (MJD 56230–56465; see Figure 4.1), is shown by filled squares. The strictly simultaneous *Fermi*-LAT and VERITAS spectra during the quiescent state are shown by black filled and unfilled circles and contours. The symbol "†" indicates that the LAT spectrum corresponds to data strictly simultaneous with VERITAS observations.

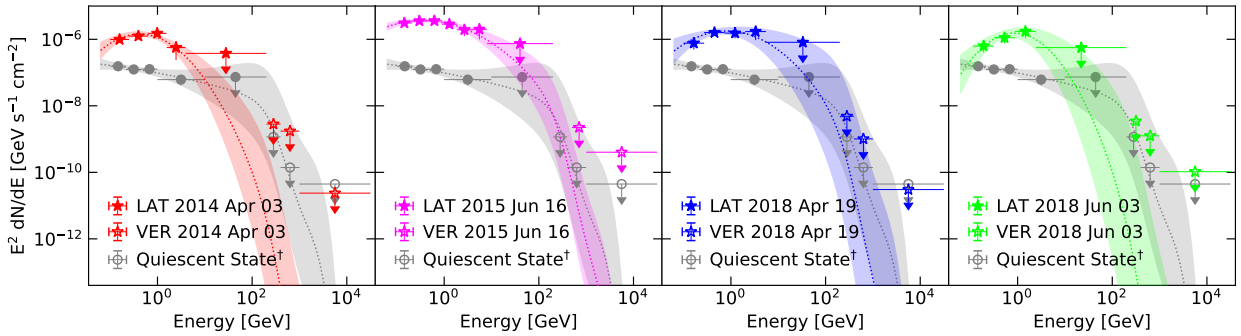


Figure 4.10: *Fermi*-LAT and VERITAS spectra of 3C 279 during four flares, strictly simultaneous with the VERITAS observations. For comparison, the quiescent spectrum (gray circles and contour) is shown in the four panels. The strictly simultaneous quiescent state LAT data and VERITAS upper limits are represented with filled and unfilled markers, respectively. The LAT spectra are extrapolated to the VERITAS energy range, incorporating EBL absorption.

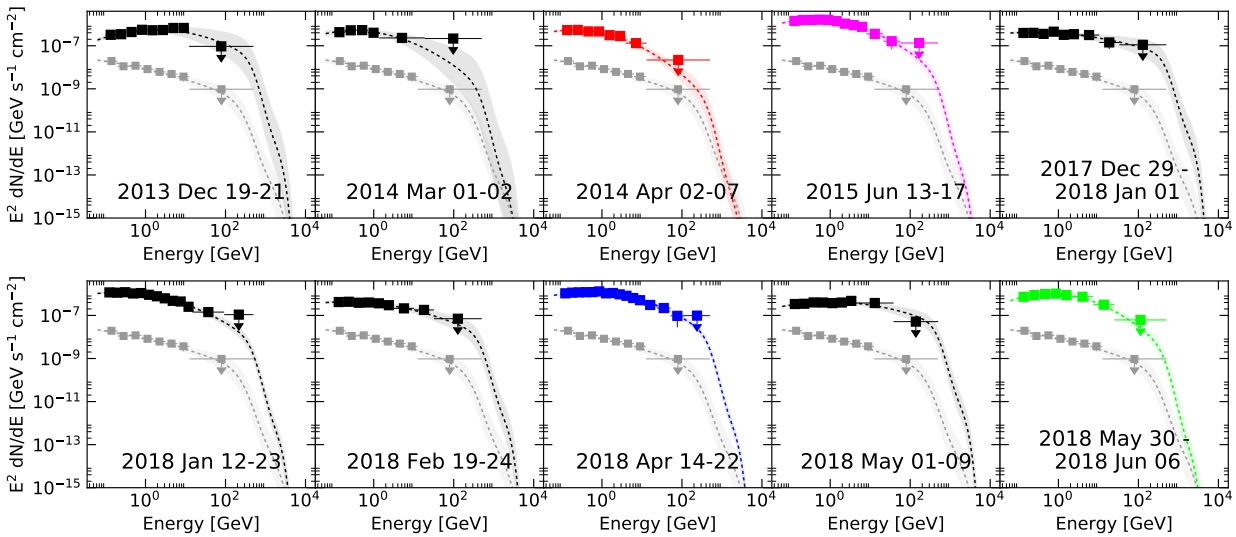


Figure 4.11: *Fermi*-LAT spectra of 3C 279 during ten flares, for the intervals selected using the algorithm proposed in this work and described in Section 4.4. For comparison, the LAT low state spectrum is shown in gray squares in all of the panels. The four flares shown in color have a corresponding spectrum in Figure 4.10.

ton cooling of electrons and to absorb energetic gamma rays by pair production, leading to a cutoff in the gamma-ray spectrum above the GeV band (e.g. Ghisellini et al., 1998). For 3C 279, we refer the reader to those works in the literature in which multiwavelength SED modeling of the epochs considered here has been performed, and we do not perform any additional modeling (see for example, Hayashida et al., 2015; Ackermann et al., 2016; Prince, 2020; Yoo and An, 2020).

PKS 1222+216 was first detected at VHE energies by MAGIC during a flaring event in June 2010 (Aleksić et al., 2011), and multiwavelength SED modeling of this event has been performed by e.g. Tavecchio et al., 2011. We restricted our SED modeling of the source to the duration of the second VHE detection by VERITAS in February and March 2014. We considered data from all instruments taken from UT 2014-02-26 to 2014-03-10, inclusive.

Ton 599 has not been studied as extensively as the other two sources. Prince, 2019 and Patel and Chitnis, 2020 model its variability characteristics and multiwavelength SED,

respectively, during the high state in December 2017, but do not have access to TeV data. We therefore modeled the multiwavelength SED of Ton 599 during the VERITAS detection in December 2017. We considered data from all instruments taken from UT 2017-12-15 to 2017-12-16, inclusive.

To assemble our multiwavelength SEDs, in addition to gamma-ray data from VERITAS and *Fermi*, we incorporated X-ray and ultraviolet data from the XRT and UVOT instruments aboard the *Swift* satellite and optical observations from the Steward Observatory.

We fit the multiwavelength SEDs of the two FSRQs using a multi-component synchrotron self-Compton (SSC) blob-in-jet model, implemented using the framework of the "`Bjet`" code, developed by Hervet, Boisson, and Sol, 2015 and based on Katarzyński, Sol, and Kus, 2001. We modeled the radiative interactions of a compact leptonic emission zone (a blob), including an EIC emission component resulting from the interactions of the blob particles with the thermal accretion disk emission reprocessed by the BLR. Figure 4.12 shows a schematic illustration of the components producing the blazar emission in this model.

In our model, we considered the blob position to be stationary in the rest frame of the galaxy. This assumption can be interpreted as a relativistic particle flow crossing a standing shock, with the blob radius representing the distance from the shock in the jet frame required for the particles to cool, which is roughly equal to the size of the shock if the cooling time is shorter than the shock-crossing time. However, we note that in a stationary SSC model this scenario is formally indistinguishable from that of a relativistically moving blob.

We consider a simplified BLR model with a normalized density profile, based on Nalewajko, Begelman, and Sikora, 2014, where $\rho_{\rm BLR}(r)$ is at a maximum at the characteristic BLR radius $r = r_{\rm BLR}$ and decreasing as r^{-2} with the distance to the core such that

$$\rho_{\rm BLR}(r) = \frac{(r/r_{\rm BLR})^2}{1 + (r/r_{\rm BLR})^4} \ , \tag{4.8}$$

with $r_{\rm BLR}$ scaled to the bolometric disk luminosity L_d as $r_{\rm BLR} = 0.1\sqrt{L_d/1 \times 10^{46}\ {\rm erg\ s^{-1}}}$ pc (Sikora et al., 2009; Ghisellini and Tavecchio, 2009). From SED modeling of PKS 1222+216 and Ton 599 we deduce a BLR radius of 0.17 pc and 0.15 pc respectively. We assume an isotropic diffusion of the disk light by the BLR where the specific intensity of this field can be expressed as

$$I_{\rm BLR}(\nu, T_d, r) = \epsilon_{\rm BLR}\rho_{\rm BLR}(r)\frac{L_d}{4\pi r^2}\frac{I_p(\nu, T_d)}{(\sigma_{\rm SB}/\pi)T_d^4}, \tag{4.9}$$

where $\sigma_{\rm SB}$ is the Stephan-Boltzmann constant, I_p is the Planck intensity, and $\epsilon_{\rm BLR}$ is the covering factor. This equation is similar to Eq. 12 in Hervet, Boisson, and Sol, 2015 with the addition of the BLR density profile. Only the extension of the BLR in front of the blob plays a significant role in our modeling since it drives the number of gamma rays produced by the blob that will be absorbed by pair production. The BLR is by default defined between $r = 0$ and $r = 100\ r_{\rm BLR}$. Given the fast convergence of the BLR opacity ($I_{\rm BLR} \propto r^{-4}$), the maximum extension of the BLR does not play a significant role in the model. Although we assume for simplicity that the BLR is isotropic, any anisotropy should have a small effect on the opacity (e.g. Abolmasov and Poutanen, 2017, Figure 14).

Figures 4.13 and 4.14 show the multiwavelength SED models of PKS 1222+216 and Ton 599. In these figures, the synchrotron and SSC emission are shown by solid blue lines. The subdominant second-order self-Compton emission caused by the interactions of the electrons with the self-Compton photons is shown by a dotted blue line. The thermal emission from the accretion disk is modeled as a point source radiating as a black body, and is shown by a heavy dashed green line. The inverse Compton emission due to the interaction of the electrons with the disk photons reprocessed in the broad line region is shown by a dashed green line. Table 4.10 gives the parameters characterizing the SED models.

Our model does not include any secondary radiation from pair cascades produced by the

Parameter	PKS 1222+216	Ton 599	Unit
$\theta_{\rm obs}$	1.0	1.0	deg
Blob			
δ	40	53	
$N_e^{(1)}$	2.0×10^4	2.7×10^5	$\rm cm^{-3}$
n_1	2.1	2.5	–
n_2	3.9	3.0	–
$\gamma_{\rm min}$	5.5×10^2	3.0×10^2	–
$\gamma_{\rm max}$	3.0×10^5	7.0×10^4	–
$\gamma_{\rm brk}$	5.0×10^3	1.5×10^4	–
B	3.0×10^{-2}	3.0×10^{-2}	G
R	5.5×10^{16}	6.0×10^{16}	cm
$D_{\rm BH}$	3.56	2.33	pc
Nucleus			
$L_{\rm disk}$	2.8×10^{46}	2.2×10^{46}	$\rm erg\ s^{-1}$
$T_{\rm disk}$	2.8×10^4	1.1×10^4	K
$\epsilon_{\rm BLR}$	2.0×10^{-2}	2.0×10^{-2}	–

Table 4.10: Parameters of the SED models. $\theta_{\rm obs}$ is the angle of the blob direction of motion with respect to the line of sight. The electron energy distribution between Lorentz factors $\gamma_{\rm min}$ and $\gamma_{\rm max}$ is given by a broken power law with indices n_1 and n_2 below and above $\gamma_{\rm brk}$, with $N_e^{(1)}$ the normalization factor at $\gamma = 1$. The blob Doppler factor, magnetic field, radius, and distance to the black hole are given by δ, B, R, and $D_{\rm BH}$, respectively. $D_{\rm BH}$ is reported in the host galaxy frame. The disk luminosity and temperature are given by $L_{\rm disk}$ and $T_{\rm disk}$, while $\epsilon_{\rm BLR}$ is the covering factor of the broad line region.

absorption of gamma rays in the BLR. While detailed modeling of this effect is beyond the scope of this paper, we estimate that the potential contribution of such cascades would be $\ll 1\%$ of the total bolometric luminosity for PKS 1222+216 and no more than 1.1% for Ton 599, given the respective levels of absorption in our models, which are described below. This effect may be noted as a source of systematic uncertainty when interpreting our results.

We note that our model requires that the dust torus luminosity be negligible compared to the disk luminosity. As evidence of far-infrared dust torus thermal emission is lacking in the SED, we consider this assumption to be reasonable in our study. Observing campaigns with good microwave to IR coverage would be needed to fully confirm this approach. The

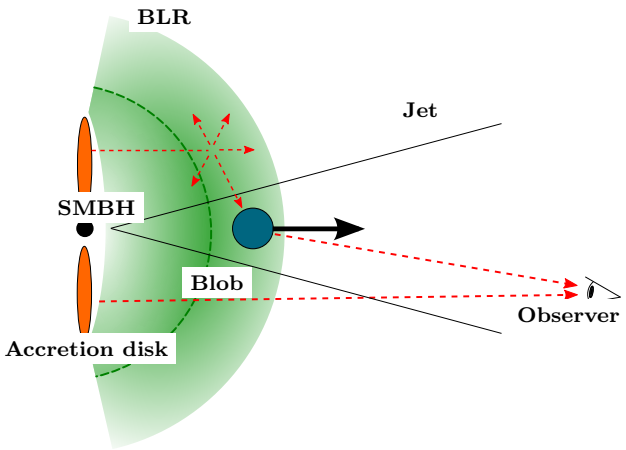


Figure 4.12: A schematic illustration of the emission model used in this work (not to scale). The green dashed arc represents the nominal BLR radius $r_{\rm BLR}$ corresponding to the region of the maximal BLR density. The observer measures the beamed emission from the blob interacting with the BLR as well as the accretion disk's thermal emission. The accretion disk is assumed to be a point source.

presence of strong dust torus emission would require that the gamma-ray emission zone be farther downstream in the jet so as not to produce too large an opacity by pair production.

4.7.1 PKS 1222+216 Modeling

In order to investigate the necessity of including an EIC component, we fit the multi-wavelength SED of PKS 1222+216 with a one-zone pure SSC model, shown in Figure 4.13 (*left*). As can be seen by the similar amplitudes of the synchrotron and inverse Compton peaks in the figure, the SED is only weakly Compton dominated, with the inverse Compton luminosity about twice the synchrotron luminosity. The *Swift*-XRT spectrum contains a well-resolved break showing the transition between synchrotron and inverse Compton dominated emission, which sets a strong constraint on the model. Our best attempt does not provide a satisfying representation of the observed SED. The main issue is that the optical-to-X-ray components of the SED have steep slopes which would require a narrow, sharp synchrotron bump to achieve a good fit, while the X-ray-to-VHE needs a wide, flat inverse

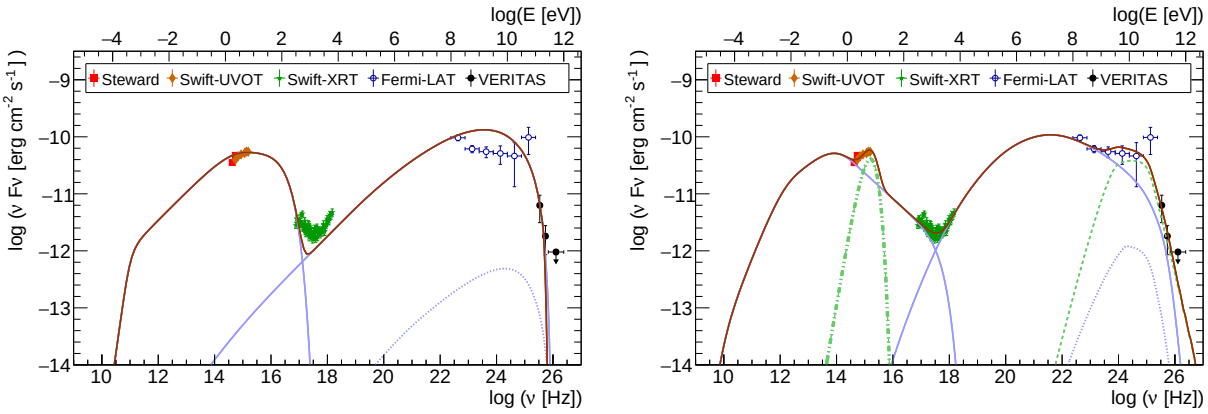


Figure 4.13: Broadband SED of PKS 1222+216 during the VERITAS detection from UT 2014-02-26 to 2014-03-10. *Left:* Pure synchrotron self-Compton model. *Right:* Model considering an external inverse Compton component at high energy from the interaction of blob particles with the thermal accretion disk emission reprocessed by the BLR. The SED model components are described in the main text. The EBL absorption is taken into account considering the model of Franceschini and Rodighiero, 2017.

Compton bump. This is not compatible with the usual simple SSC framework, especially when the SED is not heavily Compton dominated.

In our EIC model, the IR-to-UV SED is dominated by the blackbody big-blue-bump emission of an accretion disk (see Figure 4.13, *right*), which resolves the tension by eliminating the constraint on the synchrotron spectral shape. This allows for a broad SSC component matching the spectral break observed in the X-ray band. In this scenario, the VHE emission is produced by the EIC process between a relativistic blob and the disk thermal emission reprocessed by the BLR. The blob is set to a distance of 3.56 pc from the SMBH, corresponding to 21.3 r_{BLR}. It should be noted that a thermal EIC process was favored in previous models of PKS 1222+216 where clear disk emission and a strongly Compton-dominated SED were observed during a major outburst in 2010 (Tavecchio et al., 2011).

Because the peak frequency of the EIC emission is directly proportional to the blob Lorentz factor, this scenario imposes a strong constraint on the jet parameters. In the case of PKS 1222+216, in order to match the VHE spectrum, the bulk Lorentz factor needs to

be above approximately 23, which was achieved by assuming a Doppler factor $\delta = 40$ and an angle with the line of sight $\theta_{\rm obs} = 1°$. This assumption is consistent with the jet constraints derived by Hervet, Boisson, and Sol (2016) from the fastest motion observed in the radio jet of PKS 1222+216, which led to estimations of $\theta_{\rm obs} = 1.3$, $\delta = 41.3$ and $\Gamma = 29.2$.

Our model predicts a possible minimal variability of 18 hours, relatively close to the fastest variability observed during the full period of GeV flaring activity. The total power of the jet is approximately 3.4×10^{45} erg s^{-1}, in a particle-dominated regime with the equipartition parameter $U_B/U_e = 1.7 \times 10^{-3}$.

4.7.2 Ton 599 Modeling

Contrary to PKS 1222+216, the SED of Ton 599 is heavily Compton dominated, with a ratio of inverse Compton to synchrotron luminosity of approximately one order of magnitude. This is a usual signature of an EIC component dominating the gamma-ray emission. We therefore consider the same scenario as for PKS 1222+216. As shown in Figure 4.14, the model provides a good fit to the data.

As in the case of PKS 1222+216, the thermal EIC emission imposes strong constraints on the properties of the emitting region. The largest constraint comes from the gamma-ray opacity by pair creation from the luminous thermal field surrounding the blob. We found that only for a Doppler factor of $\gtrsim$50 is the EIC emission at VHE strong enough to produce the observed VHE gamma rays, given the BLR opacity. The solution presented in Figure 4.14, with $\delta = 53$, is consistent with a maximum VHE emission undergoing strong BLR absorption ($E_{\rm max} = 630$ GeV), with an opacity of $\tau_{\gamma\gamma,E_{\rm max}} = 2.8$. In this scenario we set the blob at a distance of 2.33 pc from the SMBH, corresponding to 15.7 $r_{\rm BLR}$.

By coincidence, the fastest possible variability of Ton 599 from the model is 18 hours, similar to that of PKS 1222+216, consistent with the variability timescale of $\sim$2 days observed with VERITAS. The jet is estimated to have a total power of approximately 1.2×10^{46} erg s^{-1},

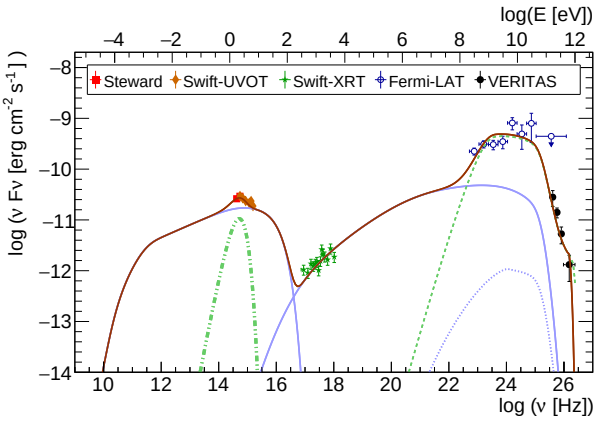


Figure 4.14: Broadband SED of Ton 599 during the VERITAS detection from UT 2017-12-15 to 2017-12-16. The SED model components are described in the main text.

and to be extremely particle-dominated with equipartition parameter $U_B/U_e = 3.8 \times 10^{-4}$.

4.8 Lorentz Factors and Locations of the Gamma-ray Emitting Regions

We determined constraints on the Lorentz factor Γ and distance r of the gamma-ray emission region from the central black hole for PKS 1222+216 and Ton 599 following the method and assumptions of Nalewajko, Begelman, and Sikora (2014). The constraints are plotted in Figure 4.15. The parameters used to determine the constraints are given in Table 4.11. In order to obtain a conservative SSC constraint, we set the SSC luminosity equal to the observed gamma-ray luminosity $L_{\rm gamma}$. For PKS 1222+216 the fastest variability is observed with *Fermi*-LAT, while for Ton 599 it is observed with VERITAS. We therefore set the maximum energy $E_{\rm cool}$ for the EIC cooling constraint equal to the geometric mean of the energy ranges observed by *Fermi*-LAT and VERITAS for the two sources, respectively.

Source	z	D_L Gpc	$t_{\rm var}$ day	$L_{\rm syn}$ erg s^{-1}	$L_{\rm gamma}$ erg s^{-1}	L_d erg s^{-1}	$M^a_{\rm BH}$ $M_\odot$	$E_{\rm cool}$ GeV	$\epsilon^b_{\rm BLR}$	$\epsilon^b_{\rm IR}$
PKS 1222+216	0.434	2.44	10.0	3.5×10^{46}	7.8×10^{46}	2.8×10^{46}	3.47×10^8	7.07	0.02	0.2
Ton 599	0.725	4.54	2.0	4.4×10^{46}	1.2×10^{48}	2.2×10^{46}	6.8×10^8	326	0.02	0.2

Table 4.11: Parameters used to calculate the parameter space. z and D_L are the redshift and luminosity distance of the source. $t_{\rm var}$ is the variability timescale of cooling derived from each flare's fitted exponential decay. $L_{\rm syn}$, $L_{\rm gamma}$, and L_d are the synchrotron luminosity, gamma-ray luminosity, and disk luminosity from the SED model. $M_{\rm BH}$ is the black hole mass. $E_{\rm cool}$ is the maximum photon energy due to the external Compton cooling of relativistic electrons. $\epsilon_{\rm BLR}$ and $\epsilon_{\rm IR}$ are the covering factors of the broad line region and IR-emitting torus region, respectively. [a]Farina et al. (2012) and Liu, Jiang, and Gu (2006). [b]Tavecchio et al. (2011).

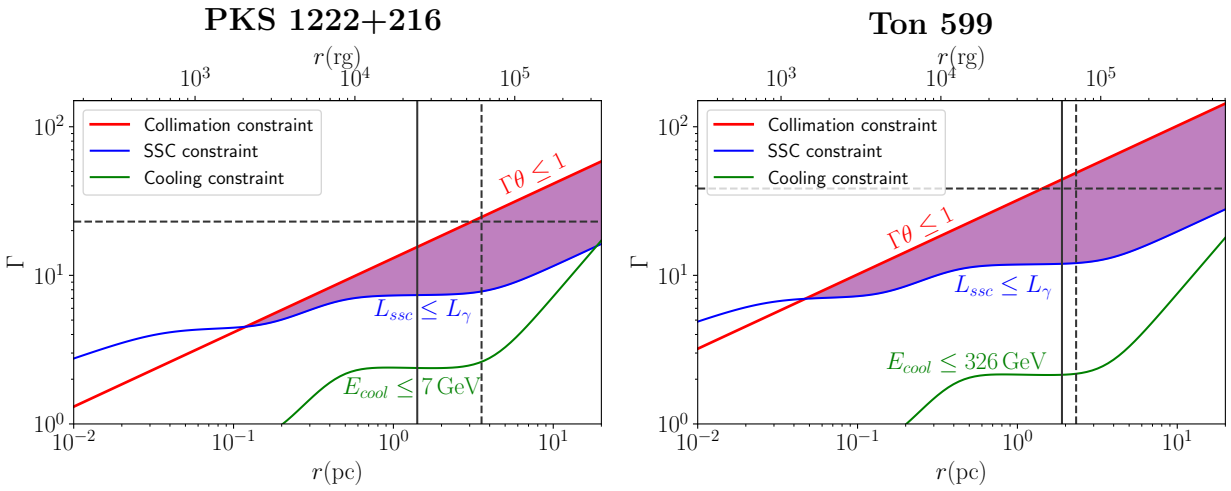


Figure 4.15: Constraints on the Lorentz factor Γ and distance r between the gamma-ray emission location and central black hole, adapted from Nalewajko, Begelman, and Sikora, 2014. The allowed region is filled in purple. The black vertical line shows the opacity constraint on r from the BLR modeling. The values of Γ and r derived from the SED model are shown with dashed black lines.

Three constraints on Γ and r are calculated. The collimation constraint requires that the emission region be smaller than the size of the jet at the emission region location such that $\Gamma\theta \leq 1$, where θ as defined by Nalewajko, Begelman, and Sikora (2014, Eq. 1) is the angle subtended by the blob expanding while propagating. In our case, considering a stationary shock, it would refer to the expansion of the relativistic flow passing through the shock.

A caveat of this constraint is the underlying assumption that the blob size is defined by the observed variability such that $R = c\delta t_{\mathrm{var,obs}}/(1+z)$. However, the observed variability only gives an upper limit on the blob radius, meaning that the actual size of the emission zone is likely smaller than that extrapolated from the observed variability. Indeed, our modeling of PKS 1222+216 and Ton 599 predicts a minimal observed variability much shorter than the one observed within the reconstructed SED periods. This discrepancy explains why the parameters predicted by our model lie only just below the line $\Gamma\theta = 1$.

The SSC constraint requires that the SSC luminosity not exceed the total gamma-ray luminosity, which includes contributions from external radiation fields (Nalewajko, Begelman, and Sikora, 2014, Eq. 5). The cooling constraint requires that electrons radiatively emitting

gamma rays at energies above $E_{\rm cool}$ cool through interactions with external radiation fields faster than the flare decay timescale (Nalewajko, Begelman, and Sikora, 2014, Eq. 9).

These parameter spaces do not take into account the constraints given by the BLR and dust torus opacity on the gamma-ray emission. We show with black vertical lines the minimum distance r from the black hole in the SED models where the BLR would become fully opaque for the maximum observed energy $E_{\rm max}$ (370 GeV for PKS 1222+216 and 630 GeV for Ton 599). We consider the BLR opaque when $\tau_{{\rm BLR},E_{\rm max}} > 5$, meaning that less than 1% of the gamma rays can escape. We can clearly see that considering the BLR opacity significantly tightens the constraints on the gamma-ray emission location in Ton 599, as mentioned in the previous section. The opacity constraint on PKS 1222+216 is weaker, as in that case the blob does not have to be as deep inside the BLR to reproduce the observed EIC emission.

4.9 Neutrino Emission During VHE Flares

Luminous gamma-ray flares of FSRQs are potential sources of PeV-scale ($\sim$100 TeV – $\sim$10 PeV) neutrino emission (e.g. Mannheim, 1993; Dermer, Murase, and Inoue, 2014; Kadler et al., 2016). It has been proposed that the first blazar associated with the production of high-energy neutrinos, TXS 0506+056, may be an FSRQ masquerading as a BL Lac object (Padovani et al., 2019). While the lack of point sources observed in IceCube data suggests that FSRQs are not the dominant population of neutrino sources, the possibility of neutrino emission during rare, bright flares has not been excluded (Murase and Waxman, 2016). While Righi et al. (2020) have suggested that the bulk of the average neutrino emission from FSRQs occurs in the sub-EeV – EeV energy range, their results do not exclude PeV-scale neutrino emission during outlier states. In the SED modeling of the VHE flares of PKS 1222+216 and Ton 599 presented here, a purely leptonic model gives an adequate fit to the data, and

performing full hadronic modeling is beyond the scope of this work.

However, we can place analytic constraints on the potential PeV-scale neutrino flux produced during these events by considering a lepto-hadronic scenario in which synchrotron emission from secondary electrons produced by pion decay contributes a subdominant component to the second peak of the SED, similar to models used to describe the flaring emission of TXS 0506+056 coincident with the detection of a neutrino by IceCube (Gao et al., 2019; Keivani et al., 2018; Reimer, Böttcher, and Buson, 2019). In this section, we make use of the assumptions and methods of Gao, Pohl, and Winter (2017), particularly Appendix A of that work. All quantities in the following equations are in the comoving frame of the blob, unless explicitly noted with the superscript "ob".

We consider neutrinos produced by the $p\gamma$ interaction via the $\Delta^+(1232)$ resonance with threshold energy $\epsilon_{p\gamma,\mathrm{th}} \sim 0.3$ GeV. The characteristic proton energy is $E_{p,\mathrm{char}} \sim E_\nu/K_\nu \sim 2$ PeV, where $E_\nu = E_\nu^{\mathrm{ob}}(1+z)/\Gamma$ and $K_\nu \sim 0.05$ (Murase, Inoue, and Dermer, 2014). Therefore, to check whether these sources could in principle support PeV neutrino emission, we first estimate the maximum energy to which protons can be accelerated in the source without escaping, following Hillas (1984), as

$$E_{p,\mathrm{max}} = Ze\beta cBR, \tag{4.10}$$

where the atomic number $Z = 1$ for protons, e is the elementary charge, $\beta = v/c \sim 1$ for highly relativistic particles, c is the speed of light, B is the magnetic field in the source, and R is the size of the source. Using the values in Table 4.10, the maximum energy to which protons could have been accelerated in the gamma-ray emission regions for the flares of PKS 1222+216 and Ton 599 is $E_p \sim 500$ PeV, equivalent to an upper limit on the neutrino energy of $E_\nu^{\mathrm{ob}} \sim 400$ PeV, so PeV-scale neutrino emission is certainly feasible.

A limit on the neutrino flux can be imposed by considering a steady state in which the

synchrotron luminosity of the secondary electrons equals the power injected by pion decay. The steady-state proton energy density at $E_{p,\rm char}$ is given by (Gao, Pohl, and Winter, 2017):

$$u_p(E_{p,\rm char}) = \frac{\alpha_{\rm fs}\epsilon_{p\gamma,\rm th}}{c\sigma_{p\gamma}K_{p\gamma}K_{\pi\to e}}\frac{m_pc^2}{E_{p,\rm char}}\frac{\nu F^{\rm ob}_{\nu,2}}{\nu F^{\rm ob}_{\nu,\rm t}}, \tag{4.11}$$

where $\alpha_{\rm fs} = 4c/(3R)$ is the free-streaming escape rate, $\sigma_{p\gamma} = 5.0\times10^{-28}$ cm^2 is the $p\gamma$ cross-section, $K_{p\gamma}\sim0.2$ is the average inelasticity for the proton in the $p\gamma$ interaction, $K_{\pi\to e}\sim1/8$ is the fraction of energy transferred to $e^\pm$ pairs from pion decay, $\nu F^{\rm ob}_{\nu,2}$ is the observed flux due to synchrotron emission from the secondary electrons, and $\nu F^{\rm ob}_{\nu,\rm t}$ is the observed flux of the target photons of the $p\gamma$ interaction at $E_{\rm t}\sim\epsilon_{p\gamma,\rm th}m_pc^2/E_{p,\rm char}$, which is directly constrained by the *Swift*-XRT measurement at $E_{\rm t}\Gamma/(1+z)\sim2$ keV. We can estimate $\nu F^{\rm ob}_{\nu,2}$ from the SED at the synchrotron peak of the secondary electrons at

$$\nu^{\rm ob}_2 = \frac{ce(K_{p\gamma}K_{\pi\to e})^2}{2\pi(m_ec^2)^3}\frac{\Gamma}{1+z}BE_{p,\rm char} \approx 10^{22}\ \left(\frac{\Gamma}{23}\right)\left(\frac{B}{30\ \rm mG}\right)\left(\frac{E_{p,\rm char}}{2\ \rm PeV}\right)\ \rm Hz, \tag{4.12}$$

for redshift $z\sim0.5$. The corresponding power in protons can be estimated as

$$L_p \sim \Gamma^2u_p(E_{p,\rm char})\alpha_{\rm esc}V(R), \tag{4.13}$$

where for simplicity we assume the proton escape time $\alpha_{\rm esc} = 0.1\alpha_{\rm fs}$ and let $V(R) = 4\pi R^3/3$ for a spherical blob. Parameterizing this power by the Eddington luminosity boosted into the jet frame yields

$$L_p \sim 1\left(\frac{\Gamma}{23}\right)^2\left(\frac{R}{6\times10^{16}\ \rm cm}\right)\left(\frac{M}{5\times10^{10}\ \rm M_\odot}\right)^{-1}\left(\frac{E_{p,\rm char}}{2\ \rm PeV}\right)^{-1}\left(\frac{\nu F^{\rm ob}_{\nu,2}/\nu F^{\rm ob}_{\nu,\rm t}}{0.5}\right)\ L_{\rm Edd}, \tag{4.14}$$

where $\nu F^{\rm ob}_{\nu,2} \lesssim 0.5\ \nu F^{\rm ob}_{\nu,{\rm t}} \sim 1 \times 10^{-12}$ erg cm^{-2} s^{-1} is a conservative estimate of the largest energetically reasonable contribution[5] to the SED at $\sim 10^{22}$ Hz. The contribution is clearly subdominant. We can then estimate the observed neutrino energy flux, where $\dot{N}_{p\gamma}$ is the $p\gamma$ event rate per physical volume, using the relation

$$\dot{N}_{p\gamma} \sim c\sigma_{p\gamma} \frac{u_{\rm ph}(E_{\rm t})}{E_{\rm t}} \frac{u_p(E_{p,{\rm char}})}{E_{p,{\rm char}}} = \frac{\alpha_{\rm fs} u_\nu(E_\nu)}{E_\nu}, \tag{4.15}$$

where $u_{\rm ph}(E_{\rm t})$ is the energy density of the target photons. As $u_\nu(E_\nu)/u_{\rm ph}(E_{\rm t}) = \nu F^{\rm ob}_{\nu,\nu}/\nu F^{\rm ob}_{\nu,{\rm t}}$, we obtain the simple relation

$$\nu F^{\rm ob}_{\nu,\nu} = \frac{K_\nu}{K_{p\gamma} K_{\pi\to e}} \nu F^{\rm ob}_{\nu,2} \sim 2\ \nu F^{\rm ob}_{\nu,2}. \tag{4.16}$$

The number of PeV-scale neutrinos of any flavor expected to be detected by IceCube during the VHE flare of PKS 1222+216 or Ton 599 is then

$$N_\nu \lesssim 0.001 \left(\frac{\nu F^{\rm ob}_{\nu,\nu}}{2 \times 10^{-12}\ {\rm erg\ cm^{-2}\ s^{-1}}}\right) \left(\frac{\Delta T}{5\ {\rm day}}\right) \left(\frac{\mathcal{A}_{\rm eff}}{10^6\ {\rm cm^2}}\right), \tag{4.17}$$

where ΔT is the duration of the VHE flare, $\mathcal{A}_{\rm eff} \sim 10^6$ cm^2 is the IceCube effective area for extremely high-energy real-time alerts in the PeV range (M. Aartsen et al., 2017), and $\Delta\nu \sim \ln(10)$ is assumed for the width of the neutrino spectrum. We conclude that it is plausible that PKS 1222+216 and Ton 599 could have produced PeV-scale neutrinos during their TeV flaring activity at a flux consistent with a null detection by current instruments.

To reduce the model-dependence of our constraints, R could also be estimated using the timescale of gamma-ray flare variability,

$$R \sim \frac{\delta}{1+z} c\Delta T, \tag{4.18}$$

[5]This assumption requires about 5×10^{-3} protons for every electron, from Eq. 4.11 and the electron energy distributions reported in Table 4.10.

from which estimates of $R \sim 7 \times 10^{17}$ cm and 1×10^{17} cm are obtained for PKS 1222+216 and Ton 599. For the two sources, the constraints on the maximum neutrino energy are loosened to $E_\nu^{\rm ob} \sim 5$ EeV and 700 PeV, respectively, and the required proton luminosities are increased by a factor of $\sim$10 and $\sim$2, or within a few times the Eddington luminosity for both sources.

4.10 Contributors to this Work

The work presented in this chapter is summarized from the paper "Variability and Spectral Characteristics of Three Flaring Gamma-ray Quasars Observed by VERITAS and *Fermi*-LAT", of which the author is a co-corresponding author (Brill et al., 2021, in prep. for submission to ApJ). This paper represents a joint effort of members of the VERITAS and *Fermi*-LAT collaborations. The author led the paper and wrote the bulk of the text, and performed the VERITAS analysis (Section 4.2.1), study of the *Fermi*-LAT flux distributions (Section 4.3), flare selection (Section 4.4), analysis of flare asymmetry and fluence (parts of Section 4.5), and study of neutrino emission during VHE flares (Section 4.9).

Janeth Valverde performed the *Fermi*-LAT analysis (Section 4.2.2) and generated the sub-daily *Fermi*-LAT light curves (parts of Section 4.5 and Appendix A) and *Fermi*-LAT spectra (Section 4.6 and Appendix B). Olivier Hervet performed the SED modeling (Section 4.7) and analysis of the constraints on the VHE emission region (Section 4.8). Qi Feng performed the *Swift*-XRT and *Swift*-UVOT analysis (Sections 4.2.3 and 4.2.4). These collaborators contributed to the text and figures, particularly in the sections noted.

Chapter 5: Prospects for a Measurement of the Luminosity Function of TeV Gamma-ray Blazars with VERITAS

Now that we have closely examined three individual blazars, we move on to considering these objects as a population. The properties of TeV blazars as a population are still poorly understood, largely due to observational biases intrinsic to the operation of IACTs. Because of their low integration time and narrow field of view, IACTs have not generally performed blind surveys, and as IACT observations are often triggered by flaring states of variable objects, reported TeV fluxes may not accurately represent blazars' true emission over time. These biases pose challenges for conducting population studies with IACTs. For this study, we focus on BL Lac objects, which make up the bulk of the steady extragalactic gamma-ray sources detected at TeV energies.

In this chapter, we develop methods to address these challenges, and apply them to evaluate the prospects for measuring the luminosity function (LF) of TeV-emitting HBLs with VERITAS. In Section 5.1, we review existing measurements of gamma-ray blazar luminosity functions at sub-TeV energies, and in Section 5.2, we discuss the scientific motivations for extending these measurements to the TeV energy range. In Section 5.3, we review the mathematical framework and some commonly used parametric models for calculating the LF. In Section 5.4, we discuss the challenges faced by IACTs when conducting population studies, including observing biases and the impact of variability. To deal with these challenges, in Section 5.5, we propose methods to model observing biases in archival data from IACTs and to correct them by relating the IACT observations to the unbiased measurements available in the GeV band from *Fermi*-LAT. Next, in Section 5.6, we develop a procedure to select

sources for a TeV survey using a multiwavelength catalog and investigate how variability can influence source selection. Drawing on these methods and frameworks, VERITAS is currently conducting a large-scale observing program to measure the LF of TeV-emitting HBLs which will combine new observations with over a decade of archival data. In Section 5.7, we describe this program and estimate using simulated data how well VERITAS can expect to resolve the TeV blazar LF.

5.1 Blazar Luminosity Functions

Blazar population studies reveal how these objects are distributed in space and evolve over cosmic time. A complete, flux-limited survey in a given waveband provides the flux and possibly other source properties such as the spectral index for all sources in the survey region down to a limiting flux determined by the sensitivity of the survey. Using the redshifts of the surveyed objects as a distance measure, the space density of sources can be derived given a cosmological model. The LF describes the number of sources per unit comoving volume per unit luminosity, and may also be a function of redshift or other observable source properties (see e.g. Kembhavi and Narlikar, 1999). If the LF increases with redshift, so that there were more sources in the past, the population is said to exhibit positive evolution; the opposite indicates negative evolution.

Extensive measurements of the blazar LF have been performed in the radio band (e.g. Dunlop and Peacock, 1990; Mao et al., 2017) and in the soft and hard X-ray bands (e.g. Beckmann et al., 2003; Ueda et al., 2003). These studies have consistently found that FSRQs exhibit positive evolution up to a potentially luminosity-dependent redshift cutoff. On the other hand, evidence has been found suggesting positive as well as negative evolution for BL Lac objects, with possible factors contributing to this mixed picture being artifacts caused by the substantial redshift incompleteness affecting this population and confusion of different

source classes such as HBLs and LBLs.

In the gamma-ray band, the blazar LF and evolution were first directly measured using data from EGRET by Chiang and Mukherjee (1998), although earlier studies had been conducted that assumed a direct scaling between the radio and gamma-ray luminosity functions (e.g. Stecker and Salamon, 1996). Bhattacharya, Sreekumar, and Mukherjee (2009) further considered the evolution of BL Lac objects and FSRQs separately, finding evidence for (positive) evolution of FSRQs but none for BL Lac objects.

The larger sample of GeV blazars available in the *Fermi* era has allowed for a better determination of the GeV blazar LF. Ajello et al. (2012) measured the LF of FSRQs detected by *Fermi*-LAT, finding that it is well-described by a broken power-law model exhibiting luminosity-dependent density evolution (LDDE), from which the intrinsic de-beamed LF could be recovered. In the LDDE model, the FSRQ population evolves positively up to a redshift peak that increases with luminosity, so that the brightest sources appeared at the earliest times. Ajello et al. (2014) performed a similar study for *Fermi* BL Lac objects, showing that their behavior differs by spectral class, with the low-luminosity HBLs exhibiting negative evolution and the IBLs and LBLs behaving similarly to FSRQs. Intriguingly, the number density of HBLs begins to increase at the same epoch as that of FSRQs decreases. To calculate the LF for a large, effectively redshift-complete BL Lac sample even though many objects lacked a spectroscopic redshift, Ajello et al. (2014) obtained a robust statistical estimate of the redshift distribution that incorporated a variety of measured redshift constraints. Building on this work, Ajello et al. (2015) calculated an overall *Fermi* blazar LF, combining FSRQs and BL Lac objects, to get their total contribution to the extragalactic gamma-ray background in order to place limits on the dark matter annihilation cross section.

However, the TeV blazar LF is not well understood. For reasons that will be discussed below, existing IACT observatories have not performed any large, uniform surveys of the extragalactic sky. CTA will perform an extragalactic survey covering a quarter of the sky to

support making a determination of the TeV blazar LF, but it will be many years until this survey is completed (CTA Consortium, 2019). As discussed in Chapter 2, the majority of blazars detected at TeV energies are BL Lac objects, mostly HBLs. Using a sample of 23 TeV-detected HBLs and IBLs observed for a variety of reasons under different conditions, Broderick, P. Chang, and Pfrommer (2012) estimated the LF of TeV blazars at low redshifts, finding that it was well described by the bolometric quasar luminosity function of Hopkins, Richards, and Hernquist (2007) shifted to lower luminosities and number densities. Using a slightly larger sample of 28 objects, they inferred an empirical flux limit defining the observations. However, the conclusions that can be drawn from this LF estimate are limited, given that it was derived from a highly heterogeneous sample affected by multiple selection effects and observational biases (Section 5.4), for which only very rough and incomplete corrections were performed.

5.2 Scientific Motivations for a TeV Blazar Luminosity Function

5.2.1 Extragalactic Radiation Fields

The total flux expected from the entire source population can be obtained by integrating over the LF. The contribution made by blazars to the diffuse extragalactic gamma-ray background (EGRB) at TeV energies can then be estimated by subtracting the flux from resolved sources. Measuring the LF for different source classes such as FSRQs and BL Lac objects allows these contributions to be resolved separately.

In addition, we can use the LF as a source-independent probe of the IGMF (Section 2.4.2). Blazar emission above ∼50-100 GeV can pair produce and generate secondary cascade emission, which in the presence of a strong IGMF, would become isotropized and contribute to the MeV-GeV EGRB (Coppi and F. A. Aharonian, 1997). We can convolve the TeV blazar LF with an EBL absorption model to get the total emission fed into the intergalactic medium

in gamma rays as a function of redshift. From this, the total cascade emission at 10-100 GeV can be calculated and compared to the *Fermi* EGRB, giving a source-independent method to constrain the IGMF. Understanding these radiation fields helps explain the diffuse backgrounds detected by *Fermi* and may constrain hypothesized contributions from unresolved astrophysical source populations and dark matter.

5.2.2 Relationships Among Astrophysical Source Populations

Blazars may contribute to the multi-TeV neutrino flux detected by IceCube. Direct evidence for this link comes from the spatial and sometimes temporal coincidence between neutrino and gamma-ray emission from the gamma-ray blazar TXS 0506+056 (M. G. Aartsen et al., 2018). The TeV blazar LF can supplement studies of individual sources by providing a source-independent, population-based measurement with which to estimate the extent to which blazar jets produce high-energy neutrinos. The contribution of blazars to the TeV EGRB, discussed above, directly constrains the analog of the diffuse neutrino flux measured by IceCube. As shown in Figure 5.1, the energy density of this contribution can be extrapolated and compared to that of the diffuse high-energy neutrino flux (Ahlers, 2019). The paucity of point sources in IceCube data suggests that numerous, faint misaligned AGN may be more likely sources than rare, bright blazars to produce the bulk of the diffuse IceCube emission (Murase and Waxman, 2016; Hooper, Linden, and Vieregg, 2019). A measurement of the TeV blazar LF could confirm or falsify this prediction.

Another outstanding puzzle in AGN physics is the nature of the connection, if any, between FSRQs and BL Lac objects. These objects may have a genetic relationship in which FSRQs transition into BL Lac objects as their accretion rate slows down (Cavaliere and D'Elia, 2002; Böttcher and Dermer, 2002). This scenario predicts an inverse evolutionary relationship between FSRQs and BL Lac objects, with BL Lac objects undergoing negative evolution at late times. By measuring the LF of TeV BL Lac objects, we can investigate

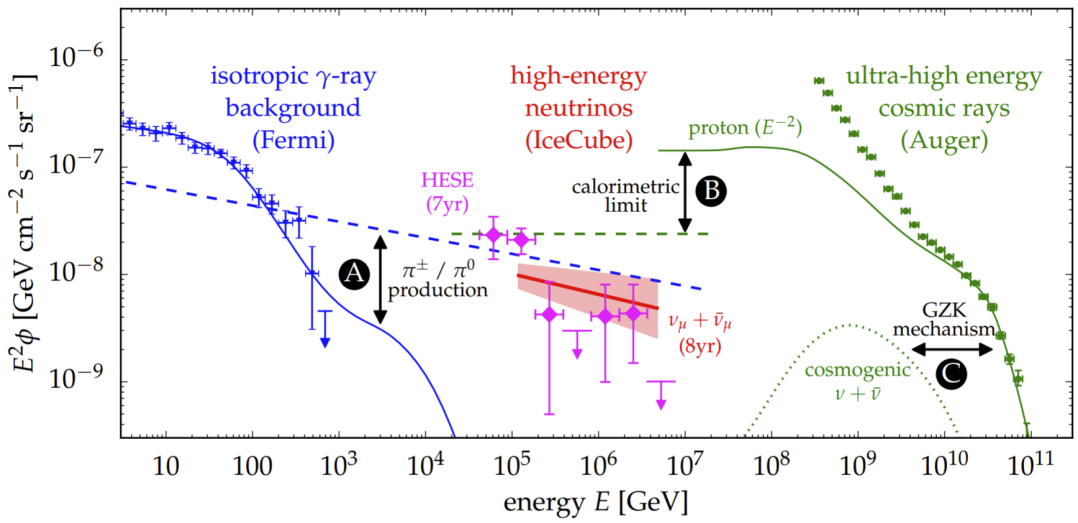


Figure 5.1: The diffuse astrophysical fluxes observed with three cosmic messengers, taken from Ahlers (2019). The comparable energy densities suggest a common origin. Gamma rays and neutrinos are linked by the common production mechanism of pion decay, marked in the figure as relationship A, with relationships B and C showing connections between other messengers.

the existence of this pattern of evolution, although the redshift range that can be effectively probed is limited by EBL absorption.

