Puzzle in a Haystack: Understanding & Enhancing Long Context Reasoning

Stanford CS224N **Custom** Project

Sudharsan Sundar
Department of Computer Science
Stanford University
sjsundar@stanford.edu

Jessica Chudnovsky
Department of Computer Science
Stanford University
jchud@stanford.edu

Salman Abdullah
Department of Computer Science
Stanford University
salman01@stanford.edu

Abstract

As the context windows of Large Language Models (LLMs) increase, they have the ability of accepting entire novels to textbooks as input. The popular Needle in a Haystack (NIAH) evaluation, while providing a minimum standard for evaluating this long context performance, falls short of assessing the more important reasoning and information synthesis capabilities of possible with long context LLMs. Hence, we introduce the Progressive Needles Test: a simple logic puzzle to evaluate a model's ability to reason over, synthesize, and deduce information from multiple parts of its inputted context. In the Progressive Needles test, we place information relevant to a query ("needles") within a larger text of thematically related but irrelevant information ("haystack"). The needles are logically connected to one another, necessitating the models to engage in deep reasoning to extract and synthesize this scattered information to arrive at the correct answer. We generate Progressive Needles questions for haystacks for both natural language numerical/mathematical reasoning tasks as well as code tasks, the latter simulating chained function calls across code bases. We find that LLMs like GPT-4, GPT-3.5, and Mixtal exhibit a marked decline in performance on the Progressive Needles test when the size of the haystack is increased and queries are made to require more complex reasoning, exposing gaps both within current long context benchmarks and weaknesses in LLM's reasoning abilities. By fine-tuning GPT-3.5 on the Progressive Needles tasks, we also demonstrate that learning to solve Progressive Needles tasks leads to a tangible improvement of $\sim$2% in performance on the real-world QuALITY benchmark, suggesting that our task helps enhance LLM reasoning capabilities and other real world tasks.

1 Key Information to include

- Mentor: Tathagat Verma.
- External Collaborators (if you have any): N/A.
- Sharing project: No.
- *Sudharsan's contribution*: Primarily wrote the report, developed Progressive Needles task generation, developed research directions, and helped design the poster. *Jessica's contribution*: Primarily designed the poster, ran key Progressive Needles evaluations, ran fine-tuning experiments, thought of research directions, and wrote the report. *Salman's contribution*: Pri-

marily set up evaluation on QuALITY datasets, ran the fine-tuning experiments, developed research directions, edited the report, and helped design the poster.

2 Introduction

Recently released advanced large language models (LLMs) have boasted context windows in the tens or hundreds of thousands of tokens, with some even allowing for million-token inputs (Achiam et al., 2023; Jiang et al., 2024; Team et al., 2023; Anthropic, 2024). Such advances enable LLMs to be prompted in-context with the entirety of large texts—such as whole novels, screenplays, and textbooks, being asked to analyze and reason over this entire corpus of information. However, due to the novelty of such large context windows and the challenge that comes of gathering enough data to push these large context models to their limits, there is a lack of methods for evaluating the ability of these models to effectively use such large corpuses of information when context is passed in.

One of the most popular methods currently used for long-context evaluation is the Needle-In-a-Haystack test (NIAH). This tests the model's ability to retrieve one (or many) isolated and independent facts (the "needles"), which are inserted at various points in a large text of irrelevant information (the "haystack") Team et al. (2023); Anthropic (2024). The Gemini 1.5 announcement claims high performance across long context windows due to its high performance on NIAH. Though such NIAH methods are useful as a minimum standard for effective long-context performance, they do not capture one of the most important promises of long context *LLMs*: the ability to *reason* over a large amount of information, such as by *synthesizing* various pieces of information in the text and, thereby, being able to *deduce* non-trivial conclusions. We introduce the novel *Progressive Needles Test*, a form of puzzle and method for evaluating long context *reasoning* in LLMs. We task models with *synthesizing* various pieces of information in order to generate the correct conclusion to a query. In particular, we insert N pieces of relevant information (the "needles") into a large, irrelevant body of text (the "haystack"), where the nth piece of information ("needle") directly *depends* on the $n + 1$th piece of information (another "needle"), e.g. the information "The value of Needle 0 is the value of Needle 1 plus 6" directly *depends* on the information "The value of Needle 1 is 5". We then provide the model with the haystack with needles inserted (i.e. relevant information randomly inserted in a large irrelevant text), and query the model to answer a question. The question requires the model to synthesize the information from all N needles scattered throughout the haystack in order to deduce the correct conclusion. We find that advanced long context LLMs such as Mixtral 8x7B Mixture of Experts model and the January 2024 release of GPT-3.5 **experience a sharp decline in performance when the relevant information is scattered across a large corpus** as opposed to when solely the relevant information is passed in.

