FACTOR: Factoring Complexity and Context Length in Long-Context LLM Evaluation

Hongyi Liu*‡, Yang Zhou‡, Zhuoming Chen‡,
Yuandong Tian§, and Beidi Chen‡

‡Carnegie Mellon University
§FAIR, Meta

November 7, 2024

Abstract

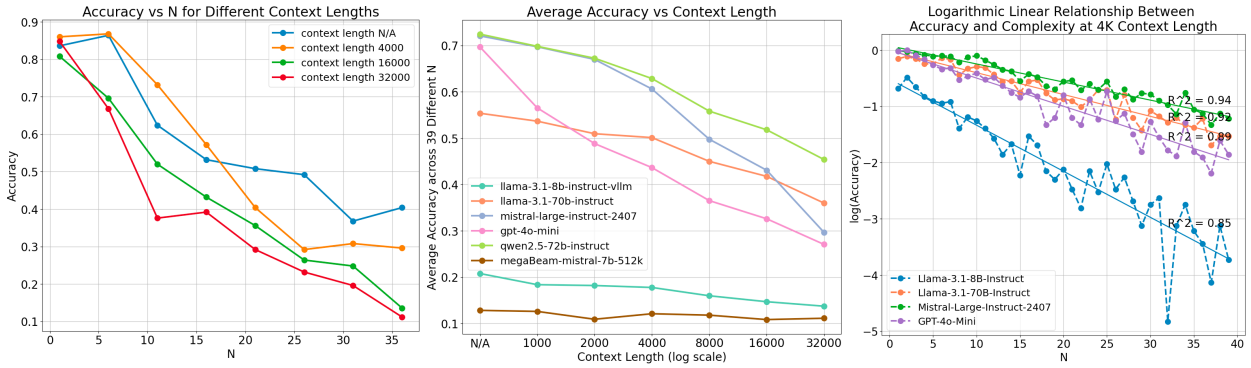
Large language models (LLMs) with extended context windows have shown remarkable capabilities, especially with contexts up to 128K tokens. However, whether these resource-intensive LLMs genuinely surpass simpler Retrieval Augmented Generation (RAG) techniques remains debated. We precisely delineate differences between long-context LLMs and RAG methods, emphasizing the unique long-context reasoning abilities of LLMs that RAG cannot replicate. Existing benchmarks often focus on retrieval tasks and contain weak if not none complex reasoning tasks, hindering assessment of reasoning over extended contexts. We introduce the **FACTOR** benchmark (**F**actoring **A**nalysis of **C**omplexity and **T**extual **C**ontext in **R**easoning). FACTOR consists of two suites of tasks, covering both the *symbolic* and *real-world* facets of reasoning evaluation. Also, both suites are carefully curated to delineate task complexity and context length when evaluating LLMs. We present detailed evaluations of popular LLMs on FACTOR. Besides mere accuracy scores, we also model the relationship between accuracy and complexity given the context length. A simple but consistent log-linear relationship works surprisingly well across various models. From the log-linear relationship fitted, two explainable parameters, the slope or Complexity Decay Factor (CDF) and the y-intercept or Contextual Decay Offset (CDO) are shown to offer separate and insightful measures of the models' complex reasoning and long context innate ability. Our findings highlight distinct failure modes linked to task complexity and context length, underscoring the unique reasoning capabilities of long-context LLMs unattainable by RAG methods.

1 Introduction

Recently, large language models with extended context windows have demonstrated exceptional performance in real-world applications (Achiam et al., 2023; Team et al., 2024). With the advent of next-generation models (Dubey et al., 2024), context lengths of up to 128K tokens are becoming the new norm. Despite these advancements, a persistent debate exists (Li et al., 2024; Yu et al., 2024) regarding whether sophisticated and resource-intensive long-context LLMs genuinely offer advantages over more straightforward and cost-effective Retrieval Augmented Generation (RAG) techniques. This paper aims to precisely delineate the differences between what LLMs can accomplish with extended context capabilities and what is attainable through RAG methods. We contend that long-context LLMs possess unique long-context reasoning abilities that are inherently challenging for RAG-based methods to replicate.

While existing benchmarks, such as RULER (Hsieh et al., 2024) and ∞Bench (Zhang et al., 2024), cover a range of tasks and are becoming the new paradigm for evaluating the long-context models, they often fail to capture the fundamental distinctions between the reasoning capabilities of long-context LLMs and the retrieval strengths of RAG methods. Specifically, existing benchmarks exhibit two key limitations that inadvertently favor RAG-based approaches: (1) they heavily present tasks focused on tasks that assess the model's retrieval ability for long context (key characteristics being the complexity independent of context length) and (2) even though some tasks indeed see complexity increases with context length, e.g. Variable Tracking (Hsieh et al., 2024), they are often still too simple, and doesn't require model's reasoning ability to solve. We empirically showed that simple RAG techniques easily get perfect scores on the Variable

*Internship done in Carnegie Mellon University.



(a) General Relationship Between Accuracy, Complexity, and Context Length

(b) Average Accuracy of 39 Different Complexity Settings Across Context Length

(c) Modeling Accuracy and Complexity with Logarithm Linear Relationship

Figure 1: (a) presents representative performance of Llama-3.1-70B-Instruct on FACTOR. From the trend, we can see that as N or the complexity increases, the accuracy decreases. Also, as the context length of the question prompt increases, the curve shifts downward. (b) shows the ranking of mainstream long-context LLMs on the average accuracy taken from all 39 different complexity settings N. (c) presents our attempt to model accuracy versus complexity for various models generally. Surprisingly, we observe that the logarithm of FACTOR accuracy linearly correlates nicely with the complexity N.

Tracing tasks. Using the MPnetv2 Song et al. (2020) as the encoder and Llama3.1 8B Instruct Dubey et al. (2024) model as the generator, the system achieves 100% and 98% accuracy on 131k and 1M context length with only 1024 actual context during the retrieval. Experiments show that solving VT is viable through simple backtracking only the occurrence of the query variable without grasping the entire problem setting and states of other variables. We contend that benchmarks hide the unique but impressive reasoning advantages that long-context LLMs can provide over RAG techniques. Therefore, the long-context language model needs a benchmark that can evaluate the LLM ability beyond retrieval.

Therefore, in this paper, we introduce the **FACTOR** (**F**actoring **A**nalysis of **C**omplexity and **T**extual C**O**ntext in **R**easoning) benchmark, designed to systematically evaluate language models' long context complex reasoning ability through carefully curated synthetic tasks that require models' grasp of the entire problem to succeed. Similar to previous benchmarks, FACTOR disentangles task complexity and context length, each can be independently varied. Specifically, the FACTOR benchmark relies on a suite of synthetic task generators by adjusting the following two key knobs for independently controlling task complexity and context length.

- **Number of Variables (Task Complexity)**: Defines the complexity of the reasoning task via the number of interdependent variables. For most evaluations, the number of variables is limited to less than 40.
- **Length of Filler Text (Context Length)**: To independently control the task complexity, we insert the question prompt with text irrelevant to the necessary portion of logic arguments, referring to them as Filler Tex. Filler text lengths are selected from predefined lengths: 0, 4K, 8K, 16K, 32K, 64K, and 128K tokens.

FACTOR consists of two subsets: symbolic and realistic suites. The most striking difference between the two is the problem world setting, where symbolic represents all variables as "Vx" for an integer x, while realistic assigns variables with realistic names and contain real-world variable relationships and hierarchies. In both suites, the model is presented with long chains of variables, and they can only provide correct output when they correctly capture the relationship of all variables that appeared. The rest of the context length is filler text. Computation operators are limited to grade school level similar to GSM8K (Cobbe et al., 2021).

The general trend of FACTOR is shown in Figure 1 (a). We comprehensively evaluate state-of-the-art pre-trained LLMs on FACTOR, where Figure 1 presents model names enumeration. Besides, the aforementioned rag technique only achieves 4.5% accuracy on the realistic tasks of FACTOR for op=5 and 4K, even given 2K context length, far worse than the full attention LLM counterpart (33%). A snippet of our evaluation is shown in Figure 1 (b). Besides, we also perform explainable mathematical modeling of the performance of various models on FACTOR. To our surprise, we observe that the logarithm of accuracy generally correlates with the task complexity *linearly* across all LLMs evaluated, as shown in Figure 1 (c). Through modeling, we obtain more insightful comparisons between different models.

Moreover, we found that two parameters (slope and y-intercept) used in the linear regression possess explainable meanings and be used as quantitative metrics for describing models' abilities and behaviors. For a given context length, the slope, referred to as **Contextual Decay Factor** (CDF), indicates the rate of degradation of the model when solving increasingly longer context. The y-intercept, or **Contextual Decay Offset** (CDO) captures the model's baseline performance at the given context length. We can separately conclude both the model's reasoning ability and the long-context tracking ability from CDF and CDO. Specifically, benefiting from FACTOR design to isolate complexity and context length, we found that the mainstream models generally exhibit the following two patterns.

- **Decreasing CDO with Increasing Context Length**: Indicates that the model struggles with processing long contexts, declining in its baseline performance regardless of task complexity.
- **Increasingly Negative CDF at Longer Contexts**: Suggests that the model's ability to handle increasing complexity is impaired with longer contexts.

Our contribution can be summarized as follows:

- **Revealing a Log-Linear Accuracy Pattern in Long-Context Reasoning**: In Section 4, we show that the relationship between task accuracy and complexity can be modeled using a simple and consistent log-linear model.
- **Identifying Mechanisms Behind Performance Decay**: Also in Section 4, we uncover two primary mechanisms causing the decay of performance in long-context reasoning: the degradation of logical reasoning ability, quantified by the **Complexity Decay Factor (CDF)**, and the decline in baseline performance with longer contexts, represented by the **Contextual Decay Offset (CDO)**.
- **Reproducing Failure Modes Through Fine-Tuning Strategies**: In Section 5, our experiments show that the observed failure modes can be reproduced using different fine-tuning methods—*course learning* (as in Llama models) and *mixed sequence length training* (as in GPT-4o-mini). This highlights the impact of training methodologies on models' abilities to handle complex reasoning over long contexts.
- **Unveiling Limitations via Repeated Sampling**: Further in Section 5, we investigate inference-time strategies like repeated sampling and find that, although they have potential to improve overall performance, inherent biases limit models' abilities to indefinitely extend their reasoning capabilities. The longer the context, the more challenging it becomes to recover performance levels seen with clean context training.

