A SIMPSON'S PARADOX IN IN-CONTEXT LEARNING CURVES

Anonymous authors
Paper under double-blind review

ABSTRACT

A language model's average prediction loss often falls as it reads further into a document. This decrease is commonly used to measure in-context learning, but later positions also contain different tokens. We find a striking reversal on web text: average loss falls while loss rises in every group defined by how often a token has appeared before. Later positions contain more repeats, which have lower loss than first occurrences; their increasing share lowers the average. This is a Simpson's paradox. Across eight text domains and two Pythia sizes, domain accounts for 99% of the variance in a standard position-based score, compared with 0.2% for size. Controlled synthetic experiments also change the score more through the data than through model size. Yet models do benefit from context, and continue improving after this score stabilizes. We therefore recommend reporting token-group losses alongside the average, and measuring context benefit by comparing predictions of the same tokens with different amounts of context. An early loss decrease on first-occurrence tokens provides an additional diagnostic of improvements that the conventional score misses.

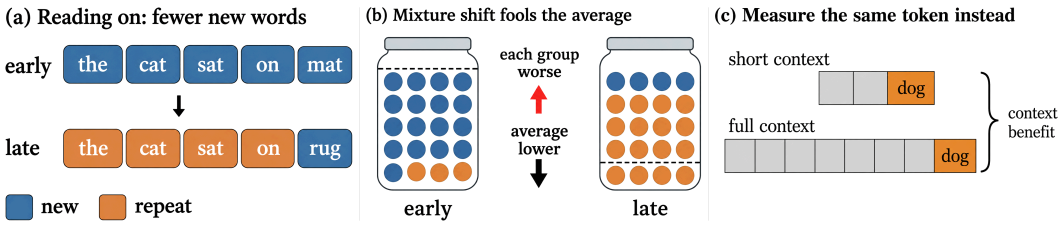

Figure 1: **A changing mixture can look like learning.** (a) As a document is read, fewer tokens are new and more repeat earlier ones. (b) Repeats are easier to predict, so their growing share lowers the average loss even if loss within each group rises; Figure 2 shows this on web text. (c) Comparing predictions of the same target tokens with short and full context isolates the benefit of context (§6).

1 INTRODUCTION

A trained language model often predicts tokens near the end of a document with lower loss than tokens near the beginning. Because later predictions have more preceding text available, this decrease is commonly interpreted as in-context learning: improved prediction from context, without updating model weights (Kaplan et al., 2020; Olsson et al., 2022).

There is a second difference between early and late positions: the tokens being predicted. Early in a document, many tokens appear for the first time. Later, more tokens repeat something already seen. Repeats tend to have lower loss. Their increasing share can therefore lower the average even if prediction loss within each group increases. Figure 2 shows exactly this reversal on web text. Figure 1 summarizes the argument.

For a simple illustration, suppose first occurrences have loss 4 and repeats have loss 2. With equal shares, average loss is 3. Later, suppose both losses rise, to 4.2 and 2.2, but repeats now make up three quarters of the tokens. Average loss falls to 2.7. These illustrative numbers show why the

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.

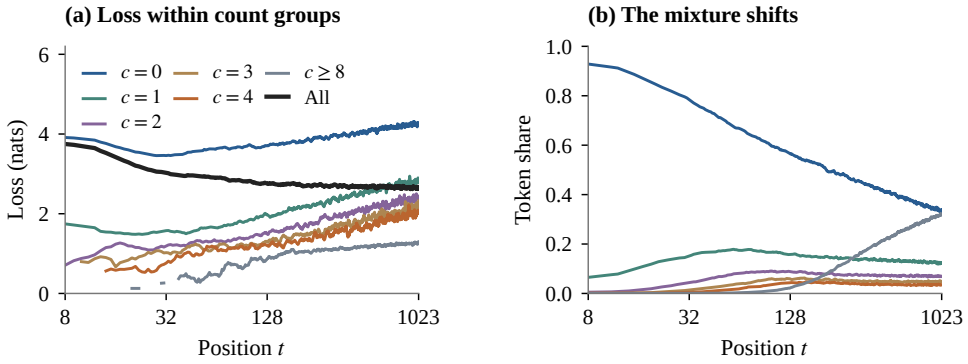

Figure 2: **Average loss falls while loss within every token group rises.** Pythia-1.4B on FineWeb. Left: the black curve averages all tokens; coloured curves group tokens by their number of earlier occurrences in the document ($c = 0$: first occurrence; $c = 1$: seen once before; $c \geq 8$: seen at least eight times). Lower loss means better prediction. Right: later positions contain fewer first occurrences and more frequent repeats, which have lower loss. The reversal concerns positions 64–1023. Curves use centred 9-position means; §3 quantifies the effect.

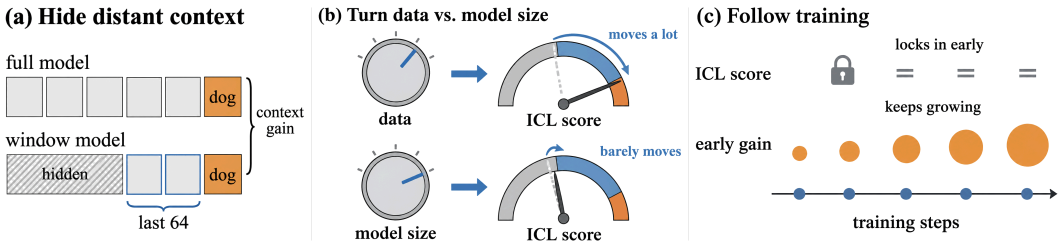

Figure 3: **Three tests distinguish changing token proportions from improved context use.** (a) A window model sees only the last 64 tokens; comparing it with the full model on the same targets measures the gain from distant context (§6). (b) Changing the text distribution moves the ICL score far more than changing model size (§4). (c) During training the score settles early, while the early gain on first occurrences keeps growing (§5).

average alone cannot tell us whether prediction improved within either group. This reversal is known as Simpson's paradox (Simpson, 1951). In our experiments, we use finer groups based on how many times each token has appeared before.

To connect this example to the conventional metric, write $L(t)$ for average loss at position t. A conventional ICL score is $L(500) - L(50)$: a more negative score means a larger loss decrease. This compares different target tokens at two positions. To measure the benefit of extra context for a fixed target, we instead need to predict the *same token* twice, once with shorter context and once with longer context. These two comparisons answer different questions.

Our claim concerns the interpretation of the position-based score on natural documents. Models really do use context: restricting context to the previous 64 tokens worsens prediction in our experiments. Nevertheless, a decreasing position curve, or a larger decrease on one corpus than another, does not by itself isolate better context use. We study the score after the early training transition examined by Olsson et al. (2022), rather than testing few-shot task learning.

We first explain the reversal by separating changes in token shares from changes in loss within groups (§3). We then test its practical consequence: the score changes much more across text domains than across the model sizes we compare (§4). Finally, we show what the score misses, using early first-occurrence losses and a comparison that holds target tokens fixed (§5–6). Figure 3 outlines the three tests.

2 WHAT WE MEASURE

Per-position loss and groups of tokens. For a document $x_1, \ldots, x_T$ and a model q, the loss at position t is $\ell_t = -\log q(x_t \mid x_{<t})$, measured in nats (natural-log units), and $L(t)$ is the mean of ℓ_t over documents.

The *prior count* of the token at position t is $c_t = |\{s < t : x_s = x_t\}|$. In "the cat sat on the mat" the second "the" has $c_t = 1$ and "mat" has $c_t = 0$. We call tokens with $c_t = 0$ *first occurrences* (labelled "novel" in figures) and all others *repeats*. A count group contains tokens with the same number of earlier occurrences. Writing $P(c \mid t)$ for the share of tokens with prior count c at position t and $L_c(t)$ for their mean loss, the per-position loss is exactly a mixture,

$$L(t) = \sum_c P(c \mid t)\, L_c(t). \tag{1}$$

For each group, multiply its share by its average loss, then add the results. Here "first occurrence" means unseen in the current document, not absent from training data. Tokens are tokenizer units, which can be words or parts of words; repeated identity refers to exactly the same token ID.

Data and model comparisons. Our main dataset is FineWeb (Penedo et al., 2024). For each tokenizer, we evaluate the first 3000 documents with at least 1023 tokens, using the model's beginning-of-sequence (BOS) token; the $T = 2047$ evaluation uses 2000 documents. To compare tokenizer families on identical text, we also use 2000 documents that reach 1023 tokens under all four tokenizers. Domain comparisons use 500 documents from each of eight Pile subsets (Gao et al., 2020).

