An autonomous search for extremal objects

Certified lower bounds for sum-difference and Fourier entropy–influence constants

`op_idempotent` — generated by the accompanying software

August 23, 2026

Abstract

We describe a system that searches for the finite objects witnessing lower bounds on open constants in additive combinatorics and Boolean function analysis, chooses for itself which constant to work on, and certifies every number it reports in exact arithmetic. Two families are implemented: the Katz–Tao sum-difference exponents $\mathrm{SD}(S;s)$, whose witnesses are finitely supported measures on $\mathbb{Z}^2$, and the Fourier entropy–influence constant C_{71}, whose witnesses are balanced Boolean functions. The two are unrelated as mathematics but identical in shape — a finite explicit object, expensive to find and cheap to check — and are served by one verifier that uses exact rationals and interval-enclosed logarithms and reports the *lower* endpoint of its enclosure, so that a reported bound is valid even if the search that produced it is wrong. A second, independent implementation of the verifier in a different language cross-checks every certificate. Target selection is a bandit over (problem, strategy) arms with a headroom-normalised reward. Nobody supplies the target: over 15 autonomous episodes the system chose its own problems, took the first certified value on 3 previously untouched instances, and exposed an error in the equivalence test it had been given. Because acceptance is conditioned on a *certified* improvement, the reported state is monotone — it can be left running indefinitely and cannot regress. We report 566 logged attempts across 11 problems, identify the further constants to which this machinery transfers essentially unchanged, and state plainly where the current results fall short of the published records.

1 Introduction

A curated repository of open optimisation constants in mathematics [1] records, for roughly one hundred constants, the best known upper and lower bounds together with their provenance. For many of these the lower bound is *constructive*: it is witnessed by a finite object, and verifying the witness is far cheaper than finding it. That asymmetry is exactly the shape that admits automation, and it is the shape this system exploits.

Three design commitments follow from it.

1. **Search is untrusted.** The optimiser is floating point, nonconvex and may be wrong in any way. Nothing it outputs is reported.

2. **Certification is exact.** A proposal becomes a result only after it is rounded to exact rationals and re-derived from scratch in interval arithmetic, with the lower endpoint of the enclosure reported. The worst a broken search can do is waste time.

3. **Everything is logged.** Failed attempts stay in the record with the hypothesis they were testing. A search log that shows only successes conveys no information about the landscape.

4. **The target is chosen by the system.** With eleven problems and several strategies each, allocation is itself an inference problem and is handled as one (Section 5) rather than by a standing human guess.

Together these give a system that can be left alone. Section 5.1 makes precise the sense in which running it longer cannot make the reported state worse, which is what turns "unattended" from a convenience into a guarantee.

Section 2 sets out the two problem families and what they have in common. Section 3 describes the verifier. Section 4 describes the searches. Section 5 describes the autonomous target selection. Section 6 reports what was achieved, including where it falls short.

2 Two families

2.1 Sum-difference exponents

For finite $A, B \subset \mathbb{R}$ and a graph $G \subset A \times B$ write $A \overset{G}{+} \sigma B := \{a + \sigma b : (a, b) \in G\}$. For a finite set S of slopes in $\mathbb{Q} \cup \{\infty\}$ and a target slope $s \notin S$, $\mathrm{SD}(S; s)$ is the least exponent such that

$$\left|A \overset{G}{+} sB\right| \;\le\; \Big(\max_{\sigma \in S} \left|A \overset{G}{+} \sigma B\right|\Big)^{\mathrm{SD}(S;s)}$$

for all such A, B, G. Two members have an active literature: $C_{3b} = \mathrm{SD}(\{0, 1, \infty\}; -1)$ and $C_{3c} = \mathrm{SD}(\{0, 1, 2, \infty\}; -1)$ [2, 4]. The infimum of $\mathrm{SD}(S; s)$ over the family is the subject of the arithmetic Kakeya conjecture of Katz and Tao [3].

