

Square-difference-free sets of exponent 0.7580758

Ordered intervals, composite digit codes, and a binary recursion

Eric Naslund*

naslund.math@gmail.com

October 6, 2026

Abstract

Let $D(N)$ be the largest cardinality of a subset of $\{1, \dots, N\}$ containing no two elements whose difference is a nonzero perfect square. We give a computer-assisted construction proving $D(N) \geq N^{0.7580758318008816-o(1)}$, improving the exponent 0.752796455874514... of Krachun's ranked-block construction. The residues we use carry subintervals of $[0, 1]$ that are ordered along every modular square difference. A finite stopping-word argument shows that if finitely many such alphabets, in pairwise coprime perfect-square moduli, have interval moments whose contributions sum to more than α , then α is an attainable exponent. Krachun's prime chains reproduce his exponent in this framework. The gain comes from an alphabet modulo a power of 4, built by a 25-state recursion, and from two composite alphabets, at the primes 5, 43 and at 19, 23, in which the first differing digit is read separately at each prime. The numerical conclusion is computer-assisted. The theorem, including every finite certificate, is also formalized in Lean 4.

***AI use:** This paper was written by GPT-6-Astra and revised by Claude Opus 5.5 under my prompting and supervision. I have had the AI write a clearer introduction and outline of the key ideas, and produce a dependency graph which can be found on page 6. However, I recommend that readers explore the paper through a dialogue with AI, rather than a traditional reading. Upload the paper to your preferred AI assistant and ask questions such as: "What are the key ideas that improve on the previous best bound?" or "Where is the meat of the argument?" In this manner, one can ask the paper questions, and understand the mathematics and the contribution through conversation, rather than directly reading the text.

CONTENTS

1.	Introduction	2
2.	Ranked alphabets and their integer realization	7
3.	Ordered interval alphabets and the moment criterion	9
4.	Odd primes: Paley chains and composite codes	11
5.	The prime 2: a recursion with parity windows	14
6.	The nine components and the proof of Theorem 1.1	16
7.	Exact finite verification	19
8.	What the construction leaves open	21
A.	Reproducing the entire finite verification	22
B.	Certificate identity	23
C.	Index of numbered statements	24

1 Introduction

1.1 The problem and the quantitative landscape

A set A of integers is *square-difference-free* if $a - a'$ is never a nonzero perfect square for $a, a' \in A$. For a positive integer N , put

$$D(N) = \max\{|A| : A \subseteq \{1, \dots, N\} \text{ is square-difference-free}\}.$$

Furstenberg and Sárközy independently proved that $D(N) = o(N)$ [1, 2]. The quantitative upper-bound problem has subsequently developed through the work of Pintz, Steiger and Szemerédi, Bloom and Maynard, and Green and Sawhney [3, 4, 5]. In particular, Green and Sawhney prove a bound of the form $D(N) \ll N \exp(-c\sqrt{\log N})$ for an absolute $c > 0$ [5, Theorem 1.1]. Our concern is the complementary construction problem. We call $\alpha \geq 0$ an *attainable exponent* if for every $\varepsilon > 0$ the bound $D(N) \geq N^{\alpha-\varepsilon}$ holds for all sufficiently large N , and we write $D(N) \geq N^{\alpha-o(1)}$ for this. The goal is to make α as large as possible. All logarithms in this paper are natural.

Ruzsa's construction gives exponent

$$\frac{1}{2} + \frac{\log 7}{2 \log 65} = 0.733077 \dots$$

by restricting alternating base-65 digits and leaving the others free [6]. Beigel and Gasarch, and independently Lewko, use a 12-element square-difference-free residue set modulo 205 to obtain

$$\frac{1}{2} + \frac{\log 12}{2 \log 205} = 0.7334117970 \dots \quad [7, 8].$$

For this construction family, a conjectural restriction on the size of square-difference-free sets modulo square-free moduli gives a natural three-quarter barrier; it is not an upper bound on $D(N)$ [6, 8]. Further bounds for square-avoiding sets modulo square-free integers are studied by Gabdullin [9]. These address local square avoidance, whereas the local objects here permit directed square differences. Computational exploration by Georgiev, Gómez-Serrano, Tao and Wagner recovered the modulus-205 construction without improving its exponent [10, Section 16]; this records a search outcome, not an optimality theorem.

In the function-field analogue, Naslund [11, Conjecture 13] conjectured that square-difference-free sets of polynomials of degree below n over $\mathbb{F}_q$ have size at most $q^{3n/4}$, for odd q and $4 \mid n$. Jones [12]

disproves this at $q = 3$ for every $n \geq 8$ divisible by 4, with a Lean-verified construction registered in Palomar.

Krachun crossed the integer three-quarter barrier by allowing modular square differences, provided that a rank orients them consistently, and then combining the ranks over coprime moduli [13]. His exponent is

$$\alpha_K = \frac{10 + \sum_{i=1}^{10} \log p_i / \log t_i}{1 + 2 \sum_{i=1}^{10} \log p_i / \log t_i} = 0.752796455874514 \dots, \quad (1)$$

where

$$(p_i, t_i) = (3, 2), (7, 3), (11, 4), (19, 5), (23, 5), (31, 7), (43, 7), (59, 9), (71, 9), (103, 11).$$

Here t_i is the length of a *Paley chain* modulo the prime p_i : a list of residues in which every later entry minus every earlier one is a nonzero square modulo p_i (Definition 4.3). Bounds for such chains explain some limitations of this particular family [14, 13]; those limitations do not apply to the alphabets of this paper.

1.2 The main result

Theorem 1.1. *Let $\alpha_0 = 473797394875551/625000000000000 = 0.7580758318008816$. For every $\varepsilon > 0$ there is $N_0(\varepsilon)$ such that, for every integer $N \geq N_0(\varepsilon)$, there exists a square-difference-free $A \subseteq \{1, \dots, N\}$ with*

$$|A| \geq N^{\alpha_0 - \varepsilon}.$$

In fact $D(N) \geq N^{\alpha_0}/Q$ for every $N \geq 1$, where Q is a fixed (astronomically large) constant.

Relative to (1), the increase in the exponent is approximately 0.0052793759. The comparison concerns Krachun's construction just described, not an upper bound on the true optimum.

1.3 From Ruzsa to Krachun to ordered intervals

All three constructions start from residues modulo a perfect square and build integers from them digit by digit. They differ in how they treat square differences between the residues.

Free digits (Ruzsa). Let m be square-free and let S be a set of residues modulo m , no two of which differ by a square modulo m . Ruzsa takes the integers whose base- m digits lie in S at even positions and are arbitrary at odd positions. Equivalently, the $m|S|$ residues $s + mu$ ($s \in S$, $0 \leq u < m$) modulo m^2 have no square difference at all, and integers are formed from them in base m^2 . The *free digits* u cost nothing, which is the source of the term $1/2$ in the exponent $\frac{1}{2} + \log|S|/(2 \log m)$ (Remark 2.11). This exceeds $3/4$ only if $|S| > m^{1/2}$, which the conjectural restriction mentioned above would rule out.

Ranks (Krachun). Krachun keeps the free digits but allows the restricted digits to have square differences, provided these can be oriented by a rank. A *ranked alphabet* modulo a perfect square b (Definition 2.3) is a set of q residues with a rank r taking h values, its *height*, such that $r(y) > r(x)$ whenever $y - x$ is a nonzero square modulo b . Choosing integer representatives that decrease as the rank increases removes every square difference, at the price of a factor h in the length of the ambient interval (Lemma 2.4). Ranks of alphabets in coprime moduli add rather than multiply (Lemma 2.5), so each alphabet can be assessed on its own: if one takes words of length k_i over the i th alphabet with $h_i^{k_i}$ roughly equal to a common H , the glued alphabet has height about nH , while

its size and modulus are the products of the $q_i^{k_i}$ and the $b_i^{k_i}$. Letting $H \rightarrow \infty$ in this bookkeeping, an alphabet with $h \geq 2$ has *contribution*

$$\varphi = \frac{\log q - \alpha \log b}{\log h} \quad (2)$$

at a target exponent α , and α is attainable when the contributions of the alphabets sum to more than α (Remark 3.3 and Theorem 3.4). Krachun’s alphabets are Paley chains: for a prime $p \equiv 3 \pmod{4}$ and a chain $s_0, \dots, s_{t-1}$ modulo p , the residues $s_j + pu$ modulo p^2 , with rank j , give $q = pt$, $b = p^2$ and $h = t$ (Lemma 4.4). Their contributions are $1 - (2\alpha - 1) \log p / \log t$, and the condition that the ten contributions sum to more than α is exactly $\alpha < \alpha_K$ (Remark 4.6).

Ordered intervals (this paper). We replace the rank of a residue x by an interval $I_x = [a_x, a_x + w_x] \subseteq [0, 1]$ and require that I_x lie entirely before I_y whenever $y - x$ is a nonzero square (Definition 3.1). A rank of height h is the special case of h consecutive intervals of equal length. What governs an alphabet is now its *moment* $Z_f = \sum_x w_x^f$: the alphabet has contribution f at exponent α if $Z_f \geq b^\alpha$, and the *interval-moment criterion* (Theorem 3.4) shows that α is attainable whenever such contributions sum to more than α . Its proof reduces to Krachun’s mechanism: it selects words of digits whose intervals all have nearly the same width, rounds the intervals to ranks, and glues the components. For equal widths the criterion returns (2) exactly, so intervals change nothing for Paley chains. They pay off for alphabets whose square differences are far from forming a chain: residues with no square difference between them may share most of an interval, and only residues on long directed paths of square differences need narrow intervals. Intervals are what make the binary component below possible, and they improve the composite codes slightly. The composite codes, on the other hand, would already help within Krachun’s framework.

Composite codes at two primes. Instead of one prime at a time, we restrict the digits at two primes jointly (Section 4.3). A point of such a code is a pair of words of base- p digits, one for each prime, and two points are related only if, *at each prime separately*, their words agree or differ first (that is, at the least significant differing digit) by a nonzero square (Definition 4.7). Lemma 4.9 adds free digits and yields an alphabet modulo the square of $F = \prod_p p^{e_p}$, where e_p is the number of restricted digits at p . The relation at one prime can thus be oriented by the other. At $(5, 43)$, the prime $5 \equiv 1 \pmod{4}$, useless for chains because its square relation is symmetric, separates points without ordering them, as in Ruzsa’s construction, while 43 orients the remaining relations, as in Krachun’s (Remark 4.8). Our $(5, 43)$ code is built from a 17-point code with one digit at each prime, whose longest directed path has 11 points; even as a ranked alphabet it contributes 0.0255 at α_0 , against 0.0023 for Krachun’s chain modulo 43. At $(19, 23)$ the corresponding code has 27 points, two more than a product of maximal chains, and contributes 0.0597 as a ranked alphabet, against 0.0557 for the chain modulo 19 (the chain modulo 23 contributes nothing at α_0). Our $(19, 23)$ code takes three digits at 19 and two at 23.

