AdamW Maintains What Delays Grokking

Anonymous authors
Paper under double-blind review

Abstract

A network that groks fits its training data thousands of updates before it generalizes. What keeps it waiting? We find that the wait depends on which content the network's embedding holds once the training data are fit, and that the AdamW updates that keep rebuilding this content against weight decay lengthen it. In one- and two-layer Transformers learning modular arithmetic, the operand embedding decomposes into Fourier frequencies, so we can fork a run and change one thing: delete what a set of frequencies holds, put it back, or stop the optimizer from rebuilding it. Masking a random half of the frequencies during training halves the time to generalize, although the normally trained model relies on that half, so removing what a trained model relies on need not slow its learning. In one-layer models, putting back what those frequencies held before generalization, the network's own early content, into the same frequencies of the same model state delays generalization by over two thousand updates; a scrambled copy of equal energy, the network's later content, or another network's early content barely delays it. Weight decay alone would erase most of this early content within 2,000 updates, yet in ordinary training it keeps its size: AdamW's updates put back what decay removes. After the same deletion, blocking the rebuild brings generalization forward beyond deletion alone and beyond a control that shrinks each update to the norm blocking would leave. Over that control, the paired median gain is 850 updates at one layer and 800 at two, both on fresh seeds. Deleting the content and blocking its rebuild, which acts on a single direction of the embedding, recovers about 0.9 of the speed-up from masking. Part of the wait thus lies in early embedding content that AdamW keeps in place.

1 Introduction

A network can fit its training data and then wait thousands of updates before it generalizes (Power et al., 2022). What is it doing during that wait? Existing accounts point to weight decay and the optimizer: the memorizing solution is cleaned up as a more efficient generalizing one forms (Nanda et al., 2023; Varma et al., 2023), the delay tracks how long decay takes to shrink the weight norm from its memorizing to its generalizing level (Truong et al., 2026b), and removing the radial part of each gradient removes the delay (Prieto et al., 2025). They motivate a local question: how does the content the embedding holds when the training data are fit evolve during the wait? At our learning rate and weight decay, anything in the embedding that no update puts back decays to 0.135 of its size within 2,000 updates, yet what the embedding holds when the training data are fit, which we call its *early content*, is still at full size thousands of updates later (Figure 2a; in a random half of its Fourier frequencies, Section 2). Something is putting it back. Why does the early content survive decay, and does keeping it there delay generalization? Ablating a trained model cannot answer either question: ablation shows what the finished model relies on now (Wang et al., 2023; Conmy et al., 2023; Liu, 2026), and a part's effect on learning need not match its role at the end (Chen et al., 2024). We ask which content delays generalization, what keeps it there, and how much blocking that maintenance shortens the wait.

The question reaches beyond grokking: whenever a network has already fit its data one way and a better solution must form afterwards, something decides how long the old fit stays in the way. Grokking on modular arithmetic lets us answer it with clean interventions: the operand embedding decomposes into Fourier frequencies, the same ones in which the known solutions are written (Nanda

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.

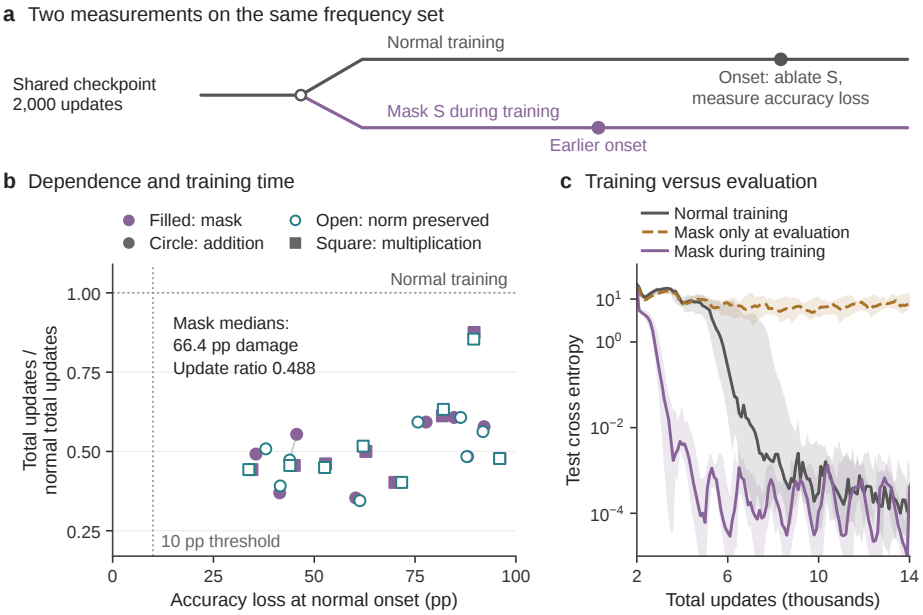

Figure 1: Masking frequencies that the trained model relies on brings generalization forward. S is a random half (28 of 56) of the operand embedding's Fourier frequencies. (a) One run, two measurements: the normal branch is ablated at its onset to read reliance; the other branch trains with S masked and generalizes earlier. (b) All points lie right of the 10-point threshold and below 1; 16 fresh runs, two joined points each, one per masking operator. (c) Median test cross-entropy with interquartile bands; weights masked only at evaluation never reach onset within 50,000 updates.

et al., 2023), so a set of frequencies is a region whose *content*, the embedding's component in those frequencies, we can delete, replace or hold fixed at any point in training. Reliance and delay can point in opposite directions: ablating a set of frequencies from a trained model destroys most of its accuracy, yet masking the same set during training roughly halves the time to generalize (Figure 1). Masking changes two things at once, what the set held at the fork and how later updates can treat it, so the acceleration alone cannot say which of the two helps. With the set and the model state held fixed, which content adds delay? We replace the set's content and compare when the branches generalize: the run's own early content adds far more delay than an energy-matched scrambled copy or its *later* content, held after generalization (Figure 3). And once the same content has been deleted, do the updates that rebuild it still delay generalization? We branch from the same state right after deletion and change only how the following updates treat that direction (Figure 2). The first identifies the effect of the content itself, the second the added effect of the rebuilding updates.

Why does this early content persist? In ordinary training AdamW adds the early content back about as fast as decay removes it. Removing the part of each AdamW update that adds it back advances generalization in most runs of three fresh cohorts, and deleting the content once and blocking its rebuild recovers most of the speed-up from masking. From the same state after deletion, blocking the rebuild brings generalization forward in all 16 one-layer runs, by a median of 1,200 updates over deletion alone and 850 over update-norm scaling (Figure 2). The account thus has three links, each tested directly: this early content delays generalization when put back, AdamW keeps rebuilding it against decay, and blocking the rebuild shortens the wait. All experiments use AdamW.

Contributions. In the AdamW-trained Transformers we study, the delay depends on which content a subspace holds, beyond its norm, and on the updates that rebuild it. *Content.* At a fixed subspace and state, the network's own early content delays generalization far more than a scrambled copy of equal energy, its later content, also at matched norm, or another network's early content (Section 3.3). *Maintenance.* AdamW rebuilds this early content as fast as decay removes it; after the same deletion, blocking the rebuild along its fixed direction beats deletion alone and update-norm scaling on fresh seeds at one and two layers, and deletion with blocking recovers most of the gain from masking

thousands of dimensions (Section 3.4). *Method.* Replacing content at a fixed state, and branching from one state right after deletion, separate what a subspace holds from how training treats it.

2 Interventions and generalization time

Our interventions follow one pattern. We train a run until it fits its training data, fork it before it generalizes (by default at update 2,000), and let branches that differ in one thing continue with the fork's optimizer state and randomness; then we compare when each branch generalizes (terms and symbols in Table 1). Branches of one run are therefore not independent: a count such as 13/16 counts runs, eight per task, and a prespecified rule passes when at least 7 of 8 runs per task meet it.

Models and runs. We follow the architecture and optimizer of Nanda et al. (2023). The model is a one-layer decoder-only Transformer without LayerNorm (two layers, with and without LayerNorm, in Appendices B.3 and B.5), with residual width 128, four attention heads and MLP width 512, reading its answer at the final token of `[a,b,=]` for arithmetic modulo 113. It is trained on 30% of operand pairs and tested on the other 70%, with full-batch AdamW (Loshchilov & Hutter, 2019) at learning rate 10^{-3}, weight decay 1 and betas $(0.9, 0.98)$, for 50,000 updates. The 16 runs where the pattern was found are the *discovery cohort*; each confirmation below uses 16 new seeds of its own, trained once against a rule fixed in advance (Table 46; one two-layer seed had a prior smoke run, Appendix B.3).

Masking a set of frequencies. Every coordinate of the operand embedding, read across the tokens in group order, is a function on the group and decomposes into the group's Fourier frequencies, in which the known solution is written (Nanda et al., 2023); the parts we mask are sets S of them. Let $E \in \mathbb{R}^{128\times n}$ be the operand embedding in group order ($n = 113$ operand tokens for addition, 112 for multiplication), $\mu = \frac{1}{n}E\mathbf{1}$ its column mean, and Q_S the projector onto the sine and cosine directions of S. Masking S replaces E on every forward pass by

$$E^{(-S)} = \mu\mathbf{1}^\top + (E - \mu\mathbf{1}^\top)(I - Q_S), \tag{1}$$

which zeroes the components at the frequencies in S and leaves the mean and everything else unchanged. Gradients pass through the projection, so the rest of the model adapts around the gap, but nothing re-enters S while the mask is on. Ablating S applies the same projection once to a finished model; the damage is the resulting loss of test accuracy. A set of 28 frequencies spans 55 or 56 real directions; for multiplication the operands are ordered by powers of a primitive root.

Table 1: **Terms and symbols.**

Term	Meaning
run / fork / branch	one seed and data split / its checkpoint at update 2,000 / a continuation from the fork with the fork's optimizer state and randomness
S	a set of Fourier frequencies, here a random half fixed before initialization, without the finished model
mask / ablate	project S out on every forward pass during training (Eq. 1) / apply the same projection once to a finished model
content, early / later; energy	the embedding's component in S, $(E - \mu\mathbf{1}^\top)Q_S$; what is there before / after generalization; its squared Frobenius norm
own / scrambled / nothing	put back the normal branch's own content (*native* in figures and appendices) / a phase-scrambled copy of equal energy (*phase*) / nothing (*sham*)
onset T, ratio	first of five consecutive evaluations, 100 updates apart, with test accuracy ≥ 0.99; $T_{\text{intervention}}/T_{\text{normal}}$
structured output, onset	the part of the logits that depends only on the operands' sum in group coordinates (cross-fitted) / first of three consecutive readouts with its cross-entropy at most 0.1
fresh run; cohort	a run whose seed was never trained before, with the protocol frozen before training; a group of fresh runs trained for one test (16 by default)
masking gain	$T_{\text{normal}} - T_{\text{mask}}$; a branch recovers the fraction $(T_{\text{normal}} - T_{\text{branch}})/(T_{\text{normal}} - T_{\text{mask}})$

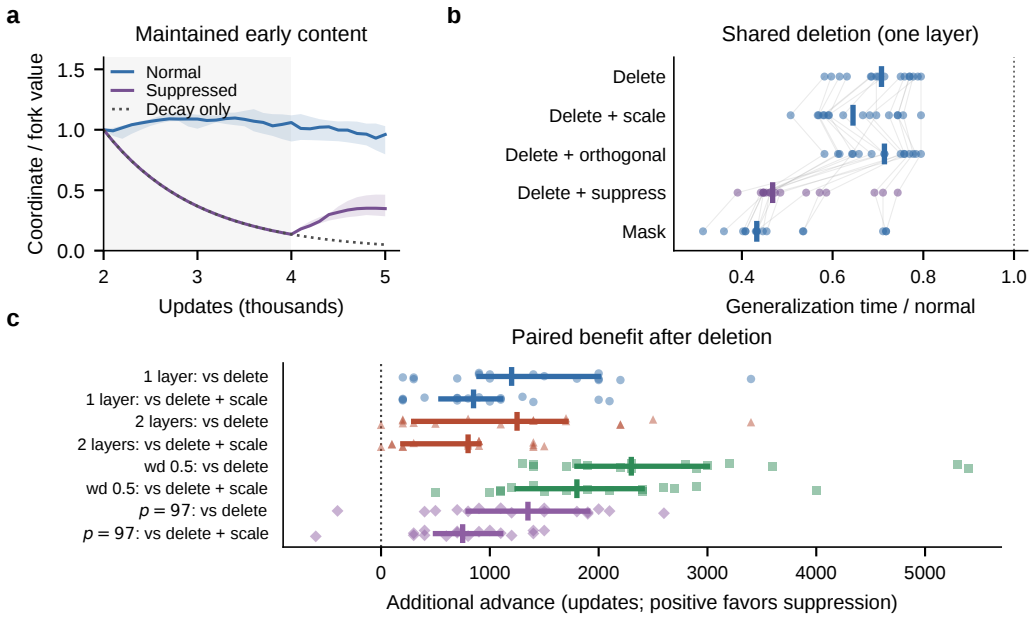

Figure 2: Suppressing the rebuild advances generalization beyond deletion alone and update-norm scaling. (a) AdamW maintains the early content: its coordinate along its unit direction d, relative to its value at 2,000, under ordinary training, suppression (positive updates along d removed during 2,001–4,000, shading) and weight decay alone (dotted); medians and interquartile ranges, 16 one-layer runs. (b) Total updates to generalization divided by each run's normal branch, one-layer cohort; points are runs, bars medians. The four deletion rows start from the same deleted state and then use ordinary AdamW updates, updates shrunk to the norm suppression leaves (scale), the same removal along a direction orthogonal to d, or suppression; mask is S masked throughout. (c) Within-run advance of deletion with suppression over each control: fresh-seed runs at one and two layers, and extensions on 16 existing runs at weight decay 0.5 and p=97; thick marks: paired medians, 95% bootstrap intervals.

Masking, unmasking and restoring. The mask stays on for the rest of training, or only for a window after which the set is unmasked. One update after unmasking, we put back the normal branch's own content in S at that moment, the increment $(E_N - E_W)Q_S$ with E_N and E_W the normal and window branches' embeddings (*native*); a scrambled copy with the same energy at every row and frequency, each such pair rotated by an independent random angle (*phase*); or nothing (*sham*). Energies and norms refer to this increment, which sets the restored content's norm to within 0.1% (Appendix B.26).

Operators. Masking acts on the forward pass (Eq. 1); at unmasking the effective embedding $E^{(-S)}$ is written into the parameter, so S starts empty. Norm-preserving masking rescales the centered masked embedding to its pre-mask Frobenius norm on every pass; embedding-norm scaling shrinks the whole centered embedding to the norm masking would leave, keeping every frequency.

Measuring generalization time. We report a branch's onset (Table 1) as its ratio of total updates, counting those before the fork, to normal training's. Branches follow their cohort's budget and stopping rule with no exclusions; one that never reaches onset is reported as right-censored.

3 Early content, its maintenance, and the grokking delay

3.1 A set the model relies on brings generalization forward when masked

How much does a trained model rely on a set of frequencies, and what does removing that set during training do to learning? We fix S, a random half of the frequencies (28 of 56), before each of 16 fresh

runs is initialized, and fork every run into normal training, masking, norm-preserving masking and embedding-norm scaling.

The generalizing model relies on S: ablating it from the normal branch when it first generalizes costs a median of 66.4 accuracy points, more than ten in every run (Appendix B.1). During learning, removing the same set helps: masking it brings generalization forward in all 16 runs, to a median ratio of 0.488, while embedding-norm scaling gives 1.011 and the same weights masked only at evaluation never reach onset (Figure 1c). The prespecified rule, ablation costs more than ten points and both masking operators generalize earlier, holds in 8/8 runs in each task (Figure 1b).

The rule holds at modulus 97 and at two layers (16 fresh runs each; Appendices B.2, B.3), and in the one-layer setting uniformly drawn halves speed learning: all 640 drawn on 32 fresh runs generalize earlier (median ratio 0.43; 305 of 320 on a designed grid; Appendices B.53, B.54). Masking from update 1,000 or 3,000 is also earlier in 16/16 runs, and masking as late as 5,000 still saves about three quarters of the remaining wait (Appendices B.7, B.40). Masking 24 frequencies outside the four the finished model relies on most gives the full speed-up (ratio 0.471, 16/16), and that branch's finished model still relies on those four, losing a median 0.966 of its accuracy without them (Appendix B.15).

With LayerNorm the rule fails (3/8 in each task): normal training itself generalizes only 1,550 updates after the fork; forked earlier, at 1,000, masking still brings generalization forward in 14/16 runs (ratio 0.684) while the rule stays unmet (6/8 and 5/8; Appendices B.5, B.6).

3.2 A SINGLE DELETION HELPS, AND A BRIEF MASK KEEPS MOST OF THE GAIN

Masking deletes the set's content at the fork and keeps the set empty afterwards. Does acceleration require keeping S empty throughout, or does a single deletion or a short masking window suffice? We compare a single deletion, a freeze, a 500-update window and sustained masking on the discovery cohort's 16 runs (Appendix Figure 6); the window length was chosen using earlier outcomes.

Freezing S at its early content delays generalization (median ratio 1.479 against normal training; no run earlier), whereas deleting the content once advances it (0.737), and a 500-update mask recovers most of the gain from sustained masking (0.481 against 0.486; paired median difference zero; Appendix Table 22). On 16 fresh runs the window is again earlier than normal training in every run and within 200 updates of sustained masking in 13/16, one short of its prespecified rule, and a random (Haar) subspace masked for 500 updates gives 0.932 (Appendices B.18, B.48, B.19, B.17).

Table 2: **Main comparisons** (16 runs, 8 per task). Effects are paired medians of full onset in updates (suppression alone: against normal training), median ratios of total updates to onset, or shares of the masking gain; slashes separate the settings named in each row (1L, 2L: one, two layers); commas separate addition and multiplication. F: fresh seeds, rule fixed in advance; E: extension on existing runs; scale: update-norm scaling. The 2L after-deletion rows use the fresh-seed two-layer confirmation; the 2L share of the masking gain uses the original two-layer cohort. †: the run count in this row is one of the nine Holm-corrected conclusions (Appendix Table 4); rows without † are outside that family; effects are descriptive.

Comparison	Effect	Runs	Appendix	Kind
Mask a random half vs. normal (1L)†	ratio 0.488	16/16 earlier	B.1	F
at p=97 / two layers†	0.518 / 0.462	16/16 / 16/16	B.2, B.3	F
Restore own early content vs. scrambled†	2,600 later	16/16	B.20	F
Restore early vs. later content, same norm†	1,900 later	16/16	B.26	E
Restore 24 of S (top 4 left out) vs. nothing†	2,350 later	16/16	B.28	E
Suppression alone vs. normal and scaling†	1,150 earlier	14/16 (rule)	B.32	F
After deletion: suppress vs. delete (1L/2L)	1,200 / 1,250 earlier	16/16 / 15/16	B.32, B.33	F
at weight decay 0.5 / p=97	2,300 / 1,350 earlier	16/16 / 15/16	B.33	E
After deletion: suppress vs. scale (1L/2L)	850 / 800 earlier	16/16 / 15/16	B.32, B.33	F
at weight decay 0.5 / p=97	1,800 / 750 earlier	16/16 / 15/16	B.33	E
Deletion + suppression vs. normal (1L/2L)†	share 0.93 / 0.91	16/16 / 16/16	B.32 / B.33	F
at weight decay 0.5 / p=97	share 0.92, 0.88 / 0.88, 0.93	16/16 / 16/16	B.34 / B.35	F

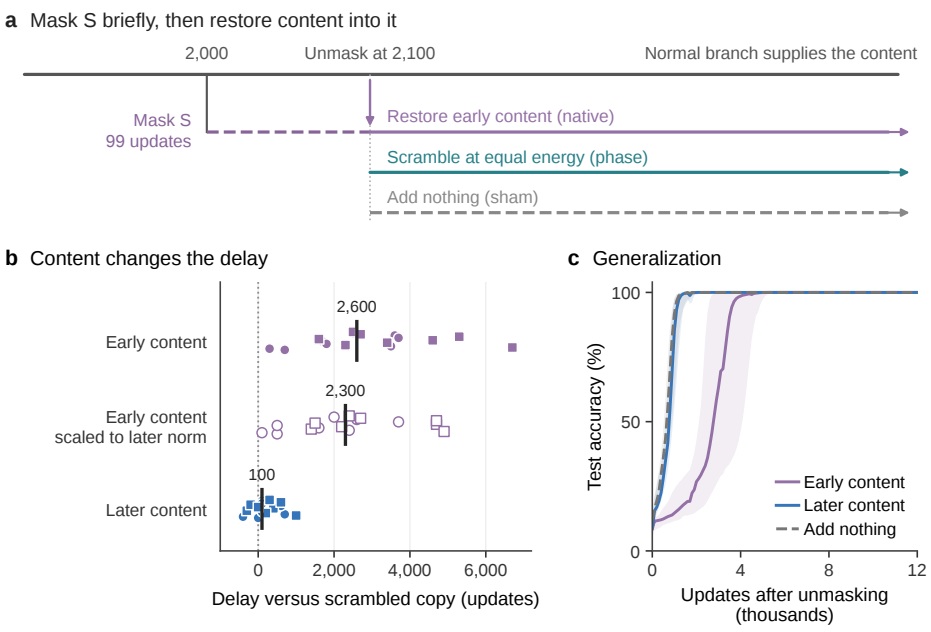

Figure 3: Restored early content of S delays generalization; later content barely does. *Early* content is what normal training holds in S at update 2,100, before generalizing; *later* content what it holds at confirmed generalization. (a) Design: S is masked from 2,000 to 2,100, one content is then placed into it, and training continues. Native, phase and sham are the run's own early content, supplied by the normal branch, a scrambled copy that keeps its energy in every embedding row and frequency, and nothing. (b) Updates by which onset comes later with each content than with its own scrambled copy. The middle row (open markers) shrinks early content to the later content's norm in each run; filled markers restore content at its own norm. Ticks and numbers are medians; circles are addition, squares multiplication. (c) Median test accuracy after unmasking, with interquartile bands. Early-content restoration was tested on 16 fresh runs under a prespecified protocol; later-content and norm-matched replacements were subsequent extensions on the same runs.

3.3 In the same set, early content delays generalization and later content barely does

We next test whether putting the content back reinstates part of the delay, and whether that depends on which content is put back. *Content replacement* fixes a subspace S and a model state (here the state whose S was masked briefly and then unmasked), places one content into S, trains on, and reads the onset against the scrambled control; only the content varies. A scrambled copy of equal energy asks whether energy alone matters. The run's own later content asks whether being its own content is enough, and another run's early content whether any memorizing fit will do. The test uses 16 further fresh runs; we first fix the receiving state at 2,100 and vary the content (Figure 3).

Early content brings back delay. Put the normal branch's own early content back at 2,100 and generalization comes a median of 2,600 updates later than with scrambled content of equal energy, in all 16 runs; the scrambled content itself behaves like restoring nothing (Appendix B.20). The delay therefore depends on the content's pattern, beyond its energy.

Later content and another run's early content cause little delay. Is it enough that the content is the network's own? Its content from confirmed generalization or from its 50,000-update endpoint, restored into the same state, delays generalization by medians of 100 and 0 updates against its own scrambled control (Appendix B.24), yet the later content is kept (0.94 of its energy after 1,000 updates) and used: content that stays in place can still cause little delay. Is it the norm? Scaled to the later content's norm, the early content still delays more in all 16 runs, and the later content scaled up to the early norm delays by 0 (Appendices B.26 and B.27). Will any memorizing fit do? Another

run's early content, projected onto this run's S at about the same norm, delays generalization by a median of 0 updates against its scrambled control, within 200 in 14/16 runs (Appendix B.27).

Where within S does the delay lie? The delay and the finished model's reliance sit in different frequencies: restoring only the content of the four most relied-on frequencies gives almost none of the delay, and restoring the other 24 gives 0.944 of it (Appendices B.28–B.30). Run by run, delay and extra ablation damage are only loosely coupled: the prespecified secondary rule for addition, delay together with more than 20 extra points of damage, holds in only 4/8 runs (Appendix Figure 7c).

3.4 Rebuilding the early content is part of the delay

Section 3.3 showed that the early content delays generalization when it is put back. Does the optimizer keep it in place in ordinary training, and does keeping it there delay generalization? We block the rebuild by removing, from each AdamW update, only its positive component along the early content's fixed direction, an operation we call suppression. On 16 fresh runs trained normally to update 2,000, let $D = (E_{2000} - \mu \mathbf{1}^\top) Q_S$ be the early content of a random half, $d = D/\|D\|$ (a single direction in the embedding's parameter space, spanning every row and all frequencies of S), and U the change an AdamW step makes to the embedding, excluding weight decay; the moments are updated as usual. For updates 2,001–4,000, suppression replaces U by $U - \max(\langle U, d\rangle, 0)\, d$. The update-norm scaling control shrinks U to the norm suppression would leave; the orthogonal control removes the same amount along a fixed unit direction in S orthogonal to d; deletion sets D to zero once at the fork. The *rebuild ratio* is $\sum_t \langle U_t, d\rangle$ divided by what weight decay removes along d over the same updates. *Restoration* is our putting content back (Section 3.3); the *rebuild* is what AdamW's updates put back; *maintenance* is the persistence that results.

Ordinary training maintains the content. Over updates 2,001–4,000 AdamW adds back a median 1.03 times what decay removes along d (0.79–1.08), so the coordinate does not shrink (Figure 2a); decay alone would leave 0.135 of it. The content carries the training fit: deleting it at 2,000 raises the training loss by 4.4 to 14 nats. At a second fresh cohort's forks a plain gradient step would return 1.5×10^{-8} to 2.2×10^{-5} of what decay removes (15/16 states; Appendix B.39).

Blocking the rebuild adds a benefit beyond deletion. Does blocking the rebuild help beyond deleting the content once? A single deletion already brings generalization forward, by a paired median of 2,050 of the 4,700 updates normal training still needs, so deletion with suppression beating normal training says little about the rebuild. We therefore compare branches that start from identical states right after deletion and differ only in the updates that follow. In the one-layer cohort, suppression advances full onset by a paired median of 1,200 updates relative to deletion alone and 850 relative to deletion with update-norm scaling, in 16/16 runs each; the prespecified rule, both onsets (full generalization and the structured-output onset of Table 1) earlier than both controls, holds in 7/8 and 8/8 runs (Figure 2b,c; Appendix B.32, Table 32). These comparisons measure the effect of changing the update rule from a shared state; because the branches' contents diverge afterwards, the delay due to maintenance at matched content trajectories remains unmeasured.

At two layers, an extension on existing runs gave smaller gains (850 and 550 updates; the rule met in 6/8 per task). On 16 fresh two-layer runs the rule then holds (7/8 and 8/8), with paired medians of 1,250 updates over deletion alone and 800 over scaling, earlier in 15/16 runs each (Figure 2c). Extensions at weight decay 0.5 and modulus 97 agree, with all four intervals excluding zero (Table 2); re-reading onset at 0.98 or 0.99 accuracy over three or five evaluations leaves every earlier-run count unchanged in the one-layer cohort and the three extensions (post hoc; Appendix B.33).

Relation to masking. Suppression without deletion helps less (Table 2; on full onset alone, 15/16 runs earlier). As shares of the masking gain, suppression alone recovers 0.31, deletion alone 0.58 and both 0.93. Masking removes and keeps out 128×55 or 128×56 dimensions, while these interventions touch one direction d. The same removal along an orthogonal direction has no median effect alone (ratio 1.000) and, after deletion, is 1,250 updates later than suppression (Table 32).

The added benefit depends on which content is rebuilt. On the runs of Section 3.3, suppressing the rebuild of restored early content advances both onsets in every run (medians 1,450 and 1,300 updates), the same operation on restored later content gives no gain (1 earlier, 5 tied, 6 later of 12), and

the early gain exceeds the later one in 12/12 runs, as prespecified (Appendix B.31); a norm-matched radial-update control, which removes each update's positive component along the whole embedding's radial direction and rescales the rest to our suppression's norm, shows most of the same dependence and depletes the content along d more than ours (Appendix B.31; Section 4).

Confirmation and extensions. Suppression alone is the least robust arm. On a second fresh cohort, continuous suppression misses its rule by one run (13/16; the 2,000-update window, 12/16); on a third it advances onset in every run (Appendices B.37, B.38). At depth two AdamW's updates restore about nine tenths of what weight decay removes (Appendix B.33).

How a small gradient can offset decay. Weight decay shrinks every coordinate by a fixed fraction per update; a coordinate keeps its size only if the updates add back as much. Writing c_t for the coordinate along d and u_t for AdamW's update along it, decoupled decay gives $c_{t+1} = (1-\eta\lambda)c_t + u_t$, so the rebuild ratio equals $1 + (c_T - c_0)/(\eta\lambda T\bar{c})$, $\bar{c}$ the mean coordinate: a ratio near one means the coordinate did not shrink. Because AdamW divides each gradient by its running magnitude, a steady gradient far too small for an SGD step to offset decay still yields updates that do, down to gradients of order ϵ (a constant-gradient illustration in Appendix B.39, Eq. 3; the balance itself is known, Wang & Aitchison, 2025; cf. Kosson et al., 2024). What we measure is the coordinate along d and how much blocking its maintenance shortens the wait. Lowering ϵ from 10^{-8} to 10^{-12} at the fork advances onset modestly but misses its prespecified rule (5/8 and 6/8), and in a post hoc audit raises the coordinate along d over the window in only 5/16 runs, so this global change does not isolate the rebuild (Appendix B.42).

Holding the content in place. Must the content decline before generalization can occur? No. A floor that keeps one coordinate, the one along d, at or above its value at the fork through 12,000 (the freeze of Section 3.2 fixed all of S) is active in 16/16 runs, and 13/16 generalize before its release; the prespecified sufficiency test, a delay of more than 200 updates, failed (5/8 and 6/8; paired median delays of 350 updates in addition and 500 in multiplication; Appendix B.38). The floor and suppression impose different coordinate trajectories (Appendix B.32), so the two experiments measure different quantities.

Being maintained is not enough to delay generalization: the later content is maintained and used as well (Section 3.3; Appendix Figure 7c,d).

4 Related Work

Reliance and training. Ablation and patching measure how much a fixed model relies on its parts (Wang et al., 2023; Conmy et al., 2023); the measure depends on the operator (Li & Janson, 2024) and can be an artifact of it (Makelov et al., 2024), and other components can compensate without any learning (McGrath et al., 2023; Gong et al., 2026). Decodability need not imply use (Huang & Chang, 2025; Sharma et al., 2026), stitching tests whether one network's layers can feed another's (Bansal et al., 2021; Csiszárik et al., 2021; Almudévar & Ortega, 2026), and separately trained models can share an algorithm (Chughtai et al., 2023; Schiffman, 2026; Sun & Toneva, 2026); all concern fixed models. Retraining after input removal (Hooker et al., 2019) and lottery-ticket retraining (Frankle & Carbin, 2019) restart from initialization, Singh et al. (2024) clamp head activations from the start of training, and Liu (2026) compares checkpoint reliance with matched retraining.

Grokking and cleanup. Accounts of grokking emphasize competing circuits and learning speeds (Merrill et al., 2023; Varma et al., 2023; Song & Ye, 2026), representation learning and weight norm or rank (Liu et al., 2022; Kumar et al., 2024; Liu et al., 2023; Yunis et al., 2024; Truong et al., 2026b;a), optimizer dynamics and numerical effects (Thilak et al., 2022; Prieto et al., 2025), the frequencies, symmetries and algorithms the solution uses (Gromov, 2023; He et al., 2026; Morwani et al., 2024; Zhong et al., 2023; Hwang & Park, 2026), and structure that forms before the accuracy jump (Hidajat et al., 2026; Sivasankar, 2026; Howe, 2026; Gomezjurado, 2026; Swaroop, 2026). We measure which embedding content the optimizer maintains and how much blocking its maintenance shortens the wait. Nanda et al. (2023) show that a few key frequencies carry the solution and that ablating the rest slightly improves a trained model; Yang (2026) describes the transition as spectral recoding of a distributed circuit with no module switch.

Pruning and acceleration. Pruning to a good subnetwork (Minegishi et al., 2025) and optimizer or geometric methods (Lee et al., 2024; Walker et al., 2025) accelerate grokking; our deletions and blocks serve to separate content from its rebuild, and removing maintained content may explain part of those gains. Pruning memorizing neurons after training removes memorized corrupted labels (Doshi et al., 2024), and a late switch of implicit bias can itself produce the delay (Lyu et al., 2024). Early training is a sensitive period (Achille et al., 2019; Frankle et al., 2020b;a); shrinking and perturbing a warm-started model (Ash & Adams, 2020), re-initializing part of a trained one (Zhou et al., 2022), transferring embeddings (Xu et al., 2025) and briefly suppressing a structure the final model needs (Chen et al., 2024) can all help.

Relation to optimizer explanations. We share a premise with ⊥Grad (Prieto et al., 2025): after the fit the gradient is small and close to radial, and d overlaps the embedding's radial direction ($\cos(d, E/\|E\|)$ is 0.62–0.70 at the saved states; Appendix B.44); we add that what AdamW keeps in place is a particular fit. Our norm-matched radial-update control also depletes the content along d and shows most of the same early-versus-later dependence (the prespecified rule that ours separates the two better than this control failed; Appendix B.31). What our suppression adds is locality: it acts along one fixed direction, while the radial control rescales the rest of the update by a mean factor of 14 to 24. We identify the content these interventions act on, consistent with radial-update accounts.

5 Discussion

What these results establish. At a fixed state and subspace, early and later content of matched norm delay generalization by different amounts, so what the subspace holds matters beyond its norm. From the same state after deletion, blocking the rebuild along one fixed direction is earlier than both deletion alone and update-norm scaling, so the updates after a deletion change the outcome. A brief mask, or deletion followed by blocking along one direction, recovers most of the gain of sustained masking while the rest of the subspace stays trainable. A training-time ablation that speeds learning has removed content and restricted what later updates can rebuild; here each adds to the gain.

What does the content slow? With cross-entropy the gradient is $J^\top r$, r the error signal on the training pairs and J the network's response. With the early content present the fit is nearly exact and r almost vanishes ($|r| \approx 8 \times 10^{-9}$, against 2×10^{-2} with the content deleted, a contrast in orders of magnitude that persists under float64 recomputation; Appendices B.44, B.39). The maintained fit appears to leave the generalizing solution almost no error signal (cf. Pezeshki et al., 2021). A direct test of this path failed both prespecified rules: when training takes its loss from a forward pass with the content removed, that forward pass generalizes nearly as early as under masking, while the actual model, which still holds the content, generalizes only after the content is released (Appendix B.44).

Limitations and scope. We work in modular arithmetic because these questions need a known basis in which to replace content. Nine paired comparisons survive a Holm correction (Table 2); the other prespecified tests are uncorrected. At depth two the restoration test did not pass (6/8 in each task; Appendix B.4), and with LayerNorm the masking confirmation misses its rule (Section 3.1). Maintenance is measured under AdamW, with a saved-state calculation for plain gradient descent (Appendix B.39); which component of AdamW does the maintaining is open. All runs are full-batch, with one learning rate (10^{-3}) and one training fraction (30% of pairs). The benefit of blocking beyond deletion is a fresh-seed result at one and two layers; at weight decay 0.5 and modulus 97 it comes from extensions on existing runs. The gradient decomposition is one-layer at one weight decay; suppression alone misses its rule in one task at depth two, at weight decay 0.5 and at modulus 97 (Appendices B.33, B.34, B.35). The gradient analysis and the onset-reading check are post hoc; every failed prespecified test is listed in Appendix A.

Conclusion. In our settings the delay has a content-level cause that interventions can add and remove: putting the early content back delays generalization, AdamW keeps rebuilding it against decay, and blocking the rebuild after the same deletion brings generalization forward beyond deletion alone and update-norm scaling. This locates part of the grokking delay in maintained embedding content, complementing optimizer-level accounts; which downstream computation the content slows is the next question.

AI USE STATEMENT

The authors fixed the confirmatory protocol, ran the experiments, audited the code and outputs, and manually verified the reported statistics, citations, and claims. Generative AI tools were used to refine hypotheses, provide feedback on experimental design, assist with code implementation and debugging, support result interpretation and literature search, assist with translation, create or edit scientific figures and research artifacts, and propose, structure, draft, or edit manuscript text, including the title and keywords. References suggested during AI-assisted literature search were checked against their sources before inclusion. The authors reviewed every AI-assisted hypothesis and passage against the cited sources and the shipped records, and take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

The appendices specify cohort rules, source selection, intervention operators, timing and censoring, and Appendix A maps claims to protocols and scope limits. The anonymized artifact, uploaded as supplementary material, ships the frozen protocol, results files and independent verifier output of every cohort, including every cohort whose prespecified gate failed, and for most cohorts the per-branch trajectories, intervention records and composed states; one script recomputes every headline number in the text from the verifier outputs alone, the figure builders read only these files, and the artifact's README states which cohorts' full trajectories and checkpoints are retained with the experiments instead. Three levels of reproduction are available: recomputing statistics from recorded evaluations, which every cohort supports; replaying updates from shipped composed states, which the restoration decomposition supports; and re-running an intervention from a source checkpoint, which needs the retained checkpoints (about six gigabytes, kept with the experiments and released with the code). "Prespecified" means that a protocol was frozen and hashed locally before its runs.

REFERENCES

Pratyush Acharya and Habish Dhakal. Grokking as a variance-limited phase transition: Spectral gating and the epsilon-stability threshold, 2026. URL https://arxiv.org/abs/2603.15492. arXiv preprint.

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *International Conference on Learning Representations*, 2019.

Antonio Almudévar and Alfonso Ortega. Bridging functional and representational similarity via usable information. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2601.21568.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/288cd2567953f06e460a33951f55daaf-Abstract.html.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 225–236, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/01ded4259d101feb739b06c399e9cd9c-Abstract.html.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.07311.

Bilal Chughtai, Lawrence Chan, and Neel Nanda. A toy model of universality: Reverse engineering how networks learn group operations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6243–6267. PMLR, 2023. URL https://proceedings.mlr.press/v202/chughtai23a.html.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5656–5668, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2cb274e6ce940f47beb8011d8ecb1462-Abstract.html.

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*, 2024.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3259–3269. PMLR, 2020a. URL https://proceedings.mlr.press/v119/frankle20a.html.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020b.

Laura Gomezjurado. The long delay to arithmetic generalization: When learned representations outrun behavior, 2026. URL https://arxiv.org/abs/2604.13082. arXiv preprint.

Zhiren Gong, He Lu, Tiantong Wang, Yichi Zhang, Yixin Wang, Zihao Zeng, Ming Xiao, Chau Yuen, and Wei Yang Bryan Lim. Conditional co-ablation: Recovering self-repair backups in transformer circuits, 2026. URL https://arxiv.org/abs/2607.01940. arXiv preprint.

Andrey Gromov. Grokking modular arithmetic, 2023. URL https://arxiv.org/abs/2301.02679. arXiv preprint.

Jianliang He, Leda Wang, Siyu Chen, and Zhuoran Yang. On the mechanism and dynamics of modular addition: Fourier features, lottery ticket, and grokking, 2026. URL https://arxiv.org/abs/2602.16849. arXiv preprint.