We evaluate thirteen models: four Pythia sizes (Biderman et al., 2023), three GPT-2 sizes (Radford et al., 2019), three SmolLM2 sizes (Ben Allal et al., 2025), and three Qwen2.5 sizes (Yang et al., 2024). Table 1 lists their sizes. Training comparisons use 12, 9 and 7 checkpoints for Pythia-160M, 410M and 1.4B. Pythia-160M denotes step 64000 unless stated: its released final checkpoint has about 0.5 nats higher mean loss on FineWeb and the Pile. Appendix A records the revisions and BOS exceptions. Window experiments re-evaluate the same targets using only the previous $W = 64$ tokens.

Reporting. The two-point score $L(500) - L(50)$ is noisy: its standard error, from resampling documents, is 0.07–0.08 nats at 3000 documents. We therefore also use a position-averaged score, the mean loss over positions 450–550 minus the mean over 25–75, whose standard error is 0.013. Unless stated, "ICL score" below means the position-averaged score. Two-point values are kept where we compare with Olsson et al. (2022) (Appendix L, Figure 9, left). All uncertainties below are standard errors from resampling documents.

3 WHY THE AVERAGE FALLS WHILE GROUP LOSSES RISE

On FineWeb, the overall loss decreases even though loss rises within each group of tokens with the same prior count. For Pythia-1.4B, average loss falls from 2.90 to 2.60 nats between positions 64 and 1023. Over the same interval, all nine count groups become harder to predict (Figure 2). Groups at different positions contain different token instances; no fixed target is becoming harder merely because we provide more context.

The explanation is the changing mixture. First-occurrence tokens have about 2 nats higher mean loss than repeats, but their share falls from 0.67 to 0.34. Meanwhile, tokens seen at least eight times grow from 0.002 to 0.31 of the total. The average shifts toward lower-loss groups. Appendix C reports numerical comparisons (Table 4) and checks using longer documents and a grouping based on repeated pairs of adjacent tokens.

As a document grows, it keeps introducing token identities, but increasingly reuses ones already seen. The number of distinct tokens therefore grows more slowly than document length. This pattern is known as Heaps' law (Heaps, 1978; Font-Clos et al., 2013). It explains why the first-occurrence share tends to decline; the exact exponent is not needed to understand the reversal. Appendix I gives the relation between vocabulary growth and first-occurrence probability.

To quantify how the mixture changes the average, we separate two contributions. First, the shares shift toward groups with lower loss; we call this the *composition* effect. Second, loss changes within the

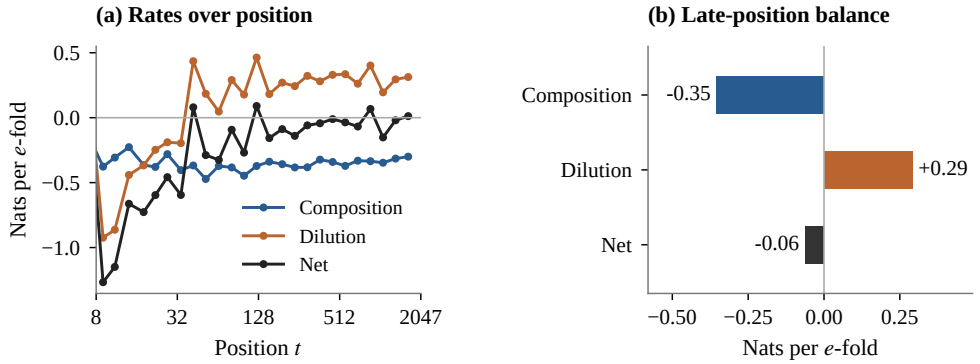

Figure 4: **The late slope is a small difference of two large rates.** Pythia-1.4B on FineWeb, $T = 2047$. Left: composition (blue), dilution (orange), and net loss slope (black). Right: their means for bin centres in $[100, 2047)$.

groups; we call this *within-group change* ("dilution" in the figures). We estimate the two contributions separately, holding one factor fixed while changing the other, then averaging the two possible orders (Kitagawa, 1955). Appendix I gives the exact decomposition. Rates below are measured per e-fold of position: multiplying position by $e \approx 2.72$.

Figure 4 shows that the two terms nearly cancel at late positions. For Pythia-1.4B over $t \in [100, 2047]$, composition contributes -0.35 ± 0.04 and within-group change $+0.29 \pm 0.08$ nats per e-fold (mean $\pm$ standard deviation across position bins), leaving a net slope of -0.06. Composition is similarly negative down to $t \approx 40$. Within-group change is negative only near the start ($t \lesssim 33$), where loss still decreases within groups. Thus the late curve is nearly flat even though the proportion of first occurrences keeps declining: between positions 900 and 2047, loss falls by only 0.04 nats while the first-occurrence proportion falls from 0.34 to 0.28.

Control 1: independent tokens. To test whether rising group losses require context use, we generate independent tokens from a Zipf distribution, whose token probabilities decrease with frequency rank (exponent 1.1, 50k types). Using the generating probabilities to score these tokens gives constant expected loss across positions. Yet the first-occurrence proportion falls from 0.86 to 0.37 over $t = 8 \to 1023$ and within-group losses rise steeply ($7.1 \to 10.2$ for novel tokens, $2.7 \to 7.3$ for once-seen tokens), because later first occurrences and later repeats are rarer types.

Control 2: a 64-token context window. We predict the same targets with Pythia-160M but limit its context to the previous 64 tokens. This *window model* cannot access text before that window. Its aggregate on FineWeb is flat (slope $+0.04$ per e-fold for $t \geq 128$), with composition -0.46 and within-group change $+0.51$. The same model with full context gives -0.45 and $+0.36$. In this comparison, allowing context beyond 64 tokens reduces the within-group increase by 0.15 nats per e-fold, about a third of the window model's rate; composition changes little.

4 The score varies more across data than across model sizes

If changing token shares strongly affects the position curve, how should we interpret differences between models? We compare model sizes on the same text, then compare text domains, and finally change synthetic data while controlling model size. The question is whether a larger loss decrease reliably identifies a better model.

Alongside neural models, we use a predictor that counts earlier occurrences in the document and gives repeated tokens more probability. We call it the *count cache*. A static bigram predictor, fitted on separate text, supplies probabilities using only the previous token, including for tokens unseen in the current document. The cache has four parameters fitted to token sequences, never to neural-model losses (Appendix B). Its role is to test how much of the score a simple repetition rule can reproduce; matching this baseline does not imply that a neural model implements that rule.

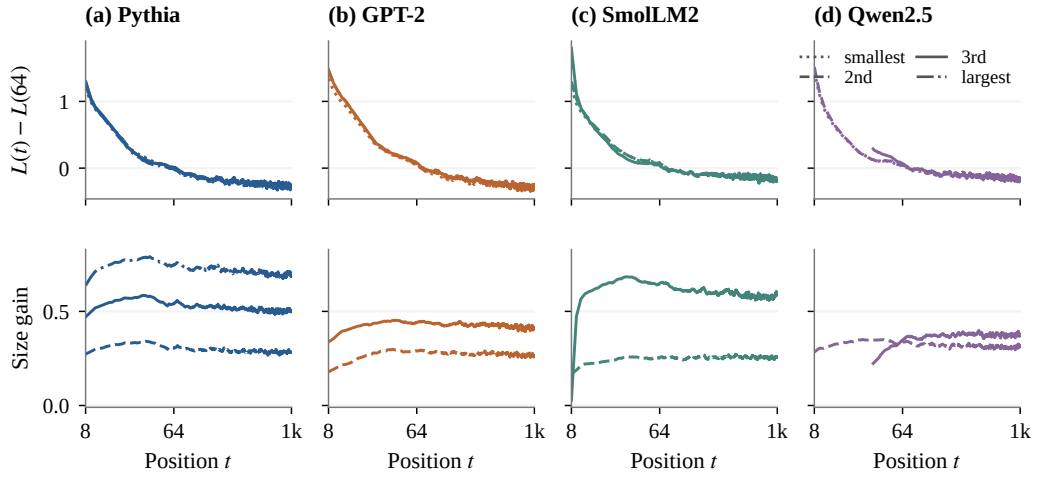

Figure 5: **Model sizes share a shape and differ mainly by an offset.** Thirteen models on 2000 shared FineWeb documents. Columns separate families. Top: loss relative to position 64. Bottom: advantage over the smallest family member. Dotted, dashed, solid and dash-dot lines denote increasing sizes. Qwen2.5-3B starts at $t = 32$. Centred 15-position means; offsets are in Table 1.

