hexagon:2611.01270v1

An Explicit Computer-Assisted Upper Bound for the Quadratic Logarithmic Divisor-Gap Constant

H. Matsunami
Independent researcher, Japan

November 28, 2026
Version 0.3: integrated submission candidate

Abstract

Let $1 = d_1 < \cdots < d_{\tau(n)} = n$ be the positive divisors of an integer $n \geq 2$, and put $F_2(n) = \sum_{i<\tau(n)} \log^2(d_{i+1}/d_i)$. The limiting mean coefficient of this statistic can be expressed as the expected sum of squared complementary-gap lengths of a Poisson–Dirichlet subset-sum set. We give a computer-assisted bound for this coefficient, namely $c_2 \leq U < 0.248794 < 1/4$, with U an explicit rational number. The proof uses six GEM stick-breaking variables and retains the remaining random set through an exact self-similar upper bound. A finite dyadic certificate reduces the estimate to polynomial integrals evaluated by integer and rational arithmetic. The argument yields $c_2 \leq A + Bc_2$ with explicit A and $0 \leq B < 1$. The quantitative coupling of Haddad and Koukoulopoulos then gives $\sum_{n\leq x} F_2(n) = c_2 x \log^2 x + O(x \log x)$. In particular, the coefficient $1/4$, suggested by an exploratory finite-range comparison, is excluded. No analytic continuation or closed-form evaluation of c_2 is claimed.

Keywords. Consecutive divisors; logarithmic gaps; Poisson–Dirichlet distribution; GEM distribution; subset sums; computer-assisted proof.

Computational verification status (November 28, 2026). The complete certificate and standalone C++17 exact-integer verifier are publicly available in artifact release v1.0.0 [5]. A replay from the tagged archive checked all 1,259,080 leaves, recovered the exact rational coefficients, and rejected ten malformed inputs. All twelve SHA-256 manifest entries and all thirteen Git blob identifiers matched. Earlier GCC and Clang runs used the same C++ algorithm; historical Python result files agree exactly, but their source programs were not rerun. Independent human specialist review and formal journal peer review have not been completed.

1 Introduction and main statements

We study the geometry of the positive divisors of an integer on a logarithmic scale. Throughout, log denotes the natural logarithm, and $F_2(1) = 0$. For $n \geq 2$, write

$$F_2(n) = \sum_{i=1}^{\tau(n)-1} \left(\log \frac{d_{i+1}}{d_i}\right)^2, \qquad 1 = d_1 < \cdots < d_{\tau(n)} = n. \tag{1.1}$$

The logarithmic gaps sum to $\log n$. Their squared sum gives greater weight to large empty intervals than to a collection of small intervals of the same total length. Our purpose is to bound a particular mean coefficient, not to introduce this statistic.

1.1 The random-set constant

For a sequence $a = (a_j)_{j\ge1}$ of nonnegative numbers with $\sum_j a_j = 1$, let

$$K(a) = \left\{ \sum_{j\ge1} \varepsilon_j a_j : \varepsilon_j \in \{0,1\} \right\}. \tag{1.2}$$

This is a compact subset of $[0,1]$ containing both endpoints. If K is such a set, denote by $\mathcal{G}(K)$ the connected components of $[0,1] \setminus K$ and define

$$H(K) = \sum_{I\in\mathcal{G}(K)} |I|^2. \tag{1.3}$$

For a ranked sequence a with distribution PD(1), set

$$c_2 = \mathbb{E}H(K(a)). \tag{1.4}$$

The following is the principal result of the certification record.

Theorem 1.1 (Computer-assisted upper bound)**.** *The constant in* (1.4) *satisfies*

$$c_2 \le U := \frac{5565525599521087701564905}{22370037828734660780893862} < \frac{248794}{10^6} < \frac{1}{4}. \tag{1.5}$$

The numerical part of the proof is a finite certificate evaluation, not a Monte Carlo estimate. The certificate is specified in Section 4; its recorded exact output is stated in Proposition 5.1. All remaining steps of the upper-bound argument are given below.

Corollary 1.2 (Integer mean)**.** *With the same* c_2*,*

$$\sum_{n\le x} F_2(n) = c_2 x \log^2 x + O(x \log x) \qquad (x \longrightarrow \infty). \tag{1.6}$$

In particular, this sum is not asymptotic to $\frac{1}{4} x \log^2 x$*.*

The transfer to integers uses [2, eq. (1.5)]. The Poisson–Dirichlet upper bound itself does not use that coupling theorem. This separation of dependencies is useful both mathematically and for verification.

1.2 Prior work and the scope of the result

Mendès France and Tenenbaum [3, eqs. (3.6), (6.1)] define the normalized family

$$F_\alpha^{\mathrm{MFT}}(n) = \sum_{i<\tau(n)} \left(\frac{\log(d_{i+1}/d_i)}{\log n} \right)^\alpha.$$

Thus $F_2^{\mathrm{MFT}}(n) = F_2(n)/\log^2 n$ exactly. Their normal-behavior results must be distinguished from the arithmetic mean and the explicit bound considered here. We make no novelty claim for the definition of the normalized moment.

A different consecutive-divisor statistic studied by de Koninck and Ivić [1, eq. (1.6), Theorem 2] is

$$H_{\mathrm{DKI}}(n) = \sum_{i=2}^{\tau(n)} \frac{1}{d_i - d_{i-1}}.$$

It concerns reciprocal additive gaps. Its mean-value theorem is not a theorem about the coefficient c_2 in (1.6).

Haddad and Koukoulopoulos [2] provide the quantitative coupling used in Section 6. The existence and identification of the mean coefficient here are applications of that external result and an elementary Lipschitz estimate. The contribution claimed in this manuscript is the explicit certificate-based upper bound for the resulting random-set expectation. A definitive priority claim is intentionally not made: absence of an equivalent earlier bound has not been established by the present manuscript preparation.

