WEIGHT-SCALE GROWTH TRACKS TRAINING EFFORT, NOT LEARNING QUALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Does weight growth mean learning? Previous work showed that data predictability forecasts growth in the Weibull weight-scale parameter λ during transformer training. We test what this growth records, why it can diverge from generalization, and how it accumulates. In a controlled factorial grid crossing token shuffling with fixed-budget repetition, fully shuffled, highly repeated data produces the largest $\Delta\lambda^2$ but the worst held-out loss. Under fixed AdamW, $\Delta\lambda^2$ closely tracks accumulated radial-alignment work. A component- and time-resolved follow-up further shows that data conditions change both when this effort accumulates and whether it is routed toward Transmission or query/key Selection weights. Within the main grid, pairing $\Delta\lambda^2$ with the train–held generalization gap separates learning, memorization, and nothing-to-learn without seed-level label flips. The same frozen thresholds preserve this separation under matched corner interventions on C4 and Python code. These experiments trace weight-scale growth from controlled data construction through optimizer work and component-resolved weight evolution to behavior. Weight-scale growth therefore describes training activity, but alone does not determine whether the fitted structure generalizes.

1 INTRODUCTION

Transformer weights grow systematically during training (Merrill et al., 2021). Within a two-parameter Weibull description of element-wise weight magnitudes $|W|$, recent work separated the fitted magnitude scale λ from the central-body shape k, and linked the growth of λ to pre-training data predictability (Ding, 2026a;c). What that growth records — learning or merely optimization effort — remains untested. The AdamW update makes the ambiguity explicit: it gives a signed leading-order budget for $\Delta\|W\|^2$ (Ding, 2026b), decomposing scale motion into radial alignment, injection, and decay, and $\Delta\lambda^2$ provides a matched fitted-scale readout of this motion when the central-body shape is stable. Both learning structure and repeatedly fitting the same tokens can increase this quantity. We ask what the growth records and how data conditions change its accumulation.

We test this distinction with a 4×4 factorial grid over WikiText-103 that varies the token-shuffling fraction f and the fixed-budget tiling factor ρ, while holding the architecture and optimizer fixed. Three measured coordinates summarize the resulting streams: local predictability (bigram conditional entropy D), retained structure (S), and redundancy (R). The decisive cell — fully shuffled, highly repeated data — produces the largest $\Delta\lambda^2$ in the grid while attaining the worst held-out loss. Pairing the weight-side signal with the train–held generalization gap separates learning, memorization, and nothing-to-learn without seed-level label flips.

Across the fixed-AdamW grid, injection co-scales with alignment and decay varies much less than the observed scale response. In this setting, $\Delta\lambda^2$ therefore closely follows accumulated radial-alignment work. A component- and time-resolved follow-up shows how data shapes that work: the repetition-dose peak moves with evaluation time in both architectures, and scale activity is routed differently between FFN/attention-output Transmission weights and query/key Selection weights. Under the same corner interventions, the paired diagnosis is retained on C4 and Python code, and these experiments locate a boundary of the earlier data-to-scale relation.

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.

We make three contributions. **First**, a factorial intervention separates weight-scale growth from generalization quality. **Second**, an AdamW budget analysis and time-resolved follow-up identify the dominant source, timing, and component destination of the scale response. **Third**, paired weight and behavioral measurements distinguish three outcomes and are tested across corpora, learning rates, architectures, and a preliminary model-size sequence. The interventions are f and ρ; D, S, and R describe the resulting streams.

2 Related Work

Weight-scale and weight-space diagnostics. Parameter norm grows systematically during pretraining (Merrill et al., 2021); margin-based theory combines norm terms with normalized margins rather than treating raw norm as a standalone quality measure (Bartlett et al., 2017; Neyshabur et al., 2018); Heavy-Tailed Self-Regularization reads spectral *shape* as quality-related (Martin & Mahoney, 2021a;b; Loftus, 2026); and training-time signals can invert (Power et al., 2022; Liu et al., 2023; Kumar et al., 2024; Nakkiran et al., 2020; Andriushchenko et al., 2023). Recent weight-space methods recover coarse training-data mixtures from released model weights (Huang et al., 2026). We study what a simple magnitude-scale coordinate records during controlled pretraining and how optimizer dynamics generate it.

Optimizer-induced scale dynamics. Under scale-invariant parameterizations, weight norm sets the effective learning rate and approaches an equilibrium governed by η and weight decay (van Laarhoven, 2017; Kosson et al., 2024; Wan et al., 2021); the AdamW three-force budget of Ding (2026b) decomposes squared-norm change into alignment, injection and decay. We use this signed budget to identify the accumulated radial contribution that the measured scale change records.

Memorization, repetition, and behavioral diagnostics. Networks fit random labels readily (Zhang et al., 2017; Arpit et al., 2017), memorization is accompanied by weight displacement (Nagarajan & Kolter, 2019), and random-label training leaves structured traces in the weights (Maennel et al., 2020). Exposure count is a strong predictor of extractable memorization: memorization grows with duplication count (Carlini et al., 2023), and regeneration can increase superlinearly with repeated occurrence (Kandpal et al., 2022); repetition also carries quantified training costs (Muennighoff et al., 2023; Hernandez et al., 2022), and memorization dynamics scale with model size (Tirumala et al., 2022). Most directly, Chudnovsky et al. (2026) find that evaluation damage from duplicated data can peak at an intermediate repeat count. Memorization outcomes are quantified by extraction- and behavior-based measures (Carlini et al., 2021; Morris et al., 2025), and are not uniformly harmful: long-tail examples may require memorization for near-optimal generalization (Feldman, 2020; Feldman & Zhang, 2020). We examine a related weight-side response under controlled repetition and local corruption, and compare it with the generalization gap. This connects repetition studies to a directly measured training-time weight signal without treating that signal as a standalone measure of learning quality.

3 Methods

Method overview. Our analysis follows a data–optimization–weights–behavior chain. We vary token shuffling f and repetition ρ, and describe the resulting streams by predictability D, retained structure S, and redundancy R. During AdamW training (Loshchilov & Hutter, 2019), we track weight scale λ, central-body shape k, row-scale heterogeneity H^W, and the signed norm budget. Training and held-out behavior are then compared with these weight trajectories. The main grid measures the effort–quality relation; the component-resolved follow-up measures its timing and routing. Table 1 summarizes the two protocols, with implementation details in Appendix A.1.

Interventions. The main grid independently varies shuffle fraction f and tiling factor ρ under a fixed token budget. D and S describe the resulting loss of local structure, while R measures repeated exposure. Approximately, $R_\rho \simeq 1 - (1 - R_1)/\rho$, with measured values $0.017/0.752/0.938/0.984$ for $\rho = 1/4/16/64$. The follow-up records separate q/k/v/o trajectories, enabling analyses of timing and component routing.

Table 1: Two complementary experimental protocols and their division of labor.

Item	Main $f\times\rho$ grid	Component-resolved follow-up
Role	controlled $f\times\rho$ response surface, described by (D, S, R); effort–quality separation and regime classification	when scale growth accumulates; which component receives it
Model & init	Pythia-70M (Biderman et al., 2023), fused QKV; HF default init ($\lambda_0 = 0.01776$, $k_0 \approx 1.20$); 3 independent seeds	Llama-style 70M (Touvron et al., 2023), separately stored q/k/v/o; 3 paired-initialization seeds
Data & design	WikiText-103 (C4, Python code cross-corpus cells); seeded partial shuffle f + tiling ρ; 4×4 grid: 48 primary + 24 cross-corpus + 4 $\eta/2$ runs; structure-matched held-out	WikiText-103 full stream, same construction; 2×2 corners + dose line (f=0, $\rho \in \{1, 4, 16, 64\}$): 18 runs; structure-matched held-out, scheduled evaluations
Budget & optimizer	8,000 steps ($T = 98.3$M); AdamW $\beta = (0.9, 0.999)$, $\eta_{\text{peak}} = 3\times10^{-4}$, $\lambda_{\text{wd}} = 0.01$	5,000 steps (fully shuffled ρ=1, 8,000-step sensitivity extension); AdamW $\eta_{\text{peak}} = 10^{-3}$, $\lambda_{\text{wd}} = 0.1$
Readouts	terminal $\Delta\lambda^2$ (pooled middle-80% Weibull fit); accumulated I_{align}; gap; behavioral probes	$I_{\text{align}}(t)$, $\Delta\text{RMS}^2(t)$, (λ, k, H^W) per component, q/k separate; gap trajectories
Inference & checks	within-grid effects; 3-seed statistics; logging and stability checks in Appendix A.1	within-follow-up mechanism bridge; paired contrasts; budget-closure, paired-init and leakage checks in Appendix A.1
Cross-expt.	within-protocol comparisons; cross-protocol agreement is assessed by direction and ordering	
Size screen	fixed-budget Pythia 70M/160M/410M screen on four corner cells; one seed per cell (Appendix B.7)	

Behavioral readouts. We use gap $= \ell_{\text{held}} - \ell_{\text{train}}$, with a structure-matched held-out stream. BLiMP and 64-step verbatim extraction provide complementary behavioral evidence (Appendix B.2.4).

Weight readouts. Weight fits pool the measured Transmission subset shared by both loggers: FFN and attention-output matrices. λ is a matched-condition magnitude scale; per-matrix fits reproduce the pooled ordering (Spearman $\rho = 0.97$, median over matrices, seed 0). The fitted shape k remains within $\pm0.7\%$ of its initialization value $k_0 \approx 1.20$ (measured 1.206, Table 3) across the main grid. In the follow-up, H^W is the standard deviation of per-row log-RMS, with q and k projections analyzed separately. Low values indicate similar row scales; larger values indicate greater row-scale dispersion. The direct second moment ΔRMS^2 of the same matrices confirms that the main ordering is not created by Weibull fitting (Appendix B.2.2); the value of λ is that it separates central-body scale from the accompanying shape coordinate k and can be measured from stored checkpoints without optimizer logs.