Kai Hidajat, Solden Stoll, and Joseph An. Grokking as structural inference: Transformers need Bayesian lottery tickets, 2026. URL https://arxiv.org/abs/2605.15787. arXiv preprint.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fe4b8556000d0f0cae99daa5c5c5a410-Abstract.html.

Gunner Levi Howe. Structure-specific representational priors causally control the grokking delay, 2026. URL https://arxiv.org/abs/2607.04333. arXiv preprint.

Lianghuan Huang and Yingshan Chang. Causality $\neq$ decodability, and vice versa: Lessons from interpreting counting ViTs, 2025. URL https://arxiv.org/abs/2510.09794. arXiv preprint.

Hyeonbin Hwang and Yeachan Park. Intrinsic task symmetry drives generalization in algorithmic tasks, 2026. URL https://arxiv.org/abs/2603.01968. arXiv preprint.

Zeyu Jia. Beyond embedding transfer: Component roles in grokking transfer and stability, 2026. URL https://arxiv.org/abs/2609.18078. arXiv preprint.

Chon-Fai Kam, Miloud Bessafi, and Frederic Cadet. Structured features overfit where random features grok, 2026. URL https://arxiv.org/abs/2609.15047. arXiv preprint.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25333–25369. PMLR, 2024. URL https://proceedings.mlr.press/v235/kosson24a.html.

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*, 2024. URL https://openreview.net/forum?id=vt5mnLVIVo.

Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients, 2024. URL https://arxiv.org/abs/2405.20233. arXiv preprint.

Maximilian Li and Lucas Janson. Optimal ablation for interpretability. In *Advances in Neural Information Processing Systems*, volume 37, pp. 109233–109282, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c55e6792923cc16fd6ed5c3f672420a5-Abstract-Conference.html.

Yi Liu. Learned, relied upon, or necessary? Separating checkpoint dependence from task-level value in sheaf GNNs, 2026. URL https://arxiv.org/abs/2607.25387. arXiv preprint.

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

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zDiHoIWa0q1.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? An interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70b8505ac79e3e131756f793cd80eb8d-Abstract-Conference.html.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. The Hydra effect: Emergent self-repair in language model computations, 2023. URL https://arxiv.org/abs/2307.15771. arXiv preprint.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models (ME-FoMo)*, 2023. URL https://arxiv.org/abs/2303.11873.

Gouki Minegishi, Yusuke Iwasawa, and Yutaka Matsuo. Bridging lottery ticket and grokking: Understanding grokking from inner structure of networks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=eQeYyup1tm.

Depen Morwani, Benjamin L. Edelman, Costin-Andrei Oncescu, Rosie Zhao, and Sham M. Kakade. Feature emergence via margin maximization: case studies in algebraic tasks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=i9wDX850jR.

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. URL https://openreview.net/forum?id=9XFSbDPmdW.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1256–1272, 2021. URL https://arxiv.org/abs/2011.09468.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177. arXiv preprint.

Lucas Prieto, Melih Barsbey, Pedro A. M. Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TvfkSyHZRA.

Joshua S. Schiffman. Transformers converge to invariant algorithmic cores. In *ICML 2026 Workshop on Mechanistic Interpretability*, 2026. URL https://arxiv.org/abs/2602.22600.

Aryan Sharma, Cutter Dawes, and Shivam Raval. Dissociating decodability and causal use in bracket-sequence transformers. In *ICML 2026 Workshop on Compositional Learning*, 2026. URL https://arxiv.org/abs/2604.22128.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45637–45662. PMLR, 2024. URL https://proceedings.mlr.press/v235/singh24c.html.

Achyuthan Sivasankar. Circuit synchronization precedes generalization: A causal precursor to grokking, 2026. URL https://arxiv.org/abs/2606.12966. arXiv preprint.

Yiding Song and Hanming Ye. Model capacity determines grokking through competing memorisation and generalisation speeds, 2026. URL https://arxiv.org/abs/2605.09724. arXiv preprint.

Alan Sun and Mariya Toneva. Tracking equivalent mechanistic interpretations across neural networks. In *International Conference on Learning Representations*, pp. 38617–38648, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/412e7a0143cb33d3a7a9b6abe8d90470-Paper-Conference.pdf.

Anand Swaroop. Latent algorithmic structure precedes grokking: A mechanistic study of ReLU MLPs on modular arithmetic, 2026. URL https://arxiv.org/abs/2603.23784. arXiv preprint.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua M. Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon, 2022. URL https://arxiv.org/abs/2206.04817. arXiv preprint.

Xuan Khanh Truong, Quynh Hoa Truong, Duc Trung Luu, and Thanh Duc Phan. First-passage prediction of grokking delay: A calibrated law under AdamW with causal validation, 2026a. URL https://arxiv.org/abs/2605.18845. arXiv preprint.

Xuan Khanh Truong, Quynh Hoa Truong, Duc Trung Luu, and Thanh Duc Phan. The norm-separation delay law of grokking: A first-principles theory of delayed generalization, 2026b. URL https://arxiv.org/abs/2603.13331. arXiv preprint.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency, 2023. URL https://arxiv.org/abs/2309.02390. arXiv preprint.

Thomas Walker, Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. GrokAlign: Geometric characterisation and acceleration of grokking, 2025. URL https://arxiv.org/abs/2506.12284. arXiv preprint.

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*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Xi Wang and Laurence Aitchison. How to set AdamW's weight decay as you scale model and dataset size. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62222–62250. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25b.html.

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. URL https://arxiv.org/abs/2504.13292.

Dekun Yang. Where grokking happens: Distributed utility and Fourier recoding without a module switch, 2026. URL https://arxiv.org/abs/2609.17571. arXiv preprint.

David Yunis, Kumar Kshitij Patel, Samuel Wheeler, Pedro Henrique Pamplona Savarese, Gal Vardi, Karen Livescu, Michael Maire, and Matthew Walter. Grokking, rank minimization and generalization in deep learning. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=6NHnsjsYXH.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 27223–27250, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/56cbfbf49937a0873d451343ddc8c57d-Abstract-Conference.html.

Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron Courville. Fortuitous forgetting in connectionist networks. In *International Conference on Learning Representations*, 2022.

A Evidence and Scope Index

Where to look. The nine conclusions that survive the Holm correction (Table 4) can be verified from eight appendices alone: Appendices B.1–B.3 (masking a relied-on set, at two moduli and two depths), B.20 and B.26 (content replacement), B.28 (the frequencies that carry the delay), and B.32 and B.33 (suppressing the rebuild). The remaining appendices hold the scope tests, controls, extensions, failed gates and development records; the main text cites each where it uses it, and Table 3 lists every one with its evidence kind.

Terms. Masking a set applies Eq. (1) on every forward pass during training, and unmasking stops it; ablating applies the same projection once to a trained model. Ordinary and norm-preserving removal are the two versions of that projection, so removal of a set during training means masking it. A gate is a prespecified pass criterion. Older appendices also say *source* for the fork, *character* for a Fourier frequency plane, *panel* or *batch* for a cohort, *replenishment* for the rebuild, *attainment* or *arrival* for onset, and *native*, *phase* and *sham* for the original, phase-scrambled and no-restoration branches. Appendix headings use the main-text names; the older words survive only in prose, table labels and figure titles. Table 1 in the main text defines the current terms.

How to read the appendix. Three kinds of experiment appear. *Fresh-run confirmations* (F), with seeds, sets and protocols fixed before any of their models was initialized. *Extensions* (E), which reuse runs whose earlier outcomes were known; where readings or rules were fixed before launch, the appendix section states them. *Post-hoc analyses* (P) of shipped trajectories and checkpoints, which set no gate. The earlier panels (Appendix B.8) and the cohort that failed its validity gate (Appendix B.56) precede the fresh-seed work; the two development studies (Appendices B.21 and B.45) fixed designs and are not evidence for the claims. Table 3 maps each finding to its evidence; each appendix states its own scope.

Reading paths. For the *core evidence*, start with restoration, content age and amplitude control (Appendices B.20, B.24, B.26), then ordinary training and its two-layer confirmation (Appendices B.32 and B.33). For the *limits of the explanation*, read the localization within S, the binding floor, and the two-layer and LayerNorm boundaries (Appendices B.28, B.38, B.4, B.5). For *exploratory explanations and their tests*, see gradient routing, removal timing, parameter trajectories and the failed replication (Appendices B.36, B.40, B.42, B.43). The F/E/P labels above describe evidence status independently of these reading paths.

Table 3: Evidence index.

Finding	Where	Kind	Key numbers
Masking a set the normal model relies on brings generalization forward	B.1	F	damage at onset 66.4 pp; earlier 16/16, ratio 0.488; rule met 8/8 per task
The same at modulus 97	B.2	F	rule met 7/8 and 8/8; earlier 16/16, ratio 0.518; damage 69.6 pp
The same in a two-layer model	B.3	F	rule met 8/8 and 8/8; earlier 16/16, ratio 0.462; damage 80.6 pp
The same with LayerNorm	B.5	F	rule not met, 3/8 and 3/8 (two unfinished); earlier 9/14, ratio 0.921; damage 90.6 pp; normal needs 1,550 after the fork, masked 1,150
The same with LayerNorm at an earlier fork	B.5	F	rule not met, 6/8 and 5/8; random removal earlier 14/16, median 900, ratio 0.684; lag 1,100 to 1,450; lag reading met in addition (7/7), not in multiplication (2/4)
The lag after masking across cohorts	B.6	P	masked branches need 1,050 to 1,350 after the fork in every cohort; earlier in all 58 with 3,000 or more left, 2 of 5 with under 1,500
The same from other fork times	B.7	E	earlier 16/16 at 1,000, 2,000, 3,000; ratios 0.262, 0.396, 0.517
The set need not be chosen well	B.9, B.10	F	low set 0.487, earlier 16/16; against random removal earlier in 5, tied 1, later 10; scaling and magnitude pruning do not bring it forward
The mask must act during training	B.9, B.11	F, P	evaluation-only masking later in all 14 that attain; historical replay 21/21
Every sampled half is earlier	B.53, B.54	E, F	all 640 uniform halves on 32 fresh runs, median 0.43; 305 of 320 on the designed grid
Endpoint damage and generalization time for two sets	B.12	P	88.6 against 0.034 pp at the endpoint; ratios 0.486 and 0.487

Finding	Where	Kind	Key numbers
Which 28 are removed changes generalization time	B.50	E	energy-rich minus energy-poor $+0.277$ $[+0.159, +0.439]$; least relied-upon half -0.066; most relied-upon half 0.526, earlier 16/16
Source features rank sets modestly	B.53, B.54	E, F	held-out Spearman $+0.732$; on new runs mean 0.351 $[0.267, 0.436]$
A set's effect depends on the background	B.15, B.54	E, F	K: 2,300 alone, 200 with N (interaction $-2{,}050$, 13/16); 1,500 and 300 on 64 fresh runs (48/64); the N branch keeps K (0.966)
Reversals exceed a random four; onsets are noisy	B.55	E	K reverses in 5 and 3 of 64, a random four in 0; a 10^{-4} perturbation moves normal onsets by more than 200 in 18 of 64, with B masked in 1
Deleting the content helps; a brief mask keeps most of the gain	B.16, B.17	E	one-shot 0.737; freeze 1.479; 500-update window 0.481 against sustained 0.486
The window on fresh runs	B.18	F	earlier 16/16; the 200-update margin met in 13/16, so the conjunction is not met
Orientation matters at fixed rank	B.13, B.14, B.19	E	sustained Haar 0/16; Haar window 0.932 against 0.481; rotation 11.3° 0.591, 22.5° none
Restoring the early content delays generalization	B.20	F	rule met 7/8 and 8/8; later than both controls 16/16; 2,600, 650, 350 at the three cuts
It fits the training set and is replenished	B.20, B.22	F, E	training CE 1.95 against 3.38 on arrival, 16/16; retained energy 1.00 against 0.14 under decay alone
Restoration in the two-layer model	B.4	F	rule not met, 6/8 and 6/8; native later than phase by a median 550; test CE higher on arrival 16/16, training CE lower 1/16
The restoration at weight decay 0.5	B.23	F	later than both controls 16/16, median 3,700; retained 0.92 against 0.37
Reset, dose, timing and phase	B.22	E	after an AdamW reset 16/16; the time of the state restored into sets the delay in 15/16; coherent rotation 0/16
The solution's later content causes little delay	B.24, B.25	E	100 and 0 updates (600 and 50 at weight decay 0.5); kept 0.94; relied on 0.79
In ordinary training the optimizer maintains the early content, and suppressing that brings generalization forward	B.32	F	maintenance 1.03; suppression earlier than normal and scaling 7/8 and 7/8 (1,150 and 900); after deletion 7/8 and 8/8; deletion with suppression 0.93 of the masking gain, within 200 of masking in 8/16; complementary half 8/8 and 3/8
Replenishing early content is part of the delay in restored models	B.31	E	suppressing it advances structured output and generalization against ordinary training and a scaling control in 12/12 runs (1,300 and 1,450); later content: no consistent gain
Replenishment in ordinary training, modulus 97	B.35	F	deletion with suppression earlier on both onsets 8/8 and 8/8 (0.88 and 0.93 of the masking gain; medians $-4{,}000$, $-2{,}350$); suppression alone 6/8 and 8/8 (rule missed in addition); maintenance ratio 1.08 and 1.06
Replenishment in ordinary training, weight decay 0.5	B.34	F	deletion with suppression earlier on both onsets 8/8 and 8/8 (0.92 and 0.88 of the masking gain; medians $-6{,}850$, $-6{,}250$); suppression alone 7/8 and 5/8 (rule missed in multiplication; full onset alone 7/8 and 8/8, $-1{,}300$, -800); maintenance ratio 1.21 and 1.22
Replenishment in ordinary training, two layers	B.33	F	deletion with suppression earlier on both endpoints 8/8 and 8/8 (0.91 of the masking gain); suppression alone 6/8 and 7/8, median 700 (rule missed in addition); maintenance ratio 0.894
Additional benefit after deletion, two layers	B.33	E	32 new continuations on 16 known runs; full onset earlier than deletion 13/16, paired median 850 $[300, 1{,}400]$; than deletion with scaling 12/16, 550 $[200, 900]$; both endpoints earlier than both controls 6/8 per task
Additional benefit after deletion, two layers, fresh seeds	B.33	F	rule met 7/8 and 8/8; full onset earlier than deletion 15/16, paired median 1,250 $[300, 1{,}700]$; than deletion with scaling 15/16, 800 $[200, 900]$; run last; rule hashed before any model was created; added to the submission after the outcome
Additional benefit after deletion, weight decay 0.5 and modulus 97	B.33	E	32 new continuations on 16 known runs each; full onset earlier than deletion 16/16 and 15/16, paired medians 2,300 $[1{,}800, 3{,}000]$ and 1,350 $[800, 1{,}900]$; than deletion with scaling 16/16 and 15/16, 1,800 $[1{,}250, 2{,}400]$ and 750 $[500, 1{,}100]$; both endpoints earlier than both controls 8/8, 8/8 and 8/8, 7/8
Onset reading for the shared-deletion comparisons	B.33	P	thresholds 0.98 or 0.99, three or five evaluations; original reading reproduces 192/192 onsets; earlier counts unchanged in all settings; two-layer leave-one-out medians 800–900 and 400–700
Duration, release and clamp	B.37	F	suppression for 2,001–4,000 earlier in 12/16 (pooled 27/32, median 800); continuous suppression earlier 7/8 and 6/8 (R4 fails); longer suppression adds little (R1 fails); release regrowth: 0 of 17 low-share, 27 of 32 high-share, Spearman 0.86 (low class passes, high count fails); clamp at initial size does not bind (R3 fails)

Finding	Where	Kind	Key numbers
Holding the content in place	B.38	F	floor at the fork value to 12,000 later than ordinary training by 200 in 5/8 and 6/8 (R1 fails; medians $+350$, $+500$); binding 16/16, onset before release 13/16; o control 1/8 and 1/8; suppression throughout earlier 8/8 and 8/8 (medians $-1{,}900$, $-2{,}050$)
Holding the content in place, two layers	B.38	E	floor later than ordinary training by 200 in 4/8 and 6/8 (R1 fails; medians $+250$, $+550$); binding 16/16; onsets inside the hold 7/8 and 8/8
Why the content is maintained	B.39	F	deleting it at 2,000 costs 6.2–11.4 nats in 16/16; at the same learning rate, the raw gradient step replenishes only 1.5×10^{-8}–2.2×10^{-5} of the decay displacement in 15/16 states, and depletes in the other; original R2 statistic and gate retained with corrected definition
When removal acts	B.40	E	masked lag 1,350/1,300 at 2,000, 850/1,050 at 4,000, 900/1,100 at 5,000 (R1 mixed); masking at 5,000 earlier than ordinary training 6/6 and 7/7 (R2 met), saving 0.76/0.73 of the wait; deletion with suppression 0.86–0.92 of the masking gain at every fork; motion outside S straighter under the mask 16/16 but 0.987 against 0.974
Epsilon and direction, one layer	B.42	E	AdamW ϵ 10^{-12} earlier than 10^{-8} by 200 in 5/8 and 6/8 (R1 fails; medians -350, $-1{,}050$); median alignment to each branch's own final embedding 0.53 under masking, -0.22 under normal training; masking higher 16/16 (R3 met), normal negative 14/16
Epsilon coordinate audit	B.42	P	Smaller ϵ: full onset earlier/tied/later 14/1/1 (one delay of 100); coordinate higher at 4,000 in 11/16 but higher on average over 2,001–4,000 in 5/16; no sustained stronger hold
Radial update control on the restoration cohort	B.31	E	24 continuations on the 12 known runs; discrimination reading ($I_i - J_i > 0$ on both endpoints) fails, 2/6 per task; radial arm earlier than ordinary training in 12/12 for early and later content (full onset 1,750 and 850); early-minus-later gain 1,000 and 1,150 against the directed suppression's 1,500 and 1,300
Precision audit of saved-state loss and gradients	B.39	P	float32 against float64 at 48 states: relative to float64, the error-signal norm changes by at most 3.2% and the gradient along d by at most 18%, cosine 0.89 to 1.00; float32 softmax exactly one for a median 75 (at most 1,286) of 3,830 pairs, float64 at most 2; a plain gradient step returns at most 2.7×10^{-4} of decay in either precision
When removal acts, two layers	B.41	E	masked lag 1,200/1,200 at 2,000, 1,200/900 at 3,000, 1,200/800 at 4,000 (R1 mixed); masking at 4,000 earlier than ordinary training 5/5 and 7/7 (R2 met), saving 0.77/0.70 of the wait; deletion with suppression 0.90/0.93 of the masking gain at 2,000, 0.68/0.81 at 4,000 as the coordinate along d falls to 0.72–0.93; straightness 0.991 against 0.990
Direction and epsilon, two layers	B.42	E	masking higher alignment 16/16 (protocol R1 met), normal negative 7/16; smaller ϵ earlier by more than 200 in 2/8 and 1/8 (protocol R3 fails), paired medians -100, -100; reference onsets agree in every run
Signal versus response	B.44	P, E	swapping the error signal alone reproduces the removed model's preconditioned gradient direction (cosine 0.90/0.61/0.76; raw 0.97/0.96/0.94) while swapping the response does not (0.04/0.25/0.25); readings met on the duration cohort 15/16, 16/16, 16/16 and 16/16
A circuit built without the content	B.44	E	loss from a forward without the content, content held in the weights: the removed forward generalizes at 3,850/4,000 (masking 3,350/3,300) while the actual model waits for release (13,000/13,050; later than the clamp 7/8, 8/8; R1 and R2 fail); accuracy 0.53 at 12,000
A scalar at arrival does not predict the delay	B.44	P	fit-intact wins within run 13/16 but pools to 0.25 (rule 0.5); energy 3/16, -0.03; at weight decay 0.5: 10/16, 0.07 and 6/16, 0.17; both readings fail; own-content pairs fit better and delay in 16/16 and 15–16/16
Second copy of the fit outside S (failed replication)	B.43	E	masked-evaluation training loss falls by 0.5 nat in 7/8, 8/8 then 6/8, 7/8 (R1 fails on the second cohort); masked test accuracy below 0.10 in 14/16 then 10/16 (fails); content built outside S and K redundant in 16/16 then 14/16 (fails) and receding in 16/16 then 15/16 (met); the embedding's displacement alone raises the masked loss
Where the gradient goes	B.36	P	deleted content rebuilt to 0.53 by 2,600; every neuron reads the early content at share 0.49; gradient share along d 0.58 in ordinary training, 0.02 after deletion with suppression, key-frequency share 0.79; regrowth after release where the share is 0.46, none at 0.02
Leakage under rotation	B.14	P	mean leakage of clean planes 0.050, 0.183, 0.353 at $t = 0.25$, 0.5, 0.75, as the formula gives; accuracy falls with leakage across rotations (Spearman -0.82), not within one

Finding	Where	Kind	Key numbers
Not amplitude, not any disturbance	B.26	E	early content at the confirmed amplitude 2,300, 1,900 more in 16/16; Gaussian content at the fork 0.69 against deletion 0.72
Another run's content causes no delay	B.27	E	partner content delay 0, retained 0.24 as scrambled; scaled confirmed content later than sham in 2/16, relied on 0.80
The relied-upon part of S causes little of the delay	B.28	E	K_S alone 100 updates, like a random four (within 200 in 9/16); the other 24 carry 0.944 of the delay; not proportional to restored norm
Nested subsets and a random 24	B.29, B.30	E	delay shares 0.12 to 0.61 at 8 to 20 frequencies, below energy shares 0.28 to 0.71 (intermediate); random 24 unresolved, within 200 in 6/16
The fit is regained at once	B.46	P	every arm back above 99% training accuracy within one or two records
No detectable relapse	B.47	P	no arm worse than its own normal branch at a sign-test p below 0.21
Circuit-efficiency signature not resolved	B.49	P	fastest arm's excluded rise $+0.8$ $[-1.2, +2.9]$; the interval includes the predicted effect
The fork precedes generalization, not formation	B.51	P	formation onset precedes the fork in 16/16
A branch can leave the normal circuit and still be fast	B.52	P	most relied-upon half keeps 0.002 of its logit power there; earlier 16/16
Earlier panels, 12 task–activation cells	B.8	E	core removed 1.67; others removed to the core's energy 0.294

The main conclusions under a family-wise correction. Table 4 lists the nine conclusions the main text rests on, each as a paired count over 16 runs (F: fresh-seed confirmation; E: extension on the restoration cohort) with its exact two-sided sign-test p and the Holm-adjusted p across the nine. All nine survive; the largest adjusted value is 4.2×10^{-3}. The correction covers these conclusions, not every gate in the appendices.

Table 4: **Main conclusions, paired counts over 16 runs, and Holm-adjusted sign-test p-values.**

Conclusion	Count	Kind	Where	p	Holm
Masking a random half brings generalization forward	16/16	F	B.1	3×10^{-5}	2.7×10^{-4}
The same at modulus 97	16/16	F	B.2	3×10^{-5}	2.7×10^{-4}
The same in a two-layer model	16/16	F	B.3	3×10^{-5}	2.7×10^{-4}
Restored early content is later than both controls	16/16	F	B.20	3×10^{-5}	2.7×10^{-4}
Early content is later than later content at equal amplitude	16/16	E	B.26	3×10^{-5}	2.7×10^{-4}
Suppression beats ordinary training and the scaling control (first cohort, prespecified rule)	14/16	F	B.32	4.2×10^{-3}	4.2×10^{-3}
Deletion with suppression is earlier than ordinary training (first cohort)	16/16	F	B.32	3×10^{-5}	2.7×10^{-4}
The same at depth two	16/16	F	B.33	3×10^{-5}	2.7×10^{-4}
The 24 frequencies outside K_S delay generalization against sham	16/16	E	B.28	3×10^{-5}	2.7×10^{-4}

Analyses added after the confirmatory runs. The deletion controls at depth two, at weight decay 0.5 and at modulus 97 (Appendix B.33) reuse the original checkpoints of those cohorts. Their paired intervals are descriptive and add no confirmation gate; the original criteria remain unchanged. The ϵ coordinate audit (Appendix B.42) is post hoc. Neither analysis is added to the nine-comparison Holm family. The fresh-seed two-layer confirmation of the deletion comparison (Appendix B.33) was run last, after every other experiment. Its rule was hashed before any of its models was created, and its protocol requires a failure to be reported as a failure; it was added to this submission after its outcome was known. It is outside the Holm family. The norm-matched radial control on the restoration cohort (Appendix B.31) likewise reuses known runs; its single prespecified reading failed and is listed as such. The precision audit (Appendix B.39) and the restored-norm check (Appendix B.26) are post hoc and touch no run.

The prespecified-gate ledger. More than twenty prespecified gates appear across these experiments. Holm-adjusted values are reported for the nine conclusions in Table 4; **the remaining tests have no family-wise correction**. Their protocols fix different quantities and criteria, but the individual values outside that table do not control error across the full collection. Of the tests on fresh runs, the main confirmation (Appendix B.1) and its modulus-97 and two-layer repeats (Appendices B.2 and B.3), but not its two-layer LayerNorm repeat (3/8 per task at the 2,000 fork and 6/8 and 5/8 at a 1,000 fork; Appendix B.5), the restoration confirmation (Appendix B.20) and its weight-decay 0.5 replication (Appendix B.23) met their prespecified criteria, and the new-run ranking test met

its criterion at update 2,000, with the 1,000-update analysis secondary; the window conjunction did not, missing its within-margin condition at 13/16 (Appendix B.18), and a further prespecified cohort failed its own validity gate (Appendix B.56). The replenishment test, fixed after four pilot runs and run on the other twelve restoration runs, met both of its readings (Appendix B.31); the replenishment test in ordinary training, on 16 fresh runs, met both primary readings and missed its secondary one on the complementary half (Appendix B.32); and the restoration in the two-layer model did not meet its rule, at 6/8 per task (Appendix B.4). The subset-size and random-24 extensions returned their *intermediate* and *unresolved* labels (Appendices B.29 and B.30). The gradient-routing, masked-lag and leakage readings are post hoc and carry no gate. Their prospective test on 16 fresh runs (Appendix B.37) passed the release reading's low-share class (0 of 17 regrew) and failed its high-share count (27 of 32, five violations), the duration reading and the clamp reading, and narrowly missed its continuous-suppression criterion (7/8 and 6/8). At depth two (Appendix B.33) deletion with suppression met its rule (8/8 and 8/8) and suppression alone did not (6/8 and 7/8); on fresh two-layer runs, suppression after the same deletion met its rule against deletion alone and deletion with scaling (7/8 and 8/8; Appendix B.33). The direction extension passes its alignment criterion in 16/16, while its epsilon criterion fails (2/8 and 1/8; Appendix B.42). The restoration confirmation's secondary rule for addition, delay together with more than 20 extra points of damage, held in 4/8 (Appendix B.20). The orthogonal-control reading, suppression earlier than the orthogonal control on both onsets in at least 7 of 8 runs per task, failed in all four ordinary-training cohorts: 6/8 and 7/8 at one layer, where ties count against it, 6/8 and 6/8 at two layers, 7/8 and 5/8 at weight decay 0.5, and 6/8 and 7/8 at modulus 97 (Appendices B.32, B.33, B.34, B.35). The floor's sufficiency reading failed at both depths (5/8 and 6/8; 4/8 and 6/8; Appendix B.38); the epsilon reading failed at one layer as well (5/8 and 6/8; Appendix B.42); the second-copy account failed three of its four readings on replication (Appendix B.43); and the norm-matched radial control's discrimination reading (2/6 per task), both readings of the training run with the content out of view, and the two arrival-scalar readings failed (Appendices B.31, B.44). On the discovery cohort, the sampled-mask prediction test returned its strongest prespecified outcome (Appendix B.53). The background test estimates a directional witness yield and specifies no pass-count gate (Appendix B.54). Appendix B.48 gives distribution-free intervals for two gates that were prespecified as one-sided counts and fell short. The orientation, nested, window and fitting–delay arms are extensions specified with earlier outcomes known, and the scope column of each row above says so. The relocation, phase-timing, drawdown and paired-interval analyses carry no gate at all and are labeled post-hoc wherever they appear.

B Frequency Removal: Protocols, Confirmations and Controls

Table 5: **The sets used in the main text.**

Set	Size	Chosen by	Endpoint information
core / non-core	4 / 4	most energetic at the historical endpoint / four others	yes
low set	28	below the median source energy	no
random set R	28	uniform draw, fixed before initialization	no
K	4	most damaging single frequencies at the normal endpoint	yes
B	24	fixed rule from source measurements, disjoint from K	via K

This appendix holds the experiments behind the paper's claims: the protocols that fixed each design, the independent checks' outcomes, and the controls and failed gates that bound the reading. Appendices B.1–B.11, B.50 and B.53–B.55 support Section 3.1 (masking a relied-on set, the choice of set, evaluation-only masking); Appendices B.16–B.19 support Section 3.2 (deletion, freeze and window); Appendices B.20–B.30 support Section 3.3 (content replacement); Appendices B.31–B.39 support Section 3.4 (the rebuild and its suppression); Appendices B.40–B.44 are the mechanism probes discussed in Section 5; and Appendices B.12–B.15, B.45–B.49, B.51, B.52 and B.56 hold controls, robustness checks and failed gates.

B.1 Main Confirmation of Damage and Earlier Generalization on New Seeds

This panel tests the paper's central pairing as a single prespecified event on training runs that did not exist when the earlier analyses were written.

Seeds and sets. Sixteen runs were used: ReLU addition seeds 910501–910508 and ReLU multiplication seeds 910511–910518. Each run's removal set R of 28 frequencies was written to the manifest before any formal model was initialized, without reading weights, energy rankings or any later outcome. No earlier experiment had used these seeds, and sets and seeds were not exchanged on the basis of later behavior. The protocol, manifest, scripts and tests were frozen before any model was initialized.

Design. Each run forks at update 2,000 into four branches, retaining AdamW state, counters and random states, and runs to update 50,000: normal continuation, removal of R, norm-preserving removal of R, and the matched scaling of R. The normal branch is additionally scored under each of the two removal operators as an evaluation-only curve. That gives 64 training executions.

Prespecified joint event. Per run: both removal operators reduce the normal model's test accuracy by more than ten percentage points at the checkpoint where that model first reaches stable generalization after the source, *and* both corresponding branches with sustained removal reach stable generalization earlier than the normal branch. A run whose normal branch already generalized before the source, or which never attains within budget, is retained and scored as not meeting the event rather than dropped from the denominator. The prespecified rule requires at least 7 of 8 runs in each task, and both tasks must pass; pooling cannot rescue a task that fails.

Outcome. The joint event holds in 8/8 addition and 8/8 multiplication runs (Table 6). Under a null in which each run meets the event independently with probability one half, the one-sided count in a task of eight gives $p = 2^{-8}$. The prespecified rule requires both tasks, so the reported value is the intersection-union statistic, the maximum of the two per-task p-values. All 64 executions completed and passed the integrity checks. Independent reconstruction reproduced the recorded scores with maximum cross-entropy error 0.0 and maximum accuracy error 1.11×10^{-16}; the maximum relative norm error is 2.79×10^{-7}.

Table 6: **Main confirmation, 16 fresh runs.** Median paired ratios of total updates against normal training, paired onset gaps in updates as Hodges–Lehmann estimates with distribution-free 95% intervals, and the test-accuracy damage from ablating the same set from the normal branch. The lower block reports ordinary removal, not norm-preserving removal. "Earlier" and "over 10 pp" count runs out of 16; the evaluation-only comparisons involve censored curves.

Branch	Median ratio	Gap, HL [95% CI]	Earlier	Beats eval-only
Removal of R	0.488	$-3{,}475$ $[-4{,}450, -2{,}700]$	16/16	16/16
Norm-preserving removal	0.481	$-3{,}550$ $[-4{,}400, -2{,}700]$	16/16	16/16
Matched scaling	1.011	$+150$ $[-100, +400]$	6/16	—
Damage audit point	**Median (pp)**	**HL [95% CI] (pp)**	**Over 10 pp**	
Normal first stable onset	66.4	65.7 [52.8, 79.8]	16/16	
Normal confirmation	64.4	64.5 [51.1, 78.5]	16/16	
Onset $+1{,}000$ updates	64.5	69.3 [55.2, 80.6]	16/16	
Update 50,000	88.0	83.3 [74.5, 91.5]	16/16	

By task, median ratios of total updates for removal are 0.523 in addition and 0.469 in multiplication, and for norm-preserving removal 0.496 and 0.467. Matched scaling is later than normal in 8 of 16 runs, earlier in 6 and tied in 2. Both evaluation-only branches were censored in every run in which the corresponding sustained branch attained.

Reading and limits. The dependence measured at update 50,000 is not an artifact of training far past generalization: ablating the same set already reduces accuracy by a median 66.4 points at the checkpoint where the normal model first generalizes, in 16/16 runs. Because the two operators, the two audit points and the two tasks are properties of the same runs, they are not additional independent seeds, and the prespecified analysis treats one run as the unit. The panel shares the architecture, task family, data fraction and 2,000-update trigger of the discovery cohort, so it replicates that design on new seeds rather than extending its scope. Following an early-attainment case observed in the matched-control extension, this panel also records the last ten evaluations, endpoint accuracy and

cross-entropy, and the rate of falling back below 0.99 after confirmation; those are secondary and do not replace the prespecified first-attainment criterion.

The stricter secondary joint event additionally requires the normal, removal and norm-preserving removal branches all to have accuracy at least 0.99 at each of their last ten evaluations. It holds in 14/16 runs (7/8 per task). The two exceptions arise from different branches: normal training in addition seed 910506 briefly falls to 0.9757 and ends at 1.0000; ordinary removal in multiplication seed 910518 falls to 0.9870 and ends at 0.9964. All 16 branches with norm-preserving removal pass the criterion for the last ten evaluations. These secondary outcomes qualify persistence without changing the prespecified first-attainment result.

What the prespecified p carries. Two disclosures bound it. The prespecified rule asks for 7 of 8 runs in each task, so its intersection–union size is 0.0352 and 0.0039 is the smallest value attainable at eight runs per task: the reported figure is the floor of the test, not a measure of surprise. And neither conjunct was a coin flip a priori. Earlier generalization under a half-set mask was already 16/16 in the discovery cohort and 24/24 in the historical below-median arm before this protocol froze; and a random half of a 56-character basis damages a solution concentrated on four or five of them with high prior probability, which the realized base rate confirms — 15 of 16 random sets reduce the normal endpoint's accuracy by between 49.6 and 96.9 points (Appendix B.12). What this test records is the *co-occurrence* of the two outcomes on runs fixed in advance, not the unlikelihood of either one alone.

B.2 The Main Confirmation at Modulus 97

This batch changes one factor of the main confirmation, the modulus, and repeats its four-arm design and its prespecified rule on 16 fresh runs: ReLU addition seeds 940501–940508 and multiplication seeds 940511–940518, with every other setting of the recipe unchanged. At $p = 97$ there are 48 nonconstant frequencies; each run's random set is 24 of them, drawn from the run identifier alone by the main confirmation's rule with a new mask domain, and multiplication is ordered by powers of a primitive root of 97. The protocol, written before any run was trained and with every other result of the paper known, expected confirmation with removal ratios near 0.5 and matched scaling near 1.

Outcome. Table 7 gives the numbers. The prespecified event holds in 7 of 8 addition and 8 of 8 multiplication runs, so the batch confirms. The one addition run that does not count, seed 940502, fails on damage, not on timing: ablating its set reduces the normal model's accuracy by 7.7 points at onset, below the ten-point threshold, while removal still brings its onset from 12,500 to 4,200. Every run generalizes earlier under both removal operators; matched scaling is later in 6, earlier in 8 and tied in 2. The evaluation-only curves never attain in 15 runs, and in the sixteenth (seed 940508) attain at 14,000 and 14,600 against 3,100 for the sustained branches. The normal, removal and norm-preserving removal branches all hold at least 0.99 over their last ten evaluations in 16 of 16.

Table 7: **Main confirmation at modulus 97, 16 fresh runs.** Median ratios of total updates against normal training (median normal onset 7,050), paired onset gaps as Hodges–Lehmann estimates with distribution-free 95% intervals, and the test-accuracy damage from ablating the set from the normal branch at its first stable generalization.

Branch	Median ratio	add / mul	Gap, HL [95% CI]	Earlier
Removal of R	0.518	0.539 / 0.478	$-3{,}550$ $[-4{,}350, -2{,}900]$	16/16
Norm-preserving removal	0.518	0.547 / 0.475	$-3{,}500$ $[-4{,}350, -2{,}800]$	16/16
Matched scaling	0.993	1.000 / 0.986	0 $[-150, +150]$	8/16
Damage at normal onset	median 69.6 points, HL 66.3 $[49.0, 79.5]$, over ten points in 15/16			

Reading. The pairing of the main confirmation, a set the normal model relies on whose masking brings generalization forward, holds at a second modulus with the same recipe. It says nothing about other architectures or data fractions.

B.3 The Main Confirmation in a Two-Layer Model

This batch changes one factor of the main confirmation, the number of Transformer layers, from one to two, still without LayerNorm, and repeats its four-arm design and its prespecified rule on 16 fresh runs: ReLU addition seeds 950501–950508 and multiplication seeds 950511–950518, with every other setting of the recipe unchanged and the random set of 28 frequencies fixed by the same rule before initialization. The protocol was written before any formal run was trained, with every other result of the paper known. One addition run, 950501, had been run as a smoke before the protocol was written and stays on the roster, trained again from its seed. The budget is 40,000 updates instead of 50,000, fixed in advance to fit the run time; every normal branch attains within it (latest onset 12,200), and no source had generalized at the fork (test accuracy at most 0.146).

Outcome. Table 8 gives the numbers. The prespecified event holds in 8 of 8 addition and 8 of 8 multiplication runs, so the batch confirms; without the smoke run, addition is 7 of 7. Every run generalizes earlier under both removal operators; matched scaling is later in 6, earlier in 8 and tied in 2. The evaluation-only curves never attain. The normal, removal and norm-preserving removal branches hold at least 0.99 over their last ten evaluations in 13, 12 and 14 of 16.

Table 8: **Main confirmation in a two-layer model, 16 fresh runs.** Median ratios of total updates against normal training (median normal onset 6,500), paired onset gaps as Hodges–Lehmann estimates with distribution-free 95% intervals, and the test-accuracy damage from ablating the set from the normal branch at its first stable generalization.

Branch	Median ratio	add / mul	Gap, HL [95% CI]	Earlier
Removal of R	0.462	0.490 / 0.423	$-3{,}600$ $[-4{,}800, -2{,}650]$	16/16
Norm-preserving removal	0.452	0.474 / 0.431	$-3{,}550$ $[-4{,}600, -2{,}500]$	16/16
Matched scaling	0.993	1.007 / 0.980	-50 $[-150, +100]$	8/16
Damage at normal onset	median 80.6 points, HL 80.0 $[72.0, 89.9]$, over ten points in 16/16			

Reading. The pairing of the main confirmation holds in a second architecture of the same family. Both depths lack LayerNorm and share the task, data fraction and trigger, so this says nothing about other architectures.

B.4 Restoration in the Two-Layer Model

On the 16 runs of Appendix B.3, a random half is masked for updates 2,001–2,099 and unmasked at 2,100, where it receives the normal branch's content (native), a scrambled version of equal energy (phase) or nothing (sham), as in the one-layer restoration confirmation. The budget is 40,000 updates. The protocol was frozen before the two-layer main confirmation's outcomes were known. The reading fixed in advance: restoration is confirmed if native is later than both sham and phase by more than 200 updates in at least 7 of 8 runs per task.