There is an equivalent entropy formulation, and it is the one every known lower bound has used [4]: $\mathrm{SD}(S; s)$ is the least C with

$$\mathbf{H}(X + sY) \;\le\; C \max_{\sigma \in S} \mathbf{H}(X + \sigma Y) \tag{1}$$

for all discrete random variables X, Y. Consequently *any* law p on $\mathbb{Z}^2$ furnishes a lower bound,

$$\mathrm{SD}(S; s) \;\ge\; R_p \;:=\; \frac{\mathbf{H}(X + sY)}{\max_{\sigma \in S} \mathbf{H}(X + \sigma Y)}, \qquad (X, Y) \sim p, \tag{2}$$

provided the denominator is nonzero. A witness is therefore a finite support in $\mathbb{Z}^2$ together with a probability vector on it. We represent the slope $\sigma = p/q$ by the primitive integer form $F_\sigma(x, y) = qx + py$, so that $\sigma = 0$ gives x, $\sigma = \infty$ gives y, and $\sigma = -1$ gives $x - y$.

2.2 The Fourier entropy–influence constant

For $f : \{-1, 1\}^n \to \{-1, 1\}$ with Fourier expansion $f = \sum_S \hat{f}(S)\chi_S$, the spectral entropy and total influence are

$$\mathbf{H}[\hat{f}^2] = \sum_S \hat{f}(S)^2 \log_2 \frac{1}{\hat{f}(S)^2}, \qquad \mathrm{Inf}[f] = \sum_S \hat{f}(S)^2 |S|.$$

Friedgut and Kalai conjectured a universal C with $\mathbf{H}[\hat{f}^2] \le C\,\mathrm{Inf}[f]$ for every Boolean f; whether any finite C exists is open [6]. We write C_{71} for the least such constant.

The composition theorem of O'Donnell and Tan [7] amplifies a single example. For *balanced* f with $\mathrm{Inf}[f] > 1$,

$$C_{71} \;\ge\; \frac{\mathbf{H}[\hat{f}^2]}{\mathrm{Inf}[f] - 1}. \tag{3}$$

Remark 1 (balance is load-bearing)**.** The hypothesis that f be balanced cannot be dropped. Take $f = \mathrm{AND}_2$, whose four Fourier coefficients are $\pm\frac{1}{2}$: then $\mathbf{H}[\hat{f}^2] = 2$ and $\mathrm{Inf}[f] = 1$ *exactly*, so the right-hand side of (3) divides by zero and would appear to refute a conjecture that is wide open. Our verifier therefore checks $\hat{f}(\emptyset) = 0$ exactly and refuses the certificate otherwise.

A witness is a truth table, which we store packed in hexadecimal.

2.3 What the two have in common

Both reduce to: a finite explicit object, a score computable exactly from it, and an asymmetry between the cost of search and the cost of verification. That is why one verifier serves both, and why the same autonomous controller can allocate effort between them.

Both also have a *scale invariance* that bounds what brute enlargement can buy.

Lemma 2 (tensor invariance)**.** *Let p be a law on $\mathbb{Z}^2$ with ratio R_p as in* (2)*. Take k independent pairs (X_i, Y_i), each with law p, and set*

$$X = \sum_{i=1}^{k} N^{\,i-1} X_i, \qquad Y = \sum_{i=1}^{k} N^{\,i-1} Y_i,$$

with N larger than the diameter of every set $F_\sigma(\operatorname{supp} p)$ for $\sigma \in S \cup \{s\}$. Then the ratio of (X, Y) equals R_p.

Proof. Each F_σ is linear, so $F_\sigma(X, Y) = \sum_i N^{\,i-1} F_\sigma(X_i, Y_i)$. For N that large the digits do not interact, so the map from $(F_\sigma(X_1, Y_1), \ldots, F_\sigma(X_k, Y_k))$ to $F_\sigma(X, Y)$ is injective and $\mathbf{H}_\sigma = k\,\mathbf{H}_\sigma(p)$ by independence. This holds for every σ including the target, and the maximum over S is attained at the same slope for every k, so numerator and denominator both scale by k. □

Lemma 3 (composition invariance)**.** *Let f be balanced with $I := \mathrm{Inf}[f] > 1$ and $H := \mathbf{H}[\hat{f}^2]$, and let $f^{(2)}$ denote f composed with independent copies of itself on disjoint blocks of variables. Then $\mathrm{Inf}[f^{(2)}] = I^2$ and $\mathbf{H}[\widehat{f^{(2)}}^{\,2}] = H(1 + I)$, so*

$$\frac{\mathbf{H}[\widehat{f^{(2)}}^{\,2}]}{\mathrm{Inf}[f^{(2)}] - 1} = \frac{H(1+I)}{I^2 - 1} = \frac{H}{I - 1}.$$

Proof. The two identities are the composition theorem of [7] for balanced inner functions. Given them, $I^2 - 1 = (I - 1)(I + 1)$ cancels the factor $1 + I$. □

Both lemmas say the same thing: making a witness bigger by repeating it changes nothing. Gains must come from new structure. Both are checked numerically in the verifier's self-test — for Lemma 3, MAJ_3 and $\mathrm{MAJ}_3 \circ \mathrm{MAJ}_3$ both certify to $4.000\ldots$ on 3 and 9 variables respectively.

