COORDINATE-DEPENDENT GENERALIZATION IN TRANSFORMERS: CONTROLLED INTERVENTIONS AND PROSPECTIVE PREDICTION ON MODULAR ARITHMETIC

Anonymous authors
Paper under double-blind review

ABSTRACT

The same mathematical relation can be learnable or unlearnable for a transformer depending only on the coordinates in which it is presented. We isolate this effect with modular arithmetic tasks whose targets are fixed by theorems, where the presentation can be changed while the underlying relation, the information available to the learner, the training data and the held-out inputs are preserved. Re-expressing both inputs and outputs in discrete-logarithm coordinates turns a nonlinear target into an affine one and raises held-out accuracy from chance to between 0.459 and 0.853 on two theorems, while changing either side alone leaves both at chance. In a second, matched intervention, two presentations that share their relabelled inputs, algebraic degree and agreement with the closest affine map reach 0.977 and 0.006, so the internal structure of the presented target, not these coarser properties, carries the effect. Statistics of the complete presented target, computed before training, correctly predict the outcome on all 18 held-out tasks, ten of them with predictions specified before training. Measurements of the trained networks show that the presentation determines whether a low-complexity solution is available, while weight decay affects whether training selects it. All core results use positional-digit encodings.

1 INTRODUCTION

When a network fits its training data and fails on unseen inputs, the failure is usually attributed to the architecture, the optimizer or the task. There is a fourth candidate that benchmarks rarely control: the representation in which the task is presented. The same relation can be a simple rule in one representation and a complicated map in another, even when both preserve every bit of information, and a network is only ever shown one of them.

A small example makes this concrete. Take the task $r \mapsto r^2 \bmod 7$ on the nonzero residues. In ordinary residue coordinates it is nonlinear. But every nonzero residue can be written as a power of a fixed generator, $r = 3^x$, and then $r^2 = 3^{2x}$, so in these *discrete-logarithm* coordinates the same task reads $x \mapsto 2x \bmod 6$, which is affine. Nothing about the relation or the information it carries has changed; only its form has. Our experiments apply exactly this idea to arithmetic targets that are fixed by theorems.

Figure 1a shows the main instance. Ramanujan's congruence $\tau(p) \equiv 1 + p^{11} \pmod{691}$ fixes, for every prime p, the residue of a classical arithmetic function in terms of $r = p \bmod 691$. Shown this table as digits, a small decoder-only transformer fits every training example and scores 0.003 on held-out residues. In discrete-logarithm coordinates the same theorem becomes $x \mapsto 11x \bmod 690$. With the relation, the information, the held-out residues, the optimizer and the training budget unchanged, held-out accuracy rises to 0.459, and on a second congruence of the same form it rises from 0.000 to 0.853. Because each formula is a theorem (Ramanujan, 1916; Deligne, 1974; Serre, 1987), which we call a *certified target*, the statement that the target is affine in these coordinates is proved rather than assumed. The central observation of the paper is therefore this: *the same mathematical relation, with the same information and the same held-out inputs, can change from a memorization problem to a generalization problem solely because of the coordinates in which it is presented.*

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.

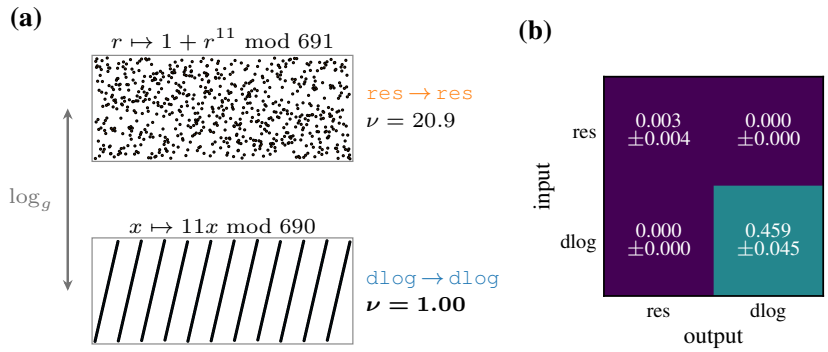


Figure 1: **One theorem, two coordinate systems, two outcomes.** (a) Ramanujan's congruence as two presented label tables on the same 690 inputs, before and after discrete logarithms $x = \log_g r$ and $z = \log_g(y - 1)$; the network is shown one table or the other. In residue coordinates the table is spread across many Fourier components ($\nu = 20.9$); in discrete-log coordinates it is affine ($\nu = 1.00$). (b) Held-out accuracy with input and output coordinates crossed, three seeds, same held-out residues: only the affine cell rises above chance, to 0.459; the other three stay at or below 0.003 (other models and moduli in Table 1).

We develop this observation in three steps. First, the presentation alone changes the outcome, and only a change of both input and output coordinates does so (§3). Second, what matters is the structure of the *presented target*, the map from encoded input to encoded output: generalization follows how close that map is to affine, and a matched design shows that this structure matters even when coarser properties are held fixed (§4); the same structure, measured before training, predicts the outcome on held-out tasks (§5). Third, the structure must be both available and selected: measuring the trained networks shows that the presentation determines whether a low-complexity solution exists, while weight decay promotes its selection under our protocol (§6). Section 7 tests how far the effect extends across architectures, radices, a second field and a published benchmark.

Our contribution is to isolate this effect experimentally. That the learnable class of modular polynomials is narrow is known (Doshi et al., 2024), that a chosen number representation can widen it is known (Zhou et al., 2026; McLeish et al., 2024), and mechanistic work has reverse-engineered the algorithm a grokked network runs on one fixed presentation (Nanda et al., 2023; Zhong et al., 2023). None of this work isolates the presented structure as the cause. We do so with three complementary designs: crossed input and output coordinates, which distinguish an obstacle in the input coordinates, an obstacle in the output coordinates and an effect of the presented target as a whole; a matched construction that holds algebraic degree and best-affine agreement fixed while varying only the structure of a relabelled block; and predictions specified before the runs that scored them. In this setting, algorithmic capability is not representation-invariant. This raises a broader concern for algorithmic benchmarks: one task can present as two problems of different difficulty, and two differently named tasks can present as the same problem, which a simple check finds on our own benchmark (§2).

C1. Two controlled interventions (§3, §4). Re-encoding a certified target in coordinates where it is affine lifts it from chance on two theorems and in two radices, while both mixed coordinate systems stay at chance; with the relabelled inputs, degree and closest affine map fixed, the structure of a relabelled block alone separates 0.977 from 0.006. Both effects replicate on a two-layer LayerNorm transformer.

C2. Prediction from the presented target (§5, §7). A fixed rule on the presented target predicts the outcome on 18 held-out tasks, ten of them with predictions specified before training, all correctly; on 16 new tasks with identical best-affine agreement, ν tracks the ordering of outcomes but not their level. With digit tokens, two of the nine operations of the original grokking benchmark generalize, against eight in published atomic-token runs, as predicted in advance.

C3. Availability and selection, measured on the model (§6). Decoding the function each trained network implements, and varying only weight decay, separates whether a low-complexity solution is available in the presented coordinates from whether training selects it.

2 SETUP: PRESENTED TARGETS, ONE STATISTIC AND THE PROTOCOL

Terms and notation used throughout	
underlying task	the mathematical relation being predicted, for example $a_p \bmod 691$ as a function of $p \bmod 691$
coordinate system	the bijection used to name the inputs and outputs: `res` names a nonzero residue by r itself, `dlog` by its discrete logarithm x, where $r = g^x$ for a fixed generator g
encoding	how coordinates become tokens; here always positional digits in a stated *radix* (10 unless noted)
presented target	the function the network is trained on: the underlying task composed with the input and output coordinate systems; `dlog` → `dlog` presents the affine map $x \mapsto ex \bmod (\ell - 1)$
p, ℓ, r, a_p	a prime, a prime modulus, the residue $r = p \bmod \ell$, and the label, a coefficient of one of 24 classical modular forms reduced mod ℓ
two theorems	$a_p \equiv 1 + p^{11} \pmod{691}$ (Ramanujan) and $a_p \equiv 1 + p^{15} \pmod{3617}$: in residue coordinates each is the power map $r \mapsto 1 + r^e$
ν, ν_g	Fourier complexity of the presented target (defined below), and of the function a trained network computes (§6)

The statistic. For a presented target $f : \mathbb{Z}/\ell \to \mathbb{Z}/\ell$, the statistic ν measures how spread out the Fourier representation of its label table is, averaged over output characters. Write $\omega = e^{2\pi i/\ell}$ and define

$$F[j,k] \;=\; \frac{1}{\ell} \sum_{r \in \mathbb{Z}/\ell} \omega^{jr - kf(r)}, \qquad \nu(f) \;=\; \frac{1}{\ell} \sum_{j,k} \big|F[j,k]\big|, \qquad \varepsilon(f) \;=\; \max_{j} \max_{k \neq 0} \big|F[j,k]\big|. \tag{1}$$

An affine map puts each output character's whole mass on one input character, so $\nu = 1$; a random map at $\ell = 691$ spreads it to about 23.[1] Because ν is computed from the complete label table, it is exact rather than estimated. Cells whose inputs or outputs are discrete logarithms live on $\mathbb{Z}/(\ell - 1)$; they use the same definition with each cell's own input group and output size (Appendix B.1, which also covers the excluded zero residue and out-of-range digit predictions).

Three notions of affine structure share a name, and we keep them apart. *Best-affine agreement* is the fraction of the domain on which the target matches its closest affine map; *piecewise-affine structure* is how few affine pieces it decomposes into; *Fourier concentration* is what ν measures. The first and third can be pulled apart, which is the design of §4; the second and third rise together on our families.

The protocol. Unless noted, each experimental condition, or *cell*, trains a one-layer decoder-only transformer (Vaswani et al., 2017) on radix-10 digits, with AdamW (Loshchilov & Hutter, 2019), a matched 12k-step budget and three seeds. The base model has no LayerNorm, so the read-out analysis of §6 is exact; the results replicate on a standard two-layer LayerNorm transformer. Four details matter. *(i)* The answer is emitted as one softmax per digit position, because these targets are often bijections, so a flat softmax over ℓ classes would have to predict held-out labels it never saw in training; such a head scored zero in practice (Appendix A.1). *(ii)* Held-out accuracy is exact match on all answer digits. *(iii)* The split is over distinct inputs at training fraction 0.8: the 690 nonzero residues are the distinct inputs, and the roughly $14{,}000$ prime-derived examples per run repeat them. *(iv)* The four cells of §3 hold out the same residues. Remaining hyperparameters are in Appendix A.

Named tasks can coincide. Because $a_p \bmod \ell$ depends on a Galois representation that several modular forms can share, forms with different names, weights and levels can define byte-identical label tables: our 138 named tasks are 129 distinct label functions, so holding out a different form can hold out nothing, and string or embedding deduplication (Lee et al., 2022; Sainz et al., 2023) cannot see it. The forms are generated in-house with no external mathematical library and checked at theorem level (Appendix B); we release the benchmark with a detector that compares label tables directly.

[1] The Fourier L^1 (spectral) norm (Shpilka et al., 2017), averaged over output characters; not the low-frequency preference of spectral bias (Rahaman et al., 2019).

Table 1: Held-out accuracy for two certified targets in four coordinate systems: $e = 11$ at $\ell = 691$ ($n = 3$ in radix 10, $n = 8$ in radix 7, $n = 3$ on a standard two-layer LayerNorm transformer, 2L+LN, radix 10) and $e = 15$ at $\ell = 3617$ ($n = 3$). Within a column the data, budget and held-out residues are identical and train accuracy is 1.000 (one 2L+LN `res` $\to$ `dlog` run fit 0.846), so only the presentation differs. The base-model `dlog` $\to$ `dlog` cells at $\ell = 691$ do not meet the 0.5 criterion; the 2L+LN cell meets it on its three-seed mean (one seed at 0.438) and the $\ell = 3617$ cell in every seed. "Predicted by" names the hypotheses of §3 that predict generalization in that cell.

			$\ell = 691$			$\ell = 3617$
Cell	**Map as presented**	**Predicted by**	radix 10	radix 7	2L+LN	radix 10
`res` $\to$ `res`	$r \mapsto 1 + r^e \bmod \ell$	none	0.003 ± 0.004	0.001 ± 0.003	0.000 ± 0.000	0.000 ± 0.001
`dlog` $\to$ `res`	$x \mapsto 1 + g^{ex} \bmod \ell$	H_{in}	0.000 ± 0.000	0.003 ± 0.005	0.000 ± 0.000	0.000 ± 0.000
`res` $\to$ `dlog`	$r \mapsto e \log_g r \bmod (\ell - 1)$	H_{out}	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.001
`dlog` $\to$ `dlog`	$x \mapsto ex \bmod (\ell - 1)$ (**affine**)	all three	$\mathbf{0.459 \pm 0.045}$	$\mathbf{0.284 \pm 0.082}$	$\mathbf{0.624 \pm 0.133}$	$\mathbf{0.853 \pm 0.023}$

3 Intervention I: the coordinates alone change the outcome

We begin with the most direct test of the claim: take a certified target that a network fails to learn, change only the coordinates in which it is presented, and ask which part of the change is responsible.

Two real congruences stay unlearned in residue coordinates. On Ramanujan's congruence train accuracy is 1.000 and held-out accuracy 0.003 in the base model (Table 1), and at most 0.013 over four wider, deeper or longer-trained two-layer variants; the weight-16 congruence at $\ell = 3617$ fails too, while an affine control at that ℓ reaches 0.997. The usual artifacts do not explain these failures: the network fits every training label, no held-out residue appears in training, the same digit head reaches 0.98 or more on affine tasks, and a failing network matches no other simple global rule we tested (Appendix F).

Design. Discrete logarithms turn $\tau(p) \equiv 1 + p^{11} \pmod{691}$ into $\log_g(\tau(p) - 1) = 11x \bmod 690$, and the weight-16 congruence into $15x \bmod 3616$. Crossing input with output coordinates gives four cells, and three hypotheses, each of which predicts that discrete logarithms help, disagree about which cell should generalize. If the obstacle is the input coordinate (H_{in}), `dlog` $\to$ `res` generalizes; if it is the naming of the answers (H_{out}), `res` $\to$ `dlog` does; if generalization follows the structure of the presented target as a whole (H_{spec}), only `dlog` $\to$ `dlog` does. All cell predictions were fixed before any cell was trained.

Result. Only the double change of coordinates helps. Held-out accuracy on Ramanujan's congruence rises from 0.003 to 0.459, while each mixed cell stays at or below 0.003; the same pattern holds in radix 7, on the two-layer LayerNorm transformer (0.624) and on the weight-16 congruence, where the affine presentation reaches 0.853 against at most 0.001 for the other three cells on a modulus five times larger (Table 1; Figure 1b). Each alternative hypothesis predicted success in a different mixed cell, and every mixed cell fails in every column. The radix-7 rerun also removes a digit shortcut: in radix 10 the last digit of $11x \bmod 690$ equals the last digit of x, and in radix 7 no digit is free, yet the pattern is unchanged. Mapping each `dlog` $\to$ `dlog` prediction back through $y \mapsto 1 + g^y$ and scoring it against the original congruence agrees with direct scoring to within 0.001, the residual coming from out-of-range digit outputs (Appendix C), so the change of coordinates loses no information. The network does not compute the discrete logarithm; the presentation supplies it, which makes the intervention a diagnostic rather than a recipe.

The effect is large but not uniform. Against the 0.5 criterion fixed before the runs, the base-model cells at $\ell = 691$ fall short (0.459 in radix 10 and 0.284 in radix 7), the LayerNorm model meets it on its three-seed mean although one seed reaches only 0.438, and the $\ell = 3617$ cell meets it in every seed. The re-encoded $\ell = 691$ presentation is also far more data-hungry than a native affine target, reaching 0.008 at training fraction 0.2 and 0.548 at 0.9. Something about the presented target, then, decides the outcome. The next section asks which property it is.

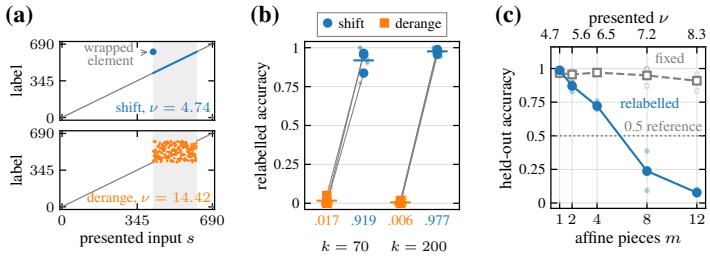


Figure 2: **Same relabelled residues, same best-affine agreement, different structure.** (a) Both variants relabel one block of $r \mapsto 1 + r$ (shaded). A cyclic shift keeps the block affine except for the wrapped element, $\nu = 4.74$; a derangement scatters it, $\nu = 14.43$. (b) Accuracy on the relabelled residues, one line per (k, seed) pair: the shift is learned and the derangement is not, in six pairs of six. (c) The same 200-residue block cut into m affine pieces at fixed relabelled set, best-affine agreement and degree: accuracy on the relabelled residues falls below 0.5 between four and eight pieces. Full values in Tables 9 and 10.

4 Intervention II: The structure of the presented target matters

Intervention I changes a whole presentation at once, so it shows that the presentation matters without saying which of its properties does. Here we change one property while holding the obvious candidates fixed, which gives the paper's cleanest causal contrast.

A first attempt shows that the effect is graded but cannot say why. Let π_k relabel k of the 691 residues and fix the rest, every rung of this ladder presenting $r \mapsto 1 + r$. The underlying target stays affine, so its degree, image size, label entropy and invertibility are constant, while the presented map has algebraic degree 689 at every nonzero rung against 1 at the unpermuted one. Presented ν climbs from 1.00 to the random-map value 23.25, and accuracy falls monotonically with it, Spearman -0.983. But the network scores 0.000 on every relabelled residue at every rung, so any quantity monotone in k explains the ladder, including the fraction of the target left affine. The matched family removes that ambiguity.