1.3 Why compare with $1/4$?

Let $d_3(n) = \#\{(a,b,c) \in \mathbb{N}^3 : abc = n\}$. Then

$$\sum_{n\geq 1} \frac{d_3(n)}{n^s} = \zeta(s)^3 \quad (\Re s > 1), \qquad \sum_{n\leq x} d_3(n) = \tfrac{1}{2} x \log^2 x + O(x \log x).$$

For completeness, the last leading term follows by writing the sum as $\sum_{ab\leq x} \lfloor x/(ab) \rfloor$: the floor errors contribute $O(x \log x)$, while $\sum_{ab\leq x} 1/(ab) = \frac{1}{2}\log^2 x + O(\log x)$. Consequently, comparison with one half of the summatory function of d_3 suggests the coefficient $1/4$. Such a comparison arose in exploratory computations recorded in the research history. It is not presented as a previously published conjecture, nor as the prediction of a specified independent-prime model. The proof of Theorem 1.1 does not depend on this motivation.

2 Subset-sum geometry and GEM self-similarity

2.1 Compactness, gaps, and stability

Let

$$\Delta_\infty = \{a \in \ell^1 : a_j \geq 0,\ \sum_j a_j = 1\}.$$

The map from $\{0,1\}^{\mathbb{N}}$ to $\mathbb{R}$ in (1.2) is continuous: after fixing the first m coordinates, any two images differ by at most $\sum_{j>m} a_j$. Compactness of the product space therefore implies compactness of $K(a)$. Equivalently, $K(a)$ is the closure of the set of finite subsums. Distinct subsums may coincide; we always work with a set rather than with multiplicities of representations.

The complement of a closed set in $[0,1]$ containing the endpoints is a countable union of disjoint open intervals. In particular,

$$0 \leq H(K) \leq \left(\sum_{I \in \mathcal{G}(K)} |I| \right)^2 \leq 1. \tag{2.1}$$

No assertion that K has measure zero is needed for this inequality or for the proof of the upper bound.

Lemma 2.1 (Distance identity and Lipschitz estimate)**.** *For compact $K, L \subset [0,1]$ containing 0 and 1,*

$$H(K) = 4\int_0^1 \operatorname{dist}(t,K)\,dt, \qquad |H(K) - H(L)| \leq 4 d_H(K,L). \tag{2.2}$$

Moreover, for $a, b \in \Delta_\infty$,

$$d_H(K(a),K(b)) \leq \|a-b\|_1, \qquad |H(K(a)) - H(K(b))| \leq 4\|a-b\|_1. \tag{2.3}$$

Proof. On a gap $(u, u+g)$, the distance to K equals $\min(t-u, u+g-t)$, whose integral is $g^2/4$. On K the distance is zero. Summing these nonnegative integrals proves the first identity, also for infinitely many gaps and for sets of positive measure. The distance functions differ pointwise by at most $d_H(K, L)$, which gives the next inequality. For each binary sequence ε, pair its subsum in $K(a)$ with the subsum having the same binary coefficients in $K(b)$. Their distance is at most $\sum_j |a_j - b_j|$. Apply the argument in both directions and then use (2.2). □

Two additional elementary facts will be used. If $K \subset L$ and both sets contain the endpoints, then $H(L) \le H(K)$ by (2.2). If a set in $[0,1]$ is translated and scaled into an interval $[b, b+r]$, its gap-square sum, computed relative to that interval, is multiplied by r^2.

2.2 Stick breaking and the common tail

Let $V_1, V_2, \dots$ be independent uniform random variables on $[0,1]$. Define

$$P_j = V_j \prod_{h<j} (1 - V_h), \qquad R_m = \prod_{j=1}^{m} (1 - V_j). \tag{2.4}$$

This is the GEM(1) representation; its decreasing rearrangement has law PD(1), with the convention in [2, pp. 1566–1567]. Since K is invariant under a rearrangement of its summable weights, we may use the GEM order throughout.

Almost surely $R_m \to 0$ and $\sum_j P_j = 1$. For example, $-\log(1 - V_j)$ are independent with mean 1, so the strong law applies. For fixed m, put

$$T_m = \left\{ \sum_{j=1}^{m} \varepsilon_j P_j : \varepsilon_j \in \{0,1\} \right\}.$$

Let K'_m be the subset-sum set formed from the normalized tail $(P_{m+j}/R_m)_{j \ge 1}$. Except on a null event where $R_m = 0$, this tail is a fresh GEM sequence. Consequently,

$$K = \bigcup_{s \in T_m} (s + R_m K'_m), \qquad K'_m \perp (V_1, \dots, V_m), \qquad \mathbb{E} H(K'_m) = c_2. \tag{2.5}$$

All translates on the right use *the same* random set K'_m. They are not independent copies.

3 Path bounds with residual bridges

Lemma 3.1 (Arbitrary-path bound)**.** *Let $K \subset [0,1]$ be compact with $0, 1 \in K$, and let $z_0 = 0, z_1, \dots, z_q = 1$ be a finite sequence of points of K. Then*

$$H(K) \le \sum_{j=1}^{q} |z_j - z_{j-1}|^2. \tag{3.1}$$

The sequence need not be increasing or have distinct vertices.

Proof. Fix a gap $I = (a, b)$ of K. Each vertex lies either at or to the left of a, or at or to the right of b. The sequence begins on the left and ends on the right. Hence some consecutive pair spans $[a, b]$. Assign I to the first such pair. For a fixed edge of length L, all assigned gaps are disjoint intervals inside that edge. If their lengths are g_i, then $\sum_i g_i^2 \le (\sum_i g_i)^2 \le L^2$. Summing over the finitely many edges gives the assertion. Countably many gaps cause no difficulty because all contributions are nonnegative. □

For the GEM representation and a fixed m, use vertex labels $\varepsilon \in \{0,1\}^{m+1}$ with

$$z_\varepsilon = \sum_{j=1}^{m} \varepsilon_j P_j + \varepsilon_{m+1} R_m. \tag{3.2}$$

Every such vertex belongs to K: the last coefficient selects either none or all of the tail. The labels can represent coincident points; this does not affect Lemma 3.1. A path is a finite list of these labels, starting with the all-zero label and ending with the all-one label. An edge is called a *residual bridge* when only its last bit changes. Its geometric interval is then $[s, s + R_m]$ for some $s \in T_m$.

