NoLiMa: Long-Context Evaluation Beyond Literal Matching

Ali Modarressi [1 2 *] **Hanieh Deilamsalehy** [3] **Franck Dernoncourt** [3] **Trung Bui** [3] **Ryan Rossi** [3] **Seunghyun Yoon** [3] **Hinrich Schütze** [1 2]

Abstract

Recent large language models (LLMs) support long contexts ranging from 128K to 1M tokens. A popular method for evaluating these capabilities is the needle-in-a-haystack (NIAH) test, which involves retrieving a "needle" (relevant information) from a "haystack" (long irrelevant context). Extensions of this approach include increasing distractors, fact chaining, and in-context reasoning. However, in these benchmarks, models can exploit existing literal matches between the needle and haystack to simplify the task. To address this, we introduce NoLiMa, a benchmark extending NIAH with a carefully designed needle set, where questions and needles have minimal lexical overlap, requiring models to infer latent associations to locate the needle within the haystack. We evaluate 13 popular LLMs that claim to support contexts of at least 128K tokens. While they perform well in short contexts (<1K), performance degrades significantly as context length increases. At 32K, for instance, 11 models drop below 50% of their strong short-length baselines. Even GPT-4o, one of the top-performing exceptions, experiences a reduction from an almost-perfect baseline of 99.3% to 69.7%. Our analysis suggests these declines stem from the increased difficulty the attention mechanism faces in longer contexts when literal matches are absent, making it harder to retrieve relevant information. Even models enhanced with reasoning capabilities or CoT prompting struggle to maintain performance in long contexts. We publicly release the dataset and evaluation code at https://github.com/adobe-research/NoLiMa.[1]

[*]Work done during an internship at Adobe Research. [1]Center for Information and Language Processing, LMU Munich, Germany [2]Munich Center for Machine Learning (MCML) [3]Adobe Research. Correspondence to: Ali Modarressi <amodaresi@cis.lmu.de>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]Recent models not covered in the main body of this paper (such as GPT-4.1) have improved long-context performance. However, the main finding of this paper also holds for these newer models: performance on NoLiMa starts declining rapidly on contexts that are relatively short compared to their claimed context length. See Appendix E for detailed results.

1. Introduction

In recent years, large language models (LLMs) have made remarkable advancements in handling long-context inputs (Chen et al., 2023; Xiong et al., 2024; Peng et al., 2024). This capability has unlocked new possibilities in various NLP tasks that require understanding or generating content over extended documents. Examples include long- or multi-document question answering (QA), summarization, and many-shot in-context learning (Lee et al., 2024; Chang et al., 2024; Agarwal et al., 2024). To evaluate these models' effectiveness in handling long contexts, several benchmarks have been developed. One prominent benchmark is Needle-in-a-Haystack (NIAH), which tests a model's ability to search for and retrieve a specific fact (the "needle") hidden within irrelevant information (the "haystack") (Kamradt, 2023; Mohtashami & Jaggi, 2023). While the baseline NIAH task assesses surface-level retrieval capabilities, recent adaptations have increased its complexity. These enhancements include introducing multiple needles, incorporating additional distractor material, and interconnecting facts to necessitate in-context reasoning (e.g., fact-chaining) (Hsieh et al., 2024; Levy et al., 2024; Kuratov et al., 2024). Other benchmarks, such as long-, multi-document QA, and long conversation understanding, have also been proposed to evaluate long-context comprehension in a more downstream task manner (Liu et al., 2024; Yen et al., 2024; Zhang et al., 2024; Dong et al., 2024; Wang et al., 2024; Maharana et al., 2024).

Arguably, these tasks share a common foundation: the ability to recall previously seen information (Goldman et al., 2024). This broader category, termed association recall tasks, has been extensively studied in machine learning (Graves et al., 2014; Ba et al., 2016). A key argument is that the attention mechanism, which is the underlying foundation of many LLMs, is inherently adept at identifying and recalling associations present in the input (Olsson et al., 2022; Arora et al., 2024). However, this raises an important question: Long-context benchmarks feature tasks where the queried input (e.g., a question or a task) has literal matches with the provided context. *Do such literal matches make it easier for language models to locate relevant information and output correct answers?*

We argue that many existing long-context benchmarks either explicitly (e.g., synthetic tasks or NIAH-based) or implicitly (e.g., multi-document or long-document QA) contain such literal matches. To address this, we introduce **NoLiMa**, a benchmark designed to minimize literal overlap between questions and their corresponding needles. In NoLiMa, questions and needles contain keywords that are related through associative links, such as real-world knowledge or commonsense facts. By embedding these needles in a haystack, NoLiMa challenges models to leverage latent associative reasoning capabilities rather than relying on surface-level matching.

We evaluate NoLiMa over 13 state-of-the-art language models, all claiming to support token lengths of at least 128K, including GPT-4o, Gemini 1.5 Pro, and Llama 3.3 70B (Hurst et al., 2024; Gemini Team et al., 2024; Meta, 2024). Unlike NIAH-based evaluations, which contain literal matches and exhibit near-saturated performance, NoLiMa presents a more demanding challenge that highlights the limitations of these models. Their performance declines noticeably as context length increases, with considerable drops even at 2K–8K tokens. For instance, at 32K tokens, 11 out of 13 models achieve only half of their short-context performance.

We conduct extensive analyses using NoLiMa, yielding the following insights:

- **Impact of Latent Hops and Fact Direction**: We demonstrate how the number of associative reasoning steps (latent hops) and the ordering of elements within a fact statement influence task performance. (§ 4.4.1)
- **Context Length vs. Needle Position**: Our aligned-depth analysis shows that as latent reasoning complexity grows, performance depends more on context length than needle position. Without surface cues, longer contexts overwhelm the attention mechanism. (§ 4.4.2)
- **Chain-of-Thought (CoT) Prompting and Reasoning-based Models**: While CoT prompting or reasoning-based models such as GPT-o1 (OpenAI et al., 2024) improve performance by encouraging step-by-step reasoning, they fail to fully mitigate the challenge, particularly in contexts exceeding 16K tokens. (§ 4.4.3)
- **Ablation Tests**: We confirm that the presence of literal matches significantly simplifies the task, enabling models to achieve high accuracy in answering questions. In contrast, when literal matches serve as distractors, they severely impair accuracy. (§ 4.4.4)

	R-1	**R-2**	**R-L**
Long-document QA			
∞Bench QA (Zhang et al., 2024)	0.966	0.545	0.960
∞Bench MC (Zhang et al., 2024)	0.946	0.506	0.932
RAG-style (Multi-doc) QA			
RULER QA (Hsieh et al., 2024)	0.809	0.437	0.693
HELMET (RAG) (Yen et al., 2024)	0.689	0.304	0.555
Recall-based			
Vanilla NIAH (Kamradt, 2023)	0.905	0.789	0.855
RULER S-NIAH (Hsieh et al., 2024)	0.571	0.461	0.500
BABILong (0K) (Kuratov et al., 2024)	0.553	0.238	0.522
NoLiMa	**0.069**	**0.002**	**0.067**

Table 1. ROUGE precision scores between the input document and the question: higher ROUGE scores indicate greater literal matches between the question and the relevant context.

Through NoLiMa, we reveal the limitation of literal matching in long-context benchmarks and introduce a novel approach for evaluating models' latent reasoning in longer contexts.

2. Related Work

With the increasing popularity of long-context language modeling, numerous benchmarks have been introduced to evaluate this capability. Needle-in-a-Haystack (NIAH) is the most well-known and widely used benchmark (Mohtashami & Jaggi, 2023; Kamradt, 2023). However, due to performance saturation, various extensions have been proposed. These include increasing complexity by adding more needles, chaining needles to require inter-needle reasoning (fact-chaining), or incorporating arithmetic or code reasoning (Kamradt, 2023; Hsieh et al., 2024; Levy et al., 2024; Kuratov et al., 2024; Hengle et al., 2024; Zhang et al., 2024; Vodrahalli et al., 2024). Some tasks increase the complexity to such an extent that they become overly difficult even in short-context scenarios. For instance, BABILong includes tasks that perform poorly (e.g., the counting task achieves 28% accuracy) even without any irrelevant background text (0K) (Kuratov et al., 2024). Similarly, the Ancestral Tree Challenge (ATC) employs extensive fact-chaining, resulting in tasks that are overly complex even for short contexts (<1K) (Li et al., 2024). While such tasks challenge language models in long contexts, they raise the question of whether the tasks are inherently too complex for models to handle, regardless of context length.