The matched family. Two variants permute the *same* contiguous block of k residues at the same offset, so the relabelled residues, the fixed residues, the best-affine agreement and the presented algebraic degree are all identical, and only the structure of the relabelled block differs; ν is our measure of it. A random derangement is spectrally dense, $\nu = 9.44$ and 14.43 for $k = 70$ and 200. A cyclic shift presents two affine pieces and is sparse, $\nu = 4.58$ and 4.74 (Figure 2a). At $k = 200$ the shift is learned on the relabelled residues and the derangement is not, 0.977 against 0.006 (Figure 2b); at $k = 70$ the two reach 0.919 and 0.017, and the sign is positive in six of six (k, seed) pairs. On the two-layer LayerNorm transformer the $k = 200$ block separates the same way, 0.964 against 0.010 in every pair (Table 22).

Algebraic degree cannot account for the difference. Both variants present a permutation of $\mathbb{F}_{691}$, which caps the degree at $\ell - 2 = 689$; we compute it from the reduced interpolant, and it equals 689 in every variant and at every piece count below. Across both families, accuracy on the relabelled residues spans 0.006 to 0.987 at identical presented degree.

Locating the transition. The same 200-residue block, at the same best-affine agreement of 0.711, can be cut into m affine pieces for m from 1 to 12 (Figure 2c; Table 10). Accuracy on the relabelled residues falls monotonically with m, from 0.987 to 0.077, Spearman -0.975, while accuracy on the fixed residues stays at or above 0.909, so the decline is confined to the block. The 0.5 crossing lies between four and eight pieces, where ν lies between 6.5 and 7.2. This family does not separate ν from piece count, which rise together by construction: the sparse variants are sparse *because* they are piecewise affine. What the two interventions establish jointly is that generalization follows how close the presented target is to affine. If that is right, the outcome should be predictable from the presented target before any training, which is the question of the next section.

5 PROSPECTIVE TASK-LEVEL PREDICTION FROM THE PRESENTED TARGET

This is an oracle, task-level diagnostic: the predictor has access to the complete presented target, including labels withheld from network training. It asks whether this training protocol will generalize on a given task, not how to predict unseen labels from the training set alone.

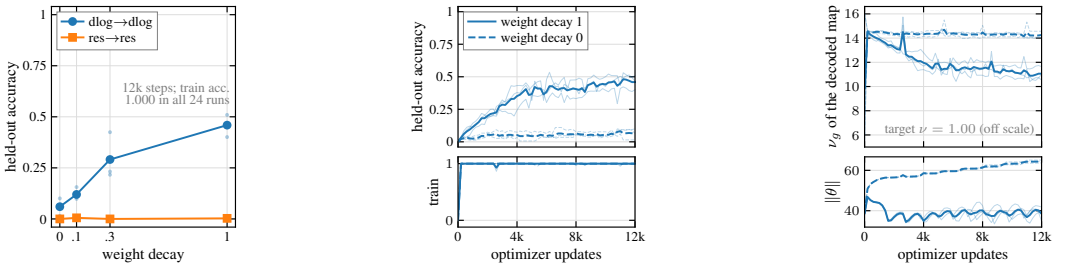
Spearman values for a predictor below use its frozen direction, so a positive value means the statistic orders accuracy as that direction predicts (for ν the score is $-\nu$).

Predictions on held-out tasks. We predicted LEARN or FAIL for 18 held-out tasks, ten of them with predictions specified before training, at moduli 251, 389 and 577, using one fixed rule on the presented target: LEARN exactly when $\nu < 2$, and FAIL otherwise (Table 34). The rule was fixed before the ten two-term tasks at $\ell = 389$ and 577 were trained, which is what makes those ten predictions prospective; the eight power-map tasks were evaluated with the same unchanged rule. The success criterion, specified before the runs as held-out accuracy above 0.7 for LEARN and below 0.1 for FAIL, scores the predictions and is not part of the rule. All 18 are correct on three-seed means, with LEARN tasks at held-out accuracy ≥ 0.94 and FAIL tasks at ≤ 0.0145, so all 18 also fall on the predicted side of the 0.5 criterion. The cutoff is not doing the work: the LEARN tasks sit at $\nu = 1.00$ and the FAIL tasks at $\nu \geq 12.49$, so any cutoff between the two gives the same 18 predictions. The per-task value of ν, the prediction and the three-seed held-out accuracy for all 18 tasks are listed in `held_out_predictions.md` in the released repository (Table 35).

What predicts is presented affineness. To ask which property of the table carries the prediction, we computed eleven statistics from the label table alone on 42 cells, froze each direction on four development cells and scored the other 38 (Appendix D): spectral entropy reaches AUROC 0.997 and ν 0.988, algebraic degree only 0.606. On two families the tournament never saw, 31 cells with every formula and threshold frozen, spectral entropy, ν and ε reach AUROC 1.000 and degree stays at 0.527 (Table 14). But so does *best-affine agreement*, the simplest presentation-aware statistic, added with its direction fixed in advance; it leads on Spearman ($+0.975$ against $+0.874$ for ν), and no spectral statistic beats it by the 0.05 margin we required. What predicts is how close the presented target is to affine, which the spectral statistics measure (Proposition 1).

Where the closest affine map is held constant, ν still tracks accuracy. The matched and piece-count families of §4 hold best-affine agreement and degree constant while the spectral statistics vary, so they test whether ν adds anything. Scored there against accuracy on the relabelled residues, spectral entropy and ν track it across its full range, Spearman $+0.755$ and $+0.743$ on the matched family and $+0.972$ and $+0.966$ on the piece-count family, while best-affine agreement carries no signal, -0.094 on the matched family and constant by design on the other (Table 12); ε tracks on only one of the two. This analysis is post hoc, with formulas and directions frozen and nothing refitted. A follow-up that also matches piece count and jump magnitude, with mean displacement matched to within one sub-block, finds the lower-ν member ahead in 10 of 12 pairs (one-sided sign test $p = 0.019$), though its mean lead of $+0.052$ misses its 0.10 target; the lead is $+0.092$ at nine pieces and $+0.012$ at eleven, where displacement ranks the pairs better than ν (Appendix C).

Order, but not level. Conjugating the four block variants of §4 by multiplication by $d = 7$ or 10 turns each relabelled block into an arithmetic progression while leaving ν, the other spectral summaries, best-affine agreement, degree and the number of affine rules unchanged; local continuity and displacement do change. On these 16 new tasks, with predictions specified before training, raw ν orders accuracy on the relabelled residues within each dilation (Spearman -0.894 and -0.958 against a target of -0.70), and over the original and dilated tasks together it tracks accuracy more closely than local continuity (-0.865 against -0.599) and slightly more closely than mean displacement (-0.822). The measured summaries do not determine the level of accuracy, however: at identical values, dilation by 7 lowers accuracy by up to 0.70 and dilation by 10 by at most 0.27. How the relabelled set aligns with the digits is a plausible explanation, proposed after the results (Appendix D.1). Within these families, then, ν tracks the ordering of outcomes. The prediction says which tasks admit a generalizing solution; it does not say what the trained network does, which we measure next.



(a) weight-decay sweep (b) accuracy along training (c) ν_g and parameter norm

Figure 3: **Presentation and weight decay during training**, $\ell = 691$, $e = 11$, radix 10, 12k steps, three seeds. (a) Held-out accuracy against weight decay for the two extreme cells of Table 1; train accuracy is 1.000 in every run. (b) Held-out and train accuracy of $\texttt{dlog} \to \texttt{dlog}$ at weight decay 1 (solid) and 0 (dashed). (c) The decoded ν_g of the function the network implements and the parameter norm $\|\theta\|$ on the same runs: with weight decay, ν_g decreases as held-out accuracy increases; without it ν_g stays near 14 and the norm grows (Appendix E).

6 What the network implements, and what weight decay contributes

A presented target permits some solutions and excludes others; which one a network ends up implementing is a separate question, and this section measures it directly. Two statements about the entries of Eq. 1, which are exponential sums, follow from Weil's bound (Weil, 1948) and are proved in Appendix B.2. Proposition 1: an affine presented target has $\nu = \varepsilon = 1$, while a degree-e power map with $2 \le e < \ell$ has every non-trivial coefficient at most $(e-1)/\sqrt{\ell}$. Proposition 2: for every hypothesis g, $\mathrm{agree}(f, g) \le 1/q + \varepsilon\,\nu_g$, where agree is the fraction of the domain on which $g = f$, q the output size, ε is certified on the target and ν_g measured on the model; at $\ell = 691$, agreeing with a degree-2 target on half its domain requires $\nu_g \ge 13.11$. The bound excludes a sparse hypothesis on a dense target; it does not exclude memorization.

The function each network implements. Rather than assume the step from bound to network, we decode the prediction function of each trained checkpoint over the whole domain and compute its ν_g (Appendix E). Across a $200\times$ capacity range, 3,872 to 799,232 weights, networks trained on the degree-1 target converge to $\nu_g = 2.3$ to 2.8 against a target ν of 1.00, and networks trained on the degree-2 target sit at 23 to 24, at the random-map value and above the 13.11 the bound requires (Table 27). In the 2×2 of §3, the $\texttt{dlog} \to \texttt{dlog}$ network implements a function at $\nu_g = 10.9$ and the three failing networks at 22.6 to 24.3. Part of that is forced: fitting 0.8 of a dense target imposes a floor on ν_g through Proposition 2, 21.0 for the degree-2 target and 6.16 for $\texttt{res} \to \texttt{res}$ (Appendix E), so a failing network's high ν_g is partly what fitting its training labels requires. The failing networks are memorizers with spectral room to spare, so the question is not whether a generalizing solution is permitted but whether training selects it.

The representation carries the function, but not in a Fourier-specific way. Three interventions on the trained model, detailed in Appendix E.1, fix what the representation results do and do not show. On independently trained $\texttt{dlog} \to \texttt{dlog}$ checkpoints averaging 0.474, removing the top 40 of 690 Fourier components of the read-out gives 0.000 against 0.450 for 40 random ones, but removing as much energy along about ten singular directions collapses the model as fully, to 0.048. Supplying a failing degree-2 task with twenty character tokens lifts it from 0.007 to 0.785, and twenty structured but spectrally dense non-Fourier tokens do the same, 0.777, where unstructured relabellings leave it at 0.004. And a Fourier fit to the read-out on training coordinates alone, which raises held-out accuracy from 0.474 to 0.627, is beaten in every checkpoint by the same fit to the one-hot training labels with no network, 0.975, so that gain is interpolation of the presented target rather than structure the network adds. The network depends on the structure the presentation makes available, and the Fourier basis is one description of that structure rather than the mechanism.

6.1 PRESENTATION AND WEIGHT DECAY ARE TWO SEPARATE FACTORS

If the boundary is a matter of selection, the training protocol should be able to move it on the side where a generalizing solution exists and not on the other. We reran the two extreme cells of §3 across weight decay 0, 0.1, 0.3 and 1.0, changing nothing else (Table 21). At the matched 12k budget, held-out accuracy on the affine presentation climbs with weight decay, from 0.060 at 0 to 0.459 at 1.0 (Figure 3a), while the non-affine presentation stays between 0.000 and 0.005 throughout. Weight decay does not merely speed training up: at $40{,}000$ steps the affine presentation reaches 0.080 without it, against 0.494 with it (Appendix E.2).

The effect is visible in the decoded function. With weight decay 1, ν_g falls by 3.45 on average between step 200 and step 12,000 while the parameter norm falls by 8.3 and held-out accuracy rises by 0.428; without weight decay, ν_g moves by 0.30 and stays near 14, the norm grows by 13.6, and accuracy moves by 0.039 (Figure 3b,c). The fall in ν_g is concurrent with the rise in accuracy in every seed, not ahead of it, so it describes what weight decay does without predicting when. At the matched budget neither factor alone comes close to the two together: 0.003 with weight decay in residue coordinates, 0.060 in discrete-log coordinates without weight decay, and 0.459 with both. Under this protocol, weight decay promotes the selection of a sparse solution (Varma et al., 2023; Nanda et al., 2023; Morwani et al., 2024), and the presentation determines whether one exists.

7 HOW FAR THE EFFECT EXTENDS

The remaining question is whether the effect belongs to one architecture, one modulus, one field or one benchmark. Transformer, MLP and LSTM under one encoding give degree-1 accuracy ≈ 0.99 and degree-2 ≈ 0 at $\ell = 389$, and the coordinate effect survives the change of architecture (Table 32): at the matched 12k budget the LSTM reaches 0.525 and the MLP 0.391 in discrete-log coordinates, while both mixed coordinate systems sit at 0.000. Input radices 2, 7 and 16 leave the degree-1 versus degree-2 contrast intact (Table 24). At $\ell = 10007$, a $14\times$ larger group, degree 1 is learned at 0.997 and 0.937 at two model sizes and degree 2 fails at 0.001 and 0.000. At $\ell = 691$, neither a $4.5\times$ larger training fraction nor a 120,000-step budget lifts a failing cell above 0.0207 (Figure 6a, Table 26).

A second field. Keeping the task's shape and changing the field, $r \mapsto r^e$ on $\mathbb{F}_{691}^*$ in radix 10 becomes $w \mapsto w^e$ on $\mathrm{GF}(2^n)^*$ in radix 2, where squaring is linear, so an exponent that is a power of two presents an affine map ($\nu = 1$). Three of four predictions are confirmed (Table 33): the tested power-of-two exponent is learned in element coordinates at 1.000, the other tested exponents fail there at mean 0.003, and ν orders accuracies at Spearman -0.837. The fourth splits: every discrete-logarithm cell beats its element cell, 6 of 6, but the absolute 0.60 threshold is missed in 3 of 6.

Atomic versus digit tokens on published benchmarks. Doshi et al. (2024) find that networks generalize on modular polynomials of the form $h(g_1(a) + g_2(b)) \bmod p$ and fail outside it. On our reading their class is the affine class after per-operand coordinate changes: with one token per operand, the embedding and unembedding rows supply g_1, g_2 and h for free, leaving addition at $\nu = 1$. This reading reproduces their classification on 5 of 6 tasks on the quadratic MLP of their construction and on 3 of 6 on our ReLU transformer, so what is reachable depends jointly on encoding and architecture (Appendix F.1). It also makes a prediction about tokens. On the nine operations modulo 97 of Power et al. (2022), the reachable-coordinates rule reproduces the published record on 5 of 9 with no training and never predicts LEARN wrongly (Table 29); with digit tokens the free relabellings disappear, the plain ν applies, and only $x + y$ and $x - y$ are predicted learnable. We trained all nine with digit tokens, with predictions specified before the runs: two generalize against eight in the published atomic-token runs, all nine land on the predicted side of the 0.5 criterion, and eight of nine meet their bars, the conditional operation reaching 0.206 against a bar of 0.10 (Table 30). The comparison is with the published runs rather than a matched atomic-token condition in our protocol, so it is a predicted outcome, not an isolated effect of tokenization; some runs also fit their training set incompletely, the lowest at train accuracy 0.701 (Appendix F.1).

8 Related work

Grokking work explains how structured representations emerge: phase behaviour (Liu et al., 2022; Gromov, 2023), circuit competition (Varma et al., 2023), the Fourier algorithms inside grokked models (Nanda et al., 2023; Zhong et al., 2023; Li et al., 2025), feature-learning dynamics for modular addition and group composition (He et al., 2026a;b; Chen et al., 2026; Sivasankar, 2026), and Fourier-type structure in non-neural feature learners (Mallinar et al., 2025). That line, and circuit-level interpretability (Olah et al., 2020), asks what a network learned on one presentation; we ask which presentations make a structured solution available, then measure the trained network against the answer, a question that comes before reverse-engineering. Representation design shows that Fourier features (Tancik et al., 2020) or arithmetic-aware encodings (Zhou et al., 2026; Hwang et al., 2026; McLeish et al., 2024) make hard functions learnable, and transformers' difficulty with modular and multi-digit arithmetic is established (Dziri et al., 2023; Anil et al., 2022; Zhou et al., 2024); our work is a diagnostic of when such a method can work. Number-theoretic data as a substrate follows He et al. (2025); Zubrilina (2025); Charton (2024); Davies et al. (2021), and the affine versus nonlinear split belongs to statistical-query and leap-complexity accounts (Kearns, 1998; Abbe et al., 2023; Barak et al., 2022), short of an impossibility claim (Appendix B.3).

9 Scope and limitations

The paper establishes selection among permitted solutions, not expressivity, and a conditional predictor, not a theory of learnability. Four limits follow.

The step from bound to network is unproved. The theory bounds hypotheses of a given Fourier complexity, not networks of a given size. Fourier complexity, parameter count and optimization accessibility are distinct quantities, and the paper relates only the first to the outcome. Why gradient descent with weight decay lands on the sparse solution rather than another permitted one is unanswered; four optimizer-side diagnostics do not separate the cells (Appendix E.2).

Nothing here is a Fourier-specific representation (§6). Hook placement bounds but does not settle how much measured sparsity is the encoding passing through (Appendix A.1), and the fixed-piece-count lead of §5 misses its 0.10 margin.

The statistic predicts the side within the studied regime, not the sample complexity or the near-boundary case. The re-encoded $\ell = 691$ cell needs far more data than a native affine target (§3), and at matched $\nu \approx 7$, piecewise-affine maps score 0.009 and perturbed-affine maps 0.78 (Appendix D.2). On untouched families best-affine agreement predicts as well as any spectral statistic, so the defensible claim concerns presented affineness, and which statistic carries the finding was decided after the results.

Every core result uses a positional-digit encoding. The effect replicates across four input radices and three architectures and transplants to $\mathrm{GF}(2^n)$, but independence from that encoding family is not established; on the Boolean cube ν orders held-out accuracy at Spearman -0.956, but the hard cells fail to fit their training set, so the ordering transfers and the regime does not (Appendix F.2).

10 Conclusion

On digit-encoded arithmetic with theorem-certified targets, whether a network generalizes depends on the coordinates in which the target is presented, not only on its mathematical content. The dependence is shown by intervention, predictable from the presented target before training, and replicated on a standard two-layer LayerNorm transformer; the presentation determines whether a low-complexity solution exists, and under our protocol weight decay promotes its selection. How the learner chooses among permitted solutions remains open, as does independence from the digit encoding. The practical lesson is short: before concluding that a model cannot learn a relation, measure the relation in the coordinates the model is shown.

Ethics statement

This work uses no human subjects and no personal data, and all task data is generated from classical mathematical objects. The primary societal consideration is a positive one: §2 and Appendix A.1 identify evaluation artifact modes, including a benchmark contamination mode invisible to standard deduplication, and we release the diagnostic so that other benchmark builders can run it before publishing.