Table 1: Shared-document comparison. Offsets are relative to the smallest model in each family (mean $\pm$ s.d. over position). ICL scores are position-averaged, with s.e. 0.017. All values are nats.

Family	Size	Size offset	ICL score	$L(64) - L(1023)$
Pythia	160M	—	-0.32	0.32
	410M	0.29 ± 0.02	-0.29	0.30
	1.4B	0.51 ± 0.03	-0.27	0.27
	2.8B	0.71 ± 0.04	-0.26	0.30
GPT-2	small	—	-0.39	0.19
	medium	0.27 ± 0.02	-0.38	0.18
	large	0.42 ± 0.03	-0.37	0.16
SmolLM2	135M	—	-0.25	0.22
	360M	0.26 ± 0.02	-0.25	0.24
	1.7B	0.59 ± 0.04	-0.17	0.23
Qwen2.5	0.5B	—	-0.21	0.21
	1.5B	0.32 ± 0.02	-0.18	0.19
	3B†	0.37 ± 0.03	-0.29	0.21

Offsets use $t \geq 8$, except $^\dagger t \geq 32$. Static bigram / cache ICL by tokenizer: Pythia $-0.01/-0.31$; GPT-2 $-0.02/-0.32$; SmolLM2 $+0.05/-0.24$; Qwen2.5 $+0.04/-0.26$.

4.1 Model sizes on the same text

Figure 5 (bottom row) plots the difference between the curves of two sizes of the same family on the shared documents. The larger model's loss advantage is nearly constant across positions: the standard deviation over positions $t \geq 8$ is 0.02–0.04 nats in all nine pairs, against offsets of 0.26–0.71 (Table 1). The 0.71 offset compares Pythia-160M at step 64000 with the final Pythia-2.8B; the step-64000 revision published for 2.8B carries the same weights as its final release (Appendix A), so a matched comparison is not available for that size.

This similarity extends across model families. All four tokenizers (49k–152k types) yield a vocabulary-growth exponent of 0.78 over $t = 64$–1023 and a final novel share of 0.30–0.35. Their shifted curves align after $t \approx 30$ (Figure 5, top). Every model's ICL score lies within 0.08 nats of its tokenizer's cache, whereas memoryless references score near zero on FineWeb (Table 1); the 64-token window model scores $+0.02$ on its own 3000 documents. Family differences remain: GPT-2 sits 0.06 below

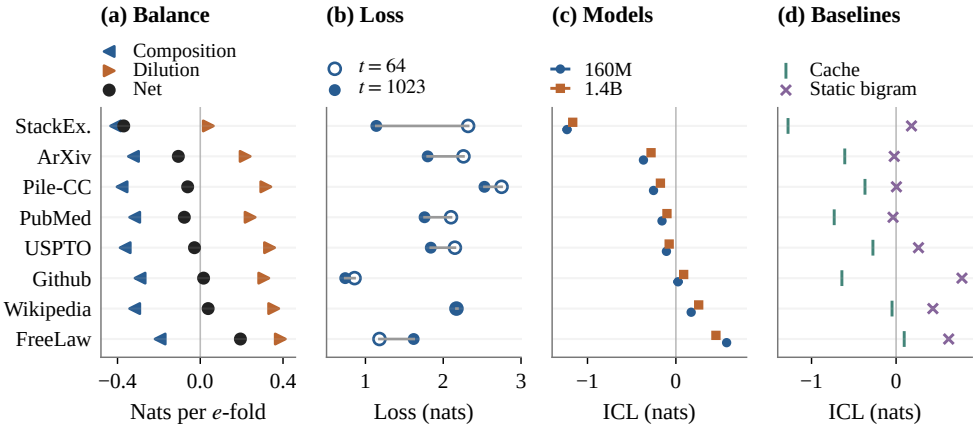

Figure 6: **Domain differences dominate size differences.** Eight Pile domains, position-averaged ICL scores. (a) Composition, dilution and net rates over $t = 64$–1023. (b) Loss at the interval endpoints. (c) Pythia-160M and 1.4B with bootstrap standard errors. (d) Count-cache and static-bigram scores. Shared rows expose how the rate balance changes across domains, while model scores largely agree ($r = 0.993$).

its cache and the largest SmolLM2 0.07 above. Thus model scores are close to a token-count baseline, with smaller differences across model families.

4.2 The same models across text domains

With the same Pythia-1.4B model, loss rises on legal text (FreeLaw), stays roughly flat on Wikipedia and Github, and falls steeply on StackExchange. The two effects from §3 explain the difference: shifting shares lower the average in every domain, but the increase within groups can be small enough to permit a steep decline, or large enough to reverse it (Figure 6).

Across eight Pile domains and two Pythia sizes, a two-way decomposition assigns 99.3% of score variance to domain, 0.2% to size, and 0.5% to their interaction. The two models' scores correlate at 0.993. This is a statement about these domains and model sizes, not a universal variance attribution. The count cache tracks the domain ranking reasonably well ($r = 0.89$) but misses the score values (mean absolute error 0.36 nats). Token counts capture part of the pattern, not all domain differences. Appendix N gives the full comparisons and a prediction of each curve's direction.

4.3 A controlled change to the data

Domain comparisons alone do not establish what causes the score differences. We therefore construct nine synthetic data distributions and train a 4M- and a 21M-parameter transformer on each. The generating rule either repeats a token or introduces one; two parameters control how often new types arrive and how uneven their probabilities are. This gives a known optimal predictor against which to compare the trained models. Appendix G gives the settings and results.

Changing the data changes the trained models' scores by 0.65 nats, whereas increasing model size about $5\times$ changes the score by at most 0.09. The best predictor of the optimal score combines the shift in first-occurrence share with the loss difference between first occurrences and repeats ($r = 0.98$). The rate of vocabulary growth alone is insufficient. Both trained sizes remain worse than the optimal predictor at this training budget, so this experiment establishes a data effect at the tested scale, not model independence at every scale.

Together, the domain comparisons and the synthetic intervention show that the position score is more sensitive to the text distribution than to model size. This does not make it independent of the model. Additional controls show that later repeats remain harder after accounting for token frequency, distance from the previous occurrence, and whether the preceding two-token sequence has appeared before. The raw text shows a corresponding decrease in recurrence probability (Appendix K). These

Table 2: Early first-occurrence loss reveals size differences that the ICL score largely misses. GPT-2 is not monotonic; Qwen2.5-3B has an early-position artifact. $T_{\text{new}} = \bar{L}_{\text{new}}[8, 16] - \bar{L}_{\text{new}}[48, 64]$ and the same on repeats, T_{old}, in nats (bootstrap s.e. 0.03 and 0.04), on each tokenizer's own 3000 FineWeb documents (hence the ICL scores differ slightly from Table 1, which uses the shared 2000).

	Pythia				GPT-2	
	160M	410M	1.4B	2.8B	small	medium / large
T_{new}	0.14	0.17	0.24	0.28	0.22	0.35 / 0.31
T_{old}	0.10	0.08	0.12	0.12	0.22	0.28 / 0.23
ICL score	-0.33	-0.30	-0.28	-0.28	-0.40	-0.39 / -0.38
	SmolLM2			Qwen2.5		
	135M	360M	1.7B	0.5B	1.5B	3B[†]
T_{new}	0.04	0.12	0.20	0.12	0.18	0.92
T_{old}	-0.04	-0.01	0.02	-0.07	-0.05	0.19
ICL score	-0.27	-0.27	-0.20	-0.23	-0.20	-0.31

[†]early-position artifact (Appendix A).

analyses help explain the within-group rise; the measurement problem already follows from the observed reversal.

5 What the position score misses during training

A score that changes little with model size might still track progress during training. We test this next, then examine an earlier part of the document that the conventional score largely misses.

At each checkpoint we evaluate the two-point score $L(500) - L(50)$ and mean loss on the same documents, without further weight updates. For all three Pythia sizes, the score drops sharply by step 1000 (2.1B training tokens). It then stays within 0.07 nats of the count-cache value, -0.28, for another $60\times$ of training, while mean loss falls from 4.4–4.6 to 2.8–3.3 nats (Appendix L, Figure 9, left).

The early drop coincides with the emergence of copying from earlier context reported by Armeni et al. (2024) and induction heads studied by Olsson et al. (2022). Later score variation is comparable to its standard error. The score therefore detects an early change but does not resolve the subsequent improvement in prediction loss.

