Color-split frame reductions for queen domination on even boards of sizes 50 through 64

Edwin Chan*

November 27, 2026

Abstract

For every even n from 50 through 64, the queen's graph Q_n has domination and independent domination numbers $\gamma(Q_n) = i(Q_n) = n/2 + 1$. To prove the lower bounds, we group $n/2$-queen placements by their outermost empty rows and columns and refine the resulting counts by chessboard color. We refute all 290 542 SAT instances with DRAT proofs checked by `drat-trim`. The published independent placements of Östergård and Weakley give matching upper bounds for both parameters. We also give a 31-queen independent placement on Q_{60} and report the time, memory, and storage used by the computations.

hexagon:2611.01169v1

1. Introduction

The queen's graph Q_n has the squares of an $n \times n$ chessboard as vertices. Distinct squares are adjacent if they share a row, column, or diagonal. The domination number $\gamma(Q_n)$ is the least size of a set of queens that occupies or attacks every square. The independent domination number $i(Q_n)$ is the minimum size of such a set with pairwise non-attacking queens. Thus $\gamma(Q_n) \le i(Q_n)$.

For even n, the classical lower bound is $\gamma(Q_n) \ge n/2$, due to Spencer (as reported by Cockayne [3]) and Raghavan and Venkatesan [10]. For even $26 \le n \le 120$, the constructions of Östergård and Weakley [9, Section 5] give $i(Q_n) \le n/2+1$. We determine which of the two values occurs on the eight even boards from Q_{50} through Q_{64}.

Theorem 1.1. *For every even* $50 \le n \le 64$, $\gamma(Q_n) = n/2 + 1$. *Explicitly,*

$$\gamma(Q_{50}) = 26, \quad \gamma(Q_{52}) = 27, \quad \gamma(Q_{54}) = 28, \quad \gamma(Q_{56}) = 29,$$
$$\gamma(Q_{58}) = 30, \quad \gamma(Q_{60}) = 31, \quad \gamma(Q_{62}) = 32, \quad \gamma(Q_{64}) = 33.$$

Proof. A dominating set of size at most $k = n/2$ extends to one of size k by adding squares. By Theorem 6.2, some image of this set under a board symmetry would extend to a satisfying assignment of a generated SAT instance. The DRAT checks in Section 7 refute every instance on the eight boards. The placements in Appendix A give the upper bounds. □

Corollary 1.2. *For every even* $50 \le n \le 64$, $i(Q_n) = \gamma(Q_n) = n/2 + 1$.

Proof. Theorem 1.1 gives $n/2+1 = \gamma(Q_n) \le i(Q_n)$. The published independent placements [9, Section 5, p. 11] give the reverse bound. See also Weakley [13, Proposition 21] and the coordinate checks in Appendix A. □

*Gemini 4 Neon carried out the original research in Google Antigravity. See Section 9 for contributions.

1.1. Previous work and proof outline

The two problems go back to de Jaenisch's 1863 appendix [9, 2017 corrigendum] and appear as Problem C18 in Guy [6]. Their sequences in the OEIS are A075458 and A075324, respectively.

Neuhaus [8, Section 2.4 and Chapter 4] reported integer-programming results for several even boards through $n = 46$ and for selected larger boards. His analysis of frame sides used parity to track excess diagonal coverage. Weakley [13, Sections 2–3] obtained balance conditions for bichromatic sets and exclusions for monochromatic ones. The `BSC` Lean proof for Q_{26} also uses parity and polynomial identities [7].

Hint and Conquer [11] split the SAT search by the outermost empty rows and columns and used side counts to certify even values through $n = 38$. Davies [4, Sections 2 and 4] refined this split by the deficits in the numbers of occupied lines and certified further even values through $n = 46$. His even-board runs used the default CaDiCaL 3.1.0 configuration without symmetry breaking or cubing. Table 2 of that paper reports single-core CPU times on a shared machine. Đukić [5] reported a DRAT-certified computation for $n = 48$. On the even boards with $32 \le n \le 48$, these ordinary values and the published independent placements also give $i(Q_n) = n/2 + 1$.

Our reduction counts border and side incidences separately for the two chessboard colors. These counts rule out some frames algebraically and split the rest into SAT instances with tighter constraints. Sections 2–6 derive the reductions and prove that the instances cover every possible dominating $n/2$-set. Section 7 reports the checks, and Appendix A lists the placements.

2. Frame geometry and dihedral symmetry

Coordinates $(x, y) \in \{0, \dots, n-1\}^2$ index columns and rows. The difference and sum diagonals through (x, y) have indices $t = y - x$ and $s = x + y$, respectively. The chessboard color of a square is $(x + y) \bmod 2$, taken in $\{0, 1\}$. Each diagonal has the color of its index modulo two, since $y - x \equiv x + y \pmod 2$.

Unless stated otherwise, take $n = 2k \ge 4$ and let D be a dominating set of k queens. Write R, C for the occupied rows and columns and T, S for the occupied difference and sum diagonals. We identify each line by its type and index, so T and S are disjoint. For each line family the *deficit* is the number of queens minus the number of occupied lines.

$$d_R = k - |R|, \quad d_C = k - |C|, \quad d_T = k - |T|, \quad d_S = k - |S| \qquad (\text{all } \ge 0).$$

At least two rows and two columns are empty because $n - |R|, n - |C| \ge k \ge 2$. Let $a < b$ be the outermost empty columns and $c < d$ the outermost empty rows (Figure 1). Use integer intervals throughout.

- The margins are $m_l = a$, $m_r = n - 1 - b$, $m_b = c$, and $m_t = n - 1 - d$.
- The frame has width $w = b - a \ge 1$ and height $h = d - c \ge 1$.
- The vertical sides are $V = \{a, b\} \times [c, d]$ and the horizontal sides are $H = [a, b] \times \{c, d\}$. The border $\mathrm{Bd} = V \cup H$ counts each corner once.

Every column outside $[a, b]$ and every row outside $[c, d]$ is occupied. Thus

$$m_l + m_r \le |C| = k - d_C, \qquad m_b + m_t \le |R| = k - d_R. \tag{1}$$

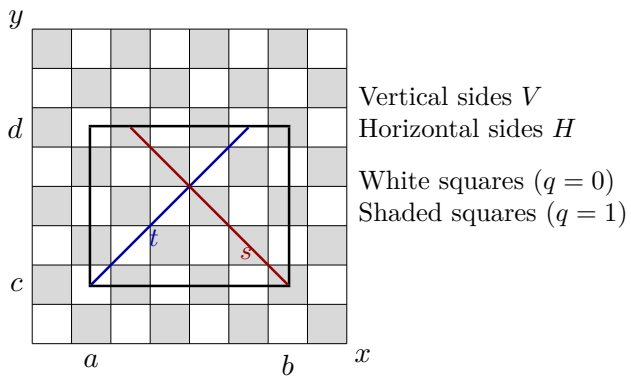

Figure 1. The border $\mathrm{Bd} = V \cup H$ on an 8×8 board, drawn through the centers of the boundary squares. The difference diagonal $t = 0$ has color 0 and the sum diagonal $s = 7$ has color 1.

A square is *needy* if its row and column are both empty. Every needy square must lie on an occupied diagonal. The border contains no queens because each side lies on an empty line.

Let z_q count queens of color q, so $z_0 + z_1 = k$, and let T_q, S_q be the occupied diagonals of that color. If $d_T = 0$ then $|T_q| = z_q$, and if $d_S = 0$ then $|S_q| = z_q$. Write $e_p(R)$ and $e_p(C)$ for the numbers of empty rows and columns of parity p. A placement using one color is *monochromatic*. A placement using both is *bichromatic*.

