EQUAL DEPTH EXPOSURE, DIFFERENT MODELS: ORDER EFFECTS IN RECURRENT-DEPTH TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recurrent-depth language models apply one shared block a variable number of times, and their training schedules are usually described only by how often each depth is used. We ask whether the *order* of depths changes the trained model when these counts are held exactly fixed. We train 67M-parameter models on permutations of one fixed multiset of 20,500 per-update depths, with identical counts for each depth across runs. Paired runs share initialization, data order, and learning rates; we evaluate each model at depths 1 to 16. Sorting depths in descending rather than shuffled order raises held-out loss at the deepest trained depth in all five seed pairs, by 0.25 nats on average (95% interval 0.20 to 0.31), and by about 0.6 nats three iterations beyond the trained band. The penalty retains its sign on larger and disjoint evaluation windows, a second multiset, a 102M-parameter model, a C4 corpus slice, and training without either auxiliary loss (three seed pairs). Ascending order, which resembles a depth curriculum, yields losses close to those of shuffled order on depths 5 to 13 but, after training on depths 2 to 10, raises loss by 0.29 nats six iterations beyond the trained band. The penalty depends on when the deepest updates arrive: moving the 5,000 deepest updates from the start to the end of the descending schedule leaves 2 to 6% of the penalty, and 300 to 2,050 further updates at depth 13 bring a checkpoint with 2 nats of excess loss to within 0.03 to 0.11 nats of the shuffled model. Depth order is therefore a training variable in its own right: schedules should be reported by when each depth is used, not only how often.

1 INTRODUCTION

Recurrent-depth, or looped, language models apply one shared block a variable number of times, so the same parameters can be run with more or less computation at inference time (Dehghani et al., 2018; Giannou et al., 2023; Geiping et al., 2025). We call the number of iterations the *depth* T. A model trained on a band of depths is evaluated at several *readout depths* t, where a shared output head decodes the state after t iterations, and its quality matters at every depth that may be used (Graves, 2016; Banino et al., 2021; Raposo et al., 2024). Training therefore requires a schedule that assigns a depth to every update. Such schedules are usually summarized by a mean depth or an *exposure histogram*, the number of updates that use each depth, which records how often each depth is used but not *when*. This leaves an open question: if the update count for each depth is fixed across schedules, can their order still change the final model?

Existing work leaves this question open. Curricula and stochastic optimization show that the order of training examples matters (Bengio et al., 2009; Hacohen & Weinshall, 2019; Shumailov et al., 2021), and depth curricula for recurrent models grow the number of iterations during training (McLeish et al., 2025). These curricula change the exposure histogram and the total computation together with the order, so they measure a combined effect. Order could also matter on its own, because the training rule is not symmetric under reordering: the optimizer carries state across updates, each gradient depends on the depth assigned to its step, and the learning rate changes along the step index.

We isolate depth order with a matched-exposure design (Section 2; Figure 1a,b). We fix one *multiset* of 20,500 per-update depths, a list of depths with their multiplicities, and train on permutations of it: a random shuffle as the reference, an ascending sort that resembles a depth curriculum, its descending reversal, and two intermediate arrangements. Each permuted run is paired with a shuffled

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.

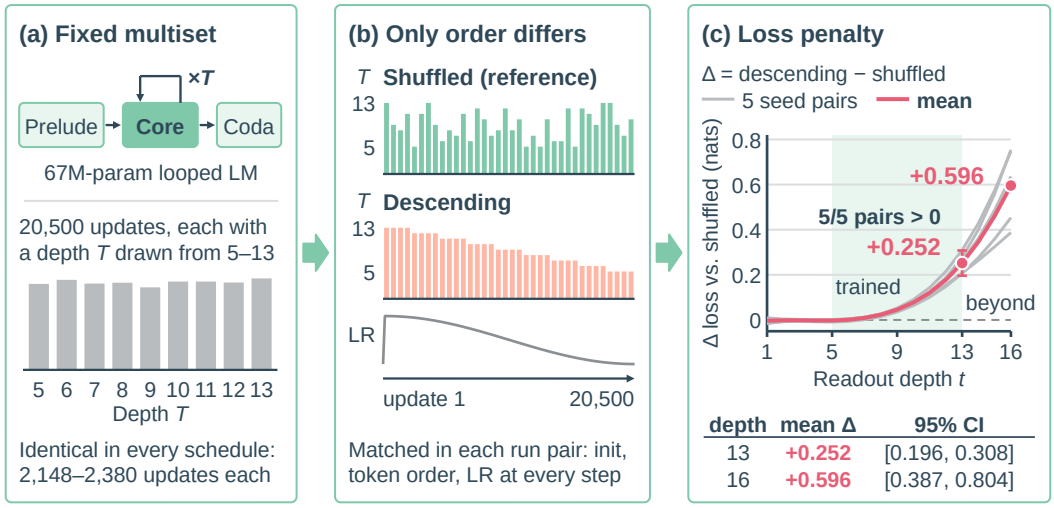


Figure 1: **Same depth exposure, different models.** (a) A 67M-parameter looped language model (LM) applies one shared core block T times per update. Each of the 20,500 training updates is assigned a depth $T \in \{5, \ldots, 13\}$, and the count for each depth (2,148–2,380) is identical across schedules. (b) Schedules differ only in the temporal arrangement of this multiset, sketched with 36 bars per trace (bar height T): the shuffled reference versus a globally descending sort. Within each run pair, initialization, token order, and the learning rate (LR) at every update are matched (warmup plus cosine decay; the curve is schematic); interactions of depth with data phase, learning rate, and optimizer state are part of the treatment. (c) Descending-minus-shuffled held-out loss at each readout depth t for five seed pairs (grey) and their mean (raspberry) on the 160-sequence validation window. Shading marks the trained depths 5–13; readouts 14–16 extrapolate. The penalty is near zero at shallow readouts, positive in all five pairs at the deepest trained depth ($+0.252$ nats; bar: 95% Student confidence interval (CI), $[0.196, 0.308]$), and reaches $+0.596$ nats three iterations beyond the trained band; the inset table lists both means with their intervals.

run that shares its initialization, data order, and learning rate at every step, so the permutation of depth assignments is the only planned difference. Equal depth counts also fix the total number of shared-block applications. The permutation determines which depths coincide with which data, learning rates, and optimizer states; these interactions form part of the treatment. Every arrangement is fixed before its run begins.

Depth order changes the final model, and the effect is concentrated at deep readouts, where additional recurrence is meant to help (Figure 1c). A globally descending schedule raises held-out loss (cross-entropy, in nats) at the deepest trained depth in every seed pair, by 0.25 nats on average, and this penalty grows when the model is run beyond its trained band. The contrast has the same sign across evaluation windows, depth multisets, model widths, corpora, and training objectives, and every paired run in three five-pair replication groups is positive. The cost of ascending order depends on the training band and the readout depth: it stays close to shuffled order in the primary band of depths 5–13 and raises extrapolated loss after training on depths 2–10, by 0.29 nats at depth 16. Interventions that move or continue parts of the schedule show that the penalty depends on when the deepest updates arrive: moving the 5,000 deepest updates to the end of training leaves 2–6% of it, and a few hundred late depth-13 updates bring a checkpoint with 2 nats of excess loss to within 0.11 nats of shuffled. Depth schedules should therefore be reported and compared by their temporal arrangement as well as by their exposure histograms.

- **A matched-exposure design.** We permute one fixed multiset of 20,500 depth assignments, so depth counts and total shared-block applications are identical across schedules, and match initialization, token order, and the per-step learning rate within each run pair (Section 2).
- **A replicated order effect.** Descending instead of shuffled order raises held-out loss at depth 13 by 0.252 nats (95% interval [0.196, 0.308]; five of five seed pairs positive) and at depth 16 by 0.596 nats, with the same sign on larger and disjoint evaluation windows, a second multiset, a

102M-parameter model, a C4 slice (Raffel et al., 2019), and training without the objective's two auxiliary losses (Sections 3 and 4).