Reproducibility statement

The task family is dependency-free and deterministic. The released code includes the experimental configurations, the analysis scripts that produce every reported number, and the stored results, and Appendix G gives an overview of the experiments and their supporting results. Analyses made after seeing the data are labelled post hoc where they appear. Proofs are in Appendix B.2, hyperparameters in Appendix A, and the evaluation failure modes we found are pinned by tests in the released code (Appendix A.1).

Code and data availability

Code, task data and the stored results are available at `https://anonymous.4open.science/r/modular_arithmetic-CB46/`, together with the analysis scripts and the supporting experimental results indexed in Table 35. The benchmark is released separately as ArithBench-Galois: 138 tasks from 24 eigenforms, each a `.npz` of input-label pairs with a manifest recording form, weight, level, modulus, encoding and prime limit. Generation depends on no external mathematical library, so the ground truth can be audited rather than downloaded. The collision detector of §2 is released as a standalone procedure to run before publishing an arithmetic benchmark.

Compute

All training runs were performed on a single NVIDIA GeForce RTX 4080 SUPER GPU (16 GB). The arithmetic layer, the theory verification, the predictor tournament and every analysis script run on CPU; the checkpoint analyses of §6 (decoded ν_g, ablation and reconstruction) train nothing, while the supplied-feature experiment trains new models.

AI use statement

We used Claude to assist with code writing and manuscript polishing. The tool did not generate scientific content, design experiments, analyze data, or select citations. All edits were reviewed and approved by the authors, who take full responsibility for the final manuscript.

References

Emmanuel Abbe, Enric Boix-Adsera, and Theodor Misiakiewicz. SGD learning on neural networks: Leap complexity and saddle-to-saddle dynamics. In *Conference on Learning Theory (COLT)*, 2023.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham 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 (NeurIPS)*, 2022.

François Charton. Learning the greatest common divisor: Explaining transformer predictions. In *International Conference on Learning Representations (ICLR)*, 2024.

Zitong Andrew Chen, Junaid Hasan, Akhil Srinivasan, Hemkesh Bandi, and Jarod Alper. Multiplication beyond groups: Stratified Fourier mechanisms in transformer circuits. *arXiv preprint arXiv:2607.07066*, 2026.

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600(7887):70–74, 2021.

Pierre Deligne. La conjecture de Weil. I. *Publications Mathématiques de l'IHÉS*, 43:273–307, 1974.

Darshil Doshi, Tianyu He, Aritra Das, and Andrey Gromov. Grokking modular polynomials. *arXiv preprint arXiv:2406.03495*, 2024.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Andrey Gromov. Grokking modular arithmetic. *arXiv preprint arXiv:2301.02679*, 2023.

Jianliang He, Leda Wang, Siyu Chen, and Zhuoran Yang. On the mechanism and dynamics of modular addition: Fourier features, lottery ticket, and grokking. *arXiv preprint arXiv:2602.16849*, 2026a.

Jianliang He, Leda Wang, Fengzhuo Zhang, Siyu Chen, and Zhuoran Yang. Neural networks provably learn spectral representations for group composition. *arXiv preprint arXiv:2606.02993*, 2026b.

Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Alexey Pozdnyakov. Murmurations of elliptic curves. *Experimental Mathematics*, 34(3):528–540, 2025. arXiv:2204.10140.

Hyunsang Hwang, Suhyun Bae, and Donghun Lee. Prime Fourier embeddings: A principled basis for modular arithmetic. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

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

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Chenyang Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Tianyi Zhou. Fourier circuits in neural networks and transformers: A case study of modular arithmetic with multiple inputs. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pp. 523–531. PMLR, 2025. arXiv:2402.09469.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Neil Mallinar, Daniel Beaglehole, Libin Zhu, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Emergence in non-neural models: Grokking modular arithmetic via average gradient outer product. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, and Tom Goldstein. Transformers can do arithmetic with the right embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *International Conference on Learning Representations (ICLR)*, 2024.

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.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001.

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 A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 5301–5310, 2019.

Srinivasa Ramanujan. On certain arithmetical functions. *Transactions of the Cambridge Philosophical Society*, 22(9):159–184, 1916.

Kenneth A. Ribet. On modular representations of $\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$ arising from modular forms. *Inventiones Mathematicae*, 100(2):431–476, 1990.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Jean-Pierre Serre. Sur les représentations modulaires de degré 2 de $\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$. *Duke Mathematical Journal*, 54(1):179–230, 1987.

Amir Shpilka, Avishay Tal, and Ben Lee Volk. On the structure of Boolean functions with small spectral norm. *Computational Complexity*, 26(1):229–273, 2017.

Achyuthan Sivasankar. Circuit synchronization precedes generalization: A causal precursor to grokking. *arXiv preprint arXiv:2606.12966*, 2026.

H. P. F. Swinnerton-Dyer. On ℓ-adic representations and congruences for coefficients of modular forms. In *Modular Functions of One Variable III*, volume 350 of *Lecture Notes in Mathematics*, pp. 1–55. Springer, 1973.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 7537–7547, 2020.

The LMFDB Collaboration. The L-functions and modular forms database. `https://www.lmfdb.org`, 2024.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

André Weil. *Sur les courbes algébriques et les variétés qui s'en déduisent*. Number 1041 in Actualités Scientifiques et Industrielles. Hermann, Paris, 1948.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *International Conference on Learning Representations (ICLR)*, 2024.

Tianyi Zhou, Deqing Fu, Mahdi Soltanolkotabi, Robin Jia, and Vatsal Sharan. FoNE: Precise single-token number embeddings via Fourier features. In *International Conference on Learning Representations (ICLR)*, 2026.

Nina Zubrilina. Murmurations. *Inventiones mathematicae*, 241:627–680, 2025.

APPENDIX

The appendix is organized so that a reader with a specific concern can go straight to the section that answers it. Experimental subsections follow one order throughout: the question, the design, the result, and what it establishes. Code, configurations and stored results for every experiment are in the released repository.

If your question is	**see**
Is the evaluation valid? Splits, output head, leakage, the 18 failure modes we found	Appendix A (protocol), A.1 (validity conditions)
Are the targets and the theory correct? Task construction, Fourier conventions, proofs	Appendix B, B.1, B.2
The two interventions in full: radix-7 control, second theorem, ladder, matched and piece-count families	Appendix C
How were the predictors defined, scored and validated?	Appendix D, D.1, D.2
What does the trained network implement? Decoded ν_g, ablations, weight decay	Appendix E, E.1, E.2
Does it hold elsewhere? Architectures, radices, scale, long budgets, published benchmarks, Boolean cube	Appendix F, F.1, F.2
Where is each claim's evidence, and which predictions preceded the runs?	Appendix G

A EXPERIMENTAL SETUP AND EVALUATION

Every run comes from a deterministic, seeded pipeline. Unless stated otherwise a cell trains a one-layer decoder-only transformer (Vaswani et al., 2017) over a ten-token digit vocabulary with a compositional output head, optimized full-batch with AdamW (Loshchilov & Hutter, 2019) at a matched 12,000-step budget, split over inputs rather than over primes. Every value below was read from the stored run configurations, that is, what the runs used rather than the library defaults, which differ. Throughout the paper, $\pm$ is the standard deviation across seeds, population form (ddof $= 0$).

Architecture. $d_{\text{model}} = 256$, 8 heads, 1 layer, MLP width 1024, ReLU, **no LayerNorm**, learned positional embeddings; 799,232 parameters at $\ell = 691$. The absence of LayerNorm is load-bearing for §6, and it is worth saying exactly what it buys. Attention and the MLP remain nonlinear with or without it. What is exact in this regime is the additive decomposition of the residual stream into component contributions, followed by a linear unembedding, which is the decomposition the read-out analysis operates on. The stream also additively retains the input encoding, which is why a random-matrix null is invalid there (mode A10).

Output head. The answer is emitted as $\lceil \log_b \ell \rceil$ digits in radix b, one softmax per digit position, rather than one flat softmax over ℓ classes. This is required, not cosmetic: these targets are frequently bijections, so under an input split the train and test label sets are disjoint, so a flat head has no supervision for the held-out classes; it scored zero in practice (mode A3). The radix b is a scientific parameter, since the shortcut audit of §3 turns on it, and is 10 unless stated, 7 in the matched-radix control, 2 over $\mathrm{GF}(2^n)$.

Optimization. AdamW, learning rate 10^{-3}, $\beta = (0.9, 0.98)$, **weight decay 1.0**, 100 warmup steps, **full batch**, 12,000 steps, float32, no gradient clipping, no dropout, no decay after warmup. The high weight decay and full-batch regime follow the standard grokking protocol, which is what makes the delayed-generalization measurement of §6 comparable to that literature. Given §6.1, weight decay is not only a comparability choice: it is one of the two factors the result is about, and the sweep reported there varies it deliberately while holding everything else in this appendix fixed.

Data. Primes below 2×10^5, giving 17,984 primes; no cap on examples. The split is over distinct input token sequences at a training fraction of 0.8, giving roughly 14,000 training examples per run. In the four intervention cells every cell sets `drop_zero_residue` and shares a split key, so all four hold out *the same residues*; without the second condition each coordinate system gets its own held-out set and the comparison is confounded.

Step counts. The trainer iterates over `steps + 1` and applies its update before evaluating, so a nominal 12,000-step run performs 12,001 updates and the record labelled step 0 is already one update in. Every step label in the trajectory and grokking measurements is shifted by one update on that account, including the memorization step and threshold crossings quoted in §6 and mode A12; the shift is uniform, so no comparison between runs is affected. The initialization probes of Appendix E.2 are exempt, because they build the model fresh and never call the trainer, so they measure initialization as stated.

Evaluation. Test accuracy is exact match on all answer digits. The interval is 2,000 steps except where resolution is the object of study: 100 for the grokking measurement of §6 (at 2,000 the memorization step and the threshold crossing fall in one bucket and force a delay of exactly 1.0 by construction, mode A12), 1,000 for the capacity sweep, 500 for the Doshi et al. arms, 5,000 for the 120,000-step budgets. Table 2 lists everything that varies between experiments, with a pointer from each row to where the result appears.

Seed counts, stated plainly. Three is the default; eight, ten or two are used where noted. The capacity sweep is single-seed, adequate for a contrast of 0.986 against 0.000 and not for a near-boundary quantity; §9 says where thin seed counts support a concession rather than a claim. The analyses of §6 and Appendix E that read existing checkpoints use all ten available seeds. Hardware and total compute are stated in the Compute statement following the main text.

Confidence intervals. Every 95% interval is a bootstrap over the independent unit of its comparison, which differs between experiments: training seeds for the intervention gaps (base 7, eight seeds; $\ell = 3617$, three seeds), target pairs for the fixed-piece-count differences of Table 11, checkpoints for the direct-label baseline of Table 17, and cells for the AUROC intervals of Table 14.

A.1 Conditions for a valid comparison

A comparison between presentations is valid only under conditions that several of our own early experiments violated. Six are required. *(i)* The split is over distinct inputs, never over primes: 18k primes collapse onto ℓ residues, so a per-prime split puts nearly every input on both sides and gave 100% held-out accuracy on a target the network had not learned. *(ii)* The output representation must support prediction of held-out labels: with one embedding row per residue the held-out rows never receive a gradient, and with a flat softmax over ℓ classes the held-out labels of a bijective target are classes never seen in training; the digit encoding with one softmax per digit position resolves both. *(iii)* The held-out residues are held fixed across coordinate systems, otherwise each presentation gets its own held-out set and the comparison is confounded. *(iv)* Every generated target and every stored configuration is checked against the intended task before analysis, by regenerating the task and verifying it against the run's own train accuracy. *(v)* Duplicate task functions are detected: 138 named tasks are 129 distinct label functions (Appendix B). *(vi)* A control must match the quantity that matters, energy or rank, not the count of components; this last condition changes the reading of the representation experiments and is stated again beside them (Appendix E.1).

One correction note is needed to read the supply results. An earlier implementation reused a random feature draw across some supplied-feature conditions, so two arms that should have been independent draws were two training seeds on one draw. The affected comparisons were rerun with three independent draws per arm; the corrected results in Table 20 supersede the earlier ones. Table 3 lists all eighteen modes in one line each.

Table 2: What varies between experiments. Everything unlisted matches the defaults above. Parameter counts are at the stated modulus; ranges arise where the sequence length or answer-digit count varies within an experiment.

experiment	where	arch	d / d_{mlp} / L	params	steps	seeds	other
coordinate intervention	§3, Table 1	transformer	256 / 1024 / 1	799,232	12k	3	$\ell = 691$, base 10
40k convergence check	App. C	transformer	256 / 1024 / 1	799,232	40k	2	–
matched radix-7 control	App. C	transformer	256 / 1024 / 1	798,208	12k	**8**	input and output radix 7, 4 answer digits
graded ladder	§4, Table 8	transformer	256 / 1024 / 1	799,232	12k	3 perm.	$\ell = 691$
Fourier supply	§6, Table 20	transformer	256 / 1024 / 1	0.80–0.81M	12k	6*	*runs per cell: 3 feature draws × 2 seeds (random, scramble); sequence length 5–24
grokking, fine grid	§6	transformer	256 / 1024 / 1	799,232	12k	3	eval every 100 steps
architecture arms (boundary)	§7	MLP, LSTM	256 / 1024 / 2	1.06M, 2.13M	12k	2	$\ell = 389$, degrees 1–3
architecture arms (rescue)	§7, Table 32	MLP, LSTM	256 / 1024 / 1	0.54M, 1.08M	12k	2	$\ell = 691$
$\mathrm{GF}(2^n)$ transplant	§7, Tables 32, 33	transformer	256 / 1024 / 1	0.80M	12k	3	base 2, $n = 8, 10, 12$
scale, $\ell = 691$ large	App. F, Table 25	transformer	512 / 2048 / **2**	6,319,616	12k	2	–
scale, $\ell = 10007$ large	App. F, Table 25	transformer	512 / 2048 / **2**	6,330,880	12k	2	5 answer digits
scale, $\ell = 10007$ small	App. F, Table 25	transformer	256 / 1024 / 1	804,864	12k	2	–
capacity sweep	App. E	transformer	16–256 / 64–1024 / 1	3,872–799,232	**20k**	1	heads 2–8; longer budget
training-fraction grid	App. F, Figure 6a	transformer	256 / 1024 / 1	799,232	12k	3	fractions 0.2–0.9
long budget, $\mathbb{Z}/\ell$	App. F, Table 26	transformer	256 / 1024 / 1	799,232	**120k** / 60k	2	weight decay 0.1, 1.0, 3.0
long budget, $\mathrm{GF}(2^n)$	App. F, Table 26	transformer	256 / 1024 / 1	0.80M	**120k** / 60k	2	weight decay 0.1, 1.0, 3.0
Doshi et al., ReLU arm	§7, Table 28	transformer	128 / 512 / 1	222,592	25k	2	$\ell = 97$, atomic-pair tokens, flat head, fraction 0.5
Doshi et al., quadratic arm	§7, Table 28	MLP	128 / 512 / **2**	521,984	25k	2	**quadratic activation**, otherwise as above
second certified negative	§3, App. C, Table 1	transformer	256 / 1024 / 1	802,048	12k	3	$\ell = 3617$, weight-16 form, 4 answer digits
held-out predictions	§5	transformer	256 / 1024 / 1	799,232	12k	3	$\ell = 251, 389, 577$
weight decay × presentation	§6.1, Table 21	transformer	256 / 1024 / 1	799,232	12k	3	**weight decay 0, 0.1, 0.3, 1.0**
40k convergence control	§6.1	transformer	256 / 1024 / 1	799,232	**40k**	3	weight decay **0**
repaired Boolean cube	App. F.2, Table 31	transformer	128 / 512 / 1	207.5k–207.7k	**8k**	3	4 heads; $(\mathbb{Z}/2)^n$, $k = 6$ carrier, 64 classes
standard two-layer transformer	App. F, Table 22	transformer	256 / 1024 / **2**	1,589,504	12k	3	**LayerNorm**, 8 heads
two layers, no LayerNorm	Tables 17, 23	transformer	256 / 1024 / **2**	1,586,944	12k	**5** / 3	seeds: 5 for dlog, 3 for res
input-base ablation	App. F, Table 24	transformer	256 / 1024 / 1	798.7k–800.8k	12k	**2**	$\ell = 389$, **input radices 2, 7, 16**; answer radix 10
ν_g trajectory	App. E	transformer	256 / 1024 / 1	799,232	12k	3	decoded at intervals, no training

Table 3: The eighteen evaluation failure modes found during the project, each of which produced a publishable-looking result and none of which a shuffled-label control catches. All are pinned by tests in the released code repository, which gives the complete account of each.

mode	what went wrong
A1	residue encoding with a per-prime split: 18k primes collapse onto ℓ inputs, so nearly every input is on both sides
A2	one embedding row per residue: held-out rows never receive a gradient (norms 0.934 trained against 0.052 held out)
A3	a flat softmax over ℓ classes under an input split: on a bijective target the held-out labels are classes never seen in training, and the head scored zero
A4	a modulus above the Deligne bound, so the reduction is injective and the label is the raw integer
A5–A9	a tokenizer that leaked answer length; a positive control solvable by copying; duplicated evaluation inputs; seed averaging over failed runs; a difficulty label from the statistic later used to predict it
A10	a random-matrix null for the residual stream without LayerNorm, which keeps the structured digit encoding
A11	"maximum gap over snapshots > 0.5" as a decision rule
A12	a 2000-step evaluation interval
A13	a 0.60 threshold set from an $n = 1$ pilot in the $\mathrm{GF}(2^n)$ arm
A14	a control that changed the exponent when copyability came from the radix
A15	a power-matched ablation control inside the Fourier basis when the components of interest carry 66% of the spectrum

mode	what went wrong
A16	a five-component reconstruction of 690 classes
A17	a random feature draw reused across supplied-feature conditions, so two supply arms meant to be independent trained on one draw
A18	piece-count rotations that are distinct only modulo the sub-block length

B MATHEMATICAL BACKGROUND AND PROOFS

Construction. The 24 eigenforms are built from eta quotients and Eisenstein series using FFT power-series arithmetic modulo ℓ, at weights 2–26 and levels 1–36, covering CM and non-CM forms. The generator depends on no external mathematical library, LMFDB (The LMFDB Collaboration, 2024), SageMath or PARI, so the ground truth can be audited rather than downloaded.

