LOFT: Scalable and More Realistic Long-Context Evaluation

Anonymous ACL submission

Abstract

Long-context language models (LCLMs) have the potential to revolutionize our approach to tasks traditionally reliant on external tools like retrieval systems or databases. Leveraging LCLMs' ability to natively ingest and process entire corpora of information offers numerous advantages. It enhances user-friendliness by eliminating the need for specialized knowledge of tools, provides robust end-to-end modeling that minimizes cascading errors in complex pipelines, and allows for the application of sophisticated prompting techniques across the entire system. To assess this paradigm shift, we introduce LOFT, a benchmark of real-world tasks requiring context up to millions of tokens designed to evaluate LCLMs' performance on in-context retrieval and reasoning. We find LCLMs possess the surprising ability to rival state-of-the-art retrieval and RAG systems, despite never having been explicitly trained for these tasks. However, LCLMs still face challenges in areas like compositional reasoning that are required in SQL-like tasks. Notably, prompting strategies significantly influence performance, emphasizing the need for continued research. Overall, LOFT provides a rigorous testing ground for LCLMs, showcasing their capabilities to tackle existing paradigms.[1]

1 Introduction

Long-context language models (LCLMs) (Beltagy et al., 2020; Guo et al., 2022; OpenAI, 2023; Anthropic, 2024; Team, 2024) hold the promise of reshaping artificial intelligence by enabling entirely new tasks and applications while eliminating the reliance on tools and complex pipelines previously necessary due to context length limitations (Guu et al., 2020; Lewis et al., 2020). By consolidating complex pipelines into a unified model, LCLMs ameliorate issues like cascading errors (Barnett et al., 2024) and cumbersome optimization (Lee et al., 2022; Xiong et al., 2021), offering a streamlined end-to-end approach to model development. Moreover, techniques such as adding instructions (Kojima et al., 2022; Wei et al., 2022a; Chung et al., 2024), incorporating few-shot examples (Brown et al., 2020), and leveraging demonstrations via chain-of-thought prompting (Nye et al., 2022; Wei et al., 2022b) can be seamlessly integrated to optimize LCLMs for the task at hand.

However, understanding the full potential of LCLMs requires rigorous evaluation on truly long-context tasks that approximate real-world applications. Existing benchmarks fall short, relying on synthetic tasks like "needle-in-a-haystack" (Kamradt, 2023), or otherwise use fixed-length datasets that fail to keep pace with the evolving definition of "long-context" (Bai et al., 2023). Critically, existing evaluations do not adequately test LCLMs on any paradigm-shifting tasks.

To address this, we introduce the **Lo**ng-Context **F**ron**t**iers (LOFT) benchmark, a suite of six tasks consisting of 35 datasets which span text, visual, and audio modalities. LOFT tests LCLMs on more realistic user queries and knowledge sources than previous benchmarks, while allowing automatic scaling to longer context lengths. While the current version extends to one million tokens, it can easily be extended further to billions of tokens, ensuring realistic evaluation as LCLMs continue to scale. LOFT focuses on the following areas where LCLMs have the potential for disruption:

- **Retrieval**: LCLMs can directly ingest and retrieve information from a corpus, eliminating the need for separate dual-encoder models (Karpukhin et al., 2020; Lee et al., 2024; Radford et al., 2021). This addresses long-standing challenges in retrieval systems such as multi-hop reasoning, instruction following, and few-shot

[1] The LOFT benchmark is provided in the supplementary material and will be available online.

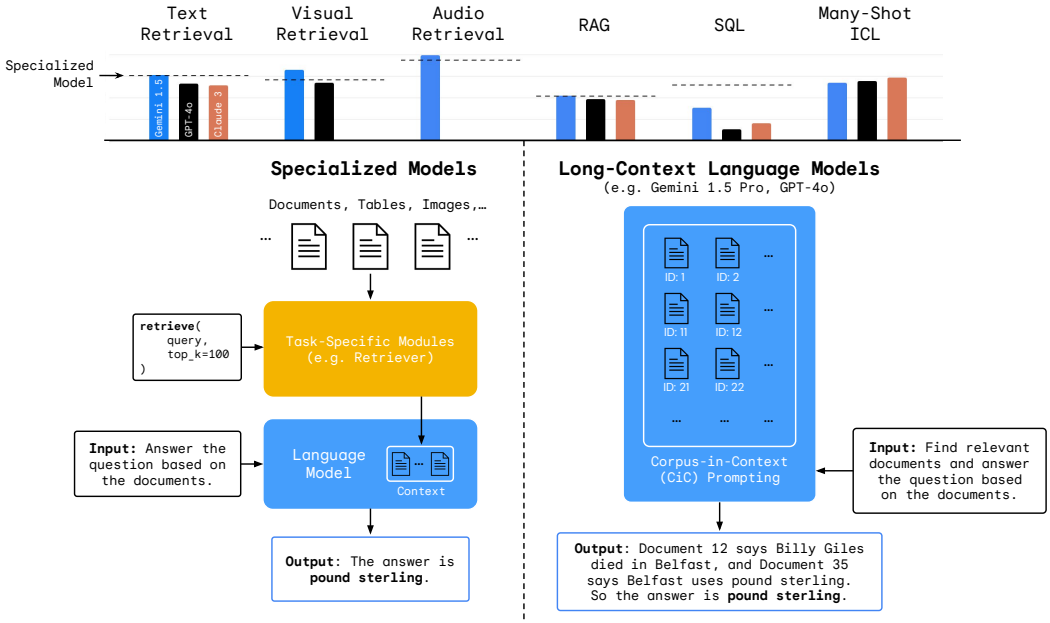

Figure 1: An overview of the LOFT benchmark, made of six tasks which measure LCLMs' ability to do in-context retrieval, reasoning, and many-shot learning on corpora up to millions of tokens. We compare the performance of LCLMs against *specialized models* (*e.g.,* CLIP for visual retrieval), which often rely on complex task-specific fine-tuning or pipelining. Unlike specialized models, we show how LCLMs can simplify various tasks through Corpus-in-Context Prompting (§3).

task adaptation. We assess retrieval performance across text, visual, and audio modalities.

- **Retrieval-Augmented Generation (RAG)**: LCLMs simplify RAG pipelines by directly reasoning over a corpus, overcoming challenges like query decomposition (Perez et al., 2020) and cascading errors from retrievers (Sciavolino et al., 2021; Longpre et al., 2021).

- **SQL**: We explore LCLMs' capacity to process entire databases as text. By asking the model to directly generate the answer based on a database and a natural language query, we bypass the need for conversion to formal query languages like SQL (Zhong et al., 2017). While this is not an immediate practical solution, it potentially enables more expressive querying and handling of noisy or mixed-structured data.

- **Many-Shot ICL**: LCLMs can scale the number of examples from the tens in the traditional in-context learning setup to hundreds or thousands (Yu et al., 2020a; bench authors, 2023), removing the need to find the optimal set of few-shot examples to use (Luo et al., 2023).

The LOFT benchmark opens up a novel line of research on long-context prompting, which we introduce as Corpus-in-Context (CiC) Prompting (§3). Using this approach, we evaluate Gemini 1.5 Pro (Team, 2024), GPT-4o (OpenAI, 2023), and Claude 3 Opus (Anthropic, 2024) on LOFT. Figure 1 summarizes the performance of these LCLMs and specialized models that were carefully hand-optimized for each task, showcasing how LCLMs can tackle LOFT tasks without specialized pipelines.

Our evaluation on LOFT reveals several key insights when comparing state-of-the-art LCLMs with specialized, task-specific models. At the 128k token level, the largest size comparable across all models, LCLMs rival the performance of Gecko (Lee et al., 2024), a leading text retrieval system. Notably, Gemini (Team, 2024) also surpasses strong multi-modal retrieval models such as CLIP (Radford et al., 2021). However, LCLMs lag significantly on complex multi-hop compositional reasoning tasks, indicating substantial room for improvement. Furthermore, rigorous ablations reveal substantial variations in performance depending on the prompting strategy, such as the inclusion of chain-of-thought reasoning, underscoring the need for further research to enhance LCLMs robustness and instructability. Taken together, our results on LOFT demonstrate that LCLMs can match the performance of many specialized models, while also revealing ample headroom for improvement in robust long-context reasoning as context windows continue to scale.

2 LOFT: 1 Million+ Token Benchmark

LOFT consists of 35 datasets across three modalities, covering various tasks including retrieval, retrieval-augmented generation (RAG), reasoning over structured databases (SQL), and other tasks measuring in-context learning. The full list of tasks can be found in Table A.1. Unlike existing synthetic datasets, LOFT makes use of carefully chosen academic datasets that are derived from real user queries (see Appendix A on how the datasets are selected). To test how models perform with different context lengths, LOFT supports 32k, 128k, and 1M token contexts and can easily scale to even larger contexts (e.g., 1 billion tokens) in the future.

Retrieval & RAG In retrieval and RAG tasks, the LCLM takes an entire corpus as the LCLM's context, and outputs the relevant document IDs for retrieval or the answer to a query for RAG. We include diverse text retrieval and RAG datasets, covering heterogeneous retrieval tasks from BEIR (Thakur et al., 2021), multi-turn conversational QA (Adlakha et al., 2022), multi-hop QA (Yang et al., 2018; Trivedi et al., 2022), as well as multi-target QA that requires set operations such as unions or differences (Amouyal et al., 2022; Malaviya et al., 2023). For retrieval, we also include multimodal datasets, covering image, video, and audio (Young et al., 2014; Xu et al., 2016; Conneau et al., 2022).