5.2.3 Blazar Physics

The shape of the LF depends on the physical processes that produce TeV gamma-ray emission. Relativistic beaming amplifies or reduces the apparent luminosity, depending on the orientation of the jet with respect to the observer. The observed LF is therefore spread over a wider range of luminosities than the intrinsic one. We can recover information on jet physics in blazars by fitting convolved beaming and intrinsic LF models to the observed LF (Urry and Shafer, 1984; Urry and Padovani, 1991; Padovani and Urry, 1992). The intrinsic LF in turn informs our understanding of the parent population of relativistic particles powering the emission in these sources (Mücke and Pohl, 2000).

Formally, following Urry and Shafer (1984), the dependence of the observed LF $\Phi_{\rm obs}(L)$

on the intrinsic LF $\Phi_{\rm int}(\mathcal{L})$ can be written as

$$\Phi_{\rm obs}(L) = \int P(L \mid \mathcal{L}) \Phi_{\rm int}(\mathcal{L})\, d\mathcal{L}. \tag{5.1}$$

Recalling Equations 2.22 and 2.23, for a given Lorentz factor Γ, the observed luminosity depends solely on the viewing angle θ, so we can write

$$P(L \mid \mathcal{L})\, dL = P(\theta)\, d\theta = \sin\theta\, d\theta, \tag{5.2}$$

for an isotropic distribution of viewing angles. From Equations 2.22, 2.23, and 5.2, we then obtain

$$P(L \mid \mathcal{L}) = \frac{1}{\beta\gamma p} \mathcal{L}^{\frac{1}{p}} L^{-\frac{p+1}{p}}. \tag{5.3}$$

In the simplest realistic case, we can consider an intrinsic LF distributed as a power law,

$$\Phi_{\rm int}(\mathcal{L}) = K\mathcal{L}^{-q}; q > 1, \mathcal{L}_{\rm min} < \mathcal{L} < \infty. \tag{5.4}$$

A given observed luminosity L can only result from intrinsic luminosities in the range $\frac{1}{2\Gamma}\mathcal{L}_{\rm min} \leq \mathcal{L} \lesssim 2\Gamma\mathcal{L}_{\rm min}$, for viewing angles $\theta \in [0°, 180°]$. Because of this constraint, the integral in Eq. 5.1 has a piecewise form (Urry and Shafer, 1984):

$$\Phi_{\rm obs}(L) = \frac{K}{\beta\gamma pC} \begin{cases} 0, & L < L_{\rm min} \\ \mathcal{L}_{\rm min}^{-C} L^{-\frac{p+1}{p}}, & L_{\rm min} \leq L < L_{\rm br} \\ (2\Gamma)^{pC} L^{-q}, & L \geq L_{\rm br} \end{cases} \tag{5.5}$$

where $C = q - (1/p) - 1$, $L_{\rm min} = (2\Gamma)^{-p}\mathcal{L}_{\rm min}$, and $L_{\rm br} = (2\Gamma)^{p}\mathcal{L}_{\rm min}$. Above the break, the slope of the beamed LF reproduces that of the intrinsic LF. In the more complicated case

in which the Lorentz factors are distributed as a power law, the beamed LF can still be determined analytically for an isolated relativistic jet (Urry and Padovani, 1991).

5.3 Calculating the Luminosity Function

It is useful to consider two special cases of luminosity functions. In pure density evolution, the space density of the population evolves identically for all source luminosities. We can then separate the LF into two parts

$$\Phi(L, z) = \Phi_0(L)\rho(z), \tag{5.6}$$

where $\Phi_0(L)$ is the local LF at $z = 0$ and $\rho(z)$ is the evolution function. In pure luminosity evolution, on the other hand, the form of the LF stays constant but the luminosities change over time. In this case, the LF can be written as

$$\Phi(L, z) = \Phi_0(L/\rho(z)), \tag{5.7}$$

where $\rho(0) = 1$. It should be recognized that the evolution function describes statistical changes in the blazar population over cosmic time, not changes to individual objects. If the LF has the form of an unbroken power law, it can be shown that these two special cases are equivalent. The evolution function $\rho(z)$ can have any form, with one simple representation being

$$\rho(z) = (1 + z)^k. \tag{5.8}$$

For more general patterns of evolution (such as LDDE), the LF can always be written as

$$\Phi(L, z) = \Phi_0(L)\rho(z, L). \tag{5.9}$$

5.3.1 Luminosity-Volume Test for Evolution

The evolution of a complete, flux-limited sample can be measured without the need to represent the LF with any particular functional form by means of the Luminosity-Volume, or $V/V_{\rm max}$, test (Schmidt, 1968). To describe the test, we first consider a Euclidean universe for simplicity, in which a source's luminosity is determined from its measured flux F and distance r by $L = 4\pi r^2 F$. Then, for an all-sky survey with limiting flux $F_{\rm lim}$, each source has two corresponding volumes, $V = \frac{4\pi}{3} r^3$ and $V_{\rm max} = \frac{4\pi}{3} r_{\rm max}^3$, where $r_{\rm max} = \sqrt{L/(4\pi F_{\rm lim})}$ is the farthest distance at which the source could have been and still be included in the sample. More complicated scenarios are allowed, such as the coherent combination of separate survey regions with different limiting fluxes, by taking V and $V_{\rm max}$ to refer in general to the total volumes enclosed by and accessible to the source (Avni and Bahcall, 1980). To model an actual survey covering only part of the sky or having some estimated incompleteness, the calculated volume may include a sky coverage factor $0 \leq \Omega(L, z) \leq 4\pi$. If there are multiple limiting fluxes defining the sample in the same survey region, such as in a multiwavelength survey, the most constraining one is used.

For an expanding universe, we define the luminosity distance $D_L(z)$ such that $L = 4\pi D_L^2(z) F$. In this case, the accessible volume varies with redshift such that $V = V(z)$ and $V_{\rm max} = V(z_{\rm max})$, with $z_{\rm max}$ being the maximum redshift at which the source could have been detected. Then, if the sources are uniformly distributed, the mean value of $V/V_{\rm max}$ is

$$\langle V/V_{\rm max} \rangle = \frac{\int_{L_{\rm min}}^{L_{\rm max}} dL \int_0^{z_{\rm max}(L)} (V/V_{\rm max}) \Phi(L, z)\, dV(z)}{\int_{L_{\rm min}}^{L_{\rm max}} dL \int_0^{z_{\rm max}(L)} \Phi(L, z)\, dV(z)} = \frac{1}{2}, \tag{5.10}$$

when the luminosity function $\Phi = \Phi(L)$ is independent of z, i.e., there is no evolution.

Assuming that the sample is complete, a value of $\langle V/V_{\rm max} \rangle$ greater than 1/2 indicates positive evolution, with the opposite indicating negative evolution. The uncertainty on the $\langle V/V_{\rm max} \rangle$ statistic is $1/\sqrt{12N}$, where N is the number of sources in the sample (e.g. Peterson,

1997). The $V/V_{\rm max}$ test measures the presence and direction of evolution, but does not give any information on its specific functional form.

In the TeV gamma-ray band we have the additional complication of EBL absorption (Section 2.4.1), which must be included when inferring the luminosity from the measured flux and redshift:

$$L = 4\pi D_L^2(z) F e^{\tau(z)}. \tag{5.11}$$

One way to see that Eq. 5.10 still holds without modification is to define

$$D_{\rm EBL}^2(z_{\rm max}) \equiv D_L^2(z_{\rm max}) e^{\tau(z_{\rm max})} = \frac{L}{4\pi F}, \tag{5.12}$$

which we can solve numerically to obtain $z_{\rm max}(L)$.

5.3.2 Luminosity Function using Accessible Volumes

A similar approach to the $V/V_{\rm max}$ test can be used to obtain an unbiased non-parametric estimate of the LF (Schmidt, 1968). In this method, called the method of accessible volumes, the value of the LF in each bin $L_i \in (L \pm \Delta L/2)$ is given by

$$\Phi(L_i) = \frac{1}{\Delta L} \frac{N_i}{V_{{\rm max},i}}. \tag{5.13}$$

If evolution is present, redshift bins $z_j \in (z \pm \Delta z/2)$ may be used as well, in which case the accessible volumes are the shells enclosed within each redshift bin. Alternatively, evolution can be accounted for in the local ($z = 0$) LF by weighting $V_{\rm max}$ using an assumed evolution function (Schmidt and Green, 1983)

$$V_{\rm max} = \int_{z_{\rm min}}^{z_{\rm max}} \Omega(L_i, z) \frac{\rho(z, L_i)}{\rho(z_{\rm min}, L_i)} \frac{dV}{dz}\, dz, \tag{5.14}$$

where $\rho(z, L)$ is the evolution function as defined in Eq. 5.9 and dV/dz is the differential comoving volume per unit solid angle (e.g. Hogg, 1999).

5.3.3 Luminosity Function using Maximum Likelihood

Another way to calculate the LF is to fit a parametric model by maximizing the expected likelihood over the observed data (Marshall et al., 1983). This method can be used to compare analytical representations of the LF and evolution functions, and is particularly useful for smaller samples which cannot be finely binned in luminosity and redshift.

The simplest useful LF model is a power law,

$$\Phi(L, z) = \frac{A}{(\ln 10)L} \left(\frac{L}{L_{\rm scale}}\right)^{-\gamma}, \tag{5.15}$$

where A is the normalization amplitude, $L_{\rm scale}$ is the fixed scale luminosity, and γ is the power-law index. A break in the slope, such as might be expected due to relativistic beaming, can be modeled using a smoothly broken double power law such as

$$\Phi(L, z) = \frac{A}{(\ln 10)L} \left[\left(\frac{L}{L_*}\right)^{\gamma_1} + \left(\frac{L}{L_*}\right)^{\gamma_2}\right]^{-1}, \tag{5.16}$$

where L_* is the break luminosity, and γ_1 and γ_2 are the power-law indices before and after the break, respectively. Following Marshall et al. (1983), the mean number of sources in the differential element $dLdz$ is given by

$$n(L, z)\, dLdz = \frac{\Omega(L, z)}{4\pi} \Phi(L, z)\, dLdV(z), \tag{5.17}$$

where $\Omega(L, z)$ is the sky coverage (Section 5.3.1). The expected luminosity and redshift distributions dN/dL and dN/dz can be obtained by marginalizing over $n(L, z)$. Assuming the sources are distributed independently, the observed number of sources in each element

follows a Poisson distribution and is, for sufficiently small bins, either zero or one. The likelihood function is then given by joint Poisson probabilities:

$$\Gamma = \prod_i e^{-n(L_i,z_i)\,dLdz} n(L_i, z_i)\,dLdz \prod_j e^{-n(L_j,z_j)\,dLdz}, \tag{5.18}$$

where the first factor runs over all of the sources in the sample, and the second over all of the differential elements with no sources. Introducing the log-likelihood $S = -2\ln\Gamma$ and dropping model-independent terms, we obtain

$$S = -2\sum_i \ln\Phi(L_i, z_i) + 2\int_{L_{\min}}^{L_{\max}} \int_{z_{\min}}^{z_{\max}} \Omega(L, z)\Phi(L, z)\,dLdV(z), \tag{5.19}$$

where the integration runs over the ranges of luminosity and redshift seen in the data. A parametric LF model can then be fit to the data using any optimization technique.

5.4 Challenges Facing Population Studies with IACTs

The methods in the previous section require a complete sample in the sense that every source in the survey region with a flux higher than a well-defined flux limit is included. We noted that incompleteness, if sufficiently well understood, can be handled by incorporating a sky coverage factor $\Omega(L, z)$ into the model. However, performing a population study of blazars with a pointed instrument such as an IACT array is especially challenging and subject to multiple potential observing biases that do not affect all-sky instruments.

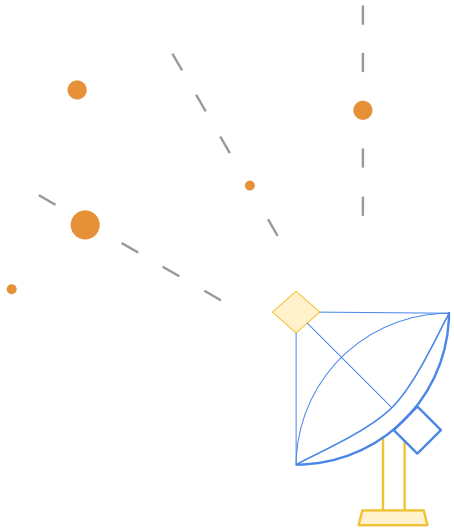
5.4.1 Observing Biases with Current Instruments

The typical observation strategy of VERITAS and other current-generation IACT arrays when observing extragalactic sources results in several biases, illustrated in Figure 5.2. First, VERITAS performs *targeted observations*, focusing on known and hypothesized TeV

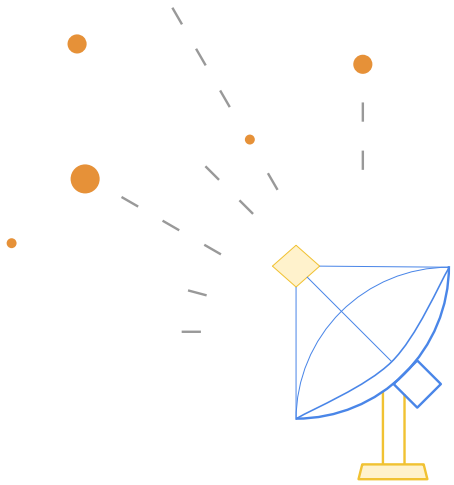
gamma-ray sources selected from GeV gamma-ray and X-ray source catalogs for a variety of scientific motivations (Benbow, 2019). These observations, while extensive, do not constitute a survey. While it is likely that TeV blazars will appear in multiwavelength catalogs, it is not guaranteed. For example, selecting *Fermi*-LAT sources as targets may miss extreme TeV blazars whose gamma-ray emission peaks in the TeV band (Tavecchio et al., 2011).

Second, the amount of data taken on each target varies, complicating any attempt to derive an implicit flux limit from the observations. These *biased exposures* result from the varying scientific objectives of the observations, as well as from variation imposed by changing weather conditions, competition for visibility windows, and seasonal differences in available observing time. Given the lack of a uniform flux limit, one might be tempted to determine a flux limit independently for each source using its actual exposure. However, such a source-specific flux limit may be misleading. For example, a deep exposure might be taken on a bright source for reasons unrelated to merely detecting it, for example, to perform regular monitoring or to resolve a detailed spectrum. The likelihood of performing such observations clearly depends on source luminosity and redshift in a way that is difficult to quantify, making any inferred accessible volume very hard to interpret. Conversely, if sources are intentionally observed until detected, but not more, they will tend to fall near the edge of their accessible volume more often that expected by chance, confounding the measurement of evolution.

In principle, these biases can be avoided by conducting a uniform survey evenly tiling a portion of the extragalactic sky. This is the strategy of the future CTA extragalactic survey, which is expected to detect 30-150 blazars in a quarter of the sky in about 1000 hours of observing time (CTA Consortium, 2019). Attempting a similar program with VERITAS, which has an order of magnitude less sensitivity, is infeasible. The survey would be either too narrow or too shallow, reducing the number of objects detected to a handful, unless observing time comparable to the entire lifetime of VERITAS were allocated.

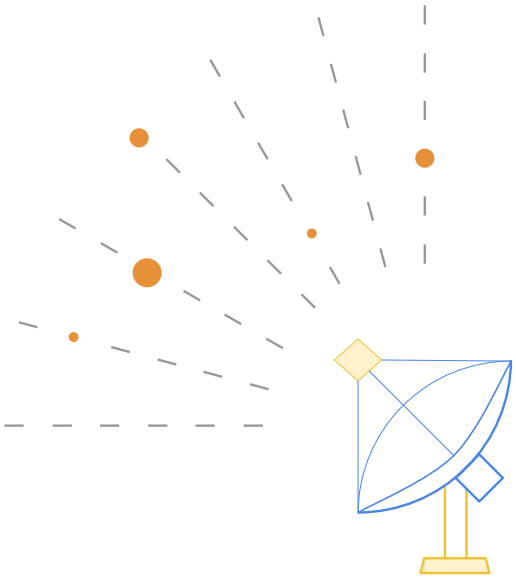


(a) Targeted Observations



(b) Biased Exposures

(c) Low Sensitivity



(d) Unbiased Survey

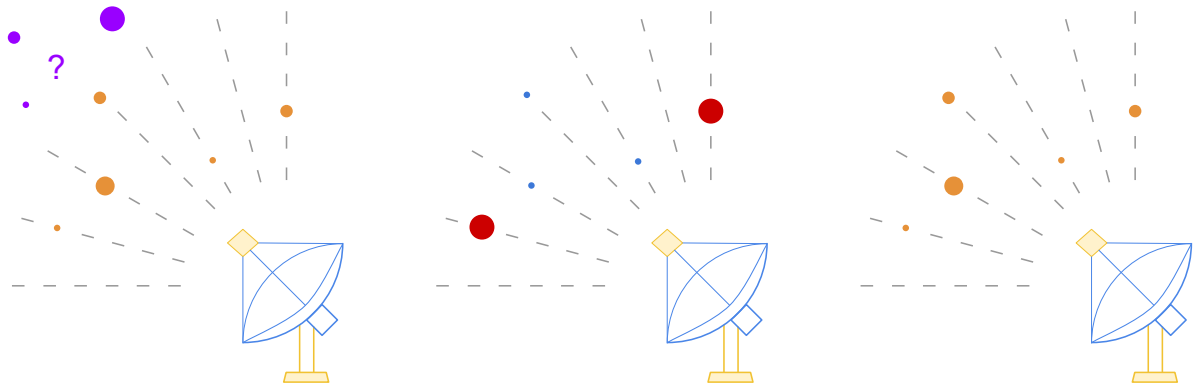
Figure 5.2: Observing biases affecting extragalactic surveys performed with VERITAS and other current-generation IACT arrays. An ideal unbiased survey is illustrated for comparison.

5.4.2 Intrinsic Properties of TeV Blazar Emission

Even with future instruments, TeV blazars have intrinsic properties that can cause observing biases if not taken into account. These observing biases are illustrated in Figure 5.3. First, EBL absorption causes a cutoff at high energies and redshifts, resulting in biased estimations of the intrinsic source luminosity and accessible volume unless properly modeled. As discussed in Sections 2.4.1 and 5.3.1, this can be done using a theoretically or empirically derived EBL model.

More significantly, blazars are highly variable at all wavelengths, further complicating the issues already discussed. A multiwavelength catalog used for target selection may omit TeV sources that by chance missed the threshold for inclusion when the catalog was created. In addition, IACT observations often occur disproportionately during flares, which are in some cases the only times in which a source can be detected, biasing the estimate of the average luminosity.

Even in a completely uniform survey, some fraction of the sources will flare by chance while the survey is being conducted, depending on the *a priori* unknown duty cycle of variability. An individual source's level of activity while being observed may greatly affect its detection probability and measured luminosity. While variability should average out on the population level, for the small to moderate samples expected in TeV blazar surveys it presents a source of systematic uncertainty. The systematic impact of blazar variability depends on the observing strategy, which can be quantified using simulated light curves (Giomi, Gerard, and Maier, 2016). If flaring and quiescent states have different physical origins, it may be desirable to calculate the LF for a particular state only.



(a) EBL Absorption (b) Variability (c) Unbiased Survey

Figure 5.3: Observing biases affecting surveys performed with any IACT array. An ideal unbiased survey is illustrated for comparison.

5.5 Modeling and Correcting Observing Biases

The best way to deal with potentially biased data is to avoid collecting it in the first place, and as described below in Section 5.7, that is what an ongoing VERITAS observing program dedicated to measuring the luminosity function is attempting to do. However, as noted above, extragalactic surveys with current-generation IACT arrays operate in a statistics-limited regime, and VERITAS has already collected thousands of hours of extragalactic observations. It is worthwhile to explore methods to model and correct for observing biases in order to greatly increase the size of the usable dataset at the cost of allowing some systematic error. First, we develop a framework in which the observing biases discussed in the previous section can be described mathematically and modeled in the likelihood function. Next, we investigate a method to correct for biased observations of variable TeV blazars by matching the TeV fluxes to flux states derived from the continuous coverage in the GeV band provided by *Fermi*-LAT.

5.5.1 Sky Coverage of a Biased Survey

The sky coverage Ω may be written as a function of luminosity, redshift, and other parameters of a source, such as the spectral index, as in the case of *Fermi*-LAT (Abdo et al., 2010b). For clarity in the discussion that follows, we suppress any potential dependence on parameters other than luminosity and redshift. For a uniform survey, we can write

$$\Omega(L, z) = \Omega_0 p_{\rm det}(L, z), \tag{5.20}$$

where Ω_0 is the solid angle covered by the survey ($\Omega_0 = 4\pi$ for an all-sky survey) and $p_{\rm det}(L, z)$ is the probability that a source of luminosity L and redshift z will be detected in the survey, which could be quantified by analyzing simulated data.

For an ideal flux-limited survey,

$$\Omega(L, z) = \Omega_0 \begin{cases} 0 & F(L, z) < F_{\rm lim} \\ 1 & F(L, z) \geq F_{\rm lim}, \end{cases} \tag{5.21}$$

where $F_{\rm lim}$ is the limiting flux. If, rather than conducting a uniform survey, targets are selected from a multiwavelength catalog, incompleteness resulting from target selection may be modeled with an additional term:

$$\Omega(L, z) = \Omega_0 p_{\rm sel}(L, z) p_{\rm det}(L, z). \tag{5.22}$$

As discussed in Section 5.6, the selection and detection probabilities can both be affected by variability. We can model the emission from a variable source by replacing the luminosity L with a probability density function (PDF) over luminosity, which we call a luminosity distribution f_L, which is directly proportional to the flux distribution. Each source's contribution to the LF is therefore spread over a range of luminosities given by its luminosity

distribution. The luminosity distribution capturing the variability of an individual source over time, which is typically no more than an order of magnitude in scale, is much narrower than the variation in luminosity from source to source, which can span at least four orders of magnitude in both the GeV band (e.g. Ajello et al., 2014) and TeV band (Section 5.7.3). Methods to parameterize f_L and estimate its value using *Fermi*-LAT data are discussed below in Sections 5.5.2 and 5.5.3. We can then write:

$$\Omega(L, z) = \Omega_0 p_{\rm sel}(f_L, z) p_{\rm det}(f_L, z). \tag{5.23}$$

It may be possible to model the impact of non-uniform, intermittent, or biased exposures given the known observation strategy and cadence. For an unbiased but discontinuous survey, the systematic error due to observation cadence can be modeled using simulated light curves (Giomi, Gerard, and Maier, 2016), but for existing data, the observation strategy may be more difficult to model. The simplest procedure is to simply exclude data known to have been taken in response to an external flare trigger. This approach is conservative, but has the drawbacks of potentially discarding the most informative high-flux data and requiring judgment to determine what exactly constitutes "biased" data. Another option, which we do not explore further, is to develop an "exposure function" giving the probability distribution of exposure on a source as a function of its flux or luminosity. This could be derived from historical VERITAS observations, although an estimate of the unknown luminosity distribution of each source would be required.

5.5.2 Likelihood Function of a Luminosity Distribution

To adapt the likelihood function to account for variability, we must replace L with f_L, include an additional PDF for each parameter of f_L, and replace $\Omega(L, z)$ with the sky coverage expression defined by Eq. 5.23. As a simple example of a model for f_L, suppose that a

blazar emits TeV gamma rays in two discrete luminosity states: quiescent, which is the predominant, low-luminosity state, and flaring, which a rare, high-luminosity state. In this model, the luminosity is equal to either the quiescent luminosity $L_{\rm q}$ or the flaring luminosity $L_{\rm f}$, such that

$$f_L(L_{\rm q}, L_{\rm f}, p_{\rm f}) = (1 - p_{\rm f})\delta(L - L_{\rm q}) + p_{\rm f}\delta(L - L_{\rm f}), \tag{5.24}$$

where $p_{\rm f}$ is the fraction of the time that the source is in the flaring state. Other models could be chosen that are more realistic and may reflect different physical pictures of blazar emission. For example, instead of discrete values, the luminosity during the two states may follow normal or log-normal distributions (e.g. Valverde et al., 2020), or the assumption of multiple discrete states may be dropped entirely, with the luminosity as a whole following a single normal, log-normal, or other distribution, such as the SDE model studied in Section 4.3.

To perform the fit, we would also need a PDF to model the intrinsic distribution of each parameter of f_L. Considering without loss of generality the simple three-parameter luminosity distribution proposed in Eq. 5.24, we could assume, for example:

$$\begin{aligned} L_{\rm q} &\sim f_{L_{\rm q}} = \text{LogNorm}(\mu_{L_{\rm q}}, \sigma_{L_{\rm q}}) \\ L_{\rm f} &\sim f_{L_{\rm f}} = \text{LogNorm}(\mu_{L_{\rm f}}, \sigma_{L_{\rm f}}) \\ p_{\rm f} &\sim f_{p_{\rm f}} = \text{Norm}(\mu_{p_{\rm f}}, \sigma_{p_{\rm f}}). \end{aligned} \tag{5.25}$$

The parameter distributions $f_{L_{\rm q}}$, $f_{L_{\rm f}}$, and $f_{p_{\rm f}}$ in Eq. 5.25 are independent, but this assumption is not required, so that the flare probability may depend on the flare luminosity, for example. In any case, the joint PDF is then

$$f_N(L_{\rm q}, L_{\rm f}, p_{\rm f}, z) = \Phi(L, z) f_L f_{L_{\rm q}} f_{L_{\rm f}} f_{p_{\rm f}} \frac{dV}{dz}, \tag{5.26}$$

and we can now adapt the standard expression for the number of blazars in the differential element $dLdz$ (Eq. 5.17) to account for variability:

$$n(L_{\rm q}, L_{\rm f}, p_{\rm f}, z) dL_{\rm q} dL_{\rm f} dp_{\rm f} dz = \frac{\Omega(f_L, z)}{4\pi} f_N dL_{\rm q} dL_{\rm f} dp_{\rm f} dz. \tag{5.27}$$

Following the logic of Eqs. 5.18 and 5.19, we obtain:

$$\begin{aligned} S = & -2 \sum_i \ln f_N(L_{\rm q}, L_{\rm f}, p_{\rm f}, z) \\ & + 2 \int_{L_{\rm q,min}}^{L_{\rm q,max}} \int_{L_{\rm f,min}}^{L_{\rm f,max}} \int_{p_{\rm f,min}}^{p_{\rm f,max}} \int_{z_{\rm min}}^{z_{\rm max}} n(L_{\rm q}, L_{\rm f}, p_{\rm f}, z) dL_{\rm q} dL_{\rm f} dp_{\rm f} dz. \end{aligned} \tag{5.28}$$

In practice, some of these parameters may be poorly constrained. In particular, $p_{\rm f}$ cannot easily be obtained from irregularly sampled TeV data; for weaker sources, $L_{\rm q}$ may only be constrained by upper limits; and since observations of rare flares may be difficult to acquire (or excluded from the dataset), values of $L_{\rm f}$ may be missing as well. The distribution $f_{p_{\rm f}}$ would need to be estimated using unbiased, fairly complete gamma-ray data, such as that reported by FACT (Dorner et al., 2019) or *Fermi*-LAT, perhaps extrapolating from a subset of particularly bright, well-sampled sources. This estimation would be similar to the study reported below in Section 5.5.3.

Using a maximum likelihood approach allows us to deal with missing values of $L_{\rm q}$ and $L_{\rm f}$ in a manner similar to the method developed by Ajello et al. (2014) to handle missing redshifts in their determination of the LF of *Fermi*-detected BL Lac objects. Of the 211 objects in their sample, only 103 had a spectroscopic redshift measurement, with all but five of the remaining sources having some redshift constraints. Ajello et al. (2014) used an iterative approach to incorporate these constraints, which we can adapt for our purpose in the following manner. Suppose some of the blazars in our sample lack a well-measured $L_{\rm q}$

or $L_{\rm f}$, but may have constraints placed on these values. Start by assuming prior PDFs of the observed values, $f_{L_{\rm q},\rm obs}$ and $f_{L_{\rm f},\rm obs}$. Then the PDFs specific to each source are

$$\mathrm{PDF}(L_s) = f_{L_s,\rm obs} \times \prod_i^n C_i(L_s), \tag{5.29}$$

for $s \in \{\mathrm{q}, \mathrm{f}\}$, where the $C_i(L_s)$ are the constraints on the relevant luminosity parameters. Typically, each state would have at most one constraint, an upper limit, which would have the form of a step function. One can then proceed by drawing redshifts for each source from these PDFs; computing the LF using the sampled redshifts; calculating the predicted $f_{L_{\rm q},\rm obs}$ and $f_{L_{\rm f},\rm obs}$ from the LF; replacing the prior PDFs with the predicted distributions; and repeating this entire process until the prior PDFs converge. The predicted luminosity parameter distributions $f_{L_{\rm q},\rm obs}$ and $f_{L_{\rm f},\rm obs}$ can be calculated by marginalizing over the LF:

$$\begin{aligned} f_{L_{\rm q},\rm obs} &= \int_{L_{\rm f,min}}^{L_{\rm f,max}} \int_{p_{\rm f,min}}^{p_{\rm f,max}} \int_{z_{\rm min}}^{z_{\rm max}} n(L_{\rm q}, L_{\rm f}, p_{\rm f}, z)\, dL_{\rm f} dp_{\rm f} dz \\ f_{L_{\rm f},\rm obs} &= \int_{L_{\rm q,min}}^{L_{\rm q,max}} \int_{p_{\rm f,min}}^{p_{\rm f,max}} \int_{z_{\rm min}}^{z_{\rm max}} n(L_{\rm q}, L_{\rm f}, p_{\rm f}, z)\, dL_{\rm q} dp_{\rm f} dz. \end{aligned} \tag{5.30}$$

5.5.3 Estimating the Luminosity Distribution with *Fermi*-LAT

If we knew the true TeV flux distributions of our sources, we could correct for biases in observation timing and duration. Estimating the flux distribution is also a prerequisite for applying the likelihood method developed in the previous section. One possible way to do this is to extrapolate from the flux distribution observed in a different waveband with continuous sky coverage, such as GeV gamma rays using data from *Fermi*-LAT, taking advantage of long-term correlations between wavebands (e.g. Valverde et al., 2020). For this method to work, the GeV and TeV fluxes of the sources under consideration (such as HBLs) must be well-correlated in their overall distributions and in the timing of flux states at timescales

down to the typical cadence of observations (days to months, depending on the source). This assumption is motivated by considering that the GeV and TeV emission comes from a single component such that the emission is well-correlated on the days-weeks timescales resolvable by *Fermi*, justified, for example, by the compatible power spectral densities measured in this frequency range (e.g. H.E.S.S. Collaboration et al., 2017). This method is only applicable for observations taken after October 2008, when *Fermi* began collecting science data.

To investigate this approach, we modeled the TeV flux distribution by determining *Fermi*-LAT flux states using the Bayesian blocks algorithm (Scargle, 1998; Scargle et al., 2013) and extrapolating them into the TeV band. By weighting TeV data by block duration, runs during short flares can be weighted less in the average flux than those during long low states. This approach maximizes the use of VERITAS data and automatically nullifies biases in the timing of data collection. While this study considered the time-averaged flux, this method could also be used to fit a parametric flux distribution model or allow high and low states across sources to be considered separately. It also provides an implicit estimate of the duty cycle of flaring TeV blazars, a key input when estimating the rate of time-dependent correlations between neutrino emission and gamma-ray flares (e.g. Palladino et al., 2019). However, while this method is intended to provide a better estimate of the true average flux of a detected TeV source, it does not address the effect of biased exposures in determining whether a source is detected at all.

VERITAS performs nightly observations of the bright, nearby ($z = 0.03$) HBL Markarian 421 (Mrk 421), making this source an excellent test case for understanding how well GeV flux states provided by *Fermi*-LAT can be used as a proxy for TeV variability. For this study, a preliminary VERITAS light curve of Mrk 421 was used that covered the period from May 6, 2007 to June 27, 2019, containing about 400 daily bins (Park and Cervantes, 2019). The data include observations during flares as well as regular monitoring, which was performed nightly starting in Fall 2018. At each bin, the integral flux above 200 GeV was calculated

with the spectrum fixed to the global average spectrum, with only the normalization left free. The light curve was not calibrated to take into account long-term changes in the optical response and detector performance of VERITAS, and should be considered preliminary (Nievas Rosillo, 2021).

To get the *Fermi*-LAT light curve of Mrk 421, we performed a likelihood analysis using the LAT `Fermitools` 1.2.1 and `P8R3_SOURCE_V2` IRFs with the `FermiPy` 0.17.4 analysis package (Wood et al., 2017). The data were restricted to the period from October 27, 2008 to August 1, 2019. The energy range of 1-500 GeV was considered in order to maximize the energy overlap with VERITAS while retaining sufficient statistical significance to support short time bins. Photons with zenith angle greater than 90° were excluded to reduce contributions from the Earth's limb. For each source, the region of interest considered was a square of width 15° surrounding the catalog source position. The background model consisted of all sources in the 4FGL catalog (Abdollahi et al., 2020) within a 30° box centered on the source. The gamma-ray emission from Mrk 421 was fit using a log-parabola spectral model, which was the best-fit model in the 4FGL catalog.

A *Fermi*-LAT light curve of Mrk 421 was generated with a bin size of three days, which was chosen to achieve a sufficiently fine time resolution without incurring excessive computational cost. In each time bin, the parameters of the source model were freed while the background model was kept fixed to the best-fit model of the baseline analysis. Out of 1312 light curve bins, 1267 (97%) were detected with a test statistic (TS) greater than 16. `FermiPy` 0.17.4 was found to crash if any time bin contained no events, which can occasionally occur when using time bins shorter than about a month, depending on the observing mode of the *Fermi* spacecraft. To deal with this issue, we developed a pipeline[1] around `FermiPy` to split the light curve analysis into sections, clean up and restart the analysis when an error occurs, and reduce the incidence of errors by automatically skipping time intervals with no events.

[1] `https://github.com/aribrill/fermipipe`

The light curve was split piecewise into segments of approximately constant flux using the Bayesian blocks algorithm. The prior parameter p_0, which is used to determine the significance threshold needed to create a change point splitting two blocks, was set to the value equivalent to $5\,\sigma$ using Equation 13 of Scargle et al. (2013). The average flux was calculated separately within each block. The three-day binned *Fermi*-LAT light curve of Mrk 421 overlaid with the Bayesian blocks is shown in Figure 5.4.

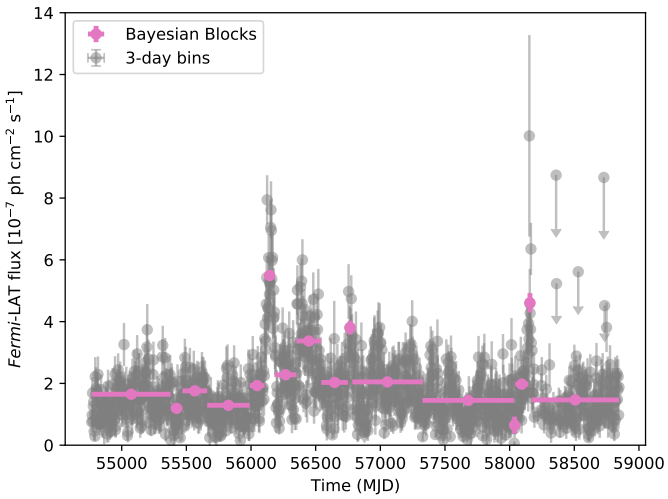


Figure 5.4: *Fermi*-LAT light curve of Mrk 421 with three-day time bins. Bayesian blocks are shown by pink lines.

The VERITAS flux corresponding to each block was calculated by computing the average of the VERITAS flux points within that block, weighted by the reciprocal of the squared error on the flux. A VERITAS flux was available for 13 of the 16 blocks, covering 95.7% of the total time period analyzed with *Fermi*-LAT. Using the 13 blocks with VERITAS data only, a scaling factor of 0.0008 between the VERITAS and LAT fluxes was estimated by taking the average of the flux ratios between the two instruments, weighted by the reciprocal of the squared relative error on the LAT flux. A comparison between the *Fermi*-LAT and VERITAS fluxes during these blocks is shown in Figure 5.5. The overall trend appears

roughly linear, with a dynamic range of about one order of magnitude for both VERITAS and *Fermi*-LAT.

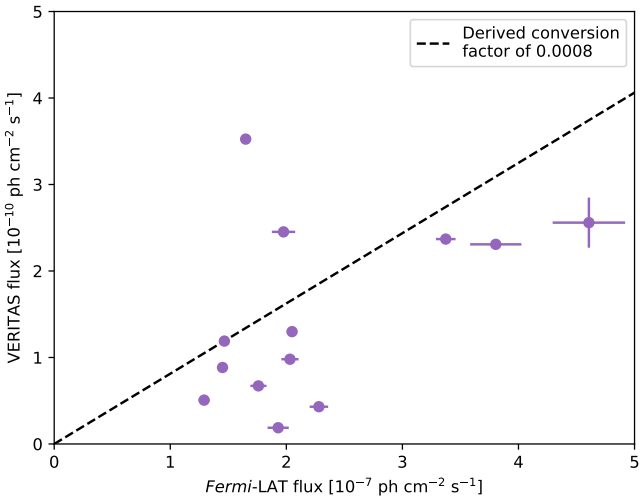
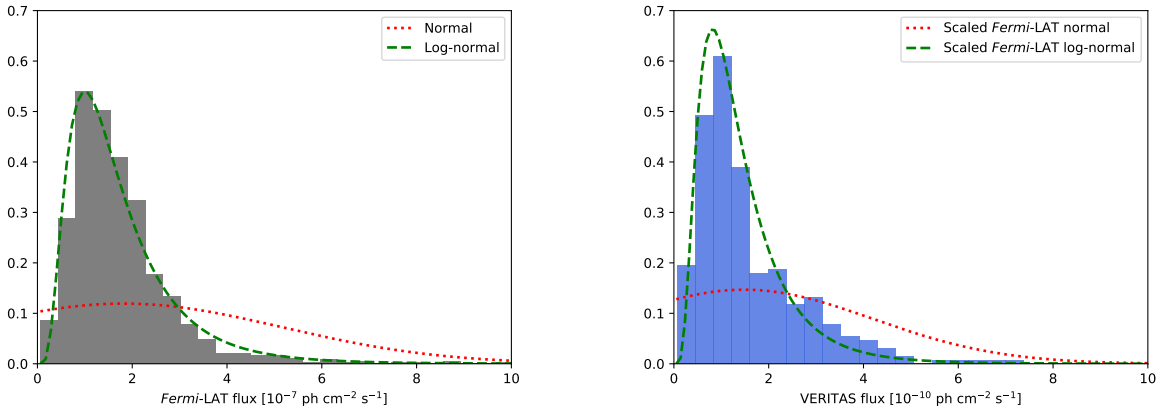


Figure 5.5: Comparison between the VERITAS and *Fermi*-LAT fluxes during Bayesian blocks defined by *Fermi*-LAT. A conversion factor of 0.0008 derived from the data points is shown with a dashed black line. The error bars on the VERITAS points are the standard error on the average flux determined using the observations within each block.

The *Fermi*-LAT and VERITAS flux distributions are compared in Figure 5.6. The VERITAS data points prior to the start of *Fermi* observations have been excluded. A log-normal model approximates the *Fermi*-LAT flux distribution well, with the best-fit normal distribution also shown for comparison. The same models with their parameters scaled by the conversion factor of 0.0008 are overplotted on the VERITAS flux distribution. The scaled log-normal model gives a good apparent fit to the data, although the flux density at low fluxes may be slightly underpredicted relative to that at high fluxes. The average flux of VERITAS within each block and the flux estimated by scaling the *Fermi*-LAT flux by the conversion factor are overplotted on the VERITAS light curve of Mrk 421 in Figure 5.7.

We can use these results to make preliminary estimates of the error induced by weighting observed VERITAS fluxes by a flux distribution estimated from *Fermi*-LAT and using scaled



(a) *Fermi*-LAT (b) VERITAS

Figure 5.6: *Left*: *Fermi*-LAT flux distribution of Mrk 421. The best-fit normal and log-normal models are shown with a dotted red and dashed green line, respectively. *Right*: VERITAS flux distribution of Mrk 421. The models from the left panel, scaled by a factor of 0.0008, are overplotted.

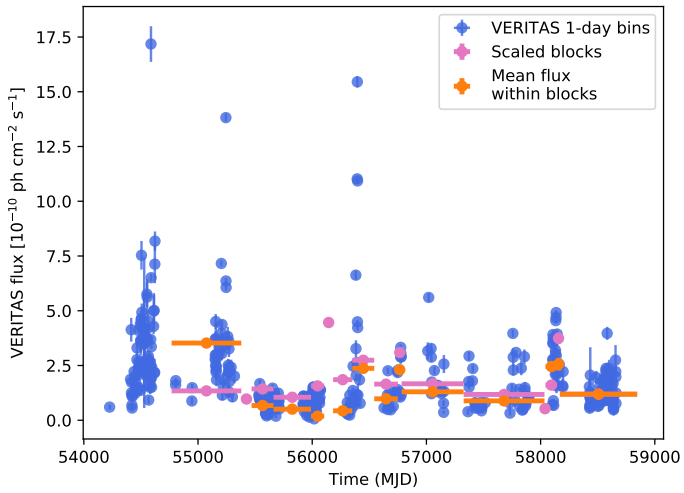


Figure 5.7: VERITAS light curve of Mrk 421 overlaid with the VERITAS fluxes averaged within the *Fermi*-LAT Bayesian blocks (orange) and the LAT fluxes scaled using a linear conversion factor (pink).

GeV observations to fill in gaps in TeV coverage. Considering only the time period of overlap with *Fermi*, the average flux of the one-day VERITAS bins is 1.72×10^{-10} ph cm^{-2} s^{-1}. To counteract biased exposures, one can weight the flux of each VERITAS observation by the duration of its corresponding block, divided by the number of observations falling in that block. Doing so in this case has a minimal effect, slightly increasing the mean daily flux to 1.75×10^{-10} ph cm^{-2} s^{-1}, suggesting that flare observations and low-state monitoring occur roughly proportionately in this dataset. Estimating the mean VERITAS flux by scaling the *Fermi*-LAT flux by the conversion factor yields 1.48×10^{-10} ph cm^{-2} s^{-1}, 14% lower than the mean VERITAS flux.

From this exploratory study, a correlation between the gamma-ray emission measured by VERITAS and *Fermi*-LAT of the iconic HBL blazar Mrk 421 is clearly apparent. The greatest discrepancy between VERITAS and *Fermi*-LAT occurred early in the dataset, corresponding to flaring activity observed by VERITAS in 2010 (Aleksić et al., 2015), while the correlation is strongest in the most recent data, after VERITAS began regularly collecting unbiased monitoring observations. While these correlations and discrepancies may reflect real patterns in the GeV and TeV emission of Mrk 421 in the period studied, they may also reflect systematic effects in the VERITAS data, including the uneven sampling of flux states, a shorter time binning compared to *Fermi*-LAT, and seasonal offsets in flux calibration. A more detailed investigation in future work would be needed to disentangle these factors.

5.6 Investigating the Impact of Variability on Source Selection

This section's contents have been published in the proceedings of the 36th International Cosmic Ray Conference (ICRC2019), where they were presented by the author (Brill, 2019).

As discussed in Section 5.4, constructing a valid TeV blazar LF requires emulating a uniform, flux-limited survey. One method to do this is to consider subsets of targets in

a multiwavelength catalog that fulfill physically motivated selection criteria, which could then be used to derive an approximate equivalent TeV flux limit. Assuming that all TeV gamma-ray emitters of the relevant source class are in the underlying catalog, and that the catalog itself provides an unbiased, complete sample, this process results in a complete, uniform survey up to the flux limit. The chosen catalog and selection criteria must maximize completeness while minimizing false positives. Multiple catalogs could be combined to select targets of different source classes.

This method can be investigated using the objects selected by Costamante and Ghisellini (2002, hereafter CG02), who produced a list of 33 candidate (and 5 already-known) TeV BL Lac objects using selection limits on the X-ray and radio energy flux. Their sources were selected from several samples of BL Lac objects for which radio, optical and X-ray observations were all available. Of their 38 candidate and known sources, 30 now have TeV detections and all have 4FGL associations (Abdollahi et al., 2020). Of these objects, 31 are visible to VERITAS (defined as having a declination between $-10°$ and $+70°$), which has detected 20 and published upper limits on 8 of them (Archambault et al., 2016).

The CG02 TeV candidate BL Lac objects can be used to explore the potential, and possible pitfalls, of this source selection approach for emulating a TeV blazar survey. To be useful, a source selection method based on an external catalog must both allow for the establishment of an effective TeV flux limit and provide a reasonably complete sample.

One way to obtain a predicted TeV flux given observed fluxes at other wavelengths is to derive it from an SED model fitted to the multiwavelength data. Fortunately, this is exactly what CG02 have already done. They apply two models, a one-zone Synchrotron Self-Compton (SSC) model and a parameterization from Fossati et al., 1998 which is built to describe sources with synchrotron and gamma-ray peaks of equal power, and use both to predict the energy fluxes above 40 GeV, 0.3 TeV and 1 TeV without incorporating absorption by the EBL. Figure 5.8 shows the distributions of the flux predictions above 0.3 TeV for the

two models, with EBL absorption additionally applied using Model C of Finke, Razzaque, and Dermer (2010). The redshift of each source was obtained from TeVCat or SIMBAD (Wenger et al., 2000) if known, assuming $z = 0.2$ otherwise, and the average energy calculated assuming a power-law spectrum with $E_{\rm th} = 0.3$ TeV and $\Gamma = 3.5$.

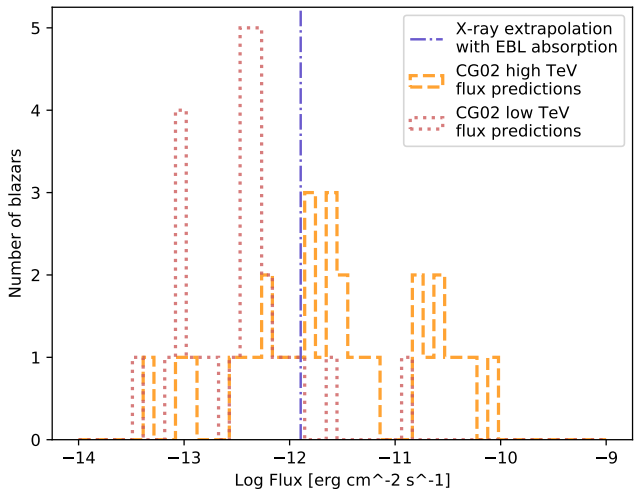


Figure 5.8: Distributions of the predicted TeV fluxes above 0.3 TeV of CG02, with EBL absorption applied using Model C of Finke, Razzaque, and Dermer (2010). The "low" and "high" flux predictions refer to the SSC model and parameterization of Fossati et al. (1998), respectively.

A second way to obtain a TeV flux limit is to use a simple physically motivated relationship between luminosities at different wavelengths, such as that of Stecker, de Jager, and Salamon (1996), $\nu_{TeV} F_{TeV} \sim \nu_X F_X$ for X-ray selected BL Lac objects. An extrapolation using this relationship of the X-ray selection flux limit of CG02, $F_X = 1.46$ μJy, with the same EBL absorption correction applied, is also shown in Figure 5.8. The predicted fluxes span over three orders of magnitude, with the extrapolated X-ray limit over an order of magnitude higher in flux than the lowest TeV prediction of each of the two models. For these predictions and luminosity relation to provide a useful flux limit, the predicted fluxes should cut off sharply at the low-flux end at a level consistent with the extrapolated flux limit. However, this behavior is not evident.

