THE MISMATCH BETWEEN NEURAL AFFINITY AND SYMBOLIC SIMPLICITY

Anonymous authors
Paper under double-blind review

ABSTRACT

A large body of work on simplicity bias shows that neural networks favor simple functions. Because the target functions in many real problems are expected to be highly structured, this preference is believed to aid generalization. But does this simplicity bias still hold in the space of symbolic rules? In other words, is the "simplicity" that networks prefer the "simplicity" that humans find easy to understand? We prove a mismatch between the two and establish that it is irreducible: almost every Boolean function has a longer description than parity yet sits at a lower frequency, and even at matched frequency, description lengths range from logarithmic to exponential. Experiments confirm the mismatch: what is cheap to describe is not what is cheap to learn. Even when a compact parity rule and a randomly drawn function fit the training labels equally well, MLPs learn the random function first, and so does a Transformer. These findings challenge the expectation that symbolically simple targets benefit from neural simplicity bias: short symbolic rules can instead face a learning disadvantage.

1 INTRODUCTION

Several works have proposed simplicity bias — the tendency of standard network training to find low-complexity solutions — (Arpit et al., 2017; Nakkiran et al., 2019; Valle-Pérez et al., 2019). A closer look at neural networks through the lens of Fourier analysis exposes this bias: networks prioritize learning the low-frequency modes (Rahaman et al., 2019; Cao et al., 2021). Because the rules behind real data are expected to be simple, this bias is widely trusted to do useful work: when training data can be explained equally well by two rules, we expect the network to converge to the simpler one.

The expectation breaks in a minimal experiment (Figure 1): when both candidates fit every training label, the network learns the random table first rather than the compact parity rule. This observation matches the frequency-domain prediction: a random table distributes its energy broadly across frequencies, while parity concentrates all of its energy at the highest frequency. The network prefers the spectrally simpler candidate, not the symbolically simpler one. For a human the verdict re-

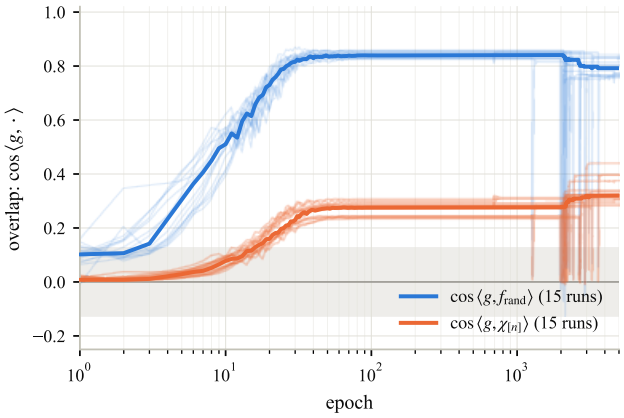


Figure 1: Each training label is the value parity and a random lookup table both assign to an 8-bit input. Thin lines show the cosine overlap with each candidate in 15 runs; bold lines are medians, and the shaded band is the chance baseline. Every run converges to the random table; parity never takes over within the training budget (5,000 epochs).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

verses: we make sense of the world through symbolic descriptions, and a random table has neither the simplicity nor the interpretability such understanding requires.

The two notions of simplicity were never the same: symbolic complexity measures how short a rule's description can be, and frequency content shows where a function's energy sits. Ordering reversals are unavoidable: some functions have longer descriptions but lower frequency than others.

We therefore keep three coordinates distinct: symbolic complexity, the frequency-energy profile, and neural affinity — a network's learning advantage for a function under a fixed training protocol. Within these coordinates, the paper makes four contributions.

First, we **verify spectral bias in the Boolean input space**: a ReLU MLP trained on an eight-bit target with one Fourier component at every frequency recovers the components strictly from the lowest frequency up.

Second, we **prove that the mismatch is irreducible**: no grammar in a natural class of description languages ranks functions by frequency, and within a narrow frequency band, description costs span from logarithmic to exponential.

Third, we **confirm the mismatch empirically**: in the fully enumerable four-bit function space, symbolic cost and low-frequency mass rank functions differently under all seven grammars; functions sharing one frequency-energy profile span a wide range of symbolic costs; and paired training controls show that learning difficulty follows the spectral coordinate, not the symbolic one.

Finally, we **demonstrate adverse consequences when rules compete**: under labels that two candidates explain equally well, the network initially commits to the random table, and the compact rule overtakes it only late, if at all — never, in almost every Transformer run.

2 Related Work

We review the most directly related work on inductive bias and on neural networks in rule discovery. A broader, per-paper discussion appears in Appendix J.

Spectral bias and its mechanism. Neural networks learn the low-frequency components of a target function before the high-frequency ones. Rahaman et al. (2019) name this tendency *spectral bias* and document it through Fourier analyses of neural networks; Xu et al. (2019) state it as a *frequency principle*, verified on high-dimensional real datasets. A theoretical line grounds the tendency in training dynamics: overparameterized training follows the kernel gradient of the neural tangent kernel (Jacot et al., 2018), whose spectrum sets the learning speed of each frequency (Basri et al., 2019; Cao et al., 2021; Bietti & Mairal, 2019), with extensions to non-uniform densities (Basri et al., 2020) and to spectral decompositions of the full learning process (Bordelon et al., 2020; Su & Yang, 2019; Yang & Salman, 2019). Within this line, low frequency and simplicity are used interchangeably. Rahaman et al. (2019) view their low-frequency finding as a first step towards formalizing the observation that deep networks *prioritize learning simple patterns*; Cao et al. (2021) read spectral bias as a bias towards "less complex functions". What these accounts measure is spectral; the simplicity gloss is an interpretation, and the symbolic complexity of a rule plays no part.

Simplicity bias and description length. A parallel line asks which functions a network prefers, and measures simplicity directly. Dingle et al. (2018) show that input–output maps in natural systems are strongly biased towards simple outputs; Valle-Pérez et al. (2019) trace generalization to a simplicity bias of the parameter–function map, with simplicity read out as Lempel–Ziv complexity; and Mingard et al. (2019; 2025) identify a prior towards low-entropy functions that outweighs their exponential rarity. Shah et al. (2020) show that the bias can be an outright pitfall — networks may rely on the simplest feature alone while ignoring more complex, equally predictive ones — but these studies lack a unified definition of simplicity. Behind these measures is description-length thinking: simplicity means a short description (Rissanen, 1978; Li & Vitányi, 2019), yet none of it brings in the perspective of symbolic rules, and the description length of a rule itself is never quantified. Our framework and experiments ask whether the compression-based measure and the spectral account agree on which functions are simpler.

Neuro-symbolic rule discovery. Symbolic rule discovery increasingly runs through neural networks. Systems search expression spaces directly (Schmidt & Lipson, 2009), use networks to aid the search (Udrescu & Tegmark, 2020; Petersen et al., 2021), or read the rule out of the trained network (Townsend et al., 2020): decision trees extracted by querying it (Craven & Shavlik, 1994; Andrews et al., 1995) or layer by layer (Zilke et al., 2016), equations from its own structure (Sahoo et al., 2018; Liu et al., 2025), and symbolic models distilled from it (Frosst & Hinton, 2017; Cranmer et al., 2020). If the two notions of simplicity diverge, a pipeline recovers the network's preferred rule, not the simplest one.

3 THE THREE COORDINATES

To state and test the mismatch, we first make the three coordinates precise. We use *function* for the mathematical input–output map and *rule* for the same map when emphasizing its symbolic specification.

3.1 THE SYMBOLIC COORDINATE: COMPLEXITY

The symbolic coordinate measures how concisely a function can be expressed. Within a fixed grammar, the shorter a rule's description, the easier it is for humans to understand (Feldman, 2000) and the more regular it is (Rissanen, 1978; Grünwald, 2007), so we measure symbolic complexity by the length of the shortest description.

Definition 3.1 (Symbolic complexity)**.** Fix a description language G with program set $\mathcal{P}_G$, evaluation map eval_G, and positive additive node costs c_G. The symbolic complexity of a function f is

$$C_{\mathrm{sym}}(f; G) = \min_{p \in \mathcal{P}_G :\, \mathrm{eval}_G(p) = f} c_G(p),$$

with value ∞ if no program in G computes f.

3.2 THE SPECTRAL COORDINATE: PROFILES AND THEIR SCALAR PROJECTIONS

The spectral coordinate describes how a function's energy is distributed across frequencies in the Fourier basis (O'Donnell, 2014); we read a rule $f : \{0,1\}^n \to \{0,1\}$ as $g : \{\pm 1\}^n \to \{\pm 1\}$ through $g(x) = 1 - 2f((1-x)/2)$, applied coordinatewise.

Definition 3.2 (Frequency-energy profile)**.** Let $g : \{\pm 1\}^n \to \{\pm 1\}$ and let μ be the uniform input measure. For $S \subseteq [n]$, where $[n] = \{1, \ldots, n\}$, define the Fourier character $\chi_S(x) = \prod_{i \in S} x_i$ and the normalized coefficient

$$\hat{g}(S) = \mathbb{E}_{x \sim \mu}[g(x)\chi_S(x)] = 2^{-n} \sum_{x \in \{\pm 1\}^n} g(x)\chi_S(x).$$

The frequency-energy profile is $(E_0(g), \ldots, E_n(g))$, where

$$E_k(g) = \sum_{|S| = k} \hat{g}(S)^2, \qquad k = 0, \ldots, n.$$

The characters form an orthonormal basis under the inner product induced by μ, so Parseval's identity gives $\sum_{k=0}^{n} E_k(g) = 1$.

Table 1 defines the scalar projections, with the full profile as our primary descriptor.

3.3 THE AFFINITY COORDINATE: LEARNING DYNAMICS

Neural affinity is a network's learning advantage for a function under a training protocol; we measure it by training difficulty D, and larger D indicates lower affinity.

Definition 3.3 (Training difficulty)**.** For a classification target f, fix a training protocol $\mathcal{T}$ specifying the network, data sampling, optimizer, and evaluation measure, together with an accuracy threshold τ. For a particular training run, let $a_t(f; \mathcal{T})$ denote evaluation accuracy after epoch t. The training difficulty of the target f is

$$D_\tau(f; \mathcal{T}) = \inf\{t \in \mathbb{N}_0 : a_t(f; \mathcal{T}) \geq \tau\},$$

with value ∞ if the threshold is never reached. We suppress τ and $\mathcal{T}$ when fixed.

Table 1: Scalar summaries of the frequency-energy profile. The cutoff k specifies which frequencies are grouped.

Symbol	Definition	Name
L_k	$\sum_{j=0}^{k} E_j$	low-frequency mass
T_k	$\sum_{j=k}^{n} E_j$	tail mass
$\bar{k}$	$\sum_{j=0}^{n} jE_j$	mean frequency

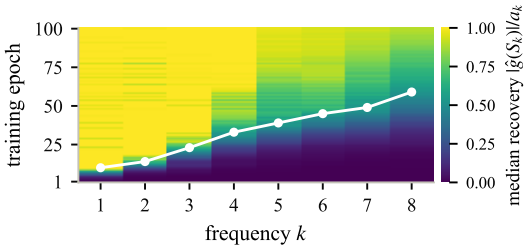


Figure 2: Spectral bias on the Boolean cube. A ReLU MLP is trained on the full domain $\{\pm 1\}^8$ to fit a real-valued target summing one unit-amplitude Fourier character at every frequency $k = 1, \ldots, 8$. Color shows the median recovery $|\hat{g}(S_k)|/a_k$ of each target character across five training seeds; the white curve marks 50% recovery, first reached at epochs $10, 14, 23, 33, 39, 45, 49, 59$.

Relation to the spectral coordinate. Spectral bias links neural affinity to frequency content (Rahaman et al., 2019; Cao et al., 2021). On $\{\pm 1\}^8$, a 6×256 ReLU MLP trained on a real-valued target summing eight Fourier characters, one at each frequency from 1 through 8, recovers them strictly in frequency order (Figure 2); Appendix C gives the details.

4 The Mismatch Is Irreducible

Knowing that neural affinity follows spectral structure, we ask whether symbolic simplicity aligns with it. We first measure ordering reversals with Kolmogorov complexity (Section 4.1) and extend the reversal to explicit grammars (Corollary 4.1). We finally quantify how widely symbolic complexity varies within a narrow frequency band (Section 4.2).

Throughout, F_n is the space of 2^{2^n} Boolean functions on n bits, read as ±1-valued through the mapping of Section 3.2, with characters χ_S under the uniform input measure. Both theorems use the mean frequency $\bar{k}$ of Table 1.

4.1 Measuring ordering reversals with Kolmogorov complexity

In place of the grammar-dependent C_{sym} of Section 3.1, we price functions by Kolmogorov complexity, which involves no grammar, and recover the conclusion for explicit grammars in Corollary 4.1. Fix an optimal universal machine U. The *plain Kolmogorov complexity* $K(f)$ is the length in bits of the shortest binary program that makes U output the complete truth table of f in a fixed input order (Li & Vitányi, 2019); optimality means that U can simulate any fixed computable decoder with constant additive overhead in program length. Under this benchmark, parity — the character $\chi_{[n]}(x) = \prod_{i=1}^{n} x_i$ on all n coordinates — is cheap while its mean frequency is maximal: a fixed program prints $\chi_{[n]}$ from an $O(\log n)$-bit encoding of n, so

$$K(\chi_{[n]}) \le \log_2 n + c_0$$

for a constant c_0 depending only on U (Appendix A). Say that a description cost $L : F_n \to \mathbb{N}\cup\{\infty\}$ is *order-preserving* with respect to $\bar{k}$ if $\bar{k}(f) > \bar{k}(g)$ implies $L(f) \ge L(g)$ for all $f, g \in F_n$.