All queries in each retrieval and RAG dataset share a single corpus, mimicking real retrieval applications. To create this shared corpus, we first include all gold passages from few-shot, development and the test queries, and then sample passages uniformly until reaching the desired context size. This construction ensures smaller corpora (e.g., 128k) are subsets of larger ones (e.g., 1M). Gold and random passages are shuffled to avoid positional biases. Note that the negatives for this sampling method are all random, so it might be easier than a traditional re-ranking task. However, at the longest context lengths, the model must differentiate between the gold passage and every other passage in-context, which naturally includes hard negatives. We report how well specialized retriever models perform on the same setup for comparison.

SQL We evaluate SQL-like reasoning on Spider, a single-turn text-to-SQL dataset (Yu et al., 2018), and SparC, its multi-turn variant (Yu et al., 2019). The model input is a natural language query concatenated with its associated database, consisting of one or more tables. Unlike traditional semantic parsing, where the output is a SQL query, in our long-context setting, we require the model to output chain-of-thought reasoning in natural language followed by the prediction, unassisted by SQL. This setting directly stress-tests the models' ability to do complex compositional reasoning.

To construct the corpus for each context length, we select databases such that all of its tables are no larger than that context length. We do this by selecting the largest databases that will fit into that context. This means that the databases for the 1M token setting would not fit into the smaller context lengths. Therefore, unlike other tasks that share a corpus, the queries differ across LOFT sizes.

Many-shot ICL We adapt datasets from BigBench Hard (BBH) (bench authors, 2023; Suzgun et al., 2023) and LongICLBench (LIB) (Yu et al., 2020a; Li et al., 2024) to evaluate many-shot in-context learning (ICL) capabilities. We construct shared many-shot ICL contexts, ensuring training examples in smaller contexts are included in larger ones. Since our many-shot ICL datasets are classification tasks, we guarantee that each class is represented at least once.

For each dataset in LOFT, we sample up to 100 test queries, 5 few-shot queries, and 10 development queries. To allow testing the same set of queries over different context lengths, we process each dataset to have the same evaluation queries across 32k, 128k and 1M context lengths[2] (except for SQL as detailed above). Given a maximum context length of $N \in \{32\text{k}, 128\text{k}, 1\text{M}\}$, we create a corpus up to a size of $0.9N$, to account for differences in tokenizers, as well as to reserve room for instructions and formatting, which will be explained in more detail in §3.

3 Corpus-in-Context Prompting

Traditionally, utilizing large corpora of passages, data tables, or training examples requires specialized recipes or systems. LCLMs now enable direct ingestion and processing of entire corpora within their context window. This unlocks a novel prompting-based approach for solving new and existing tasks, which we call **C**orpus-**i**n-**C**ontext prompting (**CiC**, pronounced "seek").

[2] Since the gold documents of 100 test queries alone often exceed 32k tokens, we do not include test queries for the 32k version and report the development set performance for 32k.

3.1 Prompt Design

CiC prompting effectively combines established prompting strategies, tailoring them to leverage the unique capabilities of LCLMs for learning, retrieving and reasoning over in-context corpora. Figure 2 illustrates our key design choices, which we rigorously evaluate in §5. Appendix D provides examples of instructions for all datasets in LOFT, along with details on the chain-of-thought reasonings.

Instructions We first provide task-specific instructions to guide the LCLM's behaviors (Kojima et al., 2022; Wei et al., 2022a; Chung et al., 2024). As an example, for the retrieval task shown in Figure 2, we ask the model to read the corpus carefully and find relevant documents to answer the question.

Corpus Formatting We then insert the entire corpus into the prompt. Each candidate (e.g., passage, image, audio) in a corpus is assigned a unique identifier (ID) that can be referenced as needed for that task.[3] For instance, in retrieval tasks, the LCLMs must output the correct candidate IDs. The structure of the corpus significantly impacts retrieval performance. Careful formatting, such as putting document IDs both before and after the passage in text retrieval, can mitigate the effects of causal attention in decoder-only LCLMs.

Few-Shot Examples Providing a limited number of demonstrations helps the LCLM grasp the desired response format and improves task accuracy (Brown et al., 2020). We ground all examples to refer the same corpus, aiming to teach the model to also learn more details about the specific corpus it needs to use.[4] To facilitate automated evaluation, answers within each few-shot example are formatted as a list (e.g., "Final Answer: [54, 0]" in Figure 2), thus guiding the model to generate responses in a similar structure that can be readily parsed and compared against ground truth labels. Finally, each few-shot example is accompanied by Chain-of-Thought reasoning (Nye et al., 2022).

Query Formatting The evaluation query is formatted similar to the few-shot examples. For multi-turn datasets, we prepend previous turns to the current turn, ensuring that the model's generation is conditioned on its prior responses.

[3] Original candidate IDs from each dataset were not used as they inadvertently disclose the gold documents.

[4] As shown in §4, the positions of the answers of these few-shot examples can guide the model's attention, mitigating "dead zones" in attention distribution.

3.2 Design Consideration

Given the variation in instructions, format, and tokenizers across different CiC prompting techniques, the resulting context lengths can differ substantially. To accommodate this diversity, we allocate ample space for prompt customization, as detailed in §2. To ensure a fair comparison among LCLMs, each model uses only the corpus and examples present in that version. Similarly, we only evaluate models on the maximum size that can fit into their context length without truncating the corpus or any of the individual examples.

3.3 Discussion on Efficiency

Encoding a one million token context can be computationally expensive. One key advantage of CiC prompting is its compatibility with prefix-caching in autoregressive language models (Google, 2024) as the query appears at the end of the prompt. This means *the corpus only needs to be encoded once*, similar to the indexing process in traditional information retrieval (see ablation in §5).

4 LOFT Tasks and Primary Results

We evaluate three state-of-the-art LCLMs on LOFT: Google's **Gemini 1.5 Pro** (Team, 2024), OpenAI's **GPT-4o** (OpenAI, 2023), and Anthropic's **Claude 3 Opus** (Anthropic, 2024). In all LOFT tasks, the LCLMs have no task-specific fine-tuning. For point of comparison, we also present results from *specialized* models with fine-tuning and system design that is limited to that specific domain. We select each specialized model that exemplifies recent task-specific advancements. We chose these three LCLMs because their APIs support most of the modalities. Their respective maximum context lengths are 2M, 128k, and 200k tokens at the time of the evaluation. We use their official APIs[5,6,7] for the evaluation. Their prompts were chosen based on their performance on the development queries over the 128k token context.

4.1 Text Retrieval

We use Gecko (Lee et al., 2024), a state-of-the-art dual encoder as the specialized model for the retrieval task. Gecko is fine-tuned on extensive text retrieval and similarity tasks. To ensure fair comparison, we use exactly the same corpus to test both the LCLMs and Gecko.

[5] ai.google.dev/gemini-api

[6] platform.openai.com/docs/models/gpt-4o

[7] www.anthropic.com/api

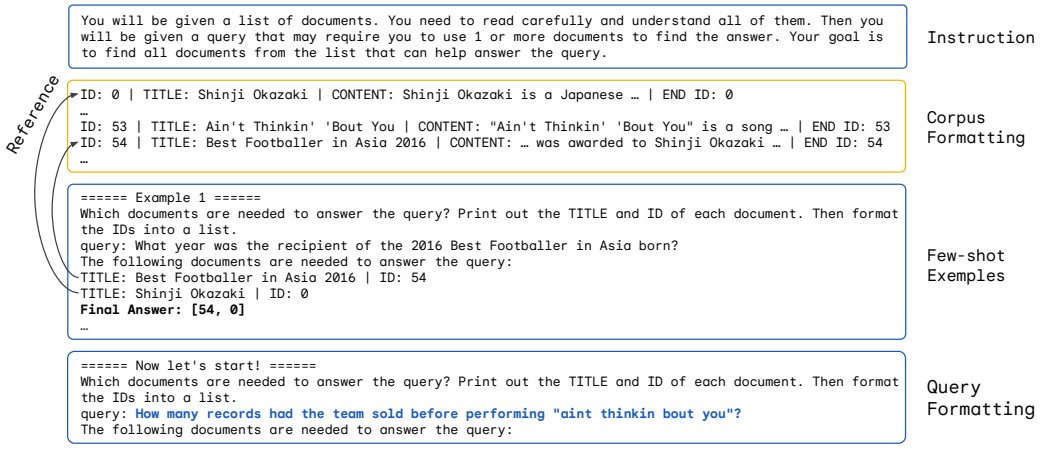

Figure 2: **Example of Corpus-in-Context Prompting** for retrieval. CiC prompting leverages large language models' capacity to follow instructions, leverage few-shot examples, and benefit from reasoning demonstrations to retrieve and reason over large corpora provided in context.

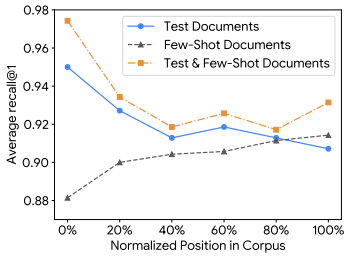

Figure 3: **Positional Analysis**. We vary gold document positions of queries within the corpus (0% = beginning, 100% = end).

Results Table 1 demonstrates that Gemini 1.5 Pro performs comparably to Gecko at 128k context length, and all of the LCLMs perform surprisingly well. This is notable, as LCLMs have not undergone specialized contrastive learning for retrieval. While LCLMs's performance does degrade when scaling the corpus to millions of tokens (Figure 4), the parity at 128k still suggests the potential of LCLMs to be used for retrieval tasks.