2.4 When two instances are the same problem

Lemma 4 (equivalence under injective linear maps)**.** *Let $M \in \mathbb{Z}^{2\times 2}$ with $\det M \neq 0$, and let φ_M be the induced map on slopes, defined by $F_\sigma \circ M = c\, F_{\varphi_M(\sigma)}$ for some $c \neq 0$. If φ_M carries S_B onto S_A and s_B to s_A, then $\mathrm{SD}(S_A; s_A) = \mathrm{SD}(S_B; s_B)$.*

Proof. M is injective on $\mathbb{Z}^2$, so for a law p the pushforward M_*p satisfies $\mathbf{H}(F_\sigma(M(X, Y))) = \mathbf{H}(F_{\varphi_M(\sigma)}(X, Y))$, entropy being blind to the nonzero scalar c and to injective relabelling. Hence the B-ratio at M_*p equals the A-ratio at p, giving $\mathrm{SD}(S_B; s_B) \geq \mathrm{SD}(S_A; s_A)$. The adjugate $\mathrm{adj}(M)$ is again an integer matrix with nonzero determinant and induces φ_M^{-1}, which gives the reverse inequality. □

Remark 5. Requiring $|\det M| = 1$ instead of $\det M \neq 0$ — as we did at first — silently misses genuine restatements. The instance $\mathrm{SD}(\{0, 1, 2, \infty\}; -2)$ is C_{3c} under $(X, Y) \mapsto (-2Y, -X)$, a map of determinant -2. Surjectivity is not needed: entropy only requires that distinct points stay distinct.

3 Certification

3.1 Exact arithmetic

A certificate carries exact rational data: for the sum-difference family a list of integer support points and rational weights summing to 1; for the Boolean family an integer n and a truth table. Push-forwards are summed exactly over rationals. For the Boolean family the Walsh–Hadamard transform of a ± 1 table is pure integer arithmetic, giving $\hat{f}(S) = a_S/2^n$ with $a_S \in \mathbb{Z}$, so $\hat{f}(S)^2 = a_S^2/4^n$ is an exact rational; Parseval, $\sum_S a_S^2 = 4^n$, is checked as an integer identity and rejects a corrupted table outright.

3.2 Rigorous logarithms

Entropies need logarithms, which are irrational. We enclose them. Fix B fractional bits and represent a real by an integer pair (ℓ, h) standing for $[\ell 2^{-B}, h 2^{-B}]$, with every operation rounded outward. For $x > 0$ rational, range-reduce $x = m2^e$ with $m \in [1, 2)$ and use

$$\ln x = e \ln 2 + 2 \operatorname{artanh}(z), \qquad z = \frac{m-1}{m+1} \in [0, \tfrac{1}{3}), \qquad \operatorname{artanh}(z) = \sum_{k \text{ odd}} \frac{z^k}{k}.$$

Proposition 6 (truncation bound)**.** *Truncating the series after z^K with K odd leaves a remainder*

$$0 \;\leq\; \sum_{\substack{j \geq K+2 \\ j \text{ odd}}} \frac{z^j}{j} \;\leq\; \frac{z^{K+2}}{(K+2)\,(1-z^2)} \;\leq\; \frac{9}{8}\,\frac{z^{K+2}}{K+2},$$

the last step using $z \leq 1/3$.

That bound is added to the upper endpoint, so the returned interval provably contains $\ln x$. We take $B = 256$ and $K = 199$, for which the remainder is below 2^{-300}. Since p_S depends only on a_S^2, equal squares are collapsed before taking logarithms; an $n = 18$ table then needs a handful of enclosures rather than a quarter of a million.

Finally the ratio is formed by interval division and the *lower* endpoint reported. This is the step that makes the whole pipeline safe: the number is a valid lower bound on the constant regardless of how the optimiser behaved.

