Empirical Investigation of an Open Conjecture: Can Single-Shuffle SGD be Better than Reshuffling SGD and GD?

Agentic NL→Lean 4 Pipeline
Job #48

April 26, 2026

Abstract

This report documents the empirical investigation of an open mathematical conjecture that could not be formally proved or disproved in Lean 4 with Mathlib. Numerical experiments were conducted to gather evidence for or against the conjecture. The empirical verdict is: **Empirically Refuted**. The conjecture remains formally open.

1 Conjecture Statement

Conjecture 1.

```
Can Single-Shuffle SGD be Better than Reshuffling SGD and GD?

Let n >= 2, K >= 1, and d >= 1. Let S_n denote the set of all permutations
    of {1,...,n}. For real symmetric d x d matrices X, Y, write X   Y for the
    Loewner order (i.e., Y - X is positive semidefinite), and let ||·||_2 be
    the spectral (operator) norm.

Given symmetric matrices A_1, ..., A_n, define for each    S_n the without-
    replacement product  P_ := _{i=1}^{n} A_ {(i)} = A_ {(n)}   A_ {(1)}.

Define:
  W_SS := (1/n!) Σ_ {   S_n} ( P_)^K        (single-shuffle)
  W_RS := ((1/n!) Σ_ {   S_n}  P_)^K         (random reshuffling)
  W_GD := ((1/n) Σ_{i=1}^{n} A_i)^{nK}       (gradient descent)

CONJECTURE: For every n >= 2 and K >= 1 there exists a constant _{n,K}
    (0,1] (depending only on n and K, not on d or on the specific matrices)
    such that, whenever (1 - _{n,K})I   A_i   I for all i   {1,...,n}, one has
    :

  ||W_SS||_2   ||W_RS||_2   ||W_GD||_2

Equivalently: under uniform near-identity well-conditioning, single-shuffle
    never yields a larger spectral norm than random reshuffling, and random
    reshuffling is never worse than the full-gradient proxy W_GD, with
    constants uniform over dimension and instances.
```

2 Status

Formal Status: OPEN — no Lean 4 proof or disproof was found.
Empirical Verdict: Empirically Refuted

The pipeline attempted formal verification in Lean 4 with Mathlib but was unable to produce a compiling proof or disproof. Empirical testing was then conducted to gather numerical evidence.

