GATHER, DON'T SCATTER: TRAINABLE PLACEMENT ROBUSTNESS FOR MULTI-HOP REASONING IN MULTI-DOCUMENT LONG-CONTEXT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are increasingly given multiple documents in context to answer questions that require external, specialized, or up-to-date information. However, their answers can depend on how those documents are arranged. We examine this dependence in multi-hop question answering, where every document, gold or distractor, is placed directly in the context and each gold document supplies one step of the reasoning. We separately examine three aspects of document arrangement: the order of gold documents in the reasoning chain, their position in the context, and whether they are grouped together or scattered among distractors. Across two 8B base models, whether the gold documents are together or scattered matters most. Keeping them together helps consistently wherever the block sits, while chain order and exact block position have smaller effects. We then vary only the arrangement of the fine-tuning data. Training on examples with scattered gold nearly eliminates placement gaps and improves pooled accuracy by 20–24 percentage points, whereas training on a fixed gold-at-the-front layout induces severe brittleness when the gold moves. These findings hold across three datasets, with contexts up to 32k tokens, under harder retrieved distractors, and in long-form generation with citations. The main result also holds for a 27B model from a third family. Diagnostic probes further show that many remaining failures occur even when the individual supporting facts are accessible, identifying multi-hop composition as an important residual bottleneck.

1 INTRODUCTION

Language models are increasingly given a set of documents in context so that they can answer questions using information beyond their parameters (Xu et al., 2025). This setting is especially relevant for compact models, whose lower inference cost and latency make them attractive when long inputs must be processed. In practice, however, documents may be retrieved or assembled in an order unrelated to the reasoning needed to answer the question. This matters because models can miss even a single relevant fact when it appears in the middle of a long context (Liu et al., 2024).

Multi-hop questions make this problem harder and qualitatively different. The evidence needed to answer them is spread across several supporting, or gold, documents, with each document supplying one step, or hop, of the reasoning (Yang et al., 2018; Trivedi et al., 2022). The model must find all of these documents and connect their information while ignoring the surrounding distractors. With several gold documents, arrangement no longer refers only to where one relevant fact appears. It also includes whether the documents follow the reasoning chain and whether they are gathered together or scattered across the context. We therefore ask which aspects of document arrangement most affect multi-hop question answering, and whether models can be trained to remain accurate when the evidence arrives in an arbitrary layout. Figure 1 summarizes our controlled evaluation and training comparison.

We answer these questions through a controlled study that holds the documents fixed while varying three aspects of their arrangement. We vary the order of the gold documents within the reasoning chain, their position in the context, and whether they are together as one block or scattered among the distractors. Across two 8B base models, the last factor matters most. Gathering the gold documents

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

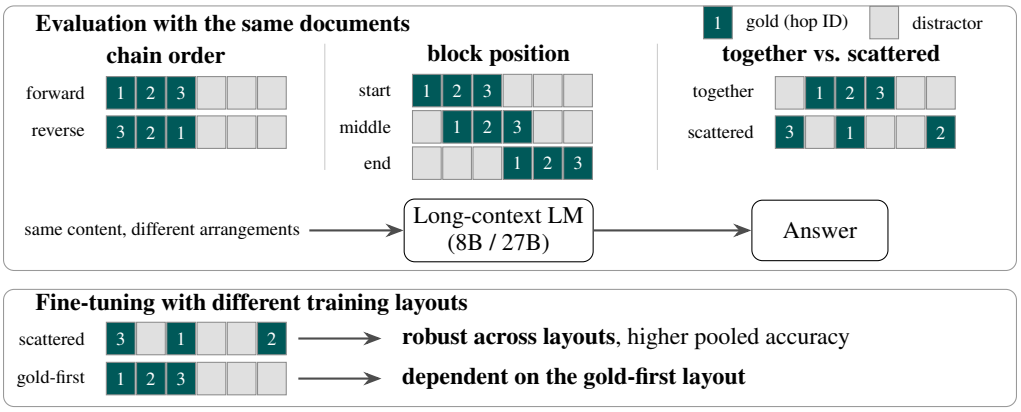


Figure 1: Controlled evaluation and training intervention. We hold document content fixed while varying chain order, block position, and whether the gold documents are together or scattered. During fine-tuning, only the training layout changes. Scattered training yields robustness across layouts, whereas gold-first training induces dependence on the front layout.

helps consistently, while chain order and exact block position have smaller effects. We then change only the arrangement of the fine-tuning data. Training on scattered gold nearly eliminates placement gaps and improves pooled accuracy by 20–24 percentage points, whereas training on a fixed gold-at-the-front layout makes the models brittle when the evidence moves. Together, these results show that the layout of the training examples strongly shapes robustness to evidence placement.

We test the scope of this result across two unseen datasets, contexts up to 32k tokens, harder retrieved distractors, reasoning-enabled inference, and long-form generation with citations. We also reproduce the main result on a 27B model from a third family. Diagnostic probes show that many failures remain even when the individual supporting facts are accessible, identifying multi-hop composition as an important residual bottleneck. Our contributions are given below.

- We disentangle three aspects of document arrangement in multi-hop question answering and show that whether the gold documents are together or scattered matters more consistently than their chain order or block position.
- We show that the arrangement of the fine-tuning data strongly shapes placement robustness. Scattered training nearly eliminates placement gaps and improves accuracy, while fixed gold-first training induces brittleness.
- We validate the result across datasets, context lengths, distractor types, reasoning settings, long-form generation, and model families. Our diagnostic analyses further identify multi-hop composition as an important source of residual error.

2 Related Work

Document-based QA and evidence placement. Document-based QA systems answer questions from a set of retrieved or provided documents, requiring the reader to find relevant evidence while ignoring distractors. Irrelevant context degrades these readers (Shi et al., 2023; Yoran et al., 2024), especially when distractors are retrieved or resemble the gold documents (Jin et al., 2025; Bhuiya et al., 2024). Performance also depends on evidence placement: models use a relevant passage more reliably at the start or end of a long context than in the middle (Liu et al., 2024), and are sensitive to what surrounds it (Cuconasu et al., 2024). Long-context evaluations range from single-target retrieval (Kamradt, 2023) to broad suites (Hsieh et al., 2024a; Bai et al., 2024) and tests with several dispersed facts (Wang, 2025). However, they largely study retrieval or general long-context capabilities, or hold the arrangement of jointly supporting evidence fixed.

Mitigating position and order sensitivity. These failures motivate methods that make readers less dependent on where evidence appears. Input-level remedies filter irrelevant passages, control retrieval through self-reflection (Yoran et al., 2024; Asai et al., 2024), or reorder retrieved

Table 1: Comparison with closely related work. *Causal* denotes a controlled training-layout intervention that both removes placement sensitivity (scattered training) and induces it (fixed-layout training).