Lemma 3.2 (Residual-bridge bound)**.** *For a fixed prefix $v = (V_1, \dots, V_m)$ and a path π, let $a_\pi(v)$ be the sum of squared lengths of its non-bridge edges and let $n_R(\pi)$ be its number of residual bridges. Then, for each admissible tail set,*

$$H(K) \le a_\pi(v) + n_R(\pi) R_m(v)^2 H(K'_m). \tag{3.3}$$

If the path is a measurable function of the prefix alone, then

$$\mathbb{E}[H(K) \mid V_1, \dots, V_m] \le a_\pi(v) + n_R(\pi) R_m(v)^2 c_2. \tag{3.4}$$

Proof. Use the assignment of gaps to edges in Lemma 3.1. A non-bridge edge is treated there. For a bridge interval $[s, s + R_m]$, the set $s + R_m K'_m$ is contained in K and includes both endpoints of the interval. Each assigned gap of K is therefore contained in a gap of that scaled tail set.

Inside one scaled tail gap of length $R_m g'$, the assigned gaps are disjoint and have total length at most $R_m g'$. Their squared lengths sum to at most $R_m^2 (g')^2$. Sum over the tail gaps, and then over the bridges, to obtain (3.3). Overlap between bridge intervals is harmless: each actual gap was assigned to just one edge. Repeated tail terms on the right are upper bounds, not assertions of independence.

Finally, the path and R_m depend only on the prefix, while K'_m is independent of that prefix and has the same gap-energy expectation as K. Conditional expectation proves (3.4). □

Remark 3.3. The path may be chosen by a heuristic, for instance by ordering vertex values at the center of a parameter cell. Its order can change elsewhere in the cell without invalidating (3.3). This is what makes a fixed combinatorial path useful on an entire dyadic cell.

Lemma 3.4 (Separated first-step contribution)**.** *On $V_1 \ge 1/2$, with K'_1 the normalized tail after the first break,*

$$H(K) = (2V_1 - 1)^2 + 2(1 - V_1)^2 H(K'_1). \tag{3.5}$$

Accordingly,

$$\mathbb{E}[H(K) \mathbf{1}_{\{V_1 \ge 1/2\}}] = \frac{1}{6} + \frac{1}{12} c_2. \tag{3.6}$$

Proof. The two copies $(1 - V_1)K'_1$ and $V_1 + (1 - V_1)K'_1$ lie in disjoint intervals, apart from a possible common endpoint when $V_1 = 1/2$. The central gap has length $2V_1 - 1$. The other gaps are those of the two scaled copies, giving (3.5). The tail is independent of V_1, and

$$\int_{1/2}^{1} (2v - 1)^2 \, dv = \frac{1}{6}, \qquad 2 \int_{1/2}^{1} (1 - v)^2 \, dv = \frac{1}{12}.$$

□

4 Finite dyadic certificates and exact integration

4.1 What the certificate specifies

We now fix $m = 6$. The prefix space is $[0,1]^6$ with uniform product measure. One dyadic split divides it into $V_1 \le 1/2$ and $V_1 \ge 1/2$. The second region is treated by Lemma 3.4, with no restrictions on $V_2, \dots, V_6$. The first region is partitioned further, and each of its leaves carries a path using the $2^7 = 128$ labels in (3.2).

The certificate is a preorder encoding of a finite binary tree. Its header is `C2PATH_V2 6`. The records are:

Record	Meaning
`S i`	Split coordinate i at its midpoint; the left subtree is followed by the right subtree. Coordinates are numbered $0, \dots, 5$.
`L q p0 ...`	A path with q vertex labels. Labels lie between 0 and 127; the first is 0 and the last is 127.
`E`	An analytic leaf on which $V_1 \ge 1/2$ and the other five coordinates remain unrestricted.

The lower six bits encode $P_1, \dots, P_6$; bit 6 encodes R_6. Thus a bridge is recognized by the bitwise exclusive-or condition $p \operatorname{xor} p' = 64$.

Starting with the root cube, a valid split replaces a box by its two children. Induction gives coverage and disjoint interiors; boundaries have measure zero. A complete parse, with exactly two children for each split and no unconsumed non-whitespace data, verifies the combinatorial partition. A separate exact volume check is a useful consistency check, but volume equality alone is not a proof of disjointness.

4.2 Exact cell moments

A leaf is a box

$$C = \prod_{i=1}^{6} [k_i 2^{-d_i}, (k_i+1)2^{-d_i}], \qquad \operatorname{vol}(C) = 2^{-s_C}, \qquad s_C = \sum_i d_i.$$

Conditionally on C, its coordinates are independent uniform variables on their respective intervals. Writing $n_i = 2^{d_i}$,

$$\mathbb{E}_C V_i = \frac{2k_i + 1}{2n_i}, \qquad \mathbb{E}_C V_i^2 = \frac{3k_i^2 + 3k_i + 1}{3n_i^2}. \tag{4.1}$$

Every P_j and R_6 is a multilinear polynomial in the prefix coordinates. Thus any squared vertex difference has degree at most two in each coordinate. Its cell integral is obtained exactly from (4.1); no numerical quadrature is needed.

For the path π_C of a non-analytic leaf, let

$$A_C = \operatorname{vol}(C)\mathbb{E}_C a_{\pi_C}(V), \qquad B_C = \operatorname{vol}(C) n_R(\pi_C)\mathbb{E}_C R_6^2.$$

Define

$$A = \frac{1}{6} + \sum_{C \text{ path leaf}} A_C, \qquad B = \frac{1}{12} + \sum_{C \text{ path leaf}} B_C. \tag{4.2}$$

Lemmas 3.2 and 3.4, integrated over the partition, imply

$$c_2 \le A + Bc_2. \tag{4.3}$$

The tail has not been replaced by an approximate finite set. Its contribution remains in the term Bc_2, so an additional truncation error such as 2^{-6} must not be added to, or silently omitted from, this argument.

