MATHDISCOVERIES
Mathematical Preprint

Working Draft

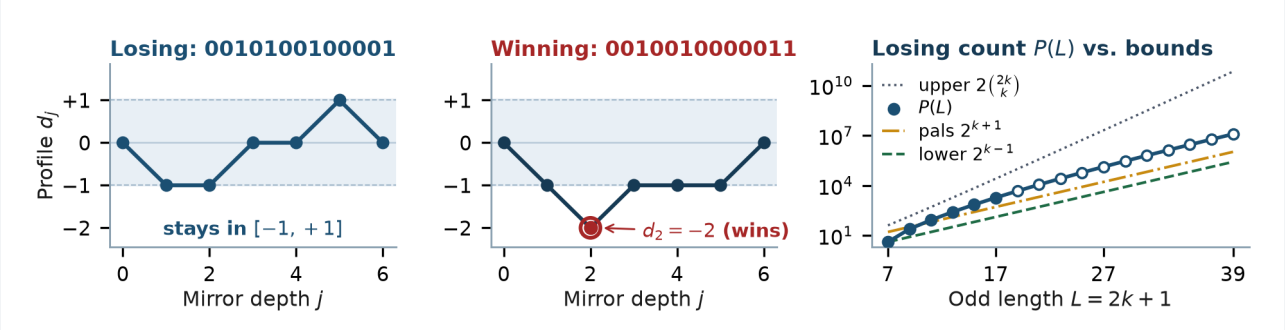
Structural Bounds and Mirror-Profile Evidence for the Palindrome Deletion Game

Tyler B. Hughes*,†

†Independent Researcher, United States

Ⓢ *Supporting Information:* `recompute.py` *(standalone Python verifier solving all* 262,142 *binary words of length* $L \le 17$*)*

Disclosure: AI-assisted working draft directed by Tyler B. Hughes. Not for submission as is. No human mathematician has yet refereed the proof.



ABSTRACT: In the palindrome deletion game on $\{0,1\}^L$, players alternate deleting one letter from a non-palindrome w; leaving a palindrome wins. For odd $L = 2k+1$ and $0 \le j \le k$, let $d_j(w)$ be the leading-j minus trailing-j 1-count. Around the unsettled conjecture that every odd losing word keeps $|d_j(w)| \le 1$, we prove four structural results and one exact verification: (1) odd $d_k(w) \neq 0$ wins and even $|d_k(w)| \ge 2$ loses, giving $P(2k+1) \le 2\binom{2k}{k} - 2^{k+1}$; (2) wrapping $v \mapsto ava$ preserves outcome, yielding $P(L) = 2P(L-2) + Q(L)$ and reducing the conjecture to $w_1 \neq w_L$; (3) lifting the four length-7 losing cores as XcX^R gives $P(2k+1) \ge 2^{k-1}$ $(k \ge 3)$; and (4) assuming the conjecture at $2k-1$, even words of length $2k$ reaching ± 3 or both $+2$ and -2 lose. Exact replay verifies worked words and all 262,142 words of length $L \le 17$; external sweeps verify odd lengths to 39 (12,222,116 losing), 41, and a re-check to 45 (204,753,080 losing), plus subgames to 63 and 127. Conjecture 6 remains unsettled; no Lean 4 formalization is claimed.

Verification Scope (G11): WARN: Human referee: pending — **Full Trust Boundary:** Section 5.
Main results: 0 of 5 in Lean Kernel (0%) · 1 Cert Only · 4 written · 1 open (not counted)
Abstract claims: 0 of 5 in Lean Kernel (0%) · 1 Cert Only · 4 written · 1 open (not counted)

MSC 2020: 91A46, 68R15, 05A05, 05A15 **Keywords:** combinatorial games, binary words, palindromes, deletion games, impartial games, prefix-suffix balance

1. INTRODUCTION AND MAIN RESULTS

Start from a binary word $w \in \{0,1\}^L$ $(L \ge 2)$ that is not a palindrome. Two players alternate turns; a move deletes a single letter of w, shortening the word by 1. The player whose move leaves a palindrome wins immediately. Because every length-1 word is a palindrome, every game ends with a winner, so every non-palindrome is either a *winning* position ($\mathcal{N}$-position, where the player to move can force a win) or a *losing* position ($\mathcal{P}$-position, where every move hands the opponent a win). This impartial game was formulated by AI research agents rather than taken from a previously printed open problem; its losing positions exhibit an unexpectedly rigid global balance constraint.

For any word $w = w_1 w_2 \dots w_L \in \{0,1\}^L$ with $k = \lfloor L/2 \rfloor$ and each $0 \le j \le k$, define the *mirror-prefix profile*

$$d_j(w) := \sum_{i=1}^{j} w_i - \sum_{i=1}^{j} w_{L+1-i} = \sum_{i=1}^{j} (w_i - w_{L+1-i}). \quad (1)$$

The sequence $d_0(w), d_1(w), \dots, d_k(w)$ starts at $d_0(w) = 0$ and takes one step in $\{-1, 0, +1\}$ per mirror pair (w_i, w_{L+1-i}), reading from the outside of the word inwards. Every palindrome satisfies $w_i = w_{L+1-i}$ for all i, so its profile is identically zero. On even lengths, losing words escape the band $[-1,1]$ immediately (at $L = 4$, the losing words 0011 and 1100 reach $d_2 = \mp 2$). On odd lengths $L = 2k+1$, however, exhaustive computation through $L = 45$ finds zero losing words whose profile ever leaves $[-1,1]$: whenever a prefix and suffix of equal length $j \le k$ differ by at least 2 in their 1-counts, the player to move appears always to win.

Let $P(L)$ denote the number of losing words in $\{0,1\}^L$, and let $Q(L)$ denote the number of losing words $w \in \{0,1\}^L$ with differing end letters $(w_1 \neq w_L)$. We collect our four proved structural results (Theorems 1–3 and Proposition 4), our standalone exact local verification (Proposition 5), and the main unsettled conjecture (Conjecture 6) below. Throughout, what is proved, what is checked locally or by external sweeps, and what remains unsettled are kept strictly separate.

Theorem 1 (Half-Weight Theorem (Case $j = k$)). *For every integer $k \geq 1$:*

(A) *Every odd-length word $w = AcB \in \{0,1\}^{2k+1}$ (with $|A| = |B| = k$ and $c \in \{0,1\}$) satisfying $d_k(w) \neq 0$ is a winning word.*

(B) *Every even-length word $w = AB \in \{0,1\}^{2k}$ (with $|A| = |B| = k$) satisfying $|d_k(w)| \geq 2$ is a losing word.*

In particular, every odd-length losing word satisfies $d_k(w) = 0$, and $P(2k+1) \leq 2\binom{2k}{k} - 2^{k+1}$ for all $k \geq 1$.