Method	Multi-hop	Isolates gath./scat.	Minimal fix (data only)	Causal (remove+induce)	Robust 32k+hardneg	Long-form + cites
Lost in the Middle (Liu et al., 2024)	✗	✗	✗	✗	✗	✗
Found in the Middle (Hsieh et al., 2024b)	✗	✗	✗	✗	✗	✗
PINE (Wang et al., 2025)	✗	✗	✗	✗	✗	✗
Set-LLM (Egressy & Stühmer, 2025)	✗	✗	✗	✗	✗	✗
FILM (An et al., 2024)	✗	✗	✗	✗	✗	✗
Never Lost in the Middle (He et al., 2024)	✓	✗	✗	✗	✗	✗
Long-Context LLMs Meet RAG (Jin et al., 2025)	✓	✗	✗	✗	✓	✗
Ours	✓	✓	✓	✓	✓	✓

documents (Jin et al., 2025). Inference-time and architectural methods calibrate positional attention (Hsieh et al., 2024b), scale a hidden dimension (Yu et al., 2025), make inter-document attention order-invariant (Wang et al., 2025), or enforce permutation invariance (Egressy & Stühmer, 2025). Training-time methods use information-intensive or position-agnostic objectives (An et al., 2024; He et al., 2024), shuffled long-context pre-training (Tian et al., 2025), or consistency and rank distillation across passage orders (Lee et al., 2025).

Our setting and intervention. We study multi-hop QA, where several gold documents jointly support the answer, and independently vary their chain order, block position, and consolidation (together versus scattered). Our training intervention changes only the arrangement of these documents, adding no task, loss, module, or architectural constraint. We use *causal* in this controlled intervention sense: scattered training removes the model's accuracy gap across layouts, whereas fixed gold-first training induces a preference for that layout. Thus, changing only the training layout moves placement sensitivity in both directions. Table 1 summarizes these distinctions. Reasoning at inference time is orthogonal; Section 4.3 compares it with our intervention.

3 Experimental Setup

We study multi-hop question answering over a set of documents provided in context. The gold documents jointly support the answer, with one document corresponding to each reasoning hop; the remaining documents are distractors. We place the entire document set in the context and ask the model to produce an answer, without using a retriever.

Arrangement conditions. With multiple supporting documents, arrangement concerns not only where the evidence appears, but also whether it follows the reasoning chain and whether it is gathered or dispersed. Let the gold documents be k passages, where k is the number of reasoning hops. We obtain this one-gold-per-hop structure from the dataset annotations, which identify exactly one supporting paragraph for each hop; we do not impose it as a modeling assumption.

We construct seven arrangements. In the first six, the gold documents are *together* in a contiguous block. Terms *forward* and *reverse* specify their order within that block: forward follows the reasoning chain from hop 1 to hop k, whereas reverse runs from hop k to hop 1. Terms *first*, *middle*, and *last* specify where the block appears relative to the distractors. Crossing the two chain orders with the three block positions gives the six structured arrangements – *gold forward, first*, *gold reverse, first*, *gold forward, last*, *gold reverse, last*, *gold forward, middle*, *gold reverse, middle*, and *scattered*.

In the seventh arrangement, *scattered*, we remove the block and place the k gold documents at independent random positions among the distractors. Within each example, we hold the document content fixed and change only the placement of the gold documents. We average the scattered condition over three random seeds. Separately, we evaluate a *closed-book* condition with no documents as a memorization control.

Evaluation. We primarily evaluate Llama-3.1-8B-Instruct (Dubey et al., 2024) and Qwen3-8B (Yang et al., 2025) on 150 MuSiQue validation questions (Trivedi et al., 2022), balanced across 2, 3, and 4 hops, and test transfer to 2WikiMultiHopQA (Ho et al., 2020) and HotpotQA (Yang et al., 2018). HotpotQA identifies the gold documents but not their reasoning order, so we vary

Table 2: Base LMs on MuSiQue by arrangement (mean, with standard error in parentheses; $n = 150$). Accuracy barely changes across chain order (forward vs. reverse) and block position (first/last/middle). The scattered baseline is the lowest for both models. "Structured mean" averages the six together arrangements.

Arrangement	Llama `em_soft`	Qwen `em_soft`	Llama judge	Qwen judge
gold forward, first	0.467 (±.041)	0.440 (±.041)	0.593	0.593
gold reverse, first	0.440 (±.041)	0.467 (±.041)	0.567	0.620
gold forward, last	0.493 (±.041)	0.407 (±.040)	0.653	0.580
gold reverse, last	0.507 (±.041)	0.407 (±.040)	0.653	0.573
gold forward, middle	0.440 (±.041)	0.393 (±.040)	0.560	0.533
gold reverse, middle	0.453 (±.041)	0.407 (±.040)	0.600	0.573
structured mean	0.467	0.420	0.604	0.579
scattered	**0.398** (±.036)	**0.362** (±.036)	**0.538**	**0.507**

only position and together-vs. scattered. We use Gemma-3-27B-it (Gemma Team, 2025) to test a larger model from a third family, and evaluate long-form generation on ALCE and controlled MuSiQue (Stelmakh et al., 2022; Amouyal et al., 2023; Fan et al., 2019; Gao et al., 2023). We report `em_soft`, a formatting-robust containment match, and accuracy from a self-hosted Llama-3.3-70B-Instruct judge that sees only the question, gold answer, and candidate answer; long-form evaluation uses ALCE correctness and judge-based citation entailment. Unless noted otherwise, models answer directly under greedy decoding. Appendices B and A give full data, model, metric, and decoding details.

Training regimes. We fine-tune with LoRA (Hu et al., 2022) on MuSiQue training questions disjoint from evaluation, changing only the arrangement of the training documents. In the *scattered* regime, we place each gold document at an independently sampled position; in the *gold-first* regime, we place the gold documents in a forward block at the start. We hold the prompt, training procedure, and evaluation fixed, so differences between the resulting LMs isolate the training arrangement. Section 4.4 applies the scattered regime to cited long-form answers. Appendices A and D provide the implementation details and leakage audit.

4 Results

We first identify which aspect of document arrangement affects base LMs. We then ask whether this sensitivity can be changed through training and what errors remain once placement becomes irrelevant. Finally, we test whether the resulting robustness survives longer contexts, harder distractors, reasoning, long-form generation, and changes in model scale and family.

4.1 Effect of Document Arrangement

At the base model, together vs. scattered matters most. Table 2 shows a simple pattern: both base models perform better when the gold documents form a contiguous block than when they are scattered among distractors. On judge accuracy, the structured mean exceeds the scattered condition by 6.6 points for Llama and 7.2 points for Qwen. In contrast, moving the gold block from the beginning to the middle or end, or reversing the order of the reasoning chain, produces only small and inconsistent changes. The lower absolute `em_soft` scores tell the same story. Thus, whether the evidence is gathered matters more than its exact position or order. Paired tests support these conclusions (Appendix C), and a closed-book control rules out parametric recall as the source of the differences (Appendix E).

Placement robustness is trainable Changing only the arrangement of the training data reverses the models' behavior (Table 4). With scattered training, accuracy is nearly constant across arrangements: the front-minus-scattered gap falls to $+0.020$ for Llama and $+0.002$ for Qwen, while pooled `em_soft` rises from 0.457 to 0.697 and from 0.412 to 0.597, respectively. Gold-first training has the opposite effect. It preserves high accuracy when the gold block appears first but causes sharp drops when that block moves to the middle or end, or when the gold documents are scattered. Placement

Table 4: Training controls the position effect in both directions (MuSiQue, mean em_soft ± standard error; $n = 150$). Scattered training improves accuracy and nearly eliminates the front−scat gap. Gold-first training favors the front but reduces accuracy below the base model at the end and in scattered contexts. The last column uses a pooled judge; "Pooled" averages all seven arrangements.