The prime 2. Modulo 4^m , a square difference is either divisible by 4, in which case the factor 4 can be cancelled, or odd and then congruent to 1 modulo 8 (Lemma 5.1). Alphabets modulo 4^m can therefore be built recursively, one base-4 digit at a time. A *state* of the recursion records two subintervals of $[0, 1]$, its *windows*, which hold the intervals of the even and of the odd residues (Section 5). Each state is built from up to four child states, its *branches*, placed in its windows by affine maps of different scales, so the widths are unequal by construction. A policy with 25 states and 94 branches makes the moment grow geometrically with m . The earlier constructions use only odd moduli; here the prime 2 contributes 0.155, the largest gain over Krachun’s construction.

Table 1 shows where the improvement comes from. At our exponent α_0 , Krachun’s chains fall short by about 0.19. The binary component supplies most of the difference, and the composite codes the rest. Of the composite codes’ gain of 0.0366, 0.0295 is available with ranks; unequal widths add

Alphabet	Contribution at $\alpha = \alpha_0$	
	with ranks (2)	with intervals
Paley chains at 3, 7, 11, 31, 59, 103	0.50854	0.50854
Paley chains at 19 and 43	0.05805	not used
Composite code at (5, 43)	0.02550	0.02650
Composite code at (19, 23)	0.06201	0.06812
Binary recursion modulo $4^{10^{20}}$	—	0.15492
Total of Krachun’s chains (positive terms)	0.56659	
Total of the components of this paper		0.75808

Table 1: Where the improvement comes from. An exponent α is attainable when the contributions of the alphabets used sum to more than α ; all entries are evaluated at the exponent α_0 of Theorem 1.1. The eight of Krachun’s chains with positive contributions supply only 0.5666 (his chains at 23 and 71 contribute -0.0056 and -0.0013 there; all ten give 0.5597). We replace the chains at 19 and 43 by the two composite codes, gaining 0.0366, and add the binary component, gaining 0.1549. In the first column a composite code is ranked by the number of points on the longest directed path of square differences ending at each point; these paths have 1331 and 686 points. The second column uses the certified unequal widths of Section 6. The binary recursion has unequal widths by construction, and we know no ranked counterpart. The first-column values for the composite codes are illustrations, not used in the proof.

0.0010 and 0.0061. The nine alphabets of Section 6 have contributions, exact rationals, that sum to $\alpha_0 + 15468032462 \cdot 10^{-25}$. The exponent α_0 is the 16-decimal truncation of the largest exponent these alphabets can certify when the depth of the binary recursion may grow (Definition 3.2 and Section 6.5).

1.4 Organization and dependency graph

The proof of Theorem 1.1 has four steps.

- Section 2 develops the mechanism of Ruzsa and Krachun: ranked alphabets, their integer realization, the gluing of coprime components, and amplification to every N . Nothing in it is new in substance; it fixes notation and proofs.
- Section 3 introduces ordered interval alphabets and proves the interval-moment criterion. This is the main new general tool.
- Sections 4 and 5 build alphabets at the odd primes and at the prime 2, and reduce the criterion’s hypotheses for them to finitely many checkable inequalities.
- Section 6 supplies nine explicit alphabets and checks those inequalities. These certificates are the only statements established by computation; Section 7 explains how they are checked.

Figure 1 shows how the numbered results depend on one another. Its columns are Sections 2–5 and its bottom band is Section 6, each read from top to bottom; an arrow $X \rightarrow Y$ means that the statement or proof of Y uses X . Thick borders mark the new ingredients. A reader who wants the method should follow the two left columns down to Theorem 3.4 (about five pages). A reader who wants the new alphabets should read Sections 4.3 and 5, where Lemmas 4.9, 5.3 and 5.5 carry the weight. A reader who wants to check the numbers should read the bottom band, Section 6, and

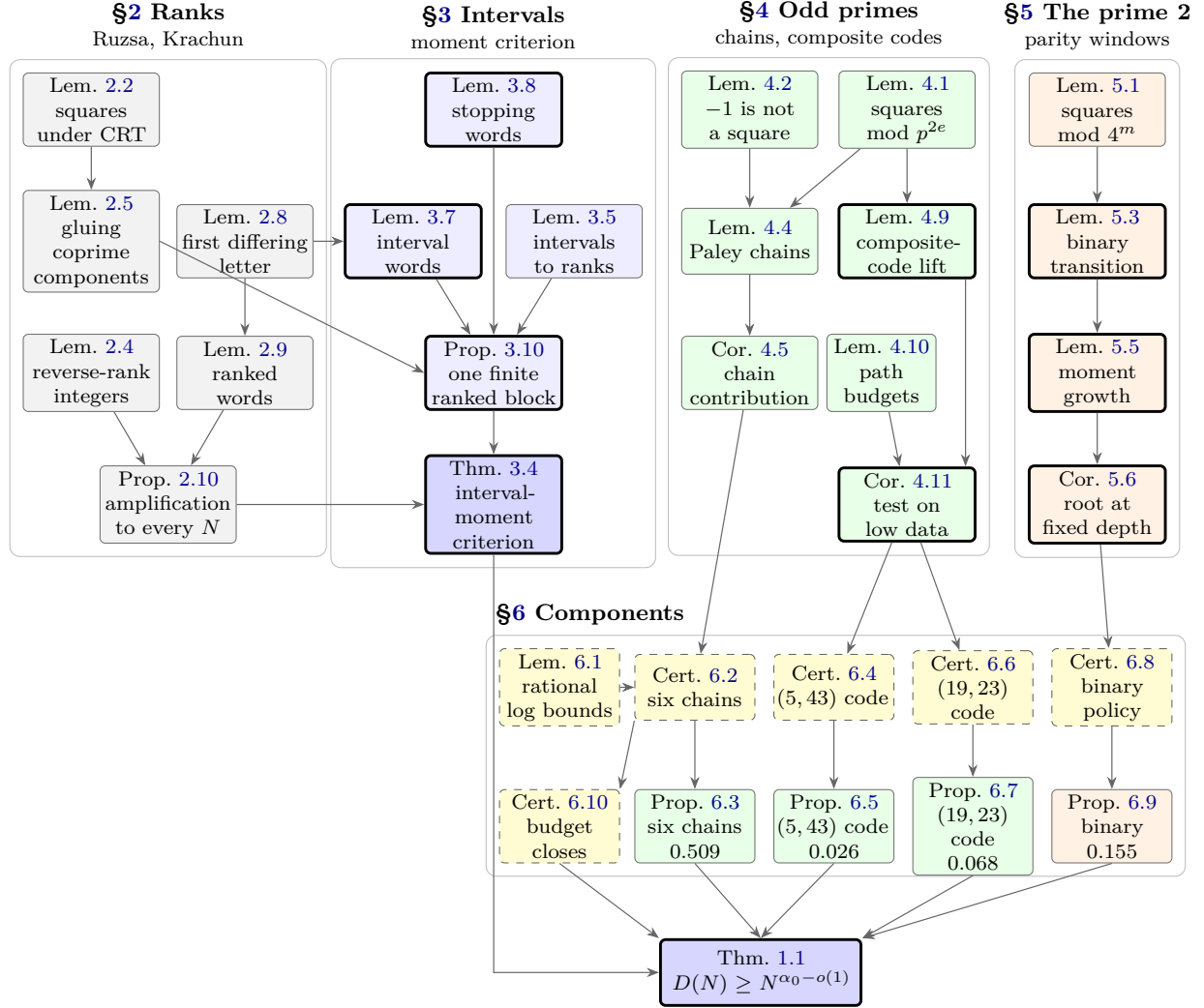


Figure 1: Dependencies between the numbered results. Each framed column is one section, read from top to bottom; an arrow $X \rightarrow Y$ means that Y uses X , and arrows implied by a path of drawn arrows are omitted. Thick borders mark the new ingredients; the remaining statements restate known arguments or elementary facts, with proofs. Dashed nodes are certificates, established by exact computation (Section 6). All four component certificates use the rational logarithm bounds of Lemma 6.1, and the budget certificate uses all four; to keep the figure legible only one arrow of each kind is drawn. The numbers in the component propositions are their contributions (Table 2), which sum to more than α_0 . Lemma 2.2 is also used in Lemma 4.9. Definitions, remarks and the one-line Lemmas 2.7 and 3.9 are not shown. Abbreviations: Lem. lemma, Cor. corollary, Prop. proposition, Thm. theorem, Cert. certificate.

Appendix A. Definitions and remarks are not shown in the figure; each definition is stated where its term first occurs, and Appendix C lists every numbered statement with the statements it uses.

1.5 Scope, computer assistance and formalization

Theorem 1.1 is a computer-assisted theorem. Its analytic and combinatorial implications are proved in the text; the specific finite rational inequalities collected in the certificates of Section 6 are established by the supplied executable programs. No optimization output, floating-point eigenvalue, or claim that a search was exhaustive is an assumption of the proof.

The L^AT_EX source contains a complete Python attachment, including all witnesses and checking programs. Compiling that one source file writes the attachment and embeds it in the PDF. Section 7 specifies its data and acceptance conditions, and Appendix A gives extraction and execution instructions. The large integer arrays are distributed as an attachment, not as thousands of typeset rows. Standard Python 3 and exact integer and rational arithmetic suffice.

A separate Lean 4 development proves Theorem 1.1 itself [15, 16]. It proves Theorem 3.4, the lifting and recursion lemmas, and every finite condition of the certificates in Section 6, which Lean’s kernel checks with its own exact certificates; its proofs use only Lean’s standard axioms. It bypasses Lemma 4.10 by recording interval starts explicitly, and Lemma 4.2 by checking the orientation of every chain pair. The development is publicly available at <https://github.com/enaslund/sarkozy-lower-bound-0.758>. The English argument here is complete on its own and uses no Lean theorem as a premise.

2 Ranked alphabets and their integer realization

This section develops the mechanism behind the constructions of Ruzsa and Krachun. The integer realization and the gluing of ranks over coprime moduli are Krachun’s Lemmas 4 and 5 [13]; we use the opposite rank convention and include short proofs.

Definition 2.1 (Square residues and square differences). For an integer $b \geq 1$ let $\mathcal{Q}(b) = \{z^2 \bmod b : z \in \mathbb{Z}\}$ be the set of square residues modulo b , zero included. Residues are represented canonically in $\{0, \dots, b-1\}$. For distinct residues x, y we write $x \rightarrow_b y$, and say that $y - x$ is a *square difference* modulo b , if $y - x \in \mathcal{Q}(b)$.

The relation is directed: it may hold in one direction, in both, or in neither. For composite b , a nonzero square residue may vanish modulo some prime factor of b ; invertibility is never assumed.

Lemma 2.2 (Squares under the Chinese remainder theorem). *Let $M_1, \dots, M_n$ be pairwise coprime with product M . Reduction $x \mapsto (x \bmod M_i)_i$ is a bijection from residues modulo M to tuples of residues, and if $y - x \in \mathcal{Q}(M)$ then $y_i - x_i \in \mathcal{Q}(M_i)$ for every i .*

Proof. The first statement is the Chinese remainder theorem. If $y - x \equiv z^2 \pmod{M}$, reduce this congruence modulo each M_i . Some coordinates may satisfy $y_i = x_i$. $\square$

Definition 2.3 (Ranked alphabet). A *ranked alphabet* modulo b is a finite set C of residues modulo b together with a function $r : C \rightarrow \{0, \dots, h-1\}$, its *rank*, such that $x \rightarrow_b y$ implies $r(x) < r(y)$ for all $x, y \in C$. The integer $h \geq 1$ is its *height*.