Literal Matching in Long-Context Benchmarks. Another frequent pattern in many long-context benchmarks is the presence of literal matches between the facts required to answer a question and the question itself. This fact is not limited to synthetic recall-based tasks (e.g., vanilla NIAH, RULER retrieval-based sets) but also affects downstream-

Question	Needle	Keyword Type
Which character has been to W_q?	(Def.) Actually, [CHAR] lives next to the W_n. (Inv.) W_n is next to where [CHAR] lives.	W_n Buildings&Landmarks (e.g., Semper Opera) W_q Countries, cities, states (e.g., Dresden)

Table 2. An example template of the proposed needle set in NoLiMa. (All templates are available in Appendix A.) The placeholders [CHAR], W_q, and W_n represent the randomly selected character (also the answer), the query keyword, and the needle keyword, respectively. Def.: default order. Inv.: inverted order.

like QA-based benchmarks (Hsieh et al., 2024; Liu et al., 2024; Zhang et al., 2024; Bai et al., 2024; Yen et al., 2024), which often implicitly include literal matches between the relevant document and the question. Although many of these studies introduce complexity by adding similar documents as distractors, literal matches can still provide cues. These cues may help models focus on potential relevant facts based on matches, as attention mechanisms excel at recalling repetitive patterns (Olsson et al., 2022; Arora et al., 2024). We later demonstrate to what extent literal matches simplify recall-based questions (cf. 4.4.4). To quantify the prevalence of these matches in popular benchmarks, we compute ROUGE (R-1, R-2, and R-L) precision scores[2] (Lin, 2004) between the question and the context – the needle (in recall-based tasks), the relevant document (in multi-document setups), or the full document (in long-document QA). This analysis measures the degree of literal overlap between the question and the context. Table 1 demonstrates that NoLiMa has much less literal overlap than other datasets.

3. NoLiMa

The goal of NoLiMa is to design a task that is inherently simple to solve through associative reasoning, but for which surface-level matching has zero utility. As a result, NoLiMa allows us to cleanly investigate associative reasoning in long-context scenarios without confounding from surface-level effects.

The main elements of NoLiMa are similar to vanilla NIAH. A "needle" – a single key piece of information – is placed within a "haystack", i.e., a long irrelevant text (in our case, snippets from books). Given a question, the model is then tested on its ability to find the needle. The needle is designed to be a clearly relevant answer to the question. In contrast to existing NIAH tasks, we impose the condition that the question have minimal literal match with the needle. To achieve this, we design a set of needles and corresponding questions, collectively referred to as a "needle set." Table 2 presents one of the constructed needle set templates (see Appendix A for the full list). Each needle consists of a unique character and specific information about them. Example:

Actually, Yuki lives next to the **Semper Opera House**.

The needle contains a keyword (W_n, here "Semper Opera House") that serves as the critical link between needle and question. The question is designed to retrieve this information by asking which character possesses a specific attribute W_q, "Dresden" in the example:

Which character has been to **Dresden**?

The Semper Opera House is located in Dresden. Thus, the model should be able to identify the latent association link between W_q ("Dresden") in the question and W_n ("Semper Opera House") in the needle. Since there is no literal overlap between needle and question, the model must rely on this latent association link to retrieve "Yuki", the correct answer. For some of our needles, the association involves commonsense reasoning instead of world knowledge. Example: "Then Yuki mentioned that he has been vegan for years." → "Which character cannot eat fish-based meals?" To push the limits of the model's ability to identify hidden associations, we include questions that require two hops to connect W_q with W_n, for example:

Which character has been to **the state of Saxony**?

Here, the model should tap into its knowledge that Dresden (and hence the Semper Opera) is located in the state of Saxony. This two-hop setup further increases the difficulty of identifying the latent association of W_q with W_n.

To make NoLiMa an effective benchmark for evaluating LLM long-context abilities, we impose several constraints on the needle set. (i) We select keywords that ensure simplicity – so that, without irrelevant context, the associations are clear and the model can identify the correct answer. (ii) We randomize the assignment of character names from a diverse pool to minimize sensitivity to tokenization problems and mitigate ethnic bias (Navigli et al., 2023; Jiang et al., 2024). Names already occurring in the haystacks are excluded. (iii) We ensure W_n is uniquely associated with W_q, avoid language-based cues, and in most cases employ preface phrases–short lead-ins or contextual buildup (e.g., "Actually," "In 2013, after waiting in line...")–to isolate needles from preceding context. See Appendix A for details.

[2] We use precision as our metric to measure how many of the question's tokens occur in the relevant context, rather than the reverse.

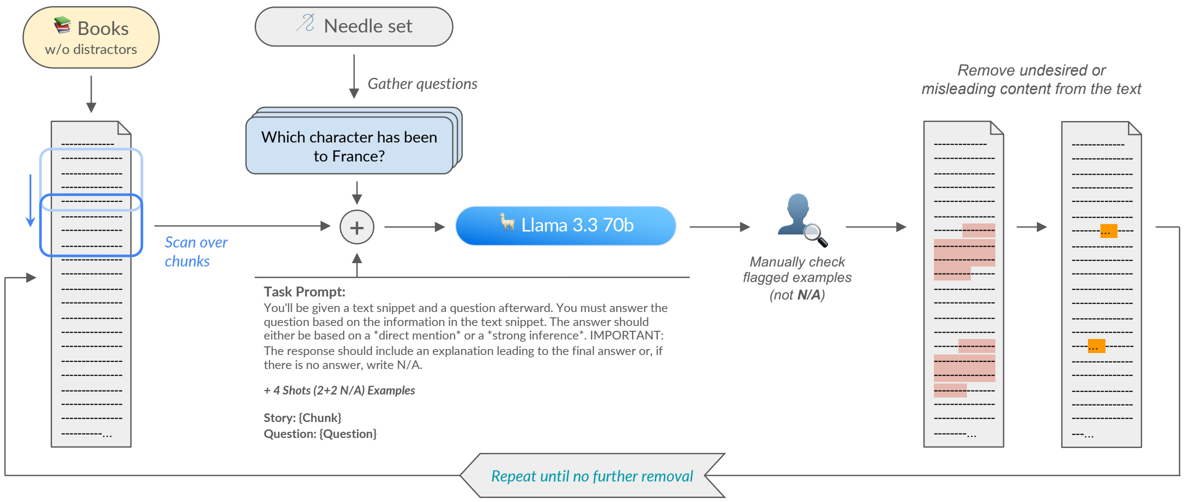


Figure 1. Haystack filtering pipeline for undesired or misleading content

3.1. Haystack Filtering Pipeline

We devise a filtering process to ensure that the haystack does not contain: (1) Any distracting words that have extreme literal or high semantic similarity with the key points mentioned in the question, (2) Any information that explicitly or through inference is a potential false answer to the question.

Distractor Filtering. For this step, we use an embedding function, Contriever (Izacard et al., 2022), to find similar words in the haystack to the keywords of the questions. First we gather all words in the haystack and compute their respective embedding. Then using dot-product similarity we compute their similarity to the question keywords. We manually inspect the top-20 similar words per each W_q and flag those with high semantic or substring similarity for removal. In the removal process those sentences that contain flagged words are removed from the haystack. This initial filtering step helps to avoid an uncontrolled set of superficial distractors that could undesirably disrupt the experimental results. We will discuss the impact of distractors on the model performance in our analysis (Section 4.4.4).

