import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import pearsonr

# Parameters

grid\_size = 256

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

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

max\_radius = 50

num\_rings = 10

num\_trials = 10000

# Generate MBT curvature field

def generate\_mbt\_field(center, amplitude=1.0, spread=300):

    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 / (1 + spread \* r2 / 10000)

    return field

# Compute ring profile

def compute\_ring\_profile(field, center):

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

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

    Lon, Lat = np.meshgrid(lon, lat)

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

    profile = []

    for i in range(num\_rings):

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

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

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

        profile.append(np.mean(field[mask]))

    return profile

# Generate reference CMB profile (synthetic)

cmb\_field = generate\_mbt\_field(cmb\_center, amplitude=-1.0, spread=500)

cmb\_profile = compute\_ring\_profile(cmb\_field, cmb\_center)

# Run randomized MBT rotations

correlations = []

for \_ in range(num\_trials):

    rand\_lat = np.random.uniform(-90, 90)

    rand\_lon = np.random.uniform(0, 360)

    mbt\_field = generate\_mbt\_field((rand\_lat, rand\_lon))

    mbt\_profile = compute\_ring\_profile(mbt\_field, cmb\_center)

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

    correlations.append(r)

# Plot histogram

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

plt.hist(correlations, bins=50, color='lightgreen', edgecolor='black')

plt.axvline(-0.98, color='red', linestyle='--', label='Observed MBT-CMB Correlation (−0.98)')

plt.xlabel("Pearson Correlation Coefficient")

plt.ylabel("Number of Random Rotations")

plt.title("MBT Field Rotation: Correlation Distribution")

plt.legend()

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.show()

# Print tail probability

extreme\_hits = np.sum(np.array(correlations) <= -0.98)

p\_value = extreme\_hits / num\_trials

print(f"Fraction of MBT rotations with r ≤ −0.98: {p\_value:.4f} ({extreme\_hits} out of {num\_trials})")