Residues without square differences between them may share a rank. In particular a ranked alphabet of height one has no square differences at all.

Lemma 2.4 (Reverse-rank realization). *A ranked alphabet C of size q modulo M with height h gives the square-difference-free set $\{T(x) = 1 + x + M(h - 1 - r(x)) : x \in C\}$ of size q in $\{1, \dots, Mh\}$.*

Proof. The numbers $T(x)$ lie in the stated range and are distinct modulo M . If $T(y) - T(x) = z^2 > 0$, then $x \neq y$ and reduction modulo M gives $x \rightarrow_M y$, so $r(y) \geq r(x) + 1$. But then $T(y) - T(x) = (y - x) - M(r(y) - r(x)) \leq (M - 1) - M < 0$. $\square$

This is the only place where integers, rather than residues, appear.

Lemma 2.5 (Gluing coprime components). *Let C_i be ranked alphabets modulo pairwise coprime M_i , with q_i elements and heights h_i ($1 \leq i \leq n$). The residues modulo $M = \prod_i M_i$ whose reductions lie in the C_i form a ranked alphabet with $\prod_i q_i$ elements, rank $r = \sum_i r_i$, and height $1 + \sum_i (h_i - 1)$.*

Proof. By Lemma 2.2 there are $\prod_i q_i$ such residues, and a square difference modulo M is a square difference or an equality in every coordinate. Equal coordinates keep their rank and unequal ones raise it; at least one coordinate is unequal. The rank range follows. $\square$

Heights add while sizes multiply; this is what lets each component be assessed separately.

Definition 2.6 (Square base and words). A *square base* is an integer $b = s^2 > 1$. A *word* of length k is a sequence $(x_0, \dots, x_{k-1})$ of residues modulo b ; it *encodes* the residue $X = \sum_{j < k} x_j b^j$ modulo b^k , so the first letter is the least significant digit.

Encoding is injective, by uniqueness of base- b expansions.

Lemma 2.7. *If $s \geq 1$ and $z \in \mathbb{Z}$ satisfy $s^2 \mid z^2$, then $s \mid z$.*

Proof. For every prime ℓ , $2v_\ell(s) \leq 2v_\ell(z)$, where v_ℓ denotes the ℓ -adic valuation. $\square$

Lemma 2.8 (First differing letter). *Let $b = s^2 > 1$, and let distinct words (x_j) and (y_j) of length k encode X and Y . If $Y - X \in \mathcal{Q}(b^k)$ and j is the first position with $x_j \neq y_j$, then $x_j \rightarrow_b y_j$.*

Proof. Let $Y - X \equiv z^2 \pmod{b^k}$. The integer $Y - X$ is divisible by b^j , and $j < k$, so $s^{2j} = b^j$ divides z^2 and $s^j \mid z$ by Lemma 2.7. Dividing the congruence by b^j and reducing modulo b gives $y_j - x_j \equiv (z/s^j)^2 \pmod{b}$. $\square$

Since b is a square, a square difference of words is decided by the first letter where they differ. This is why all bases in this paper are squares.

Lemma 2.9 (Ranked words). *If C is a ranked alphabet modulo a square base b , of size q and height h , then its words of length k form a ranked alphabet modulo b^k of size q^k and height h^k , ranked by $R(X) = \sum_j r(x_j) h^{k-1-j}$.*

Proof. Let $X \rightarrow_{b^k} Y$, with first differing position j . By Lemma 2.8, $r(y_j) \geq r(x_j) + 1$, which raises R by at least h^{k-1-j} ; the later positions lower it by at most $\sum_{i > j} (h - 1) h^{k-1-i} < h^{k-1-j}$. $\square$

Proposition 2.10 (Amplification). *Let $\alpha \geq 0$. If a ranked alphabet of size Q modulo a square base M has height h and $(Mh)^\alpha \leq Q$, then $D(N) \geq N^\alpha / Q$ for every $N \geq 1$.*

Proof. Put $L = Mh > 1$. By Lemma 2.9, words of length t form a ranked alphabet of size Q^t modulo M^t with height h^t , and Lemma 2.4 turns it into a square-difference-free set of size Q^t in $\{1, \dots, L^t\}$ (for $t = 0$ this is $\{1\}$). Each $N \geq 1$ has $t \geq 0$ with $L^t \leq N < L^{t+1}$; then $N^\alpha \leq L^{(t+1)\alpha} \leq Q^{t+1}$, so $D(N) \geq Q^t \geq N^\alpha / Q$. $\square$

Remark 2.11 (Ruzsa's construction). Let m be square-free and let S be a set of residues modulo m with $s' - s \notin \mathcal{Q}(m)$ for distinct $s, s' \in S$. The $m|S|$ residues $s + mu$ ($s \in S, 0 \leq u < m$) modulo m^2 have no square differences. Indeed, if two of them differ by a square modulo m^2 , their low digits agree, by reduction modulo m ; the difference is then $m(u' - u)$ with $0 < |u' - u| < m$, and a prime $p \mid m$ with $p \nmid u' - u$ exists because m is square-free. Modulo p^2 that difference has p -adic valuation exactly one, which no square residue has. So these residues form a ranked alphabet of height one, and Proposition 2.10, with $h = 1$, gives the exponent $\log(m|S|)/\log m^2 = \frac{1}{2} + \log|S|/(2\log m)$. Its words are exactly Ruzsa's integers with alternating restricted and free base- m digits.

3 Ordered interval alphabets and the moment criterion

This section introduces the main new general tool. Ranks are replaced by intervals, and the cardinality and height of an alphabet by a single moment.

Definition 3.1 (Ordered interval alphabet, moment, test). An *ordered interval alphabet* modulo b is a finite set C of residues modulo b , each $x \in C$ carrying an interval $I_x = [a_x, a_x + w_x] \subseteq [0, 1]$ of width $w_x > 0$, such that $x \rightarrow_b y$ implies $a_x + w_x \leq a_y$ for all $x, y \in C$. For $f \geq 0$ its *moment* is

$$Z_f(C) = \sum_{x \in C} w_x^f.$$

If b is a square base and $Z_f(C) \geq b^\alpha$, we say that C *passes the test at α with contribution f* .

Along a cycle of square differences the positive widths would have to sum to at most zero, so such an alphabet has no cycles. Residues without a square difference between them may have overlapping intervals.

Definition 3.2 (Admissible contributions and exact crossing). Let C be an ordered interval alphabet in a square base b with all widths below one. A power $f \geq 0$ is *admissible at α* if C passes the test at α with contribution f . Since Z_f is strictly decreasing in f , the admissible powers form an interval $[0, f^*(\alpha)]$ when $Z_0(C) = |C| \geq b^\alpha$, and no power is admissible otherwise; $f^*(\alpha)$ is the *largest admissible contribution*. For finitely many fixed alphabets, $\alpha \mapsto \sum_i f_i^*(\alpha) - \alpha$ is strictly decreasing; its zero α^* is their *exact crossing*, and every $\alpha < \alpha^*$ admits admissible contributions summing to more than α .

Remark 3.3 (Ranks are equal widths). A ranked alphabet of size q and height $h \geq 2$ in a square base b becomes an ordered interval alphabet with $I_x = [r(x)/h, (r(x) + 1)/h]$. All its widths are $1/h$, so $Z_f = qh^{-f}$, and it passes the test at α with contribution f exactly when $f \leq \varphi$, the quantity of (2). Since equal widths are one possible choice of intervals, optimizing the widths can only increase the largest admissible contribution (Definition 3.2).

Theorem 3.4 (Interval-moment criterion). *Let $C_1, \dots, C_n$ ($n \geq 1$) be ordered interval alphabets in pairwise coprime square bases b_i , with all widths in $[\sigma, \rho]$ for some $0 < \sigma \leq \rho < 1$. If $\alpha \geq 0$, $f_i \geq 0$,*

$$Z_{f_i}(C_i) \geq b_i^{\alpha} \quad (1 \leq i \leq n) \quad \text{and} \quad \Phi := \sum_{i=1}^n f_i > \alpha,$$

then α is attainable. More precisely, $D(N) \geq N^\alpha/Q$ for all $N \geq 1$ and a constant Q depending only on the alphabets.

The proof occupies the rest of this section. It selects one word length in each component (Lemma 3.8), turns the selected words into one finite ranked alphabet (Proposition 3.10), and amplifies that alphabet (Proposition 2.10).

Lemma 3.5 (Rounding intervals to ranks). *If every interval of an ordered interval alphabet has width at least $\gamma > 0$, then $r(x) = \lfloor a_x/\gamma \rfloor$ is a rank of height at most $\lceil 1/\gamma \rceil$.*

Proof. If $x \rightarrow_b y$, then $a_y \geq a_x + w_x \geq a_x + \gamma$, so $r(y) \geq r(x) + 1$. Moreover $0 \leq a_x \leq 1 - \gamma$. $\square$

Definition 3.6 (Word intervals). Let C be an ordered interval alphabet modulo a square base b . For $x \in C$ let $F_x(t) = a_x + w_x t$, the increasing affine map from $[0, 1]$ onto I_x . The word $(x_0, \dots, x_{k-1})$ encoding X receives the interval $I_X = F_{x_0} \circ \dots \circ F_{x_{k-1}}([0, 1])$, of width $W_X = \prod_j w_{x_j}$.

Thus the first letter chooses a subinterval of $[0, 1]$, the second a subinterval of that, and so on.

Lemma 3.7 (Interval words). *With the word intervals of Definition 3.6, the words of length k over C form an ordered interval alphabet modulo b^k , with moment $Z_f(C)^k$.*

Proof. Let $X \rightarrow_{b^k} Y$, with first differing position j . By Lemma 2.8, $x_j \rightarrow_b y_j$, so I_{x_j} lies before I_{y_j} . The words' intervals are the images of subintervals of I_{x_j} and I_{y_j} under the common increasing map $F_{x_0} \circ \dots \circ F_{x_{j-1}}$, which preserves their order. The moment factors because widths multiply. $\square$

Lemma 3.8 (Stopping and selecting a length). *Fix an ordered interval alphabet in a square base b with widths in $[\sigma, \rho]$, $\rho < 1$, a power $f \geq 0$, and $q = b^\alpha$, and suppose $Z = Z_f \geq q$. For every integer $K \geq 1$ and $\delta = \rho^K$ there are $1 \leq k \leq K$ and a set D of words of length k with*

$$\sigma\delta < W_X \leq \delta \quad (X \in D), \quad q^k \leq (K+1)|D|\delta^f.$$

Proof. Grow words from the empty word, of width one. A child word stays *active* while its width exceeds δ and otherwise becomes *terminal*; only active words are extended. Every word of length K has width at most $\rho^K = \delta$, so nothing is active at length K . A terminal word's parent was wider than δ , so its own width exceeds $\sigma\delta$. Let A_j be the sum of W^f over active words of length j , and T_j the sum over terminal words of length $j+1$. Multiplicativity of widths gives

$$A_0 = 1, \quad A_K = 0, \quad A_{j+1} + T_j = ZA_j \quad (0 \leq j < K).$$

If $T_j < q^{j+1}/(K+1)$ for every $j < K$, induction with $Z \geq q$ gives $A_j \geq q^j(K+1-j)/(K+1)$, contradicting $A_K = 0$. Choose j with $T_j \geq q^{j+1}/(K+1)$, let $k = j+1$, and let D be the terminal words of length k . Since $W \leq \delta$ and $f \geq 0$, $T_j \leq |D|\delta^f$. $\square$

The selected words have widths within a factor σ of each other, so little is lost when their intervals are rounded to ranks.