Effort and inference. For component c, accumulated radial-alignment effort is $I_{\text{align},c}(T) = \sum_{t<T} -2\eta_t\langle W_{c,t}, u_{c,t}\rangle$, with $u = \hat{m}/(\sqrt{\hat{v}} + \epsilon)$ the AdamW step direction. Together with injection and decay it closes the signed squared-norm budget $\Delta\|W_c\|^2 = I_{\text{align},c} + I_{\text{decay},c} + I_{\text{inj},c} + \varepsilon_c$ per run and channel (Appendix A.3). "Effort" throughout denotes this accumulated signed radial contribution to squared weight norm, not total update energy; cross-component comparisons use per-parameter $i_{\text{align}} = I_{\text{align}}/N_c$. Grid statistics use 3-seed means, Jonckheere–Terpstra permutation tests (Jonckheere, 1954) for monotone trends, bootstrap CIs for partial correlations, and classification thresholds fitted within the grid.

4 Results

We first relate controlled data construction to weight-scale growth, then test whether that growth reflects training effort or learning quality. We next resolve where and when the effort accumulates, before testing these findings across corpora, learning-rate and protocol changes, and model size.

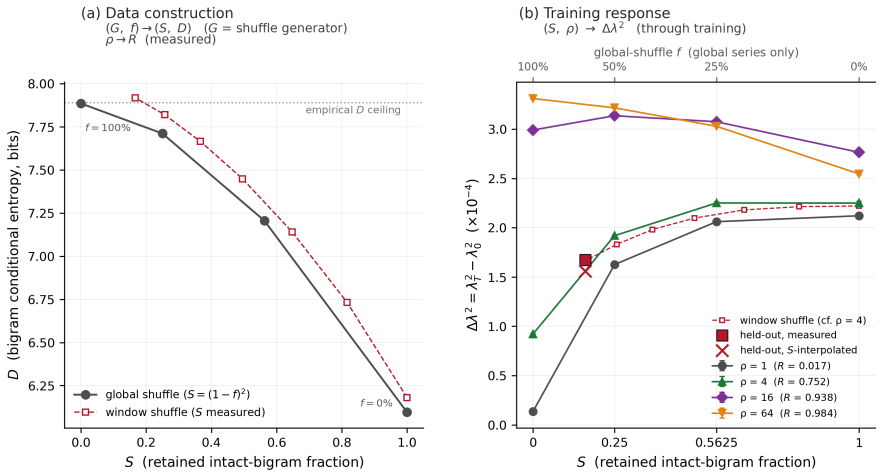


Figure 1: **Data coordinates and terminal scale response. (a)** Retained structure S versus bigram entropy D for global and window shuffling; both approach the empirical D ceiling as $S \to 0$. **(b)** Terminal $\Delta\lambda^2$ versus S at four repetition levels; labels report ρ and measured R. Repetition reverses the structure–scale slope. The window-shuffle point is compared with the approximately matched $\rho = 4$ row (interpolation rule in Appendix A.2). Curves show means $\pm$ s.d. over three seeds; the window series is single-seed.

4.1 From data construction to training response

Figure 1 relates the measured data coordinates D, S, and R to terminal $\Delta\lambda^2$; full grid values are in Table 7 (Appendix B.1).

D approaches its ceiling as retained structure vanishes and no longer resolves the response: near $D = 7.9$ bits, repetition alone produces a $24\times$ range in $\Delta\lambda^2$. Repetition also reverses the sign of the structure–scale relation. Under approximate repetition matching, the window-shuffle comparison is better described by S than by D, so the earlier D relation depends on both generator and repetition. The two coordinates differ by construction: D compresses the stream into one-step token uncertainty, whereas S records how much local order the construction retains. Global and window shuffling can therefore approach a similar D ceiling while preserving different local structure, and once D saturates, S and R still distinguish streams that D maps to nearly the same value.

4.2 Weight growth separates from learning quality

Does the largest weight growth mark the best learning? No. The fully shuffled, $64\times$-repeated cell has both the largest $\Delta\lambda^2$ in the grid ($3.31 \pm 0.01 \times 10^{-4}$) and the worst held-out loss (Figure 2a). The same condition is also largest under the direct pooled $\Delta\mathrm{RMS}^2$ readout in all three seeds (Appendix B.2.2). Pairing $\Delta\lambda^2$ with the matched gap separates LEARNED, MEMORIZED, and NOTHING-TO-LEARN conditions. The thresholds formalize the separation observed in the main grid: their labels are stable under leave-one-seed checks, and all 28 external runs retain their labels when the wikitext thresholds are refitted without each seed. The low-growth boundary is supported by only two design cells. Threshold construction and sensitivity analyses are reported in Appendices A.4 and B.2.1; BLiMP and extraction provide complementary behavioral evidence in Appendix B.2.4.

Across the 48 main-grid runs, $\Delta\lambda^2$ closely tracks I_{align} (Figure 2b). The seed-0 budget shows that injection remains a small, proportional fraction of alignment and that decay varies much less across cells than alignment (Appendix B.2.3), so under the fixed AdamW protocol, differences in weight-scale growth are mainly associated with differences in accumulated radial alignment. At $\rho = 16$, the fully shuffled stream accumulates 985 a.u. of I_{align} per nat of training-loss reduction, against 607 for the unshuffled stream, a $1.6\times$ difference (Table 10); and after linearly controlling for D and R, greater growth is associated with worse transfer ($\mathrm{pcorr}(\Delta\lambda^2, \mathrm{gap} \mid D, R) = +0.491$).

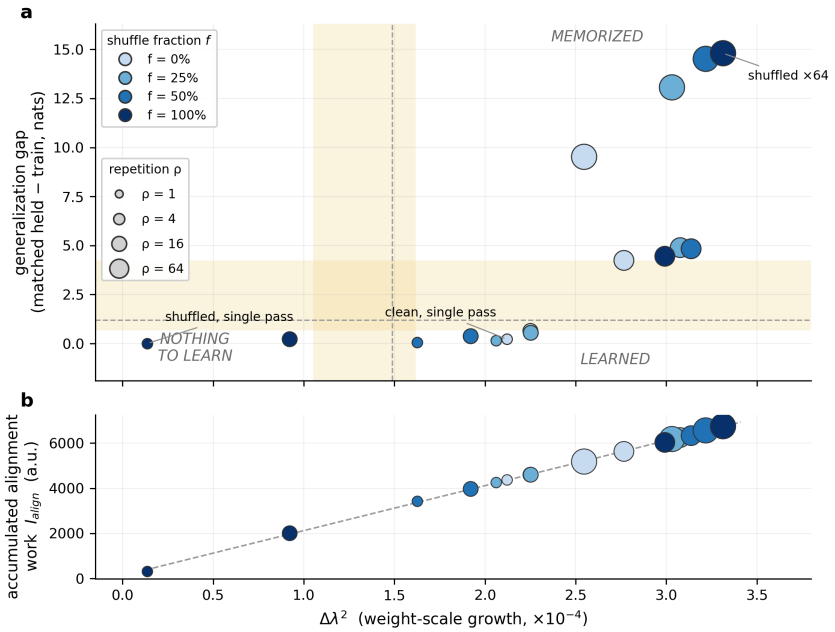


Figure 2: **Weight-scale growth separates from generalization quality and tracks accumulated radial-alignment work in the main grid.** Both panels plot the same 16 grid-cell means (3 seeds each) and share the $\Delta\lambda^2$ axis, so each cell aligns vertically across panels. **(a)** The $(\Delta\lambda^2, \text{gap})$ plane; marker color = shuffle fraction f, size = repetition ρ; error bars are mostly smaller than the markers. Gold bands mark the empty inter-cluster intervals; dashed lines are the data-driven threshold conventions. **(b)** Accumulated alignment work I_{align} against the same axis; the dashed line is a linear fit to the 16 cell means. Per-seed $\Delta\lambda^2$ values are tabulated in Table 9 (Appendix B.2.2).

This effect is not caused by taking more optimization steps, because the training budget is fixed. Repetition instead changes how long the per-step radial contributions remain positive rather than cancelling: they continue to accumulate at $\rho = 16$ but weaken after the $\rho = 64$ arm saturates (§4.4). In the fully shuffled stream, the falling training loss shows that the repeated sample is being fitted, while the poor held-out loss shows that this fit does not generalize.

4.3 Component routing links effort to training progress

The follow-up arms are D1 $= (f0, \rho1)$, D2 $= (f100, \rho1)$, D3 $= (f100, \rho64)$, D4 $= (f0, \rho64)$, and D5/D6 $= (f0, \rho4/\rho16)$ on the structured dose line.

Adding structure at low repetition increases effort in both measured subsets. At high repetition, it increases Transmission effort but reduces q/k Selection effort. Arm D3 has the largest Selection effort but only modest train-loss reduction, whereas D4 nearly drives train loss to zero without maximizing either component budget (Figure 3a; Table 2).

D1 and D4 also occupy similar low-H^W, moderate-Selection-effort states while differing by about 12 nats in gap (Figure 3b). D3 instead has the largest q/k row-scale heterogeneity (median $H^W_{q/k}$ 0.29 against 0.17–0.18 for D1 and D4), showing that its Selection row scales are more uneven; this state coordinate is distinct from the fitted-scale calibration of the measured Transmission subset in Appendix B.3. The contrast indicates that data structure changes where scale-related optimization is deposited: structured streams place more of the measured radial contribution in the FFN and attention-output paths, whereas repeated shuffled streams concentrate the increase in q/k Selection. This is consistent with the architectural roles of the components: query and key projections determine which token relations are selected, whereas FFN and attention-output weights transform and transmit the selected representations, so repeated shuffled streams may reinforce sample-specific selection constraints while structured streams place more of the radial contribution in reusable feature

transformation. This interpretation fits the observed routing pattern, but the experiment does not isolate its activation-level mechanism. Component routing and H^W show where training changed the weights, but do not determine generalization quality.

