Retrieved but Not Used: Access–Use Dissociations in Long-Context Language Models

Anonymous authors
Paper under double-blind review

Abstract

Long-context evaluations ask whether a language model can find information far back in its input. We ask a narrower question: once a value can be found, can the model use it? We construct partially evaluated programs in which a probe value is defined once and consumed once after a controlled number of intervening lines, with total length and live state held fixed, and we compare three queries at the same position: copy the value (retrieve), add a one-digit literal to it (compute), or add a one-digit local variable to it (combine). Across four Qwen3 models (4B to 32B) at 28K tokens, retrieve stays at 88.5 to 99% after 1,024 lines of separation while compute falls to 37.5 to 76% and combine to 0.5 to 43.5%. On identical programs the use failures sit inside the retrieval successes: Qwen3-8B copies the value on all 200 programs on which it then fails compute (100) or combine (198), and the paired excess of combine over retrieve is 47.5 to 86.5 points across Qwen3-8B, Qwen3-32B and Gemma-3-12B. Context structure matters in a scale-dependent way. At a fixed distance, growing the context from 3.7K to 28K tokens shows no detectable load effect on retrieve or compute at 8B, 14B or 32B, while combine falls from 95 to 66% at 32B; the far-distance collapse occurs at constant length. Across families the dissociation is a boundary rather than a universal: Llama-3.1-8B shows the one-operation gap (19 points) but not the two-variable collapse, and Gemma-3-12B shows both (37 and 61 points) where its position-dependent retrieval holds. A 2,048-token generation budget returns compute to 86 to 99% on the same items with or without an instruction to reason, while the question format at 8 or 64 tokens answers nothing; what the budget buys is not identified. A pre-specified decision rule based on a log-odds interaction returned no-go; under ceiling-level retrieval that estimator tests a different scale from the verbal hypothesis, so we make no confirmatory claim from it and label the probability-scale analysis post hoc. All evidence comes from one synthetic task.

1 Introduction

Access and use are different abilities. A language model that can locate a value in a long context may still fail to apply a single operation to it in the same forward pass, and long-context benchmarks rarely separate the two: needle-in-a-haystack tests measure access alone (Hsieh et al., 2024), composite benchmarks embed several facts in long documents and score the end task (Kuratov et al., 2024; Bai et al., 2025; Yen et al., 2025), and length-sensitivity studies pad a fixed task (Levy et al., 2024; Du et al., 2025). In each design the variables that could explain a drop move together: the distance between a piece of information and its point of use, the total context length, the amount of material preceding the information, its absolute position, the difficulty of the operation, and the scale of the model.

We vary one of these variables, the dependency distance between a value and its use, with total length and live state held fixed, and measure its effect separately for retrieval and for computation; further arms score the three queries on identical programs, vary distance and preceding material independently, vary total context at a fixed distance, and separate the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

prompt format of a scratchpad from its token budget. Each item is a partially evaluated program, a chain of modular-arithmetic assignments in which every line displays its result, and one probe value v is defined once and used once, d filler lines later. The final line asks, at the same position, for v itself (retrieve), $(v + k)$ mod 9973 for a one-digit literal k (compute), or $(v+w)$ mod 9973 for a one-digit local variable w defined on the preceding line (combine); any difference between the three at a given d is a difference in what the model must do with a value it has already located.

We report six findings, each scoped to the models and the synthetic task on which it was measured. (i) At 28K tokens and $d = 1024$, Qwen3-4B to 32B keep retrieve at 88.5 to 99% while compute falls to 37.5 to 76% and combine to 0.5 to 43.5%; the excess loss over retrieve is 41.5 to 81.5 points for combine at every scale and about 55 (4B, 8B) or 13.5 (14B, 32B) for compute. (ii) On identical programs the use failures nest inside the retrieval successes, exactly at 8B, nearly so at 32B, and in Gemma-3-12B except where its retrieval dips at $d = 128$; paired and unpaired excesses agree. (iii) Scale removes the combine gap at 3.7K tokens by 32B but only halves it at 28K. (iv) Distance and preceding load occupy different regimes: at 8B and 14B most of the loss follows the material between the value and its use, at 4B and 32B distance costs accuracy mainly when preceding load is also large, and at 32B a fixed 128-line distance loses 29 points of combine as the context grows from 3.7K to 28K tokens. (v) Outside Qwen3, Llama-3.1-8B reproduces the one-operation gap (19 points) but not the collapse, Gemma-3-12B reproduces both where its retrieval holds, and Mistral-7B fails the retrieval check. (vi) A 2,048-token budget returns compute to 86 to 99% with or without an instruction to reason, whereas the question format at 8 to 256 tokens answers little or nothing: the gain follows the budget.

A further contribution is methodological. Our pre-specified decision rule, fixed in code before the main data were collected, tested the retrieval-computation contrast as a condition $\times$ log-distance interaction in a logistic model and returned no-go. In simulations with ceiling-level retrieval, this log-odds interaction is wrong-signed with high probability even when the probability-scale hypothesis is true. We report it unchanged, make no confirmatory claim from it, and label the probability-scale analysis post hoc; combine is the robust headline within Qwen3 and compute at 14B and 32B a modest effect.

2 Related work

Length as the manipulated variable. Levy et al. (2024) pad a fixed reasoning problem and find that accuracy falls wherever the padding sits; Du et al. (2025) show that length degrades performance even when every piece of evidence is retrieved, including with irrelevant tokens masked from attention; Liu et al. (2024) show sensitivity to the position of the relevant document. These designs vary total length or position with the task fixed, so their loss combines dependency distance, preceding material and position; our arms vary distance separately.

Reasoning over retrieved content. RULER's variable tracking (Hsieh et al., 2024), BABILong (Kuratov et al., 2024), Michelangelo (Vodrahalli et al., 2024), LongBioBench (Yang et al., 2025b) and Verbose ListOps (Pan & Williams, 2025) find composite reasoning harder than retrieval in long contexts, but in each the number of facts, operations and context change together with the task; our three queries differ only in what is done with one value at one position. Lost-in-Distance (Firooz et al., 2024), Needle Threading (Roberts et al., 2025) and NoLiMa (Modarressi et al., 2025) concern the separation between relevant items or between query and needle; we measure the distance between one value and its use, with a retrieval control at the same distance.

Scratchpads and mechanism. Scratchpads and chain-of-thought prompting improve multi-step computation (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022); long-horizon execution (Sinha et al., 2026) and compositional tasks (Dziri et al., 2023) measure degradation accumulated over many steps, whereas the combine collapse here is a single-step failure that grows with distance at fixed length. Retrieval heads (Wu et al., 2025), in-context binding (Feng & Steinhardt, 2024) and representational collapse with length (Barbero et al.,

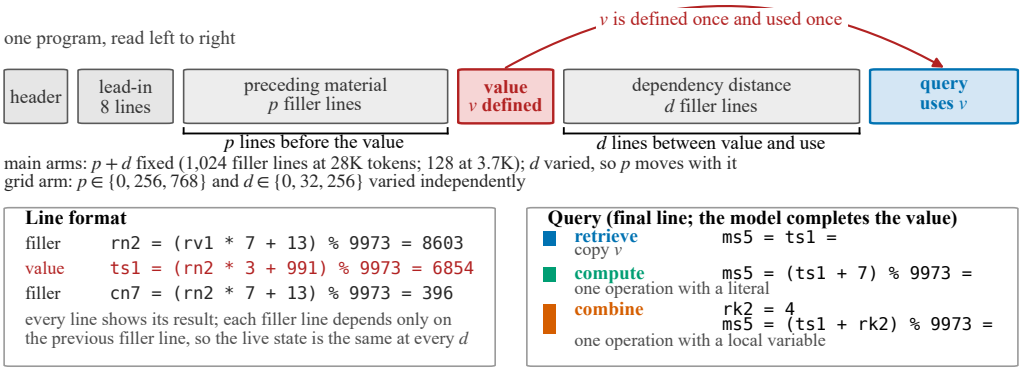

Figure 1: One item and its manipulations. A probe value v is defined once, after a lead-in and p filler lines, and used once, d filler lines later (the dependency distance). Every line shows its result and each filler line depends only on the previous filler line, so the live state is the same at every d. In the main arms $p + d$ is fixed; in the grid and load arms p and d vary independently. The final line copies v (retrieve), adds a literal (compute) or adds a local variable defined on the preceding line (combine).

2024) are candidate mechanisms; our results are behavioral and do not adjudicate among them.

3 Experimental design

Programs and queries. Each item is a partially evaluated program (Figure 1) whose lines are assignments with their computed values; variable names are opaque (two letters and a digit). An initial value $x_0 \sim \text{Uniform}\{1000, \ldots, 9972\}$ starts a chain $x_{t+1} = (7x_t + 13) \bmod 9973$. After 8 lead-in lines and p further filler lines, the probe line defines $v = (3x_t + 991) \bmod 9973$; the chain then continues from x_t for d more lines, so v is defined once and not consumed until the final line. Every intermediate value is displayed and only the final line is generated, so no error propagates. The final line introduces a new variable and asks for v (retrieve), for $(v + k) \bmod 9973$ with a literal $k \in \{2, \ldots, 9\}$ (compute), or for $(v + w) \bmod 9973$ with $w \in \{2, \ldots, 9\}$ defined on the line immediately before (combine), which adds a second, local lookup; the addition and the range of the addend are those of compute, so combine tests use under an additional local dependency rather than harder arithmetic. One-digit operands keep the arithmetic at ceiling (a four-digit pilot scored 20% on combine at $d = 0$ and was replaced before any main run; Appendix B).

