DATA COVERAGE GATES THE GROKKING TRANSITION IN SMALL TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Grokking (sudden generalization long after memorization) is well-studied in small transformers on modular arithmetic, but the role of *training-data diversity* in the transition is unclear. In this controlled setting we isolate *unique-pair coverage* (the fraction of the p^2 ordered pairs seen) from total sample count. Our central finding is that unique-pair coverage is a causal, hysteretic boundary for the *formation*, but not the *maintenance*, of the Fourier circuit that enables grokking. Raising coverage past the critical value c^* mid-training *triggers* grokking ($9/10$ seeds vs. $0/10$ control); cutting it below c^* after the circuit forms leaves it intact ($10/10$); and far below c^* the failed state is operatively absorbing over $500{,}000$ steps. *Where the boundary is:* within a budget of $100{,}000$ steps, $c^* \approx 24\%$ of unique pairs at the default hyperparameters (15.6%–34.9% across weight-decay and learning-rate settings), with $c^*(p) \propto p^{-0.36}$ under the held-out criterion, an intermediate scaling of the required pair budget between the $O(p \log p)$ and $O(p^2)$ predictions. *What crossing it changes:* the representation stays diffuse below c^* and collapses onto a sparse Fourier circuit only above it, a circuit-efficiency count whose predicted exponent falls inside the measured interval. *What it is not:* the threshold reproduces under the without-replacement epoch protocol of prior work, is not an artifact of repetition or frequency skew, and carries over within the family to a second task and a two-layer transformer of the same width. Because data-scaling experiments vary coverage and sample count together, dataset-size effects should be read with coverage held fixed.

1 INTRODUCTION

Neural networks exhibit *grokking* (Power et al., 2022): after memorizing training data, the model suddenly generalizes thousands of steps later. First observed in small transformers on modular arithmetic, it has since been replicated across architectures and datasets (Liu et al., 2022; Nanda et al., 2023; Chughtai et al., 2024; Kumar et al., 2024; Varma et al., 2023).

Understanding what triggers it matters for generalization: prior work implicated weight decay (Power et al., 2022; Nanda et al., 2023), model size (Liu et al., 2022), task structure (Chughtai et al., 2024), and training dynamics (Kumar et al., 2024; Barak et al., 2022).

However, in prior work *data diversity* and *dataset size* are confounded: Liu et al. (2022) varied size with each pair appearing once, so their "dataset size" effect may be a coverage effect. We disambiguate them with a factorial design and ask: does the training data determine *whether* the generalizing circuit forms, and whether it can then be lost? Our central finding: unique-pair coverage is a *causal, hysteretic boundary* for the formation, but not the maintenance, of the Fourier circuit that enables grokking: raising coverage past c^* mid-training *triggers* grokking ($9/10$ seeds vs. $0/10$ control), while cutting coverage below c^* after formation leaves the circuit intact ($10/10$). The transition is thus *hysteretic*: not determined by the final training set alone. Far below c^* the failed state is operationally absorbing over $500{,}000$ steps; just below it the transition is merely slow. The point extends beyond grokking: **common data-scaling experiments vary coverage and total sample count simultaneously**, and our controlled setting isolates coverage as a control variable.

The rest of the paper serves that claim:

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.

1. *Where the boundary is.* Within a budget of $100{,}000$ steps, grokking reliably occurs above $c^* \approx 24\%$ of the $4{,}489$ ordered pairs for $p = 67$ (95% CI $[22.0\%, 25.5\%]$), a soft boundary rather than a hard per-seed cutoff, since near it only some seeds grok ($6/10$ at 25%, $10/10$ at 30%); under the held-out criterion $c^*(p) \propto p^{-0.36}$ ($D_{\text{crit}} \propto p^{1.64}$), an intermediate empirical scaling over the tested range $p \in \{47, 67, 97, 113, 131, 173\}$ that brackets the $O(p \log p)$ and $O(p^2)$ predictions.

2. *What crossing it changes.* The representation stays diffuse (high entropy) below c^* and collapses onto a sparse Fourier circuit only above it; a circuit-efficiency count $c^* \approx \kappa N_{\text{eff}}/p$ predicts an exponent inside the measured confidence interval, and removing that circuit's support mid-training delays grokking, though every seed rebuilds it (Section A.9).

3. *What it is not.* The threshold is not an artifact of protocol, repetition, or skew (Section 4.5): it reproduces under the without-replacement epoch protocol of prior work, and at matched coverage concentrated frequency does not suppress grokking. Within the modular-arithmetic family it also carries over to modular multiplication and to a two-layer transformer of the same width (Section 4.4).

2 Related Work

2.1 Grokking and Phase Transitions

The grokking phenomenon (sudden generalization long after overfitting) was first systematically studied by Power et al. (2022) on modular arithmetic, which identified weight decay as a critical driver; Nanda et al. (2023) showed the mechanism is the gradual amplification of a discrete-Fourier circuit. Varma et al. (2023) formalized this as a circuit-efficiency competition in which grokking occurs once a generalizing circuit is more weight-efficient than memorization, defining a critical dataset size D_{crit} below which memorization wins. We reframe D_{crit} as a *coverage* threshold: what is fixed is the critical number of *unique* pairs, c^*p^2, independent of repetition (Section 4.5). Tian (2025) proved a two-layer network needs only $O(M \log M)$ samples for group operations.

Liu et al. (2022) showed that larger models grok later and that the grokking epoch scales with inverse dataset size; there each pair appeared once, so their "dataset size" effect may be primarily a coverage effect; Liu et al. (2023) extended grokking beyond algorithmic tasks. Related work links grokking to hidden progress (Barak et al., 2022), lazy-to-rich dynamics (Kumar et al., 2024), and task geometry (Mohamadi et al., 2024). Accounts also differ on what *kind* of transition this is: Zhang et al. (2025) sample the Boltzmann entropy landscape of small networks and, finding no entropy barrier, argue against first-order descriptions (Appendix A.11).

Very recent concurrent work independently converges on a coverage-centered account. Ootani (2026) studies a fully tractable, many-seed network at 12K parameters and reports that grokking is *conditional*: whether a run generalizes is determined by the coverage of the training data rather than by the number of samples, with the critical coverage depending on the modulus (output cardinality), the same conditioning our factorial design isolates (Section 4.5). Using a discrete-Fourier synchronization measure, Sivasankar (2026) provides causal evidence that the generalizing circuit synchronizes hundreds to thousands of steps *before* the accuracy transition, reading grokking as circuit formation followed by weight-decay-driven readout; this matches the coverage-gated circuit *formation* we isolate causally (Section 4.2). Three further results bear on our account without reproducing it: Nguyen & Reddy (2025) locate the analogous transition in in-context learning at a task-diversity threshold set by a dynamical competition between memorizing and generalizing sub-circuits rather than by capacity; Boursier et al. (2025) derive the two-phase structure (a fast phase onto a critical manifold followed by slow norm minimisation under weight decay) that our formation/readout split instantiates; and Mallinar et al. (2025) obtain the transition, with the same block-circulant Fourier features, in a non-neural model. Two further results bound the account from opposite sides: Prieto et al. (2025) trace the delay itself to floating-point softmax collapse when such tasks are trained *without* regularization (a regime our weight-decayed runs (1.0 here, 0.3–3.0 in Section 4.5) do not occupy), and Bernal et al. (2026) require only that the training set *break the task's symmetry*, a condition our random subsets meet everywhere on the grid. Appendix A.16 states for each of them what it does and does not establish here.

2.2 Data Diversity in Deep Learning

Data diversity is recognized across many areas of deep learning (Gao et al., 2021); Edelman et al. (2022) showed that data diversity determines which generalizable variables are learned, and Neyshabur et al. (2018) that over-parameterized models generalize with limited data only when it covers the underlying structure. Carvalho et al. (2025) find that skewed, long-tail sampling can induce grokking through a training–test distribution shift rather than data sparsity per se. Our work operationalizes diversity as *coverage* (unique examples): skew suppresses grokking only through the effective coverage it retains (Section 4.5).

3 Experimental Setup

3.1 Task: Modular Addition

We study the modular addition task $a + b \equiv c \pmod{p}$, where $p = 67$ (and $p \in \{47, 97, 113, 131, 173\}$ for cross-modulus scaling experiments). This is the standard benchmark for studying grokking (Power et al., 2022). The input is a triple of tokens $(a, b, =)$, and the model must predict the correct result c. There are p^2 unique input-output pairs ($4{,}489$ for $p = 67$).

3.2 Model Architecture

We use a one-layer transformer (Vaswani et al., 2017) with $d_{\text{model}} = 128$, $n_{\text{heads}} = 4$, and $d_{\text{ff}} = 512$ ($\sim$216K parameters); for cross-architecture generalization (Section 4.4), we also use a two-layer transformer with the same per-layer dimensions ($\sim$414K parameters). The vocabulary size is $p + 1$ (p residues plus "="), with positional embeddings for the three input positions, following prior grokking studies (Power et al., 2022; Nanda et al., 2023).

3.3 Training Procedure

We train with AdamW (Loshchilov & Hutter, 2019) at learning rate 10^{-3}, weight decay 1.0, and batch size 512 for $100{,}000$ steps (cosine schedule). Each batch is drawn i.i.d. with replacement from the training pairs, so a unique pair's sampling probability per step is $1/(c\,p^2)$. We evaluate accuracy every $1{,}000$ steps on all p^2 pairs. Because the reported total accuracy a_{tot} counts the $c\,p^2$ training pairs as correct, we use *held-out-only* accuracy as the primary metric, recovered from the stored per-step curves via $a_{\text{held-out}} = (a_{\text{tot}} - c\,a_{\text{train}})/(1-c)$ (training accuracy reaches $\sim$100% shortly after memorization; Appendix A.1.2). The *grokking step* is the first evaluation where held-out accuracy exceeds 90%. Each experiment uses 10 seeds; we assess significance with two-tailed Fisher exact (success rates) and log-rank (time-to-grokking, never-grokked seeds censored) tests, reporting 95% CIs (Wilson for proportions, bootstrap otherwise) and Kaplan–Meier median grokking times. We locate c^* by fitting $p(\text{grok}) = \text{sigmoid}\big((c - c^*)/\beta\big)$ to the dense scan via binomial maximum likelihood, always on a shared coverage grid across conditions so the estimates are directly comparable.

3.4 Data Diversity Manipulations

We manipulate training data diversity in three ways. Throughout, *coverage* c is the fraction of the p^2 ordered pairs (a, b) used for training, the p diagonal pairs (a, a) included, so $c = 1$ corresponds to all $4{,}489$ pairs for $p = 67$ and held-out pairs are always a fraction $1 - c$ of the same p^2 pool:

1. **Dense diversity scan** (Experiment 1): We sample $r \in \{0.05, 0.10, 0.125, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.50, 0.70, 1.0\}$ of unique pairs, each shown once, 10 seeds per condition (Table 5).
2. **Factorial design** (Experiment 2): We independently vary unique-pair coverage (25%, 50%, 100%) and repetition ($1\times$, $2\times$, $4\times$), 7 conditions, 10 seeds each.
3. **Skewed distributions** (Experiment 3): We compare uniform, long-tail, and Zipf sampling, plus a coverage-matched uniform control.

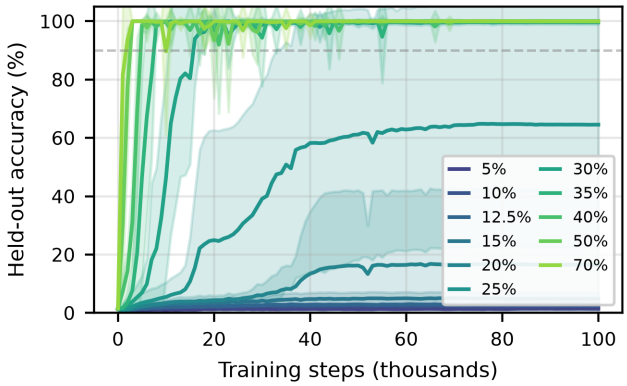

Figure 1: Dense diversity scan under the held-out criterion: held-out accuracy curves across coverage levels (± 1 std over 10 seeds); the dashed line marks the 90% grokking threshold.

3.5 Fourier Circuit Analysis

Following Nanda et al. (2023) and the mechanistic-interpretability literature on transformer circuits (Elhage et al., 2021; Olsson et al., 2022; Elhage et al., 2022; Wang et al., 2023), we project the unembedding weights onto the Fourier basis, building on the spectral bias of neural networks toward low-frequency functions (Rahaman et al., 2019; Tancik et al., 2020). Writing $W \in \mathbb{R}^{p \times d_{\text{model}}}$ for the unembedding matrix, we take its unitary transform along the input axis, $\hat{W}_k = p^{-1/2} \sum_n W_n e^{2\pi i k n/p}$, its power $P_k = \sum_j |\hat{W}_{kj}|^2$, and $q_k = P_k / \sum_{k'} P_{k'}$, the frequency distribution of the unembedding. The *Fourier entropy* is $H(W) = -\frac{1}{\log p} \sum_{k=0}^{p-1} q_k \log q_k \in [0, 1]$, the Shannon entropy of q in units of $\log p$: it is 1 when power is spread uniformly over all p frequencies (the random-init value), and near 0 when the unembedding concentrates on a few: $p^{H(W)}$ is the effective frequency count N_{eff} used below (Table 17). Entropy therefore measures *circuit focusing*: falling entropy tracks the emergence of a sparse Fourier circuit, though tracking is not by itself causal evidence.

4 Results

4.1 A Probabilistic Critical Coverage Threshold

Figure 1 shows the held-out accuracy curves across the dense scan; Appendix Figure 4 shows the median grokking step and never-grok rate near the threshold.

At 5%–20% diversity (224–897 unique pairs) the model memorizes and almost never generalizes (only $1/10$ seeds grok at 20%); held-out accuracy only clearly exceeds chance from 15% onward, while the reported total accuracy looks higher only because memorized training pairs count as correct (Zhang et al., 2021) (Appendix A.1.2).

Reliable grokking first appears at 25% diversity (1,122 pairs), where $6/10$ seeds generalize (95% Wilson CI $[0.31, 0.83]$; Kaplan–Meier median 37k, CI $[18\text{k}, 72\text{k}]$). At 30% (1,346 pairs), all 10 seeds grok (median 12k, CI $[9\text{k}, 14\text{k}]$); the 20% vs. 30% difference is highly significant (Fisher exact, $p = 10^{-4}$; log-rank, $p < 10^{-4}$). A logistic fit to the dense held-out scan locates $c^* = 24.0\%$ (95% CI $[22.0\%, 25.5\%]$). Above 30% the median grokking step falls sharply to $\sim$1,000 by 70% (Table 5): small coverage increases near the threshold yield disproportionately large generalization (Appendix A.1.2). Because individual seeds are stochastic near the boundary, c^* is where grokking *reliably* occurs: a soft boundary, not a hard per-seed cutoff.

4.2 The Formation Boundary Is Bidirectional and Causally Controllable

The threshold of Section 4.1 is correlational: crossing c^* *between* runs changes outcomes. To test causality, we intervene mid-training on nested subsets $S_{\text{small}} \subset S_{\text{big}}$ (one shared per-seed permuta-

tion; protocol in Appendix A.3): *injection* arms start at the absorbing $c = 15\%$ and swap to $c = 30\%$ (past c^*) at step 25k or 50k, a *control* stays at 15% for the whole run, and a *removal* arm starts at $c = 40\%$ (grokked) and swaps down to $c = 20\%$ (below c^*) at step 50k (arm sizes in Table 6; Figures 2 and 5).

The sub-threshold state is absorbing. Re-training every non-grokked seed out to 500,000 steps, none of the 10 seeds at 15% ever groks, whereas near c^* the transition is merely slow ($4/9$ seeds at 20%, $3/4$ at 25%): below the boundary the memorizing state is a stable fixed point, not a slow transient: grokking remains unobservable in this data-scarce regime (Singh et al., 2025).

Injecting coverage triggers formation. Holding 15% for 25,000 steps and then raising coverage to 30% triggers grokking in $9/10$ seeds (grokking steps 26k–74k, i.e. 1–49k steps after the swap), whereas the identical 15% start groks $0/10$ by 100,000 steps if coverage never changes (Fisher, one-sided $p \approx 6 \times 10^{-5}$; Wilson 95% CI $[60\%, 98\%]$ vs. $[0, 28\%]$; Appendix A.3). c^* is a boundary the optimizer can be carried across. The one non-grokked seed memorizes the swapped-in pairs but stays at chance held-out. The same injection at step 50,000 rescues only $3/5$ under cosine scheduling but $5/5$ at a fixed learning rate (grokking at 53–87k), with 15% absorbing either way: the late-injection gap is a learning-rate decay effect.

Removal does not destroy a formed circuit. Cutting coverage from 40% to 20% (below c^*) at step 50,000 leaves all 10 seeds grokked: final held-out on the 60% of pairs never seen at *any* training phase recovers to $\geq$95.7% in every seed. This survival is not the learning-rate decay: the same removal at a constant 10^{-3}, with no schedule, groks $10/10$ again and keeps $9/10$ above the 90% criterion (Appendix A.3).

Coverage thus gates circuit *formation*, not *maintenance*: crossing c^* upward is required to enter the generalizing state, not to remain there. The boundary is *hysteretic*: at the same final coverage, the outcome depends on whether the state was reached from below by injection or from above by the static scan (Figure 2).