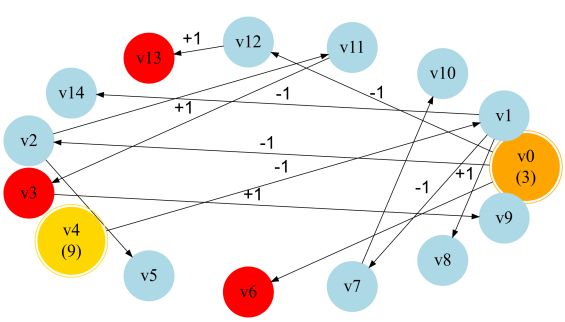
2 Related Work

2.1 Long-context Language Models

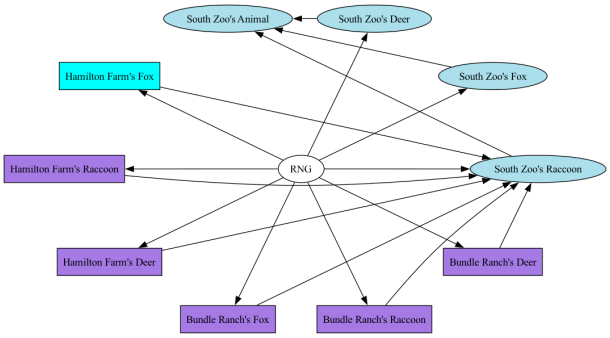
Various works related to the Long-context Language Model have been proposed. Flash attention(Dao et al., 2022), Flash attention2(Dao, 2024), Ring attention(Liu et al., 2023a), and Tree attention(Shyam et al., 2024) significantly reduced the memory footprint and communication overhead for processing long context in engineering level across multiple nodes. Architectural level innovations such as sparse attentions represented by sliding window attention(Beltagy et al., 2020), are also widely used to reduce the overhead caused by the increasing sequence length. New training strategies, such as gradually extending the training context length in the final stages of pretraining have been applied to support a long context window(Dubey et al., 2024).

2.2 Long context benchmarks and tasks

There have been a quite a few works benchmarking long-context language models. Existing comprehensive benchmarks like ∞bench(Zhang et al., 2024) cover realistic tasks including document QA, summary, and synthetic tasks including information retrieval, expression calculation, extending the context length in the benchmark to over 200k tokens. ∞bench(Zhang et al., 2024) does have mathematical reasoning tasks, however the most relevant math.calc part seems to be too difficult for SOTA models to work out. Synthetic tasks often offer more control and are less affected by parametric knowledge in comparison with realistic tasks. One comprehensive synthetic benchmark is RULER(Hsieh et al., 2024), a synthetic benchmark with tasks including retrieval, variable tracking and so on, offering some controls over context length and task complexity. Experiments with various complexities were done, but it does not provide a quantitative analysis of complexity and context length on the correctness of the task, let alone isolate two separate patterns of performance decay. Other benchmarks usually focus on simple retrieval(Github, 2023; Liu et al., 2023b), fact reasoning(Kuratov et al., 2024), the impact of long context on natural language reasoning(Levy et al., 2024) and other real-world knowledge involved tasks.



(a) Symbolic Tasks: Variable names in "Vx" for an integer x. Question: which variable is of value y? (for an integer y) Answer in **red**.



(b) Realistic Tasks: Variable names in realistic items that the pretrained LLM is familiar with. Question: How many **Fox** are there in **Hamilton Farm**? Query variable in color **Cyan**.

Figure 2: Examples of FACTOR task. (a) shows the dependence graph of an example for the symbolic portion. In the example, v4 and v6 are initial conditions given, and arrows represent dependencies. For the symbolic tasks, we ask the model to name all variables with a certain value, effectively asking the model to construct the entire dependency graph. (b) shows the dependency graph of an example for the realistic portion. In the example, light cyan surrounding the bounding box is the actual variable we need, while the rest of the blue variables are intermediate variables needed to find before reaching to the query variable, while the purple variables are redundant variables unnecessary for the minimal calculation towards the answer.

2.3 Synthesized Datasets for long-context reasoning

Synthesized tasks are simple to build and absolutely deterministic, data contamination safe, but highly effective to evaluate the certain aspects of LLM performance. Its use in long-context benchmarks is profound. Needle-in-the-haystack Kamradt (2023), a pioneering long-context synthesized task, now becomes the go-to task for evaluating LLM long-context retrieval ability. On the other hand, LLM reasoning benchmarks also sees recent efforts in synthesized tasks. Mirzadeh et al. (2024) recently proposes to use build synthesized dataset upon GSM8K Cobbe et al. (2021) to study the robustness of LLM reasoning. **Part of our work draws a strong inspiration from a series of works (Ye et al. (2024a), Ye et al. (2024b)) which systematically studies the intricacies of decoder transformers in solving grade-school level problems.** Following their footsteps, we carefully redesign the process of generating the problems so current LLMs are able to solve without training, and together with thoughtful steps in noise addition, we effectively construct an effective reasoning benchmarks to the long-context community.

3 The FACTOR Benchmark

We introduce the **FACTOR** (**F**actoring **A**nalysis of **C**omplexity and **T**extual C**O**ntext in **R**easoning), a fully synthetic benchmark to evaluate the reasoning abilities of language models. Evaluating the reasoning abilities of language models using real-world tasks presents several challenges that synthetic datasets can effectively address. Real-world reasoning tasks often require extensive prior knowledge that models may have memorized from training data, making it difficult to identify whether a model's failure is due to genuine reasoning factors or the retrieval of memorized prior knowledge. Synthetic datasets minimize reliance on external knowledge, ensuring that evaluations focus on the reasoning processes themselves. In natural datasets, it is challenging to systematically control and vary the complexity of tasks. There may be systematic biases for tasks with different levels of difficulty that affect our calibration of difficulty. Synthetic datasets allow for more precise control over task complexity, enabling fine-grained analysis of model performance across levels. This control is crucial for understanding how models handle increasing complexity and for identifying the point at which performance degrades.

Real-world tasks also have varying context lengths, which can introduce changes in performance independent of reasoning ability. Synthetic datasets can be standardized for context length or systematically varied to study their effects on reasoning, effectively separating the effects of task complexity and context length. In this way, researchers can assess how longer contexts affect a model's ability to maintain attention and accurately process relevant information. In addition, there are a limited number of inference tasks available in the real world, especially those of higher complexity and longer duration. Synthetic datasets allow for the creation of as many examples as needed, providing sufficient data points for

a robust statistical evaluation. With this large-scale data generation approach, more reliable conclusions can be drawn about the model's reasoning ability. In addition, real-world contexts may contain extraneous information or stylistic variations that can affect model performance. Synthetic datasets can maintain format consistency and minimize noise, thus ensuring that performance differences are due to reasoning ability rather than extraneous factors. By eliminating extraneous factors, researchers can more accurately attribute performance differences to specific aspects of reasoning.

The main purpose of using synthetic datasets in the FACTOR benchmark is to evaluate reasoning under controlled complexity and to assess the impact of context length. By systematically varying complexity and context length independently, we can determine the individual contribution of each factor to the overall performance degradation. This approach provides detailed quantifiable data that highlights the limitations of specific reasoning, thus guiding researchers to develop models that are better suited to handle complex reasoning tasks.

To accomplish these goals, we generated data in the following manner. We first create sets of variables and establish mathematical relationships between them to form dependency graphs. These variables and relationships are then matched to templates and embedded in contextual paragraphs containing filler text, which adds contextual length but not relevant information. We use specific separators to distinguish the variable relationships from the filler text, ensuring that they are easily recognizable and not obscured by the surrounding content.

Synthesizing data in this way allows us to generate arbitrary numbers of tasks and precisely control the complexity and context length of the tasks. This approach allows us to evaluate the reasoning ability of models in different scenarios with unparalleled accuracy, providing us with a powerful framework to deepen our understanding of how models handle complex reasoning tasks in different contexts. It addresses the limitations of real-world datasets and provides a scalable and repeatable method for evaluating and improving the reasoning ability of language models.

3.1 Symbolic Portion: Task Generation

Tasks are generated by first creating a set of variables $\{v_0, v_1, ..., v_N\}$, where N represents the *task complexity*. Mathematical relationships among these variables (e.g., $v_i = v_j \pm 1$) are then established to form a dependency graph. Consistent values that satisfy all relationships are assigned. These relationships are embedded within filler text, the randomly generated text irrelavant to the logic components, to create contexts of varying lengths, representing different *context lengths*. To distinguish the variable relationships from the filler text, they are enclosed within triple angle brackets `<<<` and `>>>`, and further enclosed within `@` symbols to separate them from the surrounding content. This ensures that the relationships are clearly identifiable and not affected by the filler text.

To avoid falling back to Variable Tracking-like tasks that use backtracking for answer generation, we choose not to allow the model to calculate only the value of certain query variables. Instead, as shown in the example, we ask the model to output all the variables of a query value, which none can be a valid answer. In practice, our approach works well in differentiating between strong and weak reasoning abilities across models. When evaluating the models, we run models with 500 examples for each operation count from 1 to 39.

A Symbolic task of FACTOR benchmark

This is the beginning of the text: @⟨⟨⟨assign $v_1 = v_4 - 1$⟩⟩⟩@@⟨⟨⟨assign $v_0 = v_4 - 1$⟩⟩⟩@@⟨⟨⟨assign $v_3 = v_4 + 1$⟩⟩⟩@@⟨⟨⟨assign $v_2 = 1$⟩⟩⟩@@⟨⟨⟨assign $v_4 = v_2$⟩⟩⟩@ This the end of the text. The text contains relationships between variables enclosed by '⟨⟨⟨' and '⟩⟩⟩'. These relationships are not sequential assignments in a programming language. They are independent mathematical equations that are all true simultaneously. Using only these relationships, determine what variable(s), if any, are equal to 2. Show your step-by-step reasoning and calculations, and then conclude your final answer in a sentence.
Answer: `v3`

3.2 Realistic Portion: Task Generation

For this portion, we tried to put questions into the real-world perspective. Importantly, however, similar to the symbolic datasets, we strictly control the difficulty level of each problem by the number of binary operations (two variables each) needed to perform to get to the final answer. Controling the difficulty level and the context length brings us solid playground to study LLM reasoning limitations. Specifically, we ensure two types of dependencies following Ye et al. (2024a): direct and hierarchical, where direct dependency is similar to the dependency in Symbolic tasks, where one variable is the sum or a scale multiple of other variables. Hierarchical dependency describes the type of relationship of category names and the instances, for e.g. "Animals" and "Fox" or "Racoon". An example is shown below.

A Realistic task of FACTOR benchmark

Problem: The number of each South Zoo's Fox is 2. The number of each Hamilton Farm's Fox is the sum of South Zoo's Animal, South Zoo's Fox, and South Zoo's Deer. The number of each South Zoo's Raccoon is 0 times each South Zoo's Fox. The number of each South Zoo's Deer is each South Zoo's Fox. The number of each Hamilton Farm's Deer is 16 times the sum of Hamilton Farm's Raccoon, Bundle Ranch's Fox, and Bundle Ranch's Deer. The number of each Bundle Ranch's Deer is 13. The number of each Bundle Ranch's Fox is 11 times each Hamilton Farm's Raccoon. The number of each Bundle Ranch's Raccoon is 2 more than each South Zoo's Deer. The number of each Hamilton Farm's Raccoon is 6. Question: How many Fox does Hamilton Farm have? **Solution (Optimal, not appeared in evaluation)**: Define South Zoo's Fox as d; so d = 2. Define South Zoo's Deer as G; so G = d = 2. Define South Zoo's Raccoon as J; R = d = 2; so J = 0 * R = 0 * 2 = 0. Define South Zoo's Animal as v; M = d + G = 2 + 2 = 4; so v = M + J = 4 + 0 = 4. Define Hamilton Farm's Fox as q; u = v + d = 4 + 2 = 6; so q = u + G = 6 + 2 = 8. Answer: 8.