Outcome. Native is later than both controls by more than 200 updates in 6 of 8 addition and 6 of 8 multiplication runs, so the restoration is *not confirmed*. Native is later than sham by more than 200 in 7 and 7, later than phase by more than 200 in 7 and 6; median onsets are 3,100 for sham, 3,500 for native and 3,100 for phase, and the median native delay against phase is 550 updates, against 2,600 in the one-layer model. Phase lies within 200 updates of sham in 14 of 16. On arrival, native content raises the test cross-entropy above phase in 16 of 16, as in the one-layer model, but lowers the training cross-entropy in only 1 of 16, where the one-layer model did in 16 of 16.

Reading. In the two-layer model the early content still delays generalization in most runs, but by less, below the prespecified rule, and without the lower training loss on arrival that it brings in one layer.

B.5 The Main Confirmation in a Two-Layer Model with LayerNorm

This batch switches on pre-LayerNorm in the two-layer model of Appendix B.3: each block applies a LayerNorm before its attention and before its MLP, with learned gain and bias trained by the same AdamW; there is no final LayerNorm. Everything else follows the main confirmation's recipe, with 16 fresh runs (ReLU addition seeds 960501–960508, multiplication 960511–960518), the random set fixed by the same rule before initialization, the same four arms and the same prespecified rule, and a budget of 30,000 updates fixed in advance from the run time. The protocol was written before any formal run was trained and before the no-LayerNorm batch's results were read. Two multiplication runs, 960517 and 960518, did not run to completion; they stay in the denominator and count as no event.

Controls under LayerNorm. The protocol noted in advance that norm-preserving removal and matched scaling are not the same controls here, since the LayerNorm renormalizes whatever the embedding provides; in the smoke, matched scaling gave 0.68. It also noted that LayerNorm models show repeated loss spikes after generalizing, so the auxiliary last-ten check was expected to fail often.

Outcome. The prespecified event holds in 3 of 8 addition and 3 of 8 multiplication runs, so the batch is *not confirmed*. The two sustained removal arms are earlier than normal training in 9 and 8 of the 14 completed runs (median ratios 0.921 and 0.898); matched scaling is earlier in 2 (1.258). Ablating the set at onset removes a median 90.6 accuracy points; in one multiplication run, 960513, it removes none, so the normal solution there does not use the set. The eval-only curves never attain. Over the last ten evaluations, normal, removal, norm-preserving and scaling branches hold 0.99 in 9, 7, 5 and 4 of 14.

Reading. Normal training in this model generalizes a median 3,550 updates after initialization, that is 1,550 after the fork, against 4,500 to 5,050 after the fork in the other cohorts, and its onsets range from 2,200 to 5,600. The masked branches generalize a median 1,150 updates after the fork, as in every other cohort (Appendix B.6). In the 7 runs with at most 1,500 updates left at the fork, masking is earlier in 4 (median difference -200); in the one run with 3,000 or more left it is earlier by 2,000. The rule fails, then, where the fork falls close to generalization, and this appendix does not separate that timing from LayerNorm itself.

The same test at an earlier fork. If the failure reflects timing, forking earlier, where normal training still has more left, should restore the effect. A prospective batch of 16 fresh runs (ReLU addition seeds 970501–970508, multiplication 970511–970518), frozen before initialization, repeats the design with the fork at update 1,000, three branches (normal, ordinary and norm-preserving removal), and a budget of 15,000; matched scaling is omitted, as it is not the same control under LayerNorm. Two readings were fixed: the P5 rule, and a lag reading that random removal is earlier than normal in at least 7 of 8 runs per task among runs whose normal branch needs at least 2,000 updates after the fork. Outcome: normal training now generalizes a median 2,300 updates after the fork (range 1,500 to 3,500). Random removal is earlier than normal in 8 of 8 addition and 6 of 8 multiplication runs (median 900 updates earlier; ratio 0.684), the masked lag is a median 1,400 and 1,450 updates for ordinary removal and 1,350 and 1,100 for norm-preserving removal, within the 900 to 1,600 the protocol expected and close to the 1,050 to 1,350 medians of the cohorts forked at 2,000 (Appendix B.6). The P5 rule is still *not met*, at 6 of 8 and 5 of 8: one addition run's solution does not use the set (damage 0.0), one norm-preserving branch is later than normal in each task, and in multiplication one random branch ties normal and one is later by 2,100 updates. The lag reading is *met in addition* (7 of 7 eligible) and *not in multiplication* (2 of 4: the tie and the reversal). Two sources had training accuracy below 0.99 at the fork and are reported, not excluded. Forking earlier moved the LayerNorm model most of the way toward the other cohorts without reaching the rule.

B.6 How Long Generalization Takes Once the Half Is Masked

A post-hoc reading across the cohorts that mask a random half from update 2,000. Table 9 gives, per cohort, the median number of updates normal training still needs after the fork and the median number the masked branch needs. The masked branch's lag is 1,050 to 1,350 updates in every cohort, the LayerNorm model included, while the normal branch's varies from 1,550 to 5,050. Pooling

the 78 runs, the masked branch is earlier than normal in all 58 runs where normal training needs 3,000 or more updates after the fork, and in 2 of the 5 where it needs fewer than 1,500. The pooled count is descriptive: the gain from masking is, by construction, the normal branch's remaining time minus the masked branch's lag, so what the table adds is that the lag itself hardly varies. In ordinary training the early content along the fixed direction of Appendix B.32 is maintained until shortly before generalization; masking removes it at the fork, and generalization follows about 1,200 updates later. The prospective test of this reading, the LayerNorm model forked at 1,000 (above), gave lags of 1,100 to 1,450 and random removal earlier in 14 of 16 runs, though its rules were met in addition only.

Table 9: **Updates to generalization after the fork at 2,000, with and without the random half masked.** Medians over the runs of each cohort; the LayerNorm cohort has 14 completed runs.

Cohort	Normal	Masked	n
Main confirmation, one layer	4,600	1,350	16
Modulus 97	5,050	1,300	16
Two layers	4,500	1,050	16
Ordinary-training replenishment test	4,700	1,100	16
Two layers with LayerNorm	1,550	1,150	14

B.7 Masking from Other Fork Times

Unless stated otherwise, interventions fork at update 2,000. This batch masks the same set S from updates 1,000, 2,000 and 3,000 on the restoration confirmation's 16 runs, one arm per fork, each to 50,000 with the fork state's own randomness and the ordinary removal operator. The 1,000 state was rebuilt by re-running the prefix from the seed and checked to continue to the saved update-2,000 source bit for bit; the 3,000 state by continuing the saved source and checked against the normal branch's recorded accuracy at 3,000; all 16 checks passed at each fork. The reading fixed in advance: a fork passes if the masked arm is earlier than normal by more than 200 updates in at least 7 of 8 runs per task; expected, a pass at all three, with the saving largest at the earliest fork.

Outcome. Masking brings generalization forward at all three forks, earlier than normal in 16 of 16 runs at each, with no censoring.

Fork	Median onset	Ratio of total updates	Ratio of remaining updates	Updates saved
1,000	2,200	0.262	0.164	5,700
2,000	3,200	0.396	0.204	4,650
3,000	4,050	0.517	0.230	3,800

The saving shrinks as the fork moves later while the remaining-update ratio barely changes. These runs were not fresh: their normal branches and the restoration outcomes were known, so this batch describes the fork dependence of a known effect and tests nothing on new seeds.

B.8 Earlier Cohorts: Twelve Task–Activation Cells

Before the discovery cohort, the same questions were asked on earlier panels that cross three activations (ReLU, GeLU, SiLU) with four operations (addition, subtraction, multiplication and division modulo 113), 12 cells per seed, each forked from an accuracy-defined source before generalization with a 25,000-update budget (Appendix C). The *core* is the four frequencies a frozen readout rule selects from the original endpoint's operand embedding, the endpoint's analogue of K. Each experiment below had its own protocol with gates fixed before any cell, and every gate passed; the full protocols, gates and per-cell tables are in the artifact.

Across activations and operations the pattern of the main text appears in miniature (Table 10). Removing the four frequencies the endpoint relies on most delays generalization, and restoring the embedding norm after removal does not undo that (1.654 against 1.672 without rescaling). Removing other frequencies brings it forward, even when they carry more source energy than the core: the

Table 10: **Earlier panels.** Median onset of stable generalization relative to the unmodified branch of the same cell, with the number of cells earlier than unmodified where recorded. Seed 900006 is the development seed, 900003–900004 the confirmation seeds, 900007–900008 a later seed extension.

Arm	Seeds (cells)	Ratio	Note
Core removed	900006 (12)	2.38	later than four random removed in 12/12
Core removed	900003–4 (24)	1.67	attains 23/24; later than random removal in 21/24
Core removed	900007–8 (24)	3.29, 2.60	later than random removal in 45/48 cells over four seeds
Four random frequencies removed	900003–4 (24)	0.70	0.68–0.75 per seed over all four seeds
Core removed, norm restored	900003–4 (24)	1.654	later than unmodified in 20/24
Core frozen at source content	900003–4 (24)	1.30	
Only core and mean may change	900003–4 (24)	1.25	attains 24/24
Others removed to the core's source energy	900003–4 (24)	0.294	earlier in 24/24; median 21 frequencies
Lowest 4 / 8 by source energy	900003–4 (24)	0.877 / 0.748	earlier in 23/24 and 24/24
All 28 below median source energy	900003–4 (24)	0.300	earlier in 24/24

energy-matched arm removes a median 21 frequencies and generalizes at 0.294 of the unmodified onset, and its endpoints still hold a median 0.935 of their energy on the original core. Selecting by source energy alone, with no endpoint information, brings it forward as well. These panels share one architecture, use two to four seeds, and define the core on the endpoint; the fresh-seed cohorts of the main text supersede them wherever the two overlap.

B.9 Frequency Interventions on the Discovery Cohort

Table 11: **Discovery cohort: effects on generalization time with retained AdamW.** Sixteen runs, eight independent seeds per task. These are *not* the runs of the main confirmation in Section 3.1 and Figure 1; the two panels share a design but no runs, which is why evaluation-only removal is censored in 16/16 there and attains in 14/16 here (†, with one censor per task excluded from its ratios). Median paired update ratios against normal training. Total includes the 2,000-update prefix; remaining excludes it. "Earlier" counts onsets across both tasks. All training arms attained in 16/16. Frequency masks remove 28 of the 56 frequencies; scaling and magnitude pruning are alternative controls.

	Addition		Multiplication		
Intervention	Total	Remaining	Total	Remaining	Earlier
Low-energy removal	0.490	0.269	0.479	0.312	16/16
Norm-preserving removal	0.496	0.290	0.483	0.312	16/16
Random frequency removal	0.507	0.302	0.439	0.232	16/16
Matched scaling	1.000	1.000	1.044	1.063	3/16
Embedding magnitude pruning	1.200	1.257	1.251	1.361	0/16
Evaluation-only removal	1.015	1.021	1.018	1.022	0/14†

This experiment tests whether masking frequencies brings generalization forward from a source chosen without future information, and whether simple norm or pruning controls reproduce the benefit. It follows the retrospective evaluation-only experiment of Appendix B.11. Its protocol, code, tests and seed manifest were frozen before any formal new-seed initialization. There was no success quota and no target of reproducing the earlier 0.30 ratio.

Cohort and source rule. Eight ReLU addition seeds (910201–910208) and eight ReLU multiplication seeds (910211–910218), each with a fresh initialization and data split, in the configuration of Appendix C in float32 throughout. The trigger is exactly 2,000 completed updates, using neither held-out metrics nor endpoint information; all six branches of a run inherit the source's weights, optimizer moments, counters and random states, and run to update 50,000 without early stopping, exclusions or budget extensions. No source had reached stable generalization (source test accuracy 0.028 to 0.303).

Fixed sets and intervention operators. Let $E \in \mathbb{R}^{128\times n}$ contain the operand embedding columns in group order, μ their mean, and $X = E - \mu$. Addition uses all 113 operands; multiplication uses

the 112 nonzero operands in multiplicative order and leaves token zero unchanged. The eligible set contains 56 nonconstant real frequencies. The fixed set S contains those strictly below the median source energy, here 28 frequencies in every run. A second fixed set R is drawn uniformly without replacement from all 56 frequencies, with $|R| = |S|$, using a seed derived from the run identifier. It is not restricted to the original endpoint's non-core set. For multiplication the Nyquist frequency has one real direction, so equal frequency counts need not give equal ranks.

Write Q_S for the orthogonal projector onto S and $L = X - XQ_S$. The six training operators are

$$E_{\text{normal}} = E, \qquad E_{\text{low}} = \mu + L, \qquad E_{\text{random}} = \mu + X - XQ_R,$$
$$E_{\text{norm}} = \mu + \text{sg}\left(\frac{\|X\|_F}{\|L\|_F}\right) L, \quad E_{\text{scale}} = \mu + \text{sg}\left(\frac{\|L\|_F}{\|X\|_F}\right) X, \quad E_{\text{magnitude}} = E \odot M. \tag{2}$$

Here sg stops gradients through the scalar ratio; gradients still flow through X and L. Denominators are clamped below at 10^{-12}; no logged norm diagnostic was degenerate. The norm-preserving arm restores its own pre-removal centered operand norm on every pass. Because it preserves μ, it also restores the full operand Frobenius norm. The largest logged relative discrepancy in that norm was 2.38×10^{-7}. Non-operand columns are not rescaled.

Matched scaling preserves frequency proportions while matching the norm that removing S would produce on that branch's current weights. Its norm equals the low-energy arm's at the source, but the branches' norms need not remain equal after their weights diverge. Each branch sets its own norm target; the experiment does not hold norms equal across branches.

The binary mask M zeros the smallest-magnitude raw operand entries at the source, with index-based tie breaking. It zeros $128\ \text{rank}(Q_S)$ entries, matching the number of scalar linear constraints imposed by frequency removal, not removed energy. Other parameter matrices are untouched. This baseline need not preserve the operand mean; it is ordinary embedding magnitude pruning, not a method that learns masks or prunes the full network. All training operators remain active during evaluation.

Evaluation-only control and outcomes. The normal branch also supplies the evaluation-only curve B: its unchanged weights are evaluated with the same fixed S mask as the sustained low-energy branch. Temporary evaluation preserves model tensors, gradients and random states. Every 100 updates we record training and test accuracy and cross-entropy (CE), before and immediately after the source intervention and throughout the 48,000-update continuation. Stable generalization requires five consecutive test accuracies at least 0.99. The onset is the first evaluation in that sequence, and confirmation occurs 400 updates later. Unconfirmed crossings do not count. Censored onsets remain null and are excluded from time ratios, with their counts reported separately.

Table 12 gives every onset. All 96 training branches completed the full budget and attained stable generalization. Low-energy, norm-preserving and random frequency removal each preceded normal training in 16/16 pairs. Matched scaling was earlier in three, tied in three and later in ten; magnitude pruning was later in all 16. Low-energy removal was earlier than random removal in five pairs, tied in one and later in ten (median onset gap for low-energy minus random removal: $+150$ updates). Thus the experiment supports earlier generalization with frequencies masked under this protocol, fixed before training, but provides no evidence that source energy ranking improves on this random selector. It is not an equivalence test.

Evaluation-only removal attained in 14/16 runs. The censored runs were addition seed 910208 and multiplication seed 910214, with final test accuracies 0.976 and 0.550. Sustained low-energy removal preceded all 14 attained B branches (median gap $-3{,}550$ updates), and attained in both censored cases. Its full-budget mean test CE was lower than B's in all 16. Mean CE is trapezoidal area over the 48,000-update continuation divided by 48,000, including censored curves. These comparisons support an effect on subsequent learning; they do not assign a mediation fraction to that effect or identify its optimization mechanism.

Scope. Source states, operational timings, mean-loss summaries and full traces are in the artifact. This targeted panel covers ReLU addition and multiplication at one fixed trigger. It does not establish generality over activations, depths, bases, trigger times or pruning methods.

Table 12: **Discovery cohort onset updates for every seed.** Subtract 2,000 for remaining updates and add 400 for confirmation. A: normal; L: low-energy; N: norm-preserving; R: random frequency; S: matched scaling; M: magnitude pruning; B: evaluation-only. "Cens." denotes no confirmed stable onset by update 50,000.

Task	Seed	A	L	N	R	S	M	B
add	910201	5100	2800	2900	3100	4900	5800	5100
add	910202	10100	4100	4100	4000	9900	11800	10100
add	910203	6800	3300	3300	4000	6900	8500	6900
add	910204	8200	4000	3900	4000	8000	9800	8200
add	910205	3600	2800	2800	2700	3600	3700	6300
add	910206	9800	4000	4100	3600	9800	11800	10000
add	910207	5900	3500	3500	3100	6100	7400	6000
add	910208	6500	3200	3300	3000	6800	8100	Cens.
mul	910211	11000	4200	4200	3200	12100	15400	11200
mul	910212	5300	3100	3100	2900	5500	6700	6500
mul	910213	8100	3900	3900	3100	8500	10300	8200
mul	910214	13100	4800	4800	4100	13600	17100	Cens.
mul	910215	4200	2700	2700	2900	4200	5100	4200
mul	910216	8400	4000	4000	3300	9000	10400	8600
mul	910217	6600	3100	3200	3200	6800	8000	6600
mul	910218	4800	3000	3100	3600	5400	5700	5000

B.10 Matched Controls for the Same Random Set

This extension keeps all 16 first-panel runs and their original random sets R, without reselecting sets or filtering seeds, and pairs each removal operator with its own acute audit and its own evaluation-only branch. It was designed after the first panel, the basis control and the original audit of the same sets at update 50,000 were known, so it adds controls and is not an independent confirmation on new runs. Its protocol, code and inputs were frozen before it ran.

Why the pairing matters. Ordinary removal and norm-preserving removal are different operators. An acute damage figure measured under one of them may not be paired with a change in generalization time measured under the other, and each branch has its own norm target rather than sharing a norm trajectory. This extension supplies both halves for each operator on the same runs.

Acute audit across training. Each of the 16 normal branches was audited at seven checkpoints: the 2,000-update source, the stable-generalization confirmation, and updates 10k, 20k, 30k, 40k and 50k. Every point was evaluated unmodified, under ordinary R removal, under norm-preserving R removal, and under the corresponding whole-embedding scaling. All 112 checkpoints verified with no missing entries; the independent evaluator reproduced the recorded cross-entropies exactly, with maximum accuracy error 1.11×10^{-16} and maximum relative norm error 2.36×10^{-7}.

Median test-accuracy damage, in percentage points, with the count of runs above ten points in brackets:

Table 13: **The dependence is present when the model first generalizes.** Medians over 16 runs; brackets count runs with damage above ten percentage points. Matched scaling is the operator-level control.

Audit point	Removal of R	Norm-preserving removal	Matched scaling
Stable-generalization confirmation	71.55 (14/16)	70.07 (14/16)	0.034 (0/16)
Update 50,000	88.56 (15/16)	88.97 (15/16)	0.00 (1/16)

By task at the confirmation point, ordinary removal reduces accuracy by 71.55 points in addition (8/8 above ten) and 69.78 in multiplication (6/8); norm-preserving removal reduces it by 70.07 (8/8) and 70.60 (6/8). At update 50,000 the figures are 88.56 (8/8) and 86.83 (7/8) for ordinary removal, and 88.97 (8/8) and 87.36 (7/8) with norm preserved. The scaling control's median is near zero but one run exceeds ten points at 50,000, so the median must not be reported as no damage in every run. The confirmation point is the fifth consecutive qualifying evaluation and therefore sits 400 updates after the first.

Effects on continued training time, paired by operator. All 48 continuation branches completed and verified. Norm-preserving removal was earlier than normal training in 16/16 runs at a median ratio of total updates of 0.481, earlier than the matched scaling of the same set in 16/16 at 0.463. For each removal operator, the sustained branch preceded the one attained evaluation-only curve and attained in the other 15 runs, whose evaluation-only curves were censored. Full-budget mean cross-entropy was lower in 16/16 for each of these comparisons. Matched scaling was earlier than normal training in only 4 of 16 runs, later in 11 and tied in one, at a median ratio of total updates of 1.027.

Limits. These are the first panel's runs, so the extension does not add independent training runs; the fresh-seed test is Appendix B.1. The audit points and operators are properties of the same runs and are not additional seeds. Acute set ablations are not energy-matched across operators, and norm restoration is defined separately for each branch. First stable attainment did not guarantee later persistence: the norm-preserving branch for addition seed 910205 attained at update 2,700 but ended at 0.9822 test accuracy. This case motivated the secondary persistence measurements in the fresh-seed panel; it remains in the original first-attainment comparison.

B.11 Evaluation-Only Versus Sustained Frequency Removal

This supplemental experiment completes the evaluation-only control for the arm removing frequencies below the median source energy. Its protocol was frozen before the new evaluation-only outcomes, after the normal trajectories, trajectories with sustained removal and their comparison at the source were known. It reuses the full 24-setting panel: add/sub/mul/div, ReLU/GeLU/SiLU, and seeds 900003–900004. It is not a fresh-seed confirmation or an independently chosen timing window.

Matched trajectories. For each source, fix the same below-median real-character set S used by the archived sustained-deletion branch. A is normal continued training and normal evaluation. B applies S only when evaluating those exact A weights. C is the archived branch trained and evaluated with S continuously removed. A and C share the source checkpoint, AdamW reset, float32 continuation, a budget of 25k further updates, and 100-update evaluation grid. Stable generalization requires five consecutive evaluations at accuracy at least 0.99; onset is the first of these, confirmation the fifth. Unattained times remain null; no censor value enters a ratio or median of attained times. No branch is selected by its new B outcome.

B's temporary hook is exception-safe and preserves model parameters, buffers, existing gradients, module modes, and Python/NumPy/CPU/CUDA random states. The replay matches the archived A train/test accuracy and CE values with maximum observed error zero at all $24 \times 251 = 6{,}024$ evaluations. This is metric parity, not a claim of full-trajectory bitwise weight certification. Selection matches C's fixed set, and B(0) matches C(0) in every setting. An independent check that imports no runner code rechecks the inputs, grids, source identities, parity, stable times, censoring and fixed-budget mean CE; the protocol, per-cell results and full trajectories are in the artifact.

Outcomes and denominators. A and C attain in 24/24; B attains in 21/24 (Figure 4). C is earlier in all 21 attained B/C pairs, with median onset gap 2,000 updates and median C/B ratio 0.292. The other three B branches remain censored, while C attains. Among jointly attained A/B pairs, B is earlier in 2, tied on the grid in 11, and later in 8; their median B/A ratio is 1.000. Similar medians do not establish equivalence. All 24 C branches have lower full-budget mean test CE than B; the across-cell medians for A/B/C are 0.968/0.802/0.082. B improves that loss summary over A in 21/24. The descriptive C/A onset ratio remains 0.300 over all 24 pairs.

Seed 900003 has 10/12 B attainments and 12/12 C attainments, with C earlier in all 10 comparable pairs (median C/B 0.286; gap 2,600). Seed 900004 has 11/12 and 12/12, with C earlier in all 11 pairs (0.318; gap 1,800). These are two seed panels across twelve task–activation combinations, not 24 independent seed repetitions. The three B-censored cells are ReLU/add at both seeds and ReLU/mul at seed 900003; their final accuracies are 0.297, 0.489, and 0.196.

Interpretation. The fixed evaluation mask improves some normal-trajectory outputs, but does not reproduce sustained removal's generalization benefit, so changing training matters in this comparison. No particular mechanism is identified, and the onset difference between B and C does not define a

mediated fraction of the benefit. Source selection is retrospective and AdamW is reset; the per-cell onsets and full-budget curves for all 24 settings are in the artifact.

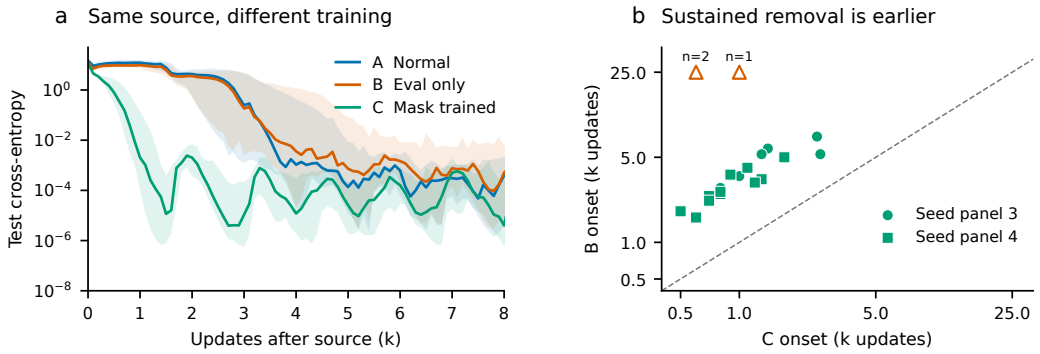

Figure 4: Historical evaluation-only comparison. A trains and evaluates normally; B applies the low-energy mask fixed at the source only to those same A weights at evaluation; C maintains it during training. (a) Median and across-cell IQR of test CE over the first 8k of 25k remaining updates; values below 10^{-8} are clipped for log display only. (b) C is earlier in all 21 attained pairs. Triangles mark the 25k bound for three B censors, two of which overlap. The 24 settings span two seed panels.

B.12 Endpoint Damage and Effect on Generalization Time for the Same Sets: A Post-Hoc Analysis

The three dissociations, in brief. Endpoint damage does not decide the sign of the effect: two sets far apart in endpoint damage (88.6 against 0.034 accuracy points) generalize at nearly the same time, and the half relied on most is still earlier in every run, as are all 640 uniform halves on 32 fresh runs (this appendix and Appendices B.50 and B.54). A set's effect depends on what else is masked: the four frequencies whose ablation damages the endpoint most delay generalization by a median 2,300 updates when masked alone and by 200 once 24 others are masked too, a pattern 64 fresh addition runs repeat, although the prespecified count of 12 of 16 runs was missed with 10 (Appendices B.15, B.48 and B.54). Orientation matters at fixed rank: a Haar-random subspace of equal rank masked throughout never generalizes, and rotating the masked subspace toward Haar removes the gain between $11.3°$ and $22.5°$ (Appendices B.13 and B.14).

This analysis connects the discovery cohort's training outcomes to acute dependence on the same frequency sets. All of the discovery cohort's outcomes and saved single-frequency audits were already known. In particular, the low-energy example in multiplication seed 910214 was identified before this supplement. The supplemental protocol records that prior knowledge and fixes the full 16-run audit; freezing it does not make the analysis blind to those outcomes.

Cohort and replay. All 48 normal, low-energy and random-set endpoints of the discovery cohort are evaluated at the common budget of 50,000 updates. The source random set's joint damage on the normal endpoint, which the runner had not stored, was added by an evaluator that imports none of the training runner's helpers; an independent replay reproduced every recorded score exactly (2,832 comparisons) and the top-four damage ordering in every endpoint.

Comparisons for the same sets. For each source set, we report its joint accuracy and CE damage at the normal endpoint, overlap with that endpoint's top four, and the corresponding normal and constrained stable onsets (Table 14). Every mask is included. The normal endpoint's accuracy ranges from 0.9991 to 1.0000. Across the 16 random sets, the median accuracy damage is 0.8856 and median CE damage is 11.4004. Fifteen sets reduce accuracy by 0.4961–0.9688; the remaining set, multiplication seed 910211, reduces it by 0.00246. All 16 sustained random removals nevertheless attain earlier, at median ratio of total updates 0.4863. Fifteen random sets intersect the normal endpoint's top four, but membership alone is not the evidence of damage: the joint ablations measure it directly.

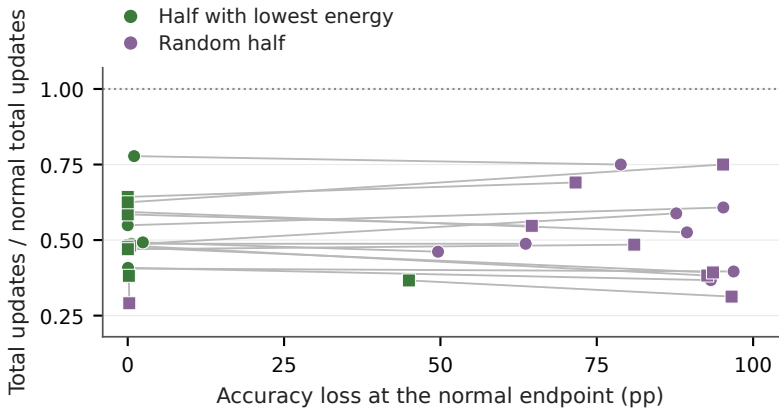

Figure 5: Discovery sets differ in endpoint damage but have similar effects on generalization time. Each line joins two sets from one of 16 discovery runs: the half with lowest energy at the source and a random half. These runs are separate from the confirmation in Figure 1. The horizontal axis measures accuracy loss when the set is removed from the normal model at update 50,000; the vertical axis is the ratio of total updates to onset under sustained removal versus normal training, including the shared prefix. Circles denote addition and squares multiplication. All 32 observations are shown; the dotted line marks a ratio of 1. This comparison was specified after the discovery outcomes were known.

The 16 low-energy sets have median accuracy damage 0.000336 and median CE damage 0.001580. Two intersect the normal endpoint's top four. Three cause more than one percentage point of accuracy damage, and one more than ten; these are descriptive thresholds, not acceptance gates of the discovery cohort. All 16 low-energy branches are earlier than normal. In the motivating multiplication example, joint removal raises CE from 0.002711 to 2.467315 and reduces accuracy from 0.999776 to 0.550173. Continued removal advances stable onset from 13,100 to 4,800 total updates (ratio 0.3664). The low-energy endpoint attains accuracy 0.999776.

What the adapted models depend on. With each trained mask still applied, the four frequencies whose single ablation raises CE most lie outside the imposed mask at all 32 adapted endpoints, and removing them jointly raises CE in every case; the per-endpoint profiles are in the artifact.

The acute endpoint measurements are not matched for norm or removed energy across sets. The random and low-energy policies match frequency count, not necessarily real rank; the one-dimensional Nyquist component accounts for the possible difference. Source and endpoint removed-energy fractions are retained in the artifact. Training times are reconstructed from all 481 saved continuation evaluations with the original stable-threshold rule, without new exclusions. Endpoint damage and the effect on generalization time are paired measurements, not additive components of a mechanism or a decomposition of optimization benefit. The audit is outcome-informed and warrants no new claim of selector superiority, frequency competition, or task-level impossibility.

B.13 Basis Control: An Equal-Rank Subspace Without Task Structure

This control asks whether an arbitrary mean-preserving constraint of the same rank reproduces the learning benefit of random frequency removal. It is a planned extension of the completed discovery cohort, not an independent new-seed confirmation: all of the discovery cohort's training outcomes and the endpoint damage audit were known when the protocol was frozen, and no subspace outcome existed at that point. The protocol imposes no new significance gate, no success threshold and no target ratio, and its interpretation rules were fixed in advance for either outcome.

Table 14: **Endpoint damage and effects on generalization time for every fixed set.** L/R denote source low-energy/random sets. "Acute acc." evaluates the normal 50k endpoint after joint removal; ΔCE is its increase from that endpoint's native CE. "Overlap" counts normal top-four frequencies in the mask. A and I are normal and intervened absolute stable-onset updates; ratio is I/A, including the common 2,000-update prefix. No pairs are excluded.

Task	Seed	Set	Acute acc.	ΔCE	Overlap	A	I	Ratio
add	910201	L	1.000	0.0001	0	5100	2800	0.549
add	910201	R	0.048	6.0432	3	5100	3100	0.608
add	910202	L	1.000	0.0005	0	10100	4100	0.406
add	910202	R	0.031	18.8572	3	10100	4000	0.396
add	910203	L	0.998	0.0085	0	6800	3300	0.485
add	910203	R	0.123	15.2222	2	6800	4000	0.588
add	910204	L	0.994	0.0209	0	8200	4000	0.488
add	910204	R	0.364	7.6389	1	8200	4000	0.488
add	910205	L	0.990	0.0494	1	3600	2800	0.778
add	910205	R	0.212	24.0423	1	3600	2700	0.750
add	910206	L	0.999	0.0009	0	9800	4000	0.408
add	910206	R	0.067	18.2716	2	9800	3600	0.367
add	910207	L	1.000	7.03×10^{-6}	0	5900	3500	0.593
add	910207	R	0.106	11.0560	2	5900	3100	0.525
add	910208	L	0.976	0.0692	0	6500	3200	0.492
add	910208	R	0.504	3.2020	1	6500	3000	0.462
mul	910211	L	0.998	0.0090	0	11000	4200	0.382
mul	910211	R	0.998	0.0099	0	11000	3200	0.291
mul	910212	L	1.000	6.75×10^{-5}	0	5300	3100	0.585
mul	910212	R	0.354	5.2420	1	5300	2900	0.547
mul	910213	L	0.996	0.0150	0	8100	3900	0.481
mul	910213	R	0.072	12.2247	3	8100	3100	0.383
mul	910214	L	0.550	2.4646	1	13100	4800	0.366
mul	910214	R	0.034	13.6693	3	13100	4100	0.313
mul	910215	L	1.000	3.29×10^{-8}	0	4200	2700	0.643
mul	910215	R	0.284	11.7447	1	4200	2900	0.690
mul	910216	L	1.000	2.50×10^{-5}	0	8400	4000	0.476
mul	910216	R	0.064	4.3936	3	8400	3300	0.393
mul	910217	L	1.000	0.0023	0	6600	3100	0.470
mul	910217	R	0.190	10.5225	2	6600	3200	0.485
mul	910218	L	1.000	1.13×10^{-6}	0	4800	3000	0.625
mul	910218	R	0.048	14.9772	3	4800	3600	0.750

Design. One arm was added to each of the discovery cohort's 16 sources at exactly 2,000 completed updates, retaining source weights, AdamW moments, counters and all random states, at the discovery cohort's unchanged budget of 50,000 total updates. For each run the real rank r equals that run's random-half rank, 55 or 56. A Haar-random r-dimensional subspace orthogonal to the constant direction is drawn from a generator seeded deterministically by the run identifier and orthogonalized by QR in float64, one draw per run, with no search, rejection or redraw. Writing $X = E - \mu$, every forward pass uses $\mu + X - XQ$ with $Q = UU^\top$ fixed and no gradient through its construction. Rank, operand axis, mean preservation, source, gradient operator form, optimizer handling and budget match the frequency arms. Removal energy, frequency sparsity, endpoint damage, task representability and GPU load do not.

Outcome. All 16 branches ran the full budget with no failures (Table 15).

Addition and multiplication each contributed 0/8. Final test accuracy had median 0.132 and range 0.059–0.979; no run reached 0.99 and one exceeded 0.50. Final train accuracy had median 1.000 and range 0.881–1.000, so the branches continued to fit the training set.

Checks. At the source the Haar subspace carries a median 50.1% of the operand embedding's energy (48.5–52.3%), against 50.21% for the character half removed by the random-half arm, a

Table 15: **Same rank, different removed subspaces.** Sixteen discovery cohort runs, eight ReLU addition and eight ReLU multiplication, from the same sources at the same budget. CE entries are medians of per-run budget-averaged test CE; brackets give the range over runs.

	Haar subspace	Characters (random half)	Normal
Stable generalization	0/16	16/16	16/16
Median onset (total updates)	not attained	3,200	6,700
Mean test CE over budget	14.776	0.082	1.013
	[1.613, 17.986]	[0.020, 0.181]	[0.087, 3.972]

similarity of distributions and not a per-run match; at update 50,000 only 2.3% remains inside it, and the operand tokens do not collapse onto one another.

Scope. The comparison changes only the orientation of the removed subspace, but it confounds representability with optimization difficulty: a random subspace of this rank need not contain any complete character plane and may exclude the solution outright. Non-attainment within budget is not unlearnability. A brief version of the same mask, unmasked after 500 updates, is tested in Appendix B.19, where both arms generalize.

B.14 Rotating the Removed Subspace Out of the Frequency Basis

The basis control of Appendix B.13 compares two subspaces of the same rank. This extension asks where, between them, the outcome turns over. It is a planned extension of completed panels: all the discovery cohort's outcomes and all basis-control outcomes were known when its protocol was frozen, and it is not a fresh-seed confirmation. It sets no new significance gate and no target ratio.

Construction. For each run let C be the orthonormal basis of the real character directions of that run's recorded random-frequency set, built by the same routine the discovery cohort uses, and let H be that run's frozen Haar basis, read rather than redrawn. With Θ the principal angles between the two subspaces, obtained from $M = (H - C\,C^\top H)(C^\top H)^{-1}$ and $\Theta = \arctan \Sigma$ for $M = Q\Sigma R^\top$, the geodesic point is

$$U(t) = \mathrm{orth}(CR\cos(t\Theta) + Q\sin(t\Theta))\,, \qquad P(t) = U(t)U(t)^\top.$$

Every principal angle is swept by the same fraction, so the mean angle is linear in t by construction. This matters: the obvious alternative of mixing the basis vectors and re-orthogonalising is badly non-uniform, placing $t = 0.75$ within 0.01 of the Haar endpoint in mean cosine.

$t = 0$ and $t = 1$ are the recorded arms and were not retrained. Before any new branch, $\|P(0) - P_{\mathrm{char}}\|_F$ and $\|P(1) - P_{\mathrm{haar}}\|_F$ were required below 10^{-10}; the realized values were at most 6.6×10^{-15} and 8.5×10^{-14} over the 16 runs. New arms are $t \in \{0.25, 0.50, 0.75\}$, 48 branches, from the same sources at the same 50,000-update budget with optimizer state retained.

Outcome. All 48 branches completed (Table 16).

Table 16: **Stable 99% attainment along the subspace rotation.** Sixteen discovery cohort runs, eight ReLU addition and eight ReLU multiplication. The angle is the median across runs of the mean principal angle to the character subspace. Ratios are medians over attained pairs against normal training. "Earlier" counts attained branches preceding normal training; unattained branches have no observed ratio. CE is the median of per-run budget-averaged test CE.

t	Mean angle	Attained	Earlier	Total ratio	Budget CE
0.00	0.0°	16/16	16/16	0.486	0.082
0.25	11.3°	16/16	15/16	0.591	0.137
0.50	22.5°	0/16	0/16	—	0.913
0.75	33.8°	0/16	0/16	—	6.097
1.00	45.0°	0/16	0/16	—	14.776

Addition and multiplication split identically at every level, 8/8 and 8/8 at the first two and 0/8 and 0/8 at the last three. The prespecified classification distinguished a threshold, a graded decline, an immediate collapse at the first intermediate level, and anything else; the observed curve matches the prespecified threshold pattern for stable 99% attainment within the 50,000-update budget. The two adjacent sampled levels have median mean angles of 11.3° and 22.5°; they do not locate an exact critical angle. Budget-averaged test CE increases across all five sampled levels in each run. This is a sampled trend, not a test of continuity. At $t = 0.25$, addition seed 910205 is later than normal training (3,900 versus 3,600 total updates); the other 15 runs are earlier.

Post-hoc threshold and persistence checks. We rescored the saved grids at accuracy thresholds 0.90, 0.95 and 0.98 (Table 17), retaining the same five-evaluation window, 100-update spacing and budget. These descriptive checks were added after the outcomes were known and do not replace the prespecified 0.99 criterion or its classification. At $t = 0.50$, all runs attain 90%, ten attain 95% and two attain 98%, while none attains 99%. Final test accuracy there has median 94.97% and range 91.99–98.75%; all final training accuracies are 100%.