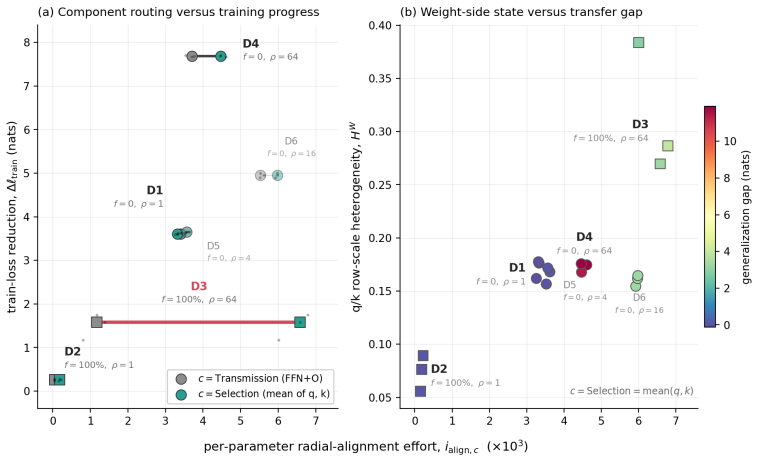


Figure 3: **Component routing links radial-alignment effort to training progress, but not to generalization quality. (a)** Train-loss reduction (steps 0–5,000) versus per-parameter effort in the measured Transmission subset and q/k Selection; connected large markers are paired-seed medians, small points seeds (circles = structured, squares = shuffled; D5/D6 faint). **(b)** Selection effort versus q/k row-scale heterogeneity H^W (mean of the q and k projections), colored by gap; higher H^W means greater disparity among q/k row scales. D1 and D4 occupy the same low-H^W region but differ by $\sim$12 nats. Full trajectories in Figure 9.

Table 2: Component-resolved paired contrasts: median paired-seed differences in per-parameter effort, $\Delta i_{\text{align}} \times 10^3$, at step 5,000 (positive = first arm larger; all signs 3/3 seed-consistent; Transmission = measured subset, FFN+O). Arm definitions are given in §4.3.

Paired contrast	meas. Trans.	q	k	Reading
D1−D2: add structure at low ρ	+3.36	+3.09	+3.13	higher in all measured subsets
D4−D3: add structure at high ρ	+2.60	−2.05	−1.89	Transmission higher; q/k lower
D4−D1: add repetition, structured	+0.26	+0.95	+1.35	similar effort; gap differs $\sim$12 nats
D3−D2: add repetition, shuffled	+1.12	+6.29	+6.56	increase concentrated in q/k

The fitted-scale readout is less accurate when row-scale mixtures change. In the structured arms the Weibull second-moment estimate tracks squared-norm change within a stable factor of 1.11–1.15; in D3's measured Transmission subset it captures about 20–60% of the norm growth. In D3 the scale change is therefore not close to a uniform rescaling of the measured matrices; the shortfall is consistent with part of the change being redistributed across rows or outside the middle-80% fitting window, so that the fitted central-body scale and the whole-distribution second moment no longer move proportionally. Marked changes in shape or row-scale heterogeneity therefore require checking k, fit quality, and H^W alongside λ (Appendix B.3; full per-arm trajectories in Appendix B.4).

4.4 Scale accumulation is scheduled, not static

The endpoint dose response is a slice of a schedule. In the structured row, the scale response peaks at $\rho = 64$ early in training but at $\rho = 16$ from step $\sim$4,500 onward in all seeds; the follow-up reproduces the ordering reversal within its own schedule (Figure 4). The protocols agree in direction, not in crossing time. The follow-up provides a budget-level explanation: the $\rho = 64$ arm's training loss saturates by step $\sim$2,450, after which its radial alignment nearly stops and its Transmission RMS^2 shrinks under weight decay at the still-high learning rate, while the $\rho = 16$

arm keeps accumulating and overtakes. The terminal dose peak therefore records when each arm stops receiving scale-related work within the fixed horizon ; this is a finite-horizon saturation effect. The S-response inversion of §4.1 is itself scheduled, appearing first in the highest-repetition row (Figure 8). The gap, in contrast, rises monotonically with repetition throughout.

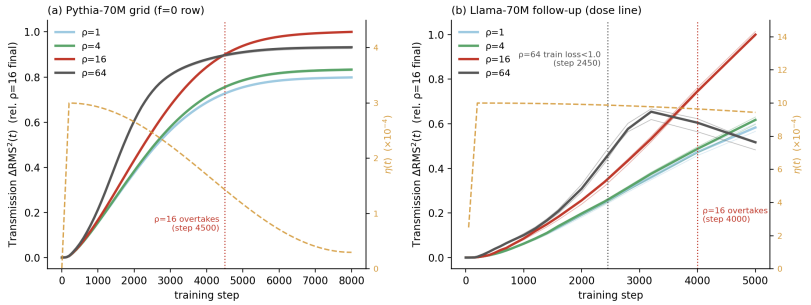


Figure 4: **The repetition-dose peak moves during training in both protocols.** Curves show measured-Transmission $\Delta\mathrm{RMS}^2(t)$, normalized by each protocol's terminal ρ=16 median; thin lines are seeds (visible in b), thick lines medians, and the dashed orange curve (right axis) is $\eta(t)$. **(a)** In the main grid, the peak moves from ρ=64 to 16 between steps 4,000 and 4,500. **(b)** The follow-up shows the same ordering reversal on its own schedule; the grey dotted line marks the step (2,450) at which the ρ=64 arm's median train loss falls below 1.0. Panel magnitudes and crossing times are protocol-specific.

4.5 Cross-corpus transport under matched interventions

The response ordering is retained on C4 and Python code, but code falls $21\% \pm 1\%$ below the D-based expectation at $\rho = 4$ (Figure 5a). Code also has the highest native redundancy ($R = 0.773$ at $\rho = 4$, above the 0.750 tiling floor): raising ρ from 4 to 64 at $f = 0$ increases $\Delta\lambda^2$ by only 0.07×10^{-4} on code, against 0.30 and 0.46×10^{-4} on wikitext and C4. This matches the direction, but not the size, of the residual. The residual is plausible because D summarizes only one-step token predictability and R counts exact repeated windows: corpora with similar D can still differ in token-frequency concentration, higher-order and long-range structure, and near-duplication. The experiment identifies a missing cross-corpus coordinate but does not determine which of these properties accounts for the difference. On the behavioral axis, the matched gap stays below threshold at $\rho = 4$ and rises far above it at $\rho = 64$ on all three corpora (Figure 5b), and all 24 cross-corpus runs fall in the expected regions of the effort–gap plane (Figure 5c). The window-shuffle generator supplies the cross-generator check of §4.1.

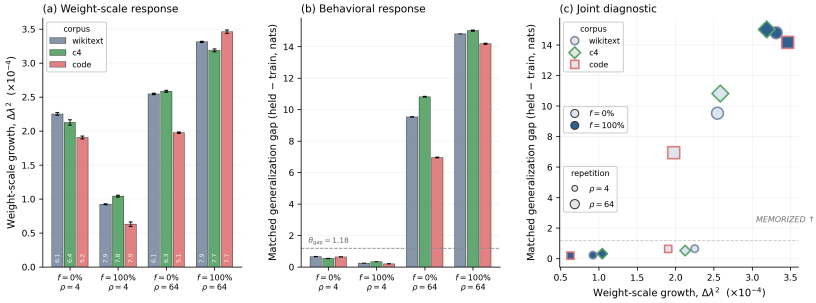


Figure 5: **Cross-corpus transfer preserves the two-coordinate separation and shows a residual of the D-only scale relation.** **(a)** Weight-scale growth across the four cross-corpus cells $\{f = 0, 100\%\} \times \{\rho = 4, 64\}$ on wikitext, C4, and Python code; numbers inside the bars report the measured D. **(b)** Matched generalization gap for the same cells. **(c)** The 24 cross-corpus runs (two corpora, four cells, three seeds) in the effort–gap plane alongside the corresponding wikitext conditions. Bars show means ± s.d. over three seeds; markers in (c) show the corresponding means.

4.6 LEARNING-RATE AND PROTOCOL CHECKS

Halving the learning rate preserves the ordering of the tested cells while reducing $\Delta\lambda^2$ by about $3\times$; the matched gap changes much less (Figure 10). This difference is expected because $\Delta\lambda^2$ is an optimizer-path-dependent scale readout, whereas the gap evaluates the fitted model under matched held-out data, so the two coordinates need not inherit the same learning-rate scaling. The main grid and component-resolved follow-up also show the same within-protocol movement of the repetition-dose peak. Their absolute magnitudes and crossing times differ because the architectures and schedules differ.

4.7 FIXED-BUDGET MODEL-SIZE SCREEN

Does the decisive counterexample survive when the model grows? A fixed-budget screen re-runs the four corner cells at Pythia-70M, 160M, and 410M with one seed each (Figure 6; Appendix B.7). The shuffled-and-repeated cell remains the joint maximum of $\Delta\lambda^2$ and gap at every size, and the shuffled-unrepeated cell stays near zero. What changes is amplitude: from 70M to 410M the three active responses fall by 24–41%, and the two structured cells exchange order at 410M, while each response reaches 90% of its terminal value at similar steps. Larger models fit the repeated streams more completely, but their held-out losses and gaps remain far above the clean condition. The smaller scale response is consistent with spreading a fixed token budget over a larger parameter set, so that the model improves its fit with less average radial scale change; the screen changes depth, width, and the token-to-parameter ratio together, however, and does not identify this dilution as the unique cause.

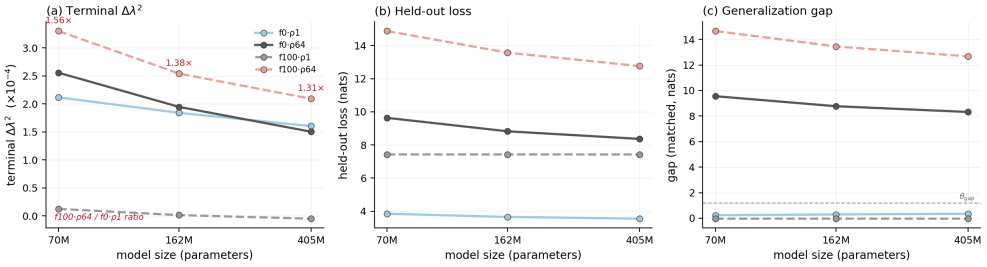


Figure 6: **The effort–quality separation persists across model sizes.** Terminal $\Delta\lambda^2$, held-out loss, and matched gap for the four corner conditions at 70M, 160M, and 410M under a fixed token budget. One run per condition and size; trajectories and t_{90} in Appendix B.7.

5 DISCUSSION

The experiments show why neither data nor checkpoint state is well described by one scalar. The interventions are the shuffle fraction f and repetition factor ρ; D, S, and R describe predictability, retained structure, and redundancy in the resulting streams. On the weight side, λ measures matched scale change, H^W measures row-scale heterogeneity, and k summarizes central-body shape. The coordinates answer different questions and need not move together.