Lemma 3.9 (Common width bounds). *A finite family of ordered interval alphabets whose widths all lie in $(0, 1)$ admits common bounds $0 < \sigma \leq w \leq \rho < 1$.*

Proof. Take the minimum and maximum of the finitely many widths together with $1/2$. $\square$

No numerical value of σ is needed, even for the very large binary alphabet of Section 5.

Proposition 3.10 (One finite ranked block). *Under the hypotheses of Theorem 3.4 there is a ranked alphabet with Q elements modulo a square base M , of height h , such that $(Mh)^\alpha \leq Q$.*

Proof. Fix K (chosen below) and put $\delta = \rho^K$. Apply Lemma 3.8 in each component with this K , obtaining lengths k_i and word sets D_i with widths in $(\sigma\delta, \delta]$ and $b_i^{\alpha k_i} \leq (K+1)|D_i|\delta^{f_i}$. By Lemma 3.7 the selected words form ordered interval alphabets modulo $b_i^{k_i}$, and Lemma 3.5 with $\gamma = \sigma\delta$ ranks them below $H = \lceil 1/(\sigma\delta) \rceil$. Lemma 2.5 glues them into a ranked alphabet with $Q = \prod_i |D_i|$ residues modulo the square base $M = \prod_i b_i^{k_i}$, of height $h = 1 + n(H-1) \leq \kappa/\delta$, where $\kappa = (n+1)(1+\sigma^{-1})$ (using $\delta \leq 1$). Multiplying the length estimates gives $M^\alpha \leq Q\delta^\Phi(K+1)^n$. Since $\alpha \geq 0$,

$$(Mh)^\alpha \leq Q \cdot \kappa^\alpha (K+1)^n \rho^{K(\Phi-\alpha)}.$$

Because $\Phi > \alpha$ and $\rho < 1$, the factor $\kappa^\alpha (K+1)^n \rho^{K(\Phi-\alpha)}$ tends to zero as $K \rightarrow \infty$ (the ratio of consecutive terms tends to $\rho^{\Phi-\alpha} < 1$). Fix K with that factor at most one. $\square$

Proof of Theorem 3.4. Proposition 3.10 and then Proposition 2.10. All alphabets, then K , and hence M , h and Q are fixed before N varies. For every $\varepsilon > 0$, eventually $N^\varepsilon \geq Q$. $\square$

Remark 3.11 (Relation to entropy). For a probability vector π on the letters of an alphabet, put

$$H(\pi) = - \sum_x \pi_x \log \pi_x, \quad R(\pi) = - \sum_x \pi_x \log w_x.$$

With $\nu_x = w_x^f / Z_f$, the inequality $\log u \leq u - 1$ gives $\sum_x \pi_x \log(\nu_x / \pi_x) \leq 0$, that is $H(\pi) - fR(\pi) \leq \log Z_f$, with equality when $\pi = \nu$ [17]. Words in which each letter x occurs with frequency π_x then give an alternative proof of Theorem 3.4. Its exponent is $U/(V+1)$, where $U = \sum_i H_i/R_i$ and $V = \sum_i \log b_i/R_i$, and the budget $\sum_i f_i > \alpha$ is exactly $U - \alpha V > \alpha$. The quantity R is the *rank cost* of a component, and its contribution can be read as $f = (H - \alpha \log b)/R$, which generalizes (2). The stopping proof above needs neither this identity nor Stirling's formula.

4 Odd primes: Paley chains and composite codes

4.1 Squares modulo odd prime powers

Lemma 4.1 (Square differences modulo p^{2e}). *Let p be prime and $e \geq 1$, and let $x \not\equiv y \pmod{p^{2e}}$ satisfy $y - x \equiv z^2 \pmod{p^{2e}}$. Let $d \in (0, p^{2e})$ represent $y - x$ and $k = v_p(d)$. Then k is even, $k < 2e$, and d/p^k is a nonzero square modulo p .*

Proof. Let $v = v_p(z)$. If $2v \geq 2e$ then $d \equiv 0$, which is impossible. Otherwise $d = z^2 + tp^{2e}$ for an integer t , and $v_p(z^2) = 2v < 2e \leq v_p(tp^{2e})$, so $k = 2v$. Writing $z = p^v z'$ with $p \nmid z'$ and dividing by p^{2v} gives $d/p^{2v} \equiv z'^2 \pmod{p}$. $\square$

In terms of base- p digits: if two integers agree in all digits below position $k < 2e$ and differ at position k , a square difference modulo p^{2e} forces k to be even and the digit difference at position k to be a nonzero square modulo p . We call the digits in even positions *low digits* and those in odd positions *free digits*.

Lemma 4.2. *Let $p \equiv 3 \pmod{4}$ be prime. Then $-1 \notin Q(p)$, and for every residue $d \not\equiv 0$ exactly one of $d, -d$ is a nonzero square modulo p .*

Proof. An element g with $g^2 \equiv -1$ would have order four in the cyclic group $(\mathbb{Z}/p\mathbb{Z})^\times$ of order $p-1 \not\equiv 0 \pmod{4}$. The nonzero squares have index two in that group, so $-d = (-1)d$ is a square exactly when d is not. $\square$

Thus for $p \equiv 3 \pmod{4}$ every two distinct residues modulo p are related in exactly one direction: square differences modulo p form a tournament, the *Paley tournament*. For $p \equiv 1 \pmod{4}$, in contrast, -1 is a square, and $x \rightarrow_p y$ holds exactly when $y \rightarrow_p x$.

4.2 Paley chains

Definition 4.3 (Paley chain). Let $p \equiv 3 \pmod{4}$ be prime. A *chain* of length t modulo p is a list of distinct residues $s_0, \dots, s_{t-1}$ such that $s_j - s_i$ is a nonzero square modulo p whenever $i < j$.

Lemma 4.4 (Prime-chain alphabet). For a chain $s_0, \dots, s_{t-1}$ modulo $p \equiv 3 \pmod{4}$, the pt residues $s_j + pu$ ($0 \leq u < p$) modulo p^2 , with interval $[j/t, (j+1)/t]$ on every residue $s_j + pu$, form an ordered interval alphabet with moment $Z_f = pt^{1-f}$. Equivalently, they form a ranked alphabet of height t with rank j .

Proof. If two residues have different low digits $s_i \neq s_j$, a square difference modulo p^2 reduces to a nonzero square modulo p , which by Lemma 4.2 and the chain property points from s_i to s_j with $i < j$; the intervals are ordered. If the low digits agree, a nonzero difference has p -adic valuation one, which Lemma 4.1 forbids. There are p residues with each interval, which gives the moment. $\square$

This is the local mechanism of [13]; the free digit u multiplies the number of residues by p without any ordering cost, as in [6].

Corollary 4.5 (Chain contribution). Let $t \geq 2$ and $f_p^* = 1 - (2\alpha - 1) \log p / \log t$. For $0 \leq f \leq f_p^*$ the chain alphabet of Lemma 4.4 passes the test at α with contribution f , that is $pt^{1-f} \geq (p^2)^\alpha$.

Proof. t^{1-f} decreases in f , and $pt^{1-f_p^*} = p \cdot p^{2\alpha-1} = p^{2\alpha}$. $\square$

A chain is useful only when $f_p^* > 0$, that is $t > p^{2\alpha-1}$.

Remark 4.6 (Krachun's exponent). With $\Lambda = \sum_{i=1}^n \log p_i / \log t_i$, the budget $\sum_{i=1}^n f_{p_i}^* > \alpha$ for n chains reads $n - (2\alpha - 1)\Lambda > \alpha$, that is

$$\alpha < \frac{n + \Lambda}{1 + 2\Lambda}.$$

For Krachun's ten chains this is $\alpha < \alpha_K$ of (1). So Theorem 3.4, applied to Paley chains alone, reproduces Krachun's exponent. Unequal widths cannot help a chain when $0 \leq f \leq 1$: residues with different low digits differ by a square modulo p^2 in one direction (a nonzero square modulo p lifts to one modulo p^2), so choosing one residue for each low digit gives a directed path, and the widest residues w_j of the t low digits satisfy $\sum_j w_j \leq 1$. The moment is then at most $p \sum_j w_j^f \leq pt^{1-f}$, by concavity of $w \mapsto w^f$. A larger exponent needs alphabets that are not chains.

4.3 Composite codes

Definition 4.7 (Low code and the prime-specific relation). Fix a finite set $\mathcal{P}$ of odd primes and depths $e_p \geq 1$, the numbers of restricted digits at each $p \in \mathcal{P}$, and put $F = \prod_{p \in \mathcal{P}} p^{e_p}$. A *low point* $\mathbf{a}$ is a tuple of digit strings $\mathbf{a}_p = (a_{p,0}, \dots, a_{p,e_p-1})$ with $0 \leq a_{p,j} < p$, one for each prime; we call $\mathbf{a}_p$ the *low word* of $\mathbf{a}$ at p (it is not read in a square base, unlike the words of Definition 2.6). For distinct low points write $\mathbf{a} \rightsquigarrow \mathbf{b}$ if, at every prime p , either $\mathbf{a}_p = \mathbf{b}_p$ or the first index j with $a_{p,j} \neq b_{p,j}$ satisfies

$$b_{p,j} - a_{p,j} \in \mathcal{Q}(p) \setminus \{0\}. \quad (3)$$

A *low code* is a finite set S of low points, each carrying an interval $I_{\mathbf{a}} \subseteq [0, 1]$ of width $w_{\mathbf{a}} > 0$, such that $\mathbf{a} \rightsquigarrow \mathbf{b}$ implies $\sup I_{\mathbf{a}} \leq \inf I_{\mathbf{b}}$. A *directed path* of $\rightsquigarrow$ is a sequence $\mathbf{a}^{(0)} \rightsquigarrow \mathbf{a}^{(1)} \rightsquigarrow \dots$ of points.

The first differing index may depend on p , and no condition is imposed at a prime whose word is unchanged. A relation needs agreement at every prime, so a pair of points is unrelated as soon as one prime separates it.

Remark 4.8 (How two primes cooperate). Take depth one at the primes 5 and 43, so a low point is a pair (a_5, a_{43}) . The nonzero squares modulo 5 are 1 and 4, so the 5-digits of related points are equal or differ by ± 1 , in either direction. Points whose 5-digits differ by ± 2 are never related, whatever their 43-digits, so their intervals may overlap freely, as in Ruzsa's construction. Points whose 5-digits are related are related in the direction given by their 43-digits, which Lemma 4.2 orients, as in Krachun's. A code must avoid pairs with equal 43-digits and 5-digits differing by ± 1 , which would be related in both directions. The 17-point code of Section 6.2 uses all five 5-digits; a chain modulo 43 has at most 7 elements, by an exhaustive search.

