

A LEAN-CHECKED RANK CENSUS FOR PAIR-ALIGNED ONETWO SOBOLO' PROJECTIONS AT $m = 16$

LENNART RUDOLPH

ABSTRACT. This research note records a reproducible finite rank census for the 345 pair-aligned four-dimensional projections of a pinned version of the OneTwo Sobol' direction-number table at $m = 16$, corresponding to 65,536 points. Nine projections have exact $t = 4$, and the remaining 336 have exact $t = 5$. The emphasis is on the accompanying Lean-checked computations and an explicit worked certificate, rather than a new construction or algorithm for computing t -values. The OneTwo paper's reported $t \leq 4$ target extends through $m \leq 15$; the census here examines the next depth. The registered Lean development checks packed finite predicates for all 345 windows. Their interpretation as matrix-rank and digital-net quality statements, and identification with the released source table, remain outside Lean. We describe this boundary, link the fixed proof source and Palomar record, and identify remaining steps toward a reusable formalization of digital-net certificates.

1. SCOPE AND CONTRIBUTION

The OneTwo construction of Bonneel, Coeurjolly, Iehl, and Ostromoukhov provides a 692-dimensional Sobol' direction-number table with designated coordinate pairs $(2k - 1, 2k)$ [1, 3]. We study projections onto two successive pairs:

$$(1) \quad W_k = (2k - 1, 2k, 2k + 1, 2k + 2), \quad 1 \leq k \leq 345.$$

Dimension labels are one-based. Thus *pair-aligned* means that the window begins at an odd dimension; the family does not include every consecutive four-dimensional window or every four-coordinate subset.

Section 5.2 of the OneTwo paper reports numerical acceptance targets $t \leq 3$ for $m \leq 10$ and $t \leq 4$ for $m \leq 15$ for these blocks [1]. Our census concerns the next depth, $m = 16$, outside the stated $t \leq 4$ target. It does not assume that either smaller-depth target is met.

General algorithms for computing digital-net and projection t -values are already available [4, 5]. The contribution here is a reproducible certificate study of one fixed table: the complete pair-aligned census, exact-arithmetic verification scripts, an inspectable certificate for dimensions 25–28, and the accompanying Lean-checked statements described in Section 4. That section distinguishes those finite formal statements from their mathematical interpretation. The note introduces neither a new sequence construction nor a general t -value algorithm.

Date: Research note. Version 2.0.0, September 14, 2026.

2020 Mathematics Subject Classification. 11K38, 65C05, 68V05, 68V20.

Key words and phrases. Sobol' sequence, digital net, finite rank certificate, Lean, reproducibility.

Version note. This research note is version 2.0.0 of the work first published as *Exact Projection Quality of OneTwo Sobol' Sequences at 65,536 Points* (version 1.0.0, doi:10.5281/zenodo.21925582) [7]. The title and presentation have changed; the numerical census is unchanged.

2. CONVENTIONS AND FINITE CENSUS

2.1. Indexed points and the rank criterion. Let $m \geq 0$ and $s \geq 1$ be integers, and let $C_1, \dots, C_s \in \mathbb{F}_2^{m \times m}$. For each integer $0 \leq n < 2^m$, write $n = \sum_{c=0}^{m-1} n_c 2^c$ with $n_c \in \{0, 1\}$, put $\mathbf{n} = (n_0, \dots, n_{m-1})^\top$, and define

$$(y_{i,1}, \dots, y_{i,m})^\top = C_i \mathbf{n}, \quad x_{n,i} = \sum_{r=1}^m y_{i,r} 2^{-r}.$$

Write $x_n = (x_{n,1}, \dots, x_{n,s})$. All counts are over input indices n : coincident geometric points count with multiplicity, including after coordinate projection.

An elementary dyadic interval has the form

$$B(d, a) = \prod_{i=1}^s [a_i 2^{-d_i}, (a_i + 1) 2^{-d_i}), \quad d_i, a_i \in \mathbb{N}, \quad 0 \leq a_i < 2^{d_i}.$$

Here $\mathbb{N} = \{0, 1, \dots\}$. For $t_0 \in \{0, \dots, m\}$, an indexed family of 2^m points in $[0, 1]^s$ is a (t_0, m, s) -net in base two if every such interval with $\sum_i d_i = m - t_0$ contains exactly 2^{t_0} indexed points. The *exact* t -value is the smallest admissible $t_0 \in \{0, \dots, m\}$. A statement $t \leq 4$ does not say that this smallest value is four.