We demonstrate two task settings for the Progressive Needles test: a *numerical* reasoning setting, and a *code* reasoning setting. We demonstrate that fine-tuning GPT-3.5 on the Progressive Needles task, hence improving a model's ability to solve this "puzzle," leads to an approximately **2% increase in accuracy on a real-world, non-synthetic question-answering benchmark**, specifically QuALITY (Pang et al., 2022). An increased performance on the Progressive Needles task is likely an indicator of better reasoning and information synthesis abilities in more practical tasks of interest.

3 Related Work

As recent language models have scaled up their context windows, significant research has been done to understand and evaluate how models perform in longer context settings. The Needle in a Haystack task (Kamradt, 2024) challenges models with retrieving a randomly placed statement when queried to do so. However, this is more aligned with a retrieval task as opposed to requiring reasoning across the corpus. Gemini 1.5 Team et al. (2023); Anthropic (2024), with context lengths of 1 Million tokens and 200k tokens respectively, are both evaluated on this task, and demonstrate near perfect performance.

Liu et al. (2024) notes that as context lengths increase, models tend to under utilize the full context and performance can degrade significantly on long multi-document question answering tasks. While our work focuses on how different parts of the corpus interact with one another, Liu et al. (2024) is an essential step toward understanding how models are currently unable to attend to all relevant parts of a larger corpus, even when the model has larger context windows and the information itself is relevant to the task.

Datasets such as HotpotQA and NarrativeQA have further emphasized the need for models that can effectively handle long contexts Kočiský et al. (2018); Yang et al. (2018). HotpotQA is a multi-hop question answering dataset that requires models to reason over multiple paragraphs, and understand connections between them, to arrive at the correct answer. NarrativeQA, on the other hand, focuses on understanding and answering questions based on long narrative texts, such as books and movie scripts. Answering questions in these datasets requires reasoning over complicated parts of long passages and understanding the relationships between them.

Srivastava et al. (2024) also proposes a framework for effective reasoning benchmarks: namely, that a static version of the benchmark should be complemented by a functional variant to accurately measure and reduce the reasoning gap between memorized and dynamically reasoned responses. This inspires our dynamic benchmark, which randomly generates needles for a provided hay, reducing the likelihood that high performance on this benchmark can be attributed to memorization.

4 Approach

The Progressive Needles test is primarily designed to pressure test the ability of long context LLMs to effectively synthesize and reason about information in large texts. To that end, we create a randomized programmatic method of generating Progressive Needle questions. Each question consists of: (1) *information*, i.e. the needles inserted in a haystack of varying size; and (2) a *query*, which requires the model the synthesize information from the needles to deduce the correct answer. An instance of the Progressive Needle test is further parameterized by the following: (1) the type of question, i.e. the type of task that the model is required to solve; (2) the number of needles used, which determines the number of reasoning steps the model must go through to solve the question; and (3) the size the haystack used, i.e. the number of tokens of irrelevant information that the needles will be inserted into. We consider two types of tasks, *numerical* reasoning and *code* reasoning, and describe their implementations below. This approach is, to the best of our knowledge, a novel method of evaluating long context reasoning. Hence, we independently wrote nearly *all* of the codebase required to run and analyze the Progressive Needles test, which can be accessed here, including the entire pipeline for generating Progressive Needles questions in various settings, running evaluation of various models, and analyzing resulting model outputs (we borrow a few data utility functions from these sources where necessary to speed up development).