Model	Regime	front	middle	end	scattered	front−scat	pooled em	pooled judge
Llama	base	0.467 (±.04)	0.440 (±.04)	0.493 (±.04)	0.398 (±.04)	+0.069	0.457	0.595
Llama	scattered	0.707 (±.04)	0.720 (±.04)	0.700 (±.04)	0.687 (±.03)	+0.020	**0.697**	**0.784**
Llama	gold-first	0.707 (±.04)	0.273 (±.04)	0.113 (±.03)	0.229 (±.02)	+0.478	0.341	0.397
Qwen	base	0.440 (±.04)	0.393 (±.04)	0.407 (±.04)	0.362 (±.04)	+0.078	0.412	0.568
Qwen	scattered	0.600 (±.04)	0.633 (±.04)	0.593 (±.04)	0.598 (±.04)	+0.002	**0.597**	**0.685**
Qwen	gold-first	0.680 (±.04)	0.373 (±.04)	0.213 (±.03)	0.302 (±.03)	+0.378	0.389	0.430

sensitivity is therefore learned rather than intrinsic: the training arrangement can either remove it or create it.

The fix transfers to unseen datasets. This learned behavior persists beyond MuSiQue. Without further training, the scattered-trained LoRAs keep the front-minus-scattered gap within ± 0.013 on both 2WikiMultiHopQA and HotpotQA (Table 3). The gold-first-trained LoRAs remain brittle, with gaps of 0.127–0.178, all significant at $p < .001$, while the base models show no significant gap. Training arrangement thus controls placement sensitivity even on datasets that the LoRAs never saw.

Table 3: Cross-dataset transfer (em_soft; gap = gold-at-front − scattered; paired McNemar p). Scattered training keeps the gap near zero on unseen datasets; gold-first training stays brittle. A star marks $p < 0.05$.

Model / regime	2Wiki gap (p)	HotpotQA gap (p)
Llama base	-0.057 (.16)	$+0.036$ (.07)
Llama scattered	$+0.007$ (.85)	$+0.002$ (.84)
Llama gold-first	$+0.143$ ($< .001$)*	$+0.178$ ($< .001$)*
Qwen base	$+0.017$ (.59)	$+0.013$ (.52)
Qwen scattered	-0.013 (.69)	-0.013 (.52)
Qwen gold-first	$+0.137$ ($< .001$)*	$+0.127$ ($< .001$)*

4.2 Robust models find the evidence but struggle to compose it

A natural concern is that the scattered-trained LoRAs learned a shortcut rather than how to read dispersed evidence. We test this possibility with three diagnostics, all conducted without further training (full results in Appendix F).

Gold-only ceiling. We remove all distractors and give the LM only the gold documents. Even in this idealized setting, judge accuracy improves by only 8 points for Llama and 10 points for Qwen over the scattered condition. Thus, placement accounts for only part of the remaining error and a substantial reasoning gap persists even when the input is perfect.

Per-hop probe. We evaluate each reasoning hop in isolation. Single-hop accuracy is high (0.76–0.80), yet roughly 40% of full-question failures occur even when the model answers every constituent hop correctly (12 of 30 failures for Llama; 21 of 51 for Qwen). The bottleneck is therefore not recovering individual facts, but composing them into a complete reasoning chain.

Attention to the gold. We measure the last token's attention to tokens in the gold documents. Relative to a token-count baseline of about 0.10, failed predictions assign at least as much attention to the gold as successful ones at the final layer (Llama: 0.166 vs. 0.104; Qwen: 0.166 vs. 0.113). The failures therefore show no evidence of arising from an inability to locate the gold evidence.

Together, these diagnostics indicate that the robust models locate and process the gold evidence wherever it appears, but still struggle to compose that evidence across hops. Improving this multi-hop composition is an important direction for future work.

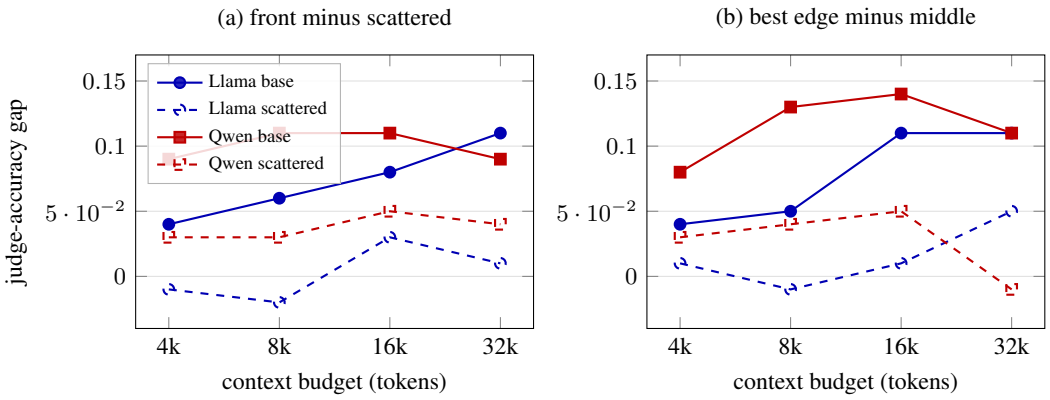


Figure 2: Placement gaps across context lengths on MuSiQue (judge accuracy, $n = 150$). The front-minus-scattered gap is accuracy with the gold block at the front minus accuracy with the gold documents scattered; the edge-minus-middle gap is accuracy at the better edge position minus accuracy in the middle. The gaps are generally larger for the base LMs (solid) and remain near zero for the scattered-trained LoRAs (dashed). Exact results, standard errors, and significance tests appear in Table 14 (Appendix H).

4.3 Placement robustness persists under harder conditions

The standard MuSiQue setting uses short contexts and rule-based distractors. We test whether the placement robustness learned through scattered training persists under three more demanding conditions: longer contexts, harder distractors, and inference-time reasoning.

Longer contexts. We lengthen each context by inserting short, topic-neutral passages generated by the 70B model and sampled from a fixed pool. This procedure preserves the gold and distractor documents while varying only the context length. We use a single reference tokenizer to ensure that both LMs receive the same document set at each budget, which ranges from 4k to 32k tokens. Figure 2 reports two judge-accuracy contrasts: the front-minus-scattered gap (gold block at the front minus scattered gold) and the edge-minus-middle gap (the better of the front and end positions minus the middle position). For the base LMs, these placement gaps generally widen as the context grows: they become significant for Llama by 16k and remain substantial for Qwen throughout. In contrast, both gaps remain small for the scattered-trained LoRAs at every context length, while their absolute accuracy remains about 0.20 higher. Scattered training therefore confers robustness beyond the short-context setting. Further details given in Table 14 (Appendix H).

Harder distractors. We replace the rule-based distractors with hard negatives retrieved by BM25 from a large pool of real passages. For each question, we select the highest-ranked non-gold passages and exclude any that contain the answer string. Because these passages are topically related to the question, they are harder to dismiss than random distractors. Under this condition, the front-minus-scattered gap increases for the base models: it reaches $+0.107$ for Qwen ($p = 0.004$) and $+0.053$ for Llama ($p = 0.15$). The scattered-trained LoRAs remain insensitive to placement under the same distractors: no contrast is significant and all satisfy $|\Delta| \leq 0.04$. By comparison, model-generated distractors do not enlarge the gap because they are no harder than the native distractors. Retrieved hard negatives provide a stronger stress test. Table 5 compares all three distractor sources.

