THE RULER AGREES WITH ITSELF, NOT WITH THE TASK: AUDITING HELD-OUT LOSS AS AN INSTRUMENT FOR CONTEXT SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

When a language model's context is too long to keep whole, a selection rule decides which tokens stay. Such rules are often compared by held-out loss, which is cheap, label-free and precise. We ask whether the ordering it produces is the ordering a task produces. It is not, and where the two differ the loss is confident. We score twenty-one selection rules we constructed, each keeping half of a context's tokens, applied as token dropping on six frozen decoders and read on three loss corpora and four task benchmarks. Every ruler, a corpus or benchmark read with its own metric, reproduces its own ordering across halves of its data at a mean rank correlation of 0.956 to 0.996 corrected to full length; the loss corpora agree with one another at 0.83 on average, and a loss corpus with a benchmark at 0.44. Two instruments this reliable and this far apart measure different things. The disagreement sits at the top of the loss ranking: the loss prefers rules that keep recent or heavily attended tokens; the CBT benchmarks usually put a rule that keeps rare tokens first, LAMBADA usually second, and SQuAD's usual leader keeps the earliest tokens. It is not a matter of resolution: of the rule pairs that the WikiText-103 loss and a benchmark both order confidently, the benchmark reverses 30–44% (16–36% under the other corpora). Training a small selector on a frozen decoder against the loss does not remove the disagreement: on most task readings the selector stays in the rank band the loss gives it, but first place is decided by the task, and on CBT a rule the loss ranks in its bottom three usually beats the selector by point estimate. The obvious repair, a key-token gate that restricts the loss to tokens needing more than a short recent window, on average matches or modestly beats the plain loss only when that window sits well below the budget; tied to the budget, it measures the budget. We argue not for abandoning the loss but for auditing it before it chooses: the proxy's reliability, the target's, then their correlation read beside both.

1 INTRODUCTION

Held-out loss is a common comparator for ways of shortening a context: it is cheap and needs no labels, and several published methods were validated on it (§2). It is also precise, and precision invites the next step, reading its ordering as a task's; Radar (Hao et al., 2025, Figure 5) and RECOMP (Xu et al., 2024, Table 1) rank selection methods by it on a frozen model without asking whether the two orderings agree. This paper asks, and finds that they do not: the loss and the tasks order the rules differently, and where they differ the loss is confident.

We score twenty-one controlled selection rules, each keeping half of a context of up to 128 tokens, on six frozen decoders, and read every rule on three loss corpora and four task benchmarks. A *ruler* is an evaluation set read with its own metric; an *instrument* is a kind of ruler, loss or task (§3.3). The use we audit is specific: rank the rules by loss, then deploy the winner on a benchmark.

The disagreement is not noise. Every ruler reproduces its own ordering of the twenty-one rules at **0.956 to 0.996** on average across halves of its data, corrected to full length; the three loss corpora

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.

(encyclopedia articles, pre-1919 books, contemporary web pages) agree at **0.83**; loss against task, **0.44**; and within-kind agreement exceeds crossing agreement on every one of the six decoders (§4). Two instruments this reliable, agreeing this little, measure different things. Transfer varies twofold with the loss corpus, and keeping one eighth of a longer window narrows the WikiText–CBT disagreement sharply (§4, §8).

The wrong part is the top, the part read when a rule is chosen: the loss prefers rules that keep recent or heavily attended tokens; the two CBT benchmarks put a rule that keeps rare tokens first or second, LAMBADA second or third, and SQuAD's usual leader keeps the earliest tokens (§4, C.1). `pos-edges` keeps the attention sink and the most recent positions; `lex-rare` keeps the rarest tokens. On Qwen3-8B, WikiText-103 loss ranks `pos-edges` 5th of 21 and `lex-rare` last, 0.0882 nat apart [0.0821, 0.0943]. LAMBADA accuracy ranks them the other way, 39.80% against 58.12%, 18.32 points [16.71, 19.89], where a random subset scores 38.07%. The loss did not miss the difference; it put the wrong rule ahead.

More data would not shrink this. Of 1260 rule pairs on six decoders, WikiText-103 loss orders 1176 at 95% confidence; pairs that a task ruler also orders at 95% the other way are *confident reversals*, and they are 30% (LAMBADA) to 44% (SQuAD) of the pairs both rulers order with confidence. Of all the pairs the loss orders, more than half are reversed by at least one task ruler and one in nine by all four (§5); the failure is direction, not precision.

Nor does training against the loss repair it. If the misordering belonged to the rules we wrote, a selector the loss builds for itself should escape it. It does not: trained end to end through a hard top-k, it reaches the loss's top nine of twenty-four arms (the twenty-one rules and the selector's three seeds) and mostly stays there on the tasks, but first place is the task's to decide, and on CBT a rule the loss ranks in its bottom three usually beats the selector by point estimate (§6). The obvious repair, a key-token gate, weights the loss toward tokens needing more than a short recent window. Repurposed to rank rules, it agrees with the tasks better on average than the plain loss only when that window sits well below the rule's budget; tied to the budget, it measures the budget (§7).

What carries beyond context selection is not the verdict but the check behind it: estimate both reliabilities before reading the correlation (§9). Measurement theory has been brought to bear on fairness (Jacobs & Wallach, 2021), benchmarks (Raji et al., 2021), bias measures (van der Wal et al., 2024), model-based judges (Norman et al., 2026) and comparisons between systems (Chouldechova et al., 2025), but not on a cheap proxy that ranks methods on a frozen model with both reliabilities measured rather than assumed.

We make three contributions; §9's procedure follows from the first, G.1's checklist from all three.

1. A reliability-and-validity audit of a comparator used to choose among methods, and of its obvious repair (§4, §7).
2. A measurement of what the disagreement costs, the confident reversals, with paired intervals on both sides (§5, C.2).
3. The audit repeated on a selector trained end to end on the loss and checked against a yoked noise-gradient twin (§6, Appendix D).

We rank no published KV-cache method (§3.2) and recommend no rule for deployment (§3.1, §8).

2 Background and position

The argument of this paper. Instruments that claim to measure the same construct must converge, and where they do not, at most one of them is measuring it (Campbell & Fiske, 1959); validity belongs to an interpretation and a use rather than to the instrument itself (Kane, 2013; Messick, 1995), which is what lets a loss be a good instrument for monitoring training and a poor one for choosing among methods. Information retrieval has measured how often a test collection reverses the ranking of two systems (Buckley & Voorhees, 2000; Urbano et al., 2013), the closest precedent for the quantity we estimate. None of that work estimates both reliabilities and reads the corrected correlation as a validity argument about a specific use; for the use audited here, each benchmark is the criterion by definition.

Where the rules come from, and who ranked them by loss. The intervention itself has direct precedent. Khandelwal et al. (2018) and Sun et al. (2021) drop context tokens from what a frozen

language model conditions on, compare against a matched random-drop control, and rank the result on held-out loss alone; O'Connor & Andreas (2021) run the same deletion, but retrain a model per ablation and rank on held-out likelihood against full- and no-information references. A second line, cache eviction, likewise settled its design choices on perplexity: StreamingLLM its attention-sink count (Xiao et al., 2024, Table 2), TOVA its head-versus-layer selection (Oren et al., 2024, Table 3), Devoto et al. (2024, Figures 3–4) their choice of low-L2 keys over random and inverse-L2 eviction (and on retrieval), as Grave et al. (2017) and Dai et al. (2019) had before them. A third line, also eviction, rests on task suites: Liu et al. (2023) report few-shot accuracy beside perplexity, and Zhang et al. (2023) and Li et al. (2024) settle their choices on accuracy. Five of our twenty-one rules are modelled on the scoring functions of the second and third lines (Zhang et al., 2023; Oren et al., 2024; Li et al., 2024; Xiao et al., 2024; Devoto et al., 2024) and the other sixteen fill out the axes of Table 1 (A.2); all are applied from one layer on as token dropping (pruning as in Rao et al. (2021), not fusing as in Liang et al. (2022)), so they are controls, not those methods (§3).

What looks like a contradiction, and why it is not. Across *models*, loss does track downstream ability: perplexity, pretraining loss and compression on external corpora all track benchmark performance closely (Gadre et al., 2025; Du et al., 2024; Huang et al., 2024), with tokenizer, data and architecture fixed in the first two and many benchmarks averaged in the first. Once shape or data varies, counterexamples appear: Tay et al. (2022, §4.1) exhibit a rank inversion across model shapes, and Ankner et al. (2025) a data-selection intervention that worsens held-out perplexity while improving downstream accuracy. One result from a setting close to ours points the other way: Gonen et al. (2023) select prompts for a frozen model by perplexity and find lower-perplexity prompts perform better. Two others support or use the audited practice: Xia et al. (2023) find that perplexity predicts in-context learning across training checkpoints with the prompt fixed, and Ram et al. (2023) rank retrieval mechanisms by perplexity without checking the ranking against a task. The distinction is the population: those results concern paraphrased prompts, retrieval mechanisms or models along a training trajectory, whereas ours is a family of rules that all keep the same fraction of the same context on one decoder, read per benchmark. Chen et al. (1998) found that the population matters for how well perplexity tracks word error rate. Lu et al. (2025) reaffirm perplexity as an indicator for context-*extension* methods across models of differing quality, where the two move together.

This subfield's evaluation has been examined before. Unaligned environments (Yuan et al., 2024), an incommensurable cost axis (Bhaskar et al., 2025), benchmark-dependent verdicts (Chen et al., 2026) and a random-eviction control that matches the scored evictors in long reasoning (Wang et al., 2026) have been reported for eviction (Appendix H); we ask not whether a method degrades on a benchmark but whether two instruments induce the *same ordering* over a designed family.

3 SETUP

3.1 THE RULE FAMILY

Twenty-one selection rules, all controls we constructed, are grouped by *what the score reads* rather than by provenance (Table 1); several groups contain a rule and its sign-reversed partner, so a claim about an axis can be checked by asking whether reversing the sign moves the rule across the ranking (the sign-reversed controls are `pos-first`, `key-norm-hi`, `hidden-norm-lo`, `value-norm-lo` and `lex-frequent`; A.2). Two reference points bracket every C.1 table: `random`, a uniformly drawn subset at the same budget, itself one of the ranked twenty-one, and `full`, the unmasked decoder, outside the family (B.2). Eighteen of the twenty-one read only their input; the other three, all lexical, carry one external statistic, a unigram table counted on WikiText-103's training split, or on PG-19's or FineWeb's for those rulers (A.2).

3.2 THE APPARATUS

The decoders. Six base models, none instruction-tuned: Qwen3-0.6B, Qwen3-1.7B and Qwen3-8B, Llama-3.2-1B, OLMo-2-1B and SmolLM2-1.7B. The set spans four families (so a result is not a property of one pretraining recipe) and one family at three sizes with tokenizer and data held fixed (mostly width; only the 8B is deeper). The families differ in grouped-query attention and QK-norm, so the key-norm score reads a QK-normalised key tensor in two of the four (A.1).

Table 1: The rule family. Twenty-one rules grouped by the signal the score reads: seven signals and one no-signal control (value geometry is listed but not claimed as an axis; §8), each defined in A.2; † marks the five modelled on published scoring functions.

Axis (what the score reads)	Rules
Position	`pos-last`, `pos-first`, `pos-edges`†, `pos-uniform`
Attention	`attn-all`, `attn-last8`, `attn-last1`†, `attn-distant`
Attention and position	`mix-attn-last`†, `mix-attn32-last`†
Key geometry	`key-norm-lo`†, `key-norm-hi`
Hidden geometry	`hidden-norm-hi`, `hidden-norm-lo`
Value geometry	`value-norm-lo`, `value-norm-hi`
Lexical frequency	`lex-rare`, `lex-frequent`, `lex-nonstop`, `lex-recurring`
None	`random`

The intervention. No decoder weight is updated in the body (§6 trains a scorer on top of the frozen decoder; Appendix E relaxes the freeze). Each loss window is $T = 256$ tokens: the first $C = 128$ are the context that selection acts on, the last 128 the continuation on which loss is measured. From the gate layer L, a third of the way up the stack, selection is an exact top-k over the C context positions with $k = 64$, half the context, on every window that fills C (C.3 varies k and C); the budget is the fraction $k/C = 0.5$ of each row's real context, so shorter rows keep fewer positions, as nearly every LAMBADA passage (median 72 tokens) and over half of SQuAD questions do (B.1). Every rule keeps the first context position, charged against the budget, so all rules spend the same budget (A.3). The mask zeroes post-softmax attention weights and renormalises, which is equivalent to removing the masked-out keys from the softmax denominator, and every run verifies before scoring that a mask of all ones is the identity (largest logit difference from the unmasked decoder 4.6×10^{-5} to 8.5×10^{-4} across runs; A.4). The mask applies to all query rows from layer L onward, so this is token dropping and not post-prefill eviction; the claims concern disagreement between instruments over a fixed rule family and need only that the intervention be identical across rules and rulers.

3.3 The rulers, each read with its own metric

Six evaluation sets (B.1) yield three *loss rulers* (WikiText-103, PG-19 and FineWeb, read as nats per token over 1024 held-out windows) and four *task rulers* (LAMBADA, CBT-NE, CBT-CN and SQuAD, read as accuracy); CBT counts as two; loss and task are the paper's two *instruments*. Every task ruler is read with its benchmark's metric (SQuAD: stricter exact match; B.1), because reading it by loss would change which rule ranks first: `lex-rare` ranks first on five of six decoders under LAMBADA's gold-continuation loss but second on five of the six under LAMBADA accuracy, and first on five of six under SQuAD loss but from first to ninth under exact match. §4 reports the within-benchmark agreement of the two readings.

Task filters and headroom. CBT and SQuAD are evaluated on the well-formed examples whose answer survives truncation to the last C tokens, since no rule can recover what is absent: 672 of 1000 on CBT-NE and 567 on CBT-CN on five decoders (B.1). A benchmark must also have headroom; with no mask, exact match on SQuAD runs from 18% to 60% across the decoders (B.2).

3.4 Two kinds of uncertainty

Rule comparisons are deterministic given a decoder and a ruler; uncertainty comes from resampling the evaluation units with a paired bootstrap at $B = 10\,000$ unless noted. Only the trained arms (§6, Appendix E) carry a second source, rerun-to-rerun variation: over ten repeats of one scorer configuration, SD 0.008 nat and range 0.024 nat (Appendix D); the two are not combined.

4 A reliable instrument that does not transfer

Each ruler reproduces its own ordering of the rules almost exactly, so the disagreement between rulers is not sampling error. Ranking the rules on each half of a ruler's evaluation units and correlating the two rankings, across 200 random splits and corrected to full length by Spearman–Brown (Spearman, 1910; Brown, 1910), puts all seven rulers between **0.956 and 0.996** averaged over de-

coders, WikiText-103 highest and CBT-CN lowest, with no decoder–ruler cell below 0.937 (B.3). That forecloses the reading that the rulers disagree because one measures unreliably; correcting the crossing correlation for both reliabilities moves it only from 0.439 to 0.446.

The reliability also sharpens what the loss corpora are. WikiText-103 agrees with *itself* at 0.996, with PG-19 at 0.81 and with FineWeb at 0.75 (Figure 1); the 0.996 shows those gaps are not sampling error. "Held-out loss" is not one ruler with imperfect reliability; it is a family of nearly perfect rulers that disagree about the answer. PG-19 and FineWeb agree at 0.93, above either with WikiText-103, a gap due mostly to the three unigram-table rules (without them all three pairs agree at 0.90 to 0.94).

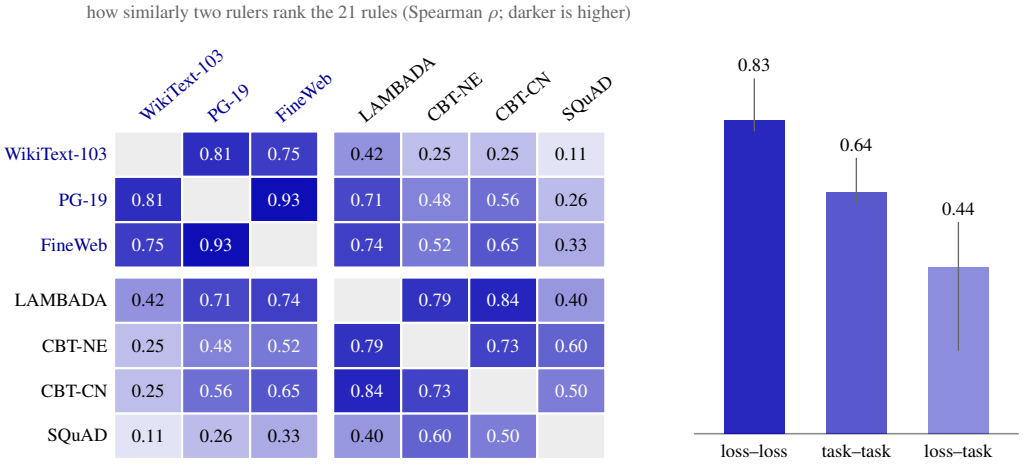

Figure 1: **Left: how similarly each pair of rulers ranks the rules.** Spearman ρ between the rule orderings of two rulers (twenty-one rules, average ranks for ties), averaged over the six decoders; darker is higher, loss rulers are named in blue, and each row compares one ruler with all the others. A ruler's agreement with itself is its split-half reliability (§4, B.3) and is not shown. **Right: the same agreement within and across the kinds.** Corpus to corpus, benchmark to benchmark and loss to task are shaded as on the left; the whiskers span the six decoders, and within-kind agreement exceeds crossing agreement on every one of them (B.3). Underlying scores in C.1.

The matrix is read in three layers. The first is not in it: holding the benchmark fixed and changing only the *metric* keeps most of the ordering (LAMBADA's two readings agree at +0.91, CBT's at +0.97, SQuAD's at +0.86), which licenses treating either reading as a measurement of the same thing; what separates loss from task, then, is mostly the text scored, not the metric read from it. The other two are in it: within kind, agreement corpus to corpus inside the loss instrument is +0.83 and benchmark to benchmark inside the task instrument +0.64 (+0.73 to +0.84 among the three cloze benchmarks, +0.40 to +0.60 for SQuAD against each); crossing from loss to task, +0.44. Read as a one-trait (rule quality), two-method fragment of a Campbell & Fiske (1959) multitrait-multimethod table, the matrix suggests method variance: on every decoder the within-kind blocks exceed the crossing block, and each ruler's nearest neighbour is of its own kind in 41 of 42 decoder–ruler cases (B.3). No loss ruler predicts any benchmark above +0.74 on average, and SQuAD sits apart from every ruler, so the task side is not one construct either; each benchmark is read as its own criterion. If anything, the five sign-reversed controls and `random` inflate the crossing block: without them it falls to +0.12 while the loss block is +0.82 (C.1).