Consider a Progressive Needles test with N needles per question and a haystack with M tokens.

4.1 Numerical Reasoning Setting

Information To generate the *information*, we first generate N needles, which are pieces of relevant information necessary to answer the query. In the numerical reasoning setting, these needles take the form

"*The value of Needle n is equal to the value of Needle $n+1$ [plus/minus] [x]*"

for all $0 \leq n < N, x \in \mathbb{N}, x \leq 10$, and

"*The value of Needle n is equal to [x]*"

for $n = N, x \in \mathbb{N}, x \leq 10$.

Then, we create a haystack of M tokens of irrelevant text by taking the first M tokens of a classic mathematical treatise by Alfred North Whitehead, "An Introduction to Mathematics" Whitehead (2017), which is thematically related to the needles but is irrelevant for answering the query. We choose a thematically similar haystack in order to more closely emulate the real-world use of LLMs, since, when passing e.g. an entire book in-context to a model and asking it to answer a specific question about the book, the model must be able to ignore thematically similar but irrelevant information when generating a response.

Finally, we insert the N needles in random order and at random positions within the haystack text to generate the information passed into the model. See Figure 1 for an illustration of this process.

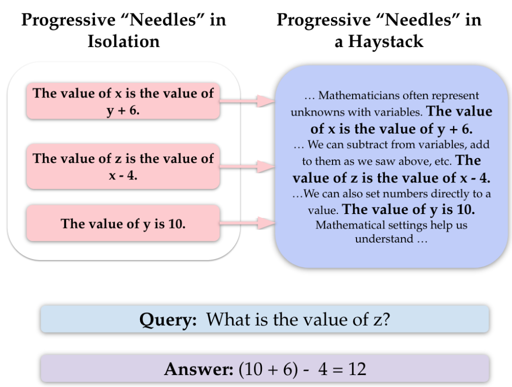

Figure 1: Process of placing numerical progressive needles in a haystack.

Query The query posed to the model is "*What is the value of Needle 0?*" Hence, by construction of the needles, correctly answering this query requires recognizing that to find the value of Needle 0, one must find the value of Needle 1, and therefore Needle 2, and so on, until reaching Needle N, for which a number value is directly provided. Furthermore, in addition to recognizing the relevant information, the model must synthesize all the various pieces of information contained in the N needles in order to arrive at the correct conclusion: the numerical value of any Needle n, $n < N$, is never explicitly mentioned in the text, so the model must deduce its value by incorporating information about the value of the $n + 1$th Needle over N steps of reasoning to reach the value of Needle 0. See Figure 2 for an illustration of the depth of reasoning and information synthesis required by the Progressive Needles query.

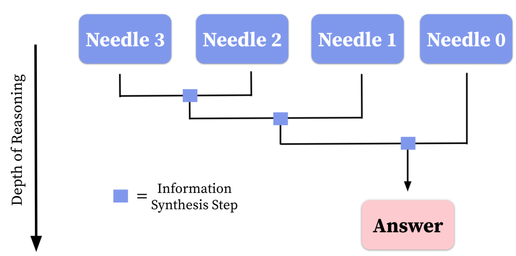

Figure 2: Reasoning over Progressive Needles Query

4.2 Code Reasoning Setting

Information The code reasoning setting is structurally identical to the numerical reasoning setting. However, the needle information is formatted in terms of a simple Python function:

```
def get_value_of_needle_n():  return get_value_of_needle_[n+1] [+/-] [x]
```

for all $0 \leq n < N, x \in \mathbb{N}, x \leq 10$; and

```
def get_value_of_needle_n():  return [x]
```

for $n = N, x \in \mathbb{N}, x \leq 10$.

For our haystack corpus, we use the first M tokens from the concatenation of all functions used in the HumanEval benchmark (in particular, the "solution" function for each HumanEval question) (Chen et al., 2021), as these functions are thematically related to the code-based needles above but irrelevant for solving the query.

Finally, we insert the N needles in random order and at random positions within the haystack text to generate the information passed into the model (see Figure 3). Since the haystack is composed of standalone functions (i.e. functions which are not nested within other functions or classes), we can insert the code needles in between functions and assure that the resulting information is a syntactically and functionally correct Python program.