Length on an unseen dataset. We repeat the same length sweep, with the same neutral filler and context budgets, on 2WikiMultiHopQA using the MuSiQue-trained LMs. Table 6 reports both placement gaps at each budget. The pattern transfers: the edge-minus-middle gap for the base models grows with context length (from 0.10 at 4k to 0.15 at 32k for Qwen), whereas it remains at most 0.06 for the scattered-trained LoRAs. The front-minus-scattered gap remains small for both because the 2Wiki base models are already relatively robust to dispersed evidence. Thus, the robustness to longer contexts transfers to a dataset not seen during training.

Reasoning also reduces placement sensitivity. All short-answer results so far use direct decoding. We repeat the seven-condition MuSiQue study with inference-time reasoning enabled and all other

Table 5: Front-minus-scattered judge-accuracy gap at 4k under three distractor sources ($n = 150$; paired McNemar test). The gap is significant for both base LMs under random distractors and for Qwen under BM25-retrieved hard negatives, but not under model-generated distractors. The scattered-trained LoRAs remain insensitive to placement across all three conditions. A star marks $p < 0.05$.

Reader	random	generated hard neg.	retrieved hard neg.
Llama base	$+0.100^{\star}$	$+0.020$	$+0.053$
Qwen base	$+0.100^{\star}$	$+0.027$	$+0.107^{\star}$
Llama scattered	n.s. ($\leq .04$)	n.s. ($\leq .03$)	n.s. ($\leq .04$)
Qwen scattered	n.s. ($\leq .04$)	n.s. ($\leq .04$)	n.s. ($\leq .04$)

Table 6: Placement gaps across context lengths on 2WikiMultiHopQA using the MuSiQue-trained LMs (judge accuracy, $n = 100$). "front−scat" denotes the front-minus-scattered gap, and "edge−mid" denotes the edge-minus-middle gap. The latter grows with context length for the base models but remains small for the scattered-trained LoRAs. The models never saw 2WikiMultiHopQA during training.

	4k		8k		16k		32k	
LM	front−scat	edge−mid	front−scat	edge−mid	front−scat	edge−mid	front−scat	edge−mid
Llama base	$+.07$	.05	$+.02$	.04	$+.08$	.06	$+.07$	.10
Llama scattered	$+.02$	.03	$+.03$	.03	$+.07$	.03	$+.06$	.04
Qwen base	$+.03$	.10	$+.02$	.14	$+.06$	.15	$+.01$	.15
Qwen scattered	$+.02$	.03	$+.07$	.03	$+.06$	.05	$+.05$	.06

settings fixed. Qwen3-8B uses its native thinking mode, whereas Llama-3.1-8B uses prompted chain-of-thought; we therefore compare each reasoning condition with its corresponding direct-answer baseline. Two findings remain unchanged: chain order has little effect (forward vs. reverse $|\Delta| \leq 0.03$, all $p > 0.4$), and the position of the gold block remains largely irrelevant (best edge vs. middle ≤ 0.02). The front-minus-scattered gap, however, narrows because reasoning disproportionately improves the harder scattered condition. It falls from $+0.087$ to $+0.011$ for Qwen and from $+0.056$ to $+0.031$ for Llama (Table 7).

Reasoning therefore provides a second, inference-time route to robustness against dispersed evidence. This result is consistent with Section 4.2: once the model locates the relevant facts, explicit reasoning can help compose them. Scattered training achieves a similar benefit without generating reasoning tokens at inference time. Its direct-answer accuracy exceeds that of the reasoning base model for Llama (0.784 vs. 0.594 pooled judge accuracy) and comes within 0.048 for Qwen (0.685 vs. 0.733). Scattered training thus offers a lower-inference-cost route to placement robustness.

Table 7: Reasoning versus scattered training on MuSiQue (judge accuracy). "Pooled" averages the seven arrangements; "scattered" is the scattered condition; and "front−scat" is the front-minus-scattered gap. Reasoning narrows this gap. Without generating reasoning tokens, the scattered LoRA outperforms the reasoning base model for Llama and approaches it for Qwen. The Llama prompted-CoT results are lower bounds because the model often does not terminate within the token budget.

Setting	Pooled	Scattered	Front−scat
Llama-3.1-8B			
base, direct answer	0.595	0.538	$+0.056$
base, + prompted CoT	0.594	0.580	$+0.031$
scattered LoRA, direct	0.784	0.787	$+0.013$
Qwen3-8B			
base, direct answer	0.569	0.507	$+0.087$
base, + native thinking	0.733	0.709	$+0.011$
scattered LoRA, direct	0.685	0.676	-0.002

Llama's prompted-CoT results are lower bounds because 18% of runs fail to terminate, especially on four-hop questions. Increasing the token budget from 1,024 to 12,288 reduces this rate from 29% to 18% but no further; Qwen's native thinking mode truncates in fewer than 1% of cases (Appendix I).

Table 8: Controlled long-form MuSiQue: structured (gold gathered as one block, averaged over the front, middle, and last placements) vs. scattered, by context budget ($n = 150$; 148 for Qwen at 32k). Each cell is structured / scattered. Coverage is the fraction of gold hop-facts stated; citation recall is judged by the 70B model. Both base readers collapse with length, while the scattered-trained LoRA stays much flatter. Gathering the gold beats scattering it for the base readers, and the coverage gap grows up to 16k; the scattered-trained LoRA keeps the two close, except Qwen coverage at 32k. Per-position results are in Appendix G.

	coverage (%), struct / scat			citation recall, struct / scat		
Reader	4k	16k	32k	4k	16k	32k
Llama base	81.5 / 77.3	76.1 / 66.0	50.4 / 44.7	38.8 / 40.1	33.5 / 31.6	11.6 / 11.2
Llama LoRA	81.7 / 80.4	78.3 / 76.4	76.6 / 74.7	42.7 / 41.7	38.2 / 37.3	34.6 / 33.9
Qwen base	76.9 / 77.1	77.1 / 73.3	71.0 / 67.7	25.5 / 25.7	23.8 / 21.9	17.5 / 14.7
Qwen LoRA	78.3 / 77.6	78.7 / 77.1	73.1 / 67.3	38.4 / 39.0	36.1 / 34.7	33.2 / 29.9

4.4 Long-form generation with citations

So far the answer is one short phrase. We now test long-form answers that must cover many facts and cite the passages that support them. To build a scattered-trained long-form LoRA we distill from the 70B model: for each training question we give the 70B the passages in scattered order and have it write a long answer with inline citations, keep only the answers whose final content matches the gold (rejection sampling), and fine-tune the 8B with LoRA on these cited answers, one such LM per base model. This teaches the format and the citing behavior while keeping the passage order uninformative, exactly as in the short-form scattered regime.

Standard benchmarks (ALCE). On three standard retrieval-based long-form citation benchmarks (ASQA, QAMPARI, and ELI5) the scattered-trained long-form LoRA improves correctness and, most clearly, citation faithfulness over the base model, and the passage order (sorted vs. scattered) has no measurable effect. These datasets have no annotated gold and only about 1.5k tokens of context, so they cannot test gold placement and do not isolate the training arrangement; the full results are in Appendix G (Table 13). We therefore use the controlled setting below as our long-form placement test.