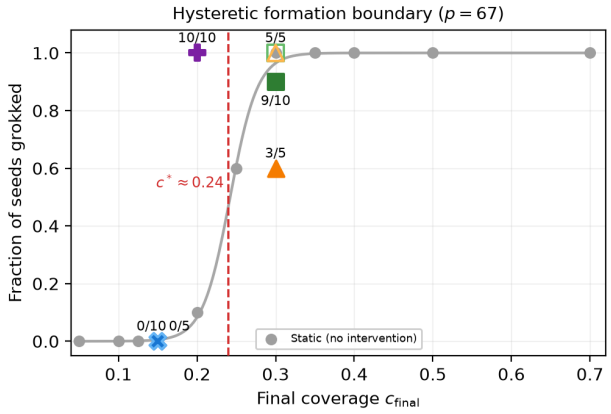

Figure 2: Bidirectional formation boundary. Static grok fraction (held-out $> 90\%$, $p = 67$) vs. final coverage (gray points and logistic fit), with $c^* = 24\%$ marked (dashed). Intervention arms are overlaid at their final coverage: injecting from 15% to 30% at step 25k groks $9/10$ (cosine); at step 50k, $3/5$ under cosine scheduling but $5/5$ at a fixed learning rate (open markers); the 15% control stays at $0/10$ (cosine) and $0/5$ (fixed LR); removing from 40% to 20% leaves $10/10$ grokked under either schedule.

4.3 Threshold Scaling with Modulus p

We next ask how the boundary *moves* with task size, a scope question rather than an independent mechanism. We fit $c^*(p)$ under the held-out criterion on a shared 7-point coverage grid (100% excluded) for $p \in \{47, 67, 97, 113, 131, 173\}$ (10 seeds per point; $p = 173$ adds four points inside its transition, Appendix A.5). The critical proportion *decreases* with p, from 28.1% at $p = 47$ to 17.5% at $p = 173$ (Figure 6, Table 7). A power-law fit over all six moduli gives $c^*(p) \propto p^{-0.36}$ ($\alpha =$

0.359, 95% CI $[0.304, 0.427]$, $R^2 = 0.97$), hence $D_{\text{crit}} \propto p^{1.64}$: an intermediate empirical scaling bracketing the $O(p \log p)$ bound of Tian (2025) and the constant-proportion prediction ($\propto p^2$) of circuit-efficiency theory (Varma et al., 2023). Restricting to the original four moduli gives the steeper $\alpha = 0.416$ ($[0.330, 0.514]$), so we read the exponent as a limited-range fit spanning a factor of $\sim$4, not a strict asymptotic law. The out-of-family point at $p = 257$ (absent from every fit) shows what extrapolation is worth: the four- and six-modulus fits predict $c^*(257) \approx 13.8\%$ and 14.9%, against a measured 15.3% ($[15.0\%, 17.6\%]$), a miss of 1.5 pp shrinking to 0.4 pp, always in the same direction, the true decay being marginally shallower than any fit. The ratio $D_{\text{crit}}/(p \log p) \propto p^{0.64}/\log p$ grows with p, so the gap to the information-theoretic bound does not close. The exponent is not a capacity artifact (at full coverage the circuit uses only $\sim$9–35 effective frequencies; Section 5.2); leave-one-modulus-out cross-validation reproduces the held-out c^*s to 0.74 pp MAE, so $p = 257$ marks where the family's extrapolation ends.

These conclusions are robust to the evaluation protocol: refitting c^* under 80–99% held-out criteria changes it by ≤ 1.1 pp for $p \geq 67$ (Appendix A.1.2).

4.4 Generalization Across Tasks and Architectures

This section delimits the claim rather than adding a finding. Within the modular-arithmetic family the threshold reproduces: modular multiplication shows the same 20–30% critical range, and a two-layer transformer of the same width groks reliably at 25% ($10/10$ vs. $6/10$ one-layer; Appendix A.6); settings beyond this family and width are untested (Section 5.5).

4.5 The Boundary Is Not an Artifact of Protocol, Repetition, or Skew

This section is exclusionary: three controls test whether the boundary is an artifact of protocol, repetition, or skew rather than a property of coverage. *First*, the informative control is a *protocol change*: re-running the scan under the without-replacement epoch protocol of prior work (Power et al., 2022) (the regime in which dataset size enters the training process) reproduces the threshold ($c^* = 23.0\%$ vs. 24.0%; Appendix A.7.1). Repetition is not an independent lever under i.i.d. resampling: each unique pair is drawn with per-step probability $1/(cp^2)$ regardless of multiplicity, so the $R\times$ and $1\times$ factorial arms (Appendix A.7.2) are the same stochastic process by construction, a consistency check, not a test. The epoch protocol is the one setting where duplication is not provably neutral, and there $R = 4$ raises per-pair presentations by 33% and grokking from $7/10$ to $9/10$ ($p = 0.58$), an exposure effect (Appendix A.7.1). *Second*, skewed distributions (Zipf, long-tail) act through reduced *effective* coverage: a uniform control matched to Zipf's effective set fails identically (Appendix A.7.3). A coverage-matched comparison in the transition region ($c \in \{0.20, 0.25, 0.30\}$, 10 seeds/arm) resolves the rest: at 20%, below c^*, neither arm groks ($1/10$ each); at 30% both grok $10/10$, the Zipf arm earlier (median 9k vs. 12k; $p = 0.03$); at 25% Zipf groks in more seeds ($10/10$ vs. $6/10$; Fisher two-sided $p = 0.09$, not significant at 10 seeds/arm). At matched coverage, concentration does not suppress grokking; it can modestly accelerate its timing above the boundary, while coverage remains the variable determining whether reliable formation occurs (Appendix Table 14). *Third*, c^* is hyperparameter-dependent, not universal: it moves from 24.0% at the default settings to 34.9% at weight decay 0.1 and 15.6% at learning rate 3×10^{-3} (Appendix Figure 9, Table 15), while monotone coverage dependence and $10/10$ grokking at 50% hold in every setting.

4.6 Mechanism Analysis: Fourier Circuit Transition

To understand *why* the threshold exists, we analyze the model's Fourier circuit structure across diversity levels (Elhage et al., 2021; 2022). Figure 3 shows the unembedding Fourier entropy (Section 3.5) together with final test accuracy.

The Fourier entropy stays near its random-init value at 12.5–15%, drops sharply across the threshold ($0.92 \to 0.75 \to 0.56$ from 20% to 30%), and stays low through 100% (Figure 3).

Tracking entropy and the *restricted* (training) vs. *excluded* (unseen) losses per step (Figure 10, Appendix A.8) relates focusing to grokking process-wise: at 30%, entropy falls from $\sim$0.999 to $\sim$0.56 as excluded loss collapses; at 15%, entropy barely moves while excluded loss *rises*, a confident lookup table on unseen pairs; at 25%, the two groups separate cleanly.

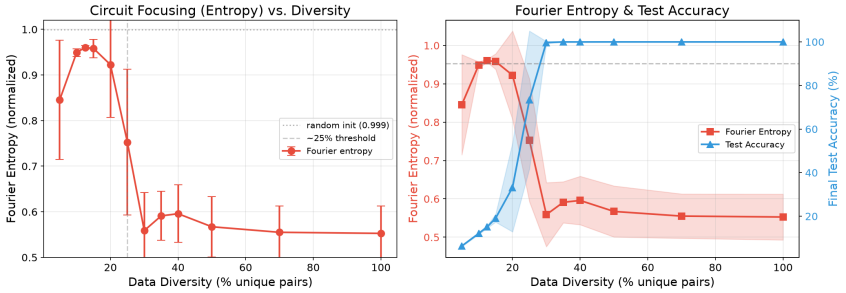

Figure 3: Fourier entropy (circuit focusing). (Left) Entropy vs. diversity, anchored to the random-init baseline (0.999). (Right) Entropy (red) and final test accuracy (blue); error bars/shading show ± 1 std over 10 seeds.

Read per seed rather than as a mean curve, entropy is a *predictive* order parameter and not merely a restatement of coverage: at 25% the seeds that will grok fall below every seed that will not from step $5{,}000$ onward (AUROC 1.0, complete separation of 6 vs. 4 seeds; $p = 0.0095$ is the smallest attainable at that ratio, Appendix A.8), thousands of steps before accuracy reveals the outcome, and at a *fixed* 20% coverage, changing only the learning rate moves the outcome from $0/10$ to $1/10$ to $10/10$ grokked with the mean final entropy following from 0.97 to 0.92 to 0.60 (Appendix A.8). The lead is confined to the *onset*: the first 1–3k steps precede the accuracy rise at 30%, but through the bulk the two move together at our 1k-step resolution. The support entropy measures is load-bearing: zeroing the top five readout pairs at step 10k drives held-out accuracy to 1.4% from noop's 54.1% ($p = 0.0002$) and delays median grokking from 10k to 18k, while freezing them, or zeroing random pairs, costs nothing ($p = 0.79$, 0.12; Appendix A.9). The intervention is on the readout, not on the process that lowers entropy, so a common cause is not excluded, but the support is not epiphenomenal. Entropy is the small-model analogue of the synchronization measure of Sivasankar (2026) and marks where coverage gates circuit formation: below c^* it stays pinned near random init (0.96 at 15%; 0.92 at 20%; Table 5), so there is no circuit for weight decay to liberate (Section 5.1).

5 Discussion

5.1 Why Does Coverage Matter for Grokking?

Our results suggest grokking requires a minimum "critical mass" of unique examples to discover the Fourier circuit (Beck et al., 2025): below the threshold the model memorizes via a lookup table that weight decay (Power et al., 2022) cannot escape, and once enough unique examples make the generalizing circuit more weight-efficient than memorization, grokking proceeds as circuit amplification (Nanda et al., 2023; Varma et al., 2023). Formally, this is a competition between a memorizing circuit and a generalizing Fourier circuit under weight decay: the generalizing circuit is more weight-efficient but becomes selectable only when it is also *feasible* on the training data: below c^* the seen pairs do not span enough of the group structure to recover it; above c^* weight decay amplifies it. Our measured $c^* \approx 24\%$ ($1{,}077$ unique pairs for $p = 67$) sits well above the $O(p \log p)$ bound of Tian (2025) (≈ 280 examples). The gap is expected: such bounds assume i.i.d. samples from the group, whereas grokking is gated by *unique coverage*.

This competition predicts the memorization state below c^* is *absorbing*: the memorizing solution is a stable fixed point that additional training cannot escape, and our trajectory evidence agrees (never-grokked seeds at 25% stay in a distinct high-entropy state with *rising* excluded loss), consistent with the formation-only intervention of Section 4.2. The asymmetry is the same *two-phase* structure recent work identifies in the circuit's temporal dynamics: an early, coverage-gated *formation* phase in which the generalizing circuit condenses out of the memorizing solution, and a later *readout* phase in which weight decay amplifies the already-formed circuit (Sivasankar, 2026). Grokking is delayed relative to formation, and the formed representation is robust to the cuts that would have prevented it.

Our factorial and intervention results align with the *conditional* picture reported for a fully tractable 12K-parameter network (Ootani, 2026), in which coverage rather than sample count sets whether grokking occurs. Our design isolates the same conditionality while holding the task family and repetition fixed (Section 4.5): the same unique-pair coverage governs grokking across protocols and sampling distributions.

5.2 Why Does $c^* \approx N_{\text{EFF}}/p$?

The measured exponent $\alpha = 0.36$ (Section 4.3) is consistent with a circuit-efficiency interpretation of the measured Fourier structure up to an order-one constant. Let H be the final Fourier entropy at full coverage and $N_{\text{eff}} = p^H$ the effective number of frequencies the trained circuit concentrates on (Section 3.5). Under the circuit-efficiency competition of Section 5.3, the generalizing circuit, a sparse combination of $\sim N_{\text{eff}}$ frequencies with one coefficient per output residue per mode ($N_{\text{eff}}\, p$ in total), is selected once that complexity drops below memorization's $C_{\text{mem}} \propto c\, p^2$, fixing the threshold $c^* \approx \kappa\, N_{\text{eff}}/p \propto p^{H-1}$, i.e. $\alpha_{\text{pred}} = 1 - H$. Measured at full coverage H is not constant across the six moduli (0.55 to 0.69, not monotone), so $\alpha_{\text{pred}} = 1 - \bar{H} = 0.402$ is the count at the average H: a check that the $N_{\text{eff}}\, p$ scale is not off by a structural factor, not a derivation of the decay, since with κ held fixed the same count is all but flat in p (fitted $\alpha = -0.03$) and the measured exponent is carried by κ falling from 1.61 to 0.87 (order-one throughout; Appendix A.10). Two limits belong with that. The frequency count does *not* keep growing sub-linearly: $N_{\text{eff}} \propto p^{0.66}$ over the four small moduli but $\propto p^{1.04}$ over all six, so the sub-linear reading holds only over the small-modulus range. And leave-one-modulus-out prediction reproduces the held-out c^*s to 0.74 pp MAE along the exponent route but degrades to 4.1 pp along the N_{eff} route, whose premise fails over the full range (Appendix A.10.3).

Caveats. H is measured at full coverage; near the threshold the representation is less focused ($H \approx 0.75$ at 25%), so full-coverage N_{eff} lower-bounds the complexity organized at c^*. The counting fixes $\kappa \gtrsim 1$; the measured order-one κ is a dynamical overhead, not a sample-counting constant. The $N_{\text{eff}}\, p$ scale bounds the *representation* the optimizer must form, not the input–output map (a linear softmax fit to the same diagonal samples already reaches 90% held-out at $c \approx 2/p$); the empirical threshold thus reflects the memorization↔circuit dynamics, not identifiability (Appendix A.10).

5.3 Resolving the Coverage Confound

The identification comes from the bidirectional interventions of Section 4.2, which flip the outcome with task family, optimizer and arithmetic held fixed ($9/10$ vs. $0/10$ on injection, $10/10$ on removal). The relevant control parameter is the *per-pair exposure*, $E = \text{steps} \times B/(c\, p^2)$: coverage c sets the pair pool, steps and batch size B set the numerator, and R enters neither, so coverage acts through the *span* of the pair pool, not the count of samples. More steps increase E and rescue near-threshold seeds; R does nothing (Section 4.5).

Why should *span* be operative rather than sample count? A memorizing lookup table must store one entry per pair it is ever scored on, so its cost grows with *total samples*; the generalizing circuit is a function of the *set* of residue pairs it must interpolate, so its feasibility is set by coverage. Exposure is necessary but not sufficient: the 15% arm at $p = 67$ gets $10\times$ the exposure of the 30% arm and groks in $0/10$ seeds against $10/10$, and a matched-exposure band across moduli groks $10/10$ down to $c/c^* = 1.29$ and $0/10$ at 0.86 (Appendix A.13). Coverage gates *whether* grokking can happen; exposure gates *whether it does* (Appendix A.12).

5.4 Implications for Data Curation

Where unique examples are well-defined (table-structured and algorithmic tasks), maximizing coverage rather than total count best promotes generalization. The harm of skew is the coverage collapse it induces, not concentrated frequency (Section 4.5). The same asymmetry appears in a real-world setting: Abramov et al. (2025) find that multi-hop QA generalizes once the ratio of inferred to atomic facts clears a threshold, and that even factually *incorrect* synthetic data helps, forcing reliance on relational rather than memorized structure.

5.5 Limitations and Future Work

First, our evidence is confined to *small models on modular arithmetic*: one- and two-layer transformers of a single width ($d_{\text{model}} = 128$; $\sim$216K and $\sim$414K parameters) under one training recipe, so the behavior of c^* at larger scale or on other task families is untested and we claim no transfer; scaling up is the clearest future work (Appendix A.6.1). Theoretical-bound comparisons carry assumptions: Tian (2025)'s $O(M \log M)$ assumes i.i.d. samples, so the $\approx$ 3–5$\times$ gap in Section 5.1 is not like-for-like, and Varma et al. (2023)'s constant-proportion prediction is tested only in scaling form. Second, c^* ranges over 15.6%–34.9% across settings, with monotone dependence and $10/10$ grokking at 50% throughout (Appendix Table 15). Third, our 12K-parameter comparison points to Ootani (2026), whose tractable many-seed setup explores the coverage axis exhaustively where our $\sim$216K-parameter runs only sample it; combining the two would test whether the hysteretic boundary survives at the smaller scale, and whether its location still tracks the output cardinality there. Fourth, the scans run at a fixed step budget, so c^* and the exponent are contours of $100{,}000$ steps: five times the budget rescues stalled seeds at 20% and 25% coverage but none at 15% (Appendix A.14). Because coverage and per-pair exposure covary inversely across the grid, the matched-exposure reanalysis of Appendix A.13 rules out exposure as a *sufficient* variable but is a re-reading of that grid: a control pinning per-pair exposure and varying coverage only, and a multi-modulus budget scan, *have not been run*.

Several concrete questions follow from the formation–maintenance asymmetry. The removal arm shows that cuts below c^* are harmless *after* formation at step 50k; whether a removal at step 10k, when the circuit is only partly formed, is still tolerated, and whether the safe-removal time tracks the entropy collapse of Section 3.5 or the accuracy rise, is open. Second, whether hysteresis is graded or all-or-nothing in the *margin* is open: shallower cuts than our $40\% \to 20\%$ would reveal how robust the formed circuit is. Third, the swap holds pair *identity* fixed (nested subsets, Appendix A.3); re-randomizing which pairs are added would test whether formation needs new pairs from the same structural family or merely that the pool reach c^*. Fourth, the removal arm used the default weight decay, so re-running it across weight decays would separate the maintenance role of coverage from that of the regularizer (lower decay raises c^*; Section 4.5). Fifth, the ablation of Appendix A.9 masks the readout only, leaving the residual stream free to reroute those frequencies, so it tests the readout's frequency support rather than the channel itself, and every arm still groks; whether a full-channel ablation or a graded dose–response over the number of masked pairs would separate necessity from timing is open.