The margins and deficits form the *case* of D,

$$\kappa(D) = (m_l, m_r, m_b, m_t;\ d_R, d_C, d_T, d_S).$$

Write $\rho(x) = n - 1 - x$. The following table gives the action of the dihedral group D_4 on squares and on the case tuple.

$$\begin{aligned}
g_0 &: (x,y) \mapsto (x,y), && (m_l, m_r, m_b, m_t;\ d_R, d_C, d_T, d_S),\\
g_1 &: (x,y) \mapsto (\rho(x),y), && (m_r, m_l, m_b, m_t;\ d_R, d_C, d_S, d_T),\\
g_2 &: (x,y) \mapsto (x,\rho(y)), && (m_l, m_r, m_t, m_b;\ d_R, d_C, d_S, d_T),\\
g_3 &: (x,y) \mapsto (\rho(x),\rho(y)), && (m_r, m_l, m_t, m_b;\ d_R, d_C, d_T, d_S),\\
g_4 &: (x,y) \mapsto (y,x), && (m_b, m_t, m_l, m_r;\ d_C, d_R, d_T, d_S),\\
g_5 &: (x,y) \mapsto (\rho(y),x), && (m_t, m_b, m_l, m_r;\ d_C, d_R, d_S, d_T),\\
g_6 &: (x,y) \mapsto (y,\rho(x)), && (m_b, m_t, m_r, m_l;\ d_C, d_R, d_S, d_T),\\
g_7 &: (x,y) \mapsto (\rho(y),\rho(x)), && (m_t, m_b, m_r, m_l;\ d_C, d_R, d_T, d_S).
\end{aligned}$$

On an even board g_1, g_2, g_5, g_6 reverse chessboard parity and exchange diagonal types. In a square frame they exchange the two corner-to-corner diagonals. The other four maps preserve parity and diagonal types. The maps g_4, g_5, g_6, g_7 exchange vertical and horizontal sides.

We take the lexicographically least of the eight case tuples. This extends the frame-only canonicalization in Davies [4, Sections 2 and 4] to include the deficits. Every placement has a dihedral image in a canonical case. For such a case κ, the stabilizer $\mathrm{Stab}(\kappa) = \{g \in D_4 : g(\kappa) = \kappa\}$ gives the symmetries still available within that case.

3. Color-split frame and side identities

We count how many needy squares the occupied diagonals cover on the border and on each pair of sides. For $X \subseteq \mathrm{Bd}$, write X_q for its squares of color q and $N_q(X)$ for the number of needy squares in X_q. At a square p, let $m(p) \in \{0,1,2\}$ count occupied diagonals. For any diagonal L, let $\beta_X(L) = |X \cap L|$.

Every diagonal meets Bd in at most two squares. A diagonal meeting both vertical sides changes height by w, so when $w \geq h$ this can occur only at opposite corners in a square frame. The horizontal statement follows by transpose.

The *box diagonals* of a square frame are the difference diagonal U through $(a,c),(b,d)$, with $t = c - a$, and the sum diagonal W through $(a,d),(b,c)$, with $s = a + d$. Write $[P]$ for the indicator of a proposition P. Set

$$u_0 = [U \in T], \qquad v_0 = [W \in S], \qquad \delta_q = [U \in T_q] + [W \in S_q].$$

Set $u_0 = v_0 = \delta_q = 0$ for nonsquare frames. The *waste* counts record unused coverage. They count missed intersections, hits on squares already covered by a row or column, and second hits on needy squares. The subscript F denotes the full border.

$$\begin{aligned}
\mathrm{waste}_F^{(q)} &:= \sum_{L \in T_q \cup S_q} (2 - \beta_{\mathrm{Bd}}(L)) \\
&\quad + \#\{(L,p) : L \in T_q \cup S_q,\ p \in \mathrm{Bd}_q \cap L \text{ non-needy}\} \\
&\quad + \#\{\text{needy } p \in \mathrm{Bd}_q : m(p) = 2\}, \\
\mathrm{waste}_V^{(q)} &:= \#\{L \in T_q \cup S_q : \beta_V(L) = 0\} \\
&\quad + \#\{(L,p) : L \in T_q \cup S_q,\ p \in V_q \cap L \text{ non-needy}\} \\
&\quad + \#\{\text{needy } p \in V_q : m(p) = 2\}.
\end{aligned}$$

Define $\mathrm{waste}_H^{(q)}$ with H in place of V. These counts also make sense on odd boards whenever at least two rows and two columns are empty.

Theorem 3.1 (Border and side counts by color)**.** *For every dominating set D of k queens on Q_n with $k \leq n-2$, and each $q \in \{0,1\}$, we have*

$$\mathrm{waste}_F^{(q)} = 2(|T_q| + |S_q|) - N_q(\mathrm{Bd}) \geq 0, \tag{2}$$

$$\mathrm{waste}_V^{(q)} = (|T_q| + |S_q|) + \delta_q - N_q(V) \geq 0 \qquad (w \geq h), \tag{3}$$

$$\mathrm{waste}_H^{(q)} = (|T_q| + |S_q|) + \delta_q - N_q(H) \geq 0 \qquad (h \geq w). \tag{4}$$

Proof. Fix a color q. Let $I_q(X)$ count incidences between occupied diagonals of color q and squares in X_q. Each needy square contributes at least one incidence, giving

$$\begin{aligned}
I_q(X) = \sum_{p \in X_q} m(p) &= N_q(X) + \#\{\text{needy } p \in X_q : m(p) = 2\} \\
&\quad + \#\{(L,p) : L \in T_q \cup S_q,\ p \in X_q \cap L \text{ non-needy}\}.
\end{aligned}$$

Counting by diagonals gives $I_q(X) = \sum_{L \in T_q \cup S_q} \beta_X(L)$. On the full border, this is twice the number of occupied diagonals of color q minus the first term of $\mathrm{waste}_F^{(q)}$. On V when $w \geq h$, each occupied diagonal contributes one hit, with zero hits for those that miss V and an extra hit

for each occupied box diagonal of color q. The corresponding count on H applies when $h \geq w$. Substituting these counts gives (2)–(4). Every term in the waste counts is nonnegative. □

Each empty row contributes two needy squares to V and each empty column two to H. All four corners are needy and counted in both, so the totals over both colors are

$$\sum_q N_q(V) = 2(n - |R|), \qquad \sum_q N_q(H) = 2(n - |C|),$$
$$\sum_q N_q(\mathrm{Bd}) = 2(n - |R|) + 2(n - |C|) - 4.$$

Write $\mathrm{waste}_F, \mathrm{waste}_V, \mathrm{waste}_H$ for the sums over both colors. For $n = 2k$ the identities reduce to those of Davies [4, Section 2].

$$\mathrm{waste}_F = 4 - 2(d_R + d_C + d_T + d_S) \geq 0, \tag{5}$$
$$\mathrm{waste}_V = -d_T - d_S - 2d_R + [w = h](u_0 + v_0) \geq 0 \qquad (w \geq h), \tag{6}$$
$$\mathrm{waste}_H = -d_T - d_S - 2d_C + [w = h](u_0 + v_0) \geq 0 \qquad (h \geq w). \tag{7}$$

In particular, $d_R + d_C + d_T + d_S \leq 2$, so at least two deficits are zero. The four are at most $k - 1$, since a nonempty placement occupies a line of each type.

4. Algebraic exclusion of cases

The bounds on waste already restrict the deficits. Four further tests exclude cases before the SAT formulas are built. The margin test takes $O(k)$ steps per case, and each parity test takes a constant number of steps.

Lemma 4.1 (NS-1: Margin bounds for nonsquare frames)**.** *Suppose $w > h$. Then $d_R = d_T = d_S = 0$ and $\mathrm{waste}_V = 0$, and the margins satisfy the following bounds.*

(i) $m_l \leq \lfloor h/2 \rfloor$ *and* $m_r \leq \lfloor h/2 \rfloor$*;*

(ii) $m_l + m_r + m_b + m_t \leq k$*;*

(iii) for every $r \in \{1, \ldots, \max(m_l, m_r)\}$, $\max(0, m_l - r + 1) + \max(0, m_r - r + 1) \leq h + 1 - 2r$*;*

(iv) For each $m \in \{m_b, m_t\}$ *with* $m > 0$*, one has* $2m \leq w$ *and* $w \leq 2h + 2$*.*

For $h > w$, transpose by g_4.

Proof. By (6), $-d_T - d_S - 2d_R \geq 0$. Thus all three deficits vanish and $\mathrm{waste}_V = 0$. Every occupied diagonal meets V, so each queen $(x, y) \in D$ satisfies

$$y - x \in [c - b, d - b] \cup [c - a, d - a],$$
$$x + y \in [a + c, a + d] \cup [b + c, b + d].$$

For a margin column $x = a - \delta$, $\delta \geq 1$, the first difference interval gives $y \leq d - w - \delta \leq c - 2$. Both sum intervals give $y \geq c + \delta$. The second sum interval gives $y \geq c + w + \delta \geq d + 2$, while both difference intervals give $y \leq d - \delta$. The remaining intersection is

$$y \in [c + \delta,\ d - \delta] \subseteq (c, d),$$

which requires $2\delta \le h$. On any side with a positive margin, its outermost occupied column gives (i).