Controlled long-form MuSiQue. To test placement in long-form without those confounds, we build a controlled long-form version of MuSiQue, where the gold passages and their positions are known, and we vary both the arrangement of the gold and the context length as before. Table 8 contrasts the structured average (the gold gathered as one block, averaged over the front, middle, and last placements) with the scattered arrangement at each budget; per-position results are in Appendix G. Two points stand out. First, length dominates. As the context grows from 4k to 32k tokens, base coverage falls sharply (Llama structured from 82% to 50%) and base citation recall collapses (Llama structured from 39 to 12), while the scattered-trained LoRA stays much flatter (coverage from 82% to 77%, citation from 43 to 35). Second, gathering the gold helps the base readers, and the coverage gap grows with length up to 16k before the collapse compresses it: the structured-minus-scattered coverage gap for base Llama goes from $+4$ at 4k to $+10$ at 16k, and for base Qwen from about 0 to $+4$. The scattered-trained LoRA keeps the two arrangements close (Llama within $+2$ at every budget), with one exception, Qwen coverage at 32k, where a gap of about $+6$ remains. So the short-answer story carries over to long-form, but the together-versus-scattered effect is milder here and the length collapse is the larger force.

4.5 Scale and model family

Our two main LMs are 8B models from two families. To check that the story is not specific to that size or family, we repeat the base, trainable, and length experiments on Gemma-3-27B-it, a larger model from a third family. The full arc reproduces (Table 9). The base model is placement sensitive: gold-at-end beats scattered by $+0.120$ ($p \approx 0$). Scattered training makes it placement-invariant and more accurate: it lifts the scattered score from 0.647 to 0.793 and every arrangement lands near

Table 9: Gemma-3-27B-it reproduces the arc (MuSiQue, $n = 150$; forward chain order shown, reverse tracks it closely). All numbers are 70B-judge accuracy, our primary metric. The base model is placement sensitive; scattered training removes it and raises accuracy; gold-first training induces brittleness. "cb" is closed book. A star marks $p < 0.05$ on the listed contrast.

Regime	front	middle	end	scattered	cb	pooled judge	key contrast
base	0.673	0.700	0.767	0.647	0.240	0.695	end − scat = $+0.120^\star$
scattered	0.767	0.780	0.800	0.793	0.220	**0.781**	front − scat = -0.027
gold-first	0.780	0.553	0.520	0.511	0.233	0.601	front − scat = $+0.269^\star$

0.77 to 0.80. Gold-first training makes it brittle: gold-at-front 0.780 vs. scattered 0.511 ($+0.269$, $p \approx 0$), which is below the base model off the trained position. In the length sweep the base spread grows from 0.067 at 4k to 0.153 at 32k (`em_soft`), while the scattered-trained LoRA stays near zero (0.000 to 0.020) and uniformly higher.

5 Discussion

Why scattered training works A base LM appears to rely on a positional heuristic: evidence at the edges of the context, or gathered into one block, is easier to use than evidence in the middle or spread out. This heuristic is cheap when support is conveniently placed, but it fails when the gold is dispersed or the context is long. Scattered training removes the heuristic by making position uninformative during fine-tuning, so the only stable cue to a gold document is its content and the LM learns content-based selection. This matches the diagnostics in Section 4.2, where the trained LM attends to the gold tokens wherever they sit and its remaining errors are about combining facts, not finding them. Gold-first training does the opposite: it makes position maximally informative during fine-tuning, so the LM comes to depend on it and breaks when it changes.

Practical recommendation. The fix is a drop-in change to the training data, not a new model, loss, or module. If you fine-tune a long-context LM to read many documents, scatter the gold across the context in the training inputs: it costs nothing beyond the normal fine-tuning run, it makes the LM answer equally well wherever the evidence lands, and it adds about 20 percentage points of accuracy on multi-hop reading. Just as important is what to avoid: do not fine-tune on a fixed, tidy layout with the gold gathered at the front, even though that is the easy thing to build, because it makes the LM depend on that layout and fail when real inputs differ. We therefore recommend scattered-gold fine-tuning as a default recipe for multi-document long-context LMs, and expect it to help most when contexts are long or the relevant passages are dispersed by a retriever.

6 Conclusion

In multi-document multi-hop reading, what governs a long-context model is whether the gold documents are gathered or scattered, not where the block sits or in what order the gold is read. This sensitivity is trainable. Training on scattered gold makes an LM that answers well wherever the evidence lands and that is more accurate, while training on a fixed layout makes it brittle. The fix holds across datasets, context lengths, harder retrieved distractors, long-form generation with citations, and a larger model from a third family. Once placement stops mattering, the errors that remain come from combining facts across hops. That composition problem, not placement, is the next thing to solve.

Limitations. The larger-model result is a single confirmatory run per regime, not a full scaling curve. The two main LMs are 8B models from two families, and all data is in English. The scattered training recipe is simple by design; we did not search over training arrangements beyond the two extremes. The residual composition errors are diagnosed but not fixed. In the reasoning ablation, prompted chain-of-thought on Llama-3.1-8B often did not terminate within budget (about 18% of cases, worst at four hops), so its reasoning scores there are a lower bound; Qwen native thinking truncated on under 1%.

AI USE STATEMENT

The authors used a generative AI coding assistant to help implement and orchestrate the experiment harness and to help draft and edit the text of this paper. All experiment code was reviewed and run by the authors, and all numerical results were produced by the authors' own runs and checked against the saved result files. All claims in the paper were verified by the authors against those results. The authors take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

Section 3 describes the task, arrangements, datasets, models, metrics, and training recipe. Appendix A gives the exact model checkpoints, hyperparameters, decoding settings, dataset construction parameters, software versions, and hardware needed to reproduce every result. Appendix B gives the source, split, and size of every dataset, and of the data used to fine-tune the LoRAs. Appendix D reports the leakage audit and Appendix E the memorization control. All runs use greedy decoding with fixed seeds, and every run records its configuration and library versions.

REFERENCES

Samuel Joseph Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. QAMPARI: A benchmark for open-domain questions with many answers. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pp. 97–110, 2023.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. Make your LLM fully utilize the context. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3119–3137, 2024.

Neeladri Bhuiya, Viktor Schlegel, and Stefan Winkler. Seemingly plausible distractors in multi-hop reasoning: Are large language models attentive readers? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Abhimanyu Dubey et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Beni Egressy and Jan Stühmer. Set-LLM: A permutation-invariant LLM. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

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

Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, Yibo Liu, Qianguo Sun, Yuxin Liang, Hao Wang, Enming Zhang, and Jiaxing Zhang. Never lost in the middle: Mastering long-context question answering with position-agnostic decompositional training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

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 *Conference on Language Modeling (COLM)*, 2024a.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14982–14995, 2024b.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan Ö. Arık. Long-context LLMs meet RAG: Overcoming challenges for long inputs in RAG. In *International Conference on Learning Representations (ICLR)*, 2025.

Greg Kamradt. Needle in a haystack: Pressure testing LLMs. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`, 2023.

Youngwon Lee, Seung-won Hwang, Daniel Campos, Filip Graliński, Zhewei Yao, and Yuxiong He. CORD: Balancing COnsistency and rank distillation for robust retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 787–796, 2025.