Positional Analysis To better understand the performance of LCLMs on longer context lengths, we investigate how the positioning of the gold document affects retrieval performance, examining the effect of the gold document for the test query and for the few shot examples (Liu et al., 2023).

Figure 3 reveals that performance drops as the gold documents of the test queries are moved towards the end of the corpus, suggesting reduced attention in later sections of the prompt. Conversely, placing the gold documents of few-shot queries at the end improves recall, indicating their ability to mitigate attention weaknesses in this region. Co-locating gold documents of few-shot and test queries consistently boosts performance as it gives the model information about where to look for the answer. This offers a promising approach to overcome performance degradation in large corpora. Per-dataset analysis is provided in Appendix E.

4.2 Visual Retrieval

We employ CLIP-L/14 (Radford et al., 2021), a widely used text-to-image retrieval model, as our specialized model. For Flickr30k and MS-COCO, CLIP performs text-to-image retrieval. For MSR-VTT, it performs text-to-video retrieval by averaging scores across frames. For OVEN, due to the lack of suitable open-source image-to-text models, we approximate image-to-text retrieval by using CLIP's text-to-image retrieval. Evaluation of Claude 3 Opus on this task was not feasible due to the current limitation of 20 images per API request.

Results Gemini 1.5 Pro outperforms GPT-4o across all four visual benchmarks (Table 1). Notably, as shown in Figure 4, Gemini 1.5 Pro maintains a performance advantage over CLIP across all visual benchmarks and context lengths.

4.3 Audio Retrieval

We choose PaLM 2 DE (Gomez et al., 2024) as a specialized model, which is a dual-encoder trained to maximize the similarity between audio and their transcription and has achieved previous state-of-the-art on the FLEURS datasets. Currently, GPT-4o and Claude 3 Opus do not support audio input.

Results Gemini 1.5 Pro demonstrates comparable performance to PaLM 2 DE across all 5 languages (Table 1). We notice that Gemini 1.5 Pro notably surpasses PaLM 2 DE in Hindi; this advantage likely stems from differences in pre-training data between Gemini and PaLM. Figure 4 further confirms Gemini 1.5 Pro's robust performance

	Dataset	Gemini 1.5 Pro	GPT-4o	Claude 3 Opus	Specialized
Text Retrieval	ArguAna	0.84	0.85	0.74	0.75
	FEVER	0.98	0.96	0.94	0.97
	FIQA	0.79	0.82	0.61	0.83
	MS MARCO	0.95	0.87	0.93	0.97
	NQ	1.00	0.99	0.96	0.99
	Quora	0.93	0.93	0.94	1.00
	SciFact	0.88	0.88	0.73	0.85
	Touché-2020	0.91	0.88	0.71	0.88
	TopiOCQA	0.49	0.30	0.42	0.36
	HotPotQA†	0.90	0.82	0.83	0.92
	MuSiQue†	0.42	0.10	0.27	0.29
	QAMPARI†	0.61	0.18	0.20	0.57
	QUEST†	0.30	0.19	0.18	0.54
	Average	**0.77**	0.67	0.65	0.76
Visual Retrieval	Flickr30k	0.84	0.65	-	0.75
	MS COCO	0.77	0.44	-	0.66
	MSR-VTT	0.76	0.72	-	0.64
	OVEN	0.93	0.89	-	0.79
	Average	**0.83**	0.68	-	0.71
Audio Retrieval	FLEURS-en	1.00	-	-	0.98
	FLEURS-es	0.99	-	-	0.99
	FLEURS-fr	1.00	-	-	1.00
	FLEURS-hi	1.00	-	-	0.74
	FLEURS-zh	1.00	-	-	1.00
	Average	**1.00**	-	-	0.94
RAG	NQ	0.84	0.89	0.85	0.71
	TopiOCQA	0.34	0.33	0.37	0.35
	HotPotQA	0.75	0.72	0.74	0.70
	MuSiQue	0.55	0.47	0.45	0.45
	QAMPARI	0.44	0.27	0.25	0.55
	QUEST	0.28	0.20	0.15	0.35
	Average	**0.53**	0.48	0.47	0.52
SQL	Spider	0.40	0.14	0.19	0.74
	SParC	0.36	0.13	0.21	0.55
	Average	0.38	0.13	0.20	**0.65**
Many-Shot ICL	BBH-date	0.88	0.81	0.92	-
	BBH-salient	0.78	0.64	0.69	-
	BBH-tracking7	0.33	0.81	0.54	-
	BBH-web	0.67	0.57	0.83	-
	LIB-dialogue	0.76	0.67	0.72	-
	Average	0.68	0.70	**0.74**	-

Table 1: **Main Results on LOFT 128k context test set**. We show performances of three LCLMs as well as specialized models that rely on task-specific fine-tuning or pipelining. For the evaluation metrics: text, visual, and audio retrieval use Recall@1; RAG uses subspan exact match; SQL uses accuracy; and many-shot ICL uses classification accuracy. †: For retrieval with multiple gold targets, MRecall@k ($k = 2, 5, 5, 3$ in order) is employed as described in Appendix A.

across various context length, highlighting the current capabilities of LCLMs while also indicating the need for more challenging audio datasets.

4.4 RAG

We set up a retrieve-and-read RAG pipeline as a specialized model, using Gecko (Lee et al., 2024) to retrieve the top-40 documents which are then put into the context of Gemini 1.5 Pro and used to generate the answer conditioned on the question and the retrieved documents.

Results Table 1 demonstrates that Gemini 1.5 Pro, with the entire corpus in context, outperforms the RAG pipeline on multi-hop datasets (HotpotQA and MusiQue). This is because LCLMs can reason over multiple passages in the context window using Chain-of-Thought (Wei et al., 2022b), a capability that RAG pipelines typically lack unless they have a separate module for planning and reasoning.

However, a specialized retriever like Gecko excels at identifying a comprehensive set of passages covering all answers. This proves particularly beneficial for multi-target datasets, such as QUEST.

Figure 4 reveals that while LCLMs match RAG performance at 128k compared to a pipeline, performance drops at 1M corresponding to the drop found in LCLM text retrieval performance.

Closed-Book Ablations To further probe LCLMs capabilities, we conduct closed-book ablations on Gemini 1.5 Pro. In this setting, we remove the corpus from the context to assess LCLM performance based solely on parametric

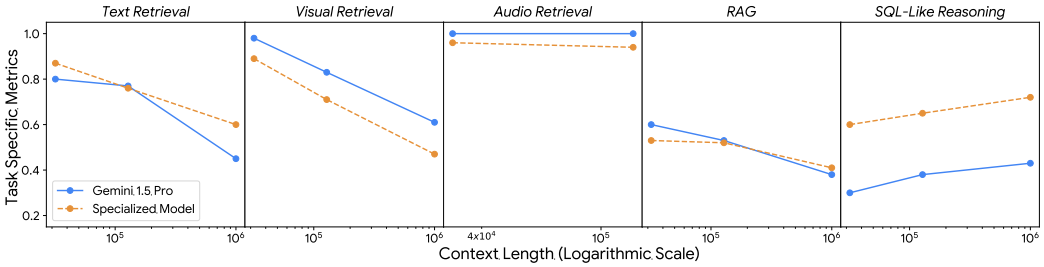

Figure 4: **Scaling results** of LCLMs compared to each specialized model by scaling the corpus size from 32k to 1 million tokens. Results are averaged over all constituent datasets in each task.

knowledge (Lewis et al., 2021; Longpre et al., 2021). Table F.1 presents the results, revealing that the closed-book performance significantly lags behind the long-context and specialized models. This underscores the tested models' effectiveness in leveraging the external corpus to enhance its reasoning capabilities.

4.5 SQL-Like Compositional Reasoning

The traditional SQL pipeline uses a trained semantic parser to translate the natural language input into a SQL query. Then, a separate SQL interpreter is used to execute the SQL query over the database. We adapt DAIL-SQL (Gao et al., 2023), a prompted LLM-based semantic parser, by using Gemini 1.5 Pro with a fixed set of few-shot examples.

Results Table 1 show that LCLMs achieve reasonable performance, though they are significantly behind semantic parsers. This reveals substantial headroom to enhance the compositional reasoning capabilities of LCLMs. This shows that LCLMs still have substantial headroom in improving their compositional reasoning over structured data, but they may still have applications in mixed-modal settings or over less well-studied formats than SQL.

Reasoning Analysis We categorize queries based on the operators in the gold SQL queries and measure Gemini 1.5 Pro's performance for each operator. Figure F.1 shows that averaging is the most difficult operation while counting is relatively easy. Moreover, we find that reasoning over equality is considerably easier than reasoning over inequality.

4.6 Many-Shot ICL

Results Table 1 compares accuracy for each of the LCLMs across all ICL datasets. For BBH, we report the accuracy on 32k, which is the maximum context length available. Gemini 1.5 Pro outperforms GPT-4o on all benchmarks, except for BBH-tracking7 where Gemini performs surprisingly poorly. On average, Claude 3 Opus achieves the best performance among LCLMs on this task.

Scaling Many Shot ICL Figure F.2 illustrates the impact of increasing the number of examples in Gemini. In LIB-dialog, accuracy improves monotonically with more examples. In contrast, results on BBH are mixed. Interestingly reasoning-intensive tasks like BBH-tracking7 and BBH-web do not benefit from scaling. These results suggest that more complicated tasks may see an earlier limit in how much models can learn from scaling the number of in-context examples.

5 CiC Prompt Ablations

W evaluate Gemini 1.5 Pro on the 128k context length version of LOFT to examine different aspects of CiC prompting. Results for each ablation are in §5, while prompts are in Appendix G.