The results are also dynamical. Data and the current weights determine the gradients, AdamW integrates their history, and the updated weights change later gradients. Repetition can therefore alter the amount of scale-related work, its component destination, and the point in training at which it stops accumulating. A terminal λ value is one endpoint of this process, not a fixed property of a corpus; intermediate checkpoints are needed to identify when accumulation, routing, or saturation changed.

The fixed-budget size screen separates response identity from response magnitude. Across 70M–410M, the decisive shuffled-and-repeated condition and the near-zero shuffled-and-unrepeated condition persist, and radial alignment remains the dominant budget component. The fitted-scale plateaus and terminal losses nevertheless change with size, and the structured conditions exchange

endpoint order at 410M. Thus the same qualitative training pattern can survive while its local weight magnitude changes.

Finally, similar measured weight states can accompany very different generalization gaps (approximately 12 nats in the follow-up). The weight coordinates reveal how training activity accumulated; matched held-out behavior is still needed to determine the outcome of that activity.

6 LIMITATIONS

Three limitations remain. **First, the causal chain is only partly resolved.** The randomized variables are shuffle fraction f and repetition ρ. D is a tokenizer-dependent one-step conditional entropy, S records retained local order, and R counts exact repeated windows; they do not cover the semantic and long-range structure of natural corpora. Data descriptors, optimizer terms, component routing, and the weights also change together, so the present experiments identify total intervention effects and several dynamical relations rather than a unique mediation path. This is why the Python-code residual is measured but not quantitatively explained.

Second, the transport tests cover a limited set of settings. The main grid and component-resolved follow-up answer complementary questions under two architectures and schedules; their within-protocol patterns agree, but absolute magnitudes and crossing times are setting dependent. The timing analysis uses fixed training horizons. The 70M–410M experiment is a one-seed, fixed-token screen and does not separate parameter count from depth, width, token-to-parameter ratio, or compute.

Third, the measurements are incomplete. I_{align} measures only the radial, norm-changing part of the update and does not capture tangential reorganization, which changes weight directions without changing their norm; fitted scale can decouple from norm growth when shape or row-scale mixtures change, and similar weight-side readouts can arise from different optimizer-budget compositions. The weight coordinates λ, k, and H^W omit signs, correlations, spectral directions, and activation interactions. On the behavioral side, the three regimes rely mainly on a matched train–held gap, with limited syntax and extraction probes. The three regimes are clearest in this grid because f and ρ deliberately create separated training conditions. Unmodified corpora may instead occupy a continuum and need not produce the same bimodal gap or relative-scale distributions. The numerical thresholds are therefore operational boundaries for the present protocol, not universal cutoffs. The C4 and Python-code experiments test transport under the same corner interventions, rather than classification of ordinary, unmodified corpora. The low-growth boundary is further supported by only two design cells. The row-level mechanism behind component routing also remains open.

7 CONCLUSION

Weight growth does not by itself mean learning. In our fixed-AdamW experiments, $\Delta\lambda^2$ closely follows accumulated radial-alignment work, yet the largest growth occurs in the worst-generalizing condition. Data interventions change when this scale-related work accumulates and how it is routed across components. Pairing the weight response with matched held-out behavior separates learning, memorization, and nothing-to-learn, and the same separation is retained under matched corner interventions on two additional corpora. A preliminary fixed-budget size screen preserves the decisive and null conditions while changing the response magnitude. Together, the experiments connect controlled data construction, optimizer dynamics, weight evolution, and behavior. Weight-scale growth is useful for describing training activity, but cannot rank learning quality on its own.

REPRODUCIBILITY STATEMENT

Tables 1 and 3 specify the experimental protocols. The appendices provide data construction, hyperparameters, threshold fitting, per-seed results, trajectories, budget checks, initialization hashes, and leakage tests. Corpus construction is deterministic and seeded; held-out streams are structure matched and leakage checked. The anonymized supplement contains the construction, training, evaluation, analysis, and figure code together with the per-run records used in the paper. These materials will be released publicly upon acceptance.

AI USE STATEMENT

We used a general-purpose large-language-model assistant for the following purposes. It gave feedback on research methodology and experimental design, in response to designs proposed by the authors. It assisted with code implementation, principally the training, logging, and analysis scripts written to author-specified procedures. It reformatted and reorganized recorded run outputs into the reported tables. It assisted with the analysis of recorded results. It assisted with translation and language polishing of author-drafted text, with result plotting and presentation, and with searching for and formatting related literature. We did not use generative AI to generate synthetic data sets or any corpus content: every training stream in this work is a deterministic, seeded transformation of a public corpus, specified by the construction code released in the supplementary material.

We have reviewed all AI-assisted work. All experiments were launched and supervised by the authors. The experimental record is fully traceable: every reported number resolves to a logged run output, and the construction, training, analysis, and figure scripts that produced them are released in the supplementary material; the figure and analysis scripts were re-executed from the archived data during manuscript preparation. All cited references were checked against their primary sources, and all scientific claims, evidence gradings, and scope limitations were decided by the authors. We take responsibility for the final content of this work.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. arXiv:2302.07011.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. arXiv:1706.05394.

Peter L. Bartlett, Dylan J. Foster, and Matus J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017. arXiv:1706.08498.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, 2023. arXiv:2304.01373.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021. arXiv:2012.07805.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations*, 2023. arXiv:2202.07646.

Jessica Chudnovsky, Joshua Kazdan, Noam Levi, Rylan Schaeffer, Yegor Denisov-Blanch, Bo He, Mehmet Donmez, Sanmi Koyejo, and David Donoho. Internal data repetition destroys language models. *arXiv preprint arXiv:2606.24998*, 2026.

CodeParrot (Hugging Face). CodeParrot-clean-valid: deduplicated Python code from GitHub. Hugging Face dataset, `https://huggingface.co/datasets/codeparrot/codeparrot-clean-valid`, 2022.

Tiexin Ding. A two-parameter Weibull framework for diagnosing transformer weight distributions. *arXiv preprint arXiv:2605.18898*, 2026a.

Tiexin Ding. Weibull weight-scale parameter evolution under AdamW training dynamics. *arXiv preprint arXiv:2606.19367*, 2026b.

Tiexin Ding. Data predictability shapes Weibull weight-scale growth in transformer training. *arXiv preprint arXiv:2608.23573*, 2026c.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2020. arXiv:1906.05271.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020. arXiv:2008.03703.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling laws and interpretability of learning from repeated data. *arXiv preprint arXiv:2205.10487*, 2022.

Tzu-Heng Huang, Aditya Goyal, John Cooper, and Frederic Sala. WARP: Weight-space analysis for recovering training data portfolios, 2026. ICML 2026 Workshop on Weight-Space Symmetries; arXiv:2607.01686.

A. R. Jonckheere. A distribution-free k-sample test against ordered alternatives. *Biometrika*, 41 (1/2):133–145, 1954.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022. arXiv:2202.06539.

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 (ICML)*, 2024. arXiv:2305.17212.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.06110.

Ziming Liu, Eric J. Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.01117.

Matthew Loftus. Spectral signatures of data quality: Eigenvalue tail index as a diagnostic for label noise in neural networks. *arXiv preprint arXiv:2603.27885*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. arXiv:1711.05101.

Hartmut Maennel, Ibrahim Alabdulmohsin, Ilya Tolstikhin, Robert J. N. Baldock, Olivier Bousquet, Sylvain Gelly, and Daniel Keysers. What do neural networks learn when trained with random labels? In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020. arXiv:2006.10455.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021a. arXiv:1810.01075.

Charles H. Martin and Michael W. Mahoney. Post-mortem on a deep learning contest: a Simpson's paradox and the complementary roles of scale metrics versus shape metrics. *arXiv preprint arXiv:2106.00734*, 2021b.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1609.07843.

William Merrill, Vivek Ramanujan, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. Effects of parameter norm growth during transformer training: Inductive bias from gradient descent. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. arXiv:2010.09697.

John X. Morris, Chawin Sitawarin, Chuan Guo, Narine Kokhlikyan, G. Edward Suh, Alexander M. Rush, Kamalika Chaudhuri, and Saeed Mahloujifar. How much do language models memorize? *arXiv preprint arXiv:2505.24832*, 2025.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023. arXiv:2305.16264.

Vaishnavh Nagarajan and J. Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:1912.02292.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1707.09564.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 2020. arXiv:1910.10683.

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

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022. arXiv:2205.10770.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Twan van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of normalized neural network using SGD and weight decay. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021. arXiv:2006.08419.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. arXiv:1912.00582.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1611.03530.

A METHODS AND REPRODUCIBILITY

A.1 EXPERIMENTAL CONFIGURATION AND REPRODUCIBILITY

Table 3: **Implementation details for the main grid and component-resolved follow-up.** Scientific roles and primary readouts are summarized in Table 1.

Implementation detail	Main $f\times\rho$ grid	Component-resolved follow-up
Model	Pythia-70M (GPT-NeoX; Black et al., 2022): 6 layers, d=512, 8 heads, FFN 2048, rotary 0.25 (Su et al., 2024), fused QKV	Llama-style 70M: 6 layers, d=512, FFN 1376, separately stored q/k/v/o
Initialization	HuggingFace GPT-NeoX default sampler, built offline from the config ($\lambda_0 = 0.01776$; measured $k_0 = 1.206$)	per-tensor Gaussian inverse-CDF draw, mean-centered and rescaled to $\sigma = 0.02$ exactly (measured $k_0 = 1.204$); identical init hashes across paired arms
Schedule	batch 24 $\times$ seq 512; 200-step warmup, cosine to the $0.1\,\eta_{\text{peak}}$ floor	batch 24 $\times$ seq 512; 200-step warmup, first 5,000 steps of a 30,000-step cosine, same floor
Batching / index schedule	random 512-token windows sampled with replacement from the training stream	sequential batches on an arm-independent index schedule, so arms differ only in stream content
AdamW implementation	`torch.optim.AdamW`, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, decoupled decay, no gradient clipping	same optimizer and β, ϵ; decoupled decay, no gradient clipping
Logging cadence	alignment / injection / decay forces and weight RMS every 100 steps; 18 Weibull checkpoints; 18 weight checkpoints	per-step force and RMS accumulation for every analyzed matrix; 21 state snapshots; 7 weight checkpoints
Numerical precision	fp32 training loop, no mixed-precision autocast	fp32 training loop; per-component force and update-norm accumulators carried in fp64
Determinism	token streams generated by seeded operations and held fixed across seeds, so seeds vary only initialization and batch order	fixed token stream and batch-index schedule across arms; seeds vary only initialization
Acceptance checks	instability guard $F_{\text{align}} < -10$, triggered on 0/76 runs under this protocol	signed-budget closure per run and channel; paired-init hash equality; zero held-out leakage
Hardware	single NVIDIA RTX 4090	single NVIDIA RTX 4090

Auxiliary runs inherit their parent protocol. Cross-corpus runs change the corpus, the $\eta/2$ control changes the peak learning rate, and the extended shuffled run changes only the training horizon (5,000 $\rightarrow$ 8,000 steps).