Lemma 4.9 (Prime-coordinate free-digit lift). *Let S be a low code. For $\mathbf{a} \in S$ and free digits $u_{p,j} \in \{0, \dots, p-1\}$ put*

$$x_p = \sum_{j < e_p} (a_{p,j} p^{2j} + u_{p,j} p^{2j+1}) \pmod{p^{2e_p}}, \quad (4)$$

and combine the x_p by the Chinese remainder theorem. Giving every such residue the interval of its low point yields an ordered interval alphabet in the square base F^2 , with $F|S|$ residues and moment $Z_f = F \sum_{\mathbf{a} \in S} w_{\mathbf{a}}^f$. The depths e_p may differ between primes.

Proof. Distinctness and the count follow from unique p -adic expansions and the Chinese remainder theorem. Let lifted residues $x \neq y$ have a square difference modulo F^2 . By Lemma 2.2, every prime coordinate with $x_p \neq y_p$ has a square difference modulo p^{2e_p} . By Lemma 4.1 its first differing p -adic digit sits at an even position $2j$, which carries a low digit, and the digit difference there is a nonzero square modulo p . All lower positions agree, so the low words agree before index j : this is the first differing low index, and (3) holds. If all low words agreed, some coordinate would differ first at an odd (free) position, which Lemma 4.1 forbids. Hence the low points $\mathbf{a}, \mathbf{b}$ of x, y differ and $\mathbf{a} \rightsquigarrow \mathbf{b}$, so the intervals are ordered. Each low point has F lifted copies, which gives the moment. $\square$

Requiring a single common first-difference index at all primes would impose extra relations that this proof does not need; reading the first difference separately at each prime is what lets a low code exceed a product of chains.

A low code is most conveniently given by its widths alone. The following lemma supplies the intervals.

Lemma 4.10 (Path budgets). *Let a finite directed acyclic graph have positive vertex weights w_x . There are intervals in $[0, 1]$ of these widths, ordered along every edge, if and only if every directed path has total weight at most one.*

Proof. Ordered intervals along a path have disjoint interiors, so their widths sum to at most one. Conversely, let d_x be the largest weight of a path ending at x , including x , and put $a_x = d_x - w_x \geq 0$. Then $a_x + w_x = d_x \leq 1$, and an edge $x \rightarrow y$ gives $d_y \geq d_x + w_y$, that is $a_y \geq a_x + w_x$. $\square$

This is the weighted-chain viewpoint underlying the chain polytope of a finite poset [18]; the elementary proof above supplies the part used here.

Corollary 4.11 (Odd test on low data). *Let S be a finite set of low points with positive widths $w_{\mathbf{a}}$. Suppose either that S , with given intervals of these widths, is a low code, or that $\rightsquigarrow$ is acyclic on S and every directed path of $\rightsquigarrow$ has total width at most one. Then S is a low code with intervals of these widths, and its lift (Lemma 4.9) passes the test at α with contribution f if and only if*

$$\log \sum_{\mathbf{a} \in S} w_{\mathbf{a}}^f \geq (2\alpha - 1) \log F.$$

Proof. In the second case Lemma 4.10 gives the intervals. The lift has moment $Z_f = F \sum_{\mathbf{a}} w_{\mathbf{a}}^f$, and $(F^2)^\alpha = F \cdot F^{2\alpha-1}$. $\square$

The number $F|S|$ of lifted residues enters only through F : a low code helps when $\sum_{\mathbf{a}} w_{\mathbf{a}}^f$ exceeds $F^{2\alpha-1}$, roughly $F^{0.516}$.

Remark 4.12 (Free-digit copies need equal widths). For a lift that keeps all free digits, the maps that add a fixed $c_{p,j}$ to each free digit $u_{p,j}$ modulo p , digit by digit, preserve square differences (they preserve the first differing p -adic position and the low digits) and act transitively on the copies of each low point. Averaging widths over them keeps all path budgets and, by concavity of $w \mapsto w^f$, does not decrease the moment. Thus giving different widths to the copies of one low point cannot help; improvements must change the low code itself.

5 The prime 2: a recursion with parity windows

Modulo 4^m there are no free digits in the sense of Section 4. Instead, square differences have a recursive structure.

Lemma 5.1 (Square differences modulo 4^m). *Let $m \geq 2$. (i) If $4d \equiv z^2 \pmod{4^m}$ then $z = 2z'$ and $d \equiv z'^2 \pmod{4^{m-1}}$. (ii) A residue $\equiv 2$ or $3 \pmod{4}$ is never a square modulo 4^m . (iii) An odd square residue modulo 4^m is $\equiv 1 \pmod{8}$.*

Proof. (i) Since $4 \mid z^2$, z is even; divide by 4. (ii) Squares modulo 4 are 0 and 1. (iii) $(2k+1)^2 = 4k(k+1) + 1$ and $8 \mid 4^m$. $\square$

Write a residue Y modulo 4^m as $Y = r + 4x'$ with $r \in \{0, 1, 2, 3\}$ and x' a residue modulo 4^{m-1} . By the lemma, two residues with the same r differ by a square exactly when their x' do, modulo 4^{m-1} (the converse of (i) is clear: $4z'^2 = (2z')^2$); residues whose r differ by 2 never differ by a square; and an odd square difference moves from a residue class k modulo 8 to the class $k+1$, cyclically. An alphabet modulo 4^m is therefore made of up to four alphabets modulo 4^{m-1} , its *branches*, placed so that the residue classes modulo 8 that occur occupy successive parts of $[0, 1]$, in cyclic order; since the order is cyclic, some class must be absent. The classes modulo 8 of the residues in a branch are determined by the parities of its residues x' , which is why the recursion tracks parities.

5.1 States and transitions

Definition 5.2 (Parity-window policy). A *state* s has an *even window* E_s and an *odd window* O_s : closed subintervals of $[0, 1]$ of positive length, each possibly absent, at least one present. State s is *valid at depth* m if it carries an ordered interval alphabet modulo 4^m in which every residue's interval lies in the window of the residue's parity.

A *policy* assigns to each state at most one *branch* for each $r \in \{0, 1, 2, 3\}$. A branch specifies a child state j , a scale $u > 0$, a shift $\theta \in \mathbb{R}$ and bits $\tau, \eta \in \{0, 1\}$ (*swap* and *reflect*). A child residue x modulo 4^{m-1} becomes

$$x' = ((-1)^\eta x + \tau) \bmod 4^{m-1}, \quad y = r + 4x' \pmod{4^m}; \quad (5)$$

its interval $[a, a+w]$ is first reflected to $[1-a-w, 1-a]$ if $\eta = 1$ and then mapped by $z \mapsto \theta + uz$. The parity of x' is that of x toggled by τ . For transformed parity $\pi \in \{0, 1\}$ let $J_{r+4\pi}$ be the image

of the child's window of parity $\pi \oplus \tau$ under the same maps; J_k is indexed by the residue class k of the parent residue modulo 8. The policy is *valid* if, whenever the windows involved are present,

$$J_{r+4\pi} \subseteq E_s \quad (r \text{ even}), \quad J_{r+4\pi} \subseteq O_s \quad (r \text{ odd}), \quad (6)$$

$$\sup J_k \leq \inf J_{k+1 \bmod 8} \quad (0 \leq k < 8). \quad (7)$$

Condition (7) includes the comparison from class 7 to class 0; it can hold only because some classes are absent. Reflection and swap let a branch reuse a child state in another orientation or parity.

Lemma 5.3 (Binary transition). *Let $m \geq 2$. If the children are valid at depth $m - 1$ and the policy is valid, then the branches define a valid state at depth m .*

Proof. Different branches have different residues modulo 4, and (5) is injective within a branch, so all residues are distinct. Containment (6) puts every interval in the window of its parity. Let $Y - X$ be a nonzero square modulo 4^m . *Same branch:* $Y - X = 4(y' - x')$, and by Lemma 5.1(i) $y' - x'$ is a square modulo 4^{m-1} . Without reflection the child's order applies directly. With reflection, $y' - x'$ is the negative of the difference of the child residues, so the child's square difference points the other way and the reflected intervals reverse it back. The shift cancels in differences and the positive scale preserves order. *Residues differing by 2 modulo 4:* excluded by Lemma 5.1(ii). *Odd difference:* by Lemma 5.1(iii), $Y - X \equiv 1 \pmod{8}$, so the classes of X, Y modulo 8 are $k, k + 1$ cyclically; their intervals lie in J_k and J_{k+1} , which (7) orders. $\square$

5.2 Moment growth and a finite depth

Definition 5.4 (Growth certificate). Let $0 \leq f \leq 1$. A *growth certificate* for a policy is a vector $v > 0$ indexed by states and a number $a > 1$ such that

$$\sum_{\text{branches } s \rightarrow j} u^f v_j \geq a v_s \quad \text{for every state } s. \quad (8)$$

Let d_s be the length of the wider present window of s and $c = \min_s d_s / v_s > 0$. The *branch matrix* B_f has entries $(B_f)_{s,j} = \sum_{\text{branches } s \rightarrow j} u^f$, so (8) reads $B_f v \geq a v$; this forces a to be at most the *Perron root* of B_f , its largest eigenvalue.

This positive-vector condition is a familiar device for bounding the growth of a nonnegative matrix [19]; here only the following elementary induction is needed.

Lemma 5.5 (Moment growth). *Under a valid policy with a growth certificate, every state is valid at every depth $m \geq 1$, with moment $Z_{s,m} \geq c a^{m-1} v_s$.*

Proof. At depth one, take the single residue 0 or 1 whose parity window is the wider one, with that whole window as its interval. A singleton has no square differences, and its moment is $d_s^f \geq d_s \geq c v_s$ because $0 < d_s \leq 1$ and $f \leq 1$. Reflection and shift preserve widths, scaling multiplies a moment by u^f , and branches are disjoint, so $Z_{s,m} = \sum_{s \rightarrow j} u^f Z_{j,m-1}$. Induct using (8); validity comes from Lemma 5.3. $\square$

Corollary 5.6 (Root at a fixed depth). *Suppose some root state has both windows equal to $[0, 1]$ and $v_{\text{root}} = 1$. Then for every $m \geq 1$ there is an ordered interval alphabet modulo 4^m , with all widths at most $1/2$, whose moment satisfies $Z_f \geq 2^{-f} c a^{m-1}$. In particular,*

$$\log c + (m - 1) \log a - f \log 2 - \alpha m \log 4 > 0 \implies Z_f > (4^m)^\alpha. \quad (9)$$

Proof. The root alphabet of Lemma 5.5 has no window restriction. Halving every width at its left endpoint keeps all orders, makes every width at most $1/2$, and multiplies the moment by 2^{-f} . $\square$

The depth m is any fixed integer, chosen before N varies. A very large depth affects only constants, and the alphabet is never listed explicitly. The binary component contributes f whenever some growth certificate has $a > 4^\alpha$ and m is large enough. As m grows, its admissible contributions therefore approach the power f at which the Perron root of B_f equals 4^α .