Validation. Correctness is established at theorem level. Hecke multiplicativity $a_{mn} = a_m a_n$ for coprime m, n holds on all 24 forms; Deligne's bound $|a_p| \leq 2p^{(k-1)/2}$ holds on all 24 forms; complex multiplication is measured from the coefficient sequence rather than asserted from the registry, which caught an error in our own registry; an independent point-counting cross-check sharing no code with the generator agrees on all 29 primes below 120; and three congruences, Ramanujan's $\tau(p) \equiv 1 + p^{11} \pmod{691}$, the weight-16 congruence at $\ell = 3617$, and the mod-11 congruence $\tau(p) \equiv a_p(X_0(11))$, an instance of Serre's weight recipe (Serre, 1987; Swinnerton-Dyer, 1973), hold exactly on every prime below 2×10^5 where they apply (17,983 or 17,984).

Difficulty regimes. Table 4 reports three Galois-image regimes against the measured accuracy ceiling and the maximum character correlation ε on the same tasks. Throughout this appendix ε is the centred sample analogue of the uncentred ε of Proposition 2: the character correlation of the label table with its marginal removed, the right quantity against a marginal-preserving permutation null. What the regime labels are should be stated precisely, because it sets what the agreement means. They are not theorem-certified image classifications. They are assigned from measured predictability, spectral support, zero density, a complex-multiplication measurement taken from the coefficient sequence, and a finite-sample congruence search over the first 3,000 primes. The spectral analysis then reads those labels, and evaluates both ε and the accuracy ceiling at the same analysis-selected best modulus. So what the table reports is agreement between two empirical arithmetic diagnostics computed from the same eigenvalue data, rather than a Fourier measurement matching an independent theorem-certified classification. Read that way the agreement is still strong: the excess of ε over the permutation null, which is the quantity the table's own columns carry, orders the regimes as the theory implies at Spearman $+0.905$, and ε predicts the measured ceiling at Spearman $+0.924$.

Table 4: Difficulty regimes measured from arithmetic alone, and the maximum character correlation ε measured on the same tasks. The 64 large-image tasks are structureless up to the determinant relation of Appendix B and double as a leak detector: generalization on them substantially above what that relation permits indicates a pipeline bug rather than a result.

Regime	n	Ceiling (mean)	Ceiling range	ε	Permutation null
reducible	25	1.000	[1.000, 1.000]	0.6502	0.0135
dihedral	41	0.666	[0.421, 0.877]	0.4044	0.0149
large-image	64	0.110	[0.003, 0.257]	0.0279	0.0189

The large-image excess is a determinant effect, not a leak. A first null treated the large-image tasks as random maps and produced an excess that looked like leakage. It was the null that was wrong: $\det \rho(\mathrm{Frob}_p) = p^{k-1}$ constrains the image, so a random-matrix null over $\mathrm{GL}_2(\mathbb{F}_\ell)$ does not describe it. A null that respects the determinant relation reproduces the observed ε in the large-image regime and leaves the reducible and dihedral excess untouched (Table 5).

Task collisions. Because $a_p \bmod \ell$ is determined by the mod-ℓ Galois representation rather than by the form, and forms of different weights and levels can share one such representation (Serre,

Table 5: The large-image excess is a determinant effect, not a leak. A null that respects $\det \rho(\mathrm{Frob}_p) = p^{k-1}$ reproduces the observed ε in the large-image regime and leaves the reducible and dihedral excess untouched. The permutation null is re-drawn per table, which is why it differs from Table 4 in the third decimal. Both tables report the centred character correlation.

Regime	n	**observed** ε	**permutation null**	**determinant null**
reducible	25	0.6502	0.0126	0.0421
dihedral	41	0.4044	0.0150	0.0285
large-image	64	0.0279	0.0190	**0.0272**

1987; Ribet, 1990), different named forms can define byte-identical tasks. The counts refer to two populations. Across the full generator registry, every pairing of a named form with a modulus the generator supports, a larger population than the release, 147 pairs of named forms define byte-identical tasks. Within the released benchmark, the 138 named tasks reduce to 129 distinct label functions. The regime analysis of Table 4 uses the 130 released tasks that carry one of its three regime labels ($25 + 41 + 64$). The collision detector compares label tables directly and is released as a procedure to run before publishing an arithmetic benchmark.

B.1 Fourier conventions for each presentation

Eq. 1 is written for a map $\mathbb{Z}/\ell \to \mathbb{Z}/\ell$. Every spectrum in the paper uses the general form behind Proposition 2: for a presented map $f : G \to \mathbb{Z}/q$ with G cyclic of order n,

$$F[j,k] = \frac{1}{n} \sum_{x \in G} \omega_n^{jx}\, \omega_q^{-kf(x)}, \qquad \nu(f) = \frac{1}{q} \sum_{j,k} \big|F[j,k]\big|, \qquad \varepsilon(f) = \max_{j} \max_{k \neq 0} \big|F[j,k]\big|, \quad (2)$$

with $\omega_m = e^{2\pi i/m}$, which reduces to Eq. 1 when $n = q = \ell$. The factor $1/n$ makes each output column the transform of a unit-modulus function, and the factor $1/q$ averages over output characters, so an affine map between groups of equal order has $\nu = 1$ in every cell. Each coordinate system fixes G and q: a residue input uses $G = \mathbb{Z}/\ell$, a discrete-log input $G = \mathbb{Z}/(\ell - 1)$, a residue output $q = \ell$ and a discrete-log output $q = \ell - 1$. The mixed cells `dlog` $\to$ `res` and `res` $\to$ `dlog` therefore have $n \neq q$, and the target values of ν in Table 16 are computed this way.

The zero residue. The residue $r = 0$ has no discrete logarithm, so it is excluded from training and evaluation in all four intervention cells, whose inputs are the 690 nonzero residues. Spectra on a residue input are nevertheless computed over the full group $\mathbb{Z}/\ell$, because its characters require every point; the one entry the data do not define is filled by a fixed convention, the formula's own value $f(0) = 1$ for `res` $\to$ `res` and a fixed value for `res` $\to$ `dlog`, where $\log_g 0$ is undefined. No conclusion depends on the convention: changing one table entry moves each coefficient with $k \neq 0$ by at most $2/n$ and leaves the $k = 0$ column unchanged, so it moves ν by at most $2(q-1)/q < 2$, against target values of 20.9 to 26.3 in the failing cells and 1 in the affine cell.

Out-of-range predictions. The digit head can emit an integer outside the output range, for example a value of at least $\ell - 1$ in a discrete-log cell. Held-out accuracy is exact match on all answer digits against the presented label, so such an output counts as wrong. The decoded prediction functions of Appendix E reduce it modulo q before the transform, and fill the one input absent from the data with a value that forces agreement.

B.2 Propositions and proofs

Proposition 1 (Affine targets saturate; nonlinear targets are suppressed)**.** *For $f(r) = cr + d$ with $c \not\equiv 0$, $F[j,k] = \omega^{-kd}\, \mathbf{1}[j = kc]$, so a single character carries the whole target. For $f(r) = a + c\, r^e$ with $2 \leq e < \ell$ and $c \not\equiv 0$, every $F[j,k]$ with $k \not\equiv 0$ is a Weil sum of a degree-e polynomial (Weil, 1948), so $|F[j,k]| \leq (e-1)/\sqrt{\ell}$. Between $e = 1$ and $e = 2$ the best available correlation falls from 1 to exactly $\ell^{-1/2}$ (a quadratic Gauss sum), and to at most $(e-1)/\sqrt{\ell}$ beyond.*

Proposition 2 (Fourier complexity needed to fit)**.** *Let G be a finite abelian group, $H = \mathbb{Z}/q$, and $f, g : G \to H$. With $\varepsilon = \max_{\chi,\, \kappa \neq 0} |F[\chi,\kappa]|$ (written $F[j,k]$ in Eq. 1 when $G = \mathbb{Z}/\ell$)*

a property of the target and $\nu_g = (1/q)\sum_{\chi,\kappa} |G[\chi,\kappa]|$ *the Fourier* L^1 *norm of the hypothesis,* $\mathrm{agree}(f,g) \le 1/q + \varepsilon\,\nu_g$ *for* every g.

Proof of Proposition 1. For $f(r) = cr + d$, $F[j,k] = \ell^{-1}\sum_r \omega^{jr-k(cr+d)} = \omega^{-kd}\ell^{-1}\sum_r \omega^{(j-kc)r}$, and the inner sum is ℓ when $j = kc$ and 0 otherwise. For $f(r) = a + cr^e$ with $k \not\equiv 0$, the exponent $jr - k(a+cr^e)$ is a polynomial in r of degree e with a nonzero leading coefficient $-kc$, so Weil's bound (Weil, 1948) gives $|\sum_r \omega^{P(r)}| \le (e-1)\sqrt{\ell}$, hence $|F[j,k]| \le (e-1)/\sqrt{\ell}$. The bound is uniform in j, in ℓ after rescaling, and in the multiplier c.

Proof of Proposition 2. Write $\mathbf{1}[x=y] = q^{-1}\sum_\kappa \chi_\kappa(x-y)$ for $x, y \in \mathbb{Z}/q$. Then $\mathrm{agree}(f,g) = |G|^{-1}\sum_x \mathbf{1}[f(x)=g(x)] = q^{-1}\sum_\kappa \langle \chi_\kappa \circ f, \chi_\kappa \circ g\rangle$. The $\kappa = 0$ term contributes exactly $1/q$. For $\kappa \neq 0$, expanding $\chi_\kappa \circ g$ in characters of G and applying Parseval bounds each term by ε times the L^1 mass of the corresponding column of g's transform, and summing over κ gives $\varepsilon\nu_g$. No property of g beyond its Fourier L^1 norm is used, so the bound holds for every g, for every finite abelian G, and independently of how g was obtained.

A sharper form of Proposition 2. The bound tightens, with no change to the proof above, to $\mathrm{agree}(f,g) \le 1/q + \varepsilon(\nu_g - 1/q)$, because $\chi_0 \circ g \equiv 1$, so the zero-output-frequency column of any hypothesis transforms to a single unit spike and contributes exactly $1/q$ to ν_g; we verified this numerically across four moduli. Every statement in this paper uses the weaker printed form, so the requirement thresholds here and the downstream constants of §6 are the conservative ones and none of them depends on the sharpening.

The Parseval bridge. Propositions 1 and 2 are statements about ε; every grid in the paper reports ν. The following connects them, and is why a bound on the largest coefficient constrains the L^1 norm.

Proposition 3 (Parseval bridge)**.** *Each nonzero-*κ *column of* F *transforms a unit-modulus function, so* $\sum_j |F[j,\kappa]|^2 = 1$ *and* $1 \le \|F[:,\kappa]\|_\infty \|F[:,\kappa]\|_1$. *With Proposition 1 this gives, for* $f = a + c\,r^e$ *with* $2 \le e < \ell$, $\frac{1}{\ell}\big[1 + (\ell-1)\sqrt{\ell}/(e-1)\big] \le \nu \le \frac{1}{\ell}\big[1 + (\ell-1)\sqrt{\ell}\big]$, *while affine* $\nu = 1$.

At degree 2 the two sides coincide, because quadratic Gauss sums attain Weil's bound. Rearranged, ε also bounds the edge over chance available to a unit-complexity hypothesis: about three points on a large-image Hecke task against about 67 on a reducible one (uncentred ε, Appendix B.3). The bound need not be attained, so what ε licenses is an exclusion rather than a prediction.

Proof of Proposition 3. Each column $F[:,\kappa]$ with $\kappa \neq 0$ is the transform of the unit-modulus function $r \mapsto \omega^{-\kappa f(r)}$, so $\sum_j |F[j,\kappa]|^2 = 1$ by Parseval. Since $\|v\|_2^2 \le \|v\|_\infty \|v\|_1$, we get $1 \le \|F[:,\kappa]\|_\infty \|F[:,\kappa]\|_1$, and substituting the Weil bound on $\|F[:,\kappa]\|_\infty$ gives the lower bracket. The upper bracket follows from $\|v\|_1 \le \sqrt{\ell}\|v\|_2$. The $\kappa = 0$ column contributes exactly 1; as a code check, omitting it makes the bracket fail on 7 of 28 configurations tested.

Verification of Proposition 1. Over 45 configurations spanning three moduli (251, 389 and 691), five degrees and three multipliers (1, 2 and 345), there are **zero violations**. Degree 1 attains $\varepsilon = 1.0000$ in every configuration, the maximum a correlation can take. Degree ≥ 2 never exceeds 0.2071 on this set (on the seven-modulus set of 35 cells below, it reaches 0.374 at $\ell = 101$), and after rescaling by $\sqrt{\ell}$ the values lie in $[1.000, 3.880]$, so the suppression is of the predicted order rather than merely smaller than one.

ν and ε are different statistics. They are not comonotone in degree (Table 6), and must not be conflated; Proposition 3 is the line that connects them.

This is not a statistical-query hardness result. The suppression is polynomial, not exponential: a correlation of $\varepsilon \approx \ell^{-1/2}$ is resolvable from about $1/\varepsilon^2 \approx \ell$ samples, roughly 691 at our headline modulus, against approximately $14{,}000$ examples per run, though the $1/\varepsilon^2$ count is a noise-resolution heuristic rather than a sample-sufficiency theorem, and those $14{,}000$ examples cover at most 691 distinct input residues, roughly 553 in the training split, so they are not $14{,}000$ independent draws over $\mathbb{Z}/\ell$. Nothing here says the target is information theoretically hard, and the network does fit

Table 6: ν and ε are different statistics and are not comonotone in degree, at $\ell = 691$. Proposition 3 is what connects them.

degree	1	2	3	5	11
ν	1.00	**26.25**	22.26	**20.67**	20.89
ε	1.0000	0.0380	0.0752	0.1293	0.1296

every dense target at train accuracy 1.000. The claim is about which fits are available in the class the learner reaches, not about what is computable.

B.3 The edge over chance on measured eigenvalues

Proposition 2 at $\nu_g = 1$ bounds the accuracy available to a unit-complexity hypothesis, and the edge column equals the uncentred ε identically because bound $- 1/\ell = \varepsilon\nu_g$ (Table 7). On a large-image task no low-complexity hypothesis beats chance by more than about three points, against about 67 points on a reducible task, a factor of 24 that is derived rather than observed. Two cautions apply. First, this is an upper bound on the edge and not an attainable one: over a prime field the $\nu_g = 1$ hypotheses are exactly the affine maps, and for a quadratic target $f(r) = 1 + r^2$ the agreement of $g(r) = ar + b$ is the number of roots of $r^2 - ar + (1 - b) \equiv 0$, at most 2, so the best affine map at $\ell = 691$ agrees on $2/691$ of the domain, an edge of $1/\ell = 0.0014$ against a permitted $\varepsilon = 0.0380$, a factor of $\sqrt{\ell} = 26.3$; the best affine fit is exact, found by scanning all ℓ^2 affine maps rather than by optimization, so the comparison is to the true optimum in the unit-complexity class. Second, the measured ceiling exceeds the bound on 72 of the 130 tasks and the bound exceeds the ceiling on the other 58, so the aggregate means landing close is an artifact of averaging, not a validation. This bound does *not* constrain the measured ceiling of Table 4, which is the best *arbitrary* function of $p \bmod M$ and carries no complexity limit.

Table 7: Proposition 2 applied to measured Hecke eigenvalues, at $\nu_g = 1$. The edge column equals ε identically, because bound $- 1/\ell = \varepsilon\nu_g$; it is an upper bound on the edge, not an attainable one, since for a quadratic target the best affine map agrees on $2/\ell$ of the domain, an edge of $1/\ell$ over chance, against a permitted edge of $\ell^{-1/2}$. Tables 4 and 5 report the centred character correlation, the right quantity against a marginal-preserving permutation null; Proposition 2 is stated for the uncentred ε, which is what appears here, and the two differ most on the dihedral regime, at 0.5591 against 0.4044. This bound does *not* constrain the measured ceiling of Table 4, which is the best *arbitrary* function of $p \bmod M$ and has no complexity limit.

Regime	ε **(uncentred)**	**bound at** $\nu_g = 1$	$1/\ell$	**edge permitted at** $\nu_g = 1$
reducible	0.6677	0.8040	0.1363	$+0.6677$
dihedral	0.5591	0.6558	0.0968	$+0.5591$
large-image	0.0282	0.1059	0.0777	$+0.0282$

Relation to leap complexity. Leap complexity (Abbe et al., 2023) is a natural candidate for a second axis beside ν, and we state why it is not currently testable rather than list it as pending work. Leap complexity predicts *staircase* dynamics, while the graded ladder produces a smooth monotone fall with no threshold, so any second axis has to be compatible with a graded response on that family. It also has to be compatible with large gaps at matched ν. And leap is defined on the hypercube, where both of our substrates failed on identified design faults (Appendix F.2); the characteristic-2 transplant is what restored field generality and it is not a hypercube. The required order of work is to define leap for $\mathbb{Z}/\ell$ first and design an experiment second.

C Controlled presentation experiments

The 0.5 threshold. The rule fixed before the runs was held-out accuracy above 0.5. The base-model $\texttt{dlog} \to \texttt{dlog}$ cell reaches 0.459 at the matched 12k-step budget, so the radix-10 intervention scores 3 of 4 cell predictions; the threshold was not moved. The cell had not converged: one run reaches

0.193 at 2k steps, 0.432 at 4k and 0.509 at 12k, and a separate 40k-step run gives 0.542 against 0.000.

The shortcut audit. In radix 10 one of three answer digits is free, because $11 \equiv 1 \pmod{10}$ and $10 \mid 690$, so the last digit of $11x \bmod 690$ is the last digit of x. Changing the exponent from 11 to 13 does not remove this; changing the radix does. The full 2×2 was therefore rerun in radix 7 on both sides ($\gcd(7, 690) = 1$), eight seeds per cell, the same held-out residues in every cell. All five predictions hold: both mixed cells stay below 0.02 (0.003 and 0.000), the affine cell leads `res` $\to$ `res` by $+0.283$ (bootstrap 95% CI $[+0.224, +0.338]$) and is largest in every seed, composed scoring against $\tau(p) \bmod 691$ agrees with direct scoring to 0.01, and the affine cell does not clear 0.5 (0.284 ± 0.082). Every answer digit of the affine cell clears its modal rate by 0.32 to 0.60, so no digit is copied. The radix-7 column of Table 1 reports these runs; its learning curve rises 0.058, 0.101, 0.172, 0.171, 0.282, 0.284 at steps 2k to 12k while the three failing cells stay at or below 0.003 throughout.