Removing task-specific instructions (`Generic Instruction`) or Chain-of-Thought reasoning (`Without CoT`) both lead to worse performance. We also observe a performance decrease for `Corpus in Each Few-Shot`. In this setting, instead of using a shared corpus, each few-shot example has its own small corpus consisting of nine random passages and one gold passage. This performance degradation could be because the few-shot examples help the model attend to the test corpus.

Placing the query at the beginning of the prompt instead of at the end (`Query at Beginning`) led to a significant and consistent performance decrease across tasks and models. This result suggests that prefix-caching actually works better than encoding the corpus conditioned on each query, which would be much more expensive. Replacing monotonic numerical IDs with random (`Alphanumeric IDs`) negatively impacts performance in most datasets. This could possibly be due to way in which numbers are tokenized, with fewer tokens for certain numbers. Only placing the IDs at the front instead of at the front and the back (`Without ID Echo`) resulted in a 5% performance drop, confirming that repeating text can compensate for missing context in causal language models (Springer et al., 2024).

Task (Metric)	Dataset	Best Prompt	Generic Instruction	Query at Beginning	Alphanu-meric IDs	Titles Only	Without ID Echo	Corpus in Each Few-shot	Without CoT
Text Retrieval	ArguAna	0.84	0.76	0.72	0.81	-	0.78	0.62	0.79
	FIQA	0.79	0.77	0.58	0.75	-	0.76	0.78	0.85
	NQ	1.00	0.98	0.98	0.99	0.91	1.00	1.00	1.00
	SciFact	0.88	0.88	0.81	0.90	0.84	0.87	0.78	0.90
	MuSiQue	0.49	0.44	0.19	0.44	0.10	0.36	0.35	0.43
	QAMPARI	0.61	0.61	0.49	0.54	0.09	0.49	0.35	0.43
	QUEST	0.28	0.28	0.22	0.30	0.05	0.27	0.22	0.30
RAG	MuSiQue	0.55	0.57	0.39	0.52	0.23	0.55	0.78	0.50
	NQ	0.84	0.81	0.76	0.80	0.39	0.83	0.83	0.82
	QAMPARI	0.44	0.42	0.35	0.42	0.09	0.36	0.36	0.33
	QUEST	0.28	0.33	0.17	0.28	0.02	0.25	0.27	0.32
	Average	0.64	0.62	0.51	0.61	0.30	0.59	0.55	0.61
	(Δ)	-	(-0.02)	(-0.13)	(-0.03)	(-0.30)	(-0.05)	(-0.09)	(-0.03)

Table 2: Ablation results of Gemini 1.5 Pro on different tasks in LOFT at 128k context length. Starting from our best prompt format (used in the rest of the experiments), individual facets of the corpus, query, and instruction are ablated to surface their relative effect on quality. Since ArguAna and FIQA do not have any title for each passage, we report the average and its difference without these two datasets for the `Titles Only` ablation.

To check if the model has memorized the documents or queries, we try keeping only the document title and ID in the corpus, removing the content (`Title Only`). This ablation significantly degraded performance, indicating the model indeed relies on the provided content.

We also study how the number of few-shot examples in the prompt affects quality in Figure F.3. Increasing the number of examples improves the quality on the retrieval task, from 0.76 at zero-shot to 0.81 at 5-shots. Finally, qualitative analysis on the model outputs is provided in Appendix H.

6 Related Work

Many popular LCLM benchmarks and methods rely on synthetic tasks (Tay et al., 2021) such as the popular "Needle-in-a-Haystack" task (Kamradt, 2023) or its extension to multi-hop QA (Levy et al., 2024). While these evaluations are also scalable to arbitrarily long contexts, they do not fully capture the nuances of real-world retrieval or reasoning (Hsieh et al., 2024). Some recent benchmarks leverage existing NLP datasets for tasks such as extreme summarization and multi-document QA (Bai et al., 2023). However, these lack the dynamic scaling capabilities of synthetic benchmarks, which makes them difficult to adapt to very long contexts.

LongAlpaca (Chen et al., 2024) and LongBench-Chat (Bai et al., 2024) evaluate instruction-following under long context settings but contain relatively low task diversity and no examples beyond 100k context length. Similar to LOFT, Ada-LEval (Wang et al., 2024) proposes a length-adaptable benchmark; however, their tasks may not resemble real-world applications.

Liu et al. (2023) apply LCLMs to long-context QA using the top retrieved documents from Natural Questions, similar to using specialized models for RAG in LOFT. However, their analysis is limited to contexts that are under 10k tokens. We extend this type of evaluation of LCLMs to context lengths of up to 1M tokens in addition to using multiple modalities and additional forms of reasoning. Finally, our work connects to the field of generative retrieval (De Cao et al., 2020; Tay et al., 2022), where models are trained to memorize and generate retrieval targets. Our research offers an alternative approach where the retrieval corpus is directly provided as context, eliminating task specific training.

7 Conclusion

As language models improve and scale, their ability to retrieve and reason over increasingly long context will unlock unprecedented use-cases. To measure this progress, we introduce LOFT, the Long Context Frontiers benchmark. LOFT is a suite of tasks that rigorously assesses LCLMs on tasks ripe for a paradigm shift: retrieval, retrieval-augmented generation, reasoning over structured data, and in-context learning. LOFT provides dynamic scaling of context lengths of up to 1 million tokens, ensuring that evaluations remain relevant as LCLMs continue to evolve. Initial findings demonstrate that despite having never been trained to do retrieval, LCLMs have retrieval capabilities rivaling task-specific hand-crafted SOTA retrieval systems. Nevertheless, there remains considerable room for advancement in reasoning, particularly as models gain access to even longer context windows. We believe that LOFT provides a fertile testing ground for measuring progress in long-context modeling.

Limitations

Our experiments were constrained by the computational resources and financial costs associated with utilizing LCLMs. The entire LOFT 128k test sets contain around 35 datasets $\times$ 100 prompts $\times$ 128k tokens $=$ 448M input tokens, which cost $\$1,568$ for Gemini 1.5 Pro, $\$2,240$ for GPT-4o, and $\$6,720$ for Claude 3 Opus at the time of writing. To reduce costs, we also release dev sets, which are 10x smaller and can be evaluated with around $\$200$ using Gemini 1.5 Pro or GPT-4o. We also expect LLM API prices to decrease over time. Another limitation of this work is that we focused on evaluating the quality of LCLMs, and leave efficiency considerations for future work. We could not measure the efficiency improvements from prefix caching (Google, 2024) due to API constraints at the time of writing. Without caching, the Gemini 1.5 Pro API has a median latency of roughly four seconds for 32k input tokens, twelve seconds for 128k input tokens, and 100 seconds for 1 million input tokens. This speed is likely slower than specialized retrievers or SQL databases; the promising quality results on LOFT encourage further investigation into optimizing LCLMs efficiency. Additionally, our retrieval and RAG tasks was limited to 1 million tokens, which still leaves a large gap from real-world applications that may involve several million or even billions of documents.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating Correctness and Faithfulness of Instruction-Following Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 12:681–699.

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Samuel Joseph Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. 2022. Qampari: A benchmark for open-domain questions with many answers. In *IEEE Games Entertainment Media Conference*.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. Longalign: A recipe for long context alignment of large language models. *Preprint*, arXiv:2401.18058.

Yushi Bai, Xin Lv, Jiajie Zhang, Hong Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *ArXiv preprint*, abs/2308.14508.

Payal Bajaj, Daniel Fernando Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Minh Nguyen, Mir Rosenberg, Xia Song, Alina Mihaela Stoica, Saurabh Tiwary, and Tong Wang. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv: Computation and Language*.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. *ArXiv preprint*, abs/2401.05856.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Christian Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of touché 2020: Argument retrieval. In *Conference and Labs of the Evaluation Forum*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2024. Longlora: Efficient fine-tuning of long-context large language models. *Preprint*, arXiv:2309.12307.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi

Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. Fleurs: Few-shot learning evaluation of universal representations of speech. *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023. Text-to-sql empowered by large language models: A benchmark evaluation. *ArXiv preprint*, abs/2308.15363.

Frank Palma Gomez, Ramon Sanabria, Yun-hsuan Sung, Daniel Cer, Siddharth Dalmia, and Gustavo Hernandez Abrego. 2024. Transforming llms into cross-modal and cross-lingual retrievalsystems. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*.

Google. 2024. Context caching guide. `https://ai.google.dev/gemini-api/docs/caching`. Accessed: 2024-06-05.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *ArXiv*.

Hexiang Hu, Yi Luan, Yang Chen, Urvashi Khandelwal, Mandar Joshi, Kenton Lee, Kristina Toutanova, and Ming-Wei Chang. 2023. Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12031–12041.

Greg Kamradt. 2023. Needle in a haystack - pressure testing llms.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher Manning, and Kyoung-Gu Woo. 2022. You only need one model for open-domain question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3047–3060, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernández Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha R. Jonnalagadda, Ming-Wei

Chang, and Iftekhar Naim. 2024. Gecko: Versatile text embeddings distilled from large language models. *ArXiv preprint*, abs/2403.20327.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *ArXiv preprint*, abs/2402.14848.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *ArXiv preprint*, abs/2404.02060.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning. *ArXiv preprint*, abs/2305.14128.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. *Companion Proceedings of the The Web Conference 2018*.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14032–14047, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6997–7008, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *ArXiv preprint*, abs/2402.15449.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny

Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv preprint*, abs/2403.05530.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

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

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. 2024. Ada-leval: Evaluating long-context llms with length-adaptable benchmarks. *Preprint*, arXiv:2404.06480.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *ArXiv preprint*, abs/2201.11903.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5288–5296. IEEE Computer Society.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020a. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940, Online. Association for Computational Linguistics.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020b. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940, Online. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang,

and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019. SParC: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *ArXiv preprint*, abs/1709.00103.