- **A band-dependent cost of order.** Ascending order is near-neutral in the primary band and costs 0.29 nats at depth 16 after training on depths 2–10 (one paired run per band), so the cost of ascending order depends on the band and the readout depth (Section 4).
- **Probes that locate the effect in training history.** Moving the 5,000 deepest updates of the descending schedule changes the final penalty from 0.02 to 3.98 times the full descending penalty, depending on where they are placed (four seed pairs). A short continuation with 300–2,050 depth-13 updates brings an update-17,500 checkpoint with 2.0 nats of excess loss to within 0.028–0.105 nats of shuffled (seed 3). At update 17,500, the descending run's worst-case readout loss already exceeds the final shuffled value by 0.553 nats, 73% of its final penalty (seed 3; Section 5).

2 Matched-exposure experiment

2.1 What is held fixed and what changes

Let $M = (T_1, \ldots, T_S)$ contain $S = 20{,}500$ recurrent-depth assignments. A schedule is a permutation M_π, so the exposure count

$$n_d = \sum_{s=1}^{S} \mathbf{1}[T_{\pi(s)} = d] \quad \text{and total core calls} \quad N_{\text{core}} = \sum_{d=5}^{13} d\, n_d = 184{,}995 \tag{1}$$

are identical across schedules. Each call applies the shared block once to one training batch. We generate the primary multiset with NumPy `default_rng(20260904).integers(5,14, 20500)`. Its mean depth is 9.02415; Appendix B gives exact counts and schedule construction.

Five arrangements share the entire training protocol and differ only in temporal order: the random draw order (*shuffled*), globally *ascending* and *descending* sorts, *block ascending* sorts within ten consecutive 2,050-update blocks, and *deep first*, which places the 9,254 updates at depths 10–13 before the 11,246 updates at depths 5–9 while shuffling within each part. The primary replicated contrast is descending minus shuffled, trained at seeds 1–5. Each pair shares initialization and training-data order. The multiset and the shuffled reference sequence are fixed across this seed group, so changing the seed measures execution variability; Section 4 varies both.

2.2 Model, objective, and budget

The primary model has 67,409,889 parameters: width 768, twelve attention heads, two prelude blocks, one shared recurrent block, and one coda block. The vocabulary projection is tied to the token embedding. If x_0 is the prelude state and $x_t = F_\theta(x_{t-1})$ applies the shared block, the same coda and output head produce logits $h(x_t)$ at each readout depth. The wider model has 102,458,081 parameters, width 1024, and sixteen heads; Appendix J itemizes both counts.

Training uses WikiText-103 (Merity et al., 2016) with GPT-2 tokenization (Radford et al., 2019), context length 512, batch size 32, and 20,500 optimizer updates, totaling 335,872,000 input-token positions. AdamW (Loshchilov & Hutter, 2017) uses weight decay 0.05, peak learning rate 7×10^{-4}, 150 warmup updates, and cosine decay. The configured maximum depth is 24; the primary sampled training band is 5–13. All objective coefficients are shared across arms:

$$\mathcal{L}_T = \mathrm{CE}(h(x_T), y) + 0.1 \sum_{t \in \{\lfloor T/4 \rfloor, \lfloor T/2 \rfloor\}} \mathrm{CE}(h(x_t), y) + \frac{1}{T}\sum_{t=1}^{T} \rho_{0.9}(d_t - c_t), \tag{2}$$

where CE is next-token cross-entropy, $\rho_q(u) = \max(qu, (q-1)u)$ is the quantile (pinball) loss, and c_t predicts the normalized state displacement d_t to the final depth. The displacement target is

$$d_t = \|\mathrm{LN}(\mathrm{sg}(x_t)) - \mathrm{LN}(\mathrm{sg}(x_T))\|_2 / \sqrt{768},$$

with layer normalization LN and stop-gradient sg, while the predictor input features receive gradients. The primary model is thus trained with two auxiliary terms; Appendix C specifies the predictor, and a three-seed plain-cross-entropy ablation removes both terms (Section 4).

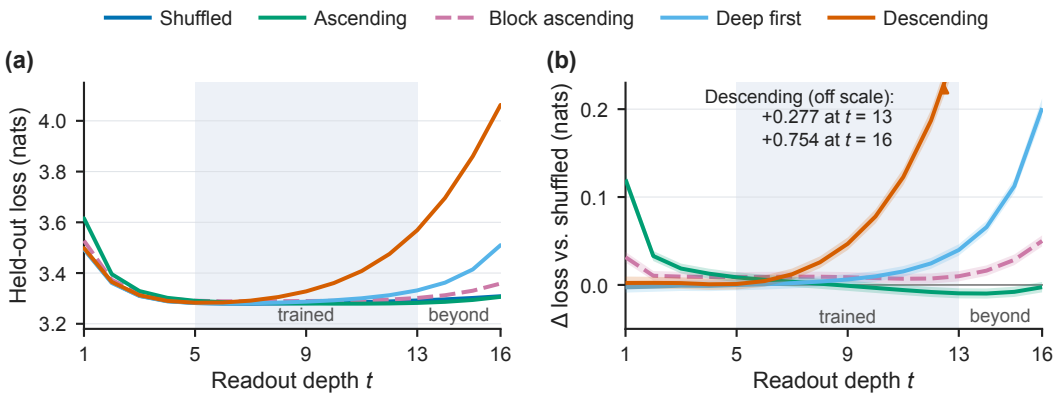


Figure 2: **Final loss for five arrangements of one depth multiset (primary seed 3).** (a) Held-out cross-entropy at each readout depth t on the 160-sequence validation window. (b) Each arrangement minus shuffled (shuffled $= 0$, grey line), paired over the same 160 sequences; shaded bands are ± 1.96 standard errors (SE) of the paired difference. Descending leaves the zoomed range after $t = 12$; its values at $t = 13$ and 16 are printed in the panel. Shading marks trained depths 5–13; readouts 14–16 extrapolate. Descending order raises loss at deep readouts, ascending stays close to shuffled, and block ascending and deep first lie in between. Exact values are in Table 2; the five-seed descending-minus-shuffled curves are in Figure 1c.

Besides fixing total shared-block applications and the objective's depth exposures, equal depth counts also equalize modeled run time: measured median step times on one L20Z GPU grow linearly from 207.09 ms at depth 5 to 254.17 ms at depth 13, so every trace is predicted to take about 4,728 seconds (Appendix I gives the fit and a resampling check).

2.3 Readouts and uncertainty

For final checkpoint (s, a), let $C_{s,a}(t)$ be mean held-out next-token cross-entropy at readout depth $t \in \{1, \ldots, 16\}$. The primary window contains 160 common validation sequences (deterministic-loader batches 10–14), each with 511 scored positions. We report the paired differences

$$\Delta_s(t) = C_{s,\mathrm{desc}}(t) - C_{s,\mathrm{shuffle}}(t) \tag{3}$$

at $t = 13$, the deepest trained depth, and $t = 16$, three iterations beyond the trained band; depth 4 lies below it.

Means and two-sided 95% Student-t intervals, $\bar{\Delta} \pm t_{0.975,4}\, s_\Delta/\sqrt{5}$ with four degrees of freedom and $t_{0.975,4} \approx 2.77645$, use the five training-seed differences and therefore capture variability across training runs; all reductions use the stored per-sequence losses in double precision. Sequence-level intervals at a fixed checkpoint quantify variability across sequences.

We also report two worst-case summaries: W, the maximum of $C(t)$ over $t \in \{4, 8, 12, 16\}$, and W_{strict}, the maximum over the two trained depths $t \in \{8, 12\}$, both taken *after* averaging sequence losses.