In addition, to take the selected sources as the basis for a flux-limited sample, the sensitivity of actual observations must match the supposed flux limit. Figure 5.9 shows the fluxes of the 33 BL Lac objects actually detected by VERITAS, as well as upper limits of blazar discovery targets observed but not detected, overlaid on the predicted fluxes of CG02. The reported blazar fluxes plotted in Figure 5.9 were calculated using fluxes in Crab Units and spectral indices from TeVCat, assuming $\Gamma = 3.5$ for any source with no reported index and $F_{\text{Mrk 501}} = 0.85$ Crab[2]. The energy fluxes were calculated by converting the Crab fluxes into photon fluxes above 0.3 TeV, assuming a Crab power-law spectrum with index 2.49 and normalization 3.2×10^{-11} cm^{-2} s^{-1} TeV^{-1}. The photon flux was converted into energy flux above 0.3 TeV using the spectral index, assuming a power law spectrum. These fluxes do not necessarily come from similar emission states, limiting the physical interpretation of this distribution, but are here intended for characterizing the sensitivity of VERITAS to these sources. The distribution of VERITAS blazar upper limits was taken from Archambault et al., 2016, with differential flux limits converted to energy flux assuming $E_{\text{th}} = 0.3$ TeV and $\Gamma = 3.5$ for all sources. A rough drop-off in both the detected and constrained fluxes is apparent around 1×10^{-12} erg cm^{-2} s^{-1}. This value approximately matches the extrapolated X-ray flux limit, but is significantly higher than the lowest fluxes predicted by CG02 using spectral modeling.

Also shown is the empirical flux limit of Broderick, P. Chang, and Pfrommer (2012) at 4.19×10^{-12} erg cm^{-2} s^{-1}, which was derived from a sample of 28 objects with publicly available well-defined SEDs observed by H.E.S.S., MAGIC, and VERITAS. This limit appears too high to describe well the flux distributions from VERITAS or the predictions of CG02, indicating that a careful consideration of the sample being used is necessary when defining a TeV flux limit.

Finally, for the source selection method to be useful, not only should VERITAS have

[2]This estimated flux is somewhat high for this source, but may be taken as representing a flaring flux.

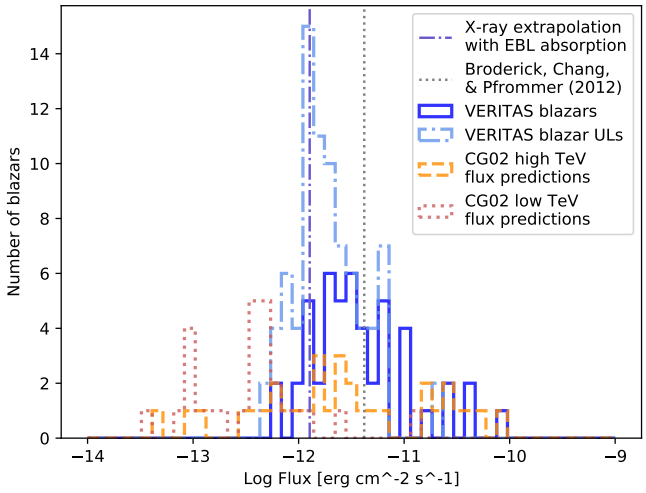


Figure 5.9: Distributions of the reported fluxes of blazars detected by VERITAS (solid dark blue) and VERITAS blazar upper limits (dotted-dashed light blue), compared to the flux predictions of CG02 shown in Figure 5.8. Two estimated flux limits are also shown.

observed all of the selected sources, but the converse must also be true: the source selection must be complete in the sense that all of the sources detected by VERITAS above the effective TeV flux limit are included. In fact, this is not the case. Of the 33 BL Lac objects detected by VERITAS, only 20 are CG02 known sources or candidates, and the other 13 are not in the catalog, a $\sim 40\%$ incompleteness rate. Figure 5.10 shows the distributions of these two subpopulations. Visually, the distributions do not differ substantially, particularly at the critical low-flux end, showing that this incompleteness cannot be captured by a simple difference in flux levels (such as excluded sources being dimmer).

Setting an effective flux limit and obtaining a complete sample are both found to be challenging. The variability of blazars at all wavelengths can explain these difficulties. First, sources in a low state when measured by multiwavelength surveys could fall below the selection criteria flux level but still be TeV emitters. The X-ray, optical, and radio data available in the literature to CG02 to set selection cutoffs and assemble SEDs were not necessarily simultaneous, which, in addition to uncertainties from their model choices and parame-

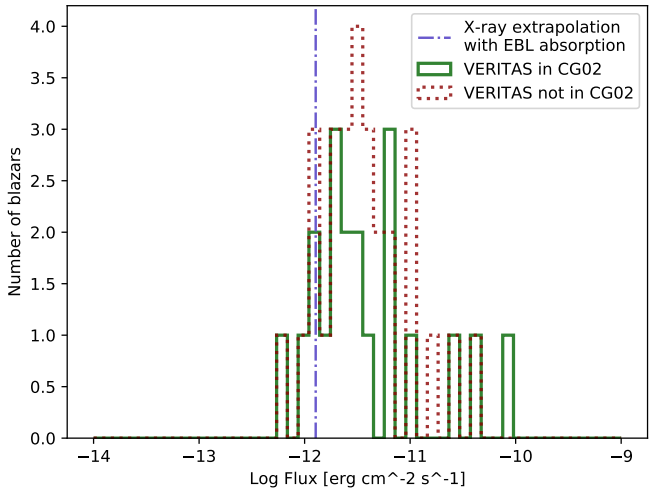


Figure 5.10: Distribution of energy fluxes of BL Lac objects detected by VERITAS from Figure 5.9, split into those included in the candidates or known sources of CG02 (solid green histogram) and those that are not (dotted red histogram).

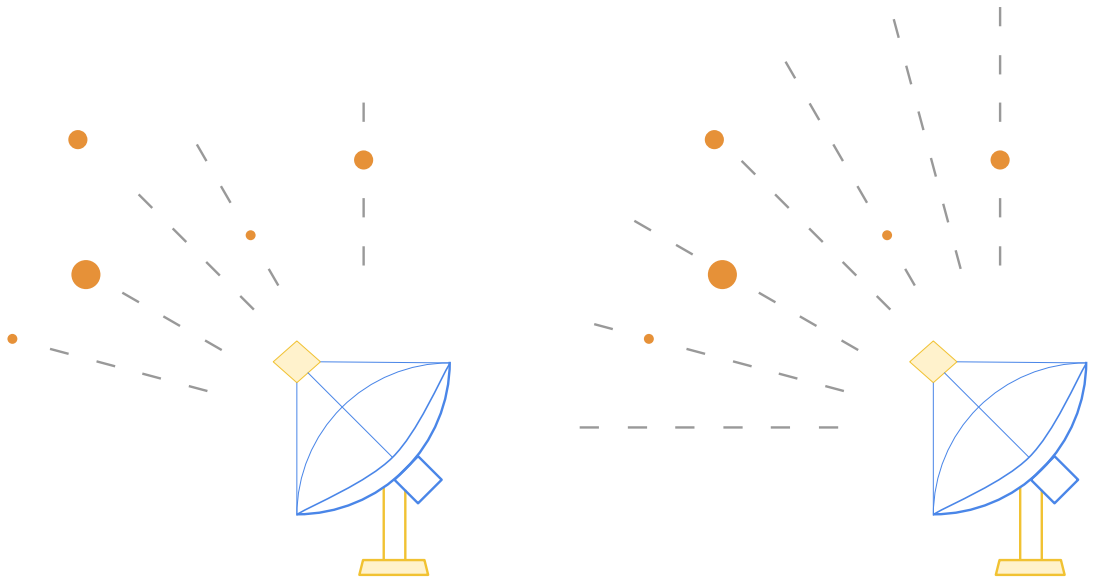
ters, likely played a role in incomplete source selection. For example, one VERITAS source missed by CG02, W Comae, was identified as a promising candidate for TeV emission by a study that performed detailed modeling of its simultaneous broadband SED and X-ray variability (Böttcher, Mukherjee, and Reimer, 2002). Since simultaneous multiwavelength measurements are not guaranteed to exist in the literature for every blazar, uncertainties in predicted fluxes and selection thresholds are to some extent inherent in any selection method for TeV blazar candidates relying on archival data.

In addition, the difficulty of predicting TeV fluxes reflects not only uncertainties in extrapolating from lower wavelengths, reducible with simultaneous measurements and detailed modeling, but also actual variability in the TeV emission. TeV blazar detectability thus depends both on the limiting flux and the flux state when observed. A source might only be detected if it by chance flared while being observed. These effects must be taken into account as potential sources of systematic error when using a target selection method based on a multiwavelength catalog.

5.7 Prospects for a VERITAS Luminosity Function Observing Program

5.7.1 Observing Program

In Fall 2019, VERITAS began a program to study a complete sample of TeV-emitting HBLs with a flux limit of approximately 1% Crab in order to measure the LF of these sources. The LF program includes both the analysis of archival data and an ongoing high-priority observing campaign to obtain a minimum unbiased exposure on each target. A uniform TeV survey is emulated by selecting 36 targets from the multiwavelength 3HSP catalog, the largest and most complete catalog of HBL blazars currently available (Y. .-L. Chang et al., 2019). The strategy of the observing program is illustrated in Figure 5.11.



(a) VERITAS LF Program (b) Unbiased Survey

Figure 5.11: Concept of the VERITAS luminosity function observing program, with an unbiased survey illustrated for comparison.

The 3HSP catalog cross-correlates sources with infrared spectra similar to those of known TeV blazars with radio and X-ray data to calculate a figure of merit (FOM) that predicts their TeV flux. FOM is defined in units of 2.5×10^{-12} erg cm^{-2} s^{-1}, which is the peak

synchrotron flux of the faintest HBL included in TeVCat (Y. .-L. Chang et al., 2019). We selected sources that are likely TeV emitters (FOM > 2), visible at high elevation with VERITAS ($1.7° \leq \delta \leq 61.7°$), and outside the Galactic plane ($|b| > 10°$), resulting in 36 HBLs, 20 of which are in TeVCat, and all of which are in the 4FGL catalog (Abdollahi et al., 2020). The 36 targets are listed in Appendix C. All but five of the sources have known redshifts. For comparison, Schmidt (1968) derived the LF of quasars using 33 sources.

VERITAS can detect a steady source with a flux of 1% Crab in about 25 hours of observations, so this program will require about 10^3 observing hours. To make it practical, we must use archival data in addition to new observations. By default, we will exclude all data judged to be potentially biased, such as target-of-opportunity observations, though as discussed in Section 5.5, we are developing further techniques that may allow us to make use of these data. Additional data will be taken only on those sources without enough preexisting exposure to make a good measurement of their flux. To reduce this program's time requirements, after completing a source's initial allocation of unbiased observations in a season, if the significance of the accumulated signal is less than 2.5σ the source is assumed to be below the intended sensitivity threshold and observations on it are terminated.

5.7.2 Simulating the Expected Data

While many of the LF program targets are known TeV sources, some are not, and all of them may be variable sources. To model these uncertainties, we randomly generated 20 different simulated datasets following the same procedure for each. We began by selecting the 36 sources listed in Appendix C from the 3HSP catalog using the selection criteria given in the previous section. The sky coverage coefficient Ω_0 resulting from the combined effect of the declination and galactic latitude cuts was numerically estimated to be 0.349.

In Figure 5.12, we gauge incompleteness in our catalog by plotting the $\log N$ - $\log S$ curve for the 3HSP sources, overlaid with the synchrotron flux limit equivalent to our FOM > 2

selection criterion. The flattening of a $\log N$ - $\log S$ plot at the low-flux end may indicate incompleteness in the sample, among other causes, such as evolution of the population and cosmological expansion (e.g. Peterson, 1997). In any case, the flux limit from the FOM cut clearly falls within the constant-slope portion of the curve, consistent with a complete sample. We therefore take systematic error due to incompleteness in the source catalog to be negligible compared to other sources of uncertainty in this study.

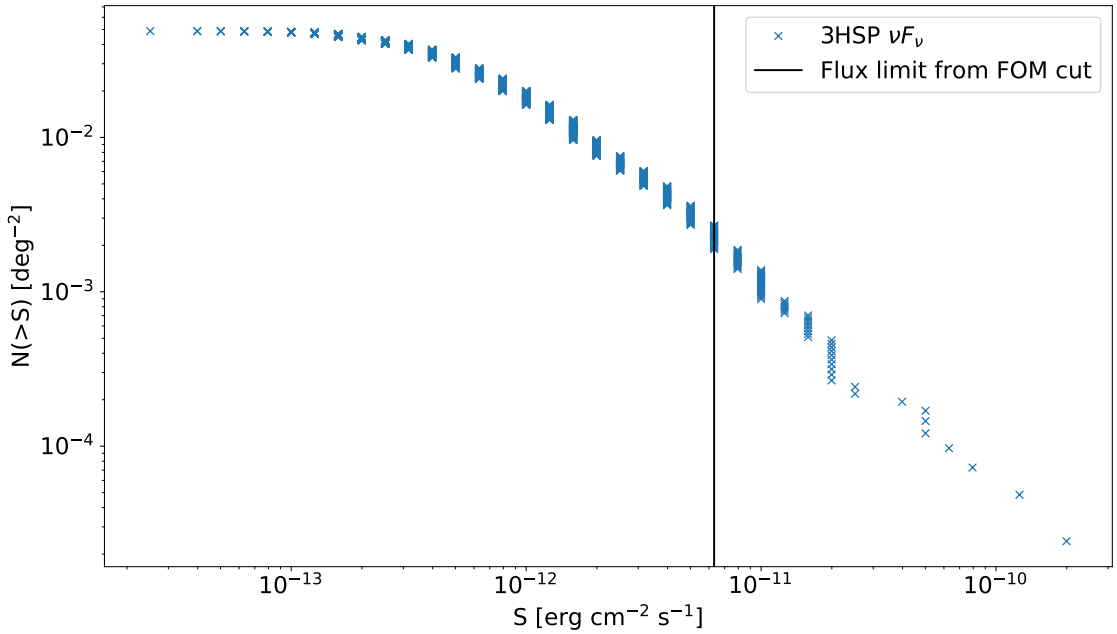


Figure 5.12: $\log N$ - $\log S$ plot for the 3HSP catalog, with the flux limit corresponding to FOM > 2 indicated. The synchrotron fluxes in the 3HSP catalog are reported to the nearest 0.1 dex.

Each source in our simulated dataset is defined by its observed flux F, power-law photon spectral index Γ, and redshift z. If available, we obtained baseline values from TeVCat for the flux in Crab Units and the spectral index, and used the redshift reported in either TeVCat or SIMBAD. A flux of 0.23 Crab was assumed for Mrk 501 based on its estimated long-term average flux measured by VERITAS (Hervet, 2020). We converted flux values from Crab Units to photon flux using the Crab spectrum reported by Hillas et al. (1998) and fixing the energy threshold to 0.2 TeV for every simulated source. To roughly account for the effect of

excluding biased data, the usable VERITAS archival exposure on each source was estimated to be 70% of the actual amount available. This estimate resulted in about 1600 hours of usable data, of which 15-20% were dedicated unbiased LF observations.

We randomly simulated parameters for each source by drawing from distributions based on the baseline values. For sources with TeVCat fluxes, the average flux was simulated by sampling from the lower half of a normal distribution with mean equal to the TeVCat flux and standard deviation equal to half that value. We allowed only negative deviations from the TeVCat flux because many reported fluxes come from flares and should be taken as upper bounds on the average flux. For sources not in TeVCat, the simulated flux was generated by sampling from a normal distribution with mean and standard deviation equal to 0.01 Crab. Negative fluxes were set to zero. Spectral indices were sampled from a normal distribution with mean equal to the spectral index reported in TeVCat, or 3.5 if unavailable, and standard deviation of 0.5, with a cutoff imposed at 1.5. Redshifts were fixed to the reported value, if available, and otherwise sampled from a normal distribution with mean equal to 0.4 and standard deviation of 0.1, with a cutoff at 0.

To determine the sky coverage, we needed to estimate the precision and sensitivity of the VERITAS telescopes. We calculated the expected signal and background as a function of flux and exposure by scaling constant values of the signal rate R_γ, background rate $R_{\rm bg}$, signal to background exposure ratio α, and flux normalization F_0 estimated from an analysis of a typical HBL blazar (Errando, 2020). The expected signal and background counts for a source with flux F and exposure t are then given by:

$$\begin{aligned} N_\gamma &= R_\gamma \frac{F}{F_0} t \\ N_{\rm bg} &= R_{\rm bg} t. \end{aligned} \tag{5.31}$$

We then have the on and off counts, $N_{\rm on} = N_\gamma + N_{\rm bg}$ and $N_{\rm off} = N_{\rm bg}/\alpha$ (T.-P. Li and Ma, 1983). Using Equation 9 of T.-P. Li and Ma (1983, Eq. 3.5) and some algebra, we obtain the error on the flux,

$$\sigma_F = F_0 \sqrt{\frac{R_{\rm bg}}{R_\gamma^2 t}\left(1 + \alpha\left(1 + \frac{R_\gamma}{R_{\rm bg}}\frac{F}{F_0}\right)\right)}, \tag{5.32}$$

and the limiting flux,

$$F_{\rm lim} = \frac{\alpha S^2 F_0}{2R_\gamma t}\left(1 + \sqrt{1 + 4\left(\frac{1+\alpha}{\alpha^2}\right)\left(\frac{R_{\rm bg} t}{S^2}\right)}\right), \tag{5.33}$$

where we set the significance threshold to claim a detection to be $S = 5$ standard deviations. We then simulated VERITAS observations as follows:

1. Determine the total first-pass exposure based on the prescription in Table 5.1.

2. Calculate the expected error on the flux using Eq. 5.32, and determine the "measured" flux by sampling from a normal distribution with mean and standard deviation equal to the expected flux and the error. The measured flux is used in the analysis.

3. If the measured flux is less than 1.2% Crab, stop.

4. Otherwise, if the total exposure is less than 20 hours, increase the total exposure to 24 hours to represent follow-up observations, and repeat Step 2.

Archival exposure [h]	Additional LF exposure [h]
< 10	8
10 – 30	6
≥ 30	0

Table 5.1: Assumed exposures used to represent the observing strategy of the VERITAS LF program. The total exposure on a source in the first pass is given by 70% of the archival exposure plus the additional LF exposure.

To account for EBL absorption, a power-law VHE spectrum was generated for each source, defined by its spectral index, minimum energy fixed to 0.2 TeV, maximum energy varying with redshift as $E_{\max} = (3z)^{-1}$ TeV, and overall normalization set by its flux. The spectra were corrected for EBL absorption using the model of Domínguez et al. (2011) and numerically integrated to obtain the total intrinsic energy flux. The intrinsic luminosity was then calculated using

$$L = \frac{4\pi F D_L^2(z)}{(1+z)^{1-\alpha}} \tag{5.34}$$

where the term in the denominator is the K-correction, which which accounts for the apparent decrease in brightness from the redshifting of the spectrum, and depends on the energy index α, which is related to the photon index as $\Gamma = \alpha + 1$. To calculate the luminosity distance, a flat ΛCDM cosmology with $H_0 = 70$ km s^{-1} Mpc^{-1}, $\Omega_M = 0.3$, and $\Omega_\Lambda = 0.7$ was assumed. The synchrotron peak luminosities were also calculated using Eq. 5.34 from the values of νF_ν in the 3HSP catalog and setting $\alpha_{\rm syn} = 1$. Distributions of the parameters of one representative simulated dataset are shown in Figure 5.13.

5.7.3 Estimating the Expected Luminosity Function

After assembling a simulated sample, we calculated the limiting redshift of each object, given as the minimum of the limiting redshifts defined by the sensitivity of the TeV observations (Eq. 5.33) and the selection threshold from the 3HSP catalog. The 3HSP FOM > 2 selection threshold was equivalent to a synchrotron flux limit of 6.31×10^{-12} erg cm^{-2} s^{-1}. Figure 5.14 shows a diagram of the limiting redshifts for a representative simulated dataset. If the simulated redshift was greater than the limiting redshift, the source was excluded from the remainder of the analysis, as it would not have been detected.

Next, a parametric LF model was fit to the data, as described in Section 5.3.3. The

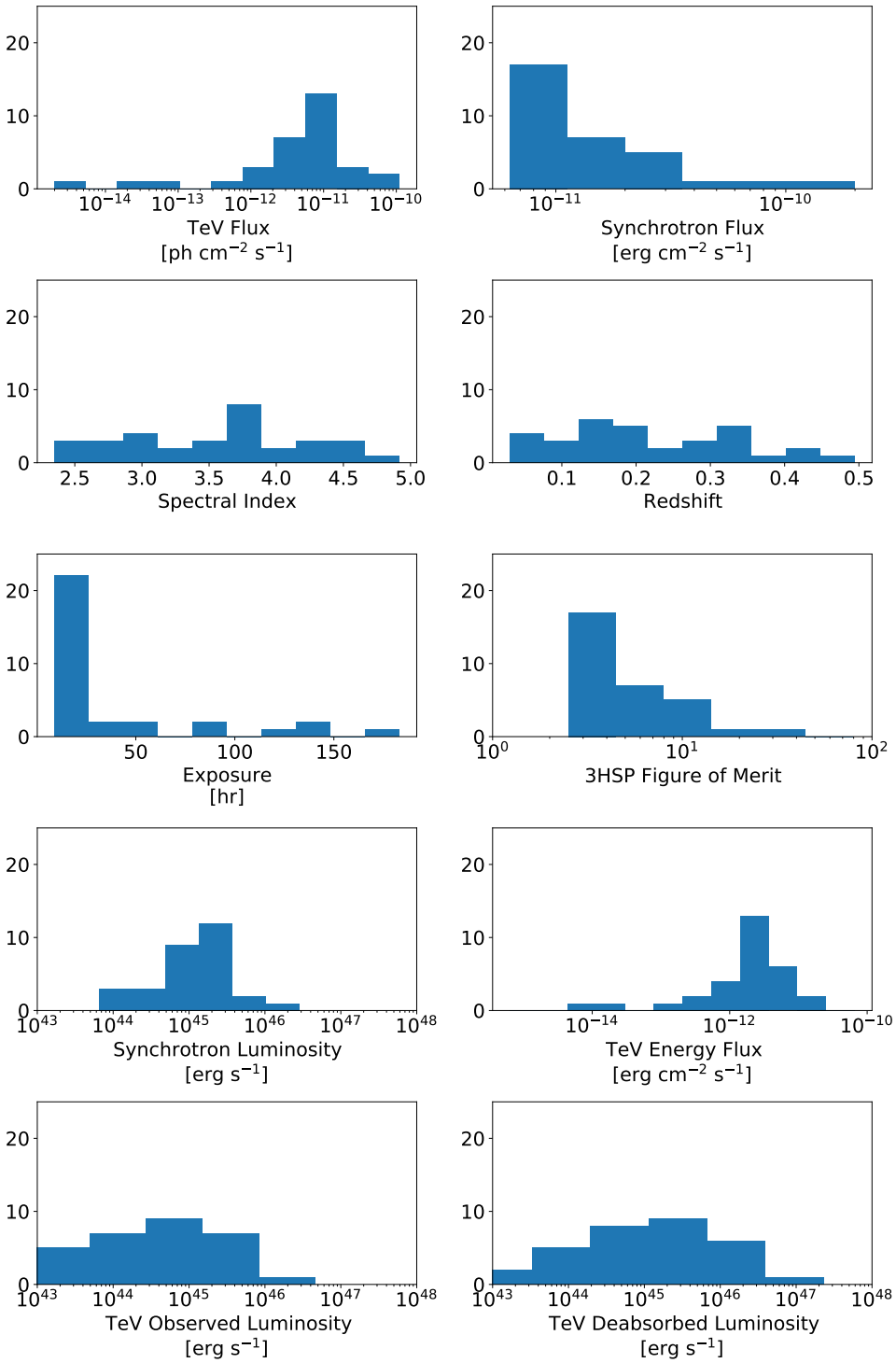


Figure 5.13: Parameter distributions derived from a representative simulated dataset.

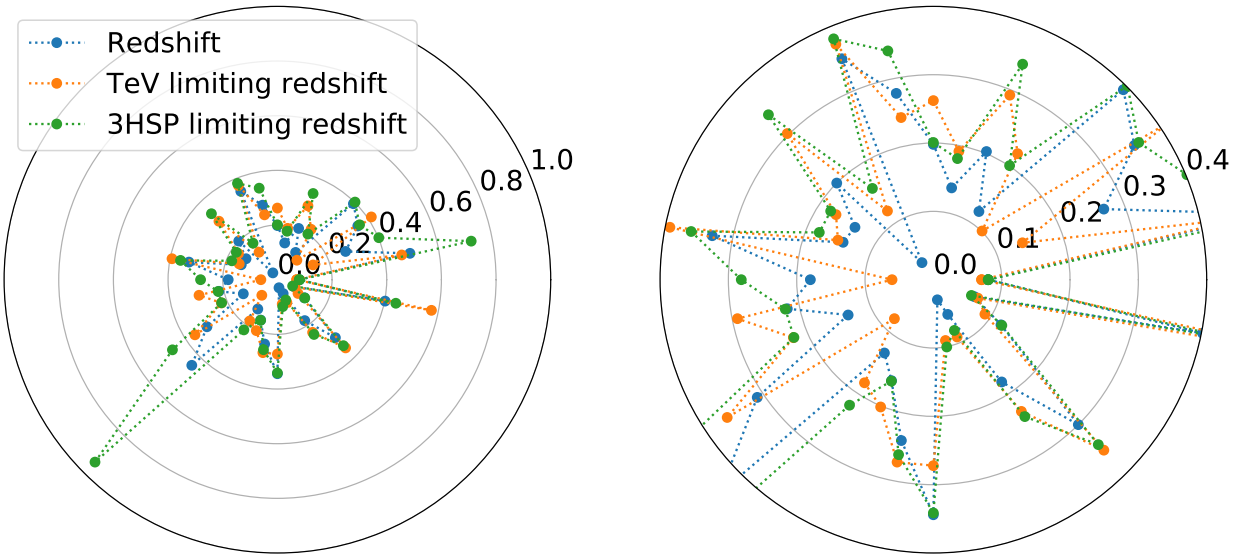


Figure 5.14: *Left*: Limiting redshifts inferred from the 3HSP synchrotron fluxes and simulated TeV gamma-ray fluxes in a representative simulated dataset. *Right*: The same plot zoomed in to show $z < 0.4$.

fitting was performed using Markov chain Monte Carlo (MCMC) sampling to produce posterior probability distributions over the model parameters, allowing both the optimal values and their uncertainties to be recovered. The *emcee* package was used to perform the sampling (Foreman-Mackey et al., 2013). The LF was modeled as a pure power law (Eq. 5.15) exhibiting pure luminosity evolution (Eqs. 5.7 and 5.8). We used the simplest possible sky coverage term (Eq. 5.21) that included no factors to account for incomplete, irregular, or biased observations. To avoid introducing too many nuisance parameters into the fit, a single flux limit was used for all objects in each sample, determined by calculating the sensitivity of VERITAS to a source with the average exposure and spectral index for that sample.

The LF model contained three free parameters, the amplitude A, index γ, and evolution parameter k. When a smoothly broken double power-law model was used (Eq. 5.16), an adequate fit could not be achieved. The two power-law indices were highly correlated, indicating that a single power-law slope gave a better description of the data. Similarly, evolution functions more complex than Eq. 5.8 were found not to be well constrained.

For each of the 20 simulated datasets, the MCMC sampling was run for 1750 steps with 10 walkers, with the first 250 steps discarded as a burn-in period. Samples were extracted from the chain every 50 steps, equivalent to about half the autocorrelation time estimated from a long chain with 1×10^4 steps. Figure 5.15 shows a representative MCMC chain from one of the simulated datasets. The sampled values from all 20 datasets were combined to obtain the overall posterior distributions of the model parameters. Figure 5.16 shows the one and two dimensional projections of these distributions, plotted using *corner* (Foreman-Mackey, 2016). The best-fit values with $1\,\sigma$ uncertainties are given in Table 5.2. The $1\,\sigma$ lower bound, best fit, and $1\,\sigma$ upper bound were taken as the 16th, 50th, and 84th percentiles of the parameter distributions, respectively. The evolution parameter k was found to be negative, though with a large uncertainty, consistent with the negative evolution measured in the GeV gamma-ray band for the subpopulation of HBL blazars (Ajello et al., 2014).

A^a	γ	k	$L_{\rm scale}{}^b$
$7.6^{+7.4}_{-3.9}$	$-0.46^{+0.13}_{-0.13}$	$-5.7^{+5.5}_{-7.8}$	10^{46}

Table 5.2: Best-fit LF parameters for a power law with pure luminosity evolution.
[a] In units of 10^{-9} Mpc^{-3} erg^{-1} s.
[b] In units of erg s^{-1}. Fixed parameter.

We also used the weighted $1/V_{\rm max}$ method (Eq. 5.14) to perform a semi-independent calculation of the LF. The weighting was performed using Eq. 5.8 with the best-fit values of k from the MCMC fit. The two estimated LF models are shown in Figure 5.17. The best-fit parametric model and its $1\,\sigma$ error region were plotted using the 16th, 50th, and 84th percentiles of the distribution of model values at each luminosity. The large error bars on the $1/V_{\rm max}$ points reflect the variance in the number of source detections in any given luminosity bin over different realizations of the observing program. Figure 5.18 shows the distributions of the number of detected sources and minimum measured flux over the simulated samples.

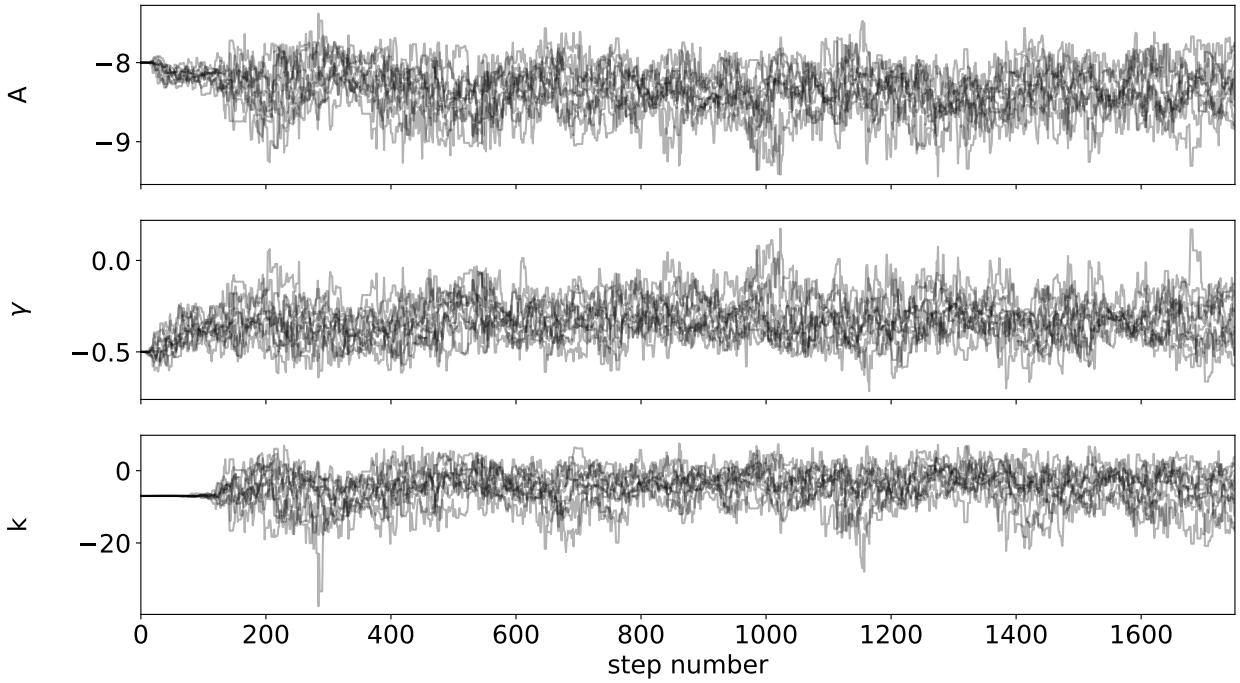


Figure 5.15: Parameter values of the MCMC sampling as a function of step, demonstrating the convergence of the model after a few hundred steps.

These simulations indicate that when the LF observing program is complete, VERITAS can expect to detect 25 ± 2 out of 36 targets and to be sensitive to evolution by detecting deviations of $\langle V/V_{\rm max} \rangle$ from 0.50 with an error of ± 0.07. The use of archival data is critical for making this measurement, as only 15-20% of the required exposure comes from dedicated observations. With a determination by VERITAS of the TeV HBL LF similar to our simulated measurement, the TeV gamma-ray emission of unresolved HBLs can be estimated and compared to measurements in other wavelengths and messengers. However, as we cannot resolve structure in the shape of the LF, it is not clear whether VERITAS will be able to deconvolve the effects of relativistic beaming to measure the intrinsic LF.

This study can be continued in several ways. First, the increased sensitivity that will be provided by the CTA extragalactic survey may enable a better estimation of the LF. The quality of a LF measurement by CTA could be estimated using a simulation framework

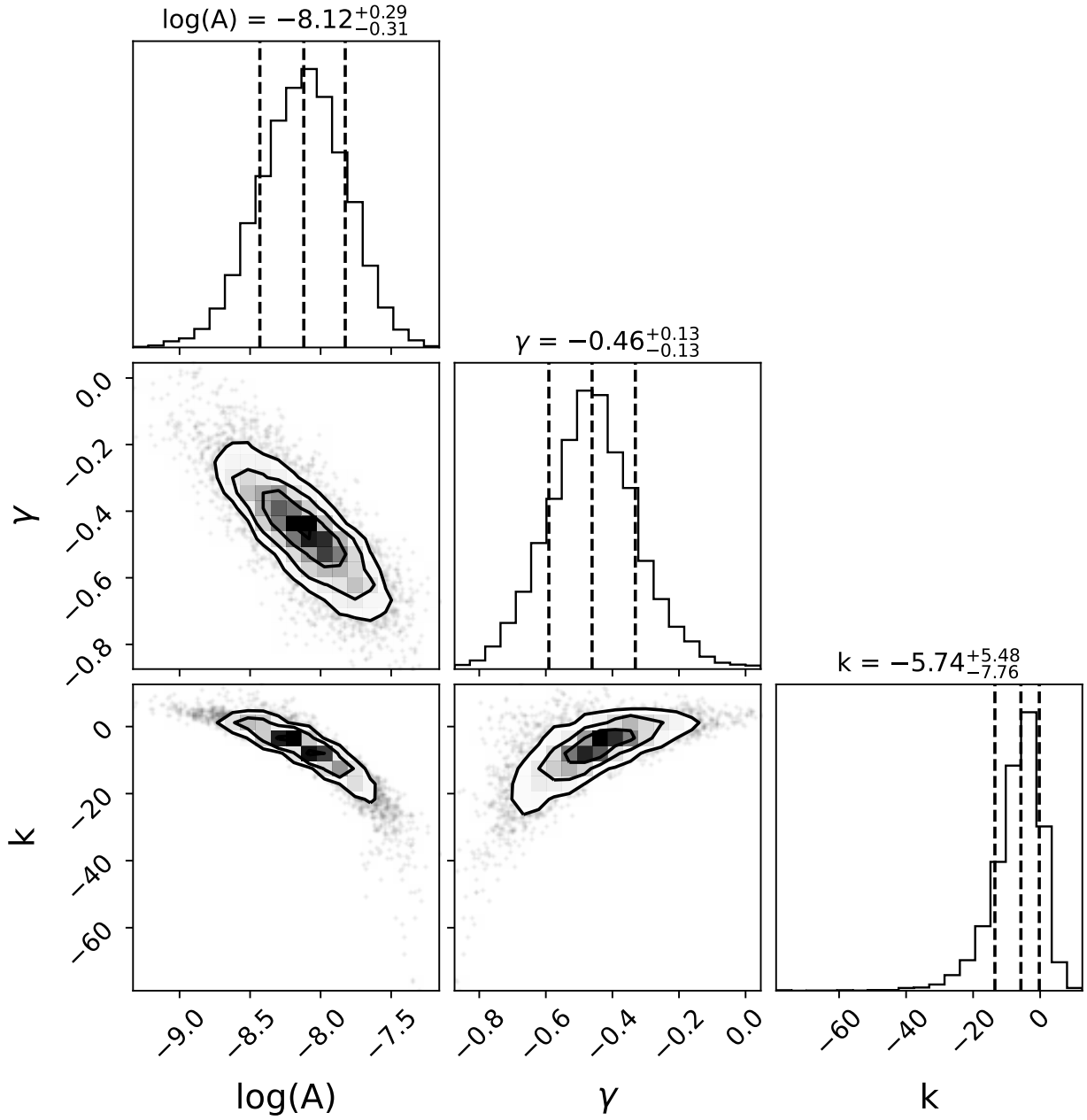


Figure 5.16: Distributions of the LF model parameters from the MCMC fit.

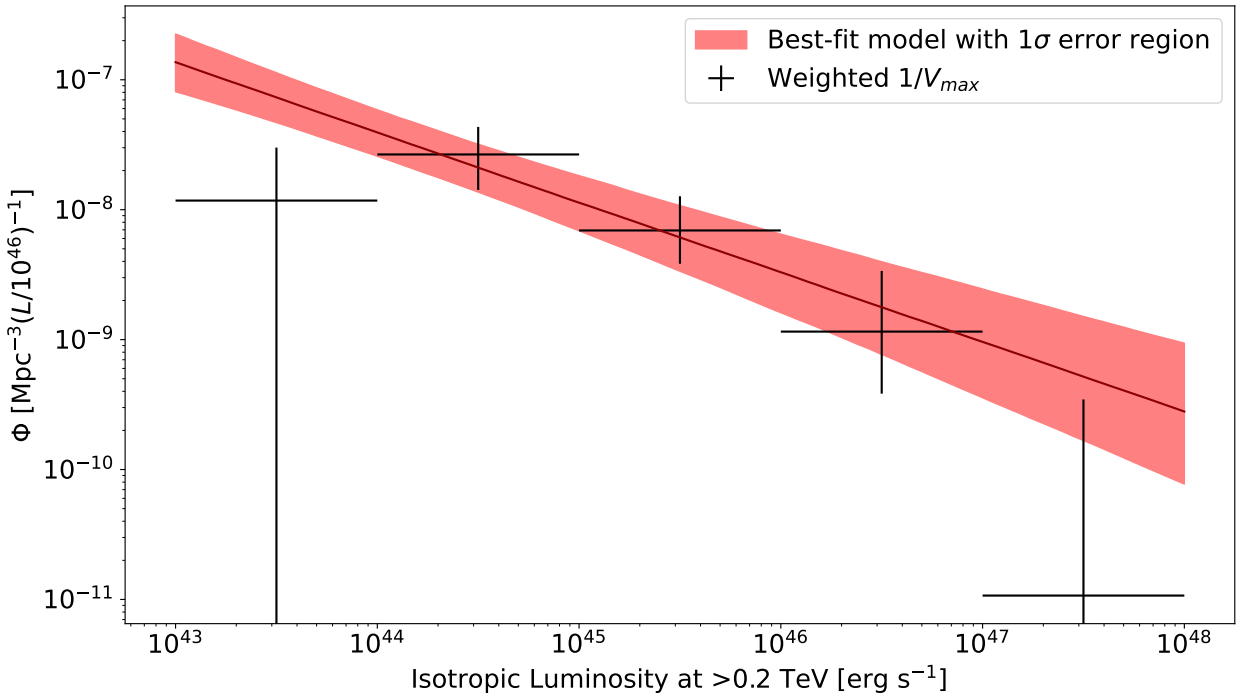


Figure 5.17: Estimated LF derived from the simulated observations. The 1σ error region around the best-fit power-law model is shown as a red band. The black points show the binned LF estimated using the weighted $1/V_{\rm max}$ method.

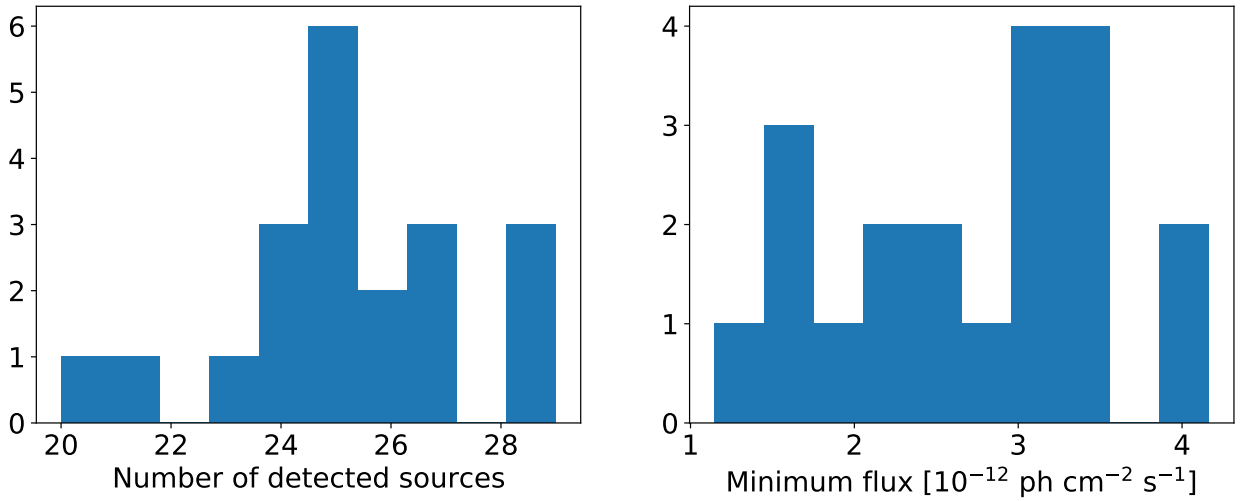


Figure 5.18: Distributions of the number of detected sources (*left*) and minimum measured flux (*right*) in the 20 simulated source samples.

similar to this one. CTA is expected to discover new TeV blazars, so the expected source population could be estimated by extrapolating from, for example, a catalog of *Fermi*-LAT blazars (e.g. 4LAC, Ajello et al., 2020). Since the CTA extragalactic survey will be uniform, source selection using a multiwavelength catalog would not be required.

While we assumed a model-independent source population based on known TeV blazars in this work, a complementary approach would be to simulate source populations based on one or more assumed LF models in order to characterize the sensitivity of the reconstruction to different LF parameterizations. As part of such a study, the systematic error on the measured flux due to variability could be more precisely quantified by simulating variable blazar light curves and sampling them to represent different observation strategies. Expectations for estimates of contributions to extragalactic radiation fields and distributions of beaming parameters could be derived from the reconstructed LF, as well.

Chapter 6: A Control and Monitoring Software System for the Prototype Schwarzschild-Couder Telescope Camera

We now turn from scientific studies conducted with the current-generation VERITAS telescopes to the development of new instruments and analysis techniques for CTA. We begin with a contribution to the commissioning and improved operation of a prototype IACT, the pSCT. In this chapter, we discuss the design of a new software system developed to control and monitor the pSCT camera. The efficient and safe operation of the pSCT facilitated by this software enables all of the scientific goals outlined in Section 3.6.

We begin by briefly describing the existing software that has been used to take data with the pSCT camera, and the camera subsystems that it controls. We then explain the motivation for developing a new software system and the functional requirements driving its high-level design. We describe in detail the implementation of that design in the `sctcamsoft` software package. We discuss an improved procedure for performing rate scans which has been included in the software, and describe a system developed to automatically generate a written log of runs taken with the pSCT camera. Finally, we acknowledge contributions made by others to `sctcamsoft`.

6.1 pSCT Camera

The pSCT camera has primarily been operated using a suite of Python and Bash scripts based around the `psct_toolkit` Python library, which was originally designed for performing laboratory tests with the pSCT backplane (C. B. Adams, 2018). In the remainder of this chapter, we refer to this suite of existing software collectively as "pSCT Toolkit".

6.1.1 Data Runs and Rate Scans

Two types of normal operations are supported in pSCT Toolkit: data runs and rate scans. Data acquisition is performed in discrete data runs, in which one data file is produced per run. In a rate scan, which is a calibration procedure, a sequence of measured trigger rates is produced but not any data. However, both data runs and rate scans are considered to be "runs" by the software and are assigned numerical run IDs in the same sequence. This is done so that both run types can be treated identically for purposes of setup, monitoring, logging, and file transfer.

In a rate scan, the trigger rate is measured at a sequence of trigger threshold values, measured in digital-to-analog converter (DAC) counts. In the pSCT, the trigger threshold value is called the *thresh*. A high *thresh* corresponds to a low threshold and results in a high trigger rate. This is the case because the trigger signal in the TARGET chip passes through three inverting amplifiers, inverting the output signal. Rate scans are therefore performed from high to low *thresh*, equivalent to low to high threshold. Some example rate scans performed with the pSCT are plotted in Figure 6.1. A rate scan plot is also known as a bias curve.

In the current pSCT camera, the trigger rate is dominated by trigger noise produced by the camera electronics. As this noise is correlated heavily with module temperature which can vary from run to run, a rate scan must be performed before each run to ensure that a manageable trigger rate is obtained. Events triggered by noise are discarded during the data analysis. The camera upgrade mentioned in Section 3.6 is expected to produce nominal performance. When electronics noise becomes subdominant, the trigger rate is expected to decrease with increasing threshold as a broken power law, with the bias curve decreasing sharply at first in the regime dominated by night sky background, followed by a flattening in the regime dominated by cosmic ray triggers at higher thresholds. In that situation,

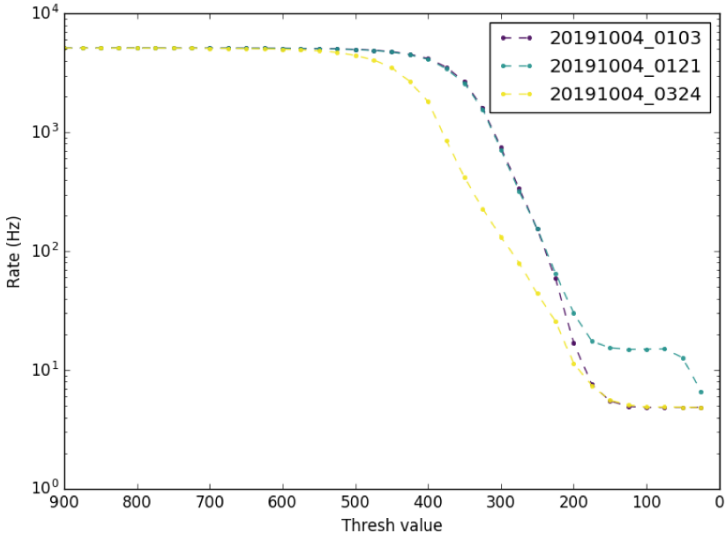


Figure 6.1: Example rate scans performed with the pSCT, from Taylor (2021). Each curve is plotted with a constant offset of 4.8 Hz. Scan "0121" (turquoise) contains a plateau caused by the flasher running at 10 Hz. In scan "0324" (yellow), the shutter was closed to block external triggers. The small resulting change indicates that most triggers are caused by internal electronics noise.

the trigger threshold should be set close to the transition point in order to set the energy threshold as low as possible without introducing too much night sky background noise.

6.1.2 Camera Subsystems

Modules and Backplane

Communication with the modules is performed using the `TargetDriver` and `TargetIO` libraries (Zorn et al., 2018). `TargetDriver` controls and monitors the modules, while `TargetIO` handles data readout. These C++ libraries are accessed through a Python interface built using SWIG[1].

A Raspberry Pi computer[2] is used to perform housekeeping on and communication with the pSCT backplane. All monitoring and control of the backplane is done using a C program

[1] `http://www.swig.org/index.php`
[2] `https://www.raspberrypi.org/`

installed on the Raspberry Pi which relies on the `bcm2835` library[3]. The C program on the Raspberry Pi is controlled from within the Python software over a network connection using the Pexpect library[4]. A "trigger mask" file located on the Raspberry Pi is used to pause triggering when needed and to prevent noisy trigger pixels from contributing to event triggers.

Some monitoring functionality, in addition to all data acquisition, is performed using the module and backplane software. The FEE temperatures and pixel currents are read out from MAX1230 analog-to-digital converter (ADC) units[5] using `TargetDriver`. The temperature and current data are recorded in text files. The FEE voltages and currents, and whether or not they are detected as powered on by the backplane ("presence"), can be read out using the software on the Raspberry Pi. While the SiPM temperatures are regulated and monitored by a micro-controller, no software to access it is included in pSCT Toolkit or the new software described in this chapter.

Auxiliary Subsystems

The camera fans, camera power supply, and network setup and monitoring are controlled by Bash scripts which are independent of the main `psct_toolkit` Python library. There is no capability to monitor the chiller temperature and pressure in pSCT Toolkit, although the chiller ControlByWeb X-320 instrumentation module provides an SNMP[6] interface which may be accessed programmatically. These variables are also displayed on a web interface and an analog gauge on the chiller.