A second theorem. To rule out a special case, the weight-16 congruence $a_p \equiv 1 + p^{15} \pmod{3617}$ was presented in the same four coordinate systems, three seeds, shared held-out residues, protocol identical to $\ell = 691$. Checks made before the runs: $\gcd(15, 3616) = 1$, so the discrete-logarithm target is a bijection; the residue presentation reproduces the form's own $a_p \bmod 3617$ on 17,983 primes, so `res` $\to$ `res` is the certified negative itself; no answer digit is free in radix 10 ($10 \nmid 3616$); and at $\ell = 3617$ there are about five primes per residue, so held-out residues are unseen. All five predictions hold (Table 1, last column): the negative reproduces at 0.000, both mixed cells fail at 0.000, the rescue gap is $+0.853$ (95% CI $[+0.824, +0.881]$), the affine cell is largest in every seed, and composed scoring against $a_p \bmod 3617$ agrees with direct scoring to within 0.001 (0.854 against 0.853; see the next paragraph). The rescue is cleaner than Ramanujan's, so the data-hunger of the $\ell = 691$ cell is a property of that cell, not of the intervention.

Information preservation. Predictions in each `dlog` $\to$ `dlog` cell were mapped back through $y \mapsto 1 + g^y \bmod \ell$ and scored against the congruence in the original residue coordinates. On every in-range prediction the bijection preserves correctness exactly, so the two scores can differ only on out-of-range digit strings, integers of at least $\ell - 1$ that the digit head can emit (Appendix B.1): direct exact-match scoring counts such an output wrong, while the composed map, which exponentiates, reduces it modulo $\ell - 1$ and so scores an output congruent to the correct logarithm as correct. The two scores coincide at $\ell = 691$ (0.459 and 0.284) and differ by 0.001 at $\ell = 3617$ (0.854 composed against 0.853 direct), a difference of scoring convention of this kind rather than of information. The coordinate change therefore loses no information. The network does not compute the discrete logarithm; the presentation supplies it, which is why the intervention is a diagnostic rather than a recipe.

The graded ladder. Table 8 gives every rung. One target function throughout, so degree, image size, label entropy and invertibility are constant, and π_k is a bijection, so information is unchanged; every presented map at a nonzero rung is a permutation of $\mathbb{F}_{691}$, and its degree, computed from the reduced interpolant, is 689 at every rung and in every matched-arm and piece-count configuration. Three permutations per rung; Spearman between ν and accuracy -0.983 across all rungs and permutations, mean within-rung standard deviation 0.015. The lower block splits each rung by whether π_k moves the input residue: the network sees digits of $s = \pi_k(r)$ and must answer $1 + r$, so "answer $1 + s$" is right exactly on the fixed points, a baseline computed from the realised permutation. Accuracy on moved inputs is 0.000 at every rung and overall accuracy is at or below the fixed-point baseline from $k = 5$ on; the required separation of at least 0.30 between fixed and moved inputs holds up to $k = 200$ and fails at $k = 500$ and 691, where no accuracy is left to separate.

Along this ladder ν and ε move in opposite directions, a limitation we noted before the runs: the ladder tests the spectral account as a whole and cannot separate the two.

Matched relabelling. The ladder cannot separate the presented spectrum from the surviving affine fraction, because both are functions of k and every correct prediction falls on the fixed residues. The design that separates them holds the moved set fixed, and with it the fixed points and the best-affine agreement, and varies only how the moved residues are permuted among themselves. Target $r \mapsto 1 + r$ at $\ell = 691$; the moved set is a contiguous block of k residues at a seed-chosen offset,

Table 8: The graded scrambling ladder, decomposed. One target function throughout, so the target's degree, image size, label entropy and invertibility are constant by construction, and every presented map at a nonzero rung has algebraic degree 689. Accuracy falls smoothly and monotonically with the presented ν across a factor of 23. Three independent permutations per rung. Fixed points are realised counts, averaged over the three permutations; at $k = 691$ they are 1, 1 and 0, none of them held out. The lower block splits each rung by whether π_k moves the input residue: the network scores exactly 0.000 on every moved residue at every rung, and never reaches the fixed-point baseline of ignoring the permutation.

k **permuted**	0	5	20	70	200	500	691
ν (presented)	**1.00**	3.69	6.53	11.02	17.05	22.59	**23.25**
ε	1.0000	0.9997	0.9844	0.9245	0.7499	0.3377	0.1406
test accuracy	**0.992**	0.969	0.931	0.723	0.372	0.019	**0.000**
sd across π	0.000	0.005	0.029	0.044	0.018	0.009	0.000
fixed points of π_k	691	686	671	622	492	192	0.7
fixed-point baseline	1.000	0.991	0.977	0.899	0.676	0.301	0.000
accuracy on fixed residues	0.992	0.978	0.953	0.805	0.554	0.062	–
accuracy on moved residues	–	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

identical for both arms. The dense arm applies a random derangement of the block; the sparse arm applies a cyclic shift, so the presented map is two affine pieces plus one exception. $k \in \{70, 200\}$, three permutation seeds, two training seeds, shared held-out residues. The sparse arm learns the moved residues and the dense arm does not, 0.919 and 0.977 against 0.017 and 0.006, positive in six of six (k, seed) pairs, with $\nu(\text{shift}) \leq \frac{1}{2}\,\nu(\text{derange})$ at identical fixed-point sets (Table 9). The target whole-domain gap of at least 0.15 at both k was not met at $k = 70$ ($+0.108$; $+0.317$ at $k = 200$).

Table 9: Same fixed points, same best-affine agreement, different spectrum. Both arms at each (k, seed) move exactly the same residues; $n = 6$ per cell.

k	arm	fixed pts	ν	best-affine agr.	test acc	baseline	acc | fixed	acc | moved
70	derange (dense)	621	9.44	0.899	0.871 ± 0.013	0.895	0.971	$\mathbf{0.017}$
70	shift (sparse)	621	4.58	0.899	0.979 ± 0.011	0.895	0.985	$\mathbf{0.919}$
200	derange (dense)	491	14.43	0.711	0.656 ± 0.014	0.734	0.892	$\mathbf{0.006}$
200	shift (sparse)	491	4.74	0.711	0.972 ± 0.016	0.734	0.971	$\mathbf{0.977}$

Piece count. To locate the transition, the same 200-residue block is split into m sub-blocks with sub-block j rotated by $j + 1$, so the block presents m distinct affine pieces at the same fixed points and the same best-affine agreement; $m = 1$ is the cyclic shift. $m \in \{1, 2, 4, 8, 12\}$, two permutation seeds and two training seeds; the range stops at twelve because rotations are distinct only modulo the sub-block length. All four predictions hold: accuracy on the moved residues falls monotonically, Spearman -0.975 across runs, from 0.987 at $m = 1$ to 0.077 at $m = 12$, with the 0.5 crossing between $m = 4$ and $m = 8$; ν rises at every step; and the fixed residues stay above 0.90 at every m (Table 10). Because ν and m rise together by construction, this family does not separate them.

Table 10: Where the transition sits. Same moved block, same fixed points, same best-affine agreement at every row; only the number of affine pieces on the block changes. Four runs per row. $m = 1$ is the `block_shift` arm of Table 9 restricted to its first two permutation seeds.

pieces m	sub-block	ν (presented)	best-affine agr.	test acc	acc | fixed	acc | moved
1	200	4.74	0.711	0.972	0.967	$\mathbf{0.987}$
2	100	5.62	0.711	0.934	0.956	$\mathbf{0.871}$
4	50	6.47	0.711	0.905	0.970	$\mathbf{0.722}$
8	25	7.20	0.711	0.764	0.949	$\mathbf{0.238}$
12	16–17	8.29	0.711	0.692	0.909	$\mathbf{0.077}$

Fixed piece count, different ν. To test whether Fourier complexity carries information beyond the number of affine pieces, we constructed twelve matched pairs of targets before any run. Each pair

has the same piece count (nine or eleven), moved residues, fixed points, sub-block length, best-affine agreement (0.7106), algebraic degree (689) and jump magnitude, with mean displacement matched to within one sub-block, and varies ν by 0.89 to 1.25; 24 tasks, three seeds each. Displacement is thus matched in tolerance, not exactly, which is why the eleven-piece rank correlations below can favour it. The lower-ν member achieved higher accuracy on the moved residues in 10 of the 12 pairs (one-sided sign test $p = 0.019$; two-sided $p = 0.039$). The mean advantage was $+0.052$ (95% CI $[+0.022, +0.084]$), below the target margin of 0.10. The advantage was concentrated at nine pieces, $+0.092$ in six of six pairs, and inconclusive at eleven, $+0.012$ with an interval covering zero, where within-piece-count rank correlations favour jump magnitude and mean displacement over ν (-0.517 and $+0.470$ against -0.196; at nine pieces ν leads at -0.717). Train accuracy was at least 0.995 and accuracy on the fixed residues at least 0.950 in every task. These results support a modest, family-dependent association between Fourier complexity and generalization at fixed piece count (Table 11).

Table 11: Matched pairs at fixed piece count with differing ν: accuracy on the moved residues, three seeds per task, with the ν range of each arm in parentheses. Held equal within each pair: piece count, moved residues, fixed points, sub-block length, best-affine agreement (0.7106), algebraic degree (689) and jump magnitude; mean displacement is matched only to within one sub-block. Differences are lower minus higher ν, with 95% bootstrap intervals over pairs; the pooled row gives the sign test (one-sided $p = 0.019$) and the mean advantage against its target of 0.10.

pieces	lower ν	higher ν	difference	95% CI	positive
9	0.931 (6.00–6.47)	0.840 (7.25–7.36)	$+0.092$	$[+0.058, +0.127]$	6 / 6
11	0.866 (6.75–6.85)	0.854 (7.76–7.86)	$+0.012$	$[-0.013, +0.034]$	4 / 6
all 12 pairs			$\mathbf{+0.052}$	$[+0.022, +0.084]$	$\mathbf{10/12}$

D PREDICTING GENERALIZATION FROM TASK STRUCTURE

Table 15 gives all eleven statistics. Every statistic is computed from the target's label table alone, with directions frozen on a development split of four cells at $\ell = 691$ and scored on 38 held-out cells at other moduli. Metrics are rank-based, so a common monotone rescaling of a statistic leaves its scores unchanged; an ℓ-dependent normalization is not of that kind, because the scored cells span several moduli, so the normalizations of Table 13 are part of each definition. Scores are reported for each statistic in its frozen direction, so a positive Spearman value means the statistic orders accuracy as predicted. Both AUROC and Spearman use average ranks, so tied values are not separated by the order they happen to arrive in. That convention turns out to matter only for the rivals. Moving from the tie-broken to the tie-aware scoring leaves the held-out AUROC of all five leading spectral entries unchanged to three decimal places, at 0.997, 0.988, 0.986, 0.977 and 0.957, and moves their Spearman values by at most 0.004; every discrete-valued or near-binary rival moves, algebraic degree by 0.093 and Fourier L^0 by 0.032 in AUROC, and on the untouched families of Table 14 by as much as 0.212. **The predictors this paper relies on are invariant to the rank-tie convention; the statistics they are compared against are not.** Two of the eleven, spectral Gini and Fourier L^0, are Fourier statistics that land in the middle of the table, and they sharpen the claim: what predicts is a summary sensitive to how the spectral mass is distributed, not the Fourier label as such.

Why there is no maximum-coefficient entry. A maximum taken over the entire transform including $F[0, 0]$ is constant at 1 for every deterministic map, since $F[0, 0] = 1$ and no other coefficient exceeds it; ε in Eq. 1 is the maximum over $k \neq 0$, which is why it carries information and that quantity does not.

The formulas. Table 13 gives every predictor in closed form, since a statistic named in prose is not a definition. Throughout, F is the transform of Eq. 1 for the presented target f. The spectral summaries are computed on the nonzero output frequencies only: by Parseval each column $F[:, \kappa]$ with $\kappa \neq 0$ satisfies $\sum_j |F[j, \kappa]|^2 = 1$, so

$$p[j, k] \;=\; \frac{|F[j, k]|^2}{\ell - 1}, \qquad k \neq 0, \tag{3}$$

Table 12: Frozen predictors on the two controlled families, Spearman against accuracy on the moved residues. Whole-domain accuracy on these families is dominated by the untouched fixed points, so every run clears the acc > 0.5 criterion and AUROC is undefined. Accuracy on moved residues spans 0.000 to 1.000 on `matched` and 0.059 to 1.000 on `piece_count`. Post-hoc: formulas and directions frozen, nothing refitted or retrained. Spearman values use each statistic's frozen direction, so a positive value means it orders accuracy as predicted; for ν the score is $-\nu$.

predictor	**`matched`** $(n = 24)$	**`piece_count`** $(n = 20)$
spectral entropy	$\mathbf{+0.755}$	$\mathbf{+0.972}$
ν	$\mathbf{+0.743}$	$\mathbf{+0.966}$
ε	$+0.719$	$+0.104$
inverse participation / eff. support	$+0.323$ / $+0.312$	$+0.965$ / $+0.975$
best-affine agreement	$\mathbf{-0.094}$	constant by design
algebraic degree	constant (689)	constant (689)

is a probability distribution over the $(\ell - 1)\ell$ nonzero-output coefficients, and every concentration summary below is a functional of it. Two statistics run over the whole spectrum instead: ν, exactly as Eq. 1 writes it, and effective support, which is a Rényi-2 participation ratio and not an exponentiated entropy. The zero-output column contributes exactly $1/q$ to ν, and failing to exclude it is what makes the maximum-coefficient quantity above constant. Algebraic degree is computed, not read from the task specification: the coefficient of $x^{\ell-1-e}$ in the reduced interpolant is $-\sum_r f(r)\, r^e$, so the degree is $\ell - 1 - e$ for the least e whose power sum is nonzero modulo ℓ. Directions are frozen on the four-cell development split, so a statistic that predicts in either orientation is not credited twice.

Two of the eleven are not independent. Effective support as implemented is a participation ratio over the whole spectrum, including the zero output frequency. Restricted to $k \neq 0$, as every other concentration statistic here is, it is exactly the reciprocal of inverse participation, so **those two rows are not independent evidence**; as implemented they differ only through the $F[0, 0]$ term, which is 1 for every deterministic map and carries no information about f. We state this rather than leave it to be found, and the leading group in Table 15 should be read accordingly: it is four distinct spectral quantities, one of which appears twice up to inversion.

Table 13: Every predictor in closed form, each verified against the stored values to twelve decimal places. p is the distribution of Eq. 3 over the nonzero-output coefficients and q_y is the fraction of inputs f maps to y. Entropies use natural logarithms and are normalised to $[0, 1]$ by the count of coefficients they range over. Effective support and image size carry ℓ-dependent normalisations; because the scored cells span several moduli, such a normalisation can change rankings across cells, so the definitions given here are the ones scored. Note the two domains: ν and effective support run over the whole spectrum, every other spectral statistic over $k \neq 0$.

predictor	**definition**	**over**
ε	$\max_j \max_{k\neq 0} \lvert F[j,k]\rvert$	$k \neq 0$
ν (Fourier L^1)	$\frac{1}{\ell}\sum_{j,k} \lvert F[j,k]\rvert$	all k
spectral entropy	$-\sum_{j,k\neq 0} p[j,k] \ln p[j,k] \,/\, \ln\big(\ell(\ell-1)\big)$	$k \neq 0$
effective support	$\ell \,/\, \sum_{j,k} \lvert F[j,k]\rvert^4$	**all** k
inverse participation	$\sum_{j,k\neq 0} p[j,k]^2$, direction $+1$	$k \neq 0$
spectral Gini	Gini coefficient of the magnitudes $\lvert F[j,k]\rvert$	$k \neq 0$
Fourier L^0	$\#\{(j,k) : \lvert F[j,k]\rvert > \tau\} \,/\, \ell(\ell-1), \tau = 10^{-9}$	$k \neq 0$
label entropy	$-\sum_y q_y \ln q_y \,/\, \ln \ell$	labels
image size	$\lvert f(\mathbb{Z}/\ell)\rvert \,/\, \ell$	labels
invertibility	$\mathbf{1}[\, f$ is a bijection $]$	labels
algebraic degree	$\ell - 1 - e$, e least with $\sum_r f(r)\, r^e \not\equiv 0$	labels
`best_affine_agreement`	$\max_{a,b}$ agree$(f,\ r \mapsto ar + b)$, all ℓ^2 scanned	labels
`affine_edge_over_eps`	(best affine agreement $- 1/\ell) \,/\, \varepsilon$	both

Untouched families, and the best-affine rival. The eleven formulas, their stored directions and the learnability rule acc > 0.5 were taken verbatim from the tournament, and two predictors were added with directions fixed a priori: `best_affine_agreement`, the exact agreement the best of all ℓ^2 affine maps reaches with the presented target, and `affine_edge_over_eps`, that agreement's

edge over chance divided by ε. All were scored, with no training, on two families the tournament never saw: `twoterm`, $r \mapsto r^a + r^b \bmod \ell$ at $\ell = 389$ and 577 (10 cells), and `scrambled`, the ladder's 21 presented maps $s \mapsto 1 + \pi_k^{-1}(s)$. Both families are near-binary in ν, so AUROC saturates and the Spearman column carries the information (Table 14). Five predictions were specified for this test. The frozen predictors transfer (confirmed); best-affine agreement reaches AUROC at least 0.85 (confirmed, 1.000); a spectral summary beats best-affine agreement by at least 0.05 (**refuted**, 1.000 against 1.000), which is the informative outcome, since the statistic that ties the spectral summaries and leads them on the continuous measure needs no spectral machinery to compute; the bootstrap lower bound exceeds 0.80 (confirmed); and algebraic degree stays at chance (confirmed, 0.527 against a bar of 0.65). Per family, spectral entropy, ν and best-affine agreement each reach 1.000 on both `scrambled` ($n = 21$) and `twoterm` ($n = 10$), degree reaches 0.625 and 1.000 respectively, and leave-one-family-out gives 1.000 for all three in both directions. On `scrambled`, every correct prediction falls on the surviving affine domain (§4), so best-affine agreement predicting it is close to tautological there.