3.3 Two implementations

The verifier is implemented twice — once in Python over exact rationals and fixed-point intervals, once in JavaScript over arbitrary-precision integers — written against this specification rather than translated from one another, and sharing no code. Agreement to the last printed digit on every certificate is the evidence that the numbers are what they claim to be. The second implementation paid for itself immediately: it applied the fixed-point scale twice in interval division and returned a 78-digit value where $1.7259\ldots$ was expected. A single implementation would have had no way to notice.

3.4 Anchors

The self-test pins the pipeline to values from the literature rather than to itself. The uniform measure on $\{(0,0),(1,0),(0,1)\}$ must reproduce Ruzsa's bound for C_{3b},

$$\frac{\log 27}{\log(27/4)} = 1.725982457878719108719\ldots,$$

to the last printed digit. Beyond that: MAJ_3 must give exactly 4 for C_{71}; parity must give 0; AND_2 must be refused; and Lemmas 2 and 3 must hold exactly.

4 Search

4.1 Measures on the lattice

For a fixed support, (2) is maximised over the simplex. The denominator is a maximum of concave functions and is not itself concave, so the problem is nonconvex and multimodal. We run many random restarts of Adam in a softmax parametrisation, batched into a single array computation, with the denominator smoothed as $\tau \log \sum_\sigma e^{\mathbf{H}_\sigma/\tau}$ and τ annealed towards zero. Since the smoothed denominator dominates the true maximum, the smoothed objective is a lower estimate throughout.

Two practical findings mattered more than the optimiser itself.

Warm starts must anneal cold. Re-running the full temperature schedule from a good point ejects the iterate from its basin, and it does not return on a smaller budget. Starting warm runs at $\tau_0 = 2 \cdot 10^{-3}$ and a reduced step moved C_{3c} from 1.674667 to 1.674719 in one pass.

Equalisation. At every good optimum we found, all constraint entropies agree to five decimal places. This is what one expects: if some σ had $\mathbf{H}_\sigma$ strictly below the maximum, mass could be moved to raise the numerator without touching the binding constraint. Annealing towards a hard maximum makes the gradient see one slope at a time and chatter between near-ties, so we add a penalty $\lambda \sum_\sigma (\mathbf{H}_\sigma/\overline{\mathbf{H}} - 1)^2$ on their spread. It helps C_{3c}, which has four slopes to tie, and does nothing for C_{3b}, which has three.

Around the inner loop sits a combinatorial one that prunes points driven to zero weight and offers a ring of new lattice neighbours, keeping a round only if the *certified* value improves.

4.2 Backward elimination, and a monotonicity lemma

To probe whether large supports are doing real work, we delete points one at a time, each time removing the one whose loss is smallest and re-optimising the survivors.

Lemma 7 (monotonicity in support size)**.** *Let $m(k)$ be the supremum of (2) over supports of size at most k. Then m is non-decreasing in k.*

Proof. Given a law on k points with ratio r, place mass ε on a new point and renormalise. All entropies are continuous in ε, so the ratio tends to r as $\varepsilon \to 0$; hence $m(k+1) \geq r$ for every $r < m(k)$. □

Lemma 7 makes the elimination curve self-diagnosing: the running maximum from small sizes upward is itself a valid lower-bound curve, and any point below that envelope is provably a failed re-optimisation at that size rather than a property of the problem. Figure 1 shows both.

4.3 Boolean functions

For C_{71} the witness is a truth table and the constraint is balance, so we anneal with *swap* moves — exchanging a $+1$ with a -1 — rather than single flips. Every candidate is then admissible by construction instead of being repaired afterwards. Starts are read-once monotone formulas (tribes and their duals, thresholds, majorities), rebalanced where necessary by flipping the surplus sign at random positions. Rebalancing destroys monotonicity but not admissibility: (3) asks only for balance and Inf > 1. Certificates that happen to remain monotone additionally witness the bound within the monotone class, which is the stronger claim the published records make.

5 Autonomous target selection

With 11 problems and several strategies each there is no reason to think a human guess about where to spend the next hour is any good. We treat the choice as a bandit over $(\text{problem}, \text{strategy})$ arms (32 of them) and let measured return decide.

Let g be the certified gain an episode produced on its problem and let h be the *headroom*: the distance to the published record where one exists, and a small relative scale where none does. The reward is

$$r \;=\; \tanh\!\Big(\frac{g}{0.1\,h}\Big)\,. \tag{4}$$

The normalisation by h is what makes families comparable: closing a tenth of the remaining gap on C_{3c} scores the same as closing a tenth of it on C_{71}, though one is measured in the sixth decimal place and the other in the first. An episode that produces the first certified value for a problem scores 1.