Table 17: **Threshold sensitivity, scored after outcomes were known.** Each attainment count uses five consecutive qualifying evaluations. Final accuracy is the across-run median at update 50,000, in percent. Angle is the median of per-run mean principal angles in degrees.

t	Angle	$\geq 90\%$	$\geq 95\%$	$\geq 98\%$	$\geq 99\%$	Final acc.
0.00	0.0	16/16	16/16	16/16	16/16	100.00
0.25	11.3	16/16	16/16	16/16	16/16	99.51
0.50	22.5	16/16	10/16	2/16	0/16	94.97
0.75	33.8	8/16	3/16	1/16	0/16	84.15
1.00	45.0	1/16	1/16	1/16	0/16	13.18

First attainment also differs from persistence. At $t = 0.25$, all 16 runs attain stable 99% accuracy, but only 10/16 stay at or above 99% for all of their last ten evaluations. Multiplication seed 910215 ends at 98.77%; the other 15 end above 99%. The corresponding last-ten count is 16/16 at $t = 0$ and 0/16 at each of the three larger rotations. Per-run threshold onsets are retained in the artifact.

Scope. Rank, operand axis, mean preservation, source, optimizer handling and budget are fixed across all five levels while the removed subspace rotates relative to the task characters. This changes the projection operator, not merely its coordinate description, and tests orientation at fixed rank. Removed energy is not matched across levels. It does not separate a subspace that is harder to learn under from one that is harder to represent in, and it does not show that the coordinates supplied by the task are necessary. The observed loss of 99% attainment is conditional on these models, this task family, this trigger and this budget. Its angle bracket describes the sampled geodesic paths, not a universal boundary.

A representability reading. Write V for the centered operand functions and Π_w for the plane of frequency w. The character mask $Q(0) = \sum_{w \in S} P_w$ leaves 28 planes untouched. Along the geodesic, $Q(t) = \sum_i u_i u_i^\top$ with $u_i = c_i \cos t\theta_i + q_i \sin t\theta_i$, and since the q_i span $C^\perp$, every unit v in an originally clean plane has $v^\top Q(t) v = \sum_i \sin^2(t\theta_i)\langle q_i, v\rangle^2$, a convex combination of the $\sin^2(t\theta_i)$: for $t > 0$ no plane is clean, and the leakage $\lambda_w(t) = \frac{1}{2}\operatorname{tr}(P_w Q(t))$ of a clean plane is at least $\sin^2(t\theta_{\min})$. With principal angles uniform on $[0, \pi/2]$, as a Haar subspace of matching rank gives, the mean leakage of an originally clean plane is $\frac{1}{2} - \sin(2A)/(4A)$ with $A = t\pi/2$: 0.050, 0.182 and 0.350 at $t = 0.25$, 0.5 and 0.75. On the recorded bases the measured medians are 0.050, 0.183 and 0.353, and the fifth-least-leaky plane leaks 0.045, 0.163 and 0.317. Across the 48 rotated branches, final accuracy falls with that leakage (Spearman -0.82). Two things the reading does not give: within a rotation, leakage does not order the runs (Spearman 0.14 at $t = 0.5$ and -0.20 at 0.75), and the five frequencies the rotated branches finally use are not the least leaky (their leakage ranks among the 56 planes have medians of 5 to 16), though they do lie outside S in 239 of 240 cases. The decline across rotations is what leakage predicts; which runs and which planes succeed at a given rotation is not.

B.15 Nested Removal: Holding the Relied-Upon Set Fixed

A set that damages the normal model can be masked throughout training and generalization arrives earlier. That leaves a decomposition open: the removed half contains a few frequencies the normal solution depends on and many it does not, so the net benefit could be a delay from the first part outweighed by a benefit from the second. This extension tests that reading with three constructed sets on the same sources. It does not partition the half-sets themselves.

Planned extension with all earlier outcomes known; not a fresh-seed confirmation, no new significance gate and no target ratio.

The two sets. K is the four frequencies with the largest single-frequency cross-entropy damage at that run's normal 50,000-update endpoint, read verbatim from the frozen endpoint damage audit rather than recomputed. **It is defined using the endpoint**, so every arm containing it is diagnostic and is not a selection rule that could be applied before the task is solved. N is 24 frequencies drawn uniformly without replacement from the 52 not in K, using a generator seeded deterministically by the run identifier, one draw per run with no search, rejection or redraw, and no selection on energy, damage or any outcome. N is not asserted to be unimportant; its effect is measured. $|K \cup N| = 28$ matches the size of the recorded half-set arms.

Design. Three new branches per run on the 16 discovery cohort sources, 48 in total, from the same sources at the same 50,000-update budget with AdamW state retained. The normal branch is the recorded one and was not retrained. Because the character directions of disjoint frequency sets are orthogonal, the projectors must satisfy $P(K \cup N) = P(K) + P(N)$; this was required below 10^{-10} and realized at about 3×10^{-14} on every run, as an independent check on the construction. Each run also replays its recorded random-frequency branch for 100 updates and requires exact agreement with the archived scores.

Outcome. All 48 branches completed; an independent check rebuilds K and N from the frozen audit and the seeding rule and rechecks additivity with a character basis it constructs itself.

Table 18: **Removing the relied-upon four alone, others alone, and together.** Median ratios of total updates against each run's recorded normal branch, over the discovery cohort's 16 runs. "Earlier" counts runs generalizing before normal training. All arms attained in 16/16.

Removed set	Size	Median total ratio	Earlier	Budget CE
K, the endpoint's top four by damage	4	1.285	3/16	1.847
N, 24 others drawn once	24	0.471	16/16	0.083
$K \cup N$	28	0.544	16/16	0.125

The prespecified comparison. $K \cup N$ against N: later in 10 of 16, earlier in 4, tied in 2, median gap $+200$ updates (interquartile range $[-50, +600]$), the same split in each task. The prespecified rule required at least 12 of 16 later to record a measurable delay, so it classifies the delay from adding K as absorbed; a count places no bound on size, and the paired interval of Appendix B.48, $+300$ $[+50, +500]$, indicates a small delay.

Reading. Removing the four frequencies the endpoint depends on most delays generalization on its own (1.285, earlier in only 3 of 16), in the direction of the earlier panels' core arm (Appendix B.8); masking 24 others brings generalization well forward, and masking their union still brings it forward. The net benefit of a large set is therefore not evidence that removing relied-upon content helps. The three sets are constructed, not a decomposition of the half-sets used elsewhere.

The effect of the relied-upon four depends on what else is removed. Per run, write $c(K) = T(K) - T(\text{normal})$ for the change in generalization time from removing K on its own and $c(K \mid N) = T(K \cup N) - T(N)$ for the change from adding it once N is already masked. Over the 16 runs $c(K)$ has median $+2{,}300$ updates, interquartile range $[+700, +3{,}700]$, and is positive in 13 of 16, whereas $c(K \mid N)$ has median $+200$, interquartile range $[-50, +600]$, positive in 10 of 16.

Their per-run difference, the interaction $c(K \mid N) - c(K)$, has median $-2{,}050$, interquartile range $[-3{,}200, -400]$, and is negative in 13 of 16.

Removing the four frequencies the endpoint depends on most therefore delays generalization about an order of magnitude less when 24 other frequencies have already been masked than when they have not. This is an interaction between two set interventions, computed post-hoc from the recorded onsets of one cohort. It does not name a mechanism, and in particular it is not evidence for frequency competition as opposed to any other account of why the background matters.

Does the earlier branch still use the relied-upon four? One reading of the N arm is that masking 24 frequencies sends training to a different solution that no longer rests on K. At each arm's update-50,000 endpoint we remove K on top of whatever that arm already had masked, and rank every frequency it can still use by single-frequency damage, hook-free on a copy of the parameter.

The N branch has not moved off K. Removing K from its endpoint reduces accuracy by a median 0.966, 95% bootstrap interval over runs $[0.952, 0.971]$, against 0.975 $[0.970, 0.986]$ at the normal endpoint, and its own four most damaging frequencies fall inside K in a median 4 of 4, at least 3 of 4 in 16 of 16 runs. The same pass recovers K exactly at all 16 normal endpoints.

The reliance scale separates the two fast arms. Reading the same endpoints one frequency at a time gives the picture in Table 19. The quantity is summed single-frequency cross-entropy damage over C, the 28 frequencies outside $K \cup N$ that every arm retains, so it is comparable across arms in support if not in level.

Table 19: **Where each endpoint's measurable reliance sits.** Summed single-frequency cross-entropy damage over the 28 frequencies available to every arm, medians over 16 runs with 95% bootstrap intervals. Update ratio is the median ratio of total updates from Table 18. Every endpoint listed is at or above 0.99 test accuracy.

Endpoint	Update ratio	Reliance on C	K available?
normal	1.000	0.001 $[0.000, 0.008]$	yes
N removed	0.471	0.079 $[0.016, 2.034]$	yes
K removed	1.285	9.242 $[4.294, 14.492]$	no
$K \cup N$ removed	0.544	20.052 $[12.445, 28.192]$	no

The two arms that keep access to K barely touch C; the two that cannot use K rely on it heavily, which for them is forced by construction. What is not forced is which side the early arms fall on. N, the earliest arm at 0.471, could have relocated onto C and did not. $K \cup N$, nearly as early at 0.544, sits at the far end of the same scale, so displacement and timing do not track each other: the K arm, displaced to 9.242 from normal's 0.001, is the *latest* in the design, at 1.285 and earlier than normal in only 3 of 16, while the arm displaced furthest of all is among the earliest.

Relocating what the solution relies on is therefore neither necessary for earlier generalization, since N is earliest without relocating, nor sufficient for it, since removing K relocates reliance and delays generalization. This does not name the mechanism. It rules out one family of explanations for these arms: any account in which masking brings generalization forward *by* moving the solution off the frequencies normal training depends on. Appendix B.52 finds the converse bound with a different measure: an arm forced off the normal endpoint's key frequencies is still earlier.

The spread is not a spread in solution quality. All four endpoints generalize: final test accuracy is at least 0.9905 on every branch of every arm, and the three trained here finish at test cross-entropy below 3.3×10^{-2}.

Scope. K is endpoint-defined and N is one draw per run, so the contrast is between two particular sets, not two selection procedures; the reliance measurements are post-hoc, one frequency at a time, and the levels in Table 19 are read as a separation of orders of magnitude, not as calibrated quantities.

Does N depend on K when it generalizes, or only by 50k? The audit above reads the 50,000-update endpoint, which licenses a claim about where the arm ends and not about how it got there: N could have leaned on other frequencies while it generalized and returned to K afterwards, leaving that

audit unchanged and the reading wrong. The runners saved each branch's first stable-generalization checkpoint, so the same measurement is available there. At N's *own* first generalization, removing K reduces accuracy by a median 0.941, 95% interval $[0.906, 0.961]$, against 0.955 $[0.930, 0.972]$ at the normal branch's own first generalization; N's four strongest frequencies overlap K in a median 4 of 4 and in at least 3 of 4 in 16/16. From its own onset to update 50,000 the dependence deepens by a median $+0.023$ accuracy, larger at 50k in 14 of 16. The reliance is therefore present when the arm arrives and is not acquired afterwards, and the temporary-migration reading is excluded. Post-hoc on saved checkpoints, measured hook-free on a copy of the parameter.

B.16 Deleting the Content Versus Preventing Its Return

Sustained removal does two things at once: it deletes the content present at the fork, and it prevents those directions from carrying content again. This extension separates them on the set that brings generalization forward.

Planned extension with all earlier outcomes known; not a fresh-seed confirmation, and no new significance gate.

Four arms, two already recorded. Table 20 lists them. With S the run's recorded random-frequency set, Q its character projector and $X = E - \mathrm{mean}(E)$ on the operand columns:

Table 20: **What each arm changes.** Only the two middle rows are new.

Arm	Source content in S	May S change after the fork	Operator
normal	kept	yes	none
one-shot	deleted once	**yes**	parameter overwritten at the fork
freeze	**kept, at source values**	no	$\mathrm{mean}(E) + (X - XQ) + X_s Q$
sustained	deleted	no	$\mathrm{mean}(E) + X - XQ$

The one-shot arm changes the parameter, not the forward pass. Removing a hook part-way would add back content that had been masked all along and jump the logits, confounding "may regrow" with "old content restored". Writing the effective embedding into the parameter and training without a hook keeps the logits continuous at the fork by construction; the realized write-back gap was below 10^{-6} on every run. Each run also replays its recorded sustained branch for 100 updates and requires exact agreement with the archived scores.

Outcome. All 32 new branches completed (Table 21; Figure 6).

Table 21: **The three arms compared.** Median ratios of total updates against each run's recorded normal branch, over the discovery cohort's 16 runs. The recovered fraction is the summary statistic $f = (1 - r_{\mathrm{arm}})/(1 - r_{\mathrm{sustained}})$ formed from those medians, with the threshold $f \geq 0.5$ fixed before the arms ran. It compares arms; it does not apportion the benefit between mechanisms.

Arm	Addition	Multiplication	Pooled	Recovered fraction
sustained	0.507 (8/8)	0.439 (8/8)	0.486 (16/16)	—
one-shot	0.737 (7/8)	0.737 (8/8)	0.737 (15/16)	$+0.512$
freeze	1.397 (0/8)	1.551 (0/8)	1.479 (0/16)	-0.932

Counts in brackets are runs generalizing earlier than normal training. All arms attained within budget in 16/16. Across the 16 runs the pooled ratios have interquartile range [0.388, 0.598] for sustained removal, [0.698, 0.811] for one-shot and [1.275, 1.694] for freeze, with ranges [0.291, 0.750], [0.527, 1.020] and [1.127, 2.130]. The one run in which one-shot removal was not earlier than normal sits at 1.020; no freeze run falls below 1.127.

Energy reappears in the removed subspace. In the one-shot arm the energy inside S is zero at the fork by construction and returns to a median 0.22 of the operand-embedding energy by update 10,000, with a wide spread across runs (0.02 to 0.44; 0.18 and 0.27 by task). It then stays flat, at medians 0.21, 0.22 and 0.22 at updates 20,000, 30,000 and 50,000. Energy returns to those coordinates while

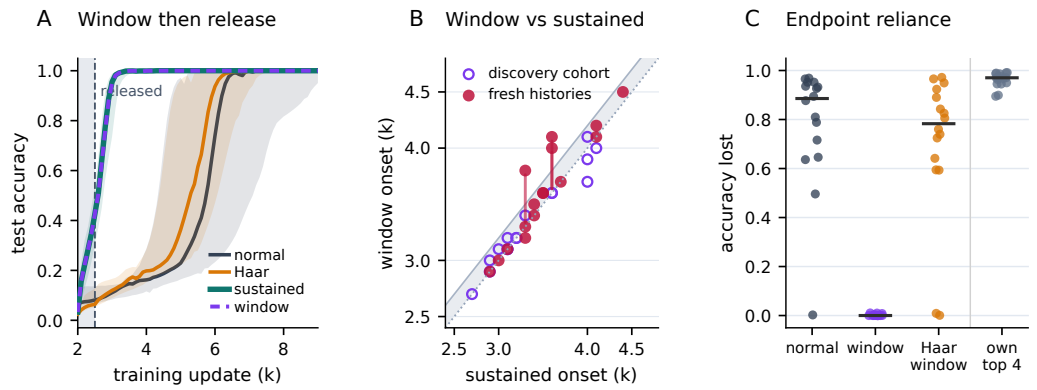

Figure 6: A bounded window, and what follows it. (A) Median test accuracy with interquartile band over the first panel's 16 runs; the strip marks the 500-update window, the dashed line where the set is unmasked. The window arm (dashed) lies on the sustained arm (solid) throughout, the same window on an equal-rank Haar subspace does not. (B) Onset for the window against sustained removal, one point per run, the window confirmation's 16 runs shown separately since no ratio is taken across cohorts; the three more than 200 updates above the diagonal are those missing the prespecified criterion. (C) Accuracy lost when the *same* character set is removed at update 50,000, with each character-window model's own four strongest frequencies as a positive control; bars are medians.

about half the benefit in updates stays with it. Whether the earlier content or computation is restored is not tested here.

Reading, with the margin stated. The prespecified classification is that deleting the content present at the fork suffices ($f \geq 0.5$), recorded here as the rule returned it. What the arms support without it is that sustaining the mask is not required for earlier generalization: one-shot deletion brings it forward in 15 of 16 runs. Two qualifications belong with that classification.

First, $f = 0.512$ clears the prespecified threshold of 0.5 by 0.012. By task it is $+0.534$ in addition and $+0.469$ in multiplication, so multiplication alone falls below the bar. The prespecified rule scores pooled medians and the classification follows it, but the accurate statement is that one-shot deletion keeps about half of the benefit in updates, not most of it. f is a ratio of two median ratios and is a descriptive summary of the arms, so it does not measure how much of the effect any mechanism contributes.

Second, the freeze result carries no such ambiguity. Retaining the source content in those directions while preventing it from changing delays generalization at 1.479 and is earlier than normal in 0 of 16, in both tasks. The advantage sustained removal holds over one-shot removal is therefore not that the constraint keeps acting; it is that sustained removal also keeps the content out.

Scope. This compares two operations on one recorded set per run and does not apportion the benefit between them. It does not identify why deleting that content helps, and it does not separate optimization from representability. The window arms, unmasked after a fixed number of updates, that bound how long the mask must act are in Appendix B.17.

B.17 How Long the Mask Has to Act

Planned extension on the 16 recorded runs of the discovery cohort, with all earlier outcomes known. Not a fresh-seed confirmation and no new significance gate. The protocol, its window rule and its reading were frozen before any cell ran.

Table 22: **Deleting the content, keeping it, and keeping it out.** Median total updates to generalization relative to normal training on the discovery cohort's 16 runs; other parameters remain trainable in every branch.

Intervention	Content at the fork	Afterwards	Ratio
Single deletion	Removed	None	0.737
Freeze	Retained	Fixed at fork values	1.479
Early window	Removed	Masked for 500 updates	0.481
Sustained masking	Removed	Masked throughout	0.486

The two arms. Both use the ordinary removal operator and each run's own frozen set S, so every comparison here stays inside one operator and one set. The early window masks S on $[2{,}000, 2{,}500)$ and then unmasks it; the late window trains normally to 2,500, masks S on $[2{,}500, 3{,}000)$, and then unmasks it. The normal, one-shot and sustained references are the recorded branches and were not retrained.

Why 500 updates. Fixed from the recorded sustained branches before this protocol was written, not from any outcome of these arms. Those branches return to 0.99 training accuracy by update 2,100–2,200, so a window shorter than about 200 updates would end before the branch had refitted and would be a one-shot deletion by another name. Their earliest stable-generalization onset over the 16 runs is update 2,700 and the median is 3,200. Five hundred updates is the longest round length that still unmasks before any run has generalized under the mask while lasting more than twice the refit time. No other length was run on this cohort.

Unmasking. At the end of its window a branch writes its current effective embedding into the parameter and then removes the forward hook. Removing the hook alone would restore content masked all along and jump the logits, confounding "the constraint stopped" with "old content came back". The realized write-back gap was 0.0 on every run and the energy inside S was zero at unmasking by construction. AdamW state is carried through unchanged.

Across these 16 runs, the early window's median total-update ratio is 0.481, with a 95% bootstrap interval $[0.405, 0.547]$, against 0.486 $[0.393, 0.588]$ for sustained removal. These intervals summarize each arm; the paired window-minus-sustained difference is reported separately below.

Table 23: **A bounded window reproduces what constraining the whole run gives.** Medians over 16 runs. Ratios are total updates against each run's recorded normal branch; paired differences are per run, in updates, negative meaning the first arm is earlier.

Arm	Median ratio	Earlier than normal	Recovered fraction
sustained, whole run	0.486	16/16	—
500-update window, then unmasked	0.481	16/16	1.011
window opened 500 later	0.527	16/16	0.921
one-shot deletion	0.737	15/16	0.512

How dependence was measured. The training-time audit was unusable because the ablation hook was not attached; every dependence number here comes from an audit of the saved checkpoints, and a hook-free route that writes the projection into a copy of the parameter agrees with it to 0.0 in every run. As a live-probe control, removing each window model's own four strongest frequencies reduces its accuracy by a median 0.970.

The prespecified comparisons. Table 23 gives the medians. Against sustained masking the early window's paired difference has median 0 updates, interquartile range $[0, +100]$, earlier in 3 runs, later in 5 and tied in 8. The prespecified allowance is 200 updates, twice the recording resolution, and it is one-sided: the prespecified condition is that the window is at most 200 updates later, not that the two differ by less than 200 in either direction. Meeting it is a statement about resolution, not a demonstration of equality. Against one-shot deletion the early window is earlier by a median 1,600

updates, in 14/16. The late window is earlier than one-shot by 1,300 in 14/16 and later than sustained by 400 in 15/16.

What the two windows do and do not show about timing. The prespecified classification is that the early window suffices: by total updates it is 350 ahead of the late window in 16 of 16. Measured from each arm's own unmasking, the early window is instead 150 updates later ($[+100, +300]$, 14 of 16); the two statistics are algebraically tied, so the pair supports no privileged early period, and both windows sit early in a 50,000-update budget.

What the unmasked models use. Energy inside S returns to a median 0.030 of the operand embedding's by update 10,000 and is 0.027 at update 50,000 for the early arm, 0.031 for the late arm. One-shot deletion returns to 0.22 on the same runs. Energy recovery and functional recovery are separate events, and here they separate: acutely removing all of S at update 50,000 changes CE by $+0.0004$ and accuracy by $+0.0001$ in the early arm, and by $+0.0002$ and $+0.0002$ in the late arm, while the strongest single frequency outside S carries a median 10.4 CE of damage. Over 16×4 endpoint top-four damage slots, 0 fall inside S for the early arm and 1 for the late arm. The same set removed from the *normal* endpoint reduces its accuracy by 88.6 points. A bounded early constraint is therefore followed, after 47,500 updates of unconstrained training, by an endpoint with very low measured sensitivity to the set it had masked. Low joint-ablation sensitivity is not the same as showing the set unused in every output, and nothing here establishes that the changed dependence is what produces the earlier arrival.

The dependence is already changed when the branch generalizes. Measuring only at update 50,000 leaves open that a branch leans on the masked set while it generalizes and sheds it later, which would make the changed endpoint dependence a late consequence rather than something present when the arrival time is decided. The runners saved each branch's first stable-generalization checkpoint, so the same quantity can be measured there. Ablating the masked set at the window branch's own first generalization reduces accuracy by a median 0.0023, at most 0.0086 in any of the 16 runs, against 0.0001 at update 50,000 for the same branches. The normal branches lose 0.7155 at their own first generalization and 0.8856 at update 50,000. The window models therefore already show very low measured dependence on the set at the moment they reach the criterion. This is a timing statement, not a mediation test.

Scope. One window length, one operator, one fork, one task family, on runs whose earlier outcomes were known. The arms show that the extra benefit sustained removal holds over one-shot deletion does not need the mask maintained, and that the unmasked model does not return to the set functionally. They do not identify why a few hundred constrained updates fix that, do not separate optimization from representability, and do not test a window placed later in training.

B.18 Fresh-Seed Confirmation of the Window Prediction, and What It Did Not Confirm

A confirmation, not an extension: 16 runs that had never been trained, with removal sets written to a manifest before initialization and the whole design frozen beforehand. No earlier experiment had used these seeds. All three arms, normal training, sustained removal and the early window, were trained here from one source per run, so nothing is borrowed across panels.

The prespecified event. Per run, with t_a the stable-generalization onset of arm a and D_a the accuracy lost when the masked set is ablated from arm a's update-50,000 endpoint, a run counts as a success when all three hold:

- **E1** $t_{\text{window}} < t_{\text{normal}}$;
- **E2** $t_{\text{window}} \leq t_{\text{sustained}} + 200$ updates;
- **E3** $D_{\text{window}} < 0.10$ and $D_{\text{normal}} > 0.10$.

The 200-update margin is twice the recording resolution and is the margin prespecified for the brief-window extension (Appendix B.17); the 0.10 thresholds are the damage threshold the earlier main

confirmation used. The gate asked for the event in at least 7 of 8 runs in each task, an intersection-union test whose p is the larger of the two one-sided binomial tails under a one-half null. Table 24 gives the outcome. All three conditions held in 16 of 16 development runs when the protocol was written.

Table 24: **What the fresh seeds reproduced.** Sixteen new runs, eight per task. Medians over runs; paired differences are per run, in updates.

	normal	sustained	500-update window
median ratio of total updates	1.000	0.411	0.416
earlier than normal	—	16/16	16/16
accuracy lost when the set is removed at 50,000	0.886	—	0.000
Prespecified conditions, over 16 runs			
E1, window earlier than normal			16/16
E2, window at most 200 updates later than sustained			13/16
E3, endpoint dependence flips			16/16
conjunction, by task		6/8 addition, 7/8 multiplication	

Outcome: two of three conditions replicate, the conjunction does not. E1 and E3 hold everywhere. The window arm is earlier than its own normal branch in 16 of 16, by a median 5,000 updates, and every window endpoint loses less than 0.10 accuracy when the masked set is ablated while every normal endpoint loses more. E2 holds in 13 of 16. The three failures are $+400$, $+500$ and $+500$ updates later than sustained removal, against a one-sided allowance of 200. The conjunction therefore reaches 6 of 8 in addition and 7 of 8 in multiplication, the intersection-union p is 0.145, and the gate is **not met**. E2 is one-sided and the gate counts runs, so failing it does not establish that the two arms differ by any particular amount; it is not an equivalence test on the time difference.

Reading this as the protocol required. The protocol stated before the runs that a failure of E2 alone would mean the bounded window does not fully substitute for the sustained constraint on new seeds, and that the outcome was to be reported without reframing. That is the outcome. On these runs the paired difference against sustained removal has median $+50$ updates, not the zero seen in development, and the window is later in 8 runs, earlier in 1 and tied in 7. The honest summary is that a bounded early window reproduces the earlier generalization and the changed endpoint dependence on seeds that had no part in designing it, and comes close to but does not match sustaining the constraint throughout.

Nothing here weakens the first finding. Sustained removal reaches 0.411 and is earlier than normal in 16 of 16 on this cohort as well, and the endpoint dependence of normal training, 0.886, replicates the value measured on the earlier panels.

Scope. One window length, one operator, one fork, one task family. The failure of E2 concerns the one-sided at-most-200-updates-later criterion, not the existence of the window effect. The design cannot say whether a longer window would close the gap, since only 500 updates were run and no length sweep is licensed by this protocol.

B.19 The Orientation Control for the Window

The first finding has an orientation control: at the same real rank, a Haar subspace of the operand embedding never reaches stable 99% accuracy within budget, and neither does a 22.5° rotation of the character subspace. The window finding had none, and a window damages the training fit about as much as the Haar subspace does, 0.85 against 0.89 at the window start, so a bounded perturbation of an arbitrary equal-rank subspace was a live alternative reading. No earlier arm answered it: the one-shot arm perturbs the character set, and the Haar arm is sustained. This experiment fills that one cell.

Frozen before any of its cells ran, and before the fresh-seed confirmation of Appendix B.18 had reported. Planned extension on the same 16 recorded runs, no new significance gate.

The same subspace, briefly. The arm masks *the run's own Haar subspace*, the one whose sustained version never generalized, on $[2{,}000, 2{,}500)$ and then unmasks it with the same write-back used elsewhere. The subspace is regenerated deterministically from the run identifier and is checked to equal, bit for bit, the projector the basis control recorded, on every run. A fresh draw would turn the comparison into "some other subspace, briefly" and is what that check exists to prevent.

Table 25: **A bounded window is not a generic perturbation.** Medians over the 16 runs of the discovery cohort. Paired differences are per run, in updates, positive meaning the Haar window is later.

Arm	Median ratio	Earlier than normal
character set, 500-update window	0.481	16/16
character set, sustained	0.486	16/16
Haar subspace, 500-update window	0.932	13/16
Haar subspace, sustained	censored	0/16
normal training	1.000	—
Haar window − character window: median $+3{,}050$, IQR $[+2{,}150, +3{,}950]$, later in 16/16		
Haar window − normal: median -450, IQR $[-1{,}950, -150]$, earlier in 13/16		

Outcome. The prespecified classification is that orientation matters (Table 25). All 16 branches attain, so a brief Haar window does not block generalization the way the sustained Haar constraint does, but the arrival advantage almost vanishes: a median ratio of 0.932 against 0.481 for the character window, later than the character window in 16 of 16 by a median 3,050 updates, and only 450 updates earlier than normal training. The prespecified thresholds partitioned the interval between 0.481 and 1.000 before the arm ran, with ≤ 0.60 reading as orientation-free and ≥ 0.85 as orientation-dependent; the observed 0.932 sits in the second.

The endpoints differ too. Removing the Haar subspace from these endpoints reduces accuracy by a median 0.907 and removing the character set reduces it by 0.783, so after a brief Haar window the model still depends heavily on both. The character window's endpoints lose 0.0001 when their masked set is ablated. A brief mask on the character set is therefore followed by a solution built elsewhere, while a brief mask on an arbitrary subspace is followed by a solution that uses the unmasked directions and the character directions alike.

Scope. One alternative orientation, one window length, one cohort whose earlier outcomes were known. It constrains how the window finding may be described, since a generic brief perturbation does not reproduce it, and it does not explain why the character directions behave differently. Attainment in 16/16 also shows that the sustained Haar failure is specific to maintaining that constraint rather than to disturbing the subspace at all.

B.20 Restoration after Unmasking: Fresh-Run Confirmation

A prespecified test on 16 runs never used before: ReLU addition seeds 920201–920208 and multiplication 920211–920218, under the unchanged recipe of the discovery cohort ($p = 113$, training fraction 0.3, one layer, $d_{\text{model}} = 128$, AdamW at 10^{-3} with weight decay 1 and $\beta = (0.9, 0.98)$). Each run trains its own uninterrupted normal prefix to update 2,000, where its masked set S is drawn by the existing random-half rule from the run identity alone and never from a later outcome. Whether a prefix has already reached 99% is reported and never changes inclusion; no seed is replaced and no arm is retried. The plan expands to 176 jobs, 16 source preparations and 160 continuations, each run to a fixed absolute update 50,000 with no stopping at attainment.

Arms. Each run contributes one normal continuation from update 2,000 and, at each of the cuts 2,100, 2,300 and 2,500, a sham, a native and a phase arm. The window masks updates 2,001 through cut $- 1$, writes the projected embedding and removes its hook before the cut update, which runs unmasked. The saved cut is therefore *after* that first unmasked update, and no claim that the content of S is zero at the cut is made. All three arms retain the window branch's entire AdamW state including moments and step count, share the normal branch's saved future randomness, and differ only in

what is added to the operand columns of the embedding: nothing, the full restoration increment $(E_N - E_W)Q_S$, or that increment with each embedding-row and frequency pair independently rotated. The rotation matches requested energy per row and frequency, not final embedding norm, immediate loss or functional disruption. Requested and realized spectral energies and leakage are recorded, and an independent Fourier reconstruction checks the transformation together with exact optimizer and randomness preservation.

The prespecified decision rule. Onset T is the first of five consecutive recorded observations at or above 99% test accuracy on the branch's own post-cut grid; prefix observations are never spliced in. The primary contrast for a run is

$$C = \big(T_{\text{native},2100} - T_{\text{phase},2100}\big) - \big(T_{\text{native},2500} - T_{\text{phase},2500}\big),$$

positive when restoring earlier delays generalization more. Within each task the statistic is the number of the eight planned runs with $C > 0$, tested one-sided against $P(C > 0) \leq \frac{1}{2}$ with a fixed denominator; ties, missing, invalid and censored runs contribute zero positives and remain in the denominator. A censored onset is never replaced by 50,000. Both tasks were required at $p \leq 0.05$, which needs seven or eight positives in each, *and* every planned job had to complete and pass validation.

Outcome. Addition returns 7/8 ($p = 0.035$) and multiplication 8/8 ($p = 0.004$). All 176 jobs completed and validated, so the batch-integrity condition holds and the prespecified conjunction is met. C is positive in 15 of 16 runs, median 2,350 updates; the exception is addition 920208 at -100, one recording interval. A positive interaction does not by itself establish that native restoration delays generalization, so the component contrasts are published alongside it.

Table 26: **Every run of the confirmation.** C is the prespecified early-minus-late interaction. The three middle columns are native-minus-phase onset differences in updates at each cut, positive when the native arm generalizes later. The last column is the native-minus-phase difference in same-S accuracy damage at update 50,000, in percentage points, at the earliest cut.

Run	C	Cut 2,100	Cut 2,300	Cut 2,500	Δ damage (pp)
addition 920201	400	700	300	300	79.9
addition 920202	1,500	1,700	400	200	−0.0
addition 920203	1,800	2,500	600	700	7.1
addition 920204	1,500	1,800	600	300	75.9
addition 920205	3,300	3,600	800	300	78.2
addition 920206	3,000	3,500	1,100	500	1.9
addition 920207	3,300	3,700	1,300	400	92.3
addition 920208	−100	300	300	400	8.6
multiplication 920211	3,800	4,600	1,900	800	58.4
multiplication 920212	2,300	2,500	600	200	73.8
multiplication 920213	2,600	3,400	700	800	79.4
multiplication 920214	2,400	2,700	1,100	300	32.9
multiplication 920215	2,100	2,300	500	200	9.5
multiplication 920216	4,900	5,300	1,400	400	64.3
multiplication 920217	6,300	6,700	1,300	400	0.0
multiplication 920218	1,400	1,600	400	200	68.5

The three arms, and why the sham matters. Native minus phase has medians 2,600, 650 and 350 updates at the three cuts; native minus sham has 2,550, 700 and 300. The phase arm is never later than sham at the earliest cut and is within one 100-update recording interval of it in 13 of 16 runs, spanning only -400 to 0. Without the sham arm the comparison would license only "later than a phase-matched increment"; with it, an energy-matched increment in scrambled phase causes no measurable delay while the native increment delays generalization by thousands of updates. Native is later than *both* controls by more than the prespecified 200-update margin in 16/16 runs at cut 2,100, 15/16 at 2,300 and 10/16 at 2,500.

The prespecified secondary, which does not follow the timing. The addition secondary counts runs where native is later than both controls by more than 200 updates *and* exceeds both in final same-S accuracy damage by more than 20 percentage points. It is observed in 4 of 8 addition runs, a procedure yield rather than a prevalence estimate, and by protocol it never changes the primary gate. The addition runs split sharply: damage differences of -0.0, 1.9, 7.1 and 8.6 points against 75.9, 78.2, 79.9 and 92.3, so the prespecified 20-point margin lies inside an empty 67-point gap and is doing no work. Multiplication is graded rather than split, from 0.0 to 79.4 points, and by protocol it is descriptive only and never an alternate gate. The timing effect is therefore present in runs whose endpoints recover no comparable dependence on S.

Scope. The three cuts change the masked duration, the recipient's age and the donor increment together, so no duration effect is isolated, no critical period is located, no irreversibility is established and no predictor of the unmasking time is fitted. Phase scrambling matches the increment's spectral energy, not the immediate loss: earliest-cut native restoration has lower training cross-entropy and higher test cross-entropy than phase. One phase draw per run is tested. Branches, cuts and sensitivity definitions are not independent units; the unit of inference is the run. The middle cut, the 98% threshold and the one-, three- and five-observation streaks are descriptive sensitivities and no replacement primary rule was selected from them. This is a bounded procedural replication conditional on the two tasks, the architecture, the recipe and the mask rule; it tests neither universality nor prevalence.

Provenance of the batch. A first attempt was interrupted by a host failure after 120 of its 176 jobs; the entire plan was re-run under an unchanged specification, and every one of the 104 cells both attempts produced agrees exactly on onset, endpoint damage and endpoint accuracy.

B.21 Restoration: the Development Study that Fixed the Intervention

The confirmation of Appendix B.20 was designed from a development study on four discovery-cohort runs (addition 910201 and 910202, multiplication 910211 and 910212, the first two identifiers per task), 94 continuations in all, with protocols recorded before their outcomes but the runs already known. It fixed the restoration increment $(E_N - E_W)Q_S$ and its phase-scrambled control, the cuts 2,100, 2,300 and 2,500, and the doses. Its outcomes, all descriptive: native-minus-phase onset differences decreased strictly across the three cuts in every run, under all six recorded onset definitions (median native-minus-sham 2,600, 650 and 300 updates); at cut 2,100 native restoration lowered training cross-entropy and raised test cross-entropy against phase in all four runs, as in the confirmation; the two earliest-cut addition branches recovered large endpoint dependence on S, while the multiplication branches were delayed without it; and exchanging whole AdamW states between the normal and window branches at cut 2,500 did not transfer the timing advantage, which stayed with the weights. A first-moment swap produced a large update shock and is not read as evidence either way. The per-run tables, the optimizer-exchange arms and the Haar transport arms are in the artifact; nothing in the paper rests on this study beyond the design it fixed.

B.22 What the Restored Content Does, and What Sets Its Delay

A post-hoc analysis of the confirmation's recorded trajectories, then three follow-up batches on its 16 runs at cut 2,100, all run after the confirmation's outcomes were known. The reset batch had its readings fixed before it ran; the others are descriptive. Nothing here carries a gate or changes the confirmed result. Every branch's trajectory, injection record and compact result are shipped, with the script that computes each number below.

