WHEN GROKKING FAILS: PHASE BOUNDARIES UNDER DATA SKEW AND LOW-DOSE RESCUE

Anonymous authors
Paper under double-blind review

ABSTRACT

Grokking is usually studied under fixed training distributions, leaving unclear how data imbalance changes whether delayed generalization is reached at all. We study this question by continuously varying data skew and weight decay in an autoregressive counting task. Increasing skew systematically contracts the region in which models grok and reveals two qualitatively different failure regimes: a boundary regime where additional training can recover generalization, and a deep-trap regime that remains ungeneralized even after 10^6 training steps. This separation shows that skew can change the accessibility of generalizing solutions, rather than merely delaying their emergence. We then ask whether this loss of accessibility can be reversed through small changes to the training distribution. Replacing fewer than 1% of minibatch slots with samples drawn from a broader distribution can restore grokking, while sample-count-matched interventions that preserve the original sampling distribution or target a narrow rare-state pool are substantially less effective. Rescue strengthens as the support of the injection distribution broadens and remains possible even when putative bottleneck states are excluded, arguing against a simple missing-rare-example explanation. The same intervention ordering transfers to modular addition. On Dyck-1, skew again produces a structured failure region and broader injection remains preferable, although complete rescue does not transfer and longer training can expose late-stage collapse. Overall, these results identify broad, repeated exposure across the training distribution as an empirical control axis for both the accessibility and recovery of grokking under data skew.

1 INTRODUCTION

Grokking, the delayed generalization of a model after it has already fit its training set, was first reported by Power et al. (2022) on small algorithmic datasets and has since been studied from several mechanistic perspectives (Liu et al., 2022; 2023a; Nanda et al., 2023). Zhao (2026) study LARGECOUNTER, an autoregressive binary counting task with succinct algorithmic circuits (Bergsträßer et al., 2026), and ask whether stratified sampling that over-represents rare carry-chain states creates grokking on those states or extends an effect that would occur anyway. Their evidence is a qualitative comparison of uniform and stratified sampling. The dependence on continuous data skew, and the intervention size needed to recover a model after failure, remain unmeasured.

Although the above work provides valuable evidence, two challenges remain. First, the evidence of previous study (Zhao, 2026) rests solely on a qualitative comparison between uniform and stratified sampling, and thus does not characterize how grokking depends on the continuous degree of data skew. Second, once a model has collapsed into a failure state, the scale of intervention needed to recover it—that is, how many samples must be injected to take effect—has yet to be quantified.

To address the gap, this paper studies two questions: (1) How does grokking depend on the continuous skew parameter, and what structure emerges from its interaction with regularization strength? (2) After a model has failed, what scale of sample-injection intervention suffices to recover it, and which form of intervention is effective?

We use one experimental setup to study both questions. The framework of our study is illustrated in Fig. 1. We apply Zipf-weighted minibatch sampling, $P_\alpha(x) \propto (x+1)^{-\alpha}$, to provide a continuous skew axis α, which we cross with weight decay λ on LARGECOUNTER. The resulting (α, λ) map separates a boundary region where more compute improves outcomes from a deep-trap region that remains stuck even at four times the originally tested budget. We then inject a small number of examples into otherwise skewed minibatches. Uniform full-state injection at about 0.8% of a minibatch rescues both observed failure modes. Controls that vary injected sample count, targeted rarity, and pool breadth show that broad full-state injection is the only tested intervention that produces complete rescue. Our contributions are:

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

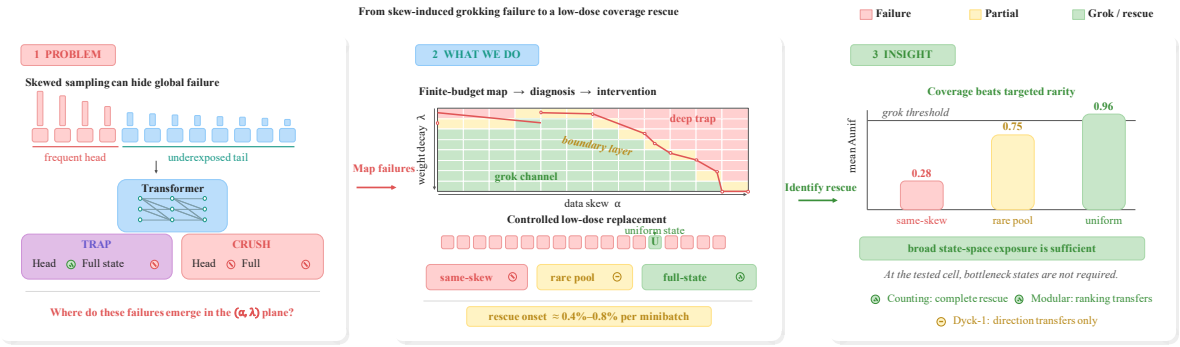


Figure 1: The framework of our proposed study.

1. An empirical, finite-budget (α, λ) phase diagram with two failure boundaries and a two-regime kinetic/deep-trap structure, reproduced at a second problem size with an exposure-matching protocol that avoids conflating training dynamics with the underlying failure structure (Section 5.1).
2. A low-dose rescue intervention validated by a controlled comparison ranking sample count, rarity, and distributional breadth against one another (Section 5.2).
3. A family-discrimination analysis of the boundary's shape, evaluated by a pre-specified held-out prediction test (Section 5.3).
4. A pre-specified local robustness study across width, learning rate, depth, and optimizer choice, with four of seven directional predictions confirmed at both tested boundary cells (Section 5.4), and transfer of the injection-ranking direction to a second arithmetic task, modular addition, under sample-count-matched controls (Section 5.5).

The scope is limited to the measured settings. The phase diagram is empirical and budget-dependent; it does not establish an asymptotic phase transition or critical exponent. The rescue experiment ranks the interventions we tested and does not establish necessity. Transfer is strongest on a second *arithmetic* task. On the non-arithmetic Dyck-1 forced-position task (Section 5.6), the failure topology and injection-ranking direction reappear, while complete rescue and budget-monotonic behavior do not.

2 Related Work

Grokking and its mechanisms. Power et al. (2022) established the toy algorithmic setting from which LARGE-COUNTER descends. Proposed mechanisms include weight-norm dynamics and a "Goldilocks zone" between memorization and confusion (Liu et al., 2022; 2023a), Fourier-structured progress measures (Nanda et al., 2023), competition between memorizing and generalizing circuits (Varma et al., 2023), group-theoretic representations (Chughtai et al., 2023), a lazy-to-rich transition with sample-complexity bounds (Mohamadi et al., 2024), implicit-bias changes under weight decay (Lyu et al., 2024), complexity and compression dynamics (Liu et al., 2023b; DeMoss et al., 2025), and representation-level precursors to generalization (Sivasankar, 2026; Howe, 2026). Koch & Ghosh (2025) connect grokking, double descent, and the information bottleneck through a two-phase timescale picture. These studies mainly ask what training dynamics produce grokking once the data distribution is fixed. Our injection experiments ask how changing coverage of that distribution changes whether grokking is reached.

Phase-transition framings. A parallel literature studies grokking with tools from statistical mechanics, including finite-size scaling and Binder-cumulant crossings (Bi et al., 2026), effective-dimensionality crossings (Wang, 2026), adaptive-kernel models of first-order transitions (Rubin et al., 2024), and spectral-entropy collapse of representation covariance (Khanh et al., 2026). Our (α, λ) map instead varies empirical skew and weight decay at fixed finite compute budgets. We use "phase diagram" for this measured boundary and do not estimate an asymptotic transition or critical exponent.

Rescuing and accelerating grokking. Prior interventions include knowledge distillation from an already-grokked model on a related distribution (Singh et al., 2025), embedding transfer from a weaker model (Xu et al., 2025), and learned gradient transformations (Zhou et al., 2025). Our intervention changes the examples presented during training. The matched controls in Section 5.2 compare extra sample count, targeted rarity, and breadth of the injection pool.

Weight decay as a modeling choice. We use AdamW (Loshchilov & Hutter, 2019) throughout; its implicit bias differs from that of SGD with momentum (Xie & Li, 2024), which we use as a comparison point in Section 5.4, and its role in grokking's timing has an independent analytic treatment orthogonal to our skew axis (Kim, 2026). We also compare our results with Doshi et al. (2024a)'s account of how weight decay affects memorizing and generalizing circuits.