Three modifications to UCB1 earn their place.

- **Optimistic initialisation with a headroom prior**, so an untried arm on a wide-open problem is taken before a well-worn arm on a saturated one.
- **Resting.** An arm returning nothing several times running is set aside for a fixed number of episodes. Plain UCB keeps revisiting a dead arm because its exploration term grows without bound; on a landscape that has genuinely saturated this is waste.
- **A wall-clock budget per episode**, so one slow strategy cannot starve the others.

Every decision is recorded with its reason and with the alternatives it passed over. A controller that cannot explain itself is indistinguishable from a random number generator.

5.1 Why it can be left running

Proposition 8 (monotone reported state)**.** *For a given constant let b_t be the largest certified value obtained after t episodes. Then every b_t is a valid lower bound on the constant, and $t \mapsto b_t$ is non-decreasing.*

Proof. Each certified value is the lower endpoint of a rigorous enclosure of the ratio at an explicit witness (Section 3), hence a valid lower bound; b_t is a maximum over a set that only grows with t. □

This is the property the project is named for: re-running the system cannot change the reported state for the worse, so applying it again is safe in the way an idempotent operation is safe.

The refinement and elimination loops enforce the same discipline internally: a round is kept only when the *certified* value improves, never when the floating-point objective merely looks

better. A bad episode therefore costs time and nothing else — there is no state a failed search can corrupt. That is what makes unattended operation safe rather than merely convenient, and it is why the controller persists its statistics between sessions and can be restarted without losing what it has learned.

What Proposition 8 does *not* say is that progress continues. The measured behaviour is the opposite — arms saturate, and the resting rule exists precisely because they do. The guarantee is directional, not a rate.

6 Results

Over 566 logged attempts spanning 87 distinct hypotheses and 6.3 CPU-hours of search, the certified state is as follows. All values are lower endpoints of rigorous enclosures and were reproduced by both verifier implementations.

	definition	certified here	size	published record	ceiling	gap
C_{3b}	$\mathrm{SD}(\{0,1,\infty\};-1)$	`1.7789861768196179`	26	`1.7789888400000000`	1.833333	2.66e-06
C_{3c}	$\mathrm{SD}(\{0,1,2,\infty\};-1)$	`1.6747273564899071`	24	`1.6747338950208248`	1.750000	6.54e-06
C_{71}	$\mathbf{H}[\hat{f}^2] \,/\, (\mathrm{Inf}[f]-1)$	`6.3470674987019188`	6	`6.5218457109230465`	∞	1.75e-01

Table 1: The three constants with an active literature. "Size" is support points for the sum-difference family and variables for C_{71}. We are below the record in every case.

For C_{3b} the certified 1.7789861… exceeds Lemm's published 1.77898 [5] but not the current record; for C_{3c} the certified 1.6747273… exceeds the AlphaEvolve and Astor entries but not the 95-point record; for C_{71} the certified 6.3470674… exceeds O'Donnell and Tan's 6.278 [7] but not Hod's 6.4547837 [8].

instance	certified here	size	status
$\mathrm{SD}(\{0,1,2,3,\infty\};-1)$	`1.5492930331763236`	99	no published bound
$\mathrm{SD}(\{0,1,3,\infty\};-1)$	`1.5857911333358468`	104	no published bound
$\mathrm{SD}(\{0,1,-1,\infty\};2)$	`1.6747244839397877`	100	$= C_{3c}$ via $(X,Y) \mapsto (-1X-1Y,\ 0X+1Y)$
$\mathrm{SD}(\{0,1,\infty\};2)$	`1.7789824701134081`	41	$= C_{3b}$ via $(X,Y) \mapsto (-1X-1Y,\ 0X+1Y)$
$\mathrm{SD}(\{0,1,\infty\};1/2)$	`1.7789825959517732`	23	$= C_{3b}$ via $(X,Y) \mapsto (-1X+0Y,\ 1X+1Y)$
$\mathrm{SD}(\{0,1,2,\infty\};-2)$	`1.6747177094971875`	128	$= C_{3c}$ via $(X,Y) \mapsto (0X-2Y,\ -1X+0Y)$
$\mathrm{SD}(\{0,1,2,4,\infty\};-1)$	`1.5648493859472928`	108	no published bound
$\mathrm{SD}(\{0,1,2,3,4,\infty\};-1)$	`1.4463861501549340`	123	no published bound

Table 2: Unnamed members of the sum-difference family. Four are restatements of the named constants under Lemma 4; their agreement with the originals is a convergence check on the search rather than a new result.