Theorem 2 (End-Stripping Invariance and Recurrence). *For every non-palindrome $v \in \{0,1\}^m$ ($m \geq 2$) and every letter $a \in \{0,1\}$, the wrapped word $w = ava \in \{0,1\}^{m+2}$ is a non-palindrome with the exact same game outcome (winning or losing) as v. Consequently:*

1. *$P(L) = 2P(L-2) + Q(L)$ for every length $L \geq 4$; and*
2. *if any odd-length counterexample to Conjecture 6 exists, a counterexample of minimal odd length must satisfy $w_1 \neq w_L$.*

Theorem 3 (Shell Lifting and Exponential Lower Bound). *Each of the four length-7 words $c \in C_7 := \{0010001, 1000100, 1101110, 0111011\}$ is losing. For every integer $k \geq 3$ and every word $X \in \{0,1\}^{k-3}$, the length-$(2k+1)$ word XcX^R is losing (where X^R denotes the reversal of X). Consequently,*

$$P(2k+1) \geq 4 \cdot 2^{k-3} = 2^{k-1} \qquad \text{for every } k \geq 3. \tag{2}$$

Proposition 4 (Conditional Even-Length Profile Companion). *Fix $k \geq 2$, and suppose Conjecture 6 holds at odd length $2k-1$. Then every even-length word $w \in \{0,1\}^{2k}$ such that either $|d_r(w)| \geq 3$ for some $1 \leq r \leq k$, or $d_r(w) \geq 2$ and $d_s(w) \leq -2$ for some $1 \leq r, s \leq k$, is a losing word.*

Proposition 5 (Exact Local Replay through Length 17). *Using only the Python standard library (dual independent solvers: memoized recursion and bottom-up bitmask dynamic programming), `recompute.py` establishes:*

1. *All 16,380 binary words of lengths $2 \leq L \leq 13$ agree identically between the two solvers, and all 262,142 words of lengths $1 \leq L \leq 17$ are solved exhaustively with 0 losing odd words violating $|d_j(w)| \leq 1$ (exact counts $P(3) = P(5) = 0$, $P(7) = 4$, $P(9) = 24$, $P(11) = 84$, $P(13) = 256$, $P(15) = 708$, $P(17) = 1{,}848$).*
2. *Worked examples: 0010001 is losing (four distinct deletions, all answered by immediate palindromes); 0000011 ($d_3 = -2$) wins by deleting any of positions 1–5 to reach the losing word 000011; 001100011 ($d_4 = 0$, $d_2 = -2$) wins uniquely via 00110011; at length 13, 0010100100001 is losing (profile touching both -1 and $+1$) while 0010010000011 ($d_2 = -2$) wins uniquely via 001010000011; and optimal play from 000100100000101 takes 13 moves.*
3. *Exact checks of Theorems 1–3 and Proposition 4 hold with 0 exceptions through $L = 17$, and the counts of balanced out-of-band words with a unique winning move at $L = 9, 11, 13, 15$ are $4, 4, 8, 16$ (all having weight or co-weight 4).*

Conjecture 6 (Mirror-Prefix Band Conjecture). *Every losing word $w \in \{0,1\}^{2k+1}$ of odd length satisfies $|d_j(w)| \leq 1$ for every $0 \leq j \leq k$. Equivalently, if an odd-length binary word has a prefix and a suffix of equal length $j \leq k$ whose numbers of 1s differ by at least 2, then the player to move can force a win.*

Figure 1 separates the proved theorems, deterministic local replay, external sweeps, and unsettled boundary. The ten-line Python check below computes game outcomes and mirror-prefix profiles directly on the key worked words (tested verbatim in `recompute.py`):

Concept in Action: Outcome and Profile Check

```
from functools import lru_cache
def pal(w): return w == w[::-1]
@lru_cache(None)
def win(w):
    return any(pal(u) or not win(u)
               for u in (w[:i] + w[i+1:] for i in range(len(w))))
def prof(w):
    d = [0]
    for j in range(1, len(w) // 2 + 1):
        d.append(d[-1] + int(w[j-1]) - int(w[-j]))
    return d
assert not win("0010001") and prof("0010001") == [0, -1, -1, 0]
assert win("001100011") and prof("001100011") == [0, -1, -2, -1, 0]
assert not win("0010100100001") and win("0010010000011")
```

Roadmap. Section 2 introduces notation, symmetries, and small examples including the full move tree of 0010001. Section 3 gives self-contained proofs of Theorems 1–3 and Proposition 4. Section 4 details the local replay (`recompute.py`), negative-mutation controls, external exhaustive sweeps through $L = 45$ and closed subgames, boundary tightness, and a cautionary sibling conjecture that held for twelve odd lengths before failing at $L = 35$. Section 5 presents the machine-generated Reader's Trust Boundary, Section 6 reviews related games, logged searches, and unread items, and Section 7 covers data availability and AI disclosure.

2. DEFINITIONS, SYMMETRIES, AND WORKED POSITIONS

Let $\Sigma = \{0,1\}$, and write a word $w \in \Sigma^L$ of length $|w| = L \geq 1$ with 1-based indexing $w = w_1 w_2 \ldots w_L$. Denote its *weight* (number of 1s) by $\mathrm{wt}(w) := \sum_{i=1}^{L} w_i$, its *co-weight* (number of 0s) by $L - \mathrm{wt}(w)$, its *reversal* by $w^R := w_L \ldots w_2 w_1$, and its *bit-complement* by $\overline{w} := (1-w_1)\ldots(1-w_L)$. For $1 \leq i \leq L$, write $w^{(-i)} := w_1 \ldots w_{i-1} w_{i+1} \ldots w_L \in \Sigma^{L-1}$ for the word obtained by deleting position i.

A word w is a *palindrome* if $w = w^R$ (every length-1 word is a palindrome). On any non-palindrome $w \in \Sigma^L$ ($L \geq 2$), the legal moves are the distinct single-letter deletions $\{w^{(-i)} : 1 \leq i \leq L\}$. Because the game is finite, impartial, and acyclic under normal play (leaving a palindrome wins immediately), the sets of winning ($\mathcal{N}$) and losing ($\mathcal{P}$) non-palindromes are uniquely defined by recursion on L:

- w is *winning* if and only if some $w^{(-i)}$ is either a palindrome or a losing word;
- w is *losing* if and only if every $w^{(-i)}$ is a winning non-palindrome.

Symmetries and profile. Both reversal $w \mapsto w^R$ and bit-complement $w \mapsto \overline{w}$ preserve palindromes and commute with single-letter deletion: $(w^R)^{(-i)} = (w^{(-(L+1-i))})^R$ and $(\overline{w})^{(-i)} = \overline{w^{(-i)}}$. Hence the Klein four-group $V_4 = \{\mathrm{id}, R, \overline{\ }, \overline{\ } \circ R\}$ preserves winning and losing status. For $k = \lfloor L/2 \rfloor$ and $0 \leq j \leq k$, the mirror-prefix profile (1) obeys $d_j(w^R) = -d_j(w)$ and $d_j(\overline{w}) = -d_j(w)$, while $d_j(\overline{w^R}) = d_j(w)$. The