The corpus decides how much the ranking is worth. Three corpora that agree with *each other* at +0.83 differ by a factor of two in how much their ranking says about task behaviour, so the choice of corpus is not a detail of the setup. All three agree on the *ordering* of the four task rulers on average (LAMBADA best predicted by loss and SQuAD worst, the two CBT subsets between) while disagreeing on the level throughout. PG-19 (books) and FineWeb (web pages) transfer more than WikiText-103 (encyclopedia; B.3). The difference is carried by the three rules that read a unigram table, which PG-19 and FineWeb count from their own text (A.2): without them, all three corpora transfer alike (+0.51 to +0.53; C.1). Even with them, FineWeb reaches only +0.33 against SQuAD, so "choose the matching corpus" is advice one can follow only after deciding which task matters. Having decided, one can evaluate on that task directly, without the proxy. The best cell, +0.74 for FineWeb against LAMBADA (B.3), is short of the +0.91 at which LAMBADA's two readings agree.

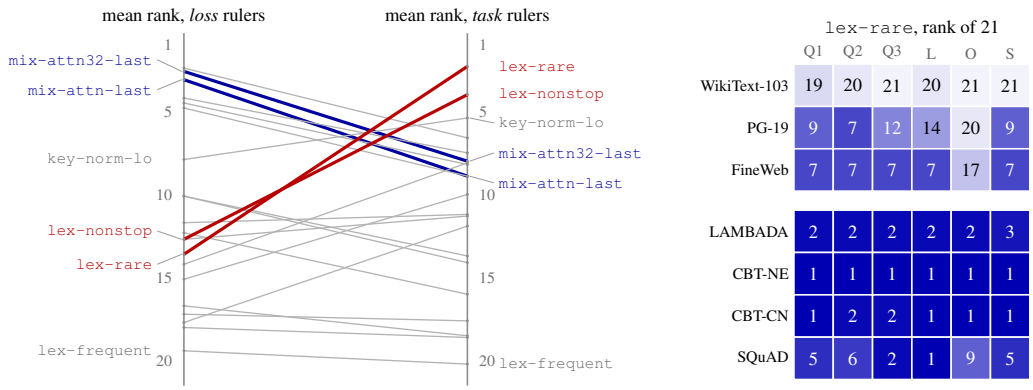

Figure 2: **Left: what the disagreement is made of.** Every rule's mean rank under the three loss rulers (left axis) against its mean rank under the four task rulers (right axis), over the six decoders; rank 1 is best, and a line that crosses many others is a rule the two kinds place far apart. **Right: the rule that moves most, on every decoder.** `lex-rare`'s rank out of 21 on each ruler and decoder, shaded as in Figure 1 (darker is a better rank): 19th to 21st on WikiText-103, 7th to 14th on PG-19 and FineWeb on five decoders (20th and 17th on OLMo-2-1B), first or second on the two CBT benchmarks, and second on LAMBADA on five decoders (third on SmolLM2-1.7B). Decoders: Q1 Qwen3-0.6B, Q2 Qwen3-1.7B, Q3 Qwen3-8B, L Llama-3.2-1B, O OLMo-2-1B, S SmolLM2-1.7B. Point estimates, no intervals; all values in C.1.

What the disagreement is made of. Figure 2 plots every rule's mean rank under the two kinds, and the disagreement is not spread evenly. Most rules move a few places. Two move far in one direction: `lex-rare` from 13.6 under the loss rulers to 2.3 under the task rulers, and `lex-nonstop`, a thresholded form of it, from 12.7 to 4.0. The two rules modelled on mixed attention-and-position methods move the other way, from 2nd and 3rd by loss to around 8th on the tasks. Six rules lead the loss column at mean ranks 2.4 to 4.8, all using recency, attention mass or a mixture, and all six fall to 6.6–8.9 on the tasks, while the task column is led by `lex-rare`, `lex-nonstop` and `key-norm-lo` at 2.3 to 5.4. No rule is in the top three of both: the loss rewards keeping what was recent and heavily attended, these benchmarks reward keeping what was rare, and `key-norm-lo`, at 7.9 and 5.4, is the only rule near the top of the loss column that rises rather than falls.

This is not an inversion, and the ranks are what rule that out. If the finding were "whatever the loss ranks worst, the tasks rank best", the sign-reversed partners at the bottom of the loss column would lead the task column. They do not: `lex-frequent` is 19.4 by loss and 20.2 on the tasks, `key-norm-hi` 18.0 and 18.6, `hidden-norm-lo` 16.7 and 18.5, `value-norm-lo` 17.2 and 17.6. Four of the five reversed partners are poor under *both* instruments: for them, reversing the sign of a signal does not improve task performance; it loses the signal. What the loss misranks is a specific thing, rarity against recency, and not the ordering in general. And it is the top of the loss ranking that is wrong, the part read when a rule is chosen.

5 What the instrument's disagreement costs

The loss is precise, and its precision makes the disagreement costly. A low correlation would be benign if the loss were merely coarse. Over 1024 windows the standard error of a loss difference is typically a few thousandths of a nat, and across all $\binom{21}{2} = 210$ rule pairs on each of the six decoders, the WikiText-103 loss orders **1176 of 1260** pairs at 95% confidence; Figure 3 plots every pair. A pair that two rulers both order at 95% in opposite directions is a *confident reversal*. Of the pairs the loss orders (1177 in C.2's bootstrap draw), LAMBADA also orders 1090 at 95% and puts **331** of them the other way; CBT-NE 362 of 936, CBT-CN 283 of 845, SQuAD 412 of 944: 30 to 44% of the pairs both rulers order confidently; under PG-19 and FineWeb, the corpora that transfer better, 19 to 36% and 16 to 34% (C.2). Pooled over rulers, 617 of 1177 are reversed by at least one task ruler and 126 by all four; 320 are concordant with all four (C.2). Nor is this a matter of narrow margins: among pairs the loss separates by 0.03 nat or more, a task ruler's point estimate agrees with its direction only 56–72% of the time.

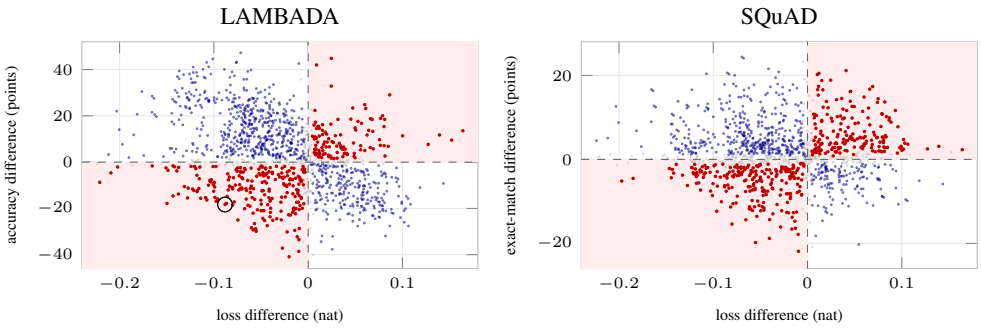

Figure 3: **Where the loss is confident and wrong.** Every point is one rule pair on one decoder, 1260 per panel: the WikiText-103 loss difference against the accuracy difference, oriented so that the shaded quadrants hold the pairs the two rulers order in opposite directions. Red: both rulers confident at 95% and disagreeing (331 on LAMBADA, 412 on SQuAD); blue: both confident and agreeing (759 and 532); grey: at least one ruler uncertain. The circled pair is the opening example; the vertical scales differ. C.2 repeats the count under PG-19 and FineWeb.

One pair, in full. Table 3 (C.2) sets out the opening pair, the widest such reversal under LAMBADA in which the loss-preferred rule follows a published scoring function: on Qwen3-8B the WikiText-103 loss puts `pos-edges` 5th of 21 and `lex-rare` last, 0.0882 nat apart [0.0821, 0.0943], and LAMBADA puts them the other way round by 18.32 points [16.71, 19.89]. The failure is not precision but direction. Deploying the WikiText-103 loss's first-ranked rule instead of the task's own costs on average 9.8 points on LAMBADA, 12.0 on CBT-NE, 2.9 on CBT-CN and 5.0 on SQuAD, confident in 20 of 24 cells (C.2).

6 Optimising against the instrument

§4 and §5 leave one escape open: if the misordering belongs to the twenty-one rules we wrote, an arm the loss builds for itself should be free of it. So we let the loss put an arm of its own near the top of its ranking. On a frozen decoder a small scorer produces one score per context position; the scorer is trained through the hard top-k selection (the gate) with a straight-through estimator (Bengio et al., 2013), so the forward pass uses the exact discrete selection the evaluation uses; training minimises WikiText loss over 8000 windows, the checkpoint has the lowest WikiText validation loss over three learning rates and 1000 steps, and we use three seeds per decoder (Appendix D). Ranks below are among **24 arms**, the twenty-one rules plus the three seeds, on all six decoders.

The optimisation is real, and modest. On the WikiText loss it minimised, the scorer lands at **rank 1–9 of 24** on every decoder (5th–7th on Qwen3-8B; Figure 4, left), but it beats the best hand-written rule only on Qwen3-0.6B, by 0.004 nat (half the rerun SD), matches it on Llama-3.2-1B and trails it by 0.006–0.023 nat on the other four (Appendix D). A yoked twin is the same loop, seed and learning rate with the gate's gradient replaced by same-scale noise. The trained arm beats it in 47 of 54 seed-and-rate pairs by paired 95% intervals (4 ties, 3 losses; 27–2–3 where the twin itself improved): every pair on four decoders, 7–1–1 on Qwen3-8B and 4–3–2 on SmolLM2-1.7B (Appendix D). Under this recipe the loss-trained arm sits where the loss's best rules already are, so the question of §4–§5, how the loss orders the rules at the top, is the question the scorer now faces.

The misranking survives optimisation. On the task rulers (Figure 4, right; the shaded bands are where the loss puts the scorer and `lex-rare`), the loss's verdict on its own arm survives as a band and not as a rank. The seeds stay inside the loss's top nine in 62 of the 72 decoder–ruler–seed cells; where they stand within it is the task's to decide: first on SQuAD exact match on four decoders of six, first on one decoder for each CBT subset (tied for first on a second CBT-NE decoder), nowhere on Qwen3-8B, and never on LAMBADA, where `attn-last1` leads on five decoders and `attn-last8` on the sixth (Table 6). The loss's verdict on the other side does not survive at all: `lex-rare`, which the loss ranks 22nd, 23rd, 23rd, 24th, 24th and 24th of 24 (bottom three on every decoder, last on three), is first on CBT-NE on five decoders of six and on CBT-CN on three, at or above the scorer, and the metric is not the reason (within a benchmark the two readings agree at $\rho = 0.86$–0.97; §4). A paired bootstrap over examples (Appendix D) keeps the band and softens the

leads: a seed is in the top nine with probability at least 0.95 in 56 of the 72 cells, the scorer leads confidently on two SQuAD decoders only, and `lex-rare` beats all three seeds confidently on two CBT-NE and three CBT-CN decoders. The pair of §1 recurs on CBT with the loss-trained arm in place of `pos-edges`: the loss the arm was trained on puts the usual winner in its bottom three.



Figure 4: **The misranking survives optimisation.** For each ruler and decoder, the dots are the ranks of the three trained seeds among the 24 arms and the red cross is `lex-rare`; the shaded bands are the ranks the loss assigns them (grey 1–9, red 22–24). On the task rulers the dots mostly stay in their band (62 of 72) and the crosses leave theirs on every decoder and every task ruler; what the loss gets wrong is first place and where `lex-rare` stands. Decoders abbreviated as in Figure 2. Point ranks, ties broken by arm order; Appendix D adds bootstrap intervals and counts the leads.

The corpus decides what transfers, and the task's own loss is not always better. The same recipe trained on FineWeb instead of WikiText, on Qwen3-0.6B, optimises its own loss less well (9th–11th of 24 on FineWeb) and leads on no ruler: 8th–10th on SQuAD, where the WikiText-trained seeds rank 1st–3rd (Appendix D); what transfers from a loss to a task depends on the corpus, as §4 found for the three rules that read a unigram table. Trained against the task's own loss instead, on SQuAD's training split and on one half of LAMBADA, read on the other, the same recipe on Qwen3-0.6B is 1st–3rd among the twenty-one rules and its own seeds on SQuAD, 1.41 points above the WikiText-trained scorer (medians 17.55% and 16.14%), and 4th–6th on LAMBADA, 2.06 points below it (38.92% and 40.98%); on Llama-3.2-1B the same pattern, +1.04 on SQuAD and -4.58 on LAMBADA (Appendix D). Where LAMBADA's optimum lies is open.

What this shows. A selector trained against the loss is in the loss's top nine on most task readings (62 of 72 cells), but which arm is first, and whether the arm the loss ranks in its bottom three is in fact the best, are answered only by the task. The ordering problem of §4–§5 survives optimisation: optimising the proxy does not repair it, and a number obtained by optimising the proxy is a claim about the proxy, not about task behaviour.

7 The obvious repair helps only below the budget

Mean loss averages over tokens whose prediction never needed the context a rule dropped, and the published response, LongPPL, weights it toward the tokens that do (Fang et al., 2025): a token's need is how much the full context improves its log probability over a short recent window, and LongPPL keeps the tokens whose need clears a fixed threshold. Our key-token gate, whose short window is the sink and the last $K - 1$ context positions, keeps the top decile of need, computed by the decoder under test (Appendix F lists the differences). Used to rank *selection rules*, the gated loss helps only when its window sits well below the budget: at $K = 16$, one eighth of the context (LongPPL's ratio on its longest documents), it agrees with the task rulers slightly more than the plain WikiText-103 loss does (mean ρ over the four crossing cells +0.34 against +0.26, a gain of +0.08 [+0.04, +0.09] by a paired bootstrap); at $K = 32$ it agrees less (+0.17) and at $K = k = 64$ hardly at all (+0.06). LongPPL's second filter, on full-context log probability, removes the $K = 16$ gain, not the deficits (Appendix F).

The cause is structural: K and k are the same kind of quantity (how many tokens are made available) on opposite sides of the measurement, K belonging to the gate and k to the rule under test. Over six decoders and twenty-one rules the rank of the last-k rule (`pos-last`) tracks the **sign of** $K - k$,

at best 17th of 21 whenever $K \geq k$ (20–21 of 21 at $K = k = 64$, Table 7) and as high as second (2–17 of 21) once $K < k$, with its sign-reversed partner `pos-first` moving the opposite way at $k = 64$ (Appendix F). Tied to the budget, the gate measures the budget. It can therefore compare position-based rules only with K fixed independently of k, well below it, and both reported; with K at LongPPL's ratio of one eighth of the context, any budget of one eighth or less puts $K \geq k$. We make no claim about ranking models, the gate's original use, where no k confounds K.

8 LIMITATIONS

The apparatus. At $C = 128$ with no decoder above 8B, we do not claim that any magnitude here transfers to the multi-thousand-token regime the published methods target. The claims are about instrument agreement, not operating points. C.3 sweeps the budget and the gate depth on two decoders and the window to $C = 512$ on all six: at the same keep fraction the disagreement persists (WikiText–CBT agreement +0.34 to +0.40 on average, against +0.25 at $C = 128$), and it narrows sharply when the body's absolute budget, now one eighth of the context, is kept at the longer window: +0.71 and +0.67 on average, where the two instruments rank `lex-rare` far closer together (C.3). The mask is shared across heads, so three of the five published-derived rules depart from their per-head form (A.5), and the intervention is token dropping, not post-prefill eviction (§3.2). The decoder is frozen, as in practice; Appendix E relaxes that on three decoders and four rules: adaptation lifts `lex-rare` level with `pos-last` on the loss, the task ordering unmeasured.

The rule family. We do not propose the lexical rules as a method, and the unigram table is only a probe; selecting context by a cheap statistic computed outside the model is already established (Jiang et al., 2023; Sood et al., 2026). Three rules are not independent probes and a fourth nearly duplicates another (A.2, C.1), so the four-rule lexical, attention and position groups are effectively smaller. The value-norm sign pair reverses inconsistently across decoders, so we report those two rules but claim no value-geometry axis. The tables are bracketed by a random floor and a full-context ceiling, not by a best-possible subset, so distance from the ceiling is not distance from the best.

The rulers. The four benchmarks agree with each other at +0.40 to +0.84, so they are not pooled as one construct; each is the criterion for its own use. SQuAD has the narrowest band of rule accuracies on average (B.2). CBT is read on two of its four configurations, and within those on the well-formed 56–67% of examples whose answer survives the window (§3.3); the other two are excluded because their split-half reliability falls short (0.84 and 0.31 against the 0.956 to 0.993 of the four task rulers used; B.1). All four ask the decoder to locate one word or span; none needs evidence assembled from several documents, which is plausibly where rules would differ most, and at $C = 128$ a multi-hop context does not fit, so the disagreement is established for local evidence, not for aggregation.

9 AUDITING ANY PROXY

Nothing here argues for abandoning held-out loss. A proxy used to choose among methods should be audited before it is trusted: any setting where a cheap quantity selects among expensive options admits the same three measurements, from per-unit scores of the proxy and the target.

The procedure has three steps. **First**, estimate the reliability of the proxy with its free choices fixed: split its evaluation units in half, rank the candidates on each half, correlate the two rankings, correct to full length. **Second**, repeat the split-half estimate for the target. **Third**, and only then, read the correlation between them, beside both reliabilities: if both are high and the correlation is not, the disagreement is real and the proxy measures something else; if either is low, the correlation is uninterpretable and the unreliable measurement must be fixed first.