An early decrease on first occurrences. To look for improvement missed by the late-position score, we examine first-occurrence tokens near the start of documents. Their prediction cannot benefit from copying a previous occurrence of the same token in the document. A loss decrease on these tokens may instead reflect other uses of context, such as topic or style information. Writing $L_{\text{new}}(t) = L_0(t)$ for their mean loss, we measure

$$T_{\text{new}} = \overline{L_{\text{new}}}_{[8,16]} - \overline{L_{\text{new}}}_{[48,64]}, \tag{2}$$

where the bars average loss over the indicated positions (bootstrap s.e. 0.03). Positive T_{new} means that first-occurrence tokens at positions 48–64 have lower mean loss than those at positions 8–16. The same quantity on repeats is T_{old} (Table 2).

T_{new} grows with size in Pythia ($0.14 \to 0.28$), SmolLM2 ($0.04 \to 0.20$) and Qwen2.5 0.5B$\to$1.5B ($0.12 \to 0.18$); GPT-2 medium and large are within noise of each other (0.35, 0.31), and Qwen2.5-3B's value (0.92) is inflated by its early-position artifact. T_{old} is small in Pythia (≤ 0.12) and about zero in SmolLM2 and Qwen2.5, but as large as T_{new} in GPT-2, so the concentration on first occurrences holds in three of the four families.

A separate comparison against the count cache also shows a larger early first-occurrence improvement at later checkpoints (Appendix L, Figure 9). This corroborates the training result without requiring the conventional score to keep changing.

These results motivate reporting the early first-occurrence decrease alongside the conventional score and overall loss. It removes the changing balance between first occurrences and repeats, but still

compares different targets at different positions. It is therefore a diagnostic of a missed improvement, not an isolated estimate of the benefit of additional context. For that estimate, we next hold the target tokens fixed.

6 Measure context benefit on the same target tokens

The previous sections compare losses at different document positions. Here we ask a more direct question: how much does distant context help predict the *same* target token? For every target, we run the model with the full prefix and with only the previous 64 tokens. Subtracting full-context loss from restricted-context loss gives a positive value when distant context helps.

On FineWeb this comparison gives an average gain of 0.343 nats for Pythia-160M and 0.334 for Pythia-1.4B on 600 documents. Thus the mixture explanation coexists with a substantial context benefit. The largest contribution comes from tokens whose earlier occurrences lie entirely outside the 64-token window. We call these *far repeats*: the restricted model cannot see any earlier occurrence, but the full-context model can. Across the evaluated corpora, they constitute 17–25% of tokens and contribute 58–73% of the gain (Appendix H).

Can repetition statistics predict this benefit? As a supporting test, we fit a linear predictor of the measured gain using token frequency and occurrence history. We train on eight corpora and test on the ninth. This predictor uses measured model gains during fitting; only its inputs at prediction time are token statistics. Appendix M specifies the features and the token groups.

For Pythia-160M, the predictor tracks the mean gain per far-repeat token across held-out corpora, which varies from 1.0 nats on FineWeb to 1.8 on Github, with correlation 0.93 and a mean absolute error of 0.08 nats (Figure 7). Counting far repeats is not enough: a baseline that uses class proportions but not token features produces predictions negatively correlated with measured total gain (-0.80). Class proportions therefore do not capture how gain per token varies across domains.

The total gain per corpus is predicted less well: mean absolute error 0.037 nats against 0.057 for predicting each corpus by the mean gain of the other eight, better on five of nine corpora and worst on FreeLaw, whose novel-token gain of 0.35 nats (against 0.12–0.23 elsewhere) is not captured by these count features; a possible explanation is greater use of topic information.

Applied to Pythia-1.4B, the same count-based prediction (0.344) is within 0.01 nats of the measured gain (0.334) showing similar average benefit from distant context in this comparison despite a $9\times$ size difference; the predictor has no model-specific input.

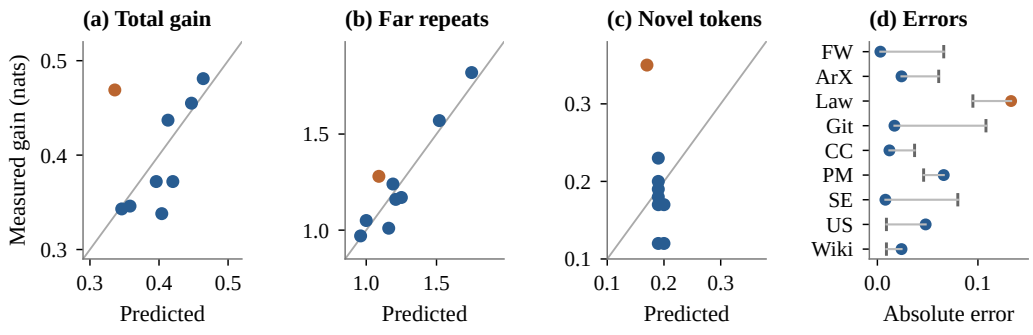

Figure 7: **The copy part of long-context gain from token statistics.** Leave-one-corpus-out predictions for Pythia-160M on FineWeb and eight Pile domains: (a) total gain beyond 64 tokens, (b) far-repeat gain, and (c) novel-token gain. Diagonals indicate exact prediction; orange marks FreeLaw, whose novel gain is underestimated. (d) Absolute total-gain errors: full predictor (dots) versus the leave-one-out constant (ticks), in nats; connected marks compare the same corpus.

7 Related work

The per-position curve. Prior work fits scaling laws to position or context length (Kaplan et al., 2020; Xiong et al., 2025; 2024), links the ICL score to induction heads (Olsson et al., 2022), and dates retrieval emergence (Armeni et al., 2024). Distant context chiefly helps repeated, copyable tokens (Armeni et al., 2022; Sun et al., 2021; Khandelwal et al., 2018). We connect these observations to the changing shares of first occurrences and repeats by grouping tokens according to prior count. Conditional-entropy exponents also enter data-scaling laws (Cagnetta et al., 2026).

Vocabulary growth and prediction loss. The relation between entropy decay $n^{\beta-1}$ and Heaps' law predates this work (Hilberg, 1990; Dębowski, 2018; 2026; Takahashi & Tanaka-Ishii, 2018). Our additions are the count decomposition, opposing-rate explanation of flattening, and separation of early model gain from the late ICL score. Bursty Zipfian data drive ICL emergence (Chan et al., 2022); Bayesian explanations include Xie et al. (2022); Arora et al. (2025). Stationary-source mixtures yield $1/t$ decay (Riechers et al., 2025); unbounded vocabulary permits fractional decay, consistent with nonparametric predictive information (Bialek et al., 2001). Our references use classical cache and Pitman–Yor models (Kuhn & De Mori, 1990; Teh, 2006).

8 Conclusion

Average loss can fall because later positions contain more low-loss repeats, even while loss rises within every count group. On our evaluated documents, this makes the conventional position score strongly dependent on the text distribution and insensitive to some improvements during training. Report group losses alongside the average, use early first-occurrence loss as a complementary diagnostic, and compare the same target tokens under different context lengths when measuring context benefit. These measurements distinguish a changing evaluation mixture from a model's benefit from additional context.

Limitations. Our mixture and score comparisons concern natural documents, mainly beyond the first few dozen tokens, not few-shot prompts. The early diagnostic examines positions 8–64 and remains sensitive to changes in target difficulty within the first-occurrence group. Domain comparisons use each tokenizer's own documents. The synthetic intervention spans 4M–21M parameters at 10^8 training tokens, where models still have higher loss than the Bayes predictor. The cache tracks cross-domain ICL scores in rank, not level.

Reproducibility statement

All evaluated models are public checkpoints; Appendix A lists the model revisions, BOS conventions and document selection for FineWeb and the Pile. The cache reference is specified in Appendix B, the regression designs in Appendices E and K, and the synthetic worlds and training settings in Appendices F and G. The supplementary material contains anonymized code for every probe, analysis and simulation, the command sequences used for the GPU runs, and text logs of the reported analyses.

AI use statement

We used generative AI tools to aid and polish the writing of this paper. They were not used to run experiments or to produce the reported results. We reviewed all AI-assisted text and take responsibility for the final content of this work.

References

Eduardo G. Altmann, Janet B. Pierrehumbert, and Adilson E. Motter. Beyond word frequency: bursts, lulls, and scaling in the temporal distributions of words. *PLoS ONE*, 4(11):e7678, 2009. doi: 10.1371/journal.pone.0007678.

Kristijan Armeni, Christopher Honey, and Tal Linzen. Characterizing verbatim short-term memory in neural language models. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pp. 405–424, 2022. doi: 10.18653/v1/2022.conll-1.28.

