Empirical Investigation of an Open Conjecture: KLS Conjecture (Kannan–Lovász–Simonovits)

Agentic NL→Lean 4 Pipeline
Job #45

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 Supported**. The conjecture remains formally open.

1 Conjecture Statement

Conjecture 1.

```
KLS Conjecture (--KannanLovászSimonovits)

For each integer


1
n1, let

 be a Borel probability measure on


R
n
 that is absolutely continuous with respect to Lebesgue measure, with
    density

: →




[
0
,∞

)
f:R
n→∞
[0,) of the form

```

(

)
= -

(

)
f(x)=e-
V(x)
, where

: →

(-∞

,∞

]
V:R
n→-∞∞
(,] is convex; equivalently,

is log-concave. Assume

is isotropic, meaning

(

)
=
0
and

(

)
=

,

R
n

xd(x)=0 and
R
n

xx

d(x)=I
n

,

where

I
n

is the

×

n×n identity matrix.

For a measurable set

AR
n
, define its

-enlargement by

=
{

```
:
dist

(

,

)


}
A



={ xR
n
:dist(x,A) } and its Minkowski boundary measure (with respect to

) by


+
(

)
=
liminf ↓



0

(


)−


(

)

.

+
(A)=↓
0
liminf
```

```
(A



)- (A)


.

Define the Cheeger (isoperimetric) constant of

 as



=
inf


 measurable
,

0
<

(

)
<
1

+
(

)
min

{

(

)
,
1-


(

)
}
.
```

```
=
A measurable, 0 <(A)<1
inf


min {(A)- ,1(A)}

+
(A)



.
Unsolved Problem

(--KannanLovászSimonovits conjecture; Kannan et al. (1995)) determine whether
     there exists a universal constant

>
0
c>0 (independent of

n and of

) such that for every dimension

n and every isotropic log-concave probability measure

 on


R
n
,





.





c.

Equivalently, if


C
n


 is the smallest number such that every isotropic log-concave
```

```
 on


R
n
 satisfies


+
(

)

1


min

{

(

)
,
1-


(

)
}
for all measurable




,

+
(A)
C
n


1


min {(A)- ,1(A)} for all measurable AR
n
,

the conjecture is that
sup
```

```
1


<∞

sup
n1


C
n∞


< (that is,


=

(
1
)
C
n


=O(1) as →∞→∞



n). Eldan (2013) introduced the stochastic localization approach. The best
    known general bound is currently


=

(
log


)
C
n


=O(
logn


), due to Klartag (2023).

Solution Claims
```

```
Accepted clai
... [truncated]
```

2 Status

Formal Status: OPEN — no Lean 4 proof or disproof was found.

Empirical Verdict: Empirically Supported

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 ===

The KLS conjecture (Kannan-Lovasz-Simonovits, 1995):
For every isotropic log-concave probability measure mu on R^n, the Cheeger
(isoperimetric) constant
        psi_mu  =  inf_A  mu^+(A) / min(mu(A), 1-mu(A))
is bounded below by a UNIVERSAL positive constant c > 0 (independent of n).

Equivalently, C_n := sup_mu 1/psi_mu satisfies sup_n C_n < infinity.
Best general result: C_n = O(sqrt(log n)) (Klartag 2023). KLS asks for O(1).

Direct exact computation of psi_mu is intractable. We instead:

  (1) Sample N points from various isotropic log-concave families:
        Standard Gaussian, isotropic cube, centered exponentials,
        Laplace, uniform on isotropic L1-ball.
      All are log-concave by classical results.

  (2) For each measure mu, compute many UPPER BOUNDS on psi_mu by trying
      a large family of test sets A and estimating
            R(A) = mu^+(A) / min(mu(A), 1-mu(A))
      Each R(A) >= psi_mu, so min_A R(A) is a Monte-Carlo upper bound on
         psi_mu.
      Test sets:
        (a) Random half-spaces  A = {<u,x> <= t}
            mu^+(A) = density of the marginal <u,x> at t (vectorised KDE).
        (b) Symmetric slabs    A = {|<u,x>| <= t}
            mu^+(A) = sum of marginal densities at +/- t.
        (c) Centred balls      A = {||x|| <= r}
            mu^+(A) = density of ||x|| at r  (relevant to the thin-shell
            conjecture, which is implied by KLS up to log factors).

