hexagon:2610.00308v2

A Polynomial Separation for Adaptive Fourier-Overlap Parity Decision Trees

Yuran Xiu

October 15, 2026

Abstract

Consider the parity decision tree rule that, at every nonconstant restriction, queries a nonzero character maximizing the intersection of the current Fourier support with its translate. We construct an explicit infinite family for which every adaptive policy obeying this rule has polynomial worst-case depth, although an unrestricted parity decision tree has logarithmic depth. For every $\ell \geq 1$, let $R = 2^\ell$. Our function has $n = 4\ell - 1 + R^2\ell + R$ input bits and exactly $k = R^6 - R^5 + R$ nonzero Fourier coefficients. Every history-dependent maximizing policy terminates and computes the function, but has a feasible path of length at least $R > k^{1/6}$; an explicit unrestricted tree has depth at most $5\ell \leq \log_2 k$. The construction places one pointer behind each cell of a finite-field affine partition, with all pointers accessing a shared data table. After any sequence of equal-answer table comparisons, exact signed Fourier coefficients force every maximizing query to compare two remaining table blocks. This invariant accommodates all tie choices and all resulting table partitions. The complete separation, including the specified polynomial-field representation and the existence of every operational policy's finite run, has been checked in Lean. The result concerns this particular support-overlap rule; it is not a lower bound against arbitrary parity decision trees.

1 Introduction

A parity decision tree may query the parity of any subset of the input bits. One approach to constructing shallow trees for a Fourier-sparse Boolean function is to fold many supported characters together at each query. If S is the current Fourier support, the number of ordered endpoints that fold in direction $\gamma \neq 0$ is

$$C_S(\gamma) = |S \cap (S + \gamma)|.$$

Choosing a direction with maximum $C_S(\gamma)$ is a natural local rule. Its behavior is inherently adaptive: after an answer, coefficients combine with signs, some sums vanish, and the next score must be evaluated on the resulting function. A bound on the initial support alone does not control that process.

We give a separation that survives this adaptivity. Every maximizing choice, including any dependence on the entire preceding query-answer history, has a long feasible branch. At the same time, reading an address and then a pointer computes the same function with few queries.

Theorem 1.1 (Main result)**.** *For every integer $\ell \geq 1$, put $R = 2^\ell$ and $n = 4\ell - 1 + R^2\ell + R$. There is an explicitly specified function $f_\ell : \mathbb{F}_2^n \to \{-1, 1\}$ with Fourier sparsity*

$$k_\ell = R^6 - R^5 + R \tag{1}$$

such that the following statements hold.

(i) *Every policy that queries an intrinsically nonzero character of maximum current support-overlap at each nonconstant restriction has a unique finite decision tree run, which computes* f_ℓ.

(ii) *For every such policy, that tree has a feasible path of length at least* $R > k_\ell^{1/6}$. *The bound permits every history-dependent tie rule and every choice of affine coordinates on the current domain.*

(iii) *An explicit unrestricted parity decision tree computes* f_ℓ *with depth at most* $5\ell \le \log_2 k_\ell$.

Furthermore, $k_\ell \to \infty$ *as* $\ell \to \infty$.

The adjective "intrinsically" excludes a parity that is constant on the current affine domain, even if its ambient coefficient vector is nonzero. The precise operational rule is defined in Section 2. In particular, the algorithm stops exactly when the restricted function is constant. A singleton Fourier support need not be constant, so it is not an alternative stopping condition.

Relation to earlier work. Tsang, Wong, Xie, and Zhang [1] study Fourier sparsity and spectral norm in parity decision tree constructions. Their coefficient-based methods should be distinguished from the support-cardinality maximization rule analyzed here. Mande and Sanyal [2] develop a folding approach and formulate a structural conjecture about the abundance of folding pairs. Hatami, Hosseini, Lovett, and Ostuni [3] refute two proposed additive structure conjectures. Their Theorem 8 bounds intersections of translates of the initial support of a constructed function; that statement itself does not quantify over the current supports along every adaptive execution.

Tarannikov [4] constructs a stronger initial-support example using an affine partition obtained from a linear spread. Section 4 of that work separates an inheritance assumption under restrictions from an adaptive depth claim. We use the finite-field spread geometry behind that partition, and give its required properties explicitly below. Addressing and table indirection are standard construction ideas. The contribution here is their composition with a shared table and the exact current-spectrum argument that forces comparisons after every possible sequence of table merges. We make no claim that the affine partition or the elementary Fourier identities are new. The theorem is a limitation of the specified maximum-overlap rule. It neither refutes the XOR log-rank conjecture nor establishes a limitation of all algorithms described as greedy.