Arms and outcome. Let p be the number of filler lines before the probe, d the number between the probe and the query, and $L = p + d$. In the long arm $L = 1024$ (27,931 to 28,101 Qwen3 tokens) and $d \in \{0, 2, 8, 32, 128, 512, 1024\}$; in the short arm $L = 128$ (3,730 to 3,796 tokens) and $d \in \{0, 2, 8, 32, 128\}$. Within an arm, total length and the live set are constant and the median token distance at each shared d agrees across arms to within one token. Because $p = L - d$, absolute position is a function of d within an arm: at the largest distance the value sits at line 9, where primacy effects can favor access (Liu et al., 2024), so the main arms cannot separate distance from position. The grid arm varies $p \in \{0, 256, 768\}$ and $d \in \{0, 32, 256\}$ independently (80 items per cell), and the load arm fixes $d = 128$ while $p \in \{0, 128, 384, 896\}$, so one dependency distance is embedded in programs of 3.7K, 7K, 14K and 28K tokens (100 items per cell, drawn independently at each p). In the paired arm (Qwen3-8B, Qwen3-32B and Gemma-3-12B), 600 programs with $L = 1024$ and $d \in \{0, 128, 1024\}$ (200 per distance) are each queried in all three ways, so the outcomes can be cross-tabulated program by program. Writing $A_o(d)$ for the accuracy of query type $o \in \{R, C, B\}$ at distance d, our summary statistic is the excess loss of a use condition over retrieval across the tested range,

$$E_o \;=\; \big[A_o(0) - A_o(d_{\max})\big] \;-\; \big[A_R(0) - A_R(d_{\max})\big], \qquad o \in \{C, B\}, \tag{1}$$

the points a condition loses beyond what retrieval loses over the same span. It is a descriptive difference of differences in probability, not a factorization into retrieval and use, and assumes neither their independence nor that use requires retrieval. In the main arms it is a contrast of cell means over distinct items; in the paired arm it is computed on the same programs.

Models and decoding. We evaluate the post-trained Qwen3-4B, 8B, 14B and 32B checkpoints (Yang et al., 2025a) as raw completions without a chat template, so thinking mode is never invoked; all results use greedy decoding in bfloat16 with 8 new tokens, and the answer is the first integer on the first generated line, which completes the query line; the rule was fixed before the main runs and scores an output with no integer there, such as an empty first line, as incorrect whatever follows, so an error need not reflect a failure of use (Section 4.4). Long-arm and short-arm cells contain 200 items, with item identities shared across models (one generator, seed 0); the 14B and 32B long arms were run at 100 items per cell first and extended with the remaining 100 (Appendix B). DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), with a 4,096-token training window, reaches 0% retrieve at 28K tokens by $d = 512$, fails the retrieval check and appears only in Appendix F. The 4B and 8B main arms ran on an RTX 4090 with transformers (Wolf et al., 2020); everything else ran on an A100 with vLLM 0.29 (Kwon et al., 2023); a 900-item replication of the 8B long arm under the second stack agreed with the first on 98.4% of items (Appendix I).

Other families. The same long and short arms, items, prompts, decoding and parser were run on Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Gemma-3-12B-it (Gemma Team, 2025). A model enters only if it passes an 80-item validity gate before its full run (prompts within the context window; retrieve at $d = 1024$ at least 70%; retrieve and compute at $d = 0$ at least 90 and 80%; at most 10% unparsed). Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) failed it (retrieve 40% at $d = 1024$; combine 20% at $d = 0$) and was not run (Appendix J). Line distance is the manipulated variable; the same programs are 22.9K tokens under Llama's tokenizer and 27.9K under Gemma's.

Scratchpad arm and format × budget ablation. In the scratchpad arm the final line is rewritten as a question followed by an instruction to work step by step and to end with ANSWER: <number>; generation is capped at 2,048 tokens and an item is correct only if a terminal ANSWER matches (first run: $d \in \{0, 1024\}$, 50 per cell). Because this changes the prompt format and the token budget at once, an ablation on 100 compute and 100 combine items at $d = 1024$ (Qwen3-8B and 32B) crosses the completion format and the question form under an answer-only or a step-by-step instruction with budgets of 8, 64, 256 and 2,048 tokens. We report unconditional accuracy, the fraction of outputs with no answer within the budget, and accuracy among answered outputs; a first run with a 384-token budget truncated 18 to 99% of outputs (Appendix H).

Pre-specified analysis. Before any long-arm data were collected, four criteria were fixed in a script (Appendix C): a near-far gap of at least 8 points on at least two models; a monotone dose-response; a negative condition × log-distance interaction in a logistic model, significant at 0.05 on at least two models, as the decisive criterion; and constant length and live set by construction. The plan was fixed in code but not registered, so we say pre-specified rather than preregistered (Nosek et al., 2018). The probability-scale analysis of Section 5, the grid and load regressions and strict scratchpad scoring lie outside that plan and are labelled post hoc; the paired, load and ablation arms and their analyses were fixed before their own data existed but after the main results were known.

4 Results

4.1 Does retrieval survive when use fails?

At 28K tokens, retrieval survives 1,024 lines of separation and computation on the retrieved value does not (Figure 2, Table 1). retrieve accuracy is at least 99% through $d = 32$ on every model and 88.5 to 99% at $d = 1024$, while compute, at 97 to 99.5% at $d = 0$, falls to 37.5 to 76% and combine to 0.5 to 43.5%. The excess of Equation 1 is 41.5 to 81.5 points for

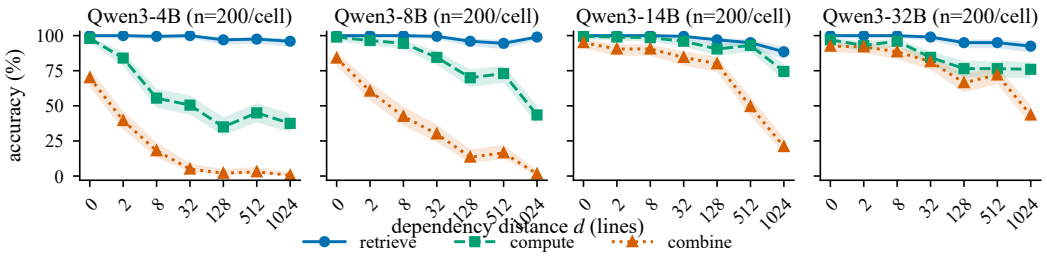

Figure 2: Long arm (28K tokens; $n = 200$ per cell): accuracy by dependency distance for each query type; shaded bands are Wilson 95% intervals (Wilson, 1927). Retrieval stays near ceiling at every distance, while compute and combine fall with distance at every scale, most steeply for the two smaller models.

Table 1: Long arm (28K tokens; $n = 200$ per cell): accuracy (%) at $d = 0 / 1024$ and the excess loss over retrieval of Equation 1 (points; item-level percentile bootstrap 95% CI, 5,000 resamples of each cell). Last column: short arm (3.7K tokens, $n = 200$), combine at $d = 0 / 128$. Below the rule, the two other families that passed the gate; Gemma's retrieval is off ceiling at intermediate distances (Section 4.4).

	accuracy, $d = 0 / 1024$			excess E_o [95% CI]		short arm
model	retrieve	compute	combine	compute	combine	combine, $d = 0 / 128$
Qwen3-4B	100.0 / 96.0	98.0 / 37.5	70.0 / 0.5	56.5 [49.0, 64.0]	65.5 [58.0, 72.0]	88.0 / 29.5
Qwen3-8B	100.0 / 99.0	99.0 / 43.5	84.0 / 1.5	54.5 [47.0, 61.5]	81.5 [76.0, 86.5]	90.0 / 18.5
Qwen3-14B	100.0 / 88.5	99.5 / 74.5	95.0 / 21.0	13.5 [6.0, 21.0]	62.5 [54.5, 70.5]	95.5 / 75.5
Qwen3-32B	100.0 / 92.5	97.0 / 76.0	92.5 / 43.5	13.5 [6.0, 21.0]	41.5 [32.5, 50.0]	95.0 / 95.5
Llama-3.1-8B	100.0 / 97.5	93.5 / 72.0	87.0 / 87.5	19.0 [11.5, 26.5]	−3.0 [−10.0, 4.0]	90.5 / 97.5
Gemma-3-12B	100.0 / 94.0	95.0 / 52.0	86.0 / 19.0	37.0 [28.5, 45.0]	61.0 [53.0, 68.5]	77.0 / 44.5

combine at every scale, with item-level bootstrap intervals that exclude zero, and 56.5 and 54.5 points for compute at 4B and 8B; at 14B and 32B the compute excess is 13.5 points (lower bound 6) against a retrieval loss of 11.5 and 7.5, present but modest. combine starts below ceiling at 4B and 8B (70 and 84%, against 98 and 99% for compute, which performs the same addition), so part of its far-distance loss is a baseline difficulty of the two-variable query, perhaps of tracking two variables or making two accesses; the 14B and 32B, starting at 95 and 92.5%, show the same ordering.

4.2 Does the dissociation hold on identical programs?

On identical programs, the use failures are largely nested inside the retrieval successes (Figure 3; Table 7, Appendix K). At $d = 1024$, Qwen3-8B retrieves the value on all 200 programs, computes with it on 100 and combines it on 2; Gemma-3-12B retrieves on 190, computes on 102 and combines on 25; Qwen3-32B retrieves on 180, computes on 146 and combines on 74. Conversely, a use query succeeded after a failed retrieval on 0 of 0 (8B), 0 of 10 (Gemma) and 6 of 20 (32B) programs at $d = 1024$; exact McNemar tests (McNemar, 1947) of retrieve against compute and against combine give $p \leq 8 \times 10^{-6}$ on every model at $d = 128$ and 1024. The nesting is exact at 8B and near-exact at 32B but not a logical implication: at $d = 128$ Gemma, whose retrieval falls to 35.5%, still computes correctly on 17 of the 129 programs it fails to retrieve, so its mid-context dip is partly a property of the copy query. The program-level excess over retrieval from $d = 0$ to 1024 is 48.5 [41.5, 55.5] (compute) and 86.5 [81.5, 91.0] (combine) points at 8B, 39.0 [31.5, 46.5] and 69.5 [62.5, 76.5] for Gemma, and 9.5 [3.0, 16.0] and 47.5 [39.5, 55.0] at 32B, within sampling error of the unpaired estimates of Table 1. The paired arm excludes the reading that Equation 1 mixes retrieval failures on some items with computation failures on others; the 32B compute effect remains the weakest result (lower bound 3 points).