Theorem 1 (Ordering reversals for Kolmogorov complexity)**.** *There is n_1, depending only on U, such that for all $n \ge n_1$ the reversal set $\mathcal{R}_n = \{f \in F_n : K(f) > K(\chi_{[n]}),\ \bar{k}(f) < \bar{k}(\chi_{[n]})\}$ occupies at least a $1 - 2^{-n} - 2^{1-2^n}$ fraction of F_n; in particular, K is not order-preserving with respect to $\bar{k}$, and the fraction tends to one as $n \to \infty$.*

Parity attains the maximal mean frequency $\bar{k}(\chi_{[n]}) = n$, while a counting argument shows that almost every $f \in F_n$ needs a description of length $\Omega(2^n)$ and, outside the pair $\pm\chi_{[n]}$, lies below the top of the spectrum; Appendix A gives the full proof.

To reach the coordinate our experiments measure, we extend the reversal to explicit grammars. Call a grammar family *lean* if a single grammar is fixed independently of n, or if its operator basis and constants grow at most polynomially in n; Appendix A states the grammar and cost model precisely.

Corollary 4.1 (Grammars inherit the mismatch)**.** *Let (G_n) be lean, and let a polynomial p be fixed before n. If $L_{G_n}(\chi_{[n]}) \le p(n)$ for all n — parity priced polynomially in the grammar's own scale — then there is n_2, depending on the family and on p, such that for all $n \ge n_2$: L_{G_n} is not order-preserving with respect to $\bar{k}$ on F_n.*

The proof, in Appendix A, reruns the same counting argument in the grammar's own units, counting derivation trees instead of programs. The polynomial-cost hypothesis covers parity built from XOR and parity over the De Morgan basis AND, OR, NOT, both admitting polynomial-size formulas with fixed operator costs. Changing the grammar among lean families that price parity polynomially cannot eliminate the mismatch.

4.2 Complexity dispersion within frequency bands

We now return to Kolmogorov complexity and ask how widely symbolic complexity can vary within a narrow band of mean frequency.

Definition 4.1 (Frequency band)**.** For an integer $v \in \{0, \dots, n\}$, define

$$B_v = \{f \in F_n : |\bar{k}(f) - v| \le 1\}.$$

Theorem 2 (Complexity dispersion within frequency bands)**.** *There is an integer $m_0 \ge 16$, depending only on U, such that for every $n \ge 4$, every $v \in \{1, \dots, n-1\}$, and every admissible scale m — every integer with $\max\{m_0, \Theta(\log n)\} \le m \le \min\{2v,\, 2(n-v)\}$, made explicit in Appendix A.2 — the band B_v contains at least $(1 - \varepsilon_m)\, 2^{2^m}$ functions f satisfying*

$$2^m - \lceil m/2 \rceil \le K(f) \le \tfrac{3}{2}\, 2^m,$$

where ε_m is an explicit error term, depending only on m, that tends to zero as $m \to \infty$.

The proof, in Appendix A.2, multiplies a uniformly random table on m coordinates by a disjoint character of order $v - \lceil m/2 \rceil$; this character shift moves the family's mean frequency into the band B_v. An explicit encoding and a counting argument then pin $K(f)$ between the two bounds; the error ε_m covers the functions that leave the band or compress below the lower bound.

Within one band, short descriptions are evident: the characters χ_S with $|S| = v$ cost at most $\log_2 \binom{n}{v} + O(\log n)$ bits. Theorem 2 guarantees the long end: at every admissible scale m, the band contains functions with descriptions of order 2^m. Thus, at matched mean frequency, symbolic complexity can vary by many scales; this complements the ordering reversals above.

5 Empirical Analysis of the Mismatch

Section 4 established that the mismatch between symbolic and spectral simplicity is irreducible in principle. This section tests whether it is also consequential in practice. We first measure the mismatch in the fully enumerable four-bit function space under explicit description grammars, and then ask which of the two coordinates predicts training difficulty and the choice between competing rules. Across the experiments, difficulty and initial candidate preference track the spectral coordinate rather than the symbolic one.

5.1 Symbolic–spectral mismatch

To measure the mismatch exhaustively, we take as our analysis space all $65{,}536$ four-bit Boolean functions under the uniform input measure — small enough that every function's symbolic cost and frequency-energy profile can be computed exactly. For each function we compute the complete profile and, for each of seven description grammars, the minimum expression cost. The grammars

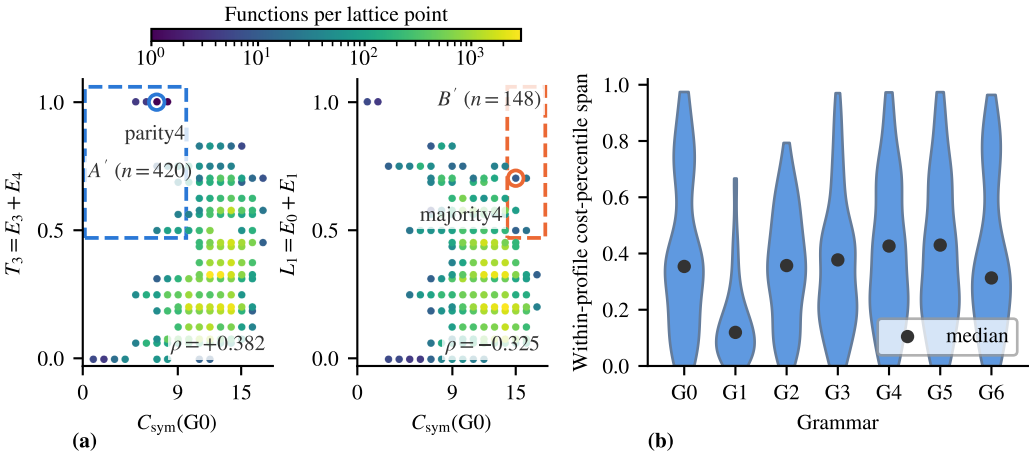


Figure 3: Symbolic–spectral mismatch across four-bit Boolean functions. **(a)** Scatter panels of baseline symbolic cost C_{sym} against T_3 and L_1; color indicates the number of functions at each plotted location on a logarithmic scale. Region A′ contains 420 functions with low C_{sym} and high T_3; region B′ contains 148 functions with high C_{sym} and high L_1. Parity4 and majority4 are highlighted as representative rules of these mismatch regions. **(b)** Violin plots of within-profile cost-percentile spans across the 138 profiles, one value per profile; black dots mark the median span per grammar.

draw on primitives used in logic, circuits, and programming, varying either the available operators or their costs; G0 is the baseline grammar used in the next experiment. Appendices B and G give the computational methods and grammar designs.

Full-space association. We measure global ranking alignment in the full function space using the Spearman correlation between symbolic cost C_{sym} and low-frequency mass L_1. A correlation of -1 would mean perfect agreement between the two notions of simplicity: functions with greater low-frequency mass would have lower symbolic costs. Under the baseline grammar G0, the correlation is -0.32 (Figure 3(a)). Across all seven grammars, the correlations range from -0.55 to -0.32, remaining far from -1; the ranking mismatch therefore persists across common symbolic description languages.

Complexity variation at a fixed profile. We next ask how widely C_{sym} can vary when the full profile $(E_0, \ldots, E_4)$ is held exactly fixed. For each profile with at least 30 functions (together 99.9% of the function space), we compute the difference between its largest and smallest cost percentiles.

Under G0, the median within-profile span is 0.35, the profile-size-weighted median is 0.72, and the maximum is 0.97. Thus, identical profiles can accommodate symbolic costs spanning nearly the entire percentile scale. Across six of the seven grammars, median spans lie between 0.31 and 0.43, and approximately one-fifth to one-third of profiles have spans of at least 0.5 (Figure 3(b)). Removing XOR (G1) produces the strongest global alignment and reduces the median span to 0.12, but its maximum span remains 0.67. Large within-profile spans are not an artifact of one grammar: they survive even the most aligned one.

5.2 Learning difficulty under symbolic and spectral controls

To examine how C_{sym} and frequency-energy structure each relate to learning difficulty, we hold one fixed while varying the other, selecting targets from the balanced four-variable Boolean functions. At fixed C_{sym}, we contrast the four lowest-T_3 against the four highest-T_3 functions within each of 12 cost groups; at fixed full profile, we sample functions at the C_{sym} extremes of 18 profiles and pair them. Each target is embedded in twelve-dimensional inputs whose labels depend only on the first four coordinates. We train ReLU and tanh MLPs ($12 \to 128 \to 128 \to 1$) and use the training difficulty D of Section 3 with $\tau = 0.99$; target-selection and training details are in Appendix D.

Table 2: Mean learning-time contrasts, ΔD (epochs). Brackets give 95% bootstrap confidence intervals over the 12 C_{sym} groups or 18 profile groups; aggregation details are in Appendix D.

Fixed quantity	High − low	ReLU	tanh
C_{sym}	T_3	13.33 [8.02, 18.23]	16.67 [11.46, 21.88]
Full frequency-energy profile	C_{sym}	0.25 [−2.01, 2.31]	−0.43 [−3.12, 2.25]

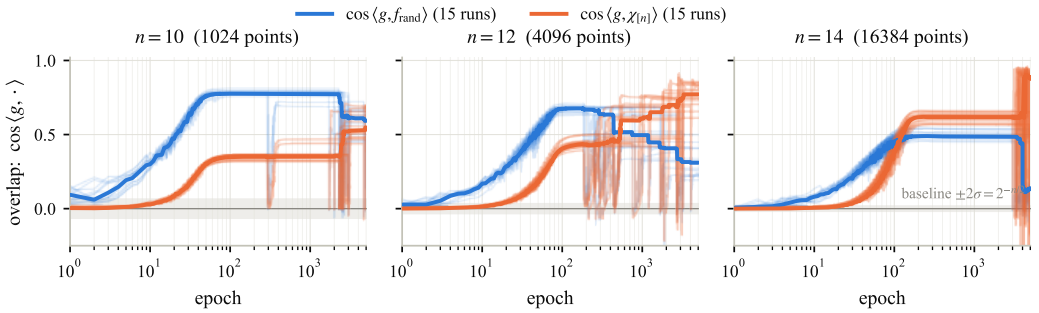


Figure 4: Learning under competing rules at the 95% accuracy level. Each panel shows the full-domain cosine overlap with the random candidate (f_{rand}) and with parity ($\chi_{[n]}$) over 5,000 epochs: thin lines are the 15 individual runs, bold lines are medians, and the shaded band is the chance baseline ($\pm 2\sigma$).

We report ΔD, the high-minus-low difference in learning time within each group, averaged across groups: high versus low T_3 at fixed C_{sym}, and high versus low C_{sym} at fixed profile (Table 2).

For both activations, higher T_3 is associated with longer learning times on average at fixed C_{sym}, whereas the learning-time difference between high- and low-C_{sym} targets is not statistically distinguishable from zero at a fixed full frequency-energy profile. Learning difficulty thus tracks spectral structure; at a fixed profile, symbolically simpler rules show no learning advantage.

5.3 Learning under competing rules

We now let the two coordinates compete. We pair parity — a compact rule at the highest frequency — with a randomly drawn Boolean function, and construct training labels that the two candidates explain equally well; the question is which one the network learns. The pair is the ordering reversal of Theorem 1: by Lemma A.1, the random function has, with probability $1 - 2^{1-2^n}$, a lower mean frequency than parity. Its description is far longer — a counting bound guarantees this with high probability (Appendix A), and incompressibility checks on the drawn tables confirm it (Appendix H).

Equally supported candidates. All inputs on which parity and the random candidate agree enter training with their common label. We then add an even number of disagreement inputs, half labeled by each candidate, so that both fit the same number of labels. We cover $n \in \{8, 10, 12, 14\}$, accuracy levels in $\{90\%, 95\%, 100\%\}$, and hidden activations in $\{\mathrm{ReLU}, \tanh\}$ (tanh at $n = 12$); full results appear in Appendix H.

Training and overlap. We train the 6×256 MLPs for 5,000 epochs and measure the full-domain cosine overlap between the network output and each candidate, which retains output amplitudes where sign accuracy discards them (Appendix H.4). Figure 4 compares the two candidates over training; architecture, run, and initialization details are in Appendix H.