Proof mechanism. The address selects one of $N = R^2$ cells. Each cell has an independent ℓ-bit pointer, and the selected pointer addresses a common table of R data bits. Along a branch that always answers a comparison by equality, the current table is described by an arbitrary partition into r blocks. We compute the signed Fourier coefficients for every such partition. The finite-field geometry bounds every query with a nonzero address or pointer component. Meanwhile, the fact that the block indicators sum to one forces some pair of their nonconstant Fourier supports to overlap substantially. That pair gives a strictly better table-comparison score. Thus all maximizers are comparisons, equality merges the chosen pair, and the argument repeats until only one nonconstant data bit remains.

2 Fourier conventions and the operational rule

For a finite-dimensional binary vector space V, let V^* denote its dual. For $\alpha \in V^*$, write $\chi_\alpha(x) = (-1)^{\alpha(x)}$. The normalized Fourier transform of $g : V \to \mathbb{R}$ is

$$\widehat{g}(\alpha) = \frac{1}{|V|} \sum_{x \in V} g(x)\chi_\alpha(x), \qquad S(g) = \{\alpha \in V^* : \widehat{g}(\alpha) \neq 0\}.$$

The zero character belongs to $S(g)$ exactly when its coefficient is nonzero. Character orthogonality gives Fourier inversion $g = \sum_\alpha \widehat{g}(\alpha)\chi_\alpha$. Indeed, a nonzero character sums to zero by pairing x with $x + x_0$ for any x_0 at which that character is -1. Applying this identity to the difference of two characters yields the orthogonality formula and inversion.

For $\gamma \in V^*$ define

$$C_g(\gamma) = |S(g) \cap (S(g) + \gamma)|. \tag{2}$$

For $\gamma \neq 0$, this is twice the number of unordered supported pairs whose sum is γ. Thus the ordered-endpoint and unordered-pair conventions give exactly the same maximizing queries.

A current domain is a nonempty affine subspace D of the original input space. Choose an affine bijection $\phi : V \to D$, and put $g = f \circ \phi$. Every ambient parity restricted to D has the form $\gamma(x) + c$ in this chart. It is *intrinsically nonzero* if $\gamma \neq 0$. The constant c changes the names of its two answers, but not its score.

Lemma 2.1 (Coordinate invariance)**.** *The maximum-overlap rule is independent of the affine chart. Every intrinsic character is the linear part of an ambient parity restricted to D.*

Proof. Two charts differ by $x = By + t$, where B is an invertible linear map. If $h(y) = g(By + t)$, then a change of variables gives

$$\widehat{h}(B^*\alpha) = (-1)^{\alpha(t)}\widehat{g}(\alpha),$$

where $B^*\alpha = \alpha \circ B$. Hence $S(h) = B^*S(g)$ and $C_h(B^*\gamma) = C_g(\gamma)$. This is a bijection between nonzero characters and preserves all ties. Finally, a linear functional on the direction space of D extends to the ambient space by extending a basis and assigning arbitrary values on the added basis vectors. Its value on the translation of D contributes only the constant c. □

Definition 2.2 (Maximizing policy)**.** A maximizing policy assigns, to every finite history of ambient queries and answers and every nonempty current affine domain on which f is nonconstant, an ambient parity whose intrinsic character is nonzero and maximizes (2) for the actual current restriction. Its run outputs the constant value at a constant domain; otherwise it makes the assigned query and continues on each answer domain with the extended history.

The score in this definition uses the support of the actual restricted function, after adding coefficients and performing all cancellations. The policy may use the entire history to choose among ties. No fixed order of maximizing directions is imposed.

Lemma 2.3 (Existence, termination, and uniqueness)**.** *For every Boolean function on a finite binary cube, maximizing policies exist. Every such policy has a unique finite run from any nonempty affine domain. The run computes the restricted function and has depth strictly less than the cardinality of that domain.*

Proof. At a nonconstant domain, a chart has at least two points, so its dual has a nonzero character. The finite nonempty set of nonzero characters has a maximum score. Lemma 2.1 extends any maximizer to an ambient query, so policies can be defined at every required state.

For an intrinsically nonzero query γ, choose x_0 with $\gamma(x_0) = 1$. Translation by x_0 interchanges the two intrinsic answer fibers. Both fibers are therefore nonempty proper affine subspaces. This remains true after any answer-label shift. Their cardinalities are strictly smaller than that of the current domain.

Strong induction on domain cardinality now constructs the run. A constant domain gives its uniquely determined constant leaf. At a nonconstant domain, the policy specifies a single query and induction constructs both subtrees. If the child domains have sizes $d_0, d_1 < d$, the induction hypothesis gives child depths at most $d_0 - 1, d_1 - 1$, and the new depth is at most $\max(d_0, d_1) < d$. The two domains partition the parent, so the resulting tree computes the function on the parent. Uniqueness follows by the same induction: the constant value or the policy's query is fixed, and both subtrees are unique. The proof also covers a singleton domain and a zero-dimensional chart. It does not require the maximum score to be positive. □

3 A specified finite-field address partition

Fix $\ell \geq 1$ and set

$$R = 2^\ell, \qquad q = 2\ell, \qquad N = 2^q = R^2, \qquad m = 2q - 1 = 4\ell - 1.$$

We first specify the field and all its coordinates, so that no favorable choice of a field representation is part of the construction.