Follow-up arms. The component-resolved follow-up contains four corner arms, D1/D2/D3/D4 $= (f0\rho1,\ f100\rho1,\ f100\rho64,\ f0\rho64)$, and two intermediate structured arms, D5/D6 $= (f0\rho4,\ f0\rho16)$. All contrasts use paired initializations across arms, so a difference between two arms reflects their data streams rather than their starting weights. Per-parameter effort i_{align} normalizes by matrix size and is compared across components; total effort I_{align} is compared across arms.

Size screen. The third protocol re-runs the four corner cells of the main grid at Pythia-70M, 160M, and 410M under the same token budget, streams, and schedule, with one seed per cell. Table 4 lists what it inherits from the main protocol, what changes, and the two derived readouts (t_{90} and the per-interval alignment share) used in Section 4.7 and Appendix B.7.

Table 4: Design of the fixed-budget size screen relative to the main grid.

Item	Specification
Varied factor	Pythia-70M / 160M / 410M ($N = 70$ / 162 / 405M; $N/T = 0.72$ / 1.65 / 4.12)
Tested cells	Four corners $\{f{=}0, 100\%\} \times \{\rho{=}1, 64\}$
Inherited protocol	Same seeded token streams (identical hashes), 98.3M-token / 8,000-step budget, batch 24×512, AdamW $\eta_{\text{peak}}{=}3 \times 10^{-4}$, $\lambda_{\text{wd}}{=}0.01$ with the same schedule, 18 weight checkpoints
Replication	One run (seed 0) per cell and size; 70M re-executed on the same hardware as the larger models, reproducing the main-grid $\lambda(t)$ to five significant digits
Primary readouts	Pooled Transmission $\Delta\lambda^2$, train loss, matched held-out loss, matched gap
Scale timing	t_{90}: first step at which $\Delta\lambda^2(t)$ reaches 90% of its own step-8,000 value (interpolated between checkpoints; undefined for the near-zero shuffled-unrepeated arm)
Budget share	$s_{\text{align}}(t) = \lvert\Delta I_{\text{align}}\rvert / (\lvert\Delta I_{\text{align}}\rvert + \Delta I_{\text{inj}} + \lvert\Delta I_{\text{decay}}\rvert)$, with ΔI the per-logging-interval contribution of each channel
Learning-rate control	One 160M shuffled$\times$64 run at $\eta/2$
Stability check	Instability guard of Table 3, $\max_t[-F_{\text{align}}(t)]$, evaluated per Transmission parameter ($N_c = 9H^2L$: 14.2M / 63.7M / 226.5M) and referred to the 70M threshold: every run stays below it, with no loss spike after step 2,000
Identification boundary	Size jointly changes depth, width, N_c, N/T and compute

A.2 Data construction and measured coordinates

Main grid. The main grid uses $f \in \{0, 0.25, 0.50, 1\}$ and $\rho \in \{1, 4, 16, 64\}$. Each wikitext-103 (Merity et al., 2017) cell is generated as

$$U = T/\rho, \qquad X_{f,\rho} = \text{Tile}_\rho\big(\text{Shuffle}_f(X_{1:U})\big), \tag{1}$$

where Shuffle_f permutes a sampled fraction f of token positions among themselves, preserving unigram counts, and Tile_ρ repeats the resulting U-token block ρ times. Every cell consumes $T = 98{,}304{,}000$ tokens; increasing ρ therefore increases exposure while reducing unique content. Streams are generated deterministically from fixed seeds and are identical across seeds, so seed variation isolates initialization and batch order.

Held-out sets and cross-corpus arms. Two 128k-token evaluation streams are taken from a wikitext block disjoint from all training cells. The clean stream is a 128k-token block beginning at token 110M. For each $f > 0$, a structure-matched stream is produced by applying Shuffle_f with an independent seed to the disjoint 4M-token tail block and taking its first 128k tokens; it is not tiled. Matching f reduces the structural distribution shift introduced by shuffling. The same generation rule is applied to C4 (Raffel et al., 2020) and deduplicated Python code (codeparrot-clean-valid; CodeParrot (Hugging Face), 2022), each subsampled to 30M tokens, for the four cross-corpus cells $\{f = 0, 100\%\} \times \{\rho = 4, 64\}$ at three seeds each.

Measured coordinates. D is the plug-in bigram conditional entropy $H(\text{next} \mid \text{prev})$, estimated on a fixed 4M-token block before tiling. S is the retained-bigram fraction, equal to $(1-f)^2$ under global shuffling. R is the fraction of length-32 token windows in the complete tiled stream that already occurred earlier in the stream, measured with rolling hashes. These are measured descriptions of the generated streams; the experimental interventions are f and ρ. Table 5 reports the measured cross-effects of the two interventions on these descriptors.

Window-shuffle reference series. The reference series consists of the seven single-seed window-shuffle runs reported by Ding (2026c). That protocol shuffles fixed-length windows of tokens, keeping the order inside each window, and tiles a 30M-token wikitext block to the same 98M-token budget, an implicit repetition of $\approx 3.3\times$. Our grid instead shuffles individual token positions and sets repetition explicitly through ρ. The reference series is therefore compared with our $\rho = 4$ row, and its S is measured directly on the generated streams, since the closed form $S = (1 - f)^2$ applies only to global shuffling. The held-out check interpolates the wikitext $\rho = 4$ row linearly in S (and, for comparison, in D, clamped at the grid ceiling) at the held-out run's measured coordinates.

Table 5: Cross-effects of the two interventions on the measured descriptors. Correlations are descriptive values over the generated cell-level streams; no inferential test is attached to the four-cell cross-corpus grids.

Corpus	$\lvert\mathrm{corr}(D,\rho)\rvert$	$\lvert\mathrm{corr}(R,f)\rvert$
wikitext-103	0.00	0.00
C4 (4 cells)	0.11	0.00
Python code (4 cells)	0.05	0.05

A.3 Evaluation and weight-side measurements

A.3.1 Behavioral evaluation protocol

At the final main-grid checkpoint, held-out losses are token-weighted mean next-token cross entropies over the 128k-token evaluation streams, evaluated in non-overlapping 512-token blocks, and train loss is the mean over the final 100-step logging interval. Table 6 defines the quantities used in the classification; the follow-up evaluates the same matched gap at its scheduled checkpoints. BLiMP reports accuracy over 67 grammatical minimal-pair subsets at the final checkpoint for all three seeds. The extraction probe prompts the seed-0 model of each cell with 100 training prefixes of 32 tokens and records exact reproduction of the next 64 tokens under greedy decoding. The corresponding behavioral results are reported in Appendix B.2.4.

Table 6: Behavioral and relative-scale quantities used in the classification.

Quantity	Data source	Use
ℓ_{train}	final 100-step training-log mean	training fit
$\ell_{\text{held,clean}}$	clean disjoint 128k-token stream	absolute held-out behavior
$\ell_{\text{held,matched}}$	same-f, independently shuffled 128k-token stream	matched comparison
g_{matched}	$\ell_{\text{held,matched}} - \ell_{\text{train}}$	primary classification axis
r_λ	$\Delta\lambda^2/\Delta\lambda^2_{\text{ref}}$	low-growth classification axis

A.3.2 Weight-side coordinates

We fit a two-parameter Weibull to matrix-entry magnitudes over the middle-80% probability-plot range, which excludes the near-zero and extreme tails. The fit returns the scale λ and the central-body shape k. Row-scale heterogeneity H^W is the standard deviation of role-aligned per-row log-RMS. The measured Transmission subset contains the FFN and attention-output matrices; Selection contains the query and key matrices; value matrices are excluded from both aggregates. As a direct pooled second-moment check, we also compute the change in the exact pooled second moment, $\Delta\mathrm{RMS}^2$, over the same Transmission matrices.

A.3.3 AdamW signed budget and logging

Following Ding (2026b), the per-step squared-norm update is recorded as an alignment force $F_{\text{align}} = -2\eta\langle W, \hat{u}\rangle$ (positive when the adaptive update contributes to squared-norm growth), an injection force $F_{\text{inj}} = \eta^2\|\hat{u}\|^2$, and a decay force $F_{\text{decay}} = -2\eta\,\lambda_{\text{wd}}\sum|W|^2$, where $\hat{u} = \hat{m}/(\sqrt{\hat{v}} + \epsilon)$ is the actual AdamW step direction read from the live moment estimates. The decomposition is a leading-order relation: terms of order $\eta^2\lambda_{\text{wd}}$ are neglected. Each force is an instantaneous per-step rate of squared-norm change, so the accumulated alignment work is a step-weighted integral over training time rather than an unweighted sum of the sparsely logged samples. In the component-resolved follow-up every step is accumulated, $I_{\text{align}}(T) = \sum_{t<T} F_{\text{align},t}$ with $\Delta t = 1$. In the main grid the forces are sampled on the logging grid of Table 3, and the integral is taken as the left-endpoint Riemann sum $I_{\text{align}}(T) = \sum_b F_{\text{align},b}\,\Delta t_b$, where Δt_b is the number of steps between consecutive logged samples. Both are evaluated over the same matrix set used for λ.

A.4 Outcome classification and statistics

A.4.1 Three-regime rule

With g_{matched} the matched gap and $r_\lambda = \Delta\lambda^2/\Delta\lambda^2_{\text{ref}}$ the relative scale growth,

$$\text{label} = \begin{cases} \text{MEMORIZED}, & g_{\text{matched}} > \theta_{\text{gap}}, \\ \text{NOTHING-TO-LEARN}, & g_{\text{matched}} \le \theta_{\text{gap}} \text{ and } r_\lambda < \theta_r, \\ \text{LEARNED}, & g_{\text{matched}} \le \theta_{\text{gap}} \text{ and } r_\lambda \ge \theta_r. \end{cases} \tag{2}$$

The gap is primary because weight motion measures effort, which memorization also produces. The reference $\Delta\lambda^2_{\text{ref}}$ provides a within-corpus, within-learning-rate normalization of absolute scale growth.