For the rulers here, corpus fixed and averaged over decoders, those three numbers are 0.996 (WikiText-103), 0.993 (LAMBADA, B.3) and 0.42 for that pair; the mean over the twelve loss–task pairs is 0.44 (Figure 1). The procedure is standard in psychometrics and would run unchanged on scores already computed when selecting pretraining data by perplexity or checkpoints by validation loss; G.1 lists what to report.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation (drafts written in the authors' first language were translated into English, and the English was then edited for readability), to implement methods (the experimental harness and the analysis scripts were written with AI assistance), to design or give feedback on the experiments, to carry out the analyses, and to interpret results (the robustness checks were developed with an AI assistant, and readings of the results were discussed with it; the interpretations and the claims in the paper are the authors' own judgement, formed in that discussion and not taken from the assistant). We have not used generative AI tools to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to supply critical ingredients for proofs or to write proofs, to propose or refine hypotheses, to clean or reformat datasets, or to support qualitative or thematic data analysis. Additionally, we used generative AI tools to draft and edit parts of the text, to write the plotting code for the figures (every figure is drawn by code from the stored outputs; no image was generated), to identify and summarise related literature, and to format references. We have reviewed all AI-assisted work: every script was read and checked by the authors before it was run, every number the paper computes was checked by the authors against the stored outputs it was computed from, every reference was checked against its published record, and every claim in the paper was formulated by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Every results table and figure in the paper is produced by a script from stored per-unit outputs. Appendix A gives the apparatus (the six decoders and their gate layers, the twenty-one rule definitions, the protected position and the identity check run before any rule is scored), Appendix B the evaluation sets, their filters and the split-half reliabilities (the procedure is defined in §4), Appendix D the scorer's training recipe, seeds, checkpoint selection and the yoked-twin construction, and Appendix F the key-token gate. The random floor is fixed per passage and `lex-nonstop`'s pseudo-random tie-break is deterministic too, so the twenty-one-rule tables are reproducible across runs. The code, the rule definitions and the stored per-unit outputs behind every table and figure will be released upon acceptance.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L. Leavitt, and Mansheej Paul. Perplexed by perplexity: Perplexity-based data pruning with small reference models. In *International Conference on Learning Representations (ICLR)*, 2025.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Adithya Bhaskar, Alexander Wettig, Tianyu Gao, Yihe Dong, and Danqi Chen. Cache me if you can: How many KVs do you need for effective long-context LMs? *arXiv preprint arXiv:2506.17121*, 2025.

William Brown. Some experimental results in the correlation of mental abilities. *British Journal of Psychology*, 3(3):296–322, 1910. doi: 10.1111/j.2044-8295.1910.tb00207.x.

Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 33–40. ACM, 2000. doi: 10.1145/345508.345543.

Donald T. Campbell and Donald W. Fiske. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin*, 56(2):81–105, 1959. doi: 10.1037/h0046016.

Alex Chen, Renato Geh, Aditya Grover, Guy Van den Broeck, and Daniel Mingyi Israel. The pitfalls of KV cache compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 41530–41553. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1926.

Stanley F. Chen, Douglas Beeferman, and Ronald Rosenfeld. Evaluation metrics for language models. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 275–280, 1998.

Alex Chouldechova, A. Feder Cooper, Solon Barocas, Abhinav Palia, Dan Vann, and Hanna Wallach. Comparison requires valid measurement: Rethinking attack success rate comparisons in AI red teaming. In *Advances in Neural Information Processing Systems*, volume 38, pp. 54040–54065. Curran Associates, Inc., 2025.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1285.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective L_2 norm-based strategy for KV cache compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18476–18499. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1027.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. In *Advances in Neural Information Processing Systems*, volume 37, pp. 53138–53167. Curran Associates, Inc., 2024.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In *International Conference on Learning Representations (ICLR)*, 2025.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Jenia Jitsev, Luca Soldaini, Alexandros G. Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks. In *International Conference on Learning Representations (ICLR)*, 2025.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10136–10148. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.679.

Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *International Conference on Learning Representations (ICLR)*, 2017.

Yongchang Hao, Mengyao Zhai, Hossein Hajimirsadeghi, Sepidehsadat Hosseini, and Frederick Tung. Radar: Fast long-context decoding for any transformer. In *International Conference on Learning Representations (ICLR)*, 2025.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The Goldilocks principle: Reading children's books with explicit memory representations. In *International Conference on Learning Representations (ICLR)*, 2016.

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. Compression represents intelligence linearly. In *First Conference on Language Modeling (COLM)*, 2024.

Abigail Z. Jacobs and Hanna Wallach. Measurement and fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, pp. 375–385. Association for Computing Machinery, 2021. doi: 10.1145/3442188.3445901.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.825.

Michael T. Kane. Validating the interpretations and uses of test scores. *Journal of Educational Measurement*, 50(1):1–73, 2013. doi: 10.1111/jedm.12000.

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 (Volume 1: Long Papers)*, pp. 284–294. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1027.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970. Curran Associates, Inc., 2024.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EViT: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations (ICLR)*, 2022.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52342–52364. Curran Associates, Inc., 2023.

Yi Lu, Jing Nathan Yan, Songlin Yang, Justin T. Chiu, Siyu Ren, Fei Yuan, Wenting Zhao, Zhiyong Wu, and Alexander M. Rush. A controlled study on long context extension and generalization in LLMs. In *Second Conference on Language Modeling (COLM)*, 2025.

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

Samuel Messick. Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist*, 50 (9):741–749, 1995. doi: 10.1037/0003-066X.50.9.741.

Justin D. Norman, Michael U. Rivera, and D. Alex Hughes. Reliability without validity: A systematic, large-scale evaluation of LLM-as-a-judge models across agreement, consistency, and bias. *arXiv preprint arXiv:2606.19544*, 2026.

Joe O'Connor and Jacob Andreas. What context features can transformer language models use? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 851–864. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.70.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state RNNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18724–18741. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1043.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1144.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*, 2020.

Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. AI and the everything in the whole wide world benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1264.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023. doi: 10.1162/tacl_a_00605.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949. Curran Associates, Inc., 2021.

Aryan Sood, Shantanu Acharya, and Gaurav Kumar Nayak. Dropping the anchor: Statistical context summarization for distributed systems via Pulsar attention. *arXiv preprint arXiv:2607.20457*, 2026.

C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

C. Spearman. Correlation calculated from faulty data. *British Journal of Psychology*, 3(3):271–295, 1910. doi: 10.1111/j.2044-8295.1910.tb00206.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*, pp. 807–822. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.62.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Julián Urbano, Mónica Marrero, and Diego Martín. On the measurement of test collection reliability. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 393–402. ACM, 2013. doi: 10.1145/2484028.2484038.

Oskar van der Wal, Dominik Bachmann, Alina Leidinger, Leendert van Maanen, Willem Zuidema, and Katrin Schulz. Undesirable biases in NLP: Addressing challenges of measurement. *Journal of Artificial Intelligence Research*, 79:1–40, 2024. doi: 10.1613/jair.1.15195.

Heng Wang, Jielin Qiu, Wenting Zhao, Cheng Qian, Liangwei Yang, Jiawei Han, Heng Ji, Silvio Savarese, Shelby Heinecke, and Huan Wang. Random attention: Rethinking KV cache eviction for efficient reasoning. *arXiv preprint arXiv:2609.03430*, 2026.

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Ves Stoyanov. Training trajectories of language models across scales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13711–13738. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.767.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiayi Yuan, Hongyi Liu, Shaochen Zhong, Yu-Neng Chuang, Songchen Li, Guanchu Wang, Duy Le, Hongye Jin, Vipin Chaudhary, Zhaozhuo Xu, Zirui Liu, and Xia Hu. KV cache compression, but what must we give in return? A comprehensive benchmark of long context capable approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4623–4648. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.266.

Yike Zhang, Zhiyuan He, Huiqiang Jiang, Chengruidong Zhang, Yuqing Yang, Jianyong Wang, and Lili Qiu. LeanK: Learnable K cache channel pruning for efficient decoding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31122–31137. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1584.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34661–34710. Curran Associates, Inc., 2023.

A Apparatus details

A.1 The decoders

The table lists each decoder's layer count, the gate layer L at which selection begins, the model width d, and the attention and key-value head counts. L is a third of the way up each stack (§3.2).

family	model	layers	L	d	heads (KV heads)
Qwen3	Qwen3-0.6B-Base	28	9	1024	16 (8)
Qwen3	Qwen3-1.7B-Base	28	9	2048	16 (8)
Qwen3	Qwen3-8B-Base	36	12	4096	32 (8)
Llama-3.2	Llama-3.2-1B	16	5	2048	32 (8)
OLMo-2	OLMo-2-0425-1B	16	5	2048	16 (16)
SmolLM2	SmolLM2-1.7B	24	8	2048	32 (32)

The three Qwen3 sizes share a tokenizer and a pretraining corpus, which is what makes size the one axis the set can vary cleanly; the other three families vary the recipe instead. Two of the four families keep one key-value head per query head and the other two use grouped-query attention; two of the four apply QK-norm (per head in Qwen3, over the whole key projection in OLMo-2), so the key-norm score of A.2 reads a QK-normalised key tensor in two families and an unnormalised one in the other two.

A.2 The twenty-one rules

Each rule produces one score per context position; selection is the exact top-k of that score with the protected position forced in (A.3). Unless stated otherwise a score is read at layer L, a third of the way up the stack. By default $C = 128$ and the budget is $k/C = 0.5$ of each row's real context, which gives $k = 64$ on a full window (§3.2). Rules marked † are modelled on the scoring function of a published cache-management method, named in the right-hand column; they are applied here as token dropping from layer L and to every query row and are therefore not those methods.

Position. Scores that read only the index.

rule	keeps	modelled on
`pos-last`	the protected position plus the most recent $k-1$	—
`pos-first`	the earliest k positions	—
`pos-edges` †	the first 4 positions plus the most recent $k-4$	StreamingLLM (Xiao et al., 2024)
`pos-uniform`	k positions spaced evenly across the context	—

`pos-uniform` exists so that one positional rule keeps a set spread across the context. Without it every positional rule keeps a block at one edge of the context or at both, and "position" cannot be separated from "being at an edge".

Attention. Scores that read the attention the position receives at layer $L-1$. All are averaged over heads before the shared top-k (A.5).

rule	score	modelled on
`attn-all`	attention received, summed over every context query row	—
`attn-last8`	attention received from the last 8 query rows	—
`attn-last1` †	attention received from the last query row only	TOVA (Oren et al., 2024)
`attn-distant`	attention received from query rows more than 32 positions away	—

`attn-distant` exists because every other attention score is dominated by near-diagonal mass, which makes "was attended to" hard to distinguish from "is recent". Restricting the sum to distant query rows measures long-range attention specifically.

Attention and position. Rules that split the budget between the two; attention is read at layer $L-1$, as above.

rule	keeps	modelled on
`mix-attn-last` †	$k/2$ most recent by position, $k/2$ by accumulated attention; the protected position is charged to the attention half	H2O (Zhang et al., 2023)
`mix-attn32-last` †	attention from the last 32 query rows, max-pooled along the context axis with kernel 7 so kept positions form clusters, plus the most recent $\min(32, k-1)$ by position	SnapKV (Li et al., 2024)

Key, hidden and value geometry. Scores that read a vector norm rather than an attention weight. Key and value norms are computed per head at layer L and averaged over heads; the hidden-state norm is taken on the residual stream entering layer L.

rule	keeps	modelled on
`key-norm-lo` †	lowest L2 norm of the key vectors	K-Norm (Devoto et al., 2024)
`key-norm-hi`	highest L2 norm of the key vectors	—
`hidden-norm-hi`	highest L2 norm of the hidden state	—
`hidden-norm-lo`	lowest L2 norm of the hidden state	—
`value-norm-lo`	lowest L2 norm of the value vectors	—
`value-norm-hi`	highest L2 norm of the value vectors	—

Note the sign on `key-norm-lo`: the published method keeps *low* key norm, which is the opposite convention to `hidden-norm-hi`, and the two do not read the same tensor.

Lexical frequency. Scores that read the token identity; three of the four read it against a unigram table and the fourth reads only the context. The table is counted on a training split, never on an evaluation split: the **WikiText-103 training split** under WikiText-103 and the four task rulers, and each loss corpus's own training windows under PG-19 and FineWeb. The three rules that read it are the only ones that bring a corpus statistic into an evaluation; the other eighteen read only the input. Under WikiText-103 the loss and the task rulers read the same table; under PG-19 and FineWeb they do not, so these rules' change in rank across the loss corpora may come from the table rather than from the text scored, and the corpus contrast of §4 rests on them (C.1). The shared WikiText table gives them no advantage on the corpus it is counted from: there `lex-rare` ranks between 19th and 21st of the twenty-one rules depending on the decoder, with a mean rank of 20.3. Its 13.6 in Figure 2 is the average over all three loss corpora, not its WikiText rank.

rule	keeps	modelled on
`lex-rare`	highest −log unigram probability, i.e. the rarest tokens	—
`lex-frequent`	lowest −log unigram probability	—
`lex-nonstop`	any token outside the 100 most frequent ids, ties broken pseudo-randomly	—
`lex-recurring`	tokens that occur again later in the context	—

None. A control whose score reads no signal.

rule	keeps	modelled on
`random`	a subset drawn by a deterministic pseudo-random score per position and passage	—

`random` is reproducible rather than resampled: the same subset is kept on every run, so the floor is a fixed comparison rather than a source of extra variance.

The sign pairs. Five pairs consist of a rule and its reversed partner: `lex-rare`/`lex-frequent`, `key-norm-lo`/`key-norm-hi`, `hidden-norm-hi`/`hidden-norm-lo`, `value-norm-hi`/`value-norm-lo`, and `pos-last`/`pos-first`. Reversing the sign is what turns an axis from a label into a hypothesis: if a score carries information, its reversal should move across the ranking, and §4 reports which pairs do. Three rules are not independent probes, and we say so where it matters: `lex-nonstop` is a thresholded form of `lex-rare`, the attention half of `mix-attn-last` is scored from the same tensor as `attn-all`, and `attn-last8`'s observation window is a shortened form of `mix-attn32-last`'s (C.1). A fourth is nearly a duplicate at the body's budget: `pos-edges` differs from `pos-last` only in a few sink positions at $C = 128$ and $k = 64$ (C.1).

A.3 Why the first context position is protected

Every rule in the family keeps the first position of the context, whatever its own score says. The reason is the attention sink. In a decoder-only transformer the earliest position absorbs a large share of the softmax mass almost regardless of which token sits there, which is the observation StreamingLLM is built on (Xiao et al., 2024). Our gate renormalises after masking, so dropping that position does not remove its mass but redistributes it over whatever remains. The resulting degradation is large, is shared by every rule that happens to drop position 0, and has nothing to do with the signal the rule reads. Without the protection, a table of twenty-one rules would partly be a table of which rules incidentally retain the sink, and the axis structure of §3.1 would be hidden behind it.

Protection is therefore a property of the apparatus rather than of any rule, and it is applied identically to all twenty-one rules, `random` included, and to the trained scorer.

The protected position is charged to the budget rather than added on top of it. Selection sets the protected position's score to $+\infty$ and *then* takes the top-k, so the rule's own score chooses the remaining $k - 1$ positions and the total kept is exactly k for every arm. At $C = 128$ and $k = 64$ this means one forced position and sixty-three selected ones. Every scoring loop asserts both invariants on every batch: that the mask sums to k, and that the protected position is kept.

On benchmarks whose rows are left-padded to a common length, the protected position is the first *real* token of each row, not the first column of the padded tensor. Protecting a pad column would protect nothing and would silently give those rows a budget of $k - 1$.

A.4 The identity check with a mask of all ones

Our intervention multiplies the post-softmax attention weights by a per-position mask and renormalises, which is algebraically the same as deleting the masked-out keys from the softmax denominator. It follows that a mask of all ones must leave the model exactly unchanged. Because the masked path performs an extra multiply and divide, the two are not bit-identical in fp32, so we measure the discrepancy rather than assume it is negligible: the largest absolute difference over all

logits between the masked-with-ones decoder and the unmasked decoder lies between 4.6×10^{-5} and 8.5×10^{-4} across the runs behind the tables.

The check runs at the start of every run, before any rule is scored, and a run that fails it aborts. The tolerance, 5×10^{-2}, is deliberately loose — about sixty times the largest observed value — because the failure it is built to catch is not numerical drift but a wiring fault that changes the model even when nothing is masked. Faults of that kind move the logits by amounts no tolerance in this range would admit, and they are dangerous precisely because the resulting tables still look like tables. A tight tolerance would instead make runs fail on harmless kernel-level variation between decoders.

This check is what licenses the claim that every difference between rules reported in the paper comes from the *content* of a mask rather than from the machinery that applies it.

A.5 One harness limitation

Our gate is a single per-position mask shared across heads. H2O, SnapKV and K-Norm are per-head methods in their published form — each head keeps its own set — so for the rules modelled on them we implement the scoring function faithfully and then average the score over heads before the shared top-k. That is a real departure from those methods. It applies to `mix-attn32-last`, `key-norm-lo` and the attention half of `mix-attn-last`, whose accumulated-attention score is also read head-averaged, so three of the five † rules inherit it and every table containing them does too.

The remaining two † rules are exact rather than approximate. `pos-edges` is exact because StreamingLLM selects by position and has no head dimension to average (the recency half of `mix-attn-last` is exact for the same reason). `attn-last1` is exact for a different reason: TOVA's published default is its layer-wise variant, which averages the last query's attention across heads exactly as we do, and its own ablation prefers that variant over the per-head one. Head-averaging is the published form there, not a departure from it.

B Evaluation sets and rulers

B.1 The six evaluation sets

Every ruler is listed here with its source, its split, what the selection rule is allowed to drop, and how it is read. Two conventions hold throughout. First, selection acts only on the $C = 128$ context positions; whatever follows (a continuation, a question, a candidate) lies outside the maskable region and is never dropped, so no rule can benefit from discarding the text being scored. Second, contexts longer than C are truncated to their **last** C tokens, which is the same truncation a deployed system applies, and rows shorter than C are left-padded (A.3).

Loss rulers. All three are read as nats per token on the continuation, over 1024 held-out windows of $T = 256$ (128 context, 128 continuation).

