hexagon:2611.01347v1

A Share-Deficit Bound and a 40-Word Local Identifying Code in the 8-Cube

Steven Minh Phat Duong
Independent Researcher, Calgary, Alberta, Canada

Hexagon version, 29 November 2026

Abstract

A local identifying code is a dominating set whose intersections with closed neighbourhoods distinguish adjacent vertices. Let $M^{\mathrm{L}}(n)$ be the minimum size of such a code in the binary n-cube. We give a self-contained share-deficit argument excluding 35-word codes in the 8-cube and an explicit 40-word construction, yielding $36 \le M^{\mathrm{L}}(8) \le 40$. The lower-bound argument classifies the equality cases of a local share estimate and shows that a hypothetical 35-word code would induce a disjoint union of three-vertex paths, contradicting its cardinality. The construction is accompanied by a short exhaustive verifier. These arguments do not depend on the project's SAT lower-bound certificates. This draft is AI-assisted and has not undergone independent human mathematical review; no priority claim is made.

1 Definition and scope

Write Q_n for the graph on $\{0,1\}^n$ whose edges join words differing in one coordinate. Addition of words is coordinatewise modulo two. Let e_j denote the jth unit word, and let $N[v]$ be the closed neighbourhood of v. For a code $C \subseteq \{0,1\}^n$, put $I(v) = N[v] \cap C$. Following Herva, Laihonen and Lehtilä [1], C is a local identifying code if $I(v) \neq \varnothing$ for every vertex and $I(u) \neq I(v)$ whenever u, v are adjacent. Separation of all pairs is not required.

The cited paper reports $35 \le M^{\mathrm{L}}(8) \le 48$. We address this one dimension. Our result is the interval 36–40, not an exact determination of $M^{\mathrm{L}}(8)$. The lower bound below is an explicit mathematical argument; the upper bound is an explicit finite witness. Other computations in the wider project are not used as premises.

2 A local share estimate

The share upper bound $s(c) \le n - 2/3$ is established in Theorem 2.4 of [1]. We give a coordinate-based derivation that also tracks equality and the deficit away from equality, which are needed to exclude 35 words. For each $c \in C$, define its share and deficit by

$$s(c) = \sum_{v \in N[c]} \frac{1}{|I(v)|}, \qquad B_n = n - \frac{2}{3}, \qquad \delta(c) = B_n - s(c).$$

Since every vertex is covered, double counting gives

$$\sum_{c \in C} s(c) = 2^n. \tag{1}$$

Translate a chosen codeword to $c = 0$. Put

$$S = \{j : e_j \in C\}, \quad p = |S|, \quad E = \{\{j,k\} : j < k,\ e_j + e_k \in C\}, \quad q_j = \deg_E(j).$$

Here E is an auxiliary simple graph on the coordinate indices, not the graph induced by C. Directly from the neighbourhoods,

$$s(c) = \frac{1}{p+1} + \sum_{j=1}^{n} \frac{1}{1 + \mathbf{1}_{j\in S} + q_j}. \tag{2}$$

Call c a *center* if $p = 2$ and $E = \varnothing$, and an *end* if $p = 1$, $S = \{a\}$ and $E = \{\{a,b\}\}$ for some $b \neq a$.

Lemma 1. *For $n \geq 4$, every codeword satisfies $s(c) \leq B_n$. Equality holds precisely at centers and ends. Every other codeword has $\delta(c) \geq 1/3$. A center has two codeword neighbours, each with exactly one codeword neighbour. An end has one codeword neighbour, which has exactly two codeword neighbours.*

Proof. Each summand of (2) decreases when q_j increases.

If $p = 0$, then $I(c) = \{c\}$. Local separation from e_j forces $q_j \geq 1$ for all j. Thus $s(c) \leq 1 + n/2$ and $\delta(c) \geq n/2 - 5/3 \geq 1/3$.

If $p \geq 3$, dropping all q_j gives

$$s(c) \leq n - \frac{p}{2} + \frac{1}{p+1} \leq n - \frac{5}{4},$$

so $\delta(c) \geq 7/12$.

If $p = 2$, relabel so $S = \{1,2\}$. With $E = \varnothing$ the share is $1/3 + 1 + (n-2) = B_n$. Adding a first edge to E decreases the share by at least $1/3$: the decreases according as the edge has two, one, or no endpoints in S are $1/3$, $2/3$, and 1. Further edges only decrease it.

If $p = 1$, relabel so $S = \{1\}$. Local separation of c and e_1 forces $q_1 \geq 1$, so choose an edge $\{1,k\} \in E$. Keeping just this edge gives share $1/2 + 1/3 + 1/2 + (n-2) = B_n$. Any further edge has one of three forms. An edge $\{1,l\}$, $l \neq k$, decreases the share by $7/12$; an edge $\{k,l\}$, $l \neq 1$, decreases it by $2/3$; a disjoint edge decreases it by 1. These are the decreases from the one-edge configuration; after the first additional edge all further additions decrease the share. Equality therefore occurs exactly in the stated end configuration.

Finally, the codeword neighbours of $e_j \in C$ are 0 and the words $e_j + e_k \in C$. Their numbers are $1 + q_j$, proving both degree assertions. □

3 Exclusion of 35 words

Theorem 2. *Every local identifying code in Q_8 has at least 36 words.*

Proof. By (1) and Lemma 1, $|C|(22/3) \geq 256$, hence $|C| \geq 35$. Suppose $|C| = 35$. Then

$$\sum_{c\in C} \delta(c) = 35 \cdot \frac{22}{3} - 256 = \frac{2}{3}. \tag{3}$$

Call a codeword bad if it is neither a center nor an end. All deficits are nonnegative and each bad vertex contributes at least $1/3$, so there are at most two bad vertices.