Filtering Undesired Answer Candidates. In this step, we implement a semi-automatic redaction process to detect and remove text spans that could be interpreted as plausible but unintended answers. As shown in Figure 1, this process takes the haystack text—already filtered for distractors—along with questions from our needle set as input. Assuming the model should infer cases within short contexts, we scan the input texts in smaller chunks.[3] To identify potential answers within a chunk, we pair each question with the chunk and input them into an instruction-tuned language model, along with a short instruction and few-shot examples. The model responds with either "N/A" (indicating no relevant information was found) or an explanation identifying a possible candidate answer. Flagged examples are manually reviewed[4] to determine whether the identified content should be removed. If no relevant content is identified, the text remains unchanged. This process is repeated across all selected haystacks until no further removals are necessary.

4. Experiments

4.1. Dataset Configuration

In NoLiMa, we use 5 groups of needles, each with two "word order" variations: *default* and *inverted*. In the default order, the answer character (CHAR) precedes the needle keyword (W_n), following the pattern "…[CHAR] … W_n" (see Table 2, column "Needle"). In the inverted order, the character name follows W_n, yielding the pattern "W_n … [CHAR] …". Each group includes 2–6 keyword sets, with some sets containing multiple W_q items to produce both one-hop and two-hop examples. This setup results in 58 question-needle pairs in total. To generate the haystacks, we select 10 open-license books, ensuring each covers at least 50K tokens. Using the filtering mechanism described in Section 3.1, we process the text to prepare it for haystack construction. To mitigate potential memorization issues—since these books are publicly available—we construct haystacks by concatenating short snippets. Specifically, we iteratively and randomly select a book, extract a continuous snippet

[3] With an 800-character stride and a 1000-character chunk size (∼250 tokens).

[4] All manual reviews—in both filtering steps—were conducted by one of the authors.

Models	Claimed Length	Effective Length	Base Score (×0.85: **Thr.**)	1K	2K	4K	8K	16K	32K
GPT-4o	128K	8K	99.3 (84.4)	98.1	98.0	95.7	89.2	81.6	69.7
Llama 3.3 70B	128K	2K	97.3 (82.7)	94.2	87.4	81.5	72.1	59.5	42.7
Llama 3.1 405B	128K	2K	94.7 (80.5)	89.0	85.0	74.5	60.1	48.4	38.0
Llama 3.1 70B	128K	2K	94.5 (80.3)	91.0	81.8	71.2	62.7	51.8	43.2
Gemini 1.5 Pro	2M	2K	92.6 (78.7)	86.4	82.7	75.4	63.9	55.5	48.2
Jamba 1.5 Mini	256K	<1K	92.4 (78.6)	76.3	74.1	70.8	62.2	52.7	43.6
Command R+	128K	<1K	90.9 (77.3)	77.0	73.5	66.2	39.5	21.3	7.4
Gemini 2.0 Flash	1M	4K	89.4 (76.0)	87.7	87.5	77.9	64.7	48.2	41.0
Mistral Large 2	128K	2K	87.9 (74.7)	86.1	85.5	73.3	51.4	32.6	18.8
Claude 3.5 Sonnet	200K	4K	87.5 (74.4)	85.4	84.0	77.6	61.7	45.7	29.8
Gemini 1.5 Flash	1M	<1K	84.7 (72.0)	68.6	61.6	51.0	44.4	35.5	28.6
GPT-4o mini	128K	<1K	84.8 (72.1)	67.7	58.2	44.2	32.6	20.6	13.7
Llama 3.1 8B	128K	1K	76.7 (65.2)	65.7	54.4	44.1	31.9	22.6	14.2

Table 3. NoLiMa benchmark results on the selected models. Following Hsieh et al. (2024), we report the effective length alongside the claimed supported context length for each model. However, we define the effective length as the maximum length at which the score remains above a threshold, set at 85% of the model's base score (shown in parentheses). Scores exceeding this threshold are underlined. Scores that are below 50% of the base score are shaded in red.

(under 250 tokens), and append it to the haystack until it exceeds 2K lines, resulting in haystacks exceeding 60K tokens. In all experiments, each needle is placed 26 times at equal intervals across the evaluated context length. With 5 randomly generated haystacks, 58 question-needle pairs, and 26 placements per context length, this setup results in 7,540 tests per context length experiment.

4.2. Models

For the filtering process, we opted for using the Llama 3.3 70b instruction tuned model (Meta, 2024). As a control test, for each question, we place its needle in 100 randomly selected chunks to determine whether the model (1) understands the filtering task and (2) is familiar with the facts and capable of inferring the answer. The model achieves a score of 99.8% in this test, indicating its ability to effectively flag conflicting information from the haystacks.

For the evaluation process, we select five closed-source models: GPT-4o, GPT-4o Mini (Hurst et al., 2024), Gemini 1.5 Pro Flash, Gemini 2.0 Flash (Gemini Team et al., 2023; 2024) and Claude 3.5 Sonnet (Anthropic, 2024), along with seven open-weight Llama models: The Llama 3.x model family (3.1 8B, 70B, 405B, and 3.3 70B) (Dubey et al., 2024; Meta, 2024), Mistral Large (Mistral, 2024), Command R+ (Cohere For AI, 2024), and Jamba 1.5 Mini (Team et al., 2024). All these models are well-known and widely used in long-context setups. In our analysis on reasoning-based prompting and models, we evaluate GPT-o1, GPT-o3 Mini (OpenAI et al., 2024; OpenAI, 2025), and DeepSeek-R1 Distill-Llama-70B (DeepSeek-AI et al., 2025). More details regarding model versions and deployment details are described in Appendix B.

4.3. Evaluation Setup & Metric

During inference, we use a task template (see Appendix C) that instructs the model to answer the question based on the provided text. Since all questions seek the name of the character mentioned in the needle, any returned answer containing the correct name is considered accurate. Accuracy is reported as the proportion of tests with correct answers.

Models are evaluated on all tasks over context lengths of 250, 500, 1K, 2K, 4K, 8K, 16K, and 32K. To take into account how models would perform on NoLiMa regardless of long-context scenario, we control the difficulty of the task by reporting a **base score**. Evaluations at context lengths of 250, 500, and 1K are used to compute the base score. These three are the shortest contexts. If a model can solve the task at these lengths, then any deterioration of its performance at greater lengths is expected to be solely due to its difficulties with generalizing over long contexts. For each question-needle example, we compute the average scores over 5 haystacks, then take its maximum score across the 250, 500, and 1K tests. The final base score is obtained by averaging these maximum scores across all question-needle examples. Inspired by Hsieh et al. (2024), we also report the **effective length** of each model. While they use the performance of Llama 2 at 4K context length as a threshold (85.6%), we define the threshold as 85% of the base score. Thus, the effective length of a model is the largest tested length that exceeds this threshold. Additionally, some plots show the **normalized score**, calculated by dividing the accuracy score by the base score.

4.4. Results

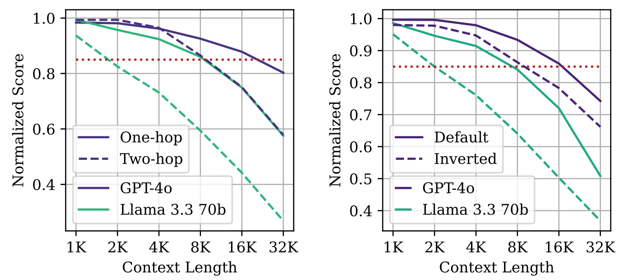
Table 3 presents the performance results of all NoLiMa tests on the selected models. Most models achieve high base scores, indicating that the designed needle set is relatively simple to answer in shorter contexts. Even models with base scores exceeding 90.0% exhibit a significantly shorter effective length than their claimed lengths, generally limited to ≤2K tokens, with GPT-4o being an exception. While GPT-4o demonstrates strong overall performance, it fails to generalize effectively beyond 8K tokens.[1] Out of the 13 models, 11 exhibit performance at 32K lengths that is half or less of their base scores. For comparison, in other benchmarks with similar settings, such as BABILong (QA1) (Kuratov et al., 2024) and RULER (Hsieh et al., 2024), Llama 3.1 70B achieves effective lengths of 16K[5] and 32K, respectively. However, in NoLiMa, Llama 3.1 70B has an effective length of only 2K and shows a significant drop in performance at 32K lengths (42.7% vs. 94.3% base score). Models such as Claude 3.5 Sonnet, Gemini 1.5 Flash, GPT-4o mini, and Llama 3.1 8B may have weaker base scores, but their effective lengths are calculated relative to these scores. This reveals an interesting observation: Models like Claude 3.5 Sonnet and Gemini 2.0 Flash, despite having a lower base score, may underperform in shorter contexts but demonstrate better length generalization than models with higher base scores, such as Llama 3.1 70B and Llama 3.3 70B. In fact, both Sonnet and Gemini 2.0 Flash achieve even higher raw scores in 4K-token experiments compared to some higher-base-score models.

