import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import pearsonr

# Parameters

grid\_size = 256

cmb\_center = (-57, 209)    # Latitude, Longitude of cold spot

mbt\_center = (-70, 180)    # MBT curvature root

max\_radius = 50            # degrees

num\_rings = 10

# Generate synthetic fields

def generate\_gaussian\_field(center, amplitude, spread):

    lat = np.linspace(-90, 90, grid\_size)

    lon = np.linspace(0, 360, grid\_size)

    Lon, Lat = np.meshgrid(lon, lat)

    r2 = (Lat - center[0])\*\*2 + (Lon - center[1])\*\*2

    field = amplitude \* np.exp(-r2 / spread)

    return field, Lat, Lon

cmb\_field, Lat, Lon = generate\_gaussian\_field(cmb\_center, amplitude=-1.0, spread=500)

mbt\_field, \_, \_ = generate\_gaussian\_field(mbt\_center, amplitude=1.0, spread=300)

# Compute radial ring profiles

ring\_edges = np.linspace(0, max\_radius, num\_rings + 1)

cmb\_profile = []

mbt\_profile = []

for i in range(num\_rings):

    r\_min, r\_max = ring\_edges[i], ring\_edges[i + 1]

    r2\_cmb = (Lat - cmb\_center[0])\*\*2 + (Lon - cmb\_center[1])\*\*2

    r2\_mbt = (Lat - mbt\_center[0])\*\*2 + (Lon - mbt\_center[1])\*\*2

    mask\_cmb = (r2\_cmb >= r\_min\*\*2) & (r2\_cmb < r\_max\*\*2)

    mask\_mbt = (r2\_mbt >= r\_min\*\*2) & (r2\_mbt < r\_max\*\*2)

    cmb\_profile.append(np.mean(cmb\_field[mask\_cmb]))

    mbt\_profile.append(np.mean(mbt\_field[mask\_mbt]))

# Compute correlation

corr, \_ = pearsonr(cmb\_profile, mbt\_profile)

# Plot profiles

plt.figure(figsize=(8, 5))

r\_centers = 0.5 \* (ring\_edges[:-1] + ring\_edges[1:])

plt.plot(r\_centers, cmb\_profile, label='CMB Temperature', color='blue')

plt.plot(r\_centers, mbt\_profile, label='MBT Curvature', color='orange')

plt.xlabel("Radial Distance from Center (°)")

plt.ylabel("Mean Field Value")

plt.title(f"Ring Profile Comparison (Pearson r = {corr:.2f})")

plt.legend()

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.show()

I then did this