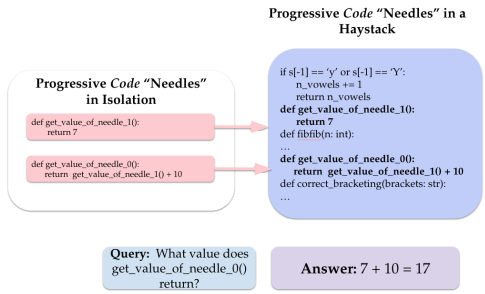

Figure 3: Process of placing code progressive needles in a haystack.

Query Similar to the numerical setting, the query for code reasoning is "*What value is returned by `get_value_of_needle_0()`?*". In this setting, correctly answering the query requires correctly following N chained function calls, since each nth function calls an $n+1$th function for $n < N$.

4.3 Baseline Comparison: No Haystack

Since the Progressive Needle evaluation focuses on long context reasoning in particular, a natural baseline to contextualize long context reasoning abilities is "short" context reasoning, i.e. model performance when it is *only* presented with relevant information, and no irrelevant information. As such, the baseline for the Progressive Needles task is when the information passed into the model is only the needles (in randomly shuffled order), i.e. "needles only", with no haystack of irrelevant information. Thus, degradation in performance relative to this "short" context setting demonstrates that a model is performing worse than it is otherwise capable of, and that this is *specifically* due to having to reason over a long context (rather than some inherent difficulty of the problem).

5 Experiments

We analyze the performance of advanced LLMs with relatively long context windows ($>$ 10k tokens) such as GPT-3.5, Mixtral 8x7B, and GPT-4 Achiam et al. (2023); Jiang et al. (2024); OpenAI et al. (2024) in both the numerical reasoning and code reasoning settings, for varying numbers of needles and haystack sizes (due to cost constraints, we consider a limited but informative number of settings). In addition, we perform fine-tuning experiments on GPT-3.5 to validate the usefulness of solving Progressive Needles tasks for increasing performance on real-world, non-synthetic tasks.

5.1 LLM Performance in the Numerical Reasoning Setting

We evaluate GPT-3.5 and Mixtral 8x7B on 75 randomly generated Progressive Needles questions in the numerical reasoning setting as described in Section 4.1; we evaluate GPT-4 on 50 questions due to budget constraints. We evaluate GPT-3.5 and Mixtral 8x7B with needles-only information (no haystack), 2k tokens of haystack, 7k tokens of haystack, and 12k tokens of haystack; likewise, we consider model performance for both $N = 2$ and $N = 4$, where N represents the number of needles used. For GPT-4, we only evaluate with needles-only information (no haystack) and 20k tokens of haystack, to provide the more advanced model with a more difficult test setting 9. Although our analysis is easily extended to large values of N, we limit our primary analysis to four needles or less since, in real-world settings, the number of reasoning and information synthesis steps required to correctly answer a query typically involves combining information from a handful of distinct facts, rather than iteratively synthesizing information over, say, 10 mutually distinct pieces of information.

To assess performance for all models, we use exact match accuracy, since the correct answer to any Progressive Needles question is a recursively calculable integer value. Furthermore, when evaluating a model, we append the instruction "Let's think step by step" to the question prompt in order to elicit greater reasoning capabilities from the model. For all models, we generate with temperature set to 0 in order to limit the stochasticity of our results.

We provide results for for both analyzed settings: two needles and four needles (Tables 1 and 2). Importantly, we find that greater haystack size decreases accuracy on numerical Progressive Needles questions. Although this result is expected for smaller models such as Mixtral, surprisingly, *this trend also holds for the most advanced model we evaluate*, GPT-4, which sees an 18% decrease in performance between the needles-only and 20k haystack information settings. In addition, the *severity of decline* in accuracy for GPT-3.5 and Mixtral in a problem setting as simple as 2 needles is surprising, as accuracy drops by over 50% for both models when going from needles-only information to needles hidden in a 12k haystack. Furthermore, we find that model performance becomes *more sensitive to the haystack size* when more needles are used, i.e. when correctly answering the question requires deeper reasoning and more steps of information synthesis. An important observation in this regard is that the results we observe with the numerical Progressive Needles evaluation *diverge significantly from previous results with Needle-in-the-Haystack (NIAH) evaluations* in that we observe significant degradation of model performance at relatively small context lengths (less than 20k tokens), in contrast to a previous NIAH evaluation which finds that both Mixtral and GPT-4 suffer approximately 0 loss in simple retrieval accuracy over up to 30k context sizes (Dhinakaran and Jolley, 2024).