A *weak composition* of q into s parts is a vector $d = (d_1, \dots, d_s) \in \mathbb{N}^s$ with $\sum_i d_i = q$. For $q \leq m$, let $C(d)$ stack the first d_i rows of each C_i , in coordinate order. It is a $q \times m$ matrix.

Proposition 2.1 (Standard rank criterion). *Let $t_0 \in \{0, \dots, m\}$. The indexed digital family is a (t_0, m, s) -net if and only if*

$$\text{rank}_{\mathbb{F}_2} C(d) = m - t_0 \quad \text{for every } d \in \mathbb{N}^s \text{ with } \sum_i d_i = m - t_0.$$

Proof. For a fixed shape d , the linear map $\mathbf{n} \mapsto C(d)\mathbf{n}$ from $\mathbb{F}_2^m$ to $\mathbb{F}_2^{m-t_0}$ records each indexed point's output prefix, which determines its box of that shape. Full row rank makes the map surjective with a kernel of size 2^{t_0} . Each fiber is a coset of that kernel, so every box has the required count. A rank deficiency leaves some prefix unattained and hence a box empty. Applying this argument to each shape proves the equivalence [4, Sections 2.1 and 3.1]. $\square$

2.2. Fixed input and statement. The input file is `sampler/onetwo_matrices.cpp` in the OneTwo repository [3], at commit

163127a0c1731ffe2d43cd685067795c2689a308.

The artifact records Git blob `f183f3d97496ee3a60a0330f199ffc7a3dcf8f7f` and complete-file SHA-256

6f865af6a63983188e1ef8a03adaaa54fc9826ef98f9d4ccda649b1683566e51.

These identify the input file; source authentication is external to Lean.

The table stores 32-bit direction integers $v_{i,c}$ for coordinate i and input column c . In the data conversion, rows and columns are zero-based and bit_0 is the least significant bit. The matrix entry is

$$(2) \quad C_i[r, c] = \text{bit}_{31-r}(v_{i,c}), \quad 0 \leq r, c < 16.$$

We use the first sixteen input columns, no scrambling, and the complete prefix $0 \leq n < 2^{16}$. Keeping the first sixteen output digits means truncating, not rounding. This preserves membership in every tested dyadic interval, since each side exponent is at most sixteen.

Theorem 2.2 (Finite census for the pinned table). *For the 345 windows in (1) at $m = 16$, the exact four-dimensional t -value is four precisely for the starting dimensions*

$$1, 7, 31, 49, 67, 195, 225, 243, 373.$$

It is five for the other 336 windows. In particular, the projection onto dimensions 25–28 has exact $t = 5$.

Finite verification and mathematical interpretation. For a window with claimed value $t \in \{4, 5\}$, the upper test checks full row rank for every composition of total $16 - t$. A lower test supplies a dependent stack of total $17 - t$. There are $\binom{q+3}{3}$ four-part weak compositions of q : the upper tests therefore cover 455 compositions at total twelve for $t = 4$, or 364 at total eleven for $t = 5$. The respective lower witnesses have totals thirteen and twelve. The stored census records the upper totals and composition counts, together with a lower witness for every window.

For $t_0 < t$, extend the deficient composition coordinatewise to total $16 - t_0$. Its dependent rows remain in the enlarged stack; this total is at most sixteen, so no coordinate requires more than sixteen rows. Thus every smaller value is excluded, and Proposition 2.1 gives the exact value. The exact-arithmetic calculations in the reproduction artifact yield the displayed distribution and starting indices. The corresponding registered packed-predicate checks are described separately in Section 4. $\square$

3. A WORKED CERTIFICATE: DIMENSIONS 25–28

The worked projection makes the distinction between $t = 4$ and $t = 5$ concrete. For each of the 364 compositions of total eleven, `upper-minors.json` records column indices selecting an 11×11 minor whose determinant over $\mathbb{F}_2$ is one. Reconstructing and checking these minors gives full row rank for every such stack, hence $t \leq 5$. These are external numerical certificates; the registered Lean checks instead reduce rows by bitwise exclusive OR (XOR).

For the lower bound take $d_* = (0, 3, 3, 6)$: no rows from dimension 25, three each from dimensions 26 and 27, and six from dimension 28. Pack a row as $\sum_{c=0}^{15} C_i[r, c]2^c$. The twelve selected rows, written in hexadecimal, are

AF05, C566, 9664;
86A7, D08A, 3444;
BFF5, E9BA, B354, E718, 8750, 7AE0.