4.3 Two algebraic evaluation routes

For a difference of labels, put $z_i \in \{-1, 0, 1\}$ for the difference of bit i, and set

$$X_7 = z_7, \qquad X_i = z_i V_i + (1 - V_i) X_{i+1} \quad (1 \le i \le 6).$$

Then X_1 is the vertex difference. If $\mu_i = \mathbb{E}_C X_i$ and $\eta_i = \mathbb{E}_C X_i^2$, conditional independence gives

$$\mu_i = z_i \mathbb{E}_C V_i + (1 - \mathbb{E}_C V_i)\mu_{i+1}, \tag{4.4}$$

$$\eta_i = z_i^2 \mathbb{E}_C V_i^2 + 2 z_i \mathbb{E}_C[V_i(1 - V_i)]\mu_{i+1} + \mathbb{E}_C[(1 - V_i)^2]\eta_{i+1}. \tag{4.5}$$

The terminal values are $\mu_7 = z_7$ and $\eta_7 = z_7^2$.

Alternatively, form the matrix of conditional moments of $W = (P_1, \dots, P_6, R_6)$. The squared-length expectation is $z^{\mathsf{T}} \mathbb{E}_C[W W^{\mathsf{T}}] z$. Each entry factors into one-variable moments. These routes provide algebraically distinct checks. They still share the same geometric lemmas, certificate format, and elementary moment identities; agreement is not a claim of absolute independence of all possible errors.

5 Certified arithmetic and completion of the proof

Proposition 5.1 (Exact finite-certificate evaluation)**.** *The public certificate [5], originating in the historical record [4] and identified by the compressed-file SHA-256 digest below, has the statistics in Table 1. Its exact integration according to* (4.2) *gives*

$$A = \frac{5565525599521087701564905}{244807478471962407878000 64}, \tag{5.1}$$

$$B = \frac{1055355009230790003453101}{122403739235981203939000 32}. \tag{5.2}$$

The certificate file is `c2_certificate.txt.gz`, with digest

```
b0793e32865fcc91b479239e8248d6827f70
d40b0e74ed71d78e0b246aa4fc09
```

The line break above is typographical; the digest is one 64-character hexadecimal string. Appendices A–B give the exact filename, the single-line digest, the record grammar, and the verification commands. A digest identifies the intended bytes; it does not prove the inequalities.

Verification procedure for Proposition 5.1. Parse the entire certificate, retaining each dyadic box. Reject an invalid split, a missing child, an invalid vertex, a path with incorrect endpoints, an invalid analytic leaf, or trailing data. On every path leaf evaluate the ordinary edges using (4.4)–(4.5) or the moment matrix, and evaluate the bridge contribution as $n_R \mathbb{E}_C R_6^2$.

Certificate statistic	Replayed value
Total leaves	1 259 080
Path leaves	1 259 079
Analytic leaves	1
Path edges, including residual bridges	77 187 941
Residual bridges	22 185 035
Maximum total subdivision depth	26
Maximum coordinate depths	$(10, 9, 8, 7, 7, 6)$
Total volume	1

Table 1: Finite-certificate statistics reproduced by the November 2026 standalone C++17 replay. The separate GCC and Clang runs use the same verifier algorithm.

Multiply by the box volume. On the analytic region integrate Lemma 3.4. Add all rational contributions and reduce the results. The two historical verifier outputs included in [5] record (5.1)–(5.2) exactly.

This proposition has also been checked from the fixed compressed certificate with the self-contained C++17 verifier in the November 21 replay package, and rechecked on November 22 in an independent packaging audit using GCC and Clang. The runner checks the compressed input digest, partition grammar, cell geometry, integer moment integrals, complete record consumption and the rational values A, B. It also rejects ten corrupted test inputs. The compiler comparison does not amount to two independent mathematical algorithms. The historical Python verifier result files are archived comparator data, not evidence that these Python sources were newly executed. On November 28, a further replay of the tagged v1.0.0 archive recovered the same exact values and rejected all ten malformed examples. Appendices D–F record the exact commands and boundaries of these checks. □

Proof of Theorem 1.1. By (2.1), c_2 is a finite, nonnegative real number. The valid partition and path bounds give (4.3). The recorded rational coefficient satisfies $0 \le B < 1$, so rearrangement yields $c_2 \le A/(1-B)$. Substituting (5.1)–(5.2) gives exactly the rational number U in (1.5). All comparisons are comparisons of integers after multiplication by positive denominators. In particular,

$$\frac{1}{4} - U = \frac{53967715325154987317121}{44740075657469321561787724} > 0. \tag{5.3}$$

This proves the asserted strict separation. □

There is also a short rounded certificate for the final comparison:

$$A < \frac{227343}{10^6}, \qquad B < \frac{8622}{10^5}, \qquad c_2 < \frac{227343}{913780} < \frac{249}{1000} < \frac{1}{4}.$$

The last comparison with $1/4$ reduces to $909372 < 913780$.

Remark 5.2 (Finite iteration). Define $u_0 = 1$ and $u_{j+1} = A + Bu_j$. Induction using (4.3) gives $c_2 \le u_j$. In particular,

$$u_4 = A(1 + B + B^2 + B^3) + B^4 < \frac{249}{1000}.$$

This is a finite consequence of the same inequality, not an independent proof or a replacement for self-similarity. The rearrangement above is already legitimate because c_2 is finite and $B < 1$.