For $c = (c_0, \ldots, c_{q-1}) \in \mathbb{F}_2^q$, put

$$L_c(X) = \sum_{a=0}^{q-1} c_a X^a, \quad P_c(X) = X^q + L_c(X), \quad \mathrm{code}(c) = \sum_{a=0}^{q-1} c_a 2^a.$$

Order the vectors by their integer codes and let p_q be the first irreducible polynomial among the P_c.

Lemma 3.1 (Explicit polynomial selection)**.** *The polynomial p_q is specified by a terminating finite search using binary coefficient arithmetic. It is monic irreducible of degree q.*

Proof. For a candidate P_c, enumerate all $a, b \in \mathbb{F}_2^q$ and test whether $L_a L_b = P_c$. Each test compares coefficients at degrees $0, \ldots, 2q$, with product coefficient

$$[X^s](L_a L_b) = \sum_{\substack{0 \leq i,j < q \\ i+j=s}} a_i b_j \quad \text{in } \mathbb{F}_2.$$

This coefficient range covers both polynomials, since $\deg(L_a L_b) \leq 2q - 2$ when the product is nonzero and $\deg P_c = q$. Accept a candidate precisely when none of these products equals it.

If P_c is reducible, it factors into two nonunits. Since P_c is nonzero of degree q, these factors have positive degrees summing to q; both degrees are less than q, so the factors occur as L_a, L_b in the test. Conversely, if $P_c = L_a L_b$ and P_c were irreducible, one factor would be a unit. The other would then have degree q, contradicting its degree being less than q. Thus acceptance is equivalent to irreducibility.

For completeness, the standard existence of a field with 2^q elements can be seen by taking the splitting field of $X^{2^q} - X$ over $\mathbb{F}_2$. Its roots form a field: Frobenius preserves sums and products, and nonzero roots are closed under inversion. The derivative is -1, so there are exactly 2^q distinct roots. A finite subgroup of a field's multiplicative group is cyclic. One elementary justification is to let e be its exponent: in a finite abelian group there is an element of order e, by taking for each

prime an element whose order has maximal prime-power part and multiplying these coprime-order parts. All group elements are roots of $X^e - 1$, so the root bound forces the group size to be at most e, and hence equal to e. Choose a multiplicative generator of the root field. It generates that field over $\mathbb{F}_2$, and its minimal polynomial is monic irreducible of degree q. Thus some candidate passes the test. Scanning the finite ordered list and stopping at its first accepted member defines p_q and terminates. □

Let

$$K = \mathbb{F}_2[X]/(p_q).$$

Represent each element by its unique polynomial of degree less than q. The ordered basis is $1, X, \ldots, X^{q-1}$, and $\lambda : K \to \mathbb{F}_2$ is the constant coefficient functional. In particular, $\lambda(1) = 1$. Arithmetic and equality in this representation are determined by polynomial arithmetic modulo p_q; every nonzero element has a unique inverse. The construction is therefore an explicit mathematical family. We do not need a running-time bound for polynomial selection.

Define the affine address space

$$\mathcal{A} = \{(u, v) \in K^2 : \lambda(u) = 1\}. \tag{3}$$

It has dimension m. Its specified chart reads the $q - 1$ nonconstant coefficients of u and all q coefficients of v. Put $H = \ker \lambda$; then the chart identifies its direction space with $X_0 = H \oplus K$, using origin $(1, 0)$. For each $i \in K$, let

$$\mathcal{A}_i = \{(u, iu) : \lambda(u) = 1\}.$$

Every u in (3) is nonzero, so the cells partition $\mathcal{A}$: the unique cell containing (u, v) is indexed by $i = v/u$. Each cell has size 2^{q-1} and density $1/N$ in $\mathcal{A}$. In the chart it is

$$a_i + V_i, \qquad a_i = (0, i), \qquad V_i = \{(h, ih) : h \in H\} \subseteq X_0.$$

Write $L_i = V_i^{\perp} \subseteq X_0^*$.

Lemma 3.2 (Annihilator geometry)**.** *For every $i \in K$, $|L_i| = N$. Every nonzero element of X_0^* belongs to at most two of the L_i.*

Proof. The direction V_i has dimension $q - 1$, while X_0 has dimension $2q - 1$. Thus its annihilator has dimension q and cardinality N.