6 The nine components and the proof of Theorem 1.1

We now fix all data. Every terminating decimal in this section denotes an exact rational. Each *certificate* below is a finite list of exact rational or integer inequalities; Section 7 describes the programs that check them. Table 2 summarizes the components. All decisive numerical comparisons use the following rational bounds for logarithms.

Lemma 6.1 (Rational logarithm bounds). *For $1 \leq x \leq 2$ and $z = (x - 1)/(x + 1)$,*

$$\log x = 2 \sum_{j=0}^{n-1} \frac{z^{2j+1}}{2j+1} + R_n, \quad 0 \leq R_n \leq \frac{2z^{2n+1}}{(2n+1)(1-z^2)}.$$

Consequently, let $\lambda^-(x) \leq \log x \leq \lambda^+(x)$ be the resulting rational bounds, extended to all positive rationals by writing $x = 2^k y$ with $1 \leq y \leq 2$ and adding k times the bounds for $\log 2$ (with lower and upper bounds of $\log 2$ interchanged when $k < 0$). Then for rationals $x > 0$, $L > 0$ and $f \geq 0$ the inequality $\lambda^+(L) \leq f\lambda^-(x)$ implies $L \leq x^f$.

Proof. Subtracting the series for $\log(1 - z)$ from that of $\log(1 + z)$ gives the expansion; all terms are positive. The tail is bounded by replacing every denominator by $2n + 1$ and summing the geometric series in $z^2 \leq 1/9$. For the consequence, $\log L \leq \lambda^+(L) \leq f\lambda^-(x) \leq f \log x$, and the exponential function is increasing. $\square$

Section 7.3 describes how these bounds are implemented.

Component	Square base b_i	Exact contribution f_i
Chain at 3 ($t = 2$)	3^2	0.1819189685063644313459576
Chain at 7 ($t = 3$)	7^2	0.0857695922261077385453455
Chain at 11 ($t = 4$)	11^2	0.1072043074619092931875010
Chain at 31 ($t = 7$)	31^2	0.0891366647927413831505830
Chain at 59 ($t = 9$)	59^2	0.0421426390460550181119771
Chain at 103 ($t = 11$)	103^2	0.0023654891206769647316637
Low code at 5, 43, depths (3, 3)	$(5^3 \cdot 43^3)^2 = 215^6$	0.0264983230432418544045862
Low code at 19, 23, depths (3, 2)	$(19^3 \cdot 23^2)^2$	0.0681150585658841330427119
Binary root, depth $m = 10^{20}$	$4^{10^{20}}$	0.1549247890379023302829202
Total		0.7580758318008831468032462

Table 2: The nine components, their bases and their contributions. The total exceeds $\alpha_0 = 0.7580758318008816$ by $15468032462 \cdot 10^{-25}$.

6.1 Prime chains

Certificate 6.2 (Six prime chains). The following lists are chains modulo the indicated primes, all $\equiv 3 \pmod{4}$: for each of the 244 ordered pairs of distinct entries, $s_j - s_i$ is a square modulo p exactly when $j > i$.

p	t	Chain $s_0, \dots, s_{t-1}$
3	2	0, 1
7	3	0, 1, 2
11	4	0, 1, 4, 5
31	7	0, 1, 8, 5, 2, 9, 10
59	9	0, 1, 28, 17, 22, 4, 26, 20, 29
103	11	0, 1, 29, 92, 2, 61, 17, 30, 93, 18, 19

For each chain, the contribution f_p in Table 2 satisfies $f_p > 0$ and $(1 - f_p)\lambda^-(t) - (2\alpha_0 - 1)\lambda^+(p) > 0$ with the rational bounds of Lemma 6.1; the smallest of these six margins exceeds $2.4 \cdot 10^{-26}$.

Proposition 6.3 (Chain components). *For each of the six primes there is an ordered interval alphabet modulo p^2 , with widths in $[1/11, 1/2]$, passing the test at α_0 with contribution f_p .*

Proof. The certified margin gives $(1 - f_p)\log t > (2\alpha_0 - 1)\log p$, that is $f_p < f_p^*$. Apply Lemma 4.4 to the chains of Certificate 6.2 and Corollary 4.5. $\square$

6.2 The composite code at 5 and 43

The low code at the primes 5 and 43 is the three-letter product S^3 of the following 17-point code S of depth one, given as pairs (a_5, a_{43}) :

$$(0, 18), (3, 25), (1, 27), (4, 29), (0, 24), (0, 22), (3, 42), (2, 5), (1, 19), \\ (2, 40), (3, 22), (4, 35), (3, 3), (0, 33), (4, 39), (1, 28), (2, 20).$$

A point of S^3 has the word (x_5, y_5, z_5) at the prime 5 and the word (x_{43}, y_{43}, z_{43}) at 43, built from three points of S . As explained in Remark 4.8, the prime 5 separates points and the prime 43 orients every related pair. The widths are not products of widths of S ; they were optimized for the full relation $\rightsquigarrow$ of Definition 4.7, whose longest directed paths have 1331 points.

Certificate 6.4 (The $(5, 43)$ low code). The witness consists of $4,913 = 17^3$ distinct low points with depths $(3, 3)$, listed in a topological order, and integer width numerators over the denominator 10^{14} . Each of the 6,031,254 related pairs $\mathbf{a} \rightsquigarrow \mathbf{b}$ has $\mathbf{a}$ before $\mathbf{b}$ in the list, and the largest total width of a directed path of $\rightsquigarrow$ is $99999974386028/10^{14} < 1$. With $f_{215} = 0.0264983230432418544045862$ (we index the two composite components by $215 = 5 \cdot 43$ and $437 = 19 \cdot 23$),

$$\log \sum_{\mathbf{a} \in S^3} w_{\mathbf{a}}^{f_{215}} - (2\alpha_0 - 1)\log(215^3) > 6.919 \cdot 10^{-18}. \quad (10)$$

Proposition 6.5 (The $(5, 43)$ component). *There is an ordered interval alphabet in the square base 215^6 , with all widths in $(0, 1)$, passing the test at α_0 with contribution f_{215} .*

Proof. The certificate shows that $\rightsquigarrow$ is acyclic on the code and bounds every directed path by one. Corollary 4.11, with $F = 215^3$ and (10), gives the lifted alphabet and the test. $\square$

6.3 The composite code at 19 and 23

The low code at the primes 19 and 23 has *unequal depths*: three low digits at 19 and two at 23, so $F = 19^3 \cdot 23^2 = 3,628,411$. Its 3,645 points form a product $B \times \{0, 1, 6, 17, 7\}$. Here B is a 729-point low code with depths (2, 2), and the list 0, 1, 6, 17, 7 is a chain modulo 19 that supplies the third 19-digit. Two points that agree in their first two 19-digits are compared at 19 through this chain. All other comparisons at 19 are decided by the first two digits. The code B descends from the 27-point depth-one code

$$\begin{aligned} & (10, 6), (14, 6), (2, 6), (10, 19), (0, 16), (0, 19), (11, 16), (0, 20), (11, 19), \\ & (11, 20), (10, 9), (16, 10), (16, 19), (10, 22), (17, 10), (16, 22), (17, 19), (14, 9), \\ & (17, 22), (14, 22), (2, 9), (2, 22), (16, 12), (0, 5), (1, 12), (11, 5), (17, 12) \end{aligned}$$

(pairs (a_{19}, a_{23})), which exceeds the 25 points of a product of two maximal chains. The code B was obtained from it by conditional transformations of the second letters, by exchanges of points, and finally by eighteen rounds of exhaustive single-point exchanges that maximize the moment. None of this history is used: only the final points and intervals enter the proof. The longest directed paths of $\rightsquigarrow$ in this code have 686 points.

Certificate 6.6 (The (19, 23) low code). The witness consists of 3,645 distinct low points with depths (3, 2), listed in a topological order, with integer interval starts and widths over the denominator 10^{16} . Each of the 2,702,580 related pairs $\mathbf{a} \rightsquigarrow \mathbf{b}$ has $\mathbf{a}$ before $\mathbf{b}$ in the list and $\sup I_{\mathbf{a}} \leq \inf I_{\mathbf{b}}$, and the largest right endpoint is $9999999218567850/10^{16} < 1$. With $f_{437} = 0.0681150585658841330427119$,

$$\log \sum_{\mathbf{a}} w_{\mathbf{a}}^{f_{437}} - (2\alpha_0 - 1) \log(19^3 \cdot 23^2) > 5.927 \cdot 10^{-18}. \quad (11)$$

Proposition 6.7 (The (19, 23) component). *There is an ordered interval alphabet in the square base $(19^3 \cdot 23^2)^2$, with all widths in $(0, 1)$, passing the test at α_0 with contribution f_{437} .*

Proof. The certificate shows that the code, with its certified intervals, is a low code. Corollary 4.11, with depths $(e_{19}, e_{23}) = (3, 2)$, $F = 19^3 \cdot 23^2$ and (11), gives the lifted alphabet and the test. $\square$

6.4 The binary component

Certificate 6.8 (The binary policy). The witness specifies 25 states, 94 branches with rational windows, scales and shifts, and a positive rational vector v with denominator 10^{45} . The root is state 1 (states are numbered from 0), its windows are both $[0, 1]$, and $v_1 = 1$. The exact parameters are

$$\begin{aligned} f_B &= 0.1549247890379023302829202, & m &= 10^{20}, \\ a &= \frac{1430135321718334301514215293356149372101}{5 \cdot 10^{38}}, \end{aligned}$$

and c is the exact rational $\min_s d_s/v_s > 0.3551$. The policy is valid: 304 containment and cyclic-order comparisons (6), (7) hold, including the wrap from class 7 to class 0. The growth rows and margins satisfy

$$\min_s \left(\sum_{s \rightarrow j} w^{f_B} v_j - a v_s \right) > 3.27 \cdot 10^{-19}, \quad (12)$$

$$\log a - \alpha_0 \log 4 > 4.91 \cdot 10^{-19}, \quad (13)$$

$$\log c + (m - 1) \log a - f_B \log 2 - \alpha_0 m \log 4 > 46.91. \quad (14)$$

The tiny margins in (12) and (13) are deliberate: a lies between 4^{α_0} and the Perron root of the branch matrix B_{f_B} (Definition 5.4), and the large depth m converts the small gap (13) into the comfortable margin (14).