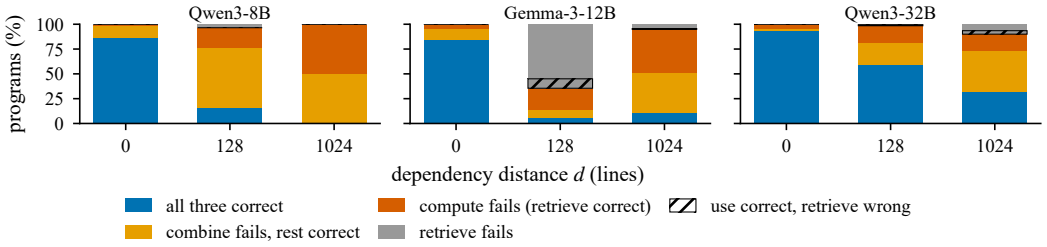

Figure 3: Paired arm ($n = 200$ programs per distance): outcome of the three queries on the same program. Bars stack the share of programs on which all three succeed, only combine fails, compute fails with retrieve correct, and retrieve fails; hatching marks programs on which a use query succeeded although retrieve failed. At $d = 1024$ the use failures sit almost entirely inside the retrieval successes; Gemma's $d = 128$ column shows its retrieval dip.

4.3 How do scale, distance and preceding load change the effect?

Scale. Within the tested range, the combine gap at 3.7K tokens is gone by 32B, whereas at 28K tokens the gaps are only attenuated (Table 1; Figure 9, Appendix G). In the short arm, retrieve is at least 99.5% everywhere, the compute excess at $d = 128$ is at most 9 points at any scale, and the combine excess falls from 58.0 (4B) and 71.5 (8B) to 19.5 (14B) and -0.5 (32B). In the long arm at $d = 1024$, the compute excess falls from about 55 points to 13.5 between 8B and 14B and is then flat, and the combine excess falls from 65.5 and 81.5 at 4B and 8B to 41.5 [32.5, 50.0] at 32B.

Distance versus preceding material. The grid arm varies p and d independently (Figure 8; post-hoc regression coefficients in Appendix G). For 8B and 14B most of the combine loss follows the material between the value and its use: 256 lines between them cost 80 and 29 points, whereas 768 lines ahead of the value, which also move it from line 9 to line 777, change combine at $d = 256$ by -12.5 points at 8B (15.0 to 2.5%; Fisher exact p-value 0.009) and $+10.0$ at 14B (p-value 0.21), a secondary effect of at most 15 points at 8B and none detectable at 14B. The 4B and 32B do not fit this account: 4B compute is 88.8% at $d = 256$ with nothing before the value and 31.2% with 768 lines before it, a strong interaction without a main effect of either factor, and 32B combine shows the same shape at a smaller amplitude (92.5 to 71.2%; Fisher exact p-value 0.0008), as does 32B compute. At both ends of the scale range distance is costly mainly when the preceding context is also long; we have no explanation for this non-monotone pattern, and with 80 items per cell these interactions warrant caution.

Preceding load at a fixed distance. The load arm asks the converse question (Figure 15 and Table 8, Appendix K). With $d = 128$ fixed and p growing from 0 to 896 lines, retrieve and compute are flat on every model (compute 85 to 96% at every load; post-hoc slopes λ of accuracy on $\log(1+p)$ all with p-values ≥ 0.25). combine shows no detectable load effect at 8B and 14B but falls at 32B from 95 to 66% ($\lambda = -0.028$, p-value 2×10^{-6}), reproducing the main arms' 3.7K-versus-28K difference at $d = 128$ (95.5 versus 66.5%) with distance fixed. At 32B, then, the mid-distance combine loss follows total context, consistent with its grid interaction, whereas its far-distance collapse occurs at constant length and cannot: the two are not one phenomenon with one scalar cause. One residual anomaly remains: the 8B's 20-point compute deficit between the arms at $d = 128$ is not a load effect (89 versus 90%) and does not reproduce across item sets of the same structure (main, paired and load-arm cells: 70, 76.5 and 90%, reproduced in one fresh engine with 96 to 100% item agreement; Appendix K), so it is heterogeneity between item sets rather than run-to-run instability, and no item feature we examined explains it.

4.4 Does the pattern hold outside Qwen3?

Only part of the pattern transfers (Table 1; Figure 14, Appendix J). Llama-3.1-8B-Instruct keeps retrieve at 93 to 100% at every distance and loses 19.0 points [11.5, 26.5] more on compute than on retrieve between $d = 0$ and 1024 (93.5 to 72%; $\hat{\gamma}_3 = -0.030$, $p = 10^{-10}$), the size of the compute effect at Qwen3-14B and 32B, but its combine curve is flat (87.0 and 87.5%; excess -3.0 [-10.0, 4.0], $p = 0.47$): the two-variable collapse, the largest effect in every Qwen3 model, does not occur in Llama at this length, and its short arm shows no gap. Its tokenizer renders the programs as 22.9K rather than about 28K tokens, so token length is not matched, and the shorter context may contribute to the missing collapse. Gemma-3-12B-it is a different case: its retrieve accuracy is 100, 100, 95 and 85.5% at $d = 0$ to 32, 39.5% at $d = 128$, 54% at $d = 512$ and 94% at $d = 1024$, so the value is retrievable from the end and the start of the context but not from the middle (Liu et al., 2024), and the retrieval premise fails for it at those two distances. At $d = 1024$, where the value is at line 9 and retrieve is 94%, the excess is 37.0 points [28.5, 45.0] for compute and 61.0 [53.0, 68.5] for combine, and its short arm shows gaps of 18 and 32 points at $d = 128$, the second uncertain because 61 short-arm combine outputs with an empty first line were scored incorrect (up to 12 points; Appendix J); because its retrieval is off the ceiling, the pre-specified logistic criterion passes on it ($\hat{\beta}_3 = -0.107$ and -0.249, $p = 0.019$ and 3×10^{-7}). In sum, the dissociation is strong and program-level in Qwen3, appears in Gemma where its retrieval holds and is substantially weaker in Llama, so it is not universal; with one raw completion format and parser for every family, these are behavioral contrasts under that protocol, not controlled comparisons of architecture (Section 7).

4.5 What recovers use: the scratchpad wording or the budget?

A step-by-step scratchpad with a 2,048-token budget returns compute to 99 and 96% and combine to 40 and 91% (8B, 32B) on the same items at $d = 1024$, against 43 and 74% and 1 and 42% under direct completion (Figure 11 and Table 5, Appendix H; exact McNemar, all $p \leq 3 \times 10^{-5}$), although at 8B half the combine outputs exhaust the budget. The question format answers nothing at 8 tokens under either instruction or at 64 tokens with the step-by-step instruction, since every output begins a derivation and is cut off, and at 256 tokens, below the median derivation length of 317 to 384 tokens, it is worse than direct completion on both models. Told to give only the number but given 2,048 tokens, the models write a derivation anyway and, scored on the last answer line, reach 96 and 86% on compute and 90 and 86% on combine (8B, 32B). Against the step-by-step instruction at that budget, the answer-only prompt is equal on 8B compute (3 versus 0 items, $p = 0.25$), better on 8B combine (52 versus 2, $p = 2 \times 10^{-13}$) because it answers before the budget runs out, worse on 32B compute (0 versus 10, $p = 0.002$) and equal on 32B combine (8 versus 13, $p = 0.38$). The recovery follows the budget, which allows a derivation under either instruction, not the question format; the wording moves the outcome by at most 10 points except where it causes truncation. Direct retrieve remains the reference, since the retrieve control is uninterpretable under the scratchpad (Appendix H). Whether the derivation reflects sequential reasoning, re-reading of the program, externalized computation or a mixture is not separable here: completed scratchpads restate chain values before answering, which may bring them into the recent context, but these observational traces do not establish restatement as the mechanism.

5 Statistical analysis

The pre-specified criterion. For each model and each $o \in \{C, B\}$, the decisive criterion fits, on the pooled retrieve and o items,

$$\operatorname{logit} P(\text{correct}_i) = \beta_0 + \beta_1 \log(1 + d_i) + \beta_2 \, \mathbf{1}[o_i = o] + \beta_3 \log(1 + d_i) \, \mathbf{1}[o_i = o], \qquad (2)$$

and passes if $\hat{\beta}_3 < 0$ with $p < 0.05$ on at least two models for at least one o; β_3 is the difference between the log-odds slopes of o and of retrieve. Criteria 1, 2 and 4 pass on all four models. Criterion 3 passes only for combine at 4B ($\hat{\beta}_3 = -0.353$, $p = 0.032$), so the rule returns no-go. At 14B and 32B the estimate is positive, and for 32B compute it is

significantly positive ($\hat{\beta}_3 = +0.313$, 95% CI $[0.064, 0.561]$, $p = 0.014$) for a condition whose raw excess over retrieval is +13.5 points (Table 2, Appendix D).

Why the two scales disagree. The verbal hypothesis, that computation degrades more than retrieval, is a statement about differences in probability; Equation 2 tests a difference in log-odds slopes, and the two diverge when the reference condition is at ceiling. retrieve is at exactly 100% in three of its seven cells on every model, so the fit reaches those cells only through a large intercept (6.4 to 8.4) and a steep log-odds slope (-0.47 to -0.91); a condition that falls from 97 to 76% then has a shallower log-odds slope (-0.31 for 32B compute) despite losing 13.5 more points. Every fit converged and Firth's bias-reduced estimates (Firth, 1993; Heinze & Schemper, 2002) are nearly unchanged (Table 4), so this is not non-existence of the maximum-likelihood estimate (Albert & Anderson, 1984). In a simulation with retrieval truly at 100% in four cells and computation truly falling from 99 to 74% (Appendix E), $\hat{\beta}_3$ is positive in every one of 2,000 datasets and significantly positive in 51% ($n = 100$ per cell) or 97% ($n = 200$); with retrieval at a flat 99.5% it detects the true negative interaction in only 21 to 47% of datasets. Its sign changes with the coding of distance (Appendix D), and on Gemma, whose retrieval is off the ceiling, the same estimator is negative and significant (Section 4.4).