Choose one queen in each margin column. Since $d_R = 0$, they occupy distinct rows, all inside (c, d). The $m_b + m_t$ occupied rows outside $[c, d]$ bring the total to at least $m_l + m_r + m_b + m_t$, giving (ii).

For (iii), the chosen queens in columns at distance at least r from $[a, b]$ must fit in the $h + 1 - 2r$ rows of $[c + r, d - r]$. There are $\max(0, m_l - r + 1)$ such columns on the left and $\max(0, m_r - r + 1)$ on the right.

For (iv), take a margin row $y = c - \delta$ with $\delta \ge 1$. The difference and sum intervals restrict x to $(X_{t,1} \cup X_{t,2}) \cap (X_{s,1} \cup X_{s,2})$, where

$$X_{t,1} = [b - h - \delta,\ b - \delta], \qquad X_{t,2} = [a - h - \delta,\ a - \delta],$$
$$X_{s,1} = [a + \delta,\ a + h + \delta], \qquad X_{s,2} = [b + \delta,\ b + h + \delta].$$

The interval $X_{t,2}$ lies strictly to the left of both sum intervals, and $X_{t,1}$ ends before $X_{s,2}$ begins. Thus

$$x \in [\max(b - h - \delta,\ a + \delta),\ \min(b - \delta,\ a + h + \delta)],$$

which is nonempty exactly when $2\delta \le w$ and $w - 2h \le 2\delta$. If $m_b > 0$, row 0 gives the first bound with $\delta = m_b$, and row $c - 1$ gives the second with $\delta = 1$. Reflection gives the bounds for m_t. □

Lemma 4.2 (NS-2: Nonsquare frames with an odd longer side)**.** *If k is odd, the longer side of a nonsquare frame has even length.*

Proof. Transpose if needed and suppose $w > h$ with w odd. The columns a, b have opposite parity, so each of the k empty rows contributes one needy square of each color to V. Lemma 4.1 gives $|T_q| = |S_q| = z_q$ and $\text{waste}_V^{(q)} = 0$. Equation (3) then gives $2z_q = k$, impossible for odd k. □

Lemma 4.3 (NS-3: Mixed side parity and deficit two)**.** *For odd k, a nonsquare frame with $w > h$, w even, and h odd cannot have $d_C = 2$. The transposed case with $d_R = 2$ is also impossible.*

Proof. Suppose $d_C = 2$ in the stated orientation. Lemma 4.1 and (3) give $N_q(V) = 2z_q$. Since h is odd, each of the $k + 2$ empty columns contributes one needy square of each color to H. Two corners of each color lie in both side sets, so $N_q(\text{Bd}) = 2z_q + k$. By (2),

$$0 \le \text{waste}_F^{(q)} = 4z_q - (2z_q + k) = 2z_q - k \implies 2z_q \ge k \qquad (q \in \{0, 1\}).$$

Since k is odd, $z_0 \ge (k + 1)/2$ and $z_1 \ge (k + 1)/2$, contradicting $z_0 + z_1 = k$. □

Lemma 4.4 (SQ-RC-Odd: A single row or column deficit)**.** *For odd k, a square frame of odd side length cannot have deficit vector $(0, 1, 0, 0)$ or $(1, 0, 0, 0)$.*

Proof. Transpose if needed and assume $d_C = 1$. By (7), $\text{waste}_H = -2 + u_0 + v_0 \ge 0$, so both box diagonals are occupied and $\text{waste}_H = 0$. The odd side length puts one box diagonal in each color and one needy square of each color on H in every empty column. Thus $N_q(H) = k + 1$ and $\delta_q = 1$. With $d_T = d_S = 0$, (4) gives $2z_q + 1 = k + 1$, again impossible for odd k. □

5. Branches by parity and deficit

The remaining cases are either nonsquare or square with total deficit at most two. Within each case we branch on the color counts, the box diagonals, or both. Each resulting SAT instance is called a *leaf*.

Write $p_a = a \bmod 2$ and $p_c = c \bmod 2$, and use $\oplus$ for addition modulo two. If w, h are both even, all four corners have color $q_* = p_a \oplus p_c$. Write $\bar{q}_* = 1 - q_*$. For an odd-sided square, let η_q indicate whether its box diagonal of color q is occupied.

5.1. Nonsquare cases

Transpose if needed so $w > h$. Then $d_R = d_T = d_S = 0$, $d_C \in \{0, 1, 2\}$, and $\mathrm{waste}_V^{(q)} = 0$ for both colors.

- **Odd w.** By Lemma 4.2, k is even. The vertical side equation gives $2z_q = N_q(V) = k$, so $|T_q| = |S_q| = k/2$ for each color. The border waste is at most $4 - 2d_C$ and vanishes when $d_C = 2$.
- **Even w, odd h.** Here $N_q(V) = 2e_{p_a \oplus q}(R) = 2z_q$. Each empty column adds one square of color q to H, with two corners already counted on V. Thus $N_q(\mathrm{Bd}) = 2z_q + k + d_C - 2$ and

$$\mathrm{waste}_F^{(q)} = 2z_q - k - d_C + 2 \geq 0 \qquad (q \in \{0, 1\}).$$

  Branch on $z_0 \in [\lceil (k + d_C - 2)/2 \rceil, \lfloor (k - d_C + 2)/2 \rfloor]$. There are at most two choices for odd k and three for even k. If a case symmetry exchanges colors, keep $z_0 \leq k - z_0$. Each choice fixes the common count $|T_q| = |S_q| = e_{p_a \oplus q}(R) = z_q$, as well as the border waste of each color.
- **Even w and h.** Put $z = z_{\bar{q}_*}$. Then $N_{\bar{q}_*}(V) = 2e_{1-p_c}(R) = 2z$. All corners have color q_*, so $N_{\bar{q}_*}(\mathrm{Bd}) = 2e_{1-p_c}(R) + 2e_{1-p_a}(C)$. With $\Delta = z - e_{1-p_a}(C)$ the two border wastes are

$$\mathrm{waste}_F^{(\bar{q}_*)} = 2\Delta, \qquad \mathrm{waste}_F^{(q_*)} = 4 - 2d_C - 2\Delta.$$

  Thus $\Delta \in \{0, \ldots, 2 - d_C\}$. The first waste vanishes at the lower endpoint and the second at the upper endpoint.

5.2. Square frames with total deficit two

The side inequalities leave only $(0, 0, 2, 0)$, $(0, 0, 0, 2)$, $(0, 0, 1, 1)$, and $(1, 1, 0, 0)$, subject to the individual bound $k - 1$. In each case (5)–(7) force $u_0 = v_0 = 1$ and zero border and side waste in both colors.

5.3. Square frames with one row or column deficit

Take $d_C = 1$ and the other deficits zero. The case $d_R = 1$ follows by transposition. Equation (7) gives $u_0 = v_0 = 1$ and $\mathrm{waste}_H = 0$.

- **Odd w.** The argument of Lemma 4.4 gives $2z_q = k$, so k is even. The vertical and border equations give $\mathrm{waste}_V^{(q)} = \mathrm{waste}_F^{(q)} = 1$ for each color.

- **Even w.** Zero horizontal waste in color $\bar{q}_*$ gives $z_{\bar{q}_*} = e_{1-p_a}(C)$. Put $\Delta = z_{\bar{q}_*} - e_{1-p_c}(R)$. Equations (3) and (2) give

$$\text{waste}_V^{(\bar{q}_*)} = \text{waste}_F^{(\bar{q}_*)} = 2\Delta, \qquad \text{waste}_V^{(q_*)} = \text{waste}_F^{(q_*)} = 2 - 2\Delta.$$

  Hence Δ is zero or one. Color $\bar{q}_*$ has zero waste on all three sets when $\Delta = 0$, and color q_* has zero waste when $\Delta = 1$.