Model scaling generally improves performance, as seen in the progression from Llama 3.1 8B to 70B, Gemini 1.5 Flash to Pro, or GPT-4o mini to GPT-4o. However, the benefits of scaling diminish at larger scales; for example, the performance gap between Llama 3.1 70B and 405B is smaller (and sometimes worse) than that between 8B and 70B. In general, "lite" models such as Gemini 1.5 Flash, GPT-4o mini, and Llama 3.1 8B perform well in shorter contexts (<1K tokens) but fail to generalize effectively in longer contexts.

Based on the dataset construction method described in Section 4.1, NoLiMa can generate haystacks at any desired length. We applied this to test GPT-4o and Gemini 2.0 Flash at 64K and 128K tokens. GPT-4o maintained over 50% of its base score at 128K, while Gemini 2.0 Flash dropped to just 16.4%. Full results and evaluation details are provided in Appendix E.

4.4.1. Latent hops & Inversion

As discussed in Section 3, our needle set also includes examples requiring two-hop associative linking from the question keyword to the needle keyword. To evaluate the impact on length generalization, Figure 2(a) presents the normalized performance of two top-performing models on one-hop



(a) One-hop vs. Two-hop (b) Default vs. Inverted

Figure 2. Impact of (a) number of hops and (b) inversion of order ("[CHAR] … W_n" vs. "W_n … CHAR") on normalized performance across GPT-4o and Llama 3.3 70B models. The red dotted line indicates the 0.85 effective threshold.

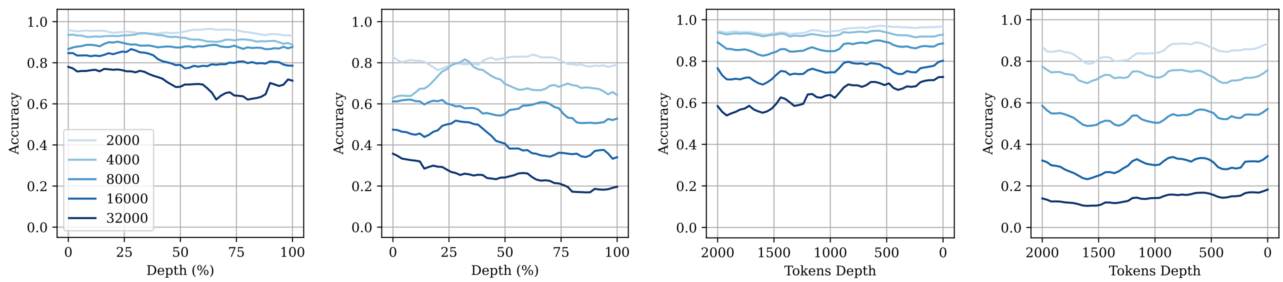
and two-hop tasks. It is evident that, for the same context lengths, questions involving two-hop latent reasoning steps are more challenging than those requiring one-hop reasoning. Notably, the performance gap between one-hop and two-hop tasks widens with increasing context lengths. GPT-4o demonstrates impressive generalization performance, handling both types of examples effectively even at context lengths up to 4K. A detailed breakdown of performance on one-hop and two-hop examples is provided in Appendix F, complementing the aggregate results shown in Table 3.

Each group of needles includes both a default and an inverted template and Figure 2(b) shows that inverted examples are more challenging to answer. We argue this difficulty arises from the model's causal attention mechanism, particularly in longer contexts where attention signals weaken. In the default template, the question – in particular W_q – can link directly to W_n, which generally will contain information about the character's name since the name appears earlier in the sequence. This allows the model to backtrace effectively from W_q through W_n to the character. In the inverted template, W_q may still attend to W_n, but since the fact is incomplete (the character hasn't been mentioned yet), the model cannot use that attention to resolve the question. Instead, it must rely on weaker signals encoded in the character's name to establish the link, which becomes harder with longer contexts due to diminishing attention strength. While these findings shed light on the challenge, deeper mechanistic analysis is beyond the scope of this paper and requires further study.

4.4.2. Needle Placement Depth Analysis

A common evaluation across NIAH-based benchmarks (Kamradt, 2023) examines the impact of needle placement within the context window. In Figure 3(a), we observe a "lost-in-the-middle" effect (Liu et al., 2024) in 32K, where model performance dips when the needle appears in the

[5] In BABILong, the effective length is also based on 85% of the 0K base performance threshold



(a) Full Sweep (One-hop) (b) Full Sweep (Two-hop) (c) Last 2K (One-hop) (d) Last 2K (Two-hop)

Figure 3. The full sweep plots (a & b) illustrate performance across the entire context window, where 0% corresponds to the beginning of the haystack and 100% to the end. The plots for the last 2K tokens (c & d) depict performance when needle placements are aligned within that range for various context lengths; 0 marks the end of the context, and larger values indicate positions farther from the end (up to 2K tokens inward). The color shading of each plot line represents the tested context length. To minimize noise and highlight trends more clearly, we increased the number of placements from 26 to 51 and applied a moving average with a window size of 12.[6]

middle of longer contexts.

Additionally, Figure 3(b) reveals a key phenomenon: longer contexts in more complex (two-hop) examples dampen the performance distribution over the full sweep depending on their length. In vanilla multi-document or NIAH-based benchmarks (Kamradt, 2023; Liu et al., 2024), models perform consistently well when the needle (or gold document) appears at the very beginning or end of the context window, with minimal impact from context length. However, in NoLiMa, as task complexity increases in two-hop scenarios, larger context sizes shift the entire trendline downward toward zero, with performance declining even at the edges of the context window.

To further investigate this issue, we devise an alternative setup that focuses on analyzing the last 2K tokens instead of sweeping across the full context. Therefore, we align the placement positions in the last 2K tokens for all context lengths (see Figure 4). This ensures that for a certain token depth the only changing factor in each plotline is the context length, which in turn means that the model has more tokens that it needs to attend to.

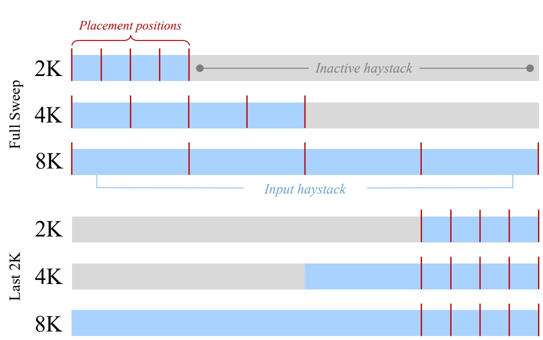


Figure 4. Needle placements in full sweep (top) vs. last 2K tokens sweep (bottom): In the last 2K setup, placement positions are aligned in different context lengths, unlike the proportion-based positioning in full sweep.

Based on the final 2K results in Figure 3(c), the one-hop setup confirms our earlier observations from the full-sweep plots. The “lost-in-the-middle” phenomenon—where performance dips toward the center of the context—primarily appears in simpler tasks. Each plotline drops as it moves toward the center, reflecting its dependence on placement position and the way the model encodes positional information. In contrast, the two-hop scenario appears to be influenced more by attention limitations than by position encoding alone. Figure 3(d) reveals that, rather than depth exacerbating performance drops, the plot lines remain relatively stable over the last 2K positions. However, context length significantly reduces the overall performance trends observed in this range. Llama 3.x models, like many other recent language models, feature rotary position embeddings (RoPE), a relative PE (Su et al., 2024). For each token depth in Figure 3(d), as the relative distance between question and fact remains the same regardless of context length, position encoding does not explain the performance drop. Instead, the main limiting factor is the increased context length: as the number of tokens grows, the attention mechanism struggles to process information effectively. In the absence of strong surface-level cues (e.g., literal matches), locating relevant facts becomes challenging for the model, regardless of their position within long contexts.