Haystack size	GPT3.5	Mixtral
0	100.00%	98.70%
2k	93.70%	89.30%
7k	29.30%	38.70%
12k	29.30%	45.30%

Table 1: Evaluation of performance on *two* needles, *numerical* setting.

Haystack	GPT3.5	Mixtral
0	98.70%	98.70%
2k	81.30%	38.70%
7k	18.70%	6.67%
12k	10.70%	5.33%

Table 2: Evaluation of performance on *four* needles, *numerical* setting.

Haystack Size (Token Length)	Accuracy
0	98%
20k	80%

Table 3: GPT-4 evaluation on *four* needles, *numerical* setting.

5.2 LLM Performance in the Code Reasoning Setting

Identical to the numerical needle setting, we evaluate GPT-3.5 and Mixtral 8x7B on 75 randomly generated Progressive Needles questions in the code reasoning setting as described in Section 4.2; we evaluate GPT-4 on 50 questions. We evaluate GPT-3.5 and Mixtral 8x7B with needles-only information (no haystack), 2k tokens of haystack, 7k tokens of haystack, and 12k tokens of haystack; likewise, we consider model performance for both $N = 2$ and $N = 4$, where N represents the number of needles used. For GPT-4, we only evaluate with needles-only information (no haystack) and 20k tokens of haystack.

Again, to assess performance for all models, we use exact match accuracy, and when evaluating a model, we append the instruction “Let's think step by step" to the question prompt in order to elicit greater reasoning capabilities; we generate with temperature set to 0.

We provide results for both the 2 needle and 4 needle settings (Tables 4 and 5). Interestingly, we find that GPT-3.5 and Mixtral performance are higher across the board in the code setting compared to the numerical setting. This likely arises from the fact that properly evaluating code with chained functions calls is likely more in-distribution with regards to the code data that these LLMs are most likely trained on, hence higher overall performance in the code setting is to be expected. That being said, we nonetheless observe a trend of increasing haystack size leading to decreased accuracy, particularly in the more challenging problem setting of 4 needles, where deeper reasoning and information synthesis is required to correctly answer a question.

Haystack size	GPT3.5	Mixtral
0	100.00%	100%
2k	100%	96.00%
7k	90.70%	90.70%
12k	73.30%	94.70%

Table 4: Evaluation of performance on *two* needles, *code* setting.

Haystack size	GPT3.5	Mixtral
0	98.70%	94.70%
2k	87%	84%
7k	84.00%	82.70%
12k	73.30%	58.70%

Table 5: Evaluation of performance on *four* needles, *code* setting.

5.3 Fine-tuning on the Progressive Needles Test

In order to establish the connection between performance on the Progressive Needles test and performance on non-synthetic, real-world data, we fine-tune GPT-3.5 on a corpora of Progressive Needles numerical reasoning questions. Each training example in this corpora consists of: information, which can vary from 1k to 13k tokens of haystack text and 2 to 6 numerical needles; a query asking for the value of Needle 0; and a procedurally generated answer that includes step-by-step reasoning for correctly answering the given query. We use 89 questions, consisting of approximately 650k tokens total; we use the default hyperparameters provided by OpenAI, as these cannot be customized.