5.4. Square frames with one diagonal deficit

Take $d_S = 1$ and the other deficits zero. For $d_T = 1$, exchange T and S. Every queen occupies a distinct difference diagonal, so $|T_q| = z_q$. The one sum diagonal occupied twice has color q_{def}, and $|S_q| = z_q - [q = q_{\text{def}}]$. We branch on q_{def}.

- **Odd w.** Let $q' = 1 - q_{\text{def}}$. By (3),

$$2z_{q_{\text{def}}} - 1 + \eta_{q_{\text{def}}} \geq k, \qquad 2z_{q'} + \eta_{q'} \geq k.$$

  For odd k, the inequalities force $z_{q_{\text{def}}} = (k+1)/2$, $z_{q'} = (k-1)/2$, and $\eta_{q'} = 1$. The color q' has zero waste on the border and both sides. For even k, they force $z_0 = z_1 = k/2$ and $\eta_{q_{\text{def}}} = 1$. In this case q_{def} has zero waste on all three sets. In each case the other color has border waste two and side wastes equal to its box-diagonal indicator.

- **Even w.** Both box diagonals have color q_*. Put $z = z_{\bar{q}_*}$, $e_V = e_{1-p_c}(R)$, $e_H = e_{1-p_a}(C)$, and $\epsilon = [q_{\text{def}} = \bar{q}_*]$. No corner has color $\bar{q}_*$, so (2)–(4) give

$$\begin{aligned}
\text{waste}_V^{(\bar{q}_*)} &= 2z - \epsilon - 2e_V, \\
\text{waste}_H^{(\bar{q}_*)} &= 2z - \epsilon - 2e_H, \\
\text{waste}_F^{(\bar{q}_*)} &= \text{waste}_V^{(\bar{q}_*)} + \text{waste}_H^{(\bar{q}_*)}.
\end{aligned}$$

  The two side wastes are at most $u_0 + v_0 - 1 \leq 1$. If $\epsilon = 0$, both are even and hence zero, as is the border waste of color $\bar{q}_*$. Thus $e_V = e_H = z$, and $u_0 + v_0 \geq 1$. The color q_* then has border waste 2 and side wastes $u_0 + v_0 - 1$. If $\epsilon = 1$, both are odd and hence one. This forces $u_0 = v_0 = 1$, $e_V = e_H = z - 1$, and zero waste of color q_* on all three sets. For $d_S = 1$ the signed count difference is $|T_{\bar{q}_*}| - 1 = |S_{\bar{q}_*}| = e_V = e_H$. When $d_T = 1$, the roles of T and S are exchanged.

5.5. Square frames with zero deficits

Proposition 5.1 (Branch parameters for square frames)**.** *Suppose $w = h$ and all four deficits are zero.*

(a) ***Odd** w **and** k. The color counts are $(k-1)/2$ and $(k+1)/2$. The smaller color has its box diagonal occupied and zero waste on the border and both sides. If a case symmetry exchanges colors, choose color zero as the smaller color.*

(b) ***Odd** w**, even** k. Both color counts are $k/2$. For each color q, $\text{waste}_V^{(q)} = \text{waste}_H^{(q)} = \eta_q$ and $\text{waste}_F^{(q)} = 2$. When $\eta_q = 1$, the border waste is the sum of the two side wastes. Branch on (u_0, v_0), dropping $(0,1)$ if a case symmetry exchanges the two bits.*

(c) ***Even*** w. *Let* $\Delta_V = z_{\bar{q}_*} - e_{1-p_c}(R)$ *and* $\Delta_H = z_{\bar{q}_*} - e_{1-p_a}(C)$. *Both lie in* $\{0,1\}$, *with* $u_0 + v_0 \geq 2\max(\Delta_V, \Delta_H)$. *This gives seven parameter tuples* $(\Delta_V, \Delta_H, u_0, v_0)$ *before the monochromatic refinement:*

$$(0,0,0,0),\ (0,0,1,0),\ (0,0,0,1),\ (0,0,1,1),\ (1,0,1,1),\ (0,1,1,1),\ (1,1,1,1).$$

If g_4 *or* g_7 *stabilizes the case, keep* $(1,0,1,1)$ *as the representative of* $(0,1,1,1)$. *Each side has zero waste in at least one color. The border has zero waste in color* $\bar{q}_*$ *if both offsets are zero, and in color* q_* *if both are one.*

Proof. For odd w, each empty row contributes one square of each color to V, and each empty column does the same on H. The two corners of each color are shared, giving $N_q(V) = N_q(H) = k$ and $N_q(\mathrm{Bd}) = 2k-2$. Thus the side inequalities become $2z_q + \eta_q \geq k$. When k is odd, they force the counts in (a) and $\eta_q = 1$ for the smaller color. Substituting into (2)–(4) gives zero wastes for the smaller color. For the larger color it gives border waste 4 and side wastes $1+\eta_q$. When k is even the inequalities force the equal counts in (b), so the side wastes equal η_q and the border waste is 2.

For even w, the box diagonals have color q_*. The vertical side contains $2e_{1-p_c}(R)$ needy squares of color $\bar{q}_*$. The two side equations and the total waste give

$$\mathrm{waste}_X^{(\bar{q}_*)} = 2\Delta_X, \qquad \mathrm{waste}_X^{(q_*)} = u_0 + v_0 - 2\Delta_X, \qquad X \in \{V, H\}.$$

Both wastes are nonnegative, giving the stated bounds on the offsets and the zero wastes on the sides. The border wastes are $2(\Delta_V + \Delta_H)$ in color $\bar{q}_*$ and $4 - 2(\Delta_V + \Delta_H)$ in color q_*. The symmetry actions in Section 2 exchange colors and box-diagonal indicators in (a) and (b), respectively. In (c), g_4 and g_7 exchange the offsets and fix the box diagonals. □

5.6. Monochromatic frames and diagonal moments

Weakley [13, Proposition 13] gives the row and column structure of monochromatic dominating sets. The following lemma identifies the corresponding frame in the case enumeration.

Lemma 5.2 (Uniqueness of the monochromatic frame)**.** *Suppose* $w = h$ *is even,* $d_R = d_C = 0$, $d_T + d_S \leq 1$, *and* $z_{\bar{q}_*} = 0$. *In the canonical frame, the margins are* $(0,1,0,1)$, *the empty rows and columns have indices* $0, 2, \ldots, n-2$, *and the queens are at* $(2i+1, 2j+1)$ *with* $0 \leq i, j < k$ *and* $j = \pi(i)$ *for a permutation* π.

Proof. For zero diagonal deficits, the offsets in Proposition 5.1(c) are nonnegative. With one diagonal deficit, the repeated diagonal has color q_* and the opposite-color empty-line counts equal $z_{\bar{q}_*}$. In either case $z_{\bar{q}_*} = 0$ forces $e_{1-p_c}(R) = e_{1-p_a}(C) = 0$. The k empty lines of each family are therefore the entire even or odd parity class. Each margin pair is $(0,1)$ or $(1,0)$. Reflecting the board gives the canonical tuple $(0,1,0,1)$, so $R = C = \{1, 3, \ldots, 2k-1\}$. The zero row and column deficits give the permutation. □

On this $k \times k$ grid, set $t' = j - i$ and $s' = i + j + 1$. Write T', S' for the occupied indices in $[-(k-1), k-1]$ and $[1, 2k-1]$, respectively.

The moment identities of Östergård and Weakley [9, Theorem 4 and Eq. (3)], also used by Weakley [13, Eqs. (1)–(2)], take the following form here.

$$d_T = 0 \implies \sum_{t' \in T'} t' = 0, \qquad d_S = 0 \implies \sum_{s' \in S'} s' = k^2,$$

$$d_T = d_S = 0 \implies \sum_{t' \in T'} (t')^2 + \sum_{s' \in S'} (s'-1)^2 = 4 \sum_{i=0}^{k-1} i^2 = \frac{2}{3} k(k-1)(2k-1).$$

The two linear identities follow by summing $j-i$ and $i+j+1$ over the permutation. With both diagonal deficits zero, each summand occurs once in its occupied family, and $(j-i)^2+(i+j)^2 = 2(i^2+j^2)$ gives the quadratic identity. With one diagonal deficit, we use the linear identity for the family with deficit zero. The implementation imposes the available identities modulo small integers; see Section 6.