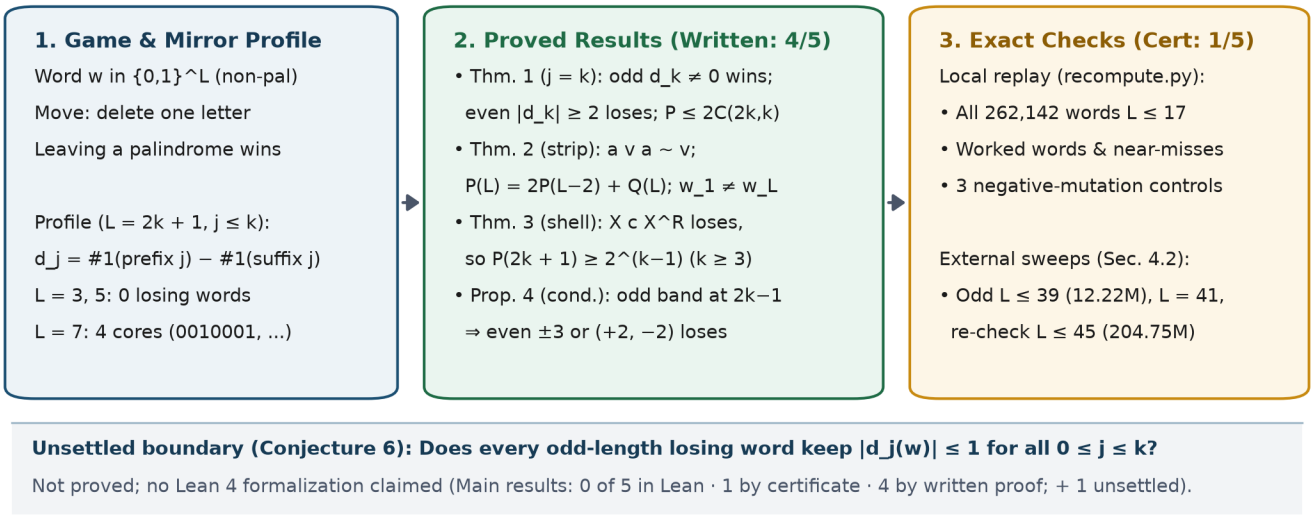


Figure 1. Discovery, proved structural results, and verification boundary for the palindrome deletion game. No Lean 4 formalization is claimed (0 of 5 proved main results in Lean 4; 1 of 5 by archived deterministic script `recompute.py`; 4 of 5 by written proof; Conjecture 6 remains unsettled outside the denominator). External exhaustive sweeps cited in Section 4 verify the band through odd length 39 (12,222,116 losing words), candidate exhaustion at 41, an independent shielded re-check through 45 (204,753,080 losing words), and closed subgames to lengths 63 and 127.

number of *mismatched mirror pairs* is $m(w) := |\{1 \le i \le k : w_i \ne w_{L+1-i}\}|$.

Small lengths and worked positions. Direct evaluation (`recompute.out`) determines all small lengths:

- *Lengths* 2–6. Every non-palindrome of length 2 (2 words), 3 (4 words), or 5 (24 words) is winning: thus $P(2) = P(3) = P(5) = 0$. At even lengths, losing words appear early: $P(4) = 2$ (0011 and 1100, whose profiles $[0,-1,-2]$ and $[0,1,2]$ already leave $[-1,1]$), and $P(6) = 22$ (out of 56 non-palindromes). From $L = 8$ onward, even losing words outnumber even winning words ($P(8) = 122$ versus 118 winning words).
- *Minimal odd losing cores* ($L = 7$). Among the 112 non-palindromes of length 7, exactly four are losing: the V_4-orbit C_7 consisting of 0010001, 1000100, 1101110, and 0111011. Figure 2 displays the full move/reply tree for 0010001: its seven single-letter deletions yield four length-6 non-palindromes (010001, 000001, 001001, and 001000), and each admits an immediate palindromic reply (10001, 00000, 00100, and 00000). Its mirror pairs are $(0,1)$, $(0,0)$, and $(1,0)$, giving profile $(0,-1,-1,0) \subset [-1,1]$.
- *Half-weight win by hand* (0000011). The length-7 word 0000011 has halves 000 and 011 (profile $(0,-1,-2,-2)$, $d_3 = -2$) and no immediate move to a palindrome. Deleting any of positions 1–5 (in particular the middle letter, position 4, as in Theorem 1) leaves the losing word 000011: if the opponent deletes a 0 (positions 1–4) to leave 00011, deleting a 0 leaves the losing word 0011; if the opponent deletes a 1 (positions 5–6) to leave 00001, deleting the remaining 1 leaves the palindrome 0000.
- *Balanced out-of-band words* ($L = 9$ *and* $L = 13$). The shortest balanced words ($d_k = 0$) whose profile escapes $[-1,1]$ occur at $L = 9$, forming one V_4-orbit: 001100011, 001110011, 110001100, and 110011100. On 001100011, the profile is $(0,-1,-2,-1,0)$, and its unique winning move deletes a 0 in positions 5–7 to leave the losing word 00110011. At $L = 13$, the earliest boundary-class losing word 0010100100001 has profile $(0,-1,-1,0,0,1,0)$ touching both -1 and $+1$ without escaping, whereas the balanced word 0010010000011 reaches $d_2 = -2$ (profile $(0,-1,-2,-1,-1,-1,0)$) and wins uniquely by deleting position 4 or 5 to leave the losing word 001010000011.

3. PROOFS OF THE STRUCTURAL RESULTS

We now prove Theorems 1–3 and Proposition 4 by elementary combinatorial induction.

Proof of Theorem 1. We prove statements (A) and (B) jointly by induction on the word length $L \ge 2$. First observe that any palindrome w satisfies $w^R = w$ and hence $d_k(w) = 0$; therefore any odd word with $d_k(w) \ne 0$ and any even word with $|d_k(w)| \ge 2$ is automatically a non-palindrome.

Base cases ($L = 2$ *and* $L = 3$). At $L = 2$ ($k = 1$), every binary word $w = w_1w_2$ satisfies $|d_1(w)| = |w_1 - w_2| \le 1$, so (B) holds vacuously. At $L = 3$ ($k = 1$), an odd word $w = acb$ with $d_1(w) = a - b \ne 0$ satisfies $a \ne b$, so $\{a,b\} = \{0,1\}$; deleting whichever of a, b differs from c leaves the palindrome cc, proving (A) at $L = 3$.

Inductive step for (B) *at even* $L = 2k \ge 4$. Assume (A) holds at odd length $2k-1$, and let $w = AB \in \{0,1\}^{2k}$ satisfy $|d_k(w)| = |\mathrm{wt}(A) - \mathrm{wt}(B)| \ge 2$, where $|A| = |B| = k$. Consider any single-letter deletion $u = w^{(-i)}$ ($1 \le i \le 2k$) of odd length $2k - 1 = 2(k-1) + 1$, and write $u = A'c'B'$ with $|A'| = |B'| = k - 1$ and middle letter $c' \in \{0,1\}$.

- If $1 \le i \le k$ (the deleted letter w_i lies in A), then $A' = A^{(-i)}$, $c' = w_{k+1}$, and $B' = w_{k+2} \dots w_{2k}$. Hence $\mathrm{wt}(A') =$