What the optimizer does with the injected content, in absolute terms. The trajectories record only the fraction of centered operand-embedding energy inside S, and a fraction is invariant under the uniform shrinkage that decoupled weight decay alone would produce, so its decay cannot be compared with a weight-decay rate. We therefore replayed the first 1,000 updates of the confirmation's sham, native and phase arms at cut 2,100 from their shipped composed inputs, which reproduces the shipped trajectories exactly (0 mismatches in 160 checked evaluations), and split each AdamW step on $Z_t = X_t Q_S$ as $Z_{t+1} = (1 - \mathrm{lr}\,\mathrm{wd})Z_t + U_t$, U_t being the non-decay part of the update, so that $\|Z_{t+1}\|^2 = (1 - \mathrm{lr}\,\mathrm{wd})^2\|Z_t\|^2 + 2(1 - \mathrm{lr}\,\mathrm{wd})\langle Z_t, U_t\rangle + \|U_t\|^2$. Decay alone would leave 0.14 of

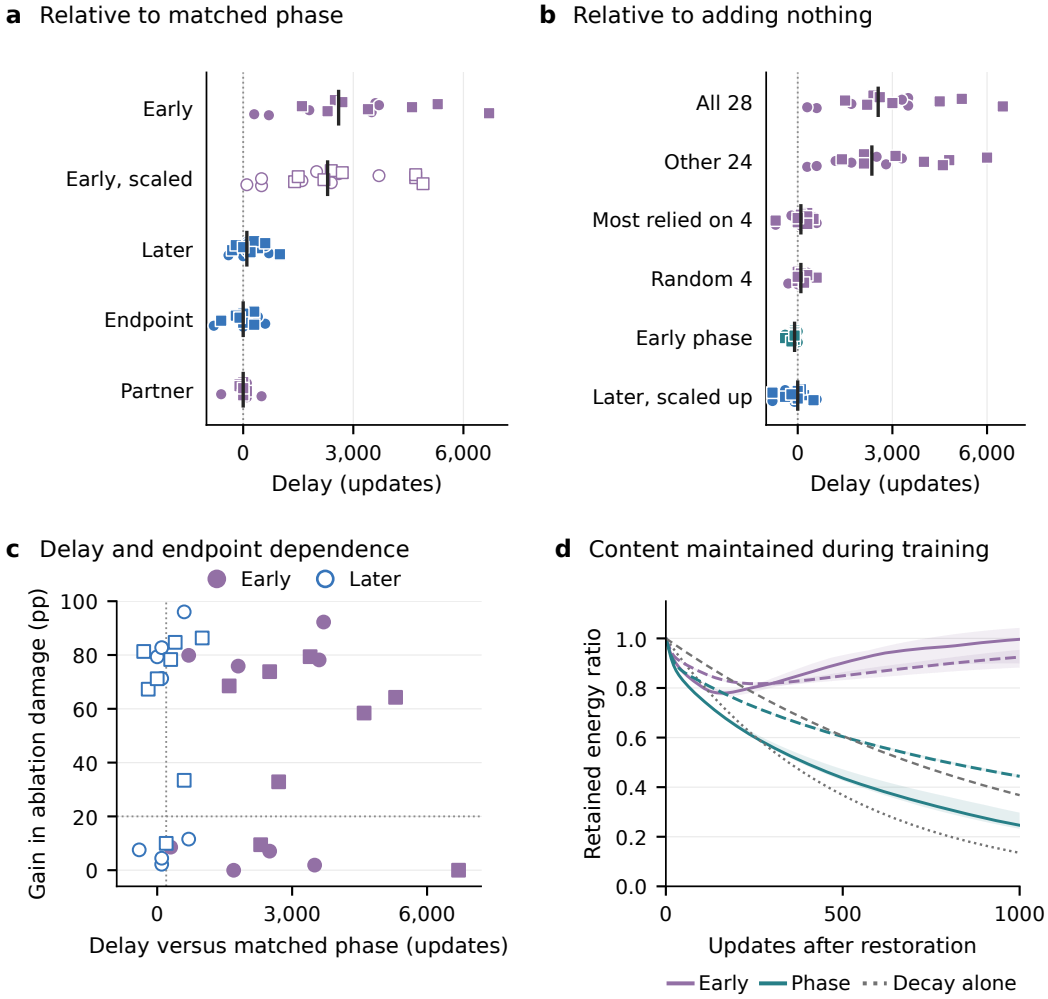

Figure 7: Restoration comparisons with explicit baselines. (a) Each native increment versus its own scrambled control at matched energy. Early content is from update 2,100, later content from confirmed generalization, endpoint content from update 50,000, and partner content from another run. Scaled early content matches the later increment's norm within each run. (b) Comparisons with adding nothing (sham). The first four rows restore early content in all 28 frequencies, the other 24, the four most relied on within S, or four random frequencies. Early phase restores the scrambled early increment; later content is scaled up to the early norm. Black ticks mark medians; every row shows all 16 runs, with circles for addition and squares for multiplication. (c) Delay and the gain in endpoint ablation damage, both relative to the corresponding phase control. Dotted lines show the prespecified 200 update and 20 percentage point margins. (d) Energy in S relative to its value immediately after restoration, with medians and interquartile ranges. Solid and dashed colored curves use weight decay 1 and 0.5, respectively, on separate cohorts of 16 runs. Gray curves show decay alone. These trajectories describe maintenance; they do not identify why content delays learning.

the injected energy after 1,000 updates. The update terms are summed along the realized trajectory without discounting by later decay, $\sum_t \left[2(1 - \mathrm{lr}\,\mathrm{wd})\langle Z_t, U_t\rangle + \|U_t\|^2\right]$, and divided by the injected energy $\|D\|^2$; the ratio compares $\|Z_{1000}\|^2$ with $\|D\|^2$. Because replenished energy decays again, the summed update terms can exceed the injected energy while the ratio stays near one. For the native increment the absolute S-energy ratio is a median 1.00 (range 0.77–1.13): the update terms return a median 1.77 times the injected energy, positive in 16/16 and exceeding half the decay in 16/16, and what remains stays aligned with the injected increment, cosine 0.97. For the scrambled increment

the ratio is 0.25 (0.21–0.41), the update terms return 0.20 and exceed half the decay in 0/16, and the whole centered embedding shrinks to 0.69 of its energy while the native arm's grows to 1.15, which is why the fraction is a misleading statistic here. The native ratio exceeds the phase ratio in 16/16. This is the optimizer's update, momentum and preconditioning included, not the instantaneous gradient; and it shows that the recipient replenishes aligned content, not why that delays generalization. The follow-up predictions below were written on the fraction's excess over sham, fit log-linearly: phase median 1.63×10^{-3} per update, native 0.33×10^{-3}, phase faster in 16/16.

A large delay does not require sustained high measured dependence. In the five runs whose native endpoint dependence on S stays below 0.15, the native arm's same-S damage peaks at 0.06–0.18 before onset, yet those runs were delayed by 300 to 6,700 updates; how long the content persists does not scale the delay (Spearman $+0.16$, $p = 0.55$). The later recipient replenishes less: at cut 2,500 the native S-energy ratio after 1,000 updates is 0.27, the update terms returning 0.22 times the injected energy, against 1.00 and 1.77 at cut 2,100.

A common AdamW reset, with predictions fixed first. The confirmation's sham, native and phase compositions at cut 2,100, then the AdamW moments emptied and the step counter restarted, the absolute clock unchanged; 48 continuations. Three predictions were written before launch. First, native later than both reset controls by more than 200 updates in at least 14/16: **16/16**, medians 2,900 against phase and 2,800 against sham (retained state: 2,600 and 2,550). Second, phase never later than sham by more than 200 *and* within one recording interval of it in at least 12/16: the first clause holds and the second does not, 9/16, because the phase arm under reset is *earlier* than sham by 0 to 500 updates, a failure in the direction that does not bear on the claim. Third, the fraction's excess decaying faster for phase than for native in 16/16 with a phase median inside $[1.2, 2.4] \times 10^{-3}$ (written before the absolute decomposition above existed, and to be read as the descriptive statistic it is): **16/16**, phase 1.52×10^{-3}, native 0.33×10^{-3}, unchanged from the retained state. The protocol's strongest reading required all three, so the batch does not reach it. Against each arm's retained-state counterpart the reset delays generalization by a median $+100$ (sham), $+200$ (native) and 0 (phase) updates. Neither the delay nor the sustaining depends on the retained optimizer state.

A coherent-phase control. The native increment with one rotation per frequency shared by every embedding row, so per-frequency energy is matched exactly and the cross-row organisation is kept while the phase in the operand direction is shifted; a check confirms every row received the same factor. Window AdamW retained; 16 continuations. Later than both phase and sham by more than 200: **0/16**. Median onset difference from native $-3{,}100$, from phase 0, from sham -100. Fraction-excess decay 1.20×10^{-3}, slower than the scrambled increment in 16/16 but about four times faster than native. Seven endpoints depend on S above 0.15 after coherent injection, and in five (920205, 920207, 920211, 920212, 920218) neither sham nor phase does. Reported as observed: an increment with the native per-frequency structure but a shifted operand-direction phase causes no delay and, like the scrambled one, is not sustained, though the endpoint can still come to depend on S.

Donor against recipient. The increment $(E_N(t_d) - E_W(t_r))\, Q_S$ applied to the window recipient at t_r with the recipient-cut normal branch's future randomness, so each cross arm shares its future with the confirmation's arms at that cut; native and scrambled versions; 64 continuations. Donor 2,500 into recipient 2,100: native minus phase median 2,550 (same-time 2,600), later than both controls by more than 200 in 16/16, fraction-excess decay 0.35×10^{-3} for native and 1.51×10^{-3} for phase, and the immediate ordering, lower training and higher test cross-entropy for native, in 16/16. Donor 2,100 into recipient 2,500: median 300 (same-time 350), later than both in 7/16, fraction-excess decay 0.95×10^{-3} for native and 1.16×10^{-3} for phase. Under the majority rule fixed in advance the recipient sets the delay in 15/16 runs and the donor in 0/16. The recipient at 2,500 sustains neither donor's content.

Half dose. $\alpha = 0.5$ of the native and of the scrambled increment at cut 2,100; 32 continuations. Native-half minus phase-half, divided by the same run's full-dose delay: median 0.51, interquartile range 0.39–0.64, range 0.29–1.00, "proportional" under the bands fixed in advance; median half-dose delay 1,350 updates. The half increment's fraction excess does not decay (-0.03×10^{-3}, a slight growth); endpoint dependence median 0.06.

Reading across the batches. Across these six conditions, whether the recipient's optimizer update sustains the injected content and whether that content delays generalization move together: an early recipient with aligned content, sustained and delaying generalization; a late recipient, scrambled content, or coherently shifted content, neither. Within the sustained condition, how long the content lasts does not set how much it delays generalization, and half of it gives half the delay. This is co-variation across conditions, not a mechanism. At update 2,100, immediately after restoration, native content has lower training cross entropy than phase (medians 1.95 versus 3.38) and higher test cross entropy (13.4 versus 8.5). Both directions hold in all 16 runs. This comparison does not show an improvement over sham. Why replenishing it delays generalization, and what the recipient loses by cut 2,500, are not answered here.

B.23 The Same Intervention under Weight Decay 0.5

Sixteen runs never used before (addition 930201–930208, multiplication 930211–930218), trained with the confirmation's recipe except weight decay 0.5 in place of 1: forked at update 2,000 with S drawn by the same random-half rule, a 99-update window, and sham, native and phase arms composed by the same code at cut 2,100, every arm run to 50,000 with the window branch's AdamW retained; a normal continuation from 2,000 supplies each run's saved time. Three predictions were fixed before launch: first, native later than the later of phase and sham by more than 200 updates in at least 7 of 8 per task; second, phase never later than sham by more than 200; third, in the absolute decomposition of the first 1,000 updates, the update terms exceed half the decay for native in at least 14 of 16 and for phase in at most 2 of 16, where decay alone now leaves 0.37 of the injected energy, against 0.14 at weight decay 1.

All three hold. Every normal branch generalizes within budget, at onsets of 9,700–40,900 (median 16,500) against a median 6,050 for the window branch. Native is later than both controls by more than 200 in 16/16, 8/8 per task ($p = 0.0039$ each), by a median 3,700 updates against phase and 3,600 against sham; phase is never later than sham (range -700 to 0, within one recording interval in 3/16). The earliest restoration gives back a median 0.35 of the saved time. Endpoint dependence again splits: native exceeds phase by more than 20 points in 9 of 16 (5 addition, 4 multiplication) and by at most 1.1 points in the other 7. In the decomposition the native S-energy ratio is 0.92 (0.72–0.99), the update terms return 0.79 of the injected energy and exceed half the decay in 16/16, and what remains has cosine 0.98 with the increment; phase is 0.44 (0.41–0.48), just above the decay-only 0.37, with update terms of 0.07 and 0/16. The replay reproduces the shipped trajectories exactly (0 mismatches in 160 evaluations). Halving the decay halves what the update has to return (0.79 against 1.77) and lifts the scrambled increment's ratio from 0.25 to 0.44, as the decay-only values predict.

B.24 Restoring the Solution's Own Content Early

Every restoration above puts back the normal branch's S content from the cut itself or from a nearby cut, and that content is not what the finished normal branch uses. Over the confirmation's 16 runs, the normal arm's centered S component at its own onset has cosine 0.51–0.70 with its S component at update 2,100, and its 50,000-update endpoint has cosine -0.00–0.40; the S energy falls from 184–251 at 2,100 to 55–230 at onset and 8–224 at the endpoint. So the confirmation does not say what happens when the solution's own content arrives early. This batch restores it. The donors are the normal arm's saved states at the update where its 0.99 onset is confirmed (5,800–14,600 across runs) and at 50,000; the recipient is the window branch at cut 2,100, and the increment $(E_N(t_d) - E_W)Q_S$ replaces the recipient's S component by the donor's, composed by the code of the donor–recipient cross with the same retained AdamW, the same future randomness and a phase-scrambled control at matched energy for each donor. Sham, native and phase at 2,100 are read from the confirmation. Readings were fixed before launch. Per run, D is the confirmation's native minus phase delay, A the age arm's native minus phase delay, and ρ the age increment's norm over the confirmation increment's norm. The half-dose batch calibrates dose (amplitude 0.5 gave a delay ratio 0.51), so dose alone predicts $A/D \approx \rho$; the dose-normalized ratio is $r = (A/D)/\rho$. For each donor age: the confirmation's own criterion (native later than both controls by more than 200 updates in at least 7 of 8 per task); *dose alone* if the median r lies in $[0.65, 1.35]$; *causes less delay than its dose predicts* below 0.5, with the count of runs in which the native arm is earlier than sham by more than 200 reported separately; *causes more delay* above 1.5; between bands otherwise.

Outcome: both ages cause less delay than their dose predicts. Table 27 gives every run. All 64 continuations completed. Neither donor age reproduces the early content's delay. The confirmed content is later than both controls by more than 200 updates in 4 of 16 runs (1 addition, 3 multiplication) and the endpoint content in 2 of 16, against 16 of 16 for the 2,100 content on the same recipients. The median native minus phase delay is 100 updates for the confirmed content (range -400 to 1,000) and 0 for the endpoint content (-800 to 600); native minus sham is 100 and -50; phase lies within 200 of sham in 14 and 13 of 16. The amplitude ratios are 0.78 (0.48–1.00) and 0.71 (0.19–0.95), so dose alone would put the confirmed content's delay near 0.78 of 2,600. The dose-normalized ratio r has median 0.08 for the confirmed content and 0.00 for the endpoint content, and is below 0.5 in 16 of 16 at both ages. The native arm is earlier than sham by more than 200 in 3 of 16 at each age, so the mature content does not bring onset forward either. On arrival the confirmed content fits the training set worse than its scrambled control in 14 of 16 and the endpoint content in 16 of 16, where the 2,100 content fits it better in 16 of 16; test cross-entropy is higher than the control's in 16 of 16 at both ages.

The mature content is kept and used. In the decomposition replay the confirmed content's absolute S-energy ratio after 1,000 updates is 0.94 (0.62–1.26) against 0.14 under decay alone, with update terms of 1.75 times the injected energy and cosine 0.91 with the increment; its scrambled control is 0.35 (0.22–0.59). The endpoint content gives 0.77 (0.64–2.25) with cosine 0.81, against 0.42 for its control; native exceeds phase in 16 and 15 of 16. At 50,000 the confirmed-content arm's same-S dependence is 0.79 at the median (0.02–0.96) against 0.00 for its phase control, exceeding it by more than 20 points in 11 of 16; the endpoint-content arm's is 0.58 (0.00–0.94), exceeding its control by more than 20 points in 9 of 16. The 2,100 content on the same recipients ends at 0.71 with 10 of 16 above the margin. The replays reproduce the shipped trajectories exactly (0 mismatches in 640 evaluations).

Reading. Restored into the same unmasked state, the normal branch's content from before its generalization delays by a median 2,600 updates; its content at its confirmed generalization, at 0.78 of the amplitude, delays by 100, and the finished model relies on it as much as on the early content. The delay therefore comes from what the normal branch holds in S before it generalizes. It is not caused by the directions being occupied, by the amount of normal-branch content restored, or by the endpoint's reliance on that content. What in the early content delays generalization is not identified here, and the batch reuses the confirmation's runs at one cut and one recipe.

Table 27: **Content age, every run.** Onsets in updates. D is the confirmation's native minus phase delay at cut 2,100. For each donor age: t_d the donor's update, ρ the increment's norm over the 2,100 increment's norm, the native minus phase delay, native minus sham, the native minus phase difference in endpoint damage on S (pp), and the native arm's same-S dependence at 50,000.

			confirmed-generalization donor						endpoint donor (50,000)				
run	normal	D	t_d	ρ	delay	$-$sham	Δdmg	dep.	ρ	delay	$-$sham	Δdmg	dep.
add 920201	5,400	700	5,800	1.00	0	-100	79.3	0.79	0.86	0	-100	79.9	0.80
add 920202	7,400	1,700	7,800	0.48	100	100	2.2	0.02	0.19	0	0	-0.1	0.00
add 920203	8,000	2,500	8,400	0.80	100	200	4.5	0.86	0.67	300	0	8.9	0.90
add 920204	7,100	1,800	7,500	0.79	100	0	71.3	0.71	0.63	0	0	46.0	0.46
add 920205	9,900	3,600	10,300	0.78	100	0	82.7	0.83	0.70	-100	-200	66.4	0.67
add 920206	10,200	3,500	10,600	0.77	700	700	11.6	0.12	0.71	600	600	0.0	0.00
add 920207	9,100	3,700	9,500	0.96	600	200	96.0	0.96	0.95	400	0	93.2	0.94
add 920208	6,600	300	7,000	0.72	-400	-700	7.6	0.08	0.61	-800	-700	-4.4	0.20
mul 920211	9,100	4,600	9,500	0.96	400	400	84.7	0.85	0.83	-100	-100	18.5	0.19
mul 920212	6,900	2,500	7,300	0.86	300	200	78.3	0.78	0.78	300	200	85.3	0.85
mul 920213	10,000	3,400	10,400	0.66	-300	-400	81.3	0.82	0.72	-600	-900	76.2	0.80
mul 920214	7,500	2,700	7,900	0.83	600	600	33.4	0.83	0.64	0	-100	20.9	0.70
mul 920215	7,900	2,300	8,300	0.65	200	100	10.0	0.10	0.48	0	0	0.0	0.00
mul 920216	11,800	5,300	12,200	0.68	-200	-300	67.3	0.67	0.66	-200	-300	64.4	0.65
mul 920217	14,200	6,700	14,600	0.83	1,000	1,000	86.3	0.86	0.74	300	300	0.0	0.00
mul 920218	6,000	1,600	6,400	0.78	0	-100	71.2	0.72	0.80	-100	-200	41.1	0.51

B.25 The Solution's Own Content at Weight Decay 0.5

The content-age comparison of Appendix B.24, repeated on the 16 runs of the weight-decay 0.5 replication (Appendix B.23; addition 930201–930208, multiplication 930211–930218), which were fresh when that replication ran. Each run's normal-branch content in S at its confirmed generalization

and at its 50,000-update endpoint is restored into the unmasked state at cut 2,100, each with a scrambled control, the cut-2,100 normal branch's future randomness and retained AdamW; 64 continuations. Readings fixed before launch use the same bands as before, with D this cohort's early-content delay (median 3,700) and $r = (A/D)/\rho$: *causes less delay than its dose predicts* if the median r is below 0.5. Expected: that reading at both ages, with the confirmed content kept and relied on.

Outcome. Both ages cause less delay than their dose predicts. The confirmed content, at a median 0.81 of the early increment's amplitude (0.49–1.02), delays by a median 600 updates against its control (range $-3{,}000$ to 1,700), later than both controls by more than 200 in 10 of 16 (5 of 8 per task, so the primary count is not met); its median r is 0.18, below 0.5 in 15 of 16. The endpoint content, at 0.69 (0.11–0.97), delays by a median 50, later than both controls in 3 of 16, with r 0.01, below 0.5 in 16 of 16. In two multiplication runs (930211 and 930218) both later contents bring onset forward by more than 2,000 updates against sham. On arrival the confirmed content fits the training set worse than its control in 13 of 16 and the endpoint content in 16 of 16, where the early content fits it better; test cross-entropy is higher than the control's in 16 of 16 at both ages. Both are kept: in the decomposition replay the retained S-energy ratio after 1,000 updates is 0.87 (0.79–1.03) for the confirmed content and 0.81 for the endpoint content, against 0.50 and 0.56 for their scrambled controls and 0.37 under decay alone, and the replays reproduce the shipped trajectories exactly. At 50,000 the endpoint's same-S dependence is a median 0.46 after the confirmed content and 0.24 after the endpoint content, against 0.00 for the controls.

Reading. At the lower weight decay the confirmed content delays generalization more than at weight decay 1 (600 against 100 updates) but still by about a fifth of what its dose predicts, and it is kept and relied on. The ordering of Appendix B.24 holds: the early content delays generalization, and the solution's later content in the same directions causes much less delay than its dose predicts.

B.26 The Early Content at the Confirmed Amplitude, and Random Content at the Fork

Two readings of the preceding results remained open: that the confirmed content causes less delay only because its increment is smaller, and that any disturbance of S at the fork, not deletion in particular, brings generalization forward. Both were tested on the confirmation's 16 runs with readings fixed before launch. *Norm match*: the 2,100 native increment scaled by each run's ρ_h, the norm ratio of its confirmed-generalization increment to its 2,100 increment (0.48–1.00, median 0.78), into the same unmasked state, with the scrambled control scaled by the same ρ_h. *Fork*: from the normal branch's fork state at update 2,000, with the source's AdamW state and future randomness, the centered S component of the operand embedding removed once (the zero arm) or replaced once by Gaussian content in S of the same energy (the noise arm), after which training continues without any constraint; read against the normal arm from 2,000 and the sham arm.

Outcome. Amplitudes here are those of the restored increment. The receiving state holds almost no content of its own in S after its first unmasked update, so the full content of S after restoration has the increment's norm to within 0.1% in every restored branch of these comparisons (144 branches: the early and phase arms scaled here, the confirmed and endpoint contents of Appendix B.24, and the scaled-up and other-run arms of Appendix B.27; largest deviation 0.099%; a post hoc reading of the first update's decay term in the decomposition replays). The early content at the confirmed content's amplitude still delays generalization. It is later than its scaled control by a median 2,300 updates (100–4,900), later than both that control and sham by more than 200 in 15 of 16 (7 addition, 8 multiplication), and later than the confirmed content on the same recipient, at the same amplitude, in 16 of 16, by a median 1,900 (300–5,000). Its delay is a median 0.77 of the full-amplitude delay, against a median ρ_h of 0.78, so the delay scales with amplitude for the early content; the confirmed content at that amplitude gives 100. Amplitude therefore does not explain the content-age result. At the fork, deleting the S content once brings generalization forward in 16 of 16 (median ratio of total updates 0.72, 0.46–0.87, replicating the discovery cohort's 0.737), and replacing it by matched-energy Gaussian content does the same (0.69, 0.50–0.99; earlier than normal by more than 200 updates in 15 of 16, earlier at all in 16 of 16). The two arms are within 200 updates of each other in 10 of 16, with a median difference of 100 (-900 to 1,100); neither approaches the brief window's 0.44. The Gaussian

content is not held: its share of the embedding's energy falls from 0.50 to 0.34 by update 5,000, and the endpoint's same-S dependence is 0.33 against 0.04 after deletion. In the decomposition replay the scaled early content's absolute S-energy ratio after 1,000 updates is 1.33 (0.77–1.87) against 0.14 under decay alone, with update terms of 2.63 times the injected energy and cosine 0.96 with the increment; its scaled control is 0.33 (0.21–0.55), and the scaled early content exceeds its control in 16 of 16. The replays reproduce the shipped trajectories exactly (0 mismatches in 320 evaluations).

B.27 Another Run's Early Content, and the Confirmed Content at the Early Amplitude

Our reading is that the early content delays generalization because it is what fits this run's training set at the cut. Two arms test that against "any coherent early Fourier content in these directions delays generalization" and against "content at the early amplitude delays generalization". Both were fixed before launch, while the previous batch was running and before any of its outcomes had been read, on the confirmation's 16 runs, all into the unmasked state at cut 2,100 with the cut-2,100 normal branch's future randomness and retained AdamW. *Partner*: the 2,100 content of the next run of the same task in the seed list, cyclically, projected onto this run's S, $(E_N^{h'}(2100) - E_W^h(2100))\, Q_S^h$, with a scrambled control at matched energy; the partner trained on a different training split from a different initialization. Its compatibility with the recipient therefore changes along with the split; this control does not isolate either factor. *Confirmed, scaled up*: this run's confirmed-generalization content scaled by $1/\rho_h$ so that its norm equals the 2,100 native increment's. Readings fixed in advance: the partner content *causes no comparable delay* if its delay against its control lies within 200 updates in at least 12 of 16, and *causes a delay like this run's own* if it is later than both controls by more than 200 in at least 7 of 8 per task; the scaled-up confirmed content *causes no comparable delay* if it is later than sham by more than 200 in at most 4 of 16, and *causes a delay* if in at least 7 of 8 per task. The account predicts that neither causes a comparable delay.

Outcome. The partner's early content, at a median 1.00 of this run's own increment norm (0.83–1.17), causes no comparable delay: its delay against its scrambled control is a median 0 updates (-600 to 500), within 200 in 14 of 16, and it is later than both controls in 0 of 16; against sham it is a median -100. It is neither kept nor used: in the decomposition replay its absolute S-energy ratio after 1,000 updates is 0.24 (0.20–0.38), the same as its scrambled control's 0.25 and close to the decay-only 0.14, with update terms of 0.21 times the injected energy; at 50,000 the endpoint's same-S dependence is 0.00 at the median. On arrival it fits the training set better than its control in 6 of 16, where this run's own early content does in 16 of 16. The confirmed content scaled up to the early amplitude (median factor 1.28, 1.00–2.06) is later than sham by more than 200 in 2 of 16, with a median difference of 0 (-800 to 600); it arrives a median 2,750 updates before the 2,100 content at the same amplitude and within 100 of the confirmed content at its own amplitude. It is kept and used: ratio 0.68 (0.40–0.92), update terms 1.19 times the injected energy, cosine 0.93 with the increment, and same-S dependence 0.80 at 50,000. The replays reproduce the shipped trajectories exactly (0 mismatches in 480 evaluations).

Reading. Maintenance alone is insufficient for delay: confirmed content at the early amplitude is kept and relied on yet has a median delay of zero. Early Fourier content need not incur a comparable delay either: the partner's content, at a comparable amplitude, decays like its scrambled control and has a median delay of zero. Both predictions hold. The native early content, which delays generalization, also has a training-fit advantage over its scrambled control in every tested run, but the partner comparison cannot separate training-split specificity from compatibility with the recipient's learned coordinates and downstream weights. These comparisons do not establish that maintenance is necessary for delay, or whether the delay reflects slower circuit formation or longer interference with a circuit already formed.

B.28 Where in the Masked Set the Delay Comes From

On the confirmation's 16 runs, the normal branch's 2,100 content is restored into the unmasked cut-2,100 state in part of S only: in K_S, the four frequencies of S whose single ablation damages the normal endpoint most; in four frequencies of S drawn by a seeded rule; or in the other 24 frequencies of S. Recipient, randomness and optimizer are the native arm's. The subsets were fixed before launch,

with every other result known. Readings fixed in advance: the protocol label *delays like any part of its size* applies if K_S and the random four lie within 200 updates of each other in at least 9 of 16, *delays more* or *less* if one is later by more than 200 in at least 9 of 16. This label concerns the two tested subsets, not every four-frequency subset. The protocol also expected, under that reading, delays in proportion to how much of the increment each subset restores.

Outcome. The protocol's similarity criterion is met: K_S against the random four is within 200 updates in 9 of 16 (later in 3, earlier in 4, median 0). Each delays generalization by a median 100 updates against sham, 0.042 and 0.029 of the full restoration's delay (median 2,550), although they carry a median 0.47 and 0.37 of the increment's norm. The other 24 frequencies carry 0.89 of the norm and delay generalization by a median 2,350 updates, 0.944 of the full delay, later than sham by more than 200 in 16 of 16; the K_S and 24-frequency delays add up to a median 0.96 of the full delay.

Reading. Most of the measured restoration delay lies outside the four most relied-upon frequencies within S. Those four and the tested random four have the same median delay, while restoring the remaining 24 recovers most of the full delay. The two small subsets account for far less of the delay than their norm shares, so the account's expectation of proportionality fails in this form. Collective action of the broadband content is one possible explanation, but this batch does not establish how many frequencies must act together or rule out a concentration of the delay within the remainder.

B.29 Nested Subsets of the Masked Set

On the same 16 runs and cut, the early content is restored into nested random subsets of 8, 12, 16 and 20 of the 28 frequencies of S, drawn by a seeded rule fixed before launch (64 continuations). With r_k the median share of the full restoration's delay and e_k the median share of the increment's energy, the readings fixed in advance were: *collective* if $r_8 < e_8 - 0.15$, $r_{12} < e_{12} - 0.15$ and $1 - r_{16} \geq 2r_{12}$; *proportional* if $|r_k - e_k| \leq 0.15$ at every size; *front-loaded* if r_8 and r_{12} exceed their energy shares by more than 0.15; *intermediate* otherwise.

Outcome. The delay shares are 0.12, 0.31, 0.45 and 0.61 at 8, 12, 16 and 20 frequencies, against energy shares of 0.28, 0.42, 0.59 and 0.71. No run lies above its own energy share by more than 0.15 at any size; at 8 frequencies 8 of 16 lie below by more than that margin. Only the first clause of the collective reading holds, so the label is *intermediate*. The nested curve is monotone within 200 updates in 14 of 16 runs.

Reading. Small parts of the early content delay generalization less than their share of its energy, and the delay grows steadily with the share restored, without a sharp rise near the full set. This supports a graded effect spread over the set; it does not show that the frequencies must act together.

B.30 A Random 24 That Keeps the Four

The 24 frequencies of S outside K_S carried 0.944 of the delay (Appendix B.28). To ask whether this reflects which frequencies they are or how many, the early content is restored on the same runs into S without the random four of Appendix B.28, a random 24 that keeps K_S. With d the onset of $S \setminus K_S$ minus that of this set, the readings fixed in advance were: *size, not identity* if $|d| \leq 200$ in at least 11 of 16; *identity* if $d > 200$ in at least 9; the *reverse* if $d < -200$ in at least 9; *unresolved* otherwise.

Outcome. $|d| \leq 200$ in 6 of 16, $d > 200$ in 7 and $d < -200$ in 3, median $+100$, so the label is *unresolved*. The two 24-frequency sets carry 0.83 and 0.94 of the full delay. In the 12 runs where the random four and K_S are disjoint, $|d| \leq 200$ in 5, $d > 200$ in 6 and $d < -200$ in 1, median $+200$. The batch does not separate which frequencies are restored from how many.

B.31 Suppressing the Rebuild of Restored Content

Section 3.4 tests whether the optimizer's replenishment of restored early content is part of the cause of its delay. This appendix gives the readout, the three exploratory steps on four pilot runs that shaped the test, and the extension to the other twelve runs of the restoration confirmation.

Structured readout. Logits are centered over all output classes. With group coordinates u, v of the operands (discrete logarithms for multiplication, zero operands excluded from this readout only) and $s = (u + v) \bmod n$, let $H(s, c)$ be the mean centered logit of class c over the pairs with sum s. The *structured* output is H minus its mean over s, the projection onto every nonconstant sum-angle mode; the *residual* is the rest of the logit. To keep training logits out of it, H is estimated on one of two fixed folds of held-out pairs, stratified by s, and scored on the other, then swapped (crossfit). Structured attainment is the first of three consecutive saved readouts with crossfit cross-entropy at most 0.1; readouts are saved every 100 updates for 2,000 updates after restoration and every 200 thereafter. The projection was validated against the existing two-dimensional Fourier implementation, for exact additive reconstruction and on a synthetic lookup-table negative control. It reads task-aligned output; it does not identify an internal circuit.

Exploratory steps on four runs. The first two runs of each task (addition 920201 and 920202, multiplication 920211 and 920212), chosen before any new outcome, were replayed from the restored states at update 2,100 with zero replay error. (i) Output diagnosis: at the update where the phase control generalizes, early native restoration leaves a residual margin of -11.5 to -25.3 on the true class, against -4.6 to -6.3 for phase, and reaches crossfit cross-entropy 0.1 later by 400 to 4,400 updates, most of its full delay; the descriptive thresholds were chosen after the outcomes were seen. (ii) Suppressing the positive update component along the restored direction for 2,000 updates advanced both endpoints for early content in all four runs and beat a perturbation of equal magnitude in an orthogonal direction; for later content it gave no consistent benefit. (iii) Because suppression also shrinks the update, a scaling control, chosen after (ii), shrinks the whole embedding update to the norm suppression would leave at the same state; it did not bring generalization forward, and suppression preceded it for both endpoints in all four runs.

Extension. A protocol fixed before any intervention outcome on the remaining twelve runs (addition 920203–920208, multiplication 920213–920218; their restoration outcomes were known) runs ordinary training, suppression and scaling from the early and the later restored state of each run, 72 continuations. With d the unit restored direction in S, fixed per run and donor, and U the adaptive part of each AdamW step (the proposed step minus weight decay), suppression subtracts $\max(\langle U, d\rangle, 0)\, d$ and scaling replaces U by $\sqrt{1 - \max(\langle U, d\rangle, 0)^2/\|U\|^2}\, U$, on updates 2,101 to 4,100; training is then ordinary to update 12,100. Weight decay, optimizer moments, counters and all other parameters are untouched by either operation. Two readings were fixed: an *early effect* if suppression is strictly earlier than both ordinary training and scaling on both endpoints, in at least 5 of 6 runs per task; and an *age effect* if the early gain exceeds the later gain in all four comparisons, in at least 5 of 6 per task.

Outcome. Both readings hold in 6 of 6 runs per task, 12 of 12 in all; no run is censored (Table 28, Figure 8). For early content, suppression advances full generalization by a median 1,450 updates against ordinary training and 1,350 against scaling, and structured attainment by 1,300 and 1,400, with every single gain between 200 and 4,500 updates; it shortens the wait for full generalization after restoration by a median 0.32 of its ordinary length. Scaling alone changes neither endpoint at the median (0; full generalization from 700 updates later to 300 earlier). For later content, suppression is earlier than ordinary training for full generalization in 1 run, tied in 5 and later in 6, by up to 800 updates, and never earlier than scaling. Under ordinary training, early content reaches structured attainment a median 2,300 updates after later content and full generalization 2,550 after, in 12 of 12. All 24 ordinary-training runs reproduce the recorded restoration trajectories exactly, and 1,008 checks find optimizer states and other parameters unchanged by the corrections.

Limits. The norm match is local: scaling matches suppression at the same state, after which the arms follow different trajectories, so their cumulative update magnitudes and content exposure are not equal, and the result does not exclude every magnitude or exposure account. It shows a causal contribution of updates along the early restored direction; it does not identify the downstream computation they delay, show that the structured readout mediates the behavioural delay, or establish necessity or sufficiency. One architecture, one weight decay, and the runs of one cohort.

Extension: a norm-matched radial update control. Because d overlaps the radial direction of the embedding parameter ($\cos(d, W_E/\|W_E\|)$ 0.62–0.70; Appendix B.44), a protocol fixed before any branch ran asked whether a content-agnostic radial suppression reproduces the content dependence

Table 28: **Suppressing replenishment, all twelve extension runs.** Updates after restoration at 2,100 to full generalization and to structured attainment, as ordinary training / suppression / scaling, for the early and the later restored content.

Run	Early, full	Early, structured	Later, full	Later, structured
add 920203	4,500 / 2,900 / 4,200	2,600 / 1,300 / 2,800	2,200 / 2,200 / 2,000	900 / 800 / 900
add 920204	2,800 / 2,300 / 2,700	2,200 / 1,300 / 2,200	1,100 / 1,200 / 1,100	600 / 600 / 600
add 920205	4,900 / 2,400 / 4,600	3,800 / 1,500 / 3,800	1,400 / 1,400 / 1,400	800 / 800 / 800
add 920206	4,800 / 3,300 / 4,700	3,200 / 2,000 / 3,400	2,000 / 1,900 / 1,900	800 / 800 / 800
add 920207	5,300 / 3,900 / 5,300	4,200 / 2,800 / 4,200	2,200 / 2,900 / 2,200	1,000 / 1,400 / 1,000
add 920208	1,900 / 1,700 / 1,900	1,000 / 800 / 1,000	900 / 900 / 900	400 / 400 / 400
mul 920213	5,200 / 3,500 / 5,200	4,000 / 2,400 / 4,000	1,800 / 1,800 / 1,800	800 / 800 / 800
mul 920214	3,800 / 2,900 / 3,900	3,000 / 1,700 / 3,000	1,800 / 1,900 / 1,200	800 / 900 / 700
mul 920215	3,300 / 2,200 / 3,300	2,200 / 1,200 / 2,200	1,200 / 1,200 / 1,200	600 / 600 / 600
mul 920216	7,300 / 4,200 / 8,000	6,800 / 3,600 / 7,400	1,800 / 2,600 / 1,800	1,100 / 1,700 / 1,100
mul 920217	8,400 / 3,900 / 8,200	7,400 / 3,000 / 7,000	2,900 / 3,400 / 2,800	2,000 / 2,200 / 2,200
mul 920218	2,700 / 2,100 / 2,700	1,500 / 1,100 / 1,500	1,100 / 1,300 / 1,100	600 / 600 / 600

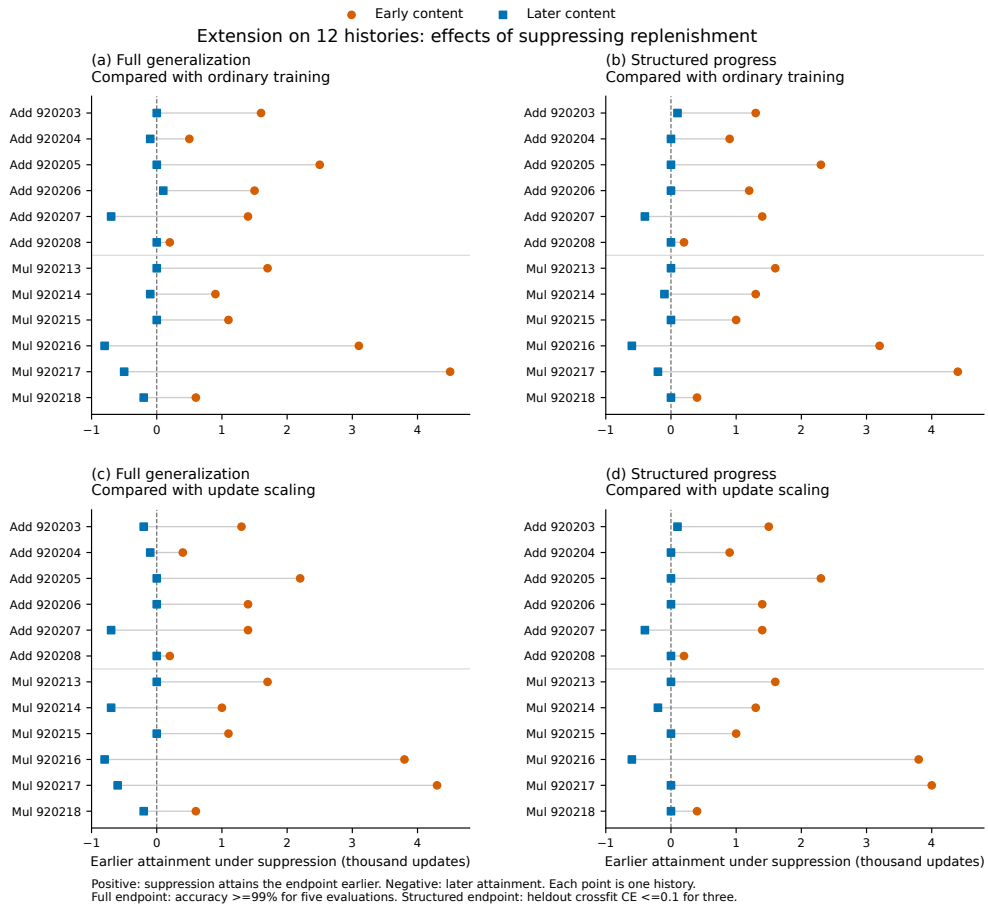

Figure 8: Suppressing replenishment on the twelve extension runs. Gain in thousands of updates from suppression, early content (circles) and later content (squares), against ordinary training (a, b) and against the scaling control (c, d), for full generalization (a, c) and structured attainment (b, d).

above. On the same twelve runs and both restored states, the *radial* arm removes, after each AdamW step in 2,101–4,100, the positive component of the adaptive update along the current radial direction of the whole operand-embedding parameter, $r = W_{\mathrm{dec}}/\|W_{\mathrm{dec}}\|$ with W_{dec} the decayed pre-step weight, and rescales what remains to the norm the directed suppression would leave at that state, $\|U - \max(\langle U, d\rangle, 0)\, d\|$; decay, moments and every other parameter are untouched, and training is ordinary to 12,100, read with the onsets of Appendix B.31 (24 continuations, none censored; the two pilot runs served only an engineering probe). The prespecified reading: with I_i the early-minus-later gain of directed suppression against ordinary training and J_i the same for the radial arm, $I_i - J_i > 0$ on both endpoints in at least 5 of 6 runs per task. *Fails*: 2 of 6 in each task. The radial arm advances full onset in every run for both contents and structured onset in every run but one (full onset: paired medians 1,750 and 850 updates for early and later content; structured: 1,300 and 200), so its early-minus-later gain, 1,000 and 1,150 updates, is close to the directed suppression's 1,500 and 1,300; $I_i - J_i$ has medians 300 and -50 and is positive in 7 and 4 of 12. The radial arm is also earlier than directed suppression for later content in 12 of 12 runs (median 950 updates for full onset) and for early content in 7 of 12 (median 700), and earlier than scaling in every run (Table 29). Two properties of the operator bear on the reading. It removes 1.7 to 3.5 times the dose the directed suppression removes and depletes the content along d more: at release the coordinate along d is a median 0.9 against 2.0 under directed suppression and 13.4 under ordinary training. And because the adaptive update after the fit is nearly radial, matching the norm rescales the remainder orthogonal to r by a mean factor of 14 to 24 per update (per-run maxima 34 to 119). The content dependence of the intervention gain is therefore not specific to the directed operator. What distinguishes the directed operator is that it removes only the positive component along one fixed direction and does not rescale the update in that direction's orthogonal complement; the radial arm's larger gains come with a rescaled remainder that this design does not separate from the removal of content.