Represent a functional on $H \oplus K$ by a pair (p, b) of linear functionals on K, restricting p to H. Two representatives differ by adding $(\lambda, 0)$, since the annihilator of H in K^* is $\{0, \lambda\}$. Membership in L_i is equivalent to

$$p + b \circ M_i \in \{0, \lambda\}, \qquad M_i(h) = ih. \tag{4}$$

If $b = 0$, this condition makes the original functional zero on $H \oplus K$, so no nonzero functional of this type belongs to any L_i. If $b \neq 0$, the map

$$K \longrightarrow K^*, \qquad i \longmapsto b \circ M_i$$

is linear and injective. Indeed, if $i \neq 0$, multiplication by i is surjective, so $b \circ M_i$ cannot vanish. Domain and codomain both have cardinality N, hence the map is bijective. The two distinct target functionals p and $p + \lambda$ in (4) therefore give exactly two indices i. This proves the asserted upper bound. □

Lemma 3.3 (Signed cell coefficients)**.** *In the specified address chart,*

$$\widehat{\mathbf{1}_{\mathcal{A}_i}}(a) = \begin{cases} N^{-1}\varepsilon_i(a), & a \in L_i, \\ 0, & a \notin L_i, \end{cases} \qquad \varepsilon_i(a) = (-1)^{a(a_i)}.$$

Moreover, $\sum_i \mathbf{1}_{\mathcal{A}_i} = 1$ *as functions.*

Proof. The coefficient is $|X_0|^{-1}(-1)^{a(a_i)} \sum_{h \in V_i} (-1)^{a(h)}$. The sum is $|V_i|$ if a annihilates V_i, and zero otherwise by character orthogonality. The ratio $|V_i|/|X_0|$ is $1/N$. The final identity follows from the partition property. □

4 The shared-table function and its short tree

Let $B = \mathbb{F}_2^\ell$, so $|B| = R$. Besides the address $x = (u, v) \in \mathcal{A}$, the input has an independent pointer $Y_i \in B$ for every $i \in K$ and an independent data bit $w_s \in \mathbb{F}_2$ for every $s \in B$. Define

$$f_\ell(x, Y, w) = (-1)^{w_{Y_{v/u}}}. \tag{5}$$

The coordinates of K, the chart of $\mathcal{A}$, and the bit order of each pointer have all been specified. Thus (5) defines a single Boolean function for every ℓ. Its number of input bits is

$$m + N\ell + R = 4\ell - 1 + R^2\ell + R.$$

The same data table is used by every pointer; it is not a separate table for each address cell.

Proposition 4.1 (Short unrestricted tree)**.** *There is an explicit parity decision tree computing* f_ℓ *with depth at most* 5ℓ*.*

Proof. Query the m address bits, which determine $i = v/u$. Query the ℓ bits of Y_i, obtaining a table location s. Finally query w_s and output $(-1)^{w_s}$. All these queries are coordinate parities, and their total number is $m + \ell + 1 = 5\ell$. □

5 Exact spectra after arbitrary table merges

Let $\mathcal{B} = (B_1, \ldots, B_r)$ be any partition of B into nonempty blocks. Restrict the table by requiring that all w_s with $s \in B_j$ equal a free bit v_j. In this current chart, the address, every pointer, and the r bits $v_1, \ldots, v_r$ remain independent. The actual restricted function is

$$F_{\mathcal{B}}(x, Y, v) = \sum_{j=1}^{r} H_j(x, Y)(-1)^{v_j}, \qquad H_j(x, Y) = \sum_{i \in K} \mathbf{1}_{\mathcal{A}_i}(x)\mathbf{1}_{B_j}(Y_i). \tag{6}$$

At every (x, Y) exactly one H_j equals one and the rest vanish, so this formula is Boolean.

For $\beta \in B^*$, put

$$b_j(\beta) = \frac{1}{R} \sum_{s \in B_j} (-1)^{\beta(s)}, \qquad P_j = \{\beta \neq 0 : b_j(\beta) \neq 0\}, \qquad p_j = |P_j|, \quad P = \sum_j p_j. \tag{7}$$

Here P is a number; the sets of pointer frequencies are P_j.

Lemma 5.1 (Block-frequency multiplicities)**.** *For all* j*,* $b_j(0) = |B_j|/R > 0$*. If* $r \geq 2$*, then* $p_j \geq 1$ *for every* j*. For every* $\beta \neq 0$*, the number* $d_\beta = |\{j : \beta \in P_j\}|$ *is either zero or at least two.*

Proof. The formula at zero is immediate. If $r \geq 2$, each B_j is nonempty and proper. If its nonconstant coefficients all vanished, Fourier inversion would make $\mathbf{1}_{B_j}$ a constant function, a contradiction. The partition identity $\sum_j \mathbf{1}_{B_j} = 1$ implies $\sum_j b_j(\beta) = 0$ for every nonzero β. Exactly one nonzero summand cannot have sum zero, proving the last assertion. □

Let

$$U = X_0^* \oplus \bigoplus_{i \in K} B^*$$

be the frequency space of the address and pointers. Write $E_i\beta$ for the pointer frequency with entry β in block i and zero elsewhere. Let e_j be the jth standard basis character on the current data bits.