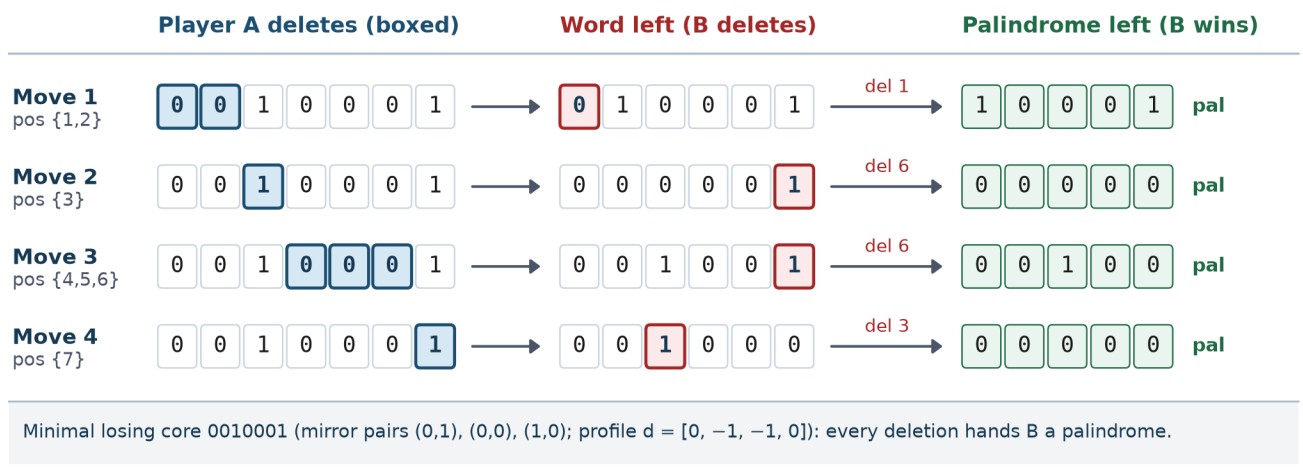


Figure 2. Complete move-and-reply verification for the minimal odd-length losing core 0010001. Deleting any letter in one of the four run-equivalent position sets $\{1,2\}$, $\{3\}$, $\{4,5,6\}$, or $\{7\}$ leaves a non-palindrome of length 6 from which Player B deletes one boxed letter to leave a length-5 palindrome immediately. Its mirror-prefix profile is $(0,-1,-1,0)$.

$\mathrm{wt}(A) - w_i$ and $\mathrm{wt}(B') = \mathrm{wt}(B) - w_{k+1}$, giving

$$d_{k-1}(u) = d_k(w) - (w_i - w_{k+1}).$$

- If $k+1 \le i \le 2k$ (the deleted letter w_i lies in B), then $A' = w_1 \dots w_{k-1}$, $c' = w_k$, and $B' = B^{(-(i-k))}$. Hence $\mathrm{wt}(A') = \mathrm{wt}(A) - w_k$ and $\mathrm{wt}(B') = \mathrm{wt}(B) - w_i$, giving

$$d_{k-1}(u) = d_k(w) + (w_i - w_k).$$

In either case, the difference of two bits lies in $\{-1,0,+1\}$, so $|d_{k-1}(u)| \ge |d_k(w)| - 1 \ge 2-1 = 1$, whence $d_{k-1}(u) \neq 0$. Thus u is a non-palindrome and, by the induction hypothesis (A) at length $2k-1$, u is winning. Because every move from w leaves a winning non-palindrome, w is losing.

Inductive step for (A) at odd $L = 2k+1 \ge 5$. Assume (B) holds at even length $2k$, and let $w = AcB \in \{0,1\}^{2k+1}$ satisfy $d_k(w) \neq 0$. Replacing w by its bit-complement $\overline{w}$ (which preserves outcome and negates d_k) if necessary, we may assume $d_k(w) \ge 1$.

- *Case 1: $d_k(w) \ge 2$.* Delete the middle letter c to leave $u = AB \in \{0,1\}^{2k}$. Then $d_k(u) = \mathrm{wt}(A) - \mathrm{wt}(B) = d_k(w) \ge 2$, so u is losing by (B) at length $2k$, and w is winning.
- *Case 2: $d_k(w) = 1$ and $c = 1$.* If A contains at least one 0, delete that 0. The middle letter $c = 1$ shifts into the left half of the resulting length-$2k$ word $u = (A^{(-0)}c)B$, so $\mathrm{wt}(A^{(-0)}c) = \mathrm{wt}(A) + 1$ while B is unchanged; hence $d_k(u) = d_k(w) + 1 = 2$, and u is losing by (B). Otherwise $A = 1^k$ has weight k, forcing $\mathrm{wt}(B) = k-1$; since $c = 1$, w consists of $2k$ ones and a single 0 (in B), and deleting that unique 0 leaves the palindrome 1^{2k}, winning immediately.
- *Case 3: $d_k(w) = 1$ and $c = 0$.* If B contains at least one 1, delete that 1. The middle letter $c = 0$ shifts into the right half of $u = A(cB^{(-1)})$, reducing its weight by 1 while leaving A unchanged; hence $d_k(u) = d_k(w) + 1 = 2$, and u is losing by (B). Otherwise $B = 0^k$ has weight 0, forcing $\mathrm{wt}(A) = 1$; since $c = 0$, w consists of $2k$ zeros and a single 1 (in A), and deleting that unique 1 leaves the palindrome 0^{2k}, winning immediately.

This completes the induction. Finally, every losing word $w = AcB \in \{0,1\}^{2k+1}$ is a non-palindrome with $\mathrm{wt}(A) = \mathrm{wt}(B)$. By Vandermonde's identity, the number of words $AcB \in \{0,1\}^{2k+1}$ with $\mathrm{wt}(A) = \mathrm{wt}(B)$ is $2\sum_{r=0}^{k}\binom{k}{r}^2 = 2\binom{2k}{k}$. All 2^{k+1} palindromes of length $2k+1$ satisfy $\mathrm{wt}(A) = \mathrm{wt}(B)$ and are excluded from losing words, yielding $P(2k+1) \le 2\binom{2k}{k} - 2^{k+1}$. □

Proof of Theorem 2. First note that for any $v \in \{0,1\}^m$ ($m \ge 2$) and $a \in \{0,1\}$, we have $(ava)^R = av^Ra$, so ava is a palindrome if and only if v is a palindrome. In particular, ava is a non-palindrome whenever v is a non-palindrome.

Next observe a helper property: *if $v \in \{0,1\}^m$ ($m \ge 2$) is a losing word, then neither va nor av is a palindrome.* Indeed, suppose va were a palindrome. Writing $v = bv'$ with $b \in \{0,1\}$ and $|v'| = m-1 \ge 1$, the palindrome $va = bv'a$ must have equal endpoints ($b = a$) and palindromic interior v'. Deleting the leading letter b from $v = bv'$ would then leave the palindrome v', contradicting the assumption that v is losing. Symmetrically, if av were a palindrome, writing $v = v'b$ would show v' is a palindrome, again contradicting that v is losing.

We now prove that ava has the same outcome as v by strong induction on $m = |v| \ge 2$.

- *Case 1: v is winning.* Then some single-letter deletion $v' = v^{(-i)}$ ($1 \le i \le m$) is either a palindrome or a losing word of length $m-1 \ge 2$. Deleting the corresponding interior letter from $w = ava$ leaves $u = av'a$. If v' is a palindrome, then $u = av'a$ is a palindrome and w wins in one move. If v' is a losing word (hence a non-palindrome of length $m-1 < m$), then by the induction hypothesis $u = av'a$ is also losing, so moving $w \to u$ wins. (At the base length $m = 2$, both non-palindromes 01 and 10 win directly to a length-1 palindrome, so the induction hypothesis is not invoked.)
- *Case 2: v is losing.* (Vacuous at $m = 2$.) We show that every single-letter deletion u from $w = ava$ is a winning

non-palindrome.

- Deleting an interior letter of v leaves $u = av'a$, where $v' = v^{(-i)}$ is a winning non-palindrome of length $m - 1 < m$ (since v is losing). Thus u is a non-palindrome, and u is winning by the induction hypothesis.
- Deleting the leftmost or rightmost letter a leaves $u = va$ or $u = av$. By the helper property above, u is not a palindrome; moreover, deleting the remaining outer a from u leaves the losing word v, so u is winning.