Remark 5.3. The incidence identities of Theorem 3.1 apply to odd boards as well. The subsequent branch enumeration and the computations reported here use $n = 2k$.

6. SAT encoding and completeness

For each branch we build a Boolean formula in conjunctive normal form (CNF). Its clauses require a placement that dominates the board and satisfies the branch constraints. We first bound the size of the counters used in those constraints.

Lemma 6.1 (Bounds for the parity counters)**.** *Write $z = z_{\bar{q}_*}$ in an even-by-even frame. The parity equalities of Section 5 admit the following bounds.*

(a) For a nonsquare frame, $z \le k-2$.

(b) For a square with one row or column deficit, $z \le k-1$.

(c) For a square with one diagonal deficit, $z \le k-2$ when $q_{\mathrm{def}} = q_*$, and $z-1 \le k-2$ otherwise.

(d) For a square with zero deficits, $z \le k-2+\max(\Delta_V, \Delta_H)$.

Proof. A side of even length ℓ has exactly $\ell/2$ interior line indices of parity opposite to its endpoints. In (a) the shorter side is at most $2k-4$, and the number of empty lines of that parity on the shorter side equals z. In (b), the corresponding equality on the side with zero waste in both colors gives a bound of $w/2 \le k-1$. In (c), a repeated diagonal of color q_* uses two queens of that color. If the repeat is of the other color, both box diagonals are occupied, so at least one queen has color q_*.

For (d), a needy corner requires at least one queen of color q_*. This gives $z \le k-1$ and suffices when either Δ is positive. When both vanish and $w \le 2k-4$, the side count suffices. The remaining frame has $w = 2k-2$ and, up to symmetry, margins $(0,1,0,1)$. If $z = k-1$, all $k-1$ odd interior rows and columns are empty, leaving a single available odd index $2k-1$ in each direction. The queens in color $\bar{q}_*$ use exactly one odd coordinate each. At most two fit in distinct rows and columns. For $k \ge 4$ this contradicts $z = k-1$. For $k = 2$ both even indices are empty boundaries, making any such queen impossible. For $k = 3$, the only available even index is 2 in either direction. The two queens would be $(5,2)$ and $(2,5)$ on the same sum diagonal, contrary to zero deficit. □

6.1. Board and auxiliary variables

The Boolean variable Q_{xy} records whether a queen occupies an available square (x, y). Squares on the four empty boundary lines or on diagonals excluded by a zero-waste constraint are unavailable. The variables $o_R(y)$, $o_C(x)$, $o_T(t)$, and $o_S(s)$ record whether a row, column, difference diagonal, or sum diagonal contains a queen. Each is the disjunction of its square variables, and is false when the line has no available square.

For each square in the closed frame, a coverage clause requires at least one of its four line variables to be true. Unit clauses force the exterior rows and columns to be occupied, covering all squares outside the frame.

The four occupancy counts are $k-d_R, k-d_C, k-d_T, k-d_S$. When the diagonal counts are specified by color, the two counts add to the same total. For each family with deficit zero, the encoder allows at most one queen per line. Such a family always exists, and its k occupied lines force exactly k queens. The branch fixes counts of empty lines by parity, imposes the relevant box-diagonal occupancies, and bounds the wastes. In an even-sided zero-deficit square, if a box diagonal is empty, the other diagonal at each of its corners must cover that corner.

The parity counters use the caps in Lemma 6.1, tightened by the lengths and offsets of the corresponding lists.

At a border square p, let N_p indicate that the square is non-needy. It is the row occupancy on a vertical side, the column occupancy on a horizontal side, and zero at a corner. Write o_L for the occupancy of diagonal L. The waste constraints count the following literals, with weights where indicated.

- Missed intersections contribute o_L with weight $2 - \beta_{\mathrm{Bd}}(L)$ for the border or $[\beta_X(L) = 0]$ for a side set X covered by Theorem 3.1.
- A hit on a non-needy square contributes $E_{L,p}$, forced true by $o_L \wedge N_p$.
- A double hit on a needy square contributes F_p, forced true by $o_T \wedge o_S \wedge \neg N_p$ at p.

A placement extends by assigning $E_{L,p}$ and F_p their conjunction values. For a zero-waste bound the encoder forbids each corresponding event directly. It omits those same per-square events from the border count if they were already forbidden on a side. To encode a bound such as $\mathrm{waste}_V^{(q)} \le u_0 + v_0 - 1$, it adds $1 - u_0$ and $1 - v_0$ to the count and bounds the sum by one.

Cardinality constraints use unary totalizers [1]. A bit at index i marks the threshold of i true literals in a group. When merging two groups, the forward clauses propagate true child thresholds i and j to threshold $i + j$ of the combined count. A zero threshold is always true. At an upper bound B, clauses forbid the first overflow $i + j = B + 1$. Inductively any count exceeding B violates a clause.

When two counts must differ by a specified offset, reverse clauses also make the parent thresholds false above the true count. The two threshold vectors can then be equated after shifting by the offset. For an exact constant count, an upper bound on the negated literals supplies the matching lower bound. At-most-one constraints along a line use pairwise exclusion or a sequential counter [12], whose prefix bit indicates whether an earlier prefix contains a queen. Assigning bits from the actual counts satisfies all these clauses.

6.2. Monochromatic formulas

For the frame in Lemma 5.2, the reduced encoder places one queen in every row and column of the odd-odd $k \times k$ grid. It indexes diagonal occupancy on color-zero squares by $t' = t/2$ and $s' = s/2$.

The only needy squares are $(2u, 2v)$ with $0 \le u, v < k$. At each of them the coverage clause is $o_T(v-u) \vee o_S(u+v)$. Since $o_S(0) = 0$, the corner $(0,0)$ forces $o_T(0) = u_0 = 1$.

With one diagonal deficit, (5)–(7) give border waste 2 and side waste $v_0 = o_S(k-1)$ on each side set. The diagonals $|t'| = k-1$ and $s' = 2k-2$ meet the frame border once, and $s' = 2k-1$ misses it. All other allocated diagonals meet the border twice and hit both side sets. All border squares of the queen color are needy, so the reduced waste bounds use only occupancy and double-coverage bits.

The encoder imposes the moment identities modulo $3, 4, 5, 7, 8$ and, for zero deficit, also modulo 9. At each partial sum it uses one bit per possible residue. The clauses start the residue at zero, add the next term's weight if its literal is true, and require the final residue to equal the target. A permutation satisfying the identities gives the required residues by direct summation. The transpose sends t' to $-t'$ and fixes s', allowing a lexicographic comparison of the difference-diagonal vectors.

6.3. Coverage of the generated formulas

Theorem 6.2. *For $n = 2k \ge 4$, every dominating k-set has a dihedral image whose queen and occupancy indicators extend to a satisfying assignment of a generated formula. Conversely, every satisfying assignment of a generated formula gives a dominating set of size k on Q_n.*

Proof. Take a minimum case of the placement under the eight maps of Section 2. The generator enumerates nonnegative margin tuples satisfying (1) and $w, h \ge 1$, and deficit tuples with each entry at most $k-1$ and sum at most two. Its side tests are the inequalities (6) and (7) with $u_0 + v_0$ replaced by its upper bound 2. Hence the minimum case is enumerated. Lemmas 4.1–4.4 show that no algebraic pruning rule discards it.

The actual parity counts and occupancy bits satisfy one of the branch descriptions. A color-flipping case stabilizer exchanges z_0 and z_1, exchanges the smaller color on an odd-sided zero-deficit square when k is odd, and exchanges the two box diagonals there when k is even. These are precisely the identifications used to omit a representative in those branches. Transposition and antitransposition interchange Δ_V, Δ_H in an even-sided square and fix the box diagonals individually, justifying the omission of $(0,1,1,1)$ when one of those maps stabilizes the case.

For the two monochromatic patterns the conditions of Lemma 5.2 force the special frame $(0,1,0,1)$. The reduced encoder covers all such placements. The general branch at that frame for a diagonal deficit in q_* and the zero-deficit branch $(0,0,1,1)$ both require $z_{\bar{q}_*} > 0$, since their zero-count placements are covered by the reduced formula. Other zero-deficit branches may overlap with it.