In the example shown above, South Zoo's Fox, Deer, and Raccoon are instances of South Zoo's Animal. When computing for South Zoo's Animal, all three instances need to be added up. Generation of problem process can be broken down into several steps. First, we generate a structure graph enlisting all the concepts and hierarchies appeared in the paper. Second, we generate solution with the dependencies of essential variables in the problem. Lastly, we fill up text with some close variables unnecessary to solving the problems. More details are in the appendix. In total, we have 19 different difficulty settings (from 2 to 20), where for each setting, FACTOR has 1000 test examples.

3.3 Noise Addition

Adding noise is essential to enriching the length of the problem in the long context regime. For FACTOR, we mainly explore two different directions of making the noise. First, we explore close noise, where the newly added filler text is closely related to the existing text but is not essential to solving the problem. Another noise addition method is the general noise, where general language is added to bloat the context. Specifically for Realistic tasks, we ask the ChatGPT OpenAI et al. (2024) to generate text that involves concepts mentioned in the essential path, but this text added is forced to not contain new relationship between mentioned concepts to avoid polluting the answer.

3.4 Evaluation Metrics

Besides just getting the accuracy, we also care about modeling the relationships between the accuracy and the complexity, thanks to the FACTOR design to isolate complexity and context length. To our surprise, the relationship can generally be captured by a consistently simple log-linear model. Also, the slope and the y-intercept of the linear model are explainable and present separate measures of complexity and context length independently.

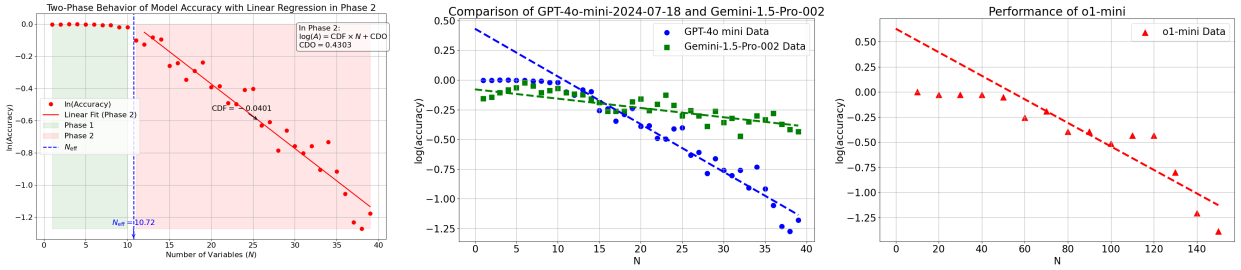
Two-Phase Accuracy Behavior - Models exhibit a characteristic two-phase behavior in accuracy as the number of variables N increases (see Figure 3). **Phase 1**: For small values of N, models maintain near-perfect accuracy, effectively handling tasks with low complexity. **Phase 2**: Beyond a critical complexity threshold (N_{eff}), accuracy declines exponentially with increasing N, indicating rapid degradation in performance for more complex tasks. This pattern suggests a limit to the task complexity that models can handle before performance significantly deteriorates.

Evaluation Metrics Definition - In Phase 2, accuracy A decreases exponentially with increasing N. By taking the natural logarithm of accuracy, we linearize this decay:

$$\log(A) = \text{CDF} \times N + \text{CDO}$$

Where **Complexity Decay Factor (CDF)** is the negative slope of the line ($\text{CDF} < 0$), representing the rate at which accuracy decays with increasing task complexity; **Contextual Decay Offset (CDO)** is the intercept of the line, capturing the baseline performance level influenced by context length.

From these parameters, we define the **Effective Complexity** N_{eff}, indicating the maximum task complexity the model can handle before significant performance degradation. It is calculated as the value of N when the extrapolated $\log(A) = 0$ (i.e., when accuracy $A = 1$):$N_{\text{eff}} = -\frac{\text{CDO}}{\text{CDF}}$. However, if the CDO is negative, the extrapolated N_{eff} becomes negative, which is not meaningful since we cannot have a negative number of variables. In such cases, the two-phase behavior is not observed—the model's accuracy declines from the outset without an initial phase of high accuracy. Therefore, for negative CDO values, we focus on the exponential decay characterized by the CDF and CDO. A more detailed description of data processing can be found in the Appendix B.



(a) (b) (c)

Figure 3: (a) Illustration of the two-phase accuracy behavior as a function of task complexity N. (b) presents the two different accuracy patterns between GPT-4o-mini and Gemini-1.5-Pro. GPT-4o-mini outperforms Gemini-1.5-Pro in low complexity tasks, while Gemini-1.5-Pro is more capable of dealing with more complex tasks. (c) shows the amazing performance of o1-mini: its has a higher N_{eff} than any other models tested and a decent CDF. This illustrates the effectiveness of the inference time strategy on reasoning.

4 Evaluation on Pretrained Models

We evaluate a range of pre-trained language models using the FACTOR benchmark to understand how they handle increasing task complexity and long context lengths, identifying their failure modes. The evaluation is structured into two parts: (1) **Benchmarking with Zero Filler Context**: Assessing models' abilities to handle task complexity independently of context length. (2) **Benchmarking with Long Contexts**: Analyzing how models that perform well without filler context degrade when exposed to long contexts.

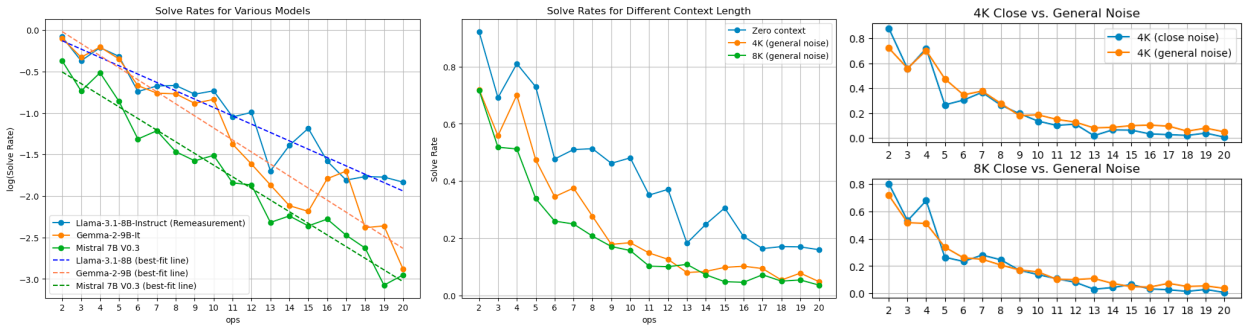
4.1 Symbolic Tasks: Benchmarking with Zero Filler Context

We evaluated 15 models, comparing their FACTOR benchmark metrics with ELO scores from the LMSYS Chatbot Arena(Chiang et al., 2024) on hard prompts with style control (see Table 1).

Table 1: Metrics with Zero Filler Context are listed in this table. We observe a strong correlation between model capability and these metrics, together with different accuracy patterns of different models.

Index	Model	CDF	CDO	N_{eff}	LMSYS ELO
1	o1-mini	-0.0117	0.6303	53.87	1294
2	Gemini-1.5-Pro-002	-0.0081	-0.0696	-8.59	—
3	Claude-3-5-Sonnet-20240620	-0.0187	-0.1447	-7.74	1268
4	GPT-4o-2024-05-13	-0.0220	0.2298	10.45	1251
5	GPT-4o-2024-08-06	-0.0205	0.0899	4.39	1237
6	Gemini-1.5-Flash-002	-0.0244	-0.0180	-0.74	—
7	Mistral-Large-Instruct-2407	-0.0279	0.2100	7.53	1231
8	GPT-4-Turbo-2024-04-09	-0.0378	0.2514	6.65	1226
9	Qwen2.5-72B-Instruct	-0.0265	0.2056	7.76	1223
10	GPT-4o-mini-2024-07-18	-0.0401	0.4303	10.73	1219
11	Llama-3.1-70B-Instruct	-0.0302	-0.0481	-1.59	1187
12	Qwen2-72B-Instruct	-0.0467	0.0123	0.26	1178
13	Claude-3-Haiku-20240307	-0.0471	-0.0848	-1.80	1173
14	Mistral-Nemo-Instruct-2407	-0.0608	0.0735	1.21	—
15	Llama-3.1-8B-Instruct	-0.0694	-0.4615	-6.65	1132

The evaluation reveals key insights into model performance concerning task complexity: Models with similar LMSYS ELO scores exhibit different behaviors as task complexity N increases(see Figure 3): models like **GPT-4o-mini-2024-07-18** (CDO $= 0.4303$, CDF $= -0.0401$) excel on simple tasks (high CDO) but degrade rapidly with increasing complexity (more negative CDF); Models like **Gemini-1.5-Pro-002**(Team et al., 2024) (CDO $= -0.0696$, CDF $=$



(a) Comparing between Models in the Zero Context Length

(b) **Llama-3.1-8B-Instruct** on context zero length, 4K, and 8K

(c) **Llama-3.1-8B-Instruct** between Close and General Noise

Figure 4: Realistic Portion Results. (a) shows that the log-linear relationship also holds true for more complex datasets as well across three different LLMs. (b) shows that the performance generally degrades as the context length increases. (c) shows that for two different noise addition methods, which shows that the two noise aren't much different to LLMs across both 4K and 8K context length.

-0.0081) struggle with simple tasks (low CDO) but handle complex tasks better as they degrade more slowly (less negative CDF). **o1-mini** achieves both high CDO (0.6303) and a less negative CDF (-0.0117), resulting in a high N_{eff} (53.87). This indicates that inference-time strategies can significantly enhance performance across task complexities.(see Figure 3)

Models may perform similarly on general benchmarks but differ on tasks requiring complex reasoning. Selecting models for applications should consider CDO and CDF to match the complexity needs. The FACTOR benchmark highlights the necessity to evaluate models on complexity handling rather than solely on overall scores. By focusing on CDO and CDF, we are able to model distinct failure modes among models as task complexity increases. Understanding these metrics aids in selecting appropriate models for specific tasks and emphasizes the importance of specialized benchmarks like FACTOR.