Hence every move from $w = ava$ leaves a winning non-palindrome, so w is losing.

This completes the induction.

For part (1), fix $L \geq 4$ and partition the losing words of length L into those with $w_1 \neq w_L$ (counted by $Q(L)$) and those of the form ava with $a \in \{0,1\}$ and $|v| = L-2 \geq 2$. By the invariance just proved (and because ava is a palindrome when v is a palindrome), ava is losing if and only if v is losing. Since there are 2 choices for a, exactly $2P(L-2)$ losing words have equal ends, giving $P(L) = 2P(L-2) + Q(L)$.

For part (2), if $w = ava$ has odd length $2k+1 \geq 5$ (so $|v| = 2(k-1)+1$), then $d_0(w) = 0$, $d_1(w) = a - a = 0$, and $d_j(w) = d_{j-1}(v)$ for every $1 \leq j \leq k$. Thus $\max_{0\leq j\leq k} |d_j(ava)| = \max_{0\leq j\leq k-1} |d_j(v)|$. Consequently, if $w = ava$ were a losing word violating Conjecture 6, then v would be a strictly shorter losing word violating Conjecture 6; hence any minimal-length counterexample must satisfy $w_1 \neq w_L$. □

Proof of Theorem 3. First we verify directly that $c_0 = 0010001$ is losing (Figure 2). It is not a palindrome, and its seven single-letter deletions yield exactly four distinct length-6 non-palindromes, each admitting a one-move reply to a length-5 palindrome:

$$\begin{aligned}
\text{delete 1 or 2} \to 010001 &\xrightarrow{\text{del 1}} 10001,\\
\text{delete 3} \to 000001 &\xrightarrow{\text{del 6}} 00000,\\
\text{delete 4, 5, or 6} \to 001001 &\xrightarrow{\text{del 6}} 00100,\\
\text{delete 7} \to 001000 &\xrightarrow{\text{del 3}} 00000.
\end{aligned}$$

Every move from 0010001 therefore leaves a winning non-palindrome, proving that 0010001 is losing. Applying reversal and bit-complement symmetry (V_4) gives the four losing cores

$$C_7 = \{0010001,\ 1000100,\ 1101110,\ 0111011\}.$$

Now fix $k \geq 3$, $c \in C_7$, and any $X = x_1x_2\ldots x_{k-3} \in \{0,1\}^{k-3}$. Applying Theorem 2 iteratively $k-3$ times from the inside out ($c \mapsto x_{k-3}cx_{k-3} \mapsto \cdots \mapsto XcX^R$) proves that $XcX^R \in \{0,1\}^{2k+1}$ is losing. Because X is uniquely determined as the length-$(k-3)$ prefix of XcX^R and c is uniquely determined as its central length-7 subword, the map $(X,c) \mapsto XcX^R$ is injective on $\{0,1\}^{k-3} \times C_7$. Therefore $P(2k+1) \geq |C_7| \cdot 2^{k-3} = 4 \cdot 2^{k-3} = 2^{k-1}$. □

Proof of Proposition 4. Let $w = w_1\ldots w_{2k} \in \{0,1\}^{2k}$ ($k \geq 2$) satisfy either (i) $|d_r(w)| \geq 3$ for some $1 \leq r \leq k$, or (ii) $d_r(w) \geq 2$ and $d_s(w) \leq -2$ for some $1 \leq r,s \leq k$. If $|d_k(w)| \geq 2$, then w is already losing unconditionally by Theorem 1(B). Assume therefore that $|d_k(w)| \leq 1$. In particular, every index r or s appearing in (i) or (ii) satisfies $1 \leq r,s \leq k-1$, and w is not a palindrome (as its profile is not identically zero).

Consider any single-letter deletion $u = w^{(-i)} \in \{0,1\}^{2k-1}$ ($1 \leq i \leq 2k$) and any depth $1 \leq j \leq k-1$:

- *Case 1:* $1 \leq i \leq k$ (*deletion in the left half*). Because $j \leq k-1$, the trailing j letters of w occupy positions $2k-j+1 \geq k+2 > i$ and are untouched in u. If $j < i$, the leading j letters are also untouched, so $d_j(u) = d_j(w)$. If $i \leq j \leq k-1$, the leading j letters of u lose w_i and gain w_{j+1} (with $j+1 \leq k$), so $d_j(u) = d_j(w) + (w_{j+1} - w_i)$. Since $w_{j+1} \in \{0,1\}$, we obtain:
  - if $w_i = 0$, then $d_j(u) \in \{d_j(w),\ d_j(w)+1\}$ for all $1 \leq j \leq k-1$;
  - if $w_i = 1$, then $d_j(u) \in \{d_j(w)-1,\ d_j(w)\}$ for all $1 \leq j \leq k-1$.
- *Case 2:* $k+1 \leq i \leq 2k$ (*deletion in the right half*). Let $i' := 2k+1-i \in \{1,\ldots,k\}$. The leading $j \leq k-1$ letters are untouched. For $j < i'$, $d_j(u) = d_j(w)$; for $i' \leq j \leq k-1$, the trailing j letters lose w_i and gain w_{2k-j}, so $d_j(u) = d_j(w) + (w_i - w_{2k-j})$. Thus:
  - if $w_i = 1$, then $d_j(u) \in \{d_j(w),\ d_j(w)+1\}$ for all $1 \leq j \leq k-1$;
  - if $w_i = 0$, then $d_j(u) \in \{d_j(w)-1,\ d_j(w)\}$ for all $1 \leq j \leq k-1$.

In all cases, the entire profile $(d_1,\ldots,d_{k-1})$ shifts monotonically in one direction by at most 1 at each coordinate: either $d_j(u) \geq d_j(w)$ for all $1 \leq j \leq k-1$ (upward shift) or $d_j(u) \leq d_j(w)$ for all $1 \leq j \leq k-1$ (downward shift), with $|d_j(u) - d_j(w)| \leq 1$ throughout.

- Under (i), $|d_r(w)| \geq 3$ ($1 \leq r \leq k-1$) implies $|d_r(u)| \geq |d_r(w)| - 1 \geq 2$.
- Under (ii), $d_r(w) \geq 2$ and $d_s(w) \leq -2$ ($1 \leq r,s \leq k-1$) imply $d_r(u) \geq d_r(w) \geq 2$ under an upward shift, and $d_s(u) \leq d_s(w) \leq -2$ under a downward shift.

Hence every deletion $u = w^{(-i)}$ is an odd-length word of length $2k-1$ satisfying $\max_{1\leq j\leq k-1} |d_j(u)| \geq 2$. Such a word cannot be a palindrome, and by Conjecture 6 at length $2k-1$, u is winning. Because every move from w leaves a winning non-palindrome, w is losing. □

Combining Theorems 1 and 3 yields the unconditional two-sided exponential bracket

$$2^{k-1} \leq P(2k+1) \leq 2\binom{2k}{k} - 2^{k+1} < 2\binom{2k}{k} \qquad (k \geq 3), \quad (3)$$

whose upper bound grows like $4^k/\sqrt{\pi k}$ while the empirical counts $P(2k+1)$ track much closer to the palindrome count 2^{k+1} (Page-1 graphical abstract, right panel).