Post-hoc probability-scale analysis. We refit the same design on the probability scale,

$$\mathbb{E}[\text{correct}_i] = \gamma_0 + \gamma_1 \log(1 + d_i) + \gamma_2 \, \mathbf{1}[o_i = o] + \gamma_3 \log(1 + d_i) \, \mathbf{1}[o_i = o], \tag{3}$$

by least squares with HC3 standard errors (MacKinnon & White, 1985; Long & Ervin, 2000), so that γ_3 is the difference in percentage-point slopes. $\hat{\gamma}_3$ is negative for both conditions on all four models, survives Holm correction over the eight tests (Holm, 1979), and a permutation test that shuffles condition labels within each distance level gives $p \leq 0.001$ throughout (Table 2). Three qualifications apply: the analysis was chosen after the pre-specified one failed and is post hoc; only this family of eight tests is corrected for multiplicity, and every other p-value in the paper (grid, load, paired and ablation contrasts) is uncorrected and descriptive; and with the retrieve cells at exactly 100% removed, the compute interaction is no longer significant at 14B ($p = 0.89$), though it remains so at 32B ($p = 0.004$), whereas combine remains significant on every model ($p \leq 2 \times 10^{-19}$ at 14B, 5×10^{-11} at 32B). The evidence therefore ranks as follows. Strongest: the combine dissociation, robust to estimator, coding and exclusions, confirmed program by program, present at 32B in long context. Moderate: the compute dissociation at 4B and 8B, and Gemma's paired compute effect. Weakest: compute at 14B and 32B (13.5 points; paired lower bound 3 at 32B; significance at 14B depends on the near-ceiling cells), cross-family generality, and any account in which distance alone carries the effect. We make no confirmatory claim from the pre-specified rule: the inferential statements here are post hoc; the descriptive statements of Table 1 are what its criteria 1 and 2, both on the probability scale, tested and passed on every model.

6 Discussion

High retrieval accuracy does not imply that a model can use the retrieved value in the next operation: copying a value across a long dependency degrades far less than performing one operation on it in the same forward pass, and on identical programs the use failures are largely nested inside the retrieval successes. Successful retrieval of a value does not ensure its successful use. The failure is neither universal nor governed by one scalar notion of distance. For 8B and 14B most of the cost lies between the value and its use, which argues against absolute position as the sole explanation at those scales; at 4B and 32B the cost appears only when preceding load is also large, and at 32B the load arm shows that the mid-distance combine loss follows total context while the far-distance collapse does not. A 2,048-token budget recovers compute on the same items whether or not the prompt asks for a derivation, so the deficit is not a fixed property of the items, although what the model does with the budget is not identified.

7 Limitations

Models and task. The evidence comes from one deliberately controlled synthetic task: modular addition of one-digit operands, chosen so the arithmetic stays at ceiling (Zhou et al., 2024), with opaque names in a fully evaluated program. That control is why it was chosen, but it establishes a behavioral phenomenon under this construction, not its prevalence in natural-language tasks; extending the design to natural-language and multi-hop benchmarks is future work. Qwen3 at 4 to 32 billion parameters is the cleanest family; the grid, scratchpad, load and sample-size extensions were run on Qwen3 alone. A DeepSeek-R1 distillation and Mistral-7B fail the retrieval check at 28K tokens and give no evidence either way, Llama-3.1-8B reproduces only the one-operation gap, and Gemma-3-12B reproduces both gaps only where its position- and query-form-sensitive retrieval holds. Cross-family contrasts need caution: every model, all post-trained, received raw completions rather than its native chat template, which may be out of distribution and cause formatting artifacts or family-specific behavior; the strict parser scores such artifacts as errors (61 Gemma short-arm and three 32B outputs); tokenizers give the same programs different lengths; and native-template evaluation, an important control, was not run. Nothing above 32B or proprietary was tested, so the results do not establish behavior in frontier proprietary or much larger open models, and scale trends should not be extrapolated beyond 32B. Every result is under greedy decoding, with Qwen3's thinking mode never invoked (the budget ablation is the closest analogue); generalization to other architectures, training regimes or decoding strategies is not established, and no mechanistic probes or alternative systems were run.

Design and inference. The pre-specified plan was fixed in code, not registered, and its decisive criterion, which returned no-go, did not fix the scale of comparison; the probability-scale analysis, the grid and load regressions and strict scratchpad scoring are post hoc. The compute excess at 14B and 32B is modest (13.5 points) and its significance at 14B depends on the near-ceiling retrieval cells. In the main arms dependency distance and absolute position are collinear; the grid breaks the collinearity only for $p \leq 768$ and $d \leq 256$, with 80 items per cell. The interactions at 4B and 32B confine the dependency-distance account to 8B and 14B. The 8B $d = 128$ compute cell varies between same-structure item sets beyond binomial error (67 to 90%), so the $d = 128$ estimates carry item-set uncertainty their intervals do not show; the $d = 1024$ cells replicate across sets. The load arm draws independent items at each load (100 per cell) rather than reusing programs; the paired arm's nesting is exact at 8B but not at 32B (6 of 20 failed retrievals at $d = 1024$ followed by a correct compute).

Scratchpad and infrastructure. The scratchpad arms use 50 and 100 items per cell, leave up to half the 8B combine outputs truncated at 2,048 tokens and render the retrieve control uninterpretable; the ablation covers $d = 1024$ and two models only and does not identify what the derivation does. The 14B and 32B ran on a different stack from the 4B and 8B; the 900-item same-model replication bounds the stack effect for the 8B only (Appendix I).

8 Conclusion

Retrieving a value from a long context does not guarantee that a model can compute with it in the same forward pass. In the Qwen3 models we tested, copying a value survived 1,024 lines of separation at 28K tokens, adding a one-digit literal to it did not, adding a local variable failed almost completely below 14B, and every use failure at 8B followed a successful copy of the same program. What the loss depends on is regime-specific: scale halved it at 28K tokens and removed it at 3.7K, and whether it follows distance, preceding load or total context depends on scale. The load and budget arms showed that the mid-distance loss at 32B is a load effect and that a long generation budget recovers compute even without an explicit scratchpad instruction. Llama-3.1-8B shows the one-operation gap only and Gemma-3-12B both: the dissociation is a boundary to be mapped across more families, larger models and natural text, not a universal, and what a model does with the tokens that let it recover remains open.

Reproducibility statement

Item generation is deterministic given a seed (Appendix A); the supplement holds all code and all 74,567 scored outputs; Appendix L lists models, decoding, software and hardware.

Ethics statement

Synthetic arithmetic programs and public model weights only; no human subjects, personal data or sensitive content; no direct path to harmful applications that we can see.

AI use statement

AI assistants were used to aid and polish the writing of this paper, and provided some coding assistance during implementation and analysis. The authors directed the work, reviewed the code, verified every reported number against the stored per-item outputs by independent recomputation, and take full responsibility for the content of the paper.

References

Adelin Albert and John A. Anderson. On the existence of maximum likelihood estimates in logistic regression models. Biometrika, 71(1):1–10, 1984.

Yushi Bai et al. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025.

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João G. M. Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. Transformers need glasses! information over-squashing in language tasks. In Advances in Neural Information Processing Systems 37, 2024.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A. Huerta, and Hao Peng. Context length alone hurts LLM performance despite perfect retrieval. In Findings of the Association for Computational Linguistics: EMNLP 2025, 2025.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In Advances in Neural Information Processing Systems 36, 2023.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In International Conference on Learning Representations (ICLR), 2024.

Hamed Firooz, Maziar Sanjabi, Wenlong Jiang, and Xiaoling Zhai. Lost-in-distance: Impact of contextual proximity on LLM performance in graph tasks. arXiv preprint arXiv:2410.01985, 2024.

David Firth. Bias reduction of maximum likelihood estimates. Biometrika, 80(1):27–38, 1993.

Gemma Team. Gemma 3 technical report. arXiv preprint arXiv:2503.19786, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Daya Guo et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. Nature, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. Statistics in Medicine, 21(16):2409–2419, 2002.

Sture Holm. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6(2):65–70, 1979.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In First Conference on Language Modeling (COLM), 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. arXiv preprint arXiv:2310.06825, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Advances in Neural Information Processing Systems 35, 2022.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In Advances in Neural Information Processing Systems 37, Datasets and Benchmarks Track, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In Proceedings of the 29th Symposium on Operating Systems Principles (SOSP), pp. 611–626, 2023.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 15339–15353, 2024.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12:157–173, 2024.

J. Scott Long and Laurie H. Ervin. Using heteroscedasticity consistent standard errors in the linear regression model. The American Statistician, 54(3):217–224, 2000.

James G. MacKinnon and Halbert White. Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. Journal of Econometrics, 29 (3):305–325, 1985.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2):153–157, 1947.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. NoLiMa: Long-context evaluation beyond literal matching. In Proceedings of the 42nd International Conference on Machine Learning (ICML), volume 267 of Proceedings of Machine Learning Research, 2025.

Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. Proceedings of the National Academy of Sciences, 115(11): 2600–2606, 2018.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. arXiv preprint arXiv:2112.00114, 2021.

Alex Pan and Mary-Anne Williams. Context is not comprehension. arXiv preprint arXiv:2506.04907, 2025.

Jonathan Roberts, Kai Han, and Samuel Albanie. Needle threading: Can LLMs follow threads through near-million-scale haystacks? In International Conference on Learning Representations (ICLR), 2025.

Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with Python. In Proceedings of the 9th Python in Science Conference, 2010.

Akshit Sinha, Arvindh Arun, Shashwat Goel, Steffen Staab, and Jonas Geiping. The illusion of diminishing returns: Measuring long horizon execution in LLMs. In International Conference on Learning Representations (ICLR), 2026.

Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, Rohan Anil, Ethan Dyer, Siamak Shakeri, Roopali Vij, Harsh Mehta, Vinay Ramasesh, Quoc Le, Ed Chi, Yifeng Lu, Orhan Firat, Angeliki Lazaridou, Jean-Baptiste Lespiau, Nithya Attaluri, and Kate Olszewska. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. arXiv preprint arXiv:2409.12640, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems 35, pp. 24824–24837, 2022.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. Journal of the American Statistical Association, 22(158):209–212, 1927.

Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45, 2020.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In International Conference on Learning Representations (ICLR), 2025.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025a.

Yijun Yang, Zeyu Huang, Wenhao Zhu, Zihan Qiu, Fei Yuan, Jeff Z. Pan, and Ivan Titov. A controllable examination for long-context language models. In Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track, 2025b.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. In International Conference on Learning Representations (ICLR), 2025.

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. Transformers can achieve length generalization but not robustly. In ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models, 2024. arXiv:2402.09371.