6 Transfer to the arithmetic mean

For $n \geq 2$, list its prime factors with multiplicity as $Q_1(n) \geq Q_2(n) \geq \cdots$, padding by ones. Define

$$a_j(n) = \frac{\log Q_j(n)}{\log n}.$$

Then $a(n) \in \Delta_\infty$. Choosing a submultiset of the prime factors is exactly choosing a divisor, and consequently

$$K(a(n)) = \{\log d / \log n : d \mid n\}, \qquad H(K(a(n))) = \frac{F_2(n)}{\log^2 n}. \tag{6.1}$$

Multiplicity is important in the first identity. Multiple selections can give the same divisor, but the resulting set is still exactly the set of normalized logarithms of all divisors.

We use the following precise input from [2, eq. (1.5), p. 1568]. If N_x is uniform on $\{1, \ldots, \lfloor x \rfloor\}$ and ψ is Lipschitz on Δ_∞ for the ℓ^1 metric, then

$$\mathbb{E}\Big[\mathbf{1}_{\{N_x \geq 2\}} \psi(a(N_x))\Big] = \mathbb{E}\psi(a) + O_\psi(1/\log x), \qquad a \sim \mathrm{PD}(1). \tag{6.2}$$

Here the expression on the left is defined to be zero on $N_x = 1$. In particular, the denominator inside $a(N_x)$ is $\log N_x$, not $\log x$.

By Lemma 2.1, $\psi(a) = H(K(a))$ is bounded and 4-Lipschitz. Substituting (6.1) in (6.2) and multiplying by $\lfloor x \rfloor$ gives

$$M(x) := \sum_{2 \leq n \leq x} \frac{F_2(n)}{\log^2 n} = c_2 x + O(x/\log x). \tag{6.3}$$

The floor changes the main term by $O(1)$.

Proof of Corollary 1.2. Partial summation gives

$$S_2(x) = M(x)\log^2 x - \int_2^x M(y)\frac{2\log y}{y}\, dy.$$

By (6.3), the first term is $c_2 x \log^2 x + O(x \log x)$. The integral of the main term is $2c_2 \int_2^x \log y\, dy = O(x \log x)$, while the integral of the error is $O(x)$. This proves (1.6). The strict inequality in Theorem 1.1 excludes the proposed coefficient $1/4$. □

Remark 6.1. The $O(x \log x)$ error in (1.6) does not identify a second coefficient. In particular, the explicit integral of the main term in partial summation must not be interpreted as proving that the second coefficient is $-2c_2$: the endpoint error in $M(x)\log^2 x$ has the same order. No second-order result from any separate research branch is used here.

7 A real-axis Dirichlet-series consequence

Because $0 \leq F_2(n) \leq \log^2 n$, the series

$$D_2(s) = \sum_{n \geq 1} F_2(n) n^{-s}$$

is absolutely convergent for $\Re s > 1$. For real $t > 0$, partial summation and (1.6) yield

$$\begin{aligned} D_2(1+t) &= (1+t)\int_1^\infty S_2(x) x^{-2-t}\, dx \\ &= \frac{2c_2}{t^3} + O(t^{-2}) \qquad (t \downarrow 0), \end{aligned}$$

since $\int_1^\infty (\log x)^k x^{-1-t}\,dx = k!t^{-k-1}$ for $k = 1, 2$. The standard leading estimate $\zeta(1+t) = t^{-1} + O(1)$ follows already by integral comparison. Thus

$$D_2(1+t) - \frac{1}{2}\zeta(1+t)^3 = \left(2c_2 - \frac{1}{2}\right) t^{-3} + O(t^{-2}). \tag{7.1}$$

By (5.3),

$$2c_2 - \frac{1}{2} \le -\frac{5396771532515498731712 1}{22370037828734660780893862} < 0.$$

The difference in (7.1) therefore tends to $-\infty$ along the real axis. Replacing $1/2$ by $2c_2$ does cancel its leading t^{-3} term:

$$D_2(1+t) - 2c_2\zeta(1+t)^3 = O(t^{-2}).$$

These are one-sided boundary asymptotics. They do not establish meromorphic continuation through $s = 1$, the existence of a pole of D_2, or any relation to the nontrivial zeros of ζ.

8 Reproducibility, limitations, and disclosure

8.1 Separation of search and verification

The candidate paths were selected using floating-point heuristics in the original research process. Such calculations are not the proof. For each valid path, Lemma 3.2 holds throughout its cell, regardless of how the path was selected. Correctness depends on the valid finite partition, the path inequalities, and the exact integrals of the stored paths.

The original record documents two algebraically distinct Python verification routes with exact rational agreement. The current self-contained artifact includes a distinct C++17 integer verifier, the complete compressed certificate, the original Python result JSONs (but not their source programs), an offline replay harness, negative tests and a recorded successful run. The full replay also passed an independent packaging audit with two compilers. Agreement between model-generated reviews is not a mathematical premise; the reproducible program and its validated input are the computational evidence.

The October 2026 scalar-check script confirms exact rational identities and small moment examples but is *not* a full-certificate verifier. By contrast, the November 2026 C++ replay evaluated the entire certificate and recovered the exact coefficient pair. The original outer ZIP and Python verifier source files have not been included in the November replay bundle; their historical digests and outputs are maintained separately.

8.2 Data and code availability

The complete computational artifact is publicly available in release `v1.0.0` of [5]:

`https://github.com/wnjhn5svw2-max/divisor-gap-c2-certificate/releases/tag/v1.0.0`.

The tag was checked on November 28, 2026, against commit

`bd1b404b6f5d83e8d245f2653a1e8382e7c619b4`

For exact revision selection, use the commit-specific tree or its source ZIP, rather than the moving default branch. The release is versioned but is not designated immutable by GitHub;

the commit identifier and the certificate SHA-256 in Section 5 identify the checked content. No archive DOI has been assigned. A local replay of the supplied tagged ZIP matched the public tree's thirteen Git blob identifiers and the twelve entries of `SHA256SUMS`.