■ 4. COMPUTATIONAL VERIFICATION AND EMPIRICAL PICTURE

Deterministic Local Verification (`recompute.py`). Using only Python's standard library, the archived script `recompute.py` executes in under 1 second (0.59 s on a single core), writing `recompute.out` and proving Proposition 5. It runs two independent solvers sharing no code: a memoized

recursive solver `wins` and a bottom-up bitmask dynamic-programming solver covering every binary word of length $1 \le L \le 17$ (262,142 words total). Both solvers agree word for word on all 16,380 binary words of lengths $2 \le L \le 13$ with 0 discrepancies. Table 1 reports exact counts from local replay ($L = 7$–17) alongside external exhaustive sweeps ($L = 19$–45).

In addition to solving all 262,142 words through $L = 17$, `recompute.py` runs three negative-mutation controls verifying that each hypothesis in Conjecture 6 and Theorem 1 is essential:

1. *Move-rule mutation* (*endpoint-only deletion*): restricting moves to deleting only the leftmost or rightmost letter immediately creates 8 out-of-band odd losing words at $L = 7$ (earliest: 0000101, profile $[0, -1, -1, -2]$, violating even Theorem 1).
2. *Band-tightening mutation:* tightening $|d_j| \le 1$ to $|d_j| = 0$ fails on all 4 losing cores at $L = 7$, and restricting losing words to one-sided profiles fails on 12 losing words at $L = 13$ (starting with 0010100100001).
3. *Parity mutation:* dropping the odd-length requirement fails immediately at even length $L = 4$ (0011 and 1100 are losing with profiles $[0, -1, -2]$ and $[0, 1, 2]$).

External Exhaustive Sweeps, Tightness, and a Cautionary Failure. To situate Proposition 5 within the full empirical record without conflating external runs with local replay, we summarize earlier exhaustive checks (zero sampling in every run):

- *Two-program exhaustive sweeps through $L = 39$ and $L = 41$.* A plain C solver solved every length $L \le 29$, and a half-weight-pruned C solver (validated against the plain solver across $L = 9$–29) solved odd lengths $31, 33, 35$ ($P(35) = 2{,}827{,}624$). An independently coded sparse C solver (written without reading the earlier code and verified byte-identical to plain brute force through $L = 25$) extended the complete odd-length census through $L = 39$ ($P(39) = 12{,}222{,}116$; 0 out-of-band losing words). A separate NumPy solver in the archive independently reproduced all counts through $L = 25$. At length $L = 41$, because any losing word w must leave a winning word after deleting w_1, every losing candidate of length 41 is a single letter followed by a one-letter insertion into a length-39 palindrome or losing word. Exhausting all $\approx 2.11 \times 10^9$ such candidates (of which $\approx 756 \times 10^6$ leave the band $[-1, 1]$) found 0 out-of-band losing words.
- *Independent shielded re-check through $L = 45$.* A later shielded re-review built two further independent solvers (set-based and sparse) that agreed byte-for-byte on the complete losing-word lists from $L = 19$ through $L = 45$ (and matched a dense solver below 19), computing $P(41) = 25{,}140{,}996$, $P(43) = 51{,}429{,}900$, and $P(45) = 104{,}730{,}072$: 204,753,080 odd-length losing words in total across $L \le 45$, with 0 violations of $|d_j(w)| \le 1$.
- *Closed subgames to lengths 63 and 127.* Deleting a letter never increases weight $\mathrm{wt}(w)$, co-weight $L - \mathrm{wt}(w)$, or the number of maximal runs of equal letters. Hence low-weight and run-bounded families form closed subgames solvable independently of the full census. Exhaustive subgame solvers confirmed Conjecture 6 with 0 violations for weight or co-weight ≤ 5 through odd $L = 63$, weight or co-weight ≤ 6 through odd $L = 57$, and at most $5, 6, 7, 8$ runs through odd $L = 127, 77, 51, 43$, respectively.

Boundary tightness and local-proof obstacles. Figure 3 highlights why Conjecture 6 is structurally tight:

- *Crowded boundary class.* The share of losing words whose profile touches *both* -1 and $+1$ grows monotonically from 4.7% (12/256) at $L = 13$ to 14.3% (264/1,848) at $L = 17$ and 39.6% (4,834,916/12,222,116) at $L = 39$. Thus the band width 1 cannot be reduced, and a putative counterexample would step only one unit outside an increasingly crowded edge.
- *Persistent one-move near-misses.* At every odd length $9 \le L \le 29$, balanced words ($d_k = 0$) with profile reaching ± 2 include cases with *exactly one* winning move (counts 4, 4, 8, 16, 24, 32, 40, 48, 56, 64, and 72; checked locally through $L = 15$ in `recompute.py`). Every one-move near-miss in this range has weight or co-weight exactly 4 (hence lies inside the low-weight subgame checked to $L = 63$); among words with 7 to $L-7$ ones, every tested out-of-band word has at least 2 winning moves.
- *Failure of bounded-horizon strategies.* Attempts to prove Conjecture 6 by a bounded-lookahead local move rule fail from $L = 11$ onward: one representative local rule fails on 8, 88, 568, 2,564, and 9,980 words at $L = 11$, 13, 15, 17, and 19. Winning moves frequently pass through even-length losing positions that lie inside the even band and depend globally on the census two letters shorter.

Cautionary sibling law that broke at $L = 35$. A sibling conjecture fitted to the exact same census asserted that the maximum number of mismatched mirror pairs in an odd-length losing word equals $2\lfloor (L + 1)/6 \rfloor$ for all odd $L \ge 7$. It matched with equality across twelve consecutive odd lengths $L = 7, 9, \ldots, 29$ (maxima $2, 2, 4, 4, 4, 6, 6, 6, 8, 8, 8, 10$; confirmed locally for $L = 7$–17 in `recompute.out`) and continued to hold at $L = 31$ and $L = 33$. At $L = 35$, however, the formula predicts 12 mismatched pairs whereas exhaustive search found maximum 10 (0 losing words with ≥ 12 mismatched pairs), before 12 mismatched pairs finally appeared at $L = 37$ (4 words) and $L = 39$ (64 words). This failure underscores plainly that exhaustive agreement across twenty odd lengths is evidence for Conjecture 6, never a proof.

Table 1. Odd-length losing-word census $P(L)$, differing-end counts $Q(L)$, proved lower bound 2^{k-1} (Theorem 3), proved upper bound $2\binom{2k}{k}$ (Theorem 1), boundary class touching both -1 and $+1$, and one-move near-misses ($d_k = 0$, $\max|d_j| \geq 2$, unique winning move). Top block ($L \leq 17$) is replayed locally in `recompute.py` (Proposition 5); bottom block ($19 \leq L \leq 45$) is cited from external exhaustive C sweeps (Section 4.2). Zero out-of-band losing words occur at any length.