A Item generation and prompts

Constants: modulus 9973; chain update $x_{t+1} = (7x_t + 13) \bmod 9973$; probe $v = (3x_t + 991) \bmod 9973$; lead-in 8 lines; $x_0 \sim \text{Uniform}\{1000, \ldots, 9972\}$; literal k and local variable w uniform on $\{2, \ldots, 9\}$; names drawn without replacement from two letters (excluding l and o) and one digit. Header: "Below is a partially evaluated program. Each line shows an assignment and its computed value. Complete the final line by giving only its numeric value." A short-arm compute item at $d = 2$ ends

```
rn2 = (rv1 * 7 + 13) % 9973 = 8603
ts1 = (rn2 * 3 + 991) % 9973 = 6854
cn7 = (rn2 * 7 + 13) % 9973 = 396
nm5 = (cn7 * 7 + 13) % 9973 = 2785
ms5 = (ts1 + 7) % 9973 =
```

with answer 6861. Long-arm prompts are 36,189 to 36,345 characters (27,931 to 28,101 Qwen3 tokens); short-arm prompts 4,942 to 4,993 characters (3,730 to 3,796 tokens); grid prompts 485 to 36,268 characters (281 to 28,078 tokens). In the long arm (200 items per cell) it spans 8 to 28 tokens at $d = 0$, 59 to 83 at $d = 2$, 220 to 250 at $d = 8$, 863 to 902 at $d = 32$, 3,452 to 3,510 at $d = 128$, 13,828 to 13,916 at $d = 512$ and 27,659 to 27,783 at $d = 1024$ (medians 19, 73, 236, 885, 3,481, 13,874 and 27,728); in the short arm, 8 to 28, 57 to 82, 218 to 247, 861 to 903 and 3,456 to 3,515 (medians 19, 73, 235, 885 and 3,481). Result files were checked against a regeneration from seed 0: answers, cells, probe positions and program lengths match on every row (81,440 rows over 35 of the 43 result files, in 37 comparisons because the diagnostic file is checked against each of its three source item sets). The two combined $n = 200$ files repeat the rows of their component runs, so the 74,567 unique scored outputs occupy 82,967 rows; the 8 files outside the check (1,527 rows) are pilot and smoke runs and the cross-stack replication.

B Pilot and design iteration

A first design used four-digit literals and locals. On a 210-item smoke test with Qwen3-4B, combine scored 20% and compute 80% at $d = 0$, where retrieval is trivial, so the ladder confounded arithmetic difficulty with retrieval demand. Operands were reduced to one digit, and a 108-item diagnostic confirmed 92 to 100% at $d = 0$ before any main run. The 14B and 32B long arms were first run at 100 items per cell (the first 100 item ids of each cell), on cost grounds, and later extended with the remaining 100 ids of each cell on the same stack and settings; the two halves differ by 3.3 (14B) and 2.6 (32B) points per cell on average and by at most 10 and 7 points, and every statement in the paper uses the combined $n = 200$.

C Pre-specified analysis plan

The four criteria and their implementation are reproduced verbatim from analyze_pilot.py in the supplementary material. Criterion 1: mean accuracy at $d = 0$ minus mean accuracy at $d = 1024$, pooled over conditions, at least 8 points on at least two models (observed 44.7, 46.3, 36.8 and 25.8 for 4B, 8B, 14B and 32B at $n = 200$). Criterion 2: Spearman correlation between distance level and pooled accuracy at most -0.7, and at most -0.5 excluding the farthest level (observed -0.94, -0.96, -1.00 and -0.96; excluding the farthest level -0.94, -0.94, -1.00 and -0.94). Criterion 3: Equation 2, as stated in Section 5. Criterion 4: program length constant across distances (1,034 or 1,035 lines) and live set constant by construction. Verdict on the four valid models: criteria 1, 2 and 4 pass; criterion 3 passes on one model (4B, combine); no-go.

D Sensitivity of the interaction estimates

Table 2 gives the pre-specified and probability-scale interactions for every model and condition; Table 3 reports the probability-scale interaction under three codings of distance and

four exclusions, and Table 4 the logistic interaction with and without Firth's bias reduction. The logistic estimate changes sign or significance across codings (for example 8B compute: -0.09, $p = 0.53$ with log coding; -1.56, $p = 0.004$ with linear coding); the probability-scale estimate keeps its sign and significance in every coding. Under exclusions, the probability-scale combine interaction stays significant on every model; the compute interaction at 14B is not significant when the retrieve cells at exactly 100% are removed ($p = 0.89$) or when $d = 1024$ is removed ($p = 0.085$), whereas at 32B it is significant under every exclusion. HC3 and classical standard errors differ by at most 15%.

Table 2: Condition × log-distance interaction under the pre-specified logistic model (Equation 2, $\hat{\beta}_3$) and under the post-hoc probability-scale model (Equation 3, $\hat{\gamma}_3$, HC3 standard errors). The column $p_{\text{excl.}}$ gives the probability-scale p after removing the retrieve cells at exactly 100%. A permutation test that shuffles condition labels within each distance level (2,000 draws) gives $p \leq 0.001$ for $\hat{\gamma}_3$ in every row; Firth bias-reduced estimates of β_3 are in Table 4.

model	cond.	$\hat{\beta}_3$ [95% CI]	p	$\hat{\gamma}_3$ [95% CI]	p	$p_{\text{excl.}}$
4B	compute	+0.191 [−0.113, 0.495]	0.219	−0.077 [−0.086, −0.067]	3×10^{-59}	2×10^{-49}
4B	combine	−0.353 [−0.676, −0.030]	0.032	−0.082 [−0.090, −0.073]	2×10^{-82}	2×10^{-66}
8B	compute	−0.086 [−0.351, 0.179]	0.526	−0.064 [−0.072, −0.055]	4×10^{-51}	4×10^{-37}
8B	combine	−0.120 [−0.381, 0.141]	0.367	−0.101 [−0.110, −0.093]	2×10^{-131}	9×10^{-93}
14B	compute	+0.344 [−0.014, 0.701]	0.060	−0.013 [−0.021, −0.006]	3×10^{-4}	0.89
14B	combine	+0.302 [−0.038, 0.641]	0.081	−0.078 [−0.087, −0.069]	4×10^{-60}	2×10^{-19}
32B	compute	+0.313 [0.064, 0.561]	0.014	−0.024 [−0.032, −0.015]	2×10^{-8}	0.004
32B	combine	+0.238 [−0.008, 0.484]	0.058	−0.050 [−0.059, −0.040]	7×10^{-25}	5×10^{-11}

Table 3: Probability-scale interaction p-values (Equation 3, HC3) under alternative codings of distance and under exclusions of cells. Values recomputed from the raw outputs.

		coding of distance			exclusion		
model	condition	log	linear	rank	drop d=0	drop d=1024	$d \geq 8$ only
4B	compute	3×10^{-59}	2×10^{-15}	1×10^{-60}	1×10^{-18}	7×10^{-53}	0.005
4B	combine	2×10^{-82}	3×10^{-40}	5×10^{-83}	8×10^{-23}	3×10^{-79}	4×10^{-5}
8B	compute	4×10^{-51}	4×10^{-39}	5×10^{-54}	3×10^{-40}	5×10^{-16}	4×10^{-29}
8B	combine	2×10^{-131}	9×10^{-103}	5×10^{-143}	2×10^{-49}	7×10^{-68}	3×10^{-21}
14B	compute	3×10^{-4}	0.003	3×10^{-4}	0.001	0.085	0.007
14B	combine	4×10^{-60}	6×10^{-63}	4×10^{-62}	1×10^{-50}	4×10^{-21}	1×10^{-47}
32B	compute	2×10^{-8}	0.003	3×10^{-8}	2×10^{-5}	2×10^{-8}	4×10^{-4}
32B	combine	7×10^{-25}	3×10^{-19}	9×10^{-26}	1×10^{-20}	2×10^{-9}	5×10^{-14}

Table 4: Logistic interaction $\hat{\beta}_3$ (Equation 2) by maximum likelihood and by Firth's bias-reduced likelihood, with p-values. Every maximum-likelihood fit converged; the bias reduction moves the estimates little.

model	condition	ML $\hat{\beta}_3$	p	Firth $\hat{\beta}_3$	p
4B	compute	+0.191	0.219	+0.161	0.270
4B	combine	−0.353	0.032	−0.379	0.015
8B	compute	−0.086	0.526	−0.105	0.419
8B	combine	−0.120	0.367	−0.140	0.273
14B	compute	+0.344	0.060	+0.315	0.070
14B	combine	+0.302	0.081	+0.269	0.102
32B	compute	+0.313	0.014	+0.294	0.016
32B	combine	+0.238	0.058	+0.219	0.070

E Simulation of the log-odds interaction under ceiling retrieval

We generated 2,000 datasets per scenario with seven distance levels and $n \in \{100, 200\}$ items per cell. The compute curve was fixed at $(0.99, 0.92, 0.90, 0.85, 0.77, 0.73, 0.74)$; the retrieve curve was either $(1, 1, 1, 1, 0.97, 0.97, 0.92)$, a flat 0.995, or a flat 0.98. Figure 4 reports how often the fitted $\hat{\beta}_3$ of Equation 2 was significantly positive (wrong-signed) or significantly negative. With four cells at exactly 100%, $\hat{\beta}_3 > 0$ in every dataset and significantly so in 51.2% ($n = 100$) and 97.5% ($n = 200$), while the probability-scale $\hat{\gamma}_3$ was negative and significant in 100%. Figure 5 shows the fitted and raw cell probabilities on the real data.

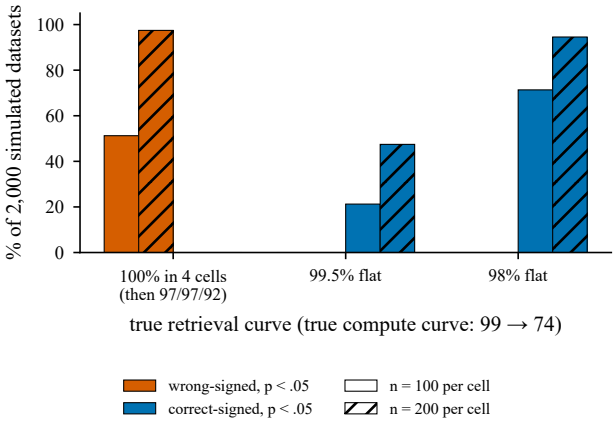

Figure 4: Simulation: share of 2,000 datasets per scenario in which the logistic interaction $\hat{\beta}_3$ is significantly wrong-signed (positive) or correctly signed (negative), for three true retrieval curves and two cell sizes.