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.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 31210–31227, 2023.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Junfeng Tian, Da Zheng, Yang Chen, Rui Wang, Colin Zhang, and Debing Zhang. Untie the knots: An efficient data augmentation strategy for long-context pre-training in language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1223–1242, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Yidong Wang. Reasoning on multiple needles in a haystack. *arXiv preprint arXiv:2504.04150*, 2025.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. Eliminating position bias of language models: A mechanistic approach. In *International Conference on Learning Representations (ICLR)*, 2025.

Peng Xu, Wei Ping, Xianchao Wu, Chejian Xu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 67244–67258, 2025.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations (ICLR)*, 2024.

Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. Mitigate position bias in LLMs via scaling a single hidden states channel. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6092–6111, 2025.

A Reproducibility details

This appendix gives the exact settings behind every result in the paper: model checkpoints, decoding and serving settings, LoRA hyperparameters, dataset construction parameters, software versions, and hardware. Section 3 gives the task, arrangement, dataset, and metric definitions at a higher level; this appendix adds the concrete numbers needed to rerun the experiments.

Model checkpoints. We use four public checkpoints, unmodified, from Hugging Face: `meta-llama/Llama-3.1-8B-Instruct` and `Qwen/Qwen3-8B` as the two main readers, `google/gemma-3-27b-it` as the third-family reader in Section 4.5, and `meta-llama/Llama-3.3-70B-Instruct` as the judge and as the teacher model for long-form distillation and neutral filler text.

Decoding. All generation is greedy (temperature 0). This holds for the readers, the judge, and the 70B teacher. The maximum generation length differs by role: 512 tokens for Llama-3.1-8B, 1024 tokens for Qwen3-8B, 2048 tokens for Gemma-3-27B-it, and 128 tokens for the judge. Qwen3-8B's native thinking mode is turned off for the short-answer runs (`enable_thinking: false`) and turned on only for the reasoning ablation in Section 4.3.

Serving. We serve every model with vLLM (version 0.26.0) on one node with 4 A100-80GB GPUs. Each 8B reader runs with tensor-parallel size 1 and two replicas, one per GPU. Gemma-3-27B-it runs with tensor-parallel size 2, as two replicas across the four GPUs. The 70B judge runs with tensor-parallel size 4, one replica across all four GPUs, because a 70B model in bf16 does not fit on one 80GB GPU. GPU memory utilization is 0.90 for the 8B readers and the judge, and 0.92 for Gemma. The maximum context length is 8192 tokens for the 8B readers, 4096 tokens for the judge, and 32768 tokens for Gemma. We do not set an explicit dtype flag, so vLLM uses each checkpoint's native dtype, which is bf16 for all four models.

LoRA fine-tuning. We fine-tune with the Hugging Face `Trainer` and PEFT. The LoRA hyperparameters are the same for Llama, Qwen, and Gemma: rank 16, alpha 32, dropout 0.05, applied to `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. We optimize with AdamW, learning rate 10^{-4}, a cosine schedule, warmup for the larger of 5 steps or 3% of total steps, and no weight decay, for 2 epochs. Micro-batch size is 1 with 16 gradient accumulation steps, so the effective batch size is 16. Training uses bf16 precision and gradient checkpointing. The random seed is 12345 for every fine-tuning run, and it controls weight initialization, data shuffling, and the train/dev split. For short-answer training, the target is one line containing "Answer:" followed by the gold answer; we mask all prompt tokens, compute loss only on this completion, and use no reasoning traces. The long-form LoRAs instead target cited multi-sentence answers distilled from

the 70B model, with the prompt tokens likewise masked. The maximum sequence length is 4096 tokens for the short-answer LoRAs and 5120 tokens for the long-form ALCE LoRAs of Section 4.4, which need room for the retrieved-passage prompts. Each LoRA job uses one GPU, so the four short-answer fine-tunes, Llama and Qwen in the scattered and gold-first regimes, run at the same time on the 4-GPU node.

Quantized fine-tuning for the 27B model. For Gemma-3-27B-it we fine-tune with 4-bit QLoRA (Dettmers et al., 2023): bitsandbytes NF4 quantization, bf16 compute dtype, and double quantization, on a single GPU. Every other LoRA hyperparameter is the same as above.

Software versions. Python 3.11. Serving environment: vLLM 0.26.0. Training environment: PyTorch 2.11 (CUDA 13.0), Transformers 5.15, PEFT 0.20.0, Accelerate 1.14.0, and bitsandbytes 0.50.1 (only needed for the Gemma QLoRA runs).

Dataset construction. Each MuSiQue example has 20 paragraphs in total. We sample 50 answerable questions per hop count (2, 3, and 4 hops, for 150 examples), with data seed 0, and use every non-gold paragraph as a distractor, so the model always sees the full $\sim$20-paragraph context. HotpotQA uses the standard distractor setting, which fixes 10 paragraphs per example (2 gold and 8 distractor). 2WikiMultiHopQA uses all candidate paragraphs per example, with hop counts 2 and 4. The scattered arrangement is averaged over 3 fixed seeds (0, 1, 2) that set the random gold positions; this seed is independent of the data-sampling seed.

Retrieved hard negatives. We build a BM25 index (Okapi BM25, $k_1 = 1.5$, $b = 0.75$) over a pool of 500 examples per hop count. For each question we retrieve the top 18 non-gold passages, drop any candidate whose text or title contains one of the gold answer strings, so a hard negative can never leak the answer, and pad with the example's native distractors if fewer than 18 candidates survive the filter.

Length sweep. We grow the context with neutral filler passages generated by the 70B model, and size every budget (4k, 8k, 16k, and 32k tokens) with the Llama-3.1-8B-Instruct tokenizer as one shared reference, so both readers see an identically sized document set at each budget. The controlled long-form length sweep of Section 4.4 uses budgets 4096, 16384, and 30720 tokens with the same reference tokenizer, the four conditions gold-forward-first, gold-forward-middle, gold-forward-last, and scattered, 50 examples per hop count, and data seed 0.

Long-form data (ALCE). Each ALCE prompt uses the top 10 retrieved passages and 2 few-shot demonstrations, a 300-example test split, and data seed 0. The scattered passage order is a fixed, deterministic shuffle per example, seeded by a hash of the example id and dataset name, so it is reproducible without storing the permutation. To build the scattered-trained long-form LoRA, we generate one teacher answer per training example from the 70B model with greedy decoding and keep it only if it is long enough, contains an "Answer:" line, contains at least one inline citation, and its extracted answer matches one of the gold answers; examples that fail any check are dropped, not resampled.

Judge. The judge sees only the question, the gold answer, and the candidate answer, never the documents, the arrangement, or which model produced the answer, and returns a two-line verdict (`Verdict: CORRECT` or `INCORRECT`, plus a one-line reason), which we parse with a fixed rule. Judge decoding is greedy with a 128-token cap, so its output is deterministic given the (question, gold, candidate) triple, and we cache judge calls by that triple.

Hardware. Every GPU experiment, serving, fine-tuning, and judging, runs on a single node with 4 A100-80GB GPUs.

Release. On acceptance we will release the code, the arrangement generator, the training and evaluation configuration files (every value above is read from these files, not hardcoded), and the analysis scripts, so every number in the paper can be regenerated from the released checkpoints and data.