  (3) Sweep dimensions n in {2,4,8,16,32,64,128} and inspect how the
      empirical infimum psi_hat(n) behaves.  KLS predicts psi_hat is
      bounded below by a positive constant.  A counterexample-style scaling
```

```
      psi_hat(n) ~ n^{-alpha} with alpha > 0 would be evidence against KLS.

  (4) Fit log psi_hat(n) = a + alpha * log n and report alpha.
      alpha approx 0 supports KLS; alpha < 0 with significant slope refutes
         it.

  (5) Auxiliary thin-shell test:  Var(||X||/sqrt(n)) -> 0?  KLS implies the
      thin-shell conjecture.  Persistent decay supports the conjecture.

Running sweeps...

--- Gaussian ---
  n=  2: psi_hat (half)=0.7910  (slab)=1.2430  (ball)=1.2016  min=0.7910
     Var(|X|/sqrt n)=0.2159
  n=  4: psi_hat (half)=0.7893  (slab)=1.2382  (ball)=1.1616  min=0.7893
     Var(|X|/sqrt n)=0.1157
  n=  8: psi_hat (half)=0.7898  (slab)=1.2453  (ball)=1.1381  min=0.7898
     Var(|X|/sqrt n)=0.0606
  n= 16: psi_hat (half)=0.7883  (slab)=1.2443  (ball)=1.1447  min=0.7883
     Var(|X|/sqrt n)=0.0309
  n= 32: psi_hat (half)=0.7914  (slab)=1.2385  (ball)=1.1448  min=0.7914
     Var(|X|/sqrt n)=0.0154
  n= 64: psi_hat (half)=0.7916  (slab)=1.2420  (ball)=1.1507  min=0.7916
     Var(|X|/sqrt n)=0.0077
  n=128: psi_hat (half)=0.7919  (slab)=1.2377  (ball)=1.1454  min=0.7919
     Var(|X|/sqrt n)=0.0038

--- Uniform cube ---
  n=  2: psi_hat (half)=0.5793  (slab)=1.1494  (ball)=1.4974  min=0.5793
     Var(|X|/sqrt n)=0.1210
  n=  4: psi_hat (half)=0.5902  (slab)=1.1643  (ball)=1.7139  min=0.5902
     Var(|X|/sqrt n)=0.0562
  n=  8: psi_hat (half)=0.6489  (slab)=1.2026  (ball)=1.7823  min=0.6489
     Var(|X|/sqrt n)=0.0265
  n= 16: psi_hat (half)=0.7700  (slab)=1.2364  (ball)=1.8439  min=0.7700
     Var(|X|/sqrt n)=0.0129
  n= 32: psi_hat (half)=0.7809  (slab)=1.2306  (ball)=1.8107  min=0.7809
     Var(|X|/sqrt n)=0.0063
  n= 64: psi_hat (half)=0.7850  (slab)=1.2214  (ball)=1.8497  min=0.7850
     Var(|X|/sqrt n)=0.0031
  n=128: psi_hat (half)=0.7878  (slab)=1.2492  (ball)=1.8335  min=0.7878
     Var(|X|/sqrt n)=0.0016

--- Exp (centered) ---
  n=  2: psi_hat (half)=0.8962  (slab)=0.9843  (ball)=0.9702  min=0.8962
     Var(|X|/sqrt n)=0.3304
  n=  4: psi_hat (half)=0.8577  (slab)=1.0165  (ball)=0.9447  min=0.8577
     Var(|X|/sqrt n)=0.2288
  n=  8: psi_hat (half)=0.8444  (slab)=1.2256  (ball)=0.8710  min=0.8444
     Var(|X|/sqrt n)=0.1476
  n= 16: psi_hat (half)=0.8421  (slab)=1.3012  (ball)=0.7241  min=0.7241
     Var(|X|/sqrt n)=0.0900
  n= 32: psi_hat (half)=0.8280  (slab)=1.2788  (ball)=0.6471  min=0.6471
     Var(|X|/sqrt n)=0.0516
```