Kristijan Armeni, Marko Pranjić, and Senja Pollak. Transformer verbatim in-context retrieval across time and scale. In *Proceedings of the 28th Conference on Computational Natural Language Learning (CoNLL)*, pp. 56–68, 2024. doi: 10.18653/v1/2024.conll-1.6. arXiv:2411.07075.

Aryaman Arora, Dan Jurafsky, Christopher Potts, and Noah D. Goodman. Bayesian scaling laws for in-context learning. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2410.16531.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. SmolLM2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural Computation*, 13(11):2409–2463, 2001. doi: 10.1162/089976601753195969.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430, 2023.

Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. Deriving neural scaling laws from the statistics of natural language. *arXiv preprint arXiv:2602.07488*, 2026. ICML 2026.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya K. Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 18878–18891, 2022.

Kenneth W. Church and William A. Gale. Poisson mixtures. *Natural Language Engineering*, 1(2): 163–190, 1995. doi: 10.1017/S1351324900000139.

Łukasz Dębowski. Is natural language a perigraphic process? The theorem about facts and words revisited. *Entropy*, 20(2):85, 2018. doi: 10.3390/e20020085.

Łukasz Dębowski. From Zipf's law to neural scaling through Heaps' law and Hilberg's hypothesis. *Journal of Machine Learning Research*, 27(194):1–35, 2026.

Francesc Font-Clos, Gemma Boleda, and Álvaro Corral. A scaling law beyond Zipf's law and its relation to Heaps' law. *New Journal of Physics*, 15(9):093033, 2013. doi: 10.1088/1367-2630/15/9/093033.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Harold Stanley Heaps. *Information retrieval: computational and theoretical aspects*. Academic Press, New York, 1978.

Wolfgang Hilberg. Der bekannte Grenzwert der redundanzfreien Information in Texten—eine Fehlinterpretation der Shannonschen Experimente? *Frequenz*, 44(9–10):243–248, 1990. doi: 10.1515/freq.1990.44.9-10.243.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 284–294, 2018. doi: 10.18653/v1/P18-1027.

Evelyn M. Kitagawa. Components of a difference between two rates. *Journal of the American Statistical Association*, 50(272):1168–1194, 1955. doi: 10.1080/01621459.1955.10501299.

Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583, 1990. doi: 10.1109/34.56193.

Nikolay Mikhaylovskiy. Zipf's and Heaps' laws for tokens and LLM-generated texts. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15469–15481, 2025. doi: 10.18653/v1/2025.findings-emnlp.837.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL `https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/`. Also arXiv:2209.11895.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 37, 2024.

Jim Pitman and Marc Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997. doi: 10.1214/aop/1024404422.

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

Paul M. Riechers, Henry R. Bigelow, Eric A. Alt, and Adam Shai. Next-token pretraining implies in-context learning. *arXiv preprint arXiv:2505.18373*, 2025.

Edward H. Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951. doi: 10.1111/j.2517-6161.1951.tb00088.x.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 807–822, 2021. doi: 10.18653/v1/2021.emnlp-main.62.

Shuntaro Takahashi and Kumiko Tanaka-Ishii. Cross entropy of neural language models at infinity—a new bound of the entropy rate. *Entropy*, 20(11):839, 2018. doi: 10.3390/e20110839.

Yee Whye Teh. A hierarchical Bayesian language model based on Pitman–Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pp. 985–992, 2006. doi: 10.3115/1220175.1220299.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 4643–4663, 2024. doi: 10.18653/v1/2024.naacl-long.260.

Yizhe Xiong, Xiansheng Chen, Xin Ye, Hui Chen, Zijia Lin, Haoran Lian, Zhenpeng Su, Wei Huang, Jianwei Niu, Jungong Han, and Guiguang Ding. Temporal scaling law for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 24463–24483, 2025. doi: 10.18653/v1/2025.emnlp-main.1244. arXiv:2404.17785.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

A MODELS, CHECKPOINTS AND DOCUMENT SETS

Table 3: Exact model revision and document set behind each result.

result	models (revision)	documents
Figs. 2, 4; Table 4	Pythia-1.4B (final)	FineWeb, Pythia tokenizer, 3000 docs ($T = 1023$) / 2000 ($T = 2047$)
Fig. 5; Table 1	Pythia 160M/410M/1.4B (step 64000), 2.8B (final); other families final	2000 docs shared by the four tokenizers
Window control (§3); §6; App. H	Pythia-160M (step 64000)	FineWeb 3000 docs; 500 docs per Pile subset
Table 6; Fig. 6	Pythia-160M (step 64000), 1.4B (final)	500 docs per Pile subset
Tables 8, 2	Pythia 160M/410M (step 64000), 1.4B and 2.8B (final); others final	each tokenizer's first 3000 FineWeb docs of ≥ 1023 tokens
Fig. 9	Pythia 160M/410M/1.4B checkpoints	FineWeb 3000 docs

Qwen2.5 has no BOS token and is scored after its end-of-text token; Qwen2.5-3B is worse than 0.5B for $t \leq 16$ under this convention and SmolLM2-1.7B has $L(1) = 13.3$. Both artifacts are confined to $t < 32$. Pythia-160M's released final checkpoint has mean loss 3.88 on FineWeb against 3.32 at step 64000; the final checkpoint gives the same Simpson table and ICL score (-0.31 two-point) and a slightly larger position coefficient in Table 8. The step-64000 revision that the model hub publishes for Pythia-2.8B carries weights identical to its final release (every tensor equal), so no step-64000 comparison exists for that size; the 2.8B numbers are its final checkpoint.

B CACHE PREDICTIVE RULE

$$p(x \mid x_{<t}) = \frac{\max(c_x - d, 0)}{\theta + t - 1} + \frac{\theta + dK_{t-1}}{\theta + t - 1}\, G(x \mid x_{t-1}), \tag{3}$$

The first term allocates probability using previous occurrences; the second uses the bigram reference, including for unseen tokens. Here d discounts each count and θ controls the reference weight, with c_x the count of x so far, K_{t-1} the number of distinct types so far, G the Kneser–Ney bigram fitted on 20,000 held-out FineWeb documents, and a second cache of the same form on bigrams. The four parameters are fitted on the first 600 documents; the cache value is -0.31 on all 3000 documents and -0.30 on the 2400 not used for fitting. Two-point values: -0.28 and -0.29.

C SIMPSON TABLES FOR ALL MODELS

Table 4: Loss by count group at $t = 64 \to 1023$ on FineWeb (nats) and the share of tokens in each group (Pythia tokenizer). Loss rises in every displayed group; overall loss falls as lower-loss groups become more common.

	$c = 0$ (novel)	$c = 1$	$c = 2$	$c = 3$	$c \geq 8$	all tokens
Pythia-1.4B	$3.60 \to 4.14$	$1.71 \to 2.70$	$1.31 \to 2.46$	$1.16 \to 2.23$	$0.19 \to 1.30$	$2.90 \to 2.60$
SmolLM2-1.7B	$3.15 \to 3.72$	$1.46 \to 2.36$	$1.10 \to 2.13$	$1.04 \to 2.10$	$0.68 \to 1.06$	$2.47 \to 2.26$
Qwen2.5-0.5B	$3.87 \to 4.61$	$1.71 \to 3.07$	$1.37 \to 2.61$	$1.23 \to 2.21$	$0.75 \to 1.31$	$3.06 \to 2.80$
share of tokens	$0.67 \to 0.34$	$0.17 \to 0.11$	$0.08 \to 0.08$	$0.05 \to 0.05$	$0.002 \to 0.31$	

Under the alternative grouping (novel; repeat whose bigram is new; repeat whose bigram was seen), Pythia-1.4B gives $3.60 \to 4.14$, $1.68 \to 2.22$ and $0.67 \to 1.05$ with shares $0.67 \to 0.34$, $0.26 \to 0.43$ and $0.07 \to 0.23$. The two-rate decomposition is produced by `scripts/analyze_two_rates.py` on log-spaced bins from $t = 4$, dropping strata with fewer than 30 tokens in a bin; the numbers in §3 are means over bins with centre in $[100, 2047]$ with the standard deviation across bins as the uncertainty.