6.1 Does support size help?

Figure 1 shows the elimination curves. The contrast is the informative part. On C_{3b} the whole path from 19 points down to 10 moves the certified value by only 6.69e-05, and the best value on the path falls at 13 points — the support size of the published record. On C_{3c} the same exercise costs 8.11e-03, about 121 times more. Whatever makes C_{3c} harder is not simply that it has one more slope; its extremal measure appears to need genuinely more geometry. Sizes falling below the envelope (11, 14, 15, 19 for C_{3b}, 12, 15, 17, 18, 19, 21 for C_{3c}) are failed re-optimisations, by Lemma 7.

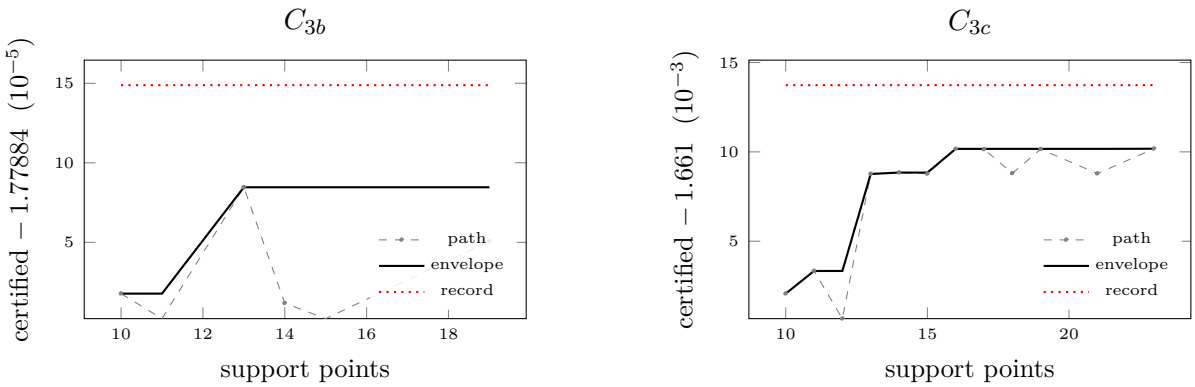

Figure 1: Backward elimination. Dashed: the raw path, with points below the frame pinned to the axis floor. Solid: the running maximum, which by Lemma 7 is itself a valid lower-bound curve. Dotted: the published record. Note the differing vertical scales and offsets; on the left one unit is 10^{-5}, on the right 10^{-3}.

6.2 Boolean functions

Best certified value by number of variables:

variables	certified
$n = 5$	`6.00000000000000`
$n = 6$	`6.34706749870192`
$n = 7$	`6.27894525270862`
$n = 8$	`6.02776461215563`
$n = 9$	`5.33123307133373`
$n = 10$	`4.31300626801112`
$n = 11$	`5.92832060630774`

Table 3: C_{71} by problem size. Small n does best: the anneal rediscovers the same $n = 6$ optimum from independent seeds, while larger n leaves it lost in a much bigger space. The published records live at $n = 14$ to 18.

6.3 What the controller chose

Over 15 episodes it pulled 11 of 32 arms, exhausting untried arms first — as optimistic initialisation dictates — and taking first certified values on 3 previously unattempted instances before switching to exploitation.

The episode worth reporting is the third. The controller chose $\mathrm{SD}(\{0, 1, 2, \infty\}; -2)$ because nothing had tried it, and the value returned was suspiciously close to C_{3c}. It *is* C_{3c}, by Lemma 4 with a determinant -2 map; our equivalence test had required $|\det| = 1$. Four of the eight unnamed instances are restatements, not three. An autonomous run found an error in the problem registry it was given.