4.2 Symbolic Tasks: Benchmarking with Long Contexts

We analyze how models degrade when exposed to long contexts by examining the CDF and CDO metrics across different context lengths. Models were tested with varying amounts of filler context, extending up to their maximum context lengths (4K to 128K tokens). (see Table 23)

Table 2: Complexity Decay Factor (CDF) for Models at Different Context Lengths. (Values scaled by 10^2)

Model	4K	8K	16K	32K	64K	128K
Mistral-Large-Instruct-2407	-3.26	-3.88	-4.04	-4.41	-15.19	—
Qwen2.5-72B-Instruct	-2.82	-3.16	-3.27	-3.08	—	—
GPT-4o-mini-2024-07-18	-4.93	-4.90	-4.95	-5.13	-5.01	-6.02
Llama-3.1-70B-Instruct	-3.88	-4.27	-4.82	-5.55	—	—
Qwen2-72B-Instruct	-4.66	-5.22	-5.56	-6.10	—	—
Mistral-Nemo-Instruct-2407	-5.15	-6.58	-29.43	-4.35	-3.58	-3.64
Llama-3.1-8B-Instruct	-7.24	-7.12	-8.48	-9.98	-10.19	—

The evaluation reveals distinct failure modes when models are exposed to longer contexts.(all CDF values are scaled by 10^2, see Figure 5) **(1) Stable CDO, Degrading CDF (e.g., Llama Series):** Models like **Llama-3.1-70B-Instruct** maintain relatively stable CDO across context lengths (from -0.014 at 4K to -0.104 at 32K), indicating consistent baseline performance. However, their CDF becomes more negative as context length increases (from -3.88 to -5.55), signifying increased difficulty with task complexity in longer contexts. **2. Stable CDF, Degrading CDO (e.g., GPT-4o-mini-2024-07-18):** models like **GPT-4o-mini-2024-07-18** maintains a consistent CDF across context lengths (approximately -4.93 to -5.13 up to 64K tokens), suggesting stable handling of task complexity. However, its CDO decreases steadily (from -0.020 at 4K to -0.711 at 64K), indicating declining baseline performance with longer

Table 3: Contextual Decay Offset (CDO) for Models at Different Context Lengths.

Model	4K	8K	16K	32K	64K	128K
Mistral-Large-Instruct-2407	0.085	−0.027	−0.153	−0.497	−0.510	—
Qwen2.5-72B-Instruct	0.055	−0.021	−0.088	−0.281	—	—
GPT-4o-mini-2024-07-18	−0.020	−0.212	−0.312	−0.494	−0.711	−0.700
Llama-3.1-70B-Instruct	−0.014	−0.065	−0.054	−0.104	—	—
Qwen2-72B-Instruct	−0.172	−0.210	−0.197	−0.296	—	—
Mistral-Nemo-Instruct-2407	−0.459	−0.609	−0.482	−1.154	−1.287	−1.287
Llama-3.1-8B-Instruct	−0.591	−0.679	−0.686	−0.608	−0.621	—

contexts. **3. Degrading CDO and CDF:** models like **Mistral-Large-Instruct-2407** show degradation in both CDO and CDF as context length increases, facing compounded difficulties with baseline performance and task complexity.

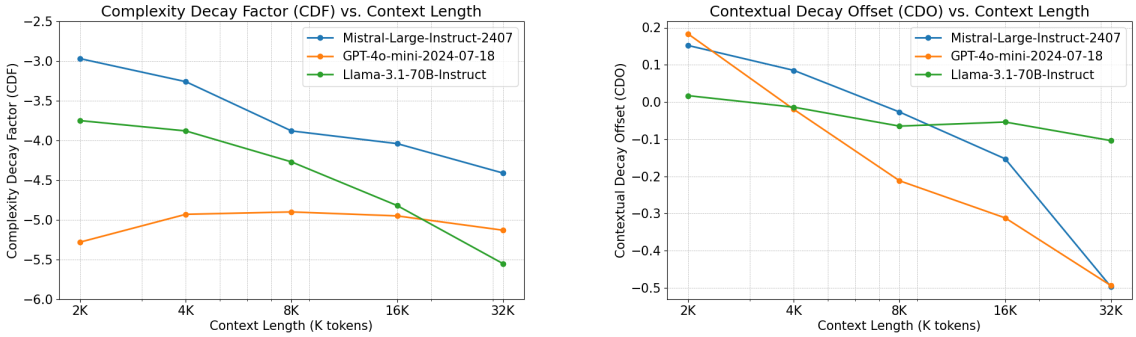


Figure 5: CDF/CDO of three models at different context length. Llama-3.1-70B-Instruct maintains a relatively stable CDO across context lengths, while its CDF decreases over context length. GPT-4o-mini, on the contrary, maintains a relatively stable CDF while its CDO decreases over context length. Mistral-Large-Instruct is observed to have both metrics decreasing over context length.

Models exhibit different failure modes with long contexts: (1) **Stable CDO, Degrading CDF**: Models maintain baseline performance but increasingly struggle with task complexity as context lengthens; (2) **Stable CDF, Degrading CDO**: Models handle task complexity consistently but suffer declining overall performance with longer contexts; (3) **Degrading CDO and CDF**: Models face difficulties with both baseline performance and task complexity. Understanding these patterns is crucial for selecting appropriate models based on task requirements and highlights the importance of enhancing model robustness in processing long contexts and complex tasks.

4.3 Realistic Tasks: Benchmarking with more complex real-world examples

We present zero-context, 4K, and 8K three different settings under the realistic portion of FACTOR. We first look at the realistic portion of FACTOR. Summarized in Figure 4, surprisingly, we found that overall accuracy versus operations follows the exponential decay as in the symbolic portion, shown in (a). The performance of mainstream LLM seems to also correlate log-linearly with the difficulty of the tasks. In (b), we can see that as we increase the context length, the performance of the long-context LLM starts to generally decrease, signifying from the overall sinking curve. Also, in (c), we compare between different length extending methods, generally both adding close or general noise leads to overall similar trend, but difference of the two types of noises do present certain number of operations.

5 Factors Affecting the Metrics

5.1 Pretraining Strategies

We investigate how different pretraining strategies influence the Complexity Decay Factor (CDF), Contextual Decay Offset (CDO), and especially the Effective Complexity (N_{eff}) in the FACTOR benchmark. By training models from scratch using various methodologies, we aim to understand their effects on the models' abilities to handle task

complexity and context length. See Appendix C for training settings and Appendix D for training data distribution. Models were trained using the following strategies:

(1) **Baseline**: Trained directly on the FACTOR benchmark training set (regenerated to avoid data leakage), mirroring the benchmark's organization. The max context length is limited to 1000 tokens. (2) **Naive Packed Training** (*Packed*): Sequences in a mini-batch are concatenated to form longer training inputs. (3) **Question-Masked Training** (*Masked*): The question portion (payload) is masked during loss computation. (4) **Clean Context Training** (*Clean*): The model is exposed only to the payloads, without filler text. (5) **Packed Training with Diagonal Attention Mask** (*Diagonal*): Similar to Packed, but with diagonal attention masks to prevent cross-sequence attention; position IDs are retained.

We evaluated the models on the FACTOR benchmark with zero filler context to assess their baseline performance. The results are presented in Table 4.

Table 4: Effects of Pretraining Strategies on Model Performance (Zero Filler Context).

Model	**CDF**	**CDO**	N_{eff}
Baseline	−0.3292	8.4685	25.72
Packed	−0.1339	2.6381	19.70
Masked	−0.3278	8.5196	25.99
Clean	−0.2672	7.0388	26.34
Diagonal	−0.3780	10.0887	26.69

The Effective Complexity (N_{eff}) indicates the maximum task complexity the model can handle before performance significantly degrades. Higher N_{eff} values reflect better performance over a broader range of complexities.

From Table 4, we observe: (1) The **Baseline**, **Masked**, **Clean**, and **Diagonal** models achieve similar N_{eff} values around 26, indicating they handle task complexities up to $N \approx 26$ effectively. (2) The **Packed** model has a lower N_{eff} of 19.70, suggesting worse performance over the range of task complexities. Naive packed training leads to poorer handling of complex tasks compared to other strategies. Despite a more negative CDF (−0.3780), the **Diagonal** model achieves the highest N_{eff} (26.69) and a higher CDO (10.0887). This means the model starts with higher baseline accuracy (due to higher CDO) and maintains decent performance for N between 25 and 33, extending the effective complexity range. **Packed Training** reduces N_{eff}, leading to worse performance across task complexities. **Diagonal Training**, although it has a more negative CDF, extends N_{eff}, improving performance for some higher N values. Therefore, while naive packed training negatively impacts the model's ability to handle complex tasks, the Diagonal strategy enhances performance at higher complexities, evidencing its benefit in extending the range over which the model maintains accuracy.

5.2 Post-Training Strategies

We examine how different fine-tuning strategies affect model performance on the FACTOR benchmark, particularly the CDF and CDO metrics, with a maximum context length of 16K tokens. Notably, although the maximum training length is 16K, the majority of sequences in the training distribution are less than 8K. The models are trained based on the checkpoing of the Baseline model in the last section. We evaluate three approaches: (1) **Gradual Sequence Length Increase (*Course*)**: Starting with shorter sequences and progressively increasing the length during fine-tuning. (2) **Mixed-Length Sequence Training (*Mixed*)**: Training on sequences of varied lengths simultaneously. (3) **Direct Long Sequence Training (*Full*)**: Fine-tuning directly on the maximum sequence length without gradual adaptation.

Table 5 and 6 present the CDF and CDO metrics at different context lengths.

Table 5: CDF at Different Context Lengths. (Values scaled by 10^{-2})

Model	**0**	**1K**	**2K**	**4K**	**8K**	**16K**
Course	−7.91	−6.44	−8.28	−8.31	−10.29	−14.56
Mixed	−8.43	−6.66	−7.79	−6.86	−8.11	−11.25
Full	−12.48	−9.34	−17.28	−12.42	−5.76	−18.82

Table 6: CDO at Different Context Lengths.

Model	0	1K	2K	4K	8K	16K
Course	0.4775	0.3539	0.4916	0.3383	0.3392	0.1038
Mixed	0.3227	0.1351	0.2076	0.0080	−0.0922	−0.3469
Full	−0.3222	−0.4789	−0.2658	−0.3260	−0.7427	−0.2435

The fine-tuning strategies exhibit distinct effects on CDF and CDO:

Despite training up to 16K contexts, the *Course* and *Mixed* strategies significantly outperform the *Full* strategy, likely due to the training distribution containing mostly shorter sequences. Sudden exposure to full-length sequences in the *Full* strategy causes a distribution shift, leading to degraded performance.

Course strategy results in models with stable CDO and degrading CDF, resembling Llama series models. This indicates that the models maintain baseline performance but struggle more with task complexity as context length increases. **Mixed** strategy yields models with stable CDF and degrading CDO, similar to some pretrained models that handle task complexity consistently but whose baseline performance declines with longer contexts, resembling models like gpt-4o-mini. **Full** strategy leads to poor performance in both metrics, due to abrupt changes in data distribution. Gradual adaptation to longer sequences during fine-tuning helps models cope better with increasing context lengths and task complexities, mitigating failure modes observed in pretrained models.