4.4.3. CoT Prompting

Since NoLiMa examples require an associative reasoning between the needle and question keywords to retrieve the

[6] All figures use Llama 3.3 70B. Plots without smoothing applied are available in Appendix G.

	4K	8K	16K	32K
		One-hop		
- w/o CoT	90.3	84.1	73.2	56.2
- w/ CoT	95.6	91.1	82.6	60.6
Increase rate	5.9%	8.3%	12.8%	7.8%
		Two-hop		
- w/o CoT	70.7	57.4	42.7	25.9
- w/ CoT	82.4	70.1	56.7	34.3
Increase rate	16.5%	22.1%	32.7%	32.4%

Table 4. Comparison of Chain-of-Thought (CoT) improvements in performance for Llama 3.3 70B, evaluated on both one-hop and two-hop tests.

	Base Score	4K	8K	16K	32K
		Llama 3.3 70b			
- w/o CoT	98.3	55.5	37.2	16.7	8.9
- w/ CoT	97.1	73.0	51.2	31.8	10.1
		Reasoning models			
GPT-o1	99.9	92.0	78.0	60.1	31.1
GPT-o3 Mini	98.8	52.8	36.9	25.5	18.9
DeepSeek R1-DL-70b	99.9	91.4	75.5	49.4	20.7

Table 5. Evaluation results of NoLiMa-Hard: Scores falling below 50% of the base score are highlighted in red.

correct answer, in this part we evaluate when the model is prompted to reason in a Chain-of-Thought (CoT) style (Wei et al., 2022) before returning a final answer (see Appendix C for more details). In Table 4, we present the results when asked for CoT compared to asking directly for the final answer. CoT prompting shows improvements over long-context tests and it shows a higher rate of improvement in two-shot. Despite the improvements, the tasks seem to remain challenging. For example, two-hop examples with CoT prompting barely achieve the scores of one-hop examples without CoT and continue to perform poorly on texts 16K tokens or longer. The challenge with CoT prompting is that the questions in NoLiMa are straightforward. They are mentioning a singular clue to the answer, meaning they cannot be further decomposed into simpler steps. This limits the benefits of CoT prompting. However, the difficulty lies in reasoning through the association between the question and the needle, which remains a significant challenge for the model.

To assess the performance of reasoning-based models (e.g., GPT-o1) on NoLiMa, we selected the 10 most challenging needle-question pairs from the 58 available, based on the results summarized in Table 3. We refer to this subset as NoLiMa-Hard and present the evaluation results in Table 5. While reasoning-based models outperform CoT prompting on Llama 3.3, they still fail to achieve full-length generalization on this subset. Across all models, performance drops below the 50% mark at 32K context length. Notably, base scores are nearly perfect, demonstrating the simplicity of the task—even within this designated "hard" subset. This means that even with intermediate reasoning steps, models still struggle to link the needle to the question in long contexts without surface-level cues.

4.4.4. Ablation Study: Literal match effect

To examine the simplifying impact of literal matches on results, we define two new sets of tests: (1) **Direct**: questions that explicitly ask about the fact stated in the needle by stating W_n in the question, resembling a vanilla NIAH evaluation (Kamradt, 2023). (2) **Multiple Choice (MC)**: questions that maintain the required latent associative reasoning while incorporating literal matches. In this setup, the question includes four character names as answer options—three from the haystack and one correct answer from the needle.

As expected, Table 6 shows that direct examples with a high degree of literal overlap between the question and the needle are straightforward for the model to answer, even in long contexts, consistent with prior findings in RULER (Hsieh et al., 2024). Additionally, literal matches significantly aid the model when the questions remain unchanged, and only the multiple-choice format is introduced. The inclusion of literal matches in the multiple-choice setup provides significant guidance to the model. By offering the character names as answer options, including the correct name from the needle, the model can focus its search within a smaller scope. This dramatically simplifies the task of identifying the correct answer, as the literal match serves as a direct hint, reducing ambiguity in the reasoning process.

Distracting Literal Matches. While literal matches serve as cues if they are part of the relevant fact, they can also act as distractors if they are irrelevant to the answer. In Section 2, we noted that some related benchmarks include similar documents in the context as distractors to test the model's ability to discern the correct answer from irrelevant ones. This setup creates matches between the query and both relevant and irrelevant documents or facts. In contrast,

	8K	16K	32K
Direct	98.3	98.5	98.5
One-hop	84.1	73.2	56.2
- w/ Literal Match (MC)	98.7	97.4	93.1
Two-hop	57.4	42.7	25.9
- w/ Literal Match (MC)	96.3	94.6	87.2

Table 6. Results in two literal match setups: direct and multiple choice (MC) questions. Model: Llama 3.3 70B

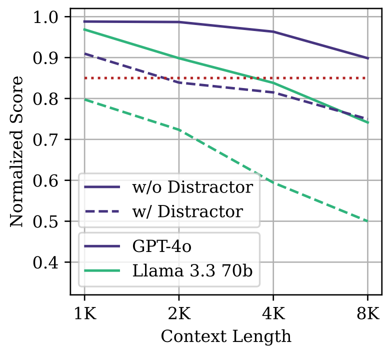


Figure 5. Normalized performance comparison across GPT-4o and Llama 3.3 70B models, with and without distractors. The red dotted line marks the 0.85 effective threshold.

NoLiMa allows us to explore a different scenario: when the context contains distracting words overlapping with the question, while the relevant fact has minimal overlap with the query. We insert a distractor sentence into the haystack (details in Appendix D) containing W_q but entirely irrelevant to both the needle and the question's intent. This setup poses a significant challenge, requiring the model to disregard irrelevant literal overlaps while identifying a relevant fact with no meaningful overlap with the query. As shown in Figure 5, such distractors have a substantial impact on degrading length generalization. GPT-4o now demonstrates an effective length of just 1K, while Llama 3.3 70B performs even worse. While adding distractors slightly lowers base scores (GPT-4o: 93.8, Llama 3.3 70B: 84.4), the normalized plots still clearly illustrate a performance drop at longer lengths. These results highlight the challenge of resolving queries in contexts where irrelevant overlaps mislead the model, and the relevant fact shares no overlap with the question.

5. Conclusion

NoLiMa provides a challenging benchmark for evaluating the reasoning capabilities of large language models in long-context settings. By removing literal overlaps between questions and relevant information, the benchmark tests models' ability to infer and link information within extensive irrelevant content. Our findings show that even state-of-the-art models struggle, especially as context length increases, revealing serious limits in their attention mechanism. While causal attention should theoretically access all previous tokens, models often rely on surface-level cues in longer contexts. This vulnerability becomes more pronounced when the context contains literal matches that fail to connect with the truly relevant fact, causing models to overlook the correct information and focus instead on superficial signals. We believe our findings with NoLiMa are likely to extend to downstream applications. For instance, in search engines or RAG systems, a relevant document containing the correct answer may have a lexical gap with the query. So, even if such a document is retrieved alongside others that likely have higher lexical similarity, language models may struggle to extract the correct answer, as they can become distracted by the lexical overlap with these other documents. This work highlights the need for benchmarks that go beyond surface-level retrieval to assess deeper reasoning. NoLiMa sets a new standard for evaluating long-context comprehension and emphasizes the importance of developing approaches capable of handling complex reasoning in long contexts.

Impact Statement

This paper presents work aimed at advancing the field of long-context language modeling by evaluating and analyzing the most commonly used LLMs. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

We thank Abdullatif Köksal, Leonie Weissweiler, and Amir Hossein Kargaran for their valuable feedback and support, particularly in the early stages of this project. We also appreciate the insights from our peers, and we are grateful to the anonymous reviewers for their constructive comments.