The camera flashers are controlled using Python scripts that run on the Raspberry Pi. For convenience, a graphical user interface to control the flashers has been built on top of these scripts. This interface has not been modified in the new software system described in

[3] `https://www.airspayce.com/mikem/bcm2835/`

[4] `https://pexpect.readthedocs.io/en/stable/`

[5] Because the ADC units introduce noise into the trigger path when active, data-taking normally must be paused to read out the FEE temperatures and pixel currents.

[6] Simple Network Management Protocol.

this chapter, and is not discussed further.

The camera shutter cannot be operated automatically using pSCT Toolkit or the new software system, but implementing automatic control or monitoring of the camera shutter via its ControlByWeb X-301 sensor could be a subject of future work.

External pSCT Software

The pSCT incorporates a number of additional software systems that are beyond the scope of the camera software described in this chapter. In particular, no software developed to align the pSCT mirrors and camera is included in the system described here. The camera software is also independent of the telescope positioner software, with the minor exception that automatic run log generator described in Section 6.6 passively connects to that software to read out the coordinates at which the telescope is pointing.

6.2 Motivation for a New Software System

The software described in Section 6.1 suffices for allowing experts on the camera system to operate the telescope, conduct engineering tests, and collect data during commissioning, but it is not suitable for performing regular scientific observations. A more complete software system is needed for the SCT to become a mature instrument.

6.2.1 Monitoring

In many cases, pSCT Toolkit does not contain software to help the observer access and interpret readouts from sensors in the pSCT camera. When this software is available, the observer must often actively run a script to retrieve the readout. It is inefficient and unrealistic for a human operator to do this continually throughout a data-taking session. Also, it may take detailed knowledge of a subsystem to tell if a sensor's readout is normal

or problematic. Observers who join the project after the commissioning phase may lack this knowledge. As a result of these deficiencies, critical issues might go undetected until too late, threatening the safety of the instrument and the quality of the data.

6.2.2 Logging

When using pSCT Toolkit, observers manually log information about every run, such as which modules were enabled, what the trigger threshold was, and whether auxiliary subsystems such as the flashers and fans were turned on. This process is repetitive and error-prone. Furthermore, there is no mechanism to record a stream of monitoring information, so identifying and fixing errors after they occur must rely solely on these high-level descriptions.

For logging to be useful for the camera operator, information from a variety of devices must be recorded, collected, and transformed into a useful format. Different uses and timescales require different log formats. First, information can be displayed in real time for monitoring purposes, as discussed above, in which case only short-term storage of logs is required. Second, standardized information on the runs taken during each observing session should be automatically recorded to inform data analysis and monitor data quality. The automatically generated logs should be supplemented with notes from the scientists taking the data. Third, for long-term, offline use, logging information should be permanently stored in a structured archive such as a database.

6.2.3 Centralization

As pSCT Toolkit consists of a collection of separate, unconnected scripts, it can be confusing for a non-expert observer to manage. Direct human intervention is needed when coordinating different camera subsystems in a high-level procedure, logging information to a centralized location, or ensuring that operations are performed in a safe order. Furthermore, future SCT camera software may be required to implement a finite-state machine (FSM)

model that exists in, and transitions between, well-defined states. Although the pSCT is not required to implement a full FSM, a unified software system must be developed to make progress towards this goal.

6.2.4 Graphical Interface

With pSCT Toolkit, the camera operators run scripts using a text-based command-line interface (CLI). The scripts are easy to modify, allowing rapid code development during pSCT commissioning. However, a more efficient and intuitive graphical user interface (GUI) would be preferred for future regular operations conducted by scientists lacking detailed knowledge of the camera software. The software described in this work is designed for compatibility with GUIs developed by other researchers, as discussed below in Section 6.4.

6.2.5 Efficiency

Because pSCT Toolkit was originally designed for testing the modules on the laboratory bench, it is not optimized for efficiently conducting astronomical observations. For example, it is common in the laboratory to modify an aspect of the modules' hardware or software, turn them on to observe their performance, and turn them off again to try something else. A full initialization of the modules is performed before every data run in pSCT Toolkit, reflecting this mode of operation. For a mature instrument, a system in which the modules are only initialized once and re-initialized if needed would increase the time available to collect data during routine operations. Other potential efficiency gains include employing a centralized server to support parallel operations and optimizing rate scans to achieve precise readings in the shortest time possible.

6.3 Software Design Considerations

To best achieve the operational goals described in Section 6.2, several key principles have been developed for designing new software to control and monitor the pSCT camera.

6.3.1 Consolidation of Domain Logic

The first principle guiding the software design is consolidating the domain logic that encodes the procedures and instructions for camera operation. This domain logic exists in the scripts and programs of pSCT Toolkit and their documentation, as well as in the implicit knowledge belonging to camera system experts. In order to support the goals of logging, centralization, and efficiency, routine manually-performed procedures should be automated and connected to the overall software system. The most straightforward way to implement this principle is through a main program that executes all domain logic of the camera system, potentially connecting with other programs to control the hardware and communicate with the user. In this setup, the software programs running different parts of the system must be compatible with each other and be able to be coordinated by the main program.

The principle of consolidation of domain logic further supports the goals of providing monitoring and graphical user interfaces, thereby contributing to the system's effectiveness, reliability, and safety. Connecting subsystems into a common framework makes it easier to obtain streams of monitoring data for display and logging. The main program is a natural place to improve the safety of the system by automatically scanning these data streams for possible problems and taking action, such as alerting the observer, if necessary. Consolidating and encoding data-taking procedures in software also makes it easier to design and implement useful graphical interfaces for these systems.

Furthermore, connecting together low-level subsystems into a higher-level framework can help allow interactions between these subsystems to be automated. For example, the trig-

ger mask must be set using the program on the Raspberry Pi before and after taking data with the modules. This logic does not belong solely to either the modules or backplane subsystems but involves them both. Safety or performance issues can also result from undesirable interactions between low-level subsystems. For example, data readout must be paused when reading the FEE temperatures, as that process introduces noise into the trigger path. Higher-level logic is needed to coordinate the behaviors of these systems.

6.3.2 Separation of Functions

Within the consolidated system, the execution of domain logic should be separated into discrete functions. This principle applies to the software's high-level architecture as well as to the implementation of specific low-level operations.

First, the main program and user interface should be separated, allowing the user interface to be implemented as a standalone GUI. Separating the GUI from the main program has several advantages. It ensures that an error related solely to the user interface does not harm the operation of the camera. It makes it possible to run more than one GUI at a time, and provides the option to run the GUI on a different computer from the main program. Isolating specialized GUI functions and the code controlling the hardware from each other makes the code simpler to write and maintain.

It is also advantageous to uncouple the parts of the code responsible for communicating with each hardware subsystem from the main program and from each other. These sections of code need not run as standalone programs but should at least be organized as distinct sections within the main program. Doing so enables the logical structure of the main program and the actual communication with hardware devices to be separated. This separation not only clarifies the code of the main program by allowing it to be written at a higher level of abstraction, but also supports the principle of modularity, discussed next.

6.3.3 Modularity

The software controlling every camera subsystem or device uses different libraries and communication protocols, depending on the requirements of the underlying equipment. However, the high-level logic of the camera's operation is independent of these low-level software details. The code controlling each device can be made modular by encapsulating these differing implementation details behind a common interface. A modular design simplifies the main program because all devices can be handled in a similar way. It makes it easier to augment the camera software by adding new devices. If a hardware or software upgrade alters low-level implementation details, only the portions of the camera software directly responsible for those details need to be modified. A caveat to this principle is that a hardware change may affect interactions between devices, as well as the operation of a single device. In this case, the high-level logic of the main program would also need to be modified.

6.3.4 Slow Control

The requirement to monitor and log the status of camera subsystems can be satisfied by developing slow control (SC) software for the camera. An SC system performs control and monitoring that is not necessarily time-critical and that serves to maintain the safety of the experiment or quality of the data. In addition to passively reporting the status of the camera on a continual basis, the SC system may actively respond to detected errors and malfunctions. In the system developed here, the only automatically triggered actions we consider are issuing alerts to the user, but we note that these actions could in general include anything, such as shutting down equipment if a safety limit is exceeded.

The SC system must support a GUI allowing the user to quickly view the status of all subsystems, including any alerts. The GUI must also allow the user to send commands to auxiliary camera subsystems to prepare the system for operation and shut it down when

observing is complete. The SC system must also enable monitoring data to be logged at an appropriate cadence. Every camera subsystem must by monitored and possibly controlled by the SC system.

6.3.5 Run Control

In addition to the slow control system, a run control (RC) system must be developed to control the data-taking operations of the camera using the main camera systems of the modules and the backplane. While the SC system continuously monitors the camera and only occasionally responds to a user command or triggers an automatic alert, by contrast, the RC system performs discrete (but possibly complex) sequences of operations based on high-level user commands, such as starting an observing run. The RC system must be compatible with the potential future requirement to implement an FSM, as discussed in Section 6.2.3, taking advantage of the principle of centralization.

The RC system must support a GUI that allows the user to quickly perform data-taking operations, including defining, starting, and stopping data runs and rate scans, while displaying the status of these operations to the user. The RC system must also provide support for logging of run information in formats suitable for both human operators and automated systems. Furthermore, the efficiency of data-taking operations can be improved in the framework of an RC system, for example, by allowing the software to maintain a record of the internal states of subsystems, rather than simply following procedural commands.

6.3.6 Camera Control Server

In accordance with the previously mentioned design principles, the camera software's main program should be implemented as a control server[7] that maintains a constant con-

[7]In this context, "server" refers to a program that receives and handles communication from other programs, called "clients", as opposed to the computer on which the server program runs.

nection with the devices listed in Section 6.1.2 using a modular framework. The RC and SC GUIs should be implemented as separate clients of this server. The control server is responsible for executing data-taking operations and for communicating with, automatically monitoring, and logging the status of the camera hardware.

Because the SC and RC systems require separate functionality and have only a limited overlap in the devices they control, an alternative approach, which was initially adopted when designing `sctcamsoft`, would be to implement them as two fully independent systems, each with a separate main program implemented as a control server. This approach creates several potential problems:

- **Control conflicts:** The TargetDriver library, which performs low-level operation of the modules, is required for both data-taking operations and slow control monitoring. However, the module software is designed such that only one program can possess the connection to and control of the modules at any given time. Independent RC and SC servers cannot both control the modules, leading to potential interruptions when taking data or gaps in monitoring and logging.

- **State machine:** Although the camera software does not define an FSM, it should be designed with this future requirement in mind. Defining the camera state involves both the run state (run control) and the hardware state (slow control). Neither a separate RC nor SC server alone would be able to define the full camera state.

- **Complexity and duplication of effort:** A system with two separate servers able to solve the above problems, for example, by having a third server to manage the first two, would risk becoming overly complex and unstable.

Having a single control server to perform both RC and SC functionality and accessing it with separate user interfaces avoids these issues.

6.3.7 Configuration and Logging

The camera software will need to interact with external data storage systems in at least three ways. First, data files are generated and stored during data taking. This task is fully handled by the `TargetIO` library and not considered further here. Second, the control server may require external configuration to set up and operate hardware. Third, the system must be able to record logging information in an external location. The data storage system used may be a database, or simply text files, for ease of use during commissioning.

6.3.8 Command-line Interface

In order to test the software during development and to provide a fail-safe option during normal operations, a CLI backup must be provided in addition to any GUIs. A CLI capable of supporting SC and RC operations has therefore been provided with the software.

6.3.9 High-level Design

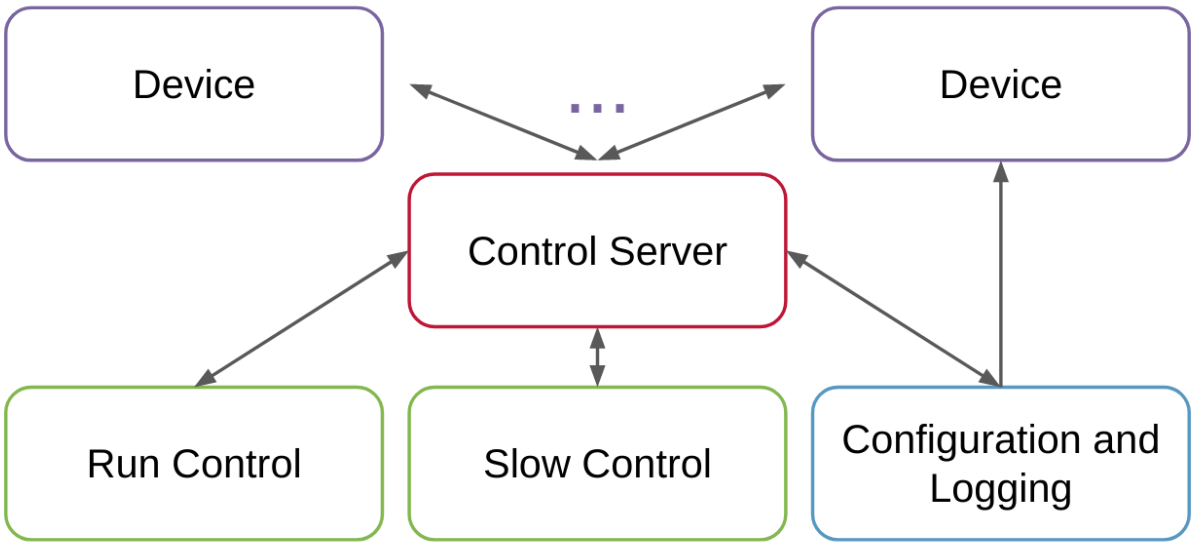


Figure 6.2: Diagram of the high-level design of the camera software.

Figure 6.2 shows the high-level design concept of the camera software which fulfills all of the principles discussed above. The domain logic of camera operation is implemented in a central control server. The user interacts with the server via specialized run control and slow control user interfaces. The actual operation of the telescope is performed for each device separately by modular code within the control server, which may depend on external configuration. The control server performs logging of run and monitoring data externally.

6.4 Software Implementation

6.4.1 `sctcamsoft` Package

The new camera control software has been implemented as a Python package called `sctcamsoft`. Using an object-oriented framework, each main element of the design is implemented as an object (generally a Python class) able to define methods, or object-specific functions; to contain data as well as other objects; and to interact with other objects in a defined manner. The code was developed and tested using Python 3.7.4. Instructions for installing and running `sctcamsoft` are given in Appendix D.

In addition to the Python standard library, `sctcamsoft` depends on several external libraries. First, data serialization for transfer between instances of the CameraControlClient class (Section 6.4.3) and the ServerController class (Section 6.4.4) is performed using the Protocol Buffers library[8]. The objects generated using this library for transferring commands and information are described in Section 6.4.2.

CTA collaborators have developed SC and RC GUIs compatible with the framework described here, which have been added to the `sctcamsoft` package. These programs are based on the PyQt5 library[9] for GUI programming and additionally rely on the Matplotlib library (Hunter, 2007) for plotting.

[8] `https://developers.google.com/protocol-buffers`
[9] `https://www.riverbankcomputing.com/software/pyqt/`

Configuration and logging are done using text files in comma-separated values (CSV) and YAML[10] formats. These files are parsed using the *pandas* (McKinney, 2010; Reback et al., 2020) and PyYAML[11] libraries. While not integrated into `sctcamsoft` proper, the automatic run log generator described in Section 6.6 depends on the Astropy (Astropy Collaboration et al., 2013, 2018) and Astroplan[12] libraries for astronomical computations.

The code used to control the camera subsystems is heavily based on pSCT Toolkit and depends on the same libraries described in Section 6.1.2. Since pre-existing code did not exist to connect to the chiller sensor, new code using the Requests library[13] was written to connect to this device. NumPy is used throughout the software for performing array-based numerical computation (Harris et al., 2020).

In Sections 6.4.2 through 6.4.6, we discuss each of the major classes implemented in `sctcamsoft`. Diagrams are provided in Unified Modeling Language (UML) format that visually represent each class as a box listing its attributes and methods. Class inheritance is indicated by an open-headed arrow between two boxes, while a filled, diamond-headed arrow indicates that one class is an attribute of another. All of the UML diagrams in this work were created using pyreverse, a UML diagram generator shipped with the Pylint package[14].

6.4.2 Command and Update Objects

The `sctcamsoft` software is built around a two-way, asymmetric communication process. The user sends commands to ServerController class (Section 6.4.4) running the control server through the CameraControlClient class (Section 6.4.3) embedded in the user interface[15].

Two types of user commands are available. A "low-level" user command has a one-to-

[10] A recursive acronym standing for "YAML Ain't Markup Language".
[11] `https://pyyaml.org/`
[12] `https://astroplan.readthedocs.io/en/latest/`
[13] `https://requests.readthedocs.io/en/master/`
[14] `https://github.com/PyCQA/pylint`
[15] As described in Section 6.4.4, "user" commands may also be initiated autonomously within the server.

one correspondence with a single command sent to a particular device. A "high-level" user command consists of an ordered list of user commands, which may be either low-level or other high-level commands. Upon receiving a high-level command, the server resolves it recursively into a list of low-level commands. All commands are defined in a YAML configuration file, `commands.yml`. The commands are defined in a language-independent format, as opposed to a Python module, to allow a GUI compatible with `sctcamsoft` to be created using a language other than Python, although this has not been done in practice. Appendix E lists all low-level commands defined in `commands.yml`.

Upon receiving a user command, the server breaks it down into one or more low-level commands to send to individual devices. A device may return an update in response to a command. The server then collects all device updates and returns them to the client in a batch. Table 6.1 lists the objects defined to transmit commands and updates at each step of this process.

Sender	Receiver	Object	Format
CameraControlClient	ServerController	UserCommand	message
ServerController	DeviceController	DeviceCommand	namedtuple
DeviceController	ServerController	Update	class
ServerController	CameraControlClient	UserUpdate	message

Table 6.1: Objects used for sending and receiving commands and updates in `sctcamsoft`. Objects in the "message" format are defined using the Protocol Buffers library and those in the "namedtuple" format are defined using `collections.namedtuple` from the Python standard library.

The UserCommand and UserUpdate objects are Protocol Buffers messages that encode Python data structures into an efficient binary format. To provide maximum flexibility, these objects encode all fields as strings, except where otherwise noted. UserCommand stores a command name and an arbitrary number of pairs of argument names and values. UserUpdate encodes a list of updates, where each update stores a device name, variable name, and unit, along with an arbitrary number of (value identifier, value) pairs and an

encoded timestamp[16]. In this way a UserUpdate can return updates from multiple devices and each device update can in turn hold multiple values. This design allows devices that have multiple components to report updates.

The server transmits commands to the DeviceController (Section 6.4.5) that controls each device using a DeviceCommand object storing the device name, the command name, and a dictionary of argument (name, value) pairs. All values are stored as strings. The relevant DeviceController may cast values to another type, such as a numerical type, if needed.

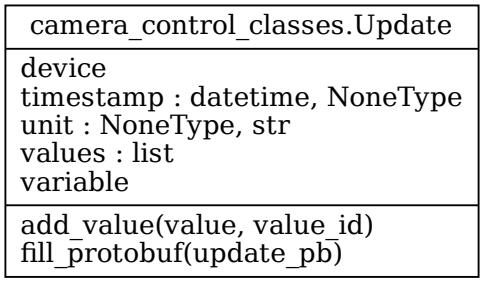


Figure 6.3: Diagram of the Update class in `sctcamsoft`. The timestamp has the `datetime.datetime` type from the Python standard library.

After executing a command, a device may return an update back to the control server using the Update class. A diagram of this class is given in Figure 6.3. The attributes of an Update are the device name, the variable name, a unit, a timestamp, and a list of (value identifier, value) pairs. If the Update only contains one value, the value identifier may be omitted. Values can be added to the list of values, along with an optional identifier, using the `add_value()` method. The `fill_protobuf()` method dumps the contents of the Update into a UserUpdate Protocol Buffers message.

[16]Stored as a `google.protobuf.Timestamp`: `https://developers.google.com/protocol-buffers/docs/reference/google.protobuf/#timestamp`

6.4.3 CameraControlClient

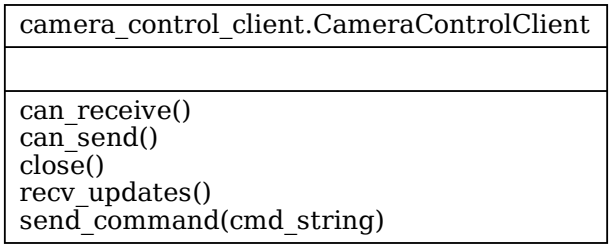


Figure 6.4: Diagram of the CameraControlClient class in `sctcamsoft`.

The CameraControlClient class is responsible for communication between the user interface and the control server. CameraControlClient can be used as part of either a CLI or GUI program. Multiple CameraControlClient instances may communicate with a single Server-Controller. The implementation relies on two modules from the Python standard library, `socket` for low-level networking and `shlex` for parsing input in a shell-like interface. Two ports are used for network communication, one for the client to input commands and another for the server to broadcast updates to the clients. Because CameraControlClient serializes data using the language-neutral Protocol Buffers framework and it relies on low-level socket communication for networking, its functionality is largely language-independent.

Figure 6.4 lists the methods belonging to the CameraControlClient. Socket communication with the server is set up when the class is initialized. The `can_send()` and `can_receive()` methods indicate whether the client is configured to send commands and to receive updates. The `close()` method closes the network connection to the server. The `send_command()` and `recv_updates()` methods send commands to and receive updates from the control server in the formats described in Section 6.4.2.

Two simple CLI programs are provided with `sctcamsoft`, called `user_input.py` and

`user_output.py`, which are based on CameraControlClient. Separate programs are used for input and output when using the CLI so that continual streams of monitoring data can be printed to the screen without disrupting the ability of the user to enter commands. The SC and RC GUIs mentioned in Section 6.4.1 are also built around CameraControlClient.

6.4.4 ServerController and Its Component Classes

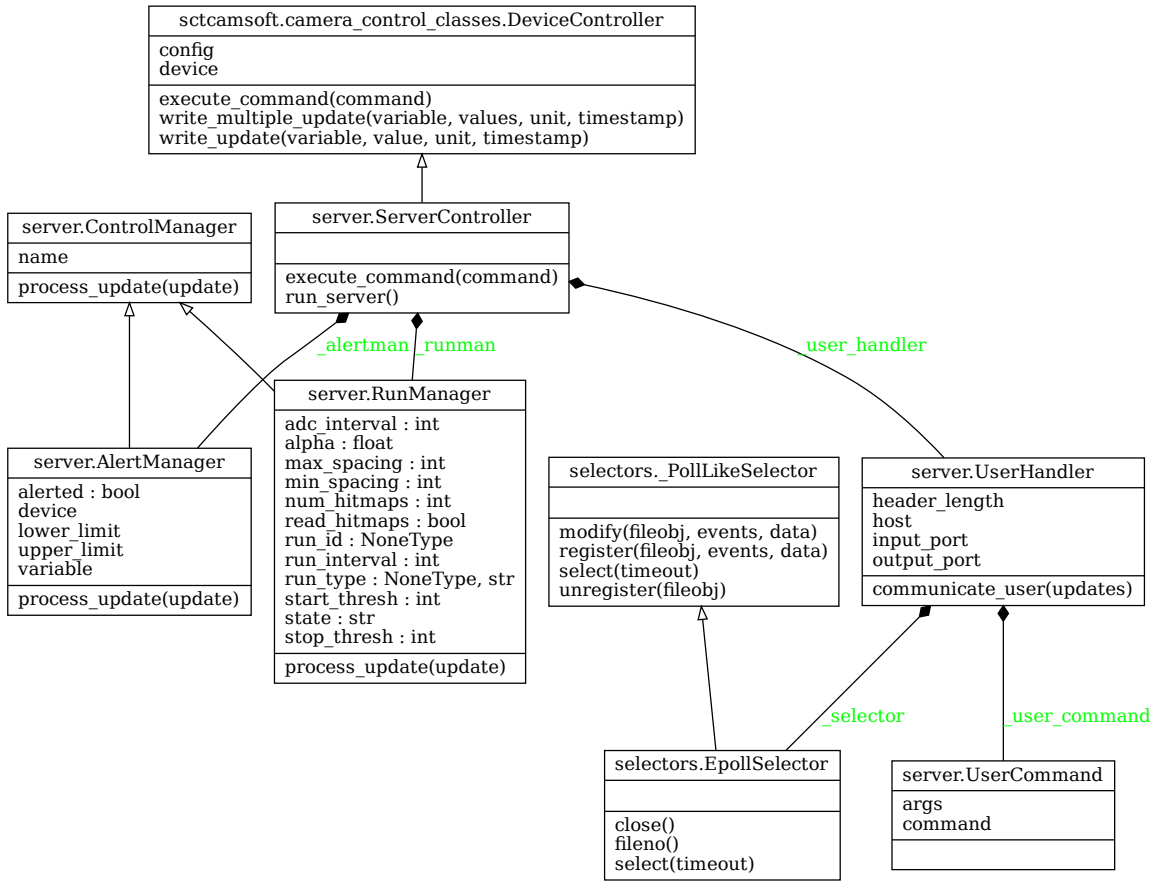


Figure 6.5: Diagram of the ServerController class and its component classes in `sctcamsoft`. While the actual UserHandler uses the `selectors.DefaultSelector` class from the Python standard library which provides the most efficient implementation for a given platform, for clarity, the diagram shows a specific implementation, `selectors.EpollSelector`.

ServerController

The ServerController class is the core of `sctcamsoft`. Figure 6.5 shows a diagram of ServerController as well as its component classes. As described in more detail below, ServerController contains a UserHandler instance, which handles communication with the client user interfaces; a RunManager instance, which implements run control functionality; and any number of AlertManager instances, each of which manages an alert on a particular variable. RunManager and AlertManager are derived classes of the more general ControlManager. In addition to those classes, ServerController contains a DeviceController subclass for each device and a dynamically modified list of commands to be executed on automatic timers, implemented using the `threading.Timer` object from the Python standard library.

The `run_server()` method of ServerController implements the server's main loop, which runs continuously until the program is ended. ServerController performs the following steps on each iteration of the main loop:

1. Package and send the list of all updates to the client user interfaces and receive in return up to one user command.

2. Clear the list of updates.

3. Resolve the user command (if one was received) into a list of device commands, as described in Section 6.4.2.

4. Refer each device command to the proper DeviceController for sequential execution, and in response to each command, add up to one update to the list of updates.

5. For each timer that has expired, resolve and execute its corresponding command as in Steps 3 and 4, and reset the timer if specified to do so.

6. For the RunManager and each AlertManager, examine every update in the list and potentially trigger a command based on its value. Resolve and execute the command as in Steps 3 and 4, if applicable.

ServerController derives from DeviceController and therefore emulates a device itself. It executes commands using the `execute_command()` method. This design allows the server to set alerts, send messages, and issue automatically repeating commands using the same framework of commands and updates as any other device.

UserHandler

ServerController communicates with its user interface clients through its component UserHandler class. UserHandler is designed to be compatible with CameraControlClient. In the ServerController main loop, Step 1 is performed by `UserHandler.communicate_user()`. The server functionality of UserHandler is built around the `selectors` module from the Python standard library. Serveral configuration settings are required to be set, including the host, input port, and output port. A simple header encoding the length of the message in bytes is appended to each message; the length of this header can be set as a configuration option. This header helps ensure complete messages are received, as messages may be split unpredictably into packets of different sizes when sent over the network. UserHandler also defines a `server.UserCommand` class, into which the contents of the UserCommand object described in Section 6.4.2 are copied. This is done to keep classes that are dependent on the Protocol Buffers library isolated to the networking portions of the code.

ControlManager

Step 6 of the ServerController main loop allows it to initiate actions autonomously. All autonomous actions are performed by subclasses of the ControlManager base class. On each

iteration of the main loop, a ControlManager instance calls its `process_update()` method on every update before it is sent to the user, which may cause a sequence of device commands to be performed, depending on the contents of the update and the internal state of the ControlManager. Each ControlManager has a name for identification by the ServerController. Two ControlManager subclasses are defined in `sctcamsoft`: AlertManager and RunManager.

AlertManager

An AlertManager checks the value of a specified variable on a particular device to determine whether it falls within specified lower and upper limits. Only variables capable of being cast into a floating-point number (Python `float`) are compatible with AlertManager. If an update violates the limits, the AlertManager causes the ServerController to issue an alert update, and the AlertManager enters an alerted state. While in the alerted state, subsequent limit-violating updates will not generate further alerts. If a subsequent update is examined that respects the limits, the AlertManager will exit its alerted state and cause the ServerController to issue an update to clear the alert. The server may set, unset, or modify an alert by creating, deleting, or modifying the corresponding AlertController.

RunManager

The server's run control functionality is performed by a RunManager. Internally, RunManager operates as an FSM. It maintains an internal state, a list of allowed states, and a list of allowed transitions between those states. Note, however, that this behavior is considered an implementation detail and is not publicly guaranteed. It should also be noted that the server as a whole does not operate as an FSM. A state diagram illustrating the allowed states and transitions is shown in Figure 6.6.

[17] Diagram made with Creately, `https://creately.com/`

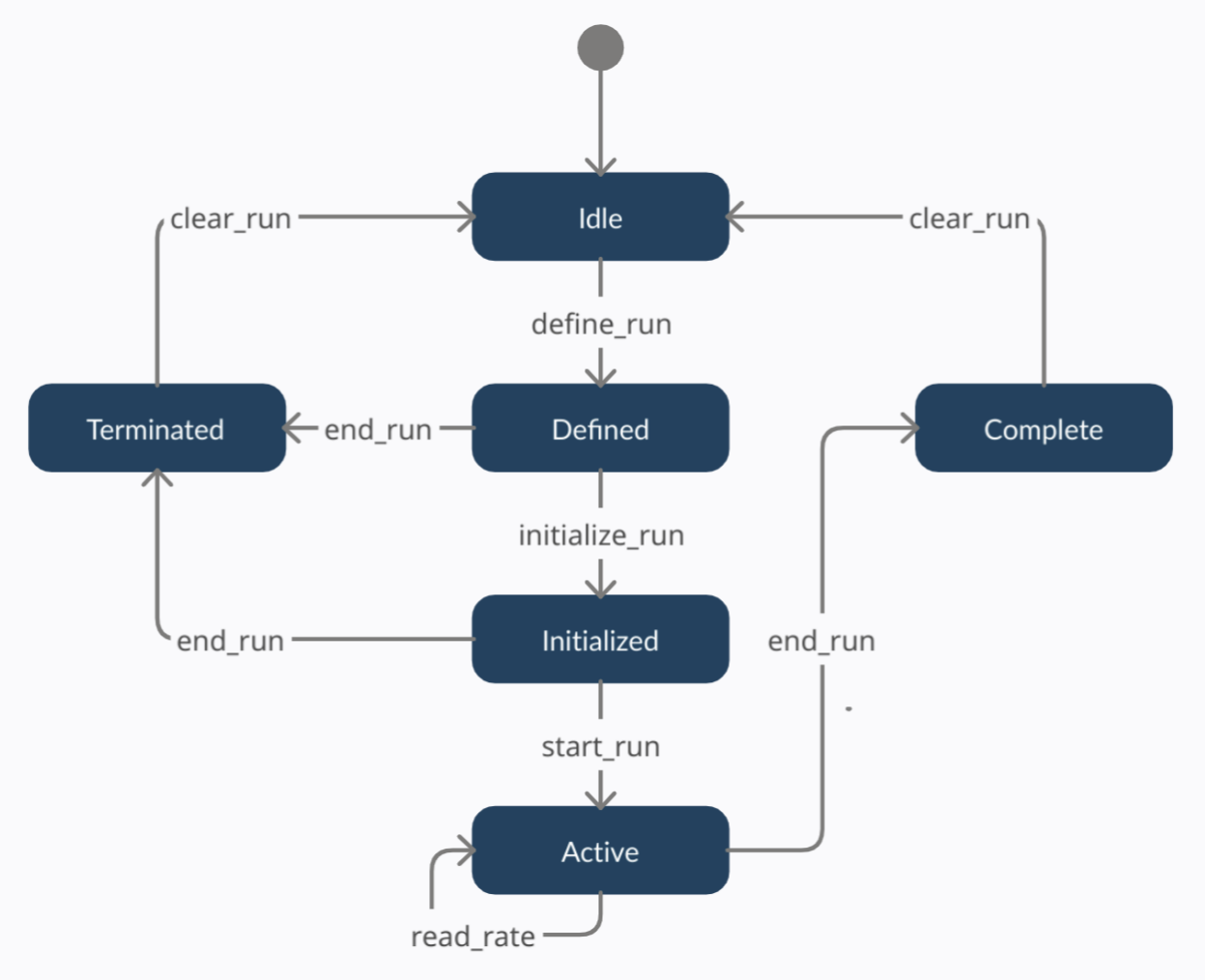


Figure 6.6: State diagram of the RunManager in `sctcamsoft`[17]. The state diagram is identical for data runs and rate scans, with the exception that the `read_rate` action applies only during rate scans.

A RunManager may exist in one of six states: Idle, Defined, Initialized, Active, Complete, and Terminated. The initial state is Idle, in which the system is not performing anything related to a run. From this state, the `define_run` action brings it to the Defined state in which a run number is defined, but no hardware systems have been initialized. The `initialize_run` action tells the system to initialize the modules to perform either a data run or rate scan, and brings the RunManager to the Initialized state. Once initialized, the run is started with the `start_run` action.

If the run is a rate scan, the `read_rate` action is performed repeatedly during the Active state to instruct the system to read the next rate. This action is initiated autonomously. The rates to scan and duration to collect trigger counts at each threshold are determined using the procedure described in Section 6.5.

The `end_run` action ends the run, transitioning it from the Active to the Complete state. This action is initiated autonomously when the run duration has elapsed, but can also be commanded manually to end the run early. The `end_run` action can also be performed in the Defined or Initialized states, in which cases the system transitions to the Terminated state instead. In the Terminated state, the settings of the ended run are available, while in the Complete state, the results of the data run or rate scan are available as well. From either state, the `clear_run` action transitions the system back to the Idle state so that a new run can be defined. This simple FSM has no built-in capability to transition to an error state and therefore relies on the user to manually end a run if a problem is observed. The state machine models only the high-level framework of taking runs, as the mechanics of initializing and operating modules and collecting data are encapsulated in the `TargetDriver` and `TargetIO` libraries wrapped by TargetController (Section 6.4.5).

6.4.5 DeviceController

Following the principle of modularity articulated in Section 6.3.3, each camera subsystem (or "device") described in Section 6.1.2 is controlled by a class following a consistent modular format defined by inheritance from the DeviceController base class. Figure 6.7 shows a diagram of this class and each of its subclasses implemented in the camera software.

Each DeviceController is initialized with a string representing the device name and a dictionary of device-specific configuration parameters. A DeviceController is required to implement the `execute_command()` method, which accepts a command specified as a DeviceCommand object, executes that command, and optionally returns an Update object containing the results (see Section 6.4.2). DeviceController provides `write_update()` and `write_multiple_update()` convenience methods for constructing these Update objects. The commands implemented by each DeviceController subclass match those defined in `commands.yml`, and are listed in Appendix E.

Each DeviceController subclass is implemented by refactoring, rewriting into Python, or, where feasible, directly importing the existing low-level code that controls its respective subsystem, described in Section 6.1. Except where otherwise noted, the attempt has been made to preserve all essential functionality of the pSCT Toolkit software.

The refactoring into the DeviceController framework provides several advantages. Large functions that perform complex operations are split into individual commands that each perform a single task, with operations for different devices isolated in different classes, making the code easier to maintain, extend, and test. In addition, DeviceController subclasses may contain internal variables representing the hardware state and desired configuration, allowing for recognition of invalid commands and efficient implementation of operations.

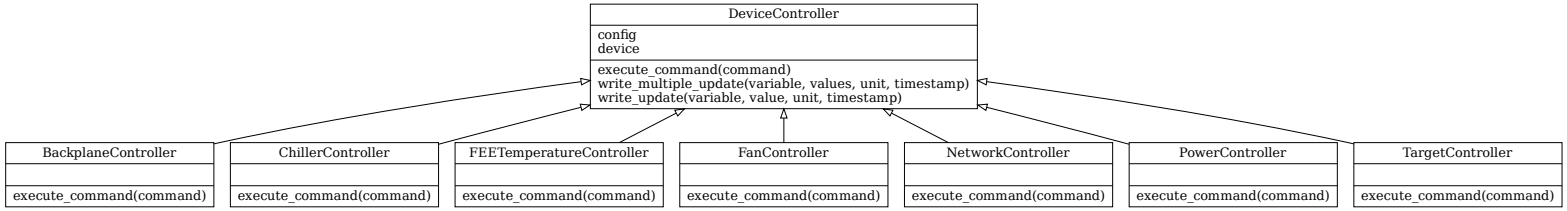


Figure 6.7: Diagram of the DeviceController subclasses in `sctcamsoft`. ServerController, although it is also a DeviceController, is shown separately in Figure 6.5.

In addition to the DeviceController subclasses described below, which are meant for use in real operations, `sctcamsoft` also provides a matching suite of "mock" DeviceController subclasses for testing purposes.

BackplaneController

BackplaneController is responsible for communicating with the backplane via the Raspberry Pi software. Commands are available to set the values of several settings on the backplane needed to initialize it for data taking, such as opening and closing the trigger mask. Modules are also powered through BackplaneController, which provides commands to power on and off individual modules as well as convenience commands to power them all on sequentially[18] and to power them all off again. Commands are provided to monitor the currents, voltages, and presence of the modules. The most recent trigger hit pattern can be read out from the backplane, as can the count of Trigger Acknowledge (TACK) messages received and the value of the backplane's internal 1 ns precision timer. BackplaneController provides commands to reboot the two DACQ boards.

ChillerController

ChillerController provides two commands that read out the temperature and pressure from the chiller sensor, respectively. This class performs only monitoring, with no control capabilities.

FanController

FanController provides commands to control and monitor the power supply of the camera fan. Commands are provided to open, close, and check the status of the connection; turn the power on and off; read the power supply current; and read the power supply voltage.

[18] Due to hardware limitations, the modules must be powered on one at a time.

FEETemperatureController

FEETemperatureController reads the module FEE temperatures from log files. It provides two commands, one to get the run number corresponding to the most recent log file and one to return the most recent temperature reading from the log file of a given run number.

FEETemperatureController, unlike the other DeviceController subclasses, exists only for convenience, as it duplicates functionality that could be performed by TargetController, described below. It is useful in practice to support an intermediate mode of operation in which pSCT Toolkit is used for run control and `sctcamsoft` for slow control. In this case, active temperature monitoring with TargetController would not be possible, but FEETemperatureController could still be used to passively monitor the FEE temperatures from the log files written out by pSCT Toolkit, with which it is compatible.

NetworkController

NetworkController is responsible for setting up and monitoring the network connections between camera subsystems. It allows the activity on any one of the four network interfaces connected to the two DACQ boards to be monitored[19]. There is a command to set up the IP addresses corresponding to the modules on the camera server computer. Finally, a command is provided to set up the IP addresses of the modules on the DACQ boards so that packets are correctly forwarded.

PowerController

PowerController controls and monitors the power supply for the SiPMs and camera electronics. Commands are provided to turn the main switch on and off; to start and stop the supply and HV power; to read the supply and HV currents; to read the supply and HV set

[19]Interfaces 6 and 7 connect to DACQ board 1 and interfaces 8 and 9 connect to DACQ board 2.

(nominal) voltage; and to read the supply and HV actual (measured) voltage.

TargetController

TargetController, named after the TARGET ASICs which are the key components of the modules, is responsible for controlling and communicating with the modules. It uses the `TargetDriver` and `TargetIO` libraries to connect to, initialize, and tune the modules. It provides commands for taking data and writing the results to a FITS (Pence et al., 2010) file, and for performing rate scans. It also provides monitoring of the FEE temperatures and pixel currents.

TargetController attempts to improve on pSCT Toolkit by maintaining the connection to the modules so that the setup needs only to be done once, at the start of a data-taking session; performing operations on multiple modules in parallel, when possible; and automatically scaling the duration when reading the trigger rate in order to achieve a constant precision, as described in Section 6.5. Note, however, that not all capabilities implemented in pSCT Toolkit have been incorporated into TargetController, such as the automatic run log generator described in Section 6.6.

TargetController implements many commands. Among others, commands are available to initialize, ready, and ping the modules; to request new values of ASIC, readout, tuning, ADC, and rate scan parameters; to set the requested ASIC, tuning, and readout parameters on each module[20]; to get the values of the parameters that have been set; to read the temperatures and currents from each module; to enable and disable data readout or triggering from particular modules; and to initialize a run, write the trigger mask, begin and end readout, read the trigger rate, and get the run result when complete. The full list of provided commands is given in Appendix E.

[20]The ADC and rate scan parameters are software variables only and do not need to be set in the hardware.

6.4.6 CameraControlError

A DeviceController may encounter errors in the course of its initialization and operation. DeviceController subclasses are required to handle errors by raising a CameraControlError, a custom Python Exception. ServerController handles any CameraControlError by transmitting an Update containing the error message to its clients and printing the stack traceback to the screen. Like DeviceController, CameraControlError is a base class from which a number of specialized subclasses are derived. Figure 6.8 shows a diagram of the CameraControlError base class and each of its subclasses implemented in `sctcamsoft`. Every CameraControlError instance has as attributes the name of the device that generated it and an informative message for the user, which varies depending on the subclass of CameraControlError.

A CommandArgumentError is raised if an argument to a command is invalid or missing. A CommandExecutionError is raised if an error is encountered during command execution. If a DeviceController receives a command with a name that does not match any valid command, a CommandNameError is raised. A CommandSequenceError is raised if a valid command is received, but in an unsupported order given the status of the device. For example, a command to open a network connection would cause this exception to be raised if the connection is already open. A CommunicationError is raised for errors communicating with a device. If a DeviceController encounters missing or invalid configuration parameters during initialization, it raises a ConfigurationError. If the server receives a command for an invalid device, a DeviceNameError is raised. Finally, if a variable is found to have an invalid type or value during the operation of a DeviceController, a VariableError is raised.

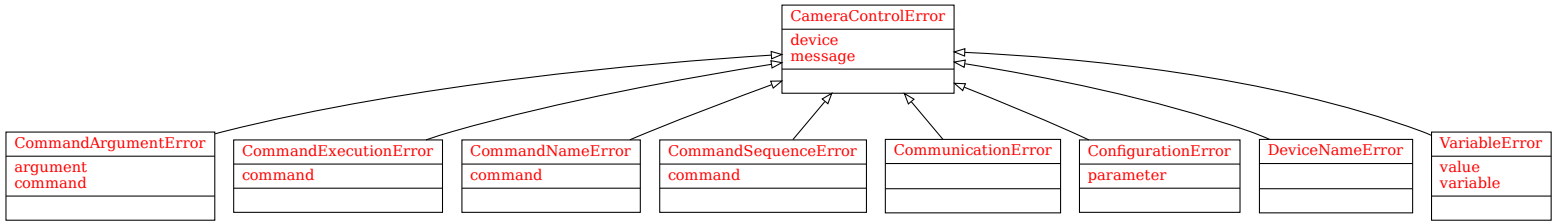


Figure 6.8: Diagram of the CameraControlError classes in `sctcamsoft`.

6.5 Rate Scan Optimization

As discussed in Section 6.1.1, during the commissioning of the pSCT, a rate scan must be performed to determine the temperature-dependent trigger threshold before each data run. It is important to optimize both the speed and precision of this frequent, time-intensive operation. This can be accomplished by eliminating unnecessary operations, clustering scanned thresholds on the most interesting parts of the rate curve, and optimizing the duration spent at each threshold to achieve equal precision at all thresholds.

A rate scan as performed using pSCT Toolkit takes approximately 14 minutes, divided into about 8 minutes to set up and tune the modules[21] and 6 minutes to perform the rate scan itself. TargetController has been designed to greatly reduce each run's setup time by initializing the modules at the start of the observing session and setting up and tuning the modules in parallel.

6.5.1 Eliminating Unnecessary Operations

In rate scans conducted with pSCT Toolkit, the module temperatures and a sample of trigger hit patterns are read out at each threshold. Since these operations dominate the time to read the rate in pSCT Toolkit[22] and are not required for data-taking operations, they have been eliminated in `sctcamsoft`. Instead, the module temperatures are read out once each at the start and end of the rate scan.

[21] This time includes five minutes spent idle after the system is initialized, as the module electronics only work properly once they have warmed up to their normal operating temperature.

[22] At each threshold, as a rough approximation assuming typical settings, pSCT Toolkit takes about 6 seconds to read the module temperatures, 5 seconds to read the trigger hit maps, and 1 second to read the rate.

6.5.2 Optimizing Threshold Spacing

Quickly-changing transition regions in the rate curve in should be scanned densely, while stable plateau regions should be scanned coarsely in order to save time. While pSCT Toolkit allows the user to scan over a specified list of unevenly spaced thresholds, optimizing the threshold spacing manually is impractical because the locations of breaks in the rate curve are not necessarily known ahead of time. Instead, a "smoothed-derivative" algorithm has been developed for `sctcamsoft` to accomplish this goal automatically.

In the smoothed-derivative algorithm, the spacing between each consecutive threshold is computed dynamically based on the absolute value of the derivative of the rate with respect to threshold. When the absolute value of the derivative is small, the spacing is large, and when it is large, the spacing is small. Particularly considering the optimized rate measurement duration (discussed next), the measured rates tend to vary smoothly and close to monotonically with the threshold. The derivative can therefore be estimated locally as the difference between the current and previous rate, divided by the difference between the current and previous threshold. This local estimate of the derivative Δ is converted to a spacing s using a sigmoid smoothing function σ,

$$s = (s_{\max} - s_{\min})\sigma(\Delta) + s_{\min} \tag{6.1}$$

where the range of σ is (0, 1), and $s_{\min}$ and $s_{\max}$ (called *min_spacing* and *max_spacing* in `sctcamsoft`) determine the asymptotically approached minimum and maximum spacings.

The sigmoid smoothing function is defined as a function of $x \equiv -\alpha \ln |\Delta|$ as

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{1}{1 + |\Delta|^{\alpha}} \tag{6.2}$$

where in the definition of x, the natural logarithm transforms the domain of the absolute

value of the derivative from $(0, \infty)$ to $(-\infty, \infty)$, the negative sign produces a large spacing at small $|\Delta|$ and vice-versa, and the α parameter (*alpha*) determines how sensitive σ is to the value of $|\Delta|$.

Finally, the *start_thresh* and *stop_thresh* parameters[23] determine the thresholds at which to start and stop the rate scan. In the implementation in `sctcamsoft`, the final threshold scanned is allowed to overshoot *stop_thresh*.

The smoothed-derivative algorithm has five configuration parameters (default values in `sctcamsoft`): *start_thresh* (600 DAC), *stop_thresh* (0 DAC), *min_spacing* (10 DAC), *max_spacing* (50 DAC), and *alpha* (1.5). The smoothed-derivative algorithm automatically accommodates any number of transition regions, at any locations. By way of comparison, obtaining a similar result manually for a rate scan with a structure consisting of a single transition region surrounded by two slowly-varying "plateaus" would require six configuration parameters (*start_thresh*, *stop_thresh*, two thresholds to demarcate the transition region start and stop, the plateau spacing, and the transition spacing), of which the transition region start and stop thresholds could not be optimized without adjusting them by trial and error for every rate scan. In fact, rate scans performed during the commissioning of the pSCT with the flasher running may have multiple transition regions.

6.5.3 Optimizing Duration at Each Threshold

In order to reduce variability in the measurements at low trigger rates while minimizing the time required to scan thresholds with high trigger rates, the duration spent measuring the rate at each threshold can be varied to achieve a constant precision at each threshold.