A.4.2 Threshold fitting and reference mapping

The thresholds are fitted once on the 48 main-grid runs. A two-component Gaussian mixture is fitted to $\log(1 + g_{\text{matched}})$ and its equal-posterior boundary gives $\theta_{\text{gap}} = 1.18$ nats. In a second stage, restricted to the 24 runs below this boundary, a second mixture is fitted to $\log r_\lambda$, giving $\theta_r = 0.70$. These operational thresholds summarize the separation observed in the main grid; their stability and frozen application to external runs are reported in Appendix B.2.1.

The reference is chosen within corpus, seed, and learning rate. The main grid and the $\eta/2$ control use the $f0 \cdot \rho1$ cell of the same seed and learning rate. The C4 and Python-code corner grids use their $f0 \cdot \rho4$ cell of the same seed, because no $\rho = 1$ cell was run on those corpora. The fitted thresholds themselves are not refitted outside the main grid.

A.4.3 Statistical tests

Main-grid values are reported as mean $\pm$ s.d. over three seeds. Monotone repetition trends use the one-sided Jonckheere–Terpstra permutation test (Jonckheere, 1954) with 2,000 permutations. Partial correlations regress out D and R; 95% confidence intervals use 5,000 bootstrap resamples.

B SUPPORTING RESULTS

B.1 DATA-COORDINATE RESPONSE (SECTION 4.1)

Table 7 gives the cell-level values behind Figures 1 and 2; its last column, $I_{\text{align}}/\Delta L_{\text{train}}$, divides the accumulated alignment work by the reduction in terminal training loss from initialization. Under the window-shuffle interpolation of Appendix A.2, the relative errors of the held-out prediction are 5% for S and 45% for D.

Table 7: **Main $f \times \rho$ grid: data, weight, and behavioral readouts** (Pythia-70M, wikitext-103). Seed-aggregated quantities are means over three seeds; s.d. is shown for $\Delta\lambda^2$ and the matched gap. D is in bits, $\Delta\lambda^2$ in 10^{-4}, losses and matched gap in nats, and I_{align} in a.u. $S = (1-f)^2$ is given by the row label.

Cell	D	R	$\Delta\lambda^2$	$\text{gap}_{\text{matched}}$	train_loss	held_clean	I_{align}	$I_{\text{align}}/\Delta L_{\text{train}}$
f000·ρ1	6.095	0.017	2.122 ± 0.004	0.23 ± 0.01	3.63	3.86	4368	599
f000·ρ4	6.095	0.752	2.250 ± 0.020	0.66 ± 0.01	3.39	4.05	4602	611
f000·ρ16	6.095	0.938	2.765 ± 0.011	4.26 ± 0.04	1.65	5.90	5631	607
f000·ρ64	6.095	0.984	2.546 ± 0.011	9.53 ± 0.02	0.09	9.62	5189	479
f025·ρ1	7.206	0.000	2.061 ± 0.008	0.16 ± 0.01	5.84	4.51	4257	836
f025·ρ4	7.206	0.750	2.251 ± 0.010	0.56 ± 0.00	5.59	4.65	4600	863
f025·ρ16	7.206	0.937	3.076 ± 0.006	4.90 ± 0.03	3.24	6.44	6245	813
f025·ρ64	7.206	0.984	3.030 ± 0.006	13.07 ± 0.02	0.14	11.17	6183	573
f050·ρ1	7.711	0.000	1.626 ± 0.008	0.07 ± 0.01	6.96	5.43	3434	867
f050·ρ4	7.711	0.750	1.920 ± 0.005	0.38 ± 0.01	6.77	5.50	3970	954
f050·ρ16	7.711	0.937	3.137 ± 0.023	4.84 ± 0.03	4.23	7.26	6328	944
f050·ρ64	7.711	0.984	3.217 ± 0.005	14.53 ± 0.03	0.18	12.68	6571	612
f100·ρ1	7.886	0.000	0.135 ± 0.008	0.00 ± 0.00	7.46	7.42	325	94
f100·ρ4	7.886	0.750	0.922 ± 0.007	0.24 ± 0.00	7.32	7.51	2023	562
f100·ρ16	7.886	0.937	2.992 ± 0.004	4.45 ± 0.02	4.80	9.17	6034	985
f100·ρ64	7.886	0.984	3.311 ± 0.008	14.81 ± 0.01	0.22	14.87	6746	630

The matched gap uses the structure-matched held-out loss, whereas `held_clean` reports performance on the disjoint clean block.

B.2 EFFORT AND LEARNING QUALITY (SECTION 4.2)

B.2.1 THRESHOLD STABILITY AND TRANSPORT

Section 4.2 uses the two operational thresholds defined in Appendix A.4. Here we test whether the resulting labels depend on their exact fitted values. In the 48-run main grid, no observations lie between 0.67 and 4.23 nats on the gap axis or between 0.44 and 0.76 on the r_λ axis; moving either threshold anywhere within the corresponding empty interval therefore leaves all labels unchanged. The three seeds agree in every one of the 16 cells. The labels also remain unchanged under leave-one-seed refitting and under all $3^3 = 27$ paired-seed resamples applied jointly to the 16 cells.

We then apply the thresholds without refitting to 28 external runs: 24 cross-corpus runs and four learning-rate controls. The nearest external observations bracketing the thresholds are 0.65 and 6.93 nats on the gap axis, and 0.50 and 1.00 on the r_λ axis; all 28 external labels are also unchanged under each leave-one-seed refit. Leave-one-cell-out refitting preserves the memorization boundary but changes the low-growth label in 3 of 16 folds. This boundary is determined by only two NOTHING-TO-LEARN cells, $f100 \cdot \rho 1$ and $f100 \cdot \rho 4$, the only six runs below it. The memorization boundary is therefore well separated, whereas the low-growth boundary is stable in the observed runs but remains design-dependent. The external results test transport of the frozen rule, not independent calibration (Table 8).

Table 8: Audit of the two classification boundaries. Flips count runs whose label changes relative to the published thresholds. External transport re-evaluates the same 28 cross-corpus and η-control runs under each leave-one-seed refit; they are not independent validation samples. Per-fold thresholds and the individual flipped cells are in the reproducibility package.

Audit	Gap boundary	Low-growth boundary	Interpretation
Leave-one-seed-out	0 flips	0 flips	seed-stable
Paired-seed resampling (3^3)	$[1.177, 1.195]$	$[0.700, 0.704]$	sampling range
Leave-one-cell-out	0 flips	3/16 folds flip	design dependence
Frozen external transport	all 28 stable	all 28 stable	transport, not calibration

B.2.2 Scale-readout agreement

The direct pooled second-moment change $\Delta\mathrm{RMS}^2 = \mathrm{RMS}^2(8{,}000) - \mathrm{RMS}^2(1)$ provides a second scale readout of the same measured Transmission weights. It shows the same main pattern as $\Delta\lambda^2$: $f100 \cdot \rho64$ is the largest cell under both measures. Its mean $\Delta\mathrm{RMS}^2$ is 4.41×10^{-4}, compared with 4.29×10^{-4} for $f50 \cdot \rho64$, and it remains the largest cell in each seed; the paired difference is positive in all three seeds under both readouts.

Across the 16 cells, the $\Delta\mathrm{RMS}^2$ and $\Delta\lambda^2$ rankings agree closely ($r_s = 0.997$). They differ only in the adjacent pair $f0 \cdot \rho4$ and $f25 \cdot \rho4$, whose means differ by 0.3%. Per-matrix Weibull fits also preserve the pooled cell ordering (Spearman $\rho = 0.97$ for the median over the 18 measured matrices, seed 0; 0.99 for the mean). The scale pattern is therefore consistent across direct and fitted readouts, pooled and per-matrix analyses, and all three seeds.

We use λ rather than RMS alone because the Weibull fit separates central-body scale from shape. Under stable k, $\Delta\lambda^2$ provides a protocol-matched checkpoint readout of scale motion, and $\Delta\mathrm{RMS}^2$ supplies a direct whole-distribution check. When k, fit quality, or H^W changes, the two readouts need not agree and λ should not be interpreted alone (Appendix B.3). Unlike I_{align}, which requires instrumented training, λ can be obtained from stored checkpoints. The direct readout thus confirms the scale phenomenon, λ places it in a joint shape–scale diagnostic, and I_{align} explains how it forms in the present fixed-AdamW experiments.

The per-seed $\Delta\lambda^2$ values are reported in Table 9. Their median cell-level CV is 0.4%; the largest CV, 5.7%, occurs at $f100 \cdot \rho1$, where the absolute growth is close to zero.

Table 9: Per-seed $\Delta\lambda^2$ ($\times 10^{-4}$) for all 16 cells — the 48 individual training runs.

cell	seed 0	seed 1	seed 2	cell	seed 0	seed 1	seed 2
f0·ρ1	2.1172	2.1262	2.1217	f50·ρ1	1.6149	1.6355	1.6276
f0·ρ4	2.2357	2.2785	2.2370	f50·ρ4	1.9156	1.9269	1.9160
f0·ρ16	2.7675	2.7778	2.7504	f50·ρ16	3.1576	3.1484	3.1039
f0·ρ64	2.5601	2.5448	2.5325	f50·ρ64	3.2236	3.2163	3.2111
f25·ρ1	2.0570	2.0721	2.0553	f100·ρ1	0.1252	0.1441	0.1365
f25·ρ4	2.2367	2.2600	2.2575	f100·ρ4	0.9115	0.9293	0.9238
f25·ρ16	3.0687	3.0836	3.0750	f100·ρ16	2.9889	2.9981	2.9886
f25·ρ64	3.0381	3.0228	3.0287	f100·ρ64	3.3015	3.3131	3.3199

B.2.3 AdamW budget attribution

This section uses the three logged AdamW forces of Appendix A.3, each integrated over the full 8,000-step run and summed over the measured Transmission subset, for the 16 seed-0 cells of the main grid. The question is which term of the signed budget $\Delta\|W\|^2 = I_{\text{align}} + I_{\text{inj}} + I_{\text{decay}}$ carries the differences in scale growth between cells.