3 Depth order changes held-out loss

Permuting a fixed depth multiset changes final held-out loss: descending rather than shuffled order raises it in all five primary seed pairs.

Five-seed effect at trained depths. At depth 13, descending order increases cross-entropy by 0.2519 nats on average (Figure 1c), with standard deviation (SD) over seeds 0.0450 and 95% interval [0.1960, 0.3078]. Every paired difference is positive, ranging from 0.2059 to 0.3092 (Table 1). The restricted summary W_{strict} increases by 0.1776 [0.1443, 0.2108], so the finding holds on trained depths alone, without below-band depth 4 or extrapolated depth 16. The W contrast equals the depth-16 contrast for these paired checkpoints because their maxima lie at depth 16.

Table 1: **Individual paired seed effects** on the primary 160-sequence window. Each row is a separately trained seed pair; W_{strict} uses depths 8 and 12 only, and the interval is Student t_4.

	Descending − shuffled (nats)		
Seed	$\Delta C(13)$	ΔW_{strict}	$\Delta C(16)$
1	0.2072	0.1561	0.3875
2	0.2059	0.1477	0.4532
3	0.2771	0.1870	0.7540
4	0.2601	0.1820	0.6364
5	0.3092	0.2151	0.7466
Mean	0.2519	0.1776	0.5955
SD	0.0450	0.0268	0.1682
95% interval	[0.1960, 0.3078]	[0.1443, 0.2108]	[0.3867, 0.8044]

Table 2: **Primary seed 3: held-out cross-entropy (nats) for all five arrangements** on the 160-sequence validation window, at trained readout depths and one extrapolated depth. Bold marks the lowest loss in each column.

	Trained readout depth t			Extrapolated
Schedule	$t = 8$	$t = 12$	$t = 13$	$t = 16$
Shuffled	**3.2797**	3.2882	3.2918	3.3087
Ascending	3.2811	**3.2801**	**3.2823**	**3.3061**
Descending	3.3055	3.4752	3.5690	4.0627
Block ascending	3.2891	3.2955	3.3017	3.3587
Deep first	3.2835	3.3127	3.3316	3.5099

Sensitivity to individual pairs. Removing each pair in turn gives primary means in $[0.2376, 0.2634]$ nats at depth 13 and $[0.5559, 0.6476]$ at depth 16. With five of five positive pairs, an exact two-sided sign test gives $p = 0.0625$, the smallest value attainable with five pairs.

Growth with readout depth. The penalty grows with readout depth (Figure 1c; per-readout values in Appendix Table 6). At depth 16 its five-seed mean is 0.5955 [0.3867, 0.8044], with a larger SD of 0.1682. In seed 3, the contrast grows from approximately 0.001 at depth 5 to 0.187 at depth 12, 0.277 at depth 13, and 0.754 at depth 16, with paired window-level ± 1.96 SE between 0.006 and 0.039. The corresponding depth-16 perplexities are 27.35 for shuffled and 58.13 for descending order on this evaluation window. Extrapolation therefore amplifies an effect that is already present inside the trained band, and the penalty is concentrated at deep readouts.

Other orderings. Ascending and shuffled losses are close in the primary seed, while block ascending and deep first have intermediate penalties at deeper readouts (Figure 2 and Table 2). The five arrangements of one multiset thus span a graded range of endpoint losses, and the five-seed descending–shuffled contrast above provides the replicated uncertainty estimate.

Evaluation windows. The effect replicates on expanded validation and disjoint test windows of 480 sequences per split (batches 0–14; Table 3), with five-seed depth-13 means of 0.2452 on validation and 0.2526 on test. The contrast is positive for every seed on every window: across the seven validation and test windows of Appendix D, the 35 seed-level values of $\Delta C(13)$ range from 0.2021 to 0.3215, and those of ΔW_{strict} from 0.1477 to 0.2240. Test batches 10–14, evaluated for seeds 1, 2, 4, and 5, give $\Delta C(13)$ from 0.2151 to 0.3441.

4 Replication across settings and orderings

The positive contrast holds across changes in multiset, width, corpus, and objective, and the cost of ascending order depends on the training band and readout depth.

Multiset, width, corpus, and objective. A second depth multiset, the 102M model, and the C4 corpus experiment (Raffel et al., 2019), whose training stream contains 170.6M GPT-2-tokenized

Table 3: **Replications of the depth-13 contrast across windows, multisets, width, corpus, and objective** (descending minus shuffled). First group: Student t_4 intervals over the same five training-seed pairs, evaluated on different windows. Middle group: separate single-pair experiments with paired sequence-level intervals. Last row: range over the three plain-objective pairs; their individual values are in Section 4.

Condition	Evaluation window	Pairs	Interval over	$\Delta C(13)$ (nats)	95% interval
Primary (67M)	Validation 10–14	5	Training seeds	0.2519	[0.1960, 0.3078]
Expanded validation	Validation 0–14	5	Training seeds	0.2452	[0.1927, 0.2976]
Disjoint test window	Test 0–14	5	Training seeds	0.2526	[0.1966, 0.3086]
Second multiset	Validation 10–14	1	Sequences	0.2873	[0.2646, 0.3099]
102M model	Validation 10–14	1	Sequences	0.2677	[0.2537, 0.2818]
C4 corpus slice	C4 validation	1	Sequences	0.4638	[0.4211, 0.5065]
Plain cross-entropy	Validation 10–14	3	—	0.4950–0.5384	range of 3 pairs

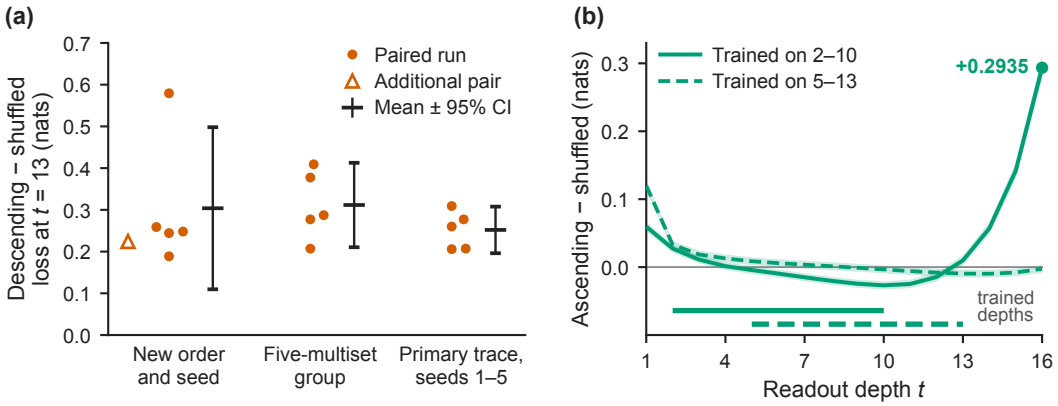


Figure 3: **Additional replications and band-dependent ascending order.** (a) Depth-13 descending–shuffled differences for three five-pair groups; dots are individual paired runs, and dark bars show each group's mean with a Student t_4 95% confidence interval (CI). The first group changes both permutation and initialization. The five-multiset group includes two pairs reported in Table 3 (primary seed 3 and the second multiset). The open triangle is one additional pair (0.2242), shown separately from the five-pair mean. (b) Ascending-minus-shuffled loss under two training bands (one paired execution each). Shaded bands are paired per-sequence ± 1.96 SE over the 160 validation sequences; bars at the bottom mark each run's trained depths; the annotated value is the depth-16 difference for the 2–10 band. Ascending stays close to shuffled under the primary 5–13 band and has higher loss at deep extrapolated readouts after training on 2–10.

tokens, each preserve the positive descending–shuffled contrast at depth 13 in a paired run (Table 3). Plain final-depth cross-entropy, removing both intermediate supervision and the predictor loss, also preserves the effect: depth-13 differences are 0.5007, 0.4950, and 0.5384 across three seed pairs, and the corresponding W differences are 0.6284, 0.6286, and 0.6868 (Appendix E). Order sensitivity therefore arises with plain cross-entropy alone; removing the auxiliary terms changes the size and shape of the loss difference.