```
  n= 64: psi_hat (half)=0.8071  (slab)=1.2645
... [truncated]
```

4 Experiment Code (Basic)

```
import matplotlib
matplotlib.use("Agg")
import numpy as np
import matplotlib.pyplot as plt
import math
import time

t_start = time.time()

print("===␣EXPERIMENT␣PLAN␣===")
print("""
The KLS conjecture (Kannan-Lovasz-Simonovits, 1995):
For every isotropic log-concave probability measure mu on R^n, the Cheeger
(isoperimetric) constant
        psi_mu  =  inf_A  mu^+(A) / min(mu(A), 1-mu(A))
is bounded below by a UNIVERSAL positive constant c > 0 (independent of n).

Equivalently, C_n := sup_mu 1/psi_mu satisfies sup_n C_n < infinity.
Best general result: C_n = O(sqrt(log n)) (Klartag 2023). KLS asks for O(1).

Direct exact computation of psi_mu is intractable. We instead:

  (1) Sample N points from various isotropic log-concave families:
        Standard Gaussian, isotropic cube, centered exponentials,
        Laplace, uniform on isotropic L1-ball.
      All are log-concave by classical results.

  (2) For each measure mu, compute many UPPER BOUNDS on psi_mu by trying
      a large family of test sets A and estimating
          R(A) = mu^+(A) / min(mu(A), 1-mu(A))
      Each R(A) >= psi_mu, so min_A R(A) is a Monte-Carlo upper bound on
         psi_mu.
      Test sets:
        (a) Random half-spaces  A = {<u,x> <= t}
            mu^+(A) = density of the marginal <u,x> at t (vectorised KDE).
        (b) Symmetric slabs    A = {|<u,x>| <= t}
            mu^+(A) = sum of marginal densities at +/- t.
        (c) Centred balls      A = {||x|| <= r}
            mu^+(A) = density of ||x|| at r  (relevant to the thin-shell
            conjecture, which is implied by KLS up to log factors).

  (3) Sweep dimensions n in {2,4,8,16,32,64,128} and inspect how the
      empirical infimum psi_hat(n) behaves.  KLS predicts psi_hat is
      bounded below by a positive constant.  A counterexample-style scaling
      psi_hat(n) ~ n^{-alpha} with alpha > 0 would be evidence against KLS.

```

```
  (4) Fit log psi_hat(n) = a + alpha * log n and report alpha.
      alpha approx 0 supports KLS; alpha < 0 with significant slope refutes
          it.

  (5) Auxiliary thin-shell test:  Var(||X||/sqrt(n)) -> 0?  KLS implies the
      thin-shell conjecture.  Persistent decay supports the conjecture.
""")

rng = np.random.default_rng(20260425)

N = 25000
DIMS = [2, 4, 8, 16, 32, 64, 128]
N_DIRS = 80
N_THRESH = 30
INV_SQRT_2PI = 1.0 / math.sqrt(2 * math.pi)

# ---------------- Samplers (each returns roughly isotropic, log-concave)
    ----------------
def sample_gaussian(n, M):
    return rng.standard_normal((M, n))

def sample_cube(n, M):
    return rng.uniform(-math.sqrt(3.0), math.sqrt(3.0), size=(M, n))

def sample_exp(n, M):
    return rng.exponential(1.0, size=(M, n)) - 1.0  # mean 0, var 1

def sample_laplace(n, M):
    return rng.laplace(0.0, 1.0 / math.sqrt(2.0), size=(M, n))  # var 1

def sample_l1_ball(n, M):
    # Uniform on unit l1 ball, then rescale to isotropic.
    e = rng.exponential(1.0, size=(M, n))
    sgn = rng.choice([-1.0, 1.0], size=(M, n))
    s = e.sum(axis=1, keepdims=True)
    u = sgn * e / s                                    # uniform on l1 unit
        sphere
    r = rng.uniform(0.0, 1.0, size=(M, 1)) ** (1.0 / n)  # radius
    pts = u * r
    var = 2.0 / ((n + 1) * (n + 2))                    # known coord
        variance
    return pts / math.sqrt(var)

distributions = {
    "Gaussian":         sample_gaussian,
    "Uniform␣cube":     sample_cube,
    "Exp␣(centered)":   sample_exp,
    "Laplace":          sample_laplace,
    "Uniform␣L1␣ball":  sample_l1_ball,
}