5.3 Inference-Time Strategies

We analyze how increasing computational efforts during inference affects model performance, particularly focusing on the accuracy and the effective complexity N_{eff}. To investigate the impact of increased computational effort during inference, we employ **repeated sampling**. This approach measures the probability of correctly solving at least one instance out of t samples. We conduct experiments on models trained with the Gradual Increase strategy at different filler context lengths, as well as on the model pre-trained with a clean context (zero filler context).

Table 7 summarizes the linear regression results for different filler context lengths. The regression equation is given by:

$$\log(N_{\text{eff}}^{\text{clean}} - N_{\text{eff}}^{\text{model}}) = k\log(t) + b$$

Table 7: Linear Regression Results for Different Filler Context Length.

Filler Context Length	k (Slope)	b (Intercept)	R-squared
0	-0.358287	2.865435	0.992356
1000	-0.333536	2.670190	0.987057
2000	-0.376972	2.865891	0.982553
4000	-0.261895	2.999738	0.995347
8000	-0.245732	3.242366	0.992648
16000	-0.150702	3.524461	0.998234

Figure 6 illustrates the relationship between N_{eff} and the number of tries t.

Key observations from the results are: (1) For models fine-tuned with gradually increasing context lengths, increasing the number of samples t leads to a slow improvement in N_{eff}, but it remains below the N_{eff} of the clean context model. (2) **Effect of Context Length**: As the filler context length increases, the value of k becomes less negative, indicating a decrease in the rate of change. Simultaneously, the intercept b increases. (3) **Interpretation**: While increasing computational efforts during inference provides marginal benefits, it does not fully overcome the challenges posed by high task complexity and long contexts.

6 Conclusion

In this paper, we introduced the FACTOR benchmark, a novel framework designed to systematically evaluate the complex reasoning abilities of large language models (LLMs) over long contexts. A key innovation of our work is the modeling of performance over task complexity, moving beyond traditional scalar evaluation metrics to capture

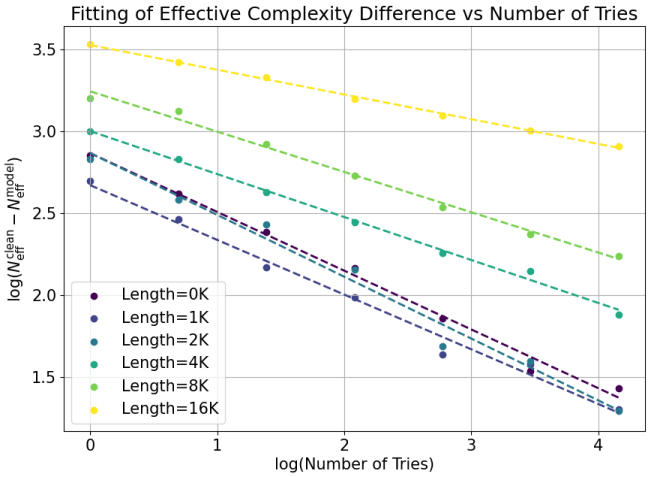


Figure 6: The relationship between N_{eff} and the number of tries t. We can see strong linear correlation between $\log(N_{\text{eff}}^{\text{clean}} - N_{\text{eff}}^{\text{model}})$ and $\log(t)$. This implies a slow improvement in N_{eff} with the increase of number of samples, and it does not fully overcome the challenges posed by high task complexity and long contexts.

the dynamic two-phase behavior in accuracy as complexity increases. Specifically, we observed that LLMs maintain high accuracy up to a certain complexity threshold, after which performance declines exponentially. By characterizing this behavior through the **Complexity Decay Factor (CDF)** and the **Contextual Decay Offset (CDO)**, we provided a nuanced understanding of how task complexity and context length independently affect model performance.

Our analysis revealed that these metrics not only quantify the degradation of logical reasoning ability and baseline accuracy but also highlight the limitations of current LLMs in handling complex reasoning tasks over extended contexts. Furthermore, we demonstrated that different fine-tuning strategies can reproduce these failure modes, emphasizing the significant impact of training methodologies on model capabilities. By modeling the two-phase behavior in accuracy rather than relying on a single performance score, the FACTOR benchmark offers a more detailed and insightful evaluation framework. This approach allows researchers to identify specific areas for improvement in LLMs and guides future developments in creating more robust language models capable of complex reasoning over long contexts.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., and Stoica, I. (2024). Chatbot arena: An open platform for evaluating llms by human preference.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dao, T. (2024). FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. (2022). FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes,

D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Tan, X. E., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Grattafiori, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Vaughan, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Franco, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Wyatt, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Ozgenel, F., Caggioni, F., Guzmán, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Thattai, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Damlaj, I., Molybog, I., Tufanov, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Prasad, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Huang, K., Chawla, K., Lakhotia, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Tsimpoukelli, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Laptev, N. P., Dong, N., Zhang, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Li, R., Hogan, R., Battey, R., Wang, R., Maheswari, R., Howes, R., Rinott, R., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Kohler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Albiero, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wang, X., Wu, X., Wang, X., Xia, X., Wu, X., Gao, X., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Hao, Y., Qian, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., and Zhao, Z. (2024). The llama 3 herd of models.

Github (2023). Needle in a haystack - pressure testing llms.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., and Ginsburg, B. (2024). Ruler: What's the

real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Kamradt, G. (2023). Needle in a haystack - pressure testing llms.

Kuratov, Y., Bulatov, A., Anokhin, P., Rodkin, I., Sorokin, D., Sorokin, A., and Burtsev, M. (2024). Babilong: Testing the limits of llms with long context reasoning-in-a-haystack.

Levy, M., Jacoby, A., and Goldberg, Y. (2024). Same task, more tokens: the impact of input length on the reasoning performance of large language models.

Li, Z., Li, C., Zhang, M., Mei, Q., and Bendersky, M. (2024). Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach.

Liu, H., Zaharia, M., and Abbeel, P. (2023a). Ring attention with blockwise transformers for near-infinite context.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2023b). Lost in the middle: How language models use long contexts.

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., and Farajtabar, M. (2024). Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). Gpt-4 technical report.

Shyam, V., Pilault, J., Shepperd, E., Anthony, Q., and Millidge, B. (2024). Tree attention: Topology-aware decoding for long-context attention on gpu clusters.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding.

Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., Mariooryad, S., Ding, Y., Geng, X., Alcober, F., Frostig, R., Omernick, M., Walker, L., Paduraru, C., Sorokin, C., Tacchetti, A., Gaffney, C., Daruki, S., Sercinoglu, O., Gleicher, Z., Love, J., Voigtlaender, P., Jain, R., Surita, G., Mohamed, K., Blevins, R., Ahn, J., Zhu, T., Kawintiranon, K., Firat, O., Gu, Y., Zhang, Y., Rahtz, M., Faruqui, M., Clay, N., Gilmer, J., Co-Reyes, J., Penchev, I., Zhu, R., Morioka, N., Hui, K., Haridasan, K., Campos, V., Mahdieh, M., Guo, M., Hassan, S., Kilgour, K., Vezer, A., Cheng, H.-T., de Liedekerke, R., Goyal, S., Barham, P., Strouse, D., Noury, S., Adler, J., Sundararajan, M., Vikram, S., Lepikhin, D., Paganini, M., Garcia, X., Yang, F., Valter, D., Trebacz, M., Vodrahalli, K., Asawaroengchai, C., Ring, R., Kalb, N., Soares, L. B., Brahma, S., Steiner, D., Yu, T., Mentzer, F., He, A., Gonzalez, L., Xu, B., Kaufman, R. L., Shafey, L. E., Oh, J., Hennigan, T., van den Driessche, G., Odoom, S., Lucic, M., Roelofs, B., Lall, S., Marathe, A., Chan, B., Ontanon, S., He, L., Teplyashin, D., Lai, J., Crone, P., Damoc, B., Ho, L., Riedel, S., Lenc, K., Yeh, C.-K., Chowdhery, A., Xu, Y., Kazemi, M., Amid, E., Petrushkina, A., Swersky, K., Khodaei, A., Chen, G., Larkin, C., Pinto, M., Yan, G., Badia, A. P., Patil, P., Hansen, S., Orr, D., Arnold, S. M. R., Grimstad, J., Dai, A., Douglas, S., Sinha, R., Yadav, V., Chen, X., Gribovskaya, E., Austin, J., Zhao, J., Patel, K., Komarek, P., Austin, S., Borgeaud, S., Friso, L., Goyal, A., Caine, B., Cao, K., Chung, D.-W., Lamm, M., Barth-Maron, G., Kagohara, T., Olszewska, K., Chen, M., Shivakumar, K., Agarwal, R., Godhia, H., Rajwar, R., Snaider, J., Dotiwalla, X., Liu, Y., Barua, A., Ungureanu, V., Zhang, Y., Batsaikhan, B.-O., Wirth, M., Qin, J., Danihelka, I., Doshi, T., Chadwick, M., Chen, J., Jain, S., Le, Q., Kar, A., Gurumurthy, M., Li, C., Sang, R., Liu, F., Lamprou, L., Munoz, R., Lintz, N., Mehta, H., Howard, H., Reynolds, M., Aroyo, L., Wang, Q., Blanco, L., Cassirer, A., Griffith, J., Das, D., Lee, S., Sygnowski, J., Fisher, Z., Besley, J., Powell, R., Ahmed, Z., Paulus, D., Reitter, D., Borsos, Z., Joshi, R., Pope, A., Hand, S., Selo, V., Jain, V., Sethi, N., Goel, M., Makino, T., May, R., Yang, Z., Schalkwyk, J., Butterfield, C., Hauth, A., Goldin, A., Hawkins, W., Senter, E., Brin, S., Woodman, O., Ritter, M., Noland, E., Giang, M., Bolina, V., Lee, L., Blyth, T., Mackinnon, I., Reid, M., Sarvana, O., Silver, D., Chen, A., Wang, L., Maggiore, L., Chang, O., Attaluri, N., Thornton, G., Chiu, C.-C., Bunyan, O., Levine, N., Chung, T., Eltyshev, E., Si, X., Lillicrap, T., Brady, D., Aggarwal, V., Wu, B., Xu, Y., McIlroy, R., Badola, K., Sandhu, P., Moreira, E., Stokowiec, W., Hemsley, R., Li, D., Tudor, A., Shyam, P., Rahimtoroghi, E., Haykal, S., Sprechmann, P., Zhou, X., Mincu, D., Li, Y., Addanki, R., Krishna, K., Wu, X., Frechette, A., Eyal, M., Dafoe, A., Lacey, D., Whang, J., Avrahami, T., Zhang, Y., Taropa, E., Lin, H., Toyama, D., Rutherford, E., Sano, M., Choe, H., Tomala, A., Safranek-Shrader, C., Kassner, N., Pajarskas, M., Harvey, M., Sechrist, S., Fortunato, M., Lyu, C., Elsayed, G., Kuang, C., Lottes, J., Chu, E., Jia, C., Chen, C.-W., Humphreys, P., Baumli, K., Tao, C., Samuel, R., dos Santos, C. N., Andreassen, A., Rakićević, N., Grewe, D., Kumar, A., Winkler, S., Caton, J., Brock, A., Dalmia, S., Sheahan, H., Barr, I., Miao, Y., Natsev, P., Devlin, J., Behbahani, F., Prost, F., Sun, Y., Myaskovsky, A., Pillai, T. S., Hurt, D., Lazaridou, A., Xiong, X., Zheng, C., Pardo, F., Li, X., Horgan, D., Stanton, J., Ambar, M., Xia, F., Lince, A., Wang, M., Mustafa, B., Webson, A., Lee, H., Anil, R., Wicke, M., Dozat, T., Sinha, A., Piqueras, E., Dabir, E., Upadhyay, S., Boral, A., Hendricks, L. A., Fry, C., Djolonga, J., Su, Y., Walker, J., Labanowski, J., Huang, R., Misra, V., Chen, J., Skerry-Ryan, R., Singh, A., Rijhwani, S., Yu, D., Castro-Ros, A., Changpinyo, B., Datta, R., Bagri, S., Hrafnkelsson, A. M., Maggioni, M., Zheng, D., Sulsky, Y., Hou, S., Paine, T. L., Yang, A., Riesa, J., Rogozinska, D., Marcus, D., Badawy, D. E., Zhang, Q., Wang, L., Miller, H., Greer, J., Sjos, L. L., Nova, A., Zen, H., Chaabouni, R., Rosca, M., Jiang, J., Chen, C., Liu, R., Sainath, T., Krikun, M., Polozov, A., Lespiau, J.-B., Newlan, J., Cankara, Z., Kwak, S., Xu, Y., Chen, P., Coenen, A., Meyer, C., Tsihlas, K., Ma, A., Gottweis, J., Xing, J., Gu, C., Miao, J., Frank, C., Cankara, Z., Ganapathy, S., Dasgupta, I., Hughes-Fitt, S., Chen, H., Reid, D., Rong, K., Fan, H., van Amersfoort, J., Zhuang, V., Cohen, A., Gu, S. S., Mohananey, A., Ilic, A., Tobin, T., Wieting, J., Bortsova, A., Thacker, P., Wang, E., Caveness, E., Chiu, J., Sezener, E., Kaskasoli, A., Baker, S., Millican, K., Elhawaty, M., Aisopos, K., Lebsack, C., Byrd, N., Dai, H., Jia, W., Wiethoff, M., Davoodi, E., Weston, A., Yagati, L., Ahuja, A., Gao, I., Pundak, G., Zhang, S., Azzam, M., Sim, K. C., Caelles, S., Keeling, J., Sharma, A., Swing, A., Li, Y., Liu, C., Bostock, C. G., Bansal, Y., Nado, Z., Anand, A., Lipschultz, J., Karmarkar, A., Proleev, L., Ittycheriah, A., Yeganeh, S. H., Polovets, G., Faust, A., Sun, J., Rrustemi, A., Li, P., Shivanna, R., Liu, J., Welty, C., Lebron, F., Baddepudi, A., Krause, S., Parisotto, E., Soricut, R., Xu, Z., Bloxwich, D., Johnson, M., Neyshabur, B., Mao-Jones, J., Wang, R., Ramasesh, V., Abbas, Z., Guez, A., Segal, C., Nguyen, D. D., Svensson, J., Hou, L., York, S., Milan, K., Bridgers, S., Gworek, W., Tagliasacchi, M., Lee-Thorp, J., Chang, M., Guseynov, A., Hartman, A. J., Kwong, M., Zhao, R., Kashem, S., Cole, E., Miech, A., Tanburn, R., Phuong, M., Pavetic, F., Cevey, S., Comanescu, R., Ives, R., Yang, S., Du, C., Li, B., Zhang, Z., Iinuma, M., Hu, C. H., Roy, A., Bijwadia, S., Zhu, Z., Martins, D., Saputro, R., Gergely, A., Zheng, S., Jia, D., Antonoglou, I., Sadovsky, A., Gu, S., Bi, Y., Andreev, A., Samangooei, S., Khan, M., Kocisky, T., Filos, A., Kumar, C., Bishop, C., Yu, A., Hodkinson, S., Mittal, S., Shah, P., Moufarek, A., Cheng, Y., Bloniarz, A., Lee, J., Pejman, P., Michel, P., Spencer,