3 Basic Empirical Testing

The following output was produced by the basic numerical experiment:

```
=== EXPERIMENT PLAN ===

Conjecture: With (1 - eta_{n,K}) I  <=  A_i  <=  I for symmetric A_i, define
  W_SS := (1/n!) sum_sigma  (A_{sigma(n)} ... A_{sigma(1)})^K
  W_RS := ((1/n!) sum_sigma   A_{sigma(n)} ... A_{sigma(1)})^K
  W_GD := ((1/n) sum_i A_i)^{nK}
The conjecture predicts ||W_SS||_2 <= ||W_RS||_2 <= ||W_GD||_2 for some
eta_{n,K} in (0,1], uniformly in d and in the matrices.

Tests we run:
1) Direct random sampling: many (n,K,d,eta) configurations with random
   symmetric matrices having eigenvalues in [1-eta, 1]. Count violations
   of (a) ||W_SS|| <= ||W_RS||  and  (b) ||W_RS|| <= ||W_GD||.
2) Eta-sweep: for fixed (n,K) shrink eta toward 0 to see whether the
   violation rate vanishes (predicted by the conjecture).
3) Counterexample search: adversarial sampling concentrating eigenvalues
   near the boundary 1-eta and using many random orthogonal frames.
4) Edge cases: commuting A_i (must give equality), A_i = I (trivial),
   K=1 large n (here W_SS = W_RS exactly), d=1 (scalars).
5) Scaling with K and n: track maximum observed violation as parameters
   change.
6) Statistical test: binomial test on whether observed violation rate
   for very small eta is consistent with zero.


--- Test 1: Random sampling sweep ---
Total trials: 5760
Violations ||W_SS|| > ||W_RS|| : 0 (0.000%)
Violations ||W_RS|| > ||W_GD|| : 0 (0.000%)

--- Test 2: Eta-sweep ---
eta= 0.500  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
   )=+0.000e+00
eta= 0.300  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
   )=+0.000e+00
eta= 0.200  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
   )=+0.000e+00
eta= 0.150  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
   )=+0.000e+00
eta= 0.100  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
   )=+0.000e+00
```

```
eta= 0.070  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
    )=+0.000e+00
eta= 0.050  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
    )=+0.000e+00
eta= 0.030  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
    )=+0.000e+00
eta= 0.020  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
    )=+0.000e+00
eta= 0.010  P(SS>RS)=0.000  P(RS>GD)=0.000  max(SS-RS)=+0.000e+00  max(RS-GD
    )=+0.000e+00

--- Test 3: Adversarial counterexample search ---
Adversarial trials: 6000
  ||W_SS||>||W_RS|| violations: 0, worst gap = -inf
  ||W_RS||>||W_GD|| violations: 0, worst gap = -inf

--- Test 4: Edge cases ---
Commuting (diagonal): SS=0.419257 RS=0.419257 GD=0.420877
  SS==RS? max diff = 5.55e-17
  RS<=GD? gap = +1.620e-03
All identity:  SS=1.000000 RS=1.000000 GD=1.000000
d=1 scalars:   SS=0.164275 RS=0.164275 GD=0.199590
K=1 sanity: SS=0.6818211762  RS=0.6818211762  diff=0.00e+00

--- Test 5: Scaling with n and K ---
n=2 K=1: worst(SS-RS)=+0.000e+00  worst(RS-GD)=-4.489e-06
n=2 K=2: worst(SS-RS)=-3.598e-10  worst(RS-GD)=-3.214e-05
n=2 K=3: worst(SS-RS)=-4.428e-08  worst(RS-GD)=-4.611e-04
n=2 K=5: worst(SS-RS)=-1.145e-08  worst(RS-GD)=-4.016e-04
n=2 K=8: worst(SS-RS)=-2.857e-07  worst(RS-GD)=-2.805e-04
n=3 K=1: worst(SS-RS)=+0.000e+00  worst(RS-GD)=-4.263e-04
n=3 K=2: worst(SS-RS)=-2.793e-07  worst(RS-GD)=-9.055e-04
n=3 K=3: worst(SS-RS)=-2.281e-07  worst(RS-GD)=-1.894e-03
n=3 K=5: worst(SS-RS)=-1.130e-06  worst(RS-GD)=-1.497e-03
n=3 K=8: worst(SS-RS)=-1.722e-06  worst(RS-GD)=-5.979e-04
n=4 K=1: worst(SS-RS)=+0.000e+00  worst(RS-GD)=-1.142e-03
n=4 K=2: worst(SS-RS)=-1.518e-06  worst(RS-GD)=-1.640e-03
n=4 K=3: worst(SS-RS)=-1.017e-06  worst(RS-GD)=-7.942e-04
n=4 K=5: worst(SS-RS)=-5.381e-07  worst(RS-GD)=-1.709e-03
n=4 K=8: worst(SS-RS)=-2.340e-06  worst(RS-GD)=-1.133e-03

--- Test 6: Statistical test, very small eta ---
At eta=0.02, n=3, K=3, d=4, trials=4000:
  P(SS>RS) = 0.0000
  P(RS>GD) = 0.0000
  Binomial test SS>RS p-value vs ~0: 1.000e+00
  Binomial test RS>GD p-value vs ~0: 1.000e+00

=== SUMMARY ===
Total trials (across all phases): ~15760
Total ||W_SS||>||W_RS|| violations: 0
Total ||W_RS||>|
... [truncated]
```