The observed number of trigger counts N at a given threshold can be modeled as having a Poisson distribution where the parameter μ is the expected number of counts,

[23] As mentioned in Section 6.1.1, rate scans with the pSCT camera are actually performed from high to low *thresh* values, where a high *thresh* is equivalent to a low threshold and produces a high rate, and vice-versa.

$$f(N;\mu) = \frac{\mu^N e^{-\mu}}{N!} \tag{6.3}$$

The Poisson distribution has variance σ^2 equal to μ, so the standard error on the expected number of counts is $\sqrt{\sigma^2} = \sqrt{\mu}$. We are interested in the expected rate of trigger counts $r = \mu/t$, where t is the measurement duration. The standard error on the expected rate is then $\sigma_r = \sqrt{\mu}/t$. Let the precision $p = \sigma_r/r$. We then have

$$p = \frac{\sigma_r}{r} = \frac{\sqrt{\mu}/t}{\mu/t} = \frac{1}{\sqrt{\mu}} \tag{6.4}$$

Rearranging and substituting Eq. 6.4 into the definition of expected rate yields

$$r = \frac{\mu}{t} = \frac{1}{p^2 t} \tag{6.5}$$

Rearranging Eq. 6.5, we obtain expressions for the expected number of counts μ and the required duration t,

$$\mu = \frac{1}{p^2}, \tag{6.6}$$

and

$$t = \frac{1}{p^2 r}. \tag{6.7}$$

The required precision p is a constant which is set as a configuration parameter. The expected trigger rate r can be estimated using the rate at the previously measured threshold. Because the trigger rate during a rate scan decreases close to monotonically and, for reasonable choices of spacing parameters, relatively slowly, using the previously measured rate as an approximation will slightly underestimate the required duration. In practice, the

resulting precision will therefore be slightly worse than the required precision, which should be recognized when setting that parameter. The rate to assume at the first threshold scanned is a configurable parameter in `sctcamsoft` with a default value of 5000 Hz.

The expected number of counts required to achieve a given precision at any threshold is independent of the trigger rate. To achieve 10% precision, for example, 100 trigger counts are needed at each threshold. In a typical pSCT rate scan, the measured trigger rates range from a maximum of ∼5000 Hz down to the rate of triggers produced by the flasher (if it is running), typically 10 Hz. At 5000 Hz and 10 Hz, measurement durations of 0.02 s and 10 s are required to achieve 10% precision. To avoid having excessively long measurement durations at the lowest trigger rates, `sctcamsoft` includes a configurable timeout parameter that sets a maximum duration to record trigger counts at each threshold.

6.6 Automatic Run Log

In addition to the development of `sctcamsoft`, a software system to automatically generate a nightly run log has been implemented in pSCT Toolkit. Integrating the automatic run log generator with `sctcamsoft` and connecting it to a database should be the subject of future work. The system records information about all runs in an observing session (usually a night) and processes it into a text format suitable for copying into a written log.

When starting a run, the user is prompted to record the status of several camera subsystems, including the chiller, camera fans, shutter and flasher. The previous run's settings are saved as defaults. Manual entry is required because these subsystems are not monitored in pSCT Toolkit. When a run ends, its settings and results are logged to a compressed file using the `pickle` module from the Python standard library.

Before taking any runs in an observing session, the user is required to run a script called `start_night.py`, which records a timestamp corresponding to the start of observations.

- Rate Scan 330013 at 2021-01-28 04:07:56.779066 UTC (21:07:56 local)
  - Backplane clock started at 2021-01-28 04:02:17.678642 UTC
  - modules 6, 7, 100, 107, 111, 112, 114, 115, 123, 124; triggering on modules 6, 7, 100, 107, 111, 112, 114, 115, 123, 124
  - masked trigger pixels (100: 4, 6, 9, 11, 12); (107: 6, 9, 11); (111: 9, 12); (112: 3, 9); (115: 2, 3, 4, 6, 7, 10, 11, 14, 15); (124: 4, 9, 12); (123: 3); (6: 1, 4, 8, 9, 11, 14); (7: 1, 3, 4, 6, 8, 9, 11, 12, 14)
  - loaded tuning temps (1: 22); (2: 13); (3: 14); (4: 14); (5: 16); (6: 16.5); (7: 17.15625); (8: 14); (9: 13); (100: 17.5625); (103: 14); (106: 13.5); (107: 19.9375); (108: 19); (111: 21.5); (112: 19.8125); (114: 18.75); (115: 17.90625); (119: 21); (121: 18); (123: 20.59375); (124: 18.25); (125: 12); (126: 12)
  - target: NONE, RA: 228.989868, Dec: 53.375271 (shutter open, HV on), 99.46% moon
  - top flasher on at 10 Hz with LED pattern 1111111111
  - middle flasher on at 10 Hz with LED pattern 1111111111
  - bottom flasher on at 10 Hz with LED pattern 1111111111
  - chiller off, fans on, FEEs at 23.62 C avg / 17.25 C min / 31.25 C max
  - thresholds scanned from 1200 to 50 DAC
  - 20 Hz rate (added flasher 10 Hz) at about 502 DAC

Figure 6.9: An example of an automatically generated log for a rate scan.

After completing the observing session, the run log is generated by running a script called `write_confluence_log.py`[24]. Only runs taken after the start-of-night timestamp are included in the log.

Figures 6.9 and 6.10 show examples of automatically generated logs for a rate scan and data run. Each log records the run type, run number, and timestamps at which the run was defined and data readout was initialized. The module numbers of modules enabled to record data and to produce triggers in the run are recorded, using user-friendly abbreviations such as "all modules" if all modules are enabled. A note is made if any modules failed to connect. For each module, the list of trigger pixels being masked (if any) and the temperature for which its tuning parameters have been optimized are recorded.

If the shutter is open, the target name, right ascension, and declination are recorded. If the Moon is up, the Moon illumination percentage is recorded, and if the Sun is up, it is noted that the run occurred during the day. The status of camera subsystems including the

[24]The output of the script is formatted to have appropriate markup for copying into an Atlassian Confluence workspace: `https://www.atlassian.com/software/confluence`

- Run 330014 at 2021-01-28 04:22:46.425876 UTC (21:22:46 local)
  - Backplane clock started at 2021-01-28 04:16:21.827794 UTC
  - all modules; triggering on modules 6, 7, 100, 107, 111, 112, 114, 115, 123, 124
  - masked trigger pixels (100: 4, 6, 9, 11, 12); (107: 6, 9, 11); (111: 9, 12); (112: 3, 9); (115: 2, 3, 4, 6, 7, 10, 11, 14, 15); (124: 4, 9, 12); (123: 3); (6: 1, 4, 8, 9, 11, 14); (7: 1, 3, 4, 6, 8, 9, 11, 12, 14)
  - loaded tuning temps (1: 26.179276315789473); (2: 19.90625); (3: 24.53125); (4: 26.179276315789473); (5: 26.1875); (6: 23.40625); (7: 25.625); (8: 20.78125); (9: 20.6875); (100: 26.34375); (103: 26.179276315789473); (106: 22.15625); (107: 32.15625); (108: 21.84375); (111: 34.78125); (112: 32.03125); (114: 30.96875); (115: 28.46875); (119: 19.96875); (121: 24.75); (123: 33.625); (124: 29.1875); (125: 26.179276315789473); (126: 26.179276315789473)
  - target: NONE, RA: 228.989868, Dec: 53.375271 (shutter open, HV on), 99.47% moon
  - top flasher on at 10 Hz with LED pattern 1111111111
  - middle flasher on at 10 Hz with LED pattern 1111111111
  - bottom flasher on at 10 Hz with LED pattern 1111111111
  - chiller off, fans on, FEEs at 28.32 C avg / 16.00 C min / 96.00 C max
  - 0:00:20 (~0:00:08.559746 livetime), 448 DAC thresh
  - 3936 packets received (0 lost) in 82 events (~9.6 Hz rate)

Figure 6.10: An example of an automatically generated log for a data run.

HV power, shutter, chiller, and camera fans are noted, as is information on the module FEE temperatures. If any of the three flashers is on, its rate and LED pattern are recorded.

For a rate scan, the range of thresholds scanned is recorded. The threshold at which a desired rate specified by the user is achieved is estimated from the measured thresholds using linear interpolation. If a plateau at approximately constant rate (such as produced by triggers from the flasher) is detected, that is noted as well.

For a data run, the run duration, estimated livetime, and threshold are recorded, where the livetime is the run duration minus the time that data-taking was paused in order to read the module temperatures. A note is made if certain non-standard run settings have been used. The numbers of data packets received and lost and of events read are recorded. Finally, the trigger rate is estimated by dividing the number of events read by the livetime.

6.7 Contributors to this Work

Software development for the pSCT is a broad, collaborative undertaking involving the work of numerous people. While this chapter is focused primarily on the author's contributions, several others have made important contributions to `sctcamsoft`.

- Colin Adams created the Flasher GUI mentioned in Section 6.1.2.
- Colin Adams, Brent Mode, and Leslie Taylor created and extended `psct_toolkit`, on which TargetController and BackplaneController are based.
- Qi Feng updated the automatic run log to report the tuning temperatures for each module and wrote the script it uses to obtain the right ascension and declination.
- Weidong Jin created the Slow Control GUI mentioned in Section 6.4.1.
- Thomas Meures wrote the Bash scripts on which PowerController and NetworkController are based.
- Brent Mode and Leslie Taylor took the runs used to produce Figures 6.9 and 6.10.
- Phil Moore and Richard Bose wrote the C program used by BackplaneController to communicate with the backplane.
- Jake Powell created the Run Control GUI mentioned in Section 6.4.1; developed the suite of "mock" DeviceController subclasses mentioned in Section 6.4.5; created CameraControlClient and ChillerController; and improved code throughout `sctcamsoft`.
- Marcos Santander coordinated overall development of the pSCT camera software.
- Justin Vandenbroucke wrote the Python script on which FanController is based and the module used within RunManager to assign run numbers.

Chapter 7: Applying Deep Learning to IACT Data Analysis

In this chapter, we develop a new analysis method applicable to all CTA telescopes, and indeed, to IACTs in general. We investigate deep learning, a type of machine learning using deep neural networks, that can make full use of rich data such as the high-resolution images produced by the SCT camera, and we apply this method to the problem of reconstructing particle shower events from IACT images. In Section 7.1, we begin by reviewing the fundamentals of machine learning and how it can be used to analyze IACT data. In Section 7.2, we introduce the concepts of neural networks and deep learning, and discuss in particular the deep neural network architectures studied in this work.

Next, we report on an exploratory study applying deep learning to simulated SCT data in Section 7.3, and in Section 7.4, we derive from that study a set of major challenges involved in applying deep neural networks to data from IACTs. In order to address these challenges, we developed two software packages, DL1-Data-Handler (Section 7.5), with which we tackle the challenge of processing hexagonally spaced IACT images (Section 7.6), and CTLearn (Section 7.7). Using CTLearn, we develop neural networks to perform stereo event reconstruction (Section 7.8), benchmark classification performance for all CTA telescopes (Section 7.9), and investigate the impact of telescope ordering when performing stereo event classification (Section 7.10). Finally, in Section 7.11, we discuss significant contributions made by others to the work reported in this chapter.

7.1 Machine Learning in the IACT Context

The traditional methods for classifying and reconstructing IACT events discussed in Section 3.4 are simple, robust, and easy to understand, but they are far from optimal. Box cuts, for example, cannot make use of nonlinear interactions between features or efficiently partition a high-dimensional feature space. Machine learning techniques can be used to improve on these methods. In the supervised machine learning approach, an algorithm learns from a set of training data to automatically create a model for making predictions on new data. The model parameters are automatically adjusted to reflect the latent structure in the data, allowing complex models to be developed even when the lack of theoretical guidance or excessive time required for trial-and-error makes it intractable to specify the parameters manually.

7.1.1 Machine Learning Applications with IACTs

Machine learning has been applied to two main problems in IACT data analysis: particle classification and event reconstruction. In particle classification, the aim is to determine the type of particle that initiated a Cherenkov shower in order to select gamma-ray-initiated showers while rejecting cosmic-ray-initiated ones. As discussed in Section 3.4, a good first approximation to the problem of cosmic-ray rejection is gamma/proton classification. Event reconstruction entails the estimation of physical properties of the primary particle, such as its energy, arrival direction, depth of first interaction, and core location on the ground, typically through the stereo combination of the telescope images.

One machine learning method that has achieved widespread use in gamma-ray astronomy in recent years is that of boosted decision trees (BDTs). A decision tree makes predictions by following a branching sequence of splitting criteria that together non-linearly partition the feature space. When the *boosting* algorithm is applied to decision trees, a sequential

ensemble of trees is trained in which each tree focuses on correcting the mistakes made by the previous trees. BDTs were first applied to the problem of gamma/hadron classification for the H.E.S.S. telescopes by Ohm, van Eldik, and Egberts (2009), and are currently being applied for cosmic ray rejection by the VERITAS Collaboration (Krause, Pueschel, and Maier, 2017). Another tree-based method in use in gamma-ray astronomy is the Random Forest method, in which an ensemble of trees is trained in parallel on a collection of bootstrap samples drawn from the training dataset, and the splitting criteria at each branch of each tree are restricted to make use of a random subset of all available features. The Random Forest method has been used by the MAGIC Collaboration to perform both particle classification and arrival direction estimation (Albert et al., 2008; Aleksić et al., 2010).

The reliance of both decision trees and conventional analysis techniques on image parameters means that these methods throw away a significant amount of information contained in the original images. Addressing this shortcoming requires the development of methods that make use of the telescope images directly. One such method that has been developed for IACT data analysis is the image template method (ITM), which was originally introduced for the CAT telescope by Le Bohec et al. (1998). In ITM, the telescope images are fit using maximum likelihood estimation to a library of template gamma-ray shower images with known physical parameters generated using a semi-analytic model or Monte Carlo simulations. The shower is then reconstructed based on the best-fit physical parameters. ITM has been applied with H.E.S.S. to perform particle classification based on the fit quality as well as shower reconstruction (de Naurois and Rolland, 2009; Parsons and J. A. Hinton, 2014), and with VERITAS for shower reconstruction (Christiansen and VERITAS Collaboration, 2017). Further improvements can be attained by training a BDT model with ITM-derived parameters (Petrashyk, 2019).

In this chapter, we focus on gamma/proton particle classification, although the methods we develop are also applicable to performing event reconstruction.

7.1.2 Training and Evaluating Machine Learning Models

In general, applying a machine learning algorithm follows three main steps. First, relevant input features may be manually derived from the raw data, taking advantage of domain expertise, such as knowledge of the physical properties of gamma rays. This process is known as *feature engineering*. The Hillas parameters discussed in Section 3.4 would be typical such features for IACT data analysis. Using these features, a machine learning algorithm is used to train a model on a dataset, sometimes also referred to as *learning* a model. Finally, *inference* is performed in which the trained model is used to make predictions on new data. In this chapter, we consider only supervised machine learning, in which the desired output values are known for the training data.

In order to evaluate the performance of a machine learning model, it is critical that the evaluation be performed on a test set entirely separate from the data on which the model is trained. Evaluating on a test set helps ensure that a model is not simply *overfitting*, or making predictions using memorized idiosyncrasies of the training data rather than meaningful insights applicable to new data. Ultimately, a model can be useful only to the extent that the training and test sets are representative of real-world data.

A number of metrics can be used to evaluate the performance of a model on a binary classification problem. In general, the four possible outcomes of a binary classifier are described by a *confusion matrix*, shown in Figure 7.1. Any particular example will yield either a true positive, false positive, true negative, or false negative. The performance of a classifier on a given test set can then be reported as a function of the relative proportions of these four outcomes. In our context, a positive example might be a gamma-ray-initiated shower, while a negative example might be one initiated by a cosmic ray.

One metric used in this work is accuracy, or the combined number of true positives and true negatives divided by the total number of examples. However, accuracy can be misleading

if the class proportions are highly imbalanced, since a high accuracy could be achieved by simply predicting that every example belongs to the predominant class. Other important metrics include the true positive rate, or the ratio of true positives to actual positives, and the false positive rate, the ratio of false positives to actual negatives. The false negative rate is 1 minus the true positive rate.

	Actual Positive	Actual Negative
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

Figure 7.1: A prototypical confusion matrix.

A binary classifier may output a score, for instance, ranging between 0 and 1 with higher values representing increased confidence that an example is positive. The classification threshold should be tuned depending on the needs of the problem at hand. One method to visualize the performance of a binary classifier is a receiver operating characteristic (ROC) curve, or a plot of the true positive rate versus false positive rate at different thresholds. From the ROC curve, another metric of overall performance can be obtained, which is the integrated area under the curve (AUC). An AUC of 1 represents a perfect classifier and a random classifier would have an AUC of 0.5. While AUC is more difficult to interpret than accuracy, it is not affected by imbalanced classes.

7.2 Neural Networks and Deep Learning

In this chapter, we study the analysis of IACT data using deep learning, a class of machine learning methods based on deep artificial neural networks. Within the past decade, deep learning methods have risen to prominence due to their unmatched success on problems such as computer vision and natural language processing (LeCun, Bengio, and G. Hinton, 2015). In particular, convolutional neural networks (CNN) and recurrent neural networks (RNN), including long short-term memory (LSTM) networks, have achieved broad success.

Artificial neural networks, which we refer to simply as "neural networks" from now on, are so named because they are loosely inspired by biological neural networks. Neural networks are algorithms consisting of discrete computational units, or neurons, that compute nonlinear functions and are organized into a layered architecture. Deep neural networks, with many layers of neurons, can hierachically extract highly abstract features directly from complex, unstructured data, allowing them to, for example, distinguish a Norfolk Terrier from a Norwich Terrier in the widely used ImageNet dataset composed of photos from the Internet (Russakovsky et al., 2015). While the fundamentals of shallow neural networks were known by the 1980s, deep neural networks appeared for several decades to be infeasible to train. It was not until the breakthrough work of Krizhevsky, Sutskever, and G. Hinton (2012), which combined an efficient CNN architecture, multiple training and regularization techniques, and a fast parallelized implementation in order to halve the error rate on the ImageNet image recognition challenge, that the power of deep learning for computer vision became widely apparent.

Neural networks and deep learning have applications in many areas of high-energy astrophysics and particle physics. In the early days of gamma-ray astronomy, Reynolds (1993) trained a shallow neural network to classify events from the Whipple Observatory using image parameters, and neural networks have been used to classify unassociated sources based on parameters in the second *Fermi*-LAT source catalog (Doert and Errando, 2014). Feng and Lin (2016), taking advantage of the recent rapid advancements in deep learning, applied a CNN to the analysis of VERITAS muon images, which have a characteristic ring shape. This CNN model was subsequently trained on 140,000 VERITAS muon rings identified by citizen scientist volunteers, obtaining better test accuracy than standard muon-tagging algorithms (Bird et al., 2020). CNNs have been used to classify events and localize particle interactions in data from the MicroBooNE neutrino detector (Acciarri et al., 2017) and to reconstruct simulated ultra-high-energy cosmic-ray air showers (Erdmann, Glombitza, and Walz, 2018).

RNNs have been used to identify beauty-quark jets in the Large Hadron Collider (Radovic et al., 2018) and detect low-luminosity gamma ray bursts in simulated CTA light curve data (Sadeh, 2019).

Deep learning methods offer multiple advantages over the existing state-of-the-art IACT image analysis methods. Unlike with BDTs, the images need not be parameterized, doing away with feature engineering and providing access to information that the parameterization may be discarding. While ITM does operate on telescope images, deep learning has the potential to be more efficient, powerful, and flexible. Both ITM and deep neural networks can make better predictions given more templates or training examples. However, template-based methods have computational time and memory requirements that are linear with the number of templates, while a deep learning model has constant prediction speed and memory usage, regardless of the number of examples used to train it. Furthermore, deep learning models can efficiently generalize beyond simply interpolating between known templates and do not require complete, uniform coverage of the parameter space. Deep learning models such as CNNs are able to obtain additional prediction power by efficiently incorporating additional information, such as time channels, without significantly increasing computational complexity (Spencer et al., 2021, see also Section 7.2.4). Finally, ITM cannot be applied directly to irregular hadronic images that lack well-defined templates, but deep learning models can, making them especially well suited for particle classification.

7.2.1 Neural Networks

We now review some of the most important concepts and techniques for building and training deep neural networks. Much of the material and presentation of this section draws from the excellent introduction to neural networks and deep learning of Nielsen (2015).

The fundamental building block of a neural network is the neuron. As illustrated in Figure 7.2, a neuron accepts a vector of inputs a and produces one output, which then

becomes an input to one or more other neurons. Each neuron first calculates the linear combination $z = \sum_j w_j a_j + b$ using a vector of weights w and a bias b associated with that neuron, where z is referred to as the *weighted input*. A nonlinear *activation function* is then applied to the weighted input to get the neuron's output. The activation function applied at each neuron is usually simple, but it is absolutely critical, as without it, the entire network would reduce to a linear classifier no matter how many neurons it contained.

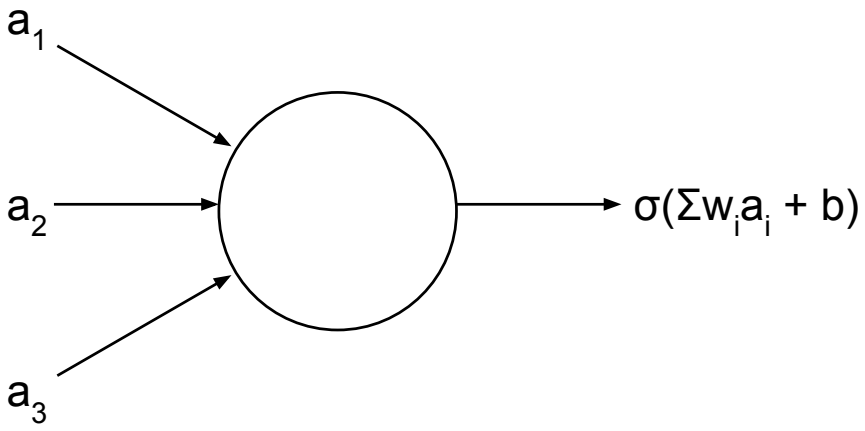


Figure 7.2: Diagram of a neuron.

The earliest neural networks used the perceptron activation function, which outputs 1 if the weighted input is positive and 0 otherwise (Rosenblatt, 1958). However, this function is discontinuous, which makes it impossible to train using the efficient backpropagation method discussed below (Section 7.2.2). To remedy this, the *sigmoid* activation was introduced, which has the form:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{7.1}$$

Because $\sigma(z) \to 1$ when $z \to \infty$, and $\sigma(z) \to 0$ when $z \to -\infty$, the sigmoid function is essentially a smoothed version of the perceptron. Other activation functions are in widespread use as well. The rectified linear unit (ReLU) has become particularly popular. It has the

form:

$$f(z) = \max(0, z). \tag{7.2}$$

ReLU results in empirically better performance with many networks. We will return to some possible reasons for this later, after our discussion of backpropagation.

Next, a set of neurons are connected to each other to form a neural network. The neurons are generally organized into discrete layers, in which each layer's outputs become the next layer's inputs. A diagram of a multi-layer neural network is shown in Figure 7.3. The first layer to the network, called the *input layer*, simply feeds in the input features to the first layer of neurons. Computation is performed by one or more *hidden layers*, and the network's ultimate output is encoded in an *output layer*. In Figure 7.3, the layers are fully connected, meaning that each neuron is connected to all of the neurons in the previous layer.

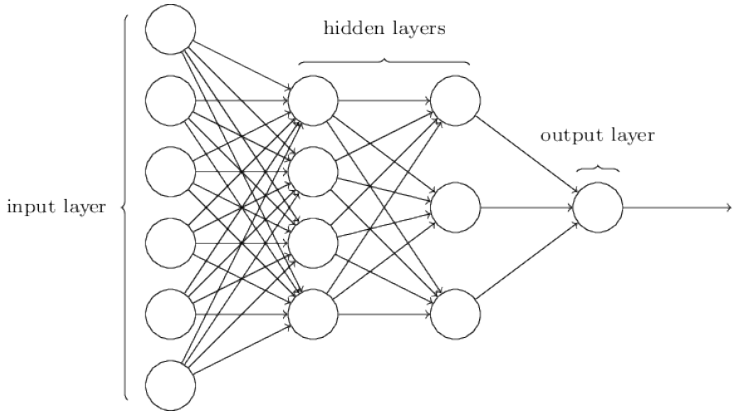


Figure 7.3: Diagram of a fully connected neural network, from Nielsen (2015).

Amazingly, it can be proved that for any continuous function, a neural network can be constructed that can compute that function to arbitrary precision using just a single hidden layer. This is the universal approximation theorem of neural networks (Cybenko, 1989). One way to intuitively understand this theorem is to imagine that the weights of all of the neurons are tuned so that the activation approximates a step function, which can be done for any

nonpolynomial activation function. Superimposing a positive and negative step function by adjusting their biases makes a box. Any continuous function can then be partitioned into small regions and approximated arbitrarily well with boxes given an arbitrary number of boxes (and therefore neurons). For an illuminating discussion and visual proof of the universality theorem, see Nielsen (2015, Chap. 4).

Despite this theoretical result that shallow neural networks can compute any function, in practice it is the composition of multiple layers that allows deep neural networks to efficiently compute highly complex functions and, in particular, to model hierarchical structure in data. Heuristically, a "deep" network is one that contains more than a couple of hidden layers.

7.2.2 Training Neural Networks

So far, we have not discussed how the potentially enormous collection of weight and bias parameters that neural networks use to make predictions are actually set. Fortunately, all of these parameters can be learned from training data through the method of gradient descent, in which the gradient is calculated using the efficient backpropagation algorithm.

Loss Function

To train a neural network, we need to quantify how good its predictions are. We do this by defining a loss function, which compares the output of the network on a training example to the known, correct output for that example, called its *label*. The loss can be considered as a smooth proxy to the real metrics of interest, such as the performance metrics for classification problems discussed in Section 7.1.2. We need a smooth function in order to perform gradient descent. Many different loss functions may be used in practice.

For training our neural networks, we used the cross-entropy loss, which operates on a softmax activation layer, in which the neurons have the form:

$$a_j = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{7.3}$$

where in the denominator we sum over all of the neurons in the layer. The outputs of the softmax activation sum to 1, so it can be thought of as producing a probability distribution. The cross-entropy loss then has the form:

$$C_j = -\sum_j (y_j \ln a_j) = -\ln a_j \tag{7.4}$$

where C is the loss (or cost) and the second equation assumes that the label has the form of a vector y_j in which the entry that corresponds to the correct output is 1 and all others are 0. Although in this work we consider only binary classification, the cross-entropy loss with softmax activation extends naturally to multi-class classification problems, such as distinguishing among Cherenkov showers initiated by gamma rays, protons, and electrons.

Next, we need to find the set of weight and bias parameters that minimizes the loss function, which can be done using the method of gradient descent.

Gradient Descent

For a given (constant) input, a neural network can be thought of as a multivariate function of the weight and bias parameters that outputs the loss. The direction that a small step starting from the current set of parameters will decrease the loss the most is the one opposite to the gradient of this function. On each iteration of the gradient descent loss-minimization algorithm, then, the gradient is calculated and the parameters w are adjusted in the direction opposite to it:

$$w \to w' = w - \eta \nabla C \tag{7.5}$$

where η is the step size, called the *learning rate*, which is a so-called hyperparameter that must programmed when training the network. The gradient for the entire training dataset is the average of the gradients for each example. Hopefully, the algorithm will converge to a global minimum. In practice, neural networks rarely become trapped in suboptimal local minima, perhaps because the parameter space is packed with many saddle points having similar values of the loss function (LeCun, Bengio, and G. Hinton, 2015).

Averaging over the gradients for every example in a large training set is slow. Instead, in practice, on each training step we can calculate the gradient using a small sample, or mini-batch, drawn from the training set. This method is called stochastic gradient descent (SGD). The batch size is a hyperparameter that is typically tuned to make the most efficient use of computing resources. The duration of training can be measured in *epochs*, or full passes through the training set.

A number of advanced optimization algorithms for neural networks have been developed that extend the basic SGD method (for a review, see Ruder, 2016). Many of these algorithms are based around the concept of "momentum" (Qian, 1999), in which the update at each time step depends both on the current gradient and on v, the accumulation of updates made in previous time steps:

$$\begin{aligned} v &\to v' = \mu v - \eta \nabla C \\ w &\to w' = w + v' \end{aligned} \tag{7.6}$$

Intuitively, v can be thought of as a velocity built up when subsequent updates are made in the same direction, speeding convergence and reducing oscillations. Although μ is called the "momentum" parameter, a better physical analogy would be to friction. In this analogy, $1 - \mu$ plays the role of coefficient of friction, such that for the maximum friction $\mu = 0$,

no velocity can build up, and for no friction $\mu = 1$, velocity will be maintained indefinitely unless modified by gradient updates. For $0 < \mu < 1$, the velocity will slowly decay.

An alternative approach to accomplish a similar goal is that of Adadelta (Zeiler, 2012) and RMSProp (G. Hinton, Srivastava, and Swersky, 2012), which accumulate a decaying root-mean-square (RMS) average of each parameter's recently seen gradient components. The update to each parameter is divided by its RMS, giving an adaptive learning rate for each parameter. In these methods, updates to parameters with large oscillations die away, while rarer parameter updates increase in importance. The Adaptive Moment Estimation (Adam) method (Kingma and Ba, 2014) improves performance further by combining a momentum term with an adaptive learning rate.

Backpropagation

In the above discussion, we skipped over a critical question - how do we calculate the gradient? Early, naive approaches required a separate computation for each parameter, which is incredibly slow for a large network. Instead, the backpropagation algorithm (Rumelhart, G. E. Hinton, and Williams, 1986) takes advantage of the chain rule, allowing us to calculate the gradient using only two passes through the network - one forwards and one backwards. The basic idea, illustrated in Figure 7.4, is that a small change to a given parameter propagates through all of the outputs of its corresponding neuron, and through all of their outputs, until the sum of all those changes results in a change in the loss. Working backwards layer by layer, then, we can associate a change in the loss with a change in any particular parameter, using the chain rule.

When performing backpropagation, we start with the standard layer-by-layer feedforward calculation of the weighted inputs a^l and activations z^l for the neurons in each layer l. We define the error δ_j^l of neuron j in layer l as:

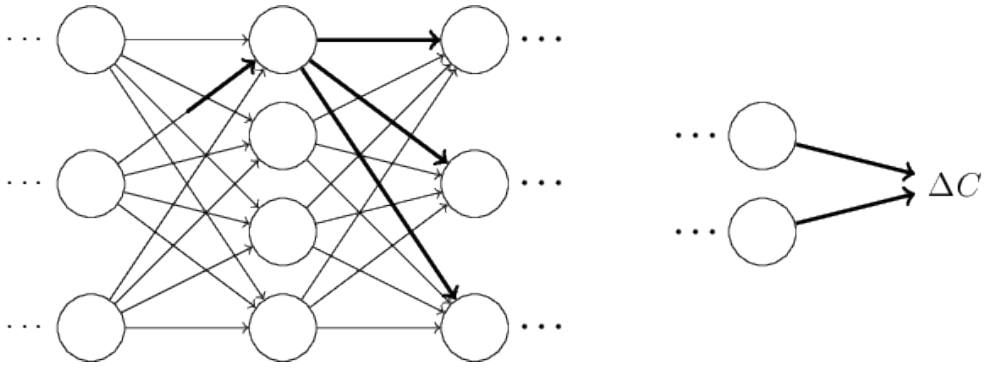


Figure 7.4: An intuitive picture of backpropagation, from Nielsen (2015).

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}. \tag{7.7}$$

We calculate the error on the neurons in the output layer L as:

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \tag{7.8}$$

where $\odot$ is the Hadamard (elementwise) vector product and σ is the activation function. We then *backpropagate* the error layer by layer, using the formula:

$$\delta^l = ((w^{l+1})^\top \delta^{l+1}) \odot \sigma'(z^l). \tag{7.9}$$

The gradient component for each bias is simply the error associated with its corresponding neuron,

$$\frac{\partial C}{\partial b} = \delta, \tag{7.10}$$

and the component for each weight is the product of the activation of its input neuron and error associated with its output neuron,

$$\frac{\partial C}{\partial w} = a_{\text{in}}\delta_{\text{out}}. \tag{7.11}$$

The parameter can then be updated using gradient descent, as discussed above. Often, the term "backpropagation" is used loosely to refer to the entire training procedure including the gradient descent step.

We mentioned above that replacing the sigmoid activation function (Eq.7.1) with ReLU (Eq. 7.2) can improve performance. The equations of backpropagation suggest some reasons why. From Eq. 7.9, we see that the error for an early layer in a deep network will contain many factors of the derivative of the activation function σ'. For the sigmoid activation, $0 < \sigma' \leq 0.25$, so the gradients in a deep network will tend to quickly decay in what is known as the *vanishing gradient problem*. Since the derivative of the ReLU activation[1] is always 0 or 1, it is not susceptible to this problem. Another potential issue affecting the sigmoid activation is *saturation* when the magnitude of the weighted input is large. The activation will be close to 0, so by looking at Eq. 7.11, we see that learning will be very slow. ReLU never saturates on positive inputs, while on the other hand, no learning will occur at all for negative inputs.

Hardware Acceleration

Both the feedforward and backpropagation calculations can be written as matrix operations, allowing them to be trained 10 to 20 times faster when performed on hardware designed to perform parallel computations, such as graphics processing units (GPUs; LeCun, Bengio, and G. Hinton, 2015). In the studies reported in this chapter, we trained our neural networks on two different computing systems featuring the similarly performing Nvidia™ GeForce GTX TITAN X Pascal and Nvidia™ GeForce GTX 1080 Ti GPUs.

[1]Technically, ReLU is not differentiable at 0, but in practice the value of the derivative can be arbitrarily set to either 0 or 1 there.

7.2.3 Overfitting and Regularization

In the discussion above, we implicitly assumed that the optimal parameters were those giving the lowest loss on the training set. What we really want, however, are the parameters that provide the best predictions on new, unseen data. A number of *regularization* techniques are available to help avoid overfitting and allow the network to better generalize. These techniques may be applied separately or in combination.

One basic procedure is *early stopping*. In this procedure, one splits off a portion of the training set to create a reserved validation set on which to periodically evaluate the model. Training is stopped when the error (e.g. accuracy) on the validation set stops decreasing. Deciding when to stop training can be non-trivial because the validation error may plateau before decreasing again. We performed early stopping for all of the networks that we trained.

In addition, a network can be regularized by modifying its structure to encourage it to learn distributed representations that do not rely on the fine-tuning of any one particular neuron. In L1 and L2 regularization, a term is added to the loss function that penalizes the sum of either the absolute values or squares of the weights, respectively. Dropout (Srivastava et al., 2014) is another commonly-used method in which, for each mini-batch while training, a fraction of neurons are randomly and temporarily deleted, helping ensure that the network is learning meaningful features instead of simply memorizing activation patterns. Batch normalization (Ioffe and Szegedy, 2015) is another technique that provides a regularization effect. In this technique, the inputs to each layer are scaled to have zero mean and unit variance within each mini-batch. The primary aim of this technique is to reduce saturation, but it induces regularization as well, since the output for a given training example must be robust to the different inputs occurring when that example is included in different random mini-batches.

The larger the training dataset, the harder it is for the network to overfit. We can emulate

the regularizing effect of enlarging the dataset by transforming existing training examples, a technique known as data augmentation (for a review, see Shorten and Khoshgoftaar, 2019). Transformations such as translations, rotations, flips, and crops can dramatically change the pixel representation of an input image without altering its interpretation. Given our already-large datasets, we did not apply data augmentation in this work, but it has been used with IACT data (e.g. Mangano et al., 2018) and is promising for future exploration.

7.2.4 Convolutional Neural Networks

The fully connected networks we have considered so far treat images as mere collections of pixels with no explicit spatial structure. Convolutional neural networks (CNNs; LeCun et al., 1998) have achieved superb performance on image classification through a network architecture the key properties of which reflect fundamental characteristics of typical image data: local receptive fields, reflecting locality; shared weights, reflecting translational invariance; and pooling, reflecting scale and distortion invariance.

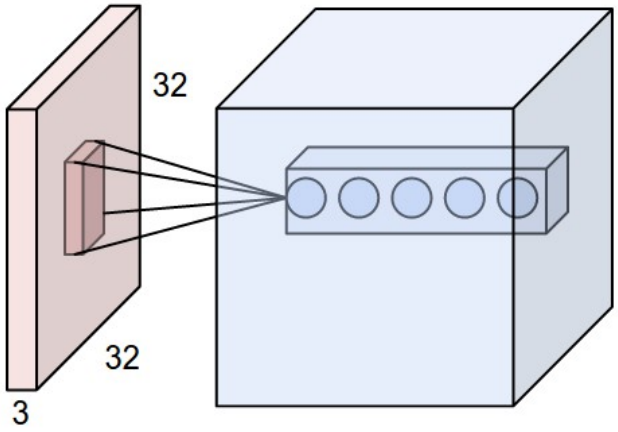


Figure 7.5: Illustration of a convolutional layer operating on an input image, from F.-F. Li et al. (2015).

In a CNN, we picture the neurons in a layer as being laid out in a grid, matching the layout of pixels in an image. In addition to the spatial height and width dimensions, the grid

may have depth, which allows the input to contain multiple channels of information at each point. For example, a color image might contain red, green, and blue color channels. In a convolutional layer, each neuron is only connected to a small local subset of the inputs, for example, a 3x3 region. Most commonly, every channel is included. This region is called a *local receptive field*. We obtain the values for all of the neurons in the output layer by sliding the local receptive field over the input. Multiple features can be calculated for each local receptive field using different parameters, and included as channels in the output. Figure 7.5 illustrates the operation of a convolutional layer on a 32x32x3 color image input, showing a particular local receptive field with multiple input and output channels.

The second key property of a CNN is that the local receptive fields do not have arbitrary parameters, but are constrained to use *shared weights and biases*. Using shared weights and biases significantly reduces the number of trainable parameters and builds in resistance to overfitting, since the parameters must be equally applicable at any location in the image. Intuitively, we slide functions to detect generally useful features, such as vertical edges, over the input. We call these functions *feature maps*. Convolutional networks are so named because this operation is equivalent to a mathematical convolution.

Most commonly, the local receptive field is shifted over one pixel at a time, but a longer so-called *stride length* can also be used. Another hyperparameter is the *kernel size* of the local receptive field. Factorizing large receptive fields such as 5x5 and 7x7 into a stack of 3x3 convolutions reduces the number of parameters and enhances expressiveness through the inclusion of more nonlinearities (Szegedy et al., 2016). For this reason, 3x3 convolutions are generally preferred to larger sizes. It should also be noted that applying a convolution reduces the spatial dimension in the next layer, which can be avoided if desired by padding the input with a boundary of blank pixels.

The third key property of CNNs is the introduction of *pooling* layers. Pooling aggregates the values of nearby neurons, reducing the dimension of the output. For example, *max*

pooling with a 2x2 region size outputs the maximum activation seen in any of four inputs, halving the overall height and width. Other types of pooling are also commonly used.

7.2.5 Recurrent Neural Networks

For some problems, such as speech recognition, the input has the form of a variable-length sequence, such as a sentence. To understand a word in a sentence, one must remember all of the previous words. However, the neural networks described so far have been entirely feedforward, that is, they contain no loops. This property prevents unstable feedback loops from occurring in which a neuron's output depends on its input, but makes it impossible for these networks to maintain an internal state or memory. Recurrent neural networks (RNNs) solve this problem by containing loops that allow a neuron's output from a given time step to affect its input on subsequent time steps. Equivalently, an RNN can be pictured as an unrolled sequence of identical sub-networks, or cells, each passing information to the next.

One of the most widely-used variants of RNN is the long short-term memory network (LSTM), which is designed to be capable of learning long-term dependencies (Hochreiter and Schmidhuber, 1997). The structure of an LSTM is illustrated in Figure 7.6. On each time step, the network accepts an input and produces an output. The input is concatenated with the previous time step's output before processing.

In an LSTM, information flows easily from cell to cell via a hidden state, which the input can only modify by tightly controlled linear interactions through three gates. The forget gate multiplies each element of the hidden state by a number between 0 and 1, allowing selected elements to be discarded. Next, the input gate can add to or subtract from elements of the hidden state. Finally, the output gate selects elements of the hidden state to incorporate into the output vector. Many variants on the LSTM exist, such as the gated recurrent unit, which combines the input and forget gates, among other simplifications (Cho et al., 2014).

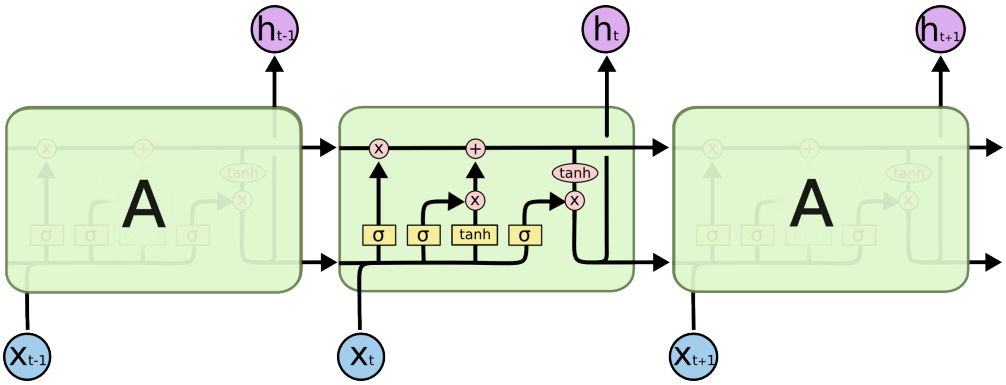


Figure 7.6: Diagram of an LSTM, from Olah (2015). On each time step t, the network combines information from the input x_t (blue circles) and from its internal memory of the previous inputs (black arrows) to produce the output h_t (pink circles).

7.3 Exploring Deep Learning for CTA

Parts of this section have been previously published in the proceedings of the 35th International Cosmic Ray Conference (ICRC2017; Nieto Castaño et al., 2017).

For our exploratory study, we performed gamma/proton particle classification using Monte Carlo simulated Cherenkov air shower images detected by an array of SCTs. The SCT camera has a high-resolution grid of square pixels, making SCT images a natural starting point for understanding the power of deep learning techniques. We used simulations because the true particle type is unknown for real data. We considered only single-image classification, as opposed to event-level stereo classification using all images detected by the telescopes in the array during an event. This study was the first demonstration that deep learning models trained with IACT images can differentiate between Cherenkov showers initiated by gamma rays and by cosmic rays.

7.3.1 Dataset

We generated the dataset for this study using the Monte Carlo simulation chain for CTA described in Acharya et al. (2013), where the atmospheric showers are simulated with

`CORSIKA` (Heck et al., 1998) and the telescope optics and camera readout are simulated with `sim_telarray` (Bernlöhr, 2008). We simulated the response of an array of eight SCTs to approximately 5×10^9 proton showers and 9×10^8 gamma-ray showers, assuming an altitude and atmospheric profile describing the conditions in the Roque de los Muchachos Observatory in La Palma, Spain, where the Northern installation of the CTA Observatory is located. The simulated array had the most telescopes possible given the technical limitations of the image converter software. Two telescope pointing positions were simulated, with the azimuth angle split evenly between 0° and 180° and the zenith angle fixed to 20°.

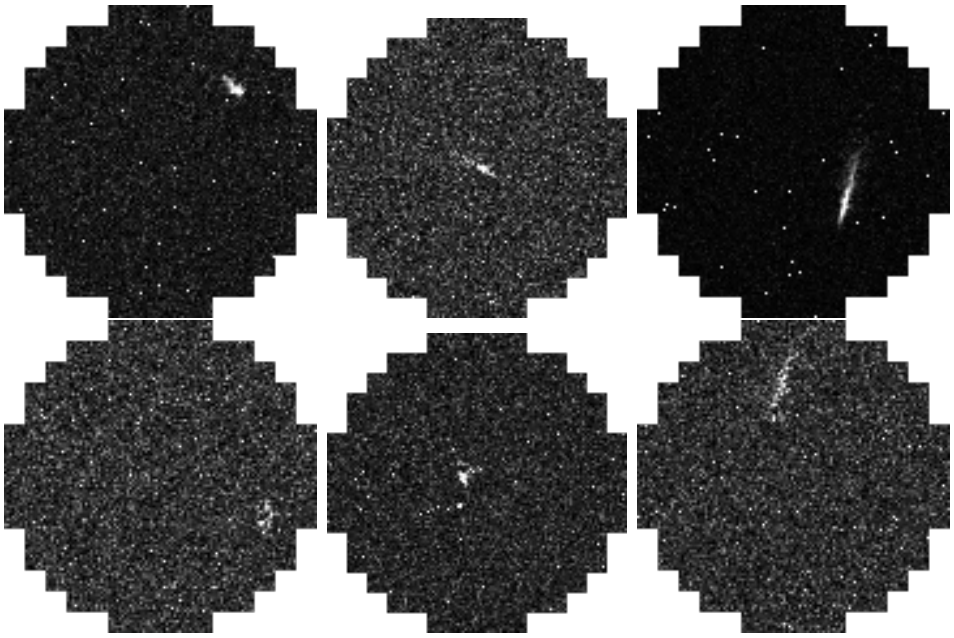


Figure 7.7: Sample simulated images, corresponding to independent events, from Nieto Castaño et al. (2017). *Upper row*: Images from gamma-ray-initiated showers. *Lower row*: Images from proton-initiated showers. The left, center, and right columns contain representative events from the low, medium, and high energy bins, respectively, as defined in the text. The images are normalized for better visualization.

The energy distribution of the initial particles ranged from 3 GeV to 330 TeV for gamma rays and 4 GeV to 600 TeV for protons. The arrival directions for both particle types were homogeneously distributed inside a cone of 10° radius with axis co-linear with the pointing position of the telescopes. The output of `sim_telarray` consists of the collection of digitized

photodetector pulses, for each triggered telescope camera, on an event-by-event basis. We used EventDisplay to convert the output of `sim_telarray` into ROOT format (Brun and Rademakers, 1997), and extracted from that 120x120 16-bit depth portable network graphics (PNG) shower images using custom code based on ROOT and the `OpenCV` library (Bradski, 2000). Relevant Monte Carlo parameters, including the particle type, particle energy, impact parameter, and triggered telescope number, were stored in the image header.

For all events, the arrival direction of the shower was constrained to offsets between 0° and 3°. The event multiplicity, or number of telescopes triggered during an event, was required to be at least three. In order to allow for a consistent reference, the same cuts used in standard BDT-based image classification in EventDisplay were applied to the data (Krause, Pueschel, and Maier, 2017). Specifically, sanity cuts were placed on the mean-scaled length and width ($-2 < MSCW < 2$, $-2 < MSCL < 5$); mean reconstructed energy, mean energy resolution, and spread in reconstructed energy between telescopes ($ERecS > 0$ TeV, $dES \geq 0$ TeV, $ECHi2S \geq 0$); and emission height (0 km < emission height < 50 km). Training was performed separately on three energy bins, corresponding to low (0.1 - 0.31 TeV), medium (0.31 - 1 TeV), and high (1 - 10 TeV) energies.

Within each bin, the data were randomly split into training, validation, and test sets, comprising 80%, 10%, and 10% of each bin's data, respectively. Only images in the training sets were used to train the CNN, while those in the validation sets were used to measure the network's performance after each epoch of training. The images in the test sets were reserved to obtain a final measure of the network's accuracy after training was complete.

7.3.2 Methods

We trained our models using the high-level neural network library Keras (Chollet et al., 2015) with Theano (Al-Rfou et al., 2016) as the computational backend. We explored two well-known models that have achieved state-of-the-art accuracy on the ImageNet image

recognition challenge (Russakovsky et al., 2015), Inception-v3 (Szegedy et al., 2016) and ResNet50 (He et al., 2015). Both are available as applications within Keras. Inception-v3 incorporates a number of techniques to achieve high performance, including factorizing convolutional layers, balancing network width and depth, and employing batch normalization. ResNet50, on the other hand, includes shortcut connections that skip over intermediate layers, allowing the network to model the residuals of the identity mapping between layers, as opposed to the modeling the mapping directly. This residual learning approach permits extremely deep networks to be trained, with fifty layers in the case of ResNet50.

We chose the optimizer by comparing the accuracy of both models using different optimizers with a small subset of data. Using the default hyperparameters in Keras, we found that Adadelta (Zeiler, 2012) provided the best performance on both models. We trained both models on our dataset for ten epochs. The networks were not initialized with any pretrained weights. The input images were resized to 240x240 arrays to meet the minimum input size for these networks. Training on the full dataset required twelve hours per model and energy bin, while classification on the test set was approximately three orders of magnitude faster.