Within a general formula the encoder compares the combined row and column occupancy vector with its images under selected members of $\mathrm{Stab}(\kappa)$. Color-reversing maps are kept only when their action preserves the branch parameters. In the even-sided square, an exchange of sides is kept only when it preserves the fixed deficits and the ordered pair (Δ_V, Δ_H) when present. This follows from the eight actions on lines above. The single-row-or-column-deficit even square uses only the half-turn. In each comparison a prefix bit is forced when the vectors agree up to that entry. The bit then activates the inequality at the next entry. A placement with lexicographically least occupancy vector in the orbit under the generated

subgroup satisfies every comparison. Its prefix bits can be set from the two vectors. The reduced formula uses the same argument with transpose.

The resulting placement satisfies the coverage and cardinality constraints. The branch restrictions follow from the incidence identities, Lemma 6.1, and the monochromatic moment identities where applicable. Its actual occupancy, waste counts, counter thresholds, and residues provide the auxiliary assignment. Conversely, a satisfying assignment defines a dominating set by the occupancy and coverage clauses, with exactly k queens by the zero-deficit family or the reduced permutation constraints. □

7. Computations and verification

7.1. Certified runs and audit records

All 290 542 formulas generated for the eight boards were unsatisfiable. For each formula, the worker wrote a DIMACS file, computed its SHA-256 digest, and requested a binary DRAT proof from CaDiCaL [2]. The unmodified `drat-trim` checker [14] accepted every proof in backward mode. The counts and timings are given in Table 1.

Table 1. Certified runs on one ten-core Apple M4 with ten workers and `--config auto`. Solve and check times sum elapsed seconds across workers. Wall time measures the worker phase. Timing totals sum the reported board times. Proof volumes are in decimal gigabytes. Proof-volume totals are calculated from unrounded byte counts.

n	k	Canonical	Pruned (%)	Leaves	Solve (s)	Check (s)	Wall (s)	DRAT (GB)
50	25	54 626	44 288 (81.1)	19 090	1 799.29	2 703.82	492.68	2.69
52	26	63 119	43 354 (68.7)	30 814	5 539.94	5 434.66	1 170.76	6.21
54	27	72 268	59 164 (81.9)	24 251	3 512.24	4 388.23	856.67	4.35
56	28	82 695	57 564 (69.6)	39 144	7 947.76	6 673.64	1 551.10	9.53
58	29	93 870	77 479 (82.5)	30 409	5 272.34	5 162.29	1 434.49	7.23
60	30	106 504	74 991 (70.4)	48 856	13 191.39	9 754.24	2 418.80	14.67
62	31	119 984	99 772 (83.2)	37 571	8 719.97	8 275.18	5 020.18	13.09
64	32	135 116	96 115 (71.1)	60 407	17 674.78	13 420.81	3 270.29	22.58
Total		728 182	552 727 (75.9)	290 542	63 657.71	55 812.87	16 214.97	80.34

The algebraic rules in Section 4 excluded 552 727 of the 728 182 canonical cases (75.9%). Table 2 gives the count by rule. The remaining cases can each give more than one formula because of the branches in Section 5.

Each log records the formula tag and digest, the solver verdict, the proof byte count, and the checker verdict. The Python auditor `scripts/verify_certificate.py` separately enumerates the canonical cases, counts by pruning rule, and expected branch tags. It checks that each expected tag appears exactly once, reconciles the summary counts with the records, checks the logged verdicts, and can verify upper-bound witnesses from their coordinates. All 290 542 recorded DIMACS SHA-256 digests match regeneration with the pinned encoder, including all 74 155 formulas for $n = 50, 52, 54$.

The raw proofs, totaling 80.34 GB, were deleted after checking. The driver keeps temporary files on verification failure or an unexpected satisfiable result. The archived JSONL files are audit records. To verify the proofs afresh, regenerate them from the pinned code in Section 8.

Table 2. Canonical cases excluded by each algebraic rule. The generator counts each excluded case under the first rule it fails.

n	NS-1	NS-2	NS-3	SQ-RC-Odd
50	15 064	24 674	3 731	819
52	43 354	0	0	0
54	20 265	32 879	5 005	1 015
56	57 564	0	0	0
58	26 684	42 975	6 580	1 240
60	74 991	0	0	0
62	34 544	55 232	8 500	1 496
64	96 115	0	0	0

7.2. Timing and resource use

For the eight runs in Table 1, ten workers on one Apple M4 used CaDiCaL 3.1.0 (commit `e4d9bd93`). The checker `drat-trim` (commit `2e3b2dc0`) reads binary proofs with options `-w -t 1000000`. The driver's `auto` mode selects CaDiCaL's `unsat` preset for branch kinds 1, 2, 5, 7, 8, 11, and 12, and the default preset otherwise.

The driver uses `std::chrono::steady_clock`. The solve column times the `solve()` calls, including DRAT output during each call. The check column times checker invocations from startup to exit. Wall time runs from the start of the worker phase through encoding, file operations, solving, and checking. Case enumeration and ordering precede that phase. Solve and check times accumulate across workers and can therefore exceed wall time.

For $n = 56, 58, 60, 62, 64$, the script `scripts/run_certified_board.py` ran the boards in order. It sampled free storage, resident set size (RSS), swap usage, macOS VM pressure, and temporary file size every 0.5 s. The monitor was set to suspend cleanly below 8 GiB of free storage or at critical VM pressure. All five boards finished without suspension, and free storage stayed above 21 GiB.

Table 3. Largest proof per board and recorded resource peaks. RSS sums the parent driver and its active checker processes. Temporary disk holds the active CNF and DRAT files. Dashes indicate runs without resource monitoring. The leaf tags omit the common frame prefix $(0, 1, 0, 1)$ and suffix `_0_0_0_0_0`.

n	Peak RSS (MiB)	Peak temp (MB)	Min free disk (GiB)	Largest proof leaf (branch on $(0, 1, 0, 1)$)	Leaf solve (s)	Leaf check (s)	Leaf DRAT (MB)
50	—	—	—	`0_0_0_1_b8`	33.96	33.66	33.65
52	—	—	—	`0_0_1_1_b3`	23.12	21.57	29.83
54	—	—	—	`0_0_0_1_b8`	155.18	234.21	133.25
56	929.2	208.0	21.59	`0_0_1_1_b3`	32.59	40.14	73.53
58	1 769.5	683.0	21.16	`0_0_0_1_b8`	937.76	496.59	671.15
60	1 335.3	519.2	21.07	`0_0_1_1_b3`	267.03	326.73	369.37
62	4 896.0	3 066.6	21.06	`0_0_0_1_b8`	2 475.08	2 544.16	3 066.20
64	3 407.8	1 953.9	22.04	`0_0_1_1_b3`	1 185.09	1 753.70	1 830.66

The largest proofs all came from the frame $(0, 1, 0, 1)$. For $n \equiv 2 \pmod 4$, the largest leaf was the one-diagonal-deficit monochromatic branch `b8`. On Q_{62} its proof took about 5 019 s to

produce and check, almost all of the 5 020.18 s worker wall time. The other 37 570 leaves used a combined 6 244.89 s of solve time and 5 731.02 s of check time across workers. The largest proof occupied 3 066.20 MB, close to the 3 066.6 MB peak for temporary files, and peak RSS was 4 896 MiB.

For $n \equiv 0 \pmod 4$, the largest leaf was the two-diagonal-deficit branch `b3`, which includes placements with either one or two colors. Its proof grew from 29.83 MB at $n = 52$ to 1 830.66 MB at $n = 64$. The zero-deficit monochromatic leaf `b12`, which uses the quadratic moment identity, took less than 0.75 s to solve and at most 1.14 s including checking on each tested board.

The worker queue gives priority to the $(0, 1, 0, 1)$ leaves in branches 8, 3, and 12. This schedules expensive proofs alongside the bulk of the queue. The large `b3` leaf finished before the end of the worker phase on both Q_{60} and Q_{64}.

7.3. Same-machine benchmarks on Q_{26}

To measure certification overhead on one Apple M4 worker, we selected up to 12 leaves per branch type in order of SHA-256 tag digest, for a total of 99 leaves. We used the pinned tools and the `auto` configuration. We ran each mode three times and alternated which mode went first in each pair. Median worker wall time was 0.33 s without proofs and 3.35 s with proof generation and checking. The sum of solver-call times had medians of 0.18 s and 0.19 s, respectively. In proof mode the sum of checker times had a median of 2.88 s, including one invocation per leaf. All 99 proofs passed in each of the three runs.