Data skew and distribution shift. Carvalho et al. (2025) formalize grokking through train/test distribution shift induced by imbalanced sub-category sampling, closely matching our focus on skew. We add a continuous skew axis and measure the corresponding finite-budget boundary. Zhu et al. (2024) study a distinct data-quantity axis and critical dataset size as model size changes. Our sampling protocol also relates to long-tailed learning (Zhang et al., 2023; De Alvis & Seneviratne, 2024) and curriculum learning (Bengio et al., 2009). In our rescue experiments, broad injection works better than targeting rare examples alone (Section 5.2).

Task foundations. The LARGECOUNTER task uses the succinctness result of Bergsträßer et al. (2026) and follows the design of Zhao (2026). Our second and third tasks draw on the modular-arithmetic grokking literature (Gromov, 2023; Doshi et al., 2024b;a; McCracken et al., 2025) and the formal-language-recognition literature establishing Transformers' ability to recognize Dyck languages (Weiss et al., 2021; Ebrahimi et al., 2020; Bhattamishra et al., 2020; Yao et al., 2021; Merrill et al., 2022; Strobl et al., 2024).

3 Sampling Formalization and Exposure Consequences

We first formalize the sampling intervention independently of the optimization dynamics. The purpose of this section is limited: it characterizes how skew and uniform injection redistribute training exposure, but does not provide a theory predicting when grokking occurs.

Let $\mathcal{X} = \{0, \ldots, N-1\}$ denote the finite state space, where $N = 2^n$ for LARGECOUNTER. For skew parameter $\alpha \geq 0$, the baseline training distribution is

$$P_\alpha(x) = \frac{(x+1)^{-\alpha}}{\sum_{j=0}^{N-1}(j+1)^{-\alpha}}, \qquad x \in \mathcal{X}. \tag{1}$$

Thus P_0 is uniform, while increasing α concentrates training frequency on low-index states without changing the support: $P_\alpha(x) > 0$ for every $x \in \mathcal{X}$.

Uniform injection. Let B denote the minibatch size. At each optimization step, a minibatch is first sampled from P_α. With probability q, one slot is replaced by a state sampled uniformly from $\mathcal{X}$. The expected injected fraction of a minibatch is therefore

$$\delta = \frac{q}{B}. \tag{2}$$

Averaging over minibatch positions, the marginal sampling distribution becomes

$$Q_{\alpha,\delta}(x) = (1-\delta)P_\alpha(x) + \delta\frac{1}{N}. \tag{3}$$

The intervention therefore does not introduce states absent from the original support. Instead, it reallocates a small amount of probability mass from the distribution head toward states that are underrepresented relative to uniform sampling.

Exposure consequence. Let C_x be the number of times state x is presented during S optimization steps. Under the sampler above,

$$\mathbb{E}[C_x] = SB\,Q_{\alpha,\delta}(x) = S\left[(B-q)P_\alpha(x) + \frac{q}{N}\right]. \tag{4}$$

Consequently,

$$\frac{\partial\,\mathbb{E}[C_x]}{\partial\delta} = SB\left(\frac{1}{N} - P_\alpha(x)\right). \tag{5}$$

Uniform injection therefore increases expected exposure for every state with $P_\alpha(x) < 1/N$, while decreasing it for sufficiently frequent states.

Equations (3)–(5) motivate exposure as a descriptive axis for the experiments below, but they do not establish increased exposure as a causal mechanism of grokking. In particular, Appendix F shows that binary support coverage saturates substantially before rescue occurs. The empirical evidence is therefore more consistent with broad repeated exposure than with a simple unseen-state explanation.

4 FRAMEWORK OF THE EMPIRICAL STUDY

4.1 LARGECOUNTER TASK AND MODEL

We follow the previous protocol Zhao (2026): an autoregressive, big-endian binary counter over vocabulary $\{0, 1, \#\}$. Each training example is a single transition $x \to x + 1$: the input token sequence is $\mathrm{bin}(x) \# \mathrm{bin}((x + 1) \bmod 2^n)$, length $2n + 1$, with x the state being sampled under P_α; the loss is masked to only the n target-bit positions after #, so the model is never asked to predict x itself, only the ripple-carry successor. Big-endian order within each n-bit half is essential: it makes carry propagation non-local, requiring the model to resolve a low-bit carry before emitting a high-order bit that appears earlier in its own n-bit target, which is what creates a genuine grokking delay rather than a trivial two-state transduction. We use a causal decoder-only Transformer (Vaswani et al., 2017) with rotary position embeddings (Su et al., 2024), 2 layers, 4 heads, $d = 64$ unless otherwise noted, trained with AdamW (Loshchilov & Hutter, 2019) at default $\beta = (0.9, 0.999)$ with no dropout and no gradient clipping, learning rate 10^{-3} with a cosine-annealed schedule over the full step budget, batch size 128, and a $10\times$ multiplicative rescaling of every `nn.Linear` weight and bias after PyTorch's default initialization; this rescaling is used for LARGECOUNTER and Dyck-1, while modular addition (Section 5.5) uses the unscaled default. Evaluation is teacher-forced accuracy over the n target-bit positions per transition, so a transition counts as correct only if every target bit matches, not free-running/sampled decoding; A_{unif} and A_{hi} average this per-transition accuracy over the 2^n possible states x. We report results at $n = 10$ (1024 states) as the primary setting, with $n = 12$ used for an exposure-matched finite-size check.

4.2 CONTINUOUS FREQUENCY-SKEW PROTOCOL

We construct the skew axis with frequency-weighted minibatch sampling. Each training example is drawn with replacement according to $P_\alpha(x) \propto (x + 1)^{-\alpha}$; $\alpha = 0$ is uniform, and larger α concentrates mass on small-x states. We use frequency weighting because fixed-fraction subset selection does not change the relevant carry-chain composition: for a contiguous low-x block, the empirical trailing-ones distribution remains flat in α and matches the full state space. The rarity-versus-frequency effect studied here therefore comes from repeated exposure under P_α.

4.3 EVALUATION AND VERDICT RULE

We classify every trained model by two population-level accuracies: A_{unif}, whole-sequence accuracy averaged uniformly over the full state space, and A_{hi}, the same accuracy restricted to the highest-weight decile under P_α. A run is GROK if $A_{\mathrm{unif}} > 0.9$; TRAP if $A_{\mathrm{hi}} > 0.9$ and $A_{\mathrm{unif}} < 0.5$; CRUSH if $A_{\mathrm{hi}} < 0.9$ and $A_{\mathrm{unif}} < 0.5$; and PARTIAL otherwise. This rule is frozen prior to the experiments reported here and used unchanged throughout. Moving the A_{unif} cutoff to 0.85 or 0.95 shifts the overall grok fraction on the main grid by at most 6 percentage points ($0.516/0.508/0.452$ at cutoffs $0.85/0.9/0.95$) and the $\alpha = 2.5$ ceiling by less than 0.04, so the qualitative topology is not an artifact of the particular threshold.

4.4 EXPOSURE MATCHING ACROSS PROBLEM SIZES

Comparing training outcomes across problem sizes n requires care: doubling the state space at a fixed step budget halves the average number of times each state is visited, so an apparent boundary shift with n can reflect this exposure change rather than a change in the underlying failure structure. We therefore scale the step budget with n (specifically 2^n) whenever comparing across problem sizes, and report both exposure-matched and unmatched comparisons where relevant.

4.5 TRANSFER TASKS: MODULAR ADDITION AND DYCK-1

For modular addition, each example is the length-4 sequence $[a, b, \mathrm{SEP}, (a + b) \bmod p]$, with vocabulary $p + 1$ over $\mathbb{Z}_p \cup \{\mathrm{SEP}\}$, $p = 97$, and SEP token id p. Loss and accuracy are computed only at the final position, which predicts $(a + b) \bmod p$ from a, b, SEP; evaluation covers all p^2 operand pairs. We skew operand a under P_α and sample b uniformly. Skewing the combined pair index $ap + b$ would place 99.95% of the probability mass on one operand value at $\alpha = 2.5$ and collapse the task. The targeted pools follow the skewed operand: "tail" contains the rarest decile of a-values, and "head" the most frequent decile. The model, optimizer, batch size, and evaluation convention match §3.1. Modular addition uses the unscaled initialization, $n = 4$ as its task-specific input width, and a $20{,}000$-step budget; the failure region is already resolved at that budget, which is used throughout Section 5.5 and Figure 5.