4 Advanced Empirical Testing

A research-grade experiment was designed with nonlinear analysis, parameter sweeps, and convergence testing. Output:

```
=== ADVANCED EXPERIMENT PLAN ===

Conjecture (COLT-2021 open problem). For symmetric A_i with
   (1 - eta_{n,K}) I   A_i   I,
the spectral norms satisfy
   ||W_SS||_2    ||W_RS||_2    ||W_GD||_2,
where
   W_SS = (1/n!) Σ_ ( P_)^K            (single-shuffle)
   W_RS = ( (1/n!) Σ_  P_ )^K          (random-reshuffling)
   W_GD = ( (1/n) Σ_i A_i )^{nK}        (full-batch GD).

This advanced experiment goes BEYOND random sampling by combining:

 (A) GRADIENT-BASED ADVERSARIAL SEARCH (L-BFGS-B, multistart) over the
     nonconvex constraint manifold A_i   [(1-)I, I] using a smooth
     eigenvalue-sigmoid parameterization. Random Monte Carlo almost
     never hits worst cases in non-convex matrix optimization; this
     actively HUNTS for counterexamples.
 (B) SYMBOLIC EXACT verification via sympy of the K=1 identity
     (W_SS = W_RS) and the →0 Taylor structure of W_RS - W_SS
     and W_GD - W_RS for K=2 (showing the leading order is PSD).
 (C) DIMENSIONAL CONVERGENCE STUDY in  d{2,4,8,16,32}: if the
     conjecture is true the worst observed violation should stay at
     floating-point noise, independent of d.
 (D)  -BOUNDARY DETECTION: probe  >1 (allowing INDEFINITE A_i, i.e.
     OUTSIDE the conjecture's hypothesis) to confirm the bound is
     tight - violations should appear there but not for  1.
 (E) STRUCTURED ENSEMBLES (GOE-projected, near-commuting, low-rank
     perturbation, near-degenerate) to test robustness across
     statistical models.
 (F) OPERATOR INVARIANT MONITORING (Loewner-cone consistency of
     traces and Frobenius norms; "energy "conservation surrogate).
 (G) MULTI-PRECISION sanity check on the worst case found by (A)
     using sympy's exact rational arithmetic to rule out f.p. noise.

EXPECTED if TRUE  : adversarial best gap   ~1e-12; symbolic identities
                    confirm; gap independent of d; violations only when
                     > 1 (hypothesis broken).
EXPECTED if FALSE : (A) finds a positive gap in  [(1-)I, I]; (G)
                    promotes it to certified counterexample.


--- (A) ADVERSARIAL L-BFGS-B COUNTEREXAMPLE HUNT ---
  n=2 d=3 K=2  =1.00  best(SS-RS)=-6.838e-10  best(RS-GD)=-2.494e-10
  n=2 d=4 K=3  =1.00  best(SS-RS)=-3.731e-08  best(RS-GD)=-8.627e-09
  n=2 d=4 K=3  =0.50  best(SS-RS)=-2.863e-08  best(RS-GD)=-9.976e-09
  n=3 d=3 K=2  =1.00  best(SS-RS)=+3.751e-03  best(RS-GD)=-1.577e-10
  n=3 d=3 K=3  =0.50  best(SS-RS)=-1.304e-07  best(RS-GD)=-4.813e-10
  n=3 d=4 K=2  =0.20  best(SS-RS)=-1.605e-06  best(RS-GD)=-1.727e-07
```

```
   overall worst adversarial gap = +3.751e-03 at (3, 3, 2, 1.0)

--- (B) SYMBOLIC sympy VERIFICATION ---
  K=1 identity (W_SS - W_RS = 0) symbolic check: True
  scalar n=2,K=2: SS-RS = 0 (should be 0 in commuting case)
  scalar n=2,K=2: GD-RS Taylor in  = -b1**3*eta**3/2 + b1**2*b2*eta**3/2 +
     b1**2*eta**2/2 + b1*b2**2*eta**3/2 - b1*b2*eta**2 - b2**3*eta**3/2 + b2
     **2*eta**2/2
  coeff of  ^2 in GD-RS  = b1**2/2 - b1*b2 + b2**2/2  (must be  0 since (b1-b2
     )^2/4)

--- (C) DIMENSIONAL CONVERGENCE (worst gap vs d) ---
  d= 2  worst(SS-RS)=-4.627e-09  worst(RS-GD)=-2.516e-05
  d= 4  worst(SS-RS)=-4.264e-06  worst(RS-GD)=-6.351e-04
  d= 8  worst(SS-RS)=-4.179e-05  worst(RS-GD)=-3.939e-03
  d=16  worst(SS-RS)=-2.087e-04  worst(RS-GD)=-9.576e-03
  d=32  worst(SS-RS)=-6.831e-04  worst(RS-GD)=-1.351e-02

--- (D)  -BOUNDARY DETECTION  (>1 violates hypothesis) ---
  =0.10 (in hypothesis)       worst(SS-RS)=-4.131e-09  worst(RS-GD)=-3.856e
     -05
  =0.30 (in hypothesis)       worst(SS-RS)=-1.661e-07  worst(RS-GD)=-6.152e
     -04
  =0.50 (in hypothesis)       worst(SS-RS)=-6.394e-07  worst(RS-GD)=-2.407e
     -03
  =0.80 (in hypothesis)       worst(SS-RS)=-1.003e-05  worst(RS-GD)=-2.096e
     -03
  =1.00 (in hypothesis)       worst(SS-RS)=-1.608e-05  worst(RS-GD)=-1.549e
     -03
  =1.20 (OUTSIDE hypothesis)  worst(SS-RS)=+1.312e-03  worst(RS-GD)=+1.046e
     -03
  =1.50 (OUTSIDE hypothesis)  worst(SS-RS)=+2.220e-02  worst(RS-GD)=+6.123e
     -02
  =1.80 (OUTSIDE hypothesis)  worst(SS-RS)=+7
... [truncated]
```