References

Agarwal, R., Singh, A., Zhang, L. M., Bohnet, B., Rosias, L., Chan, S. C., Zhang, B., Faust, A., and Larochelle, H. Many-shot in-context learning. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://openreview.net/forum?id=goi7DFHlqS.

Anthropic, A. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3, 2024.

Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., and Re, C. Zoology: Measuring and improving recall in efficient language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LY3ukUANko.

Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z., and Ionescu, C. Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29, 2016.

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., Dong, Y., Tang, J., and Li, J. LongBench: A bilingual, multitask benchmark for long context understanding. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Vol-*

ume 1: Long Papers), pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Chang, Y., Lo, K., Goyal, T., and Iyyer, M. Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Ttk3RzDeu.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Cohere For AI. c4ai-command-r-plus-08-2024, 2024. URL https://huggingface.co/CohereForAI/c4ai-command-r-plus-08-2024.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dong, Z., Tang, T., Li, J., Zhao, W. X., and Wen, J.-R. BAMBOO: A comprehensive benchmark for evaluating long text modeling capacities of large language models. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2086–2099, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.188/.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gemini Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Goldman, O., Jacovi, A., Slobodkin, A., Maimon, A., Dagan, I., and Tsarfaty, R. Is it really long context if all you need is retrieval? towards genuinely difficult long context NLP. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16576–16586, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.924. URL https://aclanthology.org/2024.emnlp-main.924/.

Graves, A., Wayne, G., and Danihelka, I. Neural turing machines, 2014. URL https://arxiv.org/abs/1410.5401.

Hengle, A., Bajpai, P., Dan, S., and Chakraborty, T. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models. *arXiv preprint arXiv:2408.10151*, 2024.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jKN1pXi7b0.

Jiang, B., Xie, Y., Hao, Z., Wang, X., Mallick, T., Su, W. J., Taylor, C. J., and Roth, D. A peek into token bias: Large language models are not yet genuine reasoners. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4722–4756, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.272. URL https://aclanthology.org/2024.emnlp-main.272/.

Kamradt, G. Needle in a haystack-pressure testing llms. *Github Repository*, pp. 28, 2023.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=e2TBb5y0yFf.

Kuratov, Y., Bulatov, A., Anokhin, P., Rodkin, I., Sorokin, D. I., Sorokin, A., and Burtsev, M. BABILong: Testing the limits of LLMs with long context reasoning-in-a-haystack. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=u7m2CG84BQ.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lee, J., Chen, A., Dai, Z., Dua, D., Sachan, D. S., Boratko, M., Luan, Y., Arnold, S. M. R., Perot, V., Dalmia, S., Hu, H., Lin, X., Pasupat, P., Amini, A., Cole, J. R., Riedel, S., Naim, I., Chang, M.-W., and Guu, K. Can long-context language models subsume retrieval, rag, sql, and more?, 2024. URL https://arxiv.org/abs/2406.13121.

Levy, M., Jacoby, A., and Goldberg, Y. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15339–15353, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.818. URL https://aclanthology.org/2024.acl-long.818/.

Li, M., Zhang, S., Liu, Y., and Chen, K. Needlebench: Can llms do retrieval and reasoning in 1 million context window?, 2024. URL https://arxiv.org/abs/2407.11963.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., and Fang, Y. Evaluating very long-term conversational memory of LLM agents. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Meta, A. Llama 3.3 model card. 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md.

Mistral, A. Mistral large 2. *Mistral Large 2 Blogpost*, 2024. URL https://mistral.ai/news/mistral-large-2407/.

Mohtashami, A. and Jaggi, M. Random-access infinite context length for transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=7eHn64wOVy.

Navigli, R., Conia, S., and Ross, B. Biases in large language models: Origins, inventory, and discussion. *J. Data and Information Quality*, 15(2), June 2023. ISSN 1936-1955. doi: 10.1145/3597307. URL https://doi.org/10.1145/3597307.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

OpenAI. Openai o3-mini system card. 2025. URL https://openai.com/index/o3-mini-system-card/.

OpenAI, : Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Peng, B., Quesnelle, J., Fan, H., and Shippole, E. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wHBfxhZu1u.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Team, G. Gemma 3. 2025. URL https://goo.gle/Gemma3Report.

Team, J., Lenz, B., Arazi, A., Bergman, A., Manevich, A., Peleg, B., Aviram, B., Almagor, C., Fridman, C., Padnos, D., et al. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*, 2024.

Vodrahalli, K., Ontanon, S., Tripuraneni, N., Xu, K., Jain, S., Shivanna, R., Hui, J., Dikkala, N., Kazemi, M., Fatemi, B., Anil, R., Dyer, E., Shakeri, S., Vij, R., Mehta, H., Ramasesh, V., Le, Q., Chi, E., Lu, Y., Firat, O., Lazaridou, A., Lespiau, J.-B., Attaluri, N., and Olszewska, K. Michelangelo: Long context evaluations beyond haystacks via latent structure queries, 2024. URL https://arxiv.org/abs/2409.12640.

Wang, M., Chen, L., Cheng, F., Liao, S., Zhang, X., Wu, B., Yu, H., Xu, N., Zhang, L., Luo, R., Li, Y., Yang, M., Huang, F., and Li, Y. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5627–5646, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.322. URL https://aclanthology.org/2024.emnlp-main.322/.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Xiong, W., Liu, J., Molybog, I., Zhang, H., Bhargava, P., Hou, R., Martin, L., Rungta, R., Sankararaman, K. A., Oguz, B., Khabsa, M., Fang, H., Mehdad, Y., Narang, S., Malik, K., Fan, A., Bhosale, S., Edunov, S., Lewis, M., Wang, S., and Ma, H. Effective long-context scaling of foundation models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.260. URL https://aclanthology.org/2024.naacl-long.260/.

Yen, H., Gao, T., Hou, M., Ding, K., Fleischer, D., Izsak, P., Wasserblat, M., and Chen, D. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*, 2024.

Zhang, X., Chen, Y., Hu, S., Xu, Z., Chen, J., Hao, M. K., Han, X., Thai, Z. L., Wang, S., Liu, Z., and Sun, M. ∞bench: Extending long context evaluation beyond 100k tokens, 2024. URL https://arxiv.org/abs/2402.13718.

A. Needle Set Design & Considerations

In Table 7, we demonstrate the full needle set that we use in NoLiMa. In designing the needle templates, there are multiple considerations involved. First, all templates in the needle set begin with a small introductory phrase or at least one word (e.g., "Actually," "In 2013,") to distinguish themselves from the preceding context. This ensures that the needle's keyword or character is not inadvertently linked to the prior context. Since a newline is appended at the end of each needle, this issue is mitigated if the keyword or character appears at the end of the needle.

Question	Needles		Keyword Types	
Which character has been to W_q?	Def.	There was [CHAR] who was an engineer living in W_n.	W_n	Countries, cities, states
	Inv.	There was an engineer living in W_n, named [CHAR].	W_q	Countries, cities, states
Which character has been to W_q?	Def.	Actually, [CHAR] lives next to the W_n.	W_n	Buildings & Landmarks
	Inv.	W_n is next to where [CHAR] lives.	W_q	Countries, cities, states
Which character has been to W_q?	Def.	In 2013, after waiting in line for hours, [CHAR] finally saw the original W_n painting up close.	W_n	Buildings & Landmarks
	Inv.	In 2013, the original W_n painting was seen up close by [CHAR], finally, after waiting in line for hours.	W_q	Countries, cities, states
Which character cannot drink W_q?	Def.	A message came in from [CHAR] saying, "I'm W_n" and nothing more.	W_n	Dietary restriction (e.g., lactose intolerant)
	Inv.	A message came in saying, "I'm W_n," from [CHAR].	W_q	Drinks & Beverages
Which character cannot eat W_q?	Def.	Then [CHAR] mentioned that he has been W_n for years.	W_n	Dietary restriction (e.g., vegan)
	Inv.	There was a W_n guest, named [CHAR].	W_q	Foods

Table 7. Our proposed needle set templates in NoLiMa. The placeholders [CHAR], W_q, and W_n represent the randomly selected character (also the answer), the query keyword, and the needle keyword, respectively. Def.: default order. Inv.: inverted order.