	WikiText-103	PG-19	FineWeb
source	Merity et al. (2017), `wikitext-103-raw-v1`	Rae et al. (2020), `pg19`	`HuggingFaceFW/fineweb`, dump `CC-MAIN-2025-05`
register	encyclopedic prose	English-language books published before 1919	contemporary web pages
splits	the dataset's own train / validation / test	by document, so no window straddles a split	by document, as PG-19: 2346–2427 / 150–155 / 301–311 documents by tokenizer
windowing	tokenize the split as one stream, reshape to T	as left, after the document split	as left, after the document split
train / val / test windows	8000 / 512 / 1024	8000 / 512 / 1024	8000 / 512 / 1024

The three are not all built alike, because the corpora are shaped differently. WikiText-103 provides train, validation and test splits, and we use them. PG-19 is a book corpus with its own book-level splits, but we build our windows from its books ourselves, and a naive token-stream split would place the validation windows inside the same handful of books as the training windows; we therefore split **by document**, cap each document's contribution so the split spreads over 54 books rather than nine (45 / 3 / 6; 53 books, 44 / 3 / 6, under SmolLM2-1.7B's tokenizer), and take each document's contribution from its **middle** so that Project Gutenberg front matter does not dominate. WikiText-103's training split is also what the unigram table behind the lexical rules is counted from (under PG-19 and FineWeb, each corpus's own training windows), so no rule sees test-split frequencies.

Task rulers. All four are read as accuracy. Each is also stored with the loss on its own gold continuation, which is what §4's within-benchmark agreement compares against; the loss reading is never used to rank the rules on a task ruler (§3.3).

	LAMBADA	CBT-NE / CBT-CN	SQuAD
source	Paperno et al. (2016), test split as pre-processed by OpenAI	Hill et al. (2016), configs NE (named entities) and CN (common nouns)	Rajpurkar et al. (2016), v1.1 dev
split	test	test	dev
maskable context	the passage minus its final word	the twenty context sentences	the passage
never maskable	the target word	the query sentence and the candidate	the question and the answer
task	predict the final word	choose among 10 candidates for a blank	produce the answer span
examples read	all 5153	the first 1000 of 2500 per config	the first 3000 of 10570
evaluated	5151–5153 passages	660–672 (NE) / 560–567 (CN)	2537–2604
accuracy	all target tokens correct	the candidate scoring highest is the gold one	all answer tokens correct

Three details a reader should know. **LAMBADA passages mostly do not fill the window**: the real context length has median 72 and mean 74.6 tokens, and only 17 of 5153 passages reach $C = 128$, so the budget, the fraction $k/C = 0.5$ of each row's real context, is typically 36 positions rather than 64. CBT contexts are truncated to their last C tokens and always fill the window; SQuAD passages longer than C are truncated likewise (1507 of the 3000 questions read on Qwen3-8B, counted before the filter below) and the rest keep the fraction k/C of their real length; over the 2559 questions evaluated the median real length is 120 tokens, so more than half fall short of C. **CBT and SQuAD are restricted to answerable examples**: the criterion is that the answer's first token must survive truncation to the last C tokens; the counts above are after that filter and after discarding malformed examples, and what the filter removes is removed by the truncation rather than by any defect in the examples. **Our exact match is stricter than the official SQuAD metric**: we require every answer token to be argmax-correct under teacher forcing, with no answer normalisation and no alias matching, so our absolute numbers are lower than published SQuAD scores and are comparable only across the arms in this paper. On CBT, of 1000 examples read per configuration, 70 (NE) and 51 (CN) are malformed, and a further 258 (NE) and 382 (CN) are dropped because the answer's first token lies earlier in the passage than the window reaches, leaving the 672 and 567 above on Llama-3.2-1B; on SmolLM2-1.7B, whose tokenizer differs, the counts are 660 and 560.

CBT and SQuAD are capped before filtering (the first 1000 of each CBT configuration's 2500 test examples, the first 3000 of SQuAD's 10570 development questions) because the cost of a cell is linear in the example count and the grid is six decoders wide. The two CBT configurations not used were scored on all six decoders and set aside by the paper's own first test: split-half reliability over 200 splits is 0.84 for the verb subset (the spread between the best and worst of the twenty-one rules is 7.5 to 10.6 accuracy points there) and 0.31 for the preposition subset (2.5 to 3.3 points), against 0.956 to 0.993 for the four task rulers used, so the preposition subset cannot reproduce its own ordering and its disagreement with every other ruler is uninterpretable rather than evidence of a different construct. The verb subset, whose answers are frequent tokens, follows the same

corpus gradient as the two used: the three loss corpora agree with it at +0.20, +0.53 and +0.66. The caps are set once and are the same for every rule and every decoder, so they cannot favour an arm. They do bound precision: at 672 examples the standard error of an accuracy near 70% is about 1.8 points, against a spread of 39 to 48 points between the best and worst rule on CBT-NE across the six decoders, which is the margin the rank correlations in §4 are computed over.

Counts vary slightly across decoders because the filters are applied after tokenization and not all the decoders share a tokenizer (LAMBADA drops only a passage whose target word tokenizes to more than eight tokens or whose context to fewer than four). The counts are as follows: LAMBADA, 5153 passages on five decoders and 5151 on SmolLM2-1.7B; CBT-NE, 672 on five decoders and 660 on SmolLM2-1.7B; CBT-CN, 567 on five and 560 on SmolLM2-1.7B; SQuAD, 2604 questions on Llama-3.2-1B and OLMo-2-1B, 2559 on the three Qwen3 sizes and 2537 on SmolLM2-1.7B.

B.2 Why a full-context accuracy is reported for every task

Every task table in C.1 carries the score of the unmasked decoder on the same examples. This is the ceiling, and it is reported because a selection rule can be measured only inside the headroom the benchmark offers. If the unmasked decoder solved only a few percent of a benchmark, all twenty-one rules would sit near zero, their differences would be resampling noise, and a rank correlation computed from them would be a correlation between two noise vectors, which would not look obviously wrong in a table.

On SQuAD the unmasked exact match runs from 18.47% on Llama-3.2-1B to 59.79% on Qwen3-8B, so there is a wide interval within which a rule's score can lie; on LAMBADA the corresponding ceiling on Llama-3.2-1B is 62.97%, against which the two rules of the Llama-3.2-1B pair in Table 3 score 9.33% and 49.31%. Reading those two numbers without the ceiling would understate how much of the available range separates them.

The floor is reported for the same reason from the other side. `random` keeps a uniformly drawn subset at the same budget, so a rule that fails to beat it has not been shown to detect anything. A table bracketed by neither number cannot distinguish a rule that beats chance from one that merely loses least, which is why both close every task table.

How much headroom each cell leaves, on our narrowest ruler. SQuAD is where the rules' band is narrowest on average, so we give it in full: the ceiling, the floor, and the band the twenty-one rules occupy between the best and the worst of them.

decoder	ceiling (`full`)	floor (`random`)	best rule	worst rule	band
SmolLM2-1.7B	21.17	9.34	12.93	6.58	6.35
Llama-3.2-1B	18.47	7.95	12.06	4.65	7.41
Qwen3-0.6B	24.74	7.42	13.99	3.24	10.75
Qwen3-1.7B	50.14	16.69	29.27	8.79	20.48
Qwen3-8B	59.79	24.15	36.38	13.17	23.21
OLMo-2-1B	51.61	17.20	32.53	8.14	24.39

On the three weaker decoders the band is 6 to 11 points, and the floor, itself one of the twenty-one, sits inside it as on every decoder, so several rules do not beat chance and the bottom of the ranking is compressed against the floor. With about 2600 questions the standard error of an accuracy near 10% is 0.6 points, so even the narrowest band holds on the order of ten distinguishable levels, a coarse resolution for twenty-one rules but not a degenerate one.

Does the band explain the low agreement? A reader is entitled to suspect that SQuAD agrees least with the loss because SQuAD is the most compressed, in which case the low agreement would reflect attenuation rather than a difference between the instruments. We checked directly. For each of the 72 (decoder, task ruler, loss corpus) cells we compared the band against that cell's rank correlation with the loss. Over all 72 cells, wider bands are associated with slightly higher agreement: ρ(band, agreement) = +0.35 pooled, and +0.48, +0.41 and +0.35 within WikiText-103, PG-19 and FineWeb respectively. We report the association because it is real and because a reader can compute it from the C.1 tables, whose parenthesised ranks (ties in table order) give each cell's agreement and whose best and worst rules give its band.

It does not, however, account for the result. Pooled across benchmarks the two are confounded: on average, SQuAD has both the narrowest bands and the lowest agreement, and LAMBADA has bands among the widest and the highest agreement of the four (CBT-NE, whose bands are the widest, ranks only third on agreement), so the pooled figure largely restates that the benchmarks differ. When the benchmark is held fixed and only the decoder and the corpus vary, the relationship is weak and does not keep its sign: LAMBADA +0.37, CBT-NE −0.20, CBT-CN +0.10, SQuAD 0.00. Within SQuAD in particular the band varies almost fourfold, from 6.35 to 24.39 points, while agreement with the loss stays low across that whole range. The cells share decoders and rule sets and are not independent; these are descriptions of the table, not tests, and no p-value is attached.

B.3 Reliability per decoder, and the correction it licenses

The first table gives each ruler's reliability as a mean over decoders; the second gives the per-decoder values behind it. The spread across decoders is narrow enough that no single cell drives the summary.

ruler	reliability
WikiText-103	0.996
PG-19	0.995
FineWeb	0.994
LAMBADA	0.993
SQuAD	0.972
CBT-NE	0.966
CBT-CN	0.956

ruler	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
WikiText-103	0.996	0.993	0.996	0.995	0.997	0.996
PG-19	0.993	0.995	0.994	0.992	0.998	0.995
FineWeb	0.994	0.994	0.993	0.997	0.994	0.993
LAMBADA	0.992	0.997	0.994	0.994	0.992	0.991
CBT-NE	0.970	0.968	0.961	0.962	0.975	0.958
CBT-CN	0.960	0.964	0.937	0.961	0.950	0.962
SQuAD	0.976	0.975	0.984	0.961	0.973	0.966

The splits are random, so each figure is itself an estimate; these are means over 200 splits. The lowest cell in the table is 0.937, CBT-CN on Qwen3-8B.

Per decoder. The block means of Figure 1 hold decoder by decoder. For each decoder the table gives the mean ρ within the loss kind, within the task kind and across the kinds, and how many of the seven rulers have a ruler of their own kind as nearest neighbour.

decoder	loss–loss	task–task	crossing	own-kind nearest
Qwen3-0.6B	+0.80	+0.62	+0.45	7 of 7
Qwen3-1.7B	+0.81	+0.63	+0.48	7 of 7
Qwen3-8B	+0.83	+0.64	+0.48	7 of 7
Llama-3.2-1B	+0.81	+0.73	+0.45	7 of 7
OLMo-2-1B	+0.94	+0.62	+0.22	7 of 7
SmolLM2-1.7B	+0.82	+0.61	+0.56	6 of 7

Within-kind agreement exceeds crossing agreement on six of six decoders for both kinds. The one ruler whose nearest neighbour is of the other kind is LAMBADA on SmolLM2-1.7B, where PG-19 and FineWeb both predict it at +0.91.

Correction for attenuation. Given both reliabilities, the loss-to-task correlation can be corrected (Spearman, 1904) by $\rho_{\text{true}} = \rho_{\text{observed}} / \sqrt{r_{\text{loss}} \cdot r_{\text{task}}}$. The correction can be applied here, and it is small.

loss ruler	task ruler	observed	corrected
WikiText-103	LAMBADA	+0.423	+0.426
WikiText-103	CBT-NE	+0.246	+0.251
WikiText-103	CBT-CN	+0.250	+0.256
WikiText-103	SQuAD	+0.107	+0.108
PG-19	LAMBADA	+0.712	+0.716
PG-19	CBT-NE	+0.475	+0.485
PG-19	CBT-CN	+0.556	+0.570
PG-19	SQuAD	+0.260	+0.264
FineWeb	LAMBADA	+0.741	+0.746
FineWeb	CBT-NE	+0.521	+0.532
FineWeb	CBT-CN	+0.652	+0.669
FineWeb	SQuAD	+0.328	+0.334
overall		**+0.439**	**+0.446**

We report this because it is the first thing a reader trained in measurement will ask for, and because the answer supports the claim rather than undermining it: a pair of instruments measuring the same construct would sit near 1 after this correction. These do not. The corrected correlation, 0.446, estimates what perfectly reliable versions of these two instruments would have shown; the Spearman–Brown and attenuation formulae are derived for Pearson correlations between parallel forms and are applied here to rank correlations as an approximation. Which reliability belongs in the denominator depends on the use. With the corpus fixed it is the split-half figure. If the corpus is a free choice, as it is in the audited practice, the reliability of "held-out loss" as used is its agreement across corpora, the loss-to-loss block mean of +0.83. Correcting with that value on the loss side and with the mean task split-half reliability of 0.97 on the task side moves the crossing correlation from 0.44 to 0.49, still far from 1.

Intervals on the cells. The cells above are point estimates over twenty-one ranks. Resampling each ruler's evaluation units together for all twenty-one rules ($B = 10\,000$, the two rulers of a pair drawn independently) and recomputing every ρ gives the six-decoder block means +0.83 [+0.82, +0.84] for loss to loss, +0.64 [+0.61, +0.65] for task to task and +0.44 [+0.42, +0.46] for loss to task, and the differences that carry the multitrait-multimethod reading, loss-to-loss minus crossing +0.39 [+0.37, +0.41] and task-to-task minus crossing +0.20 [+0.17, +0.22]; on every decoder both differences are above zero in at least 99.7% of resamples. The units are not independent: loss windows are consecutive slices of one stream, SQuAD questions come in runs from one paragraph, CBT rows from one book. Resampling fixed blocks of eight consecutive units instead of single units reproduces every point estimate above to two decimals and moves none of the stated interval endpoints by more than 0.003, and forming the split halves from blocks of eight or twenty consecutive units moves the ruler means by at most 0.005 (individual decoder cells by at most 0.011, SQuAD on SmolLM2-1.7B at blocks of twenty):

ruler	by unit	blocks of 8	blocks of 20
WikiText-103	0.996	0.995	0.994
PG-19	0.995	0.994	0.992
FineWeb	0.994	0.993	0.992
LAMBADA	0.993	0.993	0.993
SQuAD	0.972	0.971	0.967
CBT-NE	0.966	0.967	0.967
CBT-CN	0.956	0.955	0.954

How to read $\rho = 0.44$. It is not zero, and it is not a single number. Averaged over decoders, the twelve corpus–benchmark pairs run from +0.11 to +0.74; at the level of an individual decoder–pair cell they run from -0.12 to +0.91, with the corpus, the benchmark and the decoder each contributing to the spread, which is why §4 reports the range and the factors that produce it rather than the mean alone. The cells are also not independent: the two CBT subsets share passages and the three loss corpora share the rule set. The decoder is therefore the only unit we count as independent, and even that is generous, since the three Qwen3 sizes share a tokenizer and a training corpus.

C FULL RESULTS

C.1 THE FULL GRID

The seven tables below give every rule on every decoder, one table per ruler. Each ruler is read with its own metric: nats per token on the three loss corpora, accuracy on the four task benchmarks. The rank in parentheses is over the twenty-one rules, `random` included, and is computed from the unrounded values, so two losses that print alike can rank in either order; exact ties are broken in table order; the unmasked `full` ceiling is the only entry outside the ranking. The last two rows give the full-context ceiling, which no rule reaches, and the `random` floor, which several rules fall below in every table.

WikiText-103 (nats/token, lower is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
`pos-last`	2.8305 (4)	2.4864 (3)	2.1566 (6)	2.5430 (5)	2.5345 (9)	2.3534 (1)
`pos-first`	2.9078 (16)	2.5658 (16)	2.2292 (19)	2.6331 (17)	2.5511 (12)	2.4641 (15)
`pos-edges`	2.8304 (3)	2.4865 (4)	2.1551 (5)	2.5443 (6)	2.5136 (6)	2.3555 (2)
`pos-uniform`	2.8826 (11)	2.5204 (9)	2.1632 (8)	2.5734 (8)	2.5517 (13)	2.4194 (10)
`attn-all`	2.8755 (8)	2.5243 (10)	2.1819 (11)	2.5746 (9)	2.5293 (8)	2.4183 (9)
`attn-last8`	2.8344 (5)	2.4867 (5)	2.1424 (2)	2.5313 (1)	2.4770 (3)	2.3577 (3)
`attn-last1`	2.8377 (6)	2.4909 (6)	2.1471 (4)	2.5352 (3)	2.4820 (4)	2.3622 (5)
`attn-distant`	2.8888 (12)	2.5424 (13)	2.1999 (14)	2.6018 (14)	2.5358 (10)	2.4368 (14)
`mix-attn-last`	2.8269 (2)	2.4821 (1)	2.1406 (1)	2.5338 (2)	2.4579 (2)	2.3609 (4)
`mix-attn32-last`	2.8258 (1)	2.4833 (2)	2.1458 (3)	2.5378 (4)	2.4552 (1)	2.3649 (6)
`key-norm-lo`	2.8778 (9)	2.5092 (7)	2.1581 (7)	2.5540 (7)	2.4927 (5)	2.3851 (7)
`key-norm-hi`	2.8927 (13)	2.5550 (15)	2.2231 (17)	2.6572 (21)	2.6583 (19)	2.4819 (18)
`hidden-norm-hi`	2.9493 (20)	2.5700 (19)	2.2034 (15)	2.6237 (16)	2.5954 (15)	2.4293 (12)
`hidden-norm-lo`	2.9012 (14)	2.5538 (14)	2.1882 (13)	2.5948 (13)	2.6207 (18)	2.4944 (19)
`value-norm-lo`	2.9022 (15)	2.5689 (18)	2.2339 (20)	2.6021 (15)	2.6189 (17)	2.4979 (20)
`value-norm-hi`	2.9149 (17)	2.5285 (11)	2.1795 (10)	2.5934 (12)	2.5998 (16)	2.4348 (13)
`lex-rare`	2.9482 (19)	2.5823 (20)	2.2433 (21)	2.6518 (20)	2.6790 (21)	2.5042 (21)
`lex-frequent`	2.9561 (21)	2.5922 (21)	2.2187 (16)	2.6429 (18)	2.5926 (14)	2.4797 (16)
`lex-nonstop`	2.9302 (18)	2.5666 (17)	2.2286 (18)	2.6435 (19)	2.6671 (20)	2.4797 (17)
`lex-recurring`	2.8804 (10)	2.5350 (12)	2.1833 (12)	2.5907 (11)	2.5149 (7)	2.4231 (11)
`random`	2.8678 (7)	2.5162 (8)	2.1644 (9)	2.5748 (10)	2.5397 (11)	2.4040 (8)
`full` (ceiling)	2.7419	2.4109	2.0829	2.4572	2.3607	2.2716
`random` (floor)	2.8678	2.5162	2.1644	2.5748	2.5397	2.4040