6 Conclusion

Unique-pair coverage is a causal, hysteretic boundary for the formation, but not the maintenance, of the Fourier circuit that enables grokking, and common data-scaling studies vary it implicitly. Crossing c^* upward mid-training at step 25k triggers grokking in $9/10$ seeds (vs. $0/10$ control); the same injection at step 50k rescues $5/5$ at a fixed learning rate (the cosine $3/5$ is explained by learning-rate decay in the fixed-LR control); and cutting coverage below c^* after formation leaves the circuit intact in $10/10$ seeds under both the cosine schedule and a constant learning rate (Section 4.2). Three questions then remain, namely where the boundary is, what crossing it changes, and what it is not:

1. *Where the boundary is.* Grokking depends monotonically on coverage, with a critical threshold $c^* \approx 24\%$ of unique pairs within a budget of $100{,}000$ steps, below which it does not reliably occur; under the held-out criterion $c^*(p) \propto p^{-0.36}$, so the absolute threshold scales as $p^{1.64}$, an intermediate scaling over the tested range.
2. *What crossing it changes.* Above the threshold the representation collapses onto a sparse Fourier circuit (Section 3.5), and the modulus scaling is consistent with a circuit-efficiency count, $c^* \approx \kappa\, N_{\text{eff}}/p$: predicted exponent 0.402 vs. measured 0.359, a consistency check rather than an independent derivation (Section 5.2); removing that support mid-training delays grokking, though every seed rebuilds it (Appendix A.9).
3. *What it is not.* It reproduces under the without-replacement epoch protocol of prior work ($c^* = 23.0\%$ vs. 24.0%; Appendix A.7.1), skewed sampling does not suppress it at matched coverage, and the same critical range reproduces in a second task and a two-layer transformer.

AI USE STATEMENT

In this work, we used generative AI tools for code development (implementing, debugging and running the training and analysis scripts) and for editing the manuscript to improve readability and structure. We have reviewed all AI-assisted work and verified that the released code reproduces the reported numbers. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work trains small transformers from scratch on synthetic modular-arithmetic tasks. It involves no human subjects, no personally identifiable or user-generated data, and no release of any dataset beyond the generated task definitions and the per-seed result files described in the reproducibility statement. All experiments run on a single consumer GPU, for roughly 175 GPU-hours in total. We foresee no harmful applications of the findings, and the authors declare no competing interests.

REPRODUCIBILITY STATEMENT

Code and data to reproduce all experiments and figures are available at `https://anonymous.4open.science/r/Coverage-gates-grokking-432D/`. The repository contains the training and analysis scripts, the per-seed result files behind every table and figure, and a single command that regenerates all figures; Appendix A gives the model, optimization and evaluation protocol in full, with hyperparameters in Table 1 and the held-out criterion in Appendix A.1.2.

ACKNOWLEDGMENTS

We thank the maintainers of the open-source software this work builds on (PyTorch, NumPy, SciPy and Matplotlib) and the providers of the computational resources used for the experiments, whose support made this study possible.

REFERENCES

Roman Abramov, Felix Steinbauer, and Gjergji Kasneci. Grokking in the wild: Data augmentation for real-world multi-hop reasoning with transformers. In *International Conference on Machine Learning (ICML)*, volume 267, 2025.

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

Alon Beck, Noam Itzhak Levi, and Yohai Bar-Sinai. Grokking at the edge of linear separability. In *International Conference on Machine Learning (ICML)*, volume 267, 2025.

Marcel Tomàs Bernal, Neil Rohit Mallinar, and Mikhail Belkin. Breaking data symmetry is needed for generalization in feature learning kernels. *arXiv preprint arXiv:2604.00316*, 2026.

Etienne Boursier, Scott Pesme, and Radu-Alexandru Dragomir. A theoretical framework for grokking: Interpolation followed by Riemannian norm minimisation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Breno W. Carvalho, Artur S. d'Avila Garcez, Luís C. Lamb, and Emilio Vital Brazil. Grokking explained: A statistical phenomenon. *arXiv preprint arXiv:2502.01774*, 2025.

Bilal Chughtai, Alan Cooney, and Neel Nanda. Summing up the facts: Additive mechanisms behind factual recall in LLMs. *arXiv preprint arXiv:2402.07321*, 2024.

Benjamin L. Edelman, Surbhi Goel, Sham M. Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning (ICML)*, volume 162, pp. 5793–5831, 2022.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger B. Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2021.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *International Conference on Learning Representations (ICLR)*, 2024.

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.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations (ICLR)*, 2023.

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 *International Conference on Machine Learning (ICML)*, volume 267, 2025.

Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. Why do you grok? A theoretical analysis on grokking modular addition. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 35934–35967, 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.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Alex Nguyen and Gautam Reddy. Differential learning kinetics govern the transition from memorization to generalization during in-context learning. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Yoshiyuki Ootani. Grokking is conditional and fragile: A fully-tractable, multi-seed study at 12k parameters. *arXiv preprint arXiv:2607.05104*, 2026.

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.

Lucas Prieto, Melih Barsbey, Pedro Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

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

Vaibhav Singh, Eugene Belilovsky, and Rahaf Aljundi. When data falls short: Grokking below the critical threshold. *arXiv preprint arXiv:2511.04760*, 2025.

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

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)*, 2020.

Yuandong Tian. Provable scaling laws of feature emergence from learning dynamics of grokking. *arXiv preprint arXiv:2509.21519*, 2025.

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)*, pp. 5998–6008, 2017.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*, 2023.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. doi: 10.1145/3446776.

Xiaotian Zhang, Yue Shang, Entao Yang, and Ge Zhang. Is grokking a computational glass relaxation? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Additional Results and Experimental Details

This appendix mirrors the structure of the main text rather than grouping material by format: figures and tables appear inside the sections they support, each introduced by a short paragraph in the body. We first give the experimental setup and the held-out evaluation protocol, together with the tables that validate that protocol; then, for each result of the main text in order, the figure and detailed tables that support it, and the controls that separate the coverage effect from protocol, repetition, and skew. We then expand the derivation behind the circuit-efficiency count and close with three short theoretical readings of the main findings.

A.1 Experimental Setup and Evaluation Protocol

A.1.1 Model Architecture and Training Hyperparameters

The one-layer transformer used in all experiments has the following configuration:

Table 1: Model architecture and training hyperparameters.

Parameter	Value
Number of layers	1
Embedding dimension (d_{model})	128
Number of attention heads	4
Head dimension	32
Feed-forward dimension (d_{ff})	512
Activation function	ReLU
Positional encoding	Learned (3 positions)
Total parameters	$\sim$216K
Optimizer	AdamW
Learning rate	10^{-3} (default; varied in Section 4.5)
Weight decay	1.0 (default; varied in Section 4.5)
Batch size	512
Training steps	100,000
Learning rate schedule	Cosine (no warmup)
Evaluation interval	Every 1,000 steps
Loss function	Cross-entropy

All experiments were conducted on a single machine with an NVIDIA RTX 4060 Laptop GPU (8 GB VRAM), 14th Gen Intel Core i9-14900H CPU, and 32 GB RAM, running PyTorch 2.5.1 with CUDA 12.1 and Python 3.11. Each training run (100,000 steps, eval every 1,000 steps) takes approximately 5–30 minutes depending on the architecture (the 500,000-step absorption runs take about $5\times$ longer); total compute time was approximately 175 GPU-hours (about 7 days on a single GPU) across all 1,136 training runs.

A.1.2 The Held-Out Metric and Its Validation

Section 3.3 defines the primary metric, held-out-only accuracy, via the linear decomposition: because both accuracies count correct pairs, the reported total accuracy at coverage c decomposes linearly into its training and held-out components,

$$a_{\text{tot}} = c\, a_{\text{train}} + (1 - c)\, a_{\text{held-out}},$$

and since training accuracy reaches $\sim$100% shortly after memorization, the held-out-only accuracy is recovered exactly from the stored per-step curves, $a_{\text{held-out}} = (a_{\text{tot}} - c\, a_{\text{train}})/(1 - c)$. This zero-cost decomposition is the paper's primary evaluation and isolates genuine generalization from memorized training pairs. We confirm the decomposition by retraining the dense scan at $p = 67$ under held-out-only evaluation (training pairs excluded from the test set): the retrained held-out curves match the decomposition to within 0.4 pp at every coverage level (15%–50%), and the number of seeds reaching 90% held-out accuracy agrees exactly (Table 2).

Table 2 reports, for each modulus and coverage level, the number of seeds reaching 90% under the original (training-inclusive) criterion and under the held-out-only criterion, together with the number reaching 99% held-out accuracy, grokking to near-certainty on unseen pairs. Four findings:

1. For $p \geq 67$, c^* is essentially unchanged under the held-out criterion (24.0%, 20.1%, 19.8% at $p = 67, 97, 113$): the transition is a genuine generalization transition, not an artifact of training-pair inclusion.
2. At $p = 47$, c^* moves from 25.1% to 28.1% (+2.9 pp): near the threshold, several seeds plateau at 90–99% total accuracy that the held-out criterion re-classifies as partial gener-

alization: they grok the training pairs and part of the unseen pairs, but not all. At 30% coverage, e.g., 10/10 seeds exceed 90% total accuracy but only 5/10 exceed 90% held-out and 1/10 exceed 99% held-out.

3. The power-law exponent under the primary held-out criterion is $\alpha = 0.36$ ($D_{\text{crit}} \propto p^{1.64}$; the six-modulus fit is in Section 4.3); including training pairs in the test set flattens it to $\alpha = 0.27$ ($D_{\text{crit}} \propto p^{1.73}$). The bracketing conclusion, that the absolute threshold scales sub-quadratically but super-linearly, is therefore robust to the evaluation protocol, and the held-out estimate is if anything steeper.
4. At low coverage, the decomposition quantifies how much of the reported accuracy is memorization: held-out accuracy is at chance ($\approx 1.4\%$ at 5% coverage vs. chance $1/67 \approx 1.5\%$), barely above it ($\approx 2.3\%$ at 10%, $\approx 3.0\%$ at 12.5%), and only clearly exceeds chance from 15% onward ($\approx 4.8\%$ at 15%, $\approx 16.5\%$ at 20%). The apparent partial generalization at 5–10% coverage (Section 4.1) is therefore largely memorized training pairs being scored as correct.

Finally, at 100% coverage the test set coincides with the training set, so $a_{\text{held-out}}$ is undefined and the ∼1,000-step grokking step there measures memorization speed; held-out re-analysis is only meaningful for $c < 1$.

Table 2: Held-out re-analysis of the dense scans (10 seeds per condition; column definitions in the text). c^* is the binomial-MLE logistic estimate on the shared coverage grid, with 100% excluded (held-out undefined there); $\Delta = c^*_{\text{held-out}} - c^*_{\text{total}}$.

p	Coverage	# ≥ 90% total	# ≥ 90% held-out	# ≥ 99% held-out	c^* total	c^* held-out	Δ
47	10%	0/10	0/10	0/10	—	—	—
47	15%	0/10	0/10	0/10	—	—	—
47	20%	0/10	0/10	0/10	—	—	—
47	25%	4/10	4/10	0/10	—	—	—
47	30%	10/10	5/10	1/10	—	—	—
47	35%	10/10	10/10	5/10	—	—	—
47	50%	10/10	10/10	10/10	—	—	—
47	c^*	—	—	—	25.1%	28.1%	+2.9%
67	10%	0/10	0/10	0/10	—	—	—
67	15%	0/10	0/10	0/10	—	—	—
67	20%	1/10	1/10	0/10	—	—	—
67	25%	6/10	6/10	4/10	—	—	—
67	30%	10/10	10/10	10/10	—	—	—
67	35%	10/10	10/10	10/10	—	—	—
67	50%	10/10	10/10	10/10	—	—	—
67	c^*	—	—	—	24.0%	24.0%	+0.0%
97	10%	0/10	0/10	0/10	—	—	—
97	15%	0/10	0/10	0/10	—	—	—
97	20%	4/10	4/10	1/10	—	—	—
97	25%	10/10	10/10	10/10	—	—	—
97	30%	10/10	10/10	10/10	—	—	—
97	35%	10/10	10/10	10/10	—	—	—
97	50%	10/10	10/10	10/10	—	—	—
97	c^*	—	—	—	20.1%	20.1%	+0.0%
113	10%	0/10	0/10	0/10	—	—	—
113	15%	0/10	0/10	0/10	—	—	—
113	20%	7/10	7/10	7/10	—	—	—
113	25%	10/10	10/10	10/10	—	—	—
113	30%	10/10	10/10	10/10	—	—	—
113	35%	10/10	10/10	10/10	—	—	—
113	50%	10/10	10/10	10/10	—	—	—
113	c^*	—	—	—	19.8%	19.8%	+0.0%

Because the held-out accuracy is the primary criterion (Section 3.3), we also check how sensitive the located threshold is to the success threshold used to declare grokking on it. Table 3 re-fits c^*

on the shared 7-point held-out grid, declaring success at held-out thresholds from 80% to 99%. For $p \geq 67$ the estimate moves by at most 1.1 pp across the whole window (24.0%–25.1% at $p = 67$; 20.1%–20.7% at $p = 97$; a constant 19.8% at $p = 113$). Only at $p = 47$ does the strictest criterion move it materially, from 28.1% (at 90%) to 34.9% (at 99%): at the smallest modulus, near-threshold seeds generalize to 90–95% held-out but not to near-perfect accuracy.

Table 3: Sensitivity of c^* to the held-out grokking criterion: each entry re-fits c^* (binomial-MLE logistic, shared 7-point held-out grid) with success defined at that threshold.

Criterion	$p = 47$	$p = 67$	$p = 97$	$p = 113$
80%	25.0%	24.0%	20.1%	19.8%
85%	25.1%	24.0%	20.1%	19.8%
90%	28.1%	24.0%	20.1%	19.8%
95%	31.2%	25.0%	20.3%	19.8%
99%	34.9%	25.1%	20.7%	19.8%

Table 4 shows the per-seed distribution of the *best reported* (training-inclusive) test accuracy that produces this asymmetry. At $p = 47$ and 25% coverage only $4/10$ seeds ever exceed 90% and none exceed 99% (best seed 98.0%); at 30% all 10 seeds clear 90%, yet only $5/10$ reach 95% and $1/10$ reach 99%: the grokked seeds plateau at 90.2–99.5% total, which is exactly why the held-out criterion of Table 2 reclassifies most of them as partial generalizers (finding (2) above). At $p = 67$, by contrast, $5/6$ of the grokked seeds at 25% exceed 99%, so the re-analysis leaves c^* essentially unchanged there.

Table 4: Per-seed best reported (training-inclusive) test accuracy (percent) at near-threshold coverages (10 seeds per cell), explaining the $p = 47$ criterion sensitivity of Table 3 (see text).

p	Coverage	Best-acc. range	# ≥ 90%	# ≥ 95%	# ≥ 99%
47	20%	28.4–85.6	0	0	0
47	25%	36.8–98.0	4	3	0
47	30%	90.2–99.5	10	5	1
67	20%	23.4–95.6	1	1	0
67	25%	28.5–100.0	6	5	5
67	30%	99.4–100.0	10	10	10

A.2 A Probabilistic Critical Coverage Threshold

Figure 4 analyzes the dense scan in the threshold region under the held-out criterion. The left panel gives the Kaplan–Meier median held-out grokking step at each coverage level, censoring never-grokked seeds; the right panel gives the proportion of seeds that never grok, with 95% Wilson CIs. Below 20% grokking essentially never happens: all 10 seeds never-grok at 5–15% coverage. At 20% a single seed groks ($9/10$ never-grok), so the median step is still undefined; grokking becomes reliable only from 30% upward, where all 10 seeds grok and the median step falls from 12k at 30% to ∼1k at 70%. The boundary is therefore a *soft band*: 25% coverage already yields a defined median (37k) but leaves $4/10$ seeds never grokked, so $c^* \approx 24\%$ is where grokking *reliably* occurs, not a hard per-seed cutoff.

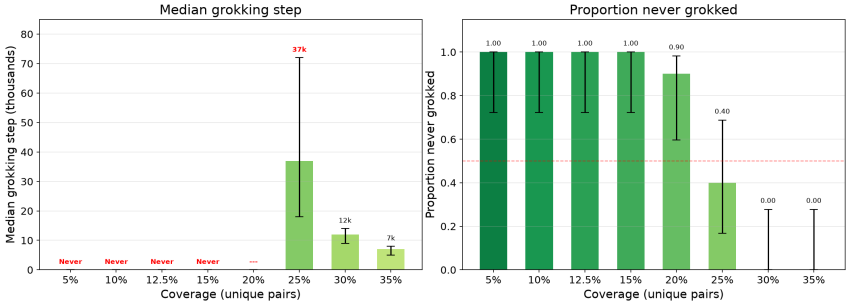

Figure 4: Threshold analysis of the dense scan. Left: Kaplan–Meier median held-out grokking step vs. coverage (never-grokked seeds censored); right: proportion of seeds that never grok, with 95% Wilson CIs.

Table 5 reports the dense scan in full (10 seeds per condition, training pairs removed from the test set). "Grokking step" is the Kaplan–Meier median of the per-seed first step at which held-out accuracy exceeds 90% (95% bootstrap CI in brackets; never-grokked seeds censored); "Never" counts seeds that fail to grok within 100,000 steps; entropy (Ent) is the normalized Shannon entropy of the unembedding layer's Fourier-coefficient distribution (Section 3.5), so lower entropy means a more focused circuit. No coverage below 25% has a defined median step (all seeds never-grok at $\leq 15\%$; $9/10$ at 20%). The entropy column shows the same transition from the representation side: it sits in the 0.85–0.96 range through 15% (close to the random-init value), dips to 0.92 at 20% and 0.75 at 25%, and settles at 0.55–0.60 for $c \geq 30\%$. At 100% coverage the test set coincides with the training set, so the held-out step is undefined (—); the row reports the training-inclusive behavior (step 1.0k, no seed ever failing to grok), which is a memorization-speed floor, not generalization.

Table 5: Dense diversity scan under the held-out criterion (10 seeds per condition; training pairs removed from the test set). Step: Kaplan–Meier median [95% bootstrap CI], never-grokked seeds censored; Never: seeds not grokked within 100,000 steps; Ent: mean Fourier entropy [95% bootstrap CI]. At 100% the held-out step is undefined; the training-inclusive step is 1.0k (memorization floor).

