A Hadronic Blazar Model: Spectra, Numerical Methods, and VERITAS Comparison

1 Introduction

Blazars are active galactic nuclei whose relativistic jets are oriented close to the line of sight, producing strong emission across the electromagnetic spectrum. In hadronic blazar scenarios, relativistic protons undergo photohadronic interactions ($p\gamma$) with the ambient synchrotron photon field. These interactions produce pions, which subsequently decay into neutrinos and gamma rays. This work presents a simple model of hadronic blazar emission and demonstrates how to fold the resulting gamma-ray spectra through a VERITAS-like effective area. We also describe the numerical methods used in the implementation.

2 Proton Spectrum Model

We assume a proton injection spectrum of the form:

$$\frac{dN_p}{dE_p} = N_0 E_p^{-s} \exp\left(-\frac{E_p}{E_{p,\mathrm{cut}}}\right), \tag{1}$$

where s is the spectral index and $E_{p,\mathrm{cut}}$ is an exponential cutoff energy. The exponential cutoff ensures the spectrum falls sharply above a maximum acceleration energy, modeling limits of the acceleration process.

The normalization N_0 is chosen so that the bolometric proton luminosity matches a specified value L_p:

$$L_p = \int_{E_{p,\mathrm{min}}}^{E_{p,\mathrm{max}}} E_p \frac{dN_p}{dE_p}\, dE_p. \tag{2}$$

In the code, this integral is computed numerically using Simpson integration to solve for N_0.

3 Photohadronic Interaction Efficiency

The $p\gamma$ interaction efficiency is approximated with a simple parametric form:

$$f_{p\gamma}(E_p) = f_0 \left(\frac{E_p}{E_0}\right)^{\beta}, \tag{3}$$

where f_0 sets the normalization, β controls the energy dependence, and E_0 is a reference energy. This captures the physical idea that higher-energy protons interact more efficiently with target photons.

The inelasticity $\kappa \simeq 0.2$ represents the fraction of proton energy transferred into pion production per interaction, a widely used typical value from standard photohadronic cross-section studies.

4 Pion Production and Energy Mapping

A fraction $f_{p\gamma}\kappa$ of the proton energy goes into pion production. We assume:

- $\frac{2}{3}$ of pion energy goes to charged pions,
- $\frac{1}{3}$ goes to neutral pions.

Charged pions decay into neutrinos, and neutral pions decay into gamma rays. Using approximate energy fractions:

$$E_\nu \approx 0.05 E_p, \tag{4}$$

$$E_\gamma \approx 0.1 E_p. \tag{5}$$

To transfer the energy injection rate from proton energy bins into neutrino or gamma energy bins, a numerical "mapping" function assigns each proton-energy contribution to the appropriate target-energy bin.

5 Differential Luminosities and Fluxes

The differential luminosity is computed from the energy injection rate via:

$$\frac{dL}{dE} = \frac{\dot{E}_{\rm inj}}{\Delta E}. \tag{6}$$

In the code, ΔE is computed using NumPy's `gradient` function:

$$\Delta E_i \approx E_{i+1} - E_i, \tag{7}$$

which is a finite-difference method estimating bin widths in a non-uniform energy grid.

The flux at Earth is then

$$F_E = \frac{1}{4\pi D_L^2} \frac{dL}{dE}, \tag{8}$$

where D_L is the luminosity distance.

6 Time-Dependent Flare

A Gaussian flare model is used:

$$F(t) = \exp\left[-\frac{1}{2}\left(\frac{t - t_0}{\sigma_t}\right)^2\right]. \tag{9}$$

This scaling multiplies the instantaneous spectra to simulate a blazar flare.

7 VERITAS-like Effective Area

To convert a theoretical gamma-ray flux into predicted observable counts, the model is folded through an approximate effective area:

$$A_{\mathrm{eff}}(E) = A_{\max}\frac{(E/E_0)^2}{1+(E/E_0)^2}. \tag{10}$$

This placeholder function captures the basic behavior of an Imaging Atmospheric Cherenkov Telescope (IACT):

- low sensitivity below threshold,
- rapid rise to peak area,
- saturation at high energies.