Proposition 6.9 (The binary component). *There is an ordered interval alphabet modulo $4^{10^{20}}$, with all widths in $(0, 1/2]$, passing the test at α_0 with contribution f_B .*

Proof. By Certificate 6.8, the policy is valid and (v, a) is a growth certificate. Lemmas 5.3 and 5.5 and Corollary 5.6, with (14), give the alphabet and its moment. $\square$

6.5 Closing the bound

Certificate 6.10 (Coprimalty and contribution budget). The prime supports $\{2\}$, $\{3\}$, $\{7\}$, $\{11\}$, $\{31\}$, $\{59\}$, $\{103\}$, $\{5, 43\}$ and $\{19, 23\}$ of the nine components are pairwise disjoint, so the nine square bases are pairwise coprime. The exact sum of the contributions in Table 2 is

$$\sum_i f_i = 0.7580758318008831468032462 = \alpha_0 + \frac{15468032462}{10^{25}}.$$

The surplus is tiny by design. The contributions of the odd components lie about 10^{-18} below their largest admissible values at α_0 (Definition 3.2). For the binary component the relevant threshold is the power f at which the Perron root of B_f equals 4^{α_0} , the limit of its admissible contributions as the depth grows (Section 5); f_B lies about 10^{-18} below it. With this convention, α_0 is the 16-decimal truncation of the exact crossing $\alpha^* = 0.75807583180088164538\dots$ of the nine alphabets. With the same codes, chains and policy geometry, any exponent below α^* could be certified by retuning the contributions, a and m ; none above α^* can be without changing a code or the policy.

Proof of Theorem 1.1. Propositions 6.3, 6.5, 6.7 and 6.9 give nine ordered interval alphabets in pairwise coprime square bases (Certificate 6.10), each passing the test at α_0 with the contribution of Table 2. They are finite with widths in $(0, 1)$, so Lemma 3.9 supplies common bounds σ, ρ . The contributions sum to more than α_0 (Certificate 6.10). Theorem 3.4 gives $D(N) \geq N^{\alpha_0}/Q$ for all $N \geq 1$, with all finite choices, including the binary depth, made before N varies. $\square$

7 Exact finite verification

7.1 Data sufficient to define the construction

The embedded archive contains a manifest and three witness files, identified by the hashes in Appendix B, together with the checking programs. The manifest fixes α_0 , the two odd contributions, the six chains and the SHA-256 hashes of the three witnesses. The binary witness fixes its own contribution, depth, vector and growth constant a . The rational chain check produces the six chain contributions of Table 2. Numerical fields left over from discovery are ignored.

For an odd witness, `primes` lists the coordinate primes, and either `depth` (a common depth) or `depths` (one per prime) gives the number of low digits. The array `points` lists the low words, one word per prime, least significant digit first. A positive integer `width_denominator` D and positive integers `width_numerators` n_j define $w_j = n_j/D$. The (19, 23) file also gives `start_numerators` s_j , defining $a_j = s_j/D$; the (5, 43) starts are defined by Lemma 4.10. Thus the data specify every interval without reference to the programs that found them.

In the binary witness, a state's window record $[a, b, c, d, q]$ means $E_s = [a/q, b/q]$ and $O_s = [c/q, d/q]$; the pair $c = d = -1$ denotes an absent odd window. Each branch lists its residue **r**, **child**, **swap** and **reflect** bits, and rational **scale** and **shift**. Equations (5)–(8) specify precisely what these data mean.

7.2 Complete geometric checks

Below, an *edge* is a related pair $\mathbf{a} \rightsquigarrow \mathbf{b}$ of low points, or a pair $X \rightarrow Y$ of residues.

The (5, 43) code. The checker verifies prime validity, distinctness of points, all digit ranges and word lengths, positivity of the widths, and every ordered pair of distinct points. For each pair it evaluates (3) separately in each coordinate, and every edge must point forward in the supplied list. It then computes the path numerators

$$d_j = n_j + \max(\{d_i : i \rightsquigarrow j\} \cup \{0\})$$

and checks $\max_j d_j \leq D$. This is the recurrence of Lemma 4.10; induction in the checked topological order proves its validity.

The (19, 23) code. The checker computes the same full relation with integer bitsets and per-prime depths. For one prime p with depth e , let $G(v)$ be the bitset of points whose coordinate word begins with the prefix v . For a target word $b = (b_0, \dots, b_{e-1})$ the allowable sources have bitset

$$G(b) \cup \bigcup_{j=0}^{e-1} \bigcup_{q \in \mathcal{Q}(p) \setminus \{0\}} G((b_0, \dots, b_{j-1}, b_j - q) \bmod p). \quad (15)$$

The first term covers identical words; each remaining term fixes the first difference and permits all later digits. Hence (15) is exactly the relation required at that prime. Intersecting over the two primes and removing the target enumerates every incoming edge. The checker verifies that every incoming index precedes its target. It then sorts points by right endpoint, sweeps targets by left endpoint while maintaining the bitset of points whose right endpoints exceed the current start, and requires an empty intersection with each incoming bitset; this is equivalent to $s_i + n_i \leq s_j$ on every edge $i \rightsquigarrow j$. Sizes, ranges, distinctness, starts, widths and endpoints are checked separately. The counts in Section 6 are outputs of these complete checks.

The binary policy. The geometry checker constructs every J window with exact fractions and checks (6) and (7), together with unique branch residues, valid child indices, positive scales and valid bits. Its 304 comparisons count two endpoint containments per present placed window and one inequality per present adjacent pair. Lemma 5.3 is what makes these finitely many conditions sufficient at every depth.

7.3 Rational arithmetic

The bounds $\lambda^-(x) \leq \log x \leq \lambda^+(x)$ of Lemma 6.1 are implemented in three ways. The (5, 43) checker uses 32 terms with exact fractions, and the binary checker uses 70. The (19, 23) checker uses 48 terms with outward rounding at scale $S = 2^{192}$: if $L_z \leq Sz \leq U_z$, it sets $L_{z^2} = \lfloor L_z^2/S \rfloor$ and $U_{z^2} = \lceil U_z^2/S \rceil$, multiplies successively with lower products rounded down and upper products rounded up, rounds each division by $2j + 1$ in the same directions, and adds the scaled tail bound

$\lceil 2U_{\text{next}}S/((2n+1)(S-U_{z^2})) \rceil$, where U_{next} bounds Sz^{2n+1} . Induction preserves enclosure through every operation [20].

A positive rational proposal L for a power x^f is accepted as a lower bound only when

$$\lambda^+(L) \leq f\lambda^-(x), \quad (16)$$

as in Lemma 6.1. Decimal arithmetic proposes candidates; it never decides validity, and a poor proposal causes rejection rather than a false positive. For the odd components the proposals are integer multiples of 10^{-24} ; with accepted $L_{\mathbf{a}} \leq w_{\mathbf{a}}^f$ the checkers prove $\lambda^-(\sum_{\mathbf{a}} L_{\mathbf{a}}) - (2\alpha_0 - 1)\sum_p e_p \lambda^+(p) > 0$, which gives (10) and (11). For the binary component, accepted lower powers give lower bounds for the rows of (8), and

$$\lambda^-(c) + (m-1)\lambda^-(a) - f_B \lambda^+(2) - \alpha_0 m \lambda^+(4) > 0$$

proves (14) without approximating the astronomically large alphabet. The chain check proves $(1 - f_p)\lambda^-(t) - (2\alpha_0 - 1)\lambda^+(p) > 0$ for each of the six rational contributions, and Certificate 6.10 is an exact rational addition performed by the chain-contribution program. The main verifier and the independent arithmetic check close the budget more generously: they use the certified lower bounds for the six f_p^* themselves instead of their 25-decimal truncations, and therefore report a slightly larger surplus ($1.5468032465 \cdot 10^{-15}$).

How the parameters were chosen. For fixed witnesses each component has a largest admissible contribution at a given α , and the exponent is limited by the crossing at which these sum to α . The odd and binary contributions in Table 2 were proposed by computing their largest admissible values with 60-digit decimal arithmetic and subtracting 10^{-18} . Each chain contribution is the 25-decimal truncation of the certified rational lower bound $1 - (2\alpha_0 - 1)\lambda^+(p)/\lambda^-(t)$ for f_p^* . The binary vector v is a Perron vector of the branch matrix rounded to denominator 10^{45} . The value a lies strictly between 4^{α_0} and the Perron root. The depth 10^{20} makes (14) positive despite the gap (13) of order 10^{-19} . These choices only propose data; the exact checks above decide everything.

7.4 Independent arithmetic and the trust boundary

A further program in the attachment rechecks the moments, the binary growth and depth inequalities, and the global budget with a different expansion,

$$\log x = \sum_{k=1}^n \frac{t^k}{k} + R'_n, \quad t = 1 - 1/x \in [0, 1/2], \quad 0 \leq R'_n \leq \frac{t^{n+1}}{(n+1)(1-t)},$$

for $1 \leq x \leq 2$, with 230 terms, outward-rounded integer bounds at scale 10^{60} , and power proposals with denominator 10^{40} . It does not replace the geometry checks; it provides independently implemented arithmetic for the same pinned witnesses. The Lean development of Section 1.5 checks the same finite conditions a third time, with separately generated certificates and Lean's kernel.

The complete supplied run accepts all the finite conditions of Section 6. The correctness of the numerical theorem uses the execution of these integer/rational programs, not merely their printed messages; the implications of the conditions are proved in the text.

8 What the construction leaves open

The proof isolates finite quantities that future work can improve: low codes and their interval moments, the grouping of primes into coprime components, and the binary transition geometry.

The exponent α_0 is a certified feasible exponent, not an asserted optimum for the method or for $D(N)$. The possible limitations of Paley chains discussed in [13, 14] do not bound composite and binary interval alphabets. The very large fixed depths make the construction useless for moderate N ; the objective is the asymptotic exponent.

Several features of the present components indicate where further gains could come from, without restricting future approaches.

- For a single prime $p \equiv 3 \pmod{4}$ one can show that chains are optimal: two residues with different first low digits are always comparable, so the first low digits of any alphabet modulo p^{2^e} form a chain (an acyclic tournament is transitive), and concavity of $w \mapsto w^f$ and the multiplicativity of moments then give at most $(pt^{1-f})^e$. For the prime 2, one can show that the window recursion loses nothing: an odd residue is a square modulo 4^m exactly when it is $\equiv 1 \pmod{8}$ (Lemma 5.1 and Hensel lifting), so cross-branch square relations depend only on residue classes modulo 8, and every alphabet modulo 4^m arises by one transition from its four branches, with the convex hulls of the class intervals as windows. The open question there is the best growth rate of that recursion.
- Gains beyond products of chains come from composite low codes whose cardinality exceeds the product of chain lengths, such as the 27-point (19, 23) code above. In the searches behind this paper, such excess appeared only for primes whose maximal chain length is small for their size, most notably 23. Moreover, such codes compose badly. For a product of two chains, the prime-specific relation makes heights add, and the rank cost per letter tends to the logarithm of a single chain length. For the 27-point code S , the rank cost of $S \times S$ (as a depth-two code) is about twice that of S , the rate of a lexicographic product. A code family that kept the excess at every level with additive heights would raise the (19, 23) contribution substantially; none is known.
- A different integer realization than Lemma 2.4, one that avoids only actual squares rather than all modular square residues, would change the overall trade-off; it would require a new amplification argument.

A Reproducing the entire finite verification

The attachment is a self-contained Python program carrying a compressed archive of the exact data and all programs needed by the checks.