7.3.3 Results

ROC curves (Section 7.1.2) on the test set for both models for the three energy bins are shown in Figure 7.8, along with those of a BDT model trained on stereo events from the same dataset for reference. Because the neural networks are classifying individual images, they are not expected to match the performance of the BDT reference model. The ResNet50 and Inception-v3 models achieved comparable classification performance, with ROC curves of the two models being almost identical for the medium and low bins. All of the models achieved the best classification performance on the higher-energy bins. This is expected because the brighter images corresponding to higher-energy particles should contain more information content. For the Inception-v3 and BDT models, the performance on the high

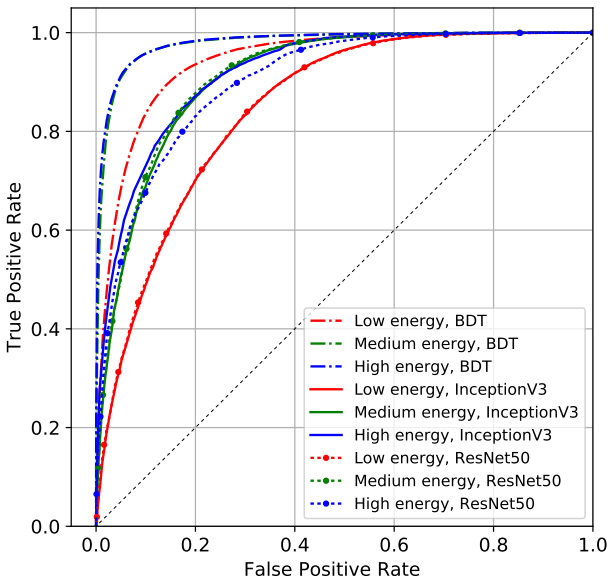


Figure 7.8: ROC curves on the test dataset classified with ResNet50 and Inception-v3 for the low, medium, and high energy bins, from Nieto Castaño et al. (2017). ROC curves from a BDT model trained on stereo events from the same dataset are shown for reference.

energy bin is about the same as on the medium energy bin, which could indicate that the relatively small sample size at the highest energies could be limiting performance. Of the three models, the reference BDT model performs the best on all three energy bins, showing the importance of accessing stereo information.

7.4 Challenges for Using Deep Neural Networks with IACT Data

Based on the lessons we learned from our exploratory work, we identified a number of challenges specific to applying deep neural networks to IACT array data.

7.4.1 Data Processing

Although storing IACT data as PNG images during our exploratory work allowed us to take advantage of pre-existing machine learning libraries for storing and manipulating data, the resulting datasets were memory-inefficient, inflexible, and difficult to search and filter. In

order to efficiently train neural networks on large volumes of IACT data, a specialized data framework should be developed providing compact storage and fast access when training and testing. Data must be stored for events with varying numbers of triggered telescopes, possibly of different telescope types. In order to maximize the usefulness of the data format, it should be flexible enough to store data from CTA as well as existing IACT observatories and to support efficient reading of both single-telescope images and entire events.

7.4.2 Hexagonally Spaced Images

CNNs conventionally operate on input tensors defined by their width and height in a Cartesian basis. However, the pixels of IACT images will be spaced hexagonally if the camera is composed of PMTs. In our exploratory work, we avoided this issue by using SCT data, but in order to apply deep learning methods to all IACTs, methods must be developed either to adapt the convolutional algorithm to accept hexagonally spaced pixels or more simply to convert the hexagonal pixels to rectangular grids without meaningfully degrading image quality in order that existing efficient CNN implementations can be applied.

7.4.3 Stereo Event Reconstruction

In order to use deep learning methods to perform IACT event reconstruction, images from multiple telescopes providing different views of an air shower event must be combined. As noted above, the resulting input data is somewhat heterogeneous. Each event triggers multiple telescopes, and the number of triggered telescopes may vary from event to event. For a large array, only a minority of telescopes may actually be triggered during any given event. Knowledge of the relative positions of the telescopes on the ground must also be integrated into the network in order to perform stereoscopic reconstruction. Performing stereo event reconstruction requires the development of specialized neural network architectures able to deal with all of these issues.

For arrays like CTA that contain several telescope types, the images from each telescope may have different shapes and relative calibrations, posing additional challenges for a neural network design. We do not tackle this aspect of the challenge in this work, and perform stereo reconstruction with homogeneous subarrays only. The software we have created is able to store and load data containing telescopes of multiple types, however.

7.4.4 Transfer Learning from Simulations

Finally, in order for deep learning to be useful in practice, the knowledge gained by a model trained on simulated data must be transferable to real data, as well. Significant systematic uncertainties exist in the modeling of cosmic-ray air showers, which could degrade the performance of models trained on simulations when applied to real data (e.g. Parsons and Schoorlemmer, 2019). The strength of deep neural networks to detect subtle features in IACT images is a weakness if those features are simulation artifacts. To understand the performance of a deep learning model in a real-life setting, it is crucial to test it on actual data. Techniques such as regularization and data augmentation (Section 7.2.3) may help improve the generalization of a deep neural network. As CTA is still under development, the studies in this work were conducted entirely using simulated data, leaving a thorough investigation of performance using actual data for future research.

The sections that follow describe the software we have developed and studies we have performed to address these challenges: to process IACT data into an efficient format for machine learning; train deep neural networks using data from any CTA telescope; transform hexagonally spaced into rectangularly spaced images; and create and study neural network architectures able to perform stereo reconstruction.

7.5 DL1-Data-Handler

In order to process IACT data into an efficient format for machine learning, including coping with hexagonally spaced images, we developed the open-source DL1-Data-Handler package[2] which is able to load and preprocess calibrated IACT images and auxiliary data from any major existing or next-generation IACT (Kim et al., 2021). DL1-Data-Handler contains modules for writing data to a format with efficient storage, fast input/output, and database-like search and selection capabilities; loading data stored in this format; applying arbitrary transformations to the data; and mapping hexagonally spaced pixels to rectangular matrices. DL1-Data-Handler is designed to store data from an array containing multiple telescope types, and allows for reading data from single telescopes of a single telescope type (*mono*), multiple telescopes of a single telescope type from the same event (*stereo*), or multiple telescopes of multiple telescope types from the same event (*multi-stereo*).

DL1-Data-Handler is a publicly available project as part of the CTA Observatory, and the code is available online[3]. DL1-Data-Handler v0.8.3 is described in this work.

7.5.1 DL1 Data Format

The data format defined by DL1-Data-Handler is built on the PyTables library (Alted, Vilata, and PyTables Developers Team, 2002), which provides a flexible, efficient interface to store data on disk organized into a hierarchical structure based on tables and arrays. PyTables is based on the HDF5 high-performance data management library and file format (The HDF Group, 1997). DL1-Data-Handler takes advantage of several HDF5 storage optimizations provided by PyTables, including indexing for fast search queries, dataset chunking for faster input/output, and compression for file size minimization.

In the DL1-Data-Handler data format, the dataset is split into a collection of files on the

[2] `https://github.com/cta-observatory/dl1-data-handler`
[3] `https://github.com/cta-observatory/dl1-data-handler`

order of one gigabyte in size, each containing data from a list of events. Each file contains data from a fixed array or subarray of telescopes. When working with simulated data, events produced by different primary particle types are stored in separate files. In order to reduce data redundancy, each file has a structure with multiple, related tables storing array-level, event-level, and telescope-level information. Full documentation of the DL1-Data-Handler data format is available online[4]. The structure of each file is as follows:

- `/` [Folder]
  - `Array_Information` [Table]
  - `Telescope_Type_Information` [Table]
  - `Events` [Table]
  - `MC_Events` [optional Table]
  - `<telescope type 1>` [Table]
  - `<telescope type 2>` [Table]
  - ...

The `Array_Information` table describes the layout of the telescope array, including the type of each telescope and its position in Cartesian coordinates. Each telescope is identified by a unique numerical telescope ID. The telescope type has the format "<optics type>_<camera type>", where "MST_NectarCam" would be an example. Specific information on each telescope type is given in the `Telescope_Type_Information` table, including the camera type, optics type, number of pixels, and an array storing the position of each pixel in the camera, for plotting images.

The `Events` table contains a record for every event. Each event is uniquely identified by the *obs_id* of the observation that produced it (for simulated data, this is the simulation run) together with an *event_id* within that observation. For simulated data, the `Events` table also contains multiple parameters useful as labels or auxiliary input for machine learning studies,

[4] `https://github.com/cta-observatory/dl1-data-handler/wiki/CTA-ML-Data-Format`

including the primary particle type, the shower zenith and azimuth angles, the coordinates of the shower core impact position, the heights of the first interaction in the shower and of the shower maximum, and the primary particle energy. For each telescope type, the event multiplicity is stored explicitly, to allow for efficient selection on this parameter.

The camera images are stored not in the `Events` table, but in separate tables denoted by the name of the telescope type. Each image includes two one-dimensional arrays of data storing each pixel's integrated charge and the peak channel, which provides the peak arrival time relative to the start of the event. The first entry in each image table (index 0) is a blank dummy image. However, the `Events` and image tables are cross-indexed. For each telescope type, the `Events` table contains a array of *<telescope type>_indices* with an entry for each telescope of that type in order of ascending telescope ID. For each telescope that triggered, its value in the array is its corresponding index in the telescope image table. If a telescope did not trigger, its value is 0, which is the index of the blank dummy image. For lookup in the opposite direction, each row in the image table stores the index of the corresponding event in the `Events` table.

Separating the `Events` and image tables has three main advantages. First, the telescope images can be efficiently stored in an array without needing to store a representation of the images of non-triggered telescopes, greatly reducing the memory requirements. Second, the images from a particular telescope type can be read directly from the telescope image table, which is particularly useful when reading data in *mono* mode. Third, this setup is more memory-efficient with multiple telescope types that may have different numbers of pixels.

When working with simulated (Monte Carlo) data, an additional `MC_Events` table may optionally be created to store the parameters of all simulated events, even ones that did not trigger the array. This increases the processing time and output file size, but can be useful for making a calculation of the sensitivity of the array.

DL1-Data-Handler provides two classes for writing data into this format: `DL1DataDumper`

and `DL1DataWriter`. Much of their underlying functionality, including data calibration, is built around *ctapipe*, an open-source data processing pipeline for CTA (Kosack et al., 2020a). To convert data from another format into the DL1-Data-Handler format, `DL1DataWriter` reads in data using a *ctapipe* `EventSource` and a `DL1DataDumper` subclass specialized for that format. DL1-Data-Handler includes a built-in `CTAMLDataDumper` subclass of `DL1DataDumper` for processing CTA simulations generated using `CORSIKA` and `sim_telarray` (Heck et al., 1998; Bernlöhr, 2008). Data from other observatories may be processed by defining custom `EventSource` and `DL1DataDumper` subclasses.

7.5.2 Reading DL1 Data

DL1-Data-Handler provides the `DL1DataReader` class to read in the resulting data files. `DL1DataReader` loads a list of data files, maps the images to multidimensional arrays, applies preprocessing, and returns the full dataset as an iterator that can be fed into a machine learning model. In addition to the data, an `example_description` object is returned providing the name, telescope type, shape, and data type of each element of the output.

Multiple parameters are available to configure the behavior of `DL1DataReader`. The most important are the mode in which to return the data - *mono*, *stereo*, or *multi-stereo* - and the telescope type or types to use. Specific telescopes may be selected by ID, as well. A PyTables selection statement may be provided in order to efficiently filter events or images, as well as custom filter functions if more complex selection operations are required. The events or images may be shuffled using a specified random seed. In addition to the telescope images, any specified contents of the `Array_Information` and `Events` tables can be returned as auxiliary data. Finally, configuration parameters for `ImageMapper` and `DL1DataProcessor` may be provided and are passed directly to their respective classes, discussed next.

7.5.3 Processing DL1 Data

The image vectors must be converted to two-dimensional images before they can be used. For cameras with square pixels, this operation is trivial, but for those with hexagonally spaced pixels, as discussed in Section 7.4.2, the optimal method to do this is not obvious. DL1-Data-Handler therefore provides the `ImageMapper` class implementing multiple methods to map hexagonally spaced pixels to a square grid. These methods are discussed in detail in Section 7.6. `ImageMapper` has a number of configuration options, including the mapping method to use for each telescope type; the amount of blank padding to apply around the resulting images, if any; the desired output image shape if applicable; and whether to mask out blank pixels when applying interpolation-based mapping methods.

After image mapping is performed, arbitrary transformations may be applied to the images and auxiliary data using the `DL1DataProcessor` class. Transformations are applied using the `Transform` class, a subclass of which is defined for each transformation. `DL1DataProcessor` applies a specified sequence of `Transform` subclasses, each of which processes the data and updates the `example_description`. A number of built-in `Transform` subclasses are defined to perform various transformations, for example, to convert distances from meters to kilometers, rename variables, and to sort telescopes according to a specified order. The user may also define custom transformations.

7.5.4 Data Reduction and Reference Dataset

We used DL1-Data-Handler to create a benchmark dataset which we used for training and testing in the work described in the rest of this chapter. The dataset was generated using simulation files from the third large-scale Monte Carlo production for CTA, the main purpose of which was to issue a final recommendation for the layout of telescopes that will define both the Northern and the Southern Hemisphere arrays of the observatory (Acharyya et al.,

Events/Images	LST	MST-F	MST-N	MST-SC	SST-1M	SST-C	SST-A	All
Training	89/187	259/770	279/891	231/626	206/440	198/440	192/472	392/3827
Test	18/39	54/160	58/185	48/130	43/92	41/92	40/98	82/796

Table 7.1: Size of our benchmark dataset broken down by telescope type. The numbers represent thousands of triggered events and total number of images generated in those events.

2019). The data reduction with DL1-Data-Handler was performed on the European Grid Infrastructure (EGI)[5]. We considered only the Southern array, containing 4 LSTs, 25 MSTs, and 70 SSTs arranged in the baseline recommended layout "S8" (following the notation in Acharyya et al. (2019)). Out of all of the simulated pointing positions, we selected runs with a Zenith angle of 20° and an Azimuth angle of 0° (North pointing). Showers initiated by gamma rays and protons were considered in equal proportions, with a diffuse arrival direction distribution assumed for both particle types.

From the full dataset satisfying the above criteria, we randomly selected about 400,000 events totaling approximately 4 million images, 80% of which we designated for training and 20% for testing. Seven telescope types proposed for CTA were simulated in our dataset: the only model for LST; the two MSTs with a Davies-Cotton optics design, equipped with FlashCam or NectarCam cameras (MST-F and MST-N respectively); the Schwarzschild-Couder MST, or SCT (MST-SC); the single-mirror SST equipped with DigiCam camera (SST-1M); and two dual-mirror SST designs, SST-ASTRI (SST-A) and SST-CHEC (SST-C). For more details on the different telescope designs for CTA, see Acharyya et al. (2019) and references therein. Table 7.1 breaks down the size of the dataset by telescope type.

7.5.5 Extensions to DL1-Data-Handler

CTA collaborators have extended DL1-Data-Handler, adding to the capabilities in the version described in this work. A `DLMAGICEventSource` has been added providing the ability to read real and simulated events from the MAGIC IACT array. In addition, the CTA

[5] `www.egi.eu`

Consortium has developed a DL1 data format based on PyTables which draws from the design choices made and lessons learned during the design of DL1-Data-Handler and is being put forward as a standard for the CTA Observatory. In this format, the images are stored by individual telescope instead of by telescope type, making it more suitable for data taken with real telescopes. DL1-Data-Handler is currently being upgraded to be compatible with this format, which has been implemented in *ctapipe* (Kosack et al., 2020b).

7.6 Mapping Hexagonally Spaced Images with `ImageMapper`

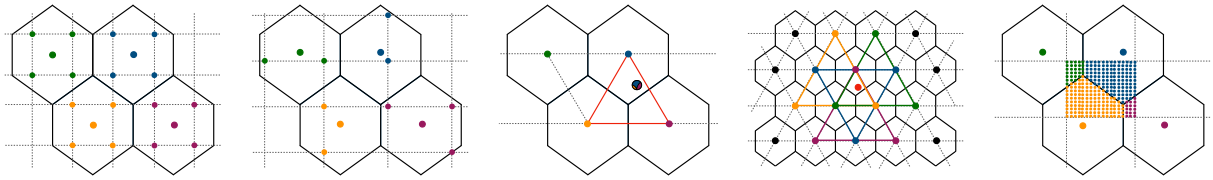
7.6.1 Mapping Methods

As discussed in Section 7.4.2, one challenge when applying CNNs to IACT data is dealing with images formed of hexagonally spaced pixels. There are two main approaches for doing so. One can either modify the CNN algorithm to accept hexagonal images, or more simply transform the images into a rectangular grid. An example of the former approach is the indexed convolution operation implemented in the IndexedConv package, in which each pixel's neighbors are explicitly specified, allowing non-Euclidian pixel grids to be supported (Jacquemont et al., 2019b; Jacquemont and Vuillaume, 2021). DL1-Data-Handler follows the latter approach and contains a number of different image transformation, or mapping, methods, which are implemented in the `ImageMapper` class (Nieto Castaño et al., 2019b).

We generally want a mapping method to minimize changes both to the overall image shape and to the values of individual pixels, trading off between these two objectives depending on the use case. The image shape contains critical information for analyzing the event, and preserving the originally measured pixel values might matter in some cases, such as if the mapping method is used a preprocessing step for further processing. DL1-Data-Handler includes mapping methods optimized for both situations.

DL1-Data-Handler includes five approximately shape-preserving mapping methods: over-

sampling, nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation, and resampling. These methods are illustrated in Figure 7.9. In the oversampling method, originally proposed by Feng and Lin (2016), all of the pixels are divided into 2x2 grids of square pixels, which are then made to be aligned. The value of each oversampled rectangular pixel is set to a quarter of the value of the original hexagonal pixel, so that the image's overall normalization is preserved. In oversampling, the number of square pixels is fixed to be four times the number of hexagonal pixels.

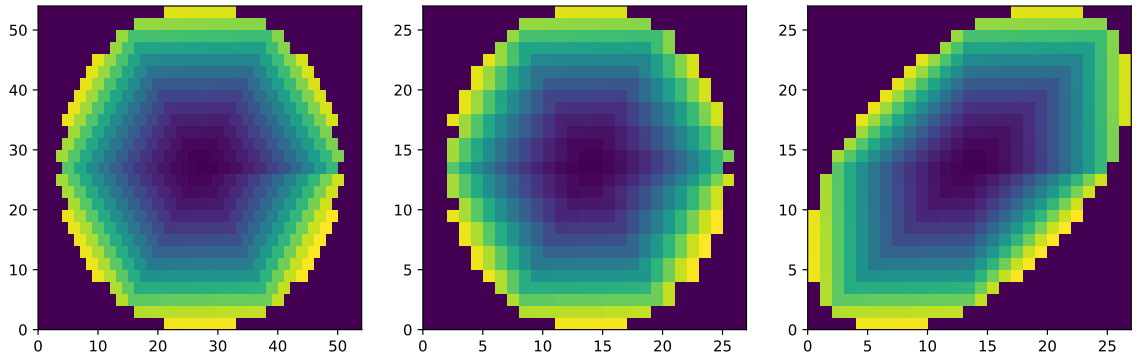


(a) Oversampling (b) Nearest interpolation (c) Bilinear interpolation (d) Bicubic interpolation (e) Rebinning

Figure 7.9: Diagrams depicting the approximately shape-preserving mapping methods in DL1-Data-Handler, from Nieto Castaño et al. (2019b).

In the other four methods, a rectangular grid of arbitrary dimensions is overlaid on the hexagonal pixels. In nearest-neighbor interpolation, the value of each rectangular pixel is set to that of the nearest hexagonal pixel. In bilinear interpolation, the value of each rectangular pixel is interpolated over the three closest hexagonal pixels using Delaunay triangulation, while in bicubic interpolation, it is interpolated over the twelve closest pixels. In the rebinning method, the pixel values are numerically rebinned into a square grid by overlaying a very fine grid over the hexagonal pixels with grid values set to the nearest hexagonal neighbor and summing within the desired rectangular pixels. In all cases, the resulting images are renormalized so that the total charge in all pixels matches that of the original image.

DL1-Data-Handler also includes several mapping methods that do not alter the pixel values, but instead deform the image into a square grid by reindexing the pixels. These methods are fast to perform, but significantly alter the image shape. In the image shifting



(a) Oversampling (b) Image Shifting (c) Axial Addressing

Figure 7.10: Pixel-preserving mapping methods in DL1-Data-Handler, illustrated using the VERITAS camera, with oversampling shown for comparison. The color scale tracks the pixel ID numbers.

method, pixels on alternate rows are shifted to the left and right in order to produce a square grid that keeps adjacent pixels roughly nearby. In the axial addressing mapping method, the pixels are reindexed along two of the hexagonal grid's lines of reflection. The resulting image has a highly warped shape, but allows the indices of neighboring pixels in the original image to be easily recalculated. Figure 7.10 demonstrates these methods. An additional mapping method is provided for use with the IndexedConv package which calculates the indices of neighboring pixels but does not transform the image vector.

`ImageMapper` implements every mapping method as a matrix multiplication to minimize the processing time per image. The initialization time needed to compute the mapping matrix varies considerably among mapping methods, however. `ImageMapper` allows the output grid to be padded with empty pixels, which can be useful to ensure it is square.

7.6.2 Performance of the Mapping Methods

Computational Feasibility

A good mapping method is fast while still allowing the network to achieve good performance. A comparison of the `ImageMapper` initialization time for each of the mapping

methods is given in Table 7.2. The most time-intensive method is bicubic interpolation, followed by rebinning and bilinear interpolation, but all of the methods are computationally feasible. Because the mapping is performed using a matrix multiplication, the time to perform the mapping does not depend on the mapping method.

Mapping Method	Initialization Time [ms]
Oversampling	251 ± 2
Nearest-neighbor Interpolation	295 ± 1
Bilinear Interpolation	525 ± 1
Bicubic Interpolation	3360 ± 8
Rebinning	1320 ± 12
Image Shifting	90 ± 0.5
Axial Addressing	105 ± 0.1

Table 7.2: Comparison of mapping method initialization times in `ImageMapper` for the LST camera.

Effect on Network Performance

The results in this section have been previously published in the proceedings of the 36th International Cosmic Ray Conference (ICRC2019; Nieto Castaño et al., 2019b).

To determine the effect of mapping method on network performance, we trained a single-telescope model with CTLearn (Section 7.7.2) on images mapped using each of the shape-preserving methods. We trained three separate models for one telescope type of each size: LST, MST with the FlashCam camera (MST-F), and the single-mirror SST (SST-1M). The performance of each network is plotted in Figure 7.11. The error bars are the standard deviation of the metrics obtained from ten identical models trained with parameters initialized using different random seeds. The performance of a similar model trained on the same datasets with GammaLearn (Jacquemont et al., 2019a) using indexed convolution is also shown. The accuracy and AUC on both the training and test sets are similar for all of the mapping methods. There is a hint that the nearest-neighbor interpolation method may be worse than the others. The best-performing method is bilinear interpolation, but

the difference between it and the other methods is not significant. Image mapping does not appear to play a significant role in network performance, at least for simulated data. Future work could explore whether this conclusion still holds when applied to real data.

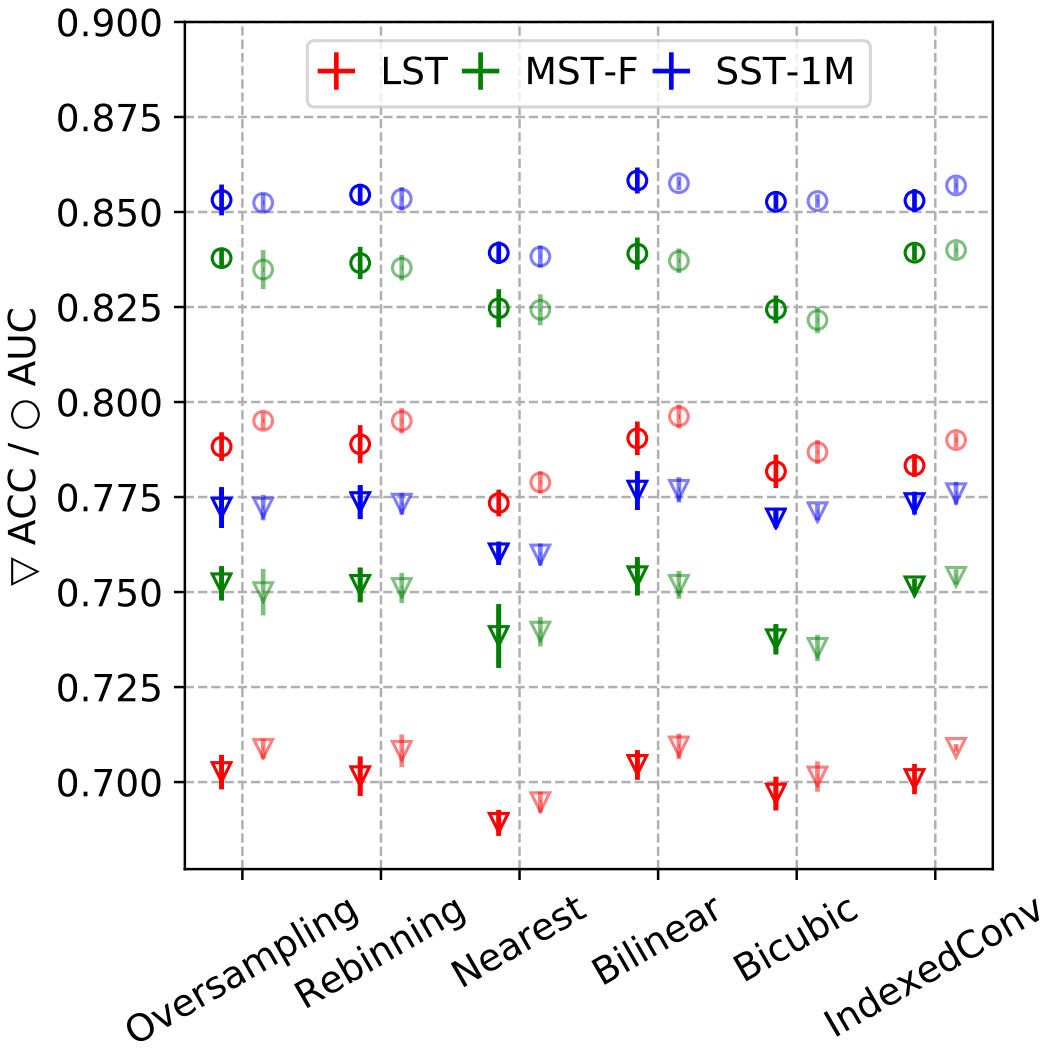


Figure 7.11: Comparison of single-telescope network performance using different mapping methods, from Nieto Castaño et al. (2019b).

7.7 CTLearn

Building on the experience we gained from the exploratory work described in Section 7.3, we have developed CTLearn, an open-source Python package for using deep learning to analyze pixel-wise camera data from arrays of IACTs[6] (Nieto Castaño et al., 2019a; Brill et al., 2019b). CTLearn provides an application-specific framework for configuring and training machine learning models and applying the trained models to generate predictions on a test set. CTLearn remains under active development by the author and others. In this work, CTLearn v0.4.0 is described, with major developments not included in that release summarized in Section 7.7.3.

The motivations for developing CTLearn are threefold. First, the software is open source and publicly available, enabling it to be used for any existing or future IACT array, including both CTA and VERITAS. Second, the software is structured around a configuration file to help encourage reproducible training and prediction. This design is similar to that of the FermiPy package (Wood et al., 2017). Third, CTLearn is based around the widely used TensorFlow deep learning library[7] (Abadi et al., 2016), which provides convenient high-level frameworks for loading data and training neural networks, as well as powerful low-level capabilities for creating custom architectures.

Besides TensorFlow, another deep learning library in common use is PyTorch (Paszke et al., 2019). A comparable package to CTLearn, called GammaLearn, has been built around the PyTorch library (Jacquemont et al., 2019a). Both CTLearn and GammaLearn rely on DL1-Data-Handler to load and preprocess IACT data.

[6] `https://github.com/ctlearn-project/ctlearn`
[7] `https://www.tensorflow.org`

7.7.1 CTLearn Framework

CTLearn allows its user to focus on developing and applying new models using TensorFlow while making use of functionality specifically designed for IACT event classification and reconstruction. All settings, including those for loading and preprocessing data with DL1-Data-Handler, are configured using a YAML configuration file, ensuring that the settings used to train a model are explicitly set and automatically recorded.

CTLearn is organized around the `run_model` module, which parses the configuration file, loads the data, and initializes the model, which may be a default model included with CTLearn or a custom model created by the user using standard TensorFlow functions. Then, depending on the mode set by the user, `run_model` either trains the model, uses the trained model to generate predictions on a test set, or just displays properties of the dataset. The trained model parameters are saved to a TensorFlow model checkpoints file, and run summary data are saved to an events file for display with the TensorBoard web application included with TensorFlow. CTLearn also saves a log file for each run, along with a timestamped copy of the configuration file for documentary purposes. CTLearn includes a number of ancillary scripts for summarizing results and making plots. Figure 7.12 provides an overview of the high-level design of CTLearn.

CTLearn v0.4.0 is based on Python 3.7.3 and TensorFlow 1.13. Other dependencies are DL1-Data-Handler, NumPy (Harris et al., 2020), and PyYAML[8], in addition to Matplotlib (Hunter, 2007), *pandas* (McKinney, 2010; Reback et al., 2020), and scikit-learn (Pedregosa et al., 2011), which are used only in the supplementary scripts.

[8] `https://pyyaml.org/`

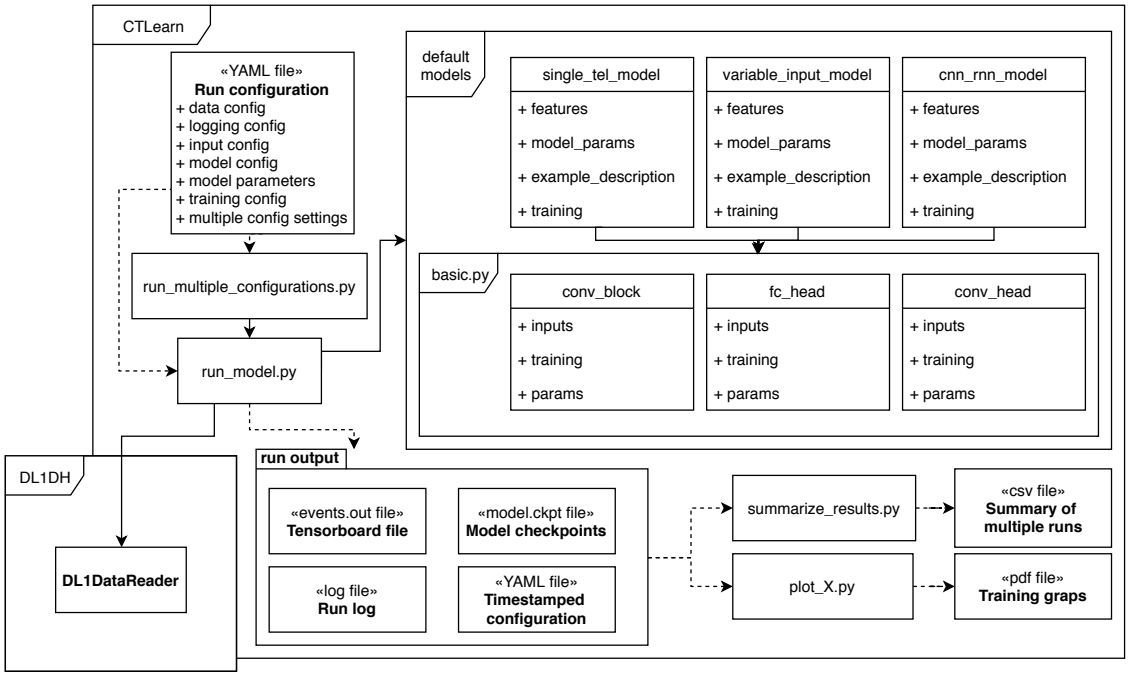


Figure 7.12: Diagram summarizing the design of the CTLearn v0.4.0 framework, from Nieto Castaño et al. (2019a).

7.7.2 Configuration and Settings

As mentioned above, CTLearn stores run settings in a configuration file. All available configuration options are documented online[9], and the main options are described below.

Data

This section contains settings for loading and processing the dataset using `DL1DataReader` and `DL1DataProcessor`. All of the settings and mapping methods described in Sections 7.5.2, 7.5.3, and 7.6 are available. Data can be loaded in *mono*, *stereo*, or *multi-stereo* modes.

[9] `https://github.com/ctlearn-project/ctlearn/blob/v0.4.0/config/example_config.yml`

Input

The user can set parameters for data input using the TensorFlow `Dataset` and `Estimator` APIs, including the number of samples to load into memory before shuffling, the random seed for shuffling, the number of consecutive samples to combine into a batch, and the maximum number of batches to be buffered when prefetching.

Model

CTLearn works with any TensorFlow model obeying the generic signature

```
logits = model(features, params, example_description, training)
```

where `logits` is a vector of raw, non-normalized values typically used as input to a softmax layer, `features` is a dictionary of TensorFlow tensors (multidimensional arrays), `params` is a dictionary of model parameters, `example_description` is a `DL1DataReader` example description, and `training` is a Boolean that's true when training and false when validating or testing. The variables to use as labels and class names to use if applicable are also specified in this section. A separate **Model Parameters** configuration section allows the user to specify parameters for custom models.

CTLearn includes three built-in models for gamma/proton classification. The hyperparameters of these models can be customized using configuration parameters. The single-tel model, which is a simple CNN, is provided to classify single telescope IACT images. By default, this model consists of four convolutional layers with 32, 32, 64, and 128 filters and a kernel size of 3 in each layer, interspaced by an activation layer followed by a max-pooling layer with a kernel size (and stride) of 2. The output of the convolution block is then flattened and fed to a fully connected layer with an output dimensionality of 2, the number of classes. In addition, the CNN-RNN and Variable Input models are provided for performing stereo reconstruction. These models are described in detail in Section 7.8.

All three models are built on a configurable module called `Basic`. `Basic` features three customizable functions. First, `conv_block` defines a CNN block given a number of convolutional layers, number of filters and kernel size for each convolutional layer, and whether to perform max pooling between convolutional layers. For use with the Variable Input model, two network "heads", or output layers, are provided: `conv_head`, which is similar to `conv_block` except that instead of pooling between convolutional layers, there is an option to perform a final average pooling over the CNN output, and `fc_head`, which defines a number of consecutive fully connected layers. All of the `Basic` functions provide the option to perform batch normalization. ReLU is used as the activation function in all hidden layers.

Training

The user can customize training hyperparameters such as the fraction of data randomly extracted from the training dataset for validation purposes, the number of validations to run, how often to evaluate on the validation set, the optimizer, and the base learning rate for the chosen optimizer. Several optimizers are available, including SGD, Adam, Adadelta, and RMSProp. Optionally, each example's loss can be weighted inversely proportionately to the predominance of its class. This option can be useful when dealing with an unbalanced dataset, such as one that is predominately protons.

Prediction

The user can specify settings such as the path to which to write the prediction file and whether to save the labels and example identifiers along with the predictions.

Logging

The user can specify the directory for storing TensorFlow checkpoints and summaries, a timestamped copy of the run configuration, and optionally a timestamped log file.

TensorFlow

The TensorFlow debugger can be optionally invoked.

Multiple Configurations

CTLearn features a tool to train models using multiple configurations in series, sourcing from a single configuration file. This tool can be used to optimize hyperparameters through grid or random searches over discrete sets or linearly or logarithmically spaced ranges.

7.7.3 Extensions to CTLearn

CTLearn is an open-source project, and collaborators in CTA and the broader community have made multiple extensions to it. Some of these contributions are described briefly below.

First, CTLearn Optimizer[10] has been created as a standalone library enabling automated hyperparameter optimization and model training with CTLearn. The supported search algorithms include random search, tree Parzen estimators, Gaussian processes, and genetic algorithms. The optimization can be performed in parallel if suitable hardware is available.

Next, while the studies described in this work focus on the problem of gamma-hadron classification, the capability to perform event reconstruction including both energy and direction estimation has been added to CTLearn. These tasks can be performed by separate models, or all together in a multi-task learning approach (e.g. Ruder, 2017) in which independent task-specific "head" layers are attached to a common backbone model. CTLearn has been used to perform single-image event reconstruction with separate models using a simulated array of LST telescopes (Nieto et al., 2021).

Support for several state-of-the-art architectural designs has also been added to CTLearn. First, a configurable implementation of a residual neural network (ResNet) model has been

[10] `https://github.com/ctlearn-project/ctlearn_optimizer`

added as a default model, similar to the ResNet50 model described in Section 7.3.2. In addition, support has been added for squeeze-and-excitation layers that use global spatial information to emphasize the most important channels by recalibrating channel feature responses (Hu, Shen, and Sun, 2018), or conversely use global channel information to emphasize spatial locations (A. G. Roy, Navab, and Wachinger, 2018), or do both. Squeeze-and-excitation can be thought of as a form of attention mechanism which focuses computational power on the most informative inputs. Attention is an extremely powerful technique that on some tasks can outperform recurrence and convolution entirely by itself (Vaswani et al., 2017).

7.8 Stereo Reconstruction with CTLearn

As discussed in Section 7.4.3, a particular challenge when applying neural networks to data from IACT arrays is performing stereo event reconstruction. One simple way to deal with this problem is to stack the telescope images, either by direct summation or by representing them as separate input channels. Mangano et al. (2018) used this method to predict the particle type, energy, direction, and height of first interaction of particle showers imaged by a simulated array of four LST telescopes.

However, this approach presents several issues. First, it is unclear how to deal with non-triggered telescopes. In the work of Mangano et al. (2018), only events that triggered every telescope were considered. However, this approach limits the applicability of the analysis and is clearly unfeasible for the larger arrays of MSTs and SSTs under development for CTA. An alternative approach is to represent the images of non-triggered telescopes as arrays of all zeroes. However, models with this architecture have been found empirically to provide degraded performance when subsequently applied to real data (Shilon et al., 2019). Another issue with this method is that for large telescope arrays, since any given event will generally trigger only a handful of telescopes, most inputs to the network will always be zero. This

raises potential accuracy and convergence issues when training the network, and also raises the possibility of overfitting caused by the network recognizing specific combinations of triggered telescopes in the training data instead of learning generalizable features.

We have developed two purely deep-learning-based models for performing stereo reconstruction and implemented them as built-in models in CTLearn. In these models, the problem is broken down into two stages. In the first stage, the images from the telescopes are processed into vector representations by a CNN, using shared weights for each image. The second stage combines the information from each telescope. These models can be trained end-to-end, allowing them to make full use of the information in the telescope images and model potentially complex interactions between images in different telescopes.

7.8.1 Variable Input Model

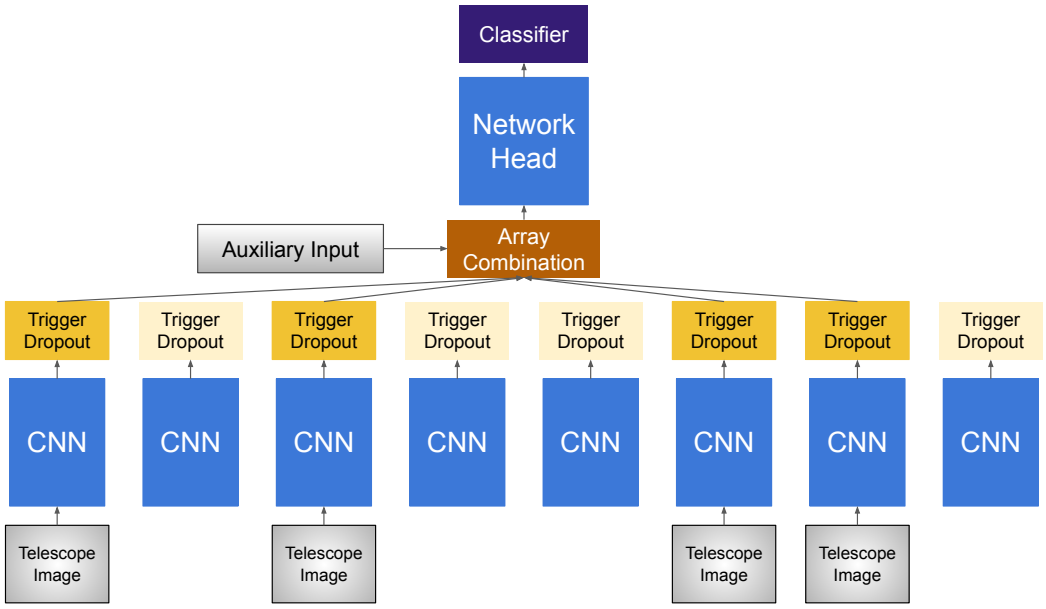


Figure 7.13: Variable Input Network

In the first model, called a Variable Input network, the telescope features are combined along with optional auxiliary event-level features using an array combination layer connected

to a fully connected or convolutional network head. In the array combination layer, the feature maps are flattened and concatenated (for a fully connected head) or stacked channel-wise (for a convolutional head). Any auxiliary features, such as the telescope positions or event-level parameters calculated using another machine learning method, are then appended to the array vector or represented as additional channels, respectively.

Non-triggered telescopes are dealt with by multiplying their telescope feature vectors by zero, essentially eliminating them as inputs to the network. We refer to this method as "trigger dropout" in reference to the Dropout regularization technique (see Section 7.2.3), which involves a similar operation. Unlike Dropout, which is applied only during training, trigger dropout is a required component of the network accounting for the fact that only certain telescopes trigger in an event, and is therefore applied at both training and test time. Since the outputs of non-triggered telescopes are identically zero, their gradients are too, and their weights are not updated in backpropagation. The representation of non-triggered telescope images is therefore arbitrary. A potential downside of this design is that array combination creates a memory bottleneck that could limit the maximum number of telescopes. However, we have not been found this to be a major problem in practice.

7.8.2 CNN-RNN Model

In the second approach, referred to as a CNN-RNN model, an RNN (specifically, an LSTM) is used to combine the telescope vectors to produce an output vector. The LSTM output vector is fed into a set of densely connected layers to produce the final prediction. The CNN-RNN architecture is very similar to the CRNN network which has been shown to improve background rejection performance on real data from the H.E.S.S. IACT array (Shilon et al., 2019; Parsons and Ohm, 2020).

In the CNN-RNN model, no auxiliary information is explicitly fed into the network, but the order of the telescope images could hypothetically encode information about the

relationships between telescopes. This hypothesis is explored in Section 7.10. Figure 7.14 shows a diagram of the CNN-RNN architecture.

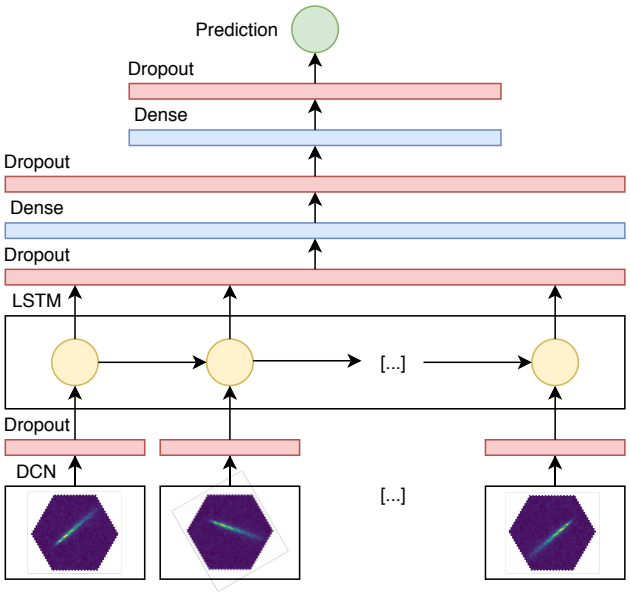


Figure 7.14: Diagram of the CNN-RNN particle classification model implemented in CTLearn, from Nieto Castaño et al., 2019a. The model uses a CNN (labeled as a deep convolutional network or DCN) to derive a vector representation of each image and combines them using an LSTM.

In the CNN-RNN implementation in CTLearn, the CNN is defined as described in Section 7.7.2. The output of each CNN is fed into a Dropout layer that in turn feeds into an LSTM layer with a hidden state size of 2048. After flattening, the output of the LSTM layer passes through three fully connected layers, each preceded by a Dropout layer, with 1024, 512, and 2 neurons respectively. The ReLU activation function is used throughout. The default Dropout fraction is 0.5.

7.9 Benchmarking Model Performance with CTLearn

The contents of this section have been previously published in the proceedings of the 36th International Cosmic Ray Conference (ICRC2019; Nieto Castaño et al., 2019a).

We trained the single-tel and CNN-RNN models on all seven telescope designs proposed

for CTA in our benchmark dataset. We trained the single-tel model on 50,000 batches of 64 images each and the CNN-RNN model on 40,000 batches of 16 events each. Both were validated every 2,500 batches. These settings were chosen to end training approximately when the validation loss stopped decreasing. Hexagonally spaced images were mapped to two-dimensional arrays using bilinear interpolation. Figure 7.15 shows the evolution of the accuracy, AUC, and loss as a function of training samples for both models and all telescope types. The accuracy and AUC values for the validation and test sets are summarized in Table 7.3. We found an excellent match between the metrics obtained from the validation and test sets, with the smallest and largest discrepancies being 0.6% and 1.2% in AUC for the single-tel model, and 0.1% and 1.0% in AUC for the CNN-RNN model. Test AUC values for the single-tel model range from 0.78 and 0.81 for the LST and SST-A telescopes, respectively, up to the 0.84 – 0.87 range where the rest of the telescope designs are located. Test AUC values for the CNN-RNN model are located around 0.90 for most telescope designs.

No quality cuts or data preselection were enforced during training, so the models were fed with all images that triggered the telescopes, as opposed to the conventional analysis, where data preselection and quality cuts are routinely performed. In order to illustrate how data preselection cuts can affect performance we trained the CNN-RNN model imposing a telescope multiplicity cut for both training and validation. As shown in Figure 7.16, the validation AUC is boosted beyond 0.90 in all telescope designs after a multiplicity cut of four triggered telescopes per event is applied, which is a standard multiplicity cut in the analysis of simulated CTA data (Acharyya et al., 2019). The AUC values reach up to 0.98 for all MST designs and the SST-1M design.

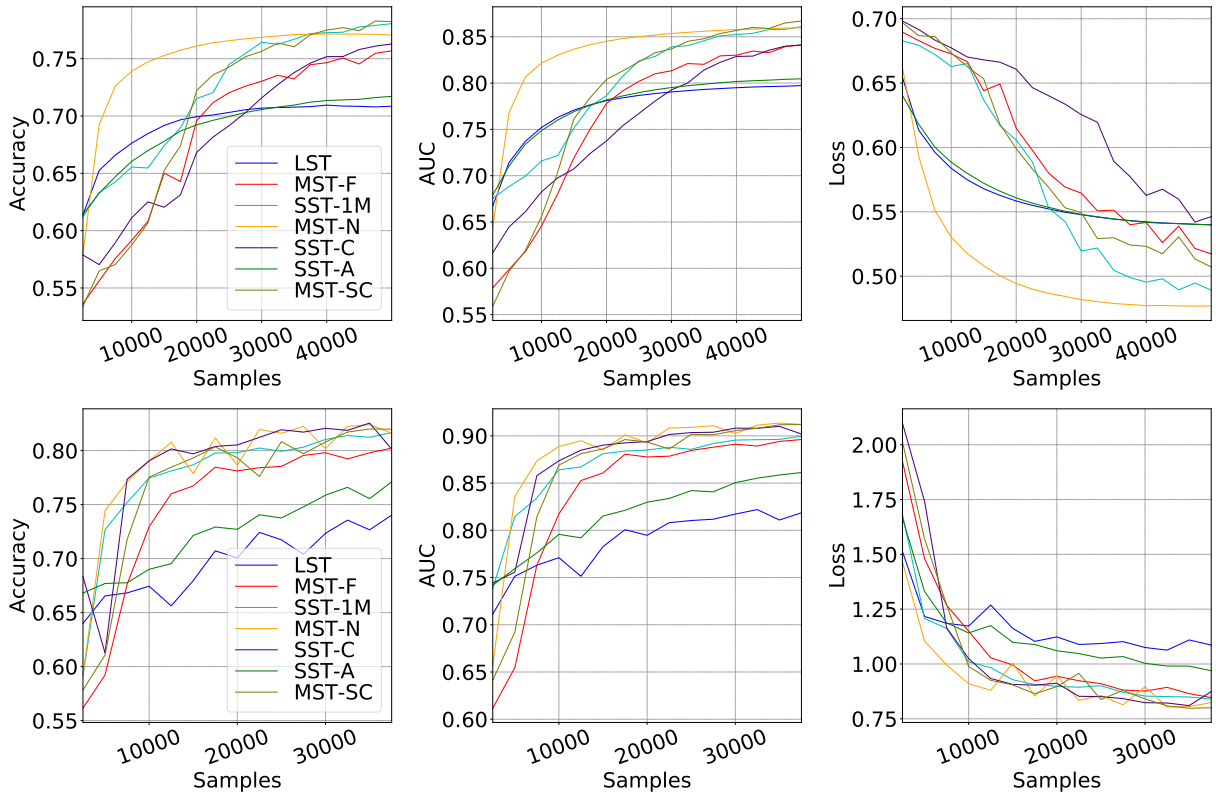


Figure 7.15: Evolution of the main learning metrics for the single-tel (top panels) and CNN-RNN (bottom panels) models as a function of number of samples, from Nieto Castaño et al. (2019a).