Div.	Unique pairs	Grokking step (KM median [CI])	Never	Ent (mean [CI])
5%	224	—	10/10	0.85 [0.75–0.92]
10%	448	—	10/10	0.95 [0.94–0.95]
12.5%	561	—	10/10	0.96 [0.96–0.96]
15%	673	—	10/10	0.96 [0.95–0.97]
20%	897	—	9/10	0.92 [0.84–0.97]
25%	1,122	37k [18k–72k]	4/10	0.75 [0.66–0.85]
30%	1,346	12k [9k–14k]	0/10	0.56 [0.51–0.61]
35%	1,571	7.0k [5.0k–8.0k]	0/10	0.59 [0.56–0.62]
40%	1,795	5.0k [4.0k–5.0k]	0/10	0.60 [0.56–0.63]
50%	2,244	3.0k [2.0k–3.0k]	0/10	0.57 [0.53–0.61]
70%	3,142	1.0k [1.0k–2.0k]	0/10	0.56 [0.52–0.59]
100%	4,489	—	0/10	0.55 [0.52–0.59]

A.3 The Formation Boundary Is Bidirectional and Causally Controllable

To test whether c^* is a *causally* controllable boundary (Section 4.2), we run eight arms at $p = 67$, using nested subsets: for each seed, one random permutation of the $p^2 = 4{,}489$ ordered pairs (including the p diagonal pairs) is split so that the small set S_{small} (c_{small} fraction) is a subset of the large set S_{big} (c_{big} fraction). *Inject* arms start on S_{small} at $c = 15\%$ (an absorbing state) and swap to S_{big} at $c = 30\%$ (above $c^* \approx 24\%$) at step 25,000 or 50,000; the *control* stays at 15% for all 100,000 steps; the *removal* arm starts on S_{big} at $c = 40\%$ (grokked by $\sim$5k steps) and swaps down to S_{small} at $c = 20\%$ (below c^*) at step 50,000. Every arm is run under both the standard cosine schedule and a fixed learning rate (10^{-3}, no decay) to separate schedule effects from basin deepening. The three core arms (inject@25k, control, removal) use 10 seeds each under the cosine schedule; the @50k arms use 5, as do the inject and control arms at a fixed rate, while the removal arm keeps all 10 there so that the two schedules can be compared seed by seed (Appendix A.3.1). Held-out accuracy is measured directly on test pairs that are never in *any* training phase (the union of the pre- and post-swap training sets), so the swapped-down removal model is judged only on pairs it never trained on: 70% of pairs for the inject/control arms (where the union equals the final set) and 60% for the removal arm, since the 20% removed pairs *were* trained on in the first half of training and are therefore excluded from held-out.

Table 6 reports each arm's outcome: the held-out accuracy just before the swap (pre-swap), its minimum after the swap, the final held-out and test accuracy, the number of seeds that grok, and the grokking steps (for the removal arms, the steps at which seeds remain grokked after the swap). For inject@25k, the single non-grokked seed (held-out $\sim$24%) pulls the final mean below the $\geq$95% that the nine grokked seeds reach. The removal arm's post-swap minimum is the one column that is *not* comparable across schedules (92.8% under cosine, 39.6% at a fixed rate); why, and why the endpoint is nevertheless comparable, is the subject of Appendix A.3.1.

Figure 5 shows the per-seed held-out trajectories behind the four cosine arms (held-out measured on pairs never in any training phase). Injecting from 15% to 30% at step 25k triggers grokking in $9/10$ seeds; the one failing seed memorizes the swapped-in pairs and stays at chance held-out. The same injection at step 50k rescues only a subset under cosine scheduling; the 15% control never

Table 6: Bidirectional coverage intervention ($p = 67$, 100,000 steps). Held-out is measured on pairs never in any training phase (protocol and arm sizes in the text above); "grok" counts seeds whose held-out first exceeds 90% (for the removal arms, seeds still above 90% at the endpoint, the median of the last five evaluations); "post-swap min" averages each seed's own minimum after the swap. At that endpoint the two removal rows read 10/10 under cosine and 9/10 at the fixed rate, while both grok 10/10 in the binary sense. The first four rows use the cosine schedule; the last four repeat the same arms at a fixed learning rate (10^{-3}).

Arm	Pre-swap	Post-swap min	Held-out final	Test	Grok	Steps
Inject @25k (cosine)	3.9%	3.4%	89.7%	92.8%	9/10	26k–37k (×8), 74k
Inject @50k (cosine)	6.4%	5.4%	75.8%	83.1%	3/5	51k, 53k, 72k
Control (cosine)	—	0.6%	4.8%	19.1%	0/10	—
Remove @50k	100.0%	92.8%	99.1%	99.3%	10/10	retained
Inject @25k (fixed LR)	4.0%	3.9%	96.0%	97.0%	5/5	27k–49k
Inject @50k (fixed LR)	5.5%	5.0%	97.7%	98.4%	5/5	53k–87k
Control (fixed LR)	—	0.6%	9.1%	22.5%	0/5	—
Remove @50k (fixed LR)	100.0%	39.6%	96.2%	97.0%	9/10	retained

leaves chance. Removing from 40% to 20% at step 50k leaves every seed grokked, with post-swap held-out recovering to $\geq 95.7\%$ in every seed; the swap step is marked by the dotted vertical line.

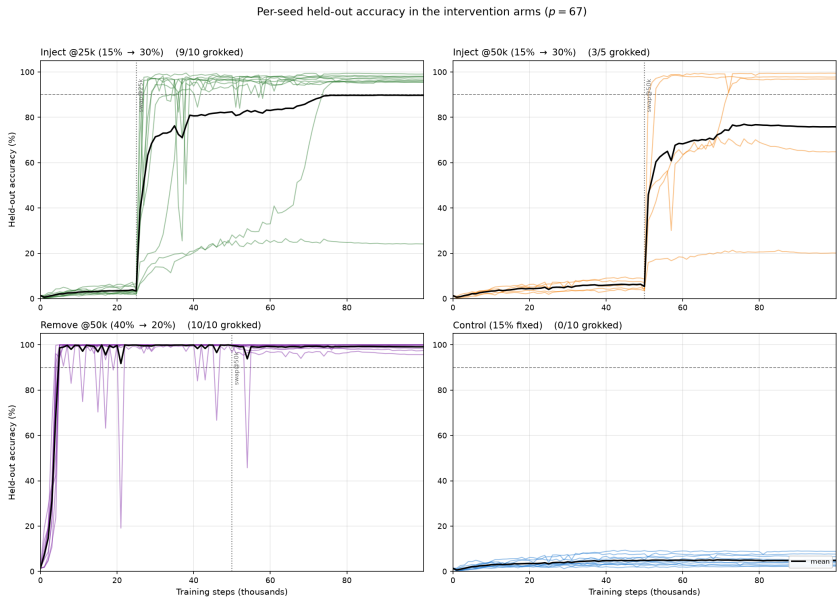

Figure 5: Per-seed held-out accuracy (pairs never in any training phase) in the four cosine arms of Table 6; the swap step is marked by the dotted vertical line.

A.3.1 The Fixed-Learning-Rate Removal Control

The last row of Table 6 is the control that the removal arm was missing. By step 50,000 the cosine rate has already fallen to half its peak, so "10/10 retained" could in principle mean the optimizer was frozen by its own schedule rather than that the circuit is stable. The arm therefore repeats the removal at a constant 10^{-3} with no decay, run on the same 10 seeds. The pairing is exact: within the arm, seed s draws the same nested permutation, the same initialization and the same batch sequence as seed s of the cosine removal arm, so the schedule is the only variable and the two can be compared seed by seed.

The endpoint does not depend on the schedule. Both arms grok again after the swap, 10/10 each. At the endpoint 10/10 seeds remain above the 90% criterion under cosine against 9/10 at the constant rate, the same count under either read of the endpoint (the final evaluation, and the median of the last five); mean held-out is 96.2% and mean test 97.0% at a constant rate against 99.1% and 99.3% under cosine, with every seed's endpoint $\geq 75.8\%$ at a post-swap training pool of 897 pairs, 20%

of the 4,489, below $c^* \approx 24\%$. Circuit strength is marginally *higher* at the fixed rate (paired $+0.13$, better in $9/10$ seeds). A live optimizer, in other words, does not walk the formed circuit back out.

What the schedule controls is the post-swap minimum. That column is not schedule-invariant: 92.8% under cosine against 39.6% at a constant rate (paired mean -53.1 pp). The reason is that under a constant rate the held-out curve is flat at 95–100% but makes isolated single-evaluation excursions: 40 post-swap drops below 90% whose neighbours on both sides are above it, every one of them recovered by the very next evaluation, against 1 under cosine. These are transient collapses of the optimization, not evaluation artifacts: at the same evaluations the training accuracy falls too (below 90% in 18 of the 40, as low as 11%) and the loss rises from 0.02 at the neighbouring evaluations to a median of 0.30 (maximum 3.58), and the next evaluation finds both back at their previous level. They are also not a response to removal, since 17 of them occur *before* the swap at 40% coverage and the same phenomenon appears in the fixed-rate *injection* arms, which remove nothing (21 after the swap at inject@25k, 15 at inject@50k). Cosine shows 9 such excursions while its rate is still high and 1 after the decay has set in. Learning-rate decay therefore quiets the transients; it does not produce the survival. Reading the minimum as evidence of an undisturbed circuit is accordingly a schedule comparison, and only the endpoint carries the maintenance claim. Analysis: `_removal_flr_analysis.py`.

A.4 Threshold Scaling with Modulus p

Figure 6 reports the modulus scan used to estimate $c^*(p)$ under the held-out criterion (training pairs removed from the test set). The left panel plots $c^*(p)$ against the modulus p on log–log axes: each point is the binomial-MLE logistic estimate on that modulus's coverage grid (100% excluded; error bars are 95% bootstrap CIs), the shared 7-point grid extended by four extra points inside the transition for $p = 173$, and the dashed line is the power-law fit over all six moduli, $c^*(p) \propto p^{-0.36}$ ($\alpha = 0.359$, 95% CI $[0.304, 0.427]$, $R^2 = 0.97$), i.e. $D_{\text{crit}} \propto p^{1.64}$. The right panel plots the Kaplan–Meier median held-out grokking step against coverage on the same shared grid, one curve per modulus (never-grokked seeds censored): in the range 20–50% the median step is non-increasing in p at every fixed coverage, and all moduli settle on a common 1–3k floor at 50%: larger moduli grok earlier and at proportionally lower coverage.

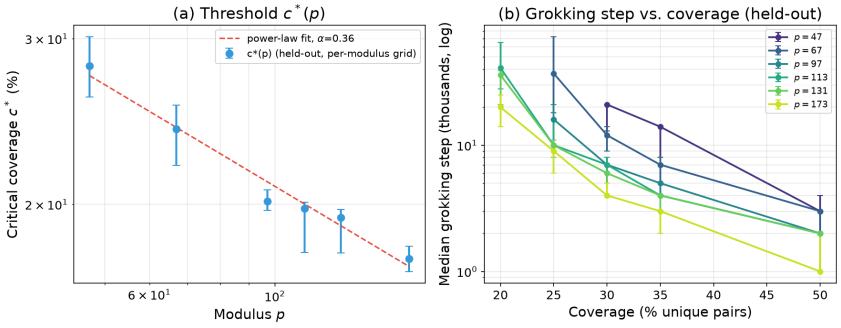

Figure 6: Critical coverage and grokking step across moduli, held-out criterion. Left: $c^*(p)$ vs. modulus p (log–log), fit per modulus on the shared 7-point coverage grid (100% excluded), 95% bootstrap CIs, and the all-six-moduli power-law fit $c^*(p) \propto p^{-0.36}$ (dashed); right: Kaplan–Meier median held-out grokking step vs. coverage on the shared grid (never-grokked seeds censored).

Table 7 lists the per-modulus estimates behind the fits, together with the number of seeds grokking at 25% coverage. c^* falls monotonically from 28.1% at $p = 47$ to 17.5% at $p = 173$; the $p = 257$ entry is an extrapolation scan on a four-point grid $\{0.125, 0.15, 0.20, 0.25\}$ and is *excluded* from every fit below. The fitted exponent depends on how many moduli enter the fit. Over the four moduli of the original scan, $p \in \{47, 67, 97, 113\}$, the fit gives $\alpha = 0.416$ (95% CI $[0.330, 0.514]$, $R^2 = 0.99$); adding $p = 131$ gives $\alpha = 0.378$ ($[0.304, 0.476]$, $R^2 = 0.97$); adding $p = 173$ gives the all-six-modulus value $\alpha = 0.359$ ($[0.304, 0.427]$, $R^2 = 0.97$) used as the headline throughout this paper. The four- and five-modulus values bracket the sensitivity of that headline to which moduli are scanned, and they are not wide: the shift from four to six moduli is 0.06 in α, against bootstrap intervals more than twice as wide. Section A.5 reports the extra scan that produces the $p = 173$ point and isolates how much of the shift it carries.

Table 7: Critical coverage $c^*(p)$ from per-modulus held-out scans (10 seeds per point; shared 7-point grid, 100% excluded). grok@25% = seeds grokking at 25% coverage. $p = 173$ additionally uses four extra grid points inside its transition (Section A.5); $p = 257$ is the out-of-family extrapolation point of Section 4.3 and is excluded from every fit.

p	c^*	95% CI	grok@25%
47	28.1%	[26.0%, 30.1%]	4/10
67	24.0%	[22.0%, 25.5%]	6/10
97	20.1%	[19.7%, 20.7%]	10/10
113	19.8%	[17.8%, 20.1%]	10/10
131	19.4%	[17.7%, 19.7%]	10/10
173	17.5%	[17.0%, 18.0%]	10/10
257	15.3%	[15.0%, 17.6%]	10/10

Figure 7 plots the residual of the power-law fit (observed minus fitted c^*, in percentage points, with 95% CIs from the per-modulus held-out logistic fits). Over the six fitted moduli every residual lies within 1.0 pp (the largest is -1.00 pp at $p = 97$), and their signs alternate with p rather than drifting, so no systematic deviation from a power law appears. The $p = 97$ interval is the only one that excludes zero. The out-of-family point at $p = 257$ sits above the fit and is discussed, together with leave-one-modulus-out cross-validation, in Section 4.3.

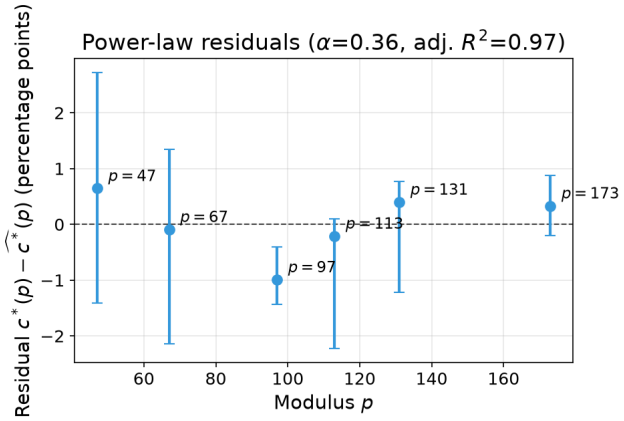

Figure 7: Residuals (percentage points) of the all-six-moduli power-law fit $c^*(p) \propto p^{-0.36}$ (observed minus fitted c^*), with 95% CIs from the per-modulus held-out logistic fits.

A.5 Extending the Scaling Scan: Two New Moduli and One Un-Scanned Point

Section 4.3 reports the scaling fit over six moduli; three of those points were run after the original scan was complete, on the same grids and with the same 10 seeds per point, and the extrapolated predictions were registered before the new runs were analyzed.

Registration of the new moduli. The published four-modulus fit predicts $c^*(131) = 18.2\%$ and $c^*(173) = 16.1\%$. Measured, they are 19.4% (95% CI $[17.7\%, 19.7\%]$) and 17.5% ($[17.0\%, 18.0\%]$). The first prediction lies inside its interval; the second does not: it undershoots by 1.4 pp. Refitting on five moduli, which includes the newly observed $p = 131$, moves the $p = 173$ prediction to 16.9%, a miss of $+0.6$ pp. Both new moduli thus land *above* the extrapolated decay, the same direction as the $p = 257$ point of Appendix A.10.3.

Grid resolution at the new moduli. On the shared 7-point grid the $p = 173$ transition falls entirely inside the interval $(15\%, 20\%)$ (0/10 seeds grok at 15% and 10/10 at 20%), so the logistic fit is saturated there and returns the degenerate interval $[18.1\%, 18.1\%]$, which is a property of the grid rather than a measurement. Adding four points inside the transition ($c \in \{16, 17, 18, 19\}\%$, with counts 1/10, 5/10, 6/10, 8/10) resolves it to $c^*(173) = 17.5\%$ with 95% CI $[17.0\%, 18.0\%]$. The same thing happened at $p = 211$ (below) and at no other modulus.

How much of the exponent rides on the last modulus. The headline exponent moves from $\alpha = 0.416$ over four moduli to 0.378 over five and 0.359 over six. That entire drift is narrower than the sensitivity of the six-modulus fit to its least identified point: holding the other five fixed and moving only $c^*(173)$ across a range wider than its own confidence interval, α runs from 0.441 at 15.0% through 0.359 at 17.5% to 0.288 at 20.0%. The six-modulus value is therefore best read as the current estimate with an additional spread of about 0.07 in either direction from this source alone, on top of the bootstrap interval already quoted.

Functional form. We also asked whether the data prefer a plain power law to one with a logarithmic correction, $\log c^* = a + b \log p + g \log \log p$. Over a factor of $\sim$4 in p, with three parameters fitted to six points, the comparison is decided by the least identified point: both the BIC difference and the fitted g move sharply when $p = 173$ is dropped from the fit. We therefore report the power law as *consistent with* the data, and claim neither that the data prefer it nor that they rule out a correction. Comparing the simplest alternatives on the same six points in native units makes the limits of the test concrete: $R^2 = 0.977$ for the power law, 0.957 for a logarithmic form $c^* = a - b \ln p$, and 0.890 for an exponential. The six points separate the exponential but not the logarithm, so within this range $c^* \propto p^{-0.36}$ describes the span rather than establishing an asymptotic law.

Widening the span. The most direct test of how much the exponent depends on the width of the fitted range is to extend the fit to the one modulus outside it. Adding $p = 257$ raises the span from $3.7\times$ to $5.5\times$ and moves the exponent from 0.359 to $\alpha = 0.348$: a shift of 0.011, an order of magnitude inside the bootstrap interval. Two caveats keep this a robustness check rather than the headline. The fit is mixed-grid, since $p = 257$ contributes four coverage points where every other modulus contributes seven; and $p = 257$ is the point this paper uses as a never-fitted out-of-family check (Section 4.3), so folding it into the fit spends that check to buy an n of seven. We therefore keep six moduli as the headline and record this as a bound on the range-dependence. Analysis: `_alpha_power.py`.