S., Feinberg, V., Xiong, X., Savinov, N., Smith, C., Shakeri, S., Tran, D., Chesus, M., Bohnet, B., Tucker, G., von Glehn, T., Muir, C., Mao, Y., Kazawa, H., Slone, A., Soparkar, K., Shrivastava, D., Cobon-Kerr, J., Sharman, M., Pavagadhi, J., Araya, C., Misiunas, K., Ghelani, N., Laskin, M., Barker, D., Li, Q., Briukhov, A., Houlsby, N., Glaese, M., Lakshminarayanan, B., Schucher, N., Tang, Y., Collins, E., Lim, H., Feng, F., Recasens, A., Lai, G., Magni, A., Cao, N. D., Siddhant, A., Ashwood, Z., Orbay, J., Dehghani, M., Brennan, J., He, Y., Xu, K., Gao, Y., Saroufim, C., Molloy, J., Wu, X., Arnold, S., Chang, S., Schrittwieser, J., Buchatskaya, E., Radpour, S., Polacek, M., Giordano, S., Bapna, A., Tokumine, S., Hellendoorn, V., Sottiaux, T., Cogan, S., Severyn, A., Saleh, M., Thakoor, S., Shefey, L., Qiao, S., Gaba, M., yiin Chang, S., Swanson, C., Zhang, B., Lee, B., Rubenstein, P. K., Song, G., Kwiatkowski, T., Koop, A., Kannan, A., Kao, D., Schuh, P., Stjerngren, A., Ghiasi, G., Gibson, G., Vilnis, L., Yuan, Y., Ferreira, F. T., Kamath, A., Klimenko, T., Franko, K., Xiao, K., Bhattacharya, I., Patel, M., Wang, R., Morris, A., Strudel, R., Sharma, V., Choy, P., Hashemi, S. H., Landon, J., Finkelstein, M., Jhakra, P., Frye, J., Barnes, M., Mauger, M., Daun, D., Baatarsukh, K., Tung, M., Farhan, W., Michalewski, H., Viola, F., de Chaumont Quitry, F., Lan, C. L., Hudson, T., Wang, Q., Fischer, F., Zheng, I., White, E., Dragan, A., baptiste Alayrac, J., Ni, E., Pritzel, A., Iwanicki, A., Isard, M., Bulanova, A., Zilka, L., Dyer, E., Sachan, D., Srinivasan, S., Muckenhirn, H., Cai, H., Mandhane, A., Tariq, M., Rae, J. W., Wang, G., Ayoub, K., FitzGerald, N., Zhao, Y., Han, W., Alberti, C., Garrette, D., Krishnakumar, K., Gimenez, M., Levskaya, A., Sohn, D., Matak, J., Iturrate, I., Chang, M. B., Xiang, J., Cao, Y., Ranka, N., Brown, G., Hutter, A., Mirrokni, V., Chen, N., Yao, K., Egyed, Z., Galilee, F., Liechty, T., Kallakuri, P., Palmer, E., Ghemawat, S., Liu, J., Tao, D., Thornton, C., Green, T., Jasarevic, M., Lin, S., Cotruta, V., Tan, Y.-X., Fiedel, N., Yu, H., Chi, E., Neitz, A., Heitkaemper, J., Sinha, A., Zhou, D., Sun, Y., Kaed, C., Hulse, B., Mishra, S., Georgaki, M., Kudugunta, S., Farabet, C., Shafran, I., Vlasic, D., Tsitsulin, A., Ananthanarayanan, R., Carin, A., Su, G., Sun, P., V, S., Carvajal, G., Broder, J., Comsa, I., Repina, A., Wong, W., Chen, W. W., Hawkins, P., Filonov, E., Loher, L., Hirnschall, C., Wang, W., Ye, J., Burns, A., Cate, H., Wright, D. G., Piccinini, F., Zhang, L., Lin, C.-C., Gog, I., Kulizhskaya, Y., Sreevatsa, A., Song, S., Cobo, L. C., Iyer, A., Tekur, C., Garrido, G., Xiao, Z., Kemp, R., Zheng, H. S., Li, H., Agarwal, A., Ngani, C., Goshvadi, K., Santamaria-Fernandez, R., Fica, W., Chen, X., Gorgolewski, C., Sun, S., Garg, R., Ye, X., Eslami, S. M. A., Hua, N., Simon, J., Joshi, P., Kim, Y., Tenney, I., Potluri, S., Thiet, L. N., Yuan, Q., Luisier, F., Chronopoulou, A., Scellato, S., Srinivasan, P., Chen, M., Koverkathu, V., Dalibard, V., Xu, Y., Saeta, B., Anderson, K., Sellam, T., Fernando, N., Huot, F., Jung, J., Varadarajan, M., Quinn, M., Raul, A., Le, M., Habalov, R., Clark, J., Jalan, K., Bullard, K., Singhal, A., Luong, T., Wang, B., Rajayogam, S., Eisenschlos, J., Jia, J., Finchelstein, D., Yakubovich, A., Balle, D., Fink, M., Agarwal, S., Li, J., Dvijotham, D., Pal, S., Kang, K., Konzelmann, J., Beattie, J., Dousse, O., Wu, D., Crocker, R., Elkind, C., Jonnalagadda, S. R., Lee, J., Holtmann-Rice, D., Kallarackal, K., Liu, R., Vnukov, D., Vats, N., Invernizzi, L., Jafari, M., Zhou, H., Taylor, L., Prendki, J., Wu, M., Eccles, T., Liu, T., Kopparapu, K., Beaufays, F., Angermueller, C., Marzoca, A., Sarcar, S., Dib, H., Stanway, J., Perbet, F., Trdin, N., Sterneck, R., Khorlin, A., Li, D., Wu, X., Goenka, S., Madras, D., Goldshtein, S., Gierke, W., Zhou, T., Liu, Y., Liang, Y., White, A., Li, Y., Singh, S., Bahargam, S., Epstein, M., Basu, S., Lao, L., Ozturel, A., Crous, C., Zhai, A., Lu, H., Tung, Z., Gaur, N., Walton, A., Dixon, L., Zhang, M., Globerson, A., Uy, G., Bolt, A., Wiles, O., Nasr, M., Shumailov, I., Selvi, M., Piccinno, F., Aguilar, R., McCarthy, S., Khalman, M., Shukla, M., Galic, V., Carpenter, J., Villela, K., Zhang, H., Richardson, H., Martens, J., Bosnjak, M., Belle, S. R., Seibert, J., Alnahlawi, M., McWilliams, B., Singh, S., Louis, A., Ding, W., Popovici, D., Simicich, L., Knight, L., Mehta, P., Gupta, N., Shi, C., Fatehi, S., Mitrovic, J., Grills, A., Pagadora, J., Petrova, D., Eisenbud, D., Zhang, Z., Yates, D., Mittal, B., Tripuraneni, N., Assael, Y., Brovelli, T., Jain, P., Velimirovic, M., Akbulut, C., Mu, J., Macherey, W., Kumar, R., Xu, J., Qureshi, H., Comanici, G., Wiesner, J., Gong, Z., Ruddock, A., Bauer, M., Felt, N., GP, A., Arnab, A., Zelle, D., Rothfuss, J., Rosgen, B., Shenoy, A., Seybold, B., Li, X., Mudigonda, J., Erdogan, G., Xia, J., Simsa, J., Michi, A., Yao, Y., Yew, C., Kan, S., Caswell, I., Radebaugh, C., Elisseeff, A., Valenzuela, P., McKinney, K., Paterson, K., Cui, A., Latorre-Chimoto, E., Kim, S., Zeng, W., Durden, K., Ponnapalli, P., Sosea, T., Choquette-Choo, C. A., Manyika, J., Robenek, B., Vashisht, H., Pereira, S., Lam, H., Velic, M., Owusu-Afriyie, D., Lee, K., Bolukbasi, T., Parrish, A., Lu, S., Park, J., Venkatraman, B., Talbert, A., Rosique, L., Cheng, Y., Sozanschi, A., Paszke, A., Kumar, P., Austin, J., Li, L., Salama, K., Kim, W., Dukkipati, N., Baryshnikov, A., Kaplanis, C., Sheng, X., Chervonyi, Y., Unlu, C., de Las Casas, D., Askham, H., Tunyasuvunakool, K., Gimeno, F., Poder, S., Kwak, C., Miecnikowski, M., Mirrokni, V., Dimitriev, A., Parisi, A., Liu, D., Tsai, T., Shevlane, T., Kouridi, C., Garmon, D., Goedeckemeyer, A., Brown, A. R., Vijayakumar, A., Elqursh, A., Jazayeri, S., Huang, J., Carthy, S. M., Hoover, J., Kim, L., Kumar, S., Chen, W., Biles, C., Bingham, G., Rosen, E., Wang, L., Tan, Q., Engel, D., Pongetti, F., de Cesare, D., Hwang, D., Yu, L., Pullman, J., Narayanan, S., Levin, K., Gopal, S., Li, M., Aharoni, A., Trinh, T., Lo, J., Casagrande, N., Vij, R., Matthey, L., Ramadhana, B., Matthews, A., Carey, C., Johnson, M., Goranova, K., Shah, R., Ashraf, S., Dasgupta,