We hypothesize that models that have learned to perform well on Progressive Needles should exhibit improved long-context reasoning capabilities on other real-world long context reasoning/synthesis tasks. We use the QuALITY dataset (Pang et al., 2022) as our real-world “long" context task. The QuALITY dataset contains over 2000 multiple-choice questions that assess question-answering based on a relatively long passage of text (on average, approximately 5k tokens). QuALITY questions are rated either easy or hard, where hard questions, on average, take a human being more than 45 seconds to answer correctly. In particular, we choose this benchmark since correctly answering the questions requires proper information synthesis and reasoning over the inputted text passage, but in a real-world context that is very different (i.e. far out of distribution) from the Progressive Needles test.

We compare our fine-tuned model against the base, non-fine-tuned GPT-3.5 model. We find that fine-tuning results in a performance boost for both the easy and hard subsets of the QuALITY dataset, with an overall increase in performance of approximately 1.9%, i.e. answering about 40 additional questions correctly 6.

This demonstrates that the “skills" required by an LLM to solve the Progressive Needles test are similar to those that also underlie information synthesis and reasoning over varied, real-world tasks, even those as relatively unrelated as the QuALITY benchmark.

Model	Question Type	Accuracy
GPT-3.5-turbo	Accuracy on Easy Questions	0.775
	Accuracy on Hard Questions	0.605
	Overall Accuracy	0.688
GPT-3.5 fine-tuned on numerical needles data	Accuracy on Easy Questions	**0.803**
	Accuracy on Hard Questions	**0.614**
	Overall Accuracy	**0.707**

Table 6: GPT 3.5-turbo vs. fine tuned model performance on QuALITY dataset

6 Analysis of Progressive Needles Performance

To further dissect LLM performance on the Progressive Needles test, we perform an error analysis of the evaluation results of models such as Mixtral 8x7B and GPT-4. In particular, we investigate using a continuous metric for scoring model responses and the incidence of model “refusals", i.e. where the model responds that the question cannot be answered.

6.1 Continuous Scoring

To validate the fact that the decrease in model performance from greater amounts of hay tokens and a greater number of needles used is not simply a byproduct of stringent evaluation criteria

Schaeffer et al. (2024), we reevaluate model responses using a continuous, less stringent metric. In particular, we use the average absolute value of the difference between the model's prediction and the correct answer (we skip responses where the model refuses to provide a number answer); since this is a loss-like metric, lower scores are better. We find that, even for GPT-4, the average deviation of the answer in the most difficult haystack setting (20k tokens) is 1.6x that of the average deviation of answers in the needles-only (no haystack) setting (Table 8). For less advanced models, this difference is more pronounced: Mixtral 8x7B exhibits up to a 100x worse performance in its most difficult question setting (numerical reasoning, four needles, 12k tokens of haystack) compared to the needles-only setting (numerical reasoning, four needles, 0 tokens of haystack; see Table 7, the four needle). Thus, this indicates that even when the model is "confident" enough to attempt to provide a number answer to the question, its ability to extract and/or synthesize the necessary information is nonetheless worse than when it performs in the small context setting with no irrelevant information.

6.2 Refusal Rates

When qualitatively analyzing model responses on the Progressive Needles test, we find that a fairly common failure mode for models on tasks with large haystacks was "refusal": responding to the query by stating incorrectly that there is insufficient information provided, and "refusing" to provide a numerical answer. When filtering for model answers marked incorrect which also include key words such as "cannot be determined" and "unable", we find that refusal rates are much higher for Mixtral 8x7B than for GPT-4 (as seen in Tables 7 and 8). Hence, an interesting implication of this finding is that "refusals" may be an important shortcoming of less advanced models for long-context reasoning tasks, particularly when the information passed in context is *sparse* in relevant information—often naturally the case when asking targeted questions about large texts. Furthermore, this again demonstrates the importance of long context *reasoning* tasks, like Progressive Needles, in particular: Mixtral 8x7B and GPT-4 perform similarly at single-needle Needle in a Haystack evaluations at context sizes of up to 30k tokens (Dhinakaran and Jolley, 2024), but, when tested on deeper reasoning over information in long context settings via the Progressive Needles test, the large quantitative and qualitative gaps in performance between the models come into sharper relief. Finally, it is important to note that refusal rates for needles-only information, with no irrelevant haystack information, are 0% for both models (due to overall model performance being near or at 100% accuracy), indicating that model refusals are a direct result of the sparsity of relevant information in context, rather than an inherent difficulty in the reasoning required to solve the task.