L	$P(L)$	$Q(L)$	2^{k-1}	$2\binom{2k}{k}$	Boundary $(-1,+1)$	Share	Near-misses	Provenance
7	4	4	4	40	0	0.0%	—	Local
9	24	16	8	140	0	0.0%	4	Local
11	84	36	16	504	0	0.0%	4	Local
13	256	88	32	1,848	12	4.7%	8	Local
15	708	196	64	6,864	68	9.6%	16	Local
17	1,848	432	128	25,740	264	14.3%	24*	Local (P, Q, bdry)
19	4,616	920	256	97,240	852	18.5%	32	External C / numpy
21	11,128	1,896	512	369,512	2,468	22.2%	40	External C / numpy
23	26,060	3,804	1,024	1,410,864	6,644	25.5%	48	External C / numpy
25	59,488	7,368	2,048	5,408,312	16,880	28.4%	56	External C / numpy
29	291,560	—	8,192	80,233,200	96,112	33.0%	72	External C
35	2,827,624	—	65,536	4,667,212,440	1,062,868	37.6%	—	External C
39	12,222,116	—	262,144	70,690,527,600	4,834,916	39.6%	—	External C
41	25,140,996	—	524,288	269,128,937,220	—	—	—	External C (re-check)
43	51,429,900	—	1,048,576	1.03×10^{12}	—	—	—	External C (re-check)
45	104,730,072	—	2,097,152	3.92×10^{12}	—	—	—	External C (re-check)

*At $L = 17$, $P(17)$, $Q(17)$, and boundary count 264 are verified locally in `recompute.py`; the near-miss count 24 is cited from the external near-miss log ($L = 9$–29).
Intermediate external odd counts: $P(27) = 132{,}912$, $P(31) = 629{,}628$, $P(33) = 1{,}341{,}836$, $P(37) = 5{,}902{,}220$.

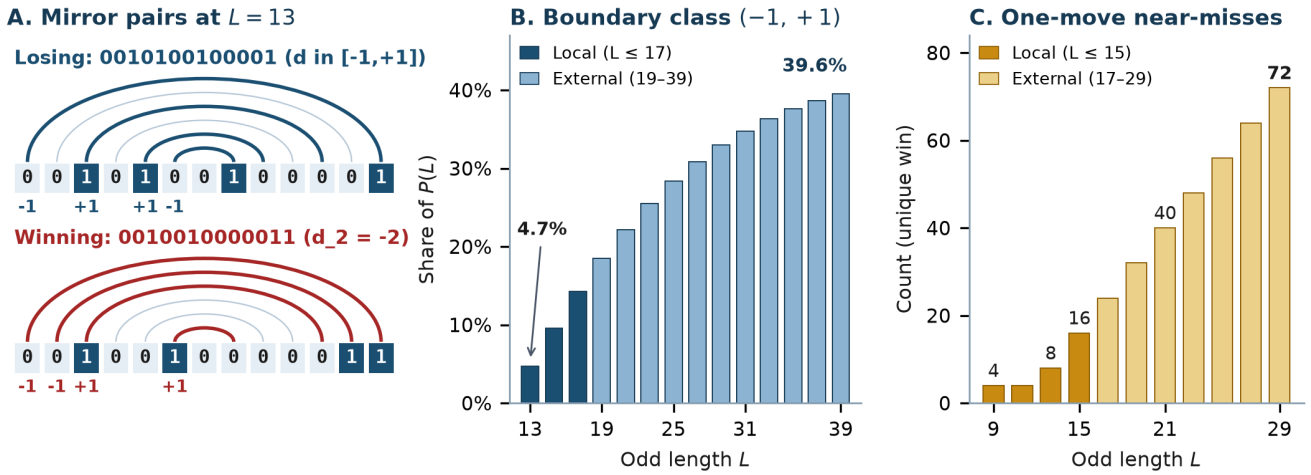


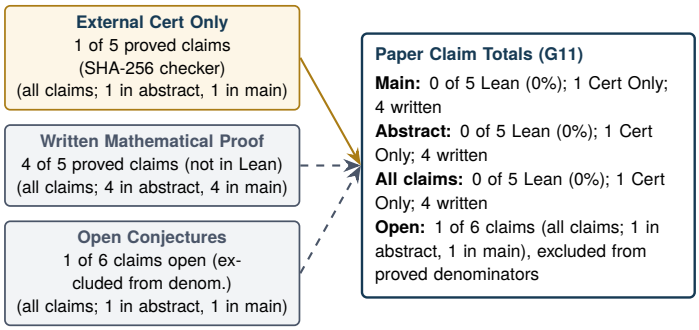
Figure 3. Tightness of the mirror-prefix band across odd lengths. Panel A: mirror-pair arcs for the losing boundary word 0010100100001 (top; profile touches both -1 and $+1$) versus the balanced winning word 0010010000011 (bottom; $d_2 = -2$, unique winning move). Panel B: growing share of losing words touching both -1 and $+1$, rising from 4.7% (12/256) at $L = 13$ to 39.6% (4,834,916/12,222,116) at $L = 39$. Panel C: counts of balanced out-of-band words with exactly one winning move across $L = 9$–29 ($4, 4, 8, 16, 24, 32, 40, 48, 56, 64, 72$; all of weight or co-weight 4).

5. FORMAL VERIFICATION & READER'S TRUST BOUNDARY

What You Must Trust to Accept This Paper:

1. **Lean 4 Kernel:** No Lean 4 formalization is claimed in this manuscript.
2. **Written Proofs:** All mathematical arguments rest on standard human-readable proofs.
3. **External Certificates & Written Proofs:** 1 SHA-256-pinned external artifact(s) plus any claims marked Written or Unsettled in Table 2 below.

Reader's Trust-Flow Architecture (Distinct Evidence Paths to Claim Totals)



Anti-Shortcut Audit Strip — No-Lean Disclosure

N/A: Lean 4: Not claimed | N/A: sorry / admit: N/A | N/A: native_decide: N/A | N/A: Escape hatches: N/A | N/A: Library coverage: N/A | N/A: Lean fidelity: N/A | WARN: Human referee: pending

Table 2. Per-Claim Reader's Trust Boundary Ledger (G11 Denominators Derived Mechanically).[a]

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 1: For all k >= 1, every odd word of length 2k+1 with d_k(w) != 0 is winning, every even word of length 2k with |d_k(w)| >= 2 is losing, and P(2k+1) <= 2*C(2k,k) - 2^(k+1)	Written	Written proof	Lean: none; fidelity: n/a	Elementary joint induction on word length L >= 2 (Section 3); verified on all 262,142 words of length L <= 17 in recompute.py. No Lean 4 formalization.
Thm. 2: For every non-palindrome v (|v| >= 2) and letter a in {0,1}, a v a has the same outcome as v; hence P(L) = 2*P(L-2) + Q(L) for L >= 4 and any minimal counterexample to Conj. 6 satisfies w_1 != w_L	Written	Written proof	Lean: none; fidelity: n/a	Elementary strong induction on |v| >= 2 using shift-palindrome exclusion for losing words (Section 3); verified on all wrapped words through L = 17 in recompute.py. No Lean 4 formalization.
Thm. 3: Each length-7 core c in {0010001, 1000100, 1101110, 0111011} is losing, and X c X^R is losing for every k >= 3 and X in {0,1}^(k-3), yielding P(2k+1) >= 2^(k-1)	Written	Written proof	Lean: none; fidelity: n/a	Direct 4-move check of 0010001, V_4 symmetry, and iterated Theorem 2 (Section 3); verified through L = 17 in recompute.py. No Lean 4 formalization.
Prop. 4: If Conj. 6 holds at odd length 2k-1 (k >= 2), then every even word of length 2k whose profile reaches +-3 or both +2 and -2 is losing	Written	Written proof	Lean: none; fidelity: n/a	Conditional on Conj. 6 at length 2k-1; elementary monotonic one-sided profile-shift proof combined with Thm. 1(B) (Section 3); checked through L = 16 in recompute.py. No Lean 4 formalization.