Table 29: **Norm-matched radial control on the restoration cohort.** Paired medians over the 12 runs, in updates, positive when the first-named arm is earlier, with earlier / tied / later counts. $I_i - J_i$ is the directed minus the radial early-minus-later gain against ordinary training.

Content	Comparison	Full	Count	Structured	Count
Early	radial vs ordinary training	1,750	12/0/0	1,300	12/0/0
Early	directed suppression vs ordinary training	1,450	12/0/0	1,300	12/0/0
Early	radial vs directed suppression	700	7/2/3	200	8/0/4
Later	radial vs ordinary training	850	12/0/0	200	11/1/0
Later	directed suppression vs ordinary training	-50	1/5/6	0	1/7/4
Later	radial vs directed suppression	950	12/0/0	200	11/1/0
Both	$I_i - J_i$	300	7/1/4	-50	4/2/6

B.32 Suppressing the Rebuild in Ordinary Training

The replenishment tests above start from restored states, which exist only because a set was masked and then unmasked. This batch asks whether the same maintenance happens in ordinary training and how much of the gain from masking it accounts for. Sixteen fresh runs, ReLU addition seeds 923401–923408 and multiplication seeds 923411–923418, are trained normally to update 2,000 with every other setting of the one-layer recipe. A random half S of the frequencies is fixed by the usual rule before initialization. At update 2,000, D is the centered operand embedding projected onto S, its early content, and $d = D/\|D\|$ is fixed for every arm. The protocol, all rules and all code were frozen before any of these runs was initialized.

Arms. Twelve continuations per run, all from the same state at update 2,000 with its optimizer state and randomness: normal training; suppression, which for updates 2,001–4,000 removes the positive component of every AdamW update along d and leaves weight decay, negative components, other directions and the moments untouched; a scaling control, which shrinks the whole embedding update to the norm suppression would leave at the same state; an orthogonal control, which removes the same length along a fixed random direction in S orthogonal to d; the same three after a one-shot deletion of D at the fork, and deletion alone; masking S for updates 2,001–4,000 and masking it throughout; and suppression and scaling along the complementary half's own early content. Full generalization and the structured output of Appendix B.31 are read as before.

Readings fixed in advance. Primary: suppression attains both endpoints earlier than both normal training and scaling in at least 7 of 8 runs per task; after deletion, suppression attains both earlier than deletion alone and deletion with scaling under the same rule. Secondary: the complementary half under the same rule, and the orthogonal comparisons. Agreement of deletion with suppression and masking is descriptive, as the count of runs within 200 updates on both endpoints.

Outcome. No branch is censored or had attained before the fork. Both primary readings pass: suppression 7 of 8 addition and 7 of 8 multiplication, suppression after deletion 7 of 8 and 8 of 8 (Table 30). Suppression is earlier than the orthogonal control on both endpoints in 6 of 8 and 7 of 8, and never later; its strict rule fails on ties. After deletion, suppression is earlier than the orthogonal control in 7 of 8 and 8 of 8. The complementary half passes in addition only (8 of 8 and 3 of 8). Along d, the optimizer's updates add back a median 1.03 times what weight decay removes over updates 2,001–4,000 (range 0.79–1.08), and the early content at update 4,000 is a median 1.06 times its size at 2,000; weight decay alone would leave 0.135 of it. Deletion with suppression recovers a median 0.93 of the gain from masking (range 0.80–1.09), deletion alone 0.58 and suppression alone 0.31. It remains later than masking in 14 of 16 (median $+250$ updates) and within 200 updates of masking on both endpoints in 8 of 16. After deletion, the scaling control also helps (0.645 against 0.708 for deletion alone), and suppression beats it by a median 850 updates.

Table 30: **Replenishment in ordinary training, 16 fresh runs.** Median onsets of full generalization and of the structured output, the median ratio of total updates against normal training, and paired median differences from normal training (negative is earlier).

Arm	Full onset	Structured onset	Ratio	Paired, full / structured
Normal training	6,700	5,700	1.000	—
Scaling control	6,850	5,800	1.015	$+100$ / $+200$
Orthogonal control	6,650	5,600	1.000	0 / 0
Suppression along d	5,750	4,700	0.827	$-1{,}150$ / -900
Deletion	4,600	3,700	0.708	$-2{,}050$ / $-1{,}900$
Deletion, scaling	4,300	3,200	0.645	$-2{,}600$ / $-2{,}200$
Deletion, orthogonal	4,650	3,700	0.714	$-2{,}100$ / $-1{,}900$
Deletion, suppression	3,300	2,700	0.468	$-3{,}550$ / $-2{,}900$
Mask to 4,000	3,100	2,600	0.433	$-3{,}800$ / $-3{,}100$
Mask throughout	3,100	2,600	0.433	$-3{,}800$ / $-3{,}100$
Complement, scaling	7,000	6,000	1.033	$+200$ / $+200$
Complement, suppression	6,000	4,900	0.892	-650 / -300

The early content over ordinary training. A post-hoc reading of the 16 normal branches: the coordinate along d, relative to its value at 2,000, rises to a peak of 1.05 to 1.26 (median 1.15) between updates 2,700 and 3,900 and then declines. At the onset of generalization it stands at a median 0.56 of its initial value (range 0.37 to 0.86). In the 10 runs where it falls below one half, it does so within 400 updates of onset (median 200 updates after). Generalization thus arrives as the maintained content gives way; whether the decline precedes or follows the circuit's completion is not resolved by this reading.

Reading. In ordinary training the optimizer maintains the early content of a random half against weight decay, and suppressing that maintenance brings generalization forward. Most of the gain from masking the half is reproduced by deleting its early content once and keeping the optimizer from restoring it along d, with the rest of S open to new content; that agreement is descriptive and not complete. The failure of the complementary half in multiplication is unexplained. Which downstream computation the maintained content slows is not identified here.

B.33 Suppressing the Rebuild in Ordinary Training, Two Layers

The test of Appendix B.32 moved to the two-layer model without LayerNorm of Appendix B.3, on 16 fresh runs (ReLU addition seeds 953501–953508, multiplication 953511–953518), with the protocol, code and readings frozen before any run was initialized. Six branches from each state at

2,000: ordinary training; suppression along d for 2,001–4,000; the scaling and orthogonal controls; one-shot deletion of D with suppression; and masking S throughout. Endpoints as before; no branch was censored or had attained before the fork.

Readings and outcomes. R1, suppression earlier than ordinary training and the scaling control on both endpoints in at least 7 of 8 per task: *fails in addition* (6/8) and passes in multiplication (7/8); the paired median gain is 700 updates for full generalization and 600 for the structured output, and the scaling control itself changes nothing (-50). R2, deletion with suppression earlier than ordinary training on both endpoints in at least 7 of 8 per task: *passes* (8/8 and 8/8), median 3,200 updates earlier, recovering a median 0.91 of the gain from masking (range 0.70–1.10). R3, the maintenance ratio in ordinary training: median 0.894 (range 0.498–0.963), below the one-layer 1.03; at depth two the early content is replenished at about nine tenths of its decay and declines slowly. R4, suppression earlier than the orthogonal control on both endpoints in at least 7 of 8 per task: *fails* (6/8 and 6/8); the orthogonal control changes nothing (paired 0).

Table 31: **Replenishment in ordinary training at depth two, 16 fresh runs.** Median onsets, the median ratio of total updates against ordinary training, and paired median differences from ordinary training.

Branch	Full onset	Structured onset	Ratio	Paired, full / structured
Ordinary training	6,750	5,900	1.000	—
Scaling control	6,600	5,900	0.993	−50 / 0
Orthogonal control	6,800	6,200	1.000	0 / 0
Suppression	5,450	5,000	0.887	−700 / −600
Deletion, suppression	3,500	2,800	0.512	−3,200 / −3,200
Masking	3,200	2,600	0.467	−3,600 / −3,400

Reading. At depth two the robust result transfers: deleting the early content and keeping the optimizer from restoring it reproduces most of the gain from masking in every run. Suppression alone is earlier than ordinary training on both endpoints in 13 of 16 (15 of 16 for full generalization), by a median 700 updates, but misses the 7 of 8 rule in addition and does not beat the orthogonal control under the rule. Ordinary training at this depth maintains the early content less completely than at depth one.

Extension: additional benefit after the same deletion. After the original six-branch outcomes were known, we added deletion alone and deletion followed by the original scaling control on all sixteen checkpoints, 32 continuations. The deletion, saved direction, optimizer state, random state, split and intervention window (2,001–4,000) are unchanged. The budget remains update 40,000, with stopping allowed after both endpoints are confirmed and update 5,000 has been reached. All 32 continuations completed without censoring. This is an extension on known runs, not a fresh-seed confirmation, and it introduces no pass criterion.

For each run we compute $\Delta^{\text{delete}} = T_{\text{delete}} - T_{\text{delete+suppression}}$ and $\Delta^{\text{scale}} = T_{\text{delete+scale}} - T_{\text{delete+suppression}}$ separately for full and structured onset. Positive values favor suppression. Intervals are 95% percentile bootstrap intervals for the median of within-run differences (20,000 resamples, seed 20260925), resampling runs within each task before pooling. They are descriptive intervals on the reused cohort and are outside the original Holm family. Censored differences would remain unknown, without imputation. Table 32 also gives the same post-hoc intervals for the original one-layer comparisons; their prespecified rules are unchanged.

At depth two, suppression advances full onset beyond deletion in 13/16 runs and beyond deletion with scaling in 12/16. Both endpoints are earlier than both controls in 6/8 runs per task. Thus the additional benefit has a positive paired median at both depths, with more variation in the two-layer extension. The original suppression-alone failures and combined-intervention criterion reported above remain unchanged. Per-run differences appear in Table 33; the artifact also supplies every onset, confirmation time and task-specific interval.

The same extension was then run on the weight-decay-0.5 and modulus-97 cohorts (Appendices B.34 and B.35; one layer, 32 continuations each), with the protocol above copied unchanged and written

Table 32: **Additional advance from suppression after deletion.** 1L and 2L fresh: fresh-seed cohorts with the rule fixed in advance; 2L ext., wd 0.5 and p=97: extensions on existing runs. Counts are earlier / tied / later out of 16. Advances and 95% paired bootstrap intervals are in updates; positive values favor deletion with suppression.

	Comparator	Full count	Full advance [95% CI]	Structured count	Structured advance [95% CI]
1L	Delete	16/0/0	1,200 [900, 2,000]	15/1/0	1,000 [400, 1,600]
1L	Delete + scale	16/0/0	850 [550, 1,100]	15/1/0	500 [200, 1,000]
1L	Delete + orthogonal	16/0/0	1,250 [900, 1,800]	15/1/0	1,000 [400, 1,600]
2L ext.	Delete	13/1/2	850 [300, 1,400]	13/1/2	700 [400, 1,400]
2L ext.	Delete + scale	12/1/3	550 [200, 900]	12/2/2	400 [200, 1,000]
2L fresh	Delete	15/1/0	1,250 [300, 1,700]	15/1/0	1,000 [600, 1,600]
2L fresh	Delete + scale	15/1/0	800 [200, 900]	15/1/0	700 [300, 1,100]
wd 0.5	Delete	16/0/0	2,300 [1,800, 3,000]	16/0/0	2,400 [2,000, 3,500]
wd 0.5	Delete + scale	16/0/0	1,800 [1,250, 2,400]	16/0/0	1,700 [1,500, 2,400]
p=97	Delete	15/0/1	1,350 [800, 1,900]	16/0/0	1,300 [500, 1,400]
p=97	Delete + scale	15/0/1	750 [500, 1,100]	15/1/0	600 [400, 1,200]

after those cohorts' six-branch outcomes were known. Before any new branch ran, exact replays of the original ordinary and deletion-with-suppression branches through update 2,100 matched, and the reused deletion-with-suppression onsets equal the recorded ones for both onsets in all 16 runs per cohort; no onset is censored. At weight decay 0.5, suppression advances full onset beyond deletion alone and beyond deletion with scaling in 16/16 runs each (paired medians 2,300 and 1,800 updates), and both onsets are ahead of both controls in 8/8 runs per task. At modulus 97 the counts are 15/16 and 15/16 (1,350 and 750 updates), with both onsets ahead of both controls in 8/8 addition and 7/8 multiplication runs. All four full-onset intervals exclude zero (Table 32).

Onset reading (post hoc). Because normal-training onsets are sensitive to small perturbations (Appendix B.55), we re-read full onset for every branch of these shared-deletion comparisons from the saved trajectories under four readings fixed and hashed before any value was computed: test accuracy of at least 0.98 or 0.99, held for three or five consecutive evaluations. The original reading (0.99, five) reproduces all 192 recorded onsets. Under all four readings the number of runs in which suppression is earlier is unchanged (16 and 16 at one layer, 13 and 12 at two layers, 16 and 16 at weight decay 0.5, 15 and 15 at modulus 97, against deletion alone and against deletion with scaling), and the pooled paired medians stay within 1,200 and 850–900 updates at one layer, 850 and 550–800 at two layers, 2,300–2,450 and 1,600–1,800 at weight decay 0.5, and 1,150–1,350 and 600–850 at modulus 97. At two layers, leaving out any one run keeps the medians at 800–900 and 400–700 updates; against deletion with scaling the two tasks differ (750 updates in addition, 350 in multiplication), so the two-layer effect is not uniform across tasks. The original definition and all original rules are unchanged.

Fresh-seed confirmation at two layers. Because the two-layer extension above fell short of the rule used at one layer (6/8 per task) and had shaped the question, we repeated the comparison on 16 two-layer runs whose seeds had never been trained (addition 993501–993508, multiplication 993511–993518). It was run last, after every other experiment in this paper, and was added to this submission after its outcome was known. The protocol, hashed before any model with these seeds was created, fixes four continuations from each run's update-2,000 state (ordinary training, deletion alone, deletion with the scaling control, and deletion with suppression over 2,001–4,000), the budget and onsets above, and a single rule: in at least 7 of 8 runs per task, deletion with suppression reaches both onsets earlier than both controls, with ties and censored suppression onsets counting against it. Before any cohort model was created, two seeds outside the cohort replayed the runner exactly, gave identical deleted states across the three deletion arms and unchanged optimizer states across all four arms, and matched the scaling control's norm to within 10^{-7}. All 64 continuations completed without censoring. The rule holds in 7/8 addition runs, the remaining run tying both controls on both onsets, and in 8/8 multiplication runs. Suppression advances full onset by paired medians of 1,250 updates over deletion alone and 800 over deletion with scaling, earlier in 15/16 runs and tied in the other for each (1,250 and 800 in addition, 1,100 and 850 in multiplication; Table 32). Onsets use the original reading only. The extension on existing runs and its 6/8 counts stand as reported above.

Table 33: **Every two-layer paired difference.** Positive values mean earlier onset under deletion with suppression.

	Full onset		Structured onset	
Task / seed	Δ^{delete}	Δ^{scale}	Δ^{delete}	Δ^{scale}
add / 953501	+700	+400	+400	+200
add / 953502	+300	+700	+400	+200
add / 953503	+2,300	+1,000	+2,200	+1,000
add / 953504	-500	-500	-400	-400
add / 953505	+0	+0	-200	-200
add / 953506	+2,700	+1,100	+2,600	+1,000
add / 953507	+1,300	+800	+1,400	+1,000
add / 953508	+2,400	+1,700	+2,400	+1,800
mul / 953511	+1,400	+900	+1,200	+1,000
mul / 953512	+300	-100	+600	+0
mul / 953513	+1,000	+300	+1,400	+600
mul / 953514	-500	-600	+0	+0
mul / 953515	+800	+400	+800	+400
mul / 953516	+900	+800	+600	+400
mul / 953517	+500	+300	+400	+200
mul / 953518	+2,300	+1,700	+2,200	+1,000

Execution audit. Before dispatch, two independent continuations reproduced the original runner's model and optimizer states exactly. All deletion arms began from identical deleted weights and unmodified moments. The release boundaries and local norm match were checked. After completion, an independent analysis reconstructed both endpoints from the prefix and saved trajectories for all five compared two-layer arms (160 endpoint checks), verified source and artifact hashes and checked each new branch's coordinate accounting. No original experiment file was modified.

B.34 Suppressing the Rebuild in Ordinary Training at Weight Decay 0.5

The test of Appendix B.32 at AdamW weight decay 0.5, the second weight decay at which the restoration and content-age results were confirmed (Appendices B.23 and B.25), on 16 fresh runs (ReLU addition seeds 963501–963508, multiplication 963511–963518; one layer, all else as in the natural program), with the same six branches from each state at 2,000, the same readings, and protocol and code frozen before any run was initialized (engineering admission on two smoke runs at weight decay 0.5 that are not part of the cohort). No branch was censored or had attained before the fork. Onsets are later at this weight decay (ordinary training a median 13,250 and 12,450 updates).

Readings and outcomes. R1, suppression earlier than ordinary training and than the scaling control on both onsets in at least 7 of 8 per task: *fails*, 7/8 and 5/8; on full generalization alone suppression is earlier than ordinary training in 7/8 and 8/8 (paired medians $-1{,}300$ and -800). R2, deletion with suppression earlier than ordinary training on both onsets in at least 7 of 8: *met*, 8/8 and 8/8 (paired medians $-6{,}850$ and $-6{,}250$); it recovers a median 0.92 and 0.88 of the masking gain. R3 (no rule), the maintenance ratio in ordinary training: 1.21 and 1.22 (range 1.11–1.25), against 1.03 at weight decay 1 and 0.89 at depth two; at the weaker decay the optimizer adds back more than decay removes and the early content grows. R4, suppression earlier than the orthogonal control on both onsets in at least 7 of 8: *fails*, 7/8 and 5/8 (paired medians $-1{,}000$ and -400).

Reading. As at depth two, deletion with suppression is the robust result (16 of 16, 0.9 of the masking gain) and suppression alone misses the joint rule in one task; the maintenance itself is stronger at the weaker decay. The mechanism's operator therefore holds at a second weight decay, and the Limitations' scope statement is narrowed accordingly.

B.35 Suppressing the Rebuild in Ordinary Training at Modulus 97

The test of Appendix B.32 at the second modulus, $p = 97$, at which so far only the main masking result was confirmed (Appendix B.2), on 16 fresh runs (ReLU addition seeds 973501–973508,

Table 34: **Replenishment suppression at weight decay 0.5, 16 fresh runs.** Median onsets of full generalization and of the structured output (addition / multiplication) and the paired median difference from ordinary training on full generalization.

Branch	Full onset	Structured onset	Paired
Ordinary training	13,250 / 12,450	11,000 / 10,000	—
Suppression	12,100 / 11,300	9,400 / 9,700	$-1{,}300$ / -800
Scaling control	13,050 / 12,550	10,500 / 10,000	
Orthogonal control	13,100 / 12,250	10,600 / 10,200	
Deletion with suppression	5,750 / 6,650	3,500 / 4,200	$-6{,}850$ / $-6{,}250$
Masking	5,150 / 5,850	3,000 / 3,200	

multiplication 973511–973518; one layer, weight decay 1, all else as in the natural program; the random half is 24 of the 48 frequencies, drawn by the same rule), with the same six branches from each state at 2,000, the same readings, and protocol and code frozen before any run was initialized (engineering admission on two smoke runs at $p = 97$ that are not part of the cohort). No branch was censored or had attained before the fork.

Readings and outcomes. R1, suppression earlier than ordinary training and than the scaling control on both onsets in at least 7 of 8 per task: *fails*, 6/8 and 8/8; on full generalization alone suppression is earlier than ordinary training in 6/8 and 8/8 (paired medians -800 and $-1{,}450$). R2, deletion with suppression earlier than ordinary training on both onsets in at least 7 of 8: *met*, 8/8 and 8/8 (paired medians $-4{,}000$ and $-2{,}350$); it recovers a median 0.88 and 0.93 of the masking gain. R3 (no rule), the maintenance ratio: 1.08 and 1.06 (range 1.00–1.11), against 1.03 at $p = 113$. R4, suppression earlier than the orthogonal control on both onsets in at least 7 of 8: *fails*, 6/8 and 7/8 (paired medians -950 and $-1{,}100$).

Table 35: **Replenishment suppression at modulus 97, 16 fresh runs.** Median onsets of full generalization (addition / multiplication) and the paired median difference from ordinary training.

Branch	Full onset	Paired
Ordinary training	8,950 / 7,700	—
Suppression	7,800 / 5,750	-800 / $-1{,}450$
Scaling control	8,950 / 7,750	
Deletion with suppression	4,100 / 3,650	$-4{,}000$ / $-2{,}350$
Masking	3,350 / 3,100	

Reading. At the second modulus, as at depth two and at weight decay 0.5, deletion with suppression is the robust result (16 of 16, 0.9 of the masking gain) and suppression alone misses the joint rule in one task. The mechanism's operator holds at both moduli.

B.36 Gradient Components during the Rebuild and Suppression

A post-hoc reading on the 16 runs of Appendix B.32, made on their saved checkpoints with the actual training gradient; nothing was trained or edited. Its prospective version is the batch described at the end of this section.

The deleted content is rebuilt. After the one-shot deletion of the early content D at update 2,000, with S left open, the coordinate along d returns to a median 0.27 of its initial size by 2,100, 0.53 by 2,600 and 0.56 by 3,000; with suppression after deletion it stays at 0.00. Under suppression alone it decays to 0.14 by 4,000 and, once the corrections end, regrows to 0.30 by 4,400 and 0.35 by 4,800; after deletion with suppression it stays at 0.00 to 0.01 (Table 30 gives the endpoints).

The content is read diffusely. D carries a median 0.494 of the centered operand-embedding energy. Every MLP neuron reads it at about that share of its input variance over tokens (median 0.491; 10th to 90th percentile 0.402 to 0.586), and every attention head's query, key and value maps read it at

0.4 to 0.6. In model space D needs 25 of the 128 directions for 90% of its energy and 43 for 99%. No neuron or head reads it preferentially, which is the isotropic null, and there is no low-rank reader subspace; module-specific mediation tests had nothing to isolate.

Gradient components change during suppression. Table 36 gives the share of the centered embedding gradient along d (the descent component that would add the content back) and the share in the five frequencies that carry most of the final normal model's embedding energy. In ordinary training the gradient keeps a 0.6 share along d for the whole window. Under suppression the share rises to 0.85 while the content decays, then falls as the structured output improves. After deletion with suppression it falls to 0.02 by 4,000 while the key-frequency share rises to 0.79. Where the corrections end at 4,000 the content regrows exactly where the share is still high: 0.46 in the suppression arm, 0.02 after deletion with suppression. The structured output halves (its cross-entropy first falls to half its value at the fork) a median 2,000 updates after the fork in ordinary training and 200 updates after it under masking or deletion with suppression; from that halving to full generalization takes a median 2,800, 900 and 1,100 updates respectively.

Table 36: **Share of the embedding gradient along the early direction d and in the final solution's key frequencies.** Medians over 16 runs, computed on the saved checkpoints with the actual training gradient.

Branch, along d / key frequencies	2,100	2,200	2,600	3,100	3,600	4,000
Normal training	0.59 / 0.44	0.63 / 0.47	0.63 / 0.50	0.62 / 0.52	0.58 / 0.53	0.58 / 0.57
Suppression	0.79 / 0.44	0.84 / 0.44	0.85 / 0.44	0.76 / 0.48	0.61 / 0.53	0.46 / 0.59
Deletion	0.81 / 0.47	0.76 / 0.50	0.55 / 0.59	0.45 / 0.65	0.34 / 0.69	0.33 / 0.71
Deletion, suppression	0.77 / 0.46	0.66 / 0.50	0.33 / 0.65	0.11 / 0.73	0.04 / 0.74	0.02 / 0.79

Reading, with two caveats. Near equilibrium the adaptive update along d balances decay. With roughly isotropic preconditioning, a large raw-gradient share along d is therefore expected from the content's energy share (0.49); the observed share near 0.6 is not independent evidence for competition. The shares are normalized and their subspaces overlap (d and the key frequencies are not orthogonal; 0.59 + 0.44 exceeds 1), so they do not partition the gradient. After deletion, a share of 0.81 still points along the missing content. With replenishment suppressed, that share falls to 0.02 as the training fit recovers, and the structured-output cross-entropy halves 200 updates after the fork instead of 2,000. These observations are consistent with competition between rebuilding early content and learning an alternative fit. They do not identify the downstream features involved or show that the structured readout mediates the behavioral delay.

The actual step, in absolute terms. Shares can rise because something else fell, so Table 37 reports one actual AdamW step taken from each saved checkpoint, weight decay included: its signed component along d and the norm of its component in the key frequencies, in parameter units. In ordinary training the step nets $+2.5 \times 10^{-3}$ along d at 2,100 to 2,600 against a decay part of -1.4×10^{-2}, and about zero from 3,100 on: maintenance. Under suppression the uncorrected step would add 1.1×10^{-2} to 2.5×10^{-2} along d (the rebuild the correction removes), and its key-frequency component is two to three times that of ordinary training at the same update (1.18×10^{-2} against 4.3×10^{-3} at 2,600). After deletion the step rebuilds d at 1.3×10^{-2} and returns to about zero by 3,100 once the content is back; after deletion with suppression the wanted rebuild falls from 2.8×10^{-2} to 4×10^{-4} by 4,000. Steps are larger overall where the training loss is higher, so the key-frequency component is read beside the step norm; the reading is descriptive.

A prospective batch on 16 further fresh runs, frozen before launch, tests the reading with the duration of suppression, with release at three times against the gradient share measured there, and with a clamp that holds the early content in place (Appendix B.37).

B.37 Suppression Duration, Release and Clamp: A Prospective Test of the Routing Reading

Sixteen fresh runs (ReLU addition seeds 923501–923508, multiplication 923511–923518), one-layer recipe, protocol and code frozen before any run was initialized; the readings were fixed from the

Table 37: **One actual AdamW step from each checkpoint, medians over 16 runs.** Signed component along d, norm of the component in the final solution's five key frequencies, and total norm of the centered operand embedding step, in parameter units.

Branch	Update	Along d ($\times 10^{-3}$)	Key frequencies ($\times 10^{-3}$)	Step norm ($\times 10^{-3}$)
Normal training	2,100 / 2,600 / 4,000	$+2.5$ / $+2.5$ / -0.3	4.6 / 4.3 / 4.5	13.9 / 8.9 / 9.8
Suppression (uncorrected step)	2,100 / 2,600 / 4,000	$+11.3$ / $+25.1$ / $+8.0$	6.7 / 11.8 / 8.2	15.6 / 26.3 / 16.0
Deletion	2,100 / 2,600 / 4,000	$+12.8$ / $+3.0$ / $+1.1$	7.4 / 4.2 / 7.9	17.6 / 8.4 / 13.3
Deletion, suppression	2,100 / 2,600 / 4,000	$+27.5$ / $+4.9$ / $+0.4$	14.9 / 4.3 / 7.3	35.8 / 10.4 / 12.7

post-hoc analysis of Appendix B.36. Nine branches from each state at 2,000: ordinary training; suppression along d until 3,000, 4,000, 6,000 or throughout; one-shot deletion of D with suppression until 4,000; and three clamps until 5,000 that reset, after every update, the coordinate along d to its initial value, to half of it, or the coordinate along the orthogonal direction o to its own initial value. At each release (3,000, 4,000, 6,000, and 4,000 after deletion) the gradient share along d was measured on the saved checkpoint, and regrowth was read as a rise of the coordinate by more than 0.10 of its initial value within 400 updates. No branch was censored or had attained before the fork.

Readings and outcomes. R1, duration: longer suppression earlier in at least 7 of 8 per task, for throughout against 4,000 and for 4,000 against 3,000. *Fails*: 5/8 and 6/8, then 4/8 and 6/8, with 5 and 4 ties; beyond 4,000 the medians barely move (Table 38). R2, release: regrowth at every release with share above 0.30 and at none below 0.10, at most one violation per class. *The low class passes*: 0 of 17 releases with share below 0.10 regrew, over 8 addition and 6 multiplication runs. *The high class fails the count*: 27 of 32 regrew, with 5 violations at shares 0.30–0.43 and rises 0.03–0.09. Regrowth follows the share by branch (share 0.78: 15/16; 0.48: 11/16; 0.21: 4/16; 0.03: 0/16) and across the 64 releases (Spearman 0.86). R3, clamp: holding d at its initial value delays generalization by more than 200 updates against ordinary training and against the o clamp in at least 7 of 8 per task. *Fails*: 3/8 and 2/8, and 4/8 and 2/8. The clamp did not bind: ordinary training holds the coordinate at 1.0 to 1.15 of its initial value until shortly before onset, so fixing it there until 5,000 changed little at a median onset of 6,950. Holding it at half brought generalization forward instead (paired -450); the o clamp changed nothing. R4: suppression throughout earlier than ordinary training in at least 7 of 8 per task. *Narrowly fails*: 7/8 and 6/8.

Table 38: **Suppression duration and clamps, 16 fresh runs.** Median onsets of full generalization and of the structured output, the median ratio of total updates against ordinary training, and the paired median difference from ordinary training (full onset).

Branch	Full onset	Structured onset	Ratio	Paired
Ordinary training	6,950	5,900	1.000	—
Suppression to 3,000	6,150	5,200	0.918	-400
Suppression to 4,000	5,650	4,600	0.907	-400
Suppression to 6,000	5,550	4,500	0.889	-500
Suppression throughout	5,550	4,500	0.889	-500
Deletion, suppression to 4,000	4,050	2,800	0.553	$-2{,}700$
Clamp d at 1.0 to 5,000	6,800	5,800	1.010	$+50$
Clamp d at 0.5 to 5,000	5,850	4,600	0.887	-450
Clamp o to 5,000	6,750	5,800	1.000	0

Reading. Suppression during updates 2,001–4,000 is earlier than ordinary training in 12 of these 16 runs (median 400 updates), against 15 of 16 and 1,150 in the first cohort; pooled, 27 of 32 runs, median 800. Deletion with suppression is earlier on both endpoints in 16 of 16 here and 32 of 32 pooled (median 3,450). The gradient share along d at release predicts regrowth as a dose–response, and no release with a low share regrew; the strict high-class count failed on five borderline releases. Longer suppression adds little beyond 4,000, and the clamp as designed could not test the converse; the next subsection does.

B.38 Holding the Early Content in Place: A Test of Sufficiency

Sixteen fresh runs (ReLU addition seeds 923601–923608, multiplication 923611–923618), one-layer recipe, protocol and code frozen before any run was initialized. The clamp above ended at 5,000, before ordinary training lets the early content decline; here the hold lasts to 12,000, spanning the period of decline in ordinary training. Six branches from each state at 2,000: ordinary training; a one-sided *floor* that, after every update through 12,000, raises the coordinate along d back to its value at 2,000 whenever the update has left it lower, and never lowers it; the two-sided clamp of the earlier design, to 12,000 and to 8,000; the same two-sided clamp on the orthogonal direction o; and suppression along d throughout. The floor was added, and made the primary arm, after an engineering replay on two old runs showed that the two-sided clamp spends its first thousands of updates removing content, because ordinary training first raises the coordinate above its fork value (to 1.16 in one replay) and only later lets it fall; the amendment is dated in the protocol before any new seed was initialized. No branch was censored or had attained before the fork.

Readings and outcomes. R1, sufficiency: the floor later than ordinary training by more than 200 updates, and later than the o clamp, in at least 7 of 8 per task. *Fails*: 5/8 and 6/8 against ordinary training (6/8 and 8/8 against the o clamp), paired medians $+350$ and $+500$; in 7/8 and 6/8 runs the onset fell inside the hold. The two-sided clamp: 5/8 and 3/8 ($+300$, $+150$). R2, binding: the content the hold put back over 5,001–12,000 exceeds 0.25 of the fork coordinate in at least 7 of 8 per task. *Met, 16/16*: 1.6 to 5.8 times the fork coordinate, acting on 3,164 to 7,000 of the 7,000 updates; in the ordinary branches the coordinate had fallen to 0.36–0.74 of its fork value by their onset. R3, release at 8,000 (no rule): 4 and 2 runs had attained before 8,000; the others attained 500, 600, 600 and 6,000, and 100, 100, 1,800, 1,900, 4,200 and 4,400 updates after release. R4: suppression throughout earlier than ordinary training in at least 7 of 8 per task. *Met*: 8/8 and 8/8, paired medians $-1{,}900$ and $-2{,}050$. R5, control: the o clamp later than ordinary training in at most 2 of 8 per task. *Met*: 1/8 and 1/8.

Table 39: **Holding the early content in place, 16 fresh runs.** Median onsets of full generalization and of the structured output and the paired median difference from ordinary training (full onset).

Branch	Full onset	Structured onset	Paired
Ordinary training	8,200	7,000	—
Floor on d at the fork value to 12,000	8,850	7,100	$+400$
Clamp d at the fork value to 12,000	8,350	6,900	$+250$
Clamp d at the fork value to 8,000	8,300	6,900	$+100$
Clamp o to 12,000	8,300	7,200	0
Suppression throughout	6,200	5,100	$-2{,}000$

Reading. The binding check passes in all 16 runs and 13 generalize while the floor remains active. Decline of this fixed early coordinate is therefore not necessary for generalization. The floor adds paired median delays of 350 and 500 updates (11/16 later by more than 200), missing R1's prespecified sufficiency criterion. This incremental comparison with ordinary training does not show that retaining the content has no causal cost. Nor does it isolate maintenance from occupancy at matched content trajectories or determine whether the natural decline causes or follows generalization. The causal contribution of replenishment is supported by the suppression interventions: the 2,001–4,000 window advances full onset in 15/16 and 12/16 runs of the first two fresh cohorts; continuous suppression advances it in 16/16 here. The frozen protocol's stronger interpretation that the natural decline is mostly a consequence of generalization is not established by these measurements.

At depth two. The same hold on the two-layer cohort of Appendix B.33 (16 runs; ordinary training and the one-sided floor along d through 12,000 from each state at 2,000; protocol hashed before any run; no branch censored). R1, the floor later than ordinary training by more than 200 in at least 7 of 8 per task: *fails*, 4/8 and 6/8, paired medians $+250$ and $+550$ (onsets 6,400 to 6,950 and 6,900 to 7,500; on the structured onset 0 and $+500$); in 7/8 and 8/8 runs the onset fell inside the hold. R2, binding: *met*, 16/16; the floor put back 4.7 to 9.0 times the fork coordinate and acted on 8,931 to 10,000 of the 10,000 updates, while in the ordinary branches the coordinate had fallen to 0.19–0.58 of its fork value by their onset. The content's presence does not block the circuit at depth two either.

B.39 WHY THE EARLY CONTENT IS MAINTAINED

Measurement-only readings on the frozen states of the same sixteen runs (the source at 2,000 and the ordinary branch at 4,000 and 5,000), taken after the batch completed. The pattern was found first on the sixteen runs of Appendix B.32, at 2,000 and at their saved states to 4,000, and the two readings were fixed before the fresh states were measured. No training step is taken.

Dependence of training loss on the content. The early content supports the training fit: deleting it along d at 2,000 raises the training cross-entropy by 6.2 to 11.4 nats in all 16 fresh runs (R1, rule 15 of 16; first cohort 4.4 to 14; two layers 3.7 to 7.4), and by a median 6.2 and 5.3 nats at 4,000 and 5,000.

Constant-gradient illustration. For a coordinate with nearly constant gradient magnitude and sign and equilibrated moments, a local balance of update and decoupled decay gives

$$|\theta^*_{\mathrm{SGD}}| = \frac{|g|}{\lambda}, \qquad |\theta^*_{\mathrm{AdamW}}| = \frac{|s|}{\lambda}, \qquad s = \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} \approx \frac{\bar{g}}{|g| + \epsilon}. \tag{3}$$

Here $\bar{g}$ is the mean gradient and $|g|$ its approximately constant size. The balance is known in the exponential-moving-average form of Wang & Aitchison (2025), and Kosson et al. (2024) derive a related norm equilibrium for noise-dominated updates; it illustrates why an adaptive step can oppose decay even when the raw gradient is small. It does not predict a changing coordinate along d, whose constituent moments and gradients evolve. The measured update projections establish maintenance along the observed trajectory; they do not validate the constant-gradient approximation over training.

Raw-gradient and decay steps at the same learning rate. Let $c = \langle E, d\rangle > 0$ and $g_d = \langle \nabla_E L, d\rangle$ at a saved state. For plain SGD without momentum, learning rate η and weight-decay coefficient λ,

$$\Delta c_{\mathrm{SGD}} = -\eta g_d - \eta\lambda c, \qquad \rho = \frac{-\eta g_d}{\eta\lambda c} = \frac{-g_d}{\lambda c}. \tag{4}$$

The learning rate scales both terms, so increasing it does not change their ratio at that state. This is the no-momentum case of PyTorch's SGD update.[1] At update 2,000, with $\eta = 10^{-3}$ and $\lambda = 1$, the decay displacement along d is $(1.4\text{–}1.5) \times 10^{-2}$. The raw gradient replenishes the content ($g_d < 0$) in 15/16 fresh states; in those states ρ ranges from 1.5×10^{-8} to 2.2×10^{-5}. The raw gradient step is therefore about 4.5×10^4 to 6.5×10^7 times smaller than the decay step. In the remaining state it points toward further depletion. These are local comparisons at the saved AdamW states, not SGD training experiments.