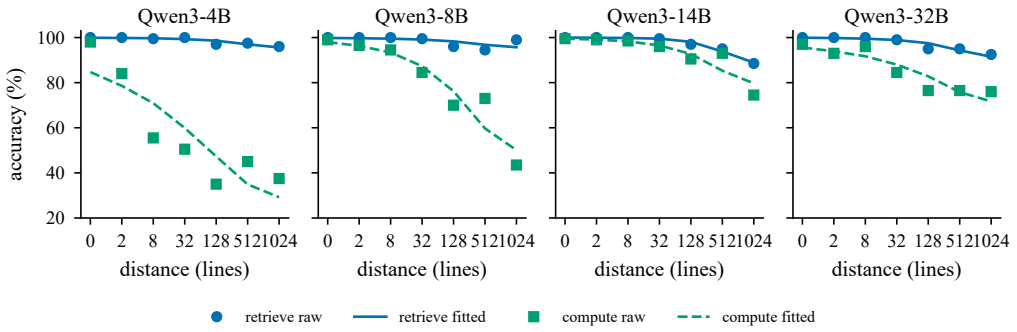

Figure 5: Raw (markers) and fitted (lines) cell probabilities under Equation 2 for retrieve versus compute on the real data. The fitted retrieval curve is steeper on the log-odds scale than the fitted compute curve at 14B and 32B, although it loses fewer points.

F Full curves, short arm, and the excluded model

Figures 6 and 7 show all five models in both arms. DeepSeek-R1-Distill-Qwen-7B is distilled from Qwen2.5-Math-7B, whose training window is 4,096 tokens (its configuration lists a 4,096-token sliding window and a RoPE base of 10,000, with the maximum-position field relabelled to 131,072). At 28K tokens its retrieve accuracy is 100, 97.5, 92.5, 49, 1.5, 0 and 0% across the seven distances; at 3.7K tokens it is 100, 100, 99.5, 93.5 and 77.5%. It fails the retrieval manipulation check at 28K tokens and is excluded from criterion 3 on that basis alone.

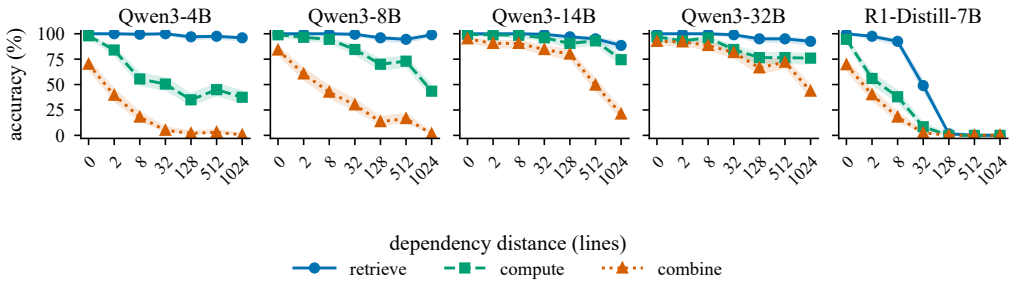

Figure 6: Long arm (28K tokens), all five models; bands are Wilson 95% intervals.

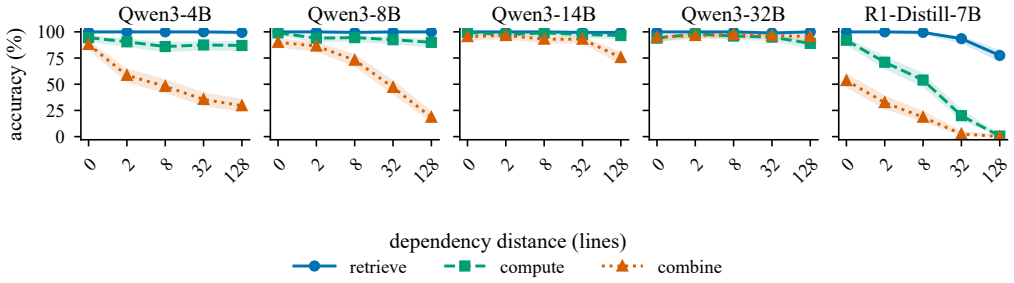

Figure 7: Short arm (3.7K tokens), all five models; bands are Wilson 95% intervals.

G Matched-distance comparison, scale and grid figures

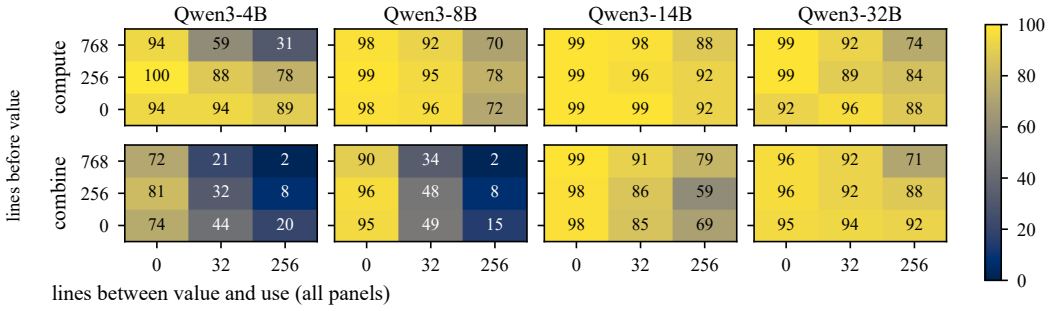

Figure 8: Grid arm ($n = 80$ per cell): accuracy as a function of the number of lines before the value (rows) and between the value and its use (columns). At 8B and 14B accuracy changes across columns and little down rows; at 4B and 32B the loss is concentrated in the cell where both are large.

Grid regressions (post hoc). Per model and condition we fit $\mathbb{E}[\text{correct}] = \delta_0 + \delta_p \log(1+p) + \delta_d \log(1+d) + \delta_{pd} \log(1+p)\log(1+d)$ with HC3 standard errors. For combine at 8B and 14B, $\delta_d = -0.143$ and -0.053 (p-values 3×10^{-69} and 6×10^{-9}), while δ_p and δ_{pd} are not significant (p-values ≥ 0.12). For 4B compute neither main effect is significant but the interaction is ($\delta_{pd} = -0.0124$, p-value 10^{-12}); for 32B combine, $\delta_d = -0.0025$ (p-value 0.71) and $\delta_{pd} = -0.0044$ (p-value 0.0035), and 32B compute shows the same interaction (p-value 0.0037).

Figure 10 shows the long-arm minus short-arm accuracy at each shared distance. For compute the difference at $d = 8, 32, 128$ is $-30.5, -37.0, -52.0$ points at 4B; $0, -8, -20$ at 8B;

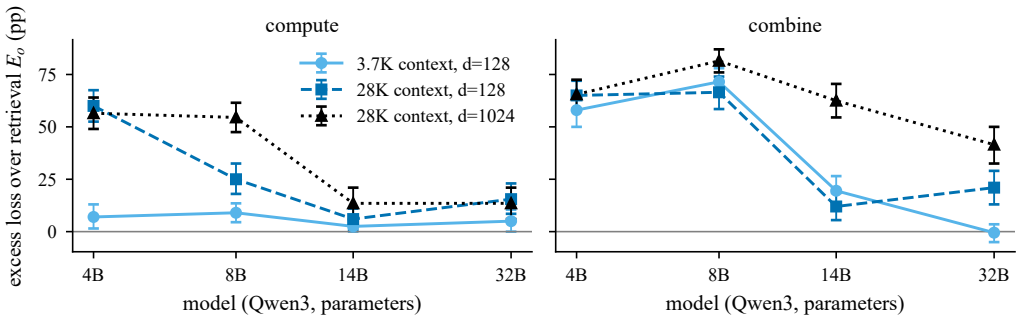

Figure 9: Excess loss over retrieval (Equation 1) against model scale for the 3.7K-token arm at $d = 128$, the 28K-token arm at the same d (identical token distance), and the 28K-token arm at $d = 1024$. Error bars are item-level bootstrap 95% intervals. At 3.7K tokens the combine excess reaches zero by 32B; at 28K tokens the $d = 1024$ series stay well above zero at every scale.

$-0.5, -1.5, -6$ at 14B; $0, -10.5, -12.5$ at 32B. For combine at 32B it is $-9, -15, -29$. This contrast confounds total length, preceding material and position and is reported descriptively.

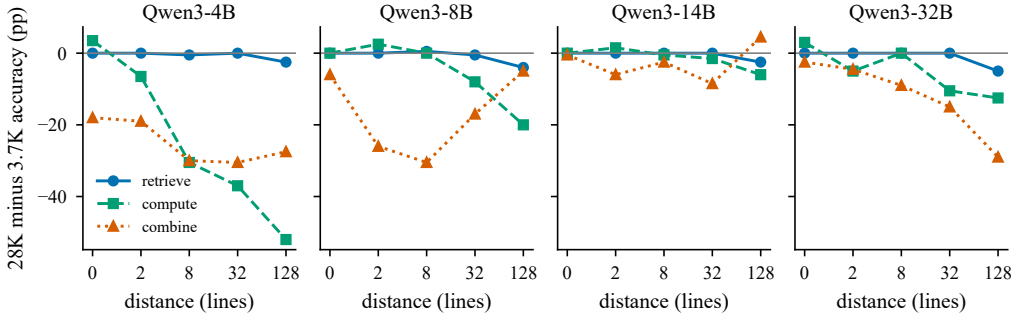

Figure 10: 28K-token minus 3.7K-token accuracy at identical token distance, by query type.

H Scratchpad runs: the 384-token run and the format × budget ablation

Table 5: Format × budget ablation on 100 compute and 100 combine items at $d = 1024$ (the same items for every cell). Each cell gives accuracy, the share of outputs with no answer within the budget, and accuracy among answered outputs (%). The completion row is the direct regime (first integer on the first line); the question rows require a terminal ANSWER.