Fresh permutations and multisets. On the primary multiset, five additional uniform permutations give positive paired contrasts from 0.189 to 0.579 (mean 0.3038; Figure 3a). Each pair shares its initialization, while initialization changes across pairs (seeds 100–104). Their sample variance, 0.0245, exceeds the primary seed group's 0.0020; one large contrast drives much of the difference, and leaving out each pair in turn gives means from 0.2350 to 0.3326. A sixth pair (initialization seed 105) gives 0.2242. A five-multiset group, formed by three new multiset draws together with the primary seed-3 pair and the second-multiset pair, also gives positive contrasts (mean 0.3116, SD 0.0815). Every pair in the three groups is positive; Table 4 summarizes them and Appendix E lists individual values.

Table 4: **Depth-13 differences for three five-pair groups; every pair is positive.** Descending minus shuffled. Intervals use Student t_4 within each observed group. The five-multiset group includes the primary seed-3 pair and the second-multiset pair.

		$\Delta C(13)$ (nats)		
Group	Pairs positive	Mean	SD	95% interval (t_4)
New order and seed	5 of 5	0.3038	0.1564	[0.1096, 0.4981]
Five-multiset group	5 of 5	0.3116	0.0815	[0.2104, 0.4128]
Primary trace, seeds 1–5	5 of 5	0.2519	0.0450	[0.1960, 0.3078]

The cost of ascending order depends on the training band. After shifting the training band to depths 2–10, the in-band worst loss over $\{4, 8\}$ is 3.2996 for shuffled, 3.2939 for ascending, and 3.5649 for descending order. At depth 16, six iterations beyond that band, these become 3.3849, 3.6784, and 7.3820. Ascending therefore incurs a 0.2935-nat extrapolation penalty in this setting (Figure 3b; one paired run per band). The cost of ascending order thus depends on the band and on the readout depths at which the model is evaluated.

A data-order stress test. A supplementary experiment permutes the *data* order, not the depth, of continued Pythia-410M training (Biderman et al., 2023). In a heavy-reuse stress test (160 passes over a 4,096-chunk pool; on the first pool every arm ends at 10.5–11.1 nats, from 3.16 before training), sorting each pass by initial loss raises held-out loss relative to shuffled passes by 0.26–0.52 nats in both disjoint training multisets (Appendix F).

5 Training history shapes the order effect

The order effect is carried by training history: moving the deepest updates, or continuing training with a short block of them, changes the endpoint loss, and in a single-seed probe the penalty persists through an optimizer-state reset. Even with the learning rate fixed at every step, order changes the optimizer trajectory, because each step's gradient depends on its depth and on the batch processed at that depth. The interventions below probe this history dependence; each states its replication and budget.

Why equal counts can give different endpoints. A linear surrogate makes the role of order explicit. For a depth-conditioned quadratic loss with shared curvature H and depth-specific forcing b_T, gradient descent with constant rate η and $A = I - \eta H$ reaches

$$\theta_S = A^S \theta_0 + \eta \sum_{s=0}^{S-1} A^{S-1-s}\, b_{T_s}. \tag{4}$$

Each depth's forcing is weighted by A^{S-1-s}, which depends on *when* that depth is used, so identical depth counts can produce different weighted sums. Exchanging two adjacent depths a and b that are followed by m further updates changes the endpoint by $\eta^2 A^m H(b_b - b_a)$, order ab minus order ba (Appendix H). Even this shared-curvature model therefore retains the arrangement of a fixed multiset; the probes below examine the trained models directly.

A simple candidate account. The most direct explanation follows from the schedule itself. Under Eq. (2), cross-entropy at a readout depth $t \geq 7$ is supervised only by updates with $T = t$. In the descending trace, all 9,254 updates with $T \geq 10$ precede every shallower update, so the final 11,246 updates, including the entire second half of the cosine decay, supervise only depths 5–9 while they continue to change the shared block. Deep readouts could therefore drift after their last direct supervision, which would also fit the growth of the penalty with readout depth. The probes below test parts of this recency account: late deep updates repair the penalty, and earlier arrival history shapes the endpoint.

Placement interacts with training history. Each placement run starts from the descending schedule and exchanges its first 5,000 updates, which hold the deepest depths (11–13), with a target window, so the multiset is unchanged (Table 5; four seed pairs per site). Effects are normalized by the

Table 5: **Placement probes: the position of the deepest block sets the endpoint penalty.** Each run starts from the descending schedule and exchanges its first 5,000 updates, the deepest depths, with the listed update range, so the multiset is unchanged. Penalty$/D_s$ is the endpoint W penalty relative to the same-seed shuffled run, divided by that seed's full descending–shuffled W difference D_s. Multi-seed rows show ranges across seeds; single-seed rows are individual interventions. LR: learning rate. Appendix Table 10 adds a control, post-window continuations, paired SEs, and exact seeds.

Placement of the deepest block	Update range	Seeds	Penalty$/D_s$
Early placement	5,000–10,000	4	1.17–3.98
Late-middle placement	10,250–15,250	4	0.83–0.94
Terminal placement	15,500–20,500	4	0.02–0.06
Early, in-window LR $= 0$	5,000–10,000	1	1.81
Early, LR $= 0$ and optimizer reset	5,000–10,000	1	1.24

same seed's full descending–shuffled W difference, D_s. Early placements yield 1.17–3.98 times this reference penalty, late-middle placements 0.83–0.94, and terminal placements 0.02–0.06. The same 5,000 updates thus give penalties from 0.02 to 3.98 times D_s by position; moving them to the end removes 94–98% of it.

Setting the learning rate to zero inside the early window still leaves 1.81 times D_s in one run, or 1.24 with an optimizer-state reset. The penalty therefore arises even without parameter updates *inside* that window. A control that keeps the descending run's first 5,000 updates and shuffles the other 15,500 gives 0.067; 5,000 further random-depth updates after a terminal placement leave 0.014–0.033.

Late deep updates repair the penalty; earlier history sets most of it. A short block of deep updates repairs deep readouts (Appendix Table 11; seed 3). The parent run sorts the multiset in descending order but holds back 2,050 depth-13 updates for its end. From its update-17,500 checkpoint ($W = 5.307$), where 950 updates at depth 5 and then the 2,050 at depth 13 remain, we restart training for 500–3,000 updates with a fresh optimizer and warmup-plus-cosine schedule. W is 5.057 after the first 500 updates, all at depth 5, and falls to 3.414, 3.355, and 3.337 once 300, 1,100, and 2,050 depth-13 updates have been applied (shuffled reference 3.309; the uninterrupted parent ends at 3.348). In the descending run itself, W at update 17,500 is already 3.862, which exceeds the final shuffled W by 0.553 nats, 0.733 of the final penalty; most of the penalty is thus in place before the last 3,000 updates.

Sorted orders leave a detectable signature in the model. A ridge probe on each validation sequence's mean-pooled, layer-normalized state (two folds of 80 of the 160 sequences; seed 3) identifies which final checkpoint produced it with accuracy 1.0 at every readout depth, for descending versus shuffled and for ascending versus shuffled, whose losses are close (95th percentile of a label-permutation null: below 0.58). Both sorted orders thus yield measurably different models from the same depth multiset. Mean predictive entropy at readout depth 16 is 5.080 for descending and 3.507 for shuffled; ascending (3.454) and the terminal-placement run (3.504) differ from shuffled by 0.05 and 0.003 nats (Appendix G).