A LOFT Dataset Creation

A.1 Dataset Selection

We initially performed a manual analysis on most of the datasets based on 20 sampled evaluation instances. Datasets that have poor annotation quality were not considered in LOFT. For instance, we only use a subset of datasets from BEIR (Thakur et al., 2021) due to their annotation quality.

Text Retrieval & RAG We test single-document retrieval on a representative subset of the BEIR benchmark (Thakur et al., 2021), prioritizing datasets with high-quality ground truth labels (Wachsmuth et al., 2018; Thorne et al., 2018; Maia et al., 2018; Bajaj et al., 2016; Kwiatkowski et al., 2019; Wadden et al., 2020). We also include TopiOCQA (Adlakha et al., 2022), which is a multi-turn conversational retrieval dataset. We measure performance on single-document retrieval using Recall@1. Additionally, we test multi-document retrieval on HotPotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), QAMPARI (Malaviya et al., 2023), where a set of documents must be retrieved to answer the query. The evaluation metric for multi-document retrieval is MRecall@k (Min et al., 2021), which gives a score of 1.0 if all k gold set items are retrieved in top-k and 0.0 otherwise. When creating the LOFT version of the multi-document retrieval datasets, we limit the number of relevant documents per query to $k = 2$, 5, 5, and 3 for HotPotQA, MuSiQue, QAMPARI, and QUEST, respectively, and the corresponding k's are used for MRecall@k (e.g. HotPotQA uses MRecall@2).

Our RAG task contains subsets of retrieval datasets, which have phrase-level answer annotations: Natural Questions, TopiOCQA, HotPotQA, MuSiQue, QAMPARI, and QUEST. We use subspan exact match (EM) (Adlakha et al., 2024) for evaluating performance of all the datasets. In case of multi-answer datasets (i.e. QAMPARI, QUEST), we first match predicted answers to gold standard answers based on whether they overlap (Dua et al., 2019) via linear sum assignment algorithm. We then give full credit if every gold answer has a perfect match with aligned predicted answers.

Visual Retrieval We employ four diverse visual benchmarks: Flickr30k (Young et al., 2014) and MSCOCO (Lin et al., 2014) for text-to-image retrieval; MSR-VTT (Xu et al., 2016) for text-to-video retrieval (sampling 3 frames per video); and OVEN (Hu et al., 2023) using the entity split for image-text retrieval where both queries and retrieval targets consist of image-text pairs. All images are resized to 512x512 and performance is assessed using Recall@1 for all datasets.

Audio Retrieval We utilize a subset of the multilingual FLEURS dataset (Conneau et al., 2022), focusing on the five most spoken languages[8]: English (en), Hindi (hi), Chinese (zh), Spanish (es), and French (fr). Recall@1 is employed as the evaluation metric, given the single gold target.

SQL We evaluate SQL-like reasoning on Spider, a single-turn text-to-SQL dataset (Yu et al., 2018), and SparC, its multi-turn variant (Yu et al., 2019). The input contains the database tables serialized as CSV and the natural language question. The model is allowed to perform reasoning in natural language before giving the final answer, which must be formatted in a Markdown code block. The extracted answers are evaluated against the execution results of the gold SQL queries. For SparC, the multi-turn questions are provided one-by-one in a conversational format, and credit is awarded only when the answers of all steps are correct.

Many-shot ICL We investigate LCLMs' many-shot ICL capabilities by repurposing datasets from Big Bench Hard (BBH) (bench authors, 2023; Suzgun et al., 2023) and LongICLBench (LIB) (Yu et al., 2020b; Li et al., 2024) to fit a many-shot ICL setting, focusing on multi-class classification tasks. The first set of datasets is drawn from Big-Bench Hard and includes: `date understanding` (BBH-date), `salient error translation detection` (BBH-salient), `tracking shuffled objects seven objects` (BBH-tracking7), and `web of lies` (BBH-web), each with up to 150 examples for prompting and up to 7 classes. Unlike other LOFT tasks, the full corpus fits within 32k tokens which leads us to also create variants from 2k to 32k context lengths. We use accuracy as our metric for Big Bench Hard. We also evaluate with DialogRE (Yu et al., 2020b), a dialogue-based relation classification dataset with 36 relation labels. We follow the LongICLBench format but use accuracy as our metric.

[8] https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

Task	Dataset	Description	Avg. Cand. Length	# Cand. (128k)	Candidates	Input	Target
Text Retrieval	ArguAna	Argument Retrieval	196	531	Passages	Query	Passage ID(s)
	FEVER	Fact Checking	176	588			
	FIQA	Question Answering	196	531			
	MS MARCO	Web Search	77	1,174			
	NQ	Question Answering	110	883			
	Quora	Duplication Detection	14	3,306			
	SciFact	Citation Prediction	301	357			
	Touché-2020	Argument Retrieval	330	329			
	TopiOCQA	Multi-turn QA	149	680			
	HotPotQA	Multi-hop QA	74	1,222			
	MuSiQue	Multi-hop QA	120	824			
	QAMPARI	Multi-target QA	132	755			
	QUEST	Multi-target QA	328	328			
Visual Retrieval	Flickr30k	Image Retrieval	258	440	Images	Text Query	Image ID
	MS COCO	Image Retrieval	258	440	Images	Text Query	Image ID
	OVEN	Image-text Retrieval	278	448	Images+Texts	Image+Text Query	Wikipedia ID
	MSR-VTT	Video Retrieval	774	140	Videos	Text Query	Video ID
Audio Retrieval	FLEURS-en	Audio Retrieval	249	428	Speech	Text Query	Speech ID
	FLEURS-es		315	343			
	FLEURS-fr		259	412			
	FLEURS-hi		292	369			
	FLEURS-zh		291	370			
RAG	NQ	Question Answering	110	883	Passages	Question	Answer(s)
	TopiOCQA	Multi-turn QA	149	680			
	HotPotQA	Multi-hop QA	74	1,222			
	MuSiQue	Multi-hop QA	120	824			
	QAMPARI	Multi-target QA	132	755			
	QUEST	Multi-target QA	328	328			
SQL	Spider	Single-turn SQL	111k	1	SQL Database	Question	Answer
	SParC	Multi-turn SQL	111k	1			
Many-Shot ICL	BBH-date	Multiple-choice QA	131	150	Training Examples	Question	Answer
	BBH-salient	Multiple-choice QA	246	104			
	BBH-tracking7	Multiple-choice QA	205	123			
	BBH-web	Multiple-choice QA	43	150			
	LIB-dialogue	Classification	266	274			

Table A.1: Tasks and datasets in the LOFT benchmark. LOFT has 6 types of tasks, 4 modalities, and 35 datasets in total. For each dataset, we show the average length of the candidates (Avg. Cand. Length) as well as the number of candidates (# Cand) in the 128k version. More details on the datasets are available in Appendices A to C.

B Datasets Processing Details

Content Filtering The language model APIs often block inputs with potentially harmful contents. When creating LOFT, we tried to remove such contents from textual and visual inputs. Our filtering was done using a classifier as well as a keyword-based filtering. Despite our best effort, some API calls still refused to provide answers, which we treated as incorrect in our evaluation.

Tokenization To measure the size of a corpus, we count the number of tokens returned by the SentencePiece tokenizer (Kudo and Richardson, 2018).

Links to Dataset Sources LOFT repurposes existing datasets for evaluating LCLMs. Here are the links to the original datasets used in LOFT.

- Text Retrieval - BEIR (Thakur et al., 2021) (ArguAna (Wachsmuth et al., 2018), FEVER (Thorne et al., 2018), FIQA (Maia et al., 2018), MS MARCO (Bajaj et al., 2016), NQ (Kwiatkowski et al., 2019), Quora, SciFact (Wadden et al., 2020), Touché-2020 (Bondarenko et al., 2020), HotPotQA (Yang et al., 2018)): https://github.com/beir-cellar/beir
- Text Retrieval - TopiOCQA (Adlakha et al., 2022): https://github.com/McGill-NLP/topiocqa
- Text Retrieval - MuSiQue (Trivedi et al., 2022): https://allenai.org/data/musique
- Text Retrieval - QAMPARI (Amouyal et al., 2022): https://github.com/samsam3232/qampari
- Text Retrieval - QUEST (Malaviya et al., 2023): https://github.com/google-research/language/tree/master/language/quest
- Visual Retrieval - Flickr30k (Young et al., 2014): https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset
- Visual Retrieval - MS COCO (Lin et al., 2014): https://cocodataset.org