After extraction, on x86-64 Linux with Python 3.11 or later and a C++17 compiler supporting signed 128-bit integers, enter the extracted repository directory and run

```
sha256sum -c SHA256SUMS
python3 -O run_replay.py > replay.local.json
```

The public repository specifies MIT for original code and associated documentation and CC0 1.0 for the certificate data and logs, to the extent that the stated permissions can be granted. The technical appendices reproduce the version 0.3 technical supplement, providing the certificate grammar and additional reproducibility details. The intended publication license for this manuscript, including the appendices, is CC BY 4.0, subject to the author's final license selection at submission.

8.3 Limits of the claim

We do not determine a closed form for c_2, its arithmetic nature, or a high-precision certified value near the exploratory estimate 0.24492. The optional large-gap quadrature and its lower bounds in [4] are omitted from this paper; they are not needed for the upper bound. We do not assert a new general coupling theorem or a new definition of logarithmic divisor-gap moments. We also do not infer a clustering law or an independent-prime model from the comparison with 1/4.

The standalone C++ verifier uses explicit mandatory checks, including when compiled with `-DNDEBUG`, and its Python runner was executed using `python3 -O`. A reader can reproduce the fixed certificate without the historical outer archive. The public computational release is distinct from the publication of this manuscript. The present version is a submission candidate, not an accepted journal article or a formal proof-assistant certification.

8.4 AI assistance and human review status

This project originated in a human-directed exploration of integer factorizations using generative AI. AI systems were used in proposing research directions, searching for literature, developing proof arguments, generating candidate certificates and verification code, reviewing those arguments and programs, and preparing the present English draft. The surviving project record names ChatGPT/Astra, Gemini, and Claude at different stages; exact historical backend identifiers have not all been independently authenticated.

No AI system is listed as an author; the human author is identified publicly as H. Matsunami. The author directed the research questions and the release process, but does not claim to have independently verified every mathematical and implementation detail without AI assistance. No independent human specialist review or journal peer review has been completed. The proof is presented for external scrutiny, with its exact certificate and reproducible verifier. The submission provenance statement and the technical appendices give additional context on AI assistance and the scope of computational checks.

References

[1] J.-M. de Koninck and A. Ivić, *On the distance between consecutive divisors of an integer*, Canadian Mathematical Bulletin **29** (1986), no. 2, 208–217. doi:10.4153/CMB-1986-034-7.

[2] T. Haddad and D. Koukoulopoulos, *On Arratia's coupling and the Dirichlet law for the factors of a random integer*, Journal de l'École polytechnique—Mathématiques **12** (2025), 1565–1604. doi:10.5802/jep.317.

[3] M. Mendès France and G. Tenenbaum, *Systèmes de points, diviseurs et structure fractale*, Bulletin de la Société Mathématique de France **121** (1993), no. 2, 197–225. doi:10.24033/bsmf.2207.

[4] *Rigorous certification of the divisor-gap mean coefficient* c_2, Japanese project report and accompanying computational records, version `C2-LITERATURE-v2-20260915`, September 15, 2026. Report filename: `c2_rigorous_certification_ja_v2_20260915.md`. This is the supplied research record, not an independent publication. The historical Japanese report is not part of the public computational release. The executable certificate and comparator data are available in [5].

[5] H. Matsunami, *Exact certificate for a quadratic logarithmic divisor-gap upper bound*, computational artifact, version 1.0.0 (November 28, 2026), GitHub. Release: `https://github.com/wnjhn5svw2-max/divisor-gap-c2-certificate/releases/tag/v1.0.0`. Commit: `bd1b404b6f5d83e8d245f2653a1e8382e7c619b4`.

Technical appendices: Certificate specification and reproducibility

Scope. These technical appendices describe the finite certificate underlying the bound

$$c_2 \le \frac{5565525599521087701564905}{22370037828734660780893862} < \frac{1}{4}.$$

It distinguishes the original certification record from the full-certificate replay completed on November 21–22 and November 28, 2026. The released computational artifact is the compressed certificate together with a self-contained C++17 integer verifier and replay harness; the original Python verifier source programs are not included in that bundle.

A Artifact identity and present availability

The baseline is `C2-LITERATURE-v2-20260915`. The Japanese report was recovered during preparation of version 0.1 and its bytes were verified against the expected SHA-256:

```
c2_rigorous_certification_ja_v2_20260915.md
377f443d486d8be7bf2d627e25b105e62b2282a96b4c381136ce83da6d3129e0
```

The original outer ZIP is identified by its historical digest. The compressed certificate has been retrieved from the November 2026 standalone replay package, and its SHA-256 was independently recomputed during the subsequent prepublication audit.

```
c2_certification_code_v2_20260915.zip
size: 37480425 bytes
sha256:
eb5849c28e114602ebd64e8207e4baf67bb926e7e89fb8d839a9bbc0bc99e5ec

c2_certificate.txt.gz
sha256:
b0793e32865fcc91b479239e8248d6827f70d40b0e74ed71d78e0b246aa4fc09
```

The historical outer ZIP was not required for the November replay: the fixed compressed certificate was supplied byte-for-byte in `data/c2_certificate.txt.gz` inside `c2_divisor_gap_review_20261121.zip`. The latter ZIP has SHA-256 digest

```
fc887cdc66e19e27afac585503e4cd7a7d9e52e1a33a7f337f45575fa5528e40
```

The original pair of Python verifier programs are not included. The two historical Python JSON outputs are present for exact-value comparison, not as newly run verifier evidence. The computational artifact was released publicly on November 28, 2026. The manuscript and artifact versions are separate.

- Artifact: H. Matsunami, *Exact certificate for a quadratic logarithmic divisor-gap upper bound*, v1.0.0.
- Release page: `https://github.com/wnjhn5svw2-max/divisor-gap-c2-certificate/releases/tag/v1.0.0`.
- Commit: `bd1b404b6f5d83e8d245f2653a1e8382e7c619b4`.
- Commit-specific source ZIP. The release tag is not marked immutable; use the commit and the compressed-certificate SHA-256 to identify the checked bytes. No DOI is assigned.