5 Experiment Code (Basic)

```
import itertools
import math
import random
import time
import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
from scipy.stats import binomtest

rng = np.random.default_rng(20260425)

print("=== EXPERIMENT PLAN ===")
```

```
print("""
Conjecture: With (1 - eta_{n,K}) I  <=  A_i  <=  I for symmetric A_i, define
  W_SS := (1/n!) sum_sigma  (A_{sigma(n)} ... A_{sigma(1)})^K
  W_RS := ((1/n!) sum_sigma   A_{sigma(n)} ... A_{sigma(1)})^K
  W_GD := ((1/n) sum_i A_i)^{nK}
The conjecture predicts ||W_SS||_2 <= ||W_RS||_2 <= ||W_GD||_2 for some
eta_{n,K} in (0,1], uniformly in d and in the matrices.

Tests we run:
1) Direct random sampling: many (n,K,d,eta) configurations with random
   symmetric matrices having eigenvalues in [1-eta, 1]. Count violations
   of (a) ||W_SS|| <= ||W_RS||  and  (b) ||W_RS|| <= ||W_GD||.
2) Eta-sweep: for fixed (n,K) shrink eta toward 0 to see whether the
   violation rate vanishes (predicted by the conjecture).
3) Counterexample search: adversarial sampling concentrating eigenvalues
   near the boundary 1-eta and using many random orthogonal frames.
4) Edge cases: commuting A_i (must give equality), A_i = I (trivial),
   K=1 large n (here W_SS = W_RS exactly), d=1 (scalars).
5) Scaling with K and n: track maximum observed violation as parameters
   change.
6) Statistical test: binomial test on whether observed violation rate
   for very small eta is consistent with zero.
""")

t0 = time.time()

def random_sym_in_interval(d, lo, hi, rng):
    """Random symmetric matrix with eigenvalues uniform in [lo, hi]."""
    # Random orthogonal matrix via QR
    G = rng.standard_normal((d, d))
    Q, R = np.linalg.qr(G)
    # Fix sign for uniform Haar measure
    Q = Q * np.sign(np.diag(R))
    evals = rng.uniform(lo, hi, size=d)
    return (Q * evals) @ Q.T

def adversarial_sym(d, lo, hi, rng, p_boundary=0.5):
    """Adversarial: many eigenvalues at the lower boundary (worst case)."""
    G = rng.standard_normal((d, d))
    Q, R = np.linalg.qr(G)
    Q = Q * np.sign(np.diag(R))
    evals = rng.uniform(lo, hi, size=d)
    mask = rng.random(d) < p_boundary
    evals[mask] = lo
    # randomly set a few to hi
    mask2 = rng.random(d) < 0.2
    evals[mask2] = hi
    return (Q * evals) @ Q.T

def compute_quantities(A_list, K):
    """Return (||W_SS||, ||W_RS||, ||W_GD||) and the matrices."""
    n = len(A_list)
    d = A_list[0].shape[0]
    sum_P = np.zeros((d, d))
```

```
    sum_PK = np.zeros((d, d))
    perms = list(itertools.permutations(range(n)))
    for sigma in perms:
        P = np.eye(d)
        for i in sigma:                # apply A_{sigma(1)} first ... A_{
            sigma(n)} last
            P = A_list[i] @ P
        sum_P += P
        # P^K
        PK = np.linalg.matrix_power(P, K)
        sum_PK += PK
    W_SS = sum_PK / len(perms)
    W_RS = np.linalg.matrix_power(sum_P / len(perms), K)
    W_GD = np.linalg.matrix_power(sum(A_list) / n, n * K)
    s_ss = np.linalg.norm(W_SS, 2)
    s_rs = np.linalg.norm(W_RS, 2)
    s_gd = np.linalg.norm(W_GD, 2)
    return s_ss, s_rs, s_gd

# ------------------------------------------------------------------
# Test 1: Random sampling sweep
# ------------------------------------------------------------------
print("\n---␣Test␣1:␣Random␣sampling␣sweep␣---")
configs = []
for n in [2, 3, 4]:
    for K in [1, 2, 3, 5]:
        for d in [1, 2, 3, 5]:
            for eta in [0.05, 0.1, 0.2, 0.4]:
                configs.append((n, K, d, eta))

records = []  # (n,K,d,eta,s_ss,s_rs,s_gd)
N_TRIALS_PER = 30
for (n, K, d, eta) in configs:
    for t in range(N_TRIALS_PER):
        A_list = [random_sym_in_interval(d, 1 - eta, 1.0, rng) for _ in
            range(n)]
        s_ss, s_rs, s_gd = compute_quantities(A_list, K)
        records.append((n, K, d, eta, s_ss, s_rs, s_gd))

rec = np.array(records)
viol_ss_le_rs = rec[:, 4] > rec[:, 5] + 1e-10
viol_rs_le_gd = rec[:, 5] > rec[:, 6] + 1e-10
print(f"Total␣trials:␣{len(rec)}")
print(f"Violations␣||W_SS||␣>␣||W_RS||␣:␣{int(viol_ss_le_rs.sum())}␣({
    viol_ss_le_rs.mean()*100:.3f}%)")
print(f"Violations␣||W_RS||␣>␣||W_GD||␣:␣{int(viol_rs_le_gd.sum())}␣({
    viol_rs_le_gd.mean()*100:.3f}%)")
if viol_ss_le_rs.any():
    worst = np.argmax(rec[:, 4] - rec[:, 5])
    print(f"␣␣Worst␣SS>RS␣gap:␣n={int(rec[worst,0])}␣K={int(rec[worst,1])}␣d
        ={int(rec[worst,2])}␣eta={rec[worst,3]:.3f}␣␣SS-RS={rec[worst,4]-rec[
        worst,5]:.3e}")
if viol_rs_le_gd.any():
    worst = np.argmax(rec[:, 5] - rec[:, 6])
```

```
    print(f"␣␣Worst␣RS>GD␣gap:␣n={int(rec[worst,0])}␣K={int(rec[worst,1])}␣d
        ={int(rec[worst,2])}␣eta={rec[worst,3]:.3f}␣␣RS-GD={rec[worst,5]-rec[
        worst,6]:.3e}")

# ----------------------------------------------------------------------
# Test 2: Eta-sweep - violation rate as eta shrinks
# ----------------------------------------------------------------------
print("\n---␣Test␣2:␣Eta-sweep␣---")
eta_grid = np.array([0.5, 0.3, 0.2, 0.15, 0.1, 0.07, 0.05, 0.03, 0.02,
    0.01])
sweep_n, sweep_K, sweep_d = 3, 2, 4
N_PER_ETA = 400
viol_rate_ss = []
viol_rate_rs =
# ... [truncated]
```