Random first, parity late or not within budget. The observed trajectories initially favor the random candidate, consistent with preferential learning of lower-frequency structure. Parity can later cross above the random candidate; we call a run in which this crossing persists to the final epoch a takeover. Under ReLU it is the exception at $n = 10$ (7 of 15 runs) and the rule at $n = 12$

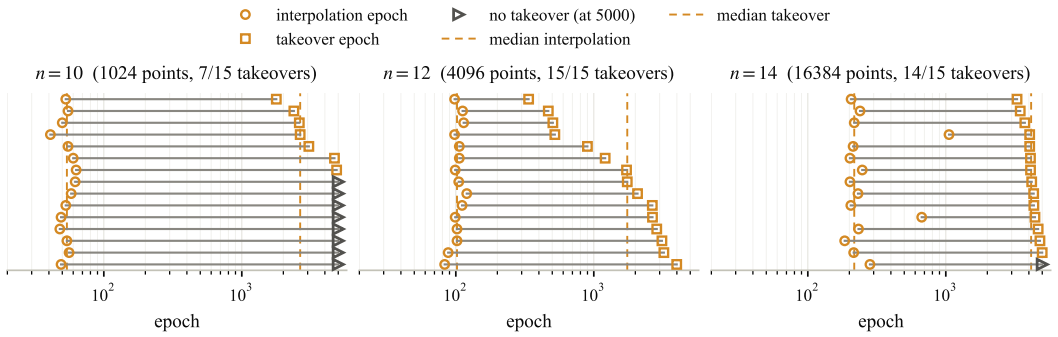


Figure 5: Fitting the labels versus switching candidate preference at the 95% accuracy level: per-run epochs of interpolation (first epoch of perfect training accuracy) and of parity takeover. Right triangles mark runs without takeover within the 5,000-epoch budget; dashed lines mark the medians.

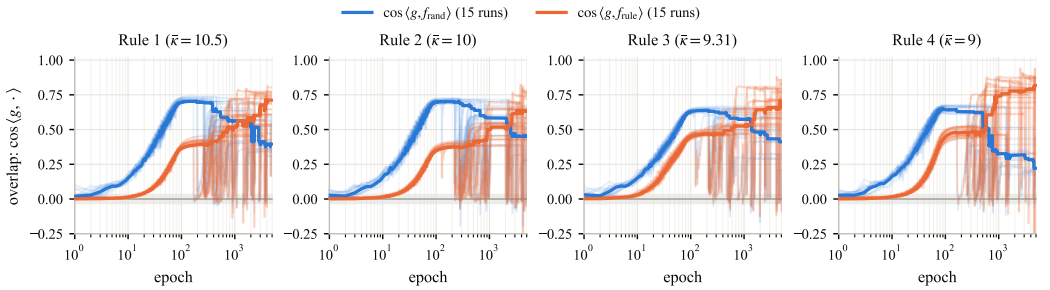


Figure 6: Competing rules beyond parity at $n = 12$, 95% accuracy level. Each panel shows the full-domain cosine overlap with the random candidate (f_{rand}) and with the rule (f_{rule}) over 5,000 epochs for a different compact rule (the title numbers the rule and gives its $\bar{k}$; Appendix H.7 gives the trajectories at both accuracy levels). Line conventions as in Figure 4.

and $n = 14$ (15 of 15 and 14 of 15), but it arrives late: median takeover epochs are 1,748 and 4,154 at $n = 12$ and $n = 14$. Under tanh activations the late takeover never arrives: at $n = 12$, none of the 45 runs takes over within the budget (Appendix H.6).

Figure 5 separates the epoch of fitting the labels from the epoch of favoring parity. At $n = 14$, the median interpolation epoch is 217, against a median takeover epoch of 4,154; every one of the 14 takeover runs switches after interpolation. If the short rule is recovered long after correct classification, recovering it is costly under a finite training budget.

Beyond parity. The competition so far pairs the random candidate with parity alone. We replace the rule with four other compact high-frequency rules at $n = 12$, bringing the tested $\bar{k}$ range to 9–12; Appendix H.7 gives the exact formulas. The random tables, protocol, and budget are unchanged. The random candidate is again learned first, and the rule takes over late in 12–15 of 15 runs across the four rules (Figure 6): the pattern is robust across compact high-frequency rules, not specific to parity.

Decoder-only Transformer extension. As a check of architecture robustness, we also run the competing-rule experiment at $n = 12$ at 90%, 95%, and 100% accuracy with a small decoder-only Transformer trained by next-token prediction on the same datasets, roughly parameter-matched with the reference MLP. The median trajectories again favor the random candidate first and remain on the random side at the end of the budget, with only one late takeover across the 27 runs (Figure 7). Architecture and protocol details are in Appendix H.8.

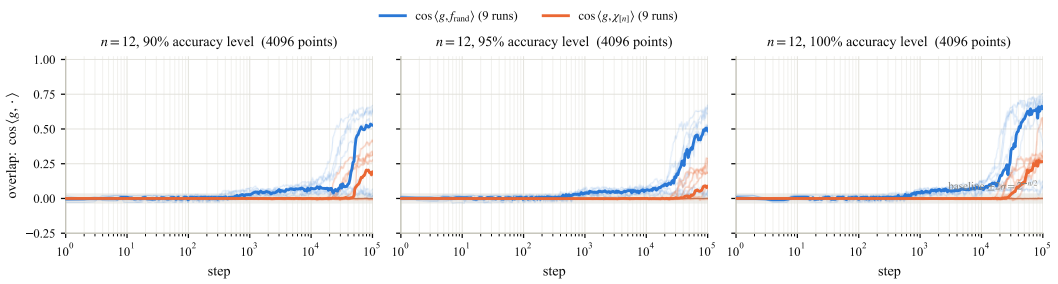


Figure 7: Decoder-only Transformer at $n = 12$ under the 90%, 95%, and 100% accuracy levels, nine runs per level; line conventions as in Figure 4, with the horizontal axis counting optimizer steps (budget 100,000). Flat lines are runs that remain at chance level.

6 Conclusion and Discussion

The mismatch between symbolic simplicity and neural affinity is provably irreducible and empirically consequential. We proved that compression itself ranks functions differently from frequency: the reversal is a theorem about Kolmogorov complexity, and the grammars we measure inherit it. In the fully enumerable four-bit space, learning difficulty follows the frequency-energy structure, and symbolic cost shows no detectable effect at a fixed profile. And in competition, the network initially prefers the random table over the compact rule every time. Together, these results show that a network's preferred completion reflects frequency content, not symbolic brevity.

Takeover and early stopping. We note that the compact rule can still win: at the larger dimensions, parity eventually overtakes the random candidate in most runs of the ReLU MLP — but only under training that runs long past the point where the labels are fit (Figure 5). This recovery would not occur under early stopping, and this is consistent with the spectral ordering of affinity: the low-frequency table is learned first, and the compact rule is reached only at the top of the frequency order, a stage that early stopping will preclude. Which training regimes, if any, let a compact high-frequency rule win within a realistic budget remains an open question.

Grammar dependence. The mismatch is not an artifact of one description language: it persists across the seven grammars we measure. The reversal is a property of symbolic compression itself, and a change of grammar only changes the price of individual rules. Claims that a rule is symbolically cheap must therefore name the language; no choice of language restores order between the coordinates.

Beyond the Boolean cube. The coordinates are not specific to Boolean functions: description length prices functions on any domain, and frequency content is defined once an input measure selects an orthonormal basis. Spectral bias was likewise established on real-valued domains, so the affinity–frequency link is no Boolean artifact. The mismatch analysis thus carries over to other function spaces.

Implications. We believe our diagnosis is useful for neural rule discovery: stop reading a network's preferred completion as the shortest explanation — the reading errs in a provable direction, toward spectrally simple, symbolically complex rules. Evaluate preference and brevity separately; neither yields the other. Whether pretraining, architecture, or curriculum can restore an advantage for symbolic simplicity is the open problem we set up.

Limitations. The empirical scope is bounded in three ways. First, exact measurements exist only in the four-bit space; larger dimensions enter through samples and fixed candidates. Second, we test fully connected MLPs with ReLU and tanh activations and one decoder-only Transformer; newer architectures and pretrained networks remain to be tested. Third, all experiments use synthetic Boolean functions; real data do not enter. Yet none of these boundaries touches the result: the mismatch is proved for all large n, and the analysis can be extended to other function spaces by determining the orthonormal bases of their input measures.

AI USE STATEMENT

We used large language models to aid and polish the writing of this paper: polishing the wording and unifying the notation. We did not use generative AI to develop the theory, construct the proofs, generate the datasets, design the experiments, or implement the code. We reviewed all AI-assisted text, checked every reported number against the experiment records, and verified all citations against the original papers. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Complete proofs of all theoretical claims appear in Appendix A, with the assumptions of each result stated where the result is introduced. The description language and its closure properties, the sampling and dataset-construction procedures, and the training and measurement protocols for every experiment are specified in Appendices B–I. All experimental code is included as supplementary material.

REFERENCES

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. In *Proceedings of the 35th Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pp. 4782–4887, 2022.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *Proceedings of the 36th Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 2552–2623, 2023.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252, 2019.

Robert Andrews, Joachim Diederich, and Alan B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332, 2019.

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242, 2017.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham M. Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems*, volume 32, pp. 4763–4772, 2019.

Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 685–694, 2020.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12873–12884, 2019.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1024–1034, 2020.

Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2205–2211, 2021.

Gregory J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13(4):547–569, 1966.

Miles D. Cranmer, Alvaro Sanchez-Gonzalez, Peter W. Battaglia, Rui Xu, Kyle Cranmer, David N. Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Mark W. Craven and Jude W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 37–45, 1994.

Amit Daniely and Eran Malach. Learning parities with neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Kamaludin Dingle, Chico Q. Camargo, and Ard A. Louis. Input–output maps are strongly biased towards simple outputs. *Nature Communications*, 9:761, 2018.

Simon S. Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Jacob Feldman. Minimization of Boolean complexity in human concept learning. *Nature*, 407: 630–633, 2000.

Leonardo Franco. Generalization ability of boolean functions implemented in feedforward neural networks. *Neurocomputing*, 70(1–3):351–361, 2006.

Ehud Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):27–35, 1998.

Nicholas Frosst and Geoffrey E. Hinton. Distilling a neural network into a soft decision tree. In *CEX Workshop at AI*IA*, 2017.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pp. 5–13. ACM, 1993.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8580–8589, 2018.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45 (6):983–1006, 1998.

Andrei N. Kolmogorov. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, 25(4):369–376, 1963.

Andrei N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8570–8581, 2019.

Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, 22(1):75–81, 1976.

Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 4th edition, 2019.

Henry W. Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, pp. 70367–70413, 2025.

Georg Martius and Christoph H. Lampert. Extrapolation and learning equations. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017.

Colin McDiarmid. On the method of bounded differences. In J. Siemons (ed.), *Surveys in Combinatorics, 1989: Invited Papers at the Twelfth British Combinatorial Conference*, volume 141 of *London Mathematical Society Lecture Note Series*, pp. 148–188. Cambridge University Press, Cambridge, 1989.

Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, and Ard A. Louis. Neural networks are a priori biased towards Boolean functions with low entropy. *arXiv preprint arXiv:1909.11522*, 2019.

Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A. Louis. Deep neural networks have an inbuilt occam's razor. *Nature Communications*, 16:220, 2025.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L. Edelman, Fred Zhang, and Boaz Barak. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems*, volume 32, pp. 3491–3501, 2019.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

Brenden K. Petersen, Mikel Landajuela, T. Nathan Mundhenk, Cláudio Prata Santiago, Sookyung Kim, and Joanne T. Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations (ICLR)*, 2021.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310, 2019.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Subham S. Sahoo, Christoph H. Lampert, and Georg Martius. Learning equations for extrapolation and control. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4439–4447, 2018.

Jürgen Schmidhuber. Discovering neural nets with low kolmogorov complexity and high generalization capability. *Neural Networks*, 10(5):857–873, 1997.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3067–3075, 2017.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 (3):379–423, 1948.

Ray J. Solomonoff. A formal theory of inductive inference, part i. *Information and Control*, 7(1): 1–22, 1964.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, and Nathan Srebro. The implicit bias of gradient descent on separable data. In *International Conference on Learning Representations (ICLR)*, 2018.

Lili Su and Pengkun Yang. On learning over-parameterized neural networks: A functional approximation perspective. In *Advances in Neural Information Processing Systems*, volume 32, pp. 2637–2646, 2019.

Joseph Townsend, Thomas Chaton, and João M. Monteiro. Extracting relational explanations from deep neural networks: A survey from a neural-symbolic perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3456–3470, 2020.

Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations (ICLR)*, 2019.

Timothy L. H. Watkin, Albrecht Rau, and Michael Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65(2):499–556, 1993.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint arXiv:1907.10599*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. Deepred – rule extraction from deep neural networks. In *Discovery Science*, pp. 457–473, 2016.

A PROOFS FOR SECTION 4

This appendix gives the full proofs of Theorems 1 and 2, Corollary 4.1, and their auxiliary lemmas. Throughout, $F_n = \{f : \{\pm 1\}^n \to \{\pm 1\}\}$, so $|F_n| = 2^{2^n}$; characters are $\chi_S(x) = \prod_{i \in S} x_i$; the normalized Fourier coefficients are $\hat{f}(S) = 2^{-n} \sum_x f(x)\, \chi_S(x)$, so that Parseval gives $\sum_S \hat{f}(S)^2 = 1$ and the registered views of Table 1 read $E_k = \sum_{|S|=k} \hat{f}(S)^2$ and $\bar{k}(f) = \sum_S |S|\, \hat{f}(S)^2$, with $0 \le \bar{k}(f) \le n$.