B DATASET DETAILS AND STATISTICS

Table 10 lists every dataset used in the paper: its source, the split we draw from, the size of that split, and the number of examples we sample and evaluate on. Table 11 lists the data used to fine-tune the LoRAs.

Table 10: Evaluation datasets. "Raw size" is the full size of the named split before any hop-count or answerability filtering. "Sampled size" is what we actually evaluate on in the paper. The ALCE datasets (ASQA, QAMPARI, ELI5) have no annotated gold documents or hop structure, so the hop-count column does not apply; we split their single released evaluation file ourselves into a test part (used for all ALCE numbers in the paper) and a train part (used to build the long-form training data, Table 11).

Dataset	Source	Split	Raw size	Sampled size	Hop counts
MuSiQue	`dgslibisey/MuSiQue` (HF)	validation	2,417	150 (50 each)	2, 3, 4
HotpotQA (distractor)	`hotpotqa/hotpot_qa` (HF)	validation	7,405	150 (all 2-hop)	2 only
2WikiMultiHopQA	`voidful/2WikiMultihopQA` (HF)	validation	12,576	100 (50 each)	2, 4
ASQA	ALCE release, GTR top-100	eval file	948	300 test + 648 train	—
QAMPARI	ALCE release, GTR top-100	eval file	1,000	300 test + 700 train	—
ELI5	ALCE release, BM25 top-100	eval file	1,000	300 test + 700 train	—
Controlled long-form MuSiQue	built from MuSiQue validation	—	—	150 × 4 conditions	2, 3, 4

MuSiQue. Every example has 20 paragraphs in total. We sample 50 answerable questions per hop count (2, 3, and 4 hops, so 150 examples), with a fixed data seed, and take every non-gold paragraph as a distractor, so the model always sees the full 20-paragraph context. The number of gold paragraphs equals the hop count (2 to 4), by the dataset's own annotation.

HotpotQA. We use the distractor setting, which fixes 10 paragraphs per example, 2 gold and 8 distractor. HotpotQA's distractor release contains only 2-hop questions, so all 150 sampled examples are 2-hop; this is why HotpotQA has no chain-order condition in Section 3, since a 2-hop chain has no interior order to permute.

2WikiMultiHopQA. We use all candidate paragraphs listed for each example, so the context size varies by example rather than being fixed like MuSiQue or HotpotQA. We sample 50 questions each at 2 and 4 hops (100 total); the dataset's own question types map cleanly to these two counts (comparison, inference, and compositional questions are 2-hop, and bridge-comparison questions are 4-hop), so there is no 3-hop category to sample. We load this dataset from a community mirror on Hugging Face rather than its canonical release, because the canonical release ships a loading script that recent versions of the `datasets` library refuse to execute; the mirror serves the same underlying questions and annotations.

ASQA, QAMPARI, and ELI5 (ALCE). We use the pre-retrieved passages shipped with the ALCE benchmark rather than retrieving ourselves: ASQA and QAMPARI ship the top 100 passages per question from a GTR dense retriever, and ELI5 ships the top 100 from BM25; we keep the top 10 for every prompt (Appendix A). ALCE ships one evaluation file per dataset, with no separate train/test division, so we make our own: a single seeded shuffle of each file yields a 300-example test set, used for every ALCE number in the paper, and the remainder (648 for ASQA, 700 for QAMPARI, 700 for ELI5) is reserved for building the long-form training data in Table 11. No example appears in both the train and test parts.

Controlled long-form MuSiQue. This is not a separate dataset; it renders the same 150-example MuSiQue validation sample above into long-form prompts, under the four conditions gold-forward-first, gold-forward-middle, gold-forward-last, and scattered, at each of the three context budgets used in Section 4.4 (giving 600 prompts per budget). It keeps MuSiQue's annotated gold, unlike ASQA, QAMPARI, and ELI5, which is what makes it a controlled placement test.

The MuSiQue training split has 19,938 answerable questions with at least two annotated gold paragraphs (14,376 at 2 hops, 4,387 at 3 hops, and 1,175 at 4 hops), so our request for 2,000 questions

Table 11: Data used to fine-tune the LoRAs. "Kept" is the number of training examples in the final SFT file; for the two distilled long-form regimes this is fewer than the number of 70B teacher generations we started from, because we drop any generation that fails the checks in Appendix A (too short, no "Answer:" line, no citation, or, for the two long-form regimes, a wrong final answer). No fine-tuning example overlaps the 150 MuSiQue questions used for evaluation (Appendix D).

Fine-tuning data	Source	Composition	Kept
Short-answer LoRA (scattered / gold-first)	MuSiQue train	2,000 / 2,000 / 1,175 per hop 2/3/4	4,875 train + 300 dev
Long-form ALCE LoRA	ASQA + QAMPARI train-side prompts	648 + 700 distilled prompts	1,343
Controlled long-form MuSiQue LoRA	MuSiQue train	500 per hop 2/3/4, scattered order only	836

per hop count is satisfied exactly at 2 and 3 hops and capped by the data itself at 4 hops, where we take every eligible question. The 1,343-example long-form ALCE set and the 836-example controlled long-form MuSiQue set are both built by rejection sampling, described in Appendix A, from 1,348 and 1,500 candidate teacher generations respectively.

C Base and fine-tuned per-condition results

Table 2 gives the base per-condition means with standard errors. The 70B judge raises absolute accuracy over `em_soft` by about 0.14, roughly uniformly across arrangements, so the paired contrasts in Table 12 are essentially unchanged under either metric. For the fine-tuned LMs, the scattered-trained LoRA is flat across all seven arrangements (Llama between 0.673 and 0.720 `em_soft`; Qwen between 0.573 and 0.633), while the gold-first-trained LoRA is high only at the front arrangements and collapses elsewhere (Llama gold forward last 0.113, Qwen 0.213). This holds at every hop count.

Table 12: Base LMs: key contrasts on MuSiQue (judge, $n = 150$, paired McNemar p). Gathering the gold helps and is significant. The block position has a small effect and the chain order has none. "Best edge" is the higher-scoring of the first and last positions. A star marks $p < 0.05$.

Contrast (judge)	Llama Δ (p)	Qwen Δ (p)
together (front) − scattered	$+0.056$ $(.047)^\star$	$+0.087$ $(.001)^\star$
together (end) − scattered	$+0.116$ $(< .001)^\star$	$+0.073$ $(.007)^\star$
position: best edge − middle	$+0.093$ $(.013)^\star$	$+0.060$ $(.064)$
chain order: forward − reverse (front)	$+0.027$ $(.59)$	-0.027 $(.52)$

D Leakage audit

Fine-tuning uses only MuSiQue training questions, and the evaluation uses the held-out validation questions. There is zero overlap of question ids. MuSiQue draws paragraphs from a shared pool, so some evaluation paragraphs also appear in training, but only as distractors in unrelated questions. No evaluation gold document was a supporting document in training, and no evaluation gold-document to evaluation-answer pair was ever seen together in training. The training-vs-base accuracy gain therefore reflects reading skill, not answer lookup.

E Memorization control

Closed-book accuracy (no documents) is low for all LMs (0.06 to 0.17 `em_soft`; 0.13 for base Llama, 0.08 for base Qwen, 0.24 for base Gemma), so the questions genuinely need the documents. Removing the questions a model answers closed book leaves every arrangement gap within about 0.02 of its full-set value, so the conclusions are not driven by parametric recall.