Within this fixed-AdamW main grid, the answer is alignment. Its integrated magnitude varies $22.8\times$ across the 16 cells (CV 0.36), whereas decay varies only $1.63\times$ (CV 0.12), because decay is set by η, λ_{wd} and a weight norm that stays similar across cells. Injection is 3.8–5.3% of alignment in every cell, because both scale with the size of the adaptive updates. Alignment also accounts for

a median 92.3% of the gross budget $|I_{\text{align}}| + I_{\text{inj}} + |I_{\text{decay}}|$ (range 63.8–93.1%, the low end being the near-frozen $f100{\cdot}\rho1$ cell). Injection co-varies with alignment, while decay changes much less across cells; consequently I_{align} dominates the between-cell variation of $\Delta\|W\|^2$, and hence of $\Delta\lambda^2$ at stable shape, in the present grid, which is why $\Delta\lambda^2$ tracks I_{align} linearly even though the other two terms are not negligible individually.

These shares are horizon totals. Within a run the composition shifts toward decay late in training, especially in the repeated cells whose alignment stops once train loss saturates (§4.4; Figure 11d); this does not affect the between-cell comparison above. Nor does larger alignment work imply better learning: at $\rho = 64$, the fully shuffled cell accumulates more alignment work than the structured cell (6,746 versus 5,189 a.u.) despite much worse held-out behavior. In the component-resolved follow-up the signed budget closes to 0.957–1.010 across all 54 run×channel cases. Table 10 reports alignment work per nat of training-loss reduction.

Table 10: Alignment work per nat of terminal training-loss reduction across the main grid, $I_{\text{align}}/\Delta L_{\text{train}}$ (3-seed mean, a.u./nat). In the near-frozen $f100 \cdot \rho1$ cell the denominator is very small, so this ratio is not used to order cells overall. The main-text contrast is the $\rho = 16$ row, 985 versus 607.

ρ	f0 (clean)	f25	f50	f100 (noise)	Largest ratio
1	599	836	**867**	94	f50
4	611	863	**954**	562	f50
16	607	813	944	**985**	f100
64	479	573	612	**630**	f100

B.2.4 Behavioral signatures of the three regimes

Using the evaluation protocols of Appendix A.3.1, BLiMP (Warstadt et al., 2020) and verbatim extraction provide complementary behavioral evidence for the three regimes. BLiMP tracks generalizable syntactic structure. In each shuffle row, BLiMP accuracy is lower in both MEMORIZED cells than in both non-memorized cells (Figure 7); the group means follow LEARNED > NOTHING-TO-LEARN ≈ MEMORIZED, and the cell with the largest $\Delta\lambda^2$ scores near chance.

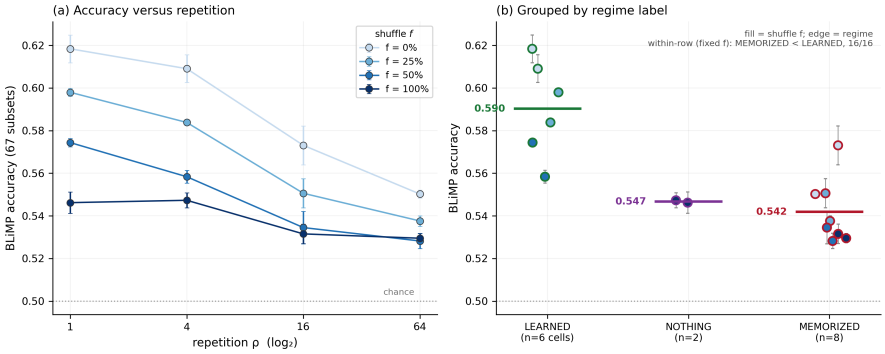


Figure 7: **The three regimes carry distinct behavioral signatures. (a)** BLiMP minimal-pair accuracy for all 16 wikitext cells (67 subsets; 3-seed mean). **(b)** The same 16 cells grouped by label ($n = 6/2/8$).

Verbatim extraction isolates literal storage. The seed-0 probe finds no exact 64-token reproduction at $\rho = 16$, whereas at $\rho = 64$ the extraction rate ranges from 57% to 94% across shuffle levels (Table 11). Exact reproduction requires all 64 greedy tokens to match, so it appears only once memorization is nearly complete: the $\rho = 64$ cells end with train loss below 0.25 nats, whereas the $\rho = 16$ cells, although MEMORIZED by their gap, end at 1.6–4.8 nats and reproduce nothing verbatim. The greedy per-token match rate stays at 0.02–0.04 in every $\rho \leq 16$ cell, clean or shuffled, so the step at $\rho = 64$ reflects literal storage rather than ordinary predictability.

Table 11: Verbatim extraction: fraction of training windows reproduced exactly by the 64-step greedy continuation probe (seed 0).

	$f = 0\%$	$f = 25\%$	$f = 50\%$	$f = 100\%$
$\rho = 16$	0%	0%	0%	0%
$\rho = 64$	94%	81%	66%	57%

B.3 Component routing (Section 4.3)

Fitted-scale calibration. For a full Weibull distribution the second moment is $\lambda^2\,\Gamma(1 + 2/k)$. Because our parameters are fitted on the middle-80% range, we use this relation as an empirical calibration bridge rather than an identity, and quantify the residual on the constant-η probe runs, which have dense RMS logging (13 λ anchors against a 137-point RMS trajectory each, two cells $\times$ two learning rates). In level, the Weibull-implied second moment $\lambda^2\,\Gamma(1 + 2/k)$ exceeds the measured RMS^2 by a stable 15–18%, a constant multiplicative offset attributable to the middle-80% fit. In growth, $\Delta(\mathrm{RMS}^2)/\Gamma$ tracks $\Delta\lambda^2$ with a per-anchor mean ratio of 0.896 ± 0.005 across all four runs and both learning rates. The two readouts track one another closely in these four probe runs, with a stable protocol-specific offset; the Weibull framing adds an explicit check of whether the fitted shape remains stable enough for λ to be read as a matched scale coordinate.

Second-moment bridge across arms. Table 12 reports the bridge in the follow-up arms. Structured arms share a stable bridge ratio of 1.11–1.15. D2's denominator is a net shrinkage under weight decay, so its ratio is reported on the shrinking side. D3's fitted scale captures only 20–60% of measured norm growth; the shortfall is not attributable to decay alone (candidates are row-scale mixing, changes outside the middle-80% fit window, and within-channel shape change); the present measurements do not distinguish among them.

Table 12: **Weibull second-moment bridge across the follow-up arms.** Per-seed ratio $N\,\Delta[\lambda^2\Gamma(1+2/k)]/\Delta\|W\|^2$ over the measured Transmission subset, steps 0→5,000; $\Delta\|W\|^2$ in 10^3 a.u. (seed median).

Arm	Condition	$\Delta\|W\|^2$	bridge/measured (per seed)	Reading
D1	$f0{\cdot}\rho1$	$+31.7$	1.11–1.12	tight, stable protocol factor
D5	$f0{\cdot}\rho4$	$+33.6$	1.11–1.12	tight
D6	$f0{\cdot}\rho16$	$+55.7$	1.15–1.15	tight
D4	$f0{\cdot}\rho64$	$+29.4$	1.11–1.11	tight
D2	$f100{\cdot}\rho1$	-3.1	1.26–1.32 (shrinking side)	decay-dominated net shrinkage
D3	$f100{\cdot}\rho64$	$+11.3$	**0.19–0.58**	fitted scale misses 42–81% of growth

Effort–quality decoupling. For each metric, we divide the median paired D4$-$D1 difference by the range of the four corner-arm medians, giving a within-follow-up standardized effect (Table 13). The comparison shows large behavioral separation but relatively small differences in the measured weight coordinates; it does not establish that the two weight states are identical.

B.4 Temporal evolution (Section 4.4)

Table 7 reports the terminal grid, but the structure–scale relation is not static during training. Figure 8 recomputes the slope against S at every logged checkpoint; the inversion order matches the memorization-onset order, and Figure 1b is a late slice of this process. In the structured row, the dose peak $\rho^*(t)$ selected by the direct $\Delta\mathrm{RMS}^2(t)$ (Figure 4a) is the same row as that selected by the fitted $\Delta\lambda^2$ at all 24 seed$\times$time points. Figure 9 reports the complete follow-up trajectories underlying Sections 4.3 and 4.4.

Table 13: **Behavioral and weight-side differences between D4 and D1.** Effects $E_j = \text{median}_s[x_{\text{D4},j,s} - x_{\text{D1},j,s}]/\text{scale}_j$, with scale_j the range of per-arm seed-medians over D1–D4, are normalized within the follow-up protocol; Seeds gives paired-seed sign agreement.

Tier	Metric	Effect	Seeds	Note
behavior	gap	$+1.00$	3/3	strong separation ($\sim$12 nats)
behavior	held-out loss	$+1.00$	3/3	strong separation
weight	i_{align} (Selection q/k)	$+0.18$	3/3	small–moderate
weight	i_{align} (meas. Trans.)	$+0.07$	3/3	small
weight	$\Delta\lambda^2$ (Selection)	$+0.09$	3/3	small
weight	$\Delta\lambda^2$ (meas. Trans.)	-0.15	3/3	small
state	H_q^W	$+0.10$	3/3	small
state	k_q (raw)	-0.08	3/3	small
state	k_q (row-normalized)	$+0.03$	2/3	negligible
state	late profile amplitude (q)	-0.23	3/3	largest weight-side entry

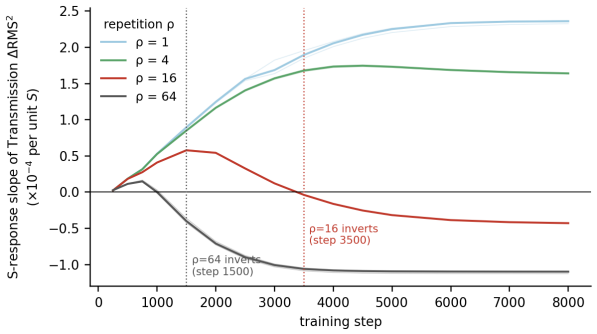


Figure 8: **The S-response inversion is scheduled, not static.** Slope of measured Transmission ΔRMS^2 against $S = (1-f)^2$ at each logged checkpoint, per repetition row (thin: seeds; thick: medians). The $\rho = 64$ row inverts by step $\sim$1,500 and $\rho = 16$ by $\sim$3,500; $\rho \leq 4$ stays positive.