6 Related work

Ordering examples and tasks. Curriculum learning changes example order or sampling difficulty (Bengio et al., 2009; Hacohen & Weinshall, 2019; Platanios et al., 2019); ordering attacks show that permutations alone can disrupt learning (Shumailov et al., 2021). Random-reshuffling theory studies order under structured stochastic optimization assumptions (Gürbüzbalaban et al., 2015). Sequential interference, rehearsal, and consolidation provide related accounts of training history (McCloskey & Cohen, 1989; Robins, 1995; Kirkpatrick et al., 2016; Parisi et al., 2018), while task-order analyses optimize sequences of distinct tasks (Bell & Lawrence, 2022; Li & Hiratani, 2025). Our primary treatment changes internal recurrent depth on a fixed token stream.

Varying recurrent computation. Universal Transformers and parameter sharing decouple unique parameters from computation depth (Dehghani et al., 2018; Lan et al., 2019). Looped models study

repeated computation and learned algorithms (Giannou et al., 2023; Geiping et al., 2025; Zhu et al., 2025); adaptive computation time (ACT), PonderNet, and routing methods instead learn or select input-dependent budgets (Graves, 2016; Banino et al., 2021; Raposo et al., 2024; Bae et al., 2025). Stochastic depth and layer dropping vary active computation during training (Huang et al., 2016; Fan et al., 2019). LoopFormer aligns supported trajectories through time/step conditioning and shortcut consistency (Jeddi et al., 2026). The closest recurrence curriculum (McLeish et al., 2025) raises mean depth from 1 to 32 and compares it with constant mean depth 32. That comparison changes depth exposure and total computation; our exact-multiset control holds both fixed and changes only the order.

Learning rates and optimizer history. Adam and AdamW carry state across updates (Kingma & Ba, 2014; Loshchilov & Hutter, 2017); learning-rate schedules and batch size alter the optimization path and its stochastic dynamics (Loshchilov & Hutter, 2016; Smith et al., 2017; Mandt et al., 2017).

7 Conclusion

With depth exposure held exactly fixed, temporal depth order changes a recurrent language model's final quality. Descending rather than shuffled order raises held-out loss in all five primary seed pairs, already within the trained band, and extrapolation magnifies the penalty. The effect has the same sign on larger and disjoint evaluation windows, a second multiset, a wider model, a C4 corpus slice, and plain cross-entropy, and the cost of ascending order depends on the band and readout depth. Placement and continuation probes show that the penalty depends on when the deepest updates arrive and that late deep updates largely repair it. We recommend reporting both the exposure histogram and the temporal arrangement of a depth schedule, and evaluating at the intended readout depths.

7.1 Limitations and future work

All models have at most 102M parameters and 512-token contexts. The second-multiset, 102M, C4, and shifted-band results are single paired runs, and the zero-learning-rate, continuation, and 41,000-update probes use one seed. Student intervals assume exchangeable, approximately normal seed effects; the exact sign test uses only their signs. Four directions extend this study; Appendix A discusses each limitation in detail. *Scale and baselines:* applying the matched-exposure design to larger models, longer contexts, more corpora, and downstream tasks, and comparing it with fixed-depth and standard curriculum recipes, would calibrate the practical size of order effects. *Replication:* multi-seed versions of the single-run settings and probes, and designs that vary permutation and initialization separately, would sharpen the secondary estimates. *Mechanism:* interventions that separate depth recency, learning-rate phase, and optimizer state, together with a predictive model of penalty accrual and repair, could identify the mechanism behind the history dependence. *Cost and practice:* measuring whole-run wall-clock time and deriving schedule-design rules for recurrent-depth training would make the findings actionable.

Ethics Statement

This work uses publicly available corpora, benchmarks, and model checkpoints. We did not conduct human-subject experiments. The work shares the risks common to language-model research, including bias and potentially harmful content in public corpora and model outputs.

REPRODUCIBILITY STATEMENT

The supplementary ZIP archive provides code, configurations, launch and analysis scripts, and recorded results, together with reproduction instructions. Experimental settings are specified in the main text and appendix.

AI USE STATEMENT

During the project, vibe coding tools (Codex and Claude Code) were used to help develop the code for the methods and experiments. All such work was conducted under human supervision and underwent human review. During manuscript preparation, AI tools were used to polish and revise the text and figures.

REFERENCES

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. *arXiv preprint arXiv:2507.10524*, 2025. URL https://arxiv.org/abs/2507.10524.

Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021. URL https://arxiv.org/abs/2107.05407.

Samuel J. Bell and Neil D. Lawrence. The effect of task ordering in continual learning. *arXiv preprint arXiv:2205.13323*, 2022. URL https://arxiv.org/abs/2205.13323.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009. doi: 10.1145/1553374.1553380.

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. *arXiv preprint arXiv:2304.01373*, 2023. URL https://arxiv.org/abs/2304.01373.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. *arXiv preprint arXiv:1807.03819*, 2018. URL https://arxiv.org/abs/1807.03819.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019. URL https://arxiv.org/abs/1909.11556.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025. URL https://arxiv.org/abs/2502.05171.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *arXiv preprint arXiv:2301.13196*, 2023. URL https://arxiv.org/abs/2301.13196.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016. URL https://arxiv.org/abs/1603.08983.

Mert Gürbüzbalaban, Asuman Ozdaglar, and Pablo Parrilo. Why random reshuffling beats stochastic gradient descent. *arXiv preprint arXiv:1510.08560*, 2015. URL https://arxiv.org/abs/1510.08560.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. *arXiv preprint arXiv:1904.03626*, 2019. URL https://arxiv.org/abs/1904.03626.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. *arXiv preprint arXiv:1603.09382*, 2016. URL https://arxiv.org/abs/1603.09382.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. LoopFormer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. *arXiv preprint arXiv:2602.11451*, 2026. URL https://arxiv.org/abs/2602.11451.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*, 2016. URL https://arxiv.org/abs/1612.00796.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. URL https://arxiv.org/abs/1909.11942.

Ziyan Li and Naoki Hiratani. Optimal task order for continual learning of multiple tasks. *arXiv preprint arXiv:2502.03350*, 2025. URL https://arxiv.org/abs/2502.03350.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. URL https://arxiv.org/abs/1608.03983.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. URL https://arxiv.org/abs/1711.05101.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate Bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017. URL https://arxiv.org/abs/1704.04289.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989. doi: 10.1016/S0079-7421(08)60536-8.

Sean McLeish, Ang Li, John Kirchenbauer, Dayal Singh Kalra, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Jonas Geiping, Tom Goldstein, and Micah Goldblum. Teaching pretrained language models to think deeper with retrofitted recurrence. *arXiv preprint arXiv:2511.07384*, 2025. URL https://arxiv.org/abs/2511.07384.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *arXiv preprint arXiv:1802.07569*, 2018. URL https://arxiv.org/abs/1802.07569.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M. Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019. URL https://arxiv.org/abs/1903.09848.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

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. *arXiv preprint arXiv:1910.10683*, 2019. URL https://arxiv.org/abs/1910.10683.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024. URL https://arxiv.org/abs/2404.02258.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995. doi: 10.1080/09540099550039318.

Ilia Shumailov, Zakhar Shumaylov, Dmitry Kazhdan, Yiren Zhao, Nicolas Papernot, Murat A. Erdogdu, and Ross Anderson. Manipulating SGD with data ordering attacks. *arXiv preprint arXiv:2104.09667*, 2021. URL https://arxiv.org/abs/2104.09667.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017. URL https://arxiv.org/abs/1711.00489.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025. URL https://arxiv.org/abs/2510.25741.

A Extended discussion of limitations

Section 7.1 summarizes the directions below as future work; this appendix states the corresponding limitations of the present evidence in full.