To measure the effect of algebraic pruning, we timed a separate benchmark on the full set of Q_{26} cases. Both modes ran on one Apple M4 with ten workers, the pinned solver and checker, and identical solver and symmetry settings. Only the algebraic pruning pass was toggled. The wall-clock measurement covered the complete command, from case enumeration and CNF construction through proof generation and checking. Across three runs per mode in alternating order, pruning reduced the formula count from 7 668 to 2 773 and median wall time from 58.49 s to 24.39 s, a 2.40× speedup on Q_{26}. All proofs passed in all six runs.

7.4. Structural results for larger boards

The DRAT checks reported here use the encoding and coverage proof in Sections 2–6. For larger boards one could use Weakley's results to exclude whole color classes before encoding them [13, Proposition 11, Theorems 17–18 and Section 3]. For $n \equiv 0 \pmod 4$, the only board sizes permitting a monochromatic dominating $n/2$-set are 4 and 12. For $n = 2k > 2$ with k odd, use Weakley's standard orientation, with coordinates $1 - k, \ldots, k$ and all occupied rows and columns of the same parity as k. In this orientation there are no excess difference-diagonal indices and there are two excess sum-diagonal indices, counted with multiplicity.

Let d_{W} be half the absolute difference of the two excess sum indices and $e_{\mathrm{W}} = |s_0 - s_2|/2$, where s_r is the least absolute value of a missing sum index in residue class r modulo four. Weakley's necessary condition is

$$d_{\mathrm{W}}^2 + (k-1)e_{\mathrm{W}}^2 = \frac{k(k^2+2)}{3}, \qquad d_{\mathrm{W}}, e_{\mathrm{W}} \in \{1, 3, \ldots, k\}.$$

For $2 < n < 238$, the equation leaves $n = 6, 10, 18, 30, 38, 54, 130, 142, 234$. Proposition 19 excludes $38, 142$ ($d_{\mathrm{W}} = e_{\mathrm{W}}$), and the negative $F(k)$ bounds in Weakley's Section 3 exclude $54, 234$. Taken together, these criteria rule out monochromatic dominating $n/2$-sets on every

even board from 50 through 128. On those boards an encoder using the criteria could omit monochromatic leaves such as `b8` and require both colors in `b3`.

For bichromatic sets with $k > 2$, Weakley [13, Proposition 11] gives the count pair $\{z_0, z_1\} = \{\lfloor k/2 \rfloor, \lceil k/2 \rceil\}$, with the additional pair $\{k/2-1, k/2+1\}$ possible only if k, w, h are even and $d_R = d_C = 0$. This would restrict the largest `b3` leaf on Q_{64} to $z_0 \in \{15, 16, 17\}$. For nonsquare even-by-even frames with $w > h$, one can intersect those count bounds with $z_{\bar{q}_*} \leq h/2$. The margin sum in Lemma 4.1(ii) and $w \leq 2k - 2$ already give $h \geq k$.

8. Code and data availability

The encoder, checker wrapper, runner, logs, diagnostics, and witness coordinates are available at `https://github.com/CosmicCrusader23/queens`. The pinned README gives the full build, audit, regeneration, and benchmark commands.

- Reproduction commit `61e8df4438cad8837f4498ad4160394028ca7232`.
- Earlier Q_{50}, Q_{52}, Q_{54} snapshot `7872bd98a9d1d72d463c6b967717bd7ab093b7d8`.
- CaDiCaL 3.1.0 commit `e4d9bd93af444c6575a908a24da2c4130cb2a1c7`,
- `drat-trim` commit `2e3b2dc0ecf938addbd779d42877b6ed69d9a985`.

The runs used a ten-core Apple M4 (arm64, 16 GB RAM) running macOS 26.5.2 (Darwin 25.5.0) with Apple Clang 17.0.0, C++20, and Python 3.14.0. The Python scripts use only the standard library.

After the dependency setup and `make -j4`, the following commands regenerate and check the Q_{56} proofs in a fresh output directory.

```
mkdir -p regeneration
python3 scripts/run_certified_board.py --n 56 --jobs 10 \
  --out regeneration/cert_n56_k28.jsonl \
  --diag-out regeneration/diagnostics_n56_k28.json
```

For other sizes in Table 1, change `--n` and the output filenames. A direct driver invocation uses the same board size and `--jobs 10 --certify --config auto`. Table 1 gives the expected leaf counts and proof volumes. If retaining proofs with `--proof-dir`, budget for the full per-board volume. The monitor above runs with temporary proof files.

For an audit of the archived logs, use the driver's `--fingerprints` option and pass the output and the matching JSONL log to `scripts/verify_certificate.py` with `--fingerprints` and `--cert`, respectively, along with the board size in `--n`. The same auditor can check the placements directly with `--witness-json results/even_upper_witnesses.json`.

9. AI contributions

Gemini 4 Neon, used through Google Antigravity, developed the mathematical reduction, wrote the C++ and Python code, ran the computations and checks, and drafted the manuscript. OpenAI Codex reviewed the proofs and encoder, checked reproduction of the computations, and helped revise the manuscript. The author initiated and coordinated the project and reviewed the computation logs and verification outputs.

A. Upper-bound placements

Write $(x, y) = (\text{column}, \text{row})$, starting at zero. The search in `src/witness_sa.cpp` placed queens on a permutation of the odd rows and columns plus one extra square. The verifier checks the size and coverage of every placement below. It also checks distinct rows, columns, and diagonals for the independent placements.

A.1. Even boards from 50 through 64

These placements give the upper bounds in Theorem 1.1. For Corollary 1.2 we separately checked the eight independent placements of Östergård and Weakley [9, Section 5, p. 11] in their original coordinates. For Q_{54}, their listed queens dominate $[-27, 26] \times [-26, 27]$, which adjoins column -27 and row 27 to Q_{53}. Their Q_{58} construction adjoins row and column -29 to Q_{57} and adds $(17, -29)$, giving the board $[-29, 28]^2$.

Q_{50} (26 queens)
$(0,0)$, $(1,3)$, $(3,35)$, $(5,25)$, $(7,15)$, $(9,9)$, $(11,39)$, $(13,49)$, $(15,19)$, $(17,27)$, $(19,11)$, $(21,45)$, $(23,29)$, $(25,1)$, $(27,21)$, $(29,41)$, $(31,47)$, $(33,23)$, $(35,7)$, $(37,17)$, $(39,37)$, $(41,5)$, $(43,31)$, $(45,13)$, $(47,43)$, $(49,33)$.

Q_{52} (27 queens)
$(0,0)$, $(1,15)$, $(3,29)$, $(5,43)$, $(7,37)$, $(9,41)$, $(11,45)$, $(13,11)$, $(15,7)$, $(17,19)$, $(19,1)$, $(21,25)$, $(23,31)$, $(25,3)$, $(27,49)$, $(29,39)$, $(31,27)$, $(33,51)$, $(35,5)$, $(37,23)$, $(39,33)$, $(41,47)$, $(43,9)$, $(45,35)$, $(47,21)$, $(49,17)$, $(51,13)$.

Q_{54} (28 queens)
$(0,0)$, $(1,13)$, $(3,53)$, $(5,25)$, $(7,15)$, $(9,37)$, $(11,23)$, $(13,45)$, $(15,51)$, $(17,1)$, $(19,7)$, $(21,17)$, $(23,47)$, $(25,27)$, $(27,21)$, $(29,35)$, $(31,29)$, $(33,49)$, $(35,39)$, $(37,5)$, $(39,11)$, $(41,3)$, $(43,19)$, $(45,9)$, $(47,31)$, $(49,41)$, $(51,43)$, $(53,33)$.

Q_{56} (29 queens)
$(0,0)$, $(1,17)$, $(3,23)$, $(5,45)$, $(7,51)$, $(9,33)$, $(11,43)$, $(13,19)$, $(15,7)$, $(17,13)$, $(19,55)$, $(21,31)$, $(23,21)$, $(25,53)$, $(27,29)$, $(29,5)$, $(31,39)$, $(33,27)$, $(35,47)$, $(37,1)$, $(39,11)$, $(41,25)$, $(43,3)$, $(45,49)$, $(47,15)$, $(49,37)$, $(51,41)$, $(53,9)$, $(55,35)$.