For Dyck-1, we use sequence length $L = 10$, giving $C_{10} = 16{,}796$ valid paths of 10 pairs and 20 bracket tokens. The model, optimizer, batch size, and evaluation convention again match §3.1, with a $60{,}000$-step primary budget and

Table 1: Grid-resolution convergence: sparse pilot grid vs. densified grid ceiling estimates ($n = 10$, 60k budget). Estimates agree within the interpolation tolerance at every shared α.

α	Sparse-grid ceiling	Dense-grid ceiling	# points (sparse / dense)
2.0	0.305	0.305	9 / 11
2.25	0.124	0.123	10 / 11
2.5	0.086	0.089	10 / 11
2.75	< 0.030	< 0.020	10 / 13
2.9	< 0.020	< 0.020	4 / 6
3.0	< 0.020	< 0.020	9 / 9

an additional 240,000-step trajectory comparison in Section 5.6. The task is *forced-position next-token prediction*. Under the Dyck-1 grammar, a prefix forces an opening bracket at depth 0 and a closing bracket when the remaining length equals the current depth; other positions admit either bracket while preserving a completable path. Loss and accuracy are computed only at forced positions, so an incorrect forced prediction supplies the error signal that a recognizer would receive from an invalid string. Training frequency is skewed by maximum nesting depth $m(\pi)$ using $P_\alpha(\pi) \propto (\mathrm{rank}(m(\pi)) + 1)^{-\alpha}$, where rank orders paths by increasing $m(\pi)$. Shallow paths therefore receive low rank and high sampling probability, while deep paths receive high rank and low probability. Maximum depth plays the role of carry-chain length because deeper paths require longer-range tracking of nesting state. The "head" injection pool is the shallowest quintile of paths by maximum depth, and the "tail" pool is the deepest quintile. As in §3.2, the frozen four-way verdict uses A_{unif} over the full enumerated path set and A_{hi} over its highest-weight decile under P_α.

4.6 Pre-Specified Analyses

Two analyses in this paper were committed to before the corresponding data were collected, with the commitment recorded by file timestamp: predictions for eight held-out α values from the family-discrimination fit (Section 5.3), and seven directional robustness predictions plus two additivity-test cells (Section 5.4). We use "pre-specified" rather than "preregistered" throughout, since a filesystem timestamp is evidence of pre-data specification but not registry-grade provenance.

5 Empirical Phase Boundaries, Rescue, and Transfer

5.1 Skew Contracts the Grokking Region and Reveals Two Failure Regimes

Figure 2 maps grokkability across weight decay $\lambda \in [0.05, 0.5]$ and skew $\alpha \in [0, 3]$ at $n = 10$, with $\alpha \leq 1.5$ additionally probed up to $\lambda = 1.0$ to bracket the wider channel at low skew. The grok-channel weight-decay ceiling decreases monotonically with α: bootstrapped 90% intervals place the ceiling at 0.764, $[0.703, 0.851]$, at $\alpha = 1.5$ within the extended $\lambda \leq 1.0$ probe; 0.305, $[0.302, 0.401]$, at $\alpha = 2.0$; 0.190 at $\alpha = 2.1$; 0.123 at $\alpha = 2.25$; and 0.051, $[0.044, 0.060]$, at $\alpha = 2.7$. Beyond $\alpha = 2.7$, none of the tested weight decays down to 0.02 rescues generalization. Refining the grid by roughly a factor of three around the boundary gives ceiling estimates statistically consistent with the coarser grid at every shared α (Table 1), indicating that the near-vertical right edge is stable to grid resolution.

Each cell is 3–5 seeds; ceilings are the linear interpolation of mean A_{unif} across the λ grid crossing the 0.9 threshold, and 90% intervals are the 5th/95th percentiles of 2,000 seed-level bootstrap resamples of that interpolation (Appendix A gives the full per-cell seed counts).

The diagram separates two regimes. Near the boundary, more compute improves outcomes: all three cells tested at both 60,000 and 240,000 steps have higher mean accuracy at the longer budget, and one moves from partial accuracy to full generalization. Each budget has its own cosine-annealed schedule over the full run (§3.1), so these are independently scheduled training protocols rather than checkpoints from one trajectory. Deeper inside the failure region, at $\alpha = 2.25$ and $\alpha = 2.5$ with $\lambda = 0.3$, none of eight runs generalizes after 10^6 steps, four times the previously tested extended budget; accuracies remain between 0.35 and 0.80 with no upward trend as the budget increases. The exposure-matched $n = 12$ grid (Table 2) shows the same structure. At $\alpha = 2.0$, its ceiling is lower than the $n = 10$ ceiling by a factor of 1.42, so the two-regime pattern is not specific to $n = 10$.

The boundary also has a kinetic signature. At fixed $\lambda = 0.3$, the fraction of runs still ungeneralized after 240,000 steps rises monotonically across fifteen seeds per cell: 0% at $\alpha \leq 2.0$, 27% at $\alpha = 2.1$, 53% at $\alpha = 2.2$, and 60% at $\alpha = 2.25$ (Figure 3). At $\lambda = 0.05$, every seed eventually generalizes within budget, but the median time rises

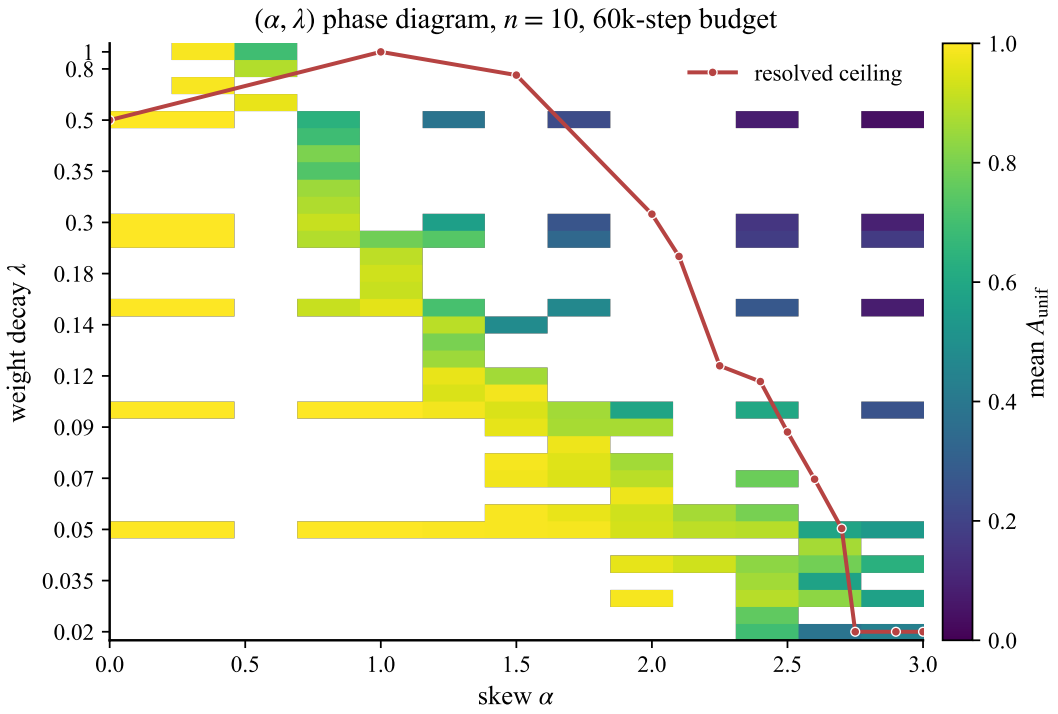


Figure 2: The (α, λ) phase diagram at $n = 10$, 60,000-step budget. Color encodes mean A_{unif} per cell; the red curve traces the resolved weight-decay ceiling.

Table 2: $n = 12$ exposure-matched grid (mean A_{unif}, 3 seeds/cell, 240,000-step budget), the full grid underlying the $n = 12$ replication.

α	λ=0.05	0.1	0.15	0.2	0.3	0.5
0.0	1.00	1.00	1.00	1.00	1.00	1.00
1.0	1.00	1.00	1.00	1.00	1.00	1.00
2.0	0.99	0.99	0.91	0.91	0.80	0.53
2.25	0.98	0.96	0.60	0.60	0.33	0.23
2.5	0.88	0.67	0.36	0.24	0.16	0.08
2.75	0.52	0.29	0.12	0.09	0.10	0.02
3.0	0.21	0.11	0.08	0.05	0.03	0.01

from 12,000 steps at $\alpha = 2.0$ to 171,250 steps at $\alpha = 2.75$. We use "critical-slowing-like" descriptively for this increase in failure probability and generalization time near the boundary; the data do not establish a divergence or critical exponent.