Task	Dataset	# Queries (Few-shot / Development / Test)	Supported Context Length	# Candidates
Text Retrieval	ArguAna	5 / 10 / 100	32k / 128k / 1M	123 / 531 / 3,891
	FEVER	5 / 10 / 100	32k / 128k / 1M	154 / 588 / 6,031
	FIQA	5 / 10 / 100	32k / 128k / 1M	148 / 531 / 4,471
	MS MARCO	5 / 10 / 100	32k / 128k / 1M	302 / 1,174 / 9,208
	NQ	5 / 10 / 100	32k / 128k / 1M	214 / 883 / 6,999
	Quora	5 / 10 / 100	32k / 128k / 1M	820 / 3,306 / 25,755
	SciFact	5 / 10 / 100	32k / 128k / 1M	86 / 357 / 2,753
	Touché-2020	5 / 10 / 34	32k / 128k / 1M	77 / 329 / 2,843
	TopiOCQA	5 / 10 / 100	32k / 128k / 1M	170 / 680 / 5,379
	HotPotQA	5 / 10 / 100	32k / 128k / 1M	319 / 1,222 / 10,005
	MuSiQue	5 / 10 / 100	32k / 128k / 1M	210 / 824 / 6,650
	QAMPARI	5 / 10 / 100	32k / 128k / 1M	186 / 755 / 5,878
	QUEST	5 / 10 / 100	32k / 128k / 1M	87 / 328 / 2,858
Visual Retrieval	Flickr30k	5 / 10 / 100	32k / 128k	115 / 440
	MS COCO	5 / 10 / 100	32k / 128k / 1M	115 / 440 / 3,448
	OVEN	5 / 10 / 100	32k / 128k / 1M	110 / 448 / 3475
	MSR-VTT	5 / 10 / 100	32k / 128k / 1M	35 / 140 / 1,101
Audio Retrieval	FLEURS-en	5 / 10 / 100	32k / 128k	104 / 428
	FLEURS-es	5 / 10 / 100	32k / 128k	77 / 343
	FLEURS-fr	5 / 10 / 100	32k / 128k	94 / 412
	FLEURS-hi	5 / 10 / 100	32k / 128k	83 / 369
	FLEURS-zh	5 / 10 / 100	32k / 128k	85 / 370
RAG	NQ	5 / 10 / 100	32k / 128k / 1M	214 / 883 / 6,999
	TopiOCQA	5 / 10 / 100	32k / 128k / 1M	170 / 680 / 5,379
	HotPotQA	5 / 10 / 100	32k / 128k / 1M	319 / 1,222 / 10,005
	MuSiQue	5 / 10 / 100	32k / 128k / 1M	210 / 824 / 6,650
	QAMPARI	5 / 10 / 100	32k / 128k / 1M	186 / 755 / 5,878
	QUEST	5 / 10 / 100	32k / 128k / 1M	87 / 328 / 2,858
SQL	Spider	1 / 10 / 100	32k / 128k / 1M	1 / 1 / 1
	SParC	1 / 10 / 100	32k / 128k / 1M	1 / 1 / 1
Many-Shot ICL	BBH-date	- / 10 / 90	32k	150
	BBH-salient	- / 10 / 90	32k	104
	BBH-tracking7	- / 10 / 90	32k	123
	BBH-web	- / 10 / 90	32k	150
	LIB-dialogue	- / 10 / 100	32k / 128k / 1M	61 / 274 / 1,059

Table C.1: Tasks and datasets in the LOFT benchmark. We show the number of queries per each split and and supported context lengths for each dataset.

- Visual Retrieval - OVEN (Hu et al., 2023): https://github.com/open-vision-language/oven
- Visual Retrieval - MSR-VTT (Xu et al., 2016): https://cove.thecvf.com/datasets/839
- Audio Retrieval - FLEURS (Conneau et al., 2022): https://huggingface.co/datasets/google/fleurs
- RAG - Same as Text Retrieval
- SQL - Spider (Yu et al., 2018): https://yale-lily.github.io/spider
- SQL - SparC (Yu et al., 2019): https://yale-lily.github.io/sparc
- Many-Shot ICL - Big-Bench Hard (bench authors, 2023; Suzgun et al., 2023): https://github.com/suzgunmirac/BIG-Bench-Hard
- Many-Shot ICL - LongICLBench (Yu et al., 2020b; Li et al., 2024): https://github.com/TIGER-AI-Lab/LongICLBench

C Detailed Statistics

In Table C.1, we show detailed statistics of the LOFT benchmark.

D Prompt Design

D.1 Dataset Instructions

In Table D.1, we show each instruction used for each dataset in LOFT.

D.2 Chain-of-Thought Reasoning

We used two types of chain-of-thought reasoning in the few-shot examples in the prompts:

- **Handwritten reasoning.** For multi-hop retrieval and RAG datasets (HotPotQA and MuSiQue), as well as SQL datasets (Spider and SparC), we manually wrote reasoning chains for the few-shot queries.
- **Relevant content.** For the rest of the retrieval and RAG datasets, we simply let the model generate the title (or the passage if the title is not available) of each gold document before generating the final answers. This helps explain why the final answers (IDs or short phrases) are the correct ones.

Dataset	Instruction
	Text Retrieval
ArguAna	You will be given a list of statements. You need to read carefully and understand all of them. Then you will be given a claim, and your goal is to find all statements from the list that can counterargue the claim.
FEVER Scifact	You will be given a list of passages. You need to read carefully and understand all of them. Then you will be given a claim, and your goal is to find all passages from the list that can help verify the claim as true of false.
FIQA MS MARCO NQ, TopiOCQA	You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query, and your goal is to find all documents from the list that can help answer the query.
Quora	You will be given a list of questions. You need to read carefully and understand all of them. Then you will be given a new question, and your goal is to find all questions from the list that are near duplicates of the new question.
Touché-2020	You will be given a list of arguments. You need to read carefully and understand all of them. Then you will be given a controversial debating topic, and your goal is to find arguments from the list that's relevant to the topic.
HotPotQA MuSiQue QAMPARI, QUEST	You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is to find all documents from the list that can help answer the query.
	Visual Retrieval
Flickr30k MS COCO	You will be given a list of images. You need to carefully watch all of them. Then you will be given a new sentence, and your goal is to find most relevant image from the list for the given sentence.
OVEN	You will be given a list of Wikipedia entries which contains Wikipedia ID, Title and Description image. You need to carefully watch all of them. Then you will be given a input image and a question related to the image, and your goal is to find most relevant Wikipedia entry from the list that can be used to best answer the question.
MSR-VTT	You will be given a list of videos which contains the video ID and video content (present as sequence of images, with timestamp in text). You need to carefully watch all of them. Then you will be given a text query, and your goal is to find most relevant video from the list that can best answer the question.
	Audio Retrieval
FLEURS-*	You will be given a list of audio which contains Audio ID and audio. You need to carefully listen all of them. Then you will be given a transcript, and your goal is to find most relevant audio from the list that matches the given transcript. Print out the Audio ID of the audio presented in the list.
	SQL
Spider SparC	You will be given a list of tables. You need to read all of the rows of each table. Then you will be given a query, and your goal is to get the answer from the tables. Then format the answer into a list of lists. When formatting the answer into a list of lists, make sure you use the exact fields that are provided in the tables.

Table D.1: Instructions used for each LOFT dataset. We omit instructions for the RAG datasets, which are almost identical to text retrieval instructions. The ICL task does not use additional instructions, but only many-shot examples in their context.

Dataset	Dev (32k)	Test (128k)
NQ	0.60 (-0.10)	0.37 (-0.47)
HotPotQA	0.60 (-0.30)	0.33 (-0.42)
MuSiQue	0.20 (-0.60)	0.10 (-0.45)

Table F.1: **Gemini's closed-book performance on RAG**. Red indicates the performance difference compared to the CiC prompting.

Many-shot ICL task did not use any chain-of-thought reasoning. We also noticed that GPT-4o performed much better without chain-of-thought reasoning, so the GPT-4o results presented in this paper did not utilize the chain-of-thought.

E Positional Analysis Detailed Results

In Figure E.1, we show the detailed positional analysis on LOFT.

F Ablation Study

In Table F.1, we show the closed-book performance of Gemini on LOFT-RAG datasets. In Figure F.1, we show the performance of Gemini per SQL operator on LOFT-Spider. In Figure F.2, we show the effect of the number of examples in LOFT-ICL. In Figure F.3, we show the effect of number of few-shot examples for LOFT.

G Ablated Prompt Examples

In Figure G.1, we show examples of task prompts in LOFT. In Figures G.2 to G.8, we show prompt examples for the ablation study.

H Qualitative Analysis of Long-Context Wins and Losses

In Tables H.1 to H.3, we show examples of wins and losses of long-context models on LOFT.

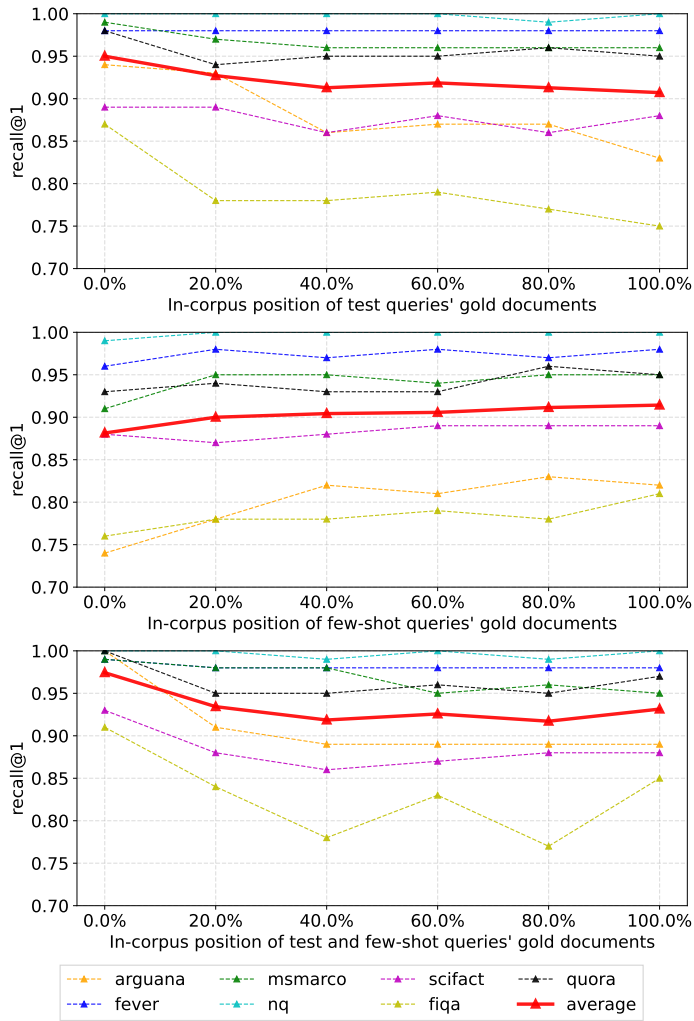

Figure E.1: Detailed metrics of the positional analysis, where we vary the position of gold documents of the test and few-shot queries. ***Top:*** we vary the gold documents position of test queries within the corpus. ***Middle:*** we vary the the gold documents position of few-shot queries within the corpus. ***Bottom:*** we group the gold documents of test and few-shot queries together, and vary their position within the corpus. The average is shown in red.