A.1 ORDERING REVERSALS AND THE GRAMMAR COROLLARY

Lemma A.1 (Top of the spectrum)**.** *$\bar{k}(f) = n$ if and only if $f \in \{\chi_{[n]}, -\chi_{[n]}\}$.*

Proof of Lemma A.1. Since $|S| \le n$ for every S, we can write $\bar{k}(f) = n - \sum_S (n - |S|)\, \hat{f}(S)^2$, a convex combination shifted by n; hence $\bar{k}(f) = n$ holds if and only if all Fourier weight sits on subsets of size n. The only such subset is $[n]$, so $\hat{f}$ vanishes off $[n]$ and $f = \hat{f}([n])\, \chi_{[n]}$. By Parseval, $|\hat{f}([n])| = 1$, and since $\hat{f}([n])$ is 2^{-n} times a sum of ± 1 values, it equals ± 1 if and only if $f(x)\, \chi_{[n]}(x)$ is constant on the cube, that is, $f = \pm \chi_{[n]}$. The converse is immediate. □

Description bound on parity. There is a constant c_0, depending only on the optimal universal machine U fixed in Section 4.1, such that for every $n \ge 1$,

$$K(\chi_{[n]}) \;\le\; \kappa(n) \;:=\; \log_2 n + c_0.$$

To obtain this bound, fix a decoder that reads the ordinary binary representation of n and prints $\prod_i x_i$ for each input in the prescribed order. Its input has $\lfloor \log_2 n \rfloor + 1$ bits. Simulating this decoder on U adds a constant overhead, which is absorbed into c_0.

Proof of Theorem 1. Fix U as in Section 4.1, and write $\mathrm{par} = \chi_{[n]}$ and $m = 2^n - n$.

1. *Counting short descriptions.* There are 2^j binary strings of length j, so the number of candidate programs of length strictly less than m is

   $$\sum_{j=0}^{m-1} 2^j = 2^m - 1.$$

   Every function $f \in F_n$ with $K(f) < m$ has at least one such program that outputs its complete truth table. Since a fixed program on U produces at most one truth table, distinct functions require distinct programs. Consequently,

   $$\big|\{f \in F_n : K(f) < 2^n - n\}\big| \le 2^{2^n - n} - 1 \le 2^{2^n - n}.$$

2. *Lower mean frequency.* After removing these short-description functions and the two functions par and $-\mathrm{par}$, the remaining set is

   $$\mathcal{H}_n = \{f \in F_n : K(f) \ge m\} \setminus \{\mathrm{par}, -\mathrm{par}\}.$$

   Step 1 gives $|\mathcal{H}_n| \ge |F_n| - 2^m - 2$. By Lemma A.1, every $f \in \mathcal{H}_n$ satisfies $\bar{k}(f) < n = \bar{k}(\mathrm{par})$.

3. *Longer descriptions than parity.* The description bound on parity gives $K(\mathrm{par}) \le \kappa(n) = \log_2 n + c_0$. For all sufficiently large n, $\kappa(n) < m = 2^n - n$. Hence every $f \in \mathcal{H}_n$ satisfies

   $$K(f) \ge m > \kappa(n) \ge K(\mathrm{par}).$$

   Together with Step 2, this makes every member of $\mathcal{H}_n$ an ordering reversal relative to parity.

4. *Fraction of reversals.* Let

$$\mathcal{R}_n = \{f \in F_n : K(f) > K(\mathrm{par}),\ \bar{k}(f) < \bar{k}(\mathrm{par})\}.$$

For all sufficiently large n, Steps 2–3 give $\mathcal{H}_n \subseteq \mathcal{R}_n$. Since $|F_n| = 2^{2^n}$, it follows that

$$\frac{|\mathcal{R}_n|}{|F_n|} \geq 1 - \frac{2^{2^n-n}+2}{2^{2^n}} = 1 - 2^{-n} - 2^{1-2^n} \longrightarrow 1. \tag{1}$$

The fraction is at most one, so it converges to one. Thus, among all n-variable Boolean functions counted uniformly, the fraction with longer descriptions but lower mean frequency than parity tends to one as $n \to \infty$.

For all n past a threshold depending only on U, this lower bound is positive, so $\mathcal{R}_n$ is nonempty and order preservation fails. This supplies the threshold n_1 in the theorem. □

A grammar G on n variables is a finite operator basis, each operator carrying an arity of at least one and a price of at least one; atoms are the n variables and finitely many constants, of unit price. The cost of a derivation tree is the sum of its node prices, and $L_G(f)$ is the minimum over trees computing f, infinite if there is none. A grammar family is *lean* if a single grammar is fixed independently of n, or if the basis and constants together grow at most polynomially in n.

Proof of Corollary 4.1.

1. Let $s(n) = |B_n| + n + q_n$ be the number of node labels (B_n the operator basis, q_n the constants). By leanness, $s(n) \leq n^C + n + n^C = n^{O(1)}$; for a single grammar fixed independently of n, take $C = 2$, so $s(n) \leq n^2$ for all large n. A derivation tree of cost at most k has at most k nodes, every node contributing at least one to the cost. The number of rooted ordered trees with m nodes is at most 4^m (a Catalan bound); assigning one of $s(n)$ labels to each node, the arity constraint only removes trees, so the number of derivation trees with m nodes is at most $(4\,s(n))^m$. Hence the number of derivation trees of cost at most k is at most $\sum_{m \leq k} (4\,s(n))^m \leq (8\,s(n))^{k+1}$.

2. Therefore the number of functions with $L_{G_n}(f) \leq p(n)$ is at most $(8\,s(n))^{p(n)+1} = 2^{O(p(n)\log n)}$. For all sufficiently large n, $2^{O(p(n)\log n)} + 2 < 2^{2^n} = |F_n|$, so there exists g with $L_{G_n}(g) > p(n)$ and $g \notin \{\chi_{[n]}, -\chi_{[n]}\}$, again by subtracting 2 and without any bound on $L_{G_n}(-\chi_{[n]})$. By Lemma A.1, $\bar{k}(g) < n$.

3. Then $L_{G_n}(g) > p(n) \geq L_{G_n}(\chi_{[n]})$ while $\bar{k}(g) < n = \bar{k}(\chi_{[n]})$: order preservation fails for every n past the threshold, which fixes n_2 in terms of the grammar family and p.

□

Concrete polynomial pricings of parity. The polynomial-cost hypothesis does not require a native XOR primitive: parity admits linear-size formulas when XOR is available and $O(n^2)$-size formulas over the De Morgan basis (AND, OR, NOT); with fixed operator costs, both constructions give polynomial description costs.

A.2 Complexity dispersion within frequency bands

Theorem (Full statement of Theorem 2)**.** There is an integer $m_0 \geq 16$, depending only on the fixed universal machine U, such that for every $n \geq 4$, every integer $v \in \{1, \ldots, n-1\}$, and every integer m satisfying

$$\max\{m_0,\ \lceil \log_2 n \rceil + \lceil \log_2(8\lceil \log_2 n \rceil)\rceil\} \leq m \leq \min\{2v,\ 2(n-v)\},$$

the band B_v contains at least $(1-\varepsilon_m)2^{2^m}$ distinct functions f with

$$2^m - \lceil m/2 \rceil \leq K(f) \leq \tfrac{3}{2}\,2^m,$$

where ε_m is the explicit error term defined in Equation (2) below; it depends only on m, satisfies $0 < \varepsilon_m < 1$ for $m \geq m_0$, and tends to zero as $m \to \infty$.

We use the same plain complexity K and universal machine U as in Section 4.1. A function depending only on $A \subseteq [n]$ is constant in every coordinate outside A. Its nonzero Fourier coefficients are indexed by subsets of A and agree with those of its restriction to $\{\pm 1\}^A$; thus its mean frequency is the same in either representation. The explicit error term in Theorem 2 is

$$\varepsilon_m = 2\exp\left(-\frac{2^m}{8m^2}\right) + 2^{\,2-\lceil m/2\rceil}. \tag{2}$$

The two terms bound, respectively, failure to enter the frequency band and the fraction of overly short descriptions in the construction family. Both tend to zero as $m \to \infty$. For $m \geq 16$, both are nonincreasing and their sum is at most $2e^{-32} + 1/64 < 1$.

Lemma A.2 (Shift by a disjoint character)**.** *Let $g \in F_n$ depend only on $A \subseteq [n]$, and let $T \subseteq [n]$ be disjoint from A. Then*

$$\widehat{g\chi_T}(R) = \hat{g}(R \triangle T), \qquad \bar{k}(g\chi_T) = \bar{k}(g) + |T|.$$

In particular, the Fourier support is $\{S \cup T : S \in \operatorname{supp}(\hat{g})\}$, and each occupied frequency order is shifted by $|T|$.

Proof. The identity $\chi_T \chi_R = \chi_{R \triangle T}$ gives the coefficient formula directly. Since $\hat{g}(S) = 0$ unless $S \subseteq A$ and $A \cap T = \emptyset$, a nonzero coefficient at S moves to $S \cup T$, whose cardinality is $|S| + |T|$. Parseval's identity therefore gives

$$\bar{k}(g\chi_T) = \sum_{S \subseteq A} (|S| + |T|)\hat{g}(S)^2 = \bar{k}(g) + |T|.$$

□

Lemma A.3 (Concentration for random tables)**.** *Let g be a uniformly random truth table on $m \geq 1$ coordinates, with 2^m independent uniform signs. Then*

$$\mathbb{E}[\bar{k}(g)] = m/2, \qquad \Pr\left[|\bar{k}(g) - m/2| \geq \tfrac{1}{2}\right] \leq 2\exp\left(-\frac{2^m}{8m^2}\right).$$

Proof. For each $S \subseteq [m]$, independence of the table entries gives $\mathbb{E}[\hat{g}(S)^2] = 2^{-m}$. Thus

$$\mathbb{E}[\bar{k}(g)] = 2^{-m} \sum_{S \subseteq [m]} |S| = 2^{-m} m 2^{m-1} = m/2.$$

For concentration, we first record an identity linking $\bar{k}$ to average sensitivity. Let $x^{(i)}$ be x with coordinate i flipped. Characters containing i change sign under the flip and all others are fixed, so $g(x) - g(x^{(i)}) = 2\sum_{S \ni i} \hat{g}(S)\,\chi_S(x)$, and orthogonality gives $\mathbb{E}_x[(g(x) - g(x^{(i)}))^2] = 4\sum_{S \ni i} \hat{g}(S)^2$. Since g is ± 1-valued, the left side equals $4\,\Pr_x[g(x) \neq g(x^{(i)})]$, hence $\Pr_x[g(x) \neq g(x^{(i)})] = \sum_{S \ni i} \hat{g}(S)^2$. Summing over i — each S is counted $|S|$ times, once per element — yields

$$\bar{k}(g) = \sum_S |S|\,\hat{g}(S)^2 = 2^{-m} \sum_{x \in \{\pm 1\}^m} \#\{i \in [m] : g(x) \neq g(x^{(i)})\}.$$

Changing a single table entry affects only the m incident edges, each counted at both endpoints. Hence it changes $\bar{k}(g)$ by at most $b = 2m/2^m$. Applying the bounded-differences inequality (McDiarmid, 1989) to the 2^m independent entries gives, for $a > 0$,

$$\Pr\left[|\bar{k}(g) - \mathbb{E}\bar{k}(g)| \geq a\right] \leq 2\exp\left(-\frac{2a^2}{2^m b^2}\right).$$

Substituting $a = 1/2$ and $2^m b^2 = 4m^2/2^m$ yields the claim. □

Lemma A.4 (Incompressibility within the construction family)**.** *Fix disjoint $A, T \subseteq [n]$ with $|A| = m$, let g be a uniformly random table on A, and set $f = g\chi_T$. For every integer $j \geq 1$,*

$$\Pr[K(f) \leq 2^m - j] \leq 2^{\,1-j}.$$

Proof. Multiplication by the fixed character is injective on the tables, so f is uniform on a family of 2^{2^m} distinct truth-table strings. For any nonnegative integer q, there are $2^{q+1}-1$ binary programs of length at most q, each producing at most one output. Thus at most $2^{q+1}-1$ members of the family have complexity at most q. Set $q = 2^m - j$ and divide by 2^{2^m} to obtain the bound. If $q < 0$, the event is empty. □

Lemma A.5 (Description upper bound for the construction family)**.** *There is a constant $c_3 \geq 0$, depending only on U and the fixed encoding conventions, such that for every $n \geq 4$, every pair of disjoint sets $A, T \subseteq [n]$ with $|A| = m$, $|T| = t$, and every table g on A,*

$$K(g\chi_T) \leq 2^m + 2n\lceil \log_2 n \rceil + c_3.$$

Proof. Let $r = \lceil \log_2 n \rceil$. A fixed decoder reads the following binary segments, in this order: n; the integers m, t; the lexicographic index of A among the m-subsets of $[n]$; the index of T among the t-subsets of $[n] \setminus A$; and the 2^m bits of g's table, ordered by the increasing coordinates of A.