Q_{58} (30 queens)
$(0,0)$, $(1,35)$, $(3,37)$, $(5,15)$, $(7,21)$, $(9,39)$, $(11,5)$, $(13,57)$, $(15,53)$, $(17,7)$, $(19,27)$, $(21,23)$, $(23,41)$, $(25,51)$, $(27,31)$, $(29,25)$, $(31,1)$, $(33,55)$, $(35,45)$, $(37,29)$, $(39,13)$, $(41,19)$, $(43,49)$, $(45,11)$, $(47,3)$, $(49,47)$, $(51,33)$, $(53,9)$, $(55,17)$, $(57,43)$.

Q_{60} (31 queens, independent)
$(0,0)$, $(1,45)$, $(3,39)$, $(5,15)$, $(7,27)$, $(9,13)$, $(11,43)$, $(13,3)$, $(15,11)$, $(17,57)$, $(19,47)$, $(21,9)$, $(23,37)$, $(25,31)$, $(27,25)$, $(29,53)$, $(31,17)$, $(33,35)$, $(35,29)$, $(37,1)$, $(39,55)$, $(41,49)$, $(43,19)$, $(45,5)$, $(47,59)$, $(49,21)$, $(51,7)$, $(53,33)$, $(55,23)$, $(57,41)$, $(59,51)$.

Q_{62} (32 queens)
$(0,0)$, $(1,15)$, $(3,21)$, $(5,47)$, $(7,33)$, $(9,19)$, $(11,9)$, $(13,43)$, $(15,55)$, $(17,45)$, $(19,57)$, $(21,39)$, $(23,27)$, $(25,59)$, $(27,5)$, $(29,25)$, $(31,17)$, $(33,41)$, $(35,1)$, $(37,29)$, $(39,61)$, $(41,3)$, $(43,37)$, $(45,51)$, $(47,11)$, $(49,23)$, $(51,53)$, $(53,35)$, $(55,13)$, $(57,7)$, $(59,49)$, $(61,31)$.

Q_{64} **(33 queens)**
$(0,0)$, $(1,41)$, $(3,23)$, $(5,53)$, $(7,43)$, $(9,25)$, $(11,7)$, $(13,17)$, $(15,59)$, $(17,5)$, $(19,11)$, $(21,31)$, $(23,37)$, $(25,57)$, $(27,51)$, $(29,9)$, $(31,33)$, $(33,61)$, $(35,21)$, $(37,35)$, $(39,29)$, $(41,49)$, $(43,55)$, $(45,1)$, $(47,19)$, $(49,13)$, $(51,3)$, $(53,47)$, $(55,15)$, $(57,63)$, $(59,27)$, $(61,45)$, $(63,39)$.

A.2. Earlier independent values

Theorem A.1 (Neuhaus [8, Section 2.4])**.** *The independent domination numbers satisfy*

$$i(Q_{32}) = 17, \quad i(Q_{34}) = 18, \quad i(Q_{38}) = 20, \quad i(Q_{40}) = 21, \quad i(Q_{44}) = 23.$$

Proof. The matching ordinary domination numbers come from *Hint and Conquer* [11] for $n = 32, 34, 38$ and Davies [4, Theorem 1.3 and Table 2] for $n = 40, 44$. The independent placements below attain these lower bounds. □

Q_{32} **(17 queens)**
$(1,25)$, $(2,2)$, $(3,19)$, $(5,9)$, $(7,27)$, $(9,1)$, $(11,17)$, $(13,11)$, $(15,3)$, $(17,29)$, $(19,21)$, $(21,15)$, $(23,31)$, $(25,5)$, $(27,23)$, $(29,13)$, $(31,7)$.

Q_{34} **(18 queens)**
$(0,32)$, $(1,9)$, $(3,19)$, $(5,25)$, $(7,31)$, $(9,5)$, $(11,17)$, $(13,11)$, $(15,3)$, $(17,29)$, $(19,21)$, $(21,15)$, $(23,27)$, $(25,1)$, $(27,7)$, $(29,13)$, $(31,23)$, $(33,33)$.

Q_{38} **(20 queens)**
$(1,19)$, $(3,11)$, $(4,34)$, $(5,27)$, $(7,5)$, $(9,35)$, $(11,13)$, $(13,17)$, $(15,1)$, $(17,31)$, $(19,15)$, $(21,21)$, $(23,33)$, $(25,3)$, $(27,9)$, $(29,23)$, $(31,37)$, $(33,7)$, $(35,25)$, $(37,29)$.

Q_{40} **(21 queens)**
$(0,38)$, $(1,35)$, $(3,19)$, $(5,13)$, $(7,7)$, $(9,1)$, $(11,23)$, $(13,37)$, $(15,29)$, $(17,15)$, $(19,39)$, $(21,5)$, $(23,17)$, $(25,27)$, $(27,3)$, $(29,33)$, $(31,11)$, $(33,21)$, $(35,31)$, $(37,9)$, $(39,25)$.

Q_{44} **(23 queens)**
$(0,42)$, $(1,37)$, $(3,13)$, $(5,7)$, $(7,25)$, $(9,31)$, $(11,41)$, $(13,11)$, $(15,19)$, $(17,43)$, $(19,1)$, $(21,35)$, $(23,23)$, $(25,3)$, $(27,17)$, $(29,21)$, $(31,5)$, $(33,39)$, $(35,29)$, $(37,33)$, $(39,9)$, $(41,27)$, $(43,15)$.

References

[1] O. Bailleux and Y. Boufkhad, *Efficient CNF encoding of Boolean cardinality constraints*, in: Principles and Practice of Constraint Programming (CP 2003), Lecture Notes in Comput. Sci. **2833**, Springer, 2003, pp. 108–122. DOI.

[2] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, and F. Pollitt, *CaDiCaL 2.0*, in: Computer Aided Verification (CAV 2024), Lecture Notes in Comput. Sci. **14681**, Springer, 2024, pp. 133–152. DOI.

[3] E. J. Cockayne, *Chessboard domination problems*, Discrete Math. **86** (1990), 13–20. DOI.

[4] J. Davies, $\gamma(Q_{35}) = 19$ *and certified queen domination numbers up to* $n = 46$, Hexagon Math, 2026. 2611.00830v1.

[5] Aleksej Đukić, comment on $\gamma(Q_{48})$ in OEIS sequence A075458, November 22, 2026. Sequence and computation summary and witness.

[6] R. K. Guy, *Unsolved Problems in Number Theory*, 2nd ed., Springer, New York, 1994, Problem C18. DOI.

[7] jkolantree, Q_{26} *grid-annihilator formalization*, Lean development in the `BSC` repository, 2026. Project and theorem statement.

[8] S. Neuhaus, *Eigenschaften kleinster dominierender Mengen und Dominanzzahlen von Damengraphen*, Ph.D. dissertation, Universität zu Köln, 2009. University repository.

[9] P. R. J. Östergård and W. D. Weakley, *Values of domination numbers of the queen's graph*, Electron. J. Combin. **8**(1) (2001), #R29. DOI.

[10] V. Raghavan and S. M. Venkatesan, *On bounds for a board covering problem*, Inform. Process. Lett. **25**(5) (1987), 281–284. DOI.

[11] T. Rostami, *Hint and Conquer: New Queen Domination Numbers by SAT Solving*, preprint, 2026. Manuscript. Author attribution follows OEIS A075458.

[12] C. Sinz, *Towards an optimal CNF encoding of Boolean cardinality constraints*, in: Principles and Practice of Constraint Programming (CP 2005), Lecture Notes in Comput. Sci. **3709**, Springer, 2005, pp. 827–831. DOI.

[13] W. D. Weakley, *Queen domination of even square boards*, Electron. J. Combin. **29**(2) (2022), #P2.50. DOI.

[14] N. Wetzler, M. J. H. Heule, and W. A. Hunt Jr., *DRAT-trim: Efficient checking and trimming using expressive clausal proofs*, in: Theory and Applications of Satisfiability Testing (SAT 2014), Lecture Notes in Comput. Sci. **8561**, Springer, 2014, pp. 422–429. DOI.