K., Larsen, R., Wang, Y., Vuyyuru, M. R., Jiang, C., Ijazi, J., Osawa, K., Smith, C., Boppana, R. S., Bilal, T., Koizumi, Y., Xu, Y., Altun, Y., Shabat, N., Bariach, B., Korchemniy, A., Choo, K., Ronneberger, O., Iwuanyanwu, C., Zhao, S., Soergel, D., Hsieh, C.-J., Cai, I., Iqbal, S., Sundermeyer, M., Chen, Z., Bursztein, E., Malaviya, C., Biadsy, F., Shroff, P., Dhillon, I., Latkar, T., Dyer, C., Forbes, H., Nicosia, M., Nikolaev, V., Greene, S., Georgiev, M., Wang, P., Martin, N., Sedghi, H., Zhang, J., Banzal, P., Fritz, D., Rao, V., Wang, X., Zhang, J., Patraucean, V., Du, D., Mordatch, I., Jurin, I., Liu, L., Dubey, A., Mohan, A., Nowakowski, J., Ion, V.-D., Wei, N., Tojo, R., Raad, M. A., Hudson, D. A., Keshava, V., Agrawal, S., Ramirez, K., Wu, Z., Nguyen, H., Liu, J., Sewak, M., Petrini, B., Choi, D., Philips, I., Wang, Z., Bica, I., Garg, A., Wilkiewicz, J., Agrawal, P., Li, X., Guo, D., Xue, E., Shaik, N., Leach, A., Khan, S. M., Wiesinger, J., Jerome, S., Chakladar, A., Wang, A. W., Ornduff, T., Abu, F., Ghaffarkhah, A., Wainwright, M., Cortes, M., Liu, F., Maynez, J., Terzis, A., Samangouei, P., Mansour, R., Kepa, T., Aubet, F.-X., Algymr, A., Banica, D., Weisz, A., Orban, A., Senges, A., Andrejczuk, E., Geller, M., Santo, N. D., Anklin, V., Merey, M. A., Baeuml, M., Strohman, T., Bai, J., Petrov, S., Wu, Y., Hassabis, D., Kavukcuoglu, K., Dean, J., and Vinyals, O. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Ye, T., Xu, Z., Li, Y., and Allen-Zhu, Z. (2024a). Physics of language models: Part 2.1, grade-school math and the hidden reasoning process.

Ye, T., Xu, Z., Li, Y., and Allen-Zhu, Z. (2024b). Physics of language models: Part 2.2, how to learn from mistakes on grade-school math problems.

Yu, T., Xu, A., and Akkiraju, R. (2024). In defense of rag in the era of long-context language models.

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. (2024). ∞bench: Extending long context evaluation beyond 100k tokens.

A Task Generation and Distribution

In this section, we provide a detailed explanation of the process used to generate the synthetic tasks in the FACTOR benchmark. These tasks are meticulously designed to evaluate the reasoning abilities of language models over varying levels of task complexity and context length, while ensuring that these two factors are independently controlled.

A.1 Task Generation Process

The generation of each task involves several steps: creating variables and relationships, generating the payload, preparing the context, inserting the payload into the context, and forming the question prompt. This process is carefully constructed to introduce variability and prevent models from relying on simple heuristics or memorization.

A.1.1 Variable and Relationship Creation

To control task complexity, we vary the number of variables N, where each variable is denoted as v_i for $i=0,1,...,N-1$. We establish interdependent relationships among these variables by generating a directed forest—a collection of trees representing dependencies.

We start by randomly shuffling the variables to introduce variability. For each variable v_i where $i \geq k$ (with k being a randomly selected parameter between 1 and N), we randomly select a parent variable v_p from the set $\{v_0,v_1,...,v_{i-1}\}$. This process creates directed edges from v_p to v_i, establishing dependency relationships.

To define the mathematical relationships between the variables, we assign simple operations to each edge. These operations are randomly chosen from{no operation, $+1$, -1}. The use of these simple operations ensures that the calculations required to solve the tasks remain within the realm of basic arithmetic, preventing models from facing overly complex computations.

Variable Value Assignment After establishing the relationships, we assign integer values to the variables while satisfying the dependencies. For root variables (those without parents), we assign random integer values between 0 and 10. This range is chosen to keep the numerical values small, again to prevent the need for complicated calculations.

For each child variable v_i, its value is computed based on its parent's value and the assigned operation:

$$\text{value}(v_i) = \begin{cases} \text{value}(v_p)+1, & \text{if the operation is } +1 \\ \text{value}(v_p)-1, & \text{if the operation is } -1 \\ \text{value}(v_p), & \text{if there is no operation} \end{cases}$$

This approach maintains simplicity in the calculations required, ensuring that the tasks assess the models' reasoning abilities rather than computational prowess.

A.1.2 Payload Generation

The payload consists of the variable relationships formatted as textual statements. Each relationship is expressed as an assignment statement, enclosed within triple angle brackets `<<<` and `>>>` to clearly distinguish them from the filler text. These statements are further enclosed within single `@` symbols when inserted into the context.

An example of a payload statement is:

$$\texttt{<<<assign } v_i = v_p \text{ [operation]}\texttt{>>>}$$

where `[operation]` is either `+ 1`, `- 1`, or left empty if there is no operation. The payload statements are shuffled to present the relationships in a non-sequential order, adding to task complexity by preventing models from relying on the order of presentation.

A.1.3 Context Preparation

To control the context length independently, we generate filler text of specified lengths. The filler text is irrelevant to the variable relationships and serves to increase the context length, simulating scenarios where critical information is embedded within large amounts of unrelated data.

The filler text is created by randomly selecting words from a predefined list related to computational topics, such as "algorithm," "data," "performance," and so on. These words are concatenated to form sentences, with occasional punctuation added to simulate natural language text. The filler text may also include random sentences or phrases inserted between the payload statements to further obscure the relationships and mimic real-world text where key information is interleaved with irrelevant content.

A.1.4 Payload Insertion

The payload of variable relationships is inserted into the filler text at random positions. The filler text is first tokenized into sentences using the NLTK library's sentence tokenizer. Insertion points are determined based on specified intervals, ensuring that the payloads are dispersed throughout the context rather than clustered together.

Each payload statement, enclosed within `@` symbols and `<<<>>>` brackets, is inserted at the selected points. The maximum group size parameter controls how many payload statements can be inserted together at a single insertion point, adding another layer of variability. Additionally, filler content may be placed between payload statements, further increasing the challenge by requiring the model to discriminate between relevant and irrelevant information.

A.1.5 Question Prompt Formation

Finally, we generate the question prompt that instructs the model on the task to perform. The prompt includes:

- Delimiters indicating the beginning and end of the text. - An explanation that the relationships enclosed by `<<<` and `>>>` are independent mathematical equations that are all true simultaneously, and not sequential assignments in a programming language. - A task instruction, asking the model to determine which variable(s), if any, are equal to a randomly selected target value. The target value is chosen from among the variable values or from values slightly outside the range of assigned values, which may result in no variables matching the target, thereby introducing unpredictability. - A request for the model to show step-by-step reasoning and conclude with the final answer in a sentence.

This structure ensures that the model must process and reason over the entire context, filtering out irrelevant information and correctly interpreting the relationships to arrive at the answer.

A.2 Task Distribution

We generate tasks across a wide range of complexities and context lengths to create a comprehensive evaluation suite.

A.2.1 Complexity Levels

The number of variables N varies from 1 to 39, covering a broad spectrum of task complexities. For each value of N, we generate multiple tasks with different configurations to ensure diversity. The parameter k, controlling the number of trees in the forest, is randomly selected for each task and ranges from 1 to N.

A.2.2 Context Lengths

Context lengths are set to predefined values: 0 (no filler text), 1K, 2K, 4K, 8K, 16K, 32K, 64K, and 128K tokens. Our benchmark in the main context only include subsets of 0 and 4K or more. For each context length, filler text

is generated accordingly, and the payloads are inserted as described. The inclusion of different context lengths allows us to evaluate how models handle tasks when key information is embedded within varying amounts of irrelevant text.

A.2.3 Sample Sizes

For each combination of N and context length, we generate 50 distinct task instances. This results in a total of 1,950 tasks for each context length setting (39 values of N times 50 tasks). The large sample size ensures that our evaluation is statistically robust and that any observed trends are not due to random chance.