The first segment encodes n by prefixing its ordinary binary representation with $\lfloor \log_2 n \rfloor$ zeros: the number of leading zeros fixes the length of the representation, so the decoder recovers n unambiguously. This uses $2\lfloor \log_2 n \rfloor + 1 \leq 2r + 1$ bits. Once n is read, each of $m, t \in \{0, \ldots, n\}$ uses $r + 1$ bits, so the header has at most $4r + 3$ bits; the two subset indices then have lengths

$$\left\lceil \log_2 \binom{n}{m} \right\rceil \leq mr, \qquad \left\lceil \log_2 \binom{n-m}{t} \right\rceil \leq tr,$$

and the length of every later segment is determined by the segments read before it, so the program parses without ambiguity. An index takes zero bits when there is only one possible subset, including $t = 0$. The decoder then enumerates all n-bit inputs in the prescribed order and outputs $g(x_A)\chi_T(x)$ for each. Since $m + t \leq n$ and $n \geq 4$, the data length is at most

$$2^m + (4 + m + t)r + 3 \leq 2^m + 2nr + 3.$$

The constant cost of simulating this decoder on U, together with the extra three bits, is absorbed into c_3. □

Proof of Theorem 2. Choose

$$m_0 = \max\{16,\ \lceil \log_2(c_3 + 2) \rceil + 2\},$$

where c_3 is from Lemma A.5. Fix n, v, m in the theorem's ranges and set $t = v - \lceil m/2 \rceil$.

Construction and range checks. The inequality $m \leq 2v$ gives $t \geq 0$, while $m \leq 2(n - v)$ gives

$$m + t = v + \lfloor m/2 \rfloor \leq n.$$

We may therefore fix disjoint sets $A, T \subseteq [n]$ of sizes m, t. As g ranges over all tables on A, the functions $f = g\chi_T$ form a family of exactly 2^{2^m} members. All probabilities below are with respect to uniform g in this construction, not uniform sampling from B_v.

Frequency-band event. By Lemma A.2, $\bar{k}(f) = \bar{k}(g) + t$. On the event $|\bar{k}(g) - m/2| < 1/2$,

$$|\bar{k}(f) - v| \leq |\bar{k}(g) - m/2| + |m/2 - \lceil m/2 \rceil| < 1,$$

so $f \in B_v$. By Lemma A.3, failure of this sufficient event has probability at most $2\exp(-2^m/(8m^2))$.

Complexity bounds. The lower range condition on m implies $2^m \geq 8n\lceil \log_2 n \rceil$, hence $2n\lceil \log_2 n \rceil \leq 2^{m-2}$. Our choice of m_0 also ensures $c_3 \leq 2^{m-2}$. Lemma A.5 therefore gives, for every family member,

$$K(f) \leq 2^m + 2n\lceil \log_2 n \rceil + c_3 \leq \tfrac{3}{2}\, 2^m.$$

Lemma A.4, with $j = \lceil m/2 \rceil$, gives

$$\Pr[K(f) \leq 2^m - \lceil m/2 \rceil] \leq 2^{\,1-\lceil m/2 \rceil} \leq 2^{\,2-\lceil m/2 \rceil}.$$

Combining the events. A union bound shows that a fraction at least $1 - \varepsilon_m$ of the family lies in B_v and satisfies $2^m - \lceil m/2 \rceil < K(f) \leq \frac{3}{2}\, 2^m$. Multiplying by the family size and weakening the strict lower inequality proves the theorem. No independence between the two failure events is required. □

Attainable scales and characters. For each admissible m, the theorem gives at least $(1-\varepsilon_m)2^{2^m}$ functions in B_v whose complexity divided by 2^m lies between $1 - \lceil m/2 \rceil / 2^m$ and $3/2$. Both ε_m and the relative lower slack tend to zero with m, and the smallest admissible m tends to infinity with n. If the admissible range is nonempty, taking $m = \min\{2v, 2(n-v)\}$ gives the largest scale guaranteed by this construction.

For any $S \subseteq [n]$ with $|S| = v$, the character χ_S has $\bar{k}(\chi_S) = v$ and hence lies in B_v. A program encodes n, v in $O(\log n)$ bits, followed by the $\lceil \log_2 \binom{n}{v} \rceil$-bit index of S, and prints χ_S on the entire domain. Thus

$$K(\chi_S) \leq \log_2 \binom{n}{v} + O(\log n).$$

At $v = \lfloor n/2 \rfloor$, the largest admissible m is n when n is even and $n-1$ when n is odd, provided the lower threshold is met; bands near either endpoint may have no admissible m.

B Details of the Baseline Grammar and Exact Enumeration

This appendix specifies the baseline grammar G0 behind every C_{sym} value in the paper and the exact algorithm that computes it.

The grammar G0. An expression is generated by

$$r ::= 0 \mid 1 \mid x_1 \mid x_2 \mid x_3 \mid x_4 \mid \mathrm{NOT}(r) \mid \mathrm{AND}(r, r) \mid \mathrm{OR}(r, r) \mid \mathrm{XOR}(r, r),$$

with atoms at cost 1, unary cost $c(\mathrm{NOT}(r)) = 1 + c(r)$, and binary cost $c(\mathrm{op}(r_1, r_2)) = 1 + c(r_1) + c(r_2)$ for each of AND, OR, and XOR. The three binary operators carry the same overhead, so parity-4 and the four-way conjunction are both priced at 7; this same-cost pair recurs throughout the experiments. The constants are atoms, so $C_{\mathrm{sym}}(0) = C_{\mathrm{sym}}(1) = 1$.

Exact enumeration by layered closure. Every C_{sym} reported in this paper is exact. The assignment runs over functions rather than expressions, as Dijkstra's algorithm on the graph whose nodes are the $65{,}536$ truth tables. Layer 1 holds the six atoms. Layer k collects every function of minimum cost exactly k, reached either as $\mathrm{NOT}(f)$ with $C_{\mathrm{sym}}(f) = k - 1$, or as a binary composition from completed lower layers with $C_{\mathrm{sym}}(f) + C_{\mathrm{sym}}(g) + 1 = k$. The layer-by-layer assignment is exact because every optimal expression has direct subexpressions of minimum cost and strictly smaller cost: replacing any direct subexpression by a cheaper equivalent would strictly decrease the total, so no function first appears at a non-minimal layer. The closure runs to exhaustion with maximum cost 17; the layer sizes for costs 1 through 17 are

$$6,\ 4,\ 18,\ 42,\ 84,\ 324,\ 415,\ 1{,}483,\ 2{,}124,$$
$$5{,}764,\ 7{,}217,\ 13{,}267,\ 12{,}175,\ 15{,}195,\ 5{,}464,\ 1{,}910,\ 44,$$

which sum to the full space. As an independent check of a different algorithmic shape, a fixpoint-relaxation enumerator reproduces every cost up to 5 key by key.

Registered checks. The preregistered reference values are reproduced by the closure: $C_{\mathrm{sym}}(x_i) = 1$, $C_{\mathrm{sym}}(\neg x_i) = 2$, $C_{\mathrm{sym}}(x_1 \wedge x_2) = 3$, and $C_{\mathrm{sym}}(\mathrm{parity}_4) = C_{\mathrm{sym}}(\mathrm{AND}_4) = 7$. The complementation bound $|C_{\mathrm{sym}}(\neg f) - C_{\mathrm{sym}}(f)| \leq 1$ holds over the entire table.

C Details of the cube8 Spectral-Bias Replication

This experiment checks whether the low-frequency-first learning pattern discussed in Section 3.3 is reproduced for a real-valued target on Boolean inputs. The theoretical motivation comes from analyses of Boolean-domain kernels (Yang & Salman, 2019) and direction-dependent convergence rates in the NTK regime (Cao et al., 2021); the experiment measures the learning order directly.

Setup. The input domain is the full cube $\{\pm 1\}^8$, and all $2^8 = 256$ inputs form the training set. There is no held-out test set or data split. With target seed 0, we select one Fourier character χ_{S_k} at

each frequency $k = 1, \ldots, 8$, and fix the real-valued target

$$g^*(z) = \sum_{k=1}^{8} \chi_{S_k}(z), \qquad |S_k| = k.$$

Each component has amplitude 1, so $g^*(z) \in \{0, \pm 2, \pm 4, \pm 6, \pm 8\}$. The character specification is

$$\begin{aligned} S_1 &= \{7\}, & S_2 &= \{3, 7\}, \\ S_3 &= \{3, 5, 7\}, & S_4 &= \{2, 3, 4, 7\}, \\ S_5 &= \{2, 3, 5, 6, 7\}, & S_6 &= \{1, 2, 3, 4, 5, 7\}, \\ S_7 &= \{1, \ldots, 7\}, & S_8 &= [8]. \end{aligned}$$

Indexing is one-based.

The real-valued construction permits independently specified component amplitudes. Parseval's identity still applies: orthonormality gives $\mathbb{E}[(g^*)^2] = 8$, whereas a ± 1-valued target has total Fourier energy 1 and cannot have eight unit-amplitude coefficients. Using one unit-amplitude Fourier character per frequency provides a discrete counterpart to a target formed from one unit-amplitude sinusoid per frequency.

We use full-batch Adam with constant learning rate 10^{-3} and MSE loss, six hidden layers of 256 ReLU units with a linear output head (331,521 parameters), a 1,000-epoch budget, and training seeds 100–104; the target remains fixed across these seeds. Recovery of each target character is computed by an exact full-domain Fourier transform of the network output, avoiding the spectral leakage associated with a sampled DFT in continuous-domain protocols (Rahaman et al., 2019).

Frequency-resolved recovery. A character counts as recovered at the first epoch whose rolling-median recovery (window of five epochs) on the seed-median trajectory reaches 50%. These epochs are $10, 14, 23, 33, 39, 45, 49, 59$ for frequencies 1 through 8, respectively. Thus, the recovery threshold is reached strictly later at higher frequencies. Every character reaches full recovery within 1,000 epochs. As a capacity check, a pure frequency-8 character trained alone converges to machine precision. This indicates that the network can represent the high-frequency component; its delayed recovery in the mixed target is consistent with a difference in learning dynamics rather than a lack of representational capacity.

Overshoot and the single-character ablation. On the seed-median trajectory, the low-frequency characters transiently exceed their target amplitude before settling at one: peak recovery is 1.38 at frequency 1 (epoch 11), 1.26 at frequency 2, and 1.11 at frequency 3, decreasing with frequency, while frequency 8 rises monotonically to its target (peak 1.00). The heatmap fixes the color scale to $[0, 1]$ following the original figure; the recorded coefficients are never clipped. In a single-character ablation, each of the eight characters serves alone as the target under the same protocol and seeds: the same transient appears at every frequency (peak 1.35 at frequency 1, 1.41 at frequency 3), all final values return exactly to one, and the largest leakage into another character's slot is 0.07 on the seed-median trajectory. The overshoot is an underdamped transient of learning a single character itself, decreasing in magnitude with frequency, rather than an interaction between the characters of the mixed target.

D Paired Training Controls

Target selection. The controls in Section 5.2 use the $\binom{16}{8} = 12{,}870$ balanced functions obtained by filtering the full four-variable Boolean function space. We use the exact C_{sym} values under the baseline grammar G0 from Section 5.1. At fixed C_{sym}, we sort functions within each of the 12 groups $C_{\mathrm{sym}} \in \{5, \ldots, 16\}$ by T_3 and select the four lowest and four highest, yielding 96 targets.

For the fixed-profile control, we group functions by (E_1, E_2, E_3, E_4), rounding each entry to ten decimal places. Since $E_0 = 0$ for balanced targets, these groups match the full frequency-energy profile, namely the total squared Fourier coefficients at each frequency. Of the 37 profiles in the balanced pool, 18 have a maximum-minus-minimum C_{sym} gap of at least 3 and at least three functions at each extreme; we use all 18. Within each profile, we sample three minimum-C_{sym} and

three maximum-C_{sym} functions without replacement, pairing them in sampling order to obtain 54 pairs and 108 targets. We order profiles by decreasing C_{sym} gap and then pool size, assign indices $j = 0, \ldots, 17$, and use sampling seeds $b + 1000 + j$ and $b + 2000 + j$ for the minimum and maximum groups, respectively, where $b = 20260826$. Target selection is performed once, and both activations and all training seeds share the resulting targets. The two controls construct their target sets separately; three functions occur in both sets and are trained separately in each control.

Input sampling. Each target is embedded in $\{0,1\}^{12}$ with label $y = f(x_1, x_2, x_3, x_4)$. Inputs and labels use $0/1$ encoding. For each of the 16 patterns of the first four coordinates, we sample 128 of the 256 possible trailing-eight-coordinate combinations without replacement, giving 2,048 training inputs. For training seed $s \in \{0,1,2\}$, the input-sampling seed is $b+10000+s$. All targets and both activations share the same input split for a given s, and the sampled inputs remain fixed throughout training. Evaluation uses the uniform full domain of 4,096 inputs, including the training inputs.

Network and training protocol. We train MLPs with widths $12 \to 128 \to 128 \to 1$, hidden activations in $\{\mathrm{ReLU}, \tanh\}$, and seeds $s \in \{0,1,2\}$. Table 3 gives the configuration. Each target is trained once per seed and activation. The training seed controls initialization and mini-batch shuffling; within each activation, resetting it for each target shares the initialization and batch order across targets. We use D from Section 3 with $\tau = 0.99$ and full-domain accuracy. Training stops when full-domain classification accuracy reaches 100% or the epoch budget is exhausted.