Single-tel model		LST	MST-F	MST-N	MST-SC	SST-1M	SST-C	SST-A
Validation	Acc	0.701	0.762	0.784	0.795	0.781	0.753	0.733
	AUC	0.786	0.849	0.869	0.878	0.862	0.828	0.818
Test	Acc	0.697	0.757	0.778	0.785	0.776	0.748	0.725
	AUC	0.778	0.842	0.863	0.866	0.853	0.822	0.808
CNN-RNN model		LST	MST-F	MST-N	MST-SC	SST-1M	SST-C	SST-A
Validation	Acc	0.740	0.802	0.816	0.820	0.817	0.801	0.771
	AUC	0.819	0.896	0.912	0.912	0.900	0.902	0.861
Test	Acc	0.732	0.800	0.816	0.812	0.809	0.796	0.771
	AUC	0.815	0.890	0.909	0.902	0.893	0.898	0.862

Table 7.3: Accuracy and AUC values for the single-tel and the CNN-RNN models for both validation and test datasets as reported in Nieto Castaño et al. (2019a).

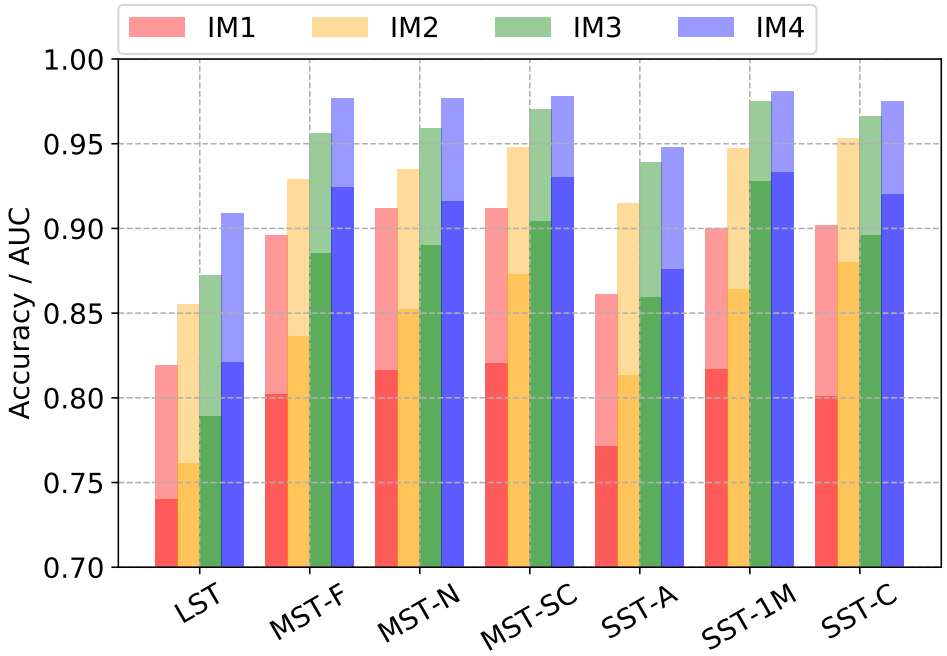


Figure 7.16: Validation accuracy (bright colors) and AUC (pale colors) for the CNN-RNN model, broken down by telescope type and minimum multiplicity cut, from Nieto Castaño et al. (2019a).

7.10 Investigating Telescope Ordering in CNN-RNN Networks

Parts of this section have been published in the proceedings of the 2019 New York Scientific Data Summit (NYSDS), where this work was presented by the author (Brill et al., 2019a).

7.10.1 Telescope Ordering in CNN-RNN Networks

Since RNNs operate on sequential data, the ordering of telescope images in a CNN-RNN network should be meaningful. In particular, it would be natural to encode the relative positions of the telescopes in order to better estimate the shower energy through stereoscopic reconstruction. In previous work using a CNN-RNN network for particle classification, the telescope images were ordered by total image amplitude, or *size*. As size can be considered

to be a proxy for proximity to the shower center, sorting on this parameter may provide an ordering given the absence of temporal information (Shilon et al., 2019).

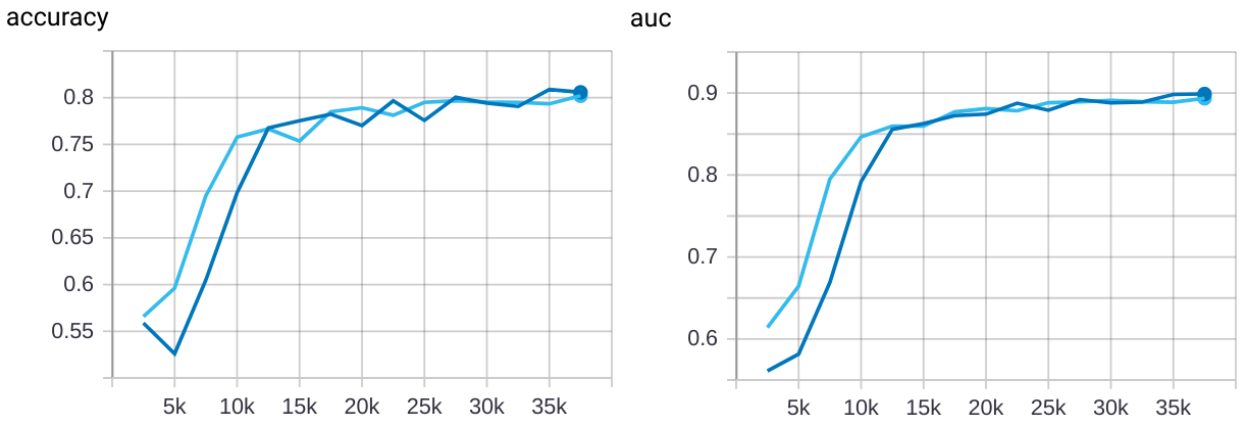
To understand the effect of this ordering on performance, we trained two CNN-RNN networks to classify IACT images as produced by a gamma ray or a cosmic-ray proton, changing only the ordering of the input images. As a control, in one network the images were ordered by telescope ID number, an arbitrary but consistent ordering, while in the other the images were ordered by size. The networks were trained using a sample of 250,000 simulated events from 25 CTA FlashCam telescopes (Gadola et al., 2015), using the dataset described in Section 7.5.4. The hexagonally spaced telescope images were mapped to a grid using the rebinning mapping method. Ten percent of the events in the sample were reserved as a validation set, which was not used for training.

7.10.2 Results

The CNN-RNN models were trained using CTLearn v0.3.0. The Adam optimizer was used with a learning rate of 0.0001. The training was performed in batches of 16 events. The networks were trained on 37,500 events, or approximately 7 epochs.

The results of this experiment are shown in Fig. 7.17. The validation metrics of the two models were approximately the same, with those of the control model being slightly higher. The control model attained validation accuracy and AUC of 80.6% and 0.899, while the model with images sorted by size reached 80.2% and 0.894. We therefore found no evidence that sorting images by size improves classification performance with a CNN-RNN model.

This finding leaves open the possibility that a different ordering of telescope images could result in improved performance. In particular, an ordering which provides sufficient information about the telescopes' position on the ground could help a CNN-RNN to perform stereoscopic reconstruction of Cherenkov air showers. This is particularly important in the context of event reconstruction, especially when estimating the energy of the originating



(a) Accuracy

(b) AUC

Figure 7.17: Validation accuracy and AUC of the CNN-RNN model with images ordered by ID (dark blue) and size (light blue) as a function of number of training steps (batches of 16 events). The model reached an accuracy and AUC of 80.6% and 0.899 when ordered by ID and 80.2% and 0.894 when ordered by size.

particle. Ensuring that telescope position information is effectively provided to neural networks may therefore not only improve their performance on background rejection but also on additional tasks critical for IACT image analysis.

While ordering by size as a proxy for distance to the shower center should provide some relative position information, it may be too incomplete to be useful to the network. Giving the neural network complete access to all relevant stereoscopic parameters may require a more fully physics-informed approach, in which the physics of Cherenkov air showers guides the design of the model architecture, loss function, or model initialization (for a review, see Willard et al., 2020).

7.11 Contributors to this Work

Developing DL1-Data-Handler and CTLearn was a highly collaborative effort, involving the contributions of multiple people. The author made a significant contribution to all of the work described in this chapter, except where otherwise noted, such as the extensions to the

software described in Sections 7.5.5 and 7.7.3. The most important of these contributions include designing the general framework of CTLearn and writing a significant portion of the code; training, optimizing, and debugging many of the neural networks used to generate the results reported in this chapter; helping define the DL1-Data-Handler data format; leading the effort to refactor DL1-Data-Handler into the classes described in this work and writing the bulk of `DL1DataReader` and `DL1DataProcessor`; designing and implementing the stereo reconstruction models discussed in Section 7.8; and initiating and performing the investigation of telescope ordering reported in Section 7.10.

While we do not attempt to disentangle the part played by every collaborator, we highlight here some particular major contributions made by others. Daniel Nieto initiated and directed the studies described in Sections 7.3, 7.6.2, and 7.9; coordinated the overall development of DL1-Data-Handler and CTLearn; and processed the datasets described in Section 7.3.1 and 7.5.4. Bryan Kim made major foundational contributions to the design and implementation of DL1-Data-Handler and CTLearn. Finally, Tjark Miener made substantial contributions to DL1-Data-Handler and CTLearn, and in particular, implemented all of the image mapping methods described in Section 7.9.

Among the extensions to DL1-Data-Handler described in Section 7.5.5, Sahil Yadav implemented `DL1MAGICEventSource` and Tjark Miener is upgrading DL1-Data-Handler to conform to the CTA Consortium format. Among the extensions to CTLearn described in Section 7.7.3, Juan Redondo created the CTLearn Optimizer library and Tjark Miener implemented the support for multi-task learning and state-of-the-art architectures.

We acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

Conclusion

In this thesis, we began by studying three FSRQs using data from VERITAS and *Fermi*-LAT, examining their long-term variability over 10 years, daily and sub-daily variability during bright flares, and multiwavelength spectra and potential for neutrino emission during VHE-detected flares. We identified the challenges posed by variability for performing blazar population studies with IACTs and estimated the prospects for a measurement of the luminosity function of HBL blazars with VERITAS. In order to contribute to the development of new instruments and analysis techniques, we described a control and monitoring software system created to enable the operation of the next-generation pSCT, and investigated how deep learning can be used for cosmic-ray background rejection in order to maximize the scientific impact of the high-resolution data from next-generation IACTs.

We now look to the future. Deep learning, in addition to its applications in data analysis, offers a novel tool for extracting scientific knowledge about blazar variability from rich multiwavelength datasets. A predictive gamma-ray variability model derived using deep learning from well-sampled, long-duration multiwavelength light curves would deepen our physical understanding of the locations, mechanisms, and duty cycle of flares, and enable TeV observatories to better plan both individual observations and surveys. For example, such a model might, given observed data in a band observed at a high, regular cadence (such as GeV gamma rays with *Fermi*-LAT), predict the expected future emission in the TeV band or other wavebands which are typically observed by pointed instruments.

Can flares be predicted, and if so, in which sources, on what timescales, and in what wavebands? A positive finding would provide key input for the development of theoretical models, while at the same time could be of immediate practical use for planning observations.

Even a demonstration that such a model cannot be built, because, for example, on relevant timescales the observed fluctuations are purely stochastic or uncorrelated between different wavebands, would be in itself an important scientific finding. A flare prediction model would also help pointed telescopes schedule observations, multiplying the scientific impact of both the monitoring performed by space telescopes such as *Fermi* and the sensitive follow-up observations carried out by ground-based IACT arrays including VERITAS and CTA.

The deep datasets collected by existing observatories and the improvements promised by next-generation instruments have great potential to grow our understanding of the variable emission from blazars, the most extreme form of AGN, and transformational state-of-the-art machine learning methods can help unlock this potential.

Bibliography

Aab, A., P. Abreu, M. Aglietta, et al. (2020). "Features of the Energy Spectrum of Cosmic Rays above 2.5 $\times 10^{18}$ eV Using the Pierre Auger Observatory". *Physical Review Letters* 125.12, p. 121106.

Aartsen, M. G., M. Ackermann, J. Adams, et al. (2018). "Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922A". *Science* 361.6398.

Aartsen, M., M. Ackermann, J. Adams, et al. (2017). "The IceCube realtime alert system". *Astroparticle Physics* 92, pp. 30–41.

Abadi, M., A. Agarwal, P. Barham, et al. (2016). "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". *arXiv e-prints*, arXiv:1603.04467.

Abbott, B. P., R. Abbott, T. D. Abbott, et al. (2017). "Multi-messenger Observations of a Binary Neutron Star Merger". *The Astrophysical Journal Letters* 848.2, p. L12.

Abdalla, H., H. Abe, F. Acero, et al. (2021). "Sensitivity of the Cherenkov Telescope Array for probing cosmology and fundamental physics with gamma-ray propagation". *Journal of Cosmology and Astroparticle Physics* 2021.02, pp. 048–048.

Abdalla, H., A. Abramowski, F. Aharonian, et al. (2018). "The H.E.S.S. Galactic plane survey". *Astronomy and Astrophysics* 612, A1.

Abdo, A. A., M. Ackermann, M. Ajello, et al. (2010a). "Gamma-ray Light Curves and Variability of Bright Fermi-detected Blazars". *The Astrophysical Journal* 722.1, pp. 520–542.

Abdo, A. A., M. Ackermann, M. Ajello, et al. (2010b). "The Fermi-LAT High-Latitude Survey: Source Count Distributions and the Origin of the Extragalactic Diffuse Background". *The Astrophysical Journal* 720.1, pp. 435–453.

Abdo, A. A., M. Ackermann, M. Ajello, et al. (2011). "Fermi Large Area Telescope Observations of Two Gamma-Ray Emission Components from the Quiescent Sun". *The Astrophysical Journal* 734.2, p. 116.

Abdo, A. A., M. Ackermann, M. Ajello, et al. (2012). "Fermi Observations of γ-Ray Emission from the Moon". *The Astrophysical Journal* 758.2, p. 140.

Abdollahi, S., F. Acero, M. Ackermann, et al. (2020). "Fermi Large Area Telescope Fourth Source Catalog". *The Astrophysical Journal Supplement Series* 247.1, p. 33.

Abeysekara, A. U., A. Albert, R. Alfaro, et al. (2017). "Observation of the Crab Nebula with the HAWC Gamma-Ray Observatory". *The Astrophysical Journal* 843.1, p. 39.

Abeysekara, A. U., A. Archer, W. Benbow, et al. (2018). "VERITAS Observations of the BL Lac Object TXS 0506+056". *The Astrophysical Journal Letters* 861.2, p. L20.

Abeysekara, A. U., A. Archer, W. Benbow, et al. (2019). "Measurement of the Extragalactic Background Light Spectral Energy Distribution with VERITAS". *The Astrophysical Journal* 885.2, p. 150.

Abolmasov, P. and J. Poutanen (2017). "Gamma-ray opacity of the anisotropic stratified broad-line regions in blazars". *Monthly Notices of the Royal Astronomical Society* 464.1, pp. 152–169.

Acciari, V. A., S. Ansoldi, L. A. Antonelli, et al. (2019). "Teraelectronvolt emission from the γ-ray burst GRB 190114C". *Nature* 575.7783, pp. 455–458.

Acciarri, R., C. Adams, R. An, et al. (2017). "Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber". *Journal of Instrumentation* 12.3, P03011.

Acharya, B., M. Actis, T. Aghajani, et al. (2013). "Introducing the CTA concept". *Astroparticle Physics* 43, pp. 3–18.

Acharyya, A., I. Agudo, E. O. Angüner, et al. (2019). "Monte Carlo studies for the optimisation of the Cherenkov Telescope Array layout". *Astroparticle Physics* 111, pp. 35–53.

Ackermann, M., M. Ajello, A. Albert, et al. (2012). "The Fermi Large Area Telescope On Orbit: Event Classification, Instrument Response Functions, and Calibration". *The Astrophysical Journal Supplement Series* 203.1, p. 4.

Ackermann, M., M. Ajello, A. Allafort, et al. (2011). "The Second Catalog of Active Galactic Nuclei Detected by the Fermi Large Area Telescope". *The Astrophysical Journal* 743.2, p. 171.

Ackermann, M., M. Ajello, L. Baldini, et al. (2018). "The Search for Spatial Extension in High-latitude Sources Detected by the Fermi Large Area Telescope". *The Astrophysical Journal Supplement Series* 237.2, p. 32.

Ackermann, M., R. Anantua, K. Asano, et al. (2016). "Minute-timescale >100 MeV γ-Ray Variability during the Giant Outburst of Quasar 3C 279 Observed by Fermi-LAT in 2015 June". *The Astrophysical Journal Letters* 824.2, p. L20.

Actis, M., G. Agnetta, F. Aharonian, et al. (2011). "Design concepts for the Cherenkov Telescope Array CTA: an advanced facility for ground-based high-energy gamma-ray astronomy". *Experimental Astronomy* 32.3, pp. 193–316.

Adams, C., R. Alfaro, G. Ambrosi, et al. (2020a). "Alignment of the optical system of the 9.7-m prototype Schwarzchild-Coulder Telescope". *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Vol. 11445, 114456A.

Adams, C., R. Alfaro, G. Ambrosi, et al. (2020b). "Verification of the optical system of the 9.7-m prototype Schwarzschild-Couder Telescope". *Optical System Alignment, Tolerancing, and Verification XIII*. Vol. 11488. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, p. 1148805.

Adams, C. B., R. Alfaro, G. Ambrosi, et al. (2021). "Detection of the Crab Nebula with the 9.7 m prototype Schwarzschild-Couder telescope". *Astroparticle Physics* 128, p. 102562.

Adams, C. B. (2018). "Software Development and Labratory Tests for the Camera of the Prototype Schwarzchild-Couder Telescope". Bachelor's thesis. University of Wisconsin–Madison.

Aharonian, F. (2000). "TeV gamma rays from BL Lac objects due to synchrotron radiation of extremely high energy protons". *New Astronomy* 5.7, pp. 377–395.

Aharonian, F., A. G. Akhperjanian, A. R. Bazer-Bachi, et al. (2006). "Observations of the Crab nebula with HESS". *Astronomy and Astrophysics* 457.3, pp. 899–915.

Aharonian, F., A. G. Akhperjanian, A. R. Bazer-Bachi, et al. (2007). "An Exceptional Very High Energy Gamma-Ray Flare of PKS 2155-304". *The Astrophysical Journal* 664.2, pp. L71–L74.

Ahlers, M. (2019). "Neutrino Sources from a Multi-Messenger Perspective". *European Physical Journal Web of Conferences*. Vol. 209. European Physical Journal Web of Conferences, p. 01013.

Ajello, M., R. Angioni, M. Axelsson, et al. (2020). "The Fourth Catalog of Active Galactic Nuclei Detected by the Fermi Large Area Telescope". *The Astrophysical Journal* 892.2, p. 105.

Ajello, M., D. Gasparrini, M. Sánchez-Conde, et al. (2015). "The Origin of the Extragalactic Gamma-Ray Background and Implications for Dark Matter Annihilation". *The Astrophysical Journal Letters* 800.2, p. L27.

Ajello, M., R. W. Romani, D. Gasparrini, et al. (2014). "The Cosmic Evolution of Fermi BL Lacertae Objects". *The Astrophysical Journal* 780.1, p. 73.

Ajello, M., M. S. Shaw, R. W. Romani, et al. (2012). "The Luminosity Function of Fermi-detected Flat-spectrum Radio Quasars". *The Astrophysical Journal* 751.2, p. 108.

Albert, J., E. Aliu, H. Anderhub, et al. (2008). "Implementation of the Random Forest method for the Imaging Atmospheric Cherenkov Telescope MAGIC". *Nuclear Instruments and Methods in Physics Research A* 588.3, pp. 424–432.

Aleksić, J., S. Ansoldi, L. A. Antonelli, et al. (2015). "Unprecedented study of the broadband emission of Mrk 421 during flaring activity in March 2010". *Astronomy and Astrophysics* 578, A22.

Aleksić, J., S. Ansoldi, L. A. Antonelli, et al. (2016). "The major upgrade of the MAGIC telescopes, Part I: The hardware improvements and the commissioning of the system". *Astroparticle Physics* 72, pp. 61–75.

Aleksić, J., L. A. Antonelli, P. Antoranz, et al. (2010). "Search for an extended VHE γ-ray emission from Mrk 421 and Mrk 501 with the MAGIC Telescope". *Astronomy and Astrophysics* 524, A77.

Aleksić, J., L. A. Antonelli, P. Antoranz, et al. (2011). "MAGIC Discovery of Very High Energy Emission from the FSRQ PKS 1222+21". *The Astrophysical Journal Letters* 730.1, p. L8.

Alted, F., I. Vilata, and PyTables Developers Team (2002). *PyTables: Hierarchical Datasets in Python*. URL: `http://www.pytables.org/`.

Archambault, S., A. Archer, W. Benbow, et al. (2016). "Upper Limits from Five Years of Blazar Observations with the VERITAS Cherenkov Telescopes". *The Astronomical Journal* 151.6, p. 142.

Archambault, S., A. Archer, W. Benbow, et al. (2017a). "Dark matter constraints from a joint analysis of dwarf Spheroidal galaxy observations with VERITAS". *Physical Review D* 95.8, p. 082001.

Archambault, S., A. Archer, W. Benbow, et al. (2017b). "Search for Magnetically Broadened Cascade Emission from Blazars with VERITAS". *The Astrophysical Journal* 835.2, p. 288.

Archer, A., W. Benbow, R. Bird, et al. (2016). "TeV Gamma-Ray Observations of the Galactic Center Ridge by VERITAS". *The Astrophysical Journal* 821.2, p. 129.

Archer, A., W. Benbow, R. Bird, et al. (2018). "Measurement of cosmic-ray electrons at TeV energies by VERITAS". *Physical Review D* 98.6, p. 062004.

Archer, A., W. Benbow, R. Bird, et al. (2020). "VERITAS Discovery of VHE Emission from the Radio Galaxy 3C 264: A Multiwavelength Study". *The Astrophysical Journal* 896.1, p. 41.

Astropy Collaboration, A. M. Price-Whelan, B. M. SipHocz, et al. (2018). "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". *Astronomical Journal* 156.3, p. 123.

Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, et al. (2013). "Astropy: A community Python package for astronomy". *Astronomy and Astrophysics* 558, A33.

Atwood, W. B., A. A. Abdo, M. Ackermann, et al. (2009). "The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission". *The Astrophysical Journal* 697.2, pp. 1071–1102.

Avni, Y. and J. N. Bahcall (1980). "On the simultaneous analysis of several complete samples. The V/Vmax and Ve/Va variables, with applications to quasars." *The Astrophysical Journal* 235, pp. 694–716.

Bai, X., B. Y. Bi, X. J. Bi, et al. (2019). "The Large High Altitude Air Shower Observatory (LHAASO) Science White Paper". *arXiv e-prints*, arXiv:1905.02773.

Bailer-Jones, C. A. L., J. Rybizki, M. Fouesneau, et al. (2018). "Estimating Distance from Parallaxes. IV. Distances to 1.33 Billion Stars in Gaia Data Release 2". *The Astronomical Journal* 156.2, p. 58.

Ballet, J., T. H. Burnett, S. W. Digel, et al. (2020). "Fermi Large Area Telescope Fourth Source Catalog Data Release 2". *arXiv e-prints*, arXiv:2005.11208.

Barkov, M. V., F. A. Aharonian, S. V. Bogovalov, et al. (2012). "Rapid TeV Variability in Blazars as a Result of Jet-Star Interaction". *The Astrophysical Journal* 749.2, p. 119.

Beckmann, V., D. Engels, N. Bade, et al. (2003). "The HRX-BL Lac sample - Evolution of BL Lac objects". *Astronomy and Astrophysics* 401, pp. 927–938.

Beckmann, V. and C. R. Shrader (2012). *Active Galactic Nuclei*. Wiley-VCH.

Bell, A. R. (1978). "The acceleration of cosmic rays in shock fronts - I." *Monthly Notices of the Royal Astronomical Society* 182, pp. 147–156.

Benbow, W. (2019). "Highlights from the VERITAS AGN Observation Program". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 632.

Berge, D., S. Funk, and J. Hinton (2007). "Background modelling in very-high-energy γ-ray astronomy". *Astronomy and Astrophysics* 466.3, pp. 1219–1229.

Bernlöhr, K., A. Barnacka, Y. Becherini, et al. (2013). "Monte Carlo design studies for the Cherenkov Telescope Array". *Astroparticle Physics* 43, pp. 171–188.

Bernlöhr, K. (2008). "Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_ telarray". *Astroparticle Physics* 30.3, pp. 149–158.

Bhattacharya, D., P. Sreekumar, and R. Mukherjee (2009). "Gamma-ray luminosity function of gamma-ray bright AGNs". *Research in Astronomy and Astrophysics* 9.1, pp. 85–94.

Binney, J. and M. Merrifield (1998). *Galactic Astronomy*. Princeton University Press.

Bird, R., M. K. Daniel, H. Dickinson, et al. (2020). "Muon Hunter: a Zooniverse project". *Journal of Physics Conference Series*. Vol. 1342, p. 012103.

Blanc, O. L., G. Fasola, J. M. Huet, et al. (2018). "Final characterisation and design of the Gamma-ray Cherenkov Telescope (GCT) for the Cherenkov Telescope Array". *Ground-based and Airborne Telescopes VII*. Ed. by H. K. Marshall and J. Spyromilio. Vol. 10700. International Society for Optics and Photonics. SPIE, pp. 299–318.

Blandford, R. D. and A. Königl (1979). "Relativistic jets as compact radio sources." *The Astrophysical Journal* 232, pp. 34–48.

Blandford, R. D. and R. L. Znajek (1977). "Electromagnetic extraction of energy from Kerr black holes." *Monthly Notices of the Royal Astronomical Society* 179, pp. 433–456.

Blumenthal, G. R. and R. J. Gould (1970). "Bremsstrahlung, Synchrotron Radiation, and Compton Scattering of High-Energy Electrons Traversing Dilute Gases". *Reviews of Modern Physics* 42.2, pp. 237–271.

Böttcher, M. and C. D. Dermer (2002). "An Evolutionary Scenario for Blazar Unification". *The Astrophysical Journal* 564.1, pp. 86–91.

Böttcher, M., R. Mukherjee, and A. Reimer (2002). "Predictions of the High-Energy Emission from BL Lacertae Objects: The Case of W Comae". *The Astrophysical Journal* 581.1, pp. 143–154.

Bradski, G. (2000). "The OpenCV Library". *Dr. Dobb's Journal of Software Tools*.

Brill, A. (2019). "Towards a Luminosity Function of TeV Gamma-ray Blazars". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 638.

Brill, A., Q. Feng, T. B. Humensky, et al. (2019a). "Investigating a Deep Learning Method to Analyze Images from Multiple Gamma-ray Telescopes". *2019 New York Scientific Data Summit (NYSDS)*, pp. 1–4.

Brill, A., J. Valverde, R. Mukherjee, et al. (2021). "Variability and Spectral Characteristics of Three Flaring Gamma-ray Quasars Observed by VERITAS and Fermi-LAT". unpublished.

Brill, A., B. Kim, D. Nieto, et al. (2019b). *CTLearn: Deep learning for imaging atmospheric Cherenkov telescopes event reconstruction*. Version v0.4.0. URL: `https://doi.org/10.5281/zenodo.3345947`.

Broderick, A. E., P. Chang, and C. Pfrommer (2012). "The Cosmological Impact of Luminous TeV Blazars. I. Implications of Plasma Instabilities for the Intergalactic Magnetic Field and Extragalactic Gamma-Ray Background". *The Astrophysical Journal* 752.1, p. 22.

Brun, R. and F. Rademakers (1997). "ROOT — An object oriented data analysis framework". *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1. New Computing Techniques in Physics Research V, pp. 81–86.

Buckley, J., L. Bergstrom, B. Binns, et al. (2019). "The Advanced Particle-astrophysics Telescope (APT)". *Bulletin of the American Astronomical Society*. Vol. 51, p. 78.

Burbidge, E. M. (1968). "Spectroscopic Observations of Twenty-Five Quasi-Stellar Objects". *The Astrophysical Journal Letters* 154, p. L109.

Burrows, D. N., J. E. Hill, J. A. Nousek, et al. (2005). "The Swift X-Ray Telescope". *Space Science Reviews* 120, pp. 165–195.

Cavaliere, A. and V. D'Elia (2002). "The Blazar Main Sequence". *The Astrophysical Journal* 571.1, pp. 226–233.

Chang, Y. .-L., B. Arsioli, P. Giommi, et al. (2019). "The 3HSP catalogue of extreme and high-synchrotron peaked blazars". *Astronomy and Astrophysics* 632, A77.

Cheung, C. C., D. Gasparrini, and S. Buson (2017). "Fermi LAT detection of GeV flaring activity from blazar Ton 599 (4C +29.45)". *The Astronomer's Telegram* 10931, p. 1.

Chiang, J. and R. Mukherjee (1998). "The Luminosity Function of the EGRET Gamma-Ray Blazars". *The Astrophysical Journal* 496.2, pp. 752–760.

Cho, K., B. van Merrienboer, C. Gulcehre, et al. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". *arXiv e-prints*, arXiv:1406.1078.

Chollet, F. et al. (2015). *Keras*. URL: `https://keras.io`.

Christiansen, J. and VERITAS Collaboration (2017). "Characterization of a Maximum Likelihood Gamma-Ray Reconstruction Algorithm for VERITAS". *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301, p. 789.

Cogan, P. (2008). "VEGAS, the VERITAS Gamma-ray Analysis Suite". *International Cosmic Ray Conference*. Vol. 3, pp. 1385–1388.

Cogan, P. (2006). "Nanosecond Sampling of Atmospheric Cherenkov Radiation Applied to TeV Gamma-Ray Observations of Blazars with VERITAS". PhD thesis. University College Dublin.

Coppi, P. S. and F. A. Aharonian (1997). "Constraints on the Very High Energy Emissivity of the Universe from the Diffuse GeV Gamma-Ray Background". *The Astrophysical Journal Letters* 487.1, pp. L9–L12.

Costamante, L., S. Cutini, G. Tosti, et al. (2018). "On the origin of gamma-rays in Fermi blazars: beyond the broad-line region". *Monthly Notices of the Royal Astronomical Society* 477.4, pp. 4749–4767.

Costamante, L. and G. Ghisellini (2002). "TeV candidate BL Lac objects". *Astronomy and Astrophysics* 384, pp. 56–71.

CTA Consortium, ed. (2019). *Science with the Cherenkov Telescope Array*. World Scientific Publishing Co. Pte. Ltd. ISBN: 978-981-327-008-4.

CTA Observatory (2019). *CTA's expected baseline performance, version prod3b-v2*. URL: http://www.cta-observatory.org/science/cta-performance/ (visited on 04/22/2021).

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function". *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Davies, J. M. and E. S. Cotton (1957). "Design of the quartermaster solar furnace". *Solar Energy* 1.2-3, pp. 16–22.

de Angelis, A., G. Galanti, and M. Roncadelli (2011). "Relevance of axionlike particles for very-high-energy astrophysics". *Physical Review D* 84.10, p. 105030.

de Naurois, M. and L. Rolland (2009). "A high performance likelihood reconstruction of γ-rays for imaging atmospheric Cherenkov telescopes". *Astroparticle Physics* 32.5, pp. 231–252.

Deil, C., C. Boisson, K. Kosack, et al. (2017). "Open high-level data formats and software for gamma-ray astronomy". *6th International Symposium on High Energy Gamma-Ray Astronomy*. Vol. 1792. American Institute of Physics Conference Series, p. 070006.

Dermer, C. D. and G. Menon (2009). *High Energy Radiation from Black Holes: Gamma Rays, Cosmic Rays, and Neutrinos*. Princeton University Press.

Dermer, C. D., K. Murase, and Y. Inoue (2014). "Photopion production in black-hole jets and flat-spectrum radio quasars as PeV neutrino sources". *Journal of High Energy Astrophysics* 3, pp. 29–40.

Dermer, C. D. and B. Giebels (2016). "Active galactic nuclei at gamma-ray energies". *Comptes Rendus Physique* 17.6, pp. 594–616.

Diaz, G. P. (2020). *CTA Telescopes*. URL: https://www.flickr.com/photos/cta_observatory/50248188677/ (visited on 04/22/2021).

Doert, M. and M. Errando (2014). "Search for gamma-ray-emitting active galactic nuclei in the Fermi-LAT unassociated sample using machine learning". *ApJ* 782.1, pp. 1–7.

Dole, H., G. Lagache, J. .-L. Puget, et al. (2006). "The cosmic infrared background resolved by Spitzer. Contributions of mid-infrared galaxies to the far-infrared background". *Astronomy and Astrophysics* 451.2, pp. 417–429.

Domínguez, A., J. R. Primack, D. J. Rosario, et al. (2011). "Extragalactic background light inferred from AEGIS galaxy-SED-type fractions". *Monthly Notices of the Royal Astronomical Society* 410.4, pp. 2556–2578.

Donato, D., G. Ghisellini, G. Tagliaferri, et al. (2001). "Hard X-ray properties of blazars". *Astronomy and Astrophysics* 375, pp. 739–751.

Dorner, D., A. Arbet-Engels, D. Baack, et al. (2019). "Flux States of Active Galactic Nuclei". *Galaxies* 7.2, p. 57.

Dunlop, J. S. and J. A. Peacock (1990). "The redshift cut-off in the luminosity function of radio galaxies and quasars." *Monthly Notices of the Royal Astronomical Society* 247, p. 19.

Durrer, R. and A. Neronov (2013). "Cosmological magnetic fields: their generation, evolution and observation". *The Astronomy and Astrophysics Review* 21, p. 62.

Dwek, E. and F. Krennrich (2013). "The extragalactic background light and the gamma-ray opacity of the universe". *Astroparticle Physics* 43, pp. 112–133.

Emery, G., M. Cerruti, A. Dmytriiev, et al. (2019). "Observations of the FSRQ 3C 279 during the flaring state of 2017 and 2018 with H.E.S.S." *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 668.

Erdmann, M., J. Glombitza, and D. Walz (2018). "A deep learning-based reconstruction of cosmic ray-induced air showers". *Astroparticle Physics* 97, pp. 46–53.

Errando, M. (2014). "Bright gamma-ray flares of the quasars 3C 279 and PKS 1222+216 observed at the highest energies with Fermi-LAT and VERITAS". *AAS/High Energy Astrophysics Division #14*. Vol. 14. AAS/High Energy Astrophysics Division, p. 106.11.

Errando, M. (2020). Personal communication.

Fanaroff, B. L. and J. M. Riley (1974). "The Morphology of Extragalactic Radio Sources of High and Low Luminosity". *Monthly Notices of the Royal Astronomical Society* 167.1, 31P–36P.

Farina, E. P., R. Decarli, R. Falomo, et al. (2012). "The optical spectrum of PKS 1222+216 and its black hole mass". *Monthly Notices of the Royal Astronomical Society* 424.1, pp. 393–398.

Fegan, D. J. (1997). "TOPICAL REVIEW: γ/hadron separation at TeV energies". *Journal of Physics G Nuclear Physics* 23.9, pp. 1013–1060.

Feng, Q. and T. T. Y. Lin (2016). "The analysis of VERITAS muon images using convolutional neural networks". *Proceedings of the International Astronomical Union* 12.S325, pp. 173–179.

Fermi, E. (1949). "On the Origin of the Cosmic Radiation". *Physical Review* 75.8, pp. 1169–1174.

Fermi-LAT Collaboration, S. Abdollahi, M. Ackermann, et al. (2018). "A gamma-ray determination of the Universe's star formation history". *Science* 362.6418, pp. 1031–1034.

Finke, J. D. and P. A. Becker (2014). "Fourier analysis of Blazar variability". *The Astrophysical Journal* 791.1.

Finke, J. D., S. Razzaque, and C. D. Dermer (2010). "Modeling the Extragalactic Background Light from Stars and Dust". *The Astrophysical Journal* 712.1, pp. 238–249.

Finke, J. D., L. C. Reyes, M. Georganopoulos, et al. (2015). "Constraints on the Intergalactic Magnetic Field with Gamma-Ray Observations of Blazars". *The Astrophysical Journal* 814.1, p. 20.

Foreman-Mackey, D. (2016). "corner.py: Scatterplot matrices in Python". *The Journal of Open Source Software* 1.2, p. 24.

Foreman-Mackey, D., D. W. Hogg, D. Lang, et al. (2013). "emcee: The MCMC Hammer". *Publications of the Astronomical Society of the Pacific* 125.925, p. 306.

Fossati, G., L. Maraschi, A. Celotti, et al. (1998). "A unifying view of the spectral energy distributions of blazars". *Monthly Notices of the Royal Astronomical Society* 299.2, pp. 433–448.

Franceschini, A., G. Rodighiero, and M. Vaccari (2008). "Extragalactic optical-infrared background radiation, its time evolution and the cosmic photon-photon opacity". *Astronomy and Astrophysics* 487.3, pp. 837–852.

Franceschini, A. and G. Rodighiero (2017). "The extragalactic background light revisited and the cosmic photon-photon opacity". *Astronomy and Astrophysics* 603, A34.

Funk, S., D. Jankowsky, H. Katagiri, et al. (2017). "TARGET: A digitizing and trigger ASIC for the Cherenkov telescope array". *AIP Conference Proceedings*. Vol. 1792. 1. AIP Publishing LLC, p. 080012.

Gadola, A., C. Bauer, F. Eisenkolb, et al. (2015). "FlashCam: A novel Cherenkov telescope camera with continuous signal digitization". *Journal of Instrumentation* 10.1.

Galbraith, W. and J. V. Jelley (1953). "Light Pulses from the Night Sky associated with Cosmic Rays". *Nature* 171.4347, pp. 349–350.

Gammell, S. (2004). "A Search for Very High Energy Gamma-ray Emission from Active Galactic Nuclei using Multivariate Analysis Techniques". PhD thesis. University College Dublin.

Gao, S., A. Fedynitch, W. Winter, et al. (2019). "Modelling the coincident observation of a high-energy neutrino and a bright blazar flare". *Nature Astronomy* 3, pp. 88–92.

Gao, S., M. Pohl, and W. Winter (2017). "On the Direct Correlation between Gamma-Rays and PeV Neutrinos from Blazars". *The Astrophysical Journal* 843.2, p. 109.

Garofalo, D., D. A. Evans, and R. M. Sambruna (2010). "The evolution of radio-loud active galactic nuclei as a function of black hole spin". *Monthly Notices of the Royal Astronomical Society* 406.2, pp. 975–986.

Gehrels, N., G. Chincarini, P. Giommi, et al. (2004). "The Swift Gamma-Ray Burst Mission". *The Astrophysical Journal* 611, pp. 1005–1020.

Ghisellini, G., A. Celotti, G. Fossati, et al. (1998). "A theoretical unifying scheme for gamma-ray bright blazars". *Monthly Notices of the Royal Astronomical Society* 301.2, pp. 451–468.

Ghisellini, G., C. Righi, L. Costamante, et al. (2017). "The Fermi blazar sequence". *Monthly Notices of the Royal Astronomical Society* 469.1, pp. 255–266.

Ghisellini, G. and F. Tavecchio (2009). "Canonical high-power blazars". *Monthly Notices of the Royal Astronomical Society* 397.2, pp. 985–1002.

Ghisellini, G. and P. Madau (1996). "On the origin of the gamma-ray emission in blazars". *Monthly Notices of the Royal Astronomical Society* 280.1, pp. 67–76.

Giannios, D. and D. A. Uzdensky (2019). "GRB and blazar jets shining through their stripes". *Monthly Notices of the Royal Astronomical Society* 484.1, pp. 1378–1389.

Giebels, B. and B. Degrange (2009). "Lognormal variability in BL Lacertae". *Astronomy and Astrophysics* 503.3, pp. 797–799.

Gilmore, R. C., R. S. Somerville, J. R. Primack, et al. (2012). "Semi-analytic modelling of the extragalactic background light and consequences for extragalactic gamma-ray spectra". *Monthly Notices of the Royal Astronomical Society* 422.4, pp. 3189–3207.

Giomi, M., L. Gerard, and G. Maier (2016). "Optimal strategies for observation of active galactic nuclei variability with Imaging Atmospheric Cherenkov Telescopes". *Astroparticle Physics* 80, pp. 8–15.

Guilbert, P. W., A. C. Fabian, and M. J. Rees (1983). "Spectral and variability constraints on compact sources". *Monthly Notices of the Royal Astronomical Society* 205, pp. 593–603.

Guo, F., Y.-H. Liu, W. Daughton, et al. (2015). "Particle Acceleration and Plasma Dynamics during Magnetic Reconnection in the Magnetically Dominated Regime". *The Astrophysical Journal* 806.2, p. 167.

H.E.S.S. Collaboration, Abdalla, H., Abramowski, A., et al. (2017). "Characterizing the long-term variability of PKS5-304 with H.E.S.S. and Fermi-LAT". *Astronomy and Astrophysics* 598, A39.

Hanna, D., A. McCann, M. McCutcheon, et al. (2010). "An LED-based flasher system for VERITAS". *Nuclear Instruments and Methods in Physics Research A* 612.2, pp. 278–287.

Harris, C. R., K. J. Millman, S. J. van der Walt, et al. (2020). "Array programming with NumPy". *Nature* 585, pp. 357–362.

Hartman, R. C., D. L. Bertsch, S. D. Bloom, et al. (1999). "The Third EGRET Catalog of High-Energy Gamma-Ray Sources". *The Astrophysical Journal Supplement Series* 123.1, pp. 79–202.

Harvey, A. L. W., M. Georganopoulos, and E. T. Meyer (2020). "Powerful extragalactic jets dissipate their kinetic energy far from the central black hole". *Nature Communications* 11.1, p. 5475.

Hassan, T., L. Arrabito, K. Bernlöhr, et al. (2017). "Monte Carlo performance studies for the site selection of the Cherenkov Telescope Array". *Astroparticle Physics* 93, pp. 76–85.

Hayashida, M., K. Nalewajko, G. M. Madejski, et al. (2015). "Rapid Variability of Blazar 3C 279 during Flaring States in 2013-2014 with Joint Fermi-LAT, NuSTAR, Swift, and Ground-Based Multiwavelength Observations". *The Astrophysical Journal* 807.1, p. 79.

He, K., X. Zhang, S. Ren, et al. (2015). "Deep Residual Learning for Image Recognition". *arXiv e-prints*, arXiv:1512.03385.

Heck, D., J. Knapp, J. Capdevielle, et al. (1998). "CORSIKA: A Monte Carlo code to simulate extensive air showers". *Report fzka* 6019.11.

Heitler, W. (1954). *Quantum theory of radiation*. Oxford University Press.

Hervet, O., C. Boisson, and H. Sol (2015). "Linking radio and gamma-ray emission in Ap Librae". *Astronomy and Astrophysics* 578, A69.

— (2016). "An innovative blazar classification based on radio jet kinematics". *Astronomy and Astrophysics* 592, A22.

Hervet, O. (2020). Personal communication.

Hillas, A. M. (1984). "The Origin of Ultra-High-Energy Cosmic Rays". *Annual Review of Astronomy and Astrophysics* 22, pp. 425–444.

— (1985). "Cerenkov Light Images of EAS Produced by Primary Gamma Rays and by Nuclei". *19th International Cosmic Ray Conference (ICRC19), Volume 3*. Vol. 3, p. 445.

— (2013). "Evolution of ground-based gamma-ray astronomy from the early days to the Cherenkov Telescope Arrays". *Astroparticle Physics* 43, pp. 19–43.

Hillas, A. M., C. W. Akerlof, S. D. Biller, et al. (1998). "The Spectrum of Teravolt Gamma Rays from the Crab Nebula". *The Astrophysical Journal* 503.2, pp. 744–759.

Hinton, G., N. Srivastava, and K. Swersky (2012). "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent".

Hochreiter, S. and J. Schmidhuber (1997). "Long Short-Term Memory". *Neural Computation* 9.8, pp. 1735–1780.

Hofmann, W. (2006). "Performance Limits for Cherenkov Instruments". *arXiv e-prints*, astro–ph/0603076.

Hofmann, W., I. Jung, A. Konopelko, et al. (1999). "Comparison of techniques to reconstruct VHE gamma-ray showers from multiple stereoscopic Cherenkov images". *Astroparticle Physics* 12.3, pp. 135–143.

Hogg, D. W. (1999). "Distance measures in cosmology". *arXiv e-prints*, astro–ph/9905116.

Holder, J. (2014). "Detection of Persistent VHE emission from PKS 1222+216 (4C +21.35) with VERITAS". *The Astronomer's Telegram* 5981.

Holder, J., R. W. Atkins, H. M. Badran, et al. (2006). "The first VERITAS telescope". *Astroparticle Physics* 25.6, pp. 391–401.

Holder, J. (2011). "VERITAS: Status and Highlights". *International Cosmic Ray Conference*. Vol. 11. International Cosmic Ray Conference, p. 137.

— (2015). "Atmospheric Cherenkov Gamma-ray Telescopes". *arXiv e-prints*, arXiv:1510.05675.

Hooper, D., T. Linden, and A. Vieregg (2019). "Active galactic nuclei and the origin of IceCube's diffuse neutrino flux". *Journal of Cosmology and Astroparticle Physics* 2019.2, p. 012.

Hopkins, P. F., G. T. Richards, and L. Hernquist (2007). "An Observational Determination of the Bolometric Quasar Luminosity Function". *The Astrophysical Journal* 654.2, pp. 731–753.

Hosking, D. N. and L. Sironi (2020). "A First-principle Model for Polarization Swings during Reconnection-powered Flares". *The Astrophysical Journal Letters* 900.2, p. L23.

Hu, J., L. Shen, and G. Sun (2018). "Squeeze-and-Excitation Networks". *IEEE Conference on Computer Vision and Pattern Recognition*.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". *Computing in Science & Engineering* 9.3, pp. 90–95.

Ioffe, S. and C. Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *arXiv e-prints*, arXiv:1502.03167.

Jacquemont, M., T. Vuillaume, A. Benoit, et al. (2019a). "GammaLearn: A Deep Learning Framework for IACT Data". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 705.

Jacquemont, M., L. Antiga, T. Vuillaume, et al. (2019b). "Indexed Operations for Non-rectangular Lattices Applied to Convolutional Neural Networks". *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,* INSTICC. SciTePress, pp. 362–371. ISBN: 978-989-758-354-4.

Jacquemont, M. and T. Vuillaume (2021). *IndexedConv/IndexedConv: v1.3*. Version v1.3. URL: https://doi.org/10.5281/zenodo.4419866.

Jorstad, S. G., A. P. Marscher, J. R. Mattox, et al. (2001). "Multiepoch Very Long Baseline Array Observations of EGRET-detected Quasars and BL Lacertae Objects: Superluminal Motion of Gamma-Ray Bright Blazars". *The Astrophysical Journal Supplement Series* 134.2, pp. 181–240.

Kadler, M., F. Krauß, K. Mannheim, et al. (2016). "Coincidence of a high-fluence blazar outburst with a PeV-energy neutrino event". *Nature Physics* 12.8, pp. 807–814.

Kalberla, P. M. W., W. B. Burton, D. Hartmann, et al. (2005). "The Leiden/Argentine/Bonn (LAB) Survey of Galactic HI. Final data release of the combined LDS and IAR surveys with improved stray-radiation corrections". *Astronomy and Astrophysics* 440, pp. 775–782.