Lemma 5.2 (Current signed spectrum)**.** *The support of H_j is exactly the disjoint union*

$$T_j = \{0\} \,\dot\cup\, \{(a, E_i\beta) : i \in K,\ a \in L_i,\ \beta \in P_j\}. \tag{8}$$

Its coefficient at 0 is $b_j(0)$, and its coefficient at $(a, E_i\beta)$ in the second set is $N^{-1}\varepsilon_i(a)b_j(\beta)$. Consequently,

$$S(F_{\mathcal{B}}) = \dot\bigcup_{j=1}^{r} (T_j \times \{e_j\}), \qquad |T_j| = 1 + N^2 p_j. \tag{9}$$

Proof. Expand $\mathbf{1}_{B_j}(Y_i)$ in its pointer Fourier basis. Its zero-frequency term is the same $b_j(0)$ for every i, so all zero-pointer contributions to H_j sum to

$$b_j(0) \sum_{i \in K} \mathbf{1}_{\mathcal{A}_i}(x) = b_j(0).$$

This cancels every nonzero pure address coefficient arising from those terms and leaves only the origin.

For a nonzero pointer frequency β, the term involving pointer i has frequency $(a, E_i\beta)$ and coefficient $N^{-1}\varepsilon_i(a)b_j(\beta)$ by Lemma 3.3. Different pairs (i, β) with $\beta \neq 0$ yield different pointer frequencies: a nonzero frequency in one block cannot equal a frequency in another block. Thus there are no further additions between these terms. The coefficient is nonzero exactly when $a \in L_i$ and $\beta \in P_j$. This proves (8) with the stated signs. Each i contributes Np_j elements, and there are N indices i, giving $|T_j| = 1 + N^2 p_j$.

Multiplication by $(-1)^{v_j}$ places the entire spectrum of H_j in data layer e_j. These data characters are distinct, so different layers cannot cancel. This proves (9). □

The origin in T_j is an *inner* frequency. Its full frequency in $F_{\mathcal{B}}$ is $(0, e_j)$, which is nonzero. This distinction will supply the strict advantage in the comparison score.

Corollary 5.3 (Initial sparsity)**.** *The function f_ℓ has exactly $R(1 + N^2(R-1)) = R^6 - R^5 + R$ nonzero Fourier coefficients.*

Proof. Initially the blocks are singletons. For a singleton $\{s\}$, $b_j(\beta) = R^{-1}(-1)^{\beta(s)}$ is nonzero for every β, so $p_j = R - 1$. Sum (9) over the R data layers and use $N = R^2$. □

6 Strict dominance of table comparisons

All scores in this section concern the actual current function $F_{\mathcal{B}}$. For subsets $Q, T \subseteq B^* \setminus \{0\}$, define

$$\mathcal{T}_Q = \{0\} \,\dot\cup\, \{(a, E_i\beta) : i \in K,\ a \in L_i,\ \beta \in Q\}.$$

For $z \in U$, write

$$I(Q, T; z) = |\mathcal{T}_Q \cap (\mathcal{T}_T + z)| = |\{u \in \mathcal{T}_Q : u + z \in \mathcal{T}_T\}|.$$

Translation by z is a bijection between the two orientations, so $I(Q, T; z) = I(T, Q; z)$.

Lemma 6.1 (All nonzero inner shifts)**.** *For every $z \neq 0$ in U,*

$$I(Q, T; z) \leq N(|Q| + |T|). \tag{10}$$

For a nonzero pure address shift $z = (a, 0)$, more precisely,

$$I(Q, T; (a, 0)) = N|Q \cap T|\,|\{i : a \in L_i\}| \leq 2N|Q \cap T|. \tag{11}$$

Proof. First suppose $z = (a, \eta)$ has $\eta \neq 0$. Choose a pointer coordinate (i, t) at which η is one. If $u \in \mathcal{T}_Q$ and $u + z \in \mathcal{T}_T$, at least one of these two endpoints has coordinate (i, t) equal to one. Every member of $\mathcal{T}_Q$ with that coordinate one is in pointer block i, and there are at most $N|Q|$ such members. Likewise, at most $N|T|$ members of $\mathcal{T}_T$ have that coordinate one. The collision set is contained in the union of the first set and the translate by z of the second set, proving (10). This cover includes the origin whenever it is an endpoint: the other endpoint then has the chosen coordinate equal to one. The argument permits any address component and any number of nonzero pointer blocks.