Table 3: Training configuration for both controls and activations in Section 5.2.

Setting	Value
Parameters	18,305
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	10^{-4}
Loss	Binary cross-entropy with logits
Batch size	512, reshuffled each epoch
Epoch budget	240
ReLU weight initialization	Library-default Kaiming-uniform
tanh weight initialization	Xavier-uniform, gain $5/3$
Bias initialization	Library default for both activations

Aggregation at fixed C_{sym}. For each function, we first take the median D over the three training seeds. Within each C_{sym} group, we average these medians separately over the four high-T_3 and four low-T_3 targets, then subtract the low-group mean from the high-group mean. The reported ΔD is the equally weighted mean of the 12 group contrasts, computed separately for each activation.

Aggregation at fixed profile. For each function pair and training seed, we subtract the low-C_{sym} target's D from the high-C_{sym} target's D. We average these differences over the three seeds, then over the three pairs within each profile. The reported ΔD is the equally weighted mean of the 18 profile contrasts, computed separately for each activation.

Confidence intervals and run coverage. We construct 95% percentile bootstrap intervals from 10,000 replicates. Each replicate samples with replacement either 12 C_{sym}-group contrasts or 18 profile contrasts and computes their mean. The interval endpoints are the 2.5th and 97.5th percentiles of these means. Bootstrap seeds are $b + 55555$ for the fixed-C_{sym} control and $b + 60606$ for the fixed-profile control. Each activation has $96 \times 3 = 288$ runs at fixed C_{sym} and $108 \times 3 = 324$ runs at fixed profile, for 1,224 runs overall. Every run reaches the accuracy threshold within budget. At fixed C_{sym}, the group contrast is positive in 10 of 12 groups under ReLU and all 12 under tanh. Table 2 reports the aggregate contrasts and intervals; Figure 8 shows the four estimates as a forest plot; Appendix F reports the input-dimension sweep.

Regression view on a per-function difficulty table. The paired controls select targets by exact matching; a regression view on a separately trained per-function table gives the same reading. The

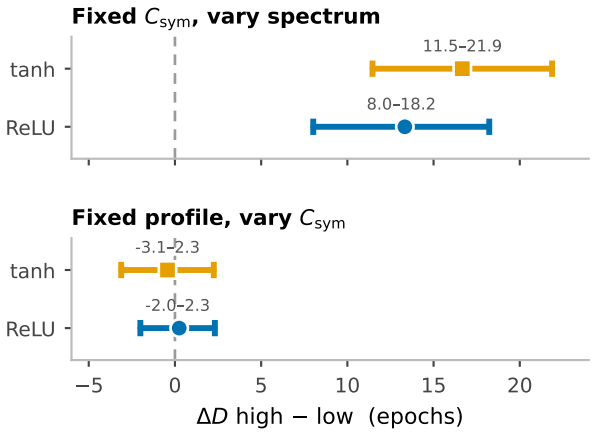


Figure 8: The four paired contrasts with 95% bootstrap confidence intervals (forest view of Table 2). Top panel: fixed C_{sym}, varying spectrum; bottom panel: fixed profile, varying C_{sym}. Marker shape and color distinguish the activations.

table covers 92 balanced functions stratified across C_{sym} layers with three seeds each (D = epochs to 99% full-domain accuracy). Its marginal Spearman correlations with difficulty are 0.74 for C_{sym} and 0.80 for T_3; conditioning on C_{sym} leaves the tail correlation at $+0.56$, while conditioning on T_3 drops the symbolic correlation to $+0.38$; a standardized regression of difficulty on both gives $\beta_{C_{\text{sym}}} = 0.49$ and $\beta_{T_3} = 0.39$ ($R^2 = 0.64$). The spectral coordinate carries the conditional signal in both views.

E Alternative Metrics and Cut-Point Sensitivity

Averaged over all 65,536 functions, frequency energies follow the binomial coefficients $\binom{4}{k}/16$: frequencies 0 through 4 hold 6.25%, 25%, 37.5%, 25%, 6.25%. Frequency 2 is the middle of this sequence and the mode across functions, with 52% of functions keeping their largest single-frequency energy there, so T_3 cuts the upper tail above the mode.

Measure-caliber correlations. Table 4 reports the Spearman correlations among the symbolic cost and five scalar calibers over all 65,536 functions: the mean frequency $\bar{k}$ (equal to the average sensitivity), the LZ76 phrase count C_{lz} on the 16-bit truth table, the support degree deg (the highest occupied frequency), and the low-frequency masses L_1 and L_2. The symbolic cost correlates only moderately with every spectral caliber ($|\rho| \leq 0.44$) and almost not at all with the LZ count; within the spectral side, $\bar{k}$ and the low-frequency masses are strongly coupled (-0.80), since both are projections of one profile.

Table 4: Spearman correlations among scalar calibers over the full four-bit space ($n = 65{,}536$).

	C_{sym}	$\bar{k}$	C_{lz}	deg	L_1	L_2
C_{sym}	1					
$\bar{k}$	0.44	1				
C_{lz}	0.10	0.09	1			
deg	0.13	0.19	0.02	1		
L_1	−0.32	−0.80	−0.05	−0.08	1	
L_2	−0.38	−0.80	−0.03	−0.18	0.42	1

Resolution of the LZ caliber. At $n = 4$ the LZ76 phrase count takes only four values over the whole space — $5, 6, 7, 8$, held by 3,288, 32,708, 28,888, and 652 functions — so this caliber ranks functions at four levels of granularity and cannot resolve cost variation at a fixed profile.

Cut-point sensitivity. Table 5 recomputes the axis correlation of Section 5.1 under alternative cut-points and all seven grammars. A complementary caliber reverses the sign exactly under Parseval (for example, $\rho(C_{\mathrm{sym}}, 1 - L_1) = +0.32$ in G0), so complementary rows are omitted. Across grammars and calibers, the low-mass correlations stay within $[-0.87, -0.32]$ and the mean-frequency correlations within $[0.44, 0.86]$: no cut-point or grammar brings the two axes into coincidence. The strongest alignment arises exactly in the grammars that price parity high — G1 without XOR and G2 over NAND — which is the reading behind the main-text range.

Table 5: Axis correlations under alternative cut-points, per grammar (Spearman ρ between C_{sym} and the named caliber, full space).

Grammar	$\rho(C_{\mathrm{sym}}, L_1)$	$\rho(C_{\mathrm{sym}}, L_2)$	$\rho(C_{\mathrm{sym}}, \bar{k})$
G0	-0.32	-0.38	0.44
G1	-0.55	-0.87	0.86
G2	-0.54	-0.83	0.83
G3	-0.39	-0.55	0.58
G4	-0.43	-0.54	0.60
G5	-0.44	-0.54	0.61
G6	-0.40	-0.47	0.54

F Details of the Input-Dimension Sweep

Protocol. The sweep repeats the paired comparisons of Appendix D with the four relevant bits placed at random positions within $d \in \{12, 16, 24, 32\}$ inputs, varying the number of nuisance coordinates; nuisance values are drawn without repetition, and training and evaluation inputs never overlap. Two details differ from the base protocol: the epoch budget is 200, with runs that never reach the threshold recorded at 205, and the 99% threshold for D is measured on a held-out reference set of 2,048 inputs, since the full cube cannot be enumerated at $d = 32$.

Result. The fixed-cost spectral contrast is positive at every dimension, while all four confidence intervals for the fixed-profile complexity contrast include zero. At $d = 32$, 14.6% and 7.1% of runs in the two arms never reach the threshold.

G Grammar Sweep: Per-Grammar Tables

Section 5.1 presents the cross-grammar correlations and within-profile cost-spread comparison. This appendix provides the grammar specifications and the per-grammar tables behind them.

Grammar designs. All seven grammars keep unit-cost atoms (the constants and the four variables) and additive node costs; they differ in the operator basis or in a single price. Table 6 lists the designs. The added operators are the ternary conditional $\mathrm{ITE}(c, t, e)$, the four-input majority MAJ_4 (at least three of four), and the four-input threshold ATL_2 (at least two of four).

Table 6: The seven grammars. Atoms are $0, 1, x_1, \ldots, x_4$ at cost 1 in every grammar; a d-ary operator costs its overhead plus the costs of its inputs.

Grammar	Operator basis	Prices	Design intent
G0	AND, OR, XOR, NOT	all overheads 1	propositional logic (baseline)
G1	AND, OR, NOT	—	the baseline without XOR
G2	NAND	overhead 1	two-level circuits
G3	G0 + ITE	ternary, overhead 1	programming conditionals
G4	G0 + MAJ_4	4-ary, overhead 1	voting thresholds
G5	G0 + ATL_2	4-ary, overhead 1	at-least-k counting
G6	AND, OR, XOR, NOT	XOR overhead 2	repricing on a fixed basis

Enumeration coverage. The layered closure of Appendix B extends to every grammar, with three additions: operator-specific overheads (G6), grammars without NOT (G2, where $\mathrm{NAND}(f, 1)$ supplies negation), and operators of arity three and four (G3–G5). All seven grammars assign costs to the full space of 65,536 functions, with maximum costs $17, 35, 51, 16, 16, 16, 20$ for G0 through G6. Removing an operator or raising a price never lowers a cost, and adding an operator never raises one.

Table 7: Per-grammar results on the full four-bit space. Correlations are Spearman coefficients between the grammar's C_{sym} and the scalar projections of the profile. Spans are within-profile cost-percentile spans (Section 5.1). $|A'|$ and $|B'|$ count functions with cheap cost and $T_3 \geq 0.5$, and expensive cost and $L_1 \geq 0.5$, under percentile-matched gates.

Grammar	$\rho(C_{\mathrm{sym}}, L_1)$	$\rho(C_{\mathrm{sym}}, T_3)$	median	weighted	max	span $\geq$0.5	$\lvert A'\rvert$	$\lvert B'\rvert$
G0	-0.32	0.38	0.35	0.72	0.97	0.33	420	148
G1	-0.55	0.87	0.12	0.22	0.67	0.05	0	0
G2	-0.54	0.83	0.36	0.49	0.79	0.24	0	14
G3	-0.39	0.55	0.38	0.46	0.97	0.22	197	28
G4	-0.43	0.54	0.43	0.60	0.97	0.30	186	6
G5	-0.44	0.54	0.43	0.60	0.97	0.30	186	6
G6	-0.40	0.47	0.31	0.58	0.96	0.28	205	24

Quadrant gates. The quadrant counts transplant the thresholds of Figure 3 to each grammar's own cost scale. The cheap gate is the mid-rank percentile of G0's cost-9 layer (0.052) and the expensive gate that of G0's cost-15 layer (0.929); the spectral gates $T_3 \geq 0.5$ (region A') and $L_1 \geq 0.5$ (region B') are unchanged.

Named rules across grammars. Table 8 tracks six named rules. The bottom of the scale is stable — the two-variable conjunction stays at cost 3 (cost 5 under G2), within the cheapest 0.1% of every grammar — while the position of a rule in the middle of the scale is a property of the basis: parity-4 moves from cost 7 in G0 to 34 in G1 (the 99.9th percentile) once XOR is removed, and MAJ_4 drops from cost 15 in G0 (93rd percentile) to 5 in G4 (below the 0.3rd percentile) once voting is native.

Table 8: Exact minimum costs of six named rules under the seven grammars.

Rule	G0	G1	G2	G3	G4	G5	G6
$x_1 \oplus x_2$	3	8	11	3	3	3	4
$x_1 \wedge x_2$	3	3	5	3	3	3	3
AND_4	7	7	13	7	7	7	7
parity_4	7	34	47	7	7	7	10
MAJ_4 ($\geq$3 of 4)	15	15	17	12	5	10	15
ATL_2 ($\geq$2 of 4)	15	15	25	11	10	5	15

Within-profile spans. The span statistic of Section 5.1 is computed on the 138 profiles that hold at least 30 functions each, covering 99.9% of the space, where a profile is an exact equality class of the degree energies $(E_0, \ldots, E_4)$. Within a profile, each function's C_{sym} is converted to its mid-rank percentile over the full space under the grammar at hand; the span is the maximum minus the minimum percentile in the profile. The median is over profiles with equal weight, and the weighted median over profiles weighted by their function counts. The full distributions are the violins of Figure 3; Table 7 quantifies them: the fraction of profiles with span at least 0.5 stays between 0.22 and 0.33 under every grammar except G1 (0.05), and the maximum span exceeds 0.6 under every grammar including G1.

H Details of Learning under Competing Rules

H.1 Random-table sampling and compression diagnostics

For each $n \in \{8, 10, 12, 14\}$, three truth tables are drawn from the operating system's entropy source and frozen once drawn rather than regenerated from a seed. Bits are packed least-significant-first within each byte; a set bit represents output $+1$ and an unset bit -1; inputs are in lexicographic order. The $n = 12$ and $n = 14$ tables were drawn fresh for this experiment. The $n = 8$ and $n = 10$ tables reuse the three highest-overlap tables of an earlier frozen batch, whose overlaps with parity lie in 51.6–55.1% at $n = 8$ and 52.0–53.3% at $n = 10$. No draw was rejected or replaced; the automated rejection rule (redraw if a table equals $\pm\chi_{[n]}$) never fired.