7 What this generalises to

Nothing in the architecture is specific to either family. What it requires is the shape identified in the introduction: a lower bound witnessed by a finite explicit object with an exactly computable score. Of the roughly one hundred constants in [1], the following have that shape, listed with the bounds recorded there at the time of writing.

arm	pulls	mean reward	gains	seconds
`SD_0_1_2_inf__m2/campaign`	2	0.502	2	140
`SD_0_1_2_4_inf__m1/campaign`	2	0.500	1	190
`SD_0_1_2_3_4_inf__m1/campaign`	2	0.500	1	190
`SD_0_1_inf__1over2/shrink`	2	0.004	1	38
`SD_0_1_inf__2/refine`	1	0.003	1	95
`C71/anneal_small`	1	0.000	0	28
`C71/anneal_large`	1	0.000	0	45
`SD_0_1_inf__2/campaign`	1	0.000	0	95
`SD_0_1_inf__2/shrink`	1	0.000	0	10
`SD_0_1_inf__1over2/campaign`	1	0.000	0	95
`SD_0_1_inf__1over2/refine`	1	0.000	0	95

Table 4: Learned arm values. Campaigns on fresh instances returned most; repeated Boolean anneals returned nothing, because the search there has saturated at a local optimum it keeps rediscovering.

constant	bounds	witness	what it needs
C_{3a} GHR sum-difference	1.1835 / $\frac{4}{3}$	a finite $U \subset \mathbb{Z}$	exact counting of $U \pm U$; the interval logarithm is reused verbatim
C_9 Shannon capacity of C_7	3.2578 / 3.3177	an independent set in $C_7^{\boxtimes n}$	an independence check; the whole verifier is a few dozen lines
C_{29} kissing number, dim 5	40 / 44	a spherical code	interval arithmetic on pairwise inner products
C_{4a} cap set	2.2203 / 2.756	an extremal family	tensor-power structure, exactly as in Lemma 2
C_{49} sunflower-free	1.551 / 1.890	an extremal family	as above

Table 5: Constants whose lower bounds have the same shape as the two implemented here. C_{71} was added to this system in a day by reusing the entropy core, which is the evidence that this estimate is not optimistic.

There is a second and less obvious opportunity. The reference repository marks with an asterisk those bounds whose "level of available verification is currently at minimal levels" — records for which no replayable certificate has been published. Supplying one is a contribution its guidance explicitly asks for, and it does not require beating anybody. A system whose output is by construction a replayable certificate is unusually well placed to do that.

8 Limitations

Upper bounds are untouched. The ceilings $2 - 1/6$ and $2 - 1/4$ are theorems of Katz and Tao [2]; for C_{71} no finite ceiling is known at all. Numerical search narrows a gap only from below. Closing one from above needs proof, which is a different machine entirely.

Scale does not substitute for structure. Lemmas 2 and 3 bound what enlargement can buy: repeating a witness changes nothing.

We are behind on all three named constants. The record table for C_{3c} runs $1.61226 \to 1.668 \to 1.67471 \to 1.67473389$ on 26 points $\to 1.6747338950208249$ on 95 points. Moving from 26 to 95 support points bought an improvement in the tenth significant digit: a saturating sequence, which suggests the remaining distance to 1.75 is not reachable by adding support points to the same ansatz.

Nothing has been submitted anywhere. The reference repository is a curated academic artifact whose contribution guidance asks for bounds a reader can recompute. A submission would be appropriate only after beating a published record, with certificate and replayable checker attached.

9 Reproducibility

The system is standard-library Python apart from the search, which uses `numpy`. The trusted verifier depends on nothing beyond the standard library by design, so it can be audited on its own. 7 certificates and 566 run records accompany this document; the self-test and the cross-implementation check are single commands.

References

[1] T. Tao *et al.*, *Optimization Constants in Mathematics*, `https://github.com/teorth/optimizationproblems`.

[2] N. H. Katz and T. Tao, *Bounds on arithmetic projections, and applications to the Kakeya conjecture*, Math. Res. Lett. **6** (1999), 625–630.

[3] N. H. Katz and T. Tao, *New bounds for Kakeya problems*, J. Anal. Math. **87** (2002), 231–263.

[4] B. Green and I. Z. Ruzsa, *On the arithmetic Kakeya conjecture of Katz and Tao*, Period. Math. Hungar. **78** (2019), 135–151.

[5] M. Lemm, *New counterexamples for sums-differences*, Proc. Amer. Math. Soc. **143** (2015), 3863–3868.

[6] R. O'Donnell, J. Wright and Y. Zhou, *The Fourier entropy–influence conjecture for certain classes of Boolean functions*, APPROX/RANDOM 2011, 330–341.

[7] R. O'Donnell and L.-Y. Tan, *A composition theorem for the Fourier entropy–influence conjecture*, ICALP 2013, 780–791.

[8] R. Hod, *Improved lower bounds for the Fourier entropy/influence conjecture via lexicographic functions*, arXiv:1711.00762 (2017).