An interpolation check at the un-scanned modulus. To test the fit where it had never been measured, we pre-registered a prediction at $p = 211$ from the five-modulus fit (the value available at the time, $\alpha = 0.378$), giving $c^*(211) = 15.7\%$, with the decision rule fixed in advance as *confirm* if the transition lands in $[14\%, 17\%]$ and *refute* if below. The first scan reproduced the $p = 173$ grid problem ($0/10$ at 15%, $10/10$ at 20%), so we added points at $14, 16, 17, 18\%$ and obtain $c^*(211) = 16.1\%$ (95% CI $[15.7\%, 16.5\%]$). That confirms the pre-registered band (weakly, and we say so): the four-modulus fit ($\alpha = 0.416$) predicts 15.0% and the six-modulus fit ($\alpha = 0.359$) predicts 16.4%, so the measured value falls between the two, nearer the shallower one, and 10 seeds per level do not separate them.

A.6 Generalization Across Tasks and Architectures

Table 8 reports the cross-task and cross-architecture results behind Section 4.4. Modular multiplication ($p = 67$, one-layer transformer) reproduces the critical range: no seed groks at 20%, $4/10$ grok at 25% (median 43k), and all 10 grok at 30% (median 13k). A two-layer transformer of the same width on modular addition groks earlier and more reliably than the one-layer model: $10/10$ seeds at 25% coverage with median step 6.0k, versus $6/10$ and 37k for the one-layer model. As in the dense scan, the held-out step is undefined at 100% coverage (the test set equals the training set); the training-inclusive step there is 1.0k, a memorization floor.

A.6.1 Scope and Scale

All claims in this paper are bounded by the following. *Scale*: our largest model is a two-layer transformer with $\sim$414K parameters ($\sim$463K at the largest modulus tested), and the one-layer model behind the primary results has $\sim$216K ($\sim$265K at $p = 257$); we did not train wider or deeper models, and we did not scale the data beyond the p^2 pair space of the moduli studied. *Task family*: every task is modular arithmetic, addition or multiplication; the coverage mechanism is stated for tasks with a well-defined set of unique examples and is not tested on language or real-world data. *Hyperparameters*: c^* moves with weight decay and learning rate (Appendix A.7.5) while the monotone coverage dependence and $10/10$ grokking at 50% are stable in every condition, but the joint weight-decay$\times$learning-rate surface is unmapped. The threshold should therefore be read as a property of this controlled setting rather than as a universal constant: whether an analogous feasibility boundary organizes generalization in larger models or in other task families is an open empirical question.

Table 8: Generalization results (10 seeds per condition) under the held-out criterion (training pairs removed from the test set). Step: Kaplan–Meier median [95% bootstrap CI], never-grokked seeds censored; Never: seeds not grokked within 100,000 steps. Multiplication uses a one-layer model; "2-layer" uses modular addition.

Condition	Div.	Grokking step (KM median [CI])	Never
Modular multiplication (cross-task)			
1-layer	20%	—	10/10
1-layer	25%	43.0k [17.0k–45.0k]	4/10
1-layer	30%	13.0k [9.0k–15.0k]	0/10
1-layer	100%	—	0/10
Two-layer transformer (cross-architecture)			
2-layer	25%	6.0k [5.0k–6.0k]	0/10
2-layer	100%	—	0/10

A.7 The Boundary Is Not an Artifact of Protocol, Repetition, or Skew

A.7.1 The Epoch-Protocol Control

Prior grokking studies train on a fixed pair set reshuffled once per epoch, drawing batches without replacement (Power et al., 2022); our main scheme instead redraws every batch i.i.d. with replacement. Under the latter, repetition is not an independent lever (Section 4.5): the $R\times$ and $1\times$ arms of Table 11 are the same stochastic process by construction. The protocol in which total dataset size actually enters training is the epoch one, so we rerun the $p = 67$ scan on the shared coverage grid $\{15, 20, 25, 30, 50\}\%$ (10 seeds per level, 100,000 steps, weight decay 1.0) under the epoch protocol of Power et al. (2022). Table 9 compares the two protocols. The held-out critical coverage is $c^* = 23.0\%$ (95% CI $[20.6\%, 25.0\%]$) under the epoch protocol vs. 24.0% ($[22.0\%, 25.4\%]$) under i.i.d. resampling (both re-fitted on this five-point grid, which is why the interval here is not character-for-character the one in Table 7, $[22.0\%, 25.5\%]$ on the seven-point grid), and the per-level grokked counts agree within seed noise at every level (e.g. $7/10$ vs. $6/10$ at 25%). The protocol change therefore does not move the threshold: coverage gates grokking under both sampling schemes, locating the effect in unique coverage rather than in the sampling process.

Table 9: Epoch-protocol control ($p = 67$, held-out criterion). "Epoch": fixed pair set, one shuffle per epoch, batches without replacement (Power et al., 2022); "i.i.d.": every batch redrawn with replacement. Grokked: seeds whose held-out first exceeds 90% within 100,000 steps; step: Kaplan–Meier median [95% bootstrap CI], never-grokked seeds censored.

Coverage	Epoch grokked	i.i.d. grokked	Epoch step (KM [CI])	i.i.d. step (KM [CI])
15%	0/10	0/10	—	—
20%	2/10	1/10	—	—
25%	7/10	6/10	28.0k [23k–47k]	37.0k [18k–72k]
30%	10/10	10/10	10.0k [9k–12k]	12.0k [9k–14k]
50%	10/10	10/10	3.0k [2k–3k]	3.0k [2k–3k]

The epoch protocol is also the one setting in which repetition is *not* provably inert, so we test it there. "Repetition" is implemented as in the factorial design below (the training pool becomes R copies of the unique set, reshuffled and drawn without replacement), and because the pool is R times longer, so is each epoch. The naive cancellation is not exact: the harness truncates the last batch of every epoch, so a longer pool wastes a smaller fraction of the fixed 100,000-step budget. Counting draws exactly at batch 512, $R = 4$ raises the presentations per unique pair from 33,333 to 44,444 at $c = 25\%$ (+33%) and from 20,000 to 22,222 at $c = 50\%$ (+11%). Table 10 gives the runs. At $c = 25\%$ grokking rises from $7/10$ to $9/10$ (Fisher exact, two-sided $p = 0.58$) with the median step falling from 28k to 20k; at $c = 50\%$ both arms are $10/10$, median 3.0k vs. 2.5k. Neither shift is significant at 10 seeds per arm, and the direction is the one the exposure arithmetic predicts. Repetition is thus not an independent lever under the epoch protocol either: where it appears to act, it is acting through the exposure it mechanically adds, bounded here at $+33\%$.

A.7.2 Repetition Is Not an Independent Lever: The Factorial Design

Table 11 reports the full factorial results (Experiment 2; Section 4.5). At fixed unique-pair coverage, repetition R does not change the grokking step within noise. At 25% coverage (1,122 unique pairs), going from $1\times$ (1,122 total samples) to $4\times$ (4,488) leaves the held-out Kaplan–Meier median at 37k

Table 10: The repetition axis under the epoch protocol ($p = 67$, held-out criterion). R is the number of copies of the unique pair set in the training pool; the $R = 1$ rows come from the same code path as Table 9. Presentations/pair is the exact draw count at batch 512 over 100,000 steps, which is not R-invariant because the last batch of each epoch is truncated. Median step is Kaplan–Meier, never-grokked seeds censored.

Coverage	R	grokked	median step	presentations/pair
25%	1	7/10	28.0k	33,333
25%	4	9/10	20.0k	44,444
50%	1	10/10	3.0k	20,000
50%	4	10/10	2.5k	22,222

[18k–72k] vs. 29k [20k–54k] and the never-grokked count at $4/10$ vs. $2/10$, within seed noise, as the protocol identity requires. At 50% the two arms coincide exactly (median 3.0k, $0/10$ never). At 100% the held-out step is undefined because the test set equals the training set; the training-inclusive step is 1.0k regardless of R, a memorization-speed floor.

Table 11: Factorial results (Experiment 2): coverage × repetition, 10 seeds per cell. Step: Kaplan–Meier median [95% bootstrap CI] under the held-out criterion, never-grokked seeds censored; Never: seeds not grokked within 100,000 steps.

Coverage	Repetition	Total samples	Grokking step (KM median [95% CI])	Never
25% (1,122)	1×	1,122	37k [18k–72k]	4/10
25% (1,122)	4×	4,488	29k [20k–54k]	2/10
50% (2,244)	1×	2,244	3.0k [2.0k–3.0k]	0/10
50% (2,244)	2×	4,488	3.0k [2.0k–3.0k]	0/10
100% (4,489)	1×	4,489	1.0k [1.0k–1.0k]	0/10
100% (4,489)	2×	8,978	1.0k [1.0k–1.0k]	0/10
100% (4,489)	4×	17,956	1.0k [1.0k–1.0k]	0/10

Note. Total samples is $R \cdot c \cdot p^2$; under i.i.d. resampling with replacement the per-step sampling probability of any unique pair is $1/(cp^2)$, independent of R (Section 3.3), so the $R\times$ and $1\times$ arms are the same stochastic process by construction and this column is inert for the training process (Section 4.5).

A.7.3 Skewed Distributions and Effective Coverage

Figure 8 shows why skewed sampling suppresses grokking. The left panel plots test accuracy curves (mean ± std over 10 seeds) for the sampling distributions and the controls; the right panel plots the training-inclusive Kaplan–Meier median grokking step with 95% bootstrap CIs (never-grokked seeds censored; the logged quantity for these runs is time to 90% accuracy on the full test set). Zipf, Zipf-counterbalanced, and the uniform control matched to Zipf's effective set never grok within 100,000 steps (10/10 seeds each); long-tail groks by median 9.0k and uniform by 2.0k. Skew therefore acts through the *effective* coverage it retains, not through concentrated frequency per se.

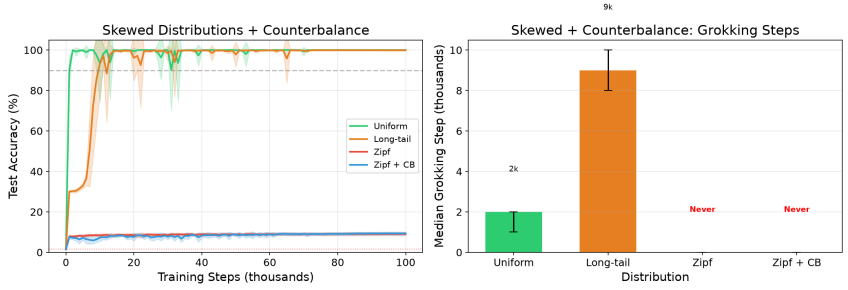

Figure 8: Skewed distributions and controls (Experiment 3). Left: test accuracy curves (mean ± std over 10 seeds); right: training-inclusive Kaplan–Meier median grokking step [95% bootstrap CI] (never-grokked seeds censored).

Experiment 3 draws 4,489 pairs *with replacement* from the skewed distribution, so the effective number of unique pairs is well below 4,489 (Table 12).

The effective-coverage view of Table 12 makes these failures mechanical: Zipf retains only $\sim$326 unique pairs ($\sim$7.3% coverage) and fails exactly like a uniform control of the same size, whereas uniform with replacement retains $\sim$2,840 pairs ($\sim$63%) and groks quickly. We size the matched control by the *first* seed's Zipf draw (326 unique pairs, drawn uniformly without replacement) rather than the mean ($\sim$340; per-seed range $\sim$316–372), a deliberate conservative choice, since a larger control could only make the two conditions more similar; the identical failure ($0/10$ both) is therefore not an artifact of an undersized control.

Table 12: Effective unique pairs under Experiment 3 sampling ($p = 67$; $4{,}489$ pairs drawn with replacement). Uniform and long-tail values are means over 10 seeds; the Zipf value is the first-seed draw that sizes the Uniform-matched control.

Distribution	Effective unique pairs	Coverage
Uniform (with replacement)	$\sim$2,840	$\sim$63%
Long-tail ($\propto 1/(r+1)$)	$\sim$1,340	$\sim$30%
Zipf ($\propto 1/(r+1)^{1.5}$)	$\sim$326	$\sim$7.3%
Uniform-matched (without replacement)	326	7.3%

Table 13 reports the per-condition steps behind the figure and the effective-coverage reading. The two rows above the rule are an above-threshold matched control: over the same per-seed 50%-unique sets, uniform and Zipf grok identically (median 2.0k each, $0/10$ never-grokked), so in this above-threshold matched control concentrated frequency neither suppresses nor accelerates grokking. The rows below the rule repeat the low-coverage conditions of Figure 8, each failing or succeeding exactly as its effective coverage (Table 12) predicts. For these low-coverage rows the training-inclusive and held-out steps coincide because the training pairs are a small fraction of the test set; where both criteria were stored (the two above-rule matched rows), the recovered held-out step is 3.0k vs. the 2.0k training-inclusive one for uniform, matching Table 5, so the conclusions are criterion-independent.

Table 13: Skewed-distribution results (10 seeds per condition; $4{,}489$ pairs drawn with replacement). Step: training-inclusive Kaplan–Meier median [95% bootstrap CI] (never-grokked seeds censored); Never: seeds not grokked within $100{,}000$ steps. Notes below the table define the matched arms.

Condition	Grokking step (KM median [CI])	Never
Uniform (50% unique)	2.0k [2.0k–3.0k]	$0/10$
Zipf‡ (matched 50% unique)	2.0k [2.0k–3.0k]	$0/10$
Uniform	2.0k [1.0k–2.0k]	$0/10$
Long-tail	9.0k [8.0k–10.0k]	$0/10$
Zipf	—	$10/10$
Zipf counterbalanced	—	$10/10$
Uniform-matched†	—	$10/10$

Note. †Coverage-matched to Zipf ($\sim$326 unique pairs). ‡Zipf frequencies over the same per-seed 50% unique-pair set as the uniform row (mean 2,244 unique pairs), holding per-pair sampling probability; the arm holds $4{,}489$ total samples vs. 2,244 in the uniform arm.

A.7.4 Near-Threshold Coverage-Matched Skew Control

The matched controls of Table 13 have effective coverages of $\sim$7% and 50% (far below and above the threshold), so the transition region around c^* was left open. To close it, we run $p = 67$, $100{,}000$ steps, cosine schedule, 10 seeds per condition, at target coverages $c \in \{0.20, 0.25, 0.30\}$ straddling $c^* \approx 24\%$. Each *uniform* arm trains on exactly $\lfloor c \cdot 4{,}489 \rfloor$ unique pairs (one row per pair); each *Zipf* arm uses the *same* per-seed unique set but draws the full $4{,}489$-row sample budget from it with Zipf frequency (weights $\propto (1+i)^{-1.5}$, followed by a per-seed permutation), so unique coverage and total sample count are matched and only the per-pair frequency distribution differs. Held-out accuracy is measured directly on the $(1-c) \cdot 4{,}489$ pairs never in the training set (no decomposition needed). The uniform arm reproduces the dense-scan threshold exactly, even per seed ($1/10$ vs. $1/10$ at 20%; $6/10$ vs. $6/10$ at 25%, median 37k in both; $10/10$ vs. $10/10$ at 30%, median 12k vs. 12k; Table 5), validating the direct held-out measure against the decomposition of Table 5. At matched coverage the Zipf arm never underperforms uniform: below the threshold neither arm groks ($1/10$ each at 20%); at 25% Zipf groks $10/10$ vs. uniform $6/10$ (Fisher exact, two-sided $p = 0.09$); at 30% both grok $10/10$, with Zipf earlier (median 9k vs. 12k; Mann–Whitney $U = 80$, two-sided

$p = 0.03$). Concentrated frequency thus shows no evidence of suppressing grokking beyond what effective coverage predicts; the control is underpowered at 10 seeds per arm (the 25% contrast, 10/10 vs. 6/10, is not significant, two-sided $p = 0.09$), and the point estimates, if anything, run toward slightly earlier grokking near and above the threshold, the modest acceleration reported in Section 4.5.

Table 14: Near-threshold coverage-matched skew control ($p = 67$, 100,000 steps, cosine, 10 seeds/arm; method in the text above). Held-out is measured directly on pairs never in the training set; Never: seeds whose held-out never exceeds 90% within 100,000 steps.

Condition	Unique pairs	Grokking step (KM median [CI])	Never	Mean final held-out
Uniform 20%	897	—	9/10	16.5%
Zipf 20%	897	—	9/10	17.3%
Uniform 25%	1,122	37.0k [18k–72k]	4/10	64.5%
Zipf 25%	1,122	21.0k [15k–21k]	0/10	99.4%
Uniform 30%	1,346	12.0k [9k–14k]	0/10	99.6%
Zipf 30%	1,346	9.0k [8k–10k]	0/10	99.8%

A.7.5 Hyperparameter Dependence of c^*

Finally, the location of the boundary is hyperparameter-dependent rather than universal. Figure 9 plots the proportion of seeds grokking against coverage, one curve per weight decay, with logistic fits (dashed). The estimate is stable at 24–25% for weight decay ≥ 0.3, rises to 34.9% when weight decay drops to 0.1, and moves to 15.6% at learning rate 3×10^{-3} (Section 4.5). Monotone coverage dependence and 10/10 grokking at 50% coverage hold in every setting.

Table 15 lists every condition. The two axes are not symmetric in what we can say about mechanism. For weight decay the direction is expected from the circuit-efficiency competition: weight decay is the field that tilts the memorizer toward the lower-norm circuit (Appendix A.11), so weakening it to 0.1 raises the coverage required, and the two-phase account of training under weight decay (interpolation followed by slow norm minimisation) predicts exactly this ordering. For the learning rate the shift runs the other way: a *larger* learning rate *lowers* c^* to 15.6%. Our data bound this effect without resolving it. Because the boundary is a basin-escape problem (Section 5.1), a larger learning rate supplies larger per-step displacement and more effective noise, which can only make the memorizing basin easier to leave; we therefore report the ordering as a constraint that any mechanism must reproduce rather than as a tested mechanism, and note that the held-out estimate at 3×10^{-3} is 17.5%, so part of the apparent drop is criterion-dependent.