Another consideration is that the needle keyword should be uniquely associated with the query keyword. For instance, in the following sentence:

There was an engineer living in *Cambridge*, named Yuki.

Although the term "Cambridge" is commonly associated with the "United Kingdom," it is not uniquely so; it could also refer to cities in the United States, Canada, or other countries. Additionally, we aim to avoid relying on language-specific markers. Many cities have distinctive elements in their names, such as orthographic features, morphological structures, or cultural naming conventions, that hint at their linguistic or geographic origins. By minimizing the influence of such markers, the needle design ensures a more rigorous evaluation of the model's ability to make meaningful connections based on learned knowledge rather than surface-level linguistic cues. For each template, we manually curated 2-6 keyword pairs, resulting in a total of 28 keyword pairs. Taking into account the order of fact statements, this generates 58 needle-question pairs.

B. Models

In Table 8, we list all the models selected for evaluation. Models that are open weights were deployed using the vLLM library (Kwon et al., 2023), with weights obtained from HuggingFace (Wolf et al., 2020).

C. Task Prompt Templates & Inference Settings

In Table 9, we present the task prompts used across all evaluations. While we do not employ the commonly used "Let's think step by step" prompt in the Chain-of-Thought (CoT) setup (Kojima et al., 2022), our prompt encourages the model to elaborate and expand its reasoning sufficiently before producing a final answer. To manage the extensive testing scope—7,540 tests per context length—we limit reasoning to three sentences or a maximum of 192 generated tokens. In the CoT setup, a test is considered successful if the final answer (on the newline) includes the correct answer. This differs with the non-CoT setup, where success is determined based on whether the correct answer is present within the

Model	Context Length	Open Weights?	Model Revision
GPT-4.1	1M	No	`gpt-4.1-2025-04-14`
GPT-4.1 mini	1M	No	`gpt-4.1-mini-2025-04-14`
GPT-4.1 nano	1M	No	`gpt-4.1-nano-2025-04-14`
GPT-4o	128K	No	`gpt-4o-2024-11-20`
GPT-4o mini	128K	No	`gpt-4o-mini-20240718`
Llama 4 Maverick	1M	Yes	`meta-llama/Llama-4-Maverick-17B-128E-Instruct`
Llama 4 Scout	10M	Yes	`meta-llama/Llama-4-Scout-17B-16E-Instruct`
Llama 3.3 70B	128K	Yes	`meta-llama/Llama-3.3-70B-Instruct`
Llama 3.1 405B	128K	Yes	`meta-llama/Llama-3.1-405B-Instruct`
Llama 3.1 70B	128K	Yes	`meta-llama/Llama-3.1-70B-Instruct`
Llama 3.1 8B	128K	Yes	`meta-llama/Llama-3.1-8B-Instruct`
Gemini 1.5 Pro	2M	No	`gemini-1.5-pro-002`
Gemini 1.5 Flash	1M	No	`gemini-1.5-flash-002`
Gemini 2.0 Flash	1M	No	`gemini-2.0-flash`
Gemini 2.5 Flash	1M	No	`gemini-2.5-flash-preview-05-20`
Gemma 3 27B	128K	Yes	`google/gemma-3-27b-it`
Gemma 3 12B	128K	Yes	`google/gemma-3-12b-it`
Gemma 3 4B	128K	Yes	`google/gemma-3-4b-it`
Claude 3.5 Sonnet	200K	No	`anthropic.claude-3-5-sonnet-20241022-v2`
Jamba 1.5 Mini	256K	Yes	`ai21labs/AI21-Jamba-1.5-Mini`
Command R+	128K	Yes	`CohereForAI/c4ai-command-r-plus-08-2024`
Mistral Large 2	128K	Yes	`mistralai/Mistral-Large-Instruct-2411`
		Reasoning-based models	
GPT-o1	128K	No	`gpt-o1-2024-12-17`
GPT-o3 Mini	128K	No	`gpt-o3-mini-2025-01-31`
DeepSeek R1-DL-70b	128K	Yes	`deepseek-ai/DeepSeek-R1-Distill-Llama-70B`

Table 8. Details of the selected models used for evaluation.

generated output. For all standard instruction-tuned models, we use greedy decoding during generation. For reasoning-based models, we utilize the default sampling decoding mechanism for GPT-o1 and GPT-o3 Mini, while R1-based models employ top-P sampling with p = 0.95 and a temperature of 0.6. In addition, we cap the maximum number of generated tokens in reasoning-based models at 1536 tokens, including both reasoning and output tokens. In all models, we apply each model's instruction-tuned chat templates.

D. Distractor Design

To construct and integrate the distractor sentences mentioned in Section 4.4.4, we devised two templates, applied uniformly across all needle-question pairs. Depending on the W_q, we use one of the following templates:

There was an article about W_q in the daily newspaper.

or

There was a photo of W_q in the daily newspaper.

Some instances of W_q may naturally include an article (e.g., ”a” or ”an”), making them better suited for the second template, while others fit the first. Regardless of the choice, the templates are designed to remain neutral and unrelated to the intent of the question or the fact stated by any needle.

To minimize interference with the needle, we randomly place the distractor sentence while ensuring a token distance of at least 20% of the context length. For example, in a 1K-token test, the distractor must be at least 200 tokens away from the needle. Additionally, to avoid any advantage from proximity to the beginning or end of the context (which may gain extra attention), we restrict placement to between the 20% and 80% marks of the context length. Together, these two constraints leave a span of 40%-60% of the context length available for random placement of the distractor sentence.

Mode	Prompt Template
w/o CoT	You will answer a question based on the following book snippet: {haystack w/ needle} Use the information provided in the book snippet to answer the question. Your answer should be short and based on either explicitly stated facts or strong, logical inferences. Question: {question} Return only the final answer with no additional explanation or reasoning.
w/ CoT	You will answer a question based on the following book snippet: {haystack w/ needle} Use the information provided in the book snippet to answer the question. Be aware that some details may not be stated directly, and you may need to INFER the answer based on the given information. Begin with a brief explanation of your reasoning in NO MORE THAN THREE (3) sentences. Then, return the final answer on a new line. Question: {question}

Table 9. Details of prompt templates utilized in our evaluation.

Models	Claimed Length	Effective Length	Base Score ($\times$0.85: **Thr.**)	1K	2K	4K	8K	16K	32K	64K	128K[7]
GPT-4.1	1M	16K	97.0 (82.5)	95.6	95.2	91.7	87.5	84.9	79.8	69.7	64.7
GPT-4o	128K	8K	99.3 (84.4)	98.1	98.0	95.7	89.2	81.6	69.7	62.4	56.0
Gemini 2.5 Flash	1M	2K	94.4 (80.2)	90.1	86.1	79.4	68.2	57.9	48.4	—	—
Gemini 2.0 Flash	1M	4K	89.4 (76.0)	87.7	87.5	77.9	64.7	48.2	41.0	33.0	16.4
Llama 4 Maverick	1M	2K	90.1 (76.6)	81.6	78.3	68.8	49.0	34.3	24.5	—	—
Gemma 3 27B	128K	<1K	88.6 (75.3)	73.3	65.6	48.1	32.7	20.2	9.5	—	—
Gemma 3 12B	128K	1K	87.4 (74.3)	74.7	61.8	39.9	27.4	16.8	7.3	—	—
Llama 4 Scout	10M	1K	81.7 (69.4)	72.3	61.8	50.8	35.5	26.9	21.6	—	—
GPT-4.1 Mini	1M	<1K	80.9 (68.8)	66.7	62.8	58.7	51.9	46.2	38.8	—	—
GPT-4.1 Nano	1M	<1K	80.7 (68.6)	60.8	48.2	36.7	28.8	19.5	9.4	—	—
Gemma 3 4B	128K	<1K	73.6 (62.6)	50.3	35.3	16.4	7.5	2.3	0.9	—	—

Table 10. NoLiMa benchmark results for GPT-4o, Gemini 2.0 Flash, and additional recent models. For models with stronger performance at 32K, we extend evaluation to 64K and 128K using the same setup. Scores above the effective threshold are underlined; scores below 50% of the base score are shaded in red.