		Qwen3-8B						Qwen3-32B					
		compute			combine			compute			combine		
format	budget	acc	trunc	among	acc	trunc	among	acc	trunc	among	acc	trunc	among
completion ("=")	8	43	0	43	1	0	1	74	0	74	42	0	42
question, answer only	8	0	100	–	0	100	–	0	100	–	0	100	–
question, answer only	2,048	96	0	96	90	0	90	86	5	91	86	6	91
question, step by step	64	0	100	–	0	100	–	0	100	–	0	100	–
question, step by step	256	3	96	75	1	99	100	11	88	92	22	77	96
question, step by step	2,048	99	0	99	40	50	80	96	1	97	91	6	97

The ablation (Table 5) was run after the 2,048-token scratchpad arm (Figure 12) on a superset of its items (100 rather than 50 per cell); on the 100 overlapping items the two

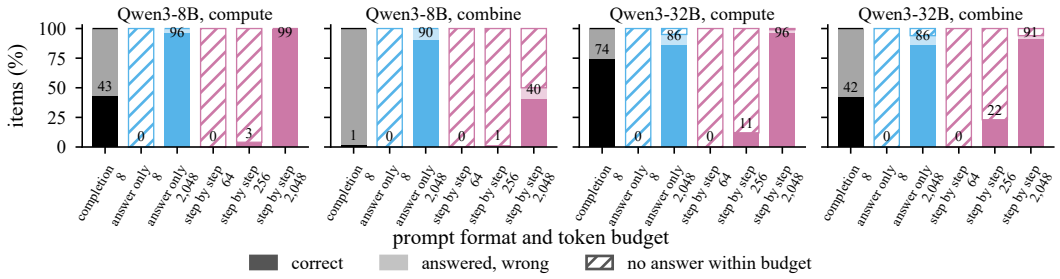

Figure 11: Format × budget ablation on 100 identical compute and 100 identical combine items at $d = 1024$. Bars give accuracy (dark), answered-but-wrong (light) and no answer within the budget (hatched) for the completion format at 8 tokens, the question with an answer-only instruction at 8 and 2,048 tokens, and the question with a step-by-step instruction at 64, 256 and 2,048 tokens. Recovery appears only at the long budget and under either instruction.

runs agree on 96% (8B) and 100% (32B) of outcomes. Under the question format both models begin a derivation in every output whatever the instruction, so the 8- and 64-token cells contain no answers; the median derivation at the 2,048-token budget is 384 (8B) and 317 (32B) tokens.

In the ablation, against direct completion on the same items, the 2,048-token step-by-step scratchpad changed 56 versus 0 (compute) and 40 versus 1 (combine) items at 8B and 25 versus 3 and 53 versus 4 at 32B; 0 (compute) and 50% (combine) of outputs were truncated at 8B and 1 and 6% at 32B, and 80% of the answered 8B combine outputs were correct. Under the question format, compute accuracy falls from 43 and 74% to 0 at 8 tokens, and at 256 tokens 1 to 23% of items are answered. Under the answer-only instruction at 2,048 tokens, 57 and 73% of outputs (8B, 32B) keep generating after the answer until the budget is spent, and 0 to 6% contain no answer. At $d = 0$ the scratchpad leaves compute at 98% on both models and costs the 8B 14 points on combine (80 to 66%; first run); asked for a value by name, the models often re-derive the chain and exhaust the budget, which makes the retrieve control uninterpretable under the scratchpad.

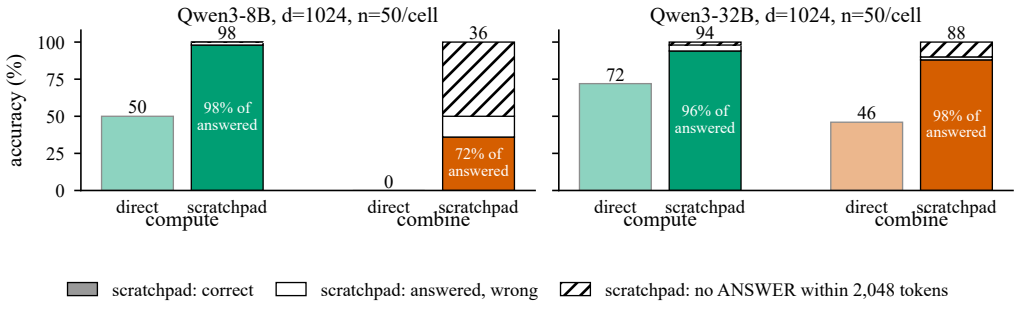

Figure 12: Direct decoding versus a 2,048-token scratchpad on identical items at $d = 1024$ ($n = 50$ per cell). Scratchpad bars are split into correct, answered but wrong, and no ANSWER within the budget; the label inside each bar is accuracy among answered outputs, a selected subset.

A first scratchpad run used a 384-token budget on 100 items per cell at $d \in \{0, 128, 1024\}$. The runner's parser fell back to the last integer in the output when no ANSWER was present, which scored truncated derivations as answers (128 of 900 items at 32B and 82 of 900 at 8B were scored correct without an ANSWER). Re-scored strictly, truncation was 70 to 86% for retrieve, 18 to 23% for compute and 20 to 40% for combine at 32B, and 87 to

99%, 24 to 36% and 64 to 83% at 8B (Figure 13). Among answered outputs, accuracy was 93 to 100% in every cell. The 2,048-token run in the main text supersedes these results. The retrieve truncation arises because the models re-derive the filler chain from the beginning when asked for a value by name; with the 2,048-token budget the retrieve truncation rate is 20% (32B) and 66% (8B) at $d = 0$ and 12% and 2% at $d = 1024$.

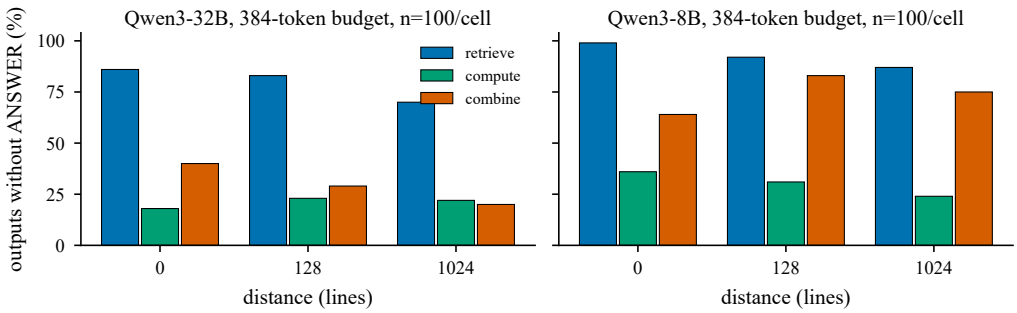

Figure 13: Fraction of outputs with no ANSWER within 384 tokens, by query type and distance.

I Cross-stack replication

The 4B and 8B long and short arms were scored on an RTX 4090 with transformers (batch size 1, SDPA attention), and the 14B, 32B, grid and scratchpad arms on an A100-80GB with vLLM 0.29. Both use bfloat16 weights and greedy decoding. To check that the two stacks agree, the 8B long-arm items at $d \in \{0, 128, 1024\}$ (the first 100 per cell, 900 items) were re-scored on the A100 under vLLM with the same decoding settings. The two stacks agreed on the correctness of 886 of the 900 items (98.4%), produced the same integer on 97.8% and byte-identical output strings on 97.1%. The 14 disagreements split 6 correct only on the 4090 and 8 correct only under vLLM (exact McNemar $p = 0.79$). Per-cell accuracy differed by at most 2 points (Table 6); the far-distance excess over retrieval on these items was 56.0 versus 54.0 points for compute and 82.0 versus 79.0 for combine. Throughput was 0.32 items per second at 28K tokens under vLLM versus 0.20 on the 4090, consistent with a prefill-bound workload. The replication shares the parser, prompts and decoding settings with every run, so it excludes a pipeline difference as the source of the scale contrast in Section 4.3; it does not exclude a numerical difference specific to the 14B or 32B weights, which would require rerunning those models under transformers.

Table 6: Cross-stack replication, Qwen3-8B, identical items ($n = 100$ per cell). Accuracy (%) on the RTX 4090 with transformers versus the A100 with vLLM, and the fraction of items with the same predicted integer.

condition	d	4090 / transformers	A100 / vLLM	difference	same integer
retrieve	0	100.0	100.0	0.0	100.0
retrieve	128	94.0	96.0	+2.0	98.0
retrieve	1024	100.0	99.0	−1.0	99.0
compute	0	99.0	99.0	0.0	100.0
compute	128	69.0	69.0	0.0	96.0
compute	1024	43.0	44.0	+1.0	96.0
combine	0	83.0	81.0	−2.0	98.0
combine	128	11.0	13.0	+2.0	93.0
combine	1024	1.0	1.0	0.0	100.0

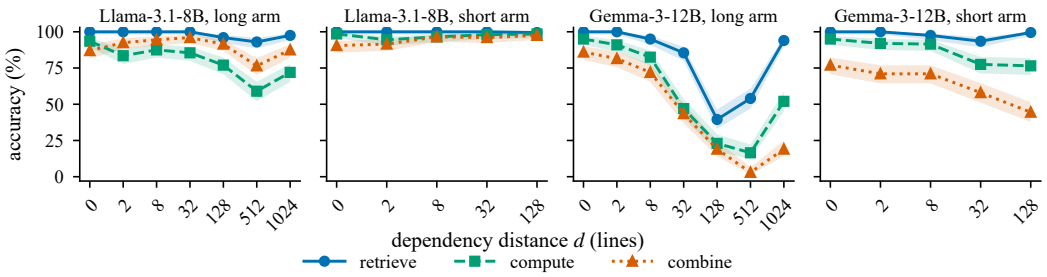

Figure 14: The two non-Qwen3 families that passed the validity gate, on the same items and settings as Figure 2 ($n = 200$ per cell; Wilson 95% bands). Llama-3.1-8B keeps retrieval at ceiling and shows a compute gap but no combine gap. Gemma-3-12B's retrieval falls to 39.5% at $d = 128$ and recovers to 94% at $d = 1024$, where the value sits at the start of the context; both gaps are large where retrieval holds.