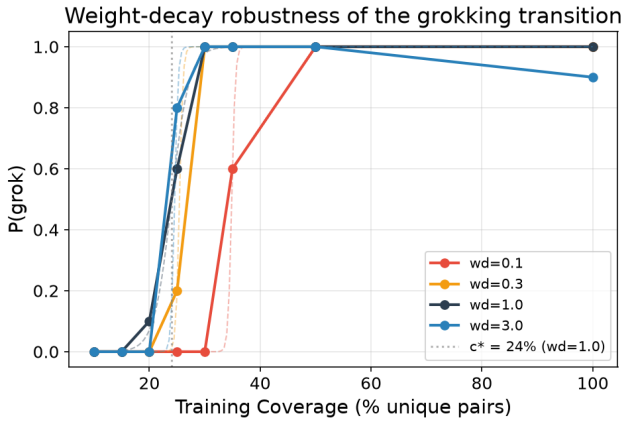

Figure 9: Robustness of c^* to weight decay: proportion of seeds grokking vs. coverage, one curve per weight decay with logistic fits (dashed).

A.8 Mechanism Analysis: Training-Time Evidence

Figure 10 gives the training-time evidence behind the mechanism analysis of Section 3.5: rows are coverage (15%, 25%, 30%) and the columns show (left) test/train accuracy, (center) Fourier entropy

Table 15: Critical coverage c^* per hyperparameter condition (Section 4.5; $p = 67$, 10 seeds per coverage point, canonical binomial-MLE logistic fit with a 20,000-resample bootstrap CI; all values are percentages). The two scans are *independent one-dimensional slices*, not a grid: weight decay was scanned at the default learning rate and learning rate at the default weight decay, and on different coverage grids, so no condition combines a non-default weight decay with a non-default learning rate and the two blocks are not directly comparable. $10/10$ seeds grok at 50% coverage in every condition.

Scan	Condition	Coverage grid	c^* (total, 95% CI)	c^* (held-out)
Weight decay (learning rate 10^{-3}, default)				
8-point	0.1 (low)	$\{0.10, \ldots, 1.0\}$	34.9 [34.3, 35.3]	34.9
8-point	0.3	$\{0.10, \ldots, 1.0\}$	25.4 [25.0, 27.4]	25.4
8-point	1.0 (default)	$\{0.10, \ldots, 1.0\}$	24.0 [22.0, 25.4]	24.0
8-point	3.0 (high)	$\{0.10, \ldots, 1.0\}$	24.6 [22.3, 25.0]	24.6
Learning rate (weight decay 1.0, default)				
5-point	3×10^{-4} (low)	$\{0.15, \ldots, 0.50\}$	30.5 [30.0, 36.2]	30.5
5-point	10^{-3} (default)	$\{0.15, \ldots, 0.50\}$	24.0 [22.0, 25.4]	24.0
5-point	3×10^{-3} (high)	$\{0.15, \ldots, 0.50\}$	15.6 [15.3, 17.6]	17.5

Note. "Total" is the training-inclusive criterion, defined at every coverage; "held-out" is the primary criterion of Section 3.3, which additionally removes the training pairs from the test set and is undefined at 100% coverage. The two criteria agree in every condition except learning rate 3×10^{-3}, where the held-out estimate is 17.5% $[15.3, 19.6]$: part of the apparent drop is criterion-dependent. The default row of each block is the same run set, reused from the dense scan (Table 5).

anchored to the random-init baseline (0.999), and (right) the restricted (training) vs. excluded (unseen) loss on a log scale, means $\pm$ std over 10 seeds, with grokked and never-grokked seeds split at 25%. Below the threshold (15%) entropy barely moves and the excluded loss *rises*: the model becomes an increasingly confident lookup table on unseen pairs. At 30% entropy collapses from $\sim$0.999 to $\sim$0.56 as the excluded loss falls, the signature of the sparse Fourier circuit's emergence. At 25% the grokked and never-grokked seeds separate cleanly in both signals, consistent with the two-state (coexisting) reading of Section 5.1.

Read per seed, the entropy trace is a *predictive* statistic rather than a description of the mean. We take t_{ent} as the first step at which a seed's entropy falls to or below a level H, and t_{grok} as its grokking step (held-out criterion; using the training-inclusive criterion instead shifts the numbers below by at most one evaluation interval). At 25% coverage the 6 seeds that grok have lower entropy than all 4 that never do from step $5{,}000$ onward (AUROC 1.0; Mann–Whitney $p = 0.0095$, the smallest attainable for 6 vs. 4), a gap that widens from 0.15 at step 20k to 0.35 at step 50k, so the eventual outcome is decodable from entropy thousands of steps before any accuracy signal separates the groups. The lead of entropy over accuracy is real but confined to the onset of the collapse: at $H = 0.9$ the drop leads grokking in $10/10$ seeds at 30% (median 2k steps, range 1–3k) and $6/6$ at 25% (median 4k), whereas at $H = 0.8$ only $3/10$ still lead (median 0) and at $H = 0.7$ none do (median 0, i.e. within one 1k evaluation interval). The ordering headline of Section 4.6 should therefore be read as an onset effect, not as a separation of the whole collapse from the accuracy rise. These are lead–lag relations within the entropy trace, not an intervention on it: they bound what a common cause driving entropy and accuracy together would have to explain, but they do not exclude one. The ablation of Appendix A.9 intervenes downstream of the trace (on the readout), so it constrains the support that entropy measures rather than the entropy dynamics themselves.

The crossed hyperparameter scans dissociate entropy from coverage. The learning-rate and weight-decay scans re-run the coverage grid at the same modulus and the same seeds 0–9, so each is a cross rather than a grid: within a coverage column the data are held fixed and only the optimizer's hyperparameters move. Doing that flips the outcome, and the final entropy follows it. At 20% coverage the learning-rate arm goes from $0/10$ to $1/10$ to $10/10$ grokked at 3×10^{-4}, 10^{-3} and 3×10^{-3}, with mean final entropy 0.968, 0.922 and 0.595; at 25% the weight-decay arm goes from $0/10$ to $2/10$ to $6/10$ to $8/10$ as weight decay rises from 0.1 to 3.0, with mean entropy 0.977, 0.863, 0.753 and 0.720. Across the 3×5 and 4×5 crossings the grok fraction and the mean final entropy correlate at $r = -0.97$ and -0.98 over all 35 cells. The relation also holds *within* cells: in all eight cells that contain both outcomes, the grokked seeds end at lower entropy than the never-grokked seeds (gaps 0.19–0.38; Mann–Whitney $p = 0.0048$ for 6 vs. 4, and $p = 0.022$ for 2 vs. 8 and 8

vs. 2). Because coverage, modulus, architecture and seed set are all fixed inside such a comparison, the entropy difference there cannot be an artefact of the data. This rules out the reading that our entropy index merely restates coverage. It does not rule out a common cause: the learning rate and weight decay are themselves plausible joint drivers of both the collapse and the transition, so this is still an association between two optimization outcomes rather than an intervention on the spectrum. Analysis: `scripts/_ablation_crossed_check.py`; values are the final-step `fourier_entropy` scalars already stored in `experiment_dense`, `experiment_lr` and `experiment_weight_decay`.

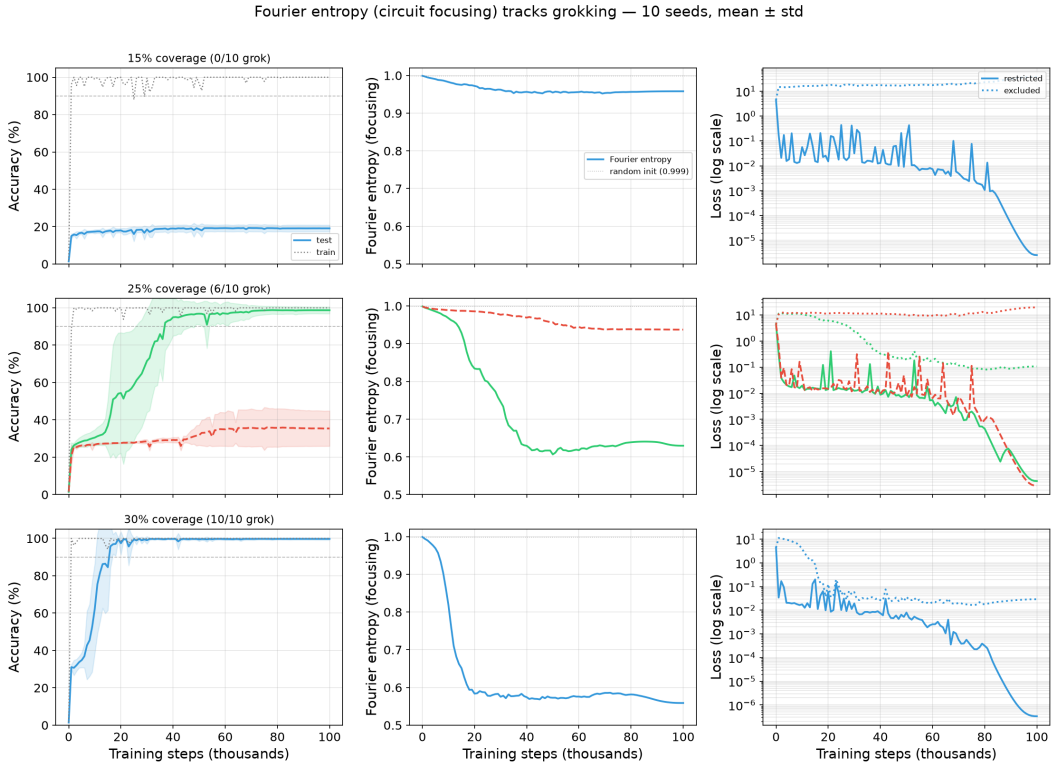

Figure 10: Training-time evolution of Fourier entropy and loss at 15%, 25%, and 30% coverage. Columns: (left) test/train accuracy, (center) Fourier entropy (baseline 0.999), (right) restricted vs. excluded loss (log scale); means ± std over 10 seeds, grokked vs. never-grokked seeds split at 25%.

A.9 Mechanism Analysis: Intervening on the Fourier Support

Everything in Section 3.5 and Appendix A.8 is observational: entropy is measured on the same runs whose grokking it is used to explain. Here we intervene on the support directly and ask whether the concentration is load-bearing. At $p = 67$ and $c = 30\%$ ($1.25 \times c^*$), we mask the unembedding $W \in \mathbb{R}^{p \times d_{\text{model}}}$ along the frequency axis. Because W is real, the unembedding power at frequency k equals that at $p - k$, so the spectrum carries only $\lfloor p/2 \rfloor$ independent two-dimensional subspaces and a mask that is not conjugate-symmetric removes half a mode rather than a mode; every mask we apply is conjugate-symmetric and the projection $P = F^H M F$ is the exact orthogonal projector onto the kept subspace. The number of masked pairs is fixed once, at t^*, and never reselected: reselecting would chase the modes the model migrates to and equal the prohibition of any concentration. We take $k = 5$ from the measured $N_{\text{eff}} = p^{H(0.30)} \approx 10.5$, which for a spectrum uniform over m conjugate pairs gives $N_{\text{eff}} = 2m$; this is a convention, not a derivation, so the runner records the power the mask actually removes, the *dose*, per seed. The 3 arms that re-project (*zero*, *rand*, and the *late* auxiliary) do so every step, which on a linear equality constraint is projected gradient descent, not a heuristic; *freeze* instead restores the coefficients to their t^* values each step (exactly zero change to W at t^* itself); the DC mode is never masked. One caveat on the optimizer: the AdamW moment buffers are *not* projected, and because $P = F^H M F$ mixes the output-residue rows of W rather than acting coordinatewise, the re-projecting arms are projected optimization under an inexact metric rather than exact projected gradient descent. That bounds the fidelity of the *zero*, *rand* and

late arms; *freeze* is unaffected, since restoring W to its t^* values every step makes it an exactly constrained run, so its clean null stands as measured. We set $t^* = 10$k, inside the entropy collapse of Section 3.5 (mean 0.82, with $6/10$ seeds already below 0.95) and below the median grokking step of the $c = 30\%$ scan (12k; Table 5), so the intervention lands in the formation phase; the *late* auxiliary arm applies the same mask at 20k, after all seeds have collapsed.

Table 16 reports the two pre-registered endpoints, which disagree. The binary endpoint is a clean null: every 10-seed arm returns to $10/10$ seeds grokked (Fisher $p = 1.0000$). The continuous endpoint is strongly positive: zeroing the top five pairs drives the *minimum* held-out accuracy after t^* to 1.4% against noop's 54.1% (Mann–Whitney $p = 0.0002$) and delays the median post-intervention grokking step from 10k to 18k, while *freezing* the same five pairs to their t^* values costs nothing (51.0%, $p = 0.7910$) and the dimension-matched random control sits between them (27.0%, $p = 0.1212$; Figure 11). The two auxiliary arms reproduce the effect (1.5%, $p = 0.0007$ one-shot; 2.6%, $p = 0.0013$ late, at a higher mean dose 0.90). We read this as: the sparse support is *not necessary* but *not inert*. Removing it destroys the readout and forces a rebuild, which costs accuracy and thousands of steps; holding it fixed rather than letting it sharpen costs nothing, so it is the presence of the support rather than its continued growth that matters.

Two limits bound the claim. First, the mask acts on the readout only: the residual stream that carries the frequencies into W is untouched, so the model can reroute a frequency and rebuild through a different readout, which is what $10/10$ recovery suggests it does; this is a test of the readout's frequency support, weaker than an ablation of the whole frequency channel. Second, the random control (mean dose 0.19, $\|\Delta W\|_F = 2.80$ versus 5.65 for top-5) matches the *dimension* of the damage and the code path, not its *magnitude*: unembedding power is concentrated, so ten random bins carry far less norm. The dose is recorded per seed for this reason. The effect is not carried by under-dosed seeds: the zeroing arm's non-grokked-seed correlation between dose and entropy at t^* is -0.75, and restricting to the 7 well-dosed seeds (dose ≥ 0.5) leaves the result intact (1.2% vs. noop, $p = 0.0008$). A third limit is methodological: the pre-registered rows asking whether the ablated models' *final* top pairs differ from noop's are untestable here, because the top-5 set read off a converged model is seed-degenerate (noop agrees with itself on $0.09/5$ pairs), so the metric has no null to compare against. A graded dose–response over $k \in \{2, 5, 10\}$ would separate necessity from timing and is left to future work.

Table 16: Mode-selective ablation of the unembedding's Fourier support ($p = 67$, $c = 30\%$, $t^* = 10$k unless noted, 100,000 steps). "Dose" is the fraction of non-DC unembedding power the mask removes, averaged over seeds; $\|\Delta W\|_F$ is measured at t^*. The binary endpoint (does the arm still grok?) is null in every row; the continuous endpoint (minimum held-out after t^*, Mann–Whitney against noop) is where the arms separate, and the last column is the Kaplan–Meier median grokking step after t^*. The last two rows are 5-seed auxiliaries.

Arm	n	Dose	$\|\Delta W\|_F$	Grok after t^*	Min. held-out	p vs. noop	Med. step
Noop (same code path)	10	—	—	$10/10$	54.1%	—	10k
Zero top-5	10	0.68	5.65	$10/10$	1.4%	0.0002	18k
Freeze top-5	10	0.68	5.65	$10/10$	51.0%	0.7910	10k
Zero 5 random	10	0.19	2.80	$10/10$	27.0%	0.1212	11k
Zero top-5, one-shot	5	0.58	—	$5/5$	1.5%	0.0007	12k
Zero top-5 @20k	5	0.90	—	$5/5$	2.6%	0.0013	23k

A.10 The Circuit-Efficiency Count: Derivation and Identifiability

This section expands Section 5.2: it derives the counting argument $c^* \approx \kappa\, N_{\text{eff}}/p$ step by step, validates it out-of-sample by leaving out one modulus at a time, and probes whether the $N_{\text{eff}}\, p$ scale is an identifiability (coefficient-recovery) threshold or a dynamical one.

A.10.1 Derivation of $c^* \approx \kappa\, N_{\text{EFF}}/p$

Under modular addition mod p, the learned generalizing circuit is most compactly parameterized in the Fourier basis $\{e^{2\pi ikn/p} : k = 0, \ldots, p-1\}$ of Section 3.5: the circuit concentrates its unembedding weights on a small set of $\sim N_{\text{eff}} = p^H$ frequencies (for each frequency k, the output is a function of the residue class $a + b \bmod p$, so the weight to output residue r is a single number). The output-side representation therefore holds one coefficient per output residue per mode, i.e. $N_{\text{eff}} \cdot p$ coefficients in total: $C_{\text{gen}} \propto N_{\text{eff}}\, p$. A memorizing circuit, by contrast, stores one output per *distinct*

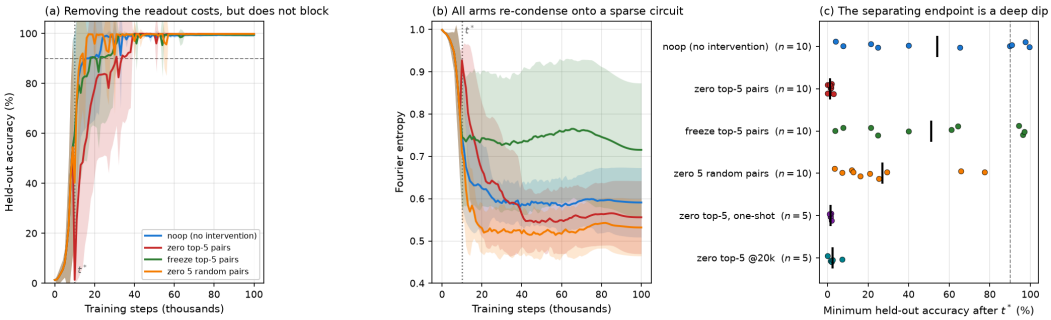

Figure 11: The ablation in three views, same arms as Table 16. (a) Held-out accuracy, mean ± std, with the masking step t^* and the 90% grokking criterion marked: removing the readout produces a deep dip, and every arm climbs back. (b) Fourier entropy: all arms re-condense onto a sparse circuit, so the intervention changes the route, not the endpoint. (c) The pre-registered continuous endpoint per seed (the minimum held-out accuracy after t^*), which is what separates zeroing from freezing and from the random control. The binary endpoint is omitted because it is null in every arm.

training pair, so $C_{\text{mem}} \propto c\,p^2$ at coverage c. Under the weight-decay circuit-efficiency competition of Section 5.3 (Varma et al., 2023), the generalizing circuit is selected once it is cheaper to represent than the memorization, $C_{\text{gen}} < C_{\text{mem}}$, which fixes the critical dataset size $D_{\text{crit}} \approx \kappa\, N_{\text{eff}}\, p$, hence $c^* = D_{\text{crit}}/p^2 \approx \kappa\, N_{\text{eff}}/p \propto p^{H-1}$, i.e. the predicted exponent is $\alpha_{\text{pred}} = 1 - H$.