Table 5: $L_c(64) \to L_c(1023)$ by prior count; total in the last column.

model	$c = 0$	$c = 1$	$c = 2$	$c = 3$	$c \geq 8$	total
Pythia-160M	$4.36 \to 5.11$	$2.07 \to 3.39$	$1.57 \to 3.05$	$1.38 \to 2.76$	$0.52 \to 1.57$	$3.50 \to 3.22$
Pythia-1.4B	$3.60 \to 4.14$	$1.71 \to 2.70$	$1.31 \to 2.46$	$1.16 \to 2.23$	$0.19 \to 1.30$	$2.90 \to 2.60$
Pythia-2.8B	$3.47 \to 3.91$	$1.64 \to 2.59$	$1.30 \to 2.44$	$1.13 \to 2.04$	$0.25 \to 1.25$	$2.80 \to 2.48$
GPT-2 small	$4.44 \to 5.26$	$2.19 \to 3.70$	$1.61 \to 3.12$	$1.60 \to 3.03$	$0.48 \to 1.62$	$3.52 \to 3.33$
GPT-2 large	$3.95 \to 4.65$	$1.81 \to 3.13$	$1.35 \to 2.76$	$1.28 \to 2.57$	$0.34 \to 1.41$	$3.09 \to 2.90$
SmolLM2-135M	$3.96 \to 4.80$	$1.85 \to 3.05$	$1.35 \to 2.87$	$1.27 \to 2.62$	$0.51 \to 1.34$	$3.09 \to 2.91$
SmolLM2-1.7B	$3.15 \to 3.72$	$1.46 \to 2.36$	$1.10 \to 2.13$	$1.04 \to 2.10$	$0.68 \to 1.06$	$2.47 \to 2.26$
Qwen2.5-0.5B	$3.87 \to 4.61$	$1.71 \to 3.07$	$1.37 \to 2.61$	$1.23 \to 2.21$	$0.75 \to 1.31$	$3.06 \to 2.80$
Qwen2.5-3B	$3.41 \to 4.05$	$1.49 \to 2.60$	$1.15 \to 2.15$	$1.03 \to 1.96$	$0.63 \to 1.16$	$2.68 \to 2.44$

D Domain scores across families and the flattening prediction

Table 6: Position-averaged ICL score per Pile domain (bootstrap s.e. 0.03–0.05; 0.12 for GPT-2-large on Github) for four models from three tokenizer families; each model scores its own tokenizer's first 500 documents of ≥ 1023 tokens per domain. Last three columns: Pythia-1.4B composition, dilution and net rates over $t = 64$–1023 (nats per e-fold).

Domain	160M	1.4B	GPT-2-L	Qwen-1.5B	Comp.	Dil.	Net
ArXiv	-0.37	-0.28	-0.62	-0.31	-0.32	$+0.22$	-0.11
FreeLaw	$+0.57$	$+0.45$	-0.39	$+0.31$	-0.19	$+0.39$	$+0.19$
Github	$+0.02$	$+0.09$	-0.10	-0.12	-0.29	$+0.31$	$+0.02$
Pile-CC	-0.25	-0.18	-0.38	-0.25	-0.38	$+0.32$	-0.06
PubMed	-0.16	-0.10	-0.68	-0.11	-0.32	$+0.24$	-0.08
StackEx.	-1.23	-1.17	-1.49	-1.12	-0.41	$+0.04$	-0.37
USPTO	-0.11	-0.08	-0.33	-0.24	-0.36	$+0.34$	-0.03
Wikipedia	$+0.17$	$+0.26$	$+0.07$	$+0.20$	-0.32	$+0.36$	$+0.04$
Correlation with 1.4B	0.99		0.88	0.99			

Flattening prediction. For each corpus the composition rate over $t \in [64, 1023]$ is computed from the token shares $P(c \mid t)$ with every stratum held at its loss at $t = 64$, and the FineWeb dilution rate $+0.30$ is added. Predicted against observed net slope (nats per e-fold): ArXiv $-0.10/-0.13$, FreeLaw $+0.12/+0.15$, Github $+0.15/0.00$, Pile-CC $-0.08/-0.07$, PubMed $-0.05/-0.08$, StackExchange $-0.14/-0.36$, USPTO $-0.04/-0.04$, Wikipedia $+0.07/+0.03$, FineWeb $-0.09/-0.08$; correlation 0.79, sign agreement $8/9$ (Github is the miss).

E Within-document regression

Pooled across documents, $\log K_t$ (types so far) absorbs the position dependence of the once-seen loss and $\log t$ turns negative, because lexically diverse documents are harder. With a separate intercept for each document (document fixed effects), $\log t$ keeps its slope ($+0.22$ for 1.4B, $+0.25$ for 160M on $c = 1$) and $\log K_t$ is -0.02 ± 0.07; the two regressors are nearly collinear within a document ($K_t \approx t^{\beta}$), so the standard errors are nine times those of the pooled regression, and for $c = 2$ the $\log K_t$ coefficient is -0.20 ± 0.08. Inside a document the loss on a repeat is diluted by the number of tokens with a slope of 0.1–0.4 rather than the exchangeable-cache value of 1; between documents the level is set by the type–token ratio.

F Bursty Pitman–Yor world

To make recent occurrences more influential, replace raw counts with weighted counts $n_x(t) = \sum_{s<t, x_s=x} (1 + (t-s)/\Delta_0)^{-\gamma}$ in the Pitman–Yor rule ($d = 0.6$, $\theta = 10$, $\Delta_0 = 8$, Zipf-1.1 base). The parameter γ controls how quickly an occurrence loses weight with distance. The slope of $-\log h$ against $\log t$ at fixed Δ for once-seen candidates is 0.9–1.2 everywhere at $\gamma = 0$ (exchangeable) and 0.5 (near) to 1.0 (far) at $\gamma = 0.5$; the slope against $\log \Delta$ beyond $\Delta = 8$ is $+0.05$, $+0.19$ and $+0.37$ at $\gamma = 0$, 0.5 and 0.3. FineWeb has 0.34 (near) to 0.62 (far) and $+0.24$ to $+0.45$. Thus this simple

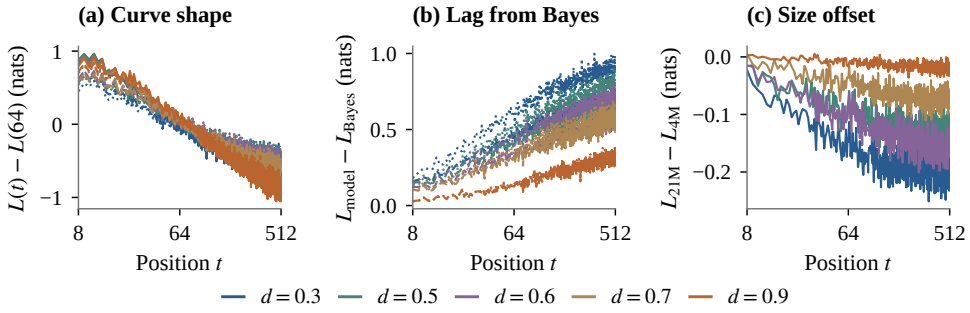

Figure 8: Discount sweep at $s = 1$. Left: loss relative to position 64 (solid: Bayes; dotted: 4M; dashed: 21M). Middle: model loss minus Bayes loss, increasing with position. Right: the nearly constant loss difference between model sizes.

recency rule can produce both position and distance effects; its parameters were not fitted to match the FineWeb measurements. Recency weighting also raises the Heaps exponent above d (0.45, 0.69, 0.78 for $\gamma = 0, 0.5, 1$).

G Intervention details

Figure 8 shows the discount sweep at $s = 1$ for the Bayes predictor and both trained sizes; the table below lists the scores in all nine synthetic worlds.

Table 7: Score $L(400) - L(50)$ in the nine synthetic worlds; the Heaps exponent is measured on the generated documents.

world	Heaps (data)	Bayes	4M	21M	$L_{21M} - L_{4M}$	lag s.d. 4M / 21M
$d = 0.3$	0.43	-0.77	-0.46	-0.50	-0.18 ± 0.03	0.15 / 0.12
$d = 0.5$	0.48	-0.84	-0.56	-0.61	-0.12 ± 0.03	0.14 / 0.12
$d = 0.6$	0.51	-0.84	-0.60	-0.68	-0.14 ± 0.03	0.13 / 0.10
$d = 0.7$	0.53	-0.96	-0.69	-0.73	-0.06 ± 0.02	0.12 / 0.10
$d = 0.9$	0.58	-1.11	-0.95	-0.96	-0.02 ± 0.01	0.08 / 0.07
$s = 0.6$	0.66	-0.83	-0.27	-0.33	-0.10 ± 0.03	0.24 / 0.22
$s = 0.8$	0.59	-0.94	-0.48	-0.52	-0.10 ± 0.03	0.19 / 0.17
$s = 1.2$	0.42	-0.75	-0.63	-0.67	-0.07 ± 0.02	0.07 / 0.06
$s = 1.5$	0.32	-0.44	-0.38	-0.40	-0.03 ± 0.01	0.02 / 0.02

Both the gap from Bayes and the size advantage shrink as repetition increases. These results concern the tested training budget, at which both models still have higher loss than Bayes.