5.2 Low-Dose Broad Injection Rescues Failed Runs

We ran a second sweep with batch size 128 and 60,000 steps fixed. At each step, one slot is replaced by a uniformly sampled state with probability q, for an expected dose $q/128$; $q = 1$ exactly matches the batch-128 endpoint of the first sweep. Pooling 24 seeds per dose point across the same three α values, mean accuracy over $q \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$ rises as $0.478 \to 0.683 \to 0.846 \to 0.927 \to 0.924$. The generalizing fraction is $0/24$ at expected doses 0.078% and 0.195%, $7/24$ at 0.391%, $19/24$ at 0.586%, and $17/24$ at 0.781%. Thus the fixed-batch sweep shows a gradual rise and a plateau within sampling noise after $q = 0.75$. The sharper jump in the original batch-varying sweep is partly confounded with batch size and update count, so we use the fixed-batch sweep to characterize the shape of the dose response. Figure 4 retains the batch-varying sweep because that protocol is also used for the injection-ranking controls and the shortened-budget experiment below. Both sweeps place the onset of substantial rescue probability in the 0.4%–0.8% range; doses below 0.2% are insufficient in these experiments. This

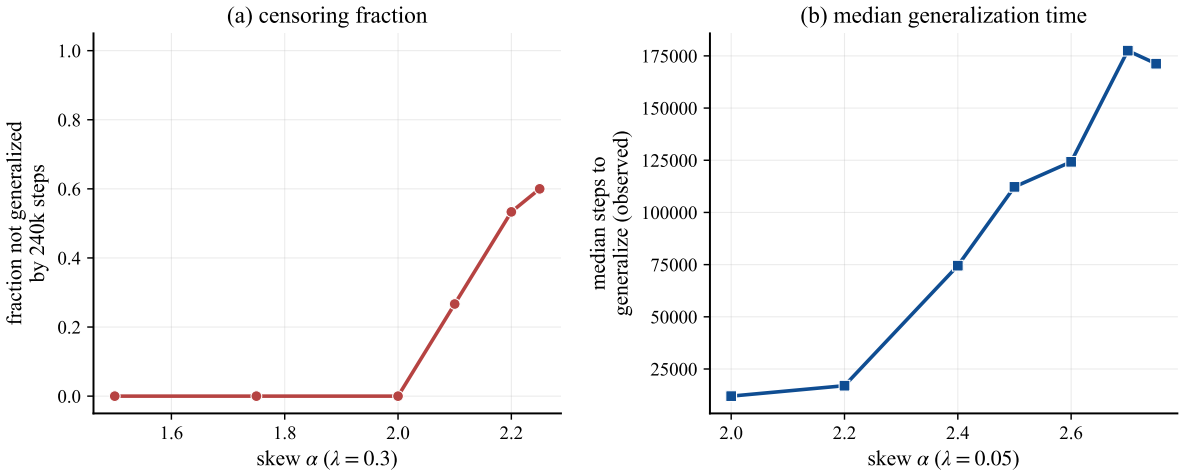


Figure 3: Kinetic signature of the boundary. (a) Censoring fraction (fraction not generalized by 240k steps), $\lambda = 0.3$, 15 seeds/cell. (b) Median generalization time (observed seeds only), $\lambda = 0.05$.

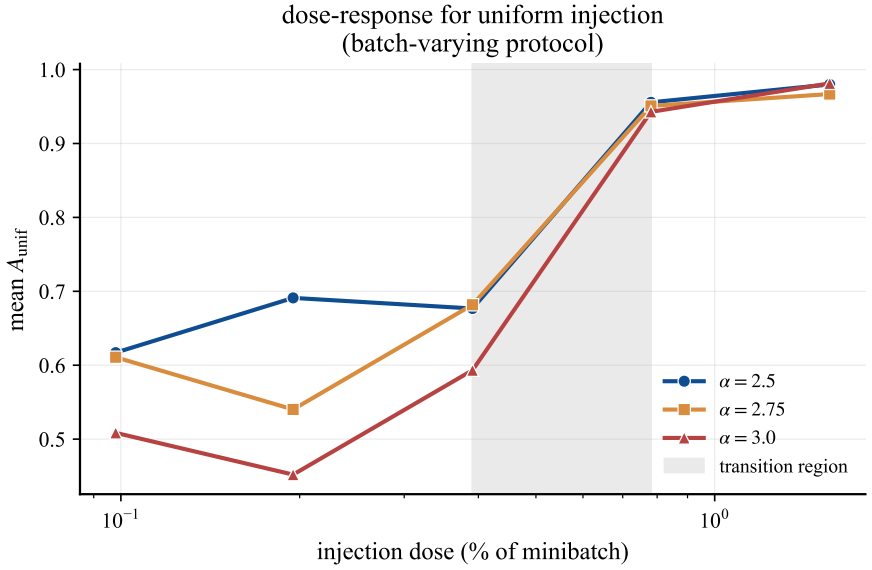


Figure 4: Dose-response curves for uniform injection at $\lambda = 0.3$, three skew values (batch-varying protocol). The shaded band marks the transition region.

absolute threshold is specific to the primary cosine-annealed protocol: under a constant learning rate, injection remains dose-beneficial but does not achieve complete rescue at the same sub-1% doses (Appendix E).

At $\alpha = 2.5$, $\lambda = 0.3$, three controls match the number of injected samples while changing where those samples come from. A minibatch of 128 replaces k existing slots, so $k = 1$ corresponds to $p = 1/128 \approx 0.78\%$ without changing batch size. Because P_α already assigns positive probability to every state, the controls differ in the distribution of the replacement slot. It is drawn either from the same head-concentrated distribution as the rest of the batch, from a targeted pool of rare states with short carry chains, or uniformly from the full state space. The targeted pool contains the $N/10$ lowest-probability states satisfying $c(x) < n/2$ and excludes all bottleneck states with $c(x) \geq n/2$.

Same-distribution replacement gives mean uniform accuracy 0.282, close to the zero-injection baseline. The rare-but-short pool reaches mean 0.750, with none of eight seeds crossing the generalization threshold. Full-state uniform injection reaches mean 0.956 and complete rescue. Among these three distributions, only full-state uniform injection rescues every seed.

Table 3: Deconfounded dose sweep: batch size (128) and step count (60,000) fixed throughout; expected dose $q/128$ implemented by probabilistic single-slot replacement. Pooled across $\alpha \in \{2.5, 2.75, 3.0\}$, 24 seeds per dose point (per-α breakdown in Appendix C, Table 10).

q	Expected dose	Fraction generalizing	Mean A_{unif}
0.10	0.078%	0/24	0.478
0.25	0.195%	0/24	0.683
0.50	0.391%	7/24	0.846
0.75	0.586%	19/24	0.927
1.00	0.781%	17/24	0.924

Table 4: Pre-specified held-out predictions vs. measured ceilings at the 5 of 8 α values whose tested weight-decay grid brackets the 0.9 threshold (interpolable); the remaining 3 ($\alpha = 1.7, 1.9, 2.65$) fall outside their tested λ range and are not resolved here.

α	$E_\alpha[c]$ prediction	$2^{-\alpha}$ prediction	Measured ceiling
2.05	0.173	0.168	0.155
2.15	0.158	0.156	0.167
2.35	0.133	0.136	0.126
2.45	0.122	0.127	0.096
2.55	0.112	0.119	0.089

This comparison changes breadth, entropy, and state composition together. Appendix D holds composition fixed by excluding bottleneck states at every setting and varies only the pool size among the remaining 992 states. Generalization rises monotonically with pool size and reaches complete rescue at $K = 600$; the $K = 992$ non-bottleneck pool has mean accuracy 0.949, close to the 0.956 mean from full-state uniform injection. Complete rescue therefore occurs without injecting bottleneck states at this cell. The $K = 102$ rare-but-short pool appears limited by its breadth. Entropy still increases with pool size, so these experiments do not separate breadth from entropy. Rescue also succeeds with 15,000 injected samples at a shortened 15,000-step budget. Under the original batch-varying protocol, increasing the dose to $k = 2$ per 128, $p{\approx}1.56\%$, makes all eight seeds generalize at every tested α, consistent with the plateau beyond $q = 0.75$ in the fixed-batch sweep.