Scale, baselines, and replication. The evidence covers a small recurrent family (67M and 102M parameters), 512-token contexts, two corpora, and fixed readout grids. We do not compare against a fixed-depth or standard-recipe baseline, so the size of the order penalty is not calibrated against a practitioner's alternatives. Only the descending–shuffled contrast is replicated over five seeds. The five-arm comparison (Table 2) uses one seed and is descriptive; the similar losses under ascending and shuffled order observed there establish neither equivalence nor a universally safe curriculum, and ascending order is not consistently preferable. The second-multiset, 102M, and C4 rows are single paired runs, the shifted-band result is one paired run per band, and the zero-learning-rate, optimizer-reset, continuation, and 41,000-update probes are single seed-3 runs. Because depth interacts with data phase, learning rate, and optimizer state, every estimate is conditional on the training recipe, and a fixed scalar learning rate does not make the effect optimizer-independent.

Statistics and evaluation. The Student interval assumes exchangeable, approximately normal seed effects, is conditional on the architecture, schedules, and evaluation sample, and is not an assumption-free sign guarantee. Leave-one-out and placement ranges are descriptive, not confidence intervals; single-pair intervals are sequence-level, not training-seed, intervals. Shared sequences and overlapping windows do not create independent training replications, and the 480-sequence windows are not whole-corpus evaluations. W and W_{strict} are correlated summaries of one loss matrix; W_{strict} covers two trained depths rather than the whole band, and the equality of the W and depth-16 contrasts need not hold for other schedules or bands. The fresh-permutation group varies order and initialization together, so its larger variance (0.0245 versus 0.0020) is not an identified variance decomposition, and summing variances would not estimate a "permutation share"; the five-multiset group reuses two pairs from Table 3 and is not independent of it. The Pythia-410M data-order experiment severely degrades the pretrained checkpoint in every continued-training arm, so it is only a stress test outside the depth-order claim.

Mechanism and compute. The probes do not identify a single mechanism, and recency of deep supervision is at most a partial account. The continuation probes restart from a checkpoint of a separate run, P8 (Appendix G), whose schedule is descending except for a final block of 2,050 depth-13 updates, and add updates with a fresh optimizer and learning-rate schedule, so they are not a free scheduling improvement. Early placements exceed the full descending penalty, which the probes do not explain, and placement also changes the arrival state, learning rate, and remaining training, so it isolates neither a position-only nor a learning-rate-only mechanism. The zero-learning-rate run does not show that zero-update computation damages weights: it also forgoes 5,000 comparatively high-learning-rate updates, which could by itself raise the endpoint loss, and surrounding allocations, optimizer evolution, and later learning still differ. The quadratic surrogate does not identify the nonlinear AdamW mechanism; a temporal kernel fit to checkpoints of the 20,500-update descending run (Appendix H) predicts $W = 5.983$ at 41,000 updates against a measured 4.064 (a 1.92-nat error); and probe separability does not locate the source of the loss penalty. No universal forgetting or repair law follows. Equal depth counts do not by themselves prove equal elapsed time: the 4,728-second figure is modeled from measured step costs, not compared across complete runs, and excludes order-dependent memory allocation, thermal state, and distributed effects.

Positioning. The Pythia data-order experiment belongs to the example-order line and does not establish transfer of the depth intervention. Our method is a controlled scheduling experiment within a fixed architecture and objective, not a new adaptive-depth architecture. Relative to the closest recurrence curriculum (McLeish et al., 2025), neither the architecture nor the auxiliary objective is a reproduction or competitive test of that training recipe.

B SCHEDULE CONSTRUCTION AND EVALUATION SCOPE

Depth counts for the primary multiset, from 5 through 13, are 2,239, 2,343, 2,247, 2,269, 2,148, 2,300, 2,299, 2,275, and 2,380. Sorting changes only temporal order. Deep-first places all depths at least 10 before the rest, independently shuffling both parts with seed 31337; the parts contain 9,254 and 11,246 updates. Block ascending sorts within ten consecutive blocks of 2,050 updates. Fresh-permutation construction uses `default_rng(20261000+i).permutation(20500)` and pair initialization seeds $100 + i$.

The reference evaluation uses deterministic loader batches 10–14 (160 sequences); expanded windows use batches 0–14 (480 sequences). At context 512, the stored WikiText-103 validation and test token streams contain 251,049 and 287,645 tokens, respectively. Thus the 480-sequence evaluations do not cover either split completely, particularly for test. The data loader returns next-token-shifted targets, and evaluation scores `[:, :-1]`, giving 511 predictions per sequence. Fixed-depth readouts use the same coda and vocabulary projection. All reductions are computed from the stored per-sequence losses; evaluation windows within a split overlap.

Table 6 lists the descending–shuffled contrast at every reported readout depth of primary seed 3, with paired window-level uncertainty; Section 3 summarizes it.

Table 6: **Per-readout profile of the contrast in primary seed 3** (descending minus shuffled, in nats; paired window-level estimates with ± 1.96 SE on the 160-sequence window). Below the band the contrast is numerically negligible; it grows through the trained depths, and extrapolation to depth 16 amplifies it.

Readout depth t	1	5	8	11	12	13	16
$\Delta C(t)$	0.002	0.001	0.026	0.123	0.187	0.277	0.754
± 1.96 SE	0.007	0.006	0.007	0.009	0.012	0.017	0.039

C AUXILIARY PREDICTOR AND LABEL CONVENTION

The auxiliary residual predictor projects a layer-normalized state to 64 dimensions and concatenates a 32-dimensional step embedding with five scalar features: normalized residuals from the current and preceding transitions, their clipped log ratio, normalized step index, and state norm. A 128-unit GELU multilayer perceptron and softplus output produce a nonnegative estimate. Stop-gradient is applied to the displacement target, not to every feature feeding this predictor. Its coefficients are identical across schedules. Its prediction accuracy is not evidence of a bound on the displacement of arbitrary future states.

The data loader supplies next-token-shifted targets. Evaluation scores logits and those shifted targets at positions `[:, :-1]`, giving 511 predictions per sequence. Fixed-depth logits use the same coda and vocabulary projection. The auxiliary predictor does not select which evaluations are included: every readout depth is evaluated, and all reported loss comparisons use measured checkpoint losses.

The displacement target in Eq. (2) is

$$d_t = \frac{\|\mathrm{LN}(\mathrm{sg}(x_t)) - \mathrm{LN}(\mathrm{sg}(x_T))\|_2}{\sqrt{768}}.$$

The target uses detached states; not all auxiliary predictor features are detached. The training code also implements an optional drift-anchor penalty; its weight is zero in every run, so Eq. (2) is the complete objective. In the plain ablation, intermediate supervision and the auxiliary predictor term are both disabled while the paired training recipe and depth traces are otherwise unchanged.

D EXPANDED EVALUATION WINDOWS

The first fifteen complete batches contain 480 sequences; five-batch subwindows contain 160. Expanded windows include their subwindows, so their estimates are not independent. Tables 7 and 8 report each training seed separately. W_{strict} is the maximum of the two mean losses at depths 8 and 12, not the mean of per-sequence maxima. Sequence-level standard errors quantify variation across sequences at a fixed checkpoint; the main text instead reports Student intervals over five training seeds.

Table 7: **Per-seed $\Delta C(13)$ on common evaluation windows.** Descending minus shuffled, in nats. Each column is one independently trained paired seed; windows are loader-batch ranges.