Table 14: Frozen predictors on two untouched families, scored on a frozen source list of 31 cells: the 21 `scrambled` cells and the 10 `twoterm` cells. The population is fixed at these 31 cells, so families added later do not change it. AUROC against acc > 0.5; Spearman against continuous accuracy; both use average ranks. Bootstrap 95% intervals on AUROC where computed. An interval of $[1.000, 1.000]$ records perfect separation on these 31 cells under resampling, not certainty about transfer to a new family; with two families and a near-binary ν, AUROC saturates and the Spearman column carries the information. Both columns use each predictor's frozen direction (for ν the score is $-\nu$), so no predictor scores below chance in that direction.

predictor	**AUROC**	**Spearman**	95% **CI (AUROC)**
`best_affine_agreement` (a priori)	1.000	$+\mathbf{0.975}$	$[1.000, 1.000]$
`affine_edge_over_eps` (a priori)	1.000	$+\mathbf{0.973}$	
ε	1.000	$+0.957$	
effective support	1.000	$+0.951$	
inverse participation ratio	1.000	$+0.939$	
spectral entropy	1.000	$+0.914$	$[1.000, 1.000]$
ν	1.000	$+0.874$	$[1.000, 1.000]$
spectral Gini	0.920	$+0.711$	
Fourier L^0	0.716	$+0.477$	
image size	0.647	$+0.415$	
invertibility	0.647	$+0.405$	
label entropy	0.607	$+0.211$	
algebraic degree	0.527	$+0.148$	$[0.325, 0.725]$

D.1 The dilated challenge set

To test whether ν adds information where the simple structural quantities cannot separate the tasks, the four block arms of the matched and piece-count families (shift, four pieces, eight pieces and derangement, all on the same 200-residue block) were conjugated by multiplication by $d = 7$ and $d = 10$ modulo 691. Conjugation maps the relabelled block onto an arithmetic progression of stride d. It leaves ν, the other spectral summaries, best-affine agreement (0.7106), degree and the number of affine rules unchanged, so the sixteen new tasks share these with their undilated originals; quantities that depend on the ordering of residues, such as local continuity and mean displacement, do change. Predictions were specified before training. ν orders accuracy on the relabelled residues within each dilation, Spearman -0.894 at $d = 7$ and -0.958 at $d = 10$ against a target of -0.70, and the order is the same at every dilation: shift, then four pieces, eight pieces, derangement. Over the original and dilated tasks together ν tracks accuracy at -0.865, against -0.599 for local continuity and -0.822 for mean displacement; the number of affine rules ties with ν because the design cannot separate them, and every run fits its training set. The measured summaries do not determine the level of accuracy: at identical values of them, dilation by 7 lowers accuracy on the relabelled residues by up to 0.70 (four pieces, 0.72 to 0.03) and reaches the fixed residues as well, while dilation by 10 costs at most 0.27 and beats dilation by 7 in every comparable pair. The alignment of the relabelled set with the digits of the encoding is a plausible explanation of that difference, proposed after the results and not tested here.

Table 15: Predictor tournament, in full. Every statistic is computed from the target's label table alone, directions frozen on a four-cell development split at $\ell = 691$, scored on 38 held-out cells, with average ranks in both columns; both columns use each statistic's frozen direction (for ν the score is $-\nu$). The five leading Fourier summaries cluster at the top, two of them the same quantity up to inversion; two further Fourier statistics that summarise support or inequality rather than mass land in the middle; algebraic degree trails the field here and is at chance on the untouched families of Table 14.

Predictor	**Held-out AUROC**	**Spearman**
spectral entropy	**0.997**	+0.861
ν (Fourier L^1)	0.988	+0.786
effective support	0.986	+0.900
ε	0.977	+0.873
inverse participation	0.957	+0.843
label entropy	0.857	+0.629
spectral Gini	0.855	+0.451
image size	0.852	+0.637
Fourier L^0	0.687	+0.416
invertibility	0.646	+0.315
algebraic degree	**0.606**	+0.199

D.2 The near-boundary regime

At matched $\nu \approx 7$, piecewise-affine maps fail at 0.009–0.025 while perturbed-affine maps learn at 0.55–0.78, a seed-confirmed gap of roughly $80\times$; the two families part the same way on the tournament cells (Figure 4). Three candidate refinements were tested and eliminated: memorizability, algebraic degree, and signal fraction. The last of these is the most natural story, that the network memorizes the exceptions and generalizes the affine base, and it does not survive.

For piecewise maps, ν and rule count are the same axis by construction and cannot be separated within that family. Within the perturbed-affine family, ν and the corruption count are deterministically linked, so that family cannot separate them either. The honest summary is that ν in the reachable basis predicts learnability at the extremes, out of sample and under intervention, and that within these two families six candidate refinements, three per regime, do not determine the transition point. On the one family where every other variable is held fixed, the piece-count family of §4, the transition is measured: under the 0.5 criterion between four and eight affine pieces on a 200-residue block, ν between 6.5 and 7.2, with a graded fall on either side, the same shape found here at matched $\nu \approx 7$.

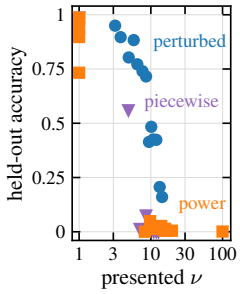


Figure 4: The tournament cells against ν alone. Accuracy falls with ν within each family; between $\nu \approx 7$ and 12 perturbed-affine maps stay far above the others, though only two clear the 0.5 criterion.

E Model analysis and regularization

Proposition 2 constrains a hypothesis of given Fourier complexity, and nothing in this paper connects Fourier complexity to parameter count. Rather than assume that step, we measure the quantity the proposition is actually about, the Fourier L^1 norm of the *prediction function* g each trained network

implements, decoded over the whole domain from existing checkpoints. The read-out analyses of §6 take spectra of activations, which is a different object; the decoded ν_g reported here, and summarized in §6, measures the prediction function itself, the quantity the proposition is about. No new training was required. Every task was regenerated and verified against the run's own train accuracy before measurement, and every checkpoint reproduces.

What the arithmetic already forces. Training covers 0.8 of the domain and train accuracy is 1.000 everywhere, so every model agrees with its target on at least 0.8 of the full domain whatever it does on held-out inputs. Proposition 2 then forces $\nu_g \geq (0.8 - 1/691)/\varepsilon$: 21.0 for the degree-2 target ($\varepsilon = 0.0380$) and 6.16 for the degree-11 residue target ($\varepsilon = 0.1296$); the "required at 50%" column of Table 16 is the weaker bound at agreement 0.5. No model in this paper can have $\nu_g \approx 1$ on a dense target while fitting its training set.

Table 16: The Fourier complexity of the function each network implements, decoded over the whole domain from existing checkpoints. "Required at 50%" is $(0.5 - 1/q)/\varepsilon$ for that cell's target. Proposition 2 holds in every cell, with no violations. Every failing cell sits above the requirement, so the bound does not exclude the memorizing solution; the rescued cell is sparser than every failing cell with non-overlapping ranges across seeds (supply rows: original sweep, two seeds; one at $K = 40$).

cell	n	test acc	agreement	ν_g decoded	required	ν target
`dlog` → `dlog` (rescued)	10	0.474	0.895	**10.93**	0.50	1.00
`dlog` → `res`	3	0.000	0.800	**24.22**	13.09	26.21
`res` → `dlog`	3	0.000	0.800	**24.23**	12.63	26.22
`res` → `res` (certified negative)	3	0.003	0.801	**22.59**	3.85	20.89
supply, $K = 0$	2	0.007	0.802	**24.25**	13.11	26.25
supply, $K = 1$	2	0.012	0.802	**24.26**	13.11	26.25
supply, $K = 5$	2	0.146	0.829	**24.45**	13.11	26.25
supply, $K = 10$	2	0.517	0.904	**25.11**	13.11	26.25
supply, $K = 20$	2	0.780	0.955	**25.68**	13.11	26.25
supply, $K = 40$	1	0.785	0.957	**25.70**	13.11	26.25
supply, scramble $K = 20$	2	0.003	0.801	**24.25**	13.11	26.25

The decoded measurement. Table 16 gives the result for the four intervention cells and the supply arm. Every failing cell has decoded ν_g above its 50% requirement (minimum ratio $1.85\times$), Proposition 2 holds in every checkpoint, and decoded ν_g rises with the number of supplied characters (Spearman $+0.938$, from 24.25 at $K = 0$ to 25.70 at $K = 40$, approaching the target's 26.25 as accuracy rises). The prediction that the rescued cell sits at least $5\times$ below the failing cells was not met: the ratio is $2.20\times$, in the predicted direction. The reading that matters for §6 is the fifth column: every failing cell's own solution carries ν_g between 22.6 and 24.3, above the 13.11 the bound requires for 50%, so the bound is satisfied by the memorizing solution and excludes only a sparse hypothesis generalizing on a dense target; which permitted dense hypothesis the learner selects is a matter of training dynamics that the bound does not decide.

How the decoded map is scored. Decoded predictions are reduced modulo the output size, because the compositional head can emit an out-of-range integer (invalid-output rates 0.00% to 0.617% across the four intervention checkpoints, with agreement identical to five decimals with and without the reduction). The at most one coordinate absent from the data is filled with a value that forces agreement, adding at most $1/691$; that is why the `res` → `res` row reads 0.801 where agreement over the 690 covered inputs is 0.800. For a map between groups of different size, ν follows Eq. 2 (Appendix B.1).

What weight decay does along the trajectory. Decoding ν_g at intervals through training, on the same two cells as §6.1, separates the two weight-decay settings sharply (Figure 3b,c). With weight decay the decoded complexity falls by 3.45 on average between step 200 and step 12,000 (2.83 to 4.04 across seeds), the parameter norm falls by 8.3, and held-out accuracy rises by 0.428; the three weight-decay seeds pass 0.103 at steps 1,200, 600 and 800. Without weight decay the complexity changes by only 0.30 and stays near 14, the norm *grows* by 13.6, and accuracy moves 0.039. This describes what weight decay does; it is not a leading indicator, and we checked. Scored by the step at

which each quantity reaches a quarter and then a half of its total change, the complexity fall and the accuracy rise are *concurrent* in all three seeds, so the fall does not precede the crossing and cannot predict when weight decay will work. The fall is also slow and continuous (in seed 0, 14.58 to 10.54 across the run) rather than a collapse inside the crossing window.

E.1 Ablation, reconstruction and supplied features

Representation sparsity. Before any ablation, the Gini coefficient of the read-out spectrum, as a multiple of a random-matrix null at the logits and at the final residual stream, separates the rescued cell from the three failing ones: $\texttt{dlog} \to \texttt{dlog}$ sits at $3.70\times$ and $17.9\times$ the null, the failing cells at 2.56 to $2.99\times$ and 14.7 to $15.1\times$. The prediction that the failing cells sit within $1.25\times$ of the null was not met; without LayerNorm the stream keeps the structured digit encoding, so the null is the wrong baseline, and only the comparative statement carries.

Ablation, count-matched and energy-matched. Across ten independently trained $\texttt{dlog} \to \texttt{dlog}$ checkpoints (baseline 0.474 ± 0.051), we filter the read-out stream's Fourier transform over the presented coordinate and push the result through the model's own unembedding. Removing the top 40 of 690 components gives 0.000 ± 0.000; removing 40 random components gives 0.450 ± 0.067, and random removal never hurts at any k; keeping only the top 40, 5.8% of the spectrum carrying 93.2% of the energy, recovers 0.495. Both arms have the correct sign in 10 of 10 seeds. Count matching leaves one inference open, that removing much more energy simply causes much more damage, and the control that settles it is not available inside the Fourier basis (no matched set of other frequencies exists when 40 bins carry 93%) but is available outside it: by Eckart–Young no rank-r projection removes more energy than the leading r singular directions, so an SVD ablation is the strongest energy-removal control. Removing a matched 93.6% of the energy needs about ten singular directions (5 to 13 across checkpoints) and collapses the model as fully, to 0.048 ± 0.046 (Table 19); the prediction that the energy-matched removal would leave accuracy at least 0.10 above the Fourier removal was not met. The necessity result is therefore that the representation carries the function, not that its Fourier structure specifically does. The keep arms, 0.495 for 40 Fourier bins against 0.178 for the ten-dimensional SVD subspace, compare forty real dimensions with ten; at matched rank the two subspaces reconstruct comparably (Figure 5a), and the direct-label baseline below shows what a Fourier reconstruction recovers.

Rank-matched reconstruction. Bins are kept in conjugate pairs and every arm is reported by real dimension $D \in \{8, \dots, 120\}$ on the same ten checkpoints: keep-Fourier(D) and keep-SVD(D) on all-input activations; a train-only Fourier arm, whose frequencies are chosen from the training-masked spectrum and whose coefficients are fitted on training coordinates only, then used to predict the held-out activations it never sees; and a train-fitted SVD arm, whose subspace is fitted on training rows and applied to the held-out rows' own activations, an asymmetry in SVD's favour. At matched D the Fourier and SVD subspaces reconstruct comparably (SVD leads at $D = 24$, 0.472 against 0.400; the predicted Fourier lead of 0.10 at $D = 40$ was not met, 0.498 against 0.475). The train-only Fourier reconstruction reaches 0.528 at $D = 40$ and 0.627 at $D = 120$, above the unfiltered model from $D = 40$ and above the train-fitted SVD control at every $D \geq 40$ (Figure 5). The realised dimension can exceed the requested D by one conjugate pair at small D; the rank-matched comparison is unaffected, and the $D = 120$ figures are exact.

The direct-label baseline. A baseline fixed before the run fits the same train-only Fourier regression to the one-hot training labels, with no network, and asks whether the read-out reconstruction beats it. It does not: the label fit reaches 0.975, above the reconstruction by 0.348 (95% CI $[0.271, 0.446]$) in 10 of 10 checkpoints, and 0.976 at two layers with and without LayerNorm (Table 17). On the read-out's own frequencies the reconstruction beats the label fit by $+0.156$ in 9 of 10, which the decision rule reads as Fourier interpolation of the target rather than structure the network learned. The same label-only fit on the four cells of the 2×2 gives 0.979 on $\texttt{dlog} \to \texttt{dlog}$ and at most 0.021 on the other three.

The matched non-Fourier control. The Fourier reconstruction regresses the presented coordinate onto activations while the SVD arms project activations the model produced, so four further arms perform the same regression with a different basis spanning the design matrix, fitted by least squares

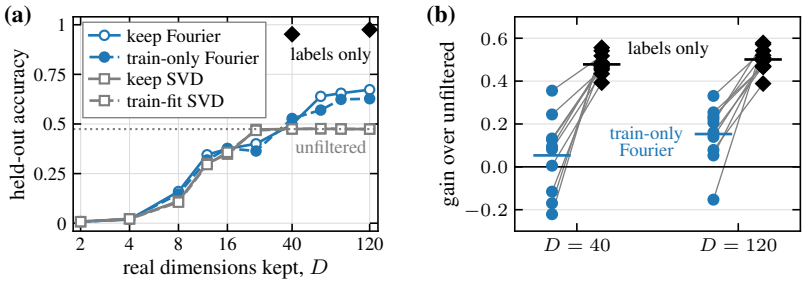


Figure 5: Reconstructing the read-out of ten `dlog` → `dlog` checkpoints from D real dimensions (§6). (a) Fourier and SVD subspaces of equal rank reconstruct comparably. The train-only Fourier fit exceeds the unfiltered model from $D = 40$, and the same regression on the training labels alone, with no network, does far better. (b) The label-only fit wins in ten of ten checkpoints at both D; the primary comparison is at $D = 120$.

Table 17: The direct-label baseline at $D = 120$. Held-out accuracy of the trained model, of the train-only Fourier reconstruction of its read-out, and of the same regression fitted to the one-hot training labels with no network, on the frequencies the read-out selected and on the labels' own frequencies. The prediction that the reconstruction beats the label-only fit is refuted: -0.348, 95% CI $[-0.446, -0.271]$, 0 of 10 checkpoints; the gain is Fourier interpolation of the target.

	1 layer, no LN ($n = 10$)	**2 layers, no LN** ($n = 5$)	**2 layers + LN** ($n = 3$)
trained model	0.474	0.284	0.624
read-out reconstruction	0.627	0.613	0.974
labels, on the read-out's frequencies	0.471	0.568	0.978
labels, own frequencies, no network	**0.975**	**0.976**	**0.976**

on training coordinates and pushed through the model's own unembedding. At $D = 40$ the Fourier arm reaches 0.528 and no other basis exceeds 0.001 (Table 18); an RBF kernel ridge tuned on the held-out metric itself reaches 0.015 and a small MLP on the same coordinates 0.043. The Fourier margin grows with dimension, $+0.627$ at $D = 120$ against $+0.527$ at $D = 40$. The reason is visible in the target: coordinates x and $x + 1$ carry answers $11x$ and $11x + 11$, which share no digits, so every basis that extrapolates by smoothness has nothing to extrapolate along.

Table 18: The matched non-Fourier control. Ten `dlog` → `dlog` checkpoints, unfiltered baseline 0.474. Every arm is the same operation, differing only in the basis spanning the design matrix.

real dims D	**Fourier**	**RBF bumps**	**spline**	**Chebyshev**	**random feat.**
8	**0.147**	0.000	0.000	0.000	0.002
40	**0.528**	0.000	0.000	0.001	0.001
120	**0.627**	0.000	0.000	0.000	0.001

Supplying a basis to a failing cell. The degree-2 power map at $\ell = 691$ never generalizes at any training fraction. We hand it K extra input tokens that carry no information beyond r: each character token gives the phase of one input character $\chi_j(r)$ in ten buckets, and each power token gives a bucket of $r^e \bmod \ell$ for an exponent e coprime to $\ell - 1$, a bijective algebraic relabelling that is structured but spectrally dense. Constancy given $p \bmod 691$ is verified for every token. $K \in \{5, 20\}$, six runs per arm, with three independent feature draws by two training seeds for the random and scrambled arms. At $K = 20$ the character arms reach 0.785 and 0.766, the power tokens 0.777, and twenty scrambled tokens 0.004 on three independent draws (Table 20). The prediction that the power tokens would fall at least 0.30 below the character arms was not met: a spectrally dense structured basis rescues the cell as well as the character basis does, so the character basis is one effective structured basis and not the only one. Top-energy characters did not beat random ones, and cannot on this target, whose per-frequency weight is constant to 10^{-14} across all 690 frequencies. The original K sweep on the top-K arm, 0.007, 0.012, 0.146, 0.517, 0.780, 0.785 at $K = 0, 1, 5, 10, 20, 40$, was unaffected by the correction note of Appendix A.1.