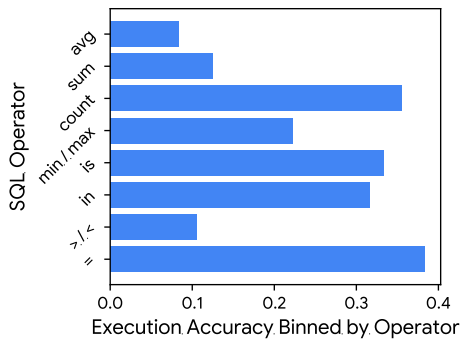

Figure F.1: **SQL Reasoning Analysis**. We bin Spider queries by operators in their SQL query and report binned Gemini performance. We group `min` and `max` into a bin and > and < into another bin.

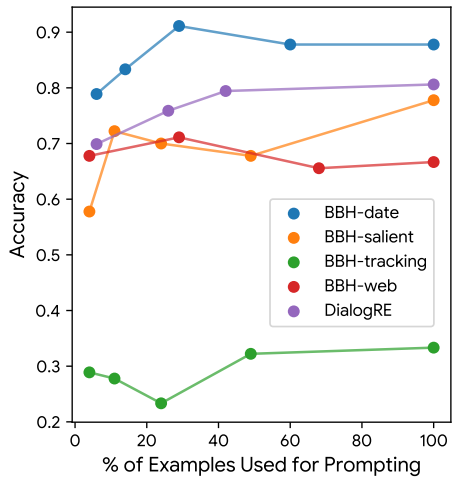

Figure F.2: **ICL Performance** as we scale the percentage of examples used up to 100%.

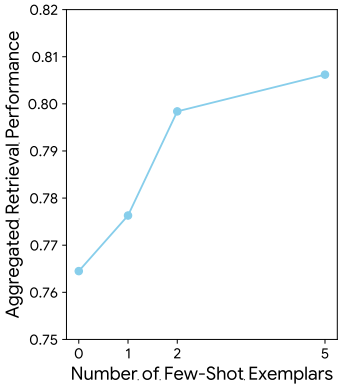

Figure F.3: **Effect of the number of few-shot examples**. The performance increases with the number of few-shot examples.

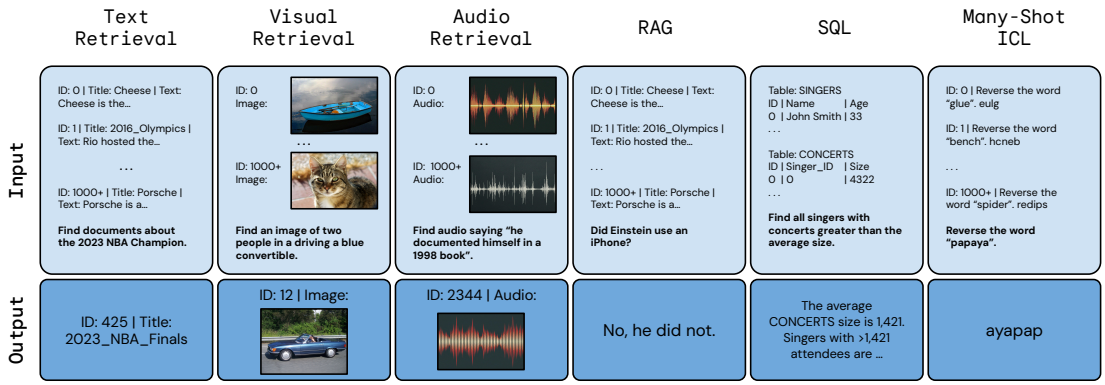

Figure G.1: Examples of the task prompts in LOFT. Each LCLM is expected to do in-context retrieval, reasoning, and many-shot learning on corpora up to millions of tokens.

```
You will be given a list of documents. You need to read carefully and understand all of them. Then you
will be given a query that may require you to use 1 or more documents to find the answer. Your goal is
to find all documents from the list that can help answer the query.
```

Instruction

```
ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…
```

Corpus Formatting

```
====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…
```

Few-shot Exemples

```
====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: How many records had the team sold before performing "aint thinkin bout you"?
The following documents are needed to answer the query:
```

Query Formatting

Figure G.2: Original CiC prompt for HotPotQA, a retrieval dataset in LOFT. The prompt contains an instruction, a corpus, few-shot examples and a query.

You will be given a list of candidates such as documents, images, videos, audios, etc. You need to check them carefully and understand all of them. Then you will be given a query, and your goal is to find all candidates from the list that can help answer the query. Print out the ID of each candidate.

Generic Instruction (valid for all datasets)

ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…

Corpus Formatting

====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…

Few-shot Exemples

====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: **How many records had the team sold before performing "aint thinkin bout you"?**
The following documents are needed to answer the query:

Query Formatting

Figure G.3: Generic Instruction Ablation, with changes to the original CiC prompt in red. The instruction is changed to a generic one that applies to all tasks in LOFT.

======You need to answer the following query. ======
Which document is most relevant to answer the query? Print out the TITLE and ID of the document. Then format the IDs into a list in the following format: [id1, id2, ...].
If there is no perfect answer output the closest one. Do not give an empty final answer.
query: **How many records had the team sold before performing "aint thinkin bout you"?**
====Here are the context you need to read to answer the query. =====

Query at the beginning

You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is to find all documents from the list that can help answer the query.

Instruction

ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…

Corpus Formatting

====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…

Few-shot Exemples

===== Now answer the query! =====

Start Answer

Figure G.4: Query at the Beginning Ablation, with changes to the original CiC prompt in red. The query is placed at the beginning instead of the end.

You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is to find all documents from the list that can help answer the query.

Instruction

ID: D5Y5 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: D5Y5
…
ID: y2h8 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: y2h8
ID: E8J2 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: E8J2
…

Corpus Formatting

====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: E8J2
TITLE: Shinji Okazaki | ID: D5Y5
Final Answer: [E8J2, D5Y5]
…

Few-shot Exemples

====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: **How many records had the team sold before performing "aint thinkin bout you"?**
The following documents are needed to answer the query:

Query Formatting

Figure G.5: Alphanumeric Document ID Ablation, with changes to the original CiC prompt in red. Instead of using sequential numeric document IDs, a unique random alphanumeric ID is generated with alternating ASCII letters and digits.

You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query that may require you to use 1 or more documents to find the answer. Your goal is to find all documents from the list that can help answer the query.

Instruction

ID: 0 | TITLE: Shinji Okazaki | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | END ID: 54
…

Corpus Formatting (CONTENT removed)

====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…

Few-shot Exemples

====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format the IDs into a list.
query: **How many records had the team sold before performing "aint thinkin bout you"?**
The following documents are needed to answer the query:

Query Formatting

Figure G.6: Title Only Ablation, with changes to the original CiC prompt in red. In this ablation, the document content is removed, keeping only the document title.

```
You will be given a list of documents. You need to read carefully and understand all of them. Then you
will be given a query that may require you to use 1 or more documents to find the answer. Your goal is
to find all documents from the list that can help answer the query.
```

Instruction

```
ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese …
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song …
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki …
…
```

Corpus Formatting (END ID removed)

```
====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 54
TITLE: Shinji Okazaki | ID: 0
Final Answer: [54, 0]
…
```

Few-shot Exemples

```
====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: How many records had the team sold before performing "aint thinkin bout you"?
The following documents are needed to answer the query:
```

Query Formatting

Figure G.7: ID Echo Ablation, with changes to the original CiC prompt in red. In this ablation, the ID is only mentioned at the beginning of each document, and we remove the ID echo at the end (e.g. "END ID:").

```
You will be given a list of documents. You need to read carefully and understand all of them. Then you
will be given a query that may require you to use 1 or more documents to find the answer. Your goal is
to find all documents from the list that can help answer the query.
```

General Instruction

```
====== Example 1 ======
ID: 0 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 0
ID: 1 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 1
…
ID: 9 | TITLE: The Lodger (2009 film) | CONTENT: The Lodger is … | END ID: 9
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then
format the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
TITLE: Best Footballer in Asia 2016 | ID: 0
TITLE: Shinji Okazaki | ID: 1
Final Answer: [0, 1]
…
```

Few-shot Exemples With Sampled Corpus

```
ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…
```

Corpus Formatting

```
====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: How many records had the team sold before performing "aint thinkin bout you"?
The following documents are needed to answer the query:
```

Query Formatting

Figure G.8: Corpus in Each Few-shot Ablation, with changes to the original CiC prompt in red. In particular, in this ablation, each few-shot example contains a sampled corpus (10 documents), the full corpus is then given before the Query part of the prompt.

```
You will be given a list of documents. You need to read carefully and understand all of them. Then you
will be given a query that may require you to use 1 or more documents to find the answer. Your goal is
to find all documents from the list that can help answer the query.
```

General Instruction

```
ID: 0 | TITLE: Shinji Okazaki | CONTENT: Shinji Okazaki is a Japanese … | END ID: 0
…
ID: 53 | TITLE: Ain't Thinkin' 'Bout You | CONTENT: "Ain't Thinkin' 'Bout You" is a song … | END ID: 53
ID: 54 | TITLE: Best Footballer in Asia 2016 | CONTENT: … was awarded to Shinji Okazaki … | END ID: 54
…
```

Corpus Formatting

```
====== Example 1 ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: What year was the recipient of the 2016 Best Footballer in Asia born?
The following documents are needed to answer the query:
Final Answer: [54, 0]
…
```

Few-shot Exemples (no CoT)

```
====== Now let's start! ======
Which documents are needed to answer the query? Print out the TITLE and ID of each document. Then format
the IDs into a list.
query: How many records had the team sold before performing "aint thinkin bout you"?
The following documents are needed to answer the query:
```

Query Formatting

Figure G.9: No Chain-of-Thought Ablation, with changes to the original CiC prompt in red. In particular, in this ablation, we remove the chain-of-thought from the few-shot examples, instead opting to output the answer directly.