6 Experiment Code (Advanced)

```
import numpy as np
import scipy.linalg as la
import scipy.optimize as opt
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
import itertools
import time
import sympy as sp
from math import factorial

t_start = time.time()
np.random.seed(2026)

# ----------------------------------------------------------------------
# 1.  PLAN
# ----------------------------------------------------------------------
print("===␣ADVANCED␣EXPERIMENT␣PLAN␣===")
print("""
Conjecture (COLT-2021 open problem). For symmetric A_i with
    (1 - eta_{n,K}) I   A_i   I,
the spectral norms satisfy
    ||W_SS||_2     ||W_RS||_2     ||W_GD||_2,
where
    W_SS = (1/n!) Σ_ ( P_)^K              (single-shuffle)
    W_RS = ( (1/n!) Σ_  P_ )^K            (random-reshuffling)
    W_GD = ( (1/n) Σ_i A_i )^{nK}          (full-batch GD).

This advanced experiment goes BEYOND random sampling by combining:

 (A) GRADIENT-BASED ADVERSARIAL SEARCH (L-BFGS-B, multistart) over the
     nonconvex constraint manifold A_i   [(1-)I, I] using a smooth
     eigenvalue-sigmoid parameterization. Random Monte Carlo almost
```

```
    never hits worst cases in non-convex matrix optimization; this
    actively HUNTS for counterexamples.
 (B) SYMBOLIC EXACT verification via sympy of the K=1 identity
    (W_SS = W_RS) and the →0 Taylor structure of W_RS - W_SS
    and W_GD - W_RS for K=2 (showing the leading order is PSD).
 (C) DIMENSIONAL CONVERGENCE STUDY in d{2,4,8,16,32}: if the
    conjecture is true the worst observed violation should stay at
    floating-point noise, independent of d.
 (D) -BOUNDARY DETECTION: probe >1 (allowing INDEFINITE A_i, i.e.
    OUTSIDE the conjecture's hypothesis) to confirm the bound is
    tight - violations should appear there but not for 1.
 (E) STRUCTURED ENSEMBLES (GOE-projected, near-commuting, low-rank
    perturbation, near-degenerate) to test robustness across
    statistical models.
 (F) OPERATOR INVARIANT MONITORING (Loewner-cone consistency of
    traces and Frobenius norms; "energy "conservation surrogate).
 (G) MULTI-PRECISION sanity check on the worst case found by (A)
    using sympy's exact rational arithmetic to rule out f.p. noise.

EXPECTED if TRUE  : adversarial best gap  ~1e-12; symbolic identities
                    confirm; gap independent of d; violations only when
                     > 1 (hypothesis broken).
EXPECTED if FALSE : (A) finds a positive gap in [(1-)I, I]; (G)
                    promotes it to certified counterexample.
""")

# ---------------------------------------------------------------------
# 2.  CORE NUMERICAL ROUTINES
# ---------------------------------------------------------------------
def specnorm(M):
    # spectral norm = largest singular value (M is generally non-symmetric)
    return np.linalg.norm(M, ord=2)

def all_perms(n):
    return list(itertools.permutations(range(n)))

_PERMS_CACHE = {}
def perms_cached(n):
    if n not in _PERMS_CACHE:
        _PERMS_CACHE[n] = all_perms(n)
    return _PERMS_CACHE[n]

def compute_W(As, K):
    """Return (W_SS, W_RS, W_GD) for symmetric matrices As=[A_1,...,A_n]."""
    n = len(As)
    d = As[0].shape[0]
    perms = perms_cached(n)
    np_ = len(perms)
    P = np.empty((np_, d, d))
    I = np.eye(d)
    for k, sigma in enumerate(perms):
        M = I.copy()
        for i in sigma:                          # P_ = A_ {(n)…}A_ {(1)}
            M = As[i] @ M
```

```
        P[k] = M
    if K == 1:
        SS = P.mean(axis=0)
    else:
        PK = P.copy()
        for _ in range(K - 1):
            PK = np.einsum('pij,pjk->pik', PK, P)
        SS = PK.mean(axis=0)
    Pavg = P.mean(axis=0)
    RS   = np.linalg.matrix_power(Pavg, K)
    Abar = sum(As) / n
    GD   = np.linalg.matrix_power(Abar, n * K)
    return SS, RS, GD

def gaps(As, K):
    SS, RS, GD = compute_W(As, K)
    return specnorm(SS) - specnorm(RS), specnorm(RS) - specnorm(GD), SS, RS,
        GD

# Random sampler with eigenvalues in [1-eta, 1]
def random_As(n, d, eta, rng):
    As = []
    for _ in range(n):
        G = rng.standard_normal((d, d))
        Q, _ = np.linalg.qr(G)
        lam = (1.0 - eta) + eta * rng.uniform(0.0, 1.0, d)
        As.append((Q * lam) @ Q.T)
    return As

# ---------------------------------------------------------------------
# 3.  PARAMETERIZATION FOR ADVERSARIAL SEARCH
#     A_i = U_i diag( (1-) +    (_i) ) U_i^T   with U_i = eigvec(M_i)
# ---------------------------------------------------------------------
def theta_to_As(theta, n, d, eta):
    Mlen = d * (d + 1) // 2
    iu = np.triu_indices(d)
    As = []
    for i in range(n):
        ti = theta[i*Mlen:(i+1)*Mlen]
        M = np.zeros((d, d))
        M[iu] = ti
        M = M + M.T - np.diag(np.diag(M))
        w, V = np.linalg.eigh(M)
        s = 1.0 / (1.0 + np.exp(-w))           # sigmoid → (0,1)
        a = (1.0 - eta) + eta * s              # eigenvalues   [1-, 1]
        As.append((V * a) @ V.T)
    return As

def neg_gap(theta, n, d, K, eta, which):
    As = theta_to_As(
# ... [truncated]
```

7 Conclusion

The conjecture remains formally open. Numerical experiments found evidence **against** the conjecture — potential counterexamples were identified. Further investigation (both formal and empirical) is warranted.