PG-19 (nats/token, lower is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
`pos-last`	3.0998 (2)	2.8158 (4)	2.5185 (5)	2.8275 (3)	2.9421 (7)	2.5115 (4)
`pos-first`	3.1797 (20)	2.8988 (20)	2.6038 (20)	2.9255 (19)	2.9786 (13)	2.6034 (20)
`pos-edges`	3.1024 (5)	2.8181 (5)	2.5202 (6)	2.8300 (5)	2.9054 (6)	2.5131 (5)
`pos-uniform`	3.1391 (11)	2.8431 (12)	2.5362 (8)	2.8627 (8)	2.9559 (10)	2.5414 (11)
`attn-all`	3.1433 (12)	2.8568 (14)	2.5551 (13)	2.8819 (11)	2.9589 (11)	2.5601 (14)
`attn-last8`	3.1014 (4)	2.8112 (2)	2.5087 (1)	2.8291 (4)	2.8918 (3)	2.5062 (1)
`attn-last1`	3.1089 (6)	2.8189 (6)	2.5116 (2)	2.8324 (6)	2.8974 (4)	2.5097 (2)
`attn-distant`	3.1542 (16)	2.8715 (17)	2.5738 (17)	2.9037 (16)	2.9671 (12)	2.5823 (16)
`mix-attn-last`	3.1004 (3)	2.8144 (3)	2.5130 (4)	2.8273 (2)	2.8801 (2)	2.5115 (3)
`mix-attn32-last`	3.0957 (1)	2.8104 (1)	2.5119 (3)	2.8253 (1)	2.8705 (1)	2.5147 (6)
`key-norm-lo`	3.1212 (7)	2.8298 (9)	2.5346 (7)	2.8618 (7)	2.8993 (5)	2.5352 (8)
`key-norm-hi`	3.1647 (19)	2.8851 (19)	2.5907 (19)	2.9660 (21)	3.0890 (21)	2.5826 (17)
`hidden-norm-hi`	3.1615 (18)	2.8494 (13)	2.5381 (9)	2.8663 (9)	3.0178 (19)	2.5482 (13)
`hidden-norm-lo`	3.1575 (17)	2.8708 (16)	2.5653 (16)	2.9245 (18)	2.9908 (15)	2.6000 (19)
`value-norm-lo`	3.1483 (13)	2.8825 (18)	2.5857 (18)	2.9016 (15)	2.9902 (14)	2.5909 (18)
`value-norm-hi`	3.1493 (15)	2.8411 (10)	2.5567 (14)	2.9227 (17)	3.0077 (17)	2.5360 (10)
`lex-rare`	3.1251 (9)	2.8270 (7)	2.5417 (12)	2.9014 (14)	3.0210 (20)	2.5355 (9)
`lex-frequent`	3.2171 (21)	2.9272 (21)	2.6143 (21)	2.9511 (20)	2.9987 (16)	2.6159 (21)
`lex-nonstop`	3.1245 (8)	2.8276 (8)	2.5405 (10)	2.8985 (13)	3.0103 (18)	2.5322 (7)
`lex-recurring`	3.1484 (14)	2.8613 (15)	2.5593 (15)	2.8963 (12)	2.9428 (8)	2.5744 (15)
`random`	3.1310 (10)	2.8420 (11)	2.5413 (11)	2.8706 (10)	2.9517 (9)	2.5437 (12)
`full` (ceiling)	3.0371	2.7564	2.4653	2.7669	2.7982	2.4617
`random` (floor)	3.1310	2.8420	2.5413	2.8706	2.9517	2.5437

FineWeb (nats/token, lower is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
`pos-last`	3.2340 (3)	2.9570 (4)	2.6812 (6)	2.8980 (5)	3.0232 (8)	2.6862 (2)
`pos-first`	3.3180 (20)	3.0393 (20)	2.7626 (21)	2.9800 (20)	3.0558 (13)	2.7723 (18)
`pos-edges`	3.2344 (4)	2.9572 (5)	2.6798 (5)	2.8990 (6)	2.9985 (6)	2.6874 (3)
`pos-uniform`	3.2705 (11)	2.9844 (11)	2.7028 (9)	2.9162 (9)	3.0334 (11)	2.7147 (12)
`attn-all`	3.2815 (15)	2.9968 (14)	2.7129 (14)	2.9318 (14)	3.0249 (9)	2.7332 (15)
`attn-last8`	3.2337 (2)	2.9528 (1)	2.6699 (1)	2.8903 (1)	2.9653 (3)	2.6837 (1)
`attn-last1`	3.2385 (6)	2.9557 (3)	2.6708 (2)	2.8923 (2)	2.9704 (4)	2.6876 (4)
`attn-distant`	3.2981 (18)	3.0163 (17)	2.7385 (18)	2.9592 (19)	3.0400 (12)	2.7477 (16)
`mix-attn-last`	3.2384 (5)	2.9586 (6)	2.6758 (4)	2.8961 (4)	2.9626 (2)	2.6887 (5)
`mix-attn32-last`	3.2322 (1)	2.9536 (2)	2.6748 (3)	2.8937 (3)	2.9554 (1)	2.6894 (6)
`key-norm-lo`	3.2677 (10)	2.9829 (10)	2.7076 (13)	2.9179 (11)	2.9888 (5)	2.7069 (9)
`key-norm-hi`	3.2892 (16)	3.0098 (16)	2.7296 (17)	2.9563 (18)	3.1110 (21)	2.7479 (17)
`hidden-norm-hi`	3.2778 (13)	2.9876 (13)	2.7051 (12)	2.9241 (13)	3.0602 (16)	2.7101 (11)
`hidden-norm-lo`	3.3032 (19)	3.0208 (19)	2.7282 (16)	2.9514 (16)	3.0633 (18)	2.7778 (20)
`value-norm-lo`	3.2903 (17)	3.0193 (18)	2.7390 (19)	2.9536 (17)	3.0659 (19)	2.7754 (19)
`value-norm-hi`	3.2753 (12)	2.9806 (9)	2.7041 (11)	2.9172 (10)	3.0564 (14)	2.7084 (10)
`lex-rare`	3.2506 (7)	2.9660 (7)	2.6871 (7)	2.9056 (7)	3.0623 (17)	2.6971 (7)
`lex-frequent`	3.3433 (21)	3.0542 (21)	2.7601 (20)	2.9827 (21)	3.0696 (20)	2.7954 (21)
`lex-nonstop`	3.2518 (8)	2.9678 (8)	2.6883 (8)	2.9088 (8)	3.0590 (15)	2.6987 (8)
`lex-recurring`	3.2799 (14)	2.9982 (15)	2.7181 (15)	2.9363 (15)	3.0141 (7)	2.7328 (14)
`random`	3.2669 (9)	2.9848 (12)	2.7031 (10)	2.9231 (12)	3.0297 (10)	2.7181 (13)
`full` (ceiling)	3.1918	2.9171	2.6442	2.8615	2.9053	2.6428
`random` (floor)	3.2669	2.9848	2.7031	2.9231	3.0297	2.7181

LAMBADA (accuracy %, higher is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
`pos-last`	27.98 (10)	33.90 (9)	40.29 (10)	33.96 (9)	32.52 (5)	37.57 (6)
`pos-first`	23.48 (16)	28.14 (16)	28.62 (18)	26.43 (17)	18.51 (16)	18.33 (18)
`pos-edges`	27.61 (11)	33.36 (10)	39.80 (11)	32.97 (11)	32.33 (6)	36.52 (9)
`pos-uniform`	24.35 (15)	30.80 (13)	39.57 (12)	32.10 (12)	22.74 (12)	28.01 (13)
`attn-all`	30.56 (7)	38.31 (6)	46.79 (7)	39.39 (7)	16.13 (17)	25.24 (14)
`attn-last8`	38.99 (3)	48.42 (3)	56.22 (3)	47.43 (3)	31.88 (7)	45.93 (1)
`attn-last1`	43.33 (1)	51.83 (1)	60.53 (1)	50.40 (1)	41.68 (1)	44.32 (2)
`attn-distant`	25.01 (13)	29.89 (14)	30.64 (16)	28.31 (15)	19.37 (15)	19.94 (17)
`mix-attn-last`	28.10 (9)	37.26 (7)	50.55 (6)	43.64 (6)	25.27 (11)	38.44 (4)
`mix-attn32-last`	28.68 (8)	34.56 (8)	41.20 (9)	33.73 (10)	33.22 (3)	36.23 (10)
`key-norm-lo`	33.61 (5)	43.86 (5)	53.08 (4)	47.41 (4)	32.76 (4)	37.22 (8)
`key-norm-hi`	11.55 (20)	12.65 (20)	17.23 (20)	9.33 (20)	11.22 (18)	9.88 (19)
`hidden-norm-hi`	27.61 (12)	32.47 (11)	41.94 (8)	34.19 (8)	27.54 (9)	37.55 (7)
`hidden-norm-lo`	19.37 (18)	24.88 (19)	28.22 (19)	22.74 (19)	4.68 (21)	6.52 (21)
`value-norm-lo`	32.00 (6)	27.50 (17)	29.44 (17)	31.17 (13)	5.63 (19)	20.11 (16)
`value-norm-hi`	15.84 (19)	29.21 (15)	39.43 (13)	27.27 (16)	27.25 (10)	32.85 (11)
`lex-rare`	40.33 (2)	49.82 (2)	58.12 (2)	49.31 (2)	33.98 (2)	40.05 (3)
`lex-frequent`	4.58 (21)	7.22 (21)	13.27 (21)	7.32 (21)	4.91 (20)	7.20 (20)
`lex-nonstop`	36.68 (4)	44.91 (4)	51.83 (5)	43.70 (5)	29.92 (8)	38.36 (5)
`lex-recurring`	21.89 (17)	26.41 (18)	33.26 (15)	26.08 (18)	20.42 (14)	21.59 (15)
`random`	24.96 (14)	31.05 (12)	38.07 (14)	30.39 (14)	22.24 (13)	28.67 (12)
`full` (ceiling)	54.28	63.87	72.19	62.97	65.13	67.68
`random` (floor)	24.96	31.05	38.07	30.39	22.24	28.67

CBT-NE (accuracy %, higher is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
`pos-last`	63.84 (9)	68.01 (11)	75.45 (10)	67.56 (10)	69.35 (6)	72.58 (6)
`pos-first`	62.35 (11)	66.07 (12)	71.13 (14)	66.07 (12)	67.26 (10)	67.42 (14)
`pos-edges`	63.54 (10)	68.90 (10)	76.04 (9)	67.26 (11)	69.49 (5)	71.52 (9)
`pos-uniform`	52.53 (17)	60.27 (17)	69.64 (16)	61.01 (16)	60.12 (13)	61.82 (17)
`attn-all`	61.01 (13)	70.54 (8)	73.96 (11)	69.20 (7)	56.85 (15)	63.94 (15)
`attn-last8`	66.82 (7)	75.15 (5)	82.59 (5)	72.32 (5)	55.65 (16)	71.97 (8)
`attn-last1`	72.77 (3)	78.12 (3)	83.33 (3)	76.19 (4)	55.21 (17)	72.27 (7)
`attn-distant`	67.86 (5)	70.68 (7)	76.49 (8)	69.79 (6)	67.41 (9)	71.36 (10)
`mix-attn-last`	61.16 (12)	70.54 (9)	76.93 (7)	68.30 (8)	59.23 (14)	69.70 (11)
`mix-attn32-last`	67.26 (6)	71.28 (6)	77.38 (6)	68.01 (9)	68.45 (8)	73.03 (5)
`key-norm-lo`	71.28 (4)	76.04 (4)	83.33 (4)	81.40 (2)	69.35 (7)	81.97 (3)
`key-norm-hi`	43.30 (20)	51.64 (20)	54.76 (20)	43.90 (20)	52.98 (18)	47.27 (20)
`hidden-norm-hi`	49.11 (18)	62.65 (16)	71.43 (13)	57.74 (18)	80.06 (2)	74.09 (4)
`hidden-norm-lo`	63.99 (8)	65.92 (13)	69.64 (17)	65.18 (13)	40.48 (20)	54.85 (19)
`value-norm-lo`	58.18 (15)	60.27 (18)	63.24 (19)	64.58 (14)	41.96 (19)	58.94 (18)
`value-norm-hi`	48.21 (19)	60.27 (19)	72.17 (12)	56.25 (19)	79.32 (4)	67.88 (13)
`lex-rare`	78.72 (1)	85.12 (1)	89.14 (1)	83.04 (1)	82.74 (1)	82.42 (1)
`lex-frequent`	30.65 (21)	36.76 (21)	47.17 (21)	38.10 (21)	38.69 (21)	43.48 (21)
`lex-nonstop`	76.49 (2)	80.80 (2)	86.16 (2)	80.51 (3)	79.76 (3)	82.12 (2)
`lex-recurring`	58.93 (14)	63.24 (15)	69.35 (18)	61.01 (17)	61.61 (12)	63.79 (16)
`random`	57.74 (16)	63.54 (14)	70.83 (15)	61.46 (15)	62.05 (11)	68.03 (12)
`full` (ceiling)	84.23	86.76	91.07	86.01	88.54	88.33
`random` (floor)	57.74	63.54	70.83	61.46	62.05	68.03

CBT-CN (accuracy %, higher is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
pos-last	78.13 (8)	80.42 (9)	86.42 (13)	82.19 (10)	82.36 (9)	85.36 (11)
pos-first	70.72 (17)	75.49 (16)	77.78 (20)	74.60 (17)	74.43 (17)	77.68 (17)
pos-edges	77.95 (9)	79.19 (13)	87.13 (11)	81.83 (12)	83.07 (8)	86.07 (9)
pos-uniform	73.72 (13)	80.42 (10)	88.89 (6)	80.78 (13)	83.25 (7)	82.32 (13)
attn-all	75.13 (11)	80.42 (11)	88.54 (8)	83.07 (9)	76.72 (15)	86.07 (10)
attn-last8	81.66 (4)	85.36 (4)	90.12 (3)	86.95 (3)	83.42 (6)	89.82 (3)
attn-last1	78.66 (6)	81.48 (8)	88.54 (9)	85.19 (4)	77.95 (14)	88.04 (7)
attn-distant	74.07 (12)	78.31 (14)	84.13 (15)	77.95 (14)	78.13 (13)	82.14 (14)
mix-attn-last	77.25 (10)	82.72 (7)	89.95 (4)	85.01 (5)	82.36 (10)	87.32 (8)
mix-attn32-last	78.48 (7)	80.07 (12)	87.13 (12)	82.19 (11)	83.95 (5)	85.36 (12)
key-norm-lo	83.25 (3)	88.01 (1)	88.01 (10)	84.48 (6)	81.48 (11)	89.64 (4)
key-norm-hi	62.61 (20)	65.78 (20)	80.95 (17)	66.67 (20)	72.66 (18)	69.64 (18)
hidden-norm-hi	81.31 (5)	84.30 (5)	89.77 (5)	84.30 (7)	86.07 (2)	88.57 (6)
hidden-norm-lo	63.67 (19)	69.84 (19)	78.66 (18)	67.72 (19)	63.32 (20)	67.50 (19)
value-norm-lo	71.25 (16)	70.72 (18)	77.60 (21)	69.31 (18)	63.32 (21)	67.14 (20)
value-norm-hi	72.31 (15)	82.89 (6)	88.89 (7)	83.42 (8)	86.07 (3)	90.00 (2)
lex-rare	84.83 (1)	87.30 (2)	91.53 (2)	88.36 (1)	86.42 (1)	90.18 (1)
lex-frequent	57.14 (21)	64.37 (21)	78.48 (19)	65.78 (21)	65.26 (19)	66.43 (21)
lex-nonstop	83.77 (2)	86.77 (3)	91.71 (1)	87.48 (2)	86.07 (4)	89.46 (5)
lex-recurring	70.72 (18)	74.25 (17)	81.13 (16)	76.37 (16)	76.37 (16)	78.75 (16)
random	73.37 (14)	77.07 (15)	84.30 (14)	77.25 (15)	78.66 (12)	81.96 (15)
full (ceiling)	85.89	89.07	92.77	89.59	91.18	92.86
random (floor)	73.37	77.07	84.30	77.25	78.66	81.96