A.3 Example Task

An example of a generated task is as follows:

Sample Task

Context:
This is the beginning of the text:
@<<<assign $v_1 = v_4 - 1$>>>@ *algorithm data optimization.* @<<<assign $v_0 = v_4 - 1$>>>@ *performance analysis.* @<<<assign $v_3 = v_4 + 1$>>>@ *code snippet.* @<<<assign $v_2 = 1$>>>@ *best practice.* @<<<assign $v_4 = v_2$>>>@
[...filler text continues...]
This is the end of the text.
The text contains relationships between variables enclosed by '<<<' and '>>>'. These relationships are not sequential assignments in a programming language; they are independent mathematical equations that are all true simultaneously.
Using only these relationships, determine what variable(s), if any, are equal to 2. Show your step-by-step reasoning and calculations, and then conclude your final answer in a sentence.
Answer:
v_3

In this example, the variables v_0 to v_4 have interdependent relationships defined by the equations provided. Filler content, such as "algorithm data optimization" and "performance analysis," is interspersed between the payload statements, increasing the context length and complexity. The task requires the model to:

- Extract the relevant variable relationships from the payloads. - Interpret and solve the system of equations. - Determine which variable(s) equal the target value (in this case, 2). - Present the reasoning and final answer in a coherent manner.

This example illustrates how the task assesses the model's ability to perform multi-step reasoning over extended contexts that include irrelevant information.

A.4 Ensuring Diversity and Avoiding Data Leakage

To prevent models from exploiting patterns or memorizing specific instances, we employ several strategies:

- Variable names are randomly shuffled for each task instance. - Relationships are presented in a random order and interleaved with filler content. - Operations assigned to edges are randomly selected from simple options to maintain calculation simplicity while adding variability. - Target values for the query are randomly chosen and may not correspond to any variable value in the task, introducing the possibility that no variables meet the condition.

These measures create a wide variety of task instances, reducing the likelihood of data leakage or overfitting. Models must genuinely understand and reason through each task rather than relying on memorization or pattern recognition.

A.5 Summary

The task generation process in the FACTOR benchmark is carefully designed to independently control task complexity and context length while preventing models from relying on shortcuts. By varying the number of variables and their interdependencies, we manipulate task complexity. By inserting filler text of specified lengths and interleaving filler content between payload statements, we manipulate context length and complexity. The use of simple operations and small integer values ensures that the focus remains on reasoning rather than computation.

This systematic approach allows us to create a diverse set of tasks that robustly evaluate models' reasoning abilities over long contexts. By disentangling the effects of task complexity and context length on model performance, the FACTOR benchmark facilitates a deeper understanding of models' strengths and limitations in reasoning over extended textual inputs, guiding future improvements in language model development.

B Data Analysis and Linear Regression Methodology

This section details the methodology used for the linear regression analyses in our study, including data selection criteria, regression techniques, handling of low accuracy values, and confidence interval calculations. The approach aligns with the code used in our interactive analysis.

B.1 Two-Phase Accuracy Behavior and Data Selection

Our experiments reveal a characteristic two-phase accuracy behavior as task complexity N increases:

(1) **Phase 1**: High accuracy plateau where models perform well on simpler tasks, typically with accuracies close to 100%.

(2) **Phase 2**: Exponential decay of accuracy as task complexity exceeds a certain threshold N_{eff}.

To model the rate of accuracy decline in Phase 2, we perform linear regression on the natural logarithm of accuracy versus task complexity N. It is essential to focus on data from Phase 2 only, to avoid distortion from the Phase 1 plateau.

B.2 Data Selection Criteria

Selecting appropriate data ranges is crucial for accurate regression. We use different accuracy ranges for different models and experimental stages to focus on Phase 2 data:

(1) **Pretrained Models**: Accuracy between [0.1,0.9]. (2) **Train-from-Scratch Models (Pretraining Stage)**: Accuracy between [0.1, 0.8]. (3) **Fine-tuned Models**: Accuracy between [0.2, 0.8]. (4)**Repeated Sampling Experiments**: Accuracy between [0.02,0.8].

These ranges exclude Phase 1 data (high accuracy plateau) and avoid extremely low accuracies that can lead to numerical instability.

B.3 Linear Regression Methodology

For each model and experimental condition, we perform linear regression to fit:

$$\log(A) = aN + b$$

where:

- A is the accuracy for task complexity N.
- a is the **Complexity Decay Factor (CDF)**, indicating the rate of exponential decay.
- b is the **Contextual Decay Offset (CDO)**, representing baseline performance at the onset of Phase 2.

B.3.1 Confidence Interval Calculation

We calculate confidence intervals for the regression coefficients using their standard errors and the inverse error function (erf^{-1}):

$$\text{CI} = \text{Coefficient} \pm \left(\text{SE} \times \sqrt{2} \times \text{erf}^{-1}\left(\frac{C}{100} \right) \right)$$

where C is the confidence level (e.g., 95%).

B.4 Summary

Our data analysis and regression methodology accurately capture the relationship between task complexity and model accuracy during the exponential decay phase. By carefully selecting data ranges, we ensure reliable regression results that provide valuable insights into models' abilities to handle complex reasoning tasks over extended contexts.

C Training Configurations

In this section, we provide a concise overview of the training configurations used for developing our language models evaluated on the FACTOR benchmark. Key configurations are summarized in Table 8, and additional details are described to clarify the training setup.

C.1 Model Architecture

We employ a Llama3 architecture configured to handle long sequences and complex reasoning tasks. The model consists of 12 Transformer layers, each with 12 attention heads, resulting in a hidden size of 768 (calculated as `num_heads` $\times$ 64). The intermediate size is set to 3,072 (four times the hidden size), following common practice to enhance model capacity.

To effectively handle long contexts, we use Rotary Position Embeddings with a RoPE theta value of 500,000.0. This allows the model to capture positional information over extended sequences. We utilize FlashAttention-2 for efficient attention computation on long sequences, which improves training speed and reduces memory consumption.

Table 8: Summary of Training Configurations.

Configuration Element	Setting
Model Architecture	LlamaForCausalLM
Number of Layers	12
Number of Attention Heads	12
Hidden Size	768
Intermediate Size	3,072
Vocabulary Size	424
Maximum Position Embeddings	32,768
RoPE Theta	500,000.0
Data Type	`bfloat16`
Peak Learning Rate	4e-4
Batch Size	192
Learning Rate Scheduler	Cosine decay with warmup ratio 0.01
Number of Training Epochs	1
Optimizer	AdamW

All computations are performed using `bfloat16` precision to optimize memory usage and computational efficiency without significantly affecting model accuracy.

C.2 Tokenizer

A custom tokenizer is used, tailored to the specific needs of the FACTOR benchmark tasks. The tokenizer has a vocabulary size of 424 tokens.

Optimizer and Scheduler The AdamW optimizer is utilized with default parameters except for the learning rate. The cosine learning rate scheduler with warmup helps in gradually adapting the learning rate, reducing the risk of training divergence at the start.

D Training Data Generation

This section details the generation of the training data used in both the pretraining and posttraining phases. Our approach carefully controls task complexity and context length to effectively train the models for evaluating their reasoning abilities on the FACTOR benchmark.

D.1 Pretraining Dataset

The pretraining dataset comprises 3 million synthetic Question-Solution pairs. It is designed to teach the model fundamental reasoning skills over a range of task complexities and shorter context lengths.

D.1.1 Task Complexity Distribution

To control task complexity, we vary the number of variables N in each synthetic example. The value of N is determined by:

$$N = \min(N_1, N_2), \quad N_1, N_2 \sim \text{Uniform}(1, 29)$$

where Uniform(1,29) denotes a discrete uniform distribution over integers from 1 to 29 (excluding 30). By taking the minimum of two independently sampled values, we bias the distribution toward smaller values of N, emphasizing simpler tasks while still including examples with higher complexities up to $N = 29$.

D.1.2 Filler Context Length Distribution

The length of the filler context L_{pretrain} is sampled from a log-normal distribution to introduce variability in context lengths while maintaining manageable sequence sizes for pretraining. Specifically, we use:

$$\ln L \sim \mathcal{N}(\mu_{\ln L}, \sigma^2_{\ln L}), \quad L_{\text{pretrain}} = \exp(\ln L) - L_{\min}$$

where: $\mu_{\ln L} = 5$, $\sigma_{\ln L} = 2.5$, $L_{\min} = 100$, $L_{\max} = 1000$

After sampling $\ln L$, we compute L, round it to the nearest integer, and clip it to the range $[L_{\min}, L_{\max}]$. We then subtract $L_{\min}$ to adjust the lengths to start from zero. This results in a distribution of filler context lengths biased toward shorter lengths but including a range up to 900 tokens.

D.2 Post-training Dataset

The post-training dataset consists of 60,000 synthetic long-context Question-Solution pairs. It is used to fine-tune the pretrained model, enhancing its ability to handle extended contexts and complex reasoning tasks.

We employ three different training strategies to investigate their effects on model performance:

1. **Gradually Increasing Context Length**: The dataset is divided into four equal parts, each containing 15,000 examples. Each part corresponds to filler context lengths that are progressively increased. Specifically, the filler context length distribution in each stage is scaled by factors of 2, 4, 8, and 16 times the pretraining filler context length L_{pretrain}, respectively. By gradually increasing the context length, the model is incrementally adapted to handle longer sequences, reaching the maximum context length in the final stage.

2. **Mixture of Different Context Lengths**: In this strategy, the examples from all four stages of the gradually increasing context length dataset are combined and shuffled. Training on this mixed dataset exposes the model to different context lengths simultaneously, which may encourage better generalization across various sequence lengths.

3. **Training Directly with Full Context Length**: The model is trained starting from a filler context length distribution scaled by $16 \times L_{\text{pretrain}}$. This approach tests the model's ability to handle long contexts without prior adaptation through shorter sequences.

D.2.1 Task Complexity Distribution

For all posttraining strategies, the number of variables N is sampled using the same method as in pretraining:

$$N = \min(N_1, N_2), \quad N_1, N_2 \sim \text{Uniform}(1, 29)$$

This ensures consistency in task complexity across both pretraining and posttraining datasets.

D.2.2 Filler Context Length Generation

For each stage in the gradually increasing context length strategy, the filler context lengths are generated by scaling the pretraining filler context lengths L_{pretrain}:

$$L_{\text{stage}} = s \times L_{\text{pretrain}}$$

where s is the scaling factor for each stage ($s = 2, 4, 8, 16$).

The filler context lengths are generated using the same log-normal distribution as in pretraining but adjusted for the scaling factor. This adjustment ensures that the filler context lengths are appropriately scaled for each stage.

D.2.3 Synthetic Data Generation Process

For each example in both the pretraining and posttraining datasets, we follow the same synthetic data generation steps as described in Section A, adjusting the filler context lengths according to the stage and strategy.

D.3 Summary

By carefully generating training data with controlled task complexity and systematically varied context lengths, we aim to investigate the impact of different training strategies on the models' abilities to handle complex reasoning tasks over long contexts. The datasets are designed to provide comprehensive exposure to the challenges posed by increased sequence lengths and to facilitate a detailed analysis of model performance under various training conditions.