A PDF viewer supporting embedded files can save this attachment. Alternatively, compiling the single `square-difference-free-sets-of-exponent-0.7580758318.tex` source with `pdflatex` writes that same Python file beside the build outputs. The source contains the actual archive bytes, not a link to them. Run

```
python3 square-difference-free-certificate.py
```

The program checks the embedded archive hash, extracts into a temporary directory, and runs the complete geometry and moment verifier, the rational chain-contribution check, and the independent arithmetic check. It makes no network requests and needs no third-party Python packages. To retain the extracted data, source code and generated reports, choose a fresh output directory:

```
python3 square-difference-free-certificate.py --extract certificate-check
```

The run finishes with

PAPER CERTIFICATE PASS: complete geometry, rational target powers, and independent arithmetic.

The wrapper and all entry points reject Python's `-O` and `-OO` modes, because some routines use assertions as checks. Compilation of the PDF does not execute these mathematical checks; they are a separate step.

The principal verification files in the extracted archive are:

- `research-notes/improved-bound/verify_bound.py`: manifest, components, chains, coprimality and the global budget (with the certified lower bounds for the chain contributions);
- `research-notes/odd-search/verify_candidate.py`: complete pairwise low-code and longest-path checking for the $(5, 43)$ code, exact rational logarithms;
- `research-notes/odd-push/verify_unequal.py`: full-edge bitset checking for the $(19, 23)$ code with per-prime depths, directed fixed-point logarithms;
- `research-notes/binary-search/check_geometry.py` and `research-notes/binary-search/certify.py`: binary transition geometry, growth rows and the fixed-depth moment;
- `research-notes/lean-planning/rational_chain_powers.py`: the six rational chain contributions (run with 25 decimals) and the exact rational addition of Certificate 6.10;
- `research-notes/binary-push/independent_arithmetic.py`: independent arithmetic with the second logarithm series.

The archive preserves the repository file names for reproducibility; they do not introduce dependencies on any other files. Discovery-only functions present in some modules are not executed.

B Certificate identity

The following SHA-256 hashes fix the exact bytes of the data files in the embedded archive. Hashes identify witnesses; the finite checks, not the hashes, establish their mathematical properties.

Manifest: `research-notes/capacity-bound/bound.json`

080735a362d5fb736ae932d4c775e74c6c05ce3a25119e17cc0f1b928022bf84

Odd $(5, 43)$: `research-notes/odd-search/q215-depth3-product.json`

dae38a756de83c99d65973ff85f23a7068ce6e85a7a4f12400a923f24cf46574

Odd $(19, 23)$: `research-notes/next-push/q437-unequal32s.json`

3ef1e78e59f6e90327bd739c8c114604a352bb4ae7f0faecc6d547b3cd0104bc

Binary: `research-notes/next-push/binary-cert-0.7580758318008816.json`

1b388dd3cfda324a57d985fc96746a8d36b8e7d2e4ac25ecd264c854a8cb8af7

The SHA-256 of the embedded ZIP archive is

088f7876b76fd83745932beec88548df796ac02a2c3b66907608c79b721a8295.

The attached program checks this identity before unpacking.

C Index of numbered statements

The table lists every numbered statement in order, what it provides, and the statements it uses (definitions included). The last column gives the corresponding node of the machine-readable dependency graph `proof-structure/graph.json` distributed with the source files; a node may cover several statements.

Statement	Content	Uses	Node
Thm 1.1	$D(N) \geq N^{\alpha_0 - o(1)}$	3.4, 3.9, 6.3, 6.5, 6.7, 6.9, 6.10	T.MAIN
Def 2.1	square residues and differences	—	D.SQREL
Lem 2.2	squares under CRT	2.1	N.CRT
Def 2.3	ranked alphabet, height	2.1	D.RANKED
Lem 2.4	reverse-rank realization	2.3	L.REVERSE
Lem 2.5	gluing, ranks add	2.3, 2.2	L.CRT
Def 2.6	square base, words	—	D.WORDS
Lem 2.7	$s^2 \mid z^2 \Rightarrow s \mid z$	—	N.SQDIV
Lem 2.8	first differing letter	2.6, 2.7	L.WORDS
Lem 2.9	ranked words	2.3, 2.8	L.WORDS
Prop 2.10	amplification to every N	2.9, 2.4	P.AMPLIFY
Rem 2.11	Ruzsa’s construction (not used)	2.10	--
Def 3.1	ordered interval alphabet, moment, test	2.1	D.INTERVAL
Rem 3.3	ranks are equal widths (not used)	2.3, 3.1	X.CONTRIB
Def 3.2	admissible contribution, exact crossing	3.1	D.INTERVAL
Thm 3.4	interval-moment criterion	3.10, 2.10	T.CRITERION
Lem 3.5	intervals to ranks	2.3, 3.1	L.ROUND
Def 3.6	word intervals	2.6, 3.1	D.WORDS
Lem 3.7	interval words	3.6, 2.8	L.WORDS
Lem 3.8	stopping and selecting a length	3.6	L.STOP
Lem 3.9	common width bounds	3.1	L.FINITEWIDTH
Prop 3.10	one finite ranked block	3.8, 3.7, 3.5, 2.5	P.BLOCK
Rem 3.11	entropy accounting (not used)	3.1, 3.7	X.ENTROPY
Lem 4.1	square differences modulo p^{2e}	2.1	N.PADIC
Lem 4.2	-1 is a non-square, $p \equiv 3 \pmod{4}$	2.1	N.QR3MOD4
Def 4.3	Paley chain	2.1	D.CHAIN
Lem 4.4	prime-chain alphabet	4.3, 4.2, 4.1	L.CHAIN
Cor 4.5	chain contribution	4.4	C.CHAINPOWER
Rem 4.6	Krachun’s exponent (not used)	4.5, 3.4	--
Def 4.7	low code, prime-specific relation	2.1	D.LOWCODE
Rem 4.8	how two primes cooperate (not used)	4.7, 4.2	--
Lem 4.9	composite-code lift, any depths	4.7, 4.1, 2.2	L.ODDLIFT
Lem 4.10	path budgets	—	L.PATHS
Cor 4.11	odd test on low data	4.9, 4.10	C.ODDTEST
Rem 4.12	fiber averaging (not used)	4.9, 4.10	X.FIBER
Lem 5.1	square differences modulo 4^m	2.1	N.DYADIC
Def 5.2	parity-window policy	3.1	D.POLICY
Lem 5.3	binary transition	5.2, 5.1	L.BINSTEP
Def 5.4	growth certificate	5.2	D.GROWTHCERT
Lem 5.5	moment growth	5.3, 5.4	L.BINGROWTH
Cor 5.6	root at a fixed depth	5.5	C.BINDEPTH

Statement	Content	Uses	Node
Lem 6.1	rational logarithm and power bounds	—	N.LOGENC
Cert 6.2	six chains and their contributions	4.3, 4.5, 6.1	W.CHAINS
Prop 6.3	chain components	4.4, 4.5, 6.2	T.CHAINS
Cert 6.4	the (5, 43) low code	4.7, 4.11, 6.1	W.ODD215
Prop 6.5	the (5, 43) component	4.11, 6.4	T.ODD215
Cert 6.6	the (19, 23) low code	4.7, 4.11, 6.1	W.ODD437
Prop 6.7	the (19, 23) component	4.10, 4.11, 6.6	T.ODD437
Cert 6.8	the binary policy	5.2, 5.4, 5.6, 6.1	W.BINARY
Prop 6.9	the binary component	5.3, 5.5, 5.6, 6.8	T.BINARY
Cert 6.10	coprimality and budget	6.2, 6.4, 6.6, 6.8	W.BUDGET

References

- [1] H. Furstenberg, *Ergodic behavior of diagonal measures and a theorem of Szemerédi on arithmetic progressions*, J. Analyse Math. **31** (1977), 204–256. doi:10.1007/BF02813304.
- [2] A. Sárközy, *On difference sets of sequences of integers. I*, Acta Math. Acad. Sci. Hungar. **31** (1978), no. 1–2, 125–149. doi:10.1007/BF01896079.
- [3] J. Pintz, W. L. Steiger, and E. Szemerédi, *On sets of natural numbers whose difference set contains no squares*, J. London Math. Soc. (2) **37** (1988), no. 2, 219–231. doi:10.1112/jlms/s2-37.2.219.
- [4] T. F. Bloom and J. Maynard, *A new upper bound for sets with no square differences*, Compos. Math. **158** (2022), no. 8, 1777–1798. doi:10.1112/S0010437X22007679.
- [5] B. Green and M. Sawhney, *New bounds for the Furstenberg–Sárközy theorem*, 2025, arXiv:2411.17448v2 (first version 2024).
- [6] I. Z. Ruzsa, *Difference sets without squares*, Period. Math. Hungar. **15** (1984), no. 3, 205–209. doi:10.1007/BF02454169.
- [7] R. Beigel and W. Gasarch, *Square-difference-free sets of size $\Omega(n^{0.7334\dots})$* , 2008, arXiv:0804.4892v3.
- [8] M. Lewko, *An improved lower bound related to the Furstenberg–Sárközy theorem*, Electron. J. Combin. **22** (2015), no. 1, Paper 1.32, 6 pp. doi:10.37236/4656.
- [9] M. R. Gabdullin, *Sets in $\mathbb{Z}_m$ whose difference sets avoid squares*, Sb. Math. **209** (2018), no. 11, 1603–1610. doi:10.1070/SM8992.
- [10] B. Georgiev, J. Gómez-Serrano, T. Tao, and A. Z. Wagner, *Mathematical exploration and discovery at scale*, 2025, arXiv:2511.02864v1.
- [11] E. Naslund, *Paley graphs and Sárközy’s theorem in function fields*, 2022, arXiv:2203.01293v3, Conjecture 13.

- [12] JD Jones, *ns-lean: Naslund's Conjecture 13 is false at $q = 3$, $k = 2$, for every n congruent to $0 \bmod 4$ with $n \geq 8$* , Palomar, entry [PALOMAR-2026-09-17-000003](#), version 1, 17 September 2026.
- [13] D. Krachun, *Square-difference-free sets beyond the three-quarter barrier*, 2026, [arXiv:2608.01325v1](#).
- [14] S. Satake, *On the restricted isometry property of the Paley matrix*, Linear Algebra Appl. **631** (2021), 35–47. [doi:10.1016/j.laa.2021.08.018](#).
- [15] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in *Automated Deduction—CADE 28*, A. Platzer and G. Sutcliffe (eds.), Lecture Notes in Computer Science **12699**, Springer, Cham, 2021, 625–635. [doi:10.1007/978-3-030-79876-5_37](#).
- [16] The mathlib Community, *The Lean mathematical library*, in *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020)*, Association for Computing Machinery, New York, 2020, 367–381. [doi:10.1145/3372885.3373824](#).
- [17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, second ed., Wiley-Interscience, Hoboken, NJ, 2006. [doi:10.1002/047174882X](#).
- [18] R. P. Stanley, *Two poset polytopes*, Discrete Comput. Geom. **1** (1986), 9–23. [doi:10.1007/BF02187680](#).
- [19] E. Seneta, *Non-negative Matrices and Markov Chains*, second ed., Springer Series in Statistics, Springer, New York, 1981, reprinted 2006. [doi:10.1007/0-387-32792-4](#).
- [20] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009. [doi:10.1137/1.9780898717716](#).