5.3 Empirical Boundary Shape and Held-Out Prediction

We compare five single-scale functions of α with the measured weight-decay ceiling. The first is the expected carry-chain length, $E_\alpha[c] = \sum_x P_\alpha(x)\, c(x)$, where $c(x)$ is the trailing-ones count of x and $c(2^n - 1) = n$. This is the same carry quantity used to define the "tail" and "rare-but-short-carry" pools in Section 5.2. The other candidates are $2^{-\alpha}$, two tail masses $H_{\text{lo}} = \sum_{x \geq N/2} P_\alpha(x)$ and $H_{\text{tail10}} = \sum_{x \geq 9N/10} P_\alpha(x)$, and the rarest-state probability $p_\star = P_\alpha(N-1)$, with $N = 2^n$. Each candidate is fit as ceiling$(\alpha) \approx C \cdot f(\alpha)$ using one scale constant C, chosen by least squares in log space on the eight measured values $\alpha \in \{1.5, 2.0, 2.1, 2.25, 2.4, 2.5, 2.6, 2.7\}$. We then freeze C and evaluate eight pre-specified held-out α values.

Three held-out values, $\alpha = 1.7$, 1.9, and 2.65, cannot be interpolated because the tested λ grid does not bracket the 0.9 threshold: all three sampled λ values lie above threshold at $\alpha = 1.7$ and 1.9, while all three lie below threshold at $\alpha = 2.65$. Table 4 reports the remaining five points without extrapolating beyond measured λ. On these points, $E_\alpha[c]$ and $2^{-\alpha}$ have mean prediction errors of 15% and 18%, with maximum errors of 27% and 33%. Both errors are smaller than the corresponding in-sample fitting errors on the original eight-point grid. $E_\alpha[c]$ is better at four of five held-out points, but the absolute error difference never exceeds 7.4 percentage points, which is too small to distinguish the two candidates here. The other three candidates fit substantially worse. For example, H_{lo} predicts 0.533 at $\alpha = 2.05$ against a measured ceiling of 0.155, and $p_\star$ has an error $20.5\times$ that of the best candidate. The held-out test supports a single-scale description of the boundary while leaving the governing function unresolved.

5.4 Local Robustness of the Boundary

Before running these experiments, we specified directional predictions for seven single-factor perturbations at two boundary-adjacent cells. The d=64, lr= 10^{-3}, 2-layer AdamW baseline from §3.1 has mean accuracy 0.974 at

Table 5: Pre-specified directional predictions and outcomes at two boundary-adjacent cells (α=2.5, λ=0.085, baseline mean 0.974; and α=2.25, λ=0.125, baseline mean 0.853), 5 seeds each. Confirmation criterion: mean accuracy moves in the predicted direction at both cells. Four of seven directional predictions confirmed at both cells; the two interaction cells test additivity and carry no direction.

Perturbation	Predicted direction	Mean accuracy (both cells)	Outcome
Width d=32	harder	0.384, 0.357	confirmed
Width d=128	easier	0.951, 0.992	confirmed at cell 2 only
Learning rate 5×10^{-4}	easier	0.970, 0.979	confirmed at cell 2 only
Learning rate 2×10^{-3}	harder	0.845, 0.813	confirmed
Depth 1 layer	harder	0.731, 0.828	confirmed
Depth 3 layers	easier	0.974, 0.807	not confirmed
d=32, lr= 5×10^{-4} (interaction)	no direction (additivity test)	0.444, 0.623	width dominates
d=128, lr= 2×10^{-3} (interaction)	no direction (additivity test)	0.930, 0.931	width dominates
SGD, momentum 0.9	substantially worse	0.001, 0.001	confirmed

α=2.5, λ=0.085 and 0.853 at α=2.25, λ=0.125. We predicted d=32 to be harder and d=128 easier; learning rate 5×10^{-4} easier and 2×10^{-3} harder; depth 1 layer harder and 3 layers easier; and SGD with momentum 0.9 substantially worse at otherwise unchanged hyperparameters. Two additional cells, d=32, lr= 5×10^{-4} and d=128, lr= 2×10^{-3}, test additivity of width and learning-rate effects and carry no directional prediction. A prediction is confirmed only when mean accuracy moves in the predicted direction at both cells (Table 5).

Four of the seven directional predictions meet this criterion: d=32, learning rate 2×10^{-3}, depth 1, and SGD with momentum. SGD reaches mean accuracy 0.001 at both cells. The d=128 and 5×10^{-4} learning-rate predictions hold only at α=2.25, λ=0.125. At α=2.5, λ=0.085, the baseline mean is already 0.974, and both perturbations produce a small decrease. The 3-layer model does not improve either cell: its mean is essentially unchanged at the first and lower at the second. Because SGD with momentum was not independently tuned, this comparison shows optimizer dependence under the tested hyperparameters rather than a controlled optimizer ranking.

5.5 Rescue Ranking Transfers to Modular Addition

We repeat the injection comparison from Section 5.2 on modular addition, $p = 97$, at $\alpha = 3.0$, $\lambda = 0.3$, using $k = 1$ per batch of 128 and the same architecture, optimizer, and evaluation convention (§3.5). Here the targeted analogues are rare, large-operand states and the most frequent states. The zero-injection baseline has mean accuracy 0.413, with 0/3 seeds generalizing. Uniform injection reaches mean 0.934 with 3/5 seeds generalizing; rare large-operand injection reaches 0.721 with 0/5; and frequent-state injection reaches 0.396 with 0/5, close to the baseline (Figure 5). The ordering therefore transfers under sample-count-matched controls, although uniform injection is less reliable than on LARGECOUNTER, where it reaches 8/8.

5.6 Dyck-1 Stress Test: Partial Transfer and Late Collapse

Because modular addition is structurally close to LARGECOUNTER, we also test Dyck-1 forced-position prediction (§3.5), a non-arithmetic task whose training frequency is skewed by maximum nesting depth. The formal boundary grid contains 540 runs with ten seeds per cell. As skew and weight decay increase, a two-dimensional failure region appears again. Its deepest cells collapse to the same degenerate accuracy across independent seeds, including a seed that had not been run previously (Figure 6).

Uniform injection improves the deep-collapse cell only partially: mean accuracy rises to 0.398, and 0/5 seeds reach the generalization threshold. Targeted and frequent-state injection both remain at the collapsed mean of 0.289. The same ranking direction is present, but the rescue magnitude is much smaller than on the two arithmetic tasks. At one boundary cell, the generalizing count changes from 2/3 at 60,000 steps to 0/3 at 240,000. The full trajectories (Figure 7) show why: the shorter runs stop during a metastable high-accuracy plateau, while the longer runs continue into a later collapse. This behavior is consistent with independently reported anti-grokking dynamics, where test

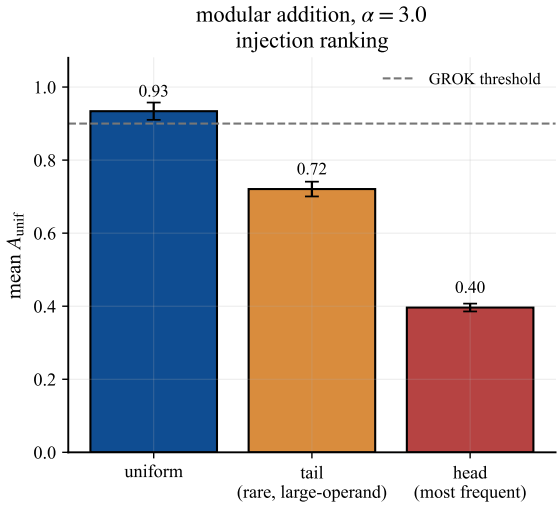


Figure 5: Injection ranking on modular addition, $\alpha = 3.0$, $\lambda = 0.3$, $k = 1$. The mean value are counted over 5 seeds.