H Anatomy of far-context gain and per-corpus prediction

FineWeb gain beyond 64 tokens averages 0.343 nats (160M) and 0.334 (1.4B) on 600 documents. First table: class shares and gains, with "both" included in "window". Remaining tables: step-64000 Pythia-160M total-gain predictions and per-token far-repeat gain (Far). Position-curve R^2: 0.67–1.00 on eight corpora; -0.41 on FreeLaw.

Class	Share	160M	1.4B
Novel	.42	.17	.21
Window	.34	.12	.12
Far repeat	.24	.97	.86

Corpus	Actual	Pred.	Const.	Far
FineWeb	0.343	0.346	0.409	0.97
ArXiv	0.437	0.413	0.376	1.24
FreeLaw	0.469	0.336	0.374	1.28
Github	0.481	0.464	0.373	1.82
Pile-CC	0.346	0.358	0.383	1.05

Corpus	Actual	Pred.	Const.	Far
PubMed	0.338	0.404	0.384	1.01
StackEx.	0.455	0.447	0.375	1.57
USPTO	0.372	0.420	0.381	1.17
Wikipedia	0.372	0.396	0.381	1.16

I ACCOUNTING FOR CHANGES IN THE AVERAGE

A count group is also called a *stratum* below. The within-group change is labelled *dilution* in the plots; it is an accounting term and does not by itself identify a mechanism.

Why first occurrences become less common. Let K_t count distinct token identities (types) among the first t tokens. Heaps' law states that $K_t \propto t^{\beta}$ with $\beta < 1$: distinct types accumulate more slowly than tokens. The expected increment in K_t is the first-occurrence probability, giving $P_{\text{new}}(t) = P(0 \mid t) \propto t^{\beta-1}$ (Heaps, 1978; Font-Clos et al., 2013). Under byte-pair-encoding (BPE) tokenizers, which all our models use, the exponent is about 0.8 (Mikhaylovskiy, 2025).

How much of the overall slope comes from changing group proportions, and how much from changing loss within groups? A standard two-factor decomposition from demography (Kitagawa, 1955) separates the changing shares from the changing conditional losses. In differential form,

$$\frac{dL}{d\log t} = \underbrace{\sum_c L_c(t)\frac{dP(c \mid t)}{d\log t}}_{\text{composition: shares change}} + \underbrace{\sum_c P(c \mid t)\frac{dL_c(t)}{d\log t}}_{\text{dilution: groups change}}. \tag{4}$$

We call the first term *composition*: proportions change while group losses are held fixed. We call the second *dilution*: losses change within the groups. Both are slopes against $\log t$; one e-fold means multiplying position by $e \approx 2.72$. We estimate them between position bins using a symmetric two-factor decomposition.

Why loss rises within groups. Suppose the aggregate is nearly flat on an interval while the shares keep shifting toward lower-loss groups. If the overall slope is zero, differentiating Eq. 1 gives $\sum_c \partial_t P(c \mid t)\, L_c(t) = -\sum_c P(c \mid t)\, \partial_t L_c(t)$. The left side is negative, so the share-weighted average of the stratum slopes must be positive. This is arithmetic, not a mechanism, and it constrains the average, not each stratum; that all nine strata rise is an empirical fact.

The two terms need not cancel on other data. The near-cancellation is not generic. For comparison, we generate synthetic documents from a rule that alternates between repeating earlier tokens and introducing new types (the Pitman–Yor process described in Appendix N, with parameter $d = 0.6$). For the predictor that knows this rule, composition is between -2.2 and -0.4 per e-fold over $t = 9 \to 630$, dilution falls from $+1.4$ to about zero, and the net slope stays between -0.2 and -0.6: the curve keeps falling, because that world's novel share decays fast (0.25 at $t = 64$, 0.03 at $t = 1024$). Natural text under BPE decays slowly ($P_{\text{new}} \propto t^{-0.25}$, still 0.28 at $t = 2047$), so its shares shift at a constant logarithmic pace and dilution keeps up.

J ADDITIONAL REFERENCE PREDICTORS AND EXPERIMENTAL DESIGNS

Reference predictors. We compare the full model with three references that isolate different sources of context gain:

Reference	Context available	What it isolates
Static bigram	Previous token	Predicts from two-token frequencies fitted on separate text; uses Kneser–Ney smoothing for rare pairs.
Window model	Previous 64 tokens	The same neural model's knowledge and local context, with distant memory removed.
Count cache	In-document counts	Increases probability for tokens already seen, reserving probability for unseen tokens through the static bigram.

This count-based predictor is called a cache because it retains occurrence counts from the current document. It has four scalar parameters fitted on token sequences, never on neural-model losses (Appendix B); the Pitman–Yor process models how new types keep appearing (Pitman & Yor, 1997; Teh, 2006). Its FineWeb ICL score is -0.31 position-averaged and -0.28 two-point; we quote the version matching the model score.

K WHY REPEATED TOKENS BECOME HARDER TO PREDICT

Holding prior count fixed still leaves differences between tokens. Later repeats may be rarer in the corpus, farther from their previous occurrence, or less often part of a repeated bigram. We test whether these features explain the within-group loss increase.

For tokens seen exactly once before ($c = 1$, $t \geq 64$; about 390k tokens per model), we fit

$$\ell_t = a + b_t \log t + b_\Delta \log \Delta + b_f \log f + b_C C + \varepsilon_t. \tag{5}$$

Here Δ is the distance to the previous occurrence, f is corpus frequency, and $C = 1$ if the current two-token sequence has occurred earlier in the document (0 otherwise). We call this last feature *copyability*. ε_t is the residual; each model has its own fitted coefficients. The coefficient b_t measures the association with position after controlling for the other features. Table 8 compares it with the slope from regressing on $\log t$ alone ("raw $\log t$").

Table 8: Regression of the loss on once-seen tokens ($c = 1$, $t \geq 64$) on four covariates, thirteen models. Standard errors that allow tokens of the same document to be correlated are 0.007 ($\log t$), 0.004 ($\log \Delta$), 0.002 ($\log f$) and 0.012–0.014 (copyable). Last row: logistic regression of whether a once-seen candidate token actually recurs, on the same covariates and no model; coefficients are on the log-odds of recurrence, with the sign flipped so that positive means harder, as above.

model	raw $\log t$	$\log t$	$\log \Delta$	$\log f$	copyable
Pythia-160M	$+0.54$	$+0.23$	$+0.20$	-0.25	-2.26
Pythia-410M	$+0.49$	$+0.22$	$+0.21$	-0.19	-2.02
Pythia-1.4B	$+0.43$	$+0.21$	$+0.20$	-0.14	-1.75
Pythia-2.8B	$+0.41$	$+0.21$	$+0.18$	-0.12	-1.66
GPT-2 small	$+0.50$	$+0.20$	$+0.21$	-0.21	-2.11
GPT-2 medium	$+0.45$	$+0.20$	$+0.20$	-0.16	-1.85
GPT-2 large	$+0.46$	$+0.20$	$+0.21$	-0.16	-1.84
SmolLM2-135M	$+0.51$	$+0.20$	$+0.21$	-0.22	-2.01
SmolLM2-360M	$+0.46$	$+0.20$	$+0.20$	-0.16	-1.77
SmolLM2-1.7B	$+0.39$	$+0.19$	$+0.18$	-0.11	-1.50
Qwen2.5-0.5B	$+0.47$	$+0.21$	$+0.20$	-0.17	-1.86
Qwen2.5-1.5B	$+0.40$	$+0.20$	$+0.19$	-0.11	-1.59
Qwen2.5-3B	$+0.38$	$+0.18$	$+0.21$	-0.09	-1.55
s.d. across models		0.014	0.010	0.048	0.22
corpus recurrence (no model)		$+0.37$	$+0.31$	-0.37	-3.73

Controlling for the other features roughly halves the position coefficient. Position and distance coefficients then vary little across models (0.18–0.23 and 0.18–0.21, respectively), whereas frequency and copyability coefficients shrink more with model quality (Table 8). Position dependence is not exactly size-invariant: its decrease from 0.23 to 0.18 spans several standard errors.