Katarzyński, K., H. Sol, and A. Kus (2001). "The multifrequency emission of Mrk 501. From radio to TeV gamma-rays". *Astronomy and Astrophysics* 367, pp. 809–825.

Keenan, M., E. T. Meyer, M. Georganopoulos, et al. (2021). "The relativistic jet dichotomy and the end of the blazar sequence". *Monthly Notices of the Royal Astronomical Society, Advance Access*.

Keivani, A., K. Murase, M. Petropoulou, et al. (2018). "A Multimessenger Picture of the Flaring Blazar TXS 0506+056: Implications for High-energy Neutrino Emission and Cosmic-Ray Acceleration". *The Astrophysical Journal* 864.1, p. 84.

Kembhavi, A. K. and J. V. Narlikar (1999). *Quasars and active galactic nuclei : an introduction*. Cambridge University Press.

Kim, B., A. Brill, T. Miener, et al. (2021). *DL1-Data-Handler: DL1 HDF5 writer, reader, and processor for IACT data*. Version v0.8.3. URL: https://doi.org/10.5281/zenodo.4575505.

Kingma, D. P. and J. Ba (2014). "Adam: A Method for Stochastic Optimization". *arXiv e-prints*, arXiv:1412.6980.

Kosack, K., J. Watson, J. Jacquemier, et al. (2020a). *cta-observatory/ctapipe: v0.8.0*. Version v0.8.0. URL: https://doi.org/10.5281/zenodo.3837306.

Kosack, K., J. Watson, M. Nöthe, et al. (2020b). *cta-observatory/ctapipe: v0.10.0*. Version v0.10.0. URL: https://doi.org/10.5281/zenodo.4279334.

Krause, M., E. Pueschel, and G. Maier (2017). "Improved γ /hadron separation for the detection of faint γ -ray sources using boosted decision trees". *Astroparticle Physics* 89, pp. 1–9.

Krizhevsky, A., I. Sutskever, and G. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, et al. Vol. 25. Curran Associates, Inc.

Kushwaha, P., S. Sahayanathan, and K. P. Singh (2013). "High energy emission processes in OJ 287 during 2009 flare". *Monthly Notices of the Royal Astronomical Society* 433.3, pp. 2380–2388.

Le Bohec, S., B. Degrange, M. Punch, et al. (1998). "A new analysis method for very high definition imaging atmospheric Cherenkov telescopes as applied to the CAT telescope." *Nuclear Instruments and Methods in Physics Research A* 416.2, pp. 425–437.

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning". *Nature* 521.7553, pp. 436–444.

LeCun, Y., L. Bottou, Y. Bengio, et al. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lessard, R. W., J. H. Buckley, V. Connaughton, et al. (2001). "A new analysis method for reconstructing the arrival direction of TeV gamma rays using a single imaging atmospheric Cherenkov telescope". *Astroparticle Physics* 15.1, pp. 1–18.

Li, F.-F., A. Karpathy, J. Johnson, et al. (2015). *CS231n: Convolutional Neural Networks for Visual Recognition*. URL: https://cs231n.github.io/convolutional-networks/ (visited on 03/17/2021).

Li, T.-P. and Y.-Q. Ma (1983). "Analysis methods for results in gamma-ray astronomy." *The Astrophysical Journal* 272, pp. 317–324.

Liu, Y., D. R. Jiang, and M. F. Gu (2006). "The Jet Power, Radio Loudness, and Black Hole Mass in Radio-loud Active Galactic Nuclei". *The Astrophysical Journal* 637.2, pp. 669–681.

Lombardi, S., O. Catalano, S. Scuderi, et al. (2020). "First detection of the Crab Nebula at TeV energies with a Cherenkov telescope in a dual-mirror Schwarzschild-Couder configuration: the ASTRI-Horn telescope". *Astronomy and Astrophysics* 634, A22.

Longair, M. S. (1994). *High energy astrophysics*. Vol. 2. Cambridge University Press.

Lott, B., D. Gasparrini, and S. Ciprini (2020). "The Fourth Catalog of Active Galactic Nuclei Detected by the Fermi Large Area Telescope – Data Release 2". *arXiv e-prints*, arXiv:2010.08406.

Lynds, C. R., A. N. Stockton, and W. C. Livingston (1965). "New Spectroscope Observations of Quasi-Stellar Sources." *The Astrophysical Journal* 142, p. 1667.

Maier, G. and J. Holder (2017). "Eventdisplay: An Analysis and Reconstruction Package for Ground-based Gamma-ray Astronomy". *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301, p. 747.

Mangano, S., C. Delgado, M. I. Bernardos, et al. (2018). "Extracting Gamma-Ray Information from Images with Convolutional Neural Network Methods on Simulated Cherenkov Telescope Array Data". *Artificial Neural Networks in Pattern Recognition*. Ed. by L. Pancioni, F. Schwenker, and E. Trentin. Cham: Springer International Publishing, pp. 243–254. ISBN: 978-3-319-99978-4.

Mannheim, K. (1993). "The proton blazar." *Astronomy and Astrophysics* 269, pp. 67–76.

Mao, P., C. M. Urry, E. Marchesini, et al. (2017). "Radio Luminosity Function of Flat-spectrum Radio Quasars". *The Astrophysical Journal* 842.2, p. 87.

Maraschi, L., G. Ghisellini, and A. Celotti (1992). "A Jet Model for the Gamma-Ray–emitting Blazar 3C 279". *The Astrophysical Journal Letters* 397, p. L5.

Marshall, H. L., H. Tananbaum, Y. Avni, et al. (1983). "Analysis of complete quasar samples to obtain parameters of luminosity and evolution functions". *The Astrophysical Journal* 269, pp. 35–41.

McEnery, J., A. v. d. Horst, A. Dominguez, et al. (30, 2019). "All-sky Medium Energy Gamma-ray Observatory: Exploring the Extreme Multimessenger Universe". *Bulletin of the AAS* 51.7. https://baas.aas.org/pub/2020n7i245.

McKinney, W. (2010). "Data Structures for Statistical Computing in Python". *Proceedings of the 9th Python in Science Conference*. Ed. by S. van der Walt and J. Millman, pp. 56–61.

Mészáros, P., D. B. Fox, C. Hanna, et al. (2019). "Multi-messenger astrophysics". *Nature Reviews Physics* 1.10, pp. 585–599.

Meures, T. (2019). "Upgrading the Prototype Schwarzschild-Couder Telescope Camera to a Wide-Field, High-Resolution Instrument". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 742.

Meyer, E. T., G. Fossati, M. Georganopoulos, et al. (2011). "From the Blazar Sequence to the Blazar Envelope: Revisiting the Relativistic Jet Dichotomy in Radio-loud Active Galactic Nuclei". *The Astrophysical Journal* 740.2, p. 98.

Meyer, M., J. Conrad, and H. Dickinson (2016). "Sensitivity of the Cherenkov Telescope Array to the Detection of Intergalactic Magnetic Fields". *The Astrophysical Journal* 827.2, p. 147.

Meyer, M., J. D. Scargle, and R. D. Blandford (2019). "Characterizing the Gamma-Ray Variability of the Brightest Flat Spectrum Radio Quasars Observed with the Fermi LAT". *The Astrophysical Journal* 877.1, p. 39.

Mirzoyan, R. (2017). "Detection of very-high-energy gamma-ray emission from the FSRQ Ton 0599 with the MAGIC telescopes". *The Astronomer's Telegram* 11061.

Mücke, A. and M. Pohl (2000). "The contribution of unresolved radio-loud AGN to the extragalactic diffuse gamma-ray background". *Monthly Notices of the Royal Astronomical Society* 312.1, pp. 177–193.

Mücke, A. and R. J. Protheroe (2000). "Modeling the April 1997 flare of Mkn 501". *AIP Conference Series*. Ed. by B. L. Dingus, M. H. Salamon, and D. B. Kieda. Vol. 515, pp. 149–153.

Mukherjee, R. (2001). "EGRET (GeV) Blazars". *AIP Conference Proceedings* 558.1, pp. 324–337.

— (2017). "VERITAS Detection of VHE Emission from Ton 599". *The Astronomer's Telegram* 11075.

Murase, K., Y. Inoue, and C. D. Dermer (2014). "Diffuse neutrino intensity from the inner jets of active galactic nuclei: Impacts of external photon fields and the blazar sequence". *Physical Review D* 90.2, p. 023007.

Murase, K. and E. Waxman (2016). "Constraining high-energy cosmic neutrino sources: Implications and prospects". *Physical Review D* 94.10, p. 103006.

Nalewajko, K., M. C. Begelman, and M. Sikora (2014). "Constraining the Location of Gamma-Ray Flares in Luminous Blazars". *The Astrophysical Journal* 789.2, p. 161.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Nieppola, E., E. Valtaoja, M. Tornikoski, et al. (2008). "Blazar sequence - an artefact of Doppler boosting". *Astronomy and Astrophysics* 488.3, pp. 867–872.

Nieto, D., T. Miener, A. Brill, et al. (2021). "Reconstruction of IACT events using deep learning techniques with CTLearn". *30th Astronomical Data Analysis Software and Systems*.

Nieto Castaño, D., A. Brill, Q. Feng, et al. (2019a). "CTLearn: Deep Learning for Gamma-ray Astronomy". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 752.

Nieto Castaño, D., A. Brill, Q. Feng, et al. (2019b). "Studying Deep Convolutional Neural Networks With Hexagonal Lattices for Imaging Atmospheric Cherenkov Telescope Event Reconstruction". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 753.

Nieto Castaño, D., A. Brill, B. Kim, et al. (2017). "Exploring deep learning as an event classification method for the Cherenkov Telescope Array". *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301, p. 809.

Nievas Rosillo, M. (2021). "The throughput calibration of the VERITAS telescopes". submitted to the proceedings of *37th International Cosmic Ray Conference (ICRC2021)*.

O'Brien, S. (2017). "VERITAS detection of VHE emission from the optically bright quasar OJ 287". *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301, p. 650.

Ohm, S., C. van Eldik, and K. Egberts (2009). "γ/hadron separation in very-high-energy γ-ray astronomy using a multivariate analysis method". *Astroparticle Physics* 31.5, pp. 383–391.

Olah, C. (2015). *Understanding LSTM Networks*. URL: `http://colah.github.io/posts/2015-08-Understanding-LSTMs/` (visited on 03/17/2021).

Osterbrock, D. E. and R. W. Pogge (1987). "Optical Spectra of Narrow Emission Line Palomar-Green Galaxies". *The Astrophysical Journal* 323, p. 108.

Padovani, P., F. Oikonomou, M. Petropoulou, et al. (2019). "TXS 0506+056, the first cosmic neutrino source, is not a BL Lac". *Monthly Notices of the Royal Astronomical Society* 484.1, pp. L104–L108.

Padovani, P. and C. M. Urry (1992). "Luminosity Functions, Relativistic Beaming, and Unified Theories of High-Luminosity Radio Sources". *The Astrophysical Journal* 387, p. 449.

Padovani, P. and P. Giommi (1995). "The Connection between X-Ray– and Radio-selected BL Lacertae Objects". *The Astrophysical Journal* 444, p. 567.

Palladino, A., X. Rodrigues, S. Gao, et al. (2019). "Interpretation of the Diffuse Astrophysical Neutrino Flux in Terms of the Blazar Sequence". *The Astrophysical Journal* 871.1, p. 41.

Park, E. and A. Cervantes (2019). "Long-term very high-energy gamma-ray flux variations of Markarian 421 with VERITAS". 2019 Annual Meeting of the APS Far West Section.

Parsons, R. D. and J. A. Hinton (2014). "A Monte Carlo template based analysis for air-Cherenkov arrays". *Astroparticle Physics* 56, pp. 26–34.

Parsons, R. D. and S. Ohm (2020). "Background rejection in atmospheric Cherenkov telescopes using recurrent convolutional neural networks". *European Physical Journal C* 80.5, pp. 1–11.

Parsons, R. D. and H. Schoorlemmer (2019). "Systematic differences due to high energy hadronic interaction models in air shower simulations in the 100 GeV-100 TeV range". *Physical Review D* 100.2, p. 023010.

Paszke, A., S. Gross, F. Massa, et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, et al. Curran Associates, Inc., pp. 8024–8035.

Patel, S. R. and V. R. Chitnis (2020). "Leptonic modelling of Ton 599 in flare and quiescent states". *Monthly Notices of the Royal Astronomical Society* 492.1, pp. 72–78.

Pedregosa, F., G. Varoquaux, A. Gramfort, et al. (2011). "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pence, W. D., L. Chiappetti, C. G. Page, et al. (2010). "Definition of the Flexible Image Transport System (FITS), version 3.0". *Astronomy and Astrophysics* 524, A42.

Peterson, B. M. (1997). *An Introduction to Active Galactic Nuclei*. Cambridge University Press.

Petrashyk, A. (2019). “Advancements in Very-High-Energy Gamma-Ray Astronomy with Applications to the Study of Cosmic Rays”. PhD thesis. Columbia University.

Petropoulou, M., D. Giannios, and L. Sironi (2016). “Blazar flares powered by plasmoids in relativistic reconnection”. *Monthly Notices of the Royal Astronomical Society* 462.3, pp. 3325–3343.

Poole, T. S., A. A. Breeveld, M. J. Page, et al. (2008). “Photometric calibration of the Swift ultraviolet/optical telescope”. *Monthly Notices of the Royal Astronomical Society* 383, pp. 627–645.

Prince, R. (2019). “Multi-frequency Variability Study of Ton 599 during the High Activity of 2017”. *The Astrophysical Journal* 871.1, p. 101.

— (2020). “Broadband Variability and Correlation Study of 3C 279 during Flares of 2017-2018”. *The Astrophysical Journal* 890.2, p. 164.

Protheroe, R. J. and T. Stanev (1993). “Electron-photon cascading of very high-energy gamma-rays in the infrared background.” *Monthly Notices of the Royal Astronomical Society* 264, pp. 191–200.

Qian, N. (1999). “On the momentum term in gradient descent learning algorithms”. *Neural networks* 12.1, pp. 145–151.

Radovic, A., M. Williams, D. Rousseau, et al. (2018). “Machine learning at the energy and intensity frontiers of particle physics”. *Nature* 560.7716, pp. 41–48.

Reback, J., W. McKinney, jbrockmendel, et al. (2020). *pandas-dev/pandas: Pandas 1.2.0*. Version v1.2.0. URL: `https://doi.org/10.5281/zenodo.4394318`.

Rees, M. J. (1966). “Appearance of Relativistically Expanding Radio Sources”. *Nature* 211.5048, pp. 468–470.

Reimer, A., M. Böttcher, and S. Buson (2019). “Cascading Constraints from Neutrino-emitting Blazars: The Case of TXS 0506+056”. *The Astrophysical Journal* 881.1, p. 46.

Resconi, E., D. Franco, A. Gross, et al. (2009). “The classification of flaring states of blazars”. *Astronomy and Astrophysics* 502.2, pp. 499–504.

Reynolds, P. T. (1993). “Neural Networks to VHE Gamma-Ray Atmospheric Cherenkov Crab Nebula Imaging Data”. *Irish Astronomical Journal* 21, p. 118.

Al-Rfou, R., G. Alain, A. Almahairi, et al. (2016). "Theano: A Python framework for fast computation of mathematical expressions". *arXiv e-prints* abs/1605.02688.

Righi, C., A. Palladino, F. Tavecchio, et al. (2020). "EeV astrophysical neutrinos from flat spectrum radio quasars". *Astronomy and Astrophysics* 642, A92.

Rolke, W. A., A. M. López, and J. Conrad (2005). "Limits and confidence intervals in the presence of nuisance parameters". *Nuclear Instruments and Methods in Physics Research A* 551.2-3, pp. 493–503.

Roming, P. W. A., T. E. Kennedy, K. O. Mason, et al. (2005). "The Swift Ultra-Violet/Optical Telescope". *Space Science Reviews* 120, pp. 95–142.

Roming, P. W. A., T. S. Koch, S. R. Oates, et al. (2009). "The First Swift Ultraviolet/Optical Telescope GRB Afterglow Catalog". *The Astrophysical Journal* 690, pp. 163–188.

Romoli, C., M. Zacharias, M. Meyer, et al. (2017). "Observation of the extremely bright flare of the FSRQ 3C279 with H.E.S.S. II". *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301, p. 649.

Rosenblatt, F. (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65.6, p. 386.

Roy, A. G., N. Navab, and C. Wachinger (2018). "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks". *International conference on medical image computing and computer-assisted intervention*. Springer, pp. 421–429.

Roy, N., R. Chatterjee, M. Joshi, et al. (2019). "Probing the jets of blazars using the temporal symmetry of their multiwavelength outbursts". *Monthly Notices of the Royal Astronomical Society* 482.1, pp. 743–757.

Ruder, S. (2016). "An overview of gradient descent optimization algorithms". *arXiv e-prints*, arXiv:1609.04747.

— (2017). "An Overview of Multi-Task Learning in Deep Neural Networks". *arXiv e-prints*, arXiv:1706.05098.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). "Learning representations by back-propagating errors". *Nature* 323.6088, pp. 533–536.

Russakovsky, O., J. Deng, H. Su, et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252.

Sadeh, I. (2019). "Deep learning detection of transients". *36th International Cosmic Ray Conference (ICRC2019)*. Vol. 36, p. 775.

Scargle, J. D. (1998). "Studies in Astronomical Time Series Analysis. V. Bayesian Blocks, a New Method to Analyze Structure in Photon Counting Data". *The Astrophysical Journal* 504.1, pp. 405–418.

— (2020). "Studies in Astronomical Time-series Analysis. VII. An Enquiry Concerning Nonlinearity, the rms-Mean Flux Relation, and Lognormal Flux Distributions". *The Astrophysical Journal* 895.2, p. 90.

Scargle, J. D., J. P. Norris, B. Jackson, et al. (2013). "Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations". *The Astrophysical Journal* 764.2, p. 167.

Schlafly, E. F. and D. P. Finkbeiner (2011). "Measuring Reddening with Sloan Digital Sky Survey Stellar Spectra and Recalibrating SFD". *The Astrophysical Journal* 737, p. 103.

Schmidt, M. and R. F. Green (1983). "Quasar evolution derived from the Palomar bright quasar survey and other complete quasar surveys." *The Astrophysical Journal* 269, pp. 352–374.

Schmidt, M. (1968). "Space Distribution and Luminosity Functions of Quasi-Stellar Radio Sources". *The Astrophysical Journal* 151, p. 393.

Schneider, D. P., G. T. Richards, P. B. Hall, et al. (2010). "The Sloan Digital Sky Survey Quasar Catalog. V. Seventh Data Release". *The Astronomical Journal* 139.6, p. 2360.

Seyfert, C. K. (1943). "Nuclear Emission in Spiral Nebulae." *The Astrophysical Journal* 97, p. 28.

Shah, Z., N. Mankuzhiyil, A. Sinha, et al. (2018). "Log-normal flux distribution of bright Fermi blazars". *Research in Astronomy and Astrophysics* 18.11, p. 141.

Shilon, I., M. Kraus, M. Büchele, et al. (2019). "Application of deep learning methods to analysis of imaging atmospheric Cherenkov telescopes data". *Astroparticle Physics* 105, pp. 44–53.

Shorten, C. and T. M. Khoshgoftaar (2019). "A survey on Image Data Augmentation for Deep Learning". *Journal of Big Data* 6.1.

Sikora, M., Ł. Stawarz, R. Moderski, et al. (2009). "Constraining Emission Models of Luminous Blazar Sources". *The Astrophysical Journal* 704.1, pp. 38–50.

Sinha, A., S. Sahayanathan, B. S. Acharya, et al. (2017). "On the Spectral Curvature of VHE Blazar 1ES 1011+496: Effect of Spatial Particle Diffusion". *The Astrophysical Journal* 836.1, p. 83.

Sironi, L. and D. Giannios (2014). "Relativistic Pair Beams from TeV Blazars: A Source of Reprocessed GeV Emission rather than Intergalactic Heating". *The Astrophysical Journal* 787.1, p. 49.

Sironi, L., M. Petropoulou, and D. Giannios (2015). "Relativistic jets shine through shocks or magnetic reconnection?" *Monthly Notices of the Royal Astronomical Society* 450.1, pp. 183–191.

Sironi, L. and A. Spitkovsky (2014). "Relativistic Reconnection: An Efficient Source of Non-thermal Particles". *The Astrophysical Journal Letters* 783.1, p. L21.

Smith, P. S., E. Montiel, S. Rightley, et al. (2009). "Coordinated Fermi/Optical Monitoring of Blazars and the Great 2009 September Gamma-ray Flare of 3C 454.3". *arXiv e-prints*, arXiv:0912.3621.

Spencer, S., T. Armstrong, J. Watson, et al. (2021). "Deep learning with photosensor timing information as a background rejection method for the Cherenkov Telescope Array". *Astroparticle Physics* 129, p. 102579.

Srivastava, N., G. Hinton, A. Krizhevsky, et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". *The journal of machine learning research* 15.1, pp. 1929–1958.

Stecker, F. W., O. C. de Jager, and M. H. Salamon (1996). "Predicted Extragalactic TeV Gamma-Ray Sources". *The Astrophysical Journal Letters* 473, p. L75.

Stecker, F. W. and M. H. Salamon (1996). "The Gamma-Ray Background from Blazars: A New Look". *The Astrophysical Journal* 464, p. 600.

Stern, S. A. and J. E. Colwell (1997). "Collisional Erosion in the Primordial Edgeworth-Kuiper Belt and the Generation of the 30-50 AU Kuiper Gap". *The Astrophysical Journal* 490.2, pp. 879–882.

Swanenburg, B. N., K. Bennett, G. F. Bignami, et al. (1978). "COS B observation of high-energy γ radiation from 3C273". *Nature* 275.5678, p. 298.

Szegedy, C., V. Vanhoucke, S. Ioffe, et al. (2016). "Rethinking the inception architecture for computer vision". *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Tavecchio, F., G. Bonnoli, and G. Galanti (2020). "On the distribution of fluxes of gamma-ray blazars: hints for a stochastic process?" *Monthly Notices of the Royal Astronomical Society* 497.1, pp. 1294–1300.

Tavecchio, F., G. Ghisellini, G. Bonnoli, et al. (2011). "Extreme TeV blazars and the intergalactic magnetic field". *Monthly Notices of the Royal Astronomical Society* 414.4, pp. 3566–3576.

Taylor, L. (2021). "Design and Performance of the Prototype Schwarzschild-Couder Telescope Camera". unpublished.

The HDF Group (1997). *Hierarchical Data Format, version 5*. URL: `https://www.hdfgroup.org/HDF5/`.

Ueda, Y., M. Akiyama, K. Ohta, et al. (2003). "Cosmological Evolution of the Hard X-Ray Active Galactic Nucleus Luminosity Function and the Origin of the Hard X-Ray Background". *The Astrophysical Journal* 598.2, pp. 886–908.

Urry, C. M. and P. Padovani (1991). "Altered Luminosity Functions for Relativistically Beamed Objects. II. Distribution of Lorentz Factors and Parent Populations with Complex Luminosity Functions". *The Astrophysical Journal* 371, p. 60.

Urry, C. M. and R. A. Shafer (1984). "Luminosity enhancement in relativistic jets and altered luminosity functions for beamed objects". *The Astrophysical Journal* 280, pp. 569–573.

Urry, C. M. and P. Padovani (1995). "Unified Schemes for Radio-Loud Active Galactic Nuclei". *Publications of the Astronomical Society of the Pacific* 107, p. 803.

Uttley, P., I. M. McHardy, and S. Vaughan (2005). "Non-linear X-ray variability in X-ray binaries and active galaxies". *Monthly Notices of the Royal Astronomical Society* 359.1, pp. 345–362.

Valtaoja, E., A. Lähteenmäki, H. Teräsranta, et al. (1999). "Total Flux Density Variations in Extragalactic Radio Sources. I. Decomposition of Variations into Exponential Flares". *The Astrophysical Journal Supplement Series* 120.1, pp. 95–99.

Valverde, J., D. Horan, D. Bernard, et al. (2020). "A Decade of Multiwavelength Observations of the TeV Blazar 1ES 1215+303: Extreme Shift of the Synchrotron Peak Frequency and Long-term Optical-Gamma-Ray Flux Increase". *The Astrophysical Journal* 891.2, p. 170.

Vassiliev, V., S. Fegan, and P. Brousseau (2007). "Wide field aplanatic two-mirror telescopes for ground-based γ-ray astronomy". *Astroparticle Physics* 28.1, pp. 10–27.

Vaswani, A., N. Shazeer, N. Parmar, et al. (2017). "Attention Is All You Need". *arXiv e-prints*, arXiv:1706.03762.

VERITAS Collaboration, V. A. Acciari, E. Aliu, et al. (2009). "A connection between star formation activity and cosmic rays in the starburst galaxy M82". *Nature* 462.7274, pp. 770–772.

Virtanen, P., R. Gommers, T. E. Oliphant, et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* 17, pp. 261–272.

Weekes, T. C., M. F. Cawley, D. J. Fegan, et al. (1989). "Observation of TeV Gamma Rays from the Crab Nebula Using the Atmospheric Cerenkov Imaging Technique". *The Astrophysical Journal* 342, p. 379.

Wehrle, A. E., E. Pian, C. M. Urry, et al. (1998). "Multiwavelength Observations of a Dramatic High-Energy Flare in the Blazar 3C 279". *The Astrophysical Journal* 497.1, pp. 178–187.

Wehrle, A. E., B. G. Piner, S. C. Unwin, et al. (2001). "Kinematics of the Parsec-Scale Relativistic Jet in Quasar 3C 279: 1991-1997". *The Astrophysical Journal Supplement Series* 133.2, pp. 297–320.

Weinstein, A. (2008). "The VERITAS Trigger System". *International Cosmic Ray Conference*. Vol. 3, pp. 1539–1542.

Wenger, M., F. Ochsenbein, D. Egret, et al. (2000). "The SIMBAD astronomical database. The CDS reference database for astronomical objects". *Astronomy and Astrophysics Supplement* 143, pp. 9–22.

Wilcoxon, F. (1945). "Individual Comparisons by Ranking Methods". *Biometrics Bulletin* 1.6, p. 80.

Willard, J., X. Jia, S. Xu, et al. (2020). "Integrating Physics-Based Modeling with Machine Learning: A Survey". *arXiv e-prints*, arXiv:2003.04919.

Willingale, R., R. L. C. Starling, A. P. Beardmore, et al. (2013). "Calibration of X-ray absorption in our Galaxy". *Monthly Notices of the Royal Astronomical Society* 431.1, pp. 394–404.

Winston, R. (1970). "Light Collection within the Framework of Geometrical Optics". *Journal of the Optical Society of America (1917-1983)* 6.2, pp. 245–247.

Wood, M., R. Caputo, E. Charles, et al. (2017). "Fermipy: An open-source Python package for analysis of Fermi-LAT Data". *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301, p. 824.

Yoo, S. and H. An (2020). "Spectral Variability of the Blazar 3C 279 in the Optical to X-Ray Band during 2009-2018". *The Astrophysical Journal* 902.1, p. 2.

Zeiler, M. D. (2012). "ADADELTA: An Adaptive Learning Rate Method". *arXiv e-prints*, arXiv:1212.5701.

Zitzer, B. and VERITAS Collaboration (2013). "The VERITAS Upgraded Telescope-Level Trigger Systems: Technical Details and Performance Characterization". *International Cosmic Ray Conference*. Vol. 33, p. 3076.

Zorn, J., R. White, J. J. Watson, et al. (2018). "Characterisation and testing of CHEC-M-A camera prototype for the small-sized telescopes of the Cherenkov telescope array". *Nuclear Instruments and Methods in Physics Research A* 904, pp. 44–63.

Appendix A: Complete set of *Fermi*-LAT flare profiles for 3C 279

Table A.1: LAT flare profile fits for 3C 279. The smallest variability time found is indicated in boldface. The amplitude F_0 and constant offset $F_{\rm const}$ are reported in units of 10^{-9} erg cm^{-2} s^{-1}.

F_0	$t_{\rm peak}$ [MJD]	$t_{\rm rise}$ [min]	$t_{\rm decay}$ [min]	$F_{\rm const}$
Flare 1 (MJD 56645.655 – 56647.655): χ^2/d.o.f.= 12.05/8 = 1.51				
9.56 ± 1.07	56646.330 ± 0.033	130 ± 45	674 ± 73	0.28 ± 0.06
Flare 2 (MJD 56717.655 – 56718.655): χ^2/d.o.f.= 16.63/10 = 1.66				
4.40 ± 0.70	56718.142 ± 0.043	445 ± 95	307 ± 86	0.55 ± 0.07
Flare 3 (MJD 56749.655 – 56754.655): χ^2/d.o.f.= 69.98/34 = 2.06				
7.27 ± 0.64	56750.382 ± 0.015	229 ± 25	267 ± 42	N.A.
2.78 ± 1.15	56751.238 ± 0.024	140 ± 82	69 ± 47	N.A.
4.80 ± 0.37	56752.532 ± 0.067	2001 ± 116	631 ± 136	N.A.
Flare 4 (MJD 57186.655 – 57190.655): χ^2/d.o.f.= 77.31/19 = 4.07				
12.07 ± 0.67	57187.446 ± 0.031	378 ± 46	1784 ± 147	N.A.
9.79 ± 2.29	57188.425 ± 0.028	216 ± 101	155 ± 64	N.A.
21.72 ± 1.59	57189.069 ± 0.008	137 ± 18	512 ± 55	N.A.
12.41 ± 1.30	57189.532 ± 0.010	220 ± 63	77 ± 25	N.A.
Flare 5 (MJD 58116.655 – 58119.655): χ^2/d.o.f.= 54.70/29 = 1.89				

Continued on next page

3.72 ± 0.20	58118.171 ± 0.069	1278 ± 220	2521 ± 309	0.06 ± 0.11
Flare 6 (MJD 58130.655 – 58141.655): χ^2/d.o.f.= 141.28/72 = 1.96				
7.08 ± 1.01	58134.520 ± 0.055	3719 ± 390	421 ± 259	N.A.
10.95 ± 3.74	58135.229 ± 0.053	718 ± 232	3535 ± 1394	N.A.
11.78 ± 7.00	58136.266 ± 0.048	349 ± 160	1839 ± 1055	N.A.
3.44 ± 0.66	58139.546 ± 0.033	233 ± 175	6119 ± 1824	N.A.
Flare 7 (MJD 58168.655 – 58173.655): χ^2/d.o.f.= 78.81/58 = 1.36				
2.36 ± 0.50	58172.345 ± 0.242	8540 ± 4159	4458 ± 2319	0.45 ± 0.59
Flare 8 (MJD 58222.655 – 58230.655): χ^2/d.o.f.= 177.25/106 = 1.67				
5.29 ± 1.29	58224.773 ± 0.105	1996 ± 716	5899 ± 4035	N.A.
17.70 ± 2.01	58227.945 ± 0.004	**36 ± 13**	329 ± 131	N.A.
16.42 ± 1.87	58228.323 ± 0.012	140 ± 54	115 ± 48	N.A.
5.59 ± 1.69	58227.139 ± 0.133	3816 ± 1450	4077 ± 2080	N.A.
Flare 9 (MJD 58239.655 – 58247.655): χ^2/d.o.f.= 46.25/34 = 1.36				
2.96 ± 0.40	58241.258 ± 0.149	2546 ± 595	2226 ± 1088	N.A.
3.27 ± 0.25	58245.648 ± 0.133	3080 ± 1384	3028 ± 303	N.A.
Flare 10 (MJD 58268.655 – 58275.655): χ^2/d.o.f.= 75.80/55 = 1.37				
6.23 ± 9.46	58269.171 ± 0.182	73 ± 236	177 ± 102	N.A.
8.81 ± 0.84	58270.137 ± 0.107	2392 ± 243	2449 ± 956	N.A.
4.46 ± 1.91	58271.223 ± 0.088	477 ± 431	5824 ± 862	N.A.

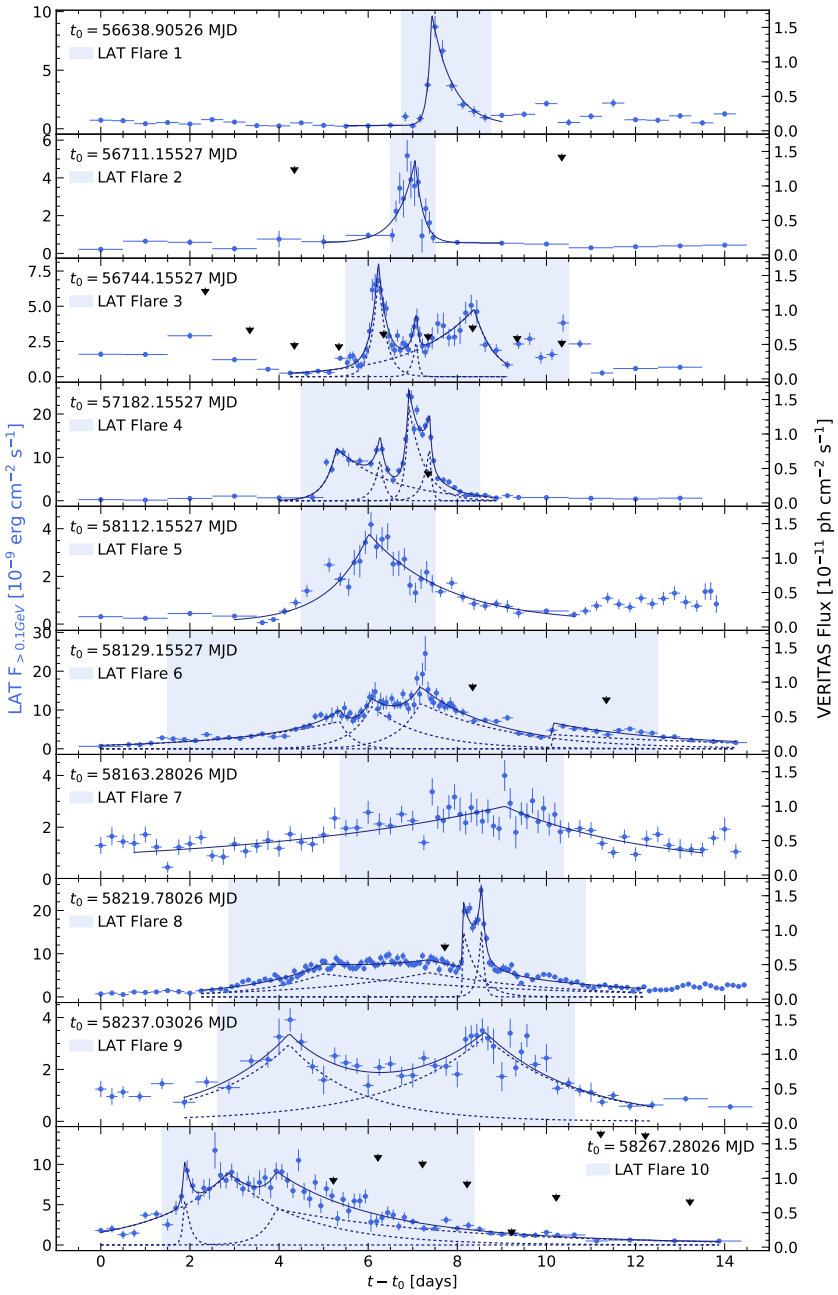


Figure A.1: 3C 279 LAT sub-daily light curves (blue points) around the flaring episodes selected as described in Section 4.4 (shaded areas). The dotted blue lines show the fitted exponential profiles, with their sums shown in solid blue. The VERITAS 95% upper limits are shown as black downwards arrows. VERITAS did not observe 3C 279 around the times of Flares 1, 5, 7, and 9.

Appendix B: *Fermi*-LAT spectral fit parameters for three FSRQs

Table B.1: *Fermi*-LAT spectral fit parameters. The normalization N_0 is in units of $10^{-10}\,\mathrm{MeV}^{-1}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ and the flux is in units of $10^{-6}\,\mathrm{ph}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

State	Epoch [MJD]	TS	N_0	α	β ($\times 10^{-2}$)	Flux
			3C 279			
Global	54682.66 – 58459.35	271945	3.33 ± 0.02	2.228 ± 0.004	6.1 ± 0.3	0.751 ± 0.004
Low state	56230.66 – 56465.66	1130	0.54 ± 0.03	2.38 ± 0.06	2.9 ± 3.1	0.14 ± 0.01
VER-LAT quiescent	Various	322	5.7 ± 0.8	2.2 ± 0.1	1.8 ± 7.0	1.4 ± 0.2
LAT Flares	56750.27 – 56750.34	578	69 ± 10	2.1 ± 0.1	31 ± 13	11 ± 2
simultaneous with	57189.17 – 57189.23	1141	173 ± 15	2.07 ± 0.07	14 ± 6	32 ± 3
VER obs.	58227.22 – 58227.27	355	79 ± 14	1.8 ± 0.2	21 ± 13	12 ± 4
	58272.18 – 58272.22	235	62 ± 15	1.5 ± 0.3	42 ± 2	7 ± 3

Continued on next page

State	Epoch [MJD]	TS	N_0	α	β ($\times 10^{-2}$)	Flux
	56645.66 – 56647.66	1633	22.0 ± 1.5	1.73 ± 0.07	9.6 ± 3.2	3.7 ± 0.3
	56717.66 – 56718.66	900	23.2 ± 2.0	2.08 ± 0.08	11 ± 6	4.5 ± 0.5
LAT	56749.66 –	7680	24.0 ± 0.8	2.20 ± 0.03	13 ± 2	5.0 ± 0.2
Flares	56754.66					
	57186.66 – 57190.66	23623	77.9 ± 1.5	2.04 ± 0.02	11 ± 1	14.7 ± 0.5
	58116.66 – 58119.66	3543	19.5 ± 0.9	2.06 ± 0.04	4.1 ± 2.2	4.0 ± 0.2
	58130.66 – 58141.66	27256	53.3 ± 0.9	2.14 ± 0.02	8.5 ± 1.1	11.0 ± 0.2
	58168.66 – 58173.66	4932	19.5 ± 0.7	2.10 ± 0.03	5.8 ± 2.0	4.1 ± 0.2
	58222.66 – 58230.66	53745	59.4 ± 0.7	2.00 ± 0.01	9.6 ± 0.7	11.2 ± 0.1
	58239.66 – 58247.66	7989	19.2 ± 0.6	1.90 ± 0.03	6.2 ± 1.4	3.6 ± 0.1
	58268.66 – 58275.66	107456	47.8 ± 1.3	1.91 ± 0.02	14 ± 2	8.2 ± 0.2

PKS 1222+216

Continued on next page

State	Epoch [MJD]	TS	N_0	α	β ($\times 10^{-2}$)	Flux
Global	54682.66 – 58459.64	94556	1.66 ± 0.01	2.305 ± 0.007	3.8 ± 0.4	0.337 ± 0.002
			Ton 599			
Global	54682.66 – 58464.49	48176	6.55 ± 0.06	2.11 ± 0.01	5.5 ± 0.5	0.161 ± 0.002

Appendix C: Selected 3HSP targets for the VERITAS luminosity function observing program

Source	3HSP Name	z	FOM	TeVCat
1ES 0120+340	3HSPJ012308.6+342048	0.270	6.31	N
B3 0133+388	3HSPJ013632.6+390559		6.31	N
RGB J0152+017	3HSPJ015239.6+014717	0.080	2.51	Y
1ES 0229+200	3HSPJ023248.6+201717	0.139	3.98	Y
RGB J0316+090	3HSPJ031612.7+090443		2.51	N
1FGL J0333.7+2919	3HSPJ033349.0+291631		2.51	N
GB6 J0540+5823	3HSPJ054030.0+582338		2.51	N
1ES 0647+250	3HSPJ065046.5+250259	0.203	7.94	Y
RGB J0710+591	3HSPJ071030.1+590820	0.120	7.94	Y
PGC 2402248	3HSPJ073326.8+515355	0.065	2.51	Y
1ES 0806+524	3HSPJ080949.2+521858	0.137	5.01	Y
87GB 083437.4+150850	3HSPJ083724.6+145820	0.278	3.98	N
RGB J0847+115	3HSPJ084712.9+113350	0.198	2.51	Y
RX J0910.6+3329	3HSPJ091037.0+332924	0.350	3.16	N
B2 0912+29	3HSPJ091552.4+293324	0.190	3.16	N
1ES 1011+496	3HSPJ101504.1+492600	0.200	7.94	Y

Continued on next page

Source	3HSP Name	z	FOM	TeVCat
1ES 1028+511	3HSPJ103118.5+505335	0.300	3.98	N
RGB J1037+571	3HSPJ103744.3+571155	0.330	3.16	N
RGB J1058+564	3HSPJ105837.7+562811	0.143	3.98	N
Mkn 421	3HSPJ110427.3+381231	0.030	79.43	Y
RXJ1117.1+2014	3HSPJ111706.3+201407	0.138	3.98	N
1ES 1218+304	3HSPJ122122.0+301037	0.180	6.31	Y
MS 1221.8+2452	3HSPJ122424.2+243623	0.218	2.51	Y
S3 1227+25	3HSPJ123014.1+251807	0.135	7.94	Y
RGB J1243+364	3HSPJ124312.7+362744	0.310	6.31	N
RBS 1366	3HSPJ141756.7+254325	0.240	3.16	N
H 1426+428	3HSPJ142832.6+424021	0.129	7.94	Y
PG 1437+398	3HSPJ143917.5+393242	0.344	2.51	N
1ES 1440+122	3HSPJ144248.2+120040	0.160	2.51	Y
PG 1553+113	3HSPJ155543.0+111124	0.443	15.85	Y
Mkn 501	3HSPJ165352.2+394536	0.030	25.12	Y
RGB J1725+118	3HSPJ172504.3+115215	0.180	5.01	Y
1ES 1727+502	3HSPJ172818.6+501310	0.055	6.31	Y
RGB J1838+480	3HSPJ183849.1+480234	0.300	3.16	N
RGB J2243+203	3HSPJ224354.7+202103		3.16	Y
B3 2247+381	3HSPJ225005.7+382437	0.119	2.51	Y

Appendix D: Installing and running `sctcamsoft`

The `sctcamsoft` code can be accessed (with privileges) at `https://forge.in2p3.fr/projects/cta/repository/show/SCT/CameraSoftware/trunk/sctcamsoft`.

All publicly-available dependencies for `sctcamsoft` can be installed using a conda-based package manager such as Anaconda[1]. To create a conda environment containing the dependencies of `sctcamsoft` using an existing Anaconda installation, run

```
conda env create -f environment.yml
```

in the root `sctcamsoft` directory. `TargetDriver` and `TargetIO` should then be installed into that environment. Once all dependencies are installed, install `sctcamsoft` into the same environment by running

```
source activate sctcamsoft
cd </installation/path>/sctcamsoft
pip install -e .
cd </installation/path>/sctcamsoft/sctcamsoft
protoc -I=. --python_out=. ./camera_control.proto
```

where the `-e` flag to `pip install` installs the package in editable mode, and can be omitted for a production (non-development) installation.

To run `sctcamsoft`, the server must first be started by running

```
source activate sctcamsoft
```

[1] `https://www.anaconda.com/`

```
cd sctcamsoft
sudo $PYTHON3 sctcamsoft/server.py config.yml commands.yml
```

where calling `sudo` is required to perform network monitoring commands implemented in NetworkController. In `sudo` mode, it is necessary to explicitly invoke the Python installation in the `sctcamsoft` conda environment (called `$PYTHON3` above).

To start the command-line user interfaces, run in separate terminals:

```
source activate sctcamsoft
cd sctcamsoft
python sctcamsoft/user_input.py config.yml commands.yml

source activate sctcamsoft
cd sctcamsoft
python sctcamsoft/user_output.py config.yml
```

To start the graphical user interfaces, run in separate terminals:

```
source activate sctcamsoft
cd sctcamsoft
python sctcamsoft/slow_control_gui/main.py config.yml commands.yml

source activate sctcamsoft
cd sctcamsoft
python sctcamsoft/run_control_gui/main.py config.yml commands.yml
```

When observing is complete, the server and command-line interfaces (if active) can be closed by pressing `Ctrl+C`.

Appendix E: Low-level commands implemented in `sctcamsoft`

The commands in Table E.1 are organized by the DeviceController that implements them and are generally named according to the pattern "`command.device/command.command`". For ServerController, `command.device` is "server" and `command.command` is the same as the command name. The "Special Commands" have no direct implementation but are used within ServerController to guide the control flow of high-level commands. In those cases, `command.device` is `None` and `command.command` is the same as the command name. The Args column lists in order the arguments required by each command (if any) and their default values if the argument is not provided. If no default is listed, the argument is required.

Table E.1: Low-level commands implemented in `sctcamsoft`.

Name	**Args**
BackplaneController	
`backplane/enable_tack`	
`backplane/disable_tack`	
`backplane/power_fee`	slot *(0-31)*
`backplane/power_off_all_fees`	
`backplane/power_on_all_fees`	
`backplane/reboot_dacq_1`	
`backplane/reboot_dacq_2`	
`backplane/read_current`	
`backplane/read_presence`	
`backplane/read_voltage`	
`backplane/read_hit_pattern`	display: `False`
`backplane/read_tack_count`	
`backplane/read_timer`	
`backplane/reset_triggers_and_timer`	
`backplane/send_sync`	
`backplane/set_hold_off_time`	

Table E.1 – continued from previous page

Name	Args
backplane/set_trigger_mask	
backplane/set_trigger_mask_closed	
ChillerController	
chiller/read_pressure	
chiller/read_temperature	
FanController	
fan/open_connection	
fan/close_connection	
fan/check_connection	
fan/turn_on	
fan/turn_off	
fan/read_voltage	
fan/read_current	
FEETemperatureController	
fee_temperature/get_last_run_num	
fee_temperature/read_temperature	run_num: <last_run_num>
NetworkController	
network/check_interface_activity	interface *(6-9)*
network/set_camera_server_module_addresses	
network/set_dacq_module_addresses	
PowerController	
power/turn_on_main_switch	
power/turn_off_main_switch	
power/start_supply	
power/stop_supply	
power/start_HV	
power/stop_HV	
power/read_supply_current	
power/read_supply_measured_voltage	
power/read_supply_nominal_voltage	
power/read_HV_current	
power/read_HV_measured_voltage	
power/read_HV_nominal_voltage	
ServerController	

Table E.1 – continued from previous page

Name	Args
write_update	variable
	value
set_alert	name
	device
	variable
	lower_limit
	upper_limit
unset_alert	name
issue_alert	name
	value
	value_id
	lower_limit
	upper_limit
	unit: “”
clear_alert	name
modify_repeating_command	name
	arg
	value
stop_repeating_command	name
	no_command_is_error: True
TargetController	
target/initialize	module_id: “all”
target/connect	module_id: “all”
target/close	module_id: “all”
target/ready	module_id: “all”
target/ping	module_id: “all”
target/activate_adc	module_id: “all”
target/deactivate_adc	module_id: “all”
target/set_thresh	module_id: “all”
target/set_asic_parameters	module_id: “all”
target/set_tuning_parameters	module_id: “all”
target/set_readout_parameters	module_id: “all”
target/read_state	module_id: “all”
target/read_temperature	module_id: “all”
target/read_currents	module_id: “all”
target/read_total_current	module_id: “all”
target/request_asic_parameter	parameter
	value

Table E.1 – continued from previous page

Name	Args
`target/request_readout_parameter`	parameter value
`target/request_tuning_parameter`	parameter value
`target/request_adc_parameter`	parameter value
`target/request_rate_scan_parameter`	parameter value
`target/get_asic_parameters`	
`target/get_readout_parameters`	
`target/get_tuning_parameters`	
`target/get_adc_parameters`	
`target/get_rate_scan_parameters`	
`target/enable_module`	module_id
`target/disable_module`	module_id
`target/enable_trigger_module`	module_id
`target/disable_trigger_module`	module_id
`target/get_module_ids`	
`target/get_trigger_module_ids`	
`target/write_trigger_mask`	
`target/initialize_run`	run_id
`target/start_readout`	
`target/stop_readout`	
`target/read_trigger_rate`	
`target/get_run_results`	
`target/read_register`	register
Special Commands	
`enter_repeat_mode`	name interval *(seconds)* num_executions: 0 *(repeat forever)* execute_immediately: `False`
`exit_repeat_mode`	