Table 17 lists the per-modulus circuit quantities. Note that κ is defined per modulus as $\kappa = D_{\text{crit}}/(N_{\text{eff}}\, p) = c^* p/N_{\text{eff}}$, so the per-modulus equality $c^* = \kappa N_{\text{eff}}/p$ holds by construction, and everything of substance is in what the components do. Three observations, each with its limit. (i) κ is order-one at every modulus (0.87–1.61, against the counting bound $\kappa \gtrsim 1$), which is what keeps the equality from being vacuous; it is not tightly constant across the extended range, and the lowest value, $\kappa = 0.87$ at $p = 173$, is computed from an entropy averaged over only the $4/10$ seeds that grok at full coverage there, so we read it as the edge of the spread rather than a precise point (dropping that modulus gives $[1.14, 1.61]$ and moves α_{pred} from 0.402 to 0.420, away from the measured 0.359; Appendix A.10.2). (ii) The H-derived exponent falls inside the measured interval: averaged over the six moduli $\alpha_{\text{pred}} = 1 - H = 0.402$, which lies inside the measured bootstrap CI at every fit size considered here (Appendix A.10.3). The scope of that agreement deserves to be stated, because H is measured rather than derived and it is not constant across the moduli: it runs 0.557, 0.553, 0.597, 0.557, 0.636, 0.689 from $p = 47$ to $p = 173$, so it is not even monotone. The model $c^* \approx \kappa\, p^{H(p)-1}$ is therefore not a clean power law, and the p-dependence the measured H supplies has to be cancelled by $\kappa(p)$ for the model to reproduce the measurement. Pinning κ at its mean (1.335) and letting only the measured $H(p)$ vary gives a fitted exponent of -0.035 ($R^2 = 0.009$), essentially flat against a measured 0.359: all of the decay is carried by κ, which falls from 1.61 at $p = 113$ to 0.87 at $p = 173$, a $1.85\times$ drift that is itself close to a power law ($\kappa \propto p^{-0.39}$, $R^2 = 0.59$). What the previous sentence establishes is that the $N_{\text{eff}}\, p$ scale is not off by a structural factor; it is not an account of why c^* decays. Analysis: `_h_drift.py`. (iii) Over the four small moduli $N_{\text{eff}} \propto p^{0.66}$ ($R^2 = 0.90$): the frequency support grows sub-linearly, so the pair space p^2 outruns it and larger moduli need proportionally fewer unique pairs. Over all six that exponent rises to 1.04 ($R^2 = 0.89$), because H itself climbs from ≈ 0.55 to ≈ 0.69; the sub-linear reading is thus a property of the small-modulus range rather than a general one, and the slow decay of c^* at large p comes from κ falling rather than from a shrinking frequency support.

A.10.2 Sensitivity to the $p = 173$ Base

The weakest input to Table 17 is $p = 173$, where only 4 of 10 seeds grok at full coverage and H is therefore an average over four. This is worth pricing in numbers rather than in a caveat. Three quantities move, and not all in the direction that a sceptical reading would predict.

The κ range. Over all six moduli $\kappa \in [0.87, 1.61]$; over the five that grok fully, $[1.14, 1.61]$. Dropping $p = 173$ removes the bottom endpoint of the range without changing the order-one reading, and moving $c^*(173)$ across its own 95% confidence interval $[17.0\%, 18.0\%]$ moves $\kappa(173)$ only from 0.84 to 0.89, so the value does not depend on where inside that interval the point estimate sits.

Table 17: Per-modulus circuit-efficiency quantities. H is the Fourier entropy at full coverage ($N_{\text{eff}} = p^H$ effective frequencies; Section 3.5); $D_{\text{crit}} = c^* p^2$ is the critical number of unique pairs; $\kappa = D_{\text{crit}}/(N_{\text{eff}}\, p) = c^* p / N_{\text{eff}}$ is defined per modulus, so the equality $c^* = \kappa N_{\text{eff}}/p$ holds by construction: the structural content is that κ stays order-one across all six moduli (0.87–1.61) rather than constant, and that the H-derived exponent $1 - H = 0.402$ (averaged over the six) falls inside the measured $\alpha = 0.359$; Appendix A.10 prices how much of the decay κ itself carries. *grok* counts the seeds reaching the 90% criterion at 100% coverage within 100,000 steps; at $p = 173$ only 4 do, the other six failing to reach 100% training accuracy at all. Those six are unfitted rather than memorizing, so their diffuse spectra are excluded from H (including them would give a meaningless N_{eff} of 87.5), and the $p = 173$ κ should be read with that narrower base in mind; Appendix A.10.2 prices the exclusion.

p	H	$N_{\text{eff}} = p^H$	D_{crit}	κ	c^*_{ho}	grok
47	0.557	8.53	621	1.55	28.1%	10/10
67	0.553	10.22	1077	1.58	24.0%	10/10
97	0.597	15.38	1891	1.27	20.1%	10/10
113	0.557	13.91	2528	1.61	19.8%	10/10
131	0.636	22.18	3324	1.14	19.4%	10/10
173	0.689	34.89	5238	0.87	17.5%	4/10

The predicted exponent. Averaged over six moduli $\alpha_{\text{pred}} = 1 - H = 0.402$ against a measured 0.359; over the five that grok fully it is 0.420. Removing the least-supported modulus therefore moves the prediction *away* from the measurement by 0.018, not toward it. The naive reading of "four seeds out of ten" is that the point flatters the account; here it is the point that agrees best.

The alternative base. Averaging H over all ten seeds, including the six that never fit the training set, gives $H = 0.868$, $N_{\text{eff}} = 87.5$ and $\kappa = 0.35$. This is the number the exclusion rule is designed to prevent: N_{eff} is supposed to count the frequencies of the circuit that generalizes, and six of these runs have no such circuit, so including them measures the diffuseness of a memorizer rather than the width of a solution.

The six failing seeds are not near-misses on the generalisation criterion. Their final *training* accuracy is 0.29%, 0.59%, 0.68%, 0.78%, 0.78% and 1.07%: the criterion they fail is fitting the training set, so the correct description is that they did not complete optimisation, not that they trained and failed to generalize. This is the completion limb of Appendix A.12 at work – their per-pair exposure is $\lambda = 1{,}711$, against 11,406 at $p = 67$, a factor of 6.7 – and it is why the $p = 173$ full-coverage point carries no leverage on $c^*(173)$, which is 17.5% whether or not that point enters the fit (on the bare shared grid, before the four refinement points, it is 18.1%; the movement comes from grid resolution, not from the failing seeds). Analysis: `_p173_sensitivity.py`.

A.10.3 Out-of-Sample Validation (Leave-One-Modulus-Out)

To test whether the scaling is predictive rather than a fit, we leave out one modulus at a time and predict its c^* without using any of its data, along two routes. *Leg A (exponent route)* fits $\log c^* = \text{const} - \alpha \log p$ on the training moduli and evaluates the fit on the held-out modulus; over the four moduli of the original scan this reproduces the held-out c^*s to mean absolute error 0.62 pp (Section 4.3). *Leg B (N_{eff} route)* instead fits $N_{\text{eff}} \propto p^s$ on the training moduli, predicts the held-out modulus's N_{eff}, and forms $c^* = \bar{\kappa}\, N_{\text{eff}}^{\text{pred}}/p$ with $\bar{\kappa}$ the mean κ over the training moduli only; over the same four moduli this reproduces the held-out c^*s to mean absolute error 1.1 pp (Section 5.2). Both routes are genuinely out-of-sample: neither uses the held-out modulus's own c^* or N_{eff}. Extended to all six moduli the two routes separate: Leg A holds up (mean absolute error 0.74 pp), while Leg B degrades to 4.1 pp, driven by held-out $p = 47$, where a power law anchored on the five larger moduli underpredicts c^* by 11.7 pp. The cause is the one identified above ($N_{\text{eff}}(p)$ is a power law only over the small-modulus range), so Leg B should be read as a check over the range where its premise holds, not as a second independent prediction.

The $p = 257$ scan is a further, out-of-range test point for Leg A (absent from every fit; held-out grid $\{0.125, 0.15, 0.20, 0.25\}$). Fitting on the four smallest moduli predicts $c^*(257) \approx 13.8\%$, on five 14.5%, and on all six 14.9%, against a measured 15.3% ([15.0%, 17.6%]): every fit overshoots the decay slightly, and the miss shrinks monotonically from +1.5 pp to +0.4 pp as moduli are added. The direction is informative: the true decay is shallower than any of these fits, consistent with the flattening of c^* visible at large p in Table 7, but even the four-modulus fit misses by only 1.5 pp, so

the extrapolation is not what limits the account. Leg B is unavailable at 257: the 100%-coverage runs there do not grok (final training accuracy below 100%), so $N_{\text{eff}}(257) = 257^H$ cannot be measured; we state this rather than fabricate a number.

A.10.4 Identifiability Probe: A Dynamical Scale, Not a Recovery Threshold

One alternative reading of the $N_{\text{eff}}\,p$ scale is that it is a *coefficient-recovery* (identifiability) threshold: the coverage at which the $N_{\text{eff}}\,p$-parameter generalizing model can be identified from the training samples. To test this, we fit a static linear-softmax model on the sum-Fourier features $\phi(s)$ of the residue class $s = a+b \bmod p$: $\text{logit}_r = A[r,:]\cdot\phi(s)$ with $A \in \mathbb{R}^{p\times 2m}$, $m = \text{round}(N_{\text{eff}})$, trained on $c\,p^2$ diagonal samples ($s \mapsto$ one-hot(s)) by cross-entropy with weight decay. This model has exactly the $N_{\text{eff}}\,p$ output-side parameter count of the circuit, but no optimization dynamics to speak of beyond a deterministic fit, so it isolates the identifiability question. Table 18 reports c_{90}, the coverage at which this static model first reaches 90% held-out accuracy, together with $\kappa_{\text{model}} = c^*_{\text{ho}}/c_{90}$.

The result is decisive: $c_{90} \approx (1.9\text{–}2.6)/p$ (Table 18), so $\kappa_{\text{model}} = 5.1\text{–}10.0$, far above the transformer's order-one κ. The static model needs only a coupon-collector number of samples: each residue class is observed with probability $1 - e^{-cp}$ after $c\,p^2$ diagonal samples (each sample lands on class s with probability $1/p$), giving held-out accuracy $1 - e^{-cp}(1 - 1/p)$ that crosses 90% at $c \approx (\ln 10)/p \approx 2.3/p$, far less than the $c^* \approx \kappa N_{\text{eff}}/p$ at which the transformer actually groks. The $N_{\text{eff}}\,p$ scale therefore does *not* bound coefficient recovery; it bounds the *representation* the optimizer must form under the memorization-vs.-circuit competition. Consistent with this, a two-layer ReLU MLP toy on the same diagonal samples never generalizes at any weight decay (0.03–1.0; held-out accuracy stays below $\sim 8\%$ even above the transformer's threshold), so a toy κ is undefined and we make no claim of deriving the numerical value of κ: the measured order-one κ remains an anchored dynamical constant, consistent with the absorbing-state finding (Section 5.1) and the weight-decay and learning-rate scans (Section 4.5).

Table 18: Identifiability probe (Appendix A.10.4). $m = \text{round}(N_{\text{eff}})$ is the number of Fourier modes in the static linear-softmax model on the sum-Fourier features of the residue class; c_{90} is the coverage at which it first reaches 90% held-out accuracy; $\kappa_{\text{model}} = c^*_{\text{ho}}/c_{90}$ compares the static recovery scale to the transformer's measured threshold. Since $c_{90} \approx 2.3/p$ is set by a coupon-collector count while the transformer threshold is $c^* \approx \kappa N_{\text{eff}}/p$, the ratio lies far above the dynamical κ of Table 17: the $N_{\text{eff}}\,p$ scale is not a coefficient-recovery threshold.

p	N_{eff}	m	c_{90}	c^*_{ho}	κ_{model}
47	8.53	9	5.5%	28.1%	5.13
67	10.21	10	2.9%	24.0%	8.27
97	15.38	15	2.3%	20.1%	8.71
113	13.91	14	2.0%	19.8%	10.03

We close the appendix with three short theoretical readings of the results above: an order-parameter view of the formation boundary (Appendix A.11), an expected-exposure identity that explains the protocol invariance of c^* (Appendix A.12), and a discussion of what the measured threshold is not: an identifiability limit (Appendix A.15). Each restates and reinterprets results already reported above; none introduces new experiments. We close by relating our account to concurrent and very recent work (Appendix A.16).

A.11 A Phase-Transition Reading of the Formation Boundary

It is useful to read the results above as a phase diagram in the coverage c, with the *grok fraction* $g(c)$ (the share of seeds whose held-out accuracy exceeds 90%) playing the role of the order parameter. Across the dense scan, g is 0 for $c \le 15\%$, rises from 0.1 at 20% to 0.6 at 25%, and reaches 1.0 for $c \ge 30\%$ (Table 5); $c^* \approx 24\%$ is where the fitted logistic passes through $g = 1/2$, locating the steepest rise of the order parameter rather than a hard per-seed cutoff. Because each cell holds only 10 seeds, the boundary is a soft band whose width reflects finite-seed rounding of an underlying per-seed Bernoulli process (Section 4.1). Read through this lens, the bidirectional intervention of Section 4.2 is evidence for a window of *coexisting states*: the same final coverage, $c = 20\%$, is reached from below by the static scan, which groks only $1/10$ seeds within 100,000 steps, with most of the remainder slow, and from above by the removal arm, which arrives at step 50k and keeps all 10 seeds grokked, with held-out on pairs never seen at any training phase recovering to $\ge 95.7\%$ in every seed. The outcome is therefore not a function of the final training set alone but

of the path by which that set was reached, so the transition is *hysteretic*: the signature of a first-order-like transition, with two locally stable solutions separated by a barrier, rather than a smooth second-order crossover fixed by c alone. Direct within-coverage evidence for two coexisting states comes from the per-seed split at 25%, where grokked and never-grokked seeds separate cleanly in their entropy and excluded-loss trajectories (Figure 10): the run that will never grok is not merely slower but occupies a different state. The reading is deliberately weak, and the finite-pair, finite-seed setting does not afford a rigorous free-energy treatment: coexistence is established with a handful of seeds and single forward/backward interventions, we neither estimate the barrier between the states nor its crossing dynamics, we make no claim about a thermodynamic limit, and we use "order parameter," "field," and "coexistence" heuristically. Under this reading, coverage plays a role that the static circuit-efficiency comparison of Varma et al. (2023), which asks which solution is cheaper once training is over, does not capture: it sets both whether the generalizing basin is *feasible* at all (enough of the group structure is seen for the circuit to exist; Section 5.1) and whether the optimizer's trajectory can *reach* it from a memorizing initialization. In the absorbing regime the optimizer does not leave the memorizer: no seed at 15% generalizes even at $500{,}000$ steps (Section 4.2). That is a statement about *reachability* from a memorizing initialization under this optimizer on the budgets we ran, not about whether a lower-loss solution exists: a $500{,}000$-step truncation cannot distinguish an absent basin from one this trajectory never finds. Just above c^*, both basins are present and the outcome is set by history. Weight decay acts like a field tilting the competition toward the lower-norm circuit once it is available, which is why it was identified as the key driver of grokking in the first place (Power et al., 2022) and why grokking is read as the amplification of a gradually formed circuit (Nanda et al., 2023). The field moves the boundary: c^* rises to 34.9% when weight decay falls to 0.1, is flat at 24–25% for weight decay ≥ 0.3 (Figure 9), and drops to 15.6% at learning rate 3×10^{-3} (Section 4.5), while monotone coverage dependence and $10/10$ grokking at 50% hold in every setting, consistent with the lazy-to-rich account in which leaving the memorizer requires entering a feature-learning phase (Kumar et al., 2024).

A recent landscape-sampling study complicates the first-order reading and deserves a direct answer. Zhang et al. (2025) sample the Boltzmann entropy landscape of small networks (the density of parameter configurations as a function of training loss, obtained by Wang–Landau molecular dynamics) and find *no* entropy barrier separating memorization from generalization, concluding that grokking is better described as a computational glass relaxation than as a first-order transition. Their argument is aimed at accounts that identify grokking with a first-order phase transition, so it applies to the reading we have just given. It does not displace that reading, because the two discuss different objects. Their entropy is a *configurational* quantity: the log number of parameter settings consistent with a given loss, a property of the loss landscape as a whole. Ours is a *spectral* one: the Shannon entropy of the unembedding's power spectrum at a single trained network (Section 3.5), which does not count the multiplicity of configurations realizing that network at all. Their reported high-entropy advantage (higher-entropy states generalizing better) and our finding that the Fourier entropy *falls* as accuracy rises are therefore not in tension; they measure different spaces. The same distinction applies to the barrier. Their claim is that the configurational free-energy surface carries no barrier along the memorization-to-generalization path; ours is an inference from hysteresis in c (the same final coverage groks or does not depending on the direction from which it was reached), which establishes two locally stable attracting states without measuring a barrier between them, as noted above. A landscape with two stable attractors and no configurational barrier between them is a description our data are consistent with; what the hysteresis does rule out is the opposite extreme, a final state fixed by c alone with no history dependence.