The repository's MIT license applies to original code and associated documentation, and CC0 1.0 applies to certificate data and logs, subject to the rights stated in the repository's license notice. The intended license for the paper and its appendices is CC BY 4.0, subject to the author's final submission selection.

B Certificate grammar and partition semantics

The compressed certificate expands to ASCII text. Its first line is

```
C2PATH_V2 6
```

The remaining records describe a finite tree in preorder:

```
S i
L q p0 p1 ... p(q-1)
E
```

A split record `S i` replaces the current box by its two midpoint children in coordinate i, numbered from 0 to 5. The next records recursively encode the left child and then the right child. A path record has $q \geq 2$ labels, all integers between 0 and 127. Its first label is 0 and its last is 127. Repeated labels and zero-length edges need not be rejected for mathematical soundness, though they affect raw edge counts. The low six bits select the first six GEM weights, and the high bit selects the entire tail mass. The test

```
p ^ p_next == 64
```

identifies residual bridges.

An `E` leaf is valid only if its first coordinate interval lies in $[1/2, 1]$ and all other coordinate intervals are $[0, 1]$. The baseline uses precisely one such leaf: the right child of the first split. Restrictions on later GEM variables would invalidate a direct replacement of the first-step tail expectation by the unconditional constant c_2.

For each coordinate retain $d_i \geq 0$ and $0 \leq k_i < 2^{d_i}$, representing $[k_i/2^{d_i}, (k_i+1)/2^{d_i}]$. A split replaces (d_i, k_i) by $(d_i+1, 2k_i)$ or $(d_i+1, 2k_i+1)$. Induction from the root proves that all leaves form a partition with disjoint interiors. The parser must also reject a truncated subtree and extra non-whitespace data after the root is complete. Checking the sum of leaf volumes is a secondary consistency test, not a standalone disjointness argument.

The full-certificate replay verified 1,259,080 leaves, including one analytic leaf, 77,187,941 path edges, 22,185,035 residual bridges, and maximum total depth 26. These quantities are recomputed from the bytes and checked against the expected statistics by the executable verifier.

C Exact integration without a floating-point step

Let C be a dyadic leaf and let $\mathbb{E}_C$ denote expectation under its conditional uniform product measure. Write

$$\mu_i = \mathbb{E}_C V_i, \quad \nu_i = \mathbb{E}_C V_i^2, \quad \rho_i = 1 - \mu_i, \quad \sigma_i = 1 - 2\mu_i + \nu_i, \quad \chi_i = \mu_i - \nu_i.$$

For $n_i = 2^{d_i}$,

$$\mu_i = \frac{2k_i + 1}{2n_i}, \qquad \nu_i = \frac{3k_i^2 + 3k_i + 1}{3n_i^2}.$$

These formulas follow directly by integrating v and v^2 over the interval and dividing by its length.

C.1 Backward moments

For the difference vector $z \in \{-1, 0, 1\}^7$ of an edge, define $X_7 = z_7$ and $X_i = z_i V_i + (1 - V_i) X_{i+1}$. Initialize $M_7 = z_7$, $Q_7 = z_7^2$, and update

$$M_i = z_i \mu_i + \rho_i M_{i+1},$$
$$Q_i = z_i^2 \nu_i + 2 z_i \chi_i M_{i+1} + \sigma_i Q_{i+1}.$$

Then Q_1 is the exact conditional second moment of the edge length.

C.2 Moment matrix

Put $W = (P_1, \ldots, P_6, R_6)$ and $G = \mathbb{E}_C[WW^{\mathsf{T}}]$. With empty products equal to 1,

$$G_{ii} = \nu_i \prod_{h<i} \sigma_h \qquad (1 \le i \le 6),$$
$$G_{ij} = \chi_i \mu_j \prod_{h<i} \sigma_h \prod_{i<h<j} \rho_h \qquad (1 \le i < j \le 6),$$
$$G_{i7} = \chi_i \prod_{h<i} \sigma_h \prod_{h>i} \rho_h \qquad (1 \le i \le 6),$$
$$G_{77} = \prod_{h=1}^{6} \sigma_h.$$

The matrix is symmetric, and the edge second moment is $z^{\mathsf{T}} G z$. For a residual bridge, do not use G_{77} as an ordinary squared-length contribution: its coefficient is assigned instead to the factor multiplying c_2.

C.3 Integer scaling and volumes

A common denominator for all one-coordinate moments is $D_i = 6n_i^2$. The corresponding integer numerators are

$$V_i^* = 3n_i(2k_i + 1), \qquad V_i^{**} = 2(3k_i^2 + 3k_i + 1),$$

with $D_i - V_i^*$, $D_i - 2V_i^* + V_i^{**}$, and $V_i^* - V_i^{**}$ for the other required moments. A conditional edge second moment has denominator dividing $\prod_i D_i = 6^6 2^{2s_C}$, where $s_C = \sum_i d_i$. Multiplication by the cell volume 2^{-s_C} changes this to $6^6 2^{3s_C}$. Thus aggregation at maximum depth D can use integer factors $2^{3(D - s_C)}$. The factor 3 comes from two powers in the moment denominator and one power from the volume.

This derives the scaling scheme; it does not replace review of the actual implementation. In a fixed-width language the maximum sizes of *intermediate* integers must also be bounded, or arbitrary-precision arithmetic must be used.

C.4 Analytic leaf and aggregate coefficients

For a first-coordinate interval $[\ell, r] \subset [1/2, 1]$ with all other coordinates unrestricted, the integrated coefficients are

$$A_E = \frac{4}{3}(r^3 - \ell^3) - 2(r^2 - \ell^2) + (r - \ell),$$
$$B_E = 2(r - \ell) - 2(r^2 - \ell^2) + \frac{2}{3}(r^3 - \ell^3).$$

For $[\ell, r] = [1/2, 1]$ they equal $1/6$ and $1/12$. On other leaves, add volume times the ordinary-edge second moments to A, and volume times $n_R \mathbb{E}_C R_6^2$ to B. The aggregate inequality is $c_2 \le A + Bc_2$.