The expected counts are:

$$N_\gamma = \int \frac{dN_\gamma}{dE} A_{\mathrm{eff}}(E) T_{\mathrm{obs}}\, dE, \tag{11}$$

numerically computed using Simpson integration.

Numerical Methods Used in the Script

This project uses several numerical techniques to evaluate particle spectra, convert luminosities into fluxes, map proton energies to secondary particle energies, and estimate observable gamma-ray counts. Each of the numerical methods described below is widely used in computational astrophysics and is well suited for the smooth, monotonic spectra that arise in hadronic blazar models.

1. Logarithmic Energy Grids

High-energy astrophysical spectra span many orders of magnitude in energy. A linear grid would waste resolution at low energies and undersample high-energy regions where the spectrum varies quickly. Therefore, the script defines proton, neutrino, and gamma-ray energy arrays using logarithmic spacing:

$$E_{\mathrm{grid}} = 10^{\mathrm{linspace}(\log_{10} E_{\min},\, \log_{10} E_{\max})}.$$

A log grid provides:

- Uniform resolution in decades of energy.
- Smooth integration behavior for power laws.
- Better accuracy for spectra of the form E^{-s}.

2. Simpson's Rule for Integrals

To compute the proton luminosity normalization and to estimate the gamma-ray counts, the script uses SciPy's implementation of Simpson's rule:

$$\mathrm{simps}(f(E), E).$$

Simpson's rule works by fitting a quadratic polynomial to every pair of adjacent grid intervals, giving higher accuracy than the trapezoidal rule for smooth functions. This is ideal because power-law spectra

$$E^{-s} \exp(-E/E_{\mathrm{cut}})$$

are smooth and differentiable.

Simpson's rule is preferred because:

- It is more accurate for smooth, monotonic spectra.
- It reduces numerical noise in luminosity calculations.
- It converges faster than the trapezoidal method for similar grid sizes.

3. Difference Between Simpson and Trapezoidal Rule

The trapezoidal rule approximates the integrand with straight lines. Simpson's rule approximates it with parabolas.

- Trapezoidal rule: good for rough estimates; linear approximation.
- Simpson's rule: much better for smooth curves; quadratic approximation.

For spectra like E^{-s}, the curvature is significant, so Simpson's rule yields smaller numerical error.

4. Finite-Difference Derivatives Using `np.gradient`

To convert binned energy-injection values into differential spectra dL/dE, the script uses NumPy's finite-difference derivative:

$$\frac{\partial L}{\partial E} \approx \frac{\Delta L}{\Delta E}.$$

This is implemented via:

```
dE = np.gradient(E_grid).
```

This method:

- Approximates derivatives using neighboring grid points.
- Is stable and accurate on log-spaced grids.
- Allows conversion from "energy per bin" to a continuous differential luminosity.

5. Delta-Function Energy Mapping

The script uses a custom function to map proton energies to neutrino or gamma-ray energies using a fixed scaling factor:

$$E_\nu = 0.05\, E_p, \qquad E_\gamma = 0.1\, E_p.$$

This assumes delta-function energy transfer, which simplifies the physics while retaining the correct order of magnitude. The mapping procedure:

1. Computes $E_t = m\, E_p$.
2. Locates the corresponding index in the target energy grid.
3. Adds the energy-injection contribution to that bin.

This approach is efficient and sufficiently accurate for a toy hadronic model.

6. Maximization via `np.argmax`

To locate the time of the flare peak, the script computes:

$$t_{\text{peak}} = \texttt{t\_eval[np.argmax(flare\_profile(t\_eval))]}.$$

`np.argmax` returns the index of the largest value in the flare profile array. This is a reliable and fast way to identify the peak of a Gaussian flare.

7. Numerical Integration of Expected Gamma-Ray Counts

The expected number of gamma-ray detections is:

$$N_\gamma = \int dE\, \frac{dN_\gamma}{dE}\, A_{\text{eff}}(E)\, T_{\text{obs}}.$$

This integral is evaluated using Simpson's rule, which is well suited because both the flux and effective area are smooth functions of energy.

The final result gives the predicted photon counts for a VERITAS-like instrument during the simulated flare.