The lower-bound values of the uniform-draw bound in Section 5.3 are approximately 99.61%, 99.90%, 99.98%, and 99.99% at $n = 8, 10, 12, 14$, respectively.

Each random table is checked against both $\chi_{[n]}$ and $-\chi_{[n]}$. We compress the packed bytes with gzip and LZMA at their maximal settings (-9); the diagnostic passes only if both outputs are at least as long as the raw bytes. Table 9 reports the measured lengths: every parity control at $n \geq 10$ shrinks under both compressors, while the $n = 8$ controls are too short for the formats' container overhead, which inflates all 32-byte outputs.

Table 9: Compression diagnostics in bytes. The three random tables of a dimension share one length (incompressible data plus fixed container overhead), except under gzip at $n = 8$ (55, 53, 55).

Table	raw	gzip -9	LZMA -9
parity_8	32	36	76
random_8 (3 draws)	32	53–55	88
parity_{10}	128	45	88
random_{10} (3 draws)	128	151	188
parity_{12}	512	56	100
random_{12} (3 draws)	512	535	572
parity_{14}	2,048	73	124
random_{14} (3 draws)	2,048	2,071	2,108

H.2 Dataset construction

Let q be the number of inputs where parity and the random table agree, and let $N_{\text{dis}} = 2^n - q$. All q agreement inputs enter training with their shared label. For accuracy target $t \in \{0.90, 0.95, 1.00\}$, choose the largest even integer $m \leq N_{\text{dis}}$ satisfying

$$\frac{q + m/2}{q + m} \geq t.$$

At $t = 1$, set $m = 0$. Permute the disagreement inputs with seed $7000 + 100n + \text{draw}$, independent of t. Add the first m inputs: label the first $m/2$ according to parity and the remaining $m/2$ according to the random table. Each candidate therefore agrees with exactly $q + m/2$ training labels. The realized accuracy level can exceed t because m is discrete. The same permutation is reused across accuracy levels, and each resulting dataset is shared across network initialization seeds. The mixing seed determines only the disagreement ordering, not the random truth table.

H.3 MLP and training protocol

The network maps n inputs through six hidden layers of 256 ReLU units each to a scalar linear output. Linear layers use the PyTorch default initialization: Kaiming-uniform weights with parameter $a = \sqrt{5}$ and uniform biases. The run's initialization is fixed by its training seed; training seeds are 100–104. Data are fixed and batches are full, so initialization is the intended source of between-seed variation.

All 2^n inputs are represented as ± 1 vectors in lexicographic order, with ± 1 targets. Loss is mean squared error over the training inputs only. We use Adam with constant learning rate 10^{-3}; the

remaining arguments hold the library defaults. There is one full-batch update per epoch, no learning-rate decay, and no early stopping; every run completes the 5,000-epoch budget.

H.4 Measurements and event definitions

The primary readout is the full-domain cosine overlap: for the network's real-valued output g_e after epoch e and either sign-valued candidate f,

$$\cos_f(e) = \frac{\mathbb{E}_x[g_e(x)f(x)]}{\sqrt{\mathbb{E}_x[g_e(x)^2]}}, \qquad x \sim \mathrm{Unif}(\{\pm 1\}^n).$$

Unlike sign accuracy, this measure retains output amplitudes. Within each epoch, training MSE is recorded before the update, the optimizer step runs, and every diagnostic thereafter — training sign accuracy, both overlaps, and the energy bins — is computed on the post-update output. A prediction is positive exactly when the output is strictly greater than zero, so an output equal to zero is classified as negative.

Other recorded quantities are the raw parity overlap $\mathbb{E}[g\chi_{[n]}]$ (the highest-frequency Fourier coefficient), sign agreement with each candidate on held-out inputs and on the full domain, full-domain mean absolute output and mean squared output, and Fourier energy grouped by frequency $|S| = 0, \ldots, n$. Between a fixed output g and uniformly random signs, the cosine overlap has mean zero and standard deviation $2^{-n/2}$; the shaded bands in the figures mark $\pm 2\sigma$ around zero, the chance baseline. For real-valued network outputs, each energy bin is divided by $\mathbb{E}[g^2]$, so the normalized bins sum to one whenever this denominator is nonzero.

Interpolation is the first epoch of perfect training sign accuracy on the training points. Takeover is durable by definition: the parity–random cosine difference must turn positive and stay positive to the final epoch, with no-takeover runs marked at the budget; a transient crossing alone does not count.

H.5 Full grid and supplementary results

The complete MLP grid contains four dimensions, three random tables per dimension, five initializations per table, and three accuracy levels, for $4 \times 3 \times 5 \times 3 = 180$ runs. All runs reach the common 5,000-epoch budget, and no run has parity leading from the start: the random overlap rises first in every one of the 180 runs. Table 10 reports the per-cell event statistics and terminal medians, and Figure 9 shows the trajectories of all twelve cells. Takeover at $n = 10$ is a per-run minority at every accuracy level (9, 7, and 4 of 15 runs at the 90%, 95%, and 100% levels); at $n = 8$ it never occurs (0/15 at every level).

Because the three tables and the dataset of a cell are shared across the five initialization seeds, the 15 runs of a cell are not fully independent; the counts and medians describe the runs of this grid rather than 15 independent replications.

One run departs from the grid's pattern: at $n = 14$, level 95%, draw 0, seed 102, training collapses at epoch 4,799 from an interpolated state — training accuracy falls to chance and stays there, with over 98% of the terminal energy in the constant bin and near-zero output amplitude. This is the single run behind the 14/15 takeover count of that cell; the other 179 runs interpolate and hold.

H.6 MLP activation ablation

The tanh arm of Section 5.3 reruns the $n = 12$ tiers of the main grid with every hidden ReLU replaced by a tanh. The frozen datasets, training seeds (100–104), learning rate, and 5,000-epoch budget are unchanged; the linear weights use Xavier-uniform initialization with the tanh gain of $5/3$ and library-default biases, the tanh protocol of Section 5.2. All 45 runs interpolate (median epochs 133, 119, and 110 at the 90%, 95%, and 100% levels), the random overlap rises first in every run, and no run takes over within the budget. Table 11 and Figure 10 give the per-tier statistics and trajectories: the terminal medians stay on the random side at every level ($+0.68$ to $+0.74$ for the random candidate against $+0.38$ to $+0.40$ for parity), and the top-frequency energy share ends at 0.14–0.16, against 0.47 for the ReLU network on the same datasets.

Table 10: Per-cell event statistics over the full MLP grid. Takeover counts are out of 15 runs; median epochs are over all 15 runs (interpolation) or over the takeover runs (takeover); terminal overlaps are medians over the 15 runs.

n	level	takeover	med. interp.	med. takeover	cos parity	cos random
8	90%	0/15	34	—	0.37	0.72
8	95%	0/15	34	—	0.34	0.77
8	100%	0/15	32	—	0.32	0.79
10	90%	9/15	52	2,902	0.57	0.55
10	95%	7/15	54	2,649	0.55	0.59
10	100%	4/15	49	2,411	0.53	0.63
12	90%	15/15	111	1,428	0.68	0.39
12	95%	15/15	102	1,748	0.77	0.31
12	100%	14/15	90	995	0.87	0.21
14	90%	15/15	550	4,022	0.80	0.20
14	95%	14/15	217	4,154	0.88	0.13
14	100%	14/15	203	3,718	0.96	0.06

Table 11: Event statistics for the tanh arm at $n = 12$: 15 runs per level, medians over the 15 runs. No run takes over at any level.

level	takeover	med. interp.	cos parity	cos random	top-freq. share
90%	0/15	133	0.40	0.68	0.16
95%	0/15	119	0.39	0.72	0.15
100%	0/15	110	0.38	0.74	0.14

H.7 Competing rules beyond parity

The four additional rules of Section 5.3, with parity as the anchor, are

$$\begin{aligned}
\text{par9_maj3}: &\quad \chi_{\{1,\dots,9\}} \cdot \mathrm{MAJ}(x_{10}, x_{11}, x_{12}), && \bar{k} = 10.5,\\
\text{par8_maj3}: &\quad -\chi_{\{1,\dots,8\}} \cdot \mathrm{MAJ}(x_9, x_{10}, x_{11} \oplus x_{12}), && \bar{k} = 10,\\
\text{mod3_chi4}: &\quad \mathbf{1}\{\mathrm{popcount}(x_1, \dots, x_8) \equiv 0 \pmod 3\} \cdot \chi_{\{9,\dots,12\}}, && \bar{k} = 9.31,\\
\text{subpar9}: &\quad \chi_S,\ |S| = 9, && \bar{k} = 9,
\end{aligned}$$

where MAJ is the majority of three bits. The frozen truth tables are authoritative; a global sign flip, as in par8_maj3, affects neither $\bar{k}$ nor the energy profile. The names are descriptive labels rather than primitives of the description grammar: for example, the mod-3 indicator is realized in the G0 grammar by a carry-save adder tree over the first eight coordinates, a disjunctive form on the four count bits, and a conjunction with the four-coordinate character — a formula of at most 839 nodes at $n = 12$; an $n = 4$ calibration gives a true minimum of 16 for mod 3, against 7 for parity-4.

Each rule is paired with the same three frozen random tables as the main experiment, at both accuracy levels, with five initializations per table: $4 \times 2 \times 3 \times 5 = 120$ runs, reusing the anchor's existing parity batches unchanged. Table 12 reports median takeover epochs and counts; Figure 11 shows all trajectories.

Table 12: Rule arms at $n = 12$: median takeover epoch among takeover runs (takeover count out of 15).

Rule	$\bar{k}$	95% level	100% level
parity12 (anchor)	12	1,748 (15/15)	995 (14/15)
par9_maj3	10.5	2,290 (13/15)	2,208 (15/15)
par8_maj3	10	3,442 (12/15)	2,710 (12/15)
mod3_chi4	9.31	2,039 (13/15)	2,020 (14/15)
subpar9	9	1,417 (15/15)	940 (15/15)

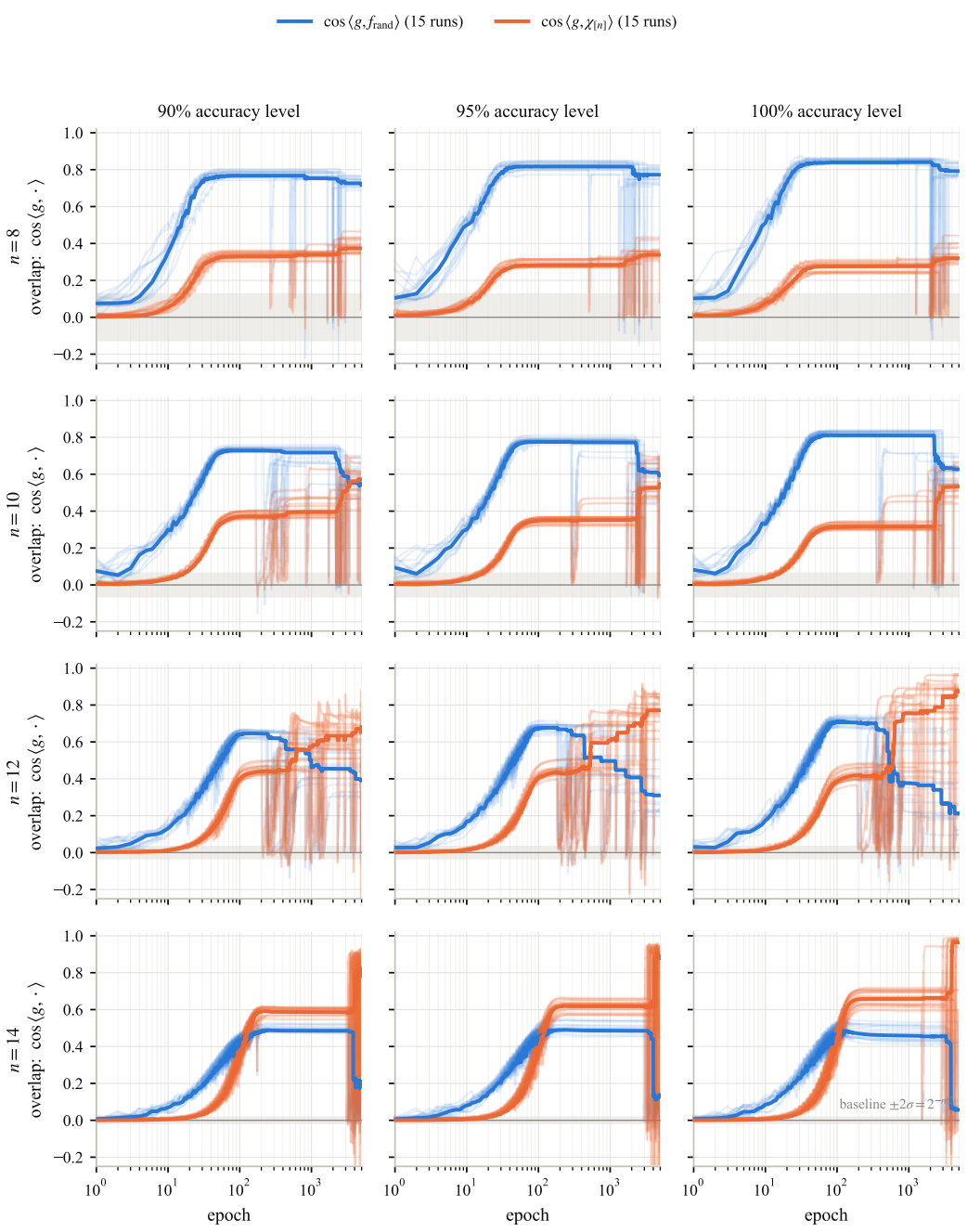


Figure 9: Competing-rule trajectories over the full grid: rows are dimensions $n \in \{8, 10, 12, 14\}$, columns are accuracy levels. Line conventions as in Figure 4.