It remains to consider $z = (a, 0)$ with $a \neq 0$. The origin cannot be an endpoint of a collision, because the only zero-pointer element of either layer is the origin. Both endpoints therefore have the same nonzero pointer frequency, forcing a common block i and a common $\beta \in Q \cap T$. Their address frequencies lie in $L_i \cap (L_i + a)$, which is L_i if $a \in L_i$ and empty otherwise. Counting gives the equality in (11). The incidence bound in Lemma 3.2 gives its inequality. Finally, $2|Q \cap T| \leq |Q| + |T|$, proving (10) also in this case. □

Lemma 6.2 (Exhaustive data-layer scores)**.** *Write a query frequency as (z, d), with $z \in U$ and $d \in (\mathbb{F}_2^r)^*$. Then*

$$C_{F_{\mathcal{B}}}(z, 0) = \sum_{j=1}^{r} I(P_j, P_j; z), \tag{12}$$

$$C_{F_{\mathcal{B}}}(z, e_j + e_h) = 2I(P_j, P_h; z) \quad (j \neq h), \tag{13}$$

$$C_{F_{\mathcal{B}}}(z, d) = 0 \quad (d \notin \{0\} \cup \{e_j + e_h : j \neq h\}). \tag{14}$$

Proof. By (9), the supported data characters are exactly the e_j. A collision from layer e_j to layer e_h requires $d = e_j + e_h$. If $d = 0$, this forces $j = h$, giving (12). If $d \neq 0$, it must have ones at exactly two distinct coordinates j, h. Only the orientations (j, h) and (h, j) are then possible. They have equal cardinalities by translation, proving (13). All other d give no collisions. □

Proposition 6.3 (Uniform bound away from pure comparisons)**.** *For every query with nonzero inner part z,*

$$C_{F_{\mathcal{B}}}(z, d) \leq 2NP. \tag{15}$$

For $z = 0$ and d neither zero nor a sum of two distinct data characters, the score is zero.

Proof. For $d = 0$, Lemmas 6.1 and 6.2 give $\sum_j N(2p_j) = 2NP$. For $d = e_j + e_h$ with $j \neq h$, they give $2N(p_j + p_h) \leq 2NP$. Every remaining nonzero data component has zero score by (14). That same formula proves the final assertion when $z = 0$. These cases exhaust all characters except the excluded zero query and the pure comparisons $(0, e_j + e_h)$. □

Lemma 6.4 (A strictly better pure comparison)**.** *If $2 \leq r \leq R$, there are distinct j, h with*

$$C_{F_{\mathcal{B}}}(0, e_j + e_h) > 2NP. \tag{16}$$

Proof. The zero inner shift gives

$$C_{F_{\mathcal{B}}}(0, e_j + e_h) = 2(1 + N^2|P_j \cap P_h|). \tag{17}$$

Indeed, $\mathcal{T}_{P_j} \cap \mathcal{T}_{P_h}$ contains the origin, and for each of the N pointer blocks it contains N address frequencies for each $\beta \in P_j \cap P_h$. The factor two is the two orientations in (13).

Double-counting the common nonzero pointer frequencies yields

$$\sum_{j<h} |P_j \cap P_h| = \sum_{\beta \neq 0} \binom{d_\beta}{2} \geq \frac{1}{2} \sum_{\beta \neq 0} d_\beta = \frac{P}{2}.$$

The inequality uses $d_\beta = 0$ or $d_\beta \geq 2$, from Lemma 5.1. There are $r(r-1)/2$ unordered pairs, so at least one has

$$|P_j \cap P_h| \geq \frac{P}{r(r-1)}.$$

Since $N = R^2 \geq r(r-1)$ and $r(r-1) > 0$, its score is at least

$$2 + \frac{2N^2 P}{r(r-1)} \geq 2 + 2NP > 2NP.$$

The contribution of the common inner origin gives the strict inequality. □

Theorem 6.5 (All-partition dominance)**.** *For every partition $\mathcal{B}$ of B into $2 \leq r \leq R$ nonempty blocks, every maximizing nonzero character for the current function $F_{\mathcal{B}}$ is exactly*

$$(0, e_j + e_h) \qquad \textit{for some distinct } j, h.$$

Proof. Lemma 6.4 supplies a query whose score is strictly larger than $2NP$. Proposition 6.3 bounds every nonzero character outside the displayed set by $2NP$. No such character can maximize. The strictness also excludes ties between a comparison and any mixed, address, pointer, or other data query. Ties among the pure comparisons are unrestricted. □

This theorem holds for every table partition. It does not assert that the spread geometry survives every restriction of the address variables. The branch used next leaves those variables free, which is exactly the condition under which the displayed current-function formulas apply.

