import numpy as np

import matplotlib.pyplot as plt

# Settings

timesteps = 200

collapse = np.linspace(0, 1, timesteps)

healing = np.linspace(1, 0, timesteps)

traj = np.concatenate([collapse, healing[1:]])

x = np.arange(len(traj))

S\_quantum = 2.828

S\_classical = 2.0

memory\_perfect = 1 - np.abs(traj - 1 \* (traj > 1))

S\_clean = S\_quantum - (S\_quantum - S\_classical) \* traj

memory\_clean = 1 - traj

noise\_levels = [0, 0.1, 0.3, 0.6]

fig, axs = plt.subplots(2, 2, figsize=(14, 8))

for i, noise in enumerate(noise\_levels):

ax = axs.flat[i]

np.random.seed(42 + i)

S\_noisy = S\_clean + np.random.normal(0, noise, size=len(S\_clean))

memory\_noisy = memory\_clean + np.random.normal(0, noise, size=len(memory\_clean))

memory\_noisy = np.clip(memory\_noisy, 0, 1)

ax.plot(x, S\_noisy, label='Bell S (MBT, Noisy)', color='royalblue', alpha=0.7)

ax.plot(x, memory\_noisy, label='MBT Memory (Noisy)', color='green', alpha=0.7)

ax.plot(x, S\_clean, '--', color='grey', label='Bell S (Ideal)', alpha=0.5)

ax.axhline(S\_quantum, ls='--', color='purple', label='Quantum Limit', alpha=0.6)

ax.axhline(S\_classical, ls='--', color='black', label='Classical Limit', alpha=0.6)

ax.set\_ylim(1.6, 3)

ax.set\_title(f"Noise level = {noise:.1f}")

ax.set\_xlabel("Time (collapse → healing)")

if i % 2 == 0:

ax.set\_ylabel("Order parameter (S) / MBT Memory")

if i == 0:

ax.legend(loc='lower right')

plt.suptitle("MBT Quantum Order & Memory vs. Noise/Collapse/Healing")

plt.tight\_layout(rect=[0, 0, 1, 0.96])

plt.show()