The frozen R2 calculation used $\ell_{\mathrm{fixed}} = \eta_{\mathrm{A}}\lambda c/(-g_d)$, with $\eta_{\mathrm{A}} = 10^{-3}$ fixed at the original AdamW rate. Its recorded range is 45–65,000 and its threshold $\ell_{\mathrm{fixed}} > 1$ is met in 15/16 states (rule: 14/16). This quantity multiplies the raw gradient to match the *fixed original* decay displacement; it is not the learning rate that balances ordinary SGD with unchanged λ. The comparison in Eq. 4 is the algebraic restatement $\rho = \eta_{\mathrm{A}}/\ell_{\mathrm{fixed}}$, with the original statistic and gate retained. AdamW's measured adaptive updates, in contrast, offset decay (Appendix B.32), consistent with its normalization sustaining content despite a small raw gradient.

Descriptively, the raw gradient reaching the frequencies outside S is 10^2 to 10^7 times larger under the mask than without it at the same weights (median 3.9×10^6), while in ordinary training the structured output falls about as much per 200-update window when the training loss is saturated (below 10^{-4}; median fall 0.22) as when it is not (0.19): the share of its total fall in unsaturated windows (median 0.47) matches their share of windows (0.56). Improvement in this structured-output readout is therefore not confined to loss spikes; it does not directly measure the construction of an internal circuit.

Precision audit (post hoc). The training losses, error signals and raw gradients above are small enough that a float32 loss computation could distort them (Prieto et al., 2025). We therefore recomputed them at the 48 states of this appendix (the 16 runs at 2,000, 4,000 and 5,000) with the whole model copied to float64 and the loss and backward pass taken in float64, on CPU, with the

[1] https://docs.pytorch.org/docs/2.14/generated/torch.optim.SGD.html

content kept and with the content along d deleted. Float32 does return a softmax of exactly one for the correct class on some training pairs: in 41 of 48 states, for a median 75 and at most 1,286 of the 3,830 pairs, against at most 2 pairs in 4 states under float64. The aggregate quantities move little: relative to float64, the float32 training loss differs by at most 0.096, the norm of the error signal by at most 0.032, the raw embedding gradient's norm by at most 0.22 and its component along d by at most 0.18, and the cosine between the float32 and float64 gradients is 0.89 to 1.00 (median 0.98); at the 17 unsaturated states every difference is below 3×10^{-4}. The contrasts the argument uses are unchanged: deleting the content raises the error signal's norm by a factor of 10 to 2.6×10^{6} in either precision, and a plain gradient step along d would return at most 2.7×10^{-4} of what decay removes at any of the 48 states in either precision. The audit covers saved states and the loss computation, not a float64 training run.

B.40 When Removal Acts: A Removal-Time Sweep

The same sixteen runs, forked from their frozen states at 2,000, 4,000 and 5,000 (the ordinary branch's saved states; a continuation from a saved state reproduces the branch bitwise, verified in admission and again by the reference arms' onsets). Three branches per fork: masking S from the fork on; one-shot deletion of the content along d at the fork with suppression of its return; and ordinary training as the reference. Protocol, readings and code were hashed before any branch ran; the reference arms and the gradient and motion records were added before any branch ran as well. No branch was censored. A run qualifies at a fork if ordinary training had not attained by it and its onset lies at least 1,000 updates later: 8/8 per task at 2,000 and 4,000, 6 and 7 at 5,000. For a run and fork T the *lag* is the arm's onset minus T, the *wait* the ordinary onset minus T, and the gain their difference.

Readings and outcomes. R1, change in the masked lag between the 2,000 and 5,000 forks: the protocol defines three regions, unchanged, shrinking by more than 300, or a lag at 5,000 below 300. It associates these with construction after removal, construction under the content, and interference, respectively; these are candidate interpretations of onset times. *Mixed*: the lag shrinks by -200, -200, -400, -500, -500, -300 in addition and -400, -200, -100, $-1{,}200$, -600, -200, -100 in multiplication (medians -350 and -200); the unchanged region counts 3/6 and 4/7, the shrinking region 3/6 and 3/7, interference 0. The median lag reductions are 200–350 updates over 3,000 additional updates of ordinary training. Converting this comparison into a circuit construction rate would require an unestablished common measure of functional progress. R2, late removal still helps: masking at 5,000 earlier than ordinary training by more than 200 in at least 7 of 8 qualifying runs per task, all but one when fewer qualify. *Met*: 6/6 and 7/7. R3 (no rule): deletion with suppression recovers a median 0.89 and 0.92 of the masking gain at 2,000, 0.91 and 0.89 at 4,000, 0.90 and 0.86 at 5,000; the coordinate along d at 4,000 and 5,000 is 0.90–1.17 of its fork value (this cohort's ordinary onsets are at a median 8,200, so the decline has not begun). R5, straightness (the motion reading): per 100-update window, the net displacement of the embedding outside S over the summed per-update displacements, masked branch against ordinary training at matched windows before either onset; higher under the mask in at least 14 of 16 at the 2,000 fork. *Met by count, negligible in size*: 16/16, but the medians are 0.987 against 0.974. The motion outside S is nearly straight with or without the content. R6 (descriptive): the cosine between consecutive raw gradients outside S is higher under the mask in 15/16.

Table 40: **Removal-time sweep, 16 runs.** Median lag after removal, median wait remaining in ordinary training, and median fraction of the wait saved, by fork and arm (addition / multiplication; full onset).

Fork	Arm	Lag	Wait	Saved
2,000	mask	1,350 / 1,300	6,000 / 7,000	0.76 / 0.80
2,000	delete, suppress	1,850 / 2,000	6,000 / 7,000	0.67 / 0.70
4,000	mask	850 / 1,050	4,000 / 5,000	0.71 / 0.78
4,000	delete, suppress	1,200 / 1,700	4,000 / 5,000	0.70 / 0.67
5,000	mask	900 / 1,100	3,450 / 5,000	0.76 / 0.73
5,000	delete, suppress	1,100 / 1,400	3,450 / 5,000	0.67 / 0.65

Reading. Interventions remain effective late in training: masking at 5,000 saves about three quarters of the remaining wait, and deletion with suppression recovers 0.86–0.92 of the masking gain across forks. The relatively small reduction in lag is consistent with limited progress toward rapid generalization before removal, but onset times alone do not identify a construction rate. The 100-update straightness statistic changes little under masking. Descriptively, and not prespecified, motion outside S is faster under masking: the masked branch's per-window path length outside S is 1.8, 1.7 and 1.6 times ordinary training's at the three forks (net displacement 1.9, 1.7 and 1.8 times). Appendix B.42 tests how the direction of this motion differs between branches; the next subsection repeats the sweep at depth two.

B.41 The Removal-Time Sweep at Depth Two

The sixteen two-layer runs of Appendix B.33, forked from their frozen sources at 2,000 and from their ordinary branches' saved states at 3,000 and 4,000 (that cohort's ordinary onsets are at a median 6,750, so no 5,000 state was saved), with the same three arms per fork, the same records and the same readings as the one-layer sweep; protocol hashed before any branch ran; no branch censored; the reference arms reproduce the cohort's onsets in all 16 runs. Qualifying runs: 8/8 per task at 2,000, 7 and 8 at 3,000, 5 and 7 at 4,000.

Readings and outcomes. R1, the change in the masked lag between the 2,000 and 4,000 forks: *mixed*, as at one layer. Addition -300, $-1{,}000$, -100, -200, -500 (median -300; the unchanged region counts 3 of 5, met by the all-but-two rule); multiplication -600, $+100$, -900, -400, -800, -500, $+100$ (median -500; unchanged 2 of 7, shrinking 5 of 7, neither reaching its rule); interference 0. R2, masking at 4,000 earlier than ordinary training by more than 200 in all but one qualifying run per task: *met*, 5/5 and 7/7, saving 0.77 and 0.70 of the remaining wait. R3 (no rule): deletion with suppression recovers 0.90 and 0.93 of the masking gain at 2,000, 0.82 and 0.89 at 3,000, and 0.68 and 0.81 at 4,000; the coordinate along d has fallen to 0.80–0.98 of its fork value by 3,000 and 0.72–0.93 by 4,000, earlier than at one layer (0.90–1.17 at 5,000), so deleting what remains along d captures less of the mask's gain late. R4 (no rule): straightness outside S per window 0.991 under the mask and 0.990 in ordinary training (higher under the mask in 11 of 16).

Table 41: **Removal-time sweep at depth two, 16 runs.** Median lag after removal, median wait remaining in ordinary training, and median fraction of the wait saved, by fork and arm (addition / multiplication; full onset).

Fork	Arm	Lag	Wait	Saved
2,000	mask	1,200 / 1,200	4,400 / 4,900	0.76 / 0.77
2,000	delete, suppress	1,750 / 1,500	4,400 / 4,900	0.67 / 0.70
3,000	mask	1,200 / 900	4,200 / 3,900	0.73 / 0.77
3,000	delete, suppress	2,000 / 1,350	4,200 / 3,900	0.62 / 0.63
4,000	mask	1,200 / 800	5,700 / 3,500	0.77 / 0.70
4,000	delete, suppress	2,700 / 1,500	5,700 / 3,500	0.53 / 0.62

Reading. Masking at depth two still saves about three quarters of the remaining wait when it starts at update 4,000. This establishes that the intervention remains effective late in training. The small change in lag after removal does not identify how quickly a circuit was forming before removal. Deletion with suppression captures about 0.9 of the masking gain at 2,000 and less at later forks, when the coordinate along d is smaller; this association does not isolate the coordinate from other changes in the recipient state. Motion outside S is similarly straight with and without the mask.

B.42 Adam's Epsilon and the Direction of the Motion outside S

The same sixteen runs, forked from their source states at 2,000; protocol and readings hashed before any branch ran. Four branches: ordinary training; the mask; and each with AdamW's ϵ set from 10^{-8} to 10^{-12} at the fork, with the moments, learning rate and weight decay unchanged. The motivation is the maintenance measurement: with the early content maintained the training loss is saturated and the raw gradient on the embedding is about 4×10^{-7} in norm, about 3×10^{-9} per coordinate, below ϵ; under the mask the per-coordinate gradient is about 10^{-5}. This motivated the comparison, but the

current gradient alone does not determine the historical second moment in AdamW's denominator. Every branch saved its embedding every 100 updates. The ordinary and masked branches reproduce the sweep's onsets in all 16 runs. For a branch, the *alignment* at an update is the cosine between its 100-update displacement outside S and the direction from that state to the branch's own final embedding outside S; matched windows end at or before the earlier of the two onsets compared.

Readings and outcomes. R1, epsilon: ordinary training with $\epsilon = 10^{-12}$ earlier than with 10^{-8} by more than 200 updates in at least 7 of 8 per task. *Fails*: 5/8 and 6/8, none later by more than 200 updates; paired medians -350 and $-1{,}050$ against ordinary onsets of 8,000 and 9,000. Over matched windows the smaller ϵ moves the embedding outside S faster, by a median factor 1.14 in addition and 1.42 in multiplication. R2 (no rule): under the mask the change is 0 and $+200$ (medians), earlier in none, later by more than 200 in 1 and 4. These are observed changes; the current gradient alone does not predict their size. R3, direction: the median alignment over matched windows higher under the mask than under ordinary training in at least 14 of 16. *Met, 16/16*: under the mask 0.36–0.66 (median 0.53); under ordinary training -0.39 to $+0.49$ (median -0.22), negative in 14 of 16. R4 (no rule): straightness over the first 20 windows, mask against ordinary: 8/8 in addition (0.52 against 0.37), 4/8 in multiplication (0.54 against 0.67). R5 (descriptive): the smaller ϵ does not consistently increase alignment with each branch's own final embedding: 0.19 against 0.31 in addition (higher in 0/8) and 0.18 against 0.22 in multiplication (3/8).

Post-hoc audit of the maintained coordinate. The archived embeddings and per-update records permit a direct check of whether smaller ϵ maintains more early content. We use the same d fixed at update 2,000 in both branches. Coordinates recomputed from every saved embedding agree with the logged $\langle E_t, d\rangle$ within 10^{-8}, and every step satisfies the recorded adaptive-plus-decay accounting to the same tolerance.

Over the fixed suppression window, updates 2,001–4,000, the mean coordinate relative to its fork value is higher with smaller ϵ in only 5/16 runs; the paired median difference is -0.0029. At update 4,000 alone it is higher in 11/16, with paired median difference $+0.0288$. The median ratios of summed adaptive projection to summed decay magnitude are 1.031 and 1.045 for $\epsilon = 10^{-8}$ and 10^{-12}, respectively. These ratios describe the net update balance, not the average coordinate level. Over the secondary, run-specific interval ending at the earlier full onset, the mean coordinate is higher with smaller ϵ in only 1/16 runs (paired median difference -0.0541). Figure 9 shows the complete shared-window trajectories.

Optimizer moments were saved at the fork and at each branch's final state, rather than at every embedding snapshot. At the common fork, the two branches have identical moments. In the raw operand-embedding coordinates, the median within-run value of $\sqrt{\widehat{v}}/\epsilon$ is 0.273 for $\epsilon = 10^{-8}$ and 2,732 for 10^{-12}. This confirms that ϵ contributes substantially to the initial denominator, but does not determine the subsequent trajectory. The per-step records supply the actual update projections without approximating the moments from the current gradient.

Reading. Smaller ϵ advances full onset in 14/16 runs, ties in one and delays one by 100 updates. The original greater-than-200 criterion remains failed. Its effect on the maintained coordinate varies across time and runs, so the data do not support a sustained stronger hold inferred from the equilibrium illustration. Motion outside S also becomes faster, but these measurements do not establish that it offsets a stronger hold or isolate either process's contribution to generalization time. This audit is post hoc and adds no confirmation gate; all per-run contrasts and source hashes are supplied with the artifact.

Acharya & Dhakal (2026) relate AdamW's ϵ to a stability threshold for grokking; our comparison is one setting of ϵ against another at a fixed fork. Masking increases alignment with each branch's own final embedding in all 16 runs. Ordinary alignment is negative in 14/16, so the 16/16 result refers to the paired increase under masking. These are geometric comparisons between trajectories ending at different parameter vectors. They do not identify progress toward a common circuit or show that ordinary training is moving in a functionally wrong direction.

Extension to two layers. The 16 runs of Appendix B.33 (addition seeds 953501–953508, multiplication 953511–953518) are continued from update 2,000 with ordinary training, masking, or ordinary training with $\epsilon = 10^{-12}$. The protocol and readings were fixed before these 48 continuations, reusing

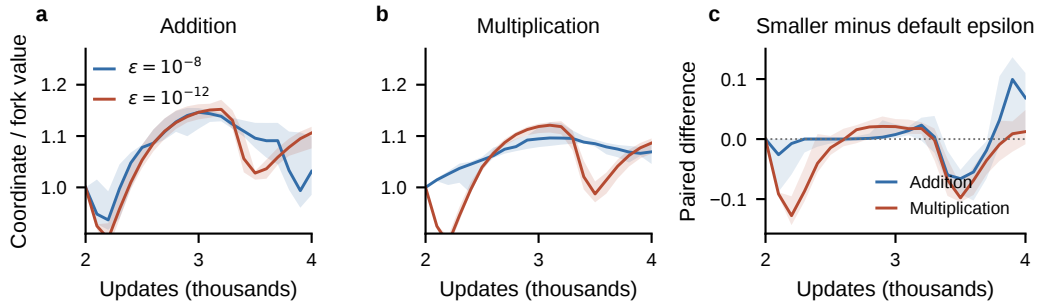

Figure 9: Smaller ϵ does not maintain a uniformly higher early coordinate. Coordinates along the same fork-fixed direction d, relative to the common initial value. Curves and bands show medians and interquartile ranges of the eight runs per task. The right panel plots the within-run coordinate difference, smaller minus default ϵ, over the fixed 2,000-update window; its band is the interquartile range of paired differences. These descriptive trajectories are distinct from a comparison of their values at a single update.

the existing source states. Embeddings are saved every 100 updates; branches stop after both onsets are confirmed and update 5,000 is reached, with a maximum of 40,000. All branches attain, and all normal and mask reference onsets agree with the earlier cohort. Under the same alignment definition, the primary direction criterion (protocol R1, at least 14/16 paired increases) passes in 16/16. Median alignments are 0.472 versus 0.082 in addition and 0.525 versus 0.014 in multiplication (mask versus normal); normal alignment is negative in only 7/16. The descriptive speed reading (protocol R2) gives median normal-to-mask path-length ratios of 0.618 and 0.623, taking each run's median ratio over matched windows and then the median over runs. The epsilon criterion (protocol R3, earlier by more than 200 in at least 7/8 per task) fails at 2/8 and 1/8, with paired median changes of -100 updates in both tasks. The replication supports higher alignment under masking at both depths, while negative ordinary alignment is not consistent across runs at depth two. A cosine to the terminal embedding remains a parameter statistic, not a measurement of which generalizing computation has formed.

B.43 Does Embedding Motion Build a Second Copy of the Fit? A Failed Replication

Measurement-only readings on ordinary-training states (no training). Let K be the top-5 frequencies by power of a run's final ordinary embedding and R the frequencies outside S and outside K; the displacement is the embedding at 5,000 minus at 2,000 on the operand columns, centered. On the clamp cohort's ordinary branches the training cross-entropy of the model with S masked at evaluation falls by a median 2.9 nats from 2,000 to 5,000 while its masked test accuracy stays near chance, and the content built in R is redundant for the full model (removing it at 5,000 costs a median 0.012 nats) and recedes by the end (0.22 of its energy at 2,000). We fixed four readings on that pattern, hashed them, and measured the duration cohort's ordinary branches (Appendix B.37; states at 2,000, 5,000 and the end).

Readings and outcomes (discovery; replication). R1, the masked-evaluation training cross-entropy falls by more than 0.5 nat from 2,000 to 5,000 in at least 7 of 8 per task: 7/8 and 8/8 (medians 3.0, 2.9); *replication fails*, 6/8 and 7/8 (medians 2.0, 1.2). R2, masked-evaluation test accuracy at 5,000 below 0.10 in at least 13 of 16: 14/16 (median 0.045); *fails*, 10/16 (median 0.088). R3, removing the R-part of the displacement at 5,000 raises the training cross-entropy by less than 1 nat in at least 15 of 16: 16/16 (median 0.012, largest 0.44); *fails*, 14/16 (median 0.029, largest 1.33). R4, the R-part's energy at the end below 0.5 of its value at 2,000 in at least 15 of 16: 16/16 (median 0.22); *met*, 15/16 (median 0.18), while the key frequencies' energy grows to 2.6 and 2.3 times. R5 (descriptive): put into the 2,000 model under the mask, the embedding's displacement raises the masked training cross-entropy (medians: the 2,000 embedding 9.73 and 9.31; plus the R-part 10.49 and 9.94; plus the K-part 10.06 and 9.63; plus the whole displacement 10.66 and 10.19), so the fall of that loss in the trajectories is carried by the rest of the network, not by the embedding's motion.

Reading. The account that this displacement builds a redundant second copy of the fit in the other frequencies is not supported: three of the four rules fail on the second cohort, and R5 shows the embedding's own motion does not improve the other half's fit. What stands in both cohorts is that the content built outside S and K between 2,000 and 5,000 is redundant for the full model in 30 of 32 runs and recedes by the end in 31 of 32. The functional role of this displacement remains unidentified.

B.44 Through Which Channel the Early Content Acts: Signal, Response, and a Circuit Built Without It

Signal versus response. With cross-entropy the gradient is $J^\top r$: r the error signal at the output (softmax minus the label, per training pair) and J the network's response, $\partial z/\partial\phi$. On the clamp cohort's saved ordinary states at 2,000, 4,000 and 5,000 we built the four cells $J_A^\top r_B$ with $A, B \in \{\text{kept}, \text{removed}\}$, where *removed* means the early content along d deleted in a diagnostic forward pass and the coordinate along d held fixed in every cell, and turned each into a direction by dividing by the square root of the state's own saved second-moment estimate plus ϵ, without the first moment and without updating either moment; these are preconditioned instantaneous gradients, not AdamW updates, and raw gradients are reported alongside. Swapping the signal alone reproduces the removed model's direction (cosine with $J_{\text{removed}}^\top r_{\text{removed}}$ 0.90, 0.61 and 0.76 at the three states; 0.97, 0.96 and 0.94 for the raw gradients), while swapping the response alone does not (0.04, 0.25, 0.25); the signal swap wins in 15, 14 and 15 of 16 runs. With the content present the signal is almost absent, $|r|$ 8×10^{-9} against 2×10^{-2} without it, and where it exists its cosine with the removed signal is 0.6. Two readings were then fixed and hashed before the duration cohort's states were measured: the signal swap wins in at least 14 of 16 at every state (*met*: 15, 16, 16), and the raw cosine exceeds 0.8 at 2,000 in at least 14 of 16 (*met*: 16). At these states, for these signals and in the gradient's direction, the content's effect on learning runs through the error signal it silences and reorganizes; by this measure the response contributes little to the direction, which does not show that the two Jacobians coincide or that their magnitudes match. The direction our suppression removes and the radial direction $\perp$Grad removes overlap in the embedding ($\cos(d, W_E/\|W_E\|)$ 0.62–0.70 across states and cohorts), so their geometry does not separate the two operators; the content-age result shows that different contents carry different learning costs, and the norm-matched radial control of Appendix B.31 shows that this dependence does not separate them either.

Training past the silencer. If the content delays through the signal it silences, a model that keeps the content in its weights but takes its loss from a forward pass without it should generalize on the masked schedule. On the same 16 runs, from 2,000, the loss was the cross-entropy of a forward pass with the content along d deleted (sharing every parameter), its gradient reached every parameter, and the coordinate along d was then reset to its fork value through 12,000; the actual model and the removed forward were both evaluated every 100 updates. (A first version that backpropagated the removed signal through the actual model, $J_{\text{kept}}^\top r_{\text{removed}}$, diverged in its engineering probes (the actual loss rose to 10^4–10^5 within 300 updates); it was withdrawn before any cohort run and is recorded with the batch.) Readings fixed in advance: R1, the actual model earlier than the clamp of Appendix B.38 by more than 200 in at least 7 of 8 per task; R2, earlier than ordinary training by the same rule. *Both fail, in the opposite direction*: the removed forward generalizes on the masked schedule (median 3,850 and 4,000 against masking's 3,350 and 3,300), but the actual model, whose only difference is the held content, has half of the held-out accuracy at 12,000 (medians 0.53 and 0.54) and generalizes only after the clamp releases and the content decays, at a median 13,000 and 13,050 (later than the clamp by more than 200 in 7 of 8 and 8 of 8; paired medians $+4{,}750$ and $+3{,}750$). This is the protocol's third case: a circuit built in the content's absence works only in its absence. What this supports is narrower than a mechanism: a solution trained with the content out of view does not keep its performance once the content is back in view. Read with Appendix B.38, where the same hold while the loss sees the content barely delays, whether training sees the content appears to set how far the resulting solution tolerates it. One candidate reading, not tested here, is that ordinary training builds a circuit that coexists with the content it keeps rebuilding, and that the slow construction under the content in Appendix B.40 is the cost of that coexistence; we have not identified such a circuit or decomposed the wait into such a cost.

Does a scalar recorded at arrival predict the delay? A post hoc analysis of the shipped restoration records, with the predictors, controls, statistic and thresholds fixed and hashed before any correlation

was computed (the delay-predictor analysis of the artifact). For every restoration branch of the confirmation cohort's 16 runs (30 arms, 480 branches, none censored) the delay is its onset minus the same run's sham; the predictors are the restored increment's energy and the training cross-entropy on arrival with its sign reversed, so that content which leaves more of the training fit intact scores higher. The rule for either predictor: it beats the other within run (Spearman over arms) in at least 12 of 16 runs and its pooled Spearman exceeds 0.5; the same rule then applies unchanged to the weight-decay-0.5 cohort (7 arms, 112 branches). *Both fail.* The fit predictor wins within run in 13 of 16 but pools to 0.25; energy wins in 3 of 16 and pools to -0.03; at weight decay 0.5 the counts are 10 and 6 of 16 with pooled 0.07 and 0.17. Energy fails because every equal-energy control (the scrambled copy, the coherent copy, another run's content, the later content scaled up) carries the full energy and no delay; arrival cross-entropy fails because it mostly measures how much the increment disrupts the recipient's fit, which grows with energy: sham has the lowest (0.33) and no delay, the coherent copy the highest (8.8) and no delay. Retained energy after 1,000 updates does better (pooled 0.59 and 0.52; descriptive) and is broken by the later content (retained 0.91, delay 100). Within matched-energy pairs, however, the ordering is clean: in every pair that restores the run's own early content (full, half dose, donor 2,500 into 2,100, after a moment reset, scaled to the later norm) the native increment fits the recipient's training set better than its scrambled copy in 16 of 16 runs and delays by more than 200 in 15–16 of 16; in every pair that restores content the recipient did not build (later, endpoint, another run's) it fits worse or equally (better in 2, 0 and 6 of 16) and delays by more than 200 in 6, 5 and 1 of 16 (medians 100, 0 and 0). The exception is 2,100 content into the 2,500 recipient: a worse fit in 15 of 16, yet later by 200 in 10 of 16 (median 300). At weight decay 0.5 the full and endpoint pairs behave the same (16 and 16 of 16; 0 and 5 of 16), while its later content fits better in 3 of 16 and is later in 12 of 16 (median 600). Neither of the two arrival scalars tested carries the content's effect; the pairwise pattern is descriptive and was not prespecified.

B.45 Representability under the Haar and Frequency Constraints

Two development probes on first-panel runs 910201 (addition) and 910211 (multiplication), starting from the normal branch's update-50,000 endpoint: the operand embedding is projected into the constraint's complement and training continues under the constraint with every other parameter free, for 100,000 updates, evaluated every 1,000. The Haar probe was first run at 6,000 and 25,000 updates; its trajectory was still rising at both, so the 100,000-update run is the one reported, and the earlier outputs are retained.

Under the basis control's Haar subspace (rank 56), which removes 49% and 55% of the endpoint's operand-embedding energy, test accuracy drops to 0.078 and 0.083, climbs to 0.933 and 0.971 by 25,000 updates and settles at 0.987 and 0.988 by 100,000, never reaching five consecutive observations at or above 99%. Under each run's own character set S (rank 56): for 910201, S holds 78% of the endpoint's energy, removal drops test accuracy to 0.048, stable 99% is reached 1,000 updates later and 1.000 by 8,000; for 910211, S holds 9% and the endpoint's four most damaging frequencies lie outside it, so removal leaves 0.9975 and the character side of that run is uninformative.

These are lower bounds on what the constrained family can reach, not upper bounds: a constrained model at 0.988 exists, so "no generalizing solution exists in the complement" is refuted, but a finite continuation that does not reach 99% cannot show that no constrained solution reaches it, since the projection first drops accuracy to about 0.08 and the continuation is a new constrained optimization problem. Whether the remaining gap is optimization or expressivity is undetermined. The equal-rank character constraint, which removes more energy, imposes no such gap in the one run it damages. Both probes use two runs started from solved models.

B.46 Splitting Each Branch into Fitting and the Delay After It

Post-hoc decomposition of trajectories already recorded for every frequency panel. No branch was retrained, no gate is attached, and the fitting threshold was chosen after all outcomes were known.

Definitions. Each branch is recorded every 100 updates from its fork at update 2,000 to update 50,000, giving 481 points. For a branch, *damage* is the training accuracy lost at the first recorded point, measured under that branch's own operator, so it is how much the intervention immediately reduces the training solution's accuracy. *Fit* is the first recorded update with training accuracy at least

0.99, an absolute update. *Delay* is the onset of stable generalization minus the fit. Both times are exact to the 100-update recording resolution. Branches that never satisfy the onset rule within budget are censored and excluded from delay medians, with the censored count reported.

Recomputation. The onsets are recomputed from the shipped trajectories by the onset rule and equal the recorded ones on all 400 branches; the per-arm fit and delay spreads are in the artifact.

Table 42: **Damage from each intervention, and where the time goes.** Medians over the 16 runs of each arm, not statements about every run. "Ratio" is total updates against that run's normal branch and "earlier" counts runs generalizing before it; "damage", "refit" and "delay" split each branch at the point where training accuracy returns to 0.99. The fork is at update 2,000 and no ratio crosses cohorts. The spreads are in the artifact.

Arm	Ratio	Earlier	Damage	Refit	Delay
normal training	1.000	—	0.00	0	4,650
random half removed, R	0.486	16/16	0.85	150	1,050
matched scaling of S	1.032	3/16	0.00	0	4,800
endpoint's top four, K	1.285	3/16	0.50	100	7,550
24 others, N	0.471	16/16	0.71	100	1,250
union of 28, $K \cup N$	0.544	16/16	0.88	200	1,500
one-shot deletion at the fork	0.737	15/16	0.85	100	2,900
freeze at source values	1.479	0/16	0.00	0	8,400
500-update window, then unmasked	0.481	16/16	0.85	150	1,050
window opened 500 updates later	0.527	16/16	0.81	100	950
Haar subspace for that window	0.932	13/16	0.89	200	4,200
Haar subspace, sustained	—	0/16	0.89	200	none in budget

What the table shows. Table 42 gives the medians. Fit sits at 2,100 or 2,200 in every intervened arm and at 2,000 in the three arms that leave the training solution intact, across damage from 0.03 to 0.89. One or two recorded steps restore a training fit that the operator had cut to as little as 0.11 accuracy. The delay ranges over almost an order of magnitude in the same cells, and two arms never reach the criterion at all.

Damage does not order the delay. Over the fourteen attaining arms the Spearman correlation between median damage and median delay is -0.66, which is an association rather than an ordering. Two cells break it in opposite directions. K removes four frequencies, reduces training accuracy by half and has the longest attained delay of any removal arm at 7,550, longer than every arm except the freeze control at 8,400, while $K \cup N$ removes those same four inside a set of 28, reduces training accuracy by 0.88 and delays 1,500. The Haar subspace and the 22.5° rotation cause the most damage and never reach the 99% criterion within budget; the rotation does reach stable 90% in 16/16 and 95% in 10/16, so this is non-attainment of the criterion, not an absence of learning. Within a single arm the correlation across its 16 runs is -0.14, so the relation is not carried inside an arm either.

Scope. This is an accounting of where the updates go, and it is weaker than a causal localization. What it supports is that the arrival-time differences are not composed of time spent regaining the training fit, since that is uniformly fast, and that the size of the disruption does not order them. It does not show that what happens during the refit is unimportant: a change made in one or two recorded steps can surface much later, which is what the window arms of Appendix B.17 demonstrate over a longer interval. *Fit* is a single recorded point at 0.99 training accuracy, not a claim that the branch has settled. The fitting threshold, the panels and the arms were fixed before this analysis, but the analysis itself was specified after their outcomes were known.

B.47 Holding the Criterion, Not Only Reaching It

Reaching a criterion and holding it are separate outcomes, and a transfer that accelerates acquisition can relapse badly under continued optimization, as can training from scratch (Jia, 2026). Every arm in this paper is scored on arrival time, so this appendix asks the other question of the same branches. It is

a post-hoc re-analysis of trajectories already shipped with the artifact: no retraining, no checkpoints, no new cohort, and no gate.

Measure. For each branch that attains stable generalization, the *drawdown* is the largest fall in held-out accuracy below that branch's own running maximum, from the onset evaluation to the end of the 50,000-update budget. Each arm is paired against the normal branch of the *same* training run, so no comparison crosses a cohort: the panels sharing the discovery cohort's sources use its normal branches, while the prespecified main confirmation and the window confirmation use their own.

Outcome. Table 43 gives all 18 arms. Normal training itself relapses, by a median 0.019 on the discovery cohort and 0.041 on the main confirmation, so this is a relative statement throughout. No arm is worse than its own normal branch at a two-sided sign-test p below 0.21, and the paired medians span -0.003 to $+0.043$. The single largest paired difference belongs to matched scaling on the main confirmation, which is the control that does not bring generalization forward; among the arms that do, the largest is $+0.012$. Earlier generalization in these panels is therefore not bought with stability that we can detect.

Scope. Drawdown is bounded by the fixed budget, so a relapse beginning after update 50,000 is outside the measurement; Jia (2026) follows relapse for 1,500 steps after a validation trigger in their prospective cohorts, in mini-batch training, a far shorter horizon than this window. Onset requires five consecutive evaluations at 0.99, so every window starts from a high plateau. Sixteen runs per arm bound the power of a paired sign test: this reports counts and a range, and an increase in drawdown smaller than the spread here would not be detected. No equivalence claim is made, and no threshold was prespecified for this analysis. The recomputation script, its inputs and its outputs are in the artifact.

Table 43: **Post-onset drawdown, every arm against its own normal branch.** "Drawdown" is the largest fall in held-out accuracy below a branch's running maximum after onset. "Normal" is the median drawdown of the matched normal branches of the same runs, and "paired" the median of the per-run differences. "Worse" counts runs in which the arm's drawdown exceeds its own normal branch's; p is an exact two-sided sign test. Post-hoc, no prespecified threshold.

Arm	n	Drawdown	Normal	Paired	Worse	p
low-energy half	16	0.014	0.019	-0.003	6/16	0.61
random half	16	0.031	0.019	+0.007	11/16	0.21
low-energy, norm preserved	16	0.024	0.019	+0.006	11/16	0.21
matched scaling	16	0.029	0.019	+0.009	11/16	0.21
magnitude pruning	16	0.035	0.019	+0.021	10/16	0.45
random half (main confirmation)	16	0.037	0.041	+0.009	10/16	0.45
norm preserved (main confirmation)	16	0.039	0.041	+0.012	9/16	0.80
matched scaling (main confirmation)	16	0.064	0.041	+0.043	10/16	0.45
endpoint top four, K	16	0.027	0.019	+0.007	10/16	0.45
24 others, N	16	0.036	0.019	+0.008	10/16	0.45
union of 28, $K \cup N$	16	0.025	0.019	+0.002	8/16	1.00
one-shot deletion	16	0.031	0.019	+0.011	10/16	0.45
freeze at source values	16	0.022	0.019	-0.001	7/16	0.80
500-update window	16	0.030	0.019	+0.011	10/16	0.45
window opened 500 later	16	0.027	0.019	+0.008	9/16	0.61
Haar subspace, same window	16	0.014	0.019	-0.001	8/16	1.00
sustained (window confirmation)	16	0.021	0.022	+0.001	8/16	1.00
window (window confirmation)	16	0.024	0.022	+0.003	8/16	1.00

B.48 Intervals for the Comparisons Prespecified as Counts

Several conditions in this paper were prespecified as one-sided counts against a margin and fell short (Appendix A). This appendix gives interval estimates for two of them: the window confirmation's within-margin condition, met in 13 of 16, and the nested design's condition for a measurable delay, met in 10 of 16. A count that misses its bar establishes neither a difference nor a closeness, so the intervals below describe both comparisons.

Measure. For each comparison, the per-run paired difference in onset updates is summarized by its Hodges–Lehmann estimate and a distribution-free 95% interval from the Walsh averages,

indexed by the exact Wilcoxon signed-rank null computed by dynamic programming, together with an exact two-sided sign test. The verdict is three-way against the prespecified margin of 200 updates: *difference* if the interval excludes zero, *within margin* if it lies entirely inside $(-200, +200)$, and *not resolved* otherwise. This is a post-hoc re-analysis of shipped output. It sets no new gate and changes no prespecified outcome.

Outcome.

Comparison	Median	HL	95% interval	Sign p	Verdict
Window − sustained, window confirmation	$+50$	$+50$	$[0, +250]$	0.039	not resolved
$K \cup N - N$, nested design	$+200$	$+300$	$[+50, +500]$	0.180	difference
Window − sustained, discovery cohort	0	0	$[-50, +50]$	0.727	within margin

Three readings follow.

On the first panel, where the window arm was developed, the window and the sustained constraint are equivalent within the prespecified margin: the interval $[-50, +50]$ is a quarter of it. That is a positive result the count rule never stated.

On the window confirmation's fresh runs the same comparison is **not resolved**. The interval reaches $+250$, past the margin, so equivalence is not established there; it also has zero at its boundary and the sign test reaches $p = 0.039$, so the window is more likely slightly later than sustained removal than not. The size of that lateness is between nothing and about 250 updates against onsets near 3,200. Equivalence held where the arm was developed and did not replicate at that strength on seeds that had no part in designing it.

For the nested design the interval says more than the count. The prespecified rule asked for $K \cup N$ to be later than N in at least 12 of 16 to record a measurable delay and observed 10, which on its own would read as the relied-upon four causing no delay inside a larger beneficial set. The signed-rank interval is $[+50, +500]$ and excludes zero, which is evidence that adding those four to an already beneficial removal delays generalization. Two cautions bound that reading. The exact sign test does not reach significance at $p = 0.180$; and the signed-rank interval assumes a symmetric distribution of paired differences, which we do not test, where the sign test assumes nothing of the kind. Taken together these *support a positive delay without establishing one*, and they do not license the stronger claim that the prespecified count rule simply lacked power. What they do remove is the reading that the delay is zero, which the count outcome alone was taken to support. Generalization is still earlier with the addition, at 0.544 against normal training, which is the claim the manuscript makes.

Scope. Sixteen runs per comparison; the intervals are wide and two of the three verdicts turn on a boundary. The signed-rank interval requires symmetry of the paired differences, which is not tested here and is the reason both tests are reported. Onsets are recorded at a 100-update resolution, so an interval endpoint at 0 or $+50$ is one grid step. The recomputation script and its output are in the artifact.

B.49 Memorization and Circuit Formation at Each Arm's Own Arrival

Circuit efficiency (Varma et al., 2023) is a candidate account this paper does not identify: memorization is broadband in the character basis and the generalizing solution is not, so a half-basis constraint would tax the memorizing route more and the earlier arms would arrive with memorization further suppressed. Nanda et al. (2023)'s restricted and excluded losses read that state, and the saved checkpoints support embedding-space analogues of both without retraining. The analogy is not an equivalence: their measures keep or remove frequencies in the *logit* Fourier decomposition, whereas the operator here projects the operand embedding, and the two are not equivalent through an attention block and a nonlinear MLP. What follows is a dependence diagnostic in the basis this paper intervenes in, read with those measures' logic; its numbers are not comparable to theirs.

Measures. For an arm with masked set W, at a given checkpoint, rank the frequencies it may still use by their individual held-out cross-entropy damage and take its own top four, T. Then

- **excluded**: the rise in *training* cross-entropy when T is ablated on top of W. A large rise means the training fit rests on T, so little memorization is left to carry it.
- **restricted**: the *held-out* cross-entropy keeping only T and the mean. A low value means the generalizing computation is already concentrated there.

Each arm is read at its *own* first stable-generalization checkpoint, the state it is in at arrival, and each is paired against the normal branch of the same run read at *its* own arrival. This is the convention the onset audits use. It is not a comparison at a common absolute time, and the checkpoint spacing gives no trajectory between arrival and 50,000 updates.