SQuAD (accuracy %, higher is better)

rule	Qwen3-0.6B	Qwen3-1.7B	Qwen3-8B	Llama-3.2-1B	OLMo-2-1B	SmolLM2-1.7B
pos-last	13.17 (2)	25.32 (4)	33.18 (6)	10.64 (6)	15.94 (15)	12.57 (2)
pos-first	13.99 (1)	29.27 (1)	36.38 (1)	11.06 (3)	32.53 (1)	12.93 (1)
pos-edges	12.04 (4)	23.95 (8)	31.14 (8)	9.95 (7)	20.55 (8)	11.82 (4)
pos-uniform	4.18 (20)	13.13 (17)	19.66 (18)	7.87 (15)	14.09 (17)	7.96 (17)
attn-all	8.05 (12)	17.62 (13)	23.76 (14)	8.49 (11)	20.24 (10)	7.49 (19)
attn-last8	8.83 (10)	18.52 (12)	26.46 (12)	8.60 (10)	18.93 (12)	9.30 (13)
attn-last1	7.54 (14)	14.81 (16)	22.70 (15)	8.29 (12)	15.17 (16)	8.20 (16)
attn-distant	12.54 (3)	25.75 (2)	32.63 (7)	10.75 (5)	28.73 (2)	12.10 (3)
mix-attn-last	8.71 (11)	19.11 (11)	27.51 (11)	7.95 (13)	20.89 (6)	9.07 (14)
mix-attn32-last	11.22 (7)	23.60 (9)	31.03 (9)	9.56 (8)	24.77 (3)	10.29 (9)
key-norm-lo	10.04 (8)	24.15 (7)	30.60 (10)	10.79 (4)	21.08 (5)	11.12 (7)
key-norm-hi	4.96 (18)	12.66 (18)	22.31 (16)	6.76 (18)	12.14 (18)	9.38 (11)
hidden-norm-hi	9.30 (9)	25.36 (3)	34.00 (4)	9.10 (9)	22.47 (4)	10.13 (10)
hidden-norm-lo	3.24 (21)	8.79 (21)	15.24 (20)	6.26 (20)	8.14 (21)	6.98 (20)
value-norm-lo	4.53 (19)	10.16 (20)	13.17 (21)	7.30 (16)	8.45 (20)	6.58 (21)
value-norm-hi	7.78 (13)	19.62 (10)	33.45 (5)	7.26 (17)	20.85 (7)	11.12 (8)
lex-rare	11.84 (5)	24.89 (6)	35.05 (2)	12.06 (1)	20.31 (9)	11.47 (5)
lex-frequent	6.06 (17)	12.08 (19)	16.18 (19)	4.65 (21)	10.94 (19)	7.57 (18)
lex-nonstop	11.76 (6)	25.21 (5)	34.04 (3)	11.48 (2)	19.70 (11)	11.35 (6)
lex-recurring	7.27 (16)	15.04 (15)	20.20 (17)	6.61 (19)	18.01 (13)	8.51 (15)
random	7.42 (15)	16.69 (14)	24.15 (13)	7.95 (14)	17.20 (14)	9.34 (12)
full (ceiling)	24.74	50.14	59.79	18.47	51.61	21.17
random (floor)	7.42	16.69	24.15	7.95	17.20	9.34

Mean rank of each rule, loss rulers against task rulers. The two mean-rank columns are what Figure 2 plots; the difference column is the task-ruler mean minus the loss-ruler mean, taken between the two rounded columns as printed. A rule that sits near the top of one and near the bottom of the other is one on which the two instruments disagree.

axis	rule	mean rank, 3 loss rulers	mean rank, 4 task rulers	difference
position	`pos-last`	4.5	8.2	+3.7
position	`pos-first`	17.7	11.9	−5.8
position	`pos-edges`	4.8	8.9	+4.1
position	`pos-uniform`	10.1	14.1	+4.0
attention	`attn-all`	11.7	11.2	−0.5
attention	`attn-last8`	2.4	6.6	+4.2
attention	`attn-last1`	4.2	7.5	+3.3
attention	`attn-distant`	15.1	10.0	−5.1
attention and position	`mix-attn-last`	3.1	8.9	+5.8
attention and position	`mix-attn32-last`	2.6	8.0	+5.4
key geometry	`key-norm-lo`	7.9	5.4	−2.5
key geometry	`key-norm-hi`	18.0	18.6	+0.6
hidden geometry	`hidden-norm-hi`	14.2	8.1	−6.1
hidden geometry	`hidden-norm-lo`	16.7	18.5	+1.8
value geometry	`value-norm-lo`	17.2	17.6	+0.4
value geometry	`value-norm-hi`	12.7	11.3	−1.4
lexical frequency	`lex-rare`	13.6	2.3	−11.3
lexical frequency	`lex-frequent`	19.4	20.2	+0.8
lexical frequency	`lex-nonstop`	12.7	4.0	−8.7
lexical frequency	`lex-recurring`	12.3	16.0	+3.7
none	`random`	10.1	13.7	+3.6

Robustness to the composition of the family. Two of the rules overlap others. The first is `attn-last8`, whose observation window is a shortened form of the one `mix-attn32-last` uses (`attn-last8` is one of the three rules A.2 lists as not independent probes; the other two, `lex-nonstop` and `mix-attn-last`, are kept in this first check); the second is `pos-edges`, which at $C = 128$ and $k = 64$ differs from `pos-last` only in a few sink positions. The WikiText gap between `pos-edges` and `pos-last` is at most 0.0021 nat on five of the six decoders (only 0.0001 on two of them) and 0.0209 on OLMo-2-1B alone. Dropping both barely moves the block means (loss–loss +0.78, task–task +0.66 and loss–task +0.43, against +0.83, +0.64 and +0.44 with all twenty-one rules; WikiText–SQuAD falls to +0.07), so the correlations are not an artefact of these two rules. Nor is the disagreement an artefact of the controls, which mostly sit near the bottom under both instruments (four of the five sign-reversed controls do; `pos-first` is 17.7 by loss but 11.9 on the tasks, and `random` 10.1 and 13.7) and so mostly add concordant pairs: without `random` the block means are +0.83, +0.63 and +0.43 (loss–loss, task–task, loss–task), and without the five sign-reversed controls as well +0.82 [+0.80, +0.82], +0.45 [+0.41, +0.48] and +0.12 [+0.09, +0.15], so, as far as the controls go, the crossing agreement the body reports is the generous reading. Without the three rules that read the unigram table (`lex-rare`, `lex-frequent` and `lex-nonstop`; A.2) the block means are +0.92, +0.54 and +0.52, and the loss–task means are +0.51, +0.53 and +0.52 under WikiText-103, PG-19 and FineWeb, against +0.26, +0.50 and +0.56 with them; on SQuAD, WikiText-103 is then slightly ahead (+0.23 against +0.21 and +0.21). On the five rules modelled on published scoring functions alone († in Table 1; ρ over five items is coarse) the block means are +0.76 [+0.67, +0.81], +0.13 [+0.01, +0.20] and −0.18 [−0.26, −0.10]: on these five, the loss and the tasks are weakly anticorrelated.

C.2 Every decoder's unordered pairs and confident reversals

Table 3 shows three pairs: the widest confident reversal under LAMBADA whose loss-preferred rule follows a published scoring function, on Qwen3-8B, then the widest pair under LAMBADA that the loss leaves unordered, on Llama-3.2-1B, and the median one, on SmolLM2-1.7B. No decoder was chosen for being the worst, and the way to show that is to give the whole population rather than to assert it: the unordered pairs first, the reversals after.

For each decoder and each task ruler, the table below gives the number of rule pairs the loss leaves unordered, how many of those the task ruler does order, and the widest of those, with both its loss

difference and its accuracy difference. The unordered set is a property of the loss alone, so it is the same 84 pairs in all four blocks; what changes between them is how many the ruler can separate and which one is widest.

Table 2: Every decoder's unordered pairs, under all four task rulers. Counts are out of the 210 pairs each decoder contributes, and the unordered set is the same 84 in every block because it is a property of the loss alone. The table lists point differences; the intervals of Table 3 come from a paired bootstrap over the evaluation units, $B = 10\,000$, drawn independently of this table, a draw that changes no point difference; the one pair the two tables share reads the same in both.

decoder	ruler	unordered	ruler orders	widest such pair	loss diff.	accuracy diff.
Qwen3-0.6B	LAMBADA	14	10	`lex-rare` vs `lex-frequent`	−0.0079	+35.75
Qwen3-1.7B	LAMBADA	18	15	`pos-edges` vs `attn-last1`	−0.0044	−18.47
Qwen3-8B	LAMBADA	17	14	`key-norm-hi` vs `lex-nonstop`	−0.0055	−34.60
Llama-3.2-1B	LAMBADA	13	12	`key-norm-hi` vs `lex-rare`	+0.0054	−39.98
OLMo-2-1B	LAMBADA	9	9	`hidden-norm-hi` vs `lex-frequent`	+0.0028	+22.63
SmolLM2-1.7B	LAMBADA	13	13	`lex-frequent` vs `lex-nonstop`	−0.0001	−31.16
Qwen3-0.6B	CBT-NE	14	8	`lex-rare` vs `lex-frequent`	−0.0079	+48.07
Qwen3-1.7B	CBT-NE	18	12	`value-norm-lo` vs `lex-nonstop`	+0.0023	−20.54
Qwen3-8B	CBT-NE	17	13	`key-norm-hi` vs `lex-nonstop`	−0.0055	−31.40
Llama-3.2-1B	CBT-NE	13	11	`lex-frequent` vs `lex-nonstop`	−0.0006	−42.41
OLMo-2-1B	CBT-NE	9	7	`hidden-norm-hi` vs `lex-frequent`	+0.0028	+41.37
SmolLM2-1.7B	CBT-NE	13	3	`lex-frequent` vs `lex-nonstop`	−0.0001	−38.64
Qwen3-0.6B	CBT-CN	14	8	`lex-rare` vs `lex-frequent`	−0.0079	+27.69
Qwen3-1.7B	CBT-CN	18	10	`value-norm-lo` vs `lex-nonstop`	+0.0023	−16.05
Qwen3-8B	CBT-CN	17	8	`value-norm-lo` vs `lex-nonstop`	+0.0052	−14.11
Llama-3.2-1B	CBT-CN	13	8	`lex-frequent` vs `lex-nonstop`	−0.0006	−21.69
OLMo-2-1B	CBT-CN	9	6	`hidden-norm-hi` vs `lex-frequent`	+0.0028	+20.81
SmolLM2-1.7B	CBT-CN	13	10	`lex-frequent` vs `lex-nonstop`	−0.0001	−23.04
Qwen3-0.6B	SQuAD	14	14	`pos-first` vs `value-norm-lo`	+0.0057	+9.46
Qwen3-1.7B	SQuAD	18	17	`pos-first` vs `value-norm-lo`	−0.0031	+19.11
Qwen3-8B	SQuAD	17	14	`pos-first` vs `value-norm-lo`	−0.0047	+23.21
Llama-3.2-1B	SQuAD	13	5	`lex-frequent` vs `lex-nonstop`	−0.0006	−6.84
OLMo-2-1B	SQuAD	9	7	`pos-first` vs `pos-uniform`	−0.0006	+18.43
SmolLM2-1.7B	SQuAD	13	8	`lex-frequent` vs `lex-nonstop`	−0.0001	−3.78

Four things are worth reading off it. The unordered counts sum to 84, and the ordered ones to 73, 54, 50 and 65 under the four rulers, so the two unordered pairs of Table 3 sit inside a population and not beside one. The decoder with the fewest unordered pairs, OLMo-2-1B at nine, is one of the two decoders (the other is SmolLM2-1.7B) on which LAMBADA orders every one of them, so a loss that leaves fewer pairs unordered is not thereby more useful. And which unordered pair is widest is set by the decoder as much as by the ruler (on SmolLM2-1.7B it is `lex-frequent` against `lex-nonstop` under all four rulers), though the ruler can move it: on Llama-3.2-1B it is `key-norm-hi` against `lex-rare` under LAMBADA and `lex-frequent` against `lex-nonstop` under CBT-NE and SQuAD (under CBT-CN the two pairs tie at 21.69 points), while SQuAD returns `pos-first` against `value-norm-lo` on all three Qwen3 sizes, which share a tokenizer and a training corpus. The cell in which the ruler separates the fewest pairs is SmolLM2-1.7B under CBT-NE, 3 of 13; on the same decoder LAMBADA separates all 13, which is why all four rulers are reported.

One caveat. Not every tie is a failure of the loss. On Qwen3-0.6B, `pos-edges` − `pos-last` = −0.0001 [−0.0017, +0.0015]; at $C = 128$ with $k = 64$ those two rules differ only in a few sink positions, so the tie is convergence of the rules, not coarseness of the loss. This pair is itself one of the 84, so not all of them are of one kind; the two shown in Table 3 pair rules from different groups.

Table 3: The widest confident reversal, the widest unordered pair, and the median unordered pair. The first block is the widest pair on which the loss and LAMBADA are both confident and disagree, among those where neither rule is a sign-reversed control, both rules score above the random floor on LAMBADA, and the rule the loss prefers follows a published scoring function (C.2 lists the two wider pairs whose loss-preferred rule has no published counterpart); it was selected as the widest, so its interval is conditional on that selection, and Figure 3 shows the population it was drawn from; the second is the pair with the largest LAMBADA gap among the 84 the loss leaves unordered; the third is the upper of the two unordered pairs whose LAMBADA gaps straddle the median gap, 8.23 points. $C = 128$, budget $k/C = 0.5$ ($k = 64$ on the loss windows, half of each passage on LAMBADA), paired bootstrap, $B = 10\,000$. Each difference is the upper rule minus the lower; bold marks the rule a ruler prefers where its 95% interval excludes zero.

Widest confident reversal — Qwen3-8B.

	WikiText loss (nats/token)	LAMBADA accuracy
`pos-edges`	**2.1551**	39.80%
`lex-rare`	2.2433	**58.12%**
difference	−0.0882, 95% CI [−0.0943, −0.0821]	−18.32 points, 95% CI [−19.89, −16.71]
`full` (ceiling)	2.0829	72.19%
`random` (floor)	2.1644	38.07%

Widest unordered — Llama-3.2-1B.

	WikiText loss (nats/token)	LAMBADA accuracy
`key-norm-hi`	2.6572	9.33%
`lex-rare`	2.6518	**49.31%**
difference	+0.0054, 95% CI [−0.0004, +0.0112]	−39.98 points, 95% CI [−41.47, −38.48]
`full` (ceiling)	2.4572	62.97%
`random` (floor)	2.5748	30.39%

Median — SmolLM2-1.7B.

	WikiText loss (nats/token)	LAMBADA accuracy
`pos-last`	2.3534	37.57%
`attn-last8`	2.3577	**45.93%**
difference	−0.0043, 95% CI [−0.0092, +0.0005]	−8.37 points, 95% CI [−9.75, −7.03]
`full` (ceiling)	2.2716	67.68%
`random` (floor)	2.4040	28.67%

Confident reversals. An unordered pair is a cost, and a bounded one: an interval that contains zero is a statement about 1024 windows, and more windows would narrow it. The cost that more windows would not shrink is a reversed direction. Among the pairs the WikiText-103 loss orders

at 95% (its interval excludes zero), the table below counts those each task ruler also orders at 95%, split by whether the two orderings agree. The procedure is the same as for Table 2; in this draw of the bootstrap the loss orders 1177 pairs rather than 1176, one pair lying at the boundary, so the counts are quoted from this draw.

task ruler	both confident	concordant	reversed	reversed share	task uncertain
LAMBADA	1090	759	331	30%	87
CBT-NE	936	574	362	39%	241
CBT-CN	845	562	283	33%	332
SQuAD	944	532	412	44%	233

Under the other two loss corpora the shares are lower and the order of the rulers the same, except that LAMBADA and CBT-CN swap places under FineWeb: PG-19 orders 1179 pairs at 95% and is reversed on 19% (LAMBADA), 29% (CBT-NE), 20% (CBT-CN) and 36% (SQuAD) of the pairs both rulers order, and on 57 pairs by all four rulers; FineWeb orders 1161 and is reversed on 17%, 28%, 16% and 34%, and on 38 pairs by all four. The body's WikiText-103 counts are the pessimistic reading, though under PG-19 and FineWeb the three table-reading lexical rules read those corpora's own tables (A.2), and the corpus that transfers best still reverses one pair in six on LAMBADA and one in three on SQuAD.

Neither the random rule nor the sign-reversed controls drive these shares. Recounted without `random` (20 rules, 190 pairs per decoder) the reversed shares are 31%, 39%, 34% and 44% for LAMBADA, CBT-NE, CBT-CN and SQuAD; without the five sign-reversed controls as well (15 rules, 105 pairs per decoder) they rise to 42%, 55%, 58% and 58%, because the pairs removed are mostly ones both instruments rank low and order the same way. Restricted to the five published-derived rules (10 pairs per decoder, 48 of the 60 ordered by the loss), the reversed shares are 56% (LAMBADA), 83% (CBT-NE), 69% (CBT-CN) and 58% (SQuAD).

Three checks on the counting itself. If a loss-ordered pair had no true difference on a task ruler, the ruler's 95% interval would still exclude zero in the reversed direction 2.5% of the time, about 30 of the 1177 pairs, against the 283 to 412 observed. Recounted at 99%, the loss orders 1151 pairs and the reversed shares are 30%, 38%, 32% and 43%; resampling fixed blocks of eight consecutive units instead of single units, so that questions from one paragraph and windows from one document move together, gives 1165 ordered pairs and 30%, 38%, 33% and 43%.

What trusting the loss costs. For each loss corpus and task ruler, the table summarises over the six decoders the task accuracy of the rule the task ranks first minus that of the rule the loss ranks first. The loss's rule is chosen on the loss windows; the task's rule is chosen on one half of the task's units and scored on the other, then the halves are swapped and the two readings averaged, so no rule is chosen on the units it is scored on. The interval comes from a paired bootstrap over the scoring half ($B = 10\,000$), and "confident" counts decoders whose interval lies above zero. The second group of three columns repeats the choice among the five published-derived rules only.

		twenty-one rules			five published-derived rules		
loss corpus	task ruler	mean	range	confident	mean	range	confident
WikiText-103	LAMBADA	9.8	3.0 to 14.7	6 of 6	10.4	6.8 to 14.7	6 of 6
PG-19	LAMBADA	10.2	0.0 to 17.3	5 of 6	9.5	0.0 to 17.3	4 of 6
FineWeb	LAMBADA	5.6	0.0 to 14.7	5 of 6	8.0	0.0 to 17.3	4 of 6
WikiText-103	CBT-NE	12.0	8.8 to 14.6	6 of 6	6.8	−0.6 to 13.1	5 of 6
PG-19	CBT-NE	11.8	6.6 to 15.0	6 of 6	5.6	−1.5 to 13.4	4 of 6
FineWeb	CBT-NE	10.4	6.6 to 14.3	6 of 6	4.3	−1.5 to 10.5	4 of 6
WikiText-103	CBT-CN	2.9	0.0 to 6.4	3 of 6	1.5	−2.1 to 5.3	3 of 6
PG-19	CBT-CN	3.3	−0.7 to 6.9	3 of 6	2.3	−1.6 to 7.9	2 of 6
FineWeb	CBT-CN	1.6	−0.7 to 6.4	1 of 6	2.2	−1.6 to 7.9	3 of 6
WikiText-103	SQuAD	5.0	−0.3 to 10.2	5 of 6	2.0	0.0 to 4.7	3 of 6
PG-19	SQuAD	4.8	1.6 to 8.3	6 of 6	2.4	0.0 to 8.4	2 of 6
FineWeb	SQuAD	5.9	2.5 to 10.8	6 of 6	2.0	0.0 to 8.4	2 of 6

A zero is, with one exception, a cell where the loss's rule is also the task's rule on both halves (the exception is Qwen3-8B under CBT-CN on WikiText-103, where the rule chosen on one half, `lex-nonstop`, ties the loss's rule on the other, so the average is 0.00 with interval [−1.41, +1.41]), and a negative entry one where the rule chosen on one half scores below the loss's rule on the other. Averaged over the six decoders, the WikiText-103 loss's rule costs 9.8 points on LAMBADA, 12.0 on CBT-NE, 2.9 on CBT-CN and 5.0 on SQuAD, and the interval is above zero in 20 of the 24 decoder–ruler cells; choosing and scoring on the same units moves these means by at most 0.8 points and leaves the count at 20. Among the five published-derived rules the WikiText-103 cost is smaller on CBT and SQuAD and slightly larger on LAMBADA (10.4 against 9.8); under FineWeb the five-rule cost is the larger one on LAMBADA and CBT-CN.