The groups come from dimensions 26, 27, and 28. The mask `0xD2F`, read from its least significant bit, selects zero-based row positions $\{0, 1, 2, 3, 5, 8, 10, 11\}$. Their XOR is zero:

$$\begin{aligned} &\text{AF05 xor C566 xor 9664 xor 86A7} \\ &\text{xor 3444 xor B354 xor 8750 xor 7AE0} = 0. \end{aligned}$$

Thus a total-twelve stack is dependent, ruling out $t \leq 4$.

Prescribe binary prefixes 100, 000, and 000000 in dimensions 26, 27, and 28. In the selected positions their parity is one, whereas the corresponding rows XOR to zero. Thus the prefix equations would require $0 = 1$. Consequently

$$[0, 1) \times [1/2, 5/8) \times [0, 1/8) \times [0, 1/64)$$

contains none of the indexed points. Its volume is 2^{-12} , while a $(4, 16, 4)$ -net would place sixteen points there. The upper minors and this dependence prove exact $t = 5$. The reproduction artifact also checks this box by direct enumeration of all 65,536 input indices.

4. LEAN DEVELOPMENT AND VERIFICATION BOUNDARY

4.1. What the registered development checks. The development uses Lean 4 [2] and is registered as PALOMAR-2026-09-05-000004, version 1 [6]. The associated proof source is fixed at commit

cc754a3bd42b2c0be28ff0a9e87c6d71e2fbfd82

in the `zenodo-21925582/palomar/` directory of the `lennrt/palomar-formalizations` repository. This proof-source pin is distinct from the direction-table pin in Section 2.

Each dimension’s sixteen 16-bit rows are packed into one natural number using 256 bit positions. The decoder extracts sixteen rows, each below 2^{16} . The Boolean function `independent` performs sixteen-column XOR elimination. For each row it scans bits from 15 down to 0. At a set bit, it XORs the stored pivot row if one exists and otherwise inserts the remaining row at that pivot. Reduction to zero returns false; successful insertion of every row returns true. This is the ordinary echelon-insertion interpretation of the code; its general soundness is discussed below.

Composition enumerators select leading rows. For each window, the encoded profile requires `independent` to return true for every composition of total $16 - t$, and false for at least one composition of total $17 - t$. The stored values are four and five. The development splits the checks across 35 source modules and assembles their results.

The three selected declarations are in the namespace `OneTwoSobolT5`:

Declaration	Encoded statement
<code>exact_t_five_certificate</code>	The worked block passes its threshold-five predicate and fails the predicates for zero through four.
<code>census_every_window</code>	Every pair-aligned window assembled from the encoded matrices passes its upper/lower profile check.
<code>census_distribution</code>	The claimed parameter list has length 345, with nine entries equal to four and 336 equal to five.

The theorem numbers in the fixed proof source refer to the earlier manuscript [7]. The names above identify the unchanged finite statements in this note.

The distribution declaration alone counts stored claims. The window-by-window declaration connects them to profiles computed from the encoded matrices. Both are needed for the census account.

The finite proofs use ordinary kernel-reduced `decide`, rather than `native_decide`. The v2 supplement includes the full registered source unchanged, replacing the worked-example-only companion distributed with v1.

4.2. Interpreting the checked statements. Theorem 2.2 is a mathematical statement about the quality of the digital point families. The registered declarations establish finite propositions expressed through the development’s packed-row predicates. Three connections in this account remain outside Lean:

- (1) *Source binding.* The hashes and the external `check_source_binding.py` script identify all 692 encoded matrices with the stated conversion of the upstream file.
- (2) *Linear algebra.* The executable independence test is not connected to abstract linear independence by a general Lean soundness theorem. The composition enumeration and row selection also lack a reusable formal correctness bridge to the mathematical definitions.
- (3) *Digital nets.* Proposition 2.1 and the resulting statements about geometric boxes and indexed counts are ordinary mathematical arguments, not selected Lean theorems.

Thus the finite calculations, their mathematical interpretation, and the external provenance checks supply distinct kinds of evidence. Palomar identifies the fixed statements and proof source; it is not cited as human peer review or a certificate of novelty.

5. REPRODUCIBILITY AND ANCILLARY CONTEXT

The supplement’s `artifact/` directory contains the numerical results in `census.json` and the main verifier `audit_sobol.py`. The verifier reconstructs the matrices and checks the worked certificate using exact arithmetic and Python’s standard library. Those worked-block checks can run offline from the bundled direction-number excerpt in `source-prefix.json`. Recomputing the full census requires the complete authenticated upstream table; see the supplemental README.