These are average associations, not accurate per-token predictions: the regression explains 11–18% of loss variance, versus 2% using position alone. A regression with a separate intercept for each document still finds a position effect after controlling for the number of distinct types (Appendix E).

The corpus has the same structure. The same questions can be asked of the raw text with no model. Let $h(t, \Delta)$ be the probability that a token last seen Δ positions ago is the token that comes next at position t, counted over every (position, candidate) pair in 3000 documents; we call it the *recurrence hazard*. Expressed as a loss, $-\log h$ rises by 0.3–0.6 per e-fold of t at fixed Δ, a within-group increase in the same direction as the models show. As a function of distance Δ, this negative log probability is U-shaped, with a minimum at $\Delta \approx 4$–16 and then a rise of $+0.24$ to $+0.45$ per e-fold: words recur in bursts (Church & Gale, 1995; Altmann et al., 2009).

A logistic regression of the recurrence event on the same four covariates (1500 documents, negatives subsampled to 8.7M rows) gives 0.37 for position and 0.31 for distance (Table 8, last row). The models show the corpus's two effects with the same sign and comparable size (the loss and log-odds scales differ, so the comparison is qualitative). A Pitman–Yor cache that ignores the order of past tokens predicts a slope of 1 against $\log t$ and no dependence on $\log \Delta$; giving recent occurrences more weight in its counts reproduces both trends (Appendix F).

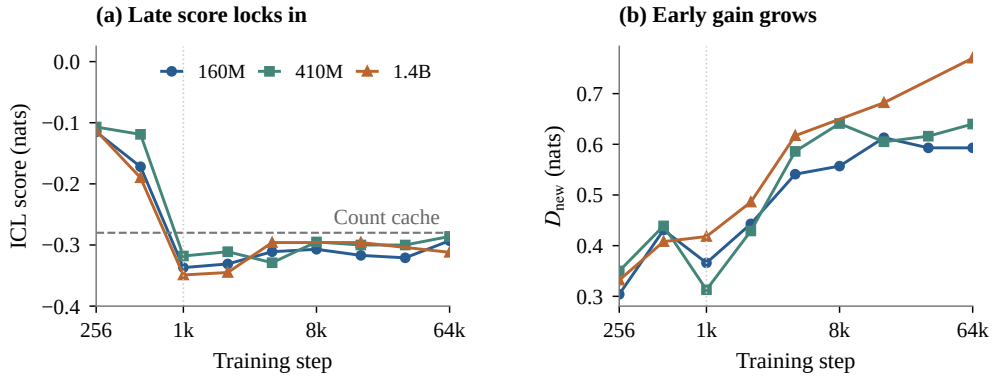

Figure 9: **The late score stabilizes while early novel-token gain grows.** Three Pythia sizes over training. Left: two-point ICL score and count-cache reference (-0.28, dashed). Right: D_{new}, the model's advantage over the cache on novel tokens gained from positions 8 to 64. The vertical guide marks step 1000.

L Early improvement relative to the cache during training

For comparison with the count cache, define D_{new} as the increase from position 8 to 64 in the loss gap $L_{\text{cache,new}} - L_{\text{model,new}}$. A positive value means the model's advantage over the cache is greater at position 64. This measure grows over training from 0.30 to 0.59 (160M), 0.35 to 0.64 (410M) and 0.33 to 0.77 (1.4B), with a dip at the lock-in step (Figure 9, right).

The ordering of model sizes by this measure holds for every end point between 32 and 256. Its magnitude keeps growing past 64, because the cache's novel-token loss rises like $\log t$, so we do not call it a transient of fixed length. This early loss decrease is consistent with the stronger early context benefit reported for larger models by Kaplan et al. (2020). In three of the four families it is concentrated on first occurrences, a distinction the score $L(500) - L(50)$ does not measure directly.

M Predicting gain from occurrence history: method

The groups in this analysis record where earlier occurrences lie relative to the 64-token window. They differ from the prior-count groups used to establish the Simpson reversal.

Method. We now measure the benefit of access to tokens more than 64 positions in the past. We evaluate each target token twice: with the full prefix and with only the last $W = 64$ tokens. The loss difference $L_W - L_{\text{full}}$ is positive when distant context improves prediction. We group targets by whether they have earlier occurrences inside the window, outside it, both, or neither. We call tokens seen only outside the window *far repeats*. For each class we fit a linear model of this gain on features computed from the ids only: log corpus frequency, log distance to the last occurrence, log number of earlier occurrences beyond the window, bigram copyability, number of occurrences inside the window, the document's type–token ratio (distinct types divided by tokens so far) and $\log t$. Far repeats are 17–25% of tokens and carry 58–73% of the gain, at one to two nats each (Appendix H).

N Detailed domain comparisons and synthetic intervention

Across domains. The decrease observed on FineWeb does not hold on every text domain. Over eight Pile domains (Pythia-1.4B) the curve rises on FreeLaw ($1.18 \to 1.62$ from $t = 64$ to 1023), is flat on Wikipedia and Github, and falls steeply on StackExchange ($2.32 \to 1.14$). Figure 6 decomposes each domain (exact values in Appendix D, Table 6). Composition is -0.19 to -0.41 nats per e-fold everywhere, a narrower range than the variation in the net slope. What differs is the within-group increase, from $+0.04$ on StackExchange, where repeats in a question-and-answer thread do not get harder, to $+0.39$ on FreeLaw, where low-loss early legal boilerplate is followed by higher-loss tokens within the same count groups, and first-occurrence loss exceeds repeat loss

by only 1.7 nats. Whether the curve falls in a domain is whether the changing shares outweigh the within-group increase there.

Predicting the direction of the curve. We combine early group losses with the domain's later token proportions: take each group's loss at $t = 64$ and the token shares of the domain, compute the composition rate they imply over $t \in [64, 1023]$, and add the FineWeb dilution rate of $+0.30$. The predicted net slope has correlation 0.79 with the observed one over the nine corpora and the right sign in eight of them (Appendix D).

Across domains, Pythia-1.4B scores span -1.17 to $+0.45$. A two-way decomposition assigns 99.3% of score variance to domain, 0.2% to size and 0.5% to interaction. The two Pythia sizes correlate at 0.993 (mean absolute difference 0.073; maximum 0.12 nats); the larger model scores 0.04 higher on average. Across tokenizer families, Qwen2.5-1.5B correlates with Pythia at 0.98–0.99 and GPT-2-large at 0.83–0.88, with its main departure on FreeLaw (0.85 nats below 1.4B; Appendix D). Figure 6(d) explains the cache's limitation: it tracks rank ($r = 0.89$) but misses levels (MAE 0.36), while static bigram scores already vary from -0.04 to $+0.78$. These static position effects can already account for part of the improvement that a count-only baseline attributes to repetition.

Intervention. Correlations cannot separate the corpus from a model effect that happens to co-vary with it, so we change the corpus by hand. Across nine synthetic data distributions, changing the data changes the ICL score by 0.65 nats; increasing model size $5\times$ changes it by at most 0.09.

We generate documents over 2048 token types using a Pitman–Yor process (design in Appendix J). At each step it either repeats a token according to prior counts or draws a new type from a base distribution. The discount d controls new-type arrivals; the Zipf exponent s controls how uneven the base probabilities are. Each parameter setting defines one synthetic "world". The Bayes predictor uses the true next-token probabilities of that world and minimizes expected loss.

Five worlds vary $d \in \{0.3, 0.5, 0.6, 0.7, 0.9\}$ at $s = 1$; four vary $s \in \{0.6, 0.8, 1.2, 1.5\}$ at $d = 0.6$. The measured Heaps exponent ranges from 0.32 to 0.66, and the Bayes-optimal score from -0.44 to -1.11. Each world supplies 120k documents of 512 tokens (200k for $d = 0.6$); the 4M- and 21M-parameter transformers each train for 6000 steps. Here the score is $L(400) - L(50)$, since documents have only 512 tokens.

The novel-share shift from positions 50 to 400 times the difference between novel-token and repeat-token loss predicts the Bayes score with $r = 0.98$ across the nine worlds. The Heaps exponent alone does not ($r = -0.77$): raising s lowers the exponent but widens the gap. The trained models' scores span 0.65 nats and correlate at 0.99; increasing size $5\times$ changes the score by at most 0.09, with a nearly constant loss offset (position s.d. ≤ 0.03; Appendix G). At this 10^8-token budget both models still have higher loss than the Bayes predictor, particularly at later positions and with more uniform base probabilities, and their scores correlate with Bayes only at 0.65. The intervention establishes a data effect at this scale, not size-independence at every scale.