# ---------------- KDE helpers ----------------
def kde_density(samples, points, h):
```

```
    """Gaussian KDE density of `samples` evaluated at `points`. Vectorised.
        """
    z = (samples[:, None] - points[None, :]) / h
    return (np.exp(-0.5 * z * z).mean(axis=0)) * (INV_SQRT_2PI / h)

def halfspace_ratios(X):
    n = X.shape[1]; M = X.shape[0]
    U = rng.standard_normal((N_DIRS, n))
    U /= np.linalg.norm(U, axis=1, keepdims=True)
    P = U @ X.T  # (N_DIRS, M)
    ratios = []
    qs = np.linspace(0.05, 0.95, N_THRESH)
    for i in range(N_DIRS):
        proj = P[i]
        sd = proj.std()
        if sd < 1e-9:
            continue
        h = 1.06 * sd * M ** (-1.0/5.0)
        ts = np.quantile(proj, qs)
        mu_A = (proj[:, None] <= ts[None, :]).mean(axis=0)
        f_t = kde_density(proj, ts, h)
        denom = np.minimum(mu_A, 1.0 - mu_A)
        mask = (denom > 0.01)
        if mask.any():
            ratios.append(f_t[mask] / denom[mask])
    return np.concatenate(ratios) if ratios else np.array([])

def slab_ratios(X):
    n = X.shape[1]; M = X.shape[0]
    K = min(50, N_DIRS)
    U = rng.standard_normal((K, n))
    U /= np.linalg.norm(U, axis=1, keepdims=True)
    P = U @ X.T
    ratios = []
    qs = np.linspace(0.1, 0.95, 20)
    for i in range(K):
        proj = P[i]; absproj = np.abs(proj)
        sd = proj.std()
        if sd < 1e-9:
            continue
        h = 1.06 * sd * M ** (-1.0/5.0)
        ts = np.quantile(absproj, qs)
        mu_A = (absproj[:, None] <= ts[None, :
# ... [truncated]
```

5 Experiment Code (Advanced)

```
import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
```

```
from scipy import sparse
from scipy.sparse.linalg import eigsh
import time, math, itertools

# Advanced KLS experiment: spectral gaps via weighted Neumann-Sturm-
    Liouville PDE,
# adversarial direction search, Eldan thin-shell estimates, and convergence/
    sensitivity tests.

t_start = time.time()

print("===␣ADVANCED␣EXPERIMENT␣PLAN␣===")
print("""
We probe the KLS conjecture beyond simple half-space Monte Carlo by
    combining
five research-grade ingredients:

(A) PDE-BASED SPECTRAL GAP.  For each isotropic log-concave family mu on R^n
    and many directions theta in S^{n-1}, we form the 1D marginal density
    rho_theta(t) via Gaussian KDE, then SOLVE the weighted Neumann
    Sturm-Liouville eigenvalue problem
        -(rho u')' = lambda rho u   on [a,b],  u'(a)=u'(b)=0
    via second-order finite differences and a sparse generalized eigensolver
    (scipy.sparse.linalg.eigsh, shift-invert).  The 2nd eigenvalue lambda_1
    is the Poincare constant of the marginal.  By Cheeger lambda_1 >= psi
        ^2/4
    and by Ledoux/Buser for log-concave  psi >= c sqrt(lambda_1).  So
    sqrt(lambda_1) tracks the Cheeger constant of the marginal up to
    universal constants.  KLS in R^n implies  inf_theta sqrt(lambda_1(theta)
        )
    >= c uniformly in n.

(B) ADVERSARIAL DIRECTION SEARCH.  Instead of averaging, we MINIMIZE over a
    pool of random Haar directions, coordinate axes, sparse directions and
    eigendirections of small empirical-cov perturbations -- attacking the
    conjecture from its worst 1D slice.

(C) THIN-SHELL CONSTANT (Eldan/Lee-Vempala).  The KLS constant satisfies
    psi >= c / sigma_n  where  sigma_n^2 := Var(|X|)/n  is the thin-shell
    width.  We track sigma_n; KLS true => sigma_n bounded.