Let $G = Q_8[C]$. An isolated vertex has $p = 0$ and deficit at least $7/3$, contradicting (3). A vertex of degree at least three is bad, as is each of its neighbours: by the last assertion of Lemma 1,

a center can have only degree-one neighbours and an end only a degree-two neighbour. This would give at least four bad vertices. Consequently every vertex of G has degree one or two.

An edge between two degree-one vertices u, v is impossible because $I(u) = I(v) = \{u, v\}$. We next exclude an edge between degree-two vertices. Translate and relabel so these are $c = 0$ and $d = e_1$, with other neighbours $a = e_2$ and $b = e_1 + e_k$, respectively.

If $k = 2$, these four vertices form a square. Each has degree exactly two and sees a coordinate-graph edge joining its two codeword-neighbour directions. The $p = 2$ calculation in Lemma 1 gives deficit at least $1/3$ at each of the four vertices, a total of at least $4/3$.

If $k \neq 2$, at c the word b supplies the edge $\{1, k\}$, with exactly one endpoint in $S_c = \{1, 2\}$. Hence $\delta(c) \geq 2/3$. Relative to d, the word a supplies the edge $\{1, 2\}$, with exactly one endpoint in $S_d = \{1, k\}$, giving $\delta(d) \geq 2/3$. This again contradicts (3).

Every edge of G therefore joins a degree-one vertex to a degree-two vertex. Every connected component is a three-vertex path. It follows that 3 divides $|C|$, contrary to $|C| = 35$. □

4 A 40-word construction

Proposition 3. *The following 40 binary words form a local identifying code in Q_8:*

00000100	00001100	00010010	00010011
00010111	00100100	00101001	00101011
00111001	00111110	01000001	01001010
01001111	01011000	01011101	01100010
01100111	01110000	01110101	01111110
10000001	10001010	10001111	10011000
10011101	10100010	10100111	10110000
10110101	10111110	11000001	11000110
11010100	11010110	11011011	11101000
11101100	11101101	11110011	11111011

Finite verification. For each of the 256 vertices, intersect the displayed code with its closed neighbourhood. Every intersection is nonempty, and the intersections differ across each of the 1,024 undirected edges. The complete calculation is specified by the short verifier in Appendix A. A fresh implementation of this direct test returned 256 covered vertices and 1,024 separated edges on 18 November 2026. Its only input is the displayed list; it does not use the search algorithm or solver logs. □

Combining Theorem 2 and Proposition 3 gives

$$\boxed{36 \leq M^{\mathrm{L}}(8) \leq 40.}$$

The construction was obtained during the project's computational search. Its validity is separate from whether that search exhausts any family. We do not infer the nonexistence of smaller codes from unsuccessful searches.

5 Evidence, limitations and availability

The share-deficit proof is independent of the project's computer-assisted lower-bound argument. No SAT solver, local SAT enumeration, or negative-search log is a premise of Theorem 2. The

construction requires only a short finite check. These distinctions allow this note to be read without the larger certificate collection.

This draft does not establish the exact optimum in Q_8, certify novelty against every later publication, or claim any result for Q_9. The broader project's Q_6, Q_7 and Q_{10} claims require separate accounts of their encodings, reductions, certificates and trust assumptions; they are not asserted or used here. A limited literature check did not locate a later matching result, but an exhaustive citation search has not been completed.

The witness list, Python verifier and verification record are available under their respective licences in the archived preprint package, Zenodo 10.5281/zenodo.24335625 (18 November 2026). The word list and the verification listing are also printed here. This version updates the availability wording and leaves the mathematical claims unchanged.

AI-use and review disclosure. AI systems assisted with the computational search, development and checking of the mathematical argument, and preparation of this text. The present argument has been inspected by AI reviewers, but no independent human mathematical review or peer review is documented. AI review is not described as independent human validation. No institutional affiliation or faculty endorsement is claimed.

A Direct verifier

Save the 40 words in the displayed order, one per line, in `q8_k40.txt`. Run the following Python 3 program from the same directory. All arithmetic is exact.

```
from pathlib import Path
words = Path("q8_k40.txt").read_text().split()
assert len(words) == len(set(words)) == 40
assert all(len(w) == 8 and set(w) <= {"0", "1"}
           for w in words)
C = {int(w, 2) for w in words}
def N(v):
    return {v} | {v ^ (1 << i) for i in range(8)}
I = {v: N(v) & C for v in range(256)}
assert all(I.values())
edges = [(v, v ^ (1 << i))
         for v in range(256) for i in range(8)
         if v < (v ^ (1 << i))]
assert len(edges) == 1024
assert all(I[u] != I[v] for u, v in edges)
print("PASS: 40 words; 256 vertices; 1024 edges")
```

Do not run Python with `-O`, which disables assertions. The supplied list has SHA-256

`eafaefaa1b8093d18a92c6eb1d2b176247b2d6efd6178d332f8dee366e7ddeb0.`

References

[1] P. Herva, T. Laihonen and T. Lehtilä, Optimal Local Identifying and Local Locating-dominating Codes, *Fundamenta Informaticae* **191**(3–4) (2024), 351–378. arXiv:2302.13351, version 3.