Wider loss margins do not make the direction reliable. The table below bins the loss-ordered pairs by the size of the loss difference and gives the share of pairs in each bin on which the task ruler's point estimate orders the pair in the same direction as the loss:

$\lvert\Delta$ loss$\rvert$ (nat)	pairs	LAMBADA	CBT-NE	CBT-CN	SQuAD
< 0.006	25	64%	60%	64%	64%
0.006–0.015	123	50%	43%	46%	41%
0.015–0.03	190	63%	54%	58%	56%
≥ 0.03	839	72%	63%	64%	56%

Pooled over rulers, 560 of the 1177 loss-ordered pairs are reversed by no task ruler (320 of them concordant with all four), 224 by one, 141 by two, 126 by three and 126 by all four; 617 are reversed by at least one, and 331 are reversed by at least one ruler while concordant with at least one other, the cases in which the task rulers disagree among themselves.

Among reversals between two rules that are neither sign-reversed controls nor `random` nor below the random floor on LAMBADA, the three widest under LAMBADA are all on Qwen3-8B and all against `lex-rare`: `value-norm-hi`, 18.69 points [17.16, 20.22] with a loss gap of 0.0639 nat; `pos-uniform`, 18.55 points [17.00, 20.10] and 0.0801 nat; and `pos-edges`, 18.32 points and 0.0882 nat. Table 3 shows the third because it is the one whose loss-preferred rule follows a published scoring function; the other two are constructs of ours with no published counterpart. The widest over all pairs is the same decoder's `lex-rare` against its sign-reversed control `lex-frequent`: the loss prefers `lex-frequent` by 0.0246 nat [0.0173, 0.0318] while LAMBADA prefers `lex-rare` by 45 points [43, 46].

C.3 Robustness to the budget, the gate depth and the window

Every number in §4–§6 is read at one operating point: half the context kept, and the gate at the depth each decoder uses throughout. The disagreement could be a property of that point. Two sweeps on two decoders suggest it is not.

Figure 5 plots the twelve distinct cells, which range from −0.07 to +0.59 with a mean of +0.34; the body's WikiText–LAMBADA cells on the swept decoders, over the same twenty, +0.36 and +0.28, sit inside that range, one above and one below its mean. No cell approaches agreement: the largest is +0.59, at the highest keep fraction in the sweep. Neither sweep rises monotonically, as an artefact of the operating point would; the gate sweep falls, and on the budget axis the two decoders do not move together.

The gate sweep is the more informative of the two. On both decoders the correlation *falls* as the gate moves deeper, and on Qwen3-0.6B the deepest gate reaches −0.07, which is no relationship at all. A deeper gate is the setting in which the selection acts on more heavily processed representations, so if anything the two instruments agree less the further the intervention is from the raw context. We report the direction and claim no mechanism for it.

With two decoders and no intervals, the two sweeps bound the operating point rather than test it; the cells share rule sets and are not independent.

The window. The third axis is the context length, and this sweep was run on every decoder. At $C = 512$ with the same fraction kept ($k = 256$), all twenty-one rules were re-scored on the WikiText-103

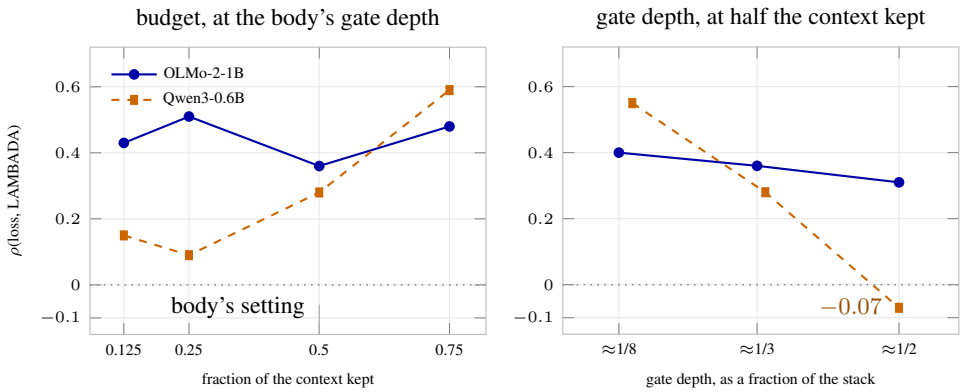

Figure 5: **The disagreement across operating points.** The loss-to-LAMBADA rank correlation over twenty rules, read away from the single operating point the body uses. **Left:** changing how much of the context is kept. Neither decoder moves monotonically and the two do not move together, which is not the pattern an artefact of the budget would produce. **Right:** moving the gate deeper into the stack. Here both fall, and on Qwen3-0.6B the deepest gate reaches -0.07, no relationship at all. Across the twelve distinct cells (the body's operating point appears in both sweeps) the correlation ranges from -0.07 to +0.59 with a mean of +0.34; the body's WikiText–LAMBADA cells on the swept decoders, over the same twenty, +0.36 and +0.28, sit inside that range, one above and one below its mean, and no cell approaches agreement. Values are in Table 4; with two decoders and no intervals, the two sweeps bound the operating point rather than test it.

Table 4: The WikiText-103 loss-to-LAMBADA rank correlation across operating points. Twenty rules, not twenty-one: the sweep's LAMBADA runs were launched without the flag that produces `attn-last8`, so that rule is absent from every swept cell. It is therefore dropped from the default cell too, and each row is read over the same twenty. That omission shifts the default point by 0.03–0.04 (+0.40 → +0.36 on OLMo-2-1B, +0.31 → +0.28 on Qwen3-0.6B, or +0.30 → +0.27 once the LAMBADA tie between `pos-edges` and `hidden-norm-hi` on Qwen3-0.6B is given average ranks); that is the size of the correction, not the size of the effect. Ties are broken in table order in this sweep, unlike the average-rank convention of the agreement cells.

decoder	$k/C = 0.125$	$k/C = 0.25$	$k/C = 0.5$	$k/C = 0.75$
OLMo-2-1B	+0.43	+0.51	+0.36	+0.48
Qwen3-0.6B	+0.15	+0.09	+0.28	+0.59

decoder	shallow gate	body's depth	deep gate
OLMo-2-1B	$L = 2$: +0.40	$L = 5$: +0.36	$L = 8$: +0.31
Qwen3-0.6B	$L = 4$: +0.55	$L = 9$: +0.28	$L = 14$: -0.07

loss and on both CBT subsets; at $C = 512$ with the body's absolute budget kept ($k = 64$, one eighth of the context), they were re-scored on the same three rulers on all six decoders. The same-fraction setting comes first:

	loss → CBT-NE		loss → CBT-CN		`lex-rare` at $C = 512$
decoder	$C = 128$	$C = 512$	$C = 128$	$C = 512$	loss / NE / CN rank
Qwen3-0.6B	+0.26	+0.45	+0.22	+0.33	21 / 5 / 1
Qwen3-1.7B	+0.32	+0.51	+0.24	+0.36	21 / 5 / 1
Qwen3-8B	+0.34	+0.43	+0.32	+0.40	21 / 1 / 1
Llama-3.2-1B	+0.32	+0.42	+0.35	+0.40	21 / 1 / 2
OLMo-2-1B	−0.09	+0.02	+0.04	+0.20	21 / 5 / 1
SmolLM2-1.7B	+0.33	+0.55	+0.32	+0.37	21 / 2 / 1
mean	+0.25	+0.40	+0.25	+0.34	

Lengthening the window at the same fraction raises the loss-to-CBT agreement from +0.25 to +0.40 (CBT-NE) and +0.34 (CBT-CN) on average, while the loss orderings at the two lengths agree with each other at +0.91 to +0.96: the crossing agreement stays far below the within-kind values of §4, and `lex-rare` is last on the loss on every decoder at $C = 512$ (19th–21st at $C = 128$) and first or second on CBT-CN. Keeping the body's absolute budget at the longer window is different. At 64 of 512 the crossing agreement rises to +0.71 (CBT-NE) and +0.67 (CBT-CN) on average over the six decoders: +0.69 to +0.84 (CBT-NE) and +0.54 to +0.83 (CBT-CN) on five of them, and +0.38 on both subsets on OLMo-2-1B. `lex-rare` now sits 19th or 20th on the loss, 14th to 17th on CBT-NE and 7th to 11th on CBT-CN:

	$C = 512$, $k = 64$		`lex-rare` at $C = 512$, $k = 64$
decoder	loss → CBT-NE	loss → CBT-CN	loss / NE / CN rank
Qwen3-0.6B	+0.78	+0.73	20 / 17 / 7
Qwen3-1.7B	+0.84	+0.77	20 / 17 / 7
Qwen3-8B	+0.80	+0.54	20 / 17 / 11
Llama-3.2-1B	+0.69	+0.74	20 / 14 / 9
OLMo-2-1B	+0.38	+0.38	20 / 16 / 7
SmolLM2-1.7B	+0.80	+0.83	19 / 17 / 11
mean	+0.71	+0.67	

At one eighth kept, the two instruments rank `lex-rare` far closer together than at either half-kept setting and the crossing agreement rises. The sweep records ranks alone, so it does not show what `lex-rare` retains of a CBT passage, nor how much of the rise is its own: the whole ordering moves with it, and the loss and CBT-NE leaders still differ on every decoder. Halving the budget once more, to one sixteenth of the window ($k = 32$ of 512), on five of the six decoders (Qwen3-0.6B, Qwen3-1.7B, Llama-3.2-1B, OLMo-2-1B and SmolLM2-1.7B; Qwen3-8B was not run at this setting) gives a similar picture: loss-to-CBT-NE is +0.58, +0.63, +0.52, +0.38 and +0.61, and loss-to-CBT-CN +0.81, +0.81, +0.83, +0.19 and +0.77, OLMo-2-1B again the outlier it is at one eighth, and `lex-rare` sits 19th or 20th on the loss, 12th to 18th on CBT-NE and 10th to 12th on CBT-CN on all five. The narrowing is therefore tied more to the fraction kept than to the window as such, which is why §3.2 holds that fraction fixed.

D THE TRAINED SCORER

This appendix carries what §6 rests on: the training recipe, the yoked-twin comparison, the second training corpus, the task-trained control, and the tables of where the scorer lands.

Scorer training, and the twin in numbers. The scorer is a small query-independent MLP (hidden width 256) producing one score per context position at the gate layer, trained with Adam at learning rates 10^{-4}, 3×10^{-4} and 10^{-3} for 1000 steps of 16 windows on 8000 WikiText training windows and evaluated every 100 steps on 512 validation windows; the reported checkpoint is the one with the lowest validation loss over the three rates and the eleven evaluations, and test loss is read on 1024 held-out windows. The twin runs the same loop on the same data seed with the gate's gradient replaced by Gaussian noise scaled to the gradient's row-mean magnitude, from an independent noise stream. Randomising only the quantity under test follows Adebayo et al. (2018), Hewitt & Liang (2019) and Zhang et al. (2025), none of which replaces a gradient while holding its scale fixed. Each trained arm is paired with its twin on the 1024 test windows; a win is a paired 95% interval on twin minus trained that lies entirely above zero, and a loss one that lies entirely below. Seeds 0–2 of every decoder are the arms ranked in §6 and Table 6 and the seeds whose arms are paired with their twins below (Qwen3-0.6B also has seeds 3 and 4 and two further noise draws, not used here: over all 65 pairings of its five trained seeds with its thirteen twin arms, each at its own picked rate, the tally is 62–2–1). On the WikiText loss, seeds 0–2 rank 7th, 2nd and 1st of 24 on Qwen3-0.6B, 7th, 6th and 9th on Qwen3-1.7B, 3rd, 2nd and 4th on Llama-3.2-1B, 6th, 5th and 3rd on OLMo-2-1B and 8th, 9th and 7th on SmolLM2-1.7B, the "1–9 of 24" of §6. Against the best hand-written rule on each decoder, the best seed's loss minus that rule's loss is -0.0037 nat on Qwen3-0.6B (`mix-attn32-last`), +0.0058 on Qwen3-1.7B (`mix-attn-last`), +0.0001 on Llama-3.2-1B (`attn-last8`), +0.0208 on OLMo-2-1B (`mix-attn32-last`), +0.0232 on SmolLM2-1.7B

(`pos-last`) and +0.0107 on Qwen3-8B (`mix-attn-last`). Qwen3-8B's arms were trained at 3×10^{-4} first and at the other two rates in a later sweep; its twin uses noise draw 0 where the other decoders use draw 1, a difference in the noise stream and nothing else. Pairs are matched on seed and learning rate: the trained arm at a (seed, rate), at its validation-picked step within that rate, against the twin at the same seed and rate at its own validation-picked step, nine pairs per decoder. Margins are twin minus trained, positive when the trained arm is better. The second tally keeps only the pairs in which the twin's picked step lies past its initialisation, since a twin that never improved at that rate shows only that training beats no training:

decoder	pairs	W–T–L	twin improved	median	range	trained loss	twin loss
Qwen3-0.6B	9	9–0–0	3–0–0 (3)	+0.055	+0.026 to +0.076	2.833 ± 0.008	2.883 ± 0.015
Qwen3-1.7B	9	9–0–0	3–0–0 (3)	+0.022	+0.009 to +0.051	2.492 ± 0.003	2.514 ± 0.011
Qwen3-8B	9	7–1–1	4–1–1 (6)	+0.011	−0.015 to +0.071	2.154 ± 0.004	2.175 ± 0.030
Llama-3.2-1B	9	9–0–0	8–0–0 (8)	+0.035	+0.009 to +0.057	2.533 ± 0.001	2.564 ± 0.016
OLMo-2-1B	9	9–0–0	8–0–0 (8)	+0.035	+0.014 to +0.042	2.480 ± 0.004	2.511 ± 0.013
SmolLM2-1.7B	9	4–3–2	1–1–2 (4)	+0.001	−0.022 to +0.067	2.384 ± 0.005	2.398 ± 0.030
all six	54	47–4–3	27–2–3 (32)				

Test losses are means over the nine arms of each side, with the between-arm standard deviation. The paired per-window test is what the counts rest on; the rerun-to-rerun variation of §3.4 (a standard deviation of 0.008 nat and a range of 0.024 over ten repeats of the Qwen3-0.6B configuration) is a separate scale, and the medians on Qwen3-1.7B, Qwen3-8B and SmolLM2-1.7B sit within that range. On Qwen3-8B, whose trained arm is missing from the stored per-window results the other ranks are read from, the training run's test pass (same windows and reference points as those results) puts seeds 0–2 at 2.1513, 2.1526 and 2.1517 nat, 5th, 7th and 6th of 24 against the twenty-one rules; its twin seeds land 1st, 6th and 12th, the first at 2.1364 nat, ahead of the best rule by 0.004 nat, about half the rerun standard deviation. The rerun figure itself comes from ten further seeds (10–19) of the Qwen3-0.6B recipe at 3×10^{-4}, whose picked-checkpoint test losses have mean 2.8332 nat, standard deviation 0.0077 and range 2.8192–2.8429, the picked step anywhere from 100 to 1000; §3.4 quotes the rounded figures.

A second training corpus. A scorer trained by the same recipe on FineWeb instead of WikiText, on Qwen3-0.6B with seeds 0–2, optimises its own loss less well than the WikiText-trained scorer does: at the validation-picked learning rate its FineWeb test loss is 3.2648, 3.2618 and 3.2648 nat, 10th, 9th and 11th of 24 against the twenty-one rules' FineWeb losses (`lex-rare`, 3.2506, is 7th and `random`, 3.2669, 12th; the training run's test pass shares its windows and reference points with the stored per-window results). On the task rulers it ranks 3rd–5th of 24 on LAMBADA (39.90–40.27%), 8th–10th on SQuAD (10.51–10.94%, against 15.75–17.70% and ranks 1–3 for the WikiText-trained seeds), 2nd–7th on CBT-NE (70.83–78.42%, `lex-rare` first at 78.72%) and 6th–8th on CBT-CN (79.54–80.42%, `lex-rare` first at 84.83%); it leads on no ruler. The SQuAD lead of §6 therefore belongs, on Qwen3-0.6B, to the WikiText-trained scorer and not to the recipe (one reading: SQuAD's passages, like WikiText-103, are Wikipedia text), and the rest is the pattern of §6 at lower ranks: a selector that optimises its objective only modestly and whose place on each task is set by that task rather than by its objective.

The three losses to the twin, and the caveats that go with them. None of the three involves an untrained checkpoint: on Qwen3-8B, seed 0 at 3×10^{-4} (step 400) loses to its twin at the same rate (step 300) by 0.015 nat [0.011, 0.019], and that twin is the best arm of any kind on the decoder's loss; on SmolLM2-1.7B, seed 0 loses at 3×10^{-4} (step 900 against 200) by 0.009 [0.005, 0.013] and at 10^{-3} (step 900 against 1000) by 0.022 [0.017, 0.026], a pair in which the twin improved its own validation loss for the full run. Of the 22 pairs whose twin never improved on its initialisation, 20 are wins for the trained arm and 2 are ties (both on SmolLM2-1.7B), so counting them raises the tally and changes nothing else. The ranks in Table 6 are point estimates, so "leads" means "is first in the table"; the paired bootstrap after the table puts intervals on them. The 54 pairs share a small number of underlying runs and are not independent, so the counts are descriptive tallies and not hypothesis tests. Rerun-to-rerun variation (a standard deviation of 0.008 nat, §3.4) is not combined with the resampling intervals. Finally, the trained scorer is kept out of the twenty-one-rule tables

throughout: the family is a deterministic control set built to cover axes, the scorer is neither, and its correlated seeds would enter a rank correlation as several non-independent rows.