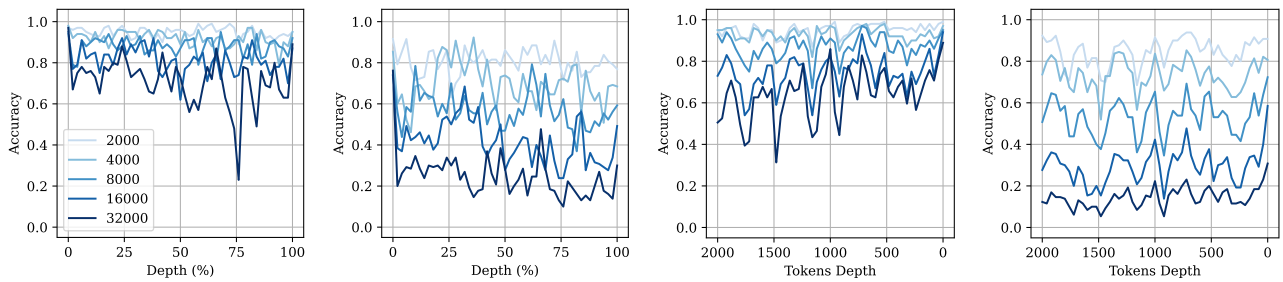
E. Results Beyond 32K & Recent LLMs

Based on the dataset configuration outlined in Section 4.1, we construct haystacks by randomly concatenating snippets extracted from the filtered books dataset. This setup enables evaluation across scalable context lengths, including 64K and 128K tokens, and **can be extended further** as model limits allow.

Two practical adjustments were made in this evaluation setup, without altering the core methodology: (1) Reduced Placement Count: Due to cost limitations, we reduce the number of needle placements from the default 26 to 11 placements per context length. This change minimizes API usage while still preserving meaningful coverage across the haystack. (2) Token Limit Constraints: GPT-4o has a strict 128,000 token limit, including both input and output tokens. To accommodate the task prompt and model response, we limit the haystack length to 127,500 tokens for this model. To ensure a fair comparison, we use the same haystack length for Gemini 2.0 Flash.

The results of this evaluation are presented in Table 10, focusing on GPT-4o and Gemini 2.0 Flash at 64K and 128K context

[7] 127,500-token haystack used based on model token limit constraints.



(a) Full Sweep (One-hop) (b) Full Sweep (Two-hop) (c) Last 2K (One-hop) (d) Last 2K (Two-hop)

Figure 6. Unsmoothed needle placement depth plots corresponding to the smoothed results in Figure 3. These plots reflect raw performance values prior to applying the moving average.

lengths. While GPT-4o does not fully meet its claimed performance at the maximum context length, it nonetheless maintains over 50% of its base score at 128K, indicating relatively strong performance at that scale. In contrast, Gemini 2.0 Flash shows a sharper decline, dropping to 16.4% at 128K.

To cover recent model releases, Table 10 also includes entries for newer models such as the GPT-4.1 series and Gemini 2.5 Flash[8] (Team, 2025; Gemini Team et al., 2024). For those that showed stronger performance at 32K, we also extend the evaluation to 128K using the same setup. GPT-4.1 shows clear improvements over prior models; however, its effective context length remains around 16K–well below the claimed 1M–and its performance drops below 65% on 128K context lengths.

F. One- & Two-hop Results

Tables 11 and 12 present detailed results on the one-hop and two-hop subsets of the NoLiMa benchmark, evaluated across the selected models. The tables follow the same format and thresholding criteria as Table 3, reporting both the claimed and effective context lengths. Note that the base scores–and consequently the thresholds–are computed separately for each subset. Although both one-hop and two-hop subsets show strong base scores, two-hop tasks generally yield shorter effective context lengths. This suggests that while models can perform well on complex reasoning in short contexts, their performance degrades more rapidly as context length increases, indicating reduced length generalization under greater reasoning demands.

Models	Claimed Length	Effective Length	Base Score (×0.85: Thr.)	1K	2K	4K	8K	16K	32K
GPT-4o	128K	16K	99.3 (84.4)	97.7	97.5	95.6	91.9	87.3	79.8
Llama 3.3 70B	128K	8K	97.7 (83.1)	97.1	93.6	90.3	84.1	73.2	56.2
Llama 3.1 405B	128K	4K	91.7 (78.0)	88.7	87.3	80.2	68.4	59.6	49.4
Llama 3.1 70B	128K	4K	96.4 (82.0)	95.2	89.8	82.7	76.9	66.3	56.9
Gemini 1.5 Pro	2M	4K	90.8 (77.2)	85.7	85.5	81.9	72.3	63.4	55.1
Jamba 1.5 Mini	256K	2K	93.1 (79.1)	80.0	80.2	77.9	71.5	62.9	56.0
Command R+	128K	2K	91.6 (77.8)	79.4	78.4	75.6	52.2	26.9	10.4
Gemini 2.0 Flash	1M	4K	92.5 (78.6)	91.5	93.5	89.6	78.4	61.8	52.8
Mistral Large 2	128K	4K	83.3 (70.8)	82.5	86.1	80.3	62.5	44.1	27.5
Claude 3.5 Sonnet	200K	8K	90.5 (76.9)	89.9	91.6	89.5	78.2	61.1	45.2
Gemini 1.5 Flash	1M	1K	85.7 (72.9)	76.4	71.8	63.6	57.0	48.7	41.3
GPT-4o mini	128K	1K	88.4 (75.2)	81.0	73.6	57.6	45.4	30.2	20.0
Llama 3.1 8B	128K	1K	83.0 (70.5)	75.7	69.3	60.7	49.6	35.7	22.7

Table 11. NoLiMa benchmark results on one-hop examples. Base scores and effective lengths are computed using only the one-hop subset. Scores above the effective threshold are underlined, while scores that are below 50% of the base score are shaded in red.

[8]Without reasoning (thinking budget = 0).

Models	**Claimed Length**	**Effective Length**	**Base Score (×0.85: Thr.)**	**1K**	**2K**	**4K**	**8K**	**16K**	**32K**
GPT-4o	128K	8K	99.3 (84.4)	98.7	98.7	95.8	85.9	74.6	57.4
Llama 3.3 70B	128K	1K	96.7 (82.2)	90.6	79.8	70.7	57.4	42.7	25.9
Llama 3.1 405B	128K	2K	95.3 (81.0)	89.4	82.0	67.4	49.8	34.6	23.8
Llama 3.1 70B	128K	1K	92.0 (78.2)	85.9	72.0	57.0	45.2	33.8	26.4
Gemini 1.5 Pro	2M	1K	94.9 (80.7)	87.1	79.4	67.4	53.6	45.7	39.6
Jamba 1.5 Mini	256K	<1K	91.7 (77.9)	71.8	66.6	62.0	50.7	40.0	28.4
Command R+	128K	<1K	90.0 (76.5)	74.0	67.4	54.7	23.8	14.3	3.8
Gemini 2.0 Flash	1M	2K	85.7 (72.8)	83.1	80.0	63.6	47.8	31.4	26.4
Mistral Large 2	128K	2K	93.6 (79.5)	90.4	84.7	64.7	37.8	18.4	7.9
Claude 3.5 Sonnet	200K	2K	84.0 (71.4)	79.8	74.5	63.0	41.5	26.9	10.9
Gemini 1.5 Flash	1M	<1K	83.5 (71.0)	59.1	49.0	35.6	28.8	19.2	12.9
GPT-4o mini	128K	<1K	80.4 (68.3)	51.4	39.3	27.6	16.9	8.8	5.9
Llama 3.1 8B	128K	<1K	68.9 (58.6)	53.4	36.0	23.6	10.0	6.5	3.8

Table 12. NoLiMa benchmark results on two-hop examples. Base scores and effective lengths are computed using only the two-hop subset. Scores above the effective threshold are underlined, while scores that are below 50% of the base score are shaded in red.

G. Raw Needle Placement Depth Plots

Figure 6 presents the needle placement depth plots corresponding to Figure 3, prior to the application of the moving average employed in the main figure.