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Prop. 5: Standalone exact replay verifies worked examples, dual-solver agreement through L = 13 (16,380 words), complete census through L = 17 (262,142 words, 0 band violations), theorem checks, and one-move near-misses for odd L = 9..15	Cert Only	recompute.py (8622ff918343) [archived; replay pending]	External cert only; fidelity: n/a	No Lean 4 formalization. Local exact verification via recompute.py (external exhaustive C sweeps through L = 39, L = 41, re-check through L = 45, and closed subgames to L = 63, 127 are cited separately in Section 4.2).
Conj. 6: Every odd-length losing word w of length 2k+1 satisfies |d_j(w)| <= 1 for all 0 <= j <= k	Unsettled	Open conjecture	Lean: none; fidelity: n/a	Explicitly unsettled open conjecture (excluded from G11 denominators); verified locally through L = 17 and by external exhaustive sweeps through odd L = 39, L = 41, and independent re-check through L = 45, plus closed subgames to L = 63 and 127, but not proved.

[a]Allowed pure-kernel axioms: `propext`, `Classical.choice`, `Quot.sound`. Actual axioms printed above are read directly from verified outputs. Provenance nonce: 8f5ccc00be9c | Fidelity SHA-256: 7af96338f150.

6. PRIOR WORK AND OPEN QUESTIONS

Origin and nearest relatives. This palindrome deletion game was invented by AI research agents rather than drawn from a printed open problem. Prior-work searches identified several related word games, each governed by different rules:

1. *OEIS* `A362368` (*Rajesh, 2023*) [4]. Carries the same name ("the palindrome game"), but uses a different rule: players alternately delete any nonempty *palindromic substring* from a binary word, and the player who removes the final palindrome wins. Its losing-word counts $(0, 0, 2, 0, 4, 0, 18, 4, \ldots)$ differ already at length 3.
2. *Ravsky* (*2003 / `arXiv:1004.1279`*) [5]. Studies the minimum number of single-letter deletions needed to turn a binary word into a palindrome or an antipalindrome. Remark 3 (p. 4, read in full) introduces an antagonistic deletion game where Player 1 chooses a length-n word, players alternately delete single letters until a palindrome or antipalindrome is reached, and Player 1's payoff is the total number of moves. Both the terminal set (including antipalindromes) and the scoring rule (move-count maximization rather than impartial last-move win) differ from the game studied here, and no profile law for losing positions is stated.
3. *StrNim* (*Mizuno, Yoshinaka, and Shinohara, 2025*) [3] `[to check]`. A string-deletion game in which each move deletes a nonempty run of identical consecutive letters rather than an arbitrary single letter, with mirror-play sufficient conditions for $\mathcal{P}$-positions and no palindrome goal.
4. *Relator games* (*Gates and Kelvey, 2021*) [2] `[to check]` *and rewrite games* (*Duchêne et al., 2020*) [1] `[to check]`. Relator achievement/avoidance games build words by appending group generators rather than deleting letters; taking-and-merging rewrite games apply local length-reducing rules $a^k \to \varepsilon$ to study regular/context-free $\mathcal{P}$-position languages rather than global palindromic termination.
5. *Competitive-programming relatives.* Variants such as Codeforces 276B (where players may freely permute letters after deletion, collapsing the state to letter counts) and Codeforces 1527B (bit-flipping or reversal rather than deletion) alter the move set fundamentally.

Prior-work searches logged. Multiple independent searches were logged and passed the repository literature-search check: 26 OEIS queries (7 hits, only `A362368` related) and 7 Stack Exchange API queries by the initial unit; 36 helper-table rows plus 9 arXiv, 7 web, 3 self-archive (Zenodo, HAL, DataCite), and 11 catalogue queries by a second unit; 22 helper-table rows plus 7 OEIS term queries, 5 self-archive, 7 catalogue (including Open Problem Garden and MathWorld), and 3 web queries by a third unit; 6 keyed CORE queries on 2026-11-19 (10 records returned, 0 relevant); and a 4-query Google Scholar sweep on 2026-11-22 (35 records inspected). Neither the single-letter palindrome deletion game nor Conjecture 6 was found in the searches logged.

Unread and unsearched items (`[to check]`). The following sources were not read in full or not searched and remain flagged `[to check]`: (1) the full texts of Mizuno, Yoshinaka, and Shinohara [3], Gates and Kelvey [2], and Duchêne et al. [1] (verified via arXiv abstracts and metadata only); (2) MathOverflow (not reached by any automated route); and (3) books in print not

consulted directly: Berlekamp, Conway, and Guy's *Winning Ways*, Albert, Nowakowski, and Wolfe's *Lessons in Play*, Siegel's *Combinatorial Game Theory*, and Lothaire's *Combinatorics on Words*.

Open questions. Beyond Conjecture 6 itself, four concrete questions remain open: (i) Does $P(2k+1)$ grow at the palindrome rate $2^{k+o(k)}$ (recall $P(45)/2^{23} \approx 12.48$), or does a sharper upper bound improve upon $2\binom{2k}{k}$? (ii) Can the even-length losing positions that witness winning moves from balanced out-of-band odd words (such as 00110011 from 001100011) be characterized structurally enough to close the induction between odd and even lengths? (iii) Do the one-move near-misses satisfy the exact formula $8k-32$ (weight or co-weight 4) for all odd $L = 2k+1 \geq 13$? (iv) Can Theorems 1–3 and Proposition 4 be formalized in pure-kernel Lean 4 to raise formal coverage from 0 of 5 to 4 of 5 or 5 of 5?

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

All manuscript sources, figure scripts, and the standalone exact verifier reside in `papers/palindrome-game/draft1/`. Running `python3 papers/palindrome-game/draft1/recompute.py` finishes in under 1 second using only the Python standard library and deterministically reproduces Proposition 5, the local block of Table 1, all three negative-mutation controls, and the Section 1 Python snippet, writing `recompute.out`. No Lean 4 formalization is claimed in this draft.

AI research agents built and directed by Tyler B. Hughes formulated the game and conjecture, proved Theorems 1–3 and Proposition 4, wrote the verification scripts and prior-work searches, and drafted this manuscript. No human mathematician has yet refereed the proof.

Drafted: November 2026 **Updated:** November 24, 2026

REFERENCES

(1) Duchêne, E.; Marsault, V.; Parreau, A.; Rigo, M. Taking-and-merging games as rewrite games. *Discrete Mathematics & Theoretical Computer Science* **2020**, *22* (4), #5; arXiv:1902.07011. DOI: 10.23638/DMTCS-22-4-5.

(2) Gates, Z.; Kelvey, R. Relator Games on Groups. *arXiv preprint* **2020**, arXiv:2011.08915.

(3) Mizuno, S.; Yoshinaka, R.; Shinohara, A. StrNim: a variant of Nim played on strings. *arXiv preprint* **2025**, arXiv:2503.16781.

(4) Rajesh, K. Sequence A362368: Number of binary strings of length n which are losing configurations in the palindrome game. *The On-Line Encyclopedia of Integer Sequences* **2023**, https://oeis.org/A362368.

(5) Ravsky, A. A new measure of asymmetry of binary words. *arXiv preprint* **2010**, arXiv:1004.1279 (originally in *Journal of Automata, Languages and Combinatorics* 8 (2003), no. 1, 71–74).