(D) MULTI-RESOLUTION CONVERGENCE.  The eigenvalue solver is run at grids
    m in {100, 200, 400, 800, 1600} and we fit the convergence order in 1/m^
        p.

(E) MOMENT/ISOTROPY MONITORING (the analogue of energy conservation).  We
    track ||hat_Sigma - I||_F (should be O(n/sqrt N)) and the empirical
    mean (should be O(1/sqrt N)) -- numerical drift in these would
    contaminate any conclusion about psi.

If KLS is TRUE: as n grows, inf_theta sqrt(lambda_1) stays bounded below by
a positive universal constant, AND sigma_n remains O(1).
If KLS is FALSE: either sqrt(lambda_1(theta_n*)) -> 0 along some direction
sequence, or sigma_n diverges.
```

```
""")

rng = np.random.default_rng(20260425)

#
    ---------------------------------------------------------------------------

# Sampling: 6 isotropic log-concave families
#
    ---------------------------------------------------------------------------

def isotropize(X):
    X = X - X.mean(axis=0)
    if X.shape[1] == 1:
        v = X.var()
        return X / np.sqrt(max(v, 1e-12))
    C = np.cov(X.T)
    U, s, _ = np.linalg.svd(C)
    s = np.maximum(s, 1e-12)
    W = U @ np.diag(1.0 / np.sqrt(s)) @ U.T
    return X @ W

def sample(name, n, N, rng):
    if name == "Gaussian":
        return rng.standard_normal((N, n))
    if name == "UniformCube":
        return rng.uniform(-1, 1, (N, n)) * np.sqrt(3.0)
    if name == "Exponential":
        return rng.exponential(1.0, (N, n)) - 1.0
    if name == "Laplace":
        return rng.laplace(0.0, 1.0 / np.sqrt(2.0), (N, n))
    if name == "Simplex":
        E = rng.exponential(1.0, (N, n + 1))
        S = E / E.sum(axis=1, keepdims=True)
        return isotropize(S[:, :n])
    if name == "CrossPolytope":
        E = rng.exponential(1.0, (N, n))
        sgn = rng.choice([-1.0, 1.0], (N, n))
        L = E * sgn
        L = L / np.sum(np.abs(L), axis=1, keepdims=True)
        r = rng.uniform(0, 1, (N, 1)) ** (1.0 / n)
        return isotropize(r * L)
    raise ValueError(name)

DISTS = ["Gaussian", "UniformCube", "Exponential", "Laplace", "Simplex", "
   CrossPolytope"]

#
    ---------------------------------------------------------------------------

# 1D weighted Neumann Sturm-Liouville solver
#
    ---------------------------------------------------------------------------

```

```
def kde_on_grid(x1d, grid, h):
    N = len(x1d)
    rho = np.zeros_like(grid)
    bs = 4096
    norm = 1.0 / (N * h * np.sqrt(2.0 * np.pi))
    for i in range(0, N, bs):
        c = x1d[i : i + bs]
        z = (grid[:, None] - c[None, :]) / h
        rho += np.exp(-0.5 * z * z).sum(axis=1)
    return rho * norm

def spectral_gap_1d(rho, dx):
    m = len(rho)
    rho_e = 0.5 * (rho[:-1] + rho[1:])
    inv_dx2 = 1.0 / (dx * dx)
    main = np.empty(m)
    main[1:-1] = (rho_e[:-1] + rho_e[1:]) * inv_dx2
    main[0] = rho_e[0] * inv_dx2
    main[-1] = rho_e[-1] * inv_dx2
    off = -rho_e * inv_dx2
    A = sparse.diags([off, main, off], offsets=[-1, 0, 1], format="csr")
    M = sparse.diags(np.maximum(rho, 1e-14), format="csr")
    try:
        vals = eigsh(A, k=2, M=M, sigma=1e-10, which="LM",
                     return_eigenvectors=False, tol=1e-7, maxiter=2000)
    except Exception:
        try:

# ... [truncated]
```

6 Conclusion

The conjecture remains formally open. Numerical experiments **support** the conjecture — no counterexamples were found across all tested parameter ranges. Further investigation (both formal and empirical) is warranted.