A.12 Feasibility Versus Completion: Expected Exposure Across Protocols

Section 5.3 introduces the per-pair exposure $E = \text{steps} \times B/(c\,p^2)$ and uses it to argue that coverage acts through the *span* of the pair pool. Here we give that quantity its formal content and draw out a consequence that the protocol control of Appendix A.7.1 already measures. Let $N = c\,p^2$ be the number of distinct training pairs, B the batch size, and T the number of steps. Under the main protocol, batches are drawn i.i.d. with replacement, so the expected number of times a fixed pair is presented over the whole run is

$$\lambda \;=\; \frac{T\,B}{N} \;=\; \frac{T\,B}{c\,p^2},$$

because each of the $T\,B$ draws hits a given pair with probability $1/N = 1/(c\,p^2)$, independent of how often that pair has appeared before (Section 3.3). Repetition R enters neither N nor the per-

draw probability, so the $R\times$ and $1\times$ factorial arms are the same stochastic process by construction (Section 4.5, Table 11); doubling R at fixed N leaves λ unchanged, so the repetition axis cannot matter, a consistency check that any theory of exposure must pass.

The same expectation holds under the without-replacement *epoch* protocol of prior work (Power et al., 2022): each epoch visits every distinct pair exactly once in a random order, so it comprises N/B batches and the T steps realize $T/(N/B) = TB/N$ epochs, giving each pair $\lfloor\lambda\rfloor$ or $\lceil\lambda\rceil$ presentations. The two protocols therefore agree in expected per-pair exposure by construction and differ only in the higher-order statistics of the per-pair counts (approximately independent counts under i.i.d. redrawing versus nearly constant counts under epoch cycling). If grokking is governed by the set of distinct pairs and how often each is seen, the threshold in c should be protocol-invariant, which is exactly what the control measures: $c^* = 23.0\%$ under the epoch protocol versus 24.0% under i.i.d. resampling, with per-level grokked counts agreeing within seed noise at every level (Table 9).

The identity also sharpens the division of labor between coverage and compute. Coverage fixes N, and with it the *feasibility* of the generalizing circuit, namely whether N exceeds the critical c^*p^2 at which the memorization↔circuit competition can resolve toward the circuit (Section 5.1). Exposure λ (steps T at fixed B and c) then fixes *completion*: the probability and speed with which a feasible run actually groks within its budget. The data line up with this split: runs just above c^* grok slowly but eventually (median 37k at 25%; at 20% and 25%, $4/9$ and $3/4$ of the seeds that had not grokked by 100,000 steps do so by 500,000), runs far above it grok fast (median 12k at 30%), and runs below it are not rescued by more exposure: no seed at 15% groks even at 500,000 steps (Section 4.2) because the optimizer never leaves the memorizer on any budget we ran, the data-scarce regime in which grokking may simply remain unobservable (Singh et al., 2025). In short, coverage gates *whether* grokking can happen; exposure gates *whether it does, and how fast*.

A.13 Exposure Is Necessary but Not Sufficient

The exposure accounting above raises the obvious alternative explanation of Section 5.3: perhaps coverage matters only because it sets λ, and the boundary is an exposure threshold in disguise. A direct control would hold λ fixed and vary c, which requires the step budget to covary with coverage ($T = \lambda\, c\, p^2/B$) and therefore trades a coverage–exposure confound for a coverage–budget one. We do not have that control. What we do have are three readings of λ on the existing grid, and all three point the other way.

First, a within-modulus reversal. Fix $p = 67$ and compare two arms whose exposure differs tenfold in favour of the one that should lose. At 15% coverage the run is handed 500,000 steps, or $\lambda =$ 380,189 presentations per pair; no seed groks ($0/10$). At 30% coverage the run gets 100,000 steps, or $\lambda = 38{,}019$; every seed groks ($10/10$). The modulus, task, optimizer and seed set are identical, and the arm with $10\times$ the exposure is the one that fails. This is the opposite ordering from the one an exposure-only account predicts.

Second, a matched-exposure band across moduli. Table 19 collects cells from four moduli whose λ agrees to within $1.10\times$. All four are $10/10$ except the last, and what separates it is not the exposure but its position relative to that modulus's own boundary: it is the only cell below c^*.

Table 19: Matched per-pair exposure across moduli ($T = 100{,}000$, $B = 512$, held-out criterion). The band is $\lambda \in [10{,}500, 12{,}500]$, a spread of $1.10\times$; c^* is the held-out threshold of Table 7. Exposure is held fixed while coverage varies by a factor of 3.3, and the outcome is ordered by c/c^*, not by λ. At $p = 67$ the only grid point inside this band is $c = 100\%$, where the held-out decomposition is undefined (test = train); under the training-inclusive criterion it groks $10/10$, consistent with the rest of the column.

p	c	λ	grokked	c/c^*
97	50%	10,883	10/10	2.48
113	35%	11,456	10/10	1.77
131	25%	11,934	10/10	1.29
173	15%	11,405	0/10	0.86

Third, the exposure at the boundary itself. Evaluating λ at each modulus's own threshold gives $\lambda^*(p) = TB/(c^*(p)\, p^2)$, which falls from 82,628 at $p = 47$ to 47,455, 27,026, 20,261, 15,406 and 9,777 at $p = 173$: a factor of 8.5 across the family. If exposure alone gated grokking, λ^* would be a constant of the task.

The converse failure – full coverage with too little exposure – is the completion limb of Appendix A.12, and it shows up at the largest modulus measured: at $c = 100\%$, where every modulus up to $p = 131$ groks $10/10$, $p = 173$ groks $4/10$ with $\lambda = 1{,}711$ against $11{,}406$ at $p = 67$ (Appendix A.10.2).

Two limits on what this establishes. It is a *re-reading of the existing grid* rather than a scan designed for the purpose, so the band of Table 19 was found after the fact rather than fixed in advance; and it excludes exposure as a *sufficient* variable, but not that some other monotone function of (T, c) could order the outcomes. A control that pins λ and sweeps c remains the clean test and has not been run (Section 5.5). Analysis: `_matched_exposure.py`.

A.14 The Threshold Is a Contour of the Step Budget

The critical coverage and the exponent are both read off runs of $100{,}000$ steps, so they describe a contour of that budget rather than a budget-free property of the task. The budget does move the boundary, and the movement is measurable at one modulus: $p = 67$ was extended to $500{,}000$ steps at three coverages, seeded with only the runs that had not grokked by $100{,}000$ (Table 20). At 20% coverage, five times the exposure rescues $4/9$ of the stalled seeds; at 25% it rescues $3/4$; at 15%, which lies below $c^*(67) = 24.0\%$, it rescues none. The boundary moves but does not dissolve, and the ordering of the three coverages is unchanged from 100k to 500k steps.

The rescue is complete by half the budget. Reading the same $500{,}000$-step trajectories at their midpoint gives counts identical to the endpoint at every coverage: of the 7 seeds that grok under the 500k schedule, all 7 have crossed by step 250k, and the latest grokking step anywhere in these runs is $173{,}000$. The 100k→500k gain is therefore a 100k→250k gain, and doubling the budget a second time buys nothing measurable at this modulus. This midpoint is a *reading of the* $500k$ *trajectories*, not an independent 250k run: the cosine schedule is bound to `n_steps` (Section 3.3), so a genuine 250k experiment would train under a different schedule entirely.

Table 20: What the step budget buys, at $p = 67$ ($T = 100{,}000$ and $500{,}000$, $B = 512$, held-out criterion). The 500k arms are seeded only with the runs that had not grokked by 100k, so their denominators are the survivors – $4/9$ and $3/4$, not $4/10$ and $3/10$ – the same convention as Appendix A.12. The exposure column is $\lambda = TB/(c\,p^2)$.

c	λ @ 100k	grokked @ 100k	survivors	λ @ 500k	grokked @ 500k
15%	76,038	0/10	10	380,189	0/10
20%	57,028	1/10	9	285,141	4/9
25%	45,623	6/10	4	228,113	3/4

The budget axis cannot be crossed with the modulus axis in the present data: p= 67 is the only modulus with a second budget, and it has only three coverage levels at 500k. A 100k/250k/500k scan across moduli would be the natural extension; what the existing runs support is the weaker statement above, which is why Section 5.5 states the budget explicitly wherever c^* is quoted. Analysis: `_budget_sensitivity.py`.

A.15 Identifiability, Search, and the Meaning of κ

The identifiability probe of Appendix A.10.4 lets us state what the measured threshold is not. A static linear-softmax model that is *given* the sum-Fourier features $\phi(s)$ of the residue class $s = a + b \bmod p$ (told the structure of the target function in advance) reaches 90% held-out accuracy at $c_{90} \approx (1.9\text{–}2.6)/p$, from 5.5% at $p = 47$ down to 2.0% at $p = 113$ (Table 18); the scale is a coupon-collector count, since after $c\,p^2$ samples each residue class has been observed with probability $1 - e^{-cp}$. At $p = 67$, the input–output map is thus identifiable from $c_{90} \approx 2.9\%$ of the pairs (≈ 130 pairs), well below the measured $c^* \approx 24\%$ (1,077 pairs) by a factor of ≈ 8. Grokking is therefore not hard because the data are too scarce to determine the function; it is hard because the optimizer must *form the representation* in which the function becomes simple (the modular-additive circuit of Section 3.5) rather than being handed that representation, and it must do so while competing with the memorizing solution.

Two numbers place this difficulty. The transformer groks at $c^* = \kappa\, N_{\text{eff}}/p$ with κ of order one (0.87–1.61; Table 17), a factor κ above the bare circuit-efficiency crossing ($\kappa \geq 1$; Appendix A.10.1), and a factor $\kappa_{\text{model}} = c^*_{\text{ho}}/c_{90} = 5.1\text{–}10.0$ above the coupon-collector scale that suffices once the correct feature is supplied. Neither factor is a count: the number of coefficients involved is fixed by $N_{\text{eff}}\,p$ in both comparisons, so the excess over the count bound and the much larger excess over

the identifiability scale are properties of the optimization dynamics, not of the dataset. Because $c^* \propto p^{-0.36}$ decays more slowly than the $c_{90} \propto 1/p$ identifiability scale, the gap between them *widens* with the modulus (κ_{model} grows from 5.1 at $p = 47$ to 10.0 at $p = 113$; Table 18): larger moduli are progressively more *discovery*-bound relative to what their data allow. This is the grokking analogue of the classical separation between statistical and computational sample complexity for learning low-degree structure under gradient dynamics: for parity, networks require more samples than information theory does because the relevant low-degree structure must be discovered rather than assumed (Barak et al., 2022). Here the analogue holds with coverage-gated *formation* playing the role of discovery, and the lazy-to-rich account supplies the mechanism: near a low-coverage memorizer the network sits in a lazy, overfit fixed point, and leaving it requires the rich, feature-learning phase in which the sum is actually learned (Kumar et al., 2024) (Section 5.1).

This reading suggests a concrete target for future theory. The order-one magnitude of κ across moduli, alongside the absorbing-state, removal, and hyperparameter results (Sections 4.2, 4.5), marks it as an anchored dynamical constant rather than a fitted accident: an account of grokking's dynamics should explain why the optimizer needs roughly one to one-and-a-half times the bare efficiency crossing, and why that overhead does not grow with the modulus. We do not attempt such a derivation here; our contribution is to have isolated the scale, and the two gaps around it, that a dynamical theory must reproduce.

A.16 Concurrent and Very Recent Work

Six recent papers bear directly on the account above. None reproduces our setting, and we state for each what it independently establishes and where its scope ends.

Dynamical competition, not capacity. Nguyen & Reddy (2025) study the emergence of in-context learning and locate the transition at a *task-diversity* threshold, the analogue of our coverage axis, with the diversity of the training distribution in place of the group structure. Their central finding is that the transition is governed by a dynamical competition between largely independent memorizing and generalizing sub-circuits, set by the relative rates at which the two are learned, rather than by a capacity bound; near the threshold the outcome is bimodal across runs, and the number of iterations required to acquire the solution is exponentially sensitive to initialization. This is the picture our 25% split gives within a single coverage level: the seeds that grok and those that never do separate in their entropy and excluded-loss trajectories rather than merely running at different speeds (Figure 10), and the seed-to-seed variability at fixed c is what makes c^* a soft band (Section 4.1) rather than a hard per-seed cutoff. Their setting is in-context learning on synthetic sequence tasks rather than modular arithmetic, so it does not speak to the scaling of c^*; it does independently support reading the threshold as a property of the optimization dynamics rather than of the data alone.

A two-phase optimization account. Boursier et al. (2025) give the cleanest formal version of the two-phase structure we invoke in Section 5.1. For gradient flow under a small weight-decay coefficient, the dynamics provably split: a fast phase drives the parameters onto a critical manifold on which the training loss is stationary, and a slow phase then drifts along that manifold, minimizing parameter norm at a rate set by the decay: times of order the inverse decay coefficient. The first phase corresponds to the formation of the generalizing circuit and the second to the readout amplification that weight decay performs once the circuit is representable, which is why in our setting accuracy keeps rising long after the entropy has collapsed (Section 3.5), and why the median grokking step is so sensitive to weight decay near the boundary (Section 4.5). Their result is proved for regression and explicitly excludes classification, so it does not derive our threshold in c; it supplies the optimization-theoretic reason to expect the formation/readout split rather than treating it as a description of the curves.

The Fourier solution is not a neural artifact. Mallinar et al. (2025) obtain grokking for modular arithmetic outside neural networks altogether, in a recursive feature machine whose features are learned through the average gradient outer product. The features that emerge are block-circulant (the same discrete-Fourier structure that the mechanistic account attributes to the transformer (Nanda et al., 2023)), and the transition again exhibits hidden progress measures that move before the accuracy does. Because both the architecture and the learning rule differ, this is evidence that the Fourier solution our analysis centers on is the task-natural one rather than a by-product of attention or of

gradient descent on a transformer, and it supports reading the entropy collapse of Section 3.5 as the discovery of the group structure rather than an architectural accident.

Coverage in a real-world setting. Abramov et al. (2025) extend grokking to real-world multi-hop question answering, where the analogue of our coverage axis is the ratio of *inferred* facts to atomic facts in the knowledge graph. Raising that ratio above a threshold unlocks generalization, and (the sharpest point for our account) augmenting with synthetic data that is *factually incorrect* still helps, because such data forces the model to rely on relational structure instead of memorized surface associations. That is the qualitative signature of the span-versus-count distinction of Section 5.3: what matters is whether the training set spans the relations the target depends on, not how many examples it holds. Their threshold is measured on a knowledge graph rather than on a group, so we do not claim it is the same constant; we claim only that the operative variable is coverage of structure in both settings.

The delay as a numerical phenomenon. Prieto et al. (2025) locate grokking's delay in floating-point arithmetic rather than in the data. Past the point of overfitting, they find, the gradient acquires a component, *naive loss minimization*, that reduces the loss by rescaling the logits along their current direction without changing the model's predictions; the logits grow until the softmax saturates and collapses, at which point a large fraction of samples contribute no gradient and learning stops. They identify this *softmax collapse* as what prevents grokking when such tasks are trained *without* regularization, and show that a numerically stable activation (StableMax) or a gradient projection that removes the logit-scaling component restores it. As an account of *why the transition is delayed*, this is a claim about the regime our protocol occupies rather than a rival to the coverage threshold. Every run reported here uses weight decay, and the located threshold is flat across the range we scan ($c^* = 25.4\%$, 24.0%, 24.6% at weight decay 0.3, 1.0, 3.0; Table 15); the coverage manipulation therefore changes the outcome with the numerics of the loss held fixed, and the mid-training injection and removal arms flip it without touching the optimizer, the schedule, or the arithmetic at all (Section 4.2). Their account also has no coverage axis (the same architecture, task and sample count can be made to grok or not by stabilizing the softmax), whereas in ours the same architecture exceeds or misses the threshold purely as a function of c. The two are complementary: numerical instability blocks learning in a region of parameter space we never enter, and the coverage threshold governs which training sets make the generalizing solution reachable once that block is absent.

Symmetry breaking as a necessary condition. Bernal et al. (2026) study the same modular-arithmetic family in the recursive feature machine of Mallinar et al. (2025) and find that the training set must *break a symmetry* of the task for generalization to appear: train–test partitions that preserve a reflection symmetry, or that withhold the fixed points, block it, while moving random training points into the test set, which breaks the symmetry, restores it. The features the machine learns encode elements of the task's invariance group, which is why the symmetry of the partition, and not its size, is what the transition tracks. This is a necessary condition our design satisfies at every point on the grid. For modular addition the label-preserving symmetries are the swap $(a, b) \mapsto (b, a)$ and the diagonal translations $(a, b) \mapsto (a + t, b - t)$, whose orbits are the p sets of constant sum; a uniformly random c-fraction of the p^2 pairs is invariant under a nontrivial translation only with probability vanishing in p, so the condition is met already at $c = 5\%$, an order of magnitude below c^*. Their result therefore does not place the threshold, and it is consistent with ours: it says the group structure must be visible to the learner at all, while c^* says how much of the table must be shown before the generalizing solution is found. It also sharpens why the operative variable is coverage of the *pair* set rather than of the sample stream, since symmetry is a property of which pairs are present, exactly the object our c counts.

Read together, these establish the qualitative claims of this paper in three settings other than ours (in-context learning, a non-neural model, and real-world multi-hop QA) while leaving the specific scaling $c^* \propto N_{\text{eff}}/p$, and the hysteretic boundary that accompanies it, untested outside modular arithmetic. The two accounts added above constrain the claim rather than extend it: one bounds the regime in which the delay we study is even reachable, and the other supplies a necessary condition our coverage grid satisfies at every point. Those remain what we claim.