7 Conclusion

The development of long context LLMs has been an important step forward in capabilities of language models, allowing for, in some cases, entire textbooks to be passed in context to a model. However, effective methods for evaluating the long context performance of LLMs have yet to catch up. Although an effective minimal standard for long context information processing, the popular Needle in a Haystack evaluations do not require sufficient depth of reasoning and information synthesis to test the higher order and most promising abilities of long context models. The Progressive Needles test, whether in the numerical or code setting, allows for an effective, objective, and automatic evaluation of long context models which also critically tests the ability of models to *reason* over the information provided in context and *synthesize* relevant pieces of information in order to *deduce* the correct conclusion. When evaluating models on the Progressive Needles test, we see a strong and reliable trend of decreased performance as haystack size increases, particularly in the numerical reasoning setting; we find this trend holds, even for advanced models such as GPT-4 when evaluated in the numerical reasoning setting. Furthermore, we find that fine-tuning on numerical Progressive Needles questions leads to an increase in performance on the real-world, non-synthetic question-answering benchmark QuALITY; we conjecture that this stems from the fact that solving the Progressive Needles test does indeed require general reasoning and information synthesis skills, which are transferable to real-world long context tasks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. `https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf`. Accessed: 2024-03-18.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Aparna Dhinakaran and Evan Jolley. 2024. The needle in a haystack test: Evaluating the performance of llm rag systems.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

G. Kamradt. 2024. Llmtest: Needle in a haystack.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36.

Saurabh Srivastava, Annarose M B, Anto P V au2, Shashank Menon, Ajay Sukumar, Adwaith Samod T, Alan Philipose, Stevin Prince, and Sooraj Thomas. 2024. Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Alfred North Whitehead. 2017. *An introduction to mathematics*. Courier Dover Publications.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

A Benefits of Progressive Needles as an Evaluation Framework

There are many benefits to using the Progressive Needles test to evaluate long context reasoning, principal among them: (1) The task is easily adaptable to any context length, since the haystack text used can be set to be an arbitrary number of tokens; (2) The task is randomized and, hence, not easily memorized/contaminated; (3) Correct answers are exact and objectively determined; (4) Correctly answering the query requires *long context reasoning to determine which information is relevant*, and *long context information synthesis* of various pieces of information in order to *deduce* the proper conclusion, such as a specific numerical value.

B Limitations

One limitation of our evaluation framework is that we want to replicate real world tasks. Mentions of needles in texts that are different in topic may not be representative of real world tasks. Further, the reasoning steps are not more complicated than addition, hence one improvement that can be made is challenging models with more rigorous reasoning problems. Another potential extension of our work is generating needles that are more natural to the context/hay they are embedded within.

C Error Analysis Tables

Here we present results for our error analysis discussion in Section 6.

Haystack size	Avg. Absolute Deviation from Correct Answer	Refusal Rate
0	0.45	0%
2k	4.79	44%
7k	9.37	72%
12k	9.27	67%

Table 7: Error analysis for Mixtral in the numerical reasoning setting, 4 needles

Haystack size	Avg. Absolute Deviation from Correct Answer	Refusal Rate
0	0.32	0%
20k	0.54	12%

Table 8: Error analysis for GPT-4 in the numerical reasoning setting, 4 needles

D Plots of Progressive Needles Performance

Here we present plotted visualizations of various GPT-3.5 and Mixtal's performance in various Progressive Needles test settings.