Table 19: The Fourier ablation against an energy-matched non-Fourier one, on the same ten $\texttt{dlog} \to \texttt{dlog}$ checkpoints; baseline 0.474 ± 0.051. Both arms project the read-out stream onto a subspace and push the result through the model's own unembedding, differing only in which subspace. Energy is the fraction of Frobenius norm removed. Necessity is not Fourier-specific at this energy fraction. The keep arms compare forty real dimensions with ten; Figure 5 makes the comparison at matched rank. The random-40 arm is a second independent draw of 40 random components.

arm	dims	energy removed	held-out accuracy
remove top-40 Fourier	40	0.932	$\mathbf{0.000 \pm 0.000}$
remove energy-matched SVD	**10**	0.936	$\mathbf{0.048 \pm 0.046}$
remove top-40 SVD	40	1.000	0.002 ± 0.004
remove random-40 Fourier	40	0.040	0.455 ± 0.058
keep top-40 Fourier	40	–	$\mathbf{0.495 \pm 0.111}$
keep energy-matched SVD	**10**	–	$\mathbf{0.178 \pm 0.072}$

Table 20: Supplying an information-free basis to the degree-2 power map at $\ell = 691$, a synthetic failing cell distinct from the certified negative of §3. Every token is a deterministic function of r alone. Six runs per cell; for the random and scramble arms, three independent feature draws by two training seeds.

K	arm	structured	Fourier	test acc (mean $\pm$ sd)	sd across draws
5	`top` (target's leading characters)	yes	yes	0.096 ± 0.109	–
5	`random` (uniform characters)	yes	yes	0.203 ± 0.158	0.147
5	`power` (buckets of r^e)	yes	**no**	$\mathbf{0.343 \pm 0.232}$	–
5	`scramble` (random relabellings)	**no**	no	$\mathbf{0.000 \pm 0.000}$	0.000
20	`top`	yes	yes	$\mathbf{0.785 \pm 0.036}$	–
20	`random`	yes	yes	0.766 ± 0.029	0.019
20	`power`	yes	**no**	$\mathbf{0.777 \pm 0.040}$	–
20	`scramble`	**no**	no	$\mathbf{0.004 \pm 0.004}$	0.000

E.2 Weight decay and optimizer diagnostics

The weight-decay sweep. Table 21 and Figure 3 give the interaction that §6.1 reports. Only weight decay differs from Table 1; train accuracy is 1.000 in every run, so every difference is selection among solutions that already fit. Weight decay does not merely speed training up: at weight decay 0 and 40,000 steps the affine arm ends at 0.098, 0.084 and 0.058 across seeds (mean 0.080), while the weight-decay arm reaches 0.494 at the same budget, a gap of $+0.414$ against $+0.399$ at 12k. Summaries that combine factors compare like budgets: at 12k, 0.003 (weight decay, residue coordinates), 0.060 (discrete-log coordinates, no weight decay) and 0.459 (both).

Four optimizer-side diagnostics, none discriminating. The natural mechanism to attach to Proposition 1, that a diffuse spectrum leaves a correlational learner nothing to move on, is a claim about the gradient signal. Four measurements test it against a shuffled-label null at identical shapes, and none separates the cells. Gradient coherence at initialization does not separate learnable from failing cells (1.11 against 0.93–0.96) nor the four intervention cells (1.13–1.18); NTK-target alignment orders nothing (1.5–1.6); coherence measured every 200 steps along training runs in the wrong direction, lower in the cell that generalizes by 4.25 standard errors; and the anti-alignment of the training gradient with generalization, -0.179 against -0.029 on the trajectory, inverts at the logit layer (-0.0118 against -0.1338) and appears in both cells, with the larger decile gap in the one that generalizes (0.636 against 0.410). We record these as four measurements that failed to discriminate, not as a class of account eliminated; train accuracy of 1.000 shows an interpolating solution is reachable and says nothing about whether a generalizing one is. Two further predictions were not supported and should not be read into the main text: Fourier structure is not a leading indicator of generalization, since sparsity keeps growing long after accuracy does, and the optimizer does not prefer the memorizing solution because it is cheaper.

Table 21: Weight decay against presentation, $\ell = 691$, $e = 11$, radix 10, three seeds per cell. Only `weight_decay` differs from Table 1: same modulus, exponent, radix, budget, split key and held-out residues. Train accuracy is 1.000 in every run, so every difference here is selection among solutions that already fit.

weight decay	`res` → `res`	`dlog` → `dlog`	**difference**
0.0	0.000	0.060	+0.060
0.1	0.005	0.120	+0.115
0.3	0.000	0.291	+0.291
1.0	0.003	**0.459**	**+0.456**

F Robustness and additional settings

A standard transformer. Table 22 repeats the design on a standard two-layer transformer with LayerNorm, under five predictions specified before the runs; four are confirmed and one, that every run fits its training set, is refuted. The design covers the four cells of the 2×2 over three seeds, and the two arms of the matched block at $k = 200$ over three permutation seeds. The three non-affine cells score 0.000 held-out in all nine of their runs; `dlog` → `dlog` reaches 0.624 (0.740, 0.695, 0.438). On the matched block at $k = 200$, accuracy on the moved residues is 0.964 under shift against 0.010 under derange, a gap of +0.954 in 3 of 3 pairs. The refuted prediction fails because one `res` → `dlog` run fit only 0.846 of its training set (held-out 0.000), so train accuracy is not 1.000 everywhere in this architecture. Table 23 collects both effects over every configuration in which each was run; the matched block was trained only on the base model and on this architecture.

Table 22: A standard two-layer LayerNorm transformer on the intervention cells and the matched block. Of five predictions specified before the runs, four are confirmed and one is refuted. Four cells over three seeds, two matched-block arms over three permutation seeds. Held-out accuracy. Train accuracy is not 1.000 everywhere: one `res` → `dlog` run fit 0.846 of its training set.

Cell	**held-out**	**note**
`res` → `res`	0.000	
`dlog` → `res`	**0.000**	every mixed-cell run at 0.000
`res` → `dlog`	**0.000**	one run fit 0.846 of its training set
`dlog` → `dlog`	**0.624**	per seed 0.740, 0.695, 0.438
matched block, $k = 200$, shift	0.964	moved residues
matched block, $k = 200$, derange	0.010	moved residues; gap +0.954, 3 of 3 pairs

Table 23: Both headline effects, in every configuration where each was run. *Rescue*: held-out accuracy of `dlog` → `dlog` minus the best of the other three cells of the 2×2. *Matched block*: accuracy on the moved residues at $k = 200$, shift minus derange, over permutation seeds (times training seeds where two were run). The matched block was run in two of the eight configurations. †Mixed cells not run; compared with `res` → `res` only.

	rescue			**matched block,** $k = 200$		
configuration	**gap**	n	**steps**	**gap**	**seeds**	**steps**
1-layer transformer, $\ell = 691$, radix 10	+0.456	3	12k	+0.971	3×2	12k
1-layer transformer, $\ell = 691$, radix 7	+0.281	8	12k		*not run*	
1-layer transformer, $\ell = 3617$, radix 10	+0.853	3	12k		*not run*	
1-layer transformer, 40k steps	+0.542†	2	40k		*not run*	
LSTM	+0.521	2	12k		*not run*	
MLP	+0.387	2	12k		*not run*	
2-layer transformer, no LayerNorm	+0.279†	5	12k		*not run*	
2-layer transformer, LayerNorm	+0.624	3	12k	+0.954	3	12k

Other input radices. Table 24 repeats the degree-1 and degree-2 cells at $\ell = 389$ with the input encoded as residue digits in radices 2, 7 and 16, vocabularies of 3, 8 and 17 tokens over 10, 5 and

4 positions, and the answer emitted in radix 10 through the digit output head, two seeds per cell at 12k steps. The degree-1 cell is learned and the degree-2 cell fails in every base. These runs are exploratory.

Table 24: Input radix. $\ell = 389$, input residue digits in the stated radix, answer in radix 10 through the digit output head, two seeds per cell at 12k steps; held-out accuracy per seed. Exploratory.

input radix	degree 1	degree 2
2	0.962, 0.744	0.000, 0.000
7	0.730, 0.763	0.000, 0.000 (seed 0 fit 0.899 of its training set)
16	0.896, 0.801	0.000, 0.011

Scale. Table 25 gives six cells separating a larger *group* from a larger *model*: $\ell = 691$ at 6.3M parameters isolates capacity, $\ell = 10007$ at 0.8M isolates the modulus. The best snapshot is shown for every cell, not only where it helps. Degree 1 is learned at 0.997 and 0.937; degree 2 fails at 0.001 and 0.000; the degree-1 minus degree-2 gap widens to 0.996 from 0.983 at $\ell = 691$, which is the direction the $\sqrt{\ell}$ scaling of the requirement predicts.

Table 25: Scale. Six cells separating a larger *group* from a larger *model*: $\ell = 691$ at 6.3M isolates capacity, $\ell = 10007$ at 0.8M isolates the modulus. `best` is shown for every cell, not only where it helps.

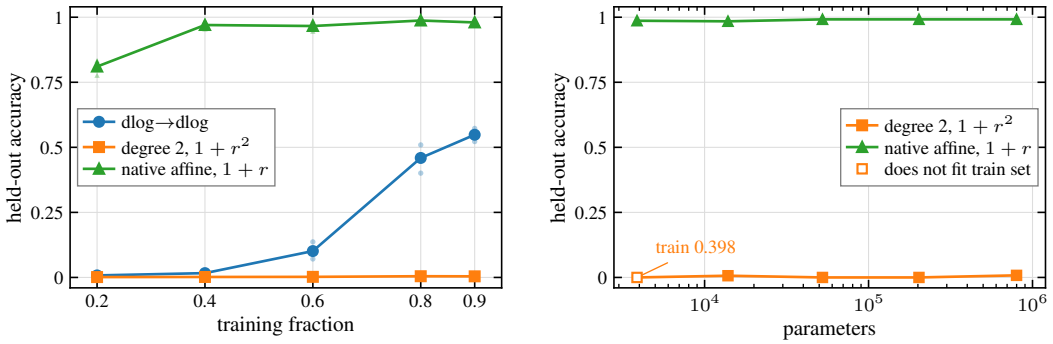
ℓ	capacity	params	degree	predicted	test (final)	best
691	large	6.32M	1	LEARN	0.983	0.992
691	large	6.32M	2	FAIL	0.000	0.006
10007	large	6.33M	1	LEARN	$\mathbf{0.997 \pm 0.001}$	0.999
10007	large	6.33M	2	FAIL	$\mathbf{0.001 \pm 0.001}$	0.002
10007	small	0.80M	1	LEARN	0.937	0.999
10007	small	0.80M	2	FAIL	0.000	0.001

The failing network implements no global rule. A natural worry is that a failing model has learned some other structured but wrong function. At $\ell = 251$ we scored the fraction of *test* predictions matching each candidate rule. The learned degree-1 model matches the true target and the affine rule $1 + r$ on 0.858 of test inputs, a linear rule $2r$ on 0.000 and a constant on 0.041; the failed degree-2 model matches the true target on 0.021, the affine rule on 0.000, the linear rule on 0.000 and a constant on 0.061. Read-out residue-explained variance is 1.000 on train and test for *both* models, so the residue is represented in both and the failure is not representational. The scope is one modulus and a small candidate set: the failed model matches no global rule we tested.

Not a sample-size effect. Across five training fractions, a $4.5\times$ increase moves the degree-2 cell from 0.001 to 0.004 (Figure 6a). A bound on hypotheses contains no sample size and therefore predicts nothing about sample size; the insensitivity is an empirical finding, consistent with the bound but not derived from it.

Not delayed grokking, on either field. At a $10\times$ budget, scored on the *maximum* test accuracy reached at any point in 120,000 steps, no failing cell on either field exceeds 0.0207 while the affine controls still reach 0.992 and 1.000 (Table 26). Three further runs at weight decay 3.0, listed in Table 2 but not in Table 26, also fail, best test at most 0.0075; the $\mathrm{GF}(2^{12})$ one fit only 0.520 of its training set, so it is not a selection test.

Capacity. Driving capacity down to 3,872 weights blocks memorization (Figure 6b): train accuracy falls to 0.398 and degree-2 held-out accuracy stays at or below 0.008 across a $200\times$ capacity range, while the degree-1 arm still learns at 0.986. On a capacity-limitation account, removing the capacity that supports memorization should leave the simpler generalizing solution intact; instead the network fails at everything. Decoding ν_g along the same sweep gives the selection curve of §6: the degree-1 arm converges to $\nu_g \approx 2.3$ against a target ν of 1.000 (2.79 at 3,872 weights), and the degree-2



(a) training fraction (b) capacity

Figure 6: Two robustness checks at $\ell = 691$. (a) Held-out accuracy against training fraction. The native affine map $1 + r$ is learned from a fraction of 0.2 and the degree-2 map $1 + r^2$ stays at chance across a $4.5\times$ increase; the re-encoded $\texttt{dlog} \to \texttt{dlog}$ cell is affine as presented and still needs most of the data. (b) Held-out accuracy against parameter count, 20k steps, one seed per point (Table 27). The degree-2 map fails across a $200\times$ capacity range and, at the smallest width, does not fit its training set (train 0.398); the affine map is learned at every width.

Table 26: Long-budget exclusion of delayed grokking, on both fields. Scored on the *maximum* test accuracy reached at any point in training, so a transient grokking spike would have counted as a refutation even if the run later collapsed.

Field	cell	ν	steps	final train	best test ever
$\mathbb{Z}/691$	degree 1 (control)	1.0	120,000	1.000	**0.9917**
$\mathbb{Z}/691$	degree 2, seed 0	26.3	120,000	1.000	0.0146
$\mathbb{Z}/691$	degree 2, seed 1	26.3	120,000	1.000	0.0136
$\mathbb{Z}/691$	degree 3	22.3	120,000	1.000	0.0075
$\mathbb{Z}/691$	degree 2, wd 0.1	26.3	60,000	1.000	**0.0207**
$\mathrm{GF}(2^{10})$	$e = 2$ (control)	1.0	120,000	1.000	**1.0000**
$\mathrm{GF}(2^{10})$	$e = 7$	25.2	120,000	1.000	0.0049
$\mathrm{GF}(2^{12})$	$e = 11$	48.1	120,000	0.981–1.000	0.0037
$\mathrm{GF}(2^{10})$	$e = 7$, wd 0.1	25.2	60,000	1.000	**0.0098**

arm sits at 23.3 at 3,872 weights and near 24 at full capacity, above the requirement and at the random-map anchor throughout (Table 27, one seed per width).

Table 27: The capacity sweep in full: 20k steps, one seed each; Spearman(ν_g, test) $= -0.786$ and Spearman(params, test) $= +0.278$ over the ten rows (Spearman on unrounded values, average ranks). "Required ν_g at 50%" is $(0.5 - 1/q)/\varepsilon$ for that row's target.

degree	d_{model}	params	train	test	decoded ν_g	target ν	ε	required ν_g at 50%
1	16	3,872	1.000	0.986	2.791	1.00	1.0000	0.50
1	32	13,888	1.000	0.984	2.785	1.00	1.0000	0.50
1	64	52,352	1.000	0.992	2.270	1.00	1.0000	0.50
1	128	203,008	1.000	0.992	2.270	1.00	1.0000	0.50
1	256	799,232	1.000	0.992	2.270	1.00	1.0000	0.50
2	16	3,872	0.398	0.000	23.275	26.25	0.0380	13.11
2	32	13,888	1.000	0.007	24.256	26.25	0.0380	13.11
2	64	52,352	1.000	0.000	24.247	26.25	0.0380	13.11
2	128	203,008	1.000	0.000	24.249	26.25	0.0380	13.11
2	256	799,232	1.000	0.008	24.250	26.25	0.0380	13.11

F.1 External benchmarks

Two architecture arms. Table 28 reports both arms at identical modulus, seeds, optimizer and budget, 2 seeds each. Train accuracy is 1.000 everywhere and the positive controls reach 1.000 on both arms, so neither protocol is broken. Total polynomial degree is identical within each learnable and failing pair, so degree cannot be the discriminating variable.

Table 28: External validation on the tasks of Doshi et al. (2024): two architecture arms at identical modulus, seeds, optimizer and budget, 2 seeds each. Train accuracy is 1.000 everywhere and the positive controls reach 1.000 on both arms, so neither protocol is broken. Total polynomial degree is identical within each learnable and failing pair.

Task	Doshi et al.	Prediction	ReLU transf.	Quadratic MLP
$L1 = (4a + b^2)^3$	learn	LEARN	0.269	**0.780**
$L2 = (2a + 3b)^4$	learn	LEARN	0.028	**1.000**
$L3 = (5a^3 + 2b^4)^2$	learn	LEARN	**0.994**	**0.999**
$N1 = L1 + ab$	fail	FAIL	0.029	0.022
$N2 = L2 - a^2$	fail	FAIL	0.041	0.033
$N3 = L3 - b$	fail	FAIL	0.693	0.527
$X_{L2,\mathrm{lin}}$ ($L2$ + degree-1 term)	–	crossed	–	0.019
$X_{L3,\mathrm{quad}}$ ($L3$ + degree-2 term)	–	crossed	–	0.921
$a + b$	learn	LEARN	1.000	1.000
ab	learn	LEARN	1.000	1.000
$a^2 + b^2$	learn	LEARN	1.000	1.000
$ab + a + b$	learn	FAIL *(rule gap)*	1.000	1.000

The reachability criterion is a joint property of encoding and architecture. It reproduces their classification on 3 of 6 tasks on our ReLU transformer and on 5 of 6 on the two-layer quadratic MLP their construction is built on. This is why "reachable basis" is scoped in two tiers rather than left open. **Tier 1** is digit encodings with no per-value parameter, where ν is computed in the presented basis with *zero* free parameters; every core claim of the main paper lives here. **Tier 2** is atomic-token encodings, where the embedding and unembedding rows supply g_1, g_2 and h for free, and the basis in which ν should be read is the one the architecture can reach; the criterion is predictive there but not parameter-free, and we mark it as such.