	$\Delta C(13)$ (nats), training seed				
Window	1	2	3	4	5
Validation 0–14	0.2043	0.2074	0.2593	0.2477	0.3071
Validation 0–4	0.2038	0.2048	0.2453	0.2319	0.3028
Validation 5–9	0.2021	0.2115	0.2555	0.2511	0.3095
Validation 10–14	0.2072	0.2059	0.2771	0.2601	0.3092
Test 0–14	0.2151	0.2091	0.2687	0.2499	0.3203
Test 0–4	0.2050	0.2074	0.2551	0.2267	0.2953
Test 5–9	0.2145	0.2048	0.2664	0.2455	0.3215

Table 8: **Per-seed $\Delta W_{\mathrm{strict}}$ on common evaluation windows.** Descending minus shuffled, in nats; W_{strict} is the larger of the mean losses at depths 8 and 12. Each column is one independently trained paired seed.

	$\Delta W_{\mathrm{strict}}$ (nats), training seed				
Window	1	2	3	4	5
Validation 0–14	0.1528	0.1502	0.1749	0.1747	0.2134
Validation 0–4	0.1519	0.1485	0.1673	0.1643	0.2106
Validation 5–9	0.1504	0.1545	0.1705	0.1778	0.2145
Validation 10–14	0.1561	0.1477	0.1870	0.1820	0.2151
Test 0–14	0.1630	0.1532	0.1823	0.1780	0.2240
Test 0–4	0.1556	0.1529	0.1733	0.1642	0.2072
Test 5–9	0.1621	0.1504	0.1792	0.1742	0.2240

A further disjoint-window evaluation covers test batches 10–14 for seeds 1, 2, 4, and 5: $\Delta C(13) = 0.2259, 0.2151, 0.2773, 0.3441$, respectively (four-seed mean 0.2656). This is a subset of the test-window evidence, not a new training replication.

E ADDITIONAL PERMUTATION, MULTISET, AND OBJECTIVE RESULTS

The five additional permutation pairs change the shuffled trace and the shared within-pair initialization together (initialization seeds 100–104). The observed variance therefore combines those changes and their interactions. The five-multiset group includes the primary and second multiset plus three new draws; the primary seed group changes initialization with the primary multiset and shuffled trace fixed. These groups are not a crossed factorial design and do not identify additive variance components. Table 4 in Section 4 summarizes the three groups.

The five new-order-and-seed differences are 0.2480, 0.2443, 0.2588, 0.1887, 0.5794. The five differences in the five-multiset group are 0.2771, 0.2873, 0.4091, 0.2070, 0.3775. One additional pair (initialization seed 105, separate permutation) gives 0.2242. It is shown separately in Figure 3, so the five-pair summaries are unchanged by it.

The plain-objective depth-13 paired differences at seeds 1, 2, and 3 are 0.5007, 0.4950, and 0.5384. The paired W differences, computed as maxima after averaging sequences, are 0.6284, 0.6286, and 0.6868, respectively. The main text emphasizes the depth-13 result to avoid conflating trained and extrapolated readouts.

F PYTHIA DATA-ORDER EXPERIMENT

This experiment changes data order in a fixed-depth pretrained model. It does not apply the recurrent-depth treatment. The Pythia-410M tokenizer converts consecutive raw C4 records into 12,288 chunks of 512 tokens: three disjoint 4,096-chunk pools supply training multiset M1, training multiset M2, and evaluation. The evaluation window is chunks 8,192–8,351 (160 chunks, all in the evaluation pool). Initial difficulty and final evaluation both score next-token cross-entropy over 511 positions, using FP32 log probabilities and the same Pythia tokenizer and chunk identifiers.

Each training pool contains 2,097,152 distinct token positions. Training consumes 656,000 chunk draws (20,500 updates, batch 32): 160 full cycles of the 4,096 chunks plus 640 further draws. The primary shuffled trace permutes each cycle; hard-first and easy-first sort each cycle by initial per-chunk loss, reversing the within-cycle order. They are *not* global sorts over all 20,500 updates. A second M1 shuffle globally permutes the same occurrence multiset, so it also changes the within-cycle repetition pattern. Segment-exchange arms swap the first 5,000 batches with batches 10,250–15,249 or 15,500–20,499 in the hard-first trace. All arms preserve the occurrence multiset, but their data sequence and batch composition differ.

All runs initialize from the same pretrained Pythia-410M checkpoint and share the absolute-step learning-rate schedule: peak 3×10^{-4}, 150 warmup updates, cosine decay over 20,500 updates; AdamW $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.1, clipping 1.0, and bf16 training. The paired sequence-level standard errors in Table 9 are conditional on one run per arm, not independent training replications.

Table 9: **Pythia-410M data-order results for training multiset M1** (nats, on the same disjoint 160-chunk window). CE is next-token cross-entropy; initial CE is the mean of the stored base-checkpoint difficulty scores on exactly those chunks. All continued-training arms severely degrade held-out quality relative to this initial checkpoint. Differences between trained arms remain measurable within that degraded regime. Paired SE is the sequence-level SE of the difference from shuffled; a dash marks an entry that does not apply or a row without a paired SE.

M1 condition	Evaluation CE (nats)	Difference from shuffled	Paired SE
Initial pretrained checkpoint	3.1599	—	—
Shuffled cycles	10.5943	*reference*	—
Second shuffle (global)	10.5278	-0.0665	—
Hard-first cycles	10.8550	$+0.2607$	0.0242
Easy-first cycles	11.0658	$+0.4715$	0.0229
Head exchanged to late-middle	11.0086	$+0.4143$	0.0219
Head exchanged to terminal	10.6398	$+0.0456$	0.0202

M2 uses the other disjoint 4,096-chunk training pool and yields hard-first-minus-shuffled 0.3713 (paired SE 0.0251) and easy-first-minus-shuffled 0.5158. Thus both sorted orders incur penalties in two multisets, with easy-first worse than hard-first. This is evidence about data order under heavy reuse of a small training pool. It does not establish effective continued pretraining, a robust modern-model depth effect, or a universal ranking of ascending and descending schedules. The terminal exchange's 0.0456 difference is smaller than the observed difference between the two shuffled controls, but that single contrast is not an equivalence test or a universal noise floor.

G PLACEMENT, CONTINUATION, AND CHECKPOINT PROBES

Let $D_s = W_{s,\mathrm{desc}} - W_{s,\mathrm{shuffle}}$. All normalized placement effects divide the intervention's difference from the same-seed shuffled reference by D_s. As in Section 2, W is a maximum across readout depths after averaging sequence losses. Ratios for seed 3 use $D_3 = 0.7540$; for example, the zero-learning-rate intervention gives $(4.6744 - 3.3087)/0.7540 = 1.8113$. These ratios compare observed endpoints; they are not parameter-free coefficients of a general model. Each placement run exchanges the first 5,000 updates of the descending schedule (depths 13, 12, and 11) with the listed window, so every placement arm is a permutation of the primary multiset. Each window row gives the update range occupied by the 5,000-update intervention window and the seeds at which it was run.

Table 10: **All placement and within-window dose interventions.** Penalty$/D_s$ is the cell's W penalty (difference from the same-seed shuffled reference) divided by the same seed's matched descending–shuffled penalty D_s. Placement rows exchange the first 5,000 updates of the descending schedule with the listed window. The head-kept control keeps those updates first and shuffles updates 5,000–20,500. Post-window rows follow a terminal placement with 5,000 further updates at depths drawn uniformly from 5–13, under a fresh warmup and cosine whose summed learning rate (dose) is 0.195 or 1.411. The paired SE is the sequence-level standard error on the shared 160-sequence window and reflects evaluation noise only, not training replication; for multi-seed rows it is the largest SE across seeds. A dash marks a row reported without an SE. LR: learning rate. Each run is an independent full training run.