Outcome. Medians over 16 runs at each arm's own arrival, with paired differences against that run's normal branch as Hodges–Lehmann estimates and distribution-free 95% intervals.

		excluded train rise		restricted test CE	
Arm	Ratio	Median	Paired vs normal	Median	Paired vs normal
normal	1.000	9.43	—	0.142	—
K	1.285	7.76	-2.72 $[-4.04, -0.42]$	0.144	-0.019 $[-0.153, +0.055]$
N	0.471	10.17	$+0.83$ $[-1.16, +2.86]$	0.064	-0.061 $[-0.113, -0.006]$
$K \cup N$	0.544	8.74	$+1.55$ $[-3.96, +7.27]$	0.111	$+0.006$ $[-0.122, +0.153]$

No clear signal appears in the predicted direction, and three things keep that from being a refutation. First, the fastest arm's excluded rise is *indistinguishable* from normal training's, $+0.83$ $[-1.16, +2.86]$: the interval contains zero and also contains the effect the account would predict, so this is a failure to detect and not evidence of absence. Second, comparing arms at their own arrivals cannot test whether cleanup happened earlier — a mechanism that moves the arrival time need not change the state at arrival, and Varma et al. (2023) concerns relative learning speed and circuit efficiency, the parameter norm a circuit needs for a given logit, rather than the residue at a fixed accuracy. Third, the one interval here that excludes zero belongs to the *latest* arm and runs the other way, which separates delay rather than earlier generalization. What the earliest arm does show is a lower restricted loss, its generalizing computation already more concentrated in its own leading frequencies when it arrives.

Scope. Six paired comparisons, three arms by two measures, with no correction applied; both intervals that exclude zero have exact two-sided sign tests at $p = 0.077$, and the signed-rank intervals assume a symmetry of paired differences that is not tested. Within-arm spread is large relative to the differences: the excluded rise ranges 3.9–15.0 across normal branches and 6.8–14.8 across N branches. Two checkpoints, no trajectory, one cohort, and the top four are re-measured at each checkpoint rather than fixed, so an arm whose leading frequencies move is scored on its current ones. These results *bound* the circuit-efficiency account in the direction its own prediction names; they do not establish an alternative, and no gate was attached. The recomputation script runs over checkpoints already saved; it and its outputs are in the artifact.

B.50 Identity at Matched Cardinality: Five Halves on One Cohort

Every 28-frequency arm reported before this one brings generalization forward by about the same amount, which leaves open whether identity matters at half the basis or only dose does. This experiment fixes the cardinality at 28 of 56 on the discovery cohort's 16 sources and varies only which 28 are removed, using two complementary splits. It also tests, at matched cardinality, whether removing modes the solution relies on differs from removing modes it does not, a comparison motivated by Kam et al. (2026), who show that, at fixed weight decay, a large active band of signal-free modes collapses held-out accuracy, and note that their mode count acts only through the teacher-weighted spectrum. Their analysis uses a fixed Fourier feature map with a linear read-out and peak held-out accuracy; here the features are learned and the outcome is time to a criterion.

Design. Protocol fixed before any cell and shipped in the artifact, with the classification below fixed in advance. Three new arms, 48 branches, on the discovery cohort's sources at update 2,000 with AdamW state retained and its unchanged 50,000-update budget; the normal branch, the low

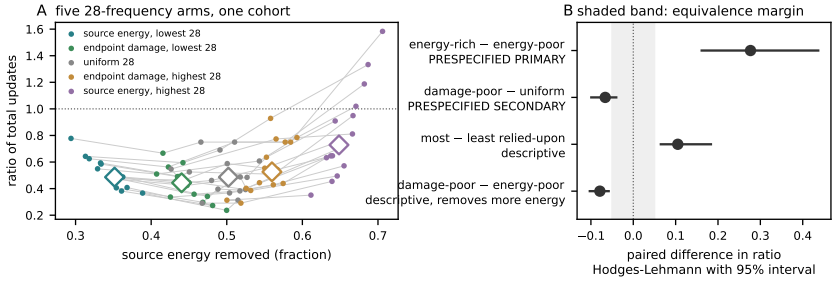

Figure 10: Frequency identity changes generalization time at matched cardinality. Five 28-frequency arms on the first panel's 16 runs, with real rank matched to within one direction. (A) Removed source energy versus total-update ratio; lines connect arms within a run, diamonds mark arm medians, dotted line marks normal training. The green arm removes more energy than the teal arm yet generalizes earlier. (B) Paired Hodges–Lehmann estimates and 95% intervals; the band is the preset ± 0.05 margin. Only the primary contrast establishes a difference beyond that margin. Lower two rows are descriptive; two arms use endpoint information (Appendix B.50).

half and the random half are the discovery cohort's recorded branches and are not retrained. The *high half* is the exact complement of the prespecified low-energy set and uses source information only. The *top-damage half* is the 28 largest single-frequency held-out cross-entropy damages at the normal 50,000-update endpoint, read verbatim from the frozen endpoint damage audit, and the *bottom-damage half* its complement; both are diagnostic and endpoint-informed. All 48 branches completed, none failed.

Cardinality is matched exactly. Real rank is matched exactly on addition and differs by one direction on multiplication, where the Nyquist frequency carries a single real direction rather than a sine/cosine pair; which half is short is recorded per run and is not systematic.

The five arms.

Arm	Selection	Removed source energy	Median ratio	Earlier
bottom-damage half	endpoint damage, lowest	0.440	0.444	16/16
random half	uniform	0.502	0.486	16/16
low half	source energy, lowest	0.352	0.487	16/16
top-damage half	endpoint damage, highest	0.560	0.526	16/16
high half	source energy, highest	0.648	0.729	12/16

Outcome, against the prespecified classification. Figure 10 shows every run. Paired per run on total-update ratio, as Hodges–Lehmann estimates with distribution-free 95% intervals and exact two-sided sign tests, against the prespecified practical-equivalence margin $\delta = 0.05$.

Contrast	Arms	HL	95% interval	Sign p	Verdict
Primary, source energy	high half − low half	$+0.277$	$[+0.159, +0.439]$	0.001	identity matters
Secondary, endpoint damage	bottom-damage − random half	-0.066	$[-0.102, -0.037]$	0.001	identity matters
Descriptive	top-damage − random half	$+0.046$	$[-0.004, +0.113]$	0.388	not resolved

Both prespecified verdicts read "identity matters"; the secondary's interval lies below zero, which here means the least relied-upon half is the earlier one.

Both prespecified contrasts exclude zero; only the primary establishes a difference larger than the ± 0.05 practical-equivalence margin. At 28 of 56, which 28 are removed changes the time to generalization by far more than the equivalence margin: selecting the energy-rich half instead of the energy-poor one delays generalization by a median $+0.226$ in ratio, more than half the size of the gain from masking itself. Per task the primary holds in 8/8 addition and 7/8 multiplication runs.

Is this dose in disguise? Removed source energy orders the arms well — the within-run Spearman correlation between removed energy and ratio over the five arms has median $+0.505$ with midranks and is positive in 15 of 16 — so the question is whether anything is left once dose is granted. One contrast answers it. The bottom-damage half removes a median $+0.105$ *more* source energy than the low half and is nevertheless earlier, by a median -0.083 in ratio, Hodges–Lehmann -0.079 with interval $[-0.105, -0.055]$, earlier in **16 of 16 runs**. Under a monotone energy-dose account the ordering would be the other way, as it is for every other pair. Energy is a strong predictor and not a sufficient one.

Where the earlier dissociation sits. The two arms whose endpoint damage differs by 84 accuracy points, the low half and the random half, differ in their effect on generalization time by a Hodges–Lehmann -0.019 with interval $[-0.053, +0.025]$ that spans zero; the sample median is -0.029 and its bootstrap interval $[-0.053, +0.015]$, and the two are different estimators. That comparison is unchanged and remains the striking case; what this experiment adds is that it does not generalize. Selecting *by* endpoint damage, at the same cardinality, moves the effect on generalization time by $+0.105$ $[+0.062, +0.186]$ in 15 of 16. Endpoint damage carries real information about the effect on generalization time; two particular selection rules happened to sit close together on it.

Scope. Sixteen runs, one cohort, one cardinality, one architecture and trigger. The top- and bottom-damage halves are read from the normal endpoint, so neither is a rule anyone could apply without first solving the task; the high half uses only source information, but the cohort's earlier outcomes were known, so the experiment is not blind to them. Damage is concentrated on about five of 56 frequencies, so the damage-ranked halves are those few plus a remainder ordered by a numerical noise floor, and the top-damage half is close in construction to the recorded $K \cup N$ arm. Removed energy and identity are not orthogonal across these five arms and no arm was energy-matched by design; the bottom-damage-against-low-half contrast is the one place the two come apart, and it is a single pair. No mechanism is identified. The recomputation script and its outputs are in the artifact.

B.51 Where the Fork Sits in the Phase Structure

The operator is applied before behavioral generalization, but whether circuit formation had begun by then is not measured in the main experiments. It can be measured on checkpoints already saved, and this appendix does so. The answer depends on the width of the mask used to read the circuit, so the mask is stated with every number.

Cohort. A dense-checkpoint cohort holds 251 checkpoints per run at every 200 epochs on a configuration *identical* to the discovery cohort's: $p = 113$, one layer, no LayerNorm, $d_{\text{model}} = 128$, $d_{\text{mlp}} = 512$, four heads, 30% training fraction, full-batch AdamW at 10^{-3}, weight decay 1, betas $(0.9, 0.98)$, 50,000 epochs. Sixteen runs are used, eight ReLU addition and eight ReLU multiplication. **The seed family differs** — these are seeds 900001–900008, the discovery cohort's are 910201–910218 — so what follows is the phase structure of this configuration, not of those sixteen runs.

Measures, in the form they are defined. The other progress-measure audit (Appendix B.49) uses embedding-space analogues, and says so. Here the measures are Nanda et al. (2023)'s own: the logits over all operand pairs form a $p \times p \times p$ tensor, and the two-dimensional real Fourier transform is taken over the operand pair. We take as key frequencies the five carrying the most logit power at the final checkpoint (their main model has five), held fixed along the trajectory; they carry a median 99% of the non-constant logit power, range 97.5–99.8%, reproducing the sparse-circuit finding on this cohort.

The logits are centered along the output-class axis before the transform, which removes components that do not affect the softmax. *Restricted* keeps the constant term and, per key frequency, the projection onto $\cos w(a+b)$ and $\sin w(a+b)$, the operator of the released analysis of Nanda et al. (2023); *excluded* subtracts the same projections and scores training loss. Two wider axis-aligned masks are reported beside it. For multiplication, operands are reindexed by discrete logarithm to the primitive root 3 and the pairs with a zero operand dropped.

The state at the fork. At update 2,000, the discovery cohort's trigger, medians over the 16 runs: training accuracy 1.000, held-out accuracy 0.102, and restricted held-out accuracy 0.277 against

1.000 at generalization. The fork sits at a median 0.256 of the way to stable generalization. The key-frequency circuit is therefore **already under way and nowhere near sufficient**: it is well above chance in every run and generalizes on its own in none.

Ordering, with symmetric markers. Marker choice decides the ordering, so both measures are read the same way: the first epoch at which each reaches a fraction q of its own total change between memorization and generalization. Medians over the 16 runs, as a fraction of the epoch of stable generalization:

q	Formation onset	Cleanup onset	Formation first
0.10	0.069	0.640	16/16
0.25	0.136	0.740	16/16
0.50	0.274	0.829	16/16

Formation precedes cleanup at every threshold and in every run, the ordering Nanda et al. (2023) report from the shape of their curves rather than at thresholds. It is the one conclusion here that does not depend on which restricted mask is used.

The fork is inside formation, not before it. Formation onset at $q = 0.10$ precedes the fork at 2,000 in **16 of 16** runs, and by the fork a median 0.464 of the total restricted-loss decline is complete. The operator is applied to a model that has memorized its training set *and* has a partly built generalizing circuit.

How much the mask choice matters. Medians over the same 16 runs:

Restricted operator	Components	Held-out acc. at fork	Decline done at fork	Fork first
Sum-angle projection (published)	2 per key	0.277	0.464	0/16
Per-frequency 2×2 block	21	0.238	0.419	0/16
Every pair of kept indices	121	0.082	0.004	12/16

The published projection and the 2×2 block agree. Only the widest mask puts the fork before formation, and it is the one that admits cross-frequency products, which are large while the model is memorizing and decay only late — so it tracks memorization as much as the circuit.

Scope. A different seed family from the panels the claims rest on; sixteen runs, one architecture, two operations. The markers are thresholds on smooth curves, and nothing here measures what the constrained arms do to this timeline. The scripts, which recompute every number above under all three masks from the shipped trajectories, are in the artifact.

B.52 Where the Constrained Circuit Goes

A post-hoc reading of the solution each 28-frequency arm of Appendix B.50 reaches, at its own first stable generalization and with its mask active, using the logit-Fourier measures of Nanda et al. (2023) built as in Appendix B.51. Deleting an embedding direction need not silence that frequency in the logits, since the MLP is nonlinear, so the share of each arm's own logit power on the normal endpoint's five key frequencies is a measurement. Medians over the 16 runs:

Arm	Ratio	n_{99}	Power on normal's keys	Above 0.05	Earlier than normal
bottom-damage half	0.4437	9.0	0.978	16/16	16/16
random half	0.4863	10.5	0.724	16/16	16/16
low half	0.4865	13.0	0.940	16/16	16/16
top-damage half	0.5263	15.0	**0.002**	**1/16**	16/16
high half	0.7288	23.0	**0.004**	**2/16**	12/16
normal	1.0000	16.5	0.964	16/16	—

The half the endpoint relies on most ends with a median 0.002 of its logit power on the normal solution's key frequencies, above 0.05 in one run of sixteen, against 0.964 for normal itself, and is still earlier than normal in 16/16. Appendix B.15 shows from the other side that a branch that generalizes earlier can also keep those frequencies: the gain does not turn on whether the normal solution's circuit is kept. Within a run, how sparse a solution an arm reaches orders its effect on generalization time (median Spearman $+0.667$ $[+0.410, +0.734]$ between the number of frequencies carrying 99% of logit power and the ratio, positive in 15/16), and a participation ratio read at the masked fork orders the five chosen arms at $+0.900$; on sampled masks and held-out runs that fork-time reading falls to $+0.604$ (Appendix B.53). Normal training needs a median 16.5 frequencies and is later than every masked arm at the median, so sparsity orders the constrained arms among themselves without accounting for the gap to normal training. All quantities are post-hoc, read at each arm's own arrival, and establish no direction of causation; the per-frequency power vectors and the full correlation table are in the artifact.

B.53 Can a Rule Read Before Training Say How a Set Will Change Generalization Time?

Everything up to here compares sets that were chosen: two selection rules in Appendix B.9, five arms in Appendix B.50, all of them picked to sit at the ends of some axis. Chosen sets can show that identity matters. They cannot say whether the effect of a set on generalization time is *predictable*, because the arms and the predictors move together by construction and because every number is read on the same runs that suggested it.

This experiment samples sets instead of choosing them, and scores a rule on runs it was not fitted on.

Design, fixed before any branch ran. Twenty masks per run on the same 16 sources, 320 branches. Each mask removes 28 of the 56 eligible frequencies, and on multiplication the Nyquist frequency is excluded from the pool so that every mask has real rank exactly 56, within and across runs. The sample is a designed grid crossed with random fill, because twenty uniform draws would all land near half the source energy and leave the ends of the space empty; the grid is described as a grid and not as a random sample. Every draw is a deterministic function of the run identifier, with no search, rejection or redraw.

The 16 runs were split by seed into a fitting half and a held-out half before any branch ran: addition 910201–910204 and multiplication 910211–910214 fit, the other eight are held out.

Three predictors, all readable at the fork: x_1, the fraction of centered source operand-embedding energy the mask removes; x_2 and x_3, how many of the source's leading eight and leading four frequencies it removes. A fourth, x_4, the share of the normal endpoint's single-frequency damage the mask takes out, is *not* available at the fork and is recorded only for the comparison below.

The prespecified quantity is the within-run Spearman correlation between an ordinary least-squares score in x_1, x_2, x_3, fitted on the fitting half alone, and the observed total-update ratio in each held-out run. The implemented fit first centres the target within each fitting run:

$$y_{hm} = r_{hm} - \bar{r}_h, \qquad \bar{r}_h = \frac{1}{20}\sum_{m=1}^{20} r_{hm}.$$

It regresses y_{hm} on an intercept and the uncentred source predictors. The resulting score therefore predicts the *relative* effect on generalization time within a run, not an absolute total-update ratio, and can be negative; substituting typical predictor values into the coefficients of Appendix B.54 gives a value near zero for exactly this reason. Neither the fitted coefficients nor the frozen predictions are affected. The new-run test uses only the score's ordering and never estimates $\bar{r}_h$ from held-out outcomes. The prespecified threshold was a held-out median correlation of 0.5.

Outcome. All 320 branches completed, none censored, none failed. The rule reaches a held-out median Spearman of $+\mathbf{0.732}$, distribution-free 95% interval $[+0.644, +0.808]$, positive in 8 of 8 held-out runs and above $+0.57$ in every one. That is the prespecified *predictive* classification: a rule using only quantities available at the fork forecasts the ordering of effects on generalization time for masks it has not seen, on runs it was not fitted on.

No held-out gain from endpoint damage in this fit. Refitting the same rule with x_4 included moves the held-out median from $+0.732$ to $+0.679$, a change of -0.053 in the direction of slightly worse. This is not because endpoint damage is uninformative. Taken on its own it orders the sampled masks within every run, at a median $+0.588$ in 16/16. Including it does not improve this fitted rule on this held-out half; that does not show it has no incremental value under other models or sampling schemes. The new-run confirmation in Appendix B.54 tests the locked source-only rule against leading-eight count, not this endpoint-augmented comparison.

What predicts, and what does not. The raw within-run correlations over the 20 sampled masks, all 16 runs, say which part of the rule is doing the work:

Read at the fork, over sampled masks	Median ρ	Positive
Leading eight frequencies removed, x_2	$+0.637$	16/16
Leading four frequencies removed, x_3	$+0.609$	16/16
Removed source energy, x_1	$+0.267$	**9/16**
Endpoint damage removed, x_4, not available at the fork	$+0.588$	16/16

Removed energy does not survive sampling. On the five chosen arms of Appendix B.50 it had a within-run median of $+0.505$ in 15 of 16. Over sampled masks it is $+0.267$ and positive in only 9 of 16. What orders the effect on generalization time is how many of the source's *leading* frequencies a mask removes, not how much energy it removes in total. The five arms could not separate those two, because they were selected along the energy axis. The negative coefficient on x_1 in the fitted rule is a partial effect of the same kind: with x_2 and x_3 absorbing the leading-frequency count, the residual association with total energy runs the other way. Read on its own, x_1 is weakly positive.

Scope. The same 16 sources throughout, so this tests prediction across masks and across runs within one cohort, not across cohorts. One architecture, two operations, one trigger, one cardinality. This experiment alone does not establish transfer across cohorts: the held-out half shares its cohort with the fitting half. A subsequent test on new runs and uniform masks finds weaker but positive ranking transfer (Appendix B.54). The protocol, the seeding rule, the split and the thresholds are in the artifact, frozen before any initialization; an independent script that imports none of the runner's analysis rebuilds every mask, rank, onset, fit and held-out correlation.

A prespecified follow-on. A fork-time participation ratio that ordered five chosen arms at $+0.900$ post hoc (Appendix B.52) was tested on these sampled masks and this held-out half with its direction fixed in advance: held-out median $+0.604$ $[+0.096, +0.759]$, positive in 7 of 8, adding $+0.029$ to the prespecified rule. The post-hoc figure was inflated by the choice of arms, as was the $+0.505$ for removed energy.

B.54 New-Run Confirmation of Ranking and Conditional Effect

This experiment tests two predictions developed on earlier runs: whether a locked source-only rule ranks the effect of new masks on generalization time, and whether the same set's effect on generalization time can reverse sign when another set is masked. The protocol fixed the runs, selection rules and analyses before new training. The two tests share runs; they are independent of development, not of each other.

Cohorts and stopping. The configuration is unchanged from the discovery cohort: one-layer ReLU Transformers on arithmetic modulo 113, with its data split, initialization and AdamW settings. There are 80 new runs: addition seeds 912001–912064 and multiplication seeds 912101–912116. All 64 addition runs enter the background test. The first 16 addition runs and all 16 multiplication runs enter the ranking test. No run is replaced or excluded.

The primary fork is update 2,000, with a prespecified secondary fork at 1,000 for ranking. Branches retain the source's model, optimizer moments, counters and random states, are evaluated every 100 updates, and stop at the fifth observation at test accuracy at least 0.99, within 50,000 updates; the 64 addition normals continue to 50,000 to supply the endpoint used to select K. All 1,552 unique branches attain, with no censoring, failures or exclusions.

Locked ranking rule. At each source, 20 distinct masks are sampled uniformly without replacement within each mask, rejecting duplicate masks. Each contains 28 eligible frequencies. Multiplication's rank-one Nyquist frequency is excluded, so every mask has real rank 56. All masks, source features and predictions are saved before treated outcomes. With the definitions of x_1, x_2, x_3 in Appendix B.53, the previously fitted coefficients are held fixed:

$$\widehat{r} = 0.06005216629972136 - 0.3490545345107044\, x_1 + 0.022294630893378742\, x_2 + 0.01150474096189158\, x_3.$$

There is no refitting, sign selection or outcome-based mask selection. Within each run, midrank Spearman correlations compare $\widehat{r}$ and the leading-eight count x_2 with T. Ranking T is equivalent to ranking $T/T(\varnothing)$ within a run. The paired increment is $\Delta\rho = \rho(\widehat{r}, T) - \rho(x_2, T)$.

The primary summaries are the means of the per-run ρ and $\Delta\rho$. Their two-sided percentile 95% intervals use 20,000 bootstrap draws, resampling 16 runs with replacement independently within each task, while preserving the pairing between each run's two correlations; the bootstrap seed is 20260920. The criterion requires all 32 runs to be scorable and both intervals' lower bounds to exceed zero at update 2,000. Scorability requires all 20 onsets and nonconstant ranks. This fixed bootstrap choice does not guarantee nominal coverage with 16 runs per stratum.

Table 44: **Locked-rule ranking on 32 new runs.** Intervals are for means, not medians. The same runs contribute at both forks.

Fork	Mean ρ [95% CI]	Mean $\Delta\rho$ [95% CI]	Median ρ
2,000, primary	0.351 [0.267, 0.436]	0.063 [0.027, 0.100]	0.357
1,000, secondary	0.290 [0.216, 0.363]	0.070 [0.048, 0.093]	0.324

All 32 runs are scorable at each fork (Table 44). At update 2,000, correlation is positive in 30/32 and the rule improves on leading-eight count in 24/32; the baseline's mean correlation is 0.289. Addition and multiplication have mean correlations 0.329 and 0.374, with mean paired gains 0.067 and 0.058. At update 1,000, correlation and paired gain are each positive in 28/32; the baseline's mean is 0.220. Task-specific mean correlations are 0.353 and 0.227, and mean gains 0.075 and 0.065. The primary criterion is met; the secondary fork cannot serve as an independent replication or rescue a failed primary test. The new median correlation, 0.357 at update 2,000, is lower than the earlier held-out 0.732. Both the run cohort and the mask distribution changed. These data establish neither absolute timing calibration nor whether endpoint damage adds predictive information beyond the locked source-only rule.

Every sampled half is earlier. Descriptively, and not part of the prespecified ranking test, every one of the 640 masks at update 2,000 generalizes before its run's normal branch, at a median ratio of total updates of 0.43 (0.42 in addition and 0.44 in multiplication), and 634 of them below 0.75; at update 1,000 all 640 do, at a median 0.30. On the discovery cohort's designed grid of Appendix B.53, which deliberately includes masks removing most of the leading frequencies, 305 of 320 do.

Selection and event for the conditional effect. For each addition run, K contains the four frequencies with the largest single-frequency test-CE damage at its normal 50,000-update endpoint, breaking ties by frequency index. It is selected separately for each run, not copied from a pilot case. The background rule draws 6,000 deterministic 24-frequency candidates disjoint from K and groups them by removed source energy fraction in bins of width 0.005. Among bins containing at least three distinct leading-eight counts, it selects the widest count span, then the bin nearest the expected uniform-draw energy, then the lowest bin index. It selects the low, middle and high counts in that bin, with lexicographic tie-breaking among masks. All three are saved, but only the middle candidate is tested. This is not a choice of the best of three treated outcomes.

The four branches are normal, K, B and $K \cup B$, all from the same 2,000-update source. The primary event fixed in the protocol requires $c(K) < -200$ and $c(K \mid B) > 200$, with strict inequalities and all four onsets observed. Five of 64 runs satisfy it (Table 45), giving 7.8% with a two-sided Clopper–Pearson 95% interval $[2.6, 17.3]\%$. All 64 designs are feasible and fully observed, so the possible-witness bounds coincide with the observed fraction. There is no pass-count gate: the

estimand is the observed-witness yield of this fixed selection procedure, not the prevalence of possible reversals under all backgrounds. The opposite direction occurs in three runs and remains secondary.

Descriptively, over all 64 runs, $c(K)$ has median $+1{,}500$ updates (interquartile range $[+300, +3{,}050]$), exceeds $+200$ in 48 and falls below -200 in 11; $c(K \mid B)$ has median $+300$ ($[+100, +900]$), exceeds $+200$ in 34 and falls below -200 in 6. The interaction $c(K \mid B) - c(K)$ has median $-1{,}400$ and is negative in 48 of 64, and $|c(K \mid B)| < |c(K)|$ in 51. These summaries were not the prespecified estimand; they reproduce, on new runs and with a different background rule, the direction of the discovery cohort's post-hoc interaction (Appendix B.15), at about a fifth of the effect of masking the same set alone where the discovery cohort gave about a tenth. An onset-noise baseline for these reversals is in Appendix B.55.

Table 45: **Every observed sign reversal beyond the 200-update margins.** All entries are absolute onset updates or their signed differences. The five primary cases and three opposite-direction cases are kept separate.

Addition seed	$T(\varnothing)$	$T(K)$	$T(B)$	$T(K \cup B)$	$c(K)$	$c(K \mid B)$
912023	12,200	11,700	3,500	4,400	-500	$+900$
912044	8,700	7,700	3,300	4,300	$-1,000$	$+1,000$
912055	6,600	5,800	3,000	4,100	-800	$+1,100$
912056	11,000	10,700	3,500	4,100	-300	$+600$
912063	5,100	4,800	2,900	3,600	-300	$+700$
912050, opposite	9,500	10,600	5,100	3,500	$+1,100$	$-1,600$
912057, opposite	8,200	9,000	4,000	3,300	$+800$	-700
912058, opposite	10,500	12,200	4,300	3,300	$+1,700$	$-1,000$

Verification and scope. A replay reproduced the four onsets of one earlier case before new training, and an independent scorer reconstructed all 1,552 onsets, designs and correlations without the runner's helpers. Ranking is tested at one cardinality in two operations; the background test covers addition alone, one fork and an endpoint-informed K. Neither result identifies a mechanism or a generally beneficial removal rule.

B.55 An Onset-Noise Baseline for the Sign Reversals

On the 64 addition runs of the background test and from the same update-2,000 sources, four arms per run were added, with the branch mechanics of that test: R, four frequencies drawn by a seeded rule from the 28 outside $K \cup B$, masked alone; $R \cup B$; normal training after a single Gaussian perturbation of the operand embedding of relative Frobenius size 10^{-4} at the fork, with nothing masked; and B masked after the same perturbation. The existing $T(\varnothing)$, $T(K)$, $T(B)$ and $T(K \cup B)$ are reused, and one $T(B)$ replayed bit for bit. Readings fixed in advance: K's reversals are *indistinguishable from a random four* if R shows at least 3 of 64 in the primary direction and *exceed it, descriptively*, if at most 2; onset jitter *exceeds the 200-update margin* if either perturbed arm moves by more than 200 in at least 7 of 64 runs.

Outcome. All 256 branches attain. R never reverses, 0 of 64 in either direction at margins of 200, 500 and 1,000 updates, against K's 5 and 3; the paired exact McNemar test cannot go below 0.0625 at these counts. R alone is earlier in every run (median $c(R) = -1{,}000$ updates, against $+1{,}500$ for K), and $c(R \mid B)$ has median -200. The perturbation moves the normal branch's onset by a median 150 updates, by more than 200 in 18 of 64 and by 500 at the 90th percentile; with B masked it moves it by more than 200 in 1 of 64, exactly zero in 60. No perturbed pair reproduces the primary reversal pattern (0 of 64).

Reading. Both prespecified readings apply: K's reversals exceed a random four, which behaves as removal of unused content does, and the onset of normal training is sensitive enough that a single 10^{-4} perturbation exceeds the 200-update margin in about a quarter of runs. Quantities referenced to the normal branch, $c(K)$ and $c(R)$, carry that noise; quantities between two constrained branches, such as $c(\cdot \mid B)$, barely do. The reversals are specific to K in this sense, and they remain few and near that noise.

B.56 A Cohort That Failed Its Validity Gate

On the earlier panels, a prespecified "delay law" predicted that the delay from removing the endpoint's top k frequencies ($k = 1, 2, 4, 8$) would track the fraction of source energy they carry. On the development seed (900006, 180 branches) it failed on both criteria: the Spearman correlation with that fraction was 0.41 against a required 0.5, and the gap was larger at the prespecified source than at the quarter source in 8/12 cells against a required 10/12. The delay from removing relied-upon frequencies rose from $k = 1$ to $k = 4$ and then saturated, while removing random frequencies brought generalization further forward the more were removed (median ratios 0.94, 0.90, 0.76, 0.54). The two-seed confirmation (900003 and 900004) failed its validity gate: the unmodified branch reached stable generalization in 19/21 cells at the quarter source against 20 required, because two subtraction cells need roughly 34,000–37,600 updates from there against a uniform 25,000-update budget fixed without checking the earlier sources. No budget was extended after the outcome. Its other three gates passed, and at the prespecified source the same pattern appears: the excess over random removal is $+900$, $+1{,}700$, $+3{,}400$ and $+2{,}350$ updates at $k = 1, 2, 4, 8$, and random ratios are 0.95, 0.88, 0.71 and 0.54. The frozen licensing declares the cohort invalid; nothing in the main text rests on it, and its tables and figures are in the artifact.

Table 46: **Cohort key.** Extensions reuse runs only where stated. The discovery and confirmation cohorts comprise 336 distinct fresh runs: sixteen cohorts of 16, plus 80 distinct runs in the ranking and background confirmations, which share 16 addition runs. Branches and checkpoints are not independent runs.

Cohort	Size	Role
Historical frequency removals	48 settings / 4 seed panels	Four core versus four non-core removals; each seed spans 12 task–activation cells
Discovery cohort	16 runs / 96 branches	Eight fresh seeds per task, fixed 2,000-update trigger and retained AdamW
Main confirmation (prespecified)	16 runs / 64 branches	A second, disjoint set of fresh seeds; seeds and removal sets fixed before initialization; scores damage and generalization time on one set
Main confirmation at modulus 97	16 fresh runs / 64 branches	The same design and rule at $p = 97$, 24 of 48 frequencies masked (Appendix B.2)
Main confirmation, two layers	16 fresh runs / 64 branches	The same design and rule with two Transformer layers, no LayerNorm, budget 40,000 (Appendix B.3)
Main confirmation, two layers with LayerNorm	16 fresh runs / 64 branches	The same design and rule with pre-LayerNorm, budget 30,000; two runs unfinished (Appendix B.5)
Main confirmation, LayerNorm, fork at 1,000	16 fresh runs / 48 branches	The same model with the prefix stopped at 1,000, three branches, budget 15,000 (Appendix B.5)
Fork sweep	16 runs / 48 branches	The restoration confirmation's runs; sustained masking from updates 1,000, 2,000 and 3,000 (Appendix B.7)
Locked-rule ranking confirmation	32 runs / 1,280 treated branches	16 new runs per task; 20 uniform masks at each of two forks, with no refit. Shares 16 addition runs and their normal branches with the next row
Background confirmation	64 runs / 256 branches	New addition runs; normal, K, B, $K \cup B$ at update 2,000. Endpoint-informed K and one fixed background rule; 1,552 unique branches across both tests including 16 multiplication normals
Same-set endpoint audit	48 saved endpoints / 32 set–delay pairs	Post-hoc normal, low-energy and random endpoints; no retraining
Subspace controls	16 runs / 64 branches	Equal-rank Haar subspace and three geodesic rotations of the removed subspace, on the first panel's sources
Nested removal	16 runs / 48 branches	The endpoint's top four by damage, 24 others drawn once, and their union, on the first panel's sources
Window confirmation	16 fresh runs / 48 branches	Seeds never trained before; normal, sustained and a 500-update window from one source each, with a prespecified three-part event
Restoration confirmation	16 fresh runs / 160 branches	Seeds never trained before; one normal continuation and sham, native and phase arms at three cuts from one source each; prespecified early-minus-late timing interaction, both tasks required
Weight-decay 0.5 replication	16 fresh runs / 64 branches	Seeds never trained before; the restoration recipe at weight decay 0.5, one cut, predictions fixed before launch (Appendix B.23)
Content age	16 runs / 64 branches	The restoration confirmation's runs; the normal arm's content at confirmed generalization or at 50,000 restored into the unmasked 2,100 state, with scrambled controls (Appendix B.24)
Replenishment tests	4 pilot and 12 extension runs / 40 and 72 branches	The restoration confirmation's runs; suppression of the update along the restored direction, with orthogonal and scaling controls (Appendix B.31)
Replenishment in ordinary training, modulus 97	16 fresh runs / 96 branches	Seeds never trained before; the six-branch design of the one-layer test at $p = 97$, random half 24 of 48 (Appendix B.35)
Replenishment in ordinary training, weight decay 0.5	16 fresh runs / 96 branches	Seeds never trained before; the six-branch design of the one-layer test at AdamW weight decay 0.5 (Appendix B.34)
Replenishment in ordinary training, two layers	16 fresh runs / 96 branches	Seeds never trained before; the six-branch design of the one-layer test at depth two (Appendix B.33)

Table 46 continued.

Cohort	Size	Role
Deletion controls at depth two	16 reused runs / 32 branches	Known-cohort extension of the two-layer maintenance test, deletion alone and deletion with scaling (Appendix B.33)
Deletion controls at depth two, fresh seeds	16 fresh runs / 64 branches	Seeds never trained before; ordinary training, deletion alone, deletion with scaling and deletion with suppression from one source each, prespecified rule (Appendix B.33)
Deletion controls at weight decay 0.5 and modulus 97	32 reused runs / 64 branches	Known-cohort extensions of the weight-decay-0.5 and modulus-97 maintenance tests, deletion alone and deletion with scaling (Appendix B.33)
Radial control	12 reused runs / 24 branches	Known-cohort extension of the restoration-cohort suppression test: norm-matched radial update control, prespecified reading failed (Appendix B.31)
Suppression duration, release and clamp	16 fresh runs / 144 branches	Seeds never trained before; suppression to 3,000, 4,000, 6,000 or throughout, deletion with suppression, and three clamps to 5,000; 64 release diagnostics (Appendix B.37)
Holding the content in place	16 fresh runs / 96 branches	Seeds never trained before; a one-sided floor and two-sided clamps along d to 12,000 or 8,000, a clamp along o, and suppression throughout (Appendix B.38); its frozen states also serve the maintenance measurements (Appendix B.39)
Removal-time sweep	16 runs / 144 branches	The clamp cohort's runs; masking, deletion with suppression, and ordinary training from the saved states at 2,000, 4,000 and 5,000, with gradient and motion records (Appendix B.40)
Epsilon and direction	16 runs / 64 branches	The clamp cohort's runs; ordinary training and the mask from 2,000, each at AdamW ϵ 10^{-8} and 10^{-12}, embeddings saved every 100 updates (Appendix B.42)
Training past the silencer	16 runs / 16 branches	The clamp cohort's runs; loss from a forward pass with the early content deleted, every parameter updated, the coordinate along d clamped at its fork value to 12,000; the actual model and the removed forward both evaluated (Appendix B.44)
Delay predictors at arrival	592 branches / no new runs	Post hoc on the restoration cohorts' shipped records (30 arms at weight decay 1, 7 at 0.5); predictors, controls, statistic and thresholds hashed before computing (Appendix B.44)
Holding the content in place, two layers	16 runs / 32 branches	The two-layer replenishment cohort's runs; ordinary training and the one-sided floor along d to 12,000 (Appendix B.38)
Removal-time sweep, two layers	16 runs / 144 branches	The two-layer replenishment cohort's runs; masking, deletion with suppression, and ordinary training from the saved states at 2,000, 3,000 and 4,000, with gradient and motion records (Appendix B.41)
Direction and epsilon, two layers	16 runs / 48 branches	The two-layer replenishment cohort's runs; ordinary training, masking and ordinary training with $\epsilon = 10^{-12}$; no new source runs (Appendix B.42)
Replenishment in ordinary training	16 fresh runs / 192 branches	Seeds never trained before; twelve arms from each normal state at 2,000: suppression, scaling and orthogonal controls, each also after a one-shot deletion, deletion alone, two masks and the complementary half (Appendix B.32)
Two-layer restoration	16 runs / 48 branches	The two-layer confirmation's runs; sham, native and phase at cut 2,100, budget 40,000 (Appendix B.4)
Nested subsets and random 24	16 runs / 80 branches	The restoration confirmation's runs; the 2,100 content restored into nested random 8, 12, 16 and 20 frequencies of S, or into S without the random four (Appendices B.29 and B.30)
Restoration into part of S	16 runs / 48 branches	The restoration confirmation's runs; the 2,100 content restored into K_S, a random four of S, or the other 24 (Appendix B.28)
Content age at weight decay 0.5	16 runs / 64 branches	The same comparison on the weight-decay 0.5 replication's runs (Appendix B.25)
Haar window	16 runs / 16 branches	The same equal-rank Haar subspace as the basis control, masked for 500 updates and then unmasked
Brief window	16 runs / 32 branches	A 500-update mask, from the fork or 500 updates later, then unmasked with the effective embedding written back
One-shot and freeze arms	16 runs / 32 branches	Separates deleting the content at the fork from preventing its return, on the first panel's sources

C Experimental Configurations and Scope

Every cohort follows the architecture and optimizer of Nanda et al. (2023): a decoder-only Transformer with d_{model}=128, four heads, d_{mlp}=512 and no layer norm, ReLU activations in every fresh-seed cohort (the historical panels also use GeLU and SiLU and the four arithmetic operations), modulus 113 (97 in Appendix B.2), one layer (two in Appendix B.3, two with pre-LayerNorm in Appendix B.5), training fraction 0.3, full-batch AdamW (Loshchilov & Hutter, 2019) at learning rate 10^{-3} with weight decay 1.0 (0.5 in the replication of Appendix B.23) and betas $(0.9, 0.98)$, with no learning-rate schedule; weight decay takes PyTorch's form, multiplying the weights by $1 - \mathrm{lr}\,\mathrm{wd}$ at each update. The betas match their released code; the paper does not state them. Their code also adds a 10-step linear learning-rate warmup, which we omit. The training lengths are ours: 50,000 updates for every one-layer fresh-seed cohort (the ordinary-training replenishment test stops a branch

once both endpoints are confirmed, after update 5,000), 40,000 for the two-layer and 30,000 for the two-layer LayerNorm confirmation, and, for the historical panels, 25,000 updates for each branch from an accuracy-defined fork, where Nanda et al. (2023) train for 40,000 epochs. Section 2 and Appendix B give the rest.