J Other families and the validity gate

Every candidate family ran the 80-item gate of Section 3 before any full run: 30 retrieve items at $d = 1024$ and 10 items per condition at $d = 1024$ and at $d = 0$, drawn from the main item file. Llama-3.1-8B-Instruct passed (retrieve 29/30 at $d = 1024$; 10/10 and 10/10 at $d = 0$; prompts 22,847 to 22,924 tokens against a 131,072 window). Gemma-3-12B-it passed (28/30; 10/10 and 10/10; 27,881 to 27,982 tokens). Mistral-7B-Instruct-v0.3 failed: prompts of 28,338 to 28,447 tokens fit its 32,768 window, but retrieve at $d = 1024$ was 12/30 and combine at $d = 0$ was 2/10; its wrong far-distance outputs are other numbers from the program, as R1-Distill's are, so it loses the value at this length and was excluded on the same rule. Full long-arm cells for the two admitted families (accuracy %, $d = 0, 2, 8, 32, 128, 512, 1024$): Llama retrieve 100, 100, 100, 100, 96, 93, 97.5; compute 93.5, 83.5, 87.5, 85.5, 77, 59, 72; combine 87, 92.5, 94.5, 96, 91.5, 76.5, 87.5. Gemma retrieve 100, 100, 95, 85.5, 39.5, 54, 94; compute 95, 91, 82.5, 47, 23, 16.5, 52; combine 86, 81.5, 72, 43.5, 19, 3, 19. In the main arm the probe sits at line $1033 - d$, so Gemma's retrieve trough at $d = 128$ and 512 (39.5 and 54%; lines 905 and 521) and its recovery at $d = 1024$ (94%; line 9) are a profile over absolute position; the design cannot separate the two for this model. Short-arm cells ($d = 0, 2, 8, 32, 128$): Llama retrieve 100, 100, 100, 100, 99.5; compute 98.5, 94.5, 96.5, 98, 98.5; combine 90.5, 91.5, 96.5, 96, 97.5. Gemma retrieve 100, 100, 97.5, 93.5, 99.5; compute 95, 92, 91.5, 77.5, 76.5; combine 77, 71, 71, 58, 44.5. Sixty-one Gemma short-arm outputs (23, 24, 8, 3 and 3 combine items at $d = 0, 2, 8, 32, 128$) had an empty first generated line and are scored incorrect under the pre-specified parser; the runner stores only the first line, so we cannot say what followed, and Gemma's combine accuracy at short distances may be underestimated by up to 12 points. These are formatting failures of unknown content, not observed arithmetic errors; every Llama output and every Gemma long-arm and paired-arm output had an integer on its first line. Median probe-to-query token distances at $d = 1024$: Llama 22,615; Gemma 27,583; Qwen3 27,728.

K Paired programs and context load

Paired arm. Each of 600 programs ($p + d = 1024$; 200 at each $d \in \{0, 128, 1024\}$) was queried three ways from the same prefix; Table 7 cross-tabulates the outcomes. Failures nest: on Qwen3-8B no program at any distance is computed or combined correctly after a failed retrieval; on Gemma-3-12B this holds at $d = 0$ and $d = 1024$ but not at $d = 128$, where 17 of 129 programs with a failed retrieve are still computed correctly (and 8 combined); on Qwen3-32B 1 of 3 and 6 of 20 failed retrievals at $d = 128$ and 1024 are followed by a correct compute. Paired cell accuracies agree with the unpaired main-arm cells within sampling error (largest difference: Gemma combine at $d = 1024$, 13.0 versus 19.0%).

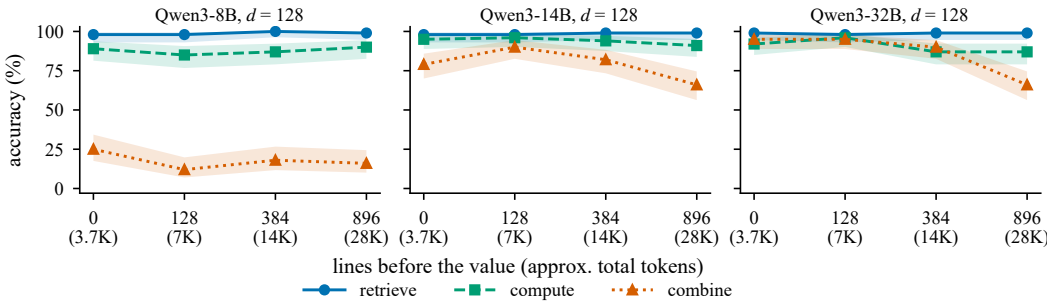

Figure 15: Load arm ($n = 100$ per cell; Wilson 95% bands): accuracy at a fixed dependency distance of 128 lines as the material before the value grows from 0 to 896 lines (3.7K to 28K tokens). retrieve and compute show no detectable load effect on any model; combine falls with load at 32B only. Items are drawn independently at each load.

Table 7: Paired arm: cross-tabulation of the three queries on the same programs ($n = 200$ per distance). R, C, B denote retrieve, compute, combine. Columns: programs retrieved; of those, programs computed and combined correctly (% of retrieved); programs not retrieved; of those, programs on which compute / combine nevertheless succeeded; exact McNemar p-values for retrieve versus compute and versus combine.

model	d	R ok	C ok | R ok	B ok | R ok	R wrong	C / B ok | R wrong	$p_{R,C}$	$p_{R,B}$
Qwen3-8B	0	200	197 (98.5)	175 (87.5)	0	0 / 0	2×10^{-1}	6×10^{-8}
Qwen3-8B	128	193	153 (79.3)	34 (17.6)	7	0 / 0	2×10^{-12}	3×10^{-48}
Qwen3-8B	1024	200	100 (50.0)	2 (1.0)	0	0 / 0	2×10^{-30}	5×10^{-60}
Gemma-3-12B	0	200	190 (95.0)	175 (87.5)	0	0 / 0	2×10^{-3}	6×10^{-8}
Gemma-3-12B	128	71	28 (39.4)	21 (29.6)	129	17 / 8	1×10^{-3}	2×10^{-8}
Gemma-3-12B	1024	190	102 (53.7)	25 (13.2)	10	0 / 1	6×10^{-27}	4×10^{-48}
Qwen3-32B	0	200	191 (95.5)	193 (96.5)	0	0 / 0	4×10^{-3}	2×10^{-2}
Qwen3-32B	128	197	163 (82.7)	129 (65.5)	3	1 / 2	2×10^{-9}	4×10^{-18}
Qwen3-32B	1024	180	146 (81.1)	74 (41.1)	20	6 / 4	8×10^{-6}	9×10^{-27}

Load arm. Table 8 and Figure 15 hold $d = 128$ fixed. The two extremes reproduce the short-arm and long-arm $d = 128$ cells on fresh items: at 32B combine is 95 and 66% here against 95.5 and 66.5% in the main arms; at 8B compute is 89 and 90% here against 90 and 70% in the main arms, so the main-arm difference at 8B is not a load effect (Section 4.3). Items are drawn independently at each load from the same seeded generator, so the arm is not paired across loads.

Diagnostic rerun of the 8B $d = 128$ cells. The main long arm (70%, $n = 200$; 69% on the same items under vLLM), the paired arm (76.5%, $n = 200$) and the load arm's 896-line cell (90%, $n = 100$) score compute on programs of identical structure (1,034 lines, probe at line 905, same header, line shapes and query). To separate item content from run-level variation, 100 compute and 100 combine items from each source were scored in one fresh Qwen3-8B engine. Each set reproduced its own level: compute 67, 73 and 90% (item agreement with the original runs 96, 100 and 100%), combine 12, 17 and 16% (99 to 100%). The spread is therefore a property of the item draws, about four standard errors under binomial sampling; no feature we examined (carry in the addition, probe digit count, the addend k) accounts for it, and the same generator produced all three sets. Smaller spreads of the same kind appear at 32B (compute 76.5, 82 and 87%) and for 14B combine (80 and 66%), whereas the $d = 1024$ cells replicate across sets (8B compute 43.5 and 50%, combine 1 and 1%).

L Reproducibility details

Models: Qwen/Qwen3-4B, -8B, -14B, -32B (post-trained checkpoints; weights obtained from the Hugging Face and ModelScope mirrors, 9 to 12 September 2026), deepseek-ai/DeepSeek-R1-Distill-Qwen-7B, meta-llama/Llama-3.1-8B-Instruct, google/gemma-3-

Table 8: Load arm: accuracy (%) at a fixed dependency distance of 128 lines as the number of lines before the value, and so the total context, grows ($n = 100$ per cell; 3.7K, 7K, 14K and 28K tokens). Last two columns: slope of accuracy on $\log(1 + p)$ with HC3 standard errors and its p-value.

model	query	$p = 0$	128	384	896	slope	p-value
Qwen3-8B	retrieve	98	98	100	99	+0.0020	0.41
Qwen3-8B	compute	89	85	87	90	−0.0009	0.88
Qwen3-8B	combine	25	12	18	16	−0.0138	0.082
Qwen3-14B	retrieve	98	98	99	99	+0.0014	0.57
Qwen3-14B	compute	95	96	94	91	−0.0038	0.39
Qwen3-14B	combine	79	90	82	66	−0.0077	0.34
Qwen3-32B	retrieve	99	98	99	99	−0.0002	0.93
Qwen3-32B	compute	92	96	87	87	−0.0065	0.25
Qwen3-32B	combine	95	95	90	66	−0.0278	1.8×10^{-6}

12b-it and mistralai/Mistral-7B-Instruct-v0.3 (the last three from the ModelScope mirror, 13 September 2026). Decoding: greedy, bfloat16, no chat template, 8 new tokens (direct) or 2,048 (scratchpad), vLLM seed 0, prefix caching disabled. Parsing: first integer on the first generated line (direct); last ANSWER: <n> (scratchpad). Software: transformers 5.x on the 4090; vLLM 0.29.0 with PyTorch 2.13 (CUDA 13.0) on the A100; analyses in Python 3.12 with statsmodels 0.15 (Seabold & Perktold, 2010). Compute: about 14 GPU-hours on the 4090 and 76 on the A100 (28 for the original sweep and replication, 22 for the sample-size extension, the 32B grid and the three additional families, and 26 for the paired, load, ablation and diagnostic runs); the 32B processes one 28K-token item in 13.1 s on the A100.