H.8 Decoder-only Transformer extension

Architecture and parameter accounting. The model is a two-block pre-LN decoder-only Transformer: each block applies LayerNorm, four-head causal self-attention (head width 30) with a residual connection, then LayerNorm and a $120 \to 480 \to 120$ GELU feed-forward network with a second residual connection; rotary position embeddings (base 10,000) carry position information with zero parameters, and a final LayerNorm and linear head produce 29 next-token logits. There is no dropout; weights are initialized from $\mathcal{N}(0, 0.02^2)$ with zero biases, and the head is not tied to

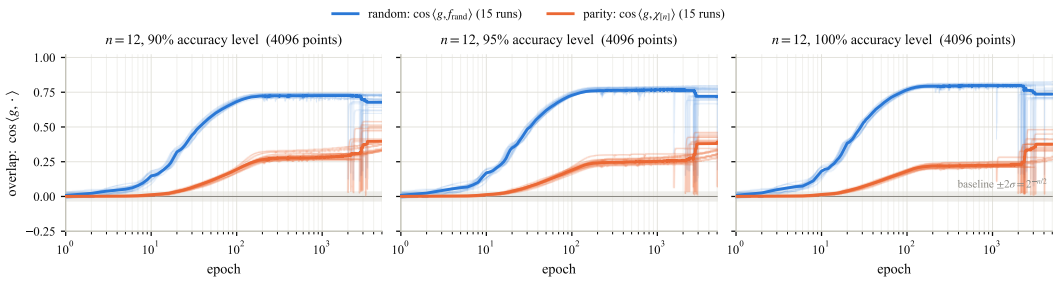


Figure 10: Competing-rule trajectories under the tanh arm at $n = 12$: columns are accuracy levels. Line conventions as in Figure 4.

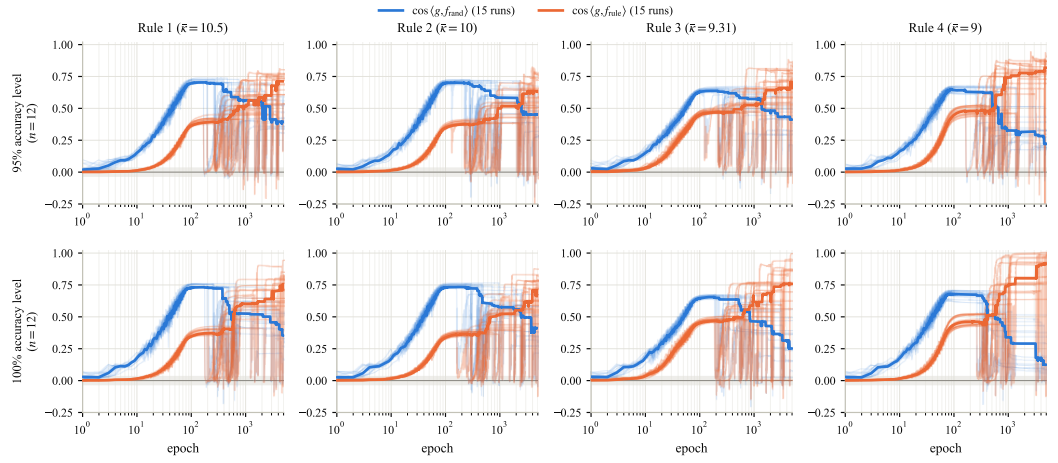


Figure 11: Rule arms at both accuracy levels ($n = 12$). Columns: the four rules as in Table 12; rows: accuracy levels. Line conventions as in Figure 4.

the embedding. The parameter count is 355,949: token embedding $29 \times 120 = 3{,}480$, each block 174,360 (attention 43,560 for query–key–value plus 14,520 for the output projection, feed-forward $58{,}080 + 57{,}720$, two LayerNorms 480), head 3,509, and final LayerNorm 240. The reference MLP at $n = 12$ has 332,545 parameters (3,328 for the input layer, five hidden transitions of 65,792, and 257 for the output head), a difference of 7.0%.

Tokenization and training. The vocabulary has 29 tokens: five structural tokens (padding, beginning of sequence, separator, and the two labels -1 and $+1$) plus 24 bit tokens, one for every (coordinate, value) pair, so that a two-layer model can bind a bit to its slot. A training document is BOS followed by 16 examples, each a separator plus the 12 bit tokens of an input and its label token, for a length of 225; the 16 examples of a document are drawn without replacement. The loss is a two-class cross-entropy read at each example's final bit position (next-token prediction restricted to the two labels); bit and separator positions enter no loss. The optimizer is Adam at a constant learning rate 5×10^{-4} with no weight decay or warmup, for 100,000 update steps of 128 examples each (8 documents). The experiment runs at $n = 12$ on the same frozen random tables and datasets as the MLP, at all three accuracy levels, with training seeds 100–102 for each of the three tables: nine runs per level, 27 in total.

Readout. Overlap is measured on a fixed evaluation stream of 237 tokens: BOS, two labeled training examples from the dataset, and 16 unlabeled query examples, at most 6% longer than a training document so that rotary positions stay near the trained range. At each query's final bit position, the two-class softmax gives $g = 2\,P(+1) - 1 \in [-1, 1]$, and the 16 queries index the cube directly; the cosine overlaps, held-out accuracies, and energy bins are computed exactly as for the

MLP, and takeover is defined identically. The evaluation stream is drawn once with a fixed seed (20260913) independent of the training seeds.

Results. All 27 runs complete the budget. Ten runs remain at chance level throughout (terminal document accuracy 0.45–0.56 and both overlaps near zero; the flat lines of Figure 7). In the remaining runs the random overlap rises first and stays ahead: at the 90% level, 5 of 9 runs leave chance, with terminal median overlaps $+0.31$ for parity and $+0.62$ for the random candidate; at the 95% level, 6 of 9, with $+0.19$ and $+0.55$; at the 100% level, 6 of 9, with $+0.29$ and $+0.69$. A single run crosses over (100% level, step 85,726). The terminal energy share at the top frequency is 0.04 at the 90% level, 0.01 at the 95% level, and 0.07 at the 100% level, against 0.47 for the reference MLP on the same datasets: the Transformer ends the budget without building the high-frequency component.

I Design Decisions for the Spectral-Tail Control

The fixed-cost contrast of Section 5.2 varies the tail mass T_3 at fixed C_{sym}. The symmetric arm — varying the low-frequency mass L_1 at fixed C_{sym} — was not run, on the evidence of a per-function difficulty table over the balanced pool (92 functions stratified across C_{sym} layers, three training seeds each): given C_{sym}, the partial correlation of difficulty with T_3 is $+0.56$, against $+0.20$ for L_1. The tail caliber carries the conditional association with difficulty that the low-frequency caliber lacks, so the control matches the tail.

J Extended Related Work

This appendix expands Section 2 into a per-paper discussion, grouped along the same lines.

Spectral bias and its mechanism. Rahaman et al. (2019) characterize spectral bias through Fourier analyses of neural networks and connect it to spectral decay. Xu et al. (2019) state the frequency principle: networks fit target functions from low to high frequencies, which explains strong generalization on natural datasets and weak generalization on parity and randomized targets. On the theoretical side, Jacot et al. (2018) introduce the neural tangent kernel (NTK), under which overparameterized training follows the kernel gradient of a fixed kernel; Lee et al. (2019) extend the linearization to networks of any depth, Du et al. (2019) prove global linear convergence under overparameterization, and Allen-Zhu et al. (2019) establish the NTK equivalence at polynomial width for ReLU networks. Basri et al. (2019) derive per-frequency eigenvalues of the NTK and thereby predict the learning time of each frequency; Cao et al. (2021) recover the same ordering from a functional-approximation analysis on the sphere; Bietti & Mairal (2019) trace the bias to the spectral decay of the kernel's reproducing kernel Hilbert space. Basri et al. (2020) extend the account to non-uniform input densities, where local density rescales convergence time; Bordelon et al. (2020) decompose full learning curves into spectral modes fitted in sequence; Su & Yang (2019) obtain a matching structure from a functional-approximation view of gradient descent; Yang & Salman (2019) develop a fine-grained spectral theory of expressivity and learning; Arora et al. (2019) use a related kernel to separate training speed under random and true labels. On implicit regularization more broadly, Soudry et al. (2018) show that gradient descent on separable data converges to the max-margin direction, and Lin et al. (2017) argue from physics that symmetric, local, and compositional targets admit cheap learning. Our contribution within this line is the observation that low frequency and simplicity, treated as interchangeable in this tradition, rank functions differently.

Simplicity bias and description length. Dingle et al. (2018) prove, via the coding theorem of algorithmic information theory, that the input–output maps of natural systems, such as RNA sequence-to-structure, are biased towards simple outputs: the probability of an output decays exponentially in its approximate Kolmogorov complexity. Valle-Pérez et al. (2019) measure the simplicity of network-representable functions with Lempel–Ziv complexity (Lempel & Ziv, 1976) and show that generalization tracks this bias; Mingard et al. (2019) prove that random perceptrons are a priori biased towards low-entropy Boolean functions, and Mingard et al. (2025) identify an architecture-level prior towards simple functions. Zhang et al. (2017) show that networks fit random labels, calling for an explanation beyond capacity; Arpit et al. (2017) find that networks learn simple patterns before memorizing. Nakkiran et al. (2019) give the dynamical version: SGD fits functions of increasing

complexity. The description-length tradition behind this line starts with Rissanen's minimum description length (Rissanen, 1978) and with Occam's razor formalized in PAC terms (Blumer et al., 1987); Hinton & van Camp (1993) applied description length to network weights, and Schmidhuber (1997) searched for networks of low Kolmogorov complexity. The complexity notion itself originates in Kolmogorov's algorithmic definition of information (Kolmogorov, 1963; 1965), with independent formulations by Solomonoff (1964) and Chaitin (1966); Shannon's entropy (Shannon, 1948) is the distribution-level counterpart. None of these accounts assigns symbolic descriptions a role; our contribution is to introduce symbolic complexity into this line.

Learning Boolean rules. The Fourier representation of Boolean functions and its learning theory (Kushilevitz & Mansour, 1993; O'Donnell, 2014) underlie our spectral coordinate; Friedgut (1998) links low average sensitivity to dependence on few coordinates. On the hardness side, parity is the canonical hard case: it cannot be learned in the statistical-query model (Kearns, 1998), and gradient descent provably fails on it (Shalev-Shwartz et al., 2017; Daniely & Malach, 2020). Barak et al. (2022) nevertheless show that SGD does learn parity near the computational limit: the loss stays flat for a long stretch while a small signal keeps growing inside the network — once the signal crosses a threshold, the loss drops abruptly. Abbe et al. (2022; 2023) parameterize the difficulty by leap complexity. Franco (2006) studied the generalization of Boolean functions implemented in networks, and Watkin et al. (1993) review the statistical mechanics of rule learning. Our experiments reuse this benchmark family to compare the two costs in competition.

Delayed emergence and grokking. Power et al. (2022) report grokking on small algorithmic datasets, and Nanda et al. (2023) reverse-engineer it into memorization, circuit-formation, and cleanup phases. Our takeover finding — the network switches to parity only after interpolating the labels — belongs to this family of delayed transitions.

Symbolic regression and rule discovery. Schmidt & Lipson (2009) search expression spaces for natural laws, and Udrescu & Tegmark (2020) add physics-inspired priors. Petersen et al. (2021) put a network inside the search, training a recurrent policy with risk-seeking gradients to propose expressions. Reading rules out of trained networks is older and larger: Craven & Shavlik (1994) query a trained network as an oracle and fit decision trees to its answers, the survey of Andrews et al. (1995) maps the decompositional and pedagogical families, Zilke et al. (2016) extend the decompositional route to deep networks, extracting and merging rules one hidden layer at a time, and Townsend et al. (2020) trace the whole tradition from a neural-symbolic perspective. Martius & Lampert (2017) and Sahoo et al. (2018) build networks from elementary activation functions whose sparse weights are the equation itself; Liu et al. (2025) prune trained spline networks and snap their edge functions to symbols; Cranmer et al. (2020) distill symbolic laws from the internal functions of graph networks, and Frosst & Hinton (2017) distill a trained classifier into a soft decision tree that generalizes better than one learned directly from the training data. Our work reveals a problem in this line. If a network's preferred completion is not the shortest explanation, reading completions as explanations propagates the mismatch into the recovered rule.