The failing class is not explained. Our account does not explain the spread *inside* the failing class: three successive hypotheses about it were refuted, and we do not offer a fourth, post-hoc one. The third was a crossed design testing whether the *degree* of the perturbation is the axis: a degree-1 perturbation of $L2$ scores 0.019 where $N2$, its degree-2 perturbation, scores 0.033, and a degree-2 perturbation of $L3$ scores 0.921 where $N3$, its degree-1 perturbation, scores 0.527 (Table 28, middle rows). Both predictions fail, so perturbation degree is not the axis. The spread is posed as an open problem.

External prediction on the operations of Power et al. (2022). The decidable rule that reproduced Doshi et al. (2024) 5 of 6 on the matched architecture, additively or multiplicatively separable implies LEARN and otherwise FAIL, was applied unchanged to nine binary operations modulo 97 from Power et al. (2022), with no training (Table 29). The rule applied to every operation without exceptions and predicts FAIL for the operation Power et al. single out as non-generalizing, as specified in advance, but agrees with the published record on 5 of 9 against a target of 7, so that prediction is refuted. The rule is conservative: it never predicts LEARN wrongly, and the four operations it calls separable ($x + y$, $x - y$, x/y, $x^2 + y^2$) all generalized, but it misses four that generalized: the conditional operation, $x^2 + xy + y^2$, $x^2 + xy + y^2 + x$ and $x^3 + xy$. For $x^2 + xy + y^2$ the reason is visible, a factoring over $\mathbb{F}_{97}$ into two affine forms in both operands that the rule does not search for; the other three misses have no account. These misses mark the limit of a rule that searches only for additive or multiplicative separability, here shown on a published benchmark. Our initial transcription of Power et al.'s operation list substituted $x \cdot y$ for their conditional operation; Table 29 is scored against their published list and Figure 2. Worth a sentence on its own: on these nine operations the basis-blind ν

does *not* order the outcomes, since the one operation that fails (68.99) has lower ν than three that succeed (96.01) and the conditional operation generalizes at 49.29, so on atomic-token pair tasks the core reading is required and the raw statistic is not merely weaker but wrong-signed. We report the 5 of 9, the four misses by name and the failed ordering, and do not present this as a validation.

Table 29: Predicting Power et al. (2022) from label tables alone. Core ν is the Fourier L^1 norm of the reachable core after free per-operand and output relabellings; basis-blind ν is the map's own.

operation	class	core ν	basis-blind ν	predicted	published	
$x + y$	additive	1.00	1.0	LEARN	learn	✓
$x - y$	additive	1.00	1.0	LEARN	learn	✓
x/y	multiplicative	1.00	96.0	LEARN	learn	✓
x/y if y is odd, otherwise $x - y$	none	n/a	49.29	FAIL	learn	**miss**
$x^2 + y^2$	additive	1.00	96.0	LEARN	learn	✓
$x^2 + xy + y^2$	none	96.01	96.0	FAIL	learn	**miss**
$x^2 + xy + y^2 + x$	none	96.01	96.0	FAIL	learn	**miss**
$x^3 + xy$	none	96.01	96.0	FAIL	learn	**miss**
$x^3 + xy^2 + y$	none	68.99	69.0	FAIL	fail	✓

The same nine operations under digit tokens. The two-tier reading above predicts that removing the free per-operand relabellings changes which operations are learnable. With each operand written as digits, the embedding rows can no longer supply g_1, g_2 and h, so the basis-blind ν of Table 29 applies and calls only $x + y$ and $x - y$ learnable. All nine operations were trained under digit tokens with the predictions specified before the runs; the conditional operation was specified and run separately, after the initial operation list was found to substitute $x \cdot y$ for it. The learnable set shrinks from eight of nine to two, all nine land on the predicted side of the 0.5 criterion, and eight of nine meet their bars; the conditional operation reached held-out 0.206 against a bar of 0.10. The prediction that every run reaches train accuracy ≥ 0.99 is refuted: six runs fall below it, the lowest at 0.701, and operation means reach down to 0.900. Every run of an operation called FAIL still keeps a train-minus-held-out gap between 0.68 and 0.99, which is memorization rather than a failure to fit. The comparison is with the published atomic-token runs rather than a matched atomic-token arm in our protocol, so it is a called outcome and not an isolated effect of tokenization. This is a test of the two-tier reading and not of the §5 statistic, whose value on the pair group does not change with tokenization.

Table 30: The nine operations of Power et al. (2022) modulo 97 under digit tokens, calls fixed before the runs. The learnable set shrinks from eight (atomic tokens, published) to two. Held-out and train are means over three seeds. Bars are the thresholds set before the runs, a floor for the two LEARN calls and a ceiling for the seven FAIL calls; "met" is against the bar, "side" against the 0.5 criterion.

operation	atomic (publ.)	basis-blind ν	call (digits)	held-out	train	bar	met	side
$x + y$	learn	1.0	LEARN	0.999	1.000	> 0.90	✓	✓
$x - y$	learn	1.0	LEARN	0.996	1.000	> 0.90	✓	✓
x/y	learn	96.0	FAIL	0.028	0.900	< 0.10	✓	✓
x/y if y odd, else $x - y$	learn	49.29	FAIL	0.206	1.000	< 0.10	**miss**	✓
$x^2 + y^2$	learn	96.0	FAIL	0.091	0.924	< 0.10	✓	✓
$x^2 + xy + y^2$	learn	96.0	FAIL	0.063	0.990	< 0.10	✓	✓
$x^2 + xy + y^2 + x$	learn	96.0	FAIL	0.015	1.000	< 0.10	✓	✓
$x^3 + xy$	learn	96.0	FAIL	0.018	0.996	< 0.10	✓	✓
$x^3 + xy^2 + y$	fail	69.0	FAIL	0.011	0.919	< 0.10	✓	✓

F.2 THE BOOLEAN CUBE

A first pair of tests on the Boolean cube produced no boundary, for three design reasons that are stated below and repaired. The repaired family gives an ordering result on a non-cyclic group and one regime mismatch that we report rather than absorb.

Three design constraints. A one-bit output puts chance at 0.5, so the interval between failure and success spans a factor of two rather than the factor of a thousand available at $\ell = 691$; the first cells ran at roughly $4{,}000\times$ the examples per class of the modular tasks, so nothing failed; and $\nu = 2^m$ confounded ν with domain size, since m also set the input count.

The repaired family. A $k = 6$ carrier gives 64 output classes and chance 0.0156, and sampling falls to $2.5\times$ the modular tasks at $m = 3$ and $9.8\times$ at $m = 4$. The knob is a bent form on the first t of m coordinate pairs, XORed onto one carrier bit, which gives $\nu = (1 + 2^t)/2$ in closed form, verified against the implementation at all nine cells. ν **now varies at fixed domain, fixed class count and fixed chance**, which is the cube analogue of the matched design of §4. A first design was rejected before any run: scrambling j carrier bits by distinct bent forms gives $\nu = 1.00, 8.50, 8.75, 8.13, 7.19, 6.72$, peaking at $j = 2$ and falling, the same saturation as mode A18.

Table 31: The repaired Boolean-cube family. A bent form on the first t of m coordinate pairs, XORed onto one bit of a $k = 6$ carrier, so $\nu = (1 + 2^t)/2$ varies at fixed domain, class count and chance (0.0156).

m	t	ν	**held-out**	**train**
3	0	1.000	1.000	1.000
3	1	1.500	1.000	1.000
3	2	2.500	0.998	0.999
3	3	4.500	0.776	**0.878**
4	0	1.000	1.000	1.000
4	1	1.500	1.000	1.000
4	2	2.500	0.849	**0.851**
4	3	4.500	0.813	**0.828**
4	4	8.500	0.651	**0.672**

ν **orders held-out accuracy at Spearman** -0.956 **pooled and** -0.949 **within the sampling-matched** $m = 3$ **arm**, over cell means (-0.824 and -0.811 over individual runs; Table 31), with the formulas and the codomain group stated in closed form rather than estimated. That is the decision rule set for this test for the account transferring to a non-cyclic group.

The regime does not match, and that limits the claim. On $\mathbb{Z}/\ell$ train accuracy is 1.000 in every cell, so that boundary is a matter of *selection among fitting solutions*. Here the hard cells **do not fit**: at $m = 4, t = 4$, train is 0.672 against held-out 0.651, and test tracks train down the column. That is an optimisation failure, not a selection failure, so this is not the same phenomenon measured on another group and we do not report it as one. The transition also sits at higher ν than on $\mathbb{Z}/\ell$: at $\nu = 8.5$ accuracy is 0.651, well above the range where the modular families have collapsed. **The defensible claim is that the ordering transfers, not that the boundary does.** The control that would close the gap, and which we have not run, is more capacity or budget on the cube until the hard cells fit, followed by a re-test of the ordering inside the fitting regime.

Table 32: The coordinate rescue transfers across architectures. The four intervention cells of Table 1 repeated with only the architecture swapped; both crossed cells sit at exactly 0.000 on both new architectures. The LSTM and MLP ran at the matched 12k steps; the transformer column is the 40k-step run.

Cell	**LSTM**	**MLP**	**Transf.**
`res` → `res`	0.004	0.004	0.000
`dlog` → `res`	**0.000**	**0.000**	n/a
`res` → `dlog`	**0.000**	**0.000**	n/a
`dlog` → `dlog`	**0.525**	**0.391**	0.542
rescue	**+0.521**	**+0.387**	+0.542

Why the characteristic-2 transplant is the valid analogue. The $\mathrm{GF}(2^n)$ design (Table 33) is matched to the modular tasks by construction on bijectivity, class count, and examples per class at 0.80 per class. It keeps the shape of the modular task and changes only the field, which is what a field-generality test requires.

Table 33: The boundary and its coordinate rescue over $\mathrm{GF}(2^n)$, for the nine tested exponents. A Frobenius exponent (binary weight 1) presents a linear map over $\mathrm{GF}(2)$ and so has $\nu = 1$ exactly; the tested one is learned at every n. Each tested exponent of binary weight ≥ 2 has ν between 11.5 and 48.1 and fails in element coordinates. We make no claim about untested exponents.

n	e	**binary**	**Frobenius?**	ν **(element)**	**element acc**	**dlog acc**
8	2	`10`	yes	1.000	$\mathbf{1.000 \pm 0.000}$	1.000 ± 0.000
8	7	`111`	no	11.459	0.007 ± 0.009	0.392 ± 0.055
8	11	`1011`	no	11.459	0.007 ± 0.009	0.190 ± 0.024
10	2	`10`	yes	1.000	$\mathbf{1.000 \pm 0.000}$	1.000 ± 0.000
10	5	`101`	no	15.985	0.002 ± 0.002	0.659 ± 0.249
10	7	`111`	no	25.226	0.000 ± 0.000	0.823 ± 0.017
12	2	`10`	yes	1.000	$\mathbf{1.000 \pm 0.000}$	0.978 ± 0.031
12	11	`1011`	no	48.113	0.000 ± 0.001	0.881 ± 0.004
12	17	`10001`	no	15.996	0.000 ± 0.000	0.335 ± 0.035

G EXPERIMENT INDEX AND RELEASED MATERIALS

Table 35 lists the experiments behind the paper's claims: where each claim is made, the prediction or question it tests, and the results that support it. Every reported number is produced by the released analysis scripts from the stored results. For the headline prediction experiment of §5, the rule ($\nu < 2$) and its success criterion (held-out accuracy above 0.7 for LEARN, below 0.1 for FAIL) were fixed before the ten two-term cells at $\ell = 389$ and 577 were trained, and the eight power-map cells at $\ell = 251$ and 389 were scored with the same frozen rule. Which statistic the paper presents as the finding was decided after the tournament and the best-affine comparison. Analyses developed after inspecting results are labelled post hoc where they appear. The full result sweeps, the configurations, the analysis scripts and the corrections noted in Appendix A.1 are in the released code repository (Code and data availability).

The held-out decision rule. Table 34 states the procedure behind the 18 calls of §5 in full, separating the rule from the criterion that scores it and the predictions specified before training from those evaluated with the unchanged rule.

Table 34: The decision rule behind the 18 held-out calls of §5, and when each part was fixed.

element	**specification**	**fixed**
input	the complete presented label table of each cell, withheld labels included (Eq. 1)	by construction
rule	LEARN exactly when $\nu < 2$; otherwise FAIL	before the two-term cells were trained
family-specific logic	none	with the rule
predictions specified before training	ten two-term cells $r \mapsto r^a + r^b$ at $\ell = 389$ and 577	before training
calls from the unchanged rule	eight power-map cells at $\ell = 251$ and 389	rule applied unchanged
success criterion	held-out accuracy > 0.7 for LEARN and < 0.1 for FAIL, on three-seed means; scores each call and is not part of the rule; all 18 cells also fall on the called side of the 0.5 criterion	before the runs
later analyses	which statistic best explains the outcomes (tournament, best-affine comparison, controlled families)	after the results; post hoc

Table 35: Experiment index: where each claim is made, the prediction or question it tests, and the results that support it. Rows marked post hoc tested no prediction specified in advance; rows marked proved are identities checked numerically; the tournament row is the development analysis.

where	**prediction or question tested**	**supporting results**
§2, App. B	ε orders Serre's regimes	`hecke_spectral_report.md`
App. B, Table 5	large-image excess is a determinant effect (post hoc)	`hecke_determinant_report.md`
§6, App. B.2	Props. 1–3 across 35 cells (proved)	`prop45_moduli_report.md`
§5	8 of 8 power-map cells, $\ell = 251, 389$	`held_out_predictions.md`
§5	10 of 10 two-term cells, $\ell = 389, 577$	`held_out_predictions.md`
§5	rivals ruled out	`held_out_predictions.md`

where	prediction or question tested	supporting results
§5	perturbed / piecewise families	`held_out_predictions.md`
§5, App. D	predictor tournament (development split)	`predictor_tournament_report.md`
§3, Table 1	the 2×2 intervention	`dlog_intervention_report.md`
§3, App. C	$e = 13$ shortcut control	`dlog_controls_report.md`
§3, App. C	40k convergence check	`dlog_controls_report.md`
§3, App. C	radix-7 control	`answer_base_report.md`
App. C	eight-seed confirmatory run	`basis_experiments_report.md`
§4	scrambled vs natural basis	`basis_experiments_report.md`
§4, Table 8	the graded ladder	`scramble_ladder_report.md`
§4, Table 8	ladder decomposition	`ladder_decomposition_report.md`
App. E.1	representation sparsity	`circuit_analysis_report.md`
§6, App. E.1	first Fourier ablation	`fourier_ablation_report.md`
§6, App. E.1	graded ablation	`fourier_ablation_graded_report.md`
§6, Fig. 5, Table 19	ten-seed replication	`ablation_replication_report.md`
§6, Table 19	energy-matched ablation	`ablation_energy_matched_report.md`
§6, App. E.1, Table 20	Fourier supply	`fourier_supply_report.md`
§6, Table 20	the scramble control	`fourier_supply_report.md`
§6	grokking at `eval_every` = 100	`grokking_report.md`
§6, App. E.2	initialization geometry	`init_signal.json`
App. E.2	gradient alignment	`gradient_alignment_report.md`
App. E.2	anti-alignment by mode	`gradient_spectrum_report.md`
App. A.1, App. E.2	trajectory probe	`trajectory_report.md`
§7, Table 32	rescue across architectures	`arch_intervention_report.md`
§7, Tables 32, 33	$\mathrm{GF}(2^n)$ transplant	`gf2_transplant_report.md`
§7, App. F, Table 25	scale at $\ell = 10007$	`scale_report.md`
§7, App. F.1, Table 28	Doshi et al., ReLU arm	`doshi_external_report.md`
§7, App. F.1, Table 28	Doshi et al., quadratic arm	`doshi_two_arm_report.md`
App. F.1	the crossed design	`doshi_two_arm_report.md`
App. E	capacity down	`capacity_down_report.md`
App. E	ν_g across capacity	`nu_g_capacity_report.md`
§6, App. E	decoded ν_g	`nu_g_decoded_report.md`
App. F.2	Boolean cube	`boolean_cube_report.md`
App. F.2	cube inversion (four predictions scored; two not run)	`cube_inversion_report.md`
App. F, Fig. 6	training-fraction grid	`phase_diagram_report.md`
App. F, Table 26	long budget, $\mathbb{Z}/\ell$	`modular_longbudget_report.md`
App. F, Table 26	long budget, $\mathrm{GF}(2^n)$	`gf2_longbudget_report.md`
§5, App. D, Table 14	predictors on untouched families	`predictor_heldout_families_report.md`
§6, Table 18	matched non-Fourier control	`reconstruction_nonfourier_report.md`
§3, Table 1	full 2×2 in radix 7	`base7_full_intervention_report.md`
§4, Fig. 2, Table 9	matched fixed points	`matched_fixed_points_report.md`
§6, Fig. 5	reconstruction controls	`reconstruction_controls_report.md`
§6, App. E.1, Table 20	supply controls, repaired	`supply_controls_repaired_report.md`
§7, App. F.1, Table 29	external prediction, Power et al.	`external_grokking_prediction_report.md`
§7, App. F.1, Table 30	Power et al. under digit tokens	`power_digits_report.md`
§7, App. F.1, Table 30	the conditional operation under digit tokens	`power_digits_conditional_report.md`
§5, App. D	dilated blocks	`dilated_blocks_report.md`
§3, Table 1	second theorem rescued	`second_theorem_rescue_report.md`
§4, Fig. 2c, Table 10	piece count	`piece_count_report.md`
App. C, Table 11	fixed piece count, varying ν	`fixed_piece_count_report.md`
§6.1, Table 21, Fig. 3	weight decay $\times$ presentation	`weight_decay_interaction_report.md`
§6.1	40k convergence control	`wd_convergence_report.md`
App. F.2, Table 31	repaired Boolean cube	`boolean_multibit_report.md`
App. E	ν_g trajectory	`nu_g_trajectory_report.md`
§3, §4, App. F	standard two-layer LayerNorm transformer	`standard_transformer_report.md`
§6, App. E.1, Table 17	direct-label baseline	`direct_label_baseline_report.md`

Where a prediction did not hold and that changed what we claim, the section carrying the claim says so: the energy- and rank-matched comparisons are why the paper does not claim a Fourier-specific mechanism; the structured non-Fourier supply arm is why §6 reports an effect of presented structure rather than of the Fourier basis; the best-affine rival is why §5 reports presented affineness as the finding; the external rule of Appendix F.1 matches 5 of 9 published outcomes, so §7 states it as conservative; and the fitting failure on the Boolean cube is why §9 reports the cube as an ordering result and not the same regime.