D Replaying the complete certificate with the available artifact

Download the source archive linked in Appendix A, extract it and enter the extracted repository directory (the directory containing the README and checksum manifest). The self-contained replay needs Python 3.11 or later and a C++17 compiler supporting `__int128`. On x86-64 Linux, run:

```
sha256sum -c SHA256SUMS
python3 -O run_replay.py > replay.local.json
```

The runner compiles `src/verifier.cc`, verifies the three fixed input digests, checks the certificate under exact integer arithmetic, compares the resulting rational values with the archived JSON comparators, and rejects ten deliberately malformed test inputs. Its default build uses `-O3 -DNDEBUG`; validation uses explicit mandatory checks that remain enabled. The full input is 37,441,710 compressed bytes (260,691,493 uncompressed bytes). The runner needs approximately 300 MB of temporary disk space. No network access is required. The optional large-gap quadrature is unrelated to the upper-bound replay.

The original historical Python sources `verify_recurrence.py` and `verify_matrix.py` were not distributed in this standalone package and were not re-executed in the November audit. It is important not to substitute the presence of historical `recurrence_final.json` and `matrix_final.json` for a claim of re-execution.

D.1 Negative tests and compiler cross-check

The November 2026 runner was executed in optimized Python mode and rebuilt the C++ verifier with `-DNDEBUG`. It rejected each of ten invalid inputs: wrong header, invalid split, truncated tree, out-of-range vertex, invalid endpoints, analytic cells in the wrong region (two cases), trailing node, noncanonical integer, and overlong line. A separate prepublication audit repeated the full replay under GCC 14.2.0 and Clang 17.0.0; both gave the same exact values and rejected all ten malformed inputs. This is a two-compiler cross-check of *one* C++ mathematical algorithm, not two independent derivations.

E Recorded checks and final release replay

The earlier internal script `checks/check_record.py` was run both normally and with `python -O`. It uses explicit exceptions for every test and no removable assertions.

The runs verified the source-report hash; $0 \le B < 1$; $U = A/(1 - B)$; all rational upper-bound comparisons; the exact quarter-margin; the equivalence of the iterated and closed-form fourth bound; and the analytic-cell integrals $1/6$ and $1/12$. For all 511 dyadic intervals of depths 0 through 8, the first and second moment formulas were compared with antiderivative evaluations. For 120 deterministic-seeded toy edges with $m = 1, 2, 3$, three routes agreed as exact rational numbers: backward moments, a product-factorized moment matrix, and direct polynomial expansion followed by monomial integration.

Those earlier checks validate small formulas, not the full tree. The later C++ replay, unlike the October scalar-only script, parsed the complete fixed tree and recomputed A, B, U exactly. The successful baseline replay is stored as `logs/replay.json` and is reproducible by the commands above. Its recorded execution was November 21, 2026, 08:47:29–08:47:32 JST. A separate prepublication audit was performed on November 22. The exact input and historical-result SHA-256 values matched the original record.

On November 28, the tagged v1.0.0 source ZIP supplied by the author was checked again: all 12 SHA-256 manifest entries matched, and all 13 Git blob identifiers computed from the archive matched the public commit tree retrieved from GitHub. The replay rebuilt the C++ verifier with GCC and ran the full tree with the optimized Python harness, obtaining the same exact A, B, U and fourth-iteration bound and rejecting 10/10 malformed inputs. This check establishes the correspondence between the supplied archive and the public tree; an unauthenticated binary download was not performed. The tag and public visibility were confirmed separately. These tests do not re-prove the analytic lemmas or establish novelty.

F Evidence categories and remaining gates

Item	Present status
Japanese v2 report	Raw bytes recovered; expected digest matched.
Path and bridge arguments	Reconstructed explicitly in the English manuscript.
HK equation (1.5)	Publisher's parsed primary text checked; its proof is cited, not re-proved.
MFT definition	Author-uploaded primary text checked by equation number.
DKI statistic	Publisher's parsed primary text checked at (1.6).
Exact scalar arithmetic	Earlier internal check script.
Original outer ZIP	Not distributed here; its certificate bytes are present.
Standalone C++ verifier + full tree replay	PASS; exact A, B, U, structure and ten negative tests.
Original Python verifier sources	Not in the standalone replay; old JSONs are comparisons.
November 22 two-compiler replay	PASS with GCC and Clang (same C++ algorithm).
November 28 tagged release	Public tree: 13/13 blob IDs; manifest: 12/12; full replay and negative tests: PASS.
Prior AI audit reports	Historical provenance, not mathematical premises.
Optional large-gap quadrature	Excluded from the present main theorem package.
External peer review and novelty	Not established by this draft.

The versioned computational artifact is public and its precise revision is identified in Appendix A. A DOI snapshot is optional and has not been created. The author must approve the manuscript license, submission metadata and disclosure before submission. Recovering and publishing the two historical Python source programs would be useful, but it is not a prerequisite for reproducing the verified result from the available C++ verifier.

G Primary references

The main text gives the bibliographic entries. The specific inputs are:

1. Tony Haddad and Dimitris Koukoulopoulos, *On Arratia's coupling and the Dirichlet law for the factors of a random integer*, J. Éc. polytech. Math. 12 (2025), 1565–1604, doi:10.5802/jep.317, especially pp. 1566–1568 and (1.5).
2. Michel Mendès France and Gérald Tenenbaum, *Systèmes de points, diviseurs et structure fractale*, Bull. Soc. Math. France 121 (1993), 197–225, doi:10.24033/bsmf.2207, especially (3.6) and (6.1).
3. Jean-Marie de Koninck and Aleksandar Ivić, *On the distance between consecutive divisors of an integer*, Canad. Math. Bull. 29 (1986), 208–217, doi:10.4153/CMB-1986-034-7, especially (1.6) and Theorem 2.