Task = **Retrieval**	
Dataset = **Natural Questions** 1M	
Input	`CiC-prompt`(query="when does monday night raw come on hulu")
Output	The following documents can help answer the query: TITLE: WWE Raw | ID: 1023 Final Answer: ['1023'] (✓)
Input	`CiC-prompt`(query="where does the synthesis of new dna from existing dna occurs")
Output	The following documents can help answer the query: TITLE: DNA replication | ID: 3355 Final Answer: ['3355'] (✓)
Input	`CiC-prompt`(query="jawaharlal nehru centre for advanced scientific research jakkur campus")
Output	TITLE: Jawaharlal Nehru Centre for Advanced Scientific Research | ID: 6911 Final Answer: ['6911'] (✗) // Incorrect ID (gold = '5691')
Dataset = **HotPotQA** 128k (multi-hop)	
Input	`CiC-prompt`(query="Which internationl football team has had a player endorse the Nike Hypervenom football boot?")
Output	The following documents can help answer the query: reasoning: The Nike Hypervenom is endorsed by players such as Marcus Rashford based on the document **ID:** 413 (title: Nike Hypervenom). reasoning: Marcus Rashford plays for the England national team based on the document **ID:** 558 (title: Marcus Rashford). Final Answer: ['413', '558'] (✓)
Input	`CiC-prompt`(query="What NHL team drafted Kaspars Astasenko in 1999 and won the Stanley Cup in 2003-04?")
Output	The following documents can help answer the query: ID: 217 | TITLE: Kaspars Astašenko | CONTENT: Kaspars Astašenko (17 February 1975 – 20 November 2012) was a Latvian professional ice hockey player. Astašenko was born in Riga, Latvia. Astašenko was drafted by the Tampa Bay Lightning in the 1999 NHL Entry Draft, 127th overall. Astašenko played parts of two seasons in the National Hockey League with the Lightning. | END ID: 217 ID: 508 | TITLE: Tampa Bay Lightning | CONTENT: The Tampa Bay Lightning are a professional ice hockey team based in Tampa, Florida. It is a member of the Atlantic Division of the Eastern Conference of the National Hockey League (NHL). The Lightning have one Stanley Cup championship in their history, in 2003–04. The team is often referred to as the Bolts, and the nickname is used on the current third jersey. The Lightning plays home games in the Amalie Arena in Tampa. | END ID: 508 Final Answer: ['217', '508'] (✓)
Input	`CiC-prompt`(query="What drum machine was introduced in 1989 and allowed 32 notes to be played simultaneously?")
Output	The following documents can help answer the query: ID: 773 | TITLE: Roland R-8 | CONTENT: The R-8 Human Rhythm Composer is an electronic drum machine introduced in 1989 by Roland Corporation, using PCM voices. ... The device has eight individual outputs, 32-voice polyphony, and four-part multitimbral MIDI. | END ID: 773 Final Answer: ['773'] (✗) // Partially correct ('666' is missing)

Table H.1: Qualitative examples showcasing the successes and failures of the CiC prompting in the text retrieval task. We use Gemini 1.5 Pro for the analysis. For HotPotQA, we observed that some of the questions are actually answerable from a single document as illustrated in the last example.

Task = **RAG**	
Dataset = **MuSiQue** 128k (multi-hop)	
Input	CiC-prompt(query="Where do greyhound buses leave from in the city the performer of Born A Lion formed in?")
Output	reasoning: Born A Lion is the debut album by Canadian rock band Danko Jones. reasoning: Danko Jones is a Canadian rock trio from Toronto, Ontario. reasoning: The Toronto Coach Terminal is the central bus station for inter-city services in Toronto, Ontario, Canada. ID: 427 | TITLE: Born a Lion ID: 383 | TITLE: Danko Jones ID: 724 | TITLE: Toronto Coach Terminal Final Answer: ['Toronto Coach Terminal'] (✓)
Input	CiC-prompt(query="When did the rx 350 model of the luxury division of the company that built Daihatsu boon change body style?")
Output	reasoning: The Daihatsu Boon is also sold as the Toyota Passo. reasoning: The Passo is sold at "Toyota Corolla Store" Japanese dealerships. reasoning: Scion is a discontinued marque of Toyota. reasoning: The Lexus RX is a mid-size luxury crossover SUV sold since 1998 by Lexus, a luxury division of Toyota. reasoning: The first generation Lexus RX was introduced in 1998 and was produced until 2003. reasoning: The second generation Lexus RX was introduced in 2003 and was produced until 2009. reasoning: The third generation Lexus RX was introduced in 2008 and was produced until 2015. reasoning: The fourth generation Lexus RX was introduced in 2015. reasoning: The RX 350 model was introduced in the second generation Lexus RX in 2003. Final Answer: ['2003', '2009', '2015'] (✗) // Incorrect (answer = 'Sales began worldwide in April 2012')
Dataset = **QUEST** 128k (multi-answer)	
Input	CiC-prompt(query="History books about Malaysia or the Qing dynasty or about South Korea")
Output	reasoning: Chronicle of Malaysia gives an account of Malaysia from January 1957 to 31 August 2007. reasoning: Treason by the Book is a historical account of the Zeng Jing case which took place during the reign of the Yongzheng Emperor of Qing China around 1730. reasoning: Korea: A Walk Through the Land of Miracles is a book by Simon Winchester. He recounts his experience walking across South Korea, from Jeju in the south to the DMZ in the north. Final Answer: ['Chronicle of Malaysia', 'Korea: A Walk Through the Land of Miracles', 'Treason by the Book'] (✓)
Input	CiC-prompt(query="Films about sisters shot in Paris")
Output	reasoning: Peppermint Soda is a 1977 French comedy-drama film directed by Diane Kurys. The film follows two teenage sisters over the course of the year 1963. The high school where the film takes place is the Lycée Jules-Ferry in Paris, France. Final Answer: ['Peppermint Soda'] (✗) // Partially correct (missing 'Le Divorce' and 'Two English Girls')

Table H.2: Qualitative examples showcasing the successes and failures of the CiC prompting in the RAG task. We use Gemini 1.5 Pro for the analysis. In each example, we underline correct reasoning chains provided by the model.

Task = **SQL**	
Dataset = **Spider** 1M	
Input	`CiC-prompt`(query="Find the year which offers the largest number of courses.")
Output	Here's a step-by-step approach using the provided tables: **1. Identify relevant data:** We need to look at the "year" column in the "section" table, which lists the year each course section was offered. **2. Count courses per year:** Group the data by "year" and count the number of unique "course_id" values for each year. This will give us the number of distinct courses offered in each year. **3. Find the year with the maximum count:** Identify the year with the highest count of distinct courses. **Based on the data provided, here's the breakdown of courses per year:** | Year | Number of Courses | |—|—| | 2001 | 29 | | 2002 | 36 | | 2003 | 38 | | 2004 | 31 | | 2005 | 31 | | 2006 | 41 | | 2007 | 40 | | 2008 | 37 | | 2009 | 35 | | 2010 | 39 | **The year 2006 has the largest number of courses offered (41).** Final Answer: ['2006'] (✓)
Input	`CiC-prompt`(query="List total amount of invoice from Chicago, IL.")
Output	Here's a step-by-step approach using the provided tables: **1. Identify relevant data:** * **Invoices:** This table contains the billing city and total for each invoice. **2. Filter invoices from Chicago, IL:** * Scan through the "billing_city" column and identify invoices where the city is "Chicago" and the state is "IL". * **Based on the data provided, invoices 24, 92, 103, 158, 287, 310, 332, and 384 are from Chicago, IL.** **3. Sum the total amounts:** * Add up the "total" values for the invoices identified in step 2. * **The total amounts for the invoices are 1.98, 1.98, 15.86, 8.91, 1.98, 7.96, 5.94, and 0.99, respectively.** * **The sum of these amounts is 36.6.** **4. Present the result:** * The final output will be a single value representing the total amount of invoices from Chicago, IL. Final Answer: ['36.6'] (✗)

Table H.3: Qualitative examples showcasing the successes and failures of the CiC prompting in the SQL task. We use Gemini 1.5 Pro for the analysis. In SQL, the long-context model first outputs a reasoning chain in natural language ideally simulating the execution of the SQL query before arriving at the prediction.