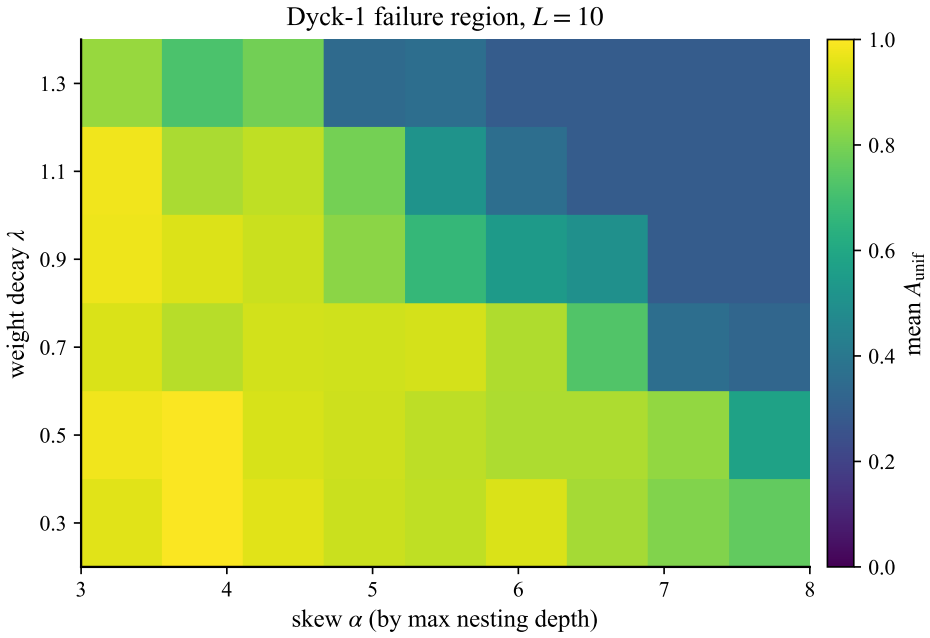


Figure 6: Dyck-1 formal boundary grid, $L = 10$, 540 runs, ten seeds per cell.

accuracy returns toward chance after earlier generalization (Prakash & Martin, 2026a;b). Appendix B reports the full Dyck-1 boundary and rescue data.

6 Discussion

Two empirical results reverse the earlier proposal: the weight-decay ceiling falls as skew increases, and broad full-state injection outperforms a targeted-rarity curriculum (Section 5.2). These reversals are why we treat the boundary and rescue mechanism as measured properties of this setup rather than consequences of a preselected theory. The additional controls sharpen this interpretation. Sampler replay shows that binary support coverage saturates before rescue, favoring broad repeated exposure over a simple unseen-state account (Appendix F). Removing the $10\times$ initialization rescaling preserves both the degradation with increasing skew and full-dose uniform rescue, while changing some

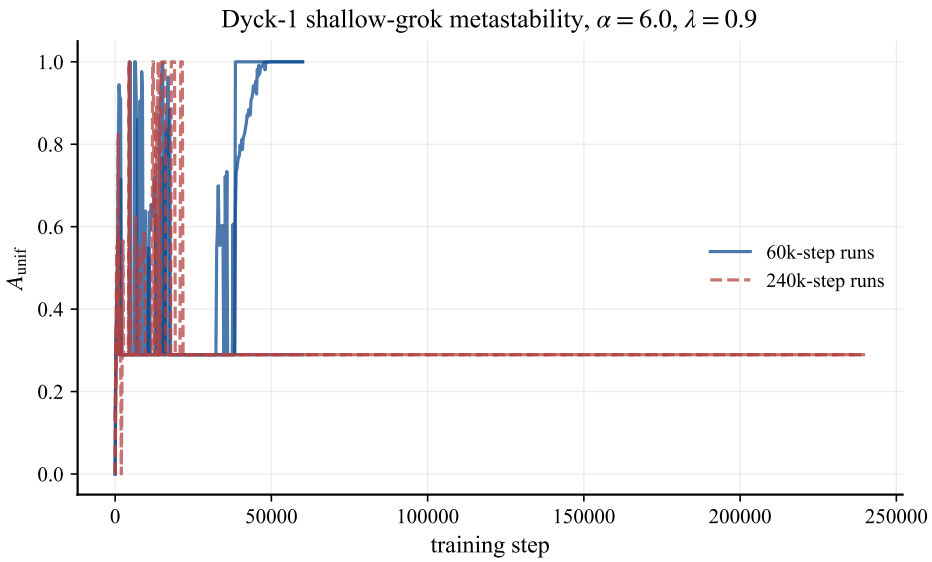


Figure 7: Full accuracy trajectories at $\alpha = 6.0$, $\lambda = 0.9$: 60k-step runs (solid) that stop inside the metastable plateau vs. 240k-step runs (dashed) on the same configuration that continue past it into collapse.

Table 6: Comparison to the nearest prior work along the axes this paper contributes.

Work	Skew axis	Finite-budget map	Rescue controls	Exposure match	Task transfer	Toy scale
Zhao (2026)	discrete (2 pts)	—	—	—	—	✓
Liu et al. (2022; 2023a)	data fraction	✓	—	—	✓	✓
Singh et al. (2025)	data quantity	—	✓(KD)	—	—	✓
Bi et al. (2026)	group order	✓(critical)	—	—	—	✓
This paper	continuous α	✓	✓(ranked)	✓	✓(2 tasks)	✓

near-boundary outcomes (Appendix **??**). Together with the learning-rate control, these results suggest that the qualitative exposure intervention is more stable than the absolute location of the empirical boundary or rescue threshold.

Table 6 summarizes how the nearest prior work compares to this paper's contributions along the axes of skew, finite-budget mapping, rescue controls, exposure matching, task transfer, and scale.

6.1 Limitations

All results are at $n \leq 12$ and $d \leq 128$ within one decoder-only Transformer family. The family-discrimination analysis in Section 5.3 supports a single-scale model class, but the two best candidates are too close on the held-out points to select a unique law. The local robustness study confirms 4 of 7 directional predictions (Section 5.4). The injection experiments show that sufficiently broad non-bottleneck pools can rescue the tested cell (Section 5.2, Appendix D), while breadth and entropy remain coupled. The absolute boundary and rescue threshold also depend on optimization details. A constant learning rate changes late-training kinetics and prevents complete rescue at the tested sub-1% doses, while removing the $10\times$ initialization rescaling changes some near-boundary success rates. Importantly, the skew ordering and broad-injection benefit persist under the standard initialization (Appendices E and **??**).

7 Conclusion

Across data skew and weight decay, LARGECOUNTER shows a finite-budget boundary with a compute-sensitive edge and a deep-trap interior. Uniform full-state injection rescues both failure modes and outperforms the tested sample-count and targeted-rarity controls. The same intervention ranking appears on modular addition under matched

sample counts. Dyck-1 preserves the failure topology and ranking direction, but complete rescue and monotonic improvement with longer training do not carry over. Sampler-level analysis further shows that binary state coverage saturates before rescue, pointing instead to broad repeated exposure as the more informative empirical descriptor.

REFERENCES

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Pascal Bergsträßer, Ryan Cotterell, and Anthony W Lin. Transformers are inherently succinct. In *International Conference on Learning Representations*, 2026.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, 2020.

Yuda Bi, Chenyu Zhang, Qiheng Wang, and Vince D Calhoun. Grokking as a falsifiable finite-size transition. *arXiv preprint arXiv:2603.24746*, 2026.

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

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In *International Conference on Machine Learning*, pp. 6243–6267, 2023.

Charika De Alvis and Suranga Seneviratne. A survey of deep long-tail classification advancements. *arXiv preprint arXiv:2404.15593*, 2024.

Branton DeMoss, Silvia Sapora, Jakob Foerster, Nick Hawes, and Ingmar Posner. The complexity dynamics of grokking. *Physica D: Nonlinear Phenomena*, pp. 134859, 2025.

Darshil Doshi, Aritra Das, Tianyu He, and Andrey Gromov. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets. In *International Conference on Learning Representations*, 2024a.

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

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4301–4306, 2020.

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

Gunner Levi Howe. Structure-specific representational priors causally control the grokking delay. *arXiv preprint arXiv:2607.04333*, 2026.

Truong Xuan Khanh, Truong Quynh Hoa, Luu Duc Trung, and Phan Thanh Duc. Spectral entropy collapse as a phase transition in delayed generalisation: An interventional and predictive framework for grokkin. *arXiv preprint arXiv:2604.13123*, 2026.

Taeyoung Kim. Grokking on the weight-decay clock: A rate hierarchy from softly broken symmetries. *arXiv preprint arXiv:2607.23967*, 2026.