The evaluator uses two rank kernels with shared parsing and control flow. A separately written set-valued implementation supplies an additional check. These programs share the input and rank criterion; agreement is corroboration, not an independent proof of that criterion. The supplement also contains the full registered Lean project and the source-binding script. Its `VALIDATION.md` separates completed external checks from the fixed Palomar verification. No fresh Lean build or axiom transcript is claimed for v2.

For context, at $m = 15$ the numerical artifact gives exact $t = 3$ for W_1 (dimensions 1–4) and exact $t = 4$ for the other 344 windows. Thus all 345 meet $t \leq 4$ at that depth. This is an ancillary computation, not a formal theorem of the registered project. Smaller-depth diagnostics and the earlier manuscript’s discrepancy-bound discussion [7] are likewise outside the selected formalization.

6. DIRECTIONS FOR FURTHER FORMALIZATION

Three extensions would broaden the formal coverage. First, decode the packed rows into $\mathbb{F}_2^{16}$ and prove that the independence test agrees with linear independence for well-formed row lists. Include completeness of composition enumeration and correctness of row selection under explicit dimension bounds.

Second, formalize the digital-net rank criterion for indexed families: full-rank prefix maps have equally sized fibers, whereas rank deficiency leaves a prefix unattained. Coordinate-projection and complete-block index-permutation lemmas, with explicit hypotheses, would support reuse.

Third, develop a certificate interface between external computation and proof checking. An evaluator could supply nonsingular minors or elimination certificates for upper bounds and nonzero dependency vectors for lower bounds. A Lean checker with proved soundness could validate them against an identified table, supporting other depths and projection families without formalizing each evaluator.

These are unimplemented tasks involving existing mathematics, not new rank criteria or t -value algorithms. They do not change the present finite census. Larger censuses would need their own input provenance and certificate scope.

AI ASSISTANCE AND HUMAN RESPONSIBILITY

For version 1.0.0 of this work, published at doi:10.5281/zenodo.21925582 [7], OpenAI Codex and Anthropic Claude were used for programming, analysis, drafting, and automated review. For this version 2.0.0, ChatGPT and OpenAI Codex assisted with manuscript revision, supplemental programming, source inspection, citation checks, and release packaging. Anthropic Claude assisted with automated review of the v2 release. This assistance is neither human peer review nor additional formal verification. The registered Lean source is unchanged; no fresh completed Lean build is claimed for v2.

The human author, Lennart Rudolph, claims responsibility for the final manuscript, mathematical claims, distributed code and certificates, identification and use of source data, citations, and the accuracy of the stated verification scope and limitations. The AI tools are not authors.

REFERENCES

- [1] N. Bonneel, D. Coeurjolly, J.-C. Iehl, and V. Ostromoukhov. Sobol' sequences with guaranteed-quality 2D projections. *ACM Trans. Graph.*, 44(4):97:1–97:16, 2025. doi:10.1145/3730821.
- [2] L. de Moura and S. Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction—CADE 28*, pages 625–635. Springer, 2021. doi:10.1007/978-3-030-79876-5_37.
- [3] LIRIS Origami. OneTwoSobolSequences: Reference matrices and sampler. GitHub repository, 2025. Commit 163127a0c1731ffe2d43cd685067795c2689a308; accessed 2026-09-12. URL: <https://github.com/liris-origami/OneTwoSobolSequences/tree/163127a0c1731ffe2d43cd685067795c2689a308>.
- [4] P. Marion, M. Godin, and P. L'Ecuyer. An algorithm to compute the t -value of a digital net and of its projections. *J. Comput. Appl. Math.*, 371:112669, 2020. arXiv:1910.02277, doi:10.1016/j.cam.2019.112669.
- [5] G. Piršic and W. C. Schmid. Calculation of the quality parameter of digital nets and application to their construction. *J. Complexity*, 17(4):827–839, 2001. doi:10.1006/jcom.2001.0597.
- [6] L. Rudolph. Exact four-dimensional projection quality of the OneTwo Sobol' table at 65,536 points. Palomar Registry, 2026. Entry PALOMAR-2026-09-05-000004, version 1; registered September 5, 2026. URL: <https://palomar-registry.org/entry?id=PALOMAR-2026-09-05-000004&version=1>.
- [7] L. Rudolph. Exact projection quality of OneTwo Sobol' sequences at 65,536 points. Zenodo preprint, 2026. Version 1.0.0. doi:10.5281/zenodo.21925582.