Intervention	Window (updates)	Seeds	Penalty$/D_s$	Paired SE
Early placement	5,000–10,000	2, 3, 4, 5	1.1746–3.9844	≤ 0.0936
Late-middle placement	10,250–15,250	2, 3, 4, 5	0.8297–0.9434	≤ 0.0226
Terminal placement	15,500–20,500	1, 3, 4, 5	0.0233–0.0605	≤ 0.0081
Descending head kept, rest shuffled	5,000–20,500	3	0.0666	0.0046
Early, in-window LR $= 0$	5,000–10,000	3	1.8113	—
Early, LR $= 0$ and optimizer reset	5,000–10,000	3	1.2379	0.0417
Terminal, post-window dose 0.195	15,500–20,500	3	0.0332	0.0044
Terminal, post-window dose 1.411	15,500–20,500	3	0.0138	0.0044

Table 11 lists the seed-3 continuation and checkpoint probes summarized in Section 5. The continuations start from checkpoints of a parent run, P8, which sorts the primary multiset in descending order except for 2,050 depth-13 updates that it places last: 330 updates at depth 13, then depths 12 down to 5, then the 2,050 updates at depth 13. From the update-17,500 checkpoint, the parent's remaining schedule is 950 updates at depth 5 followed by 2,050 at depth 13. Each continuation loads only the parent weights, uses a fresh AdamW state and a fresh warmup-plus-cosine schedule (peak 7×10^{-4}, 150 warmup updates) over its own length, and runs the first 500, 1,250, 2,050, or 3,000 updates of that remaining schedule.

Table 11: **Continuation and checkpoint probes** (seed 3, W in nats on the primary 160-sequence window; $W = 3.309$ for shuffled, 4.063 for descending). Continuations restart from a checkpoint of the parent run P8, which holds back 2,050 depth-13 updates for its end, with a fresh optimizer and learning-rate schedule. For the continuations from update 17,500, W is within 0.11 nats of shuffled once 300 or more depth-13 updates have been applied; from update 12,500, 300 depth-13 updates leave $+0.123$. †Compared with the 20,500-update descending endpoint instead: -0.008.

Probe	W (nats)	Difference from shuffled (nats)
Parent P8 at update 17,500 (restart point)	5.307	$+1.999$
From 17,500: 500 updates, all at depth 5	5.057	$+1.748$
From 17,500: 1,250 updates, last 300 at depth 13	3.414	$+0.105$
From 17,500: 2,050 updates, last 1,100 at depth 13	3.355	$+0.046$
From 17,500: 3,000 updates, last 2,050 at depth 13	3.337	$+0.028$
Parent P8, uninterrupted to update 20,500	3.348	$+0.039$
From P8 at 12,500: 950 at depth 5, then 300 at depth 13	3.431	$+0.123$
From P8 at 12,500: its next 2,050 updates (depths 7, 6)	3.674	$+0.365$
Descending run at update 17,500	3.862	$+0.553$
41,000-update descending run, modified learning rate	4.054	—†

At update 17,500, the descending run's W (3.862) already exceeds the final shuffled W (3.309) by 0.553 nats, which is 0.733 of its final penalty of 0.754 (4.063 minus 3.309); the reference is the shuffled endpoint, not a shuffled checkpoint at the same update. The last row of Table 11 is a 41,000-update descending run on the primary multiset concatenated with a second draw of 20,500 depths. Over updates 18,450–20,500, all at depth 9, its learning rate follows the 20,500-update cosine, which decays to zero at update 20,500; training then continues on the 41,000-update cosine to the end, and the final checkpoint is evaluated. Its endpoint differs from the 20,500-update descending endpoint by -0.008 nats, and the same 41,000-update run without the learning-rate change reaches 4.064 (Appendix H). These probes include additional training updates or changed allocations and should not be treated as all equal-budget interventions. Terminal placement has little penalty in the

tested cells; the results do not establish a universal rule that training must end at a particular depth. Mean predictive entropy at readout depth 16 is 3.507 for shuffled, 5.080 for descending, 3.454 for ascending, and 3.504 for the seed-3 terminal-placement run. The linear probes use, for each of the 160 validation sequences, the layer-normalized state x_t mean-pooled over its 511 scored positions; features are standardized on the training half, the ridge penalty is 1, and the null distribution comes from 200 paired label flips. Linear probe separability also holds for the ascending–shuffled pair, so separability is not specific to high-loss checkpoints.

H AN ILLUSTRATIVE LINEAR SURROGATE

For a depth-conditioned quadratic loss with shared curvature H and forcing b_T, constant-rate gradient descent with $A = I - \eta H$ follows the recursion $\theta_{s+1} = A\theta_s + \eta b_{T_s}$, which unrolls to Eq. (4) in Section 5. Equal counts do not imply equal weighted sums. Exchanging adjacent depths a, b, with m later updates, changes the endpoint for ab minus ba by $\eta^2 A^m H(b_b - b_a)$. This elementary illustration does not establish the mechanism of nonlinear AdamW training.

The temporal accrual kernel models W after s updates as $W(s) = a\, e^{-s/\tau_1} + b\,\mathrm{age}(s) + c$, where $\mathrm{age}(s)$ is the number of updates since the last update at depth 12 or more. Fitted to W of the seed-3 20,500-update descending run at checkpoints from update 2,500 to 20,500 (τ_1 by grid search over 9,140–9,200), it has $a = 2.8901$, $b = 1.079 \times 10^{-4}$, $c = 2.0289$, and $\tau_1 = 9{,}166$, with training sum of squared errors (SSE) 0.0249. Its prediction at 41,000 updates is $W = 5.983$, compared with 4.064 observed for the 41,000-update descending run. A fitted model of penalty dynamics with parameters shared across interventions also fails to transfer between them. These failures prevent treating the fit as an established extrapolation or repair law.

I MEASURED STEP TIMES AND MODELED RUN TIME

A single NVIDIA L20Z profile includes the forward/backward pass, bf16 computation, gradient clipping, and AdamW update. Each depth has 25 timed repetitions, collected round-robin. Median step times rise from 207.09 ms at depth 5 to 254.17 ms at depth 13; an affine fit is $177.61 + 5.877T$ ms, with maximum absolute residual 0.160 ms. Per-depth coefficients of variation are at most 0.29% except at depth 13, where occasional outliers raise the value to 1.65%.

Multiplying each measured depth median by its common exposure count gives 4,728.24 seconds for every schedule. A 4,000-trial resampling calculation, assuming independent step costs conditional on depth, gives a largest pairwise 95th-percentile absolute relative difference of 0.0128% and a maximum simulated difference of 0.0255%. These are predictions from measured step-time pools, not observed whole-run timing differences or confidence bounds on unmodeled order-dependent effects. The model does not capture allocator history or changes in hardware, batch geometry, or distributed execution.

J PARAMETER COUNTS

The model has two prelude blocks, one shared recurrent core, and one coda block, with context 512 and a 50,257-token vocabulary. Each block uses attention width d and feed-forward width $4d$; its parameter count is $12d^2 + 13d$, including norms and biases. Widths 768 and 1024 use twelve and sixteen heads, respectively. The output projection shares the token embedding, so its duplicate serialized tensor is counted once. Table 12 includes the auxiliary predictor and gives 67,409,889 and 102,458,081 unique parameters. The labels "62M" and "110M" in configuration and run names in the supplementary code are identifiers, not parameter totals.

Table 12: **Parameter counts by tensor group** for both model widths, from the stored checkpoint tensors.

Tensor group	Width 768	Width 1024
Token embedding (tied output)	38,597,376	51,463,168
Positional embedding	393,216	524,288
Prelude blocks ($\times 2$)	14,175,744	25,192,448
Shared recurrent core	7,087,872	12,596,224
Core layer norm	1,536	2,048
Coda block	7,087,872	12,596,224
Final layer norm	1,536	2,048
Auxiliary predictor	64,737	81,633
Stored tensor total	106,007,265	153,921,249
Less duplicate tied output tensor	38,597,376	51,463,168
Unique parameters	**67,409,889**	**102,458,081**
Actual count (SI millions)	67.41M	102.46M