7 Every adaptive policy has a long feasible branch

Lemma 7.1 (Exact equality restriction)**.** *If the current partition is $\mathcal{B}$ and the query is $v_j + v_h$, the answer zero produces $F_{\mathcal{B}'}$, where $\mathcal{B}'$ replaces B_j, B_h by their union and retains all other blocks.*

Proof. The answer says $v_j = v_h$. In (6), their two terms become $(H_j + H_h)(-1)^{v_j}$. Since the blocks are disjoint, $\mathbf{1}_{B_j} + \mathbf{1}_{B_h} = \mathbf{1}_{B_j \cup B_h}$, so $H_j + H_h$ is exactly the layer associated to the union block. All other terms are unchanged. The union's pointer coefficient is $b_j(\beta) + b_h(\beta)$, which may vanish. Its new set of nonzero frequencies is computed from this sum, as in (7). No projected or uncancelled old support is substituted for the new one. □

Proposition 7.2 (Long zero-table paths)**.** *For every maximizing policy for f_ℓ and every fixed choice of address and pointer bits, the run on the input with those inner bits and with $w_s = 0$ for all $s \in B$ has length at least R.*

Proof. Its finite run exists by Lemma 2.3. Initially the current table partition has R singleton blocks. Suppose that $r \geq 2$ blocks remain along the specified input's path. In the canonical current chart, Theorem 6.5 forces the policy's query to have intrinsic character $v_j + v_h$ for some distinct blocks, regardless of the history or tie choices. The all-zero table satisfies the equality answer. By Lemma 7.1, the new domain is precisely that for the partition with these two blocks merged. Address and pointer variables remain free, and the number of independent data bits decreases by exactly one.

The domain cannot be a constant leaf at any such stage. For example, setting every free data bit to zero makes the function $+1$ everywhere in the inner variables, whereas setting every free data bit to one makes it -1. Both assignments obey the current equality constraints. Thus the induction continues for $R - 1$ comparisons.

After these comparisons the partition has one block, namely B. Formula (6) becomes $F_{\mathcal{B}} = (-1)^{v_1}$, because its sole inner layer is identically one. This is still nonconstant, so at least one additional query is required before the run can reach a constant leaf. The total length is at least R.

To check chart freedom, an arbitrary chart may express the same restricted ambient query as $v_j + v_h + c$. The specified input then follows answer c, which imposes the same equality. Lemma 2.1 preserves all maximizers under the coordinate change. In the canonical chart the current domain consists solely of homogeneous table equalities and contains the ambient zero input. An ambient linear parity with intrinsic form $v_j + v_h$ therefore has no constant offset there; any offset in another chart merely relabels answers. Hence the branch is a branch of the actual policy run and is realized by the fixed original input. This also explains why an adaptive choice of chart cannot circumvent the induction. □

Proof of Theorem 1.1. The function is the explicitly specified Boolean function (5); its dimension and short tree are given in Section 4. Corollary 5.3 gives its exact sparsity. Lemma 2.3 applies to every policy and proves existence, uniqueness, termination, and correctness of its run. Proposition 7.2 gives the R lower bound on a feasible path of every such run. It remains to check the numerical comparison.

For $R \geq 2$,

$$k_\ell - R^5 = R^5(R-2) + R \geq 0, \qquad R^6 - k_\ell = R^5 - R > 0.$$

Thus

$$R^5 \leq k_\ell < R^6, \qquad k_\ell^{1/6} < R, \qquad 5\ell = \log_2(R^5) \leq \log_2 k_\ell.$$

The lower bound $k_\ell \geq 2^{5\ell}$ also proves divergence. □

The final remaining character has singleton support and all nonzero translation scores equal to zero. A policy is allowed to make additional queries in this situation. The proof uses only that it cannot stop while the function remains nonconstant; Lemma 2.3 guarantees eventual termination even for such zero-score ties.

8 Lean verification and reproducibility

The complete mathematical separation is formalized in Lean, including the current signed spectra, the classification of all maximizing queries, arbitrary merge sequences, and the operational semantics of every policy. The development uses normalized rational Walsh coefficients. The declarations `Walsh.realTransform_source_formula` and `Walsh.real_transform_ne_zero` identify their values and support with the real transform used in this paper. All theorem names below are in the namespace `AdaptiveParity`, except where the additional namespace `ParityTree` or `Walsh` is shown.

Main endpoints and statement match. The final declarations are

`AdaptiveParity.polynomial_family_separation`
`AdaptiveParity.polynomial_infinite_separation`.

The first quantifies over every positive integer ℓ and over every `MaxPolicy` for the specified `polynomialFunction`. It asserts Booleanity, the exact sparsity formula, nonemptiness of the policy class, the policy run's greedy property and correctness, finite depth, a feasible path longer than $k^{1/6}$, and an unrestricted tree of depth at most $\log_2 k$. The second assembles the infinite-family statement and diverging sparsity. The sharper intermediate path bound 2^ℓ is `representationGreedy_zero_table_path`; the explicit short tree and its 5ℓ bound are used in `representation_separation`.

