import numpy as np

import matplotlib.pyplot as plt

# --- PARAMETERS ---

N = 32 # Field size

timesteps = 100 # Total steps

error\_time = 40 # When to inject errors

error\_frac = 0.05 # Fraction of points to scramble

healing\_steps = 40 # Steps after error for healing

# --- INITIALIZE FIELDS ---

np.random.seed(42)

phase = np.random.uniform(0, 2\*np.pi, (N,N))

memory = np.cos(phase)

order\_curve = []

# --- PLOT SETUP ---

fig, axs = plt.subplots(1, 3, figsize=(15,4))

plt.tight\_layout()

def order\_param(mem):

# Simple MBT logic: coherence = mean absolute memory

return np.abs(mem).mean()

# --- EVOLUTION LOOP ---

for t in range(timesteps):

# Local update: mix phase w/ nearest neighbors (MBT motion principle)

for i in range(N):

for j in range(N):

neighbors = [phase[(i+1)%N,j], phase[(i-1)%N,j], phase[i,(j+1)%N], phase[i,(j-1)%N]]

avg = np.mean(neighbors)

phase[i,j] = 0.7\*phase[i,j] + 0.3\*avg

# MBT memory = cos(phase)

memory = np.cos(phase)

# Inject noise

if t == error\_time:

n\_err = int(error\_frac \* N \* N)

idx = np.random.choice(N\*N, n\_err, replace=False)

for k in idx:

x, y = k//N, k%N

phase[x,y] = np.random.uniform(0, 2\*np.pi)

error\_mask = np.zeros((N,N))

error\_mask.flat[idx] = 1

order\_curve.append(order\_param(memory))

# --- PLOTTING ---

# 1. Phase field (final)

axs[0].imshow(np.cos(phase), cmap='twilight')

axs[0].set\_title("Final MBT Phase Field (cos)")

# 2. Memory field (final, errors marked)

axs[1].imshow(memory, cmap='inferno')

if 'error\_mask' in locals():

err\_x, err\_y = np.where(error\_mask)

axs[1].scatter(err\_y, err\_x, marker='o', color='cyan', label='Error Hits')

axs[1].set\_title("Final MBT Memory Field (errors/healed)")

axs[1].legend(loc='lower left')

# 3. Order curve

axs[2].plot(order\_curve, label='Logic Order')

axs[2].axvline(error\_time, color='red', linestyle=':', label='Errors Added')

axs[2].set\_title("MBT Logic Memory Recovery")

axs[2].set\_xlabel("Timestep")

axs[2].set\_ylabel("Order (coherence)")

axs[2].legend()

plt.show()