B.5 Cross-corpus results (Section 4.5)

The D-calibrated expectation of §4.5 extrapolates the clean wikitext $\rho = 4$ cell along the D-slope of the unsaturated part of that row ($f \leq 50\%$; -0.171×10^{-4} per bit) to the code cell's measured $D = 5.20$, giving 2.40×10^{-4}; the measured code response falls 21% below it. Code's higher native redundancy is directionally consistent with this residual but does not explain its magnitude. Both interpolation rules are implemented in the supplementary material.

B.6 Learning-rate and protocol sensitivity (Section 4.6)

At $\eta/2$, cell ordering is preserved, $\Delta\lambda^2$ rescales within the known η-scaling band, and the gap changes by -5 to -9% in the $\rho = 64$ cells and by -13% in the clean $\rho = 1$ cell, relative to the three-seed mean at η (the $f100 \cdot \rho 1$ base is near zero; Figure 10). The two training protocols are compared by direction and within-protocol ordering only.

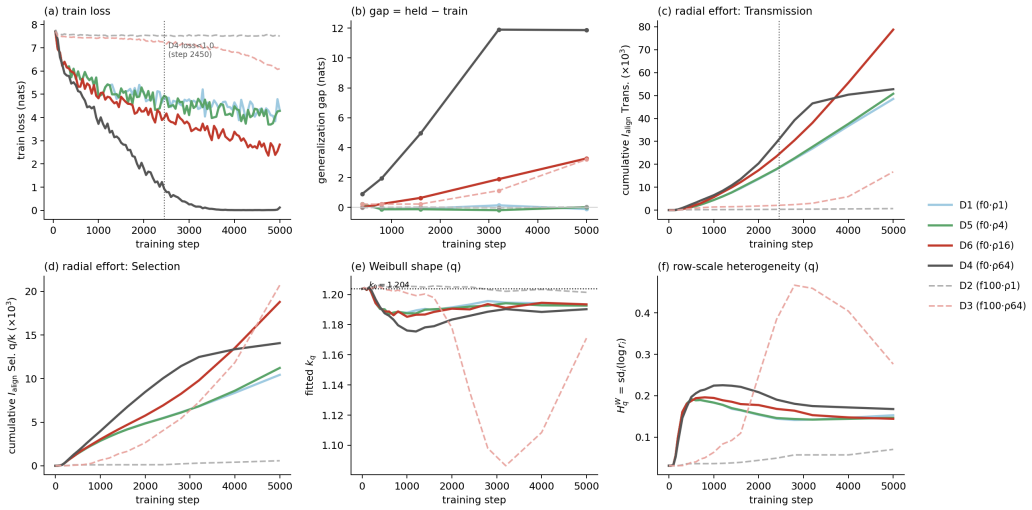


Figure 9: **Time-resolved behavior, effort and state coordinates for all six follow-up arms** (3-seed medians; solid = structured dose line $f{=}0$, dashed faint = shuffled controls $f{=}100$). **(a)** Train loss; the dotted line marks the $\rho{=}64$ arm's crossing of loss $= 1.0$ (step 2,450). **(b)** Generalization gap; its rise aligns with the train-loss breakaway in (a). **(c,d)** Cumulative radial-alignment effort by channel; the dotted line in (c) repeats the step-2,450 mark. **(e)** Fitted Weibull shape of the query projection (the key projection shows the same trend), against the protocol anchor $k_0 \approx 1.20$ (dotted line at the measured 1.204). **(f)** Row-scale heterogeneity H_q^W, q projection only. All six arms are shown throughout.

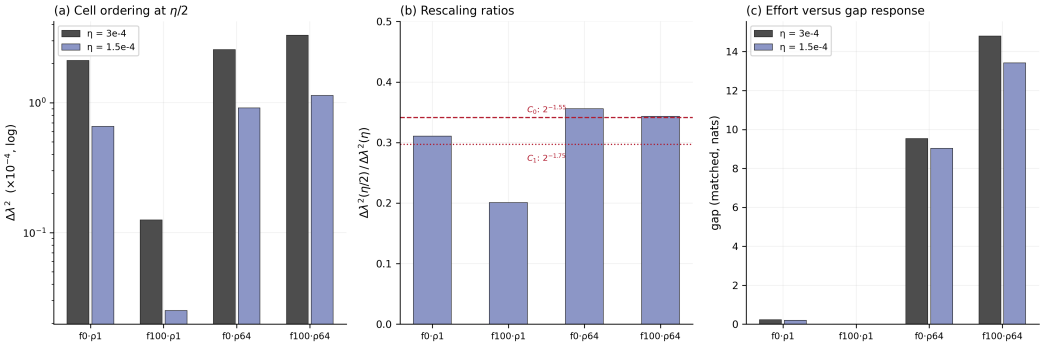


Figure 10: **Learning-rate control. (a)** Cell ordering in $\Delta\lambda^2$ at η and $\eta/2$. **(b)** Observed rescaling ratios against the learning-rate scalings of the intercept $C_0 \propto \eta^{1.55}$ (dashed, $2^{-1.55} = 0.34$) and slope $C_1 \propto \eta^{1.75}$ (dotted, $2^{-1.75} = 0.30$) of the earlier data-to-scale relation (Ding, 2026c); the mean substantial-cell ratio is 0.337 ($f100 \cdot \rho1$ excluded: near-zero base). **(c)** $\Delta\lambda^2$ rescales by about $3\times$, while the matched gap changes by -5 to -13% in the substantial cells, relative to the three-seed mean at η.

B.7 Fixed-budget model-size screen (Section 4.7)

The size screen is specified in Appendix A.1 (Table 4). Table 14 gives the terminal readouts and Figure 11 the trajectories and budget composition.

Within each active arm, t_{90} changes by less than 400 steps across sizes, and the matched $\eta/2$ control preserves the 160M/70M response ratio. Although the per-parameter accumulated alignment work and the $\Delta\lambda^2$ plateaus decrease with size, radial alignment remains the dominant interval-budget component (Figure 11d): its share falls from about 0.94 after warm-up to 0.70–0.83 at the endpoint in every active cell and size, with the repeated cells slightly below the clean cell after step 2,000.

Table 14: Terminal readouts of the fixed-budget size screen. $\Delta\lambda^2$ in units of 10^{-4}; losses and gap (= held − train, matched) in nats; t_{90} in steps. Dashes mark the near-zero shuffled-unrepeated arm, for which t_{90} is undefined. One run per cell and size.

Size	Cell	$\Delta\lambda^2$	Train	Held-out	Gap	t_{90}
70M	f0·ρ1	+2.12	3.611	3.845	+0.23	4545
70M	f0·ρ64	+2.56	0.084	9.634	+9.55	3661
70M	f100·ρ1	+0.13	7.464	7.425	-0.04	–
70M	f100·ρ64	+3.30	0.222	14.876	+14.65	3581
160M	f0·ρ1	+1.84	3.358	3.652	+0.29	4745
160M	f0·ρ64	+1.95	0.056	8.823	+8.77	3890
160M	f100·ρ1	+0.01	7.462	7.426	-0.04	–
160M	f100·ρ64	+2.54	0.124	13.571	+13.45	3521
410M	f0·ρ1	+1.60	3.200	3.547	+0.35	4917
410M	f0·ρ64	+1.50	0.047	8.363	+8.32	3959
410M	f100·ρ1	-0.05	7.463	7.425	-0.04	–
410M	f100·ρ64	+2.09	0.093	12.760	+12.67	3682

The alignment-dominated pattern persists across the 70M–410M screen, while its magnitude and its mapping to $\Delta\lambda^2$ change with model size.

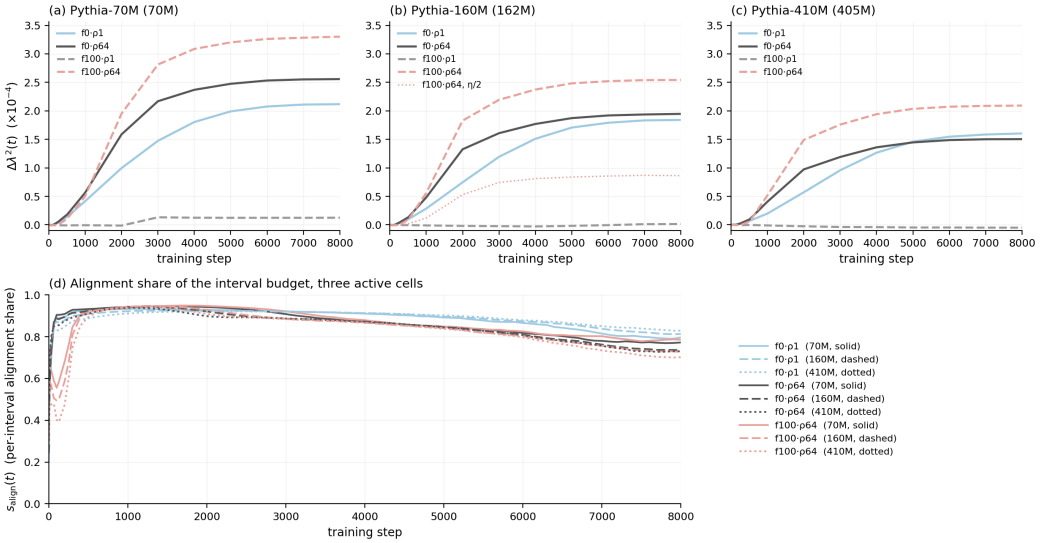


Figure 11: **Scale-response trajectories and budget composition in the size screen. (a–c)** $\Delta\lambda^2(t)$ for the four corner cells at 70M, 160M and 410M; the dotted curve in (b) is the matched $\eta/2$ control. The active plateaus decrease with size without an earlier t_{90}. **(d)** Per-interval alignment share s_{align} (edge-aware five-interval mean) for the three active cells; colours denote data conditions and line styles model sizes. The near-zero shuffled×1 arm is omitted because its share is denominator-sensitive. One run per curve.