Robert de Mello Koch and Animik Ghosh. Two-phase perspective on deep learning dynamics. *Physical Review E*, 112(2):025307, 2025.

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35: 34651–34663, 2022.

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

Ziming Liu, Ziqian Zhong, and Max Tegmark. Grokking as compression: A nonlinear complexity perspective. *arXiv preprint arXiv:2310.05918*, 2023b.

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

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon Du, Jason Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *International Conference on Learning Representations*, 2024.

Gavin McCracken, Gabriela Moisescu-Pareja, Vincent Létourneau, Doina Precup, and Jonathan Love. Uncovering a universal abstract algorithm for modular addition in neural networks. *arXiv preprint arXiv:2505.18266*, 2025.

William Merrill, Ashish Sabharwal, and Noah A Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022.

Mohamad Amin Mohamadi, Zhiyuan Li, Lei Wu, and Danica J. Sutherland. Why do you grok? A theoretical analysis on grokking modular addition. In *Proceedings of the 41st International Conference on Machine Learning*, 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*, 2023.

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.

Hari K Prakash and Charles H Martin. Late-stage generalization collapse in grokking: detecting anti-grokking with weightwatcher. *arXiv preprint arXiv:2602.02859*, 2026a.

Hari K Prakash and Charles H Martin. Detecting overfitting in neural networks during long-horizon grokking using random matrix theory. *arXiv preprint arXiv:2605.12394*, 2026b.

Noa Rubin, Inbar Seroussi, and Zohar Ringel. Grokking as a first order phase transition in two layer networks. In *International Conference on Learning Representations*, 2024.

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.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? a survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ping Wang. Grokking as dimensional phase transition in neural networks. *arXiv preprint arXiv:2604.04655*, 2026.

Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In *International Conference on Machine Learning*, pp. 11080–11090, 2021.

Shuo Xie and Zhiyuan Li. Implicit bias of adamw: ℓ_∞-norm constrained optimization. In *International Conference on Machine Learning*, 2024.

Zhiwei Xu, Zhiyu Ni, Yixin Wang, and Wei Hu. Let me grok for you: Accelerating grokking via embedding transfer from a weaker model. In *International Conference on Learning Representations*, 2025.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3770–3785, 2021.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816, 2023.

Yizheng Zhao. Do transformers grok succinct algorithms? mechanistic evidence for counting circuits. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26114–26130, 2026.

Xinyu Zhou, Simin Fan, Martin Jaggi, and Jie Fu. Neuralgrok: Accelerate grokking by neural gradient transformation. *arXiv preprint arXiv:2504.17243*, 2025.

Xuekai Zhu, Yao Fu, Bowen Zhou, and Zhouhan Lin. Critical data size of language models from a grokking perspective. *arXiv preprint arXiv:2401.10463*, 2024.

A PER-CELL SEED COUNTS, (α, λ) PHASE DIAGRAM

Table 7 gives the number of distinct λ points and the seed-count range at each α underlying Figure 2's ceiling curve (60,000-step budget, $n = 10$). Cells near a resolved ceiling were sampled at up to 10 seeds; exploratory cells away from the boundary used 3.

Table 7: Per-α grid density and seed-count range, $n = 10$, 60k-step budget.

α	# λ points	Min seeds/cell	Max seeds/cell
0.0	6	3	3
1.0	8	3	3
1.5	3	3	3
2.0	11	3	5
2.1	7	3	5
2.25	11	3	5
2.4	9	3	5
2.5	11	3	5
2.6	9	3	5
2.7	3	5	5
2.75	13	3	10
2.9	6	3	8
3.0	9	3	6

B DYCK-1 FORMAL BOUNDARY GRID AND RESCUE DATA

Table 8 gives the full (α, λ) mean A_{unif} grid underlying Figure 6 (10 seeds per cell, 540 runs total). Table 9 gives the injection-ranking means underlying Section 5.6's rescue comparison at the deep-collapse cell $\alpha = 5.5$, $\lambda = 1.3$.

Table 8: Dyck-1 formal boundary grid: mean A_{unif}, $L = 10$, 10 seeds/cell.

α	λ=0.3	0.5	0.7	0.9	1.1	1.3
3.0	0.96	0.98	0.94	0.97	0.98	0.85
3.5	1.00	0.99	0.89	0.95	0.87	0.71
4.0	0.95	0.94	0.93	0.92	0.90	0.79
4.5	0.92	0.92	0.93	0.83	0.79	0.34
5.0	0.90	0.90	0.93	0.67	0.51	0.36
5.5	0.94	0.88	0.88	0.55	0.36	0.29
6.0	0.86	0.88	0.73	0.50	0.29	0.29
7.0	0.81	0.84	0.36	0.29	0.29	0.29
8.0	0.76	0.58	0.33	0.29	0.29	0.29

C PER-α BREAKDOWN OF THE DECONFOUNDED DOSE SWEEP

Table 3 pools 24 seeds per dose point across three α values; Table 10 gives the same sweep broken out per α (8 seeds/cell).

Table 9: Dyck-1 injection ranking at the deep-collapse cell ($\alpha = 5.5$, $\lambda = 1.3$), $k = 1$ per batch of 128.

Injection	Mean A_{unif}	Seeds at threshold	n
none ($k = 0$)	0.289	0/3	3
head	0.289	0/5	5
tail	0.289	0/5	5
uniform	0.398	0/5	5

Table 10: Deconfounded dose sweep by α: fraction generalizing / mean A_{unif}, 8 seeds/cell.

α	q=0.10	q=0.25	q=0.50	q=0.75	q=1.00
2.5	0/8 (0.551)	0/8 (0.725)	3/8 (0.886)	7/8 (0.942)	7/8 (0.947)
2.75	0/8 (0.469)	0/8 (0.728)	3/8 (0.839)	6/8 (0.922)	5/8 (0.921)
3.0	0/8 (0.412)	0/8 (0.594)	1/8 (0.811)	6/8 (0.918)	5/8 (0.906)

D BREADTH VERSUS COMPOSITION IN THE INJECTION POOL

The rescue comparison in Section 5.2 shows that a targeted pool of the $N/10 = 102$ rarest non-bottleneck states, §3.4's rare-but-short-carry pool, excluding all bottleneck states with carry length $c(x) \geq n/2$, achieves only partial credit, mean 0.750, while injection drawn from the full state space, which does include the 32 bottleneck states, achieves complete rescue, mean 0.956. Because these two conditions differ in both breadth, 102 vs. 1024 candidate states, and composition, bottleneck states excluded vs. included, this comparison alone cannot say which property is responsible.

We isolate breadth by holding composition fixed: bottleneck states, $c(x) \geq n/2$, 32 of 1024 states, are excluded from the injection pool at every setting, and only the pool size K among the remaining 992 non-bottleneck states is varied, $K \in \{102, 300, 600, 992\}$, the K non-bottleneck states with the lowest P_α weight, at the same cell as the rescue battery, $\alpha = 2.5$, $\lambda = 0.3$, $k = 1$ per batch of 128, 8 seeds. $K = 102$ reproduces the rare-but-short-carry pool exactly.

Table 11: Breadth-only sweep with bottleneck states permanently excluded from the injection pool: fraction generalizing and mean A_{unif}, 8 seeds/cell.

K (non-bottleneck pool size)	Fraction generalizing	Mean A_{unif}
102	0/8	0.750
300	1/8	0.852
600	8/8	0.947
992	7/8	0.949

Generalization rises monotonically with K and reaches complete rescue at $K = 600$. At $K = 992$, the pool contains every non-bottleneck state and still excludes every bottleneck state; its mean accuracy, 0.949, statistically matches the 0.956 mean from full-state uniform injection. Complete rescue therefore occurs without bottleneck-state injection at this cell. The original $K = 102$ rare-but-short-carry pool appears too narrow to produce the same effect.

E SENSITIVITY TO THE LEARNING-RATE SCHEDULE

Under a constant learning rate, the deep-failure cell remains poor throughout training, whereas the boundary-adjacent cells exhibit substantially less stable behavior than under cosine annealing. In particular, longer training does not uniformly improve A_{unif} at the boundary. This indicates that the absolute boundary location and late-training kinetics are schedule-dependent.

As Table 13 illustrates, uniform injection remains beneficial when the learning rate is held fixed at 10^{-3} throughout training: mean A_{unif} increases monotonically from 0.081 without injection to 0.491 at the largest tested dose. However, none of the five seeds at any sub-1% dose satisfies the frozen grokking criterion. Thus, the directional benefit of broad injection persists without cosine annealing, whereas the absolute rescue threshold and the attainment of complete rescue are schedule-dependent.