F ERROR ANATOMY, FULL RESULTS

The gold-only ceiling recovers only about 8 to 10 judge points over the scattered score (Llama 0.800 to 0.880, Qwen 0.660 to 0.760). Under the judge, the residual splits into locate failures, recovered by removing distractors, and a reasoning floor: Llama 17 locate and 13 reason of 30 residual errors, Qwen 21 locate and 30 reason of 51. Of the full-question failures, about 40% are composition failures where every hop is correct in isolation, and the most common error type is a confident wrong entity (about 80%) rather than an abstention or an intermediate answer. Last-token attention onto the gold tokens is as high or higher for failing examples than for passing ones (Llama last layer fail 0.166 vs. pass 0.104; Qwen 0.166 vs. 0.113; token-count baseline about 0.10), which argues against a locate deficit.

G LONG-FORM, FULL RESULTS

Standard ALCE benchmarks. Table 13 reports the full ALCE results: correctness and citation recall on ASQA, QAMPARI, and ELI5, for the base and scattered-trained long-form LoRAs, under both passage orders. Each question comes with ten retrieved passages; *sorted* keeps them in the retriever's score order (most relevant first) and *scattered* shuffles them with a fixed seed. These datasets have no annotated gold, so the only knob is the order of the retrieved set, not the placement of the gold. The scattered-trained long-form LoRA improves correctness and, most clearly, citation recall over the base model, while the order axis is close to null: across all sixteen metric cells the sorted-minus-scattered difference is at most about four points and only one is distinguishable from zero (for example, Llama ASQA STR-EM 42.4 sorted vs. 41.1 scattered; Qwen ELI5 citation recall 53.8 sorted vs. 55.1 scattered). Because ALCE has no gold labels and only about 1.5k tokens of context, it cannot test gold placement and does not isolate the training arrangement.

Table 13: Long-form ALCE results by passage order ($n = 300$ per cell). Correctness is STR-EM (ASQA), recall@5 (QAMPARI), and claim recall (ELI5); "cit. rec." is citation recall judged by the 70B model. The base vs. LoRA columns show that the scattered-trained long-form LoRA improves correctness and citation recall; the sorted vs. scattered rows show that passage order has little effect (every sorted-minus-scattered difference is within about four points, and only one of sixteen is distinguishable from zero). We report both orders rather than averaging so the order effect is visible. These datasets have no annotated gold, so this is not a controlled placement test; see Table 8.

Dataset / metric	order	Llama base	Llama LoRA	Qwen base	Qwen LoRA
ASQA STR-EM	sorted	42.4	49.5	45.6	48.0
	scattered	41.1	49.7	44.0	47.4
ASQA cit. rec.	sorted	62.1	72.5	64.3	72.7
	scattered	66.1	73.9	62.0	71.1
QAMPARI rec@5	sorted	25.1	30.7	24.6	28.9
	scattered	25.5	30.3	23.7	29.3
QAMPARI cit. rec.	sorted	34.0	52.3	54.3	51.0
	scattered	35.6	54.5	52.3	51.8
ELI5 claim rec.	sorted	25.4	27.8	27.4	27.1
	scattered	24.9	28.0	26.8	26.1
ELI5 cit. rec.	sorted	63.8	66.5	53.8	67.2
	scattered	62.1	65.5	55.1	70.6

Controlled long-form MuSiQue, by position. On the controlled long-form MuSiQue, per-hop-fact coverage falls with the position bin as well as with length for the base model at 32k (Llama base early 51.4%, middle 44.8%, late 47.8%), while the scattered-trained LoRA stays flat across bins (early 75.9%, middle 73.3%, late 77.3%). Per-gold-document citation at 32k shows the same contrast (Llama base 13 to 23% by bin, scattered-trained LoRA 63 to 68%).

H LENGTH SWEEP, FULL NUMBERS

Table 14 gives the exact numbers behind Figure 2 on MuSiQue.

Table 14: Length sweep on MuSiQue (judge, $n = 150$; standard error about 0.04 per accuracy cell). "front−scat" is the value of gathering the gold; "edge−mid" is the lost-in-the-middle spread. Base LMs get worse with length; scattered-trained LoRAs stay flat. A star marks $p < 0.05$ on the front−scat contrast (paired McNemar).

	4k		8k		16k		32k	
LM	front−scat	edge−mid	front−scat	edge−mid	front−scat	edge−mid	front−scat	edge−mid
Llama base	+.04	.04	+.06	.05	+.08*	.11	+.11*	.11
Llama scattered	−.01	.01	−.02	−.01	+.03	.01	+.01	.05
Qwen base	+.09*	.08	+.11*	.13	+.11*	.14	+.09*	.11
Qwen scattered	+.03	.03	+.03	.04	+.05	.05	+.04	−.01

I REASONING ABLATION, FULL RESULTS

Table 15 gives the per-arrangement judge accuracy for the base LMs answering directly and for the same LMs with reasoning on (Qwen3-8B native thinking, Llama-3.1-8B prompted chain-of-thought), on the MuSiQue seven-condition study. Decoding is greedy in every case, so the only change from the base runs is the reasoning knob. Reasoning lifts Qwen at every arrangement (pooled 0.569 to 0.733) and helps the scattered case the most, which is why the together-vs-scattered gap shrinks. Prompted chain-of-thought gives Llama almost no lift in pooled accuracy (0.595 to 0.594). Closed-book accuracy with reasoning stays low (0.226 for Llama, 0.215 for Qwen), so the questions still need the documents. Llama prompted chain-of-thought did not terminate within budget on about 18% of cases, worst at four hops (29% versus 12% at two hops); those runs are graded on their truncated output, and 62 whose fallback text overflowed the judge context are dropped, so the Llama reasoning column is a lower bound with $n = 142$ to 147 per arrangement. The truncation rate falls from 29.4% at a 1024-token budget to 21.7% at 4096 and 18.3% at 12288, with diminishing returns, so more budget does not resolve it. The truncated traces enumerate all of the roughly 20 documents and often never emit an "Answer:" line, and they do not loop (only 2 of 275 repeat), which points to genuine non-termination rather than too small a budget. Qwen thinking truncated on under 1% and is graded on all 150 per arrangement.

Table 15: MuSiQue per-arrangement judge accuracy, base (direct answer) versus reasoning on (Qwen3-8B native thinking, Llama-3.1-8B prompted chain-of-thought). Chain order and block position stay flat under reasoning; reasoning lifts Qwen sharply and lifts the scattered case the most, while prompted chain-of-thought barely moves Llama. The Llama reasoning column is a lower bound (see text).

	Llama-3.1-8B		Qwen3-8B	
arrangement	base	+ CoT	base	+ thinking
gold forward first	0.593	0.611	0.593	0.720
gold reverse first	0.567	0.555	0.620	0.747
gold forward last	0.653	0.599	0.580	0.733
gold reverse last	0.653	0.662	0.573	0.760
gold forward middle	0.560	0.578	0.533	0.727
gold reverse middle	0.600	0.570	0.573	0.733
scattered	0.538	0.580	0.507	0.709
pooled (7)	0.595	0.594	0.569	0.733