The task-trained control. The control keeps the same scorer, gate, estimator, learning-rate grid, step budget and validation-picked checkpoint as the WikiText-trained arm, with the objective replaced by the task's own loss, on Qwen3-0.6B with seeds 0–2. For LAMBADA the objective is the gold-word loss on the first 2576 passages of the test set in dataset order, the last 256 of them the validation slice (2320 training passages), and the reading is on the remaining 2577, which training never sees; for SQuAD it is the answer loss on the first 8000 usable questions of the training split with the next 512 as validation, and the reading is on the dev subset, as for every other arm. The picked rates and steps are 10^{-3} at steps 300 and 400 and 3×10^{-4} at step 100 on LAMBADA, and 3×10^{-4} at step 1000 and 10^{-3} at steps 700 and 500 on SQuAD. On the held-out examples the three seeds score 38.92, 39.97 and 38.46% on LAMBADA, 4th–6th of 24 among the rules and their own seeds and 7th–9th of 27 once the WikiText-trained seeds are added (those score 40.63–43.31% and rank 2nd, 4th and 5th), and 18.68, 17.55 and 16.84% on SQuAD, 1st–3rd of 24 and 1st, 3rd and 4th of 27 (the WikiText-trained seeds 2nd, 5th and 6th). Table 5 carries the medians. The same recipe on Llama-3.2-1B ($L = 5$, seeds 0–2; picked rates and steps 3×10^{-4} at 300, 10^{-4} at 300 and 3×10^{-4} at 200 on LAMBADA, 10^{-4} at 500 and 3×10^{-4} at steps 100 and 100 on SQuAD) scores 44.63, 45.44 and 44.78% on the held-out LAMBADA half, 8th–10th of 27 and 5th–7th of 24, against 48.86–49.40% for the WikiText-trained seeds (3rd–5th) and 52.04% for `attn-last1`, and 13.67, 12.60 and 12.98% on SQuAD, 1st–3rd of 27 and of 24, against 11.83–12.14% for the WikiText-trained seeds (4th–7th) and 12.06% for `lex-rare`, the best hand-written rule there. The two decoders agree in sign: training on the task's own loss gains about a point on SQuAD and loses two to five on LAMBADA.

Table 5: The transfer test on Qwen3-0.6B, and the task-trained control repeated on Llama-3.2-1B (last two rows). Accuracy (%) of the random floor, the best hand-written rule, the WikiText-loss-trained scorer and the task-trained control (the two trained columns each the median of seeds 0–2) on each task ruler, read on the same examples: LAMBADA on the 2577 passages the control never trains on, SQuAD on the dev subset, CBT on the answerable examples (no control was trained for CBT); the ranks on every decoder (LAMBADA on the full test set) are in Figure 4.

task ruler	n	floor	best rule	loss-trained	task-trained
LAMBADA	2577	26.08	`attn-last1` 44.70	40.98	38.92
SQuAD	2559	7.42	`pos-first` 13.99	16.14	17.55
CBT-NE	672	57.74	`lex-rare` 78.72	79.61	–
CBT-CN	567	73.37	`lex-rare` 84.83	82.54	–
LAMBADA, Llama-3.2-1B	2577	30.11	`attn-last1` 52.04	49.36	44.78
SQuAD, Llama-3.2-1B	2604	7.95	`lex-rare` 12.06	11.94	12.98

Table 6: Where the trained scorer lands among the 24 arms. Ranks are over seeds 0–2 and the six decoders; "leads" counts decoders where a trained seed is first.

task ruler	trained scorer	leads	leader elsewhere
SQuAD exact match	rank 1–12	**4 of 6**	`pos-first` on Qwen3-8B and SmolLM2-1.7B
CBT-NE accuracy	rank 1–7	1 of 6	`lex-rare` on the other five (on Llama-3.2-1B tied with seed 0 at 83.04%)
CBT-CN accuracy	rank 1–20	1 of 6	`lex-rare` on three, `key-norm-lo` on Qwen3-1.7B, `lex-nonstop` on Qwen3-8B
LAMBADA accuracy	rank 2–13	**0 of 6**	`attn-last1` on five, `attn-last8` on the sixth

Intervals on the ranks. On each task ruler and decoder the per-example hits of all 24 arms were resampled together (paired bootstrap, $B = 10\,000$, as in §3.4) and the ranks recomputed on every resample, a tie sharing the better rank. The band survives: a seed's probability of a rank inside the loss's top nine is at least 0.95 in 56 of the 72 cells (17 of 18 on LAMBADA, 18 on CBT-NE, 7 on CBT-CN, 14 on SQuAD), and the sixteen cells that fail are mostly the ones Figure 4 shows at or beyond the band's edge: eleven on CBT-CN, Qwen3-8B's three on SQuAD (0.26 to 0.50), and one seed each of SmolLM2-1.7B on SQuAD and on LAMBADA. The leads soften: a trained seed

is first with probability at least 0.95 on two decoders for SQuAD (Qwen3-0.6B 1.00, OLMo-2-1B 0.96) and on none for the other three rulers, and `lex-rare` leads CBT-NE with probability 0.98 on Qwen3-8B. On the paired differences, `lex-rare` is above all three seeds at 95% on two decoders for CBT-NE (Qwen3-8B, SmolLM2-1.7B), three for CBT-CN (Qwen3-1.7B, Llama-3.2-1B, SmolLM2-1.7B), two for LAMBADA (Qwen3-1.7B and Qwen3-8B, second to `attn-last1` on both) and one for SQuAD (Qwen3-8B), and below all three on three decoders for SQuAD (Qwen3-0.6B, Qwen3-1.7B, OLMo-2-1B) and one for LAMBADA (OLMo-2-1B); in every other cell at least one of the three differences crosses zero. Nothing in Table 6 or Figure 4 changes; what the bootstrap adds is that the band claim is confident and the lead counts are not.

E When the decoder is allowed to adapt

The decoder is frozen everywhere else in this paper. That is the setting in which the audited practice operates, but it is also the most obvious assumption to question: if a decoder were trained under a fixed selection pattern, would the loss rank the rules differently? It would. The ordering is not a property of the rule.

Setup. With the decoder's own weights frozen, we attach LoRA adapters (rank 16, $\alpha = 32$) to the attention projections of the layers at or above the decoder's gate layer, and train for 1000 steps on the WikiText loss with one rule's mask applied throughout. Only rules whose mask does not read the decoder are eligible (the four position rules, the four lexical rules and `random`; A.2), and we take four of them: `lex-rare`, `lex-frequent`, `pos-last` and `random`. A mask that reads the decoder at layer L (the key and value norms) could move as the decoder adapts, and then "the decoder adapted" and "the pattern moved" could not be separated. The rules that read attention or the hidden state look below L and so stay fixed here, but would move under an adaptation that reaches lower layers; a mask that does not read the decoder stays fixed under either. We use three decoders and one seed per arm (an arm here is one rule on one decoder), with the checkpoint chosen on 512 validation windows.

Adaptation reorders the rules, and by point estimate the resulting order is the same on all three decoders.

	before adaptation	after adaptation
Llama-3.2-1B	`pos-last` > `random` > `lex-frequent` > **`lex-rare`**	**`lex-rare`** > `pos-last` > `random` > `lex-frequent`
OLMo-2-1B	`pos-last` > `random` > `lex-frequent` > **`lex-rare`**	**`lex-rare`** > `pos-last` > `random` > `lex-frequent`
Qwen3-0.6B	`pos-last` > `random` > **`lex-rare`** > `lex-frequent`	**`lex-rare`** > `pos-last` > `random` > `lex-frequent`

`lex-rare` moves, by point estimate, from last to first on Llama-3.2-1B and OLMo-2-1B and from third to first on Qwen3-0.6B. The other three keep their order relative to one another; on Llama-3.2-1B and OLMo-2-1B each drops one place, and on Qwen3-0.6B `pos-last` and `random` drop one place while `lex-frequent` stays fourth.

The cleaner statement is about the size of the gains, and it needs no reference arm. The adaptation gain is about the same for three of the four rules and substantially larger for `lex-rare`:

	range over arms	mean	arms
`lex-rare`	0.3771 to 0.4010	**0.3851**	3
the other three rules	0.2427 to 0.2716	**0.2550**	9

The nine non-`lex-rare` arms spread over 0.029 nat while `lex-rare` sits 0.130 nat above their mean. The gap is four and a half times the spread, and it is visible without knowing which rule is which.

A reference arm is needed for anything phrased as a fraction. Every adapted arm ends below the *unadapted* full-context loss, because 1000 LoRA steps on WikiText improve the decoder whether

or not anything is dropped. Measuring against the unadapted loss therefore yields fractions above one and is uninformative. We ran the same script once more on Llama-3.2-1B with the budget set to the whole window, so that no token is ever dropped and the only change is the training. On the 512 validation windows these runs report on, this moves full-context loss from 2.4456 to 2.2418 (rather than from the 2.4572 that C.1 reports on the test windows). That move is the generic gain; the adapted full-context loss it reaches, 2.2418, is the reference, and each rule's fraction is its own adaptation gain divided by the distance from its unadapted loss to that reference. Against it, on Llama-3.2-1B, `lex-rare` closes 92.1% of its gap, `pos-last` 83.9%, `random` 79.6% and `lex-frequent` 64.4%. As a sanity check, all four adapted arms remain above the adapted full-context reference.

What this does and does not license. The result belongs to this appendix and not to the body. The post-adaptation margin between `lex-rare` and `pos-last` is 0.0157, 0.0056 and 0.0088 nat, of the order of the 0.008 nat rerun-to-rerun standard deviation of §3.4 (one twice it, one below it, one at it), so the claim these margins support is that `lex-rare` has drawn level with `pos-last`, not that it is now first. The adaptation gain that moved it, 0.38 to 0.40 nat against 0.24 to 0.27 nat for the other rules (table above), is far larger than that variation. There is one seed per arm. Checkpoint selection and reporting share the same 512 validation windows, so the levels are optimistic, though the four rules share that bias. The reference arm was run on one decoder. This is the loss ordering only: whether the *task* ordering moves under adaptation is a question these runs do not answer.

It is worth noting which rule this is. `lex-rare` ranks 22nd, 23rd, 23rd, 24th, 24th and 24th of 24 on the WikiText loss (§6), is first by point estimate on CBT-NE on five of the six decoders (tied on one) and on CBT-CN on three, and is the rule whose adaptation gain is largest on all three decoders here. Two unrelated sources of evidence, the task rulers and adaptation, single out the same rule as one that the loss ranks wrongly.

F THE KEY-TOKEN GATE GRID

The gate of §7 scores a token's need as

$$\text{need} = \text{NLL}(\text{protected position} + \text{last } K-1 \text{ context positions}) - \text{NLL}(\text{full context}),$$

and keeps the tokens in the top decile of need.

Table 7: Rank of the last-k rule (`pos-last`) under the key-token gate. All six decoders, all twenty-one rules, so every cell has the same denominator. Ranges are over decoders; bold marks the cells where the rule leaves the bottom five on every decoder. The gate keeps the tokens in the top decile of need at that K.

budget	$K = 64$	$K = 32$	$K = 16$
$k = 64$	20–21 of 21	12–17 of 21	**2–15 of 21**
$k = 32$	17–20 of 21	19–20 of 21	**9–14 of 21**
$k = 16$	17–20 of 21	19–20 of 21	19–20 of 21

Read the bottom row across: the gate's window falls from 64 to 16, a factor of four, and the rule never leaves the bottom five, because K stays at or above k throughout. Read the top row across: the same change lifts the rule out of the bottom five, to as high as second on one decoder and no lower than 15th on the rest, because K crosses below k. The middle row lies between the two and leaves the bottom five only in its last cell, the only cell in that row where $K < k$. The rank tracks the sign of $K - k$, not the value of K; a gate that measured merit would not make the rank flip with that sign.

The sign-reversed partner that keeps the *earliest* tokens (`pos-first`) moves in the opposite direction over the top row, from 1st to 3rd–5th to 7th–15th. A noise artefact would not produce that opposite movement.

Does the gate repair the proxy? At the body's budget, $k = 64$, we correlated the rule ordering under the gated WikiText loss with the ordering under each of the four task rulers, as Figure 1 does

for the plain loss (Spearman with average ranks, mean over the six decoders). Need labels exist for the WikiText test windows only, so this is the WikiText row of the matrix under the gate.

WikiText loss	LAMBADA	CBT-NE	CBT-CN	SQuAD	mean
plain	+0.42	+0.25	+0.25	+0.11	+0.26
gated, $K = 64$	+0.02	+0.07	−0.01	+0.14	+0.06
gated, $K = 32$	+0.15	+0.22	+0.07	+0.22	+0.17
gated, $K = 16$	+0.42	+0.39	+0.27	+0.26	+0.34

On average, the gate at $K = k$ removes most of the agreement the plain loss had, and at $K = 32$ it still agrees less (+0.17 against +0.26); only at $K = 16$ does it add a little, mostly on CBT-NE and SQuAD, though SQuAD gains at every K by point estimate (+0.14, +0.22 and +0.26 against +0.11), within resampling noise at $K = 64$ (+0.03 [−0.06, +0.09]). The repair helps only where the gate's window is well below the budget, and then modestly. Resampling the loss windows and each task ruler's units together for all twenty-one rules ($B = 4000$) puts intervals on the four-ruler means: gated minus plain is +0.08 [+0.04, +0.09] at $K = 16$, positive in every draw and carried by CBT-NE and SQuAD (+0.15 and +0.16, from the unrounded values) with LAMBADA unchanged and CBT-CN nearly so; it is −0.09 [−0.14, −0.07] at $K = 32$ and −0.20 [−0.25, −0.17] at $K = k = 64$. The gated loss is nearly as reliable as the plain one, so the lower agreement at $K \geq 32$ is not attenuation: over 200 random splits of the 1024 test windows, with the rules ranked on each half by the pooled gated mean, the Spearman–Brown reliability (§4) averages 0.99 for the decile gate at all three K (against 0.996 for the plain loss, B.3), 0.98 to 0.99 with LongPPL's filter and 0.96 to 0.98 under its thresholds, with no decoder below 0.92, and correcting each crossing cell for both reliabilities moves the means above by at most 0.01.

Differences from LongPPL. LongPPL was proposed to compare models, not rules, and four choices differ here. Need is computed by the decoder under test rather than by a separate, stronger evaluator model, because the rules are compared within one decoder. LongPPL's second filter, which drops the tokens the full context also mispredicts, is not applied. Our short context keeps the first position (the sink) beside the last $K - 1$ context positions, fixed at the end of the context with the scored continuation always visible, where LongPPL's Eq. 2 re-encodes the last K tokens before each token alone (its implementation shares one window start across each block of 1024 tokens). And the top decile of need replaces LongPPL's fixed thresholds, which were set for documents of tens of thousands of tokens and evaluators of 8B to 123B parameters (72B by default) and do not carry over to $C = 128$, whereas a fixed decile keeps the number of gated tokens comparable across decoders and windows. LongPPL's short window of 4096 tokens on documents of up to 32k tokens is one eighth of the longest documents and a larger fraction of shorter ones; one eighth is the ratio of $K = 16$ to $C = 128$ here, the one setting in which the unfiltered gate helps. Omitting the filter is not what produces the deficits: applying it here (keeping only gated tokens whose full-context log probability exceeds −2, LongPPL's setting, which leaves 3.5 to 5% of the tokens) removes the gain rather than the deficits, with the gated loss matching the plain loss at $K = 16$ (+0.28, difference +0.02 [−0.01, +0.04]) and agreeing less at $K = 32$ and $K = 64$ (differences −0.11 and −0.25); LongPPL's own thresholds on both scores (need above 2 nat, log probability above −2), which keep 0.6 to 2.9% of the tokens and leave many windows with no gated token at all, give the same pattern (0.00, −0.14, −0.25).

G Reporting

G.1 What to report when the proxy is a loss over selection rules

The audit of §9 says whether to trust a comparison. The seven items below say how to report it; each is something this paper had to do, and the sections that show why are named beside it.

1. **Name the instrument and the corpus.** "Held-out loss" underspecifies a ranking whose transfer varies by a factor of two with the corpus (§4).
2. **Put an interval on every difference that carries an ordering**, on the task side as well as the loss side (§5).

3. **Bracket every table** with a random floor and a full-context ceiling (§3.1, B.2).
4. **Read a task benchmark with its own metric**, and if it is also read with a second metric, report the agreement between the two readings (§3.3, §4).
5. **For a learned selector, report the yoked noise-gradient twin**, not merely a comparison against random selection (§6).
6. **If the loss is gated on key tokens, report K and k separately**, and hold K fixed, well below k, when comparing rules at different budgets (§7).
7. **Name the proxy's free choices and report agreement across them.** Where a use of the proxy leaves a choice open (the corpus, the context length), the agreement across that choice is the proxy's reliability for that use, and the split-half figure is not (§4, B.3, C.3).

H How this subfield's evaluation has been examined

Yuan et al. (2024) noted that no prior work had comprehensively benchmarked long-context compression methods, cache eviction among them, in an aligned environment; Bhaskar et al. (2025) that the cost axis was not commensurable either; Chen et al. (2026) evaluated the same five methods our rules are modelled on and found that under compression some instructions in a multi-instruction prompt degrade much faster than others, and that multi-instruction prompts tend to degrade sooner than single-instruction ones; and Wang et al. (2026) evicted uniformly at random *within each attention head* during long reasoning generations and matched the strongest scored evictor, attributing the match to the prompt being kept and to redundancy in the reasoning trace at two levels, in the text and across attention heads. On the last of these four, our `random` rule, a uniformly drawn subset, never ranks better than seventh of twenty-one across all seven rulers and six decoders and sits between eleventh and sixteenth on every task ruler, so the selection signal is informative here. This is not a disagreement with their result: our setting has no generated trace, selection drops tokens from the context itself with only its first position protected (A.3), and our mask is shared across heads (A.5), so every head from the gate layer L onward drops the same positions (§3.2) and none of the three things their account rests on, a kept prompt and the two levels of redundancy, holds here.