Table 12: Constant-learning-rate control for the kinetic comparison. All runs use a fixed learning rate of 10^{-3} and are evaluated along the same 240k-step trajectory, removing the budget-dependent cosine-schedule confound. Values are mean $\pm$ s.d. over five seeds.

α	λ	60k	120k	240k
2.00	0.30	0.340 ± 0.091	0.428 ± 0.077	0.479 ± 0.068
2.10	0.30	0.401 ± 0.096	0.312 ± 0.048	0.315 ± 0.089
2.50	0.30	0.081 ± 0.032	0.115 ± 0.020	0.125 ± 0.025

Table 13: Constant-learning-rate control for uniform injection at $\alpha = 2.5$, $\lambda = 0.3$. Batch size and training length are fixed at 128 and 60k steps, respectively. Values are averaged over five seeds.

q	Expected dose	Mean A_{unif}	Grok seeds
0	0	0.081	0/5
0.25	0.195%	0.223	0/5
0.50	0.391%	0.290	0/5
0.75	0.586%	0.380	0/5
1.00	0.781%	0.491	0/5

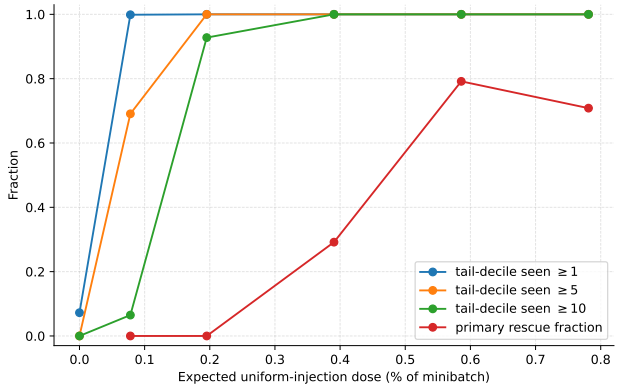


Figure 8: Sampler-level exposure diagnostics for the fixed-batch uniform-injection sweep. The blue, orange, and green curves show the fraction of states in the rarest decile under P_α that are observed at least 1, 5, and 10 times, respectively, pooled over $\alpha \in \{2.5, 2.75, 3.0\}$ and eight sampler seeds per α. The red curve reports the grokking fraction from the corresponding primary cosine-annealed experiments. Rare-tail support coverage saturates at substantially smaller doses than those at which rescue appears, arguing against a simple unseen-state explanation and indicating that binary support coverage alone is insufficient to account for rescue.

F Sampler-Level Coverage and Repeated Exposure

To examine the coverage interpretation more directly, we replay the fixed-batch sampler without retraining the model and record how often each state is presented during training. Figure 8 separates binary support coverage from repeated exposure. At $q = 0.1$, corresponding to only 0.078% expected uniform injection per minibatch, 99.9% of the rarest-decile states have already been observed at least once, yet none of the 24 corresponding primary runs groks. At $q = 0.25$ (0.195% expected dose), rare-tail support coverage reaches 100%, and 92.8% of the rarest-decile states are observed at least ten times, while the grokking fraction remains $0/24$. Rescue appears only at larger doses, reaching $7/24$ at $q = 0.5$ and $19/24$ at $q = 0.75$.

These observations argue against a simple explanation in which rescue occurs merely because previously unseen rare states become represented in the training stream. Instead, the intervention continues to increase the frequency with which broadly distributed rare states are revisited after binary support coverage has already saturated. We therefore interpret the result more narrowly in terms of *broad repeated exposure*, rather than binary support coverage alone. This analysis is descriptive: injection dose, exposure frequency, and optimization dynamics remain coupled, so the data do not establish repeated exposure as the unique causal mechanism of rescue.

Table 14: Standard-initialization control with the $10\times$ linear-layer rescaling removed (init scale 1.0). All other LARGECOUNTER settings match the primary protocol, including the cosine schedule and 60,000-step budget. Values are mean $\pm$ s.d. over five seeds.

α	λ	q	A_{unif}	A_{hi}	A_{lo}	Grok
0.0	.300	0	1.000±.000	1.000±.000	1.000±.000	5/5
2.0	.300	0	0.849±.084	0.997±.002	0.779±.113	2/5
2.5	.085	0	0.975±.020	1.000±.000	0.961±.031	5/5
2.5	.300	0	0.232±.094	0.872±.080	0.027±.037	0/5
2.5	.300	1	0.962±.019	0.984±.011	0.961±.019	5/5

Table 15: Standard-initialization mini-grid (init scale 1.0). Entries report mean A_{unif} followed by the number of grokking seeds in parentheses; three seeds per cell.

α	$\lambda = .05$	$\lambda = .10$	$\lambda = .30$
0.0	1.000 (3/3)	1.000 (3/3)	1.000 (3/3)
2.0	0.999 (3/3)	0.994 (3/3)	0.813 (1/3)
2.5	0.956 (2/3)	0.881 (1/3)	0.254 (0/3)

G INITIALIZATION-SCALE CONTROL

To test whether the observed failure structure depends on this choice, we remove this rescaling (init scale 1.0) while keeping the architecture, optimizer, cosine schedule, batch size, training budget, and frozen verdict rule unchanged. Table 14 shows that the qualitative skew dependence survives the change in initialization. At $\lambda = 0.3$, mean A_{unif} decreases from essentially 1.0 at $\alpha = 0$ to 0.849 at $\alpha = 2.0$ and 0.232 at $\alpha = 2.5$. The frequent head remains substantially easier than the rare tail: at $\alpha = 2.0$, $A_{\text{hi}} = 0.997$ while $A_{\text{lo}} = 0.779$, and at $\alpha = 2.5$ the gap widens to 0.872 versus 0.027. Thus stronger frequency skew continues to suppress global generalization under the default initialization, although the exact near-boundary outcomes change.

The rescue effect also persists. At $(\alpha, \lambda) = (2.5, 0.3)$, none of the five uninjected runs groks, whereas uniform single-slot injection with $q = 1$ raises mean A_{unif} from 0.232 to 0.962 and makes all five seeds grok. Conversely, the lower-weight-decay probe $(\alpha, \lambda) = (2.5, 0.085)$ generalizes in all five seeds without injection. The $10\times$ rescaling is therefore not required for either skew-induced failure or broad-injection rescue, although it does affect quantitative training dynamics and some near-boundary success rates.

A small (α, λ) grid gives the same qualitative picture: low-skew cells grok throughout the tested weight-decay range, whereas the grokking region contracts progressively as α increases (Table 15).

Figure 9 visualizes this initialization control: stronger skew progressively degrades uniform generalization and moves runs toward the failure regions under the frozen verdict rule.

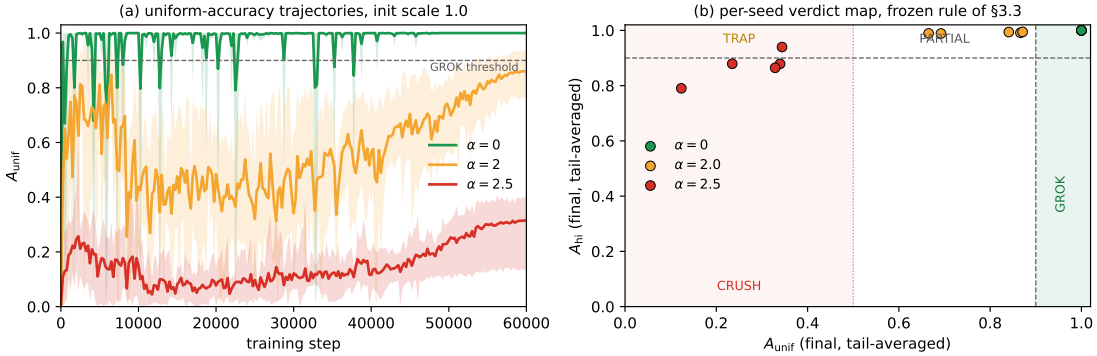


Figure 9: Skew transect with the default initialization (init scale 1.0) at $\lambda = 0.3$, 60k steps. (a) Uniform-accuracy trajectories, mean over five seeds with min–max band. (b) Per-seed outcomes under the frozen verdict rule.