Paper statement	Principal formal component
Specified polynomial field	`binary_test_iff_irreducible`, `explicitPolynomial_first`, `polynomialCoordinates_coefficient`, `polynomialCoordinates_reconstruct`
Function and input dimension	`polynomial_function_formula`, `polynomial_input_dimension`
Current spectra and dominance	`SpreadSpectrum.lean`, `LayerCollision.lean`, `AllQueryScores.lean`, `representation_current_maximizer`
Every operational policy	`ParityTree.policy_unroll`, `ParityTree.policyTree_unique`, `ParityTree.policyRun_greedy`
Feasible lower-bound path	`representationGreedy_zero_table_path`
Complete separation	`polynomial_family_separation`, `polynomial_infinite_separation`

The representation layer first proves the geometry and separation for any finite field with specified binary coordinates. The final instance is the quotient by the least-code irreducible polynomial, with its canonical power basis. The coordinate and reconstruction identities ensure that this is the representation in Section 3, rather than an unspecified choice of a field basis. Polynomial selection is an executable finite coefficient test. The semantic quotient implementation and several existence constructions in the proof library are noncomputable Lean definitions; the theorem does not assert a compiled end-to-end evaluator or a formal machine-model running-time bound.

Pins and recorded checks. The toolchain is `leanprover/lean4:v4.27.0`; the mathlib commit is

`a3a10db0e9d66acbebf76c5e6a135066525ac900`.

The accepted source-only reproduction rebuilt all 42 authored modules and recorded 408 transitive axiom queries, including the main endpoints. The only reported axioms were `propext`, `Classical.choice`, and `Quot.sound`, or no axioms for some declarations. There are no `sorry` or `admit` gaps, newly postulated mathematical axioms, native-computation proof axioms, or external certificate premises in the accepted result. Precompiled objects for the pinned dependencies were reused; the dependency library itself was not rebuilt in that reproduction. A separate two-input implementation smoke test is not used as a premise of the theorem.

Artifact layout and commands. In the associated research repository the preserved proof development is under `research/three-more-tcs-2026-10-14/02-adaptive-parity/`, with Lean sources in `lean/`. From that research directory the recorded verification commands are

```
python3 lean/clean_build_checkpoint5.py
python3 lean/source_integrity.py 5
```

The first rebuilds the authored modules in dependency order and checks the axiom output; the second audits source and dependency integrity. These historical scripts use host-specific compiler and dependency-cache locations. For a different installation, those location settings must be pointed to the same pinned toolchain and packages. They are not additional mathematical assumptions. The saved `lean/lean-toolchain`, verification metadata, and dependency manifest entries record the exact versions.

The principal saved outputs are `lean/clean-build-checkpoint5.json`, `lean/AuditCheckpoint5-verification.log`, and `lean/axiom-summary-checkpoint5.json`. The statement audit is `PARENT_FINAL_STATEMENT_AUDIT.txt`, and final acceptance is recorded in `PARENT_ACCEPTANCE.json`. A separate source-only reproduction is recorded at the repository-relative path `research/three-more-tcs-2026-10-14/parent-experiments/parity-reproduction/20261014T112415Z/PARENT_REPRODUCTION.json`. The paper's `PAPER_AUDIT.json` lists source hashes, the correspondence between its claims and these formal declarations, and citation evidence.

9 Scope of the separation

The obstruction depends on three features acting together. The spread partition keeps nonzero address-direction incidence at most two. Independent pointers place each cell's nonconstant frequencies in a distinct block. The shared table makes equality answers merge arbitrary subsets of the pointer alphabet, whose indicator coefficients still sum to zero at each nonzero frequency. This last identity supplies a better comparison at every stage, even when individual coefficients disappear under merging.

The conclusion is a worst-case depth separation for maximum current support-overlap. The construction has a logarithmic-depth unrestricted tree, so it gives no lower bound against the XOR log-rank conjecture. It also says nothing against rules based on coefficient magnitudes, look-ahead, or other query objectives. Its adaptive quantifier is precise: every permitted policy has a long realized branch, and every partition reached along that branch is covered by the same signed-spectrum and strict-dominance proof.

References

[1] Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 658–667, 2013. doi:10.1109/FOCS.2013.76. Full version: https://arxiv.org/abs/1304.1245.

[2] Nikhil S. Mande and Swagato Sanyal. On parity decision trees for Fourier-sparse Boolean functions. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 182 of *Leibniz International Proceedings in Informatics*, pages 29:1–29:16, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.29.

[3] Hamed Hatami, Kaave Hosseini, Shachar Lovett, and Anthony Ostuni. Refuting approaches to the log-rank conjecture for XOR functions. In *51st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 297 of *Leibniz International Proceedings in Informatics*, pages 82:1–82:11, 2024. doi:10.4230/LIPIcs.ICALP.2024.82.

[4] Yuriy Tarannikov. On a Boolean function without bold folding in the spectrum support and implications for greedy approaches to PDT depth. arXiv:2607.04806v1, July 6, 2026. https://arxiv.org/abs/2607.04806.