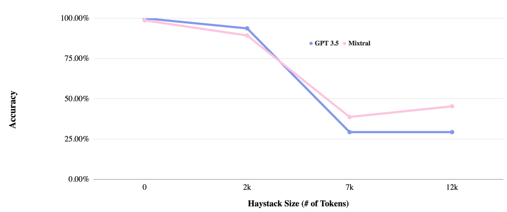

Figure 4: Two Needles with Numerical Examples

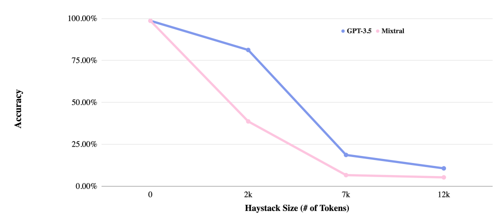

Figure 5: Four Needles with Numerical Examples

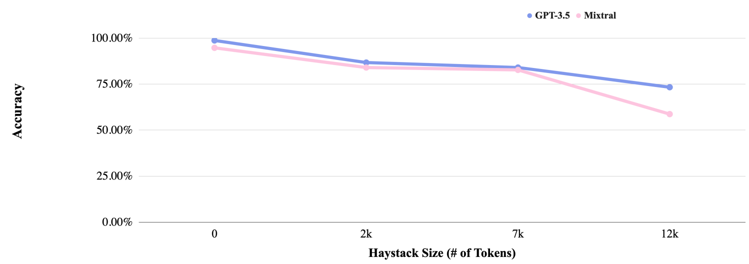

Figure 6: Four Needles with Code Examples

Model	Task type	Num needles (shuffled)	Haystack tokens (content)	Accuracy without Haystack (Needles-only)	Needles in Haystack Accuracy
Claude 3 Opus	Numerical	15	∼10k (Brothers K.)	0.82	0.24
Claude 3 Sonnet	Numerical	15	∼10k (Brothers K.)	0.79	0.38
GPT-4 (1/25)	Numerical	12	∼5k (Brothers K.)	0.92	0.73
GPT-3.5	Code	5	∼3.7k (Human eval)	0.96	0.82
GPT-3.5	Code	5	∼7.5k (Human eval)	0.98	0.6
GPT-3.5	Code	10	∼3.7k (Human eval)	0.80	0.66
GPT-3.5	Code	10	∼7.5k (Human eval)	0.80	0.44
Mixtral MoE	Numerical	2	4699 (Brothers K.)	0.86	0.68
Mixtral MoE	Numerical	4	4699 (Brothers K.)	0.94	0.24
Mixtral MoE	Numerical	2	9073 (Brothers K.)	0.86	0.7
Mixtral MoE	Numerical	4	9073 (Brothers K.)	0.94	0.14
Mixtral MoE	Code	2	3769 (Human Eval)	1	0.88
Mixtral MoE	Code	4	3769 (Human Eval)	0.9	0.92
Mixtral MoE	Code	2	7582 (Human Eval)	1	0.92
Mixtral MoE	Code	4	7582 (Human Eval)	0.96	0.84
Mixtral 7B	Numerical	2	4699 (Brothers K.)	0.84	0.28
Mixtral 7B	Numerical	4	4699 (Brothers K.)	0.74	0.06
Mixtral 7B	Numerical	2	9073 (Brothers K.)	0.84	0.16
Mixtral 7B	Numerical	4	9073 (Brothers K.)	0.76	0.00
Mixtral 7B	Code	2	3769 (Human Eval)	0.92	0.34
Mixtral 7B	Code	4	3769 (Human Eval)	0.38	0.02
Mixtral 7B	Code	2	7582 (Human Eval)	0.92	0.42
Mixtral 7B	Code	4	7582 (Human Eval)	0.38	0.04
GPT-3.5-Turbo	Numerical	2	4699 (Brothers K.)	0.86	0.72
GPT-3.5-Turbo	Numerical	4	4699 (Brothers K.)	0.94	0.64
GPT-3.5-Turbo	Numerical	2	9073 (Brothers K.)	0.86	0.58
GPT-3.5-Turbo	Numerical	4	9073 (Brothers K.)	0.94	0.24
GPT-3.5-Turbo	Code	2	3769 (Human Eval)	1	0.9
GPT-3.5-Turbo	Code	4	3769 (Human Eval)	0.98	0.76
GPT-3.5-Turbo	Code	2	7582 (Human Eval)	1	0.86
GPT-3.5-Turbo	Code	4	7582 (Human Eval)	1	0.62
Claude 3 Sonnet	Numerical	4	∼15k (Dost)	0.87	0.5

Table 9: Plots of Progressive Needles Performance